Protein Model Quality Assessment – A Machine Learning Approach

Karolis Uziela


Stockholm
University

# Protein Model Quality Assessment

A Machine Learning Approach

Karolis Uziela

# Abstract

Many protein structure prediction programs exist and they can efficiently generate a number of protein models of a varying quality. One of the problems is that it is difficult to know which model is the best one for a given target sequence. Selecting the best model is one of the major tasks of Model Quality Assessment Programs (MQAPs). These programs are able to predict model accuracy before the native structure is determined. The accuracy estimation can be divided into two parts: global (the whole model accuracy) and local (the accuracy of each residue). ProQ2 is one of the most successful MQAPs for prediction of both local and global model accuracy and is based on a Machine Learning approach.

In this thesis, I present my own contribution to Model Quality Assessment (MQA) and the newest developments of ProQ program series. Firstly, I describe a new ProQ2 implementation in the protein modelling software package Rosetta. This new implementation allows use of ProQ2 as a scoring function for conformational sampling inside Rosetta, which was not possible before. Moreover, I present two new methods, ProQ3 and ProQ3D that both outperform their predecessor. ProQ3 introduces new training features that are calculated from Rosetta energy functions and ProQ3D introduces a new machine learning approach based on deep learning. ProQ3 program participated in the 12th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction (CASP12) and was one of the best methods in the MQA category. Finally, an important issue in model quality assessment is how to select a target function that the predictor is trying to learn. In the fourth manuscript, I show that MQA results can be improved by selecting a contact-based target function instead of more conventional superposition based functions.



ISBN 978-91-7649-633-6

Printed in Sweden by US-AB, Stockholm 2017

Distributor: Department of Biochemistry and Biophysics, Stockholm University

*To my family*

# List of Papers

The following papers, referred to in the text by their Roman numerals, are included in this thesis.

PAPER I: **ProQ2: estimation of model accuracy implemented in Rosetta**
Uziela K., Wallner B. (2016) *Bioinformatics*, **32**(9), 1411-1413 (2016).
DOI: 10.1093/bioinformatics/btv767

PAPER II: **ProQ3: Improved model quality assessments using Rosetta energy terms**
Uziela K., Shu N., Wallner B., Elofsson A. *Nature Scientific Reports* **6**, 33509 (2016).
DOI: 10.1038/srep33509

PAPER III: **ProQ3D: Improved model quality assessments using Deep Learning**
Uziela K., Menéndez Hurtado D., Shu N., Wallner B., Elofsson A. *Bioinformatics* (in press).
DOI: 10.1093/bioinformatics/btw819

PAPER IV: **Improved protein model quality prediction by changing the target function**
Uziela K., Menéndez Hurtado D., Shu N., Wallner B., Elofsson A. (manuscript in preparation).

---

The following work has not been included into this thesis:

**Probe Region Expression Estimation for RNA-Seq Data for Improved Microarray Comparability**
Uziela K, Honkela A. *PLoS ONE*, **10**(5), e0126545 (2016).
DOI: 10.1371/journal.pone.0126545

# Contents

# 1. Introduction

Imagine a thread with 20 different types of beads attached to it. This thread can fold bringing some of these beads together while keeping the others apart. All of these beads have different properties and some of them like to be brought close to each other, but some of them do not. Yet, the thread always folds in such a way, that all of the beads are satisfied.

The above is a very simplified description of a protein folding process that occurs naturally in every living cell. The thread is the polypeptide and the beads are different amino acids that it consists of. The polypeptide could fold in many ways, but it always acquires the same structure that is solely dependent on the amino acid sequence [1].

Protein function is tightly associated with its three dimensional structure. Therefore, determining protein structure experimentally is an important but, unfortunately, very expensive and time consuming task. Determining the DNA sequence that directly corresponds to the amino acid sequence of a specific protein is much easier. Therefore, many computer methods have been developed for predicting protein structure from the amino acid sequence. These computer programs can rapidly generate hundreds or even thousands of protein models. However, the challenge comes when it is necessary to assess the reliability of the models and select the best one out of all that are available. This is where Model Quality Assessment Programs (MQAPs) can help.

The MQAPs have two tasks: (i) evaluating the overall (global) quality of the protein model and (ii) evaluating the quality of the specific parts of the protein model (local quality). Both tasks are important for different reasons. The global quality estimation allows evaluation of the overall reliability of the model and to select the best model out of several possible choices. The local quality estimation, on the other hand, lets us determine which regions are likely to be modelled correctly and which ones are not reliable. The low scoring regions might be subject to protein structure refinement or simply ignored in the further application of the protein structure.

Machine learning is a computer science discipline that can be applied in many areas, such as image and speech processing, spam detection, recommendation of goods to customers, etc. Many of MQAPs employ machine learning algorithms [2–10], including ProQ [2] and ProQ2 [3], which have been one of the most successful methods so far. In this thesis I present the most recent

developments of the ProQ program series as well as discuss important issues related to the protein model quality assessment in general.

# 2. Biological background

## 2.1 Protein structure

Proteins are one of the most important and complex molecules in the cell, as they carry out nearly every function. The transport of molecules in and out of the cell is performed by specific transporters or channels, the interaction with the environment is done by protein receptors, signals are transferred by signalling cascades, the chemical reactions are catalysed by proteins called enzymes, etc.

Despite being able to perform so many different functions, all proteins share the same basic structure. They are all made from 20 different amino acids connected in a chain. The chain folds itself allowing different amino acids to interact with each other and the protein in this way acquires a three-dimensional shape.

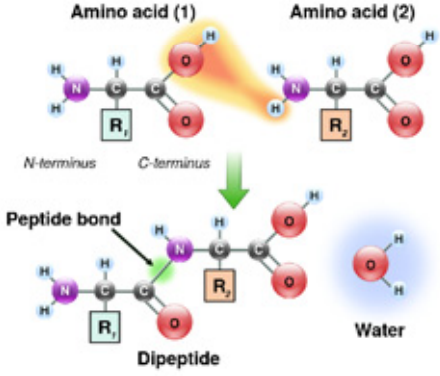



**Figure 2.1:** Peptide bond formation. Amino acid residues are denoted by R1 and R2. Image source: `https://en.wikipedia.org/wiki/Peptide_bond`

The amino acids in the protein are joined by peptide bonds. The peptide bond is formed when two amino acids join and the water is dissolved during the process (see Figure 2.1). Residues that make one amino acid different from another are always attached to a carbon atom, which is called C-$\alpha$. Two C-$\alpha$ atoms are connected by a peptide unit, which consists of C, O, N and H atoms. The peptide unit always remains in a plane, but the bonds that connect

it to the C-$\alpha$ atom can rotate. The rotation angles of these bonds are called $\Phi$ and $\Psi$ and they are the only flexible regions of the peptide chain. Because of that, knowing the sequence of $\Phi$ and $\Psi$ angles is sufficient to reconstruct the structure of the protein backbone [1].

The most basic structural motifs in a protein are called secondary structure elements. The two most common types of these elements are the $\alpha$-helix and $\beta$-sheet, which form because of regular hydrogen bonding patterns between the atoms in the protein backbone. Random coils are not considered to be secondary structure elements themselves, but rather connecting units that most often connect $\alpha$-helices and $\beta$-sheets. In graphical representations, a spiral usually denotes the $\alpha$-helix and an arrow denotes the $\beta$-sheet. These secondary structure elements fold onto each other forming the tertiary structure of a protein.

## 2.2 Protein folding

The protein is a long chain of amino acids connected by peptide bonds. After the protein is produced it rapidly folds into a structure that minimizes the free energy of the molecule. Some of the folding might already occur during the process of producing the protein—one end might start folding while the other end is still being synthesised. The final three-dimensional structure in general depends solely on the sequence of amino acid residues [1].

When the protein folds it is driven by several forces [11]. The main ones are:

- **Hydrogen bonds.** The hydrogen bonds between the protein backbone atoms are the main reason of forming secondary structure elements that stabilise the structure.
- **Van der Waals interactions.** The atoms in the structure core are tightly packed and stabilised by van der Waals interactions.
- **Backbone dihedral angle preferences.** Certain $\Phi$ and $\Psi$ angles are preferred over the others.
- **Electrostatic interactions.** Some of the residues contain positive and negative charges that constitute the attraction and repulsion forces. Partial residue charges might also contribute.
- **Hydrophobic effect.** Hydrophobic residues tend to end up in the core of a folded protein, while the hydrophilic residues remain outside. If hydrophobic molecules remained outside, they would disrupt the dynamic

hydrogen bonding patterns of water molecules, which would result in decreased entropy.

Cysteine residues also play an important role in protein folding, because they can form disulphide bridges once the protein is folded, which stabilises the final conformation. Disulphide bridges are common in proteins secreted to extracellular medium [12].

In 1969, Cyrus Levinthal noted that a protein has a huge number of degrees of freedom and would never be able to sample all possible conformations when folding [13; 14]. For example, a protein that consists of 100 amino acids will have 99 peptide bonds and 198 $\Phi$ and $\Psi$ angles. Even if each angle was only able to acquire 3 stable conformations, the total number of possible conformations would be $3^{198} = 2.95 * 10^{94}$. In practise, $\Phi$ and $\Psi$ angles are not independent, but for each $\Phi$ there are definitely at least 3 possible $\Psi$ angles, so the above calculation holds. So even if the protein could sample each conformation at a rapid pace of 1 conformation per picosecond, then it still would need more time than the age of universe to fold. Paradoxically, a protein folds on a timescale of milliseconds or sometimes even microseconds. Levinthal suggested that folding is guided by formation of local interactions that restrict the conformation space and reduce the time of folding.

# 3. Protein structure prediction

## 3.1 Motivation

Determining an amino acid sequence experimentally is much easier than determining a protein structure. Therefore, the gap between the number of proteins with a known sequence and a known structure has been widening for a long time (see Figure 3.1). With the recent advances in DNA sequencing technologies, the gap is likely to increase even further [15].

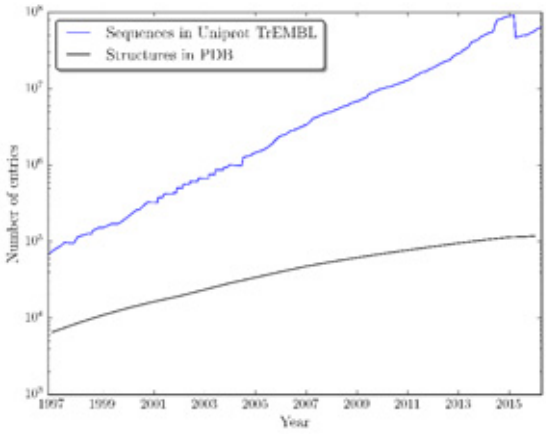



**Figure 3.1:** The growth of protein sequence and structure data bases. PDB (Protein Data Bank) is a protein structure data base. UniprotKB/TrEMBL is an automatically annotated protein sequence data base. Plot author: David Menéndez Hurtado.

Today, the most common technologies for protein structure determination are X-ray crystallography and NMR spectroscopy. Both of these are rather expensive, time consuming and they do not guarantee to produce a reliable protein structure [16]. The newest addition to the experimental protein structure determination is the Cryo-EM technology. Just like NMR, this technology does not require for the protein to be crystallised before its structure is determined. The downside is that Cryo-EM can only be used to determine the

structure of sufficiently large proteins and the experimental cost is even higher than X-ray crystallography and NMR spectroscopy.

An alternative approach would be to model protein structure *in silico*. That is, given the protein sequence, predict the protein structure. According to Anfinsen's dogma there is only one stable protein structure that corresponds to a given sequence and a protein acquires that structure *in vivo* [1].

Knowing the exact protein structure can be useful for many reasons. Firstly, the function of the protein in the cell is always related to the structure, so by knowing its structure we can greatly increase our understanding of that protein's biological significance. Secondly, knowing structure of the protein can be beneficial for industrial applications. For example, many drug targets are proteins, so by knowing the structure we can design a ligand that would inhibit a protein activity. Moreover, knowing protein structure can be useful for biotechnology applications, because we can introduce amino acid substitutions in the protein sequence that would change the structure in a meaningful way.

## 3.2 Comparative modelling

Protein structure prediction methods can be broadly divided into two groups: comparative modelling and *de novo* modelling. Comparative modelling uses a protein with a known three dimensional structure as a template for the prediction. *De novo* methods try to model a protein structure from scratch without any given template. Comparative modelling is usually more accurate if a good template can be found. However, *de novo* modelling is useful for the cases when no good quality template is available.

Comparative modelling can be divided into two groups based on how the template is selected:

- **Homology modelling.**
- **Threading (or fold recognition).**

In **homology modelling** the template is selected based on sequence similarity. This approach relies on the fact that proteins with similar sequences will have similar folds [17].

Choosing a correct template is a very important step in the homology modelling, because if a mistake is made at this stage it cannot be fixed later and it will greatly affect the model quality. Many different algorithms exist that can perform this step, among which probably the most popular ones are sequence profile based methods such as the classical Psi-blast [18] method, or the more advanced Jackhmmer [19] and HHblits [20] methods. The early methods, such

as Blast [21] or FASTA [22] used only the sequence information itself to search for homologues, but later it was noticed that the search sensitivity and specificity could be improved if a sequence profile was used instead of only the sequence itself. Sequence profile captures the evolutionary information, so it contains more information than just the sequence itself. Sequence profile can be used only for the query sequence (as it is done in Psi-blast), or both in query and data base sequence (as it is done in Jackhmmer and HHblits). In the early development stages profile-profile search methods were rather slow, but the newest methods do not lag much behind the sequence-profile or even sequence-sequence methods and provide much more accurate results [20].

Another important step in a homology modelling is choosing a correct alignment of the query sequence to the template. Most of the homology search algorithms, such as Psi-blast, Jackhmmer and HHblits, already provide an alignment between the query and the template, but some other alignment programs or manual inspection can be used to refine the given alignment. There are some rules of thumb, such as charged residues should not fall into the core of the protein unless they are compensated by other residues of an opposite charge or there is a functional reason for them to be in the core [17]. Also, insertions and deletions should not fall in the middle of the secondary structure elements. Similarly to template selection phase, the error in this step will have grave effects on the final model.

**Threading** uses a similar approach as homology modelling, but the ways the template is selected and alignment is made are different. Selecting the template based on sequence homology proved to be a very robust criterion, but sometimes there are no detectable sequence homologues. On the other hand, the number of possible folds a protein can acquire seems to be rather limited, so there are many proteins that share the same fold, even though their sequences might not be evolutionary related. The main idea of threading is to try to align a sequence to one of the folds based on the probability of the query residues acquiring the given template structure. The algorithm tries to align the query to all of the entries in a non-redundant fold data base and selects the fold that gives the highest sequence to structure alignment score. A dynamic programming algorithm similar to Needleman-Wunch [23] is often used to align the sequence to the structure, however, some modifications have to be made because of different scoring schemes. In Needleman-Wunch the scoring scheme depends only on a single sequence position, however, in threading it depends on several positions, because we need to evaluate the likelihood of a sequence fragment to acquire a certain structural motif [24].

After the template and alignments are selected, the next step is the actual modelling of the protein structure, which is done similarly both in homology modelling and threading. The modelling of the backbone of the aligned

residues is usually a rather easy process, because most of the time it is enough just to copy the aligned part from the template to the query. Modelling of the insertions and deletions is more complicated and it is usually done by inserting loops.

The loops for insertion can be taken from another homologous structure with the same insertion as the target protein. However, if the insertion is unique, the loop is modelled by searching a data base of structural fragments of the same length. Two flanking residues from each side of the insertion are often taken as the anchor points, which are used to align the new fragment. The new fragment is superimposed with these residues and the RMSD (Root Mean Square Deviation) is calculated between the two terminal residues of the fragment from each side and the anchoring residues. The fragment that has lowest RMSD, most similar sequence with the target and interferes least with the target structure is selected and annealed to the model [17].

Short deletions can be dealt with by energy minimization that brings the ends of deleted region together. If the deletion is large, it is more difficult to deal with, unless the ends of the deleted region are close together. Otherwise, it might be an indication that the model is incorrect [17].

Nowadays, threading programs are rarely used. Programs like Jackhmmer [19] and HHblits [20] can detect even very remote homologs that can be used as templates, which made the above described threading template selection method partly obsolete.

## 3.3 De Novo modelling

An alternative approach to comparative modelling is *de novo* modelling where no protein template is used. The classical *de novo* approach is all-atom molecular dynamics simulation where the protein folding problem is tackled by using first principles of protein folding. Using laws of classical physics, the forces and velocities of all particles in the system are modelled by the laws of classical physics. The collection of functional forms and parameters used to evaluate a potential energy of a given conformation is usually referred as a force field [17]. There have been a variety of force fields developed for the problem of protein folding, among the most popular ones are CHARMM [25] and AMBER [26]. As alternatives to molecular dynamics, Monte Carlo sampling or simulated annealing can be used to search the conformational space for the lowest energy structure [17].

Perhaps the most effective template-free modelling approach is fragment-based methods. The protein is split into short fragments and a structural data base is searched for fragments with similar sequences. In the end, fragments are assembled together and the feasibility of the model is evaluated by a scor-

ing function [27]. The most popular fragment-based method is ROSETTA [28]. It uses random replacements of structure with 3 and 9 residues long fragments derived from PDB. Other popular fragment-based methods include I-TASSER [29] and FRAGFOLD [30].

Recently, there has been a breakthrough in contact prediction methods that are based on evolutionary information [31–34]. The main idea of these methods is that if a mutation causes a substitution of one residue that is close in space to another, there will be a compensatory mutation leading to a substitution of the second residue. Therefore, residues that are in contact will have correlated substitutions in a multiple sequence alignment. The residues that are predicted to be in contact can be used as a constraint for *de novo* folding protocol [35].

## 3.4 CASP

The Critical Assessment of protein Structure Prediction (CASP) is a community-wide experiment that aims to evaluate methods in the protein structure prediction field [36]. The competition type experiment is held every two years during summer and it is done in collaboration with experimental structure determination teams. The amino acid sequences of protein structures that are soon going to be determined by X-ray or NMR methodologies are released to the CASP participants and they have a fixed amount of time to submit their protein structure models that correspond to these sequences. After the experimental protein structure coordinates are released, the CASP organisers evaluate all of the predictions. In the end of the same year (usually December), a CASP meeting is organised where the organisers announce the results and the most successful participants present their methods.

Protein structure prediction in CASP is divided into two main categories: server prediction and human prediction. The server prediction category was originally organised as a separate experiment (CAFASP [37]), but nowadays it has effectively merged into CASP. In this category, only fully automated method servers are allowed to participate and the deadline for submitting the predictions is very short—2 days. In the human category, the model submission deadline is longer (usually 3-4 weeks) and the human experts are allowed to use any of the server predictions that are all made available to them.

Protein structure predictions in CASP are further divided into two categories based on the target difficulty. The high Accuracy Modeling category includes so called "easy" targets where most of the groups have submitted sufficiently accurate models [38]. These models are carefully analysed to evaluate main chain, side chains, atomic accuracy, and contacts, as well as hydrogen bonds and covalent geometry. This category was formerly called "template

based", because most of the easy targets usually have a homologous protein with 3D coordinates available, so this protein can be used as a template in comparative modelling (see section 3.2). The other category is called "Topology" and it includes "hard" targets where most of the groups have submitted models of relatively low accuracy. Most of the time these targets do not have any homologous protein with a known 3D structure, therefore, this category used to be called "Free Modelling".

CASP also has several other categories that directly or indirectly are related to protein structure prediction [38]:

- The **Biological Relevance** category, in which models are assessed on the basis of how well they provide answers to biological questions.
- The **Data Assisted** category assesses how much model accuracy can be improved with the use of sparse data, such as simulated and actual sparse NMR data, crosslinking data, and low angle X-ray scattering data.
- The **Contact Prediction** category, in which the ability of methods to predict contacts between residues in the target structures is assessed.
- The **Refinement** category, in which the ability to refine the quality of a given structural model is assessed. The starting model is usually one of the best server predictions from the initial stage.
- The **Assembly** category assesses how well current methods can determine domain-domain, subunit-subunit, and protein-protein interactions. There is also a separate CAPRI [39] experiment that solely focuses on solving this problem.
- The **Accuracy Estimation** category, in which the ability of methods to predict the accuracy of a given protein model is assessed. This category is also sometimes referred as Model Quality Assessment (MQA) category.

## 3.5 CAMEO

CAMEO [40] is a similar protein structure prediction experiment as CASP, but with a few differences. The main difference is that CASP takes place every two years while CAMEO runs non-stop all the time. Every week PDB announces the sequences of the proteins whose three dimensional structures will be released. The CAMEO server sends these sequences to all participating structure prediction servers. After the structures are released (usually within a week), the predictions are evaluated. The CAMEO server allows the user

to examine the methods' performance within 1 week, 1 month, 3 months, 6 months or 1 year time range. CAMEO evaluates methods in three categories: structure prediction, model quality assessment and contact prediction.

## 3.6 Pcons and ProQ methods history and introduction of model quality assessment in CASP

In the first three rounds of CASP human experts always outperformed automated server methods. However, in CASP4, the server methods started to compete with the human experts. Only 11 out of 103 human groups have outperformed the best server method (3D-PSSM). Moreover, at the 7th place there was a semi-automated CAFASP-CONSENSUS method [37]. The idea of CAFASP-CONSENSUS method was then implemented into a fully automated server Pcons [41].

In the beginning Pcons used 6 different servers and a neural network approach to select the best model out of the available ones [42]. Pcons has translated the reported confidence scores by servers into uniformly scaled values and used them together with model similarity to select the best model [41]. Partly because of good performance of Pcons and other similar servers, it was soon realised that selecting the best model out of several possible ones is a very important problem. One approach to solve this problem would be to assess models based on their similarity to each other, as it was done in Pcons and 3DJury [43]. Initially, the difference between Pcons and 3DJury was that 3DJury used a simple average similarity between the models, while Pcons used a more involved neural network approach, however, later Pcons was simplified and used the same approach as 3DJury [44]. An alternative approach would be to assess the model quality only based on the physico-chemical properties of the protein model itself. One of the first Model Quality Assessment (MQA) methods that employed this approach was ProQ [2].

In CASP5 Björn Wallner and Arne Elofsson participated with servers Pcons and Pmodeller [42]. Both of them were meta-servers and the only difference between them was how the best model was chosen among the available ones. Pcons used the above-described approach, while Pmodeller combined Pcons with ProQ. A simple linear combination of model quality scores was used: $Pmodeller = 0.75 * Pcons + 0.17 * ProQ$ and the model with the highest score was chosen. Pmodeller has demonstrated an outstanding performance in CASP5, outperforming most human experts. Also, Pmodeller has performed consistently better than Pcons.

With an increasing understanding of the importance of model quality assessment in the field, it was introduced as a separate category in CASP7. The

participating groups are evaluated in two sub-categories: the ability to predict the quality of the whole protein model (global quality) and the ability to predict the quality of each residue (local quality).

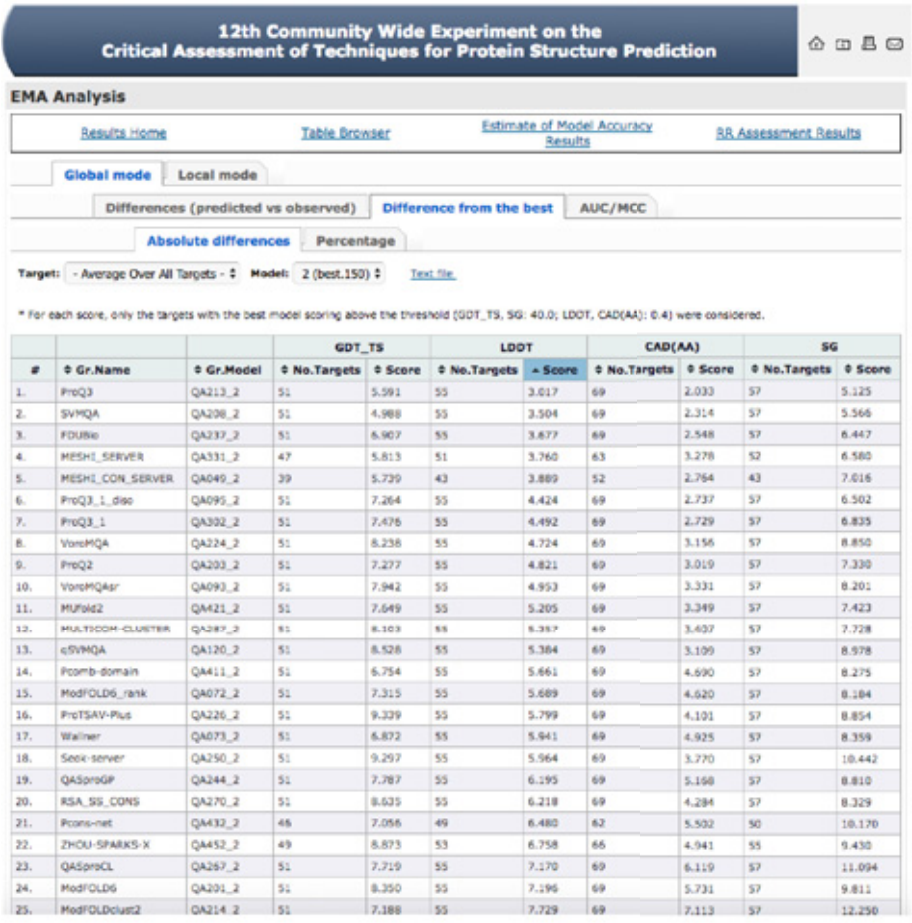
Pcons continues to successfully participate in CASP experiments both alone and in a combination with ProQ methods [45–47]. The combination of ProQ and Pcons was renamed from Pmodeller to Pcomb. There has been several new ProQ version developed: ProQ2 [3], ProQ3 [9] and ProQ3D [10] while Pcons method remained stable since CASP8. Even though, Pcons uses a simple approach, it remains among the best performing consensus methods in Model Quality assessment category [45–47].

## 3.7 CASP12 results

In CASP12, we have participated in MQA category with several ProQ versions including ProQ3 and the preliminary version of ProQ3D, as well as the classical Pcons method and Pcomb method that combines ProQ3 and Pcons using the formula $Pcomb = 0.8 * Pcons + 0.2 * ProQ3$. At the time of writing, the results are already presented at the CASP12 meeting in Gaeta, Italy and the automatic evaluation is available online, but the papers presenting the evaluation are not yet published. However, it is already clear that ProQ3 performed very well in the CASP12 MQA category. In some cases of global evaluation, it outperformed not only single-model methods, but also consensus methods. For example, when evaluating the absolute score difference from the best model according to LDDT measure, ProQ3 method has ranked in the first place among all of the methods (see Figure 3.2). In the local model quality evaluation, consensus methods still significantly outperform the single model methods. Here, Pcons and Pcomb demonstrated very good results.

The preliminary version of ProQ3D (which was named ProQ3_1) in some cases (for example, model selection) performed worse than ProQ3. The reason might be that it was not well optimised at the time of CASP12.

We have also participated with a predictor RSA_SS_CONS. This is a baseline predictor that only uses relative surface area accessibility and secondary structure agreements as well as conservation. It was interesting to see that such a simple predictor performed better than many of the more advanced methods.



12th Community Wide Experiment on the
Critical Assessment of Techniques for Protein Structure Prediction

EMA Analysis

Results Home | Table Browser | Estimate of Model Accuracy Results | RR Assessment Results

Global mode | Local mode

Differences (predicted vs observed) | Difference from the best | AUC/MCC

Absolute differences | Percentage

Target: - Average Over All Targets - Model: 2 (best.150) Text file

* For each score, only the targets with the best model scoring above the threshold (GDT_TS, SG: 40.0; LDDT, CAD(AA): 0.4) were considered.

| | | | GDT_TS | | LDDT | | CAD(AA) | | SG | |
|---|---|---|---|---|---|---|---|---|---|---|
| # | Gr.Name | Gr.Model | No.Targets | Score | No.Targets | Score | No.Targets | Score | No.Targets | Score |
| 1. | ProQ3 | QA213_2 | 51 | 5.591 | 55 | 3.017 | 69 | 2.033 | 57 | 5.125 |
| 2. | SVMQA | QA208_2 | 51 | 4.988 | 55 | 3.504 | 69 | 2.314 | 57 | 5.566 |
| 3. | FDUBio | QA237_2 | 51 | 6.907 | 55 | 3.677 | 69 | 2.548 | 57 | 6.447 |
| 4. | MESHI_SERVER | QA331_2 | 47 | 5.813 | 51 | 3.760 | 63 | 3.278 | 52 | 6.580 |
| 5. | MESHI_CON_SERVER | QA049_2 | 39 | 5.739 | 43 | 3.889 | 52 | 2.764 | 43 | 7.016 |
| 6. | ProQ3_1_disc | QA095_2 | 51 | 7.264 | 55 | 4.424 | 69 | 2.737 | 57 | 6.502 |
| 7. | ProQ3_1 | QA302_2 | 51 | 7.476 | 55 | 4.492 | 69 | 2.729 | 57 | 6.835 |
| 8. | VoroMQA | QA224_2 | 51 | 8.238 | 55 | 4.724 | 69 | 3.156 | 57 | 8.850 |
| 9. | ProQ2 | QA203_2 | 51 | 7.277 | 55 | 4.821 | 69 | 3.019 | 57 | 7.330 |
| 10. | VoroMQAsr | QA093_2 | 51 | 7.942 | 55 | 4.953 | 69 | 3.331 | 57 | 8.201 |
| 11. | MUFold2 | QA421_2 | 51 | 7.649 | 55 | 5.205 | 69 | 3.349 | 57 | 7.423 |
| 12. | MULTICOM-CLUSTER | QA287_2 | 51 | 8.103 | 55 | 5.357 | 69 | 3.407 | 57 | 7.728 |
| 13. | cSVMQA | QA120_2 | 51 | 8.528 | 55 | 5.384 | 69 | 3.109 | 57 | 8.978 |
| 14. | Pcomb-domain | QA411_2 | 51 | 6.754 | 55 | 5.661 | 69 | 4.690 | 57 | 8.275 |
| 15. | ModFOLD6_rank | QA072_2 | 51 | 7.315 | 55 | 5.689 | 69 | 4.620 | 57 | 8.184 |
| 16. | ProTSAV-Plus | QA226_2 | 51 | 9.329 | 55 | 5.799 | 69 | 4.101 | 57 | 8.854 |
| 17. | Wallner | QA073_2 | 51 | 6.872 | 55 | 5.941 | 69 | 4.925 | 57 | 8.359 |
| 18. | Seok-server | QA250_2 | 51 | 9.297 | 55 | 5.964 | 69 | 3.770 | 57 | 10.442 |
| 19. | QASproGP | QA244_2 | 51 | 7.787 | 55 | 6.195 | 69 | 5.168 | 57 | 8.810 |
| 20. | RSA_SS_CONS | QA270_2 | 51 | 8.635 | 55 | 6.218 | 69 | 4.284 | 57 | 8.329 |
| 21. | Pcons-net | QA432_2 | 46 | 7.056 | 49 | 6.480 | 62 | 5.502 | 50 | 10.170 |
| 22. | ZHOU-SPARKS-X | QA452_2 | 49 | 8.873 | 53 | 6.758 | 66 | 4.941 | 55 | 9.430 |
| 23. | QASproCL | QA267_2 | 51 | 7.719 | 55 | 7.170 | 69 | 6.119 | 57 | 11.094 |
| 24. | ModFOLD6 | QA201_2 | 51 | 8.350 | 55 | 7.196 | 69 | 5.731 | 57 | 9.811 |
| 25. | ModFOLDclust2 | QA214_2 | 51 | 7.188 | 55 | 7.729 | 69 | 7.113 | 57 | 12.250 |

**Figure 3.2:** CASP12 MQA results showing the absolute score difference from the best model. Showing the first 25 methods out of 42, sorted by LDDT score. Image source: `http://www.predictioncenter.org/casp12/qa_diff2best.cgi`

# 4. Machine Learning

## 4.1 Introduction to Machine Learning and its Application in Model Quality Assessment

Machine learning is a computer science discipline that evolved as a sub-field of Artificial Intelligence [48]. The main purpose of Machine learning is to learn and predict the rules and patterns that govern the data. It has many applications such as spam filtering, optical character recognition, detection of malicious behaviour on the Internet, etc.

Machine learning can be broadly divided into three categories:

- **Supervised learning.** In this type of problem there are data where the desired outcome is known. These data are used to train the algorithm to find patterns inside the data that allow it to predict the desired outcome. This algorithm can be later used to predict the outcome for the data where the outcome is unknown. The classic examples of supervised learning are classification and regression.

- **Unsupervised learning.** Here, there are no desired outcomes (or "labels"). Instead, the goal is to find patterns of how the data is structured. An example of unsupervised learning is clustering.

- **Reinforcement learning.** Here, the machine learning algorithm interacts with a dynamic environment trying to achieve a particular goal, for example, car driving.

In Model Quality Assessment (MQA) the most common type of machine learning used is supervised learning, regression in particular. A number of features are extracted from a protein model that describes its physico-chemical properties. These features can be used for training the method to predict model quality, which is often represented as a continuous value in a range from 0 to 1.

As mentioned in chapter 3, protein structure prediction algorithms can be used to predict the structure from the amino acid sequence of a protein whose

experimental structure is unknown. However, these algorithms can be benchmarked on the protein sequences for which the experimental structure has already been determined. The quality of these models can be evaluated by several different measures, like S-score [49; 50], TM-score [51], GDT_TS [52], LDDT [53] and CAD [54] (see Paper IV).

The models for which the experimental structure is known can be used to train MQA programs. The score that these programs are trained to predict is called a target function. For example, ProQ2 [3], ProQ3 [9] and ProQ3D [10] use S-score as a target function. After the method is trained, it can be used to predict the quality of any protein model, even without a known experimental structure.

## 4.2 Training and testing

An important issue in supervised learning is how to train the best method and test its performance in an unbiased way. There are some general rules that should be followed.

The data should be split into the training data and test data. The test data should be independent from training data, i.e. the data that has been used for training cannot be used for testing. Preferably, even similar samples (for example homologous proteins) from the training set should not appear in the test set for a completely unbiased evaluation.

In a comparison of different training methods (for example, SVMs vs deep learning), different architectures (for example, the number of hidden layers in an artificial neural network) or in tuning of hyper parameters of your method (for example, C and gamma parameters of an SVM with a radial kernel), a third data set is needed, which is called a validation set [55]. The training set is used to fit the basic parameters ("weights"), in other words to train the method. The validation set is used to compare the performance of different types of predictors, architectures, or hyper-parameters. Finally, the test set is used to independently evaluate the performance of the final model.

A common split between the training set and a validation set is 70% to 30%. Another option is to perform a cross-validation. In this case, the data set is split into N parts. At each iteration, N-1 parts are used for training and one part for testing. This routine is performed N times until all parts have been used for testing. So, for example, if you perform 10-fold cross-validation (N=10), effectively you train your method on 90% of the data, but in the end you get the performance on 100% of the data. Therefore, it is a good option in case you do not have a large data set.

An important concept in machine learning is overfitting. It occurs when the method's performance is much better on the training set than on the validation

and test sets. The reason is usually that the model has to many parameters that are over-optimised to predict the training samples. These models memorize the exact locations of the training data points, but they fail to make generalisations about the rules that govern the data (see Figure 4.1). To avoid this problem one should always test the model on unseen data (i. e. validation and test sets).

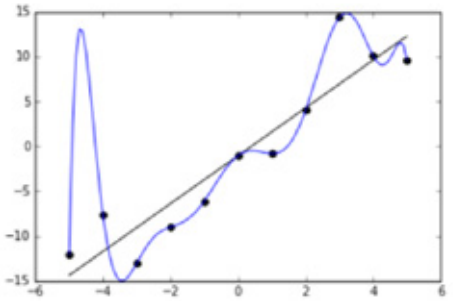



**Figure 4.1:** Overfitting data. The data points are generated with a linear function with some noise. Two fitted models are shown: linear (black) and polynomial (blue). The polynomial model matches the training samples perfectly but is very likely to fail when some new data is presented. Image source: `https://en.wikipedia.org/wiki/Overfitting`

## 4.3 Support Vector Machines

Support Vector Machine (SVM) is a supervised learning approach that can be used for classification or regression. Traditional SVMs that are used for classification divide the space of the training vectors by a hyperplane that maximises the margin between the two classes. In a linear SVM, the hyperplane will always have N-1 dimensions, where N is the number of dimensions of a feature vector [56].

There are two types of margins that are used to separate the training instances: hard margin and soft margin. A hard margin effectively separates the two classes by two parallel hyperplanes and does not allow any training instances to be present between them (see Figure 4.2). The soft margin, on the other hand, allows some of the training instances to be present between the hyperplanes, but tries to minimise the number at the same time as maximising the margin.

More formally, if the training points are linearly separable, we can find two parallel hyperplanes that separate the points and have a maximum distance ("margin") between them (see Figure 4.2). These hyperplanes can be described by equations:

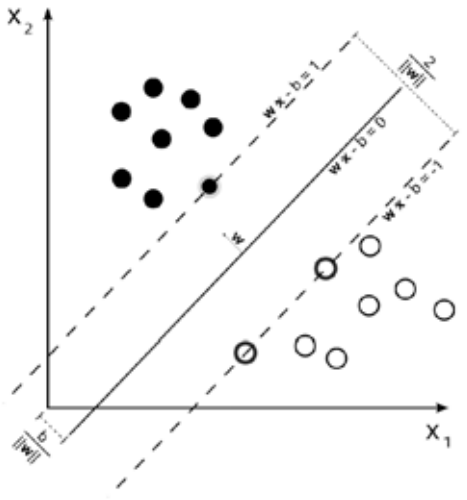



**Figure 4.2:** A maximum margin hyperplane for SVM trained on two classes. Image source: `https://en.wikipedia.org/wiki/Support_vector_machine`

$$\vec{w} \cdot \vec{x} - b = 1 \tag{4.1}$$

and

$$\vec{w} \cdot \vec{x} - b = -1. \tag{4.2}$$

The distance between these two hyperplanes is $\frac{2}{||\vec{w}||}$. So to maximise the distance, we need to minimise $||\vec{w}||$.

In order to ensure that the training points do not fall inside the margin, we need to add constrains subject to the training labels $y_i$:

$$\vec{w} \cdot \vec{x} - b \geq 1, \quad \text{if} \quad y_i = 1 \tag{4.3}$$

or

$$\vec{w} \cdot \vec{x} - b \leq -1, \quad \text{if} \quad y_i = -1. \tag{4.4}$$

So in order to train the linear SVM classifier with hard margin, we need to minimise $||\vec{w}||$ subject to the above mentioned constraints. A similar principle applies to soft margin classifier, but instead of having hard constraints, a penalty function is defined for points falling inside the margin.

Notice that the hyperplanes that separate the two classes only depend on the position of the training points that are closest to it. These training points are called support vectors.

The training points often cannot be separated by a linear function. In this case, the so called "kernel trick" can be applied that transforms the feature

space into a higher dimensional space where the points can be separated [57]. To achieve this, the simple dot product for vectors is replaced by a non-linear kernel function. Popular kernel functions include polynomial, hyperbolic tangent and Gaussian radial basis function.

The original SVM was created for the purposes of classification, but later it was modified to do a regression, too. In this case we are looking for a function that has at most $\varepsilon$ deviation from all of the training instances and is as flat as possible [58]. Analogously to the classification task, first we define a linear function $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$. Finding as flat as possible function means minimising $||\vec{w}||^2$. So the whole problem can be rewritten as:

$$\text{minimise} \quad \frac{1}{2}||\vec{w}||^2, \quad \text{subject to} \quad abs(y_i - \vec{w} \cdot \vec{x} - b) \leq \varepsilon, \tag{4.5}$$

where abs() denotes the absolute value.

The equation describes what is analogous to "hard margin" in a classification case, because all points must fall within $\varepsilon$ of our function. Sometimes this is not feasible, so it is more convenient to allow points outside the $\varepsilon$ margin, but add a penalty function (see [58] for details).

Moreover, just like in a classification case, it is possible to define other kernels that can be used instead of a dot product. Hence, the regression can be performed in a non-linear space.

## 4.4 Artificial Neural Networks and Deep Learning

An Artificial Neural Network (ANN) is a supervised machine learning model that was initially inspired by how neurons function in the brain [59]. In the canonical configuration, the network receives signals through the input layer and outputs it in the output layer (see Figure 4.3A). Between the input and the output layers there might be several hidden layers of neurons. These hidden layers are essential for the network to be able to tackle advanced problems. The network that only transmits the signal forward, but does not feed it back to the neurons of the previous layers, is called a feed-forward neural network. If all of the nodes of one layer are connected to all of the nodes in the next layer, then the network is called a fully connected feed-forward network [60] or, more traditionally, a multi layer perceptron [61].

Each neuron in the network has a weight and when the signal is transferred from one layer to the next, a weighted sum of all signals from the previous layer is calculated (see Figure 4.3B). In addition to that, a non-linear transformation function is applied on each signal. The most common non-linear function used nowadays is the rectified linear unit (ReLU). This function is defined as $f(x) = max(0, x)$, so it simply sets all of the negative values to zero [60].

Training the network means finding the optimal weight vector $\mathbf{w}$ that minimises the loss function $L(\mathbf{w})$ [60]. The loss function measures the fit between the output of the neural network and the true labels of the data. Currently, the most common algorithm for training the network is backward propagation that uses a stochastic gradient descent. At each step the current weight vector is moved towards the direction of steepest descent $d\mathbf{w}$. The amount of how much the vector is changed at each step is determined by the learning rate $\eta$. The learning rate is often gradually decreased during training. This allows to escape the local minima in the beginning and fine tune the final weights in the end of the training (see Figure 4.3C). To reduce the sensitivity of specific choice of the learning rate, the adaptive learning rate methods have been developed, such as RMSprop, Adagrad [62], Adadelta [63] and Adam [64].

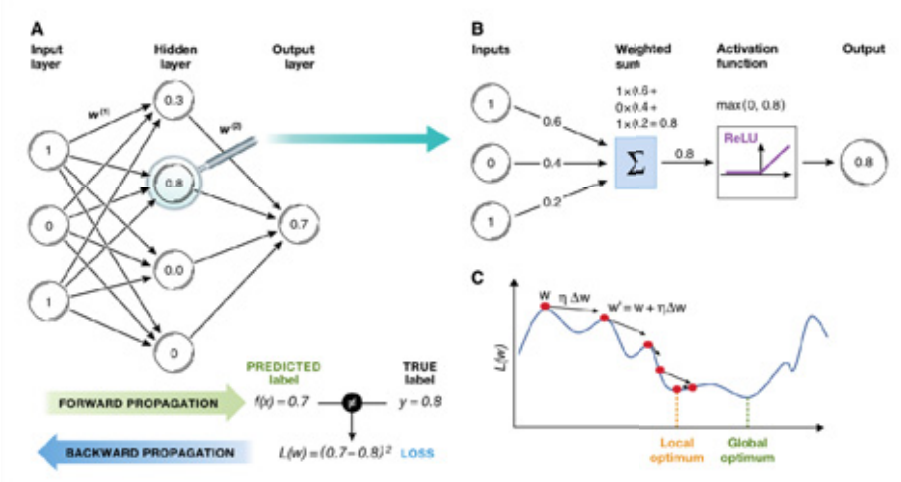



**Figure 4.3:** (A) Artificial Neural Network. (B) Calculating the weighted sum of signals. (C) Training method using a stochastic gradient descent and a decreasing learning rate. Image source: ref [60]

The idea of ANNs has been around for a long time, but in the 1990's it was overshadowed by other supervised machine learning algorithms such as Support Vector Machines and Random Forests. The interest in ANN was revived around 2006 when a group of researchers from the Canadian Institute of Advanced Research introduced a pre-training procedure [65]. This procedure has allowed to initialise the network weights to reasonable values without requiring labelled data. Such approach proved to be extremely useful when the amount of training data is small and worked remarkably well in handwriting and speech recognition. After the interest in ANNs was revived, it turned out that pre-training is only necessary when the training data set is small [66]. The advance in computational technologies, especially Graphical Processing

Units (GPUs), allowed introduction of more hidden layers to networks and together with newly developed training procedures led to revolutionary results in machine learning. Since then, the Artificial Neural Networks were rebranded as Deep Neural Networks and the whole area is nowadays referred as Deep Learning [60].

Although, the fully connected feed-forward network is the recommended default choice for supervised learning [60], in some cases a specialised architecture can provide better results. For example, Convolutional neural networks are well suited for multi-dimensional data, such as two-dimensional images or genomic data [60]. Recurrent neural networks perform very well on sequential data such as text, protein or DNA sequences [60]. Some of the deep learning approaches are also used for unsupervised learning, for example, restricted Boltzmann machines [67], autoencoders and deep belief networks [68]. These methods can learn low-dimensional feature representations from a high dimensional feature space, similar to Principal Component Analysis, but in a non-linear way [60].

An important issue in training Deep Neural Networks is overfitting. Dropout [62] is a common regularisation technique that helps to avoid it. In this method, the activation of part of the neurons is set to zero during each step of the training. This intuitively corresponds to a set of networks whose predictions are averaged. The dropout rate parameter sets the probability that each neuron is going to be deactivated during the particular iteration. A common choice for this parameter is 0.5, so approximately half of the neurons are deactivated at each step. Dropout is often combined with regularising the magnitude or parameter values by L2 and less commonly with L1 norm [60].

# 5. Model Quality Assessment

## 5.1 Scoring Functions

It has long been of interest to be able to identify the native protein structure among a set of many decoys (possible conformations). One of the first methods that tried to address this task were scoring functions. There are two main types of scoring functions: physics-based energy functions and knowledge-based statistical potentials [69].

Physics-based energy functions like AMBER [26], CHARMM [25] and OPLS [70] have been developed to describe the force fields that guide protein folding process. Electrostatics, hydrogen bonds and Van der Waals interactions are some of the forces that are included in these energy functions. The idea to perform molecular dynamics simulations based on the first principles of protein folding is attractive, but unfortunately, computationally very expensive. Therefore, the energy function parameters are optimized on much smaller systems than a protein. Moreover, to reduce the computational complexity, the interactions with water are often ignored in the calculations. Therefore, these functions fail to evaluate one of the most important aspects that make a stable protein conformation - the hydrophobic effect. Furthermore, since the simulation can only produce a single conformation at the time, it does not evaluate the entropic effect. Some of the later methods tried to take these important aspects of protein folding into account and use them together with physics-based energy functions to identify the native protein conformations [71; 72].

Another approach to identify native or native-like states is to use knowledge-based statistical potentials. The statistical potentials are often derived based on equations derived from physics (like the Boltzmann equation) and they are parametrised using the set of available protein structures in the Protein Data Bank [73]. The most common types of statistical properties that were used are residue-residue and atom-atom contact potentials [24; 74; 75].

Sippl has created the popular Prosa II program that calculates the probability that two residues with the separation of N residues in the sequence are lying at distance D in space [74]. This statistical potential is parametrised on structures from PDB and used to evaluate protein models. A very similar pairwise residue distance potential was also applied in threading to evaluate

the sequence to structure alignments in GenTHREADER program by David Jones [24]. GenTHREADER combined the pairwise residue distance based potential with five other scores and trained a neural network to predict the final sequence to structure alignment score.

Not all of the atom-atom and residue-residue potentials are distance-dependent. Miyazawa and Jernigan derived a residue-specific contact potential for all possible pairs of the 20 amino acids, which is not distance dependent [75]. This contact potential, in addition, to an attractive energy term also has a repulsive energy term operative at higher densities to prevent overpacking.

Errat [76] scoring function evaluates atom-atom contacts, but uses a slightly different approach than the other statistical potentials. The atoms are grouped into several groups based on their types, and the fraction of interactions between each group is calculated. The distributions of these contacts are derived from a set of high-resolution native structures and later used to evaluate the protein models. Unlike the other scoring functions, Errat was originally developed to identify errors in the PDB database.

The more conventional Dfire method [77] derived both distance-dependent and residue-specific all-atom statistical potential. Another state of the art method Dope [78] uses distance-dependent all-atom statistical potential, but unlike other methods it does not use the assumption of Boltzman distribution derived from statistical mechanics, but instead it derives a statistical potential entirely on probability theory.

In addition to residue-residue and atom-atom potentials, an important aspect employed by knowledge-based functions is hydrophobicity. The residue type and the degree of its burial are used as a statistical potential both in threading [24; 79] and in model quality assessment [80]. Torsion angle potentials are also employed by some knowledge-based scoring functions [81; 82].

Modern scoring functions often combine several types of knowledge-based statistical potentials. The dDfire method [83], in addition to the distance-dependent residue-residue potential that was used in its predecessor, has also added a statistical potential that describes the orientation dependence of polar atom interactions. GOAP potential [84] combines distance-dependent residue-residue potential that is defined exactly in the same way as in Dfire and a polar atom orientation angle potential, but it is defined differently than in dDfire.

Rosetta [28] is a protein modelling software that has probably the most complicated scoring function. The default scoring function "talaris2013" consists of 16 different terms and the total energy term. Van der Waals, electrostatics, solvation, hydrogen bond, side-chain packing terms and backbone angle potentials are all included into the scoring function. In addition to that, Rosetta has as many as 174 energy terms that can be included into user-defined scoring functions.

Sometimes another type of scoring functions is distinguished—learning-based functions [2]. However, for the purposes of this thesis we will assign learning-based functions to single model quality assessment method type.

## 5.2 Model Quality Assessment Programs

There is no clear distinction between a scoring function (or energy function) and a Model Quality Assessment Program (MQAP). The scoring functions are able to find the protein conformation that is closest to the native state, if all the conformations are generated for the same target and by the same method. However, the scoring functions usually fail to reliably evaluate the protein model quality if the models are created for different targets or by using different methods [4; 69]. This problem is usually addressed by MQA programs.

Being able to provide an accurate non-relative estimate of the global model accuracy is an important feature of the MQAPs. For example, models of medium quality can be useful for predicting the protein function or validating the results of experiments. On the other hand, they cannot be used for inhibitor design or docking [85]. Moreover, MQAPs are useful in protein structure prediction meta-servers when a lot of models are created by different methods and there is a need for a criterion to select the best model. On the other hand, MQAPs are often more complicated than the scoring functions and take a longer time to run.

MQAPs can be broadly divided into three categories: single-model methods that only use the information contained in the protein model, consensus (sometimes called "clustering") methods that evaluate model quality based on its similarity to other models, and so-called "combined" methods that use a combination of single-model and consensus method approach. Most of the consensus methods take a set of models as an input, however, there is a subcategory of consensus methods, called "quasi-single" methods that take only one model as an input and generate a set of models internally to use them for the comparison with the input model.

Consensus methods usually outperform single-model methods in most of the evaluation categories [45–47; 85; 86]. However, the development of single-model methods is still important for several reasons. Firstly, sometimes there is only a single model (or a very small number of models) to evaluate and the conventional consensus methods cannot be used. Quasi-single methods can still be used in this case, but they often take a longer time to run, because it takes time to generate a set of decoys that are used for consensus evaluation [87]. Secondly, the combination of single-model methods and consensus methods very often provides better results than either of the approaches alone [45–47].

Finally, single-model methods often perform better than consensus methods in model selection category of evaluation (see section 6.4).

Assessing the global model quality of a protein model is just one of the questions MQAPs are trying to address. Another important task is to evaluate the local quality—estimate the probability that each residue is modelled correctly. Some of the MQAPs address only the global evaluation problem [2; 4; 7; 8], while others perform both local and global assessment [3; 5; 6; 41; 69; 87–89].

In addition to scoring functions and MQAPs there could be a third type of quality assessors distinguished—the ones that perform stereochemistry checks. Some of the popular stereochemistry checking programs are PROCHECK [90], WHATCHECK [91] and MolProbity [92]. These methods perform a basic check whether the model stereochemistry is correct. However, the fact, that model has a correct stereochemistry, does not mean that its backbone conformation is close to the native state and vice versa. Nevertheless, some of the MQAPs perform stereochemistry checks as one of the steps to evaluate the protein model.

## 5.3 Single-model methods

Single-model quality assessment methods are the ones that use only the information contained in the protein model to estimate its quality. Many of these methods use a machine learning approach [2–10]. The relevant training features are extracted from the protein structure and sequence and fed into a supervised machine learning method to predict the target function that represents the quality. These methods differ between themselves in the machine learning method that is used, features that are extracted, training data sets and target functions.

ProQ [2] program was one of the first machine-learning-based single-model quality assessment methods. It uses a neural-network machine learning method and has two target functions: LG-score [49] and MaxSub [93]. The training features of ProQ program are atom-atom contacts, residue-residue contacts, solvent accessibility surfaces, secondary structure and fatness (the overall shape of the protein).

The atom-atom and residue-residue contacts in ProQ are defined in a similar way as in the Errat [76] scoring function. The atoms and residues are divided into certain groups based on their properties and the pairwise contacts between each group are counted. In order to avoid the dependency on protein size, the fraction of a total number of contacts is taken as opposed to the absolute counts.

Two very important features in ProQ are solvent accessibilities and secondary structure. The secondary structure feature is calculated as an agreement between the secondary structure observed in the model (evaluated by STRIDE [94]) and the one predicted from the sequence (using PSIPRED [95]). The original version of ProQ used only the solvent accessibility surfaces calculated from the model (using NACCESS [96]), however, later versions of ProQ [3; 9; 10], also calculated the agreement between solvent accessibilities observed in the model and the ones predicted from the sequence (using ACCPRO [97]). The secondary structure and solvent accessibility predictors are rather accurate, so it is likely that for good structural models these agreements will be high.

Unlike the scoring functions, MQAPs are often trained on models of varying quality rather than the native structures from PDB. ProQ was trained on LiveBench-2 data set [98].

The original ProQ program was developed to predict only the global quality of a protein model, however, the later versions are also able to predict the local (per-residue) quality [3; 9; 10; 88]. For further description of other ProQ methods see chapter 6 and the corresponding manuscripts.

Another global predictor ModelEvaluator [4] is based on Support Vector Machines (SVM) with a radial kernel. The training features include secondary structure and solvent accessibility agreements similar to ProQ. The features that differ from ProQ are predicted and observed residue contact agreement, as well as, predicted and observed contact order agreement. The training is done on models generated by three different methods (Robetta [99], Sparks3 [100] and FOLD-pro [101]) spanning 64 CASP6 targets. The target function is GDT_TS [52].

SMOQ [5] uses a similar approach to ModelEvaluator, but it can predict local quality in addition to global quality. The authors have tested three sets of features: basic, profile and SOV (Segment Overlap Measure for secondary structure), but the final version of the program included only the basic set of features. Basic features include amino acid sequence, secondary structures, solvent accessibility, and residue-residue contacts (predicted using NNcon [102]). The machine learning method used was SVM with a radial kernel and the target function was a distance between the model and the native structure. The global scores were derived using a formula similar to S-score that is also used in ProQ2. The method was trained on 85 CASP8 targets.

Statistical-based potentials (scoring functions) are sometimes included into MQAPs as training features. The Qprob method [7] utilises three scoring functions: RF_CB_SRS_OD [103], RWplus [104] and DFIRE2 [83]. These scoring functions are protein length dependent, so they are normalised by a simple linear regression before training. Other training features used are several

different descriptions of secondary structure and surface area accessibilities, as well as Euclidean compact score that describes compactness of the model. Qprob predicts only the global quality and the target function is GDT_TS. The method uses the Expectation Minimisation (EM) algorithm to select the feature weights that are optimised to decrease average GDT_TS loss. The algorithm was optimised on 99 CASP9 targets.

Lately, deep learning has been a popular approach in MQA. Wang_deep methods [6] use a combination of SVM and Stacked Denoising Autoencoders. The training features were amino acid sequence, profile, secondary structure and solvent accessibility agreements, predicted and observed residue-residue contacts (using NNcon [102]), and segment overlap measurement (SOV) for secondary structure. The features were generated using a 15-residue sliding window. The target function was per-residue distance error and the global scores were derived in a similar way as in ProQ2. The training set was CASP8 and CASP9 data sets (1 model per method).

DeepQA is another recent deep learning method. It is a meta-predictor that uses scores from ModelEvaluator [4], Dope [78], RWplus [104], Qprob [7], GOAP [84], RF_CB_SRS_OD [103], OPUS [105], ProQ2 [3] and DFIRE2 [83] as training features. In addition to that, it also includes several descriptions of secondary structure, solvent accessibility surfaces and Euclidean compact score. A deep belief network is trained with two hidden layers of Restricted Boltzmann Machines (RBMs) and one layer of logistic regression node is added at the top to output a value between 0 and 1. The training data set consists of models from CASP8, CASP9, and CASP10, 3DRobot decoys [106] and 3113 native protein structure from PISCES database [107].

Even though a machine learning approach seems to be the most popular among MQAPs, some methods do not use it. For example, the Qmean [69] method uses a simple linear combination of its scoring terms and the coefficients are optimised using regression. The scoring terms included into Qmean are distance-dependent residue-residue potential, solvation potential, torsion angle potential as well as secondary structure and solvent accessibility agreements. The data set used for optimisation was CASP6 with low quality models (GDT_TS < 0.2) removed.

Another method that does not use a machine learning approach is VoroMQA. The method is not yet published, but according to the CASP12 abstract [108] it derives a knowledge-based statistical potential based on atom-atom and residue-residue contact areas. The implementation details will be available in the upcoming paper.

The authors of all of the mentioned methods have benchmarked them on different data sets, but the most reliable and objective evaluations are the ones performed by CASP organisers [45–47; 85; 86]. The ProQ methods since their

introduction were among the best single-model quality assessment methods.

# 6. Summary of papers

## 6.1 ProQ2: estimation of model accuracy implemented in Rosetta (Paper I)

ProQ2 is a machine learning based model quality assessment method. It takes as an input several features calculated from protein model and predicts the target score, which is called S-score [49; 50]. The main input features are atom-atom contacts, residue-residue contacts, conservation, solvent accessibility surfaces and secondary structure. ProQ2 can predict quality for each residue separately by considering input features from all surrounding residues within a fixed size window. The global model quality in ProQ2 is calculated as a sum of local scores divided by the protein length. The machine learning method used in ProQ2 is a Support Vector Machine (SVM) with a linear kernel.

The original version of ProQ2 was implemented as a stand-alone software and a web-server. Here, we implemented the method inside Rosetta [28] modelling software. This allows users to use ProQ2 as a scoring function for protein conformational sampling inside Rosetta. Moreover, such integration allows to easily use ProQ2 together with other Rosetta tools. For example, one benefit was demonstrated by using ProQ2 together with Rosetta side-chain repacking protocol. Different protein structure prediction methods have different ways of modelling the side chains, so adding a side-chain repacking step eliminates these differences and allows to evaluate protein model quality based on the correctness of the backbone. Finally, ProQ2 implementation in Rosetta has removed several software dependencies, including Naccess, ProQres, Stride and SVM-light.

## 6.2 ProQ3: Improved model quality assessments using Rosetta energy terms (Paper II)

Compared to ProQ2, ProQ3 introduces new training features calculated from Rosetta energy functions. There are two types of Rosetta [28] energy functions: full-atom and centroid. A full-atom function takes into account all atoms

in the protein, while a centroid function only considers the protein backbone. We used the "talaris2013" full-atom energy function that has 16 energy terms and the total energy term. We also used four energy terms from the "cen_std" energy function and five other independent centroid energy terms. All of these energy terms were scaled between zero and one using sigmoidal transformation and their values were averaged using varying window sizes. We trained an SVM with a linear kernel on all of these energy terms averaged over different window sizes. We call the methods that use full atom and centroid energy terms ProQRosFA and ProQRosCen respectively and we call the method that uses all energy terms together with all ProQ2 training features—ProQ3.

We showed that ProQ3 consistently outperforms ProQ2 on CASP11 and CAMEO data sets. We have evaluated both local and global correlations on the whole data set as well as average per target and average per-model correlations. Moreover, we benchmarked the ability of methods to select the best model as measured by the average first ranked GDT_TS [109] scores of the selected models. This was the only evaluation category where ProQ3 did not perform significantly better than ProQ2 in our benchmark.

ProQ3 participated in CASP12 experiment and was one of the best single-model quality assessment methods. Interestingly, in CASP12 ProQ3 significantly outperformed ProQ2 even in model selection evaluation category.

## 6.3 ProQ3D: Improved model quality assessments using Deep Learning (Paper III)

ProQ3D uses the exact same input features as ProQ3, but the SVM method is replaced by a deep neural network. We have used a fully connected feed-forward neural network with two hidden layers, one with 600 and the other one with 200 neurons. Increasing the number of layers and neurons did not improve the results significantly. We have used the Adadelta adaptive learning rate and $10^{-11}$ penalty for the $L^2$ regularisation. The dropout rate was set to 0.5.

In our benchmark ProQ3D consistently outperformed ProQ3 and other single-model methods both in local and global correlations. The global correlation was 0.81 for ProQ2, 0.85 for ProQ3 and 0.90 for ProQ3D. Local correlations were 0.69/0.73/0.77 for ProQ2/ProQ3/ProQ3D respectively. We have also performed Receiver Operating Curve (ROC) analysis where ProQ3D was also superior to all other single-model methods. There was no significant improvement in model selection. The mentioned results are on CASP11 data set, but the results on CAMEO data set were similar.

## 6.4 Improved protein model quality prediction by changing the target function (Paper IV)

Here, we analyse five different scoring methods that are used to evaluate protein model quality when the native structure is determined. Three of these methods (S-score [49; 50], TM-score [110] and GDT_TS [109]) are superposition-based while two other methods (LDDT [53] and CAD [54]) are contact-based measures. We show that the correlation between the same type of methods is usually higher than the correlation between different types of methods.

Next, we train ProQ3D on all of the above measures except for GDT_TS, which does not have local evaluations that are necessary for training. We evaluate the retrained ProQ3D performance by calculating Pearson correlations with all five above mentioned methods. The correlations are calculated in three different ways: local whole data set correlations, global whole data set correlations and global per target correlations. The per target correlations are calculated for each target separately and the average is taken. We have also evaluated scoring measures in model selection category.

The correlations are usually highest when the same scoring method is used both in training and testing. Interestingly, contact-based measures achieve a higher correlation than the superposition based measures. For example, on CAMEO data set, the local correlations are 0.66 for S-score, 0.69 for TM-score, 0.74 for CAD and 0.79 for LDDT.

The biggest difference between the two types of measures is in global per target correlations where training on contact-based measures sometimes yields a higher correlation with superposition measures than training on the superposition measures themselves. For example, on CAMEO data set training on LDDT yields 0.60 correlation with S-score, while training on S-score yields only 0.53 correlation with itself.

Per target correlations are related to model selection problem, because they show how well the methods are able to distinguish good and bad models within a target. Since contact-based measures perform well in per target correlations, it is quite unsurprising that they also perform well in model-selection. Here, CAD demonstrates a superior performance. Training on it gives the best results in model selection when evaluated by any scoring method. For example, on CAMEO data set, training on CAD selects models whose average GDT_TS loss is 0.036, while training on S-score/TM-score/LDDT selects models with GDT_loss equal to 0.041/0.042/0.037, repectively (lower numbers are better).

# Sammanfattning på Svenska

Det finns många olika metoder för att skapa modeller av proteiner, dessa modeller kan vara av olika kvalitet. Att avgöra vilken av flera olika modeller av ett protein som är bäst är inte enkelt. Modelkvalitéstutvärderingsprogram (MQAPs) är utvecklade för att identifiera den bästa av de olika proteinmodellerna. Dessa program kan uppskatta hur korrekt en modell är innan proteinets struktur har bestämts. Problemet att uppskatta kvalitén kan delas upp i två delar, kvalitén för hela proteinet (global) samt den lokala kvalitén för varje rest i proteinet. ProQ2 är en av de mest framgångsrika MQAPs och kan användas för att uppskatta både den lokala och globala kvalitén. ProQ2 använder en maskininlärningsmetod.

I denna avhandling så presenterar jag mitt bidrag till MQAPs, inklusive utveckling av de senaste versionerna av ProQ. Först beskriver jag integreringen av ProQ2 i mjukvaruprogrammet Rosetta, vilket gör det möjligt att använda ProQ2 för sampling i Rosetta. Därefter presenterar jag utvecklingen av ProQ3 och ProQ3D, vilka är förbättringar till ProQ2. I ProQ3 så inkluderar jag beskrivningar av proteinet från Rosetta och i ProQ3D så har jag inkluderat ett ny djup maskininlärningsmetod. ProQ3 var en av de bästa MQAPs i CASP12, ett experiment som utvärderar olika aspekter av proteinstrukturförutsägningar. I den avslutande studien så har jag studerat olika metoder för att beskriva kvalité hos en proteinmodell och visar att MQAPs blir bättre om man använder kvalitetsmått som är baserade på kontakter än traditionella kvalitetsmått baserade på superposition.

# Acknowledgements

To begin with, this thesis would not have been possible if I did not have so many great supervisors. First and foremost, I would like to acknowledge my main supervisor Arne Elofsson. Thank you for your guidance and patience. Even during difficult periods you never stopped believing in me. Thanks for giving me great ideas and at the same time allowing me to explore my own ideas and mature as a scientist.

Next, I would like to thank Björn Wallner who has been my unofficial supervisor. Even though, you did not have any formal obligation, you always assisted me with every technical problem that I had. Your friendliness and helpfulness made this PhD a whole lot easier to me.

Furthermore, I would like to thank Nanjiang Shu who has been my co-supervisor and a great friend. Thank you for all the tips and the guidance in the beginning of my PhD that helped me to survive the difficult start. Also, thanks for setting up the ProQ3 server, this has saved me a lot of work.

All of the PhD would have been much more difficult without my colleagues who have been both friends and advisers. Firstly, I would like to thank Per Warholm who helped me with all the Swedish documents and gave me so much good advice about living in Sweden in general. Secondly, I would like to thank Kostas Tsirigos who has run so many of my errands at Stockholm University. Also, thanks for always being a fun person to talk to and for inspiring me to train at the gym. Thirdly, I would like to thank David Menéndez Hurtado who has taught me so many things about deep learning and been an excellent collaborator. I would also like to thank John Lamb for being patient with my biking skills and together with David keeping me a nice company on the trip to Kansas. I would like to thank Mirco for his company on the trips to two CASP meetings and for interesting scientific discussions after the group meetings. I would like to thank Oxana for her advice on using Git and for keeping me company when studying for the oral exam. I would like to thank Marco Salvatore and Sudha Govindarajan for all the tips before travelling to their countries, Italy and India. I would like to thank Petras Kundrotas for lunches together and wise advice regarding the career. I would like to thank all Lithuanians at Karolinska, especially Andrius and Monika for all the lunches. Finally, I would like to thank all the rest of the current and ex-members of Arne Elofsson's group: Walter Basile, Christoph Peters, Marcin Skwark, Sikander

Hayat, Rauan Sagit, Minttu Virkki, Sara Light and others. And, of course, all the other people at Scilifelab for creating a beautiful work environment. A special thanks goes to Christian Wennberg and Ozge Yoluk for helping me to keep the board game club alive!

I would like to greatly acknowledge Swedish Research Council and Stockholm University for providing me this opportunity to do a PhD.

Finally, I would like to thank all my friends and family who have supported me during the entire time. Especially to my mom for her hours of waiting until she is able to talk to me on Skype.

# References


[1] C.B. ANFINSEN. **Principles that govern the folding of protein chains.** *Science*, **181**(4096):223–230, 1973. 11, 14, 18

[2] B. WALLNER AND A. ELOFSSON. **Can correct protein models be identified?** *Protein Sci*, **12**(5):1073–1086, 2003. 11, 23, 37, 38

[3] A. RAY, E. LINDAHL, AND B. WALLNER. **Improved model quality assessment using ProQ2.** *BMC Bioinformatics*, **13**:224, 2012. 11, 24, 28, 38, 39, 40

[4] Z. WANG, A.N. TEGGE, AND J. CHENG. **Evaluating the absolute quality of a single protein model using structural features and support vector machines.** *Proteins*, **75**(3):638–647, 2009. 37, 38, 39, 40

[5] R. CAO, Z. WANG, Y. WANG, AND J. CHENG. **SMOQ: a tool for predicting the absolute residue-specific quality of a single protein model with support vector machines.** *BMC Bioinformatics*, **15**:120, 2014. 38, 39

[6] T. LIU, Y. WANG, J. EICKHOLT, AND Z. WANG. **Benchmarking Deep Networks for Predicting Residue-Specific Quality of Individual Protein Models in CASP11.** *Sci Rep*, **6**:19301, 2016. 38, 40

[7] R. CAO AND J. CHENG. **Protein single-model quality assessment by feature-based probability density functions.** *Sci Rep*, **6**:23990, 2016. 38, 39, 40

[8] RENZHI CAO, DEBSWAPNA BHATTACHARYA, JIE HOU, AND JIANLIN CHENG. **DeepQA: Improving the estimation of single protein model quality with deep belief networks**. *arXiv preprint*, **abs/1607.04379**, 2016. 38

[9] K. UZIELA, N. SHU, B. WALLNER, AND A. ELOFSSON. **ProQ3: Improved model quality assessments using Rosetta energy terms.** *Sci Rep*, **6**:33509, 2016. 24, 28, 39

[10] K. UZIELA, D MENÉNDEZ HURTADO, N. SHU, B. WALLNER, AND A. ELOFSSON. **ProQ3D: Improved model quality assessments using Deep Learning.** *Bioinformatics*, **in press**, 2017. 11, 24, 28, 38, 39

[11] K.A. DILL AND J.L. MACCALLUM. **The protein-folding problem, 50 years on.** *Science*, **338**(6110):1042–1046, 2012. 14

[12] C.S. SEVIER AND C.A. KAISER. **Formation and transfer of disulphide bonds in living cells.** *Nat Rev Mol Cell Biol*, **3**(11):836–847, 2002. 15

[13] CYRUS LEVINTHAL. **Are there pathways for protein folding**. *J. Chim. phys*, **65**(1):44–45, 1968. 15

[14] CYRUS LEVINTHAL. **How to fold graciously**. *Mossbauer spectroscopy in biological systems*, **67**:22–24, 1969. 15

[15] S. GOODWIN, J.D. MCPHERSON, AND W.R. MCCOMBIE. **Coming of age: ten years of next-generation sequencing technologies.** *Nat Rev Genet*, **17**(6):333–351, 2016. 17

[16] CARL IVAR BRANDEN AND JOHN TOOZE. *Introduction to protein structure*. Garland Science, second edition, 1999. 17

[17] MARKETA ZVELEBIL AND JEREMY BAUM. *Understanding bioinformatics*. Garland Science, 2007. 18, 19, 20

[18] S.F. ALTSCHUL, T.L. MADDEN, A.A. SCHAFFER, J. ZHANG, Z. ZHANG, W. MILLER, AND D.J. LIPMAN. **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res*, **25**(17):3389–3402, 1997. 18

[19] R.D. FINN, J. CLEMENTS, W. ARNDT, B.L. MILLER, T.J. WHEELER, F. SCHREIBER, A. BATEMAN, AND S.R. EDDY. **HMMER web server: 2015 update.** *Nucleic Acids Res*, **43**(W1):W30–8, 2015. 18, 20

[20] M. REMMERT, A. BIEGERT, A. HAUSER, AND J. SODING. **HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment.** *Nat Methods*, **9**(2):173–175, 2011. 18, 19, 20

[21] S.F. ALTSCHUL, W. GISH, W. MILLER, E.W. MYERS, AND D.J. LIPMAN. **Basic local alignment search tool.** *J Mol Biol*, **215**(3):403–410, 1990. 19

[22] W.R. PEARSON AND D.J. LIPMAN. **Improved tools for biological sequence comparison.** *Proc Natl Acad Sci U S A*, **85**(8):2444–2448, 1988. 19

[23] S.B. NEEDLEMAN AND C.D. WUNSCH. **A general method applicable to the search for similarities in the amino acid sequence of two proteins.** *J Mol Biol*, **48**(3):443–453, 1970. 19

[24] D.T. JONES. **GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences.** *J Mol Biol*, **287**(4):797–815, 1999. 19, 35, 36

[25] BERNARD R BROOKS, ROBERT E BRUCCOLERI, BARRY D OLAFSON, DAVID J STATES, S SWAMINATHAN, AND MARTIN KARPLUS. **CHARMM: a program for macromolecular energy, minimization, and dynamics calculations**. *J Comput Chem*, **4**(2):187–217, 1983. 20, 35

[26] SCOTT J WEINER, PETER A KOLLMAN, DAVID A CASE, U CHANDRA SINGH, CATERINA GHIO, GULIANO ALAGONA, SALVATORE PROFETA, AND PAUL WEINER. **A new force field for molecular mechanical simulation of nucleic acids and proteins**. *J Am Chem Soc*, **106**(3):765–784, 1984. 20, 35

[27] MARCIN J SKWARK. *Ensemble methods for protein structure prediction*. PhD thesis, Department of Biochemistry and Biophysics, Stockholm University, 2013. 21

[28] A. LEAVER-FAY, M. TYKA, S.M. LEWIS, O.F. LANGE, J. THOMPSON, R. JACAK, K. KAUFMAN, P.D. RENFREW, C.A. SMITH, W. SHEFFLER, I.W. DAVIS, S. COOPER, A. TREUILLE, D.J. MANDELL, F. RICHTER, Y.E. BAN, S.J. FLEISHMAN, J.E. CORN, D.E. KIM, S. LYSKOV, M. BERRONDO, S. MENTZER, Z. POPOVIC, J.J. HAVRANEK, J. KARANICOLAS, R. DAS, J. MEILER, T. KORTEMME, J.J. GRAY, B. KUHLMAN, D. BAKER, AND P. BRADLEY. **ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules.** *Meth Enzymol*, **487**:545–574, 2011. 21, 36, 43

[29] J. YANG, R. YAN, A. ROY, D. XU, J. POISSON, AND Y. ZHANG. **The I-TASSER Suite: protein structure and function prediction.** *Nat Methods*, **12**(1):7–8, 2015. 21

[30] D.T. JONES. **Predicting novel protein folds by using FRAGFOLD.** *Proteins*, **Suppl 5**:127–132, 2001. 21

[31] D.T. JONES, D.W. BUCHAN, D. COZZETTO, AND M. PONTIL. **PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments.** *Bioinformatics*, **28**(2):184–190, 2012. 21

[32] M. WEIGT, R.A. WHITE, H. SZURMANT, J.A. HOCH, AND T. HWA. **Identification of direct residue contacts in protein-protein interaction by message passing.** *Proc Natl Acad Sci U S A*, **106**(1):67–72, 2009.

[33] M.J. SKWARK, A. ABDEL-REHIM, AND A. ELOFSSON. **PconsC: combination of direct information methods and alignments improves contact prediction.** *Bioinformatics*, **29**(14):1815–1816, 2013.

[34] M.J. SKWARK, D. RAIMONDI, M. MICHEL, AND A. ELOFSSON. **Improved contact predictions using the recognition of protein like contact patterns.** *PLoS Comput Biol*, **10**(11):e1003889, 2014. 21

[35] M. MICHEL, S. HAYAT, M.J. SKWARK, C. SANDER, D.S. MARKS, AND A. ELOFSSON. **PconsFold: improved contact predictions improve protein models.** *Bioinformatics*, **30**(17):i482–8, 2014. 21

[36] J. MOULT, J.T. PEDERSEN, R. JUDSON, AND K. FIDELIS. **A large-scale experiment to assess protein structure prediction methods.** *Proteins*, **23**(3):ii–v, 1995. 21

[37] D. FISCHER, A. ELOFSSON, L. RYCHLEWSKI, F. PAZOS, A. VALENCIA, B. ROST, A.R. ORTIZ, AND R.L. DUNBRACK, JR. **CAFASP2: the second critical assessment of fully automated structure prediction methods.** *Proteins*, **Suppl 5**:171–183, 2001. 21, 23

[38] **12th Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction**. `http://www.predictioncenter.org/casp12/index.cgi`. Accessed: 2016-12-01. 21, 22

[39] J. JANIN. **Assessing predictions of protein-protein interaction: the CAPRI experiment.** *Protein Sci*, **14**(2):278–283, 2005. 22

[40] J. HAAS, S. ROTH, K. ARNOLD, F. KIEFER, T. SCHMIDT, L. BORDOLI, AND T. SCHWEDE. **The Protein Model Portal–a comprehensive resource for protein structure and model information.** *Database (Oxford)*, **2013**:bat031, 2013. 22

[41] J. LUNDSTROM, L. RYCHLEWSKI, J. BUJNICKI, AND A. ELOFSSON. **Pcons: a neural-network-based consensus predictor that improves fold recognition.** *Protein Sci*, **10**(11):2354–2362, 2001. 23, 38

[42] B. WALLNER. *Protein Structure Prediction: Model Building and Quality Assessment*. PhD thesis, Department of Biochemistry and Biophysics, Stockholm University, 2005. 23

[43] K. GINALSKI, A. ELOFSSON, D. FISCHER, AND L. RYCHLEWSKI. **3D-Jury: a simple approach to improve protein structure predictions.** *Bioinformatics*, **19**(8):1015–1018, 2003. 23

[44] P. LARSSON, M.J. SKWARK, B. WALLNER, AND A. ELOFSSON. **Assessment of global and local model quality in CASP8 using Pcons and ProQ.** *Proteins*, **77 Suppl 9**:167–172, 2009. 23

[45] A. KRYSHTAFOVYCH, A. BARBATO, B. MONASTYRSKYY, K. FIDELIS, T. SCHWEDE, AND A. TRAMONTANO. **Methods of model accuracy estimation can help selecting the best models from decoy sets: Assessment of model accuracy estimations in CASP11.** *Proteins*, **84 Suppl 1**:349–369, 2016. 24, 37, 40

[46] A. KRYSHTAFOVYCH, A. BARBATO, K. FIDELIS, B. MONASTYRSKYY, T. SCHWEDE, AND A. TRAMONTANO. **Assessment of the assessment: evaluation of the model quality estimates in CASP10.** *Proteins*, **82 Suppl 2**:112–126, 2014.

[47] A. KRYSHTAFOVYCH, K. FIDELIS, AND A. TRAMONTANO. **Evaluation of model quality predictions in CASP9.** *Proteins*, **79 Suppl 10**:91–106, 2011. 24, 37, 40

[48] WIKIPEDIA. **Machine Learning—Wikipedia, The Free Encyclopedia**. `http://en.wikipedia.org/wiki/Machine_learning`, 2016. [Online; accessed 2016-12-20]. 27

[49] S. CRISTOBAL, A. ZEMLA, D. FISCHER, L. RYCHLEWSKI, AND A. ELOFSSON. **A study of quality measures for protein threading models.** *BMC Bioinformatics*, **2**:5, 2001. 28, 38, 43, 45

[50] M. LEVITT AND M. GERSTEIN. **A unified statistical framework for sequence comparison and structure comparison.** *Proc Natl Acad Sci U S A*, **95**(11):5913–5920, 1998. 28, 43, 45

[51] Y. ZHANG AND J. SKOLNICK. **Scoring function for automated assessment of protein structure template quality.** *Proteins*, **57**(4):702–710, 2004. 28

[52] A. ZEMLA, VENCLOVAS, J. MOULT, AND K. FIDELIS. **Processing and evaluation of predictions in CASP4.** *Proteins*, **Suppl 5**:13–21, 2001. 28, 39

[53] V. MARIANI, M. BIASINI, A. BARBATO, AND T. SCHWEDE. **lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests.** *Bioinformatics*, **29**(21):2722–2728, 2013. 28, 45

[54] K. OLECHNOVIC, E. KULBERKYTE, AND C. VENCLOVAS. **CAD-score: a new contact area difference-based function for evaluation of protein structural models.** *Proteins*, **81**(1):149–162, 2013. 28, 45

[55] WIKIPEDIA. **Validation set—Wikipedia, The Free Encyclopedia**. `http://en.wikipedia.org/wiki/Test_set#Validation_set`, 2016. [Online; accessed 2016-12-20]. 28

[56] WIKIPEDIA. **Support vector machine—Wikipedia, The Free Encyclopedia**. `http://en.wikipedia.org/wiki/Support_vector_machine`, 2016. [Online; accessed 2016-12-20]. 29

[57] A AIZERMAN, EMMANUEL M BRAVERMAN, AND LI ROZONER. **Theoretical foundations of the potential function method in pattern recognition learning**. *Automat Rem Contr+*, **25**:821–837, 1964. 31

[58] ALEX J SMOLA AND BERNHARD SCHÖLKOPF. **A tutorial on support vector regression**. *Stat Comput*, **14**(3):199–222, 2004. 31

[59] W.S. MCCULLOCH AND W. PITTS. **A logical calculus of the ideas immanent in nervous activity. 1943.** *Bull Math Biol*, **52**(1-2):99–115; discussion 73–97, 1990. 31

[60] C. ANGERMUELLER, T. PARNAMAA, L. PARTS, AND O. STEGLE. **Deep learning for computational biology.** *Mol Syst Biol*, **12**(7):878, 2016. 31, 32, 33

[61] FRANK ROSENBLATT. **Principles of neurodynamics. perceptrons and the theory of brain mechanisms**. Technical report, DTIC Document, 1961. 31

[62] NITISH SRIVASTAVA, GEOFFREY E HINTON, ALEX KRIZHEVSKY, ILYA SUTSKEVER, AND RUSLAN SALAKHUTDINOV. **Dropout: a simple way to prevent neural networks from overfitting.** *J Mach Learn Res*, **15**(1):1929–1958, 2014. 32, 33

[63] MATTHEW D. ZEILER. **ADADELTA: An Adaptive Learning Rate Method**. *CoRR*, **abs/1212.5701**, 2012. 32

[64] DIEDERIK KINGMA AND JIMMY BA. **Adam: A method for stochastic optimization**. *arXiv preprint arXiv:1412.6980*, 2014. 32

[65] G.E. HINTON, S. OSINDERO, AND Y.W. TEH. **A fast learning algorithm for deep belief nets.** *Neural Comput*, **18**(7):1527–1554, 2006. 32

[66] Y. LeCun, Y. Bengio, and G. Hinton. **Deep learning.** *Nature*, **521**(7553):436–444, 2015. 32

[67] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. **Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion**. *J Mach Learn Res*, **11**:3371–3408, 2010. 33

[68] G.E. Hinton and R.R. Salakhutdinov. **Reducing the dimensionality of data with neural networks.** *Science*, **313**(5786):504–507, 2006. 33

[69] P. Benkert, S.C. Tosatto, and D. Schomburg. **QMEAN: A comprehensive scoring function for model quality assessment.** *Proteins*, **71**(1):261–277, 2008. 35, 37, 38, 40

[70] William L Jorgensen, David S Maxwell, and Julian Tirado-Rives. **Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids**. *J Am Chem Soc*, **118**(45):11225–11236, 1996. 35

[71] B.N. Dominy and C.L. Brooks. **Identifying native-like protein structures using physics-based potentials.** *J Comput Chem*, **23**(1):147–160, 2002. 35

[72] A.K. Felts, E. Gallicchio, A. Wallqvist, and R.M. Levy. **Distinguishing native conformations of proteins from decoys with an effective free energy estimator based on the OPLS all-atom force field and the Surface Generalized Born solvent model.** *Proteins*, **48**(2):404–422, 2002. 35

[73] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. **The Protein Data Bank.** *Nucleic Acids Res*, **28**(1):235–242, 2000. 35

[74] M.J. Sippl. **Boltzmann's principle, knowledge-based mean fields and protein folding. An approach to the computational determination of protein structures.** *J Comput Aided Mol Des*, **7**(4):473–501, 1993. 35

[75] S. Miyazawa and R.L. Jernigan. **Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading.** *J Mol Biol*, **256**(3):623–644, 1996. 35, 36

[76] C. Colovos and T.O. Yeates. **Verification of protein structures: patterns of nonbonded atomic interactions.** *Protein Sci*, **2**(9):1511–1519, 1993. 36, 38

[77] H. Zhou and Y. Zhou. **Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction.** *Protein Sci*, **11**(11):2714–2726, 2002. 36

[78] M.Y. Shen and A. Sali. **Statistical potential for assessment and prediction of protein structures.** *Protein Sci*, **15**(11):2507–2524, 2006. 36, 40

[79] J.U. Bowie, R. Luthy, and D. Eisenberg. **A method to identify protein sequences that fold into a known three-dimensional structure.** *Science*, **253**(5016):164–170, 1991. 36

[80] D. Eisenberg, R. Luthy, and J.U. Bowie. **VERIFY3D: assessment of protein models with three-dimensional profiles.** *Meth Enzymol*, **277**:396–404, 1997. 36

[81] J.P. Kocher, M.J. Rooman, and S.J. Wodak. **Factors influencing the ability of knowledge-based potentials to identify native sequence-structure matches.** *J Mol Biol*, **235**(5):1598–1613, 1994. 36

[82] D. Gilis and M. Rooman. **Predicting protein stability changes upon mutation using database-derived potentials: solvent accessibility determines the importance of local versus non-local interactions along the sequence.** *J Mol Biol*, **272**(2):276–290, 1997. 36

[83] Y. YANG AND Y. ZHOU. **Specific interactions for ab initio folding of protein terminal regions with secondary structures.** *Proteins*, **72**(2):793–803, 2008. 36, 39, 40

[84] H. ZHOU AND J. SKOLNICK. **GOAP: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction.** *Biophys J*, **101**(8):2043–2052, 2011. 36, 40

[85] D. COZZETTO, A. KRYSHTAFOVYCH, M. CERIANI, AND A. TRAMONTANO. **Assessment of predictions in the model quality assessment category.** *Proteins*, **69 Suppl 8**:175–183, 2007. 37, 40

[86] D. COZZETTO, A. KRYSHTAFOVYCH, AND A. TRAMONTANO. **Evaluation of CASP8 model quality predictions.** *Proteins*, **77 Suppl 9**:157–166, 2009. 37, 40

[87] DANIEL BARRY ROCHE, MARIA TERESA BUENAVISTA, AND LIAM JAMES MCGUFFIN. **Assessing the quality of modelled 3D protein structures using the ModFOLD server.** *Methods in molecular biology (Clifton, N.J.)*, **1137**:83–103, 2014. 37, 38

[88] B. WALLNER AND A. ELOFSSON. **Identification of correct regions in protein models using structural, alignment, and consensus information.** *Protein Sci*, **15**(4):900–913, 2006. 39

[89] K OLECHNOVIC AND C. VENCLOVAS. **VoroMQA: assessment of protein structure quality using interatomic contact areas**. `http://www.bti.vu.lt/en/departments/department-of-bioinformatics/software/voromqa`, 2016. Accessed: 2016-11-23. 38

[90] R.A. LASKOWSKI, D.S. MOSS, AND J.M. THORNTON. **Main-chain bond lengths and bond angles in protein structures.** *J Mol Biol*, **231**(4):1049–1067, 1993. 38

[91] R.W. HOOFT, G. VRIEND, C. SANDER, AND E.E. ABOLA. **Errors in protein structures.** *Nature*, **381**(6580):272, 1996. 38

[92] I.W. DAVIS, L.W. MURRAY, J.S. RICHARDSON, AND D.C. RICHARDSON. **MOLPROBITY: structure validation and all-atom contact analysis for nucleic acids and their complexes.** *Nucleic Acids Res*, **32**(Web Server issue):W615–9, 2004. 38

[93] N. SIEW, A. ELOFSSON, L. RYCHLEWSKI, AND D. FISCHER. **MaxSub: an automated measure for the assessment of protein structure prediction quality.** *Bioinformatics*, **16**(9):776–785, 2000. 38

[94] D. FRISHMAN AND P. ARGOS. **Knowledge-based protein secondary structure assignment.** *Proteins*, **23**(4):566–579, 1995. 39

[95] D.T. JONES. **Protein secondary structure prediction based on position-specific scoring matrices.** *J Mol Biol*, **292**(2):195–202, 1999. 39

[96] S. J. HUBBARD AND J. M. THORNTON. **'NACCESS', computer program.** Technical report, Department of Biochemistry Molecular Biology, University College London, 1993. 39

[97] J. CHENG, A.Z. RANDALL, M.J. SWEREDOSKI, AND P. BALDI. **SCRATCH: a protein structure and structural feature prediction server.** *Nucleic Acids Res*, **33**(Web Server issue):W72–6, 2005. 39

[98] J.M. BUJNICKI, A. ELOFSSON, D. FISCHER, AND L. RYCHLEWSKI. **LiveBench-2: large-scale automated evaluation of protein structure prediction servers.** *Proteins*, **Suppl 5**:184–191, 2001. 39

[99] D. CHIVIAN, D.E. KIM, L. MALMSTROM, J. SCHONBRUN, C.A. ROHL, AND D. BAKER. **Prediction of CASP6 structures using automated Robetta protocols.** *Proteins*, **61 Suppl 7**:157–166, 2005. 39

[100] H. ZHOU AND Y. ZHOU. **SPARKS 2 and SP3 servers in CASP6.** *Proteins*, **61 Suppl 7**:152–156, 2005. 39

[101] J. CHENG AND P. BALDI. **A machine learning information retrieval approach to protein fold recognition.** *Bioinformatics*, **22**(12):1456–1463, 2006. 39

[102] A.N. TEGGE, Z. WANG, J. EICKHOLT, AND J. CHENG. **NNcon: improved protein contact map prediction using 2D-recursive neural networks.** *Nucleic Acids Res*, **37**(Web Server issue):W515–8, 2009. 39, 40

[103] D. RYKUNOV AND A. FISER. **Effects of amino acid composition, finite size of proteins, and sparse statistics on distance-dependent statistical pair potentials.** *Proteins*, **67**(3):559–568, 2007. 39, 40

[104] J. ZHANG AND Y. ZHANG. **A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction.** *PLoS One*, **5**(10):e15386, 2010. 39, 40

[105] Y. WU, M. LU, M. CHEN, J. LI, AND J. MA. **OPUS-Ca: a knowledge-based potential function requiring only Calpha positions.** *Protein Sci*, **16**(7):1449–1463, 2007. 40

[106] H. DENG, Y. JIA, AND Y. ZHANG. **3DRobot: automated generation of diverse and well-packed protein structure decoys.** *Bioinformatics*, **32**(3):378–387, 2016. 40

[107] G. WANG AND R.L. DUNBRACK, JR. **PISCES: a protein sequence culling server.** *Bioinformatics*, **19**(12):1589–1591, 2003. 40

[108] K OLECHNOVIC AND C. VENCLOVAS. **Model Quality Assessment and Selection Using VoroMQA**. `http://predictioncenter.org/casp12/doc/CASP12_Abstracts.pdf`, 2016. Accessed: 2016-12-08. 40

[109] A. ZEMLA. **LGA: A method for finding 3D similarities in protein structures.** *Nucleic Acids Res*, **31**(13):3370–3374, 2003. 44, 45

[110] Y. ZHANG AND J. SKOLNICK. **TM-align: a protein structure alignment algorithm based on the TM-score.** *Nucleic Acids Res*, **33**(7):2302–2309, 2005. 45