VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

# DETEKCE KORELOVANÝCH MUTACÍ

DETECTION OF CORRELATED MUTATIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. TOMÁŠ IŽÁK
AUTHOR

VEDOUCÍ PRÁCE Ing. TOMÁŠ MARTÍNEK, Ph.D.
SUPERVISOR

BRNO 2013

# Abstrakt

Tato práce zkoumá existující možnosti a metody detekce korelovaných mutací v proteinech. Práce začíná teoretickým úvodem do zkoumané problematiky. Využití informací o korelovaných mutací je především při predikci terciální struktury proteinu či hledání oblastí s významnou funkcí. Dále následuje přehled v současnosti používaných metod detekce a jejich výhody a nevýhody. V této práci jsou zkoumány zejména metody založené na statistice (například Pearsonově korelačním koeficientu nebo Pearsonově $\chi^2$ testu), informační teorii (Mutual information - MI) a pravděpodobnosti (ELSC nebo Spidermonkey). Dále jsou popsány nejdůležitější nástroje s informací o tom, které metody používají a jakým způsobem. Také je diskutována možnost návrhu optimálního algoritmu. Jako optimální z hlediska úspěšnosti detekce je doporučeno využít více zmíněných metod. Také je doporučeno při detekci využít fyzikálně-chemických vlastností aminokyselin. V praktické části byla vyvinuta metoda využívající fyzikálně-chemických vlastností aminokyselin a fylogenetických stromů. Výsledky detekce byly porovnány s nástroji CAPS, CRASP a CMAT.

# Abstract

This work explores existing possibilities and methods of correlated mutations detection in proteins. At the beginning a theoretical background into explored area is provided. Exploitation of detected correlated mutations lies in a protein's tertiary structure prediction or searching functionally important sites. A state-of-the-art of existing tools and methods follows. In this work, methods based on statistics (for example Pearson correlation coefficient or Pearson's $\chi^2$ test), Information theory (Mutual information - MI) and likelihood models (ELSC or Spidermonkey) are examined. The next part is devoted to the searching for an optimal algorithm for correlated mutations detection. To combine results from multiple different algorithms, is proposed as an optimal solution. It is also advised to exploit physico-chemical properties of amino acids during the detection. In practical part, the algorithm for detection of correlated mutations was developed. It is based on physico-chemical properties of amino acids and phylogenetic trees. Results gained using this method were compared with results gained from CAPS, CRASP and CMAT tools.

# Klíčová slova

korelované mutace, koevoluce, protein, vzájemná informace, entropie, statistické metody, pravděpodobnostní modely, nástroje pro detekci, korelace, fyzikálně-chemické vlastnosti aminokyselin, fylogeneze, algoritmus

# Keywords

correlated mutations, co-evolution, protein, mutual information, entropy, statistical methods, likelihood models, detection tools, correlation, physico-chemical properties of amino acids, phylogeny, algorithm

# Citation

Tomáš Ižák: Detection of Correlated Mutations, master's thesis, Brno, FIT Brno University of Technology, 2013

# Detection of Correlated Mutations

## Prohlášení/Declaration

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechny zdroje, ze kterých jsem čerpal.
I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

. . . . . . . . . . . . . . . . . . . . . . .
Tomáš Ižák
May 17, 2013

## Acknowledgement

First of all, I would like to thank my Master's Thesis advisors Ing. Ivana Burgetová, Ph.D. and Ing. Matěj Lexa, Ph.D. for their comments during the writing of my term project. I would also like to thank my supervisor Ing. Tomáš Martínek, Ph.D. for his guidance. And, finally, thanks to my family and people close to me for their support.

# Contents

# Chapter 1

# Introduction

This work is devoted to correlated mutations in proteins, especially to possibilities of their detection. In this work, I would like to make a brief introduction to this topic, which interferes with multiple science disciplines such as proteomics, genomics or information technology.

Detected correlated mutations are useful in a protein's tertiary structure and its function prediction. This implies possible usage in the protein engineering and detection of genetic diseases are also closely connected with correlated mutations.

In the Theoretical part you will find a theoretical background needed for understanding of this work. Biological background is presented in the section 2.2, statistical background in the section 2.3 and last theoretical part, information theory, can be found in the section 2.4. Definition of algorithm for searching of correlated mutations is presented in the section 2.2.2 and motivation for this searching is described in the section 2.2.3.

The next chapter, State-of-the-art, provides a description of existing tools and algorithms for the detection of correlated mutations. Methods and tools based on SCA (Statistical coupling analysis), MI (Mutual information), Pearson correlation coefficient, Pearson's $\chi^2$ test and likelihood models. All of these approaches have their own advantages and disadvantages, that's why their comparison can be found there, too.

Next chapter describes developed algorithm for correlated mutations detection and its characteristics. This algorithm is based on multiple sequence alignment (MSA), phylogenetic tree (PT) and physicochemical properties of amino acids. This chapter is followed by tests - namely comparison with other available tools for detection of correlated mutations, comparison with different penalizations and verification tests.

Summary and conclusion of this work can be found in the final chapter.

The theoretical part (excluding the section 2.1) was made for the term project on the same subject.

# Chapter 2

# Theoretical part

The first section of this chapter describes basics of biological process of a protein creation.

Section 2.2 is devoted to the description of correlated mutations, section 2.3 provides necessary introduction into the Statistical theory and the last section (2.4) into the Information theory.

## 2.1 Biological background

DNA is a bearer of all information about an organism, including definitions of proteins. According to the central dogma of a molecular biology, protein genes (DNA sequences) are transcribed into RNA sequences and then translated into amino acid sequences (proteins). DNA sequence is passed from the parent organism to the child organism, but these sequences do not have to be exactly the same due to recombination or mutation. Change in DNA in protein gene causes change in the transcribed RNA and can cause change in an amino acid in the translated protein.

Although DNA sequences has known and simple 3D structure (double-helix with complementary DNA sequence), RNA sequences and proteins makes more complicated structures which defines their functions. Protein folds itself into one or more stable structures.

There are these basic types of bonds participating on stability of a protein's tertiary structure:

- covalent - picture 2.1
- noncovalent - picture 2.4
  - ion - Ion interactions are present between two charges and are responsible for substrate binding in proteins. Picture 2.3.
  - hydrogen - picture 2.2
  - Van der Waals

Next physicochemical property is a polarity. Some of amino acids have side chains, which are polar, this means that positive or negative charges are located in different locations. So there are for example six amino acids, which have polar side chain and are also neutral.

There is also another important physicochemical property participating in protein folding. This property is hydropathia (hydrophilia or hydrophobia) of amino acids. Amino acids with these properties tend to be close to amino acids from the same group because

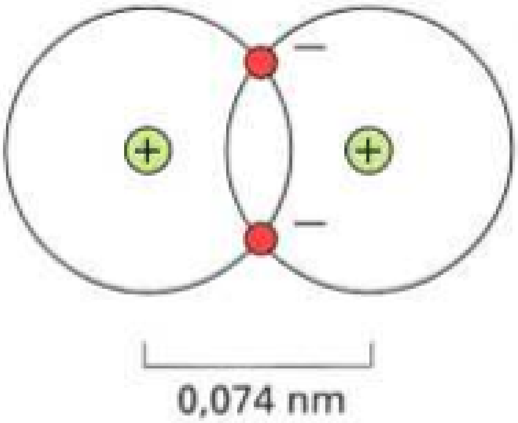


*Figure 2.1: Covalent bond between two hydrogens. Picture taken from [2].*

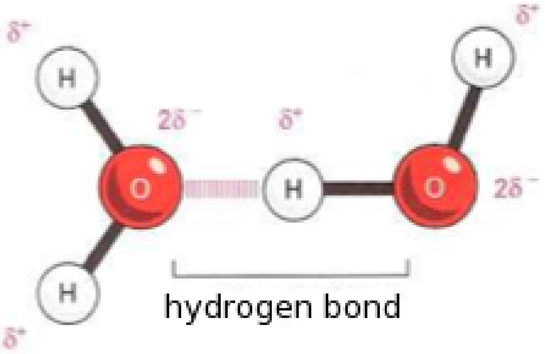


*Figure 2.2: Hydrogen bonds between two molecules of water. Picture taken from [2].*

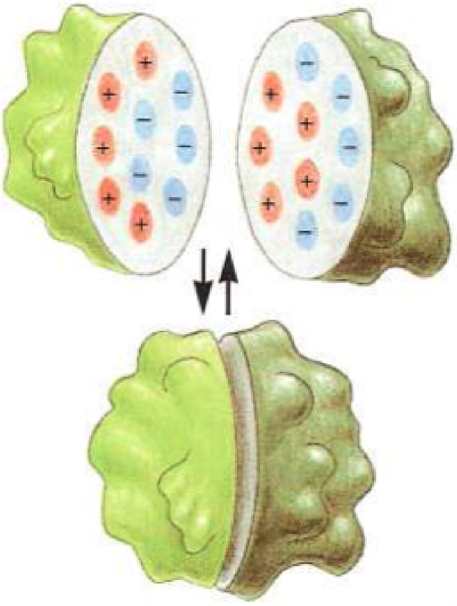


*Figure 2.3: Two proteins bind each other using different charges on the surface. Picture taken from [2].*

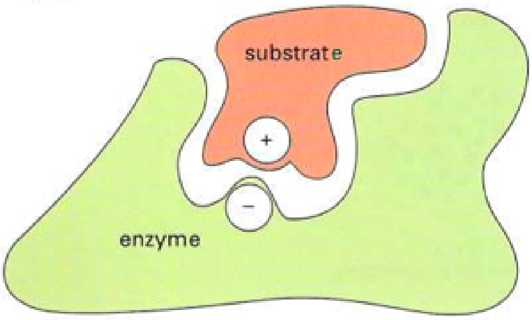


*Figure 2.4: Noncovalent bonds are responsible for substrate bindings. Picture taken from [2].*

of an aqueous environment inside cells. This is the reason why proteins tend to fold - to minimalize all intrusive forces on hydrogen bonds between water molecules.

## 2.2 Correlated mutations in proteins

### 2.2.1 Description of correlated mutations

This section describes different types of correlated mutations (CMs) and problems, which make their detection difficult.

In the previous chapter mutations in proteins were defined. But what are correlated mutations? If we studied many protein sequences (for example multiple sequence alignment) from one protein family, we would find sets of amino acids, which are mutated together only. To be more specific, one amino acid's mutation is correlated with other mutations. The reason why this becomes, is that if these amino acids do not mutate simultaneously, the whole protein collapses or loses some important functions and becomes useless. This is usually the cause of a genetic disease, which prevents spreading of this mutation to

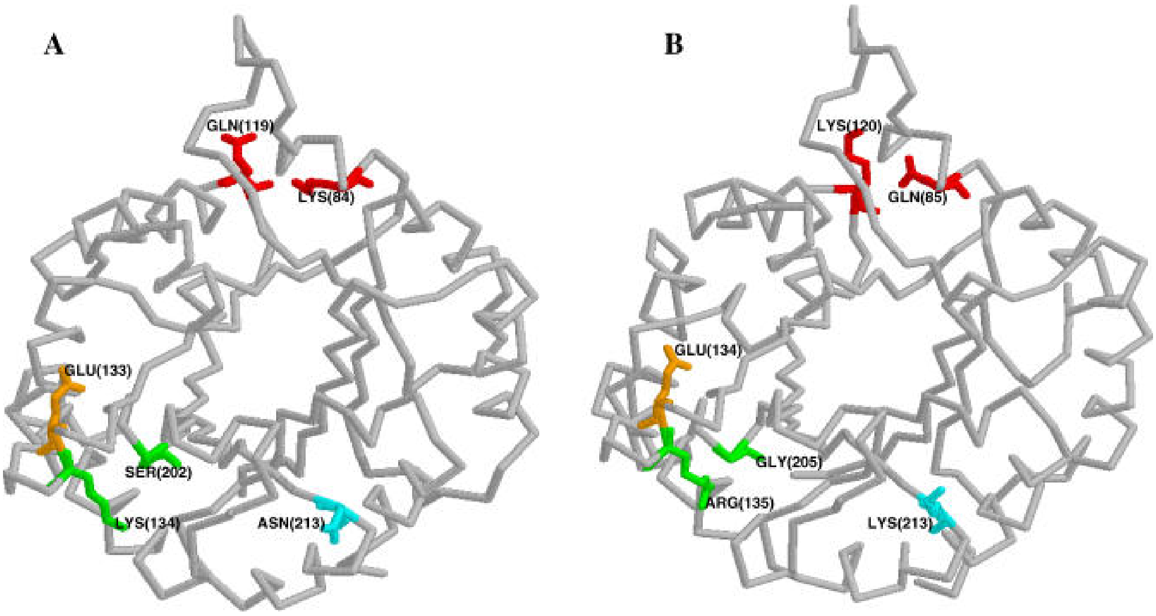


*Figure 2.5: This picture illustrates correlated mutations in two similar proteins. Structural CMs are highlighted with the same color (LYS(134) SER(202) and GLN(119) LYS(84) in the part A). Picture taken from [3].*

the next generations. In other words, for mutations destroying protein's function to survive purifying selection, the fitness of the protein must be rescued through compensatory mutations, resulting in correlated mutations. Example of these correlated mutations can be seen on the picture 2.5.

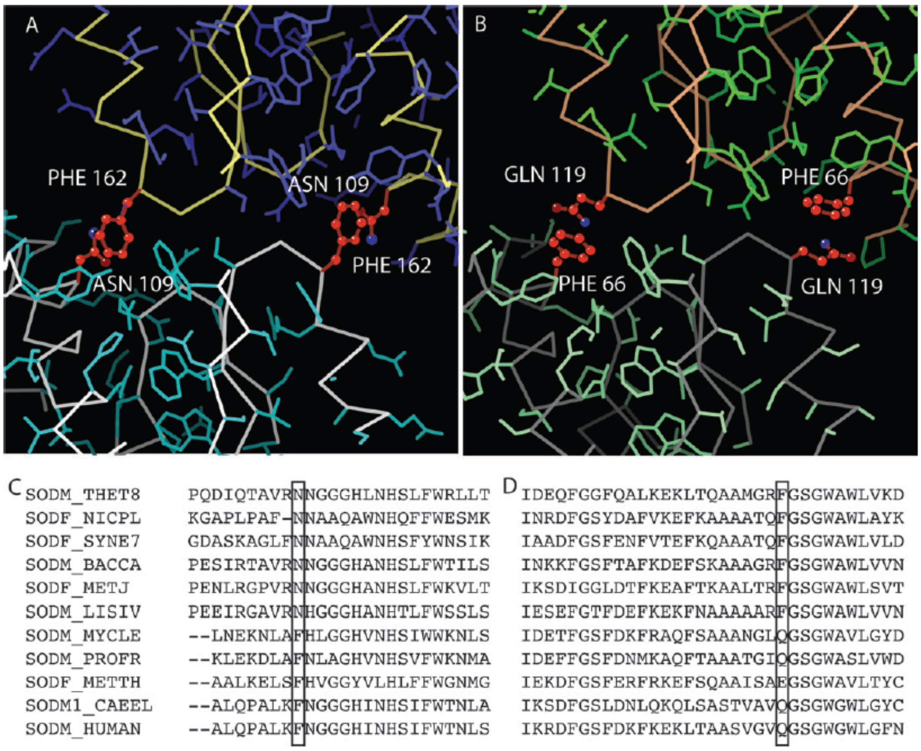
When a protein becomes useless it can be from these main reasons:

- **Structural constraints** - In this case mutations destroyed some important structural contacts, which caused total or partial collapse of the protein and losing its function (totally or partially). This mutation type is the most interesting for us. If we are able to identify structural CM, we can use this information for predicting protein's 3D structure. But it is not easy to distinguish structural and functional (described below) CMs. This problem has been solved for many years and is also described in the section 2.2.3. If we compare structural and functional correlated pairs, analysis published in the article [24] pointed out that only 16.4% of correlated pairs had a distance less than 5.5Å which would imply physical contact. That means that structural correlations are in minority.

- **Functional constraints** - Research has pointed out that co-evolving amino acids were often found to be in close proximity to functionally important sites. In this case structural ties are unaffected, but protein's products were affected by mutations, which caused losing some functions of the protein.

- **Interactional constraints** [28, 38] - These constraints make interaction with another protein possible (picture 2.3), but they are marked as functional constraints in many cases. It is not needed to bother with this extra constraint class for the purpose of CMs detection within protein on which this work is focused.

To recognize the reason of losing protein's function is very difficult because of mul-



```
SODM_THET8     PQDIQTAVRNNGGGHLNHSLFWRLLT    IDEQFGGFQALKEKLTQAAMGRFGSGWAWLVKD
SODF_NICPL     KGAPLPAF-NNAAQAWNHQFFWESMK    INRDFGSYDAFVKEFKAAAATQFGSGWAWLAYK
SODF_SYNE7     GDASKAGLFNNAAQAWNHSFYWNSIK    IAADFGSFENFVTEFKQAAATQFGSGWAWLVLD
SODM_BACCA     PESIRTAVRNNGGGHANHSLFWTILS    INKKFGSFTAFKDEFSKAAAGRFGSGWAWLVVN
SODF_METJ      PENLRGPVRNNGGGHANHSLFWKVLT    IKSDIGGLDTFKEAFTKAALTRFGSGWAWLSVT
SODM_LISIV     PEEIRGAVRNHGGGHANHTLFWSSLS    IESEFGTFDEFKEKFNAAAAARFGSGWAWLVVN
SODM_MYCLE     --LNEKNLAFHLGGHVNHSIWWKNLS    IDETFGSFDKFRAQFSAAANGLQGSGWAVLGYD
SODM_PROFR     --KLEKDLAFNLAGHVNHSVFWKNMA    IDEFFGSFDNMKAQFTAAATGIQGSGWASLVWD
SODF_METTH     --AALKELSFHVGGYVLHLFFWGNMG    IEKDFGSFERFRKEFSQAAISAEGSGWAVLTYC
SODM1_CAEEL    --ALQPALKFNGGGHINHSIFWTNLA    IKSDFGSLDNLQKQLSASTVAVQGSGWGWLGYC
SODM_HUMAN     --ALQPALKFNGGGHINHSIFWTNLS    IKRDFGSFDKFKEKLTAASVGVQGSGWGWLGFN
```

*Figure 2.6: Demonstration of an input and output of a detection algorithm. Taken from [28].*

tiple causes. One of them is not clear-cut distinction between functional and structural constraints. For example mutation of an important structural contact often implies also functional change in protein. Results in the article [24] indicate that both structurally and functionally important positions within folded protein could be likely targets for disease-associated point mutations. For the purpose of revelation the reason of losing protein's function, conserved positions could be used. More information about conservation can be found in the section 2.2.4.

Another one is because of the high cooperativity of protein folding and the plasticity of protein structure, the compensatory response to a point mutation may be distributed over a cluster of residues rather than occur at a single paired residue.[17] To describe this point in other words, some single mutations don't change protein's structure immediately, it is needed more analogical mutations nearby to accumulate a destabilization force within a protein structure for its change.

### 2.2.2 Definition of the detection algorithm

In this work, multiple sequence alignment (MSA or MA) is used as an input data for detection. Example of a MSA is shown on the picture 2.6 (section C and D) where correlated mutations are highlighted (the first position is in the picture section C, second in the section D). These positions are correlated. However this picture demonstrates inter-molecular coevolution, let's imagine it is in a single protein (intramolecular coevolution is the main subject of this work). Highlighted positions correspond to ASN(109) PHE(162) in the picture section A and PHE(66) GLN(119) in the section B.

Task of algorithms interesting for this work is to find these correlated positions which imply coevolution. For these purpose various mathematical methods appear to be effective.

Detected correlated mutations can be represented as a binary matrix, where protein sequence is on both axis and value in this matrix is 1 (true - between concrete two positions in protein sequence a CM was detected) or 0 (false - between two positions a CM was not detected). Instead of binary matrix, usual matrix can be used. Matrix values are similar to computed values in this case - there is a color assigned to each cell according to its value - output is a picture. Next option of a representation is a text file where positions of detected CMs and their percentual probabilities are written. Detected CMs can be highlighted in the corresponding protein model if this model is known.

To express the efficiency of the predictors several statistical indices are used [11]. The first and the most used index to evaluate the predictor accuracy is defined as:

$$A = \frac{N_{cp}*}{N_{cp}} \tag{2.1}$$

where $N_{cp}*$ and $N_{cp}$ are the number of correctly assigned CMs and total predicted CMs, respectively. Routinely the accuracy is evaluated for each protein and then averaged over the protein set under consideration. The improvement over a random predictor is evaluated by computing the ratio between $A$ and the accuracy of a random predictor ($\frac{N_c}{N_p}$):

$$R = \frac{A}{\frac{N_c}{N_p}} \tag{2.2}$$

where $N_c$ is the number of real CMs in the protein of length $L_p$, and $N_p$ are all possible CMs.

### 2.2.3 Exploitation of detected CMs

At the first sight it looks like that correlated mutations can't be very useful. But it can be appropriate to gaining more information about proteins. For example, if we have multiple alignment (which is also needed as an input for detection of CMs) and detected CMs, we can gain approximate structure of a protein. But as I wrote in the previous section, it is difficult to recognize which type (functional or structural) of losing protein's function each correlated mutation causes. In this case we would like to know that information, because pure functional reasons are uninteresting in this case. At present, there is no appropriate algorithm for solving this problem. That's why neural networks are sometimes used for a prediction of the protein's contact map, which is then used as an input for the calculation of protein's tertiary structure [20, 12, 10, 11].

The distance involved in the different definitions of a contact can be that between $C_\alpha - C_\alpha$ atoms, between $C_\beta - C_\beta$ atoms, and the minimal distance between atoms belonging to the side chain or to the backbone of the two residues. For the most strongly correlated residue pairs predicted to be in contact (case when two large molecules or amino acids in a backbone of the protein are closer than some threshold, which is usually somewhere between 4.5Å and 8Å), the prediction accuracy ranges from 37% to 68% and improvement ratio relative to a random prediction from 1.4 to 5.1, when simple and general method presented in the article [17] is used.

Research presented in the article [24], devoted to disease-associated point mutations, confirms that correlated mutations go well beyond contact prediction and are hallmarks of amino acid positions leading to disease when affected by mutation. Next interesting point presented there is that positions correlated with increasing numbers of other positions are increasingly more likely to be associated with a disease.

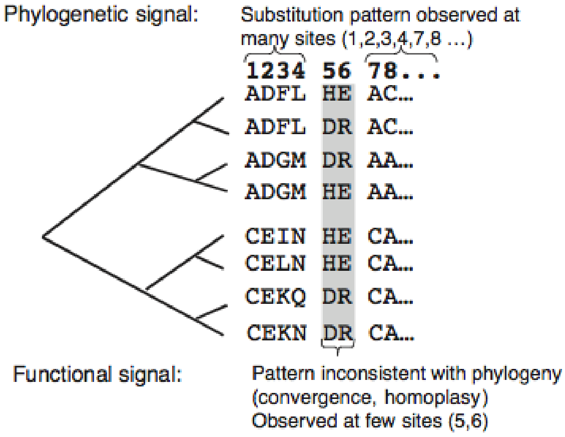


*Figure 2.7: The phylogenetic noise illustration. Picture taken from [39].*

Next area where CMs can be useful is protein engineering and design. It helps improving stability and saves time for example during designing new proteins. It is also supposed to use correlated mutations for redesigning enzyme activity and selectivity. However, reliable identification of correlated mutations by computational methods is still needed for its potential implications for a protein design. As written in the article [32], if present limitations can be overcome, this aspect of a molecular evolution may come to form the basis for new powerful design tools.

### 2.2.4 Accuracy problems

This last part of correlated mutations description is devoted to more advanced problems which should be solved for making CMs detection more accurate. At first, there is a problem with noise, which should be eliminated. Second, value of position's conservation can be used for distinguishing types of CMs.

#### Noise reduction

It is not so easy to detect correlated mutations, because it is also needed to resolve problems with noise, which make detection algorithms inaccurate.

Stochastic noise is usually present in regular input datasets for many other purposes besides CMs detection. It is caused by finite number of sequences. That is why stochastic noise reduction has to be solved. Many statistical methods are used for eliminating this problem.

Also, there usually is a phylogenetic noise (illustrated on the picture 2.7) in an input dataset along random noise. Phylogenetic noise is caused by multiple protein subfamilies in an input multiple alignment, which were evolving separately. If we didn't consider this noise, we could detect some non-existing CMs or on the other hand, more likely, not detect some important CMs. At present, this problem is solved for example by using phylogenetic trees and analyzing its subtrees separately. This approach shows promising results and that's why it is used in present tools very often. Phylogenetic tree considers back mutations and multiple mutations in a single branch. This approach is for example used in detection tool Spidermonkey (the section 3.6.2).

### Conserved positions

Since mutations at sites critically affecting the fitness of proteins, they are eliminated by purifying selection. These sites usually tend to be conserved. Conserved sequences/positions are sequences/positions, which are not changed very much, because they are too sensitive to mutations. Information about conserved positions is also sometimes used as an additional input data. We can find a relation between functional CMs and conservation also.

The most conserved positions are mainly located in the active site cleft, and the intermediately conserved positions are clustered in other functionally important regions. This finding is consistent with the intuitive expectation that a proper measure of conservation should be able to map functionally important sites of a protein. [41]

This implies problems with structural CMs revelation. That's why, at present, conservation is not used for CMs detection very often. However it can be used in another way. The problem of predicting a protein fold from a sequence information alone is difficult one, but if information about conservation is also used, structural and functional CMs can be distinguished with some efficiency.

There are various methods to estimate position-specific amino acid frequencies according to the [35]:

- Unweighted amino acid frequencies

  $$f_a^u(i) = \frac{n_a(i)}{n(i)} \tag{2.3}$$

  where $n_a(i)$ is number of sequences in which position $i$ is occupied by amino acid $a$ and $n(i)$ is the total number of aligned sequences in which position $i$ is present. There should be no gaps allowed.

- Weighted amino acid frequencies

  $$f_a^w(i) = \frac{\sum\limits_{k=1}^{n(i)} \delta(a,k,i) w_k}{\sum\limits_{k=1}^{n(i)} w_k} \tag{2.4}$$

  where $w_k$ is a given weight of a sequence $k$, and $\delta(a,k,i)$ is 1 if amino acid $a$ is in sequence $k$ at position $i$, else 0.

- Estimated independent counts

  $$f_a^{ic}(i) = \frac{n_a^{ic}}{n^{ic}} \tag{2.5}$$

  where $n_a^{ic}$ is an estimate of the number of independent observations of amino acid $a$ at position $i$ and $n^{ic}(i) = \sum\limits_{a=1}^{20} n_a^{ic}$. The idea behind this approach is to correct for the correlation between aligned sequences.

The conservation index is calculated in the next step from amino acid frequencies by one of the following strategies:

- Entropy-based measure

$$C^e(i) = \sum_{a=1}^{20} f_a \ln f_a(i) \tag{2.6}$$

  Entropy for a position $i$ is maximal if all 20 amino acids at this position have equal frequencies. Entropy with the reverse sign defined on position-specific frequencies $f_a(i)$ is used to estimate the conservation index. Entropy does not take into account possible bias in amino acid composition or similarities among amino acids. Entropy is described in the section 2.4.

- Variance-based measure

$$C^v(i) = \sqrt{\sum_{a=1}^{20} (f_a(i) - f_a)^2} \tag{2.7}$$

  A similar method has been employed in the estimation of a evolutionary conservation and coupling parameters. The advantage of this method is the use of overall amino acid frequencies, which differ for different protein families. This measure does not take into account similarities among amino acids.

- Sum of pairs measure

$$C^p(i) = \sum_{a=1}^{20} \sum_{b=1}^{20} f_a(i) f_b(i) S_{ab} \tag{2.8}$$

  where $S_{ab}$ is an amino acid scoring matrix. This conservation index will be higher for positions occupied by more similar amino acids.

This property is sometimes used also for detection of a bad alignment or for contact map prediction using neural networks and correlated mutations.

### Inaccurate multiple alignment

All of CMs detection algorithms (focused in this work) use a MSA as an input data. But tools for multiple alignment creation are not very reliable. That implies algorithms for detecting CMs work with inaccurate inputs, which leads to accumulating errors and, of course, bad detection in the result. The only way to eliminate this unpleasant phenomenon is to use a correct multiple alignment at first. But for creating a perfect multiple alignment, knowledge about protein's structure or correlated mutations is needed. If the information about protein's structure is not available, CMs should be used. But it is almost impossible to gain correlated mutations if neither reliable MSA nor 3D structure knowledge is provided. The only imaginable way how to achieve correct CMs detection from regular MSA is to make this process iterative. It is needed to detect CMs from regular (and also inaccurate) MSA, at first. Then, using these detected CMs, could be possible to make better MSA. This procedure can be repeated until results of two following iterations are different. This method could provide more accurate both MSA and CMs.

As written in the previous paragraph, current tools for multiple alignment creation are not very accurate. One of the reasons is that recombination is not supported because used algorithms are not able to detect this type of mutation. Further, deletion and insertion causes problems in existing tools but not so significant. The last mutation type - point-mutation - makes almost no problems to current MSA tools.

## 2.3 Statistical theory

### 2.3.1 Statistical functions for a random variable

**Probability function** in a value $x$ express likelihood that random variable $X$ is equal to $x$.

$$p(x) = P(X = x) \tag{2.9}$$

**Expected value** of a random variable is the weighted average of all possible values that this random variable can take on. The weights used in computing this average correspond to the probabilities in the case of a discrete random variable, or densities in the case of a continuous random variable.

$$E(X) = \sum_{x_i} x_i.p(x_i) \tag{2.10}$$

The **variation** is a measure of how far a set of numbers is spread out from each other. It is one of several descriptors of a probability distribution, describing how far the numbers lie from the mean (expected value). In particular, the variance is one of the moments of a distribution. In that context, it forms part of a systematic approach to distinguishing between probability distributions.

$$\sigma^2(X) = var(X) = D(X) = \sum_{x_i} ((x_i - E(X))^2.p(x_i)) \tag{2.11}$$

### 2.3.2 Statistical functions for a random variable couple

**Covariance** is a measure of how much two variables change together. Variance is a special case of the covariance when the two variables are identical.

$$cov(X,Y) = E\{[X - E(X)].[Y - E(Y)]\} \tag{2.12}$$

Relation between two variables can be determined according to the following rules:
$cov(X,Y) = 0 \ldots$ Two random variables are not linearly dependent.
$cov(X,Y) > 0 \ldots$ Two random variables are linearly dependent with a direct proportion.
$cov(X,Y) < 0 \ldots$ Two random variables are linearly dependent with an inverse relation.

The **Pearson product-moment correlation coefficient** (PPMCC or PCC) is a measure of the linear dependence correlation between two variables, giving a value from $< -1, 1 >$. Pearson's correlation coefficient between two variables is defined as the covariance of the two variables divided by the product of their standard deviations.

$$\rho(X,Y) = \frac{cov(X,Y)}{\sqrt{D(X)}.\sqrt{D(Y)}} \tag{2.13}$$

Output values of this function are interpreted as follows:
$\rho(X,Y) = 0 \ldots$ Two random variables are not correlated.
$\rho(X,Y) = 1 \ldots$ Dependency graph between two random variables is a growing line.
$\rho(X,Y) = -1 \ldots$ Dependency graph between two random variables is a decreasing line.

### 2.3.3 Pearson's $\chi^2$ test

This section is inspired by source [30].

Pearson's $\chi^2$ test is a standard method for measuring or detection correlation between two variables. It uses evaluation of the difference between observed frequencies of each value combination and expected frequencies. These two variables are supposed to be independent for gaining expected frequencies. For this purpose, contingency table (as shown below) is used.

$$\begin{array}{cccccc} & Y_1 & Y_2 & \cdots & Y_S & \sum \\ X_1 & a_{11} & a_{12} & \cdots & a_{1S} & r_1 \\ X_2 & a_{21} & a_{22} & \cdots & a_{2S} & r_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ X_R & a_{R1} & a_{R2} & \cdots & a_{RS} & r_R \\ \sum & s_1 & s_2 & \cdots & s_S & \text{n} \end{array} \tag{2.14}$$

Expected frequency for frequency $a_{kl}$ is expressed by following equation:

$$e_{kl} = \frac{r_k s_l}{n} \tag{2.15}$$

where

$$r_k = \sum_{l=1}^{S} a_{kl}, \tag{2.16}$$

$$s_l = \sum_{k=1}^{R} a_{kl} \tag{2.17}$$

and

$$n = \sum_{l=1}^{S} \sum_{k=1}^{R} a_{kl} \tag{2.18}$$

Pearson's $\chi^2$ is then computed as follows:

$$\chi^2 = \sum_{l=1}^{S} \sum_{k=1}^{R} \frac{(a_{kl} - e_{kl})^2}{e_{kl}} = n \sum_{l=1}^{S} \sum_{k=1}^{R} \frac{(a_{kl} - \frac{r_k s_l}{n})^2}{r_k s_l} \tag{2.19}$$

Two variables are dependent if following equation is true.

$$\chi^2 \geq \chi^2_{(R-1)(S-1)}(\alpha) \tag{2.20}$$

It is noticeable that this is a statistical significance test described in the section 2.3.4.

One variable in the case of CMs detection means one position in a MSA. Then every possible position pair (two variables) is analyzed using this Pearson's $\chi^2$ test. This method is purely statistical and not very efficient because it suffers from multiple kinds of noise, for purpose of CMs detection. But it is useful in combination with other approaches for CMs detection.

### 2.3.4 Statistical significance test

This subsection about statistical significance test (also known as statistical hypothesis testing) is based on [34].

Hypothesis about which it is needed to decide if it is true (valid) or false (invalid), is labeled as $H_0$ and called null hypothesis. There is an alternative hypothesis $H_1$ often, which is true when $H_0$ is false. Null hypothesis is tested against alternative hypothesis. If null hypothesis is tested for accepting or rejecting, one of these two errors can occur:

1. Null hypothesis is rejected, although it is true - known as **type one error**. Probability of this situation is labeled as $\alpha$. $\alpha$ is also known as significance level of a test.

2. Null hypothesis is accepted, although it is false - known as **type two error**. Probability of this situation is labeled as $\beta$. Expression $1 - \beta$ is also known as power of a test.

Rejecting or accepting of the null hypothesis is based on a deciding rule, but it is almost impossible both to achieve zero error ($\alpha = 0$ and $\beta = 0$) and to decrease both $\alpha$ and $\beta$. That's why $\alpha$ is set to some small value in advance and $\beta$ is then needed to be as small as possible.

In the case of rejecting of a null hypothesis, $H_0$ can be claimed as false ($H_1$ as true) with relatively small chance of error $\alpha$. But in the case of not rejecting of a null hypothesis, it is not possible to claim something about hypothesis validity often. It is only possible to say that null hypothesis cannot be rejected based on required small error chance $\alpha$.

That hypothesis needed to be proved reliably enough is labeled as alternative. If $H_0$ is rejected against $H_1$, alternative hypothesis $H_1$ is said to be statistically significant. If $H_0$ is not rejected, $H_1$ is said to be statistical insignificant.

Various functions as z-tests, t-tests or $\chi^2$-test can be used for test statistics (labeled as $R$). $|R|$ (or $R$) is compared to a value $z_{\alpha/2}$ (or $z_\alpha$) gained from statistical tables (it is also needed to know probability distribution). If $|R| > z_{\alpha/2}$ is true, then $H_0$ is rejected. If $|R| \leq z_{\alpha/2}$ is true, then $H_0$ is not rejected.

This test is especially used in the cases when tested data are very large (random data are selected) or incomplete.

Minimal significance level of a test when it was possible to reject null hypothesis on measured data, is called **p-value**.

## 2.4 Information Theory

Information Theory is also useful in a CMs detection. Entropy is a frequent word in articles about CMs detection. But what entropy is? Entropy provides a key measure of an information usually expressed by the average number of bits needed to store one symbol in a message. It quantifies the uncertainty involved in predicting the value of a random variable. This entropy is often used in data compression.

Entropy of $X$ is defined:

$$H(X) = -\sum_{x \in \mathbb{X}} p(x) \log p(x) \tag{2.21}$$

There are two more modifications of an entropy. The first one is the Joint entropy, which is merely the entropy of pairing of two discrete random variables.

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y) \tag{2.22}$$

The second one is the Conditional entropy defined as follows.

$$H(X|Y) = H(X,Y) - H(Y) = -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(y)} \tag{2.23}$$

Next part of Information theory is the Mutual information (MI), which measures the amount of information, which can be obtained about one random variable by observing another. MI can be expressed by following equations:

$$MI(X,Y) = I(Y,X) = H(X) + H(Y) - H(X,Y) \tag{2.24}$$

$$MI(X,Y) = H(X) - H(X|Y) \tag{2.25}$$

$$MI(X,Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.26}$$

If you want to learn more about an entropy and the Mutual information in CMs detection, read the section 3.3.

# Chapter 3

# State-of-the-art

At the beginning of this chapter, methods and tools are categorized into basic groups. Afterwards some of them are listed and described. Last section of this chapter is devoted to the selection of the best method for correlated mutations detection. Summary of the evolution of correlated mutation's detection in time are captured on the picture 3.1.

## 3.1 Basic types of detection

There are few basic types of detection of CM. At first, we divide algorithms according to its input data.

- **Detection from multiple alignment**

  This type of detection is very often due its availability and is subject of this work.

- **Detection from protein structure**

  This detection is different from previous approach, it is useful and accurate for revealing structural CMs. This work is not concerned on this type of detection, but it can provide supplementary information about structural CMs.

Methods and tools for detection CMs from MSA can be classified into one or more following categories according to their approaches.

- **Statistical approach** - Pearson's $\chi^2$ test, covariance coefficient, Statistical coupling analysis, Mutual information

  These methods are bases of CM detection and are often improved for this purpose.

- **Phylogenetic approach** - the normalized coevolutionary pattern similarity (NCPS) score, Spidermonkey

  This approach should ensure a better elimination of a phylogenetic noise. For example, a phylogenetic tree can be used here for eliminating invariant sites in subfamilies.

- **Biophysical approach**

  This approach should be the most accurate, but it is not used very often because of high complexity. The best method should be to use *abinitio* (*denovo*) modeling for output validation, but complexity would be very high. That's why only some biophysical (physicochemical) properties are used. CRASP is an example of this approach.

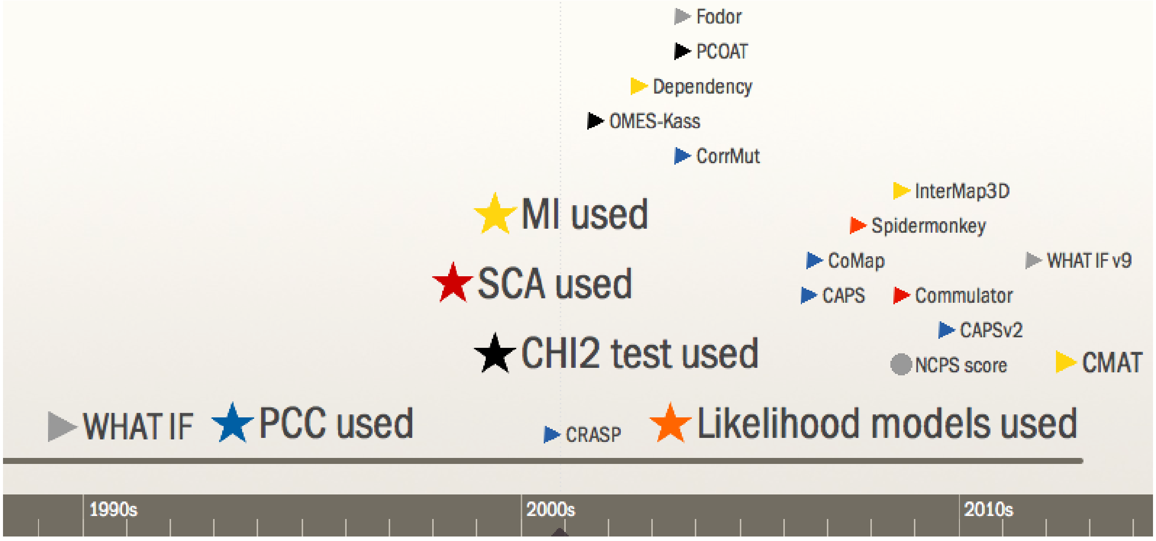


*Figure 3.1: Timeline with milestones in detection of correlated mutations*

## 3.2 SCA based detection

Statistical coupling analysis (SCA) for correlated mutations was presented in the paper [27] for the first time. Thermodynamic mutant cycle analysis, a technique that measures the energetic interaction of two mutations, provides a direct method to CMs detection.

Statistical coupling of two sites, $i$ and $j$, is defined as the degree to which amino acid frequencies at site $i$ change in response to a perturbation of frequencies at another site, $j$. This definition of coupling does not require that the overall conservation of site $i$ change upon perturbation at $j$, but only that the amino acid population be rearranged.

For an evolutionarily well sampled MSA, where additional sequences do not significantly change the distribution at sites, the probability of any amino acid $x$ at site $i$ relative to that at another site, $j$, is related to the statistical free energy separating sites $i$ and $j$ for amino acid $x$ by the Boltzmann distribution.

$$\frac{P_i^x}{P_j^x} = e^{\frac{\Delta G^x_{i \to j}}{kT*}} \tag{3.1}$$

where $kT*$ is an arbitrary energy unit.

In this method the conservation at each site $i$ is defined as:

$$\Delta G_i^{stat} = kT * \sqrt{\sum_x (\ln \frac{P_i^x}{P^x})^2} \tag{3.2}$$

or simply

$$\Delta G_i^{stat} = \sqrt{\sum_x (\ln P_i^x)^2} \tag{3.3}$$

The magnitude of the difference in these two energy vectors gives a statistical coupling energy between sites $i$ and $j$, which is more interesting for this work.

$$\Delta\Delta G^{stat}_{i,j} = kT * \sqrt{\sum_x (\ln \frac{P^x_{i|\delta j}}{P^x_{\delta j}} - \ln \frac{P^x_i}{P^x})^2} \tag{3.4}$$

or simply

$$\Delta\Delta G^{stat}_{i,j} = \sqrt{\sum_x (\ln P^x_{i|\delta j} - \ln P^x_i)^2} \tag{3.5}$$

This quantitatively represents the degree to which the probability of individual amino acids at $i$ is dependent on the perturbation at $j$.

Question becomes how to compute kT*. As it is constant, it is simply to deflate as shown above.

Example of the SCA method is a web-based system named Commulator, which is described in the article [25].

## 3.3 MI based methods

The Mutual information (MI) is a well-known measure in Information Theory (see section 2.4). Description of this method is taken from the article [29]. MI is based on Shannon's entropy - a measure of uncertainty for a random variable.

$$H(X) = -\sum_{i=1}^{K} p(x_i) \log_b p(x_i) \tag{3.6}$$

or generally 2.21, where $p(X)$ is associated probability distribution. The choice of logarithm base $b$ serves to scale the entropy. If we have a pair of random variables, then joint or pair entropy is defined as follows:

$$H(X,Y) = -\sum_{i=1}^{K}\sum_{j=1}^{L} p(x_i, y_j) \log_b p(x_i, y_j) \tag{3.7}$$

or generally equation 2.22.

Mutual information - $MI(X,Y)$ - is the reduction of uncertainty of random variable X given random variable Y (see equations 2.24, 2.25 and 2.26).

The MI between two columns in a MSA reflects the degree to which the pattern in the two columns is correlated. If amino acids occur independently at the two sites, the theoretical value for MI is zero.

There are two sources of a background noise in MI: finite sample size effects and phylogenetic influence. Phylogenetic noise in MI is illustrated in the picture 3.2.

Until these days, various modifications of this method were presented. These variants are presented below and other ones are described in the section 3.7.

### 3.3.1 MIp

MIp method corrects the phylogenetic and entropic effects by subtracting the $APC$ (Average Product Correction). $APC$ is the product of the average $MI$ values of two positions ($MI(i,\bar{x})$ and $MI(j,\bar{x})$) divided by the average of all positions ($\bar{M}I$) in the alignment.

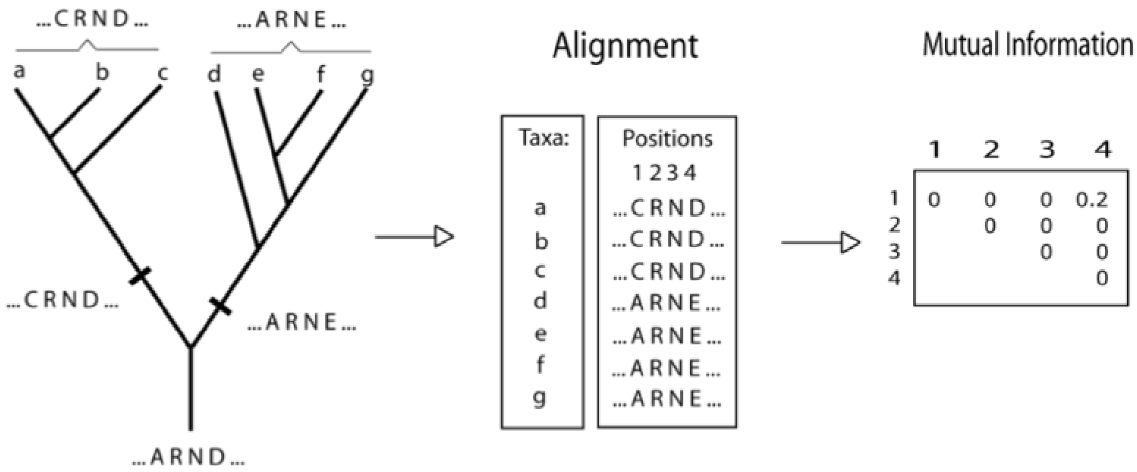
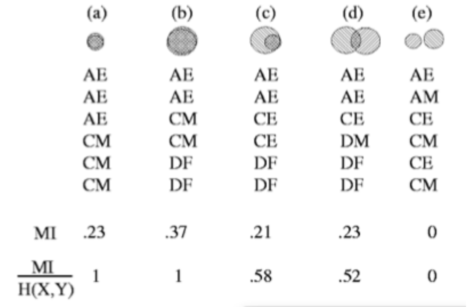


*Figure 3.2: Phylogenetic noise in the Mutual information. Picture taken from [18].*



|  | (a) | (b) | (c) | (d) | (e) |
|---|---|---|---|---|---|
|  | AE | AE | AE | AE | AE |
|  | AE | AE | AE | AE | AM |
|  | AE | CM | CE | CE | CE |
|  | CM | CM | CE | DM | CM |
|  | CM | DF | DF | DF | CE |
|  | CM | DF | DF | DF | CM |
| MI | .23 | .37 | .21 | .23 | 0 |
| $\frac{\text{MI}}{\text{H(X,Y)}}$ | 1 | 1 | .58 | .52 | 0 |

*Figure 3.3: Difference between MI and MI/E. The picture taken from [29].*

$MIp$, which is $MI - APC$, dramatically improves residue contact predictions. In the form of equation:

$$MIp(i,j) = MI(i,j) - APC(i,j) = MI(i,j) - \frac{MI(i,\bar{x})MI(j,\bar{x})}{\bar{MI}} \tag{3.8}$$

### 3.3.2 MIa

The MIa method is defined as follows:

$$MIa(i,j) = MI(i,j) - ASC(i,j) = MI(i,j) - (MI(i,\bar{x}) - MI(j,\bar{x}) + \bar{MI}) \tag{3.9}$$

ASC (Average Sum Correction) instead of APC is used in this method.

### 3.3.3 MI/E

This method was published in the article [19] and is defined as follows:

$$MI/H(X,Y) = \frac{\sum_i \sum_j P(x_i,y_j)\log(\frac{P(x_i,y_j)}{P(x_i)P(y_j)})}{-\sum_i \sum_j P(x_i,y_j)\log(P(x_i,y_j))} \tag{3.10}$$

MI/E then provides information about entropy dependency. Difference between MI and MI/E can be seen on picture 3.3.

### 3.3.4 RCW-MI

Row and Column Weighing of Mutual information was published also in the article [19] and is defined as:

$$RCW(A,B) = \frac{MI_{ij}}{\frac{MI_{i*}+MI_{*j}-2MI_{ij}}{n-1}} \tag{3.11}$$

where $*$ means summation over all sites.

### 3.3.5 MIBP

The MI model with the amino acid background distribution (MIB) and the covariation of physicochemical properties (MIP) were presented in the article [16]. New measure called MIBP was based on both MIB and MIP, and is also described in this article. Results presented there show that the MIBP measure is significantly different from methods based on amino acid distribution.

### 3.3.6 ZNMI

Another modification of MI - ZNMI (Z-scored-product Normalized Mutual information) - is described in the article [4]. At first, MI is normalized by the joint entropy. Then, the assumption that the column NMI distribution can be approximated by a Gaussian distribution, parameterized by the column NMI mean and variance. A z-score is calculated for the product $NMI(i,j)$ using equation presented in this article.

### 3.3.7 Dependency

Dependency procedure is described in [39]. It relies upon the accuracy of the alignment but it does not require any assumptions about the phylogeny or the substitution process. This tool uses Mutual information and unweighted dependency ratio, which uses entropy factor defined in equation 3.22 in section 3.7. Then unweighted dependency ratio is computed:

$$R(X_i,X_j) = D(X_i,X_j)E(X_i,X_j), \tag{3.12}$$

where $E(X_i,X_j)$ is an entropy factor and $D(X_i,X_j)$ is dependency ratio defined in [39]. Final step is to perform a statistical significance test on computed values.

### 3.3.8 InterMap3D

As presented in the article [19], there are two choices of input data types in this tool. At first, input data could be a single protein sequence. Program finds the most homologous sequences (from UniProt using BLASTP) and performs multiple sequence alignment from these sequences, in this case. The second option is to upload MSA as an input data.

Next step which InterMap3D makes is fetching the most similar 3D protein structure from the Protein Data Bank (PDB). This structure is used for visualizing highlighted CMs on 3D picture of the protein. Co-evolving residues are then predicted using three different methods: Row and Column Weighing of Mutual information (section 3.3.4), Mutual information/Entropy (section 3.3.3) and Dependency (section 3.3.7). Finally the results are mapped onto a 3D structure if possible, using the FeatureMap3D program, which searches for the most similar homologous protein with an experimentally determined 3D structure, and then uses PyMOL to plot predicted pairs of co-evolving sites onto that structure.

InterMap3D is freely available web-based tool.

## 3.4 Detection based on Pearson's $\chi^2$ test

### 3.4.1 OMES

Observed Minus Expected Squared (OMES) Covariance Algorithm used in the Fodor package is derived from OMES method Kass and Horovitz [23]. For every possible pair of columns (column $i$ vs. column $j$), it generates a list $L$ of all distinct pairs of amino acids. It discards any pairs that have a gap at either $i$ or $j$. The score for each column pair $i$, $j$ is given by:

$$\sum_{L}^{1} \frac{(N_{obs} - N_{ex})^2}{N_{valid}} \tag{3.13}$$

where $N_{valid}$ is the number of sequences in the alignment that have nongapped residues at both positions $i$ and $j$, $N_{obs}$ is the number of times that each distinct pair of residues was observed, and $N_{ex}$ is the number of times that each distinct pair of residues would be expected based only on the frequency of each residue in each column. The value of $N_{ex}$ for a given pair with residue $x$ at position $i$ and residue $y$ at position $j$ can be calculated by following equation:

$$N_{ex} = \frac{c_{xi}c_{yj}}{N_{valid}} \tag{3.14}$$

where $c_{xi}$ is the number of times residue $x$ occurs at position $i$ and $c_{yj}$ is the number of times $y$ occurs at position $j$.

### 3.4.2 PCOAT: positional correlation analysis using multiple methods

According to the article [37], PCOAT performs the positional correlation analysis in four steps.

First, the effective count of every amino acid pair at each position pair is estimated using three weighting methods (unweighted count, Henikoff weighting (HW) count and altered position-specific independent count (PSIC)). Invariant and gapped positions are removed.

Second, correlation scores of every position pair and amino acid pair are determined with corresponding statistical significances and the pairs that are significantly correlated are identified. Two statistical tests were implemented: $\chi^2$-test and likelihood ratio test.

Next, individual positions that are highly correlated with multiple positions are detected. The Z-scores of results from previous step of each position are calculated and ranked by their statistical significance.

Optional fourth step identifies the networks of highly correlated positions using clustering methods.

PCOAT should be faster than Dependency and CRASP.

## 3.5 Detection based on Pearson's correlation coefficient

### 3.5.1 McBASC

McLachlan Based Substitution Correlation (McBASC) Covariance Algorithm was presented in the publication [33].

At first, a two-dimensional N x N matrix with indexes $k$ and $l$ for each column $i$ of an input MSA, where $N$ is a number of sequences in an input MSA. Values in this matrix are taken from the McLachlan substitution matrix $s$. The correlation score between two columns $i$ and $j$ is defined as:

$$r_{i,j} = \frac{1}{N^2} \frac{\sum\limits_{kl}(s_{ikl} - \bar{s_i})(s_{jkl} - \bar{s_j})}{\sigma_i \sigma_j} \tag{3.15}$$

where $r$ with a score of $+1$ indicating highly co-varying columns. For the special case when columns $i$ and $j$ are identical, $r_{ij}$ is computed by following equation:

$$r_{ij} = \frac{N^2 - 1}{N^2} \tag{3.16}$$

If there is a gap in either sequence $k$ or $l$ at either column $i$ or $j$, $r_{ij} = 0$.

### 3.5.2 CRASP

CRASP is a web-based tool and was introduced in the publication [1]. The program package CRASP consists of three modules. Two modules are designed for the detection of dependent amino acid substitutions at a pair of positions of a protein sequence alignment and the third serves to estimate the statistical significance of the contribution of coordinated substitutions to the variability of the integral physicochemical characteristics of a protein.

Four basic steps are following:

1. Amino acids are translated into chosen physicochemical properties (hydrophobicity, electric charge, side-chain size)

2. Each column is a random variable.

3. Pearson's correlation coefficient is evaluated for each pair of columns (random variables)

4. Statistical test of significance is performed.

### 3.5.3 CAPS

Following section about Perl based tool CAPS (Coevolution Analysis using Protein Sequences) is based on an application note [9].

CAPS identifies co-evolving amino acid site pairs ($e$ and $k$) by measuring the correlated evolutionary variation at these sites. Evolutionary variation is measured using time-corrected BLOSUM values for the transition $(\theta_{ek})_{ij}$ between two amino acids at a particular site when comparing sequence $i$ to sequence $j$ at sites $e$ and $k$. The transition between two amino acids at each site is corrected by the divergence time of the sequences $i$ and $j$. The time is estimated as the mean value of substitutions per synonymous site between the two sequences being compared. Correlation of the mean variability is measured using the Pearson's correlation coefficient. Finally, the significance of the correlation coefficients is estimated by comparing the real correlation coefficients to the distribution of re-sampled correlation coefficients. Only co-evolving sites parsimony informative (if it contains at least two types of amino acids and at least two of them occur with a minimum frequency of two) are considered. The step-down permutation procedure is applied to account for multiple tests and nondependent data.

The phylogenetic co-evolution is removed from MSA by built-in script, which requires specifying the sequence names from each clade to be removed and re-do the coevolution calculations.

When the crystal protein structure is available, CAPS also tests the significance of the distance between the amino acid sites identified as co-evolving, providing useful information about the type of co-evolution. CAPS also performs a preliminary analysis of compensatory mutations by testing the correlation in the hydrophobicity as well as in the molecular weight variations between co-evolving amino acids.

CAPS is now available in version 2 for Unix and Windows based systems with all source codes. Version 2 was written in C++ language and algorithm contains small updates.

### 3.5.4 CMAT

CMAT is recently developed tool presented in the article [22]. This tool is based on multiple MI scores (MIp and MIc) and is fully automated. So, as an input, only reference sequence is needed. CMAT itself then searches for homologous sequences. If the whole MSA is uploaded as an input, it is not clear (from public information) if only this MSA is used or homologous sequences are searched too. One more feature is present in this tool. Positions with gaps are not eliminated, but sequences with at least one gap on two currently compared positions are ignored in the joint probability estimation.

### 3.5.5 CoMap

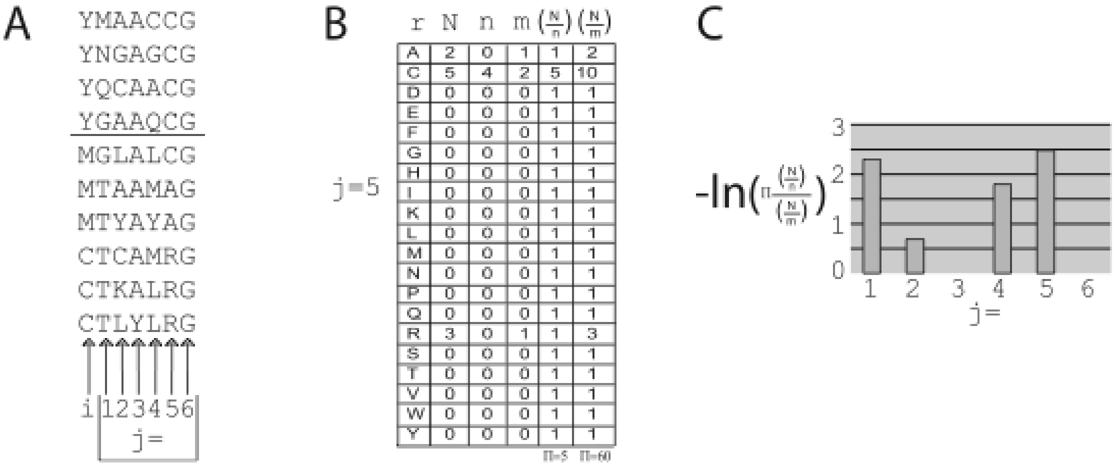
CoMap [6] is freely available for Unix, Linux, MacOS and Windows. It is written in C++ language using Bio++ libraries. Two kinds of co-evolution analyzes are provided: a pairwise analysis, presented in the article [8], and a clustering analysis in the [7]. In both cases, a parametric bootstrap approach is used to evaluate the significance of groups. CoMap's input is a sequence alignment (and optionally a phylogenetic tree). It is possible to remove conserved sites from the analysis.

#### Pairwise analysis

Pairwise analysis in CoMap provides empirical Bayesian method for the detection of coevolving positions, taking into account the uncertainty in substitution mappings, multiple substitutions, and among-site rate variation.

At first, substitution vectors are computed (see [8]). Then the amount of coevolution for a pair of sites is measured by taking the Pearson's correlation coefficient $\rho$ of the two corresponding substitution vectors. A parametric bootstrap approach is used to evaluate the null distribution of $\rho$.

This method was developed on rRNA sequences and needs to be improved to deal with protein data sets, for instance, by incorporating chemical distances and/or using mutivariate analysis. Several parameters also have to be known, namely, tree topology and branch lengths, substitution model, and rate distribution.



| r | N | n | m | $\binom{N}{n}$ | $\binom{N}{m}$ |
|---|---|---|---|---|---|
| A | 2 | 0 | 1 | 1 | 2 |
| C | 5 | 4 | 2 | 5 | 10 |
| D | 0 | 0 | 0 | 1 | 1 |
| E | 0 | 0 | 0 | 1 | 1 |
| F | 0 | 0 | 0 | 1 | 1 |
| G | 0 | 0 | 0 | 1 | 1 |
| H | 0 | 0 | 0 | 1 | 1 |
| I | 0 | 0 | 0 | 1 | 1 |
| K | 0 | 0 | 0 | 1 | 1 |
| L | 0 | 0 | 0 | 1 | 1 |
| M | 0 | 0 | 0 | 1 | 1 |
| N | 0 | 0 | 0 | 1 | 1 |
| P | 0 | 0 | 0 | 1 | 1 |
| Q | 0 | 0 | 0 | 1 | 1 |
| R | 3 | 0 | 1 | 1 | 3 |
| S | 0 | 0 | 0 | 1 | 1 |
| T | 0 | 0 | 0 | 1 | 1 |
| V | 0 | 0 | 0 | 1 | 1 |
| W | 0 | 0 | 0 | 1 | 1 |
| Y | 0 | 0 | 0 | 1 | 1 |
|  |  |  |  | $\Pi=5$ | $\Pi=60$ |

*Figure 3.4: Illustration of the ELSC detection algorithm. Position* i *is here compared to different positions* j*. There is a computation demonstration for* j=5 *in the part B. Final results are showed in the part C. Picture taken from [5].*

#### Clustering method

Next method implemented in CoMap analysis tool is a clustering method, which performs dividing analyzed CMs into groups according to their biophysical properties as Grantham physico-chemical distance, difference of volume, polarity or charge. This helps in revealing of CMs type.

### 3.5.6 CorrMut

Method implemented as web service is described in the article [13]. Phylogenetic tree is used for eliminating phylogenetic noise. Further, Miyata matrix, which provides a measure for the physicochemical differences between amino acids, is used in this tool.

As the first step in the analysis, the evolutionary history of the protein family is reconstructed by inferring the sequences of hypothetical ancestral proteins of the family using the neighbor-joining algorithm. By following the reconstructed pathway, the changes that occurred at each evolutionary step for every position are traced, which reduce the errors that arise by comparing phylogenetically distant sequences. Before the next step, gaps and positions with entropy lower than 1.1 are eliminated. The Pearson's correlation coefficients of each pair of positions are computed. Confidence intervals for the correlation coefficients of every pair not rejected in previous steps are derived using bootstrap sampling. Finally, a bootstrapping procedure is used to eliminate correlations that are statistically insignificant.

## 3.6 Detection based on likelihood models

### 3.6.1 ELSC

Different method for CMs detection was presented in the article [5]. It is called Explicit likelihood of subset covariation (ELSC) and is based on computing probability. Normalized ratio is computed by following equation:

$$\Lambda_j^{<i>} = \prod_r \frac{\binom{N_{r,j}}{n_{r,j}}}{\binom{N_{r,j}}{m_r, j}}, \tag{3.17}$$

where $r$ is residue, $N$ is number of sequences in MSA, $n$ number of sequences in the subset, $i$ and $j$ positions and $m$ is an ideally representative subset. Algorithm is illustrated on the picture 3.4.

### 3.6.2 Spidermonkey

Description of Spidermonkey is partially taken from [36] and its official web site. Spidermonkey is a component of Datamonkey suite of phylogenetic tools. Spidermonkey is publicly available both as a web application and as a stand-alone component of the phylogenetic software package HyPhy. Spidermonkey algorithm consists of these steps:

1. This tool uses neighbor-joining method for estimating phylogenetic tree from a multiple alignment, if tree is not uploaded with MSA.

2. A substitution model is fitted to these data by maximum likelihood and then the inferred ancestral sequences are used to map substitution events to branches in the tree.

3. Replicate sets of ancestral sequences can be resampled from the posterior probability distribution and analyzed in parallel.

4. Invariant sites are automatically excluded.

5. Correlated patterns of substitutions in the tree imply coevolution among sites.

6. The joint distribution of substitutions in the tree is encoded as a binary state matrix, in which each row corresponds to a unique branch and each column to a site in the alignment, and is analyzed using Bayesian graphical models (BGMs).

A Bayesian graphical model is a compact representation of the joint probability distribution of many random variables. A graph (or network) is a visual depiction of the relationship between two or more individuals, in which each individual is represented by a node. Relationships between nodes are indicated by drawing a line connecting the nodes, which is referred to as an edge. In this context, edges represent significant statistical associations between individual codon sites of an alignment, which could be caused by functional or structural interactions between the corresponding residues of the protein. Edges may be directed to indicate that one node is conditionally dependent on another; otherwise, they are undirected.

The power of BGMs lies in exploiting the Markov property (conditional independence) to isolate each node from the rest of the graph once its dependence on its neighbors is accounted for. Another advantage of using BGMs is the possibility of looking at all the variables in parallel.

Then computing the (posterior) probability of each node is made and the chain rule is used to compute the joint posterior probability of the whole graph (nodes, edges). Every graph is a hypothesis of how variables work together. It is needed to find the most likely graph. The number of possible graphs grows factorially with the number of nodes.

This is where order-MCMC (described in publication [15]) comes in. Instead of trying to pick just one graph, it is going to average over subsets of graphs, which are going to be defined by permutations in a hierarchical ranking of the nodes in the graph (node orders). With a node order, it is possible to make an assertion about which nodes can depend on other nodes. The trick is then to wander around the permutation space of node orders with a Monte Carlo Markov Chain, comparing the posterior probability of each subset of graphs defined by the current hierarchy.

## 3.7 Noise reduction method - NCPS score

The normalized coevolutionary pattern similarity (NCPS) score was presented in the article [26]. It was developed for phylogenetic noise reduction. It was assumed that the phylogenetic noise could be estimated by examining the coevolutionary relationship among residues. If the two aligned positions $i$ and $j$ have a high-CM score and they also share similar coevolutionary patterns with the other positions, then their high-CM score is likely due to the phylogenetic reasons.

Coevolutionary pattern similarity (CPS) scores between the $i$ and $j$ positions was defined as follows:

$$CPS(i,j) = \frac{1}{n-2} \sum_{k \neq i,j} CM(i,k)CM(j,k) \tag{3.18}$$

where $n$ is the number of columns in a MSA. The CPS has its maximum value when CM(i, k) and CM(j, k) are identical for all k. Since the CPS is the product of two CM scores, a normalizing factor is required. The square root of the mean of all CPS scores was used for a normalization. As a result, the normalized coevolutionary pattern similarity (NCPS) scores are defined as a measure of a background noise.

$$NCPS(i,j) = \frac{CPS(i,j)}{\sqrt{\frac{1}{n(n-1)} \sum_{i,j} CPS(i,j)}} \tag{3.19}$$

Then, corrected CM score (CMc) is computed as follows:

$$CMc(i,j) = CM(i,j) - NCPS(i,j) \tag{3.20}$$

Entropy factor is also used for noise reduction of sites with an extreme entropy, as indicated in the [39]. Entropy H of the $i$-th column of MSA is defined as follows:

$$H(i) = -\sum_{a} p(a) \log_{20} p(a) \tag{3.21}$$

where $a$ is index of an amino acid and p(a) is the amino acid probability distribution of the column. Then the entropy factor is defined:

$$E(i,j) = H(i)H(j)(1 - H(i)H(j)) \tag{3.22}$$

and the CMc score with entropy:

$$eCMc(i,j) = E(i,j)CMc(i,j) \tag{3.23}$$

Final equation defined by [26] is about best performance of this method. This best performance is when the base CMA (correlated mutation analysis) is MI and the average (aMIc) of MIc and eMIc is used.

$$aMIc(i,j) = \frac{1}{2}[\frac{MIc(i,j)}{\max MIc} + \frac{eMIc(i,j)}{\max eMic}] \tag{3.24}$$

This noise reduction was tested for following CMA types: MI (3.3), OMES (3.4.1) and McBASC (3.5.1). Results can be seen on the picture 3.5.

## 3.8 Another example of detection tool - WHAT IF

WHAT IF is a complex tool for protein's structure analyzing and modeling written in Fortran programming language. One of the available functionality is also correlated mutation analysis. As written in the article [40], WHAT IF supports correlated mutations analysis using the WALCOR module. The main idea behind correlated mutation analysis (CMA) or correlation analysis in general is that detected are residues that are conserved in sequences that perform function X, but are not conserved in the sequences that do not perform this function.

Several options exist to search for correlated behavior among residues. These options can be divided into three groups: CORMUT, CORAN1-like, and the +/- correlations. For purpose of this work, description of CORMUT category will be sufficient. The other groups differ very little.

CORMUT looks for residues that mutate in tandem. The option CORMUT requires a certain degree of variability for the residue positions. CORMUN, in contrast, does not take variability into account, and will thus call a pair of completely conserved residues highly correlated. The CORMUM option does a correlation analysis just like the basic CORMUT option, but rather than scoring binary (+1 for conserved or mutated in tandem, and 0 otherwise) CORMUM scores all pairs by the difference between the exchange matrix scores for the two positions. CORMUF does quite the same as CORMUT, but puts a penalty on missing residues.

The first version of WHAT IF program was introduced in 1990. Until now it has been improving. In late 2011 version 9 was released. WHAT IF is not distributed for free, it is a commercial software. CMs detection should be almost the same as it was introduced in 1990, because CMs detection is not the main purpose of WHAT IF.

## 3.9 Best tool for detection of correlated mutations

It is not easy to choose the best method because of these reasons:

- **There is no general comparison of methods/tools available** - some minority comparisons are available, but there is very low number of compared tools and these comparisons are made on different datasets, which makes generalization of results very difficult. The next reason is that some tools are not available to public and others are not available any longer.

- **It is very difficult to say if CM was detected correctly or wrongly** - to gain representative output of CMs detection it is needed to create it manually - it is needed to analyze protein structure and also functional sites somehow.

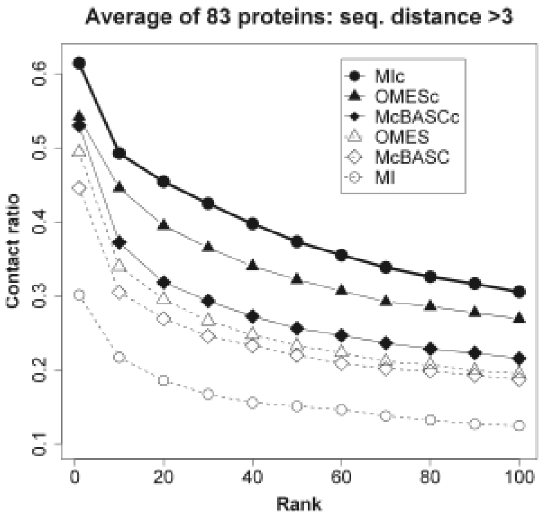


*Figure 3.5: Picture illustrates improved performance of methods after using NCPS score (methods with letter 'c' in the end of their names). Picture taken from [26].*

However, there is still possibility to gain partial comparisons. These are presented below.

The article [5] provides a comparison between two methods for CMs detection - ELSC and SCA. By metric used in this article, the ELSC algorithm has, on average, more power than the SCA algorithm as the most highly co-varying pairs of residues tend to be closer to each other for ELSC than for SCA.

Results published in the article [19] imply that RCW-MI performs better than MI/E, Dependency and even combinations of these methods.

### Fodor

Fodor package was introduced in articles [14, 5]. It consists of some modified methods introduced before. Different CM algorithms have different levels of performance with decreasing power in the order:

$McBASC > OMES > SCA > MI$

The algorithms have decreasing sensitivity to background conservation in the same order:

$McBASC > OMES > SCA > MI$

All algorithms provide almost the same results (in the case of conservation), only McBASC gives high scores to a wider range of conservation than the other algorithms (see picture 3.6).

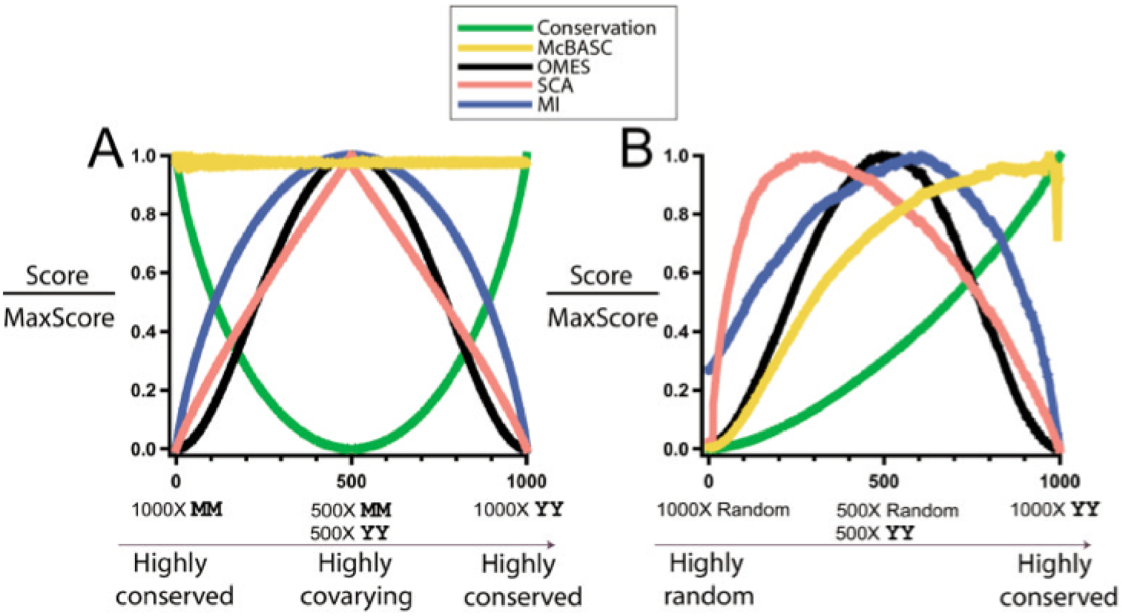


*Figure 3.6: Results from Fodor package comparison. Part A shows sensitivity to the computed conservation - result should be a horizontal lines like in McBASC (but highly conserved pairs should be excluded - they are invariant and contains no information about mutations). Part B shows process of computed values, where input pairs go from highly random to highly conserved. Picture taken from the [14].*

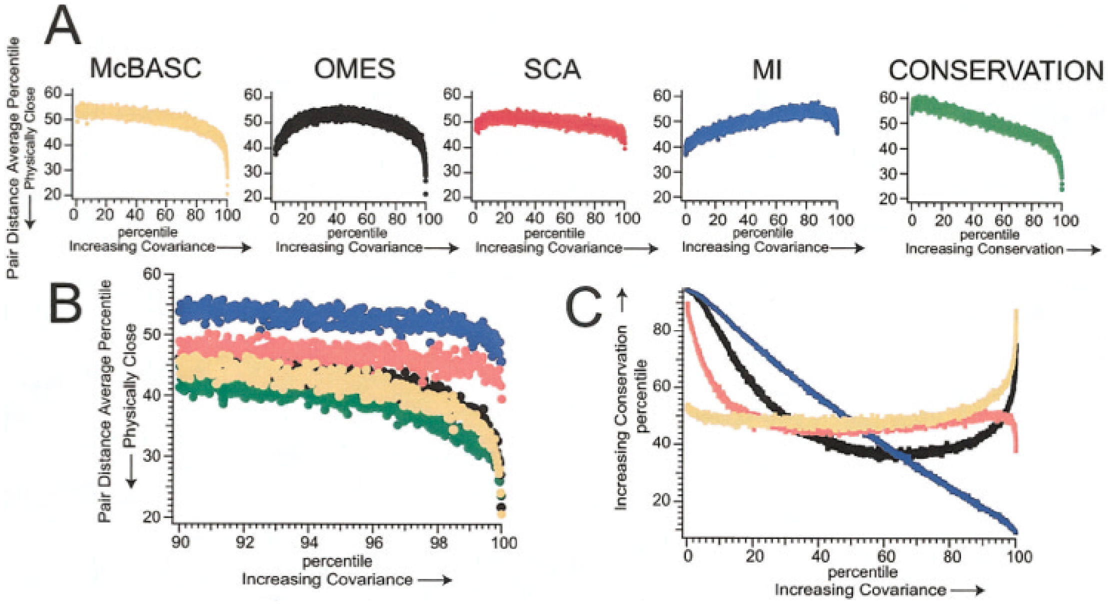


*Figure 3.7: Each panel is the histogram with each point on the x axis representing 0.025 percentile. So, for example, each of the rightmost points in (A) and (B) is the average pair distance percentile for conservation or covariance scores between the 99.975th and 100th percentile scores for each algorithm. (A) The performance of the algorithms in predicting pair distance. (B) The same data as in (A) with the x axis expanded to show only the 90th to 100th percentile. (C) The average conservation as a function of the covariance scores. Picture and description was taken from [14].*

# Chapter 4

# Algorithm development

This chapter describes development of the algorithm for detection of correlated mutations. As the first step, elimination of gapped position is solved (section 4.1). Elimination of conserved positions and positions with phylogenetic noise is described in the section 4.2. Section 4.3 describes comparing positions and revealing correlation between them. Last section of this chapter is devoted to the setting of parameters affecting the characteristics of the detection.

## 4.1 Elimination of gapped positions

| | |
|---|---|
| Input: | MSA and phylogenetic tree |
| Output: | positions with no gaps |

The first step is to eliminate gapped positions (positions with some level of gaps). Correlated mutations should be positions, which are important for protein that's why we do not care about positions with gaps. Elimination is simple in this case:

$$gaps\_score = \frac{number\_of\_gaps\_in\_MSA}{number\_of\_sequences\_in\_MSA} \tag{4.1}$$

Gaps score is then compared with a threshold which should not be too small because of possible inaccuracies in MSA creation or stochastic noise. Positions with too many gaps are filtered.

**Definition 4.1.** *Gapped position is a position in MSA where percentage of gaps is above the selected threshold.*

## 4.2 Elimination of evolutionary conserved positions

| | |
|---|---|
| Input: | positions with no gaps |
| Output: | filtered positions with phylogenetic tree with inferred predecessors |

There are two reasons for eliminating conserved positions (positions which do not mutate at all or mutate very rare). At first these positions would be correlated with all others conserved positions and this would make too many false results. The next reason is that these positions carry almost zero information value.

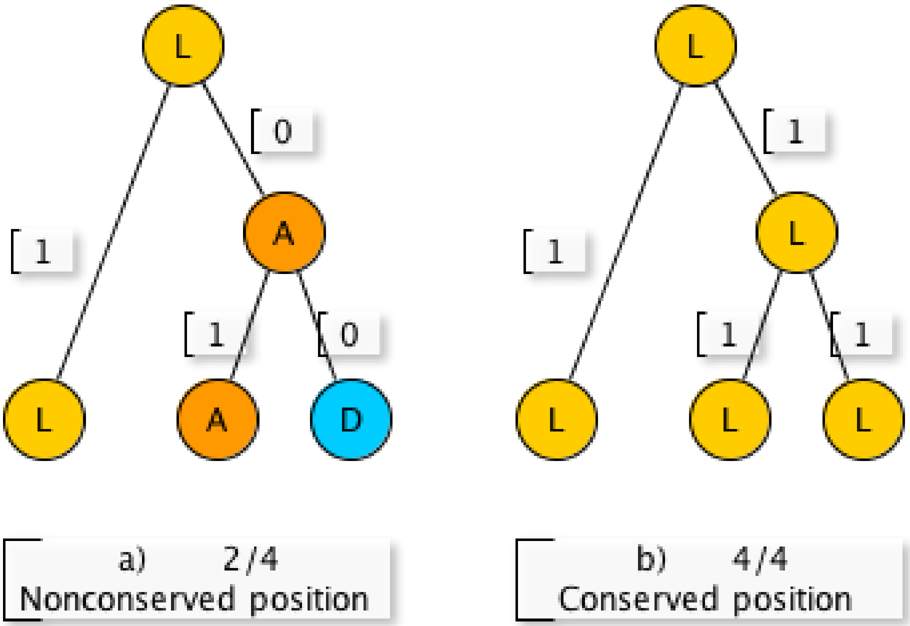


*Figure 4.1: Demonstration of computing the conservation score.*

**Definition 4.2.** *Usual conserved position is position which contains rare mutations. Subtree with root in this mutation in phylogenetic tree is usually very small. For our purpose, number of nodes in this subtree is less than 10% of total number of nodes in the phylogenetic tree.*

Detection of conserved positions can be solved with existing algorithms, but our algorithm uses different one based on a phylogenetic tree. At first, it is needed to load a phylogenetic tree and map amino acids from MSA for current position to corresponding phylogenetic end node.

The next step is to infer predecessors using Sankoff algorithm [31]. This algorithm solves small parsimony problem - finding the best assignment of internal nodes for the known phylogenetic tree. Algorithm consists of two phases:

1. Bottom-up phase - determine cost of assignments for all subtrees for all possible states.
2. Top-down phase - pick optimal states for each internal node.

McLachlan substitution matrix is used in Sankoff algorithm for the best results. If phylogenetic tree includes information about evolutionary distance, this information is used for better accuracy of an algorithm. Usual Sankoff algorithm was modified for usage of a phylogenetic distance information usually present in NHX files. If this distance is present, value is incremented by 1. If it is not present, default value of this distance is 1. Default or incremented distance is used as divisor to computed values in inner nodes. This reduces influence of distant nodes.

At this point, we have a phylogenetic tree with amino acids for current position in MSA (like on the picture 4.1) and we can study how position has been changed during the evolution. Conservation can be determined using number of mutations in this phylogenetic tree

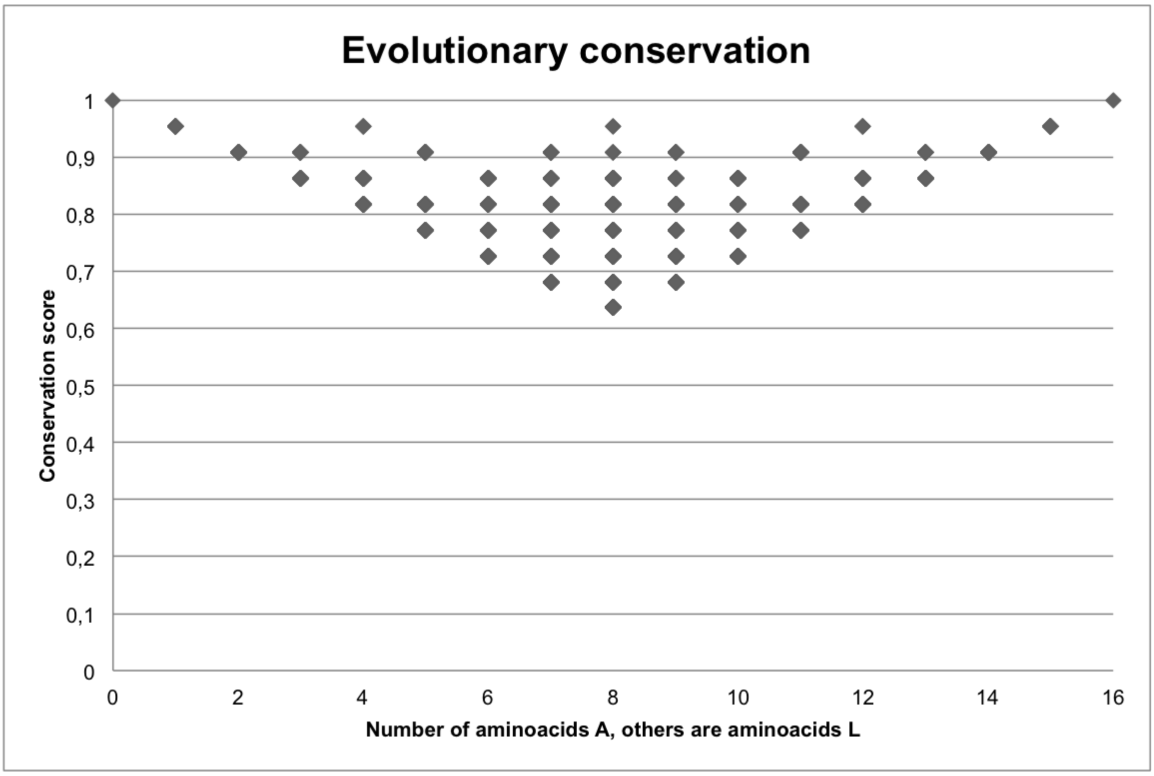


*Figure 4.2: Graph shows that the conservation score depends on positions of mutations. Input files were MSA with 16 sequences and regular (phylogenetic) tree. Sequences were made from only 2 different amino acids and all possibilities were generated in MSA. On the left side of the graph all amino acids were A, in the middle of the graph there were 8 times A and 8 times L, and on the right side of the graph all amino acids were L.*

(parent and child do not have the same amino acid). Conservation is then computed as follows:

$$conservation\_score = \frac{number\_of\_no\_changes\_in\_a\_tree}{number\_of\_edges\_in\_a\_tree} \tag{4.2}$$

Now we have a conservation score (0-1 or 0%-100%) on which we can apply a threshold and filter conserved positions.

Traditional algorithms use counting of amino acids and do not use evolutionary tree. In this algorithm it depends on position where different amino acids are located (as shown on graphs 4.2 and 4.3). Advantage of this feature is explained in the next section.

When complex protein family is examined, there can be a subfamily which is relatively different. In this case a phylogenetic noise can occur (described in section 2.2.4). This phylogenetic noise should be already removed during the elimination of conserved positions described previously. All depends on selected threshold of a conservation. As mentioned before, the conservation score depends, in which sequences different amino acids are. When they are in the same subfamily, the conservation ratio is higher (only one mutation in phylogenetic tree may occur) and this position can be eliminated as conserved. Therefore previously described detection of conserved position detects also phylogenetic noise.

Expression evolutionary conservation was defined in this work for differentiation from usual conservation computed in MSA only (section 2.2.4).

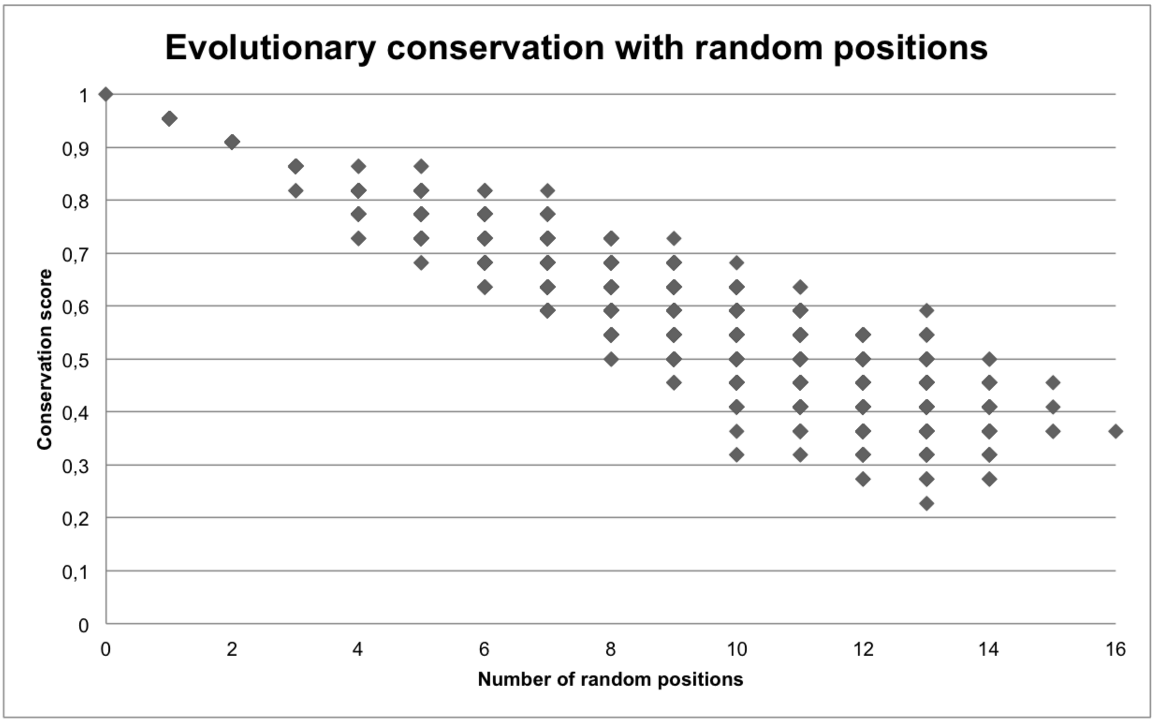


*Figure 4.3: Graph shows that the conservation score depends on positions of mutations. Input files were MSA with 16 sequences and regular (phylogenetic) tree. Sequences were made from all amino acids and all possibilities were generated in MSA. On the left side of the graph all amino acids were A, in the middle of the graph there were 8 times A and 8 times random amino acid, and on the right side of the graph all amino acids were random.*

**Definition 4.3.** *Phylogenetic noise is indicated by many identical amino acids in MSA and all of these amino acids are in the same subtree in the phylogenetic tree. This subtree contains minimum mutations and is large. For our purpose, number of nodes in this subtree is more than 10% of total number of nodes in the phylogenetic tree.*

**Definition 4.4.** *Evolutionary conservation is defined on the whole tree and expresses how much information is carried by this position (how many mutations occurred during the evolution). Usual conserved positions have also high evolutionary conservation score. Detected phylogenetic noise increases the evolutionary conservation score (despite of an usual correlation score).*

In fact, mutation affecting the whole subtree is important for a correlated mutations detection but that single mutation indicates a phylogenetic noise (see picture 4.4). In the case of more those mutations affecting more subtrees we talk about specificity, which is exactly what is needed in the detection of correlated mutations.

## 4.3 Detection of correlated pairs

Input: filtered positions with phylogenetic tree with inferred predecessors
Output: detected correlated pairs of positions

At this point, inappropriate positions were eliminated. Now we will compare all positions with each other. Phylogenetic trees of two positions will be compared and for this comparison an additional information has to be determined. During the elimination of conserved positions, information about mutation between parent node and child node was gained. Now we need to extend this information about physicochemical properties of this mutation. We are interested in change of hydropathy property, charge and polarity. Overview of these physicochemical properties can be found in the appendix. Now we have all information needed for comparison. It is needed to compare these phylogenetic trees (PTs) and compute correlated score as follows:

$$corr\_score = \frac{\sum_{edge \in edges\_in\_PT} min(1, max(0, 1 - penalization)}{number\_of\_edges\_in\_PT} \tag{4.3}$$

where *penalization* is a variable with values from table in Appendix F, where rows are mutation types of the first compared position on selected edge and columns are mutation types of second compared position on selected edge in the phylogenetic tree. Mutation types are none, polarity, big charge, small charge, polarity + small charge, hydropathy, polarity + hydropathy, small charge + polarity + hydropathy, and insertion/deletion.

**Definition 4.5.** *Correlation value represents percentage similarity of mutation types in two phylogenetic trees (two positions in MSA). Two positions are correlated if their correlation value is higher than the selected threshold.*

### 4.3.1 Modification of the correlation score computation

This modification uses the same equation 4.3, but not all edges are used in computation. Only edges with some mutation on compared edges are used. This reduces influence of conserved positions on correlation score and ensures better distribution on output interval.

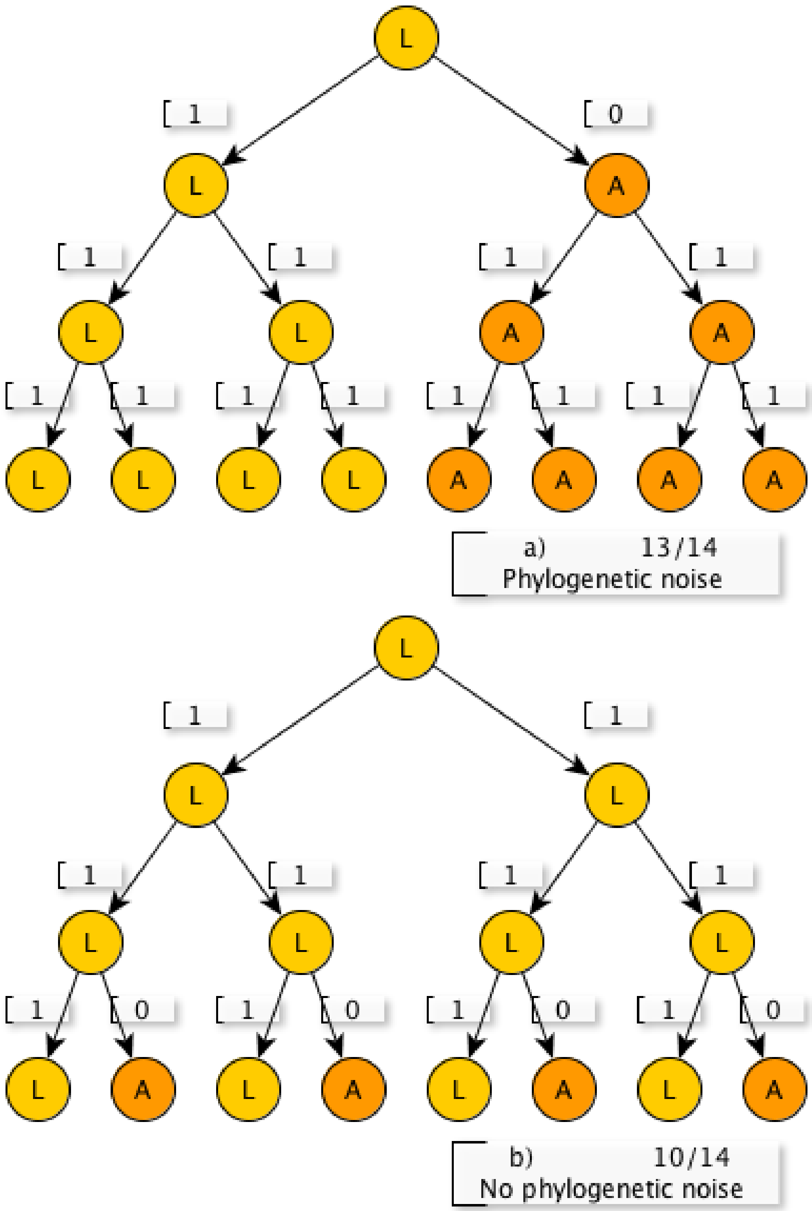


*Figure 4.4: Demonstration of a phylogenetic noise.*

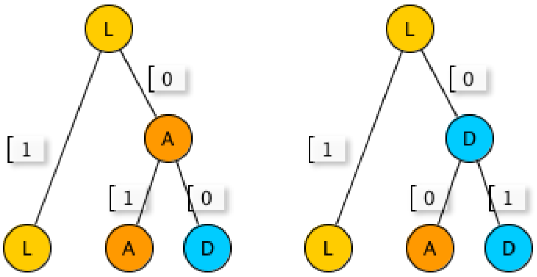


*Figure 4.5: This picture demonstrates problem with differently inferred predecessors. Edges then have different values and their comparison is more difficult. This is demonstration only, with the same substitution matrix predecessors will be the same. Problem could occur in the case of different amino acids with different values in substitution matrix.*

### 4.3.2 Sibling's problem

Previously described algorithm can suffer from incorrectly inferred predecessors as shown on the picture 4.5. This problem can be alleviated by following modification: when edges are compared (equation 4.3), sibling's edge is also taken into account and is determined if it is better to swap values on these edges. In other words, penalizations for all possibilities are computed and the best possibility with minimal penalization for all sibling's edges is chosen (picture D.2).

## 4.4 Setting of parameters

### 4.4.1 Choosing appropriate substitution matrix

Sankoff small parsimony problem is controlled by an appropriate substitution matrix. Candidates are BLOSUM, PAM, or McLachlan substitution matrix. BLOSUM and PAM are based on probability of mutation and number of mutated nucleotides needed to change amino acid. McLachlan is, on the other hand, based on amount of physicochemical change between two amino acids. Usually, BLOSUM would be a better choice but in the case of correlated mutations, probability substitution matrix produced worse assignments than McLachlan substitution matrix. That is why McLachlan substitution matrix was chosen.

There were made some transformations and changes for the purpose of this work in selected substitution matrix. Original values were from discrete interval $< 0, 9 >$, where 0 is the least probable mutation and 9 is the most probable mutation. Values needed in the Sankoff algorithm are opposite. That is why original values $x$ were transformed to $y$ using the equation $y = 9 - x$. Then cases with no mutation (the same amino acids before and after) were rated by the value 1 (some of them were 1 and some of them 0, which caused inaccurate assignments in extensive phylogenetic trees). Last modification was adding values for cases of gaps and unknown amino acids. Unknown amino acid should not be in an internal node of a phylogenetic tree (only if all children are unknown amino acids).

### 4.4.2 Penalization values

Penalization values were created using the McLachlan substitution matrix discussed in the previous section. The first step was to analyze this matrix by computing averages of its values in all possible physicochemical changes. This set of average values $V$ was transformed to values $x$ in the interval $< 0, 1 >$ using

$$x = \frac{|y - z|}{\max(V) - \min(V)} \tag{4.4}$$

where $y$ is type of mutation in processed row and $z$ is type of mutation in processed column in final matrix of penalization values.

### 4.4.3 Finding of the automatic threshold of a correlation score

Every MSA used as input can have different characteristics - more or less conserved, evolutionary distant sequences and so on. This makes computed scores of correlation also different. For this purpose, an automatic threshold for correlation score was created to select only $\frac{1}{x}$ best results.

$$auto\_threshold = max\_correlation\_score - \frac{(max\_correlation\_score - min\_correlation\_score)}{x} \tag{4.5}$$

Default value was experimentally set to $x = 8$.

### 4.4.4 User-defined parameters

The algorithm contains three parameters which user can set.

#### Evolutionary conservation threshold

The first parameter is the evolutionary conservation threshold - value, which decides if position in a MSA is conserved or is affected by a phylogenetic noise a lot. Evolutionary conservation is computed in the whole phylogenetic tree as defined in the section 4.4.3. Value can be from 0 to 1 (from 0% to 100%). Positions with scores above the selected threshold are filtered. Default value is 0.995. This means that there needs to be at least 1 mutation for every 200 edges in phylogenetic tree (not 1 mutation for 200 amino acids in the MSA).

#### Correlation threshold

Next user-defined parameter is the correlation threshold, which filters results according to their correlation score. Correlated positions with the correlation score lower than the correlation threshold or the automatic correlation threshold (section 4.4.3) are eliminated. The correlation score is also computed from edges in a tree and can have values 0-1 (0%-100%). Default value is set to 0.9. This threshold is useful for defining the value, a user is not willing to go below.

#### Threshold for eliminating gapped positions

Last parameter is a threshold for number of gaps. How much is the position gapped is computed only from MSA and expressed simply in percent. Positions with computed values above this parameter are filtered. Default value is 0.25 (25%). So 25% of sequences can have gap on this position at maximum.

## 4.5 Algorithm complexity

Time and space complexity of the developed algorithm depend on length of sequences $m$ and number of sequences $n$. Time complexity is then $T(m, n) \in O(m^2n)$ and space complexity $S(m, n) \in O(mn)$.

Both, space and time complexity, depend linearly on number of sequences. The more sequences, the more extensive phylogenetic tree is. Number of nodes in usual binary phylogenetic tree is $2n - 1$ which means linear dependence on space complexity and time complexity (it is needed to compare all of these nodes).

Space complexity also depends on length of sequences linearly. For each position, it is needed one phylogenetic tree with inferred amino acids. Situation is more interesting in the case of time complexity. For each added position in the MSA, it is needed to compare with each other, so total amount of comparisons is $\frac{m^2-m}{2}$ in the worst case (when no positions were filtered).

# Chapter 5

# Tests and results

Six random protein families from Pfam database http://pfam.sanger.ac.uk/ - PF00198 (2-oxoacid dehydrogenases acyltransferase), PF00696 (Amino acid kinase family), PF01094 (Receptor family ligand binding region), PF02737 (3-hydroxyacyl-CoA dehydrogenase), PF02826 (D-isomer specific 2-hydroxyacid dehydrogenase), and PF3466 (LysR substrate binding domain) were used (table 5.1). Whole MSA and PT were used in the case of PF00198 and PF02737, because of small number of sequences in seed. Seed (partial MSA and PT) was used in other cases. Both, phylogenetic tree and MSA are usually automatically generated and therefore highly gapped.

Basic characteristics of selected protein families are in the table below:

Corresponding PDB protein models were chosen according to the best similarity with reference sequence using searching on PDB web portal available on http://www.rcsb.org/. Selected PDB file was used for gaining distance map and contact list (distance between $C_\alpha$ is 8 Å at maximum). For this purpose, Java tool CMView 1.1.1 available on http://www.bioinformatics.org/cmview/ was used. Last step was remapping generated contacts from numbering of the PDB file to numbering of the reference sequence. For this purpose, online tool Protein BLAST (http://blast.ncbi.nlm.nih.gov) for aligning of two sequences was used.

## 5.1 Different penalization matrices

Detection depends on values in a penalization matrix, that is why different matrices were tested. Default penalization matrix derived from McLachlan substitution matrix is labeled as version 1. 4 more versions were made according to an importance of physicochemical properties and strictness of different values. All versions can be found in the Appendix.

| Protein family | reference sequence | PDB ID | # of sequences | length of MSA |
|---|---|---|---|---|
| PF00198 | C4S5L2_YERBE/174-404 | 1C4T | 6728 | 540 |
| PF00696 (seed) | A8M9I2_CALMQ/6-275 | 2HMF | 136 | 329 |
| PF01094 (seed) | Q98NL7_RHILO/48-345 | 1USG | 119 | 662 |
| PF02737 | Q3ACK8_CARHZ/3-182 | 3MOG | 5652 | 816 |
| PF02826 (seed) | A9CGR6_AGRT5/108-282 | 3BA1 | 163 | 260 |
| PF03466 (seed) | Q98A96_RHILO/94-300 | 2QL3 | 415 | 309 |

*Table 5.1: Test set used in this work.*

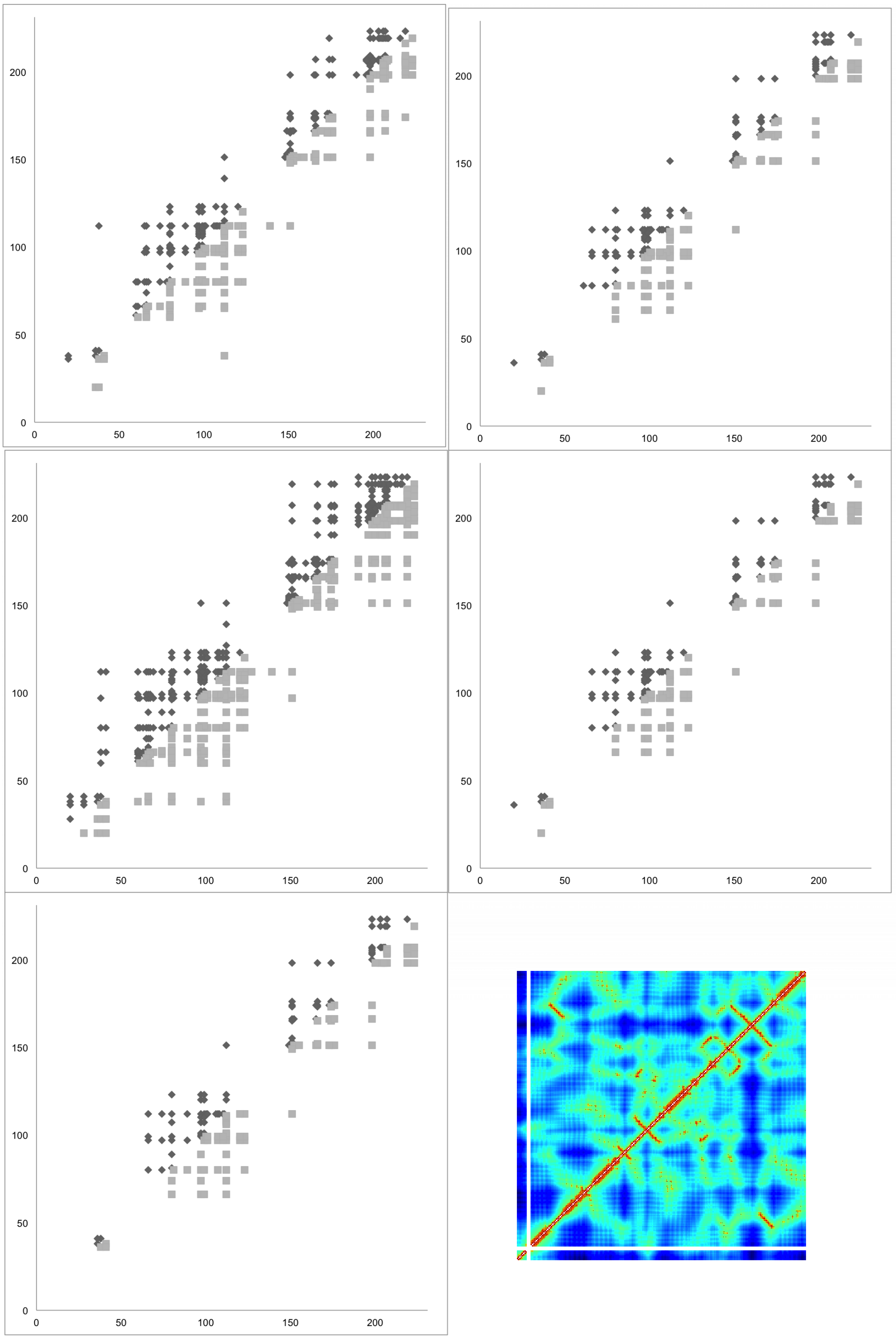


*Figure 5.1: Test performed with PF00198 for all versions of penalizations. Last picture is distance map of protein with PDB identificator 1C4T.*

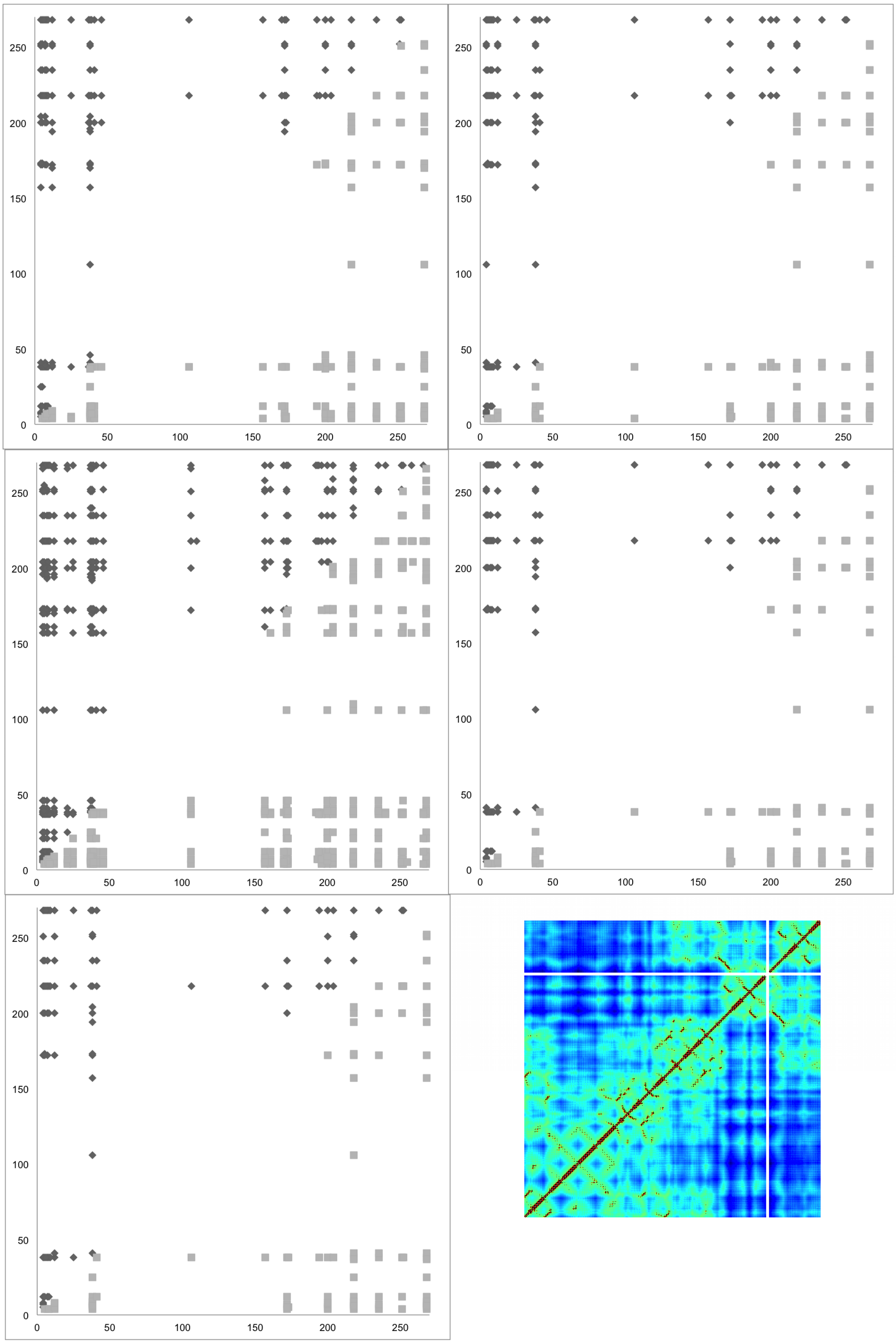


*Figure 5.2: Test performed with PF00696 (seed) for all versions of penalizations. Last picture is distance map of protein with PDB identificator 2HMF.*

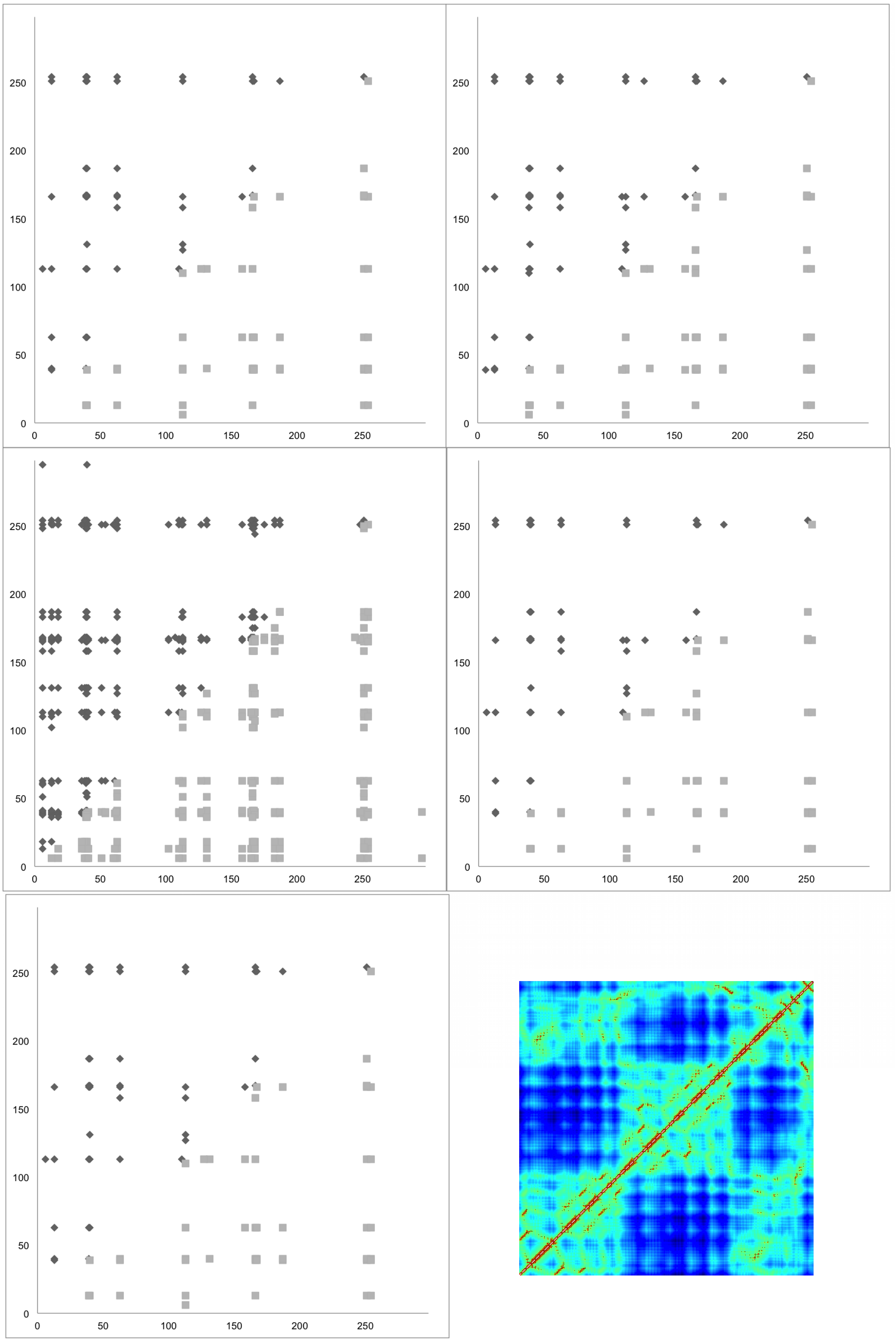


*Figure 5.3: Test performed with PF01094 (seed) for all versions of penalizations. Last picture is distance map of protein with PDB identificator 1USG.*

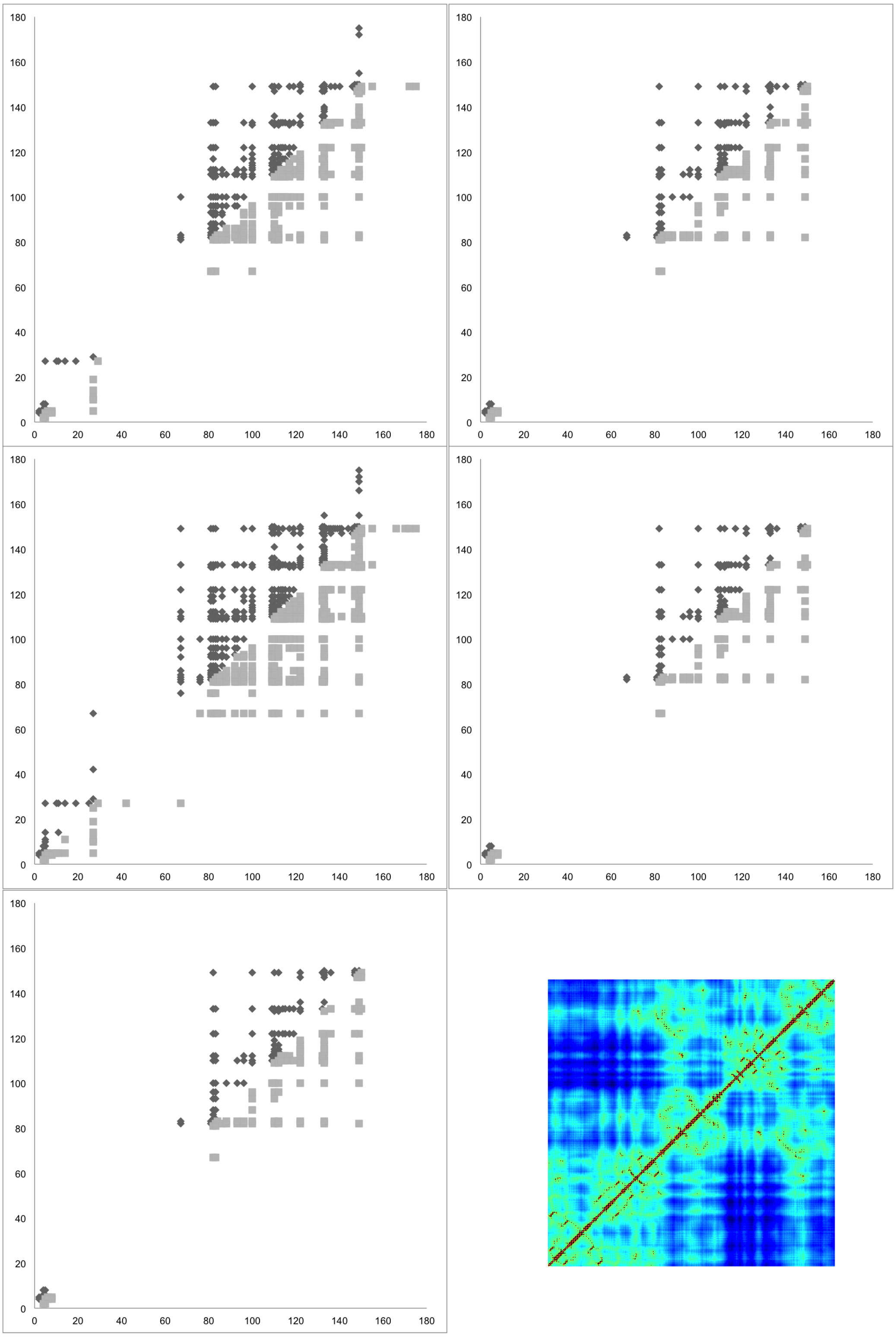


*Figure 5.4: Test performed with PF002737 for all versions of penalizations. Last picture is distance map of protein with PDB identificator 3MOG.*

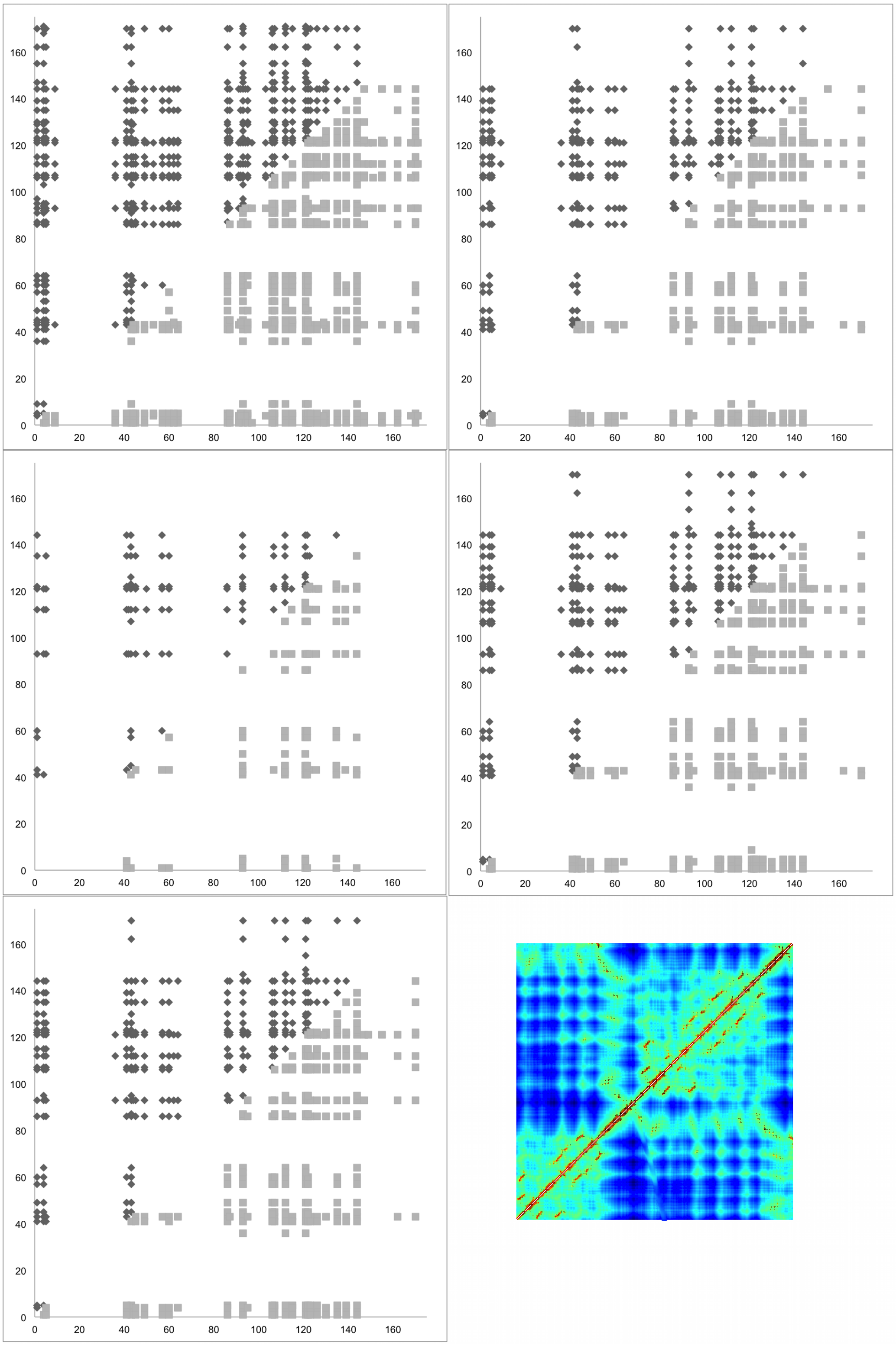


*Figure 5.5: Test performed with PF002826 (seed) for all versions of penalizations. Last picture is distance map of protein with PDB identificator 3BA1. Graph with version 3 of penalization values shows only the best 100 CMs.*

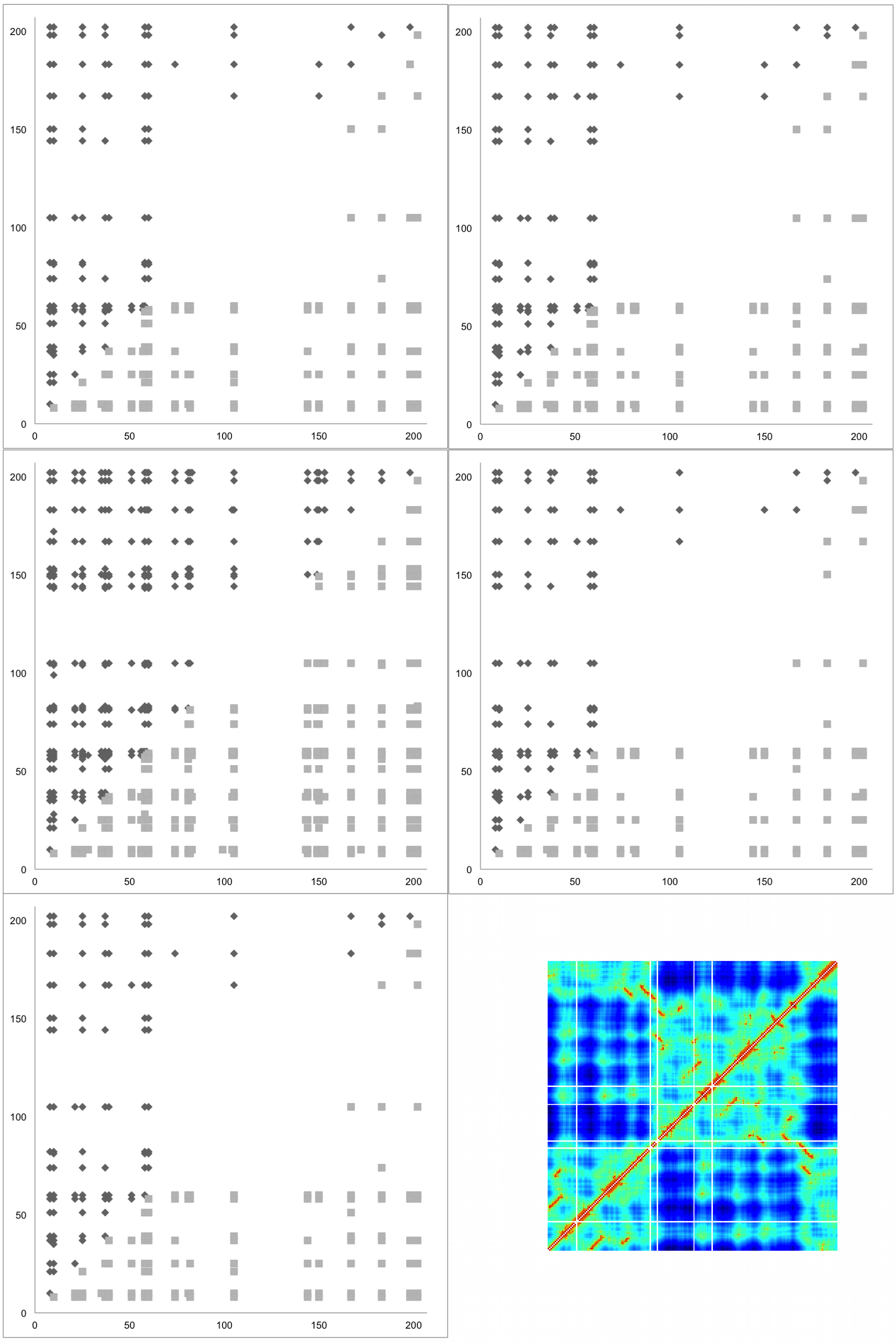


*Figure 5.6: Test performed with PF03466 (seed) for all versions of penalizations. Last picture is distance map of protein with PDB identificator 2QL3.*

| Protein family | Version 1 | Version 2 | Version 3 | Version 4 | Version 5 |
|---|---|---|---|---|---|
| PF00198 | 38 (23.2%) | 25 (24.0%) | 58 (22.5%) | 22 (26.8%) | 22 (26.8%) |
| PF00696 (seed) | 16 (11.9%) | 11 (10.9%) | 39 (13.3%) | 11 (11.7%) | 11 (12.4%) |
| PF01094 (seed) | 4 (8.7%) | 5 (9.6%) | 22 (9.6%) | 5 (10.4%) | 4 (8.7%) |
| PF02737 | 35 (20.2%) | 23 (22.8%) | 54 (20.0%) | 22 (23.2%) | 22 (24.0%) |
| PF02826 (seed) | 50 (10.4%) | 37 (11.8%) | 83 (10.2%) | 34 (12.6%) | 34 (14.8%) |
| PF03466 (seed) | 14 (10.4%) | 12 (10.8%) | 21 (8.3%) | 12 (11.3%) | 11 (10.8%) |

*Table 5.2: Table shows results from comparing with all versions of penalization matrix. Number of detected pairs which are also contacts are used for this comparison.*

| Protein family | # of detected CMs | CMs (CMAT) | CMs (CAPS) | CMs (CRASP) |
|---|---|---|---|---|
| PF00198 | 164 | 100 | x | x |
| PF00696 (seed) | 135 | 100 | 1804 | 595 |
| PF01094 (seed) | 46 | 62 | 1337 | 1955 |
| PF02737 | 173 | 100 | x | x |
| PF02826 (seed) | 483 | 19 | 284 | 674 |
| PF03466 (seed) | 134 | 52 | 199 | 802 |

*Table 5.3: Number of detected correlated mutations using each tool with default parameters.*

Results from detection of all versions on PF00198 are shown on the picture 5.1, PF00696 (seed) on the picture 5.2, PF01094 (seed) on the picture 5.3, PF02737 on the picture 5.4, PF02826 (seed) on the picture 5.5, and PF03466 (seed) on the picture 5.6. All these pictures are assembled with the same way. The first picture in the first line represents the first version of penalization matrix. The second picture in the first line represents the second version, the first graph in the second line represents the third version, the second graph in the second line represents the fourth version and the first picture on the last line represents the fifth version. Last picture is a distance map for the most similar protein structure.

Detected correlated contacts for all versions of a penalization matrix compared to all detected CMs (in percentage) are available in the table 5.2.

## 5.2 Comparison with other tools

Basic comparison with other tools (CMAT (section 3.5.4), CAPS (section 3.5.3) and CRASP (section 3.5.2)) was made. Results from CAPS and CRASP online tool are not available for protein families PF00198 and PF02737 due to performance limitations of the available CAPS and CRASP server. Default parameters were used in all cases. Number of detected correlated mutations using each tool can be found in the table 5.3. R is the improvement over random predictor (equation 2.2).

Graphs from analysis of PF00198 are shown on the picture 5.7. Detected positions (not pairs) are compared on the picture 5.13.

Graphs from analysis of PF01094 (seed) are shown on the picture 5.9.

Graphs from analysis of PF00696 (seed) are shown on the picture 5.8. Detected positions are compared on the picture 5.14.

Graphs from analysis of PF02737 are shown on the picture 5.10.

Graphs from analysis of PF02826 (seed) are shown on the picture 5.11.

Graphs from analysis of PF03466 (seed) are shown on the picture 5.12.

| | | | |
|---|---|---|---|
| **PF00198** | | | |
| CMAT | 2 (2%, R=3.240) | | |
| | developed | | |
| **PF00696** | | | |
| CMAT | 2 (2.0%, R=5.380) | | |
| CAPS | 7 (5.2%, R=1.044) | 6 (6.0%, R=2.387) | |
| CRASP | 16 (11.9%, R=7.234) | 17 (17.0%, R=10.376) | 35 (5.9%, R=1.184) |
| | developed | CMAT | CAPS |
| **PF01094** | | | |
| CMAT | 0 | | |
| CAPS | 6 (13.0%, R=4.318) | 39 (62.9%, R=20.821) | |
| CRASP | 7 (15.2%, R=3.445) | 3 (4.8%, R=1.095) | 50 (3.9%, R=1.238) |
| | developed | CMAT | CAPS |
| **PF02737** | | | |
| CMAT | 3 (3%, R=2.794) | | |
| | developed | | |
| **PF02826** | | | |
| CMAT | 2 (10.5%, R=3.318) | | |
| CAPS | 40 (14.1%, R=4.440) | 3 (15.8%, R=8.465) | |
| CRASP | 46 (9.5%, R=2.151) | 11 (57.9%, R=13.078) | 39 (19.2%, R=3.102) |
| | developed | CMAT | CAPS |
| **PF03466** | | | |
| CMAT | 5 (9.6%, R=15.299) | | |
| CAPS | 2 (1.5%, R=1.600) | 4 (7.7%, R=8.242) | |
| CRASP | 10 (7.5%, R=1.984) | 8 (15.4%, R=4.090) | 13 (7.4%, R=1.737) |
| | developed | CMAT | CAPS |

*Table 5.4: Exact similarity of detected correlated pairs. R is an improvement over random predictor (equation 2.2).*

Exact similarity of detected pairs is expressed in the table 5.4. Due to low similarity of detected pairs of two different tools, tolerancy in comparison of pairs was applied. Pairs are considered to be same if at least one position is identical and the second is less than 8Å. Results are in the table 5.5.

## 5.3 Comparison with 3D model of protein molecule

In this test, contacts gained from PDB files were used to compare detected correlated mutations. Contacts between two adjoining amino acids on protein's backbone were ignored (permanent bonds). Functional and intermolecular contacts are not involved in this test, so this test cannot be considered to be a single test for a complex tools comparison. As was written in the Theoretical part, in the article [24] was pointed out that only 16.4% of correlated pairs are contacts. From this comparison, CMAT proved the best performance in contact prediction (in percentage), but it is not clear (from description available to the public) if it uses only sequences from MSA. Result of this test is available in the table 5.6. Percentage values are equivalent to A (equation 2.1). R is the improvement over

| **PF00198** | | |
|---|---|---|
| developed | | 31 (18.9%) |
| CMAT | 23 (23.0%) | |
| | developed | CMAT |

| **PF00696** | | | | |
|---|---|---|---|---|
| developed | | 16 (11.9%) | 69 (51.1%) | 109 (80.7%) |
| CMAT | 9 (9.0%) | | 58 (58.0%) | 47 (47.0%) |
| CAPS | 54 (5.9%) | 81 (8.9%) | | 162 (17.7%) |
| CRASP | 137 (23.0%) | 125 (21.0%) | 221 (37.1%) | |
| | developed | CMAT | CAPS | CRASP |

| **PF01094** | | | | |
|---|---|---|---|---|
| developed | | 1 (2.2%) | 26 (56.5%) | 37 (80.4%) |
| CMAT | 1 (1.6%) | | 50 (80.6%) | 14 (22.6%) |
| CAPS | 44 (3.3%) | 124 (9.3%) | | 404 (30.2%) |
| CRASP | 72 (3.7%) | 47 (2.4%) | 583 (29.8%) | |
| | developed | CMAT | CAPS | CRASP |

| **PF02737** | | |
|---|---|---|
| developed | | 55 (31.8%) |
| CMAT | 29 (29.0%) | |
| | developed | CMAT |

| **PF02826** | | | | |
|---|---|---|---|---|
| developed | | 22 (4.6%) | 253 (52.4%) | 385 (79.7%) |
| CMAT | 6 (31.6%) | | 7 (36.8%) | 14 (73.7%) |
| CAPS | 106 (52.2%) | 14 (6.9%) | | 126 (62.1%) |
| CRASP | 279 (41.4%) | 48 (7.1%) | 202 (30.0%) | |
| | developed | CMAT | CAPS | CRASP |

| **PF03466** | | | | |
|---|---|---|---|---|
| developed | | 38 (28.4%) | 13 (9.7%) | 62 (46.3%) |
| CMAT | 15 (28.8%) | | 18 (34.6%) | 26 (50.0%) |
| CAPS | 6 (3.4%) | 18 (10.2%) | | 47 (26.7%) |
| CRASP | 65 (8.1%) | 75 (9.4%) | 62 (7.7%) | |
| | developed | CMAT | CAPS | CRASP |

*Table 5.5: Tolerant similarity of detected correlated pairs.*

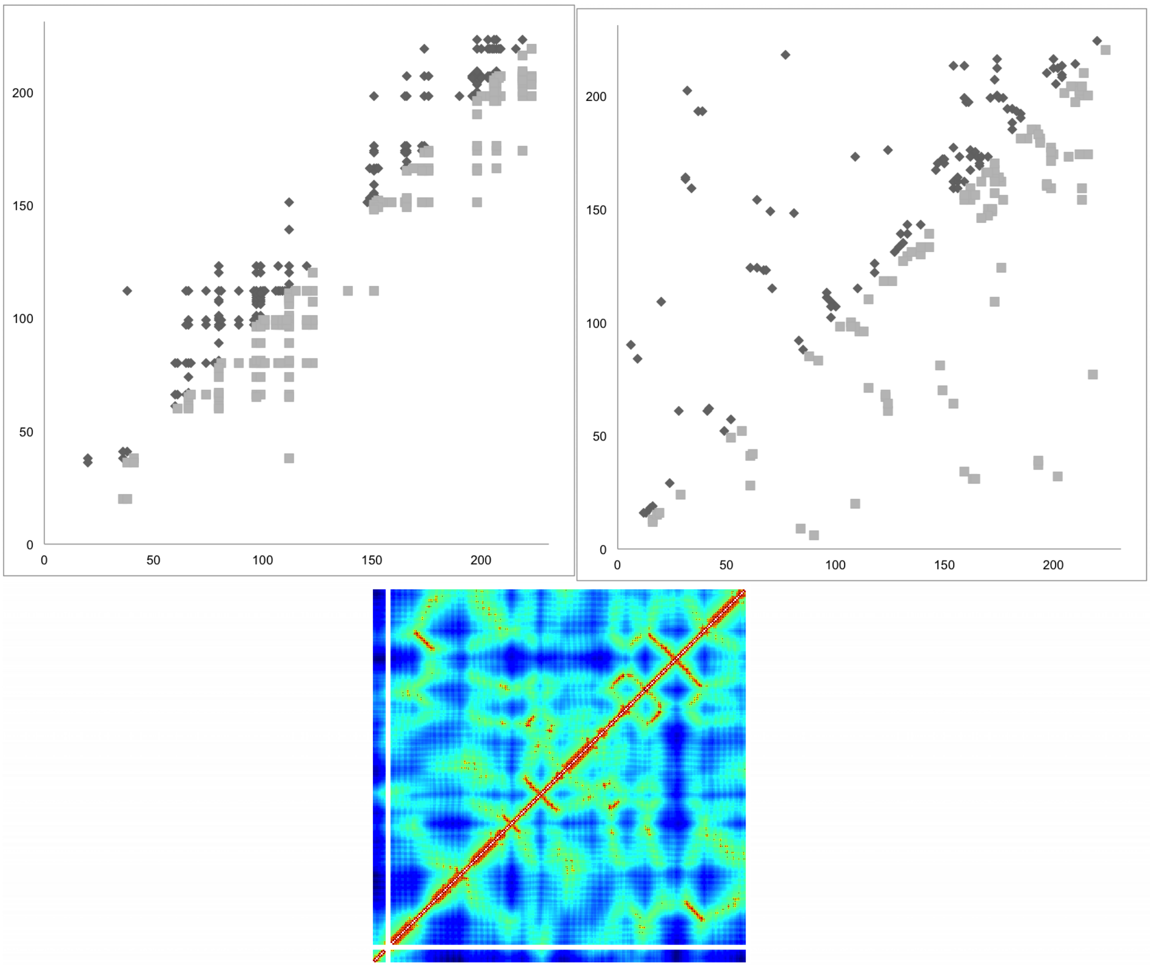


*Figure 5.7: Test performed with PF00198. The first picture in the first line shows results of developed algorithm, the second picture in the first line shows detected CMs by CMAT. All result graphs are linked to reference sequence C4S5L2_YERBE/174-404. In the second line, there is the distance map for a protein with PDB identificator 1C4T.*

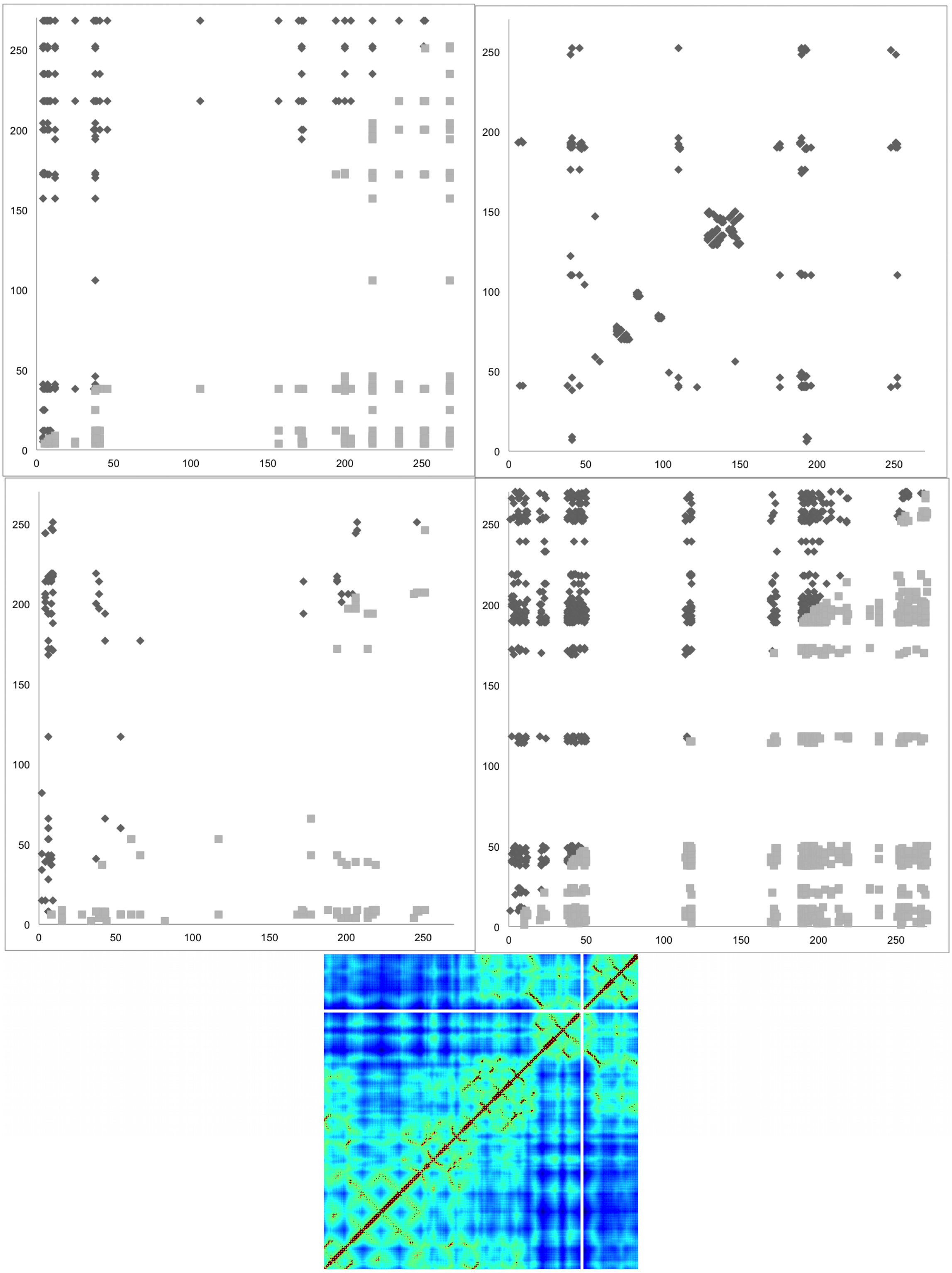


*Figure 5.8: Test performed with PF00696 (seed). The first picture in the first line shows results of developed algorithm, the second picture in the first line shows detected CMs by CMAT, the first picture in the second line shows the best 100 detected CMs by CAPS, and the second picture in the second line shows detected CMs by CRASP. All result graphs are linked to reference sequence A8M9I2_CALMQ/6-275. In the third line, there is the distance map for a protein with PDB identificator 2HMF.*

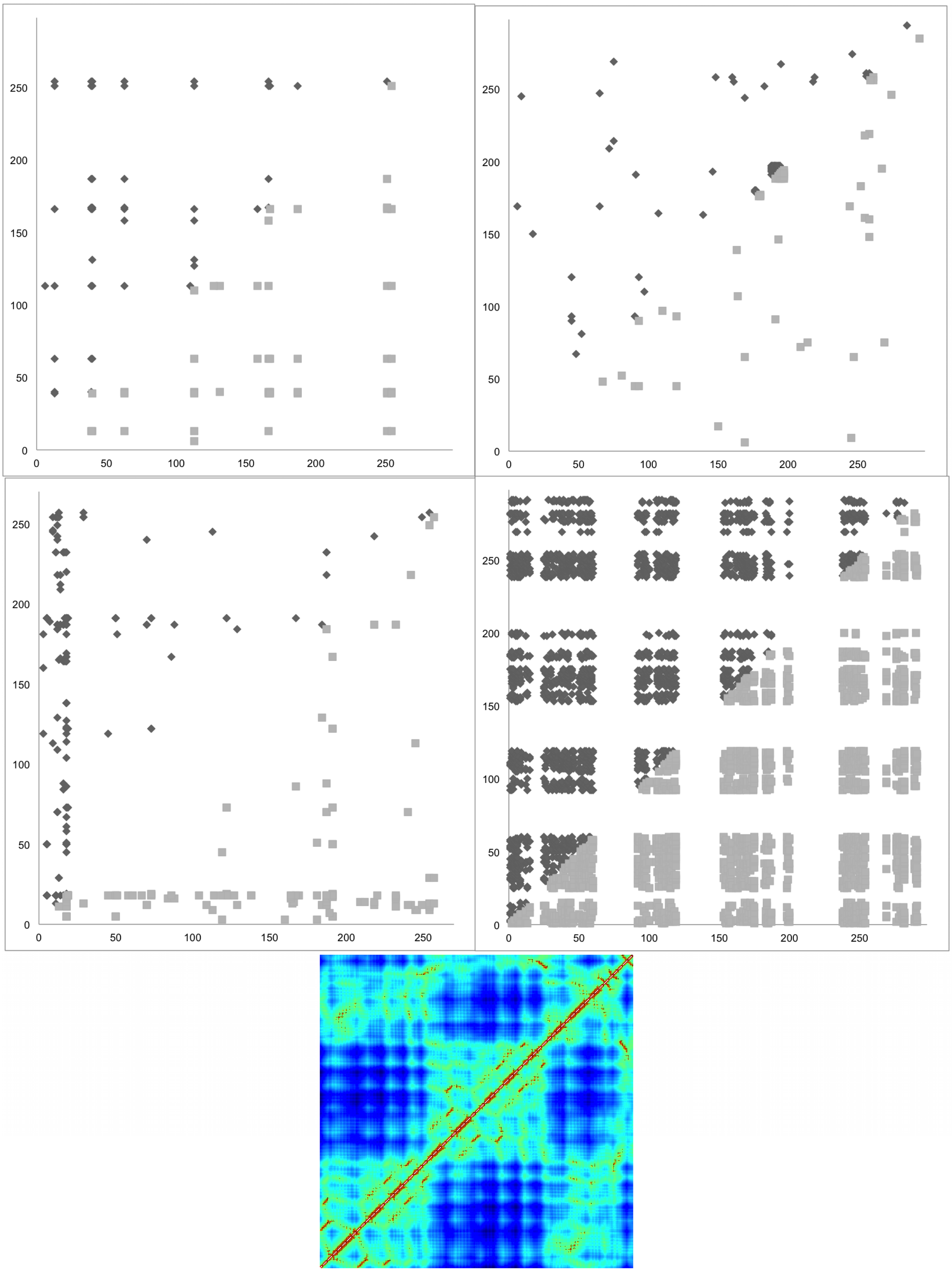


*Figure 5.9: Test performed with PF01094 (seed). The first picture in the first line shows results of developed algorithm, the second picture in the first line shows detected CMs by CMAT, the first picture in the second line shows the best 200 detected CMs by CAPS, and the second picture in the second line shows detected CMs by CRASP. All result graphs are linked to reference sequence Q98NL7_RHILO/48-345. In the second line, there is the distance map for a protein with PDB identificator 1USG.*

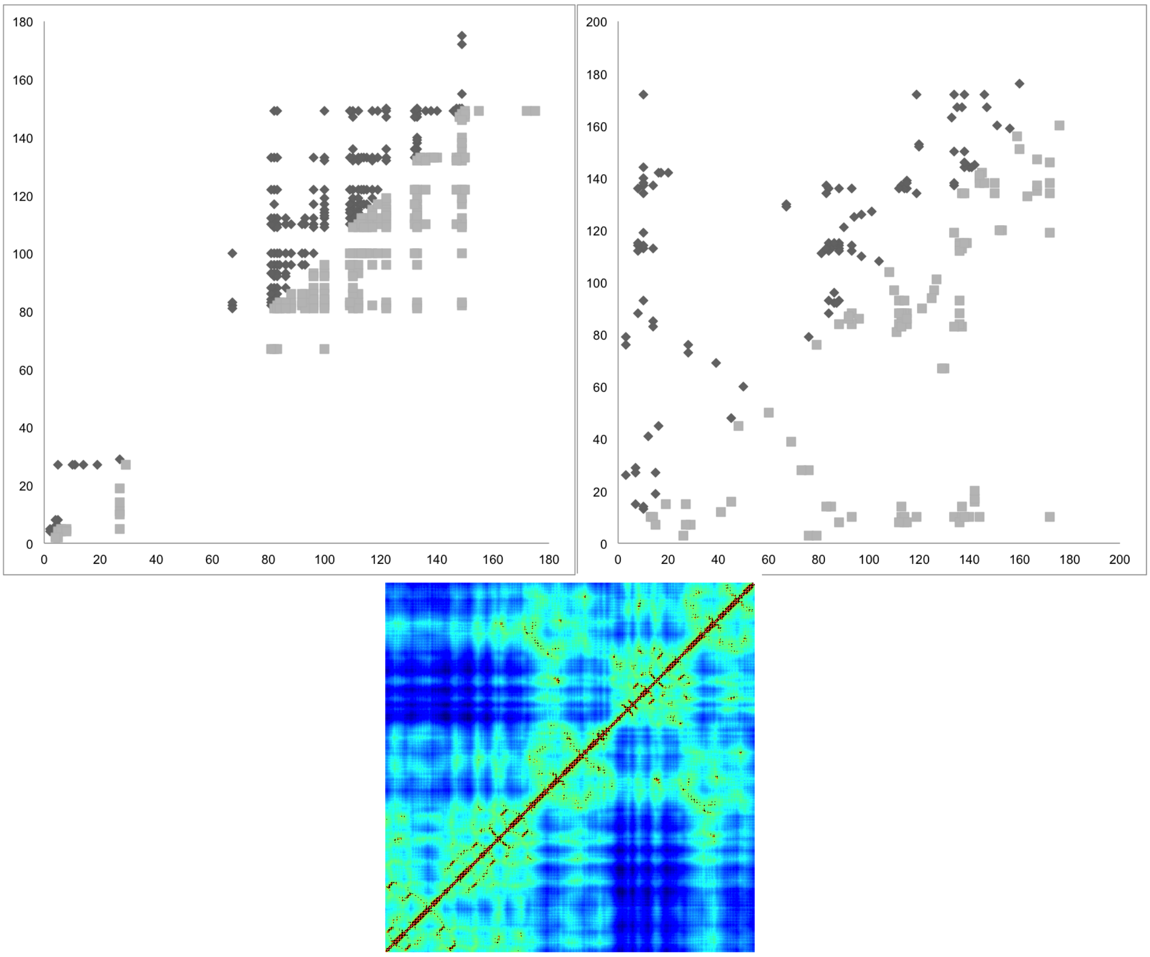


*Figure 5.10: Test performed with PF002737. The first picture in the first line shows results of developed algorithm, the second picture in the first line shows detected CMs by CMAT. All result graphs are linked to reference sequence Q3ACK8_CARHZ/3-182. In the second line, there is the distance map for a protein with PDB identificator 3MOG.*

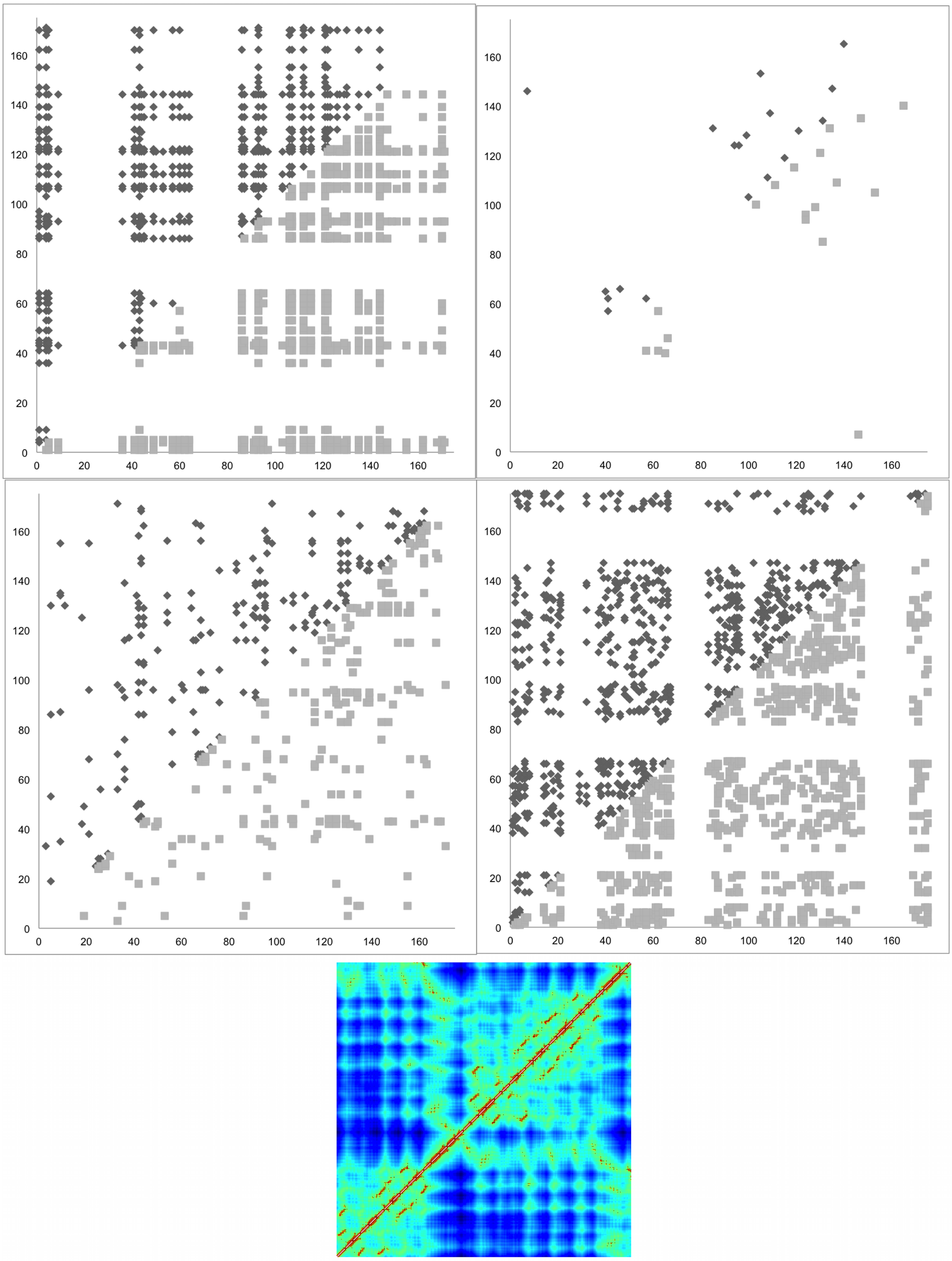


*Figure 5.11: Test performed with PF002826 (seed). The first picture in the first line shows results of developed algorithm, the second picture in the first line shows detected CMs by CMAT, the first picture in the second line shows detected CMs by CAPS, and the second picture in the second line shows detected CMs by CRASP. All result graphs are linked to reference sequence A9CGR6_AGRT5/108-282. In the third line, there is the distance map for a protein with PDB identificator 3BA1.*

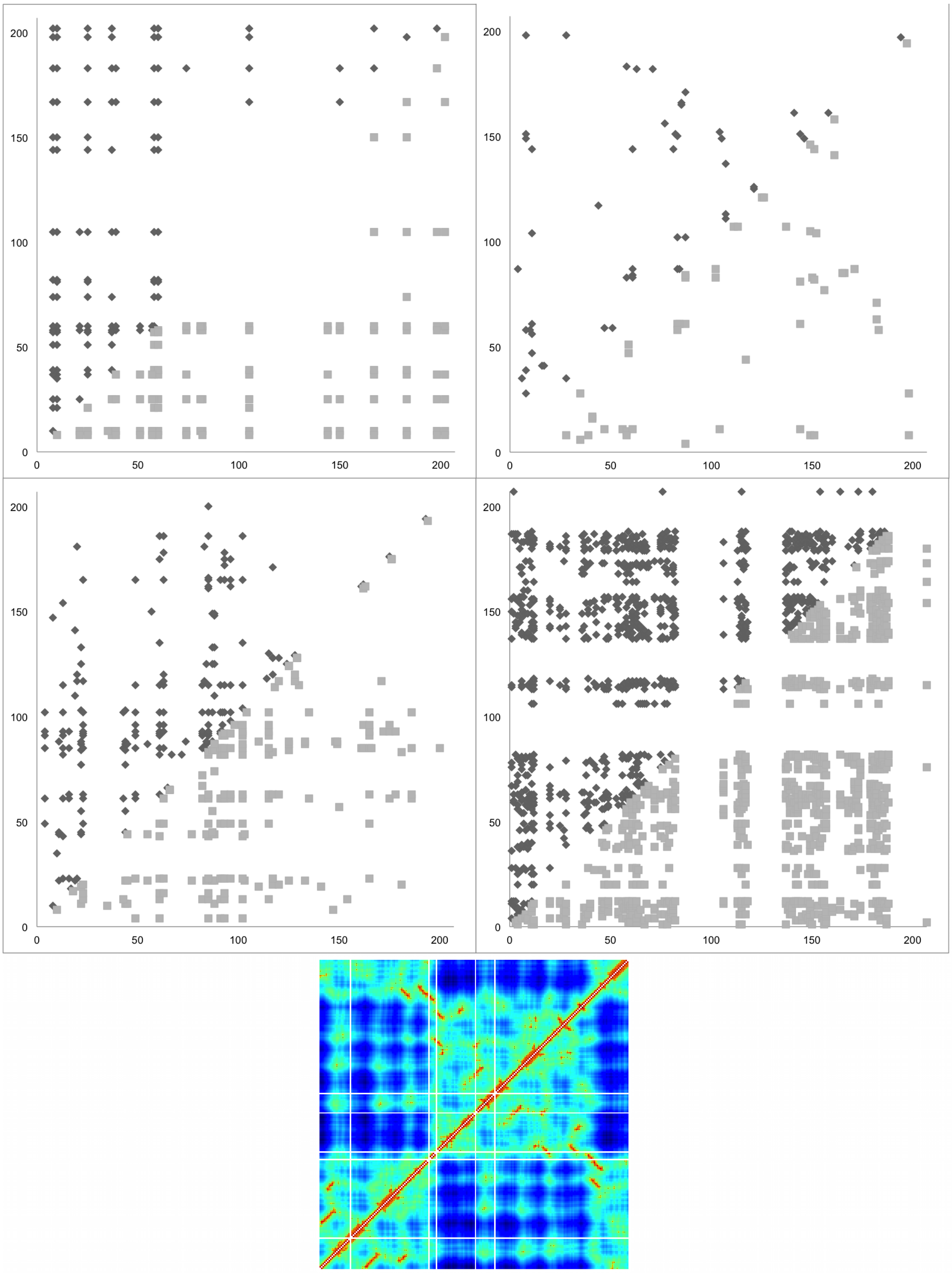


*Figure 5.12: Test performed with PF03466 (seed). The first picture in the first line shows results of developed algorithm, the second picture in the first line shows detected CMs by CMAT, the first picture in the second line shows detected CMs by CAPS, and the second picture in the second line shows detected CMs by CRASP. All result graphs are linked to reference sequence Q98A96_RHILO/94-300. In the third line, there is the distance map for a protein with PDB identificator 2QL3.*

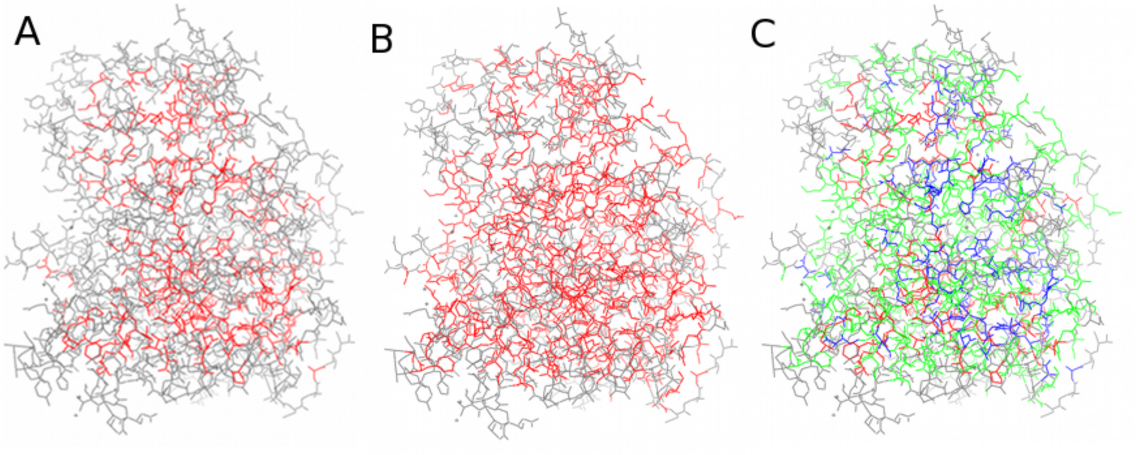


*Figure 5.13: Graphical comparison of detected correlated positions in PF00198. Grey amino acids were not detected as correlated. Red amino acids in the object A were detected as correlated by developed algorithm, in the object B by CMAT. The object C represents comparison. Red amino acids were detected only by developed algorithm, green amino acids only by CMAT and blue amino acids by both algorithms.*

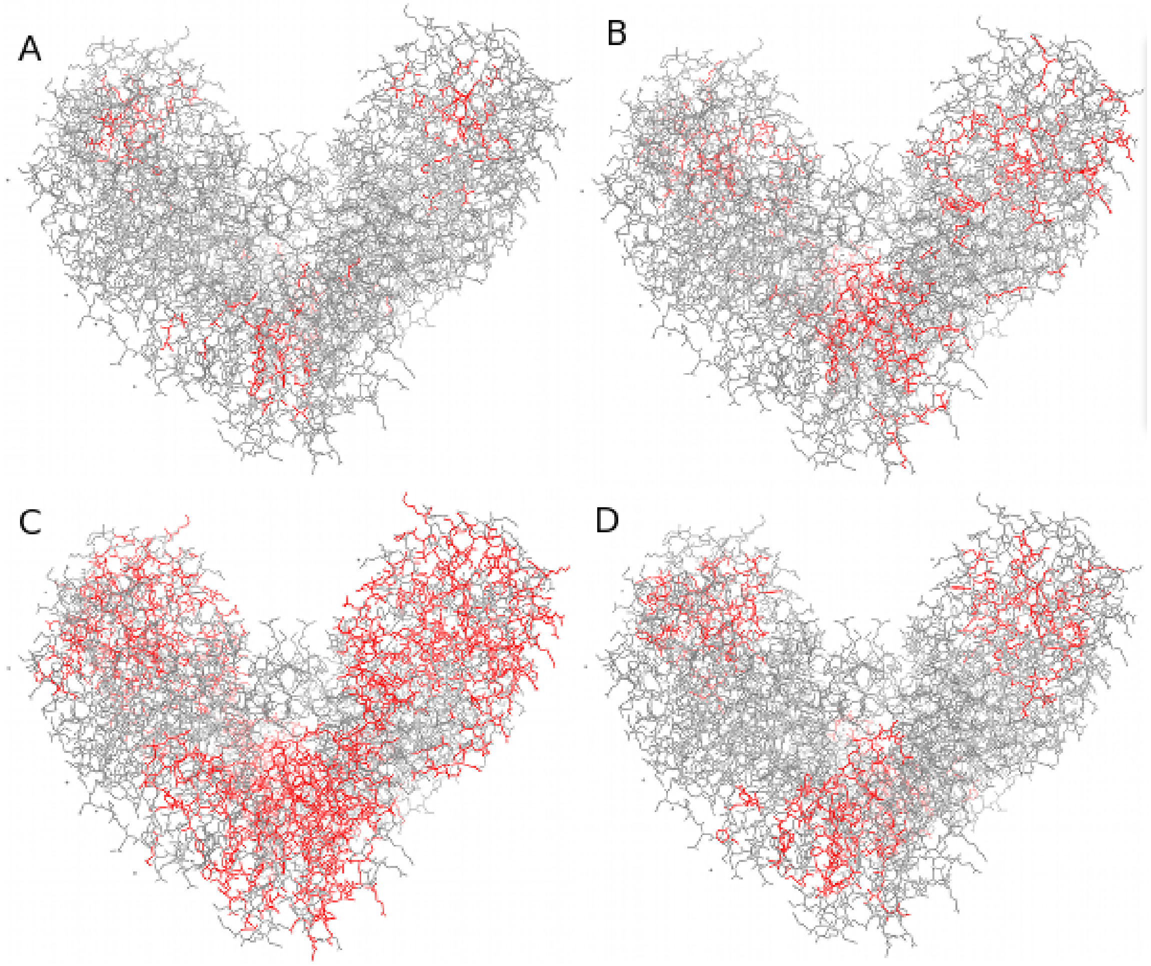


*Figure 5.14: Graphical comparison of detected correlated positions in PF00696. Grey amino acids were not detected as correlated. Red amino acids in the object A were detected as correlated by developed algorithm, in the object B by CMAT, in the object C by CAPS and in the object D by CRASP.*

| **PF00198** | detected CMs | <8Å | R | <16Å | R |
|---|---|---|---|---|---|
| developed algorithm | 164 | 38 (23.2%) | 7.129 | 111 (67.7%) | 3.092 |
| CMAT | 100 | 54 (54.0%) | 16.613 | 89 (89.0%) | 4.065 |

| **PF00696 (seed)** | detected CMs | <8Å | R | <16Å | R |
|---|---|---|---|---|---|
| developed algorithm | 135 | 16 (11.9%) | 3.902 | 79 (58.5%) | 2.667 |
| CMAT | 100 | 21 (21.0%) | 6.913 | 81 (40.5%) | 3.692 |
| CAPS | 913 | 93 (10.2%) | 3.353 | 353 (38.7%) | 1.762 |
| CRASP | 595 | 58 (9.7%) | 3.209 | 290 (48.7%) | 2.222 |

| **PF01094 (seed)** | detected CMs | <8Å | R | <16Å | |
|---|---|---|---|---|---|
| developed algorithm | 46 | 4 (8.7%) | 3.603 | 16 (34.8%) | 2.006 |
| CMAT | 62 | 8 (12.9%) | 5.346 | 29 (46.8%) | 2.698 |
| CAPS | 1337 | 80 (6.0%) | 2.479 | 342 (25.6%) | 1.475 |
| CRASP | 1955 | 68 (3.5%) | 1.441 | 371 (19.0%) | 1.095 |

| **PF02737** | detected CMs | <8Å | R | <16Å | R |
|---|---|---|---|---|---|
| developed algorithm | 173 | 35 (20.2%) | 4.553 | 142 (82.1%) | 2.631 |
| CMAT | 100 | 48 (48.0%) | 10.801 | 96 (96%) | 3.077 |

| **PF02826 (seed)** | detected CMs | <8Å | R | <16Å | R |
|---|---|---|---|---|---|
| developed algorithm | 483 | 50 (10.4%) | 2.333 | 312 (64.6%) | 2.099 |
| CMAT | 19 | 14 (73.7%) | 16.602 | 18 (94.7%) | 3.079 |
| CAPS | 203 | 33 (16.3%) | 3.663 | 115 (56.7%) | 1.841 |
| CRASP | 678 | 79 (11.7%) | 2.626 | 326 (48.1%) | 1.563 |

| **PF03466 (seed)** | detected CMs | <8Å | R | <16Å | R |
|---|---|---|---|---|---|
| developed algorithm | 134 | 14 (10.4%) | 5.119 | 45 (33.6%) | 2.549 |
| CMAT | 52 | 14 (26.9%) | 13.190 | 28 (53.8%) | 4.087 |
| CAPS | 176 | 13 (7.4%) | 3.619 | 48 (27.3%) | 2.070 |
| CRASP | 802 | 40 (5.0%) | 2.444 | 147 (18.3%) | 1.391 |

*Table 5.6: Table shows number of detected CMs in close proximity. Distance less than 8Å is considered to be a contact, distance less than 16Å can be considered to be close enough for intramolecular structural CMs. Distance more than 16Å implies functional or intermolecular structural CMs.*

random predictor (equation 2.2). Graphical comparison for the case of contacts is available on the picture 5.15.

## 5.4 Comparison with important sites

Some PDB files (for PF00198, PF00696 and PF2737) downloaded from Protein DataBank contain information about important functional sites. These sites can be correlated sometimes. The table below shows success in their detection using correlated mutations analysis. Developed algorithm detected only 135 correlated pairs in the case of PF00696.

| Protein family | possible pairs of sites | developed algorithm | cmat | caps | crasp |
|---|---|---|---|---|---|
| PF00198 | 9 | 1 | 1 | x | x |
| PF00696 (seed) | 165 | 0 | 12 | 16 | 12 |
| PF02737 | 7 | 0 | 0 | x | x |

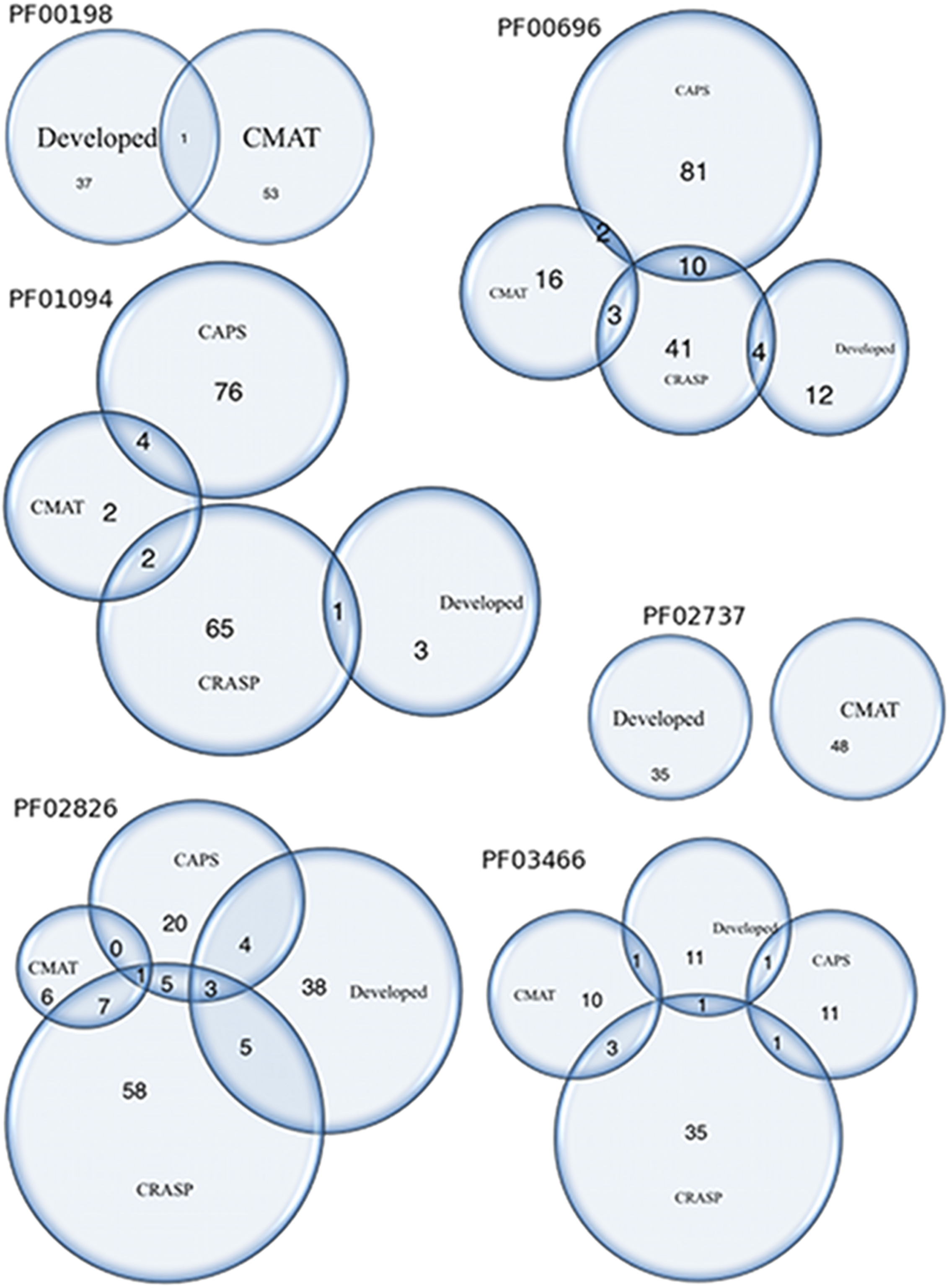


*Figure 5.15: This picture illustrates same structural correlated pairs detected by different algorithms.*

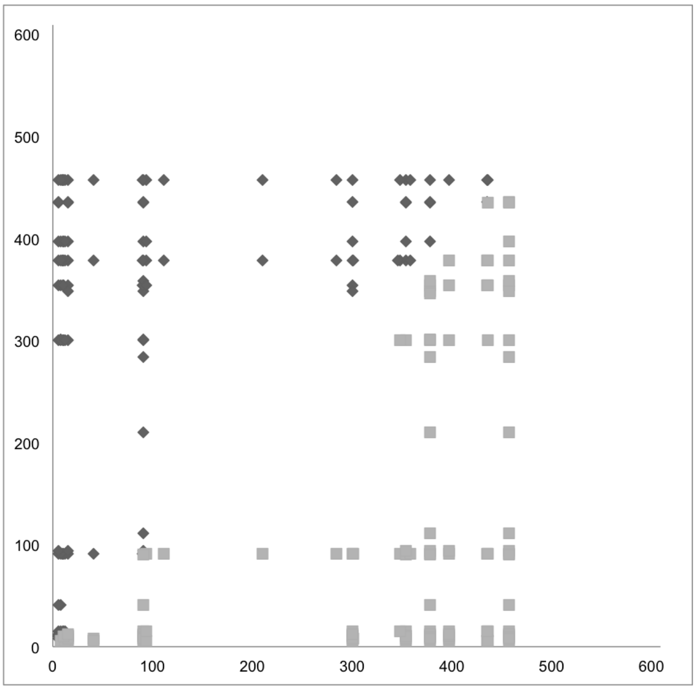


*Figure 5.16: Test was performed with PF00696 (seed). There are 609 positions (amino acids and gaps) in the MSA in total. Last 150 positions in sequences are random, these 150 positions should not correlate with any others. Penalization matrix version 1 was used in this case.*

## 5.5 Random sequences test

This test is based on detection of correlated mutations in selected protein family (PF00696) where 150 random amino acids were added to every sequence. Premise is that these last 150 positions should not correlate with any other positions. Results confirm this premise as seen on the picture 5.16. Test was successful in all five versions of the penalization matrix.

## 5.6 Test for verification of algorithm's characteristics

Final test for verification of an essential algorithm's characteristics was made. It was important to verify behavior in these cases:

- conserved positions - parameter determines minimal number of mutations
- gapped positions - parameter determines maximal number of gaps on the same position
- phylogenetic noise - the same as conserved positions
- stochastic noise - tolerance to random noise (incorrect MSA, incorrectly sequenced protein) - should be solved by setting a parameter of the minimal correlation score.
- correct identification of correlated positions
- tolerance to incorrect predecessors inference - see section 4.3.2

| Protein family | evolutionary conserved | gapped | total |
|---|---|---|---|
| PF00198 | 0 | 313 | 313 of 540 (58%) |
| PF00696 (seed) | 0 | 268 | 268 of 329 (81%) |
| PF01094 (seed) | 0 | 343 | 343 of 662 (52%) |
| PF02737 | 0 | 638 | 638 of 816 (78%) |
| PF02826 (seed) | 3 | 88 | 91 of 260 (35%) |
| PF03466 (seed) | 0 | 107 | 107 of 309 (35%) |

*Table 5.7: Table shows number of eliminated positions. Maximal evolutionary conservation was 99.5% and maximal gaps 25%. These values were chosen as default.*

| Protein family | evolutionary conserved | gapped | total |
|---|---|---|---|
| PF00198 | 19 | 305 | 324 of 540 (60%) |
| PF00696 (seed) | 2 | 165 | 167 of 329 (51%) |
| PF01094 (seed) | 0 | 271 | 271 of 662 (41%) |
| PF02737 | 15 | 634 | 649 of 816 (80%) |
| PF02826 (seed) | 6 | 78 | 84 of 260 (32%) |
| PF03466 (seed) | 0 | 93 | 93 of 309 (30%) |

*Table 5.8: Table shows number of eliminated positions. Maximal evolutionary conservation was 98% and maximal gaps 75%.*

For this purpose, the phylogenetic tree from PF00696 (seed) was used. MSA was made manually to cover all possibilities needed. Each sequence in this MSA has 35 amino acids (positions), where 10 amino acids were generated randomly. The developed algorithm's output corresponded to requirements. Test was made with the penalization matrix version 1.

## 5.7 Eliminated positions

Evolutionary conserved positions (section 4.2) and gapped positions (section 4.1) are eliminated before the computation of the correlation ratio of all remaining pairs. In test protein families with default parameters, following number of positions were eliminated as evolutionary conserved or gapped (with more than 25% of gaps on the position).

According to the table 5.7, elimination of evolutionary conservation is not very useful. Elimination of gapped positions is running as first process. Default parameter for evolutionary conservation is very high for tested input MSA, which was not created manually. For maximal evolutionary conservation 98% and maximal gaps 75%, number of filtered positions is showed in the table 5.8.

For example, positions with more than 10% of gaps and perfectly conserved positions are eliminated from analysis in the article [21].

## 5.8 Use case

The developed algorithm detects different correlated pairs. For example, the most correlated pairs in the case of PF00198 were 97-99, 99-112, and 97-112 (positions in the reference sequence). These positions creates one group (positions 268, 270, and 283 in the corresponding PDB file) which is illustrated on the picture 5.17. This group of positions was

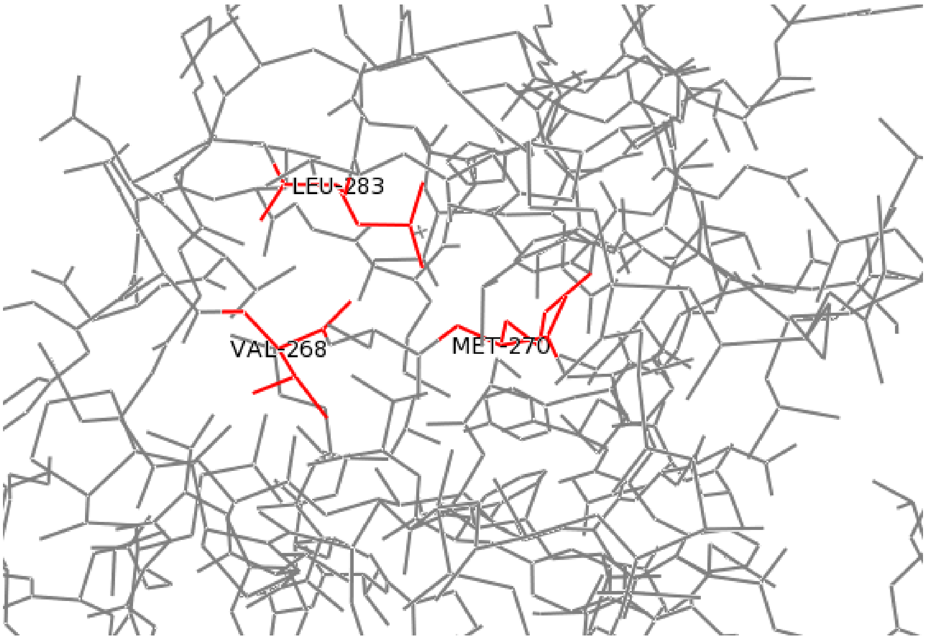


*Figure 5.17: Some correlated positions detected by the developed algorithm in PF00198.*

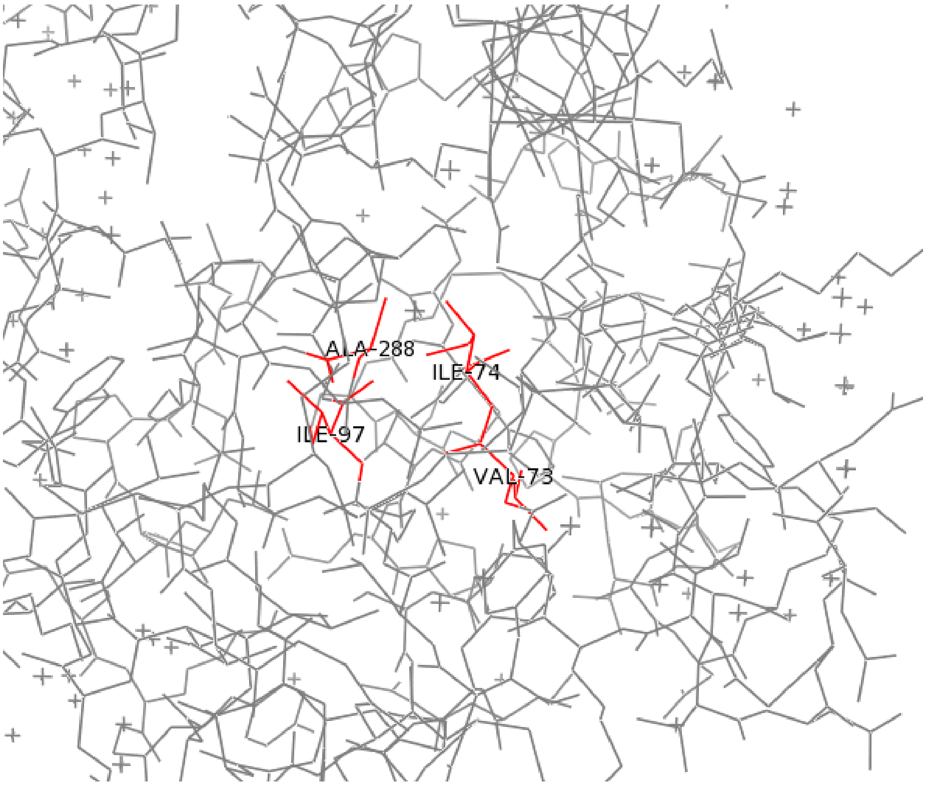


*Figure 5.18: Some correlated positions detected by the developed algorithm in PF01094.*

not detected by CMAT.

The most correlated pairs in the case of PF01094 were 39-40, 39-63, 40-63, 39-251, 40-251, and 63-251 (positions in the reference sequence). These positions creates one group (positions 39, 40, 63, and 251 in the corresponding PDB file) which is illustrated on the picture 5.18. This group of positions was not detected by CMAT nor CAPS. CRASP detected only pair 39-40.

## 5.9 Summary of results

Different tools produces unexpectly different results. Approximately only 12.65% of detected pairs were detected by two compared tools (even between CMAT x CAPS, CAPS x CRASP, and CMAT x CRASP). Due to this low shared pairs, tolerancy was introduced. Then, approximately 29.58% of correlated pairs are shared with two compared tools.

About 20% of detected correlated pairs are closer than 8Å (in large data sets), which corresponds with finding in the article [24], where was written that only 16.4% of correlated pairs had a distance less than 5.5Å. Tests also showed that tools also tend to detect pairs with distance <8Å more than pairs with distance <16Å (according to the improvement over the random predictor).

Most correlated pairs from two protein families create correlated groups (picture 5.17 and 5.18), which were not detected by other tools.

Tests were made on extensive protein families with great noise. For regular usage, manually prepared input is advised.

# Chapter 6

# Implementation

Algorithm is implemented as a web server because of simplicity of use from the view of a casual user - usually a biologist not experienced in programming languages. Essential part is written in Perl and is triggered through CGI from PHP. Results of each job are stored in a XML file and are available for download in a CSV file.

## 6.1 Input and output definition

Algorithm needs a multiple sequence alignment and a phylogenetic tree for correct analysis of correlated mutations. User can choose if he wants to input these data from file or by text fields. MSA has to be in FASTA format only and all sequences need to have a name for mapping to the phylogenetic tree. A phylogenetic tree has to be compatible with the Newick format and an evolutionary distance can be contained. There are some predefined input data, so user can try analysis without collecting valid input data.

Output is defined as couples of detected correlated positions. These positions are identified by their positions in a MSA and also by their position in a reference sequence (the first sequence in a MSA, gaps are ignored). Identification by position in reference sequence is used for easier mapping to 3D structure of proteins (and also is used as a single position information in some tools). Correlated positions are supplemented by the correlation score and results are sorted in descending order by this value. There are also evolutionary conservation scores for better idea about significance of these correlated positions. The less this conservation values are, the more significant correlation positions are. For completeness there are also lists of eliminated conserved and gapped positions.

### 6.1.1 The FASTA format

A FASTA format is used to store a single sequence or multiple sequence alignment. A single sequence begins with character $' >'$, optionally followed with a sequence identifier and description. Then, on the new line, sequence itself begins. Each amino acid is written as the single-letter code (these codes can be found in the appendix). Some supplementary codes are added for a gap (. or -) and an unknown amino acid (X). Code B means asparagine or aspartic acid, and is replaced by asparagine (amino acid N) in our implementation. Code Z means glutamine or glutamic acid, and is replaced by glutamine (amino acid Q). Code * is not used in our implementation. Sequence itself should contain a symbol of a new line each 60 codes.

Multiple sequence alignment is similar to a single sequence case, but single file can contain more sequences in format described before. All these sequences in a single file have the same length and identifiers of sequences are unique. Example of a MSA follows:

```
>A8M9I2_CALMQ/6-275
NVT–V-AKVGGSLLK-PG———–DVEK-VLSKVIE——R———–
————HMVD——–DGKLILVVSA—MK—–G—-VTDLLIKA—
—————–Y———–DEGKPHLIKDAIQPYL—NEAY—–RFG
LSKLGSLIEGVGERLETLINVREPWVRDNVVVH-GELLSV-MLIESILYN———-
ELGVDAGAVYDPGITT——-NEDWGKASVLG—-VSSHYVKHRLTWALSRR-SIVV
VPGFLG—ISL—-NG—-RLTSL—————GRGGSDYTASLIAAYINA
—P—RLIFYTDVEGIMTGDPRII-GDA-KVIPSLTHEEAYVAS————–
——–LTGAKKFHPGTFKPL-ID–S–NVNVMVTN
>A1RR70_PYRIL/1-257
MKP–V-VKIGGSLLR-TA———–GDFL-KAAEFIS——————
————LF———-KEPPVVVVSA—IK—–G—-VTDMLLEL—
—————–E———–KTRSYLLYEEILHKHLAVARLLGVEEKITPM
LKELEEALK———-LPRAEWTADYFASF-GERLSA-TILYAVCEK——KGIP
AKLFIAPIRTNSR————–FGNAEPL——QLEQKEEIADGN—-TVAV
VTGFIG—RDG—-EG—-RYTTV—————GRGGSDYTATYIGKEIGA
—R—KVSLVTDAPGVMTADPKEV-EDA-EVLPLMSIQEAIEAA————–
——–KAGAKNFHPRTFIPV-IE–A–DMSVEVRS
>Q97ZL7_SULSO/1-280
MAL–I-VKIGGSIQK-DE———–KDYE-LIVKKIQ——D———–
————FSKK——–SDKIIVVTSA—IK—–N—-VTNELISA—
———–TSNTDNSPNI—-VTEIYERHIKLLSKLADGKE—FENSFK-DISRL
SDELFRVAWSIR—–VLDEVTPRVRDYILSF-GERMAT-LLLSAILRS——NGIE
AEGIITPPFLTDEN———–YGEANVIED—-LSKNEIANILE–NAKA-NVIV
LPGFIG—RTR—-EG—-RYTTL—————GRGGSDYTATLLGKLVGV
—R—EVRLVTEVPGIMTGDPKKF-ENA-KTISRLSLEEAIELS————–
——–QLGAKRLHPRTFDPV-FG–S–DMKVIVES
```

This format became the standard in protein and nucleic acid sequences storage.

### 6.1.2 The Newick format

A phylogenetic tree is stored in Newick compatible format usually. Brackets surround each node pair and each node contains sequence identifier (the same as in the MSA) or its descendants followed by an internal node identifier. Then, optionally, character ':' follows with a phylogenetic distance value. Two nodes in node pair are separated by the ',' character. A node pair should contain two nodes only, but no one should rely on this assumption. Example of the phylogenetic tree is illustrated below (code and picture 6.1).

```
((Q18J38_HALWD/59-287:0.47836,Q4T581_TETNG/23-279:0.58547)0.610:0.02963,
(((A7M0B0_BACOV/6-234:0.42691,Q1VVE2_9FLAO/12-242:0.53657)0.850:0.07375,
(B4U7N9_HYDS0/1-223:0.46978,A6EJY3_9SPHI/4-225:0.59169)0.680:0.03659)
0.460:0.01296,A4C1S4_9FLAO/161-389:0.5763)0.600:0.02318)0.720:0.02185
```

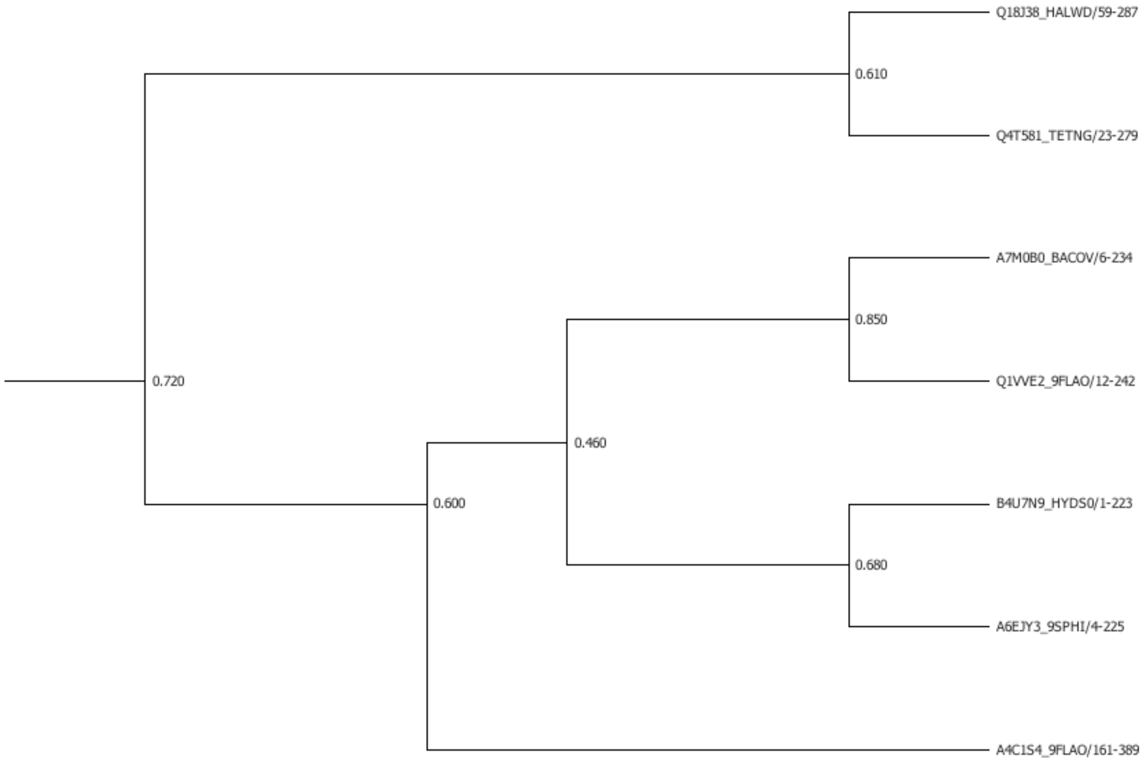


*Figure 6.1: Demonstrated phylogenetic tree*

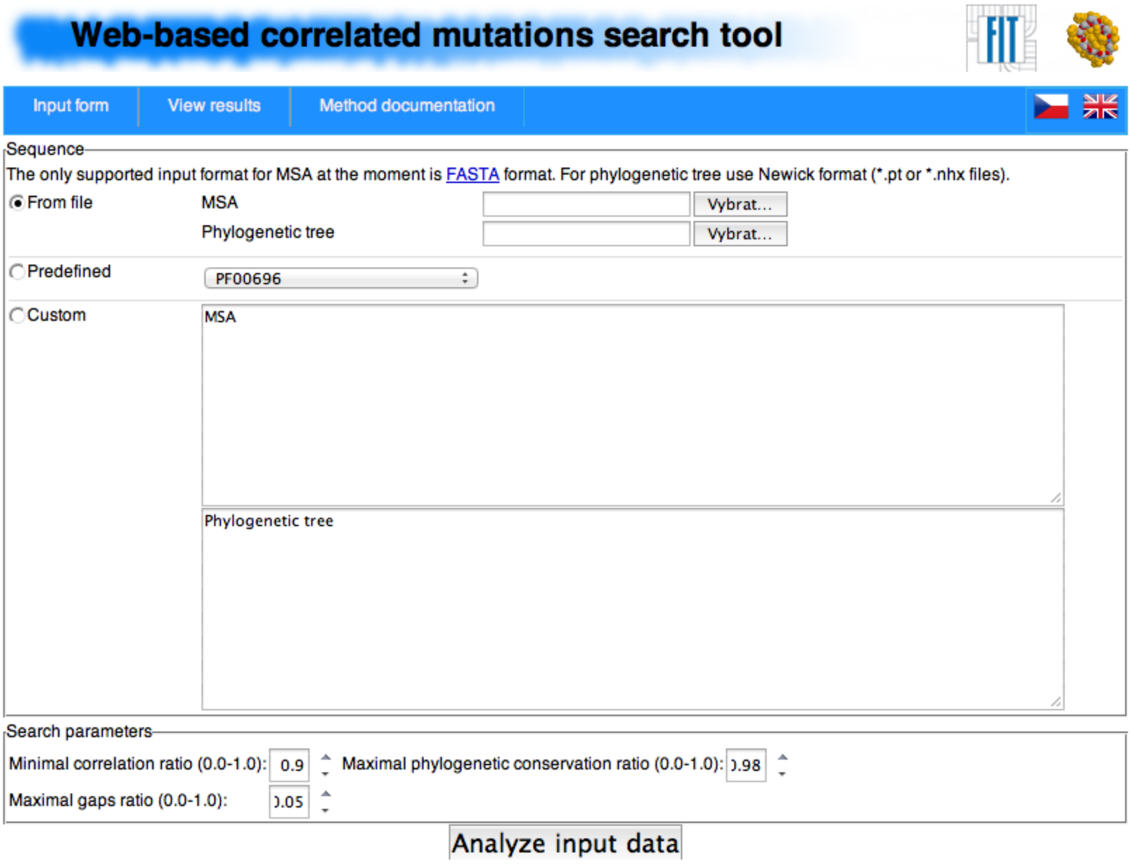
## 6.2 User interface

User interface is very simple and consists of two essential parts - input form and results presentation. All pages are bilingual (Czech and English), user can switch between languages using icons with national flags placed in the right upper corner on all pages.

Input form (picture 6.2) offers multiple ways for data input. Five protein families used in this work are prepared as select options for an easy trial. Further, FASTA MSA and Newick PT file inputs are present. Last option is to insert input data (FASTA MSA and Newick phylogenetic tree) through text fields. Before submitting input data, three parameters affecting the detection (section 4.4.4) can be modified: the minimal correlation ratio, the maximal conservation ratio, and the maximal gaps ratio.

After submitting input data, page (picture 6.3) showing information about processing of the job is opened. This page contains reference ID (automatically generated for each job, it is needed for reopening results), running time of the process, and links to files used as an input. Page is automatically reloaded each few seconds to inform user about job status. Once detection is completed, page shows results.

Page with results (picture 6.4) contains selected parameters, used reference sequence (the first sequence in a uploaded MSA), links to input files, links to output files (XML file with all needed data and CSV file with detected correlated pairs), and detected correlated pairs - positions in MSA, positions in reference sequence, computed correlation ratio and conservations of both positions. Page with results can be displayed after inputting reference ID into the form on page for showing the results. If cookies are enabled, reference IDs from last 7 days started from client's computer are displayed with current job status.



Web-based correlated mutations search tool

Sequence

The only supported input format for MSA at the moment is FASTA format. For phylogenetic tree use Newick format (*.pt or *.nhx files).

● From file — MSA [ ] Vybrat... — Phylogenetic tree [ ] Vybrat...

○ Predefined — PF00696

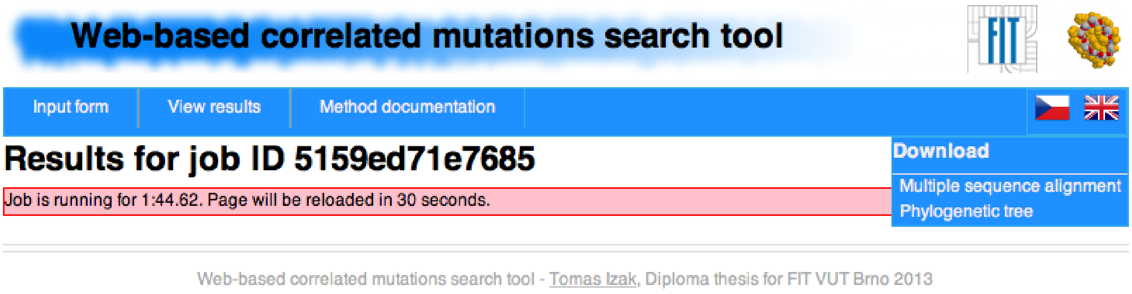
○ Custom — MSA — Phylogenetic tree

Search parameters

Minimal correlation ratio (0.0-1.0): 0.9 — Maximal phylogenetic conservation ratio (0.0-1.0): ).98

Maximal gaps ratio (0.0-1.0): ).05

*Figure 6.2: Input form of the developed web-based tool*



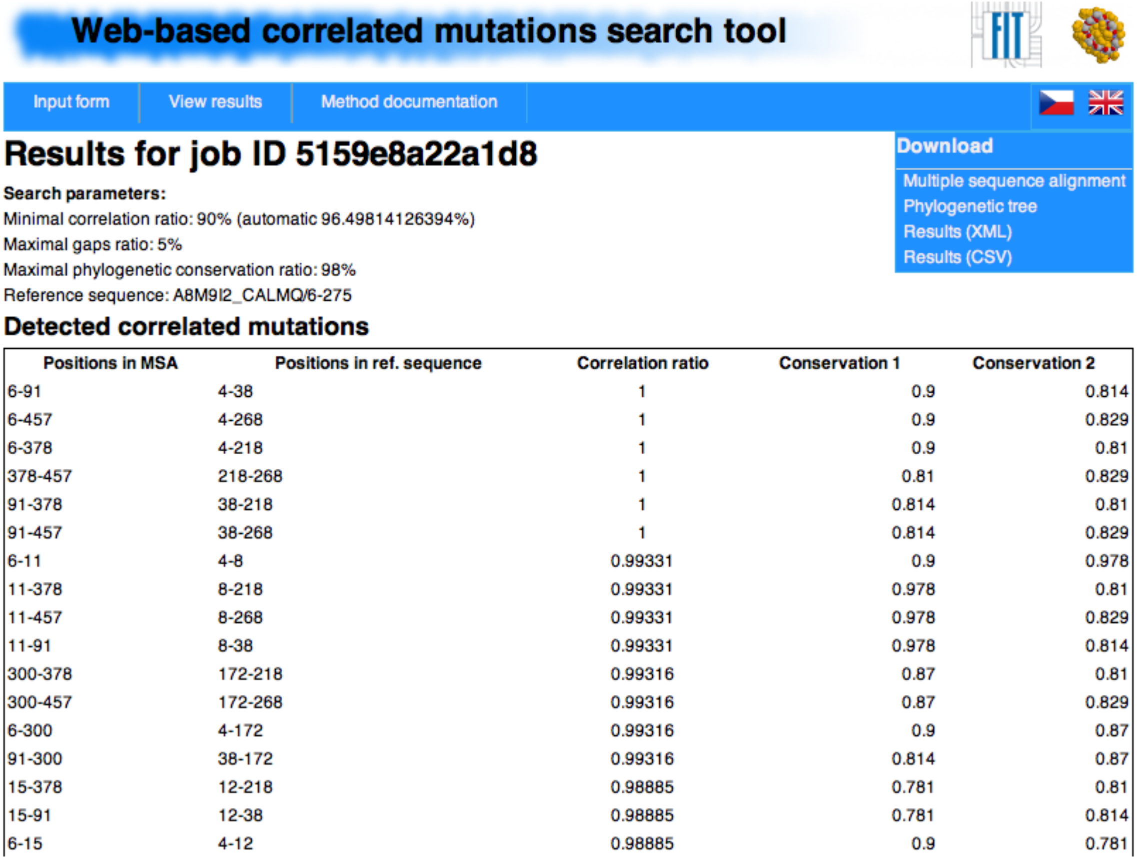
Web-based correlated mutations search tool

Results for job ID 5159ed71e7685

Job is running for 1:44.62. Page will be reloaded in 30 seconds.

Web-based correlated mutations search tool - Tomas Izak, Diploma thesis for FIT VUT Brno 2013

*Figure 6.3: Computation screen of the developed web-based tool*



## Results for job ID 5159e8a22a1d8

**Search parameters:**
Minimal correlation ratio: 90% (automatic 96.49814126394%)
Maximal gaps ratio: 5%
Maximal phylogenetic conservation ratio: 98%
Reference sequence: A8M9I2_CALMQ/6-275

### Detected correlated mutations

| Positions in MSA | Positions in ref. sequence | Correlation ratio | Conservation 1 | Conservation 2 |
|---|---|---|---|---|
| 6-91 | 4-38 | 1 | 0.9 | 0.814 |
| 6-457 | 4-268 | 1 | 0.9 | 0.829 |
| 6-378 | 4-218 | 1 | 0.9 | 0.81 |
| 378-457 | 218-268 | 1 | 0.81 | 0.829 |
| 91-378 | 38-218 | 1 | 0.814 | 0.81 |
| 91-457 | 38-268 | 1 | 0.814 | 0.829 |
| 6-11 | 4-8 | 0.99331 | 0.9 | 0.978 |
| 11-378 | 8-218 | 0.99331 | 0.978 | 0.81 |
| 11-457 | 8-268 | 0.99331 | 0.978 | 0.829 |
| 11-91 | 8-38 | 0.99331 | 0.978 | 0.814 |
| 300-378 | 172-218 | 0.99316 | 0.87 | 0.81 |
| 300-457 | 172-268 | 0.99316 | 0.87 | 0.829 |
| 6-300 | 4-172 | 0.99316 | 0.9 | 0.87 |
| 91-300 | 38-172 | 0.99316 | 0.814 | 0.87 |
| 15-378 | 12-218 | 0.98885 | 0.781 | 0.81 |
| 15-91 | 12-38 | 0.98885 | 0.781 | 0.814 |
| 6-15 | 4-12 | 0.98885 | 0.9 | 0.781 |

*Figure 6.4: Result presentation of the developed web-based tool*

# Chapter 7

# Conclusion

This work was devoted to correlated mutations in proteins. Its detection helps in protein engineering and in gaining new knowledge of proteins. Despite its importance, no general summary and comparison are available. A complex summary on this subject had been made during the term project of this thesis and can be found in the Theoretical part. Various tools and methods (mainly based on statistics or probabilistic models) have been developed until these days, but many of them suffer from evolutionary noise or non-usage of physico-chemical properties, which makes results inaccurate. That's why detection tool based on these features was developed in this work. This tool uses a multiple sequence alignment (MSA) and a phylogenetic tree (PT) as an input. Main idea is to compare all pairs of trees (positions) with inferred predecessors and evaluate their correlation score using the penalization matrix based on physico-chemical properties of amino acids, also presented in this work.

Developed algorithm was then tested and compared with other tools and with contact maps of proteins (section 5.2). Other tools were represented by CAPS (section 3.5.3), CRASP (section 3.5.2), and recently presented CMAT (section 3.5.4). Tests showed surprisingly different results between all tested tools (approximately only 12.65% detected pairs were detected by both compared tools).

Nonexistence of the general and accurate definition of correlated mutations results in nonexistence of complex test sets, that is why tests were made on six randomly selected protein families and results are not clear-cut. Graphs with detected correlated pairs show that developed algorithm detects pairs with low distance (but not necessarily contacts - correlated mutations can occur from more reasons). Test with contacts was chosen for its possibility of gaining necessary data and assembling of mostly automated tests. Tests showed that approximately 10% of detected correlated mutations in small input data (marked as *seed* in this text) were also contacts. Approximately 20% of detected CM pairs were also contacts in large input data (PF00198 and PF02737). Structural correlated mutations were detected by CMAT more often in some cases.

In this master's thesis, the algorithm not based on SCA, MI or probabilistic models was developed. Phylogenetic tree and physico-chemical properties of amino acids (hydropathy, polarity, and charge) gave other characteristics of detected pairs than usual methods. These differences were demonstrated by comparing outputs of multiple tools for detection of correlated mutations.

## 7.1 Further research

The most important (and the most difficult) part of the future work is to create an extensive test set with all needed information (especially correct output). As was written before, correct output (correlated pairs) is difficult to gain due to nonexistence of accurate definition of correlated mutations. All mechanisms concerning correlated mutations even do not have to be known.

Once a quality and extensive test set is prepared, it can be used for penalization matrix refinement using, for example, neural networks or evolutionary algorithms. Mutation types can be extended with multiple physico-chemical properties of amino acids like volume, presence of disulfide bonds, ionizability, etc. During the penalization matrix finding (f.e. using neural networks) can be determined that some of these groups are not necessary for detection of correlated mutations.

Next ideas are concerning user's interface. For easier use, an automatic generation of the phylogenetic tree from MSA could be used, when phylogenetic tree is not uploaded. Grouping of detected pairs could be also implemented for better presentation of results. Also task manager should be used to enqueue jobs in the case of frequent usage.

The last improvement will reduce time needed for detection. It concerns parallelization, various parts of algorithm can be processed in parallel (inferring of predecessors and computing of an evolutionary conservation, comparing of positions).

Protein engineering is more and more important domain, where correlated mutations help to reduce time needed for development of new proteins. Because this information is useful especially for biology and pharmacology, it is more than adequate to closely cooperate with colleagues working in this field.

# Bibliography

[1] D.A. Afonnikov and N.A. Kolchanov. Crasp: a program for analysis of coordinated substitutions in multiple alignments of protein sequences. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC441589/, 2004.

[2] Alberts B. and Bray D. *Základy buněčné biologie*. Espero Publishing, 2005.

[3] I.N. Berezovsky, K.B. Zeldovich, and E.I. Shakhnovich. Positive and negative design in stability and thermal adaptation of natural proteins. http://www.ncbi.nlm.nih.gov/pubmed/17381236, 2007.

[4] Ch.A. Brown and K.S. Brown. Validation of coevolving residue algorithms via pipeline sensitivity analysis: Elsc and omes and znmi, oh my! http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2879359/, 2010.

[5] J.P. Dekker, A. Fodor, R.W. Aldrich, and G. Yellen. A pertubation-based method for calculating explicit likelihood of evolutionary co-variance in multiple sequence alignments. http://bioinformatics.oxfordjournals.org/content/20/10/1565.short, 2004.

[6] J. Dutheil. Comap manual. http://home.gna.org/comap/, 2011.

[7] J. Dutheil and N. Galtier. Detecting groups of coevolving positions in a molecule: a clustering approach. http://www.biomedcentral.com/1471-2148/7/242, 2007.

[8] J. Dutheil, T. Pupko, A. Jean-Marie, and N. Galtier. A model-based approach for detecting coevolving positions in a molecule. http://mbe.oxfordjournals.org/content/22/9/1919.abstract, 2005.

[9] M.A. Fares and D. McNally. Caps: coevolution analysis using protein sequences. http://bioinformatics.oxfordjournals.org/content/22/22/2821.full, 2006.

[10] P. Fariselli and R. Casadio. A neural network based predictor of residue contacts in proteins. http://peds.oxfordjournals.org/content/12/1/15.full.pdf, 1999.

[11] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact maps with neural networks and correlates mutations. http://peds.oxfordjournals.org/content/14/11/835.full, 2001.

[12] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. http://www.ncbi.nlm.nih.gov/pubmed/11835493, 2001.

[13] S.J. Fleishman, O. Yifrach, and N. Ben-Tal. An evolutionarily conserved network of amino acids mediates gating in voltage-dependent potassium channels. http://ibis.tau.ac.il/wiki/nir_bental/index.php/Image:AECN.pdf, 2004.

[14] A.A. Fodor and R.W. Aldrich. Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. http://onlinelibrary.wiley.com/doi/10.1002/prot.20098/full, 2004.

[15] N. Friedman and D. Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. http://www.springerlink.com/content/nq13817217667435/, 2003.

[16] H. Gao, Y. Dou, J. Yang, and J. Wang. New methods to measure residues coevolution in proteins. http://www.biomedcentral.com/content/pdf/1471-2105-12-206.pdf, 2011.

[17] U. Göbel, Ch. Sander, S. Reinhard, and A. Valencia. Correlated mutations and residue contacts in proteins. https://cbio.mskcc.org/publications/papers/sander/114.pdf, 1994.

[18] R. Gouveia-Oliveira and A. G. Pedersen. Finding coevolving amino acid residues using row and column weighting of mutual information and multi-dimensional amino acid representation. http://www.almob.org/content/2/1/12, 2007.

[19] R. Gouveia-Oliveira, F. S. Roque, R. Wernersson, T. Sicheritz-Ponten, Peter W. Sackett, A. Mølgaard, and A. G. Pedersen. Intermap3d: predicting and visualizing co-evolving protein residues. http://bioinformatics.oxfordjournals.org/content/25/15/1963.short, 2009.

[20] L.H. Holley and M. Karplus. Protein secondary structure prediction with a neural network. http://www.pnas.org/content/86/1/152.full.pdf, 1989.

[21] D.S. Horner, W. Pirovano, and G. Pesole. Correlated substitution analysis and the prediction of amino acid structural contacts. http://bib.oxfordjournals.org/content/9/1/46.short, 2007.

[22] Chan-Seok J. and Dongsup K. Reliable and robust detection of coevolving protein residues. Protein Engineering, Design & Selection vol. 25 no. 11 pp. 705–713, 2012.

[23] I. Kass and A. Horovitz. Mapping pathways of allosteric communication in groel by analysis of correlated mutations. http://onlinelibrary.wiley.com/doi/10.1002/prot.10180/full, 2002.

[24] A. Kowarsch, A. Fuchs, D. Frischman, and P. Pagel. Correlated mutations: A hallmark of phenotypic amino acid substitutions. http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1000923, 2010.

[25] R. K. P. Kuipers, H.-J. Joosten, E. Verwiel, S. Paans, J. Akerboom, J. van der Oost, N. G. H. Leferink, W. J. H. van Berkel, G. Vriend, and P. J. Schaap. Correlated mutation analyses on super-family alignments reveal functionally important residues. http://www.ncbi.nlm.nih.gov/pubmed/19274741, 2009.

[26] B.-C. Lee and D. Kim. A new method for revealing correlated mutations under the structural and functional constraints in proteins.
`http://bioinformatics.oxfordjournals.org/content/25/19/2506.short`, 2009.

[27] S.W. Lockless and R. Ranganathan. Evolutionarily conserved pathways of energetic connectivity in protein families.
`http://www.sciencemag.org/content/286/5438/295`, 1999.

[28] S.C. Lovell and D.L. Robertson. An integrated view of molecular coevolution in protein-protein interactions.
`http://mbe.oxfordjournals.org/content/27/11/2567.short`, 2010.

[29] L. C. Martin, G. B. Gloor, S. D. Dunn, and L. M. Wahl. Using information theory to search for co-evolving residues in proteins.
`http://bioinformatics.oxfordjournals.org/content/21/22/4116.short`, 2005.

[30] T. Martinek, M. Lexa, and I. Burgetova. Biowiki: Cormut portal.
`http://bioware.fit.vutbr.cz/mediawiki/index.php/Detection_of_correlated_mutations`, 2012.

[31] Tomáš Martínek. Phylogenetic trees - course slides for bioinformatics. FIT VUT Brno - course slides, 4 2013.

[32] S. B. Nagl. Can correlated mutations in protein domain families be used for protein design? `http://www.ncbi.nlm.nih.gov/pubmed/11589588`, 2001.

[33] O. Olmea, B. Rost, and A. Valencia. Effective use of sequence correlation and conservation in fold recognition.
`http://www.sciencedirect.com/science/article/pii/S0022283699932084`, 2002.

[34] J. Pavlik. *Aplikovana statistika*. Vysoka skola chemicko-technologicka v Praze, 2005.

[35] J. Pei and N. V. Grishin. Al2co: calculation of positional conservation in a protein sequence alignment.
`http://bioinformatics.oxfordjournals.org/content/17/8/700.short`, 2001.

[36] A. F. Y. Poon, F. I. Lewis, S. D. W. Frost, and S. L. Kosakovsky Pond. Spidermonkey: rapid detection of co-evolving sites using bayesian graphical models.
`http://bioinformatics.oxfordjournals.org/content/24/17/1949.short`, 2008.

[37] Y. Qi and N. V. Grishin. Pcoat: positional correlation analysis using multiple methods.
`http://bioinformatics.oxfordjournals.org/content/20/18/3697.short`, 2004.

[38] S. Richter, A. Wenzel, M. Stein, R.R. Gabdoulline, and R.C. Wade. webpipsa: a web server for the comparison of protein interaction properties.
`http://nar.oxfordjournals.org/content/36/suppl_2/W276.short`, 2008.

[39] E.R.M. Tillier and T.W.H. Lui. Using multiple interdependency to separate functional from phylogenetic correlations in protein alignments.
`http://bioinformatics.oxfordjournals.org/content/19/6/750.short`, 2002.

[40] G. Vriend and J. Mol. Graph. What if. http://swift.cmbi.ru.nl/whatif/, 2012.

[41] F. Xu, P. Du, H. Shen, H. Hu, Q. Wu, and et al. Correlated mutation analysis on the catalytic domains of serine/threonine protein kinases. http://www.plosone.org/article/info:doi/10.1371/journal.pone.0005913, 2009.

# Appendix A

# DVD content

- source - source codes ofdeveloped web-based tool (prototype)
  - detection.cgi - Perl script
  - www
    - data - storage for input and output files, each job has its own folder inside
    - detection
      - data.pm - currently used Perl library, contains functions and constants needed for detection
      - phylo.pm - currently used Perl library, contains functions for phylogenetic trees
      - data_v1.pm - backup file - first version of penalization constants
      - data_v2.pm - backup file - second version of penalization constants
      - data_v3.pm - backup file - third version of penalization constants
      - data_v4.pm - backup file - fourth version of penalization constants
      - data_v5.pm - backup file - fifth version of penalization constants
  - languages - folder with language modifications (PHP file and PNG file for each language)
  - predefined - folder with predefined input files (MSA and NHX file for each predefined protein family)
  - config.php - contains variables and functions for setting the environment
  - index.php - generates main page for data input and executes the CGI script
  - show.php - generates page for results presentation
  - general.css - stylesheet
  - _del.php - function for deleting files inside the *data* folder
  - logo.png - main logo file
- text - LaTex source code files of this diploma thesis + final PDF file
- tests
  - tests_v1 - tests for default penalization matrix
  - tests_v2 - tests for penalization matrix version 2

- tests_v3 - tests for penalization matrix version 3
- tests_v4 - tests for penalization matrix version 4
- tests_v5 - tests for penalization matrix version 5
- scripts
    - cr_remover_uniq.sh - script for removing ’\r’ characters and duplicated rows
    - csv_compare_tolerancy.pl - script for comparing approximately same pairs (using distance information) from 2 CSV files
    - csv_compare.pl - script for comparing exactly same pairs from 2 CSV files
    - transform_from_ref_seq.pl - script for tranformation of positions from reference sequence numbering to protein numbering and storing pymol commands for coloring
    - transform_to_ref_seq.pl - script for extracting positions from CMView output and storing transformed positions into CSV file
    - transform_to_ref_seq-sites.pl - script for transformation of positions of sites gained from PDB file to reference sequence numbering

# Appendix B

# Manual

In this work, web based tool was developed using PHP and Perl via CGI. Source codes are available on attached DVD in *source* folder.

To deploy server tool:

1. Copy file *detection.cgi* to your CGI folder and verify its rights
2. Copy content of *www* folder where you want to deploy
3. Modify variable *cgi_path* in *config.php* file (in deployed folder) to path to CGI script (path to your CGI folder + *detection.cgi*)
4. Set correct permissions - read and launch for www users; write for www users is needed for *data* folder
5. Start web server

Developed tool was tested on Unix based machines (Linux 2.6.32-5-amd64 with Perl 5.10.1 and PHP 5.3.3; MacOS X 10.8.3 with Perl 5.12.3 and PHP 5.3.15). Perl script needs XML::Writer package. Prototype is available on `http://bioware.fit.vutbr.cz/~izak/`.

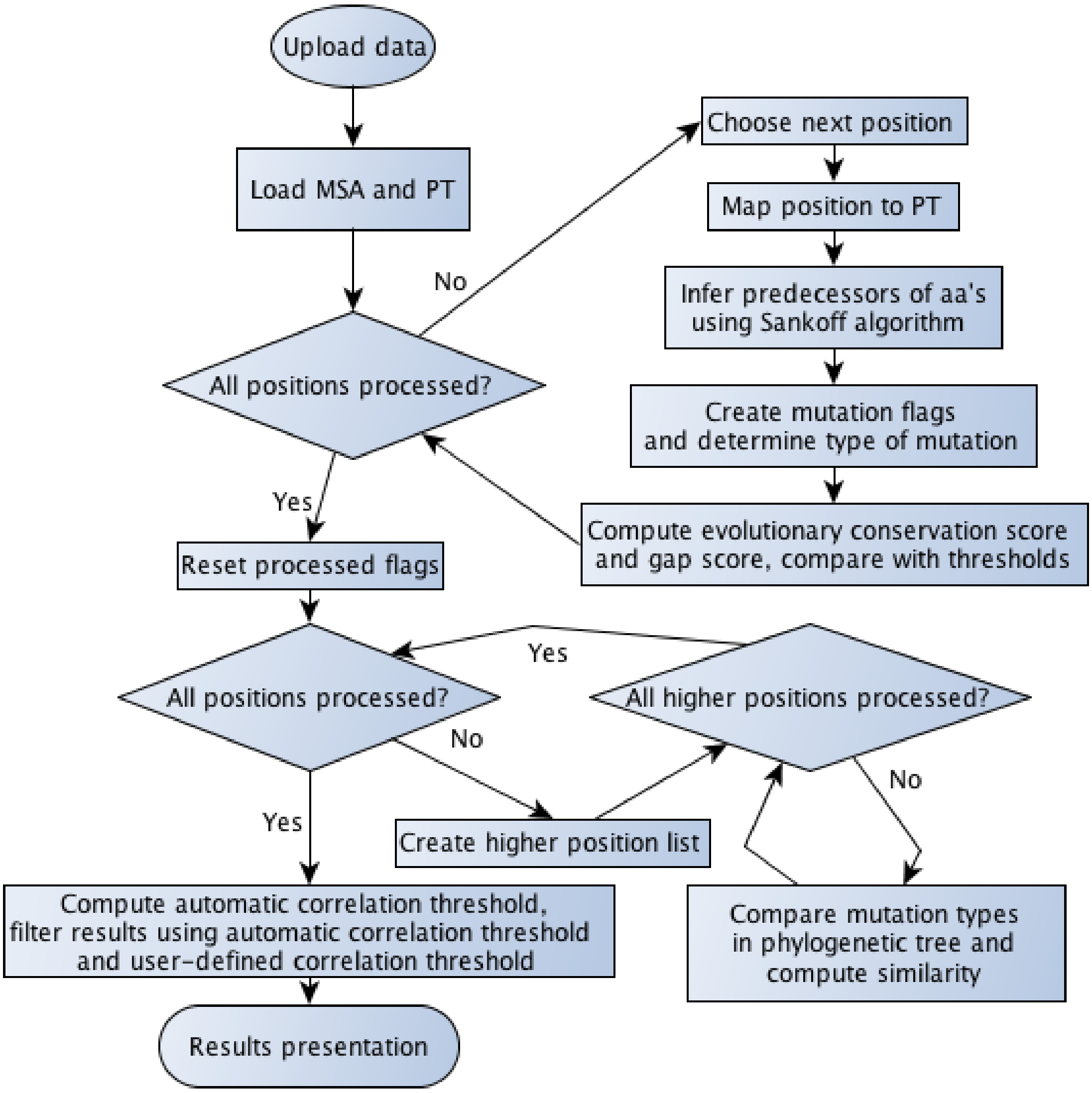
# Appendix C

# Physicochemical properties of aminoacids

| | Amino Acid | Side-chain polarity | Side-chain charge | Hydropathy index |
|---|---|---|---|---|
| A | Alanine | nonpolar | neutral | 1.8 (hydrophobic) |
| R | Arginine | polar | positive | -4.5 (hydrophilic) |
| N | Asparagine | polar | neutral | -3.5 (hydrophilic) |
| D | Aspartic acid | polar | negative | -3.5 (hydrophilic) |
| C | Cysteine | nonpolar | neutral | 2.5 (hydrophobic) |
| E | Glutamic acid | polar | negative | -3.5 (hydrophilic) |
| Q | Glutamine | polar | neutral | -3.5 (hydrophilic) |
| G | Glycine | nonpolar | neutral | -0.4 (hydrophilic) |
| H | Histidine | polar | neutral | -3.2 (hydrophilic) |
| I | Isoleucine | nonpolar | neutral | 4.5 (hydrophobic) |
| L | Leucine | nonpolar | neutral | 3.8 (hydrophobic) |
| K | Lysine | polar | positive | -3.9 (hydrophilic) |
| M | Methionine | nonpolar | neutral | 1.9 (hydrophobic) |
| F | Phenylalanine | nonpolar | neutral | 2.8 (hydrophobic) |
| P | Proline | nonpolar | neutral | -1.6 (hydrophiilc) |
| S | Serine | polar | neutral | -0.8 (hydrophilic) |
| T | Threonine | polar | neutral | -0.7 (hydrophilic) |
| W | Tryptophan | nonpolar | neutral | -0.9 (hydrophilic) |
| Y | Tyrosine | polar | neutral | -1.3 (hydrophilic) |
| V | Valine | nonpolar | neutral | 4.2 (hydrophobic) |

# Appendix D

# Flowchart of the algorithm



Upload data
Load MSA and PT
No
All positions processed?
Choose next position
Map position to PT
Infer predecessors of aa's using Sankoff algorithm
Create mutation flags and determine type of mutation
Compute evolutionary conservation score and gap score, compare with thresholds
Yes
Reset processed flags
Yes
All positions processed?
No
All higher positions processed?
No
Yes
Create higher position list
Compute automatic correlation threshold, filter results using automatic correlation threshold and user-defined correlation threshold
Compare mutation types in phylogenetic tree and compute similarity
Results presentation

*Figure D.1: Flowchart of the developed algorithm*

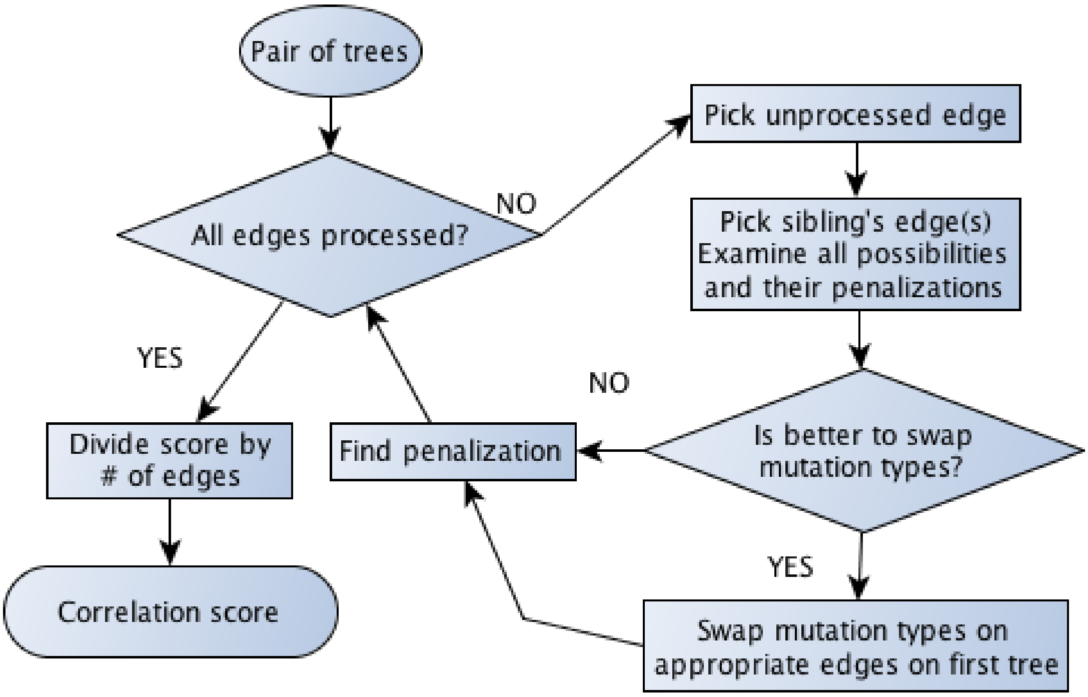


*Figure D.2: Flowchart of the comparison of the pair of trees*

# Appendix E

# McLachlan substitution matrix

MCLA710101 -McLachlan, A.D., Tests for comparing related amino-acid sequences cytochrome c and cytochrome c551, J. Mol. Biol. 61, 409-424 (1971)

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

# Appendix F

# Penalization values

- ∅ - mutation with no change of physicochemical property
- H - mutation with change of hydropathy (from hydrophilic to hydrophobic and vice versa)
- P - mutation with change of polarity
- SCh - mutation with change of charge (except from negative to positive and vice versa)
- BCh - mutation with change of charge (from negative to positive and vice versa)
- + - indicates combination of changes

## F.1 Version 1

This is default version used in all tests. Values in matrix were derived from McLachlan substitution matrix (see section 4.4.2).

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ∅ | 0 | | | | | | | |
| H | 0.84 | 0 | | | | | | |
| P | 0.65 | 0.19 | 0 | | | | | |
| SCh | 0.3 | 0.54 | 0.35 | 0 | | | | |
| BCh | 0.48 | 0.36 | 0.18 | 0.18 | 0 | | | |
| P+SCh | 0.5 | 0.34 | 0.15 | 0.2 | 0.02 | 0 | | |
| P+H | 0.69 | 0.41 | 0.04 | 0.39 | 0.21 | 0.19 | 0 | |
| P+SCh+H | 1 | 0.16 | 0.35 | 0.7 | 0.52 | 0.5 | 0.31 | 0 |
| | ∅ | H | P | SCh | BCh | P+SCh | P+H | P+SCh+H |

## F.2 Version 2

This version contains „random“ selected values, which reflects influence of hydropathy, charge and polarity on bonds between amino acids.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ∅ | 0 | | | | | | | |
| H | 1 | 0 | | | | | | |
| P | 0.5 | 1 | 0 | | | | | |
| SCh | 0.5 | 1 | 0.5 | 0 | | | | |
| BCh | 1 | 1 | 0.75 | 0.25 | 0 | | | |
| P+SCh | 1 | 1 | 0.25 | 0 | 0 | 0 | | |
| P+H | 1 | 0.5 | 0.5 | 0.75 | 0.75 | 0.5 | 0 | |
| P+SCh+H | 1 | 0.75 | 0.5 | 0.75 | 0.5 | 0.25 | 0.25 | 0 |
| | ∅ | H | P | SCh | BCh | P+SCh | P+H | P+SCh+H |

## F.3 Version 3

This version contains „random“ selected values, which reflects big influence of charge and polarity and small influence of hydropathy on bonds between amino acids.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ∅ | 0 | | | | | | | |
| H | 0.1 | 0 | | | | | | |
| P | 0.5 | 0.6 | 0 | | | | | |
| SCh | 0.5 | 0.6 | 0.25 | 0 | | | | |
| BCh | 0.75 | 0.85 | 0.5 | 0.25 | 0 | | | |
| P+SCh | 1 | 1 | 0 | 0 | 0.25 | 0 | | |
| P+H | 0.6 | 0.5 | 0.1 | 0.35 | 0.65 | 0.1 | 0 | |
| P+SCh+H | 1 | 1 | 0.35 | 0.1 | 0.35 | 0.1 | 0.1 | 0 |
| | ∅ | H | P | SCh | BCh | P+SCh | P+H | P+SCh+H |

## F.4 Version 4

This version contains „random“ selected values, which reflects influence of hydropathy, charge and polarity on bonds between amino acids. This version is more strict than version 2.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ∅ | 0 | | | | | | | |
| H | 1 | 0 | | | | | | |
| P | 0.75 | 1 | 0 | | | | | |
| SCh | 0.75 | 1 | 0.75 | 0 | | | | |
| BCh | 1 | 1 | 0.75 | 0.25 | 0 | | | |
| P+SCh | 1 | 1 | 0.25 | 0.25 | 0.25 | 0 | | |
| P+H | 1 | 0.75 | 0.75 | 1 | 1 | 0.9 | 0 | |
| P+SCh+H | 1 | 0.9 | 0.75 | 0.75 | 0.75 | 0.75 | 0.25 | 0 |
| | ∅ | H | P | SCh | BCh | P+SCh | P+H | P+SCh+H |

## F.5 Version 5

This version contains „random“ selected values, which reflects influence of hydropathy, charge and polarity on bonds between amino acids. This version is the most strict of all five versions.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ∅ | 0 | | | | | | | |
| H | 1 | 0 | | | | | | |
| P | 1 | 1 | 0 | | | | | |
| SCh | 1 | 1 | 0.5 | 0 | | | | |
| BCh | 1 | 1 | 0.5 | 0.5 | 0 | | | |
| P+SCh | 1 | 1 | 0.1 | 0.1 | 0.5 | 0 | | |
| P+H | 1 | 1 | 0.5 | 0.75 | 0.75 | 0.75 | 0 | |
| P+SCh+H | 1 | 1 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0 |
| | ∅ | H | P | SCh | BCh | P+SCh | P+H | P+SCh+H |

# List of Figures