SHAPE ANALYSIS IN PROTEIN STRUCTURE ALIGNMENT

A THESIS SUBMITTED TO

THE UNIVERSITY OF KENT AT CANTERBURY

IN THE SUBJECT OF STATISTICS

FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY BY RESEARCH

By

Theodoros Gkolias

April 2018

Abstract

In this Thesis we explore the problem of structural alignment of protein molecules using statistical shape analysis techniques. The structural alignment problem can be divided into three smaller ones: the representation of protein structures, the sampling of possible alignments between the molecules and the evaluation of a given alignment. Previous work done in this field, can be divided in two approaches: an adhoc algorithmic approach from the Bioinformatics literature and an approach using statistical methods either in a likelihood or Bayesian framework. Both approaches address the problem from a different scope. For example, the algorithmic approach is easy to implement but lacks an overall modelling framework, and the Bayesian address this issue but sometimes the implementation is not straightforward.

We develop a method which is easy to implement and is based on statistical assumptions. In order to asses the quality of a given alignment we use a size and shape likelihood density which is based in the structure information of the molecules. This likelihood density is also extended to include sequence information and gap penalty parameters so that biologically meaningful solution can be produced. Furthermore, we develop a search algorithm to explore possible alignments from a given starting point. The results suggest that our approach produces better or equal alignments when it is compared to the most recent structural alignment methods. In most of the cases we managed to achieve a higher number of matched atoms combined with a high TMscore.

Moreover, we extended our method using Bayesian techniques to perform alignments based on posterior modes. In our approach, we estimate directly the mode of the posterior distribution which provides the final alignment between two molecules. We also, choose a different approach for treating the mean parameter. In previous methods the mean was either integrated out of the likelihood density or considered as fixed. We choose to assign a prior over it and obtain its posterior mode.

Finally, we consider an extension of the likelihood model assuming a Normal density for both the matched and unmatched parts of a molecule and diagonal covariance structure. We explore two different variants. In the first we consider a fixed zero mean for the unmatched parts of the molecules and in the second we consider a common mean for both the matched and unmatched parts. Based on simulated and real results, both models seems to perform well in obtaining high number of matched atoms and high TMscore.

Acknowledgements

I would like to thank my supervisor Dr Alfred Kume for his patience and continuous support during the four years of my study. He introduced me to the subject of statistical Shape Analysis, expanded my research interests, was always there when I needed help to solve difficult problems and being patient during the writing of this Thesis. Fred, I really enjoyed our discussions and it was a pleasure working with you.

I would also like to thank Dr Steffen Krusch for all his helpful comments which helped me to improve the structure of this Thesis and the School of Mathematics Statistics and Actuarial Science for providing funding during my PhD studies.

I would like to thank all my friends back in Greece Grigori, Kosta, Giorgi, Amalia, Nansy, Spyro, Stavro, Margarita, Aggeliki, Christina, Mitso, Koumi for all the support and being there for me in times of need all these years. I am also grateful to all the new people met in Canterbury. George and Giota, thanks for all the dinners, I would never forget our Walking Dead nights. Katia, thank you for all our discussions and how you always managed to relief some of the pressure. Also, I really enjoyed all these crazy nights out. Katherine, An, Lucy thanks for making our times in the office so enjoyable. Bill, Brendan, Ela it was a pleasure meeting you. A special thanks to Claire, for being there and fixing any problem for all the students. You are the best.

Finally, I would like to thank my family. My parents Nikos and Olympia and my sister Anny, for all their love, support and trust. I would never be able to complete this Thesis without them. Thank you.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

In this Thesis we explore the problem of protein structure alignment from a statistical point of view. Our research is focused on developing modelling techniques to optimize the alignment for two or more protein molecules. We mainly focus on two areas, one concerns measuring the quality of an alignment using statistical models and the other concerns the development of search techniques in order to explore different alignments between two or more proteins.

1.1 Protein structure

Proteins are large biomolecules consisting of one or more polypeptides and play a vital role in all living organisms. To better describe the protein structure we first explain the structure of an amino acid.

Amino acids are the main *ingredients* of a protein molecule and there are about 20 different of them. Their structure representation can be seen in Figure 1.1. They consist of a main *Carbon - alpha* atom (Ca) in the centre, an amino and carboxyl group on either side and a side chain R which is connected to the Ca atom. The structure of the side chain determines the type and properties of each amino acid. For example if the structure of the side chain is just a hydrogen atom H then this amino acid will be *Glycine* (G), if it is a methyl group CH_3 then the amino acid will be *Alanine* (A). Each amino acid is classified based on their properties. There are two main big groups *hydrophobic*, which do not interact with water and *hydrophilic*, which interact with water. Then each of these two groups can be further divided into smaller groups such as aromatic, alkyl, basic, acidic etc.

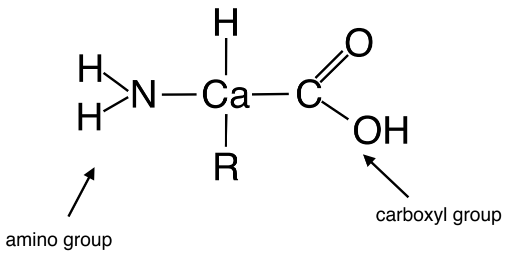


Figure 1.1: Amino acid structure.

When many amino acids form *peptide bonds* with each other they create polypeptides and when these polypeptides are folded under certain properties they determine the 3-d structure and functions for each protein molecule. A *peptide bond* is formed when the carboxyl of one amino acid is joined with the amino group of another resulting to a loss of a water molecule. An example of this chemical process, between two amino acids is shown in Figure 1.2. Each amino acid which is connected with a peptide bond is also often referred to as a *residue.*

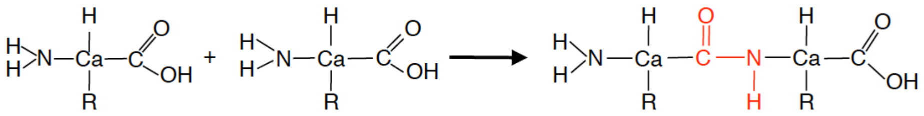


Figure 1.2: Polypeptide formation with peptide bond shown in red.

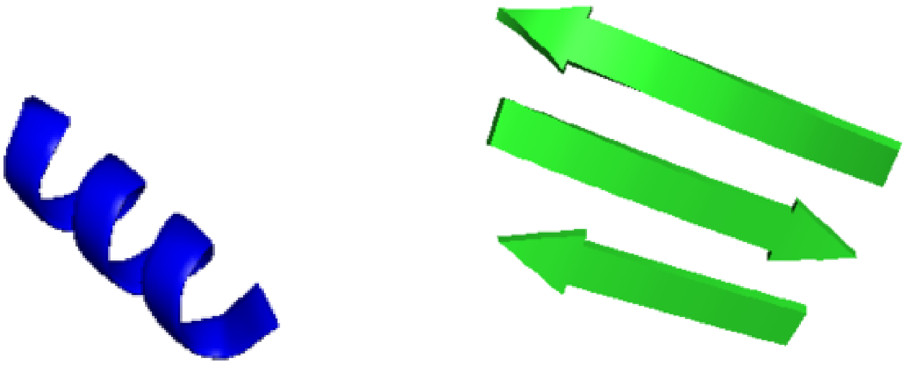
The protein structure is a collection of hundreds amino acids. It follows a specific hierarchy and can be described with the following four levels:

- **Primary structure**

  The primary structure of a protein is the 1-dimensional sequence of amino acids as in Figure 1.2. The final shape of the protein will depend on its primary structure since the type of each amino acid and their place in the sequence will determine the folding properties of the protein. We will also call the primary structure as the backbone of the protein.

- **Secondary structure**

  The secondary structure of a protein describes local formations of amino acid sequence. Each specific formation is a result of the rotations that are happening in the side chains R of each amino acid. The two most distinct patterns of the secondary structure are the α-*helix* and the β-*sheet* (Pauling et al., 1951). The α-*helix* formation occurs when the backbone chain folds into a spiral form with about 3-5 residues per turn. In a β-*sheet* formation the backbone of the protein chain extends in one way and returns back in a parallel formation where hydrogen bonds are now connecting the amino group from one amino acid with the carboxyl of another. An example of both of these patterns is shown in Figure 1.3



(a) α-helix (b) β-sheet

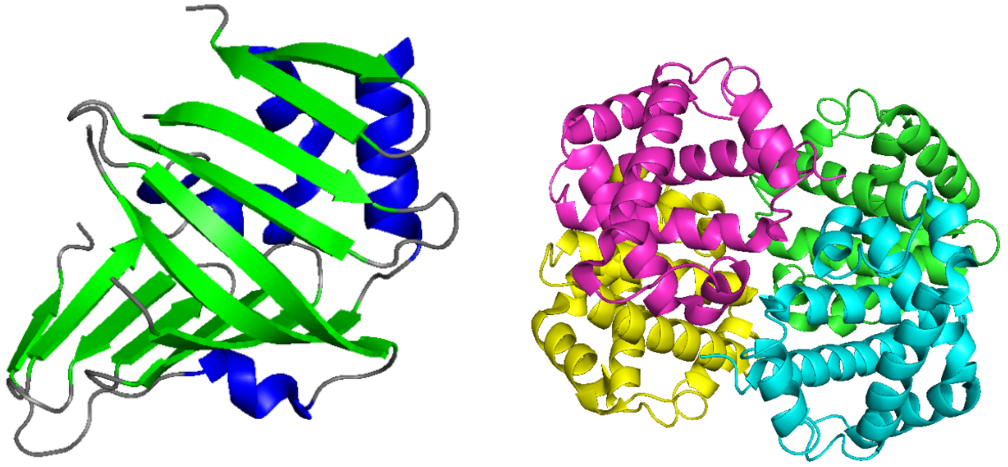
Figure 1.3: α-helix and β-sheet patterns.

- **Tertiary structure**

  The tertiary structure of a protein is the complete folding pattern of the protein backbone which determines its overall 3-dimensional shape. Each folding is specific to each protein and will happen in the same way every time that protein is formed and is directly related to its functions.

- **Quaternary structure**

  The quaternary structure is a selection of tertiary structures that fold together in order to form a larger protein. Figure 1.4 displays the tertiary and quaternary structure of two protein molecules.



(a) tertiary structure

(b) quaternary structure

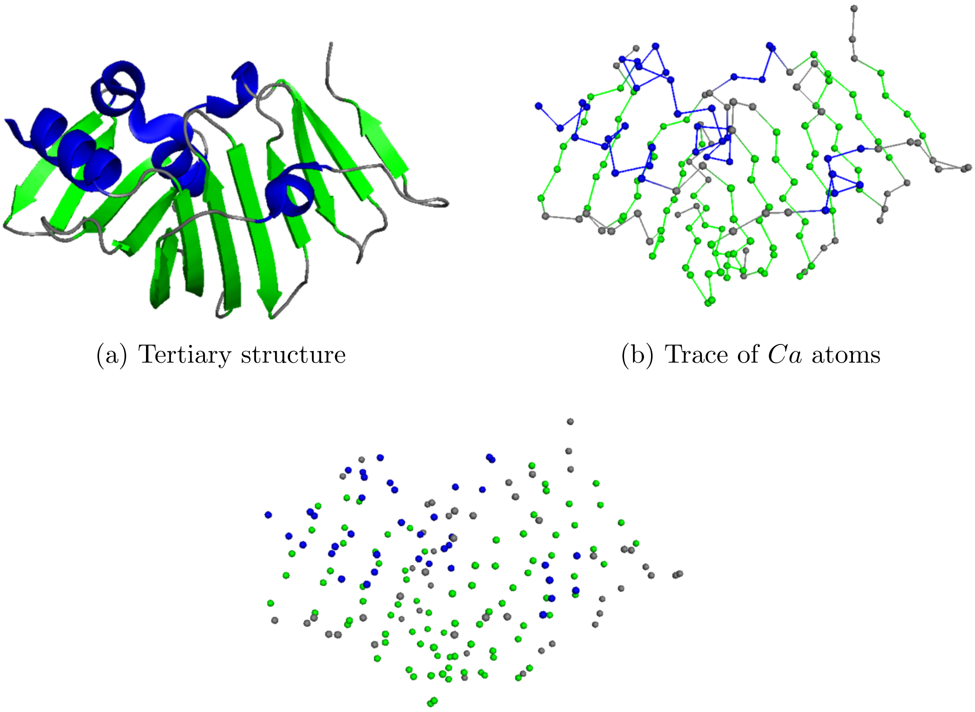
Figure 1.4: Tertiary and Quaternary structures of a protein.

1.2 Protein structural data

The two main techniques used in order to obtain the atomic and molecular structure of a protein molecule are *X-ray crystallography* and the *NMR spectroscopy*. In *X-ray crystallography* an x-ray source is aimed at the molecule and then the diffraction pattern created is studied in order to determine the 3-dimensional structure of a molecule. The *Nuclear Magnetic Resonance (NMR) spectroscopy* is a more complex technique for obtaining information regarding the structure and the dynamics of proteins. It consists of several phases and techniques which are based on the magnetic properties of each atom.

The Protein Data Bank (Berman et al., 2002) is a database which contains 3-dimensional structure data of proteins. It counts more than a hundred thousand different structures where most of them have been obtain using the methods described above. When the structure of a protein is determined using either of these two methods each atom is orbitally labelled. Most of the data we use in later Chapters have been obtained from this database.

Also for our purposes we use the tertiary structure of proteins and especially we use the 3-dimensional coordinates of the Ca atoms, since they can sufficiently describe the overall shape of the molecule. An example of how our data look can be seen in Figure 1.5



(a) Tertiary structure

(b) Trace of Ca atoms

(c) Location of Ca atoms

Figure 1.5: Tertiary structure and trace and locations of Ca atoms.

1.3 Structural alignment of proteins

In computational biology the alignment of protein structures has been one of the most important problems since the work of Rossmann and Argos (1978). Alignment of protein structures refers to finding a correspondence of amino acid between them, whereas structure comparison is focused on analysing the similarities between two or more structures. Structure alignment methods have started developing in the last 25 years and a review of the most recent methods can be found in Carugo (2007).

Many important tasks in biology rely in the comparison of protein structures. For example, the protein functionality is based both on the structure characteristics and the amino acid sequence information (Godzik et al., 2007). Also, another more recent problem, that of the prediction of a protein structure is based on structure alignment techniques to evaluate its prediction accuracy. Finally, protein classification databases such as SCOP (Andreeva et al., 2004) and CATH (Greene et al., 2006) rely on the results of structure comparison in order to categorize proteins into different families.

The use of structure alignment is often preferred to sequence alignment since the protein structure is much more conserved than the amino acid sequence (Chothia and Lesk, 1986). As noted also in Rost (1997) and Rost (1999), proteins with a sequence identity below 20% are very difficult to be aligned based only on their sequence information the structure alignment should be preferred.

In the book of Gu and Bourne (2009) the structural alignment problem is divided into three smaller ones:

1. Representation of the protein structures.
2. Sampling the possible alignments between two or more proteins.
3. Assessing the quality of a given alignment.

In Bioinformatics literature a plethora of fast and reliable structural alignment algorithms have been made available. Some of the most popular methods which we mention throughout the Thesis, include *DALI* (Holm and Sander, 1993), *CE* (Shindyalov and Bourne, 1998), *LGA* (Zemla, 2003) and *TMalign* (Zhang and Skolnick, 2005). Most of these methods are based on computational heuristic algorithms and the optimal alignment between two given proteins is proposed by either minimizing the overall distance between the molecules or maximizing a certain similarity score. In particular:

- DALI divides the protein structure into hexapeptides creating a matrix with the distances between all atoms and then uses a Monte Carlo simulation to estimate a score function for producing a final alignment.

- CE represents each structure as a set of distances between eight consecutive atoms and then uses a combinatorial extension algorithm to align atom pairs under a predefined threshold.

- TMalign uses the TMscore function and dynamic programming to assess the similarity between two molecules and to decide about the optimal alignment.

- LGA applies the *longest continuous segment* (LCS) and *global distance test* (GDT) algorithms in an iterative procedure to obtain the final matching under a predefined distance cut-off.

Most of these algorithms provide a framework that determines some alignment between two or sometimes more than two protein molecules in a fast and easy-to-implement way. However, one important aspect is that each method chooses to optimize a different score or distance metric in which the final alignment is based, lacking an overall modelling framework. Also, most of them do not allow for flexibility in choosing the parameters for each alignment which sometimes can be an issue since such predefined parameter values do not always adequately generate an optimal matching between two proteins. As mentioned in Koehl (2001) although many alignment algorithms can provide good results an overall score is needed for assessing the quality of each comparison.

Finally, we should note that each method has a primary *target* for its alignment. For example this can be that the final alignment should have a very low distance between the atoms of each molecule. To achieve this, some methods perform local alignments by matching only specific parts of each protein (secondary structures). On the other hand, the *target* can be to match as many atoms as possible between two proteins. This approach will have an effect on the final overall distance between the two proteins. Godzik (1996), showed that each alignment algorithm could produce different solutions especially when proteins with low sequence similarity are compared.

During the last years there have been developments in tracking the protein structure alignment problem from a statistical point of view. This approach consists of unlabelled shape analysis methods usually combined in a Bayesian framework. The problem of matching unlabelled 2-dimensional shapes has been studied in image analysis by Rangarajan et al. (1997) and Chui and Rangarajan (2003). Kent et al. (2004) use an EM approach to obtain a matching between two protein molecules. Dryden et al. (2007) and Schmidler (2007) developed a Bayesian model based on a procrustes likelihood to obtain an alignment between two proteins. The former uses a Metropolis sampling approach to determine possible matches, whereas the latter makes use of a geometric hashing algorithm. In another approach by Green and Mardia (2006) they use a full Bayesian model, assigning prior distributions in each transformation parameter. Finally, recent extensions include the work of Rodriguez and Schmidler (2014) and Fallaize et al. (2014) in which the sequence information is combined with the structure so that more biologically meaningful alignments are produced.

1.4 Aims and thesis outline

As we discussed in the previous Section, the adhoc algorithmic approach is fast and simple to use but lacks an overall modelling framework, whereas the Bayesian methods address this need but often is not so simple for the user since a lot of the parameters need to be pre determined. The aim of this Thesis is to bridge this gap by developing a method which is easily implemented by the user and at the same time is based in more robust modelling assumptions.

As we mentioned before the structure alignment problem can be divided into three different parts: *Representation*, *Optimization* and *Scoring*. In this Thesis we focus developing on last two of these. For the first part, in order to represent the two structures of a pair of molecules we use the 3-dimensional coordinates of the Ca atoms as shown in Section 1.2 and a match matrix M (for representing the atom correspondence) which we describe in the following Chapter.

The motivation for exploring the *Scoring* part comes from the fact that most of the Bioinformatics methods do not take into account any error that has been generated through the process of obtaining the structural data. Also, since most of the scores are distance based this leads to different results based on which parametrization of the distance score is used. The Bayesian approach solves this problem but many times the sampling techniques required for obtaining the posterior distributions make these methods difficult to implemented for the structure comparisons in protein databases. In our study, we present a scoring approach which is based on a size and shape likelihood function in order to asses the quality of a given match between two or more proteins. Using this method we select the best possible match for each atom while also considering any underling error.

The protein alignment is NP-hard problem (Lathrop, 1994), hence is it not computationally easy to explore all possible matching combinations. For the *Optimization* part we develop different search strategies in order to explore as much as possible of alignment space. In order to consider a new matched pair, our algorithms explore all possible matching combinations of atoms from a given starting point and make use of the Hungarian algorithm for obtaining an initial alignment.

1.4.1 Thesis structure

In Chapter 2 we present the general likelihood framework of our model by providing a brief introduction to different methods and techniques we will be using throughout the Thesis.

In Chapter 3 we establish the core likelihood framework in which our study is based. We present the estimation process of the unknown parameters in our model and also present the structural alignment algorithm that we will use to obtain the final matching between two proteins. The last part of this Chapter is about extending the likelihood (*scoring*) function with an adjustment in the alignment algorithm so that the optimal solutions preserve the amino acid sequence order. Finally, we extend our method for aligning simultaneously more than two molecules.

In Chapter 4 we use real and simulated data in order to asses the performance of the methods presented in Chapter 3. In particular, we compare our method with alternative ones using two benchmark datasets. In the last Section we show how our method can be adopted for estimating the evolutionary distance of two proteins.

In Chapter 5 we present an alignment approach based on the posterior modes estimation. Our method differs from the previous Bayesian approaches in the way we choose to assign the prior over the mean parameter and that we also choose to estimate directly the posterior mode of the matching distribution. Finally, we use simulated and real data to asses and to compare our approaches with alternative ones.

In Chapter 6 we present an extension of the likelihood method from Chapter 3. In this approach we use a Normal distribution to describe the whole molecule with a diagonal covariance structure allowing different variances between the matched and unmatched parts of a protein. We also compare all likelihood based approaches using real and simulated data.

In Chapter 7 we present a summary of the Thesis and discuss the contributions of our approach. Finally, we discuss possible improvements and areas of future work.

Chapter 2

Background Theory

2.1 Introduction

In this Chapter we explain the basic theoretical background needed for the protein alignment problem that is used throughout this Thesis. In Section 2.2 we explain some basic concepts of statistical shape analysis and how it can be connected to protein matching. In Section 2.3 a general likelihood framework for the estimation of the unknown parameters is described. In Section 2.4 we give a brief representation regarding the EM algorithm which is used in the later Chapters of the Thesis. Additionally, in Sections 2.5 and 2.6 we explain the parametrization of the rotation matrix and the Holonomic gradient method which is used later for integrating the rotation matrices form the likelihood function. Finally, in Sections 2.7 and 2.8 we focus on the protein alignment describing an assignment method called Hungarian algorithm and the similarity metrics we use to assess the quality of a matching between two proteins.

2.2 Shape models for protein alignment

We represent the geometrical information of protein molecules using 3 - dimensional configuration matrices. Let, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ be two configuration matrices of dimensions $m \times k$ and $m \times l$ respectively. For the rest of the Thesis we consider only $m = 3$ but most of the methodology can be applied to a general m. Each column of the matrices $\boldsymbol{X}_i$ consists of the 3-dimensional coordinates of the Ca atoms from the protein chain. The k, l landmarks have been labelled arbitrary and there is no prior knowledge for the correspondence between them. The objective is to obtain an alignment between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ under a common mean configuration after both configurations have been optimally rotated and translated.

2.2.1 Match matrix

Since no information is available for the correspondence between the atoms of the two protein molecules, in order to make inference about them we make use of a matching matrix $\boldsymbol{M}$ with dimensions $k \times l$ and which entries can only take the values of 1 and 0. There are many different definitions to matching matrices, but in this case we allow only one to one matches where each row and column of $\boldsymbol{M}$ can have at most one entry with 1 and the rest have to be 0. Then, for $1 \leq i \leq k$ and for $1 \leq j \leq l$, if $\boldsymbol{M}_{ij} = 1$, the i-th point of $\boldsymbol{X}_1$ is considered as a match to the j-th point of $\boldsymbol{X}_2$, and if $\boldsymbol{M}_{i\cdot} = 0$ or $\boldsymbol{M}_{\cdot j} = 0$, the i-th point of $\boldsymbol{X}_1$ and the j-th point of $\boldsymbol{X}_2$ do not have a match, where $\boldsymbol{M}_{i\cdot}$ and $\boldsymbol{M}_{\cdot j}$ represent the i-th row and the j-th column of $\boldsymbol{M}$ respectively. Other models also use this type of matching matrix, Green and Mardia (2006) and Fallaize et al. (2014) use a similar matrix with no duplicate matches between the landmarks, whereas Taylor et al. (2003) and Dryden et al. (2007) use a match matrix where multiple matches are allowed. In Mardia et al. (2012) both types of matching matrices are considered for inference.

2.2.2 Distributional assumptions and similarity transformations

For a given $\boldsymbol{M}$ each configuration matrix is partitioned into matched and unmatched parts. We refer to the matched parts $\boldsymbol{X}_1^M$ and $\boldsymbol{X}_2^M$ with dimensions $3 \times p$, where p is the number of matched landmarks between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. For each $\boldsymbol{X}_1^M$ and $\boldsymbol{X}_2^M$ the correspondence for each landmark is known and these matrices can be treated as the usual shape configurations. The unmatched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are defined as $\boldsymbol{X}_1^{-M}$ with dimensions $3 \times (k-p)$ and $\boldsymbol{X}_2^{-M}$ with dimensions $3 \times (l-p)$. For those matrices the correspondence between landmarks is unknown given $\boldsymbol{M}$.

Now, let us consider that the matched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are noisy observations from a common mean matrix $\boldsymbol{\mu}$ under a mapping of the *size and shape* transformations as

$$\mathcal{R}_1 \boldsymbol{\Delta}_1^M \boldsymbol{O}_1^M = \boldsymbol{\mu} + \epsilon_1 \qquad \mathcal{R}_2 \boldsymbol{\Delta}_2^M \boldsymbol{O}_2^M = \boldsymbol{\mu} + \epsilon_2 \tag{2.2.1}$$

with ϵ_i the errors, $\boldsymbol{\Delta}_i^M \boldsymbol{O}_i^M$ the *size and shape* variables of $\boldsymbol{X}_i^M$ and $\mathcal{R}_i$ represent the unknown size and shape transformations (Dryden and Mardia, 1998) defined as

$$\mathcal{R}_i = \left\{ \boldsymbol{R}_i \boldsymbol{X}_i + \tau 1_p^t : \boldsymbol{R}_i \in SO(3), \tau \in \mathbb{R}^3 \right\} \tag{2.2.2}$$

with $\boldsymbol{R}_i$ being a 3-dimensional rotation matrix and τ a $m \times 1$ translation vector.

Later, in the model described in Chapter 3 we assume a Normal distribution for the errors ϵ_i with zero mean and variance σ^2. In that case, the matched parts $\boldsymbol{X}_1^M$ and $\boldsymbol{X}_2^M$ can be treated as observations from a Normal distribution with common mean and variance as

$$(\mathcal{R}_1 \boldsymbol{\Delta}_1^M \boldsymbol{O}_1^M, \mathcal{R}_2 \boldsymbol{\Delta}_2^M \boldsymbol{O}_2^M) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2) \tag{2.2.3}$$

Also, we assume that the unmatched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are regarded as observations from a Uniform distribution

$$(\boldsymbol{\Delta}_1^{-M} \boldsymbol{O}_1^{-M}, \boldsymbol{\Delta}_2^{-M} \boldsymbol{O}_2^{-M}) \sim \mathrm{Unif}(V) \tag{2.2.4}$$

where V represents the volume in which the Uniform distribution is defined. The value of V can be defined in different ways, here we consider it as the volume of a cube that is big enough to contain both configuration matrices. Hence, our model can be regarded as a type of mixture model with a Normal distribution for the matched points and a Uniform for the unmatched, but we are not directly interested in estimating the proportion between the two mixtures rather than obtaining an optimal alignment between the landmarks of the two matrices, provided that we optimize over the unknown parameters.

This modelling approach is considered in Chapters 3, 4 and 5 whereas in Chapter 6 a different modelling framework is adopted, using a Normal likelihood for the whole configuration matrix $\boldsymbol{X}_i$.

2.3 General likelihood framework

In this Section we describe the general framework needed for obtaining an alignment between two or more protein molecules. Using the distributional assumptions of (2.2.3) and (2.2.4), and assuming independece between the matched and unmatched parts of a molecule $\boldsymbol{X}$ the likelihood function of the matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ will be the product between the matched and the unmatched densities as

$$\mathcal{L}(\boldsymbol{M},\boldsymbol{\mu},\sigma^2,\mathcal{R}_i|\boldsymbol{X}_1,\boldsymbol{X}_2,V) = f_M(\mathcal{R}_i\boldsymbol{\Delta}_i^M\boldsymbol{O}_i^M|\boldsymbol{\mu},\sigma^2,\boldsymbol{M}) \times f_{-M}(\boldsymbol{\Delta}_i^{-M}\boldsymbol{O}_i^{-M}|\boldsymbol{M},V) \tag{2.3.1}$$

where, $\mathcal{R}_i$ are the unknown size and shape transformations of (2.2.2) and $\boldsymbol{\Delta}_i^M\boldsymbol{O}_i^M$, $\boldsymbol{\Delta}_i^{-M}\boldsymbol{O}_i^{-M}$ the size and shape variables of $\boldsymbol{X}_i^M$ and $\boldsymbol{X}_i^{-M}$ respectively. Our aim is to maximize the likelihood function of (2.3.1) under the unknown parameters $\boldsymbol{M},\boldsymbol{\mu},\sigma^2$ and $\mathcal{R}_i$. The volume parameter V is initially considered as fixed and a discussion of its effect is included in Chapter 4. For the unknown parameters, the joint estimation is not straightforward since the likelihood space defined is both continuous in the parameters $\boldsymbol{\mu}$ and σ^2 and discrete in matching matrix $\boldsymbol{M}$. Our interest is mainly to obtain the likelihood mode of $\boldsymbol{M}$ which will give us the optimal alignment between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. In order to achieve this, first we need to estimate $\boldsymbol{\mu}$ and σ^2, but this estimation is depending on $\boldsymbol{M}$ since it imposes the current correspondence between the atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Hence, the likelihood of (2.3.1) can also be written as a function of $\boldsymbol{M}$ as:

$$\mathcal{L}_{\dagger}(\boldsymbol{M}) = \mathcal{L}(\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}, \boldsymbol{M}|\boldsymbol{X}_1, \boldsymbol{X}_2) \tag{2.3.2}$$

Then finding the mode of $\boldsymbol{M}$ will depend on the following two step optimization:

$$\hat{\boldsymbol{M}} = \arg\max_{\boldsymbol{M}} \left[\hat{\mathcal{L}}_{\dagger}(\boldsymbol{M})\right] \tag{2.3.3}$$

where

$$\hat{\mathcal{L}}_{\dagger}(\boldsymbol{M}) = \arg\max_{\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}} \mathcal{L}(\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}, \boldsymbol{M}|\boldsymbol{X}_1, \boldsymbol{X}_2) \tag{2.3.4}$$

As we can see the problem of estimating the mode of $\boldsymbol{M}$ can be divided into two smaller optimization problems. First optimizing over $\hat{\boldsymbol{\mu}}$ and $\hat{\sigma}^2$ for a given alignment $\boldsymbol{M}$ and we discuss this in Sections 3.2-3.3. Second optimizing $\hat{\boldsymbol{M}}$ for which different techniques are presented in Sections 3.4, 3.5 and 3.6. These two optimization steps need to be implemented simultaneously since for any updated $\boldsymbol{M}$ a new pair of $\boldsymbol{\mu}_{(\boldsymbol{M})}$ and $\sigma^2_{(\boldsymbol{M})}$ needs to be calculated.

Another important aspect of this problem is the variations of the density for the matched parts $f_M(\cdot)$. In statistical protein alignment literature the common practice is to use a Normal distribution where two different versions exist. The first is an asymmetrical approach used by Dryden et al. (2007), Schmidler (2007) and Rodriguez and Schmidler (2014), where Procrustes estimation for the optimal rotation and translation of $\boldsymbol{X}_1$ to $\boldsymbol{X}_2$ is used. The second version considered by Green and Mardia (2006), Mardia et al. (2013) and Fallaize et al. (2014) is a symmetrical approach where now the rotation and translation are included as unknown parameters in the model. A comparison between the two approaches is made by Kenobi and Dryden (2012) where they found out that depending on the value of σ^2 each approach performs better than the other.

In Chapter 3 we define our own version of $f_M(\cdot)$, which can be considered as a combination of both previous definitions, since we estimate the rotation and translation parameters from our data but we also keep the symmetry using a common mean between $\boldsymbol{X}_1^M$ and $\boldsymbol{X}_2^M$. Finally, in Chapter 3 we explore three different versions for $f_M(\cdot)$, one including only the geometrical information of $\boldsymbol{X}_1^M$ and $\boldsymbol{X}_2^M$, another one using the geometrical and sequence information and last one including a gap penalty function.

2.4 EM algorithm

The Expectation - Maximization (EM) algorithm developed by Dempster et al. (1977) is an iterative optimization method for obtaining the maximum likelihood estimation of parameters when part of the data are incomplete or unobserved.

Consider the set of the full data $\boldsymbol{Y} = (\boldsymbol{X}, \boldsymbol{Z})$ where $\boldsymbol{X}$ is the partially observed data and $\boldsymbol{Z}$ the missing or unobserved data. Assuming that $\mathcal{L}(\theta|\boldsymbol{X})$ is the partial data likelihood where θ is an unknown parameter, then the maximum likelihood estimate of θ using the EM will be obtained by maximizing iteratively a function $\mathcal{Q}(\theta|\theta^t)$ with the following steps:

- Expectation step : $\mathcal{Q}_{\theta^t}(\theta|\theta^t) = \mathbb{E}_{(\boldsymbol{Y}|\theta^t)}\left[\log \mathcal{L}(\theta|\boldsymbol{X}, \boldsymbol{Z})\right]$
- Maximization step : $\theta^{t+1} = \arg\max_{\theta} \mathcal{Q}_{\theta^t}(\theta|\theta^t)$

where $\mathcal{L}(\theta|\boldsymbol{X}, \boldsymbol{Z})$ is the full data likelihood. The algorithm iterates among these steps until a convergence criterion is reached. The EM algorithm guarantees that in each step the likelihood $\mathcal{L}(\theta|\boldsymbol{X})$ will increase monotonically and a local maximum mode of θ will be reached at the convergence. In order to see this, note that at the $t-th$ iteration the likelihood $\mathcal{L}(\theta|\boldsymbol{X})$ can be written as

$$\mathcal{L}(\theta^t|\boldsymbol{X}) = \mathcal{Q}_{\theta^t}(\theta|\theta^t) - \mathcal{H}(\theta|\theta^t)$$

where $\mathcal{H}(\theta|\theta^t) = \mathbb{E}_{(\boldsymbol{Z}|\theta^t)}\left[\log f(\boldsymbol{Z}|\boldsymbol{X}, \theta^t)\right]$. Since log is a concave function, using Jensen's inequality

$$\mathcal{H}(\theta^{t+1}|\theta^t) \geq \mathcal{H}(\theta|\theta^t)$$

and considering that θ^{t+1} maximizes $\mathcal{Q}_{\theta^t}(\theta|\theta^t)$ one can see that

$$\mathcal{L}(\theta^{t+1}|\boldsymbol{X}) \geq \mathcal{L}(\theta^t|\boldsymbol{X})$$

As a result, it is guaranteed that through maximizing the function $\mathcal{Q}(\cdot)$ we also maximize locally the likelihood function $\mathcal{L}(\theta|\boldsymbol{X})$. However, the EM algorithm is sometimes sensitive to starting point selection and is not always possible to reach the global maximum especially if the algorithm reaches a saddle point.

Previous work on protein matching using the EM has been done by Taylor et al. (2003), Kent et al. (2004) where they consider applications of protein matching both in 2 and 3 dimensions and the missing data are the probabilities of the matching between the landmarks. An extension of this method was later developed by Mardia et al. (2012) where they consider an application of matching protein gels in 2 dimensions.

2.5 Rotation matrix parametrization

One important issue regarding the protein alignment problem from a statistical point of view is the estimation of the rotation matrix. In three dimensional space Raffenetti and Ruedenberg (1969) and Khatri and Mardia (1977) showed that a rotation matrix can be represented using the Euler angles θ as follows

$$R = R_1(\theta_1)R_2(\theta_2)R_3(\theta_3)$$

where $R_1(\theta_1), R_2(\theta_2), R_3(\theta_3)$ are i.i.d rotations defined as

$$R_1(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \qquad R_2(\theta_2) = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix}$$

$$R_3(\theta_3) = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

One approach, as used in Dryden and Mardia (1998) and Rodriguez and Schmidler (2014) is to estimate the rotation matrix using the Procrustes registration. The alternative way, as described in Green and Mardia (2006) is to consider R as an unknown parameter. In this case, in order to derive the posterior distribution a Gibbs sampler is implemented for updating the rotation angles θ using a conjugate matrix Fisher distribution as a prior. For a definition of these distributions see Downs (1972) and Mardia and Jupp (2009).

In our approach we parametrize the rotation matrix using unit quaternions (Moran, 1975; Wood, 1993; Prentice, 1984). A unit quaternion $x = \{x_1, x_2, x_3, x_4\}$ is considered as a point in a 4-dimensional unit sphere $S_3 = \{x : x \in \mathbb{R}^4, xx^t = 1\}$ and a 3-dimensional rotation matrix can be derived as

$$R = \begin{bmatrix} x_1^2 + x_4^2 - x_2^2 - x_3^2 & 2x_1x_2 - 2x_3x_4 & 2x_1x_3 + 2x_2x_4 \\ 2x_1x_2 + 2x_3x_4 & x_2^2 + x_4^2 - x_1^2 - x_3^2 & 2x_2x_3 - 2x_1x_4 \\ 2x_1x_3 - 2x_2x_4 & -2x_2x_3 + 2x_1x_4 & x_3^2 + x_4^2 - x_1^2 - x_2^2 \end{bmatrix} \qquad (2.5.1)$$

The representation (2.5.1) for the uniformly distributed X in S_3, leads to an one-to-one relationship with the Bingham distribution on S_4 and the matrix Fisher distribution on $SO(3)$(Prentice, 1984) and as is later described in Chapter 3, our likelihood evaluation depends on the estimation of the normalizing constant of the Bingham distribution in S_4.

2.6 Holonomic gradient method

In this Section we briefly describe the Holonomic gradient method which used in Chapter 3 for the derivation of the normalizing constant of the Bingham distribution. For more details regarding the Holonomic gradient method and its relation to the calculation of the normalizing constant of the Bingham distribution see Sei et al. (2010) and Sei and Kume (2015). Also, Fallaize and Kypraios (2016) has considered the same problem under a Bayesian framework.

Before we define the Holonomic gradient method we need to define what a holonomic function is. A function f is called *holonomic* if there exist non-zero polynomials as

$$p_0(x)f(x) + p_1(x)f^{'}(x) + \cdots + p_r(x)f^{(r)}(x) = 0 \tag{2.6.1}$$

where $f^{(r)}$ are the derivatives of r-order. Now let $\boldsymbol{a} = (a_1, \ldots, a_d) \in \Theta$ and $f(\boldsymbol{a})$ be a holonomic function, with Θ being a subset of the d-dimensional Euclidean space and $\boldsymbol{g}(\boldsymbol{a})$ a column vector of the partial derivatives of $f(\boldsymbol{a})$. Then since $f(\boldsymbol{a})$ is a holonomic function $\boldsymbol{g}(\boldsymbol{a})$ will satisfy (Sei et al., 2010) the following system of linear partial differential equations

$$\partial_i \boldsymbol{g}(\boldsymbol{a}) = \boldsymbol{P}_i(\boldsymbol{a})\boldsymbol{g}(\boldsymbol{a}) \tag{2.6.2}$$

where $\boldsymbol{P}_i(\boldsymbol{a})$ is a square matrix of rational functions. We call the equation (2.6.2) the *Pfaffian system* of $\boldsymbol{g}$.

The *Holonomic Gradient Method* (HGM) is an algorithm for evaluating a particular value of a holonomic function for a local optima $\boldsymbol{a}$. Assume that $\boldsymbol{g}(\boldsymbol{a}^{(0)})$ is given for some point $\boldsymbol{a}^{(0)} \in \Theta$. Let $\bar{\boldsymbol{a}}(t), \quad t \in [0,1]$, be a smooth curve in Θ, such that $\bar{\boldsymbol{a}}(0) = \boldsymbol{a}^{(0)}$ and $\bar{\boldsymbol{a}}(1) = \boldsymbol{a}^{(1)}$. Also, define $\bar{\boldsymbol{g}}(t) = \boldsymbol{g}(\bar{\boldsymbol{a}}(t))$. Then $\bar{\boldsymbol{g}}(t)$ is the solution of the ordinary differential equation below

$$\frac{d}{dt}\bar{\boldsymbol{g}}(t) = \sum_{i=1}^{d} \frac{d\bar{a}_i(t)}{dt} \boldsymbol{P}_i(\bar{\boldsymbol{a}}(t))\bar{\boldsymbol{g}}(t) \tag{2.6.3}$$

for a given starting point of $\bar{\boldsymbol{g}}(0) = \boldsymbol{g}(\boldsymbol{a}^{(0)})$.

Then the HGM algorithm can be described with the following steps

1. Solve numerically the ODE (2.6.3) over $t \in [0, 1]$.

2. Return $\boldsymbol{g}(\boldsymbol{a}^{(1)})$

Later in Section 3.3.2 our optimization of the $\boldsymbol{\mu}$ and σ^2 parameters is directly connected to the evaluation of the normalizing constant from the Bingham distribution. Hence, we make use of the HGM and the relevant work shown in the papers of Dryden et al. (2015) and Sei and Kume (2015) to calculate the normalizing constant and obtain the maximum likelihood estimators of $\boldsymbol{\mu}$ and σ^2.

2.7 Hungarian algorithm

The Hungarian algorithm was developed by Kuhn (1955) and is an optimization method for providing a solution to the assignment problem. In matrix interpretation the assignment problem involves a cost matrix of n workers and n available tasks, with a cost for each worker to be assigned in each task. The Hungarian method tries to provide an optimal assignment by assigning each worker to one task while minimizing the overall cost. It can be performed using the following steps :

Hungarian algorithm

1. Subtract the smallest element in each row from all the elements of this row.

2. Subtract the smallest element in each column from all the elements of this column.

3. *Cover* all zeros in the matrix using a minimum number of horizontal and vertical lines.

4. If the minimum covered number of rows is n the assignment is possible, otherwise go to step 5.

5. Find the smallest element that is not covered by a line, subtract from all the elements that are uncovered and add it to the elements that are covered twice, then go to step 3.

The corresponding steps can be easily seen in the following simple example where we have four workers a, b, c, d to perform 4 tasks with costs a_i, b_i, c_i, d_i, $i = 1, \ldots, 4$ for each worker to perform each task:

$$\text{Workers}\overset{\text{Tasks}}{\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix}} \xrightarrow{\text{Step 1}} \text{Workers}\overset{\text{Tasks}}{\begin{bmatrix} a'_1 & a'_2 & 0 & a'_4 \\ b'_1 & 0 & b'_3 & b'_4 \\ c'_1 & c'_2 & 0 & c'_4 \\ 0 & d'_2 & d'_3 & d'_4 \end{bmatrix}} \xrightarrow{\text{Step2}} \text{Workers}\overset{\text{Tasks}}{\begin{bmatrix} a'_1 & a'_2 & 0 & a'_4 \\ b'_1 & 0 & b'_3 & b'_4 \\ c'_1 & c'_2 & 0 & c'_4 \\ 0 & d'_2 & d'_3 & 0 \end{bmatrix}}$$

$$\xrightarrow{\text{Step 3}} \text{Workers}\overset{\text{Tasks}}{\begin{bmatrix} a'_1 & a'_2 & \mathbf{0} & a'_4 \\ \mathbf{b'_1} & \mathbf{0} & \mathbf{b'_3} & \mathbf{b'_4} \\ c'_1 & c'_2 & \mathbf{0} & c'_4 \\ \mathbf{0} & \mathbf{d'_2} & \mathbf{d'_3} & \mathbf{0} \end{bmatrix}} \xrightarrow{\text{Step 4}} \text{Workers}\overset{\text{Tasks}}{\begin{bmatrix} a'_1 & a'_2 & 0 & a'_4 \\ b'_1 & 0 & b'_3 & b'_4 \\ 0 & c'_2 & 0 & c'_4 \\ 0 & d'_2 & d'_3 & 0 \end{bmatrix}} \xrightarrow{\text{Step 5}} \text{Workers}\overset{\text{Tasks}}{\begin{bmatrix} \mathbf{a'_1} & \mathbf{a'_2} & \mathbf{0} & \mathbf{a'_4} \\ \mathbf{b'_1} & \mathbf{0} & \mathbf{b'_3} & \mathbf{b'_4} \\ \mathbf{0} & \mathbf{c'_2} & \mathbf{0} & \mathbf{c'_4} \\ \mathbf{0} & \mathbf{d'_2} & \mathbf{d'_3} & \mathbf{0} \end{bmatrix}}$$

Then the optimal assignment will be :a_3, b_2, c_1, d_4. When a matrix with different number of rows and columns is available, as usually is the case when we have data from protein molecules extra zero rows and columns can be added so the cost matrix can be square.

2.8 Protein similarity metrics

The purpose of aligning protein molecules is to find a solution that minimizes the final distance between them and at the same time match as many atoms as possible. However, it is not easy to compare different alignments especially the ones with similar characteristics, since there has been no evidence so far suggesting an analogy between the number of matched atoms and the final distance of the proteins. Due to the nature of the problem, it can be deduced that the more atoms are matched, the bigger the final distance between the molecules will be.

In the literature of structural bioinformatics, the distance between the molecules is usually measured in Angstroms expressed by Å and many metrics exist which attempt to quantify the aforementioned uncertainty and produce a total number that is comparable between two or more alignment solutions. The most popular metric used in Bioinformatics is the *Root Mean Square Deviation* (RMSD), which is the average distance of the matched atoms between two aligned proteins and is defined as

$$RMSD(\boldsymbol{X}_1^M, \boldsymbol{X}_2^M) = \sqrt{\frac{1}{p}\sum_{i=1}^{p} ||\boldsymbol{X}_1^M - \boldsymbol{X}_2^M||^2} \tag{2.8.1}$$

where p is the number of matched atoms between the aligned parts of the proteins $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ and $||\cdot||$ is the Euclidean distance.

The RMSD metric has 0 as lower bound, with optimal solutions being closer to 0. The major advantage of the RMSD is that it is very easy to use and to explain, but still it does not take into account other parameters of the final alignment, such as the total protein length or the proportion of atoms matched from the whole protein chain. Hence, sometimes RMSD favours solutions with fewer matched atoms as they will have a lower RMSD value and is very sensitive to points that have been mismatched or matched with large distance.

Another popular metric used, is the *Template Modelling Score* (TMscore) developed by Zhang and Skolnick (2004). This metric is more robust than RMSD because it also takes into account the number of atoms that have been matched and the total length of the protein. It takes values between 0 and 1, with 1 providing the optimal solution. As described in Xu and Zhang (2010), TMscore values of ≥ 0.5 indicate proteins that have a high probability of belonging in the same fold, whereas values of TMscore ≤ 0.2 indicate unrelated proteins. The TMscore is defined as

$$TMscore(\boldsymbol{X}_1^M, \boldsymbol{X}_2^M) = \max\left(\frac{1}{N_{max}}\sum_{i=1}^{p}\frac{1}{1+\left(\frac{d_i}{d_0(N_{max})}\right)}\right) \tag{2.8.2}$$

where p is the number of matched points between the aligned proteins $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, d_i is the distance of the i-th pair, N_{max} is the length of the largest protein molecule and $d_0(N_{max}) = 1.24\sqrt[3]{p-15} - 1.8$.

The final measure that we are going to use for our comparisons is the Structure Overlap (SO %) defined as the proportion of aligned atoms from the whole protein chain that are within a distance d_0 after the two molecules have been optimally rotated, and given by the equation

$$SO = 100 \times \frac{1}{z}\sum_{d_{ij}\leq d_0}^{p} 1 \tag{2.8.3}$$

where z is the smallest number of atoms between the two protein molecules, p is the number of matched atoms, d_{ij} is the Euclidean distance between atoms i and j and d_0 is the cut-off distance usually taking the value 3.5Å. The Structure Overlap can give a sense of how much from the whole protein chain has been closely matched.

Chapter 3

Likelihood alignment and extensions

3.1 Introduction

In this Chapter we describe a likelihood based method for the structural alignment of protein molecules. Section 3.2 presents a size and shape likelihood density based on the theoretical background from Chapter 2. The likelihood density is introduced in Section 2.3 explained with more details in this Chapter defines our core modelling approach in which the rest of the Chapter and this Thesis is based on. Section 3.3 is about the first optimization step (2.3.4). We present an EM algorithm in order to estimate the unknown parameters of mean and variance. We also discuss the concept of how to evaluate the likelihood density when the rotation parameter is integrated out and the connection with the normalizing constant of the Bingham distribution. Section 3.4 is about the second optimization step of (2.3.3), in order to obtain a likelihood mode for the matching matrix $\boldsymbol{M}$. We describe a structural alignment algorithm for protein molecules which is using the Hungarian method from Section 2.6.

In Sections 3.5 and 3.6, we discuss extensions of our likelihood model with the inclusion of sequence information and a penalty function for penalizing gaps in the sequence order. In Section 3.7, we discuss the effect of starting points and present an algorithm which automatically selects a set of starting points when user input is not available. Finally, in Section 3.8 we extend the previous method by simultaneously aligning more than two molecules. We also discuss the limitations of this approach and present an alternative matching algorithm when aligning many molecules at the same time is required.

3.2 Size and shape density

Consider two protein molecules represented by the configuration matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, with dimensions $3 \times k$ and $3 \times l$ respectively. As seen from Chapter 2 the likelihood density of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ will be the product of the matched and unmatched parts as

$$\mathcal{L}(\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2 | \boldsymbol{X}_1, \boldsymbol{X}_2) = f_M(\boldsymbol{X}_1^M, \boldsymbol{X}_2^M | \boldsymbol{M}, \boldsymbol{\mu}, \sigma^2) f_{-M}(\boldsymbol{X}_1^{-M}, \boldsymbol{X}_2^{-M} | \boldsymbol{M}) \quad (3.2.1)$$

In addition, we consider the Singular Value Decomposition of matrix $\boldsymbol{X}_i$ as

$$\boldsymbol{X}_i = \boldsymbol{R}_i \boldsymbol{\Delta}_i \boldsymbol{O}_i \quad (3.2.2)$$

with $\boldsymbol{R}_i \in SO(3)$ a matrix with dimensions 3×3, $\boldsymbol{O}_i$ a matrix with dimensions $3 \times p$ where $\boldsymbol{O}_i \boldsymbol{O}_i^t \in SO(3)$ and $\boldsymbol{\Delta}_i = diag(\lambda_1, \lambda_2, \lambda_3)$ a diagonal matrix, in which λ_j are the eigenvalues of $\boldsymbol{X}_i \boldsymbol{X}_i^t$. Then under the Lebesgue measure, $d\boldsymbol{X}_i$ can be decomposed as shown in Muirhead (2009) and Diaz-Garcia et al. (1997) as

$$d\boldsymbol{X}_i \propto d\boldsymbol{R}_i d\boldsymbol{\Delta}_i d\boldsymbol{O}_i \quad (3.2.3)$$

where $d\boldsymbol{\Delta}_i = \prod_{j=1}^3 \prod_{j=1}^3 \lambda_j^{(k-2)/2} \prod_{j>r} (\lambda_j - \lambda_r) \prod_{j=1}^3 d\lambda_j$. As a result, each matrix $\boldsymbol{\Delta}_i \boldsymbol{O}_i$ can be considered as the size and shape variables of the $\boldsymbol{X}_i$ in the corresponding space (Kendall et al., 2009). Hence, the $\boldsymbol{\Delta}_i \boldsymbol{O}_i$ will represent the observed $\boldsymbol{X}_i$ under some unknown and unobserved rotations $\boldsymbol{R}_i$.

In the general case when the transformation parameters of rotation and translation are known and by using the modelling framework described in Section 2.2 the density function of the matched parts can be described using a Normal distribution as:

$$f_M(\boldsymbol{X}_i^M|\boldsymbol{M},\boldsymbol{\mu},\sigma^2)=\left(\frac{1}{2\pi\sigma^2}\right)^{3p}\exp\left\{-\frac{\sum\limits_{i=1}^{2}||\boldsymbol{X}_i^M-\boldsymbol{\mu}||^2}{2\sigma^2}\right\} \tag{3.2.4}$$

with an alignment given by the match matrix $\boldsymbol{M}$. Similarly, the density function for the unmatched parts is described by a Uniform distribution as:

$$f_{-M}(\boldsymbol{X}_i^{-M}|\boldsymbol{M})=\left(\frac{1}{V}\right)^{k+l-2p} \tag{3.2.5}$$

where p is the number of matched residues between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ and V is the volume of a space that includes both molecules.

In order to be able to make statistical inference, the likelihood density of (3.2.1) needs to be invariant under the similarity transformations of (2.2.2). Since, the unknown parameters of $\boldsymbol{\mu}$ and σ^2 appear only in the density of the matched part of each $\boldsymbol{X}_i$ we only need to remove the location information from the $\boldsymbol{X}_1^M$ and $\boldsymbol{X}_2^M$, hence we multiply each of them with the *Helmert sub-matrix* with dimensions $(p-1)\times p$. This matrix is a special case of the full orthogonal *Helmert matrix* with dimensions $p\times p$ when the first row is removed. The *Helmert sub-matrix* $\boldsymbol{H}$ is defined as

$$\boldsymbol{H}=\begin{bmatrix}-1/\sqrt{2} & 1/\sqrt{2} & 0 & \dots & 0\\ -1/\sqrt{6} & 1/\sqrt{6} & 2/\sqrt{6} & \dots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ -1/\sqrt{p(p-1)} & -1/\sqrt{p(p-1)} & -1/\sqrt{p(p-1)} & \dots & (p-1)/\sqrt{p(p-1)}\end{bmatrix} \tag{3.2.6}$$

The new landmarks $\boldsymbol{X}_i^h=\boldsymbol{H}\boldsymbol{X}_i$ are called *Helmertized landmarks* (Dryden and Mardia, 1998) and are invariant under the translation information. Different choices are available for creating landmarks that are invariant under location, i.e. set one landmark of $\boldsymbol{X}_i$ to zero and the rest as the differences to this landmark. However, *Helmertized landmarks* are chosen because the covariance matrix of the transformed coordinates remain the same as the original, since the matrix $\boldsymbol{H}$ is orthogonal. For simplicity in notation, the configuration matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are assumed to be in the *Helmertized landmarks* for the rest of this Chapter.

Finally, in order to derive the marginal size and shape density under the Normal distribution of $d\boldsymbol{X}_i$ we need to integrate the rotation parameter out of (3.2.1). This leads to the size and shape density of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ with rotation and translation invariance as follows:

$$\begin{aligned}
f_S(\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2) &= \int\limits_{\boldsymbol{R}_i \in SO(3)} \mathcal{L}(\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2|\boldsymbol{X}_1^\star, \boldsymbol{X}_2^\star) d\boldsymbol{R}_i \\
&= \int\limits_{\boldsymbol{R}_i \in SO(3)} \left(\frac{1}{2\pi\sigma^2}\right)^{3p} \exp\left\{-\frac{\sum_{i=1}^{2} ||\boldsymbol{R}_i\boldsymbol{X}_i^M - \boldsymbol{\mu}||^2}{2\sigma^2}\right\} \left(\frac{1}{V}\right)^{k+l-2p} d\boldsymbol{R}_i \\
&= \left(\frac{1}{V}\right)^{k+l-2p} \left(\frac{1}{2\pi\sigma^2}\right)^{3p} \exp\left\{-\frac{\sum_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}||^2}{2\sigma^2}\right\} \\
&\quad \times \prod_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} \exp\left\{\frac{-\mathrm{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i^M\boldsymbol{\mu}^t\right)}{\sigma^2}\right\} d\boldsymbol{R}_i \qquad (3.2.7)
\end{aligned}$$

where, $\boldsymbol{X}_1^\star$ and $\boldsymbol{X}_2^\star$ are full unobserved Normal data (which include the unknown rotation and translation parameters) and $d\boldsymbol{R}_i$ is the Haar measure in $SO(3)$. This size and shape density is similar to the one obtained from Goodall and Mardia (1992) using the QR decomposition.

3.3 Optimizing over the unknown parameters $\boldsymbol{\mu}$ and σ^2

Using the decomposition described in (3.2.3), each $\boldsymbol{X}_i$ can be regarded as the partially observed size and shape data. The missing rotations $\boldsymbol{R}_i$ can be estimated using the Expectation - Maximization (EM) algorithm (Dempster et al., 1977) from Section 2.4. The *log-likelihood* function of the complete data $\boldsymbol{X}_i$ for a given alignment $\boldsymbol{M}$, with known similarity transformations $\mathcal{R}_i$ can be defined as

$$\begin{aligned} l(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{M}, \boldsymbol{X}_1^\star, \boldsymbol{X}_2^\star) &= \log \mathcal{L}(\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2 | \boldsymbol{X}_1^\star, \boldsymbol{X}_2^\star) \qquad (3.3.1) \\ &= -3p \log(2\pi\sigma^2) - (k + l - 2p)\log(V) - \frac{1}{2\sigma^2} \sum_{i=1}^{2} ||\boldsymbol{X}_i^M - \boldsymbol{\mu}||^2 \end{aligned}$$

3.3.1 EM steps

Using the EM algorithm we can *estimate* the missing rotations $\boldsymbol{R}_i$ and be able to make inference for the unknown parameters of $\boldsymbol{\mu}$ and σ^2 by iteratively applying the following steps

- *Expectation step* : Evaluate the function $Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})$ for given values of $\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1}$ finding the expectation over the missing rotations $\boldsymbol{R}_i$:

  $$Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1}) = \mathbb{E}_{\boldsymbol{R}_i | \boldsymbol{X}_i} \left[l(\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1} | \boldsymbol{M}, \boldsymbol{X}_1^\star, \boldsymbol{X}_2^\star) \right]$$

  where

  $$\begin{aligned} \mathbb{E}_{\boldsymbol{R}_i | \boldsymbol{X}_i} \left[l(\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1} | \boldsymbol{M}, \boldsymbol{X}_1^\star, \boldsymbol{X}_2^\star) \right] &= -3p \log \sigma^2_{t-1} - \frac{\sum_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}_{t-1}||^2}{2\sigma^2_{t-1}} \\ &+ \sum_{i=1}^{2} \log \int_{\boldsymbol{R}_i \in SO(3)} \exp\left\{ \frac{\operatorname{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}_{t-1}^t \right)}{\sigma^2_{t-1}} \right\} d\boldsymbol{R}_i \\ &+ \log C \end{aligned}$$

  with $C = 3pV^{-(k+l-2p)} \log(2\pi)$.

- *Maximization step* : Maximize the function $Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})$ with respect to $\boldsymbol{\mu}$ and σ^2 as

  $$\frac{\partial Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})}{\partial \boldsymbol{\mu}} = 0 \qquad \frac{\partial Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})}{\partial \sigma^2} = 0$$

The new updated values of $\boldsymbol{\mu}_t, \sigma_t^2$ as also mentioned in the paper of Dryden et al. (2015) will be

$$\hat{\boldsymbol{\mu}} = \frac{1}{2}\sum_{i=1}^{2}\left[\frac{\displaystyle\int\limits_{\boldsymbol{R}_i \in SO(3)} \boldsymbol{R}_i \boldsymbol{X}_i^M e^{A_i} d\boldsymbol{R}_i}{\displaystyle\int\limits_{\boldsymbol{R}_i \in SO(3)} e^{A} d\boldsymbol{R}_i}\right]$$

$$\hat{\sigma}^2 = \frac{1}{6p}\left[\sum_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 - ||\hat{\boldsymbol{\mu}}||^2\right]$$

with $A_= - \frac{\mathrm{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}^t\right)}{\sigma^2}$.

Alternating the algorithm between the *Expectation* and *Maximization* steps, in the t-th iteration the updated parameters of $\boldsymbol{\mu}$ and σ^2 will be given as

$$\boldsymbol{\mu}_t = \hat{\boldsymbol{\mu}}, \qquad \sigma_t^2 = \hat{\sigma^2}$$

The convergence of the algorithm is achieved when the following condition is satisfied : $Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_t, \sigma_t^2) - Q(\boldsymbol{\mu}, \sigma^2 | \boldsymbol{\mu}_{t-1}, \sigma_{t-1}^2) \leq \epsilon$, where ϵ is some predefined tolerance level.

3.3.2 Rotation integration

An important part of the *Expectation step* concerns the computation of the integral over the missing rotations $\boldsymbol{R}_i$. Using the decomposition of $\boldsymbol{X}_i$ from (3.2.3) where $\boldsymbol{X}_i = \boldsymbol{\Delta}_i \boldsymbol{O}_i$ are the observed size and shape data we can write the integral part as

$$\mathcal{I}_1 = \int\limits_{\boldsymbol{R} \in SO(3)} \boldsymbol{R} \exp\left\{\frac{\mathrm{tr}\left(\boldsymbol{R}\boldsymbol{X}^M \boldsymbol{\mu}^t\right)}{\sigma^2}\right\} d\boldsymbol{R}$$

Then, by taking $\boldsymbol{U}_1 \boldsymbol{\Phi} \boldsymbol{U}_2^t$ as the singular value decomposition of $\frac{\boldsymbol{X}^M \boldsymbol{\mu}^t}{\sigma^2}$ with $\boldsymbol{U}_1, \boldsymbol{U}_2^t \in SO(3)$ and $\boldsymbol{\Phi} = \mathrm{diag}(\phi_1, \phi_2, \phi_3)$, we can write $\mathcal{I}_1$ as

$$\mathcal{I}_1 = \boldsymbol{U}_2 \int\limits_{\boldsymbol{R} \in SO(3)} \exp\left\{\operatorname{tr}\left(\boldsymbol{R}\Phi\right) d\boldsymbol{R}\right\} \boldsymbol{U}_1^t$$

Now, by setting $\mathcal{I}_2 = \int\limits_{\boldsymbol{R} \in SO(3)} \exp\left\{\operatorname{tr}\left(\boldsymbol{R}\Phi\right)\right\} d\boldsymbol{R}$, Dryden et al. (2015) showed that the evaluation of $\mathcal{I}_1$ can be reduced to a 3-dimensional gradient problem as

$$\mathcal{I}_1 = \operatorname{diag}\left(\nabla_{\boldsymbol{\Phi}_{jj}} \log \mathcal{I}_2\right) \tag{3.3.2}$$

As we previously discussed in Section 2.5 $\mathcal{I}_1$ is related to the normalizing constant of the Bingham distribution. The Bingham distribution in q dimensions with respect to a uniform measure dS_q is defined as

$$f(x; \boldsymbol{A}, q) = \frac{e^{x^t \boldsymbol{A} x}}{B_q(\boldsymbol{A})} dS_q \tag{3.3.3}$$

with $x^t x = 1$, $\boldsymbol{A}$ is a symmetric matrix and $B_q(\boldsymbol{A}) = \int_{x \in S_q} e^{x^t \boldsymbol{A} x} dS_q$ is the normalizing constant.

Without loss of generality we may assume that the matrix $\boldsymbol{A}$ is diagonal and parametrized as $\boldsymbol{A} = \operatorname{diag}(\xi_1, \xi_2, \xi_3, \xi_4)$. Expressing the rotation matrix by using quaternions (Wood, 1993), we can write $\mathcal{I}_2$ as

$$\mathcal{I}_2 = \frac{B_4(\boldsymbol{A})}{2} \tag{3.3.4}$$

where $\xi_4 = \phi_1 + \phi_2 + \phi_3$ and $\xi_i = 2\phi_i - \xi_4$, for $i = 1, 2, 3$ and ϕ_i the diagonal values of the matrix $\boldsymbol{\Phi}$. As a result, the gradient of $\mathcal{I}_1$ can be expressed as

$$\mathcal{I}_1 = \operatorname{diag}\left(\nabla_{\boldsymbol{\Phi}_{jj}} \log \frac{B_4(\boldsymbol{A})}{2}\right) \tag{3.3.5}$$

Finally, as shown in Kume and Wood (2005) the partial derivatives of $B_4(M)$ relate to those of higher order, hence the required gradient $\mathcal{I}_1$ can be expressed as

$$\mathcal{I}_1 = \operatorname{diag}\left(\nabla_{\boldsymbol{\Phi}_{jj}} \log \mathcal{I}_2\right) = I_3 - \begin{bmatrix} \frac{B_6(\boldsymbol{A}_2)+B_6(\boldsymbol{A}_3)}{\pi B_4(\boldsymbol{A})} & 0 & 0 \\ 0 & \frac{B_6(\boldsymbol{A}_1)+B_6(\boldsymbol{A}_3)}{\pi B_4(\boldsymbol{A})} & 0 \\ 0 & 0 & \frac{B_6(\boldsymbol{A}_1)+B_6(\boldsymbol{A}_2)}{\pi B_4(\boldsymbol{A})} \end{bmatrix} \tag{3.3.6}$$

with I_3 a 3-dimensional identity matrix.

Finally, we can use the HGM described in Section 2.6 to solve the required gradients of $\mathcal{I}_1$, where now $f(\boldsymbol{a}) = \mathcal{I}_2$ is the function we are interested in and the column vector $g(\boldsymbol{a})$ is defined as

$$g(\boldsymbol{a}) = \left[\mathcal{I}_2, \frac{\partial \mathcal{I}_2}{\partial \phi_1}, \frac{\partial \mathcal{I}_2}{\partial \phi_2}, \frac{\partial \mathcal{I}_2}{\partial \phi_3}\right]^t$$

3.4 Alignment algorithm for optimizing $\boldsymbol{M}$

In the previous Sections we have presented how we can estimate the nuisance parameters $\boldsymbol{\mu}$ and σ^2 for an alignment given by the match matrix $\boldsymbol{M}$. In this section we present an algorithm that updates the alignment of the matching matrix and explores all possible pairwise matches between the residues of two protein molecules to find the likelihood mode that corresponds to the optimal matching between them.

3.4.1 Algorithm for pairwise matching

We propose an algorithm for pairwise matching which is based on the size and shape likelihood defined in (3.2.7), using the EM algorithm described in section 3.3 and the Hungarian method from in Section 2.7. An issue with alignment methods is multi modality, as mentioned in Dryden et al. (2007), such algorithms tend to get stuck in local modes. More recent papers, (Kenobi and Dryden, 2012; Schmidler, 2007) use different sampling schemes, but still the problem is not completely solved. Here, we suggest a search algorithm which examines all possible pairs of atoms in order to find the best one among them. The criterion for comparing these pairs is based on the optimal value of the size and shape likelihood from (3.2.7). The algorithm by design will only add or remove pairs of atoms which increase the total value of the likelihood. Hence, the final mode will very likely be the best one from a given starting point. The procedure consists of two main steps *adding* and *removing* and one optional step of *jumping.*

- *Adding* (Steps: 3-9)

  Starting from a set of matched atoms by a given $\boldsymbol{M}$ we try to add as many as possible new pairs in $\boldsymbol{M}$. In order to do this, first we estimate the likelihood value of (3.2.7) for all pairwise combinations between the unmatched atoms, when each pair is considered as a new match. Hence, each likelihood represents the *cost* for this pair of atoms to be added in $\boldsymbol{M}$. Next, using the new likelihood values we create a *likelihood-cost* matrix for the unmatched atoms and apply the Hungarian method to obtain an initial assignment between them. Finally, we order each pair of the initial assignment by their likelihood value and we examine if by adding them one at a time in $\boldsymbol{M}$ the value of (3.2.7) is increased. We repeat the last part until we have added all the pairs or reach a likelihood mode.

- *Removing* (Steps: 10-13)

  The next step of removing pairs from $\boldsymbol{M}$ is included to overcome the effect of the starting point selection and consists of removing as many already matched atoms as possible (provided that the likelihood is increasing). Given the alignment of $\boldsymbol{M}$ from the previous step, we estimate the likelihood of (3.2.7) when each pair of matched atoms from $\boldsymbol{M}$ is removed keeping the rest fixed. Then if the maximum likelihood from those calculated is higher than the likelihood of the given $\boldsymbol{M}$ we remove the pair that corresponds to this likelihood value from $\boldsymbol{M}$. We repeat this step of removing until we have no more atoms to remove or we end in a likelihood mode.

We alternate between the *adding* and *removing* steps until we reach a likelihood mode of (3.2.7).

- *Jumping* (Steps: 15-17)

  Finally, this is an optional step of jumps and is included as an attempt to explore as much as possible the likelihood modes of (3.2.7). For a number of *jumps* defined by the user, we uniformly select a pair from the remaining unmatched pairwise combinations of atoms as a new match and restart the algorithm from the *adding* step.

The steps of the alignment algorithm can be summarized as below

Algorithm 1 Structural Alignment algorithm

```
1: Input : $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{M}, J$
2: Estimate the starting likelihood value $f_{S_0}(\boldsymbol{X}_1, \boldsymbol{X}_2 | \boldsymbol{M}, \boldsymbol{\mu}_0, \sigma_0^2)$
3: for $(i, j) \in \boldsymbol{M}_{ij} = 0$ do
4:     Consider pair $(i, j)$ as a new match and estimate $f_{S_{ij}}(\boldsymbol{X}_1, \boldsymbol{X}_2 | \boldsymbol{M}, \boldsymbol{\mu}_{ij}, \sigma_{ij}^2)$
5: end for
6: Create the likelihood-cost matrix $C$
7: Use the Hungarian method on matrix $C$ to obtain an initial alignment.
8: Order the pairs of atoms from Step 6 based on their likelihood values.
9: Add sequentially each pair from Step 7 until a likelihood mode (3.2.7) is reached.
10: for $(i, j) \in \boldsymbol{M}_{ij} = 1$ do
11:     Remove pair $(i, j)$ and estimate $f_{S_{ij}}(\boldsymbol{X}_1, \boldsymbol{X}_2 | \boldsymbol{M}, \boldsymbol{\mu}_{ij}, \sigma_{ij}^2)$ keeping the rest pairs fixed.
12: end for
13: If the value of $\max f_{S_{ij}}(\boldsymbol{X}_1, \boldsymbol{X}_2 | \boldsymbol{M}, \boldsymbol{\mu}_{ij}, \sigma_{ij}^2)$ is higher than the current likelihood value, remove pair $(i, j)$ from $\boldsymbol{M}$ and go to Step 9.
14: Repeat Steps 3-12 until we reach a mode of (3.2.7)
15: for 1 : $J$ do
16:     From the remaining unmatched pairs, uniformly select a pair $(i, j)$ as a new match and go to Step 3.
17: end for
18: Return : $\boldsymbol{M}$
```

3.5 Sequence - structure alignment

In this Section we extend the size and shape likelihood of (3.2.7) by including the sequence information of the molecules. With this addition we are able to simultaneously make inference using both the geometrical information provided by the structure of the molecule and the sequence information provided by the amino acid chain. In the papers of Rodriguez and Schmidler (2014) and Fallaize et al. (2014) similar likelihoods are used under a Bayesian framework.

Amino acid chains are represented by a one dimensional sequence of letters. Consider two amino acid sequences J^{X_1} and J^{X_2} with lengths k and l respectively. The elements for each sequence are letters from a set $\mathcal{J}$ which represents the 20 different amino acids. The objective of sequence alignment is to match each amino acid with another that is usually either of the same type or from the same family based on a score matrix.

Sometimes it is essential to create gaps in one or both sequences so that the overall alignment score is maximized. The concept of gaps and how to penalize over them is discussed later in Section 3.6. Sequence alignment was the first attempt of aligning protein molecules for establishing if they share common properties (Bishop and Thompson, 1986; Gerstein and Levitt, 1998).

3.5.1 PAM matrices

Protein sequences are evolved into time where *deletion*, *addition* or *mutations* of amino acid are happening. These evolutionary changes can be described using scores from substitution matrices. The two most commonly used matrices are: PAM (Dayhoff and Schwartz, 1978) and BLOSSUM (Henikoff and Henikoff, 1992). Although both matrices are for the same purpose, they have differences in their properties and in the way they have been created. As a result, they might produce different alignments depending on the evolutionary distance of the two proteins. In this Section we use only the PAM matrices although the implementation is exactly the same when the BLOSSUM matrices are used.

PAM matrices are a selection of 20 by 20 symmetrical matrices in which each entry is a score between a pair of amino acids which have been created after examining all mutations that happened over time in a large dataset of closely related protein sequences. A PAM matrix is characterised by the evolutionary distance d, usually with $d = 1, \ldots, 250$, where large numbers indicating that the corresponding sequences are distant evolutionary relatives. PAM matrices are created using Markov chain theory, estimating the probability of a mutation happening first at time 1 and then sequentially creating the probabilities up to time d. For instance a PAM-1 matrix has the scores for each pair of amino acid when 1 mutation over 100 amino acids had happened, PAM-50 the scores from 50 mutations over 100 amino acids and so on.

The entries of a PAM-d matrix can be written in the form of

$$\Psi_d(a, b) = 10 \log_{10} \left(\frac{q_d(ab)}{f_a f_b} \right) \tag{3.5.1}$$

where, a and b represent the two amino acids, $q_d(ab)$ the probability of observing a pair between amino acids a and b at evolutionary time d and f_a, f_b the marginal probabilities of amino acids a and b appearing in a protein sequence over all evolutionary times. A detailed explanation of how these probabilities are derived can be see in Dayhoff and Schwartz (1978). The diagonal of a PAM matrix has the highest positive values, meaning that the best possible pair for a given amino acid is with another one of the same type. Small positive scores (+1, +2) usually are between amino acids that belong to the same group (see Chapter 1) and the negative scores are between incompatible amino acids. The total score of an alignment is the sum of all the scores between the pairs of amino acids with high positive scores indicate a good alignment. The PAM250 matrix we will use in Chapter 4 can be seen in Table 3.1.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Table 3.1: PAM 250 matrix

As an example, Table 3.2 displays a part of the aligned sequences from the pair 101m-1mba, along with each individual score for each amino acid match by using the PAM250 matrix. The, the final score for this alignment without penalising for each gap will be the sum of all the individual scores as : $1+2+1+1-1+1+2-2=5$.

	+1	+2	+1	+1	-1	-	-	+1	-	+2	-2
J^{x_1}	I	L	K	K	K	-	-	G	-	H	H
J^{x_2}	F	V	N	N	A	A	N	A	G	K	M

Table 3.2: Part of the aligned sequences for the pair 101m-1mba, with the corresponding PAM250 score for each match.

3.5.2 Sequence -structure likelihood

The sequence likelihood which we are using here is the same as the one used by Rodriguez and Schmidler (2014) and Fallaize et al. (2014). In particular, given two amino acid sequences J^{X_1} and J^{X_2} the sequence likelihood for a given alignment $\boldsymbol{M}$ and a given evolutionary distance d is defined by

$$P(J^{X_1}, J^{X_2}|\boldsymbol{M}, \Psi_d) = \prod_{(i,j)\in \boldsymbol{M}} q_d(J_i^{X_1}, J_j^{X_2}) \prod_{i\notin M} f_{J_i^{X_1}} \prod_{j\notin M} f_{J_j^{X_2}} \tag{3.5.2}$$

where $q_d(J_i^{X_1}, J_j^{X_2})$ represents the probability that amino acid i from sequence J^{X_1} is matched with the amino acid j from the sequence J^{X_2} and $f_{J_i^{X_1}}, f_{J_j^{X_2}}$ represent the marginal probabilities for the unmatched amino acids in each sequence. The equation (3.5.2) is a standard way of expressing a sequence likelihood (Bishop and Thompson, 1986).

Next, by assuming that the sequence and the structure likelihood from (3.2.7) are independent, a joint structure-sequencee likelihood for a given alignment $\boldsymbol{M}$ and given parameters $\boldsymbol{\mu}, \sigma^2$ and d will be given by

$$\begin{aligned} f_{SS}(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2, d) &= f_S(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2) \times P(J^{X_1}, J^{X_2}|\boldsymbol{M}, \boldsymbol{\Psi}_d) \\ &= \left(\frac{1}{V}\right)^{k+l-2p} \left(\frac{1}{2\pi\sigma^2}\right)^{3p} \exp\left\{ -\frac{\sum\limits_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}||^2}{2\sigma^2} \right\} \\ &\times \prod_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} e^{\frac{\operatorname{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}^t\right)}{\sigma^2}} d\boldsymbol{R}_i \prod_{(i,j)\in \boldsymbol{M}} q_d(J_i^{X_1}, J_j^{X_2}) \prod_{i\notin \boldsymbol{M}} f_{J_i^{X_1}} \prod_{j\notin \boldsymbol{M}} f_{J_j^{X_2}} \end{aligned} \tag{3.5.3}$$

The value of the evolutionary distance d is kept fixed and we usually use either $d = 120$ or $d = 250$ in our examples, however it can also be treated as an unknown parameter and its estimation is possible as described in Section 4.5. The EM steps for the estimation of $\boldsymbol{\mu}$ and σ^2 remain the same as we described in Section 3.3 since the missing rotation parameters are independent of the sequence information and the sequence likelihood (3.5.2) can be treated as a constant value throughout the Expectation and Maximization steps. The **Algorithm 1** also remains the same with the only difference that the structure-sequence likelihood of (3.5.3) is used.

3.6 Gap penalty

In this Section we extend the likelihood function of (3.5.3) by including a gap penalty. So far, we have not considered conditioning on an order for the amino acid sequence. So the alignments generated only rely on the geometrical information of the molecules. By adding a gap penalty function to our method we appropriately penalise over *gaps* so that more meaningful solutions are generated.

To explain the meaning of a *gap* we go back to the example of Table 3.2. A *gap* is created when an amino acid from one sequence is not matched to an amino acid of the other. For example, the sixth amino acid A of the sequence J^{X_2} which is assigned to a '-' indicating a gap in the first sequence J^{X_1}. This is considered as a *gap - opening.* On the other hand, as a *gap length* is defined the number of unmatched amino acids in the sequence until another one is matched.

The affine gap penalty function which we are also using here is the most common penalty function in Bioinformatics literature and has also been used byAalberse (2000) Rodriguez and Schmidler (2014) and Fallaize et al. (2014), as a prior over the match matrix $\boldsymbol{M}$. It is defined as

$$U(g,h) = -gS(\boldsymbol{M}) - h\sum_{i=1}^{S(\boldsymbol{M})} l_i(\boldsymbol{M}) \qquad (3.6.1)$$

where, $\boldsymbol{M}$ is the match matrix, g and h the parameters of gap opening and gap extension, $S(\boldsymbol{M})$ is the total number of gap openings for each sequence and $l_i(\boldsymbol{M})$ is the length of each gap from each sequence. Going back to the example of Table 3.2 the total number of gap openings $S(\boldsymbol{M})$ will be 2 and the gap lengths $l_i(\boldsymbol{M})$ for each opening will be 2 and 1. We need to note that we want each gap opening to *carry* a strong penalty in the total likelihood value, hence we choose to use a rate of $\exp(U)$ (as similarly done in the Bayesian methods) as the final gap penalty instead of simply using the affine function of (3.6.1).

Finally, we consider the gap opening and extension parameters g and h as fixed and in particular as suggested by Gerstein and Levitt (1998) we choose g

to be about 40 times larger than h. In other studies (Rodriguez and Schmidler, 2014; Fallaize et al., 2014) the gap parameters are treated both as fixed and as unknown values, where Gamma priors are assigned over them.

3.6.1 Sequence - structure likelihood with gap penalty

We can easily include the gap penalty function in the structure-sequence likelihood from (3.5.3). Now, for a given alignment $\boldsymbol{M}$ and a PAM matrix d with fixed gap penalty parameters g and h the sequence likelihood function (3.5.2) becomes

$$P(J^{X_1}, J^{X_2}|\boldsymbol{M}, \boldsymbol{\Psi}_d, g, h) = \exp\{U(g,h)\} \prod_{(i,j)\in \boldsymbol{M}} q_d(J_i^{X_1}, J_j^{X_2}) \prod_{i\notin M} f_{J_i^{X_1}} \prod_{j\notin M} f_{J_j^{X_2}} \tag{3.6.2}$$

and the structure-sequence with gap penalty likelihood is

$$\begin{aligned} f_{SSG}(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2, d, g, h) &= f_S(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2) \times P(J^{X_1}, J^{X_2}|\boldsymbol{M}, \boldsymbol{\Psi}_d, g, h) \\ &= \left(\frac{1}{V}\right)^{k+l-2p} \left(\frac{1}{2\pi\sigma^2}\right)^{3p} \exp\left\{ -\frac{\sum_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}||^2}{2\sigma^2} \right\} \\ &\times \prod_{i=1}^{2} \int_{\boldsymbol{R}_i \in SO(3)} e^{\frac{\mathrm{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}^t\right)}{\sigma^2}} d\boldsymbol{R}_i \prod_{(i,j)\in \boldsymbol{M}} q_d(J_i^{X_1}, J_j^{X_2}) \prod_{i\notin \boldsymbol{M}} f_{J_i^{X_1}} \prod_{j\notin \boldsymbol{M}} f_{J_j^{X_2}} \\ &\times \exp\left\{ -gS(\boldsymbol{M}) - h \sum_{i=1}^{S(\boldsymbol{M})} l_i(\boldsymbol{M}) \right\} \qquad (3.6.3) \end{aligned}$$

Notice also that a version of (3.6.3) with structure and only a gap penalty function can be easily obtained as

$$f_{SG}(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2, g, h) = f_S(\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2) \times \exp\{U(g,h)\} \tag{3.6.4}$$

Similar to before, the estimation of $\boldsymbol{\mu}$ and σ^2 is done using the EM algorithm described in Section 3.3, where the EM steps remain the same since the gap information is considered fixed during the optimization procedure. The **Algorithm 1** also remains the same with substituting the relevant terms in the likelihood of

(3.6.3) and (3.6.4).

3.6.2 Alignment algorithm for preserving sequence order

The **Algorithm 1** is based only on the geometrical information of the protein molecules. Using the likelihood densities of (3.6.3) or (3.6.4) we penalize for gaps in each sequence. However, in order to preserve the sequence order of the alignment a different approach for the optimization of the match matrix $\boldsymbol{M}$ is required.

Hence, we modify **Algorithm 1** by exploring only pairwise combinations of atoms that follow the sequence order from a given starting point and not all possible pairwise combinations available. The process of selecting the appropriate pairs that follow the sequence order is described below:

Consider two protein molecules $\boldsymbol{X}_{1_i}$ with atoms $i = 1, \ldots, k$ and $\boldsymbol{X}_{2_j}$ with atoms $j = 1, \ldots, l$ and a set of starting points p represented by the indices P^1_q, P^2_q with $q = 1, \ldots, p$ for each $\boldsymbol{X}_1, \boldsymbol{X}_2$. Then, we order the matched atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as :

$$P^1_1 < P^1_2 < \cdots < P^1_p \quad \text{and} \quad P^2_1 < P^2_2 < \cdots < P^2_p$$

Now in order to find the next possible match for $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ we explore only those pairs created by the combinations of atoms using the following rules:

List 3.1: Create pairs of atoms that preserve sequence order.

1. Find all possible combinations of pairs from $\boldsymbol{X}_{1_i}$ and $\boldsymbol{X}_{2_j}$ created by the unmatched atoms of those two in which $P^1_i < P^1_1$ and $P^2_j < P^2_1$.
2. Find all possible combinations of pairs created by the atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ in which $P^1_1 < P^1_i < P^1_2$ and $P^2_1 < P^2_j < P^2_2$.
3. Repeat Step 2 (p-1) times until $P^1_{p-1} < P^1_i < P^1_p$ and $P^2_{p-1} < P^2_j < P^2_p$.
4. Find all possible combinations of pairs created by the atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ in which $P^1_i > P^1_p$ and $P^2_j > P^2_p$.

Using this approach sometimes requires a good set of starting points, preferably a pair of atoms that are spread throughout the protein chain and not concentrated around a small area.

Finally, a simple adjustment should be made to the Hungarian algorithm described in Section 2.7, since not all combinations of atoms are now considered as possible new matches. As a result, there will be some empty entries on the *likelihood-cost* matrix that we should deal with. Hence, in order to modify the *likelihood-cost* matrix so it can be suitable for the Hungarian method we use these three following steps :

List 3.2: Adjustments for the likelihood-cost matrix.

1. Add extra zero rows or columns in order for the *likelihood-cost* matrix to become square.
2. Fill the empty entries of the *likelihood-cost* matrix that correspond to the atoms which are not selected by List 3.1 with a very small negative value.
3. Subtract this value from all the other entries.

Using these three steps we will now have a square *likelihood-cost* matrix with non-negative entries and by applying the Hungarian method we will have an initial alignment between the atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

Finally, the Alignment with Sequence Order algorithm will be the same as **Algorithm 1** where now the Steps 3-6 become:

Algorithm 2 Alignment with Sequence Order

```
1: Use the instructions of List 3.1 to create the available combination of pairs.
2: for pair $(i, j)$ from Step 1 do
3:     Consider the pair $(i, j)$ as a new match and estimate (3.6.3).
4: end for
5: Use the instructions of List 3.2 to create the likelihood-cost matrix $C$.
```

The **Algorithm 2** can be applied any of the likelihood densities described in (3.6.3) or (3.6.4).

3.7 Selection of starting points

So far for initializing the **Algorithm 1** we have assumed that a given set of starting points was available either by using a part or the whole alignment solution from other methods or by selecting them visually. However, this is may not always be possible and a different way of selecting the starting points is needed. In this Section, we describe an algorithm to automatically select a set of starting points using both the sequence and the geometrical information of our data.

3.7.1 Algorithm for automatic selection of starting points

Since our alignment algorithm is based on an EM approach and as previously discussed in Chapter 2 the starting point selection might have an impact on the final solution. We describe a method for selecting a number of atoms so they can be used as starting points for **Algorithm 1** or **Algorithm 2**.

Using the same set up as before where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ represent two protein molecules, the first step of the algorithm would be to perform an initial sequence alignment. Many different sequence alignment algorithms exist such as the Needleman-Wunsch (Needleman and Wunsch, 1970) for global alignment, the Smith-Waterman (Smith and Waterman, 1981) for local alignment or the MUSCLE method (Edgar, 2004) for aligning multiple sequences. For our purpose, all these algorithms have similar performance and we choose to use the Needleman-Wunsch since it is very simple to use and easily accessible in R.

After obtaining an initial alignment, the second step will be to optimally rotate and translate the aligned data from before. Next, for each aligned pair the TMscore of (2.8.2) is calculated in order to assess the quality of each matched pair. As a starting cut-off point we choose the value of 20%, since as shown in Xu and Zhang (2010) a TMscore below that value indicates that the two protein molecules are unrelated.

We continue by removing the matched pairs which are below the selected cut-off point and this process is repeated until there are no more available pairs to remove. However, in the case that we compare two proteins with low sequence similarity the initial sequence alignment might not provide a good start. As a result, the remaining matched pairs might be less than 4. If that is the case, we restart the algorithm but with setting a lower new cut-off point (e.g. half of its previous value). The steps for the selection process of the starting points can be summarized as follow :

Algorithm 3 Algorithm for selecting k - starting points

```
1: **Inputs** $\boldsymbol{X}_1, \boldsymbol{X}_2, c_0, k$
2: Use Needleman-Wunsch algorithm to obtain an initial sequence alignment between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.
3: Set $p$ the number of matched pairs from Step 2.
4: **while** $p \geq 4$ **do**
5:     Translate and rotate optimally the aligned data from Step 2.
6:     Compute TMscore for each aligned pair.
7:     Remove the pairs that are below the TMscore cut-off point $c_0$.
8:     **if** $p < 4$ **then**
9:         Set $c_0 \leftarrow c_0/2$ and go to Step 3.
10:    **end if**
11: **end while**
12: **return** $k$ - pairs with the highest TMscore.
```

Algorithm 3 can be used for initializing any of the algorithms presented in Sections 3.4.2 or 3.6.2.

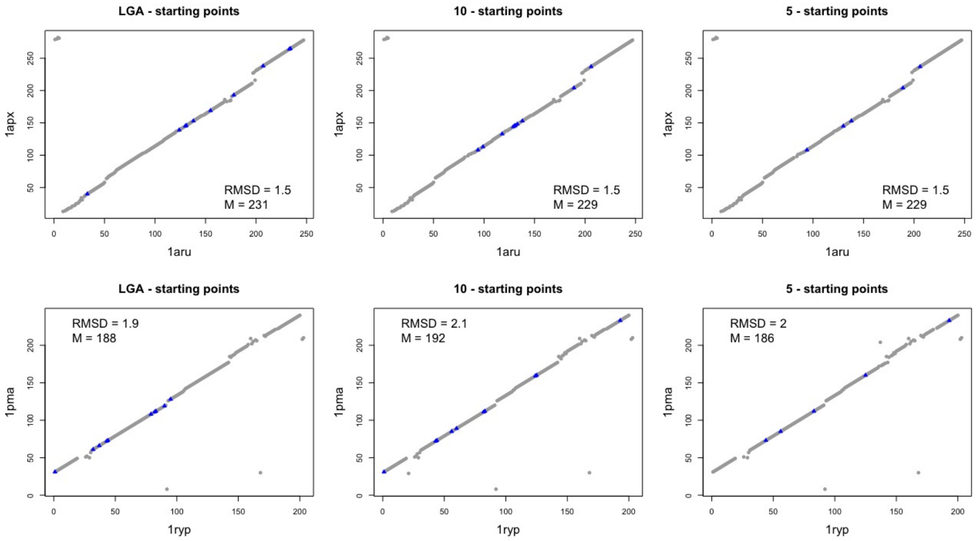


Figure 3.1: Starting point comparison for the pairs of 1aru- 1apx (top row) and 1ryp - 1pma (bottom row). The starting points for each alignment are with blue.

Figure 3.1 displays the effect of different starting points for two protein pairs. The first one is a pair of *peroxidases* 1apx - 1aru with a low sequence similarity of 27% and the second, is a pair of the *antibody* 1rup with the *protease enzyme* 1pma with a sequence similarity of 42%. We choose 3 different sets of starting points, the first is a subset of the aligned atoms from the solution of the LGA method and the other two are a set of 5 atoms and a set of 10 atoms as being selected by **Algorithm 3**. We also used the likelihood with the structure information of (3.2.7).

For the first case, we can see that all three different set of starting points give almost the same solutions having all the same RMSD 1.5Å with the set of starting points from LGA resulting to 231 matched atoms compared to 229 atoms using the starting points from **Algorithm 3**. In the second case, each set of starting points results to slightly different solutions. The set obtained from LGA gives an alignment of 188 matched atoms with RMSD of 1.9Å, compared to 192 matched atoms and RMSD of 2.1Å or 186 and RMSD 2.0Å when the **Algorithm 3** is used.

From this simple example we can see that **Algorithm 3** can provide a reasonable good start for **Algorithm 1**. In both cases, our final alignments were very similar despite the different choice of starting points. However, this might not always be the case, especially if we compare pairs of proteins with low sequence or structure similarity.

3.8 Multiple alignment

In this Section we extend the likelihood model described in Sections 3.2 and 3.3 for the case of matching more than one protein molecules simultaneously. This problem has also been studied in the Bayesian literature before by Dryden et al. (2007) . In that paper, one of the molecules is treated as the target and all the others are matched to it, then the Procrustes registration is used for the estimation of rotation and translation parameters. On the other hand, Ruffieux and Green (2009) extended the pairwise matching model described in Green and Mardia (2006) to the multiple case by allowing also partial matches between the molecules. Finally, another approach in multiple matching is by Mardia et al. (2011) where they use a Bayesian hierarchical template algorithm for aligning parts of protein molecules.

3.8.1 Size and shape density for multiple matching

We consider a set of protein molecules represented by the configuration matrices $\tilde{\boldsymbol{X}} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n]$ where n is the number of molecules and each individual $\boldsymbol{X}_i, i = 1, \ldots, n$ is a 3-dimensional matrix with k_i number of atoms. Similarly, we consider a set of matching matrices $\tilde{\boldsymbol{M}} = [\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_n]$ where each $\boldsymbol{M}_i, i = 1, \ldots, n$ is a matrix of dimensions $k_i \times \min(k)$ with ones and zeros representing the correspondence of the atoms between each $\boldsymbol{X}_i$ and the common mean $\boldsymbol{\mu}$.

In addition, assuming the parameters of rotations and translations are independent among each molecule $\boldsymbol{X}_i$ then, the set of size and shape transformations is defined as

$$\mathcal{R}_i = \left\{ \boldsymbol{R}_i \boldsymbol{X}_i + \tau 1_p^t : \boldsymbol{R}_i \in SO(3), \tau \in \mathbb{R}^3 \right\}$$

with $\boldsymbol{R}_i$ a 3-dimensional rotation matrix and τ_i a 3-dimensional translation vector corresponding to each molecule $\boldsymbol{X}_i$.

Following the general likelihood framework described in Chapter 2 and by using the Singular Value Decomposition of matrix $\boldsymbol{X}$ as in (3.2.3) and partitioning each $\boldsymbol{X}_i$ in matched and unmatched parts, which follow the distributional assumptions of (2.2.3) and (2.2.4), it is straightforward to extend the likelihood density described from (3.2.7) to the multiple molecule matching case as

$$\begin{aligned} f_M(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}, \sigma^2) &= \int\limits_{\boldsymbol{R}_i \in SO(3)} \mathcal{L}(\tilde{\boldsymbol{M}}, \boldsymbol{\mu}, \sigma^2|\boldsymbol{X}_i) d\boldsymbol{R}_i \\ &= \int\limits_{\boldsymbol{R}_i \in SO(3)} f_M(\boldsymbol{X}_i^M|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}, \sigma^2) d\boldsymbol{R}_i f_{-M}(\boldsymbol{X}_i^{-M}|\tilde{\boldsymbol{M}}, V) \\ &= V^{-\left(\sum\limits_{i=1}^{n} k_i + np\right)} (2\pi\sigma^2)^{-\frac{3pn}{2}} \exp\left\{ -\frac{\sum\limits_{i=1}^{n} ||\boldsymbol{X}_i^M||^2 + n||\boldsymbol{\mu}||^2}{2\sigma^2} \right\} \\ &\quad \times \prod_{i=1}^{n} \int\limits_{\boldsymbol{R}_i \in SO(3)} \exp\left\{ \frac{\operatorname{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}^t\right)}{\sigma^2} \right\} d\boldsymbol{R}_i \qquad (3.8.1) \end{aligned}$$

where n is the number of molecules, p the number of common matched atoms across all molecules and $\tilde{\boldsymbol{M}}, \boldsymbol{\mu}$, σ^2 are the parameters of interest that need to be optimized.

Therefore, the likelihood density of (3.8.1) can be easily extended to incorporate the extension of the sequence information from (3.5.3) or the Gap penalty function from (3.6.1). The optimization over the parameters of $\boldsymbol{\mu}$ and σ^2 remains the same as using the EM algorithm described in Section 3.3 since the expected rotations $\boldsymbol{R}_i$ are independent among each $\boldsymbol{X}_i$.

3.8.2 Likelihood - cost matrix for multiple matching

The structural alignment algorithm for pairwise matching described in **Algorithm 1** can be easily extended to the multiple matching case by replacing the likelihood density of (3.2.7) or (3.6.3) by the density described in (3.8.1).

The only adjustment that needs to be made is in the process of defining the *likelihood-cost* matrix that is used by the Hungarian algorithm for obtaining an initial assignment between the atoms. Since the Hungarian algorithm is designed for two objects as in the pairwise matching of two molecules, we need to extend the *likelihood-cost* matrix by including the likelihood information for all n available molecules.

Now $\mathbb{H}$ has dimensions $\min(k_i)-p\times\prod\limits_{j\in\{i\setminus\min(k_i)\}} k_i-p$ and each entry represents the *likelihood-cost* for each tuple of unmatched atoms to be added as a new match. The rows of $\mathbb{H}$ represent the actual unmatched atoms from the $\boldsymbol{X}_i$ with the minimum number of dimensions and the columns of $\mathbb{H}$ serve as an index to all the possible combinations over the unmatched atoms of the rest $\boldsymbol{X}_i$.

In order to illustrate the process of defining $\mathbb{H}$ we use a simple example. Consider three molecules $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$ with number of atoms $5, 5, 7$ respectively. Also, without loss of generality we assume that the first 3 atoms from each $\boldsymbol{X}_i$ are considered as already matched. Then, the remaining atoms form the following possible combinations as candidates for new matches

$$\left[\begin{array}{ccc} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \boldsymbol{X}_3 \\ \hline 4 & 4 & 4 \\ 5 & 4 & 4 \\ 4 & 5 & 4 \\ 5 & 5 & 4 \\ \vdots & \vdots & \vdots \\ 4 & 5 & 7 \\ 5 & 5 & 7 \end{array}\right]$$

Hence, the matrix $\mathbb{H}$ will be formed as follows

$$\mathbb{H} = \begin{array}{cc|cccccccc} & & \multicolumn{8}{c}{\boldsymbol{X}_2, \boldsymbol{X}_3} \\ & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline \mathbf{X}_1 & 4 & h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} \\ & 5 & h_{51} & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} & h_{57} & h_{58} \\ \hline \end{array}$$

where, each element of h_{ij} represents the *likelihood-cost* of this tuple to be added as a new match. For example h_{41} will be the likelihood for the tuple $(4, 4, 4)$, h_{51} the likelihood for the tuple $(5, 4, 4)$ and so on.

Finally, the algorithm for multiple structural alignment will be the same as the one described in **Algorithm 1** with the likelihood of (3.2.7) replaced by (3.8.1) and with the cost matrix C being replaced by $\mathbb{H}$.

3.8.3 Alternative structural alignment algorithm for multiple matching

One important problem with the method described in the previous Section is that when we want to align many molecules simultaneously then it becomes a very big combinatoric problem since we need to explore all possible combinations of atoms in order to find the best local likelihood mode of $\boldsymbol{M}$. This approach is generally feasible for small number of molecules 3 or sometimes 4 regardless of their number of atoms.

However, when our sample size becomes bigger, the number of combinations that we need to explore becomes significantly larger even for small molecules of 30-40 atoms each. In this case although the computing time increases due to the design of the search, using parallel computing makes the *Adding* step of **Algorithm 1** still manageable. Problems start to appear when the Hungarian algorithm needs to be applied. The Hungarian algorithm solves the assignment problem in polynomial time of order $O(n^3)$ (Munkres, 1957). This n represents the number of pairs we need to explore each time before we consider a new match in $\boldsymbol{M}$. Hence, it is easy to see how big the problem becomes especially when we have a number of molecules more than 5. Furthermore, a square matrix is needed as an input and by using the approach described in Section 3.8.2 for adding extra empty rows or columns to make it square will lead to a *likelihood-cost* matrix with larger dimensions requiring a lot of memory.

For all the above reasons, when we want to align simultaneously more than 3 or 4 molecules we need to adjust **Algorithm 1** by dropping the step for obtaining an initial alignment from the Hungarian algorithm and directly adding the new matches in $\boldsymbol{M}$ based only on the likelihood values. This adjustment reduces significantly the computing time and the memory usage and although it is an approximation to what the Hungarian algorithm does, is necessary to make our approach feasible in large scale comparisons. The steps of the Alternative Structural Alignment for multiple matching are summarized below:

Algorithm 4 Alternative structural alignment algorithm for multiple matching

```
1: **Input** : $\tilde{\boldsymbol{X}} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n], \tilde{\boldsymbol{M}}, J$
2: Estimate the starting likelihood value $f_{M_0}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_0, \sigma_0^2)$
3: **for** tuple $S_j \in \tilde{\boldsymbol{M}}_{S_j} = 0$ **do**
4:     Consider $S_j$ as a new match and estimate $f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2)$
5: **end for**
6: **if** $\max f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2) > f_{M_0}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_0, \sigma_0^2)$ **then**
7:     Add the tuple which corresponds to $\max f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2)$ as a new match in $\tilde{\boldsymbol{M}}$
8:     Set $f_{M_0}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_0, \sigma_0^2) \leftarrow \max f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2)$
9: **end if**
10: Repeat Steps 3-8 until a mode of $\tilde{\boldsymbol{M}}$ is reached.
11: **for** tuple $S_j \in \tilde{\boldsymbol{M}} = 1$ **do**
12:     Remove tuple $S_j$ from $\tilde{\boldsymbol{M}}$ and estimate $f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2)$ keeping the rest tuples fixed.
13: **end for**
14: **if** $\max f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2) > f_{M_0}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_0, \sigma_0^2)$ **then**
15:     Remove the tuple which corresponds to $\max f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2)$ from $\tilde{\boldsymbol{M}}$,
16:     Set $f_{M_0}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_0, \sigma_0^2) \leftarrow \max f_{M_j}(\boldsymbol{X}_i|\tilde{\boldsymbol{M}}, \boldsymbol{\mu}_j, \sigma_j^2)$
17:     Go to Step 11.
18: **end if**
19: Repeat Steps 3-18 until we reach a mode of $\tilde{\boldsymbol{M}}$
20: **for** $1 : J$ **do**
21:     From the remaining unmatched tuples, uniformly select a tuple $S_j$ as a new match in $\tilde{\boldsymbol{M}}$ and go to Step 3.
22: **end for**
23: **Return** : $\tilde{\boldsymbol{M}}$
```

Chapter 4

Comparisons and real data applications

4.1 Introduction

In this Chapter we test and compare the various approaches for structural alignments proposed in Chapter 3. In Section 4.2, we simulate data using different values for the parameter σ and measure the ability of our models to correctly identify the landmarks either as matched or unmatched. Furthermore, we explore the effect the volume parameter V has in our model and also provide a brief discussion regarding the computational time needed for different protein lengths. In Section 4.3 we use two different benchmark datasets of protein pairs to compare our approach with other known algorithms from Bioinformatics literature.

In Section 4.4 we compare our approach with that of Rodriguez and Schmidler (2014), using all the different likelihood densities introduced in Chapter 3. In Section 4.5 we explain how our approach can be also used to estimate the evolutionary distance between two proteins and present a small example which is also been analysed in previous studies. Finally, in Section 4.6 we test the method of simultaneously aligning protein molecules using the two different methods described in Section 3.8.

4.2 Simulations

In this Section we evaluate the performance of our Likelihood Alignment method from Chapter 3 using simulated data. We also explore the effect the volume parameter has in the final alignment with simulations and an example using protein data. Finally, we discuss the computational time needed for our method and the limitations that arise.

4.2.1 Simulated data

For generating the simulated data we use a similar algorithm as the one described in Kenobi and Dryden (2012) by choosing the following parameters :

- 1000 samples of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ with 25 and 30 landmarks respectively.
- The first 20 landmarks from each matrix represent the *correct* matches and are observations from a Normal distribution with a common mean $\boldsymbol{\mu}$ and variance σ^2.
- The locations for each landmark of $\boldsymbol{\mu}$ are drawn from a Uniform distribution with the restriction that each location should have at least a minimum distance (d_{min}) from the others.
- The *unmatched* landmarks of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are sampled from a Uniform distribution inside a cube of volume V^3.
- We set as $d_{min} = 2, V = 20$ and $\sigma = 0.1, 0.5, 1, 2, 2.5$

Figure 4.1 displays the empirical matching probabilities of each landmark. When $\sigma = 0.1$ or 0.5 we see that our method has a very high probability of success. In particular, there is above 95% chance that each landmark is correctly identified as matched (first 20 landmarks) or as unmatched (last 5 landmarks). As σ increases (together with the ratio of σ/d_{min}) these percentages seem to drop. When $\sigma = 1$ we can again very efficiently match the first 20 landmarks correctly, but there also seems to be a low number of false positives matches especially from the 5 unmatched landmarks. However, this is more noticeable when the ratio $\sigma/d_{min} \geq 1$, which is when $\sigma = 2$ or $\sigma = 2.5$. In these two cases the matched and unmatched landmarks are mixing and it becomes harder to identify which ones are from the Normal distribution and which ones from the Uniform. Now, the percentage for the 20 first landmarks is still good of about 77% for $\sigma = 2$ and 66% for $\sigma = 2.5$, but the proportion of falsely identifying an unmatched landmark as being matched has also increased with 54.5% and 65% respectively for the two σ's.

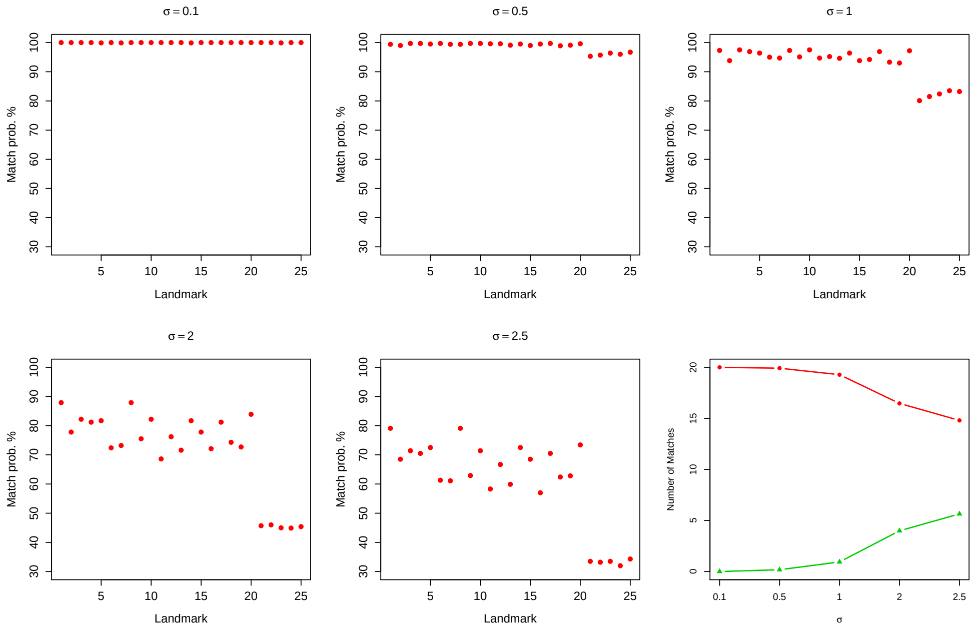


Figure 4.1: Mean proportions of each landmark to be identified successfully either as a 'correct' match or as 'unmatched' landmark. The bottom right plot is the number of correct and false positive matches for each of the first 20 landmarks for varying σ.

A similar simulation study has been conducted also in Kenobi and Dryden (2012) where they compare the models of Dryden et al. (2007) and Green and Mardia (2006). Their results were of similar performance with identifying also a potential cut-off point for the ratio of σ/d_{min} where the performance of these two methods is changing.

Finally, based on the bottom right plot of Figure 4.1 with the mean number of correct and false positive matches for each σ and also from the distribution of correct and false positive matches of Figure 4.2 we can conclude that even when the standard deviation is equal or higher than the minimum distance of 2, the number of correct matches remains high with an average of 16.5/20 for $\sigma = 2$ and 14.9/20 for $\sigma = 2.5$. Similarly, the average of false positive values is as low as 3.9 and 5.5 respectively, indicating that the algorithm performs relatively well in situations of large variance.

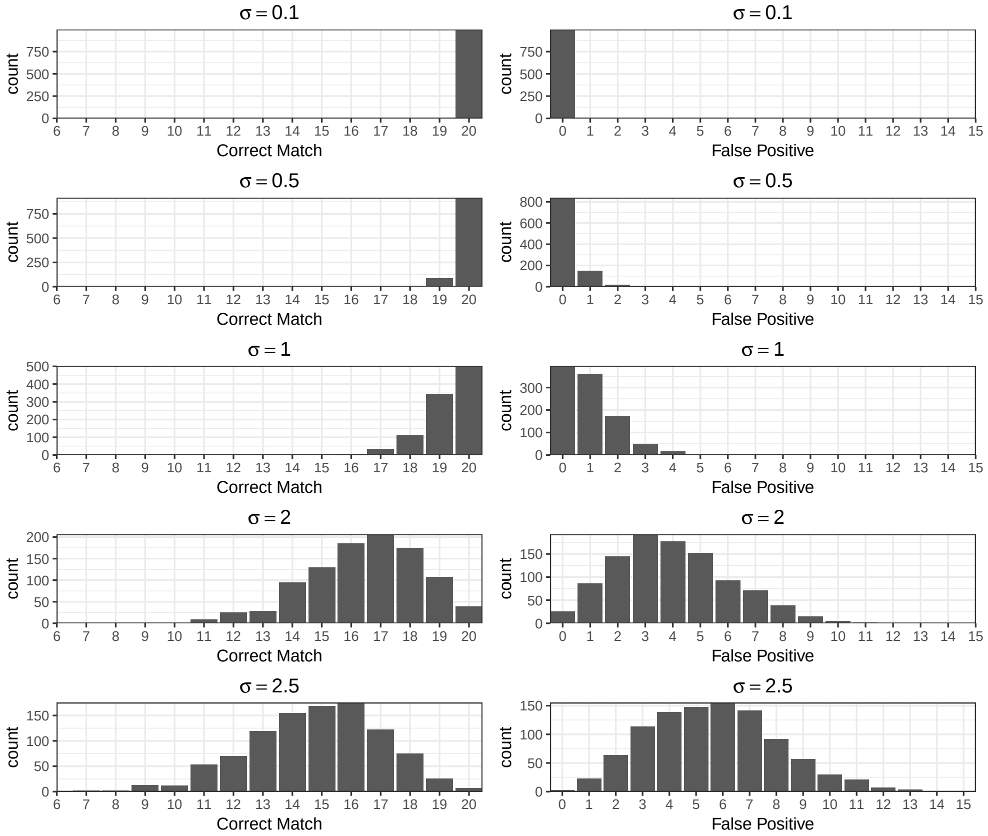


Figure 4.2: Distribution of correct matches and false positives for different values of σ.

4.2.2 Effect of the volume parameter V

So far, we have considered a fixed value for the volume parameter V, usually the one calculated from the data by multiplying the range from the protein coordinates. As mentioned in the papers of Rodriguez and Schmidler (2014) and Fallaize et al. (2014), its value can have an effect in the final solution. In general, larger values of volume will mean a bigger space with the two molecules inside, hence the distance between each landmark will be larger relative to the variance σ^2. The opposite is happening with small values of V where now the landmarks are more clustered and mixed, making harder to identify the *correct* matches.

Therefore, the value of the volume can also be used as a tuning parameter in order to obtain final solutions with more or less matches and bigger or smaller RMSD.

In order to explore the effect of the volume parameter in our model we test it using the simulated data from the previous Section. Figure 4.3 displays the number of correct and false positives matches for the first 20 landmarks for volumes of V ranging between 1000 and 10000. As we can see there is no variation in the number of either the correct matches or false positives for $\sigma = 0.1, 0.5$ or 1. For the other two values of $\sigma = 2$ and $\sigma = 2.5$ there seems to be a small amount of variation on both types of matches. However this is very small with approximately 3 more matched landmarks for both correct and false positives.

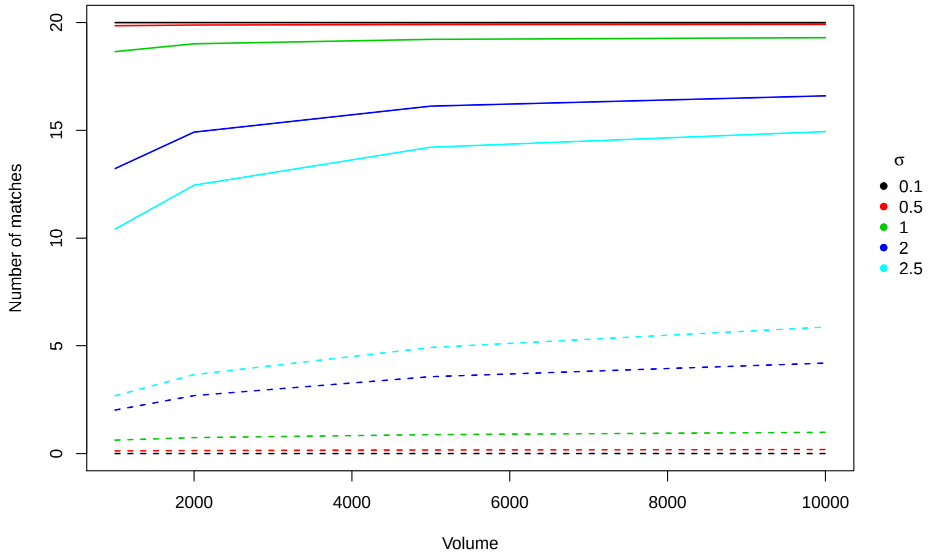
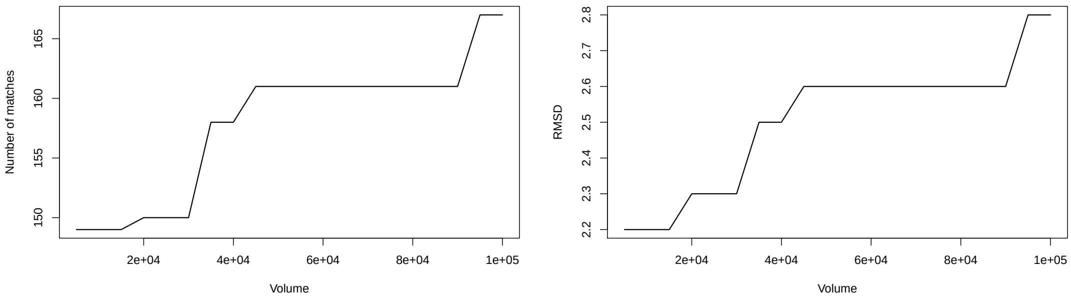


Figure 4.3: Mean number of correct matches (continuous lines) and false positive (dashed lines) for different values of σ and V.

In order to illustrate the volume effect in real data we use the protein pair of 1gky-2ak3. Figure 4.4 shows that the value of the volume can have some effect on the final solution, which can range from 148 matched atoms and a RMSD of 2.2Å for $V = 5000$ to 167 and 2.8Å for $V = 100000$. However, after a certain value of $V \approx 40000$, the volume parameter does not seem to have an effect both in RMSD and the number of matches, since the space that the two molecules are considered to be inside is large enough. Although there is some variation on the final results this is of a magnitude of about 10%, which suggests, as previously shown in the simulations, that the choice of V in our methods does not have a big impact on the final alignment. Fallaize et al. (2014) also discuss the volume effect in their model. For the same pair of proteins they report a number of matches ranging from 117 to 152 with a RMSD from 2.0Å to 2.97Å. By comparing these results, it seems that our model is less affected by the volume, although further exploration is needed since the alignment of protein molecules can have significant variations from pair to pair.



(a) Number of matches (b) RMSD

Figure 4.4: Number of matched atoms and RMSD values of the pair 1gky-2ak3 for values of volume ranging from 5000 to 100000.

4.2.3 Computational time

An important aspect of structural alignment algorithms is the computational time. In general most of these algorithms from the field of Bioinformatics perform the alignments in a matter of just a few seconds (see later Table 4.1). Green and Mardia (2006) report a time around 1 minute for the alignment of configurations with 40-50 landmarks. The time needed for the Likelihood Alignment approach we described in Chapter 3 is connected to the total number of atoms from the two proteins. Since we are exploring all the possible pairwise combinations of atoms the total computational time needed will increase geometrically. Figure 4.5 shows the computational time needed for our algorithm depending on the total number of atoms. Although, the growth is of geometric rate, we can see that even for 500 atoms the total computational time needed is approximately 20 seconds, a very good time compared to the other methods. In order to achieve this, we made use of parallel computing, because there is no need to explore the pairwise combinations sequentially and hence we can divide our problem into smaller parts.

However, this type of approach has some limitations. Although, for any number of atoms in pairwise alignment this approach is feasible, when we want to align multiple structures simultaneously, the total number of combinations becomes significantly large. We can still achieve good computational times when we have 3 molecules but in the case of aligning more than 3 simultaneously, we have a limit of about 30-40 atoms for each protein. This problem discussed also in Section 3.8.3 arises mainly from the construction of the *likelihood-cost* matrix that is needed for the Hungarian algorithm. By adding the extra rows or columns so that it becomes square, its dimensions become very big and difficult to handle, both computationally and in terms of memory management.

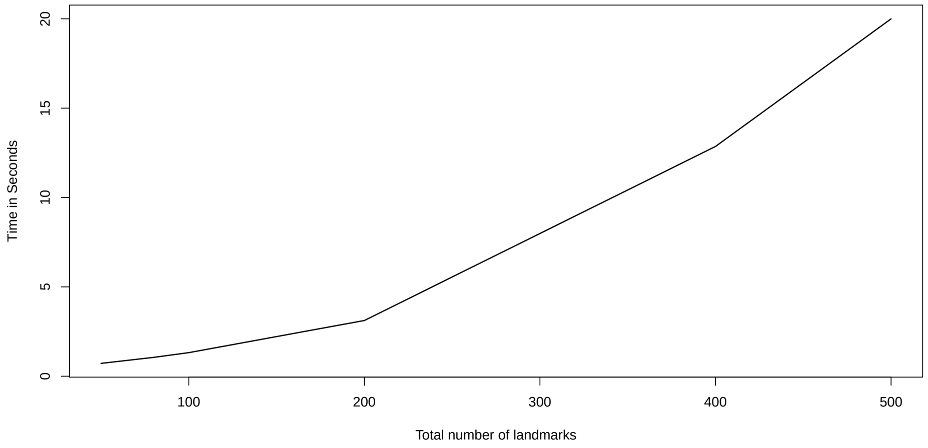


Figure 4.5: Computational time needed for the Likelihood Alignment method using ***Algorithm 1***

4.2.4 Conclusions

- Based on the simulation results of Section 4.2.1, the Likelihood Alignment approach seems to perform well in identifying which landmarks should be matched and which not, especially when σ is low. When $\sigma/d_{min} \geq 1$ it seems that the correct match percentage is still good but the false positive percentage is increased.

- The volume parameter V after a certain value does not have a big effect in the final alignment, although further exploration is needed when treated as a non-fixed parameter in the model.

4.3 Protein data

4.3.1 1stmA-1bmvI

We present an example of using the likelihood of (3.2.7) and **Algorithm 1** to align a pair of proteins. The first molecule is the *virus* 1stmA consisting of 141 atoms and the second is the *RNA virus* 1bmvI with 185 atoms. This pair has a sequence identity of 28% from the BLAST method. Using **Algorithm 1** we obtain an alignment of 85 matched atoms with an RMSD (2.8.1) of 2.0Å, a TMscore (2.8.2) of 0.51 and a Structure Overlap (2.8.3) of 57.45%.

Figure 4.6 displays the one-to-one correspondence of the final alignment. As we can see although we do not use a gap penalty in the likelihood function the amino acid sequence order is mostly preserved. The TMscore value is just above the threshold 0.50 suggesting that the two proteins might belong to the same fold. However, the Structure Overlap of 57.45% is not very high, meaning that only about half of the aligned atoms have a distance of less than 3.5Å

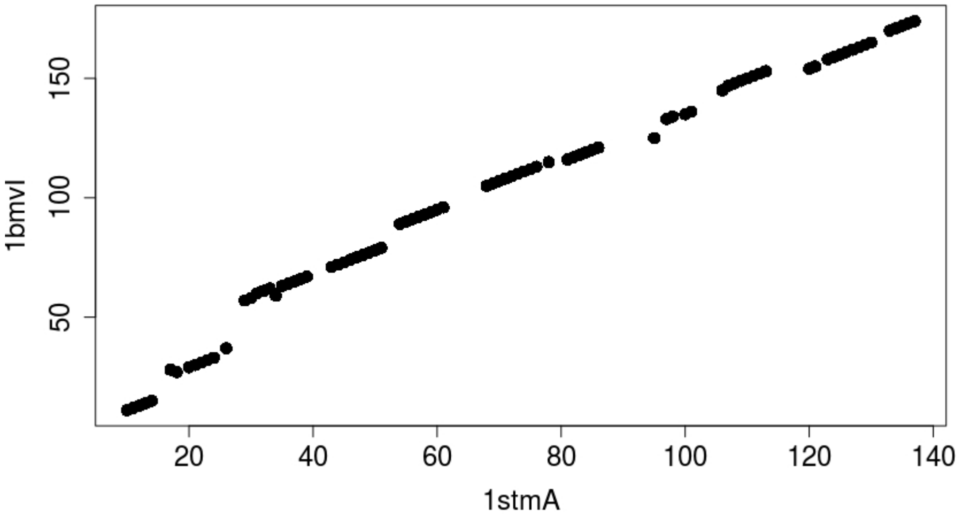
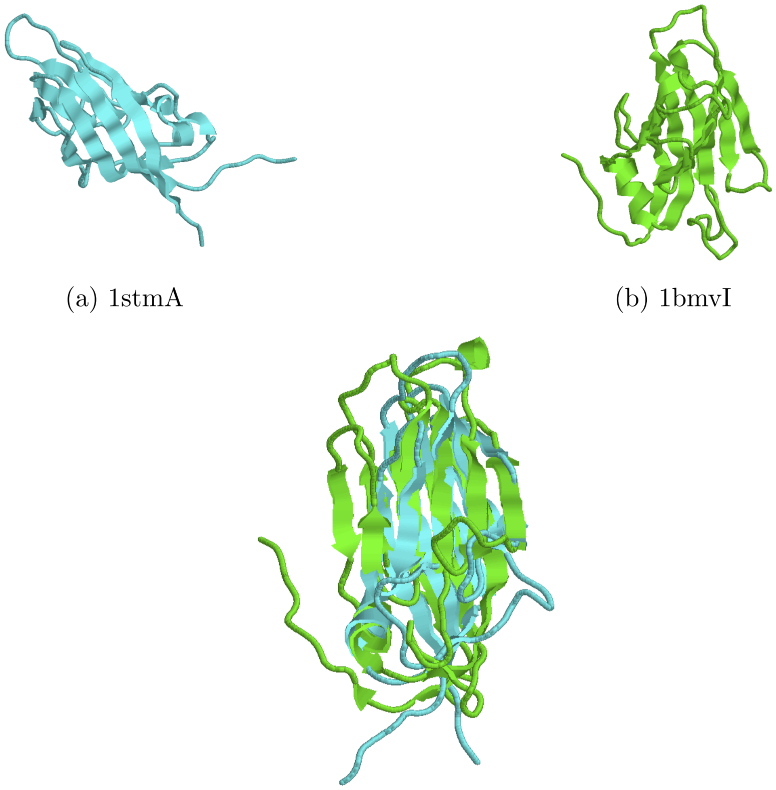


Figure 4.6: Residue correspondence of 1stmA - 1bmvI

Figure 4.7 displays the full atom structure of the two molecules before and after the alignment. The secondary structure of the two molecules present some similarities where the *α-helix* seems to be aligned well between them, as do most of the *β-sheets*.



(a) 1stmA

(b) 1bmvI

Figure 4.7: Full atom structure and structural alignment of proteins 1stmA and 1bmvI

4.3.2 TMalign benchmark dataset

In this subsection we test our method using the benchmark dataset which have also been used in the development of the TMalign method by Zhang and Skolnick (2005). It consists of 200 non-homologous protein molecules with a sequence identity of less than 30%. The lengths of each protein chain ranges from 46 to 1058 atoms. We explore all possible pairwise matches resulting in 19900 distinct comparisons.

Table 4.1 displays the comparison between the methods of CE (Shindyalov and Bourne, 1998), SAL (Kihara and Skolnick, 2003) and TMalign for the pairwise comparisons of the 200 non-homologous protein pairs. To compare the results we use the following metrics :

- Number of matched atoms (M)
- Root Mean Square Distance (RMSD) (2.8.1)
- Template Modelling score (TMscore) (2.8.2)
- The proportion of matched atoms from the whole chain (Coverage)
- Time in seconds needed of each alignment (Time)

As starting points for our method we choose a set of 5 atoms as selected by the **Algorithm 3**.

Algorithm	RMSD	M	TMscore	Coverage (%)	Time(seconds)
Likelihood Alignment	4.34	124.9	0.365	59.5	22.3
CE	6.52	64.3	0.169	34.7	2.25
SAL	7.33	95.3	0.229	47.3	10.00
TMalign	4.99	87.4	0.253	42.0	0.51

Table 4.1: Structural alignments by different algorithms for the 200 non-homologous protein data. Results from CE, SAL, TMalign are taken from Table 1 of Zhang and Skolnick (2005)

The Likelihood Alignment approach finds better scores in all categories of comparison. We have the highest number of matched atoms with an average of 124.9, about 30 more compared to the second highest that of SAL with 95.3.

Also, the RMSD value is the lowest among all methods with 4.34Å. These two results suggest that our final alignments managed to combine more matched atoms with smaller distance between them. For the Coverage proportion we achieve an average value of 59.5%, about 12% more than any other algorithm. Moreover, the TMscore obtained by the Likelihood Alignment is also higher than any other method with an average of 0.365.

Finally, as seen in Table 4.1 our method is the slowest, needing approximately 22 seconds and the fastest is that of TMalign which needs only 0.51 seconds to perform one pairwise alignment. The reasons behind this time difference are:

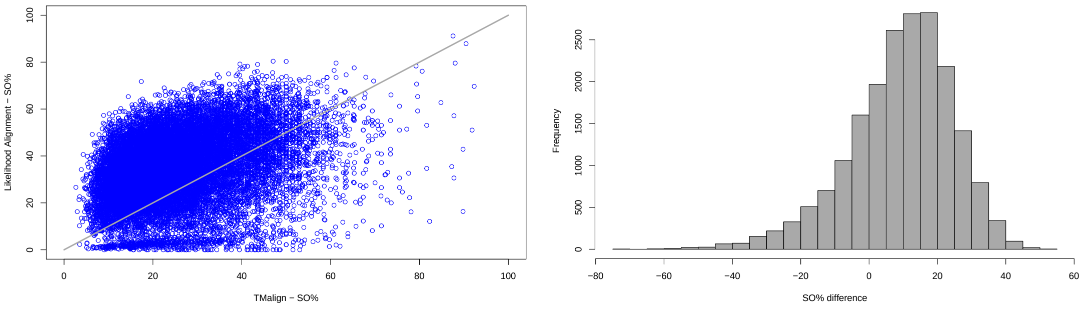
1. The complexity of our algorithm is of order $O(n^3)$ since we need to explore all possible pairwise combinations of atoms and for large proteins this is particularly time consuming.

2. Most of the other algorithms have been coded in programming languages like C or C++, which are significantly faster when compared to R where our method is implemented.

So far the criteria we used for evaluating and comparing the different methods are based only on the geometric similarities of the final alignments, namely the best solutions are those that combine a high number of matched points with the minimum distance. A different way of assessing the structural similarity between two proteins is if they belong to the same fold families based on classifications using CATH (Orengo et al., 1997; Dawson et al., 2017) or SCOP (Murzin et al., 1995) databases. The TMscore tries to quantify this classification approach and as shown in Zhang and Skolnick (2004) it has a strong correlation with the folding properties of an alignment.

In Figure 4.8 we use the Structure Overlap defined in (2.8.3) to compare the Likelihood Alignment method against the TMalign. Figure 4.8a) displays a comparison of the Structure Overlap scores between the two methods for the 19900 pairwise alignments. In 15066 cases the Likelihood Alignment obtained higher SO% scores and for 327 the SO% were the same between the two methods.

Figure 4.8b) shows the distribution of the difference between the SO % scores for each algorithm. As we can see it is skewed to the left having a mean of 9.89%, meaning that about 9.89% of the aligned atoms from the Likelihood Alignment had smaller final distance than the ones from the TMalign.



(a)

(b)

Figure 4.8: Structure Overlap scores for the dataset of the 200 non-homologous proteins between TMalign and the Likelihood Alignment.

4.3.3 HOMSTRAD database

In this subsection, we explore a subset from the HOMSTRAD database using 64 protein pairs with low structure similarity. The Structure Overlap for these ranges between 30% and 70% with an RMSD of at least 2.5Å. This dataset has also been used as a benchmark for the CLICK method (Nguyen et al., 2011) and also previously analysed by Brown et al. (2015).

Table 4.2 displays the average results for the 64 alignments between the Likelihood Alignment and the alternative algorithms: TMalign, SPalignNS (Brown et al., 2015), SPalign (Yang et al., 2012), CLICK, FlexSnap (Salem et al., 2010), MICAN (Minami et al., 2013), HOMSTRAD (Mizuguchi et al., 1998), SALIGN (Braberg et al., 2012), DALI (Holm and Sander, 1997), GANGSTA (Guerler and Knapp, 2008), Geometric Hashing (Bachar et al., 1993) and FATCAT (Ye and Godzik, 2004). All these aforementioned methods are among the most popular ones used for structural alignment and a comparison between them will give us a good idea of the potential of our approach, especially using such a challenging dataset as these 64 protein pairs.

The Likelihood Alignment has the third highest number of matched atoms with 81 tied with SPalign and HOMSTRAD, while also achieves the smallest RMSD compared to these three methods and the 7th overall. As we can see there in not a significant difference with the smallest RMSD reported by SPalignNS and Geometric Hashing which is 1.91Å, however we have matched 9 more atoms compared to them.

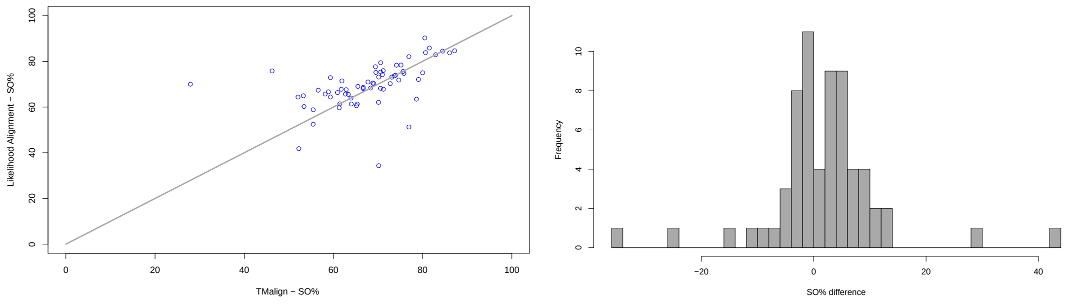
Focusing on the Structure Overlap we notice that our approach has the second best (69.64%) only behind SPalignNS with 72.83%, meaning that about 70% of the matched atoms which have been aligned have a distance of less than 3.5Å. Finally, the Likelihood Alignment also achieves the highest TMscore among the only three methods that we are able to calculate it.

Algorithm	RMSD	M	SO (%)	TMscore
Likelihood Alignment	2.22	81	69.64	0.531
TMalign	2.95	84	67.71	0.493
SPalignNS	1.91	72	72.83	0.527
SPalign	2.66	81	69.27	-
CLICK	1.96	67	68.90	-
FlexSnap	2.23	66	61.37	-
MICAN	2.91	82	61.30	-
HOMSTRAD	3.15	81	59.40	-
SALIGN	2.02	-	67.20	-
DALI	2.00	-	63.00	-
GANGSTA	1.99	-	61.90	-
Geometric Hashing	1.91	-	59.50	-
FATCAT	2.36	-	59.10	-

Table 4.2: Summaries of structural alignments by different algorithms for the "difficult to align" 64 pairs from the HOMSTRAD database. The figures of all alternative the methods except TMalign (which we used the available software online) are taken from Table 2 of Brown et al. (2015).

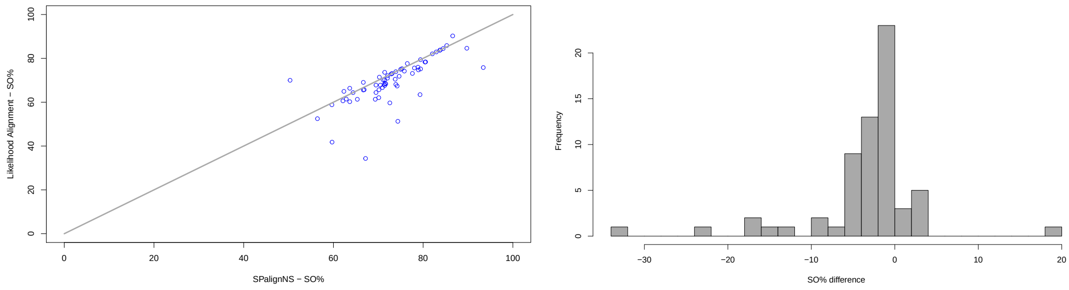
In Figures 4.9 and 4.10 we compare the Likelihood Alignment against the TMalign and SPalignNS methods using the Structure Overlap measure. The left plots of Figures 4.9 and 4.10 display a comparison of the Structure Overlap scores between the Likelihood Alignment and the TMalign and SPalignNS methods respectively, for the 64 protein pairs of Table 4.2. Out of the 64 possible alignments we achieved a better Structure Overlap score in 36 cases more than TMalign and in 10 cases more than SPalignNS. We had the same scores with

TMalign in 9 alignments and with the SPalignNS in 13. The histograms display the distributions for the differences between these three different algorithms. As we can see both distributions are centred close to 0, meaning that there is not much difference in the number of aligned atoms with a distance of less than 3.5Å. More specifically, the Likelihood Alignment compared to TMalign has a mean difference of 2.1% and compared to SPalignNS it has a difference of -3.1%.



(a) (b)

Figure 4.9: Structure Overlap % comparison for the 64 pairs from the HOMSTRAD database between TMalign and Likelihood Alignment.



(a) (b)

Figure 4.10: Structure Overlap % comparison for the 64 pairs from the HOMSTRAD database between SPalignNS and Likelihood Alignment.

4.3.4 Conclusions

- On the TMalign benchmark data we managed to perform better that all the other algorithms in every metric comparison (see Table 4.1), especially in the number of aligned atoms where we had at least 30 more combined with a lower total RMSD.

- On the HOMSTRAD dataset although no method is universally better in all categories, our final alignments result to a good combination of the numbers of matched atoms, RMSD, SO% and TMscore. Also we achieved a TMscore of above 0.5 meaning that we were able to identify protein pairs that might belong to the same fold.

4.4 Comparisons between different likelihood densities

In this Section we test the Likelihood Alignment method against the method proposed by Rodriguez and Schmidler (2014). We also explore the effect of including the amino acid sequence information using the likelihood density of (3.5.2) and the use of the gap penalty function from (3.6.3). We use a dataset of 16 protein pairs that have also been analysed by Ortiz et al. (2002) and Rodriguez and Schmidler (2014). Their protein chain lengths vary from 56 to 188 atoms. As a starting point for these comparisons a set of five atoms from the LGA method has been used. Also, for the gap penalty parameters we follow the approach of Gerstein and Levitt (1998) which suggests that the gap opening penalty should be about 40 times larger than the gap extension penalty. Hence, we choose for the gap opening parameter $g = 4$ and for the gap extension parameter $h = 0.1$. Finally, the PAM250 matrix is used for all the comparisons.

Table 4.3 displays the results between the comparison of the structure-sequence-gap likelihood densityfrom (3.6.3) and the method of Rodriguez and Schmidler (2014) which from now on will be referred to as *RS2014*. Our results are closer to those from *RS2014* when $\lambda = 7.6$ is used. In most of the cases we achieved alignments with solutions of equal or higher number of matched atoms combined with lower RMSD values. For example, it is noticeable the difference in the pair of 1aba-1dsbA where we have the same number of matched atoms but with lower RMSD of about 0.8Å. Moreover, the pairs of 1tnfA-1bmvI and 3chy-1rcf have also significant differences, because in both cases we achieved an alignment with 30 more matched atoms than *RS2014* combined with a lower RMSD of 0.3Å.

Nevertheless, in some cases our algorithm does not perform very well as in the pair of 1mjc-5tssA, where we matched only 29 atoms compared to 52 from *RS2014*. This is probably due to the starting point selection or the choice of the gap parameters. As we discuss later in Table 4.5,these results can vary based on which likelihood function we choose to use. Finally, although the other two λ values of *RS2014* generate solutions with similar number of matched atoms as our method, we have almost in every case obtained a lower RMSD.

Protein1	Protein2	f_{SSG}		*RS2014* ($\lambda = 7.6$)		*RS2014* ($\lambda = 8.6$)		*RS2014* ($\lambda = 9.6$)	
		RMSD	M	RMSD	M	RMSD	M	RMSD	M
1aba	1dsbA	0.8	24	2.2	24	3.7	57	4.7	76
1aba	1trs	2.4	71	3.0	65	3.4	72	3.6	75
1acx	1cobB	2.5	98	2.1	66	3.8	86	4.1	93
1acx	1rbe	2.0	17	2.5	25	2.8	31	4.2	50
1mjc	5tssA	0.8	29	2.3	52	3.0	60	3.9	66
1pgb	5tssA	1.5	39	2.3	39	3.3	55	3.1	55
1plc	1acx	3.5	81	3.4	71	4.0	84	4.6	89
1ptsA	1mup	1.5	54	3.0	76	3.1	83	3.5	88
1tnfA	1bmvI	2.4	107	2.7	70	4.2	109	4.3	113
1ubq	1frd	2.2	64	3.0	62	2.9	62	3.1	65
1ubq	4fxc	2.3	68	2.3	46	2.9	61	3.4	66
2gb1	1ubq	1.7	42	2.1	44	3.4	51	3.3	51
2gb1	4fxc	1.8	41	3.5	35	3.9	53	4.1	55
2rslC	3chy	2.4	75	2.6	43	3.8	76	4.0	81
2tmvP	256bA	2.3	86	2.3	65	2.9	79	4.0	89
3chy	1rcf	2.7	122	3.0	80	4.5	122	4.7	126

Table 4.3: Comparison between the likelihood density with structure, sequence and gap penalty with the method of RS2014.

In the Table 4.4, we compare the effect of the three size and shape likelihood densities presented in Chapter 3: the one with only structure information (3.2.7), the one with structure-sequence information (3.5.3) and the one with structure and a gap penalty function (3.6.4). The likelihood density with only the structure information seems to perform better. The addition of the amino acid sequence information has little effect on our results, leading to almost identically alignment solutions. On the other hand, the inclusion of a gap penalty has a much higher impact on the final alignment, resulting to reduced RMSD values and number of matched atoms. This behaviour is somehow expected since a matched pair that does not follow the sequence order will carry a big penalty and as a result will be dropped out of the final alignment.

Protein1	Protein2	f_S		f_{SS}		f_{SG}	
		RMSD	M	RMSD	M	RMSD	M
1aba	1dsbA	2.8	79	2.8	79	0.9	26
1aba	1trs	2.4	71	2.4	71	2.4	71
1acx	1cobB	2.7	101	2.7	102	2.5	99
1acx	1rbe	3.3	67	3.3	67	2.0	17
1mjc	5tssA	1.7	59	1.7	59	0.7	27
1pgb	5tssA	2.2	54	2.2	54	1.5	39
1plc	1acx	3.6	90	3.7	91	3.0	26
1ptsA	1mup	3.0	100	3.0	100	1.5	54
1tnfA	1bmvI	3.4	131	3.4	131	2.9	122
1ubq	1frd	2.3	66	2.6	70	2.2	64
1ubq	4fxc	2.4	70	2.4	70	2.3	68
2gb1	1ubq	2.6	52	2.6	52	1.7	42
2gb1	4fxc	2.0	44	2.0	45	1.8	42
2rslC	3chy	4.0	110	4.0	110	3.1	94
2tmvP	256bA	2.5	90	2.5	90	2.3	86
3chy	1rcf	2.9	127	2.9	127	2.9	127

Table 4.4: Comparison of the structure, structure-sequence and structure-gap likelihood densities.

Table 4.5 displays a summary of the results for the aforementioned methods compared also to the algorithms of LGA, DALI and TMalign. No approach seems to perform better than the others in all categories of comparison. The $f_S(\cdot)$ density has the most matched atoms with 82 but the RMSD value is higher by 0.7Å compared to the likelihood with structure-sequence and gap penalty, but this difference comes mostly because it has 18 more matched atoms on average. Compared to the method of *RS2014*, it seems to perform also better, since it has more matched atoms and only for $\lambda = 7.6$ it has slightly higher RMSD by 0.1Å.

Another important point to mention is that all versions of the likelihood densities have a TMscore above 0.5, which is considered as a good indication that two proteins belong to the same fold. Finally, very good is also the performance in the score of Structure Overlap especially for the structure and sequence-structure densities where about 70% of the aligned atoms are closer than 3.5Å.

Method	RMSD	M	TMscore	SO (%)
f_S	2.74	82	0.60	70.30
f_{SS}	2.76	82	0.60	70.32
f_{SG}	2.11	63	0.50	59.78
f_{SSG}	2.05	64	0.51	61.29
LGA	2.41	57	0.50	61.67
DALI	3.02	69	0.50	58.85
TMalign	3.06	73	0.52	-
RS2014 $(\lambda = 7.6)$	2.64	54	-	-
RS2014 $(\lambda = 8.6)$	3.48	71	-	-
RS2014 $(\lambda = 9.6)$	3.91	77	-	-

Table 4.5: Summary of structural alignment by different methods for the data of Ortiz et al. (2002).

4.4.1 Conclusions

- The likelihood density of (3.2.7) with only the structure information seems to give the best results based on the TMscore number of matched atoms and Structure Overlap.

- Compared with the method of *RS2014* our approach manages to generate more matched atoms with a smaller overall RMSD in most of the cases. (see Table 4.3).

4.5 Estimation of evolutionary distance

The structure-sequence density (3.5.3) or the structure-sequence with gaps density (3.6.3) provides the opportunity of estimating the evolutionary distance between two proteins. In our modelling approach, the evolutionary distance is characterized through the choice of d in the PAM matrix.

The evolutionary distance is defined as the number of amino acid substitutions that have happened between two protein sequences during a time d. A common practice modelling protein sequences that share a big evolutionary distance is through the use of the PAM250 matrix. However, since structure is more conserved than sequence across time, combing the structural and sequence information during the estimation of the evolutionary distance can give us a better understanding of the relationship of two proteins.

However, there are times that evolutionary distance estimation is important. Since structure is much more conserved than sequence across time incorporating the structural information of a protein within the sequence information is more important.

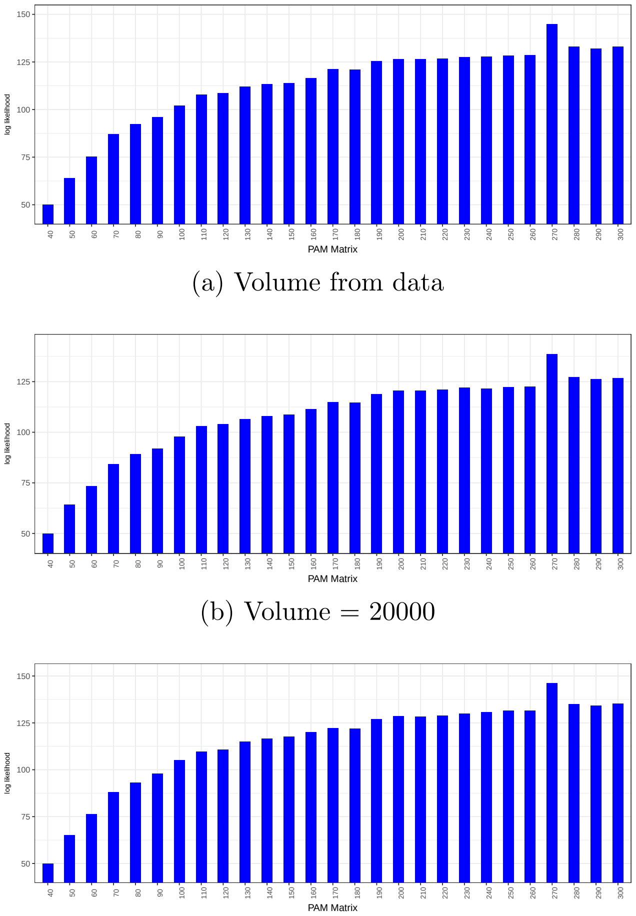
Previous attempts on estimating the evolutionary distance have been made by Koehl and Levitt (2002), Wood and Pearson (1999) and Levitt and Gerstein (1998). In Challis and Schmidler (2012) a diffusion process is used to model the evolutionary distance and a Bayesian approach using sequence information only is adopted by Zhou (1998). Whereas Rodriguez and Schmidler (2014) and Fallaize et al. (2014) use a combined sequence-structure approach and estimate the posterior distribution for the evolutionary distance d.

In our case, we follow a similar approach as in Rodriguez and Schmidler (2014) and Fallaize et al. (2014) but in a likelihood framework. Using **Algorithm 1** and any of the likelihood densities of (3.5.3) or (3.6.3) we can estimate the likelihood mode of the evolutionary distance d. Since the number of the PAM matrices is finite we obtain a likelihood value for each PAM matrix by keeping all the other parameters fixed. The process is similar as before described in Section 2.3 with the addition of an extra optimization step for d. The steps now are the following:

- $\hat{\boldsymbol{M}} = \arg\max_{M} \left[\arg\max_{\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}} \mathcal{L}(\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}, \boldsymbol{M} | \boldsymbol{X}_1, \boldsymbol{X}_2, d) \right]$
- $\hat{d} = \arg\max_{d} \mathcal{L}(d | \boldsymbol{X}_1, \boldsymbol{X}_2, \hat{\boldsymbol{M}}, \hat{\boldsymbol{\mu}}_{(\boldsymbol{M})}, \hat{\sigma}^2_{(\boldsymbol{M})})$

In Figure 4.11 we estimate the evolutionary distance for the pair of *kinases* 1gky-2ak3. We used the likelihood density of (3.6.3). A set of 5 atoms from the LGA solution were selected as stating points and we also fixed the gap opening parameter to be 40 times larger than the gap extension. Last, we consider a set of PAM-d matrices with $d = \{40, 50, 60, \ldots, 300\}$ and three different values for the volume, one calculated from the data, 20000 and 50000.

As we can see in Figure 4.11 a PAM270 mode for d is obtained in all three cases. The same pair of proteins has also been analysed by Rodriguez and Schmidler (2014), Fallaize et al. (2014) and Zhou (1998). The first report a posterior mode between PAM200 and PAM210 , the second report a posterior mode of around PAM260 and the third a multimodal posterior with modes at PAM110, PAM140 and PAM200. However as mentioned in the comments of the second method the volume parameter is affecting both the number of matches and the evolutionary distance estimation, but in our case volume seems not to have an effect as we have obtained almost identical estimations for d in all three cases.



(a) Volume from data

(b) Volume = 20000

(c) Volume = 50000

Figure 4.11: Evolutionary distance estimation for the pair 1gky-2ak3, using three different values for Volume

4.6 Multiple matching example

For this example we use three different datasets to evaluate the performance of the multiple alignment method from Section 3.8. The first dataset is three steroid molecules from the CoMFA database (Cramer et al., 1988). The molecules from this database have been extensively used as a benchmark for testing drug design methods or for evaluating the 3-dimensional quantitative structure-activity relationship QSAR (Coats, 1998). Here, we select three steroid molecules, the *aldosterone*, the *cortisone* and the *prednisolone*. Each of these three molecules has 54 atoms in 3 dimensions. For this and the following examples we only consider atoms which are matched in all molecules and not partial matches between some of them.

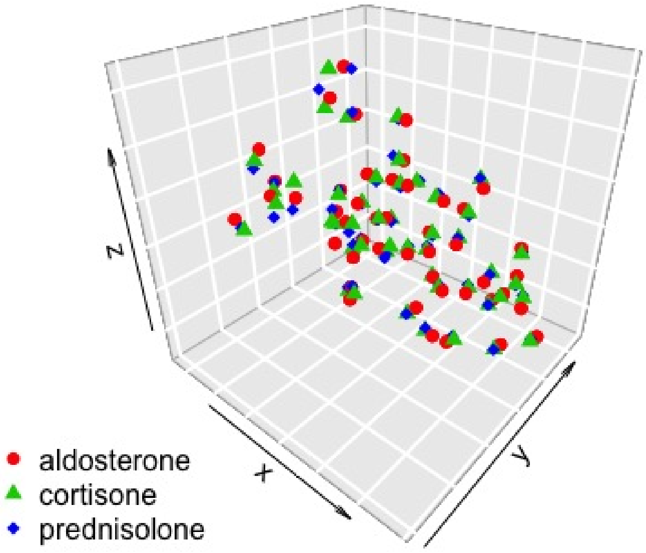
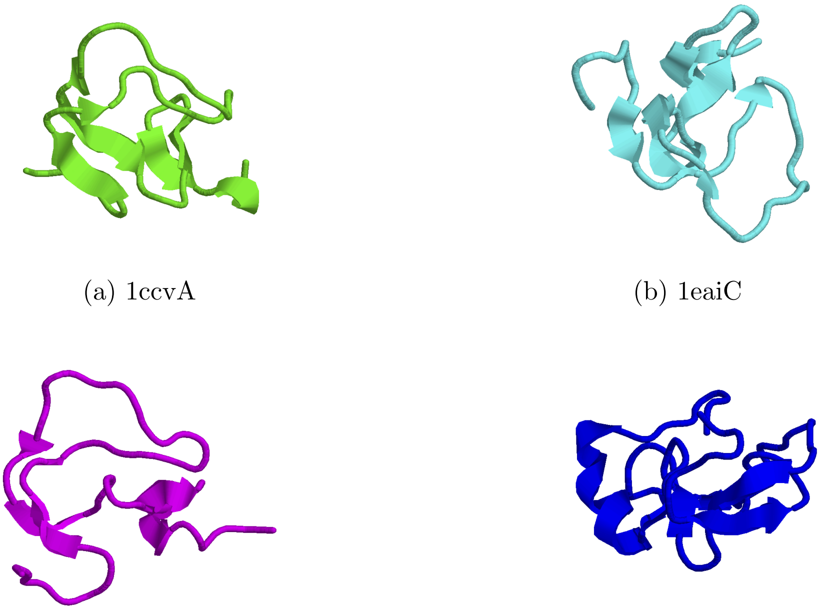


Figure 4.12: Structural alignment of the three steroid molecules aldosterone, cortisone and prednisolone from the CoMFA database.

To align the three molecules we use **Algorithm 1** alongside with the likelihood density for multiple alignment of (3.8.1) and the process described in Section 3.8. Thus, we obtain an alignment of 47 common matched atoms between them with an average RMSD of 0.2Å, a TMscore of 0.86 and the Structure Overlap is 87.04%. The structural alignment of the optimally rotated data can be seen in Figure 4.12. In the paper of Ruffieux and Green (2009) where the same dataset has been analysed they report a total of 44 matched atoms for the three molecules. Both methods have similar results and the 44 matched pairs of atoms from Ruffieux and Green (2009) are also present in our alignment.



(a) 1ccvA

(b) 1eaiC

(c) 1ate

(d) 1couA

Figure 4.13: Full atom structure of 1ccvA, 1eaiC, 1ate and 1couA.

For the second example we use a group of *serine protease inhibitors* from the HOMSTRAD database. This group is composed by four molecules, the *chymotrypsin inhibitor* 1ccvA with 56 atoms, *the trypsin inhibitor* 1ate with 62 atoms, the *chymotrypsin/elastase isoinhibitor* 1eaiC with 61 atoms and the *anticoagulant protein* 1couA with 85 atoms. They share a sequence identity of 36%. The full atom structure of these molecules can be seen in Figure 4.13.

For the structural alignment the same procedure as in the previous example used. The set of starting points in now from the solution of the MASS method by Dror et al. (2003a) and Dror et al. (2003b). Figure 4.14 displays the full atom structure of the four proteins after they have been aligned. We obtained 34 matched atoms among them with an average RMSD of 4.2Å. As we can see from Figure 4.13 the four proteins have some differences in their structure especially in their secondary structure. Only the 1couA has both an α-*helix* and a β-*sheet* with the other three having only β-*sheets* in their secondary structure. We managed to align some parts between the β-sheets of 1couA,1eaiC and 1ate.

However, there also seems to be some misalignment in some parts between them, hence the increased RMSD. This could either be from the choice of the starting points or from the fact that since we only consider common matches between all four molecules. For example, some parts of 1ccvA have been aligned although they do not share a very similar structure with the parts of the other three molecules.

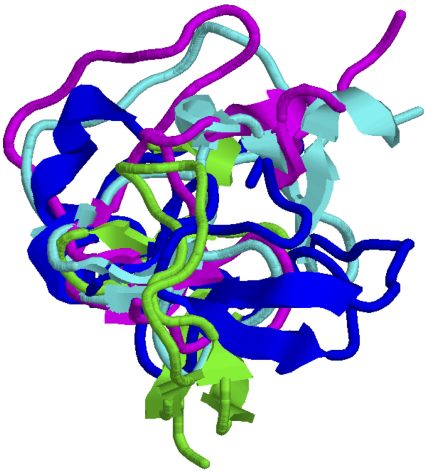


Figure 4.14: Structural alignment of 1ccvA, 1eaiC, 1ate and 1couA.

Chapter 5

Posterior mode alignment

5.1 Introduction

In this Chapter we explore a different approach for obtaining the mode of the matching matrix $\boldsymbol{M}$. A posterior mode alignment method is considered in which prior distributions over the unknown parameters of $\boldsymbol{\mu}$, σ^2 and $\boldsymbol{M}$ are assigned.

In the Bayesian literature previous work has been done in this area. Green and Mardia (2006) use a symmetric model with a Poisson process as a prior for the matching matrix $\boldsymbol{M}$, whereas Dryden et al. (2007) and Schmidler (2007) use a *Procrustes* model with a uniform prior for $\boldsymbol{M}$. In the papers of Rodriguez and Schmidler (2014), Kenobi and Dryden (2012) and Fallaize et al. (2014) extensions of the previous models are considered, introducing different priors over $\boldsymbol{M}$.

Although, the *full* Bayesian approach is available for these methods and the posterior distribution of $\boldsymbol{M}$ is defined in every case, in structural alignment of protein molecules, a point estimate of the match matrix $\boldsymbol{M}$ is often required. Most of the aforementioned approaches, due to the restriction of one-to-one matches, use optimization algorithms to obtain a single alignment through the posterior distribution of $\boldsymbol{M}$.

In our approach, we follow the ideas presented in Section 2.3 where we conditionally optimize over the unknown parameters in order to estimate the posterior mode of $\boldsymbol{M}$ directly. Since we are not interested in the whole posterior distribution, this approach simplifies the overall alignment procedure, making it more efficient computationally especially when comparisons within a protein database are needed.

In Section 5.2 we discuss the prior selection for the unknown parameters of $\boldsymbol{\mu}, \sigma^2$ and $\boldsymbol{M}$. Our approach differs from the previous Bayesian models on how we treat the mean parameter $\boldsymbol{\mu}$. For example, Dryden et al. (2007) and Schmidler (2007) fix one of the two molecules as the mean and try to align the other to it. Green and Mardia (2006) choose to integrate the mean $\boldsymbol{\mu}$ out of the likelihood density. We choose to treat $\boldsymbol{\mu}$ as a random parameter assigning a prior to it. Since in every step of the optimization process of $\boldsymbol{\mu}$ and σ^2 a fixed alignment is required, our prior mean $\boldsymbol{\mu}_0$ is defined as a function of the match matrix $\boldsymbol{M}$. In the following Section we explain in more details how this prior is selected in every step.

In Section 5.3 we describe the optimization steps to obtain the modes for $\boldsymbol{\mu}$ and σ^2 and how we estimate the posterior mode of $\boldsymbol{M}$. In Section 5.4, we test the efficiency of our approach using simulated data and in Section 5.5 we present some examples using real protein data and compare the results with the Likelihood approach and other alignment algorithms.

5.2 Prior selection

In Section 2.3 we described the general likelihood density $\mathcal{L}(\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2)$ of (3.2.1), which consists of two components a Normal density $f_N(\boldsymbol{X}_1^M, \boldsymbol{X}_2^M|\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2)$ for the matched parts and a Uniform density $f_U(\boldsymbol{X}_1^{-M}, \boldsymbol{X}_2^{-M}|\boldsymbol{M}, V)$ for the unmatched. Since we treat the volume parameter V as fixed, we need to specify prior distributions for $\boldsymbol{M}, \boldsymbol{\mu}$ and σ^2.

5.2.1 Priors for μ and σ^2

Here, we consider a joint prior distribution for the parameters $\boldsymbol{\mu}$ and σ^2. Following the Normality assumption for the matched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ a common choice in Bayesian literature (Gelman et al., 2014) is the conjugate *Normal - Inverse Gamma* distribution with parameters $\boldsymbol{\mu}_0, \lambda, \alpha_0, \beta_0$. The mean $\boldsymbol{\mu}$ corresponds only to the matched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, hence it depends in the current alignment specified by $\boldsymbol{M}$. Thus, the prior parameter $\boldsymbol{\mu}_0$ will also depend on $\boldsymbol{M}$. By $\boldsymbol{\mu}_{0_{(M)}}$ we refer to the prior mean $\boldsymbol{\mu}_0$ for a given alignment $\boldsymbol{M}$. The prior density of $\boldsymbol{\mu}$ and σ^2 can be defined as follows

$$\pi(\boldsymbol{\mu}, \sigma^2) = \left(\frac{\lambda}{2\pi\sigma^2}\right)^{-\frac{3p}{2}} \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \left(\frac{1}{\sigma^2}\right)^{\alpha_0+1} \exp\left\{-\frac{2\beta_0 + \lambda||\boldsymbol{\mu} - \boldsymbol{\mu}_{0_{(M)}}||^2}{2\sigma^2}\right\} \tag{5.2.1}$$

Note that the prior density (5.2.1) can be written as $\pi(\boldsymbol{\mu}, \sigma^2) = \pi(\boldsymbol{\mu}|\sigma^2)\pi(\sigma^2)$ which will lead to

$$\pi(\boldsymbol{\mu}|\sigma^2) = \left(\frac{\lambda}{2\pi\sigma^2}\right)^{-\frac{3p}{2}} \exp\left\{-\frac{\lambda||\boldsymbol{\mu} - \boldsymbol{\mu}_{0_{(M)}}||^2}{2\sigma^2}\right\} \tag{5.2.2}$$

$$\pi(\sigma^2) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \left(\frac{1}{\sigma^2}\right)^{\alpha_0+1} \exp\left\{-\frac{\beta_0}{\sigma^2}\right\} \tag{5.2.3}$$

Now, we describe the definition of prior mean $\boldsymbol{\mu}_{0_{(M)}}$. As we mentioned before the prior mean is considered as a function of the match matrix $\boldsymbol{M}$, since for the estimation of the common mean $\boldsymbol{\mu}$ only the matched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ at a given time are involved. Therefore, as described in **Algorithm 1** for the optimization of $\boldsymbol{M}$ we explore new possible matches between the atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ in each step, hence $\boldsymbol{M}$ and as a result $\boldsymbol{\mu}_{0_{(M)}}$ will change.

Before we describe the process of selecting $\boldsymbol{\mu}_{0_{(M)}}$, we should explain the intuition behind this prior choice. In general the prior mean $\boldsymbol{\mu}_0$ should represent our beliefs for the mean locations which create the matched parts of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. The problem arises from the difference in the dimensionality between the two proteins and the fact that we have no prior information regarding the correspondence between each atom making the process of defining a common prior mean $\boldsymbol{\mu}_0$ difficult.

To overcome these problems, we choose to define two matrices $\boldsymbol{\mu}_{0_1}$ and $\boldsymbol{\mu}_{0_2}$, corresponding to each $\boldsymbol{X}_i$. These two matrices will contain the prior beliefs for the locations of each atom for each $\boldsymbol{X}_i$ and will act as a *pool* of prior information. In order to create $\boldsymbol{\mu}_{0_{(M)}}$ we will only select the relevant atoms which are matched at a given time (based on $\boldsymbol{M}$).

We use a simple example to illustrate the selection process of $\boldsymbol{\mu}_{0_{(M)}}$. Consider two protein molecules $\boldsymbol{X}_{1_i}$ with atoms $i = 1, \ldots, 6$ and $\boldsymbol{X}_{2_j}$ with atoms $j = 1, \ldots, 8$. Our prior beliefs suggest that the following pairs should be considered as matched:

$$(\boldsymbol{X}_{1_1}, \boldsymbol{X}_{2_1}), (\boldsymbol{X}_{1_2}, \boldsymbol{X}_{2_3}), (\boldsymbol{X}_{1_5}, \boldsymbol{X}_{2_6}), (\boldsymbol{X}_{1_6}, \boldsymbol{X}_{2_8})$$

and the prior locations for the matched atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be written as $\left[\mu_{0_1}^M, \mu_{0_2}^M, \mu_{0_3}^M, \mu_{0_4}^M\right]$, where $\mu_{0_1}^M$ contains the prior location for the pair $(\boldsymbol{X}_{1_1}, \boldsymbol{X}_{2_1})$, $\mu_{0_2}^M$ for the pair $(\boldsymbol{X}_{1_2}, \boldsymbol{X}_{2_3})$ and so on. Now, each $\boldsymbol{\mu}_{0_i}$ will have the following form:

$$\boldsymbol{\mu}_{0_1} = \begin{bmatrix} \mu_{0_1}^M \\ \mu_{0_2}^M \\ \mu_{0_3}^1 \\ \mu_{0_4}^1 \\ \mu_{0_3}^M \\ \mu_{0_4}^M \end{bmatrix} \qquad \boldsymbol{\mu}_{0_2} = \begin{bmatrix} \mu_{0_1}^M \\ \mu_{0_2}^2 \\ \mu_{0_2}^M \\ \mu_{0_4}^2 \\ \mu_{0_5}^2 \\ \mu_{0_3}^M \\ \mu_{0_7}^2 \\ \mu_{0_4}^M \end{bmatrix} \tag{5.2.4}$$

where $\mu_{0_3}^1$ contains prior information for the atom $\boldsymbol{X}_{1_3}$, $\mu_{0_2}^2$ contains prior information for atom $\boldsymbol{X}_{2_2}$ and so on.

Now, let assume that at a given step of the optimization process for $\boldsymbol{M}$ the following pairs of atoms are considered as matched :

$$(\boldsymbol{X}_{1_1}, \boldsymbol{X}_{2_1}), (\boldsymbol{X}_{1_2}, \boldsymbol{X}_{2_4}), (\boldsymbol{X}_{1_3}, \boldsymbol{X}_{2_5}), (\boldsymbol{X}_{1_5}, \boldsymbol{X}_{2_8})$$

Then, one way of defining $\boldsymbol{\mu}_{0_{(M)}}$ for this step will be to take the average locations of the corresponding atoms from each $\boldsymbol{\mu}_{0_1}$ and $\boldsymbol{\mu}_{0_2}$ as

$$\boldsymbol{\mu}_{0_{(M)}} = \begin{bmatrix} \frac{\mu_{0_1}^M + \mu_{0_1}^M}{2} \\ \frac{\mu_{0_2}^M + \mu_{0_4}^2}{2} \\ \frac{\mu_{0_3}^1 + \mu_{0_5}^2}{2} \\ \frac{\mu_{0_3}^M + \mu_{0_4}^2}{2} \end{bmatrix}$$

5.2.2 Priors for the match matrix M

Here, we describe the prior choices for the matching matrix $\boldsymbol{M}$. The first choice is a uniform prior, a similar prior has also been used by Dryden et al. (2007). The second choice is a gap penalty prior which also been used by Schmidler (2007), Rodriguez and Schmidler (2014) and Fallaize et al. (2014).

As we have previously described in Section 2.2 the match matrix $\boldsymbol{M}$ has dimensions of $k \times l$, with only one non-zero entry in each row and column. Then without loss of generality, if $k \leq l$ and by assuming that each row of $\boldsymbol{M}$ is independently distributed the uniform prior for the ith-row of $\boldsymbol{M}$ will be

$$\pi_1(\boldsymbol{M}_{ij} = 1) = \frac{1-q}{l} \qquad j = 1, \ldots, l \tag{5.2.5}$$

where q is the probability of atom i to be unmatched. We choose $q = \frac{1}{l+1}$ hence, under this prior density the match matrix $\boldsymbol{M}$ is uniformly distributed in the space of all possible $k \times l$ match matrices. The motivation for choosing a uniform prior although it might not seem a natural choice was that we wanted all prior information regarding the possible matches of $\boldsymbol{M}$ to be drawn from the geometrical information provided by the prior of (5.2.1) for $\boldsymbol{\mu}$ and σ^2 as also done in the likelihood approach described in Chapter 3.

Our second prior choice for $\boldsymbol{M}$ is a gap penalty prior. In Section 3.6 we considered a gap penalty in the likelihood function in order to penalize for any gap openings in the sequence order. Here, we use the same gap penalty function but as a prior for the match matrix $\boldsymbol{M}$. This prior has also been used in the

Bayesian literature by Schmidler (2007), Rodriguez and Schmidler (2014) and Fallaize et al. (2014). For given gap opening and extension parameters g and h respectively the gap penalty prior for $\boldsymbol{M}$ will be

$$\pi_2(\boldsymbol{M}|g,h) = \mathcal{C}(g,h)\exp\{U(g,h)\} \tag{5.2.6}$$

where $\mathcal{C}(g,h)$ is the normalizing constant, and $U(g,h)$ the gap penalty function described in (3.6.1). In comparison with the Uniform prior of (5.2.5) this choice provides extra information on possible matches between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ by penalizing for gaps created in the sequence order.

5.3 Posterior alignment

5.3.1 Posterior distribution

Using the prior assumptions from the previous Section we can describe the two possible posterior distributions as below:

- Using the Uniform prior on $\boldsymbol{M}$:

$$p_1(\boldsymbol{M},\boldsymbol{\mu},\sigma^2|\boldsymbol{X}_1,\boldsymbol{X}_2,V) \propto \mathcal{L}(\boldsymbol{X}_1,\boldsymbol{X}_2|\boldsymbol{M},\boldsymbol{\mu},\sigma^2,V)\pi(\boldsymbol{\mu}|\sigma^2)\pi(\sigma^2)\pi_1(\boldsymbol{M}) \tag{5.3.1}$$

- Using the gap penalty prior on $\boldsymbol{M}$:

$$p_2(\boldsymbol{M},\boldsymbol{\mu},\sigma^2|\boldsymbol{X}_1,\boldsymbol{X}_2,V,g,h) \propto \mathcal{L}(\boldsymbol{X}_1,\boldsymbol{X}_2|\boldsymbol{M},\boldsymbol{\mu},\sigma^2,V)\pi(\boldsymbol{\mu}|\sigma^2)\pi(\sigma^2)\pi_2(\boldsymbol{M}|g,h) \tag{5.3.2}$$

where $\mathcal{L}(\boldsymbol{X}_1,\boldsymbol{X}_2|\boldsymbol{M},\boldsymbol{\mu},\sigma^2,V)$ is the likelihood function from (3.2.7).

Our main objective is to obtain the posterior mode of $\boldsymbol{M}$ from the posterior distribution of either (5.3.1) or (5.3.2) defined above. In order to do this, we follow the same procedure as in the likelihood case which is described in Section 2.3 which depend on the following optimization

$$\hat{\boldsymbol{M}} = \arg\max_{\boldsymbol{M}} \left[\arg\max_{\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}} p(\boldsymbol{\mu}_{(\boldsymbol{M})}, \sigma^2_{(\boldsymbol{M})}, \boldsymbol{M} | \boldsymbol{X}_1, \boldsymbol{X}_2, V) \right] \tag{5.3.3}$$

Before we proceed to the optimization steps to derive the posterior modes of $\boldsymbol{M}, \boldsymbol{\mu}$ and σ^2 the posterior densities of (5.3.1) and (5.3.2) should be invariant under the transformation parameters of translation and rotation.

As described in Chapter 3 by using the decomposition of (3.2.3) the data $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ represent the observed *size and shape* data $\boldsymbol{\Delta O}$. Hence, both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are observed under the similarity transformations of (2.2.2), with a rotation and translation parameter.

To remove the location information we use the Helmertized landmarks described in Section 2.2. For making our data invariant under the rotation effect we choose to integrate the rotation parameter $\boldsymbol{R}$ out of the posterior densities. Note that by $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ throughout the rest of this Chapter we refer to the *observed* size and shape data after the Helmertized transformation. This is similarly done in Chapter 3, which will lead to use the *size and shape* likelihood of (3.2.7). Then the two *size and shape* posterior densities become:

- For the Uniform prior on $\boldsymbol{M}$:

$$\begin{aligned} p_{S_1}(\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2 | \boldsymbol{X}_1, \boldsymbol{X}_2, V) &\propto \int\limits_{\boldsymbol{R}_i \in SO(3)} p_1(\boldsymbol{M}, \boldsymbol{\mu}, \sigma^2 | \boldsymbol{X}_1^\star, \boldsymbol{X}_2^\star, V) d\boldsymbol{R}_i \\ &\propto \left(\frac{1-q}{l}\right)^p V^{-(k+l-2p)} (\sigma^2)^{-\alpha} \exp\left\{ -\frac{2\beta_0 + \lambda ||\boldsymbol{\mu} - \boldsymbol{\mu}_{0_{(M)}}||^2 + \sum\limits_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}||^2}{2\sigma^2} \right\} \\ &\times \prod_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} \exp\left\{ \frac{\operatorname{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}^t\right)}{\sigma^2} \right\} d\boldsymbol{R}_i \quad (5.3.4) \end{aligned}$$

- For the gap penalty prior on $\boldsymbol{M}$:

$$p_{S_2}(\boldsymbol{M},\boldsymbol{\mu},\sigma^2|\boldsymbol{X}_1,\boldsymbol{X}_2,V,g,h) \propto \int\limits_{\boldsymbol{R}_i\in SO(3)} p_2(\boldsymbol{M},\boldsymbol{\mu},\sigma^2|\boldsymbol{X}_1^\star,\boldsymbol{X}_2^\star,V,g,h)d\boldsymbol{R}_i$$
$$\propto \exp\{U(g,h)\}\,V^{-(k+l-2p)}(\sigma^2)^{-\alpha}\exp\left\{-\frac{2\beta_0+\lambda||\boldsymbol{\mu}-\boldsymbol{\mu}_{0_{(M)}}||^2+\sum\limits_{i=1}^{2}||\boldsymbol{X}_i^M||^2+2||\boldsymbol{\mu}||^2}{2\sigma^2}\right\}$$
$$\times\prod_{i=1}^{2}\int\limits_{\boldsymbol{R}_i\in SO(3)}\exp\left\{\frac{\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i^M\boldsymbol{\mu}^t\right)}{\sigma^2}\right\}d\boldsymbol{R}_i \quad (5.3.5)$$

where $\alpha=\alpha_0+\frac{9p}{2}+1$ and $V,\boldsymbol{\mu}_0,\alpha_0,\beta_0,\lambda,g$ and h are considered as fixed parameters. Also, $\boldsymbol{X}_1^\star$ and $\boldsymbol{X}_2^\star$ represent the full unobserved Normal data.

5.3.2 Posterior modes of $\boldsymbol{\mu}$, σ^2 and $\boldsymbol{M}$

The first part for obtaining the posterior mode of $\boldsymbol{M}$ as seen in (5.3.3) is to optimize (5.3.4) or (5.3.5) over $\boldsymbol{\mu}$ and σ^2 for a given alignment $\boldsymbol{M}$. In order to do this we use the EM algorithm of Section 2.4 which can also sufficiently estimate the modes of a posterior distribution (Gelman et al., 2014). Since in this step $\boldsymbol{M}$ is fixed the conditional log-posterior of (5.3.4) or (5.3.5) is the same and is given by:

$$\log p_{S_1}(\boldsymbol{\mu},\sigma^2|\boldsymbol{M},\boldsymbol{X}_1,\boldsymbol{X}_2) \propto -\alpha\log\sigma^2-\frac{2\beta_0+\lambda||\boldsymbol{\mu}-\boldsymbol{\mu}_{0_{(M)}}||^2+\sum\limits_{i=1}^{2}||\boldsymbol{X}_i^M||^2+2||\boldsymbol{\mu}||^2}{2\sigma^2}$$
$$+\sum_{i=1}^{2}\log\int\limits_{\boldsymbol{R}_i\in SO(3)}\exp\left\{\frac{\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i^M\boldsymbol{\mu}^t\right)}{\sigma^2}\right\}d\boldsymbol{R}_i \quad (5.3.6)$$

Again, as in Chapter 3 the missing data in our case are the rotations $\boldsymbol{R}_i$ and since the prior density of (5.2.3) for $\boldsymbol{\mu}$ and σ^2 does not depend on $\boldsymbol{R}_i$ the steps of the EM at the $t-th$ iteration will be the following:

- *Expectation step*: Evaluate the function $Q(\boldsymbol{\mu},\sigma^2|\boldsymbol{\mu}_{t-1},\sigma^2_{t-1})$ for given values of $\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1}$ as follows:

$$\begin{aligned} Q(\boldsymbol{\mu}, \sigma^2|\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1}) &= \mathbb{E}_{\boldsymbol{R}_i|\boldsymbol{X}_i}\left[\log p_{S_1}(\boldsymbol{\mu}, \sigma^2|\boldsymbol{M}, \boldsymbol{X}_1, \boldsymbol{X}_2)\right] \\ &= \mathbb{E}_{\boldsymbol{R}_i|\boldsymbol{X}_i}\left[\log f_S(\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{M}, \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})\right] + \log \pi(\boldsymbol{\mu}_{t-1}|\sigma^2_{t-1}) \\ &\qquad + \log \pi(\sigma^2_{t-1}) \end{aligned}$$

where $f_S(\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{M}, \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})$ is the *size and shape* density from (3.2.7). As we can see the *Expectation step* is the same as in the case of Likelihood Alignment.

- *Maximization step* : Maximize the function $Q(\boldsymbol{\mu}, \sigma^2|\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})$ with respect to $\boldsymbol{\mu}$ and σ^2.

  For $\boldsymbol{\mu}$:

$$\frac{\partial Q(\boldsymbol{\mu}, \sigma^2|\boldsymbol{\mu}_{t-1}, \sigma^2_{t-1})}{\partial \boldsymbol{\mu}} = 0$$

  and expanding this we have

$$-\frac{\lambda(\boldsymbol{\mu} - \boldsymbol{\mu}_{0_{(M)}})}{\sigma^2_{t-1}} + \frac{1}{\sigma^2_{t-1}} \sum_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} \frac{(\boldsymbol{R}_i \boldsymbol{X}_i^M - \boldsymbol{\mu}) e^{A_i} d\boldsymbol{R}_i}{e^{A_i} d\boldsymbol{R}_i} = 0$$

  solving for $\boldsymbol{\mu}$ the updated value at the t-th iteration will be

$$\boldsymbol{\mu}_t = (2+\lambda)^{-1} \left[\sum_{i=1}^{2} \frac{\int\limits_{\boldsymbol{R}_i \in SO(3)} \boldsymbol{R}_i \boldsymbol{X}_i^M e^{A_i} d\boldsymbol{R}_i}{\int\limits_{\boldsymbol{R}_i \in SO(3)} e^{A_i} d\boldsymbol{R}_i} + \lambda \boldsymbol{\mu}_0 \right] \tag{5.3.7}$$

  where $A_i = -\frac{\mathrm{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i^M \boldsymbol{\mu}_{t-1}^t\right)}{\sigma^2_{t-1}}$

  For σ^2:

$$\frac{\partial Q(\boldsymbol{\mu}, \sigma^2|\boldsymbol{\mu}_t, \sigma^2_{t-1})}{\partial \sigma^2} = 0$$

Which leads to

$$-\alpha + \frac{\beta_0}{\sigma^2} + \frac{\lambda||\boldsymbol{\mu}_t - \boldsymbol{\mu}_{0_{(M)}}||^2}{2\sigma^2} + \frac{1}{2\sigma^2}\sum_{i=1}^{2}\frac{\int\limits_{\boldsymbol{R}_i \in SO(3)} ||\boldsymbol{R}_i\boldsymbol{X}_i^M - \boldsymbol{\mu}_t||^2 e^{A_i} d\boldsymbol{R}_i}{\int\limits_{\boldsymbol{R}_i \in SO(3)} e^{A_i} d\boldsymbol{R}_i} = 0$$

and solving for σ^2

$$\sigma_t^2 = \frac{1}{\alpha}\times\left[\beta_0 + \frac{\lambda||\boldsymbol{\mu}_t - \boldsymbol{\mu}_{0_{(M)}}||^2 + \sum_{i=1}^{2}||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}_t||^2}{2} - \operatorname{tr}\left(\sum_{i=1}^{2}\frac{\int\limits_{\boldsymbol{R}_i \in SO(3)} \boldsymbol{R}_i\boldsymbol{X}_i^M e^{A_i}\boldsymbol{\mu}_t^t d\boldsymbol{R}_i}{\int\limits_{\boldsymbol{R}_i \in SO(3)} e^{A_i} d\boldsymbol{R}_i}\right)\right]$$

Furthermore, substituting $\sum_{i=1}^{2}\frac{\int\limits_{\boldsymbol{R}_i \in SO(3)} \boldsymbol{R}_i\boldsymbol{X}_i^M e_i^A d\boldsymbol{R}_i}{\int\limits_{\boldsymbol{R}_i \in SO(3)} e_i^A d\boldsymbol{R}_i} = \boldsymbol{\mu}_t(2+\lambda) - \lambda\boldsymbol{\mu}_0$ from (5.3.7), the new update of σ^2 at the t-th iteration as

$$\sigma_t^2 = \frac{1}{\alpha}\left[2\beta_0 + \lambda||\boldsymbol{\mu}_t - \boldsymbol{\mu}_{0_{(M)}}||^2 + \sum_{i=1}^{2}||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}_t||^2 - (2+\lambda)\operatorname{tr}(\boldsymbol{\mu}_t) + \lambda\operatorname{tr}\left(\boldsymbol{\mu}_{0_{(M)}}\right)\right] \tag{5.3.8}$$

The prior $\boldsymbol{\mu}_{0_{(M)}}$ acts as an extra observation, with the parameter λ as a weight quantifying our confidence about it. Using $\lambda = 1$ our model essentially becomes the likelihood model with 3 observations. Large values for λ indicate a strong believe about the prior, hence the posterior mean will be shifted towards $\boldsymbol{\mu}_{0_{(M)}}$.

The optimization on $\boldsymbol{M}$ of either posterior distributions (5.3.4) or (5.3.5) can be carried out along the same ideas we described in Section 3.4 for the likelihood alignment case. We can use again **Algorithm 1** replacing the likelihood density with the corresponding posterior densities from (5.3.4) or (5.3.5). Also, the same applies if we want to align more than two proteins simultaneously since the posterior distributions remain the same if we have more than two molecules. Therefore, using the likelihood function of (3.8.1) and the procedure described in Section 3.8 we can obtain the posterior mode of $\boldsymbol{M}$ when more than two proteins are involved.

5.4 Simulations

In this Section we test the performance of the Posterior Alignment method presented in the previous Sections using simulated data. This Section has two parts. First we estimate the effectiveness of our approach to obtain the posterior modes of $\boldsymbol{\mu}$ and σ^2. This corresponds to the first part of the optimization from (5.3.3). Second, we test our method on obtaining the posterior mode of $\boldsymbol{M}$, which is the second part of the optimization from (5.3.3).

5.4.1 Simulation for the posterior mode of $\boldsymbol{\mu}$ and σ^2

For this simulation study we are interested in the mode estimation of $\boldsymbol{\mu}$ and σ^2 from the posterior density of (5.3.4) for a given $\boldsymbol{M}$. Since, $\boldsymbol{M}$ is fixed we can treat our data as regular shape observations not concerning about the alignment part. Therefore, to test the performance of our approach we choose to simulate different sample sizes for $\boldsymbol{X}_i$.

The process of creating the simulated data is similar to the one described in Section 4.2. Inside a cube with volume L^3, we create a mean shape with 25 landmarks subject to the constraint that each landmark has at least a minimum distance d_{min} with all the others. Our simulated data are n Normal observations from that mean with a variance σ^2. The parameter settings used are the following:

- $d_{min} = 2, L = 20, n = \{10, 50, 100, 500\}, \sigma = \{0.5, 1, 2, 2.5\}$
- The prior mean $\boldsymbol{\mu}_0$ is the true mean with some random Normal error.
- $\alpha_0 = 5, \beta_0 = 15$
- $\lambda = \{1, n/2, n\}$

Table 5.1 displays the simulation results for the data created above. The mode estimations seem to be good for both the mean and variance . More specifically for $\sigma = 0.5$ even for the sample sizes of 10 or 50 the distance between the posterior and the true mean is really close and as the sample size increases this distance becomes smaller ($d(\hat{\mu}, \mu) = 0.151$ for $n = 500$). The estimation of σ seems to be also good and in every case it tends to the true value. The λ parameter seems to have small effect on the actual estimates of both $\boldsymbol{\mu}$ and σ. As we mentioned earlier the prior mean $\boldsymbol{\mu}_0$ acts as an extra observation with a weight specified by λ. However, even for a small sample size of $n = 10$ and a weight of $\lambda = n$ the difference between the estimates is small, at the magnitude of 0.01.

λ	Sample	$\sigma = 0.5$		$\sigma = 1$		$\sigma = 2$		$\sigma = 2.5$	
		$d(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})$	$\hat{\sigma}$	$d(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})$	$\hat{\sigma}$	$d(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})$	$\hat{\sigma}$	$d(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})$	$\hat{\sigma}$
$\lambda = 1$	n = 10	0.859	0.456	1.689	0.86	3.427	1.692	5.063	2.103
	n = 50	0.445	0.479	0.905	0.948	1.978	1.889	2.656	2.36
	n = 100	0.315	0.481	0.693	0.957	1.331	1.912	1.736	2.405
	n = 500	0.151	0.484	0.313	0.966	0.689	1.932	0.933	2.423
$\lambda = n/2$	n = 10	0.858	0.459	1.682	0.866	3.306	1.705	4.301	2.125
	n = 50	0.444	0.48	0.893	0.95	1.869	1.896	2.424	2.369
	n = 100	0.314	0.482	0.629	0.959	1.251	1.915	1.625	2.417
	n = 500	0.153	0.484	0.309	0.969	0.654	1.933	0.846	2.423
$\lambda = n$	n = 10	0.858	0.461	1.677	0.87	3.244	1.717	4.072	2.137
	n = 50	0.443	0.481	0.887	0.951	1.815	1.898	2.312	2.373
	n = 100	0.313	0.482	0.625	0.959	1.212	1.917	1.568	2.418
	n = 500	0.153	0.484	0.308	0.967	0.638	1.933	0.815	2.423

Table 5.1: Simulation results for the posterior mode estimation of $\boldsymbol{\mu}$ and σ^2.

5.4.2 Simulation for the posterior mode of $\boldsymbol{M}$

In this part we test the effectiveness of our method in estimating the posterior mode of $\boldsymbol{M}$ and also compare the Posterior with the Likelihood approach. We use the same simulated data from Section 4.2 which include 1000 samples of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ of 25 and 30 landmarks respectively. The first 20 landmarks from each $\boldsymbol{X}_i$ are considered as the matched and the remaining as the unmatched. We only use the posterior density of (5.3.1) with the uniform prior on $\boldsymbol{M}$. The prior mean $\boldsymbol{\mu}_0$ was selected as the true mean, for the prior of σ^2 we use $\alpha = 5$ and $\beta = 10$ and set $\lambda = 1$.

Figure 5.1 displays comparison between the Likelihood and the Posterior alignment methods for correctly identifying each landmark either as a matched (landmarks 1-20) or unmatched (landmarks 21-25). For $\sigma = 0.1$ or, $\sigma = 0.5$ the Posterior and the Likelihood Alignment perform similarly having above 95% success rate for correctly identifying each landmark either as matched or unmatched.

When $\sigma = 1$ the two methods start to differ, for the first 20 landmarks both have similar success rates above 90%. However, the Posterior Alignment seems to perform better in terms of correctly identifying which landmarks should be left unmatched, (landmarks 21-25). This behaviour becomes even more clear when the ratio of $\sigma/d_{min} \geq 1$. In that case when $\sigma = 2$ the success rate of identifying the unmatched landmarks is 72% for the Posterior Alignment versus 45% for the Likelihood Alignment and 61% versus 35% when $\sigma = 2.5$.

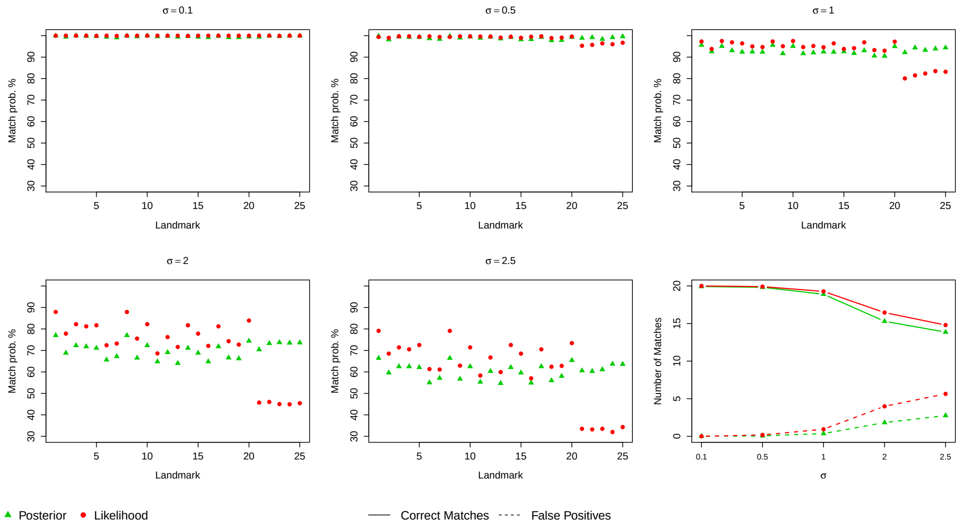


Figure 5.1: Comparison of between Posterior and Likelihood alignment of the mean proportions for each landmark to be identified successfully either as a 'correct' match or as 'unmatched' landmark. The last plot presents the number of correct and false positive matches for each of the first 20 landmarks.

Figure 5.2 displays the histograms of correct and false positive matches for each of the first 20 landmarks using the Posterior Alignment method. Comparing this Figure with that of 4.2 we see that the distribution for the correct matches does not differ a lot between the two methods, although the mean number of correct matches seems to be slightly higher for the Likelihood Alignment. The opposite happens in the case of false positives, especially when $\sigma = 2$ or $\sigma = 2.5$ there is a clear difference in the two distributions with the Posterior Alignment having on average 1.8 false positive matches compared to 3.9 of the Likelihood Alignment when $\sigma = 2$ and 2.7 compared to 5.5 when $\sigma = 2.5$.

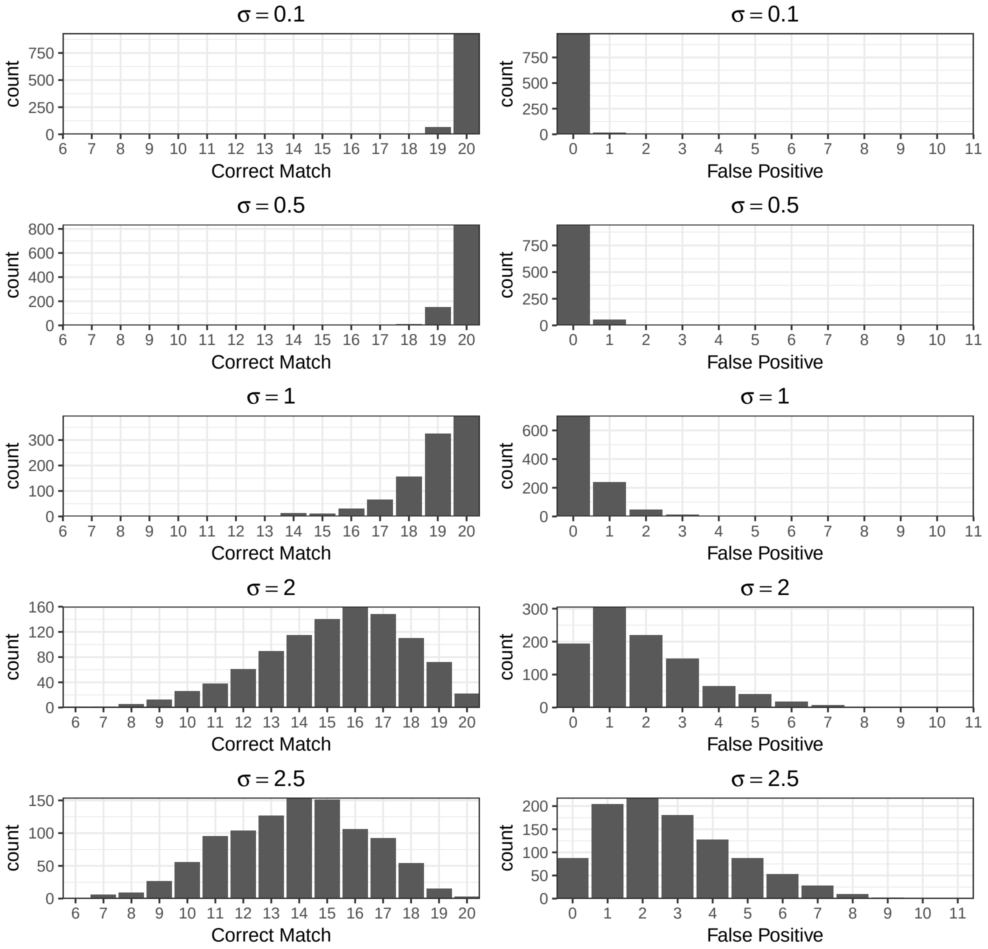


Figure 5.2: Distribution of correct matches and false positives for different values of σ.

5.4.3 Conclusions

- From the simulation results regarding the posterior mode estimation of $\boldsymbol{\mu}$ and σ^2 our approach seems to perform well for estimating the correct posterior mode for all λ values.

- From the simulation results regarding the posterior mode estimation of $\boldsymbol{M}$, our Posterior Alignment algorithm seems to performs better in identifying which landmarks should be left unmatched compared to that of the Likelihood Alignment especially when the ratio of $\sigma/d_{min} \geq 1$

- The proportion of identifying correct matches is similar between the Likelihood and the Posterior Alignment methods, but when $\sigma = 2$ or $\sigma = 2.5$ the Likelihood Alignment seems to have a slightly higher success rate.

5.5 Protein data

In this Section we evaluate the performance of the Posterior Alignment using two different datasets. The first dataset is that of 16 protein pairs from Ortiz et al. (2002) and the second dataset consists of 64 protein pairs from the HOMSTRAD database which are difficult to align due to low sequence similarity. These two datasets have also been used in Chapter 4 for testing the Likelihood Alignment method.

The solution from the likelihood alignment was used as the prior mean $\boldsymbol{\mu}_{0_{(M)}}$, α_0 was set to 5 and β_0 to 10, giving a prior mean of 2.5 for σ^2. Finally three different values for λ were used as $1, 10$ and 50. In order to compare the results the following similarity metrics were used:

- Number of matched atoms (M)
- Root Mean Square Distance (2.8.1)
- TMscore (2.8.2)
- Structure Overlap (2.8.3)

The same metrics have also been used in Chapter 4 for the testing of the Likelihood Alignment method.

Table 5.2 displays the results of the Posterior Alignment method using a uniform and a gap prior for the data of Ortiz et al. (2002). The choice of the prior seems to have an effect of the final results. Almost in all of the pairs the use of a uniform prior results in alignments with more matched atoms and a higher RMSD, whereas the gap prior suggests alignments with fewer atoms and closer matched together. This performance is somehow expected since the gap prior penalises matches that do not follow the sequence order, making more difficult for a new match to be accepted. In comparison with Table 4.3 we can see that the in most of the cases, the Posterior Alignment had more matched atoms with less RMSD compared to the method of *RS2014*. Also, we should note that both methods fail to produce a good alignment for the pair of *1plc-1acx*. The uniform prior has a solution of 87 matched atoms but with an RMSD of 7.2Å and the gap prior a solution of only 15 matched atoms and an RMSD of 5.9Å. This might be due to the choice of the prior mean and the starting points, having as a result our algorithm to get stuck in a local mode and not allowing to remove the *bad* matches. Another reason might be that this specific pair has regions with very different structure and since we perform a global alignment, we will also align regions with no structure similarity resulting to an increased RMSD value.

Protein1	Protein2	Post. Align (Unif)		Post. Align (Gap)	
		RMSD	M	RMSD	M
1aba	1dsbA	2.0	60	2.0	60
1aba	1trs	2.3	70	2.2	67
1acx	1cobB	2.4	95	1.3	52
1acx	1rbe	2.1	30	2.1	31
1mjc	5tssA	1.6	57	1.6	57
1pgb	5tssA	1.5	38	1.1	29
1plc	**1acx**	**7.2**	**87**	**5.9**	**15**
1ptsA	1mup	2.2	80	1.5	54
1tnfA	1bmvI	2.6	112	2.4	107
1ubq	1frd	2.2	65	1.7	52
1ubq	4fxc	2.4	69	2.3	68
2gb1	1ubq	1.7	42	1.7	42
2gb1	4fxc	1.8	42	1.8	42
2rslC	3chy	3.0	93	2.4	75
2tmvP	256bA	2.2	83	2.2	83
3chy	1rcf	2.8	125	2.5	116

Table 5.2: Posterior Alignment with a uniform and a gap prior for the data of Ortiz et al. (2002).

Table 5.3 displays the summary of different metrics from various alignment methods. The choice of λ seems to have little effect on the final alignment for both prior choices. When $\lambda = 10$ or $\lambda = 50$ the corresponding alignments have about 2 to 3 less matched atoms on average and an RMSD of about the same rate. In general most of the alignment methods perform fairly similarly, with a similar number of matched atoms, RMSD, TMscore and Structure Overlap. The two methods that seem to differ are the Posterior Alignment with the Uniform prior and the Likelihood approach. They have a higher number of matched atoms compared to the others and also a very good TMscore of 0.55 and 0.6 respectively.

Method	RMSD	M	TMscore	SO (%)
Posterior Alignment (Unif, $\lambda = 1$)	2.50	72	0.55	65.67
Posterior Alignment (Unif, $\lambda = 10$)	2.89	68	0.53	65.21
Posterior Alignment (Unif, $\lambda = 50$)	3.00	69	0.54	66.16
Posterior Alignment (Gap, $\lambda = 1$)	2.17	59	0.49	61.20
Posterior Alignment (Gap, $\lambda = 10$)	2.69	60	0.49	61.12
Posterior Alignment (Gap, $\lambda = 50$)	2.69	59	0.48	60.80
Likelihood Alignment	2.74	82	0.60	70.30
LGA	2.41	57	0.50	61.67
DALI	3.02	69	0.50	58.85
TMalign	3.06	73	0.52	-
RS2014 ($\lambda = 7.6$)	2.64	54	-	-
RS2014 ($\lambda = 8.6$)	3.48	71	-	-
RS2014 ($\lambda = 9.6$)	3.91	77	-	-

Table 5.3: Summary of structural alignment by different methods for the data of Ortiz et al. (2002)

However, in such a small sample size of 16 protein pairs the results of the Posterior Alignment will be highly affected by the *outlier* of 1plc-1acx. Hence, in Table 5.4 we display the same metric results but without taking into account the pair of 1plc-1acx. Now, we can see more clearly the small effect of the λ choice. Furthermore, the Posterior Alignment with a uniform prior although has about 10 less matched atoms than the Likelihood Alignment methods, it has better results in terms of RMSD (2.18Å to 2.68Å) and Structure Overlap (70.42% to 69.51%), meaning that it tends to produce solutions with fewer matched atoms but much closer aligned together.

Method	RMSD	M	TMscore	SO (%)
Posterior Alignment (Unif, $\lambda = 1$)	2.18	71	0.57	70.42
Posterior Alignment (Unif, $\lambda = 10$)	2.18	71	0.57	70.49
Posterior Alignment (Unif, $\lambda = 50$)	2.27	73	0.57	70.57
Posterior Alignment (Gap, $\lambda = 1$)	1.92	62	0.51	64.85
Posterior Alignment (Gap, $\lambda = 10$)	1.93	62	0.51	64.76
Posterior Alignment (Gap, $\lambda = 50$)	1.93	61	0.51	63.39
Likelihood Alignment	2.68	81	0.59	69.51
LGA	2.39	56	0.50	61.71
DALI	3.03	69	0.50	57.55
TMalign	3.04	73	0.51	-
RS2014 ($\lambda = 7.6$)	2.61	52	-	-
RS2014 ($\lambda = 8.6$)	3.35	70	-	-
RS2014 ($\lambda = 9.6$)	3.51	75	-	-

Table 5.4: Summary of structural alignment by different methods for the data of Ortiz et al. (2002) without the pair of 1plc-1acx.

We now test the Posterior Alignment method on the second dataset of 64 protein pairs from the HOMSTRAD database. These pairs present a challenging case for alignment since they have a low structure similarity ranging from 30% to 70%. The Posterior Alignment method now suggests solutions with the lowest RMSD among the other methods (1.70Å and 1.88Å), however the number of matched atoms is smaller compared to the Likelihood approach. Furthermore, the Posterior Alignment method with a uniform prior performs similar to the method of SPalignNS (Brown et al., 2015) having similar RMSD, number of matched atoms and TMscore. Finally, in comparison to the rest of the Bioinformatics algorithms, although the Posterior Alignment approach has fewer matched atoms it has better Structure Overlap, suggesting that a higher proportion of the matched atoms are aligned with a distance smaller than 3.5Å.

Algorithm	RMSD	M	SO (%)	TMscore
Posterior Alignment (Unif)	1.88	71	68.38	0.527
Posterior Alignment (Gap)	1.70	62	59.68	0.453
Likelihood Alignment	2.22	81	69.64	0.531
TMalign	2.95	84	67.71	0.493
SPalignNS	1.91	72	72.83	0.527
SPalign	2.66	81	69.27	-
CLICK	1.96	67	68.90	-
FlexSnap	2.23	66	61.37	-
MICAN	2.91	82	61.30	-
HOMSTRAD	3.15	81	59.40	-
SALIGN	2.02	-	67.20	-
DALI	2.00	-	63.00	-
GANGSTA	1.99	-	61.90	-
Geometric Hashing	1.91	-	59.50	-
FATCAT	2.36	-	59.10	-

Table 5.5: Structural alignments by different algorithms for the difficult to align 64 pairs from the HOMSTRAD database.

In this last part, we explore a particular pair of *transferases* which consists of the protein 1gky with 186 atoms and protein 2ak3 with 226 atoms. The same pair has also been analysed by Rodriguez and Schmidler (2014) and Fallaize et al. (2014) so we can compare our results with these two methods.

Figure 5.3 presents the atom correspondence of the alignment solutions from the Likelihood and Posterior approaches. The Posterior approach finds an alignment with 150 matched atoms and RMSD of 2.3Å compared to the 167 and RMSD of 2.8Å from the Likelihood method. Also, we can see that most of the matched pairs are common between the two methods (blue colour). Furthermore, we notice that since the likelihood solution was used as the prior mean for the Posterior Alignment, the latter removed the matched pairs which have been probably mismatched. For example the pair of atoms (26 - 218), (68 - 101) or (186 - 108). Also, Fallaize et al. (2014) for the same pair reports a solution with 131 matched atoms and RMSD of 2.25Å, whereas Rodriguez and Schmidler (2014) reports two solutions depending whether the amino acid information is used one with RMSD of 3.5Å and one with 1.95Å. In comparison to our method we have at least similar alignments. In particular, compared to the solution of Fallaize et al. (2014) we managed to match 36 more atoms at an increase of only 0.05Å in the RMSD value.

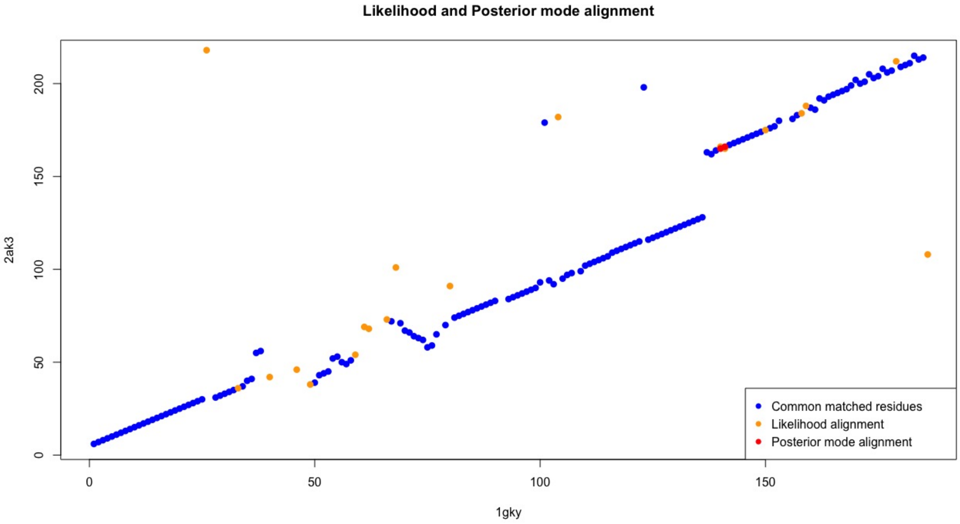


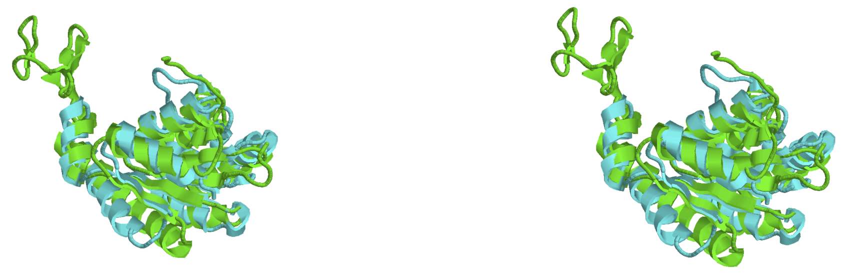
Figure 5.3: Atom correspondence of the solution from the Posterior and Likelihood alignments for the pair 1gky-2ak3. The blue colour indicates the pairs of atoms which have been matched by both methods. The red colour indicates the pair of atoms which have been matched only by the Posterior method and the yellow color the pairs which matched only from the Likelihood method.

Finally, Figure 5.4 illustrates the full atom structure of the two molecules 1gky, 2ak3 and Figure 5.5 the full atom alignment using the Posterior and Likelihood methods. Both structures have a high number of α - *helices* with 7 in 1gky and 17 in 2ak3, representing most of the structure body for both molecules. In Figure 5.5 we see that both methods aligned the 7 helices of 1gky. The six out of seven helices seem to have been closely matched in both cases with only the bottom left helix having a slightly bigger distance.

(a) 1gky

(b) 2ak3

Figure 5.4: Protein pair 1gky - 2ak3



(a) Posterior Alignment

(b) Likelihood Alignment

Figure 5.5: Protein pair 1gky - 2ak3

5.5.1 Conclusions

- From the examples presented in Tables 5.2, 5.3 and 5.5 the value of λ has very little effect on the final solutions.

- The Posterior Alignment method produces solutions with less matched atoms compared to the likelihood approach, however it has better RMSD and Structure Overlap scores, meaning that the matched atoms are closer together.

- On average the Posterior Alignment provides better solutions (more matched atoms, less RMSD) compared to the method of *RS2014* (Table 5.4)

- The prior choice has an effect on the final solution. The uniform prior produces alignments with more matched atoms, whereas the gap prior tends to alignments with less matched atoms and smaller RMSD.

5.6 Discussion

The Posterior Alignment approach presented in this Chapter is an alternative method to that used for aligning protein molecules. Due to the nature of the problem, sampling the full posterior distribution of the match matrix is often unnecessary, since at the end a one-to-one correspondence for each atom is needed for evaluating the final solution. Methods in the Bayesian literature make use of linear optimization techniques to derive the final correspondence from the posterior distribution of $\boldsymbol{M}$. Our approach avoids this step and tries to directly estimate the posterior mode of $\boldsymbol{M}$.

Chapter 6

Generalized matching model

6.1 Introduction

In this Chapter we extend the likelihood methodology of matching protein molecules presented in Chapter 3. We now consider a more general framework assuming a Normal distribution for both the matched and unmatched parts of a protein molecule. We propose two different approaches, one is shown in Section 6.2 in which two independent Normal distributions are considered for the matched and unmatched parts of a molecule. In particular, we consider different variances for each part, while the mean of the unmatched part is fixed to 0. The other approach is shown in Section 6.3, where now we consider as one entry and do not separate it any more into matched and unmatched parts and a diagonal covariance matrix is considered with only two different entries. Section 6.4 is about the alignment algorithm for these two approaches, which we use to obtain the final matching. It is based on **Algorithm 1** with a small addition of the Generalized EM algorithm. Finally, in Section 6.5, we test our two modelling approaches using both simulated and real data.

6.2 Normal density for the unmatched parts

In this Section we present a different parametrization for the matching model described in Chapters 2 and 3. So far, most of the statistical approaches that have been used for the protein alignment problem, including the ones presented in the previous Chapters, involve a likelihood that has two terms. The Normal density for the matched and a Uniform density for the unmatched parts. Here, we present a new modelling approach where each part of the molecule is now following a Normal distribution with a different mean and variance.

Consider two protein molecules represented by the configuration matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ with dimensions $3 \times k$ and $3 \times l$ respectively. As we discussed in the previous Chapters each $\boldsymbol{X}_i$ is observed under some similarity transformations (3.8.1). Then by using the *Singular Value Decomposition* of (3.2.3) each $\boldsymbol{X}_i$ will represent the observed size and shape variables $\boldsymbol{\Delta}_i \boldsymbol{O}_i$ (Kendall et al., 2009). This process is similar to the one described in all the previous Chapters. To remove the location effect from the observed $\boldsymbol{X}_i$ we apply the *Helmertized* transformation of (3.2.6) independently in the matched and unmatched parts of each molecule.

By using the *Helmertized* landmarks we bring the centre of both $\boldsymbol{X}_i^M$ and $\boldsymbol{X}_i^{-M}$ to 0. Thus, we consider the following Normal distributions for each of the parts

$$\begin{aligned}(\boldsymbol{X}_1^M, \boldsymbol{X}_2^M) &\sim \mathcal{N}(\boldsymbol{\mu}^M, \sigma^2)\\ (\boldsymbol{X}_1^{-M}, \boldsymbol{X}_2^{-M}) &\sim \mathcal{N}(\boldsymbol{0}, \sigma_0^2)\end{aligned} \tag{6.2.1}$$

Fixing the mean for the unmatched parts to 0, allow us to eliminate the rotation effect for these parts of the molecules. We also expect the variance σ_0^2 of the unmatched part to be higher than the variance of the matched part σ^2, since each unmatched landmark would be further away than its mean compared to the matched ones.

Finally, in order to remove the rotation effect, we use the same approach as before by integrating the rotation parameter out of the likelihood. We can derive the *size and shape* densities of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as follows

$$
\begin{aligned}
f_{FM}(\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{M}, \boldsymbol{\mu}^M, \sigma^2, \sigma_0^2) &= \prod_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} f_N(\boldsymbol{X}_i^{*^M}|\boldsymbol{M}, \boldsymbol{\mu}^M, \sigma^2) d\boldsymbol{R}_i f_{N_0}(\boldsymbol{X}_i^{*^{-M}}|\boldsymbol{M}, \boldsymbol{0}, \sigma_0^2) \\
= \prod_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} (2\pi\sigma^2)^{-3p} \exp\left\{-\frac{||\boldsymbol{R}_i\boldsymbol{X}_i^M - \boldsymbol{\mu}^M||^2}{2\sigma^2}\right\} & (2\pi\sigma_0^2)^{-3(k-p)(l-p)/2} d\boldsymbol{R}_i \exp\left\{-\frac{||\boldsymbol{X}_i^{-M} - 0||^2}{2\sigma_0^2}\right\} \\
= (2\pi\sigma^2)^{3p}(2\pi\sigma_0^2)^{-3(k-p)(l-p)/2} \exp & \left\{-\frac{\sum\limits_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 + 2||\boldsymbol{\mu}^M||^2}{2\sigma^2} - \frac{\sum\limits_{i=1}^{2} ||\boldsymbol{X}_i^{-M}||^2}{2\sigma_0^2}\right\} \\
& \times \prod_{i=1}^{2} \int\limits_{\boldsymbol{R}_i \in SO(3)} \exp\left\{\frac{\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i^M\boldsymbol{\mu}^{M^t}\right)}{\sigma^2}\right\} \quad (6.2.2)
\end{aligned}
$$

where $\boldsymbol{X}_i^*$ represents the full unobserved data.

6.2.1 Parameter optimization & EM steps

Optimizing over the unknown parameters of $\boldsymbol{\mu}^M$,σ^2 and σ_0^2 does not significantly differ from the procedure used in Chapter 3. Again rotations $\boldsymbol{R}_i$ are treated as an unobserved part of the data and the EM algorithm is used for inference. The *Expectation step* will be the same as the one described in Section 3.3 since by using a fixed 0-mean for the unmatched parts the rotation is only present in the matched part of the likelihood (6.2.2). The *Maximization step* remains the same as before for both $\boldsymbol{\mu}^M$ and σ^2, leading to the following estimates

$$
\hat{\boldsymbol{\mu}}^M = \frac{1}{2}\sum_{i=1}^{2} \frac{\int\limits_{\boldsymbol{R}_i\boldsymbol{X}_i^M \in SO(3)} R_i \exp\left\{\frac{\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i^M\boldsymbol{\mu}^{M^t}\right)}{\sigma^2}\right\} d\boldsymbol{R}_i}{\int\limits_{\boldsymbol{R}_i \in SO(3)} \exp\left\{\frac{\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i^M\boldsymbol{\mu}^{M^t}\right)}{\sigma^2}\right\}} \tag{6.2.3}
$$

$$
\hat{\sigma^2} = \frac{1}{6p}\left(\sum_{i=1}^{2} ||\boldsymbol{X}_i^M||^2 - ||\hat{\boldsymbol{\mu}}^M||^2\right) \tag{6.2.4}
$$

Similarly for the unmatched parts we need to estimate only the sample variance σ_0^2 since the unmatched mean is assumed to be fixed to 0.

$$\hat{\sigma}_0^2 = \frac{\sum_{i=1}^{2} ||\boldsymbol{X}_i^{-M}||^2}{3(k-p)(l-p)} \tag{6.2.5}$$

6.3 Diagonal covariance matrix

Here, in this Section we extend the modelling framework presented before. We still keep the Normality assumption, but now we consider $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ to be observations from a common Normal distribution as

$$\text{vec}(\boldsymbol{X}_1, \boldsymbol{X}_2) \sim N(\text{vec}(\boldsymbol{\mu}), \Sigma) \tag{6.3.1}$$

where vec($\boldsymbol{\mu}$) has dimensions of $1 \times 3(k+l)$ and Σ is block diagonal so that $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are independent and has dimensions of $(k+l) \times (k+l)$. Then, considering also the partition into matched and unmatched parts we can write vec($\boldsymbol{\mu}$) as

$$\text{vec}(\boldsymbol{\mu}) = [\text{vec}(\mu_1), \text{vec}(\mu_2)] \tag{6.3.2}$$

where

$$\text{vec}(\boldsymbol{\mu}_1) = \left[\overbrace{\mu^M, \ldots, \mu^M}^{(p)-\text{times}}, \overbrace{\mu^{-M}, \ldots, \mu^{-M}}^{(k-p)-\text{times}} \right] \quad \text{vec}(\boldsymbol{\mu}_2) = \left[\overbrace{\mu^M, \ldots, \mu^M}^{(p)-\text{times}}, \overbrace{\mu^{-M}, \ldots, \mu^{-M}}^{(l-p)-\text{times}} \right]$$

Furthermore, the joint covariance matrix Σ can be partitioned as

$$\Sigma = \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array} \right] \tag{6.3.3}$$

where Σ_i is a diagonal covariance matrix for each $\boldsymbol{X}_i$ as

$$\Sigma_1 = s_1 \mathcal{I}_k, \qquad \Sigma_2 = s_2 \mathcal{I}_l \tag{6.3.4}$$

with

$$s_1 = \left[\overbrace{\sigma^2,\ldots,\sigma^2}^{(p)-\text{times}},\overbrace{\sigma_0^2,\ldots,\sigma_0^2}^{(k-p)-\text{times}}\right] \qquad s_2 = \left[\overbrace{\sigma^2,\ldots,\sigma^2}^{(p)-\text{times}},\overbrace{\sigma_0^2,\ldots,\sigma_0^2}^{(l-p)-\text{times}}\right]$$

Hence, the distributional assumptions for $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be written as

$$\begin{aligned}\text{vec}(\boldsymbol{X}_1) &\sim N(\text{vec}(\boldsymbol{\mu}_1), \Sigma_1)\\ \text{vec}(\boldsymbol{X}_2) &\sim N(\text{vec}(\boldsymbol{\mu}_2), \Sigma_2) \qquad (6.3.5)\end{aligned}$$

where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are independent and the matched and unmatched parts have variance σ^2 and σ^2 respectively.

Finally, as we have mentioned in the previous Section and Chapters by using the Singular Value Decomposition of (3.2.3) each $\boldsymbol{X}_i$ represent the observed size and shape variables $\boldsymbol{\Delta}_i\boldsymbol{O}_i$ under the similarity transformations of (3.8.1). So far, for removing the location parameter τ_i we used the *Helmertized* landmarks by multiplying each configuration matrix $\boldsymbol{X}_i$ by the *Helmert* matrix $\boldsymbol{H}$ of (3.2.6). Now, instead of using the resulting *Helmertized* landmarks we choose to multiply each $\boldsymbol{X}_i$ with a matrix $\boldsymbol{L}_i$. Hence, each of the transformed covariance matrices Σ_i will be in the form of $\boldsymbol{L_1}\Sigma_1\boldsymbol{L_1}^t$ and $\boldsymbol{L_2}\Sigma_2\boldsymbol{L_2}^t$ and since that each Σ_i is diagonal with only two different elements σ^2 and σ_0^2 the final transformed covariance matrices will be as follows

$$\begin{aligned}\Sigma_1^* &= \boldsymbol{L_1}\Sigma_1\boldsymbol{L_1}^t = \Sigma_{1_{k-1}} + \sigma^2\mathcal{I}_{k-1}\\ \Sigma_2^* &= \boldsymbol{L_2}\Sigma_2\boldsymbol{L_2}^t = \Sigma_{2_{l-1}} + \sigma^2\mathcal{I}_{l-1} \qquad (6.3.6)\end{aligned}$$

where, $\boldsymbol{L}_1 = (-1_{k-1}, \mathcal{I}_{k-1})$ and $\boldsymbol{L}_2 = (-1_{l-1}, \mathcal{I}_{l-1})$. For simplicity, in the rest of this Chapter the use of Σ_i will mean the covariance matrices after they have been multiplied by $\boldsymbol{L}_i$ as in (6.3.6).

Again to induce the size and shape densities of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ the rotation parameter $\boldsymbol{R}_i$ is considered as unobserved and integrated out of the likelihood. Thus, we have

$$
\begin{aligned}
f_D(\boldsymbol{X}_1, \boldsymbol{X}_2 | \boldsymbol{M}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2) &= \prod_{i=1}^{2} \int_{\boldsymbol{R}_i \in SO(3)} f_N(\boldsymbol{X}_i^* | \boldsymbol{M}, \boldsymbol{\mu}_i, \Sigma_i) d\boldsymbol{R}_i \\
= (2\pi|\Sigma_1|)^{-\frac{3k}{2}} (2\pi|\Sigma_2|)^{-\frac{3l}{2}} & \prod_{i=1}^{2} \int_{\boldsymbol{R}_i \in SO(3)} \exp\left\{ -\frac{(\boldsymbol{R}_i\boldsymbol{X}_i - \boldsymbol{\mu}_i)^t \Sigma_i^{-1} (\boldsymbol{R}_i\boldsymbol{X}_i - \boldsymbol{\mu}_i)}{2} \right\} d\boldsymbol{R}_i \\
= (2\pi)^{-\frac{3(k+l)}{2}} (\sigma^2)^{-\frac{3(k+l)p}{2}} (\sigma_0^2)^{-\frac{3}{2}(k^2+l^2+(k+l)p)} & \exp\left\{ -\frac{1}{2} \sum_{i=1}^{2} \mathrm{tr}\left(\boldsymbol{X}_i \boldsymbol{X}_i^t \Sigma_i^{-1} + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^t \Sigma_i^{-1} \right) \right\} \\
& \times \prod_{i=1}^{2} \int_{\boldsymbol{R}_i \in SO(3)} \exp\left\{ \mathrm{tr}\left(\boldsymbol{R}_i \boldsymbol{X}_i \boldsymbol{\mu}_i^t \Sigma_i^{-1} \right) \right\} d\boldsymbol{R}_i \quad (6.3.7)
\end{aligned}
$$

The rationale of choosing this particular modelling approach was to treat each protein molecule as one entry, instead of partitioning in two parts in distributional sense as has been done in most of the approaches so far. For example, to estimate now the mean we optimally rotate the whole molecule instead of only rotating the part which corresponds to the matched atoms. This can provide a more natural representation as the parts of the protein structure do not act independently.

6.3.1 Parameter optimization & GEM steps

Before we move onto the estimation of the match matrix $\boldsymbol{M}$ which will give us the optimal alignment between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, we need to optimize over the remaining unknown parameters of $\boldsymbol{\mu}_i$ and Σ_i. For this, we make use of the *Generalized EM* algorithm (Dempster et al., 1977) which is a variation of the EM algorithm described in Section 2.4. Now, during the Maximization step we do not seek to maximize over the unknown parameters but obtain some other value which increases the total likelihood function. We make use of the GEM because it is not possible to jointly maximize both $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and Σ_1, Σ_2 due to the difference in dimensionality for each parameter.

Since each $\boldsymbol{X}_i$ is independent, the missing rotations $\boldsymbol{R}_i$ will also be independent and the *Expectation* step of the GEM at the t-th iteration will be as follows

$$Q(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2 | \boldsymbol{\mu}_{1_{t-1}}, \boldsymbol{\mu}_{2_{t-1}}, \Sigma_{1_{t-1}}, \Sigma_{2_{t-1}}) = \mathbb{E}_{\boldsymbol{R}_i|\boldsymbol{X}_i}\left[\log f_D(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}_{i_{t-1}}, \Sigma_{i_{t-1}})\right]$$
$$= -\left(\frac{3(k+l)}{2}\right)\log(2\pi) - \left(\frac{3(k+l)p}{2}\right)\log(\sigma^2_{t-1}) - \left(\frac{3}{2}(k^2+l^2+p(k+l))\right)\log(\sigma^2_{0_{t-1}})$$
$$-\frac{1}{2}\sum_{i=1}^{2}\operatorname{tr}\left(\boldsymbol{X}_i\boldsymbol{X}_i^t\Sigma^{-1}_{i_{t-1}} + \boldsymbol{\mu}_{i_{t-1}}\boldsymbol{\mu}^t_{i_{t-1}}\Sigma^{-1}_{i_{t-1}}\right) + \sum_{i=1}^{2}\log\int\limits_{\boldsymbol{R}_i\in SO(3)}\exp\left\{\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i\boldsymbol{\mu}^t_{i_{t-1}}\Sigma^{-1}_{i_{t-1}}\right)\right\}d\boldsymbol{R}_i \tag{6.3.8}$$

The *Maximization step* now differs from the standard EM approach. First, we try to optimize for the mean parameter $\boldsymbol{\mu}_i$. As explained earlier, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are a combination of the matched and unmatched means, such that they share the same p elements that correspond to the matched atoms of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Hence, by differentiating (6.3.8) over $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and setting equal to 0 we obtain the following estimates for each $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ as

$$\frac{\partial Q(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2 | \boldsymbol{\mu}_{1_{t-1}}, \boldsymbol{\mu}_{2_{t-1}}, \Sigma_{1_{t-1}}, \Sigma_{2_{t-1}})}{\partial \boldsymbol{\mu}_1} = 0 \quad \Rightarrow \quad \boldsymbol{\mu}_1 = \frac{\int\limits_{\boldsymbol{R}_1\in SO(3)} \boldsymbol{R}_1\boldsymbol{X}_1 e^{A_1} d\boldsymbol{R}_1}{\int\limits_{\boldsymbol{R}_1\in SO(3)} e^{A_1} d\boldsymbol{R}_1} \tag{6.3.9}$$

$$\frac{\partial Q(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2 | \boldsymbol{\mu}_{1_{t-1}}, \boldsymbol{\mu}_{2_{t-1}}, \Sigma_{1_{t-1}}, \Sigma_{2_{t-1}})}{\partial \boldsymbol{\mu}_2} = 0 \quad \Rightarrow \quad \boldsymbol{\mu}_2 = \frac{\int\limits_{\boldsymbol{R}_2\in SO(3)} \boldsymbol{R}_2\boldsymbol{X}_2 e^{A_2} d\boldsymbol{R}_2}{\int\limits_{\boldsymbol{R}_2\in SO(3)} e^{A_2} d\boldsymbol{R}_2} \tag{6.3.10}$$

with $A_i = -\operatorname{tr}\left(\boldsymbol{R}_i\boldsymbol{X}_i\boldsymbol{\mu}_i^t\Sigma_i^{-1}\right)$. Thus, the estimate of the common matched mean for $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ at time t can be obtained as

$$\hat{\boldsymbol{\mu}}_t^M = \frac{1}{2}\sum_{i=1}^{2}\frac{\displaystyle\int\limits_{\boldsymbol{R}_i\in SO(3)} \boldsymbol{R}_i\boldsymbol{X}_i^M e^{A_i} d\boldsymbol{R}_i}{\displaystyle\int\limits_{\boldsymbol{R}_i\in SO(3)} e^{A_i} d\boldsymbol{R}_i} \tag{6.3.11}$$

and the common unmatched mean of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be obtained as

$$\hat{\boldsymbol{\mu}}_t^{-M} = \frac{1}{k+l-2p}\left[\sum_{j=p+1}^{k}\frac{\displaystyle\int\limits_{\boldsymbol{R}_1\in SO(3)} \boldsymbol{R}_1\boldsymbol{X}_{1_j}^{-M} e^{A_1} d\boldsymbol{R}_1}{\displaystyle\int\limits_{\boldsymbol{R}_1\in SO(3)} e^{A_1} d\boldsymbol{R}_1} + \sum_{j=p+1}^{l}\frac{\displaystyle\int\limits_{\boldsymbol{R}_2\in SO(3)} \boldsymbol{R}_2\boldsymbol{X}_{2_j}^{-M} e^{A_2} d\boldsymbol{R}_2}{\displaystyle\int\limits_{\boldsymbol{R}_2\in SO(3)} e^{A_2} d\boldsymbol{R}_2}\right] \tag{6.3.12}$$

On the other hand, obtaining the estimates for σ^2 and σ_0^2 is not as straightforward, because a closed form expression from (6.3.8) is not easily derived. Instead, we choose to optimize numerically (6.3.8) over σ^2 and σ_0^2 considering $\hat{\boldsymbol{\mu}}_t^M$ and $\hat{\boldsymbol{\mu}}_t^{-M}$ as fixed.

Several optimization techniques are available in the literature. For our purpose we make use of the *optim* function in R which among others include the NelderMead method (Nelder and Mead, 1965) and the BFGS, a quasi Newton optimization algorithm(Fletcher, 2013). Adopting the numerical optimization approach for σ^2 and σ_0^2 increases the speed of the algorithm while the overall estimates remain accurate and allow us to apply constraints on the two parameters ensuring that the variance of the matched parts is always smaller than that of the unmatched(i.e $\sigma^2 < \sigma_0^2$.

In summary, the steps for obtaining the estimates of $\boldsymbol{\mu}_i$ and Σ_i for a given matching matrix $\boldsymbol{M}$ using the GEM algorithm are as follows

Algorithm 5 GEM for obtaining estimates of $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \sigma^2, \sigma_0^2$

```
Input $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{M}, \epsilon, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \sigma^2, \sigma_0^2$.
$t \leftarrow 1$.
while $\log f_D(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}_{i_t}, \Sigma_{i_t}) - \log f_D(\boldsymbol{X}_i|\boldsymbol{M}, \boldsymbol{\mu}_{i_{t-1}}, \Sigma_{i_{t-1}}) > \epsilon$ do
    Expectation - step : Evaluate $Q(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2|\boldsymbol{\mu}_{1_{t-1}}, \boldsymbol{\mu}_{2_{t-1}}, \Sigma_{1_{t-1}}, \Sigma_{2_{t-1}})$
    Maximization - step :
        • Obtain $\hat{\boldsymbol{\mu}}_t^M, \hat{\boldsymbol{\mu}}_t^{-M}$
        • For the updated values of $\hat{\boldsymbol{\mu}}_t^M, \hat{\boldsymbol{\mu}}_t^{-M}$, optimize numerically to obtain
          $\hat{\sigma}_t^2, \hat{\sigma}_{0_t}^2$
end while
```

where ϵ represents the convergence criterion.

6.4 Alignment algorithm

In order to obtain an alignment between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ we use the same optimization approach for the matching matrix $\boldsymbol{M}$ as the one described in **Algorithm 1** of Section 3.4. A few adjustments should now be made as follows:

- Before we start exploring possible matches between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ we should remove the location information by creating $\boldsymbol{L}_1, \boldsymbol{L}_2$ and obtain $\boldsymbol{L}_1\boldsymbol{X}_1, \boldsymbol{L}_2\boldsymbol{X}_2$ and Σ_1^*, Σ_2^* as in (6.3.6).
- The optimization for $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and σ^2, σ_0^2 should be done using the GEM asdescribed in **Algorithm 5**.
- The likelihood density $f_D(\cdot)$ of (6.3.7) should be used.

Finally, we note that if sequence or gap penalty information is included in the model, then the likelihood of (6.3.7) can be easily extended to incorporate them using the same ideas described in Sections 3.5 and 3.6.

6.5 Simulations

In this Section we compare the two different models presented using simulated and real data. In order to generate the simulated data we used a similar process to that described in Section 4.2 and is based on Kenobi and Dryden (2012).

Inside a cube of volume L^3 we create a mean shape of 21 landmarks subject to the constrain that each landmark has at leas a minimum distance d_{min} with all the others. The matched part of the data coming from the first 20 landmarks of $\boldsymbol{\mu}$ are n observations from a Normal distribution as

$$(\boldsymbol{X}^M_{1_{[1:20]}}, \boldsymbol{X}^M_{2_{[1:20]}}) \sim N(\boldsymbol{\mu}_{[1:20]}, \sigma^2)$$

and for the unmatched parts coming from the 21st landmark of $\boldsymbol{\mu}$ are n Normal observations as follows

$$\boldsymbol{X}^{-M}_{1_{[21:25]}} \sim N(\boldsymbol{\mu}_{[21]}, \sigma_0^2) \qquad \boldsymbol{X}^{-M}_{2_{[21:30]}} \sim N(\boldsymbol{\mu}_{[21]}, \sigma_0^2)$$

The parameters used for this simulation are the following:

- $L = 20, d_{min} = 2, n = 1000$
- $\sigma = \{0.1, 0.5, 1, 2, 3\}$
- $\sigma_0 = 5$

6.5.1 Simulation results

Figure 6.1 displays the simulation results for the simulated data created before using the two different models described in Sections 6.2 and 6.3. For simplicity we call the model of Section 6.2 with the unmatched mean fixed to zero using the likelihood (6.2.2) as the *Fixed Mean* model and the model from Section 6.3 with the diagonal covariance structure as the *Diagonal* model which is based on the likelihood of (6.3.7).

As we can see from Figure 6.1 for $\sigma = 0.1$ both models perform similarly in terms of finding which landmarks should be matched (1 to 20) and which should be left unmatched (21 to 25). When $\sigma = 0.5$ the probability of identifying the matched landmarks remains high for both models (approximately of 97%) but the probability of correctly finding the unmatched landmarks drops to 80%.

This behaviour becomes more present for higher values of σ. In particular when the ratio $\sigma/d_{min} \geq 1$, both models seem to fail into identifying the unmatched landmarks. For example, when $\sigma = 2$ the *Fixed Mean* model has a 23% chance of identifying the unmatched landmarks compared to a 17% chance for the *Diagonal* model.

In the case of $\sigma = 3$ we observe that for both approaches more false positives matches are identified and the correct matching percentages are dropped, which is something we expect since the minimum distance between each mean is less than our σ value and it is not so clear which landmark belongs to which mean. However, both models perform relatively well in finding the correct match for each of the first 20 landmarks, where the *Diagonal* models has a 68% chance of success and the *Fixed mean* a 62%.

Finally, we report an interesting pattern between the matched (1 to 20) and unmatched (21 to 25) landmarks. The *Diagonal* model preforms always better in finding the correct match for the first 20 landmarks but the *Fixed mean* model performs better in identifying the unmatched landmarks (last 5).

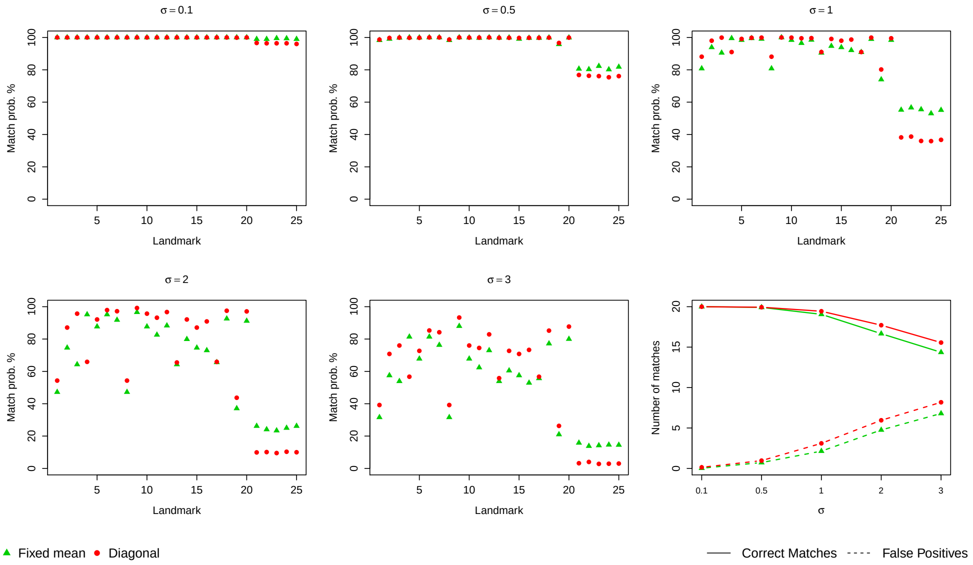


Figure 6.1: Comparison between the Fixed mean and Diagonal model using simulated data. The first 5 plots present the mean proportions for each landmark to be identified either as a 'correct' match or as 'unmatched' landmark. The last plot presents the number of correct and false positive matches for each of the first 20 landmarks.

Figures 6.2 and 6.3 display the histograms of the number of correct and false positives matches for the two different models using various values of σ for the 1000 samples of simulated data in created before. For small σ's (0.1, 0.5) the distribution of correct and false positives matches for both models is concentrated around a few values indicating that we can successfully identify which landmarks match with each other. On the other hand, when $\sigma = 2$ or $\sigma = 3$ these distributions change and become more skewed. For the correct matches we observe a negative skewness and for the false positives a positive skewness which is something good since it shows that the average correct and false positive matches tend to the desired values of 20 and 0 respectively. By comparing the two methods we also see that the false positive distribution for the *Fixed Mean* model seems to have a larger skewness than the *Diagonal* model meaning that on average it produces less false positives especially when $\sigma = 2$ or $\sigma = 3$. The opposite seems to happen for the correct match distribution when these two are compared with the diagonal model having more correct matches in this situation.

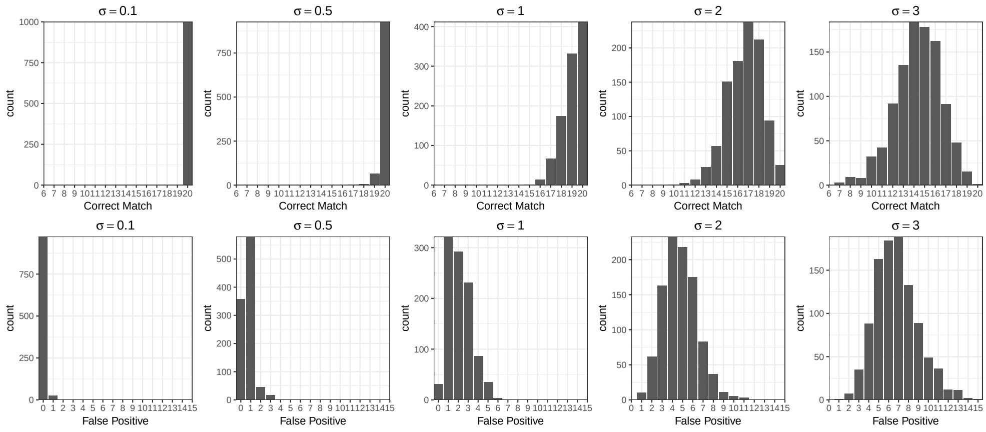


Figure 6.2: Distribution of correct matches and false positives for different values of σ for the Fixed Mean model.

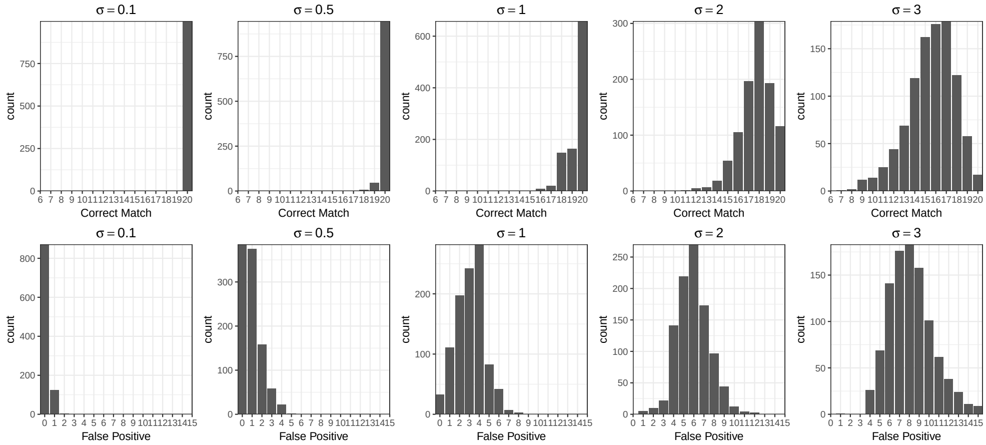


Figure 6.3: Distribution of correct matches and false positives for different values of σ for the Diagonal model.

Moreover, in Table 6.1 we see the results for the mean estimates of correct matches and false positives as the final estimations for the standard deviation of the matched parts σ and the unmatched σ_0. For $\sigma \leq 1$ both methods perform similarly well, matching on average more than 19 landmarks out of 20 and the number of false positives remains low with a maximum of about 2 or 3. The *Diagonal* model seems to perform a little better in finding slightly more correct matches and the *Fixed Mean* model does a little better in finding less false positives. However, when $\sigma \geq 2$ the number of correct matches drops and the false positive increases, which is something we anticipate since the landmarks are mixing. In the example of $\sigma = 3$ the *Fixed Mean* model has on average 14.36 correct matches compared to the 15.56 of the *Diagonal* model and the opposite happens in the false positive matches with the *Diagonal* model having about 1.3 more matched landmarks.

Estimating the value σ and σ_0 is not easy since we only have two observations, however the *Fixed Mean* model performs relatively well. For example, for values of $\sigma = 0.1, 0.5, 1$ the corresponding estimates are close to the real values and also estimates of σ_0 are close to the true value of 5. When σ becomes bigger, the *Fixed Mean* model tends to underestimate σ and to overestimate σ_0. This is probably because of the mix up between the landmarks since the σ/d_{min} ratio is ≥ 1. In comparison, the *Diagonal* model seems to not perform well in the variance estimations for both the matched and unmatched parts producing very small values in both cases. One of the reasons that this might happen is because we estimate σ and σ_0 using the whole set of landmarks and not treat these estimations separately. Nevertheless, this is a behaviour that needs further exploring in future work.

	Fixed mean				Diagonal covariance			
	CM	FP	σ	σ_0	CM	FP	σ	σ_0
$\sigma = 0.1$	19.99	0.02	0.07	4.99	19.99	0.13	0.001	0.86
$\sigma = 0.5$	19.92	0.72	0.36	5.12	19.93	0.96	0.01	0.90
$\sigma = 1$	19.05	2.13	0.70	5.82	19.43	3.09	0.02	0.99
$\sigma = 2$	16.66	4.75	1.27	7.78	17.71	5.95	0.07	1.50
$\sigma = 3$	14.36	6.79	1.76	9.04	15.56	8.17	0.13	2.22

Table 6.1: Mean estimates for the number of correctly matched and false positive landmarks as long as σ estimates for the fixed mean and diagonal covariance models.

6.5.2 Conclusions

- Both models perform similarly for small values of σ, having a good success rate of finding which landmarks should be matched and which should be left unmatched.

- For high values of σ's there is a drop in the correct matches and an increase in false positives.

- The *Fixed Mean* model had consistently a better chance of identifying the unmatched landmarks, whereas the *Diagonal* model had a higher probability of finding the correct match for each landmark.

- The *Fixed Mean* model performs outperforms the *Diagonal* model estimating σ and σ_0.

- The *Diagonal* model tends to significantly underestimate σ and σ_0.

6.6 Protein data application

In this Section, we test the two models of Section 6.2 and 6.3 on two pairs of protein data. The first consists of the *cytochrome b5*, a membrane bound hemoprotein, usually found in animals and plants called *1aqa* with 82 atoms and the *cytochrome b5 ascaris suum*, a protein found in parasitic worm called *1x3x* with 84 atoms. The structures for both molecules are shown in Figure 6.4.

The second pair is that of the *hemogoblin 4hhbD* which is an iron-oxygen binding protein found in the human red cells with 146 atoms and the *myogoblin 1mbo* an iron-oxygen binding protein found in the muscle tissue of animals with 153 atoms . Both of these structures are shown in Figure 6.5.

(a) 1aqa (b) 1x3x

Figure 6.4: Protein molecules 1aqa and 1x3x.

(a) 4hhbD (b) 1mbo

Figure 6.5: Protein molecules 4hhbD and 1mbo.

Figure 6.6 displays the atom correspondence between 1aqa and 1x3x using the *Fixed Mean* model from (6.2.2) and the *Diagonal* model from (6.3.7). As we can see, both methods find almost the same alignment solution with the *Fixed*

Mean model having 10 more matched atoms which have not been identified by the *Diagonal* model.

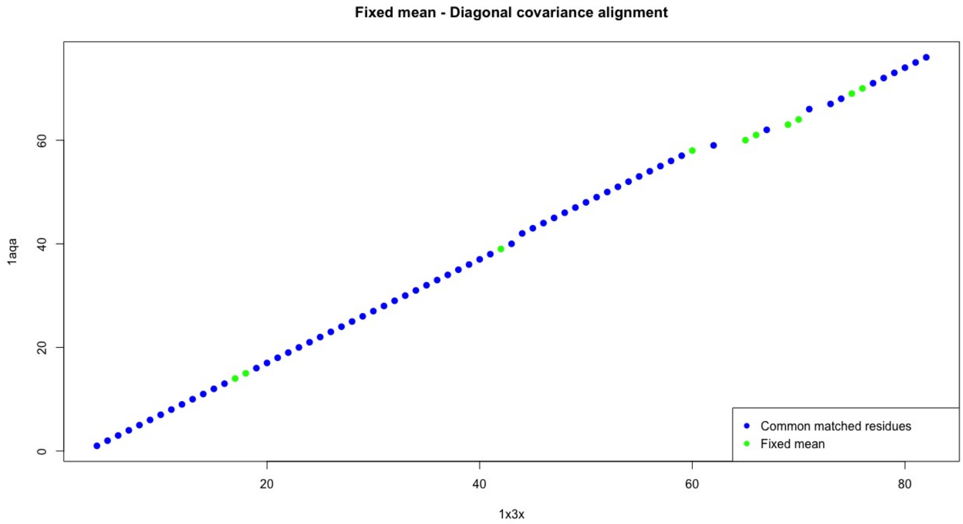


Figure 6.6: Alignment of the pair 1aqa - 1x3x using the fixed mean and diagonal covariance model.

Table 6.2 presents the similarity metrics for these two methods compared also with the Likelihood Alignment approach from Chapter 3. The *Fixed Mean* model and the Likelihood Alignment seem to perform very similarly. The TMscore is higher for the *Fixed Mean* model 0.79 compared to 0.77 of the Likelihood method as is the Structure Overlap, 90.24% compared to 89.02%. Although there seem to be some differences in the alignments between the *Fixed Mean* and the *Diagonal* models, both of them managed to have TMscores above 0.5 indicating that the two proteins might belong to the same fold. The only category in which the *Diagonal* model performs better is the RMSD value. This is expected since it has 10 less matched atoms. Finally, the σ estimations for the two models of this Chapter are quite different, following the same pattern as in the simulated results where the *Diagonal* model underestimates the variance.

	Fixed Mean	Diagonal	Likelihood Align.
M	74	64	73
RMSD	1.3	1.0	1.4
σ	0.37	0.001	0.39
σ_0	6.89	0.74	-
TMscore	0.79	0.71	0.77
SO (%)	90.24	78.05	89.02

Table 6.2: Similarity metrics for the pair of 1aqa - 1x3x using the Fixed Mean, Diagonal and Likelihood alignment methods.

Figure 6.7 displays the full atom alignment for the two molecules after they have been optimally rotated. Again we can see that the two solutions are very similar with the only difference in the matching of the loop in the lower right corner. Although the sequence similarity for these two molecules is quite low (of about 17%), which does not indicate that the two proteins are related, we can see that based on the structure alignment, they match quite well especially in their secondary structures where all the *α-helices* have been closely aligned.

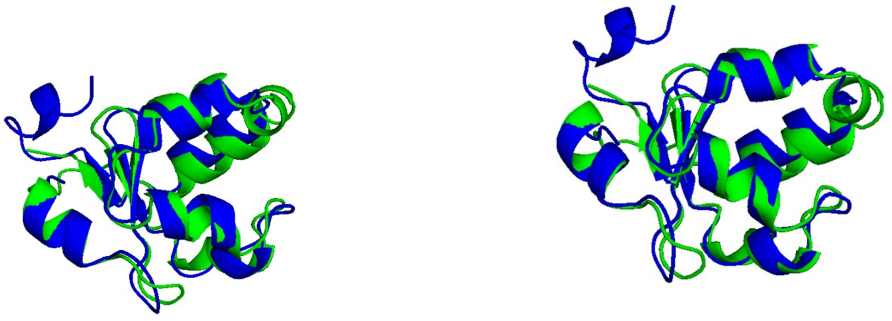

(a) Fixed Mean alignment (b) Diagonal alignment

Figure 6.7: Alignment solutions of 1aqa - 1x3x using the fixed mean and diagonal covariance models.

Our second example is the pair 4hhbD - 1mbo. Figure 6.8 displays the atom correspondence using the two different models. Again both methods find almost the same solution with most of the matched atoms being the same between them. The difference is in the atom pairs of $(83 - 83), (136 - 137), (145 - 146), (139 - 140), (142 - 143)$ which have been matched only by the *Fixed Mean* model and the pair $(120 - 121)$ which is matched only by the *Diagonal* model.

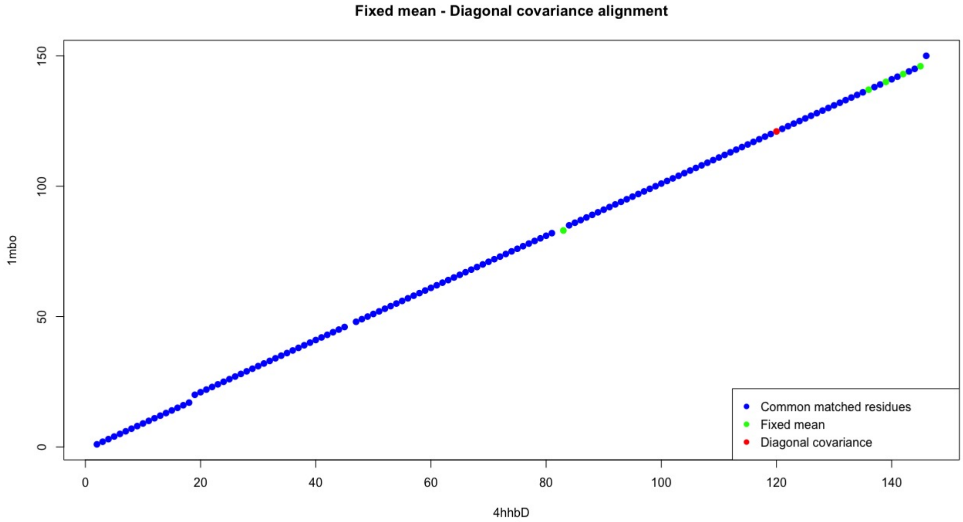


Figure 6.8: Alignment of the pair 4hhbD - 1mbo using the Fixed Mean and Diagonal model.

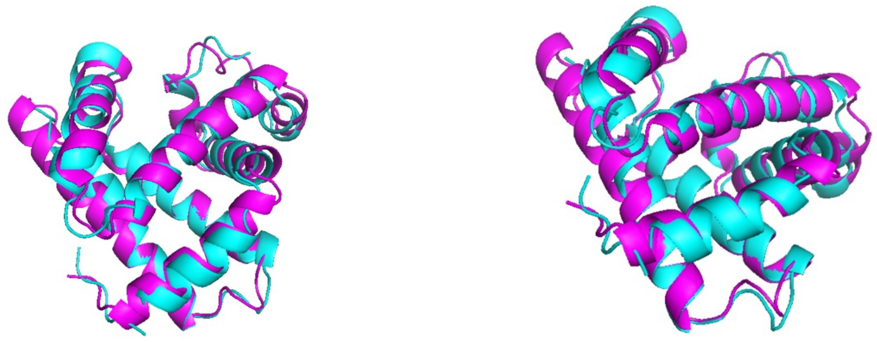
Table 6.3 displays the similarity metrics of the 4hbD - 1mbo alignment using the two models of this Chapter and the Likelihood Alignment method. Again all these three models have performed similarly, with almost identical alignment solutions.

The *Fixed Mean* model has 4 matched atoms more than the *Diagonal* but the RMSD value of 1.4Å is the same for both of them. All three have very similar TMscores and Structure Overlap . In particular the Structure Overlap is at least 95% for all methods, meaning that 95% of the matched atoms are within a distance of 3.5Å. The only significant difference we can observe between the *Fixed Mean* and the *Diagonal* model is in the estimation σ, a behaviour we also observed in the previous example and during our simulation tests. Overall the two alignments are very similar and the different estimation of σ by the *Diagonal* model does not seem to have a significant effect in the final solution.

	Fixed Mean	Diagonal	Likelihood Align
M	142	138	143
RMSD	1.4	1.4	1.5
σ	0.42	0.001	0.42
σ_0	10.93	0.39	-
TMscore	0.87	0.85	0.89
SO (%)	97.26	94.52	97.95

Table 6.3: Similarity metrics for the pair of 4hhbD - 1mbo after the alignment with the fixed mean and diagonal covariance models.

Finally, Figure 6.9 displays the full structure alignment of the two protein molecules. As the previous results suggest both alignment solutions are very similar and the two structures seem to align very well despite the not so high sequence identity which is at about 25%. Most of the secondary structure of the two proteins have been aligned really well, except the N terminus of 4hhbD which is in the upper left corner.



(a) Fixed mean alignment

(b) Diagonal covariance alignment

Figure 6.9: Alignment solutions of 4hhbD - 1x3x using the fixed mean and diagonal covariance models.

6.7 Discussion

In this Chapter we presented a different approach in modelling the protein alignment problem. We introduced a more general approach considering a Normal distribution for both the matched and unmatched parts of the molecules. Furthermore, we allowed different variances among these two parts in order to distinguish the matched and unmatched atoms. Finally, we considered two different approaches one with a fixed zero mean for the unmatched part and one for a mean estimated by the data.

The simulation results suggested that both methods work well in identifying the correct matches. Some difference was observed in identifying which atoms should be left unmatched, where the *Fixed Mean* model performed better than the *Diagonal* model. The same situation is observed with the estimations σ and σ_0, where the *Diagonal* model tends to underestimate both of them.

However, when we tested our models in real data the difference in the estimation of σ between the two models did not have a big effect in the final alignments, providing almost the same results with both the *Fixed Mean* and the Likelihood Alignment model.

Chapter 7

Discussion & future work

7.1 Summary and conclusions

The aim of this study was to explore the problem of protein structure alignment from a statistical point of view. So far, there have been two approaches in the structural alignment literature. One includes an adhoc algorithmic approach, which although is fast and easy to implement lacks an overall modelling framework and the other one is a Bayesian approach which provides this modelling framework but sometimes is not straightforward to implement. In this Thesis, we developed techniques that bridge this gap, borrowing elements from both approaches.

In Chapter 3 we introduced a likelihood based approach for providing a *score* between a given alignment of two or more molecules. It is based on a symmetric size and shape likelihood and the EM algorithm for estimating the unknown parameters. This likelihood density is our core model and the different extensions presented are based on this. Furthermore, we introduced a Structural Alignment algorithm for estimating a possible alignment between two or more protein molecules and an extension of it which also considers the sequence order of the amino acid chain.

As the results suggested our best performing method is the one that includes only the structural information. It seems that most of the times provides solutions which combine more matched atoms with less RMSD compared to other alternative algorithms either from Bioinformatics or current statistical models. In addition, almost in all the examples explored our TMscore scores were higher than all of the other approaches. The extra information in the likelihood (sequence or gap penalty) seems to make not much difference in the final results, suggesting that our solutions are mostly based on the information provided by the structure. However, it can be used when solutions with more *biological* meaning are needed, for example preserving the amino acid sequence order.

In Chapter 5 we explore the same problem using Bayesian modelling approach. Recent methods in the Bayesian literature estimate the posterior distribution of the match matrix and then use optimization algorithms to derive its posterior mode in order to produce a final one-to-one alignment. In our approach we try to estimate directly the posterior mode of the match matrix. Another difference with our approach is that we choose to assign a prior distribution over the mean matrix instead of treating it as a fixed parameter or integrating it out of the likelihood.

From the simulation and real data results the posterior alignment approach seemed to be better in identifying which atoms do not have a corresponding match. Also, the choice of the uniform prior on the match matrix seemed to provide better results in terms of both more matched atoms and lower total RMSD compared to the gap prior.

Finally, in Chapter 6 we presented a different approach for the protein matching. We considered a Normal distribution for both matched and unmatched parts while we allow different variances between the two parts of the molecule. This approach provides a more natural interpretation since the whole molecule is rotated instead of only the matched part.

7.2 Future work

One of the main difficulties we encountered during this study was the selection of starting points. Since our derivation for the optimal alignment between two molecules is based on a discrete optimization algorithm a good starting point is required. In Chapter 3 we present an algorithm for automatic selection of a set of starting points but further work should be done in this area. A possible direction could be a ranking matching system among the atoms, selecting those with the highest score for starting points or the exploration of different combinations of number of starting points and selecting the one with the highest likelihood.

Another area for future work is the multiple matching of proteins. Due to the design of exploring all possible combination of atoms our method has a limitation on the number of proteins that can be simultaneously aligned. A different approach with a possibility of selecting a subset of all the combinations should be considered.

Finally in the last part of the Thesis we introduced a diagonal covariance matrix for the size and shape likelihood of the two molecules. This approach although is working well in terms of matching two proteins it fails to estimate the correct variance of the mode.l This is an issue that also needs further exploring. A final extension of this model would be to consider allowing general covariance among the atoms of each protein.

Bibliography

Aalberse, R. C. Structural biology of allergens. *Journal of allergy and clinical immunology*, 106(2):228–238, 2000.

Andreeva, A., Howorth, D., Brenner, S. E., Hubbard, T. J., Chothia, C., and Murzin, A. G. Scop database in 2004: refinements integrate structure and sequence family data. *Nucleic acids research*, 32(suppl 1):D226–D229, 2004.

Bachar, O., Fischer, D., Nussinov, R., and Wolfson, H. A computer vision based technique for 3-d sequence-independent structural comparison of proteins. *Protein Engineering*, 6(3):279–287, 1993.

Berman, H. M., Battistuz, T., Bhat, T., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., et al. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

Bishop, M. and Thompson, E. A. Maximum likelihood alignment of dna sequences. *Journal of molecular biology*, 190(2):159–165, 1986.

Braberg, H., Webb, B. M., Tjioe, E., Pieper, U., Sali, A., and Madhusudhan, M. S. Salign: a web server for alignment of multiple protein sequences and structures. *Bioinformatics*, 28(15):2072–2073, 2012.

Brown, P., Pullan, W., Yang, Y., and Zhou, Y. Fast and accurate non-sequential protein structure alignment using a new asymmetric linear sum assignment heuristic. *Bioinformatics*, page btv580, 2015.

Carugo, O. Recent progress in measuring structural similarity between proteins. *Current protein and peptide science*, 8(3):219–241, 2007.

Challis, C. J. and Schmidler, S. C. A stochastic evolutionary model for protein structure alignment and phylogeny. *Molecular biology and evolution*, 29(11): 3575–3587, 2012.

Chothia, C. and Lesk, A. M. The relation between the divergence of sequence and structure in proteins. *The EMBO journal*, 5(4):823, 1986.

Chui, H. and Rangarajan, A. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.

Coats, E. A. The comfa steroids as a benchmark dataset for development of 3d qsar methods. In *3D QSAR in drug design*, pages 199–213. Springer, 1998.

Cramer, R. D., Patterson, D. E., and Bunce, J. D. Comparative molecular field analysis (comfa). 1. effect of shape on binding of steroids to carrier proteins. *Journal of the American Chemical Society*, 110(18):5959–5967, 1988.

Dawson, N. L., Lewis, T. E., Das, S., Lees, J. G., Lee, D., Ashford, P., Orengo, C. A., and Sillitoe, I. Cath: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research*, 45(D1):D289–D295, 2017.

Dayhoff, M. O. and Schwartz, R. M. A model of evolutionary change in proteins. In *In Atlas of protein sequence and structure*. Citeseer, 1978.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Diaz-Garcia, J. A., Jaimez, R. G., and Mardia, K. V. Wishart and pseudo-wishart distributions and some applications to shape theory. *Journal of Multivariate Analysis*, 63(1):73–87, 1997.

Downs, T. D. Orientation statistics. *Biometrika*, 59(3):665–676, 1972.

Dror, O., Benyamini, H., Nussinov, R., and Wolfson, H. Mass: multiple structural alignment by secondary structures. *Bioinformatics*, 19(suppl 1):i95–i104, 2003a.

Dror, O., Benyamini, H., Nussinov, R., and Wolfson, H. J. Multiple structural alignment by secondary structures: algorithm and applications. *Protein Science*, 12(11):2492–2507, 2003b.

Dryden, I., Kume, A., and Wood, A. Mle in size and shape space. In *LASR 2015 Geometry-Driven Statistics and its Cutting Edge Applications: Celebrating Four Decades of Leeds Statistics Workshops*, 2015.

Dryden, I. L. and Mardia, K. V. *Statistical shape analysis*, volume 4. Wiley Chichester, 1998.

Dryden, I. L., Hirst, J. D., and Melville, J. L. Statistical analysis of unlabeled point sets: comparing molecules in chemoinformatics. *Biometrics*, 63(1):237–251, 2007.

Edgar, R. C. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.

Fallaize, C., Green, P., Mardia, K., and Barber, S. Bayesian protein sequence and structure alignment. *arXiv preprint arXiv:1404.1556*, 2014.

Fallaize, C. J. and Kypraios, T. Exact bayesian inference for the bingham distribution. *Statistics and Computing*, 26(1-2):349–360, 2016.

Fletcher, R. *Practical methods of optimization*. John Wiley & Sons, 2013.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.

Gerstein, M. and Levitt, M. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7(2):445–456, 1998.

Godzik, A. The structural alignment between two proteins: Is there a unique answer? *Protein science*, 5(7):1325–1338, 1996.

Godzik, A., Jambon, M., and Friedberg, I. Computational protein function prediction: are we making progress? *Cellular and molecular life sciences*, 64(19): 2505–2511, 2007.

Goodall, C. and Mardia, K. V. The noncentral bartlett decompositions and shape densities. *Journal of Multivariate Analysis*, 40(1):94–108, 1992.

Green, P. J. and Mardia, K. V. Bayesian alignment using hierarchical models, with applications in protein bioinformatics. *Biometrika*, 93(2):235–254, 2006.

Greene, L. H., Lewis, T. E., Addou, S., Cuff, A., Dallman, T., Dibley, M., Redfern, O., Pearl, F., Nambudiry, R., Reid, A., et al. The cath domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic acids research*, 35(suppl_1):D291–D297, 2006.

Gu, J. and Bourne, P. E. *Structural bioinformatics*, volume 44. John Wiley & Sons, 2009.

Guerler, A. and Knapp, E.-W. Novel protein folds and their nonsequential structural analogs. *Protein Science*, 17(8):1374–1382, 2008.

Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

Holm, L. and Sander, C. Protein structure comparison by alignment of distance matrices. *Journal of molecular biology*, 233(1):123–138, 1993.

Holm, L. and Sander, C. Dali/fssp classification of three-dimensional protein folds. *Nucleic acids research*, 25(1):231–234, 1997.

Kendall, D. G., Barden, D., Carne, T. K., and Le, H. *Shape and shape theory*, volume 500. John Wiley & Sons, 2009.

Kenobi, K. and Dryden, I. L. Bayesian matching of unlabeled point sets using procrustes and configuration models. *Bayesian Analysis*, 7(3):547–566, 2012.

Kent, J. T., Mardia, K. V., and Taylor, C. C. Matching problems for unlabelled configurations. *Bioinformatics, Images, and Wavelets*, pages 33–36, 2004.

Khatri, C. and Mardia, K. The von mises-fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 95–106, 1977.

Kihara, D. and Skolnick, J. The pdb is a covering set of small protein structures. *Journal of molecular biology*, 334(4):793–802, 2003.

Koehl, P. Protein structure similarities. *Current opinion in structural biology*, 11 (3):348–353, 2001.

Koehl, P. and Levitt, M. Sequence variations within protein families are linearly related to structural variations. *Journal of molecular biology*, 323(3):551–562, 2002.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Kume, A. and Wood, A. T. Saddlepoint approximations for the bingham and fisher–bingham normalising constants. *Biometrika*, 92(2):465–476, 2005.

Lathrop, R. H. The protein threading problem with sequence amino acid interaction preferences is np-complete. *Protein Engineering, Design and Selection*, 7(9):1059–1068, 1994.

Levitt, M. and Gerstein, M. A unified statistical framework for sequence comparison and structure comparison. *Proceedings of the National Academy of sciences*, 95(11):5913–5920, 1998.

Mardia, K. V. and Jupp, P. E. *Directional statistics*, volume 494. John Wiley & Sons, 2009.

Mardia, K. V., Nyirongo, V. B., Fallaize, C. J., Barber, S., and Jackson, R. M. Hierarchical bayesian modeling of pharmacophores in bioinformatics. *Biometrics*, 67(2):611–619, 2011.

Mardia, K. V., Petty, E. M., and Taylor, C. C. Matching markers and unlabeled configurations in protein gels. *The Annals of Applied Statistics*, pages 853–869, 2012.

Mardia, K. V., Fallaize, C. J., Barber, S., Jackson, R. M., and Theobald, D. L. Bayesian alignment of similarity shapes. *The annals of applied statistics*, 7(2): 989, 2013.

Minami, S., Sawada, K., and Chikenji, G. Mican: a protein structure alignment algorithm that can handle multiple-chains, inverse alignments, c α only models, alternative alignments, and non-sequential alignments. *BMC bioinformatics*, 14(1):24, 2013.

Mizuguchi, K., Deane, C. M., Blundell, T. L., and Overington, J. P. Homstrad: a database of protein structure alignments for homologous families. *Protein science*, 7(11):2469–2471, 1998.

Moran, P. Quaternions, haar measure and the estimation of a paleomagnetic rotation. *Perspectives in probability and statistics*, pages 295–301, 1975.

Muirhead, R. J. *Aspects of multivariate statistical theory*, volume 197. John Wiley & Sons, 2009.

Munkres, J. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

Needleman, S. B. and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

Nelder, J. A. and Mead, R. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

Nguyen, M., Tan, K. P., and Madhusudhan, M. S. Clicktopology-independent comparison of biomolecular 3d structures. *Nucleic acids research*, 39(suppl 2): W24–W28, 2011.

Orengo, C. A., Michie, A., Jones, S., Jones, D. T., Swindells, M., and Thornton, J. M. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Ortiz, A. R., Strauss, C. E., and Olmea, O. Mammoth (matching molecular models obtained from theory): an automated method for model comparison. *Protein Science*, 11(11):2606–2621, 2002.

Pauling, L., Corey, R. B., and Branson, H. R. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951.

Prentice, M. J. A distribution-free method of interval estimation for unsigned directional data. *Biometrika*, pages 147–154, 1984.

Raffenetti, R. C. and Ruedenberg, K. Parametrization of an orthogonal matrix in terms of generalized eulerian angles. *International Journal of Quantum Chemistry*, 4(S3B):625–634, 1969.

Rangarajan, A., Chui, H., and Bookstein, F. L. The softassign procrustes matching algorithm. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 29–42. Springer, 1997.

Rodriguez, A. and Schmidler, S. C. Bayesian protein structure alignment. *The Annals of Applied Statistics*, 8(4):2068–2095, 2014.

Rossmann, M. G. and Argos, P. The taxonomy of binding sites in proteins. *Molecular and cellular biochemistry*, 21(3):161–182, 1978.

Rost, B. Protein structures sustain evolutionary drift. *Folding and Design*, 2: S19–S24, 1997.

Rost, B. Twilight zone of protein sequence alignments. *Protein engineering*, 12 (2):85–94, 1999.

Ruffieux, Y. and Green, P. J. Alignment of multiple configurations using hierarchical models. *Journal of Computational and Graphical Statistics*, 18(3): 756–773, 2009.

Salem, S., Zaki, M. J., and Bystroff, C. Flexsnap: flexible non-sequential protein structure alignment. *Algorithms for Molecular Biology*, 5(1):12, 2010.

Schmidler, S. C. Fast bayesian shape matching using geometric algorithms. *Bayesian statistics*, 8:471–490, 2007.

Sei, T. and Kume, A. Calculating the normalising constant of the bingham distribution on the sphere using the holonomic gradient method. *Statistics and Computing*, 25(2):321–332, 2015.

Sei, T., Takayama, N., Takemura, A., Nakayama, H., Nishiyama, K., Noro, M., and Ohara, K. Holonomic gradient descent and its application to fisher-bingham integral. *arXiv preprint arXiv:1005.5273*, 2010.

Shindyalov, I. N. and Bourne, P. E. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein engineering*, 11(9): 739–747, 1998.

Smith, T. F. and Waterman, M. S. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

Taylor, C. C., Mardia, K. V., and Kent, J. T. Matching unlabelled configurations using the em algorithm. *Proceedings in Stochastic Geometry, Biological Structure and Images*, pages 19–21, 2003.

Wood, A. T. Estimation of the concentration parameters on the fisher matrix distribution on so(3) and the bingham distribution on $s_q, q \geq 2$. *Australian Journal of Statistics*, 35:69–79, 1993.

Wood, T. C. and Pearson, W. R. Evolution of protein sequences and structures. *Journal of molecular biology*, 291(4):977–995, 1999.

Xu, J. and Zhang, Y. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

Yang, Y., Zhan, J., Zhao, H., and Zhou, Y. A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. *Proteins: Structure, Function, and Bioinformatics*, 80(8):2080–2088, 2012.

Ye, Y. and Godzik, A. Fatcat: a web server for flexible structure comparison and structure similarity searching. *Nucleic acids research*, 32(suppl 2):W582–W585, 2004.

Zemla, A. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

Zhang, Y. and Skolnick, J. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

Zhou, G.-P. An intriguing controversy over protein structural class prediction. *Journal of protein chemistry*, 17(8):729–738, 1998.