RICARDO DIOGO RIGHETTO

VALIDATION OF STRUCTURAL HETEROGENEITY IN CRYO-EM DATASETS BY CLUSTER ENSEMBLES

VALIDAÇÃO DE HETEROGENEIDADE ESTRUTURAL EM DADOS DE CRIO-ME POR COMITÊS DE AGRUPADORES

CAMPINAS
2014

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

RICARDO DIOGO RIGHETTO

VALIDATION OF STRUCTURAL HETEROGENEITY IN CRYO-EM DATASETS BY CLUSTER ENSEMBLES

Orientador: Prof. Dr. Fernando José Von Zuben
Coorientador: Dr. Rodrigo Villares Portugal

VALIDAÇÃO DE HETEROGENEIDADE ESTRUTURAL EM DADOS DE CRIO-ME POR COMITÊS DE AGRUPADORES

Master dissertation presented to the Electrical Engineering Graduate Program of the School of Electrical and Computer Engineering of the University of Campinas to obtain the M.Sc. grade in Electrical Engineering, in the field of Computer Engineering.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Elétrica, na área de Engenharia de Computação.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO RICARDO DIOGO RIGHETTO E ORIENTADO PELO PROF. DR. FERNANDO JOSÉ VON ZUBEN

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

Righetto, Ricardo Diogo, 1986-
R449v Validation of structural heterogeneity in Cryo-EM datasets by cluster ensembles / Ricardo Diogo Righetto. – Campinas, SP : [s.n.], 2014.

Orientador: Fernando José Von Zuben.
Coorientador: Rodrigo Villares Portugal.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Aprendizado do computador. 2. Microscopia eletrônica de transmissão. 3. Biologia estrutural. 4. Mineração de dados (Computação). 5. Análise de cluster. I. Von Zuben, Fernando José,1968-. II. Portugal, Rodrigo Villares. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Validação de heterogeneidade estrutural em dados de Crio-ME por comitês de agrupadores
Palavras-chave em inglês:
Machine learning
Transmission electron microscopy
Cluster analysis
Cryoelectron microscopy
Molecular biology
Área de concentração: Engenharia de Computação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Fernando José Von Zuben [Orientador]
Marin van Heel
Eduardo Alves do Valle Junior
Data de defesa: 08-08-2014
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Ricardo Diogo Righetto

Data da Defesa: 8 de agosto de 2014

Título da Tese: "Validation of Structural Heterogeneity in Cryo-EM Datasets by Cluster Ensembles"

Prof. Dr. Fernando José Von Zuben (Presidente): Fernando José Von Zuben

Prof. Dr. Marin van Heel:

Prof. Dr. Eduardo Alves do Valle Junior: Eduardo A. do Valle Jr.

Abstract

Single Particle Analysis is a technique that allows the study of the three-dimensional structure of proteins and other macromolecular assemblies of biological interest. Its primary data consists of transmission electron microscopy images from multiple copies of the molecule in random orientations. Such images are very noisy due to the low electron dose employed. Reconstruction of the macromolecule can be obtained by averaging many images of particles in similar orientations and estimating their relative angles. However, heterogeneous conformational states often co-exist in the sample, because the molecular complexes can be flexible and may also interact with other particles. Heterogeneity poses a challenge to the reconstruction of reliable 3D models and degrades their resolution. Among the most popular algorithms used for structural classification are k-means clustering, hierarchical clustering, self-organizing maps and maximum-likelihood estimators. Such approaches are usually interlaced with the reconstructions of the 3D models. Nevertheless, recent works indicate that it is possible to infer information about the structure of the molecules directly from the dataset of 2D projections. Among these findings is the relationship between structural variability and manifolds in a multidimensional feature space. This dissertation investigates whether an ensemble of unsupervised classification algorithms is able to separate these "conformational manifolds". Ensemble or "consensus" methods tend to provide more accurate classification and may achieve satisfactory performance across a wide range of datasets, when compared with individual algorithms. We investigate the behavior of six clustering algorithms both individually and combined in ensembles for the task of structural heterogeneity classification. The approach was tested on synthetic and real datasets containing a mixture of images from the Mm-cpn chaperonin in the "open" and "closed" states. It is shown that cluster ensembles can provide useful information in validating the structural partitionings independently of 3D reconstruction methods.

Keywords: Single Particle Analysis, Transmission Electron Microscopy, Cryo-EM, Clustering Algorithms, Unsupervised Classification, Consensus Clustering.

Resumo

Análise de Partículas Isoladas é uma técnica que permite o estudo da estrutura tridimensional de proteínas e outros complexos macromoleculares de interesse biológico. Seus dados primários consistem em imagens de microscopia eletrônica de transmissão de múltiplas cópias da molécula em orientações aleatórias. Tais imagens são bastante ruidosas devido à baixa dose de elétrons utilizada. Reconstruções 3D podem ser obtidas combinando-se muitas imagens de partículas em orientações similares e estimando seus ângulos relativos. Entretanto, estados conformacionais heterogêneos frequentemente coexistem na amostra, porque os complexos moleculares podem ser flexíveis e também interagir com outras partículas. Heterogeneidade representa um desafio na reconstrução de modelos 3D confiáveis e degrada a resolução dos mesmos. Entre os algoritmos mais populares usados para classificação estrutural estão o agrupamento por k-médias, agrupamento hierárquico, mapas autoorganizáveis e estimadores de máxima verossimilhança. Tais abordagens estão geralmente entrelaçadas à reconstrução dos modelos 3D. No entanto, trabalhos recentes indicam ser possível inferir informações a respeito da estrutura das moléculas diretamente do conjunto de projeções 2D. Dentre estas descobertas, está a relação entre a variabilidade estrutural e manifolds em um espaço de atributos multidimensional. Esta dissertação investiga se um comitê de algoritmos de não-supervisionados é capaz de separar tais "manifolds conformacionais". Métodos de "consenso" tendem a fornecer classificação mais precisa e podem alcançar performance satisfatória em uma ampla gama de conjuntos de dados, se comparados a algoritmos individuais. Nós investigamos o comportamento de seis algoritmos de agrupamento, tanto individualmente quanto combinados em comitês, para a tarefa de classificação de heterogeneidade conformacional. A abordagem proposta foi testada em conjuntos sintéticos e reais contendo misturas de imagens de projeção da proteína Mm-cpn nos estados "aberto" e "fechado". Demonstra-se que comitês de agrupadores podem fornecer informações úteis na validação de particionamentos estruturais independetemente de algoritmos de reconstrução 3D.

Palavras-chave: Análise de Partículas Isoladas, Microscopia Eletrônica de Transmissão, Crio-ME, Classificação Não-Supervisionada, Agrupamento de Dados, Agrupamento por Consenso.

Index

Dedicado a Sérgio, Adriana, Elisa e Fábio.

Agradecimentos pessoais

Primeiramente, aos meus pais, Adriana e Sérgio, que desde cedo me deram condições e me incentivaram a observar e interpretar o mundo, e aos meus irmãos, Elisa e Fábio, que são os melhores companheiros que eu poderia desejar ter ao meu lado. Também, à minha tia Luciana e aos meus queridos avós Célia e Henrique, Iolita e José (*in memorian*).

Aos meus grandes amigos do COTUCA e adjacências, companheiros de todas as horas, desde 2003: Noia, Haira, Pasti, Tata, Lucão, Anne, Alex, Kbção, Carol, Büllzinho, Ana Claudia e Büllzão.

Ao pessoal da Elétrica que me acompanhou desde o início da graduação, em 2006, especialmente àqueles que se aventuraram comigo no CABS: Pedrão Nardelli, Zolezzi, Chaim, Paty, Tomé, Edward, João Ito, Bulhões, Pet, Pedro Rubira, Tato, Tropeço, Zé, Sarah, Folgado, Feira, Goiano, Carol e muitos outros. Ao Pedrão e ao Érico, que me acompanharam no trajeto Barão Geraldo-Sousas e além, desde os tempos de COTUCA e também por todos os anos de graduação.

Aos grandes amigos (e companheiros de trilhas) que tive oportunidade de encontrar no LBiC e na pós da FEEC: Alan, Rosana, André, Wilfredo, Saullo, Kurka, Marcos, Andrea, Kamila, Salomão (*in memorian*), Carlos, Hamilton, Eliezer e Raul.

Aos amigos do CNPEM, com quem tive o prazer de conviver desde 2009 e que muito contribuíram para minha formação pessoal e científica: Edwar, Jimy, Binho, Davi, Wu, Taís, Maysa, Luciano, Fernanda, João Paulo, Zaira, Gabriel Brunheira e Rodrigo Guerra. A Antonio Ramirez, que propiciou minha primeira oportunidade de trabalhar com ciência, a qual me iniciou no mundo da microscopia eletrônica. Ao Daniel Stroppa, por ter se tornado não apenas um grande colaborador, mas também um grande amigo. Aos meus amigos e companheiros de grupo de pesquisa no LNNano: Murilo, Ricardo, Markito, Alexandre, Betinho, Vinícius, Carlos e Nico.

Também, aos amigos pós-graduandos que conheci nas inesquecíveis viagens que tive a oportunidade de fazer durante o mestrado: B2, Júlio e Diorge; Dominik, Jil e Anindya.

À minha namorada, Ana Paula, a quem não tenho palavras para agradecer por todo o apoio e carinho, que muito me inspiram em todos os momentos.

Por fim, gostaria de agradecer imensamente aos meus orientadores, Fernando e Rodrigo, por terem confiado no meu trabalho, por toda a paciência e compreensão que tiveram em momentos de adversidades, e, acima de tudo, pela inspiração pessoal e profissional propiciadas ao me guiarem através desta empreitada.

Certamente, os nomes mencionados acima são insuficientes e provavelmente minha falha memória cometeu injustiças, pelas quais peço desculpas. A todos vocês, digo que sou extremamente grato por ter encontrado tantas pessoas incríveis ao longo desta vida!

Acknowledgments

We would like to acknowledge Dr. Junjie Zhang (Texas A&M University) and Dr. Wah Chiu (Baylor College of Medicine) for kindly providing us the set of real Mm-cpn images used in this work.

The author of this work was funded by the Brazilian Research Council (CNPq).

"*Nothing is stronger than an idea whose time has come.*"

(Victor Hugo)

List of Figures

List of Tables

List of Algorithms

List of Abbreviations and Acronyms

ANMI	Average Normalized Mutual Information
ARI	Adjusted Rand Index
CA	Correspondence Analysis
Cryo-EM	Cryoelectron Microscopy
CSPA	Cluster-based Similarity Partitioning Algorithm
EM	Expectation-Maximization
GMM	Gaussian Mixture Model
HAC	Hierarichical Ascendant Clustering
HBGF	Hybrid Bipartite Graph Formulation
HDC	Hierarchical Descendant Clustering
HGPA	Hyper Graph Partitioning Algorithm
ICA	Independent Component Analysis
MAP	Maximum *a posteriori*
MCLA	Meta-CLustering Algorithm
ML	Maximum-Likelihood
MSA	Multivariate Statistical Analysis
NMI	Normalized Mutual Information
NMR	Nuclear Magnetic Resonance
PCA	Principal Component Analysis
SNR	Signal-to-Noise Ratio
SOM	Self-Organizing Map
SPA	Single Particle Analysis
SPR	Single Particle Reconstruction
TEM	Transmission Electron Microscope

1. Introduction

Since the invention of the optical microscope, in the 16th century, mankind has been enhancing the ability to observe the world in extremely small scales, down to the order of a few Angstroms (10^{-10} meters) nowadays. The potentials of exploring matter in such magnification are well described in the superb lecture by Richard Feynman called "There's Plenty of Room at the Bottom" (Feynman, 1960), considered by many to be a landmark of nanotechnology. Among the many visionary predictions made by Feynman were the advantages of an electron microscope "100 times better" than those existing at the time, thus enabling the investigation of certain properties of proteins and nucleic acids. The functional activity of these entities inside the cells depends not only on their chemical composition but also on their shapes, and understanding these aspects is the scope of structural biology. Discoveries in this field often lead to applications in medicine, agriculture and renewable energy sources.

However, the impressive development of structural biology seen in the last decades was only possible thanks to the considerable expansion of our ability to store and manipulate data in digital form. Interestingly, the impact that nanotechnology would have on the evolution of computing devices was also predicted by Feynman in the same lecture. Evidently, not only the devices have evolved but also the ways we manipulate and understand data. In this work, we will investigate how images of molecules with distinct shapes recorded by transmission electron microscopy can be classified by algorithms with minimal human supervision. Thus the subject of this dissertation lies on the intersection of these three fields: electron microscopy, structural biology and computer engineering.

Feynman was right about how important the electron microscope would become for structural and molecular biology, but that did not happen in a straightforward manner. A challenge for observation of biological particles in the transmission electron microscope (TEM) is the radiation damage imposed by the beam on the sample. In order to reduce radiation damage, the sample has to be coated with heavy metal salts or embedded in vitreous ice, following a groundbreaking preparation protocol proposed Adrian *et al.* (1984). Particularly, this latter method allows the observation of the molecules in their quasi-native states, and therefore TEM is one of the main

instruments used nowadays for investigation of “molecular machines” (Robinson, Sali & Baumeister, 2007). Nevertheless, the electron dose supported by the sample is still low (Henderson, 1995), resulting that the signal-to-noise ratio (SNR) of the collected images is very poor. Therefore, it only becomes possible to perform image recognition and 3D reconstruction after a reasonable amount of signal processing and statistical treatment efforts (van Heel, 1984).

The method known as “single particle analysis” (SPA) allows retrieving the 3D structure of relatively large molecules, greater than about 300 kDa. The name of the method comes from the fact that the projection images used for 3D reconstruction correspond to individual particles in solution. This is in contrast to X-ray crystallography, perhaps the most popular structural resolution technique, where particles are arranged in a crystalline lattice and the diffraction data recorded correspond to the average of all particles in the crystal.

Single particle analysis has been employed for the study of proteins, viruses and other complex structures, like the ribosome. Basically, the method consists on assigning a set of angles on the Euler sphere to each projection image. Once these hidden variables have been estimated, it is possible to obtain a 3D volume by applying a reconstruction algorithm such as filtered back-projection (Harauz & van Heel, 1986). The angle assignment and reconstruction steps are iterated up to the convergence of the structure. However, the correct orientation for each image is missing and diverse interference sources hamper its estimation, making single particle analysis a genuinely ill-posed inverse problem. Due to the low SNR, many images have to be used for a satisfactory reconstruction, typically tens of thousands.

However, in general it is not true that all the isolated particles used in the reconstruction are stable copies of the macromolecule under analysis. They are flexible structures which may assume distinct conformational states, and adding them up on a single model implies losing resolution. This problem is known since the early years of SPA (Frank & van Heel, 1982) and was held as an obstacle for electron microscopy of molecular assemblies. But the recent advances in data processing algorithms and computing power have allowed the *in silico* purification of the sample, turning SPA into a powerful technique for investigation of molecular dynamics (Klaholz; Myasnikov; van Heel, 2004; van Heel *et al.*, 2012). In the general case, the separation of conformational states is an unsupervised classification problem, as no reference structures are available for the generation of a training image set.

The existing methodologies aiming at the recognition and separation of structural heterogeneity have in common that they all approach this classification task concomitantly to the 3D reconstruction (van Heel *et al.*, 2012; Scheres *et al.*, 2007; Scheres, 2012; Chen *et al.*, 2013; Spahn; Penczek, 2009). This implies a considerable computational effort, as every image must be evaluated somehow against the multiple existing structures in the current iteration. Considering that the SPA data are two-dimensional projection images, and thus are dependent on the whole 3D structure of the object, we identify the lack of methods that can assess structural heterogeneity within a given dataset without the need of performing reconstructions.

Specifically, it would be of great value to have algorithms that can inform, by directly analyzing the dataset, *how many* meaningful structures are present and *what* are the structural labels for each image. A tool that makes such information available could be employed at the beginning of the SPA workflow for partitioning the dataset, at least approximately, for independent processing of the distinct structures. Another application would be at the other end of the workflow, for validating the obtained structures by a method independent of the reconstruction process adopted. The problem of estimating the number of structures *a priori* in a reliable manner is somewhat involved, as explained in Chapter 3. For simplicity, we will focus here on how an unsupervised classification system can be useful for validating heterogeneous reconstructions.

In this scenario, the number of structural classes and a tentative list of labels for each image are already available from one of the conventional processing methods chosen by the user, and can be used in comparison with our method for partitioning validation purposes. Validation of results obtained by SPA and associated techniques is currently a trending topic among the structural biology community, as they gather more and more popularity (Henderson *et al.*, 2012).

We observe that images belonging to different 3D volumes lie on distinct high-dimensional manifolds, as expected from the properties of the projection operator. The projection data from distinct conformations clearly form characteristic “clouds” when visualized in the proper spaces, as shown in Chapter 6. This concept is illustrated in Figure 1.1. We then make use of this information to separate the images belonging to different objects. If the data clouds are sufficiently well separated, simple clustering algorithms should be able to discriminate them (Duda, Hart & Stork, 2000). However, in general the task is more complicated, due to the diversity of biological objects studied and the wide range of experimental conditions found.

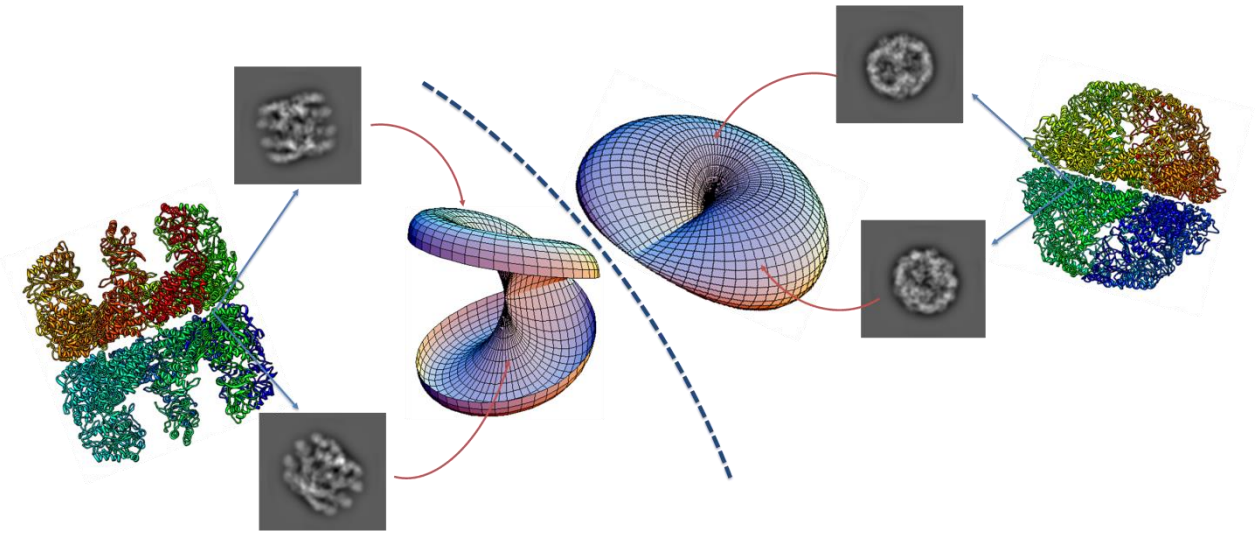


Figure 1.1 - Projection images from different conformations of a macromolecule lie on distinct high-dimensional manifolds. The separability of such manifolds depends on which subset of the multidimensional feature space they are observed. Atomic models represent the Mm-cpn chaperonin in states "open" and "closed" (Zhang *et al.*, 2010).

Difficulties such as noise and the high dimensionality of the dataset (typical image sizes are in the order of 100×100 pixels) also must be taken into account. Even more concerning in this context is that the matching between datasets and optimal classification setups is unknown beforehand. Keeping these issues in mind, we propose that the unsupervised classification task for SPA datasets be performed by an ensemble of clustering algorithms (Strehl & Ghosh, 2002). The diagram shown in Figure 1.2 illustrates how our proposal of consensus clustering is inserted into the single particle analysis workflow.

Ensembles tend to provide more accurate solutions than individual clustering algorithms, and have performance robustness across a wide range of dataset characteristics (Ghosh & Acharya, 2011). Furthermore, they can build a complex classification solution by combining results from relatively simple algorithms, which is interesting from the computational effort point of view. Also, the individual clusterings may be run in parallel (*distributed clustering*). The use of cluster ensembles is also known as *consensus clustering*, because they aim to optimize a consensus measure among the available labeling solutions.

The consensus becomes especially effective when the clustering algorithms that integrate the ensemble are diverse, in the sense that they work differently (*robust clustering*) and/or analyze the dataset from different perspectives (*multiview clustering*). To this end, we chose to work with some widely known clustering algorithms with distinct philosophies. Among them are those based on cluster compactness like k-means (Bishop, 2006) and hierarchical clustering (Duda *et*

al., 2000), as well as those based on cluster connectedness like spectral clustering (von Luxburg, 2007) and METIS (Karypis & Kumar, 1998), which are graph partitioning methods.

We note that the term "classification" is used as a synonym for "clustering" in the SPA literature (e.g. van Heel, 1989), often in the context of averaging similar images to improve SNR. In this text, these terms will also appear interchangeably unless explicitly distinguished. However, the task we are approaching is to classify images in an unsupervised fashion according to the 3D object to which they belong. A brief discussion about the formal distinction between classification and clustering in the machine learning context can be found in Chapter 3.

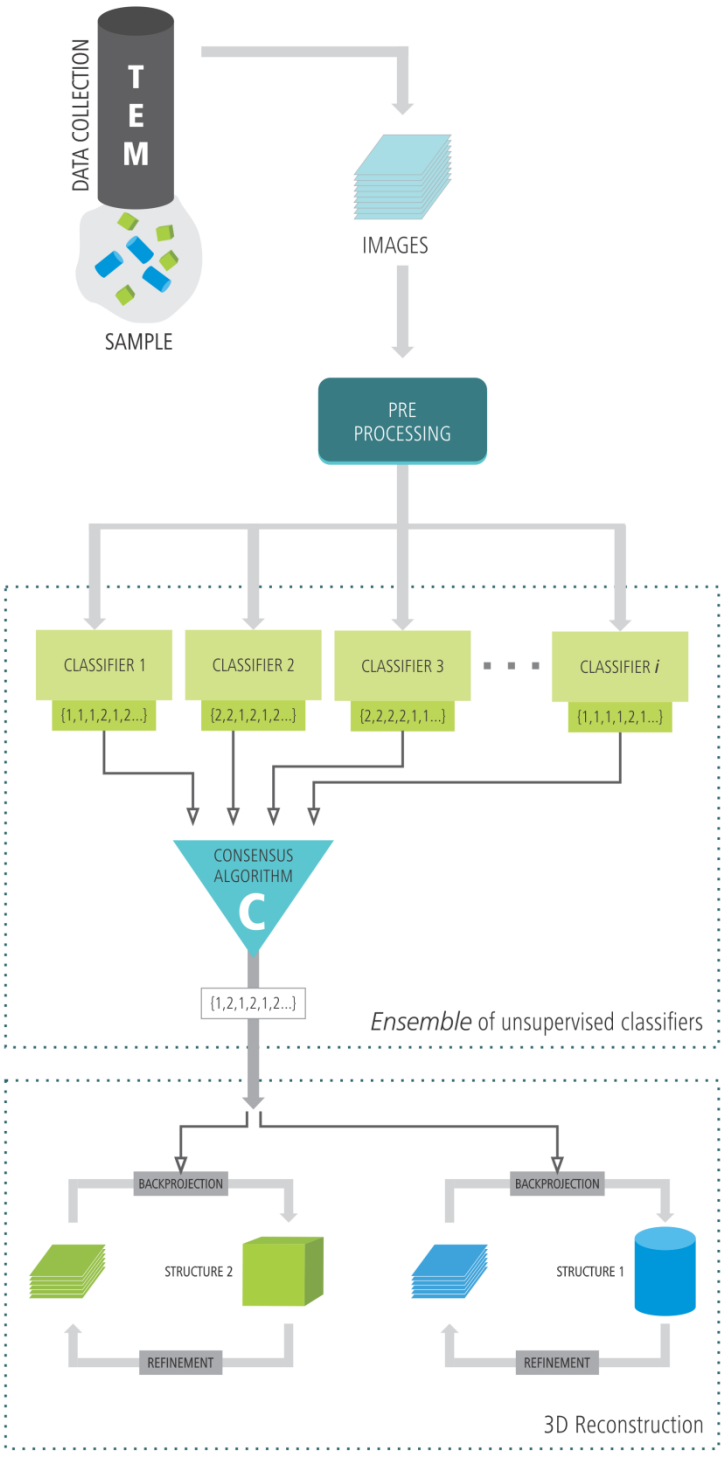


Figure 1.2 - Diagram describing the reconstruction of heterogeneous structures with cluster ensembles. A sample containing macromolecules in random orientations is analyzed in the TEM, generating a set of projection images from isolated particles. After a pre-processing step, the data are labeled by an ensemble of unsupervised classifiers. Finally, the mixture is separated according to their labels for independent reconstruction of the heterogeneous structure set.

The results obtained in this investigation show that the structural heterogeneity may indeed be identified and separated, in the synthetic and real-world datasets analyzed, with minimal need for data pre-processing. Notably, for an experimental dataset containing images from two conformations of the Mm-cpn protein (Zhang *et al.*, 2010), it was possible to achieve more than 80% classification accuracy without any rotational alignment, which is typically an expensive step in the reconstruction process.

1.1 Main goal of the research

We aim to assess whether a mixture of transmission electron microscopy images from heterogeneous structures can be adequately separated by cluster ensembles. We will evaluate the scenarios in which this might be or not be the case, the necessary tools and the context into which this task is applicable.

1.2 Structure of the dissertation

Chapter 2 presents an overview of single particle analysis, including a brief outline of transmission electron microscopy and a comparison with other structural resolution techniques. We go through a review of the literature focusing on classification methods and the reconstruction of heterogeneous samples.

In **Chapter 3**, we introduce the concept of data clustering, its relationship to the classification task in machine learning, and a literature review of the algorithms employed or related to in this study. In this Chapter, we also present the problem of defining the number of clusters, and present some data pre-processing concepts that facilitate clustering in our application, such as dimensionality reduction. Special attention is given to committee and ensemble techniques.

Chapter 4 presents in detail our classification proposal based on cluster ensembles. Concepts presented in Chapters 2 and 3 will be brought together to justify our approach, its underlying assumptions and performance expectations. The experiments performed will also be explained here.

Chapter 5 describes the synthetic and real-world datasets employed in this investigation. We tried to cover different degrees of challenge and to identify how the cluster ensemble proposal reacts to different characteristics commonly found on SPA datasets. The pre-processing stages and experimental setups, as well as algorithm implementations, are also explained in this Section.

Chapter 6 displays and discusses the results obtained, from initial investigations up to the core findings.

Finally, in **Chapter 7**, we draw the conclusions of our experiments, including a discussion of the cases of success and the observed limitations of our proposal. We also point topics that we think are worth dealing with on future investigation.

2. Single Particle Analysis

Single particle analysis (SPA) is a set of techniques used for retrieving the three-dimensional (3D) structures of macromolecules of biological interest. Examples of such macromolecules are proteins, viruses and cell organelles. The physical dimensions of these entities range from a few hundred Ångström (1 Å = 10^{-10} m) to a few micrometers (1 μm = 10^{-6} m), and their mass is in the order of a million Daltons (1 Da = 1.66053892 × 10^{-27} kg). SPA primary data are images of isolated particles collected with the transmission electron microscope (TEM). Although details are outside the scope of this text, brief descriptions of how the TEM works and sample preparation for SPA are outlined in Sections 2 and 2.2, respectively. In this context, Section 2.3 will provide an insight on why TEM images of biological specimens are so noisy. We will then outline the general 3D reconstruction procedure in Section 2.4. Section 2.5 will explain what "structural heterogeneity" means, and why this concept makes SPA reconstructions both more challenging and more interesting. After this introduction, in Section 2.6 the relation of SPA to other structural techniques is briefly discussed. Finally, in Section 2.7 we review the literature on the classification task within SPA, which is the main focus of this work. Sections 2.3 through 2.5 and 2.7.13 are central to the comprehension of this work.

2.1 Transmission Electron Microscopy

The first transmission electron microscope was built in 1931 by Ernst Ruska and Max Knoll in Berlin, Germany. Ruska was awarded the 1986 Nobel Prize in Physics for his invention. The name of the instrument is due to the image being formed from the electron beam transmitted through the sample. Figure 2.1 displays an example of a TEM used for materials and life sciences. The reason for imaging objects with electrons instead of visible light is because the de Broglie wavelength of high-energy electrons is much smaller than that of visible light. The wavelength of electrons accelerated at 100 kV is 0.0037 nm (Frank, 2006), while the smallest wavelength of visible light is of about 400 nm (Serway & Jewett, 2013). Therefore, in theory, electrons allow

the observation of the specimen in much greater detail. In practice, conventional TEMs can achieve a resolution around 2 Å (Frank, 2006), while the best light microscopes cannot exceed 0.5 μm (Glaeser, 2008).

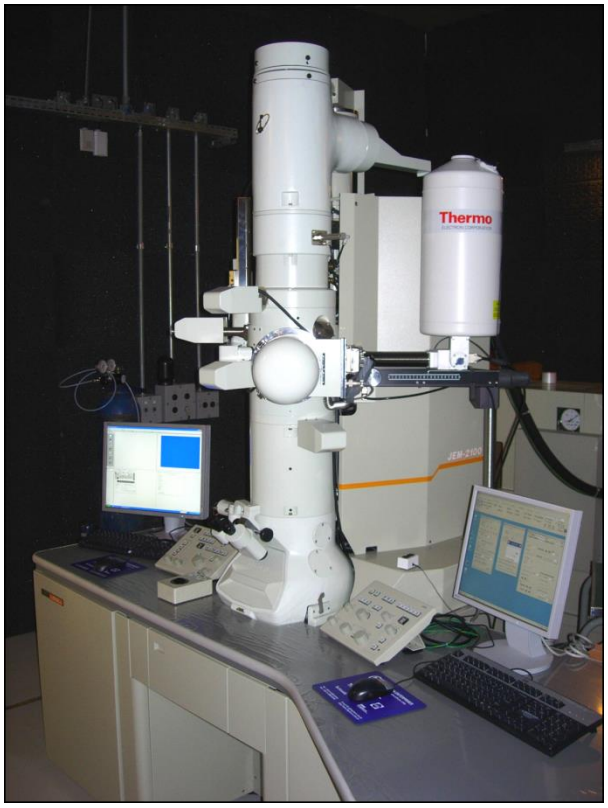


Figure 2.1 - The JEM 2100 transmission electron microscope manufactured by JEOL (Tokyo, Japan) installed at LNNano. Acceleration voltage: 200 kV. The electron source is a LaB_6 crystal. Source: LNNano website[1].

The TEM is composed of many parts, whose simplified diagram can be seen in Figure 2.2. A thermionic or field-emission cathode acts as an electron source inside the vacuum chamber of the microscope. These electrons are focused onto the sample by electromagnetic condenser lenses. The wave function of the electron beam transmitted through the sample forms a diffraction pattern in the back focal plane of the objective lens. The wave function is composed of scattered and un-scattered parts, whose recombination and magnification by further lenses can be visualized in the image plane (Glaeser, 2008). The image can be recorded in photographic film for later digitization, or directly in digital form by means of a charge-coupled device (CCD) camera or by the more modern direct electron detectors (Grigorieff, 2013). Typical sizes for detector

[1] http://lnnano.cnpem.br/wp-content/uploads/2011/08/tem-msc.png

or digitized film frames range from 1,024×1,024 (older) to 4,096×4,096 (modern) pixels. Current detectors can record events with a resolution of about 1 Å per pixel. The pixel size imposes the fundamental limit to the spatial resolution of information acquired in the TEM. Such limit is calculated from the Nyquist sampling theorem, and equals twice the pixel size (van Heel *et al.*, 2000).

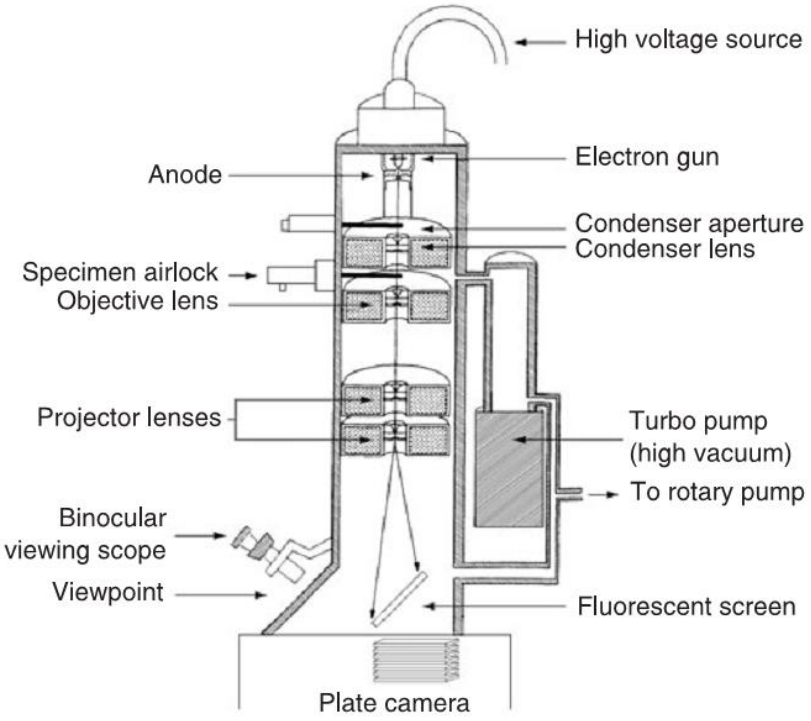


Figure 2.2 - Simplified schematic diagram of a transmission electron microscope. Extracted from Frank (2006).

In general, the specimen is thin enough to not disturb the wave function intensity significantly. What is dramatically affected by the interaction of the electron beam with the sample is the phase of the wave function, and therefore the most interesting information resides on the *phase contrast* images.

2.1.1 Contrast Transfer Function

An important aspect of TEM imaging is the instrument's inherent *contrast transfer function* (CTF). The CTF is responsible for characteristic amplitude modulations and phase reversals in the image spectrum (Mindell & Grigorieff, 2003), as exemplified in Figure 2.3. The Fourier

transform of the CTF is the *point spread function* (PSF), which is defined in the pixel space and can be understood as how the information is spread across the image. The CTF is dependent on the *defocus* (distance from focus) applied when acquiring the image. A specific defocus condition called *Scherzer focus* maximizes the information transfer for high frequencies, which correspond to fine details in the image and therefore is interesting for attempting high-resolution 3D models. On the other hand, low frequencies are dramatically dampened at the Scherzer focus, and they are necessary for spotting particles or any other interesting objects registered on the micrograph (van Heel *et al.*, 2000). For single particle analysis purposes, micrographs must be acquired with varying defocus values, to assure that the whole spectrum is reasonably covered in the set of images. CTF profiles for different defocus conditions are presented in Figure 4, and are also exemplified in Figure 2.5. Afterwards, the CTF parameters must be estimated by computational methods for correcting the images (van Heel *et al.*, 2000; Mindell; Grigorieff, 2003).

For further details on the principles of TEM image formation, please refer to the book by Reimer and Kohl (2008) and the one by Williams and Carter (2009). For specificities regarding biological TEM imaging, the books by Frank (2006) and Jensen (2010) can be consulted.

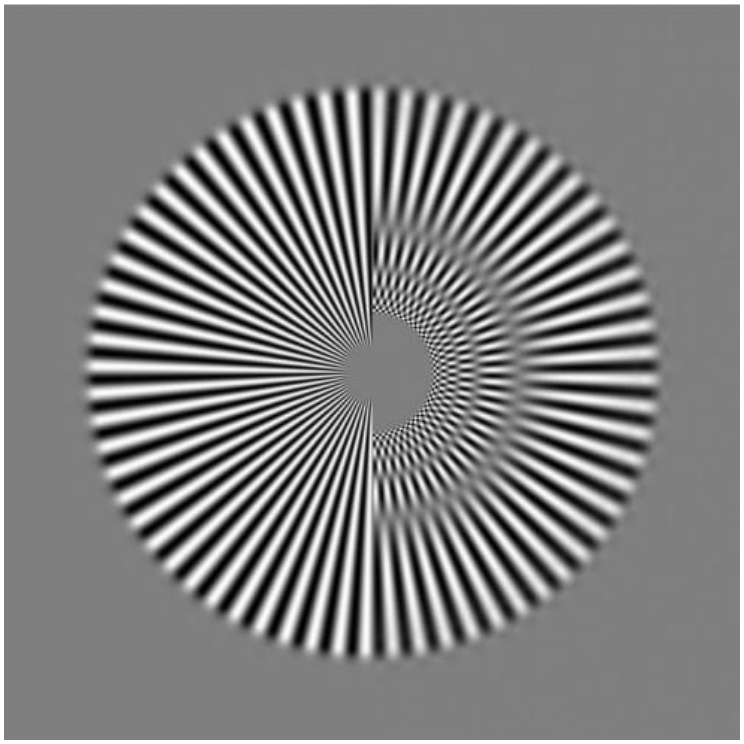
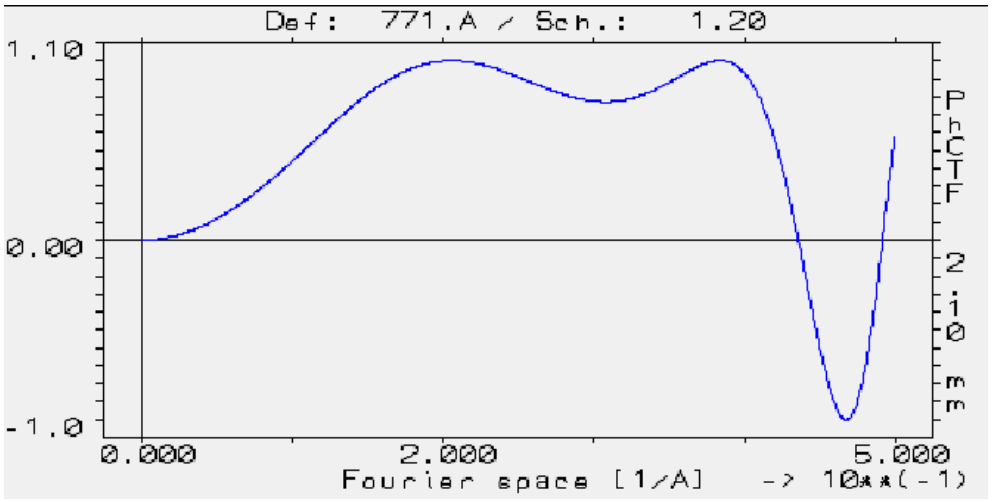


Figure 2.3 - Effects of the CTF on the Siemens star image. Courtesy of Prof. Marin van Heel.

a)



b)

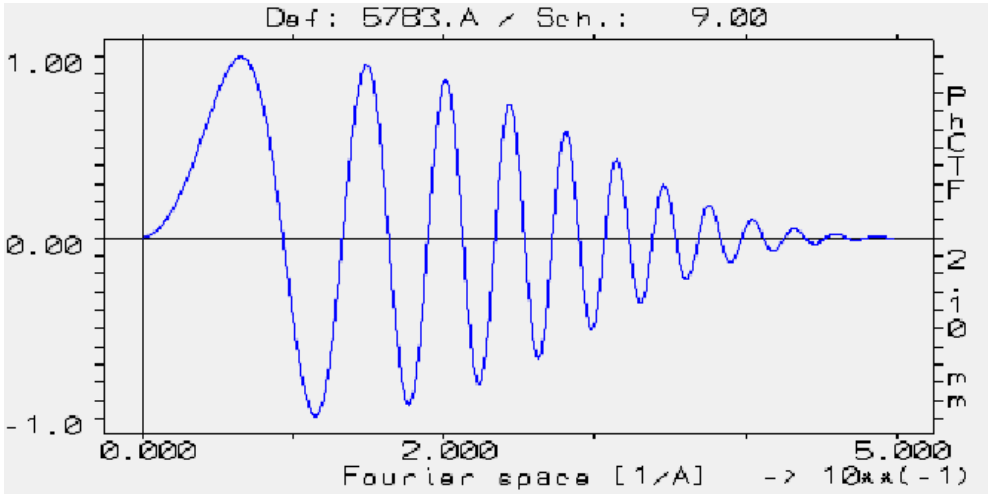


Figure 2.4 - Two examples of CTF profiles in Fourier space. a) At Scherzer focus there is maximal transfer of medium and high frequencies, whereas very low frequencies are dampened. b) At strong defocus, transfer of low frequencies is improved, at the expense of losing high frequencies. Frequency-dependent phase reversals become more present with increasing defocus values. Extracted from van Heel *et al.* (2000).

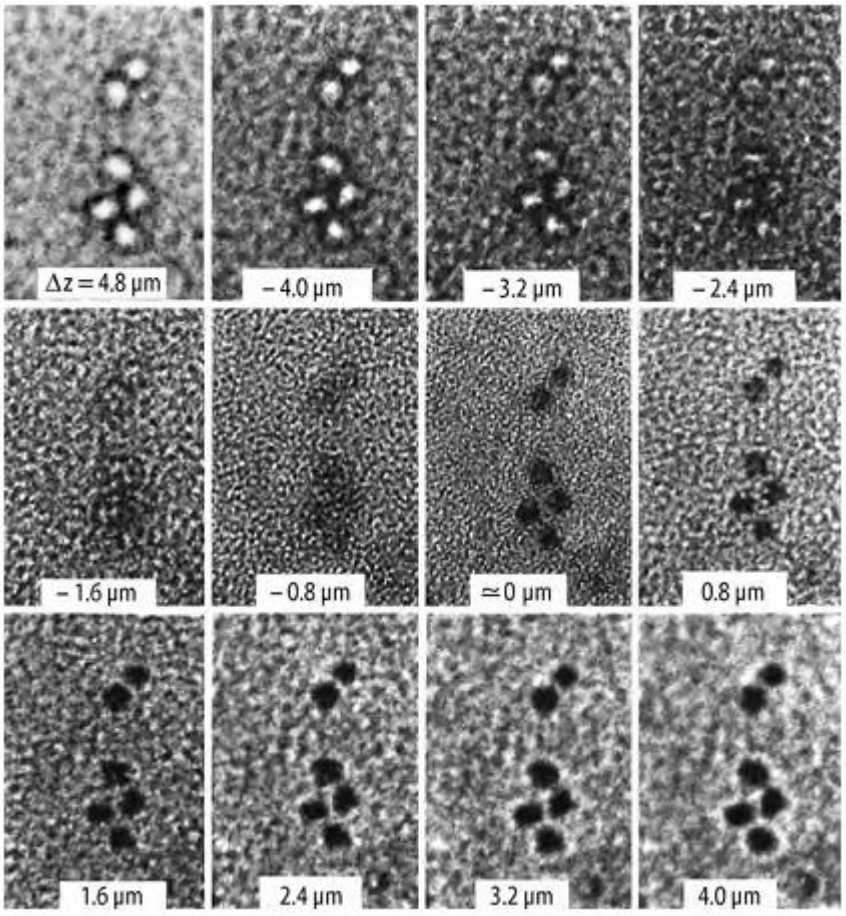


Figure 2.5 – Defocus series (in μm) of ferritin molecules on a 5 nm carbon supporting film, imaged by a 100 kV TEM. Extracted from Reimer & Kohl (2008).

2.2 Sample preservation

2.2.1 Negative stain

A challenge when imaging biological specimens with the TEM is the radiation damage imposed by the electron beam on the sample. In order to enhance specimen protection, special care must be taken when preparing the samples. A common method used for decades (Horne, Brenner, Waterson & Wildy, 1959) is called *negative stain* and consists of coating the specimen with heavy metal salts, such as uranyl acetate or uranyl molybdate. Negative staining provides high contrast images, but it comes with some downsides. One of them is the distortion of particle shape and damaging of particle details (Frank, 2006). Another issue is that the images contain

only low resolution information, due to the large size of the saline crystals surrounding the particles. Despite these aspects, its simplicity makes this method useful in projects where high resolution is not a requirement, and for initial assessment of sample quality and imaging parameters in more complex experiments (Frank, 2006).

2.2.2 Cryo-EM

Initial developments of a transmission electron microscope with a cooling system for the lensens and samples took place in the 1960's (Fernández-Morán, 1966) to observe biological specimens. However, it was the method proposed by Adrian, Dubochet, Lepault & McDowall (1984) that achieved a quantum leap in sample preservation (van Heel *et al.*, 2000). This more powerful protocol consists of embedding the specimen in amorphous ice and keeping it cooled in cryogeny during image collection. The protocol became so popular that it originated the now commonplace term "cryo-EM", short for "cryoelectron microscopy". The vitreous ice layer is thin and light enough to preserve the structural shapes close to their native states, and the cooling severely reduces radiation damage, allowing longer beam exposure times during collection (van Heel *et al.*, 2000). On the downside, cryo-EM images usually have low contrast and low signal-to-noise ratio (SNR); they, however, contain high-resolution information and allow imaging internal particle details. Optimizing the sample preparation parameters such as ice layer thickness and freezing time can be tricky, but is surely rewarding. Figure 2.6 illustrates the different conditions the particles experiment in negative staining and vitreous ice, while examples of images obtained with the two sample preservation methods can be seen in Figure 2.7.

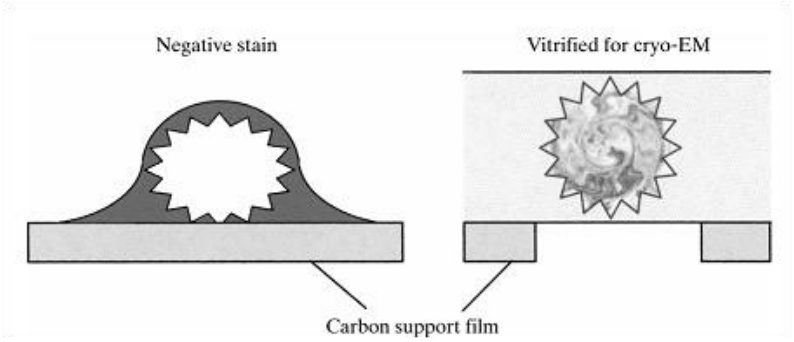
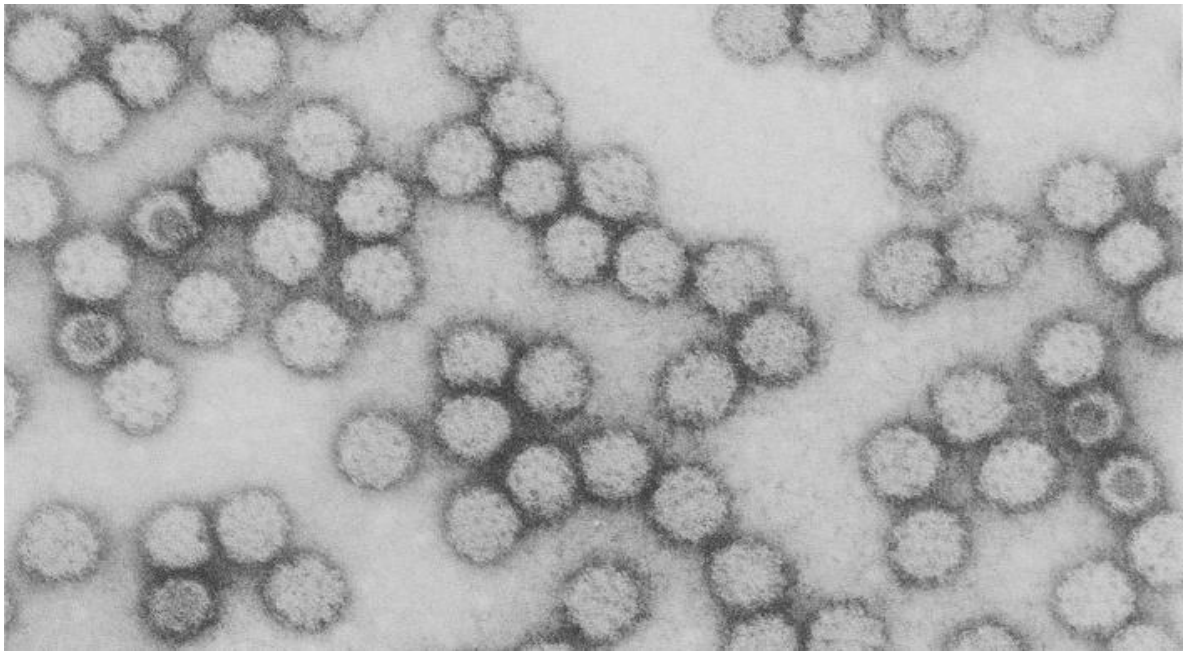


Figure 2.6 – In negative stain (left), the envelope of a virus can be imaged in contrast to the heavy-metal salt crystals involving it. If suspended in vitreous ice, the particle is preserved in its native state and its internal details can be retrieved. Extracted from Saibil (2000).

a)



b)

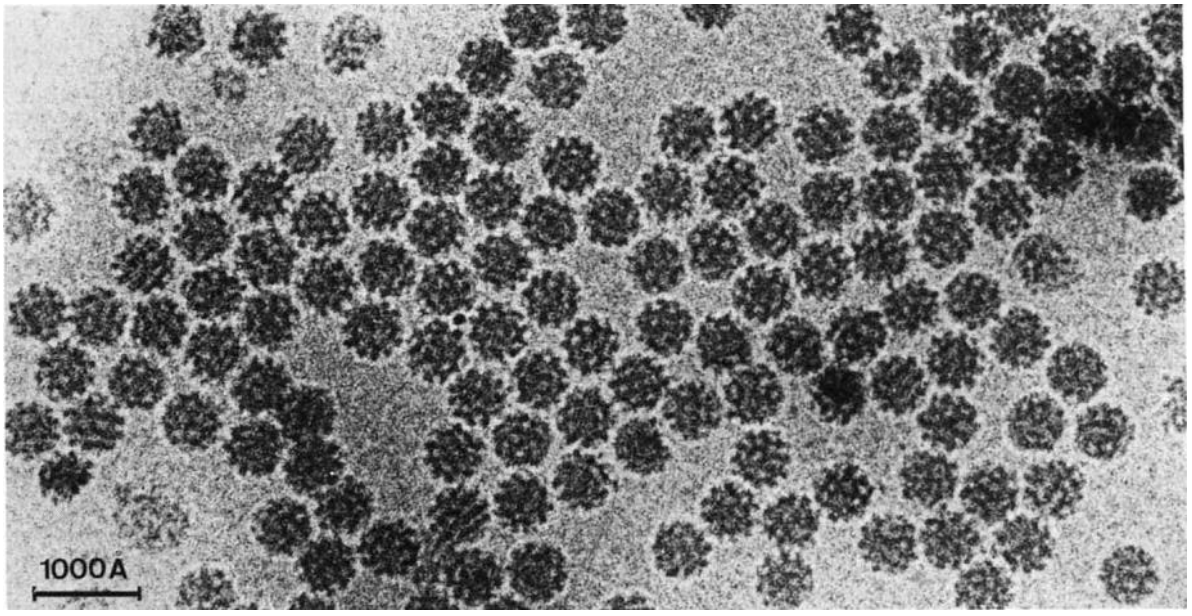


Figure 2.7 – Micrographs of negatively-stained and vitreous ice-embedded specimens. a) Semliki Forest viruses (SFV) in negative stain. Extracted from Söderlund, von Bonsdorff & Ulmanen (1979); b) SFVs embedded in vitreous ice. Extracted from Adrian *et al.* (1984).

2.3 Electron dose and SNR

Even with the sample preservation methods presented in the previous Section, the overall electron dose supported by biological specimens before disintegrating remains considerably low. Typical electron doses that preserve high resolution features of the molecules are in the order of 10 $e^-/Å^2$. In single particle analysis and related techniques, the *signal-to-noise* ratio (SNR) is conveniently defined as in 2.1:

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}} \tag{2.1}$$

where σ^2_{signal} is the variance of the signal and σ^2_{noise} is the variance of the noise.

The SNR is proportional to the electron exposure level used during data collection. As mentioned previously, the exposure tolerance of the samples is very low, often yielding an SNR $\ll 1$ in the acquired micrographs. Therefore, collection of many images is necessary for a significant statistical representation of the signal. For this reason, the sample must contain many randomly-oriented copies of the macromolecule under analysis. In theory, only 12,600 images of such isolated particles would be necessary to achieve a 3 Å resolution three-dimensional reconstruction of a very small protein (Henderson, 1995). Many factors prevent the realization in practice of such resolution with so few images; among them are detector quality and particle motion (Grigorieff & Harrison, 2011). Large and well-ordered structures such as virus envelopes can support higher doses, and their high symmetry implies a greater amount of information per detected particle. The smaller and less symmetric the molecular assembly, the more images will be required, ranging from a few thousands up to a few millions depending on the target resolution (Rosenthal & Henderson, 2003).

Further considerations on the electron dose and its associated effects in single particle EM may be found in the papers by Henderson (1990, 1995), Glaeser & Hall (2011) and the book Chapter by Baker & Rubinstein (2010).

2.4 Reconstruction procedure

2.4.1 Particle picking

After micrographs have been CTF corrected, one has to identify the positions where particles are located and "box" them out to individual images. Typical sizes for these boxes range from 64×64 to 200×200 pixels, depending on the molecule dimensions and pixel size. This procedure is known as particle "picking" and in principle can be done manually, by visually inspecting the micrographs. However, as single particle datasets grow in size towards atomic resolution, manual particle picking may become unfeasible. Remember from Section 2.3 that millions of boxed particles may be necessary depending on the macromolecule size and symmetry, and the target resolution desired for the 3D model. First efforts towards automated particle selection were based on cross-correlation (Saxton & Frank, 1976) and local image variance (van Heel, 1982) which is a reference-free method. Popular recent techniques take manual picking of a few particles as input for automated procedures, which can be based on cross-correlation (Ludtke, Baldwin & Chiu, 1999) or edge detection (Woolford *et al.*, 2007), for example. Figure 2.8 displays the screen of a particle picking program. Automated picking can produce frustrating results depending on the size and variability of particle views, as well as the defocus of the micrograph. At close to focus (Scherzer) conditions, low frequencies are effectively absent. Most picking algorithms rely on low frequencies because they are related to the particle size. Straightforward template matching methods must be used with care due to the noise level, which can lead to false positives and severe reference bias (Henderson, 2013; Sigworth, 1998).

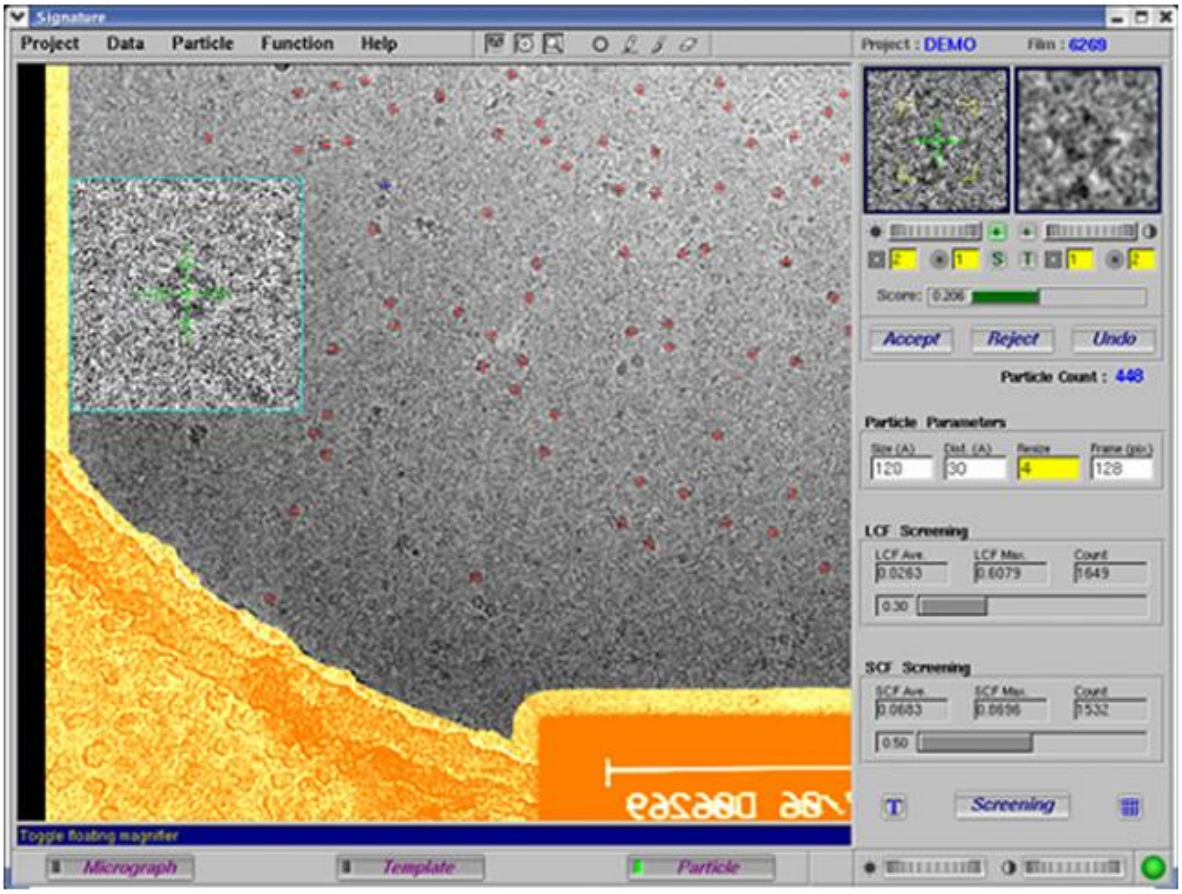


Figure 2.9 – Example of automated particle picking with the SIGNATURE software. Red dots indicate the selected particles. Extracted from Chen & Grigorieff (2007).

Automation of particle picking procedures is an active research topic and overviews of commonly used algorithms may be found in review papers by Nicholson and Glaeser (2001) and Zhu *et al.* (2004). Among recent proposals for reference-free particle selection are the use of neural networks (Ogura & Sato, 2004), support vector machines (Arbeláez *et al.*, 2011) and semi-supervised learning (Langlois, Pallesen & Frank, 2011).

2.4.2 Angle assignment

After selecting and boxing particles, they are commonly masked, filtered and normalized (van Heel *et al.*, 2000). Masking is applied to reduce the influence of the background noise. Soft masks are preferred to avoid introduction of high frequency artifacts in Fourier space. Band-pass filtering is applied basically to remove pixel intensity offsets and to suppress high-frequency

noise (van Heel, Portugal & Schatz, 2009). Each image is then set to a mean value of zero, as pixel intensities of phase-contrast images are modulations in relation to a constant background (Borland & van Heel, 1990). Variance normalization is also necessary to assure that all images are within a comparable intensity range, as they are projections from allegedly identical objects (Unser, Trus & Steven, 1989; van Heel & Stöffler-Meilicke, 1985). Finally, the stack of selected particles is ready for the most important step towards a three-dimensional reconstruction of the object: angle assignment.

To further clarify this task, we must first define a model for the formation of the acquired images. We will follow a notation similar to that in the papers by Scheres *et al.* (2012; 2007). The weak-phase object approximation (Frank, 2006) gives the linear image formation model in 2.2:

$$\mathcal{F}\{i_n\} = CTF_n \boldsymbol{P}_{\phi_n} \mathcal{F}\{v\} + G_n \tag{2.2}$$

where:

- $F\{i_n\}$ is the Fourier transform of the *n*-th image in the dataset, $n = 1, \dots, N$;
- $\mathcal{F}\{PSF_n\}$ is the contrast transfer function for the *n*-th image (remember that the CTF is the Fourier transform of the point spread function);
- $\boldsymbol{P}_{\phi_n}$ is the projection matrix for the *n-th* image, with positional parameters ϕ_n which we want to estimate in this step;
- $\mathcal{F}\{v\}$ is the Fourier transform of the 3D volume (structure) which we ultimately want to determine;
- G_n is independently distributed Gaussian noise in Fourier space for the *n*-th image.

In order to perform a 3D reconstruction, we must estimate the five missing positional parameters ϕ_n, namely two translations and three rotations, described in 2.3:

$$\phi_n = \{x_n, y_n, \alpha_n, \beta_n, \gamma_n\} \tag{2.3}$$

where $\{x_n, y_n\}$ are translational shifts and $\{\alpha_n, \beta_n, \gamma_n\}$ are the three Eulerian angles as defined in Figure 2.10:

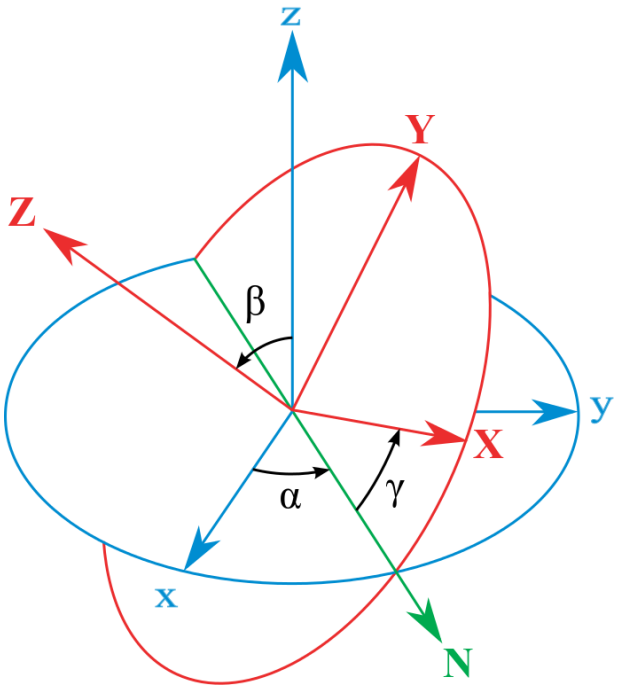


Figure 2.10 – Definition of Eulerian angles α, β and γ. Source: Wikimedia Commons[2].

Single particle reconstruction is thus an ill-posed problem, because solutions that satisfy the missing information are not unique (Scheres, 2012a). The Eulerian angle α is defined to be the in-plane rotations of the projection images. Therefore, if all images are properly centered and rotationally aligned to a common reference, all that is left to estimate are the out-of-plane rigid-body rotations β and γ. See Section 2.4.2.1 for a brief explanation on centering and rotational alignment. A technique closely related to single particle analysis, called *electron tomography*, provides the Eulerian orientations by acquiring several images of each particle at different tilt angles. However, tomography is restricted to very large macromolecular assemblies due to electron dose (see Section 2.3) and defocus variation (see Section 2.1.1) during sample tilting (Robinson *et al.*, 2007; van Heel *et al.*, 2000). Tomographic techniques will not be covered in this text. There are basically two "zero-tilt" methods to figure out the 3D orientation of a projection image: *angular reconstitution* and *projection matching*. In a sense, angular reconstitution can be understood as an "unsupervised" method, because it seeks to find intrinsic angular orientation between images; in contrast, projection matching is a "supervised" method as it assigns angles by comparing the images to references whose orientations are previously known.

[2] http://en.wikipedia.org/wiki/File:Eulerangles.svg

2.4.2.1 Averaging and alignment

Before attempting to assign angles to the images, they need to be reasonably well centered and rotationally aligned. That is, one must estimate parameters $\{x_n, y_n, \alpha_n\}$ to assure all images are aligned to the same coordinate system. However, as images are projections of different views of the macromolecule, it only makes sense to align them to a representative reference of the view they contain. Such references may be obtained by clustering together similar images. In single particle analysis, clustering procedures are synonymous to classification procedures. As with all the steps towards 3D reconstruction devised here, classification of electron microscopy images is a broad research topic in itself, which will be presented in detail in Section 2.7. For the moment, we may only consider that the benefits of clustering images are twofold: 1) it provides the most representative views in the dataset ("top-view", "side-view", etc), and 2) the SNR of such representative views is improved by averaging similar images (van Heel *et al.*, 2009).

Typically, the dataset is initially centered in relation to its rotational average. One may then proceed by classifying the dataset in a pre-defined number of clusters, which is approximately the number of desired representative views. The average image of each cluster, known as a *classum*, serves as alignment references for the individual images. They are aligned in relation to their most similar reference. This procedure is known as *multi-reference alignment* (MRA) (van Heel & Stöffler-Meilicke, 1985; van Heel *et al.*, 2000). By applying successive iterations of alignment and classification, high quality classums may be obtained. Typically the number of clusters is increased along iterations to capture the most diverse possible set of representative orientations of the object. The noise power in the average image decreases proportionally to the number of averaged images (van Heel *et al.*, 2009). Due to the low signal-to-noise ratio of the datasets (see Section 2.3), angular assignments and initial reconstructions are normally performed with classums instead of individual images.

2.4.2.2 Angular reconstitution

Angular reconstitution is based on the "common line projection theorem", which can be stated as follows: "a 1D projection (line projection) of a 2D density is equivalent to a 1D central line through the 2D Fourier transform of the 2D density distribution, and vice versa" (van Heel, 1987). The theorem can be extended to two and three dimensions as well: "two 2D projections of the same 3D object will always have one 1D or line projections in common" (van Heel, 1987). This theorem allows the determination of relative orientations in the Euler sphere unambiguously for a set of at least three images, as illustrated in Figure 2.11:

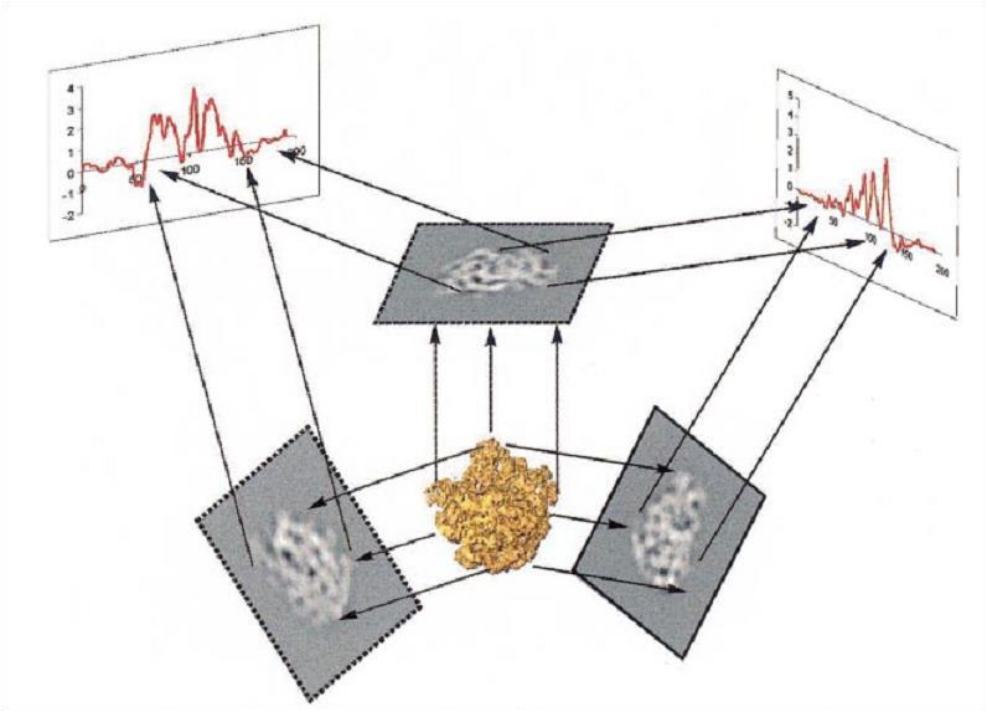


Figure 2.11 – Angular reconstitution is based on the common lines theorem. 2D projections of the same 3D object always share at least one line through the origin of their Fourier transforms. Finding such lines allows the determination of relative orientations for a set of three projections from an asymmetrical structure (in this illustration, the 50S unit of the ribosome). Extracted from van Heel *et al.* (2000).

The search for such common lines can be performed in real space by means of *sinogram correlation*. A sinogram is a set of line projections over all possible rotations of a 2D image. The rotation step determines the precision of the angular reconstitution. For two images, their line projections are correlated against each other to find what is the most similar (ideally identical) pair of lines. The symmetry of the object restricts the search space for common lines. This is the

reason why solving an icosahedral structure (60-fold symmetry) like a large virus is easier than solving an also large asymmetrical structure like a ribosome. Figure 2.12 illustrates the common lines search by sinogram correlation functions for a set of three images.

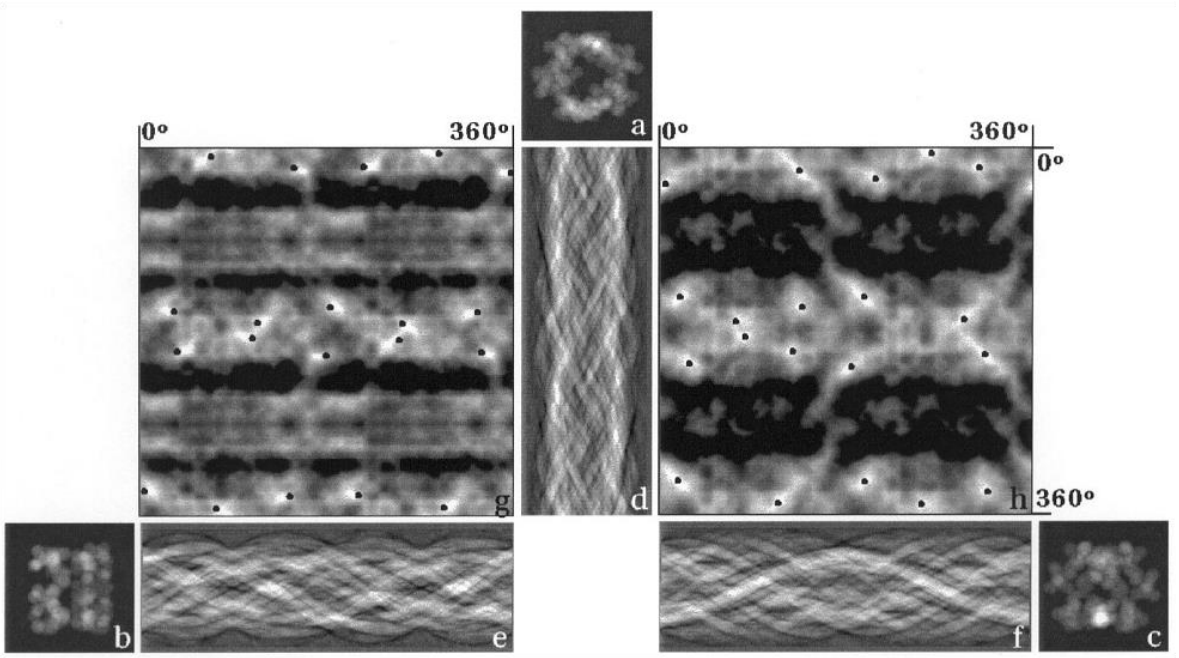


Figure 2.12 – Sinogram correlation functions for three projection images (a, b, c) of a 3D object. The sinograms for each image (d, e, f), when correlated against each other, yield correlation maps (g, h). The global maximum of such correlation maps indicate the common lines for each pair of images and respective orientations. The multiplicity of map peaks (black dots) depends on the symmetry of the object. Extracted from van Heel *et al.* (1997).

Once the relative orientations of a small set of projection images have been reliably established, such set may be deemed the *anchor set*. Anchor sets may also be formed from re-projections of known 3D structures. Angular reconstitution of further images will then be relative to the anchor set. Details regarding angular reconstitution procedures can be found in the papers by Marin van Heel *et al.* (Schatz *et al.*, 1997; van Heel, 1987, 1997).

2.4.2.3 Projection matching

While angular reconstitution can be used for *de novo* angular assignment, projection matching relies on a set of projection images whose orientations relative to the 3D object are known. Such knowledge may come from a previously determined 3D density map, for example in lower resolution, or by other techniques such as X-ray crystallography; or from a previous iter-

ation of the current 3D reconstruction procedure. An initial model may also be constructed by randomly assigning the Euler angles (Harauz & van Heel, 1985). The method consists of taking a known 3D structure, re-projecting it across different orientations, and correlating the set of experimental images to this set of re-projections (Harauz & Ottensmeyer, 1983; Harauz & van Heel, 1985; van Heel *et al.*, 2000). The angles assigned to the images are those of their most similar re-projection. This process may be iterated to improve the 3D reconstruction under course, as illustrated in Figure 2.13. The imposed angular sampling of the Euler sphere determines the number of re-projections, and consequently the precision of the angular assignment. Again, object symmetry plays an important role by determining the number of re-projections needed for the comparisons. Projection matching may be used not only for assigning the β and γ angles, but also for performing translational and rotational (in plane) alignments (Orlova & Saibil, 2011). For involving exhaustive comparisons of N experimental images against a set of M references, projection matching is quite a computationally expensive procedure.

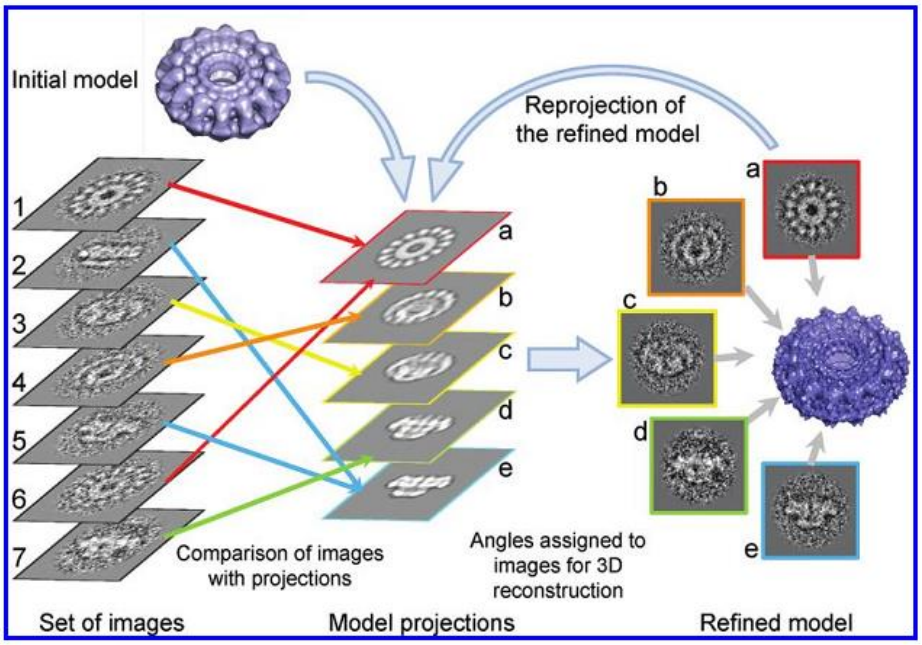


Figure 2.13 – Overview of the projection matching procedure. The stack of experimental images (1-7) is compared against a set of re-projections (a-e) from a previous model or existing structure. Each image is aligned to its most similar reference and receives its Euler angles. A new 3D structure is calculated and its re-projections may be used for the next refinement iteration. Extracted from Orlova & Saibil (2011).

2.4.3 Reconstruction algorithms

Once images have been aligned and assigned Euler angles, finally one may obtain a 3D model out of the single particles imaged in the TEM. This 3D model corresponds to the density map of the molecule. The basic reconstruction procedure consists in back-projecting each 2D projection image through the 3D space according to its assigned orientation, by means of the inverse Radon transform (Radon, 1986). Thus, the three-dimensional density map of the structure is approximated from the collected images. The first reconstruction procedure for electron microscopy images of biological specimens was proposed by De Rosier and Klug (1968), where the "central Section theorem" was introduced to this purpose. Aaron Klug received the 1982 Nobel Prize in Chemistry for the development of electron crystallography, a related technique in which the particles are orderly arranged in a crystalline lattice. The central Section theorem is the extension of the central line theorem to higher dimensions: the Fourier transform of a 2D projection is a slice through the origin of the Fourier transform of the 3D object (see Section 2.4.2.2). Figure 2.14 illustrates this concept. However, simply back-projecting the 2D images through the 3D space does not work properly, because the overlap of frequency components increases towards the origin of the Fourier space, thus overly emphasizing low frequencies. To overcome this effect, Harauz and van Heel (1986) proposed the filtered (or "weighted") back-projection algorithm. Improvements on such methods have taken place since then, as well as alternative proposals have appeared. Overviews of reconstruction techniques can be found in works by van Heel *et al.* (2000) and Frank (2006).

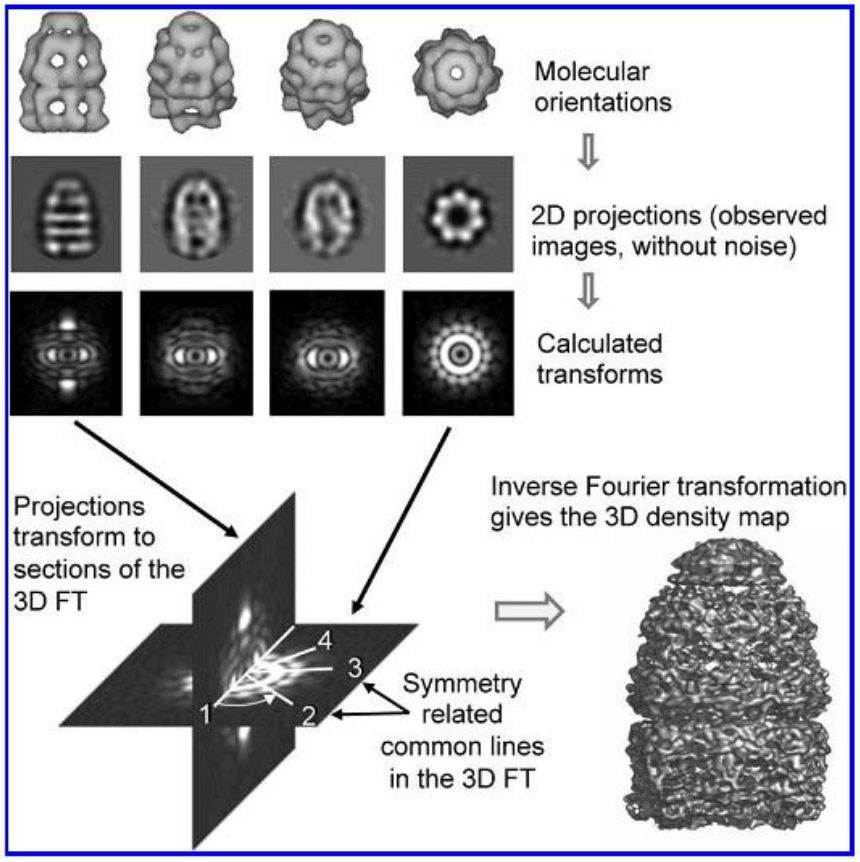


Figure 2.14 – Overview of the 3D reconstruction process in Fourier space. Extracted from Orlova & Saibil (2011).

2.4.4 Overview of the iterative reconstruction workflow

It is important to notice that the averaging and alignment (Section 2.4.2.1), angular assignment (Sections 2.4.2.2 and 2.4.2.3) and reconstruction (Section 2) steps briefly explained above constitute a cyclic workflow that must be iterated until convergence of the 3D density map. A current iteration of the structural model may also be used to generate references for particle picking (Section 2.4.1) in some cases. Variations of such workflow exist and are many, but the general procedure is synthesized in Figure 2.15.

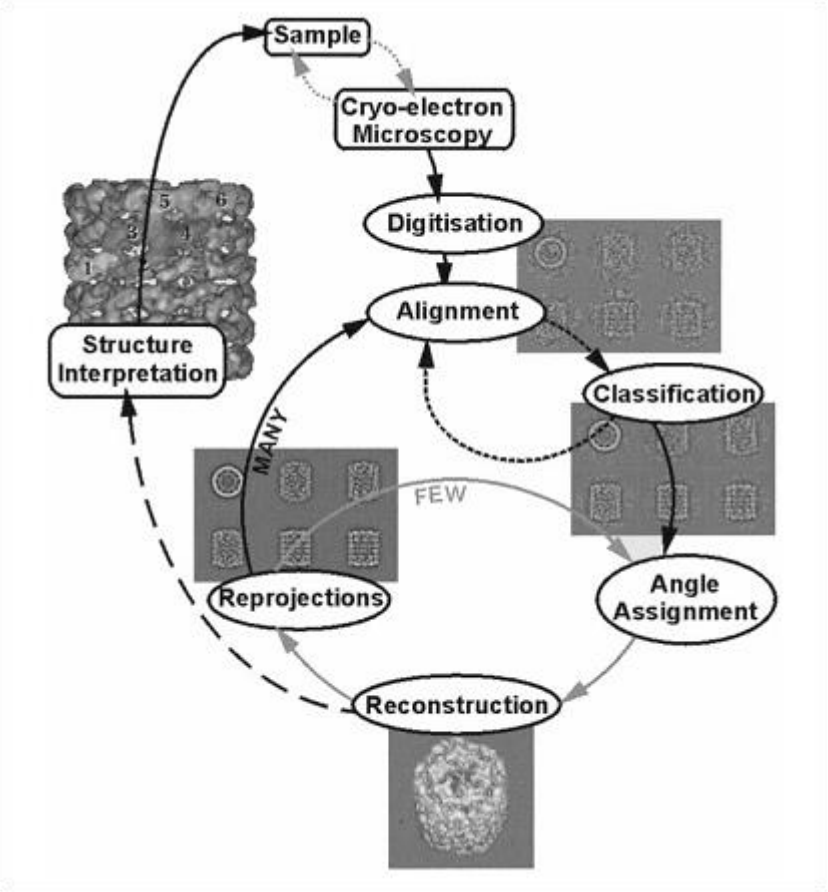


Figure 2.15 – The general iterative single particle reconstruction workflow. After the sample containing multiple copies of the desired macromolecule is biochemically optimized, projection images are acquired in cryogenic conditions (or negative staining) in the TEM to reduce radiation damage. The images are digitized directly from the microscope detector or afterwards if collected on photographic film. After boxing particles, they are iteratively classified according to the view they represent and aligned in relation to representative cluster references. When the particle images are sufficiently well aligned and averaged to improve SNR, angles can be assigned. From these angles, a 3D reconstruction can be performed, whose re-projections may be used many times to improve the alignment, and a few times to provide the angle assignment. Once the 3D density map has converged, the structure can be biologically interpreted or combined with data from other techniques. If something went wrong or there is room for sample improvement, a new data collection is performed and the process restarts. Extracted from van Heel *et al.* (2000).

Some excellent textbooks and review articles have been produced on single particle analysis and its related techniques. For thorough explanations of the whole process, the reader may refer to the books by Frank (2006), Glaeser *et al.* (2007) and Jensen (2010a, 2010b, 2010c). Also, the reviews by van Heel *et al.* (2000), Henderson (2004), Zhou (2008) and Orlova and Saibil (2011) are highly-recommended readings.

2.5 Structural heterogeneity

So far, the single particle reconstruction technique outlined in Section 2.4 assumed the sample is *homogeneous*, which means that all the particles analyzed are stable copies of the same molecular assembly. This does not hold true in practice. Proteins and other macromolecular assemblies are flexible structures. In order to perform their activities in the cell, they may change their shape, assuming different conformational states. For example, see the Mm-cpn protein, on which most of the datasets analyzed in this work are based. Mm-cpn has a barrel-like structure that opens and closes its lids, as illustrated in Figure 2.16, to aid the folding of other proteins (Zhang *et al.*, 2010). Also, molecular complexes may interact with other molecules by binding, as for example the ribosome does with the elongation factor G (EF-G), illustrated in Figure 2.17. The ribosome is a complex machine that "reads" the amino-acid sequences in the messenger ribonucleic acid (mRNA) to synthesize proteins, a process called *translation*. George E. Palade was awarded the 1974 Nobel Prize in Chemistry for discovering the ribosome from observations with the TEM (Palade, 1955), and Venkatraman Ramakrishnan, Tomaz A. Steitz and Ada Yonath also received the 2009 Nobel Prize in Chemistry for solving its atomic structure by X-ray crystallography (Ban *et al.*, 2000; Wimberly *et al.*, 2000; Schluenzen *et al.*, 2000). When "molecular machines" like these are trapped in vitreous ice in their native states for EM observation, different structures in fact co-exist in the sample, making it *heterogeneous*.

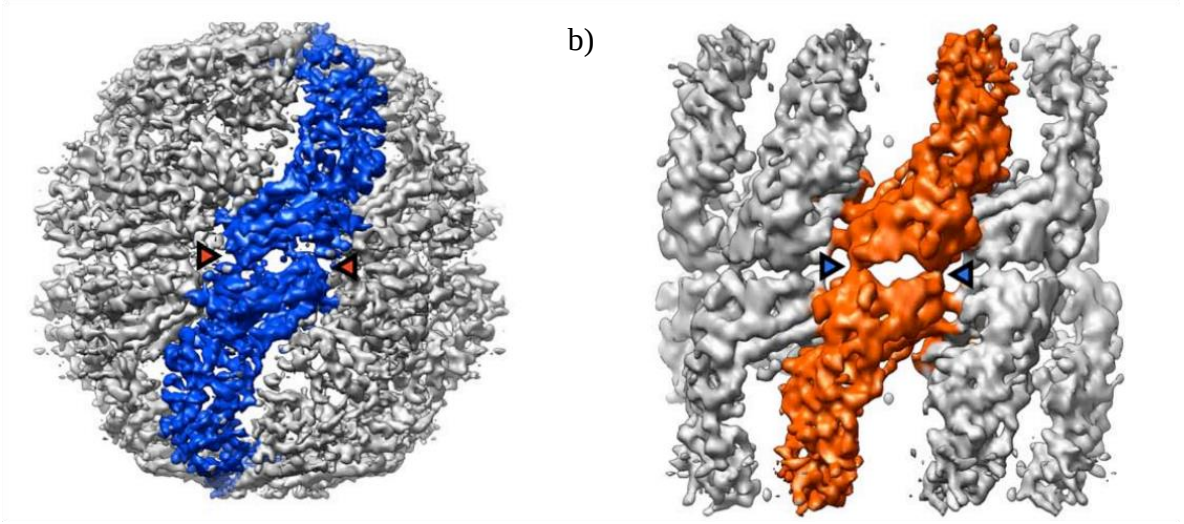


Figure 2.16 – Density maps for the Mm-cpn chaperonin (molecular weight: ~1 MDa) in two conformational states. The two states have been resolved separately by cryo-EM, but are shown together to illustrate the protein flexibility. a) The "closed" state resolved at 4.3 Å; b) the "open" state resolved at 8 Å. The resolution discrepancy is attributed to the flexibility of the lid arms in the "open" state. Two lid subunits across its equator ring are highlighted in blue and orange, respectively. Adapted from Zhang *et al.* (2010), supplementary material.

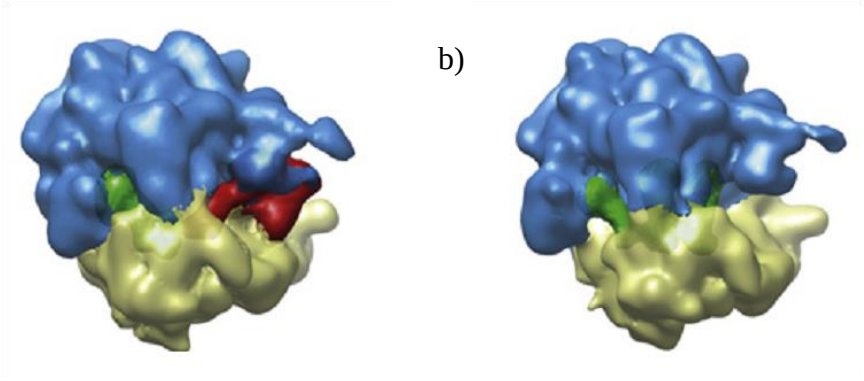


Figure 2.17 – Density maps for the *E. coli* 70S ribosome (molecular weight: ~2.5 MDa) resolved by cryo-EM from the same sample. The ribosome is composed of a large subunit (50S), shown in blue, and a small subunit (30S), shown in yellow; transfer RNA (tRNA) is shown in green. a) Ribosome containing elongation factor G, shown in red, resolved at 21 Å; b) ribosome without EF-G, resolved at 20 Å. Adapted from Scheres (2012a).

Sample heterogeneity in EM is recognized since long time ago (Frank & van Heel, 1982), but it was regarded as a curse then, because the structural flexibility degrades the achievable resolution of the averaged 3D models. However, with the evolution of software development and computing power, sample heterogeneity is now seen as an advantage of the technique, because snapshots of different conformational states may now be extracted from a single experiment (van Heel *et al.*, 2012). A structure resolved in multiple configurations from the same sample was first reported by Mellwig & Böttcher (2001), which was the ATP synthase, a membrane protein of

chloroplasts. The assumption of multiple structures co-existing in the dataset introduces an additional complication in the image formation model from Equation 2.2, which now becomes:

$$\mathcal{F}\{i_n\} = CTF_n \boldsymbol{P}_{\phi_n} \mathcal{F}\{v_k\} + G_n \tag{2.4}$$

The introduction of the structural index $k = 1, \ldots, K$ in Equation 2.4 means now it is also necessary to figure out from which structure each projection image comes, and we change their parameters (Equation 2.3) accordingly:

$$\phi_n = \{x_n, y_n, \alpha_n, \beta_n, \gamma_n, k\} \tag{2.5}$$

In order to estimate the parameter *k* in Equation 2.5, a typical approach is to initially assign each image to a 3D structure randomly, and refine this assignment iteratively by projection matching, as described in Sections 2.4.2.3 and 2.4.4. This is the basic principle of the competitive 3D assignment employed by the IMAGIC package (van Heel *et al.*, 2012). The number of different structures *K* must be known *a priori*. The computational requirements for this kind of procedure are much greater than that of homogeneous single particle reconstructions, because exhaustive comparisons of each image against re-projections of *K* structures are performed. This is the challenge that motivated this project. Are there other ways to assign a projection image to a 3D structure, perhaps more efficiently? If yes, what are the confidence levels and limitations of such methods? Such questions will be addressed in Section 2.7, where classification methods in single-particle electron microscopy are reviewed.

2.6 Comparison of structural biology techniques

Cryoelectron microscopy has come to complement other well-established structural techniques in biology, namely X-ray crystallography and nuclear magnetic resonance spectroscopy (NMR). Basically, cryo-EM is able to resolve larger and more heterogeneous molecular complexes than these alternative techniques (Zhou, 2008). It does not require crystalline samples, which is a limiting factor to which kinds of structures can be studied by X-ray and electron crystallography (Frank, 2006), and also a restriction to observation of molecular dynamics. In fact, a

much smaller amount of biological material is needed for cryo-EM experiments than with other techniques (Frank, 2006). On the other hand, crystallography and NMR are routinely able to achieve atomic resolution, while the detail level of structures resolved by electron microscopy strongly depends on the object symmetry (Chiu, 1993). Also, cryo-EM is restricted to "large" proteins and complexes: small proteins do not yield sufficient SNR for reconstruction from the micrographs (Saibil, 2000). A comparison of the advantages and limitations of each technique is given in Table 2.1.

Table 2.1 – A comparison of qualities and limitations of popular structural resolution techniques. Numerical values are approximations. [1] (Frank, 2006); [2] (Saibil, 2000); [3] (van Heel *et al.*, 2000).

	Requires crystals?[1]	**Amount of required material[1]**	**Molecular weight restrictions[2]**	**Resolution range[3]**
X-ray	Yes	Large (500 pmol)	No	Atomic (2-3 Å)
NMR	No	Very Large (0.2-0.4 μmol)	< 100 kDa	Atomic (2-3 Å)
Cryo-EM	No	Little (0.25 pmol)	> 100 kDa	Near-atomic (5-10 Å)

2.7 Image classification in single particle analysis

We shall now explore in greater detail the task of grouping together TEM images of single particles introduced in Section 2.4.2.1. When classification of images is mentioned in the context of single particle reconstructions, there are two possible interpretations. The first is that of averaging similar views of the particle to improve the SNR, which is the task described in Section 2.4.2.1. The other one concerns classifying the projection images according to the 3D object they come from, which is necessary for the structurally heterogeneous datasets introduced in Section 2.5. These two types of classification are directly related to the two types of clustering which will be explained in Chapter 3: clustering based on "compactness" and clustering based on "connectedness". However, in single particle analysis, structural classification algorithms are a relatively recent topic, and they have an intimate historical relationship to the classification algorithms of similar projection views. A literature review covering the history of such classification algorithms will now be presented. This review is not intended to be exhaustive, in the sense that it is not productive, if not impossible, to report every algorithm ever proposed and their respective applica-

tions. The algorithms and works hereafter reported were selected based on their popularity within the field, their computational originality in this context, and/or their ability to solve interesting biological problems.

2.7.1 Multivariate Statistical Analysis

The first systematic method to average images corresponding to similar views of the object was proposed in a seminal work by van Heel & Frank (1981). They introduced multivariate statistical analysis (MSA) tools in the field, by using Correspondence Analysis (CA) to reduce the dataset dimensionality (Benzécri, 1992). By visually clustering the data projected onto two factors, they were able to separate four different views of horseshow crab hemocyanin half-molecules in negatively stained micrographs. As will be demonstrated by many works later, MSA approaches to dimensionality reduction are very useful for this kind of analysis because they both alleviate the computational efforts of classification and statistically relegate the influence of noise to the least significant components or factors. The principles of CA and the related method of Principal Component Analysis (PCA) will be outlined in Chapter 3. Such approaches are based on the eigenvector-eigenvalue decomposition of the dataset covariance matrix. In following works, the authors and colleagues present the theoretical details and potentials of Correspondence Analysis applied to electron microscopy images (Bretaudiere & Frank, 1986; Frank & van Heel, 1982; van Heel, 1986). Interestingly, in these first works they already acknowledge the structural heterogeneity within the sample as a possible source of statistical variability, in the case where the dataset is homogeneous in respect to the particle orientations.

In face of the growing size of the datasets, van Heel (1984) makes the classification procedure automatic by combining Correspondence Analysis with hierarchical ascendant classification (HAC). The chosen criterion for class-merging is that of “minimum added intra-class variance”, or Ward criterion (Ward, 1963). More details on hierarchical classification algorithms and class-merging criteria can be found in Chapter 3. This is the approach used to identify characteristic views of the 30S ribosomal subunit from *E. coli* and *B. stearothermophilus* on a landmark work on this complex (van Heel & Stöffler-Meilicke, 1985). In a later review, van Heel (1989) compares the use of HAC with the k-means clustering algorithm, which is also covered in Chapter 3. The main critic against the use of k-means is its dependence on the random initial seeding.

To circumvent this problem, the notion of "stable clustering" is introduced here in the form of the "dynamic cloud clustering" algorithm (DCC), which generates "cross-partitions" from a set of different runs of the k-means algorithm (Diday, 1971). However, the HAC algorithm with the proposed post-processing heuristic still provides superior results according to the Ward criterion, as well as in respect to the balance of class members. Frank *et al.* (1988) extend the ideas from this work by combining the dynamic cloud clustering k-means procedure with HAC acting over image factors obtained by Correspondence Analysis. Curiously, the concept of stable clustering is rescued about 30 years later in a now popular algorithm for generating classums (Yang *et al.*, 2012), and is also a motivation for the use of cluster ensembles in data mining (Strehl & Ghosh, 2002). Such proposals will be covered later in this text.

Another important Chapter in the history of MSA-based classification methods is the work of Borland and van Heel (1990) on conjugate representation spaces. They demonstrate the symmetry of classifying the dataset both in the image/data space, and in the pixel/feature space. They present the conversion formulas to commute from one space to another and demonstrate the usefulness of classification in pixel space to assess localized variability in the projection images. Also in this work, the *modulation* metric is presented, which implicitly applies normalization to the data. The modulation metric and the Euclidean distance deal with both negative and positive data. Remember from Section 2.4.2.1 that the projection images have intensity values floating around zero, which is adequate to phase contrast images. Correspondence Analysis, on the other hand, employs the χ^2 metric which is suited to positive data only. Often in this context, the term "MSA" refers to the use of the modulation metric in dimensionality reduction, while "PCA" is used when the Eucliden distance is employed.

MSA techniques are useful for other reasons besides classification, for example analyzing the molecule symmetry by inspecting the *eigenimages* of the dataset. Since they first appeared in the field, efficient implementations of the algorithms above mentioned were contained in the IMAGIC package (van Heel *et al.*, 1996). Due to their interesting properties and computational efficiency in handling large datasets, MSA approaches have also spread to other electron microscopy packages, although sometimes with slightly different terminology according to the exact method implemented: MSA and CA in the SPIDER package (Shaikh *et al.*, 2008), singular value decomposition (SVD) in EMAN/EMAN2 (Ludtke *et al.*, 1999; Tang *et al.*, 2007), PCA in XMIPP (Sorzano *et al.*, 2004), among others. A recent review on the mathematical and computa-

tional aspects of MSA methods and their parallelized implementations can be found in the paper by van Heel, Portugal and Schatz (2009).

2.7.2 Invariant transformations

The dimensionality reduction performed by MSA and related approaches does not alleviate the need for precise rotational alignment of the images for the clustering step when averaging similar views. To this end, some invariant transformations have been proposed to be used instead of the images themselves. Schatz and van Heel (1990) propose the use of double auto-correlation functions (DACF) to achieve translational and rotational invariance. Auto-correlation is the correlation of a signal with itself, and is a shift-invariant operation. If an image is converted from Cartesian to cylindrical coordinates, rotations become shifts in the transformed image. Therefore rotational invariance is then achieved by auto-correlating the ACF converted to cylindrical coordinates. One of the downsides of auto-correlation is that it is equivalent to the power spectrum of the image, in which the amplitudes of Fourier components are squared. This tends to overly emphasize low-frequency components. To overcome this problem, double *self*-correlation functions (DSCF) were proposed (Schatz & van Heel, 1992). Self-correlations are defined as the inverse Fourier transforms of the amplitude spectrum of a signal. DSCFs work the same way as DACFs, but the amplitudes of Fourier components are not squared. The main problem of using the DACF and the DSCF is the information loss in these transforms, as they rely twice on the amplitudes of Fourier transforms alone.

Another approach to achieve rotational invariance in classification was proposed by Penczek *et al.* (1996; 1992). Their method is based on the k-means algorithm. However, when comparing an image to the current cluster centroids during the assignment step of k-means, no straightforward distance comparison is applied; instead, the minimum distance found across all possible rotations of the image is used. This method does not suffer from information loss, but is obviously computationally more expensive. On the other hand, rotational alignment is then performed simultaneously to classification.

2.7.3 Self-organizing maps and growing neural networks

Another paradigm for clustering of single particle images was introduced by Marabini & Carazo (1994). In this work, they employ self-organizing maps (SOM), a kind of neural network used for unsupervised data classification formulated by Kohonen (2001). From the algorithm construction, the cluster prototypes, called *code vectors*, are arranged in a two-dimensional map in such a way that neighbor prototypes tend to be more similar than distant prototypes. The SOM will be presented in greater detail in Chapter 3. They have used SOM both to classify views of randomly oriented chaperonin particles (GroEL) and to find structural differences in a dataset of aligned images of the TCP-1 complex. The classification *per se* was done by visually segmenting the maps. The first case (randomly oriented particles) was meant to demonstrate the application of the method in reference-free alignment. The second case demonstrated the assessment of structural heterogeneity using the SOM. To this end, the authors also employed the supervised counterpart of the SOM, called learning vector quantization (LVM). The goal of LVM is not to produce a summarized analysis of the dataset as in the SOM, but to find optimal class representatives for further classification of unlabeled data. A similar SOM-based approach was used to analyze structural heterogeneity in bi-dimensional crystals (Fernández & Carazo, 1996).

Since then, the research group around José-Maria Carazo has proposed many other SOM-based approaches to classification in single particle analysis. Among them, there is a combination of SOM with an earlier proposal of themselves using fuzzy c-means (FCM) clustering (Carazo *et al.*, 1990; Pascual *et al.*, 1999), called Fuzzy Kohonen Clustering Network (FKCM). The claimed advantage of this method is the reduced susceptibility of FKCM to falling into a local minimum of clustering in comparison to conventional FCM, and at the same time providing the summarized visual analysis of SOM. This algorithm was evaluated both on 338 images and 2,458 rotational power spectra of the GP40 helicase of *Bacillus subtilis* bacteriophage SPP1 in negative stain. When clustering the images they were able to discriminate molecular handedness across homogeneous views of the molecule, and on the rotational power spectra analysis, FKCM clustered the data according to rotational symmetry. FKCM classification accuracy was compared to conventional SOM. Another work by them (Pascual-Montano *et al.*, 2001) introduces a formal objective function to be optimized by the SOM. This method is called *Kernel Probability Density Estimator SOM* (KerDenSOM) and assures that the code vectors formally represent the underly-

ing probability distribution of the data. Deterministic annealing is used during parameter estimation. KerDenSOM was also evaluated on two datasets: the same rotational power spectra data of the FKCM work, and on 2,822 cryo-EM images of the simian virus 40 (SV40) large T-antigen. On the rotational power spectra dataset they were able to recognize previously undetected rotational symmetry groups, and on the large T-antigen data they were able to assess some structural variability within the data. The images have been centered and rotationally aligned previously to classification. KerDenSOM has later been extended to classify electron sub-tomograms (Pascual-Montano, Taylor, Winkler, Pascual-Marqui & Carazo, 2002; Yu & Frangakis, 2011). Implementations of SOM approaches for classification of electron microscopy data are provided by the XMIPP package (Sorzano *et al.*, 2004).

A variant of the self-organizing map, called *growing neural gas* (GNG) (Fritzke, 1995) was employed by Ogura, Iwasaki and Sato (2003) to cluster similar views of a macromolecule. The advantages of GNG over SOM include automatic determination of the number of code vectors and the inclusion of prototypes solely on data-populated regions of the hyperspace. In this proposal, the map representation is optimized by a simulated annealing heuristic. They demonstrate the algorithm usability in averaging 11,000 projection images containing views of a membrane protein, the sodium channel, using 520 apoferritin images as artificially introduced contaminants. The authors provide a visual comparison of 49 classums obtained by GNG, SOM and MSA/HAC to assess class purity, and the GNG is shown to perform better than those on the analyzed dataset.

2.7.4 Multi-step classification: solving heterogeneity

As mentioned previously, for a long time the co-existence of heterogeneous structures in the sample was held as an obstacle to the reconstruction of an accurate 3D model by the single particle method. Structural variations can be hard to assess only from noisy 2D projections, although some of the algorithms above presented had already been able to recognize them to some extent. Nevertheless, the advances on microscopy instruments, computing resources and methodologies for data analysis have rendered this problem gradually more tractable. We shall now overview some of these methodologies.

Burgess *et al.* (1997) have employed a mixed classification approach to observe three different conformations of myosin heads, a motor protein involved in muscular contraction. Firstly, they have used k-means to group similar views of their negative stain data in two steps. Clustering was performed directly on the images, with no data compression technique applied previously, and different masks were used to select for the motor and regulatory domains at each step. Then, a simulated negatively stained structure was generated from an atomic model solved by X-ray crystallography of the myosin sub-fragment (S1), and re-projections of this model were used for visual comparison with the class averages obtained by k-means. Such comparison confirmed at least three distinct conformations of the myosin head. Although 3D models from these configurations were not reconstructed, the observation of the myosin head flexibility in the class averages helped to elucidate its functioning mechanism.

A similar but more automated approach was devised later by Burgess *et al.* (2004) in investigating structural flexibility of myosin and dynein. In this work, an initial classification of the datasets was performed using k-means with variable number of classes. The most meaningful partitioning, for each dataset, was defined by visual inspection assessing the tradeoff between SNR and diversity of views. In this sense, the authors stress that the k-means algorithm is more interesting than HAC because it tends to balance the number of class members, which means that classums will have approximately the same SNR, while in HAC they do not (for a pre-defined number of classes). Such classums were then used to realign the projection data, and the failure to correctly align certain images was taken as an indication of structural flexibility. Again, the datasets were split at this point according to alignment criteria to obtain class averages for each conformation, using the multi-step classification approach with different masks to assess relative movements between head, stem and stalks of these motor proteins.

Mellwig and Böttcher (2001) were able to unveil 3D structures of the adenosine triphosphate (ATP) synthase in two conformations. ATP synthase is a 550 kDa asymmetric enzyme from the membrane of chloroplasts. After applying the MSA method for classification with subsequent alignment and angular reconstitution steps, a single 3D model was obtained from the cryo-EM data. However, visual inspection confirmed that certain classums did not agree well with the model re-projections in the given orientations. Then those classums were separated from the others and two 3D models were calculated using the previously assigned Euler angles. Re-projections of these two models were then used to refine the alignment and class assignment of

the experimental images, and this process was iterated until the convergence of the two structures. According to the knowledge of the author of this dissertation, this has been the first case where multiple three-dimensional models were obtained from the same heterogeneous sample.

In the work by White *et al.* (2004), a strategy for discriminating projections of particles with size variation is presented. This kind of conformational flexibility can usually be detected on the eigenimages of the dataset after applying MSA techniques. Classifying the data using only their coordinates on these selected eigenimages allows the *in silico* purification of the sample according to particle size. Their method has been demonstrated on synthetic and real data, achieving a reconstruction of the heatshock protein Hsp26 at 9.5 Å resolution in two conformations.

Another important work among the first ones dealing with heterogeneous datasets was made by Klaholz, Myasnikov and van Heel (2004). They developed a MSA-based strategy for solving two conformations of the *E. coli* 70S ribosome bound to release factor RF3 while studying protein synthesis. After attempting a single reconstruction from the dataset, it was verified that the angular assignment of some images did not stabilize. They also observed that the region corresponding to the 30S subunit in the density map appeared less ordered than the rest of the structure. Therefore, they decided to classify the images within each orientation class using MSA and HAC focusing on this region, applying a selection mask. From the structural class assignments, they reconstructed two structures using the previously assigned Euler angles. Reprojections of both structures were then generated to improve the alignment and structural classification of the images, in an iterative loop. This work provided the basis for development of the "competitive 3D assignment" in the IMAGIC 4D workflow (van Heel *et al.*, 2012). This later paper contains an interesting review of heterogeneous sample processing by MSA-related approaches.

Following this line, Elad *et al.* (2008) formalize the two-step classification approach for sorting images in heterogeneous datasets. Images are firstly grouped according to the particle view they represent, and subsequently discriminated according to the structure they belong to within each class obtained in the first step. This approach is names "double MSA" by the authors, and represents an important step toward process automation. They detect structural heterogeneity from the multimodal distributions of the MSA coordinates in the most informative eigenimages. The approach is demonstrated on simulated datasets of GroEL/ES chaperones and two experi-

mental datasets, one containing mixed populations of GroEL-GroES-ADP complexes and the other containing the 70S ribosome bound or not to the elongation factor EF-G.

A slightly different strategy for classification of structural heterogeneity was proposed by Sorzano *et al.* (2010), called CL2D. They use an information-theoretic based similarity measure called *correntropy* instead of the usual squared Euclidean distance, and perform classification with a hierarchical but divisive strategy instead of ascendant. Correntropy is claimed to be better suited to measure similarities between non-linear and non-Gaussian process outcomes (Liu, Pokharel & Príncipe, 2007). This is the case of TEM data if higher-order components are accounted for in the image formation model. The class assignment decision function, which the authors call *robust clustering criterion*, is very similar to the Ward criterion. They first assess the clustering for multi-reference alignment, i.e. the clustering of similar views of the macromolecule, in a simulated bacteriorhodopsin dataset in two different noise levels. The quality criterion is the dispersion of Euler angles within each class – it is expected that images clustered together have similar orientations. Detection of heterogeneous structures is tested on a simulated 70S ribosome dataset and on a real p53 dataset. They act as in the classification of views, but assess within each class the percentage of images belonging to each conformation. In other words, it is assumed that heterogeneity classification can be performed concomitantly with the clustering of views. The problem with this strategy is that, in practice, the conformational labels are unknown, so it is not possible to separate conformations in this way for *ab initio* reconstructions. In all the analyzed cases, the results from CL2D were compared against other popular classification approaches: maximum-likelihood classification from the ML2D algorithm (Scheres *et al.*, 2005), which will be introduced soon in this Section; SVD/MSA from EMAN (Ludtke *et al.*, 1999), which is simply MSA/k-means; and also PCA/Diday (dynamic cloud clustering), PCA/HAC and PCA/k-means from SPIDER (Frank *et al.*, 1996). CL2D outperforms all of these, according to the criteria above mentioned, on the given datasets.

2.7.5 Maximum-likelihood Estimation

Another important branch of heterogeneity classification comprises probabilistic modeling approaches. The family of maximum-likelihood estimation (MLE) methods was introduced by Sigworth (1998) for image alignment. The expectation-maximization algorithm, which will

be explained in Chapter 3, allows learning the hidden variables that define the underlying probability distribution of the observed data, including a formal description of the noise. The greatest advantage of using MLE for alignment is that it reduces the reference bias. The reference bias is the problem that arises when averaging very noisy images aligned to a given template: the average image may resemble the template, even if each aligned image contains just random noise. If cross-correlation is used to estimate the shifts, this is especially likely to happen with single particle images. Maximum-likelihood estimators avoid reference bias by weighting every possible shift by their corresponding *a posteriori* probability for each observed image.

This probabilistic model is further extended by Scheres *et al.* (2005) to perform multi-reference refinement of single particle images via maximum-likelihood estimation (ML2D). By "refinement" it is meant the iterative alignment and classification procedure, as in MRA. In this approach, the user specifies the number of references (classums) to be obtained and the number of different structures or conformations mixed in the dataset. Then, the expectation-maximization takes place in determining the alignment parameters and structural assignments, by maximizing the likelihood of observing the data according to the probabilistic model. They compare ML with cross-correlation MRA on cryo-EM (simian virus SV40 large T-antigen) and negative stain (*Bacillus subtilis* bacteriophage SPP1 G40P helicase) experimental datasets and also on a simulated phantom dataset. More remarkable is the observation of two conformations of the SV40 large T-antigen dataset in complex with an asymmetric DNA probe. The density corresponding to this DNA probe had not been observed before.

The problem with maximum-likelihood approaches is that, in principle, a broad search of the parameter space must be conducted, something that can be very time-consuming. The authors have presented a restricted search heuristic that, after the first iteration, covers only a region arbitrarily close to the highest probability peak of the parameter space. This reduced-search approach dropped the computing time from 162 to 25 hours for the large T-antigen dataset comprising 3812 projection images (Scheres, Valle & Carazo, 2005).

Furthermore, the maximum-likelihood method is extended to include 3D refinements (ML3D), enabling near-automatic reconstructions from heterogeneous datasets (Scheres *et al.*, 2007). The ML3D refinement is equivalent to the projection matching approach, but instead of assigning a single value to each of the alignment and class assignment parameters, these are replaced by probability-weighted integrations over the parameter space. The refinement method is

demonstrated on heterogeneous datasets of the 70S *Escherichia coli* ribosome and the SV40 large T-antigen, and compared with conventional projection matching against known structures. The ML approach achieved similar or better results in 3D reconstructions, with the added advantage that no *a priori* knowledge about the structural variability had to be provided. However, performance of likelihood optimization was still an issue: the ribosome dataset took six CPU months to be processed in a supercomputing facility.

Since then, several improvements in the probabilistic model used for likelihood optimization have been introduced. These included: the introduction of colored noise, i.e., the noise behavior was assumed to be independently Gaussian distributed over Fourier components, instead of the previously considered independency across pixels (Scheres *et al.*, 2007); the restriction of structure "brightness" to correct for normalization errors in the experimental images (Scheres, *et al.*, 2009); and the substitution of multivariate Gaussian distributions for multivariate t-distributions, which have heavier tails, thus penalizing more severely outliers in the dataset (Scheres & Carazo, 2009). These modifications have been tested on the heterogeneous dataset of the 70S *E. coli* ribosome, among others, revealing then unknown structural classes and improved classification rates, when compared to the previous formulation of the ML algorithm. The maximum-likelihood classification and refinement framework has been initially implemented in XMIPP (Scheres *et al.*, 2008), although now it can also be found in other packages.

Concerning the computational efficiency of the maximum-likelihood method, a fast adaptive search has been proposed by Tagare, Barthel & Sigworth (2010). The "E" step of the expectation-maximization algorithm requires integration over all possible values for each parameter of the model; instead, they begin by searching over a coarse discrete grid in the integration domain, which is adaptively refined across iterations to focus on the regions of greater contribution to the log-likelihood. They implemented the adaptive algorithm using graphics processing units (GPUs) and tested the proposal on 2D classification of ribosome images, comparing with the previous implementation of ML2D. The proposed Adaptive-EM heuristic achieved similar classification results with a speed gain between 10 times in the first iteration to over 60 times in the final iterations.

2.7.6 Bayesian estimation

A probabilistic framework directly related to maximum-likelihood estimation is the *a posteriori* or Bayesian estimation, introduced by Jaitly *et al.* (2010) to generate an *ab initio* 3D model from 2D class averages. Such class averages are obtained previously from MSA or maximum-likelihood alignment and classification programs. Bayesian estimation, as will be clarified in Chapter 3, weights the observed data with prior knowledge of the problem. Maximum-likelihood estimation, in contrast, takes only the observed data into account. The prior knowledge is incorporated into the probabilistic model in the form of *a priori* distributions for the parameters. In this case, the voxel values of the 3D density model are constrained to have smooth transitions. This type of regularization constraint is interesting to avoid obtaining an overfitted model whose structural features are mostly originated by random noise from the data. The method is demonstrated on one synthetic and four experimental datasets (ATP synthase, GroEL, 70S ribosome and V-type ATPase) whose structures were already known.

The Bayesian approach was then extended by Scheres (2012) to comprise not only 3D reconstruction but also image alignment and classification, including heterogeneity separation. The proposed *maximum a posteriori* (MAP) algorithm assumes smoothness of component amplitudes in the 3D Fourier space, following the previous work on the maximum-likelihood method (Scheres *et al.*, 2007). The MAP method requires little human intervention and robustly avoids map overfitting, as is demonstrated in a comparison between reconstructions of the *archaeal* thermosome solved by MAP and by conventional projection matching. Avoiding overfitting does not necessarily mean a conservative smoothing of the density map, as is shown with a GroEL reconstruction in slightly higher resolution (8.0 Å) than that yielded from conventional projection matching (8.8 Å). MAP was also able to identify an under-represented class of the 50S ribosomal subunit in the previously studied 70S ribosome binded to tRNA and EF-G. This method is implemented in the RELION package (Scheres, 2012b). Despite its advantages, Bayesian estimation suffers from the same computational performance issues that maximum-likelihood methods have. An interesting adaptation of the method was devised by Lyumkis *et al.* (2013), using Bayesian estimation solely for heterogeneity classification purposes. This approach was implemented in the FREALIGN package (Grigorieff, 2007).

2.7.7 Classification of single projection reconstructions

Scheres *et al.* (2005b) have also devised a *single projection reconstruction* (SPR) method to classify heterogeneity in datasets of icosahedral viruses. The approach consists in building a 3D reconstruction out of each projection image, and to compare the variance in 3D space. If there are variance peaks in this volume, the projection images are then classified according to their density in this region. Regions of high variance in the 3D analysis are possible representative of structural heterogeneity, like the presence or absence of attached proteins. This classification may be done by manually thresholding the intensity histogram of a particular set of voxels, or by applying a clustering algorithm like HAC to these intensities. After separation, conventional reconstruction approaches are started and the class assignment may be improved. The authors first test the method on phantom data and compare the performance of different reconstruction algorithms, such as the algebraic reconstruction technique (ART) and weighted back-projection (WBP), among others. They follow with the application of the method to a real dataset of the adenovirus mutant dl313, from which reconstructions in two states are recovered. SPR classification works particularly well for highly symmetric particles because each individual reconstruction is a reasonable approximation to the real structure, and is less expensive than conventional competitive 3D assignments.

2.7.8 Bootstrap methods

Another flavor of statistically grounded methods for heterogeneity assessment are those based on *bootstrapping*, firstly presented in the work by Penczek *et al.* (2006). In a sense, bootstrap methods can be understood as an extension of classification by single projection reconstructions (Scheres *et al.*, 2005) to general symmetries. They are related to the analysis of variance in pixel space as introduced by Borland and van Heel (1990). Bootstrapping consists on re-sampling the dataset with repetitions; from each sample of projection images drawn, a 3D model is generated. This approach can also be understood as the unsupervised counterpart of the bagging (Breiman, 1996) and RANSAC methods (Fischler & Bolles, 1981). The images have been assigned Euler angles previously by any chosen reconstruction process. With this set of bootstrap

models, it is possible to estimate the variance of the density maps. Regions displaying high variance are possibly flexible parts of the macromolecule, thus indicating structural heterogeneity.

Penczek, Frank and Spahn (2006) also apply a method of focused classification following the bootstrap technique presented above. After the estimation of 3D variance by bootstrapping, the regions of highest variances can be selected by using spherical masks. Then the projection images can be clustered by k-means acting only on the selected regions, thus discriminating for structural flexibility. The method is demonstrated by classifying experimental images of the 70S ribosome with and without tRNA ligands and the elongation factor EF-G.

Liao and Frank (2010) further extend the bootstrap classification method by combining it with Principal Component Analysis in 3D space. They perform PCA on the set of bootstrap volumes, and then generate 2D projections of the eigenvolumes. Each experimental image is then assigned a similarity score in relation to the projections of the eigenvolumes in the same orientation of the given image. These similarity coefficients are then clustered by the k-means algorithm. The authors investigated how well this clustering corresponded to the structural labels by testing the method on real and synthetic 70S ribosome datasets, containing mixed populations with and without EF-G binding. Results were compared with ML2D (Scheres *et al.*, 2005). For this analysis, 40,000 bootstrap volumes were generated for each dataset. Classification performance was compared by accuracy and quality of the reconstructions of each conformation, which showed to outperform ML2D.

One of the latest advances in bootstrap techniques was presented by Penczek, Kimmel and Spahn (2011), who created the *codimensional PCA* approach. The method re-samples the projection images uniformly within similar Euler orientations, which are assigned by reconstructing the average 3D map. In this way, artifacts in the bootstrap volumes caused by non-uniform sampling employed in previous methods are avoided. From the set of bootstrap density maps, the eigenvolumes of the dataset are calculated and their re-projections are used to assign factors to the images that are associated with the 3D structural variability. These coefficients are clustered by k-means to perform the heterogeneity separation, in a similar approach when compared to that used by Liao and Frank (2010). To demonstrate the method, the authors apply it in the analysis of the *Thermus thermophilus* 70S ribosome structural dynamics.

2.7.9 Graph partitioning

Recently, there has been a growing interest in graph representation for cryo-EM datasets. Ueno, Kawata and Umeyama (2005) first demonstrated the usefulness of unsupervised graph partitioning by using *spectral clustering* to average similar views of a macromolecule. Spectral clustering is based on the eigenvalue-eigenvector decomposition of an adjacency matrix of the dataset, which in turn is derived from a similarity matrix. "Spectral" in this context refers to the spectrum of eigenvalues of the adjacency matrix. More details on graph partitioning and their objective functions will be presented on Chapter 3. They employed the normalized cross-correlation to measure pairwise similarity between the images in the dataset, and a Gaussian kernel to penalize large distances in the similarity matrix. The approach was demonstrated on synthetic and real datasets of the 70S ribosome, and compared with class averages obtained by classification using Correspondence Analysis. Spectral clustering was shown to obtain classes with higher fidelity to the diverse views of the ribosome present in the datasets.

In a following work, Ueno *et al.* (2007) used negative staining to study pH-dependent conformations of the human serum albumin, a very small protein with a 67 kDa mass. To average similar views of the macromolecule, they employed spectral clustering again. Pairwise distances between images were obtained by an approach close to the one used in the rotation-invariant k-means (Penczek *et al.*, 1996), and a Gaussian kernel was applied to construct the similarity matrix. This classification was also able to discriminate images from the macromolecule in monomer configuration from the dimer configuration, in both pH conditions imposed. Projection matching was used to compare the experimental images to re-projections of an atomic model previously solved by X-ray crystallography.

The work by Herman and Kalinowski (2008) formulates the heterogeneity separation problem as a graph partitioning task. They introduce a similarity measure for projection images that is grounded on the "common lines" theorem, which was explained in Section 2.4.2.2. Basically, the images are compared not directly by their pixel values, but by equally-spaced line integrals. Because the TEM images contain 3D information projected onto a 2D space, the principle underlying this kind of measure is accounting different views of the same structure as more similar than views that look alike but come from different structures. The similarity between pairs of images is used to induce a weighted graph. The classification procedure goes by obtaining a

"Min-cut" of the graph in *k* partitions. A Min-cut is the graph partitioning that minimizes the similarity between partitions, measured by the sum of the weights on the edges traversing partitions. The cost function is optimized by a tabu search heuristic: cut options that do not decrease the Min-cut are banned from being revisited for a number of iterations. The total number of iterations is pre-defined. This proposal is very interesting for dealing with the heterogeneity separation problem independently of 3D reconstructions. However, the method was evaluated solely on synthetic images.

Shatsky *et al.* (2010) present a three-step classification approach that combines projection matching from an initial model to obtain class averages from similar orientations, MSA/HAC to obtain class averages with improved SNR within each angular group, and spectral clustering to further combine these classums in structurally homogeneous groups. The similarity graph for spectral clustering is constructed from a similarity measure between 1D projections, as in the work by Herman and Kalinowski (2008). Angular assignment and structural classification are then refined by iterative projection matching. The authors also propose an interesting method to determine the number of distinct conformations present in the mixture: the number of structural classes is increased until no new structures can be extracted. This multi-model approach was tested on a synthetic and three real datasets, including the 70S ribosome. Although the density maps obtained could indeed be further refined, this work demonstrates perhaps the most automated approach to classification and reconstruction of heterogeneous datasets. In the ribosome case, structural classification accuracy was compared with the known class labels, achieving up to 85% correct classification on average with five rounds of projection matching refinement.

Although not a classification method, the work by Coifman *et al.* (2010) shall be considered here for inferring properties of the density map directly from analysis of the set of 2D projections. The authors derive an interesting relationship between central lines in the Fourier transform of a 3D structure and points on the unit sphere. By mapping the pairwise common lines of the projection images onto a weighted graph, the proposed algorithm is able to infer the angular relationships from the intrinsic dataset structure. The approach is called *Globally Consistent Angular Reconstitution* (GCAR) and relies on spectral analysis of the adjacency matrix, like spectral clustering. The demonstration of GCAR was performed on a synthetic dataset containing noisy projections of the 50S ribosomal subunit.

Another graph-based classification approach is the one presented by Singer *et al.* (2012) to group projection images with similar orientations. They employ the rotation-invariant k-means algorithm (P. A. Penczek *et al.*, 1996) to align and extract relative rotations for every image pair in the dataset. The distance matrix is made sparse by thresholding using a pre-defined number of nearest neighbors or an arbitrary distance value. The authors note that Euclidean distance by itself is not a reliable measure for the noisy cryo-EM images, as projections from very different orientations of the molecule may appear similar due to random patterns introduced by noise. The converse may also occur, when images that are actually from the same orientation may appear to be very distant from each other. However, the optimal in-plane rotation angle found in the alignment step provides useful information when evaluating whether the measured distances are meaningful. These angles are then used to induce a similarity measure, from which a sparse Hermitian matrix is constructed. An Hermitian matrix is a complex square matrix that is equal to its conjugate transpose. The classification technique then follows quite similarly to spectral clustering. Three specific eigenvectors of the Hermitian matrix are calculated, and from them an affinity measure is produced. Clustering the similar views is finally achieved by thresholding the affinity measure so to group together images within a certain neighborhood in the graph. Although the justifications for this method are mathematically involved, the resulting algorithm is quite simple. The main advantage of the proposed method is its robustness to noise, shown by obtaining meaningful class averages on datasets with an SNR of 1/64. However, it has not been demonstrated on real or heterogeneous datasets.

2.7.10 Stability of alignment and classification

Often, intermediate and final results of the dataset manipulation in single particle analysis are difficult to reproduce. This may be due to local optima reached by randomly initialized algorithms, or to explicit and implicit biases introduced in the data processing. Thinking about this, Yang *et al.* (2012) proposed a version of the rotation-invariant k-means algorithm (Penczek *et al.*, 1996) combined with multi-reference alignment that seeks to attain stable and reproducible clusters. The algorithm is called *Iterative Stable Alignment and Clustering* (ISAC) and is implemented in the SPARX package (Hohn *et al.*, 2007). While their approach may resemble the dynamic cloud clustering method (Diday, 1971; van Heel, 1989), ISAC performs a multipartition match-

ing that returns only the consistent clusters found across different runs of k-means; in contrast, DCC returns every cross-partition found, which typically results in more clusters than the number requested. Currently, ISAC is limited to four independent runs of k-means clustering. ISAC also enforces balanced classes in k-means. The main idea is to obtain robust class averages for angular assignment, in the sense that they may be easily reproduced and thus are more likely to be correct; by the same reasoning, the algorithm is also able to exclude unstable images as outliers. Although no explicit connection is made in the ISAC proposal, the reasoning behind the algorithm keeps many similarities with *consensus clustering* concepts (Strehl & Ghosh, 2002).

2.7.11 Other methods

A few other proposed methods to classify electron microscopy images of single particles are worth mentioning. A statistical approach to clustering projections of similar orientations was presented by Samsó *et al.* (2002). This method was based on Bayesian Gibbs sampling, modeling classes as a mixture of non-isotropic Gaussian distributions. The algorithm devised was also able to select the relevant features from Correspondence Analysis or Principal Component Analysis for classification. The authors tested their method against HAC classification using the same features extracted by CA and PCA, on synthetic datasets of the 50S ribosomal subunit. Artificial and real noise, extracted from carbon film portions of micrographs, was added to the images for a range of SNR values. Overall, Gibbs classification was shown to produce classes with higher purity than HAC.

Kawata and Sato (2007) present a statistical method for 2D alignment of particle views. Although the classification part of their approach contains no innovation, the work is worth mentioning here due to its interesting underlying assumptions. The method is called *Multi-Reference Multiple Alignment* (MRMA). The proposed algorithm estimates several candidate shifts and rotations for each image in parallel, in relation to multiple references. Due to the severe noise observed in the TEM images, several correlation peaks may appear when estimating these candidate alignments; however, the distribution density of such peaks is expected to be higher around the "true" peak. Therefore, the reported method finds the optimal alignment by statistical analysis of the correlation peaks. After alignment, images are compressed by Correspondence Analysis and clustered by HAC. The approach was firstly demonstrated on the "Lena" image with differ-

ent levels of Gaussian noise and several rotated and shifted copies. The MRMA method was compared against the conventional MRA approach, which considers only the highest correlation peak for alignment. The alignment quality was assessed by the intra-class variance after HAC classification. Experimental datasets of the Transient Receptor Potential C3 and the sodium channel were also evaluated, and the pixel intensity histograms of the class averages were also used to compare MRA and MRMA. While histograms of classums generated from MRA tended to be unimodal and largely dispersed, those obtained from MRMA tended to present clearly defined peaks, indicating that this method was able to achieve better alignments.

Fu, Gao and Frank (2007) report an approach towards heterogeneity classification denoted *cluster tracking*. In this method, projection images are first grouped in overlapping classes according to their orientation (2D classification). Such orientations are given by projection matching with re-projections from a previous model. The class overlap is determined by the neighborhood of the projection orientations on the Euler sphere. Next, images within each class are classified according to their 3D conformation. PCA and k-means are used, and the relevant components for classification are selected by visual inspection of the coordinates histogram. The eigenimages containing information related to structural heterogeneity are retained, and the others are discarded. If there are neighbor classes already analyzed, the classification results across them are combined. The algorithm proceeds iteratively until all classes have been analyzed according to structural variability. The principle behind analyzing structural heterogeneity in each orientation class at a time is to obtain a more reliable classification by combining the neighborhood information in the process. Afterwards, the assignments can be refined by projection matching. The method is demonstrated on a simulated 70S ribosome dataset, with and without ligands binding. The authors show that even small variations on the objects mass, of about ~4%, may produce recognizable features on the 2D projections that may allow their separation according to the 3D configuration; however, they also point out that more complex conditions found in real datasets may prevent the recognition of such features.

To the extent of the author's knowledge, Schwander *et al.* (2010) have been the first to formally apply *manifold learning* techniques to the conformational classification of heterogeneous datasets. The authors explicitly assume the relationship between pixel intensities and the Euler orientation of the projection as a manifold mapping. Thus, the orientations can be inferred by the position of the data point on the low-dimensional manifold, an approach that conceptually

resembles that used by the GCAR algorithm (Coifman *et al.*, 2010). More importantly, the authors demonstrate how distinct conformational states occupy different manifolds. This is valid both for cryo-EM data as well as X-ray Free Electron Laser (XFEL) data. XFEL is a structural biology technique somewhat similar to single particle analysis that employs X-rays instead of electrons for imaging. By using approaches based on *Generative Topographic Mapping* (GTM) (Bishop, Svensén & Williams, 1998), the authors are able to infer the manifolds underlying the dataset. Due to GTM being a generative technique, it becomes possible to estimate views of the object in any orientation, as well as to obtain arbitrary snapshots of the macromolecule dynamics, as much as the collected data allows. The authors first use GTM for orientation classification of synthetic cryo-EM images of the small protein chignolin; they then test the approach on synthetic XFEL data of the adenylate kinase (ADK) protein in two conformational states. It is observed that two manifolds are learned by the algorithms without any input regarding the number of conformations, and such manifolds correspond to generative models of the projection data for the ADK "open" and "closed" states. Nevertheless, the authors acknowledge that the computing power necessary to run manifold learning algorithms on high-dimensional cryo-EM and XFEL data can be an issue; they also point the advantages of graph-based and Riemannian approaches to heterogeneity sorting, which do not require knowledge about the manifold dimensionality. On the other hand, neither such methods, nor the others above presented, have the generative explanatory power that manifold learning techniques have.

Katsevich, Katsevich and Singer (2013) present a theoretical framework to solve the heterogeneity separation problem by estimating the dataset covariance matrix. This work follows the lines of Herman and Kalinowski (2008), Coifman *et al.* (2010), and Singer *et al.* (2012), among others, that infer 3D structural information by manipulation of the intrinsic relationships across the observed images. The method is based on eigenvector analysis of the covariance matrix, and is related to high-dimensional PCA. They demonstrate the correctness of the approach by recovering the heterogeneous 3D objects from low SNR synthetic data.

At this point, the reader probably has noted that there are many available tools for classification of single particle images, and that there is no systematic way to choose a particular approach. In practice, even for experienced users, classification requires frequent visual feedback and experimenting with the algorithms' parameters. Thinking about this, Yoshioka *et al.* (2013) developed MASKITON, a web-based tool for interactive masking and classification in 2D. With

this graphical tool, the user may easily create selection masks and try different classification algorithms, obtain the correspondent class averages, and compare the results. This kind of analysis may be particularly useful to discriminate heterogeneous subsets of images. MASKITON is related to APPION (Lander *et al.*, 2009), an also web-based pipeline for single particle analysis that acts as a front-end to other electron microscopy packages. The rationale of such tools is to make the user able to transparently benefit from the most interesting features each software can offer, without having to worry about file formats and conventions.

2.7.12 Supervised classification

It shall be remarked that, whenever models of different conformations are available, supervised classification by projection matching can be applied to sort out the structural heterogeneity of the dataset. These models may be available as structures previously solved by other techniques, such as X-ray crystallography or NMR, or from previous iterations of the current processing. In this case, one of the various methods mentioned above can provide these initial models. In general, the reference models must be low-pass filtered before classification to avoid reference bias and overfitting. As the reconstructions improve, higher frequencies may be incorporated to refine the new models. Supervised classification with the SPIDER package is outlined in the review by Shaikh *et al.* (2008), but similar procedures can also be conducted with other software.

2.7.13 Concluding remarks

This literature review points that the classification task appears in two related yet distinct stages of the reconstruction process. The first one is to group projection images containing views of the macromolecule in the same orientation (2D classification). The goal of this classification is to generate class representatives with higher SNR for further angle assignment, as explained in Section 2.4.2.1. The other role of classification is to discriminate the projection images according to the three-dimensional state of the object they were generated from (3D classification). The distinct 3D configurations arise in the form of structural flexibility and/or interactions with other molecules. In the general case, 3D classification is an unsupervised task, as no previously solved

structures are available or the conformational modes are unknown. There are three main branches of classification approaches for both 2D and 3D purposes. The first ones to appear, at the beginning of the 1980's and which remain in use nowadays, were those based on multivariate statistical analysis, initially with Correspondence Analysis and later with Modulation Analysis and Principal Component Analysis, most commonly combined with k-means or hierarchical ascendant clustering. The beginning of the 1990's saw the appearance of the methods based on Kohonen self-organizing maps and their variants. Finally, the 2000's marked the popularization of the approaches based on probabilistic modeling, like maximum-likelihood and maximum *a posteriori* estimation. High quality reviews of heterogeneity classification are available, for example, by Leschziner and Nogales (2007), Spahn and Penczek (2009), Scheres (2010), van Heel *et al.* (2012), and in the book by Frank (2006). Recently, methods based on other mathematical concepts have emerged: for example, those based on bootstrapping and graph partitioning. It was observed that new methodological proposals for heterogeneity classification are either assessed by the quality of the 3D models obtained, or by comparison with previous 3D models and respective structural labels, if available. However, sometimes the proposed algorithms are demonstrated in the context of solving new or relatively unknown structures, making the comparison with established methods difficult. On the other hand, some works use only synthetic data, leaving room for the question of whether the proposed method is able to tackle "real world" challenges. Actually, the validation of single particle microscopy reconstructions is currently an issue within the structural biology community. Whether the results attained by this technique agree with those obtained by X-ray crystallography and NMR, or even if reconstructions performed using different protocols achieve similar solutions (Henderson *et al.*, 2012) are questions still subject to debate.

In general, all consolidated approaches to sorting the structural heterogeneity of the datasets depend on iterative reconstructions of 3D models and comparisons with their reprojections. Computational effort becomes then a concern, as instruments and detectors are being improved and datasets are growing routinely larger towards the goal of achieving atomic resolution models. This is especially critical to maximum-likelihood and Bayesian estimation methods, which have to evaluate a large number of parameter configurations for every image. However, a few works point out that it should be possible to detect the structural assignments by analyzing directly the set of 2D projection images (Coifman *et al.*, 2010; Herman & Kalinowski, 2008; Katsevich *et al.*, 2013). The existence of multidimensional "conformational manifolds" in a hy-

perspace has also been suggested while using MSA techniques (Elad *et al.*, 2008; van Heel, 1984; van Heel *et al.*, 2012) and demonstrated in one work with manifold learning (Schwander *et al.*, 2010), but this fact has not been much explored yet. It may be early to evaluate the relevance of such proposals, although they certainly represent important theoretical advances. Curiously, the history of classification methods in single particle analysis shows that methodological deepness is not synonym for popularity or usefulness: many of the methods presented may be biased towards specific datasets, or require expert knowledge to operate, and therefore remain little known to the general structural biology community.

We therefore identify a scarcity of methods seeking to assess the structural heterogeneity problem directly from the 2D projection images. The recognition of conformational "clouds" by multivariate statistical analysis as suggested previously should be useful to provide initial estimations of class assignments, or to validate the structural classification performed by conventional methods. MSA data compression is a common step in the reconstruction workflow implemented in many packages, but the information provided by it may have not been explored to its full extent (see for example Borland & van Heel, 1990). Also, defining the number of structural classes co-existing in the dataset often requires *a priori* knowledge of the molecular complex under analysis. There is currently no well-established criterion to define this number, and usually different values are tried and the "best" is determined according to subjective structure interpretation criteria. Only a few works have been identified explicitly attempting an unsupervised approach to determining the number of classes (Schwander *et al.*, 2010; Shatsky *et al.*, 2010). It is therefore desirable to design a classification tool that provides useful heterogeneity information to the user independently of the reconstruction method employed, considering the limitations of computing resources, and avoiding the use of *a priori* information about the dataset.

3. Data Clustering

Clustering is the task of grouping together data points according to some pre-defined similarity criteria, based on the intrinsic dataset structure. As we have seen in Chapter 2, the clustering task serves two purposes in the analysis of single particle images acquired by transmission electron microscopy (TEM). The first one, generally referred to as "2D classification", is to average projection images containing similar views of the macromolecule. This averaging improves the signal-to-noise ratio (SNR) for angular assignment. The other one is known as "3D classification" and regards discriminating the projection images according to the conformational state of the macromolecule. In this Chapter, we will introduce statistical and machine learning concepts that are useful for this latter problem. Many of these concepts have been introduced in Chapters 1 and 2 and will be explained in greater detail now. Section 3.1 will present the notation to be used in the subsequent mathematical formulations. Section 3.2 presents alignment-invariant feature selection and extraction tools that are considered relevant for 2D and 3D classification of electron microscopy images. Next, in Section 3.3 we introduce multivariate statistical analysis (MSA) tools that reduce the dimensionality of the dataset and aid the discrimination of information from noise. Finally, in Section 3.4 the concepts of clustering are explicitly introduced. We begin by discussing the formal differences between clustering and classification from the machine learning point of view, in Section 3.4.1. Then, in Section 3.4.2, the different types of clustering are introduced. Next, in Section 3.4.3, we present widely known clustering algorithms, with special focus on those that have been commonly employed in the field of single particle analysis (SPA) and those who have been applied in this work. Section 3.4.4 brings the problem of defining the number of relevant clusters, with emphasis on SPA datasets. Finally, we introduce ensemble techniques for data clustering in Section 3.4.5, which are at the core of the approach proposed in this dissertation. The key sections for understanding the methods employed in this work are 3.3.1, 3.4.2, 3.4.3 and 3.4.5.

3.1 Basic definitions

The basic type of data we will consider in this work are projection images of isolated particles in solution, collected by means of a transmission electron microscope. Regarding Chapter 2, these are the boxed images extracted from the micrographs (Section 2.4.1). Each *data point* is an image, which in turn is represented by a *row vector* $\boldsymbol{x}_n, n = 1, \dots, N$, where N is the total number of data points. Each vector is comprised by P *features* $x_{np}, p = 1, \dots, P$, which, in the simplest case, are the density values of the P pixels considered. P can be the set of all pixels in the image, or only those within an area selected by a binary or real-valued mask. The set $\{\boldsymbol{x}_n\}$ of all feature vectors in our dataset may be arranged in an $N \times P$ matrix $\boldsymbol{X}$. Each row contains the feature vector representation of an image, and each column contains the intensity values for a specific pixel over all images. Later, the original data matrix $\boldsymbol{X}$ may give place to some other convenient matrix generated by means of feature extraction (Section 3.2) and/or dimensionality reduction (Section 3.3) techniques.

When clustering, the dataset shall be partitioned in K groups according to a given similarity criterion, with $K \leq N$. In 2D classification, K is the number of expected relevant views of the particle to be found within the dataset, whereas in 3D classification K is the number of expected conformations. Section 3.4.2.1 will discuss the different ways a data point can be assigned to a partition. The Euclidean distance (Equation 3.15) will be used as the default dissimilarity measure, unless otherwise noted.

3.2 Alignment-invariant features

One of the main challenges in the classification of electron microscopy images of single particles comes from differences in translational and rotational alignments. Projection images containing the same view of the object should ideally always be recognized as identical. However, if there are shifts or in-plane rotations between these images, it becomes harder for pattern recognition algorithms to acknowledge that they contain essentially the same information. To circumvent this problem, a new set of features that are invariant to alignment parameters can be

extracted from the images. A common example of invariant feature used in image processing is the histogram count of pixel values. More specifically interesting for electron microscopy images are the double autocorrelation function (DACF), the double self-correlation function (DSCF) and Zernike moments.

3.2.1 Double Auto-Correlation Function

The cross-correlation is a similarity measure between two signals that is defined as the *sliding* inner product between them. In the case of images, the sliding is discrete and is performed across the pixel positions. The cross-correlation preserves an analogy with the *convolution* operation in that it satisfies in Fourier space the relation shown in Equation 3.1, where the symbol $\star$ denotes the cross-correlation operation and $\mathcal{F}^*$ is the complex conjugate of the Fourier transform:

$$\mathcal{F}\{\boldsymbol{x}_a \star \boldsymbol{x}_b\} = \mathcal{F}^*\{\boldsymbol{x}_a\}F\{\boldsymbol{x}_b\} \tag{3.1}$$

The auto-correlation function (ACF) is then the cross-correlation of a signal with itself. It can be noted that the ACF is simply the inverse Fourier transform of the power spectrum (PS) of a signal, and thus it has a squaring effect on the amplitudes of the Fourier components. The ACF is translation-invariant because the amplitudes of Fourier components do not change if the pixel values are simply shifted within the image. To achieve rotational invariance, the ACF is then converted to polar coordinates, which are defined in Equations 3.2 and 3.3. The coordinates (x, y) give the position of the pixel in relation to the center of the image, r is the corresponding radial distance from the origin and θ its respective distance.

$$r = \sqrt{x^2 + y^2} \tag{3.2}$$

$$\theta = \arctan\left(\frac{y}{x}\right) \tag{3.3}$$

In the output image, the angular sampling can be controlled to achieve finer or coarser representations. It is important to notice that pixels close to the origin will be over-represented in the transformed image when compared to those closer to the borders. Weighting functions can be applied to compensate for this effect. Also, pixels distant from the origin by a larger amount than

the image width, i.e., those near the corners, will not be converted to the transformed image, causing information loss. In polar coordinates, rotations in the original image become translational shifts. Thus, rotational invariance is achieved by applying the ACF again to the first ACF converted to polar coordinates. Schatz & van Heel (1990) proposed this double ACF (DACF) approach for invariant classification of molecular views. Figure 3.1 illustrates the conversion process from a pair of misaligned images through their respective DACFs.

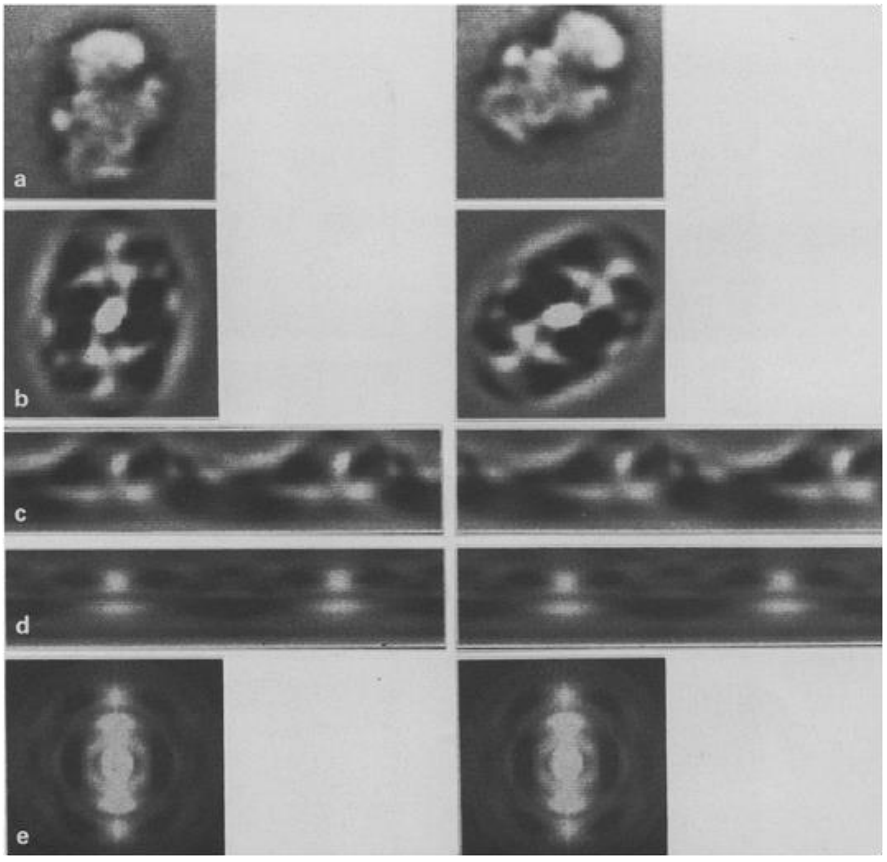


Figure 3.1 – Double auto-correlation functions of a test image taken from a 30S ribosomal subunit dataset. a) The original image and a shifted and rotation version of it; b) the respective ACFs of the image pair; c) the ACFs converted to polar coordinates; d) the ACFs of c); d) the ACFs from d) converted back to Cartesian coordinates, thus illustrating the translation and rotational invariance of the DACF. Extracted from Schatz & van Heel (1990).

3.2.2 Double Self-correlation Function

The previous Section mentioned that the DACF squares the amplitudes of Fourier components. This is a problem for the images commonly found in single particle analysis, because it tends to over-emphasize the low frequencies, whose amplitudes are much larger than those of the medium and high frequencies. To avoid this effect, Schatz & van Heel (1992) proposed the *self-correlation function* (SCF), which is defined in Equation 3.4. $\mathcal{F}^{-1}$ denotes the inverse Fourier transform and abs denotes the absolute value or the magnitude of a complex number.

$$SCF\{\boldsymbol{x}_a\} = \mathcal{F}^{-1}\{abs(F\{\boldsymbol{x}_a\})\} \tag{3.4}$$

Therefore, the SCF does not square the amplitudes of Fourier components. The double SCF is achieved by applying the SCF again to a first SCF converted to polar coordinates, just in the same way as the DACF. On the other hand, the SCF throws away the phases of the image Fourier transform, thus causing potentially severe information loss.

3.2.3 Zernike moments

Another way to achieve rotational invariance is to project the image onto a space of *Zernike polynomials* which are defined within the unit circle. These are named after Frits Zernike, winner of the 1953 Nobel prize in Physics for the invention of the phase contrast microscope. *Zernike moments* can be understood as a type of weighted average of an image's pixel intensities whose weights are given by the Zernike polynomials. Following the notation used by Chang & Ghosh (2000), the Zernike moment Z_{ab} of order a with repetition b for an image $I(x, y)$ is given by Equation 3.5:

$$Z_{ab} = \frac{a+1}{\pi} \sum_x \sum_y I(x,y) V_{ab}^*(r,\theta) \tag{3.5}$$

with its respective Zernike polynomial V_{ab} given by Equation 3.6:

$$V_{ab}(r,\theta) = \left[\sum_{s=0}^{0.5(a-|b|)} (-1)^s \frac{(a-s)!}{s!\left(\frac{a+|b|}{2}-s\right)!\left(\frac{a-|b|}{2}-s\right)!} r^{a-2s} \right] e^{jb\theta} \tag{3.6}$$

where a is a non-negative integer, b is integer subject to constraints $(a-|b|)$ be even and $|a| \leq b$, and $x^2 + y^2 \leq 1$.

There are at least three interesting properties of Zernike moments. The first is that the Zernike polynomials form a fixed orthogonal basis which may be useful for dimensionality reduction (see Section 3.3). The second one is that the magnitudes of Zernike moments are rotation invariant. And the other one is that the rotation between two images can be inferred by the phase difference of their Zernike moments. Thus an interesting representation of a set of images may be given by calculating their Zernike moments for a range of orders and repetitions. Figure illustrates the magnitudes of a few Zernike polynomials.

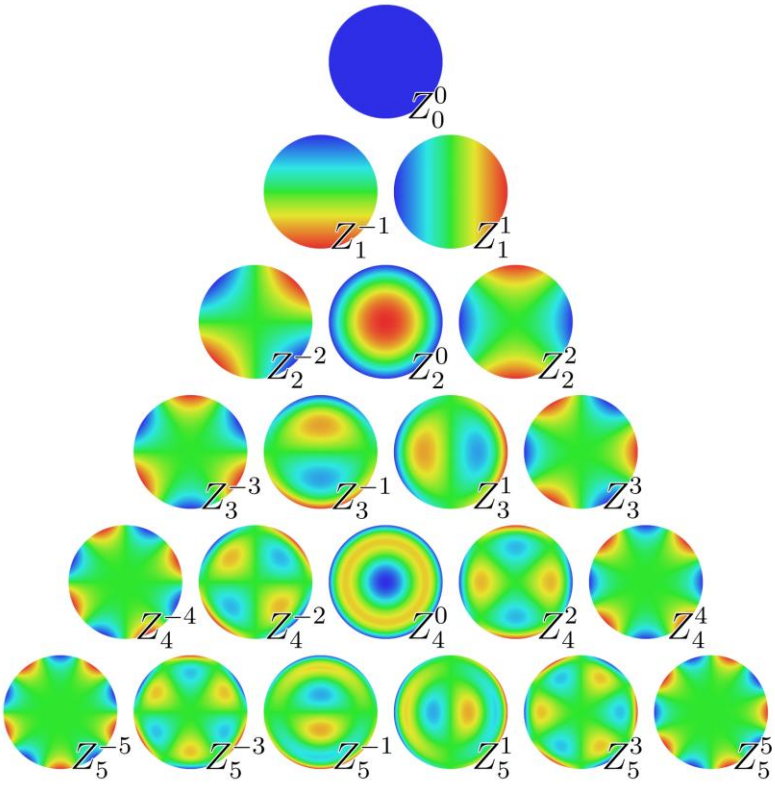


Figure 3.2 – Magnitudes of the first 21 Zernike polynomials. Blue correspond to small values and red correspond to large values. Source: Wikimedia Commons[3].

[3] http://upload.wikimedia.org/wikipedia/commons/3/3d/Zernike_polynomials2.png

Zernike moments have been used in the classification of 3D objects from 2D silhouettes in military aviation (Chang & Ghosh, 2000) and in 3D feature extraction of protein atomic models (Grandison, Roberts & Morris, 2009), among other applications. However, it has not been explored in the classification of projection images for single particle analysis. Results of clustering experiments with Zernike moments will be shown in Chapter 5.

3.3 Dimensionality reduction

Classification of electron microscopy images of biological entities has two main challenges. The first one is the high dimensionality of the native data representation, which are the pixel densities in the images. The invariant transforms presented in Section 3.2 may alleviate this problem but still may be not enough in this sense. High-dimensional data essentially requires more computational effort to be processed both in time and memory, and, what is conceptually worse, is less likely to allow an optimal classification. The growing complexity of a machine learning task in function of the number of features is known as the *curse of dimensionality* (Bishop, 2006; Duda *et al.*, 2000). The curse is related to how distance measures behave in low and high dimensional spaces – the degrees of freedom for a function grows exponentially with the number of dimensions. The other problem is more specific to TEM images acquired in low electron dose conditions, as explained in Section 2.3, and is the low signal-to-noise ratio of the data. The severe noise makes comparisons performed directly on the image pixels often meaningless. This is also the reason why conventional feature selection and extraction procedures, like removing redundant or correlated variables (Duda *et al.*, 2000; Guyon & Elisseeff, 2003) are of little use on this kind of data. In order to avoid these problems, it is desired to achieve a representation of the dataset with *reduced dimensionality*. If the dataset can be plotted in two or three dimensions, it becomes possible to visualize it and get a better intuition of what is happening in the hyper-dimensional space; for example, natural groupings or "clouds" of data may become apparent. Not only a representation with a small number of features is desired, but it should also be meaningful in some sense. This leads us to *component analysis*, which is a set of techniques aiming to find the most "interesting" directions to observe our data. How "interesting" is defined depends on the specific technique adopted.

3.3.1 Principal Component Analysis

Principal Component Analysis (PCA), also known as the Karhunen-Loéve transform, is one of the most powerful and widespread techniques of dimensionality reduction and data visualization. The goal of PCA is to find the optimal representation of the *P*-dimensional dataset $\boldsymbol{X}$ onto an *M*-dimensional subspace, $M < P$, preserving as much information as possible, in a sum-of-squared-errors sense. This subspace is composed of the directions of largest variance within the data cloud, as illustrated in Figure 3.7.

Following the explanation by Bishop (2006), let's first consider a one-dimensional representation of our data in this way, i.e. $M = 1$. In the original *P*-dimensional space, the direction of this projection will be given by a unit vector $\boldsymbol{v}_1$. Consider the empirical mean of the sample given by the formula in Equation 3.7:

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \tag{3.7}$$

The variance of the projected data is then given by Equation 3.8:

$$\frac{1}{N}\sum_{n=1}^{N} \{\boldsymbol{v}_1 \boldsymbol{x}_n^{\mathrm{T}} - \boldsymbol{v}_1 \overline{\boldsymbol{x}}^{\mathrm{T}}\}^2 = \boldsymbol{v}_1 \boldsymbol{C} \boldsymbol{v}_1^{\mathrm{T}} \tag{3.8}$$

where $\boldsymbol{C}$ is the $P \times P$ dataset covariance matrix defined in Equation 3.9:

$$\boldsymbol{C} = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}_n - \overline{\boldsymbol{x}})^{\mathrm{T}} (\boldsymbol{x}_n - \overline{\boldsymbol{x}}) \tag{3.9}$$

Therefore, in order to maximize the variance of the projected data, one must find the maximum of $\boldsymbol{v_1 C v_1^{\mathrm{T}}}$ with respect to $\boldsymbol{v}_1$, subject to $\|\boldsymbol{v}_1\| = 1$. This constrained optimization problem can be converted to an unconstrained version by introducing the Lagrange multiplier λ_1 as in the optimization problem of Equation 3.10:

$$\max_{\boldsymbol{v}_1} \boldsymbol{v}_1 \boldsymbol{C} \boldsymbol{v}_1^{\mathrm{T}} - \boldsymbol{\lambda}_1 (1 - \boldsymbol{v}_1 \boldsymbol{v}_1^{\mathrm{T}}) \tag{3.10}$$

By deriving Equation 3.10 with respect to $\boldsymbol{v}_1$ and setting the result to zero, Equation 3.11 shows clearly that $\boldsymbol{v}_1$ is an eigenvector of the covariance matrix. Also, λ_1 is an eigenvalue associated with $\boldsymbol{v}_1$ and corresponds to the projected variance, as shown in Equation 3.12.

$$\boldsymbol{C} \boldsymbol{v}_1^{\mathrm{T}} = \lambda_1 \boldsymbol{v}_1^{\mathrm{T}} \tag{3.11}$$

$$\boldsymbol{v}_1 \boldsymbol{C} \boldsymbol{v}_1^{\mathrm{T}} = \lambda_1 \tag{3.12}$$

Solving this eigenvector-eigenvalue problem yields the first principal component $\boldsymbol{v}_1$ and its associated variance λ_1. In fact, this component corresponds to the unit vector in the direction of the sample mean from the origin of the coordinate system, if the mean has not been removed for the calculation of the covariance matrix as in Equation 3.9. Nevertheless, the eigenvector-eigenvalue formulation is useful for finding the other principal components. The second principal component $\boldsymbol{v}_2$ will be given by the direction of maximum variance orthogonal to $\boldsymbol{v}_1$; the third one must maximize the projection variance orthogonally both to $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, and so on. The set of *M* principal components can be found by determining the first *M* eigenvectors of the covariance matrix $\boldsymbol{C}$, as the eigenvectors of a real symmetric matrix have the orthogonality property. Efficient algorithms for performing this *eigendecomposition* are available, such as the iterative "power method" (Golub & Van Loan, 1996; van Heel *et al.*, 2009). The total number of eigenvectors associated with nonzero eigenvalues is determined by the rank of the dataset matrix $\boldsymbol{X}$. Therefore, the principal components are found in decreasing order of their associated projection variance, which can be understood as a measure of "importance" in explaining the data distribution. The first components explain the most of the data variance, while the last ones explain the least. The *spectrum* of eigenvalues of the covariance matrix is an important tool in determining the number of relevant principal components for a given dataset. Figure 3.3 illustrates an exemple of eigenvalue spectrum.

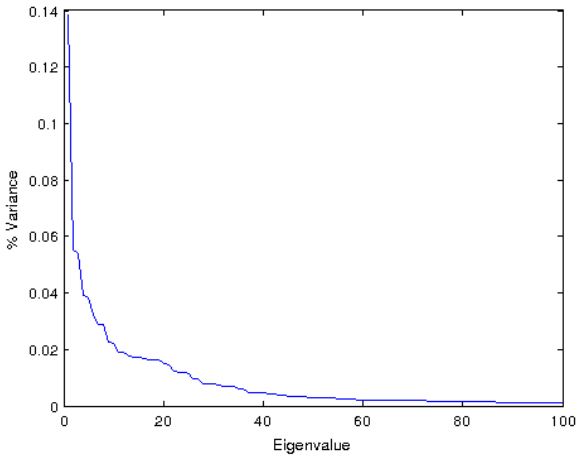


Figure 3.3 – Eigenvalue spectrum of the covariance matrix for one of the datasets analyzed in this work. The vertical axis show the magnitude of the eigenvalues as a fraction of the total dataset variance.

However, a particular component may be important to describe the whole dataset, but not a particular data point, which means that the projection of this point onto this component is close to zero. The converse may also occur, in which case the specific component is likely to be found among the last ones. For being orthonormal, the principal components span a subspace onto which the data can be linearly decomposed, as described in Equation 3.13:

$$\boldsymbol{u}_n = \boldsymbol{\alpha}_{n1}\boldsymbol{v}_1 + \boldsymbol{\alpha}_{n2}\boldsymbol{v}_2 + \cdots + \boldsymbol{\alpha}_{nM}\boldsymbol{v}_M \tag{3.13}$$

Therefore, the linear decomposition of the dataset $\boldsymbol{X}$ onto M principal components arranged as rows of the $M \times P$ matrix $\boldsymbol{V}$ is given by $\boldsymbol{U}$, which contains the projected data, as in Equation 3.14:

$$\boldsymbol{U} = \boldsymbol{X}\boldsymbol{V}^{\mathrm{T}} \tag{3.14}$$

When using the projection coefficients of PCA as features for further manipulation of the data, like clustering, it is important to bear in mind the associated variance of each component implies a natural "weighting" for each dimension (Figure 3). If one expects the components to have the same weight, one can normalize the coefficients on each of them by dividing by the square root of the respective eigenvalue (variance) (van Heel, 1984). This is especially important if one arbitrarily selects components for the desired task. One of the interesting things of dealing with an image dataset is that the eigenvectors can be visualized as "eigenimages". This visual

feedback can give insight on the "meaning" of each principal component, and what are the most relevant features of the dataset. In single particle analysis, inspection of the eigenimages gives relevant information on the symmetry of the molecule (van Heel *et al.*, 2009). The "eigenfaces" method represents also an important application of PCA in face recognition (Turk & Pentland, 1991). The eigenimages associated with the smallest eigenvalues are likely to represent only noise. By truncating the number of principal components used, one obtains a compressed representation of the dataset. Bringing back the compressed data to the original *P*-dimensional space yields an "eigen-filtered" version of an image, as illustrated in Figure 3.4.

Figure 3.4 – Eigendecomposition of images from the Olivetti Research Face Database. a) 100 images consisting of 112×92 grayscale pixels ($P = 10{,}304$); b) Reconstitutions of the images in a) using the 49 eigenfaces in c) ("eigenfiltering"); c) the eigenimages of the dataset in a) ("eigenfaces"). Adapted from (Barber, 2012).

As explained in Section 2.7.1, PCA and related multivariate statistical analysis methods have been widely used in single particle analysis since the beginning of the 1980's (van Heel & Frank, 1981; van Heel *et al.*, 2000; van Heel *et al.*, 2009). This is due to the high explanatory power contained in a compressed representation of the data, which is robust to noise and tends to reduce the computational efforts of classification algorithms. Also, they may allow the identification of relevant features related to molecular properties, such as symmetry, size and structural heterogeneity, as presented in Section 2.7. Figure 3.5 shows eigenimages for a typical SPA dataset, in which the aforementioned molecular features can be observed. A note of caution is worth mentioning, however: the projection directions provided by PCA may be useful for visualization and data compression, but this does not imply they are necessarily good for classification or clustering. Other variants of principal components analysis that have been developed include non-

linear PCA (Scholkopf, Smola & Muller, 1996), probabilistic PCA (Tipping & Bishop, 1999) and principal surfaces (Chang & Ghosh, 2001). These variants remain yet to be explored on single particle analysis datasets.

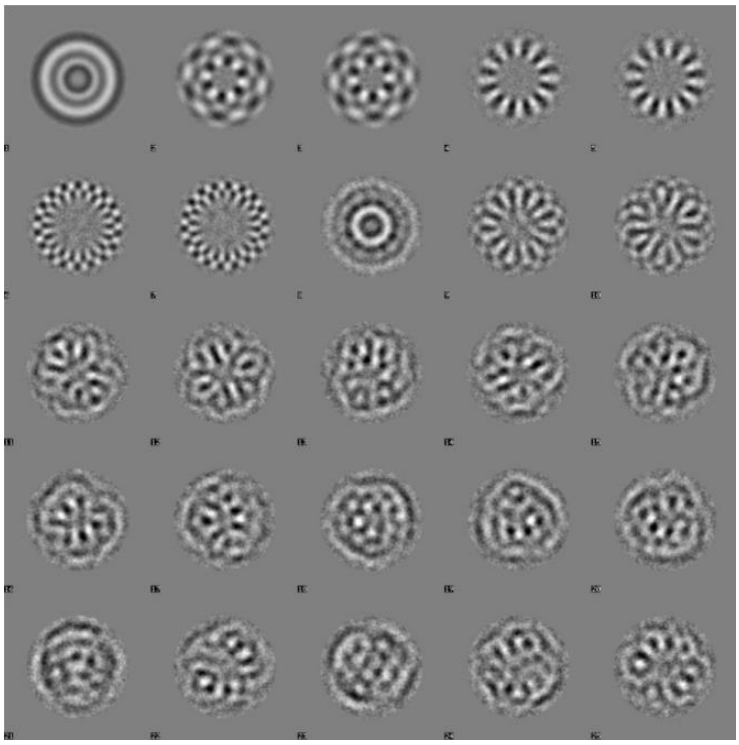


Figure 3.5 – The first 25 eigenimages of a dataset containing 7,300 projection images of *Lumbricus terrestris* hemoglobin with circular mask. The first eigenimage reveals the average molecule size, while eigenimages 2 to 7 clearly contain symmetry-related information. The SNR of the eigenimages degrades towards the smaller eigenvalues, indicating they are mostly associated with random fluctuations. Extracted from van Heel *et al.* (2009).

3.3.2 Correspondence Analysis and other metrics

One important thing to notice about conventional PCA is that it relies on the Euclidean distance for assessing the covariance of the dataset. Euclidean distance has a close relation to the correlation $C(\boldsymbol{x}, \boldsymbol{y})$ between two signals $\boldsymbol{x}$ and $\boldsymbol{y}$, as demonstrated by Equation 3.15:

$$
\begin{aligned}
d(\boldsymbol{x},\boldsymbol{y})^2 = \|\boldsymbol{x}-\boldsymbol{y}\|_2 & \\
&= \sum_i (x_i - y_i)^2 \\
&= \sum_i (x_i^2 + y_i^2 - 2x_i y_i) \\
&= \sum_i x_i^2 + \sum_i y_i^2 - 2\sum_i x_i y_i = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} + \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} - 2\,\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}} \\
&= C(\boldsymbol{x},\boldsymbol{x}) + C(\boldsymbol{y},\boldsymbol{y}) - 2C(\boldsymbol{x},\boldsymbol{y})
\end{aligned}
\tag{3.15}
$$

Therefore, when the Euclidean distance between two data points is small, their correlation is large, and vice-versa. The covariance matrix defined in Equation 3.9 can be understood as a measure of the correlation between each pair of columns of the dataset matrix $\boldsymbol{X}$, that is, the correlation between features. The disadvantage of simple correlation is that it is sensitive to multiplication by a constant. However, multiplication by a constant does not affect the information contained in a vector, and hence should not impact its similarity in relation to other vectors. The Chi-squared (χ^2) metric is able to correct for this distortion by normalizing each signal vector by its average:

$$
\chi^2(\boldsymbol{x},\boldsymbol{y}) = \left(\frac{1}{x_{avg}}\right)\boldsymbol{x}\left(\frac{1}{y_{avg}}\right)\boldsymbol{y}^{\mathrm{T}} \tag{3.16}
$$

If the χ^2 distance defined in Equation 3.16 is used in the covariance matrix instead of the conventional correlation between the features of the dataset (Equation 3.9), Principal Component Analysis then becomes *Correspondence Analysis* (CA). This technique has been proposed to analyze contingency tables (Benzécri, 1992). Historically, CA was the first and longest used dimensionality reduction technique employed to investigate data clouds of single particle images (van Heel & Frank, 1981), and it was used because the computer programs developed by Jean-Paul Bretaudiére were readily available in that context (van Heel *et al.*, 2009). Figure 3.6 presents a manual cluster analysis performed in this seminal work. Later, the problems of using CA became apparent. The χ^2 distance is suitable only for positive-valued data, like histogram data. If the dataset has negative values, an explosive behavior appears in Equation 3.16 if the average of a signal vector is close to zero. For the reasons presented in Section 2.4.2, the boxed particles are usually normalized to have zero mean value. One could argue that negative values could be discarded for the use of CA, but this obviously implies losing dataset information. Another option

would be to add a constant value to the data so all values become positive, but this would turn large-magnitude negative values into small-magnitude positive values, underestimating their contribution to the total dataset variance (van Heel *et al.*, 2009). Borland & van Heel (1990) then proposed the use of the *modulation metric*, also known as the *normalized correlation*, which normalizes the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ by their respective standard deviations σ_x and σ_y when assessing the similarity between them:

$$C_{mod}(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{1}{\sigma_x}\right)\boldsymbol{x}\left(\frac{1}{\sigma_y}\right)\boldsymbol{y}^T \tag{3.17}$$

The modulation distance defined in Equation 3.17 is well suited for real-valued data and also corrects for any constant multiplication factors between the signals being compared. Nevertheless, if the data is normalized by pre-processing steps as depicted in Section 2.4.2, the conventional correlation is expected to work just fine. In their review of multivariate statistical analysis techniques applied to cryo-EM data, van Heel *et al.* (2009) summarize the formulation of PCA/CA with general metrics, also covering the analysis in the reciprocal space, where the pixels are regarded as observations and the images as features.

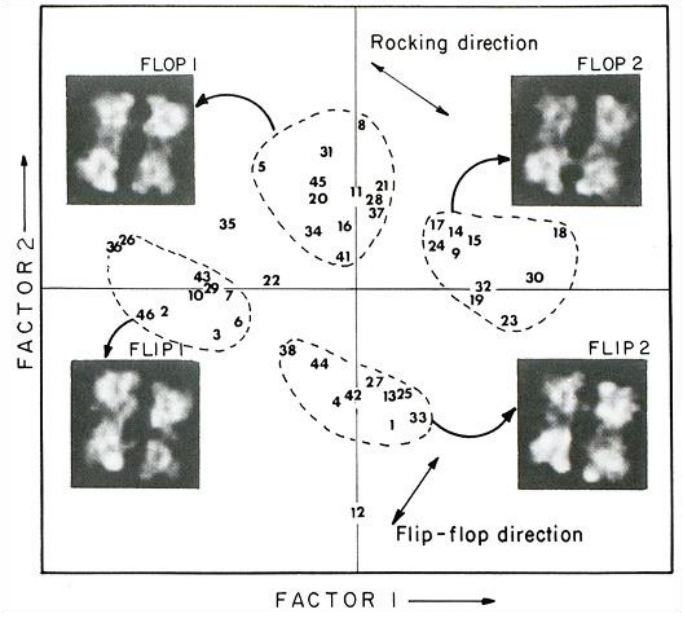


Figure 3.6 – Correspondence analysis of a dataset containing projection images of *Limulus polyphemus* hemocyanin particles embedded in negative stain. This is the first example of a dimensionality reduction procedure applied to single particle analysis. Extracted from van Heel & Frank (1981).

3.3.3 Independent Component Analysis

Whereas PCA and related techniques aim to decompose the dataset onto a lower-dimensional subspace that minimizes the squared error with the original representation, *Independent Component Analysis* (ICA) seeks directions that are *statistically independent* from each other. This difference is illustrated in Figure 3.7. ICA is typically employed in *blind source separation* problems, like distinguishing mixed acoustic sources collected by a set of microphones, or separating the contributions from each electrode attached to a patient's head in brain activity imaging. While the orthonormality of principal components imply they are *uncorrelated*, this is does not assure their statistical independence. Hyvärinen, Karhunen & Oja (2001) treat independence as *non-linear uncorrelatedness*, i.e., two independent random variables should remain uncorrelated even after an arbitrary non-linear transformation be applied. Independence also means that the random variables must carry minimal mutual information, or maximal mutual entropy.

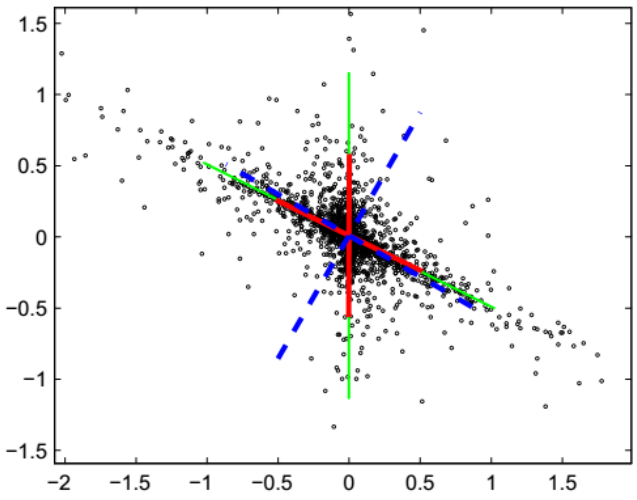


Figure 3.7 – Comparison between ICA and PCA. The data has been generated by sampling from a 2D exponential distribution along the green lines, whose directions are given by the mixing matrix $\boldsymbol{A}$ in Equation 3.18. Blue dashed lines represent the orthogonal directions of largest variance obtained by PCA. The red lines represent the directions estimated by ICA, which correspond to the directions from which the data coordinates have been independently sampled. Extracted from Barber (2012).

For the basic ICA model, suppose the observed data matrix $\boldsymbol{X}$ is generated by a linear mixture of K independent signals arranged in rows of the matrix $\mathbf{S}$:

$$\boldsymbol{X} = \boldsymbol{AS} \tag{3.18}$$

Although the mixing coefficients in matrix $\boldsymbol{A}$ of Equation 3.18 are unknown, the linearity of the operation still allows the estimation of the source signals that compose the matrix $\boldsymbol{S}$. To account for the non-linear uncorrelatedness, the model from Equation 3.19 can be applied (Duda *et al.*, 2000). The matrix $\boldsymbol{W}$ gives the weights of the model and $\boldsymbol{w}_0$ is a bias vector. $f[\cdot]$ is an arbitrary non-linear function, like a sigmoid, for example.

$$\boldsymbol{Y} = f[\boldsymbol{WX} + \boldsymbol{w}_0] \tag{3.19}$$

The task of ICA is then to find estimates for the *K* rows of matrix $\boldsymbol{Y}$ in Equation 3.19 that are as independent as possible from each other. Classically, this independency can be measured by the *joint entropy* (Shannon, 1948) between random variables, as defined in Equation 3.20, which then becomes the criterion to be maximized. $p(x_i, y_j)$ is the joint probability of occurrence of particular values x_i and y_j.

$$H(\boldsymbol{x}, \boldsymbol{y}) = -\sum_i \sum_j p(x_i, y_j)[log_2 p(x_i, y_j)] \tag{3.20}$$

Depending on the algorithm used for ICA, this optimization may be carried out by gradient descent, where a learning rule is derived for $\boldsymbol{W}$ and $\boldsymbol{w_0}$, or by the Expectation-Maximization algorithm (Section 3.4.3.3). The basics of Independent Component Analysis can be found in the pattern recognition books by Duda, Hart & Stork (2000) and Bishop (2006), and for further details the book by Hyvärinen, Karhunen and Oja (2001) specifically covering ICA is recommended.

3.4 Unsupervised classification

We shall now turn our discussion to the main topic of this dissertation, which is the unsupervised classification of data. While the previous Sections presented some pre-processing and feature extraction methods, we will now introduce some of the algorithms that seek to group the dataset in homogeneous partitions. Special attention will be given to the methods that have gained popularity among the cryo-EM community, as presented in Chapter 2, and also to those that have been chosen specifically for this project. Both individual and ensemble methods will be discussed. But before that, we shall discuss some of the formalities regarding supervised and unsupervised classification and the different types of clustering.

3.4.1 Clustering and Classification

One probably has noted that the words "clustering" and "classification" have been used interchangeably in this text and in the SPA literature in general. Nevertheless, it is necessary to clarify their formal differences. In the machine learning literature, *classification* refers to the task of assigning data points to pre-determined discrete categories or classes, based on a model that was trained using previously observed data with known labels (Bishop, 2006). These data are usually referred to as the *training set*. "Training", in this context, means tuning the parameters of a mathematical model, namely the *classifier*. It is often an optimization problem, whose basic goal is to maximize the performance of the classifier on the training set. But also the classifier must have *generalization power*, so to have a satisfactory performance on data that is not in the training set. This can be achieved by regularization or cross-validation techniques (Haykin, 1999). Given that it uses a labeled training set, classification is a *supervised* learning task. Probably the simplest classifier is the k-nearest neighbors algorithm, which decides the class for each new data-point by voting among its *k* closest neighbors in the training set (Duda *et al.*, 2000).

The unsupervised counterpart of classification is *clustering*, which seeks to partition the dataset into *K* groups based solely on its intrinsic structure (Bishop, 2006). Clustering algorithms look for natural groupings or "clouds" of data. How "natural" is defined depends on the underlying assumptions each algorithm makes about the data distribution, and the similarity measures

employed (Duda *et al.*, 2000). Usually, the number of groups *K* is defined by the user, based on specific domain knowledge. However, the clusters obtained need not correspond to a well-defined class in the human interpretable sense. It just means that the data points assigned to the same cluster are more similar to each other than to points assigned to a different cluster, according to the algorithm's criterion. The simplest clustering procedure is the k-means algorithm, which seeks to discover *K* centroids in the dataset and the clusters are formed by the points sharing the same nearest centroid (Section 3.4.3.2). A *centroid* is the arithmetic mean of points within a cluster. The books by Everitt, Landau & Leese (2001) and Barber (2012) extensively covers the clustering task, while the basic concepts and algorithms may also be found elsewhere (Bishop, 2006; Duda *et al.*, 2000; Haykin, 1999).

In general, what we are interested in when discussing classification of electron microscopy data of single particles, being it with respect to 2D or 3D information, is *unsupervised* classification. Often, no training data like a previous structural model is available, and producing it manually is very challenging due to the low SNR and the large amounts of collected data, typically in the order of tens of thousands of images. Therefore, unless we explicitly specify a supervised classification procedure, the word "classification" here will always refer to "unsupervised classification", and therefore will be used as a synonym for "clustering".

3.4.2 Types of clustering

Clustering algorithms can provide different types of outcomes for the cluster assignments, depending on whether a data point belongs to only one or possibly to multiple clusters simultaneously, or according to the uncertainty of the assignments. The different types of assignment will be presented in Section 3.4.2.1. Also, there are different fundamental assumptions one can make about what a "cluster" means, something which will be discussed in Section 3.4.2.2.

3.4.2.1 Hard, soft and fuzzy assignments

A *hard* assignment is the most basic type of cluster label. It means that a given object must belong to one, and only one, of the K clusters. Following the notation from Hruschka, Campello, Freitas & de Carvalho (2009), if we represent our clusters as non-empty collections of data points $\boldsymbol{C} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \dots, \boldsymbol{C}_K\}$, $\boldsymbol{C}_k \neq \emptyset$ and $\boldsymbol{C}_i \cap \boldsymbol{C}_j = \emptyset$ for $i \neq j$, then with hard partitions we have $\boldsymbol{C}_1 \cup \boldsymbol{C}_2 \cup \dots \cup \boldsymbol{C}_K = \boldsymbol{X}$. $|\boldsymbol{C}_k|$ is the *cardinality* of set $\boldsymbol{C}_k$ and corresponds to the number of elements it contains. The $N \times 1$ vector of labels $\boldsymbol{\lambda}$ contains the cluster assignments for each object, $\lambda_n \in \{1, \dots, K\}$, $n = 1, \dots, N$. Algorithms like hierarchical clustering (Section 3.4.3.1) and k-means (Section 3.4.3.2) provide hard partitions in their conventional formulations. Throughout this text, we will assume the *canonical form* of label lists for hard partitions (Strehl & Ghosh, 2002), which satisfies two conditions:

i) the label of the first object in the list is 1;

ii) the label for any of the successive objects in the list has either one of the already assigned values, or a value one greater than the highest previously assigned, up to K.

If overlapping partitions are allowed, then we may have a *fuzzy* clustering setup, where each data point belongs to one or more clusters with different *degrees* of membership. A special case of fuzzy clustering is when the objects may fully belong to one or more clusters with equal degrees of membership. The classical fuzzy clustering algorithm is fuzzy k-means (Duda *et al.*, 2000). Note that the degree or *strength* of cluster membership provided by fuzzy clustering algorithms cannot be confused with *probabilistic* assignments, like those provided by a Gaussian mixture model (GMM), for example (Section 3.4.3.3.1). Probabilistic algorithms deal with *uncertainty* in cluster assignments, and therefore assess the likelihood of a data point belonging to one or another cluster, but this does not necessarily mean that the given point really belongs to more than one group. Fuzzy and probabilistic assignments belong subtypes of *soft* clustering. For overlapping cluster assignments, the vector list becomes an $N \times K$ matrix $\boldsymbol{\Lambda}$, where each element λ_{nk} gives the assignment of point $\boldsymbol{x}_n$ to cluster k. If the partition is overlapping in the strict sense, $\boldsymbol{\Lambda}$ may be a binary matrix, i.e. $\lambda_{nk} \in \{0, 1\}$. If the partition is fuzzy or probabilistic, $\boldsymbol{\Lambda}$ is a real-valued matrix, i.e. $\lambda_{nk} \in [0, 1]$, $\sum_k \lambda_{nk} = 1$. The constraint $\sum_k \lambda_{nk} = 1$ is not mandatory in fuzzy partitions.

3.4.2.2 Compactness vs. Connectedness

Another important differentiation among cluster algorithms is the kind of data groups that they look for. Algorithms based on *compactness* expect data belonging to the same cluster to have *internal similarity* according to a given metric, or, in other words, an *intra*-cluster similarity higher than *inter*-cluster similarity. This is the case of conventional clustering procedures like k-means, hierarchical clustering or mixture models. On the other hand, data points may be assigned to the same cluster if they present some *connectivity* pattern or *external similarity* (Everitt *et al.*, 2001). In this case, two points belonging to the same cluster need not be close to each other in a metric sense, but the distribution of the cluster in the feature space indicates that these points are part of a characteristic pattern, as if sampled from a multidimensional *curve*. To capture this property, often a graph or manifold representation is employed.

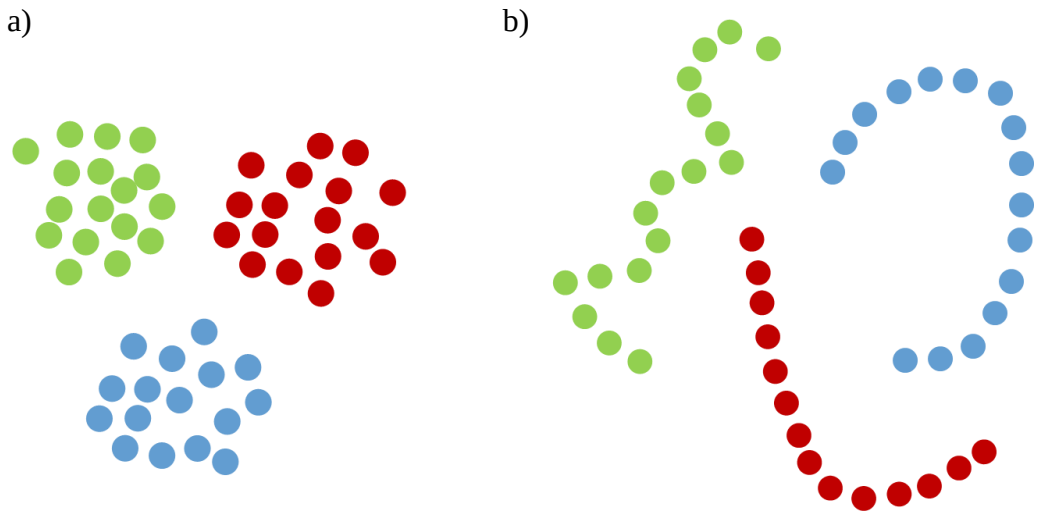


Figure 3.8 – Two types of patterns that clustering algorithms look for within the dataset. a) *Compactness* or internal similarity; b) *connectedness* or external similarity.

3.4.2.3 Assessing clustering performance

There are different ways of measuring the performance of a given clustering procedure:

- **Based on the algorithm's own cost function**

If the clustering procedure employs the optimization of a cost function, the comparison of its value after the application of the procedure is the straightforward way of assessing perfor-

mance. For example, the k-means cost function is the average distance of all data points to the centroids of the clusters they have been assigned to (Section 3.4.3.2). The partitioning solution with smallest value for the cost function is the best in this sense. This strategy is only valid when comparing solutions from different runs of the same algorithm, or from algorithms that employ the same cost function, and with the same number of clusters *K*. Also, this method only applies if there is reason to believe that the cost function properly characterizes the data clusters.

- **Based on a clustering index that is algorithm-independent**

There are clustering indexes developed to assess the quality of a partitioning solution independently of the algorithm employed. Examples are the Davies-Bouldin Index (DBI) (Davies & Bouldin, 1979) and the scatter separability criterion (Dy & Brodley, 2004). Often, clustering indexes employ some form of regularization that allows comparing solutions with different number of clusters. This is the case of DBI and also of information-theoretic model selection criteria, like Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC) and Integrated Likelihood Criterion (ICL) (McLachlan & Peel, 2000).

- **Comparing against the ground truth**

If the true labels are available for the analyzed dataset, one can simply compare how the clusters match the real classes. This is the case when one desires to assess whether the found clusters correspond to "real-world" classes, and therefore if the algorithm is able to discriminate these classes in an unsupervised fashion. Often this strategy is used to demonstrate that a totally unsupervised classification procedure is feasible on an application where only supervised classification has been employed before. This will be the preferred method of performance evaluation in this work, because our proposal will be tested on datasets whose true labels are known. When comparing lists of labels, the cluster correspondence problem arises (Strehl & Ghosh, 2002), which is, to find the correspondence between partitions on different lists that may have been generated in a different order or using different conventions. Then one may compare the percentage of matching pairs between two label lists, or employ an information-theoretic criterion that measures how well they "agree" (Acharya & Ghosh, 2013). More details about these comparisons will be given in Section 3.4.5.2.

- **Confusion matrices**

Another useful tool when comparing different hard partition solutions are the *confusion matrices* (Kuncheva, 2004). Confusion matrices display how the data is scattered across clusters in two different partitioning solutions. These solutions may be provided by two clustering algorithms, or by one clustering algorithm and the ground truth. Assuming one of the solutions is the ground truth or the best in some sense, confusion matrices allow us to assess the *purity* of clusters, which is the percentage of cluster members that are indeed from a given class. See Table 3.1 for more details.

Table 3.1 - A generic confusion matrix for label lists $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$, containing K_a and K_b clusters, respectively. $K_a \neq K_b$ is possible. n_{ij} is the number of data points from cluster i in $\boldsymbol{\lambda}^a$ that were assigned to cluster j in $\boldsymbol{\lambda}^b$.

	$\boldsymbol{C}_1^b$	$\boldsymbol{C}_2^b$	…	$\boldsymbol{C}_{K_b}^b$	*sum*
$\boldsymbol{C}_1^a$	n_{11}	n_{12}	…	n_{1K_b}	n_1^a
$\boldsymbol{C}_2^a$	n_{21}	n_{22}	…	n_{2K_b}	n_2^a
…	…	…	…	…	…
$\boldsymbol{C}_{K_a}^a$	n_{K_a1}	n_{K_a2}	…	$n_{K_aK_b}$	$n_{K_a}^a$
sum	n_1^b	n_2^b	…	$n_{K_b}^b$	N

3.4.3 Clustering algorithms

We shall now present and analyze properties of some specific clustering algorithms, chosen based on their popularity for 2D and 3D classification within the cryo-EM community (as seen in Chapter 2), or because they are potentially useful for the unsupervised classification of structural heterogeneity in cryo-EM datasets (Section 2.5). These are the algorithms that effectively take the dataset matrix $\boldsymbol{X}$, either in its native representation or after some feature transformation operation (Sections 3.2 and 3.3), and partition it into K groups.

3.4.3.1 Hierarchical Clustering

Hierarchical clustering algorithms do not seek to partition the dataset in K groups right from the beginning. Instead, they seek to discover the complete cluster structure of the dataset.

Interestingly, this structure can be visualized by means of a hierarchical tree of *N* levels called a *dendrogram* (Figure 3.9). Hierarchical a*gglomerative* or *ascendant* clustering (HAC) is a bottom-up procedure that begins considering every single data point $\boldsymbol{x}_n$ as a cluster, and, at each step, two clusters are merged according to the *minimization* of a cluster distance measure related to one of the linkage criteria to be presented in Section 3.4.3.1.1. The procedure may go up to the level in which all objects belong to the same cluster. *Divisive* or *descendant* hierarchical clustering (HDC) is the top-down counterpart of HAC: all objects begin assigned to a single cluster, and at each step a cluster is splitted in order to *maximize* one of the linkage criteria explained in Section 3.4.3.1.1.

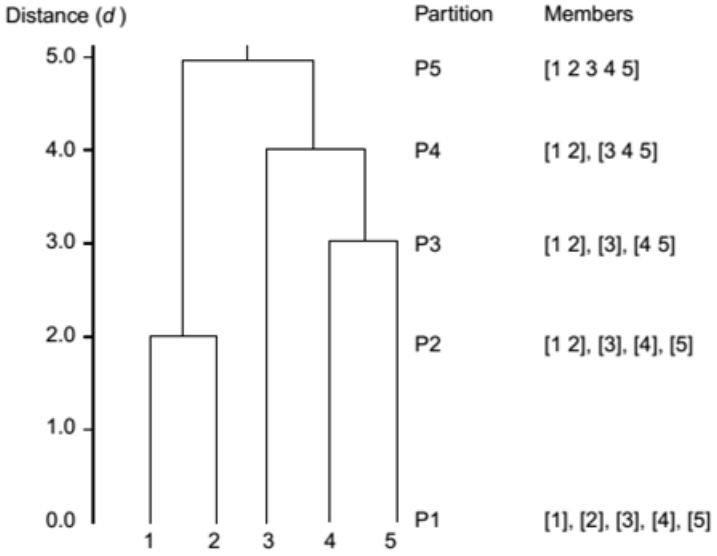


Figure 3.9 – Example of a dendrogram for a hierarchical clustering procedure applied on a dataset comprised of five objects. Extracted from Everitt *et al.* (2001).

In order to obtain *K* clusters, one can simply prune the dendrogram at the corresponding level. Note, however, that while the partition may be optimal at a given merging level of the hierarchical clustering procedure, like in a greedy optimization algorithm, it will not necessarily be optimal for a particular value of *K*. The criteria may then be refined by moving objects across clusters according to some post-processing heuristic (van Heel *et al.*, 2009). It is also possible to prune the dendrogram in order to obtain clusters with balanced number of members (van Heel, 1984). The basic algorithm for HAC returning *K* clusters is presented in Algorithm 3.1.

Algorithm 3.1: Hierarchical Ascendant Clustering

Input: dataset $\{\boldsymbol{x}_n\},\ n = 1, \dots, N$; number of clusters K; cluster distance measure $d(\boldsymbol{C}_i, \boldsymbol{C}_j)$.
Output: clusters $\boldsymbol{C} = \{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\}$.

```
1  begin
2     C_n ← {x_n},  n = 1, ..., N
3     k ← N
4     do k ← k − 1
5        i, j ← arg min_{i,j} d(C_i, C_j)
6        C_i ← C_i ∪ C_j
7     until k = K
8     (optional) apply moving objects heuristic in order to refine min Σ_i Σ_j d(C_i, C_j)
9     return C = {C_1, ..., C_K}
10 end
```

3.4.3.1.1 Cluster merging criteria

Several cluster merging or *linkage* criteria have been proposed, with distinct properties each. These criteria are the responsible for the cluster distance measure provided as input in hierarchical clustering algorithms (Algorithm 3.1). The description and discussion of each criterion essentially follows that presented by Everitt *et al.* (2001).

- **Single linkage:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = min\|\boldsymbol{x}_a - \boldsymbol{x}_b\|_2 , \boldsymbol{x}_a \in \boldsymbol{C}_i, \boldsymbol{x}_b \in \boldsymbol{C}_j$

This criterion compares two clusters by the shortest Euclidean distance between a point in cluster $\boldsymbol{C}_i$ and a point in cluster $\boldsymbol{C}_j$. It does not take into account the external cluster structure of the dataset and favors the "chaining" of clusters.

- **Complete linkage:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = max\|\boldsymbol{x}_a - \boldsymbol{x}_b\|_2 , \boldsymbol{x}_a \in \boldsymbol{C}_i, \boldsymbol{x}_b \in \boldsymbol{C}_j$

This criterion compares two clusters by the largest Euclidean distance between a point in cluster $\boldsymbol{C}_i$ and a point in cluster $\boldsymbol{C}_j$. It does not take into account the external cluster structure of the dataset, and tends to find compact clusters of similar diameters.

- **Average linkage:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = \frac{1}{|\boldsymbol{C}_i||\boldsymbol{C}_j|}\sum_{\boldsymbol{x}_a \in \boldsymbol{C}_i}\sum_{\boldsymbol{x}_b \in \boldsymbol{C}_j}\|\boldsymbol{x}_a - \boldsymbol{x}_b\|_2$

This criterion compares two clusters by the average Euclidean distance between points in cluster $\boldsymbol{C}_i$ and points in cluster $\boldsymbol{C}_j$. It is an intermediate measure between single and complete linkage, taking into account the external cluster structure of the dataset, and tends to join clusters with small variance while leaving aside those with larger variance.

- **Weighted average linkage:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = \frac{d(\boldsymbol{C}_j, \boldsymbol{C}_g) + d(\boldsymbol{C}_j, \boldsymbol{C}_h)}{2}$

This criterion compares two clusters by the average of the average linkage distances between each parent ($\boldsymbol{C}_g$ and $\boldsymbol{C}_h$) of a cluster $\boldsymbol{C}_i$ and the other cluster $\boldsymbol{C}_j$, in a recursive way. It therefore seeks to balance the influence of the number of members in each cluster when using average linkage. It is suited to cases where clusters are expected to be highly uneven sized.

- **Centroid linkage:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = \left\| \bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j \right\|_2$

This criterion compares two clusters by the Euclidean distance between the centroid of cluster $\boldsymbol{C}_i$ and the centroid of cluster $\boldsymbol{C}_j$. The cluster that contains more members between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ will have dominating influence over the new merged cluster. An issue with this criterion is that the cluster centroids are very likely to move from one level of the dendrogram to another, possibly complicating the analysis.

- **Median linkage:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = \left\| \tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j \right\|_2$

This criterion compares two clusters by the Euclidean distance between the weighted centroid of cluster $\boldsymbol{C}_i$ and the weighted centroid of cluster $\boldsymbol{C}_j$. The weighted centroid $\tilde{\boldsymbol{x}}_i$ is defined recursively as the midpoint between the centroids of the clusters that generated $\boldsymbol{C}_i$. Therefore, the weighted centroid of the new cluster will be the midpoint between $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_j$. This is the size-balanced counterpart of centroid linkage.

- **Ward criterion:** $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = \sqrt{\frac{2|\boldsymbol{C}_i||\boldsymbol{C}_j|}{|\boldsymbol{C}_i| + |\boldsymbol{C}_j|}} \left\| \bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j \right\|_2$

The Ward criterion (Ward, 1963) or *minimum added intra-class variance* criterion compares two clusters by the increase on variance (sum of squared distances to the centroid) if merging them. It seeks to minimize *intra-cluster* variance while at the same time maximizing *inter-cluster* variance. Therefore, at each level of the hierarchical procedure, a pair of clusters will be merged if their added variance is the minimum across all pairwise cluster combinations. It tends to find balanced and spherical clusters, but is sensitive to outliers. It is the most common linkage

method used for image clustering in single-particle analysis (van Heel *et al.*, 2009; van Heel, 1984, 1989).

3.4.3.2 k-means

The k-means algorithm (MacQueen, 1967) is perhaps the simplest unsupervised classification procedure, and it is one of the most used algorithms in data mining (Wu *et al.*, 2007). The goal of k-means is to find *K* representative *prototypes* or *centroids* of the dataset. The cluster assignment of each data point is then given by its nearest centroid. The basic k-means algorithm is provided in Algorithm 3.2.

Algorithm 3.2: k-means clustering

Input: dataset $\{\boldsymbol{x}_n\},\ n = 1, \dots, N$; number of clusters K; initial prototypes $\{\boldsymbol{\mu}_k\},\ k = 1, \dots, K$; divergence measure $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Output: clusters $\boldsymbol{C} = \{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\}$

begin
  $i \leftarrow 0$
  do
    $\boldsymbol{C}_k^{(i)} \leftarrow \{\boldsymbol{x}_n : k = arg\ \min_j d(\boldsymbol{x}_n, \boldsymbol{\mu}_j)\ \forall\ n,\ 1 \le n \le N\}$
    $\boldsymbol{\mu}_k \leftarrow \frac{1}{|\boldsymbol{C}_k|} \sum_{\boldsymbol{x}_n \in \boldsymbol{C}_k} \boldsymbol{x}_n$
    $i \leftarrow i + 1$
  until $\boldsymbol{C}^{(i)} = \boldsymbol{C}^{(i-1)}$
  return $\boldsymbol{C} = \{\boldsymbol{C}_1^{(i)}, \dots, \boldsymbol{C}_K^{(i)}\}$
end

Usually, the divergence measure $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is taken to be the squared Euclidean distance (Equation 3.15). The evolution of cluster assignments across the k-means iterations is illustrated in Figure 3.10. In optimization terms, the k-means algorithm is shown to minimize the cost function in Equation 3.21 (Bishop, 2006):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} d(\boldsymbol{x}_n, \boldsymbol{\mu}_k)$$
$$r_{nk} = \begin{cases} 1, \text{if } \boldsymbol{x}_n \in \boldsymbol{C}_k \\ 0, \text{otherwise.} \end{cases} \tag{3.21}$$

However, Algorithm 3.2 is guaranteed to achieve only a local minimum of Equation 3.21. The quality of this local optimum is highly dependent on the initialization of the cluster prototypes $\{\boldsymbol{\mu}_k\}$. Conventional initializations randomly sample K points from the P-dimensional space in which the data lies, or choose K points at random from the dataset. A common procedure to obtain satisfactory partitions with k-means is to try several different initializations and retain the result with the lowest value for the cost function (Equation 3.21). The k-means++ algorithm employs a probabilistic sampling method that greatly improves the chance of achieving the global optimum of Equation 3.21 (Arthur & Vassilvitskii, 2007). The increase in complexity of the sampling procedure is compensated by a decrease in the number of iterations of the k-means internal loop.

Variations of k-means include the *k-medians* algorithm, which forces the prototypes to be the P-dimensional *medians* of each cluster (Bradley, Mangasarian & Street, 1997), and the *k-medoids* algorithms, which forces the prototypes to be actual data points in each cluster (Kaufman & Rousseeuw, 1987). There is also the fuzzy counterpart of k-means, often denoted as the *fuzzy C-means* algorithm, which assigns different degrees of cluster membership for each data point (Everitt *et al.*, 2001). Such degree is often taken to be a measure inversely proportional to the distance between each point and the cluster prototypes, and it is used as a weight when updating the cluster centroids. In fact, k-means is a particular case of the Expectation-Maximization algorithm that will be presented in Section 3.4.3.3, in which the probabilistic cluster assignments in a Gaussian mixture model are hardened by forcing each point to belong only to the cluster with highest likelihood (Bishop, 2006). It was demonstrated that the dissimilarity measure used in k-means clustering can be any of a class called *Bregman divergences* (Banerjee *et al.*, 2005). Within the specific context of single particle analysis, Penczek, Zhu & Frank (1996) proposed a version of k-means that combines clustering with rotational alignment of images.

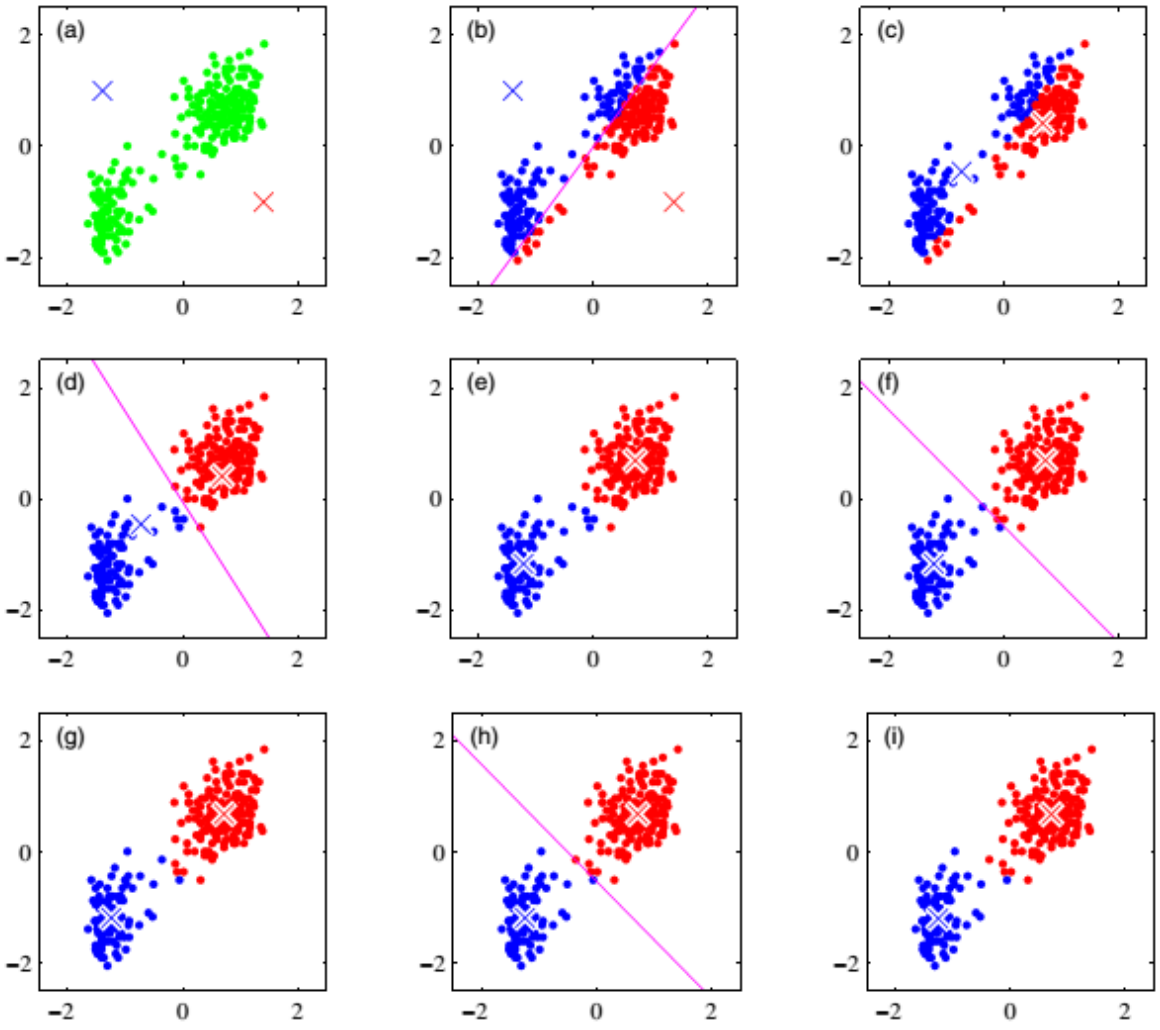


Figure 3.10 – Evolution of cluster assignments in the k-means algorithm, illustrated on the "Old Faithful" dataset in 2D and taking $K = 2$. a) Green points represent the data, the blue and red crosses are the initial prototypes $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, respectively; b) each data point is first assigned to its nearest prototype (line 4 of Algorithm 3.2); c) the prototype positions are re-calculated as the centroids of each cluster assigned in b) (line 5 of Algorithm 3.2); d) the cluster assignments are then updated following the new prototype positions, which is equivalent to classifying each data point according to which side of the bisector perpendicular to the two centroids they lie on (magenta line). The bisection is also denoted Voronoi diagram. d-i) the process is repeated until the positions of the centroids converge. Extracted from Bishop (2006).

3.4.3.3 Expectation-Maximization

The Expectation-Maximization (EM) algorithm is an iterative procedure for estimating the parameters of a probabilistic model with latent variables (Dempster, Laird & Rubin, 1977). *Latent* or *hidden* variables are those which somehow "explain" the data set, but cannot be observed. For example, in unsupervised classification, the labels for each data point are hidden variables. Whereas $\boldsymbol{X}$ is the observed or *incomplete* data, we shall denote $\boldsymbol{Z}$ as the unobserved data,

and $\{\boldsymbol{X}, \boldsymbol{Z}\}$ as the *complete* data, following the notation by Bishop (2006). Assuming the observed data were sampled from a given joint probabilistic distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$ (the *model*), the EM algorithm provides *maximum likelihood* estimates for the parameters $\boldsymbol{\theta}$. As a result, the *a posteriori* probabilities of the latent variables $\boldsymbol{Z}$ are maximized. The likelihood is defined as in Equation 3.22:

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \tag{3.22}$$

The EM algorithm iterates two steps successively until convergence to a local maximum of the likelihood function (Equation 3.22). The first is the *E step*, in which the *a posteriori* probabilities of the hidden variables, $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old})$, are calculated using the current estimates of the model parameters, $\boldsymbol{\theta}^{old}$. Using this posterior distribution for the latent variables, the expectation of the complete data log-likelihood can be computed for generic parameters $\boldsymbol{\theta}$, using Equation 3.23:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{old}) \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \tag{3.23}$$

The next step, namely the *M step*, consists of obtaining new estimates for the parameters, $\boldsymbol{\theta}^{new}$, by direct optimization of the expectation of the log-likelihood (Equation 3.23). The basic EM algorithm is depicted in Algorithm 3.3.

Algorithm 3.3: EM algorithm

Input: observed data $\boldsymbol{X}$, joint distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$, initial parameter estimates $\boldsymbol{\theta}^{(0)}$.
Output: parameter estimates $\boldsymbol{\theta}$.

```
1  begin
2    i ← 0
3    do
4      evaluate p(Z|X, θ^(i))
5      θ^(i+1) ← arg max_θ Q(θ, θ^(i))
6      i ← i + 1
7    until θ^(i) = θ^(i−1)
8    return θ^(i)
9  end
```

Expectation-Maximization is regarded as the most general unsupervised learning procedure (Schlesinger & Hlavac, 2002). An interesting property of the EM algorithm is its monotonicity, that is, given an estimate $\boldsymbol{\theta}^{old}$ for the parameter values, a lower bound for the log-likelihood is always guaranteed (Bishop, 2006; Schlesinger & Hlavac, 2002).

In the single particle analysis field, the EM algorithm was introduced by Sigworth (1998) in order to improve the translational and rotational alignments of images. Instead of "hard" alignments performed by cross-correlations, the expected aligned version of the image set $\boldsymbol{X}$ is computed by weighting all possible alignment values, which are the hidden variables $\boldsymbol{Z}$, by their posterior probabilities $p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})$. This statistical procedure reduces the reference bias in aligning noisy images. Scheres *et al.* (2005) later used Expectation-Maximization to provide maximum-likelihood estimates for multi-reference alignment and classification of noisy images, both in 2D and 3D, using a probabilistic image formation model (Scheres *et al.*, 2007; Scheres, Núñez-Ramírez, *et al.*, 2007). These maximum-likelihood approaches were then extended to Bayesian versions (Jaitly *et al.*, 2010; Scheres, 2012a), which impose a prior distribution $p(\boldsymbol{\theta})$ over the parameters (Bishop, 2006). The advantage of Bayesian methods is that they provide a formal way of introducing prior knowledge to the problem at hand (Eddy, 2004), which is a form of regularization (Scheres, 2012a, 2012b). An inherent computational complexity problem with methods that employ the EM algorithm is that they require integration over the whole parameter space for estimating the posterior probabilities of the hidden data in the *E step* (line 4 of Algorithm 3.3). Nevertheless, discrete approximations can be used to alleviate the computing effort (Scheres, Valle & Carazo, 2005; Scheres, 2012b; Tagare, Barthel & Sigworth, 2010). We shall now present a particular case of the EM algorithm used for data clustering, namely the Gaussian Mixture Model (GMM).

3.4.3.3.1 Mixture of Gaussians

We shall now assume that the underlying generative model for our data is a mixture of Gaussian distributions. The *P*-dimensional multivariate Gaussian or *normal* distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is given in Equation 3.24:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{P/2}} \frac{1}{\sqrt{\det(\boldsymbol{\Sigma})}} exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\right\} \tag{3.24}$$

The corresponding mixture model with *K* multivariate Gaussians having *mixing coefficients* π_k is then presented in Equation 3.25:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.25}$$

However, although we may be able to describe our observed data as a Gaussian mixture, what we really want to discover is from *which* of the *K* Gaussian distributions a particular data point was drawn. Following a simplified version of the explanation by Bishop (2006), let us assume then a hidden indicator variable z_k, such that $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$. By introducing $\boldsymbol{z}$, the mixture model becomes that in Equation 3.26:

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} \tag{3.26}$$

The conditional probabilities $p(z_k = 1|\boldsymbol{x})$, also known as the responsibilities $\gamma(z_k)$, will then be computed in the *E* step using Equation 3.27:

$$\gamma(z_k) = p(z_k = 1|\boldsymbol{x}) = \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{3.27}$$

Given the set of all observations $\boldsymbol{X}$, the log-likelihood function is that in Equation 3.28:

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{3.28}$$

By deriving Equation 3.28 and setting it to zero with respect to each parameter, the updates to be computed in the *M* step are given by Equations 3.29, 3.30 and 3.31:

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})\boldsymbol{x}_n\,, \qquad N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{3.29}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \tag{3.30}$$

$$\pi_k = \frac{N_k}{N} \tag{3.31}$$

The final GMM algorithm is provided by Algorithm 3.4.

Algorithm 3.4: Gaussian Mixture Model

Input: observed data $\boldsymbol{X}$, initial estimates for the K means $\left\{\boldsymbol{\mu}_k^{(0)}\right\}$, covariances $\left\{\boldsymbol{\Sigma}_k^{(0)}\right\}$ and mixing coefficients $\left\{\boldsymbol{\pi}_k^{(0)}\right\}$ of the distributions.

Output: parameter estimates $\{\boldsymbol{\mu}_k\}$, $\{\boldsymbol{\Sigma}_k\}$, $\{\boldsymbol{\pi}_k\}$ and posterior probabilities $\{\gamma(z_{nk})\}$.

begin
  $\mathcal{L}^{(0)} \leftarrow \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k^{(0)}\, \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)}\right)\right\}$
  $i \leftarrow 0$
  do
    $\gamma^{(i)}(z_{nk}) \leftarrow \frac{\boldsymbol{\pi}_k^{(i)}\mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k^{(i)}, \boldsymbol{\Sigma}_k^{(i)}\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}_j^{(i)}\boldsymbol{\pi}_k^{(i)}\mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j^{(i)}, \boldsymbol{\Sigma}_j^{(i)}\right)}\ \forall\, n,\ 1 \le n \le N$
    $N_k^{(i)} \leftarrow \sum_{n=1}^{N} \gamma^{(i)}(z_{nk})$
    $\boldsymbol{\mu}_k^{(i+1)} \leftarrow \frac{1}{N_k^{(i)}}\sum_{n=1}^{N} \gamma^{(i)}(z_{nk})\boldsymbol{x}_n\ \forall\, k,\ 1 \le k \le K$
    $\boldsymbol{\Sigma}_k^{(i+1)} \leftarrow \frac{1}{N_k^{(i)}}\sum_{n=1}^{N} \gamma^{(i)}(z_{nk})\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(i+1)}\right)^{\mathrm{T}}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(i+1)}\right)\ \forall\, k,\ 1 \le k \le K$
    $\boldsymbol{\pi}_k^{(i+1)} \leftarrow \frac{N_k^{(i)}}{N}\ \forall\, k,\ 1 \le k \le K$
    $\mathcal{L}^{(i+1)} \leftarrow \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k^{(i+1)}\, \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k^{(i+1)}, \boldsymbol{\Sigma}_k^{(i+1)}\right)\right\}$
    $i \leftarrow i + 1$
  until $\mathcal{L}^{(i)} = \mathcal{L}^{(i-1)}$
  return $\left\{\boldsymbol{\mu}_k^{(i)}\right\}$, $\left\{\boldsymbol{\Sigma}_k^{(i)}\right\}$, $\left\{\boldsymbol{\pi}_k^{(i)}\right\}$,$\{\gamma^{(i)}(z_{nk})\}$
end

Using Algorithm 3.4 for clustering yields the posterior probabilities $\gamma(z_{nk})$ of object $\boldsymbol{x}_n$ belonging to cluster $\boldsymbol{C}_k$. Figure 3.11 shows the evolution of the GMM algorithm on an example dataset. k-means (Algorithm 3.2) is a particular case of GMM where, at each iteration of GMM,

every object is forced to belong only to the distribution with nearest mean. It is also the equivalent of using GMM imposing identical, isotropic covariance matrices for each distribution (Bishop, 2006). A clear advantage of GMM over k-means is the ability to recognize ellipsoidal clusters, as illustrated in Figure 3.11. The partitions provided by GMM can be hardened by assigning each object to its highest posterior probability distribution *after* Algorithm 3.4 ends, if one desires to make a comparison with the labels obtained by k-means, for example. The mixture model in Equation 3.26 can be adapted to probability distributions other than the normal distribution. Scheres & Carazo (2009) improved the robustness of maximum-likelihood structure determination in SPA by using *t*-distributions, which have heavier tails than the Gaussian.

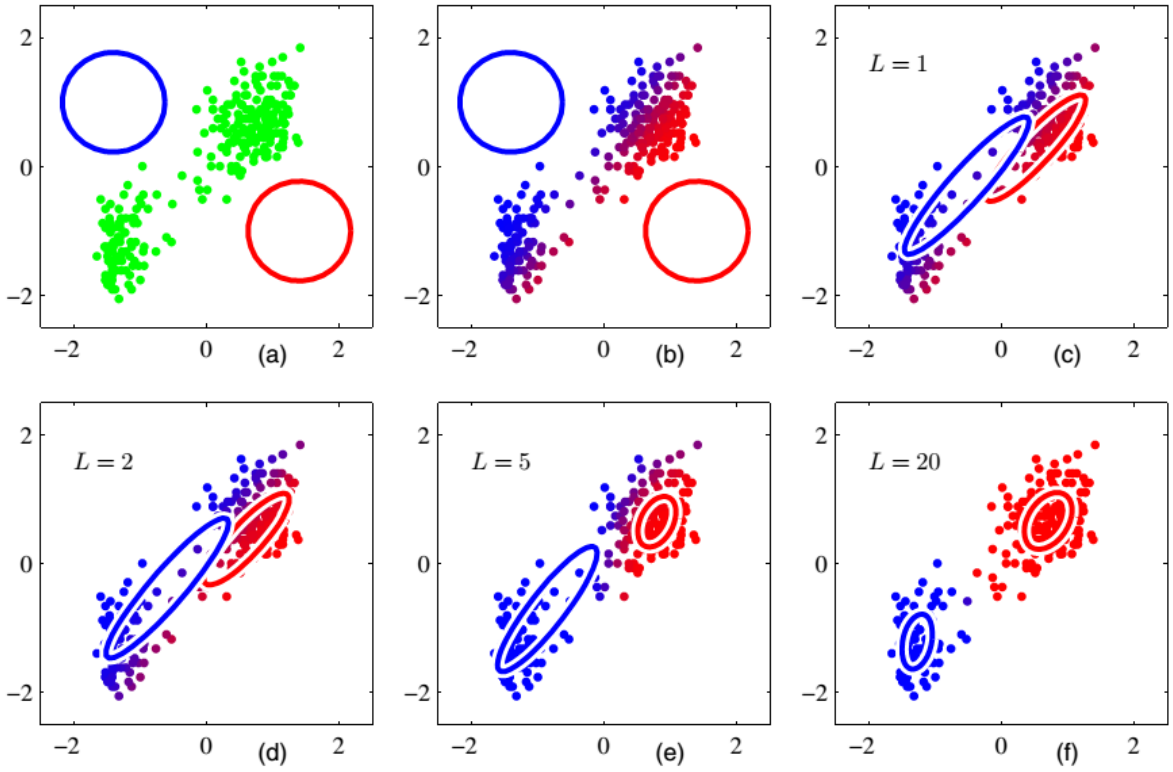


Figure 3.11 – Evolution of the determination of a two multivariate Gaussian mixture model on the "Old Faithful" dataset across iterations (L) of the GMM algorithm, in contrast to Figure 3.10. The blue and red ellipses represent one standard deviation from the mean for each of the two distributions. Extracted from Bishop (2006).

3.4.3.4 Self-Organizing Maps

Self-Organizing Maps (SOMs) are a family of unsupervised artificial neural networks (ANNs) for clustering and dimensionality reduction first proposed by Kohonen (1982). The goal of a SOM is to map the set of *N* data vectors onto a set of *M* prototypes or *nodes* in a *topologically ordered* fashion (Duda *et al.*, 2000). *M* may or may not correspond to the *K* clusters expected, depending on how the SOM is used, so that the relationship $K \leq M \leq N$ holds. The prototypes or *neurons* of the network are adjusted so to best represent the distribution of the observed data while preserving a neighborhood relationship (Duda *et al.*, 2000). This relationship, or *topology*, is defined on a low-dimensional grid, called the feature *map*, typically defined in 1D, 2D or 3D for ease of visualization. The process of learning the neuronal weights has a competitive nature (Theodoridis & Koutroumbas, 2008), but one which affects not only the winning node but also its neighbors, being inspired by the *synaptic plasticity* of the human brain (Kohonen, 2001). The prototypes are described both by a representation in the feature space of the data and by their position on the map, and they provide a *summarized* version of the dataset. This mapping is illustrated in Figure 3.12. This summarized version may greatly benefit the visualization of the data or the application of other supervised and unsupervised classification algorithms afterwards (Duda *et al.*, 2000).

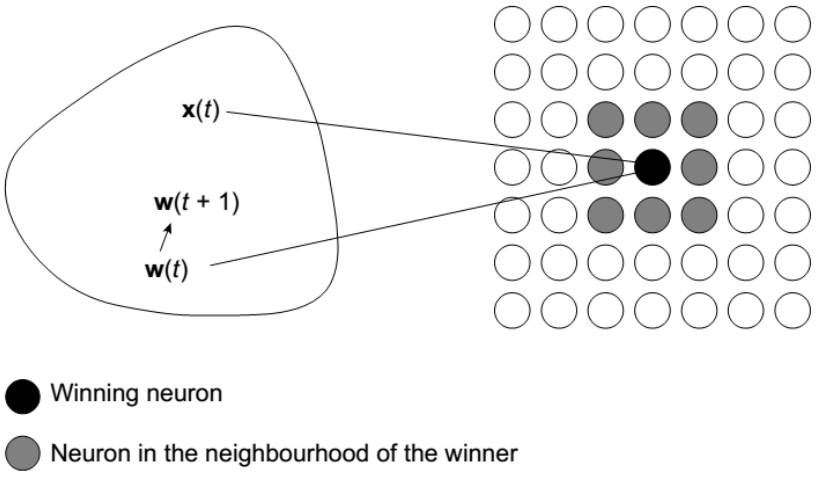


Figure 3.12 – The mapping between the input layer (observed data) on the left, and the output layer (map nodes or prototypes) on the right. The "winning neuron" is the one most similar to the input pattern presented at the moment. It is adjusted, in the feature space, to be more similar to this pattern. Its neighbors (grey nodes) are also adjusted towards the same pattern, but at a lower rate. Adapted from Everitt *et al.* (2001).

Learning the weights or features of the prototypes is a process called *training*, in which the data patterns are presented one at a time to the network, resembling an "online" version of the k-means algorithm,. Defining and training a SOM requires the specification of a few parameters. Some of them are related to the topology of the map: in 1D, we can have the prototypes arranged in a line or circle; in 2D, they may be arranged on a sheet, a cylinder or a torus, depending on how the edges of the map are joined. Also, it is necessary to impose a type of neighborhood to the neurons, which can be hexagonal or rectangular. 3D maps may also be considered, but they are not going to be used in this work. The topology may be encoded in a binary matrix $\boldsymbol{M}$ whose elements define the neighbors for each prototype. The range or "influence" of the neighborhood is usually set to decrease across iterations by a window function Λ, for stability reasons. Figure 3.13 illustrates two examples of window functions in 1D and 2D. We might also choose a *learning rate* that decreases with the number of iterations. The learning rate is the proportion to which the neurons are adjusted towards a presented data point.

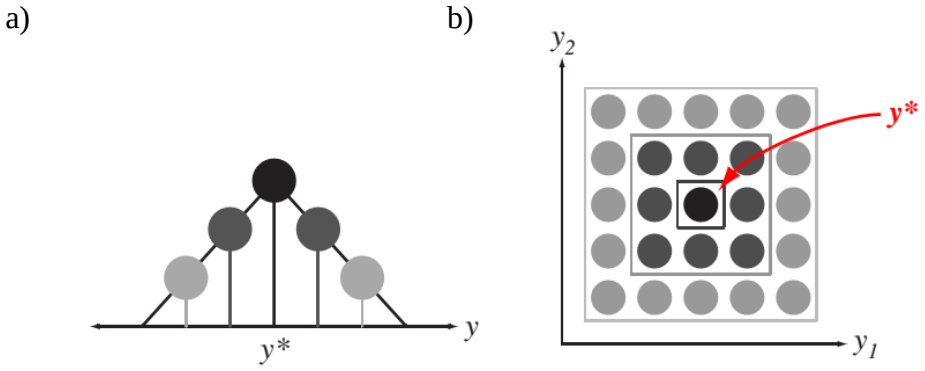


Figure 3.13 – Examples of window functions in 1D which decrease the influence of an adjustment applied to a winning neuron, here denoted $\boldsymbol{y}^*$, across its neighborhood. a) A 1D window function; b) a 2D rectangular window function. Extracted from Duda *et al.* (2000).

The basic procedure consists of presenting a data pattern at a time, in random order, and determining its most similar prototype in the map, which we shall call the *best matching unit* (BMU). The weights of the BMU and its neighbors are then adjusted to make them more similar to the matched pattern, proportionally to the learning rate and the window function. The SOM is the unsupervised counterpart of *learning vector quantization* (LVQ), a supervised algorithm that makes the neurons more or less similar to the matched pattern depending on the label of the object being correctly recognized or not (Kohonen, 1990). The basic SOM algorithm is presented in Algorithm 3.5.

Algorithm 3.5: Self-Organizing Map

Input: observed data patterns $\{\boldsymbol{x}_n\}$, M initialized prototypes $\left\{\boldsymbol{w}_m^{(0)}\right\}$, map topology $\boldsymbol{M}$, neighborhood function $\Lambda(\boldsymbol{M}, m, t)$, learning rate function $\alpha(t)$, number of iterations t_{max}.
Output: trained prototypes $\{\boldsymbol{w}_m\}$.

```
begin
    t ← 0
    do
        randomize the order of the patterns {x_n}
        for each x_n
            i ← arg min_m ||x_n − w_m^(t)||_2
            w_m^(t+1) ← w_m^(t) + α(t)Λ(M, i, t)(x_n − w_m^(t)) ∀ m, 1 ≤ m ≤ M
        end
        t ← t + 1
    until t = t_max
    return {w_m^(t)}
end
```

There are two measures commonly employed to assess the quality of a trained SOM (Kohonen, 2001). The first one is the *quantization error* (QE), which is defined as the average distance between each data point and its BMU. It quantifies how well the SOM approximates the data set, and is equivalent to the k-means cost function (3.21). The other one is the *topographic error* (TE), which is the fraction of data points for which their first two BMUs are not neighbors in the map. TE is a measure of topology preservation.

After training a SOM using Algorithm 3.5, arises the natural question of how to effectively obtain clusters. In the simplest case, one may choose $M = K$, and then cluster simply by assigning to each object the index of its BMU. In this case, SOM becomes an enhanced version of k-means, in which the cluster prototypes influence each other and tend to concentrate on regions of the feature space that are more densely populated. For an example of this kind of usage see the work by Strehl, Ghosh & Mooney (2000). Another option, in which $K < M < N$, is to provide the trained prototypes as inputs to k-means, and then cluster the native data indirectly by the label of its BMU (Kohonen, 2001). Because the trained prototypes have a "local average" nature, this may be useful when the native data is noisy, and/or the number of data points is so large that it is not practical to input them directly to k-means. An example of trained SOM can be seen in Figure 6.5, with the images representing the weights of each neuron in the map arranged on a sheet.

However, perhaps the greatest potential of using a SOM is in exploratory data analysis, because of the summarized data representation and the constrained low-dimensional topology. If

the data have a visual nature, like a set of images or electrical signals, one can inspect the prototypes arranged on the map and recognize characteristic features for each region of the map. Also, a very useful visual tool in detecting clusters in a SOM is the U-matrix (Ultsch & Siemon, 1990), which is a plot of the final distances between the weight vector of each neuron and of its direct neighbors. The U-matrix may also allow the visual determination of the number of clusters *K*. An example of U-matrix is presented in Figure 6.6.

A disadvantage of the SOM is that it is not guaranteed to optimize any cost function, which may pose challenges to parameter tuning and determining convergence during training (Bishop, 2006). The *Generative Topographic Mapping* (GTM) is a formal statistical derivate of the SOM (Bishop, Svensén & Williams, 1998). The group of Carazo and colleagues have used the SOM and its variants for unsupervised classification of single particle images (Marabini & Carazo, 1994; Pascual, Merelo, Carazo & Autnoma, 1999), including a statistical formulation based on kernel methods (Pascual-Montano *et al.*, 2001; Pascual-Montano, Taylor, Winkler, Pascual-Marqui & Carazo, 2002).

3.4.3.4.1 Growing Neural Gas

Another popular variant of the SOM is the *Growing Neural Gas* (GNG) (Fritzke, 1995). GNG differs from SOM in which it learns the data topology by automatically determining the number of neurons and which regions of the feature space they should lie on. This avoids the problem of having prototypes on "empty" regions, which sometimes may occur with the SOM by imposition of the neighborhood configuration. Also, it may automatically determine the number of clusters, by splitting unconnected sets of prototypes as representatives of different clusters. GNG has also been used on classification of cryo-EM images (Ogura, Iwasaki & Sato, 2003). The topological differences between the SOM and the GNG can be visualized in Figure 3.14.

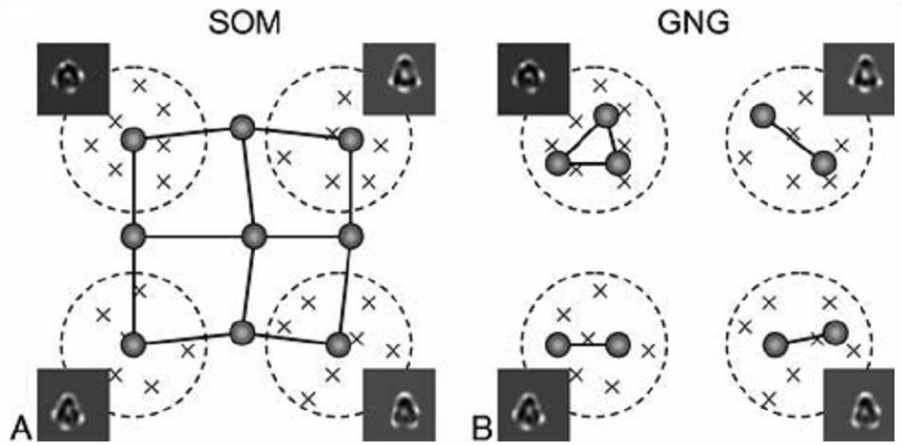


Figure 3.14 – Schematic comparison between topologies in a SOM and in a GNG. The SOM uses a fixed topology which may cause nodes to lie on unpopulated regions of the feature space (left). By changing the number of nodes and their neighborhood relationships, GNG is able to put representatives only on dense regions of the feature space (here illustrated as electron microscopy projection images of the sodium channel), and thus "naturally" determining the clusters (right). Extracted from Ogura *et al.* (2003).

3.4.3.5 Graph partitioning

The algorithms presented so far assume that clusters in the dataset are of the "compact" type presented in Section 3.4.2. Possible exceptions are the SOM and the GNG, depending on how they are used. We shall now discuss clustering by means of *graph partitioning*, which accounts for the "connectedness" of data clouds. The graph representation is particularly useful to detect the intrinsic structure of the dataset, as, for graphs, what matters are the relationships between objects, and not the feature space from which they come from. Thus, graph partitioning algorithms are able to detect clusters of arbitrary shape.

Consider an *undirected similarity graph* $G = (\boldsymbol{V}, \boldsymbol{E})$. $\boldsymbol{V} = \{v_1, \dots, v_N\}$ is the set of *vertices* or *nodes*, which in our case correspond to the N objects. $\boldsymbol{W} = \{w_{ij}\}, i, j = 1, \dots, N$ is the set of *edge weights*, where w_{ij} is the weight of the edge connecting node i and node j; $w_{ij} \geq 0$. The higher w_{ij} is, the more similar nodes i and j are, as illustrated in Figure 3.15. The *degree* of a node is given by Equation 3.32 and is a measure of its "significance" (Theodoridis & Koutroumbas, 2008). The diagonal matrix $\boldsymbol{D}$ contains the degrees d_i as its non-zero entries. $\boldsymbol{W}$ is arranged as an $N \times N$ symmetric *adjacency matrix* (because G is undirected, $w_{ij} = w_{ji}$). There are different ways to define the neighborhood and the weights of a similarity graph, which will be discussed in the next section, based on the explanation by von Luxburg (2007).

$$d_i = \sum_{j=1}^{N} w_{ij} \tag{3.32}$$

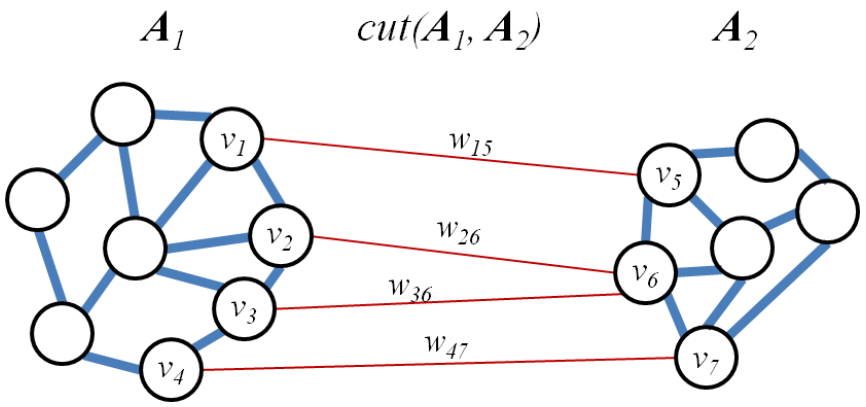


Figure 3.15 – Example of a partitioned graph. The thicker the edge between two vertices is, the more similar the vertices are (weighted graph). The red edges indicate those belonging to the *cut* between partitions $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ (Section 3.4.3.5.2).

3.4.3.5.1 Types of similarity graph

Given a dataset $\boldsymbol{X}$ whose pairwise similarities are in matrix $\boldsymbol{S}$, there are three common ways of building an adjacency matrix $\boldsymbol{W}$:

- **$\boldsymbol{\varepsilon}$-neighborhood graph**

  In this graph, the adjacency matrix $\boldsymbol{W}$ is obtained by imposing a threshold on the elements of the similarity matrix $\boldsymbol{S}$. Only nodes with a similarity higher than ε remain connected. The matrix $\boldsymbol{W}$ usually becomes sparse using this type of graph.

- **$\boldsymbol{k}$-nearest neighbors graph**

  In this graph, the adjacency matrix $\boldsymbol{W}$ is obtained by connecting each node only to its k most similar neighbors. However, if this criterion is applied in a straightforward way, a directed graph may result: a given node i may have node j among its k nearest neighbors; but i is not necessarily among the k nearest neighbors of j. This situation occurs, for example, if the dataset contains patterns that are considered "outliers", i.e., points which are located far apart from most of the data cloud. In order to make this graph undirected, one may consider $w_{ij} > 0$ when node i has

node *j* as a neighbor and/or when the converse holds; this way, each node will be connected to *at least k* neighbors. The other strategy is to consider $w_{ij} > 0$ only when node *i* has node *j* as a neighbor and when the converse also holds; this way, each node will be connected to *at most k* neighbors. The latter is known as the *mutual* k-nearest neighbors graph. This strategy may cause nodes to become isolated, and therefore this may be an approach to detect outliers. The matrix $\boldsymbol{W}$ usually becomes sparse using this type of graph.

- **Fully connected graph**

  In this graph, the adjacency matrix $\boldsymbol{W}$ is obtained by simply connecting every node to each other, weighted by their similarities. However, it may be interesting to penalize the similarity of nodes that are far apart by applying, for example, a Gaussian kernel. This is done in order to reinforce the neighborhood relationships within the graph.

With the ε-neighborhood and the k-nearest neighbors strategies, if the connected nodes are very similar to each other, one may consider the use of an *unweighted* graph ($w_{ij} \in \{0, 1\}$).

3.4.3.5.2 Goal functions

The goal of graph partitioning is to split the nodes in *K* disjoint sets $\boldsymbol{A}_k$, $\boldsymbol{A}_1 \cup \boldsymbol{A}_2 \cup \ldots \cup \boldsymbol{A}_K = \boldsymbol{V}$. $\overline{\boldsymbol{A}_k}$ is the *complement* of partition *k*, i.e., the set of nodes *not* belonging to $\boldsymbol{A}_k$. The "quality" of a partitioning may be measured according to different criteria, which are the *goal functions* of the partitioning task. We present here the three most common goal functions (von Luxburg, 2007):

- **Cut:** $cut(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K) = \sum_{k=1}^{K} \sum_{i \in \boldsymbol{A}_k, j \in \overline{\boldsymbol{A}_k}} w_{ij}$

  A graph *cut* is given by the sum of the weights of edges crossing partitions. One desires to minimize the cut (Min-cut), which is, to obtain partitions that are the least similar to each another, like in Figure 3.15. A disadvantage of Min-cut is that it may obtain meaningless or trivial cuts by simply isolating outlier vertices (Theodoridis & Koutroumbas, 2008).

- **RatioCut:** $Rcut(\boldsymbol{A}_1, \dots, \boldsymbol{A}_K) = \sum_{k=1}^{K} \frac{cut(\boldsymbol{A}_k, \overline{\boldsymbol{A}_k})}{|\boldsymbol{A}_k|}$

  The RatioCut is given by the sum of the weights of edges crossing partitions, normalized by the number of vertices in each partition. RatioCut seeks to obtain balanced partitions, that is, partitions containing approximately the same number of vertices (Hagen & Kahng, 1992).

- **N-cut:** $Ncut(\boldsymbol{A}_1, \dots, \boldsymbol{A}_K) = \sum_{k=1}^{K} \frac{cut(\boldsymbol{A}_k, \overline{\boldsymbol{A}_k})}{vol(\boldsymbol{A}_k)}$

  The N-cut is given by the sum of the weights of edges crossing partitions, normalized by the *volume* of the partitions. The volume of a partition is defined in Equation 3.33:

$$vol(\boldsymbol{A}) = \sum_{i,j \in \boldsymbol{A}} w_{ij} \tag{3.33}$$

  N-cut seeks to obtain partitions with approximately the same volume (Shi & Malik, 2000).

The next Sections will discuss graph partitioning algorithms used for clustering in this work: spectral clustering (Section 3.4.3.5.3) and METIS (Section 3.4.3.5.4).

3.4.3.5.3 Spectral clustering

Spectral clustering is a technique derived from *spectral graph theory* (Chung, 1997). It relies on the *eigenvalue spectrum* of the Laplacian matrix for a similarity graph. The unnormalized graph Laplacian is defined as:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{3.34}$$

The Laplacian matrix defined in Equation 3.34 has many interesting properties which will be mostly omitted here for simplicity (Mohar & Alavi, 1991; Mohar, 1997; von Luxburg, 2007).

What is most important to understand spectral clustering is the observation that, for any vector $\boldsymbol{f} \in \mathbb{R}^N$, the following relationship holds:

$$\boldsymbol{f}\boldsymbol{L}\boldsymbol{f}^{\mathrm{T}} = \frac{1}{2}\sum_{i,j=1}^{N} w_{ij}(f_i - f_j) \tag{3.35}$$

From Equation 3.35 it becomes clear that if $\boldsymbol{f}$ is a constant vector, it is the eigenvector corresponding to the eigenvalue zero. Also, for all $w_{ij} > 0$, Equation 3.35 can only be equal to zero if $f_i = f_j$. Therefore, we can conclude that, if *G* is composed of *K* connected components, the zero eigenvalue will have a multiplicity of *K*, with respective eigenvectors having constant values for the connected nodes. In this case, $\boldsymbol{W}$ is block-diagonal and consequently $\boldsymbol{L}$ also is so, as in Equation 3.36.

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_1 & & & \\ & \boldsymbol{W}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{W}_K \end{pmatrix}, \qquad \boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_1 & & & \\ & \boldsymbol{L}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{L}_K \end{pmatrix} \tag{3.36}$$

Thus, in this idealized case, if we want to extract the *K* connected components or clusters represented in the graph *G*, all we would have to do is to find the eigenvectors corresponding to the *K* lowest eigenvalues of the Laplacian matrix $\boldsymbol{L}$. Interestingly, even in the more practical case that the block-diagonality of $\boldsymbol{W}$ does not hold precisely, perturbation theory assures that the eigenvectors and eigenvalues of $\boldsymbol{L}$ shouldn't change significantly (von Luxburg, 2007). The indicator vectors $\boldsymbol{f}$ will then be approximately constant for the members of the roughly connected *K* components of *G* – which correspond to the clusters we are looking for from the beginning. The eigenvectors corresponding to the *K* lowest eigenvalues may then be provided as input to some other conventional clustering algorithm, say k-means. Therefore, we can say that spectral clustering *changes the representation* of the data from a connectivity point of view (the graph) to a compactness point of view (the indicator vectors). The basic algorithm for unnormalized spectral clustering, which uses the unnormalized Laplacian from Equation 3.34 and approximates a solution for the RatioCut index (Section 3.4.3.5.2) is given by Algorithm 3.6.

Algorithm 3.6: Unnormalized Spectral Clustering

Input: adjacency matrix $\boldsymbol{W}$, number of clusters K.
Output: clusters $= \{\boldsymbol{C_1}, \ldots, \boldsymbol{C_K}\}$.

```
begin
    D ← degree matrix for W
    L ← D − W
    U ← N × K matrix containing, as columns, the eigenvectors of L corresponding to its K smallest eigenvalues
    C ← K clusters returned by k-means applied on the rows of U as objects
    return C = {C_1, ..., C_K}
end
```

We note that Algorithm 3.6 receives as input an adjacency matrix $\boldsymbol{W}$ computed by one of the strategies in Section 3.4.3.5.1. Normalized versions of this algorithm have also been proposed in order to approximate the N-cut. These involve the eigendecomposition of the normalized Laplacians $\boldsymbol{L_{sym}} = \boldsymbol{D^{-1/2}LD^{-1/2}}$ (Ng, Jordan, Weiss & others, 2002) and $\boldsymbol{L_{rw}} = \boldsymbol{D^{-1}L}$ (Shi & Malik, 2000), respectively (von Luxburg, 2007). How the choices for the adjacency matrix affect the results of spectral clustering has been assessed by Maier, von Luxburg & Hein (2012). Recently, the use of spectral clustering has become a trend for 2D and 3D classification of cryo-EM images, as already presented in Section 2.7.9 (Shatsky, Hall, Nogales, Malik & Brenner, 2010; Ueno, Kawata & Umeyama, 2005; Ueno, Mio, Sato & Mio, 2007).

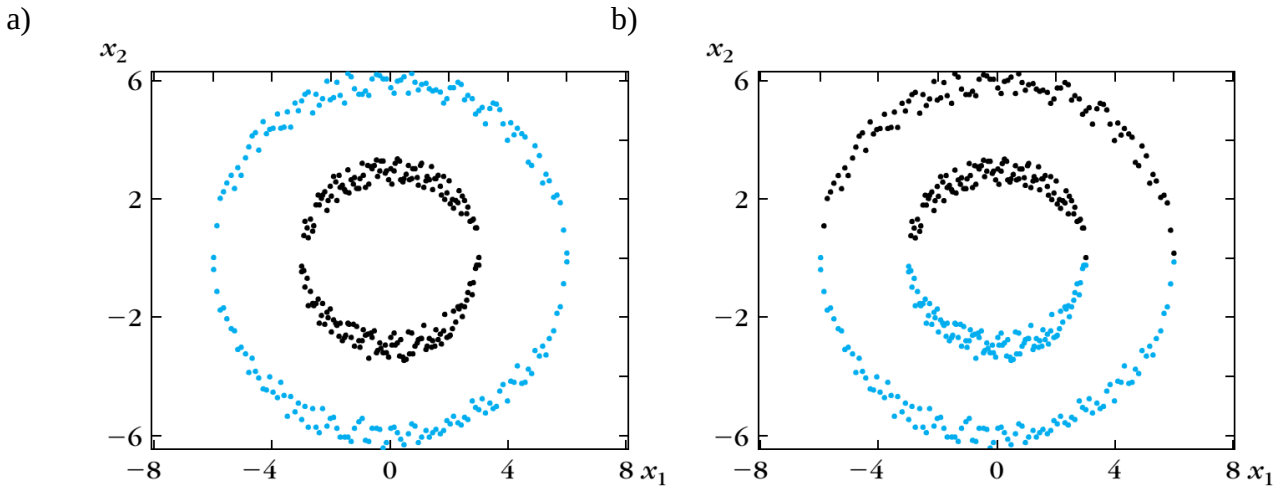


Figure 3.16 – Comparison between a) spectral clustering and b) k-means clustering applied to a dataset for which the data lie approximately over two concentric circles. Adapted from Theodoridis & Koutroumbas (2008).

3.4.3.5.4 METIS

METIS is a *multilevel* graph partitioning algorithm that is known to achieve high quality partitions across a wide range of applications, like VLSI circuit design, finite element methods, load balance in distributed computing, among others (Karypis & Kumar, 1995b; Karypis, 2013). We have included it here because it will be part of some of the cluster ensemble algorithms presented in Section 3.4.5.3 (Strehl & Ghosh, 2002), and we decided to evaluate it as a clustering algorithm too. Multilevel graph partitioning (Hendrickson & Leland, 1995) is a scheme in which a coarsened version of the graph is first partitioned, and then this partition is progressively extended and refined to finer levels of the graph back to its original size, as depicted in Figure 3.17. In this way, both global (the coarsened graph) and local (the refinement process) properties of the graph are addressed in the partitioning (Karypis & Kumar, 1998).

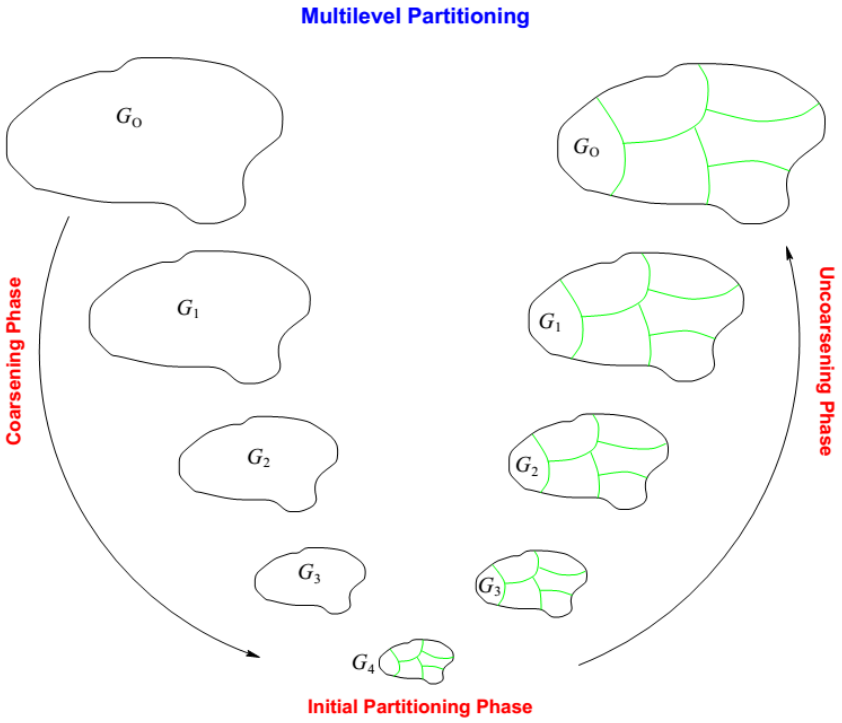


Figure 3.17 – General multilevel graph partitioning scheme. Extracted from Karypis (2013).

The multilevel scheme adopted by METIS comprises three phases: the coarsening phase, the initial partitioning phase and the uncoarsening phase (Figure 3.17). Algorithm 3.7 below pre-

sents the basic METIS procedure for data clustering. The details of each phase will be explained in the next sections.

Algorithm 3.7: METIS

Input: adjacency matrix $\boldsymbol{W}$, number of clusters *K*.
Output: clusters $= \{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\}$.

1 **begin**
2 $G_{coarse} \leftarrow$ coarsen the graph that has $\boldsymbol{W}$ as adjacency matrix, using one of the strategies from Section 3.4.3.5.5
3 $\{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\}_{coarse} \leftarrow$ partition G_{coarse} using one of the algorithms from Section 3.4.3.5.6
4 $\{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\} \leftarrow$ uncoarsen G_{coarse} and refine the partitions using one of the algorithms from Section 3.4.3.5.7
5 **return** $\boldsymbol{C} = \{\boldsymbol{C}_1, \dots, \boldsymbol{C}_K\}$
6 **end**

3.4.3.5.5 Coarsening phase

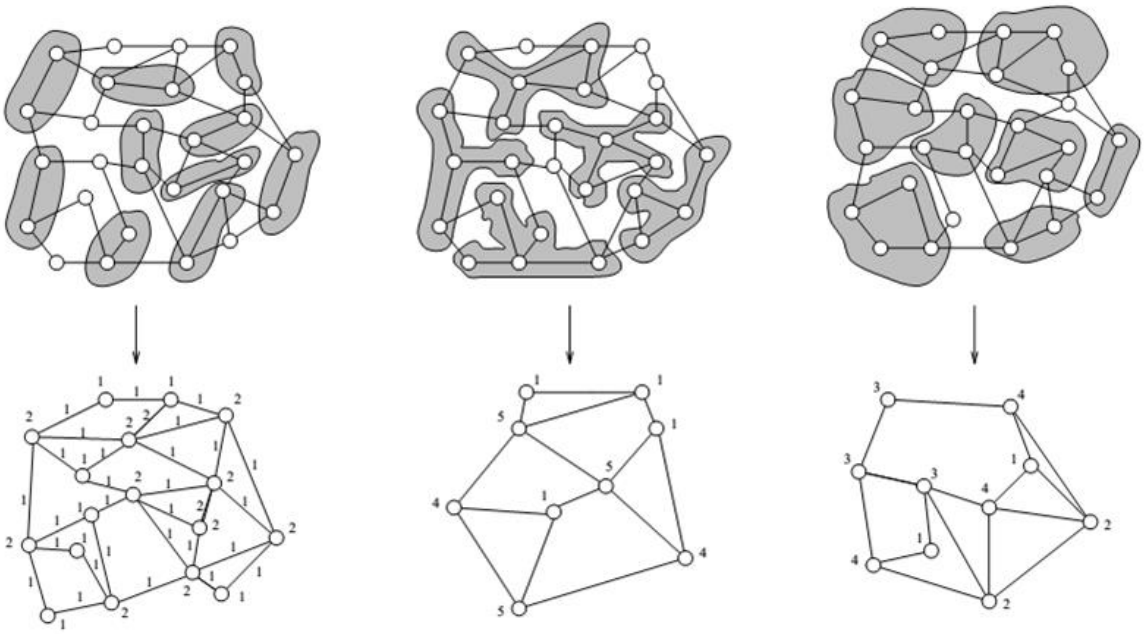


Figure 3.18 – Different ways of coarsening a graph. Extracted from Karypis & Kumar (1998).

The goal of coarsening is to obtain a reduced graph that reflects the properties of the original graph in terms of vertex and edge weights (Karypis & Kumar, 1995a). The central concept of graph coarsening is *matching*. To match nodes means combining adjacent nodes of *G* into a *multinode*, such that the weight of the multinode equals the sum of the weight of the matched vertices (if the vertices are weighted), and the edges connecting the multinode are the union of the edges of the matched vertices connecting external (multi)nodes. Different ways of matching nodes and

respective effects on vertice and edge weights are shown in Figure 3.18. METIS may use one of the following algorithms for matching nodes (Karypis & Kumar, 1995a):

- **Random Matching (RM)**

Vertices are visited randomly. If vertex has not been matched yet, it is randomly matched to one of its adjacent vertices.

- **Heavy Edge Matching (HEM)**

Vertices are visited randomly. If vertex has not been matched yet, it is matched to its most similar (heaviest edge) adjacent vertex.

- **Sorted Heavy Edge Matching (SHEM)**

Similar to HEM, but vertices are visited in ascending degree order. This reduces the occurrence of unmatched vertices on each iteration. Vertices with the same degree are visited in random order.

- **Light Edge Matching (LEM)**

Similar to HEM, but vertices are matched to their less similar adjacent vertex. It may be useful because it makes the average degree of the coarser graph much higher than that of the current graph.

- **Heavy Clique Matching (HCM)**

A *clique* is a fully connected subgraph of *G*. With HCM, vertices are visited randomly, and it matches a vertex to its adjacent vertex with the highest degree (i.e., it seeks to collapse cliques).

3.4.3.5.6 Partitioning the coarsest graph

Once the coarsest graph has been obtained by successive application of one of the algorithms presented in Section 3.4.3.5.5, a first partition shall be obtained. METIS uses the Min-cut as its goal function (Section 3.4.3.5.2). In order to produce meaningful, non-trivial cuts with Min-

cut, METIS requires the partitions to be unbalanced only by at most a user-specified percentage (Karypis, 2013). At this coarsest level, partitioning algorithms tend to run very fast because the size of the graph is small (~100 vertices). For this reason, often different initializations are attempted for the first partition, and that producing the lowest Min-cut is retained. METIS uses one of the following algorithms for partitioning the coarsest graph, constrained to the balancing requirements:

- **Spectral biSection (SB)**

Unnormalized spectral partitioning, as presented in Section 3.4.3.5.3.

- **Kernighan-Lin algorithm (KL)**

The KL algorithm begins with random partition assignments. It then searches for a pair of vertices that, if their partitions are swapped, it decreases the edge cut (Min-cut from Section 3.4.3.5.2) (Kernighan & Lin, 1970). The algorithm proceeds until no decrease in the edge cut is possible given the current state of the partitions. The success of the KL algorithm is dependent on the initialization and the average degree of the graph. The Fiduccia and Mattheyses (FM) algorithm is a modification of KL in which a single vertex swaps partitions, not a pair. The number of iterations can be limited for speed (Karypis & Kumar, 1998).

- **Graph Growing Partitioning algorithm (GGP)**

GGP selects a vertex at random. Then it grows a region around it in a breadth-first fashion, until $1/K$ of the vertices have been included (or $1/K$ of the total vertex weight). It grows more regions if $K > 2$. This partitioning is then provided as initialization to the KL algorithm (Karypis & Kumar, 1998).

- **Greedy Graph Growing Partitioning algorithm (GGGP)**

Similar to GGP, but vertices are included in the growing region in sorted order by their contribution to decreasing the cut (Karypis & Kumar, 1998).

3.4.3.5.7 Uncoarsening phase

The last phase of METIS is the uncoarsening and refinement phase. In this stage, the coarsening process is undone in the reverse order it was applied. At each level of uncoarsening, the partitions are projected from the coarser level to the finer one. Then, the edge cut is refined by application of one of the following algorithms:

- **KL refinement**

Uses projected partitions as initialization to the KL algorithm for refinement. The KL(1) version performs a single pass of KL across the list of nodes.

- **Boundary KL refinement (BKL)**

BKL is the KL refinement performed over only the vertices at the boundary of the partitions, as these are more likely to be swapped. BKL(1) performs a single pass of KL. BKL(*,1) performs BKL if graph is small (vertices on the partition boundaries are less than 2% of the number of vertices in the original graph), and BKL(1) if graph is larger than that (Karypis & Kumar, 1998).

3.4.3.6 Manifold learning and other approaches

Besides the specific algorithms presented in this Section (3.4.3), many other approaches have been proposed for unsupervised data classification. For example, there are combinations of different methods, like the popular Chameleon algorithm which performs hierarchical clustering by means of graph partitioning (Karypis, Han & Kumar, 1999), the DBSCAN algorithm which takes into account the density of data points (Ester, Kriegel, Sander & Xu, 1996), and evolutionary algorithms that seek to optimize partitioning goals (Hruschka *et al.*, 2009).

Data clustering has a close relationship to the more general task denoted *manifold learning*, also known as nonlinear dimensionality reduction (Theodoridis & Koutroumbas, 2008). Manifold learning algorithms seek to learn the low-dimensional subspace onto which the relevant information within high-dimensional data lie on (Bishop, 2006). Many of these algorithms have been formulated as non-linear extensions of PCA (Chang & Ghosh, 2001; Scholkopf *et al.*, 1996) and of the Kohonen SOM (Bishop *et al.*, 1998), as already commented in Sections 3.3.1 and

3.4.3.4, respectively. Autoassociative or autoencoder neural networks are also among the oldest and most useful tools for manifold learning. These networks are trained to mimic the input data in their output layer, using a reduced number of neurons in their intermediate layers (Bishop, 2006; Haykin, 1999). Only recently the use of manifold-oriented algorithms have been introduced in single particle analysis, and they delivered promising results for heterogeneity separation (Schwander *et al.*, 2010). Nevertheless, manifold learning algorithms may require estimating the parameters of complicated topology models, while similarity graphs, although also costly to build, may provide similar explanatory power for clustering without making assumptions about the data distribution (Gorban, Kégl, Wunsch & Zinovyev, 2007). On the other hand, manifold learning approaches provide generative models that may be more informative about the data behavior (Bishop *et al.*, 1998; Bishop, 2006). For specific details of manifold learning concepts and other popular algorithms like local linear embeddings (LLE), isometric mapping (ISOMAP) and Laplacian eigenmaps, the book edited by Gorban, Kégl, Wunsch & Zinovyev (2007) is an indicated reference.

3.4.4 Defining the number of clusters

The definition of the number of clusters K when performing unsupervised classification is sometimes misleading, as no absolute criteria exist for this task. Often, K is selected based on side information about the dataset or by domain-specific knowledge. Visualization of the data may also provide good clues with respect to the number of clusters. In single particle analysis, for example, it is quite unlikely to have more than a few dozens of different macromolecule views in 2D classification, or more than half a dozen stable, recognizable conformations in 3D classification.

When this kind of information is not available, however, there are a few methods that promote the determination of a reasonable value for K based on intrinsic information from the data. Essentially, the number of relevant clusters depends on the *scale* at which the data is being observed (Duda *et al.*, 2000). For methods that minimize a well-defined cost function, like hierarchical clustering (Section 3.4.3.1) or k-means (Section 3.4.3.2), the “elbow” or “kink” method is often employed. The elbow method consists on running the algorithm with different choices for K, and plotting the resulting cost (or distortion measure) as a function of K. If the algorithm is

initialization-dependent, it should be run many times for a given K, and the variability of the resulting cost must be taken into account; alternatively, only the lowest cost achieved for each value of K can be considered. Of course, the cost of the partitioning decreases as the number of groups increases; yet, the "natural" number of clusters can be found by the value at which the cost function ceases to drop significantly. The elbow method is principled on the fact that, for values of K smaller than the "ideal" value K^*, the true clusters will lie cohesively within the imposed clusters, and approaching K^* will decrease the value of the cost function by large amounts at each step. After K^* is reached, if K is kept increasing, the true clusters will be split across the imposed clusters, but this decreases the cost function only by a small amount at each step. Therefore, a "kink" in a plot like the one shown in Figure 3.19 indicates that the natural partitioning has been achieved. Hastie, Tibshirani & Friedman (2008) formalize the kink method with the *gap statistic*. An analogous method is available for spectral clustering algorithms, which consists in observing an abrupt rise in the plot of the eigenvalues (the spectrum) of the Laplacian matrix (von Luxburg, 2007). Another approach is to resort to *clustering indices* or information-theoretic model selection criteria (McLachlan & Peel, 2004; Theodoridis & Koutroumbas, 2008), as presented in Section 3.4.2.3. These measures regularize for the number of clusters (model complexity), and thus allow comparing the quality of solutions with different values for K.

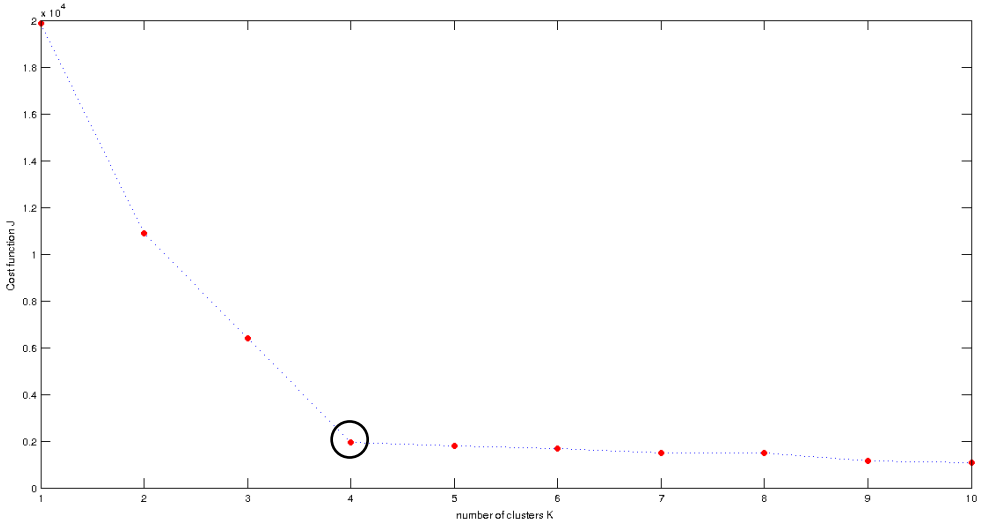


Figure 3.19 – Illustration of the "elbow method" for an artificial dataset composed of four real clusters. The k-means algorithm was run with different values for K, varying from 1 to 10. The value of the cost function (Equation 3.21) for each case was plotted as a function of K. Highlighted by the black circle is the "kink" or "elbow" in the plot, that is, the value of K for which the cost function begins to decrease smoothly. In this case, it perfectly matches the true number of clusters.

3.4.5 Ensemble methods

From this Section onwards, we will analyze unsupervised classification procedures that combine multiple solutions in order to produce the final result. In supervised learning, ensemble methods have been in use since the mid 1980's, in the form of *committee machines* or *mixtures of experts*, among others (Bishop, 2006; Haykin, 1999). The goal of ensemble methods is to achieve improved performance on a particular machine learning task by combining a set of independently trained models, like classifiers or regressors. For the clustering task, ensemble methods first appeared formally in the seminal work by Strehl & Ghosh (2002), on which this Section is mostly based.

In this classical formulation, cluster ensembles seek to obtain a solid partitioning solution *without* accessing the original features of the object set. That is, all the ensemble algorithms have access to is the set of labels from previous clusterings. Combining partitioning solutions like this is an interesting approach in several different scenarios (Ghosh & Acharya, 2011):

- **Improved solution quality**

Given a performance criterion, ensemble or "consensus" methods tend to perform better, on average, than an individual solution. This is because the ensemble tends to have *reduced bias* when compared to individual models, even when using simple approaches like ensemble averaging in supervised learning (Haykin, 1999).

- **Robust clustering**

Clustering algorithms are inevitably doomed to perform poorly on datasets which do not match their underlying assumptions. For example, k-means is not able to cope with arbitrarily shaped, non-convex clusters. An ensemble of clustering algorithms may then provide a "meta" clustering solution that achieves satisfactory performance over a wide range of datasets. This is also interesting from the point of view of the user, who, by using an ensemble, does not need to worry about parameter tuning of specific algorithms on specific datasets, or will worry less about it.

- **Knowledge reuse**

Consensus clustering may be used to consolidate data labels obtained from multiple previous projects or experiments. This may be useful for organizational and data storage purposes, or to provide a reasonable starting point for new projects or experiments that require an initial estimate for the solution.

- **“Multiview” clustering**

There are cases in which one must combine labeling solutions that have been obtained each with a different perspective on the dataset. For example, the clustering algorithms may have had access to different subsets of the dataset features (*feature-distributed clustering*) or to distinct subsets of objects (*object-distributed clustering*). These may happen for privacy or ownership constraints, like in internet databases, or even for logistic and computational resources constraints, when it is not possible to bring the whole dataset together into a single location.

The key aspect of ensemble performance is *diversity* among the base solutions. This is related to the individual model biases, which hopefully will be averaged out in the consensus solution. Diversity in clustering solutions may occur in different forms, like, for example (Kuncheva, 2004; Strehl & Ghosh, 2002):

- **Different subsets of features**

This source of diversity is inherent to multiview clustering cases, but may also be purposely introduced if needed. Subsets of features are related to the perspective an algorithm takes on the dataset. With images, for example, algorithms may cluster the dataset analyzing distinct regions of interest, or different bands of the Fourier spectrum (filtering). Also, other features like the intensity histogram and invariant attributes may be used concomitant to each other and to the pixels themselves.

- **Different number of clusters**

The cluster ensemble framework allows the integration of base solutions that have each different number of partitions. Varying *K* may also be deliberately applied in order to introduce diversity in the ensemble, or if one desires to use the consensus optimization approach for determining the number of clusters (Section 3.4.5.1).

- **Randomization**

Algorithms that use random initializations or random presentation of the data points, like online methods, may be run many times and then integrated into a single, stable, consensus solution. Randomization combined with multiview clustering may even be used for dimensionality reduction, by projecting the data onto random subspaces, clustering, and then looking for a consensus solution (Bingham & Mannila, 2001; Urruty, Djeraba & Simovici, 2007).

- **Algorithm portfolio**

A diverse ensemble may be attained by including distinct clustering algorithms. For example, one may include both compactness-based (k-means, HAC) as well as connectivity-based (graph partitioning) algorithms. Also, one may simply use the same algorithm(s) with distinct parameter setups, like different linkage criteria (Section 3.4.3.1.1) or different similarity measures, among other variations.

The performance of an ensemble of clustering algorithms over distinct datasets is illustrated in Figure 3.20. It can be clearly seen that a single algorithm may not perform well on the four datasets. And, what is more interesting is that the consensus solution achieves satisfactory performance on all four datasets, even when some individual solutions perform poorly. It must be observed that the quality of the consensus solution may not necessarily be better than that of an individual base solution (e.g. the YAHOO dataset, bottom row of Figure 3.20). However, it always provides solutions that are safely better than the worst individual performances. In practical scenarios, often it is not known beforehand which algorithms will have good performance on which datasets, thus justifying the ensemble approach.

For unsupervised classification of electron microscopy images of single particles, regarding the 3D heterogeneity of the dataset, we are most interested in achieving high performance and

robust classification for considerably noisy data. Particularly, we hope to use knowledge that may be already available in the single particle reconstruction workflow, like the principal components computed for dimensionality reduction, and to build a complex classification solution from a relatively simple algorithm portfolio, i.e., those that do not require the realization of 3D reconstructions.

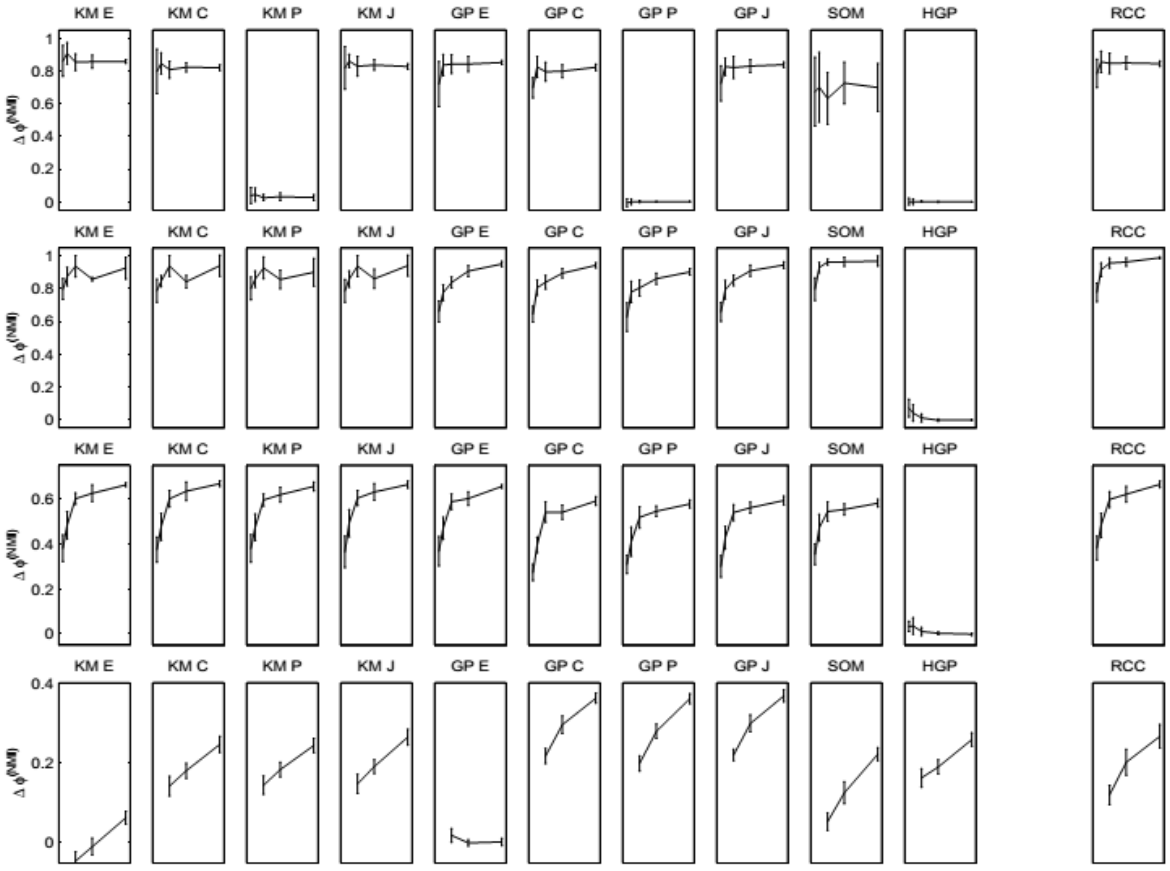


Figure 3.20 – Learning curves of the same cluster ensemble over four publicly available datasets: 2D2K (top row), 8D5K (second row), PENDIG (third row) and YAHOO (bottom row). A learning curve is a measure of performance as a function of the amount of data available. Here, performance is measured by the increase in Normalized Mutual Information (see Section 3.4.5.2) between the algorithm's solution and the ensemble, in comparison to a random labeling. Error bars indicate $\pm$ 1 standard deviations for 10 runs of each algorithm. The first 10 columns correspond to the clustering algorithms: k-means with Euclidean distance (KME); cosine similarity (KMC); correlation (KMP); Jaccard similarity (KMJ); graph partitioning with Euclidean distance (GPE); cosine similarity (GPC); correlation (GPP); Jaccard similarity (GPJ); self-organizing map (SOM), and hypergraph partitioning (HGP). The last column correspond to the robust consensus clustering (RCC), provided by the best of three consensus heuristics in each run: CSPA, HGPA and MCLA. See Section 3.4.5.3 for more details on these heuristics. Extracted from Strehl & Ghosh (2002).

3.4.5.1 Consensus clustering

A first approach in combining *R* base solutions is to perform *voting* to define the label of each object (Kuncheva, 2004). This method assumes that the base solutions follow a common convention and have the same number of partitions. However, achieving this "common convention", or solving the *cluster correspondence problem*, is not straightforward and has no perfect solution. A slightly more elaborate approach is to treat the $N \times R$ label matrix as a new representation of the data, and then achieve the final clustering by providing this matrix as input to some other clustering algorithm, like k-means. The rationale behind this strategy is that objects that are often clustered together are more likely to be similar, and thus should appear together in the final solution too. This method has the advantage that the number of groups may vary across the base clusterings, and must only be specified for the final solution. Nevertheless, this solution is still subject to the pitfalls of the algorithm chosen for the final clustering.

Strehl & Ghosh (2002) formulate the cluster ensemble problem as a combinatorial optimization problem. For simplicity, we will only present here the case of hard cluster assignments, and in which labels are known for all objects in every base solution. Let the set of *R* base solutions, possibly containing varying number of groups, be contained in the matrix $\boldsymbol{\Lambda}$, and $\phi(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b)$ be a measure of similarity or *agreement* between two label vectors, $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$. Then the average agreement between a labeling solution $\boldsymbol{\lambda}^*$ and a set of base solutions $\boldsymbol{\Lambda}$ is given by Equation 3.37:

$$\phi^{(avg)}(\boldsymbol{\Lambda}, \boldsymbol{\lambda}^*) = \frac{1}{R}\sum_{r=1}^{R} \phi(\boldsymbol{\lambda}^r, \boldsymbol{\lambda}^*) \tag{3.37}$$

Therefore, the cluster ensemble aims to find a single labeling $\boldsymbol{\lambda}^*$, with predefined number of clusters *K*, that maximizes $\phi^{(avg)}(\boldsymbol{\Lambda}, \boldsymbol{\lambda}^*)$. That is, the solution that best agrees, on average, with the set of available solutions:

$$\boldsymbol{\lambda}^{K-opt} = arg \max_{\boldsymbol{\lambda}^*} \phi^{(avg)}(\boldsymbol{\Lambda}, \boldsymbol{\lambda}^*) \tag{3.38}$$

Note that the optimization problem posed in Equation 3.38 may be solved for distinct values of *K*, and the one producing the highest value for $\phi^{(avg)}(\boldsymbol{\Lambda}, \boldsymbol{\lambda}^{K-opt})$ may be selected as the "best". Thus, the cluster ensemble approach also provides a model selection method, in addition to those mentioned in Section 3.4.4 (Ghosh, Strehl & Merugu, 2002; Strehl & Ghosh, 2002). This approach to the definition of the number of clusters is depicted in Figure 3.21.

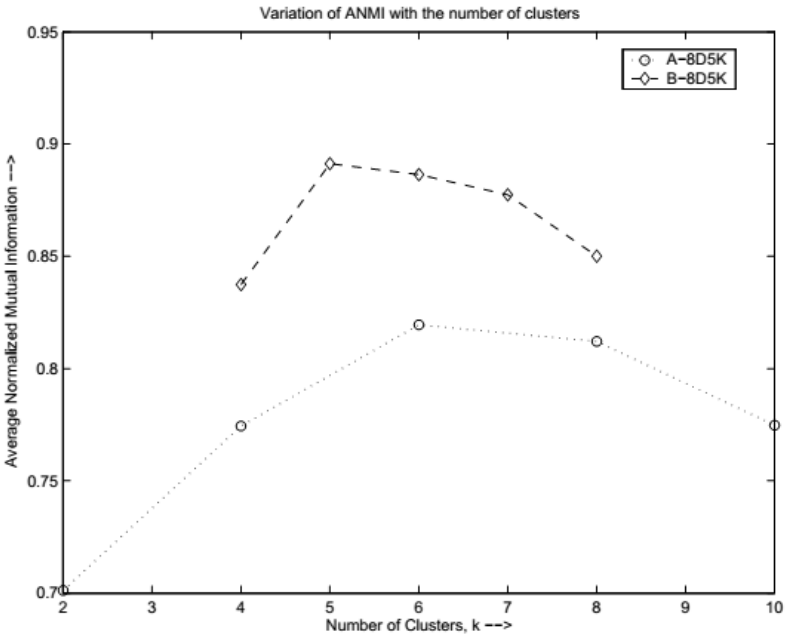


Figure 3.21 – Variation of ANMI, a measure of consensus agreement (see Section 3.4.5.2), as a function of *K*, for two distinct datasets. Extracted from Ghosh *et al.* (2002).

3.4.5.2 Comparing labeling solutions

Equation 3.37 requires that we somehow measure the agreement, or similarity, between label lists. Ideally, the result of this measure should lie on a pre-defined range and correct for the expected value, i.e., embed an "adjustment for chance" to compensate for random agreements (Acharya & Ghosh, 2013). We shall now present some common similarity indices for lists of labels. Interestingly, they do not require that the solutions being compared follow the same labeling conventions.

3.4.5.2.1 Adjusted Rand Index

The Adjusted Rand Index (ARI) was proposed by Hubert & Arabie (1985) and is a pair-counting based measure. It has the interesting property that it adjusts for cluster overlaps that may

occur by chance. If we have two labeling solutions, $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$, we define $n_{ij} = \left|\boldsymbol{C}_i^a \cap \boldsymbol{C}_j^b\right|$. The Adjusted Rand Index is then given by Equation 3.39:

$$\phi^{(ARI)}(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b) = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \frac{S_a S_b}{\binom{N}{2}}}{\frac{1}{2}(S_a + S_b) - \frac{S_a S_b}{\binom{N}{2}}} \tag{3.39}$$

where $S_a = \sum_i \binom{|\boldsymbol{C}_i^a|}{2}$ and $S_b = \sum_j \binom{|\boldsymbol{C}_j^b|}{2}$.

Although the ARI has a maximum value of 1 when the two label lists match perfectly, and a value of 0 when the index matches the expected value, it may produce negative results which are meaningless (Acharya & Ghosh, 2013).

3.4.5.2.2 Normalized Mutual Information

The Normalized Mutual Information (NMI) was proposed by Strehl & Ghosh (2002) and is an information-theoretic based measure, given by Equation 3.40:

$$\phi^{(NMI)}(\lambda^a, \lambda^b) = \frac{H(\lambda^a) + H(\lambda^b) - H(\lambda^a, \lambda^b)}{\sqrt{H(\lambda^a)H(\lambda^b)}} = \frac{I(\lambda^a, \lambda^b)}{\sqrt{H(\lambda^a)H(\lambda^b)}} \tag{3.40}$$

where $H(\boldsymbol{\lambda}^a)$ is the *entropy* of $\boldsymbol{\lambda}^a$, defined by Equation 3.41:

$$H(\boldsymbol{\lambda}^a) = -\sum_i \frac{|\boldsymbol{C}_i^a|}{N} \log\left(\frac{|\boldsymbol{C}_i^a|}{N}\right) \tag{3.41}$$

and $H(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b)$ is the mutual entropy of $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$, given by Equation 3.42:

$$H(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b) = -\sum_{i,j} \frac{n_{ij}}{N} log\left(\frac{n_{ij}}{N}\right) \tag{3.42}$$

The NMI as given in Equation 3.42 has the interesting property that it is constrained to the interval $[0, 1]$, allowing easy interpretation of the results. Note that $H(\boldsymbol{\lambda}^a)$ and $H(\boldsymbol{\lambda}^b)$ are the entropies of lists $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$, respectively, and $H(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b)$ is the mutual entropy between them, as defined previously in Equation 3.20. $I(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b)$ is denoted the *mutual information* between $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$ (Shannon, 1948). When the NMI is used in the cluster ensemble optimization problem, Equation 3.37 is called the *Average NMI* (ANMI), and will be the default choice for the rest of this explanation, following Strehl & Ghosh (2002), unless otherwise noted.

3.4.5.2.3 Normalized Variation of Information

The Normalized Variation of Information (NVI) is also an information-theoretic based measure (Xiong, Wu & Chen, 2009), defined in Equation 3.43:

$$\phi^{(NVI)}(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b) = 1 - \frac{2I(\boldsymbol{\lambda}^a, \boldsymbol{\lambda}^b)}{H(\boldsymbol{\lambda}^a) + H(\boldsymbol{\lambda}^b)} \tag{3.43}$$

NVI was proposed in order to normalize the original Variation of Information (VI) measure (Meilă, 2003), which was restricted to label lists of the same size and having the same number of partitions. Please note that the VI and NVI are metrics, while the NMI is not (Ghosh & Acharya, 2011).

3.4.5.3 Algorithms

Directly solving the consensus clustering problem defined in Equation 3.38 is a difficult combinatorial optimization problem. As pointed out by Strehl & Ghosh (2002), even for a very small dataset containing 16 objects, there are 171,798,901 manners of grouping them into mere four clusters. They then proposed a greedy search algorithm to optimize Equation 3.38. Kuncheva (2004) proposes a randomized version of this algorithm that is more likely to achieve a good local maximum, which is depicted in Algorithm 3.8.

Algorithm 3.8: Greedy Consensus Clustering

Input: $N \times R$ matrix $\boldsymbol{\Lambda}$ of base clustering solutions; number of clusters K.
Output: consensus solution $\boldsymbol{\lambda}^*$.

1 **begin**
2 $r_{init} \leftarrow arg\ \max_r \phi^{(avg)}(\boldsymbol{\Lambda}, \boldsymbol{\lambda}^r)$
3 $\boldsymbol{\lambda}^{(0)} \leftarrow \boldsymbol{\lambda}^{r_{init}}$
4 $i \leftarrow 0$
5 **do**
6 **for** $n = 1, \ldots, N$ (in random order)
7 $\lambda_n^{(i)} \leftarrow arg\ max_k\ \phi^{(avg)}\left(\boldsymbol{\Lambda}, \boldsymbol{\lambda}^{(i)}: \lambda_n^{(i)} = k\right),\ k = 1, \ldots, K$
8 **end**
9 $i \leftarrow i + 1$
10 **until** $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{(i-1)}$
11 **return** $\boldsymbol{\lambda}^{(i)}$
12 **end**

Algorithm 3.8 begins by selecting the available solution that best agrees with the other solutions, on average (line 2). Then, it swaps the cluster assignment for all λ_n, one at a time and in random order, to the one of the other $K - 1$ possible labels that maximizes the average agreement (line 8). After all λ_n have been swept, the order of the list is again randomized and the search restarts. The algorithm proceeds like this until no labels have been changed, which means that a local optimum has been achieved. However, the computational complexity of this approach makes it impractical for large datasets.

Strehl & Ghosh (2002) proposed more efficient heuristics to achieve a consensus solution, which are based on a *hypergraph* representation. A hypergraph is a generalization of the graph representation, in which an edge may connect more than two vertices. To illustrate this concept and its derivate heuristics, let's consider a simple example containing seven objects and four distinct clustering solutions, as in Table 3.2 (Strehl & Ghosh, 2002). Note that the base solutions are diverse in that they do not follow the same convention (cluster correspondence) in terms of labels, do not necessarily contain the same number of clusters, and do not necessarily contain assignments to all objects.

Table 3.2 – An example $\boldsymbol{\Lambda}$ matrix containing seven objects and four labeling solutions. Note that solution $\boldsymbol{\lambda}^4$ has less clusters than the other solutions, and does not contain assignments to all objects. Also, the labeling convention varies across solutions. Extracted from (Strehl & Ghosh, 2002).

$\boldsymbol{\Lambda}$	$\boldsymbol{\lambda}^1$	$\boldsymbol{\lambda}^2$	$\boldsymbol{\lambda}^3$	$\boldsymbol{\lambda}^4$
$\boldsymbol{x}_1$	1	2	1	1
$\boldsymbol{x}_2$	1	2	1	2
$\boldsymbol{x}_3$	1	2	2	?
$\boldsymbol{x}_4$	2	3	2	1
$\boldsymbol{x}_5$	2	3	3	2
$\boldsymbol{x}_6$	3	1	3	?
$\boldsymbol{x}_7$	3	1	3	?

The base label matrix $\boldsymbol{\Lambda}$ can be converted into a binary representation scheme, matrix $\boldsymbol{H}$, shown in Table 3.3. $\boldsymbol{H}$ is composed of submatrices $\{\boldsymbol{H}^1, \ldots, \boldsymbol{H}^R\}$, each corresponding to one of the base clusterings. Note that this form of representation avoids the cluster correspondence problem. Also, a careful analysis of this matrix indicates that it may be possible to infer the cluster assignment of objects that do not have labels in all base solutions, like objects $\boldsymbol{x}_3$, $\boldsymbol{x}_6$ and $\boldsymbol{x}_7$. More importantly, this matrix can be seen as the adjacency matrix of a hypergraph, in which the objects are the vertices, and each cluster is a hyperedge connecting its member objects, which are those that have been assigned a value of "1" in the matrix. The four heuristics that follow, CSPA (Section 3.4.5.3.1), HGPA (Section 3.4.5.3.2), MCLA (Section 3.4.5.3.3) and HGBF (Section 3.4.5.3.4) all benefit from this hypergraph representation in different ways.

Table 3.3 – The $\boldsymbol{H}$ matrix: binary hypergraph representation of matrix $\boldsymbol{\Lambda}$ (Table 3.2). Extracted from (Strehl & Ghosh, 2002).

	$\boldsymbol{H}^1$			$\boldsymbol{H}^2$			$\boldsymbol{H}^3$			$\boldsymbol{H}^4$	
$\boldsymbol{H}$	$\boldsymbol{h}_1$	$\boldsymbol{h}_2$	$\boldsymbol{h}_3$	$\boldsymbol{h}_4$	$\boldsymbol{h}_5$	$\boldsymbol{h}_6$	$\boldsymbol{h}_7$	$\boldsymbol{h}_8$	$\boldsymbol{h}_9$	$\boldsymbol{h}_{10}$	$\boldsymbol{h}_{11}$
v_1	1	0	0	1	0	0	1	0	0	1	0
v_2	1	0	0	1	0	0	1	0	0	0	1
v_3	1	0	0	1	0	0	0	1	0	0	0
v_4	0	1	0	0	1	0	0	1	0	1	0
v_5	0	1	0	0	1	0	0	0	1	0	1
v_6	0	0	1	0	0	1	0	0	1	0	0
v_7	0	0	1	0	0	1	0	0	1	0	0

3.4.5.3.1 Cluster-based Similarity Partition Algorithm

The *Cluster-based Similarity Partition Algorithm* (CSPA) uses matrix $\boldsymbol{H}$ (Table 3.3) to build a real-valued coassociation matrix $\boldsymbol{S}$. Hence, an induced similarity measure is obtained from the fact that more similar objects tend to be more often clustered together. The $N \times N$ entries of matrix $\boldsymbol{S}$ lie in the interval $[0, 1]$, where a value of 1 indicates that objects *i* and *j* always appear together, and 0 means they are never assigned to the same group, among the base clusterings. $\boldsymbol{S}$ is obtained by a simple matrix multiplication with $\boldsymbol{H}$, as shown in Equation 3.44.

$$\boldsymbol{S} = \frac{1}{R}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} \tag{3.44}$$

Table 3.4 shows the coassociation matrix for the example from Table 3.2, in which $R = 4$. The constitution of $\boldsymbol{S}$ for this example is also shown graphically in Figure 3.22.

Table 3.4 – The coassociation matrix $\boldsymbol{S}$ induced from Table 3.3.

$\boldsymbol{S}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	0.75	0.50	0.25	0	0	0
v_2	0.75	1	0.50	0	0.25	0	0
v_3	0.50	0.50	1	0.25	0	0	0
v_4	0.25	0	0.25	1	0.50	0	0
v_5	0	0.25	0	0.50	1	0.25	0.25
v_6	0	0	0	0	0.25	1	0.75
v_7	0	0	0	0	0.25	0.75	1

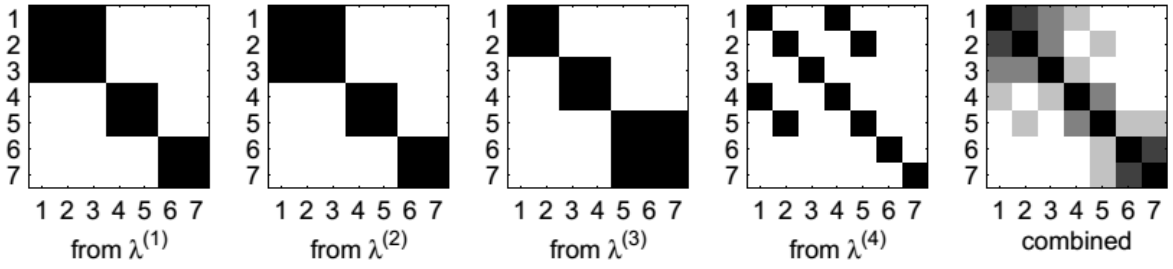


Figure 3.22 – Visualization of the binary coassociation matrices $\{\boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3, \boldsymbol{H}^4\}$ and the weighted coassociation matrix $\boldsymbol{S}$ (rightmost) corresponding to Table 3.4. Extracted from Strehl & Ghosh (2002).

After the coassociation matrix is formed, it can be interpreted as the adjacency matrix of a weighted, undirected, similarity graph. Such graph is illustrated in Figure 3.23. Then, a graph partitioning algorithm can be applied to obtain the consensus solution with *K* clusters. Strehl & Ghosh (2002) use METIS (Section 3.4.3.5.4) due to its scalability and high-quality partitioning solutions for many types of graphs (Karypis & Kumar, 1998). CSPA is perhaps the simplest "smart" heuristic and is able to achieve good quality solutions (Ghosh *et al.*, 2002; Strehl & Ghosh, 2002). However, its computational complexity is proportional to N^2 both in time and memory, which renders it impractical for very large datasets. Considering worst case time complexity alone, CSPA is $\mathcal{O}(N^2KR)$, meaning its running time scales proportionally to N^2KR. Punera & Ghosh (2008) proposed sCSPA, the extension of CSPA to soft clusterings (see Section 3.4.2.1). The extension is quite straightforward, by replacing the binary cluster indicators in matrix $\boldsymbol{H}$ (Table 3.3) by the corresponding fuzzy (or probabilistic) coefficients.

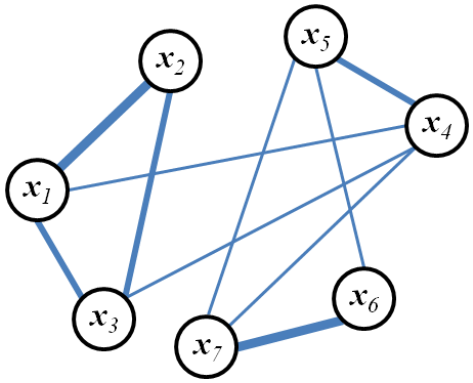


Figure 3.23 – The induced similarity graph to be partitioned in CSPA, corresponding to the matrix ***S*** (Table 3.4) of the example from (Strehl & Ghosh, 2002) worked out in this Section. Thickness of edges indicate their relative weights (self-edges not shown).

3.4.5.3.2 HyperGraph Partitioning Algorithm

The *HyperGraph Partitioning Algorithm* (HGPA) is quite similar to CSPA, but instead of using the induced similarity matrix, it uses the binary matrix ***H*** directly (Table 3.3). HGPA therefore formulates the cluster ensemble problem as a *hypergraph* partitioning task, as illustrated in Figure 3.24. The partitioning is performed by cutting the *minimal* number of hyperedges that leaves the graph composed of *K* disjoint partitions. To handle this task, Strehl & Ghosh (2002) use HMETIS (Karypis, Aggarwal, Kumar & Shekhar, 1997), the extension of METIS to hypergraphs.

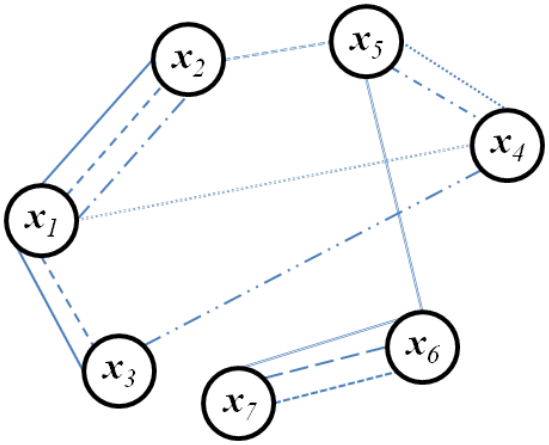


Figure 3.24 - The hypergraph to be partitioned in HGPA, corresponding to the matrix ***H*** (Table 3.3) of the example from (Strehl & Ghosh, 2002) worked out in this Section. The eleven types of lines indicate the eleven hyperedges connecting objects.

HGPA has the advantage that it is the fastest of the three heuristics proposed by Strehl & Ghosh (2002), with a worst-case time complexity of $\mathcal{O}(NKR)$. A disadvantage is that practical hypergraph partitioners consider only the removal of entire hyperedges, possibly achieving solutions worse than if only partially removing hyperedges was allowed. Because of this, two or more radically different consensus solutions will then have equivalent quality from the hypergraph partitioning point of view (Strehl & Ghosh, 2002).

3.4.5.3.3 Meta-CLustering Algorithm

The *Meta-CLustering Algorithm* (MCLA) is the most elaborate of the three consensus heuristics proposed by Strehl & Ghosh (2002). In a certain sense, it uses the transpose of the binary coassociation matrix $\boldsymbol{H}$ from Table 3.3. In the MCLA formulation, a "meta-graph" is formed having each clustering $\boldsymbol{h}_r$ as a vertex. The edges are weighted by the Jaccard similarity measure, given in Equation 3.45. The cluster similarity matrix of this "meta-graph" is exemplified in Table 3.5, and the respective visualization is shown in Figure 3.25.

$$J(\boldsymbol{C}_i, \boldsymbol{C}_j) = \frac{|\boldsymbol{C}_i \cap \boldsymbol{C}_j|}{|\boldsymbol{C}_i \cup \boldsymbol{C}_j|} = \frac{\boldsymbol{h}_i^{\mathrm{T}} \boldsymbol{h}_j}{\|\boldsymbol{h}_i\|_2^2 + \|\boldsymbol{h}_j\|_2^2 - \boldsymbol{h}_i^T \boldsymbol{h}_j} \tag{3.45}$$

This cluster similarity graph is then *k*-way partitioned by METIS. In the next step, the clusters allocated to each partition are then merged, giving place to *meta-clusters*. A meta-cluster may contain each object $\boldsymbol{x}_n$ many times, depending on how often the object appears in the collapsed clusters. At the final stage of the algorithm, the meta-clusters compete for objects. This means that an object is definitely allocated to the meta-cluster in which it appears most often, as if the meta-clusters were bidding for the object. In case of ties, the object is randomly allocated to one of the meta-clusters in which it appears equally often. A very interesting byproduct of this competition for objects is that a confidence measure is provided by the relative occurrence of the objects on each meta-cluster.

Table 3.5 – Example of the cluster similarity matrix for the meta-graph used by MCLA. Entries are the Jaccard similarity between hyperedges in matrix $\boldsymbol{H}$ from Table 3.3.

	$\boldsymbol{h}_1$	$\boldsymbol{h}_2$	$\boldsymbol{h}_3$	$\boldsymbol{h}_4$	$\boldsymbol{h}_5$	$\boldsymbol{h}_6$	$\boldsymbol{h}_7$	$\boldsymbol{h}_8$	$\boldsymbol{h}_9$	$\boldsymbol{h}_{10}$	$\boldsymbol{h}_{11}$
$\boldsymbol{h}_1$	1	0	0	1	0	0	0.67	0	0	0.25	0
$\boldsymbol{h}_2$	0	1	0	0	1	0	0	0.33	0.25	0.33	0.33
$\boldsymbol{h}_3$	0	0	1	0	0	1	0	0	0.67	0	0
$\boldsymbol{h}_4$	1	0	0	1	0	0	0.67	0.25	0	0.25	0.25
$\boldsymbol{h}_5$	0	1	0	0	1	0	0	0.33	0.25	0.33	0.33
$\boldsymbol{h}_6$	0	0	1	0	0	1	0	0	0.67	0	0
$\boldsymbol{h}_7$	0.67	0	0	0.67	0	0	1	0	0	0.33	0.33
$\boldsymbol{h}_8$	0	0.33	0	0.25	0.33	0	0	1	0	0.33	0
$\boldsymbol{h}_9$	0	0.25	0.67	0	0.25	0.67	0	0	1	0	0.25
$\boldsymbol{h}_{10}$	0.25	0.33	0	0.25	0.33	0	0.33	0.33	0	1	0
$\boldsymbol{h}_{11}$	0	0.33	0	0.25	0.33	0	0.33	0	0.25	0	1

MCLA has a worst-case complexity in time of $\mathcal{O}(NK^2R^2)$. Assuming $K, R \ll N$, it tends to be almost as fast as HGPA (Section 3.4.5.3.2), while producing good quality solutions (Ghosh *et al.*, 2002; Strehl & Ghosh, 2002). However, if the base clusterings are very "diverse", it tends to perform worse than CSPA and HGPA (Strehl & Ghosh, 2002). This is because MCLA implicitly assumes that there are notable correlations (or correspondences) between clusters. Punera & Ghosh (2008) extended MCLA to soft clusterings with sMCLA. In their formulation, the hyperedges, instead of binary vectors, are converted into feature vectors having the soft assignments as features. A pairwise similarity between them is then computed based on the Euclidean distance; the remainder of the algorithm is the same.

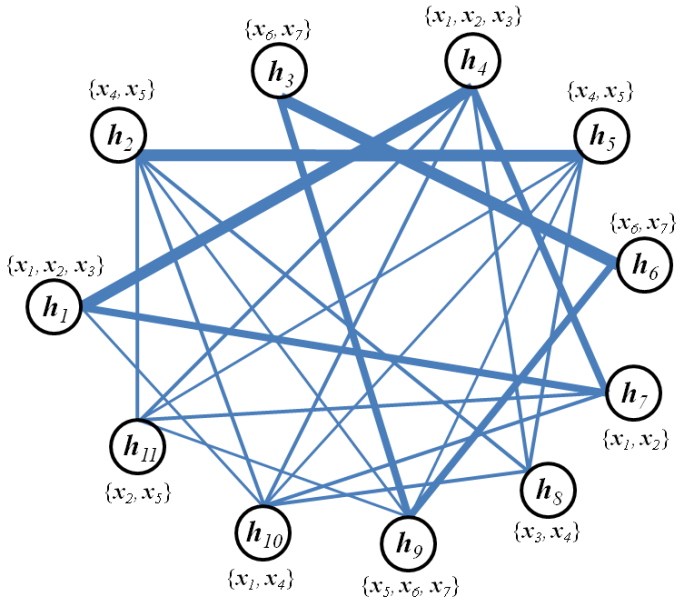


Figure 3.25 – The meta-graph used by MCLA, corresponding to the cluster similarity matrix from Table 3.5 of the example from Strehl & Ghosh (2002) worked out in this Section. Hyperedges are the vertices of this graph, and the edges correspond to the Jaccard similarity (Equation 3.45) between them. Edge thickness corresponds to its relative weight. Next to each vertex is the set of objects associated with the hyperedge.

3.4.5.3.4 Hybrid Bipartite Graph Formulation

A fourth consensus heuristic based on (hyper)graphs is the *Hybrid Bipartite Graph Formulation* (HGBF) proposed by Fern & Brodley (2004). This approach considers both objects and base clusters as vertices of a bipartite graph, and they are linked by unweighted edges simply whenever object $\boldsymbol{x}_n$ belongs to cluster $\boldsymbol{C}_i^r, r \in \{1, \ldots, R\}, i \in \{1, \ldots, K_r\}$; $\boldsymbol{C}_i^r$ is therefore a hyperedge in matrix $\boldsymbol{H}$ from Table 3.3. A bipartite graph is one in which its vertices belong to two disjoint partitions, such that all the edges of the graph connect vertices in different partitions. An example of such bipartite graph is given in Figure 3.26. After the graph is formed, it can be partitioned using METIS or spectral clustering (Fern & Brodley, 2004).

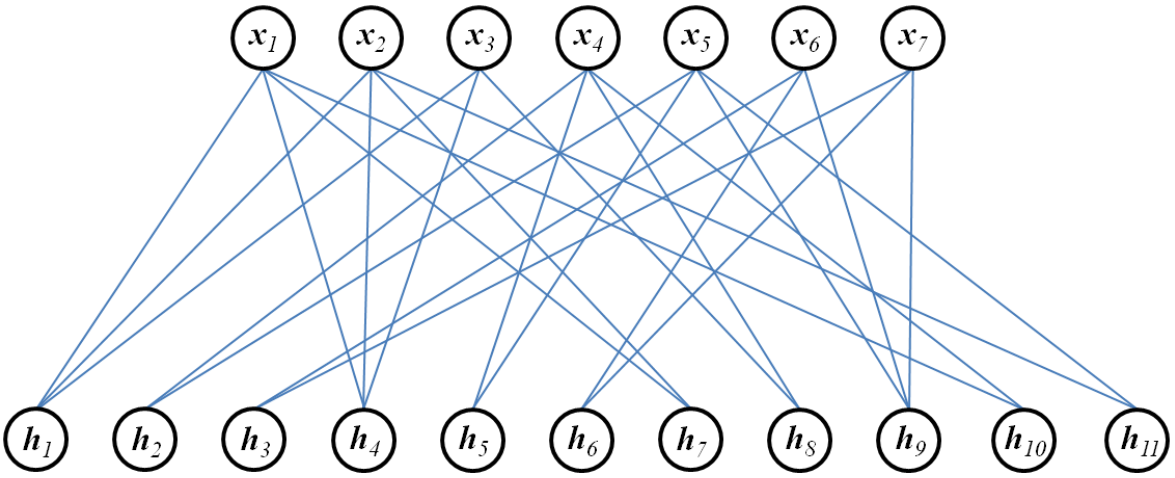


Figure 3.26 – Example of bipartite graph used by HBGF. This is the graph corresponding to the example from (Strehl & Ghosh, 2002) worked out in this Section, derived from the matrix $\boldsymbol{H}$ (Table 3.3).

Fern & Brodley (2004) argue that this representation has two main advantages. First, it is a lossless representation, which means that the original set of base clusterings can be fully recovered from the bipartite graph. From the three heuristics presented previously, only HGPA has this property. The other advantage is that HGBF incorporates both similarities between data points and similarities between clusters. This circumvents flaws that occur in formulations that take into account only one of these two types of information. For example, CSPA may treat as being barely similar points that are rarely clustered together, although them both may be similar to other points that often appear together. An analogous problem may occur with MCLA: two clusters containing disjoint sets of objects (i.e., there is no overlap between them) will be regarded as totally dissimilar even though the union of their member objects may often appear together in other clusters. Also, it is not subject to the hypergraph partitioning limitations suffered by HGPA (see Section 3.4.5.3.2). Solving HBGF has worst-case complexity of $\mathcal{O}(NKR)$ in time. Punera & Ghosh (2008) also formulated a version of HBGF for soft clustering, sHBGF. It simply weights the edges of the bipartite graph by the corresponding value of the soft assignment.

3.4.5.4 Remarks on cluster ensembles

We observe that the formulation of the cluster ensemble problem as a graph partitioning task may suffer from the balancing constraints commonly employed in (hyper)graph partitioning algorithms when the base clusterings are highly unbalanced. Strehl & Ghosh (2002) employed (H)METIS for the CSPA, HGPA and MCLA heuristics, but the balancing constraints also exist

and may be even more severe in other graph partitioning approaches, like spectral clustering (Section 3.4.3.5.3). Regarding the algorithms for ensembles of soft clusterings proposed by Punera & Ghosh (2008), it is important to notice that, although the base clusterings are of the soft type, the output consensus are hard assignments.

Although we have concentrated on (hyper)graph models for cluster ensembles due to their acknowledged quality and feature-independent representations, these are not the only approaches available. We point out that more advanced approaches based on cumulative voting exist (Dudoit & Fridlyand, 2003), and that using the set of labels from the base clusterings as a new representation of the data allows not only using a conventional clustering algorithm to obtain consensus, as pointed out in Section 3.4.5.1, but also to formulate probabilistic cluster ensembles. These include a mixture model formulation for consensus clustering (Topchy, Jain & Punch, 2004), adaptive cluster ensembles (Topchy, Minaei-Bidgoli, Jain & Punch, 2004) and Bayesian cluster ensembles (Wang, Shan & Banerjee, 2011). These methods use the Expectation-Maximization algorithm from Section 3.4.3.3 to improve the log-likelihood of the consensus solution. There are also cluster ensembles proposals that count on access to the original set of features of the dataset (Domeniconi & Al-Razgan, 2009). For an overview of the cluster ensemble problem and its associated applications and algorithms, please refer to the book by Kuncheva (2004) and the reviews by Acharya & Ghosh (2011; 2013).

4. Classification of Heterogeneous Cryo-EM Data

This Chapter will present our proposal for unsupervised classification of structural heterogeneity on cryo-EM data. The structural heterogeneity problem has been introduced in Section 2.5. The data clustering concepts and algorithms outlined in Chapter 3 will now be brought together in a framework that aims to discriminate conformational states without performing 3D reconstructions, thus being useful to validate the 3D classification performed by conventional single particle reconstruction (SPR) methods. The description of our unsupervised classification scheme will follow that depicted in Figure 1.2. Finally, we will present details about the data chosen for the tests of our proposal. The specific implementation details and parameter choices are depicted in Chapter 5, and the results of our study are presented in Chapter 6.

4.1 Data collection

After a potentially heterogeneous sample is prepared following the proper biochemical protocols (Frank, 2006), a series of micrographs are collected in the transmission electron microscope (TEM). We will emphasize the analysis of cryo-EM datasets because sample preservation by negative stain usually does not allow the observation of conformational differences (Section 2.2.1). During the process of CTF correction (Section 2.1.1), some micrographs may be discarded due to insufficient quality. For the initial analyses and reconstructions, it may be desirable to coarsen the micrographs by a factor of 2 or even 4, for computational speedup. From the remaining micrographs, the particles are windowed using manual or semi-automated picking procedures (Section 2.4.1). The stack of boxed particles constitutes the set of projection images that will be further analyzed and classified.

4.2 Data pre-processing

Common pre-processing steps include normalizing the projection images to zero mean and the same variance. Also, the images must be masked by a circular disk for removal of background information that is of no interest. The mask may be binary or real-valued depending on whether they have a hard or soft edge. Soft-edged masks are required if operations on Fourier space will be conducted, as they do not introduce high-frequency artifacts (Section 2.4.2). Images may be band-pass filtered for suppression of noise and low-frequency artifacts. We note that filtering is a crucial step for our analysis as it directly limits the magnitude of structural flexibility that remains observable, but the specific filtering parameters will largely depend on the type of molecule under investigation and the image acquisition conditions. We will consider both the more general case when the images are just corrected for translational misalignments and the more advanced case when the images are also rotationally aligned. Beyond these standard normalization and alignment procedures, other feature extraction (Section 3.2) and dimensionality reduction (Section 3.3) operations may be necessary to achieve success on the task at hand. In our experiments, Principal Component Analysis (Section 3.3.1) will be used to compress the datasets.

4.3 Unsupervised classifiers

After the feature vectors have been obtained for each projection image, they will be classified in an unsupervised fashion by clustering algorithms, independently of the specific reconstruction procedure that may be in course. We focus here on the problem of validating heterogeneous reconstructions (Henderson *et al.*, 2012), so we will assume that the number of conformational states expected is already known. No assumptions are made about the specific distribution each conformational manifold may impose on the data, if any. Therefore, we will use clustering algorithms with different underlying motivations: Hierarchical Ascendant Clustering (HAC), k-means, Self-Organizing Maps (SOM), Gaussian Mixture Models (GMM), Spectral Clustering and METIS. HAC (van Heel *et al.*, 2009), k-means (P. A. Penczek *et al.*, 1996), SOM (Marabini & Carazo, 1994) and spectral clustering (Shatsky *et al.*, 2010; Ueno *et al.*, 2005) have already been employed for 2D and 3D classification of single particles data. Gaussian mixtures have al-

ready been used within the maximum-likelihood structural refinement context (Scheres & Carazo, 2009), while we will use it just for unsupervised classification. METIS was chosen because it was shown to have impressive graph partitioning performance on a wide range of domain applications, and it was readily available, as it is part of some consensus heuristics whose performance we will also investigate (see Section 4.4). We also note that, in general, the most computationally expensive step in using graph partitioning algorithms is to build the adjacency matrix. Adjacency matrices are already going to be generated for spectral clustering, so they can also be provided as input to METIS. These algorithms were presented in Section 3.4.4. We will assess the performance of clustering algorithms for heterogeneity classification both individually and as a cluster ensemble.

4.4 Consensus clustering

After a set of labels is provided by the clustering algorithms, we explored consensus clustering approaches by means of an ensemble. The potential advantages of using ensemble clustering in this application are many, as explained in Chapter 3. We are mainly interested in constructing a complex classification solution, regarding the structural heterogeneity of the data, from a set of relatively simple clustering algorithms. We are also interested in robust classification for noisy data, as we make no prior assumptions about which kind of algorithm may be best suited for a given dataset.

The first consensus method we attempted was a basic majority voting scheme. Later, we introduced the three heuristics proposed by Strehl & Ghosh (2002) to efficiently solve the cluster ensembles problem: CSPA, HGPA and MCLA. We also investigated the use of the k-means algorithm as a simple method for obtaining a consolidated clustering solution by using the base set of labels as a new representation for the dataset (Section 3.4.5.1), and compared it with the other approaches.

As we will use data that has already been labeled by conventional reconstruction methods, they will serve as the “ground truth” for the evaluation of our unsupervised classifiers. The partitioning validation procedure consists of verifying how well the structural assignments provided by the iterative reconstruction matches our unsupervised classification framework. Nevertheless,

we also attempted to use cluster ensembles internal agreement to determine the number of structural classes (Section 3.4.5.1).

4.5 The dataset

The dataset on which we have concentrated the tests of our clustering framework consists of a mixture of projection images from the Mm-cpn protein in its "open" and "closed" states. This structure has a molecular weight of ~1 MDa and D8 symmetry, which means its subunits are repeated along two perpendicular axes, one of two-fold symmetry and the other of eight-fold symmetry. Mm-cpn is a group II chaperonin, from the archaea *Methanococcus maripaludis* organism. Chaperonins are macromolecular machines that aid the folding of cellular proteins in eukaryotes and archaea. Mm-cpn is a barrel-like structure that accommodates polypeptide chains in its central cavity, as shown in Figure 4.1. With its lids closed, energy is provided to the contained polypeptide substrate (a protein) by adenosine triphosphate (ATP) induction. When the protein is released with this added energy, it may potentially reach a stable folding state, akin to an annealing procedure. The structure in both conformations has been resolved by electron cryomicroscopy, albeit from separate samples. Details about Mm-cpn role in the cell and its structure determination can be found in the work by Zhang *et al.* (2010). The structure in the "open" state was determined by single particle reconstruction at 8 Å, while a resolution of 4.3 Å was achieved for the structure in the "closed" state. This discrepancy is mainly attributed to the flexibility of the lids in the open conformation. The distinct states were biochemically induced, and then imaged and reconstructed separately. While we acknowledge that this case does not represent the more general situation in which the sample contains a mixture of conformations, it still allow us to mixture *in silico* the images collected from each sample in a single dataset, and perform our analysis with a perfect standard for comparison. The downside of deliberately separating conformational states prior to TEM imaging is that no information about intermediate states can be retrieved. Also, the striking different structural configurations assumed in the two states, notably by the closing of the arm lids, make it suitable for the investigation of how variations in structure conformation affect the data distribution in the feature space used for classification. We have tested our approach both on synthetic and real Mm-cpn models, as detailed in Chapter 5.

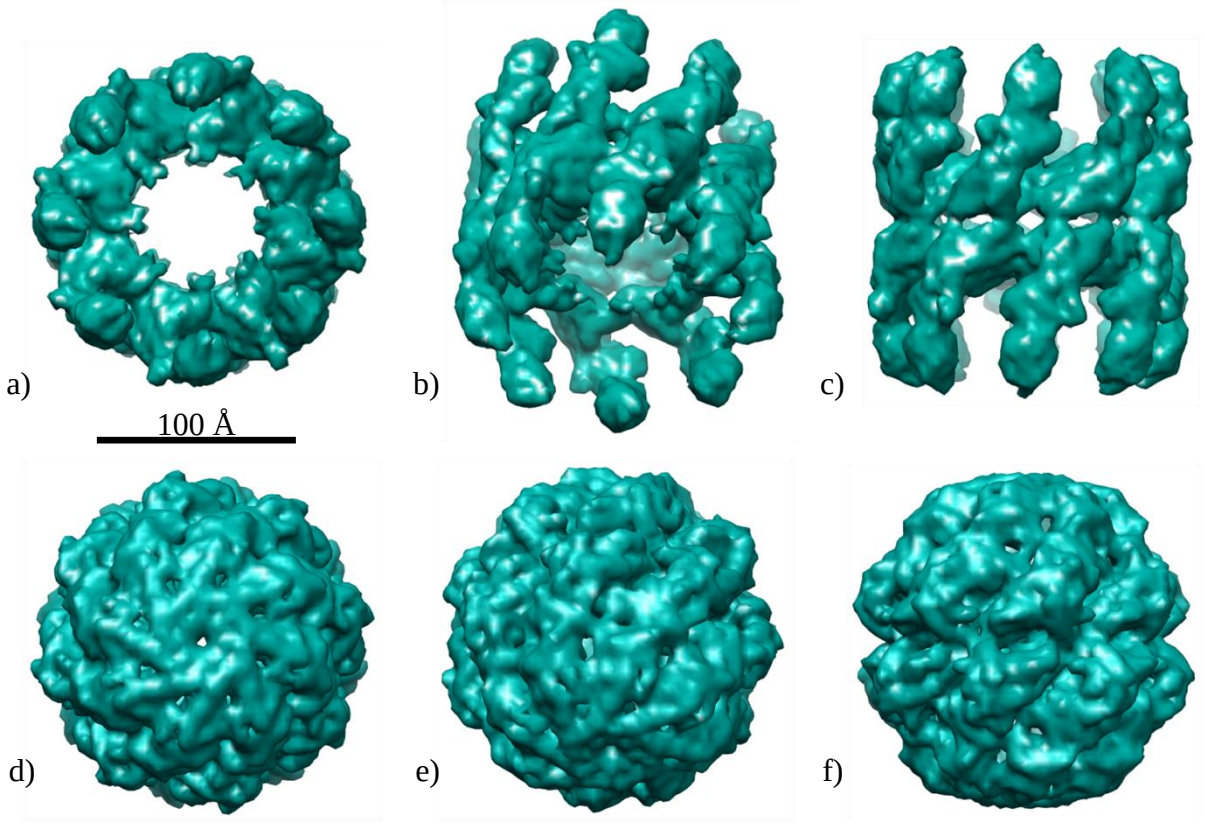


Figure 4.1 – Views of the Mm-cpn density maps. Top row: the macromolecule in the "open" state; bottom row: the macromolecule in the "closed" state. a,d) top view; b,e) intermediate view; c,f) side view. Density map generated in the IMAGIC package (van Heel *et al.*, 2012) from the atomic models deposited at the online Protein Data Bank (http://www.wwpdb.org) (Bernstein *et al.*, 1977), entries 3LOS for the "open" state and 3IYF for the "closed" state (Zhang *et al.*, 2010). Visualizations were generated using the UCSF Chimera package (Pettersen *et al.*, 2004).

5. Materials and Methods

This Chapter describes the methods and implementations used in the experiments performed along the project period. All computer programs and scripts were created by the author of this text, except for those which are explicitly denoted otherwise. Besides the clustering and consensus algorithms themselves, data manipulation and performance evaluation routines were also coded. The corresponding results and discussions are available in Chapters 6 and 7, respectively.

5.1 Datasets

We employed in our experiments synthetic and real datasets containing projection images of the Mm-cpn protein in "open" and "closed" conformations (Section 4.5). Its large size and high symmetry and, especially, the striking variations in its structural features between the two states make Mm-cpn a reasonable choice for investigating the feasibility of our classification approach. Samples containing Mm-cpn in the open and closed states embedded in vitreous ice were imaged separately on a JEM3200FSC (JEOL) transmission electron microscope operated at 300 kV acceleration voltage at the National Center for Macromolecular Imaging (NCMI), Houston, TX, USA. Micrographs were acquired on $4{,}096 \times 4{,}096$ Gatan CCD detectors. Particles were selected from the micrographs using the BOXER program and density maps for both states were obtained following the EMAN single particle reconstruction (SPR) workflow (Ludtke *et al.*, 1999). From the density maps, the atomic structures of Mm-cpn in both states were modeled and deposited in the Protein Data Bank (PDB) publicly accessible database (http://www.wwpdb.org) (Bernstein *et al.*, 1977), under accession codes 3LOS (open conformation) and 3IYF (closed conformation). Further details on sample preparation, imaging, reconstruction and atomic model generation can be found in the work by Zhang *et al.* (2010). In what follow we describe how we have prepared the synthetic and real datasets for our experiments.

5.1.1 Synthetic Mm-cpn projection images

Our first experiments were solely based on synthetic data. We simulated projection images of Mm-cpn in the open and closed conformations from the atomic models deposited at the PDB, using the following protocol:

1. Emulate density maps from the 3D atomic models, each with $100 \times 100 \times 100$ voxels;
2. Low-pass filter the 3D volumes;
3. Project each volume in 10,000 images, 100×100 pixels each with a 3 Å pixel size. Projection orientations are random.
4. High-pass filter the 2D projection images, in order to emulate one of the main characteristics of the TEM contrast transfer function (CTF) (Section 2.1.1).
5. Normalize images to zero mean and unit variance;
6. Add Gaussian noise of zero mean and variance of 10 to the images, in order to impose an SNR of 0.10 to the data, which is typical for cryo-EM datasets;
7. Randomly shift the images to simulate slight misalignments. We chose to draw the shift magnitudes from a 2D Gaussian distribution with mean at the image center and a standard deviation of 2 pixels;
8. Apply a circular binary mask to exclude most of the image background. The mask radius is of 45 pixels (90% of half the image side), which results in 6,349 valid pixels;
9. Re-normalize the pixels within the mask to zero-mean and unit variance;
10. Reduce the dataset dimensionality to 100 components using PCA.

All these steps were conducted with the IMAGIC package (van Heel *et al.*, 2012), except for step 7 which was performed in MATLAB. Filters in IMAGIC have Gaussian fall-offs in Fourier domain. The effect of high-pass filtering the images to roughly account for the effect of the CTF is displayed in Figure 5.1. We refer to the noiseless synthetic dataset as S1, and the noisy and misaligned dataset as S2. Examples of images from both synthetic datasets are shown in Figure 5.2. The native datasets have dimensions of $20{,}000 \times 6{,}349$, being 10,000 projection images from each conformation. After dimensionality reduction, they become $20{,}000 \times 100$ datasets.

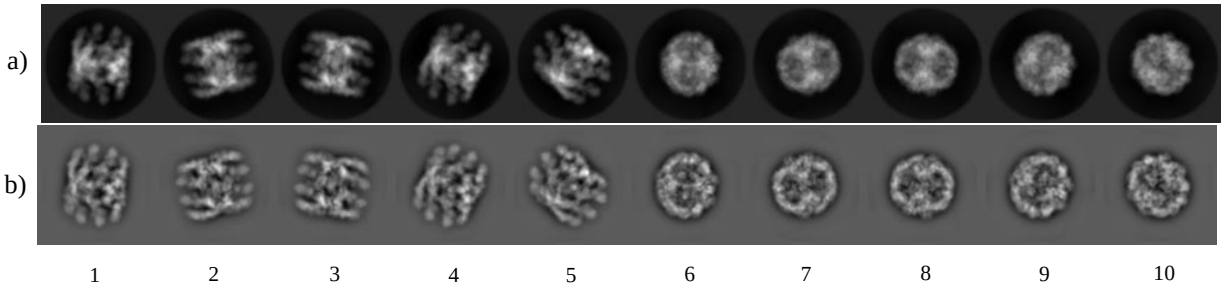


Figure 5.1 – Examples of images from the synthetic Mm-cpn datasets. Images 1 to 5 represent projections from the "open" state, while images 6 to 10 represent projections from the "closed" state. a) Aligned and noiseless projection images; b) images from a) after high-pass filtering (dataset S1);

5.1.2 Real Mm-cpn projection images

Besides the simulated images generated from the PDB models presented above, we also conducted experiments using the very own TEM images used for the derivation of such models. This set of real images was kindly provided by Dr. Junjie Zhang (Texas A&M University) and Dr. Wah Chiu (Baylor College of Medicine). This dataset contains 10,000 projection images, being 5,000 from each conformational state. Images have dimensions of 120×120 pixels each, with a pixel size of 2.6 Å. CTF correction (Section 2.1.1) had already been performed on these images. We prepared them for our experiments according to the following protocol:

1. Band-pass filter the 2D projection images, in order to suppress noise and low-frequency artifacts.
2. Normalize images to zero mean and unit variance;
3. Align the images. In order to account for different alignment conditions commonly found in cryo-EM datasets, we generated datasets using four different types of alignments:
    a. Center the whole dataset with respect to its average image (dataset R1);
    b. Center the whole dataset with respect to 10,000 random re-projections of the density maps obtained from the PDB models (Section 5.1.1), being 5,000 from each conformation (dataset R2). This alignment is expected to be better than that performed in a), as the images are aligned to "perfectly" centered corresponding re-projections;

   c. The same as b), but generating 24 rotated copies of each image in 15° steps (dataset R3). This procedure aims to emulate a larger dataset with increased number of rotations for each view. The rotational sampling of the projections impacts the quality of the subspace spanned by the eigenimages, as will be shown in Chapter 6.
   d. The same as b), but also performing rotational alignment of the images. This corresponds to a nearly optimal alignment condition (dataset R4).
4. Apply a circular binary mask to exclude most of the image background. The mask radius is of 43.2 pixels (72% of half the image side), which results in 5,785 valid pixels;
5. Re-normalize the pixels within the mask to zero-mean and unit variance;
6. Reduce the dataset dimensionality to 100 components using PCA.

A comparison between the synthetic and real images can be seen in Figure 5.2. Regarding dataset R3, the principal components were calculated from the 240,000 images dataset with artificially increased rotational sampling. However, only the coordinates of the original 10,000 images projected onto this subspace were considered on the experiments which make use of this dataset. More details about the application of PCA to the synthetic and real datasets will be presented in Chapter 6, as well as their eigenimages.

We point out that all four real datasets correspond to practical situations found in the Single Particle Analysis workflow. Dataset R1 corresponds to the first round of translational alignment, in which the images are simply centered against the whole dataset center of mass. Dataset R2 would occur in a later round of alignment, where a given iteration of a 3D model is already available and whose reprojections can be used to refine the image shifts. Dataset R4 follows the same logic, but includes rotational alignments as well. Dataset R3 is conceptually similar to R2, but emulates a situation in which a larger dataset is available.

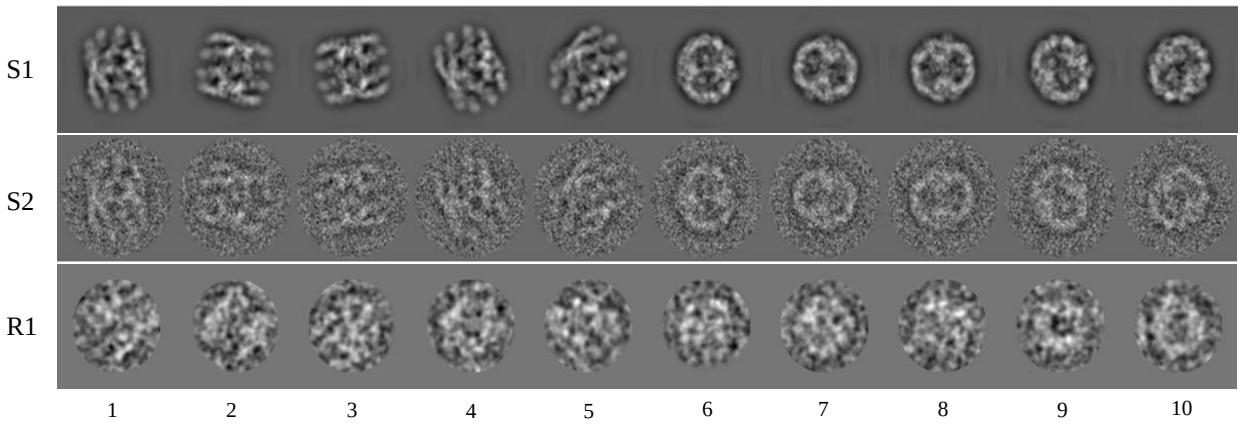


Figure 5.2 - Selected images from the analyzed datasets. The first five images are from the "open" state and the following five are from the "closed" state. S1: noiseless synthetic dataset; S2: Noisy and misaligned synthetic dataset; R1: experimental dataset after band-pass filtering and centering.

5.2 Experiments

5.2.1 Exploratory Data Analysis: SOM

The initial experiments were performed using the Self-Organizing Map. The goal of these experiments was to check whether the SOM is able to discriminate the conformational states on the synthetic datasets (S1 and S2). Several SOMs were trained having different sizes and topology configurations. We analyzed them based on the U-matrix and the distribution of the best matching units (BMUs) across the map. The maps displayed in this work had 10×25 neurons arranged on a sheet, with hexagonal neighborhood. Neurons were initialized with random values. The functions from the SOM Toolbox for MATLAB (Vesanto, Himberg, Alhoniemi & Parhankangas, 2000) were used to train and analyze these maps.

5.2.2 Experiment 1: Consensus by simple agreement

Later, we decided to evaluate whether an unsupervised consensus classification could provide higher accuracy and stability in discriminating the conformations on datasets S1 and S2. The native representation of the data by the pixels as feature vectors was employed. We used two algorithms: SOM and spectral clustering. The SOM was chosen in order to benefit from the results of previous experiments. Spectral clustering was chosen due to its ability to detect clusters

of arbitrary shape. For simplicity, we considered the number of clusters to be equal to the true number of classes, i.e., $K = 2$.

- **SOM clustering:** Cluster labels were assigned to the map prototypes by manually segmenting the U-matrix. The labels were then extended to the dataset by assigning to each object the label of its BMU.

- **Spectral clustering:** an ε-neighborhood similarity graph was constructed by calculating the pairwise normalized cross-correlation (Equation 3.17) between all images, and applying a similarity threshold $\varepsilon = 0.65$. The normalized cross-correlation lies in the interval $[0, 1]$, where 1 denotes perfect similarity. The adjacency matrix of the graph is thus a $20{,}000 \times 20{,}000$ sparse matrix. With this matrix as input, the unnormalized spectral clustering algorithm was applied (Algorithm 3.6).

- **Consensus clustering:** as a method of consensus, we used the simple agreement between the solutions of the two algorithms above. The canonical labeling scheme from Section 3.4.2.1 was employed, and whenever the two solutions agreed on the label for a given object, the object was retained on the consensus solution; if they disagreed, the object was discarded.

We then evaluated the *purity* of the classes obtained, which is the proportion of correctly assigned members in relation to the total number of class members. By "correctly assigned" we adopt the majority of cluster members belonging to the same class in the true solution.

5.2.3 Experiment 2: Determining the number of clusters

In a subsequent experiment, we decided to include more clustering algorithms in the ensemble, and try to automatically determine the number of structural classes. Datasets S1, S2 and R1 were analyzed by the data projections on the 100 components extracted with PCA. This is the maximum number of components allowed by the efficient parallel implementations of MSA al-

gorithms in the current version of the IMAGIC package (van Heel *et al.*, 2009). The model selection criterion was the optimization of ANMI according to the number of clusters (Section 3.4.5.1). Here, we adopted the three consensus heuristics proposed by Strehl & Ghosh (2002): CSPA, HGPA and MCLA, described in Section 3.4.5.3. A "meta-consensus" strategy was employed, running the three heuristics and choosing that yielding the highest ANMI values. If the number of clusters determined by the ANMI peak is different from the true number of clusters, the correspondence between the obtained clusters and the true labels can be verified by a confusion matrix. The base solutions were provided by HAC, k-means, SOM and three versions of spectral clustering ($R = 6$). The number of clusters was changed from 2 up to 10 for each algorithm, making a total of 54 base solutions. The implementations and parameter setup of each algorithm were the following:

- **Hierarchical Ascendant Clustering:** the HAC algorithm (Algorithm 3.1) was applied using Euclidean distance and the Ward criterion for cluster merging. The **linkage** and **cluster** MATLAB functions were employed to this end. This is similar to the classification procedure employed by IMAGIC (van Heel *et al.*, 2009), but without post-processing the clusters to improve the Ward criterion.

- **k-means:** we applied k-means (Algorithm 3.2) in a straightforward manner. An efficient implementation for MATLAB was employed (Chen, 2012). This program randomly selects *K* data points as initial prototypes.

- **Self-Organizing Map:** to avoid having to manually segment the U-matrix as done on Experiment 1, which can be sometimes a very subjective procedure, we used the SOM in a more simple fashion. The SOM was trained having a number of neurons equal to the number of desired clusters *K*. After training, each data point was assigned to its BMU, and this assignment was regarded as the object's cluster label. This is similar to the procedure adopted by Strehl, Ghosh & Mooney (2000).

- **Spectral Clustering:** we applied three versions of spectral clustering (Algorithm 3.6): the unnormalized version (von Luxburg, 2007), the normalized version ac-

cording to Shi & Malik (2000), and the normalized version according to Ng, Jordan & Weiss (2002). While the first version approximates a solution to the RatioCut, the two latter approximate solutions to the Ncut (Section 3.4.4.5.2). The graph adjacency matrix was the same from Experiment 1. The idea behind using these three versions is to have the inherent properties of graph partitioning algorithms, like finding clusters of arbitrary shape, while at the same time accounting for diversity in the ensemble, at least theoretically.

- **Consensus solutions:** consensus solutions were obtained by applying the CSPA, HGPA and MCLA heuristics implemented in the ClusterPack toolbox for MATLAB (Strehl, 2011). Internally, these heuristics make use of the METIS and HMETIS (hyper)graph partitioning algorithms (Karypis *et al.*, 1997; Karypis & Kumar, 1998).

5.2.4 Experiment 3: The influence of alignment quality

In this experiment, we aimed at verifying the influence of the alignment quality between the images on the separation of conformational states by unsupervised algorithms. We analyzed datasets S1, S2, R1, R2, R3 and R4 projected along 100 principal components, as before, and also using only the first 10 components. For the real datasets, we also evaluate how the classification is influenced by the projection orientation, as some specific views of the macromolecule may appear quite similar in both conformations. In order to emulate a more practical scenario, we imposed a number of clusters equal to the true number of conformations present in the dataset. This implies our cluster ensemble approach is being used as a partitioning validation tool, that is, the user takes the result from the robust unsupervised classification as an indicator of the accuracy of the structural assignments performed by conventional SPR methods. The underlying assumption is that the existence of distinct conformational manifolds may be observable from the ensemble of base clustering solutions. Additionally, we included two more algorithms in order to improve the diversity of the ensemble: a mixture of Gaussians and METIS (Karypis, 2013), and used only one version of spectral clustering ($R = 6$). Implementation details are as follows:

- **Hierarchical Ascendant Clustering:** HAC was used in the same way as in Experiment 2.

- **k-means:** k-means was used in the same way as in Experiment 2.

- **Self-Organizing Map:** SOM was used in the same way as in Experiment 2.

- **Gaussian Mixture Model:** we performed clustering by modeling the dataset as a mixture of two multivariate Gaussian distributions (Algorithm 3.3). Each data point was assigned in a "hard" fashion to the distribution with the highest *a posteriori* probability of membership. Additionally, the *a posteriori* probability can be used to assess the reliability of the cluster assignments. Functions **gmdistribution.fit** and **cluster** of the MATLAB Statistics Toolbox were used to implement this clustering method.

- **Spectral Clustering:** only the unnormalized version of spectral clustering was used this time. Using the pairwise normalized cross-correlation as similarity measure, a sparse adjacency matrix for an unweighted graph was created by connecting each data point to its 10 nearest neighbors or more; vertices were connected either if object *i* was among the most similar to object *j* or the converse. In this way, each vertex is connected to *at least* 10 other vertices (Section 3.4.3.5.1). A publicly available spectral clustering toolbox that implements the methods described in von Luxburg (2007) was used to create the adjacency matrices (Buerk, 2012).

- **METIS:** the METIS algorithm was used from the interface provided by the ClusterPack toolbox to its binaries (Strehl, 2011). We used the default configurations, which include the *Sorted Heavy-Edge Matching* (SHEM) for graph coarsening down to 100 vertices, *Greedy Graph Growing Partitioning* (GGGP) for initial biSection, and the "adaptive" Boundary Kernighan-Lin (BKL(*,1)) refinement for

uncoarsening. For more details on these methods, please refer to Section 3.4.3.5.4 and to the manual by Karypis (2013).

- **Consensus solutions:** besides CSPA, HGPA and MCLA, we also used k-means as a consensus method by providing it with the set of base labels as new features for the data, due to its simplicity and speed (Section 3.4.5.1). We also used the cluster confidence measure provided intrinsically by MCLA to assess the quality of the assignments by this algorithm.

5.3 Performance assessment

We will use agreement measures like the Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI), both explained in Section 3.4.5.2, as well as the percentage of matches between the label lists. We note that this latter form of comparison, although of intuitive appeal, may be misleading because it does not account for random matches in any form. To aid the performance analysis, we may also use confusion matrices to understand the data assignments across clusters (Section 3.4.3) and plots of the data points projected along selected principal components.

6. Results

In this Chapter, the results from the experiments described in Chapter 5 will be presented and briefly discussed. The conclusions from these experiments are drawn in Chapter 7. Our most important findings derive from sections 6.2.1 and 6.2.4.

6.1 Principal Component Analysis

We begin by analyzing the distribution of the data points on the subspace spanned by their principal components. This may allow us to get an intuition of how the data behaves in the higher dimensional feature space of pixel intensities. Figure 6.1 shows the first 10 eigenimages for the six Mm-cpn datasets used in this work. Eigenimages are visualizations of the principal components for image sets. Note that the second eigenimage is the same for datasets S1 and S2, except for a signal inversion. This inversion is irrelevant for PCA because they represent the same projection direction. Interestingly, the eigenimages for datasets S1 and S2 are quite similar, because they contain essentially the same underlying information except for the noise. This indicates that PCA is able to extract components that contain most of the true signal information despite it being disturbed by severe random noise, which is due to the statistical preservation nature of PCA (Section 3.3.1). Also, dataset S2 has two eigenimages (highlighted in red) introduced by the misalignments between images (Dube *et al.*, 1993). We also observe that, as the quality of the alignment improves for the real datasets, the first principal components become more informative regarding the true underlying signal. This improvement can be clearly seen by comparing datasets R1 and R2 in Figure 6.1. Also, as more data becomes available for determining the principal components, their signal-to-noise (SNR) ratio improves. This can be noted by observing that eigenimages for dataset R3, which were calculated from a set of 240,000 artificially rotated images (originally 10,000), are more similar to those of dataset S1, which are for the noiseless dataset containing 20,000 images.

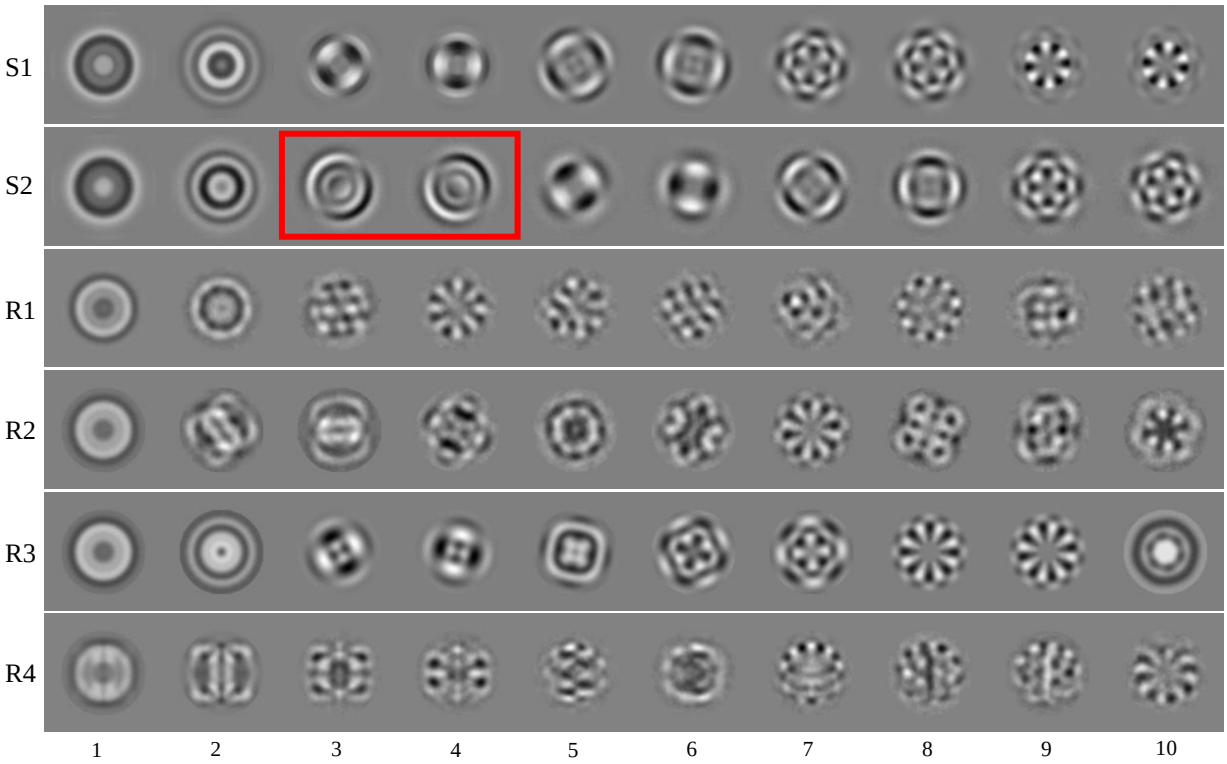


Figure 6.1 - The first 10 eigenimages for each of the six Mm-cpn datasets analyzed in this work. For dataset S2, highlighted in red are eigenimages 3 and 4, which do not have similar correspondents in dataset S1.

We can also evaluate the spectrum of eigenvalues for the datasets in order to better understand how the first principal components "explain" the variance of the data distributions (Section 3.3.1). Such spectra are shown in Figure 6.2. It is remarkable how the introduction of noise spreads the variance across the principal components, as seen for dataset S2 and for the real datasets, in comparison to the eigenvalue spectrum of dataset S1. Additionally, it can be seen that the quality of the alignment tends to concentrate more variance on the first principal components. This can be observed for dataset S1 (perfectly aligned) in relation to dataset S2 (imposed misalignments), dataset R2 (centering according to 3D re-projections) in relation to R1 (centering according to the dataset's average image), which is probably the most remarkable case, and also for dataset R4 (translational and rotational alignment according to 3D re-projections) in relation to dataset R3 (centering according to 3D re-projections, but with artificially increased rotational sampling for calculation of the principal components).

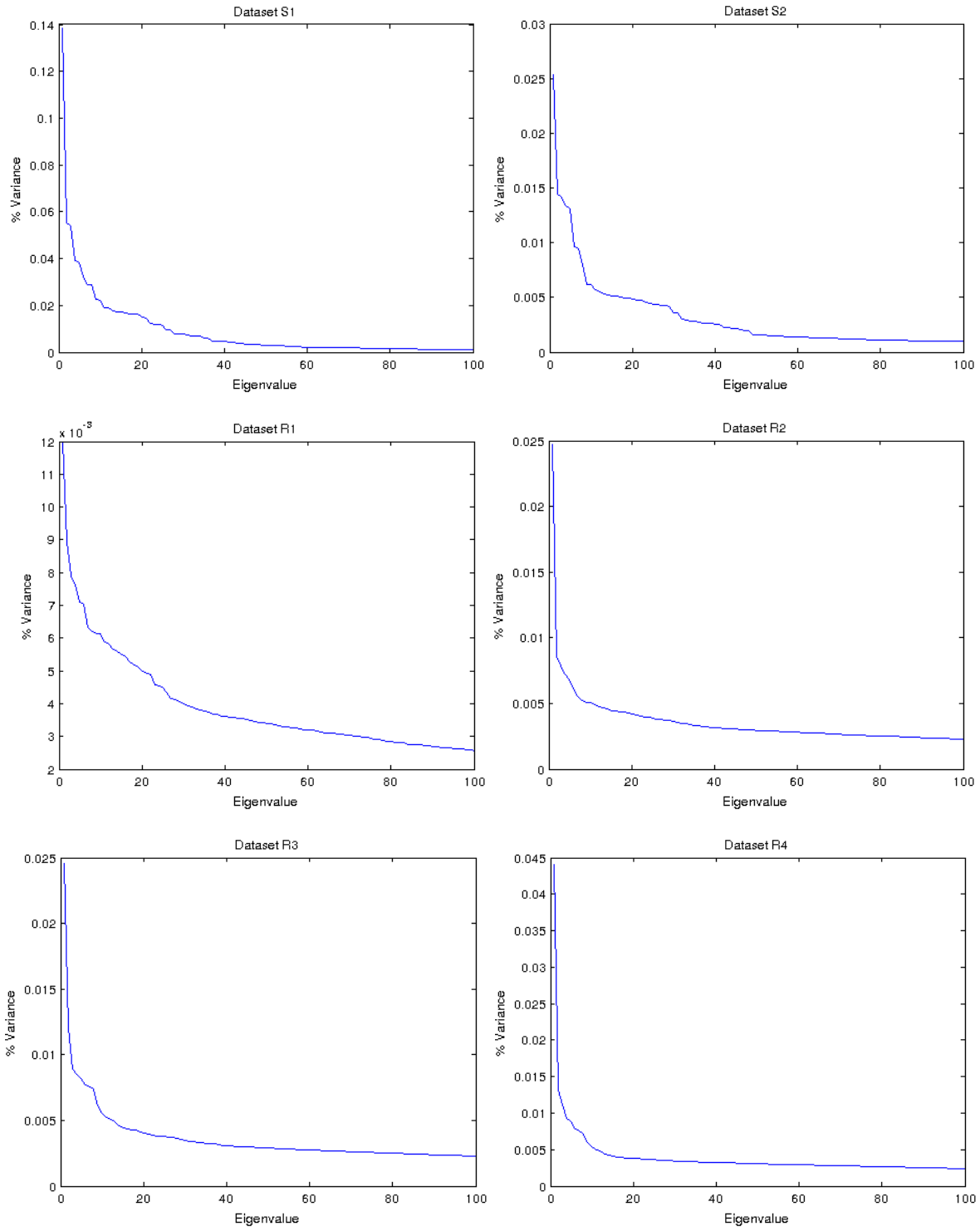


Figure 6.2 – The spectrum of the first 100 eigenvalues for the Mm-cpn datasets analyzed in this work. Eigenvalues are plotted as the fraction of the total dataset variance they contain.

The accumulated variance for the first 10 and the first 100 principal components are shown in Table 1. Combined with the previous figures, this table allows us to observe how low

the SNRs of these datasets are. Random noise causes most of the dataset variance to be distributed more or less evenly across the eigenvalue spectrum, except for the very few first ones, as shown in the plots from Figure 6.2. These first are the ones which mostly "explain" the true molecular projection signal, as can be seen in Figure 6.1. The increase on variance associated with the first 10 principal components as consequence of alignment improvement, mentioned before, can also be inferred from Table 6.1. The remaining of the eigenimages mostly describes random variations, in relation to the whole dataset statistics. These observations demonstrate the power of PCA both in filtering the noise out of the data, and compressing the relevant variance of a high-dimensional dataset (6,349 dimensions for datasets S1 and S2, and 5,785 dimensions for datasets R1, R2, R3 and R4) onto a subset of only a few dimensions (10 or 100 in this case).

Table 6.1 – Accumulated variance on the first 10 and first 100 principal components, for the six datasets analyzed in this work. The values are displayed as the percentage of the total dataset variance they contain.

%	First 10	First 100
S1	45.97	91.54
S2	11.99	32.40
R1	7.54	39.34
R2	8.24	35.88
R3	9.66	36.91
R4	12.05	39.83

Besides dimensionality reduction and "eigenfiltering", PCA is also very useful for visualization. For example, as eigenimages 1 and 2 are essentially the same for datasets S1 and S2, we can see how the introduction of noise and misalignments between images affects the data distribution, as plotted in Figure 6.3. By applying the original data labels, it is also possible to observe how the projection images from each conformational state are distributed in a given subspace.

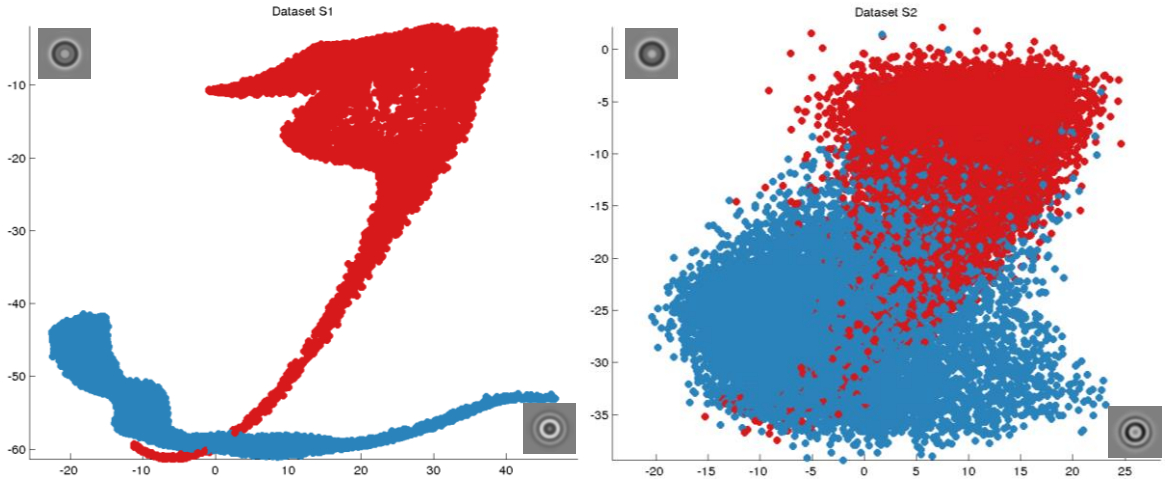
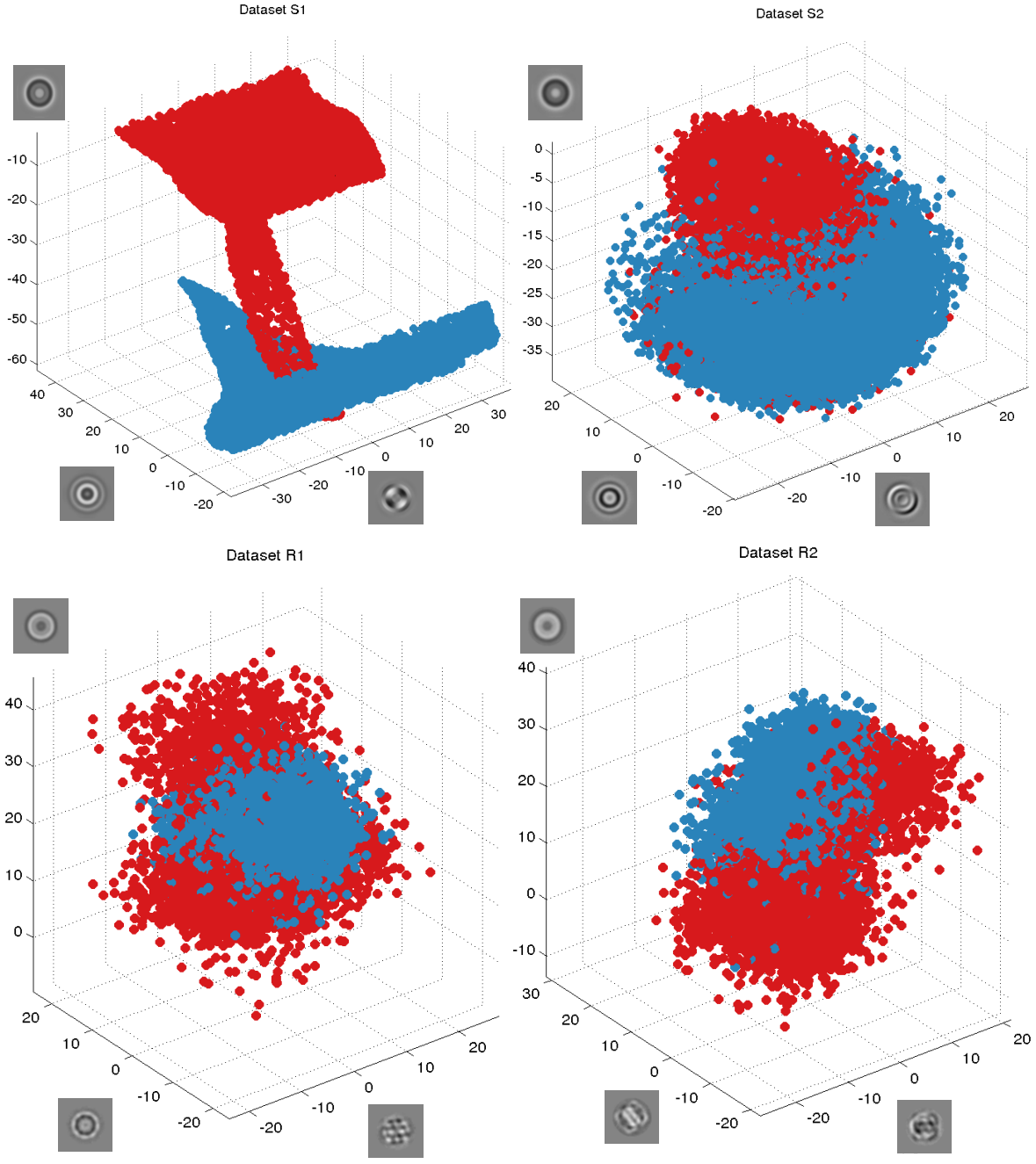


Figure 6.3 – Datasets S1 and S2 plotted along their respective first two principal components. The insets show the eigenimages corresponding to each axis. Data points belonging to the "open" state are colored red, while data points from the "closed" state are colored blue.

From Figure 6.3 it becomes clear that, for the synthetic datasets, the projections from each conformational state occupy relatively well-defined manifolds in a multidimensional feature space. As seen for dataset S2, when images are noisy and not perfectly aligned, such manifolds may become hardly recognizable. In order to get an insight on the data distribution in relation to the conformational class on all datasets, Figure 6.4 shows the data projected onto the first three principal components for each dataset. It is important to bear in mind that, although these first three components are those with largest associated variance, they are not necessarily the best to discriminate the conformational manifolds. This becomes especially critical if the number of projections of different conformations is highly imbalanced in the dataset. Due to the statistical representation properties of PCA, in this case, components associated with the discrimination of conformational changes would likely not be found among the first ones. These plots may also be useful to evaluate the performance of the unsupervised classification algorithms applied in the experiments in comparison to the ground truth.


Dataset S1
Dataset S2
Dataset R1
Dataset R2

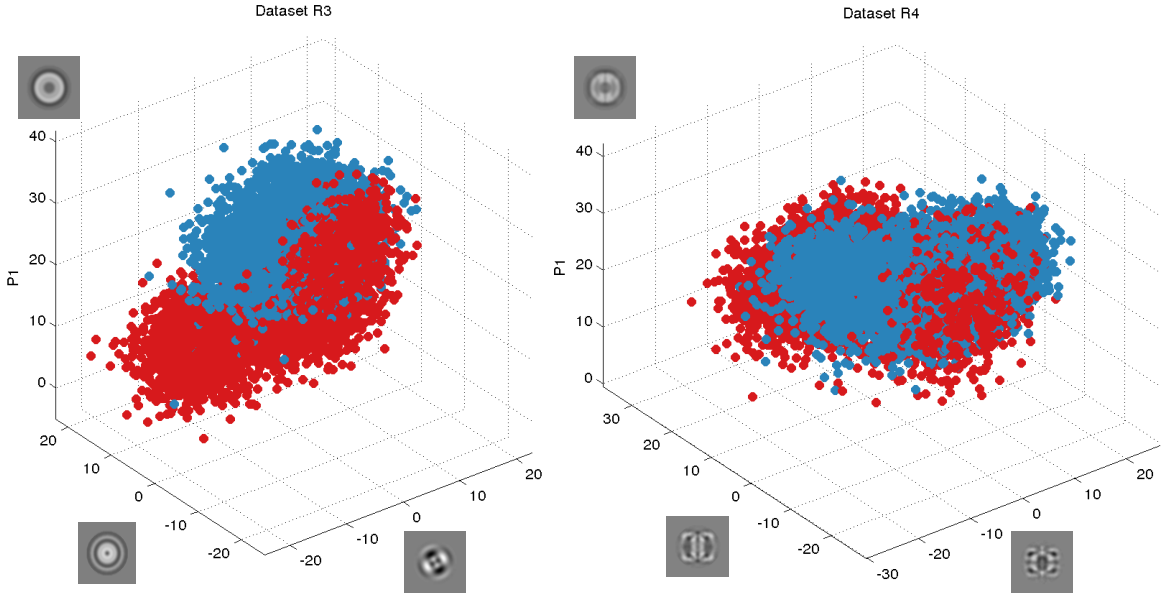


Figure 6.4 – The datasets plotted along their respective first three principal components. The insets show the eigen-images corresponding to each axis. Data points belonging to the "open" state are colored red, while data points from the "closed" state are colored blue.

6.2 Experiments

6.2.1 Exploratory Data Analysis: SOM

Our first experiments using the Kohonen SOM (Kohonen, 2001) were very important in order to verify that an unsupervised learning algorithm could indeed recognize the separation of conformational states in the Mm-cpn synthetic datasets. What is most interesting is that this separation can be observed even in the absence of rotational alignments between images. Previous uses of the SOM in single particle analysis had only discriminated structural heterogeneity on aligned sets of projection views, as described in Section 2.7.3 (Marabini & Carazo, 1994). The SOMs shown in this Section were trained using the native data representation, that is, the image pixels.

A first analysis that can be conducted with the SOM when trained on sets of images is to directly observe the neurons or *codebooks* after training, as shown in Figure 6.5. In the case of dataset S1, it can be clearly seen that the neurons from the region to the left of the map learned to represent views of Mm-cpn in the open state, with different in-plane rotations, while the right-

most part of the map learned to represent views from the closed state. For dataset S2, the map also learned to represent views of distinct conformations on distinct regions: the lower left portion of the map is associated with views from the open state, while the upper and rightmost parts are associated with views from the closed state.

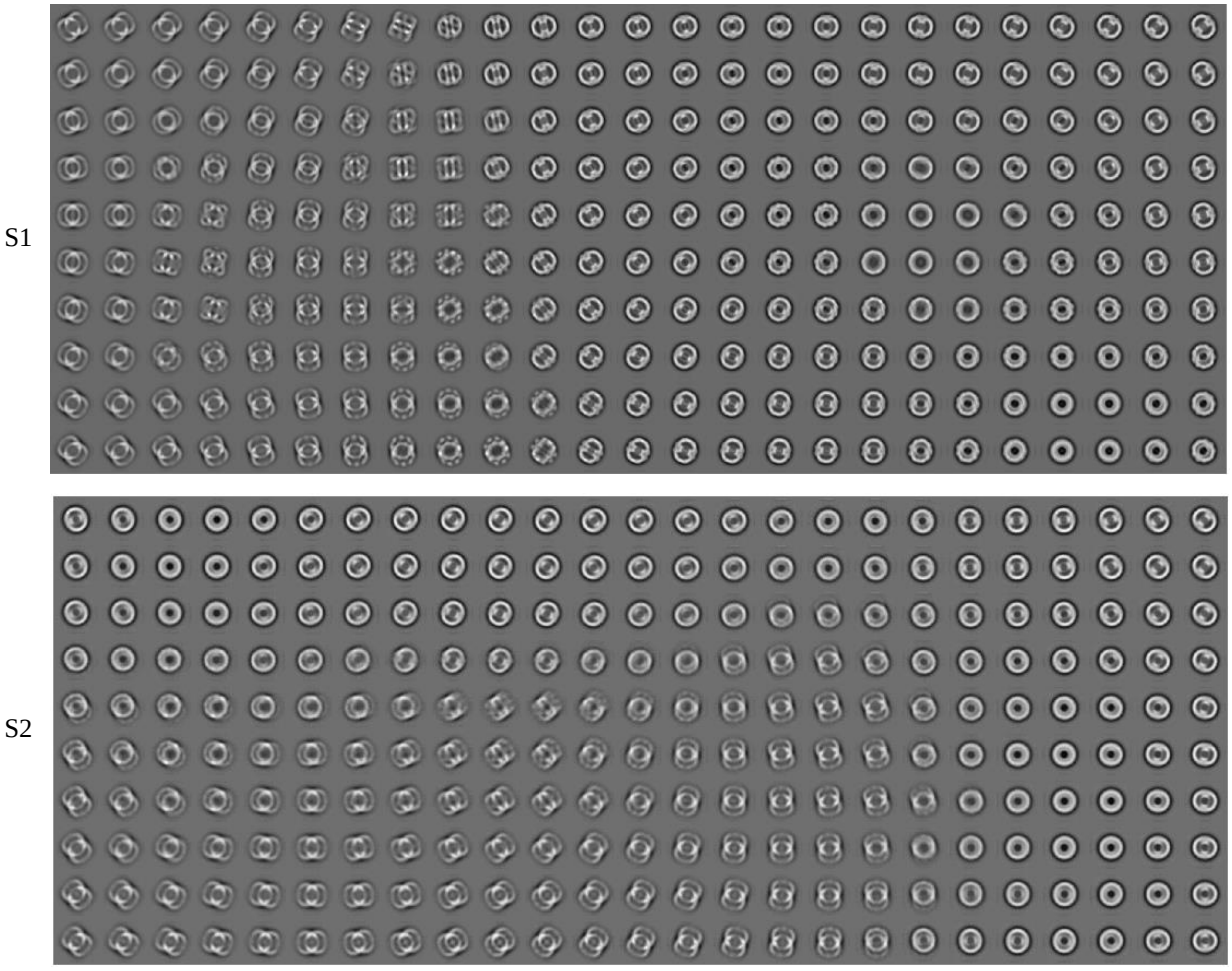


Figure 6.5 – The codebooks or neurons of the SOM after training on datasets S1 and S2.

In order to detect clusters within the map, we can plot the U-matrix explained in Section 3.4.3.4 (Ultsch & Siemon, 1990). The U-matrices for the trained SOMs presented in this Section are shown in Figure 6.6. It can be clearly observed that the inter-neuron distances are greater in the regions where the maps transition from representing the open conformation to the closed conformation, if we compare Figure 6.5 and Figure 6.6. It can also be observed that within the regions corresponding to the open state (the leftmost part of the map for dataset S1, and the lower leftmost part of the map for dataset S2) the inter-neuron distances are higher, because views in

different orientations of the "open" Mm-cpn are more diverse than views from the "closed" Mm-cpn. For example, the top-view of the open state has dimensions very distinct to those observed on a side-view of the same state (Figure 4.2). An interesting feature of these U-matrices is the relative *continuity* displayed by neurons associated with a same conformational state. This is an indication that the projection images from the distinct Mm-cpn conformations lie on different multidimensional manifolds. While the data clouds shown in Figure 6.4 indicated a similar behavior, they were projected on an arbitrary subspace, while the SOM allows this interpretation from the native feature space of the data.

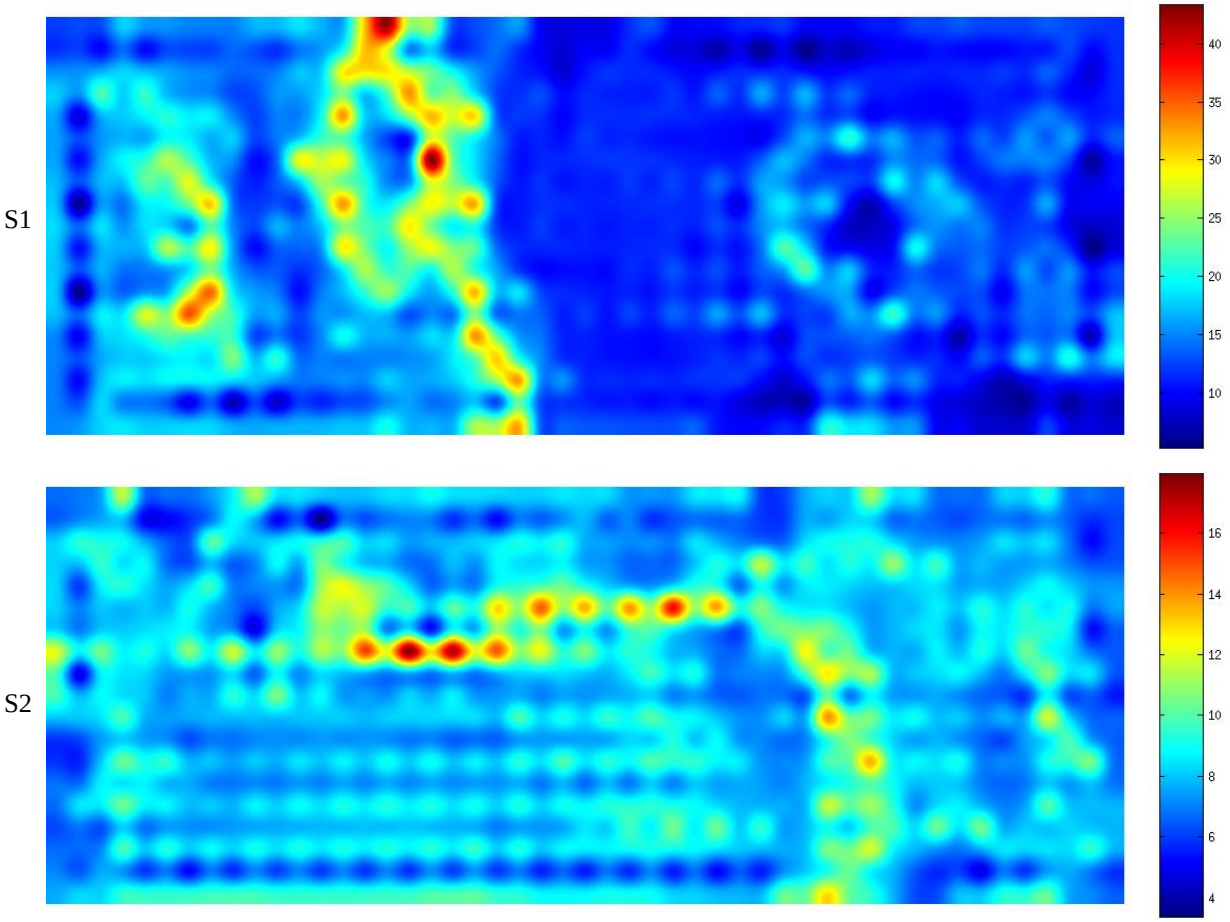


Figure 6.6 – The U-matrices for the SOM trained on the S1 and S2 datasets. Color code indicates average Euclidean distances between a codebook and its neighbors.

Figure 6.5 and Figure 6.6 suggest that, if we segment the U-matrix, we may be able to classify the codebooks according to the conformational state they belong to. Consequently, the whole dataset can also be labeled according to this interpretation. This can be done by assigning a

data point the label of its most similar neuron on the map, called its Best Matching Unit (BMU). Figure 6.7 shows the labels applied to the map codebooks by a straightforward manual segmentation of the U-matrices. These labels were used later in Experiment 1 (Section 6.2.2).

Figure 6.7 – The SOM trained on datasets S1 and S2 labeled by manual segmentation of the U-matrix. Red corresponds to neurons associated with the open conformation, while blue corresponds to those associated with the closed conformation.

Indeed, we can plot on the map the number of hits for each neuron, that is, for how many data points a given neuron is the best matching unit. Such plot is shown in Figure 6.8. We can see that some neurons are not the BMU for any data point. Interestingly, these are the neurons which have the highest average distances to their neighbors in the U-matrix, something which can be verified in Figure 6.6. This means that these neurons lie on low density regions of the multidimensional feature space, i.e., they are just transitional neurons from one data-populated hypervolume to another.

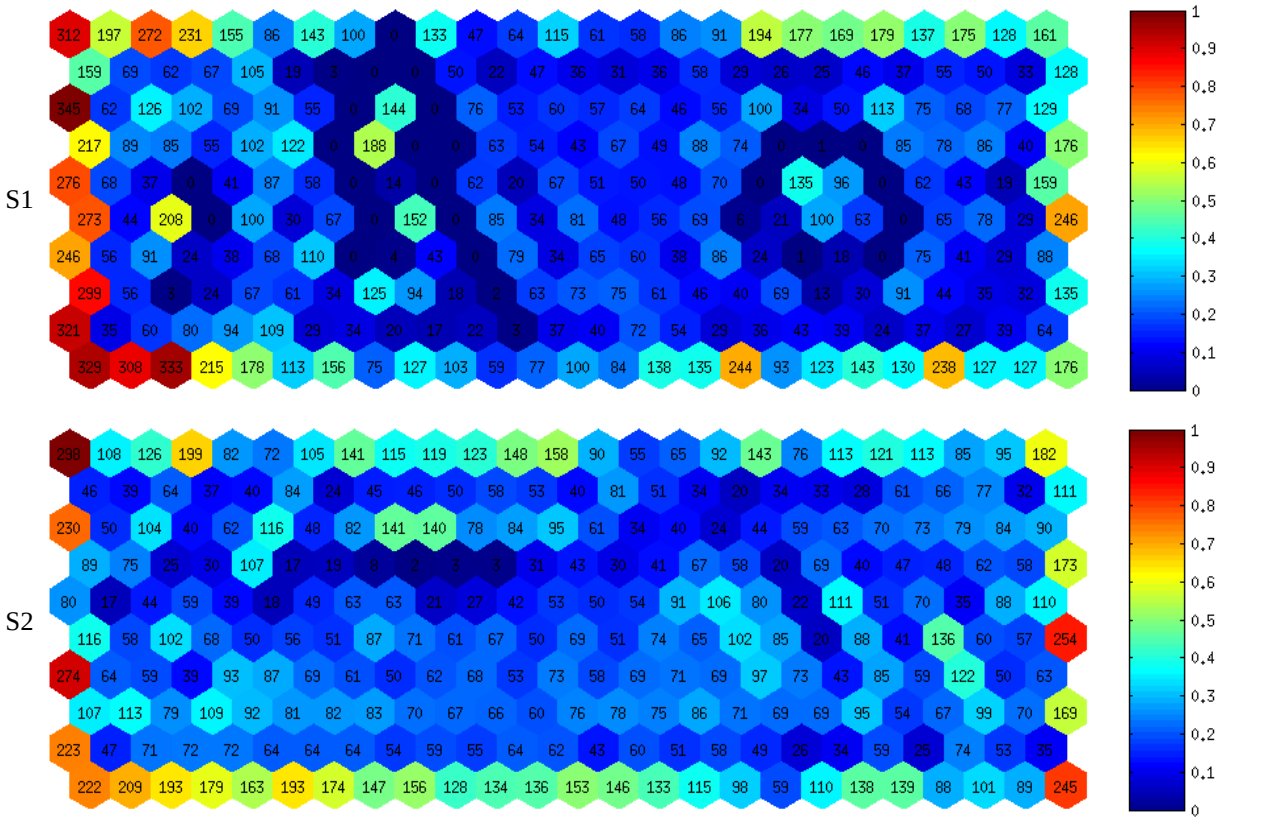


Figure 6.8 – The distribution of "hits" for the SOM trained on datasets S1 and S2. Color code indicates the density of data points associated with a given neuron on the map.

Up to this point, we have been emulating what could be analyzed on a general unsupervised scenario, not using the true data labels in the assessment of the SOM discrimination performance. Using these labels, we can verify that the SOM indeed learns to discriminate the conformational states, as presented in Figure 6.9. In the case of the noiseless dataset S1, all neurons are associated only to data points from the same conformation, that is, they are 100% pure in terms of conformational representation. Interestingly, there is a set of isolated neurons associated with the open conformation in the middle of the region associated with the closed conformation. This detail was not perceived by our naïve segmentation approach of the U-matrix, because the inter-neuron distances between this isolated region and its surroundings are not impressively large (Figure 6.6). See Experiment 1 (Section 6.2.2) for the classification errors implied by this method. For the noisy dataset, it can be seen that neurons in transitioning regions of the map tend to exhibit intermediate purity.

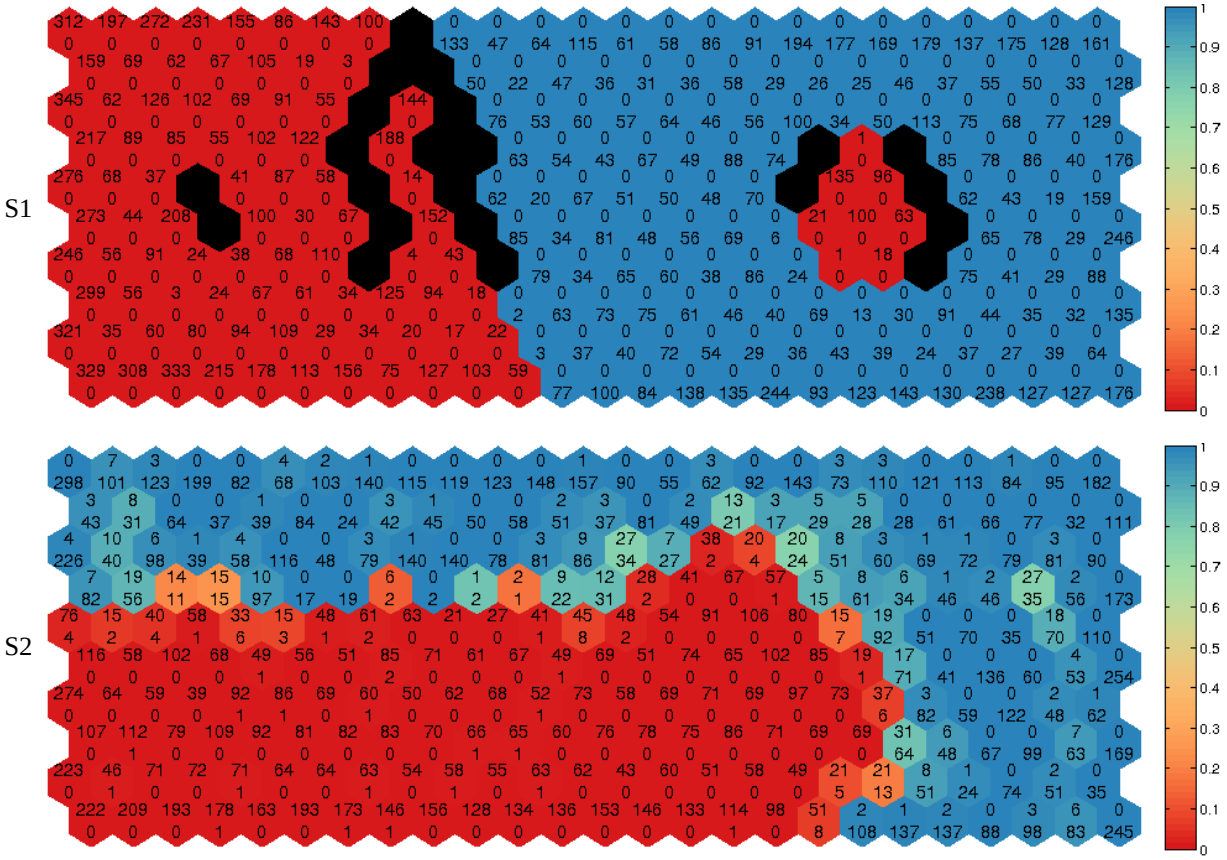


Figure 6.9 – The number of hits on each neuron of the map discriminated by their true conformational labels, for datasets S1 and S2. Color code indicates the purity within the total hits on a given neuron. Colors towards blue indicate that most data points associated with the neuron are from the closed state, while colors towards red indicate that most associated data belong to the open state. Intermediate colors indicate data points from mixed conformational states, that is, a purity close to 50%. Black neurons are those with zero hits.

6.2.2 Experiment 1: Consensus by simple agreement

In Experiment 1, we tried a simple consensus approach between the classification results obtained by the SOM, using the segmentation shown in the previous subsection, and the labels provided by unsupervised spectral clustering. Table 6.2 displays the results for dataset S1 and Table 6.3 displays the results for dataset S2. As expected from the observations of data clouds (Figure 6.3) and the U-matrices (Figure 6.6), heterogeneity classification on the noisy and misaligned dataset is harder. Nevertheless, both SOM and spectral clustering could achieve a classification accuracy (average class purity) of about 90% on dataset S2. What is more important, though, is that the consensus solution obtained by simple agreement between the two base solutions achieved higher purities than any algorithm alone, in both cases. We acknowledge that such

high accuracy came at the cost of rejecting images (about 14% of dataset S2). However, it is in principle possible to compensate for this cost, by collecting more images in the TEM if the rejection rate becomes prohibitively high.

Table 6.2 - Unsupervised classification results from Experiment 1, dataset S1.

	Spectral	SOM	Consensus	Ground truth
"open"	8,716	9,676	8,589	10,000
"closed"	11,284	10,324	10,197	10,000
Rejected	0	0	1,214	0
Errors in "open" class	116	162	4	0
Errors in "closed" class	1,400	486	471	0
Total error (without rejects)	1,516	648	475	0
Rejected [%]	0	0	6.07	0
"open" class purity [%]	98.67	98.33	99.95	100
"closed" class purity [%]	87.59	95.29	95.38	100
Total purity (without rejects) [%]	92.42	96.76	97.47	100

Table 6.3 –Unsupervised classification results from Experiment 1, dataset S2.

	Spectral	SOM	Consensus	Ground truth
"open"	8,957	10,460	8,296	10,000
"closed"	11,043	9,540	8,879	10,000
Rejected	0	0	2,825	0
Errors in "open" class	806	1,164	244	0
Errors in "closed" class	1,849	704	605	0
Total error (without rejects)	2,655	1,868	849	0
Rejected [%]	0	0	14.12	0
"open" class purity [%]	91.00	88.87	97.06	100
"closed" class purity [%]	83.26	92.62	93.19	100
Total purity (without rejects) [%]	86.72	90.66	95.06	100

6.2.3 Experiment 2: Determining the number of clusters

In a second experiment, we aimed to improve the consensus clustering approach, avoiding rejections. To this end, we employed the three heuristics proposed by Strehl & Ghosh (2002): CSPA, HGPA and MCLA. The set of base solutions was provided by six clustering algorithms: HAC, k-means, SOM and three versions of spectral clustering. The number of clusters requested for each algorithm and each consensus solution varied from 2 up to 10. This makes a total of 54 base clustering solutions. In order to avoid the subjectivity of the manual segmentation of the U-matrix, as well as avoiding the processing time required by training a large map, we used the SOM with the number of neurons equal to the number of requested clusters.

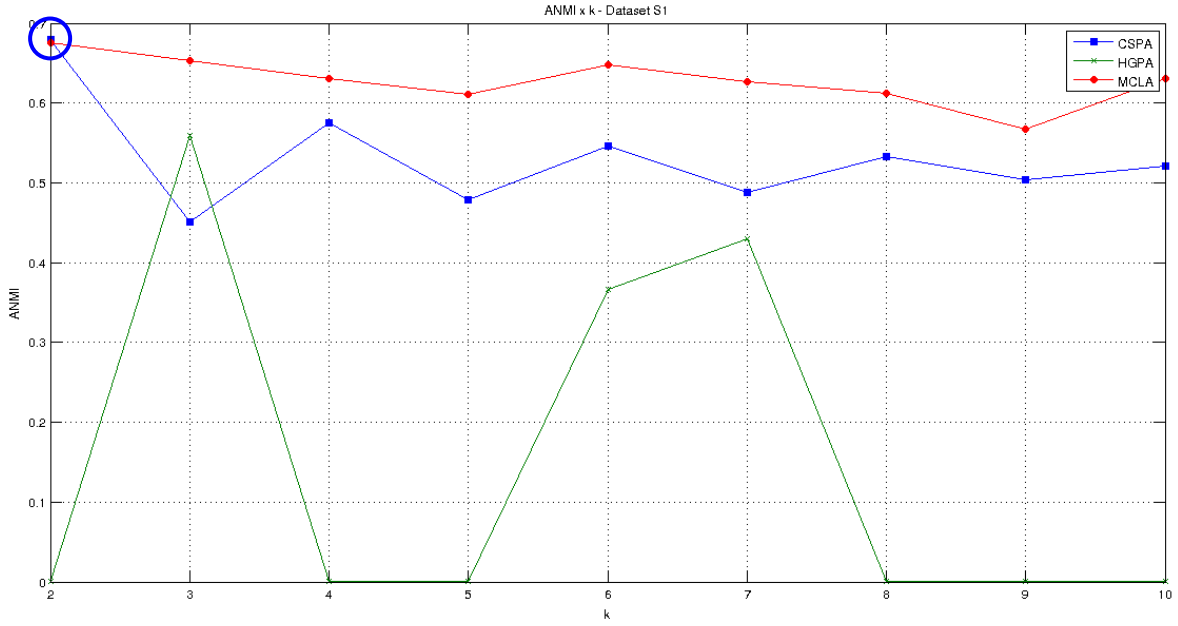
First, we evaluated whether the ensemble provided more accurate solutions than the individual algorithms. Table 6.4 shows the classification accuracy for all base solutions and consensus heuristics when $K = 2$.

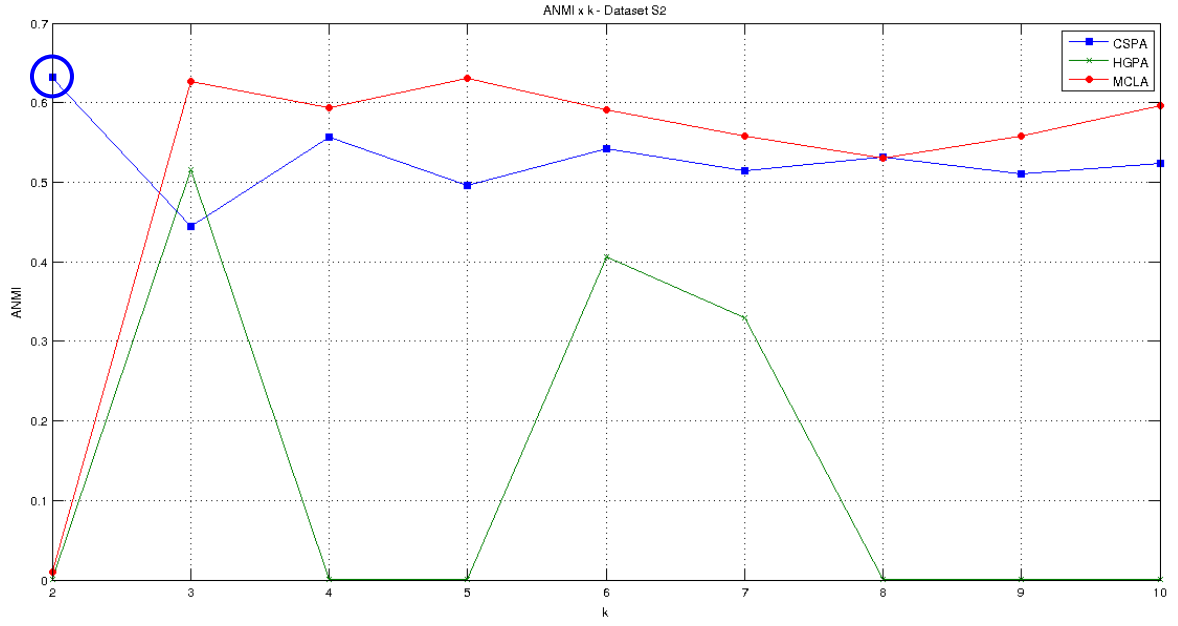
Table 6.4 – Classification accuracy of the cluster ensemble employed in Experiment 2 for the solutions containing only two clusters. Results are given as the percentage of matches with the true labels. The best performers are highlighted in bold for the individual algorithms and the consensus heuristics.

%	HAC	k-means	SOM	Spectral A	Spectral B	Spectral C	CSPA	HGPA	MCLA
S1	98.57	97.43	97.56	**100**	**100**	**100**	**100**	49.90	98.57
S2	91.79	91.82	87.86	99.98	99.98	**99.99**	**99.98**	49.66	50.03
R1	63.25	**63.46**	44.44	49.99	49.98	49.98	**78.41**	50.00	49.92

We also employed the inherent model selection method available for cluster ensembles, in order to determine the number of structural classes in a totally unsupervised fashion. Using the three consensus heuristics, we observed how the Average Normalized Mutual Information (ANMI) (Section 3.4.5.1.1) between the consensus and the base solutions varied with the number of clusters requested. These measurements are shown in Figure 6.10. We then evaluated more carefully the consensus solution with the highest ANMI, that is, the one which best agreed with the set of base clusterings. CSPA was the winning heuristic for both synthetic datasets, while MCLA achieved the highest ANMI for dataset R1. Interestingly, for datasets S1 and S2 the ANMI peak coincided with the true number of clusters ($K = 2$), which is what we would ideally expect. For dataset R1, the ANMI peak was found at $k = 6$, closely followed by the value at $k = 5$. Consensus solutions with similar ANMI values like these mean that the partitionings with higher number of clusters simply split existing groups into smaller ones, without affecting the remaining parti-

tions significantly. In this case, the solution with six clusters is equivalent to the one with five clusters, only with one of its groups subdivided into two new groups. It is also remarkable how the ANMI peak for dataset R1 is much lower (0.1881) than for datasets S1 and S2 (0.6791 and 0.6319, respectively). This indicates that the set of base clusterings for dataset R1 is more diverse, that is, the individual algorithms came up with more "conflicting opinions" about the data clusters than those obtained for datasets S1 and S2. Consequently this is evidence that unsupervised structural classification on real cryo-EM datasets can be quite challenging.





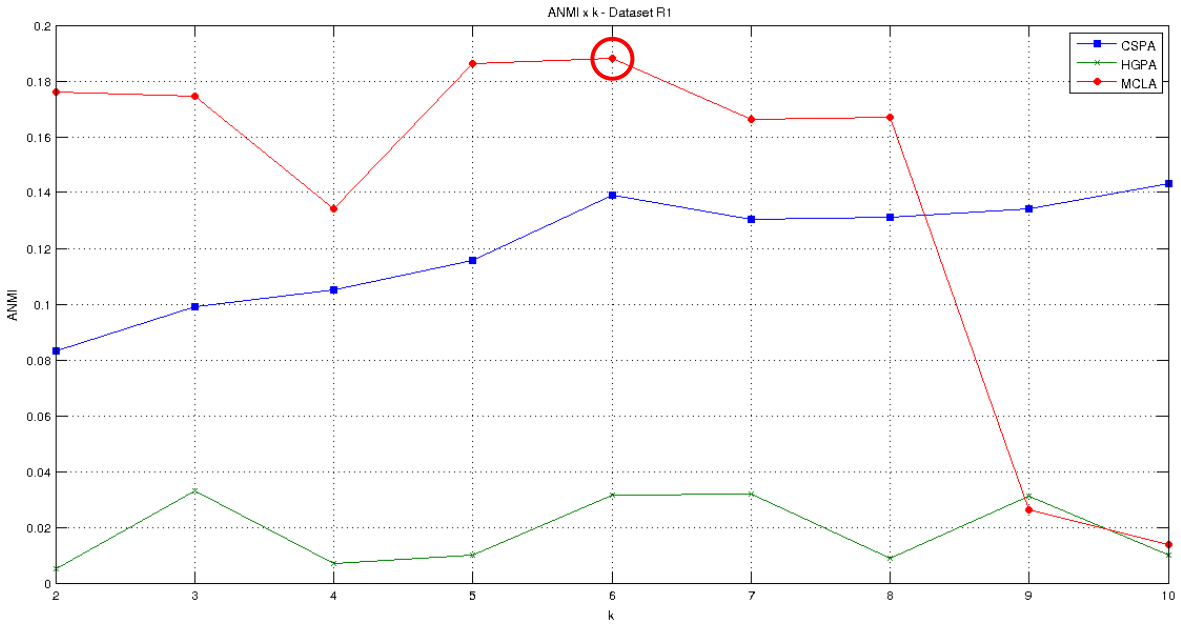


Figure 6.10 – The variation of the ANMI between the consensus and the base solutions, for datasets S1, S2 and R1. Measurements for heuristics CSPA, HGPA and MCLA are reported. The ANMI peak for each dataset is highlighted by a circle.

Table 6.5 shows the confusion matrices for the solutions obtained from the ANMI peak on each dataset. For R1, we see that, although the total number of clusters found was six, only two of them were densely populated. Two of them were empty (clusters 2 and 3), which is an issue that can happen with the MCLA algorithm when the meta-clusters compete for objects (Section 3.4.5.1.6). Small meta-clusters, i.e., those with only a few objects, tend to lose their objects to bigger ones. Besides these two empty clusters, we see another two that are barely populated, clusters 4 and 5. Regarding conformational classification, such small clusters could be discarded without prejudice of the reconstruction procedure. This is why they were not taken into account when assessing the total classification accuracy on this dataset. Therefore, the solution obtained from the ANMI peak for dataset R1 produced only two valid clusters, which coincides with the true number of conformations expected, and the classification accuracy was similar to that obtained when we requested only two clusters (Table 6.4), although slightly lower.

Table 6.5 – Confusion matrices for the consensus solutions given by the ANMI peak on each dataset, Experiment 2. The average purity for dataset R1 was calculated discarding clusters 4 and 5 which were barely populated.

S1

class \ *cluster*	1	2	*sum*
open	0	10,000	10,000
closed	10,000	0	10,000
sum	10,000	10,000	20,000
purity [%]	100	100	
average [%]		100	

S2

class \ *cluster*	1	2	*sum*
open	9,997	3	10,000
closed	1	9,999	10,000
sum	9,998	10,002	20,000
purity [%]	99.99	99.97	
average [%]		99.98	

R1

class \ *cluster*	1	2	3	4	5	6	*sum*
open	341	0	0	114	5	4,540	5,000
closed	2,550	0	0	0	0	2,450	5,000
sum	2,891	0	0	114	5	6,990	10,000
purity [%]	88.20	-	-	100	100	64.95	
*average** [%]						76.58	

Figure 6.11 shows the ensemble solution obtained by the ANMI peak for dataset R1 projected along its first two principal components, in comparison to the true classes. Clearly, it can be seen that the cluster corresponding to the "closed" state ended up approximately split in half between a pure cluster and the cluster mostly associated with the "open" class.

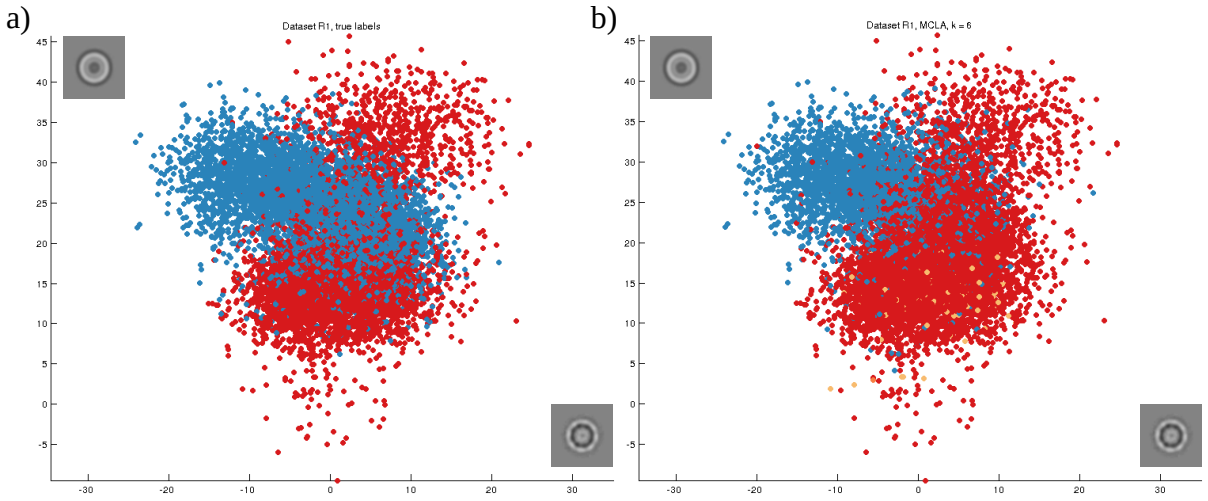


Figure 6.11 – Dataset R1 projected along its first two principal components (shown in details). a) True labels. Red corresponds to images from the "open" state, blue corresponds to images from the "closed" state. b) Labels assigned by the MCLA consensus heuristic with six clusters. Two clusters are empty and two others are barely populated, colored in orange.

Another interesting aspect to be observed from this experiment is the behavior of "compactness"-based algorithms, like k-means, against "connectedness"-based algorithms, like spectral clustering. Especially for dataset S1, in which the conformational manifolds can be readily recognized along the principal components, we can see the different interpretation these two algorithms take on the data distribution, as shown in Figure 6.12. While k-means could achieve a high classification accuracy, its solution is very different from that obtained by spectral clustering, which achieved perfect classification. The cluster convexity requirement intrinsically assumed by k-means prevents it from detecting cluster distributions of arbitrary shape (Section 3.4.4.2). With this example, we illustrate the importance of plotting the data distribution when assessing clustering performance.

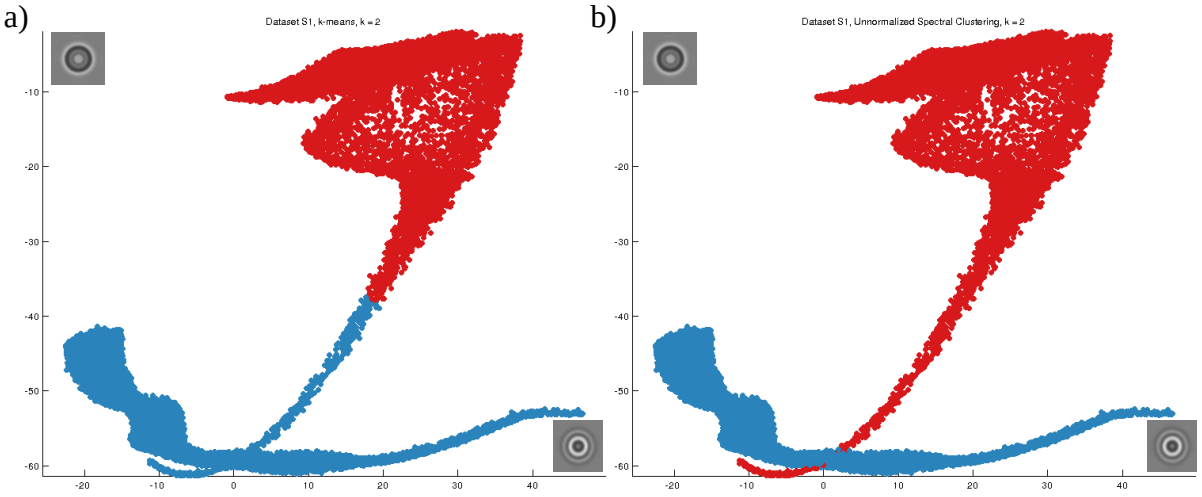


Figure 6.12 - Dataset S1 projected along its first two principal components (shown in details), clustered in two groups. a) Labels provided by k-means, which have a 97.43% matching with the ground truth. b) Labels assigned by the unsupervised spectral clustering algorithm, which have a 100% matching with the ground truth. Red corresponds to images from the "open" state, blue corresponds to images from the "closed" state.

Finally, we would like to draw attention to the performance of the HGPA consensus heuristic. Table 6.4 shows it performed poorly when requesting two clusters, achieving an essentially random classification solution for all datasets. However, if we look at Figure 6.10, we see that HGPA had its ANMI peak at $k = 3$ for all three datasets, and such peak was very close to the ANMI values achieved by the CSPA and MCLA heuristics with three clusters, except in the case of dataset R1. We then decided to plot the HGPA solution with three clusters along the principal components, in order to gain insight on what happened. Figure 6.13 displays this plot for dataset S2. While we see that the "closed" class (blue dots) was essentially correctly recognized by HGPA, the "open" class (red and green dots) became split in two clusters, and such division seems to be random. Therefore, when requested to provide three clusters, HGPA provided an essentially correct solution for two classes, if we consider the union of both clusters corresponding to the "open" class. However, such correspondence may be difficult to verify in practice. This issue is probably caused by the hypergraph partitioning restrictions found by HGPA, as explained in Section 3.4.5.1.5.

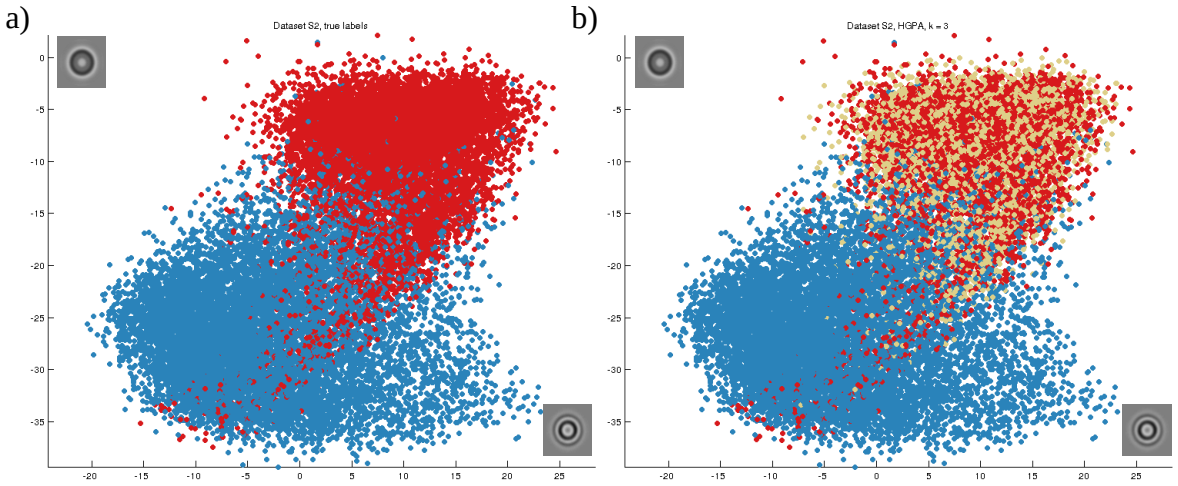


Figure 6.13 - Dataset S2 projected along its first two principal components (shown in details). a) True labels. Red corresponds to images from the "open" state, blue corresponds to images from the "closed" state. b) Labels assigned by the HGPA consensus heuristic with three clusters. The third cluster is labeled green.

6.2.4 Experiment 3: The influence of alignment quality

The third and last experiment we conducted aimed at observing the influence of the alignment quality on the detection of the conformational clusters. We also assessed the robustness of the ensemble solutions. The two synthetic and the four datasets were analyzed, as they represent different data quality conditions or different stages of the reconstruction procedure. In this context, we assumed that the cluster ensemble is being employed to validate the structural assignments obtained from a conventional reconstruction procedure. Therefore, the number of clusters requested in all solutions is the true number of conformations expected ($K = 2$). Six clustering algorithms were used to provide the base solutions: HAC, k-means, SOM, Gaussian Mixture Model (GMM), unsupervised spectral clustering and METIS. The consensus solutions were provided by four algorithms: k-means, CSPA, HGPA and MCLA. The clustering algorithms analyzed the datasets using 10 and 100 principal components, and in each case 10 runs of each algorithm were executed to account for randomized initializations. The mean and standard deviation of the matches with the true labels are reported in Table 6.6 and Table 6.7, respectively.

Table 6.6 – Performance of the cluster ensemble from Experiment 3 on the six analyzed datasets, using the first 10 principal components. Mean and standard deviation of the classification accuracy for 10 runs are reported. The best performing algorithm is highlighted in bold, both among the base solutions and among the consensus solutions.

%	HAC	k-means	SOM	GMM	Spectral	METIS	k-means	CSPA	HGPA	MCLA
S1	0.9825 ± 0.0000	0.9723 ± 0.0000	0.9080 ± 0.1227	0.9929 ± 0.0000	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9868 ± 0.0052	0.9869 ± 0.0032	0.4990 ± 0.0000	**0.9930 ± 0.0035**
S2	0.9613 ± 0.0000	0.9199 ± 0.0000	0.7938 ± 0.1694	0.9509 ± 0.0001	0.9775 ± 0.0000	**0.9789 ± 0.0000**	0.9715 ± 0.0003	**0.9738 ± 0.0041**	0.4993 ± 0.0000	0.9674 ± 0.0025
R1	0.6576 ± 0.0000	0.6928 ± 0.0005	0.4807 ± 0.1237	0.7123 ± 0.0891	0.6500 ± 0.0000	**0.7538 ± 0.0000**	0.6980 ± 0.0165	**0.7513 ± 0.0066**	0.5000 ± 0.0000	0.6646 ± 0.0111
R2	0.8569 ± 0.0000	0.8549 ± 0.0000	0.7682 ± 0.1030	0.8168 ± 0.0779	**0.8632 ± 0.0000**	0.7753 ± 0.0000	**0.8666 ± 0.0028**	0.7957 ± 0.0191	0.5000 ± 0.0000	0.8657 ± 0.0030
R3	0.8599 ± 0.0000	0.8613 ± 0.0000	0.8089 ± 0.0979	0.8651 ± 0.0386	**0.8688 ± 0.0000**	0.7878 ± 0.0000	**0.8712 ± 0.0015**	0.7956 ± 0.0223	0.5000 ± 0.0000	0.8693 ± 0.0033
R4	0.8883 ± 0.0000	0.8641 ± 0.0000	0.8569 ± 0.0266	0.7731 ± 0.0901	0.8540 ± 0.0000	**0.9858 ± 0.0000**	0.8725 ± 0.0055	**0.9518 ± 0.0252**	0.5000 ± 0.0000	0.8767 ± 0.0054

Table 6.7 - Performance of the cluster ensemble from Experiment 3 on the six analyzed datasets, using the first 100 principal components. Mean and standard deviation of the classification accuracy for 10 runs are reported. The best performing algorithm is highlighted in bold, both among the base solutions and among the consensus solutions. In case there is a tie between the mean performances, the solution with the smallest dispersion is declared to be the best.

%	HAC	k-means	SOM	GMM	Spectral	METIS	k-means	CSPA	HGPA	MCLA
S1	0.9857 ± 0.0000	0.9743 ± 0.0000	0.9514 ± 0.0389	0.9411 ± 0.1863	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9915 ± 0.0073	0.9972 ± 0.0059	0.4990 ± 0.0000	**0.9991 ± 0.0030**
S2	0.9179 ± 0.0000	0.9180 ± 0.0001	0.7289 ± 0.1750	0.9822 ± 0.0003	0.9991 ± 0.0000	**0.9995 ± 0.0000**	**0.9841 ± 0.0003**	0.9841 ± 0.0006	0.4993 ± 0.0000	0.9589 ± 0.0069
R1	0.6314 ± 0.0000	0.6821 ± 0.0020	0.5472 ± 0.1421	**0.7425 ± 0.0899**	0.4419 ± 0.0000	0.6490 ± 0.0000	0.6926 ± 0.0089	**0.7233 ± 0.0231**	0.5000 ± 0.0000	0.6334 ± 0.0647
R2	0.8534 ± 0.0000	**0.8613 ± 0.0001**	0.8145 ± 0.0306	0.6411 ± 0.2079	0.8427 ± 0.0000	0.7713 ± 0.0000	0.8617 ± 0.0015	0.7860 ± 0.0057	0.5000 ± 0.0000	**0.8643 ± 0.0017**
R3	0.8613 ± 0.0000	0.8618 ± 0.0000	0.7609 ± 0.1761	0.6255 ± 0.2191	0.8518 ± 0.0000	**0.8982 ± 0.0000**	0.8673 ± 0.0014	0.8633 ± 0.0344	0.5000 ± 0.0000	**0.8678 ± 0.0041**
R4	0.8795 ± 0.0000	0.8637 ± 0.0000	0.8333 ± 0.0694	0.7360 ± 0.1081	0.8612 ± 0.0000	**0.9837 ± 0.0000**	0.8698 ± 0.0025	**0.9396 ± 0.0455**	0.5000 ± 0.0000	0.8748 ± 0.0030

However, the interpretation of unsupervised classification performance by means of the fraction of matches with the ground truth can be misleading, due to the cluster correspondence problem (Section 3.4.3). Even if the canonical labeling convention is used (Section 3.4.2.1), the solutions being compared are still subject to the assignment of the first object in the list. The label of the first object is always "1", but the remaining objects belonging to cluster "1" may be very different in solutions $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$. That is, cluster "1" in $\boldsymbol{\lambda}^a$ may not correspond to cluster "1" in $\boldsymbol{\lambda}^b$. For this reason, we also evaluated the performance of the algorithms in this experiment using label-independent clustering similarity measures.

Table 6.8 and Table 6.9 display the performance using the Normalized Mutual Information (NMI) (Section 3.4.5.1.2.2) for 10 and 100 principal components, respectively. The equivalent results using the Adjusted Rand Index (ARI) (Section 3.4.5.1.2.1) are contained in Table 6.10 (10 principal components) and Table 6.11 (100 principal components).

Table 6.8 - Performance of the cluster ensemble from Experiment 3 on the six analyzed datasets, using the first 10 principal components. Mean and standard deviation of the Normalized Mutual Information with the ground truth for 10 runs are reported. The best performing algorithm is highlighted in bold, both among the base solutions and among the consensus solutions.

NMI	HAC	k-means	SOM	GMM	Spectral	METIS	k-means	CSPA	HGPA	MCLA
S1	0.8901 ± 0.0000	0.8318 ± 0.0001	0.6581 ± 0.3016	0.9462 ± 0.0000	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9133 ± 0.0283	0.9128 ± 0.0176	0.0000 ± 0.0000	**0.9483 ± 0.0237**
S2	0.7865 ± 0.0000	0.5996 ± 0.0000	0.3659 ± 0.1687	0.7238 ± 0.0007	0.8509 ± 0.0000	**0.8595 ± 0.0000**	0.8231 ± 0.0013	**0.8314 ± 0.0219**	0.0000 ± 0.0000	0.8120 ± 0.0096
R1	0.0870 ± 0.0000	0.1184 ± 0.0005	0.0415 ± 0.0276	0.1580 ± 0.0665	0.0819 ± 0.0000	**0.1948 ± 0.0000**	0.1244 ± 0.0175	**0.1909 ± 0.0105**	0.0000 ± 0.0000	0.0932 ± 0.0090
R2	0.4362 ± 0.0000	0.4326 ± 0.0000	0.2593 ± 0.1315	0.3733 ± 0.1864	**0.5024 ± 0.0000**	0.2313 ± 0.0000	**0.4842 ± 0.0175**	0.2712 ± 0.0409	0.0000 ± 0.0000	0.4781 ± 0.0162
R3	0.4566 ± 0.0000	0.4518 ± 0.0001	0.3385 ± 0.1194	0.4911 ± 0.1066	**0.5139 ± 0.0000**	0.2543 ± 0.0000	**0.4996 ± 0.0107**	0.2715 ± 0.0496	0.0000 ± 0.0000	0.4931 ± 0.0133
R4	0.5645 ± 0.0000	0.5007 ± 0.0000	0.4418 ± 0.0738	0.2875 ± 0.2449	0.5059 ± 0.0000	**0.8925 ± 0.0000**	0.5379 ± 0.0108	**0.7303 ± 0.1069**	0.0000 ± 0.0000	0.5451 ± 0.0109

Table 6.9 - Performance of the cluster ensemble from Experiment 3 on the six analyzed datasets, using the first 100 principal components. Mean and standard deviation of the Normalized Mutual Information with the ground truth for 10 runs are reported. The best performing algorithm is highlighted in bold, both among the base solutions and among the consensus solutions.

NMI	HAC	k-means	SOM	GMM	Spectral	METIS	k-means	CSPA	HGPA	MCLA
S1	0.9060 ± 0.0000	0.8527 ± 0.0000	0.7540 ± 0.1564	0.9023 ± 0.3089	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9442 ± 0.0480	0.9815 ± 0.0390	0.0000 ± 0.0000	**0.9932 ± 0.0214**
S2	0.6608 ± 0.0000	0.5930 ± 0.0004	0.2608 ± 0.2282	0.8816 ± 0.0016	0.9898 ± 0.0000	**0.9941 ± 0.0000**	**0.8952 ± 0.0013**	0.8939 ± 0.0044	0.0000 ± 0.0000	0.7918 ± 0.0246
R1	0.0662 ± 0.0000	0.1048 ± 0.0019	0.0602 ± 0.0464	**0.2098 ± 0.0840**	0.1011 ± 0.0000	0.0650 ± 0.0000	0.1149 ± 0.0098	**0.1508 ± 0.0320**	0.0000 ± 0.0000	0.0811 ± 0.0060
R2	0.4644 ± 0.0000	0.4537 ± 0.0002	0.3176 ± 0.0631	0.2286 ± 0.1549	**0.4735 ± 0.0000**	0.2243 ± 0.0000	0.4617 ± 0.0038	0.2511 ± 0.0108	0.0000 ± 0.0000	**0.4888 ± 0.0081**
R3	0.4485 ± 0.0000	0.4559 ± 0.0002	0.3073 ± 0.0998	0.2340 ± 0.1786	0.4885 ± 0.0000	**0.5253 ± 0.0000**	0.4814 ± 0.0113	0.4308 ± 0.0894	0.0000 ± 0.0000	**0.4948 ± 0.0109**
R4	0.5344 ± 0.0000	0.5006 ± 0.0000	0.3978 ± 0.1263	0.2130 ± 0.1821	0.5136 ± 0.0000	**0.8799 ± 0.0000**	0.5265 ± 0.0048	**0.6888 ± 0.1454**	0.0000 ± 0.0000	0.5349 ± 0.0055

Table 6.10 - Performance of the cluster ensemble from Experiment 3 on the six analyzed datasets, using the first 10 principal components. Mean and standard deviation of the Adjusted Rand Index with the ground truth for 10 runs are reported. The best performing algorithm is highlighted in bold, both among the base solutions and among the consensus solutions.

ARI	HAC	k-means	SOM	GMM	Spectral	METIS	k-means	CSPA	HGPA	MCLA
S1	0.9312 ± 0.0000	0.8922 ± 0.0001	0.7199 ± 0.3130	0.9718 ± 0.0000	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9481 ± 0.0204	0.9483 ± 0.0124	0.0000 ± 0.0000	**0.9722 ± 0.0140**
S2	0.8512 ± 0.0000	0.7051 ± 0.0000	0.4487 ± 0.1933	0.8132 ± 0.0003	0.9118 ± 0.0000	**0.9172 ± 0.0000**	0.8891 ± 0.0011	**0.8979 ± 0.0156**	0.0000 ± 0.0000	0.8739 ± 0.0092
R1	0.0993 ± 0.0000	0.1486 ± 0.0008	0.0565 ± 0.0375	0.2087 ± 0.0879	0.0899 ± 0.0000	**0.2576 ± 0.0000**	0.1577 ± 0.0266	**0.2526 ± 0.0132**	-0.0001 ± 0.0000	0.1087 ± 0.0148
R2	0.5095 ± 0.0000	0.5038 ± 0.0000	0.3259 ± 0.1628	0.4234 ± 0.1908	**0.5276 ± 0.0000**	0.3031 ± 0.0000	**0.5376 ± 0.0083**	0.3511 ± 0.0481	-0.0001 ± 0.0000	0.5349 ± 0.0087
R3	0.5181 ± 0.0000	0.5221 ± 0.0001	0.4161 ± 0.1461	0.5385 ± 0.1119	**0.5440 ± 0.0000**	0.3312 ± 0.0000	**0.5510 ± 0.0044**	0.3513 ± 0.0576	-0.0001 ± 0.0000	0.5455 ± 0.0097
R4	0.6031 ± 0.0000	0.5302 ± 0.0000	0.5121 ± 0.0759	0.3274 ± 0.2394	0.5012 ± 0.0000	**0.9440 ± 0.0000**	0.5550 ± 0.0165	**0.8188 ± 0.0900**	-0.0001 ± 0.0000	0.5676 ± 0.0162

Table 6.11 - Performance of the cluster ensemble from Experiment 3 on the six analyzed datasets, using the first 100 principal components. Mean and standard deviation of the Adjusted Rand Index with the ground truth for 10 runs are reported. The best performing algorithm is highlighted in bold, both among the base solutions and among the consensus solutions.

ARI	HAC	k-means	SOM	GMM	Spectral	METIS	k-means	CSPA	HGPA	MCLA
S1	0.9436 ± 0.0000	0.8998 ± 0.0000	0.8206 ± 0.1325	0.9032 ± 0.3062	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9666 ± 0.0288	0.9890 ± 0.0233	0.0000 ± 0.0000	**0.9963 ± 0.0118**
S2	0.6984 ± 0.0000	0.6990 ± 0.0004	0.3198 ± 0.2745	0.9300 ± 0.0012	0.9962 ± 0.0000	**0.9980 ± 0.0000**	**0.9375 ± 0.0010**	0.9373 ± 0.0024	0.0000 ± 0.0000	0.8425 ± 0.0253
R1	0.0690 ± 0.0000	0.1326 ± 0.0029	0.0815 ± 0.0623	**0.2643 ± 0.0978**	0.0135 ± 0.0000	0.0887 ± 0.0000	0.1486 ± 0.0138	**0.2013 ± 0.0412**	-0.0001 ± 0.0000	0.0861 ± 0.0274
R2	0.4995 ± 0.0000	**0.5221 ± 0.0002**	0.3990 ± 0.0717	0.2352 ± 0.1484	0.4697 ± 0.0000	0.2943 ± 0.0000	0.5233 ± 0.0043	0.3273 ± 0.0132	-0.0001 ± 0.0000	**0.5307 ± 0.0049**
R3	0.5221 ± 0.0000	0.5236 ± 0.0001	0.3839 ± 0.1171	0.2357 ± 0.1795	0.4950 ± 0.0000	**0.6342 ± 0.0000**	0.5397 ± 0.0041	0.5321 ± 0.0978	-0.0001 ± 0.0000	**0.5412 ± 0.0119**
R4	0.5760 ± 0.0000	0.5291 ± 0.0000	0.4615 ± 0.1398	0.2647 ± 0.1983	0.5218 ± 0.0000	**0.9359 ± 0.0000**	0.5470 ± 0.0075	**0.7803 ± 0.1426**	-0.0001 ± 0.0000	0.5617 ± 0.0088

Some interesting observations can be made regarding the different cluster similarity measures. Firstly, it can be seen that the k-means algorithm achieved the best individual performance for dataset R2 with 100 principal components according to the percentage of label matches with the ground truth (Table 6.7) and also according to the ARI (Table 6.11), but, according to the NMI measure (Table 6.9), the best algorithm for this dataset was spectral clustering. Thus, it can be seen that the cluster correspondence problem is present in the analyses of Table 6.6 and Table 6.7. Also, while the k-means and CSPA consensus solutions had the same average performance for dataset S2 with 100 components in Table 6.7, according to NMI (Table 6.9) and to

ARI (Table 6.11) the k-means average performance was slightly higher than that of CSPA. We see also that the performance of the HGPA consensus heuristic was practically zero according to NMI (Table 6.8 and Table 6.9) and to ARI (Table 6.10 and Table 6.11), but by the fraction of matches with the ground truth (Table 6.6 and Table 6.7) it had about 50% accuracy. This means that the results of HGPA for solutions with two clusters are essentially random, an issue we had already observed in Experiment 2 (Section 6.2.3). Still regarding the performance of HGPA, we draw attention to some close-to-zero negative ARI values in Table 6.10 and in Table 6.11, which are meaningless for practical effects (Section 3.4.5.1.2.1). Despite avoiding the cluster correspondence problem, a difficulty in interpreting clustering similarity with NMI and ARI is their non-linear behavior in relation to the fraction of matching labels between two solutions, as shown in Figure 6.14. This plot may aid the interpretation of Tables Table 6.6 through Table 6.11. In this figure, a percentage of zero matches between two clustering solutions $\boldsymbol{\lambda}^a$ and $\boldsymbol{\lambda}^b$means that the cluster labels are inverted between them. For NMI and ARI, both solutions are identical. For simplicity, let's assume that $\boldsymbol{\lambda}^b$ is a fixed solution in which the objects are equally distributed in two clusters. As the objects have their assignments changed to the other cluster in solution $\boldsymbol{\lambda}^a$, the fraction of matches increases up to 50%, which indicates that all of the objects belong to the same cluster in $\boldsymbol{\lambda}^a$. By changing the assignments of the second cluster in $\boldsymbol{\lambda}^a$, the percentage of matches keeps increasing up to 100%, when $\boldsymbol{\lambda}^a$ becomes identical to $\boldsymbol{\lambda}^b$ in terms of label convention.

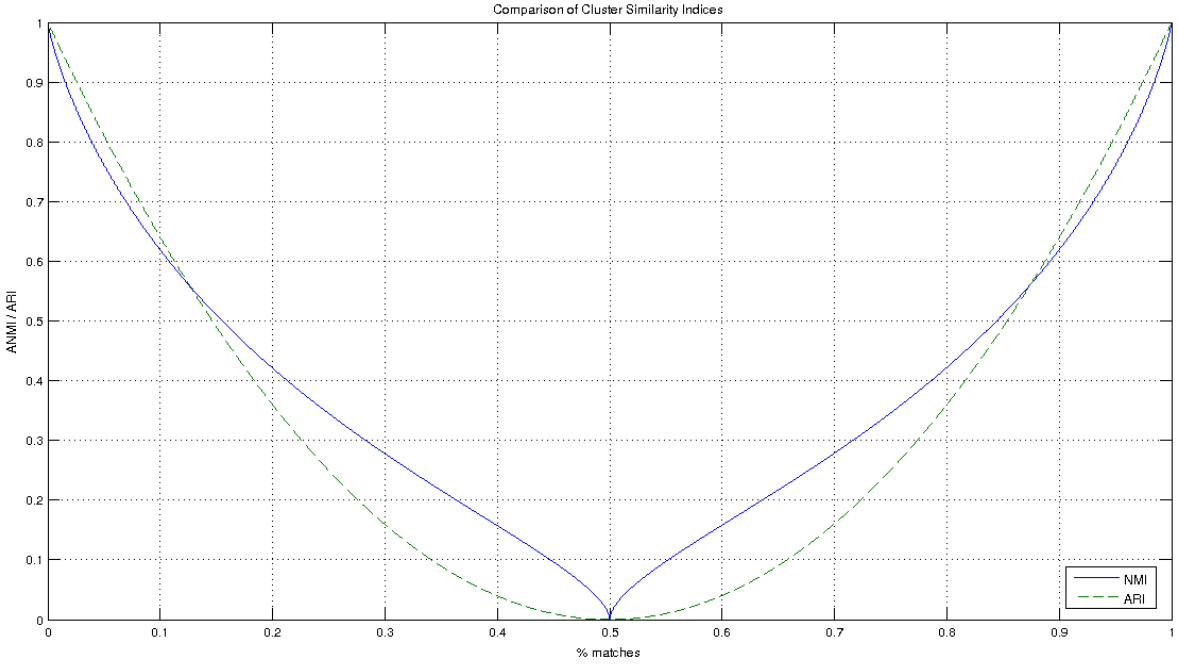


Figure 6.14 – Comparison of cluster similarity indices NMI (blue continuous line) and ARI (green dashed line) against the percentage of matching labels between two solutions containing two clusters.

Apart from cluster similarity indices, the central aspects of Experiment 3 can be better assessed by plotting the classification performances. To this end, we chose to plot the fraction of matching labels between each solution obtained and the ground truth, due to its ease of interpretation. However, as we want to address relative performance and robustness, NMI and ARI would also serve. Figure 6.15 shows the performance plots for each dataset using 10 principal components. Figure 6.16 is the equivalent for 100 principal components.

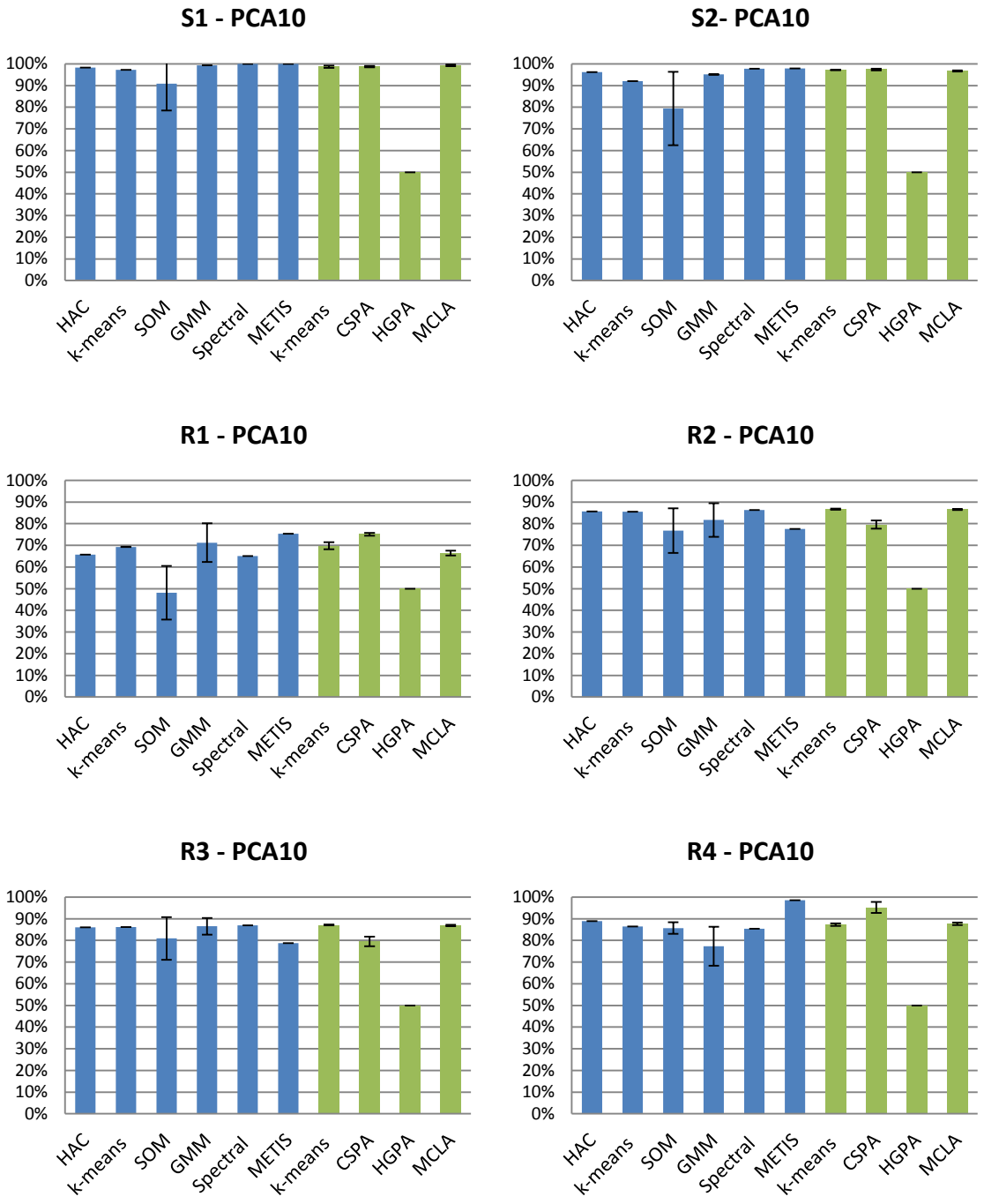


Figure 6.15 – Plots of the average classification accuracy on the six datasets using 10 principal components in Experiment 3 (Table 6.6). Base solutions are shown in blue and consensus solutions are shown in green. Error bars correspond to ± 1 standard deviation from 10 runs of each algorithm.

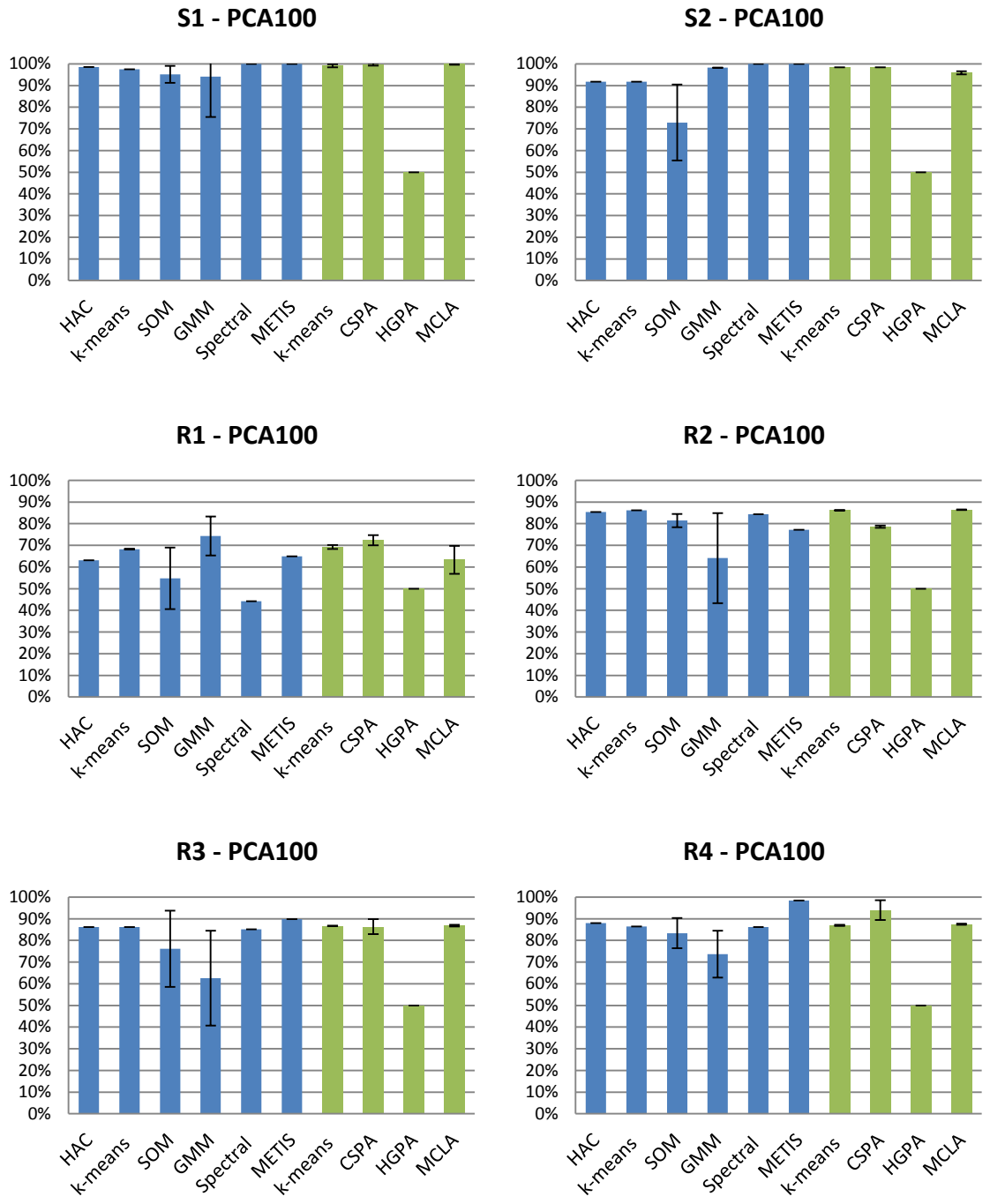


Figure 6.16 - Plots of the average classification accuracy on the six datasets using 100 principal components in Experiment 3 (Table 6.7). Base solutions are shown in blue and consensus solutions are shown in green. Error bars correspond to ± 1 standard deviation from 10 runs of each algorithm.

The careful analysis of Figure 6.15 and Figure 6.16 lead us to several interesting observations. The first of them is that in all cases the consensus solutions provided classification accuracy comparable to the best individual algorithm, or better. The exception is the HGPA heuristic,

which requires a more elaborate approach when specifying the number of clusters (see Experiment 2, Figure 6.13) and for this reason will be ignored in this analysis. Also, the best individual performances were not always provided by the same algorithm, even though graph partitioning approaches seem to be better suited to this task. Therefore, the ensemble approach is justified by providing a solution robust to variations across particular choices of clustering algorithms. This can also be supported by the additional observation that the dispersion of consensus performances is much smaller than that of certain individual solutions, most notably those provided by the SOM and the GMM algorithms, which have many parameters and are highly dependent on their initializations. Perhaps surprisingly, the simple consensus approach using the k-means algorithm yielded solutions comparable to those from more sophisticated heuristics like CSPA and MCLA on our datasets. Even though we cannot determine an absolute winner among these three consensus approaches, we would recommend the use of k-means or MCLA because of their linear computational complexity, in contrast to the quadratic complexity of CSPA (see Section 3.4.5.1.3). Also, the difference in performance between using 10 or 100 principal components is negligible, even though only a small fraction of a dataset's variance is concentrated on the first 10 principal components (Table 6.1). There are two possible and complementary explanations for this. The first one is related to the curse of dimensionality (Section 3.3): although 100 principal components provide a better statistical representation of the data, it renders the search space exponentially larger when finding an optimal clustering solution. The other one is related to the own nature of the data investigated: as they have extremely low SNRs (except for dataset S1), practically all the relevant information is concentrated on the first principal components, while the remainder of them mostly describe random noise, as previously observed in Section 6.1. Therefore, the inclusion of more components does not provide useful information about the data clusters, and their noisy nature may even confuse the clustering algorithms. In addition to common practices in single particle analysis (Frank, 2006; van Heel *et al.*, 2000), we observe that no more than 10 principal components should be enough to obtain a reasonably good clustering on most cryo-EM datasets.

Regarding the real datasets, it can be seen from Figure 6.15 and Figure 6.16 that the improvement on the alignment quality clearly allows more accurate unsupervised classification. There is a remarkable improvement in the classification of dataset R2, which was translationally aligned with re-projections of 3D models, in relation to dataset R1, which was just centered

against the dataset's average image. Then there is a slight overall improvement in dataset R3 when compared to dataset R2; dataset R3 had its principal components determined with artificially introduced rotated versions of the images, emulating a situation in which more data is available. Finally, when complete alignment information (translation and rotation) becomes available, as in dataset R4, clustering algorithms may achieve very high classification performance, with an outstanding 98.58% accuracy obtained by the METIS algorithm using 10 principal components. This result is even more impressive if we see that the conformational clusters are not separable when observed along the first principal components, as shown in Figure 6.17. Nevertheless, it is important to acknowledge that all classification results are biased by the translational and/or rotational alignment steps performed previously. Considering the results obtained by spectral clustering in Experiments 2 and 3, it can also be seen that graph partitioning algorithms are powerful tools in recognizing conformational heterogeneity in cryo-EM datasets.

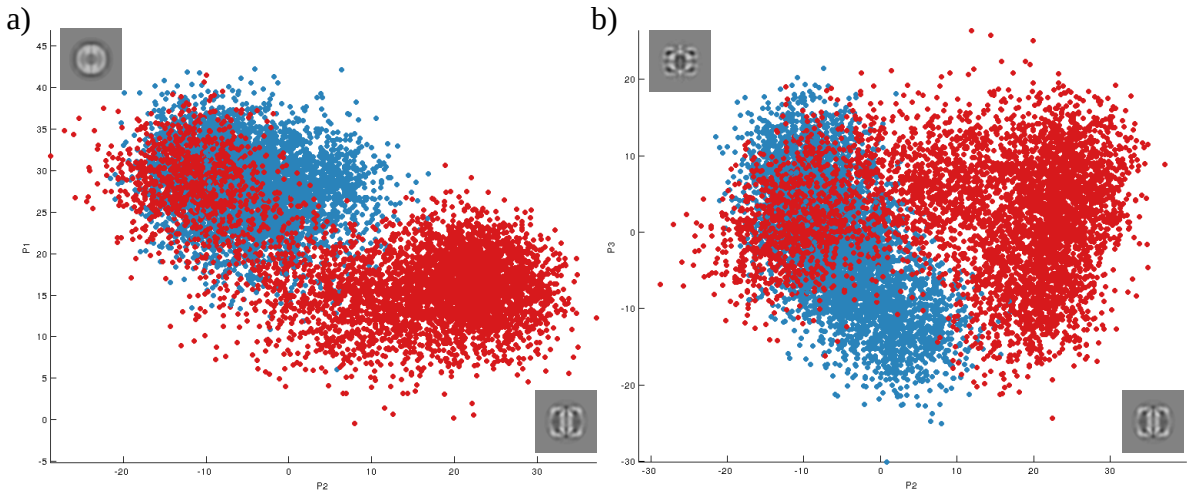


Figure 6.17 - Dataset R4 projected along selected principal components (shown in details), clustered in two groups by METIS using a similarity matrix constructed from 10 principal components. a) Plot along the first and second principal components. b) Plot along the second and third principal components. Red points correspond to images from the "open" state, blue points correspond to images from the "closed" state. This clustering solution has 98.58% matching in relation to the true conformational labels.

Finally, we would like to demonstrate the use of the inherent clustering confidence measure provided by the MCLA heuristic. The competitive stage of this heuristic, in which the meta-clusters dispute objects, allows the observation of how "strong" the label assignment is for a given object. If this object appears in a meta-cluster much more often than in the others, the confidence of this assignment will be high; if it appears with approximately the same frequency in two or more meta-clusters, the confidence will be low. As an example, we illustrate the confidence of

the assignments made by MCLA in dataset R4, using 10 principal components, shown in Figure 6.18. From these plots, we see that a small portion (about 3%) of the data has a very low confidence on their cluster assignments (data points colored towards red). Therefore, in a single particle reconstruction procedure, the user could discard these data as they probably represent low quality images. Interestingly, the data with lowest confidence estimates lie on regions of the multidimensional space that overlap the two conformational manifolds, depending on the direction of observation. In other words, these are data for which the base clusterings cannot agree unanimously on their labels.

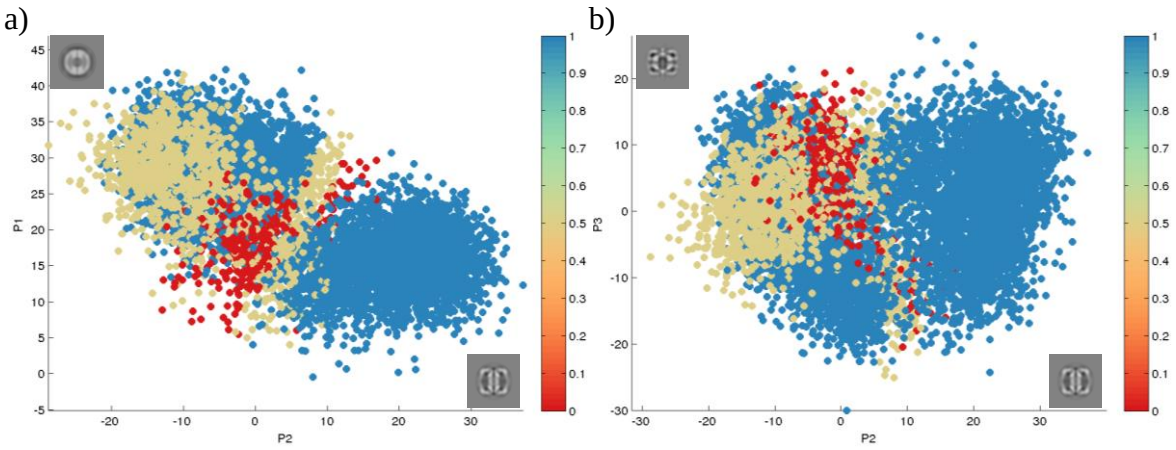


Figure 6.18 – The cluster assignment confidence from one run of MCLA on dataset R4 using 10 principal components. The dataset is shown projected along selected principal components (shown in details). The confidence is normalized within each cluster such that 1 is the highest and 0 is the lowest. a) Plot along the first and second principal components. b) Plot along the second and third principal components.

7. Conclusions

This dissertation presented the problem of structural heterogeneity in the study of macromolecular assemblies by transmission electron microscopy. More specifically, we aimed at separating the electron microscopy data according to the structural conformation represented in the image by means of unsupervised classification algorithms.

In the last decades, the transmission electron microscope has become an invaluable instrument in structural biology. In comparison to the other well established techniques in this field, namely X-ray crystallography and nuclear magnetic resonance, transmission electron microscopy does not require the crystallization of the particles, and allows the observation of relatively large structures. This makes it suitable for investigating the mechanisms of large molecular assemblies, like protein complexes and other cellular machinery, such as the ribosome.

In order to prevent damage by the radiation beam, the dose has to be lowered and the samples must be embedded in negative stain or vitreous ice. The latter method is commonly referred to as electron cryo-microscopy, or cryo-EM, and is the preferred sample preservation method for achieving high resolution 3D reconstructions, in the order of only a few Angstroms. A three-dimensional reconstruction of the structure density map can be obtained by iteratively estimating the projection direction, called Euler angles, for each of the collected images. This process is referred to as single particle reconstruction.

However, as the signal-to-noise ratio of the images is low, large datasets are required in order to obtain reasonable reconstructions, typically containing tens of thousands of images. Usually, projection images containing similar views of the structure are averaged in order to improve the SNR, in a process called 2D classification.

As molecular assemblies are flexible structures, they may assume different conformations while performing their function in the organisms. Such structural heterogeneity may prevent the reconstruction of reliable models by cryo-EM. Therefore, the set of images must be classified according to the structural configuration of the particle projected, in a process called 3D classification.

Several methods have been proposed to this end, using multivariate statistical analysis, maximum-likelihood and Bayesian statistical modeling, graph representations, and, more recently, manifold learning approaches. However, all of the currently available 3D classification methods involve the iterative reconstruction of the heterogeneous models to some degree.

Based on the assumption that projection data from distinct conformations lie on different manifolds in a multidimensional space, we used unsupervised classification techniques to detect such manifolds or "conformational clusters" without the need of performing 3D reconstructions. Such approach may be useful for initial partitioning of the datasets, as shown in Figure 1.1, or to validate conformational assignments obtained by conventional reconstruction procedures.

Unsupervised learning is an area that is heavily grounded on the fields of statistics and machine learning, and has provided useful tools to many other knowledge domains. Dimensionality reduction techniques may reduce the computational efforts of unsupervised learning tasks, besides supporting exploratory data visualization. We have mainly used Principal Component Analysis, due to its longtime demonstrated suitability to cryo-EM datasets.

Many unsupervised algorithms have been proposed in the machine learning community, and we have presented and employed six of them: k-means clustering, hierarchical clustering, Gaussian mixture models, self-organizing maps, spectral clustering and METIS. These algorithms make different assumptions on the distribution of the data clusters. For example, the first four algorithms assume that data clusters are compact, although with different nuances among them, while spectral clustering and METIS employ graph-based representations of the data in order to partition clusters of arbitrary shape. Also, cluster assignments may be overlapping or not, and data points may have different cluster membership degrees, as is the case when clustering is performed by a mixture of Gaussian distributions.

In order to obtain a more accurate classification solution in cryo-EM datasets, we employed an ensemble of clustering solutions. Cluster ensembles tend to provide more robustness and higher accuracy than individual clustering algorithms. They are also useful to integrate and reuse distributed knowledge about the data labels. In general, a consensus solution is obtained by maximizing an agreement measure throughout the set of base clusterings.

We have used three efficient heuristics to obtain consensus among the results from the individual algorithms mentioned above: CSPA, HGPA and MCLA. These heuristics transform the cluster ensemble problem into a (hyper)graph partitioning problem. This representation has the

advantages of avoiding the cluster correspondence problem, as well as not requiring access to the original data features. Additionally, we have used the k-means algorithm applied directly to the set of base labels in order to obtain a single, consolidated clustering.

In our experiments, we used synthetic and real datasets containing projection images of the Mm-cpn protein in its "open" and "closed" conformations. The two synthetic datasets contained 20,000 projection images, 100×100 pixels each, while the four real datasets contained 10,000 projection images, 120×120 pixels each. The distribution of conformational states within all datasets was 50%-50%. One of the synthetic datasets (S1) was noiseless, while the other one (S2) had Gaussian noise added in order to obtain a signal-to-noise ratio of 0.10. This dataset was also randomly misaligned around the center of the image. The four real datasets differ in the quality of the image alignment, emulating different experimental conditions. Dataset R1 was just centered in relation to the average of the images. Dataset R2 was centered by comparison against re-projections of Mm-cpn 3D models in the two conformations. Dataset R3 is similar to R2, but its principal components were calculated on an extended dataset with artificially introduced rotated copies of each image. This procedure aims to emulate a situation in which more data is available. The last dataset, R4, was aligned both translationally and rotationally by comparison against re-projections of Mm-cpn 3D models in the two conformations.

All datasets were compressed to 100 dimensions using PCA. Visual inspection of the eigenimages (Figure 6.1) shows that the improvement on alignment quality makes the first components more informative with respect to the true underlying signal. This can be assessed by observing structural features of Mm-cpn on these components, including features related to its D8 symmetry. Also, the eigenvalue spectrum of the datasets (Figure 6.2) indicate that most of the relevant variance is concentrated on the first few principal components, while the rest of them are mostly associated with random noise.

By plotting the projection coordinates of the images onto two or three principal components, we can gain insight on the data distribution in the higher-dimensional feature space. Notably, by applying the structural labels on these plots (Figures 6.3 and 6.4), it can be clearly seen that projection data from the "open" and "closed" states occupy different manifolds. As expected, the more noisy and misaligned the datasets, the more difficult becomes the recognition of such manifolds.

Initially, we used a relatively large self-organizing map with 10×25 neuron units to explore the synthetic datasets. Figures 6.5 and 6.6 indicate that the SOM was able to recognize the conformational separation within the dataset in a totally unsupervised fashion, even for dataset S2. Such separation was later confirmed by using the true data labels, as shown in Figure 6.9.

In our first experiment using consensus clustering, we confirmed that higher accuracy could be achieved in determining the structural labels for the synthetic datasets. We used a simple agreement method to obtain a consensus between the previously trained SOMs and spectral clustering. Whenever the two algorithms disagreed, the data point was discarded. Discarding low-quality images is a common procedure in single particle reconstructions, and, if needed, more images can be collected to compensate for data prunning. However, such consensus method is naïve and only suitable to a small number of individual solutions, as disagreements tend to be more frequent with the inclusion of more base clusterings.

In our second ensemble experiment, we used the CSPA, HGPA and MCLA heuristics to obtain a consensus between HAC, k-means, SOM and three versions of spectral clustering. The number of clusters requested was changed from 2 up to 10, making a total of 54 base solutions. Datasets S1, S2 and R1 were analyzed based on their first 100 principal components. We confirmed that the consensus heuristics can provide more accurate solutions than individual algorithms, most notably for the real dataset.

Also, we aimed to determine the number of clusters automatically by choosing the consensus solution with highest Average Normalized Mutual Information with the base clusterings. The relatively low ANMI peak for the real dataset indicates that individual algorithms came up with very diverse opinions about the data clustering, in comparison to datasets S1 and S2. While the ensemble could correctly determine the number of conformational clusters for the synthetic datasets, it came up with six clusters for dataset R1, provided by MCLA. Two out of these six clusters were empty, and another two were barely populated. In a practical scenario, these small clusters could be discarded without prejudice to the reconstruction procedure. The two remaining densely populated clusters then corresponded to the true clusters with relatively high accuracy (76.58%). This result is evidence that cluster ensembles provide a valuable tool for model selection in unsupervised classification of cryo-EM data. We observed that the HGPA heuristic failed to provide meaningful solutions with two clusters, but it could correctly recognize the two conformational clusters using a three-way partition (Figure 6.13). Therefore, while we cannot under-

estimate the quality of the HGPA heuristic, the analysis of its performance is more intrincate than for the other approaches on our datasets.

Our third and last experiment assumed that the true number of clusters was known, as in a validation scenario. In this context, the user wants to verify whether the structural assignments performed by conventional reconstruction methods are consistent with the data distribution. We then assessed the stability of individual and ensemble algorithms, by performing 10 runs for each of them, and also whether the quality of the alignment among the real datasets implied higher classification accuracy. Clustering was performed using the first 10 and the first 100 principal components for all datasets. We used six diverse individual algorithms: HAC, k-means, SOM, Gaussian Mixture Model, spectral clustering and METIS. The consensus solutions were provided by k-means, CSPA, HGPA and MCLA. We compared the labels provided by the algorithms and consensus solutions against the ground truth using three different cluster similarity indices: the percentage of matches between solutions, the Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI). While the percentage of matches has intuitive appeal, it is subject to the cluster correspondence problem. On the other hand, NMI and ARI have non-linear behavior (Figure 6.14), rendering difficult interpretation.

From the observation of Figures 6.15 and 6.16, we see that the difference in performance for 10 or 100 principal components is negligible. This is evidence that the relevant dataset information is concentrated on the first few components, and they should be enough to achieve high unsupervised classification accuracy. Regarding the consensus solutions, we see that the ensemble always provides solutions of comparable quality, if not better, than the best performing individual solutions. Also, consensus solutions are more stable than certain individual algorithms, notably those highly dependent on the initialization of parameters like the SOM and the GMM. In all cases, k-means, CSPA and MCLA provided consensus solutions of comparable quality. Therefore, an ensemble is a *safer* approach than resorting to any specific unsupervised algorithm individually. However, we would recommend avoiding the CSPA heuristic due to its quadratic computational complexity in time and memory.

Relatively high accuracy (almost 80%) could be achieved even for our most challenging dataset, R1. Nevertheless, we see that the improvement on the alignment quality makes the conformational clouds more distinguishable to clustering algorithms. Therefore, we see that alignment is a crucial step for the success of unsupervised conformational classification.

We would like to draw special attention to the METIS algorithm, as it could achieve 98.58% accuracy on a real dataset (R4), standing apart from every other individual solution. This is probably a consequence of its ability to address both global and local aspects of the dataset distribution when represented as a similarity graph. Finally, we have showed that the MCLA provides an inherent cluster assignment confidence level that can be useful to exclude unreliable data in the context of single particle reconstructions (Figure 6.18).

7.1 Future research directions

We hope that the investigation here presented leads to improvements in solving 3D models from heterogeneous cryo-EM samples. One thing to be addressed is certainly whether the cluster ensemble approach can be useful in *ab initio* structure determination. Such question can only be answered by performing 3D reconstructions using the partitions provided by the cluster ensemble and comparing them to conventional heterogeneous reconstruction approaches. We could not perform this kind of experiment due to time constraints. We also acknowledge that Mm-cpn may be a relatively easy biological system for unsupervised classification, in face of its rather striking conformational differences between the "open" and "closed" states. It is expected that such low-frequency variations will be readily reflected in the distribution of the data in the feature space. Nevertheless, it served well the purpose of demonstrating that unsupervised conformational classification is feasible on real cryo-EM datasets. Therefore, we think that analyzing datasets with more subtle, high-frequency conformational oscillations, by cluster ensembles is a topic worth investigating.

We would like also to draw attention to the fact that the ensemble framework could be extended to take into account not only different clustering algorithms, but also different perspectives on the datasets (feature-distributed clustering). For example, selected areas of interest within the images could be classified separately, as well as different bands of the frequency spectrum. This latter approach is equivalent to clustering data using different filtering parameters. Classification in conjugate spaces (co-classification), as proposed by Borland & van Heel (1990) still remain to be properly explored and could be useful to detect conformational changes, especially when combined with modern co-clustering approaches (Busygin, Prokopyev & Pardalos, 2008).

Additionally, invariant features could be extracted from the images in order to investigate whether they contribute to the observation of conformational manifolds.

Finally, from the results obtained with spectral clustering and METIS, we point that graph partitioning algorithms are very promising in detecting conformational clusters without any assumption on their shapes, and likely represent computationally cheaper alternatives to conventional heterogeneous reconstruction procedures or to intrincated manifold learning algorithms.

References

Acharya, A., & Ghosh, J. (2013). A survey of Consensus Clustering. *Handbook of Cluster Analysis*.

Adrian, M., Dubochet, J., Lepault, J., & McDowall, A. W. (1984). Cryo-electron microscopy of viruses. *Nature*, *308*(5954), 32–36.

Arbeláez, P., Han, B.-G., Typke, D., Lim, J., Glaeser, R. M., & Malik, J. (2011). Experimental evaluation of support vector machine-based and correlation-based approaches to automatic particle selection. *Journal of Structural Biology*, *175*(3), 319–328.

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027–1035). Society for Industrial and Applied Mathematics.

Baker, L. A., & Rubinstein, J. L. (2010). Chapter Fifteen-Radiation Damage in Electron Cryomicroscopy. *Methods in Enzymology*, *481*, 371–388.

Ban, N., Nissen, P., Hansen, J., Moore, P. B., & Steitz, T. A. (2000). The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science*, *289*(5481), 905–920.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005). Clustering with Bregman Divergences. *Journal of Machine Learning Research*, *6*, 1705–1749.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Benzécri, J.-P. (1992). *Correspondence analysis handbook*. Marcel Dekker New York.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer Jr., E. F., Brice, M. D., Rodgers, J. R., … Tasumi, M. (1977). The protein data bank: A computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, *112*(3), 535–542.

Bingham, E., & Mannila, H. (2001). Random projection in dimensionality reduction: Applications to image and text data. In *KDD*. San Francisco.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer Berlin / Heidelberg.

Bishop, C. M., Svensén, M., & Williams, C. K. I. (1998). GTM: The Generative Topographic Mapping. *Neural Computation*, *10*(1982).

Borland, L., & van Heel, M. (1990). Classification of image data in conjugate representation spaces. *Journal of the Optical Society of America A*, *7*(4), 601.

Bradley, P. S., Mangasarian, O. L., & Street, W. N. (1997). Clustering via concave minimization. *Advances in Neural Information Processing Systems*, 368–374.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Bretaudiere, J.-P., & Frank, J. (1986). Reconstitution of molecule images analysed by correspondence analysis: a tool for structural interpretation. *Journal of Microscopy*, *144*(1), 1–14.

Buerk, I. (2012). *Fast and efficient spectral clustering*.

Burgess, S. A., Walker, M. L., Thirumurugan, K., Trinick, J., & Knight, P. J. (2004). Use of negative stain and single-particle image processing to explore dynamic properties of flexible macromolecules. *Journal of Structural Biology*, *147*(3), 247–258.

Burgess, S. A., Walker, M. L., White, H. D., & Trinick, J. (1997). Flexibility within myosin heads revealed by negative stain and single-particle analysis. *Journal of Cell Biology*, *139*(3), 675–681.

Busygin, S., Prokopyev, O., & Pardalos, P. M. (2008). Biclustering in data mining. *Computers & Operations Research*, *35*(9), 2964–2987.

Carazo, J. M., Rivera, F. F., Zapata, E. L., Radermacher, M., & Frank, J. (1990). Fuzzy sets-based classification of electron microscopy images of biological macromolecules with an application to ribosomal particles. *Journal of Microscopy*, *157*(2), 187–203.

Chang, K., & Ghosh, J. (2000). Three-Dimensional Model-based Object Recognition and Pose Estimation using Probabilistic Principal Surfaces. *SPIE CANNIP*, 1–12.

Chang, K., & Ghosh, J. (2001). A unified model for probabilistic principal surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(1), 22–41.

Chen, J. Z., & Grigorieff, N. (2007). SIGNATURE: a single-particle selection system for molecular electron microscopy. *Journal of Structural Biology*, *157*(1), 168–73.

Chen, M. (2012). *kmeans clustering*.

Chen, S., McMullan, G., Faruqi, A. R., Murshudov, G. N., Short, J. M., Scheres, S. H. W., & Henderson, R. (2013). High-resolution noise substitution to measure overfitting and validate resolution in 3D structure determination by single particle electron cryomicroscopy. *Ultramicroscopy*, *135*(0), 24–35.

Chiu, W. (1993). What does electron cryomicroscopy provide that X-ray crystallography and NMR spectroscopy cannot? *Annual Review of Biophysics and Biomolecular Structure*, *22*(1), 233–255.

Chung, F. R. K. (1997). *Spectral graph theory* (Vol. 92). American Mathematical Soc.

Coifman, R. R., Shkolnisky, Y., Sigworth, F. J., & Singer, A. (2010). Reference Free Structure Determination through Eigenvectors of Center of Mass Operators. *Applied and Computational Harmonic Analysis*, *28*(3), 296–312.

David L. Davies, & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(2), 224–227.

De Rosier, D. J., & Klug, A. (1968). Reconstruction of three dimensional structures from electron micrographs. *Nature*, *217*(5124), 130–134.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorith. *Journal of the Royal Statistical Society*, *39*(1), 1–38.

Diday, E. (1971). Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, *19*(2), 19–33.

Domeniconi, C., & Al-Razgan, M. (2009). Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *2*(4), 17.

Dube, P., Tavares, P., Lurz, R., & Van Heel, M. (1993). The portal protein of bacteriophage SPP1: a DNA pump with 13-fold symmetry. *The EMBO Journal*, *12*(4), 1303.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification*. John Wiley & Sons.

Dudoit, S., & Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, *19*(9), 1090–1099.

Dy, J. G., & Brodley, C. E. (2004). Feature Selection for Unsupervised Learning, *5*, 845–889.

Eddy, S. R. (2004). What is Bayesian statistics? *Nature Biotechnology*, *22*(9), 1177–8.

Elad, N., Clare, D. K., Salbil, H. R., & Orlova, E. V. (2008). Detection and separation of heterogeneity in molecular complexes by statistical analysis of their two-dimensional projections. *Journal of Structural Biology*, *162*(1), 108–120.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

Everitt, B. S., Landau, S., & Leese, M. (2001). Cluster analysis. 2001. *Arnold, London*.

Fern, X. Z., & Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning* (p. 36). ACM.

Fernández, J. J., & Carazo, J. M. (1996). Analysis of structural variability within two-dimensional biological crystals by a combination of patch averaging techniques and self organizing maps. *Ultramicroscopy*, *65*(1-2), 81–93.

Fernández-Morán, H. (1966). High-resolution electron microscopy with superconducting lenses at liquid helium temperatures. *Proceedings of the National Academy of Sciences of the United States of America*, *56*(3), 801.

Feynman, R. P. (1960). There's plenty of room at the bottom. *Engineering and Science*, *23*(5), 22–36.

Fischler, M. A., & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, *24*(6), 381–395.

Frank, J. (2006). *Three-Dimensional Electron Microscopy of Macromolecular Assemblies* (2nd ed.). Oxford University Press.

Frank, J., Bretaudiere, J.-P., Carazo, J. M., Verschoor, A., & Wagenknecht, T. (1988). Classification of images of biomolecular assemblies: a study of ribosomes and ribosomal subunits of Escherichia coli. *Journal of Microscopy*, *150*(2), 99–115.

Frank, J., Radermacher, M., Penczek, P., Zhu, J., Li, Y., Ladjadj, M., & Leith, A. (1996). SPIDER and WEB: Processing and Visualization of Images in 3D Electron Microscopy and Related Fields. *Journal of Structural Biology*, *116*(1), 190–199.

Frank, J., & van Heel, M. (1982). Correspondence analysis of aligned images of biological particles. *Journal of Molecular Biology*, *161*(1), 134–137.

Fritzke, B. (1995). A Growing Neural Gas Network Learns Topologies. In *Advances in Neural Information Processing Systems 7* (pp. 625–632). MIT Press.

Fu, J., Gao, H., & Frank, J. (2007). Unsupervised classification of single particles by cluster tracking in multi-dimensional space. *Journal of Structural Biology*, *157*(1), 226–239.

Ghosh, J., & Acharya, A. (2011). Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(4), 305–315.

Ghosh, J., Strehl, A., & Merugu, S. (2002). A Consensus Framework for Integrating Distributed Clusterings Under Limited Knowledge Sharing. In *Proc. NSF Workshop on Next Generation Data Mining*. Baltimore.

Glaeser, R. M. (2008). Cryo-Electron Microscopy of Biological Nanostructures. *Physics Today*, *61*(1), 48–54.

Glaeser, R. M., Downing, K., DeRosier, D., Chiu, W., & Frank, J. (2007). *Electron crystallography of biological macromolecules*. Oxford University Press New York:

Glaeser, R. M., & Hall, R. J. (2011). Reaching the information limit in cryo-EM of biological macromolecules: experimental aspects. *Biophysical Journal*, *100*(10), 2331–7.

Golub, G. H., & Van Loan, C. F. (1996). Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA*, 374–426.

Gorban, A. N., Kégl, B., Wunsch, D. C., & Zinovyev, A. (2007). Principal Manifolds for Data Visualization and Dimension Reduction.

Grandison, S., Roberts, C., & Morris, R. J. (2009). The application of 3D Zernike moments for the description of "model-free" molecular structure, functional motion, and structural reliability. *Journal of Computational Biology : A Journal of Computational Molecular Cell Biology*, *16*(3), 487–500.

Grigorieff, N. (2007). FREALIGN: high-resolution refinement of single particle structures. *Journal of Structural Biology*, *157*(1), 117–25.

Grigorieff, N. (2013). Direct detection pays off for electron cryo-microscopy. *eLife*, *2*.

Grigorieff, N., & Harrison, S. C. (2011). Near-atomic resolution reconstructions of icosahedral viruses from electron cryo-microscopy. *Current Opinion in Structural Biology*, *21*(2), 265–73.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Hagen, L., & Kahng, A. B. (1992). New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, Ieee Transactions on*, *11*(9), 1074–1085.

Harauz, G., & Ottensmeyer, F. P. (1983). Direct three-dimensional reconstruction for macromolecular complexes from electron micrographs. *Ultramicroscopy*, *12*(4), 309–319.

Harauz, G., & van Heel, M. (1985). Direct 3D Reconstruction from Projections with Initially Unknown Angles. In H. Lemke, M. Rhodes, C. C. Jaffee, & R. Felix (Eds.), *Computer Assisted Radiology / Computergestützte Radiologie SE - 104* (pp. 649–653). Springer Berlin Heidelberg.

Harauz, G., & van Heel, M. (1986). Exact filters for general geometry three dimensional reconstruction. *Optik*, *78*(4), 146–156.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*.

Haykin, S. (1999). *Neural Networks - A Comprehensive Foundation*. Prentice Hall.

Henderson, R. (1990). Cryo-protection of protein crystals against radiation damage in electron and X-ray diffraction. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *241*(1300), 6–8.

Henderson, R. (1995). The potential and limitations of neutrons, electrons and X-rays for atomic resolution microscopy of unstained biological molecules. *Quarterly Reviews of Biophysics*, *28*(02), 171–193.

Henderson, R. (2004). Realizing the potential of electron cryo-microscopy. *Quarterly Reviews of Biophysics*, *37*(01), 3–13.

Henderson, R. (2013). Avoiding the pitfalls of single particle cryo-electron microscopy: Einstein from noise. *Proceedings of the National Academy of Sciences of the United States of America*, 1–5.

Henderson, R., Sali, A., Baker, M. L., Carragher, B., Devkota, B., Downing, K. H., … Lawson, C. L. (2012). Outcome of the first electron microscopy validation task force meeting. *Structure*, *20*(2), 205–14.

Hendrickson, B., & Leland, R. W. (1995). A Multi-Level Algorithm For Partitioning Graphs. *SC*, *95*, 28.

Herman, G. T., & Kalinowski, M. (2008). Classification of heterogeneous electron microscopic projections into homogeneous subsets. *Ultramicroscopy*, *108*(4), 327–338.

Hohn, M., Tang, G., Goodyear, G., Baldwin, P. R., Huang, Z., Penczek, P. A., … Ludtke, S. J. (2007). SPARX, a new environment for Cryo-EM image processing. *Journal of Structural Biology*, *157*(1), 47–55.

Horne, R. W., Brenner, S., Waterson, A. P., & Wildy, P. (1959). The icosahedral form of an adenovirus. *Journal of Molecular Biology*, *1*(1), 84–IN15.

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & de Carvalho, A. C. P. L. F. (2009). A Survey of Evolutionary Algorithms for Clustering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *39*(2), 133–155.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. *NIR news* (Vol. 19). John Wiley & Sons.

Jaitly, N., Brubaker, M. A., Rubinstein, J. L., & Lilien, R. H. (2010). A Bayesian method for 3D macromolecular structure inference using class average images from single particle electron microscopy. *Bioinformatics*, *26*(19), 2406–2415.

Jensen, G. (2010a). *Cryo-EM Part A: Sample Preparation and Data Collection: Sample Preparation and Data Collection* (Vol. 481). Academic Press.

Jensen, G. (2010b). *Cryo-EM Part B: 3-D Reconstruction*. Academic Press.

Jensen, G. (2010c). *Cryo-EM Part C: Analyses, Interpretation and Case Studies*.

Karypis, G. (2013). METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices.

Karypis, G., Aggarwal, R., Kumar, V., & Shekhar, S. (1997). Multilevel Hypergraph Partitioning: Application In Vlsi Domain. *Design Automation Conference, 1997. Proceedings of the 34th*.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*.

Karypis, G., & Kumar, V. (1995a). *METIS* Unstructured Graph Partitioning and Sparse Matrix Ordering*.

Karypis, G., & Kumar, V. (1995b). Multilevel Graph Partitioning Schemes. In *International Conference on Parallel Processing* (pp. 113–122).

Karypis, G., & Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, *20*(1), 359–392.

Katsevich, G., Katsevich, A., & Singer, A. (2013). Covariance Matrix Estimation for the Cryo-EM Heterogeneity Problem. *arXiv Preprint arXiv:1309.1737*.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

Kawata, M., & Sato, C. (2007). A statistically harmonized alignment-classification in image space enables accurate and robust alignment of noisy images in single particle analysis. *Journal of Electron Microscopy*, *56*(3), 83–92.

Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, *49*(2), 291–307.

Klaholz, B. P., Myasnikov, A. G., & Van Heel, M. (2004). Visualization of release factor 3 on the ribosome during termination of protein synthesis. *Nature*, *427*(6977), 862–5.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*(1), 59–69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer.

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Lander, G. C., Stagg, S. M., Voss, N. R., Cheng, A., Fellmann, D., Yoshioka, C., … Carragher, B. (2009). Appion: an integrated, database-driven pipeline to facilitate EM image processing. *Journal of Structural Biology*, *166*(1), 95–102.

Langlois, R., Pallesen, J., & Frank, J. (2011). Reference-free particle selection enhanced with semi-supervised machine learning for cryo-electron microscopy. *Journal of Structural Biology*, *175*(3), 353–61.

Leschziner, A. E., & Nogales, E. (2007). Visualizing flexibility at molecular resolution: analysis of heterogeneity in single-particle electron microscopy reconstructions. *Annual Review of Biophysics and Biomolecular Structure*, *36*(December 2006), 43–62.

Liao, H. Y., & Frank, J. (2010). Classification by bootstrapping in single particle methods. In *Proceedings of the 2010 IEEE international conference on Biomedical imaging: from nano to Macro* (pp. 169–172). Piscataway, NJ, USA: IEEE Press.

Liu, W., Pokharel, P. P., & Príncipe, J. C. (2007). Correntropy: properties and applications in non-Gaussian signal processing. *Signal Processing, IEEE Transactions on*, *55*(11), 5286–5298.

Ludtke, S. J., Baldwin, P. R., & Chiu, W. (1999). EMAN: semiautomated software for high-resolution single-particle reconstructions. *Journal of Structural Biology*, *128*(1), 82–97.

Lyumkis, D., Brilot, A. F., Theobald, D. L., & Grigorieff, N. (2013). Likelihood-based classification of cryo-EM images using FREALIGN. *Journal of Structural Biology*, *183*(3), 377–388.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, p. 14).

Maier, M., von Luxburg, U., & Hein, M. (2012). How the result of graph clustering methods depends on the construction of the graph. *ESAIM: Probability and Statistics*, *eFirst*.

Marabini, R., & Carazo, J. M. (1994). Pattern recognition and classification of images of biological macromolecules using artificial neural networks. *Biophysical Journal*, *66*(6), 1804–14.

McLachlan, G., & Peel, D. (2000). *Finite Mixture Models*. John Wiley & Sons.

McLachlan, G., & Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning theory and kernel machines* (pp. 173–187). Springer.

Mellwig, C., & Böttcher, B. (2001). Dealing with particles in different conformational states by electron microscopy and image processing. *Journal of Structural Biology*, *133*(2-3), 214–20.

Mindell, J. a., & Grigorieff, N. (2003). Accurate determination of local defocus and specimen tilt in electron microscopy. *Journal of Structural Biology*, *142*(3), 334–347.

Mohar, B. (1997). *Some applications of Laplace eigenvalues of graphs*. Springer.

Mohar, B., & Alavi, Y. (1991). The Laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, *2*, 871–898.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, *2*, 849–856.

Nicholson, W. V, & Glaeser, R. M. (2001). Review: automatic particle detection in electron microscopy. *Journal of Structural Biology*, *133*(2-3), 90–101.

Ogura, T., Iwasaki, K., & Sato, C. (2003). Topology representing network enables highly accurate classification of protein images taken by cryo electron-microscope without masking. *Journal of Structural Biology*, *143*(3), 185–200.

Ogura, T., & Sato, C. (2004). Automatic particle pickup method using a neural network has high accuracy by applying an initial weight derived from eigenimages : a new reference free method for single-particle analysis. *Journal of Structural Biology*, *145*, 63–75.

Orlova, E. V, & Saibil, H. R. (2011). Structural analysis of macromolecular assemblies by electron microscopy. *Chemical Reviews*, *111*(12), 7710–48.

Palade, G. E. (1955). A small particulate component of the cytoplasm. *The Journal of Biophysical and Biochemical Cytology*, *1*(1), 59.

Pascual, A., Merelo, J. J., Carazo, J., & Autnoma, N. D. B. U. (1999). Application of the Fuzzy Kohonen Clustering Network to Biological Macromolecules Images Classification. *Lecture Notes on Computer Science*.

Pascual-Montano, A., Donate, L. E., Valle, M., Bárcena, M., Pascual-Marqui, R. D., Carazo, J. M., & Barcena, M. (2001). A novel neural network technique for analysis and classification of EM single-particle images. *Journal of Structural Biology*, *133*(2-3), 233–245.

Pascual-Montano, A., Taylor, K. A., Winkler, H., Pascual-Marqui, R. D., & Carazo, J.-M. (2002). Quantitative self-organizing maps for clustering electron tomograms. *Journal of Structural Biology*, *138*(1-2), 114–22.

Penczek, P. a, Frank, J., & Spahn, C. M. T. (2006). A method of focused classification, based on the bootstrap 3D variance analysis, and its application to EF-G-dependent translocation. *Journal of Structural Biology*, *154*(2), 184–94.

Penczek, P. a, Yang, C., Frank, J., & Spahn, C. M. T. (2006). Estimation of variance in single-particle reconstruction using the bootstrap technique. *Journal of Structural Biology*, *154*(2), 168–83.

Penczek, P. A., Kimmel, M., & Spahn, C. M. T. (2011). Identifying Conformational States of Macromolecules by Eigen-Analysis of Resampled Cryo-EM Images. *Structure*, *19*(11), 1582–1590.

Penczek, P. A., Zhu, J., & Frank, J. (1996). A common-lines based method for determining orientations for N > 3 particle projections simultaneously. *Ultramicroscopy*, *63*(3–4), 205–218.

Penczek, P., Radermacher, M., & Frank, J. (1992). Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy*, *40*(1), 33–53.

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., & Ferrin, T. E. (2004). UCSF Chimera—A visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, *25*(13), 1605–1612.

Punera, K., & Ghosh, J. (2008). Consensus-based Ensembles of Soft Clusterings. *Applied Artificial Intelligence*, *22*(7-8), 780–810.

Radon, J. (1986). On the determination of functions from their integral values along certain manifolds. *Medical Imaging, IEEE Transactions on*, *5*(4), 170–176.

Reimer, L., & Kohl, H. (2008). *Transmission electron microscopy: physics of image formation* (Vol. 36). Springer.

Robinson, C. V, Sali, A., & Baumeister, W. (2007). The molecular sociology of the cell. *Nature*, *450*(7172), 973–82.

Rosenthal, P. B., & Henderson, R. (2003). Optimal Determination of Particle Orientation, Absolute Hand, and Contrast Loss in Single-particle Electron Cryomicroscopy. *Journal of Molecular Biology*, *333*(4), 721–745.

Saibil, H. (2000). Macromolecular structure determination by cryo-electron microscopy. *Acta Crystallographica Section D*, *56*(10), 1215–1222.

Samsó, M., Palumbo, M. J., Radermacher, M., Liu, J. S., & Lawrence, C. E. (2002). A Bayesian method for classification of images from electron micrographs. *Journal of Structural Biology*, *138*(3), 157–170.

Saxton, W. O., & Frank, J. (1976). Motif detection in quantum noise-limited electron micrographs by cross-correlation. *Ultramicroscopy*, *2*(0), 219–227.

Schatz, M., Orlova, E. V, Dube, P., Stark, H., Zemlin, F., & van Heel, M. (1997). Angular Reconstitution in Three-Dimensional Electron Microscopy: Practical and Technical Aspects. *Scanning Microscopy*, *11*, 179–193.

Schatz, M., & van Heel, M. (1990). Invariant Classification of Molecular Views in Electron Micrographs. *Ultramicroscopy*, *32*, 255–264.

Schatz, M., & van Heel, M. (1992). Invariant recognition of molecular projections in vitreous ice preparations. *Ultramicroscopy*, *45*(1), 15–22.

Scheres, S. H. (2010). Classification of structural heterogeneity by maximum-likelihood methods. *Meth. Enzymol.*, *482*, 295–320.

Scheres, S. H. W. (2012a). A Bayesian view on cryo-EM structure determination. *Journal of Molecular Biology*, *415*(2), 406–18.

Scheres, S. H. W. (2012b). RELION: Implementation of a Bayesian approach to cryo-EM structure determination. *Journal of Structural Biology*, *180*(3), 519–530.

Scheres, S. H. W., & Carazo, J. M. (2009). Introducing robustness to maximum-likelihood refinement of electron-microscopy data. *Acta Crystallographica. Section D, Biological Crystallography*, *65*(Pt 7), 672–8.

Scheres, S. H. W., Gao, H., Valle, M., Herman, G. T., Eggermont, P. P. B., Frank, J., & Carazo, J. (2007). Disentangling conformational states of macromolecules in 3D-EM through likelihood optimization. *Nature Methods*, *4*(1), 27–29.

Scheres, S. H. W., Marabini, R., Lanzavecchia, S., Cantele, F., Rutten, T., Fuller, S. D., … Martin, C. S. (2005). Classification of single-projection reconstructions for cryo-electron microscopy data of icosahedral viruses. *JOURNAL OF STRUCTURAL BIOLOGY*, *151*(1), 79–91.

Scheres, S. H. W., Núñez-Ramírez, R., Gómez-Llorente, Y., San Martín, C., Eggermont, P. P. B., & Carazo, J. M. (2007). Modeling experimental image formation for likelihood-based classification of electron microscopy data. *Structure (London, England : 1993)*, *15*(10), 1167–77.

Scheres, S. H. W., Nunez-Ramirez, R., Sorzano, C. O. S., Maria Carazo, J., & Marabini, R. (2008). Image processing for electron microscopy single-particle analysis using XMIPP. *Nature Protocols*, *3*(6), 977–990.

Scheres, S. H. W., Valle, M., & Carazo, J.-M. (2005). Fast maximum-likelihood refinement of electron microscopy images. *Bioinformatics*, *21*(suppl 2), ii243–ii244.

Scheres, S. H. W., Valle, M., Grob, P., Nogales, E., & Carazo, J.-M. J.-M. (2009). Maximum likelihood refinement of electron microscopy data with normalization errors. *Journal of Structural Biology*, *166*(2), 234–240.

Scheres, S. H. W., Valle, M., Nunez, R., Sorzano, C. O. S., Marabini, R., Herman, G. T., & Carazo, J. M. (2005). Maximum-likelihood multi-reference refinement for electron microscopy images. *Journal of Molecular Biology*, *348*(1), 139–149.

Schlesinger, M. I., & Hlavac, V. (2002). *Ten lectures on statistical and structural pattern recognition* (Vol. 24). Springer.

Schluenzen, F., Tocilj, A., Zarivach, R., Harms, J., Gluehmann, M., Janell, D., … Yonath, A. (2000). Structure of functionally activated small ribosomal subunit at 3.3 Å resolution. *Cell*, *102*(5), 615–623.

Scholkopf, B., Smola, A., & Muller, K. R. (1996). *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*. Tubingen, Germany.

Schwander, P., Fung, R., Phillips Jr, G. N., & Ourmazd, A. (2010). Mapping the conformations of biological assemblies. *New Journal of Physics*, *12*(3), 35007.

Serway, R., & Jewett, J. (2013). *Physics for scientists and engineers*. Cengage Learning.

Shaikh, T. R., Gao, H., Baxter, W. T., Asturias, F. J., Boisset, N., Leith, A., & Frank, J. (2008). SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. *Nature Protocols*, *3*(12), 1941–1974.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, *27*(July 1928), 379–423.

Shatsky, M., Hall, R. J., Nogales, E., Malik, J., & Brenner, S. E. (2010). Automated multi-model reconstruction from single-particle electron microscopy data. *Journal of Structural Biology*, *170*(1), 98–108.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Sigworth, F. J. (1998). A maximum-likelihood approach to single-particle image refinement. *Journal of Structural Biology*, *122*(3), 328–39.

Singer, A., Zhao, Z., Shkolnisky, Y., & Hadani, R. (2012). Viewing Angle Classification of Cryo-Electron Microscopy Images Using Eigenvectors. *SIAM J Imaging Sci.*, *4*(2), 723–759.

Söderlund, H., von Bonsdorff, C.-H., & Ulmanen, I. (1979). Comparison of the Structural Properties of Sindbis and Semliki Forest Virus Nucleocapsids. *Journal of General Virology*, *45*(1), 15–26.

Sorzano, C. O. S., Bilbao-Castro, J. R., Shkolnisky, Y., Alcorlo, M., Melero, R., Caffarena-Fernandez, G., … Caffarena-Fernández, G. (2010). A clustering approach to multireference alignment of single-particle projections in electron microscopy. *Journal of Structural Biology*, *171*(2), 197–206.

Sorzano, C. O. S., Marabini, R., Velázquez-Muriel, J., Bilbao-Castro, J. R., Scheres, S. H. W., Carazo, J. M., … Velazquez-Muriel, J. (2004). XMIPP: a new generation of an open-source image processing package for electron microscopy. *JOURNAL OF STRUCTURAL BIOLOGY*, *148*(2), 194–204.

Spahn, C. M. T., & Penczek, P. A. (2009). Exploring conformational modes of macromolecular assemblies by multiparticle cryo-EM. *Current Opinion in Structural Biology*, *19*(5), 623–631.

Strehl, A. (2011). *ClusterPack Matlab / Octave Toolbox*.

Strehl, A., & Ghosh, J. (2002). Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, *3*, 583–617.

Strehl, A., Ghosh, J., & Mooney, R. (2000). Impact of Similarity Measures on Web-page Clustering. In *Workshop of Artificial Intelligence for Web Search*.

Tagare, H. D., Barthel, A., & Sigworth, F. J. (2010). An adaptive Expectation-Maximization algorithm with GPU implementation for electron cryomicroscopy. *Journal of Structural Biology*, *171*(3), 256–265.

Tang, G., Peng, L., Baldwin, P. R., Mann, D. S., Jiang, W., Rees, I., & Ludtke, S. J. (2007). EMAN2: An extensible image processing suite for electron microscopy. *Journal of Structural Biology*, *157*(1), 38–46.

Theodoridis, S., & Koutroumbas, K. (2008). *Pattern Recognition*. (A. Press, Ed.) (3rd. ed.).

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *61*(3), 611–622.

Topchy, A., Jain, A. K., & Punch, W. (2004). A mixture model of clustering ensembles. In *Proc. SIAM Intl. Conf. on Data Mining*. Citeseer.

Topchy, A., Minaei-Bidgoli, B., Jain, A. K., & Punch, W. F. (2004). Adaptive clustering ensembles. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.

Ueno, Y., Kawata, M., & Umeyama, S. (2005). Intrinsic classification of single particle images by spectral clustering. *Proceedings of Biosignal Processing and Classification. Portugal: INSTICC Press*, 60–67.

Ueno, Y., Mio, M., Sato, C., & Mio, K. (2007). Single particle conformations of human serum albumin by electron microscopy. *J Electron Microsc (Tokyo)*, *56*(3), 103–110.

Ultsch, A., & Siemon, H. P. (1990). Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. In *INNC'90: International Neural Network Conference*. Paris: Kluwer.

Unser, M., Trus, B. L., & Steven, A. C. (1989). Normalization procedures and factorial representations for classification of correlation-aligned images: a comparative study. *Ultramicroscopy*, *30*(3), 299–310.

Urruty, T., Djeraba, C., & Simovici, D. (2007). Clustering by random projections. *Advances in Data Mining. Theoretical Aspects and Applications*, 107–119.

Van Heel, M. (1982). Detection of Objects in Quantum-Noise-Limited Images. *Ultramicroscopy*, *1982*(8), 331–342.

Van Heel, M. (1984). Multivariate Statistical Classification Of Noisy Images (Randomly Oriented Biological Macromolecules). *Ultramicroscopy*, *13*, 165–184.

Van Heel, M. (1986). Finding the characteristic views of macromolecules in extremely noisy electron micrographs. *Pattern Recognition in Practice*, *2*, 291–299.

Van Heel, M. (1987). Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction. *Ultramicroscopy*, *21*(2), 111–123.

Van Heel, M. (1989). Classification of very large electron microscopical image data sets. *Optik*, *82*(3), 114–126.

Van Heel, M., & Frank, J. (1981). Use of Multivariate Statistics in Analysing the Images of Biological Macromolecules. *Ultramicroscopy*, *6*, 187–194.

Van Heel, M., Gowen, B., Matadeen, R., Orlova, E. V, Finn, R., Pape, T., … Patwardhan, A. (2000). Single-particle electron cryo-microscopy: towards atomic resolution. *Quarterly Reviews of Biophysics*, *33*(4), 307–69.

Van Heel, M., Harauz, G., Orlova, E. V, Schmidt, R., Schatz, M., & VanHeel, M. (1996). A new generation of the IMAGIC image processing system. *Journal of Structural Biology*, *116*(1), 17–24.

Van Heel, M., Orlova, E. V, Harauz, G., Stark, H., Dube, P., Zemlin, F., & Schatz, M. (1997). Angular Reconstitution in Three-Dimensional Electron Microscopy: Historical and Theoretical Aspects. *Scanning Microscopy*, *11*, 195–210.

Van Heel, M., Portugal, R., Rohou, A., Linnemayr, C., Bebeacua, C., Schmidt, R., … Schatz, M. (2012). Four-dimensional cryo-electron microscopy at quasi-atomic resolution: IMAGIC 4D. *International Tables for Crystallography*, *F*., 624–628.

Van Heel, M., Portugal, R., & Schatz, M. (2009). Multivariate Statistical Analysis in Single Particle (Cryo) Electron Microscopy. *3D-EM Network of Excellence*, 1–47.

Van Heel, M., & Stöffler-Meilicke, M. (1985). Characteristic views of E. coli and B. stearothermophilus 30S ribosomal subunits in the electron microscope. *The EMBO Journal*, *4*(9), 2389.

Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (2000). *SOM Toolbox for Matlab 5*.

Von Luxburg, U. (2007). A Tutorial on Spectral Clustering. *CoRR*, *abs/0711.0*.

Wang, H., Shan, H., & Banerjee, A. (2011). Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, *4*(1), 54–70.

Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, *58*(301), 236–244.

White, H. E., Saibil, H. R., Ignatiou, A., & Orlova, E. V. (2004). Recognition and Separation of Single Particles with Size Variation by Statistical Analysis of their Images. *Journal of Molecular Biology*, *336*(2), 453–460.

Williams, D. B., & Carter, C. B. (2009). *Transmission Electron Microscopy: a Textbook for Materials Science*. *Transmission Electron Microscopy* (pp. 3–22). Springer.

Wimberly, B. T., Brodersen, D. E., Clemons, W. M., Morgan-Warren, R. J., Carter, A. P., Vonrhein, C., … Ramakrishnan, V. (2000). Structure of the 30S ribosomal subunit. *Nature*, *407*(6802), 327–339.

Woolford, D., Ericksson, G., Rothnagel, R., Muller, D., Landsberg, M. J., Pantelic, R. S., … Banks, J. (2007). SwarmPS: Rapid, semi-automated single particle selection software. *Journal of Structural Biology*, *157*(1), 174–188.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., … Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1–37.

Xiong, H., Wu, J., & Chen, J. (2009). K-means clustering versus validation measures: a data-distribution perspective. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *39*(2), 318–331.

Yang, Z., Fang, J., Chittuluru, J., Asturias, F. J., & Penczek, P. A. (2012). Iterative Stable Alignment and Clustering of 2D Transmission Electron Microscope Images. *Structure*, *20*(2), 237–247.

Yoshioka, C., Lyumkis, D., Carragher, B., & Potter, C. S. (2013). Maskiton: Interactive, web-based classification of single-particle electron microscopy images. *Journal of Structural Biology*, *182*(2), 155–63.

Yu, Z., & Frangakis, A. S. (2011). Classification of electron sub-tomograms with neural networks and its application to template-matching. *Journal of Structural Biology*, *174*(3), 494–504.

Zhang, J., Baker, M. L., Schröder, G. F., Douglas, N. R., Reissmann, S., Jakana, J., … Chiu, W. (2010). Mechanism of folding chamber closure in a group II chaperonin. *Nature*, *463*(7279), 379–83.

Zhou, Z. H. (2008). Towards atomic resolution structural determination by single-particle cryo-electron microscopy. *Current Opinion in Structural Biology*, *18*(2), 218–28.

Zhu, Y., Carragher, B., Glaeser, R. M., Fellmann, D., Bajaj, C., Bern, M., … Potter, C. S. (2004). Automatic particle selection: results of a comparative study. *Journal of Structural Biology*, *145*(1–2), 3–14.

List of publications

Belatini Jr., L. R., Righetto, R. D., Monteiro, M. F. G., Silva, C. A., Von Zuben, F. J., van Heel, M., & Portugal, R. V. (2013). A hyperspace viewer: visualization and classification of large image datasets. In Gordon Research Conference on Three-Dimensional Electron Microscopy (3DEM). New London, USA.

Righetto, R. D., Portugal, R. V. & Von Zuben, F. J. (2013). Classificação de Diversidade Estrutural em Dados de Microscopia Eletrônica de Transmissão através de Comitês de Máquinas. In Anais do VI Encontro de Alunos e Docentes do DCA (Vol. 2, pp. 59–62). Campinas, Brazil.

Righetto, R. D., Von Zuben, F. J. & Portugal, R. V. (2014). Validation of Structural Diversity on Cryo-Electron Microscopy Datasets by Clustering Algorithms. In 43a. Reunião Anual da SBBq. Foz do Iguaçu, Brazil.

Courses attended

The 5th Brazil School for Single-Particle Cryo-EM (2012). Socorro, Brazil.

EMBO Practical Course on Image Processing for Cryo-EM (2013). Birkbeck College, London, UK.