Internal Report CS Bioinformatics Track 13-01 March 2013

Leiden University

Computer Science

Bioinformatics Track

Prediction of Protein Three-Dimensional Structure

Dimitar Kolev

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Table of Contents

Introduction

Illnesses related to bacteria and viruses are the most common. But there is a third category of illness. Those are the ones related to the malfunction of our own bio-molecular mechanisms. Diseases like Creutzfeldt-Jakob disease, bovine spongiform encephalopathy (mad cow disease), Alzheimer's disease[18] Huntington's disease[19], Parkinson's disease[19], and others are considered to be caused by internally produced misshapen, or otherwise called misfolded, proteins that interact, with the surrounding elements, in unexpected and often negative ways. The accumulation of these misfolded molecules can lead to cell damage and in a large percentage of the cases to cell death.

There is a clear need for understanding the processes that govern protein folding. Such knowledge may one day help us design better treatments or even cures for the diseases caused by improperly folded proteins. Furthermore, the medical sector is not the only one who stands to benefit. Bioengineering of microbes in order to make them more efficient in producing desired chemicals or completely new ones can present lucrative business opportunities. We can easily imagine altering the cell's signalling pathways of bacteria and fungi to enable them to create antibiotics, to fight viruses inside the human body, to create biofuels, even to gather natural resources.

Despite the big interest in solving the process of protein folding, an already several decades old problem, a solution has not yet been found. Calculating all possible foldings for the amino acid sequence of a protein, until the right three-dimensional structure is reached, is slow and impractical as it would require a lot of computational power as the problem is known to be NP-complete. Of course, being NP-complete makes it a perfect candidate for solving, it in the future, through the use of a quantum computer.

There is a clear need for devising heuristic algorithms. The CASP initiative tracks the progress of such methods. The CASP9 experiment conducted two years ago showed that despite making progress the state of the art predictors, such as Quark, Rosetta, I-TASSER, are still largely unreliable and inaccurate in their predictions. The most current CASP experiment, CASP10, is in its final stage and will soon give us a more current outlook on the progress that has been made in the past two years.

In this paper we will describe a novel approach to solving the protein folding problem. The method is based on algorithm trained on experimentally obtained three-dimensional protein structures where higher weight is given to the three-dimensional structures experimentally obtained by X-Ray Crystallography. This is supported by research[7][8] suggesting that this method produces more accurate results compared to, for example, protein nuclear resonance spectroscopy (Protein NMR) [7]. Furthermore, our algorithm gives several different foldings each representing the native conformation of the protein depending on the host organisms [9][10]. In order to benchmark our approach we have created a framework in order to compare our own method with two of the top protein folding toolkits employing the scoring algorithm used by CASP.

This paper is organised as follows. Chapter 1 gives an introduction to the protein molecule, its amino acid structure, as well as the secondary, tertiary, and quaternary structures. Furthermore, we delve into the chemical composition of amino acids, the experimental ways of obtaining the native conformations of proteins, and parallel to this we discuss the current work in computational prediction.

In Chapter 2 we describe our own approach to the prediction of the tertiary structure of proteins, as well as our reasoning and observations is given. The following Chapter 3 contains information about our protein folding prediction framework, as well as the scoring algorithm used. We also go into more detail about the two third-party protein folding toolkits explaining their algorithms and past CASP performances.

The final chapter, Chapter 4, contains our experimental setup and the comparison between results obtained by our algorithm and ones obtained by the two-third party tool-chains. We have categorised our experiments into two groups:

1 Using our algorithm to predict the native conformation of the proteins given by CASP8 and CASP9. Here we compare the our method with the CASP8 and CASP9 ground-truth.
2 We have used our two third-party prediction toolkits on CASP8 and CASP9 and the results have been compared with our own.

The end of the paper is marked by the summary and the appendices.

Chapter 1 Proteins

Proteins are biochemical compounds made of one or more polypeptides. Each polypeptide represents a separate domain and can exist on its own[37]. Different proteins can be comprised from the same domain type (See Figure 1). Each domain has its own amino acid sequence which dictates the shape of the three-dimensional structure it can fold into. The correctly folded domains of a protein are referred to as the protein's native conformation.

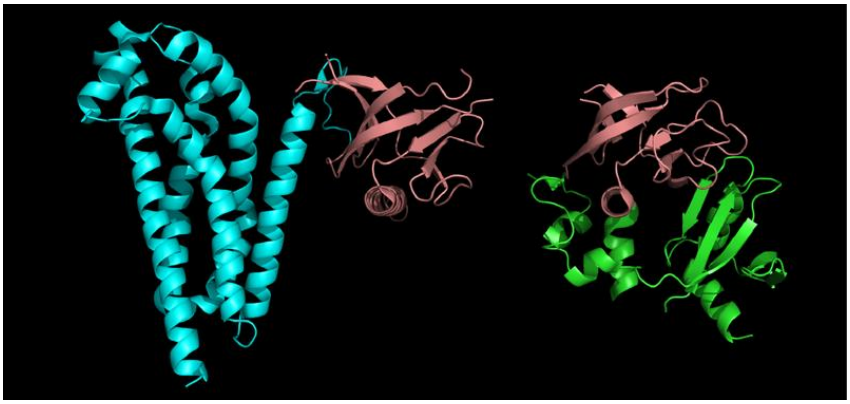


Figure 1 Protein domains. The two shown protein structures share a common domain (maroon), the PH domain, which is involved in in phosphatidyl-inositol triphosphate binding (source: Fdardel, 2011)

The amino acid sequence of a protein domain is encoded by a specific gene, thus multi-domain proteins are defined by more than one gene[37]. Through the processes of gene expression the genetic code of a gene is first transcribed into ribonucleic acid (RNA) and later translated into the amino acid sequence of a specific domain.

The amino acid sequence of the domain begins to fold before the translation step has completed. In most cases this is not a problem as the forces which govern the folding process can later on correct any misfolding. But when it comes to larger amino acid sequences the misfoldings can persist and lead to the inability of the protein to reach its native conformation. In the latter case special proteins called molecular chaperones[38] are needed to unfold the domain and guide it to its correct three-dimensional shape.

Before the domain has reached its native conformation the secondary structures have been formed. With the help of intermolecular forces and sometimes chaperon intervention (mostly for large proteins) the correct three-dimensional shape is reached. In the case of single-domain proteins this is the final stage of folding for the protein, but if we consider multi-domain ones every domain has to be folded and the folded domains have to stumble onto each other in order for them to be attracted and attach to one another.

Special sites, called binding sites (or active sites), are formed on the surface of proteins (See Figure 2). Through them a protein can interact with other molecules to performs specific functions. The shape and atomic composition of the binding sites dictates what other molecules can bind to it. It can be clearly seen why it is important for the correct native conformation to be reached as otherwise the correct binding sites will not be formed and the protein will not perform its function.

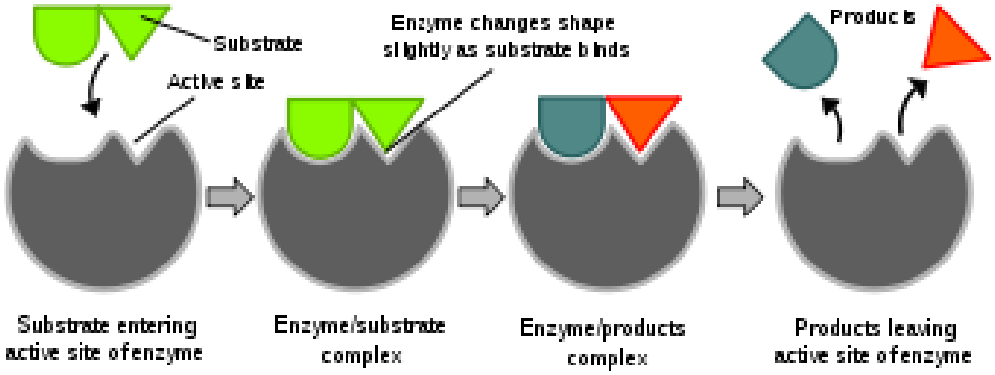


Figure 2 Induced fit hypothesis of enzyme action. (source: Vickers, 2006)

Amino Acids

Amino acids are the building blocks of proteins. They are composed from the four chemical elements nitrogen, oxygen, hydrogen, and carbon. There are around 500 amino acids from which 22 participate in the assembly of proteins. Twenty of them are naturally occurring, and the other two are formed inside the protein molecules themself. From here on when we talk about amino acids we will refer to the above mentioned 22. Nine of the amino acids are classified as “essential” for humans as they need to be ingested through our diet because our organism cannot produce them. The others are conditionally essential, depending on age or medical condition. The length of the amino acid sequence of a protein can vary from a few dozen to several thousand residues. These 22 amino acids have the same general structure, a backbone and a side chain, represented as R (See Figure 3). The side-chain of each is different whereas the backbone is the same.

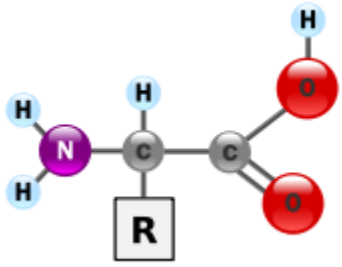


Figure 3 General structure of amino acids
(source: YassineMrabet, 2007)

The middle carbon atom is called the alpha-carbon. Amino acids chain to each other through their nitrogen atom and right-most carbon atom. Figure 4 depicts the process, through which amino acids link to form chains, called polymerization. The process starts when amino acids are

attached to transfer RNA (tRNA) molecules [32] and are chained together by the ribosomal complex in the direction from N-terminus to C-terminus.

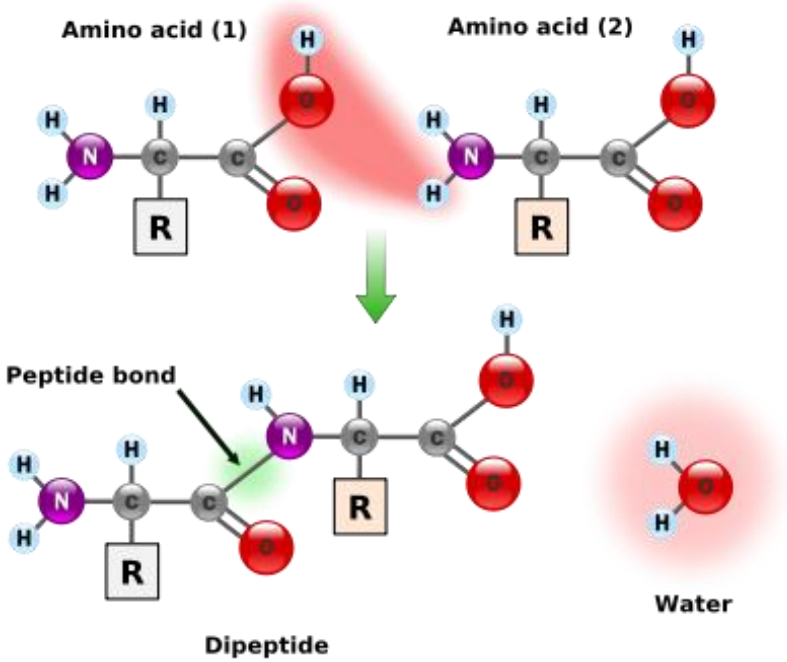


Figure 4 The condensation of two amino acids to form a peptide bond
(source: YassineMrabet, 2007)

Types of protein structures

When considering protein molecules we can observe, given single- or multi-domain polymers, three to four main structural organizations respectively (See Figure 5). They are referred to as the primary, secondary, tertiary, and quaternary structure.

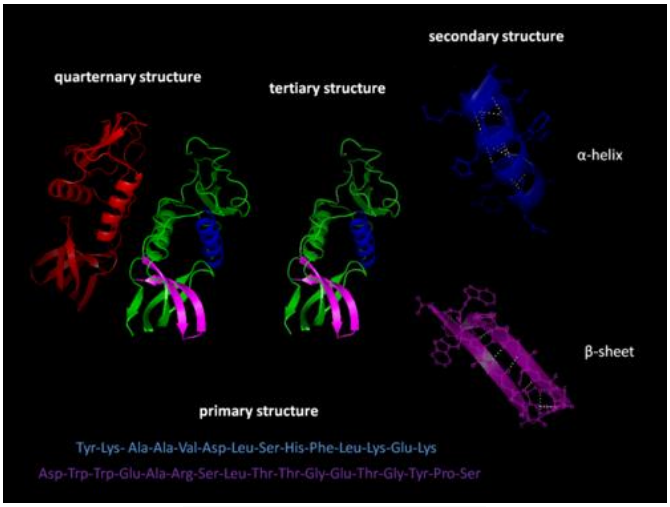


Figure 5 Protein structure types
(source: Holger87, 2012)

The primary structure is the amino acid sequence of the protein. After a gene has been transcribed, RNA splicing performed, and translation carried out we are left with an unfolded, or partly folded, strip of chained amino acids ready to begin proper folding. These residues are held by covalent or peptide bonds. The strip has two ends referred to as the carboxyl terminus (C-terminus) and the amino terminus (N-terminus).

After the primary structure has been assembled the protein can start to fold to its proper native conformation. Often what is first folded are the secondary structures. These are commonly occurring local amino acid motifs and any number of each can be present. The secondary structures are stabilized by hydrogen bonds and have regular geometry. They can be further subcategorized as 3-turn helix[2], 4-turn helix[2], Pi helix[2], hydrogen bonded turn, parallel or antiparallel beta sheet conformation[2], single pair beta sheet[2], bend[2], and coil[2].

The tertiary structure is the overall global shape of a protein, its three-dimensional atomic composition. It represents the spatial relationship between the secondary structures. It is stabilized by non-specific hydrophobic interactions, salt bridges[35], hydrogen bonds[36], and disulfide bonds[34]. In the case of a single domain protein this structure is regarded as its native conformation.

The quaternary structure represents the three-dimensional shape of the individual domains comprising the protein, including their inter-domain connectivity. This shape only applies to multi-domain proteins and is referred to as their native conformation.

Protein folding

Protein folding is the process in which the amino acid sequence, for a protein molecule, is assembled into the protein's three-dimensional structure[3] (See Figure 6). Because the folded shape is dictated by the amino acid sequence of the polypeptide[4] the native conformation is unique for each protein. Nonetheless closely related polymers do not necessarily have similar native conformations[6] as the process depends also on external factors.

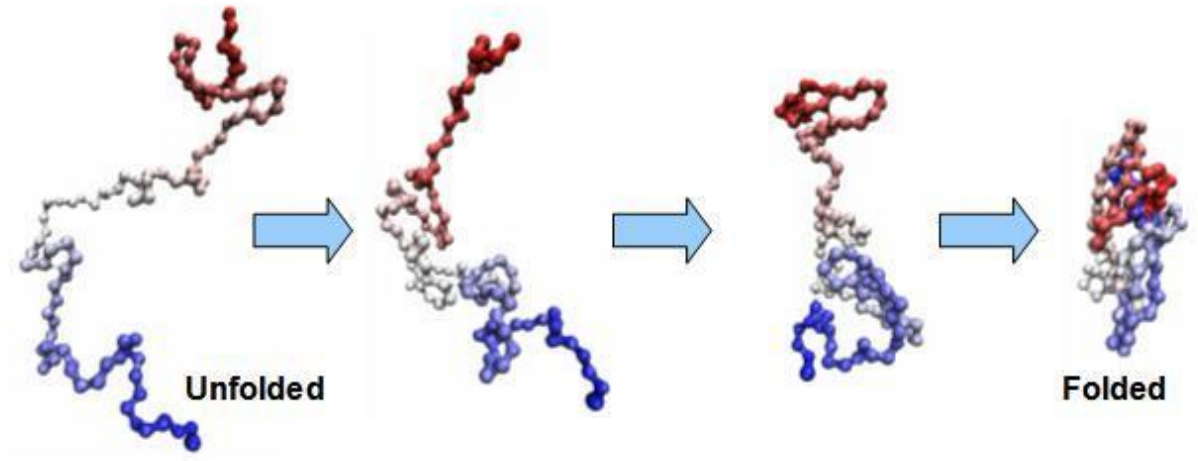


Figure 6 From unfolded amino acid sequence to properly folded 3d structure.
(source: Public Domain)

During its folding stage molecules of the same kind follow more or less the same route and go through the same intermediates and transition states. There are two experimentally confirmed paths that proteins can take during folding. There are proteins which prefer one path over the other and there are some that can fold by following either.

1. The diffusion collision model is when the nucleus is first formed, then secondary structures are folded, and the result is tightly packed to form the native conformation.
2. The nucleation-condensation model, in which the secondary and tertiary structures fold at the same time.

The folding process depends on the solvent[5], the concentration of salts, the temperature of the system, and the presence of molecular chaperones. These chaperones repair the incorrectly folded proteins and guide the proper folding of others. They are found in prokaryotes, in the cytosol of eukaryotes, and in mitochondria. Not all organisms, or cells for that matter, have the same chaperones. Furthermore, different chaperones are present in different locations of the cell and many of them are heat shock proteins, meaning that they are highly expressed in response to heightened temperatures, as protein folding is negatively affected by it.

Protein 3D Structure Determination by Experiment

In order to experimentally determine the native conformation of a protein, using most current methods, the protein first needs to be isolated and put into an environment where it can be traced. Usually the gene(s) coding for it is genetically modified, with a special marker appended to its end, so that it can be traced. There are several ways to experimentally obtain the three-dimensional structure of proteins. Protein nuclear magnetic resonance spectroscopy (Protein NMR) [39] and X-Ray crystallography[40] are the most used ones. Others are dual polarisation interferometry[41] and vibrational circular dichroism of proteins[42]. In this section we will focus on the first two in greater detail.

Nuclear magnetic resonance spectroscopy

Protein NMR is a routinely used technique for studying the natural conformation of proteins. It is used extensively by laboratories and research institutions. Several specialized phases are followed. The protein in question is prepared, resonances are assigned, restraints are generated, and the three-dimensional structure is calculated and validated.

During the sample preparation phase the target protein is isolated from its host environment, a concentration between 0.1 - 3 millimolar of it is placed in an aqueous environment of 300 to 600 microliters. The data collection step involves the use of multidimensional nuclear magnetic resonance experiments to obtain information about its atomic configuration. It is based on the assumption that in an ideal case each individual nucleus in the molecule will emit a distinct signal. Performing a multidimensional experiment is preferred because in reality there can typically be several thousand nucleuses and a one-dimensional experiment will inevitable have

overlaps in the detected resonances. Furthermore, there is considerable noise pollution thus it can take hours or even days to carry out a single experiment in order to obtain a suitable-signal-to-noise, ration through signal averaging.

The resonance assignment is important as we need to distinguish which nucleus resonance frequency corresponds to which atom. This is typically achieved by utilizing information derived from several different types of NMR experiments. As a next step restraints are applied in order for the structure of the protein to be calculated. There are several different forms of restraints such as distance, angle, and orientation. The programs CYANA, ARIA, and UNIO are used to calculate the fold, using the NMR information and restraints.

The typical mass limit of the proteins that can be passed through PNMR is ~35 kg/mol but through the employment of new techniques it has become possible to extend this limit to ~900 kg/mol[22][23], meaning that this technique could possibly be used on large sized proteins. The reliability of the experimentally determined structures decreases with an increased molecular size because there is less time to detect the individual signals. Usually proteins of size ~20 kg/mol can have their three-dimensional structures reliably determined as there will be less atoms thus less resonance overlap.

X-Ray Crystallography

This method is used to determine the arrangement of atoms within a crystal. A beam of X-rays is shot at the crystal which causes its light to spread into specific directions. From the angles and intensities of the diffracted beams, a three-dimensional picture of the density of the electrons can be produced. From the electron density, the mean positions of the atoms in the crystal, their chemical bonds, and other information, can be determined.

There are three steps concerning this method. The first step is usually the most difficult one as it involves obtaining a good enough crystal for the target molecule. It has to be larger than 0.1 mm in all dimensions and regular in structure, with no significant imperfections. Next, the crystal is mounted and X-rays are shone on it. While rotating it new reflections are created and the intensity of every atom is extrapolated. In the final step the data is combined with chemical information, using a computer aided method, to produce a model representing the atomic superpositions.

As long as a pure regular crystal can be obtained or created, as is the usual case with proteins, the superposition of its atoms can be determined with a high degree of accuracy. Sometimes techniques which can improve the crystal structures, such as macromolecular crystal annealing [25], can be used.

If the molecule that is being targeted is too large the data becomes fuzzy and less pronounced. We can consider two cases of X-ray crystallography depending on the number of atoms.

1 Small-molecule crystallography involves fewer than 100 atoms. The crystal structure in this case is very well preserved and each atom is seen as a globe.

2 Macromolecular crystallography involves crystals with tens of thousands of atoms and is generally less well-resolved. Nonetheless, this technique has been used successfully on viruses made of hundreds of thousands of atoms. In this case the quality of the result is quite low.

Others

Vibrational circular dichroism

Vibrational circular dichroism (VCD) extends circular dichroism spectroscopy to the infrared and near infrared spectrum. It can extrapolate the three-dimensional structure of a target molecule. Its application in determining the native conformation of a protein has been rare but there have been some instances[27][28]. To our knowledge there is no study that compares the results of VCD against well-established techniques such as X-Ray Crystallography or Protein NMR. It has been reported that experimental results have been obtained within the carbon-hydrogen (C-H) region of 23 amino acids and that the technique is suitable for large molecules[29].

Dual polarization interferometry

Dual polarization interferometry (DPI)[26] is a technique that can be used to observe the conformational changes of protein molecules during their lifetime. This technique is not used to determine the precise position of atoms within a structure, but rather to study the biochemical interactions between proteins, to measure reaction rates, and thermodynamics. The technique is not widely spread and in 2011 it was announced that the product, which utilized it, will be discontinued.

Comparison

According to a paper published by the MIT department of Biology “X-ray vs. NMR structures as templates for computational protein design”[7], when using a template based method for computationally predicting the three-dimensional structure of proteins, templates obtained from experimentally determined structures by X-Ray crystallography generate more accurate results compared to Protein NMR. This suggests that the experimentally obtained native conformations of proteins using X-Ray crystallography are more accurate than if they are obtained using NMR. This notion is further supported by another paper “Discrepancies between the NMR and X-ray Structures of Uncomplexed Barstar”[8] which states that “The packing densities of Protein structures determined by NMR are unreliable”.

Computationally predicting proteins’ native conformations

There are around twenty thousand proteins in the human body alone. Experimentally predicting this amount of protein native conformations is costly and slow. This becomes even more apparent if we also consider proteins from other organisms. Being able to computationally predict the native conformation would be very beneficial, not only for the pharmaceutical industry but also for the biotechnology sector, by creating disease resistant crops, biofuels, vitamin and antibiotic producing organisms, harvesting organisms (petroleum recovery), and

etc. Based on the predicted protein structure we can categorise computational predictions into three main structural categories: secondary, tertiary, and quaternary.

Protein secondary structure prediction

Computational methods for predicting the secondary structure of proteins base their calculations on the amino acid sequence of the target molecule. The scoring of those algorithms is often based on the results of the DSSP[44] method applied to the crystal structure of the protein. DSSP is an algorithm for labelling secondary structures in an already experimentally determined native conformation.

There have been several high accuracy algorithms such as the Chou-Fasman method, the GOR method, machine learning in the form of neural networks and support vector machines. Furthermore, it has been proven that external factors play a key role in the formation of the secondary structures, such as the local environment[9], solvent accessibility of residues[10], the protein structural class[11], and the expression system[12]. By taking these factors into consideration the accuracy of the predictors can be improved significantly. The best method for protein secondary structure prediction, called JPred (See Table 1), so far achieves 80% accuracy.

There are two main initiatives with the goal of benchmarking the current progress of secondary structure predictors - LiveBench [30] and EVA [31]. Currently LiveBench is down and probably will not get back up as it has lost funding. EVA is updated weekly and results can be obtained from their website. Table 1 represents the latest ranking of the predictors.

method	Q3	ERRsigQ3	sov	ERRsigsov	info	ERRsiginfo	class	ERRsigclass
JPred	84.5	+/-10.0	79.1	+/-10.0	0.38	+/-10.00	100.0	+/-10.0
PHD	75.5	+/-10.0	74.0	+/-10.0	0.34	+/-10.00	100.0	+/-10.0
PHDpsi	82.7	+/-10.0	84.1	+/-10.0	0.40	+/-10.00	0.0	+/-10.0
PROF_king	80.9	+/-10.0	87.0	+/-10.0	0.40	+/-10.00	0.0	+/-10.0
PROFsec	80.9	+/-10.0	88.3	+/-10.0	0.40	+/-10.00	0.0	+/-10.0
Prospect	70.9	+/-10.0	63.7	+/-10.0	0.22	+/-10.00	0.0	+/-10.0
PSIpred	70.9	+/-10.0	63.7	+/-10.0	0.22	+/-10.00	0.0	+/-10.0
SAM-T99sec	78.2	+/-10.0	74.7	+/-10.0	0.25	+/-10.00	0.0	+/-10.0
SSpro2	79.1	+/-10.0	86.8	+/-10.0	0.34	+/-10.00	0.0	+/-10.0

Table 1 Secondary structure predictors. **Q3** is the per-residue accuracy score. **ERRsigQ3** represents the deviation, **sov** is the per-segment accuracy, **class** is the correctness of predicting the secondary structure class according to DSSP.

Protein tertiary structure prediction

Predicting the protein's native conformation from its amino acid sequence is a very complex and still largely unresolved problem. There are a few paths that can be taken when it comes to these prediction methods, the so called ab-initio methods and comparative modelling methods.

Ab-initio methods calculate folds based only on the given amino acid sequence without considering previously experimentally solved homologs. Some of them try to mimic the protein folding process and others apply stochastic algorithms such as calculating the global minimum energy function. These methods are computationally expensive and can be used only with powerful servers like Blue Gene[45], MDGRAPE-3[46], Folding@home[47], and others. Despite the time constraint and computational power needed, this field is very active as the benefits to be gained are worth the research and the computational power. Currently, according to the latest CASP experiment some of the best ab-initio methods are QUARK, Zhang-Server, and human groups that manually adjust the results such as the ProQ2[52], Zhang-IRU[56], keasar.

Comparative protein modelling is based on previously solved native conformations or templates. The best template based methods, according to the latest CASP experiment, are the Rosetta, TASSER, and human groups that manually adjust the results such as the baker lab[53], Kloczkowski lab[54], CNIO[55]. These techniques can be further split into two groups[9].

- Homology modelling is based on the assumption that homologous proteins will have similar native conformations as they will have similar primary structure. It has been suggested that the drawback to this technique is the sequence alignment rather than the predictor itself[13]. The reason for this is that searching for homologous sequences is not always straightforward. Tools, such as BLAST[48], FASTA[49], and Modeller[50], can provide different homologous matches over the same set of sequences leading to different results in the predicted native conformations.
- Protein threading is based on comparing the amino acid sequence of the target to a database of solved structures. A scoring function is used to determine if the known structure can be applied and if so a possible folding is provided.

Figure 7 and Table 2, located in the next section, show the top ten protein tertiary structure predictors from CASP8 and CASP9 respectively. It can be noted that in the two years between the experiments, the Rosetta server, which is considered one of the best, has fallen from 5th place to 11th. The QUARK server, which is an offshoot of the TASSER server previously placed 3rd, has taken the lead. Furthermore, the TASSER server is no longer in the top 10. Considerable progress has been made in the protein tertiary structure prediction field in the past four years but still there is a lot to be desired. Two years have passed since CASP9 concluded, and CASP10 is currently being finalized. Soon we will have a more clear view of which method at this moment can be considered the most accurate.

CASP

Critical Assessment of protein Structure Prediction (CASP) is an initiative with the goal of helping advance the methods of protein structure prediction based on amino acid sequence. They provide an objective platform for judging and benchmarking the performance of protein structure predictors. The testing is done as a blind method where groups can register and submit their predicted protein structures and independent judges will grade them, through visual inspection, based on predetermined criteria. The predictions are carried out on proteins which do not have a publicly available experimentally determined three-dimensional structure but will have one in the near future. Nine complete CASP experiments have been carried out with the tenth one near completion. Experiments are conducted two years apart from each other with the first one in 1994.

There are seven questions that CASP tries to answer after the conclusion of each of its experiments. It has to be noted that the answer to question one is the goal of every predictor and the rest of the questions are there to help guide future predictors and pinpoint in which field more research is needed. The questions are presented as they are on the website of CASP[51].

1. Are the models produced similar to the corresponding experimental structure?
2. Is the mapping of the target sequence onto the proposed structure (i.e. the alignment) correct?
3. Have similar structures that a model can be based on been identified?
4. Are comparative models more accurate than can be obtained by simply copying the best template?
5. Has there been progress from the earlier CASPs?
6. What methods are most effective?
7. Where can future effort be most productively focused?

The groups participating in a CASP experiment are periodically given amino acid sequences for a few proteins, referred to as targets, for which they have to predict the native conformation. Usually around 100 proteins are targeted for each CASP experiment. Each group can submit up to several candidate models for each target.

There are three types of groups.

1. Human
2. Server or fully automatic
3. Human/Server or semi-automatic

The difference between the first two groups is that the server ones must be fully automated. Meaning that a human interaction must not occur after a submission to the server. In the first and last case the groups are allowed to refine the models by any means possible such as visual inspection and educated guesses as to what the atomic spatial conformation of the polymer should be. At the end all groups have to predict all targets. The distinction is there to give an indication how good each group is performing against the same type of predictors in addition to their overall ranking. Additionally, the above mentioned separations can be further subcategorized based on ab-initio and template based protein modelling.

The scoring algorithms used are refined during each experiment. In the latest one four scores, GDT, Contact Scores, TenS, and QCS, have been used and at the end combined into one. A more comprehensive overview can be grasped from the official website[20].

1 GDT is a score which represents the distance from one model to another. Usually it is represented from 0 to 1 with 1 indicating that the two are identical. Or in other words “It measures the fraction of residues in a model within a certain distance from the same residues in the structure after a superposition”.
2 Contact Scores (CS or TR) are scores calculated from a comparison of intramolecular distances within a given model. The difference between GDT and CS is that the first are intermolecular distances based on superposition and the later depend on a rewarding system where an atom, which is close to its original place, is rewarded depending on the accuracy of the predicted distance and penalized if it is too close to other atoms. This introduces a system where errors are taken into account as well as successes.
3 TenS is an automatically generated score which uses “six different structural measures (GDT, intra-molecular distance, Dali, TM, Mammoth and SOV) and four alignment scores (Qlga, QDali, QTM, and Qmammoth)” [20].
4 QCS is a score based on manual assessment where judges grade the models, in a blind study, according to a visual inspection.
5 Ratio Score is derived from the top four scoring categories.

Figure 7 represents the official results from CASP8. It shows that even if individually used the TS, TR, and CS scores rank the groups almost identically. There is no official preference as to which score to sort by or which is the most accurate.

Sorted by TS score

#	GROUP	TS ↓	TR	CS
		SUM		
1	Zhang-Server	9318.25	8297.62	9712.55
2	RAPTOR	9000.08	7977.42	9259.29
3	pro-sp3-TASSER	8964.02	7959.25	9190.89
4	Phyre_de_novo	8909.48	7965.76	9149.64
5	BAKER-ROBETTA	8826.10	7878.24	9198.23
6	METATASSER	8823.90	7875.96	9062.31
7	MULTICOM-CLUSTER	8797.18	7910.21	8998.57
8	MULTICOM-REFINE	8764.51	7854.40	8965.33
9	MUProt	8757.33	7842.98	8990.42
10	GS-KudlatyPred	8705.50	7727.30	8795.92

Sorted by TR score

#	GROUP	TS	TR ↓	CS
		SUM		
1	Zhang-Server	9318.25	8297.62	9712.55
2	RAPTOR	9000.08	7977.42	9259.29
3	Phyre_de_novo	8909.48	7965.76	9149.64
4	pro-sp3-TASSER	8964.02	7959.25	9190.89
5	MULTICOM-CLUSTER	8797.18	7910.21	8998.57
6	BAKER-ROBETTA	8826.10	7878.24	9198.23
7	METATASSER	8823.90	7875.96	9062.31
8	MULTICOM-REFINE	8764.51	7854.40	8965.33
9	MUProt	8757.33	7842.98	8990.42
10	HHpred5	8639.65	7790.63	8689.33

Sorted by CS score

#	GROUP	TS	TR	CS ↓
		SUM		
1	Zhang-Server	9318.25	8297.62	9712.55
2	RAPTOR	9000.08	7977.42	9259.29
3	BAKER-ROBETTA	8826.10	7878.24	9198.23
4	pro-sp3-TASSER	8964.02	7959.25	9190.89
5	Phyre_de_novo	8909.48	7965.76	9149.64
6	SAM-T08-server	8607.36	7627.56	9072.22
7	METATASSER	8823.90	7875.96	9062.31
8	MULTICOM-CLUSTER	8797.18	7910.21	8998.57
9	MUProt	8757.33	7842.98	8990.42
10	MULTICOM-REFINE	8764.51	7854.40	8965.33

Figure 7 Server rankings on all targets in domains for three scores. On all 143 domains, ranking does not change much with score, illustrating that 1) scores correlate with each other and 2) the ranking is robust. (source: www.predictioncenter.org), BAKER-ROBETTA is the Rosetta predictor.

The top eleven groups from the CASP9 experiment can be seen in Table 2.

#	GR name	SUM Z-score (GDT_TS)	AVG Z-score (GDT_TS)	AVG GDT_TS	AVG GDT_HA	AVG Mammoth (Z-Score)	AVG Dali (Z-Score)	AVG response time, min
1.	QUARK	115.788	0.788	62.675	45.669	16.998	14.843	3358.736
2.	Zhang-Server	113.242	0.770	62.765	45.772	17.127	14.650	3347.378
3.	RaptorX-MSA	103.270	0.703	61.774	44.942	17.018	15.090	3586.239
4.	RaptorX	103.010	0.701	61.731	44.671	17.029	14.814	3587.406
5.	RaptorX-Boost	99.845	0.679	61.453	44.223	17.047	14.729	3587.241
6.	HHpredB	93.104	0.633	59.528	44.013	15.907	14.317	4.334
7.	HHpredA	93.104	0.633	59.528	44.013	15.907	14.163	4.405
8.	HHpredC	91.821	0.625	59.361	43.899	15.867	14.276	4.398
9.	Seok-server	89.542	0.609	60.158	43.936	16.069	14.363	3735.850
10.	MULTICOM-CLUSTER	88.944	0.605	59.987	43.461	16.294	14.376	1030.446
11.	BAKER-ROSETTASERVER	87.240	0.602	58.768	42.552	16.139	13.914	3518.860

Table 2 CASP 9 top 10 protein tertiary structure predictors

During the CASP9 experiment groups that have produced good results, have been the Rosetta server, the Quark server (based on the I-TASSER server), the I-TASSER server, HHPred, and others with Rosetta taking 11th place [21]. When looking at the overall results the groups have not benefited to any high degree from manually adjusting their models. Meaning that fully automated servers have been performing as good, and even better in the case of some servers than human and human/server groups. This indicates that at present there is not much to be gained from human intervention when it comes to the results of the top automated predictors.

Protein Data Bank

The Protein Data Bank (PDB) is an initiative with the goal of gathering in one place experimentally determined structures of proteins, nucleic acids, and complex assemblies. It has the biggest protein database, which is actively maintained, with new releases added every day. The uploaded data has to conform to the pdb file standard which can be found on their website[17]. Its unified data format, and large sample pool makes it attractive and as such most of the predictors use its content to train and validate their internal algorithms. Following their example, we have also selected our train and test cases from it.

Chapter 2 LiacsFold

LiacsFold is an algorithm for automatically predicting the tertiary structure of proteins from their amino acid sequences. This method uses information obtained from experimentally determined protein native conformations in order to construct the native conformation of a target amino acid sequence. Its purpose is to prove or disprove a hypothesis based on the research papers described below [5][33][43][7][8].

In the paper “Macromolecular crowding perturbs protein refolding kinetics: implications for folding inside the cell"[5] it was shown that the solvent and the temperature of the system were important external factors in the protein folding process, whereas the paper “The role of molecular chaperones in protein folding”[33] points out that the molecular chaperones are a big internal factor. The underlying algorithm of our predictor is based on these research papers as well as the physical forces that govern inter amino acid interactions.

Molecular chaperones are very important when it comes to protein folding. Under normal conditions most proteins are capable of properly folding themselves. This is true for small sized and normal sized proteins but larger compounds have a higher risk of misfolding, which is something that chaperones correct. Different organisms have different chaperones, different chaperones help different types or proteins, and populate different parts of the organism. Some chaperones, called heat shock proteins, are expressed in higher quantities when the organism is under stress from external factors such as heat[43].

In the case of using a different organism as the expression system we run into a situation where it is possible that a chaperon, needed for the protein to be properly folded, is not present[5]. This situation is something that our algorithm is trying to address by explicitly giving the choice of the expression system and calculating what the protein native conformation would be in that expression system. Of course, at present there is no way to confirm if the calculated native conformation is correct, if we do not have it already experimentally determined but this is true for any predictor at present.

Furthermore, the data used for training has been prioritized based on its experimental method of acquisition. The reasoning behind this comes from two research papers, “X-ray vs. NMR structures as templates for computational protein design” [7] and “Analysis Suggests That Packing Densities of Protein Structures Determined by NMR Are Unreliable” [8], which indicate that the experimentally predicted natural conformations, using X-Ray Crystallography, are more reliable than the ones estimated by NMR. We have ignored the other experimental methods for atomic structure determination because most of the experimentally determined native conformations on the PDB website have been estimated by the first two approaches.

Method

The method behind our predictor can be divided into two steps: data processing and predictor training, and prediction of the protein native conformation. In the next subsections we will go into more details on each one.

Data processing and predictor training

The PDB database contains more than 80,000 experimentally determined protein native conformations and our algorithm is based on information obtained from them. The database gets updated once a week with new information which gives the possibility for the continued refinement of the predictor.

We have subcategorized the entire pdb database based on several criteria. As a first step the files were split into two subdirectories, in one we place the pdb files that contain missing coordinates for atoms and in the other the ones that do not. Missing coordinates occur when the experimental technique used to determine the native conformation of a protein was not able to extrapolate the three-dimensional coordinates for all of the atoms. With this segregation we hope to limit the chance of errors, by our predictor, based on the reasoning that if there are pdb files with missing atomic coordinates then the experimental method used to obtained the native conformation did not perform well or there were complications. We reason that these files may contain more errors compared to pdb files with no atomic coordinates missing. For example, the pdb file pdb1at9.ent has its first amino acid missing as well as all amino acids from 232 to 248.

Furthermore, the two subdirectories were split according to the experimental method of obtaining the protein native conformations: X-Ray Crystallography and Protein NMR. As discussed in the beginning of this chapter we have based our predictor on research indicating that X-Ray is a more reliable technique compared to Protein NMR. For training the predictor we have used information from single-domain proteins. In the case of multi-domain proteins, there are more forces exerted onto the atoms of amino acids located at the places where the domains of the protein dock. Thus those amino acids will have their atomic coordinates displaced compared to the same amino acids in different locations. Furthermore, we are not using a program to extrapolated the location of the docking sites for multi-domain proteins, thus we cannot know onto which amino acids to apply the misplacement information. As a final sub-categorization we have used the expression organism. Given the importance of the expression system, discussed in the beginning of this chapter, we have determined that a distinct separation was needed. Figure 8 gives an idea of the final result.

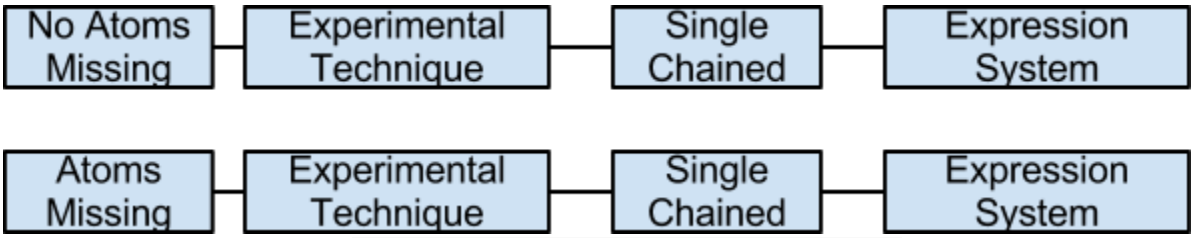


Figure 8 PDB database directory tree. (source: own work)

To obtain the training data, from the pdb files, we need to calculate the difference between the x, y, z coordinates of the N (nitrogen) atom of an amino acid and the N (nitrogen) atom of its previous neighbour (See Formula 1). We also calculate the difference between the N (nitrogen) atom and the rest of the atoms of the amino acid (See Formula 2). Table 3 shows an example of a pdb file.

$$CalculatedCoordinatesAtomN(x) = CurrentAminoAcidAtomN(x) - PreviousAminoAcidAtomN(x)$$
$$CalculatedCoordinatesAtomN(y) = CurrentAminoAcidAtomN(y) - PreviousAminoAcidAtomN(y)$$
$$CalculatedCoordinatesAtomN(z) = CurrentAminoAcidAtomN(z) - PreviousAminoAcidAtomN(z)$$

Formula 1

$$CalculatedCoordinatesAtomX(x) = CurrentAminoAcidAtomX(x) - CurrentAminoAcidAtomN(x)$$
$$CalculatedCoordinatesAtomX(y) = CurrentAminoAcidAtomX(y) - CurrentAminoAcidAtomN(y)$$
$$CalculatedCoordinatesAtomX(z) = CurrentAminoAcidAtomX(z) - CurrentAminoAcidAtomN(z)$$

Formula 2 The notaion **AtomX** denotes any atom.

/	atom_N	type	Amino_Acid	Amino_Acid_N	x	y	z
ATOM	1	N	MET	1	65.039	110.904	94.786
..	..	..	MET	1	..	..	..
ATOM	7	N	ALA	2	66.482	111.01	94.524
ATOM	8	CA	ALA	2	67.021	109.753	93.813
ATOM	9	C	ALA	2	66.728	109.508	92.635
ATOM	10	O	ALA	2	67.23	111.287	95.827
ATOM	11	CB	ALA	2	66.79	112.586	96.504
ATOM	12	CG	ALA	2	67.025	114.021	95.476
ATOM	13	SD	ALA	2	68.749	114.207	95.077
ATOM	14	CE	ALA	2	67.793	108.985	94.553
ATOM	15	N	TER	3	67.093	108.085	94.053

Table 3 An excerpt of the atom model from a pdb file. The **type** column represents a code indicating the type of the atom inside the amino acid. N means nitrogen. The **X, Y, Z** columns are the spatial coordinates of the atoms. The **Amino_Acid_N** column represents the location of the amino acid relative to the amino acid sequence of the protein.

We have carried out these calculations on amino acid fragments of sizes 3, 5, 7, and 9, where the coordinates of the atoms for the amino acids have been estimated. We have different sized fragments because depending on the type of the surrounding amino acids the spatial position of the atoms of the middle amino acid differ. For example, if we have the fragment X-SER-Y, depending on what amino acids X and Y are, the coordinates for the atoms of SER differ greatly. The possibilities narrow when we consider fragments of larger sizes as the most important factor in protein folding is the amino acid configuration. Thus by choosing larger

amino acid fragments we limit the possible coordinate space the atoms of the middle amino acid can occupy. Below is the pseudo code for the procedure.

```
void ObtainAngleStatistics( string& directory_angle_statistics, string& file_read_from,
                            int fragment_length, AminoAcidChain &amino_acid_chain )
{
    IF amino_acid_chain.Size() < fragment_length* 3 THEN return ENDIF

     IF fragment_length< 3 THEN return ENDIF

    FOR amino_acid_id =  fragment_length/2 TOamino_acid_chain.Size - fragment_length/2
      amino_fragment = amino_acid_chain[amino_acid_id] TO
                       amino_acid_chain[amino_acid_id + fragment_length]
       IF amino_fragment has missing backbone atoms
          THEN Go to the next amino_acid
       ELSE continue ENDIF

      amino_acid_file = directory_angle_statistics + "\" + amino_fragment
      OPEN_FILE amino_acid_file in append mode

       FOR fragment_acid_id = amino_acid_id TO amino_acid_id  + fragment_size
          IF fragment_acid_id == 0    // Note: The amino_acid of the amino_acid_chain not amino_fragment
             THEN OUTPUT to amino_acid_file N : (0, 0, 0)
          ELSE OUTPUT to amino_acid_file   // Note: Formula 1
             N : (amino_acid_chain[fragment_acid_id ].Atom(N) - amino_acid_chain[fragment_acid_id  -1].Atom(N))
          ENDIF

          FOR atom_id = 1 TO amino_acid_chain[fragment_acid_id ].NumberOfAtoms
            OUTPUT to amino_acid_file
            amino_acid_chain[fragment_acid_id].Atom[atom_id].Name : ( // Note: Formula 2
            amino_acid_chain[fragment_acid_id].Atom(atom_id) - amino_acid_chain[fragment_acid_id  -1].Atom(N))
          ENDFOR
     ENDFOR
   ENDFOR
}
```

Prediction of the protein native conformation

The amino acid string for the target protein, that is to have its native conformation predicted, is read from left to right. Our algorithm requires a ranking file described in the previous section. We calculate the three-dimensional position of the nitrogen atom of the other amino acids by utilizing the nitrogen coordinates we calculated through Formula 1. (See Formula 3). The rest of the atomic positions, in the amino acids, are calculated from their currently extrapolated nitrogen atom coordinates based on Formula 4. The predicted native conformation is written in the PDB file format to be used as an input to our third-party scoring application. Furthermore, below is presented the pseudo code for the algorithm.

```
// Return an amino_fragment around the from_amino_acid.
string construct_fragment( from, fragment_size, amino_acid_sequence );

// Calculate angles for a fragment of amino acids.
CalculateAngles( amino_acid_sequence, from, fragment_size, STORE angles );

void calculate_native_conformation( AminoAcidChain &amino_acid_chain )
{
   fragment_size = 9
   UNTIL fragment_file_name == 0
      fragment = construct_fragment( 0, fragment_size, amino_acid_sequence )
      fragment_file_name = find_fragment_file( fragment, coordinate_directories, fragment_size )
   ENDUNTIL

   CREATE AminoAcidChain calculated_models[number_of_fragment_samples( fragment_file_name )]

   FOR model = 0 TO  number_of_fragment_samples( fragment_file_name )
      COPY amino_acid_chain to calculated_models[model]
     // Calculate the N coordinates of the first fragment_size amino acid range by Formula 3.
     CalculateCoordinates( calculated_models[model].Fragment( 0, fragment_size ), fragment )

     current_amino_acid = fragment_size

     WHILE current_amino_acid < amino_acid_sequence.size

       fragment_size = 9
       UNTIL fragment_file_name == 0
         fragment = construct_fragment( current_amino_acid, fragment_size, amino_acid_sequence )
         fragment_file_name = find_fragment_file( fragment, coordinate_directories, fragment_size )
       ENDUNTIL

      CONTAINER model_angles
      CalculateAngles( calculated_models[model], current_amino_acid, fragment_size, model_angles )

      OPEN FILE fragment_file_name
      UNTIL END_OF_FILE
         READ one fragment at a time
         CONTAINER fragment_angles
         CalculateAngles( fragment , 0, fragment_size, fragment_angles)
         compare = CompareAngles( model_angles, fragment_angles )
         IF compare has the best score so far
            THEN saved_fragment = fragment
                 fragment_score = compare
        ENDIF
      ENDUNTIL

  CalculateCoordinates( calculated_models[model].Fragment( current_amino_acid, current_amino_acid + fragment_size/ 2), fragment.Sub(fragment_size/2 +1, fragment_size) )

      model_score[model] += fragment_score
```

```
    WHILEND

  ENDFOR

  best_model = BestScore( model_score[model] )
  SaveBestModelToPDBFile( calculated_model[ best_model ] )
}

string construct_fragment( from, fragment_size, amino_acid_sequence )
{
  IF from == 0 OR from <= fragment_size /2
    THEN return amino_acid_sequence[0] TO amino_acid_sequence[fragment_size]
  ELSEIF amino_acid_sequence.size <= from + fragment_size /2
    THEN return amino_acid_sequence[end] TO amino_acid_sequence[end - fragment_size]
  ELSE
   THEN return amino_acid_sequence[from - fragment_size/2] TO amino_acid_sequence[from + fragment_size/2]
  ENDIF
}

// Calculate angles for a fragment of amino acids.
CalculateAngles( sequence, from, fragment_size, STORE angles )
{
  // Example: sequence A-B-C-D-E-F
  // Example: from =5 (E)
  // The angles are calculated by Formula 5.
  CASE fragment_size == 3
     Angle( C, D, E )
  CASE fragment_size == 5
    RUN Previous case
    Angle( B, C, D )
    Angle( B, C, E )
  CASE fragment_size == 7
    RUN Previous case
    Angle( A, B, C )
    Angle( A, B, D )
   CASE fragment_size == 9
    Angle( Empty( 0, 0, 0), A, B )
    Angle( Empty( 0, 0, 0), A, C )

   // Extra angles, such as ABD, are calculated in order to pinpoint which of the two possible
   // angles the Angle function returns. The angle returned by Angle is unsigned, thus we do not know if
   // it is, for example +45 degrees or -45 degrees. The extra angles can help us pinpoint it.
}
```

$$CurrentAminoAcidAtomN(x) = CalculatedCoordinatesAtomN(x) + PreviousAminoAcidAtomN(x)$$
$$CurrentAminoAcidAtomN(y) = CalculatedCoordinatesAtomN(y) + PreviousAminoAcidAtomN(y)$$
$$CurrentAminoAcidAtomN(z) = CalculatedCoordinatesAtomN(z) + PreviousAminoAcidAtomN(z)$$

Formula 3

$$CurrentAminoAcidAtomX(x) = CalculatedCoordinatesAtomX(x) + CurrentAminoAcidAtomN(x)$$
$$CurrentAminoAcidAtomX(y) = CalculatedCoordinatesAtomX(y) + CurrentAminoAcidAtomN(y)$$
$$CurrentAminoAcidAtomX(z) = CalculatedCoordinatesAtomX(z) + CurrentAminoAcidAtomN(z)$$

Formula 4 The notaion **AtomX** denotes any atom.

DV1 = DirecotionVector(Point2 - Point1)
DV2 = DirectionVector(Point2 - Point3)
CS = CrossPorduct(DV1, DV2)
DP = DotProduct(DV1, DV2)
Angle = atan2(L2Norm(CS), DT)
Formula 5 Angle between 3 points in 3D space

Validation

We have validated our method through two types of experiments. The first one is by predicting the CASP8 and CASP9 targets and comparing the results against the experimentally determined coordinates for the targets. The second method of validation is by comparing the same results against ones obtained by Rosetta and HHPred. This gives us an objective way to both test the accuracy of our algorithm, and how it ranks compared to some of the top predictors out there.

Chapter 3 Protein folding framework

In addition to LiacsFold the framework includes two open-source third-party tool-chains for predicting the secondary and tertiary structure of proteins. They are placed there for general use as well as to be used in validating the results from our predictor. Moreover, there is a third party tool for identifying homologous sequences, through gene alignment, which can be used as an input by one of the protein structure predictors.

The third party protein folding predictors used are Rosetta and Modeller, whereas the homology sequence identifier is HHSuite. We needed to add HHSuite as to make The Modeller fully autonomous. Furthermore a third party scoring application is included for running validation tests. The program is called TMscore and is obtained from the same laboratory that created the I-TASSER and the Quark protein prediction servers.

The suite is fully automated and highly customizable. It provides a pipeline for the arguments passed to the predictors to be altered and if the default parameters are used only an input file with the amino acid sequence of the protein is needed for the prediction. The sequence is automatically run for all predictors, or it can be selectively run for only a subset of them, and the outputs can be found in a separate directory clearly marked to indicate which predictor made which model. Furthermore, a directory with amino acid sequence files can be specified and all the files will be run one after the other in an automated way.

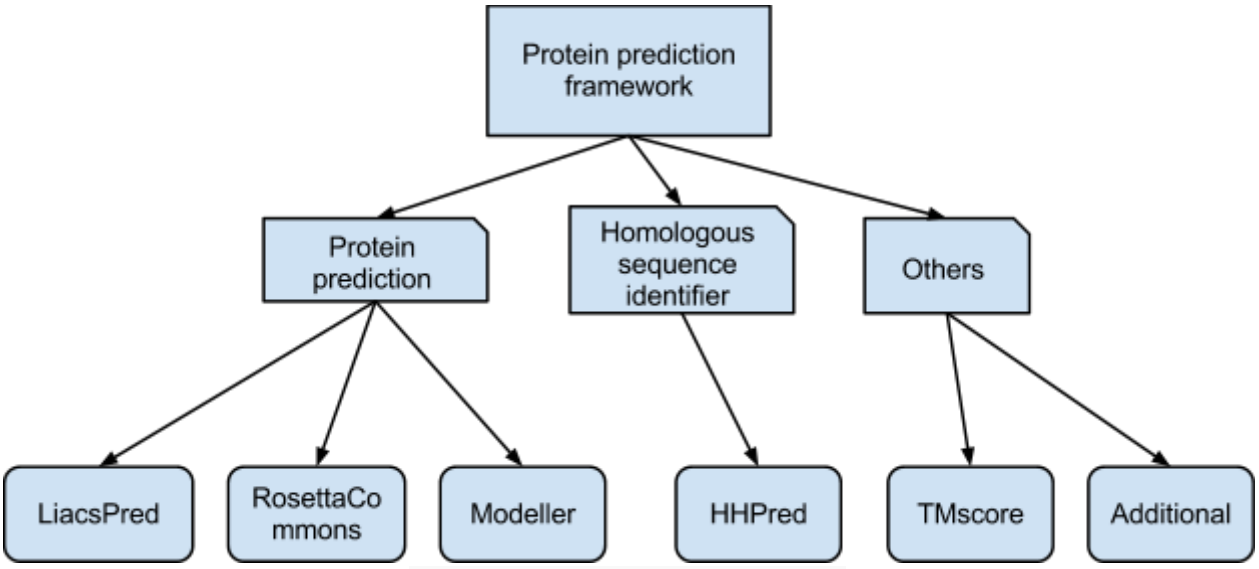
All third party tool-chains and their databases have been updated to their latest version and academic licenses obtained:

- Rosetta version **3.4**
- HHSuite version **2.0.15** (for HHPred)
- Modeller version **9.10**
- TMScore version **2012/06/05**

Framework structure

In this chapter we will explain the structure of the protein prediction framework. It is partitioned in two categories: main framework and additional applications. If predicting protein structures is the main reason for using this tool-chain then the main framework can be used and the rest ignored. The additional applications are more focused on examining secondary and tertiary structures, extracting and processing information, categorizing files, extracting amino acid spatial coordinates. Based on them one can create their own protein prediction algorithms.

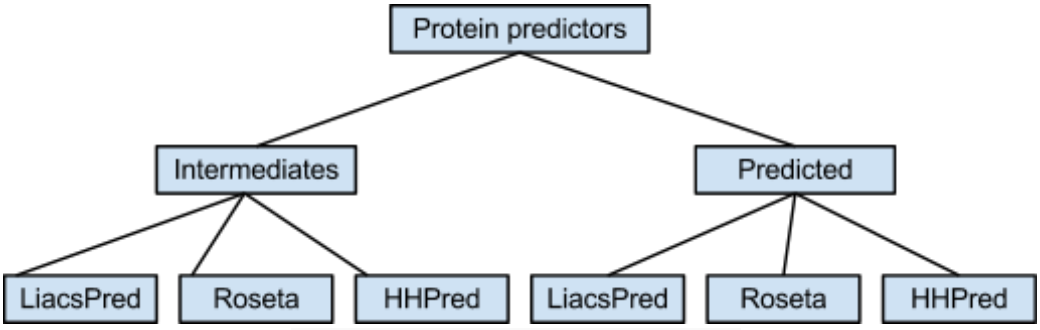
Main framework



Scheme 1 Protein folding framework structure

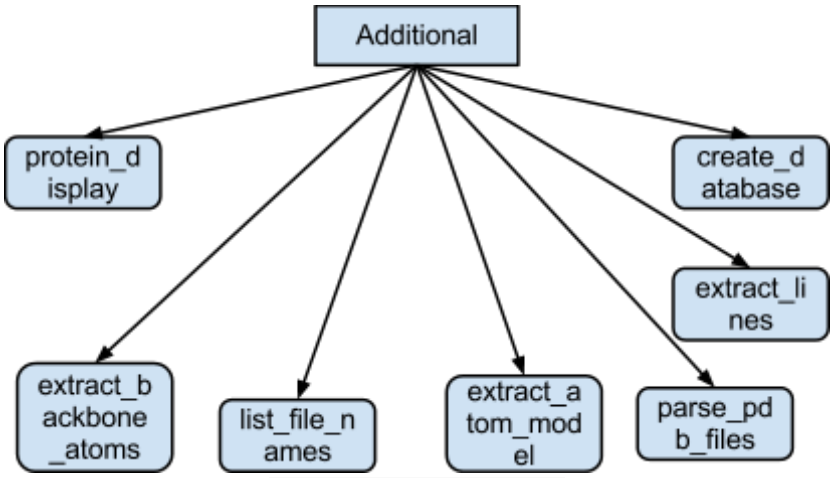
The structure of the protein folding framework (See Scheme 1) is pretty straight forward. It has three main tool-chain categories: protein prediction, homologous sequence identifier, and others. The protein predictors can be accessed through one main application or through separate ones. By default the only input needed is an amino acid sequence.

There are two directories concerning the protein prediction tools (See Scheme 2). The **"Intermediates"** directory is used to store files created throughout the execution of the various tools under their respective subdirectories. By default those files will be deleted when the predictors have finished their work. As one can expect the **"Predicted"** directory contains the calculated secondary or tertiary structures. In the case of predicting a directory of amino acids, subdirectories will be created with the name of the supplied directory within each of the tool-chains.



Scheme 2 Protein predictors directory structure

Additional tools



Scheme 3 Additional programs

Additional programs, made by us, have also been included in order to server various needs (See Scheme 3). These tools can be used not only for parsing pdb files and protein structures but also for general file processing and directory organisation.

Database_creater creates a directory like tree structures and populates it with files based on information contained inside them. As an example you can sort the pdbfiles database according to the experimental way the three-dimensional structures were obtained.

Extract_lines is an application that can extract data from files based on a template. For example the spatial coordinates for all nitrogen atoms can be extracted.

Protein_display is a program that can output the coordinate structure of proteins. It is based on OpenGL and hundreds of thousands of amino acids can be viewed simultaneously without penalty to performance, especially on a modern middle range computers.

Extract_atom_model is a program that can extract individual atom models from pdb files and save them into pdb format. The extracted atom models can later be displayed using the protein_display program or used by the TMscore scoring application.

Extract_backbone_atoms is a program that extracts only the backbone atoms from the pdb files and saves it as a atom model in the pdb format.

List_file_names is an application that lists the names of the files in a given directory.

Parse_pdb_files is a program that prepares the input file for our predictor.

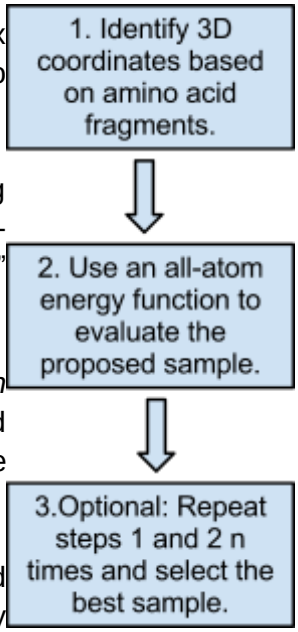
Rosetta

Rosetta is an open-source framework for predicting the tertiary structure of proteins from their amino acid sequences, predicting protein-protein interactions (docking), and provides facilities to help in protein design. It is developed by the Baker laboratory of the University of Washington's Department of Biochemistry.

The tool-chain has two different algorithms for predicting the tertiary structure of proteins. The first one is an ab-initio approach and the second one is template based. The main difference between the two approaches is that in the second one homologous sequences, to the protein, are used to guide the prediction. Of course this only makes an impact as long as such sequences do exist and their tertiary structures have been experimentally obtained. In our framework only the ab-initio approach is used and its workings are described in the following section.

Method

The ab-initio approach can be accessed by running the AbinitioRelax program inside the Rosetta tool-chain. The algorithm consists of two main steps.



During the ab-initio step the algorithm identifies fragments of varying sizes, for each amino acid of the protein sequence, and based on pre-computed (x, y, z) coordinates for the fragments, creates a "sample" tertiary structure of the protein.

Each fragment is comprised of an amino acid coupled with an *n* number of neighbours. A fragment of size three for the ALA amino acid would have the following form: X-ALA-Y, where X and Y are the surrounding amino acids.

In the Relax step an all-atom energy function is used to evaluate and adjust the sampled coordinates. Given that it applies the energy function on the full model this step can take considerably more time than the first one. Optionally the first two steps can be repeated *n* number of times and the best sample is chosen by Rosetta using a clustering approach.

Rosetta in CASP

Rosetta is arguably the most well-known protein structure prediction framework. It has a lot of functionality and flexibility and scales well on multiple servers. Despite its popularity it has mixed results in the CASP experiments as it can be seen from Chapter 1.

During CASP9 it ranked number 11 for best server predictor with a combined score of 87.240. When it comes to comparing it to human/server groups it is placed as 36 with a score of 50.221. The lower score comes from the fact that the first category considers targets which are released

only for servers. In general all groups, human and server, have to predict all targets but different rankings exit in order to give perspective on how good a group is in its own category.

Rankings for CASP8 are performed in a different way. Instead of categorizing by human/server and server targets we have a score for free-modelling and template based modelling. In both Rosetta scores in the 22 place with cumulative score of 40.786 and 48.802 respectively.

Modeller

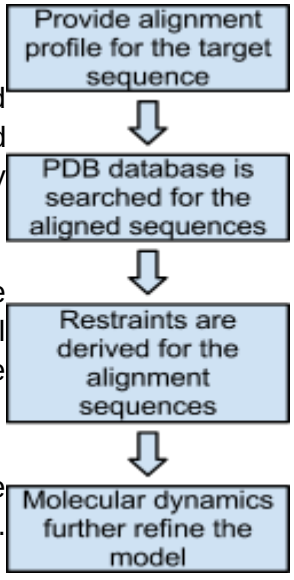
The Modeller is a prediction tool for computationally predicting protein native conformations employing a homology based approach. It does not perform the gene alignment on its own but rather such alignment must be derived by other means and the result fed to the program as an input. It is open-source and available free in the form of an academic license.

Additional restraints can be placed on the predictor. By default our framework does not ask for additional restraints beyond an alignment file, which will be generated automatically by the HHPred tool-chain, but they can be additionally supplied.

1 related protein structures (comparative modelling)
2 NMR experiments (NMR refinement)
3 rules of secondary structure packing (combinatorial modelling)
4 cross-linking experiments
5 fluorescence spectroscopy
6 image reconstruction in electron microscopy
7 site-directed mutagenesis
8 intuition
9 atom-atom potentials of mean force

Method

The algorithm behind the Modeller operates in three main steps.



A file including the amino acid sequence for the target protein is supplied along with an alignment profile against the pdb database. This file should include only the alignments that have a high degree of comparability towards the target sequence.

Second, the pdb database is being search for the files containing the experimentally obtained native conformations of the alignments which will be used to construct the tertiary structure for our target. Restraints are taken into account if such have been supplied.

The Modeller derives its restraints automatically from the native conformations in the alignment profile, for which a pdb file is available. Molecular dynamics are applied to refine the model.

Modeller in CASP

In CASP Modeller has been used in conjunction with HHPred. For further details see the next section.

HHPred

HHPred is a sequence alignment tool designed to search for remote homologs. It uses Profile Hidden Markov Models to generate an alignment. In a benchmark comparison HHPred outperformed BLAST, PSI-BLAST, HMMER, PROF_SIM, and COMPASS. Compared to them it is faster, has 50-100% more sensitivity, and generates more accurate alignments[14]. By coupling it with PSIPRED in order to capture secondary structure its sensitivity can be improved by an additional 20%[14].

Method

It is based on a modified profile-sequence comparison which is better than the typical sequence-sequence comparison. In profile-sequence comparison information about the frequency of the 20 amino acids for each column of multiple alignment is used, as is the case with PSI-BLAST. HHPred goes one step further by including information about the frequency of insertions and deletions at each column. This technique is more powerful as it uses larger set of restraints to carry out its calculations[14].

Two main algorithms are available.

1 HHBlits
2 HHMake

The difference between the two methods is that the first one is much faster and only slightly less sensitive due to its less tight restrictions. Because of the small difference in sensitivity but the gain in performance we have used it in our framework.

HHPred in CASP

HHPred has performed consistently well in the CASP experiments. During CASP8 the predictor was ranked as 9th and in CASP9 it ranks as 6th.

Scoring

For scoring TMScore is used. It is made by the same people who created the I-TASSER and Quark servers. The program is based on global alignment of two protein molecules in pdb format[15]. Scores are between 0 and 1. Anything above 0.5 indicates that the molecules have significant similarity and anything below 0.3 points to no structural similarity[16].

Another scoring tool is the ProQ2[52]. It predicts the S-score for each individual residue. This score is the transformation of the normal RMSD for each residue based on the formula: `(1/sqrt(1+RMSD_i^2/9))`, with RMSD_i representing the "*local RMSD deviation for residue ibased*

on a global superposition trying to maximize essentially the sum of S-score over the whole model". For more scoring algorithms check the CASP section.

Chapter 4 Experiments and Conclusions

For testing our protein structure prediction framework we have used the CASP experiments. Scores for the predictions are given by TMScore and each predictor has been tested on at least 100+ targets. Furthermore, the results are discussed and explanations are given as to why they are high or low.

Rosetta

The default parameters for Rosetta have been used and only 1 model per sequence has been created. For Rosetta the scores hover around 0.2 (See Table 4). It should be noted that on the website of Rosetta it is advised that as many as 20.000 to 200.000 models are to be predicted per amino acid sequence. On an average home computer the calculation of each model takes around 30 minutes. From this we could infer that the Rosetta application would definitely require a cluster of servers to operate at an optimum level and as such it is next to impossible for it to be used as a personal proteins' native conformation predictor.

CASP9

target	score	target	score	target	score	target	score
2kjx	0.1829	3p1t	0.1757	3ni8	0.1521	3nrf	0.1315
2kxy	0.136	3pnx	0.2012	3nie	0.1662	3nrh	0.2153
2ky4	0.2102	3qtd	0.1539	3njc	0.163	3nrl	0.1736
2ky9	0.1598	3mwx	0.1261	3nkd	0.2027	3nrt	0.2063
2kyt	0.1945	3mx3	0.1216	3nkg	0.2142	3nrv	0.185
2kzw	0.1196	3mx7	0.148	3nkh	0.1593	3nrw	0.2227
2l01	0.2547	3n05	0.1333	3nkl	0.1774	3nwz	0.1589
2l02	0.2531	3n0x	0.1514	3nkz	0.1881	3nxh	0.1263
2l09	0.1815	3n53	0.1695	3nlc	0.1107	3nyi	0.2098
2l0b	0.1795	3n6y	0.1219	3nmb	0.1659	3nym	0.2088
2l0c	0.138	3n6z	0.1259	3nmd	0.2557	3nyw	0.1703
2l0d	0.1419	3n72	0.1606	3nnq	0.1885	3nyy	0.1679
2l3b	0.1413	3n8u	0.1354	3no2	0.1657	3nzl	0.1979
2l3f	0.2158	3n91	0.1333	3no6	0.186	3nzp	0.1237
2l3w	0.2011	3na2	0.1426	3noh	0.1963	3o14	0.1606
2xrg	0.1134	3nat	0.1989	3npf	0.1805	3obh	0.1407
3mqo	0.215	3net	0.166	3npp	0.1398	3on7	0.1618
3mqz	0.1166	3neu	0.14	3nqk	0.1463	3oox	0.1485
3mr7	0.1794	3nf2	0.1759	3nqw	0.2014	3oql	0.1616
3mt1	0.1693	3nfv	0.1783	3nra	0.1796	3oru	0.1638
3mwt	0.1235	3nhv	0.1996	3nrd	0.1858	3os6	0.1811
3ot2	0.17	3ni7	0.1559	3nre	0.1181	3os7	0.1617

Table 4 CASP9 results for Rosetta

CASP8

Due to the poor performance and high computational demand of the Rosetta predictor in our CASP9 experiments, it was decided that the CASP8 targets will not be tested by Rosetta. The need for 20000+ predictions on the same protein is clear and only the computational power of dedicated servers can achieve that within a reasonable time frame.

Modeller

The scores for the CASP9 experiment are given below (See Table 5) with higher meaning better. They range from 0.16 to 0.92. It can be clearly seen that for a substantial percentage of the predictions Modeller got an accuracy above 0.7. As the algorithm is entirely dependent on the correct determination of homologous sequences it stands to reason that its lower scores are likely due to no homologous sequences found by HHPred or wrongly determined ones.

CASP9

target	score	target	score	target	score	target	score
2kjx	0.165	3mwx	0.8715	3nlc	0.1902	3nym	0.1033
2ky4	0.808	3mx3	0.1867	3nmb	0.7013	3nyy	0.1105
2ky9	0.2409	3mx7	0.1395	3nmd	0.6557	3nzl	0.1627
2kyt	0.2417	3n05	0.4623	3nnq	0.5135	3nzp	0.8445
2kyw	0.4631	3n0x	0.4704	3nnr	0.8673	3o14	0.5976
2kyy	0.5373	3n1u	0.9065	3no2	0.1556	3o1l	0.8939
2kzw	0.454	3n53	0.3417	3no3	0.3952	3obh	0.2213
2l01	0.7317	3n6y	0.0979	3no6	0.808	3obi	0.8877
2l02	0.723	3n72	0.1968	3noh	0.157	3on7	0.8481
2l06	0.8078	3n8u	0.8894	3npf	0.1781	3oox	0.8495
2l09	0.6582	3n91	0.148	3npp	0.1574	3oql	0.7888
2l0b	0.1649	3neu	0.3063	3nqk	0.1259	3oru	0.7142
2l0c	0.3906	3nf2	0.1847	3nqw	0.9544	3os6	0.9162
2l0d	0.6825	3nfv	0.2236	3nra	0.8649	3os7	0.8753
2l3b	0.7475	3ngw	0.7915	3nrd	0.9191	3ot2	0.7866
2l3f	0.7271	3nhv	0.1737	3nre	0.8516	3pfe	0.6874
2l3w	0.5393	3ni7	0.4471	3nrf	0.1429	3pnx	0.6229
2xrg	0.2106	3ni8	0.1767	3nrg	0.6288	3qtd	0.9647
3mqo	0.5139	3nie	0.2145	3nrl	0.1506		
3mqz	0.3569	3njc	0.2884	3nrt	0.3463		
3mr7	0.3358	3nkd	0.6433	3nrv	0.2882		
3mse	0.4715	3nkh	0.5356	3nwz	0.738		
3mt1	0.9509	3nkl	0.6532	3nxh	0.1712		
3mwt	0.9943	3nkz	0.3719	3nyi	0.4799		

Table 5 CASP9 results for Modeller

CASP8

In CASP8 (See Table 6) Modeller was able to produce predictions which can be considered from pretty good to almost excellent, and some that are abysmal. The strength of this predictor is in the number of experimentally obtained native conformation for highly identical homologs to the target protein.

target	score	target	score	target	score	target	score
2k3i	0.57	3d0f	0.5334	3dal	0.1521	3dlb	0.8242
2k4m	0.2966	3d0k	0.5955	3dao	0.8279	3dlc	0.7629
2k4v	0.1556	3d19	0.8859	3dax	0.8796	3dlm	0.2631
2k4x	0.5125	3d1l	0.7089	3db0	0.8573	3dls	0.8222
2k53	0.7805	3d1p	0.8116	3db3	0.1534	3dm3	0.8363
2k54	0.5201	3d37	0.8684	3db5	0.8534	3dm4	0.94
2k5c	0.2947	3d3o	0.7832	3db9	0.1357	3dma	0.6151
2k5d	0.1509	3d3q	0.7464	3dc7	0.7679	3dmb	0.7754
2k5e	0.5862	3d3s	0.6625	3dcd	0.88	3dmc	0.7647
2k5i	0.5042	3d3u	0.9649	3dcp	0.8113	3dme	0.9097
2k5j	0.4291	3d3y	0.8513	3dcx	0.7809	3dmn	0.5354
2k5l	0.9175	3d4e	0.4676	3dcy	0.8158	3dn7	0.7166
2k5r	0.4355	3d4o	0.8478	3ddv	0.9198	3dnh	0.7938
2k5w	0.565	3d4r	0.2231	3ded	0.4559	3dnp	0.7769
2kdl	0.2386	3d5n	0.8307	3dev	0.6528	3dnx	0.3287
2kdm	0.7853	3d5p	0.7108	3dew	0.7511	3do5	0.9582
2vsv	0.1664	3d6j	0.9543	3dex	0.9555	3do6	0.9715
2vsw	0.8427	3d6k	0.9605	3df8	0.75	3do8	0.6555
2vuw	0.1654	3d6w	0.7667	3dfa	0.1997	3dou	0.9239
2vux	0.25	3d7i	0.6847	3dfe	0.2742	3dr5	0.9133
2vwr	0.8255	3d7l	0.901	3dh1	0.207	3dsm	0.7165
2vx2	0.2148	3d89	0.7294	3dhn	0.823	3dup	0.3688
2vx3	0.2349	3d8h	0.1478	3di5	0.7761	3,00E+03	0.9633
3cyn	0.9488	3d8p	0.9357	3djb	0.8088	3,00E+38	0.1531
3czp	0.468	3d8u	0.8948	3dka	0.7575	3g5a	0.9946
3czq	0.7304	3da1	0.9443	3dkp	0.8602	3gwl	0.3602
3czu	0.186	3da2	0.1663	3dkz	0.9109		
3d01	0.8797	3dai	0.9053	3dl1	0.9836		

Table 6 CASP8 results for Modeller

LiacsPred

Table 3 provides the results of our predictor on the CASP9 targets. The columns with red text indicate the target protein and the columns with black text denote the score. It can clearly be seen that the results are very poor. This could be due to the limited number of training data for fragments of sizes 5, 7, and 9 (naturally, the larger the amino acid fragment, the smaller the training set for it). Five tests have been run, each using different training data to calculate each target's native conformation. The training data used comes from the pdb files of Escherichia Coli experimentally determined by X-Ray Diffraction and NMR.

1. Prioritizing based on the largest fragment size from the above mentioned pdb files.
2. Using training data from NMR pdb files with missing atomic information.
3. Using training data from X-Ray pdb files with missing atomic information.
4. Using training data from NMR pdb files with no missing atomic information.
5. Using training data from X-Ray pdb files with no missing atomic information.

CASP9

target	frag_size	m_nmr	m_x-ray	nm_nmr	nm_x-ray
2kjx	0.1443	0.1865	0.1233	0.1777	0.1432
2kxy	0.1271	0.1562	0.1050	0.1127	0.1385
2ky4	0.1708	0.1515	0.1245	0.2040	0.1560
2ky9	0.1038	0.1263	0.1389	0.1354	0.1192
2kyt	0.2097	0.1457	0.2105	0.1539	0.1353
2kyw	0.1329	0.1265	0.1272	0.1818	0.1214
2kyy	0.1304	0.1458	0.1365	0.1167	0.1256
2kzw	0.1048	0.1028	0.1230	0.1224	0.0999
2l01	0.1178	0.1743	0.1723	0.1365	0.1367
2l02	0.1471	0.1416	0.1529	0.1399	0.1689
2l06	0.1412	0.1508	0.1486	0.1043	0.1323
2l09	0.1535	0.1877	0.2391	0.1486	0.1589
2l0b	0.1183	0.1298	0.1128	0.1140	0.1567
2l0c	0.1513	0.1428	0.1313	0.1490	0.1335
2l0d	0.1213	0.1266	0.1134	0.0970	0.1410
2l3b	0.1024	0.1387	0.1015	0.1082	0.1206
2l3f	0.1233	0.1604	0.1170	0.1157	0.1153
2l3w	0.1127	0.1269	0.1560	0.1628	0.1610
2xgf	-	-	-	-	-
2xrg	0.1042	0.0775	0.0811	0.1217	0.0966
2xse	-	-	-	-	-
3mqo	0.1609	0.1291	0.1691	0.1440	0.1083
3mqz	0.1161	0.1028	0.1111	0.1170	0.1153
3mr0	0.1460	0.1967	0.1559	0.0981	0.1204
3mr7	0.1234	0.1484	0.1537	0.1613	0.1401
3mse	0.1093	0.1545	0.1050	0.1177	0.1325
3mt1	0.0894	0.0917	0.1270	0.1100	0.0933
3mwt	0.1906	0.0893	0.1659	0.1192	0.1101

3mwx	0.0709	-	0.1352	0.1004	0.0924
3mx3	0.1443	0.0937	0.1188	0.1072	0.1127
3mx7	0.1370	0.1366	0.1258	0.1165	0.1440
3n05	0.1003	0.0971	0.1350	0.1158	0.1234
3n0x	0.1528	0.1159	0.1268	0.1222	0.1238
3n1u	0.1660	0.1263	0.1453	0.1282	0.1455
3n53	0.1227	0.1234	0.1468	0.1455	0.1359
3n6y	0.0892	0.1176	0.1101	0.1158	0.0889
3n6z	0.1112	0.0911	0.0908	0.1118	0.0839
3n70	-	-	-	-	-
3n72	0.1416	0.0785	0.1056	0.1167	0.1544
3n8u	0.1185	0.1230	0.1350	0.1369	0.1573
3n91	0.1129	0.1159	0.1522	0.1466	0.1044
3na2	0.1392	0.1451	0.1326	0.1464	0.1439
3nat	0.1515	0.1169	0.1469	0.1358	0.1281
3nbm	-	-	-	-	-
3ne8	0.0859	0.0786	0.0213	0.0590	0.0444
3net	0.1394	0.1231	0.1167	0.0918	0.1242
3neu	0.1596	0.1724	0.1238	0.1052	0.1603
3nf2	0.1166	0.1464	0.1879	-	0.0992
3nfv	0.0954	0.0874	0.1181	0.0976	0.0974
3ngw	0.1492	0.1126	0.1716	0.1739	0.1220
3nhv	0.1234	0.1367	0.1682	0.1217	0.1121
3ni7	0.1232	0.1654	0.1354	0.1498	0.1481
3ni8	0.0875	0.1194	0.1093	0.1389	0.1078
3nie	0.1073	0.1077	0.1209	0.1242	0.0831
3njc	0.1167	0.1399	0.1578	0.1267	0.1354
3nkd	0.1303	0.1355	0.1261	0.1316	0.1669
3nkg	0.1303	0.1127	0.1280	0.1256	0.1476
3nkh	0.1198	0.1314	0.1361	0.1345	0.1201
3nkl	0.1378	0.1350	0.1667	0.1398	0.1196
3nkz	0.1399	0.1492	0.1585	0.1816	0.1989
3nlc	0.1224	0.1186	0.1368	0.1082	0.0997
3nmd	0.1621	0.1173	0.1604	0.1637	0.1585
3nnq	0.1734	0.1403	0.1330	0.1833	0.1506
3nnr	0.1567	0.1263	0.1281	0.1536	0.1140
3no2	0.0874	0.1619	0.1559	0.1133	0.0864
3no3	0.1727	0.1003	0.1198	0.1009	0.1341
3no6	0.1394	0.1365	0.1245	0.1275	0.1310
3noh	0.0913	0.1512	0.1160	0.1380	0.1025
3npf	0.1622	0.1290	0.1713	0.1007	0.1622
3npp	0.1146	0.1340	0.1657	0.1226	0.1056
3nqk	0.1229	0.1042	0.1135	0.1061	0.1161
3nqw	0.1690	0.1871	0.1707	0.1768	0.1155
3nr8	-	-	-	-	-
3nra	0.1255	0.1182	0.1227	0.1207	0.0856

3nrd	0.1680	0.1168	0.1439	0.1301	0.1385
3nre	0.1092	0.0967	0.1064	0.0907	0.1012
3nrf	0.1242	0.1509	0.1580	0.1253	0.1229
3nrg	0.1554	0.1130	0.1393	0.1308	0.1406
3nrh	0.1351	0.1514	0.1280	0.0945	0.1228
3nrl	0.1338	0.1269	0.1179	0.1358	0.1205
3nrt	0.1238	0.1831	0.1594	0.1526	0.1115
3nrv	0.1487	0.1252	0.1463	0.1622	0.1333
3nrw	0.1937	0.1514	0.1588	0.1509	0.1910
3nwz	0.1139	0.1458	0.1274	0.1200	0.1548
3nxh	0.1059	0.0832	0.1076	0.0980	0.1013
3nyi	0.1631	0.1300	0.1355	0.1338	0.1150
3nym	0.1400	0.1485	0.1519	0.1445	0.1442
3nyw	0.1339	0.1283	0.1247	0.1184	0.0931
3nyy	0.1170	0.0827	0.1346	0.1186	0.1072
3nzl	0.7021	0.1451	0.1184	0.7021	0.1376
3nzp	0.1141	0.1100	0.0768	0.1125	0.1019
3o14	0.1079	0.0818	0.1057	0.1247	0.1039
3o1l	0.0995	0.1363	0.1337	0.1000	0.1448
3obh	0.1456	0.1188	0.1338	0.1271	0.1393
3obi	0.1297	0.1011	-	0.1051	0.1401
3on7	0.1238	0.1082	0.1272	0.1224	0.1343
3oox	0.1067	0.1020	0.0853	0.1225	0.1450
3oql	0.1097	0.1195	0.1483	0.1351	0.1220
3oru	0.1434	0.1489	0.1102	0.1000	0.1175
3os6	0.1305	0.1269	0.1135	0.1094	0.0812
3os7	0.1140	0.1353	0.1170	0.0842	0.0942
3ot2	0.1443	0.1164	0.1646	0.1682	0.1191
3p1t	0.1243	0.1207	0.0987	0.1098	0.1123
3pfe	0.1060	0.1096	0.1139	0.1184	0.0943
3pnx	0.1380	0.1227	0.1088	0.1671	0.1367
3qtd	0.1029	0.1219	0.1072	0.1228	0.0896
3voq	-	-	-	-	-

Table 7 CASP9 results for LiacsPred.
Each of the five columns represents a different way of calculating the native conformation.
frag_size: based on the largest fragment size
m_nmr: based only on information from nmr e-coli pdb files with missing information
m_x-ray: based only on information from x-ray e-coli pdb files with missing information
nm_nmr and **nm_x-ray** follow the same logic only pdb files with no missing information has been used.

CASP8

Table 8 provides an overview of the results obtained by the LiacsPred tool. The same overall performance can be seen, as shown by its results for the CASP9 experiment. The reasoning behind these results is follows the same path (See Table 7).

target	frag_size	m_nmr	m_x-ray	nm_nmr	nm_x-ray
2k3i	0.1545	0.1229	0.1358	0.1930	0.1378
2k4m	0.1315	0.1122	0.1474	0.1164	0.1351
2k4n	0.1198	0.1562	0.1494	0.1301	0.1755
2k4v	0.1349	0.1321	0.1469	0.1159	0.1294
2k4x	0.1641	0.1308	0.1446	0.1455	0.1407
2k53	0.1206	0.1438	0.1371	0.1596	0.1544
2k54	0.1360	0.1534	0.1370	0.1246	0.1536
2k5c	0.1536	0.1427	0.1545	0.1436	0.1459
2k5d	0.1157	0.1433	0.1485	0.1245	0.1310
2k5e	0.1616	0.1545	0.1620	0.1669	0.1878
2k5i	0.1442	0.1227	0.1581	0.1185	0.1579
2k5j	0.1353	0.1560	0.1441	0.1561	0.1403
2k5l	-	0.1378	0.1405	0.1533	-
2k5r	0.1556	0.1365	0.1258	0.1611	0.1573
2k5w	0.1180	0.1270	0.1261	0.1280	0.1085
2kdl	0.1882	0.1332	0.1615	0.1882	0.1648
2kdm	0.3995	0.1478	0.1638	0.3995	0.1850
2vsv	0.1278	0.1397	-	0.1271	0.1483
2vsw	-	0.1288	0.1443	0.1333	0.1280
2vuw	0.4211	0.0940	0.4042	0.1444	0.1126
2vux	0.1413	0.1567	0.1272	0.1467	0.1478
2vwr	0.1503	0.1343	0.1187	0.1298	0.1936
2vx2	0.1139	0.1047	0.1785	0.1059	0.1286
2vx3	0.1249	0.1094	0.1155	0.1123	0.1562
3cyn	0.1623	0.1027	0.1250	0.1411	0.1079
3czp	0.1230	0.1143	0.1204	0.1169	0.1078
3czq	0.1371	0.1494	0.1408	0.1272	0.1514
3czu	0.1204	0.1356	0.1152	0.0969	0.1190
3czx	0.1307	0.1494	0.1451	0.1058	0.1659
3d01	0.1128	0.1423	0.1214	0.1323	0.1397
3d0f	0.1473	0.1416	0.1847	0.1411	0.1600
3d0j	0.1336	0.1641	-	0.1330	0.1156
3d0k	0.0972	0.1228	0.1160	0.1247	0.1188
3d19	0.1515	0.1701	0.1319	0.1415	0.1072
3d1l	0.1206	0.1184	0.1471	0.1335	0.1231
3d1p	0.1486	0.1098	0.1185	0.1423	0.1411
3d37	0.1243	0.0985	0.1038	0.1026	0.1018
3d3o	0.1052	0.1262	0.1078	0.1150	0.1243
3d3q	0.1865	0.1200	0.1128	0.1394	0.1563
3d3s	0.1541	0.1392	0.1495	0.1326	0.1293

3d3u	0.0979	0.1158	0.0983	0.1172	0.1010
3d3y	0.1332	0.1504	0.1095	0.1270	0.1158
3d4e	0.1305	0.1233	-	0.0962	0.1387
3d4o	0.1211	0.1208	0.1304	0.1519	0.0955
3d4r	0.1490	0.1289	0.1804	0.1544	0.1220
3d5n	0.1363	0.1227	0.1045	0.1075	0.1345
3d5p	0.1714	0.1327	0.1778	0.1300	0.1276
3d6j	0.1276	0.1392	0.1596	0.1265	0.0935
3d6k	0.1674	-	0.0979	0.1102	0.1372
3d6w	0.1269	0.1404	0.1414	0.1017	0.1316
3d7i	0.1505	0.1692	0.1625	0.1597	0.1619
3d7l	0.1511	0.1248	0.1160	0.1246	0.1168
3d89	0.1391	0.1176	0.1373	0.1355	0.1321
3d8b	0.1384	0.1062	0.1391	0.1373	0.1320
3d8h	0.1210	0.1300	0.1647	0.1052	0.1289
3d8p	0.1251	0.1306	0.1295	0.1709	0.1251
3d8u	0.1157	0.1457	0.1223	0.1261	0.1119
3da1	0.0904	0.1455	0.1118	0.0976	0.0843
3da2	0.1032	0.0941	0.1077	0.1146	0.1128
3dai	0.1296	0.1582	-	0.1527	0.1274
3dal	0.1477	0.1127	0.1165	0.1603	0.1211
3dao	0.1331	0.1683	0.1454	0.1093	0.1454
3dax	-	0.1036	-	0.1161	0.1638
3db0	0.1748	0.1370	0.1420	0.1209	0.1261
3db3	0.2309	0.8469	0.1573	0.0931	0.1272
3db5	0.1338	0.1261	0.2023	0.1226	0.1364
3db9	0.1375	0.1144	0.0767	0.0998	0.0958
3dc7	0.1165	0.1636	0.1295	0.1229	0.1106
3dcd	0.1173	-	0.1398	0.1103	0.1485
3dcp	0.1617	0.1117	0.1575	0.1275	0.1293
3dcx	0.1871	0.1346	0.1429	0.1467	0.1423
3dcy	0.1543	0.1460	0.1220	0.0995	0.1352
3ddv	0.1518	0.1496	0.1403	0.1197	0.1394
3ded	0.1322	0.1254	0.1480	0.1117	0.1468
3dee	0.1467	0.1484	0.1412	0.1265	0.1438
3dev	0.1563	0.1493	0.1566	0.1331	0.1409
3dew	0.1603	0.1425	0.1465	0.1063	0.1370
3dex	0.1692	0.1379	0.1505	0.1218	0.1499
3df8	0.1290	0.1489	0.1623	0.1721	0.1622
3dfa	0.1305	0.1263	0.1271	0.1265	0.1118
3dfd	0.1475	0.1398	0.1252	0.1510	0.1123
3dfe	0.1668	0.1205	0.1289	0.1967	0.1322
3dh1	0.1172	0.1319	0.1504	0.1678	0.1309
3dhn	0.1240	0.1417	0.1372	0.1108	0.1389
3di5	0.1617	0.1687	0.1600	0.1283	0.1406
3djb	0.1726	0.1445	0.1526	0.1194	0.1461

3dka	0.1540	0.1507	0.1370	0.1337	0.1417
3dkp	0.1130	0.0990	0.1363	0.1383	0.1319
3dkz	0.1144	0.1066	0.1308	0.1289	0.1182
3dl1	0.1166	0.1131	0.1205	0.1529	0.1141
3dlb	0.1277	0.1064	0.1279	0.1244	0.0838
3dlc	0.1161	0.1208	0.1637	0.1223	0.1484
3dlm	0.1417	0.1172	0.0871	0.1206	0.1628
3dls	0.1178	0.0967	0.1297	0.1472	0.1418
3dm3	0.1023	0.1104	0.1133	0.1502	0.1208
3dm4	0.1230	0.1486	0.1382	0.1425	0.1257
3dma	0.1159	0.1572	0.1048	0.1706	0.1085
3dmb	0.1204	0.1256	0.1462	0.1181	0.1420
3dmc	0.1572	0.1177	0.1590	0.1639	0.1122
3dme	0.1271	0.0888	0.0865	0.1434	0.1223
3dmn	0.1284	0.1426	-	0.1359	0.1567
3dn7	0.1234	0.1743	0.1626	0.1209	0.1163
3dnh	0.1635	0.1148	0.1074	0.1392	0.1304
3dnp	0.1467	0.1366	0.1420	0.1567	0.1406
3dnx	0.1496	0.1243	0.1554	0.1739	0.1179
3do5	0.1724	0.1483	0.1466	0.1470	0.1116
3do6	0.1171	0.1053	0.0967	0.1084	0.0996
3do8	0.1272	0.1288	0.1390	0.1762	0.1476
3do9	0.1227	0.1137	0.1651	0.1469	0.1254
3doa	0.1337	0.1329	0.1166	0.1462	0.1326
3dou	0.1163	0.1352	0.1401	0.1464	0.1206
3dr5	0.1453	0.1077	0.1488	0.1412	0.1504
3dsm	0.1338	0.1279	0.0932	0.1317	0.1175
3dtd	0.0907	0.1225	0.0873	0.1101	0.1506
3dup	0.1211	0.1492	0.1676	0.1381	0.1490
3e03	0.1117	0.1546	0.1358	0.1029	0.0914
3e38	0.0846	0.1089	0.0824	0.0937	0.1733
3g5a	0.1193	0.1651	0.1357	0.1529	0.1360
3gwl	0.1544	0.1639	0.1616	0.1426	0.1410

Table 8 CASP8 results for LiacsPred.
Each of the five columns represents a different way of calculating the native conformation.
frag_size: based on the largest fragment size
m_nmr: based only on information from nmr e-coli pdb files with missing information
m_x-ray: based only on information from x-ray e-coli pdb files with missing information
nm_nmr and **nm_x-ray** follow the same logic only pdb files with no missing information has been used.

Comparison

Here we will present a comparison of the results from the different predictors based on the CASP experiments. For each CASP experiment we will present the results in one table.

CASP9

target	frag_size	m_nmr	m_x-ray	nm_nmr	nm_x-ray	Modeller	Rosetta
2kjx	0.1443	0.1865	0.1233	0.1777	0.1432	0.165	0.1829
2kxy	0.1271	0.1562	0.1050	0.1127	0.1385	-	0.136
2ky4	0.1708	0.1515	0.1245	0.2040	0.1560	0.808	0.2102
2ky9	0.1038	0.1263	0.1389	0.1354	0.1192	0.2409	0.1598
2kyt	0.2097	0.1457	0.2105	0.1539	0.1353	0.2417	0.1945
2kyw	0.1329	0.1265	0.1272	0.1818	0.1214	0.4631	-
2kyy	0.1304	0.1458	0.1365	0.1167	0.1256	0.5373	-
2kzw	0.1048	0.1028	0.1230	0.1224	0.0999	0.454	0.1196
2l01	0.1178	0.1743	0.1723	0.1365	0.1367	0.7317	0.2547
2l02	0.1471	0.1416	0.1529	0.1399	0.1689	0.723	0.2531
2l06	0.1412	0.1508	0.1486	0.1043	0.1323	0.8078	-
2l09	0.1535	0.1877	0.2391	0.1486	0.1589	0.6582	0.1815
2l0b	0.1183	0.1298	0.1128	0.1140	0.1567	0.1649	0.1795
2l0c	0.1513	0.1428	0.1313	0.1490	0.1335	0.3906	0.138
2l0d	0.1213	0.1266	0.1134	0.0970	0.1410	0.6825	0.1419
2l3b	0.1024	0.1387	0.1015	0.1082	0.1206	0.7475	0.1413
2l3f	0.1233	0.1604	0.1170	0.1157	0.1153	0.7271	0.2158
2l3w	0.1127	0.1269	0.1560	0.1628	0.1610	0.5393	0.2011
2xgf	-	-	-	-	-	-	-
2xrg	0.1042	0.0775	0.0811	0.1217	0.0966	0.2106	0.1134
2xse	-	-	-	-	-	-	-
3mqo	0.1609	0.1291	0.1691	0.1440	0.1083	0.5139	0.215
3mqz	0.1161	0.1028	0.1111	0.1170	0.1153	0.3569	0.1166
3mr0	0.1460	0.1967	0.1559	0.0981	0.1204	-	-
3mr7	0.1234	0.1484	0.1537	0.1613	0.1401	0.3358	0.1794
3mse	0.1093	0.1545	0.1050	0.1177	0.1325	0.4715	-
3mt1	0.0894	0.0917	0.1270	0.1100	0.0933	0.9509	0.1693
3mwt	0.1906	0.0893	0.1659	0.1192	0.1101	0.9943	0.1235
3mwx	0.0709	-	0.1352	0.1004	0.0924	0.8715	0.1261
3mx3	0.1443	0.0937	0.1188	0.1072	0.1127	0.1867	0.1216
3mx7	0.1370	0.1366	0.1258	0.1165	0.1440	0.1395	0.148
3n05	0.1003	0.0971	0.1350	0.1158	0.1234	0.4623	0.1333
3n0x	0.1528	0.1159	0.1268	0.1222	0.1238	0.4704	0.1514
3n1u	0.1660	0.1263	0.1453	0.1282	0.1455	0.9065	-
3n53	0.1227	0.1234	0.1468	0.1455	0.1359	0.3417	0.1695

3n6y	0.0892	0.1176	0.1101	0.1158	0.0889	0.0979	0.1219
3n6z	0.1112	0.0911	0.0908	0.1118	0.0839	-	0.1259
3n70	-	-	-	-	-	-	-
3n72	0.1416	0.0785	0.1056	0.1167	0.1544	0.1968	0.1606
3n8u	0.1185	0.1230	0.1350	0.1369	0.1573	0.8894	0.1354
3n91	0.1129	0.1159	0.1522	0.1466	0.1044	0.148	0.1333
3na2	0.1392	0.1451	0.1326	0.1464	0.1439	-	0.1426
3nat	0.1515	0.1169	0.1469	0.1358	0.1281	-	0.1989
3nbm	-	-	-	-	-	-	-
3ne8	0.0859	0.0786	0.0213	0.0590	0.0444	-	-
3net	0.1394	0.1231	0.1167	0.0918	0.1242	-	0.166
3neu	0.1596	0.1724	0.1238	0.1052	0.1603	0.3063	0.14
3nf2	0.1166	0.1464	0.1879	-	0.0992	0.1847	0.1759
3nfv	0.0954	0.0874	0.1181	0.0976	0.0974	0.2236	0.1783
3ngw	0.1492	0.1126	0.1716	0.1739	0.1220	0.7915	-
3nhv	0.1234	0.1367	0.1682	0.1217	0.1121	0.1737	0.1996
3ni7	0.1232	0.1654	0.1354	0.1498	0.1481	0.4471	0.1559
3ni8	0.0875	0.1194	0.1093	0.1389	0.1078	0.1767	0.1521
3nie	0.1073	0.1077	0.1209	0.1242	0.0831	0.2145	0.1662
3njc	0.1167	0.1399	0.1578	0.1267	0.1354	0.2884	0.163
3nkd	0.1303	0.1355	0.1261	0.1316	0.1669	0.6433	0.2027
3nkg	0.1303	0.1127	0.1280	0.1256	0.1476	-	0.2142
3nkh	0.1198	0.1314	0.1361	0.1345	0.1201	0.5356	0.1593
3nkl	0.1378	0.1350	0.1667	0.1398	0.1196	0.6532	0.1774
3nkz	0.1399	0.1492	0.1585	0.1816	0.1989	0.3719	0.1881
3nlc	0.1224	0.1186	0.1368	0.1082	0.0997	0.1902	0.1107
3nmd	0.1621	0.1173	0.1604	0.1637	0.1585	0.6557	0.2557
3nnq	0.1734	0.1403	0.1330	0.1833	0.1506	0.5135	0.1885
3nnr	0.1567	0.1263	0.1281	0.1536	0.1140	0.8673	
3no2	0.0874	0.1619	0.1559	0.1133	0.0864	0.1556	0.1657
3no3	0.1727	0.1003	0.1198	0.1009	0.1341	0.3952	-
3no6	0.1394	0.1365	0.1245	0.1275	0.1310	0.808	0.186
3noh	0.0913	0.1512	0.1160	0.1380	0.1025	0.157	0.1963
3npf	0.1622	0.1290	0.1713	0.1007	0.1622	0.1781	0.1805
3npp	0.1146	0.1340	0.1657	0.1226	0.1056	0.1574	0.1398
3nqk	0.1229	0.1042	0.1135	0.1061	0.1161	0.1259	0.1463
3nqw	0.1690	0.1871	0.1707	0.1768	0.1155	0.9544	0.2014
3nr8	-	-	-	-	-	-	-
3nra	0.1255	0.1182	0.1227	0.1207	0.0856	0.8649	0.1796
3nrd	0.1680	0.1168	0.1439	0.1301	0.1385	0.9191	0.1858
3nre	0.1092	0.0967	0.1064	0.0907	0.1012	0.8516	0.1181

3nrf	0.1242	0.1509	0.1580	0.1253	0.1229	0.1429	0.1315
3nrg	0.1554	0.1130	0.1393	0.1308	0.1406	0.6288	-
3nrh	0.1351	0.1514	0.1280	0.0945	0.1228	-	0.2153
3nrl	0.1338	0.1269	0.1179	0.1358	0.1205	0.1506	0.1736
3nrt	0.1238	0.1831	0.1594	0.1526	0.1115	0.3463	0.2063
3nrv	0.1487	0.1252	0.1463	0.1622	0.1333	0.2882	0.185
3nrw	0.1937	0.1514	0.1588	0.1509	0.1910	-	0.2227
3nwz	0.1139	0.1458	0.1274	0.1200	0.1548	0.738	0.1589
3nxh	0.1059	0.0832	0.1076	0.0980	0.1013	0.1712	0.1263
3nyi	0.1631	0.1300	0.1355	0.1338	0.1150	0.4799	0.2098
3nym	0.1400	0.1485	0.1519	0.1445	0.1442	0.1033	0.2088
3nyw	0.1339	0.1283	0.1247	0.1184	0.0931	-	0.1703
3nyy	0.1170	0.0827	0.1346	0.1186	0.1072	0.1105	0.1679
3nzl	0.7021	0.1451	0.1184	0.7021	0.1376	0.1627	0.1979
3nzp	0.1141	0.1100	0.0768	0.1125	0.1019	0.8445	0.1237
3o14	0.1079	0.0818	0.1057	0.1247	0.1039	0.5976	0.1606
3o1l	0.0995	0.1363	0.1337	0.1000	0.1448	0.8939	-
3obh	0.1456	0.1188	0.1338	0.1271	0.1393	0.2213	0.1407
3obi	0.1297	0.1011	-	0.1051	0.1401	0.8877	-
3on7	0.1238	0.1082	0.1272	0.1224	0.1343	0.8481	0.1618
3oox	0.1067	0.1020	0.0853	0.1225	0.1450	0.8495	0.1485
3oql	0.1097	0.1195	0.1483	0.1351	0.1220	0.7888	0.1616
3oru	0.1434	0.1489	0.1102	0.1000	0.1175	0.7142	0.1638
3os6	0.1305	0.1269	0.1135	0.1094	0.0812	0.9162	0.1811
3os7	0.1140	0.1353	0.1170	0.0842	0.0942	0.8753	0.1617
3ot2	0.1443	0.1164	0.1646	0.1682	0.1191	0.7866	0.17
3p1t	0.1243	0.1207	0.0987	0.1098	0.1123	-	0.1757
3pfe	0.1060	0.1096	0.1139	0.1184	0.0943	0.6874	-
3pnx	0.1380	0.1227	0.1088	0.1671	0.1367	0.6229	0.2012
3qtd	0.1029	0.1219	0.1072	0.1228	0.0896	0.9647	0.1539
3voq	-	-	-	-	-	-	-
3nmb	-	-	-	-	-	0.7013	0.1659

Table 9 CASP9 result comparison for LiacsPred, Modeller, and Rosetta.
Each of the five columns represents a different way of calculating the native conformation.
frag_size: based on the largest fragment size
m_nmr: based only on information from nmr e-coli pdb files with missing information
m_x-ray: based only on information from x-ray e-coli pdb files with missing information
nm_nmr and **nm_x-ray** follow the same logic only pdb files with no missing information has been used.
Modeller: results by Modeller
Rosetta: results by Rosetta

CASP8

target	frag_size	m_nmr	m_x-ray	nm_nmr	nm_x-ray	Modeller
2k3i	0.1545	0.1229	0.1358	0.1930	0.1378	0.1683
2k4m	0.1315	0.1122	0.1474	0.1164	0.1351	0.1379
2k4n	0.1198	0.1562	0.1494	0.1301	0.1755	0.16
2k4v	0.1349	0.1321	0.1469	0.1159	0.1294	0.1493
2k4x	0.1641	0.1308	0.1446	0.1455	0.1407	0.1275
2k53	0.1206	0.1438	0.1371	0.1596	0.1544	0.1659
2k54	0.1360	0.1534	0.1370	0.1246	0.1536	0.1893
2k5c	0.1536	0.1427	0.1545	0.1436	0.1459	0.1627
2k5d	0.1157	0.1433	0.1485	0.1245	0.1310	0.1706
2k5e	0.1616	0.1545	0.1620	0.1669	0.1878	0.1533
2k5i	0.1442	0.1227	0.1581	0.1185	0.1579	0.1566
2k5j	0.1353	0.1560	0.1441	0.1561	0.1403	0.1257
2k5l	-	0.1378	0.1405	0.1533	-	0.1445
2k5r	0.1556	0.1365	0.1258	0.1611	0.1573	0.1622
2k5w	0.1180	0.1270	0.1261	0.1280	0.1085	0.0997
2kdl	0.1882	0.1332	0.1615	0.1882	0.1648	0.1464
2kdm	0.3995	0.1478	0.1638	0.3995	0.1850	0.1463
2vsv	0.1278	0.1397	-	0.1271	0.1483	0.1422
2vsw	-	0.1288	0.1443	0.1333	0.1280	0.1833
2vuw	0.4211	0.0940	0.4042	0.1444	0.1126	0.1615
2vux	0.1413	0.1567	0.1272	0.1467	0.1478	0.1824
2vwr	0.1503	0.1343	0.1187	0.1298	0.1936	0.1572
2vx2	0.1139	0.1047	0.1785	0.1059	0.1286	0.229
2vx3	0.1249	0.1094	0.1155	0.1123	0.1562	0.1406
3cyn	0.1623	0.1027	0.1250	0.1411	0.1079	0.1882
3czp	0.1230	0.1143	0.1204	0.1169	0.1078	0.1432
3czq	0.1371	0.1494	0.1408	0.1272	0.1514	0.1912
3czu	0.1204	0.1356	0.1152	0.0969	0.1190	0.1573
3czx	0.1307	0.1494	0.1451	0.1058	0.1659	0.2048
3d01	0.1128	0.1423	0.1214	0.1323	0.1397	0.1529
3d0f	0.1473	0.1416	0.1847	0.1411	0.1600	0.1711
3d0j	0.1336	0.1641	-	0.1330	0.1156	0.1511
3d0k	0.0972	0.1228	0.1160	0.1247	0.1188	0.166
3d19	0.1515	0.1701	0.1319	0.1415	0.1072	0.2238
3d1l	0.1206	0.1184	0.1471	0.1335	0.1231	0.2047
3d1p	0.1486	0.1098	0.1185	0.1423	0.1411	0.1644
3d37	0.1243	0.0985	0.1038	0.1026	0.1018	0.2011
3d3o	0.1052	0.1262	0.1078	0.1150	0.1243	0.1982
3d3q	0.1865	0.1200	0.1128	0.1394	0.1563	0.1482

3d3s	0.1541	0.1392	0.1495	0.1326	0.1293	0.1763
3d3u	0.0979	0.1158	0.0983	0.1172	0.1010	0.1974
3d3y	0.1332	0.1504	0.1095	0.1270	0.1158	0.1289
3d4e	0.1305	0.1233	-	0.0962	0.1387	0.1334
3d4o	0.1211	0.1208	0.1304	0.1519	0.0955	0.2218
3d4r	0.1490	0.1289	0.1804	0.1544	0.1220	0.161
3d5n	0.1363	0.1227	0.1045	0.1075	0.1345	0.2039
3d5p	0.1714	0.1327	0.1778	0.1300	0.1276	0.212
3d6j	0.1276	0.1392	0.1596	0.1265	0.0935	0.1459
3d6k	0.1674	-	0.0979	0.1102	0.1372	0.1079
3d6w	0.1269	0.1404	0.1414	0.1017	0.1316	0.159
3d7i	0.1505	0.1692	0.1625	0.1597	0.1619	0.1835
3d7l	0.1511	0.1248	0.1160	0.1246	0.1168	0.1625
3d89	0.1391	0.1176	0.1373	0.1355	0.1321	0.1608
3d8b	0.1384	0.1062	0.1391	0.1373	0.1320	0.196
3d8h	0.1210	0.1300	0.1647	0.1052	0.1289	0.1741
3d8p	0.1251	0.1306	0.1295	0.1709	0.1251	0.1897
3d8u	0.1157	0.1457	0.1223	0.1261	0.1119	0.1831
3da1	0.0904	0.1455	0.1118	0.0976	0.0843	0.1738
3da2	0.1032	0.0941	0.1077	0.1146	0.1128	0.1668
3dai	0.1296	0.1582	-	0.1527	0.1274	0.1666
3dal	0.1477	0.1127	0.1165	0.1603	0.1211	0.0873
3dao	0.1331	0.1683	0.1454	0.1093	0.1454	0.2073
3dax	-	0.1036	-	0.1161	0.1638	-
3db0	0.1748	0.1370	0.1420	0.1209	0.1261	0.1537
3db3	0.2309	0.8469	0.1573	0.0931	0.1272	0.1902
3db5	0.1338	0.1261	0.2023	0.1226	0.1364	0.1784
3db9	0.1375	0.1144	0.0767	0.0998	0.0958	0.1471
3dc7	0.1165	0.1636	0.1295	0.1229	0.1106	0.189
3dcd	0.1173	-	0.1398	0.1103	0.1485	0.1496
3dcp	0.1617	0.1117	0.1575	0.1275	0.1293	0.2024
3dcx	0.1871	0.1346	0.1429	0.1467	0.1423	0.1561
3dcy	0.1543	0.1460	0.1220	0.0995	0.1352	0.1527
3ddv	0.1518	0.1496	0.1403	0.1197	0.1394	0.152
3ded	0.1322	0.1254	0.1480	0.1117	0.1468	0.184
3dee	0.1467	0.1484	0.1412	0.1265	0.1438	0.1523
3dev	0.1563	0.1493	0.1566	0.1331	0.1409	0.1876
3dew	0.1603	0.1425	0.1465	0.1063	0.1370	0.1774
3dex	0.1692	0.1379	0.1505	0.1218	0.1499	0.1709
3df8	0.1290	0.1489	0.1623	0.1721	0.1622	0.1846
3dfa	0.1305	0.1263	0.1271	0.1265	0.1118	0.1758

3dfd	0.1475	0.1398	0.1252	0.1510	0.1123	0.1658
3dfe	0.1668	0.1205	0.1289	0.1967	0.1322	0.1808
3dh1	0.1172	0.1319	0.1504	0.1678	0.1309	0.183
3dhn	0.1240	0.1417	0.1372	0.1108	0.1389	0.1701
3di5	0.1617	0.1687	0.1600	0.1283	0.1406	0.15
3djb	0.1726	0.1445	0.1526	0.1194	0.1461	0.1797
3dka	0.1540	0.1507	0.1370	0.1337	0.1417	0.1625
3dkp	0.1130	0.0990	0.1363	0.1383	0.1319	0.1983
3dkz	0.1144	0.1066	0.1308	0.1289	0.1182	0.1407
3dl1	0.1166	0.1131	0.1205	0.1529	0.1141	0.192
3dlb	0.1277	0.1064	0.1279	0.1244	0.0838	0.1463
3dlc	0.1161	0.1208	0.1637	0.1223	0.1484	0.1964
3dlm	0.1417	0.1172	0.0871	0.1206	0.1628	0.2181
3dls	0.1178	0.0967	0.1297	0.1472	0.1418	0.1621
3dm3	0.1023	0.1104	0.1133	0.1502	0.1208	0.155
3dm4	0.1230	0.1486	0.1382	0.1425	0.1257	0.1583
3dma	0.1159	0.1572	0.1048	0.1706	0.1085	0.2105
3dmb	0.1204	0.1256	0.1462	0.1181	0.1420	0.1279
3dmc	0.1572	0.1177	0.1590	0.1639	0.1122	0.1543
3dme	0.1271	0.0888	0.0865	0.1434	0.1223	0.1201
3dmn	0.1284	0.1426	-	0.1359	0.1567	0.1625
3dn7	0.1234	0.1743	0.1626	0.1209	0.1163	0.1593
3dnh	0.1635	0.1148	0.1074	0.1392	0.1304	0.1503
3dnp	0.1467	0.1366	0.1420	0.1567	0.1406	0.2201
3dnx	0.1496	0.1243	0.1554	0.1739	0.1179	0.1577
3do5	0.1724	0.1483	0.1466	0.1470	0.1116	0.1842
3do6	0.1171	0.1053	0.0967	0.1084	0.0996	0.1442
3do8	0.1272	0.1288	0.1390	0.1762	0.1476	0.1503
3do9	0.1227	0.1137	0.1651	0.1469	0.1254	0.1689
3doa	0.1337	0.1329	0.1166	0.1462	0.1326	0.1749
3dou	0.1163	0.1352	0.1401	0.1464	0.1206	0.1699
3dr5	0.1453	0.1077	0.1488	0.1412	0.1504	0.1602
3dsm	0.1338	0.1279	0.0932	0.1317	0.1175	0.1465
3dtd	0.0907	0.1225	0.0873	0.1101	0.1506	0.177
3dup	0.1211	0.1492	0.1676	0.1381	0.1490	0.1661
3e03	0.1117	0.1546	0.1358	0.1029	0.0914	0.215
3e38	0.0846	0.1089	0.0824	0.0937	0.1733	0.2537
3g5a	0.1193	0.1651	0.1357	0.1529	0.1360	0.1833
3gwl	0.1544	0.1639	0.1616	0.1426	0.1410	0.1722

Table 10 CASP8 result comparison for LiacsPred, Modeller, and Rosetta.
The first four are from LiacsPred and the last is for Modeller.

Conclusion

The results from LiacsPred and Rosetta are equal, with Rosetta having only slightly better score, but a fair comparison cannot be made given that we could not run Rosetta in an optimal way duo to not enough computing power. Modeller on the other hand performs good on some targets and bad on others. This is duo to the fact that for the good results it was able to find close homologous matches.

Summary

The third-party predictors produced very different results from each other. In our experiments Modeller significantly outperformed Rosetta. This is could be due to the nature of the Rosetta algorithm and the fact that we did not run it with its optimal parameters. Rosetta heavily relies on its function for calculating the minimum energy of a molecule and refines the model based on the results. The Rosetta algorithm requires 20,000 - 30,000 models to be calculated for each target and in our tests we calculated just 1 (See Chapter 4 for details). On the other hand, the much faster, in terms of processing, algorithm of The Modeller is based on calculating the native conformation by taking into account the experimentally determined native conformation of the targets' homologues.

In the case of The Modeller, it was able to predict some of the native conformations with an accuracy of 90%. and some with an accuracy of 10%. The unreliable nature of this algorithm correlates with its dependency on already solved native conformations coupled with an extra restriction that they have to be of closely related homologous. This might work for a range of proteins, but for a large number of them it will deliver abysmal results. Of course, this tool can be restricted to the calculation of the native conformation of targets for whose closely related homologous have their native conformations experimentally determined. This test can be quite reliable as tools such as HHSuite, BLAST, FASTA are very good at identifying homologs.

Our predictor, LiacsPred, produced results in the same domain as Rosetta. Part of the problem can be easily explained. We used an algorithm which calculates the atomic coordinates based on pre-calculated coordinates of amino acids from experimentally determined native conformations. We set out to test if limiting the range of the training set of our predictor can produce satisfactory results. The reasoning behind this decision lies within several research papers which point that the expression organism, the external and the internal factors, and the experimental setup for obtaining the native conformation are very important. At present we were not able to achieve desired results, because there are not enough experimentally determined native conformation of proteins from the Escherichia coli organism. Moreover, the imposed restriction on using experimentally determined native conformations obtained by X-Ray Crystallography narrowed our training set even further. Not all atomic coordinates were predicted with information gathered from the pdb files of Escherichia coli. From this it could be argued that with more training samples, at least double the amount, we could obtain much better results. This was the case because we were not able to find all the necessary amino acid fragments in those files and had to use information from other species.

It should be noted that for future experiments the best course of action is to couple, together with LaicsPred, the same or similar energy function like the one of Rosetta and a homologous discriminator such as the one used by The Modeller. We hope that LiacsPred will one day be able to calculate what the native conformation for a protein will be given different expression systems.

Appendix A

Common Amino Acids

Essential	Nonessential
Histidine	Alanine
Isoleucine	Arginine
Leucine	Asparagine
Lysine	Aspartic acid
Methionine	Cysteine
Phenylalanine	Glutamic acid
Threonine	Glutamine
Tryptophan	Glycine
Valine	Ornithine
	Proline
	Selenocysteine
	Serine
	Taurine
	Tyrosine

References

1 Brocchieri L, Karlin S (2005-06-10). Protein length in eukaryotic and prokaryotic proteomes. Nucleic Acids Research 33 (10): 3390–3400.

2 Pauling L, Corey RB, Branson HR (1951). The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. Proc Natl Acad Sci USA 37 (4): 205–211.

3 Alberts, Bruce; Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walters (2002). The Shape and Structure of Proteins. Molecular Biology of the Cell; Fourth Edition.
4 Anfinsen CB. (20 July 1973). Principles that Govern the Folding of Protein Chains. Science. 181 (4096): 223–230.

5 van den Berg, B., Wain, R., Dobson, C. M., Ellis R. J. (August 2000). Macromolecular crowding perturbs protein refolding kinetics: implications for folding inside the cell. EMBO J. 19 (15): 3870–5.

6 Alexander, P. A., He Y., Chen, Y., Orban, J., Bryan, P. N. (2007). The design and characterization of two proteins with 88% sequence identity but different structure and function.

7 Schneider M, Fu X, Keating AE. X-ray vs. NMR structures as templates for computational protein design.

8 Girish S. Ratnaparkhi, S. Ramachandran,| Jayant B. Udgaonkar,| and R. Varadarajan Discrepancies between the NMR and X-ray Structures of Uncomplexed Barstar. “Analysis Suggests That Packing Densities of Protein Structures Determined by NMR Are Unreliable”.

9 Zhong L, Johnson WC Jr (1992). Environment affects amino acid preference for secondary structure. Proc Natl Acad Sci USA 89 (10): 4462–5.

10 Macdonald JR, Johnson WC Jr (2001).Environmental features are important in determining protein secondary structure. Protein Sci. 10 (6): 1172–7.

11 Costantini S, Colonna G, Facchiano AM (2006). Amino acid propensities for secondary structures are influenced by the protein structural class. Biochem Biophys Res Commun. 342 (2): 441–451.

12 Marashi SA, et al. (2007). Adaptation of proteins to different environments: a comparison of proteome structural properties in Bacillus subtilis and Escherichia coli. J Theor Biol 244 (1): 127–132.

13 Zhang Y and Skolnick J (2005). The protein structure prediction problem could be solved using the current PDB library. Proc Natl Acad Sci USA 102 (4): 1029–34.

14 Bioinformatics toolkit. (2011) About HHPred. [ONLINE]. Available from: http://toolkit.tuebingen.mpg.de/hhpred/help_ov [Accessed 1st of April 2013]

15 Y. Zhang, J. Skolnick, Scoring function for automated assessment of protein structure template quality, Proteins, 2004 57: 702-710

16 J. Xu, Y. Zhang, How significant is a protein structure similarity with TM-score=0.5? Bioinformatics, 2010 26, 889-895

17 Biological Macromolecule Resource. (n.d.) PDB Standard. [ONLINE]. Available from: http://www.rcsb.org/pdb/home/home.do [Accessed 1st of April 2013]

18 Hammarstrom, P., et al., Prevention of Transthyretin Amyloid Disease by Changing Protein Misfolding Energetics. Science, 2003. 299(5607): p. 713-716.

19 Dennis J. Selkoe (2003). Folding proteins in fatal ways. Nature 426 (6968): 900–904.

20. CASP. (n.d.) CASP scoring methods. [ONLINE]. Available from: http://prodata.swmed.edu/CASP9/evaluation/Scores.htm. [Accessed 1st of April 2013]

21 CASP (2011) CASP9 experiment ranking. [ONLINE]. Available from: http://predictioncenter.org/casp9/groups_analysis.cgi?type=server&tbm=on&tbmfm=on&fm=on&submit=Filter [Accessed 1st of April 2013]

22 Pervushin K, Riek R, Wider G, Wüthrich K (November 1997). Attenuated T2 relaxation by mutual cancellation of dipole-dipole coupling and chemical shift anisotropy indicates an avenue to NMR structures of very large biological macromolecules in solution. Proc. Natl. Acad. Sci. U.S.A. 94 (23): 12366–71.

23 Markus MA, Dayie KT, Matsudaira P, Wagner G (October 1994). Effect of deuteration on the amide proton relaxation rates in proteins. Heteronuclear NMR experiments on villin 14T. J Magn Reson B

24 Pervushin K, Riek R, Wider G, Wüthrich K (November 1997). Attenuated T2 relaxation by mutual cancellation of dipole-dipole coupling and chemical shift anisotropy indicates an avenue to NMR structures of very large biological macromolecules in solution. Proc. Natl. Acad. Sci. U.S.A. 94 (23): 12366–71.

25 Harp, JM; Timm, DE; Bunick, GJ (1998). Macromolecular crystal annealing: overcoming increased mosaicity associated with cryocrystallography. *Acta crystallographica D* **54** (Pt 4): 622–8.

26 Cross, G; Reeves, AA; Brand, S; Popplewell, JF; Peel, LL; Swann, MJ; Freeman, NJ (2003). A new quantitative optical biosensor for protein characterisation.

27 P. Malon, R. Kobrinskaya, T. A. Keiderling (1988). Vibrational Circular Dichroism of Polypeptides XII. Re-evaluation of the Fourier Transform Vibrational Circular Dichroism of Poly-gamma-Benzyl-L-Glutamate. *Biopolymers* **27** (5): 733–746.

28 S. C. Yasui, T. A. Keiderling (1988). Vibrational Circular Dichroism of Polypeptides and Proteins.*Mikrochimica Acta* **II**: 325–327.

29 Laurence A. Nafie (2008). Vibrational Circular Dichroism: A New Tool for the Solution-State Determination of the Structure and Absolute Configuration of Chiral Natural Product Molecules. *Natural Product Communications* **3** (3): 451–466

30 LiveBench. (n.d.) LiveBench. [ONLINE]. Available from: http://meta.bioinfo.pl/livebench.pl [Accessed 1st of April 2013]

31 Eva. (n.d.) Eva. [ONLINE]. Available from: http://www.pdg.cnb.uam.es/eva/sec/res_sec.html [Accessed 1st of April 2013]

32 Ibba M, Söll D (May 2001). The renaissance of aminoacyl-tRNA synthesis. *EMBO Reports* **2** (5): 382–7. doi:10.1093/embo-reports/kve095 (inactive 2010-02-18).

33 Joseph P. Hendrik and F.-Ulrich Hartl. The role of molecular chaperones in protein folding.

34 Sevier, C. S. and Kaiser, C. A. (2002). Formation and transfer of disulphide bonds in living cells. Nature Reviews Molecular and Cellular Biology 3 (11): 836–847.

35 Dougherty, Dennis A. (2006). Modern Physical Organic Chemistry. Sausalito, CA: University Science Books.

36 Nic, M.; Jirat, J.; Kosata, B., eds. (2006–). hydrogen bond. IUPAC Compendium of Chemical Terminology(Online ed.).

37 Tooze, John; Brändén, Carl-Ivar (1999). Introduction to protein structure. New York: Garland Pub.

38 R. John Ellis Molecular chaperones: assisting assembly in addition to folding. Trends in Biochemical Sciences 31 (7): 395–401.

39 J. Cavanagh, W.J. Fairbrother, A.G. Palmer III, N.J. Skelton: Protein NMR Spectroscopy Academic Press (1996)

40 Carter, Charles W. Jr. and Robert M. Sweet eds. Methods in Enzymology. 276, [2] (1997). Rhodes, Gale. Crystallography Made Crystal Clear. Academic Press, San Diego, 1993.

41 Swann MJ, Peel LL, Carrington S, Freeman NJ Dual-polarization interferometry: an analytical technique to measure changes in protein structure in real time, to determine the stoichiometry of binding events, and to differentiate between specific and nonspecific interactions.

42 P. Pancoska, L. Wang, and T. A. Keiderling Frequency analysis of infrared absorption and vibrational circular dichroism of proteins in D2O solution.

43 De Maio A (January 1999). Heat shock proteins: facts, thoughts, and dreams. Shock (Augusta, Ga.) 11(1): 1–12.

44 DSSP. (n.d.) DSSP. [ONLINE]. Available from: http://swift.cmbi.ru.nl/gv/dssp/ [Accessed 1st of April 2013]

45 IBM Research. (n.d.) Blue Gene. [ONLINE].Available from: http://www.research.ibm.com/bluegene/index.html [Accessed 1st of April 2013]

46 Curiosity. (n.d.). MDGrape-3. [ONLINE].Available from: http://curiosity.discovery.com/question/what-is-mdgrape-3 [Accessed 1st of April 2013]

47 FoldIt. (2010). FoldIt. [ONLINE].Available from: http://folding.stanford.edu/English/HomePage [Accessed 1st of April 2013]

48 Blast. (n.d.). Blast. [ONLINE].Available from: http://blast.ncbi.nlm.nih.gov/ [Accessed 1st of April 2013]

49 FASTA. (n.d.) FASTA. [ONLINE].Available from: http://www.ebi.ac.uk/Tools/sss/fasta/ [Accessed 1st of April 2013]

50 Modeller. (n.d.). Modeller. [ONLINE].Available from: http://www.salilab.org/modeller/ [Accessed 1st of April 2013]

51 CASP. (n.d.). CASP. [ONLINE].Available from: http://predictioncenter.org/ [Accessed 1st of April 2013]

52 ProQ2. (n.d.). ProQ2. [ONLINE].Available from: http://proq2.theophys.kth.se/index.php?about=proqm [Accessed 1st of April 2013]

53 The Baker Laboratory. (n.d.). The Baker Laboratory. [ONLINE].Available from: http://depts.washington.edu/bakerpg/drupal/ [Accessed 1st of April 2013]

54 Laboratory of theory of Biopolymers. (n.d.). Kloczkowski. [ONLINE].Available from: http://biocomp.chem.uw.edu.pl/publications.php?nazwisko=A.%20Kloczkowski [Accessed 1st of April 2013]

55 Centro Nacional de Investigaciones Oncologicas. (n.d.). CNIO. [ONLINE].Available from: http://www.cnio.es/ing/ [Accessed 1st of April 2013]

56 Zhang initiative research unit. (n.d.). Zhang laboratory. [ONLINE].Available from: http://www.riken.jp/zhangiru/index.htm [Accessed 1st of April 2013]l