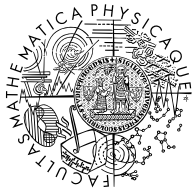
Similarity-based approaches in molecular function discovery

David Hoksza

habilitation thesis



FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Prague, September 2019

Contents

Preface

This habilitation thesis is a compilation of publications from the domain of structural bioinformatics authored or co-authored by David Hoksza. The techniques introduced in the publications were developed between the years 2011 and 2019 mainly at Charles University, Prague. The motivation was to aid molecular function discovery with the focus on i) similarity modeling of RNA structures and ii) protein binding site detection.

The thesis is divided into two parts. Part I contains the commentary to the contributions on which the thesis is based. As the presented research relates to the domain of bioinformatics, we decided to include a primer introducing the main bioinformatics concepts so that the thesis can be easily followed also by readers without any molecular biology background. This introduction, together with the motivation which follows, form Chapter 1. Chapter 2 and 3 outline the particular contributions of the thesis. Part II then consists of the 9 publications in which these contributions were introduced. The list of the publications follows:

1. David Hoksza and Daniel Svozil. Efficient RNA pairwise structure comparison by SETTER method. *Bioinformatics*, 28(14):1858–1864, 2012. DOI: 10.1093/bioinformatics/bts301

2. Petr Čech, Daniel Svozil, and David Hoksza. SETTER: web server for RNA structure comparison. *Nucleic acids research*, 40(W1):W42–W48, 2012. DOI: 10.1093/nar/gks560

3. David Hoksza and Daniel Svozil. Multiple 3D RNA structure superposition using neighbor joining. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(3):520–530, 2014. DOI: 10.1109/TCBB.2014.2351810

4. Petr Čech, David Hoksza, and Daniel Svozil. MultiSETTER: web server for multiple RNA structure comparison. *BMC bioinformatics*, 16(1): 253, 2015. DOI: 10.1186/s12859-015-0696-8

5. Richard Elias and David Hoksza. TRAVeLer: a tool for template-based RNA secondary structure visualization. *BMC bioinformatics*, 18(1):487, 2017. DOI: 10.1186/s12859-017-1885-4

6. Radoslav Krivák and David Hoksza. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *Journal of cheminformatics*, 7(1):12, 2015b. DOI: 10.1186/s13321-015-0059-5

7. Radoslav Krivák and David Hoksza. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10(1):39, 2018. DOI: 10.1186/s13321-018-0285-8

8. L. Jendele, R. Krivak, P. Skoda, M. Novotny, and D. Hoksza. PrankWeb: a web server for ligand binding site prediction and visualization. *Nucleic Acids Res.*, 47(W1):W345–W349, Jul 2019. DOI: 10.1093/nar/gkz424

9. Jan Jelínek, Petr Škoda, and David Hoksza. Utilizing knowledge base of amino acids structural neighborhoods to predict protein-protein interaction sites. *BMC bioinformatics*, 18(15):492, 2017. DOI: 10.1186/s12859-017-1921-4

Part I

Commentary

Chapter 1

Introduction

For many people, the terms bioinformatics and DNA sequencing coincide and although, indeed, the advent of sequencing would not have been possible without bioinformatics, bioinformatics is a much broader field. Its origins date way back in time to the 1960s when protein sequence determination became available. The Edman degradation method [Edman and Begg, 1967], which was typically used at that time, was suitable for the determination of only short sequences and computational methods had thus to be developed to recover sequences of longer proteins [Gauthier et al., 2018]. The necessity to organize, store and distribute the obtained experimental data led to the development of resources such as the Atlas of Protein Sequence and Structure [Dayhoff, 1965], the first computerized protein collection [Strasser, 2010], ancestor of current major bioinformatics resources. At approximately the same time, advances in crystallography enabled to determine the three-dimensional structure of several proteins, which later became the foundation of the Protein Data Bank [Berman, 2008]. The Atlas and PDB thus formed the roots of the two main subfields of bioinformatics: sequence bioinformatics and structural bioinformatics.

The existence of resources such as the Atlas was and still is essential as it allows using algorithms to integrate and relate various types of data from a broad range of species. Such integration efforts then open ways for recovering information hidden in the experimental data and support discovery by means of transfer of knowledge between different levels of organization and species. This approach, fueled by the existence of the experimental data repositories, effectively enabled the transformation of biology and life sciences in general from purely experimental science towards more information-oriented science.

The above-mentioned knowledge transfer is only possible due to the role which similarity plays in molecular biology (see section 1.2). Indeed, the notion of similarity is deeply entrenched in the roots of bioinformatics and

underlies many of the computational approaches in the field. To understand why this is the case (section 1.2), we first need to introduce the main molecular biology concepts (section 1.1), how they are related to each other and how they map to the concepts in computer science (section 1.1.2).

1.1 Molecular players and their interactions

Although there exists a range of molecules involved in biological processes, the major focus of bioinformatics is on studying three classes of biopolymers: 1) DNA; molecules where the genetic information is stored, 2) RNA; molecules which both have the capacity to transfer the information stored in DNA and to carry out enzymatic function on their own and 3) proteins; the major functional molecules which carry out a plethora of biological functions in living systems such as catalytic function (enzymes), carrying small molecules (transport proteins), storage of other molecules for later use (storage proteins), building various structures such tendons, ligaments, hair, nails, silk and others (structural proteins), protection against external factors such as bacterial infection (defensive proteins), regulation of various molecular processes such as metabolism or gene transcription (signal/regulatory proteins) or detection of stimuli, e.g. on cell membrane (receptor proteins). Apart from the large biomolecules, also a plethora of small molecules is of interest to molecular biology since these either modulate the function of the macromolecules or are the sole purpose of existence of some of the proteins. For example, the purpose of hemoglobin is to carry around oxygen molecules.

The three macromolecules are closely linked as the information encoded in DNA is transferred to RNA which then translates to proteins. An RNA molecule is thus built on the blueprint of a part of a DNA molecule, i.e. gene, and similarly, a protein is built on the blueprint of an RNA molecule. This main pathway of information passing, together with DNA/RNA replication and RNA reverse transcription, is called the central dogma of molecular biology. The dogma thus describes the way how information is copied in the system [1].

The copying of information is one part of the flow of information in the living systems. The second part consists in spreading of the information. As soon as the information takes its shape, i.e. the functional form of a molecule is formed, the information starts spreading through the system by

[1] It should be emphasized that the copying is not one-to-one as some regions of DNA are removed during RNA splicing. Moreover, there exist other indirect sources of information as the resulting proteins are further edited by a range of posttranslational processes and information about these processes is not encoded in the gene coding for given protein.

means of interacting with its environment. Interaction ensures the spreading of information because the function of a molecule is determined by its interaction with other molecules. This molecular interplay happens between two or more molecules, which can be one of DNA, RNA, protein, or small molecule. Binding of molecules can affect the system in various ways. For example, one of the interacting molecules can change shape, opening the possibility for new action. A biological process then consists of a (not necessarily linear) series of such actions among molecules leading to given product or change. These series of actions securing specific functions are called pathways. Pathways can be seen as rather well-separated subsystems where the information is being received and produced via a set of well-defined inputs and outputs. Disruption of a pathway can then lead to a disease state, or, conversely, by targeting molecules in a pathway one can alleviate disease symptoms. However, before targeting a molecule, one first needs to understand which pathways the molecule is involved in because altering a molecule (for example, via genetic modification of a gene encoding given protein product or via introducing a molecule acting as an inhibitor or activator) can affect all the pathways in which given molecule is involved. However, if the change is targeted in such a way that it affects specific interaction site of a protein but keeps the others, then the impact of the change might not have an effect on the protein's other functions. Therefore, information about all the functions of a molecule in the system and how these are carried out is of great importance. And so are the methods for their detection.

1.1.1 Levels of molecular structure organization

As the molecules exist and interact with each other in a three-dimensional environment, 3D representation of a molecule, i.e. the spatial arrangement of atoms and their corresponding physico-chemical properties, contains all the information governing the molecule behavior. However, to obtain the structure of a molecule is more difficult than obtaining its sequence. So the full structure might not always be available or it might not even be required as it might add an unnecessary level of complexity. For those reasons, different levels of organization/abstraction known as primary, secondary, tertiary and quaternary structure are being exploited when working with the DNA, RNA and protein molecules.

DNA, RNA, and protein molecules are linear polymers which fold into intricate three-dimensional shapes via intra-molecular residue interactions. The monomeric units (residues) are (di)nucleotides, in case of DNA and RNA, and amino acids, in case of proteins. The linear sequence which is formed by covalent bonds between the residues is called the primary structure, sequence

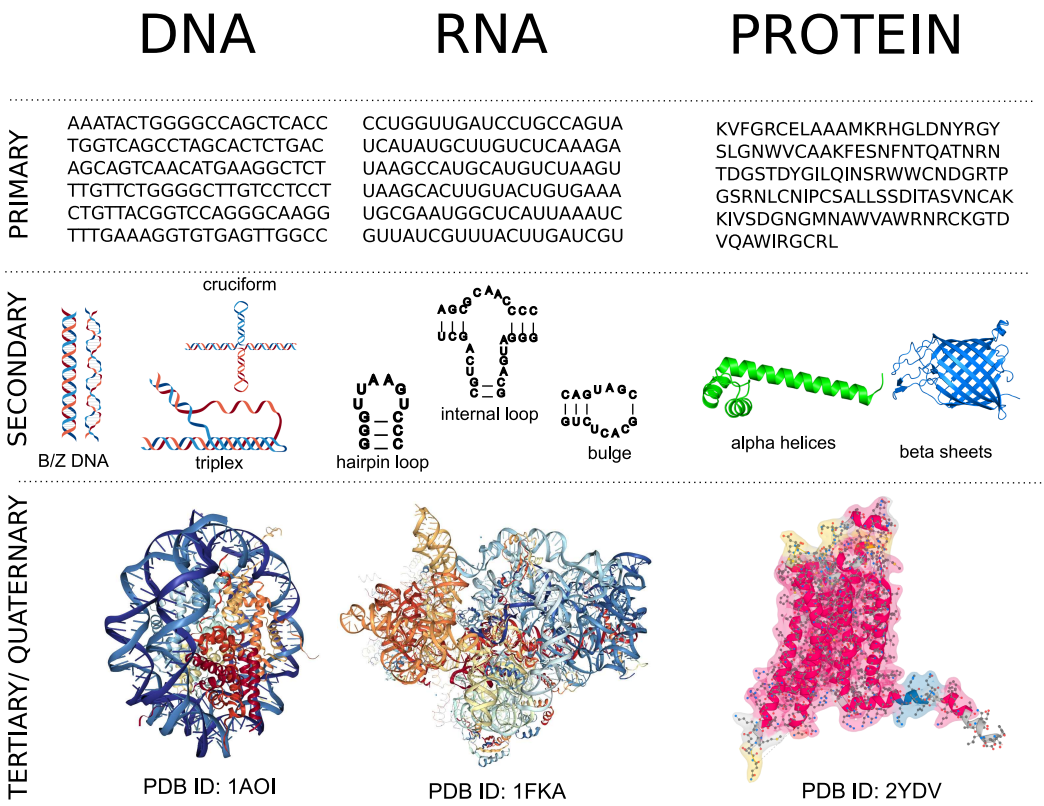


Figure 1.1: Illustration of the levels of molecular organization. Please note that the primary and secondary levels do not represent real molecules. Secondary level images only show several secondary structure motifs for the purpose of illustration. The illustration of DNA secondary structure is taken from [Bochman et al., 2012]. The full secondary structure can be much more complex. For example Figure 2.3 shows full secondary structure human small ribosomal RNA subunit. The tertiary/quaternary structures, on the other hand, are visualizations of real experimental data. The DNA (PDB 1AOI) visualization shows a nucleosome complex consisting of histone proteins in the middle encircled by a DNA loop. The RNA visualization (PDB ID 1FKA) is an example of small ribosomal RNA subunit. Finally, the protein visualization (PDB ID 2YDV) is an example of a GPCR protein, concretely the adenosine A2A cell membrane receptor. Here, the red part, consisting of a stack of seven alpha-helices, is the part of the protein which is nestled in the cell membrane while the rest of the structure is able to communicate information from the exterior of a cell to its interior.

or chain. The atoms which are connecting the residues in the chain form so-called sugar-phosphate (DNA, RNA) or polypeptide (proteins) backbone.

The secondary structure is characterized by the hydrogen bonds, or possibly other non-covalent bonds, within the molecule. These bonds tend to form regular substructures which are known as secondary structure elements. DNA consists of two chains where nucleotide i is hydrogen bound to a nucleotide $n - i$ (n being the length of the nucleotide chain), but apart from this regular and well-known pattern DNA secondary structure can also form non-standard elements such as G-quartets, cruciform or triplex which were long thought to be only in vitro artefacts [Bochman et al., 2012]. RNA, which contains only a single chain which folds onto itself, forms a complicated secondary structure with typical secondary structure elements being hairpin loops, bulges, internal loops, and multibranch loops [Hendrix et al., 2005]. In the case of proteins, we recognize two main types of secondary structure elements: α-helix and β-sheet. Multiple secondary structure elements can come together in the 3D space, forming so-called structural motifs, or super secondary structure. These are important since the specific arrangement of secondary structure motifs tend to be associated with particular chemistry and thus function. Although the same structural motif can be found in molecules with dissimilar function, they can be used as an indicator of a specific behavior. For example, lipocalins, a family of proteins which transport hydrophobic molecules, are known to share beta barrel in its structure[Flower et al., 1993].

The positions of all the atoms of the constituent residues are referred to as the tertiary structure of a molecule. The tertiary structure is formed by a process called folding when the linear chain of residues folds into its target three-dimensional shape.

If multiple chains need to come together for a molecule to assume a functional form, the resulting multimer is referred to as the quaternary structure. In the case of proteins, the quaternary structure consists solely of polypeptide chains. In the case of nucleic acids, a combination of nucleic acids with other types of molecules can be called quaternary structure as well.

As mentioned at the beginning of this section, the tertiary/quaternary structure of a molecule dictates the interaction with its environment and therefore, out of all the organizations, it is closest to the function. However, structure determination is a complicated process compared to sequence determination. Luckily, as expressed by the central dogma, the sequence codes for structure and as such, it can be used for gaining insights into molecular function in cases when the structural information is not available. When, moreover, the secondary structure is available, it can be used as another piece of information because the secondary structure can serve as an approximation of the tertiary structure. This is because knowing which residues are bonded by non-covalent bonds also gives information about which residues are close to each other in the space. So in cases where the tertiary structure is not

available, sequence, possibly enriched by secondary structure can take the role of a proxy to the 3D structure. In any case, having the structure and being able to operate on it is highly desirable.

1.1.2 Common computational representation

Each of the levels of organization introduced in the previous section can be expressed by a particular data structure. The ability to express a molecule via a standard computational representation is important as it allows one to abstract the molecules into concepts which are easily approachable with existing computer science techniques. Such techniques can be either used directly, but more often they need to be adapted to fit the specifics of the particular application.

Since the approaches introduced later in this thesis operate over structures of RNA and protein molecules, we will introduce here only representations which are being used for these two types of molecules.

Primary structure As the primary structure of all the biopolymers we are interested in is linear, it can be expressed as a word over an alphabet of either 4 letters, in case of nucleic acids, or 20 letters[2] in case of proteins. The letters are often associated with additional information pertinent to given residue.

The representation of a molecule as a sequence is very helpful because it allows to either directly apply or adapt the plethora of string algorithms. Examples include sequence alignment and similarity assessment, searching for motifs, efficient database search and many others [Gusfield, 1997].

Secondary structure As protein secondary structure does not have any intricate internal structure, it is commonly represented as a list of hydrogen-bonded amino acids. A similar representation is possible also for the secondary structure of RNA molecules. However, unlike in case of proteins where individual elements are treated separately (at least from the point of view of the computational representation), RNA secondary structure is often treated as a single set of hydrogen bonds (nucleotide pairs) covering the whole molecule. This set is represented either as a graph of a tree. In the graph representation, the nucleotides correspond to nodes and bonded nucleotides form edges. A more intriguing option is to represent the secondary structure as an ordered tree [Elias and Hoksza, 2017]. In this representation, inner nodes represent base pairs (bonded nucleotides) and unpaired nucleotides

[2] considering only the basic amino acid types

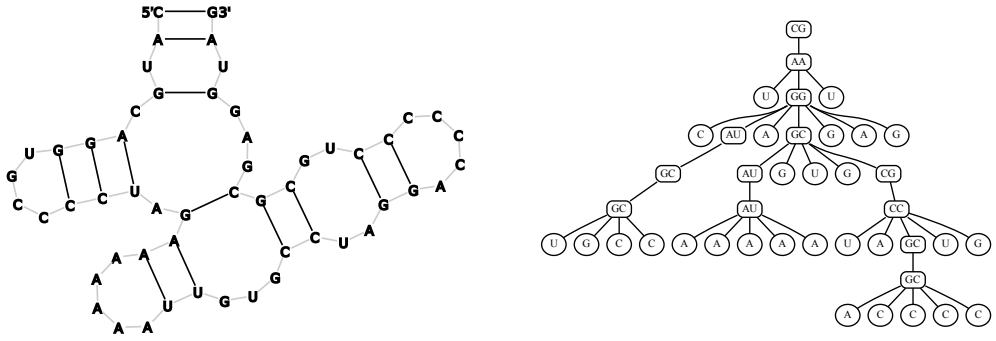


Figure 1.2: Tree-based RNA secondary structure representation. Example of a secondary structure (a) and its corresponding tree representation (b). In (a) the gray lines represent the covalent bonding, i.e. the sequence, while the black lines represent hydrogen bonding, i.e. the secondary structure.

form leaves of the tree as illustrated in Figure 1.2[3]. To build such a tree from an input structure (which can be given as a sequence accompanied by a list of base pairs), one simply traverses the secondary structure in sequence order from both ends simultaneously and transforms the encountered paired and unpaired nucleotides into inner nodes or leaves of the nascent tree. The order of neighboring nodes is defined by the order in which the nodes are encountered during the traversal.

Again, the ability to represent secondary structure as a tree (or possibly a graph) opens the way for utilization of complex graph or tree algorithms (such as tree edit distance [Elias and Hoksza, 2017]) for secondary structure analysis.

Tertiary/Quaternary structure The tertiary structure is essentially an ordered set of labeled 3D coordinates corresponding to every single atom of the molecule. However, these coordinates have an internal structure, therefore both protein and RNA molecules use to be stored as a tree with the list of chains being at the top level, followed by an ordered list of residues for each chain and list of atoms for each of the residues. The atoms are further labeled by the atom type, such as C-α amino acid atoms. One can thus discern which

[3]It can happen that the structure also contains so-called pseudoknots; base pairs which invalidate the tree structure as they are formed between nucleotides in such a way, that the graph contains cycles. In such a situation, these pairs are usually either ignored or treated in a special way.

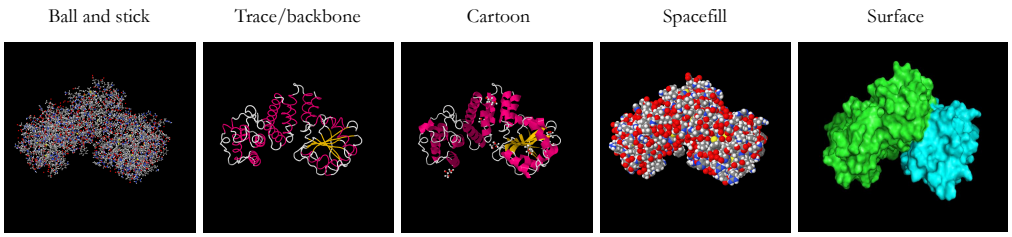


Figure 1.3: Example of different types of visualization of a protein (PDB ID 4CRW) with two chains. Ball and stick and spacefill visualizations show every single atom, but in the spacefill visualization, the size of an atom corresponds to its Van der Waals radius. Backbone visualization tracks only C-alpha atoms of every amino acid. Cartoon visualization is based on the backbone atoms but emphasizes secondary structure types (helices and sheets). The surface visualization is a smoothed version of the spacefill visualization. The coloring of the surface is based on chains demonstrating how two tertiary structures form the protein's quaternary structure.

of the atoms form the polypeptide or sugar-phosphate backbone. As many algorithms do not need to work with the full atomic resolution, each residue is often represented by one of its, usually backbone, atoms.

A natural representation of the set of the coordinates themselves, apart for an unordered set, is the graph representation. One natural way to transfer a protein structure to a graph is to store residues or atoms as nodes and chemical interactions as edges. Often a more general condition for edge formation is used: nodes representing residues or atoms are connected by an edge if they are within a certain distance from each other [Diallo and Dhifli, 2015, Dong et al., 2014, Jelínek et al., 2017].

Each of the residues exhibits physico-chemical properties which are commonly represented as vectors of features alongside the atoms or residues. These have an impact on the local environment of the protein and thus often also the points in the close neighborhood are stored and utilized by algorithms [Krivák and Hoksza, 2015b, 2018, Jendele et al., 2019].

As there are multiple approaches to visualize the tertiary structure, some of which we use in this thesis, Figure 1.3 showcases the major types which one can encounter.

1.2 Similarity as the driving force for in-silico function discovery

Having introduced the key molecular players, how they relate to each other and how they are represented, let us now discuss how the concept of similarity arises in molecular biology and why it is so prevalent.

In section 1.1, we described the flow of information in terms of information copying and spreading. This is in line with Pevsner's view [Pevsner, 2015] of the field of bioinformatics and genomics. Pevsner sees the field from three perspectives: 1) the perspective of the cell and the central dogma of molecular biology; ii) the local perspective of the organism, which shows changes between the different stages of development and regions of the body and iii) the global perspective of the tree of life, in which millions of species are grouped into the evolutionary branches [Diniz and Canduri, 2017]. The contributions of this thesis find its application in function discovery and are thus located in the second, local, perspective domain. However, all the contributions are, in fact, enabled by principles expressed in the global perspective from which comparative approaches, i.e. approaches based on the transfer of knowledge between macromolecules, draw. By the knowledge transfer, we understand the transfer of information about one molecule to another molecule, possibly from another species. For example, having two similar proteins and knowing the position of an active site for one of them allows us, due to the shared similarity, to hypothesize about the position of the active site in the other molecule.

A well-known concept in drug discovery is the similarity principle due to which the transfer of knowledge between different molecules is possible. The principle states that similar molecules tend to show similar activities. An analogous principle can also be seen in molecular biology, however in molecular biology, one needs to distinguish between similarity due to evolution and similarity due to shared structural and physico-chemical properties.

Molecular evolution describes how genes behave under the pressure of molecular and population forces acting upon a genome. As novel genetic variations arise through mutations, they start spreading across a population and eventually result in speciation. Mutations can be either simple point mutations, but can also include larger modifications such as gene duplication. Both original and duplicated genes continue gathering modifications both before and after speciation, resulting in a group of genes sharing a common ancestor gene. The genes in such set are called homologous genes. From the previous follows that homologous genes can come either from the same species or from different species.

As homologous genes share a common ancestral gene and the divergence happened by a series of mutation events, it is possible to define a mapping between every pair of residues of homologous genes. The further the point in time in which the genes split, the higher divergence and lower similarity of the genes. As the modifications result in the change of structure, the same holds for the function. Although the mutations can happen anywhere in a gene, some regions tend to be more conserved than others. These are typically regions which either correspond to active sites of regions which are important in maintaining the overall shape of the molecule. On the other hand, mutations in functionally unimportant regions, especially on the protein surface, tend to be more common. Also, genes coding for molecules which are expressed in every cell and essential for some core function are more conserved. This is, for example, the case of genes coding for ribosomal RNAs which are one of the most conserved genes across all species [Isenbarger et al., 2008] as the ribosomal RNA majority of the ribosomal complexes, the factories where proteins are assembled.

Naturally, the nucleotide mapping between homologous genes extends to the RNA and protein products of these genes. Therefore, knowing the evolutionary mapping of two RNA or protein molecules tells us the correct alignment of their sequences and correct structural superposition of their tertiary structures. If one of the molecules is annotated on the residue level, for example there exist knowledge about which residues are part of an active site or which regions tend to be often mutated, this knowledge can be potentially transferred to the rest of the homologous molecules. Of course, the information needs to be scrutinized because due to the accumulated mutations, given property might be lost.

We should, however, emphasize that the fact that two genes or their products are similar, either on the level of sequence or structure, does not imply they are also homologous. So one needs to be careful when using the term similarity in the context of molecular biology. In the case of function discovery, the transfer of residue-level knowledge is only reasonable in a case when the objects between which the transfer happens are homologous.

On the other hand, it is important to realize that the presence of a shared structural and physico-chemical similarity between molecules can still point toward common behavior. Examples are supersecondary structure motifs such as the beta barrel mentioned in section 1.1.1. Of course, this is a property stemming from the structural arrangement of physico-chemical properties rather than from common molecular ancestry. Therefore, using commonalities on the structural level is valid, however, it needs to be realized that there is no direct evolutionary basis.

Although similarity does not necessarily imply shared ancestry, it is

commonly accepted that proteins (similar case can be done for RNA molecules) which can be aligned in such a way that they share at least 25% identical residues tend to be conserved. This percentage increases with the dropping length of the sequence [Krieger et al., 2003]. Therefore, one can, with care, use similarity as the sign of common ancestry and carry out the transfer.

As basically all molecules share a common ancestry, the notion of similarity is indeed omnipresent in the analysis of protein, RNA and DNA molecules. Although the distinction between similarity due to common ancestry (homology) and similarity emerging simply by chance needs to be considered, both types have their role in function discovery.

One last comment needs to be made to the relation of sequence and structure concerning the knowledge transfer. As the function of a molecule is based on its tertiary structure, the structure tends to be more conserved than sequence. The reason for this higher conservation is that genetic code is ambiguous, with different nucleotide triplets coding for the same amino acid. Thus, two different DNA sequences can code for the same protein sequence. Moreover, how a molecule folds in the three-dimensional shape and what molecules it can recognize is dependent on the size, shape and physico-chemical properties of the constituent amino acids. Therefore, mutations leading to similar amino acids can be quite easily accepted because they result in very similar three-dimensional folds, sometimes even in cases when the underlying sequences differ substantially. Using structure, if available, can therefore in some cases bring insights which sequence alone could not.

1.3 Contributions of the thesis to the field of functional analysis

The problem of function discovery is a multi-faceted one since life is an emergent property of a complex interplay of many different types of molecules. To fully understand the system, one needs to decipher how different molecules recognize each other and interact. In the previous section, we outlined the two complementary views of similarity and its application in bioinformatics: the one rooted in molecular evolution and the one which is based on the emergent properties of structure and associated physico-chemical properties. The contributions of this thesis are based on both of these approaches. In the following list outlining the contributions, the RNA approaches are enabled by the evolution-based similarity, i.e. homology, while the algorithms for protein active sites detection are based on learning structural and physico-chemical properties associated with activity without considering which molecules these

properties come from, and thus do not assume any level of homology.

- Approaches applicable for RNA function discovery
  - Modeling RNA structural similarity (section 2.1)
  - Supporting visualization of RNA secondary structure (section 2.2)
- Approaches applicable for protein function discovery
  - Prediction of protein-ligand binding sites (section 3.1)
  - Prediction of protein-protein binding sites (section 3.2)

1.3.1 Comment on software development

As the goal of the presented research is to support practical function discovery, all the computational approaches are backed up by software solutions implementing the proposed algorithms. The software solutions are always developed as command-line applications so that users can plug them in larger computational pipelines. In some cases, the solution is also available via a web interface or API. We strongly believe that software development is an integral component of bioinformatics research and thus developing only concepts without functional software is not enough. This is especially true considering the fact that computational biology is becoming increasingly complex, incorporating a plethora of various, often heterogeneous, data and computational approaches. The discoveries are then made only via integration of the data into complex pipelines consisting of multiple different applications. Therefore, it is becoming important to not only develop algorithmic solutions to problems but also to accompany them with software solutions which can be used as parts of complex computational pipelines.

Easy-to-apply software also helps to foster collaboration and gain visibility. This is the case of the Traveler [Elias and Hoksza, 2017] and P2Rank [Kriv´ak and Hoksza, 2018, Jendele et al., 2019] software tools, two of the contributions of this thesis.

Traveler solves a previously unapproachable problem of visualization of the secondary structure of large RNA molecules for RNA analysis. The solution is distributed as an easy-to-use command-line application which is currently incorporated into RNACentral The RNAcentral Consortium [2018], the world-leading resource of RNA-related data hosted by the European Bioinformatics Institute (EBI), and is supporting visualization of more than seven million RNA molecules.

P2Rank [Kriv´ak and Hoksza, 2018, Jendele et al., 2019] is a software solution for detecting protein-ligand binding sites on a protein structure.

P2Rank is distributed as a command-line application, web application and existing predictions are also available via its REST API. Because a lot of effort was put into making it a robust software solution it is now the major contributor of protein-ligand annotations into EBI's Protein Data Bank in Europe - Knowledge Base (PDBe-KB) [PDBe-KB consortium, 2019], the new PDBe's major resource of integrated protein data.

Chapter 2

RNA functional analysis

For a long time, it was believed that the only role of RNA is to transfer genetic information from the DNA to ribosomes where it is used to translate it into proteins. However, it was shown that not all RNA is translated into proteins. Indeed, many RNA molecules which are transcribed from DNA, have a function on their own. Such molecules are called non-coding RNAs and are involved in a range of functions from gene regulation to being the main structural components of the ribosomal complexes.

The growing understanding of the different roles which the RNA molecule plays in various biological processes resulted in an increasing interest in the study of RNA. This, in turn, led to the growth of the experiment backed-up RNA data including sequence information, structural information, but also the availability of high-quality computationally generated data such as predicted RNA tertiary structures. However, not only the number of known RNA molecules has grown; so has their size, calling for novel methods for retrieval and analysis of large RNA structures.

2.1 Secondary structure-supported tertiary structure similarity modeling

Similarly to other molecules, the function or RNA is largely determined by its tertiary structure and thus it is reasonable to involve the structure information in the function discovery process. As the resources of RNA structures with determined function grow, it makes sense to use these molecules as a proxy to the function of molecules with yet undetermined function.

With a database of molecular structures with determined function and a query molecule with known structure but unknown function, it is possible to devise a similarity measure which can be used to retrieve from the database

structurally and thus also possibly functionally similar molecules to the query molecule. To be able to reason about homology and visually inspect and interpret the results, it is desirable for the similarity procedure to be able to output not only a similarity value but also superpositions of the query and identified database structures.

2.1.1 Pairwise RNA structure superposition

As the number and size (several thousand nucleotides in case of ribosomal RNAs) of structures is growing a similarity measure should be both efficient and effective. In [Hoksza and Svozil, 2012], we introduced a solution for pairwise RNA tertiary structure comparison called SETTER (SEcondary sTructure-based TERtiary Structure Similarity Algorithm). Although there had already existed several solutions for RNA structure comparison at the time of introduction of SETTER (ARTS [Dror et al., 2005], DIAL [Ferrè et al., 2007], iPARTS [Wang et al., 2010], SARSA [Chang et al., 2008a] and R3DAlign [Rahrig et al., 2010a]), SETTER was able to achieve comparable or higher effectiveness in several times lower runtime. Moreover, it was able to superpose even the largest structures which had not been possible before.

In most superposition-based comparative approaches, a mapping between residues of the molecules to be compared is found and a superposition is then obtained, typically via root-mean-square deviation (RMSD) or similar method. This results both to a score which is interpreted as the distance (or inversely similarity) of the molecules and to a superposition which can be visualized and inspected. The bottleneck in terms of time complexity is the process of mapping between the residues. Since the outcome of this phase also determines the resulting superposition, this step is also essential in providing high-quality effective similarity measure. In all of the referenced methods, the complexity of this step is at least $O(N^2)$ with N being the number of atoms to be superimposed. Although none of the methods works with full atom representation of the molecules, but consider only one representative atom per nucleotide, N can still be in more than five thousand for the largest RNA structures.

The main idea of SETTER is thus to decrease the number of elements to be compared by representing each RNA structure by a set of non-overlapping generalized secondary structure units (GSSUs). The notion of GSSU is a novel idea introduced by SETTER. Each GSSU consists of three parts: a stem, a neck and a loop (see Figure 2.1), thus forming a standardized unit of secondary structure. The GSSU was proposed to be able to represent the RNA molecule as a set of function-related, simplified, standardized components. The relation to function is based on the fact that secondary structure motifs are known

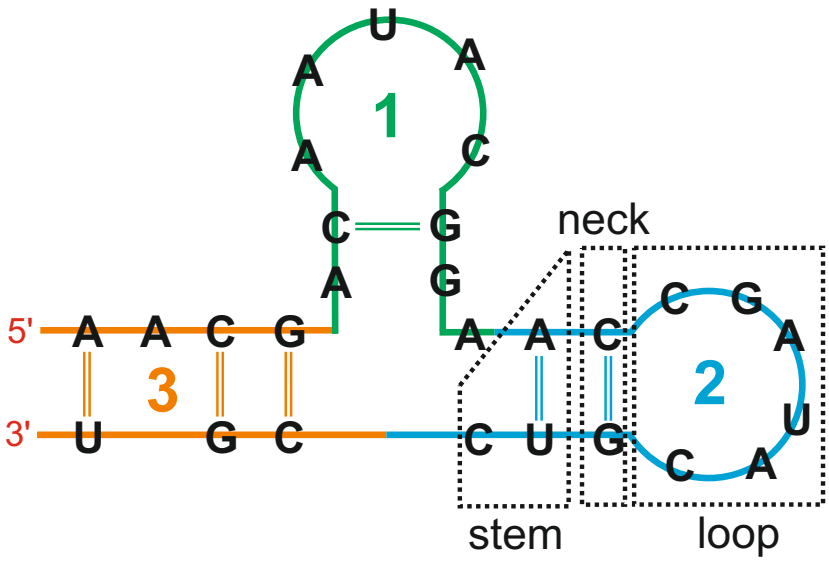


Figure 2.1: Example of partitioning of a secondary structure into three generalized secondary structure motifs (GSSU)

to be important for molecular recognition and a secondary structure-based division would probably follow functional motifs, or it will not, at least, break them.

The structural alignment of RNA structures S_1 and S_2 is then obtained by using a distance measure considering RMSD-based transformations between all possible pairs of GSSUs. To align a pair of GSSU structures, a set of triplets of key residues from each GSSU are superimposed by the Kabsch algorithm [Kabsch, 1976]. After finding the best-aligned GSSUs, the rest of the GSSUs are aligned in a linear fashion resulting in a score representing the distance of the two RNA structures. Finally, this score is a subject to statistical significance evaluation. The resulting p-value can then be used in situations when SETTER is used to scan a collection of RNA structures. All the steps are in greater detail described in [Hoksza and Svozil, 2012].

The algorithm scales as $O(n^2)$ with the size of GSSU and $O(n)$ with the number of GSSUs in the structure. A further speedup is achieved by considering only the same type of residues (stem, loop, neck) for pairing when mapping GSSUs. Finally, SETTER uses an early termination heuristic condition in the all-to-all GSSU comparison phase. These features combined result in SETTER's superior speed compared to the existing solutions allowing to superpose even the largest RNA structures which had not been possible before the introduction of SETTER.

Naturally, the speed of a solution need not come for the price of accuracy,

which is the ultimate goal of most of the bioinformatics applications. However, to asses the quality of a structure similarity method is not a straightforward task because the exact evolutionary relations between the nucleotides, i.e. the mapping, is difficult to obtain. And thus, we cannot with certainty tell whether a mapping, and thus given structural superposition, is correct. Therefore, to estimate the quality of RNA structure superposition, typically two, rather indirect, types of measures are used: geometry-based and function-based.

The geometry-based assessment tests how close two superposed structures are in the 3D space, often taking account which types of residues are aligned because one would expect more often the same nucleotide types to be close together in 3D space. The measures include RMSD, the percentage of structural identity (PSI), the percentage of sequence identity (PID) [Capriotti and Marti-Renom, 2008, 2009], the number of mapped nucleotides and the number of exact base matches [Rahrig et al., 2010b].

Table 2.1: ACC and AUC comparison of SETTER, iPARTS, and SARA on the FSCOR and T/R-FSCOR datasets. The values are given in % and are reported for exact/similar classification. In the "similar" classification, molecules having the same parent class are treated as if they shared class. iPARTS should be compared to SETTER with the p-value threshold of 1.0 (i.e. no filtering applied), and SARA should be compared to SETTER with the p-value threshold of 0.013 which corresponds to a significance threshold applied in SARA's publication. For iPARTS, ACC was not reported and necessary tests could not be performed using the iPARTS web interface.

	FSCOR		T/R-FSCOR	
	AUC	ACC	AUC	ACC
iPARTS	72/92	?	77/90	?
$STR_{pV=1.0}$	82/91	61.8/72.8	87/89	67.4/71.8
SARA	61/83	81.4/95.3	58/85	78.0/94.5
$STR_{pV=0.013}$	71/87	80.5/95.1	83/91	91.7/95.0

The function-based evaluation assesses the quality of a method by its ability to correctly assign a class to a query RNA from the SCOR database [Tamura et al., 2004]. SCOR is a human-curated hierarchically organized database of RNA structures based on the function and tertiary interactions of RNAs. As such, it can be used in benchmarking of similarity methods where the goal is to mimic the SCOR class assignment. Using SCOR, one can simulate the database search scenario where the database structures coming from the same class as the query structure form the positive set. It is then possible to measure

accuracy (ACC) as the ability of the method to identify a structure from the correct class as the most similar to the query. However, to have a better picture of the overall performance, ROC (receiver operating characteristics) curve and AUC (area under the AUC) are more suitable. To obtain the ROC curve, the alignments of all pairs of RNA structures are sorted by their scores (P-values in case of SETTER). A threshold score is varied between the minimum and maximum of the sorted scores. All aligned structures with a score exceeding the threshold are considered positives while the ones below the threshold are considered negatives. From here, the number of true/false positives/negatives can be obtained and ROC/AUC computed.

Although our method was evaluated using both geometric and functional approaches, we believe that the function-based evaluation is closer to the intended application, i.e. the function determination and annotation. Table 2.1 displays SETTER's performance in terms of ACC and AUC with respect to structurally diverse subsets of SCOR (FSCOR, T-SCOR, F-SCOR) [Capriotti and Marti-Renom, 2008] showing that SETTER performs better than existing solutions in most cases.

2.1.2 Multiple RNA structure superposition

The efficiency of SETTER allowed us to apply the principle of GSSU decomposition to a more complex problem of multiple structure superposition (MSS). The goal of MSS is to superpose multiple structures onto each other in such a way that positions which share common ancestry are aligned onto each other as best as possible. This is analogous with the well-studied problem of multiple sequence alignment which is the basis for phylogenetic reconstruction, motif finding and a whole range of other applications [Chatzou et al., 2015].

While several multiple RNA sequence alignment algorithms exist [Kiryu et al., 2007, Moretti et al., 2008], the choice of methods for multiple 3D structure alignment is rather limited. In existing solutions, the 3D structure alignment is based either on the RNA secondary structure [Torarinsson et al., 2007, Tabei et al., 2008] or on the projection of structural features into a sequence followed by the sequence alignment [Chang et al., 2008b]. However, an algorithm for direct multiple 3D RNA structure alignment was missing. Therefore, we introduced the MultiSETTER [Hoksza and Svozil, 2014], structure-based multiple superposition solution based on the GSSU decomposition. MultiSETTER was the first 3D structure-based solution at the time of publication of the method.

The main idea of MultiSETTER is to adapt the well-known ClustalW algorithm [Thompson et al., 1994] which was developed for solving the multiple sequence alignment problem. In MultiSETTER, we transform the sequence-

specific parts of ClustalW into their structure counterparts resulting in the following steps (details can be found in [Hoksza and Svozil, 2014]):

- Each pair of RNA structures is aligned by SETTER, and the distance matrix is constructed from the pairwise distances.
- A guide tree is calculated from the distance matrix using the neighbor-joining algorithm [Saitou and Nei, 1987].
- Two most closely related structures are first superposed and the so-called average RNA structure is constructed by merging (averaging) positions of individual atom pairs. The alignment then progressively continues, merging more and more structures, until the root of the guide tree is reached. The root contains the global average structure, which we will call simply the average structure from now on.
- Each of the input structures is superposed onto the average structure resulting in multiple structure superposition.

The GSSU decomposition not only allows for efficient pairwise comparison but equally important is its role in the structure merging. To merge a pair of structures, GSSU mapping is carried out and the actual merging then happens on the level of the individual GSSUs. The procedure is relatively straightforward due to the standardized nature of GSSUs. The merging process is further adjusted to be robust with respect to outliers.

To benchmark our solution, we again utilized the SCOR database from which we extracted 96 structures classified into 14 families of various sizes and intrinsic diversities. To test the ability of functional classification of a molecule Q using MultiSETTER, we computed the average structure for each of the classes, not considering Q. Then we used SETTER to compute the distance of Q to each of the average structures, i.e., representatives of classes and assigned Q to the class with the closest average structure. We applied similar approach for the pairwise alignment where instead of using the average structure, we used an average of pairwise distances between Q and every structure in a family as a proxy. We showed that using multiple superposition provides better results than using pairwise alignments. Moreover, we also demonstrated that MultiSETTER leads to better results than existing sequence-based RNA multiple alignment solutions.

Alternatively, MSS can also be used to guide the pairwise structural superposition. Here, a set of homologous structures can be used to improve the pairwise alignment by anchoring the two structures to be superposed to the average structure of the set of homologs. This nicely illustrates a quote

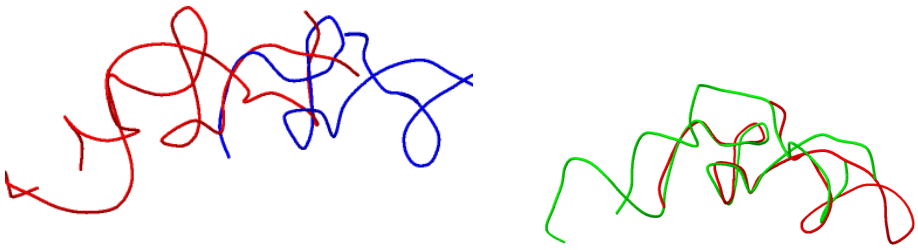


Figure 2.2: The structure alignment of 4ABR and 1P6V structures belonging to the RF00023 Rfam [Burge et al., 2012] family. a) Pairwise structure alignment of 4ABR and 1P6V produced by SETTER. b) The alignment of 4ABR and 1P6V taken from the multiple structure alignment of the whole RF00023 family produced by MultiSETTER

attributed to Artur Lesk: Öne or two homologous sequences whisper . . . a full multiple alignment shouts out loud. Figure 2.2 illustrates this by showing how using the average structure of a family can lead to a better superposition of two members of the same family than when using the members alone.

2.1.3 Software solution

Both SETTER and MultiSETTER approaches are distributed in a single software solution. This solution consists of a command-line application and web interface. The command-line application, which also serves as the back end for the web solution, contains four modules: i) pairwise superposition module, ii) multiple superposition module, iii) batch pairwise superposition module, which is passed a list of structures and computes all pairwise similarities and iv) batch multiple superposition module, which is passed an average structure obtained from the multiple superposition module and a list of structures which need to be superposed over the average structure. Each of the steps returns detailed information about the superpositions and a Jmol [jmo] script which is used to visualize the results in the web application or can be used offline by the user. The application supports both Windows and Linux operating systems and most of the steps of the algorithm are also

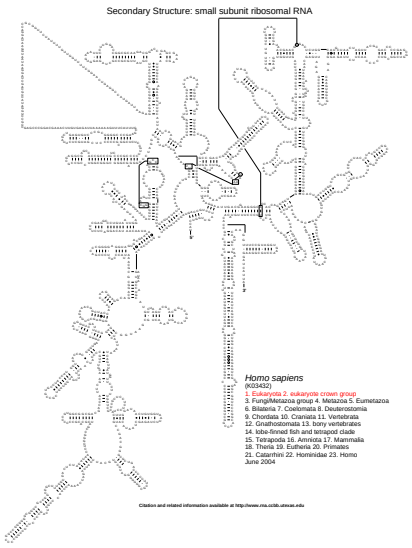
parallelized, providing substantial speed up (see [Hoksza and Svozil, 2014]) when running on multi-core architectures. The web application provides an interface to the MultiSETTER command-line application providing the user with the ability to submit a pair of or multiple structures. These are processed and the results, including the 3D superpositions, are subsequently visualized. Moreover, the user is provided with detail statistics for offline inspection.

2.2 Template-based secondary structure visualization

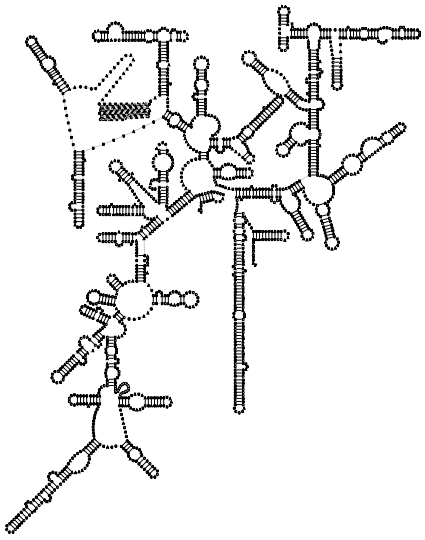
Although tertiary structures of RNA molecules are becoming readily available in structural databases such as PDB, structural biologists are more used to describe functional regions of RNA molecules in terms of their secondary structure. This is especially the case with large RNA molecules which have thousands of nucleotides and inspecting their three-dimensional structure can be challenging. For that reason, secondary structure visualization is the go-to choice when visual inspection of large RNA structures is needed. Three layouting strategies for displaying the secondary structure exist: linked graph, circular graph, and classical structure [Wiese et al., 2005]. In the linked graph display, the nucleotides are drawn on a straight line in the sequence order, and base-paired residues are linked by an arc. The circular graph is similar to the linked graph representation with the nucleotides laying on a circumference of a circle and connected with curves. However, both of these representations lack the ability to capture the secondary structure motifs and therefore the classical structure is used when detailed visual analysis of secondary structure motifs and their interaction is required. In the classical structure display, the positions of nucleotides are chosen so that the secondary structure motifs can be discerned [Elias and Hoksza, 2017].

As desscribed in section 1.1.2, the secondary structure of an RNA molecule can be represented as a graph and thus using existing graph drawing algorithms seems to be a natural solution to the problem of laying out the structure. However, these standard solutions which tend to optimize the typical aesthetics criteria are not applicable for RNA secondary structure visualization which has its specifics [Auber et al., 2006]. These specifics constraint lengths of some of the edges and define how local motifs should be laid out. Although these criteria, however vague, exist for small motifs there are no such rules for how these motifs should be positioned with respect to each other. This leads to virtually infinitely many possibilities of how to lay out a complex RNA structure. The absence of rigid criteria when assessing the quality of a layout leads to the fact that secondary structure visualization is largely habitual and while the layout of small secondary structure motifs, such as hairpins, are similar in different tools, their mutual positions differ greatly across the existing visualization tools. Not only do the layouts differ but for large structures the layouts are virtually ineligible (see Figure 2.3) and therefore only semi-manually created layouts had been used for large RNA molecules.

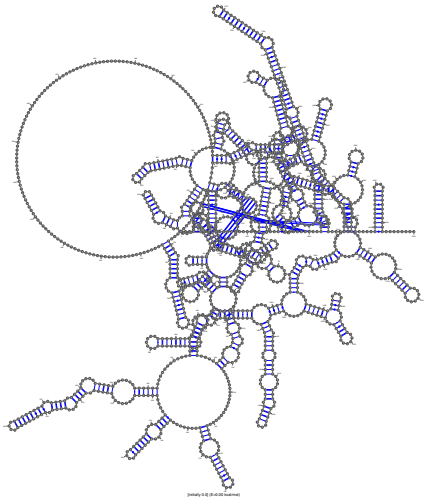
The fact that large, i.e. ribosomal, RNA structures tend to be highly



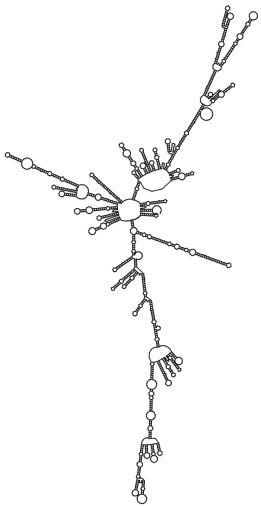
(a) Layout in the form biological community is used to (downloaded from the CRW website Cannone et al. [2002]).



(b) Layout generated by Traveler using fruit fly as a template.



(c) Layout generated by VARNA (version 3-93).



(d) Layout generated by RNAplot.

Figure 2.3: Layout of small subunit of human ribosomal RNA (GenBank accession number K03432) by two most often used tools - VARNA [Darty et al., 2009] and RNAplot [Lorenz et al., 2011].

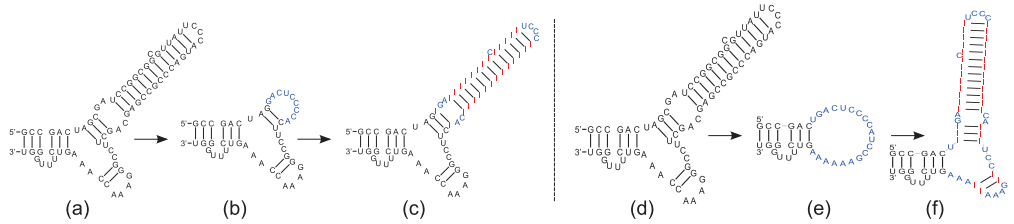
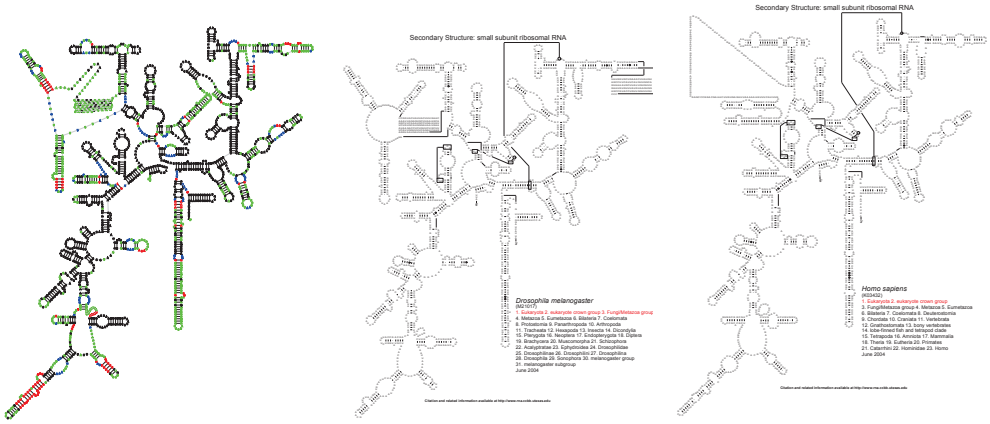


Figure 2.4: Traveler's ability to recreate layouts. On the left-hand side, we took a structure with two hairpins (a), removed part of a stem and used the original structure as the template (b). Then we reinserted the residues and used (b) as a template to obtain (c). Similarly, (d), (e) and (f) show re-creation of the starting structure with a more drastic middle step where the two hairpins lose residues so that the remaining residues form a loop. (f) demonstrates that Traveler can successfully recreate the original structure. For clarity, the new residues were labeled I and shown in red, while the residues which needed to be repositioned are shown in blue.

conserved led us to the observation that the layouts of such structures which are used in the scientific literature tend to be conserved as well. Therefore, one could use a molecule with a known layout as a template for laying out a similar molecule with yet unknown layout. This idea was implemented in our template-based RNA secondary structure approach called Traveler [Elias and Hoksza, 2017]. The algorithm takes on its input a structure to be laid out (target) and a homologous structure with known layout (template). Both target and template are converted into the respective tree representations and tree edit distance is applied to identify the minimal sequence of tree edit operations which would turn the template tree into the target one. Each tree edit operation is assigned a visual edit operation which is then used to transform the template layout into the target one. Here, Traveler greatly benefits from the underlying tree representation since the tree can be used to identify which nodes are affected by a given operation. For example, if a node needs to be removed, then this corresponds to the removal of the node and application of a shift to all nodes in the subtree of the removed node.

Since the set of edit operations that drives the transformation corresponds to minimal tree edit distance, the conservation of common properties of the layouts is ensured. The method is thus capable of producing a secondary structure complying with the intuition of a biologist when a homologous structure with an appropriate layout is already available.

Testing a layouting solution is rather difficult, especially in the case when no direct qualitative criteria are available. Therefore, our experiments have



(a) Generated human 18S rRNA using fruit fly's 18S rRNA layout from CRW as a template.

(b) Fruit fly 18S rRNA.

(c) Human 18S rRNA.

Figure 2.5: Visualization of human 18S rRNA with Traveler. (a) shows the target layout, (b) is the template layout while (c) is the desired layout as stored in the CRW. The Traveler's output is colored so that red represents inserted residues, green are relabeled residues and blue are residues that needed to be shifted due to indels happening within given hairpin (see supplementary material for full-color coding definition).

been mainly visual, for example validating that the algorithm is able to recreate motifs 2.4. However, the ultimate goal is to be able to use the algorithm to visualize secondary structures of ribosomal RNAs which had not been possible before the Traveler approach was introduced. Example of this can be seen in 2.5. Another indirect confirmation of validity of the solution is that Traveler is now being used as a part of the *auto-traveler* pipeline [RNAcenral] supporting visualization of more than 7 million RNA secondary structures in RNAcentral [RNAcentral].

2.2.1 Software solution

Traveler is implemented and distributed as a command-line application for Linux systems. On its input, it accepts a target secondary structure, a template secondary structure, and a template layout and outputs the template-target mapping and target layout. The software is also proviede as a containerized version for easier distribution.

Chapter 3

Protein functional analysis

As the function of a protein directly depends on which molecules it can recognize and interact with, the techniques for detection of the interaction sites play a great role in the protein function determination.

In the previous section, we motivated our research with the growth of RNA structure databases which called for new methods for handling the newly available RNA data. But proteins have been in the center of interest long before the surge of interest in RNA. However, due to the difficulty of obtaining protein structure, the structural databases started to grow only relatively recently with the advance of protein structure determination methods. Currently, the number of available structures in the PDB allows its mining for the purpose of protein function discovery.

In the following two sections, we describe two contributions to the field of detection of functionally important regions on the protein surface. The first contribution consists in the proposal of a novel machine learning-based approach for protein-ligand binding site detection while the second is a method for detection of protein-protein interaction sites utilizing mining a database of structural motifs.

3.1 Protein-ligand binding sites discovery

Not only due to its application in rational drug discovery, plethora of protein-ligand binding site (pocket) detection techniques has been proposed in recent years. These include i) purely geometric methods, which focus on the detection of concave pockets and clefts on the surface of a 3D structure []; ii) energetic methods which aim at approximating binding energies by placing probes around the protein surface and calculating interaction energies of those probes, [],; 3) methods that make use of evolutionary conservation and are

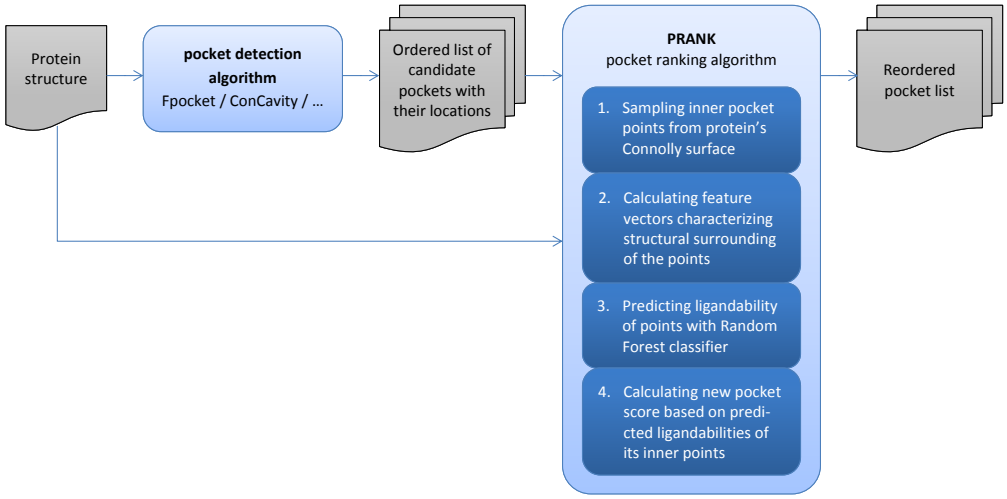


Figure 3.1: Flowchart of the pocket ranking (PRANK) approach.

thus based on the presumption that binding sites are evolutionary conserved [] and finally 4) consensus methods, which are meta approaches combining results of multiple detection methods [].

3.1.1 Protein-ligand binding sites rescoring

A pocket detection method takes a protein structure on its input and produces an ordered list of putative pockets which represent the locations on the protein surface where ligands are expected to bind. However, as these pockets are putative, not all of them represent true binding sites and the number of false positives can be substantial. Pocket ranking, therefore, plays an important role. While many ligand-binding detection approaches employ complex and inventive algorithms to locate the pockets, the final ranking is often done by a simple method such as ordering by size or scoring pockets by a linear combination of few pocket descriptors [Krivák and Hoksza, 2015b]. Therefore, we introduced a novel machine learning-based pocket rescoring algorithm called PRANK (Protein RANKing) to be used as a post-processing step which improves the performance of any pocket detection method.

Figure 3.1 outlines PRANK's four-step ranking algorithm. Detailed description can be found in [Krivák and Hoksza, 2015b], here we just briefly comment on several points which are important for the success of PRANK and also for the success of the second generation of the algorithm which is

able to carry out full-scale pocket detection (see section 3.1.2):

- The Connolly surface [Connolly, 1983], i.e. the solvent accessible surface on which the inner points are placed, is built in such a way that the points are positioned in the distance at most 4 Å from the surface. This threshold was chosen because protein-ligand binding happens in this distance.

- Atomic feature vectors (AFV), are computed for atoms which consist the putative pockets and which are in the neighborhood of any inner point. These feature vectors are then aggregated into pocket inner points feature vectors (IFV). AFVs consist of physico-chemical features derived either directly from the atom or are inherited from amino acids which given atom is part of (see the Supplementary of [Krivák and Hoksza, 2015b] for the list of used physico-chemical features). To calculate the feature vector of an inner pocket point (IFV), the AFVs from its atomic neighborhood are aggregated using a simple aggregation function and concatenated with a vector of features computed specifically for that point from its local neighborhood (e.g. the number of H-bond donors/acceptors or the protrusion index).

$$\mathrm{IFV}(P) = \sum_{\mathrm{A}_i \in\ \mathrm{A}(P)} \mathrm{AFV}\left(\mathrm{A}_i\right) \cdot w(\mathrm{dist}(P, \mathrm{A}_i)) \quad || \quad \mathrm{FV}(P)\ , \tag{3.1}$$

  where FV is the vector of the inner points-specific features and w is the distance weight function :

$$w(d) = 1 - d/8\ . \tag{3.2}$$

- When training the model, all sampled inner pocket points located within 2.5 Å from any ligand atom were labeled as positive.

- The choice of Random Forest as our machine learning technique was important due to its robustness with respect to the presence of irrelevant or correlated features which thus do not need to be scaled [Nayal and Honig, 2006] or filtered [Boulesteix et al., 2012].

- The resulting score for a pocket is computed as the squared sum of positive class probabilities of all the inner points as returned from the Random Forest classifier. We found that summing gives steadily better performance then averaging. Moreover, squaring puts more emphasis on the impact of points with probability closer to 1.

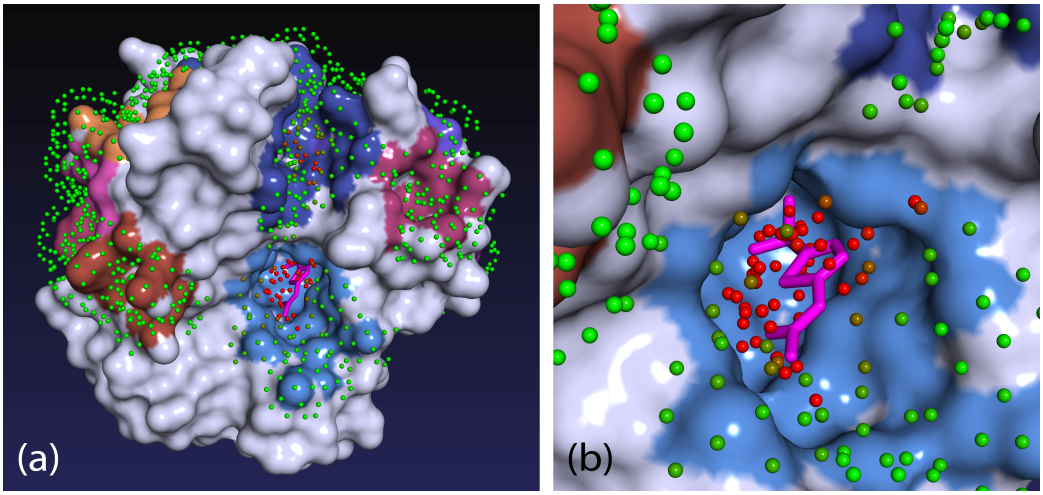


Figure 3.2: Visualization of inner pocket points. (a) Displayed is the protein 1AZM bound to one ligand (magenta). Fpocket predicted 13 pockets that are depicted as colored areas on the protein surface. To rank these pockets, the protein was first covered with evenly spaced Connolly surface points (probe radius 1.6 Å) and only the points adjacent to one of the pockets were retained. Color of the points reflects their ligandability (green = 0...red = 0.7) predicted by Random Forest classifier. PRANK algorithm rescores pockets according to the cumulative ligandability of their corresponding points. Note that there are two clusters of ligandable points in the picture, one located in the upper dark-blue pocket and the other in the light-blue pocket in the middle. The light-blue pocket, which is, in fact, the true binding site, contains more ligandable points and therefore will be ranked higher. (b) Detailed view of the binding site with the ligand and the inner pocket points.

Figure 3.2 shows how the Connolly points are distributed across the putative binding sites and illustrates how they are scored with points deep in a true pocket having, indeed, a higher score.

To evaluate our approach we first performed cross-validation experiments to attest viability of our method and then we trained our model on one (CHEN11 [Chen et al., 2011]) of the available datasets and used it to test the rest of the datasets to show generalization ability of the solution. We observed a substantial increase in the pocket identification success rate in the top N scored pockets, N being the number of true pockets in a protein. This was demonstrated with the two most commonly used pocket detection methods Fpocket [Le Guilloux et al., 2009] and ConCavity [Capra et al., 2009] (see Table 3.1).

Table 3.1: Rescoring Fpocket and ConCavity predictions with PRANK: cross-validation results on CHEN11 dataset and the results of the final prediction model (trained on CHEN11-Fpocket) for all datasets.

Dataset	Top-n [%]	Rescored [%]	All [%]	Δ	%possible*	P	R	MCC
Fpocket predictions								
CHEN11 (CV)**	47.9	58.8	71	+10.6	47.1	0.60	0.32	0.41
CHEN11 ***	47.9	67.9	71	+20	86.4	0.87	1.0	0.98
ASTEX	58	63.6	81.1	+5.6	24.2	0.56	0.41	0.46
DT198	37.5	56.2	80.2	+18.8	43.9	0.31	0.38	0.33
MP210	56.6	67.7	78.8	+11.1	50	0.58	0.42	0.47
B48	74.1	81.5	92.6	+7.4	40	0.58	0.45	0.49
U48	53.7	77.8	88.9	+24.1	68.4	0.55	0.36	0.42
ConCavity predictions								
CHEN11 (CV)**	47.9	50.7	52.3	+2.8	63.3	0.44	0.76	0.40
CHEN11 ***	47.9	52.3	52.3	+4.4	100	0.80	0.82	0.75
ASTEX	55.2	62.9	65.7	+7.7	73.3	0.60	0.55	0.46
DT198	45.8	61.5	65.6	+15.6	78.9	0.33	0.55	0.34
MP210	57.4	66.1	68.2	+8.7	80.6	0.63	0.53	0.49
B48	66.7	77.8	81.5	+11.1	75	0.61	0.53	0.47
U48	64.8	74.1	77.8	+9.3	71.4	0.58	0.46	0.43

Abbreviations: P precision, R recall, MCC Matthews correlation coefficient
* percentage of improvement that was theoretically possible to obtain by reordering pockets [Δ / (All – Top-n)]
** cross-validation results
*** results where the test set was de facto the same as the training set for the Random Forest classifier (included here only for completeness)

3.1.2 Protein-ligand binding sites detection

After successful application of PRANK to improve the ranking of the putative pockets generated by a third party pocket detection method, we decided to extend PRANK procedure to also generate pockets on its own, resulting in a self-contained pocket detection solution. We called this improved version P2RANK [Krivák and Hoksza, 2018]. The following list outlines the method [Krivák and Hoksza, 2015a]:

1. Generating a set of regularly spaced points lying on the protein's Connolly surface (referred to as *Connolly points*).
2. Calculating feature descriptors of Connolly points based on their local chemical neighborhood:
    a) computing property vectors for protein's solvent-exposed atoms,
    b) projecting distance weighted properties of the adjacent protein atoms onto Connolly points,
    c) computing additional features describing Connolly point neighborhood.

3. Predicting ligandability score of Connolly points by Random Forest classifier.

4. Clustering points with high ligandability score and thus forming pocket predictions.

5. Ranking predicted pockets by cumulative ligandability score of their points.

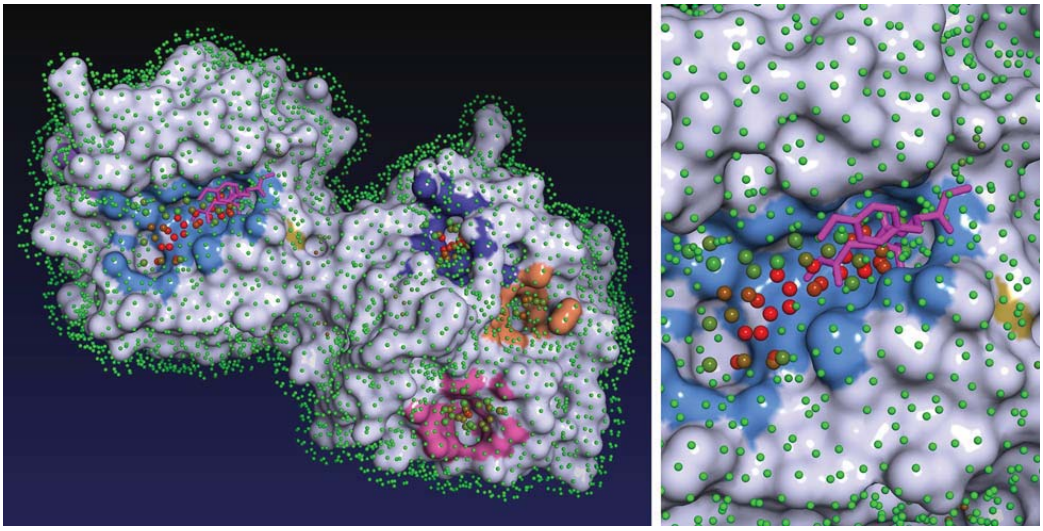


Figure 3.3: Visualization of ligand-binding sites predicted by P2Rank for a structure with PDB ID 1FBL. The protein is covered by a layer of points lying on the Solvent Accessible Surface of the protein. Each point represents its local chemical neighborhood and is colored according to its predicted ligandability score (from 0 = green to 1 = red). Points with high ligandablity score are clustered to form predicted binding sites (marked by coloring adjacent protein surface). In this case, the largest predicted pocket (shown in the close-up) is indeed the correctly predicted true binding site that binds a known ligand (magenta). Visualization is based on a PyMOL script produced by P2Rank.

The procedure builds on the PRANK algorithm, but instead of working only with putative pocket points, the full surface is covered by Connolly points (step 1) and processed. This illustrates Figure 3.3 where, unlike in Figure 3.2, the points cover the full surface of the protein. Moreover, a clustering step where points with sufficient ligandability score are joined (step 4) is added. In this step, first Connolly points that have ligandability score lower than given threshold are filtered out and single linkage clustering procedure is applied on the remaining points. Predicted pocket is then associated with the set of Connolly points in a cluster. For each pocket, we compute the set of protein solvent-exposed atoms that form the putative ligand-binding surface patch. P2RANK includes into the output all pockets that are defined by 3 or more Connolly points.

Table 3.2: Benchmark on COACH420 [Yang et al., 2013] and HOLO4K [Schmidtke et al., 2010] datasets.

	COACH420		HOLO4K	
	Top-n	Top-(n+2)	Top-n	Top-(n+2)
Fpocket 1.0	56.4	68.9	52.4	63.1
Fpocket 3.1	42.9	56.9	54.9	64.3
SiteHound* [Hernandez et al., 2009]	53.0	69.3	50.1	62.1
MetaPocket 2.0*,[Zhang et al., 2011b]	63.4	74.6	57.9	68.6
DeepSite*,[Jiménez et al., 2017]	56.4	63.4	45.6	48.2
P2Rank	72.0	**78.3**	68.6	74.0
P2Rank+Cons.†	**73.2**	77.9	**72.1**	**76.7**

Comparing identification success rate [%] measured by the DCA criterion (distance from pocket center to closest ligand atom) with 4 Å threshold considering only pockets ranked at the top of the list (n is the number of ligands in the considered structure).
*Failed to produce predictions for some of the input proteins. Here we display calculated success rates based only on those protein subsets for which the corresponding method was finished successfully.
† P2Rank with conservation (the default prediction model of PrankWeb)

The set of features associated with atoms and Connolly points are the same for PRANK and P2RANK. However, since binding sites tend to be more conserved than other parts of the molecule, later in [Jendele et al., 2019] we extended the set of features by conservation score assigned to each amino acid. This score is propagated to the atom level. To obtain the conservation, P2RANK implements a complex conservation pipeline (see the flowchart and supplementary material in [Jendele et al., 2019]). The results of P2RANK with conservation, i.e. the most recent version of the method, taken from [Jendele et al., 2019] are summarized in Table 3.2. Although we list 17 pocket detection

methods in [Jendele et al., 2019], only the five listed in the table allow for batch processing; either as a command-line application or via REST API as P2RANK does. Also, prediction speeds vary greatly between tools, ranging from under one second (Fpocket, P2Rank) to ¿10 h (COACH) for prediction on one average-sized protein (2500 atoms).

3.1.3 Software solution

P2RANK and PRANK are currently distributed as a single software solution written in Scala and thus available on all common platforms. It is distributed with a pre-trained model but includes a module allowing users to train a model from their data. With the model available, the user can submit one or more structures for which the pockets should be detected. The result includes detail statistics about the identified pockets, including which atoms were labeled as parts of which pocket and PyMOL [Schrödinger, LLC, 2015] script for visualization of the results. The P2RANK application serves as the backend to the PrankWeb [Jendele et al., 2019] web application, which provides users with a simple way how to annotate a submitted protein structure. The result page allows the users to visually inspect the putative binding sites and download detailed information about the sites, including a PyMOL script for offline visual inspection. Moreover, the web interface enables the users to contrast the putative pockets with the conservation information to further aid the interpretation of the results.

3.2 Protein-protein binding sites discovery

Alongside our efforts focused on protein-ligand detection, we also studied methods which could be used to discover protein-protein interaction (PPI) sites. Unlike in protein-ligand detection, the typical goal of PPI detection is to decide whether a residue is part of a PPI, i.e. it is a binary classification task. Note that this is a simpler task than finding the full pocket because it is missing the clustering phase present in P2RANK. Similarly to protein-ligand binding site detection, the PPI detection methods can be grouped into three, possibly intersecting, classes: evolutionary-based, template-based, and machine learning-based methods. The evolutionary-based methods utilize the co-evolution principle which is based on the observation that changes in one interaction site are compensated by changes in the opposite interaction site in order to preserve the functionality [Reš et al., 2005]. In the template-based approaches, the methods use another protein with known interaction sites which can be transferred to the protein of interest [Zhang et al., 2010, 2011a]. However, the information required by evolutionary and template-based predictors is often not available and thus machine learning methods are often utilized. Machine learning methods try to learn surface characteristics which are observed with PPI amino acids and their neighborhoods. A model is then trained to recognize the characteristics and patterns commonly exhibited by PPIs [Chen and Zhou, 2005, Zhang et al., 2010, 2011a, Zellner et al., 2012, Bendell et al., 2014, Dong et al., 2014, Wierschin et al., 2015].

The detection of PPI sites is more challenging than the detection of protein-ligand sites as the PPI sites tend to be more flat, thus more difficult to reveal. Indeed, when analyzing features which were most important for P2RANK prediction, we found out that protrusion was the most important feature. Although it was not sufficient to explain the ligand-binding site by itself it was an important indicator. This is not the case with PPIs. For that reason, we decided rather to focus on the topology of the structural neighborhood of amino acids and features of the residues in this neighborhood, which led us to the development of PPI detection method called INSPiRE (INteraction Sites PREdictor) [Jelínek et al., 2017]. INSPiRE is a knowledge-based three-step procedure: i) it extracts patterns representing local structural neighborhoods and interface/non-interface information for all the amino acids of every knowledge base protein; ii) it converts the patterns into a suitable data format for efficient storage and retrieval and iii) it labels amino acids of unknown proteins as interface or non-interface based on how often their structural neighborhood appears as interface/non-interface in the knowledge base.

Protein structures in INSPiRE are represented as graphs where nodes

correspond to amino acids. Two nodes are connected by an edge if they are at most 6 Å apart. Our knowledge base, which was built using the whole PDB, contained over 60,000 complexes with over 54 million amino acids (nodes) and almost 293 million edges. An amino acid is marked as an interface amino acid if at least one of its atoms is sufficiently close to any atom of any other chain. Moreover, each atom is assigned amino acid type and value representing the fraction of the amino acid's surface that is exposed to a solvent (RASA value). For each node N in the graph, INSPiRE extracts a subgraph, called structural element, which is induced either by nodes up to graph distance i from N (the central node) or by nodes up to Euclidean distance d from N. This subgraph is then stored in the knowledge base.

In the prediction phase, INSPiRE needs for each structural element of a query protein to find out how many similar or identical structural elements are in the knowledge base. Since the knowledge base contains close to 55 millions structural elements, we needed an efficient way to store and retrieve those elements. The problem of finding matching or similar elements translates into the NP-complete problem of subgraph isomorphism and is time demanding even for small graphs, which is our case. We considered three possibilities to solve this problem: graph-based data storage, relational data storage and molecular fingerprints stored in binary format.

As we showed in [Hoksza and Jelínek, 2015], where we tried to use Neo4j graph database for this purpose, searching for induced subgraphs of stored graphs is viable for structural elements only up to about 12 edges. However, in our knowledge base approximately 45% of nodes had more than 12 edges in the structural neighborhood with graph distance of size 1. So even for such small neighborhoods, the use of the graph database is not an option.

Using relational data storage is possible with precomputing the neighborhood information and storing it in the database [Hoksza and Jelínek, 2015]. We were able to implement this approach for radial pattern, where only the center and edges going from the center were considered. However, adding edges among the remaining nodes leads to false positives which need to be filtered out. Moreover, we found out that the filtration ratio of the database query is strongly dependent on the distribution of the employed feature types and often turned out to be quite weak.

In the end, we took inspiration from cheminformatics, specifically, we utilized molecular fingerprints which are traditionally used in virtual screening of small-molecule libraries. The basic principle of molecular fingerprints is to capture structural features of a molecular graph and encode them in a bit string which can be used later when assessing the similarity of a pair of compounds. The advantage is that such representation is highly storage-efficient, and the time-consuming operation of comparison of two molecular graphs is reduced

Table 3.3: Comparison on PlaneDimers [Zellner et al., 2012] & TransComp1 [Zellner et al., 2012] datasets in terms of MCC.

	PlaneDimers	TransComp1
INSPiRE	**0.681**	**0.529**
SPPIDER Porollo and Meller [2007]	0.330	0.150
PresCont Zellner et al. [2012]	0.330	0.170
MetaPPISP Qin and Zhou [2007]	0.040	0.311

to a highly time-efficient bit string comparison. The resulting fingerprints, i.e. the encoded structural elements, can not be used directly to identify exact matches due to the employed hashing and because more amino acids can share a feature value and thus their stored images are ambiguous. Therefore, if an exact match is required, matched fingerprints still need to be scanned for false positives. On the other hand, using fingerprints allows us to efficiently mine similar structural elements that are not exact matches. This is due to the fact that similarity of fingerprints and structural elements similarity correlate [Jelínek et al., 2017].

When using fingerprints, the knowledge base contains for each of the amino acids a bitstring representing its structural neighborhood. In the prediction phase, the query protein is translated into its graph representation and labeled with the selected features (amino acid type or RASA). Then, for each query amino acid A_Q and its neighborhood graph N_Q, the subset of the knowledge base is selected for which the value of the central amino acid is the same as for A_Q. From this filtered set, INSPiRE picks n structural elements which are most similar to N_Q. Afterward, the retrieved elements are divided based on whether their central amino acid is labeled as an interface (set I), or non-interface (set N). Finally, the probability of A_Q being interface is estimated as $|I|/|N|$.

Since only two of the PPI prediction methods were available for large scale evaluation, we chose six most often cited methods which were tested

Table 3.4: Comparison on the DS188 dataset [Zhang et al., 2010].

	MCC	Precision	Recall	ACC	F1
INSPiRE	**0.481**	**0.534**	0.567	**0.879**	**0.550**
PredUs Zhang et al. [2011a]	0.345	0.503	0.575	0.726	0.530
PrISE Jordan et al. [2012]	0.338	0.480	0.432	0.806	0.455
RAD-T Bendell et al. [2014]	0.222	0.285	**0.647**	0.652	0.355
MetaPPISP Qin and Zhou [2007]	0.262	0.490	0.267	0.811	0.346

on public datasets and evaluated INSPiRE on the same data. Because our knowledge base was built over the full PDB, when searching for similar structural elements to a query in the evaluation, all the query protein's structural elements in the knowledge base were disregarded. Table 3.3 and Table 3.4 show that INSPiRE substantially outperforms existing methods in terms of MCC. Although INSPiRE uses the full PDB as its knowledge base the removal of the query structural elements is, as we believe, sufficient to prove that INSPiRE is at least on par with the best state-of-the-art methods. Currently, we are developing a benchmarking dataset containing dissimilar time-based subsets of the PDB and preliminary results indeed confirm that INSPiRE is able to outperform existing solutions.

3.2.1 Software solution

The software solution, intended as a complex PPI framework, is currently being developed and is accessible at `https://github.com/Jelinek-J/INSPiRE`. The framework exposes individual steps of the PPI process, such as the creation of the knowledge base, features extraction or annotation of a single protein given a knowledge base.

Bibliography

Jmol: an open-source Java viewer for chemical structures in 3D. `http://www.jmol.org/`. Accessed: 2019-08-29.

David Auber, Maylis Delest, Jean-Philippe Domenger, and Serge Dulucq. Efficient drawing of rna secondary structure. *J. Graph Algorithms Appl.*, 10(2):329–351, 2006.

Calem J. Bendell, Shalon Liu, Tristan Aumentado-Armstrong, Bogdan Istrate, Paul T. Cernek, Samuel Khan, Sergiu Picioreanu, Michael Zhao, and Robert A. Murgita. Transient protein-protein interface prediction: datasets, features, algorithms, and the rad-t predictor. *BMC Bioinformatics*, 15(1): 1–12, 2014.

Helen M Berman. The protein data bank: a historical perspective. *Acta Crystallographica Section A: Foundations of Crystallography*, 64(1):88–95, 2008.

M. L. Bochman, K. Paeschke, and V. A. Zakian. DNA secondary structures: stability and function of G-quadruplex structures. *Nat. Rev. Genet.*, 13 (11):770–780, Nov 2012.

Anne-Laure Boulesteix, Silke Janitza, Jochen Kruppa, and Inke R König. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):493–507, 2012.

Sarah W Burge, Jennifer Daub, Ruth Eberhardt, John Tate, Lars Barquist, Eric P Nawrocki, Sean R Eddy, Paul P Gardner, and Alex Bateman. Rfam 11.0: 10 years of RNA families. *Nucleic acids research*, 41(D1):D226–D232, 2012.

Jamie J Cannone, Sankar Subramanian, Murray N Schnare, James R Collett, Lisa M D'Souza, Yushi Du, Brian Feng, Nan Lin, Lakshmi V Madabusi, Kirsten M Müller, Nupur Pande, Zhidi Shang, Nan Yu, and Robin R Gutell.

The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC bioinformatics*, 3:2, 2002.

John A Capra, Roman A Laskowski, Janet M Thornton, Mona Singh, and Thomas A Funkhouser. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3d structure. *PLoS computational biology*, 5(12):e1000585, 2009.

Emidio Capriotti and Marc A Marti-Renom. RNA structure alignment by a unit-vector approach. *Bioinformatics*, 24(16):i112–i118, 2008.

Emidio Capriotti and Marc A Marti-Renom. SARA: a server for function annotation of RNA structures. *Nucleic acids research*, 37(suppl_2):W260–W265, 2009.

Petr Čech, Daniel Svozil, and David Hoksza. SETTER: web server for RNA structure comparison. *Nucleic acids research*, 40(W1):W42–W48, 2012.

Petr Čech, David Hoksza, and Daniel Svozil. MultiSETTER: web server for multiple RNA structure comparison. *BMC bioinformatics*, 16(1):253, 2015.

Yen-Fu Chang, Yen-Lin Huang, and Chin Lung Lu. SARSA: a web tool for structural alignment of RNA using a structural alphabet. *Nucleic Acids Res*, 36(Web-Server-Issue):19–24, 2008a.

Yen-Fu Chang, Yen-Lin Huang, and Chin Lung Lu. SARSA: a web tool for structural alignment of RNA using a structural alphabet. *Nucleic acids research*, 36(suppl_2):W19–W24, 2008b.

Maria Chatzou, Cedrik Magis, Jia-Ming Chang, Carsten Kemena, Giovanni Bussotti, Ionas Erb, and Cedric Notredame. Multiple sequence alignment modeling: methods and applications. *Briefings in Bioinformatics*, 17(6): 1009–1023, 11 2015.

Huiling Chen and Huan-Xiang Zhou. Prediction of interface residues in protein–protein complexes by a consensus neural network method: test against NMR data. *Proteins: Structure, Function, and Bioinformatics*, 61 (1):21–35, 2005.

Ke Chen, Marcin J Mizianty, Jianzhao Gao, and Lukasz Kurgan. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. *Structure*, 19(5):613–621, 2011.

Michael L Connolly. Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221(4612):709–713, 1983.

Kévin Darty, Alain Denise, and Yann Ponty. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15):1974–1975, 2009.

Margaret O Dayhoff. Atlas of protein sequence and structure. 1965.

Abdoulaye Baniré Diallo and Wajdi Dhifli. Pgr: A novel graph repository of protein 3d-structures. *Journal of Data Mining in Genomics & Proteomics*, 6(02), 2015.

W Diniz and F Canduri. Bioinformatics: an overview and its applications. *Genet Mol Res*, 16, 2017.

Zhijie Dong, Keyu Wang, Truong Khanh Linh Dang, Mehmet Gültas, Marlon Welter, Torsten Wierschin, Mario Stanke, and Stephan Waack. CRF-based models of protein surfaces improve protein-protein interaction site predictions. *BMC bioinformatics*, 15(1):277, 2014.

O. Dror, R. Nussinov, and H. Wolfson. ARTS: alignment of RNA tertiary structures. *Bioinformatics*, 21 Suppl 2, September 2005.

Pehr Edman and Geoffrey Begg. A protein sequenator. *European Journal of Biochemistry*, 1(1):80–91, 1967.

Richard Elias and David Hoksza. TRAVeLer: a tool for template-based RNA secondary structure visualization. *BMC bioinformatics*, 18(1):487, 2017.

Fabrizio Ferrè, Yann Ponty, W. A. Lorenz, and Peter Clote. DIAL: a web server for the pairwise alignment of two RNA three-dimensional structures using nucleotide, dihedral angle and base-pairing similarities. *Nucleic Acids Res*, 35(Web-Server-Issue):659–668, 2007.

Darren R Flower, ANTHONY CT North, and Teresa K Attwood. Structure and sequence relationships in the lipocalins and related proteins. *Protein Science*, 2(5):753–761, 1993.

Jeff Gauthier, Antony T Vincent, Steve J Charette, and Nicolas Derome. A brief history of bioinformatics. *Brief. Bioinform*, 3, 2018.

Dan Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge university press, 1997.

Donna K Hendrix, Steven E Brenner, and Stephen R Holbrook. RNA structural motifs: building blocks of a modular biomolecule. *Quarterly reviews of biophysics*, 38(3):221–243, 2005.

Marylens Hernandez, Dario Ghersi, and Roberto Sanchez. SITEHOUND-web: a server for ligand binding site identification in protein structures. *Nucleic acids research*, 37(suppl_2):W413–W416, 2009.

David Hoksza and Jan Jelínek. Using neo4j for mining protein graphs: A case study. In *2015 26th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 230–234. IEEE, 2015.

David Hoksza and Daniel Svozil. Efficient RNA pairwise structure comparison by SETTER method. *Bioinformatics*, 28(14):1858–1864, 2012.

David Hoksza and Daniel Svozil. Multiple 3D RNA structure superposition using neighbor joining. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(3):520–530, 2014.

Thomas A Isenbarger, Christopher E Carr, Sarah Stewart Johnson, Michael Finney, George M Church, Walter Gilbert, Maria T Zuber, and Gary Ruvkun. The most conserved genome segments for life detection on earth and other planets. *Origins of Life and Evolution of Biospheres*, 38(6): 517–533, 2008.

Jan Jelínek, Petr Škoda, and David Hoksza. Utilizing knowledge base of amino acids structural neighborhoods to predict protein-protein interaction sites. *BMC bioinformatics*, 18(15):492, 2017.

L. Jendele, R. Krivak, P. Skoda, M. Novotny, and D. Hoksza. PrankWeb: a web server for ligand binding site prediction and visualization. *Nucleic Acids Res.*, 47(W1):W345–W349, Jul 2019.

José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, AS Rose, and Gianni De Fabritiis. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

Rafael A. Jordan, Yasser EL-Manzalawy, Drena Dobbs, and Vasant Honavar. Predicting protein-protein interface residues using local surface structural similarity. *BMC Bioinformatics*, 13(1):1–14, 2012.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

Hisanori Kiryu, Yasuo Tabei, Taishin Kin, and Kiyoshi Asai. Murlet: a practical multiple alignment tool for structural rna sequences. *Bioinformatics*, 23(13):1588–1598, 2007.

Elmar Krieger, Sander B Nabuurs, and Gert Vriend. Homology modeling. *Methods of biochemical analysis*, 44:509–524, 2003.

Radoslav Krivák and David Hoksza. P2RANK: Knowledge-Based Ligand Binding Site Prediction Using Aggregated Local Features. In *International Conference on Algorithms for Computational Biology*, pages 41–52. Springer, 2015a.

Radoslav Krivák and David Hoksza. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *Journal of cheminformatics*, 7(1):12, 2015b.

Radoslav Krivák and David Hoksza. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics*, 10(1):39, 2018.

Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10(1): 168, 2009.

Ronny Lorenz, Stephan H Bernhart, Christian zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. {ViennaRNA} Package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

Sebastien Moretti, Andreas Wilm, Desmond G Higgins, Ioannis Xenarios, and Cedric Notredame. R-Coffee: a web server for accurately aligning noncoding RNA sequences. *Nucleic acids research*, 36(suppl_2):W10–W13, 2008.

Murad Nayal and Barry Honig. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. *Proteins: Structure, Function, and Bioinformatics*, 63(4):892–906, 2006.

PDBe-KB consortium. PDBe-KB: a community-driven resource for structural and functional annotations. *Nucleic Acids Research*, 2019. Submitted.

Jonathan Pevsner. *Bioinformatics and functional genomics*. John Wiley & Sons, 2015.

Aleksey Porollo and Jarosław Meller. Prediction-based fingerprints of protein–protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 66(3):630–645, 2007.

Sanbo Qin and Huan-Xiang Zhou. meta-PPISP: a meta web server for protein-protein interaction site prediction. *Bioinformatics*, 23(24):3386–3387, 2007. . DOI: 10.1093/bioinformatics/btm434.

Ryan R. Rahrig, Neocles B. Leontis, and Craig L. Zirbel. R3D Align: global pairwise alignment of RNA 3D structures using local superpositions. *Bioinformatics*, 26(21):2689–2697, November 2010a.

Ryan R Rahrig, Neocles B Leontis, and Craig L Zirbel. R3D Align: global pairwise alignment of RNA 3D structures using local superpositions. *Bioinformatics*, 26(21):2689–2697, 2010b.

I Reš, I Mihalek, and Olivier Lichtarge. An evolution based classifier for prediction of protein interfaces without using protein structures. *Bioinformatics*, 21(10):2496–2501, 2005.

RNAcenral. Auto Traveler. `https://github.com/RNAcentral/auto-traveler`. Accessed: 2019-09-11.

RNAcentral. RNAcentral release 13. `https://blog.rnacentral.org/2019/09/rnacentral-release-13.html`. Accessed: 2019-08-29.

Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

Peter Schmidtke, Catherine Souaille, Frédéric Estienne, Nicolas Baurin, and Romano T Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of chemical information and modeling*, 50(12):2191–2200, 2010.

Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

Bruno J Strasser. Collecting, comparing, and computing sequences: the making of margaret o. dayhoff's atlas of protein sequence and structure, 1954–1965. *Journal of the History of Biology*, 43(4):623–660, 2010.

Yasuo Tabei, Hisanori Kiryu, Taishin Kin, and Kiyoshi Asai. A fast structural multiple alignment method for long RNA sequences. *BMC bioinformatics*, 9(1):33, 2008.

Makio Tamura, Donna K Hendrix, Peter S Klosterman, Nancy RB Schimmelman, Steven E Brenner, and Stephen R Holbrook. SCOR: Structural Classification of RNA, version 2.0. *Nucleic acids research*, 32(suppl_1): D182–D184, 2004.

The RNAcentral Consortium. RNAcentral: a hub of information for non-coding RNA sequences. *Nucleic Acids Research*, 47(D1):D221–D229, 11 2018.

Julie D Thompson, Desmond G Higgins, and Toby J Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

Elfar Torarinsson, Jakob H Havgaard, and Jan Gorodkin. Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, 23(8):926–932, 2007.

Chih-Wei Wang, Kun-Tze Chen, and Chin Lung Lu. iPARTS: an improved tool of pairwise alignment of RNA tertiary structures. *Nucleic Acids Res*, 38 Suppl:W340–7, 2010.

Torsten Wierschin, Keyu Wang, Marlon Welter, Stephan Waack, and Mario Stanke. Combining features in a graphical model to predict protein binding sites. *Proteins: Structure, Function, and Bioinformatics*, 83(5):844–852, 2015.

Kay C Wiese, Edward Glen, and Anna Vasudevan. jViz. Rna-A Java tool for RNA secondary structure visualization. *IEEE transactions on nanobioscience*, 4(3):212–218, 2005.

Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

Hermann Zellner, Martin Staudigel, Thomas Trenner, Meik Bittkowski, Vincent Wolowski, Christian Icking, and Rainer Merkl. Prescont: Predicting protein-protein interfaces utilizing four residue properties. *Proteins: Structure, Function, and Bioinformatics*, 80(1):154–168, 2012.

Qiangfeng Cliff Zhang, Donald Petrey, Raquel Norel, and Barry H. Honig. Protein interface conservation across structure space. *Proceedings of the National Academy of Sciences*, 107(24):10896–10901, 2010. . DOI: 10.1073/pnas.1005894107.

Qiangfeng Cliff Zhang, Lei Deng, Markus Fisher, Jihong Guan, Barry Honig, and Donald Petrey. PredUs: a web server for predicting protein interfaces using structural neighbors. In *NAR*, 2011a.

Zengming Zhang, Yu Li, Biaoyang Lin, Michael Schroeder, and Bingding Huang. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. *Bioinformatics*, 27 (15):2083–2088, 2011b.

Part II

Publications