UNIVERSITE D’ORLEANS
UNIVERSITE DE ZAGREB

MASTER SCIENCES ET TECHNOLOGIES

mention : BIOLOGIE-BIOCHIMIE

spéciality : BIO-INDUSTRIAL TECHNICS

Stage report

Functional classification of Adenylation domains by Latent Semantic Indexing (LSI)

by

Damir BARANAŠIĆ

From January 31 to June 10, 2011

Section for Bioinformatics, PBF, University of Zagreb
Pierottijeva 6, 10 000, Zagreb, Croatia

Superviser of the stage:	**Assistant professor Antonio Starčević, PhD**
	Senior Scientific Assistant Jurica Žučko, PhD
Referee:	**Professor Daslav Hranueli, PhD**
	Professor Philippe Jacques, PhD

Date of defense: July 1, 2011

Abstract:

Latent semantic indexing (LSI) is an information retrieval method which has relatively recently been introduced into computational biology. In this work, LSI was adapted for prediction of the amino acid substrates which are activated by adenylation domains (A-domains). A-domains are obligatory subunits of non-ribosomally synthesised peptide synthetases (NRPS) modules which recognise and activate the amino acid that must be incorporated into the final product, non-ribosomally sythesised peptides. Knowing the specific A-domain substrate for every sequenced A-domain would enable us to predict the final product of linear NRPS and perhaps design novel biologically active natural products. Two methods were used to vectorize A-domain protein sequences and to construct the resulting term-document matrix: "*n*-grams" method and a novel "tokenization" method. The "*n*-grams" method finds *n*-peptides in the protein sequence, and the "tokenization" method creates specific "tokens", which couple amino acid residues with the corresponding positions in the multiple sequence alignment. LSI uses a mathematical method called singular value decomposition (SVD) to reduce the unreliable information from the term-document matrix. The number of dimensions used in analysis was obtained computationally and was found to be in accordance with the empirically obtained optimal number of dimensions. Predictions obtained were satisfactory using both "*n*-grams" and "tokenization" as vectorization methods. "Tokenization" method generally showed better precision and robustness. A novel clustering method based on LSI was also developed. It showed satisfactory clustering results without the need to guess the numbers of clusters in advance which methods such as k-means clustering require.

Key words: LSI, A-domains, protein tokenization, protein clustering, SVD, dimension reduction, specificity prediction

Résumé

Indexation sémantique latente (LSI) est une méthode de recherche d'information qui a relativement récemment été introduite en biologie computationnelle. Dans cette thèse, LSI a été adaptée pour la prédiction des substrats acides aminés qui sont activés par des adénylation domaines. A domaines sont des sous-unités obligatoires des modules des enzymes de synthèse des peptides non ribosomiques (NRPS) qui reconnaissent et actvent l'acide aminé qui doit être incorporé dans le produit final, c'est-à-dire les peptides non ribosomique. La connaissance du substrat spécifique du A domaine pour chaque A domaine séquencé nous permet la prévision du produit final des NRSP enzymes linéaires et la pssibilité de designer de nouveaux produits biologiquement actifs. Deux méthodes ont été utilisées pour vectoriser les séquences protéine des A domaines et pour construire la matrice résultante des termes et des documents: la „n-grammes" méthode et la nouvelle „tokenisation" méthode. La „n-grammes" méthode trouve les n-peptides dans la séquence protéine, et la méthode „tokenisation" crée des „tokenes" spécifiques qui couplent les résidus acides aminés avec les positions correspondantes dans l'alignement séquentiel multiples. LSI utilise une méthode mathématique appelée décomposition en valeurs singulières (SVD) pour la réduction des information inutiles de la matrice des termes et des documents. Le nombre des dimensions optimale utilisées dans l'analyse a été obtenu mathématiquement et est en conformité avec le nombre obtenu empiriquement. Les prévisions obtenues sont satisfaisantes en utilisant les deux méthodes („n-grammes" et „tokenisation") comme méthodes de vectorisation. La méthode de „tokenisation" a généralement montré une meilleure précision et robustesse. Une nouvelle méthode pour les groupes des protéines basée sur la LSI est également développée. Cette méthode donne des résultats satisfaisants du regroupement sans avoir besoin de deviner le nombre de groupes à l'avance, par des méthodes telles que k-means exige.

Mots clés: LSI, A domaines, tokenisation des séquences protéine, protéine group, SVD, réduction des dimensions, prédiction de spécificité

Table of Contents:

1. INTRODUCTION **1**

2. THEORY **2**

2.1. NATURAL PRODUCTS 2
2.2. LINEAR NRPS 3
2.3. ADENYLATION DOMAINS 4
2.4. A-DOMAIN SUBSTRATE SPECIFICITY PREDICTION PROGRAMS 5
2.5. LATENT SEMANTIC INDEXING (LSI) 6
2.6. AIMS OF THE WORK 7

3. MATERIALS AND METHODS **8**

3.1. MATERIALS 8
3.1.1. Computers and operating systems *8*
3.1.2. Packages and tools used *8*
3.1.2.1 MATLAB 8
3.1.2.3. ClustalX2 8
3.1.2.2. Similarity search 9
3.1.3. Protein sequences *10*
3.2. METHODS 10
3.2.1. Protein sequences acquisition *10*
3.2.2. Substrate specificity identification *10*
3.2.3. Term-document matrix construction *11*
3.2.3.1. n-peptides (n-grams) 11
3.2.3.2. Protein sequence tokenization 12
3.2.4. Singular value decomposition *12*
3.2.5. Query vector construction *13*
3.2.6. Computing cosine values and specificity assignment/prediction *13*
3.2.7. Method precision determination (Determination of method precision) *14*
3.2.8. Novel clustering method *14*

4. RESULTS **16**

4.1. TOKENIZATION OF PROTEIN SEQUENCES 16
4.2. THE CHOICE OF FACTOR *K* FOR OPTIMAL DIMENSION REDUCTION 17
4.3. DETERMINING COSINE THRESHOLD 18
4.4. NOVEL CLUSTERING METHOD 20

5. DISCUSSION **24**

6. CONCLUSIONS **29**

7. THANKS **30**

8. REFERENCES **31**

9. ANNEXES **35**

9.1. SUPPLEMENTARY FIGURES 35
9.2. ABBREVIATION OF AMINO ACIDS 37

1. Introduction

In the last few decades, the discovery of new antibiotics slowed down markedly. The reason for that is because the probability of finding useful antibiotics from microbes using conventional techniques was very low. However, the war against pathogen microorganisms is not won yet, and unfortunately, will probably never be (Demain, 2009). The most recent example is the new strain of *E. coli* strain O104:H4 which has resistance genes to multiple classes of antibiotics (Nature Publishing Group, 2011).

There is now unprecedented opportunity to access the natural diversity of small molecules made by microbes by the isolation of metagenomic DNA and heterologous expression of biosynthetic pathways in fermentable hosts. Discovery of novel biosynthetic gene clusters is the first goal of this culture-independent research that requires the application of molecular bioinformatics to identify DNA sequences of interest. In the Section for Bioinformatics at the Faculty of Food Technology and Biotechnology, a program package named *ClustScan* was developed for this purpose. It provides a semi-automatic annotation of DNA sequences encoding modular biosynthetic enzymes including polyketide synthases (PKS), non-ribosomal peptide synthetases (NRPS) and hybrid (PKS/NRPS) enzymes. It also predicts polyketide aglycons chemical structures (Starcevic *et al.*, 2008).

There is also a tool named NRPS.predictor which predicts the amino acid activated by NRPS A-domains. To predict the substrate specificity of subtypes of a given protein sequence family it uses support vector machine (SVM)-based approach. This tool represents the protein sequences as vectors based on the physico-chemical properties of the amino acid residues. Amino acid residues used for vector construction in this tool are A-domain residues aligned to residues of gramicidin synthetase A that are 8 Å around the substrate amino acid in the active site cleft. The main limitation of this aproach is that specificities for very similar substrates that frequently show cross-specificities were pooled to the so-called composite specificities and predictive models were built for them (Rausch *et al.*, 2005).

In this work we developed a method based on latent semantic indexing (LSI) which among other things can also predict the amino acid substrate activated by NRPS A-domains. Unlike NRPS.predictor, our method predicts a specific amino acid which activates the A-domain and not a general group of similar amino acids. The goal of this work was to extend existing *ClustScan* functionalities, which could than also predict the structure of NRPS enzyme products and NRPS/PKS hybrid enzyme products.

2. Theory

2.1. Natural products

In the twentieth century, the human lifespan has doubled. One of the reason for that is more extensive utilization of microbial and plant natural products (Demain, 2009). For instance, microbial natural products can be used as antibiotics, antitumor agents, immunosuppressant, hypolipemics, enzyme inhibitors, etc. (Demain, 1999). Summarising the numbers, there are around 100 000 discovered compounds with the molecular mass less than 2 500 Daltons produced by microorganisms and plants (Demain, 2009; Demain, 1999). Until 2005, over 22 500 microbial bioactive molecules were found of which 17% are synthesized by non-filamentous bacteria, 45% by filamentous bacteria (actinomycetes), and 38% by filamentous fungi. Of all natural products synthesized by actinomycetes, 80% of them are produced by the genus *Streptomyces* (Demain, 2009).

The two biggest groups of natural compounds are polyketides (PK) and non-ribosomally synthesized peptides (NRP) (Hranueli *et al.*, 2005). Gene clusters that code for enzymes which utilise both mechanisms were also reported (Schwarzer and Marahiel, 2001).

Polyketides are one of the largest groups of natural products. They are being synthesized by a consecutive condensation steps, introducing simple organic acids residues such as acetate, propionate and butyrate. The small building blocks are activated as thioesters by coenzyme A, and the product of each cycle is a β-keto group which can later be reduced (Hranueli *et al.*, 2005). Polyketides are synthesized by enzymatic complexes called polyketide synthases (PKS).

There are three types of PKS, named Type I, II and III respectively. PKS Type I, in charge of the biosynthesis of complex polyketides, are the most important ones for pharmaceutical industry. Biochemical reactions involved in biosynthesis of complex polyketides are conducted by catalytically active protein regions called domains which are grouped as modules of one or more multifunctional proteins. Unlike PKS Type I, the active domains of PKS Type II are located on monofunctional or bifunctional proteins for every enzymatic reaction (Sattely *et al.*, 2005). On the other hand, PKS Type III enzymes catalyze a varying numbers of iterative condensations utilizing a diverse set of large starter molecules on a single catalytic site (Austin and Noel, 2003).

Non-ribosomally synthesized peptides are low molecular weight compounds which in their main structure frequently incorporate cyclic or branched peptide backbone and can also introduce non-proteinogenic amino acid residues (Schwarzer and Marahiel, 2001). The examples of non-ribosomally synthesized peptides are: ACV, penicillin precursor, vancomycin (Mootz *et al.*, 2002), immunosuppressant cyclosporine and cytostatic bleomycin (Strieker *et al.*, 2010.). Their synthesis is carried by a consecutive amino acid condensation on large multimodular enzymes called nonribosomal peptide synthetases (NRPS) (Strieker *et al.*, 2010).

There are three different types of NRPSs. Linear NRPS (Type A) contain several elongation modules, each responsible for one amino acid incorporation in the growing chain. Amino acid sequence in their case is completely determined by the number and the order of modules. Iterative NRPS (Type B) on the other hand, use their modules or domains more than once which results in synthesis of peptide chain consisting of repeated smaller amino acid sequences. Nonlinear NRPS (Type C) do not have the same order of core domains in the module as linear NRPS do. Because of that, products of these NRPS often introduce uncommon structural forms like cycled or branched peptides (Mootz *et al.*, 2002).

In microorganisms, natural product biosynthetic genes are usually located on contiguous DNA fragments known as gene clusters (Donadio *et al.*, 2007). Gene clusters encoding for natural products are one of the largest known and can span on more than 100 kb of DNA (Fischbach *et al.*, 2008). From the bioinformatics point of view, it is easy to detect PKS and NRPS modules based on sequence similarity, and it is also possible to predict the biosynthetic pathway and chemical product because the order of module regions in the DNA sequence matches the order of protein modules in the polypeptide chain. Directions for biosynthesis of novel natural products could be obtained by investigating the recombination of polyketide gene clusters *in silico*, or by some other approach, but the number of such “un-natural” products is still unknown (Hranueli *et al.*, 2005).

2.2. Linear NRPS

As building blocks, NRPS use amino acids (Caboche *et al.*, 2009) or hydroxyl acids forming an amide or ester bond (Donadio *et al.*, 2007). NRPS are made up by a series of modules where every module is responsible for specific monomer incorporation into the final product (Strieker *et al.*, 2010). Each NRPS module consists of peptidyl carrier protein (PCP), adenylation domain (A) and condensation domain (C). In most NRPSs there is a thioesterase-like domain (Te) at the C-terminal end (Mootz *et al.*, 2002).

The A-domains catalyze the first step in the non-ribosomally synthesized peptide formation, together with the amino acid substrate recognition and activation as amino-acyl adenylate, as shown on Figure S1a (Schwarzer and Marahiel, 2001). A-domains often have a relaxed substrate specificity compared to aminoacyl-tRNA synthetase, which carry out similar type of reaction (Schwarzer and Marahiel, 2001). After the activation, the amino acid is taken by the PCP-domain as a thioester (Finking and Marahiel, 2004) at the terminal thiol group of the cofactor 4*'*-phosphopantethein (Figure S1b) (Schwarzer and Marahiel, 2001). The cofactor is bound on the conserved serine residue of the carrier post-translationally, and acts like a flexible "hand" allowing the bound amino acyl or peptidyl to "travel" from one catalytic site to another (Finking and Marahiel, 2004). The C-domains are responsible for elongation of the peptidyle chain (Lautru and Challis, 2004). They catalyze the peptide bond formation (Figure S1c) between the aminoacyl-S-PCP of the current module and the aminoacyl-S-PCP or the peptidyl-S-PCP from the precedent module (Finking and Marahiel, 2004). The Te-domain catalyzes the peptide release whether in the linear, cyclic or branched-cyclic form (Figure S1d) (Schwarzer and Marahiel, 2001).

Along with the previously mentioned obligatory domains, the NRPS modules can incorporate auxiliary domains which contribute to enhanced structural diversity of the final peptide. These domains can catalyze different reactions such as epimerization, methylation, cyclization, oxidation, etc. (Schwarzer and Marahiel, 2001).

2.3. Adenylation domains

By determining the crystal structure of phenylalanine activating A-domain in gramicidin S antibiotic production (GrsA) it was discovered that the polypeptide chain consists of two sub domains, the large N-terminal domain and the small C-terminal domain (Figure S2). Substrate recognition occurs by numerous hydrogen bonds between charged and polar groups, where most of the amino acid residues involved in substrate recognition are located on the large N-terminal domain. However, strongly conserved lysine residue located on the C-terminal domain is involved in two key interactions: one with amino acid substrate and other with adenosine. It fixes their positions at the active site and anchors the C-terminal domain in an appropriate conformation (Conti *et al.*, 1997).

Observing the existing A-domain’s crystal structure it was concluded that the A-domain acquires at least two conformations. Based on that, a model describing the action mechanism of the A-domain and conformational changes which the A-domain assumes during one reaction cycle of the D-alanin-D-alanyl carrier protein ligase (DltA) is provided (Figure S3). This action mechanism starts with the open conformation of DltA. In this conformation, a few interactions are present between the large N-terminal and small C-terminal domain and the conserved lysine residue is located near the active site. After the substrates (D-alanin, ATP and Mg^{2+}) “dock” on the active site, the domain assumes the adenilation conformation in which the small C-terminal domain is rotated due to the incurred interactions. In this conformation, the active site is closed and “protected” from the surrounding solution. The conserved lysine residue comes in proximity with the ATP molecule and enters the interaction. The D-alanyl adenylate creation and pyrophosphate release enable a new conformation acquisition. In this conformation the adenilation and the reverse reaction are now disabled. After the D-alanin transportation on the phosphopantethein carrier, with yet unknown mechanism, the A-domain acquires the open conformation again (Yonus *et al.*, 2008).

The amino acid binding site resembles a “pocket” with the entrance at the concave surface of the large N-terminal domain (Conti *et al.*, 1997). It is assumed that eight amino acid residues that “lie” within this “pocket” determine the A-domain specificity (Challis *et al.*, 2000). Comparing the amino acid residues which “lie” in the phenylalanine binding pocket in GrsA with the corresponding amino acid residues in other A-domains it is possible to define some general rules for substrate specificity determination (Stachelhaus *et al.*, 1999). Such analysis makes it possible to predict the substrate specificity of A-domains with unknown specificities. It is also possible to use this knowledge and perform a site-specific mutagenesis to acquire new compounds with new properties (Challis *et al.*, 2000). The binding site of phenylalanine in GrsA is demonstrated in the Figure S4.

2.4. A-domain substrate specificity prediction programs

There are several programs and program packages that predict natural product structure and/or identify gene clusters in DNA sequences which code for enzymes involved in natural product biosynthesis. Some of these tools and their functions are mentioned in Table 1.

Table 1. Programs and program packages which predict the chemical structure of natural products and/or identifies gene clusters which code for enzymes involved in natural product synthesis

Program	Function	Reference
SEARCHPKS	Predicts and analyses PKS domains in the polypeptide sequence	Yadav *et al.*, 2003
MAPSI	Contains methods for computer based analysis of PKS Type I gene clusters in genomic sequences	Hongseok *et al.*, 2009
Biogenerator	Simulates PKS manipulation and generates a virtual macrolyde library	Zotchev *et al.*, 2006
NRPS.predictor	Predicts the specificity of A-domains in NRPS	Rausch *et al.*, 2005
ClustScan	Program package for semi-automated annotation of modular biosynthetic gene clusters in *in silico* conditions. It is also capable of novel chemical structure prediction	Starcevic *et al.*, 2008
CLUSEAN	A tool for automated computer based analysis of bacterial gene clusters for secondary metabolite biosynthesis	Weber *et al.*, 2009
NP.searcher	Identifies and predicts gene clusters in DNA sequences that contain a genetic instruction for potential natural product. It also predicts the chemical structure synthesized by PKS, NRPS or hybrid NRPS-PKS clusters	Li *et al.*, 2009

2.5. Latent Semantic Indexing (LSI)

The main problem in most information retrieval techniques is synonymy and polysemy. This problems can cause that documents relevant to the query may not be retrieved, while the documents irrelevant to the query could be overrepresented. LSI bypasses this problem by discovering the latent semantic structure of information.

LSI is a two-mode factor analysis method. Unlike the traditional, one-mode factor analysis, which begins with a matrix of associations between all pairs of similar object (e.g. documents), the two-mode analysis begins with a rectangular matrix with different entities presented on the rows and columns. In the case of LSI, rows are presented as terms and columns are representing documents, so the matrix is called a matrix of terms and documents. The matrix of terms and documents is then being subjected to a process called singular value decomposition (SVD), which decomposes the matrix of documents and terms into other three matrices of a very particular form. These matrices can later be reduced in such a manner that it should always be possible to reconstruct the matrix of terms and documents without greater loos in fidelity. This new matrix of terms and documents is similar to the original matrix of terms and documents. It is not important to reconstruct the matrix of terms and documents perfectly, because the derived structure expresses what is reliable and important in the underlying use of terms as document referents.

The method can be used to evaluate pattern similarity of occurrence across the set of documents measuring differences between two terms, by comparing two row vectors in the matrix of document and terms. The similar procedure would be to compare two documents, except the fact that documents are represented by column vectors. To represent the query input, it is important to treat the “pseudo-document” (query) like a part of the original matrix of terms and documents. That means that the query must be processed with the same operations like columns in the original matrix of terms and documents (Deerwester *et al.*, 1990).

Such approach has great potential for application in biology. Some of the examples where LSI could be used would be gene comparison and categorization (Couto *et al.*, 2007), classification of proteins (Yuan *et al.*, 2005) or gene clustering by analysis of abstracts in MEDLINE (Homayouni *et al.*, 2005).

Technical details, as well as way of LSI and SVD implementation for this work are described in Materials and Methods.

2.6. Aims of the work

The aim of this work was to test if LSI is a suitable method to predict the A-domain substrate specificity. We also wanted to create a protein vectorization method which would be more accurate and robust than the existing ones.

As there is no statistical method for information clustering that satisfies our needs, we wanted to create a new clustering method which would allow better classification and would not be dependent on right input of the number of clusters optimal for data separation.

The novel clustering method should introduce a threshold value which can later be used to separate relevant data from irrelevant data. With this approach the method of substrate specificity prediction can become even more accurate and precise.

3. Materials and methods

3.1. Materials

3.1.1. Computers and operating systems

Computer used to perform thesis analyses was a work station with processor AMD Phenom™ II X6 1055T with six cores and 16 GB of working memory. Operating systems installed on the station were Ubuntu release 10.04 (lucid) with kernel Linux 2.6.32-30-server and GNOME 2.30.2 with Windows XP Professional on Oracle VM VirtualBox. Part of analyses and most of the thesis writing were performed on ASUS notebook with Intel® Core™ i5 M 460 @ 2.53 GHz processor, 4 GB of working memory and Windows 7 Professional installed.

3.1.2. Packages and tools used

3.1.2.1 MATLAB

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis and numeric computation. For this purpose, built-in mathematics functions for linear algebra and statistics as well as 2-D and 3-D functions for visualizing data were used.

MATLAB can be used in a wide range of applications. Specific toolboxes are collections of special-purpose MATLAB functions available separately. They extend the MATLAB environment to solve particular classes of problems in specific application areas.

Bioinformatics Toolbox offers computational molecular biologists and other research scientists an open and extensible environment in which to explore ideas, prototype new algorithms, and build applications in drug research, genetic engineering and other genomics and proteomics projects. With the Bioinformatics Toolbox, genomic, proteomic and gene expression file formats can be written and read. It also has sequence analysis tools including functions for pairwise and multiple sequence alignment (The Mathworks, 2011).

3.1.2.3. ClustalX2

Comparison or alignment of protein and/or DNA sequences is the basis of modern bioinformatics. It is used to study evolution and relationships between different organisms. Sequences can be aligned in two ways: globally, across their entire length, or locally, only in certain regions.

Clustal series of programs are the most popular and most commonly used programs for creating multiple sequence alignments. The multiple alignment is built up progressively by a series of pairwise alignments, following the branching order in a guide tree. ClustalW is the third generation of the Clustal series. It incorporates a number of improvements to the alignment algorithm, including improved sequence weighing, position-specific gap penalties and the automatic choice of suitable residue comparison matrix at each stage of the multiple alignment. ClustalX is the latest member of the Clustal series of programs with graphical user interface. The alignments produced with it are the same as those produced by ClustalW, but the program offers user friendly display of the multiple alignment in a scrollable window and guides users during the alignment creation process. ClustalW on the other hand is just a command prompt utility program (Chenna *et al.*, 2003).

ClustalW2 and ClustalX2 were rewritten in C++ to make the code easier to maintain and to make future modifications or replacements of alignment algorithms easier. ClustalX2 program has the same functionality as ClustalX, but new options are included to enable faster alignment of large data sets (Larkin *et al.*, 2007).

3.1.2.2. Similarity search

BLAST (**B**asic **L**ocal **A**lignment **S**earch **T**ool) is a sequence comparison tool with a simple and robust algorithm. It is possible to apply it in a wide range of applications such as straightforward DNA and protein sequence database search, more complex motif searches, gene identification searches and in the analyses of multiple regions of similarity in long DNA stretches. BLAST uses a heuristic approach and is order of magnitude faster than the sequence comparison tools with the comparable sensitivity (Altschul *et al.*, 1990). BLAST accomplishes this speed increase by implementing dynamic programming methods in algorithm design.

Program package HMMER implements profiles made from hidden Markov models (profile HMMs) to biological sequence analysis. Profile HMMs are manners for representing multiple sequence alignments as “profiles” using probabilistic models generally applicable to time series or linear sequences called hidden Markov models (HMMs). They are primarily used to build database search models from pre-existing alignments (Eddy, 1998). For the purpose of this work, HMMER was used to detect A-domains in the NRPS sequences using respective profiles created by the same program package from trusted multiple sequence alignments.

3.1.3. Protein sequences

The train sample consisted of 397 A-domains with known substrate specificities. For analyses purpose, four series of protein sequences representing each of 397 A-domains were created and referred to as: *complete sequences, active site sequences, 8 Å sequences* and *binding pocket sequences* respectively. *Complete sequences* imply the entire A-domaisn. Their length is 400 residues in average. *Active site sequences* represent A-domain regions containing only the active site. Their length varies from 136 to 150 residues. *8 Å sequences* refer to total of 34 residues found in proximity of up to 8 Å radius from the substrate binding site. *Binding pocket sequences* comprise of 10 residues that directly interact with the amino acid substrate. All sequences were written in commonly used FASTA format. Except for the sequence itself, this type of representation consists of header that precedes the sequence, includes readable sequence information and always starts with the symbol ">". Base pairs or amino acids are represented as single-letter codes and follow the header in a new line.

3.2. Methods

3.2.1. Protein sequences acquisition

Active site sequences, 8 Å sequences and binding pocket sequences with known substrate specificity were taken from Rausch and collaborators (2005) supplementary materials and written in FASTA format using a function from MATLAB Bioinformatics toolbox named `fastawrite`. Accession numbers were written in a separated document and NRPS sequences containing all 397 A-domains were downloaded using Entrez Batch utility. Complete A-domain sequences were detected and extracted from longer NRPS sequences using HMMER version 3.0 with A-domain HMM profile created by the same program package previously.

3.2.2. Substrate specificity identification

Identification of substrate specificity for *complete sequences* was accomplished using local BLAST+ application. For this purpose BLAST+ version 2.2.17 with default settings together with function from MATLAB Bioinformatics toolbox called `blastlocal,` were used. As a database for the local BLAST implementation, *complete sequences* were formatted with BLAST *formatdb* tool and then utilized. As a query, *active site sequences* were used. Substrate specificity of the sequences used as a query was assigned based on the best scoring *complete sequence* from the database and written in the FASTA format header of the respective *complete sequence*.

3.2.3. Term-document matrix construction

Term-document matrix (called "A matrix" later in text) was constructed using two different methods of protein sequence vectorization. As documents for constructing the A matrix, A-domain sequences were employed in both methods. The methods mainly differ in terms being used to construct the A matrix. The first method is well known and utilizes *n*-peptides (*n*-grams) as terms for matrix construction. The second is a novel method we developed for this thesis, called protein sequence tokenization. This method uses the so called, "tokens" as terms for document search, described latter in the text. The A matrix cell values were calculated using weight factors. The method to calculate weight factors is given by Equation 3-1, where f_{ij}, referred as local factor corresponds to incidence frequency of the term *i* in the document *j* where *D* is a number of documents and d_i, referred as global factor, is an incidence frequency of the term *i* in all documents (Garcia, 2006). A matrix is an *m* × *n* size matrix, where *m* represents the number of terms and *n* represents the number of documents or in our case protein sequences being used.

$$a_{ij} = f_{ij} \cdot \log\left(\frac{D}{d_i}\right) \qquad \text{Equation 3-1}$$

3.2.3.1. n-peptides (n-grams)

The expression *n*-gram refers to a subsequence of length *n* in a given sequence. In the case of protein sequences, the *n*-grams can be referred as *n*-peptides. The first step in A matrix construction is the construction of all possible *n*-peptides. As the number of proteinogenic amino acids is twenty, the number of all possible *n*-peptides is 20^n. The *n*-peptides of length 1 are called unipeptides, of length 2 dipeptides and of length 3 tripeptides, respectively. All possible combinations of *n*-peptides, *n* ranging from 1 to 3, were constructed with the self-written MATLAB script. The A matrix cell a_{ij} was constructed using Equation 3-1, where local factor represent incidence frequency of *n*-peptide *i* in the A-domain sequence *j*, and global factor is incidence frequency of *n*-peptide *i* in all A-domain sequences (Couto *et al.*, 2007).

3.2.3.2. Protein sequence tokenization

The novel protein sequence tokenization method employs specific amino acid residues on the specific places in the multiple sequence alignment as terms for A matrix construction. Amino acid residues coupled with their position in the multiple alignment are called “tokens”. For example, amino acid residue A from column 17, in multiple sequence alignment, would become token “A17” in our proposed tokenization method. To retrieve all possible “tokens” for the A matrix construction, the A-domain sequences obviously must be aligned first. *Complete sequences* and *active site sequences* were aligned using ClustalX2 version 2.1 with default settings. Tokens were constructed by method described above, where amino acid residues were combined with respective positions (columns) in multiple sequence. Tokens created in this way were recorded as a list. From the list containing all tokens, unique tokens were extracted and a new list was created containing only single occurrences of tokens. The number of tokens in this list corresponds to a number of terms used to create the A matrix (m). When computing A matrix weighed factors, local factors could obtain one of two possible values. If the amino acid residue of the sequence j on the position assigned to the token i, was the same as the amino acid residue assigned to the token i, the local factor f_{ij} was 1. In other case, when residues mismatched, it was 0. The global factor d_i represents the number of occurrences of given token in all protein sequences being used for A matrix construction.

3.2.4. Singular value decomposition

Singular value decomposition (SVD) analysis was performed using built-in MATLAB functions, `svd` and `svds`. Function `svd` calculates all the singular values of a given matrix. Function `svds` calculates only the k largest singular values and k must also be entered as input. The products of the SVD are three new matrices that must satisfy the Equation 3-2.

$$A = USV' \qquad \text{Equation 3-2}$$

The parameter k is the number of singular values used to reconstruct the A matrix. The choice of the parameter “k” was one of critical points for the success of this thesis work. The parameter k must be large enough that all relevant data fits in the reconstructed matrix, but also small enough to lose background noise from the data sample. The parameter k was obtained computationally, and compared to the one obtained-empirically. The optimal value of parameter k was obtained by calculating the relative variance of every singular value in the

S matrix. The formula for calculating the relative variance of singular values is given by Equation 3-3. v_i is a relative variance of the singular value S_i from *r* singular values in the A matrix.

$$v_i = \frac{S_i^2}{\sum_{j=1}^{r} S_j^2}; j = 1,2,3 \dots r \qquad \text{Equation 3-3}$$

Singular values in which relative variance is less than 0.7/n, where n is the number of protein sequences in the A matrix, are considered insignificant. After calculating the parameter k, all three matrices resulting from SVD were reduced in the way that fulfils the requirement given by the equation 3-4 (Couto *et al.*, 2007).

$$A \approx A_k = U_k S_k V_k{}' \qquad \text{Equation 3-4}$$

3.2.5. Query vector construction

Query vectors (q) were calculated for the A-domain sequences for which substrate specificity was to be determined. The query vector was computed in the same manner as the protein vectors in the A matrix, using global factors obtained from A matrix construction method. The Equation 3-5 is applied to gain vectors compatible with the protein vectors from the A matrix (Garcia, 2006). The method described above is commonly known as folding-in.

$$q = q' U_k S_k^{-1} \qquad \text{Equation 3-5}$$

3.2.6. Computing cosine values and specificity assignment/prediction

There are several ways to compute the similarity between query vectors and protein vectors from the A matrix. The most frequently used are the cosine value between vectors and the Euclidean distance. For this purpose, cosine value of the angle between two vectors was used. It is computed as a scalar product between two vectors following the Equation 3-6 (Garcia, 2006).

$$\cos\theta = \frac{q \cdot d}{|q||d|}, d = V_k{}'_i, i = 1,2,3, \dots, n \qquad \text{Equation 3-6}$$

The substrate specificity prediction for the query sequence was assigned based on the principle of computing a probability coefficient for each amino acid substrate present in the train sample. The amino acid substrate with the highest coefficient is assigned to the query sequence as its substrate specificity. Cosine value between the query vector and each protein vector from the A matrix was calculated. Estimated cosine values were sorted in descending order, from the most likely predictions to the unlikely ones. Sorted cosine values were then indexed in increasing order starting with 1. Coefficients for each amino acid substrate were evaluated by the Equation 3-7, where z_i is the coefficient of the amino acid substrate i, $\cos\theta_i$ is the cosine value between the protein vector from the A matrix with the specificity *i* and the query vector and *j* is the index of the $\cos\theta_i$.

$$z_i = \sum_{j=1}^{n} \frac{\cos\theta_i}{j} \qquad \text{Equation 3-7}$$

3.2.7. Method precision determination (Determination of method precision)

To evaluate the predictions being made by this procedure, a method called leave-one-out (*loo*) test was applied (Rausch *et al.*, 2005). The *loo* test consists of using one protein sequence as a query sequence and all the rest as train sequences for A matrix construction. The *loo* test was performed for all the sequences for which specific substrate is present, determined and occurs more than once in the train sample. Precision is measured by comparing the predicted query sequence substrate with the actual query sequence substrate. Precision is calculated as a percentage. This percentage is described as ratio of the number of correctly assigned substrate specificities and the total number of analyzed sequences multiplied by one hundred respectively.

3.2.8. Novel clustering method

A novel clustering method was also developed. Method strongly relies on the following steps: A matrix construction, SVD analysis and cosine value calculation between all analyzed sequences as previously described. The product of this analysis is $n \times n$ sized matrix (V’V matrix) where n is the number of analyzed sequences. All the cosine values from the V’V matrix were sorted in two arrays. The first array consisted of all cosine values between A-domain sequences with the same substrate specificity (matching array) and the other list consisted of all other non-matching A-domain sequences (non-matching array). Analyzing the obtained arrays, a threshold was defined so that the ratio of the matching array members above the threshold value would be maximized compared to the number of the non-matching array members below the threshold value. All members of the same cluster in this novel method would then have a cosine value greater than the threshold value.

4. Results

4.1. Tokenization of protein sequences

To estimate efficacy of each tokenization method for the A matrix construction, LSI analyses were also performed with n-peptides method using unipeptides, dipeptides and tripeptides as term~~s~~ for A matrix construction. All methods were evaluated using *loo* test. LSI analyses were carried out using *complete sequences*, *active site sequences*, *8 Å sequences* and *binding pocket sequences* to estimate the influence of sequence length on the precision of the method. Precisions of each method depending on sequence length are shown in Figure 1.

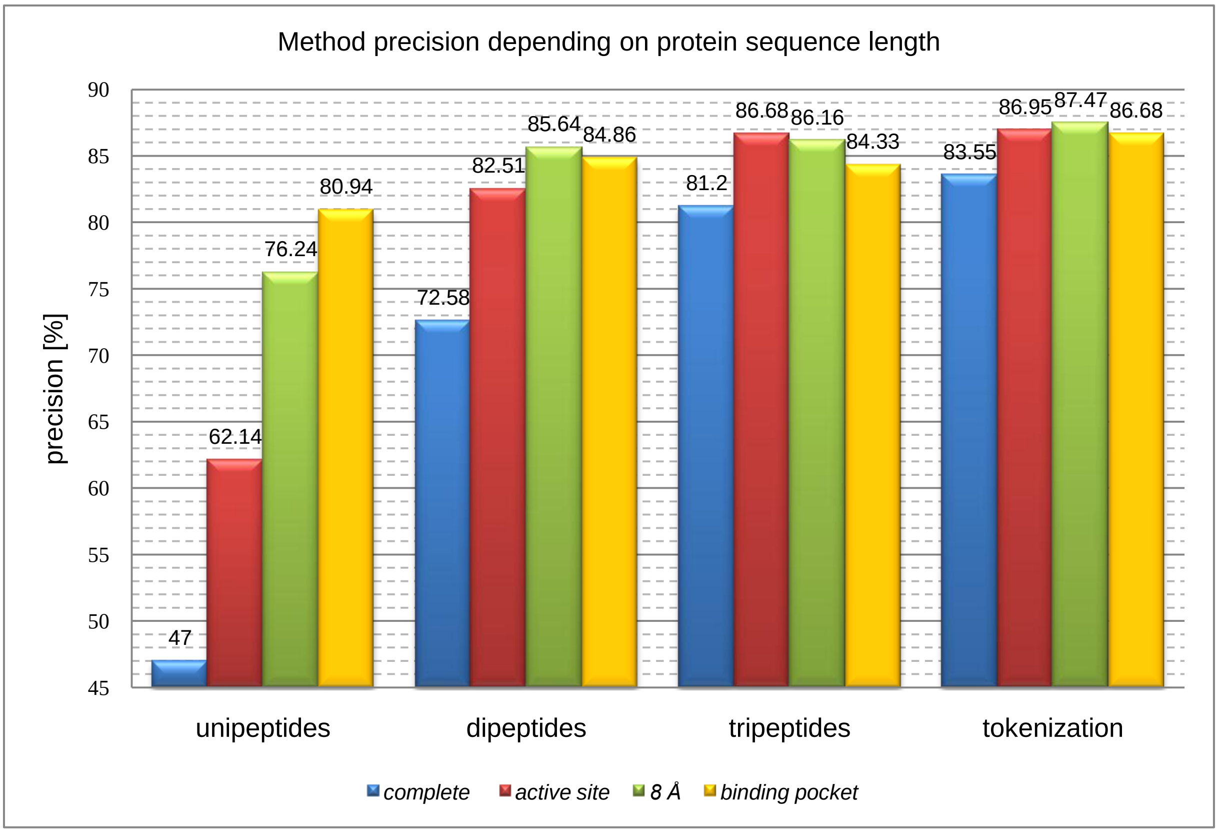

Figure 1. The effect of tokenization method and sequence length on LSI substrate prediction

The unipeptide method showed precision in the range of 47 to 80.94%. The lowest precision was gained by analyzing *complete sequences* and the highest precision was gained by analyzing *binding pocket sequences*. Precisions for the dipeptide method varied from 72.58% for *complete sequences* to 85.64% for *8 Å sequences*. The tripeptide method precisions covered the interval from 81.2% to 86.68%. The maximal precision value was gained when *active site sequences* were used for substrate determination, and the minimal precision value was gained by analyzing *complete sequences*. The precision of our novel substrate specificity determination method range from 83.55% for *complete sequences* to 87.47% for *8 Å sequences*.

4.2. The choice of factor *k* for optimal dimension reduction

In this work, to estimate the optimal number of dimensions needed for the A matrix reconstruction, a computational method was used as described previously. It was necessary to compare the calculated factor *k* with the empirically assigned factor *k*. Because it was not possible to trust the empirical estimate of the optimal factor *k* value, the same analysis was performed with factors *k* ranging from the minimal possible value which is 2 to the maximal value of 200 which was computationally feasible. The analysis consisted of taking 200 randomly selected A-domain sequences as a train sample and the rest (197 A-domain sequences) as a test sample. The train sample was used to create the A matrix which was then used as knowledge base for the members of the test sample LSI method substrate prediction. Predictions made in this way were compared with the actual substrate specificities in order to obtain method precision. For this analysis, *active site* sequences were used. The results of the analysis are shown on Figure 2.

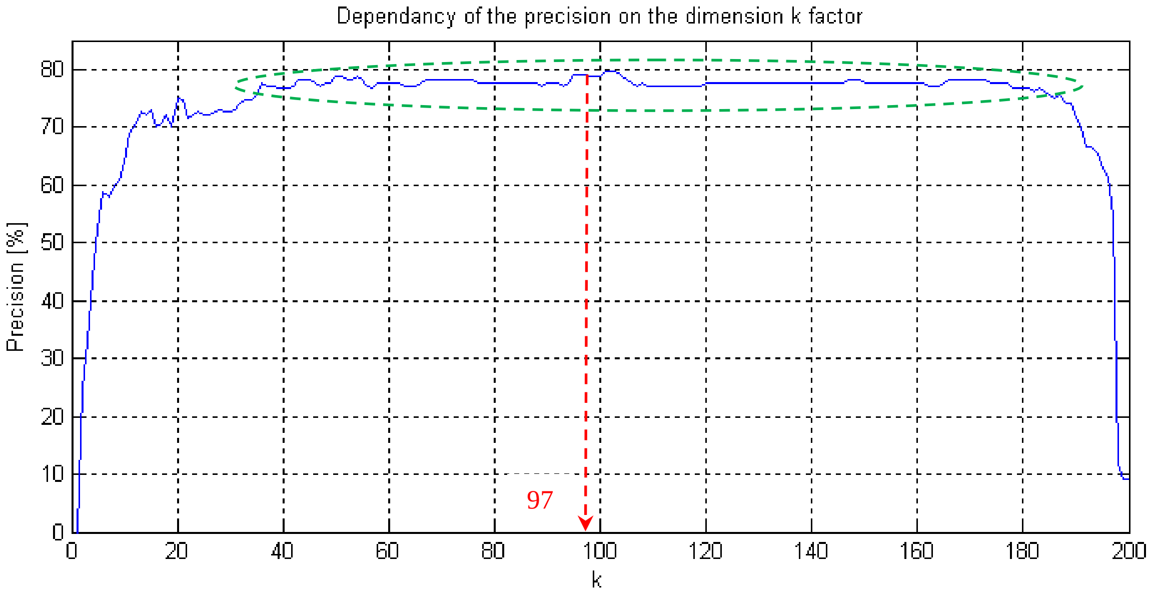

Figure 2. Dependency of the dimension reduction (factor *k*) on the method precision. The area marked in green represents the plateau area. The red line represents the value of the factor k computationally gained with the Equation 3-3.

From Figure 2, it can be seen that the dependency of the dimension reduction on the method precision curve has three distinct phases. The first phase is a growing phase. In this phase the method precision is rapidly growing with the factor *k*. The method precision has grown from approximately 20% gained with factor k = 2 to approximately 70% for the factor k = 20. For the values of factor *k* ranging from 20 to 40 the method precision growth is beginning to slow down. In the second phase, for the factor *k* from 40 to 180 the method precision value is approximately constant (plateau, around 78% precision) and it can be seen as a green tagged area on Figure 2. After the factor *k* value reaches 180 dimensions, the curve enters in the third, declining phase where the method precision value is rapidly dropping with the factor *k* growth. From the Figure 2 it can be seen that the maximal value of the dependency curve is around factor k ≈ 100 which is mentioned in literature as rule of thumb value (Deerwester *et al.*, 1990). The optimal factor *k*, gained computationally, is marked with the red line on Figure 2. Its value is 97, which is remarkably close.

4.3. Determining cosine threshold

To determine optimal cosine threshold, protein vectors were constructed using tokens as terms and *binding pocket sequences* of all A-domains in the training sample as documents. SVD analysis was performed and cosine value between all protein vectors was calculated. Matching array and non-matching array were obtained as previously explained. Matching array and non-matching array are shown in Figure 3.

Obtained arrays were utilized to choose the possible cosine threshold. The optimal cosine threshold needs to satisfy the requirement that there are more matching array members than non-matching array members. As it can be seen in Figure 3, the initial cosine threshold is 0.165 (red line) as this is the smallest cosine value that fulfils the specified requirement. From Figure 3 it can be clearly seen that many of the matching array members are above the initial threshold (Figure 3a, red area), 1541 out of 3256 respectively. Most of the non-matching array members are below the minimal cosine threshold (Figure 3b, red area), 73 811 out of 75 350 respectively. Choosing the optimal cosine threshold, among the cosine values greater than 0.165, will be discussed in the next paragraph. Members of the matching array above the cosine threshold are called true positives (tp), and the ones below the cosine threshold false negatives (fn). The non-matching array members below the cosine threshold are called true negatives (tn), and the ones above the cosine threshold false positives (fp). This classification for the minimal cosine threshold value is labelled in Figure 3.

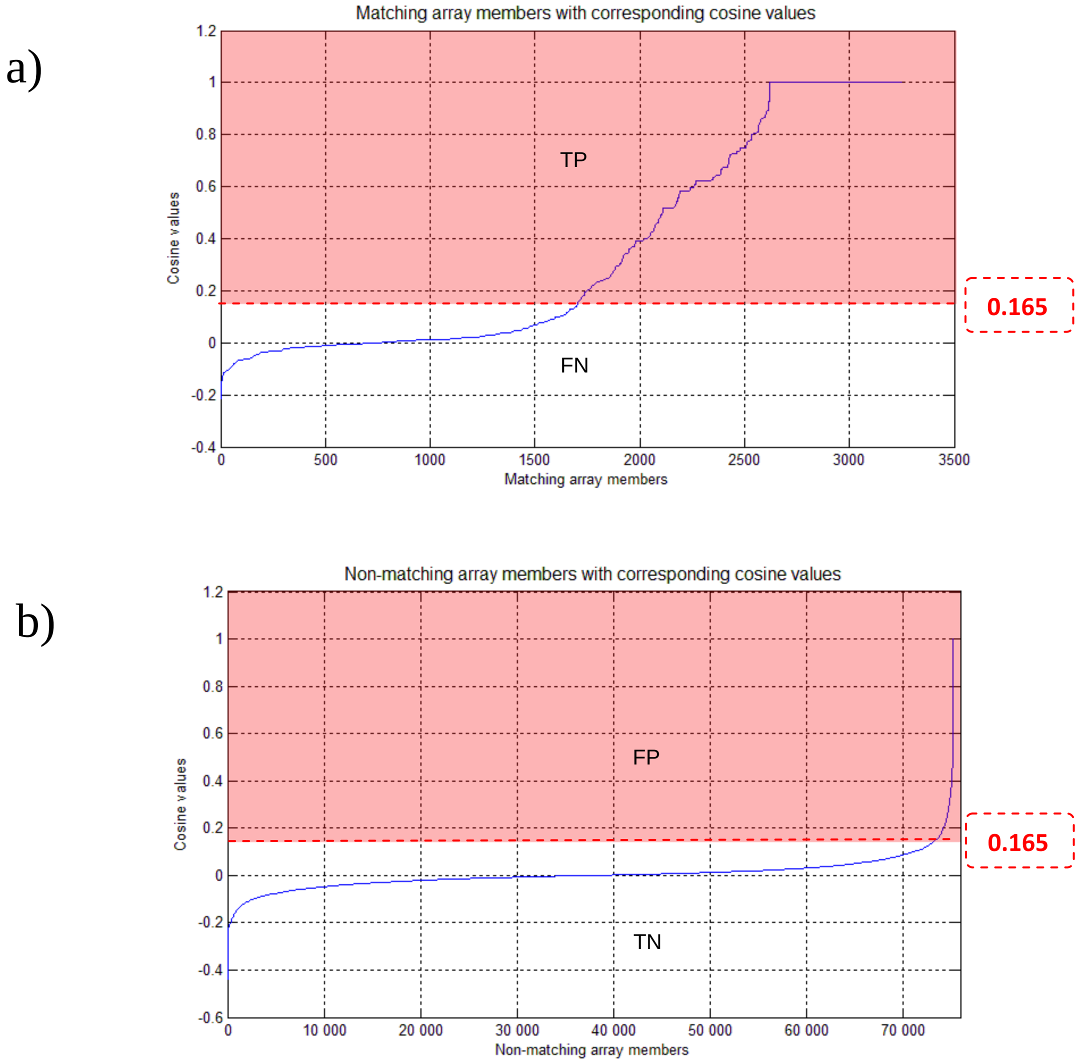

Figure 3. Arrays used to determine cosine threshold value: a) matching array and b) non-matching array. TP = true positives, FN = false negatives, FP = false positives, TN = true negatives.

Observing the matching array and the non-matching array, it was possible to calculate two more parameters: sensitivity and specificity of the method. For this purpose the specificity can be described as a fraction of relevant retrieved information compared to all retrieved information, and can be calculated by the Equation 4-1.

$$Specificity = \frac{tp}{tp + fp} \qquad \text{Equation 4-1}$$

Sensitivity is described as a fraction of retrieved relevant information compared to all relevant information. It is given by Equation 4-2.

$$Sensitivity = \frac{tp}{tp + fn} \qquad \text{Equation 4-2}$$

Specificity and sensitivity were calculated using cosine value threshold in the interval ranging from 0.165 to 1. The results are shown in Figure 4. The sensitivity is falling linearly with the growth of the cosine value threshold. The gained results can be approximated with the regression line represented by equation:

$$y = -0.3727x + 0.5294,$$

with the R-squared value $R^2 = 0.9831$ indicating almost perfect linearity. Specificity is growing with the cosine threshold until it reaches a constant value around 0.9 at the cosine threshold value of 0.41.

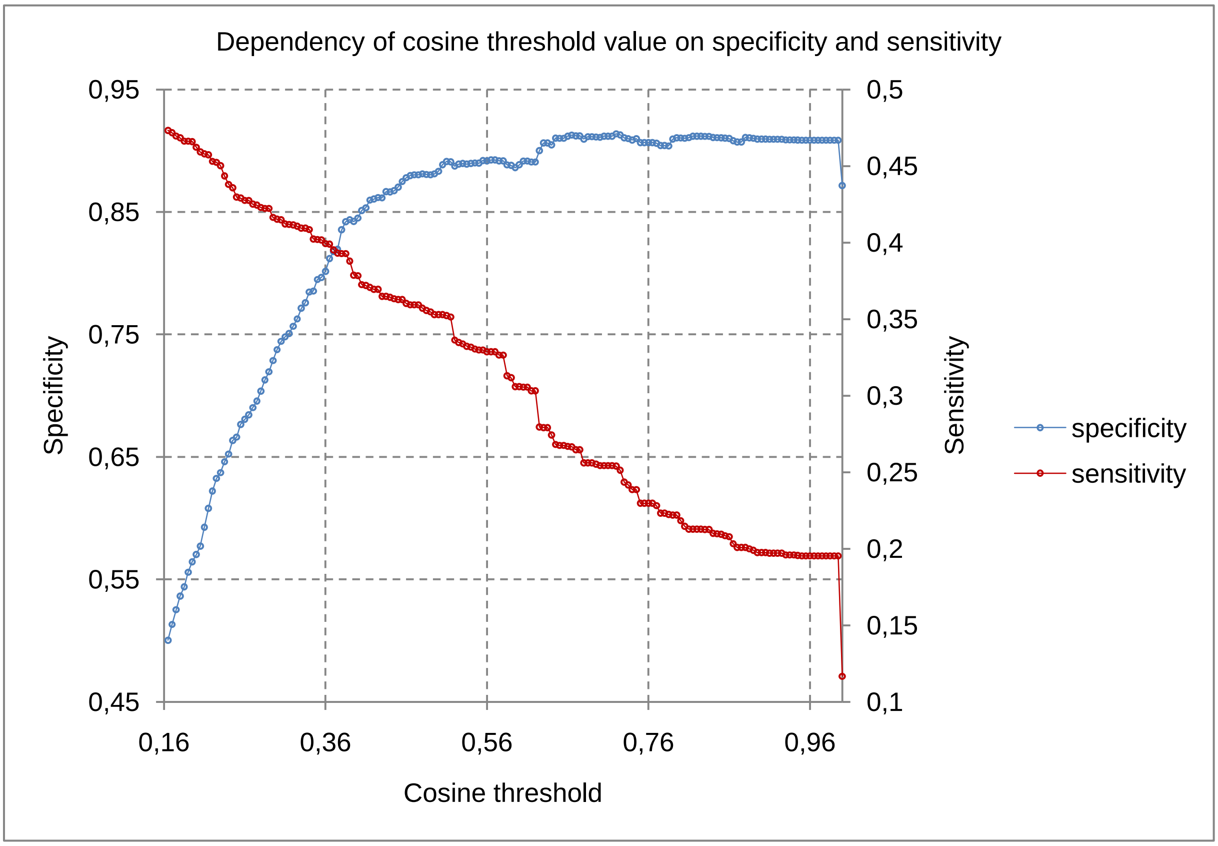

Figure 4. Dependency of the method specificity and sensitivity on the cosine threshold value

4.4. Novel clustering method

To group the A-domains based on their substrate specificity, a novel clustering method was developed as previously described. To perform a clustering analysis, a dividing threshold must be chosen. As shown in the previous chapter the minimal possible threshold value is 0.165. To estimate the optimal threshold value which would lead to correct number of clusters, dependency of cosine threshold used the on the number of clusters (Figure 5) as well as the influence of substrate abundance in data sample on the number of clusters (Figure 6) were observed and compared to actual data.

From Figure 5, it can be seen that the cluster number grows linearly with the cosine threshold value (Figure 5, blue markers). Thus, it is possible to trace a trend line and show the dataset as a function. The initial cluster for the minimal cosine threshold is 37, and the cluster number for the cosine threshold of 0.995 is 192. The linear function equation was calculated and it was:

$$y = 200.13x + 8.3313,$$

with R-squared value $R^2 = 0.9885$, respectively. The obtained function was compared with the theoretical one in which the initial cluster number for the cosine threshold 0 would be 1, meaning all sequences representing single substrate category, and the cluster number for the cosine threshold 1 would be 397 meaning every sequence would be in its own cluster (Figure 5, green markers). The equation of such theoretical function would be:

$$y = 396x + 1.$$

Comparing these two functions it can be noted that the cosine of the angles between them is approximately 1. There is a difference between these two function’s slope and *y*-intercepts. The slope of the theoretical function is almost two folds greater than the empirically gained function slope. The *y*-intercept of the theoretical straight line is 1, whereas it is 8.3313 for the empirically gained straight line.

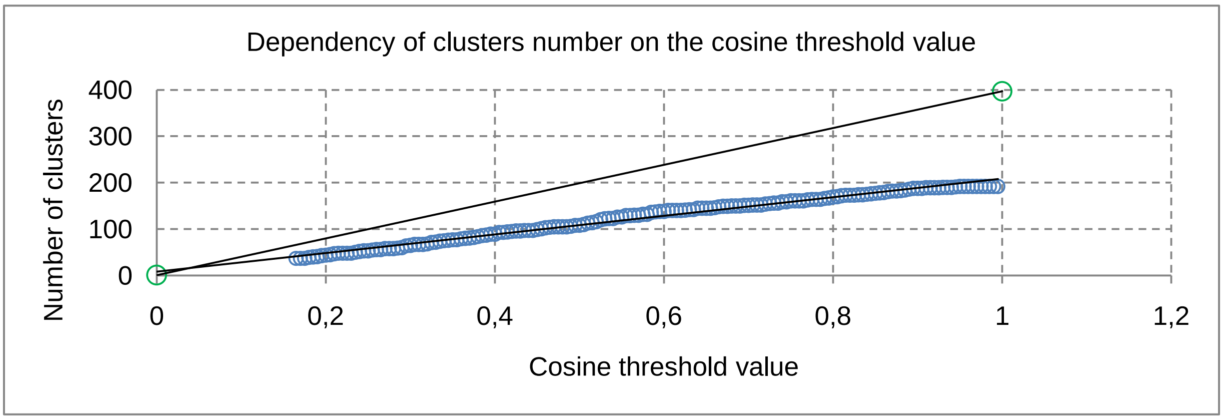

Figure 5. Dependency of cosine threshold value on the cluster number. Blue markers represent the real dataset. Green markers show the theoretical dataset.

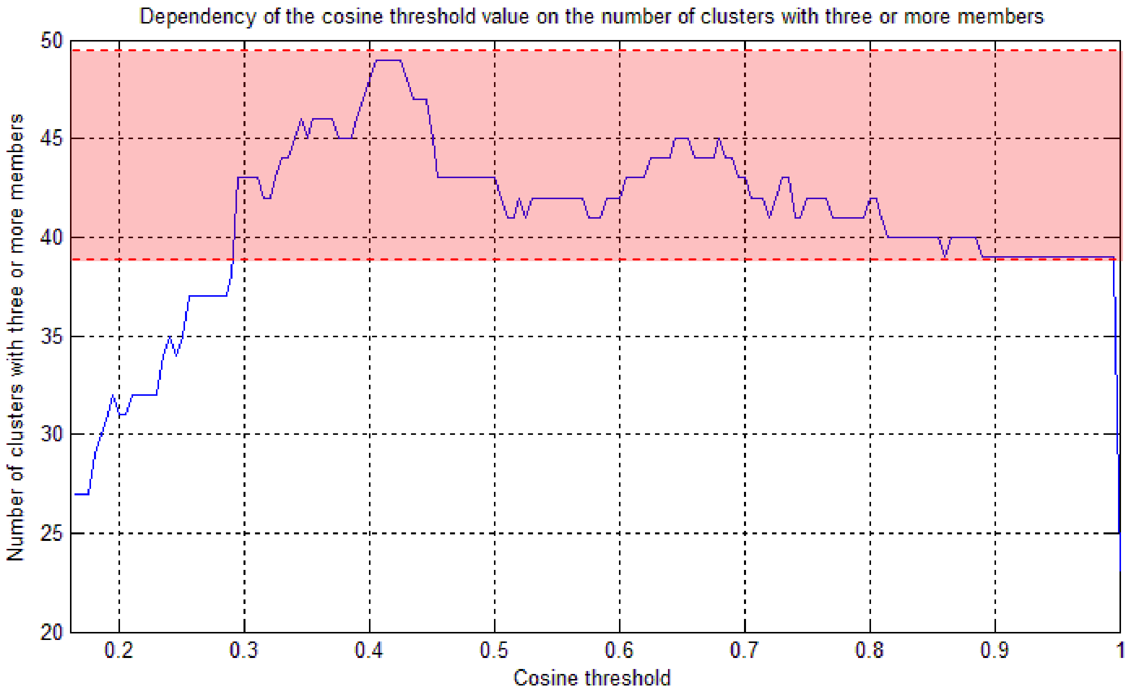

Figure 6. Dependency of the cosine threshold value on the number of clusters with three or more members. The red area represent the interval between 39 an 49

Figure 6 shows the dependency of the cosine threshold value on the clusters with three or more members. The curve is growing with the cosine threshold value until it reaches its plateau at the cosine thresholds around 0.41. After reaching the plateau, the curve has a falling trend. The curve reaches a constant value 39 and sharply falls at the end. For most of the cosine thresholds, number of clusters with three or more members fit into the interval between 39 and 49. After extracting these elements from the curve, their median value was calculated as 42.

Figure 7 visualizes clusters obtained using the minimal cosine threshold value. The obtained number of clusters using this threshold is 37 of which 27 have three or more members. In 13 clusters all cluster members have the same substrate specificity. There are 4 clusters with just one member. The largest cluster has 61 members in it, mostly A-domains which activate predominantly hydrophobic amino acids.

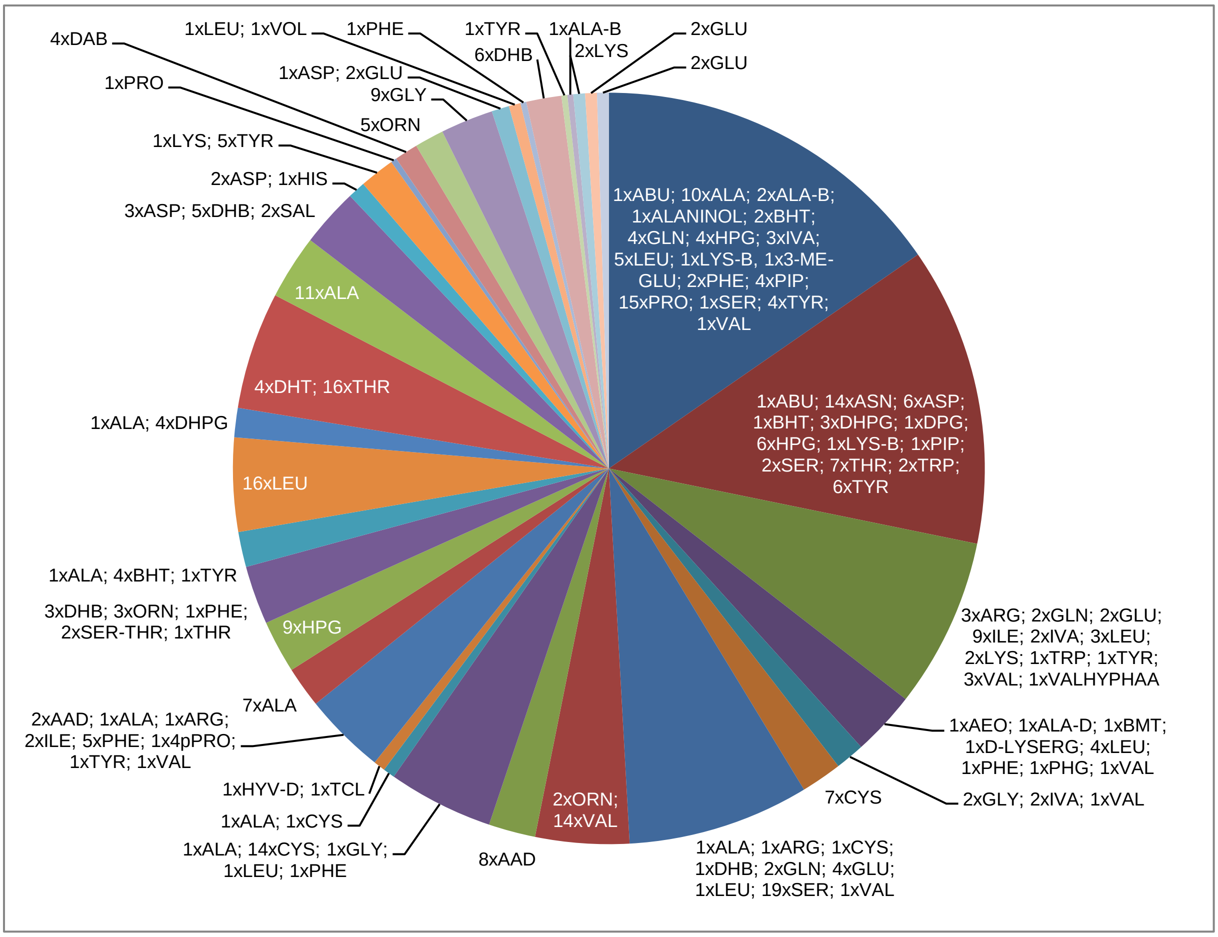

Figure 7. A-domain clusters obtained using the minimal cosine threshold values with corresponding A-domain members. Abbreviations of amino acids are shown in annex.

5. Discussion

In the last few years several methods were developed for A-domain substrate specificity prediction based on analysis of the protein sequences (Stachelhaus *et al.*, 1999; Challis *et al.*, 2000; Rausch *et al.*, 2005). These methods were derived from the fact that phylogenic analysis of A-domain sequences, showed grouping based on the substrate specificity and not on the genetic origin (Stachelhaus *et al.*, 1999). So far, most of the methods were based on analysis of the amino acid residues which take part in substrate binding or are located in the substrate binding area. For this to be successful, a lot of effort needs to be introduced in active site analysis which can be done only in the laboratory. In this work whole A-domain sequences without prior active site knowledge were used for substrate analyses, with comparable success rate to the active site regions. Because of the fact that the same A-domain can be indicative in activation of several different amino acids, some of the previous methods (for example, the one used in NRPS.predictor) are constricted to prediction of a group of amino acids based on their physico-chemical properties and not based on the actual amino acid which the A-domain predominantly activates (Rausch *et al.*, 2005). Our method can predict actual amino acid substrate with a satisfactory precision without any prior knowledge outside the training data set, and without any active site knowledge.

In the LSI performance, the key step is the protein vectorization method. For this purpose, a method called “sliding window” or “*n*-gram” and a novel method called “tokenization” were used and compared. The unipeptides method, as expected, showed the worst results both in robustness and precision. Tripeptides have shown better robustness and precision with respect to dipeptides and unipeptides. This implies that this method is less dependent on the length of the protein sequence. From the obtained results, it can be assumed that using *n*-grams with more amino acids will yield better results. But as the computational resources are limited, computational time becomes too much of an issue and the A matrix gets too large for practical purposes.

In the case of tripeptides, the method precision has grown when analyzing longer protein sequences up to a certain level. This can be explained by the fact that the method did not analyse just a specific amino acid residue, but also surrounding residues, which have thus shown to “contain” valuable information. This was not observed with *complete sequences*. The assumption was that *complete sequences* yielded worse results because other features other than substrate specificity can interfere. Dipeptides method showed similar behavior with the exception of *active site sequences* which have shown lesser precision than *8 Å sequences*. This can be explained with higher random probability occurrence of dipeptides in the protein sequence. The unipeptides method showed reduction of precision with the extension of the protein sequence. That is in accordance with the fact that unipeptides method calculates the amino acid frequency in the protein which varies more in longer sequences.

The novel tokenization method consists of composite indexes from the multiple alignment, as previously described. It shows greater robustness and precision than *n*-gram methods. The greater precision results from the fact that the method gives greater significance to variable positions which are responsible for different specificity. Greater robustness can be explained by the fact that longer sequences do not differ significantly between them because regions among binding pocket residues are more conserved than binding pocket residues themselves and the method gives tends to score these residues lower. The method is also faster than *n*-gram methods because it creates "tokens" from the pre-existing multiple alignment and it makes the whole process less computationally intense. The limitation of the method is dependency on the multiple alignment – to be more specific, on the quality of the multiple alignment. This limitation can be observed when comparing *complete sequence* analysis result with other sequence analysis results. The difference in precision is attributed to the lower quality of multiple alignment of *complete sequences*.

The number of values used to reconstruct the A matrix (factor *k*) is critical. As mentioned previously, the factor *k* must be large enough to fit all relevant data in the reconstructed matrix, but also small enough to remove all "background noise" and irrelevant information. The number of dimension which starts to introduce too much information (noise) is around 185 as for this factor *k* values of the method precision starts to fall (see Figure 2). The number of dimensions at which the reconstructed matrix starts to lose relevant data is below 40. This can be deduced from the fact that up to this value the method precision is sharply growing with the factor *k* value. At the values of factor *k* between 40 and 185 the precision value reaches a plateau. This means that using factor *k* values from this interval is optimal to fit all relevant data in the reconstructed matrix and to exclude all irrelevant information from the reconstructed matrix. It can be seen that method precision reaches its optimal value using factor *k* starting with 40, which is close to the number of substrate specificities in the train sample (47). This is in accordance with our assumption that the number of specific substrates coincide with the number of relevant information in the A matrix. The method precision value reaches its maximum around the value $k \approx 100$. This value is in accordance with literature, where the optimal number of dimensions to reconstruct the document-term matrix is between 100 and 300 (Deerwester *et al.*, 1990). As it was not possible to evaluate the optimal factor *k* empirically, a computational method was used. The computational method produced a factor *k* value of 97. This value was close enough to the optimal value and obtained precision of the method was large enough to regard the entire method as satisfactory.

To determine the cosine threshold, LSI analyses were performed on the train sample of 397 A-domain sequences, as described previously. Matching array and non-marching array were obtained. The analyses were performed by vectorizing the *binding pocket sequences* using the tokenization method. These sequences and method were used because they produced the most suitable matching array and non-matching array for cosine threshold determination. As the minimal cosine threshold value was chosen, being the value in respect to which more matching array members than the non-matching array members were above the threshold value. The aim of choosing the optimal cosine threshold was to create protein clusters with members having the same substrate specificity. With the previously mentioned requirement the number of members with different substrate specificity in the same protein cluster is minimized. The cosine value 0.165 is the smallest value that fulfills this requirement. As the cosine threshold value grows, less non-matching array members would be above the cosine threshold value. Unfortunately, this also applies to the matching array members. The ideal case would be that all matching array members are above the cosine threshold value and all non-matching array members are below the cosine threshold value, but the real data do not show such behavior and such cosine threshold doesn’t exist.

To determine the optimal cosine value two parameters were calculated, sensitivity and specificity, as previously described. Sensitivity is the ratio of matching array members above the cosine threshold value in respect to all members above the cosine threshold value. Specificity is the ratio of matching array members above the cosine threshold value in respect to all matching array members. As it can be seen from Figure 4, the sensitivity curve is falling linearly, while the specificity curve is not. This phenomenon makes it possible to determine the optimal cosine value, and it was the value at which the specificity curve started to stagnate, which was 0.41 respectively. At this cosine threshold value, approximately 85% of values above the threshold are matching array members, but unfortunately, only 37% of total matching array members number are found above the threshold. Observing greater cosine threshold values it can be seen that sensitivity of those values is lower but the specificity is not significantly higher. The specificity of the initial cosine threshold value was around 0.5, as expected. The reason for this is because the initial cosine threshold value is the first value that fulfills the main requirement and it is obvious that the ratio of matching array members, and non-matching array members above the threshold value would be close to 1. An interesting coincidence is that the sensitivity for this cosine threshold is also around 0.5. The relation between these two numbers still has to be studied.

All existing statistical clustering methods require the input of empirical number of clusters to perform the clustering analysis. In this case we actually wanted to predict the number of clusters and test the prediction by real data comparison. To accomplish the task, it was necessary to develop a novel clustering method which prosuces the number of clusters as result and doesn’t just test empirical values. To estimate the cosine threshold value which would result in the optimal clusterization, clustering analysis was performed with different cosine threshold values. The threshold values used were in interval from 0.165, which is the minimal cosine threshold value, to 1, which is the maximal cosine value. Number of clusters obtained has grown linearly with the cosine threshold values (see Figure 5). Moreover, to estimate the empirically obtained regression function, a theoretical regression function was constructed. It was based on the theory that at cosine threshold value of 0, all protein vectors would be classified in single large cluster. Using the cosine threshold value 1, each A-domain would be clustered as an individual cluster. Comparing the theoretical line with the empirical one, it can be seen that they are very similar because the cosine value between them is approximately 1. However, there are some differences between these two functions. At the cosine threshold value 0, the empirical line *y*-intercept is not 1 as in theoretical line, but as can be seen from its equation, the *y*-intercept value is around 8. It is interesting to state that this is also the number of groups at which amino acids can be classified by their physico-chemical properties (Rausch *et al.*, 2005). The empirical line does not go beyond the value of 192. This number can be assigned to maximal number of possible clusters in A-domain grouping. This could be explained by the fact that, in theory, all A-domains would be separated in singular different clusters, whereas empirically, A-domains separate in only so many clusters, as there are different substrates. A-domains which activate amino acids with similar properties group together in larger cluster respectively.

Another parameter for the evaluation of quality of the clustering method was estimated. This parameter is the number of clusters with three or more members in dependence on the cosine threshold value. As expected, at small cosine threshold values, the curve has grown with the threshold. It reached its maximum and then started to fall. This happened because at small cosine threshold values A-domain groups separate and fall out from the large clusters. At the maximal value, all A-domain should be grouped with other A-domains with the same substrate specificity. After the curve reaches its maximum, A-domains with the same substrate specificity start to separate from one another and start to form new smaller clusters. This theory can be confirmed by the fact that the number of single member clusters is shown to be growing, whereas the number of clusters with three or more members is falling. It can be noticed that after reaching its maximum, the curve falls slowly. It reaches a constant value at the number of clusters with three or more members at 39. As the majority of the curve falls in range between 39 and 49, the median was calculated for this interval. This value was used as a rough approximation of the number of substrates activated by A-domains. This number is not significantly different from the real number of specific substrates in the train sample (47). The assumption is that this novel clustering method can be used to roughly predict the number of substrates that activate all known A-domains. In literature this is estimated to be around 400 (Caboche *et al.*, 2008), but it would be interesting to test this number.

To visualize the clustering method, clustering of A-domains was performed using the minimal cosine threshold value. Clustering is assumed to be successful. There are 13 clusters out of 37 that have more than one member, in which all members activate the same amino acid. Only 4 clusters have just one A-domain in it. This indicates that there were cases of proteins too divergent from the rest of the data sample, for which vector representation failed to give cosine values above threshold. A-domains clustered together which do not activate the same amino acid usually activate amino acids with similar physico-chemical properties.

6. Conclusions

Following the results of undertaken analyses, several conclusions can be deduced:

- Using LSI, it is possible to predict the amino acid substrate that activates the A-domain by comparing it with A-domains of known substrate specificities.
- A new protein tokenization method was developed which represent proteins as vectors using existing multiple sequence alignment. The method yielded better accuracy and precision compared with existing protein vectorization methods.
- A new protein clustering method was developed based on LSI analysis and cosine threshold value determination. The novel clustering method shows satisfactory protein clustering results, and the obtained cosine threshold value can be used to improve the performance of LSI based protein specificity prediction method.

7. Thanks

First of all, I would like to thank the person who guided me trough my stage. The person who took me into the world of bioinformatics. The person who altruistically shared with me his knowledge and expirience, and stimulated my creativity. I would like to thank my mentor, Antonio Starčević.

I would also like to thank all the staff of the Section for Bioinformatic, especially Jurica Žučko, for their patience and good advices. Without them Iwould never be able to do my job.

I would like to thank all my friend who believed in me and supported me with good advices and moral support. I would also like to thank all the people who gave me the opportunity to enrol this program.

At the end, I want to thank my family: my mother, father, sister and niece. They gave a meaning to my life. Without them I would never be able to finish my studies. They have been an enormous support in difficult moments, and even if I tough there is no way out, they believed in me.

8. References

Altschul S. F., Gish W., Miller W., Myers E. W., and Lipman D. J. (1990) Basic local alignment search tool. *J Mol Biol* **215(3)**: 403-410.

Austin M. B., and Noel J. P. (2003) The chalcone synthase superfamily of type III polyketide synthases. *Nat Prod Rep* **20(1)**: 79-110.

Caboche S., Pupin M., Leclere V., Fontaine A., Jacques P., and Kucherov G. (2008) NORINE: a database of nonribosomal peptides. *Nucleic Acids Res* **36**: D326-331.

Caboche S., Pupin M., Leclere V., Jacques P., and Kucherov G. (2009) Structural pattern matching of nonribosomal peptides. *BMC Struct Biol* **9**: 15.

Challis G. L., Ravel J., and Townsend C. A. (2000) Predictive, structure-based model of amino acid recognition by nonribosomal peptide synthetase adenylation domains. *Chem Biol* **7(3)**: 211-224.

Chenna R., Sugawara H., Koike T., Lopez R., Gibson T. J., Higgins D. G., and Thompson J. D. (2003) Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res* **31(13)**: 3497-3500.

Conti E., Stachelhaus T., Marahiel M. A., and Brick P. (1997) Structural basis for the activation of phenylalanine in the non-ribosomal biosynthesis of gramicidin S. *EMBO J* **16(14)**: 4174-4183.

Couto B. R., Ladeira A. P., and Santos M. A. (2007) Application of latent semantic indexing to evaluate the similarity of sets of sequences without multiple alignments character-by-character. *Genet Mol Res* **6(4)**: 983-999.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., and Harshman R. (1990) Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* **41(6)**: 391-407.

Demain A. L. (1999) Pharmaceutically active secondary metabolites of microorganisms. *Appl Microbiol Biotechnol* **52(4)**: 455-463.

Demain A. L. (2009) Antibiotics: Natural products essential to human health. *Medicinal Research Reviews* **29(6)**: 821-842.

Donadio S., Monciardini P., and Sosio M. (2007) Polyketide synthases and nonribosomal peptide synthetases: the emerging view from bacterial genomics. *Natural Product Reports* **24(5)**: 1073.

Eddy S. R. (1998) Profile hidden Markov models. *Bioinformatics* **14(9)**: 755-763.

Finking R., and Marahiel M. A. (2004) Biosynthesis of Nonribosomal Peptides1. *Annual Review of Microbiology* **58(1)**: 453-488.

Fischbach M. A., Walsh C. T., and Clardy J. (2008) Chemical Ecology Special Feature: The evolution of gene collectives: How natural selection drives chemical innovation. *Proceedings of the National Academy of Sciences* **105(12)**: 4601-4608.

Garcia E. (2006) Latent Semantic Indexing (LSI) How-to-Calculation <http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-4-lsi-how-to-calculations.html>. Accessed February 12, 2011.

Homayouni R., Heinrich K., Wei L., and Berry M. W. (2004) Gene clustering by Latent Semantic Indexing of MEDLINE abstracts. *Bioinformatics* **21(1)**: 104-115.

Hongseok T., Sohng J. K., and Park K. (2009) Development of an Analysis Program of Type I Polyketide Synthase Gene Clusters Using Homology Search and Profile Hidden Markov Model. *J Microbiol Biotechnol* **19(2):** 140-146.

Hranueli D., Cullum J., Basrak B., Goldstein P., and Long P. F. (2005) Plasticity of the streptomyces genome-evolution and engineering of new antibiotics. *Curr Med Chem* **12(14)**: 1697-1704.

Larkin M. A., Blackshields G., Brown N. P., Chenna R., McGettigan P. A., McWilliam H., Valentin F., Wallace I. M., Wilm A., Lopez R., Thompson J. D., Gibson T. J., and Higgins D. G. (2007) Clustal W and Clustal X version 2.0. *Bioinformatics* **23(21)**: 2947-2948.

Lautru S., and Challis G. L. (2004) Substrate recognition by nonribosomal peptide synthetase multi-enzymes. *Microbiology* **150(6)**: 1629-1636.

Li M. H. T., Ung P. M. U., Zajkowski J., Garneau-Tsodikova S., and Sherman D. H. (2009) Automated genome mining for natural products. *BMC Bioinformatics* **10(1)**: 185.

Mootz H. D., Schwarzer D., and Marahiel M. A. (2002) Ways of assembling complex natural products on modular nonribosomal peptide synthetases. *Chembiochem* **3(6)**: 490-504.

Nature News (2011) Phage on the rampage <http://www.nature.com/news/2011/110609/full/news.2011.360.html>. Accessed June 18, 2011.

NCBI (1997) Phenylalanine Activating Domain Of Gramicidin Synthetase 1 In A Complex With Amp And Phenylalanine <http://www.ncbi.nlm.nih.gov/Structure/mmdb/mmdbsrv.cgi?uid=47919>. Accessed April 29, 2011.

Pettersen E. F., Goddard T. D., Huang C. C., Couch G. S., Greenblatt D. M., Meng E. C., and Ferrin T. E. (2004) UCSF Chimera--a visualization system for exploratory research and analysis. *J Comput Chem* **25(13)**: 1605-1612.

Rausch C. (2005) Specificity prediction of adenylation domains in nonribosomal peptide synthetases (NRPS) using transductive support vector machines (TSVMs). *Nucleic Acids Research* **33(18)**: 5799-5808.

Sattely E. S., Fischbach M. A., and Walsh C. T. (2008) Total biosynthesis: in vitro reconstitution of polyketide and nonribosomal peptide pathways. *Natural Product Reports* **25(4)**: 757.

Schwarzer D., and Marahiel M. A. (2001) Multimodular biocatalysts for natural product assembly. *Naturwissenschaften* **88(3)**: 93-101.

Stachelhaus T., Mootz H. D., and Marahiel M. A. (1999) The specificity-conferring code of adenylation domains in nonribosomal peptide synthetases. *Chem Biol* **6(8)**: 493-505.

Starcevic A., Zucko J., Simunkovic J., Long P. F., Cullum J., and Hranueli D. (2008) ClustScan: an integrated program package for the semi-automatic annotation of modular biosynthetic gene clusters and in silico prediction of novel chemical structures. *Nucleic Acids Research* **36(21)**: 6882-6892.

Strieker M., Tanović A., and Marahiel M. A. (2010) Nonribosomal peptide synthetases: structures and dynamics. *Current Opinion in Structural Biology* **20(2)**: 234-240.

The Mathworks (2011) MATLAB - The Language Of Technical Computing <http://www.mathworks.com/products/matlab/>. Accessed April 29, 2011.

Weber T., Rausch C., Lopez P., Hoof I., Gaykova V., Huson D. H., and Wohlleben W. (2009) CLUSEAN: A computer-based framework for the automated analysis of bacterial secondary metabolite biosynthetic gene clusters. *Journal of Biotechnology* **140(1-2)**: 13-17.

Yadav G. (2003) SEARCHPKS: a program for detection and analysis of polyketide synthase domains. *Nucleic Acids Research* **31(13)**: 3654-3658.

Yonus H., Neumann P., Zimmermann S., May J. J., Marahiel M. A., and Stubbs M. T. (2008) Crystal Structure of DltA: IMPLICATIONS FOR THE REACTION MECHANISM OF NON-RIBOSOMAL PEPTIDE SYNTHETASE ADENYLATION DOMAINS. *Journal of Biological Chemistry* **283(47)**: 32484-32491.

Yuan Y., Lin L., Dong Q., Wang X., and Li M. (2005) A Protein Classification Method Based on Latent Semantic Analysis. *Conf Proc IEEE Eng Med Biol Soc* **7**: 7738-7741.

Zotchev S. B., Stepanchikova A. V., Sergeyko A. P., Sobolev B. N., Filimonov D. A., and Poroikov V. V. (2006) Rational design of macrolides by virtual screening of combinatorial libraries generated through in silico manipulation of polyketide synthases. *J Med Chem* **49(6)**: 2077-2087.

9. Annexes

9.1. Supplementary figures

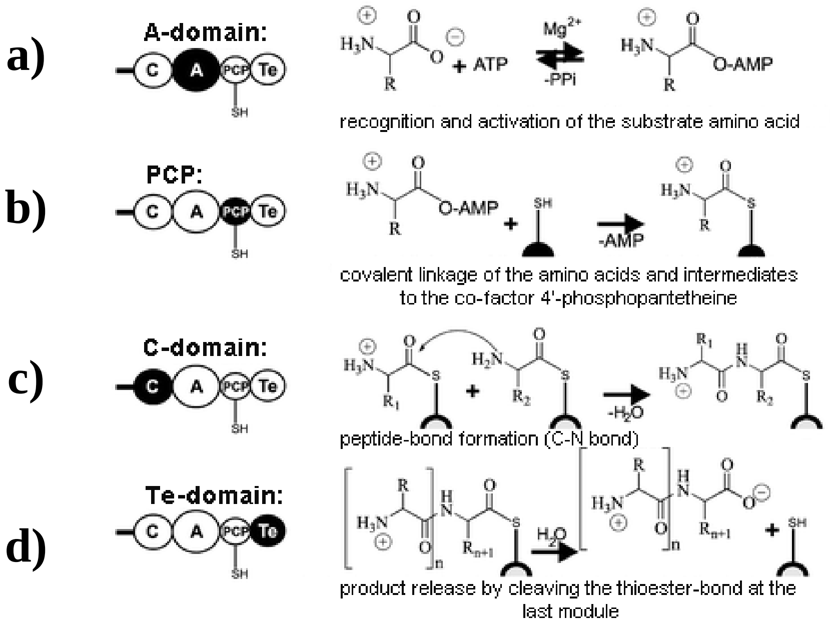
a)
A-domain:
C
A
PCP
Te
SH
+ ATP
Mg^{2+}
-PPi
O-AMP
recognition and activation of the substrate amino acid
b)
PCP:
-AMP
covalent linkage of the amino acids and intermediates to the co-factor 4'-phosphopantetheine
c)
C-domain:
$-H_2O$
peptide-bond formation (C-N bond)
d)
Te-domain:
H_2O
product release by cleaving the thioester-bond at the last module

Figure S1. Obligatory NRPS domain chemical reactions (Schwarzer and Marahiel, 2001).

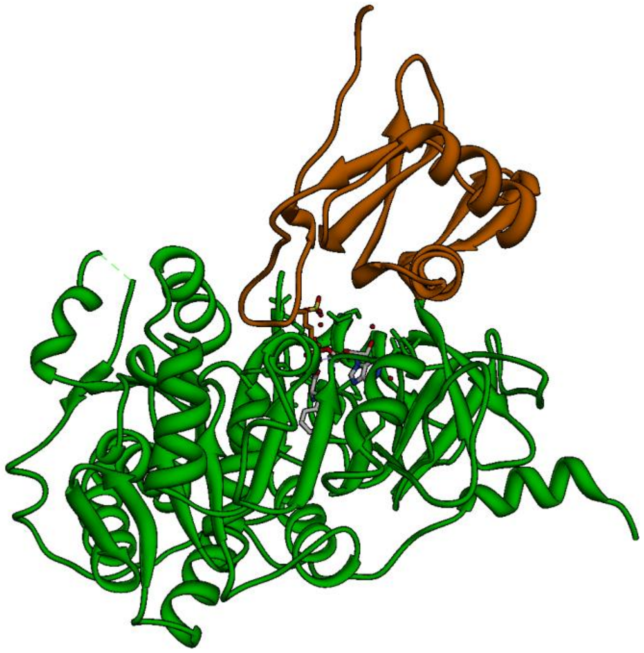

Figure S2. Three dimensional representation of A-domain which activates the amino acid phenylalanine in antibiotic gramicidin S biosynthesis with bound substrates. The large N-terminal domain is shown in green, whereas the small C-terminal domain is showed in orange. The figure was downloaded from the Web page (http://www.ncbi.nlm.nih.gov/Structure/mmdb/mmdbsrv.cgi?uid=47919) and adapted in the program Chimera version 1.5.2 (Pettersen *et al.*, 2004).

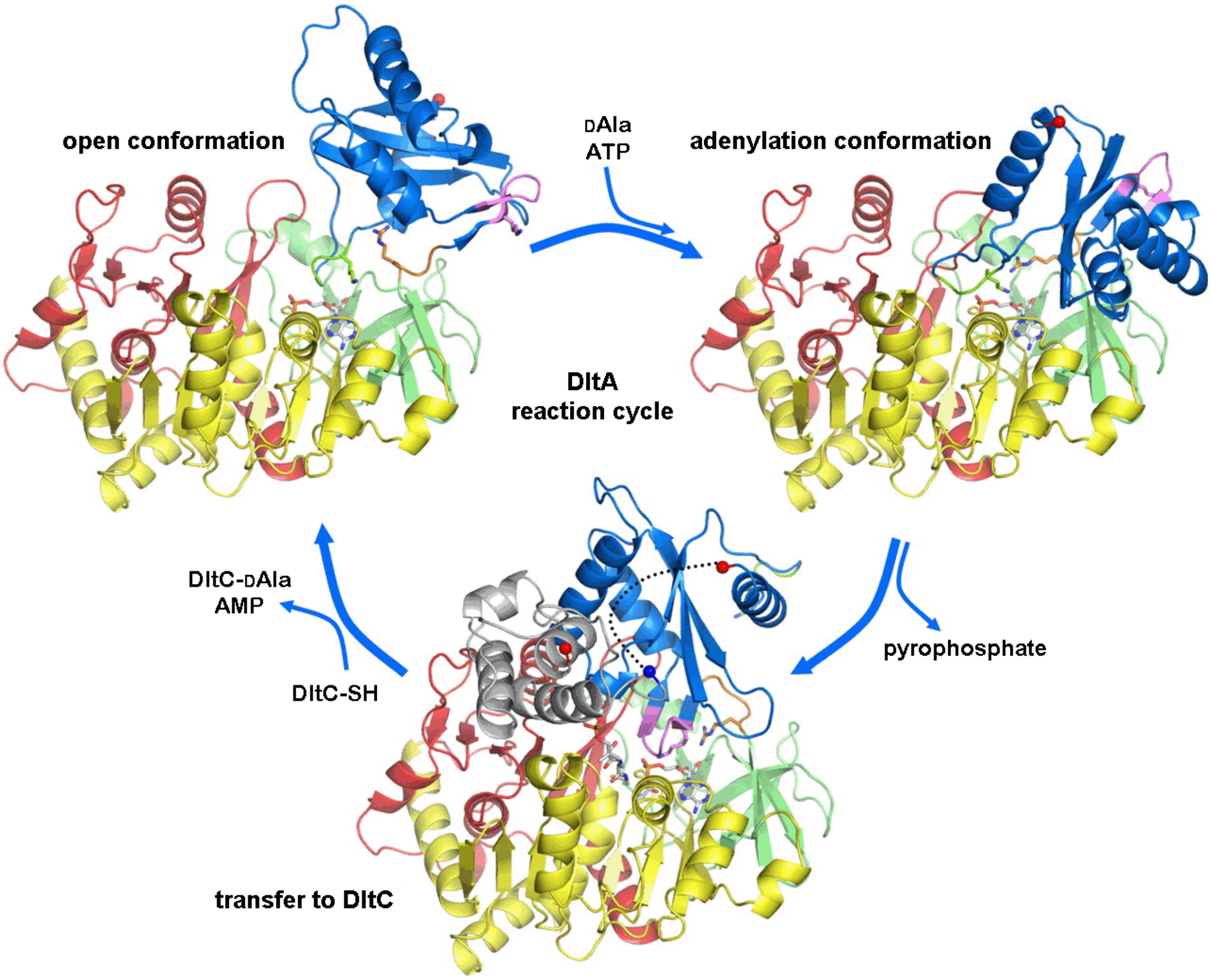

Figure S3. Proposed DltA reaction cycle (Yonus *et al.*, 2008).

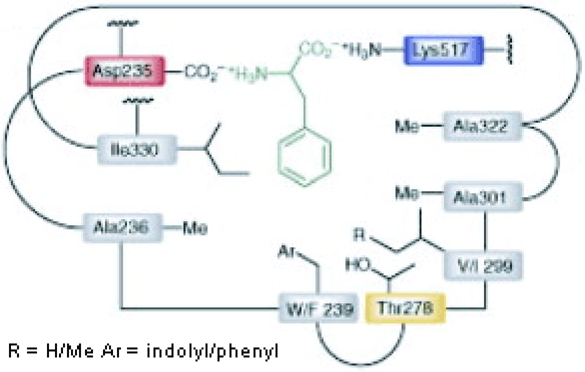

Figure S4. Phenylalanine binding pocket in GrsA (Challis *et al.*, 2000).

9.2. Abbreviation of amino acids

	Abbreviation	Amino acid		Abbreviation	Amino acid
1	'AAD'	2-amino-adipic acid	25	'HYV-D'	2-hydroxy-valeric acid
2	'ABU'	2-amino-butyric acid	26	'ILE'	isoleucine
3	'AEO'	2-amino-9,10-epoxy-8-oxodecanoic acid	27	'IVA'	isovaline
4	'ALA'	alanine	28	'LEU'	leucine
5	'ALA-B'	β-alanine	29	'LYS'	lysine
6	'ALA-D'	D-alanine	30	'LYSB'	β-lysine
7	'ALANINOL'	-	31	'3-ME-GLU'	3-methyl-glutamate
8	'ARG'	arginine	32	'ORN'	ornitine
9	'ASN'	asparagine	33	'PHE'	phenylalanine
10	'ASP'	aspartatic acid	34	'PHG'	phenyl-glycine
11	'BHT'	beta-hydroxy-tyrosine	35	'PIP'	pipecolic acid
12	'BMT'	(4R)-4[(E)-2-butenyl]-4-methyl-L-threonine	36	'4pPRO'	4-propyl-proline
13	'CYS'	cysteine	37	'PRO'	proline
14	'DAB'	2,4-diamino-butyric acid	38	'SAL'	salicylic acid
15	'DHB'	2,3-dihydroxy-benzoic acid	39	'SER'	serine
16	'DHPG'	3,5-dihydroxy-phenyl-glycine	40	'SER-THR'	serine or threonine
17	'DHT'	dehydro-threonin	41	'TCL'	(4S)-5,5,5-trichloro-leucine
18	'D-LYSERG'	D-lysergic acid	42	'THR'	threonine
19	'DPG'	3,5-dihydroxy-phenyl-glycine	43	'TRP'	tryptophan
20	'GLN'	gluatmine	44	'TYR'	tyrosine
21	'GLU'	glutamic acid	45	'VAL'	valine
22	'GLY'	glicine	46	'VALHYPHAA'	valine or hydrophobic amino acid
23	'HIS'	histidine	47	'VOL'	valinol
24	'HPG'	4-hydoxy-phenyl-glycine			