Mixed sequence-structure based analysis of proteins, with applications to functional annotations

Romain Tetley

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Analyse mixte de protéines basée sur la séquence et la structure – applications à l'annotation fonctionnelle

Romain Tetley

ABS, Inria Sophia Antipolis Méditerranée

Présentée en vue de l'obtention du grade de docteur en Informatique d'Université Côte d'Azur
Dirigée par: Frédéric Cazals
Soutenue le: 21/11/2018

Devant le jury, composé de:
J. Cortés, Directeur de Recherches, CNRS
J-F. Gibrat, Directeur de Recherches, INRA
M. Weigt, Professeur, UPMC
F. Rey, Directeur de Recherches, CNRS
D. Mazauric, Chargé de Recherches, Inria
F. Cazals, Directeur de Recherches, Inria

Analyse mixte de protéines basée sur la séquence et la structure – applications à l'annotation fonctionnelle

Jury:

Directeur

F. Cazals, Directeur de Recherches, Inria

Rapporteurs

J. Cortés, Directeur de Recherches, CNRS
J-F. Gibrat, Directeur de Recherches, INRA

Examinateurs

M. Weigt, Professeur, UPMC
F. Rey, Directeur de Recherches, CNRS

Invités

D. Mazauric, Chargé de Recherches, Inria

Mixed sequence-structure based analysis of proteins, with application to functional annotations

In this thesis, the focus is set on reconciling the realms of structure and sequence for protein analysis. Sequence analysis tools shine when faced with proteins presenting high sequence identity ($\geq 30\%$), but are lack-luster when it comes to remote homolog detection. Structural analysis tools present an interesting alternative, but solving structures –when at all possible– is a tedious and expensive process. These observations make the need for hybrid methods – which inject information obtained from available structures in a sequence model – quite clear.

This thesis makes four main contributions toward this goal. First we present a novel structural measure, the $\mathrm{RMSD}_{\mathrm{Comb.}}$, based on local structural conservation patterns – the so called *structural motifs*.
Second, we developed a method to identify structural motifs between two structures using a bootstrap method which relies on *filtrations*. Our approach is not a direct competitor to flexible aligners but can provide useful to perform a multiscale analysis of structural similarities.
Third, we build upon the previous methods to design *hybrid Hidden Markov Models* which are biased towards regions of increased structural conservation between sets of proteins. We test this tool on the class II fusion viral proteins – particularly challenging because of their low sequence identity and mild structural homology. We find that we are able to recover known remote homologs of the viral proteins in the Drosophila and other organisms.
Finally, formalizing a sub-problem encountered when comparing *filtrations*, we present a new theoretical problem – the D-family matching – on which we present various algorithmic results. We show –in a manner that is analogous to comparing parts of two protein conformations– how it is possible to compare two clusterings of the same data set using such a theoretical model.

Keywords: Structural biology; Structural alignment; Topology; Persistence; Structural motifs; Hidden Markov Models; Protein annotation; Graph theory; Clustering

Analyse mixte de protéines basée sur la séquence et la structure – applications à l'annotation fonctionnelle

Dans cette thèse, l'emphase est mise sur la réconciliation de l'analyse de structure et de séquence pour les protéines.
L'analyse de séquence brille lorsqu'il s'agit de comparer des protéines présentant une forte identité de séquence ($\geq$ 30%) mais laisse à désirer pour identifier des homologues lointains. L'analyse de structure est une alternative intéressante. Cependant, les méthodes de résolution de structures sont coûteuses et complexes –lorsque toutefois elles produisent des résultats. Ces observations rendent évident la nécessité de développer des méthodes hybrides, exploitant l'information extraite des structures disponibles pour l'injecter dans des modèles de séquence.

Cette thése produit quatre contributions principales dans ce domaine. Premièrement, nous présentons une nouvelle distance structurale, le $\mathrm{RMSD}_{\mathrm{Comb.}}$, basée sur des patterns de conservation structurale locale, les *motifs structuraux*.
Deuxièmement, nous avons dévelopé une méthode pour identifier des motifs structuraux entre deux structures exploitant une méthode de bootstrap dépendant de *filtrations*. Notre approche n'est pas un compétiteur direct des aligneurs fléxibles mais permet plutôt de produire des analyses multi-échelles de similarités structurales.
Troisièmement, nous exploitons les méthodes sus-cités pour construire des *modéles de Markov cachés hybrides* biaisés vers des régions mieux conservées structurellement. Nous utilisons un tel modéle pour charactériser les protéines de fusion virales de classe II, une tâche particulièrement ardue du fait de leur faible identité de séquence et leur conservation structurale moyenne. Ce faisant, nous parvenons à trouver un certain nombre d'homologues distants connues des protéines virales, notamment chez la Drosophile.
Enfin, en formalisant un sous-problème rencontré lors de la comparaison de *filtrations*, nous présentons un nouveau problème théorique – le D-family matching – sur lequel nous démontrons des résultats algorithmiques variés. Nous montrons – d'une façon analogue à la comparaison de régions de deux conformations d'une protéine – comment éxploiter ce modèle théorique pour comparer deux clusterings d'un même jeu de données.

Mots-clés: Biologie structurale; Alignment structural; Topologie; Persistence; Motifs structuraux; Modéles de Markov Cachés; Annotation de protéine; Théorie des graphes; Clustering

Acknowledgments

First and foremost, I would like to thank my supervisor, Frédéric Cazals. Of course there are all the customary reasons one would thank his supervisor for: scientific guidance, conversations, questions, writing, and all the time and energy spent in these activities. Other reasons come to mind: shared moments of success, but also failure; shared moments of joy, but also abysmal despair. I thank you for all. Above everything else, I believe your careful supervision, and your acceptance of (most of) my eccentricities produced (near) optimal conditions for writing this thesis. I hope you will indulge me this final jest!

I would of course like to thank Dr Gibrat and Dr Cortés for their careful reading of my manuscript. Their pertinent remarks helped improve the presentation of this work. I would also like to thank Dr Weigt and Dr Felix Rey for accepting to be part of my jury.

I have to thank Felix once again, as well as Pablo and Juliette for the exciting discussions during our three-year hunt for class II fusion proteins.

A thesis is comprised of a lot of scientific work, but also involves bureaucracy. I must thank Florence for her patience in that regard. We are (at least I know I am) terrible with filling papers on time and you always saved the day without giving us a well deserved scolding...

I would now like to thank all my fellow ABS cell mates, I mean PhD students, in order of importance (or not, I let you decide): Andrea, Deepesh, Alix, Simon, Augustin, Denys, Mélinée, Thimotée, Ha. Simon and Augustin, being those who had to tolerate me the longest, deserve a special mention. Thank you for the good work atmosphere. Mélinée also deserves a special mention for prompting the creation of our regular Bomberman games. Thank you for helping me to be less serious and allow for some leisure time in my long work days.

ABS isn't only comprised of PhD students. I would like to thank all other team members: Serkan, Dorian, Rémi, Tom. Dorian and Rémi proved to be ideal team-mates in my exploration of other fields, notably graph theory. I would however like to thank them for our ruthless ping-pong competitions. Additionally, Dorian was always a lending ear and an invaluable ally in many occasions. Thank you for that. Tom was my North star for navigating our holy and beloved SBL. Once it becomes a fully formed religion, I will be rooting for your ascension to sainthood. Thank you for all your help. And the countless debates.

Many other people have contributed to this thesis through their help and useful comments. I would like to thank Noël for his help with Apurva and other structural alignment methods. I would also like to thank Dave Ritchie for his help with Kpax.

Obviously, I would like to thank all my friends who provided much needed respite. William in particular gets the privilege of getting called out by name, thanks buddy.

These acknowledgments couldn't be complete without a very special thank you to those who are closest to me. Laura, my partner, had to endure my emotional land-slides at home. Thank you for your relentless support. Live with someone for long enough and you will start to think of their family as an extension of your own: to all your family, thank you as well! Last but not least, I want to thank my mother, my father, my brother and all my family. All of them, each in their own way, have contributed to making me the person I am today. Thank you.

To my old man.

Contents

List of Figures

List of Tables

Chapter 1

Preface

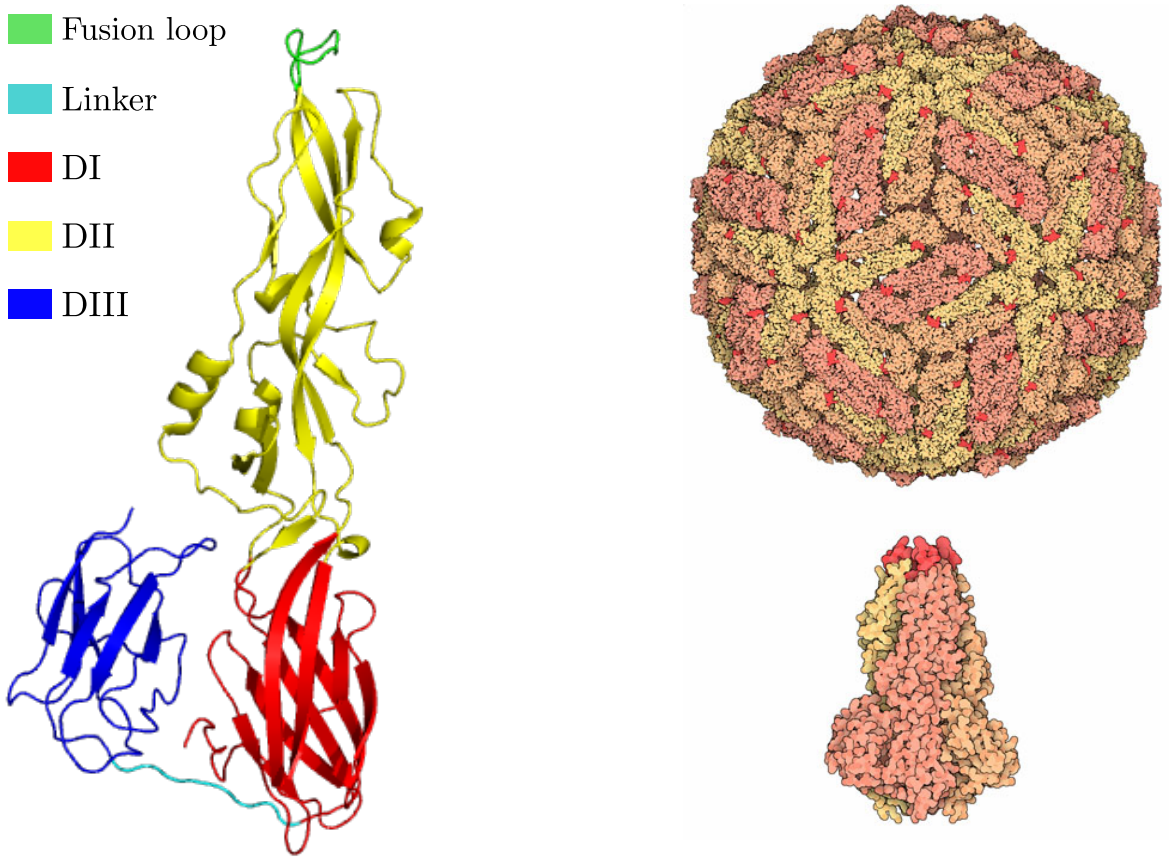
I was first introduced to the field of structural biology six years ago, as an undergraduate student. I have been fiercely loyal to it ever since. Very few fields offer such a wide array of challenging problems for an aspiring computer scientist. As is often the case, the work presented in this thesis started with a very specific problem in mind. Research involves a lot of trial and error, a great deal of exploration and an even greater deal of failure. This project did not derogate from this rule so that, in the end, the initial problem was lost in all the subsequent contributions. The challenge in assembling this manuscript was ensuring that the motivation was not lost to the reader. This preface is intended as an Ariadne's thread, explaining how all the individual contributions are tied together and form the outline of this work. It all started with viral fusion proteins.

Viral replication. As organisms which only replicate by hijacking the cellular apparatus of other organisms, enveloped viruses use a lipid bi-layer (the envelope) to protect their genomes during extra-cellular transport to a new host cell. Initiating infection requires fusing the envelope to the host cell membrane resulting in delivering the viral genome inside the host cell cytoplasm. It is therefore straightforward why understanding the fusion mechanism is such an important stake. Catalyzing the fusion of two lipid bi-layers is a complex process requiring in particular the desolvation of the first hydration shell of the layers to be fused. This step is usually followed by the formation a hemifusion stalk, and finally by the pore formation. This process is triggered by a (low pH induced) conformational change of a membrane glycoprotein, referred to as fusion protein [KR06, WHG07, WDBS08, RR11, Kie14, Har15]. These fusion proteins are ascribed to three classes denoted I, II and III. Despite their variety (class I, II, III), it is believed that fusion proteins follow the same generic mechanism. Evidence for this belief relies on the spatial proximity of the fusion loop targeting the cell membrane and the trans-membrane anchor attaching the protein to the virus envelope. While this proximity in all known post-fusion structures strongly hints at a common mechanism, it does not delineate the mechanism.

Class II fusion proteins. Class II fusion proteins, which are scrutinized in this thesis, are elongated molecules with three domains (DI, DII, DIII) composed primarily of β-sheets (Fig. 1.1). The central DI domain connects via a flexible hinge to the longer DII. Typically, DII contains several conserved disulfide bonds as well as the so-called fusion loop at its tip. Additionally, a linker region connects DI to the DIII domain, which has an Immunoglobulin (Ig)-like fold. From its pre-fusion monomeric conformation, the class II fusion protein arranges itself in a trimer (post-fusion conformation) which spikes the host cell with its fusion loop.

Recent studies have unveiled class II fusion proteins in non viral organisms. Particularly, the structure of the EFF1 fusogen for the nematode was solved in 2014 [PVKV$^+$14] and the structure of the Hapless 2 protein involved in gametes fusion was solved for an algae, a plant and a trypanosome in 2017 [FFL$^+$18]. The HAP2 protein is also known to exist (through sequence studies) in various arthropodes but has yet to be identified in larger organisms. These findings have led class II fusion proteins to be under close scrutiny.

Figure 1.1 Dengue fever virus fusion protein. **(Left)** Class II fusion protein monomer in its post-fusion conformation. Note the hierarchical structure organized in three domains. The fusion loop is the region which spikes the host cell. **(Top right)** Dengue fever virus capsid. **(Bottom right)** Class II fusion protein trimer in its post-fusion conformation.



Indeed finding remote homologs of viral fusion proteins can mean a better understanding of viral evolution, devising new ways of blocking viral infection (through a better understanding of the fusion mechanism) and many other therapeutic applications.

Motivation. The original goal of this project was to identify class II fusion proteins in large eukaryotes, for example vertebrates. This involves building models which characterize such proteins, a particularly challenging task for two reasons [PVKV$^+$14]. First, the sequence identity is very low ($< 15\%$), making finding viable candidates out of reach of classical sequence analysis methods. Second, global structural alignments yield mild lRMSD (or the order of 15Å), blotting out smaller and more conserved regions. These statistics caliber the difficulty of the endeavor aiming at unveiling conservation thresholds, both for the sequence and the structures, corresponding to structural features accounting for the biological function i.e. fusion. Additionally, there are only about a dozen class II fusion structures known to this date, meaning that any method devised would need to work with little data.

Contributions. The entire manuscript can be read in light of this goal. From the few structures that constituted our learning set, we were determined on identifying the features which account for their function. These features could then be exploited to identify new fusion class II protein candidates. Keeping in mind the central idea of structural biology – that it is the structure of a protein which determines its function – we started our endeavor by using structural comparisons. Of course there are a number of state of the art methods allowing for such comparisons, but most of them focus on a global comparison. The statistics discussed earlier show the need for local comparisons. We therefore developed a method which exploits structural conservation scores to detect motifs of increased similarity. Coincidentally, we formalized a new form of distance measure exploiting such motifs. This can of course be extended to any definition of motifs and does not require to detect them automatically (one could specify them by hand). These contributions are presented in Chapters 3 and 4.

Unfortunately, protein structural data is hard to come by. Solving a protein structure requires a lot of effort and is usually expensive. Detecting local structural similarities is no good for finding proteins if there is no data available... There is however abundant protein sequence data available. The statistics discussed earlier rendered attempts at finding new class II fusion proteins unfruitful. From these observations, we decided to try and use the structural motifs we found and incorporate them in sequence models. This contribution is the subject of Chapter 5.

Although we deviate from structural biology in Chapter 6, the contributions presented are still very much tied to our original project. In developing our method for the detection of structural similarities, one step necessitated a bit more attention. This led to the formalization of a problem in Graph Theory which had direct applications in clustering comparison. We present the theoretical results as well as experiments on clusterings.

In the end, the work presented here was a real exercise in research. We failed at identifying class II fusion proteins in vertebrates, but in the process we made a number of contributions in very different fields. A great number of new research questions have been opened by our inquiries, most of which prompted new and ongoing projects. If anything, it is my hope that the reader will have the curiosity to pursue some of these new questions and will find a solid basis to do so in this manuscript.

Chapter 2

Introduction

Alongside water, nucleic acids, lipids and carbohydrates, proteins form an essential component of a living organism. They perform a staggering amount of functions: maintaining cell structure, DNA replication, molecular transport, cellular signaling, etc .... Loss of function in proteins can cause a number of diseases so that many medical treatments involve proteins, be it by restoring function or destroying infectious agents. As such, estimating the function of proteins is crucial to modern research.

One of the driving ideas behind structural biology is that protein function arises from structure and dynamics. It follows from this assumption that proteins sharing a common structure would have an equivalent function. While this proves to be true when two proteins have a high sequence identity – and therefore fold in the same manner – things are far less obvious when a family of proteins displays very little sequence identity, a mild structural homology and yet still perform the same function.

Towards the goal of finding and estimating the function of new proteins, the previous observations are invaluable allies. Sequence comparison tools allow the construction of powerful models which characterize close homologs. These tools thrive in the first scenario described above. Structural comparison tools broaden the perspectives to find remote homologs, in which structure and function have been preserved, but not the sequence. Unfortunately, the cost and complexity of the current methods to solve a protein structure – when possible at all– are detrimental to using structural comparison tools on a wide scale.

The shortcomings of both range of tools render the systematic identification of new proteins when in the second scenario out of reach of the state of the art. In this thesis, the focus is set on reconciling the sequence and structural realms for proteins by exploiting structural information to build hybrid sequence models aiming at characterizing such protein families.

2.1 A primer on proteins and their structure

This section is intended for the computer scientist unaccustomed with structural biology. The curious reader is referred to the basic text books ([BT98, PR08]) as well as to [Fer99].

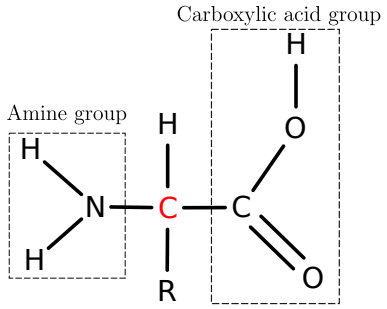
A basic model of a protein looks like an un-ordered mass of atoms. The most convenient way to understand its structure is to look at the hierarchical organization of a protein from its most basic building block up to its most complex level of arrangement: several proteic chains linked together to form a macro-molecule.

2.1.1 Basic building blocks

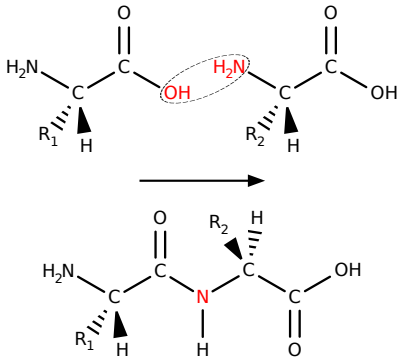
Amino acids. Proteins are composed of organic compounds, called amino-acids. An amino acid is a molecule which contains one amine group, one carboxylic acid group and a side chain (often denoted R when unspecified). When the amine and carboxylic acid groups are attached to the same carbon, called the C_α carbon, we refer to it as an α-amino acid (Fig. 2.1). There are 20 α-amino acids found in proteins which differ solely in their side chains. The side chains confer different bio-chemical properties to the amino-acids

Figure 2.1 The main building blocks of proteins. **a)** Structure of an α amino acid. The C_α carbon, connected to all three groups, is depicted in red. **b)** The formation of a peptide bond between two amino acids by condensation between their carboxyl and amino groups. The peptide bond is *trans* —the side chains are on opposing side of the main chain— and planar. **c)** A list of all 20 α-amino acids found in proteins and their respective codes. Each of these differ solely in their side chains. They are split in three major groups. Glycine is sometimes separated in its own group.

a)



b)



c)

Name	3 letter code	1 letter code
Hydrophobic amino-acids		
Alanine	Ala	A
Valine	Val	V
Phenylalanine	Phe	F
Proline	Pro	P
Methionine	Met	M
Isoleucine	Ile	I
Leucine	Leu	L
Glycine	Gly	G
Charged amino acids		
Arginine	Arg	R (+)
Aspartic acid	Asp	D (-)
Lysine	Lys	K (+)
Glutamic acid	Glu	E (-)
Polar amino acids		
Cysteine	Cys	C
Glutamine	Gln	Q
Histidine	His	H
Asparagine	Asn	N
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y

and they can be separated in three main groups: hydrophobic, charged (positively or negatively) and polar. We now use the terms amino-acid or residue to refer to an α-amino acid.

Peptide bonds. The smallest building block of a protein is the peptide bond. It is the chemical bond formed by condensation between the amino group of one residue with the carboxyl group of another. Although there are a few rare exceptions, the peptide bond always has the same structure: it is trans, meaning that the functional groups of the involved amino acids are on opposing sides of the carbon chain, and planar (Fig. 2.1).

2.1.2 Protein structure

We denote four levels of structural organization in a protein. We present these four levels in Fig. 2.2 on the class II fusion protein of the Dengue virus.

The sequence of residues linked together with peptide bonds constitutes the **primary structure** of the protein. For nearly all intra-cellular proteins, this consists of a unique linear polypeptide chain [Fer99]. However, extra-cellular proteins often contain covalent disulfide cross bridges formed by the thiol groups of two cysteins. Note that in general scientists distinguish between polypeptides and proteins through structure. Proteins are polypeptides which fold into a fixed structure. A protein can be constituted of one or more polypeptide chains.

The polypeptide chains are organized in hydrogen-bonded regions. In these regions, the C=O groups form hydrogen bonds with the NH groups. Through these interactions, the chain forms repeated motifs: α-helixes and β-sheets. These are called *secondary structure elements*. The **secondary structure** of a protein refers to its decomposition into secondary structure elements surrounded by non organized regions called *loops*.

The spatial conformation of all the atoms of a protein is called **tertiary structure.** Because it is driven by the burial of the hydrophobic side chains, the folding of a protein in its final conformation requires: i) that the structures be compact to minimize the contact area of hydrophobic side chains with water, and ii) that any buried hydrogen bonding groups should be paired. The second requirement is insured by the formation of secondary structure elements. The first requirement is insured by the packing of secondary structure elements so that the side-chain of their constituent atoms stack together in an intertwined fashion. Note that the packing density of a protein (0.75, dimensionless) is very close to that of a crystal ($0.7 - 0.78$), so that proteins are not loose but actually very tightly packed [Fer99]. Oftentimes, the **tertiary structure** is composed of several interconnected *domains* —parts of the polypeptide chain that can fold independently— made up of repetitive motifs of secondary structure elements.

Two or more folded polypeptide chain can associate through non-covalent bonds to form a single multimeric protein. This is then called the **quaternary structure** of the protein.

2.1.3 Available data: main sources

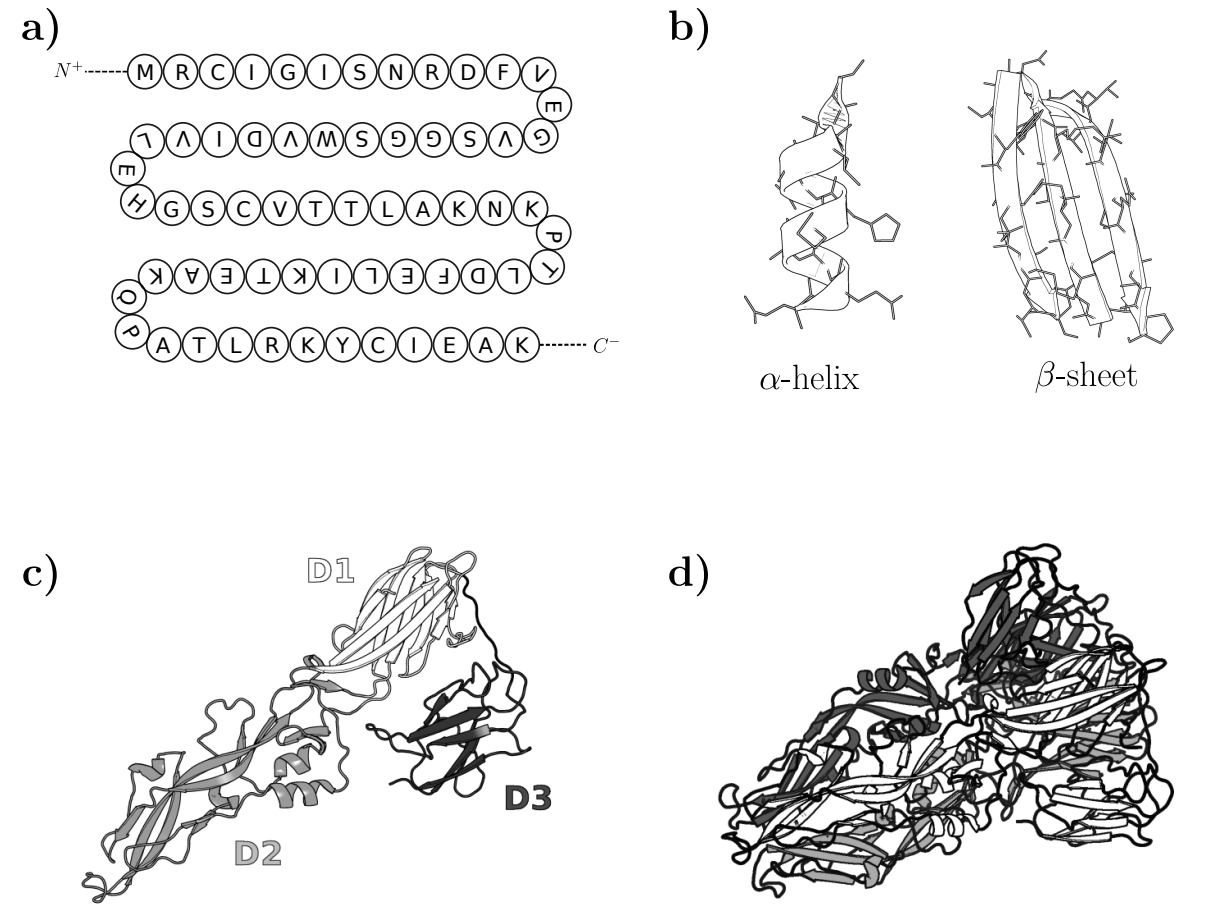
Protein data is made available from a number of sources. The most common one are described in the following.

`UniProtKB` The UniProt knowledge base [The17] is a database of functionally annotated protein sequences, many of which come from the various genome sequencing projects. More than half a million entries in `UniProtKB` are manually curated by experts who critically review experimental and predicted data for each protein. The remainder of the data is automatically annotated using a rule set based on the expert curated knowledge.

Protein Data Bank The Protein Data Bank (PDB) ([BHN03] and Fig. 2.3(B)) is the main resource for structural data of large bio-molecules such as proteins. The data is generally obtained via X-ray crystallography, NMR-spectroscopy, and increasingly cryo-electron microscopy.

EMDB The Electron Microscopy Data Bank [LPB$^+$16] is a repository of three-dimensional electron microscopy (3DEM) density maps for proteins and associated bio-molecules. There has been a peak of interest in 3DEM in recent years as recent technological advances have enabled the elucidations of previously inaccessible macro-molecular complexes.

Figure 2.2 Different structural levels of the Dengue 2 virus envelope glycoprotein ectodomain (PDB: 1OK8). a) Primary structure. A chain of residues linked together through peptide bonds. Represented from the N terminal to the C terminal side. **b) Secondary structure.** Through hydrogen bonding interactions, the residues fold to form a first level of organization: α-helixes and β-sheets. **c) Tertiary structure.** The 3D conformation of a unique chain, held together by covalent bonds. The chain is separated in three domains color-coded on the figure.**d)Quaternary structure.** Three chains linked together to form a trimer.



2.2 Protein comparison and function prediction

The identification of distant evolutionary relationship is generally referred to as remote homology detection, a problem usually tackled using three classes of methods, namely alignment methods, discriminative methods, and ranking methods [CGWL16]. Alignment methods resort to multiple sequence alignments [JPKB$^+$98], position specific profiles [AMS$^+$97] as well as profile hidden Markov models (HMM) [KBM$^+$94, Söd04]. Discriminative methods treat remote homology detection as a supervised classification problem aiming at training a classifier–see the numerous references in [CGWL16]. Using classical protein classifications such as `SCOP`, proteins in the same super-family but not in the same family are considered remote homologous proteins. Finally, ranking methods approach remote homology detection as a database search. Upon embedding known protein structures into a (generally fixed dimensional) space, the query protein is used to retrieve the nearest neighbors–see again [CGWL16] and the references therein. As discriminative and ranking methods fall out of the scope of this thesis, the rest of this section is aimed at introducing the concepts required for alignment methods. The structure - function paradigm stipulates that it is the structure (and the dynamics) of proteins which accounts for their function. Additionally, it is accepted that the structural conformation of a protein is encoded in its sequence of amino-acids. The prediction of function from sequence and/or structure data is therefore of paramount importance. The search of sequence and/or structural homology may be tackled at three levels, namely for whole proteins, protein domains, and protein motifs within domains. Indeed, different functional constraints typically apply to different regions of the proteins, and even more, within a domain, internal and surface regions undergo different selection pressure depending on their involvement in the structure and/or function.

At the center of methods for comparing protein sequence or structure is the notion of *alignment*, a mapping between subsets of residues from two proteins which characterizes similar features.

Definition. 2.1. *Consider two polypeptide chains S and S'. A pairwise alignment of S and S' is a mapping between the residues of S and S'. It is not bijective: a residue of either sequence can be mapped with a "gap". Usually, it is order preserving: if S_i and S'_j are aligned, then for $k > i$, S_k cannot be aligned with S'_l if $l < j$.*

Typically, an alignment involves an objective function so that when two residues are aligned one can associate a cost to the aligned pair (be it two residues or one residue and a gap). Alignments can be local or global, underlining the granularity at which a comparison is done.

2.2.1 Sequence analysis

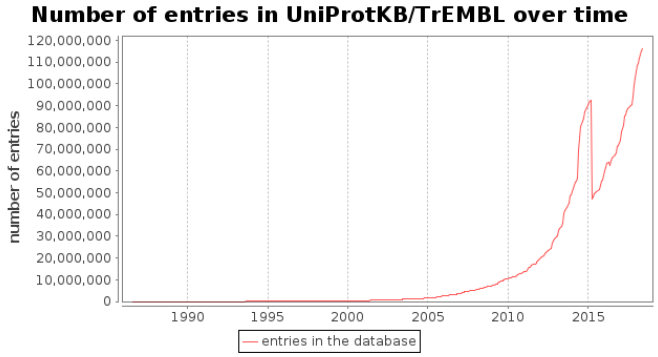
Since the human genome project was completed in 2003, there has been a tremendous increase in sequence data with many more full genomes being sequenced. According to the most recent tally of the Uniprot data base, it contains no less than 110 million protein sequences (Fig. 2.3). In comparison, the protein data bank, contains around 130 000 structures of proteins and their complexes, the diversity of which seems to be reaching a plateau ([Lev07]).

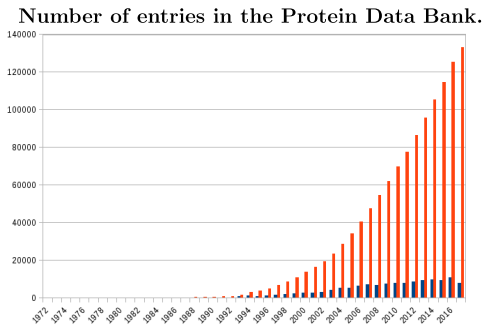
Protein sequence data being so plentiful, the ability of finding underlying features which characterize protein homology from their sequences is desirable. This is especially true since `UniProtKB`/Swiss-Prot (the manually curated and functionally annotated part of UniProt) only contains about half a million entries (0.45% of the total database!). In searching for such features, a relatively simple case is that of proteins harboring high sequence identity, say above 30% [Xia06]. In that case, pairwise sequence alignments (PSA) are generally sufficient to infer homology.

Pairwise sequence alignments

The most convenient way of modeling PSAs is by placing both sequences on a rectangular grid, a path from the upper left cell of the grid to the lower right cell being an acceptable alignment. In fact it can be shown that there is a one to one mapping between the set of alignments and the set of such paths. The PSA problem consists in finding the alignment that makes the most biological sense. This is typically done by using a

Figure 2.3 Sequence data is abundant compared to structural data. `UniProtKB` and the Protein Data Bank both show an exponential increase in recent years. However there is hundred times more protein sequences available. The sharp dip observed in the number of entries in `UniProtKB` is due to proteome redundancy minimization implemented in March 2015 [BBB+16].





scoring function: substitutions between amino-acids with similar properties score higher (with the highest score being two identical residues), opening and prolonging gaps is penalized. This problem has been shown to be identical to finding the single source optimal path in a directed multilayered network and can be solved efficiently with a dynamic programming strategy. The algorithm was first described by Needleman-Wunsch [NW70] and has since been optimized by divide and conquer strategies and by using affine gap penalties (denoting a cost for opening a gap as well as one for extending a gap). The current complexity is $O(L^2)$ in time and $O(L)$ in space where $L = \frac{n+m}{2}$ for n and m the length of the two sequences. By slight modifications on the in the dynamic program, one can also compute a local alignment, searching for the highest scoring sub-sequences. This is called the Smith-Waterman algorithm ([SW81]). The situation deteriorates below the previous threshold, as pairwise sequence alignments fail to capture evolutionary relationships known on the base of structure and function, especially for sequence identity less than 20% [JPKB+98].

Multiple sequence alignments

Multiple sequence alignments are a direct generalization of PSAs for N sequences (with $N \geq 3$). By expanding the aforementioned grid to an N-dimensional grid, the same dynamic strategy could be used to find an optimal alignment of the N sequences. However such a strategy is infeasible, as it shows a time complexity of $O(2^N L^N)$. In general, finding an optimal MSA is NP-hard [WJ94]. Modern methods rely on clever heuristics. MSA algorithms can be divided into five groups: i) exact methods use the aforementioned DP approach, ii) progressive methods use guide trees to combine PSAs to obtain the final alignment, iii) iterative methods first computes a sub-optimal solution and then a DP approach to improve it and reach a point of convergence, iv) consistency based methods construct a database of alignments used as a stepping stone to find a final alignment v) structure based methods rely on external information of the 3-dimensional structure of the protein.

There are such a variety of programs designed to compute an MSA that an extensive review of the state of the art falls out of the scope of this thesis. We reference the main algorithms that were used in our contributions.

ClustalΩ. Clustal [SWD+11] is probably the most widespread algorithm for calculating MSA. It is a progressive method and relies on the computation of a guide tree. Traditionally, this entails computing all $N(N-1)/2$ distances amongst all N input sequences which would result in a $O(N^2)$ complexity. However, in its latest iteration, Clustal reduces that cost to $O(N(\log N)^2$ by computing distances to $(\log N)^2$ random

seeds and performing a hierarchical clustering. The guide tree —which is only intended as a guide for the MSA, not a precise phylogeny— is then used to align larger and larger groups of sequences in the branching order. At each steps, two alignments are aligned. Initially these are individual sequences but are later modeled as profiles, as more and more sequences get added to them. In ClustalΩ, the intermediary step is entirely handled by hidden Markov Models. Sequences and profiles are converted to HMM which are in turn aligned.

Muscle Muscle [Edg04] is a hybrid method which works in three stages. In stage 1, the aim is producing a multiple sequence alignment emphasizing speed over accuracy. To achieve this goal, a distance which is derived from the fraction of common k-mers (a contiguous sub-sequence of length k) is used to build a tree. This tree is then used to build an initial MSA. Stage 2 improves on the previously built MSA by using the Kimura distance and re-assessing the previously computed tree. A new MSA is build. Stage 1 and 2 are progressive methods. Stage 3 relies on an iterative approach. Each iteration involves separating the previously computed tree in two sub-trees which are used to build two separate MSA profiles which are then re-aligned.

2.2.2 Hidden Markov Models

Hidden Markov models (HMM) are probabilistic models often used to represent a sequence of observations. HMM are widely used in the fields of computational biology and bioinformatics at large: for protein topology detection, protein secondary structure prediction, gene identification, study of SNPs, study of behavioral patterns, animal movement simulations, etc ...

Definition. 2.2. *HMM*

A HMM is a Markov process that involves an emission probability associated with each state. It can be defined as a quintuplet: $H = (S, \Sigma, \alpha, a, p)$.

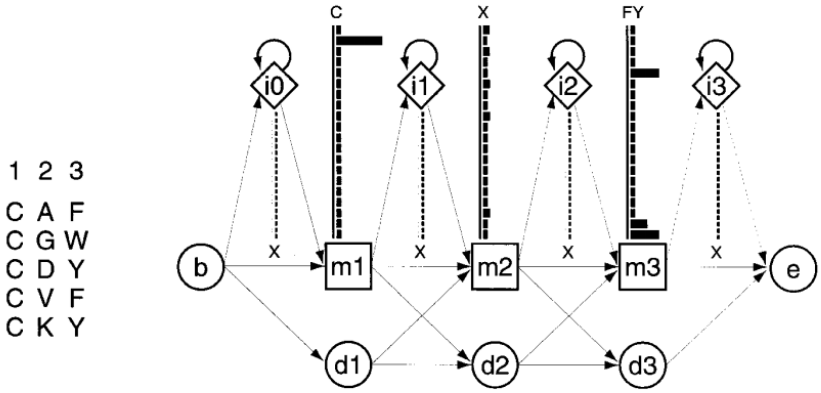
- $S = \{S_1, ..., S_N\}$ *is a finite set of states.*
- Σ *is an alphabet.*
- $\forall S_i \in S, \alpha(i)$ *is the probability of* S_i *being the initial state.*
- $\forall S_i, S_j \in S, a_{i,j}$ *is the transition probability from* S_i *to* S_j.
- $\forall S_i \in S, \forall c \in \Sigma, p(c, S_i)$ *is the probability of issuing symbol* c *when in state* S_i

Profile HMM. A convenient HMM architecture used to model a multiple sequence alignment profile was first introduced in [KBM+94]. A "match" state is associated to each column of the MSA, emission probabilities reflecting the rates of appearance of amino-acids in that column. A "delete" and "insert" state are added between the match states of two columns. An toy example is presented in Fig. 2.4.

Profile hidden Markov models are a fundamental tool to study protein homology [KBM+94, Edd98, Pev15]. In the course of this thesis, two types of profile HMM were used to perform and validate annotations. More precisely, a multiple sequence alignment (MSA) yields a *query*, modeled as a profile hmm. Using this query, we wish to determine whether a *target* sequence from a database (from `UniProtKB`) is homologous to the query. In a second step, HMM-HMM comparisons [Söd04] are used to gain confidence on the hits obtained. In the following, we provide some background and detail these two steps.

Sequence homology assessment as statistical hypothesis testing. This question can be cast into the statistical framework of hypothesis testing [MB06]. In doing so, the null hypothesis is that the query and the target are unrelated, the alternative hypothesis stating that they are biologically related–typically having a common ancestor. The test statistic can be assessed via a p-value calculation using an analytical model for the extreme distribution [MB06] or via a permutation test [PS10]. However, computing the distribution of scores under the null is in general difficult [YH01]. We now review these ingredients more specifically for scores associated with HMM.

Figure 2.4 An example profile HMM representing a small MSA (right) taken from [Edd98]. Each column is associated with a match state (m_1, m_2, m_3). Each match state has emission probabilities for all of the 20 amino acids (represented as a histogram). Insert states are found between all match states $(i_0, \ldots, i_4)$. Delete states can override a match state $(d_1, \ldots, d_3)$. The profile HMM also has an initial state (b) and a final state (e).



Scores for HMM and associated p-values. Given a HMM (our query) and a target sequence, two questions are easily answered using dynamic programming. First, the Viterbi algorithm identifies the most plausible sequence of states which generated the target, together with the corresponding probability. Second, the forward algorithm computes the probability that target was generated by the query. Phrased differently, the Viterbi algorithm identifies the optimal alignment $\overline{\pi}$ between the query and the target; instead, the forward algorithm performs a sum over all possible alignments π, which correspond to sequences of states of the HMM of all lengths.

Denoting H the hmm of the query and R a hmm associated with the null hypothesis, this yields the

$$\text{Viterbi score:} \quad V = \log_2 \frac{\max_\pi \mathbb{P}\left[x, \pi \mid H\right]}{\mathbb{P}\left[x \mid R\right]}, \tag{2.1}$$

and the

$$\text{Forward score:} \quad F = \log_2 \frac{\sum_\pi \mathbb{P}\left[x, \pi \mid H\right]}{\mathbb{P}\left[x \mid R\right]} \tag{2.2}$$

To evaluate the probability that a sequence s got generated by a given HMM, one resorts to the log-odds with respect to a simpler statistical model [BHK97], namely:

$$S = \log_2 \frac{\mathbb{P}\left[s \mid HMM\right]}{\mathbb{P}\left[s \mid null\right]}. \tag{2.3}$$

The null model is naturally critical, and various options have been studied for profile HMM [KBH98]. Of particular interest are reverse sequence null models (i.e. the score of the reverse sequence with the HMM), which uses a sequence of the same length and composition [KKSH05].

The study of distributions of these scores is a delicate topic [MSZW11, Edd08], with a number of conjectures exploited for the calculation of p-values [hmm].

Databases searches and e-values. When using a profile HMM to query a database, the significance of the value $\mathbb{P}\left[s \mid HMM\right]$ is assessed by an e-value–e stand for expectation, which provides an estimate of the number of sequences that would get this score (or better) in a database of the same size containing only random sequences [KKSH05, MB06]. (Intuitively, the p-value and the e-value are related by $e - value =$

$N \times p - value$, with N the database size [Pea98].) The e-value calculation requires the distribution of scores for such random sequences, so that the exact formula depends on the alignment / scoring method and statistical properties thereof. [MB06]. Quoting [hmm] "*The e-value statistics of local and global alignment remain poorly understood*". The reader is referred to [Edd08] and [hmm] for conjectures on the distribution of Viterbi and forward scores used in the computation of e-values returned by `HMMER`.

When dealing with the "twilight zone" of protein sequences (sequences harboring $\leq 30\%$ identity), another route is using structural information. In this effect, powerful alignment schemes exist to compare proteins.

2.2.3 Structural analysis

The structure - function paradigm has served of guideline to investigate two broad classes of problems.

The first class is the analysis of (homologous) protein structures, a key endeavor for protein function annotation [TXK$^+$03]. In this realm, finding structural alignments actually requires solving two problems at once: first, identifying the sub-sequences matching one-another; second, optimizing some (geometric) criterion qualifying the alignment. We note that these two problems are easy when solved independently: finding the alignment for fixed positions is amenable to dynamic programming; finding the optimal superimposition given the alignment is the classical rigid superimposition problem.

The second class is the analysis of thermodynamics and dynamics from simulations (molecular dynamics, Monte Carlo sampling and their generalizations). In this setting, it is indeed well known that reliable thermodynamic and kinetic estimates requires accurate clustering algorithms to group conformations easily inter-convertible into one another [NAKH14].

Structural alignment methods

For these two classes of problems, a variety of structural alignment methods have been developed, which may be classified using their three main ingredients [HH09], namely the molecular representation used, the associated scoring function, and the optimization algorithm run.

• Classical representations favor geometric or topological features. The former are based on Cartesian coordinates and/or internal distances. The latter are based on graphs coding geometric and/or topological properties. A popular class of graphs are contact maps, namely graphs whose edges code the spatial proximity between two a.a. Contact maps may be defined from pairs of C_α which are within a distance threshold [GS94], or using physical contacts between a.a., such as those obtained from Voronoi models [LAI06, OKV13]. We note that graph based representations tend to target more flexible alignments [GS94], as the network of contacts may be locally conserved yet globally geometrically distorted.

• Representations naturally call for specific scores. Geometric representations are typically assessed using the root mean square deviation of internal distances (dRMSD) [HS93, HS95], the coordinate (least) root mean square deviations (lRMSD), or variants [KL04, GL98]. More recently, geometric scores assessing the *compatibility* of backbone fragments using local frames have also been introduced [RGMV12]. Topological representations typically call for scores based on conserved contacts.

• Finally, optimization algorithms used are especially important when it comes to accommodate time execution constraints, and aligners may also be classified by the hardness of the problem solved. On the one hand, selected aligners solve easy i.e. polynomial time solvable problems. Approximate aligners, be they iterative [Zem03, BOPR03, RGMV12] or not [KL04] fall in this category. On the other hand, search for optimal alignments in terms of contacts generally tackle NP-hard (and hard to approximate) optimization problems [CK98, GIP99], which in turns calls for approximate solutions when it comes to handling large scale comparisons [AMDY11].

As seen from this mini-review, the problem of performing structural alignments triggered developments in complementary directions, and pairwise comparisons have evidenced that alignments yielded by individual methods may indeed considerably differ [MDL07]. This state of affairs prompted the development of tools easing the comparison of pairwise aligners [WMDAK12].

Definition. 2.3. *CMO*

In CMO (Contact Map Overlap maximization), proteins are modeled as adjacency matrices. Two a.a. whose Euclidean distance is less than a user defined threshold are termed in contact*. The maximum Contact Map Overlap problem consists in finding the alignment which maximizes the number of common contact i.e. if two residues are in contact their aligned counterparts should also be in contact. Several algorithms and scoring schemes have been used in the case of proteins.*

The following list is a presentation of the main aligners used in the course of this thesis.

Apurva A_purva [AMDY11] is an exact branch and bound algorithm which exploits an integer programming model for CMO. The bounds are obtained through a novel Lagrangian relaxation.

Kpax Kpax [Rit16], [RGMV12] uses an iterative alignment approach. Kpax computes a seed alignment via dynamic programming (DP) with a novel score. The K-score is based on the similarities of a local sliding window on the backbone. From this seed alignment, Kpax then performs a specified number of iterations of: (i) re-adjusting the two structures based on the current alignment (ii) computing a new alignment by DP with a score based on the spatial proximity of two residues.

Scoring structural alignments: RMSD , lRMSD and their variants

The problem of geometrically comparing two point sets of the same cardinality, assuming a one-to-one correspondence between the points, has long been recognized as central in science and engineering. It is in general desired to perform a comparison oblivious to rigid motions, which prompts a solution computing concomitantly the geometric similarity measure and the associated optimal rigid motion. The most celebrated solution to this problem is the so-called least root mean square deviation (lRMSD)[Kab76], namely the RMSD of positions upon applying the optimal rigid motion. This number, which is usually expressed in Å(We note that the lRMSD is a coordinate RMSD, not the be confused with the $RMSD_d$, namely the RMSD of internal distances.) In the sequel, we review previous work, by restricting ourselves to structural bioinformatics.

Several strategies to compute the lRMSD were developed long ago [Kab76, AHB87, Ume91], and it was also noted that the lRMSD induces a metric [Ste02]. Owing to these properties, the lRMSD has been one of the most used similarity criteria in structural biology and bioinformatics. On the other hand, several limitations prompted developments from the design and computational perspectives.

On the design side, efforts were made to circumvent several limitations. The lRMSD is inherently hard to interpret, as medium values may stem from a fuzzy structural conservation contributed by all atoms, or from small regions that underwent large conformational changes while their complement is isometrically conserved. Possibly worse, the lRMSD involves the co-variance matrix of centered atomic positions (see below), so that points far from the center of mass get more weight. This fact also has another consequence: by a packing argument, large proteins distribute atoms farther from the center of mass, so that the lRMSD depends on protein size. In order to weigh all points evenly and also to obtain a normalized measure, a variation of the RMSD obtained by restricting the calculation to unit vectors along the backbone and performing the optimization over rotations was developed [KCE99]. Normalized alternatives were also proposed, respectively based on the radii of gyration of the molecules compared [MC95], and on normalization factors inferred from typical distributions of RMSD values [OCP01, BS01]. In a complementary line of attack, various superposition-free measures were proposed. To compare structures and models, one may use the lDDT which is an average value (computed over four thresholds) of fraction of distances within the chosen threshold [MBBS13]. Similarly, the contact area difference quantifies differences of contact areas between a model and a structure [OKV13]. Finally, a normalized measure based on the Binet-Cauchy kernel, which inherently computes a scalar product between vectors defining the volume of tetrahedra was recently proposed [GT14].

Improvements were also reported to speed-up calculations. Efficient calculations targeting subsets of the aligned structures were investigated [Shi07]. As an alternative to lRMSD calculations based on matrix

decompositions–such as SVD, a fast determination of the optimal rotation matrix based on a Newton-Raphson quaternion-based method was reported [LAT10]. Finally, recent work addressed the calculation of RMSD between flexible structures, the flexibility being modeled using collective coordinates obtained via normal mode analysis or PCA [NPH+18]. This latter work is especially interesting as it targets deformable structures modeled by means of collective motions.

To conclude, we also note that RMSD and lRMSD calculations are tightly related to the calculation of structural alignments between two structures. The intrication between scores and alignment methods was first exploited in [Zem03], which performs an iterative alignment, guided by two scores, namely `GDT` (the fraction of residues (largest set, not sequence contiguous) that fit under a distance cutoff), and `LCS` (the fraction of amino acids defining the longest contiguous segment fitting under a given RMSD cutoff (positions of molecules fixed)). Since then, a variety of methods were proposed to detect and score structural motifs [WMDAK12], including `DALI` [HS95], `TM-align` [ZS05], `Apurva` [AMDY11], `LGA` [Zem03], `Kpax` [RGMV12], or our persistence based method [CT18b].

2.2.4 Hybrid comparison methods

Because protein structures and functions tend to be more conserved than sequences [PPL06], the use of combined sequence - structural information also offers various routes to improve the detection of remote homologs. These methods target different features of proteins, including folds, pockets and clefts, active sites, interfaces and protein - protein interactions [TXK+03, WLT05, LRO07, LWT05]. These methods are useful but require care. First, no best structural alignment scheme exists, and different methods typically trade the length of the alignment against its quality [HH09, WMDAK12]. Second, selected protein regions–loops in particular–may not have a unique structural alignment, so that purely geometric approaches may loose important sequence information. Finally, the stringency thresholds used for sequence and structure information must be chosen with care, as restrictive thresholds may generate information loss on homology distant molecules, while loose ones may yield a signal dilution [TXK+03]. Despite these features, combined sequence - structure based methods proved instrumental in enhancing remote homolog detection. Of particular interest are hybrid multidimensional alignment profiles (HMAP, [TXK+03]), a method combining primary, secondary and tertiary structure information. Under suitable hypothesis, HMAPs enhanced SCOP super-family and fold detection [TXK+03], and it was shown that sequence and structure contain some unique information.

2.3 Geometric models and associated constructions

This chapter is meant to introduce geometric and topological concepts for structural biology as well as some additional constructions used in the course of this thesis.

2.3.1 Representing proteins: space filling diagrams

The first efforts to represent proteins date back to 1865 when a basic *ball and stick* model was developed by August Wilhelm von Hoffman. Such models involve spheres of equal radii, to represent atoms, interconnected with rods, which represent the bonds. With the rise of computer graphics, powerful new models now exist to visualize protein structure. The *cartoon diagrams* (Fig. 2.5 a) display the overall path and organization of the protein backbone. *Stick models* (Fig. 2.5 b) are pretty much a computerized version of the ball and stick model described above.

Space-filling diagrams (Fig. 2.5 b) arise out of the desire to represent molecules so to outline their electronic surfaces and the way they interact with one another. The *Van der Waals force* captures the interactions between atoms. At short range, the force is attractive. At very short range, the force is strongly repulsive. We can then assign to the atoms *Van der Waals radii* so that the force changes from attractive to repulsive when these spheres touch. The most common space-filling diagram of a protein involves a union

Figure 2.5 Three representations of the Dengue virus class II fusion protein. a) The *cartoon* representation models the geometry of the backbone. **b)** The *stick* representation models the chemical connectivity of the protein. Generally an atom is simply a vertex and a stick represents a chemical bond. **c)** The *space-filling* diagram models the protein by the space it occupies. The atoms are represented by solid spheres or balls the union of which is the protein.

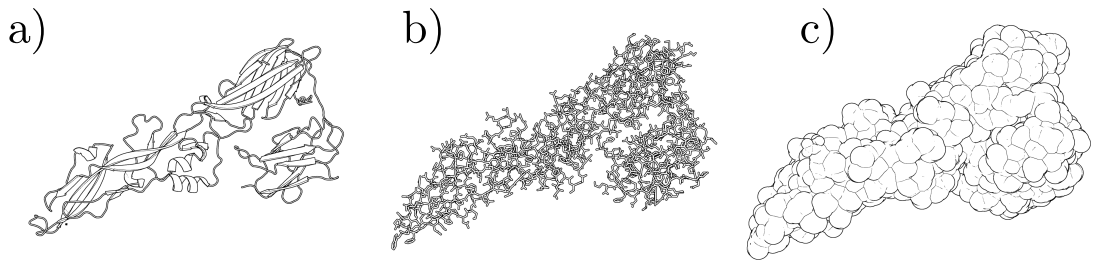
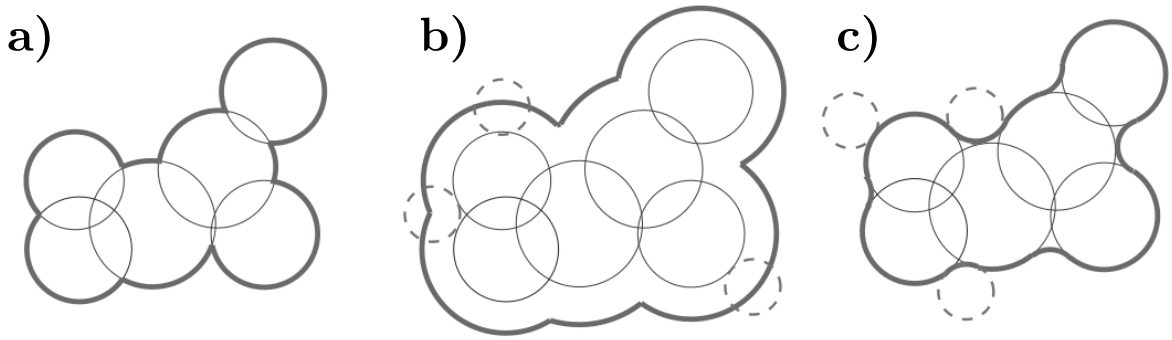


Figure 2.6 Common space-filling models for representing proteins (2D example), image taken from [GOT17]. a) Van der Waals. The space-filling diagram is the union of balls with Van der Waals radii. The Van der Waals surface (bold lines) is defined as the boundary of this diagram. **b) Solvent Accessible Surface (SAS).** Each Van der Waals ball is enlarged by the radius of a water probe (typically $1.4\mathring{A}$) **c) Connolly surface.** Obtained by rolling a solvent sphere on top of the Van der Waals surface. Corresponds to the boundary of space inaccessible to the solvent.



of balls representing atoms the radii of which are proportional to their corresponding atoms Van der Waals radii. Center to center distances correspond to the chemical bond lengths. These diagrams are called *CPK models* after their three developers Robert Cory, Linus Pauling and Walter Koltun. The boundary of this space-filling diagram is called the *Van der Waals surface* (Fig. 2.6 a). In a space-filling diagram, by growing the balls by the radius of a water probe (typically 1.4 $\mathring{A}$), one obtains the *Solvent Accessible Surface model* (or SAS model, Fig. 2.6, b).

2.3.2 Basic definitions

First let us introduce some basic concepts. In the following, we consider points and balls in $\mathbb{R}^d$.

Definition. 2.4. *Simplicial complex A* k-simplex *is a defined as the convex hull of $k+1$ vertices which are affinely independent. A* face *(resp.* coface*) of a simplex is defined similarly using a strict subset (resp. superset) of its vertices. A* simplicial complex *K is a set of simplices which satisfies two conditions: (i) every face of a simplex from K also belongs to K, (ii) the intersection of two simplices s_1 and s_2 from K is void or is a face of s_1 and s_2. A simplicial complex K' is said to be a* subcomplex *of K if $\forall \sigma, \sigma \in K' \Rightarrow \sigma \in K$*

For example, a 0-simplex is a point, a 1-simplex a segment and a 2-simplex a triangle. We display some example simplicial complexes in Fig. 2.7.

The ordering of these complexes yields a *filtration*.

Definition. 2.5. *Filtration A* filtration *of a simplical complex K is a set of subcomplexes $\{K_0, \dots, K_m\}$ with $K_m = K$, such that for any two subcomplexes K_i and K_j, $i < j \Rightarrow K_i \subseteq K_j$.*

A convenient way to represent a filtration of a simplical complexe, or any partially ordered set for that matter, is a so-called *Hasse diagram*.

Definition. 2.6. *Hasse diagram A* Hasse diagram *is the visual representation of the inclusion relationships in a partially ordered set. Each member of the set is represented as a vertex. For two vertices s_1, s_2, if $s_2 \subset s_1$, there is an arrow from s_1 to s_2 in the diagram. An example is displayed in Fig. 2.7*

2.3.3 Voronoi diagrams, Delaunay triangulation and α-complex

Let us now introduce the *Voronoi diagram* of a collection of balls and its dual structure, the *Delaunay triangulation*. The *Power distance* from a point p to a ball $b(c, r)$ of center c and radius r is the square length of tangent line from p to B:

$$\pi(p, b) = (p - c)^2 - r^2. \tag{2.4}$$

Given two balls b and b', the portion of space which contains all points with same power to the two balls is called their *bisector*.

For a set of points and more generally a set of balls with equal radii, note that this bisector matches the one defined from Euclidean distance.

For a given set of n balls $\mathcal{B}$ in a d-dimensional space $\mathbb{R}^d$, a *Voronoi cell* associated to a ball $b_i \in \mathcal{B}$ is the portion of the space that consists of all points p for which $\pi(p, b_i)$ is minimized, intuitively, all points which are closest to b_i. Formally:

$$Vor(b_i) = \{p \in \mathbb{R}^d | \pi(p, b_i) \leq \pi(p, b_j), \forall b_j \in \mathcal{B}\} \tag{2.5}$$

Such a Voronoi cell, associated to one point or ball, is deemed to have dimension d. Consider a set of $i+1$ d-dimensional Voronoi cells, with $0 < i < d+1$. *Generically*, their intersection defines a Voronoi cell of dimension $d - i$. Generically means that we omit here so-called degenerate configurations (such as e.g. four cocircular points in 2D; see [BY98] for a thorough discussion on such cases.) As an example, in two dimensions:

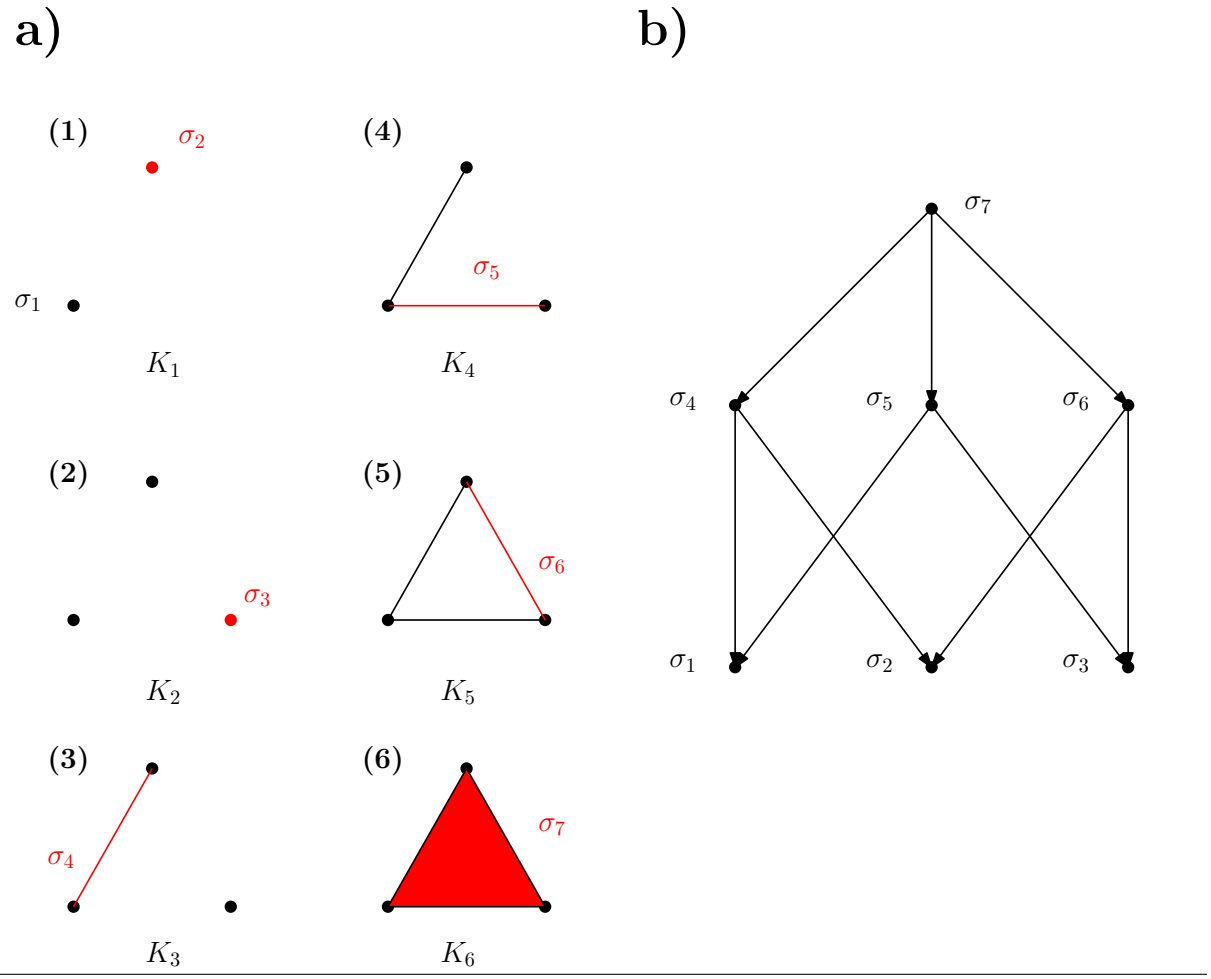
- two 2-dimensional *Voronoi cells* intersect on a portion of the *bisector* of their corresponding balls. This is a 1-dimensional *Voronoi cell* and is called a *Voronoi face*.
- Similarly, two *Voronoi faces*, or three 2-dimensional *Voronoi cells* intersect on a point. This is a 0-dimensional *Voronoi cell* and is also called a *Voronoi vertex*.

The union of *Voronoi cells* of all dimensions forms the *Voronoi diagram* [Aur87, GOT17] (Fig. 2.8).
The Voronoi diagram admits a dual structure, the *Delaunay triangulation*.

Definition. 2.7. *Delaunay triangulation For a set of balls S, let $\Delta(S)$ be the simplex formed by the convex hull of the centers of the balls (one ball forms a 0-simplex, two balls a 1-simplex and three balls a 2-simplex). For a given set of points, or balls of equal radii, $\mathcal{B}$ The* Delaunay triangulation *$DT(\mathcal{B})$ is defined as:*

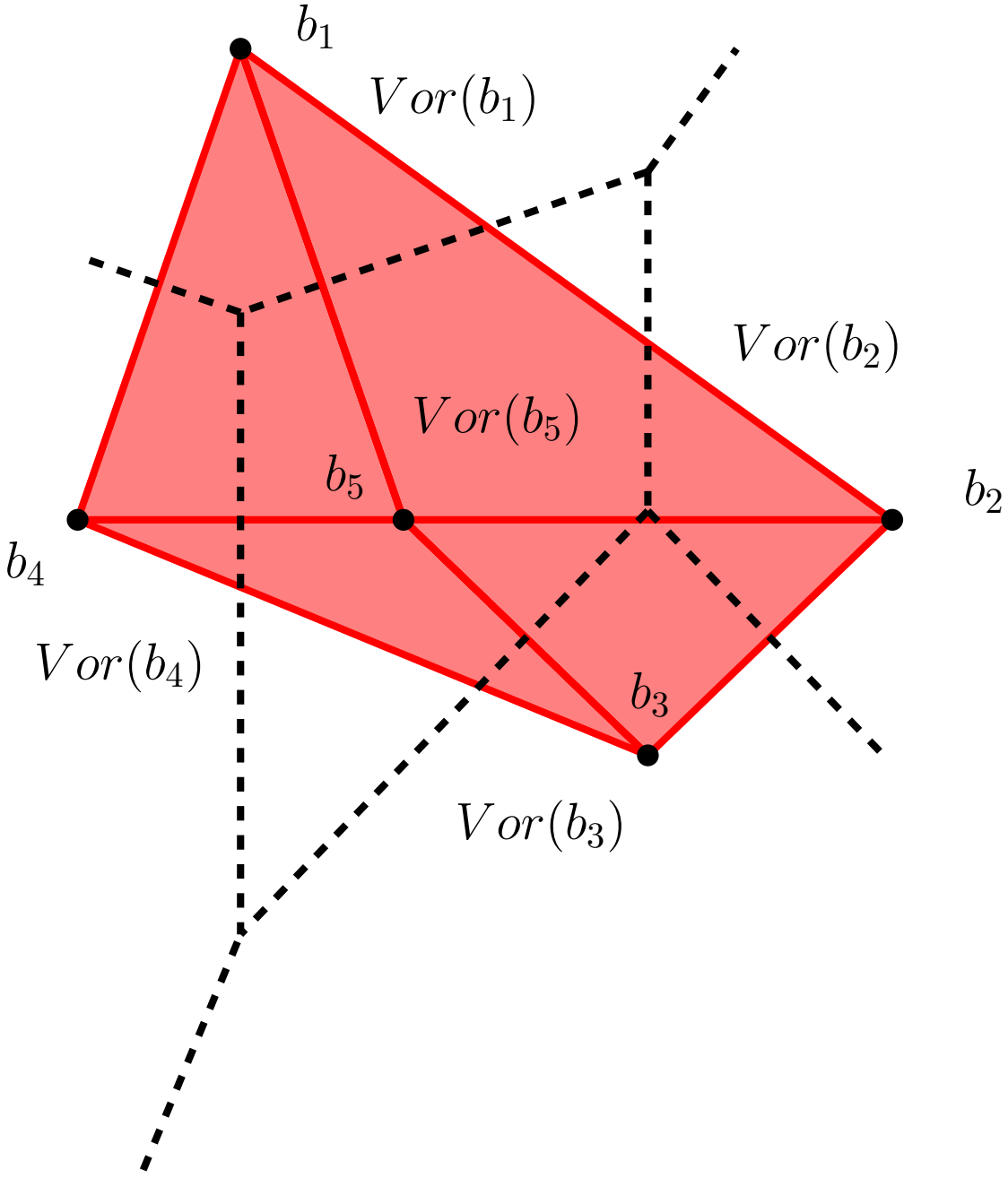
$$DT(\mathcal{B}) = \{\Delta(S) | S \subset \mathcal{B}, \bigcap_{b \in S} Vor(b) \neq \emptyset\} \tag{2.6}$$

Figure 2.7 Filtration and Hasse diagram of simplicial complexes. a) A set of six two-dimensional simplicial complexes. Note that $K_{i+1} = K_i \cup \{\sigma\}$, with σ a simplex. The simplex added is highlighted in red. The set $\mathcal{K} = \{K_1, K_2, K_3, K_4, K_5, K_6\}$ defines a filtration of K_6: $K_1 \subset K_2 \subset K_3 \subset K_4 \subset K_5 \subset K_6$. **b)** Hasse diagram for the vertex sets defining the simplices. An arrow represents an inclusion relationship.



Note that under suitable genericity assumptions, the Delaunay triangulation is a simplical complex. Please refer to Fig. 2.8 for an example.

Figure 2.8 Voronoi diagram and Delaunay triangulation of a set of points – or equivalently of balls of equal radius. The dashed lines represent the 1-faces of the Voronoi diagram of the set of points $\mathcal{B} = \{b_1, b_2, b_3, b_4 b_5\}$. Each point in $Vor(b_i)$ is closer to b_i than $b_j, \forall j$. Note that there is no degenerate case here (such as four co-circular points) so that two 2-dimensional cells intersect on an edge (1-dimensional cell) and three 2-dimensional cells intersect on a point (0-dimensional cell). Following Eq. (2.6), the Delaunay triangulation involves vertices, edges (red bold) and triangles. Each segment $b_i b_j$ from the Delaunay triangulation is perpendicular to the Voronoi edge between $Vor(b_i)$ and $Vor(b_j)$. Here $DT(\mathcal{B})$ contains five 0-simplices (vertices), eight 1-simplices (edges) and four 2-simplices (triangles).



In this thesis, we make extensive use of the α-complex, a nested sequence of simplical complexes which are subsets of the Delaunay triangulation.

Definition. 2.8. *α-complex Given a set of n balls $\mathcal{B} = \{b_1, \ldots, b_n\}$, we denote by $S_{b_i,\alpha}$ the sphere centered in b_i with radius $\sqrt{r_i^2 + \alpha}$, where r_i is the radius of b_i. For a given parameter α, the α-complex K_α is a simplicial complex defined as follows:*

$$\forall T \subset \mathcal{B}, \Delta(T) \in K_\alpha \Leftrightarrow \bigcap_{b_i \in T} (Vor(b_i) \cap S_{b_i,\alpha}) \neq \emptyset \tag{2.7}$$

Note that growing α amounts to considering a set of growing balls, which fill the entire space. Such a model is sometimes called a *space-filling* model (SFM). The SFM associated with a value α is denoted $\mathcal{F}_\alpha$.

Importantly, the simplices which appear along this process, as specified from Eq. (2.7), can be computed from the Delaunay triangulation. The calculation is non trivial, see e.g. [Ede95, Ede92].

An example of the α-complexes associated to a set of growing points is provided in Fig. 2.9.

A number of features can be tracked through the evolution of these growing balls. For example, the Betti numbers allow us to track topological features. The *$k-th$ Betti number* corresponds to the number of *k-dimensional holes* of a topological space:

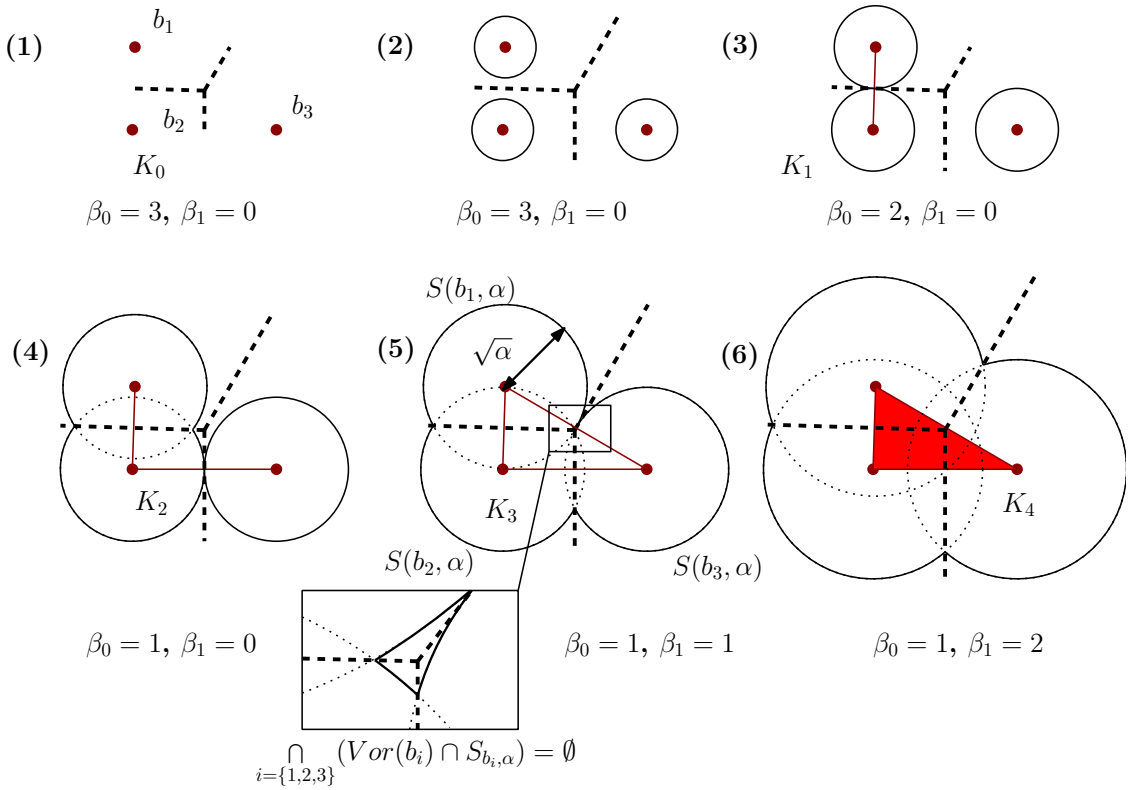
- β_0 is the number of connected components,
- β_1 is the number of cycles,
- β_2 is the number of holes.

In Fig. 2.9, we track β_0 and β_1 as an example. For a simplicial complex, the alternated sum of Betti number or counts of simplices of all dimensions define the Euler characteristic. That is, denoting $\#s_i$ the number of i-simplices, one has [EH10]:

$$\chi = \sum_i (-1)^1 \beta_i = \sum_i (-1)^i \#s_i. \tag{2.8}$$

As an example, the Euler characteristic of the triangulation of Fig. 2.8 is equal to $\chi = 5 - 8 + 4 = 1$ (using counts of simplices) and also $\chi = 1 - 0 + 0$ (using Betti nunmbers).

Figure 2.9 Alpha complex of a collection of growing points. We display the alpha complex for 6 values of α. Note that the set of complexes K_i ordered from $i = 0$ to 4 yields a filtration of K_4. K_4, the last possible K_α is equivalent to the Delaunay triangulation. Under each panel we display the evolution of Betti numbers β_0 and β_1. Of particular interest is panel 5. As illustrated by the inlay the restriction of spheres $S(b_i, \alpha)$ to their respective Voronoi regions $Vor(b_i)$ do not intersect. This implies that the triangle (b_1, b_2, b_3), which does not fulfill condition Eq. (2.7), does not belong to K_3 and that $\beta_1 = 1$: there is a 1-dimensional hole or cycle. Note that in this case, the Euler characteristic reads as $\chi = 1 - 1 = 3 - 3 = 0$.



2.3.4 Persistence diagram for connected components in a filtration

We have seen above that growing balls define a space filling model $\mathcal{F}_\alpha$, and that in using the restriction of balls to their Voronoi cells, one can define the α complex via Eq. (2.7). The last α-complex matches the Delaunay triangulation, and it is convenient to denote the corresponding filtration as follows:

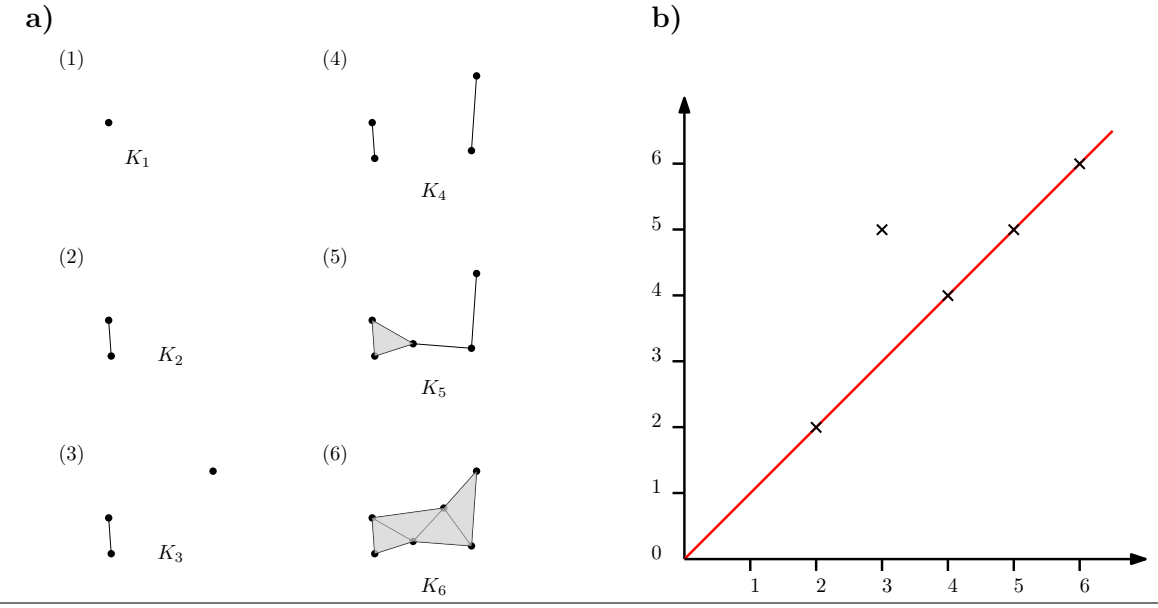
$$\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K \tag{2.9}$$

We have also seen that the addition of simplices triggered by Eq. (2.7) was triggering an evolution of topological features and Betti numbers along the filtration. Of particular interest are the moments in time (counted in units of α) when these topological features evolve.

In this thesis, we take particular interest in tracking connected components. For example, for non intersecting balls, one initially has a number of connected component matching the number of balls. But as soon as two balls merge along the growth process defining the SFM, one connected component disappears.

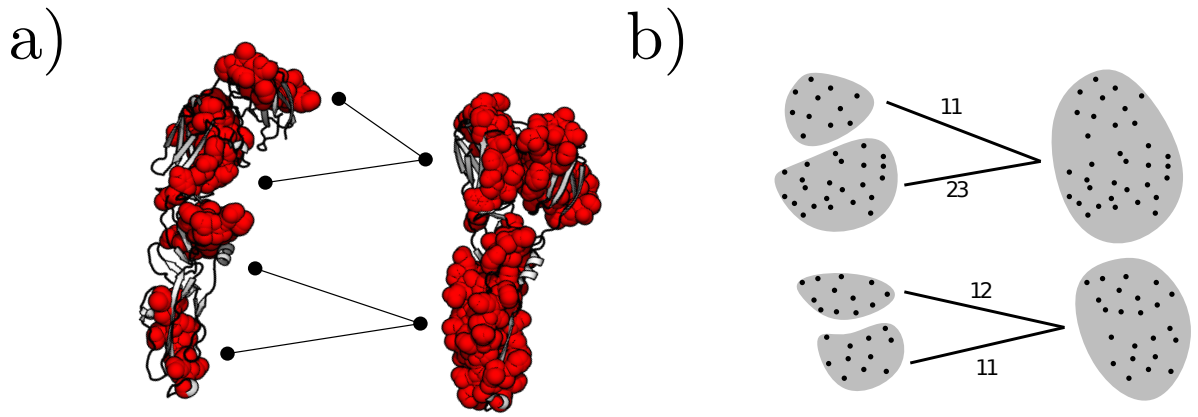
The *filtration value* at which a connected component appears is its *birth date*, the *filtration value* at which it merges with an older connected component is termed its *death date*. The oldest connected component never dies. Each connected component can then be paired with a 2D point consisting of its birth and death date. By plotting all such points, one obtains the *persistence diagram* (Fig. 2.10). The *persistence* of a connected component corresponds to its lifespan (*death* − *birth*).

Figure 2.10 Persistence diagram. a) Filtration of simplical complexes: $K_1 \subset K_2 \subset K_3 \subset K_4 \subset K_5 \subset K_6$ **b)** Persistence diagram of connected components upon processing the simplical complexes in the order of the filtration. The main merge event happens at 5 when the connected component created at 3 merges with the oldest connected component. The corresponding connected component has a persistence of 2. Note that the most persistent component, nested at time 1, never dies.



It should be noticed that the notion of persistence actually holds for all generators of topological features accounted for in Betti numbers; but such properties are not used in this thesis.

Figure 2.11 Comparing structural motifs. **a)** Two motif sets on two different conformations of the same protein structure. Most motifs on the left hand side are either nested or intersect with the motifs on the left (w.r.t their constituting amino-acids). Intersections are depicted as an edge. **b)** Two clusterings of the same dataset. The situation is completely analogous to the comparison of protein conformations (as depicted on the left of this figure). Here we also add the number of common points between two clusters as edge weights.



2.4 Clustering methods and their comparison

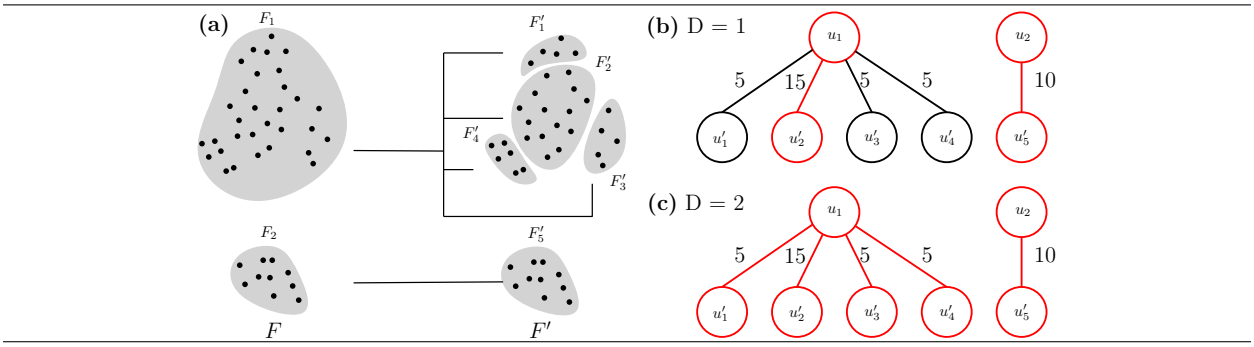
2.4.1 From protein comparison to clustering

When comparing protein conformations and searching for structural similarities, it can be of interest to identify the commonalities between two structures. Consider two sets of structural motifs, each one from a different conformation of the same protein structure (Fig. 2.11). Finding which parts of the motifs are common two both conformations is a very natural question. In dealing with such considerations, the parallel to clustering becomes clear: the protein structure corresponds to the data set (both conformations contain the same residues, albeit in a different geometrical conformation), and the motifs correspond to the clusters.

2.4.2 Clustering methods

Clustering, namely the task which consists in grouping data items into dissimilar groups of similar elements, is a fundamental problem in data analysis at large [XW05]. Existing clustering methods may be ascribed to the following categories. *Hierarchical clustering* methods typically build a dendrogram whose leaves are the individual items, the grouping aggregating similar clusters [DH73]. *k-means and variants* perform a grouping induced by the Voronoi cells of the cluster representatives, which are updated in an iterative fashion [AV07]. In *density based clustering* methods, a density estimate is typically computed from the data, with clusters associated to the catchment basins of local maxima [Che95]. Topological persistence may be used to select the significant maxima [CGOS13]. The notions of culminance and proeminance, which have been used since the early days of topography by geographers to define summits on mountains, can also be used to define clusters [RL14]. Finally, *spectral clustering* methods define clusters from the top singular vectors of the matrix representing the data (or their similarity) [VL07]. Each of these categories comes with its intrinsic difficulties. For example, the smart seeding strategy of k-means yields an algorithm with an approximation guarantee [AV07]; yet, k-means++ still suffers from instabilities when the number of centers used is larger than the *exact* number of clusters, as the clustering obtained depends on the initial distribution of centers

Figure 2.12 Comparing two clusterings of the same 2D data set involving 40 points. (a) Clustering F contains 2 clusters of respectively 30 and 10 points. Clustering F' contains 5 clusters of respectively 5, 15, 5, 5, and 10 points. In (b) and (c), the intersection graph associated with the two clusterings is depicted: one node per cluster, an edge between two nodes if the corresponding clusters share at least one point, the weight of an edge being the number of points shared by the two clusters. Our method groups clusters within meta-clusters. It is parameterized by the diameter D of the sub-graphs connecting clusters within meta-clusters (in red). Existing methods based on (maximum) graph matching correspond to $D = 1$. **(b)** With $D = 1$, a matching is obtained: F_1 with F'_2 and F_2 with F'_5. **(c)** With $D = 2$, $\{F_1\}$ is matched with the meta-cluster involving $\{F'_1, F'_2, F'_3, F'_4\}$, while $\{F_2\}$ is matched with $\{F'_5\}$.



within the clusters [VL07].

This type of difficulty together with the vast array of clustering schemes actually raises two important questions. The first one is the design of cluster quality measures, a topic for which recent work has put emphasis on the performances of spectral clustering methods [KVV04]. The second one is the problem of deriving a consensus for an ensemble s of clusterings. We briefly review previous work on these.

2.4.3 Clusterings: comparison and stability assessment

To describe existing cluster comparison methods, we consider two clusterings F and F' of some data set $Z = \{z_1, \ldots, z_t\}$ composed of t items. Recall that the contingency table of F and F' is the matrix in which a cell counts the number of data items common to any two clusters from F and F'. For the sake of exposure, we define a *meta-cluster* of a clustering as a set of clusters of this clustering.

In *set matching based comparisons* [LA99, Don00], a *greedy best effort* 1-to-1 matching between clusters is sought from the contingency table. A statistic is designed by adding up the contributions of these pairs. The resulting measure is often called the *minimal matching distance (MMD)* [Lux10]. To define MMD for the k-means algorithms, assuming that k clusters are produced, each identified by a label, denote $\Pi = \{\pi\}$ the set of all permutations of the k labels. The MMD is defined by $d_{MMD}(F, F') = \frac{1}{t} \min_{\pi \in \Pi} \sum_{i=1}^{t} \mathbf{1}_{F(x_i) \neq \pi(F'(x_i))}$, where $F(x_i)$ ($F'(x_i)$, respectively) is the cluster of F (F', respectively) containing x_i. Finding the best permutation reduces to a maximum perfect matching, and thus has polynomial complexity. Likewise, to compare clusterings with different numbers of clusters, one computes a maximum weight bipartite matching. However, MMD is inherently based on a 1-1 mapping between clusters, a stringent condition we shall get rid off–MMD shall be covered by the diameter constraint $D = 1$ in our framework. See Figure 2.12.

In *pair counting methods* [Mei02], each pair of items is ascribed to a category out of four (in the same cluster in F and F', in different clusters in F and F', in the same cluster of F but in different clusters in F', and vice versa). A statistic is then computed (e.g. the Rand index). While relevant for problems where pairs are of paramount importance, such methods do not provide any insight on the relationships between clusters of the two clusterings.

In *information theoretical methods* [Mei02], the coherence between clusters of F and F' is assessed using the variation of information (VI) between the clusterings. In short, VI is defined from the mutual information between the two clusterings [CT06], namely the Kullback–Leibler divergence between the joint distribution

and the marginals defined from the contingency table. While VI defines a metric, it exhibits the drawbacks of pair counting methods.

Finally, *optimal transportation based methods* [ZLZ05] aim at mapping the clusters with one another, and also at accommodating the case of soft clustering. These methods actually rely on the earth mover distance [RTG00]–a linear program which may be seen as a particular case of optimal transportation. In short, this LP involves the distances between the clusters representatives (centroids), and solves for the weight assigned to the match between any two clusters. This approach is powerful, as fractional cluster matching goes beyond the 1-to-1 greedy matching alluded to above. However, the involvement of cluster centroids masks individual contributions from the items themselves, so that the approach does not apply when commonalities between groups of clusters from F and F' are sought.

Naturally, when the two clusterings studied stem from two runs of the same algorithm (randomized or with different initial conditions), the previous quantities can be used to assess the stability of this algorithm [Lux10]. In particular, the minimal matching distance has been used to study the stability of k-means.

2.4.4 Ensemble clustering

The comparison of two clusterings is also related to the *ensemble clustering* problem, which aims to find a so-called median or consensus partition best summarizing an ensemble of partitions [Jai10]. Approaches either focus on the items clustered, of the clusterings themselves. For the former, [FJ02], the consensus clustering is derived from a partitioning (MST based) of the co-occurrence matrix–which measures how often two data points are clustered together. In [LZY05], the problem is reduced to finding a correspondence matrix mapping the labels of a clustering into those of another clustering. (The objective criterion involves the Frobenius norm of the difference between the membership matrices characterizing the two clusterings.) For the latter, the ensemble clustering problem can be phrased as a minimization problem on the space of clusterings endowed with a proper metric, yielding NP-complete problems [BL93]. In [Mei02], the mutual information between two clustering is used to compare two clusterings and also to define the functional specifying the consensus clustering. In [TJP05], the quadratic mutual information is used to define the consensus clustering.

2.5 Software for computational structural biology

2.5.1 A need for consensus

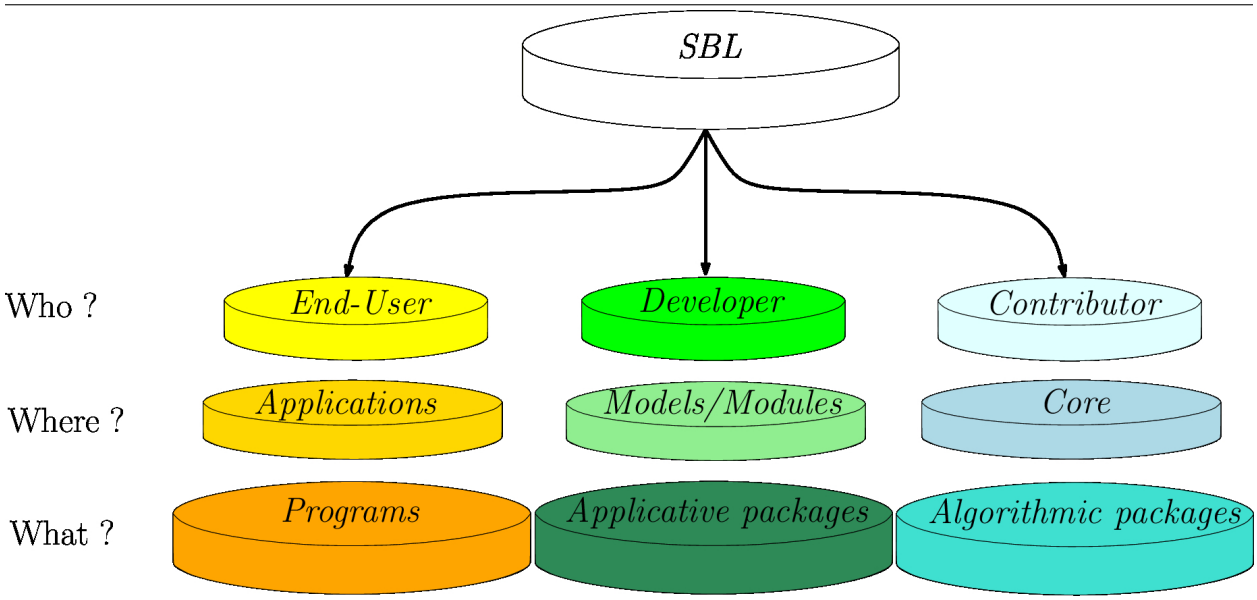
When designing software for structural biology, one is faced with many challenges, the first of which boils down to the lack of consensus in the community. As an example, take something as basic as a protein sequence: every sequence analysis program, every sequence database, comes with its own format. As a result, there are currently dozens of formats which represent sequences and various added annotations. Protein structure is no stranger to that rule. The PDB file format is the standard format to handle atom coordinates. Some efforts exist to handle PDB files in a standardized manner (Biopython, ESBTL, BiopLib) but most tools come packed with their own PDB file reader.

There exists a plethora of application software and libraries for structural bioinformatics. Over 350 are classified at `http://www.ks.uiuc.edu/Development/biosoftdb/biosoft.cgi`. This state of affairs is a consequence of the application driven mindset which dominates the field. Because of their enabling of notable advances in the fields of biophysics and computational biology, some large software platforms and libraries stand out as major scientific achievements. Noteworthy platforms include MODELLER, Rosetta, and the Integrative Modeling platform.

2.5.2 Software design for end-users and developers

Most main contributions to the realm of structural bioinformatics software focus on specific applications which require low-level structures and algorithms. Albeit usually being high quality applications, all of

Figure 2.13 The SBL is compartmentalized in focus areas for end-users, developers and contributors. For end-users, the SBL offers a number of high-level applications targeting specific biophysical problems. For high-level developers, the SBL offers specific biophysical models as well as inter connectable modules which perform specific computations. For low-level developers and contributors, the SBL provides core algorithmic packages targeted at specific tasks (CADS: Combinatorial Algorithms and Data Structures, GT: Computational geometry and topology, CSB: Computational Structural Biology, IO: Input/Output).



them fail to provide a framework for developers to design their own software solutions from these low-level building blocks. The Structural Bioinformatics Library aims at changing status quo by providing a modular framework matching the best standards in two directions. It is an ongoing endeavor within the A.B.S group (`https://team.inria.fr/abs/`). On the one hand, the SBL provides end-user applications with state of the art performances. On the other hand, the SBL allows access to the low-level structures and algorithms developed for these applications. The design matches that of comprehensive C++ libraries such as CGAL or boost.

End-users. For end-users, the SBL provides a number of applications each targeted at specific biophysical problems. In this effect, applications are organized in different packages:

- *Space filling models:* Applications dealing with molecular models defined by union of balls (Var der Waals or solvent accessible model).
- *Conformational analysis:* Applications dealing with molecular flexibility.
- *Large assemblies:* Applications dealing with macro-molecular assemblies.
- *Integrated analysis:* Applications combining several ingredients from the previous groups.
- *Data management:* Applications easing data management in large scale experiments.

Developers. For developers, the SBL provides Core packages, which correspond to the low-level units used to build the applications. They are organized in domains depending on their use:

- *CADS* for Combinatorial Algorithms and Data Structures.

- *GT* for Geometry and Topology
- *CSB* for Computational Structural biology
- *IO* for Input/Outputs

2.5.3 Modeling proteins

Another tough challenge one is faced with when designing software for structural bioinformatics is the variety of protein models.

A mix of geometry, topology, biophysics and biology. Proteins are indeed complex objects. Their description typically involves:

- Geometrical information: the coordinates of the atoms which can be Cartesian or internal.
- Topological information: the covalent bonds, or covalent structure.
- Biophysical annotations: the hierarchical organization of a protein (polypeptide chains, domains, SSE, atoms), and selected annotations.

The difficulty resides in the unification of all this information, as novel algorithms tend to exploit most of it concurrently.

2.6 Contributions and thesis overview

The contributions of this thesis are as follows.

2.6.1 Chapter 3: Molecular distances

We propose an elementary formula combining lRMSD measures, each associated with its own optimal rigid motion, so as in particular to assess molecular flexibility. With respect to the afore-discussed limitations of the lRMSD , our so-called *combined RMSD* has the following advantages:

- Flexibility is characterized by local structural alignments between structural motifs of the compared molecules, rather than with a global parameterization of motion.
- Flexibility can be assessed at multiple scales, by parameterizing the number of rigid motions used–this number is equal to the number of motifs mapped to one-another.
- The dependency of protein size is alleviated.

2.6.2 Chapter 4: Structural motifs

Consider two structures, typically two polypeptide chains, for which we wish to identify structurally conserved motifs. The structures may represent the same conformation of a given molecule, or the structures of two (homologous) polypeptide chains.

Our work is motivated by a refined analysis of flexible alignments, an inherently delicate problem since flexibility is inherently related to scale–smaller regions tend to be more rigid. In order to perform a multiscale analysis of structural motifs, we are guided by the following principles:

- A motif is defined by an alignment between two sets of a.a. of the same size.
- Motifs must inherently exhibit a multiscale structure, trading size for structural conservation.

- At any scale, motifs must be amenable to independent structural comparisons (using say lRMSD), as these lRMSD may be combined to assess more global deformations [CT18a].

To meet these requirements, we make several contributions:
1. A generic framework for multiscale structural alignments. First, we introduce a generic bootstrap framework to report structurally conserved motifs, based on (i) an initial alignment, and (ii) a topological analysis of so-called filtrations coding conserved distances in the structures. We provide four instantiations obtained by combining two seed alignment methods (`Kpax` [RGMV12], `Apurva` [AMDY11]) and two *filtrations*. The rationale for using two seed alignment is as follows: `Apurva` is a structural aligner based on contact maps, favoring long and *flexible* alignments [AMDY11]; `Kpax` is a structural aligner based on a geometric representation of the backbone, favoring a geometric measure known as the G-score [RGMV12].

The rationale for using filtrations is that they inherently accommodate a multiscale analysis of structural conservation. More precisely, filtrations, a concept from computational topology [EH10], are used to encode distance conservation at multiple scales. Practically, we use two filtrations. The first one, defined from distance difference matrices [HS93], favors distance conservation regardless of connectivity; the second one, defined from space filling diagrams [Ede95], favors spatial connectivity.

2. Motifs and multiscale analysis. We introduce an assessment of motifs based on Hasse diagrams to handle the multiscale nature of motifs, and Pareto fronts defined in the space (lRMSD , alignment size) of motifs. Using the latter, we investigate in particular the merits of the two filtrations used.

3. Motifs to qualify flexible deformations via combined lRMSD . We show that the optimal rigid motions associated to our motifs leverage combined lRMSD , a notion of *flexible* lRMSD mixing local (motif based) lRMSD into a global score assessing a global deformation [CT18a].

4. Motifs to seed iterative alignments. Iterative aligners consist in iteratively finding the alignment for fixed positions of the conformations and finding optimal superimposition given the alignment [BOPR03, RGMV12]. We show that our motifs can be used to seed iterative structural alignments, yielding alignments outperforming those of the seed aligners (`Apurva`, `Kpax`).

Validation: test set. We evaluate our contributions using a dataset of class II fusion proteins, which is especially interesting and challenging for two reasons: on the one hand, these proteins accomplish the same function; on the other hand, they harbor loose sequence identity, and loose global structural similarity.

Two important comments are in order. First, our generic method is not *stricto sensu* a direct competitor of flexible aligners. Indeed, a key goal is to dissect a seed alignment into motifs, so as to capture the various structural conservation scales it may contain. Second, our approach bears two major differences with the flexible alignment methods FATCAT [YG03] and the phenotypic plasticity measure (PPM) [CBZ08]. Both methods require the prior identification of rigid blocks. In [YG03], a flexible alignment is obtained using dynamic programming, by interleaving the blocks (called aligned fragment pairs) with connections defined by extensions, gaps or twists. In [CBZ08], a flexible alignment is then defined by a spanning tree connecting blocks, and the associated PPM score measures the similarity between nodes (blocks) and edges (connections between blocks). Once all possible blocks and topological edges connecting them have been computed, the optimal alignment is obtained from the A* algorithm. As opposed to these two methods, we do not rely on precomputed blocks, but identify them in a multiscale fashion; also, we perform a multiscale analysis of motifs, which are qualified in terms of Hasse diagrams and Pareto envelopes.

2.6.3 Chapter 4: Protein function prediction

We present a method performing a functional characterization of a set of proteins with low sequence identity and loose structural conservation, which delivers profile HMM that can be used to query sequence databases. Similarly to HMAPs [TXK$^+$03], our method combines structure and sequence information; yet, it bears major differences. First, the method relies on structurally conserved motifs that may span SSE and loops, as opposed to secondary and tertiary structure elements. Second, it involves a unique parameter, used to tune the stringency threshold of structural information termed relevant – see above our discussion on stringency

thresholds. Third, our structural motifs are used to produce profile profile HMM biased towards structurally conserved regions. These HMM are then utilized to query databases such as `UniProtKB`, in order to retrieve the sequences of proteins which may exhibit the function of interest.

2.6.4 Chapter 6: Clustering comparison

Rationale. Previous work has overlooked two issues. First, the comparison of two clusterings has been done globally, i.e. without providing insights on the mapping between clusters–if one omits the elementary matching case. Second, in assessing and comparing clusterings, the *scale* at which clusters merge has not been studied. Phrased differently and following [SG02]: "*In fact, the* `right` *number of clusters in a dataset often depends on the* `scale` *at which the dataset is inspected*". Indeed, VI or co-occurrence matrices which count the number of times two data points are clustered together do not provide such insights.

In this chapter, we fill this gap by studying the problem of grouping clusters into meta-clusters. To do so, we define the family-matching problem on the intersection graph G constructed from F and F'. A node of G represents a cluster, an edge between two nodes means that the intersection between the two corresponding clusters is not empty, and the weight of an edge is the number of items (elements) shared by the two clusters (that necessarily belong to different clusterings). The family-matching problem consists in computing disjoint subsets of nodes (clusters of clusters, or meta-clusters) such that **(i)** every such subset induces a sub-graph of G of diameter at most a given constant $D \geq 1$, and **(ii)** the number of items, for which the two clusters that contain it (in F and in F') are in a same meta-cluster, is maximum. This parameter corresponds to the score of a solution.

The constraint on the diameter D actually sheds light on previous work. The case $D = 1$ corresponds to previous work focused on 1-1 matchings. The case $D = 2$ solves the case for which one cluster of F corresponds to different smaller clusters of F' (and vice versa). We prove in Lemma 6.4 that the optimal score for $D = 2$ can be arbitrarily large compared to the optimal score for $D = 1$ – an incentive to introduce this diameter constraint. The case $D > 2$ (constant) deals with the case where different clusters of F correspond to different clusters of F' (and vice versa) but without a *good* matching between these clusters. In that case, the value of D is a measure of the complexity of the two clusters of clusters (the meta-cluster involving these two). Finally, when D is finite, it relates to previous work on cut problems on graphs [GH94, NI00, SV95, ZNI99]. This is not appropriate for clusterings comparison. Indeed, if the sub-graphs corresponding to meta-clusters have a finite but (too) large diameter, then we cannot *finely* describe the differences between the two clusterings.

Contributions. Our work, which investigates the relationship between two clusterings, shedding light on the way clusters from one have been merged / split / edited to define one clustering from the other, consists of the following contributions. In Section 6.1, we introduce a new combinatorial optimization problem on the intersection graph, namely the D-family-matching problem, so as to compare two clusterings. In Section 6.2, we prove that the problem is very hard to solve: NP-completeness results and unbounded approximation ratio of simple strategies. In Section 6.3, we design exact polynomial time dynamic programming algorithms for some classes of instances (trees, paths, cycles, graphs of maximum degree two). In Section 6.4, we describe efficient algorithms for general graphs, introducing a variant of the problem with spanning tree constraints. In Section 6.6, we illustrate the ability of our algorithms to identify relevant meta-clusters between a given clustering and an edited version of it, and compare our scores against the Variation of Information. In both cases, we show that parameter D yields insights on the *scale* at which clusters coalesce.

Due to the lack of space, all the details and proofs can be found in appendix.

2.6.5 Chapter 7: Software contributions

All of the contributions sketched above are made available as software packages in the `SBL` (`http://sbl.inria.fr`). On top of providing end-user applications, a number of Core packages allow developers to use the building blocks of these applications to design their own software. The main software contributions associated with this thesis are:

- An application computing the so-called combined RMSD .
- An application to identify *structural motifs*.
- An application to exploit *structural motifs* and build hybrid hidden Markov models.
- An application to compute the *D-family matching* for two clusterings.

Chapter 3

Combining lRMSD measures

The root mean square deviation (RMSD) and the least RMSD are two widely used similarity measures in structural bioinformatics. Yet, they stem from global comparisons, possibly obliterating locally conserved motifs. We correct these limitations with the so-called *combined RMSD* , which mixes independent lRMSD measures, each computed with its own rigid motion. The combined RMSD can be used to compare (quaternary) structures based on motifs defined from the sequence (domains, SSE), or to compare structures based on structural motifs yielded by local structural alignment methods.

We illustrate the benefits of combined RMSD over the usual RMSD on three problems, namely (i) the analysis of conformational changes based on combined RMSD of rigid structural motifs (case study: a class II fusion protein), (ii) the calculation of structural phylogenies (case study: class II fusion proteins), and (iii) the assignment of quaternary structures for hemoglobin. Using these, we argue that the combined RMSD is a tool a choice to perform positive and negative discrimination of degree of freedom, with applications to the design of move sets and collective coordinates.

Combined RMSD are available within the Structural Bioinformatics Library (`http://sbl.inria.fr`).

3.1 Method: combining independent lRMSD measures

3.1.1 Structures, alignments, and motifs

Let A and B represent either two distinct structures or two conformations of the same molecule or complex. Assuming these are proteins, we denote their number of amino acids n_A and n_B, respectively. Our structural comparisons are based on the coordinates of *particles*. In comparing two conformations of the same molecule, the particles may refer to all atoms, all heavy atoms only, or C_α atoms; for two distinct molecules, we assume the particles are C_α atoms.

Our comparisons shall use variants of the lRMSD, whose calculation requires an alignment of the two structures to be compared. We first recall:

Definition. 3.1. *Consider two sequences of length n_A and n_B. An alignment of length $N \leq \min(n_A, n_B)$ is defined by two sets of indices $I = (i_1, i_2, \ldots, i_N)$ with $1 \leq i_1 < i_2 < \cdots < i_N \leq n_A$ and $J = (j_1, j_2, \ldots j_N)$ with $1 \leq j_1 < j_2 < \ldots j_N \leq n_B$. An* alignment *is specified by the perfect matching $\{(i_1, j_1), \ldots, (i_N, j_N)\}$.*

Once the two structures have been aligned, abusing notations, we may reduce them to the two ordered point sets $A = \{a_i\}_{i=1,\ldots,N}$ and $B = \{b_i\}_{i=1,\ldots,N}$.

Structural motifs. To *localize* structural comparisons, we assume that *structural motifs* have been identified. Practically, two types of motifs may be used: features of proteins (domains, SSE), or motifs yielded by local structural alignment methods–see Introduction. More formally, we define:

Definition. 3.2. *Consider two structures A and B. A* motif *is a pair of set of particles $M^{(A)} \subset A$ and $M^{(B)} \subset B$ of the same size, together with an alignment between them.*

The alignment allows computing lRMSD$(M^{(A)}, M^{(B)})$. The fact that motifs may overlap calls for the following processing.

Motif graph for overlapping motifs. When several motifs exist for two structures, an important question is to handle them coherently. Since motifs may overlap, we define (Fig. 3.1):

Definition. 3.3. *(Motif graph) The* motif graph *of a list of motifs $\{(M_i^{(A)}, M_i^{(B)})\}_{i=1,\dots,p}$ is defined as follows: its node set is the union of the particles A and B; its edge set is the union of two types of edges:*

- matching edges*: the edges associated with the matchings defined by the motifs. NB: such edges are counted without multiplicity, that is, a matching edge present in several motifs is counted once.*
- motif edges*: edges defining a path connecting all amino acids in a motif.*

Consider a connected component (c.c.) of the motif graph. Restricting each c.c. to each structure yields two subgraphs. The set of all such subgraphs is denoted $\{C_i^{(A)}, C_i^{(B)}\}_{i=1,\dots,m}$.

The following observations can be made (Fig. 3.1). A subgraph ($C_i^{(A)}$ or $C_i^{(B)}$) may not be connected. Also, the motif graph does not define, in general, a matching between the vertices associated with particles. More precisely, the *multiplicity* of a particle in a motif graph is defined as the number of edges incident to this particle. Despite these features, as we shall see below, the edges connecting the particles from $C_i^{(A)}$ and $C_i^{(B)}$ can be used to define a variant of the classical lRMSD.

3.1.2 Vertex weighted and edge weighted lRMSD

We introduce generalizations of the lRMSD and RMSD, using connected components of the motif graph. Before presenting these, we recall the construction of the weighted lRMSD (Def. 3.4).

Vertex weighted lRMSD : lRMSD$_{\mathbf{vw}}$. Consider two point sets $A = \{a_i\}$ and $B = \{b_i\}$ of size N. Also consider a set of positive weights $\{w_i\}_{i=1,\dots,N}$, meant to stress the importance of certain particles. The weighted RMSD reads as:

$$\mathrm{RMSD_w}(A, B) = \sqrt{\frac{1}{\sum_i w_i} \sum_{i=1,\dots,N} w_i \left\| a_i - b_i \right\|^2} \tag{3.1}$$

Let g a rigid motion from the the special Euclidean group $SE(3)$. To perform a comparison of A and B oblivious to rigid motions, we use the so-called *least RMSD* [Kab76]:

Definition. 3.4. *The* vertex weighted *lRMSD is defined by*

$$lRMSD_{vw}(A, B) = \min_{g \in SE(3)} RMSD_w(A, g(B)). \tag{3.2}$$

The rigid motion yielding the minimum is denoted g^{OPT} (A,B) or g^{OPT} for short.
The weight of the $lRMSD_{vw}$ is defined as $W_{vw}(A, B) = \sum_i w_i$.

Note that the celebrated lRMSD is the particular case of the previous with unit weights:

$$\mathrm{lRMSD}(A, B) = \mathrm{lRMSD_{vw}}(A, B) \text{ with } w_i \equiv 1. \tag{3.3}$$

Denote R the sought rotation matrix [AHB87] and C the covariance matrix

$$C = \sum_i w_i b_i a_i^{\mathsf{T}}. \tag{3.4}$$

Upon centering the data, computing the lRMSD amounts to maximizing Trace(RC) [AHB87, LAT10], a calculation which can be done with an SVD calculation.

Edge weighted lRMSD : $\mathbf{lRMSD_{ew}}$. Consider now the case where motifs have been defined for the two structures A and B. We wish to compare A and B exploiting the information yielded by the connected components of the motif graph (Def. 3.3). Consider the i-th c.c. of the motif graph. Let e_i be the number of matching edges of this c.c. As usual, let $g(b_j)$ the position of atom b_j from $C_i^{(B)}$ matched with atom a_j from $C_i^{(A)}$, upon applying a rigid motion g. We define:

Definition. 3.5. *The* edge weighted $\mathrm{lRMSD_{ew}}$ *of the i-th c.c. of the motif graph is defined by*

$$lRMSD_{ew}(C_i^{(A)}, C_i^{(B)}) = \min_{g \in SE(3)} \sqrt{\frac{1}{e_i} \sum_{j=1}^{e_i} \|a_j - g(b_j)\|^2} \tag{3.5}$$

The rigid motion yielding the minimum is denoted g_i^{OPT}.
The weight of the $lRMSD_{ew}$ is defined as $W_{ew}(C_i^{(A)}, C_i^{(B)}) = e_i$.

To compute this quantity, we proceed as for the $\mathrm{lRMSD_{vw}}$, except that the covariance matrix from Eq. (3.4) is now obtained by summing over edges of the bipartite graph rather than on vertices. We also make the following

Observation. 1. *If the motif graph defines a perfect matching between the particles of a connected component, then $lRMSD_{vw} = lRMSD_{ew}$ for that component.*

3.1.3 Combined RMSD : $\mathbf{RMSD_{Comb.}}$

Since the $\mathrm{lRMSD_{ew}}$ values are defined for each c.c. of the motif graph, we combine them to obtain a comparison of A and B.

Denote m the number of connected components of the motif graph and let $N_e = \sum_i e_i$. The edge weighted lRMSD can be combined into the following *edge-weighted combined RMSD*

$$\mathrm{RMSD_{Comb.}}(A,B) = \sqrt{\frac{1}{N_e} \sum_{i=1}^{m} \sum_{j=1}^{e_i} \left\| a_j - g_i^{\mathrm{OPT}}(b_j)) \right\|^2} = \sqrt{\sum_{i=1}^{m} \frac{e_i}{N_e} \mathrm{lRMSD}_{\mathrm{ew}}^2(C_i^{(A)}, C_i^{(B)})} \tag{3.6}$$

It is easily checked that the previous is a particular case of the following combined RMSD, which mixes individual lRMSD, be they vertex weighted ($\mathrm{lRMSD_{vw}}$) or edge weighted ($\mathrm{lRMSD_{ew}}$):

Definition. 3.6. *Consider two structures A and B for which non-overlapping regions $\{C_i^{(A)}, C_i^{(B)}\}_{i=1,\dots,m}$ have been identified – Def. 3.3. Assume that a lRMSD has been computed for each pair $(C_i^{(A)}, C_i^{(B)})$. Let w_i be the weights associated with an individual lRMSD. The* combined RMSD *is defined by*

$$RMSD_{Comb.}(A,B) = \sqrt{\sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} lRMSD^2(C_i^{(A)}, C_i^{(B)})}. \tag{3.7}$$

The following bounds are straightforwards convexity inequalities (proof in SI Sec. A.1.1):

Observation. 2. *The* combined RMSD *satisfies the following upper and lower bounds:*

$$RMSD_{Comb.}(A,B) \geq \sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} lRMSD(C_i^{(A)}, C_i^{(B)}). \tag{3.8}$$

Let $l_{min} = \min_i lRMSD(C_i^{(A)}, C_i^{(B)})$ and $l_{max} = \max_i lRMSD(C_i^{(A)}, C_i^{(B)})$. One has

$$RMSD_{Comb.}(A,B) \leq \sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} lRMSD(C_i^{(A)}, C_i^{(B)}) + 2\Big(\sqrt{\frac{l_{min} + l_{max}}{2}} - \frac{\sqrt{l_{min}} + \sqrt{l_{max}}}{2}\Big). \tag{3.9}$$

Remark 3.1. *Equation (3.7) defines a RMSD rather than a lRMSD. To see why, observe that Eq. (3.2) defines a number which is the minimum of a quadratic optimization problem involving Eq. (3.4). Instead, Eq. (3.7) defines a number obtained by mixing solutions of such problems for connected components of the motif graph.*

Remark 3.2. *Combined RMSD can be used hierarchically. Consider two conformations of a a complex containing say two chains, each decomposed into motifs. In a first step, the combined RMSD exploiting the motifs can be used to compare the two instances of each chain across the two complexes. In a second steps, the combined RMSD of these combined RMSD can be computed. We illustrate this strategy to provide insights on novel conformations of quaternary complexes of hemoglobin in section 3.3.3.*

3.2 Implementation

All methods are available in the Structural Bioinformatics Library (`http://sbl.inria.fr`, [CD17]).

Three executables implementing the methods (`sbl-rmsd-flexible-proteins.exe`, `sbl-rmsd-flexible-conformations.exe`, `sbl-rmsd-flexible-motifs.exe`) are provided in the package Molecular_distances_flexible (`https://sbl.inria.fr/doc/Molecular_distances_flexible-user-manual.html`). Additional details are provided in SI Section A.1.2.

Also of particular interest is the Structural_motifs package (see `https://sbl.inria.fr/doc/Structural_motifs-user-manual.html`), which makes available various methods to compute structural motifs [CT18b].

3.3 Results

We illustrate insights yielded by the combined RMSD, which are out of reach for the classical lRMSD.

3.3.1 Assessing conformational changes: the example of a class II fusion protein

Biological context. Recall that the central data set used in this thesis is a set of class II fusion proteins (see Chapter 1). In the sequel, we use the combined RMSD to illustrate the conformation changes undergone by a prototypical class II fusion protein, from the tick-borne encephalitis virus. The ectodomain of this protein was crystallized both in soluble form (PDB: 1SVB, [RHM$^+$95], 395 residues) and in postfusion conformation (PDB: 1URZ, [BSA$^+$04], 400 residues) (Fig. 3.2). We use structural motifs computed using a method reported in Chapter 4 to demonstrate the $\text{RMSD}_{\text{Comb.}}$ in such a setting (RMSD_MODE_MOTIF, SI Sec. A.1.2).

Results. At first glance, a global lRMSD calculation yields lRMSD = $11.32\mathring{A}$. To further assess the presence of rigid motifs moving relatively to one another, we computed motifs (Def. 3.2) using the method presented in the companion paper [CT18b]. In a nutshell, this methods identifies structurally conserved motifs, each such region being a connected domain whose connectivity is ensured by pairs of atoms whose distance is conserved between the structures studied. (We note in passing that this strategy departs from the classical approaches targeting quasi-isometric motifs via the calculation of cliques [MDAY10].)

For the tick-borne encephalitis virus, the method yields a total of $p = 31$ structural motifs distributed in $m = 2$ connected components ($|C_1| = 109$, $|C_2| = 51$; Fig. 3.2). Finally, in computing the combined RMSD between the two connected components (Def. 3.3), one obtains $\text{RMSD}_{\text{Comb.}} = 2.50\mathring{A}$.

As illustrated by this example, the ability to identify structurally conserved motifs and to combine their lRMSD makes a dramatic difference in the overall comparison of two conformations. This problem is well known e.g. for the analysis of molecular dynamics trajectories, where erroneous assignments of structures to the same clusters / meta-stable states may jeopardize free energy calculations [NAKH14].

3.3.2 Building a phylogeny from structural data: the example of class II fusion proteins

Biological context. Following on the previous example, we now illustrate the interest of combined RMSD to build phylogenies of class II fusion proteins. As a dataset, we use 6 monomers of class II fusion proteins in their post-fusion conformation (Fig. 3.3). In order to assess the merits of the usual lRMSD and that of the combined RMSD to build phylogenies via dendograms, we carry out the analysis at two levels, namely using whole domains and the 22 motifs defined by SSE (Fig. 3.3(b)). (This corresponds to the setting RMSD_MODE_SEQ for homologous proteins, see SI Section A.1.2.)

Results. Out of the six structures and for each pairwise comparison–15 of them, we used the executables provided in the `SBL` to compute: (1) a structural alignment using the `Apurva` algorithm [AMDY11]. This yields a lRMSD (Table A.1), (2) the $\text{RMSD}_{\text{Comb.}}$ upon processing the regions defined by the three domains (Table A.2), (3) the $\text{RMSD}_{\text{Comb.}}$ upon processing the regions defined by the SSE labels (Table A.3). From the distance matrices displayed in the aforementioned tables, we perform three complete linkage hierarchical clusterings (Fig. 3.4). Each level of comparison conveys different information.

- **Full structure.** This level of information is enough to classify the two flaviviruses together. Out of the 6 structures described in Fig. 3.3, we know that 2 (HRV-Hanta. and RBV-Rubi.) have a domain swap. This adds considerable noise to the clustering.
- **Domains.** Here the two flaviviruses as well as the two bunyaviruses are clustered together, regardless of the domain swap.
- **SSE.** The two flaviviruses as well as the two togoviruses are clustered together, regardless of the domain swap.

To conclude, while the global lRMSD falls short from providing a satisfactory classification, combined RMSD fixes this limitation. One can further use the individual lRMSD value on a per domain basis to illustrate the structural diversity of these domains (SI Fig. A.2).

3.3.3 Assigning quaternary structures: the example of hemoglobin

Biological context. Hemoglobin is the gas transporting metalloprotein in mammals. In humans, the predominant form of hemoglobin has a quaternary structure based on four subunits (SI Fig.A.3), namely two α chains (141 amino acids each), and two chains β (146 amino acids each). Each subunit contains a heme group consisting of a charged iron (Fe) ion held in an heterocyclic ring called porphyrin. It had long been believed that binding of O2 to one monomer triggered the transition of the tetramer from the tense (T) state to the relaxed (R) state, a mechanism at the core of cooperative binding [Per73]. Along this process, one pair of subunits rotates of an angle $\sim 15\,\text{deg}$ about the other ([BC79] and SI Fig.A.3). However, the mechanism is more complex, and a third composite quaternary state, usually denoted R2 or Y was discovered long ago [SLC91]. Based on these reference structures, a combination of various biophysical experiments [EHHM06] and macroscopic analysis (using angles and distances to qualify the quaternary structures) [DCJ11] had provided insights on gas binding by hemoglobin. Recently, these models were questioned by crystal structures which revealed that each tetramer captured hemoglobin in three quaternary conformations [SSTP14]. (NB: the three crystal structures are: half-liganded with phosphate (HL+ , PDB: 4N7P), half-liganded without phosphate (HL- , PDB: 4N7O), and fully water-liganded met-hemoglobin with phosphate (FL+ , PDB: 4N7N).)

Importantly, each new crystal was found to contain 3 new quaternary conformations denoted A, B, C. An assignment procedure based on difference distance matrices of the $\alpha_1\beta_2$ subunits using R as a reference, (SI Sect. A.1.4) supported the following: the A, B, C states respectively assumes states R2, R, and T. Moreover, visual inspection of the data [SSTP14, Fig. 2] shows that upon alignments, the tetramers tagged A or B superimpose almost perfectly, while those tagged C exhibit less coherent SSE (in particular helices from the FL+ crystal.

Results. As recalled above, assignment of quaternary states was mainly done so far using four rigid-body parameters related to the twofold symmetry [DCJ11], or using difference distance matrices with R as a reference state [SSTP14]. We revisit this problem, performing a complete hierarchical clustering of dimers using two combined RMSD : a combined RMSD mixing the lRMSD of the two chains; a combined RMSD mixing two combined RMSD –one for the 7 helices of chain α and one for the 8 helices of chain β. We use these combined RMSD to perform a hierarchical clustering (single linkage) of all combinations of α and β chains. Of particular interest are the clusterings of $\alpha_1\beta_1$ and $\alpha_1\beta_2$ dimers (Fig. 3.5, SI Fig. A.4, SI Fig. A.5), which evidence three interesting facts.

First, let us consider the ability to group coherently the states A, B, C, versus states $R2, R$ and T. The combined RMSD at the chain level is unable to separate the aforementioned conformations A, B, C (Fig. 3.5(Left), SI Fig. A.4(Left)). On the other hand, the combined RMSD at the SSE level does retrieve the A, B, C conformations and groups them coherently with the $R2, R$ and T states (Fig. 3.5(Right), SI Fig. A.4(Right)).

Second, it confirms the information yielded by visual inspection, according to which conformations C of the tetramer are less coherent. Indeed, the hierarchical clustering isolates conformation C from the FL+ crystal (PDB: 4N7N). It also singles out HL- [A], which is less coherent with the other two [A] but also with R and $R2$ (SI Tables A.4 and A.5). This somewhat mitigates the analysis from [SSTP14], where HL- [A] is reported as a relaxed state between R and R2.

Third, the $\text{RMSD}_{\text{Comb.}}$ values required to form the clusters are tighter than those corresponding to RMSD values of clusters formed using the aforementioned angles and distances [DCJ11]. While a much larger dataset was used in this latter paper, the mean values reported for lRMSD between R2, R and T states are 0.40, 0.43 and 0.36, against worst case values $\sim$ 0.27, 0.27 and 0.37 in our case.

Summarizing, the combined RMSD at the SSE level provides insights on the assignment of quaternary structures of hemoglobin, without using any reference state. Since SSE characterize quaternary structures, one can also assess the conformational changes undergone by helices in conjunction with the heme group (SI Fig. 3.6).

3.4 Discussion and outlook

The combined RMSD mixes independent lRMSD measures, each computed with its own rigid motion, therefore avoiding the global parameterization of conformational changes undergone by structures. Moreover, it can be computed at multiple scales, namely to compare secondary or tertiary structures based on structural motifs defined from the sequence or local structural alignments, or to compare quaternary structures from the same motifs. Finally, combined RMSD can be cascaded, so as for example to compare quaternary structures based on motifs defined from SSE elements. The notion of scale is in fact central to combined RMSD : on the one hand, the lRMSD computed for a whole structure typically yields an average signal shadowing the phenomenon scrutinized; on the other hand, computing a combined RMSD for too small structural elements would yield small values void of significance.

We illustrated the interplay between combined RMSD and pertinent scales on three non trivial examples, namely the analysis of large conformation changes, the design of phylogenies based on structural comparisons, and the identification of the quaternary structures of hemoglobin.

These examples may be discussed in the context of dynamical analysis of molecular machines, using the concepts of negative and positive discrimination of degrees of freedom. To articulate these notions, recall that two cornerstones of molecular simulations are move sets and collective coordinates. Move sets, on the one hand, are used in Monte Carlo methods and variants to generate conformations which are diverse and low in energy. Collective coordinates, on the other hand, are key to explore transition paths (and discover transient conformations), and compute free energy landscapes. Importantly, both concepts require understanding which degrees of freedom (dof) are key to account for the conformational changes studied.

To bridge the gap between move sets, collective coordinates, structural motifs, and combined RMSD, let us reconsider the scrutinized structures.

One the one hand, the example of the class II fusion protein undergoing a conformational change illustrates the notion of negative discrimination of dof. Indeed, the combined RMSD of the identified motifs being extremely small while the global lRMSD is large shows that in studying conformational changes between the two conformations studied, one can focus on those dof of atoms outside the motifs. In a sense, the combined RMSD rules out the dof of the motifs it qualifies.

On the other hand, the ability to cluster and qualify the quaternary structures of hemoglobin illustrates the notion of positive discrimination of dof. Indeed, while the global lRMSD does not yield any information, the restriction of lRMSD calculation to SSE, and the combination of the obtained values, yields valid biophysical classification. In other words, the combined RMSD positively discriminates the dof of the motifs it qualifies, calling for further studies to unveil the mechanism scrutinized. Such studies may focus on static analysis of crystal structures (detection of biophysical commonalities, formation/destruction of salt bridges, helix-to-coil transitions, etc). But they may also be dynamic, as the dof identified by may be targeted by complex move sets boosting the identification of structural intermediates.

Our methods to compute structural motifs and combined RMSD are made available within the Structural Bioinformatics Library (`http://sbl.inria.fr`). We anticipate that these tools will prove pivotal to conduct a wide array of structural analysis, both on static and dynamic structure of macro-molecules and their complexes.

3.5 Artwork

Figure 3.1 A motif graph (Def. 3.3). A toy system with two structures A and B involving 6 and 5 particles–say C_α-s, respectively. There are three motifs, namely $\{(M_1^{(A)}, M_1^{(B)}), (M_2^{(A)}, M_2^{(B)}), (M_3^{(A)}, M_3^{(B)})\}$. Motif edges are vertical edges connecting the particles; matching edges connect particles from the two structures. The three motifs induce two connected components, respectively containing 4 and 2 matching edges.

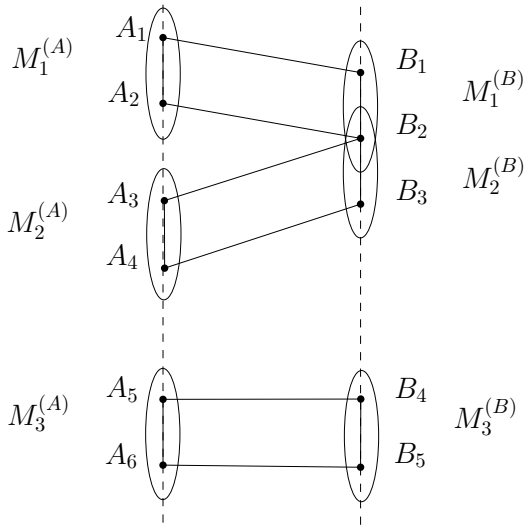
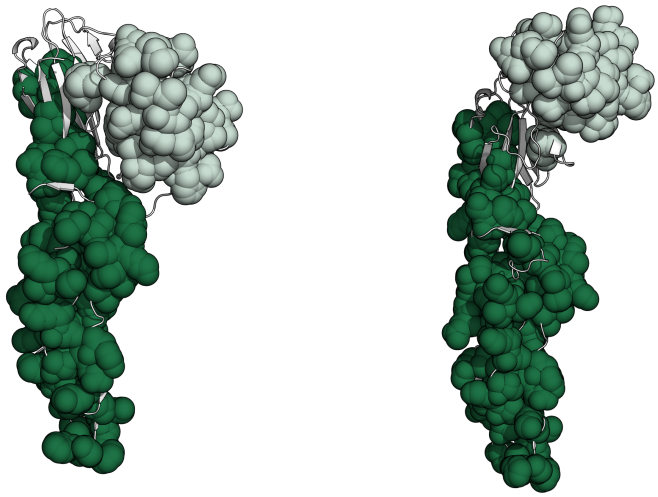


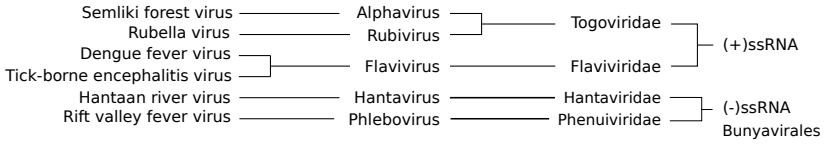
Figure 3.2 $\text{RMSD}_{\text{Comb.}}$ on overlapping structural motifs impervious to conformational changes: example on a class II fusion protein in soluble and post-fusion conformation. We display the two connected components, composed by the 31 structural motifs found by our method [CT18b]. Most of the motifs overlap, which justifies a definition for overlapping motifs (Def. 3.3).



lRMSD	$\text{RMSD}_{\text{Comb.}}$
$11.32\mathring{A}$	$2.50\mathring{A}$

Figure 3.3 Class II fusion structures. **a)** Taxonomy of structures used in this study. **b)** Breakdown of the structures used in this study, as well as their domain and label labels. **c)** Domain decomposition of DFV-Flavi..

a)



b)

Class II fusion proteins		
Virus	Acronym	PDB code
Semliki Forest virus	SFV-Alpha.	1RER
Tick-borne encephalitis virus	TBEV	1URZ
Dengue fever virus	DFV-Flavi.	1OK8
Rift valley fever virus	RVFV-Phlebo.	6EGT
Rubella virus	RBV-Rubi.	4ADI
Hantaan river virus	HRV-Hanta.	5LJX
Domain SSE		
Domain label	SSE labels	
D1	B0,C0,D0,E0,F0,G0,H0,I0	
D2	a, b, c , d, e, f, g, i, j	
D3	A, B, C, E, F	

c)

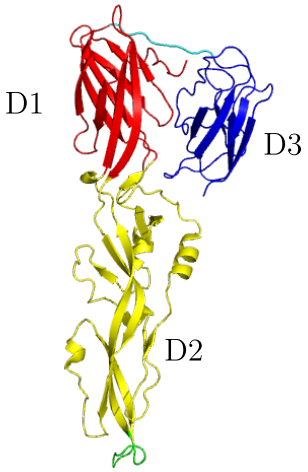


Figure 3.4 $RMSD_{Comb.}$ sharpens hierarchical clustering obtained for class II viral fusion proteins. Complete linkage hierarchical clustering of the structures defined in Fig. 3.3. **(Left)** Clustering obtained upon processing distances from Tab. A.1. Global lRMSD after aligning structures with the `Apurva` algorithm. **(Center)** Clustering obtained upon processing distances from Tab. A.2. $RMSD_{Comb.}$ using domains I, II and III. **(Right)** Clustering obtained upon processing distances from Tab. A.3. $RMSD_{Comb.}$ using motifs corresponding to SSE.

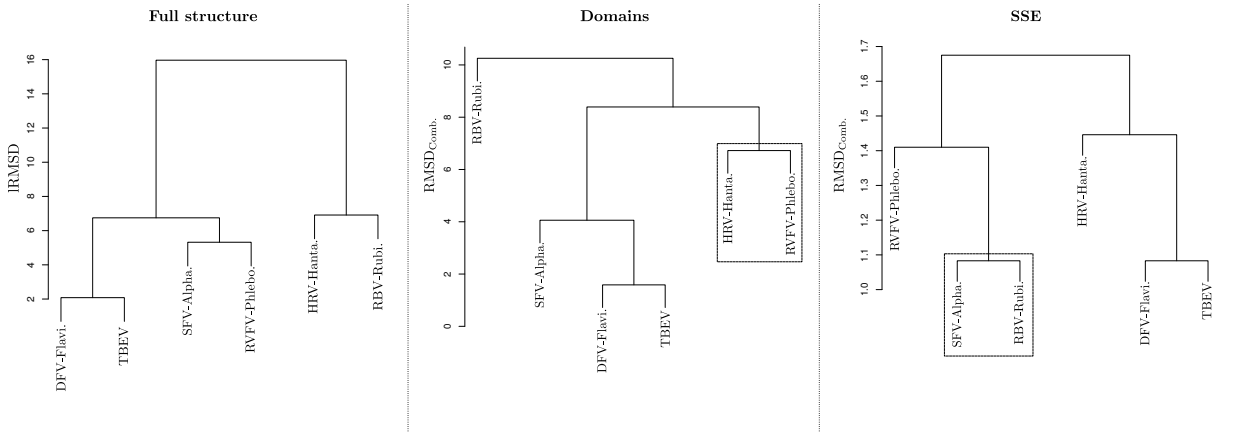


Figure 3.5 Assigning quaternary structures of hemoglobin using $\alpha_1\beta_1$ dimers. The goal is to check which similarity measures allow one to cluster coherently the newly reported conformations A, B, C of hemoglobin tetramers ([SSTP14] and Sec. 3.3.3.), assumed to adopt quaternary structures corresponding to the R2, R and T states. The displayed hierarchical clusterings were built using the single linkage scheme. **(Left)** Using $\text{RMSD}_{\text{Comb.}}$ combining the lRMSD of the two chains α_1 and β_1. The hierarchical clustering obtained does not cluster coherently states A, B, C, and does not provide a coherent clustering with states R2, R and T either. **(Right)** Using $\text{RMSD}_{\text{Comb.}}$ combining the $\text{RMSD}_{\text{Comb.}}$ of the two chains α_1 and β_1, the former (resp. latter) based on the 7 (resp. 8) lRMSD between its helices. The clusters of conformations A, B and to a lesser extent C are well formed and coherent with the R2, R and T states.

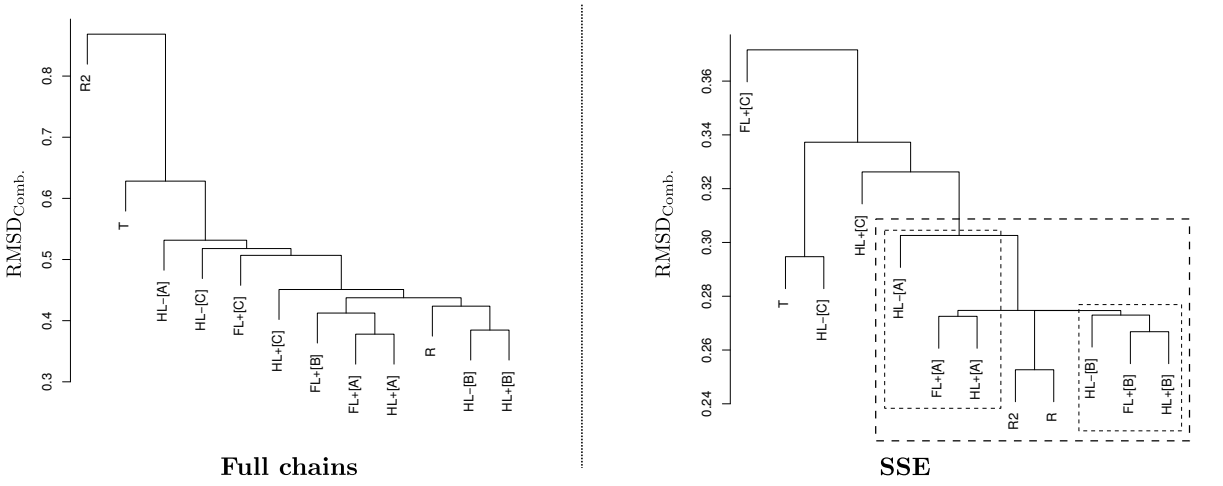
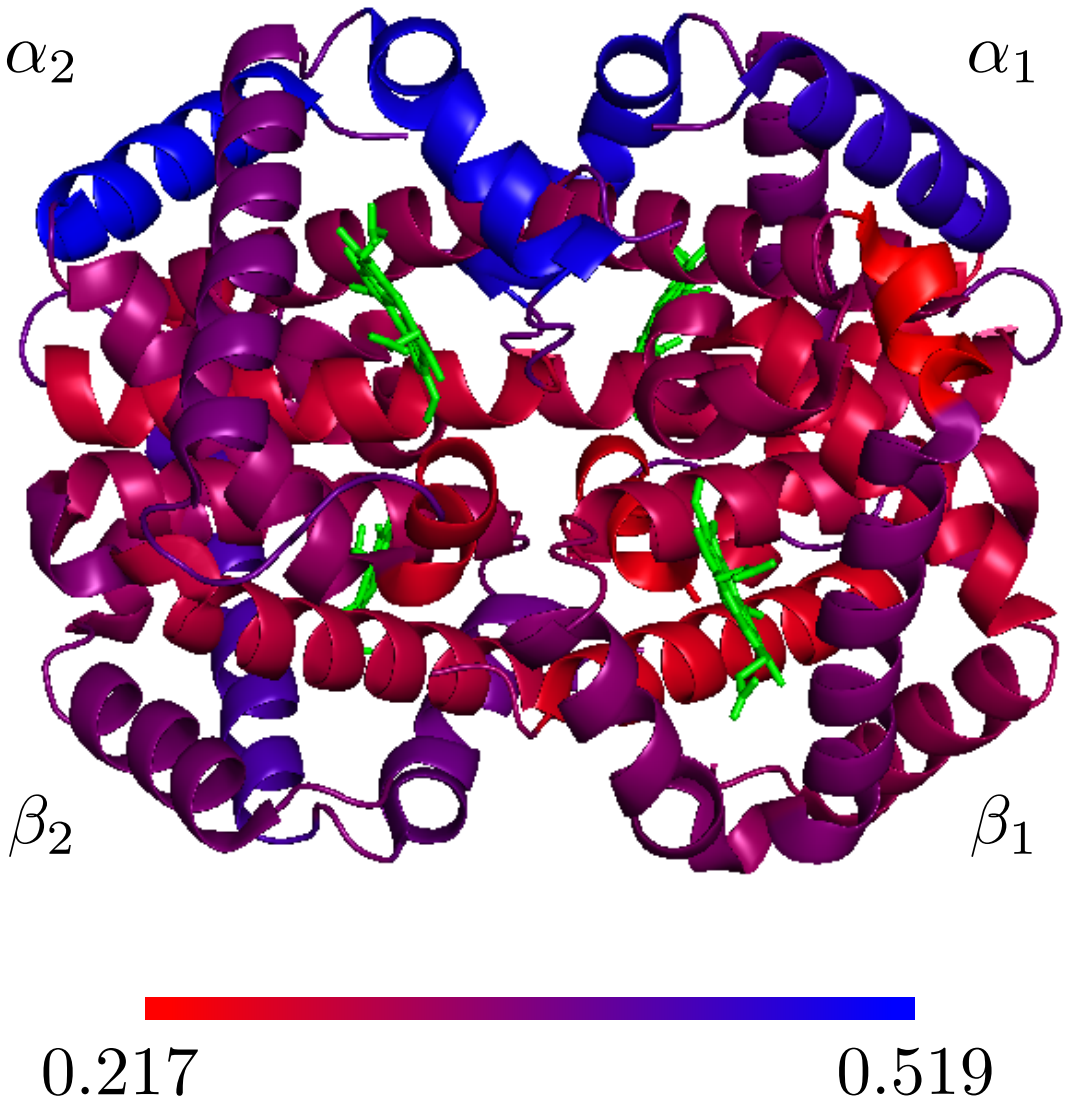


Figure 3.6 Structural conservation of hemoglobin. The α and β chains were respectively decomposed into 7 and 8 helices (Main text). For each helix, all pairwise lRMSD were computed using the 12 structures. Each helix was then color coded according to the gradient indicated. Visualization done with T conformation (pdbid: 2dn2).



Chapter 4

Multiscale analysis of structurally conserved motifs

The previous chapter introduced the notion of *combined RMSD*. This new distance measure for protein structures uses smaller components of the whole structure – which we call *structural motifs* – to perform a finer grain comparison. *Structural motifs* can be any number of things, from secondary structure elements to whole protein domains or anything in between. Finding such motifs is therefore a natural question. In the following, we share five contributions to the problem of reporting structurally conserved motifs shared by two structures (homologous proteins, conformations) undergoing conformational changes:

- We introduce a generic framework reporting structurally conserved motifs, based on (i) an initial a.k.a. seed alignment, and (ii) a topological analysis of so-called filtrations coding conserved distances in the structures. We provide four instantiations obtained by combining two seed alignment methods (`Kpax` [RGMV12], `Apurva` [AMDY11]) and two filtrations (defined from distance difference matrices and space filling diagrams).

- We show that our methods code structural conservation in a hierarchical way, via Hasse diagrams, as opposed to classical alignment methods.

- We asses our methods using Pareto fronts defined in the space (lRMSD, alignment size) of motifs. We show that the two filtrations yield different motifs.

- We show that the optimal rigid motions associated to our motifs leverage combined lRMSD, a notion of *flexible* lRMSD mixing local (motif based) lRMSD into a global score assessing a global deformation.

- We show that our motifs can be used to seed iterative structural alignments.

Illustrations are provided on a test set of class II fusion proteins.

We anticipate that our method will prove instrumental to perform structural comparisons, and assess molecular deformations based on combined RMSD of motifs.

Our method is available within the Structural Bioinformatics Library (`http://sbl.inria.fr`).

4.1 Method: extracting structural motifs

Our method to detect structural motifs mixes ingredients from structural alignments, graph theory, computational geometry (space filling diagrams), and computational topology (filtrations, persistence diagrams, Betti numbers). Readers not familiar with these fields are referred to SI section B.2.

4.1.1 Structural motifs

Let A and B be the two structures to be compared. Without loss of generality, we assume that the particles considered are C_α atoms, denoted $A = \{a_i\}$ and $B = \{b_i\}$. We define:

Definition. 4.1. *A* motif *shared by A and B is a pair of set of particles $M^{(A)} \subset A$ and $M^{(B)} \subset S_B$ of the same size, together with an alignment, that is a one-to-one correspondence between their a.a..*

The alignment allows computing the least RMSD , lRMSD(M_A, M_B). We further compare this quantity to that of the structures themselves:

Definition. 4.2. *(least RMSD ratio) Consider two structures A and B, and assume that a structural alignment between them has been computed, together with the corresponding lRMSD(A, B). The* least RMSD ratio *of the motif is defined by:*

$$r_{lRMSD}(M_A, M_B) = lRMSD(M_A, M_B)/lRMSD(A, B). \tag{4.1}$$

The sets M_A and M_B are called structural motifs *provided that $|M_A| = |M_B| \geq \tau_{MS}$ and $r_{lRMSD}(M_A, M_B) \leq \tau_/$, for appropriate thresholds $\tau_{MS}(> 0)$ and $\tau_/(\in (0, 1))$.*

Remark 4.1. *We note that the alignment used in Def. 4.1 depends on the context–homologous proteins versus conformations. In the former case, we use two methods respectively favoring long and flexible alignments (*`Kpax`*, [RGMV12], and backbone geometry (*`Apurva`*, [AMDY11]). In the latter case, the alignment is the trivial identity alignment. That is, we retain the a.a. common to both conformations–as some may be missing e.g. loops in crystal structures.*

Remark 4.2. *Motifs are local structural patterns. They can be combined to compare whole structures using the notion of combined $RMSD_{Comb.}$, as recalled in SI Section B.2 and [CT18a].*

4.1.2 Structural motifs from alignments: overview

Let us first convey the intuition of our method to compare two structures A and B. For the time being, consider the simple case where the two structures correspond to two conformations of a molecule consisting of say two domains connected by a flexible linker. If the relative position of the two domains changes in switching from one conformation to the other, the only conserved distances are those involving any two C_α from the same domain. Assume that we have computed a score assessing the conservation of the distance between two C_α. A natural strategy to identify the two domains therefore consists in incrementally processing scores, and maintaining the connected components linking the a.a. involved in the scores, using a union-find data structure [CLRSon]. We implement this idea via a four step method (Fig. 4.1):

Step 1: Computing the seed alignment and its scores. We first compute a *seed* structural alignment between the two structures A and B. If the two structures are two conformations of the same molecules, the alignment is trivially the identity. We then compute the distance difference matrix associated with the alignment, so as to qualify the conservation of distances between C_α between the two structures.

Step 2: Building the filtration and its persistence diagram. We build the conserved distances and space filling diagram filtrations already mentioned, together with the associated persistence diagrams, denoted PD_A and PD_B for the two molecules. Points on these PD correspond to regions in the structures.

Step 3: Computing structural motifs with bootstrap. We identify relevant pairs (a, b) of points, with $a \in \mathrm{PD}_A$ (resp. $b \in \mathrm{PD}_B$). We compute a structural alignment involving the connected components of the sublevels sets associated with these points, from which we compute a structural alignment and retrieve structural motifs complying with Eq. (4.1).

Step 4: Filtering structural motifs. Motifs are filtered to get rid of redundancy (inclusion), and to retain those which are statistically significant.

We now detail these steps.

4.1.3 Step 1: Computing the seed alignment and its scores

Consider a structural alignment between the two structures A and B. We denote $d^A_{i,j}$ the distance between the C_α carbons of indices i and j on chain A, and likewise on chain B. These quantities are used to define the *distance difference matrix (DDM)*, a $N \times N$ (where N is the number of C_α carbons) symmetric matrix whose entries, also called *scores*, are given by:

$$s_{ij} =\mid d^A_{i,j} - d^B_{i,j} \mid, i = 1, \ldots, N, j = 1, \ldots, N. \tag{4.2}$$

4.1.4 Step 2: Building the filtration and its persistence diagram

We use the $\binom{N}{2}$ scores of Eq. (4.2) to define two filtrations. The first one exploits the ordering of a.a. pairs induced by scores, to perform a multiscale analysis of structurally conserved pairs of a.a. The second one exploits an ordering of individual a.a. induced by scores, to also perform a multiscale analysis of structurally conserved regions.

Conserved distances filtration (CD filtration)

Score index. Consider the sorted sequences of scores from Eq. (4.2). An edge involves two a.a., so that in processing edges by increasing scores, we iteratively connect a.a. This processing is actually Kruskal's algorithm to build a minimum spanning tree (MST) [KT06]. We define:

Definition. 4.3. *(Score index) The scores contributing to a MST connecting the a.a. are called* non redundant*. Denoting N the alignment size, there are $N-1$ such scores. The* score index *of a non-redundant score is its index in the sequence of non redundant scores.*

Filtration. Our rationale is to identify rigid domains by incrementally processing scores contributing to the MST. Recall that Kruskal's algorithm processes edges by increasing scores. The connexion between connected components of the MST as characterized by the first and second Betti numbers β_0 and β_1, and scores/edges as processed in Kruskal's algorithm is as follows:

- (c.c. creation) edge triggering the creation of a connected component: the score involves two C_α not processed previously; this score creates a new c.c. in the MST. In terms of topology of the MST: $\beta_0 += 1$.
- (accretion) edge triggering accretion: the score involves only one C_α not processed previously, which therefore contributes to the extension of an existing c.c. In terms of topology, neither β_0 nor β_1 change.
- (c.c. destruction) edge triggering the destruction of a c.c.: the score involves two C_α already involved in scores processed previously, and these C_α belong to two different c.c.. These c.c. merge, so that $\beta_0 -= 1$. (NB: if the two C_α belong to the same c.c., then β_0 remains unchanged but $\beta_1 += 1$. Such edges creating cycles and are not used in the MST.)

Persistence diagram. The persistence diagram codes the lifetime of c.c. in the MST during its construction. Note that the persistence diagram constructed is the same for the two structures.

Remark 4.3. *The number of points in the PD is equal to the alignment length N. To see why, note that in running Kruskal's algorithm, vertices are associated with two types of* creation *events. For an edge triggering a c.c. creation: two c.c. are created–one for each vertex, but one dies immediately and has a null persistence. For an edge triggering accretion, one c.c. corresponding to the new vertex is created, and this c.c. also has a null persistence. Finally: every vertex of the graph is involved in exactly one of these two edge types.*

Space filling diagram filtration (SFD filtration)

We now exploit the scores s_{ij} in a different way, by focusing on individual C_α rather than edges. We assign to each C_α carbon an integer in the range $1, \ldots, \binom{N}{2}$, called its *rank*.

C_α ranks. We assess the structural conservation of the neighborhood of a given C_α with (Fig. 4.2):

Definition. 4.4. *(C_α rank) The* absolute C_α rank *of a C_α is the smallest index $\in 1, \ldots, \binom{N}{2}$ of the score s_{ij} involving this C_α. The largest absolute C_α rank is denoted N_r, and the number of (distinct) absolute C_α ranks n_r.*

The C_α rank *of a C_α is the index of its absolute C_α rank – an integer in the range $1, \ldots, n_r$.*

The amino-acids of structures A and B whose C_α ranks are equal to i are denoted $A_{(i)}$ and $B_{(i)}$, respectively.

Note that absolute C_α ranks are identical for atoms mapped with one another. We summarize with the following:

Observation. 3. *The following upper and lower bounds are tight:*

$$\lceil N/2 \rceil \leq n_r \leq N_r \leq \binom{N-1}{2} + 1. \tag{4.3}$$

Proof. Consider first absolute C_α ranks. For the lower bound, since a given score contributes to at most two ranks, $\lceil N/2 \rceil$ scores are needed to cover the N C_α.

For the upper bound, consider a (fictitious) molecule with a rigid region of $N-1$ C_α-s, and one mobile C_α. The absolute C_α rank of the mobile atom is equal to $\binom{N-1}{2} + 1$ since it is determined by the most conserved distance out of the $N-1$ distances to the C_α-s of the rigid region. This number is worst-case, since for the C_α with largest C_α rank, out of the $N-1$ distances to the remaining C_α-s, only the smallest one determines its C_α rank.

The bounds C_α ranks are trivial given those on absolute C_α ranks. □

Filtration. Consider now the processing of a.a. by increasing C_α rank. We maintain for each molecule a SFD, and report contacts between a.a. using the associated α-complex (section B.2.3). Consider a SFD (that of molecule A or B). A C_α may be ascribed to the following categories, which we distinguish in terms of topological changes:

- (c.c. creation) The C_α creates a new c.c. of the SFD; that is, $\beta_0+ = 1$.
- (c.c. destruction) The C_α bridges the gap between two existing c.c.; that is, $\beta_0- = 1$.
- (cycle creation) The C_α creates a cycle in an existing c.c.; that is, $\beta_1+ = 1$.
- (accretion) The C_α contributes to the enlargement of an existing c.c.; neither β_0 nor β_1 change.

Persistence diagram. Consider chain A – we proceed mutatis mutandis for chain B. Using the ordering of a.a. defined by C_α ranks, let $\mathcal{A}_i$ be the set of a.a. whose rank is at most i, that is $\mathcal{A}_i = \cup_{j=1,\ldots,i} A_{(i)}$. We denote the SFD of this collection of balls by $\mathrm{SFD}_i^{(A)}$. The connected components of this SFD evolve upon inserting a.a. (SI Fig. B.2). Because the whole polypeptide chain defines a connected domain, each c.c. but the one created by $A_{(1)}$ is characterized by two dates, namely its birth date and it death date. Furthermore:

Observation. 4. *Consider two chains A and B whose space filling diagrams are connected. Upon processing the last C_α rank, the space filling models $SFD_{n_r}^{(A)}$ are $SFD_{n_r}^{(B)}$ are connected.*

Note, however, that as opposed to the PD obtained with scores, those obtained with C_α ranks are not identical. Indeed, the connectivity obtained along the filtration depends on the relative position of a.a., which differ in A and B.

Comments

Accretion. The two types of filtrations undergo different topological events, as characterized by the evolution of Betti numbers. One common feature though is the presence of *accretion*: accretion is the addition of one edge (conserved distances filtration) or one a.a. (SFD filtration), without any change in β_0 or β_1.

Accretion corresponds to the incremental formation of a motif, in such a way that no motif is *stable*. In topological persistence terms, accretion is characterized by null persistence.

Practically, accretion calls for two comments:

- While accretion does occur, our experiments show that its extent is not such that motifs with significant persistence do not exist.
- As we shall see, iterative alignments provide a natural way to rescue regions of motifs plagued by accretion.

Plots. To study the correlation between C_α ranks and properties of the sequences, we define (Fig. 4.3):

Definition. 4.5.
- Score plot: *score s_{ij} as a function of the C_α rank.*
- Sequence shift plot: *for chain A (or chain B), the function $j - i$ (distance along the sequence) as a function of the C_α rank.*
- C_α distance plot: *for chain A, the function $d^A_{i,j}$ (or $d^B_{i,j}$) as a function of the C_α rank.*

4.1.5 Step 3: Computing structural motifs with bootstrap

This step exploits either of the aforementioned persistence diagrams. It also relies on an alignment algorithm, which is assumed to be identical to the one used to compute the seed alignment and the scores.

From connected components to motifs. Consider a point $a = (b_a, d_a) \in \mathrm{PD}_A$, with b_a and d_a the birth and death dates of a, respectively. To point a, we associate the set of connected component of the sublevel set defined by d_a, which we denote $F_A(a) = \{c_1, \ldots, c_{n_A}\}$. For $b \in \mathrm{PD}_B$, we obtain similarly the set $F_B(b) = \{c'_1, \ldots, c'_{n_B}\}$.

Definition. 4.6. *(Structural motif) Consider two points from the PD, namely $a \in PD_A$ and $b \in PD_B$, and the associated connected components of sublevel sets, i.e. $F_A(a) = \{c_1, \ldots, c_{n_A}\}$ of $\mathcal{F}^A_a$ and $F_B(b) = \{c'_1, \ldots, c'_{n_B}\}$.*

Consider pair of large enough c.c., that is $(c_i, c'_j) \in F_A(a) \times F_B(b)$ with $\min(|c_i|, |c'_j|) \geq \tau_{MS} (= 10)$.

A structural motif *for that pair is defined by the structural alignment obtained for (c_i, c'_j).*

It should be noticed that since a motif consists of a.a. singled out by a local alignment, it may not be connected in 3D space, even when the persistence diagram is associated with the SFD filtration. Note that by construction, however, the motif contains the same number of a.a. on each molecule – a property yielded by the structural alignment.

Localizing comparisons. Definition 4.6 requires processing all pairs of c.c. in $F_A(a) \times F_B(b)$. On the other hand, under suitable assumptions, persistence diagrams are known to be stable [CSEH07]. As a heuristic to reduce the number of points from PD_A and PD_B compared, we require the Euclidean distance in PD space is less than a threshold τ:

$$\|p - q\|_2 \leq \tau_{\mathrm{PD}}. \tag{4.4}$$

The case of homologous proteins. In comparing homologous proteins, to reduce the number of pairs of c.c. processed, we further require the two compared c.c. to have comparable size, that is:

$$|\,|p| - |q|\,| \leq \tau_{\mathrm{MSD}} \tag{4.5}$$

The case of two conformations. When comparing two conformations of the same structure, the situation is easier since one has the identical alignment between chains of the two conformations, see Remark 4.1. In that case, we perform a one step clustering of c.c. using the D-family matching algorithm [CMTW17] – a method to group clusters into meta-clusters with a one-to-one correspondence between meta-clusters. We then process the pairs corresponding to meta-clusters.

4.1.6 Step 4: Filtering structural motifs

Motif inclusion. Consider the subsequences of a.a. defining motifs. Due to the nested structure of sets obtained from filtrations, motifs may be nested. Since a motif reduces to a set of a.a., we may consider the partial order defined by set inclusion, and the associated Hasse diagram. (NB: the Hasse diagram is the directed graph whose edges precisely code inclusion between motifs.) Of particular interest are the terminal nodes of the Hasse diagram, which may be seen as maximal motifs in terms of size.

Statistical significance. The statistical significance of our motifs can also be assessed using a two-sample test involving random motifs (SI Section B.3).

4.1.7 Expanding motifs with iterative aligners

Iterative alignments [BOPR03, RGMV12] consist in iteratively finding the alignment for fixed positions of the conformations–via dynamic programming, and finding optimal superimposition given the alignment– the classical rigid superimposition problem. The process is typically iterated until a fixed point or a cycle is reached. Seeding such alignments with our motifs is especially interesting for two reasons. First, the construction of motifs inherently copes with accretion–Sec. 4.1.4. Therefore, iterative alignments allow the extension of motifs with a.a. involved in accretion. Second, motifs define motif graphs (Def. B.1, SI Sect. B.2.1). Given a c.c. of a motif graph, an iterative alignment can be seeded with each motif from this cc. In the end, we retain the alignment maximizing the G-score [RGMV12].

It should be noted that when processing two conformations, the trivial/identity alignment is used (remark 4.1), and the process simplifies to the following two steps:

1. (Alignment given relative position) Add to the current alignment each residue pair whose distance is below a threshold τ_L.
2. (Relative position given alignment) Use the previously computed alignment to perform a rigid registration.

4.2 Implementation

All methods are available in the Structural Bioinformatics Library (`http://sbl.inria.fr`, [CD17]).

4.2.1 Seed aligners

A structural alignment algorithm is required to get the seed alignment and to compare two points from the persistence diagrams. Practically, we use the aforementioned state-of-the-art options `Apurva` [AMDY11] (`SBL` package: `https://sbl.inria.fr/doc/Apurva-user-manual.html`) and `Kpax` [RGMV12] (`SBL` package: `https://sbl.inria.fr/doc/Iterative_alignment-user-manual.html`).

4.2.2 Persistence diagrams

Consider the score filtration—Sec. 4.1.4. We build a complete edge weighted graph whose nodes are the a.a.. The weight of an edge is the score associated to this edge Eq. (4.2). Sorting the edges of this graph yields a filtration, from which the persistence diagram for connected components is easily maintained using

a union-find data structure. Practically, we use the Morse theory based analyzer from the SBL (SBL package: https://sbl.inria.fr/doc/group__Morse__theory__based__analyzer-package.html).

Consider the SFD filtration–4.1.4. We build a vertex weighted graph whose nodes are the a.a., sorted by increasing C_α rank. For each molecule, we sequentially insert the a.a., maintain the corresponding α-complex, and build the associated persistence diagram, also using the Morse theory based analyzer from the SBL.

4.2.3 Motifs

Methods: Non-iterative aligners. Combining the two seed aligners (Apurva, Kpax) and the two persistence diagrams (CD: conserved distances, SFD: space filling diagram) yields four options. One also needs to distinguish conformations versus homologous proteins, as the former involve the trivial identity alignment (remark 4.1). Summarizing, we report in Experiments results for the following combinations (in parenthesis, the color used for plots when appropriate):

- Aligners requiring an alignment: Align-Kpax-CD, Align-Kpax-SFD, Align-Apurva-CD, Align-Apurva-SFD;
- Aligners with the identity alignment: Align-Identity-CD, Align-Identity-SFD.

Practically, these methods are implemented within the following executables from the SBL– each giving access to the CD and SFD filtrations: sbl-structural-motifs-chains-apurva.exe, sbl-structural-motifs-chains-kpax.exe, sbl-structural-motifs-conformations.exe.

Methods: Iterative aligners. Seeding Kpax with our motif yields four different iterative aligners, namely Align-Kpax-SFD/iter, Align-Kpax-CD/iter, Align-Apurva-SFD/iter, Align-Apurva-CD/iter.

Packages from the SBL and main parameters. Methods and executables are summarized in SI Table B.1. The programs computing motifs are provided in the package Structural_motifs package (SBL package: https://sbl.inria.fr/doc/Structural_motifs-user-manual.html), and are used in particular to compute flexible distance measures in the package Molecular_distances_flexible (https://sbl.inria.fr/doc/Molecular_distances_flexible-user-manual.html). Iterative aligners are provided in the package Iterative_alignment (https://sbl.inria.fr/doc/Iterative_alignment-user-manual.html).

The main parameters of these programs are: τ_{MS} for the motif size (Def. 4.6), $\tau_{/}$ for the lRMSD ratio (Def. 4.2), τ_{PD} for the comparison of persistence diagrams (Eq. 4.4). Practically, the following parameters are used:

- comparing homologous proteins: $\tau_{\text{MS}} = 10$, $\tau_{/} = 0.8$, $\tau_{\text{MSD}} = 10$; SFD filtration: $\tau_{\text{PD}} = 20$; CD filtration: $\tau_{\text{PD}} = 0$.
- comparing conformations: $\tau_{\text{MS}} = 10$, $\tau_{/} = 0.5$; SFD filtration: $\tau_{\text{PD}} = 5$; CD filtration: $\tau_{\text{PD}} = 0$.

4.3 Results

4.3.1 Datasets

Structural comparisons and motifs. To understand properties of motifs, we use a dataset of eight class II fusion proteins (SI Fig. B.1), namely proteins used by enveloped viruses to trigger the fusion between their membrane and that of the target cell [Kie14]. These 8 structures (sizes varying from 380 to 461 a.a., SI Fig. B.1) yield $8 \times 7/2 = 28$ pairwise comparisons. The rationale for this dataset is twofold. Structurally, class II fusion proteins are elongated molecules with three domains (DI, DII, DIII) composed primarily of β-sheets. The central DI domain connects via a flexible hinge to the longer DII. Typically, DII contains

several conserved disulfide bonds as well as the so-called fusion loop at its tip. Additionally, a linker region connects DI to the DIII domain, which has an Immunoglobulin (Ig)-like fold.

On the biological side, (class II) fusion proteins are challenging structures, as they harbor a low sequence identity (typically less than 20%), and overall loose structural homology (typically of the order of 10Å). Yet, they have the same function, for a reason which has remained elusive. Thus, in investigating structurally conserved motifs of such proteins, we provide insights on fusion mechanisms(Chapter 5)

On the computational side, such a dataset remains tractable. To see why, recall that we use two aligners, namely `Apurva` (with a focus on contacts) and `Kpax` (with a focus on backbone geometry), at two stages: first, to obtain the seed alignment; second, to obtain the motifs by performing a local alignment when processing pairs of points from the persistence diagrams (Def. 4.6). This latter operation is the computational bottleneck. While `Kpax` is fast and typically runs under a minute even on large examples, `Apurva` may be quite slow even on small examples. With default parameters, run time can reach one hour on a single instance. The comparison of two PD may thus take several hours, which greatly limits the size of the data set when performing large scale comparisons.

Comparison against flexible aligners. For a direct comparison against flexible aligners, we use the 10 'difficult' structures from [YG03].

4.3.2 Method illustration: investigating a conformational change

We illustrate each step of our method to study the conformational change undergone a prototypical class II fusion protein, from the tick-borne encephalitis virus. The ectodomain of this protein was crystallized both in soluble form (PDB: 1SVB, [RHM$^+$95], 395 residues) and in postfusion conformation (PDB: 1URZ, [BSA$^+$04], 400 residues). For these structures, the identity alignment identifies 376 a.a.

Step 1 – scores and C_α ranks. We compute C_α ranks as described in Sec. 4.1.3 and analyze the C_α distance plot, the sequence shift plot, and the score plot. (Fig. 4.3). No correlation is observed between C_α ranks and (i) the proximity along the sequence, and (ii) the location on SSE (inferred from C_α distances). That is, C_α ranks identify rigid pairs throughout the structures.

Step 2 – persistence diagrams. Persistence diagrams exhibit different amounts of persistent events / accretion (SFD filtration, Fig. 4.4; CD filtration, SI Fig. B.3). When using a SFD filtration, most main merging events happen before C_α rank 100. As expected, in the case of conserved distances, the PD is denser and less prone to accretion.

Step 3 and 4 – motifs. The hierarchical structure of the motifs found is coded in the Hasse diagram coding motif inclusion (Sec. 4.1.6; Fig. 4.4). On this example, the two motifs at the root of the Hasse diagram characterize the hinge motion associated with the conformational change (Fig. 4.4(c)). This hinge motion is best characterized by comparing the global lRMSD yielded by our seed aligners, and the combined RMSD which mixes the lRMSD of the two motifs (Eq. (B.1, [CT18a]):

- `Kpax`: #a.a.: 296, lRMSD : 9.28 Å;
- `Apurva`: #a.a.: 370; lRMSD : 11.1 Å;
- Motifs yielded by `Align-Identity-SFD`: #a.a.: 152; $\text{RMSD}_{\text{Comb.}}$: 2.53 Å.
- Motifs yielded by `Align-Identity-CD`: #a.a.: 161; $\text{RMSD}_{\text{Comb.}}$: 1.26 Å.

Iterative alignment. Motifs can be used to seed iterative aligners–Sec. 4.1.7.

Using the SFD filtration with default parameters, `Align-Identity-SFD`/iter (See SI Table B.1) identifies the rigid groups formed by the DI and DII domains on the one hand, and the DIII domain on the other. These motifs are characterized by (Fig. 4.4):

- Motif 1 (DI, DII): #a.a.: 171, lRMSD : 0.86.
- Motif 2 (DIII): #a.a.: 93, lRMSD : 0.58
- Overall comparison: #a.a.: 264, $\text{RMSD}_{\text{Comb.}}$: 0.76

Using the CD filtration, `Align-Identity-CD`/iter with tuned parameters $\tau_{\text{PD}} = 0$, $\tau_{\text{MS}} = 5$, $\tau_{/} = 0.5$, $\tau_L = 0.8$ (Nb: τ_L specific to iterative aligners for conformations). even identifies all three domains of the structure (Fig. 4.5). These motifs are characterized by:

- Motif 1 (DII): #a.a.: 115, lRMSD : 0.51
- Motif 2 (DI): #a.a.: 79, lRMSD : 0.37
- Motif 3 (DIII): #a.a.: 40, lRMSD : 0.49
- Overall comparison: #a.a.: 234, $\text{RMSD}_{\text{Comb.}}$: 0.46

As illustrated by this example, using motifs in combination with the combined RMSD yields a reduction of ~ 20 of the metric used to perform the structural comparison.

4.3.3 Motifs: a case study for homologous proteins

Comparing the four aligners: overall comparison

Method. We challenge our four methods (`Align-Kpax-CD`, `Align-Kpax-SFD`, `Align-Apurva-CD`, `Align-Apurva-SFD`) and also compare them to the seed aligners (`Apurva`, `Kpax`), using all pairwise comparisons of the class II fusion dataset. In a first step, we analyze statistics describing the motifs yielded by the contenders. In a second step, we exploit motifs to perform an overall comparison of whole structures based on motifs. For the seed aligners `Apurva` and `Kpax`, we report the lRMSD associated with the seed alignment, together with the alignment length. For our four aligners, we report the combined $\text{RMSD}_{\text{Comb.}}$ associated with our motifs and the number of a.a. involved in this calculation [CT18a].

Seed aligners. The alignments returned by our base aligners `Apurva` and `Kpax` for the 28 comparisons call for two comments (Fig. 4.6). First, `Apurva` alignments have a lRMSD which can be up to four times larger than `Kpax` alignments; second, `Kpax` alignments are smaller –the median is $\simeq 100$ residues smaller. This is in line with the goals of these aligners, as `Apurva` tries to maximize the numbers of contacts, while `Kpax` favors the coherence of backbone geometry.

Motifs per se. The motifs returned by our four methods provide insights on the merits of the seed aligners and the filtrations (see also SI Sec. B.4.1 for the statistical significance of motifs.) Three facts emerge from the general statistics obtained on the whole dataset (Table 4.1).

First, the CD filtration yields longer motifs (Table 4.1; median size 38 and 31.5, versus 18 and 18). This is expected, as defining connected components from a space filling diagram is more stringent than from the MST exploiting conserved distances. Second, the lRMSD of motifs returned by methods using `Apurva` as seed aligner are larger (Table 4.1; median values of 4.45 and 5.13, against 2.46 and 2.78). This is also expected, as the seed alignment of `Apurva` favors length, while that of `Kpax` favor geometric coherence. The third is that the lRMSD ratio of motifs returned by methods using `Apurva` tends to be smaller (Table 4.1; median values of 0.48 and 0.64, against .62 and 0.70). This stresses the importance of the seed alignment size in the definition of the lRMSD ratio (Def. 4.2).

Motifs to compare whole structures. The lRMSD of motifs can be combined to compare whole structures [CT18a] (Fig. 4.6). The $\text{RMSD}_{\text{Comb.}}$ is always globally better than the lRMSD, except in the case of `Align-Apurva-SFD` in which it performs poorly in few instances (Fig. 4.6, top left). For `Align-Kpax-SFD` and `Align-Kpax-CD`, the median $\text{RMSD}_{\text{Comb.}}$ is $\simeq$ 1 Å smaller than the initial lRMSD median. For `Align-Apurva-SFD`, the median $\text{RMSD}_{\text{Comb.}}$ is comparable to the initial lRMSD median but both the top and lower quartile are $\simeq 2.5\mathring{A}$ smaller so that the results are more homogeneous. When using a SFD filtration, the median number of residues involved in the $\text{RMSD}_{\text{Comb.}}$ is $\simeq$ 50 residues. In some instances, it can be an order of magnitude (between 30 and 40 times) smaller than the seed alignment length (Fig. 4.6, bottom left). With conserved distances, the median number of residues involved in $\text{RMSD}_{\text{Comb.}}$ is $\simeq$ 100 residues.

Comparing the four aligners: pairwise comparison

Method. We now compare the four methods, exploiting the multiscale nature of the motifs returned. Each motif is characterized by a 2D point (motif size, lRMSD). The comparison of two structures can therefore by summarized in this space by a point cloud and the associated Pareto envelope. Note that in (motif size, lRMSD) space, the Pareto front is defined using the following notion of domination: a motif a dominates a motif b iff a has more a.a. than b, and the lRMSD of a is smaller than that of b.

In the sequel, we compare the Pareto fronts of the four methods for each of the 28 pairwise comparisons, in order to assess whether a particular method stands out (for the full matrix of Pareto fronts, see SI Fig. B.4.1).

Each point cloud and curve is color coded with respect to one of the four methods: `Align-Kpax-CD` (blue), `Align-Kpax-SFD` (orange), `Align-Apurva-CD` (green), `Align-Apurva-SFD` (red).

Results. Four scenarios emerge from the 28 pairwise comparisons.
1. Blue dominates (Fig. 4.7): this is the most common case. Generally, `Align-Kpax-CD` performs better than other methods w.r.t lRMSD and motif sizes. In the example, even though the blue motif is $\simeq 2\times$ larger than the orange motif (which is itself significantly larger than any other motif), its lRMSD is slightly lower.
2. Mixed domination, green then blue (Fig. 4.8): About one third of the comparisons show a relayed domination between a given method and `Align-Apurva-CD` (green). The example from Fig. 4.8 is particularly striking as `Align-Apurva-CD` returns very large motifs (343 residues).
3. Orange dominates (Fig. 4.9): In a few other cases, `Align-Kpax-SFD` (orange) dominates. The example from Fig. 4.9 shows the particularities of each method. The blue motif (found by `Align-Kpax-CD`) is composed of many smaller connected components distributed through out the structure. On the other hand, the orange motif (found by `Align-Kpax-SFD`) is localized on the top of the structure and formed of a unique, larger, connected component.
4. Mixed domination, all (Fig. 4.10): Finally, in some cases, we observed a four-way relayed domination in which each method dominates at a certain lRMSD range. The example from Fig. 4.10 shows how each method returns a larger and larger motif until spanning the entire structure.

To summarize, although `Align-Kpax-CD` dominates more often (w.r.t motif lRMSD and sizes), there is no clear better method. Qualitatively, the motifs returned by each method are quite different, the rule being that with SFD filtration motifs tend to be more localized and connected, compared to CD filtration which returns larger albeit more dispersed motifs.

4.3.4 Comparisons against flexible aligners

As noted in Introduction, our methods are not direct competitors of flexible aligners since a key goal is to capture the various structural conservation scales a seed alignment may contain.

Nevertheless, we compare our aligners against classical flexible aligners, in particular `FATCAT` and `Kpax`, using the 10 *difficult* structures from [YG03] (SI Section 4.3.4, SI Tables B.3, B.4, B.5 and B.6).

`Align-Kpax-SFD` and `Align-Apurva-SFD` fail to find motifs on certain structures. This is to be expected on smaller structures as they are prone to accretion. In most instances our methods yield $\text{RMSD}_{\text{Comb.}}$ values which are a significant improvement to the lRMSD of the `FATCAT` alignment, although at a cost in size. Recall that motifs are constrained areas of the structures. In some instances, $\text{RMSD}_{\text{Comb.}}$ is comparable to `FATCAT` (slightly better or slightly worse). In a few cases, our methods yield poor results: a striking example is that of 1CRL vs 1EDE for which we find a $\text{RMSD}_{\text{Comb.}}$ of 7.47 (against a lRMSD of 3.55 for `FATCAT`). These poor results are due to the dependency to the seed alignment and the parameterization of our method. In the same example (1CRL vs 1EDE), the seed alignment has a lRMSD of 8.08 and $\tau_{/} = 0.8$. Running the same comparison with $\tau_{/} = 0.3$ yields a very different result: 84 residues involved and $\text{RMSD}_{\text{Comb.}} = 2.38$. This is a demonstration of how constraining $\tau_{/}$ enables the discovery of more conserved motifs. Seeding iterative alignments with our motifs yields results that are comparable to `FATCAT`. These alignments are very similar to the ones provided by `Kpax` in terms of lRMSD and size. However comparing structural alignments based only on the lRMSD is very constraining and these final alignments should not be dismissed, they can provide additional information.

Remark 4.4. *The previous assessment calls for an important remark: lowest lRMSD, albeit commonly used, is not always a good evaluation criterion when comparing alignment methods.*

Remark 4.5. *Normally, after computing an alignment,* `Kpax` *checks for distant aligned pairs and removes them from the alignment [RGMV12]. The* `SBL` *implementation of* `Kpax` *does not do this by default, this can create some poor alignments (w.r.t lRMSD). As already noted, by running* `Kpax` *(*`SBL`*) on 1CRL vs 1EDE yields an alignment with a high lRMSD (8.08). By adding a distance threshold of* $0.7\mathring{A}$ *on aligned residues, we obtain a 137 residue long alignment with a lRMSD of* $1.72\mathring{A}$*.*

4.4 Discussion and outlook

Molecular flexibility is a continuous process, with characteristic spatial and time scales, so that a key difficulty in understand the role of flexibility and dynamics in protein function is to perform a multiscale analysis of structurally conserved motifs. Our work precisely addresses this task, by proposing a generic framework to automatically detect the multiple flexibility scales which may exist when comparing two structures, be they conformations of the same molecule, or homologous proteins. To this end, our framework bootstraps from a seed alignment, and further exploits the hierarchical information contained in so-called filtrations and the associated persistence diagrams, defined from distance difference matrices.

Our motifs naturally accommodate a hierarchical representation in terms of Hasse diagram, which provides the basis of multiscale analysis. As a first intent, terminal motifs (i.e. motifs defining roots of the aforementioned Hasse diagram) can be used in conjunction with the recently proposed combined lRMSD. More precisely, since a motif has its own optimal rigid motion and lRMSD, these lRMSD can be combined into a global measure to compare two structures. We show that the resulting combined RMSD may reduce the global lRMSD by one order of magnitude, providing a tool of unprecedented accuracy to compare two structures which otherwise may appear as radically different from a structural standpoint.

Importantly, our framework enjoys two major design choices. The first one is the seed alignment used, which may favor topological information (conserved contacts, via contact map optimization), or geometric information (e.g. conserved backbone geometry). Practically, we tested two seed aligners, namely `Apurva`, which solves the contact map overlap problem with guarantees, and `Kpax`, an iterative aligners favoring the coherence of backbone geometry. The second one is the type of filtration used, which may favor either motifs stemming from connected regions in 3D (defined via space filling diagrams), or motifs stemming from conserved distance between distant amino-acids. Each of the four resulting structural aligner targets a specific type of flexibility, and we exhibit such examples for class II fusion proteins.

In a more general perspective, these examples refer to different scenarios in terms of molecular mechanisms. The presence of structurally distant yet conserved motifs may be related to cooperative and allosteric

phenomena. The presence of compact and conserved motifs, such as those observed in a classical hinge motion, may just reveal conformational changes via relative domain motions. In any case, strongly conserved motifs may hint at regions coupled to a specific function. Under this assumption, the sub-sequences associated to motifs may be used to design hybrid sequence-structure based (profile HMM) models of protein families. This strategy has been successfully pursued and is presented in Chapter 5, where two of our methods (`Align-Kpax-CD`,`Align-Kpax-SFD`) identified from `UniProtKB` unknown class II fusion proteins in drosophilia melanogaster. Importantly, the notion of (local) motif, rather than (overall) structural alignment, turned out to be instrumental.

Our methods to report structural motifs are made available via the Structural Bioinformatics Library (`http://sbl.inria.fr`). We anticipate that they will be of interest in all problems dealing with structural analysis, and also with the identification / annotation of distant homologs.

As future work of utmost interest, we wish to point out the interest of our motifs to seed iterative structural alignments. While such alignments have traditionally focused on a small number of (optimal) solutions, in using the variety of our motifs, it might be possible to thoroughly explore the space of similar motifs. Successfully implementing this idea would yield the counterpart of potential energy landscape exploration algorithms.

4.5 Artwork

Figure 4.1 The four step method to identify structural motifs

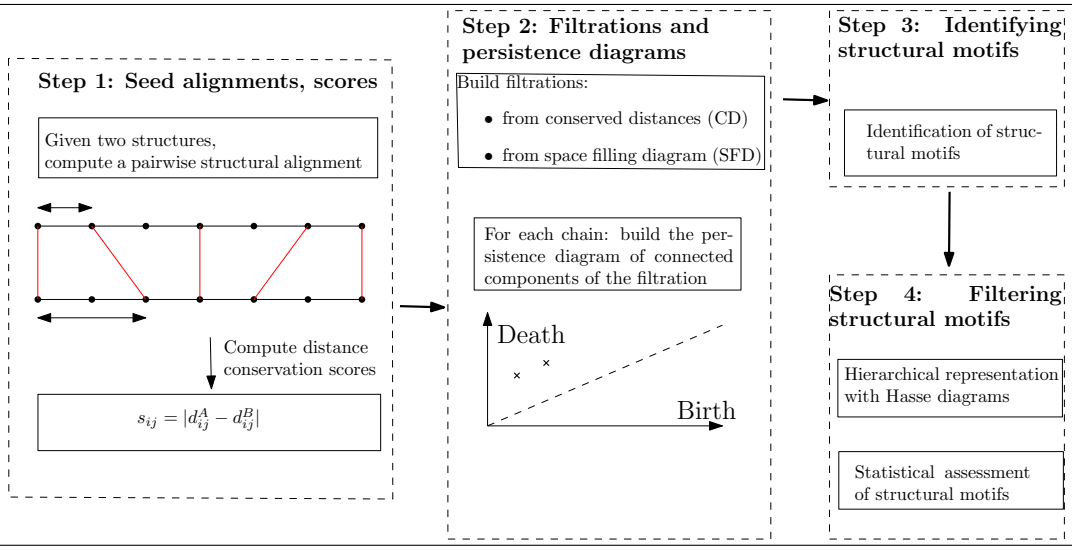


Figure 4.2 Step one, Computing the seed alignment and its scores: method. (A) Consider the alignment $(a_i \leftrightarrow b_i)_{i=1,\dots,N}$ between two fictitious chains (bold line-segments) of length $N = 5$. **(B)** The $\binom{5}{2} = 10$ scores are sorted. The scores involved in the definition of the conserved distances (CD) filtration, which also define a spanning tree connecting the a.a. of each structure, are: $s_{1,2}, s_{2,3}, s_{3,4}, s_{4,5}$. **(C)** On this toy example, the same scores contribute to the definition of C_α ranks, from which the space filling diagram filtration is defined.

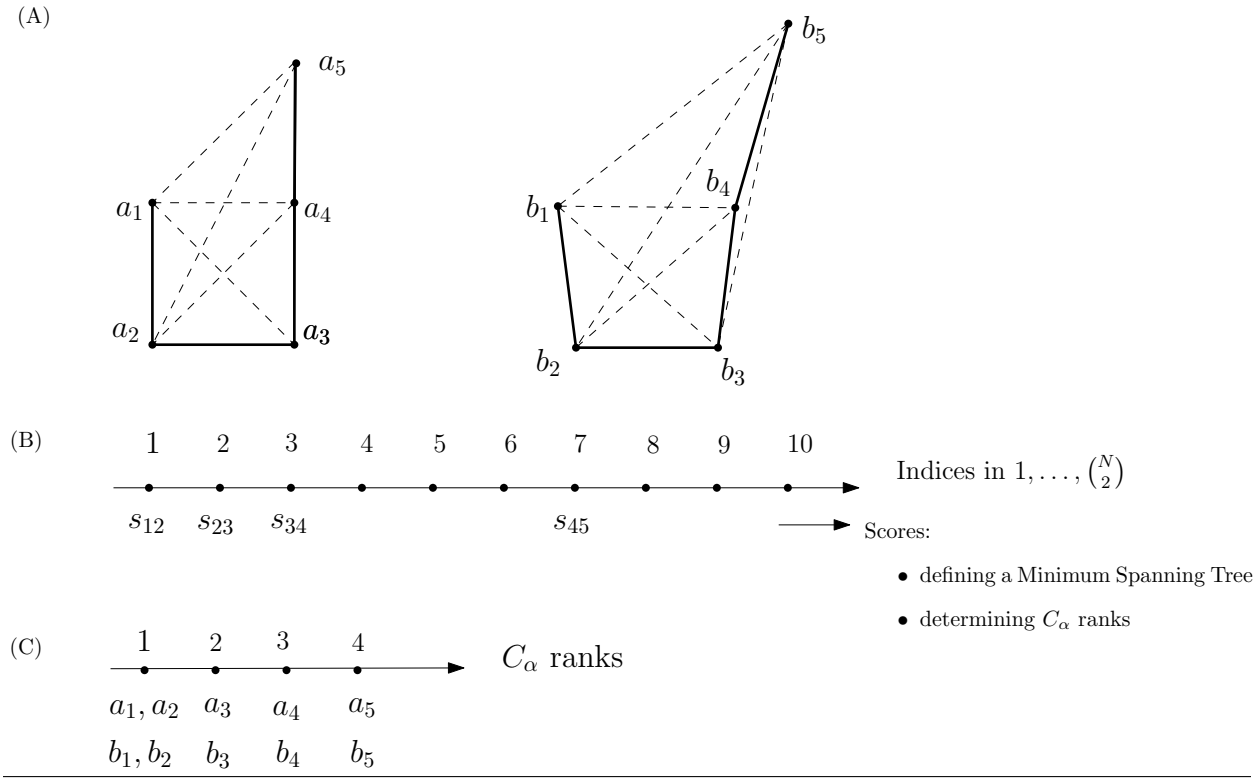


Figure 4.3 Step 1, Computing the seed alignment and its scores: illustration for a class II fusion protein of TBEV in two different conformations (pre-fusion (PDB: 1SVB), post-fusion (PDB: 1URZ)) Plots definition: see Def. 4.5; overview of structures, see SI Fig. B.1. **(Top)** C_α distance plot **(Middle)** Sequence shift plot **(Bottom)** Score plot No correlation is observed between C_α ranks and (i) the proximity along the sequence, and (ii) the location on SSE. That is, C_α ranks identify rigid pairs throughout the structures.

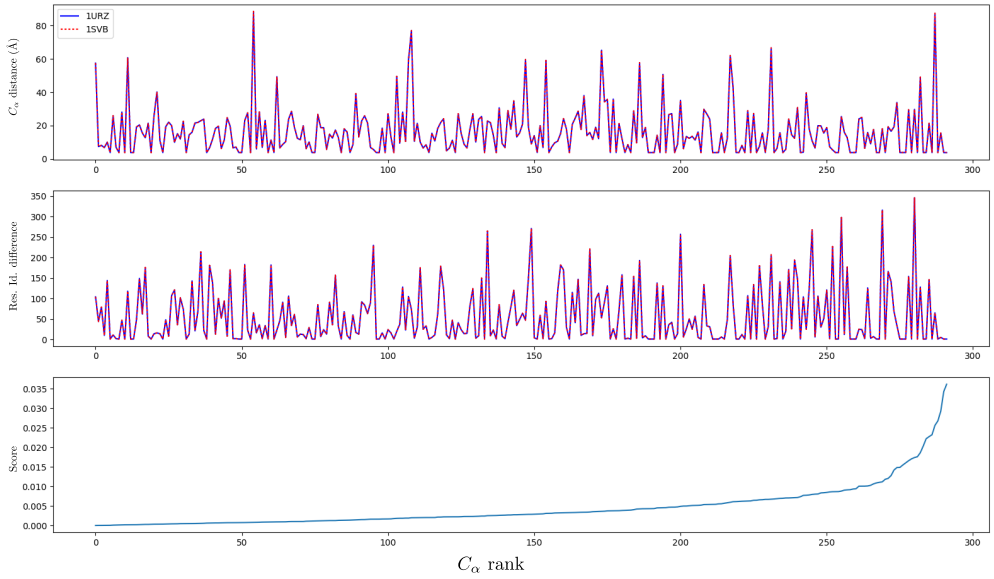
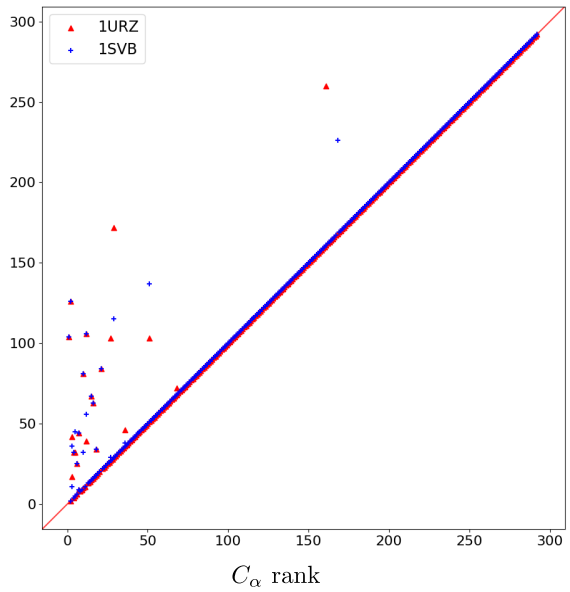
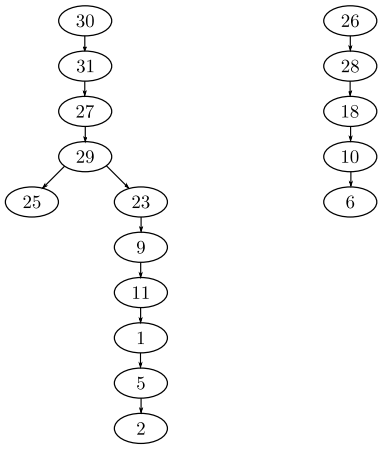


Figure 4.4 Step 2, Building the filtration and its persistence diagram: illustration for 1URZ-1SVB with `Align-Identity-SFD`. Comparing two conformations of TBEV class II fusion protein yields two nested sets of structural motifs which successfully characterize the two parts of the hinge motion. **(A)** Persistence diagram for SFD filtration. **(B)** Hasse diagram of structural motifs. Each motif has its unique index. **(C)** Selected motifs. Each motif corresponds to one part of the hinge motion associated to the two conformations. **(D)** Statistics for the structural motifs.

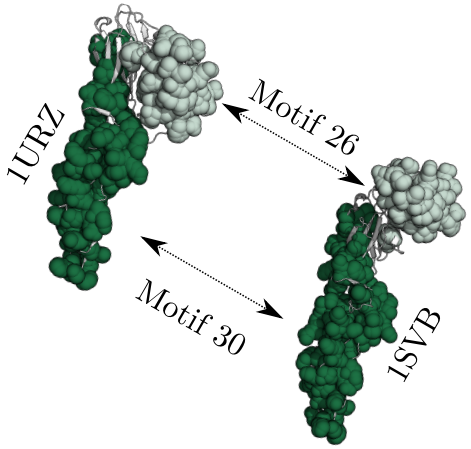
(A)



(B)



(C)



(D)

Motif index	lRMSD	length
30	1.82	109
31	1.81	107
27	1.66	85
29	1.62	83
25	0.59	11
23	0.59	58
9	0.59	39
11	0.61	36
1	0.28	18
5	0.29	17
2	0.29	16
26	3.77	43
28	3.92	39
18	4.23	32
10	4.12	24
6	4.51	18

Figure 4.5 Using motifs found with `Align-Identity-CD` as seeds for an iterative alignment characterizes all three domains of the TBEV class II fusion proteins.

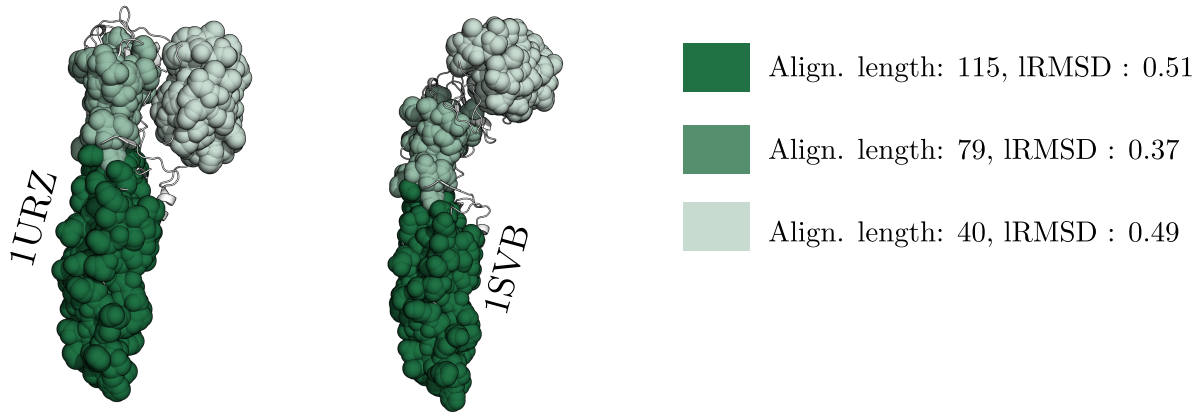


Table 4.1 General statistics for motifs returned by our four methods. Statistics are reported for the 28 pairwise comparisons on the class II fusion proteins.

Method	# motifs	Size			lRMSD			lRMSD ratio		
		min	median	max	min	median	max	min	median	max
`Align-Kpax-SFD`	333	11	18	143	0.67	2.46	4.51	0.16	0.62	0.80
`Align-Kpax-CD`	509	11	38	171	0.74	2.78	6.75	0.28	0.70	0.80
`Align-Apurva-SFD`	418	11	18	157	0.63	4.45	12.30	0.07	0.48	0.80
`Align-Apurva-CD`	333	11	31.5	354	0.62	5.13	14.37	0.04	0.64	0.80

Figure 4.6 Motif based comparison of the four aligners `Align-Kpax-CD`, `Align-Kpax-SFD`, `Align-Apurva-CD`, `Align-Apurva-SFD`. Note that an aligner is defined by the conjunction of an alignment method (`Apurva`, `Kpax`) and a filtration method (SFD: Space Filling Diagram; CD: conserved distances). The comparison is based on two statistics: for seed aligners (`Apurva`, `Kpax`), the lRMSD of the alignment and the alignment size; four our four aligners: the combined RMSD $\text{RMSD}_{\text{Comb.}}$ defined from the motif graph, and the number of a.a. involved (the number of vertices of the motif graph).

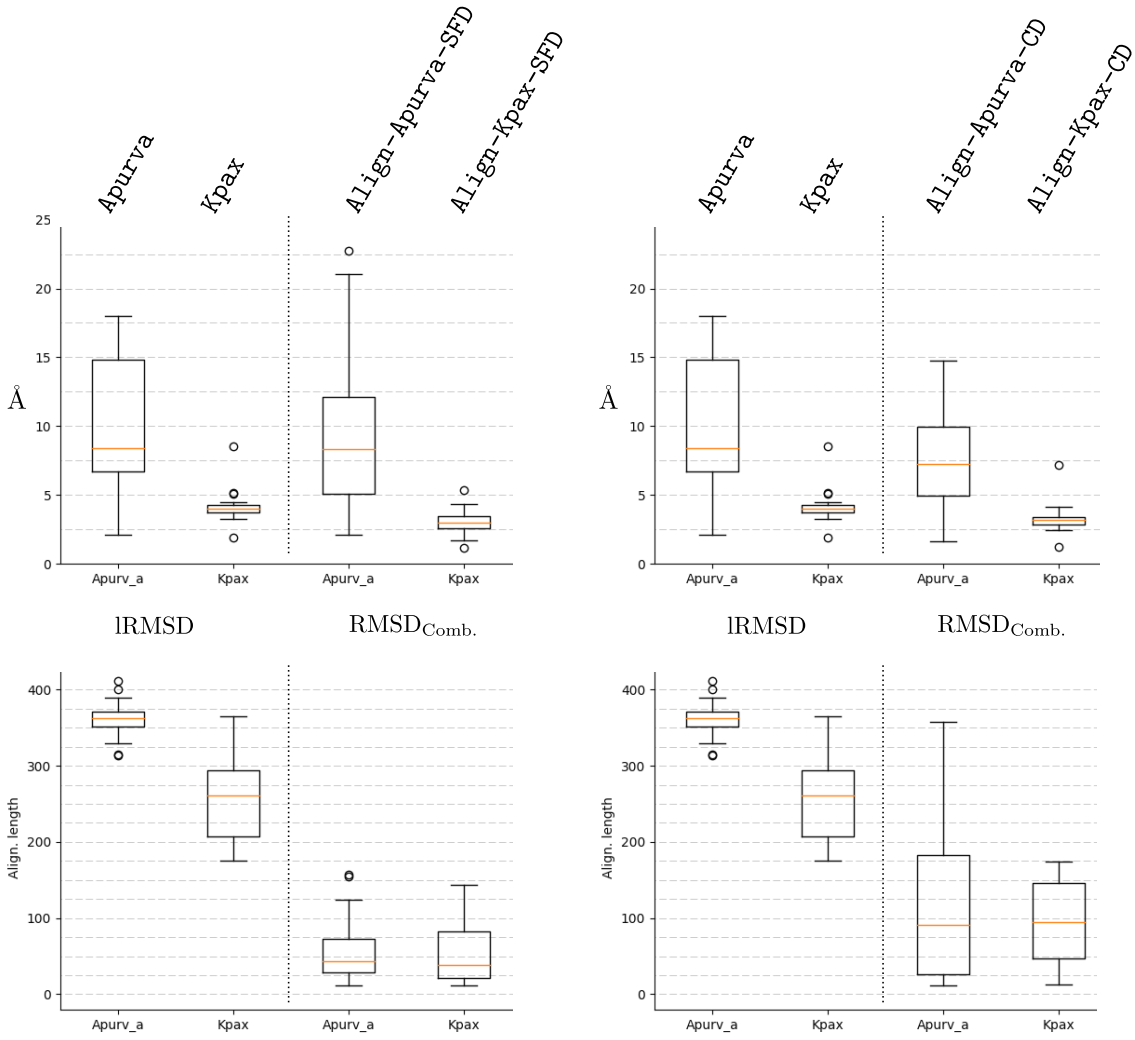


Figure 4.7 Comparison of RVFV-Phlebo. with HRV-Hanta.: (Left) Point clouds and Pareto envelopes of structural motifs found with each method. `Align-Kpax-CD` (blue) dominates all the other methods. **(Right)** Visualization of two structural motifs corresponding to the corner points of the `Align-Kpax-CD` (blue) and `Align-Kpax-SFD` (orange) curves.

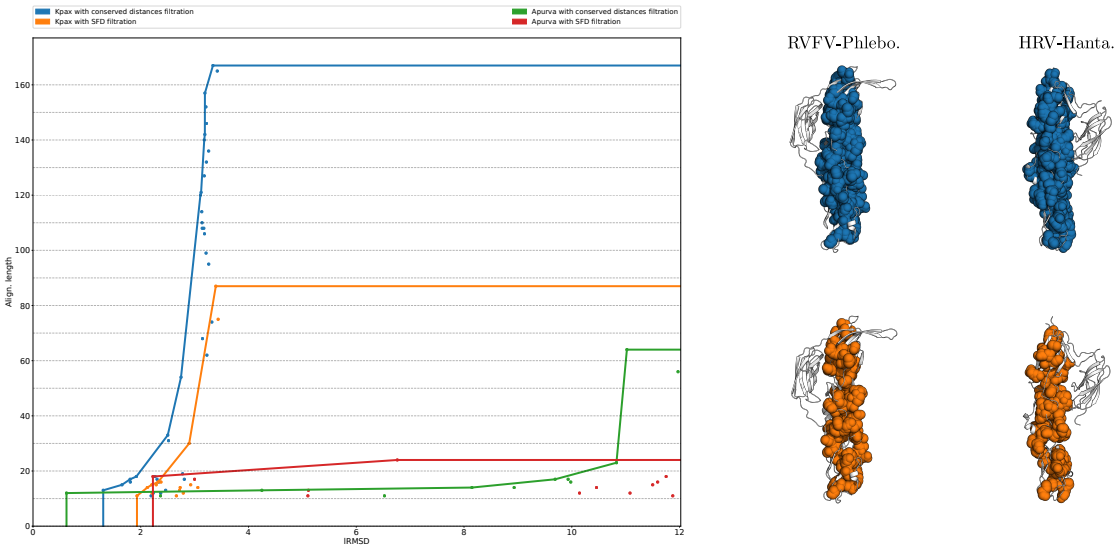


Figure 4.8 Comparison of DFV-Flavi. and RVFV-Phlebo.: (Left) Point clouds and Pareto envelopes of structural motifs found with each method. Initially, `Align-Apurva-CD` (Green) dominates until a critical point is reached and `Align-Kpax-CD` (Blue) takes over. **(Right)** Visualization of two structural motifs corresponding to the corner points of the `Align-Apurva-CD` (Green) and `Align-Kpax-CD` (Blue) curves.

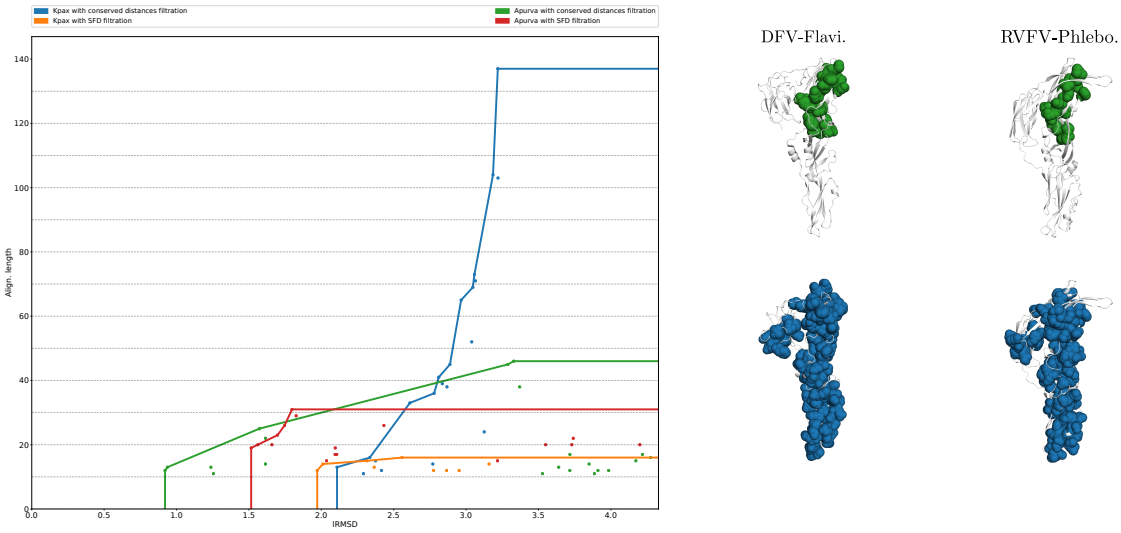


Figure 4.9 Comparison of SFV-Alpha. and RVFV-Phlebo.: (Left) Point clouds and Pareto envelopes of structural motifs found with each method. `Align-Kpax-SFD` (Orange) dominates all the other methods. **(Right)** Visualization of two structural motifs corresponding to the corner points of the `Align-Kpax-SFD` (Orange) and `Align-Kpax-CD` (Blue) curves.

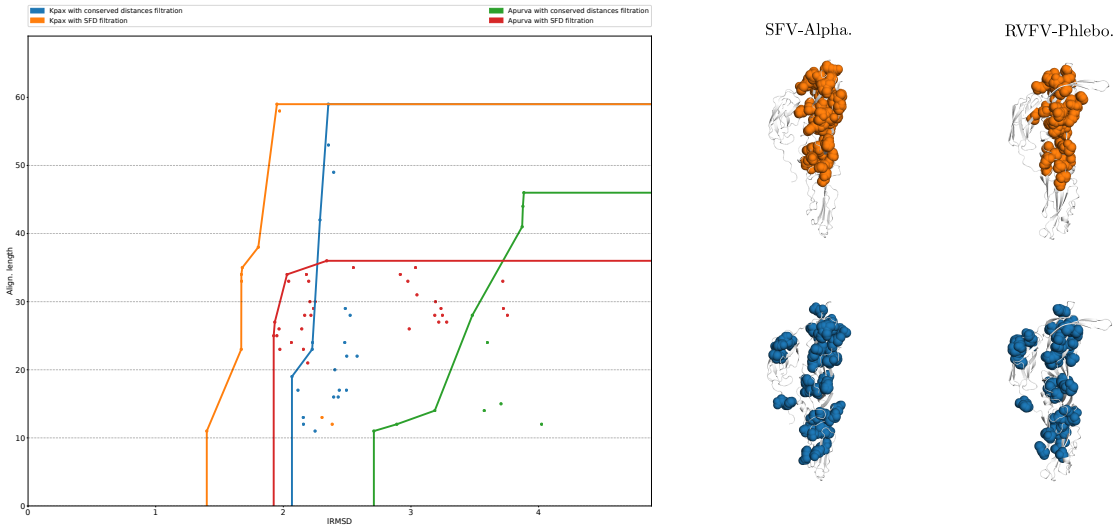
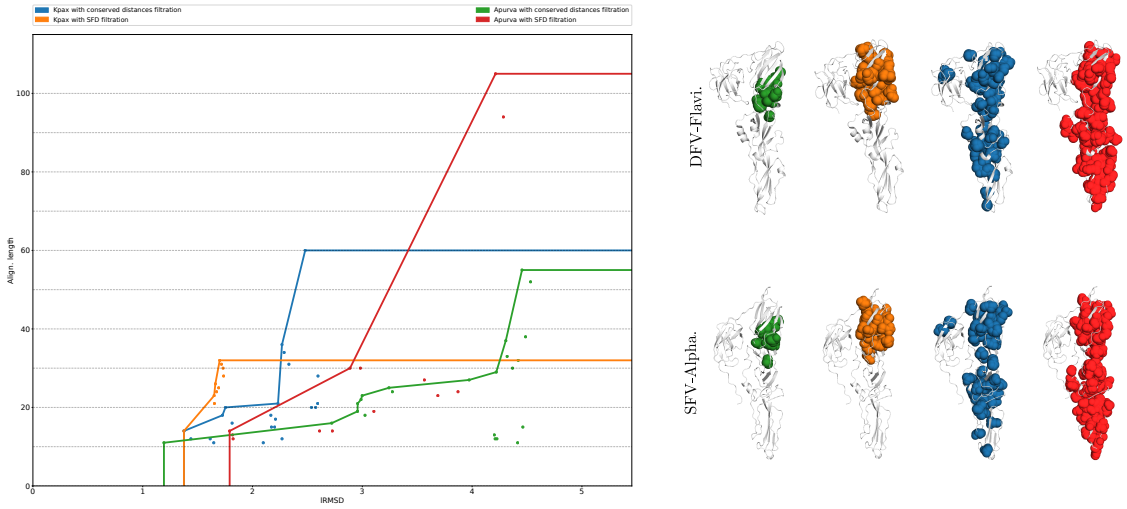


Figure 4.10 Comparison of DFV-Flavi. and SFV-Alpha.: (Left) Point clouds and Pareto envelopes of structural motifs found with each method. Initially, `Align-Apurva-CD` (Green) dominates until a critical point is reached and `Align-Kpax-CD` (Blue) takes over. **(Right)** Visualization of motifs corresponding to the corner points of each method.



Chapter 5

Functional characterization of proteins with low sequence identity and loose structural conservation

The previous chapters – through new distance measures and the systematic detection of structurally conserved regions which we call *structural motifs* – dealt mainly with protein structure. This sets the stage for the following chapter, in which we try to exploit harvested structural information to build sequence models. This is deeply rooted in the structure-function paradigm.

We present a sequence-structure based method characterizing a set of functionally related proteins exhibiting low sequence identity and loose structural conservation. Given a (small) set of structures, our method consists of three main steps. First, pairwise structural alignments are combined with multi-scale geometric analysis to produce structural motifs i.e. regions structurally more conserved than the whole structures. Second, the sub-sequences of the motifs are used to build profile hidden Markov models (HMM) biased towards the structurally conserved regions. Third, these HMM are used to retrieve from `UniProtKB` proteins harboring signatures compatible with the function studied, in a bootstrap fashion.

We apply the method to class II viral fusion proteins, an especially challenging case since known structures exhibit low sequence identity (less than 15%) and loose structural similarity (of the order of 15Åin lRMSD). We show that our hybrid HMM retrieve species not found by sequence based HMM, and vice-versa. The number of unique species reported by both classes of methods are comparable, stressing the importance of our hybrid models.

In a more general setting, our method should be of interest for all cases functional families with low sequence identity and loose structural conservation.

Our software tools are available from the `FunChaT` package of the Structural Bioinformatics Library (`http://sbl.inria.fr`).

5.1 Material

Viruses. Recall that the central goal in this thesis revolves around viral fusion proteins (Chapter 1). In this study, we use 6 viral fusion proteins that range from four different families of viruses: Togaviridae, Hantaviridae, Phenuiviridae and Flaviviridae. Hantaviridae and Phenuiviridae belong to the same order Bunyavirales. The others do not currently have an assigned order (Fig. 5.1).

Eukaryotes. Recent work allowed the identification of two class II fusion proteins in Eukaryotes: EFF1 [PVKV$^+$14], HAP2 [FLPA$^+$17, FFL$^+$18]. We add all the known crystallized structures to our study which include a nematod (C. Elegans), and an algae (C. Reihnardtii) (Fig. 5.1).

Groupings. As outlined in Introduction, our method extracts conserved structural motifs so as to produce biased profile HMM. In order to assess the incidence of these conserved motifs and the resulting HMM on the the sequences retrieved from `UniProtKB`, we run our methods on various groups:

- **Group 1:** All Togoviridae (2 structures).
- **Group 2:** All Flaviviridae (2 structures).
- **Group 3:** All Bunyavirales (2 structures).
- **Group 4:** All Eukaryotes (2 structures).
- **Group 5:** One of each kind of virus genus wise (5 structures).
- **Group 6:** EFF-1 and selected viruses (4 structures).
- **Group 7:** HAP2-C and selected viruses (4 structures).
- **Group 8:** One member of each taxonomic family (5 structures).

Each of these groups is scrutinized with respect to their overall sequence identity and structural similarity. We report the minimum, mean and maximum value of sequence identity and lRMSD for each pairwise comparison in a group. We form three super-groups (Fig. 5.1(c)):

- **Pairs:** there are four groups (1-4) which contain only two structures. Among the four, group 2 deserves a mention for its very high sequence identity (39.59%) and very low lRMSD (2.08). Flaviviridae are very conserved.
- **Mildly heterogeneous:** Group 5 and 6 have a medium variability in sequence identity ($[14.81\%, 19.6\%]$ and $[15.01\%, 19.16\%]$ respectively).
- **Heterogeneous:** Group 7 and 8 are the most heterogeneous groups when it comes to sequence identity ($[14.14\%, 22.33\%]$ and $[12.72\%, 19.16\%]$ respectively).

More specifically, **Groups 1-5** use taxonomic groupings, while **Groups 6-8** maximize heterogeneity by balancing members of the two similarity groups.

5.2 Method: hybrid profile HMM design and database search

5.2.1 Overview

We characterize a set of functionally related proteins with known structures in three steps (Fig. 5.2):

- (Step 1) *Structural motifs* defined from pairwise structural alignments are collected. A motif harbors a lRMSD significantly smaller than that associated with a structural alignment between its two defining structures–the ratio between these two lRMSD is called the *lRMSD ratio*.
- (Step 2) Sub-sequences associated to motifs together whole protein sequences are used to build multiple sequence alignments (MSA). Combining sub-sequences and whole sequences is meant to bias the MSA towards the structurally conserved regions–deemed important, while yet retaining information on linkers connecting these regions. The MSA are then used to build profile HMM.
- (Step 3) HMM are used to query `UniProtKB`. The obtained hits are filtered, so as to retain the sequences with properties related to the function studied.

5.2.2 Step 1: Structural motifs

Given a set $\mathcal{S} = \{S_1, \ldots, S_N\}$ of N polypeptide chains, we define motifs on a pairwise basis.

Structural motifs for two structures. A large lRMSD between two structures possibly obliterates regions of smaller size and with a smaller lRMSD . To find such regions, given two structures, we define:

Definition. 5.1. *Consider two structures S_i and S_j, and assume that a structural alignment between them has been computed. Given the two sets of a.a. identified by this alignment, i.e. $M_i \subset S_i$ and $M_j \subset S_j$, we define the* least RMSD ratio *as follows:*

$$r_{lRMSD}(M_i, M_j) = lRMSD(M_i, M_j)/lRMSD(S_i, S_j). \tag{5.1}$$

The sets M_i and M_j are called structural motifs *provided that $|M_i| = |M_j| \geq \tau_{MS}$ and $r_{lRMSD}(M_i, M_j) \leq \tau_{/}$, for appropriate thresholds τ_{MS} and $\tau_{/}$.*

A black box returning such motifs is taken for granted. A generic framework to identify motifs was recently proposed in a companion paper [CT18b]. The strategy hinges on two ingredients, namely an initial alignment providing a distance difference matrix (DDM), and a topological analysis of so-called filtrations coding conserved distances in the structures. We tested the four instantiations developed in Chapter 4, and obtained similar results (data not shown).

In the sequel, we report results obtained with two methods, namely `Align-Kpax-CD` (uses `Kpax` [RGMV12] as aligner; builds a filtration from conserved distances), and `Align-Kpax-SFD` (uses `Kpax` [RGMV12] as aligner; builds a filtration from a space filling diagram).

Thresholds used. In the sequel, we use $\tau_{\text{MS}} = 20$ and vary $\tau_{/}$ in the range $0.5 \ldots 0.8$. Additionally, we define *nuggets*, motifs with the more stringent thresholds $\tau_{\text{MS}} = 20$ and $\tau_{/} = 0.5$.

Remark 5.1. *Note that for a given structure, a structural motif is not necessarily connected, neither on the structure, nor on the sequence. This stems from the fact that a motif is defined upon performing a structural alignment(Chapter 4).*

Structural motifs for N structures. We collect motifs for all pairs of comparisons. Sorting those with lRMSD ratio less than the threshold $\tau_{/}$ yields the following list:

$$\mathcal{R}_S \equiv \{r_1, \ldots, r_{k-1}, r_k, r_{k+1}, \ldots, r_{\#r}\}, \text{ with } r_j \leq r_{j+1} \text{ and } r_{\#r} \leq \tau_{/}. \tag{5.2}$$

5.2.3 Step 2: From multiple sequence alignments to profile HMM

Parameterized consensus sequences To exploit the sequence information of the motifs, we define a set of nested sub-sequences, parameterized by the lRMSD ratio found in the sorted list $\mathcal{R}_S$ of Eq. (5.2):

Definition. 5.2. *(PCS) Consider a structure $i \in \mathcal{S}$ together with a lRMSD ratio $r \in \mathcal{R}_S$. The* parameterized consensus sequence *$PCS^i_{\leq r}$ associated with this structure is defined as the sequence of this structure, into which every amino-acid position not involved in any motif with lRMSD ratio less then r is replaced by a gap. The set of all* parameterized consensus sequences *is denoted PCS.*

Central in our method is the notion of PCS. On the one hand, stringent thresholds enforcing structural conservation are expected to yield motifs which may be too specific; on the other hand, too lenient thresholds may yield motifs which may lack specificity. Additionally, the structures present in $\mathcal{S}$ condition the motifs retrieved; in turn, the consensus sequences associated with these motifs are expected to be more or less specific of these sequences. This precisely motivates the definition of the various groups of interest (section 5.1).

Multiple sequence alignments and profile HMM. Using whole sequences and PCS, we define two sets of Multiple Sequence Alignment (MSA): $\text{MSA}^{\text{Seq.}}$, a MSA involving the input protein sequences only; $\text{MSA}^{\text{Hyb.}}$, a MSA involving the union of the protein sequences and the parameterized consensus sequences PCS. Practically, we use two multiple sequence aligners: ClustalΩ [SWD+11] and `MUSCLE` [Edg04].

We exploit these MSA using profile hidden Markov models [KBM+94, Edd98] and the `HMMER` implementation [FCE11, Edd15]. That is, to study the bias imposed by the addition of PCS to the full sequences, we define $\mathrm{HMM}^{\mathrm{Seq.}}$ based upon $\mathrm{MSA}^{\mathrm{Seq.}}$, and $\mathrm{MSA}^{\mathrm{Hyb.}}$ based upon $\mathrm{HMM}^{\mathrm{Hyb.}}$. These HMM models, whose complexity is measured by the number of match states, are used to perform database queries.

5.2.4 Step 3: Database queries and filtering

Database queries. A HMM is used to query `UniProtKB`. Such a query return *hits* in the form of `UniProtKB` accesion codes (unique identifiers tied to a protein sequence). The significance of hits being assessed with e-values (SI C.1.3). For a given hit, we retrieve the taxonomic information from the `NCBI Taxonomy` database [Fed12] (SI Sect. C.1.3.) Note that we focus on species rather than protein sequences, since fragments, isoforms, or variants of a given protein typically correspond to separate entries.

Filtering hits using annotations: generic and fusion class II specific. The hits obtained can also be filtered using various criteria, which we illustrate for class II fusion proteins.

A first filter identifies transmembrane proteins. To find a transmembrane region in a hit from its FASTA sequence, we use `Phobius` [KKS04]. The second filter aims at identifying class II candidates, based on three conditions: (i) at least one transmembrane region, (ii) an ectodomain (protein region extending into extracellular space) involving ≥ 400 residues, (iii) at least six cysteins in the ectodomain.

Cross-validation using HMM-HMM comparisons. Sequences obtained from `UniProtKB` can be re-scored via HMM-HMM comparison (`HHpred`, [Söd04]; see also Section 2.2.2). This strategy being of special interest to check the coherence with a HMM associated with a structure, we practically launch a query on the target database PDB_mmCIF70 (see `https://toolkit.tuebingen.mpg.de/#/tools/hhpred`).

Remark 5.2. *With respect to our overarching goal, which is to annotate sequences from* `UniProtKB`*, note that we use* `HMMER` *rather than* `HHpred`*, since since* `HHpred` *searches selected databases (PDB,* `Pfam`*, SMART, ...) rather than* `UniProtKB`*. See* `https://toolkit.tuebingen.mpg.de/#/tools/hhpred`*.*

Remark 5.3. *We note that different filtering criteria for class II fusion proteins could be:*

- *one conserved disulfide bond*
- *lots of beta strands: use some SSE predictors and find out a ratio of residues belonging to beta strands*
- *beta rich region in a 200 residue window from the TM-region.*
- *an Ig-like domain close to the TM-region (30 to 50 residues away). This involves finding a way to recognize Ig-like domains (HMM or pattern).*

Additionally, DIII being an Ig-like domain can add a lot of noise to the models. One could devise a filtering scheme in two steps:

1. *Build a HMM using DI and DII and query databases.*
2. *Build a new HMM using DIII and filter the results by scoring the previous hits with the new model.*

Bootstrap. The previous calculation may be integrated into a bootstrap strategy, with the sequences of the filtered hits incorporated so as to define new HMM (SI Fig. C.3). We apply this strategy by running three bootstrap iterations on each previously defined groups with $\mathrm{HMM}^{\mathrm{Hyb.}}$ and $\mathrm{HMM}^{\mathrm{Seq.}}$. Note in passing that only sequences which yield an e-value ≤ 0.01 are considered for the bootstrap step.

Comparison of $\mathrm{HMM}^{\mathrm{Hyb.}}$ and $\mathrm{HMM}^{\mathrm{Seq.}}$: protocol. Summarizing the previous discussion, we compare $\mathrm{HMM}^{\mathrm{Hyb.}}$ and $\mathrm{HMM}^{\mathrm{Seq.}}$ as follows: (i) Fix a group of input sequences and structures, (ii) Fix a threshold $\tau_{/}$ to define structural motifs – Eq. (5.2)), (iii) Iteratively build the HMM models, perform the queries, count the species yielded, (iv) Bootstrap.

5.2.5 Implementation and software

The software implementing our methods is available in the Structural Bioinformatics Library [CD17] at `http://sbl.inria.fr`). The two main packages involved are Structural_motifs for the detection of motifs (`https://sbl.inria.fr/doc/Structural_motifs-user-manual.html`), and FunChaT for the functional characterization of proteins (`https://sbl.inria.fr/doc/FunChaT-user-manual.html`).

5.3 Results

5.3.1 Structurally conserved motifs

On the structural conservation of SSE. Class II proteins have a hierarchical structure, with SSE defining 23 structural units (SI Table C.1), and a natural question is therefore to check whether the simplest structural elements, i.e. SSE, harbor any structural information. As established using hierarchical clustering methods, the short answer is no (SI Sec. C.1.3; SI Figs. DI: Fig. C.4, DII: C.5, DIII:C.6). This prompts for further analysis seeking structural conservation beyond SSE.

Motifs: structure and sequence conservation. To go beyond SSE, we used structural motifs yielded by the aforementioned algorihtm `Align-Kpax-CD` [CT18b]. With thresholds $\tau_{\text{MS}} = 20$ and $\tau_{/} = 0.8$, our method detects 188 structural motifs with sizes ranging from 20 to 116 (SI section C.1.4) and lRMSD ranging from 0.63 to 10.73. Note that there can be redundancies (motifs that show little variation with respect to their constituting residues – typically less than 5). In using more stringent values ($\tau_{\text{MS}} = 20$ a.a. and lRMSD ratio $\tau_{/} = 0.5$), 118 structural motifs remain. Out of the 118, we handpicked 28 to minimize redundancies (SI Table C.1 and SI Fig. C.7). These nuggets are characterized by a size range $20 \dots 67$ residues, and lRMSD ratio range $0.09, \dots, 0.5$ (SI Table C.1).

From a functional standpoint, motifs may contribute indirectly (e.g. in defining the fold) or indirectly to the function of class II proteins. Various functional features of such proteins have indeed been characterized [WDBS08, Kie14, Har15, GCR17]. Of critical importance is the hydrophobic fusion loop which is inserted into the target membrane, as well as the disulfide bonds stabilized the two loops emanating from the central domain I.

Closer inspection of our motifs reveals two important features (Fig. 5.3). On the one hand, several motifs sandwich (half)disulfide bonds. On the other hand, a motif typically spans several SSE elements (Fig. 5.3(Bottom)). This shows the difficulty of identifying such structurally conserved regions, as selecting the combination of SSEs together with their sub-components faces a combinatorial explosion–which we handle using geometric and topological techniques [CT18b].

5.3.2 Performances of hybrid HMMs for sequence retrieval

As a first assessment, since our hybrid HMM exploit sequence and structure information, we compare them to pure sequence based HMM to identify relevant sequences within `UniProtKB`. By varying the learning set and the threshold used to define structural motifs, we compare the species containing class II candidates retrieved over iterations. For these experiments, `MUSCLE` [Edg04] is used to build the multiple sequence alignments.

More specifically, we compare the number of species identified both by $\text{HMM}^{\text{Hyb.}}$ and $\text{HMM}^{\text{Seq.}}$ (dark blue), those exclusively identified by $\text{HMM}^{\text{Hyb.}}$ (orange), and those identified by $\text{HMM}^{\text{Seq.}}$ (light blue) (Fig. 5.4, Sect. 2.2.2). Additionally, we investigate the complexity of the models (Table 5.2). Alongside the number of species, the hatched bars correspond to the number of emit states in a given HMM.

General comments are in order:

- In general, both models provide specific information i.e. species that are not found by the other model (Fig. 5.4; all figures, SI Sect. C.1.5).

- Overall, both methods report comparable number of species (Table 5.1), with $\text{HMM}^{\text{Hyb.}}$ slightly more stable than $\text{HMM}^{\text{Seq.}}$ (std. dev. of 271 versus 290). This overall observation hides cases where each model alternatively dominates the other in a stable fashion (Fig. 5.4(A,B) and cases where both models fare equivalently (Fig. 5.4(C)).

- Bootstrap iterations are either characterized by a sharp rise of the number of species reported (Fig. 5.4(D,E)), or by a rather stable behavior (Fig. 5.4(A,B,C)).

- The performances of $\text{HMM}^{\text{Hyb.}}$ are conditioned to the set of motifs used, which is itself parameterized by the ratio $\tau_{/}$ – Eq. (5.2). Stringent threshold indeed result in a smaller bias imposed by structural motifs. At threshold ($\tau_{/} = 0.5$), it appears that only group 3 and 7 enjoy unique species identified (SI Fig. C.9). Beyond that threshold, no clear rule emerges to select $\tau_{/}$ (SI Figs. C.10, C.11, C.12).

- In terms of model complexity, both $\text{HMM}^{\text{Hyb.}}$ and $\text{HMM}^{\text{Seq.}}$ fare equally. However, $\text{HMM}^{\text{Hyb.}}$ is, on average, slightly smaller at the initial step, and grows slightly larger in the final stages (Table 5.2). From the first bootstrap step and on-wards, the model sizes nearly double.

To further these insights, we inspects characteristic scenarios (Fig. 5.4).

- Even though both types of models are relatively stable, the complexity of the model may *explode* at a given iteration, inducing a net increase in the number of species found and also potentially noise. The implementation of HMM may be unable to handle a large number of emit states (Fig. 5.4(A), $\text{HMM}^{\text{Hyb.}}$, 3rd iteration).

- Adding or removing sequences from a model can have any type of effect. Addition of new sequences to expand the MSA and the associated HMM is an expected behavior. But the opposite is also observed. Consider the case where the initial calculation is such that $\text{HMM}^{\text{Hyb.}}$ does not retrieve any specific sequence (Fig. 5.4(D), leftmost column). Therefore, the sequences used to build the MSA for $\text{HMM}^{\text{Hyb.}}$ are a strict subset of the sequences used to build the MSA for $\text{HMM}^{\text{Seq.}}$. Yet, at the first bootstrap iteration, $\text{HMM}^{\text{Hyb.}}$ retrieves more species than $\text{HMM}^{\text{Seq.}}$ (Fig. 5.4(D), first column).

Remark 5.4. *Upon investigation some hits do not meet the requirements of our filter because the sequence is partial–regions containing the cysteins and/or the trans-membrane region may be missing. This reflects poorly on group 1, which displays very little species. In reality, both HMMs are able to recover most of the species of its corresponding taxonomic group.*

5.3.3 Performance of hybrid HMMs to retrieve homologs of HAP2-GCS1

We noted in Introduction that fusogens in viruses and eukaryotes evolve at different speeds, due in particular to the selection pressure imposed to the former by immune systems. As a second assessment of our HMMs, we therefore narrow down our focus on eukaryotes, and study the ability of our hybrid HMM to identify homologs of HAP2-GCS1.

Species. With a focus on eukaryotes irrespective of the differences between EFF-1 and HAP2-C , we further investigate performances of both HMM models using Group 4 at threshold $\tau_{/} = 0.7$. This group was chosen because the HAP2-GSC1 family is currently of high interest; a general aim is to find members of this family among larger organisms, such as vertebrates. The $\tau_{/} = 0.7$ threshold seems to yield the best results for $\text{HMM}^{\text{Hyb.}}$. We wish to estimate how well the models characterize the HAP2-GSC1 protein family. To do this, we exploit `UniProtKB` sequence annotations (`http://www.uniprot.org/help/sequence_annotation`) to find hits which have been labeled as having a HAP2-GSC1 domain. At the initial step, $\text{HMM}^{\text{Hyb.}}$ recovers 254 hits spread across 149 species; $\text{HMM}^{\text{Seq.}}$ finds 244 hits (2 of which are exclusive) across 145 species. At the third and final bootstrap iteration, $\text{HMM}^{\text{Hyb.}}$ 228 hits across 126 species; $\text{HMM}^{\text{Seq.}}$ finds 298 hits across 167 species. In this case, $\text{HMM}^{\text{Hyb.}}$ finds 12 exlusive hits across 10 species. The following remarks are in order:

- A very small training set is enough to recover many homologs (for both models).
- Initially $\text{HMM}^{\text{Hyb.}}$ performs better at finding HAP2-GSC1 family members.
- After the bootstrap iteration, $\text{HMM}^{\text{Seq.}}$ performs better although $\text{HMM}^{\text{Hyb.}}$ still has exclusive species.

Note that this tally is not exhaustive as some proteins that are known HAP2-GSC1 family members are not annotated. For example, the Tetrabaena socialis HAP2 protein (`UniProtKB` accession code A0A2J8AI85), does not have the HAP2-GSC1 domain annotation. Even though it is a know HAP2 protein, it will not show up in the HAP2-GSC1 domain count. Additionally, members of HAP2-GSC1 in larger organisms (such as vertebrates) are theorized to be distant homologs. A model that is too specific could be a hindrance to finding such a protein so that loosing some hits in the further steps should not be necessarily seen as an obstacle torward that goal.

Homologs of HAP2-GCS1. To further constrain the models, we restrict the learning set to the domains II of three HAP2-GSC1 structures known at the time of this study. The three structures accession numbers are 5MF1 (HAP2e) [FLPA+17], 5OW3 (AtHAP2) and 5OW4 (TcHAP2) [FFL+18]. Using motifs at threshold $\tau_{/} = 0.8$, we build a new $\text{HMM}^{\text{Hyb.}}$ (SI Fig. C.8 for its sequence logo.) When results are displayed, the particular method used is indicated (by method we refer to `Align-Kpax-CD`, `Align-Kpax-SFD`, see Section 5.2.2, and the multiple sequence alignment used).

To assess the ability of our method to find remote homologs, we check for the recovery of the most distant known homolog to our training set, the HAP2-GCS1 in the drosophila fly. It was indeed shown [Gar12] that D. melanogaster gene CG34027, an ortholog of HAP2-GCS1, codes for a protein involved in plasma membrane fusion. Different combinations of filtration methods as well as sequence aligners yield different results (Tab. 5.3). From the first iteration, we recover a number of known HAP2 sequences, notably in arthropodes (34 species, data not shown). From the first bootstrap iteration, we recover 17 plausible HAP2-GCS1 candidate sequences in 10 different Drosophila species. We cross validate these results using `HHpred` (Sect. 5.2.4 on HMM-HMM comparisons). Of particular interest are comparisons against the known gamete fusion protein in chlamydomonas reinhardtii (HAP2 structure, pdbid 5MF1). Eleven of the sequences returned a very low e-value leading to the conclusion that they are most probably HAP2 proteins (Tab. 5.3). One of these corresponds to the aforementioned gene CG34027 (`UniProtKB` identifier: Q2PDQ0).

Importantly, none of these eleven sequences mentions the `Pfam` domain PF10699 (`https://pfam.xfam.org/family/PF10699`). This fact illustrates the ability of our structural models to capture structural information which is more localized than whole `Pfam` domains.

5.4 Discussion and outlook

Remote homologous proteins are proteins sharing low sequence identity, yet having similar structures and functions. As the development of sequencing techniques yields a rapid increase in the number of known sequences, while the number of solved structures grows more slowly, the ability to detect remote homology is a central problem in structural bioinformatics. Our work contributes a novel method for this problem, combining sequence-structure information. As an application, we focus on the problem of identifying within `UniProtKB` sequences which might be class II fusion proteins with high probability. Using a diverse learning set involving structures from viruses and eukaryotes, we show that our hybrid HMM models retrieve proteins from species which are not identified from pure sequence based HMM. Moreover, given the relative diversity of structures for class II fusion proteins, we also show that using a narrower learning set involving eukaryotic structures only, our method identifies remote homologs in D. melanogaster, which are not retrieved using the relevant PFAM domain.

To discuss the merits of our method, it is informative to consider in turn the three types of information a remote homology detection method may enjoy: sequences, structures, and dynamic information.

When sequences and/or structures of proteins with a common function are known, the classical route consists in modeling these sources of information with multiple sequence alignments and/or profile HMM

and/or ad hoc feature spaces. Along this line, our contribution is to show that biasing profile HMM with structural features detected amongst a learning set (a handfull of structures) indeed helps to identify novel sequences compatible with the function targeted. Further cross-validation of the sequences retrieved via HMM-HMM comparison against witness HMM models with known structures provides a high level of confidence. However, this strategy is only partially satisfactory, since sequences that do not yield a low e-value via HMM-HMM comparison may still be of high interest.

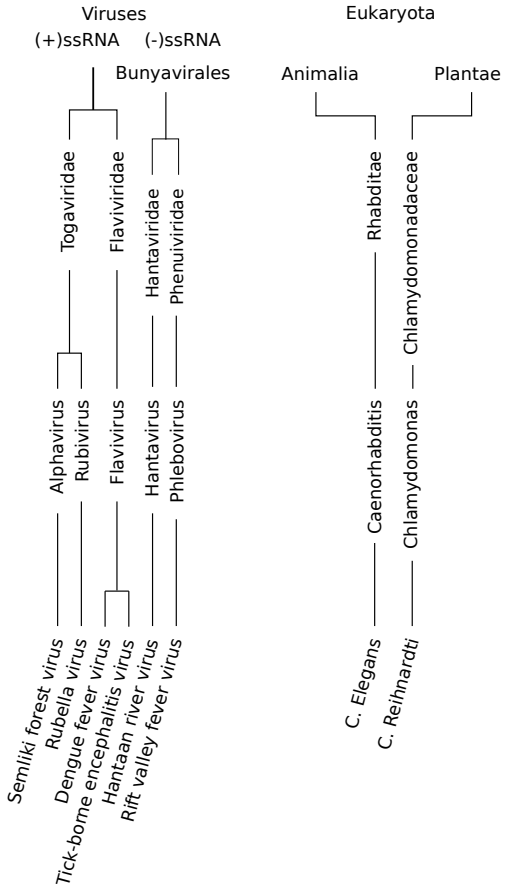
Our ability to retrieve remote homologs of HAP2 in drosophilia melanogaster stresses the relevance of the structural motifs underlying our hybrid HMM, which typically span portions of SSE elements, and whose detection is a non trivial endeavor. We also note that biasing profile HMM with structural information, albeit a straightforward idea, is a rather subtle strategy for two reasons. On the one hand, biased HMM models, when compared to pure sequence based models, do bring unique features–as evidenced by the fact that they are the only ones to identify selected sequences from `UniProtKB`; yet, both classes of models have unique traits since they do identify unique species. On the other hand, the learning set plays a crucial role in particular in terms of diversity, and qualifying the output as a function of this diversity remains an open problem.

So far, we have excluded from the design of remote homology detection methods information related to the dynamics, including structural (meta-stable states), thermodynamic (occupancy probabilities), and kinetic information (transition rates between states). In the presence of large amplitude conformational changes, this type of information is admittedly out of reach in most cases for experimental and simulation methods. However, amino-acids involved in key structural, thermodynamic or dynamic events would naturally help improving all types of remote homology detection methods (alignment methods, discriminative methods, ranking methods), and would also alleviate the aforementioned cross-validation step in the absence of obvious witnesses. More generally, such insights might dramatically reduce the region of sequence space in which distant homology is sought. In fact, in moving from sequence to dynamics across structures, one uses finer mechanism related information, which is however more challenging to get. Finding the optimal combination appears as a very promising research avenue, calling for deep insights on the connexions between sequence, structure, dynamics and function.

5.5 Artwork

Figure 5.1 Structures used in this study. a) Embedding of each structure in their respective taxonomic tree (one for viruses and one for eukaryotes). We only detail the names for the genus and family ranks. The viruses are arranged in groups and the eukaryotes in kingdoms. **b)** Here we provide the files used in the study as well as the acronym used for each structure throughout this article. **c)** The groups of structures as presented in Sec. 5.1. For each group, we display pairwise sequence identity statistics as well as structural similarity. Regarding sequence identity, we denote three cases (which are color coded): pairs of structures (for which there is only one value), mildly heterogeneous groups (with a small interval of sequence identity values) and heterogeneous groups.

a)



b)

Viruses		
Name	PDB code	Acronym
Semliki Forest virus	1RER	SFV
Dengue fever virus	1OK8	DFV
Tick-borne encephalitis virus	1URZ	TBEV
Hantaan river virus	5JLX	HRV
Rift valley fever virus	6EGT	RVFV
Rubella virus	4ADI	RBV
Eukaryota		
C.Elegans	4OJC	EFF-1
C. Reihnardtii	5MF1	HAP2-C

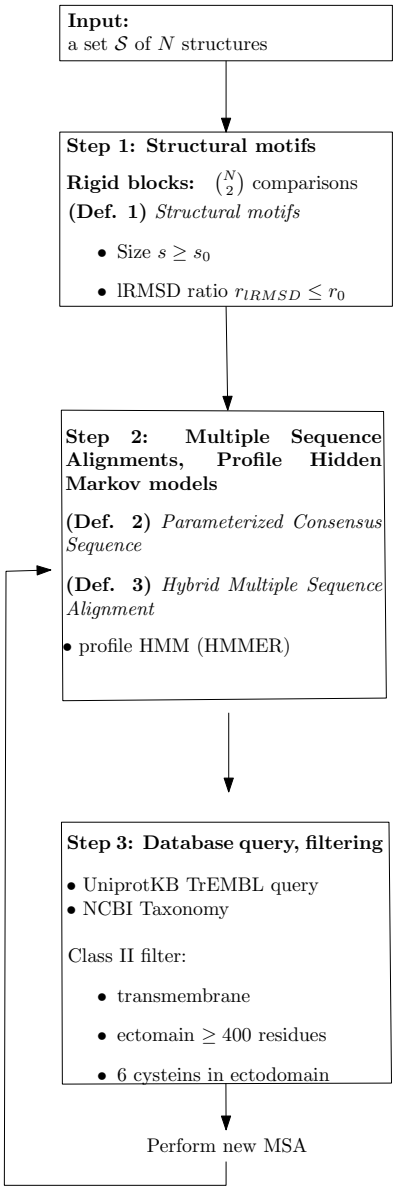
c)

		Seq. Id. (%)	lRMSD
Group 1	SFV, RBV	21.88	15.97
Group 2	DFV, TBEV	39.59	2.08
Group 3	RVFV, HRV	18.73	15.03
Group 4	HAP2-C , EFF-1	21.16	7.73
Group 5	DFV, RVFV, HRV, SFV, RBV	min: 14.81	5.32
		mean: 16.12	1.33
		max: 19.16	15.97
Group 6	HRV, SFV, RBV, EFF-1	min: 15.01	6.86
		mean: 16.65	12.73
		max: 19.16	16.57
Group 7	HRV, SFV, RBV, HAP2-C	min: 14.14	6.86
		mean: 18.07	13.19
		max: 22.33	18.01
Group 8	HRV, SFV, RBV, HAP2-C , EFF-1	min: 12.72	6.86
		mean: 15.83	12.65
		max: 22.33	18.01

Pairs
Mildly heterogeneous
Heterogeneous

Figure 5.2 Sequence-structure based characterization of functionally related proteins: workflow.

We study N structures with low sequence identity and loose structural homology, yet harboring the same function.



We identify structurally conserved motifs, characterized by a least RMSD ratio smaller than yielded by the whole structures.

Sub-sequences of motifs and whole sequences are used to build multiple sequence alignments. Next, these MSA are used to build profile hidden Markov models (HMMER).

HMM are used to query the Uniprot database. Taxonomic information for a hit is obtained through the NCBI Taxonomy database.
Hits are filtered through custom criterions related to the function studied. Here we present the filter we used for Class II fusion protein candidates. Note that we can also filter on taxonomic criteria.

The method can be bootstrapped by using the sequences of filtered hits to build a new MSA.

Main outputs:

For each comparaison pair:

- Hasse diagram of motifs
- Statistics (distance plot, score plot, ...)
- Statistical significance of motifs

- Alignment file (optional)
- HMM file (optional)
- Query results file (optional)

List of accession codes for all proteins which meet the filter requirements

Figure 5.3 Comparing RVFV-Phlebovirus to DFV-Flavivirus: structural motifs. (Top) Motif represented with a solvent accessible model; the motif is localized on the tip of DII, which contains several disulfide bonds. **(Bottom)** Zoom on the motif, displaying the motif itself (red and blue amino-acids, respectively, on the two molecules), and the disulfide bonds within the motif.

RVFV-Phlebo. DFV-Flavi.

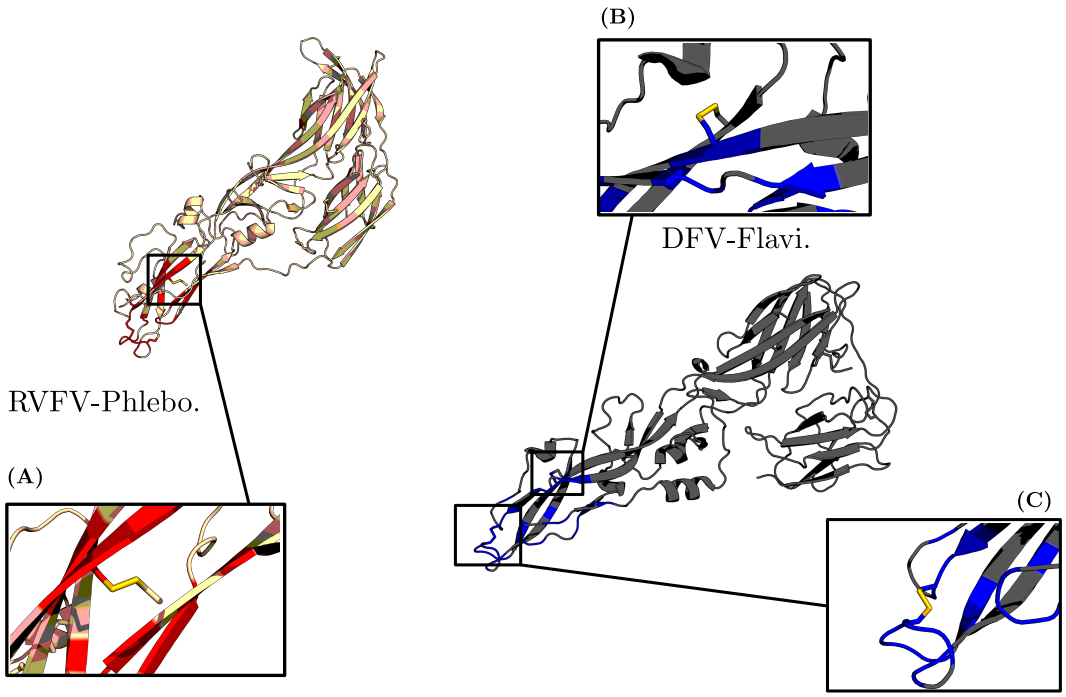


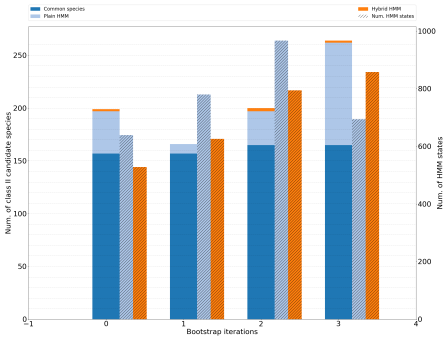
Table 5.1 Retrieved species: statistics over the four runs and all $\tau_/$ thresholds determining motifs. Statistics are reported over four runs i.e. the initial run + three bootstrap iterations; three values of parameter $\tau_/$, which determines structural motifs, were used: $\tau_/ = (0.5, 0.6, 0.7, 0.8)$. Species variations refers to the variation in-between two consecutive runs.

Model	# species		# species variation	
	Mean	σ	Mean	σ
$HMM^{Hyb.}$	284	271	103	188
$HMM^{Seq.}$	283	290	118	222

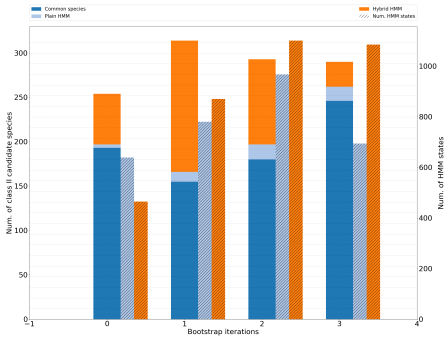
Table 5.2 HMM complexity: size of the model i.e. number of emit states on each bootstrap iteration. Initially, $HMM^{Hyb.}$ is slightly smaller and more stable than $HMM^{Seq.}$. In later stages, the opposite behavior is observed.

Model	HMM num. states							
	Initial model		First bootstrap		Second bootstrap		Third bootstrap	
	Mean	σ	Mean	σ	Mean	σ	Mean	σ
$HMM^{Hyb.}$	479	59	1319	886	1383	754	1275	530
$HMM^{Seq.}$	558	100	1351	876	1356	757	1163	328

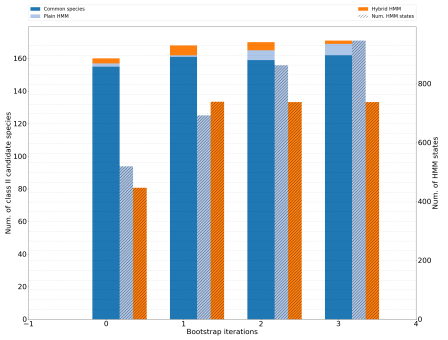
Figure 5.4 Various scenarios of domination when bootstrapping HMMs. For each iteration (0 to 3, x-axis), the 3 bars read as follows: first bar: species found by $\text{HMM}^{\text{Seq.}}$ and $\text{HMM}^{\text{Hyb.}}$ (solid blue), species found by $\text{HMM}^{\text{Seq.}}$ only (light blue), species found by $\text{HMM}^{\text{Hyb.}}$ only (orange); second bar: hatched light blue: number of emit states of $\text{HMM}^{\text{Seq.}}$; third bar (hatched orange): number of emit states of $\text{HMM}^{\text{Hyb.}}$. **(A)** $\text{HMM}^{\text{Seq.}}$ consistently dominates $\text{HMM}^{\text{Hyb.}}$. **(B)** $\text{HMM}^{\text{Hyb.}}$ consistently dominates $\text{HMM}^{\text{Seq.}}$. **(C)** Both types of HMM yield comparable number of specific species. **(D)** At each bootstrap iteration, $\text{HMM}^{\text{Hyb.}}$ shows a large increase in number of species until the model becomes too complex and the HMM implementation used fails to manage it. **(E)** $\text{HMM}^{\text{Seq.}}$ displays a peak number of species at the second bootstrap iteration.



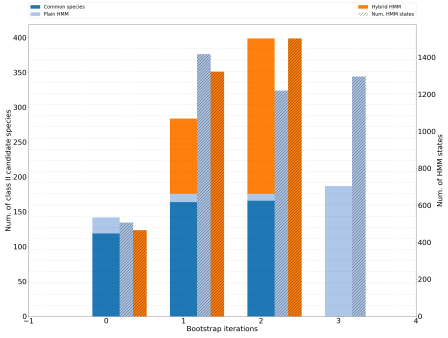
(A) Group 8, $\tau_{/} = 0.5$



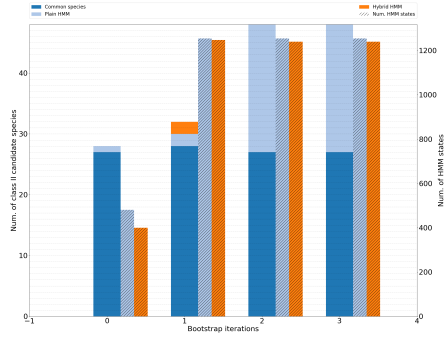
(B) Group 8, $\tau_{/} = 0.8$



(C) Group 6, $\tau_{/} = 0.7$



(D) Group 3, $\tau_{/} = 0.5$



(E) Group 1, $\tau_{/} = 0.7$

Table 5.3 Searching for remote HAP2-GSC1 homologs: hits in the drosophilia fly. Cross-validation of the sequences yielded by our method –see Section 5.2.4. Reported are the top hits obtained with `HHpred` (3rd column), with small e-values indicating a likely HAP2 protein (4th column). The pdbids associated with the hits are also given; those corresponding to known HAP2 structures are marked in bold.

Species	`UniProtKB` accession	`HHpred` top hit (PDB ID)	e-value
`Align-Kpax-CD`, ClustalΩ			
Drosophila ananassae	B3M263	5MF1	3e-51
Drosophila mojavensis	A0A0Q9X6U7	5MF1	8.7e-55
Drosophila virilis	A0A0Q9WDI4	5MF1	2.1e-53
`Align-Kpax-CD`, Muscle			
Drosophila mojavensis	A0A0Q9X6U7	5MF1	8.7e-55
Drosophila virilis	A0A0Q9WDI4	5MF1	2.1e-53
`Align-Kpax-SFD`, ClustalΩ			
Drosophila ananassae	B3M263	5MF1	3e-51
Drosophila mojavensis	A0A0Q9X6U7	5MF1	8.7e-55
Drosophila virilis	A0A0Q9WDI4	5MF1	2.1e-53
`Align-Kpax-SFD`, Muscle			
Drosophila erecta	B3P6S0	5MF1	2.7e-39
Drosophila sechellia	B4IJG1	5MF1	2.7e-39
Drosophila ananassae	B3M263	5MF1	3e-51
Drosophila melanogaster	Q2PDQ0	5MF1	6.5e-31
Drosophila mojavensis	A0A0Q9X6U7	5MF1	8.7e-55
Drosophila mojavensis	A0A0Q9WXE1	5MF1	1.5e-61
Drosophila mojavensis	A0A0Q9WWX9	5MF1	1.4e-67
Drosophila yakuba	B4PUD6	5MF1	1.3e-32
Drosophila mojavensis	A0A0Q9WX02	5MF1	1.6e-45
Drosophila ficusphila	A0A1W4VL29	5MF1	1.8e-52
Drosophila virilis	A0A0Q9WDI4	5MF1	2.1e-53
Drosophila virilis	A0A0Q9WEW4	5IJ0	3.7
Drosophila ananassae	A0A0P8YDK7	3S84	10
Drosophila mojavensis	A0A0Q9X9J0	3S84	9.3
Drosophila mojavensis	B4KU79	3S84	8.4
Drosophila simulans	A0A0J9RKJ6	3S84	8.5
Drosophila pseudoobscura pseudoobscura	Q28WV3	3S84	9.5

Chapter 6

On the stability of clusterings: the D-Family matching problem

Recall from Chapter 4 that to identify structural motifs in two conformations of the same protein we use filtrations built from a structural conservation score. Upon processing these filtrations, we record the persistence of connected components in the two conformations. We then compare points from the two resulting persistence diagrams (Figure 6.1 a). Each point has a number of connected components associated to it (the sub-level sets). This prompts the question of which comparisons to instantiate? This situation is perfectly analogous to clustering comparison (6.1).

In this next chapter, we take a break from structural biology and venture into the realm of clustering.

Clustering is a fundamental problem in data science, yet, the variety of clustering methods and their sensitivity to parameters make clustering hard. To analyze the stability of a given clustering algorithm while varying its parameters, and to compare clusters yielded by different algorithms, several comparison schemes based on matchings, information theory and various indices (Rand, Jaccard) have been developed. We go beyond these by providing a novel class of methods computing meta-clusters within each clustering– a meta-cluster is a group of clusters, together with a matching between these.

Let the intersection graph of two clusterings be the edge-weighted bipartite graph in which the nodes represent the clusters, the edges represent the non empty intersection between two clusters, and the weight of an edge is the number of common items. We introduce the so-called D-family-matching problem on intersection graphs, with D the upper-bound on the diameter of the graph induced by the clusters of any meta-cluster. First we prove NP-completeness and APX-hardness results, and unbounded approximation ratio of simple strategies. Second, we design exact polynomial time dynamic programming algorithms for some classes of graphs (in particular trees). Then, we prove spanning-tree based efficient algorithms for general graphs.

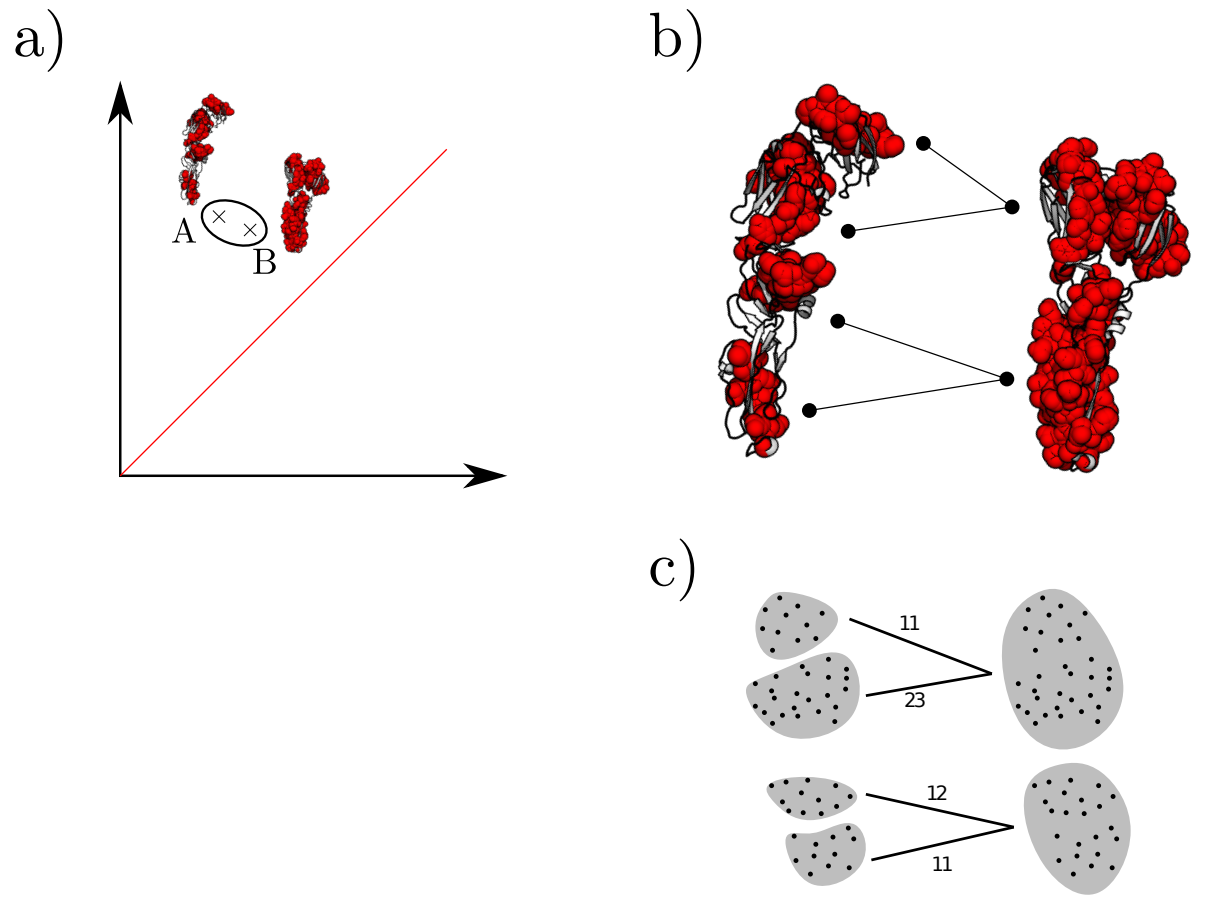
Our experiments illustrate the role of D as a scale parameter providing information on the relationship between clusters within a clustering and in-between two clusterings. They also show the advantages of our built-in mapping over classical cluster comparison measures such as the variation of information (VI).

6.1 Comparison of clusterings: formalization as graph problems

In this section, we formalize the D-family-matching problem modeling the comparison of clusterings. Let $t \geq 1$ be any positive integer. Let us consider a set of elements $Z = \{z_1, \ldots, z_t\}$. We are given two different families F and F' of disjoint subsets of Z. Let $r \geq 1$ be the size of F. Formally $F = \{F_1, \ldots, F_r\}$, where $F_i \subseteq Z$, $F_i \neq \emptyset$, and $F_i \cap F_j = \emptyset$ for every $i, j \in \{1, \ldots, r\}$, $i \neq j$. Let $r' \geq 1$ be the size of F'. In a analogous way, $F' = \{F'_1, \ldots, F'_{r'}\}$, where $F'_i \subseteq Z$, $F'_i \neq \emptyset$, and $F'_i \cap F'_j = \emptyset$ for every $i, j \in \{1, \ldots, r'\}$, $i \neq j$.

Definition. 6.1 (Edge-weighted intersection graph)**.** *The edge-weighted intersection graph $G = (U, U', E, w)$ associated with Z, F, and F', is constructed as follows. The set $U = \{u_1, \ldots, u_r\}$ corresponds to the*

Figure 6.1 From comparing persistence diagrams to clustering comparison. a) Two comparable points in the persistence diagrams obtained upon using the method presented in Chapter 4 to find structural motifs in the two conformations (A and B) of the Dengue fever virus class II fusion protein. **b)** Upon observing the two sublevel sets, one notices that most connected components on the right are nested in the ones on the left. Finding a way to detect this would yield an elegant way for instantiating comparisons. **c)** This is analogous to comparing two clusterings of the same dataset.



clustering F. To each vertex u_i, we associate the set $F_i \in F$. The set $U' = \{u'_1, \ldots, u'_{r'}\}$ corresponds to the clustering F'. To each vertex u'_i, we associate the set $F'_i \in F$. The set of edges of G is $E = \{\{u_i, u'_j\} \mid F_i \cap F'_j \neq \emptyset, 1 \leq i \leq r, 1 \leq j \leq r'\}$. The weight of any edge $e = \{u_i, u'_j\} \in E$ is $w_e = |F_i \cap F'_j|$.

In the rest of the paper we will write intersection graph instead of edge-weighted intersection graph and $w_{u,u'}$ instead of $w_{\{u,u'\}}$ to denote the weight of an edge $\{u, u'\} \in E$. See Figure 2.12 and Figure D.1 in appendix (Section D.1) for two detailed examples. We prove in Lemma 6.1 that any edge-weighted bipartite graph G with positive integers, is an intersection graph for some Z, F, and F'. A bipartite graph is a graph whose nodes can be partitioned into two disjoint sets such that every edge has an extremity in the first set and has its other extremity in the second set.

Lemma. 6.1. *Let $G = (V, E, w)$ be any edge-weighted bipartite graph such that $w_e \in \mathbb{N}^+$ for every $e \in E$. Then, there exist Z, F, and F' for which G is the intersection graph.*

By Lemma 6.1, we can focus on any intersection graph without necessarily considering the corresponding

Z, F, and F'. In the rest of the paper, an intersection graph will be denoted $G = (V, E, w)$. Let us define some notations. We denote by $n = |V|$ the number of nodes of G, by $m = |E|$ the number of edges of G, and by $\Delta = \max_{v \in V} |N_G(v)|$ the maximum degree of G, where $N_G(v)$ is the set of neighbors of $v \in V$ in G. The diameter of a graph is the maximum number of edges of a shortest path in this graph. The set $cc(G)$ represents the set of maximal connected components of G. We now define the notion of D-family-matching.

Definition. 6.2 (D-family-matching)**.** *Let $D \in \mathbb{N}^+$. Let $G = (V, E, w)$ be an intersection graph. A D-family-matching for G is a family $\mathcal{S} = \{S_1, \ldots, S_k\}$, $k \geq 1$, such that, for every $i, j \in \{1, \ldots, k\}$, if $i \neq j$, then: $S_i \subseteq V$, $S_i \neq \emptyset$, $S_i \cap S_j = \emptyset$, and the graph $G[S_i]$ induced by the set of nodes S_i has diameter at most D.*

The score $\Phi(\mathcal{S})$ of a D-family-matching $\mathcal{S}$ is $\Phi(\mathcal{S}) = \sum_{i=1}^{k} \sum_{e \in E(G[S_i])} w_e$. Let $\mathcal{S}_D(G)$ be the set of all D-family-matching for G. We now formalize the D-family-matching problem. Intuitively, we wish to compute a D-family-matching which minimizes the inconsistencies.

Definition. 6.3 (D-family-matching problem)**.** *Let $D \in \mathbb{N}^+$. Given an intersection graph G, the D-family-matching problem consists in computing $\Phi_D(G) = \max_{\mathcal{S} \in \mathcal{S}_D(G)} \Phi(\mathcal{S})$.*

From such an optimal solution, we deduce an optimal number of sets $k \geq 1$ for our clusterings comparison. Observe that the 1-family-matching problem is equivalent to the problem of computing a maximum weighted matching in weighted bipartite graphs. Since this problem can be solved in $O(n^2 \log n + nm)$ [FT87], we deduce the following result.

Lemma. 6.2. *Given any intersection graph G, the 1-family-matching problem can be solved in $O(n^2 \log n + nm)$.*

6.2 Hardness of the D-family-matching problem and greedy strategies

In this section, we prove that the D-family-matching problem is NP-complete and that simple strategies can be arbitrarily bad. All the proofs can be found in appendix (Section D.4).

As explained before, the 1-family-matching problem is polynomial-time solvable, thus we now focus on higher values of D. Moreover, we will prove in Section 6.3 that the problem is polynomial-time solvable for bipartite graphs of maximum degree $\Delta = 2$. We prove that these two cases are actually the only pairs (D, Δ) leading to polynomial problems, all other being NP-complete and even APX-hard. We were also able to prove hardness results for some (D, Δ) even in the case where edge weights are within a fixed range of values.

Theorem. 6.1. *Let $D \geq 2$ be any integer. The decision version of the D-family-matching problem is APX-hard for :*

- *bipartite graphs of maximum degree 3;*
- *bipartite graphs of maximum degree 4 when the maximum weight is constant.*

Moreover, the 2-family-matching problem is APX-hard for bipartite graphs of maximum degree 3 with unary weights.

Although we leave as an open question whether D-family-matching is in APX (namely, whether it admits a polynomial-time algorithm achieving a constant approximation ratio), we show that two natural strategies for obtaining approximation algorithms are hopeless.

Perhaps the most natural idea to solve the problem is a greedy approach, which would iteratively seek for maximal collections of subgraphs of diameter D. Unfortunately, we show that even for $D = 2$, there exist instances (having unary weights) for which picking a maximum biclique (complete bipartite graph) first leads to an arbitrarily bad solution.

Lemma. 6.3. *For any integer $\lambda \geq 1$, there exists an intersection graph $G = (V, E, w)$ such that $\Phi_2(G)/\Phi(\mathcal{S}_{bic}) \geq \lambda - 2$, where $\mathcal{S}_{bic}$ is an optimal solution for the 2-family-matching problem among those containing a maximum biclique. Such a graph has unary weights, is of maximum degree λ, and contains $\lambda(\lambda - 1) + 1$ vertices.*

Notice that this result is in sharp contrast to the non-bipartite version of the problem, where we want to find a partition of the vertices of a graph in cliques which covers the maximum number of edges. It has been shown [CMWY09] that a greedy algorithm similar to the one described previously achieves a 2-approximation ratio if one is allowed to pick cliques of size at most some constant r only. Hence, this strategy gives a 2-approximation algorithm in graphs of maximum degree Δ, while it cannot achieve an approximation ratio better than Δ in the bipartite case. This gives the intuition that the complexity of the problem increases with the value of the diameter.

From this observation, another greedy strategy consists in first solving the problem with a smaller value of D. Here again, we show that such an algorithm cannot achieve a fixed approximation ratio. More precisely, we analyze the ratio between scores of optimal solutions for the D-family-matching problem for increasing values of the diameter, that is $\Phi_D(G)/\Phi_{D'}(G)$ for $D' < D$. Unfortunately, we show that this ratio is not bounded even for very simple classes of instances

Lemma. 6.4. *For any integer $n \geq 1$, then there exists an intersection graph $G = (V, E, w)$ composed of n nodes such that $\Phi_2(G)/\Phi_1(G) \geq n - 1$.*

In the next sections, we provide polynomial-time algorithms for simple graphs classes, and then use some of them to obtain efficient algorithms in general bipartite graphs.

6.3 Polynomial time dynamic programming algorithms for some classes

In this section, we prove polynomial-time complexity exact dynamic programming algorithms for the D-family-matching problem for some classes of graphs: trees, paths, cycles, graphs of maximum degree two. All the proofs can be found in appendix (Section D.5). In the following, we explain our exact polynomial time dynamic programming algorithm to solve the D-family-matching problem when the graph is a tree.

Theorem. 6.2 (Computation of $\Phi_D(G)$ for trees)**.** *Let $D \in \mathbb{N}^+$. Consider any intersection tree $T = (V, E, w)$ of maximum degree $\Delta \geq 0$. Then, there exists an $O(D^2\Delta^2 n)$-time complexity algorithm for the D-family-matching problem for T.*

Proof. Consider the tree T rooted at any node $r \in V$. We call this rooted tree T_r. Given any node $v \in V$, let T_v be the sub-tree of T_r rooted at v such that $V(T_v)$ contains all the nodes $v' \in V$ such that there is a simple path between v' and r in T_r that contains v in T_r. A simple path is a path such that each node is contained at most once in it. We define the function Ψ_D as follows. For every $v \in V$ and every $i \in \{-1, 0, \ldots, D\}$, then $\Psi_D(T_v, i)$ is the score of an optimal solution $\mathcal{S}$ for the D-family-matching problem, for the intersection tree T_v, such that:

- if $i \geq 0$, then there exists $S \in \mathcal{S}$, $v \in S$, and the sub-tree induced by the set of nodes S has depth at most i;
- if $i = -1$, then for every $S \in \mathcal{S}$, we have $v \notin S$.

Note that $\Psi_D(T_v, 0)$ is the score of an optimal solution $\mathcal{S}$ when $\{v\} \in \mathcal{S}$ (say otherwise, v is alone in a set). In the following, we abuse the notation writing $\Psi_D(v, i)$ instead of $\Psi_D(T_v, i)$.

First of all, for every leaf $v \in V$ of T_r and every $i \in \{-1, 0, \ldots, D\}$, then $\Psi_D(v, i) = 0$. A leaf is a node of degree one and different than the root r.

Let $v \in V$ be any node that is not a leaf. Let $N(v) = \{v_1, \ldots, v_q\}$ be the set of $q \geq 1$ neighbors of v in T_v. Suppose we have computed $\Psi_D(v_j, i)$ for every $j \in \{1, \ldots, q\}$ and every $i \in \{-1, \ldots, D\}$. We prove that we can compute $\Psi_D(v, i)$ for every $i \in \{-1, \ldots, D\}$. The computation is divided into two different cases (claims).

Claim 6.3. *For every $i \in \{-1, 0\}$, then*

$$\Psi_D(v, i) = \sum_{j \in \{1,\dots,q\}} \max_{i \in \{0,\dots,D\}} \Psi_D(v_j, i).$$

Proof of Claim 6.3. Let us first consider $i = -1$. We consider here an optimal solution for the D-family-matching problem for T_v such that v does not belong to any set. Thus, we compute for every sub-tree T_{v_j}, $1 \leq j \leq q$, the score of an optimal solution for the D-family-matching problem for T_{v_j}. Note that the depth of the sub-tree (set) rooted at v_j in the solution has no importance here. The score of such a score is $\max_{i \in \{0,\dots,D\}} \Psi_D(v_j, i)$. Then, $\Psi_D(v, -1)$ is the sum of all such scores.

Let us now consider $i = 0$. It means that we consider an optimal solution for the D-family-matching problem for T_v such that v is alone in a set. Observe that $\Psi_D(v, 0) = \Psi_D(v, -1)$. □

Claim 6.4. *For every $i \in \{1, \dots, D\}$, then*

$$\Psi_D(v, i) = \max_{j \in \{1,\dots,q\}} (\Psi_D(v_j, i-1) + w_{v,v_j} +$$

$$\sum_{j' \in \{1,\dots,q\} \setminus \{j\}} \max(\max_{i' \in \{1,\dots,\min(i-1, D-i-1)\}} \Psi_D(v_{j'}, i') + w_{v,v_{j'}}, \max_{i' \in \{1,\dots,D\}} \Psi_D(v_{j'}, i'))).$$

Proof of Claim 6.4. We compute here the score $\Psi_D(v, i)$ of an optimal solution for the D-family-matching problem for T_v such that the depth of the sub-tree (set) that contains v in the solution is exactly i. We denote S_v the set of nodes of such a sub-tree. To do that, we first need to choose one sub-tree T_{v_j}, for some $j \in \{1, \dots, q\}$, such that the set (sub-tree) that contains v_j in the solution for T_v, is such that the sub-tree induced by $S_v \cap V(T_{v_j})$ has depth $i-1$. In order to compute such j, we enumerate the q different possibilities. For every possible choice ($j = 1, \dots, q$), we compute the largest possible score. Such a score is $\Psi_D(v_j, i-1)$ plus the the weight w_{v,v_j} of the edge $\{v, v_j\}$ plus the largest possible score for the other neighbors of v. More precisely, for every $j' \in \{1, \dots, q\}$, $j' \neq j$, there are two cases.

- $S_v \cap V(T_{v_j}) = \emptyset$. In that case, the largest possible score corresponding to the sub-tree $T_{v_{j'}}$ is $\max_{i' \in \{1,\dots,D\}} \Psi_D(v_{j'}, i')$.
- $S_v \cap V(T_{v_j}) \neq \emptyset$. In that case, the largest possible score is $\max_{i' \in \{1,\dots,\min(i-1, D-i-1)\}} \Psi_D(v_{j'}, i') + w_{v,v_{j'}}$. Indeed, we add the weight $w_{v,v_{j'}}$ by assumption and we then compute the score of an optimal solution for the D-family-matching problem for $T_{v_{j'}}$ such that $v_{j'}$ is in a sub-tree (set) of depth at most $D-i-1$ and also at most $i-1$. Otherwise, the diameter of S_v would be at least $D+1$ and/or the depth would be at least $i-1$.

We determine the maximum score between these two scores. We finally obtain an optimal score and we determine a best choice for j in order to compute $\Psi_D(v, i)$. □

For every $v \in V$, the time complexity of the computation of $\Psi_D(v, i)$, for all $i \in \{-1, \dots, D\}$, is $O(qD)$ for the first case and $O(q^2 D^2)$ for the second case. We get that the time complexity of the algorithm is $O(D^2 \Delta^2 n)$. Note that $\Delta \leq n-1$ and $D \leq n-1$. Finally, when we have computed $\Psi_D(r, i)$ for every $i \in \{-1, \dots, D\}$, we can deduce an optimal solution $\mathcal{S}$ for the D-family-matching problem for T. Indeed, $\Phi(\mathcal{S}) = \Phi_D(G) = \max_{i \in \{-1,\dots,D\}} \Psi_D(r, i)$. □

Using similar ideas, we obtained the following results, whose proof can be found in appendix (Section D.5).

Theorem. 6.5. *For any $D \in \mathbb{N}^+$, the D-family-matching problem can be solved:*

- *in $O(Dn)$ time if G is a path;*
- *in $O(D^2 n)$ time if G is a cycle(s) or a graph of maximum degree 2.*

Notice finally that the problem can be solved in $O(|cc(G)| \max_{C \in cc(G)} f(C))$ for G if D-family-matching can be solved in $O(f(C))$ time for any $C \in cc(G)$, where $cc(G)$ denotes the set of maximal connected components of G.

Algorithm 1 Generic algorithm for the D-family-matching problem.

Require: An intersection graph $G = (V, E, w)$, an integer $D \geq 1$, a property Π, a spanning tree generator $\mathcal{R}$, and an algorithm $\mathcal{A}$.

1: $\mathcal{M} := \emptyset$, $\lambda := 0$
2: **while** $\neg\, \Pi(\mathcal{M})$ **do**
3: $\quad \lambda := \lambda + 1$; Compute the spanning tree $T^\lambda := \mathcal{R}(G, \lambda)$
4: $\quad$ Compute $\mathcal{S}^\lambda$ by using Algorithm $\mathcal{A}(G, T^\lambda, D)$; $\mathcal{M} := \mathcal{M} \cup \mathcal{S}^\lambda$
5: **return** $\mathcal{S} \in \mathcal{M}$ of maximum score

6.4 Generic approach based on spanning trees

In this section, we provide a generic approach for solving the problem in general instances. All proofs can be found in appendix (Section D.6). This approach relies on computing a solution having a particular structure defined by a given spanning tree T of the input graph. Formally:

Definition. 6.4 (D-family-matching constrained by a tree)**.** *Let $D \in \mathbb{N}^+$. Let $G = (V, E, w)$ be an intersection graph and T be a spanning tree of G. A D-family-matching for G constrained by T is a D-family-matching $\mathcal{S}$ for G such that all $S \in \mathcal{S}$ induces a connected subtree in T.*

We thus obtain the following sub-problem of D-family-matching.

Definition. 6.5 (D-family-matching problem constrained by a tree)**.** *Let $D \in \mathbb{N}^+$. Given an intersection graph G and a spanning tree T of G, the D-family-matching problem consists in computing $\Phi_D(G, T) = \max_{\mathcal{S} \in \mathcal{S}_D(G,T)} \Phi(\mathcal{S})$, where $\mathcal{S}_D(G, T)$ is the set of all D-family-matching constrained by T.*

We are now ready to define our generic algorithm (Algorithm 1). Informally, it iteratively generates a spanning tree T of G, and compute a D-family-matching constrained by T. Let us describe the main ingredients of Algorithm 1 by explaining the three parameters needed.

- **A property** $\Pi(\mathcal{M})$, depending on the set $\mathcal{M}$ of already computed D-family-matchings, represents the halting condition of the algorithm.
- **A spanning tree generator** $\mathcal{R}(G, \lambda)$ computes the rooted spanning tree T^λ of G that is used at step $\lambda \geq 1$ by Algorithm $\mathcal{A}$.
- **An algorithm** $\mathcal{A}(G, T^\lambda, D)$ computes a D-family-matching $\mathcal{S}^\lambda$ constrained by T^λ.

The interest of this approach is twofold. The first one is the fact that solving optimally the D-family-matching constraint by T, for every spanning tree T, leads to an optimal solution of the general D-family-matching problem. This is the point of the following result.

Lemma. 6.5. *Let $D \in \mathbb{N}^+$. Let G be any intersection graph. Then, there exists a rooted spanning tree T of G such that $\Phi_D(G) = \Phi_D(G, T)$.*

Then, we show that it is possible, given a spanning tree T, to compute an optimal D-family-matching constrained by T in an efficient way, provided the diameter D and the maximum degree Δ of the input graph are bounded by a constant.

Lemma. 6.6 (Computation of $\Phi_D(G, T)$)**.** *Let $D \in \mathbb{N}^+$. Let $G = (V, E, w)$ be any intersection graph and T be any spanning tree of G. Then, there exists a $O(2^{D\Delta \log_2(\Delta)} n)$-time algorithm for the D-family-matching problem for G constrained by T.*

Finally, we obtain an efficient (polynomial) heuristic using our dynamic programming algorithm for trees described in Theorem 6.2. Indeed, an optimal solution for the D-family-matching problem on a spanning tree T is a D-family-matching for G constrained by T. We present an implementation of this heuristic and results of experiments in the next section.

6.5 On the choice of D

6.5.1 Rationale

As seen in the previous sections, parameter D acts as a scale parameter providing information of the structure of the intersection graph. When this graph is dense or has a specific topology (star-shaped), trivial values of Φ are obtained for small values of D, and a unit change of D may trigger an abrupt change of Φ. However, in more complex situations, large values of D may be required.

As a general strategy to choose D, we suggest identifying drops in Φ when changing D. Indeed, a solution is to find range(s) of D corresponding to a *plateau* for Φ, in order to get subset of diameters that give similar Φ. Phrased differently, consider a set of pairs $\{(D, \Phi_D)\}$ for consecutive values of D. In the 2D space (D, Φ_D), consider the bounding box of these points: the longer and the thinner this bounding box, the better. In the sequel, we present three strategies to compute such plateaus:

- Section 6.5.2: a strategy computing a set of non-overlapping plateaus optimizing a functional favoring *long* and *thin* plateaus.
- Section 6.5.3: a strategy allowing the user to specify the number of plateaus to be obtained.
- Section 6.5.4: a strategy delivering a hierarchical decomposition into plateaus, which is of interest if there are several *vertical scales*.

We note in passing that optimizing the width and/or height of plateaus is similar in spirit to other strategies which were developed in particular to estimate the *correct* number of clusters: the elbow method tracks jumps of the derivative of the objective function [KP13]; the gap method uses the change in within-cluster dispersion [TWH01]; the split-and-merge procedure for `k-means` [AA13] requires a parameterization of high-density areas; the integrated completed likelihood (ICL) method [BRC$^+$10] tracks significant drops in the entropic penalty complementing the BIC criterion; the density based - mode seeking method exploits a separation gap (in the persistence diagram) for local maxima [CGOS13], etc.

6.5.2 Computation of tradeoff-plateaus of large widths and small heights

We first develop a strategy to compute plateaus of large width and small height. More precisely, we develop a quadratic time algorithm for the problem of computing the diameter D that minimizes a tradeoff between the widths and the heights of the plateaus.

Definition. 6.6 (Tradeoff-plateau problem)**.** *Let τ_w and τ_h be two increasing functions of one variable such that $\tau_w(y) > 0$ and $\tau_h(y) > 0$ for any y. Given a set $\{\Phi_1(G), \ldots, \Phi_{D_G}(G)\}$, the Tradeoff-plateau problem consists in determining $\mu \in \{1, \ldots, D_G\}$ plateaus (intervals) $I_1, \ldots, I_\mu$ of $[1, D_G]$ such that $I_1 \cup \ldots \cup I_\mu = \{1, \ldots, D_G\}$, $I_x \cap I_{x'} = \emptyset$ for every $1 \leq x < x' \leq \mu$, and such that the following function is minimum:*

$$-\sum_{x=1}^{\mu} \frac{\tau_w(|I_x|)}{\tau_h(\max_{D,D' \in I_x \cap \mathbb{N}} \Phi_{D'}(G) - \Phi_D(G))},$$

where $|I_x|$ is the size of plateau I_x.

Note that as the simplest choices for τ_w and τ_h, one can simply use two simple linear functions of the width and height of the bounding box containing the consecutive points $\{(D, \Phi_D)\}$ (recall that we must have $\tau_h(y) > 0$). This yields the following objective function

$$-\sum_{x=1}^{\mu} \frac{|I_x| - 1}{\max_{D,D' \in I_x \cap \mathbb{N}} \Phi_{D'}(G) - \Phi_{D+1}(G) + 1}.$$

The following theorem proves in a constructive way, using dynamic programming, that the computation of such plateaus can be done in quadratic time (proof in Section D.7):

Theorem. 6.6. *There is an $O(D_G^2)$-time complexity algorithm that computes an optimal solution for the Tradeoff-plateau problem.*

Once plateaus have been computed, the *plateau plot* displays one rectangle (containing the corresponding points (D, Φ_D)) for each plateau. Since gaps between plateaus are of interest, we define:

Definition. 6.7. *(Increment in a plateau plot) The increment of a plateau is defined as the mean value of its scores, minus the average value computed on the preceding plateau.*

We note that a natural choice for the value of D is the last plateau with a *significant* increment – in a spirit analogous to the strategies recalled in section 6.5.1.

Remark 6.1. *In practice, the range of D values explored is set to $1, \ldots, D_G$, with D_G the diameter of the intersection graph. When $D = D_G$ an exact algorithm solving the D-family matching problem would return a number of meta-clusters corresponding to the number of connected components of the intersection graph – the exact solution. However, since we only use a heuristic based on spanning trees, at $D = D_G$, the number of meta-clusters may be over-estimated. However this is not an issue since for $D = D_G$, the exact solution is known.*

6.5.3 Computation of multiple sets of plateaus of small heights

We develop an algorithm computing an optimal set of μ plateaus in terms of sum of heights (of the plateaus) for every possible value for $\mu \in \{1, \ldots, D_G\}$.

Definition. 6.8 (Plateau problem)**.** *Let $G = (V, E)$ be a graph of diameter D_G. Given a set $\{\Phi_1(G), \ldots, \Phi_{D_G}(G)\}$ and a number of plateaus $\mu \in \{1, \ldots, D_G\}$, the Plateau problem consists in determining μ plateaus (intervals) $I_1, \ldots, I_\mu$ of $[1, D_G]$ such that $I_1 \cup \ldots \cup I_\mu = \{1, \ldots, D_G\}$, $I_x \cap I_{x'} = \emptyset$ for every $1 \leq x < x' \leq \mu$, and such that the following function is minimum:*

$$\sum_{x=1}^{\mu} \max_{D, D' \in I_x \cap \mathbb{N}} \Phi_{D'}(G) - \Phi_D(G).$$

Theorem. 6.7. *There is an $O(D_G^5)$-time complexity algorithm that computes an optimal solution for the Plateau problem for every $\mu \in \{1, \ldots, n\}$.*

Our dynamic programming algorithm computes for every $D \in \{1, \ldots, D_G\}$, for every $y \in \{1, \ldots, D\}$ and for every $x^-, x^+ \in \mathcal{P}_D = \{\Phi_1(G), \Phi_2(G), \ldots, \Phi_D(G)\}$ with $x^- \leq x^+$, the optimal solution $\rho_{y,x^-,x^+}(D)$ for the sub-problem defined by the set $\Phi_1(G), \Phi_2(G), \ldots, \Phi_D(G)$, a number y of plateaus, and such that $x^- = \min_{x \in I_y \cap \mathbb{N}} x$ and $x^+ = \max_{x \in I_y \cap \mathbb{N}} x$ (I_y is the last plateau of the optimal solution for this sub-problem, that is the current plateau if we consider the original instance of the plateau problem). In other words, $\rho_{y,x^-,x^+}(D)$ is the optimal solution for the sub-problem induced by the first D values of Φ, such that the number of plateaus is y and the last plateau has minimum value x^- and maximum value x^+. If there is no admissible solution, we set $\rho_{y,x^-,x^+}(D) = \infty$.

To prove Theorem 6.7, we first prove in Lemma 6.7 the correctness of the computation of $\rho_{y,x^-,x^+}(D+1)$.

Lemma. 6.7. *First, $\rho_{1,\Phi_1(G),\Phi_1(G)}(1) = 0$. For every $D \in \{1, \ldots, D_G - 1\}$, for every $y \in \{1, \ldots, D+1\}$, for every $x^-, x^+ \in \mathcal{P}_{D+1}$ with $x^- \leq x^+$, then:*

- *If $\Phi_{D+1}(G) \in]x^-, x^+[$, then $\rho_{y,x^-,x^+}(D+1) = \rho_{y,x^-,x^+}(D)$;*
- *If $\Phi_{D+1}(G) = x^-$ and $x^- < x^+$, then $\rho_{y,x^-,x^+}(D+1) = \min_{x \in \mathcal{P}_{D+1}, x^- \leq x \leq x^+}(\rho_{y,x,x^+}(D) + x - x^-)$;*
- *If $\Phi_{D+1}(G) = x^+$ and $x^- < x^+$, then $\rho_{y,x^-,x^+}(D+1) = \min_{x \in \mathcal{P}_{D+1}, x^- \leq x \leq x^+}(\rho_{y,x,x^+}(D) + x^+ - x)$;*
- *If $x^- = x^+ = \Phi_{D+1}(G)$, then $\rho_{y,x^-,x^+}(D+1) = \min(\rho_{y,x^-,x^+}(D), \min_{x,x' \in \mathcal{P}_{D+1}, x \leq x'}(y-1, x, x'))$;*

Algorithm 2 Construction of the tree representing the hierarchical plateaus.

Require: The values $(\Phi_1(G), \dots, \Phi_{D_G}(G))$.

The set of nodes is $V = L \cup U$, where $L = \{v_1, \dots, v_{D_G}\}$ and $U := \{u_1, \dots, u_{D_G-1}\}$.
$d := D_G$, $x := 1$, $E := \emptyset$.
Let $(I_1, I_2, \dots, I_d)$ be the $d = D_G$ initial plateaus each composed of 1 point.
We associate a leaf node to every initial interval: $v(I_i) = v_i$ for every $i \in \{1, \dots, d\}$.
while $d \geq 2$ **do**
  $i' := \arg\min_{i \in \{1,\dots,d\}} \max_{D, D' \in I_i \cup I_{i+1}} \Phi_{D'}(G) - \Phi_D(G)$.
  $E := E \cup \{v(I_{i'}), u_x, v(I_{i'+1}), u_x\}$.
  $I_{i'} := I_{i'} \cup I_{i'+1}$ and $I_j := I_{j+1}$ for every $j \in \{i'+1, d-1\}$.
  $v(I_{i'}) = u_x$ and $v(I_j) = v(I_{j+1})$ for every $j \in \{i'+1, d-1\}$.
  $d := d - 1$, $x := x + 1$.
return $T = (V, E)$

- *If $\Phi_{D+1}(G) < x^-$ or $\Phi_{D+1}(G) > x^+$, then $\rho_{y,x^-,x^+}(D+1) = \infty$ (there is no solution).*

The proof of Lemma 6.7 can be found in Section D.7. By Lemma 6.7, the score of an optimal solution for every number $y \in \{1, \dots, D_G\}$ of plateaus is:

$$\min_{x^-, x^+ \in \mathcal{P}_D, x^- \leq x^+} \rho_{y,x^-,x^+}(D_G).$$

We finally prove the time-complexity of our dynamic programming algorithm in Lemma 6.8 (proof in Section D.7).

Lemma. 6.8. *The time complexity of the algorithm of Lemma 6.7 is $O(D_G^5)$.*

6.5.4 Hierarchical Plateaus

As a final interpretation of role of D, we perform a hierarchical construction of plateaus, in a spirit analogous to the construction of a dendogram.

More precisely, we construct the rooted tree $T = (V, E)$ representing the hierarchical plateaus as follows. There is a leaf per possible value of D and there are $D_G - 1$ internal nodes (including the root). Let $L = \{v_1, \dots, v_{D_G}\}$ be the set of leaves of T and let $U = \{u_1, \dots, u_{D_G-1}\}$ be the set of internal nodes of T. Let $V = L \cup U$. Algorithm 2 formally describes the set of edges of T.

6.6 Experiments

We present experiments in three directions:

- In Section 6.6.2, we compare a clustering and an edited version of it, in order to assess the value of D required to counter-balance the magnitude of edits.
- In Section 6.6.3, we investigate the role of D to compare clusterings yielded by `k-means++` and recover the *right* number of clusters when the number of clusters passed to `k-means++` is too large.
- In Section 6.6.4, we argue that a normalized score associated with our matchings provides more accurate information than the information theoretical measure variation of information (VI).

6.6.1 Implementation

Generic code. We implemented a version of Algorithm 1, which takes as input a graph $G = (V, E, w)$ and a diameter D. This implementation is integrated to the *Core / Combinatorial algorithms and data structures*

(CADS) component of the Structural Bioinformatics Library (`http://sbl.inria.fr`). Documentation can be accessed directly from `http://sbl.inria.fr/doc/D_family_matching-user-manual.html`. As it may be noticed, the main class T_Spanning_tree_solver which is a direct implementation of Algorithm 1, is templated by an intersection graph type and **(i)** a spanning tree generator $\mathcal{R}$, **(ii)** a stop condition ($\Pi\mathcal{M}$) and **(iii)** an algorithm $\mathcal{A}$.

Instantiation for our experiments. For the following experiments, we use an instantiation of the previous generic algorithm, $STS(G, D)$ which has the following ingredients: **(i)** the spanning tree generator $\mathcal{R}$ returns a *maximum spanning tree*, or a *random spanning tree*; **(ii)** the property $\Pi(\mathcal{M})$ returns true once we have computed a solution on the maximum spanning tree, as well as a solution on $n_i = (10,000)$ distinct random spanning trees (for a given n_i); **(iii)** $\mathcal{A}$ is the algorithm described in Theorem 6.2 (Section 6.3) with an additional step: edges for which both extremities belong to the same meta-cluster are added to the said meta-cluster. (In general, the intersection graph is indeed not a tree, so that such edges were unaccounted for.) The solution returned for a given graph G and a diameter D is the best yielded by the aforementioned $1 + n_i$ spanning trees.

The corresponding executable from the Structural Bioinformatics Library is sbl-d-family-matching.exe. Individual running times (< one minute on a laptop computer) are not further analyzed.

6.6.2 Experiments on random and edited clusterings

Rationale. We test our algorithm on pairs of clusterings (F, F'), with F a random clustering, and F' an edited version of F. The goal is to assess the ability of our algorithm to retrieve matchings such as the one of Figure 2.12, stressing the role of parameter D.

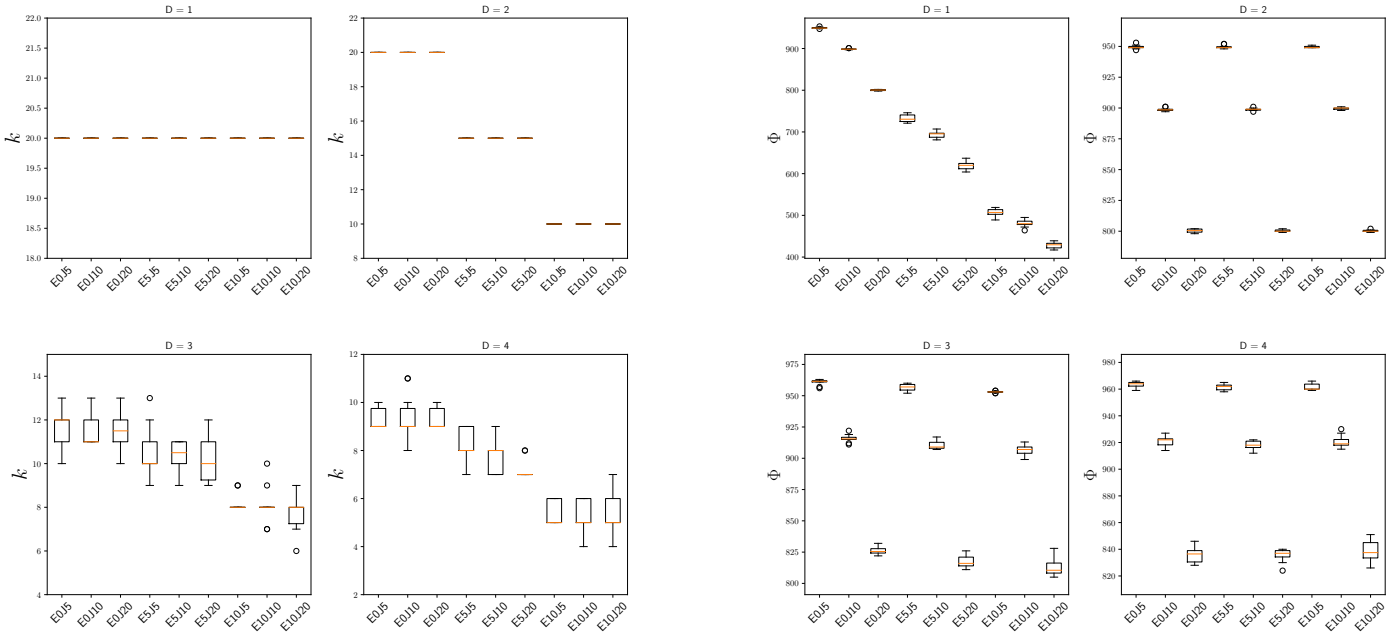
Dataset: random clusterings. The number of clusterings of a set Z of size t into r clusters is the number of distinct partitions of this set into r nonempty subsets. Its number is the Stirling number of second kind [GKP89]. Adding up all these numbers yields the number of partitions of the set Z into any number of subsets, which is the Bell number $B(t)$ [FS09]. Such clusterings were generated using a Boltzmann sampler [DFLS04, Example 5]. Since clustering usually aims at grouping data points into a relatively small number of clusters, two pairs of parameters were used ($t = 1\ 000, r = 20$) and ($t = 3\ 000, r = 50$). Both yielded similar results, so that we only report on the former. Due to the randomness, the process is repeated $N_r = 10$ times for each pair (t, r).

Dataset: edited clusterings. We build random pairs of clusterings (F, F') by copying F into F' and editing F', in two steps. First, we perform e union operations to reduce the number of clusters to $r - e$. Second, the elements of the remaining clusters are jittered: for each cluster, a fraction τ of its items are distributed amongst the remaining $k-1$ clusters uniformly at random. Practically, we take $e \in \{0, \lfloor r/4 \rfloor, \lfloor r/2 \rfloor\}$ and $\tau \in \{0.05, 0.1, 0.2\}$. Note that for $e = 0$, F' is a jittered version of F (i.e. the numbers of clusters are identical). Summarizing, this setup yields $N_r \times \#(t, r) \times \#e \times \#\tau = 180$ comparisons, which are ascribed to 9 scenarios (3 values for $e \times 3$ values for τ) denoted $EeJy$, where $y = 100\tau$.

Statistics. These 180 comparison pairs are fed to algorithm $STS(G, D)$ for $D \in \{1, 2, 3, 4\}$. Since each protocol is repeated $N_r = 10$ times, we report a moustache plot of the score Φ as well as the number of meta-clusters k, collected over the N_r repeats (for each value of D).

Results. For $D = 1$ (left panel, top left), our algorithm always outputs $k = 20$ meta-clusters. On closer inspection, this is due to the policy with regards to singletons. That is, our algorithm with $D = 1$ returns a matching (meta-clusters involving two clusters) plus singletons (meta-clusters with a single cluster; note that these do not affect the score $\Phi(\mathcal{S})$). After removing these singletons, we do get the correct number of meta-clusters (20, 15, and 10). For $D = 2$ (left panel, top right), as expected, our algorithm recovers the correct number k of meta-clusters (20, 15, and 10) for each comparison scenario ($e = 0$, $e = 5$, and $e = 10$ fusions). The returned scores (right panel, top right) confirm that our algorithm matches the merged clusters in F' with their split counterparts in F at any jitter level.

Figure 6.2 Algorithm $STS(G, D)$ for clusterings with $(t = 1\ 000, r = 20)$. **(Left panel)** Best value for k as a function of the 9 scenarios. **(Right panel)** Scores $\Phi_D(\cdot)$ as a function of the 9 scenarios.



This is made clear by comparing scores for scenarios in which we perform fusion operations (E5 or E10) to the ones where we do not (E0). Across all these scenarios, at an equivalent jitter level, the scores are nearly identical. Moreover, the fact that for $D = 1$, the score (right panel, top left) decreases linearly with respect to the number of fusions bolsters this hypothesis.

For $D = 3$, Φ is comparable to previous case ($D = 2$). The situation for k is more contrasted. We still identify three plateaus corresponding to the three different fusion scenarios. The scenarios in which we perform no fusions (E0) (resp. 5, E5, and 10, E10) tend to have a k which oscillates around 12 (resp. 10 and 8). The jitter seems to have little effect. For $D = 4$, we notice a similar behavior as for $D = 3$ but with different k-values. These results prompted the strategy for the choice of D presented in Section 6.5 as we can see that increasing $D \geq 2$ yields near identical scores but merges the clusters.

Conclusion. These experiments illustrate that our algorithm behaves as expected:

- On the one hand, scores are remarkably stable with respect to unions, as unions are retrieved within meta-clusters.
- On the other hand, the scores continuously degrade as a function of the jitter level, which is also expected since in general the items which have been shuffled cannot be recovered.

6.6.3 On the separability of clusters and the role of D

Rationale. We assess the role of D to provide insights of the *correct* number of clusters in a dataset. In other words, we provide a quantitative assessment on the following observation [SG02]: "*In fact, the* `right` *number of clusters in a dataset often depends on the* `scale` *at which the dataset is inspected*".

More precisely, we use `k-means++` on three datasets, with two values: $k_1 = 20$ and $k_2 = 50$ (k_i refers to the k parameter of `k-means++`). Since both these values yield an over-segmentation of the datasets, we challenge our method to retrieve the segmented clusters, using in particular plateaus plots (Section 6.5).

Along the way, we compare the results to those yielded by the gap statistic [TWH01].

Datasets. We generate three datasets composed of five random samples each drawn from a $2D$ Gaussian distribution. The relative position of the Gaussians is determined by a distance parameter d controlling the separability of the four clusters associated with the five random samples (Figure 6.3). Practically, we use $d = 5, 20, 50$. While these are 2D datasets for the sake of exposure, our machinery naturally applies in high-dimensional spaces where inferring the structure of a clustering is much more challenging.

Results. Three scenarios can be identified:

- **The dataset is not separable beyond the connected components of the intersection graph.** This is an easy case since the score Φ_D reaches the maximum possible value when meta-clusters correspond to connected components of the intersection graph (Figure 6.4 (A); $D = 8$ and $k = 4$ meta-clusters).
- **Dataset is separable: the plateaus strategy yields an unambiguous choice for D.** The best choice for D stems from the analysis of plateaus increments (Def. 6.7; Figure 6.6 (B), $D = 8$ yields $k = 3$ meta-clusters).
- **The dataset is not separable: no clear choice for D.** In this case, each successive plateaus has a significant increment and there is no clear break (Figure 6.4(C)).

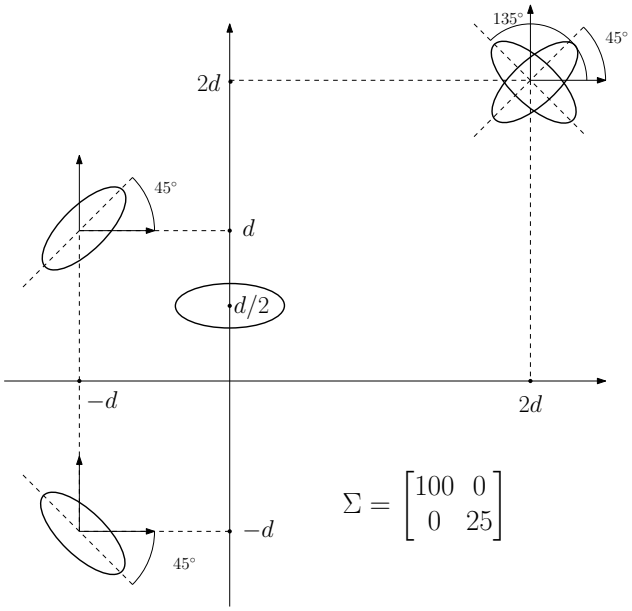
Gap statistic. The gap statistic performs comparably to our algorithm (Figure 6.5). However, it requires a reference distribution obtained via randomization, so that the number of clusters returned may be subject to variation.

Conclusion. Summarizing, the plateaus based analysis of scores provides insights on the plausible number of clusters.

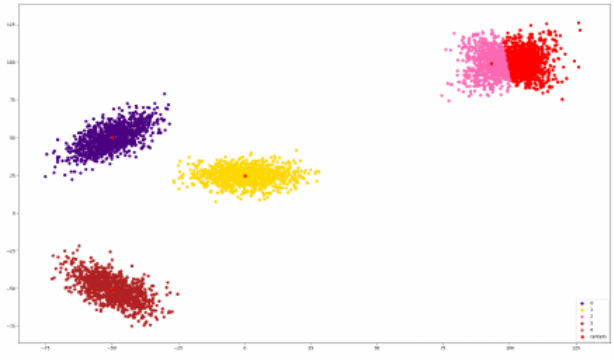
Remark 6.2. On the sensitivity to outliers.

When clustering with `k-means++` *the assignment of outliers to the different clusters is inherently unstable. When comparing two such clusterings, this creates edges with small weights in the intersection graph. For large values of D, these edges trigger the coalescence of meta-clusters. As a heuristic, such edges may be pruned from the intersection graph – a strategy not used in our experiments.*

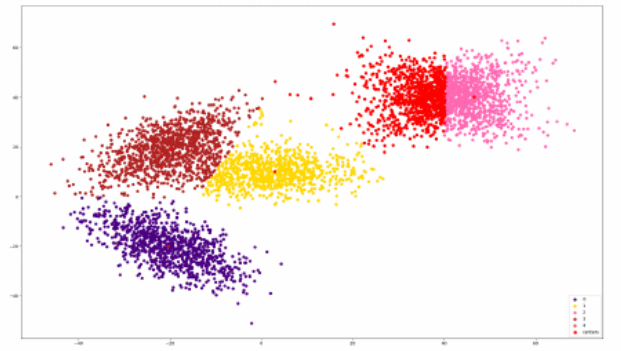
Figure 6.3 Parameterized dataset defined from a mixture of five Gaussians. **(A)** The distance parameter d controls the relative position of the five Gaussian blobs. The covariance matrix of the Gaussians is provided in the figure. **(B, C, D)** Random samples of $t = 5,000$ points for $d = 50, 20, 5$ respectively. Four regions/clusters are well separated for large values of d. Each point random sample was clustered using `k-means++` ($k = 5$).



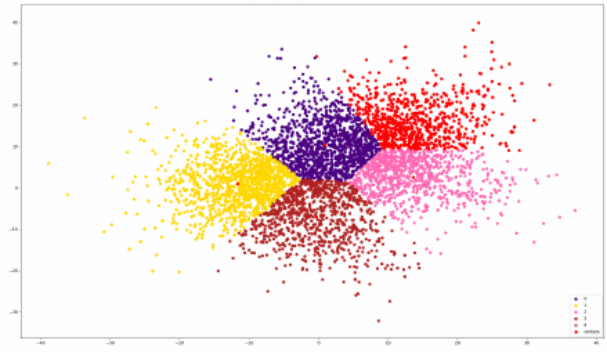
(A)



(B)

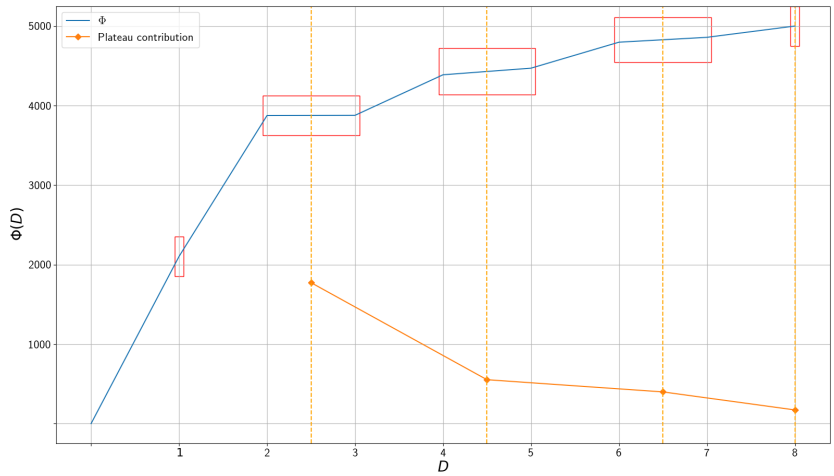


(C)

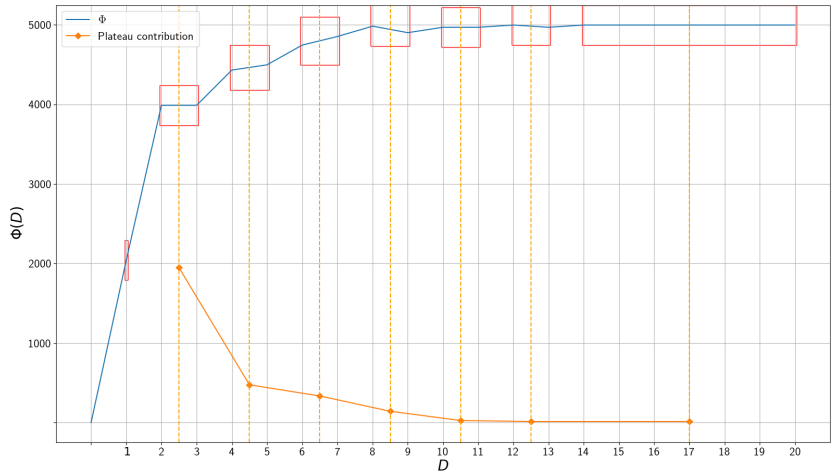


(D)

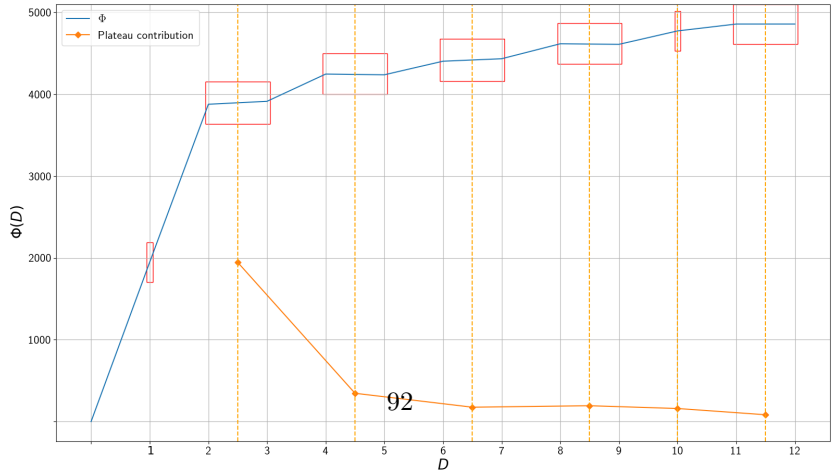
Figure 6.4 The plateaus plots for the three data sets–see text for details. **(A)** $d = 50$, $k = 4$ meta-clusters suggested for $D = 8$. **(B)** $d = 20$, $k = 3$ meta-clusters suggested for $D = 8$. **(C)** $d = 5$ No obvious choice for the number of meta-clusters.



(A)

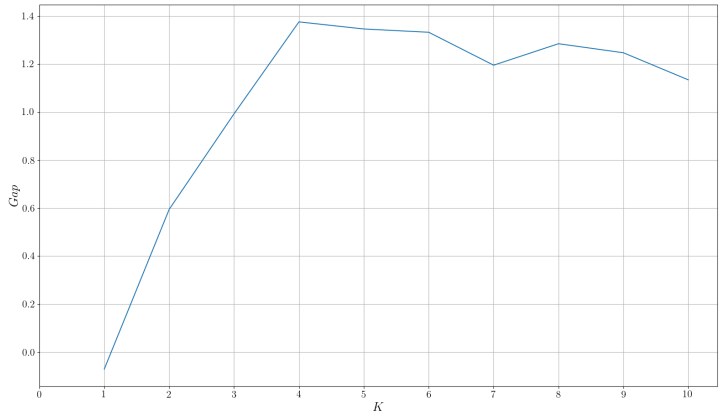


(B)

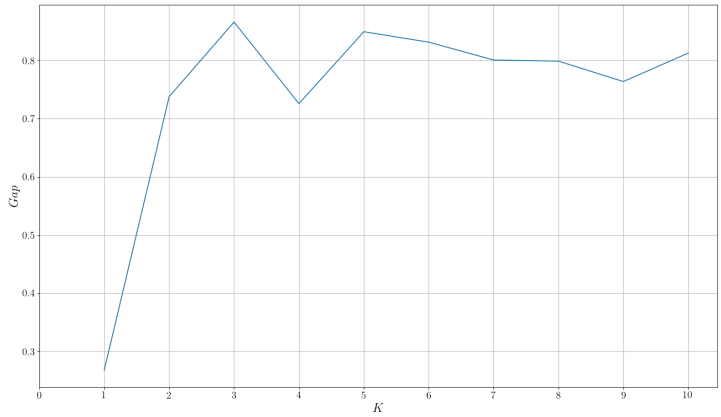


(C)

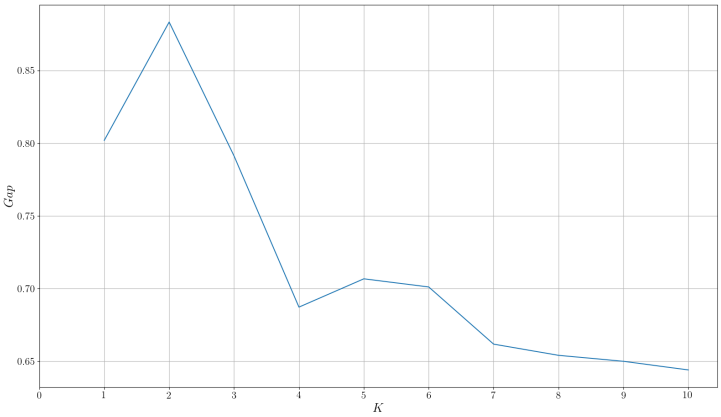
Figure 6.5 The gap statistic from [TWH01] for the three data sets. (A) d=50; the maximum value of the gap statistic hints at 4 clusters. **(B)** d=20; 3 clusters suggested, with a comparable value for 5 clusters. **(C)** d=5; 2 clusters suggested.



(A)

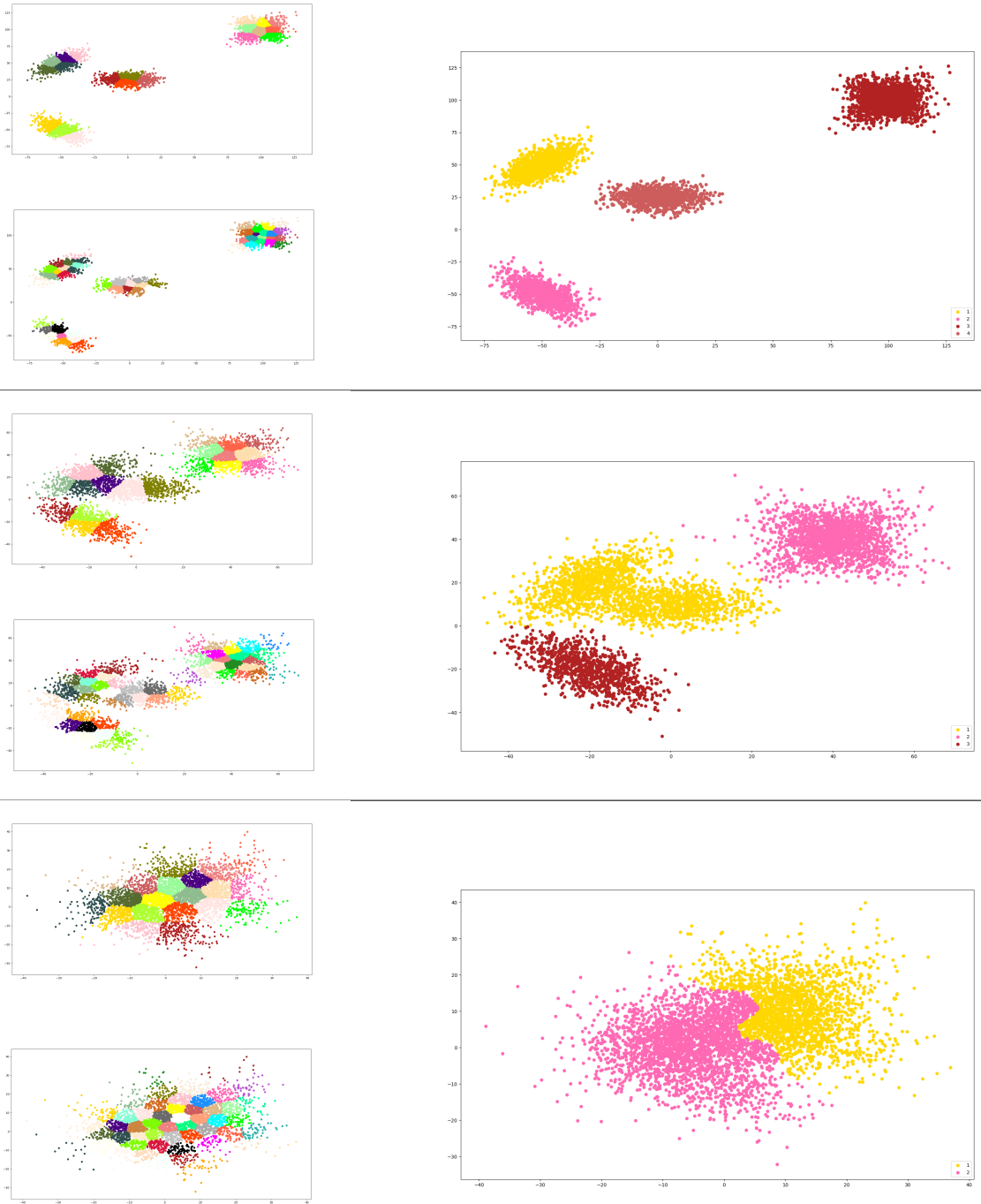


(B)



(C)

Figure 6.6 The meta-clusters for the three data sets. Left column: the two clusterings compared; right column: meta-clusters **(Top)** $d = 50$: two of the five Gaussians have merged and are separated from the other three– a data set which is not separable beyond the connected components of the intersection graph. The plateau plot suggests $D = 8$ and 4 meta clusters (Section 6.6.3). **(Middle)** $d = 20$: the 5 Gaussians define a dataset that may be separated into four connected components. The plateau plot suggests $D = 8$ and 3 meta clusters (Section 6.6.3). **(Bottom)** $d = 5$: the data set is not separable. The plateau plot does not suggest any specific number of meta clusters(Section 6.6.3).



6.6.4 Comparison to the Variation of Information (VI)

Rationale. As noticed in the conclusion of Section 6.6.2, edited clustering can be used to study the sensitivity of cluster analysis methods to merges and shuffles. We therefore compare the Variation of Information [Mei02] against our method in this respect. More specifically, we compare the normalized variation of information VI defined as $s_{VI} = \mathrm{VI}/\log t$ against our normalized score $s_\Phi = 1 - \Phi_D(\cdot)/t$. Recall that t is the number of points.

Dataset. We use the dataset from Section 6.6.2.

Results. Focusing on s_{VI} and s_Φ, our analysis relies on scatter plots of values (Figure 6.7) and standard deviations at fixed jitter levels (Figure 6.8) and fixed number of unions (Figure 6.9). We use different symbols depending on the considered scenario; the copy number of a symbol represents the number of repeats.

For $D = 1$, s_Φ is smaller than VI only in scenarios with no union operations. This is expected as our algorithm returns a perfect matching, so that unions are detrimental to the score.

For $D > 1$, we note several key differences with VI:

- s_Φ is always smaller than s_{VI} (Figure 6.7);
- s_Φ is more robust since $\sigma(s_\Phi) < \sigma(s_{VI})$ (Figs. 6.8 and 6.9).
- s_Φ is remarkably stable against merges, as evidenced by two facts. First, the standard deviation at any fixed jitter level is always very close to 0 (Figure 6.8). This stability is not observed for s_{VI}. Second, the jitter level has the same effect irrespective of the union scenario (Figure 6.9).

Conclusion. The variation of information, which is a global measure, is sensitive to cluster edits (merges, splits). On the opposite, the ability of our method to identify merges and splits makes it more suitable when insights on correspondences between clusterings are sought.

Figure 6.7 Normalized score s_{VI} versus normalized score s_Φ of algorithm $STS(G, D)$. See text for definitions. Each marker is a different union scenario and each color represents a different jitter scenario following the legend on the upper right. We plot the $y = x$ function for reference.

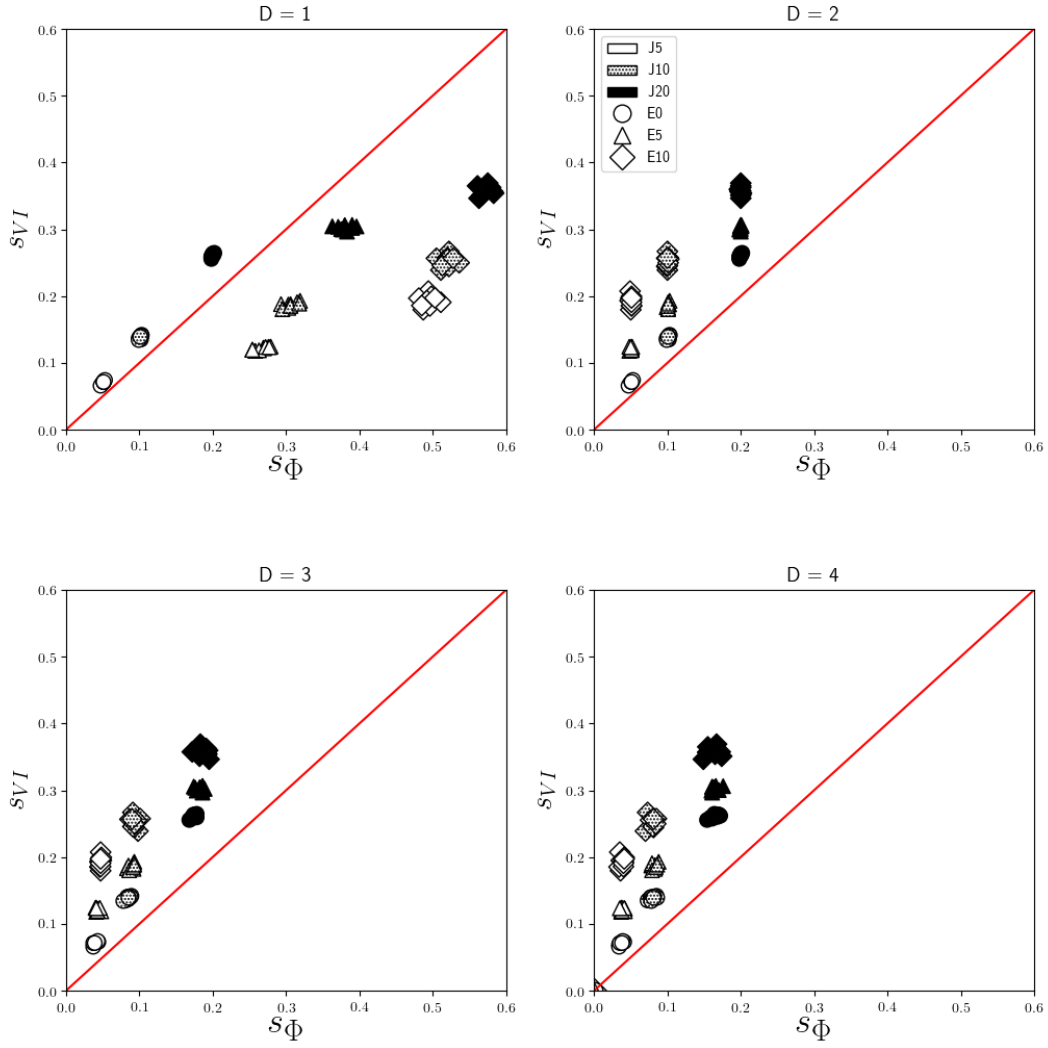


Figure 6.8 **σ of normalized score s_{VI} versus σ of normalized score s_Φ of algorithm $STS(G, D)$ with respect to jitter levels (i.e. experiments corresponding to all edits aggregated).** See text for definitions. Each color represents a different jitter scenario following the legend on the upper right. We plot the $y = x$ function for reference.

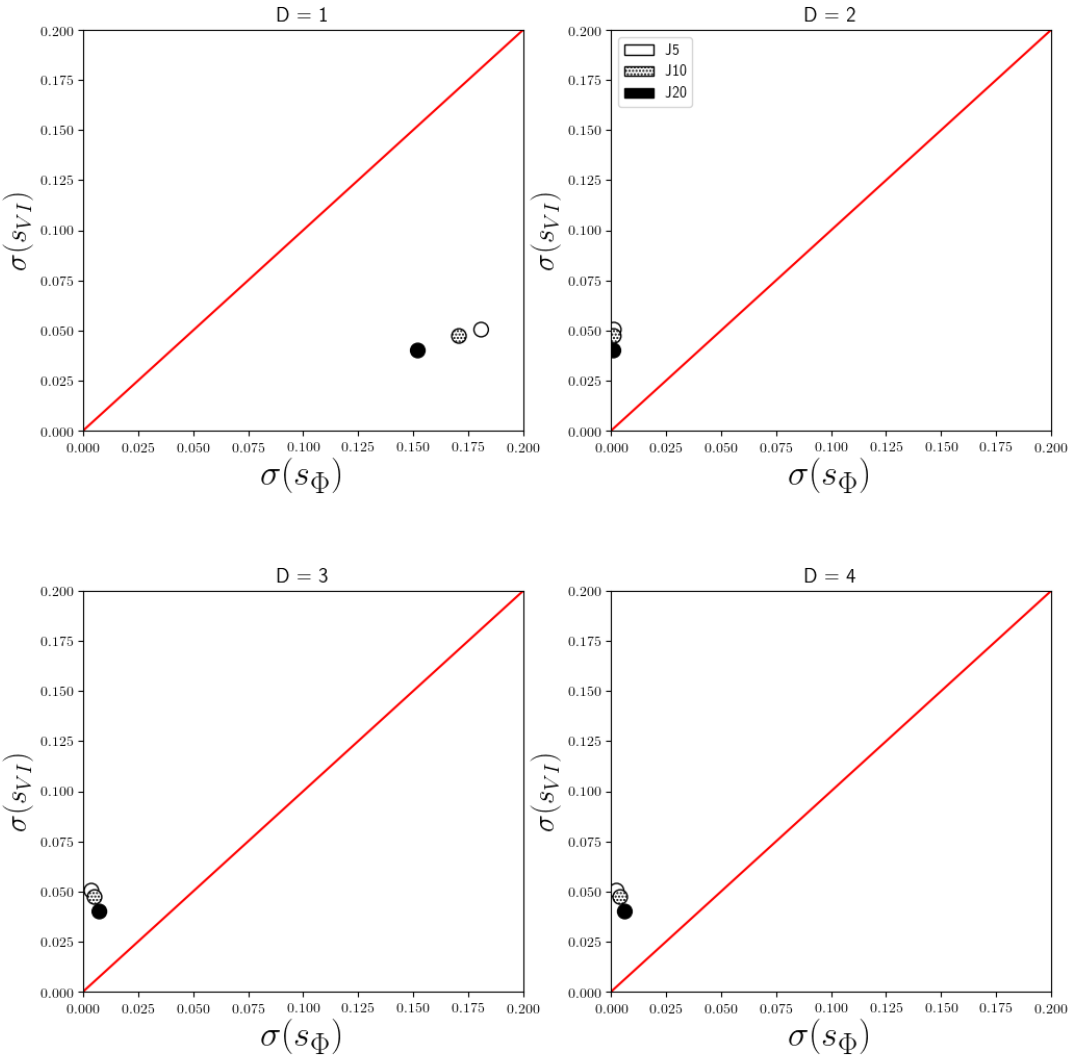
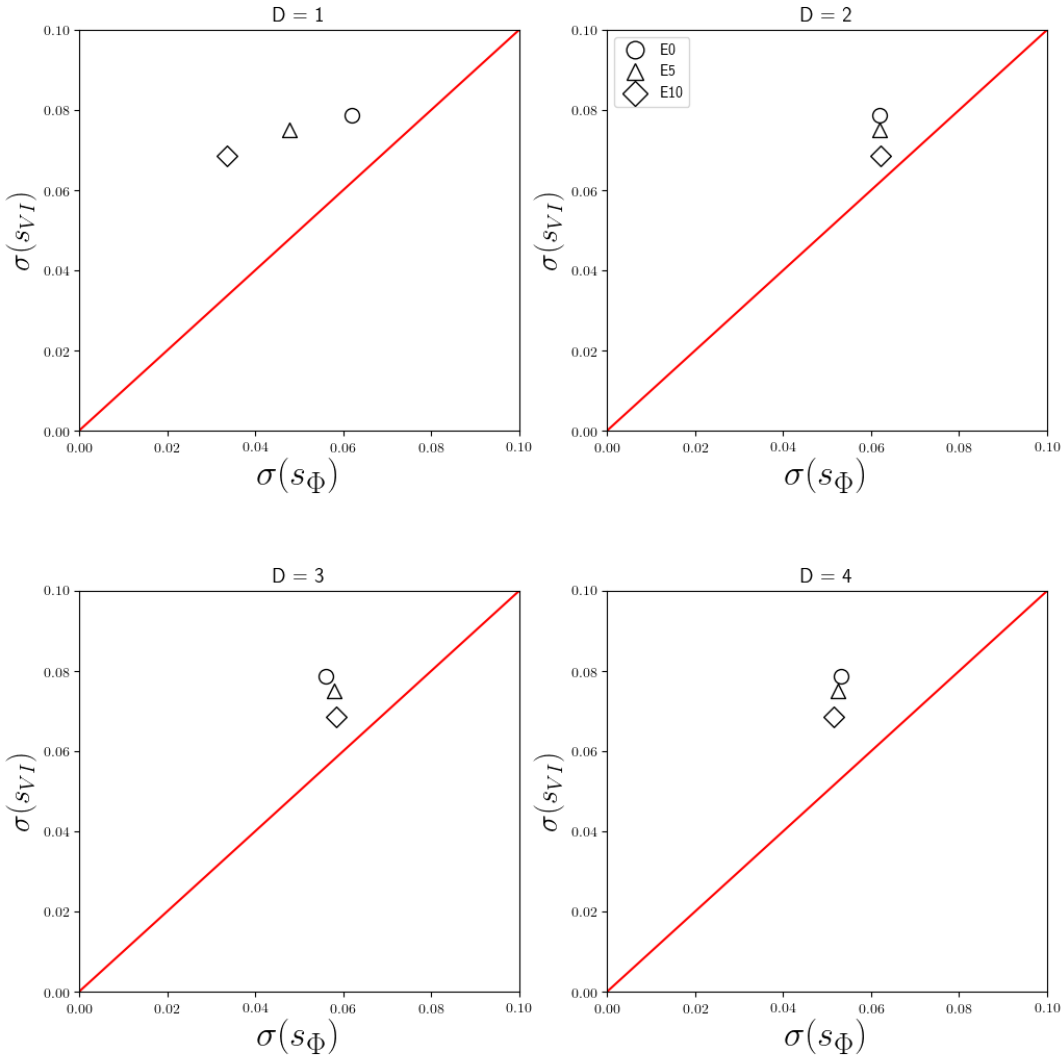


Figure 6.9 **σ of normalized score s_{VI} versus σ of normalized score s_Φ of algorithm $STS(G, D)$ with respect to number of edits (i.e. experiments corresponding to all jitters aggregated).** See text for definitions. Each marker represents a different union scenario following the legend on the upper right. We plot the $y = x$ function for reference.



6.7 Conclusion

This chapter introduces a new tier of algorithms to compare two clusterings, based on the identification of groups of clusters matching one-another. These problems are proved to be hard for general bipartite graphs (even if the maximum degree is at most three), with however polynomial time dynamic programming algorithms for specific graphs (in particular trees). These algorithms can in turn be used to design efficient algorithms, based on spanning trees, for general graphs. From a practical standpoint, experiments illustrate several key features of our algorithms. First, the meta-clusters obtained are highly effective to identify splits and merges between clusters. This ability yields a marked improvement when comparing two clusterings, with respect to global methods such as the variation of information, which are very sensitive to such edits. Second, in a manner analogous to the elbow method or variants, the stability of scores associated to meta-clusters, as a function of the diameter parameter D, provides a novel method to suggest the *correct* number of clusters in a clustering. Overall, we anticipate that our algorithms will prove instrumental to identify stable meta-clusters amidst clusterings (from different algorithms, or from the same algorithm with different parameters).

In terms of future work, we foresee two problems of particular importance, which were barely touched upon in previous work. The first one deals with the complexity of the problems we tackle. In the spirit of Lemma 6.5 (proving the existence of at least one spanning tree T of G such that an optimal solution for the family-matching problem for G constrained by T gives an optimal solution for the family-matching problem for G) we conjecture that there exists at least one spanning tree T of G such that an optimal solution for the family-matching problem for T (that can be obtained in polynomial time) gives a constant factor approximation for the family-matching problem for G. Furthermore, we conjecture that the D-family-matching problem is not in APX (recall that we proved that it is APX-hard). Note that both conjectures can be true because the existence of the previous tree would not guarantee a polynomial algorithm for determining it.

The second one deals with the stability of meta-clusters. Understanding which assumptions are indeed required to guarantee that our approach yields stable meta-clusters, in particular in terms of separability of the input sample points, would indeed leverage clustering by removing the arbitrariness inherent to the various algorithms and options available.

Chapter 7

Software

7.1 Protein representation

▶ **Group: Core::CSB / Package: Protein_representation**
`https://sbl.inria.fr/doc/group__Protein__representation-package.html`

7.1.1 Pre-requisites

The `SBL` provides many applications which rely on different structures tied to a polypeptide chain:

- topological information i.e. the covalent bonds –

  ▶ **Group: Core::CSB / Package: Molecular_covalent_structure**
  `https://sbl.inria.fr/doc/group__Molecular__covalent__structure-package.html`

- geometric information i.e. the coordinates which may be Cartesian or internal –

  ▶ **Group: Core::CSB / Package: Molecular_coordinates**
  `https://sbl.inria.fr/doc/group__Molecular__coordinates-package.html`

- biophysical annotations: hierarchical information (atoms, amino-acids, whole chain), an selected annotations (SSE, domains, etc...) –

  ▶ **Group: Thid_party_libraries / Package: ESBTL**
  `http://esbtl.sourceforge.net/`

This package answers the need for a unified structure allowing high level access to all this information.

7.1.2 Main classes

▶ **Pack: Protein_representation / Class: Polypeptide_chain_representation** Gives access to a number of high level accessors and iterators to manipulate a polypeptide chain. Allows access to the three structures described in the previous sub-section in a unique class.

▶ **Pack: Protein_representation / Class: Protein_representation** Gives access to a number of specified polypeptide chains from a protein quaternary structure.

▶ **Pack: Protein_representation / Class: Protein_loader** Gives access to a file loader which loads proteins and instantiates the two previous classes from a PDB file.

7.2 Molecular distances

7.2.1 Pre-requisites

Following the contributions from [CT18a], we provide a Core package as well as an application computing the $\text{RMSD}_{\text{Comb.}}$ in a number of situations:

- Homologous polypeptide chains
- Conformations of a polypeptide chain
- Motifs

For the Core packages, please refer to:
`https://sbl.inria.fr/doc/group__Molecular__distances-package.html`
For the application, please refer to:
`https://sbl.inria.fr/doc/group__Molecular__distances__flexible-package.html`

7.2.2 Main classes

▶ **Pack: Molecular_distances / Class: SBL::CSB::RMSD_comb_for_motifs** We provide a new class for the Molecular_distances package. Given a set of *structural motifs*, this class builds the *motif graph* (defined in [CT18a]), and computes the $\text{RMSD}_{\text{Comb.}}$.

▶ **Pack: Molecular_distances / Class: SBL::Modules::RMSD_comb_for_motifs_module** We provide the module enabling the use of the previous class in a workflow.

▶ **Group: SBL::Applications / Package: Molecular_distances_flexible** We provide an application which, given a set of polypeptide chains as well as "subdomain" definitions (labeled residue ranges), computes the $\text{RMSD}_{\text{Comb.}}$. The specification of labels is provided from SBL::Models::MolecularSystemLabelTraits. Example specification files can be found in the documentation. The application provides three executables:

- sbl-flexible-rmsd-proteins.exe is used to compare chains from homologous proteins.
- sbl-flexible-rmsd-conformations.exe is used to compare conformations of an identical protein
- sbl-flexible-rmsd-motifs.exe is used to compute the $\text{RMSD}_{\text{Comb.}}$ of two chains with user specified *structural motifs*.

7.3 Structural motifs

7.3.1 Pre-requisites

Following the contributions from ADDREF, we provide a novel package in the `SBL`. Given two polypeptide chains, the goal of this package is to identify *structural motifs* using any of the four methods from ADDREF.
`https://sbl.inria.fr/doc/group__Structural__motifs-package.html`

7.3.2 Main classes

▶ **Group: SBL::Applications / Package: Structural_motifs**

SBL::Applications::Structural_motifs: The application provides three executables:

- sbl-structural-motifs-chains-apurva.exe Searches for structural motifs in two polypeptide chains by bootstrapping the `Apurva` algorithm. The user can specify which filtration should be used (space filling or conserved distances)
- sbl-structural-motifs-chains-kpax.exe Searches for structural motifs in two polypeptide chains by bootstrapping the `Kpax` algorithm. The user can specify which filtration should be used (space filling or conserved distances)
- sbl-structural-motifs-conformations.exe Searches for structural motifs in two conformations of the same chain. The user can specify which filtration should be used (space filling or conserved distances)

7.4 Iterative alignment

7.4.1 Pre-requisites

We provide a package providing a modular environment for the easy design of iterative aligners (ADDREF). Additionally, we offer one such aligner.

▶ **Group: Core::CSB / Package: Iterative_alignment**
`https://sbl.inria.fr/doc/group__Iterative__alignment-package.html`

7.4.2 Main classes

▶ **Pack: Iterative_alignment / Class: Iterative_aligner** Provides a templated class allowing the instantiation of iterative aligners. The aligner requires:

- A structure type – defines the type for the structures to be aligned
- A seeder – computes seeds to initiate the iterative alignment
- A score computer – computes the scores for the DP matrix

▶ **Pack: Iterative_alignment / Class: Seeder_basic** Provides a class for the basic seeder, for which the seeds are inputed.

▶ **Pack: Iterative_alignment / Class: Seeder_DP_score** Provides a templated class allowing the instantiation of iterative alignment seeders using a DP strategy. Requires:

- A structure type
- A score computer – computes the scores for the DP matrix

▶ **Pack: Iterative_alignment / Class: Score_computer_K** Provides a class to compute the K score for the Kpax algorithm

▶ **Pack: Iterative_alignment / Class: Iterative_aligner_instantations** Provides two instantiations of the Iterative_aligner class:

- Kpax: Seeder_DP_score with Score_computer_K

▶ **Pack: Alignment_engines / Class: Alignment_engine_structures_kpax** An alignment engine instantiated with the Kpax iterative aligner. For details, see
`https://sbl.inria.fr/doc/group__Alignment__engines-package.html`

7.4.3 Executables

Additionally, this package provides one executable:

- sbl-kpax.exe allows the user to align two structures using the Kpax algorithm.

7.5 FunChaT

7.5.1 Pre-requisites

Following the contributions from ADDREF, we provide several packages to build hybrid hidden Markov models from a set of input polypeptide chains. These packages use the `UniProtKB` and `NCBI Taxonomy` databases.

7.5.2 Main classes

▶ Group: SBL::Core / Package: DB_manipulator
`https://sbl.inria.fr/doc/group__DB__manipulator-package.html`
Provides python classes which allow the concurrent manipulation of the `UniProtKB` and `NCBI Taxonomy` databases. This is done in order to get robust taxonomic information from `UniProtKB` protein entries.

- DB_manipulators.py provides wrappers to access Sqlite versions of the `NCBI Taxonomy` and `UniProtKB` databases. A script to construct these Sqlite DB from the online dumps is provided in the `SBL`.
- DB_hit_manager.py provides a class to handle hits from `UniProtKB`: extract taxonomic information, annotate sequences, etc...

▶ Group: SBL::Core / Package: HMMER_wrapper
`https://sbl.inria.fr/doc/group__HMMER__Wrapper-package.html`
Provides a python wrapper to streamline the construction and the query of hidden Markov model from python using the HMMER suite. Given an input set of protein sequences, HMMER_wrapper.py builds a multiple sequence alignment, using either Clustal Omega or Muscle. From this MSA, it then builds a HMM which can be used to query a given database.

▶ Group: SBL::Core / Package: Protein_sequence_annotator
`https://sbl.inria.fr/doc/group__Protein__sequence__annotator-package.html`
Provides python classes to handle sequence annotations and filters exploiting these annotations. Some annotations can be inferred from third party software, such as Phobius which identifies transmembrane regions.

- Sequence_annotators.py provides the Annotated_sequence class, as well as a pre-implemented Phobius annotator.
- Sequence_filters.py provides a two pre-implemented sequence filters: (i) a transmembrane filter, (ii) a filter to find fusion class II candidates.

▶ Group: SBL::Applications / Package: FunChaT
`https://sbl.inria.fr/doc/group__FunChaT-package.html`
FunChaT (Functional Characterization Tool) is a python application which uses structural motifs computed with the SBL::Applications::Structural_motifs package, and all the previously described classes to: (i) Build a HMM, (ii) Query a database, (iii) Filter results Provided are two executables:

- sbl-FunChaT-step-one.py uses the SBL::Applications::Batch_manager package to run a series of specified comparisons using the executables provided in SBL::Applications::Structural_motifs. The user specifies an input data set containing say n proteins, the executable runs the $\binom{n}{2}$ comparisons.

- sbl-FunChaT-step-two-and-three.py considers that the user has run structural comparisons on a input protein set and uses the results to build an HMM and query a specified database. By default, results are filtered according to fusion class II protein criterions.

Chapter 8

Conclusion

8.1 Conclusions

In this thesis, the focus was set on designing new tools for protein functional characterization. To this effect, the class II fusion proteins were the perfect data set for two reasons: (i) they share a loose structural homology coupled with little sequence identity which impedes the discovery of remote homologs with the current tools, (ii) recent work has shed light on several class II fusion proteins in Eukaryota (HAP2, EFF1) so that we know that remote homologs do exist.

The specificities of the scrutinized proteins naturally led us to work on local structural similarities. Surprisingly, to the best of our knowledge, classical structural distance measures such as the lRMSD had yet to be extended to deal with finer grain comparisons. In that respect, we developed the combined RMSD which circumvents the weaknesses of the lRMSD by mixing independent lRMSD measures each computed with their own rigid motion. This novel distance measure can be computed at several scales to compare secondary or tertiary structures based on structural motifs.

Having laid ground work in local structural comparison, the next step was identifying local similarities or *structural motifs*. We developed a bootstrap framework which, from a seed alignment, exploits filtrations and their associated persistence diagrams to uncover such regions. Through the detection of structural motifs, our work can be extended in two directions. On the one hand, for the comparison of protein structures, it allows finer grain comparisons and perfectly interfaces with the combined RMSD . On the second hand, structural motifs can be indicators on what characterizes a protein family.

Finding structural motifs implies identifying which local regions are comparable and can be generalized to the comparison of two clusterings of the same data set. This led to the formalization of the D-family-matching problem. Exploiting a graph structure modeling clusterings of the same data set, we developed an algorithm to form *meta-clusters*. This is akin to *consensus clustering* and allows for interesting applications in data science.

Building upon our new methods, we then explored a way of leveraging structural motifs for functional characterization. This was done by designing a hybrid method which exploits previously computed structural motifs to create a biased hidden Markov Model. The rationale behind is directly derived from the structure - function paradigm as it assumes that structurally conserved regions also harbor interesting features on a sequence standpoint.

These newly methods were used to perform a multivariate analysis of class II fusion proteins. Through structural analysis, this allowed to demonstrate the pertinence of comparing homologous protein at a local scale. The hybrid hidden Markov proved to be complimentary to the standard models by uncovering candidate class II sequences which were out of its reach.

8.2 Future work

This work lays the foundations for research topics in several directions.

8.2.1 Extending structural motifs

We developed a novel method to find structural motifs. However this method is restricted to pairwise comparisons. Extending this method to multiple comparisons is of interest. This could be done in two independent manners:

- if n proteins are the input, designing a novel method which processes the n proteins simultaneously to find common motifs. This is akin to multiple structural alignment methods such as `Kpax` [RGMV12].
- a different route would be processing the results of $\binom{n}{2}$ pairwise comparisons. The problem would then require finding the largest common sub-motif.

8.2.2 New iterative alignments

Another research direction is exploring new iterative alignments. This can be seen as the counterpart of potential energy landscape exploration algorithms, with the energy function being the alignment scores and the conformations being the rigid registration at each iteration. The stake is to explore this landscape as exhaustively as possible. Several solutions could be explored. From our previous method, using structural motifs as seeds is an interesting prospect. Our initial experiments yielded promising results, notably when comparing conformations of a single protein. Structural motifs were used to seed iterative alignments from different parts of a conformation allowing the detection of rigid areas. A similar method has yet to be tried for homologous proteins. Designing new score functions is also of interest. Some initial work was done but the results are not yet publishable. This is made easy by the templated design of the `SBL`.

8.2.3 Protein function characterization

With regards to protein function characterization, our work infused existing hidden Markov models with structural information. This idea, albeit a little naive, did yield results. Future work may be envisioned in two different areas: (i) finding a better way to exploit the structural similarities, by automatically locating patterns on the sequence, (ii) exploiting novel methods from deep learning in text recognition to design a new profile. Of particular interest are Sequence to sequence models as well as Long short term memory (LSTM) networks. To our knowledge, very few references exist for their application to protein function characterization.

8.2.4 Consensus clustering

The D-Family matching allowed us to explore a new research topic. Future work of the utmost interest would to design a way of exploiting meta-clusters to find the "correct" number of clusters in a "seperable" data-set. In the case of K-means this simply boils down to finding the K parameter. From several runs with different parameters, one could compare the obtained clusterings with our method and try to deduce the number of clusters from the meta-clusters. Doing this systematically would be very useful and is comparable to consensus clustering, in which from an input set of n clusterings, an algorithm tries to find the clustering which best characterizes the input set.

Bibliography

[AA13] R. Aldahdooh and W. Ashour. DSMK means *density-based split-and-merge k-means clustering algorithm. Journal of Artificial Intelligence and Soft Computing Research*, 3(1):51–71, 2013.

[AE96] N. Akkiraju and H. Edelsbrunner. Triangulating the surface of a molecule. *Discrete Appl. Math.*, 71:5–22, 1996.

[AHB87] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-square fitting of two 3D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[AMDY11] R. Andonov, N. Malod-Dognin, and N. Yanev. Maximum Contact Map Overlap Revisited. *J. of Computational Biology*, 18(1):1–15, January 2011.

[AMS+97] S. Altschul, T. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *NAR*, 25(17):3389–3402, 1997.

[Aur87] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM J. Comput.*, 16:78–96, 1987.

[AV07] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *ACM-SODA*, page 1035. Society for Industrial and Applied Mathematics, 2007.

[BBB+16] B. Bursteinas, R. Britto, B. Bely, A. Auchincloss, C. Rivoire, N. Redaschi, C. O'Donovan, and M. J. Martin. Minimizing proteome redundancy in the uniprot knowledgebase. *Database: The Journal of Biological Databases and Curation*, 2016.

[BC79] J. Baldwin and C. Chothia. Haemoglobin: the structural changes related to ligand binding and its allosteric mechanism. *JMB*, 129(2):175–220, 1979.

[BHK97] C. Barrett, R. Hughey, and K. Karplus. Scoring hidden markov models. *Computer applications in the biosciences: CABIOS*, 13(2):191–199, 1997.

[BHN03] H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide protein data bank. *Nature Publishing Group*, 10(980), 2003.

[BL93] J-P. Barthélemy and B. Leclerc. The median procedure for partitions. *Partitioning data sets*, 19:3–34, 1993.

[BOPR03] R. Blankenbecler, M. Ohlsson, C. Peterson, and M. Ringnér. Matching protein structures with fuzzy alignments. *PNAS*, 100(21):11936–11940, 2003.

[BRC+10] J-P. Baudry, A. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of computational and graphical statistics*, 19(2):332–353, 2010.

[BS01] M. Betancourt and J. Skolnick. Universal similarity measure for comparing protein structures. *Biopolymers*, 59(5):305–309, 2001.

[BSA+04] S. Bressanelli, K. Stiasny, S. Allison, E. Stura, S. Duquerroy, J. Lescar, F. Heinz, and F. Rey. Structure of a flavivirus envelope glycoprotein in its low-ph-induced membrane fusion conformation. *The EMBO journal*, 23(4):728–738, 2004.

[BT98] Carl Branden and John Tooze. *Introduction to protein structure*. Garland, 2 edition, 1998.

[BY98] J.-D. Boissonnat and M. Yvinec. *Algorithmic geometry*. Cambridge University Press, UK, 1998. Translated by H. Brönnimann.

[CBZ08] G. Csaba, F. Birzele, and R. Zimmer. Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, 24(16):i98–i104, 2008.

[CD17] F. Cazals and T. Dreyfus. The Structural Bioinformatics Library: modeling in biomolecular science and beyond. *Bioinformatics*, 7(33):1–8, 2017.

[CGOS13] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. *J. ACM*, 60(6):1–38, 2013.

[CGWL16] J. Chen, M. Guo, X. Wang, and B. Liu. A comprehensive review and comparison of different computational methods for protein remote homology detection. *Briefings in bioinformatics*, 19(2):231–244, 2016.

[Che95] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE PAMI*, 17(8):790–799, 1995.

[CK98] P. Crescenzi and V. Kann. How to find the best approximation results–a follow-up to garey and johnson. *ACM SIGACT News*, 29(4):90–97, 1998.

[CKL11] F. Cazals, H. Kanhere, and S. Loriot. Computing the volume of union of balls: a certified algorithm. *ACM Transactions on Mathematical Software*, 38(1):1–20, 2011.

[CLRSon] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009 (3rd edition).

[CMTW17] F. Cazals, D. Mazauric, R. Tetley, and R. Watrigant. Comparing two clusterings using matchings between clusters of clusters. 2017. Under revision.

[CMWY09] F. Chataigner, G. Manic, Y. Wakabayashi, and R. Yuster. Approximation algorithms and hardness results for the clique packing problem. *Disc. Appl. Math.*, 157(7):1396–1406, 2009.

[CSEH07] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[CT06] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley & Sons, 2006.

[CT18a] F. Cazals and R. Tetley. Characterizing molecular flexibility by combining lRMSD measures. 2018. Under revision.

[CT18b] F. Cazals and R. Tetley. Multiscale analysis of structurally conserved motifs. 2018. Submitted.

[DCJ11] S. Dey, P. Chakrabarti, and J. Janin. A survey of hemoglobin quaternary structures. *Proteins: Structure, Function, and Bioinformatics*, 79(10):2861–2870, 2011.

[DDL13] K. Dabrowski, M. Demange, and V. V. Lozin. New results on maximum induced matchings in bipartite graphs and beyond. *Theoretical Computer Science*, 478:33 – 40, 2013.

[DFLS04] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4-5):577–625, 2004.

[DH73] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. Wiley, 1973.

[Don00] S. Dongen. Performance criteria for graph clustering and markov cluster experiments. 2000.

[Edd98] S.R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 1998.

[Edd08] S.R. Eddy. A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS Comput Biol*, 4(5):e1000069, 2008.

[Edd15] S. Eddy. HMMER user's guide. biological sequence analysis using profile hidden markov models. 2015.

[Ede92] H. Edelsbrunner. Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, Dept. Comput. Sci., Univ. Illinois, Urbana, IL, 1992.

[Ede95] H. Edelsbrunner. The union of balls and its dual shape. *Discrete Comput. Geom.*, 13:415–440, 1995.

[Edg04] Robert C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[EH10] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[EHHM06] W. Eaton, E. Henry, J. Hofrichter, and A. Mozzarelli. Is cooperative oxygen binding by hemoglobin really understood? *Rendiconti Lincei*, 17(1-2):147–162, 2006.

[FCE11] R. Finn, J. Clements, and S.R. Eddy. HMMER web server: interactive sequence similarity searching. *NAR*, page gkr367, 2011.

[Fed12] S. Federhen. The ncbi taxonomy database. *Nucleic acids research*, 40(D1):D136–D143, 2012.

[Fer99] A. Fersht. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*. Freeman, 1999.

[FFL+18] J. Fedry, J. Forcina, P. Legrand, G. Pehau-Arnaudet, A. Haouz, M. Johnson, F. Rey, and T. Krey. Evolutionary diversification of the HAP2 membrane insertion motifs to drive gamete fusion across eukaryotes. *PLoS Biology*, In press(NA):NA, 2018.

[FJ02] A. Fred and A.K. Jain. Data clustering using evidence accumulation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 276–280. IEEE, 2002.

[FLPA+17] J. Fédry, Y. Liu, G. Péhau-Arnaudet, J. Pei, W. Li, M.A. Tortorici, F. Traincard, A. Meola, G. Bricogne, N. Grishin, W.J. Snell, F.A. Rey, and T. Krey. The ancient gamete fusogen hap2 is a eukaryotic class ii fusion protein. *Cell*, 168(5):904–915, 2017.

[FS09] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge University press, 2009.

[FT87] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, July 1987.

[Gar12] V. Garcia. A generative cell specific 1 ortholog in drosophila melanogaster. Technical Report Master thesis, 2012.

[GCR17] P. Guardado-Calvo and F.A. Rey. The envelope proteins of the bunyavirales. volume 98 of *Advances in Virus Research*, pages 83 – 118. Academic Press, 2017.

[GH94] O. Goldschmidt and D. S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Mathematics of operations research*, 19, 1994.

[GIP99] D. Goldman, S. Istrail, and C. Papadimitriou. Algorithmic aspects of protein structure similarity. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 512–521. IEEE, 1999.

[GKP89] R. Graham, D. Knuth, and O. Patashnik. *Concrete mathematics: a foundation for computer science*. Addison-Wesley, 1989.

[GL98] M. Gerstein and M. Levitt. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7(2):445–456, 1998.

[GOT17] J.E. Goodman, J. O'Rourke, and C.D. Tóth, editors. *Handbook of Discrete and Computational Geometry, Third edition*. CRC Press, 2017.

[GR01] M. Gerstein and F.M. Richards. Protein geometry: volumes, areas, and distances. In M. G. Rossmann and E. Arnold, editors, *The international tables for crystallography (Vol F, Chap. 22)*, pages 531–539. Springer, 2001.

[GS94] A. Godzik and J. Skolnick. Flexible algorithm for direct multiple alignment of protein structures and sequences. *Bioinformatics*, 10(6):587, 1994.

[GT14] F. Guyon and P. Tufféry. Fast protein fragment similarity scoring using a Binet-Cauchy kernel. *Bioinformatics*, 30(6):784–791, 2014.

[Har15] S.C. Harrison. Viral membrane fusion. *Virology*, 479–480:498–507, 2015.

[HH09] H. Hasegawa and L. Holm. Advances and pitfalls of protein structural alignment. *Current opinion in structural biology*, 19(3):341–348, 2009.

[hmm] HMMER. http://hmmer.org.

[HS93] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *Journal of molecular biology*, 233(1):123–138, 1993.

[HS95] L. Holm and C. Sander. Dali: a network tool for protein structure comparison. *Trends in biochemical sciences*, 20(11):478–480, 1995.

[Jai10] A.K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[JPKB+98] Jong J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *JMB*, 284(4):1201–1210, 1998.

[Kab76] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

[Kan91] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Inf. Process. Lett.*, 37(1):27–35, January 1991.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[KBH98] K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

[KBM+94] A. Krogh, M. Brown, I.S. Mian, K. Sjölander, and D. Haussler. Hidden Markov Models in computational biology: Applications to protein modeling. *JMB*, 235(5):1501–1531, 1994.

[KCE99] K.Kedem, P. Chew, and R. Elber. Unit-vector rms (urms) as a tool to analyze molecular dynamics trajectories. *Proteins: Structure, Function, and Bioinformatics*, 37(4):554–564, 1999.

[Kie14] M. Kielian. Mechanisms of virus membrane fusion proteins. *Ann. Rev. Virol.*, 1:171–89, 2014.

[KKS04] L. Käll, A. Krogh, and E. Sonnhammer. A combined transmembrane topology and signal peptide prediction method. *Journal of molecular biology*, 338(5):1027–1036, 2004. phoebius.

[KKSH05] K. Karplus, R. Karchin, G. Shackelford, and R. Hughey. Calibrating e-values for hidden markov models using reverse-sequence null models. *Bioinformatics*, 21(22):4107–4115, 2005.

[KL04] R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. *PNAS*, 101(33):12201–12206, 2004.

[KP13] T. Kodinariya and P. Prashant. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[KR06] M. Kielian and F. Rey. Virus membrane-fusion proteins: more than one way to make a hairpin. *Nature Reviews Microbiology*, 4(1):67–76, 2006.

[KT06] Jon Kleinberg and Éva Tardos. *Algorithm design*. Pearson Education India, 2006.

[KVV04] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.

[LA99] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *ACM SIGKDD*, pages 16–22. ACM, 1999.

[LAI06] C. Leslin, A. Abyzov, and V. Ilyin. TOPOFIT-DB, a database of protein structural alignments based on the topofit method. *Nucleic acids research*, 35(suppl_1):D317–D321, 2006.

[LAT10] P. Liu, D. Agrafiotis, and D. Theobald. Fast determination of the optimal rotational matrix for macromolecular superpositions. *Journal of computational chemistry*, 31(7):1561–1563, 2010.

[Lev07] M. Levitt. Growth of novel protein structural data. *PNAS*, 104(9):3183–3188, 2007.

[LPB+16] C. L. Lawson, A. Patwardhan, M. L. Baker, C. Hryc, E. S. Garcia, B. P. Hudson, W. Chiu, I. Lagerstedt, S. Ludtke, G. Pintilie, R. Sala, J. D. Westbrook, H. M. Berman, G. J. Kleywegt, and W. Chiu. Emdatabank unified data resource for 3dem. *Nucleic Acids Research*, 44, 2016.

[LRO07] D. Lee, O. Redfern, and C. Orengo. Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, 8(12):995–1005, 2007.

[Lux10] U. Von Luxburg. *Clustering Stability*. Now Publishers Inc, 2010.

[LWT05] R. Laskowski, J. Watson, and J. Thornton. ProFunc: a server for predicting protein function from 3d structure. *Nucleic acids research*, 33(suppl 2):W89–W93, 2005.

[LZY05] B. Long, Z. Zhang, and P. Yu. Combining multiple clusterings by soft correspondence. In *IEEE Int'l Conf. on Data Mining*, pages 8–pp. IEEE, 2005.

[MB06] A-Y. Mitrophanov and M. Borodovsky. Statistical significance in biological sequence analysis. *Briefings in Bioinformatics*, 7(1):2–24, 2006.

[MBBS13] V. Mariani, M. Biasini, A. Barbato, and T. Schwede. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

[MC95] V. Maiorov and G. Crippen. Size-independent comparison of protein three-dimensional structures. *Proteins: Structure, Function, and Bioinformatics*, 22(3):273–283, 1995.

[MDAY10] N. Malod-Dognin, R. Andonov, and N. Yanev. Maximum clique in protein structure comparison. In P. Festa, editor, *9th International Symposium on Experimental Algorithms*, pages 106–117, Ischia Island, Italy, 2010. Springer Berlin / Heidelberg.

[MDL07] G. Mayr, F. Domingues, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Structural Biology*, 7(1):50, 2007.

[Mei02] M. Meila. Comparing clusterings. 2002.

[MSZW11] L. Meng, F. Sun, X. Zhang, and M.S. Waterman. Sequence alignment as hypothesis testing. *Journal of computational biology*, 18(5):677–691, 2011.

[NAKH14] L. Nedialkova, M. Amat, I. Kevrekidis, and G. Hummer. Diffusion maps, clustering and fuzzy markov modeling in peptide folding transitions. *The Journal of chemical physics*, 141(11):09B611_1, 2014.

[NI00] H. Nagamochi and T. Ibaraki. A fast algorithm for computing minimum 3-way and 4-way cuts. *Mathematical Programming*, 88(3):507–520, 2000.

[NPH+18] E. Neveu, P. Popov, A. Hoffmann, A. Migliosi, X. Besseron, G. Danoy, P. Bouvry, and S. Grudinin. RapidRMSD: Rapid determination of RMSDs corresponding to motions of flexible molecules. *Bioinformatics*, 2018.

[NW70] Saul B. Needleman and Christian D. Wunsch. 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.

[OCP01] Oliviero O. Carugo and S. Pongor. A normalized root-mean-spuare distance for comparing protein three-dimensional structures. *Protein science*, 10(7):1470–1473, 2001.

[OKV13] K. Olechnovič, E. Kulberkytė, and C. Venclovas. CAD-score: A new contact area difference-based function for evaluation of protein structural models. *Proteins: Structure, Function, and Bioinformatics*, 81(1):149–162, 2013.

[Pea98] W.R. Pearson. Empirical statistical estimates for sequence similarity searches. *Journal of molecular biology*, 276(1):71–84, 1998.

[Per73] M. Perutz. Stereochemistry of cooperative effects in haemoglobin1. In *From theoretical physics to biology*, pages 247–285. Karger Publishers, 1973.

[Pev15] J. Pevsner. *Bioinformatics and functional genomics*. John Wiley & Sons, 2015.

[PPL06] C. Pál, B. Papp, and M. Lercher. An integrated view of protein evolution. *Nature Reviews Genetics*, 7(5):337, 2006.

[PR08] Gregory A Petsko and Dagmar Ringe. *Protein structure and function*. Oxford University Press, 2008.

[PS10] B. Phipson and G.K. Smyth. Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

[PVKV+14] J. Pérez-Vargas, T. Krey, C. Valansi, Ori O. Avinoam, A. Haouz, M. Jamin, H. Raveh-Barak, B. Podbilewicz, and F. Rey. Structural basis of eukaryotic cell-cell fusion. *Cell*, 157(2):407–419, 2014.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425 – 440, 1991.

[RGMV12] D. Ritchie, A. Ghoorah, L. Mavridis, and V. Venkatraman. Fast protein structure alignment using Gaussian overlap scoring of backbone peptide fragment similarity. *Bioinformatics*, 28(24):3274–3281, 2012.

[RHM+95] F. Rey, F. Heinz, C. Mandl, C. Kunz, and S. Harrison. The envelope glycoprotein from tick-borne encephalitis virus at 2 å resolution. *Nature*, 375(6529):291, 1995.

[Ric77] F. M. Richards. Areas, volumes, packing and protein structure. *Ann. Rev. Biophys. Bioeng.*, 6:151–176, 1977.

[Rit16] David W Ritchie. Calculating and scoring high quality multiple flexible protein structure alignments. *Bioinformatics*, page btw300, 2016.

[RL14] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[RR11] Michael G Rossmann and Venigalla B Rao. *Viral molecular machines*, volume 726. Springer Science & Business Media, 2011.

[RTG00] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[SG02] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[Shi07] T. Shibuya. Efficient substructure RMSD query algorithms. *Journal of Computational Biology*, 14(9):1201–1207, 2007.

[Sim08] S. Simic. On a global upper bound for Jensen's inequality. *Journal of Mathematical Analysis and Applications*, 343(1):414–419, 2008.

[SLC91] F. Smith, E. Lattman, and C. Carter. The mutation β99 Asp-Tyr stabilizes Y-A new, composite quaternary state of human hemoglobin. *Proteins: Structure, Function, and Bioinformatics*, 10(2):81–91, 1991.

[Söd04] J. Söding. Protein homology detection by hmm–hmm comparison. *Bioinformatics*, 21(7):951–960, 2004.

[SSTP14] N. Shibayama, K. Sugiyama, J. Tame, and S-Y. Park. Capturing the hemoglobin allosteric transition in a single crystal form. *Journal of the American Chemical Society*, 136(13):5097–5105, 2014.

[Ste02] B. Steipe. A revised proof of the metric properties of optimally superimposed vector sets. *Acta Crystallographica Section A: Foundations of Crystallography*, 58(5):506–506, 2002.

[SV95] H. Saran and V. Vazirani. Finding k-cuts within twice the optimal. *SIAM J. Comp.*, 24, 1995.

[SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 – 197, 1981.

[SWD+11] F. Sievers, A. Wilm, D. Dineen, T.J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J.D. Thompson, and D.G. Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular Systems Biology*, 7(1), 2011.

[The17] TheÂ UniProtÂ Consortium. Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169, 2017.

[TJP05] A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1866–1881, 2005.

[TWH01] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[TXK+03] C. Tang, L. Xie, I. Koh, S. Posy, E. Alexov, and B. Honig. On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. *Journal of molecular biology*, 334(5):1043–1062, 2003.

[Ume91] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 1991.

[VL07] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[WDBS08] J.M. White, S.E. Delos, M. Brecher, and K. Schornberg. Structures and mechanisms of viral membrane fusion proteins: multiple variations on a common theme. *Critical reviews in biochemistry and molecular biology*, 43(3):189–219, 2008.

[WHG07] W. Weissenhorn, A. Hinz, and Y. Gaudin. Virus membrane fusion. *FEBS letters*, 581(11):2150–2155, 2007.

[WJ94] LUSHENG WANG and TAO JIANG. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994. PMID: 8790475.

[WLT05] J. Watson, R. Laskowski, and J. Thornton. Predicting protein function from sequence and structural data. *Current opinion in structural biology*, 15(3):275–284, 2005.

[WMDAK12] I. Wohlers, N. Malod-Dognin, R. Andonov, and G. Klau. CSA: comprehensive comparison of pairwise protein structure alignments. *Nucleic acids research*, 40(W1):W303–W309, 2012.

[Xia06] Z. Xiang. Advances in homology protein structure modeling. *Current Protein and Peptide Science*, 7(3):217–227, 2006.

[XW05] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[YG03] Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19(suppl_2):ii246–ii255, 2003.

[YH01] Y-K. Yu and T. Hwa. Statistical significance of probabilistic sequence alignment and related local hidden markov models. *Journal of Computational Biology*, 8(3):249–282, 2001.

[Zem03] A. Zemla. LGA: a method for finding 3D similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

[ZLZ05] D. Zhou, J. Li, and H. Zha. A new mallows distance based metric for comparing clusterings. In *ICML*, pages 1028–1035. ACM, 2005.

[ZNI99] L. Zhao, H. Nagamochi, and T. Ibaraki. *Approximating the Minimum k-way Cut in a Graph via Minimum 3-way Cuts*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[ZS05] Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

Appendix A

Combining lRMSD measures

A.1 Supporting information

A.1.1 Method

Proof.[Proof of Observation 2] The square root function is concave and the weights positive. A direct application of Jensen's inequality, which provides an upper bound for convex functions, yields a lower bound in our case.

Define $a = l_{\min} = \min_i \mathrm{lRMSD}(C_i^{(A)}, C_i^{(B)})$ and $=lmax = \max_i \mathrm{lRMSD}(C_i^{(A)}, C_i^{(B)})$. To prove the upper bound, consider the domain $I = [a, b]$ over which the convex function is considered. The following is proved is [Sim08] (Fig. A.1):

$$\sum_i w_i f(x_i) - f(\sum_i w_i x_i) \leq C_{ab} \text{ with } C_{ab} = f(a) + f(b) - 2f(\frac{a+b}{2}), \tag{A.1}$$

which, for a short-hand, we rewrite as – denoting μ the mean (a linear function) applied to the set $\{x_i\}$

$$\mu(f) - f(\mu) \leq f(a) + f(b) - 2f(\frac{a+b}{2}). \tag{A.2}$$

Consider the interval $I = [a, b]$ with $a = l_{\min} = \min_i \mathrm{lRMSD}(C_i^{(A)}, C_i^{(B)})$ and $b = l_{\max} = \max_i \mathrm{lRMSD}(C_i^{(A)}, C_i^{(B)})$. Using $f = -\sqrt{\cdot}$, we get

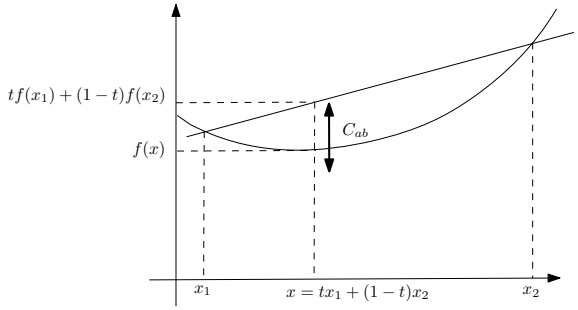
$$\mu(-f) - (-f(\mu)) \leq f(a) + f(b) - 2f(\frac{a+b}{2}) \tag{A.3}$$

$$\Leftrightarrow -\mu(f) + f(\mu) \leq f(a) + f(b) - 2f(\frac{a+b}{2}) \tag{A.4}$$

$$\Leftrightarrow f(\mu) \leq \mu(f) + 2\big(\sqrt{\frac{a+b}{2}} - \frac{\sqrt{a}+\sqrt{b}}{2}\big). \tag{A.5}$$

□

Figure A.1 The global upper bound for Jensen's inequality from [Sim08], used to prove an upper bound on the combined RMSD



A.1.2 Software

Implementation

(mode RMSD_MODE_SEQ) Structural motifs from amino acid ranges

Structures concerned. This mode is concerned with the comparison of homologous chains, or conformations of the same structure. Each of the combined lRMSD requires an alignment (Def. 3.1, which requires distinguishing two cases:

- *Conformations of the same molecule.* The alignment between the amino acids of the chain(s) is the identity alignment.
- *Homologous proteins.* For two homologous proteins, or two structures of the same protein obtained by different experiments, a structural alignment of their amino-acids is computed using algorithm `Apurva` [WMDAK12].

Defining motifs. Each motif is defined from the sequence, using a range of amino-acids. It is assumed (and enforced) that ranges do not overlap. The corresponding notion of combined RMSD is the combined RMSD with vertex (weighted) lRMSD .

Hierarchical comparisons: quaternary structure, chains, motifs. We provide two comparison modes:

- For proteins with quaternary structure i.e. decomposed into chains, we compare chains pairwise, and aggregate the lRMSD using the combined RMSD .
- For a chain decomposed into motifs, we compare motifs pairwise, and aggregate the lRMSD using the combined RMSD .

Executables. The following two executables are provided:

- `sbl-rmsd-flexible-proteins.exe`
- `sbl-rmsd-flexible-conformations.exe`

(mode RMSD_MODE_MOTIF) Structural motifs from local structural alignments

Structures concerned. This mode is concerned with the comparison of homologous chains, or conformations of the same structure.

Defining motifs. Motifs computed by structural algorithms require the explicit definition of an alignment (Def. 3.1). It does not matter in which configuration the user is (homologous chains or conformations), since the alignment is enforced by the motif definition, hence the single executable. In this mode, motifs are allowed to overlap. The corresponding notion of combined RMSD is the combined RMSD with edge (weighted) lRMSD .

Executables. Executable: `sbl-rmsd-flexible-motifs.exe`

A.1.3 Class II fusion proteins

Table A.1 Comparing class II viral fusion proteins, full structure: We display the lRMSD of the monomers in Fig. 3.3 after aligning them with the `Apurva` algorithm [AMDY11]. Numbers in Å.

	SFV-Alpha.	HRV-Hanta.	DFV-Flavi.	RVFV-Phlebo.	RBV-Rubi.	TBEV
HRV-Hanta.	13.813					
DFV-Flavi.	5.716	14.887				
RVFV-Phlebo.	5.321	15.011	6.426			
RBV-Rubi.	15.567	6.913	14.449	14.812		
TBEV	5.766	14.727	2.081	6.749	14.153	

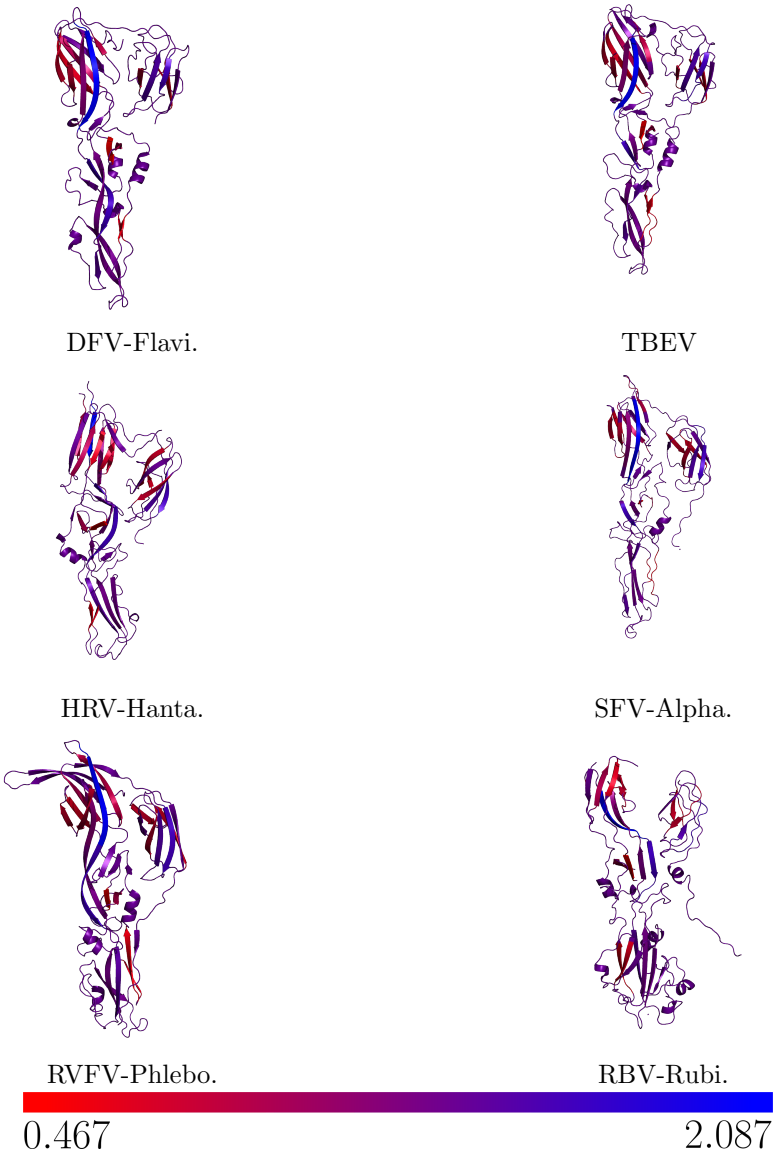
Table A.2 Comparing class II viral fusion proteins, Domain level: We display the $\text{RMSD}_{\text{Comb.}}$ of the monomers using their domain labels as defined in Fig. 3.3.

	SFV-Alpha.	HRV-Hanta.	DFV-Flavi.	RVFV-Phlebo.	RBV-Rubi.	TBEV
HRV-Hanta.	4.373					
DFV-Flavi.	3.987	8.932				
RVFV-Phlebo.	4.170	6.724	4.523			
RBV-Rubi.	10.254	8.012	6.263	9.298		
TBEV	4.059	7.827	1.586	7.150	5.807	

Table A.3 Comparing class II viral fusion proteins, SSE level: We display the $\text{RMSD}_{\text{Comb.}}$ of the monomers using their SSE labels as defined Fig. 3.3.

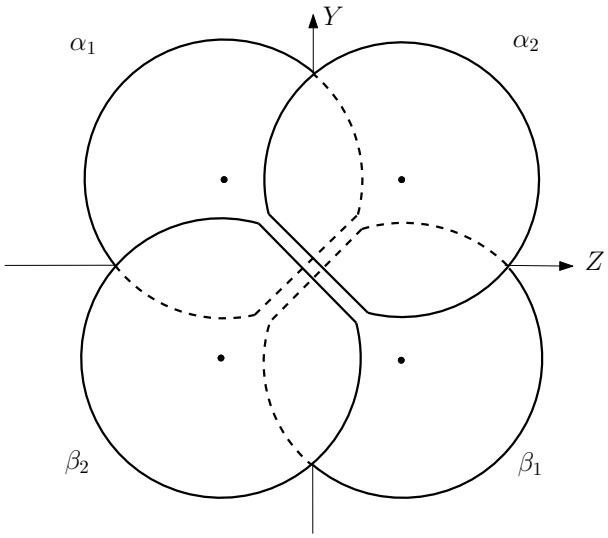
	SFV-Alpha.	HRV-Hanta.	DFV-Flavi.	RVFV-Phlebo.	RBV-Rubi.	TBEV
HRV-Hanta.	1.248					
DFV-Flavi.	1.256	1.446				
RVFV-Phlebo.	1.314	1.586	1.375			
RBV-Rubi.	1.083	1.423	1.593	1.410		
TBEV	1.578	1.123	1.083	1.558	1.675	

Figure A.2 Class II fusion, SSE conservation. For each SSE label defined in 3.3, we compute the $\binom{2}{6}$ pairwise comparisons and extract the median lRMSD value. We then display SSE conservation as a color map ranging from the minimum to the maximum lRMSD median.



A.1.4 Hemoglobin

Figure A.3 Hemoglobin: naming conventions for the α and β subunits relative to the reference axis, from [BC79]. Axis Y is the dyad axis relating $\alpha_1\beta_1$ to $\alpha_2\beta_2$. Note that the X axis is perpendicular to the figure. Upon oxygen binding, $\alpha_2\beta_2$ rotates of $\sim 15\,\text{deg}$ relative to $\alpha_1\beta_1$ around the X axis.



Naming conventions for subunits.

Data. The data used for reference states are as follows:

- T-state (PDB: 2DN2)
- R-state (PDB: 2DN3)
- R2-state (PDB: 1BBB)

The novel crystal structures reported in [SSTP14] are as follows:

- (HL+ , PDB: 4N7P): half-liganded with phosphate
- (HL- , PDB: 4N7O): half-liganded without phosphate
- (FL+ , PDB: 4N7N): fully water-liganded met-hemoglobin with phosphate

Assignment of quaternary states in [SSTP14]. The procedure used to evidence these conformations is as follows [SSTP14]: "*The assignments of quaternary states are based on difference distance matrices of the $\alpha_1\beta_2$ subunits using R as a reference (Figure 3a and Table 2). Also, a quaternary deviation from R can be measured by superimposing the C_α atoms of the $\alpha_1\beta_1$ dimer of each structure on those of R, and calculating the root-mean-square deviation (RMSD) in the C_α atoms for the non-superimposed $\alpha_2\beta_2$ dimers [SSTP14, Figure 3b].* "

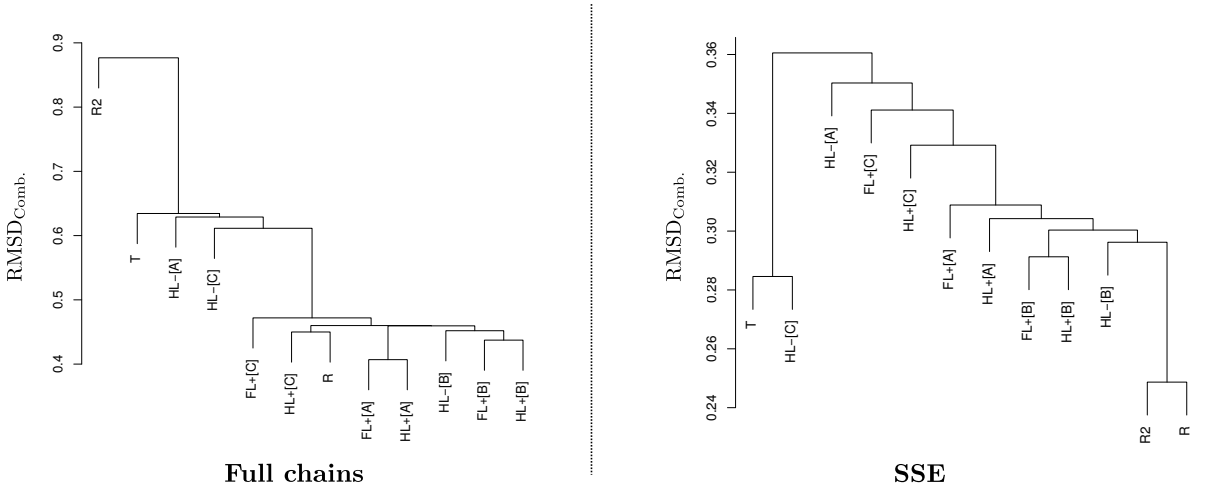
Table A.4 Comparing $\alpha_1\beta_1$ subunit of hemoglobin, SSE level: We display the $\text{RMSD}_{\text{Comb.}}$ of the monomers using their SSE labels as defined in Fig. ADDREF.

	R2	T	FL+[A]	FL+[B]	FL+[C]	HL-[A]	HL-[B]	HL-[C]	HL+[A]	HL+[B]	HL+[C]
T	0.412										
FL+[A]	0.278	0.418									
FL+[B]	0.289	0.403	0.292								
FL+[C]	0.378	0.481	0.383	0.382							
HL-[A]	0.339	0.450	0.343	0.353	0.453						
HL-[B]	0.333	0.401	0.340	0.304	0.410	0.382					
HL-[C]	0.363	0.295	0.361	0.349	0.436	0.397	0.346				
HL+[A]	0.275	0.392	0.273	0.303	0.391	0.303	0.348	0.353			
HL+[B]	0.315	0.386	0.326	0.267	0.384	0.371	0.273	0.337	0.317		
HL+[C]	0.378	0.388	0.380	0.366	0.399	0.450	0.366	0.344	0.385	0.346	
R	0.253	0.381	0.320	0.287	0.372	0.367	0.275	0.339	0.328	0.280	0.326

Table A.5 Comparing $\alpha_1\beta_2$ subunit of hemoglobin, SSE level: We display the $\text{RMSD}_{\text{Comb.}}$ of the monomers using their SSE labels as defined in Fig. ADDREF.

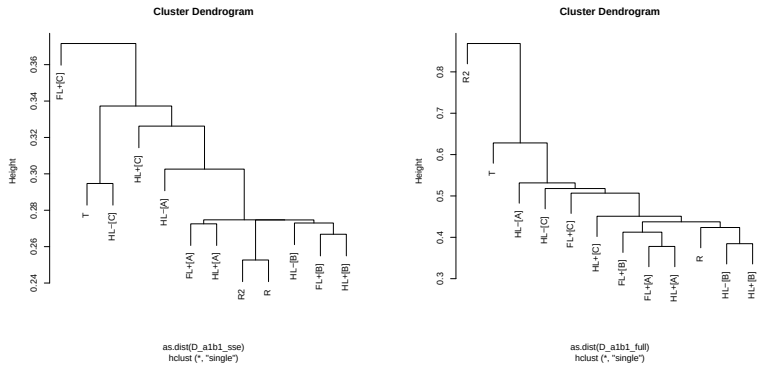
	R2	T	FL+[A]	FL+[B]	FL+[C]	HL-[A]	HL-[B]	HL-[C]	HL+[A]	HL+[B]	HL+[C]
T	0.382										
FL+[A]	0.309	0.435									
FL+[B]	0.324	0.410	0.348								
FL+[C]	0.342	0.420	0.372	0.343							
HL-[A]	0.357	0.402	0.386	0.386	0.404						
HL-[B]	0.342	0.396	0.383	0.334	0.367	0.410					
HL-[C]	0.371	0.285	0.398	0.378	0.407	0.398	0.361				
HL+[A]	0.304	0.406	0.324	0.352	0.385	0.350	0.399	0.383			
HL+[B]	0.335	0.394	0.374	0.291	0.353	0.392	0.302	0.373	0.367		
HL+[C]	0.388	0.412	0.410	0.381	0.367	0.452	0.360	0.390	0.416	0.366	
R	0.249	0.364	0.345	0.321	0.341	0.385	0.296	0.361	0.350	0.300	0.329

Figure A.4 Assigning quaternary structures of hemoglobin using $\alpha_1\beta_2$ dimers. Similarly to $\alpha_1\beta_1$, Fig. 3.5, combined RMSD for the $\alpha_1\beta_2$ dimer yields a satisfactory classification of quaternary structures.



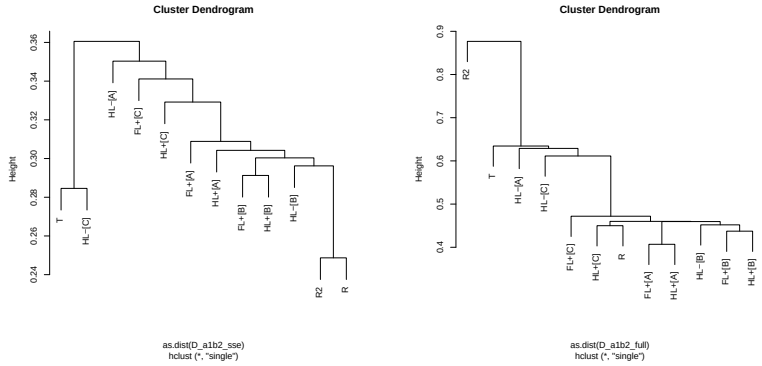
Full chains **SSE**

Figure A.5 Single linkage hierchical clusterings: four combinations of α and β subunits; $\mathbf{RMSD_{Comb.}}$ based on chains and SSE.



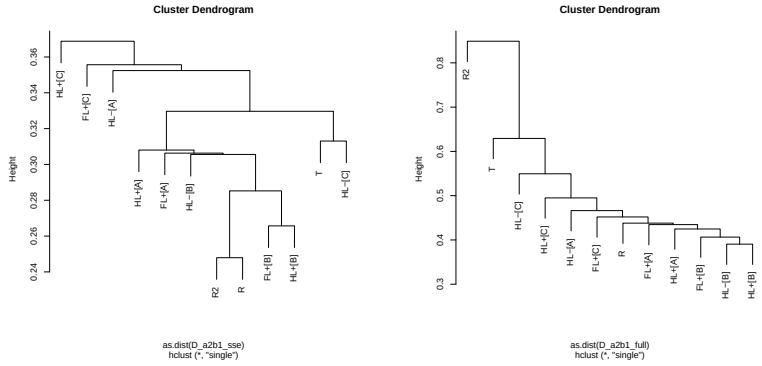
$\alpha_1\beta_1$ SSE

$\alpha_1\beta_1$ Full



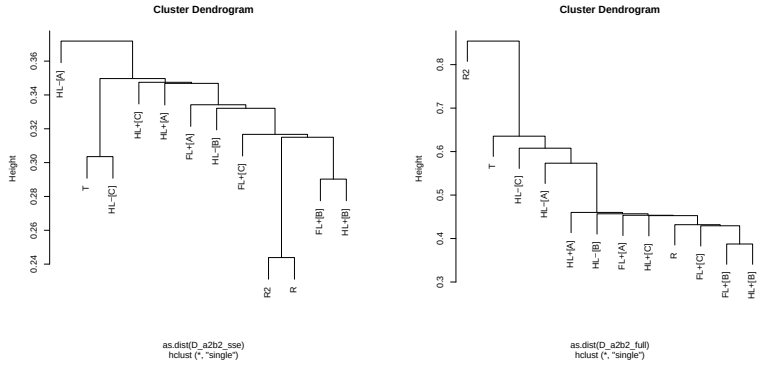
$\alpha_1\beta_2$ SSE

$\alpha_1\beta_2$ Full



$\alpha_2\beta_1$ SSE

$\alpha_2\beta_1$ Full



$\alpha_2\beta_2$ SSE

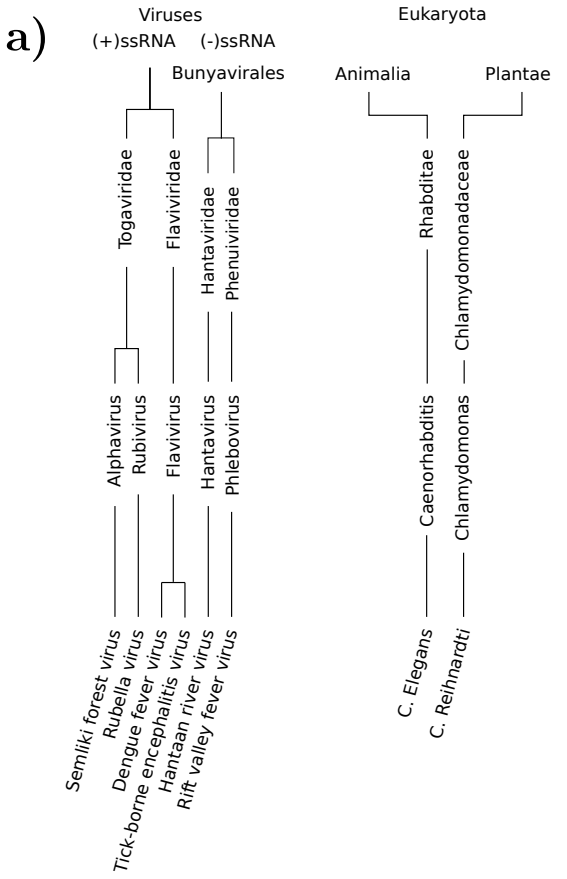
$\alpha_2\beta_2$ Full

Appendix B

Multiscale analysis of structurally conserved motifs

B.1 Supplemental: material

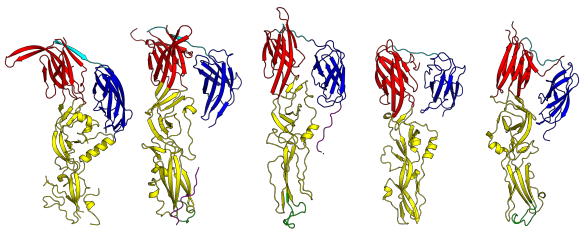
Figure B.1 Structures used in this study. **a)** Embedding of each structure in their respective taxonomic tree (one for viruses and one for Eukaryotes). We only detail the names for the genus and family ranks. The viruses are arranged in groups and the eukaryotes in kingdoms. **b)** Here we provide the files used in the study as well as the acronym used for each structure throughout this article. **c)** Visualization of selected structures generated with PyMol. We present one structure per genus in the taxonomic tree. Each structure is decomposed in three main domains: DI (red), DII (yellow), DIII (blue).



b)

Viruses			
Name	PDB code	Acronym	# a.a.
Semliki Forest virus	1RER	SFV	391
Dengue fever virus	1OK8	DFV	380
Tick-borne encephalitis virus	1URZ	TBEV	382
Hantaan river virus	5JLX	HRV	418
Rift valley fever virus	6EGT	RVFV	446
Rubella virus	4ADI	RBV	437
Eukaryota			
C.Elegans	4OJC	EFF-1	461
C. Reihnardtii	5MF1	HAP2-C	446

c)



B.2 Supplemental: mathematical and algorithmic background

B.2.1 Structural comparisons, motifs, and motif graphs

Structural motifs, motif graph and combined lRMSD measures. To combine several motifs and perform a whole structure/complex comparison, we resort to the so-called motif graph [CT18a]:

Definition. B.1. *(Motif graph) The* motif graph *of a list of motifs* $\{(M_i^{(A)}, M_i^{(B)})\}_{i=1,\dots,p}$ *is defined as follows: its node set is the union of the particles A and B; its edge set is the union of two types of edges:*

- matching edges*: the edges associated with the matchings defined by the individual motifs. (NB: such edges are counted without multiplicity, that is, a matching edge present in several motifs is counted once.)*

- motif edges*: edges defining a path connecting all a.a. in a motif.*

Consider a connected component (c.c.) of the motif graph. Restricting each c.c. to each structure yields two subgraphs. The set of all such subgraphs is denoted $\{C_i^{(A)}, C_i^{(B)}\}_{i=1,\dots,m}$.

Each such c.c. can then be used to combined a so-called combined lRMSD [CT18a]:

Definition. B.2. *Assume non-overlapping connected components* $\{(C_i^{(A)}, C_i^{(B)})\}_{i=1,\dots,m}$ *have been identified for A and B. Also assume that a lRMSD has been computed for each pair* $(C_i^{(A)}, C_i^{(B)})$*, and denote* w_i *the corresponding weight. The* combined RMSD *is defined by*

$$RMSD_{Comb.}(A,B) = \sqrt{\sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} lRMSD^2(C_i^{(A)}, C_i^{(B)})}. \tag{B.1}$$

Note that for a $\text{RMSD}_{\text{Comb.}}$ calculation, the number of a.a. is the number of nodes in the motif graph.

B.2.2 Sparse quasi-isometric deformations

The following provides insights on the rationale for using the conserved distances filtration.

Local rigidity. To characterize the rigidity of a shape R, a molecule in our case, we focus on the variation of the distance between two points p_i and p_j of the shape over time, that is

$$D_{p_ip_j}(t,t') = d_{p_jp_j}(t') - d_{p_jp_j}(t). \tag{B.2}$$

One typically uses a threshold ε to control this deformation score:

$$\mid D_{p_ip_j}(t,t') \mid \leq \varepsilon. \tag{B.3}$$

When shape R consists of N points, the condition of Eq. (B.3) may be used in two ways.

Global rigidity. As the strongest condition, one may require Eq. (B.3) to hold for all pairs of points. This global characterization yields maximum clique calculations on so-called alignment graphs, a hard problem [MDAY10].

Sparse quasi-isometric deformation. On the other hand, the weakest condition consists in requiring Eq. (B.3) to hold for specific edges connecting the points of interest. To characterize connected regions, this requires the existence of a spanning tree connecting the points in R, with Eq. (B.3) holding true for each edge of the tree. Since a spanning tree is the sparest structure connecting the points, we term a deformation complying with this property a *sparse quasi-isometric deformation*. When a molecule deforms, the changes undergone by bond lengths are usually smaller than those undergone by dihedral angles, so that one expects subtrees of the covalent structure to comply with conditions of quasi-isometric deformations. However, we shall see that C_α carbons non consecutive along the sequence are often *more rigid* than pairs consecutive along the sequence.

B.2.3 Weighted graphs, Space filling diagrams, and filtrations

Space filling diagrams. Molecular models defined by union of balls [Ric77] are called space filling diagrams (SFD) [Ede95]. In the so-called the solvent accessible model, atomic radii are taken as van der Waals radii expanded by the radius of a water probe [GR01], so as to capture interactions between atoms which are nearby in 3D space yet not covalently bonded. More formally, let the *restriction* of an atom be the intersection of the expanded ball and its Voronoi (power) region. Interactions are captured by neighboring restrictions. The SFD naturally defines the *contact graph* which codes contacts between a.a.: its vertices are

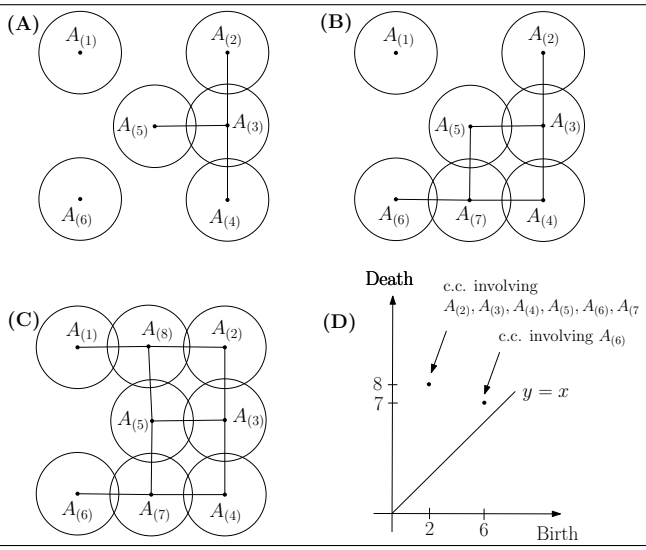
the individual a.a.; two a.a. are linked by one edge provided there exist two atoms, one for each a.a., whose restrictions are neighbors.

SFD are best manipulated using the α-complex, which is a subset of the Delaunay (regular) triangulation of the balls [AE96, CKL11]. In particular, the α-complex gives a direct access to the contact graph. Topological properties of the SFD, such as its connected components (and their number, the first Betti number β_0), and its cycles (and their number β_1) are also easily computed from the α-complex of the SFD [EH10].

Filtration defined from conserved distances. Consider a graph whose vertices represent a.a., and edges are weighted by a score conveying information on the conservation of the distance between the C_α of these a.a.. (The score will be the distance difference between the two a.a., measured on the two molecules.) Assume that edges have been sorted in ascending order (details in section 4.1.4). To each edge weight w, corresponds a graph containing only those edges whose weight is $\leq w$. In processing edges by increasing value, these graphs are nested. We call this sequence of graphs the *conserved distances (CD) filtration*.

Filtration defined from a space filling diagram. Consider now a SFD such that each individual a.a. has been assigned an index (details in section 4.1.4). To each index i, corresponds a SFD containing only those a.a. whose index is $\leq i$. In processing indices by increasing value, these SFD are nested. We call this sequence of SFD the *SFD filtration*.

Figure B.2 Step 2, Building the filtration and its persistence diagram: filtration from space filling diagram (SFD). A SFD involving eight amino-acids $A_1, \ldots, A_8$, each sketched with a ball, is incrementally constructed: at step $i \in 1, \ldots, 8$, the a.a. $A_{(i)}$ is added. The solid edges define the graph connecting the a.a. (**D**) The associated persistence diagram summarizes the birth and death of connected components. For example, the c.c. born with the insertion of A_2 (birth date = 2) dies at time 8, when it merges with the c.c. born with A_1, due to the connexion created by A_8. The c.c. born with A_1 never dies.



Order zero persistence diagram of a filtration. For either filtration, let us focus on connected components, which appear and disappear when inserting edges (CD filtration) or a.a. (SFD filtration). A union-find data structure can be used to maintain these c.c. [CLRSon], and track in particular the *birth date* and the *death date* of each component. The collection of all these points $\{(\text{birth date}, \text{death date})\}$ defines a 2D diagram called the *persistence diagram* [EH10]. Note that all points lie above the diagonal $y = x$; moreover, the distance of a point to the diagonal is the lifetime of the c.c. associated to that point.

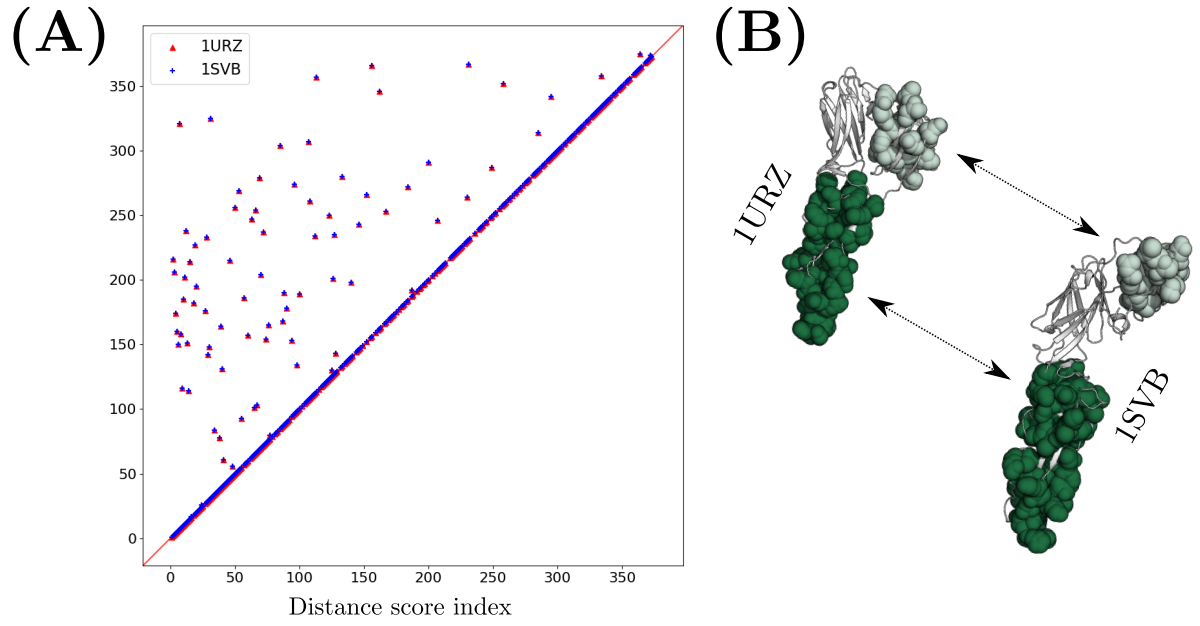
B.3 Supplemental: method

B.3.1 Step 2, illustration

For the sake of comparison / illustration, we provide results on two methods – see also Table B.1:

- Fig. 4.4: `Align-Identity-SFD`
- Fig. B.3: `Align-Identity-CD`

Figure B.3 Step 2, Building the filtration and its persistence diagram: illustration for 1URZ-1SVB with the `Align-Identity-CD`. See caption of Fig. 4.4. Executables listed in SI Table B.1. **Remark:** In the case of conserved distances filtration, the method yields as many as 71 structural motifs. For convenience, we do not report their statistics or the Hasse diagram in this figure.



B.3.2 Statistical significance of motifs

Denote p the number of motifs obtained for two structures. For a given structure, define a random motif of a given size as a random subgraph of the contact graph (Section B.2.3). For each structure, we generate p random motifs, say $\{r_1^A, \ldots, r_p^A\}$ and and $\{r_1^B, \ldots, r_p^B\}$. We then run a structural alignment between r_i^A and r_i^B, $i = 1, \ldots, p$, and compute the associated lRMSD . The i-th pair of random motifs is thus characterized by the pair (lRMSD , number of a.a. aligned), and by the ratio ratio lRMSD/#amino acids–the smaller the better.

To compare our motifs and of the random ones, we compare the two lists of ratios lRMSD/#amino acids. To do so, we use the Wilcoxon Mann-Whitney U test, which comes with a simple effect size, namely the Hodges-Lehmann estimate Δ for the population shift. Note that in comparing our motifs against the random ones, we expect a small p-value and a negative effect size – as the smaller the ratio lRMSD/#amino acids the better. When the null hypothesis is not rejected, all motifs for the pair of interest are discarded.

Remark B.1. *To comply with the independence assumption of samples, one can focus on motifs whose pairwise intersection (in terms of amino-acids) is void.*

Table B.2 Statistical significance of our motifs, when compared against random motifs with two non parametric two-sample tests. Second column: p-value for the Wilcoxon Mann-Whitney U test. Third column: Effect size. Note that low p-values indicate that there is no evidence to believe that our motifs and random motifs have identical parameter signatures.

Partners	Ref. lRMSD	p-value	Effect size
DFV-Flavi. vs EFF-1	6.61	0.000010	-0.224137
DFV-Flavi. vs HRV-Hanta.	14.91	0.043465	-0.252118
EFF-1 vs HRV-Hanta.	15.48	0.000002	-0.158204
HAP2-C vs DFV-Flavi.	8.05	0.000036	-0.236277
HAP2-C vs EFF-1	7.73	0.000758	-0.305677
HAP2-C vs HRV-Hanta.	17.08	0.000006	-0.197719
HAP2-C vs RVFV-Phlebo.	8.03	0.001649	-0.226539
RBV-Rubi. vs DFV-Flavi.	14.45	0.000024	-0.117758
RBV-Rubi. vs EFF-1	16.57	0.000009	-0.124759
RBV-Rubi. vs HRV-Hanta.	6.86	0.056718	-0.250126
RBV-Rubi. vs TBEV	14.28	0.000000	-0.280079
RVFV-Phlebo. vs DFV-Flavi.	6.42	0.000000	-0.321564
RVFV-Phlebo. vs EFF-1	5.15	0.054799	-0.352183
RVFV-Phlebo. vs HRV-Hanta.	15.03	0.117667	-0.136867
RVFV-Phlebo. vs TBEV	6.69	0.000000	-0.357342
SFV-Alpha. vs DFV-Flavi.	5.73	0.000280	-0.315946
SFV-Alpha. vs EFF-1	7.61	0.000111	-0.217348
SFV-Alpha. vs HAP2-C	7.35	0.062893	-0.192960
SFV-Alpha. vs RBV-Rubi.	15.97	0.001770	-0.147043
SFV-Alpha. vs RVFV-Phlebo.	5.32	0.000000	-0.295729
SFV-Alpha. vs TBEV	5.73	0.059286	-0.394517
TBEV vs DFV-Flavi.	2.08	0.000081	-0.430133
TBEV vs HRV-Hanta.	14.72	0.010568	-0.233234

B.3.3 Methods and programs

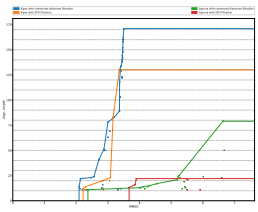
Table B.1 Method-executable correspondence. When qualified by the suffix *iter*, e.g. `Align-Kpax-CD`/iter, a method is used to seed an iterative aligner with our motifs (Sec. 4.2).

Method	`SBL` executable	Option
`Align-Apurva-SFD`	sbl-structural-motifs-chains-apurva.exe	
`Align-Apurva-CD`	sbl-structural-motifs-chains-apurva.exe	–use-cd-filtration
`Align-Kpax-SFD`	sbl-structural-motifs-chains-kpax.exe	
`Align-Kpax-CD`	sbl-structural-motifs-chains-kpax.exe	–use-cd-filtration
`Align-Identity-SFD`	sbl-structural-motifs-conformations.exe	
`Align-Identity-CD`	sbl-structural-motifs-conformations.exe	–use-cd-filtration

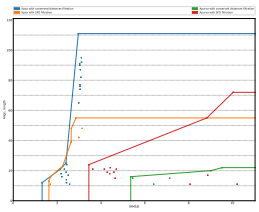
B.4 Supplemental: results

B.4.1 Statistical significance of motifs

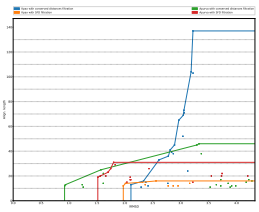
We illustrate the statistical significance of the motifs yielded by `Align-Apurva-SFD`. The p-values for the Wilcoxon Mann-Withney U-test for each comparison can be found in Table B.2. The effect sizes are always negative. We report high p-values (≥ 0.05) in 5 cases (highlighted in gray in Table B.2). In most instances p-values are ≤ 0.001.



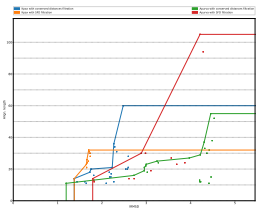
DFV-Flavi. vs HAP2-C



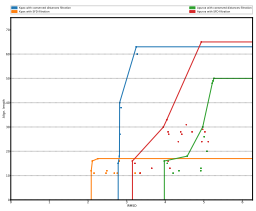
DFV-Flavi. vs RBV-Rubi.



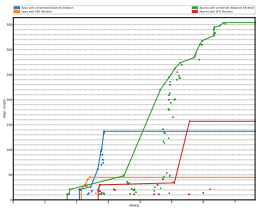
DFV-Flavi. vs RVFV-Phlebo.



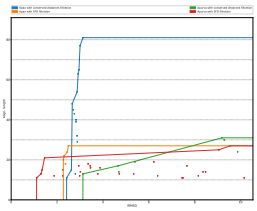
DFV-Flavi. vs SFV-Alpha



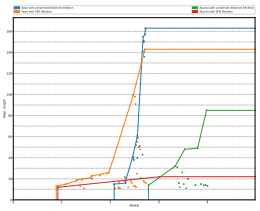
EFF-1 vs DFV-Flavi.



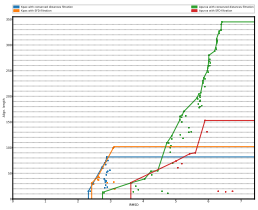
EFF-1 vs HAP2-C



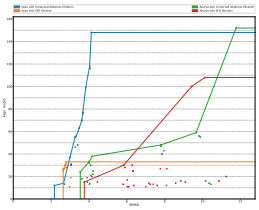
EFF-1 vs RBV-Rubi.



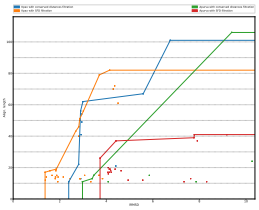
EFF-1 vs RVFV-Phlebo.



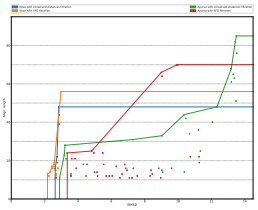
EFF-1 vs SFV-Alpha.



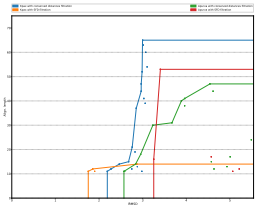
HRV-Hanta. vs DFV-Flavi.



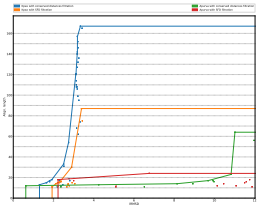
HRV-Hanta. vs EFF-1



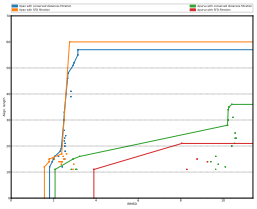
HRV-Hanta. vs HAP2-C



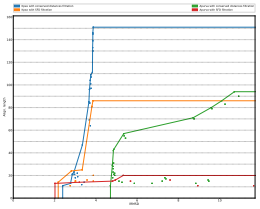
HRV-Hanta. vs RBV-Rubi.



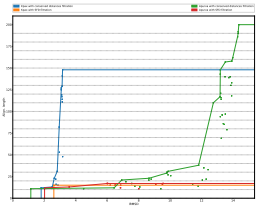
HRV-Hanta. vs RVFV-Phlebo.



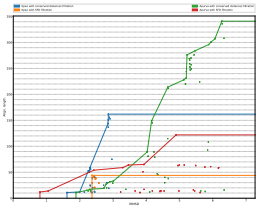
HRV-Hanta. vs SFV-Alpha.



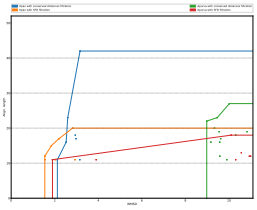
HRV-Hanta. vs TBEV



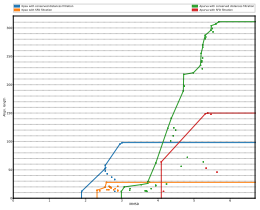
RBV-Rubi. vs HAP2-C



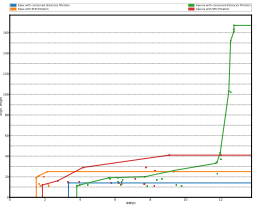
RVFV-Phlebo. vs HAP2-C



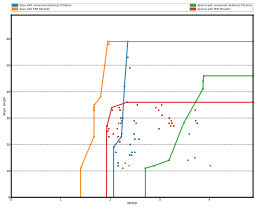
RVFV-Phlebo. vs RBV-Rubi.



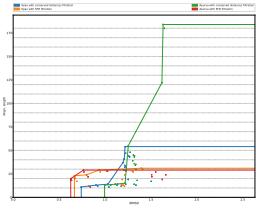
SFV-Alpha. vs HAP2-C



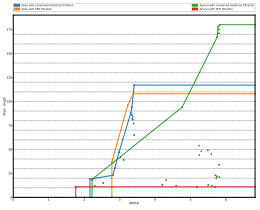
SFV-Alpha. vs RBV-Rubi.



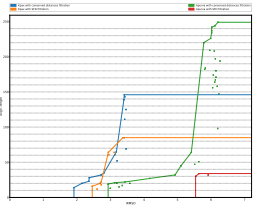
SFV-Alpha. vs RVFV-Phlebo.

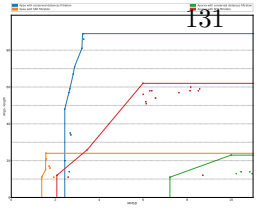


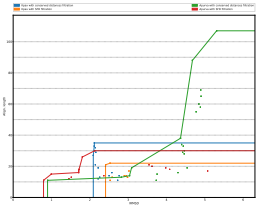
TBEV vs DFV-Flavi.

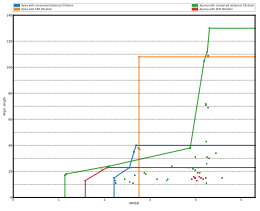


TBEV vs EFF-1









B.4.2 Comparisons against flexible aligners

Table B.3 Performances of `Align-Apurva-CD` on a set of difficult structures. Following the procedure in [YG03], we compare the performances of our method to `FATCAT` on 10 known 'difficult' cases for structural alignment. Parameters: $\tau_{/} = 0.8, \tau_{\text{PD}} = 0, \tau_{\text{MS}} = 10$

		Motifs			Apurva		Iter. Align (motif seeds)		`FATCAT`	
Pro1	Pro2	Size	RMSDcomb	Num. Components	Size	lRMSD	Size	lRMSD	Size	lRMSD
1BGE	2GMF	84.00	7.95	1.00	119.00	7.36	105.00	4.01	100	3.19
1CEW	1MOL	67.00	1.93	1.00	83.00	2.52	76.00	1.81	83	2.44
1CID	2RHE	79.00	3.02	1.00	103.00	3.94	99.00	3.50	100	3.11
1CRL	1EDE	281.00	9.96	1.00	297.00	11.45	222.00	8.05	269	3.55
1FXI	1UBQ	40.00	2.96	1.00	66.00	3.70	60.00	2.39	63	3.01
1TEN	3HHR	52.00	1.44	2.00	87.00	1.97	85.00	1.65	87	1.9
1TIE	4FGF	70.00	2.75	1.00	119.00	3.55	111.00	2.53	117	3.05
2AZA	1PAZ	79.00	2.71	1.00	88.00	4.45	87.00	4.03	87	3.01
2SIM	1NSB	193.00	4.66	1.00	317.00	5.72	281.00	3.12	286	3.07
3HLA	2RHE	51.00	3.31	3.00	88.00	4.53	76.00	3.06	79	2.81

Table B.4 Performances of `Align-Apurva-SFD` on a set of difficult structures. Following the procedure in [YG03], we compare the performances of our method to `FATCAT` on 10 known 'difficult' cases for structural alignment. Parameters: $\tau_{/} = 0.8, \tau_{\text{PD}} = 20, \tau_{\text{MS}} = 10$

		Motifs			Apurva		Iter. Align (motif seeds)		`FATCAT`	
Pro1	Pro2	Size	RMSDcomb	Num. Components	Size	lRMSD	Size	lRMSD	Size	lRMSD
1CEW	1MOL	17.00	0.97	1.00	83.00	2.52	78.00	1.89	83	2.44
1CID	2RHE	41.00	2.39	1.00	103.00	3.94	99.00	3.49	100	3.11
1TIE	4FGF	57.00	2.78	1.00	119.00	3.55	111.00	2.53	117	3.05
2AZA	1PAZ	12.00	1.38	1.00	88.00	4.45	86.00	3.33	87	3.01
3HLA	2RHE	36.00	3.49	1.00	88.00	4.53	82.00	3.38	79	2.81

Table B.5 Performances of `Align-Kpax-CD` on a set of difficult structures. Following the procedure in [YG03], we compare the performances of our method to `FATCAT` on 10 known 'difficult' cases for structural alignment. Parameters: $\tau_{/} = 0.8, \tau_{\text{PD}} = 0, \tau_{\text{MS}} = 10$

		Motifs			Kpax		Iter. Align (motif seeds)		`FATCAT`	
Pro1	Pro2	Size	RMSDcomb	Num. Components	Size	lRMSD	Size	lRMSD	Size	lRMSD
1BGE	2GMF	30.00	3.37	1.00	107.00	4.36	104.00	4.30	100	3.19
1CEW	1MOL	42.00	1.29	1.00	78.00	1.89	78.00	1.89	83	2.44
1CID	2RHE	46.00	1.70	1.00	97.00	2.73	97.00	2.73	100	3.11
1CRL	1EDE	246.00	7.47	1.00	226.00	8.08	227.00	8.07	269	3.55
1FXI	1UBQ	46.00	1.78	1.00	61.00	2.50	60.00	2.40	63	3.01
1TEN	3HHR	14.00	1.02	1.00	85.00	1.65	85.00	1.65	87	1.9
1TIE	4FGF	72.00	2.02	1.00	112.00	2.57	112.00	2.57	117	3.05
2AZA	1PAZ	73.00	2.46	1.00	87.00	3.53	87.00	3.83	87	3.01
2SIM	1NSB	170.00	2.52	1.00	285.00	3.14	284.00	3.14	286	3.07
3HLA	2RHE	63.00	2.58	1.00	76.00	3.24	76.00	3.18	79	2.81

Table B.6 Performances of `Align-Kpax-SFD` on a set of difficult structures. Following the procedure in [YG03], we compare the performances of our method to `FATCAT` on 10 known 'difficult' cases for structural alignment. Parameters: $\tau_{/} = 0.8, \tau_{\mathrm{PD}} = 20, \tau_{\mathrm{MS}} = 10$

		Motifs			Kpax		Iter. Align (motif seeds)		`FATCAT`	
Pro1	Pro2	Size	RMSDcomb	Num. Components	Size	lRMSD	Size	lRMSD	Size	lRMSD
1CEW	1MOL	15.00	0.69	1.00	78.00	1.89	78.00	1.89	83	2.44
1CID	2RHE	36.00	1.87	1.00	97.00	2.73	97.00	2.79	100	3.11
1CRL	1EDE	78.00	5.47	1.00	226.00	8.08	233.00	9.79	269	3.55
1TIE	4FGF	17.00	1.95	1.00	112.00	2.57	111.00	2.53	117	3.05
2AZA	1PAZ	26.00	3.05	1.00	87.00	3.53	86.00	3.36	87	3.01
3HLA	2RHE	28.00	1.93	1.00	76.00	3.24	77.00	3.25	79	2.81

Appendix C

Functional characterization of proteins with low sequence identity and loose structural conservation

C.1 Supplemental

C.1.1 Material: viruses

We study the structures from Fig. 5.1, using the SSE labels listed in SI Table C.1. We discarded the labels (*J0* and *G'*), present in two and one structures respectively, which prevents a comparison.

Semliki Forest virus (SFV)

- Order: Unassigned
- Family: Togaviridae
- Genus: Alphavirus
- Structure: 1RER

First isolated from mosquitoes by the Uganda Research institute, it was described by Smithburn and Haddow. It is usually transmitted through mosquito bites. Its structure is composed of an outer-shell of glycoproteins E1 and E2. As a model of virus life cycle, it has been extensively used in academic research. Other members of the Alphavirus family include the more famous Chikungunya.

Rift Valley Fever virus (RVFV)

- Order: Bunyavirales
- Family: Phenuiviridae
- Genus: Phlebovirus
- Structure: 6EGT

It was first reported in livestock in the Rift Valley of Kenya in the early 20^{th} century. It was then isolated in 1931. Transmission can arise through contact with infected animal skin or mosquito bites. It has an outer lipid envelope composed of two glycoproteins, G(N) and G(C), the former having a class 2 membrane fusion protein architecture.

Dengue fever virus (DENV2)

- Order: Unassigned
- Family: Flaviviridae
- Genus: Flavivirus
- Structure: 1OK8

Transmitted by a number of *Aedes* type mosquitoes, it causes the Dengue fever. More than 110 countries are concerned by Dengue and between 50 and 228 million people are infected each year. The first confirmed descriptions of an outbreak dates from 1779. Confirmation of transmission via mosquitoes was confirmed in the early 20th century.

Rubella virus

- Order: Unassigned
- Family: Togaviridae
- Genus: Rubivirus
- Structure: 4ADI

Only member of its genus, it is the pathogenic agent of rubella. The first clinical description was made by Friedrich Hoffmann in 1740. The virus was isolated in 1962. The capsid is enclosed in a lipid envelope in which E1 and E2 envelope proteins are embedded and for prominent spikes.

Hantaan River virus

- Order: Bunyavirales
- Family: Hantaviridae
- Genus: Orthohantavirus
- Structure: 5JLX

Causes the Korean hemorrhagic fever. The virus was isolated in 1976 from the lungs of field striped mice and was discovered to be the cause of renal failure, hemorrhage and shock in many American and Korean troops during the Korean war.

Tick-borne encephalitis virus (TBEV)

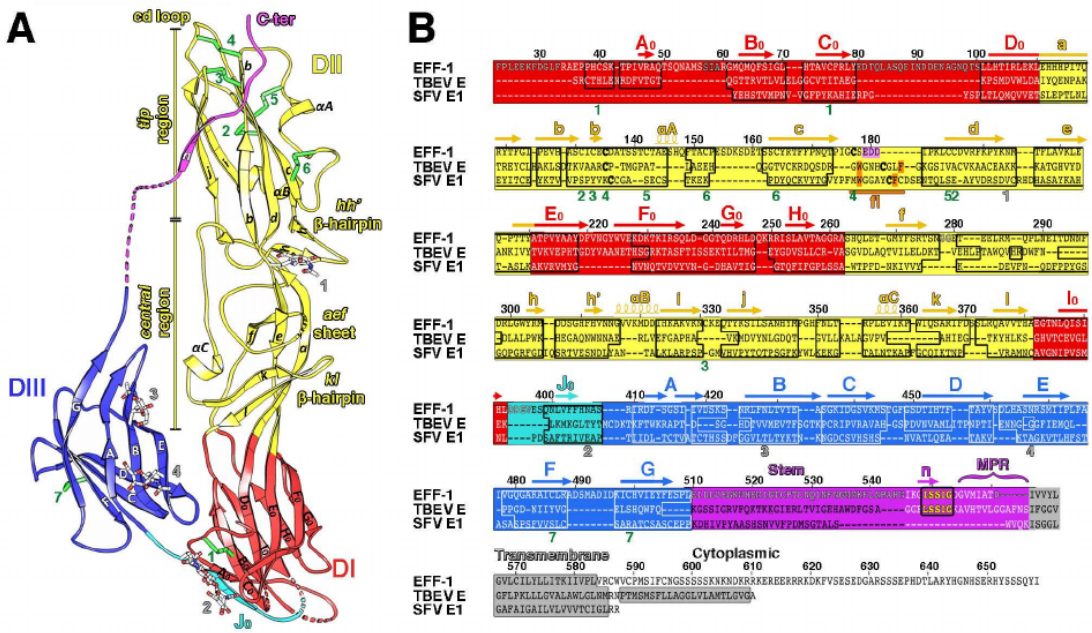
- Order: Unassigned
- Family: Flaviviridae
- Genus: Flavivirus
- Structure: 1URZ

Transmitted by the bite of several species of infected ticks, it causes Tick-borne encephalitis (TBE). It was first isolated in 1937, three sub-types are described: Western , Siberian or Far-Eastern TBEV. Russia and Europe report 5000 to 7000 cases annually.

Figure C.1 The 23 SSE labels ordered by domain. SSE labels which are common to all structures are in bold.

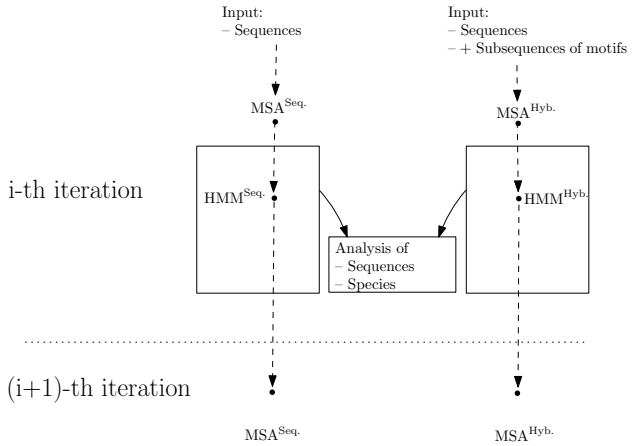
Domain	Labels	Count
1	A0, **B0, C0, D0, E0, F0, G0, H0, I0**, J0	10, **8**
2	**a, b, c, d, e, f, g, h, i, j**, k, l, j', h'	14, **10**
3	**A, B, C, E, F**, C', D, G, G'	9, **5**

Figure C.2 SSE elements on fusion domains, from [PVKV$^+$14]



C.1.2 Methods: bootstrap iterations

Figure C.3 Bootstrap iteration: sequences used. $MSA^{Seq.}$ and $MSA^{Hyb.}$ respectively refer to multiple sequence alignments involving the full protein sequences, and the sub-sequences associated to structural motifs. $HMM^{Seq.}$ and $HMM^{Hyb.}$ are the Hidden Markov Models built from these MSA.



C.1.3 Methods: structural conservation of their SSE elements

Recall that for fusion proteins, SSE involve 23 strands (SI Table C.1). We analyze the conservation of the labeled strands on the set of viral fusion proteins. Note that we only process strands which are contained in all structures (in bold in SI Table C.1).

Conservation of SSE in terms of homogeneous clusters. The matrix of pairwise distances between SSE elements can be used to build a dendogram.

In case of stringent conservation of the various SSE elements, one would expect to find, at a high enough level of separation on a dendogram, all ($N = 6$ viral structures) representatives of the same label. This situation is never observed (DI: Fig. C.4, DII: C.5, DIII:C.6), as for all domains, each label has at least one or two outliers which are significantly different from all the other ones. Additionally, the clusters contain a heterogeneous set of labels. For example, a cluster from the second domain exhibits all ten labels represented for that domain. The same situation is observed for the third domain (five out of five labels). In the first domain we can find a cluster containing six out of eight labels.

Figure C.4 Hierarchical clustering of SSE from D1 of viral structures. Note that the representatives of the same SSE seldom lie in the same cluster, hinting to little structural conservation.

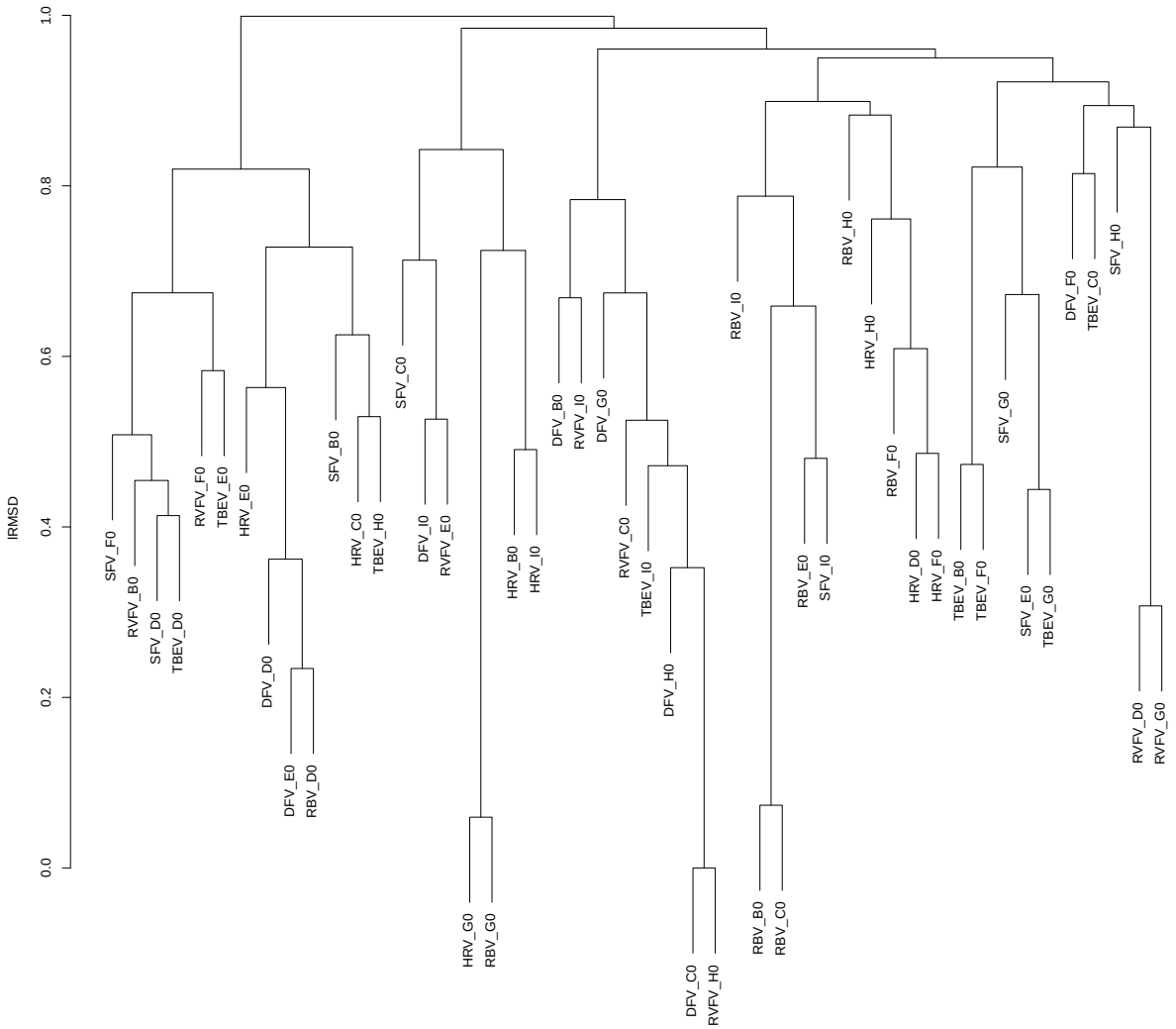


Figure C.5 Hierarchical clustering of SSE from D2 of viral structures. Note that the representatives of the same SSE seldom lie in the same cluster, hinting to little structural conservation.

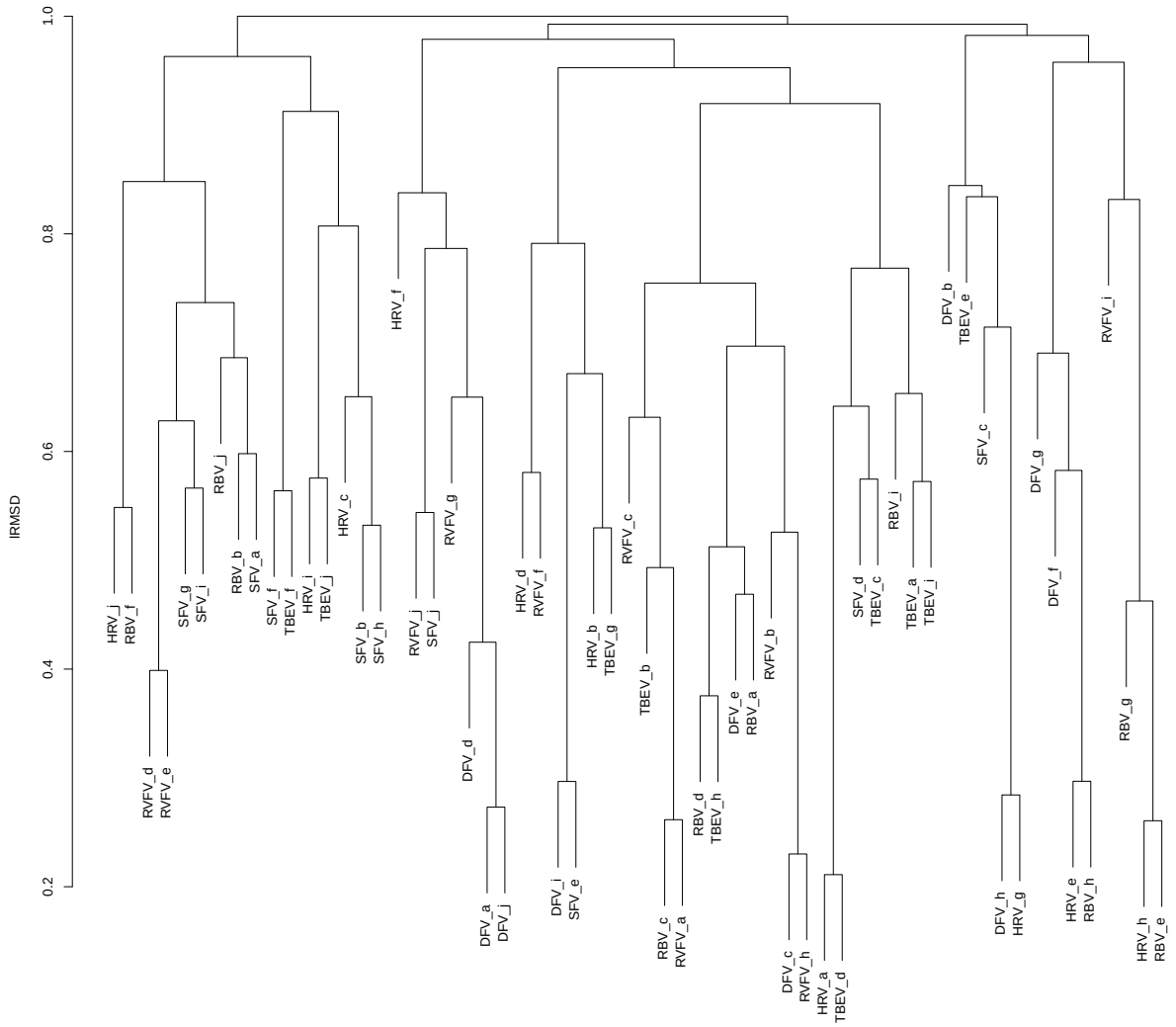
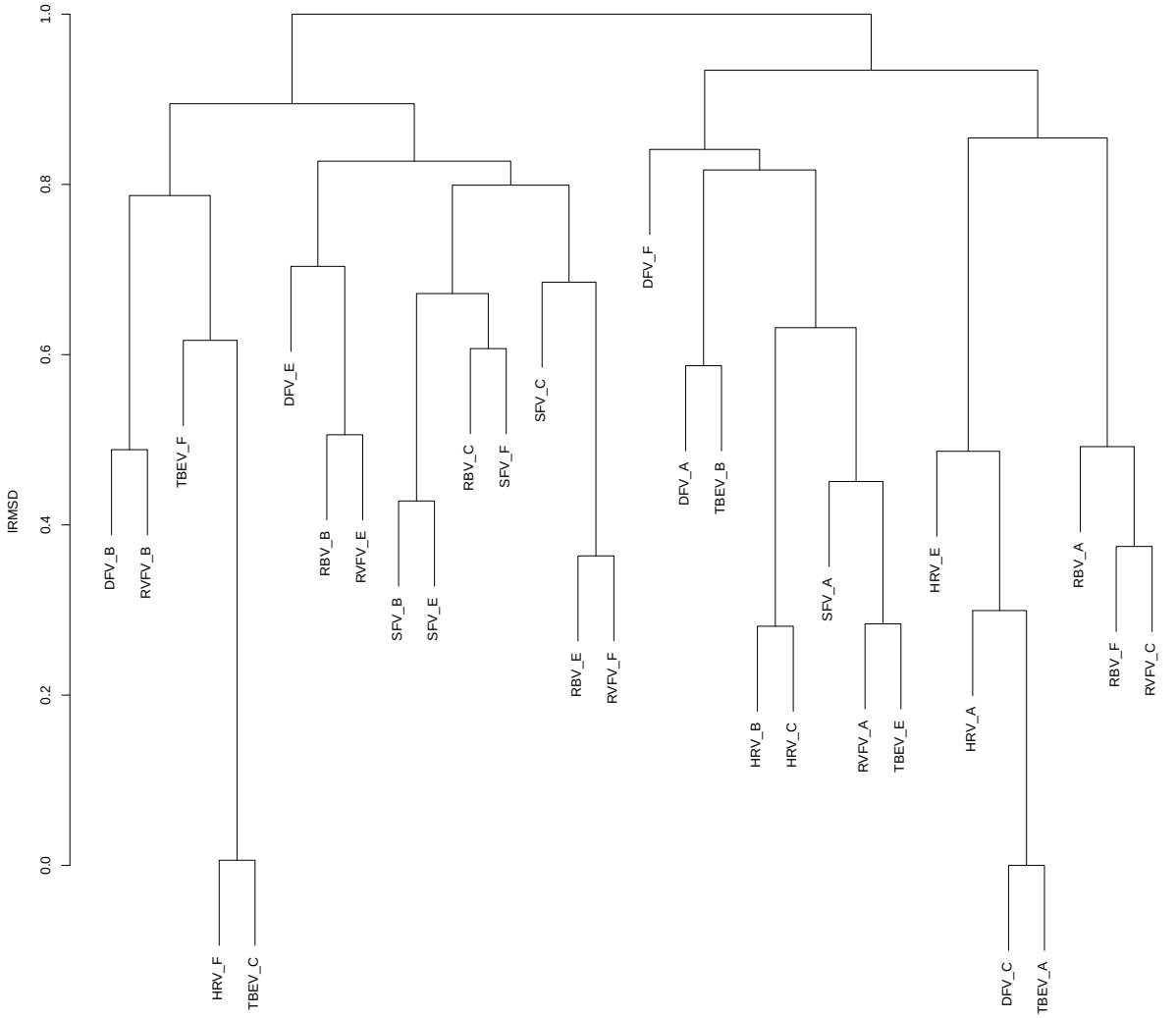


Figure C.6 Hierarchical clustering of SSE from D3 of viral structures. Note that the representatives of the same SSE seldom lie in the same cluster, hinting to little structural conservation.



HMM-HMM comparison: scores and p-values.

The alignment of protein sequences via HMM-sequence comparison was generalized using HMM-HMM comparisons [Söd04]. In a nutshell, a HMM-HMM comparison requires finding the path through the two HMM which maximizes the log-sum-of-odds-scores. (NB: Eq. 2.1 defines a log-odds; in the log-sum-of-odds score, the numerator is the sum of the probabilities yielded by the 2 HMM compared.) Finding this co-emission path is amenable do dynamic programming [Söd04].

Performances of profile HMM: querying `UniProtKB` and `NCBI Taxonomy`

As explained in section 5.2.4, the `UniProtKB` accession codes returned by `HMMER` are used to query the `NCBI Taxonomy` DB [Fed12], which contains the so-called tree of life organized in the general ranks (Super-kingdom, Kingdom, Phylum, Class, Order, Family, Genus, Species). In this classification, each super kingdom is associated with a tree (Rmk. C.1), and each node of these trees has a unique index.

Practically, we use SQLite to link `UniProtKB` and the `NCBI Taxonomy` database:

- NCBI provides a dump of their databases (`https://www.ncbi.nlm.nih.gov/guide/taxonomy/`). We then create a SQLite table containing the following values: Taxid (the unique identifier associated with each node in the tree), Name (the scientific name for the given node), Parent (the taxid referencing the parent of the current node), Rank (the rank of the node in the tree of life).

- `UniProtKB` provides a xml dump of their database (`https://www.uniprot.org/downloads`). We create a SQLite table containing the following values: `UniProtKB` Accession code, NCBI Taxid.

- Since `HMMER` [FCE11, Edd15] returns a list of `UniProtKB` accession codes (as well as values for each hit), we can now easily link a hit to a given species in the NCBI tree.

These software tools are provided in the `FunChaT` (Functional Characterization Tool) of the Structural Bioinformatics Library.

Remark C.1. *Note that, most likely out of convenience, the NCBI Taxonomy database gives the rank of super-kingdom to viruses. The recent discovery of more and more complex viruses (pandoravirus []) has rekindled the ancient debate as to whether viruses should be included in the tree of life. We do not take part and only call viruses a super-kingdom to stick with the NCBI nomenclature.*

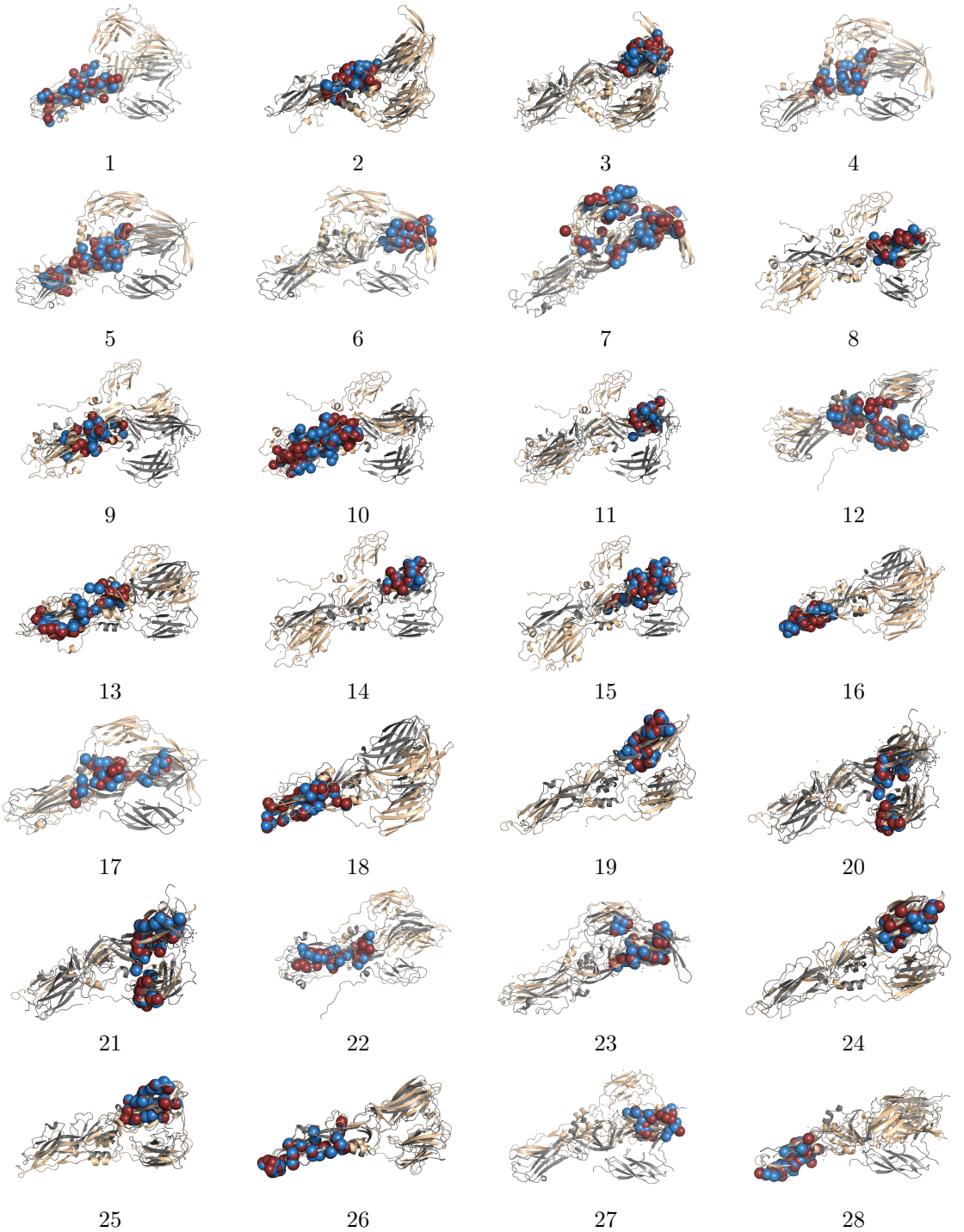
C.1.4 Results: motifs

Step one of our method consists in computing structural motifs (Sect. 5.2.2). In the sequel, we list and illustrate the specific motifs called nuggets (Def. 5.1; SI Tab. C.1 and SI Fig. C.7).

Table C.1 Top results for structural motifs, aka nuggets. Motifs involving at least 20 amino-acids, and with a lRMSD ratio ≤ 0.5, see Eq. (5.1).

First chain	Second chain	Ref. lRMSD	Motif index	lRMSD ratio	Size	Seq. Id.	Seq. Sim.
EFF-1	HRV-Hanta.	15.48	M_{25}	0.34	36	0.03	0.08
HAP2-C	DFV-Flavi.	8.05	M_{6}	0.50	25	0.04	0.08
HAP2-C	EFF-1	7.73	M_{27}	0.43	30	0.03	0.23
HAP2-C	HRV-Hanta.	17.08	M_{81}	0.21	23	0.04	0.17
HAP2-C	HRV-Hanta.	17.08	M_{88}	0.39	29	0.07	0.14
HAP2-C	HRV-Hanta.	17.08	M_{144}	0.30	23	0.04	0.22
HAP2-C	RVFV-Phlebo.	8.03	M_{47}	0.47	59	0.08	0.20
RBV-Rubi.	DFV-Flavi.	14.45	M_{13}	0.37	20	0.20	0.30
RBV-Rubi.	EFF-1	16.57	M_{34}	0.47	22	0.05	0.09
RBV-Rubi.	EFF-1	16.57	M_{109}	0.49	67	0.09	0.13
RBV-Rubi.	EFF-1	16.57	M_{110}	0.09	22	0.14	0.23
RBV-Rubi.	HRV-Hanta.	6.86	M_{5}	0.50	52	0.08	0.23
RBV-Rubi.	TBEV	14.28	M_{99}	0.42	24	0.00	0.17
RBV-Rubi.	TBEV	14.28	M_{107}	0.11	20	0.20	0.30
RBV-Rubi.	TBEV	14.28	M_{173}	0.18	34	0.12	0.21
RVFV-Phlebo.	DFV-Flavi.	6.42	M_{67}	0.28	31	0.19	0.29
RVFV-Phlebo.	HRV-Hanta.	15.03	M_{26}	0.48	24	0.12	0.25
RVFV-Phlebo.	TBEV	6.69	M_{16}	0.33	32	0.16	0.25
SFV-Alpha.	DFV-Flavi.	5.73	M_{38}	0.38	27	0.04	0.26
SFV-Alpha.	EFF-1	7.61	M_{2}	0.49	23	0.13	0.35
SFV-Alpha.	EFF-1	7.61	M_{3}	0.48	32	0.12	0.31
SFV-Alpha.	RBV-Rubi.	15.97	M_{40}	0.26	29	0.07	0.14
SFV-Alpha.	RVFV-Phlebo.	5.32	M_{2}	0.38	34	0.15	0.24
SFV-Alpha.	TBEV	5.73	M_{15}	0.34	21	0.00	0.24
TBEV	DFV-Flavi.	2.08	M_{17}	0.37	29	0.31	0.41
TBEV	DFV-Flavi.	2.08	M_{38}	0.49	28	0.68	0.68
TBEV	HRV-Hanta.	14.72	M_{11}	0.35	24	0.08	0.21
TBEV	HRV-Hanta.	14.72	M_{92}	0.36	20	0.10	0.20

Figure C.7 Overview of the structural motifs (aka nuggets) listed in Table C.1. Each plot displays the two structures hosting the motifs, and the C_α carbons of the motif. For a given pair, e.g. (DFV-Flavi., RBV-Rubi.), the first (resp. second) structure is represented as wheat (resp. grey) ribbons; likewise, the C_α carbons of the first (resp. second) structure are represented in red (resp. blue), using solvent accessible radii. The two structures are superimposed according to the rigid motion associated with the lRMSD calculation.



C.1.5 Results: HMM

Figure C.8 Sequence logo for DII of HAP2 structures, obtained with motifs detected by `Align-Kpax-CD`. Using the domain DII of the three available HAP2 structures, we build a biased HMM and query `UniProtKB`. The sequence logo is produced by the graphical interface of the hmmsearch web server `https://www.ebi.ac.uk/Tools/hmmer/search/hmmsearch`. The strong conservation of several cysteins, characteristic of class II fusion proteins, is clear in this figure.

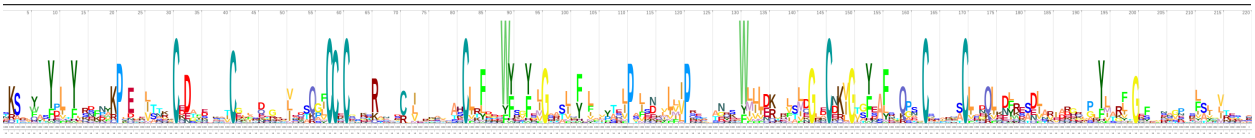
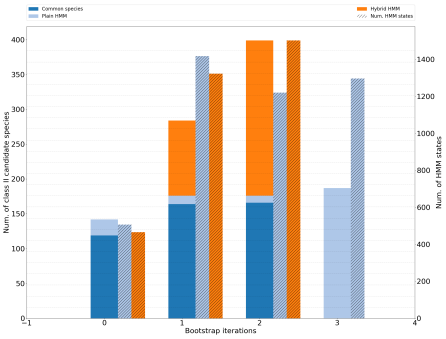
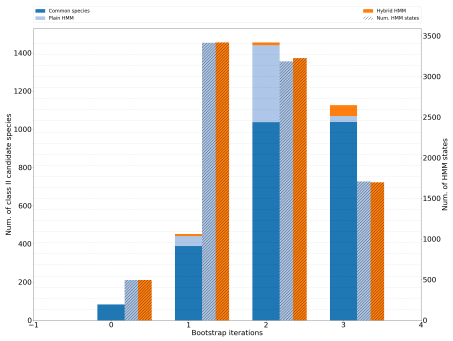


Figure C.9 Bootstrapping HMM models: hybrid HMM with $r_0 = 0.5$ ($\mathbf{HMM^{Hyb.}}$) versus plain HMM ($\mathbf{HMM^{Seq.}}$). See caption of Fig. 5.4 for the conventions.

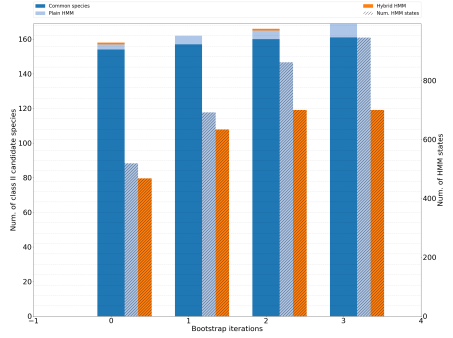


Group 1

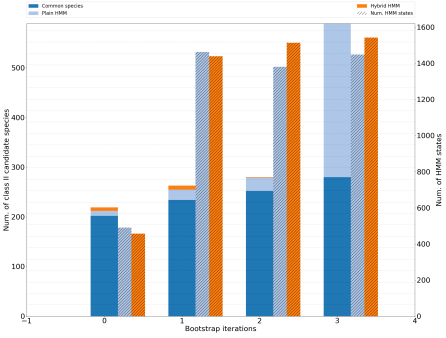


Group 2

Group 3

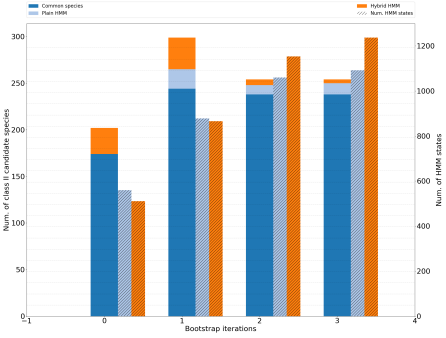


Group 4

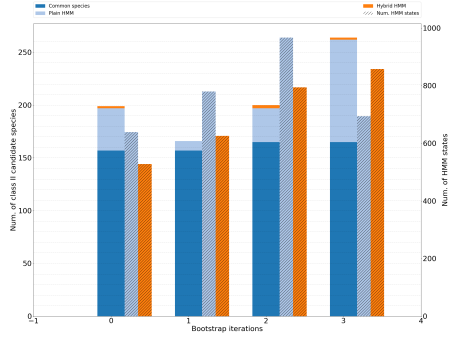


Group 5

Group 6

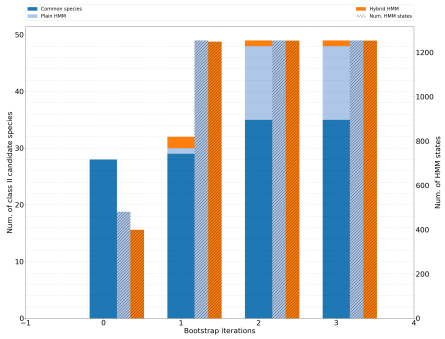


Group 7

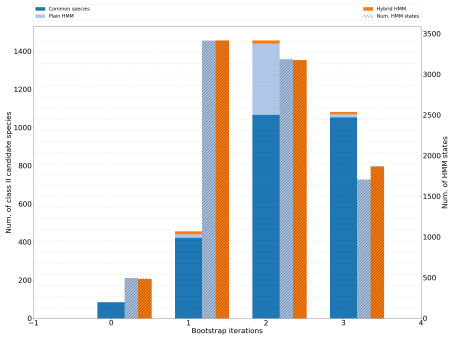


Group 8

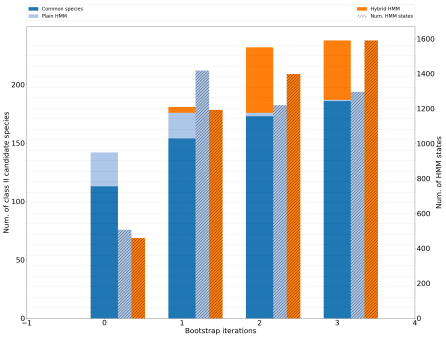
Figure C.10 Bootstrapping HMM models: hybrid HMM with $r_0 = 0.6$ ($\mathbf{HMM^{Hyb.}}$) versus plain HMM ($\mathbf{HMM^{Seq.}}$). See caption of Fig. 5.4 for the conventions.



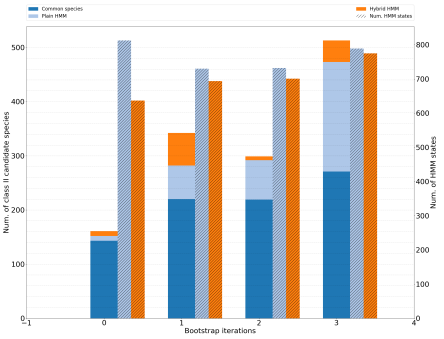
Group 1



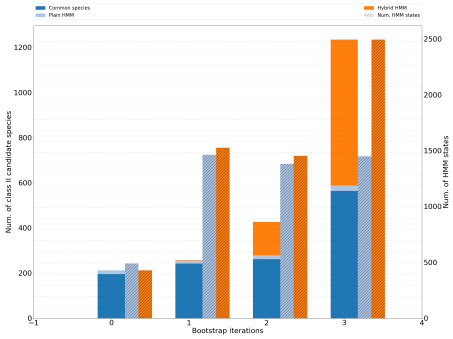
Group 2



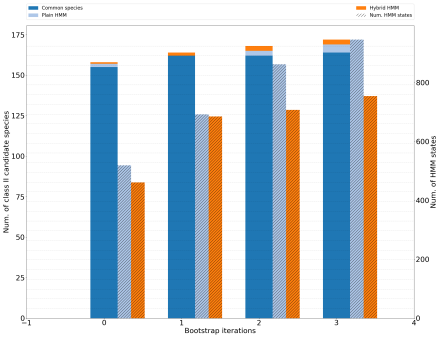
Group 3



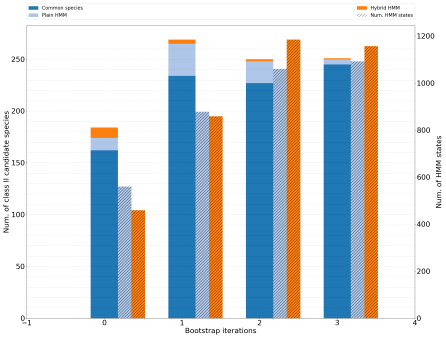
Group 4



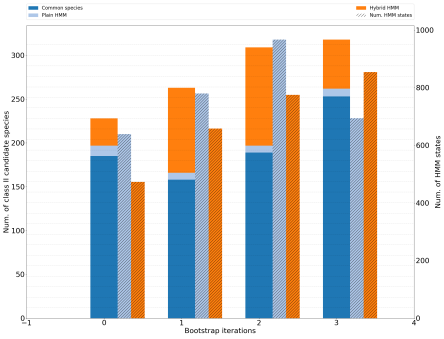
Group 5



Group 6

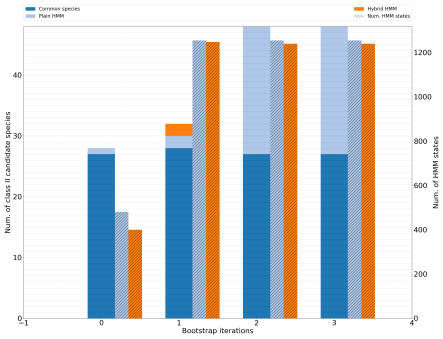


Group 7

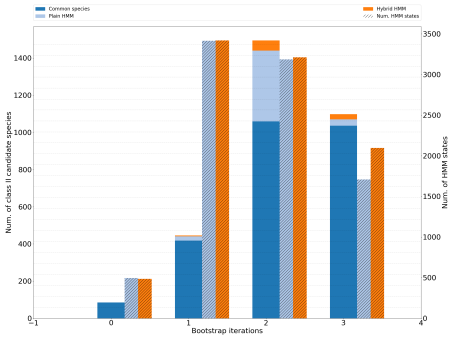


Group 8

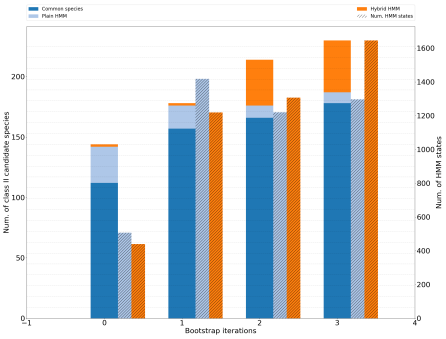
Figure C.11 Bootstrapping HMM models: hybrid HMM with $r_0 = 0.7$ ($\mathbf{HMM^{Hyb.}}$) versus plain HMM ($\mathbf{HMM^{Seq.}}$). See caption of Fig. 5.4 for the conventions.



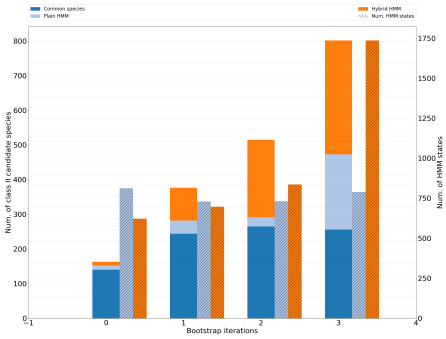
Group 1



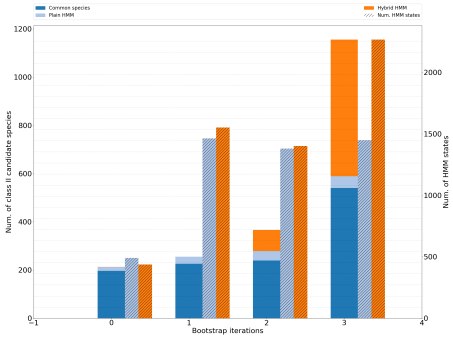
Group 2



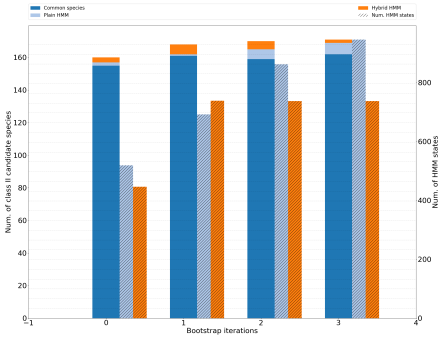
Group 3



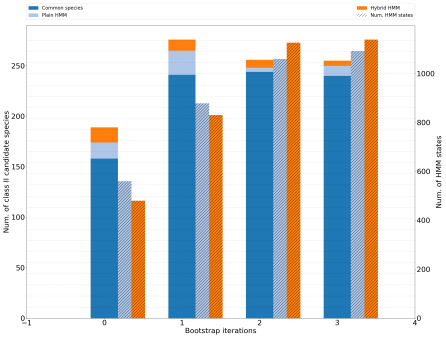
Group 4



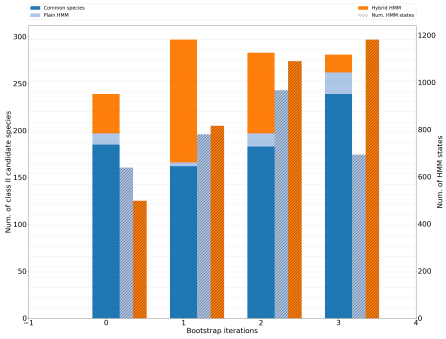
Group 5



Group 6

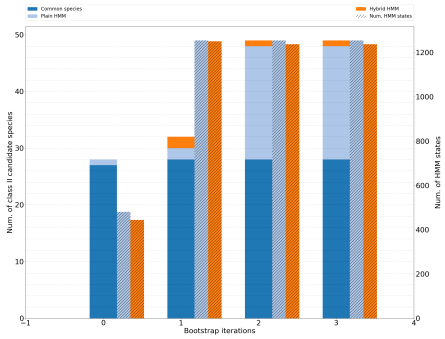


Group 7

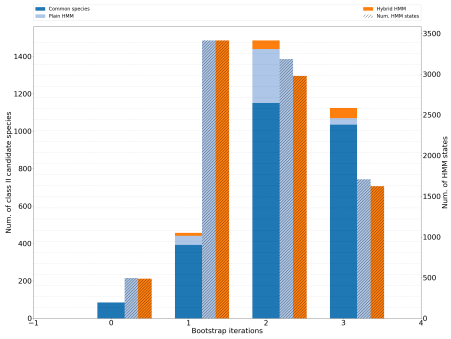


Group 8

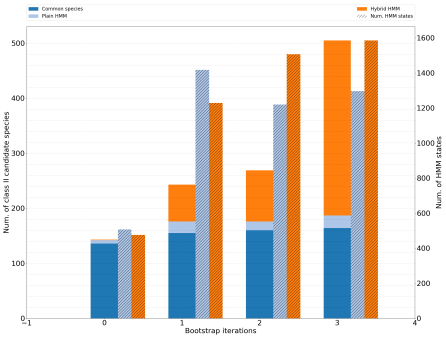
Figure C.12 Bootstrapping HMM models: hybrid HMM with $r_0 = 0.8$ ($\mathbf{HMM^{Hyb.}}$) versus plain HMM ($\mathbf{HMM^{Seq.}}$). See caption of Fig. 5.4 for the conventions.



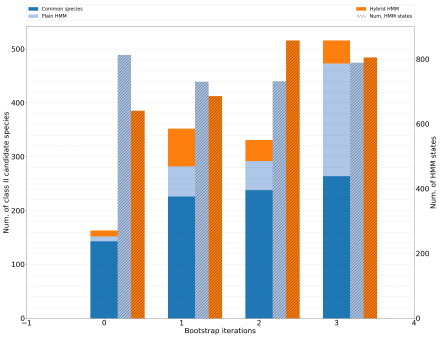
Group 1



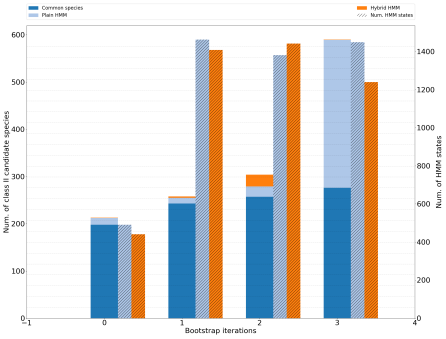
Group 2



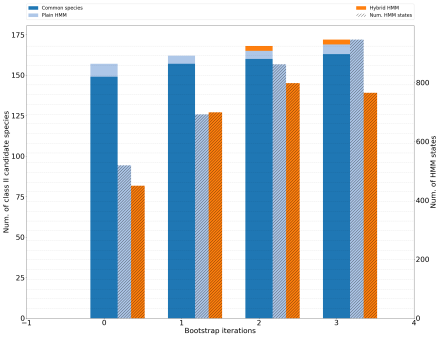
Group 3



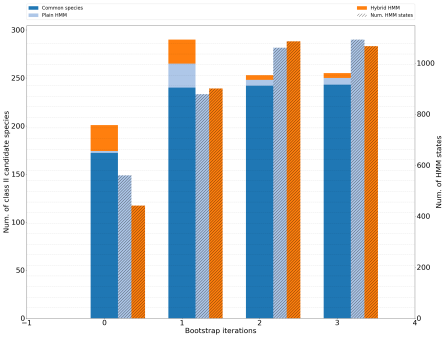
Group 4



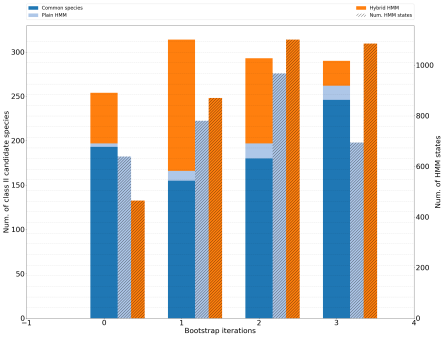
Group 5



Group 6



Group 7



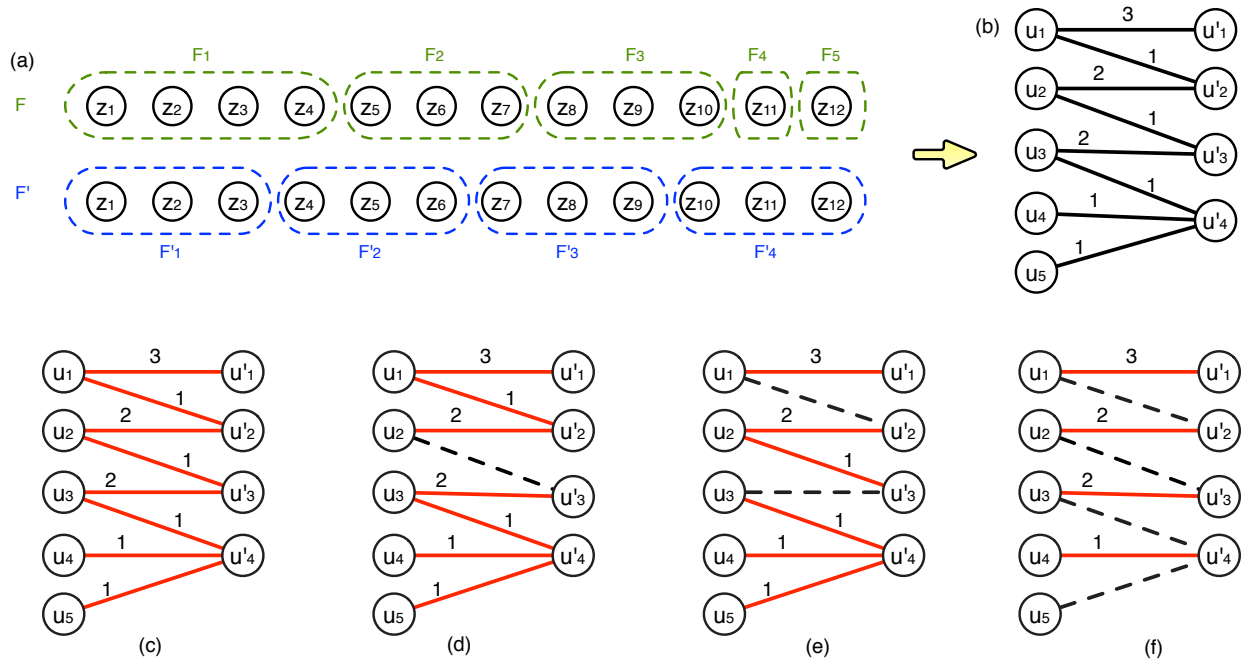
Group 8

Appendix D

On the stability of clusterings: the D-Family matching problem

D.1 Appendix - Detailed example

Figure D.1 **Simple instance of the D-family-matching problem and solutions: panels (c,d,e,f) represent optimal solutions for different values of D. (a)** Simple instance of the D-family-matching problem with $t = 12$, $r = 5$, $r' = 4$, and so $n = 9$. The family F contains five sets and the family F' contains four sets. **(b)** Intersection graph G. **(c)** Optimal solution $\mathcal{S}$ for $D \geq 7$ with $\Phi(\mathcal{S}) = \Phi_D(G) = 12$. **(d)** Optimal solution $\mathcal{S}$ for $D = 3$ with $\Phi(\mathcal{S}) = \Phi_3(G) = 11$. **(e)** Optimal solution $\mathcal{S}$ for $D = 2$ with $\Phi(\mathcal{S}) = \Phi_2(G) = 9$. Observe that there is another optimal solution by removing the two edges $\{u_2, u'_3\}$ and $\{u_3, u'_4\}$ and by adding the edge $\{u_3, u'_3\}$. **(f)** Optimal solution $\mathcal{S}$ for $D = 1$ with $\Phi(\mathcal{S}) = \Phi_1(G) = 8$.



D.2 Appendix - Table of notations

Notation	Definition
$Z = \{z_1, \ldots, z_t\}$	Set of $t \geq 1$ elements
$F = \{F_1, \ldots, F_r\}$	Family of $r \geq 1$ disjoint subsets of Z
$F' = \{F'_1, \ldots, F'_{r'}\}$	Family of $r' \geq 1$ disjoint subsets of Z
$G = (V, E, w)$	Intersection graph of $n \geq 1$ nodes and $m \geq 1$ edges
$N_G(v) = \{v' \mid \{v, v'\} \in E\}$	Set of neighbors of node $v \in V$
$\Delta = \max_{v \in V} \lvert N_G(v)\rvert$	Maximum degree of G
$cc(G)$	Set of maximal connected components of G
$\mathcal{S} = \{S_1, \ldots, S_k\}$	D-family-matching
$\Phi(\mathcal{S}) = \sum_{i=1}^{k} \sum_{e \in E(G[S_i])} w_e$	Score of a D-family-matching $\mathcal{S}$
$\mathcal{S}_D(G)$	Set of all D-family-matching for G
$\Phi_D(G) = \max_{\mathcal{S} \in \mathcal{S}(G,D)} \Phi(\mathcal{S})$	Optimal score for the D-family-matching problem
$\mathcal{S}_D(G, T_r)$	Set of all D-family-matching constrained by T_r
$\Phi_D(G, T_r) = \max_{\mathcal{S} \in \mathcal{S}_D(G,T_r)} \Phi(\mathcal{S})$	Optimal score for the D-family-matching problem constrained by T_r

D.3 Appendix - Equivalent definition of the D-family-matching problem

We first prove Lemma 6.1.

Proof of Lemma 6.1. Without loss of generality, we assume that G is connected (otherwise, we prove the result for every maximal connected component). We prove the result by induction on the number of nodes n. Let $V = U \cup U'$. Consider first that $n = |U \cup U'| = 2$. Let $U = \{u_1\}$ and $U' = \{u'_1\}$. We construct Z, F, and F' as follows. Set $Z = \{z_1, \ldots, z_t\}$ with $t = w_{u_1,u'_1}$. Set $F = \{F_1\}$ with $F_1 = \{z_1, \ldots, z_t\}$ and set $F' = \{F'_1\}$ with $F'_1 = \{z_1, \ldots, z_t\}$. Thus, G is the intersection graph for Z, F, and F'.

Suppose now that it is true for every edge-weighted bipartite graph composed of at most n nodes and such that the weights are positive integers. We prove that it is also true for every edge-weighted bipartite graph $G = (U, U', E, w)$ such that $|U \cup U'| = n+1$ and such that the weights are positive integers. Consider a node $x \in U \cup U'$ such that $G' = G[(U \cup U') \setminus \{x\}]$ is connected. By induction hypothesis, G' is an intersection graph. We define $Z^{G'}$, $F^{G'}$, and $F'^{G'}$ corresponding to G' as follows. Let $Z^{G'} = \{z_1, \ldots, z_t\}$, $F^{G'} = \{F_1, \ldots, F_r\}$, and $F'^{G'} = \{F'_1, \ldots, F'_{r'}\}$. Without loss of generality, assume that $x \in U$. Let $N_G(x) = \{u'_1, \ldots, u'_{d_x}\}$, where d_x is the number of neighbors of x in G. Without loss of generality, assume that u'_i corresponds to F'_i for every $i \in \{1, \ldots, d_x\}$ (we permute the indices otherwise). Set $w_x = \sum_{i=1}^{d_x} w_{x,u'_i}$. We construct Z, F, and F' corresponding to G as follows. Set $Z = Z^{G'} \cup \{z_{t+1}, \ldots, z_{t+w_x}\} = \{z_1, \ldots, z_t, z_{t+1}, \ldots, z_{t+w_x}\}$. Set $F = \{F_1, \ldots, F_r, F_{r+1}\}$, where $F_{r+1} = \{z_{t+1}, \ldots, z_{t+w_x}\}$. For every $i, j \in \{1, \ldots, d_x\}$, $i \neq j$, let $X_i \subseteq \{z_{t+1}, \ldots, z_{t+w_x}\}$ with $|X_i| = w_{x,u'_i}$ and such that $X_i \cap X_j = \emptyset$. Finally, set $F' = \{F''_1, \ldots, F''_{r'}\}$, where $F''_i = F'_i \cup X_i$ for every $i \in \{1, \ldots, d_x\}$, and $F''_i = F'_i$ for every $i \in \{d_x + 1, \ldots, r'\}$. We get that G is the intersection graph for Z, F, and F'. Thus, the result is true for every edge-weighted bipartite graph $G = (U, U', E, w)$ such that $2 \leq |U \cup U'| \leq n+1$ and such that the weights are integers. □

We now define an equivalent definition of the D-family-matching.

Definition. D.1 (D-family-matching)**.** *Let $D \in \mathbb{N}^+$. A D-family-matching is a family $\mathcal{P} = \{P_1, \ldots, P_k\}$, $k \geq 1$, of subsets of $F \cup F' = \{F_1, \ldots, F_r, F'_1, \ldots, F'_{r'}\}$ such that, for every $i, j \in \{1, \ldots, k\}$, $i \neq j$, then: $P_i \subseteq F \cup F'$, $P_i \neq \emptyset$, $P_i \cap P_j = \emptyset$, and $\mathcal{P}$ must satisfy the* diameter constraints*: for every $H, H' \in P_i$, then there exists a sequence $(H_0, \ldots, H_d)$ such that $d \leq D$, $H_0 = H$, $H_d = H'$, $H_j \in P_i$, and $H_j \cap H_{j+1} \neq \emptyset$ for every $j \in \{0, \ldots, d-1\}$.*

The score $f(\mathcal{P})$ of a D-family-matching $\mathcal{P}$ is defined as follows:

$$f(\mathcal{P}) = \sum_{i=1}^{k} |(P_i \cap_F F) \cap_Z (P_i \cap_{F'} F')|.$$

Let $\mathcal{P}_D(F, F')$ be the set of all D-family-matching for F, F', and D. We now formalize an equivalent definition of the D-family-matching problem.

Definition. D.2 (D-family-matching problem)**.** *Let $D \in \mathbb{N}^+$. The D-family-matching problem consists in determining a D-family-matching that maximizes the score f. Formally, we aim at computing:*

$$f_D(F, F') = \max_{\mathcal{P} \in \mathcal{P}_D(F,F')} f(\mathcal{P}).$$

Finally, we obtain the following property showing the equivalence between the two definitions of the D-family-matching problem.

Property D.1. *Let $D \in \mathbb{N}^+$. Let $L \geq 0$ be any positive real number. Consider any instance of the D-family-matching problem defined by Z, F, and F', and consider the associated intersection graph G. Then, there is a D-family-matching $\mathcal{P}$ for Z, F, and F', such that $f(\mathcal{P}) \geq L$ if and only if there is a D-family-matching $\mathcal{S}$ of G such that $\Phi(\mathcal{S}) \geq L$.*

D.4 Hardness of the D-family-matching problem and greedy strategies

D.4.1 Proof of Theorem 6.1

This section is devoted to the proof of Theorem 6.1. For the sake of readability, we splitted this proof into three parts: Theorems D.2, D.5 and D.6. Notice that the last two proofs are quite similar.

Let us first recall the definition of L-reduction in order to transfer approximation lower bounds.

Definition. D.3. *[PY91] Let Π and Π' be two maximization problems. We say that Π L-reduces to Π' is there are two polynomial-time algorithms f, g and constants $\alpha, \beta > 0$ such that for each instance I of Π:*

1. *Algorithm f produces an instance $I' = f(I)$ of Π' such that the optima of I and I', denoted by $OPT_\Pi(I)$ and $OPT_{\Pi'}(I')$, respectively, satisfy $OPT_{\Pi'}(I') \leq OPT_\Pi(I)$;*
2. *Given any solution of I' with cost c', algorithm g produces a solution of I with cost c such that $OPT_\Pi(I) - c \leq \beta(OPT_{\Pi'}(I') - c')$.*

It is known that if Π is APX-hard and L-reduces to Π', then Π' is APX-hard as well. In that case, Π' does not admit a $PTAS$ (Polynomial Time Approximation Scheme) unless $P = NP$.

Part I: $\Delta = 4$, fixed values of weights

Theorem. D.2. *For any $D \geq 2$, the D-family-matching problem is APX-hard even if the maximum degree Δ is at most 4 and the weights are 2 and 5.*

In our reduction, we use a special case of set packing problem, a well known NP-complete problem [Kar72]. Given a universe $X = \{x_1, \ldots, x_t\}$ of $t \geq 1$ elements and a family $Y = \{Y_1, \ldots, Y_p\}$ of $p \geq 1$ subsets of X, a **packing** is a subfamily $\mathcal{C} \subseteq Y$ of subsets such that all set in $\mathcal{C}$ are pairwise disjoint, that is $Y_i \cap Y_j = \emptyset$ for all $Y_i, Y_j \in \mathcal{C}$, $i \neq j$. Given X, Y, and an integer $k \geq 1$, **set packing problem** consists in determining whether there exists a packing $\mathcal{C}$ of size $|\mathcal{C}| = k$. Set packing problem is NP-complete even if $|Y_i| = 3$ for every $i \in \{1, \ldots, p\}$. Furthermore, if x_i is in at most 3 sets, for every $i \in \{1, \ldots, t\}$, then the problem

is APX-complete [Kan91]. For such a special case, the problem is known as 3-dimensional matching problem.

We initially prove the result for $D = 2$. Consider any instance $\mathcal{I}_{sp}$ of set packing problem: a universe $X = \{x_1, \ldots, x_t\}$, a family $Y = \{Y_1, \ldots, Y_p\}$ of subsets of X, and an integer $k \geq 1$. We assume that $|Y_i| = 3$ for all $i \in \{1, \ldots, p\}$ and that x_i is in at most 3 sets. We first construct the intersection graph G of the 2-family-matching problem (Definition D.4).

Definition. D.4 (Construction of the intersection graph G for the 2-family-matching problem)**.** *The intersection graph $G = (U, U', E, w)$ is defined as follows.*

- *Set $U = U_1 \cup U_2$, where*
  - *$U_1 = \{u_1^1, \ldots, u_p^1\}$ corresponds to Y*
  - *and $U_2 = \{u_1^2, \ldots, u_p^2\}$.*
- *Set $U' = U'_1 \cup U'_2$, where*
  - *$U'_1 = \{u_1'^1, \ldots, u_t'^1\}$ corresponds to X*
  - *and $U'_2 = \{u_1'^2, \ldots, u_p'^2\}$.*
- *Set $E = E_a \cup E_b \cup E_c$, where*
  - *$E_a = \{\{u_i^2, u_i'^2\} \mid 1 \leq i \leq p\}$,*
  - *$E_b = \{\{u_i^1, u_i'^2\} \mid 1 \leq i \leq p\}$,*
  - *and $E_c = \{\{u_i^1, u_j'^1\} \mid x_j \in Y_i, 1 \leq i \leq p, 1 \leq j \leq t\}$.*
- *Set $w_e = 5$ for every $e \in E_a \cup E_b$ and $w_e = 2$ for every $e \in E_c$.*

Observe that the maximum degree Δ of G is at 4.

Lemma. D.1. *If there is a solution $\mathcal{C}$ for the instance $\mathcal{I}_{sp}$ of set packing problem such that $|\mathcal{C}| \geq k$, then there is a solution $\mathcal{S}$ for the 2-family-matching problem for G such that $\Phi(\mathcal{S}) \geq 10p + k$.*

Proof of Lemma D.1. Consider any solution $\mathcal{C}$ for the instance $\mathcal{I}_{sp}$ of set packing problem such that $|\mathcal{C}| = k$. We construct a solution $\mathcal{S}$ for the 2-family-matching problem for G such that $\Phi(\mathcal{S}) = 10p + k$. Assume that $\mathcal{C} = \{Y_1, \ldots, Y_k\}$ (we permute the indices otherwise). Let $\mathcal{S} = \{S_1, \ldots, S_p\}$, where $S_i = \{u_i^1\} \cup N_G(u_i^1)$ for every $i \in \{1, \ldots, k\}$ and $S_i = \{u_i^1, u_i'^2, u_i^2\}$ for every $i \in \{k+1, \ldots, p\}$. The sets are disjoint. In other words, for every $i, j \in \{1, \ldots, p\}$, $i \neq j$, then $S_i \cap S_j = \emptyset$ because $\mathcal{C}$ is a set packing and, by construction of G, we have $N_G(u_{i'}) \cap N_G(u_{j'}) = \emptyset$ for every $i', j' \in \{1, \ldots, k\}$, $i' \neq j'$. Furthermore, for every $i \in \{1, \ldots, p\}$, the diameter of $G[S_i]$ is at most 2. Finally, we get

$$\Phi(\mathcal{S}) = \Phi(\{S_1, \ldots, S_k\}) + \Phi(\{S_{k+1}, \ldots, S_p\}) = 11k + 10(p - k) = 10p + k.$$

Thus, we have proved that $\mathcal{S}$ is a solution for the 2-family-matching problem for G such that $\Phi(\mathcal{S}) = 10p + k$. □

Lemma. D.2. *If there is a solution $\mathcal{S}$ for the 2-family-matching problem for G such that $\Phi(\mathcal{S}) \geq 10p + k$, then there is a solution $\mathcal{C}$ for the instance $\mathcal{I}_{sp}$ of set packing problem such that $|\mathcal{C}| \geq k$.*

Proof of Lemma D.2. Consider any optimal solution $\mathcal{S}$ for the 2-family-matching problem. Without loss of generality, we assume that $\mathcal{S}$ contains the smallest number of sets. In other words, $|\mathcal{S}| \leq |\mathcal{S}'|$ for any solution $\mathcal{S}'$ such that $\Phi(\mathcal{S}') = \Phi(\mathcal{S})$. We deduce that every set of $\mathcal{S}$ contains at least two nodes. Otherwise, we can remove such single sets without decreasing the score. We first prove the following claim.

Claim D.3. *Consider any node $u^1 \in U_1$. Let $S_1 \in \mathcal{S}$ be such that $u^1 \in S$. Then, $|U'_1 \cap S_1| \in \{0, 3\}$.*

Proof of Claim D.3. By contradiction. Assume that there exists a node $u^1 \in U_1$ and a set $S_1 \in \mathcal{S}$ such that $u^1 \in S$ and $|U'_1 \cap S_1| \in \{1, 2\}$. Let $u'^2 \in U'_2$ be such that $u'^2 \in N_G(u^1)$ and let u^2 be such that $\{u'^2, u^2\} \in E$. There are two cases.

- First, assume that $u'^2 \in S_1$. Thus, for any $S' \in \mathcal{S}$, then $u^2 \notin S'$. In particular, $u^2 \notin S_1$ because otherwise $G[S_1]$ would have diameter at least three. We get that $|S_1| \in \{3, 4\}$ and $\sum_{e \in E(G[S_1])} w_e \in \{7, 9\}$. Without loss of generality, assume that $\mathcal{S} = \{S_1, \ldots, S_{p'}\}$ for some $p' \geq 1$. We construct $\mathcal{S}'$ from $\mathcal{S}$ as follows. Set $\mathcal{S}' = \{S'_1, S_2, \ldots, S_{p'}\}$, where $S'_1 = \{u'^2, u^2\}$. We get that $\Phi(\mathcal{S}') - \Phi(\mathcal{S}) = 1$ if $|U'_1 \cap S_1| = 2$ and $\Phi(\mathcal{S}') - \Phi(\mathcal{S}) = 3$ if $|U'_1 \cap S_1| = 1$. A contradiction because $\mathcal{S}$ is an optimal solution for the 2-family-matching problem.

- Second, assume that $u'^2 \notin S_1$. There are two sub-cases.

  - There exists $S_2 \in \mathcal{S}$ such that $\{u'^2, u^2\} \in E(G[S_2])$. We necessarily have $|S_2| = 2$ because $N_G(u^2) = \{u'^2\}$ and $N_G(u'^2) = \{u^2, u^1\}$. Without loss of generality, assume that $\mathcal{S} = \{S_1, \ldots, S_{p'}\}$ for some $p' \geq 1$. We construct $\mathcal{S}'$ from $\mathcal{S}$ as follows. Set $\mathcal{S}' = \{S'_2, S_3, \ldots, S_{p'}\}$, where $S'_2 = S_2 \cup \{u^1\} = \{u'^2, u^2, u^1\}$. Since $w_{u'^2,u^2} = w_{u'^2,u^1} = 5$, we get that $\Phi(\mathcal{S}') - \Phi(\mathcal{S}) = 1$ if $|U'_1 \cap S_1| = 2$ and $\Phi(\mathcal{S}') - \Phi(\mathcal{S}) = 3$ if $|U'_1 \cap S_1| = 1$. A contradiction because $\mathcal{S}$ is an optimal solution for the 2-family-matching problem.
  - For any $S' \in \mathcal{S}$, then $\{u'^2, u^2\} \notin E(G[S'])$. We have that both u'^2 and u^2 are not in a set of $\mathcal{S}$ because $N_G(u^2) = \{u'^2\}$ and $N_G(u'^2) = \{u^2, u^1\}$. Indeed, recall that every set of $\mathcal{S}$ contains at least two nodes. (If it is not the case, we can remove u'^2 and u^2 without decreasing the score.) Thus, assume that $\mathcal{S} = \{S_1, \ldots, S_{p'}\}$ for some $p' \geq 1$ and for every $i \in \{1, \ldots, p'\}$, then $u'^2 \notin S_i$ and $u^2 \notin S_i$. We construct $\mathcal{S}'$ from $\mathcal{S}$ as follows. Set $\mathcal{S}' = \{S'_1, S_2, \ldots, S_{p'}\}$, where $S'_1 = \{u'^2, u^2, u^1\}$. Again, we get that $\Phi(\mathcal{S}') - \Phi(\mathcal{S}) \in \{1, 3\}$. A contradiction because $\mathcal{S}$ is an optimal solution for the 2-family-matching problem.

Thus, we have proved that $|U'_1 \cap S_1| \in \{0, 3\}$. □

By Claim D.3, we get that for every $i \in \{1, \ldots, p\}$, then there exists $S \in \mathcal{S}$ such that

- either $S = \{u'^2_i\} \cup N_G(u'^2_i) = \{u^2_i, u'^2_i, u^1_i\}$ and $\sum_{e \in E(G[S])} w_e = 10$
- or $S = \{u^1_i\} \cup N_G(u^1_i) = \{u'^2_i, u^1_i, u'^1_{j_1}, u'^1_{j_2}, u'^1_{j_3}\}$ and $\sum_{e \in E(G[S])} w_e = 11$, where $N_G(u^1_i) \cap U'_1 = \{u'^1_{j_1}, u'^1_{j_2}, u'^1_{j_3}\}$ for some $j_1, j_2, j_3 \in \{1, \ldots, t\}$, $j_1 < j_2 < j_3$.

Observe that we cannot have two sets $S, S' \in \mathcal{S}$ such that $S = \{u^2_i, u'^2_i\}$ and $S' = \{u^1_i, u'^1_{j_1}, u'^1_{j_2}, u'^1_{j_3}\}$ because we have assumed that $\mathcal{S}$ has a minimum number of sets. Indeed, $\sum_{e \in E(G[S])} w_e + \sum_{e \in E(G[S'])} w_e = 11$ but it is possible to consider one single set with same score (second case described before).

We deduce the following claim.

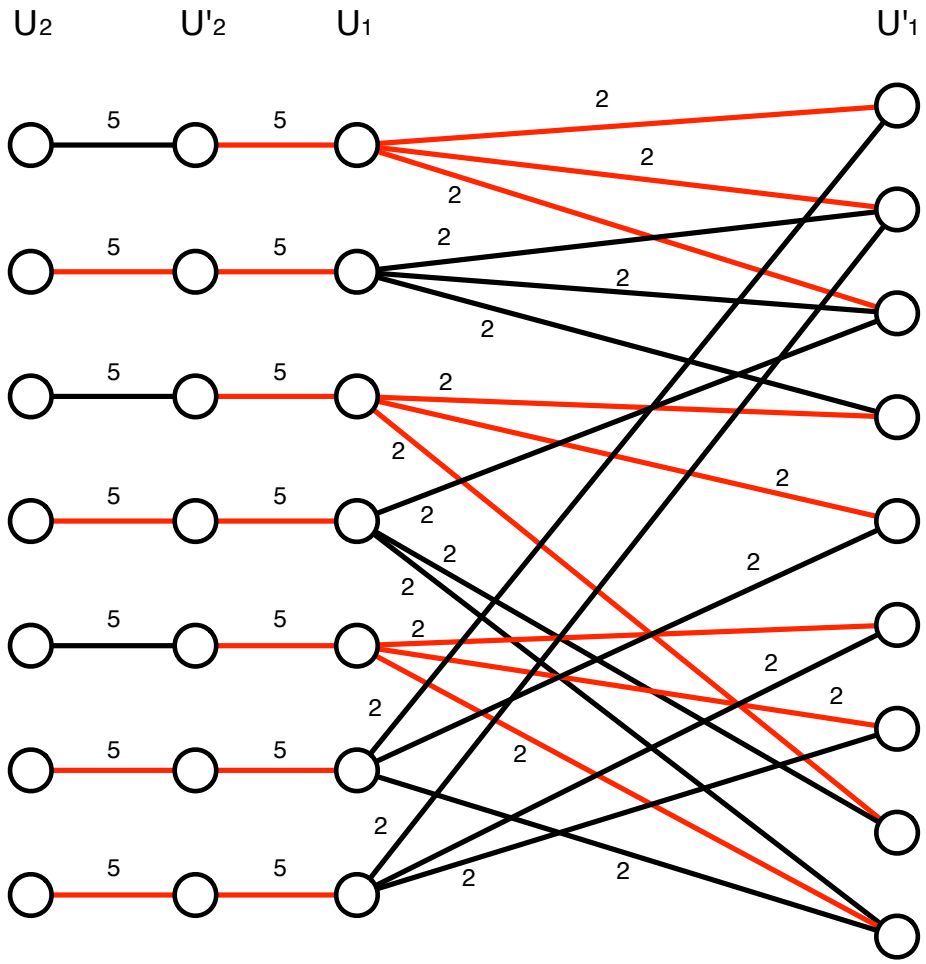
Claim D.4. *For any node $u'^1 \in U'_1$ and for any $S \in \mathcal{S}$, then $|N_G(u'^1) \cap S| \in \{0, 1\}$.*

By previous remarks and Claim D.4, we get that $\mathcal{S}$ contains exactly p sets. Recall that we assume that every set contains at least two nodes. Let $\mathcal{S} = \{S_1, \ldots, S_p\}$. We get that $\sum_{e \in E(G[S_i])} w_e \in \{10, 11\}$ and $\Phi(\mathcal{S}) = 11k + 10(p - k)$ for some $k \in \{1, \ldots, p\}$. Thus, it means that there exist $i_1, \ldots, i_k \in \{1, \ldots, p\}$, $i_1 < \ldots < i_k$, such that, for every $i \in \{i_1, \ldots, i_k\}$, then $S_i = \{u^1_i\} \cup N_G(u^1_i)$ and $\sum_{e \in E(G[S_i])} w_e = 11$. We deduce that $N_G(u^1_i) \cap N_G(u^1_{i'}) = \emptyset$ for any $i, i' \in \{i_1, \ldots, i_k\}$, $i \neq i'$. Thus, by construction of G, we finally obtain that $\mathcal{C} = \{Y_{i_1}, \ldots, Y_{i_k}\}$ is a set packing for the instance $\mathcal{I}_{sp}$ of size $|\mathcal{C}| = k$. □

We are now able to prove the theorem.

Proof of Theorem D.2. First, the reduction (Definition D.4) can be clearly done in polynomial time. Finally, Lemmas D.1 and D.2 prove that any instance $\mathcal{I}_{sp}$ of set packing can be transformed into an instance G of 2-family-matching, and any solution C of I of cost at least k can be transformed in polynomial time into a

Figure D.2 Illustration of the proof of Theorem D.2. See details in the text.



solution $\mathcal{S}$ of G of cost at least $10p+k$. We thus obtain the algorithms f and g of Definition D.3. Now, observe that any instance I of set packing considered in our reduction is such that $|Y_i| = 2$ for every $i \in \{1, \ldots, t\}$, and every element x_j belongs to at most three sets. This implies that any optimal solution $\mathcal{C}$ of I is such that $|\mathcal{C}| \geq \frac{p}{7}$. Indeed, the number of elements which belong to a set of $\mathcal{C}$ is exactly $3|\mathcal{C}|$, and any set must intersect one of these vertices. However, each of these vertices can belong to at most 2 sets outside $\mathcal{C}$. Hence $p \geq 7|\mathcal{C}|$.

These two arguments already prove item 1 of Definition D.3. We then prove item 2 of Definition D.3 with $\beta = 1$. Given a solution $\mathcal{C}$ of I obtained from a solution $\mathcal{S}$ of G, we have to show that $OPT_{SP}(I) - |\mathcal{C}| \leq \Phi_D(G) - |\mathcal{S}|$. However we have $\Phi_D(G) = OPT_{SP}(I) + 10p$, hence the required inequality is actually equivalent to $|\mathcal{C}| \geq |\mathcal{S}| - 10p$ which is true by the previous lemmas.

To conclude the proof, we show how to generalize the result for any $D \geq 3$. First, we have a path composed of D edges (each of weight 5) from every node of U_1, instead of a path of 2 edges (each of weight 5). Say otherwise, we remove U_2 and U_2', and we add such a path from each node of U_1. The reduction can be done in polynomial time. Furthermore, there is a solution $\mathcal{C}$ for the instance $\mathcal{I}_{sp}$ of set packing problem such that $|\mathcal{C}| \geq k$ if and only if there is a solution $\mathcal{S}$ for the D-family-matching problem for G such that $\Phi(\mathcal{S}) \geq 5Dp + k$. As previously, this implies an L-reduction for the problem, since D is a fixed constant. □

To illustrate the proof of Theorem D.2, consider the instance $\mathcal{I}_{sp}$ of set packing problem, where $X =$

$\{x_1, \ldots, x_9\}$, a family $Y = \{Y_1, \ldots, Y_7\}$ of subsets of X such that $Y_1 = \{x_1, x_2, x_3\}$, $Y_2 = \{x_2, x_3, x_4\}$, $Y_3 = \{x_4, x_5, x_9\}$, $Y_4 = \{x_3, x_8, x_9\}$, $Y_5 = \{x_6, x_7, x_8\}$, $Y_6 = \{x_1, x_5, x_8\}$, $Y_7 = \{x_2, x_6, x_7\}$. Note that $p = 7$. The graph G depicted in Figure D.2 is the graph obtained from Definition D.4. There is a set packing $\mathcal{C} = \{Y_1, Y_3, Y_5\}$ of size $k = 3$ and there is a 2-family-matching $\mathcal{S}$ such that $\Phi(\mathcal{S}) = 10p + k = 73$ (depicted in red in Figure D.2).

Part II: D = 2, Δ = 3, unary weights

We prove the following:

Theorem. D.5. *The 2-family-matching problem is APX-hard in graphs of maximum degree 3 with unary weights.*

Proof. We reduce from the **Induced Matching Problem**, which takes as input a graph $G = (V, E)$ and an integer k, and asks whether there exists a set M of at least k edges such that no two edges of M are joined by an edge. Such a set M is called an *induced matching*. This problem is APX-complete on graphs on maximum degree 3 [DDL13].

Let $G = (V, E)$ be a graph of maximum degree 3, with $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We construct a bipartite graph G' composed of the vertex set $V' \cup E'$, where $V' = \{v'_1, \ldots, v'_n\}$ and $E' = \{e'_1, \cdots e'_m\}$. Then, if $e_j = \{v_a, v_b\}$ is an edge of G, add to G' the edges $\{v'_a, e'_j\}$ and , $\{v'_b, e'_j\}$ and give them weight 1. We now prove that G contains an induced matching of size k if and only if G' contains a 2-family-matching of weight $m + k$.

$\Rightarrow$ Let $M \subseteq E$ be an induced matching of G of size at least k. W.l.o.g., assume that $M = \{e_1, \ldots e_k\}$. Construct, for each $j \in \{1, \ldots, k\}$, the cluster $S_j^M = \{v'_a, v'_b, e'_j\}$, where a, b are such that $e_j = \{v_a, v_b\}$. Clearly S_j^M induces a graph of diameter 2. More precisely, it induces a path of length 2, its weight is thus 2. Moreover, there are k such clusters. Now, for every edge $e_j \in E \setminus M$, let i_j be such that $v_{i_j} \in e_j$ and v_{i_j} does not belong to any edge from M, chosen arbitrarily if several choices are possible.

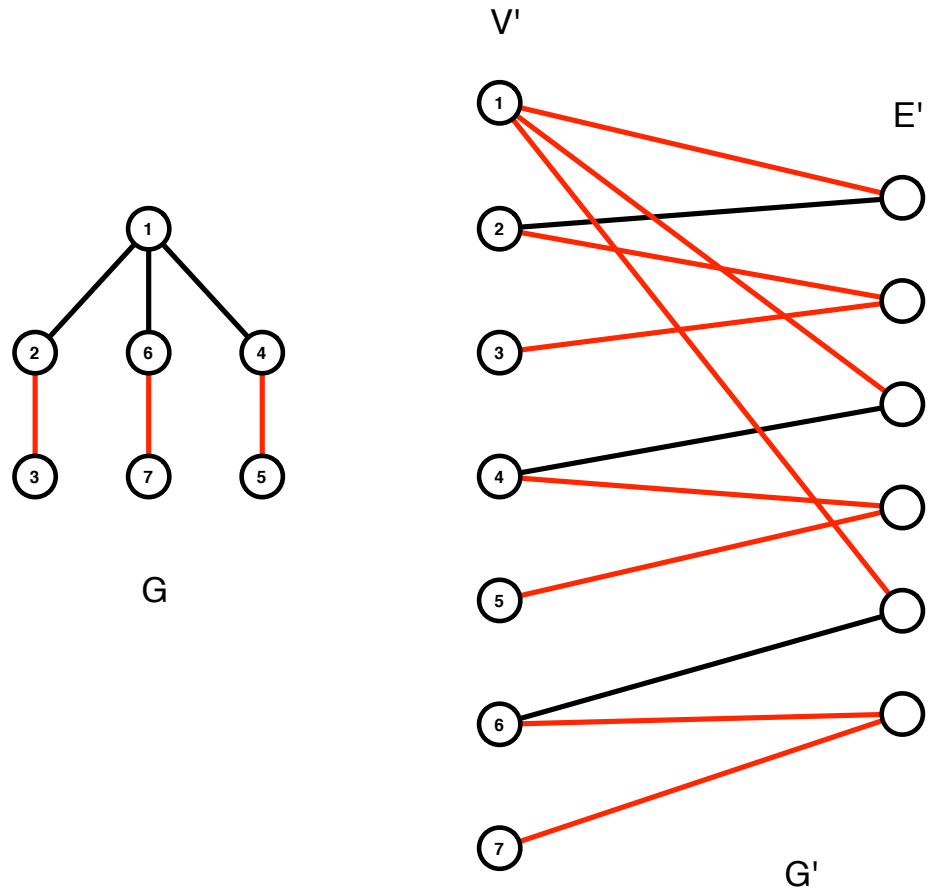
Notice that v_{i_j} is well defined, since M is an induced matching. Let $\mathcal{I} = \{i_j | e_j \in E \setminus M\}$. For every $i \in \mathcal{I}$, construct the cluster $S_i^O = \{v'_i\} \cup \bigcup_{j|i_j=i} \{e'_j\}$. Here again, one can check that the diameter of the graph induced by S_i^O is 2, since it is a star whose center is v'_i, with either one, two or three branches, in which case its weight is respectively 1, 2 or 3. Moreover, the total weight of these clusters is $m - k$. Alltogether, we obtain a 2-family-matching of weight $2k + m - k = m + k$, as desired.

$\Leftarrow$ Conversely, let $\mathcal{S}$ be a solution of weight at least $m + k$. Observe first that since the vertices of E' are of degree 2, and none of them have the same neighborhood, each cluster must induce a star, centered at either a vertex from V' or E'. Let M be the set of all $j \in \{1, \ldots, m\}$ such that e'_j is in the same cluster as v'_a and v'_b, where $e_j = \{v_a, v_b\}$. Now, we show that we can modify the solution (without decreasing its weight) so that edges of index in M form an induced matching: suppose there exist $j, j' \in M$, $j \neq j'$, $r \notin M$ such that $e_r = \{v_a, v_b\}$ with $v_a \in e_j$ and $v_b \in e_{j'}$ (*i.e.* edges e_j and $e_{j'}$ are incident). Hence, e'_r is not contained in any cluster. We remove v'_a from the cluster which contains e'_j, and create a cluster $\{v'_a, e'_r\}$. Applying iteratively this modification we obtain a solution which is still a 2-family-matching of weight at least $k + m$. Moreover, edges whose index are in M form an induced matching. Finally, the number of edges contained in a cluster of $\mathcal{S}$ is at most $2|M| + |E \setminus M| = m + |M|$ and at least $m + k$, thus $|M| \geq k$.

In order to prove that this is indeed an L-reduction, it remains to show that any maximum induced matching M is of size at least αm for some fixed $\alpha > 0$. This is indeed the case, since the input graph has maximum degree 3. Hence, if M denotes a maximum induced matching and $V(M)$ denotes the vertices of M, observe that any edge of G is either (i) in M, (ii) not in M but induced by $V(M)$, (iii) incident to some vertex of $N(V(M))$, the neighbors of $V(M)$. Since $|V(M)| = 2|M|$, we have $|N(V(M))| \leq 6M$, and thus the total number of vertices is bounded above by $8M$, which gives a linear upper bound on m, as desired. □

To illustrate the proof of Theorem D.5 consider the instance of Induced Matching problem depicted in Figure D.3 (left). Figure D.3 (right) represents the intersection graph constructed from it. There is an induced matching of G of size 3 and there is a 2-family-matching composed of 4 sets with total weight equal to 9.

Figure D.3 Illustration of the proof of Theorem D.5. See details in the text.



Part III: D ¿ 2, $\Delta = 3$

We prove the following result:

Theorem. D.6. *For any $D \geq 3$, the D-family-matching problem is APX-hard for graphs of maximum degree 3.*

Proof. Similarly to the previous case, we reduce from the **Maximum Induced Matching Problem**.

Let $G = (V, E)$ be a graph of maximum degree 3, with $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We construct a bipartite graph G' composed of the vertex set $V^1 \cup E^1 \cup \cdots \cup E^{D-1} \cup V^2$, where $V^i = \{v_1^i, \ldots, v_n^i\}$ and $E^j = \{e_1^j, \cdots e_m^j\}$ for $i \in \{1, 2\}$, $j \in \{1, \ldots, D-1\}$. Then, if $e_j = \{v_a, v_b\}$ is an edge of G, add to G' the edges $\{v_a^1, e_j^1\}$, $\{v_b^1, e_j^1\}$, $\{e_j^{D-1}, v_a^2\}$, $\{e_j^{D-1}, v_b^2\}$ and give them weight 1. Finally, for every $j \in \{1, \ldots, m\}$ and $\ell \in \{1, \ldots, D-2\}$, add the edge $\{e_j^\ell, e_j^{\ell+1}\}$ and give it weight $W = 4m + 1$. Clearly G' is a bipartite graph of maximum degree 3. We now prove that G contains an induced matching of size k if and only if G' contains a D-family-matching of weight $(D-2)Wm + 2(m+k)$.

$\Rightarrow$ Let $M \subseteq E$ be an induced matching of G of size at least k. W.l.o.g., assume that $M = \{e_1, \ldots e_k\}$. Construct, for each $j \in \{1, \ldots, k\}$, the cluster $S_j^M = \{v_a^1, v_b^1, e_j^1, \ldots, e_j^{D-1}, v_a^2, v_b^2\}$, where a, b are such that $e_j = \{v_a, v_b\}$. Clearly S_j^M induces a graph of diameter D. Moreover, its weight is $(D-2)W + 4$. Now, for every edge $e_j \in E \setminus M$, let i_j be such that $v_{i_j} \in e_j$ and v_{i_j} does not belong to any edge from M, chosen arbitrarily if several choices are possible.

Notice that v_{i_j} is well defined, since M is an induced matching. Let $\mathcal{I} = \{i_j | e_j \in E \setminus M\}$. For every $i \in \mathcal{I}$,

construct the cluster $S_i^O = \{v_i^1, v_i^2\} \cup \bigcup_{j|i_j=i} \{e_j^1, \ldots, e_j^{D-1}\}$. Here again, one can check that the diameter of the graph induced by S_i^O is D, since it is either a path on $D+1$ vertices, or two or three paths of length $D+1$ whose respective endpoints have been identified. Finally, the sum of the weights of these clusters is $|E \setminus M|\,((D-2)W+2)$. Hence, the total weight of this D-family-matching is $(D-2)Wm + 2(m+k)$, as desired.

$\Leftarrow$ Conversely, let $\mathcal{S}$ be a solution of weight at least $(D-2)Wm + 2(m+k)$. Because of the value of W, it holds that for every $j \in \{1, \ldots, m\}$ and every $\ell \in \{1, \ldots, D-2\}$, the edge $\{e_j^\ell, e_j^{\ell+1}\}$ belongs to some cluster of $\mathcal{S}$. Observe that the weight of these edges is $(D-2)Wm$, and the weight of all remaining edges, *i.e.* edges between V^1 and E^1, and edges between E^{D-1} and V^2, are 1. Hence, we may assume, w.l.o.g., that $\mathcal{S}$ contains at least $m+k$ edges among those between V^1 and E^1. Let M be the set of all $j \in \{1, \ldots, m\}$ such that e_j^1 is in the same cluster as v_a^1 and v_b^1, where $e_j = \{v_a, v_b\}$. Observe that there cannot be $j, j' \in M$, $j \neq j'$, such that e_j^1 and $e_{j'}^1$ are in the same cluster, since the diameter of such a cluster would be at least $D+1$. Now, we show that we can modify the solution (without decreasing its weight) so that edges of index in M form an induced matching: suppose there exist $j, j' \in M$, $j \neq j'$, $r \notin M$ such that $e_r = \{v_a, v_b\}$ with $v_a \in e_j$ and $v_b \in e_{j'}$ (*i.e.* edges e_j and $e_{j'}$ are incident). We move v_a^1 from the cluster which contains e_j^1 to the cluster which contains e_r^1. Let us call $\mathcal{S}'$ the obtained solution. We have the following:

- $\mathcal{S}'$ is of same weight as $\mathcal{S}$;
- $\mathcal{S}'$ is a D-family-matching: firstly, v_a^1 is a vertex of degree 1 in the graph induced by its cluster in $\mathcal{S}$, so the diameter of the cluster containing e_j^1 is not greater in $\mathcal{S}'$. Secondly, we added a vertex of degree 1 in the cluster containing e_r^1. If this cluster has diameter at least $D+1$ in $\mathcal{S}'$, it implies that in $\mathcal{S}$, there exists $r' \neq r$ such that $e_{r'}^{D-1}$ and e_r^1 are in the same cluster, but since e_r^1 is only adjacent to e_r^2 in its cluster, the shortest path between e_r^1 and $e_{r'}^1$ is of length at least $2D-3$, a contradiction.

Applying this modification leads to a solution $\mathcal{S}$ such that M represents an induced matching in G. Finally, the number of edges contained in a cluster of $\mathcal{S}$ among those between V^1 and E^1 is at most $2|M| + |E \setminus M| = m + |M|$, and at least $m+k$, thus $|M| \geq k$. As previously, m can be bounded above by a linear function of the size of a maximum induced matching, which proves that the reduction is an L-reduction. □

To illustrate the proof of Theorem D.6, consider the instance of Induced Matching problem depicted in Figure D.4 (left). Figure D.4 (right) represents the intersection graph constructed from it for $D = 4$. There is an induced matching of G of size 3 and there is a 2-family-matching composed of 4 sets with total weight equal to $18 + 12W = 318$.

D.4.2 Unbounded ratio between scores by increasing the diameter by one

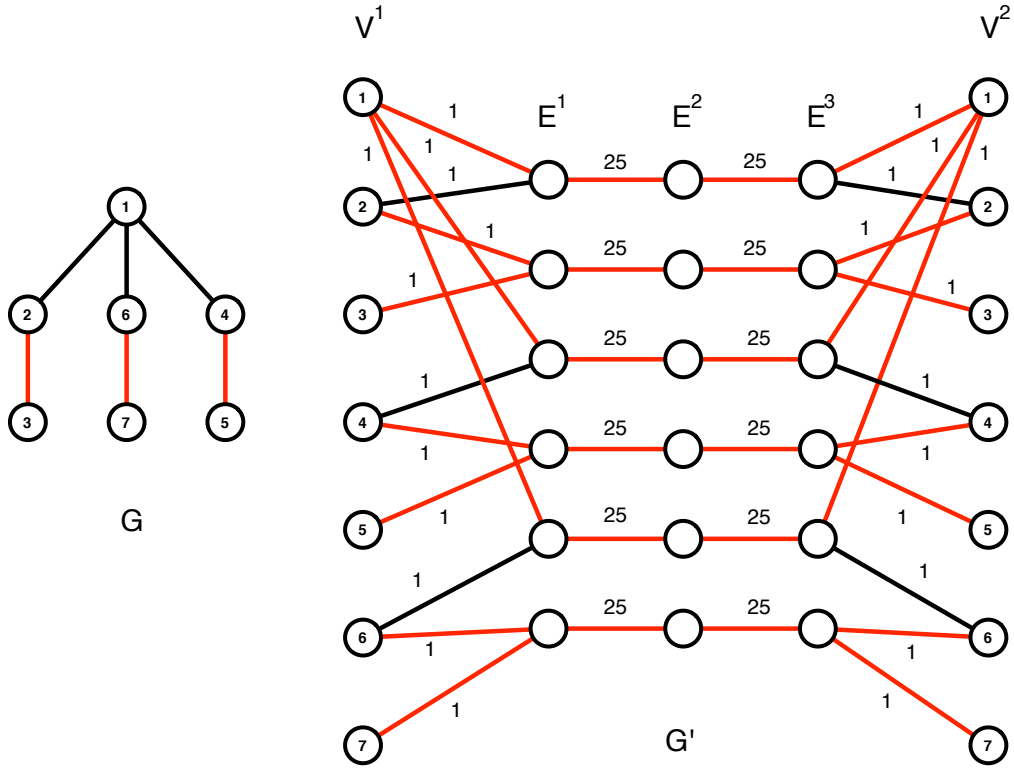
Proof of Lemma 6.4. Let $t = n - 1$. Let $Z = \{z_1, \ldots, z_t\}$, $F = \{F_1\}$, and $F' = \{F'_1, \ldots, F'_t\}$ be an instance of the D-family-matching problem, where $F_1 = \{z_1, \ldots, z_t\}$ and $F'_i = \{z_i\}$ for every $i \in \{1, \ldots, t\}$. The intersection graph $G = (U, U', E, w)$ is such that $U = \{u_1\}$, $U' = \{u'_1, \ldots, u'_t\}$, $E = \{\{u_1, u'_i\} \mid 1 \leq i \leq t\}$, and $w_e = 1$ for every $e \in E$. We first prove that any solution $\mathcal{S}^{D=1}$ for the 1-family-matching problem is such that $\Phi(\mathcal{S}^{D=1}) \leq 1$. Indeed, every sub-graph of G with diameter at most 1 is composed of at most 1 edge. Furthermore, if $\mathcal{S}^{D=1}$ contains a set S that induces a sub-graph composed of one edge, then all others sets induce sub-graphs that do not contain any edge (observe that every edge contains the central node u_1 of the star graph G). Otherwise the family $\mathcal{S}^{D=1}$ does not satisfy the property that the subsets are disjoint. Thus, $\Phi(\mathcal{S}^{D=1}) \leq 1$. Finally, let $\mathcal{S}^{D=2} = \{U \cup U'\}$. The graph induced by $U \cup U'$ is the graph G that has diameter 2 and $\Phi(\mathcal{S}^{D=2}) = t = n - 1$. □

D.4.3 Optimizing first the score of a single set can be arbitrarily bad

Proof of Lemma 6.3. Consider the intersection graph $G = (U, U', E, w)$ constructed as follows.

- Set $U = \{u_1, \ldots, u_\lambda\}$.

Figure D.4 Illustration of the proof of Theorem D.6. See details in the text.



- Set $U' = \{u'_c\} \cup U'^1 \cup \ldots \cup U'^\lambda$, where $U'^i = \{u'^i_1, \ldots, u'^i_{\lambda-1}\}$ for every $i \in \{1, \ldots, \lambda\}$.
- Set $E = E^c \cup E^1 \cup \ldots \cup E^\lambda$, where
  - $E^c = \{\{u'_c, u_i\} \mid 1 \leq i \leq \lambda\}$
  - and $E^i = \{\{u_i, u'^i_j\} \mid 1 \leq j \leq \lambda - 1\}$ for every $i \in \{1, \ldots, n\}$.
- Set $w_e = 1$ for every $e \in E$.

Observe that the graph G is bipartite.

We now prove that the sub-graph of G with diameter at most 2 and that has the maximum number of edges is the graph $G[\{u'_c, u_i, \ldots, u_\lambda\}]$ composed of λ edges that is induced by the set of nodes $\{u'_c, u_i, \ldots, u_\lambda\}$. Indeed, suppose that node u'_c is not in such a graph. Then, if we remove u'_c from G, we obtain λ disjoint stars each composed of $\lambda - 1$ edges. Thus, since $w_e = 1$ for every $e \in E$, then we get that the graph $G[\{u'_c, u_i, \ldots, u_\lambda\}]$ induced by $\{u'_c, u_i, \ldots, u_\lambda\}$ maximizes the sum of the weights. Now, if we remove such a set from G, we get disjoint isolated nodes (that is each node has degree 0). We get that $\Phi(\mathcal{S}^{max}) = \lambda$.

We finally prove that there exists a 2-family-matching $\mathcal{S}$ for G such that $\Phi(\mathcal{S}) \geq \lambda(\lambda - 2)$. Let $\mathcal{S} = \{S_1, \ldots, S_\lambda\}$ be such that $S_i = \{u_i\} \cup U'^i$ for every $i \in \{1, \ldots, \lambda\}$. Observe that $S_i \cap S_j = \emptyset$ for every $i, j \in \{1, \ldots, \lambda\}$, $i \neq j$. Furthermore, the graph $G[S_i]$ is a star and so has diameter 2. Thus, $\mathcal{S}$ is a $(\lambda, 2)$-family-matching for G. The number of edges of $G[S_i]$ is $|E(G[S_i])| = \lambda - 2$ for every $i \in \{1, \ldots, \lambda\}$. Since $w_e = 1$ for every $e \in E$, we finally get that

$$\Phi(\mathcal{S}^{max}) = \lambda, \quad \Phi(\mathcal{S}) \geq \lambda(\lambda - 2), \text{and} \quad \frac{\Phi(\mathcal{S})}{\Phi(\mathcal{S}^{max})} = \lambda - 2.$$

This concludes the proof of Lemma 6.3. □

D.5 Appendix - Polynomial time dynamic programming algorithms for some classes

D.5.1 The D-family-matching problem for paths

In this section, we illustrate the result of Theorem 6.2 by considering paths. Consider an intersection path $G = (V, E, w)$. By Theorem 6.2, given $D \geq 1$, there is an $O(D^2 n)$-time complexity algorithm for the D-family-matching problem because $\Delta = 2$. We prove in Lemma D.3 a better time complexity algorithm for the D-family-matching problem. Indeed, the time complexity is $O(Dn)$.

Lemma. D.3 (Computation of $\Phi_D(()G)$ for paths)**.** *Let $D \in \mathbb{N}^+$. Consider any intersection path $G = (V, E, w)$. Then, there exists an $O(Dn)$-time complexity algorithm for the D-family-matching problem for G.*

Proof of Lemma D.3. Let $V = \{v_1, \ldots, v_n\}$. Let $E = \{\{v_j, v_{j+1}\} \mid 1 \leq j \leq n-1\}$. We define the function Ψ_D as follows. For every $t \in \{1, \ldots, n\}$ and every $i \in \{\max(1, t-D), \ldots, t+1\}$, then $\Psi_D(v_t, i)$ is the score of an optimal solution $\mathcal{S}$ of the D-family-matching problem, for the sub-path induced by the set of nodes $\{v_1, \ldots, v_t\}$, such that $\{v_i, \ldots, v_t\} \in \mathcal{S}$. The case $i = t+1$ means that v_t does not belong to any set. Note that we consider $i \geq \max(1, t-D)$ because, otherwise we would not have an admissible solution (because of the diameter constraint). First of all, $\Psi_D(v_1, 1) = \Psi_D(v_1, 2) = 0$.

Let $t \in \{1, \ldots, n-1\}$. Suppose we have computed $\Psi_D(v_{t'}, i)$ for every $t' \in \{1, \ldots, t\}$ and every $i \in \{\max(1, t-D), \ldots, t'+1\}$. We prove that we can compute $\Psi_D(v_{t'+1}, i)$ for every $i \in \{\max(1, t-D), \ldots, t'+1\}$ in $O(D)$-time. There are two different cases (corresponding to the two following claims).

Claim D.7. *For every $i \in \{\max(1, t+1-D), \ldots, t\}$, then*

$$\Psi_D(v_{t+1}, i) = w_{v_t, v_{t+1}} + \Psi_D(v_t, i).$$

Proof of Claim D.7. The set of nodes $\{v_i, \ldots, v_{t+1}\}$, $\max(1, t+1-D) \leq i \leq t$, must be a set of the solution. Thus, we have to consider the optimal solution for the D-family-matching problem, for the sub-path induced by the set of nodes $\{v_i, \ldots, v_t\}$, such that $\{v_i, \ldots, v_t\}$ is a set of this solution. We then modify this solution by adding node v_{t+1} in the last set, and we obtain the optimal solution for the D-family-matching problem, for the sub-path induced by the set of nodes $\{v_i, \ldots, v_t\}$, such that $\{v_1, \ldots, v_{t+1}\}$ is a set of this solution. □

Claim D.8.

$$\Psi_D(v_{t+1}, t+1) = \Psi_D(v_{t+1}, t+2) = \max_{i \in \{\max(1, t-D), \ldots, t+1\}} \Psi_D(v_t, i).$$

Proof of Claim D.8. We first prove the result for $\Psi_D(v_{t+1}, t+1)$. Any solution must contain the set $\{v_{t+1}\}$. Thus, we have to consider an optimal solution for the D-family-matching problem for the sub-path induced by the set of nodes $\{v_i, \ldots, v_t\}$.

We now prove the result for $\Psi_D(v_{t+1}, t+2)$. Since node $\{v_{t+1}\}$ does not belong to any set, then we have to consider again an optimal solution for the D-family-matching problem for the sub-path induced by the set of nodes $\{v_i, \ldots, v_t\}$. □

For every $t \in \{1, \ldots, n\}$, we address the time complexity of computing Ψ as follows. For each claim, the time complexity of the computation of Ψ is $O(D)$. We get that the time complexity of the dynamic programming algorithm is $O(nD)$.

To conclude the proof of Lemma D.3, when we have computed $\Psi_D(v_n, i)$ for every $i \in \{\max(1, n-D), \ldots, n+1\}$, then we can deduce an optimal solution $\mathcal{S}$ and the optimal value for the D-family-matching problem for G. Indeed,

$$\Phi_D(()G) = \max_{i \in \{\max(1, n-D), \ldots, n+1\}} \Psi_D(v_n, i).$$

Recall that $n+1$ means that node v_n does not belong to any set of the solution. □

D.5.2 The D-family-matching problem for cycles

We now deduce in Corollary D.1 an efficient algorithm for the D-family-matching problem when G is an even cycle.

Corollary. D.1 (Computation of $\Phi_D(()G)$ for cycles)**.** *Let $D \in \mathbb{N}^+$. Consider any intersection graph $G = (V, E, w)$ that is an even cycle. Then, there exists an $O(D^2 n)$-time complexity algorithm for the D-family-matching problem for G.*

Indeed, we have

$$\Phi_D(()G) = \max_{H \in H(G,v)} (\Psi_D(G_H) + \sum_{e \in E_H} w_e).$$

To conclude Section D.5, we address in Corollary D.2 the results of Theorem 6.5 in terms of the original problem (that is for the equivalent definition proved in Section D.3).

Corollary. D.2. *Let $D \in \mathbb{N}^+$. Consider any instance of the D-family-matching problem such that:*

- *for every $i \in \{1, \ldots, r\}$, there exist $j_1, j_2 \in \{1, \ldots, r'\}$ such that $F_i \cap F'_j = \emptyset$ for any $j \in \{1, \ldots, r'\} \setminus \{j_1, j_2\}$.*
- *for every $j \in 1, \ldots, r'\}$, there exist $i_1, i_2 \in \{1, \ldots, r\}$ such that $F'_j \cap F_i = \emptyset$ for any $i \in \{1, \ldots, r\} \setminus \{i_1, i_2\}$.*

Then, there exists an $O((r + r')D^2)$-time complexity algorithm for the D-family-matching problem.

Say otherwise, Corollary D.2 shows that there is a polynomial time algorithm for the D-family-matching problem if any set in $F \cup F'$ has a non-empty intersection with at most two other sets of $F \cup F'$.

D.6 Appendix - Generic approach based on spanning trees

Let us first introduce some notations. For every $v \in V$, let $H(G, v)$ be the set of all different sub-graphs of G that contain v and of diameter at most D. Let $H(G) = \cup_{v \in V} H(G, v)$. We define $h(G, v) = |H(G, v)|$ for every $v \in V$ and $h(G) = \max_{v \in V} h(G, v)$. Let T_r be any spanning tree of G rooted at node $r \in V$. For every $v \in V$, we define $H(G, T_r, v)$ as the set of all $H \in H(G, v)$ such that the graph induced by the set of nodes $V(H) \cap V(T_v)$ is a (connected) sub-tree rooted at v. Let $H(G, T_r) = \cup_{v \in V} H(G, T_r, v)$. We define $h(G, T_r, v) = |H(G, T_r, v)|$ for every $v \in V$ and $h(G, T_r) = \max_{v \in V} h(G, T_r, v)$. Furthermore, let $\mathcal{T}(G)$ be the set of all different rooted spanning trees of G.

D.6.1 Dynamic programming algorithms under spanning tree constraint

Proof of Lemma 6.6. Consider the tree T rooted at any node $r \in V$ such that r has not degree Δ. Such a node always exist if T contains at least three nodes. We call this rooted tree T_r. We define the function Ψ_D as follows. For every $v \in V$ and every $H \in H(G, T_r, v)$, then $\Psi_D(v, H)$ is the score of an optimal solution $\mathcal{S}$ for the D-family-matching problem, for the graph $G[V(t_v)]$ induced by the set of nodes $V(T_v)$, constrained by T_v, and such that $V(H) \in \mathcal{S}$. We allow H to be the empty graph $(\emptyset, \emptyset)$.

By convention, if there is no admissible solution, we set $\Psi_D(v, H) = -\infty$.

First of all, for every leaf $v \in V$ of T_r, then

- $\Psi_D(v, H) = 0$ if $H \in \{(\emptyset, \emptyset), (\{v\}, \emptyset)\}$,
- $\Psi_D(v, H) = -\infty$ if $H \in \{(\emptyset, \emptyset), (\{v\}, \emptyset)\}$.

A leaf is a node of degree one and different than the root r.

Let $v \in V$ be any node that is not a leaf. Let $N(v) = \{v_1, \ldots, v_q\}$ be the set of $q \geq 1$ neighbors of v in T_v. Suppose we have computed $\Psi_D(v_j, H)$ for every $j \in \{1, \ldots, q\}$ and every $H \in H(G, T_r, v_j)$. We prove that we can compute $\Psi_D(v, H)$ for every $H \in H(G, T_r, v)$. There are three different cases (corresponding to the three following claims).

Claim D.9.

$$\Psi_D(v,(\emptyset,\emptyset)) = \max_{(H_1,\ldots,H_q)\in\mathcal{H}} \sum_{j=1}^{q} \Psi_D(v_j,H_j),$$

where $\mathcal{H}$ is the set of all vectors $(H_1,\ldots,H_q)$ such that $H_i \in H(G,T_r,v_i)$ for every $i \in \{1,\ldots,q\}$.

Proof of Claim D.9. We consider here an optimal solution for the D-family-matching problem for the graph $G[V(t_v)]$ induced by the set of nodes $V(T_v)$, constrained by T_v, and such that v does not belong to any set. Thus, $\Psi_D(v,(\emptyset,\emptyset))$ consists in choosing the set H_j that contains v_j (possibly empty) for every $j \in \{1,\ldots,q\}$, such that the score is maximal. □

Claim D.10.

$$\Psi_D(v,(\{v\},\emptyset)) = \max_{(H_1,\ldots,H_q)\in\mathcal{H}} \sum_{j=1}^{q} \Psi_D(v_j,H_j),$$

where $\mathcal{H}$ is the set of all vectors $(H_1,\ldots,H_q)$ such that $H_i \in H(G,T_r,v_i)$ for every $i \in \{1,\ldots,q\}$.

Proof of Claim D.10. This proof is similar to the proof of Claim D.9. Indeed, we consider an optimal solution $\mathcal{S}$ for the D-family-matching problem for the sub-graph induced by the set of nodes $V(T_v)$, constrained by T_v, and such that $\{v\} \in \mathcal{S}$. Thus, $\Psi_D(v,(\{v\},\emptyset))$ consists in choosing the set H_j that contains v_j (possibly empty) for every $j \in \{1,\ldots,q\}$, such that the score is maximal. □

Claim D.11. *Let $H \in H(G,T_r,v)$ be any sub-tree. Without loss of generality, assume that, for some q', $V(H) \cap V(T_{v_j}) \neq \emptyset$ for every $j \in \{1,\ldots,q'\}$, and $V(H) \cap V(T_{v_j}) = \emptyset$ for every $j \in \{q'+1,\ldots,q\}$. Let H_{v_j} be the intersection between H and the sub-tree T_{v_j}, that is $V(H_{v_j}) = V(H) \cap V(T_{v_j})$. Then*

$$\Psi_D(v,H) =$$

$$\sum_{e'\in E(H'_v)} w_{e'} \; + \sum_{j=1}^{q'} (\Psi_D(v_j,H_{v_j}) - \sum_{e'\in E(H_{v_j})} w_{e'}) \; + \max_{(H_{q'+1},\ldots,H_q)\in\mathcal{H}'} \sum_{j=q'+1}^{q} \Psi_D(v_j,H_j),$$

where $\mathcal{H}$ is the set of all vectors $(H_{q'+1},\ldots,H_q)$ such that $H_i \in H(G,T_r,v_i)$ for every $i \in \{q'+1,\ldots,q\}$.

Proof of Claim D.11. We consider an optimal solution $\mathcal{S}$ for the D-family-matching problem for the graph $G[V(t_v)]$ induced by the set of nodes $V(T_v)$, constrained by T_v, and such that $V(H) \in \mathcal{S}$. The sub-tree H contains v and q' sub-trees $H^{v_1},\ldots,H^{v_{q'}}$ rooted at $v_1,\ldots,v_{q'}$, respectively. Thus, $\Psi_D(v,H)$ consists in choosing, for every $j \in \{q'+1,\ldots,q\}$, the set H_j that contains v_j (possibly empty) in order to maximize the value of the solution. Note that H_j must be a graph that belongs to $H(G,T_r,v_j)$ by definition of the problem constrained by a tree. □

For every $v \in V$, we address the time complexity of computing Ψ as follows. The time complexity of the computation done in Claim D.9 is $O(\prod_{j=1}^{q} h(G,T_r,v_i))$. The time complexity of the computation done in Claim D.10 is $O(\prod_{j=1}^{q} h(G,T_r,v_i))$. The time complexity of the computation done in Claim D.11 is $O(h(G,T_r,v)(q' + |E(H_{v_j})| + \prod_{j=q'+1}^{q} h(G,T_r,v_i)))$. Since $h(G,T_r,v) \leq h(G,T_r)$ for every $v \in V$ and $q \leq \Delta - 1$ by the choice of the root of T, then we get that the time complexity of the algorithm is $O(h(G,T_r)^{\Delta} n) = O(2^{D\Delta\log_2(\Delta)} n)$ because $h(G,T) = O(2^{\Delta^D})$.

To conclude the proof of Lemma 6.6, when we have computed $\Psi_D(r,H)$ for every $H \in H(G,T_r,r)$, we can deduce an optimal solution $\mathcal{S}$ and the optimal value of the D-family-matching problem for G constrained by T. Indeed,

$$\Phi_D(G,T_r) = \max_{H\in H(G,T_r,r)} \Psi_D(r,H).$$

Note that H can be empty (in that case r does not belong to any set of $\mathcal{S}$). □

D.6.2 Algorithms based on spanning trees

Proof of Lemma 6.5. For some $k \geq 1$, consider an optimal solution $\mathcal{S} = \{S_1, \ldots, S_k\}$ for the D-family-matching problem for G. For every $i \in \{1, \ldots, k\}$, let T_i be any spanning tree of $G[S_i]$. Let T be any rooted spanning tree of G such that $E(T_i) \subseteq E(T)$ for every $i \in \{1, \ldots, k\}$. By construction of T, $\mathcal{S}$ is an admissible solution for the D-family-matching problem for G constrained by T. Thus, $\Phi_D(G, T) = \Phi_D(G)$. □

Corollary. D.3. *Given any positive integer $D \geq 1$ and any intersection graph G, Algorithm 1 returns $\Phi_D(G)$, that is an optimal solution for the D-family-matching problem for G, if:*

- $\Pi(\mathcal{M}) \Leftrightarrow |\mathcal{M}| = |\mathcal{T}(G)|$,
- $\mathcal{R}(G, \lambda) = T^\lambda$, *where* $\mathcal{T}(G) = \{T^1, \ldots, T^{|\mathcal{T}(G)|}\}$,
- *and Algorithm* $\mathcal{A}(G, T^\lambda, D)$ *returns* $\Phi_D(G, T^\lambda)$ *(Lemma 6.6).*

Furthermore, the time complexity of Algorithm 1 is $O(|\mathcal{T}(G)| \max_{T_r \in \mathcal{T}(G)} h(G, T_r)^\Delta n)$.

Lemma. D.4. *Let G be any intersection graph. Then, there exists a rooted spanning tree T of G such that $\Phi_2(G) \leq 2\Delta\Phi_2(T)$.*

Proof of Lemma D.4. For some $k \geq 1$, consider an optimal solution $\mathcal{S} = \{S_1, \ldots, S_k\}$ for the 2-family-matching problem for G. For every $i \in \{1, \ldots, k\}$, let T_i be a maximum spanning tree of $G[S_i]$. Let T be any rooted spanning tree of G such that $E(T_i) \subseteq E(T)$ for every $i \in \{1, \ldots, k\}$. For every $i \in \{1, \ldots, k\}$, we have $\Delta \sum_{e \in E(T_i)} w_e \geq \sum_{e \in E(G[S_i])} w_e$. Indeed, since $D = 2$, $G[S_i]$ is necessarily a complete bipartite graph and its number of nodes is at most 2Δ. It is sufficient to select the maximum star as T_i in order to get the inequality. Thus, by construction of T, the algorithm returns at least the desired score. □

Corollary. D.4. *Given any intersection graph G, Algorithm 1 returns a 2Δ-approximation for the 2-family-matching problem for G if:*

- $\Pi(\mathcal{M}) \Leftrightarrow |\mathcal{M}| = |\mathcal{T}(G)|$,
- $\mathcal{R}(G, \lambda) = T^\lambda$, *where* $\mathcal{T}(G) = \{T^1, \ldots, T^{|\mathcal{T}(G)|}\}$,
- *and Algorithm* $\mathcal{A}(G, T^\lambda, D)$ *returns* $\Phi_D(T^\lambda)$ *(Theorem 6.2).*

Furthermore, the time complexity of Algorithm 1 is $O(|\mathcal{T}(G)| D^2 \Delta^2 n)$.

D.7 Appendix - On the choice of D

Proof of Theorem 6.6. We first pre-compute $\frac{\tau_w(|I_x|)}{\tau_h(\max_{D, D' \in I_x \cap \mathbb{N}} \Phi_{D'}(G) - \Phi_D(G))}$ for every possible plateau I_x. There are $O(D_G^2)$ such computations. Then, we prove a direct dynamic programming algorithm. For every $D \in \{1, \ldots, D_G\}$, $\rho(D)$ is the score of an optimal solution for the tradeoff-plateau problem for the set $\{\Phi_1(G), \ldots, \Phi_D(G)\}$. We have $\rho(0) = 0$. In general, for every $D \in \{1, \ldots, D_G - 1\}$, we have

$$\rho(D+1) = \min_{i \in \{0, \ldots, D\}} (\rho(i) - \frac{\tau_w(D-i)}{\tau_h(\max_{i, i' \in [i+1, D+1]} \Phi_{i'}(G) - \Phi_i(G))}).$$

Recall that τ_w and τ_h are two increasing functions of one variable such that $\tau_w(y) > 0$ and $\tau_h(y) > 0$ for any y. If $|I_x| = 1$, then $\frac{\tau_w(D-i)}{\tau_h(\max_{i, i' \in I_x \cap \mathbb{N}} \Phi_{i'}(G) - \Phi_i(G))} = 0$. The computation consists of computing an optimal solution for every possible plateau $[i+1, D+1]$ and, among these optimal solutions, calculate an optimal one of score $\rho(D+1)$. For every $D \in \{1, \ldots, D_G - 1\}$, there are $O(D)$ optimal solutions to compute, and each of them can be computed in constant time. Finally, $\rho(D_g)$ is the optimal score and the total time computation is quadratic in the diameter D_G. □

Proof of Lemma 6.7. We prove the result by induction. Clearly, $\rho_{1,\Phi_1(G),\Phi_1(G)}(1) = 0$.

Assume that we have computed $\rho_{y,x^-,x^+}(D)$ for every $D \in \{1,\dots,D'\}$, for every $y \in \{1,\dots,D\}$, for every $x^-, x^+ \in \mathcal{P}_D$ with $x^- \leq x^+$. We prove that the five cases described in Lemma 6.7 allow to compute $\rho_{y,x^-,x^+}(D+1)$ for every $y \in \{1,\dots,D+1\}$ and for every $x^-, x^+ \in \mathcal{P}_{D+1}$ with $x^- \leq x^+$.

- Consider first the case $\Phi_{D+1}(G) \in]x^-, x^+[$. We necessarily have $\rho_{y,x^-,x^+}(D+1) = \rho_{y,x^-,x^+}(D)$ because we cannot start a new plateau since $x^- < \Phi_{D+1}(G) < x^+$.
- Assume that $\Phi_{D+1}(G) = x^-$ and $x^- < x^+$. We cannot start a new plateau because $x^- < x^+$. Thus we have to find the best y plateaus such that the lower bound is at least $\Phi_{D+1}(G) = x^-$ and at most x^+. We get that $\rho_{y,x^-,x^+}(D+1) = \min_{x \in \mathcal{P}_{D+1}, x^- \leq x \leq x^+}(\rho_{y,x,x^+}(D) + x - x^-)$.
- If $\Phi_{D+1}(G) = x^+$ and $x^- < x^+$, then it is similar than the previous case (symmetric case).
- Consider the case $x^- = x^+ = \Phi_{D+1}(G)$. It is possible to continue the current plateau or it is possible to start a new one. In the first case, the score is $\rho_{y,x^-,x^+}(D)$. In the second case, the score is minimum score among all the optimal solutions composed of $y-1$ plateaus, that is $\min_{x,x' \in \mathcal{P}_{D+1}, x \leq x'}(y-1, x, x')$. Thus, $\rho_{y,x^-,x^+}(D+1)$ is the minimum among these two scores.
- If $\Phi_{D+1}(G) < x^-$ or $\Phi_{D+1}(G) > x^+$, then there is no admissible solution and, by convention, we have $\rho_{y,x^-,x^+}(D+1) = \infty$.

□

Proof of Lemma 6.8. We first consider all the cases but the fourth. For every $D \in \{0,\dots,D_G-1\}$, for every $y \in \{1,\dots,D+1\}$, for every $x^-, x^+ \in \mathcal{P}_{D+1}$ with $x^- \leq x^+$, we have to compute $\rho_{y,x^-,x^+}(D+1)$. There are $O(D_G^4)$ such computations. All the cases (but the fourth), can be calculated in $O(D_G)$ time. Thus, we get the $O(D_G^5)$-time complexity.

Now consider the fourth case in which $x^- = x^+$. Thus, for every $D \in \{0,\dots,D_G-1\}$, for every $y \in \{1,\dots,D+1\}$, for every $x^- = x^+ \in \mathcal{P}_{D+1}$, we have to compute $\rho_{y,x^-,x^+}(D+1)$. There are $O(D_G^3)$ such computations. Furthermore, $\rho_{y,x^-,x^+}(D+1) = \min(\rho_{y,x^-,x^+}(D), \min_{x,x' \in \mathcal{P}_{D+1}, x \leq x'}(y-1, x, x'))$ can be computed in $O(D_G^2)$ time. Thus, we get the $O(D_G^5)$-time complexity. □