SEQUENCE-BASED LEARNING ALGORITHMS FOR UNDERSTANDING AND IMPROVING PROTEIN CHARACTERISTICS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 7 april 2015 om 15:00 uur

door

Bas Adriaan VAN DEN BERG

bioinformatica ingenieur
geboren te Bleiswijk, Nederland.

Dit proefschrift is goedgekeurd door de promotors:

Prof. dr. ir. M. J. T. Reinders
Prof. dr. ir. D. de Ridder

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. J. T. Reinders,	Technische Universiteit Delft, promotor
Prof. dr. ir. D. de Ridder,	Wageningen UR, promotor
Onafhankelijke commissieleden:	
Prof. dr. I. W. C. E. Arends,	Faculteit Technische Natuurwetenschappen, Technische Universiteit Delft
Prof. dr. R. A. L. Bovenberg,	Rijksuniversiteit Groningen
Prof. dr. J. Heringa,	Vrije Universiteit Amsterdam
Dr. A. F. J. Ram,	Universiteit Leiden
Prof. dr. G. Vriend,	Radboud Universiteit Nijmegen
Prof. dr. L. F. A. Wessels,	Nederlands Kanker Instituut, Amsterdam en Technische Universiteit Delft, reservelid

This work was supported by the BioRange programme of the Netherlands Bioinformatics Centre (NBIC) and was part of the Kluyver Centre for Genomics of Industrial Fermentation, subsidiaries of the Netherlands Genomics Initiative (NGI).

This work was conducted at the Delft University of Technology and in collaboration with DSM Biotechnology Center, Delft. Part of this work was conducted at the European Bioinformatics Institute, Hinxton, UK.

ISBN 978-94-6186-443-7

Printed by Gildeprint - Enschede

An electronic version of this dissertation is available at:
http://repository.tudelft.nl/

CONTENTS

1

INTRODUCTION

1

The introduction of fast DNA sequencing techniques in the 1970s [1] resulted in a rapidly growing amount of sequence data. Computational resources became essential for storing and making this data accessible [2, 3], and for biological sequence comparison [4–6]: the field of bioinformatics emerged. Since then, bioinformatics has added a new dimension to biological research. Sequence analyses have for example resulted in prediction methods that are used to identify specific functional regions in genomic DNA (genome annotation), and sequence comparison methods are used to find common evolutionary ancestors, thereby enabling the construction of phylogenetic trees. The emergence of other high-throughput measurement techniques further expanded the scope of bioinformatics research. This, for example, resulted in disease prediction methods based on gene expression profiles and provided better insight into the high-level cell organization (systems biology). Currently, advanced modeling and prediction capabilities are moving bioinformatics into a new era in which computational methods are employed for (re)designing biological molecules. This thesis deals with the development of sequence-based predictors and their use to redesign biological molecules, in particular focusing on industrial production of enzymes in biotechnology.

1.1. INDUSTRIAL PRODUCTION OF ENZYMES

Many centuries ago, humans discovered how to use microorganisms for enhancing food and beverage products. One of the earliest examples is the use of yeast for making bread and brewing beer in ancient Egypt, dating back to roughly 2000 BC [7]. Without many people realizing, nowadays many industries make use of microorganisms and their biological products for creating and modifying many products we use on a daily basis. An important biological product that is produced at industrial scale is a specific type of protein: enzymes.

Enzymes are molecules that catalyze a reaction between two or more metabolites, small chemical compounds. They are essential for living cells because they enable reaction rates that provide cells with sufficient nutrient concentrations. Enzymes have found broad application in foods and beverages, animal feed, paper and pulp, textile, and leather industries [8, 9]. More recently, a growing environmental awareness [10–12] increased interest in the production of second generation biofuels, in which enzymes are used for breaking down biomass (agriculture or municipal waste) into sugars that can be transformed into ethanol [13, 14]; in enzymatic degradation of plastics [15, 16]; and in the use of enzymes in detergents in order to replace chemicals and to allow for lower washing temperatures.

Different fungi and bacteria are used for the production of industrial enzymes. The filamentous fungus *Aspergillus niger* is one such organism that is widely used for this purpose [17]. This organism usually grows on decaying vegetation, soil and plants, where it secretes enzymes to break down biopolymers and transports the resulting products back into the cell as food. To this end, the organism has an exceptionally high secretion capacity, which makes it well suited for the production of industrial enzymes. *A. niger* is used as host organism in this work.

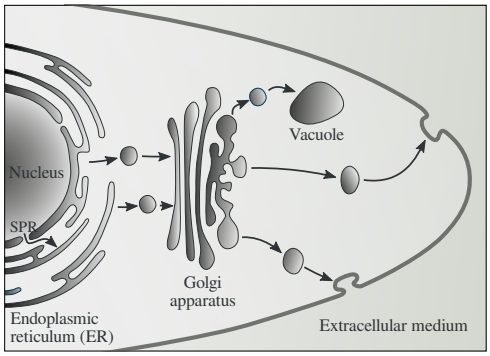


Figure 1.1: The classical secretory pathway. After transcription, mRNA translocation to the cytosol and translation initiation, an N- terminal signal peptide is recognized by a signal recognition particle (SRP) when the peptide emerges from the ribosome. The SRP directs the ribosome-nascent protein complex to the translocon in the endoplasmatic reticulum (ER) membrane, through which the protein enters the ER. Subsequently, vesicular transportation translocates the protein from the ER to the Golgi, and from the Golgi to the outer membrane where it is secreted. In both compartments, the protein might be subject to an array of post-translational modifications. The protein might also be subject to proteases at multiple locations.

In eukaryotic cells, secreted proteins are generally translocated to the cell exterior via the so-called classical secretory pathway that leads from the nucleus through the endoplasmic reticulum (ER) and the Golgi apparatus to the outer membrane (Figure 1.1) [18, Chapter 17.3]. This is a complex pathway involving many processes, including transcription, translation, co-translational translocation to the ER, post-translational modifications, protein folding, transport between organelles and to the outer membrane, and protein degradation [19–21]. Each of these processes might affect production levels, which makes production optimization difficult.

Much research effort has been spent on improving enzyme production levels. These works target both the production environment, e.g. fermentation conditions, and the host organism, e.g. strain improvement [22, 23], proteolysis control [24, 25], gene dosage [26–28], promoter selection, and codon usage [29, 30]. For homologous gene expression, this often resulted in production levels sufficient for large-scale production. With *A. niger*'s extensive secretome [31–33], this provides a range of potentially interesting targets for industrial production. The overproduction of heterologous proteins, on the other hand, often results in low extracellular concentrations, if any. Additional strategies, such as the use of fusion proteins, have been employed for improving this [34–37], but success rates are much lower in this case.

The produced enzyme itself has rarely been subjected to modification, even though it sounds likely that nature has also tinkered with the protein itself for obtaining desired production and secretion levels. Codon optimization has been used to optimize translation for the used host organism, but this is a modification at the codon level that does not affect the protein itself. The only found example that acts on the protein level is the mod-

ification of N-linked glycosylation sites. This approach has successfully been employed, but secretion levels do not always improve and the exact details on how the attached sugars contribute to production levels are unclear. In this thesis, based on observed sequence-based differences between proteins with low and high production levels, we aim to improve production levels by altering a protein's properties through sequence redesign.

1.2. PROTEIN DESIGN

Proteins are one of the four major classes of organic macromolecules in a cell, in which they perform a wide variety of functions. They are polymers composed of twenty different amino acids that are chained together by peptide bonds. Each of the twenty amino acids have the same amine (-NH2) and carboxylic acid (-COOH) groups between which the peptide bonds are formed, and each amino acid has a unique side chain that differs in size, polarity, and flexibility, thereby providing them with characteristic physicochemical properties. Within a cell, polar water molecules, which make up about 70% of the cell mass, cause the nonpolar parts of the peptide to pack together in order to minimize their contact with water. Consequently, proteins fold into their so called native fold, a state in which the protein's Gibbs free energy is minimized. However, some proteins are intrinsically disordered and many proteins have intrinsically disordered parts.

As described in section 1.1, enzymes have found broad application in our everyday lives. These products often need to be optimized for enhanced functionality or for altered characteristics in order to enable functioning in different environmental conditions. *Protein engineering* is the field of research that aims for this goal. Put simply, protein engineers change a protein's amino acid sequence to change its properties to achieve a desired goal. This provides two main challenges: exploring an enormous sequence/structure space and efficiently screening for the desired property.

Directed evolution is a successful protein engineering approach that mimics natural selection with controlled selective pressure. Random mutagenesis is used to diversify proteins after which they are screened to select those with the desired property for the next round of diversification and selection. This is usually done for multiple rounds. Directed evolution can be applied without any prior knowledge on the protein structure, but the lab work involved is expensive. Also, high-throughput screening for the desired property is not always possible.

Protein design is a method that gained popularity in the last decade [38–40]. This approach employs computational methods to select targets for site-directed mutagenesis. Relatively few constructs need to be tested in the lab, which makes it relatively cheap. However, predicting which substitutions result in desired property changes without affecting the protein's function remains a difficult problem. Even though directed evolution and protein design are two different approaches, they are not mutually exclusive [41].

The computational methods employed for protein design help finding amino acid se-

quences that 'fit' a predefined backbone fold. They search for an amino acid sequence with corresponding side-chain angles that have a minimal free energy for that backbone fold. Efficient search algorithms, such as dead-end-elimination or Monte Carlo simulation, are used to explore sequence/structure space. The fact that the bonds of the amino acid side-chains take on a limited number of angles is used to constrain the search space by using so-called rotamer libraries. A free energy function is used to optimize for fold stability. Many different variations are available, most of which combine both physics-based terms, such as the Lennard-Jones potential to model the attractive/repulsive force between two neutral atoms, and statistical terms, such as rotamer probabilities. Some methods allow for backbone flexibility to better simulate reality, others alter the energy function to compensate for using a fixed backbone.

Such tools have successfully been employed for designing novel globular folds [42, 43], but in most cases they are applied to alter properties of existing proteins. These types of problems can be roughly separated into three categories based on the scale on which they act. First, targeting the enzymatic function has a very specific focus on the active site, usually targeting only a few residues in or near the binding pocket in order to alter substrate binding properties [41, 44–46]. These type of designs require accurate structural knowledge and usually molecular dynamics simulations to test if substrate binding is affected. Second, protein interface designs, which target larger surface patches in order to alter binding affinity or specificity [47–49]. Finally, designs that change global protein properties, such as solubility [50] or thermostability [51–53], which target the entire protein or protein surface. The common issue in these design problems, including our problem of improving production levels, is that selecting target regions is not always straightforward. Human expertise is usually required to select specific mechanisms and/or specific types of amino acid substitutions that are known be related to the desired property change. Computational methods are then employed to search the restricted search space for substitutions that fit the structure.

In chapter 5 we introduce a more data-driven approach. Instead of manually selecting target regions or mechanisms which we expect to be related to the desired property change, we employ a large set of proteins for which data about the desired property is available. These example proteins are used to guide the design and therefore no prior knowledge about a relation between the structure features and the desired property change is required. This provides a less biased method in which not the human expert, but a large set of example proteins decide on what amino acid substitutions to make. A technique called classification is the basis of this approach.

1.3. CLASSIFICATION

Classification is a supervised machine learning technique that can be used for assigning class labels to objects based on a set of object features. Classifiers are trained using features of many example objects with known class labels, thereby learning the differences between objects in different classes. Afterwards, the trained classifier can be used for predicting class labels of new objects (Figure 4.1) [54]. This technique has extensively

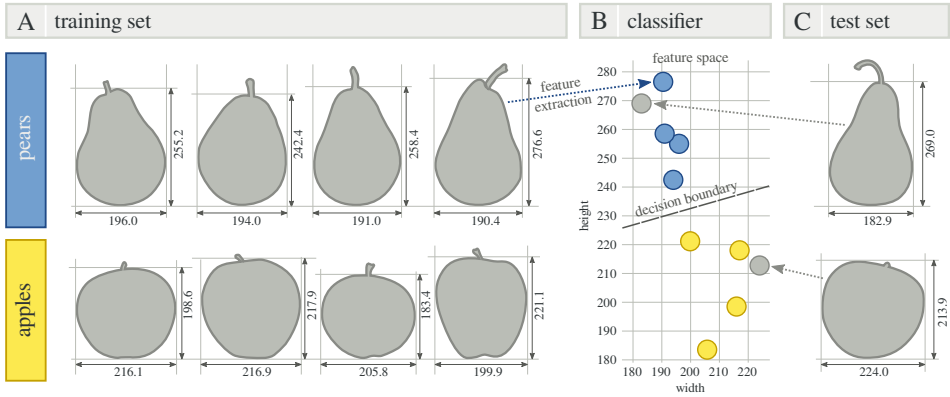


Figure 1.2: An illustrative classification example for discriminating between **(a)** two types of fruit, apples and pears. **(b)** After plotting their widths and heights on a two-dimensional grid, a line can be drawn that separates the apples from the pears. **(c)** For new fruit items, this line can now be used to predict if they are apples or pears by inspecting if their corresponding data points fall on the apple or the pear side of the decision boundary. For the given example, the objects are fruit items, the class labels are 'apple' and 'pear', and the used set of features are the width and height of the fruit. Deriving the decision boundary based on a set of objects (the training set) is called classifier training [54].

been used for developing practically applicable predictors and for identifying predictive features in order to learn about class characteristics [55]. In the field of bioinformatics, classifiers have for example been developed for finding gene expression signatures in breast cancer tumors that are predictive for a short distance to metastases [56]. This provided insight into what genes and metabolic pathways are important to the disease, and as predictor, such classifiers could help in deciding on patient treatments.

Chapters 2 and 3 of this work describe the development of classifiers that can be used to predict if the over-production of a given protein in *A. niger* will result in (relatively) high extracellular concentrations. These classifiers were trained using the sequence properties of a set of proteins with known production-levels under the conditions of overexpression in *A. niger*.

Developing protein classifiers is not new. Many such classifiers have been developed, for example for predicting a protein's subcellular localization [57–70], its nuclear localization [71], if it is soluble or not [72–76], if it has a signal peptide or not [77], its structural class [78–82], or its functional class [83–87]. The features used for characterizing proteins are predominantly derived from protein sequence and annotation data. The use of structural data is limited, due to the relatively few available protein structures. A simple but often effective set of features is the amino acid composition. For a given protein sequence, these features capture the relative frequency of occurrence of each of the twenty amino acids in this sequence. These are only twenty features that are easy to calculate, but they do not capture any information about the sequence-order or location.

Sequence-order can be captured using di- or tri-peptide compositions, possibly including gaps. However, this soon results in an explosion of the number of features. For support vector classifiers, this problem can be solved by employing the kernel trick, thereby avoiding explicit calculation and storage of the extensive amount of features (spectrum and mismatch kernels [88]). Other approaches capture sequence-order in a limited number of features by calculating residue correlation factors between every two residues at a defined distance d, which can be done for multiple distances. Features that use this approach include pseudo-amino acid composition [89, 90], quasi sequence order [91], and autocorrelation features [92–94]. Amino acids can also be clustered based on some property [95], resulting in a reduced alphabet and thereby a smaller number of possible di- and tri-peptides [96]. Finally, amino acid scales – mappings from each amino acid to a value that captures some property [97, 98] – can be used to transform amino acid sequences into property profiles. After smoothing, these can then be used for obtaining summed peak areas as features [99]. Sequence location can be included by splitting the sequence into n equal-sized parts, or using a fixed-length part of the 5' and 3' end of the protein, and calculating features for these subsequences separately. Besides deriving features from the amino acid sequence, similar features can also be derived from the protein's open reading frame (ORF), using either the nucleotide sequence or the codon sequence. The codon sequence in combination with the amino acid sequence also allows for calculating features capturing codon usage. Chapter 4 of this thesis describes a web-based tool that offers of the calculation and visualization of a range of sequence-derived protein features.

Finding predictive features not only enables *predicting* class labels of new unlabeled objects, it might also provide insight into class differences which could help in *learning* something about the underlying problem. Feature selection results can indicate what features are predictive, however, predictive feature combinations might be missed in this case [100]. Chapter 3 of this work describes an alternative approach in which inspection of the trained SVM classifier is used to derive the importance of the different features used by the classifier. This helped us to identify the most important sequence property differences between proteins with and without high production levels. This approach, however, is not applicable to all types of classifiers. In chapter 6 we therefore turned to classifiers performing a little less, but allowing interpretation of sequence-based differences between neutral and disease associated human missense mutations.

Finally, chapter 5 describes how we used the trained classifier to guide the protein design for improving production levels. This novel approach basically pushes the protein in feature space from one side of the decision boundary to the other (the side were the proteins with high-level production are). It is of course important to recognize that classifiers identify correlations between object features and class labels, whereas a causal relation is required for changing production levels with our protein design approach.

1

1.4. CONTRIBUTIONS OF THIS THESIS

Enhancing protein production levels is relevant for industrial enzyme production and can improve our understanding how protein sequence properties relate to their production levels. To achieve this, this thesis provides the following contributions.

First, we developed a classifier that predicts if the production of host-own proteins (homologous expression) under the condition of over-expression in *A. niger* will result in extracellular concentrations that are sufficiently high to make them interesting for industrial production. This classifier is made available online[1] and can be used to rank proteins by their probability of successful high-level production. This tool can be used to select potential targets for industrial production in *A. niger*. We also developed classifiers that predict successful over-production in *A. niger* for proteins from closely related organisms (heterologous expression) and showed that predictions for these proteins are much more difficult to make than for host-own proteins.

Second, we contributed by providing insight into the trained SVM classifiers. We showed which sequence properties are most predictive for (un)successful high-level production. Moreover, we showed that there is a similarity between the important sequence-properties for homologous and heterologous gene over-expression. These observations provide potential starting points for additional biological research to better understand the mechanisms that influence production levels. We also used the same SVM interpretation method to provide better insight into trained classifiers that predict if human missense mutations are neutral or disease associated.

Third, by exploring an extensive range of existing and novel protein features, we contributed by expanding the set of sequence-derived features that can be used for characterizing proteins. The feature calculations and classifier training techniques that were used for our research were made generally accessible for other research through a developed web-application[2].

Finally, we contribute by developing a protein design approach for which the selection of amino acid substitutions is, amongst other objectives and restrictions, guided by the previously trained classifiers. This approach was applied for the redesign of two enzymes, resulting in up to 10-fold improved production levels for both enzymes.

[1] http://helix.ewi.tudelft.nl/hipsec
[2] http://helix.ewi.tudelft.nl/spice

2

SEQUENCE-BASED PREDICTION OF PROTEIN SECRETION SUCCESS IN *Aspergillus niger*

Bastiaan A van den Berg, Jurgen F Nijkamp, Marcel JT Reinders,
Liang Wu, Herman J Pel, Johannes A Roubos, and Dick de Ridder

Published in Proceedings Pattern Recognition in Bioinformatics (PRIB) conference, 5th IAPR International Conference, Nijmegen, The Netherlands, September 22-24, 2010. Proceedings, Volume 6282, Page 3 – 14, doi:10.1007/978-3-642-16001-1_1.

ABSTRACT

The cell-factory *Aspergillus niger* is widely used for industrial enzyme production. To select potential proteins for large-scale production, we developed a sequence-based classifier that predicts if an over-expressed homologous protein will successfully be produced and secreted. A dataset of 638 proteins was used to train and validate a classifier, using a 10-fold cross-validation protocol. Using a linear discriminant classifier, an average accuracy of 0.85 was achieved. Feature selection results indicate what features are mostly defining for successful protein production, which could be an interesting lead to couple sequence characteristics to biological processes involved in protein production and secretion.

2.1. INTRODUCTION

The filamentous fungus *Aspergillus niger* has a high secretion capacity, which makes it an ideal cell-factory widely used for industrial production of enzymes [35]. Selecting proteins for large-scale production requires testing for successful over-expression and protein secretion. Because many proteins are of potential interest, a large amount of lab work is needed. This can be reduced by developing a software tool to prioritize proteins in advance. Such a tool might also indicate which gene or protein characteristics influence successful over-expression and secretion.

Various sequence-based classifiers have been developed, for example, to predict protein crystallization propensity [101], protein solubility [74], and protein subcellular localization [68], [65]. Subcellular localization predictors have been used to predict protein secretion [31], [102], but these methods predict if a protein is inherently extracellular, whereas our aim is to predict *successful* secretion of a protein after over-expression.

In this work, we present a classifier to predict if a homologous protein will successfully be secreted after over-expression in *A. niger*, using 25 sequence-based features and providing an accuracy of 0.85.

2.2. MATERIALS AND METHODS

DATA SET

The data set D contained 638 homologous proteins from *A. niger* CBS 513.88 [17] with a signal sequence predicted by SignalP [103]. For each protein, the open reading frame (ORF) and a binary score for successful over-expression was given. To obtain this binary success score, each protein was over-expressed through introduction of the predicted gene using the same strong glucoamylase promoter P_{GlaA}. Cultures were grown in shake-flasks and the filtered broth was put on an SDS-PAGE gel. Successful over-expression was defined as the detection of a visible band in this gel. D contained 268 successfully detected proteins (D_{pos}), and 370 unsuccessfully detected proteins (D_{neg}). The data set will be publicly available soon.

Table 2.1: Calculated features with class separability score.

	guanine	(2.5)	GC	(1.3)		
Nucleotide	adenine	(0.4)	CAI	(5.3)		
compositional	thymine	(2.3)				
	cytosine	(2.9)				
	alanine	(2.3)	leucine	(9.0)	helix {I,L,F,W,Y,V}	(0.4)
	arginine	(13.6)	lysine	(9.3)	turn {N,G,P,S}	(8.9)
	asparagine	(15.0)	methionine	(6.3)	sheet {A,E,L,M}	(10.8)
	aspartic acid	(7.2)	phenylalanine	(0.1)	acidic {N,D,E,Q}	(7.9)
Amino acid	cysteine	(0.2)	proline	(5.4)	basic {R,K,H}	(15.7)
compositional	glutamic acid	(5.6)	serine	(1.6)	charged {R,D,C,E,H,K,Y}	(5.6)
	glutamine	(0.2)	threonine	(8.3)	small {A,N,D,C,G,P,S,T,V}	(9.7)
	glycine	(9.2)	tryptophan	(6.3)	tiny {A,G,S}	(3.5)
	histidine	(4.2)	tyrosine	(13.6)		
	isoleucine	(0.9)	valine	(1.9)		
Signal-based	hydrophobic peaks	(9.1)				
features	hydrophilic peaks	(15.5)				
	GRAVY	(1.8)				
Global features	isoelectric point	(16.2)				
	sequence length	(5.4)				

SEQUENCE-BASED FEATURES.

For each item $i \in D$, a feature vector $\vec{d}_i$ with 39 sequence-based features was constructed (Table 2.1). Next to simple compositional features, features that relate to protein solubility and membrane binding were chosen, because it is expected that these characteristics influence successful protein secretion. Features are calculated using the entire ORF sequence and corresponding protein sequence, including the signal peptide. A two-sample t-test with pooled variance estimation was used as class separability criterion to evaluate the performance of each feature. Features with p-value > 0.001 (gray features in Table 2.1) were removed, resulting in a set of 25 features used for classifier development.

For this set of features, a heat map of the hierarchical clustered (complete linkage) feature matrix is shown in Figure 2.1, in which each row is a protein in D and each column is a feature. The two additional columns on the right depict the measured and predicted class labels. They show that clustering of the proteins, using this feature set, already provides a separation of D_{pos} and D_{neg}.

COMPOSITIONAL FEATURES

Given a protein sequence, its amino acid composition is defined as the number of occurrences of the amino acid (frequency count) divided by the sequence length, providing 20

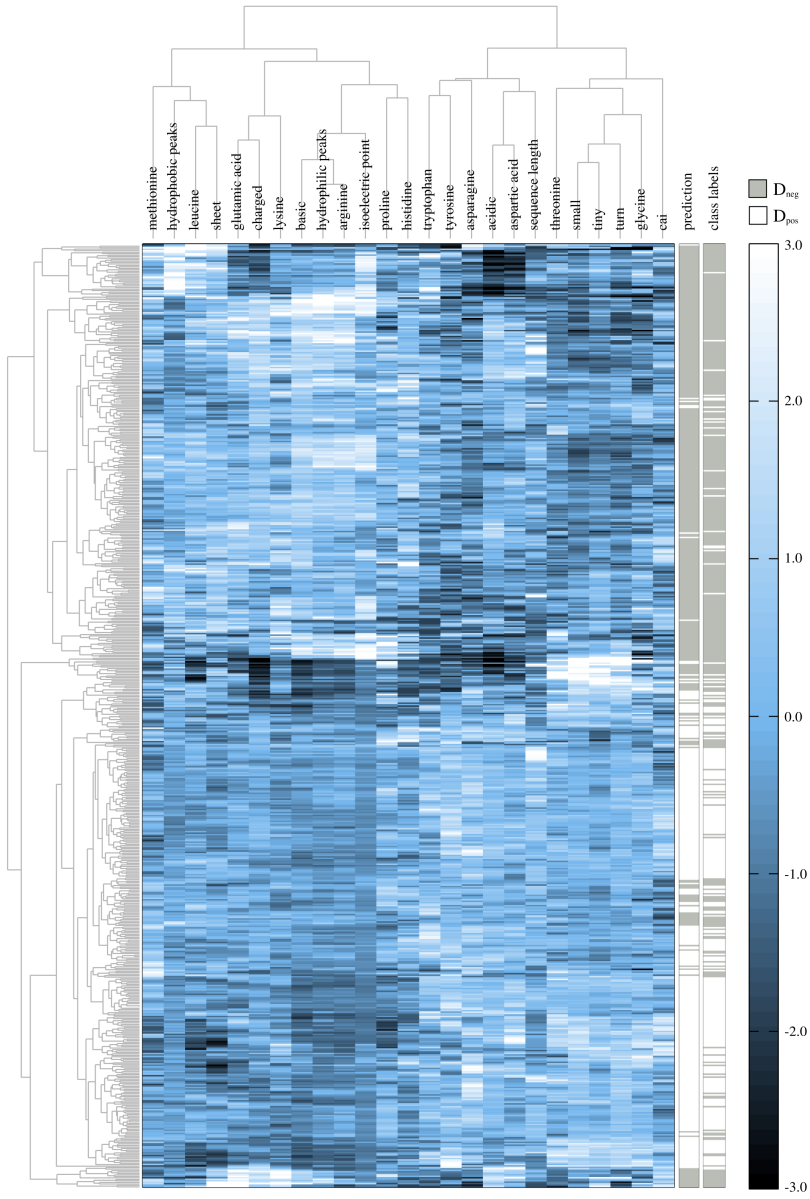


Figure 2.1: Heat map of clustered feature matrix. The rows are the proteins in D and the columns are the 25 features used for classifier development. The two columns on the right depict the predicted and measured class labels respectively.

features. The same was done for the nucleotide composition of the coding region, providing 4 features.

Additionally, we calculated the compositions of amino acid sets that share a common property. Given a protein sequence and an amino acid set, the amino acid set composition is defined as the sum of the frequency counts of each of the specified amino acids, divided by the sequence length. Eight sets were used: helix *{I,L,F,W,Y,V}*, turn *{N,G,P,S}*, sheet *{A,E,L,M}*, charged *{R,D,C,E,H,K,Y}*, small *{A,N,D,C,G,P,S,T,V}*, tiny *{A,G,S}*, basic *{R,K,H}*, and acidic *{N,D,E,Q}*. One nucleotide set was used: GC.

As final compositional feature we used the codon adaptation index (CAI)[104], which was calculated with the codon usage index of all genes in the *A. niger* genome.

SIGNAL-BASED FEATURES

Two features capture the occurrence of local hydropathic peaks: *hydrophobic peaks* and *hydrophilic peaks*, both derived from a protein hydropathicity signal [105] that was constructed using the (normalized) hydropathicity amino acid scale of Kyte and Doolitle [106].

An *amino acid scale* is defined as a mapping from each amino acid to a value. Given a protein sequence, a hydropathicity signal was obtained by replacing each residue by its amino acid scale value (Figure 2.2A). The signal was smoothed through convolution with a triangular function (Figure 2.2B). To capture the extreme values of the smoothed signal, an upper and lower threshold were set (Figure 2.2C). *Hydrophobic peaks* is defined as the sum of all areas above the upper threshold divided by the sequence length, *hydrophilic peaks* is defined as the sum of all areas below the lower threshold divided by the sequence length.

The window size and edge of the triangular function (Figure 2.2B), and both thresholds (Figure 2.2C) can be varied. In each CV loop of the training and validation protocol (Section 2.2), an exhaustive search was applied to optimize the features' class separability score, using: $window\ size = 3,5,\ldots,21$; $edge = 0.0,0.2,\ldots,1.0$; $threshold = 0.5,0.54,\ldots,0.86$ for *hydrophobic peaks* and 0.5, 0.45, ..., 0.05 for *hydrophilic peaks*.

GLOBAL FEATURES

Three global features were used: the grand average of hydrophobicity (GRAVY), i.e., the sum of all Kyte and Doolitle amino acid scale values divided by the sequence length; the isoelectric point (pI), i.e., the predicted pH at which the net charge of the protein is zero; and finally the sequence length, i.e., the number of residues in the protein sequence.

WOLF PSORT

To test whether using predicted localization would improve performance, WoLF PSORT [65] was used to predict secretion of the proteins in D. Next to the amino acid composition and the sequence length, which we also used as features, WoLF PSORT uses

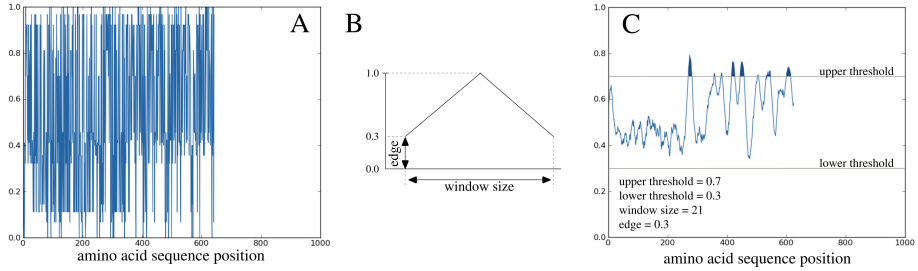


Figure 2.2: Hydropathic peaks features. A) A raw protein hydropathicity signal obtained by replacing each amino acid in the sequence by its value in the normalized Kyte and Doolitle amino acid scale. **B)** Triangular function used to smooth the raw signal. **C)** Smoothed signal obtained by convolution of the raw signal in *A* with the function in *B*.

features based on sorting signals and functional motifs. To use the prediction as feature, we assigned proteins with intracellular localization prediction a value of 0, and proteins predicted to be extracellular a value of 1.

PERFORMANCE EVALUATION

We used five measures to evaluate classification performance. Four of these are based on the confusion matrix. This matrix contains the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Let the set of positives be $P = TP + FN$, the set of negatives $N = TN + FP$, the set of predicted positives $P' = TP + FP$, and the set of predicted negatives $N' = TN + FN$. The confusion matrix-based measures are; $accuracy = (TP + TN)/(P + N)$, $sensitivity = TP/P$, $specificity = TN/N$, and Matthews correlation coefficient score $MCC = (TP \times TN - FP \times FN)/\sqrt{P \times N \times P' \times N'}$. The MCC-score [107] is suited in case of different class sizes, which applies in our case. The score ranges from 0 for random assignment, to 1 for perfect prediction.

The aforementioned scores take into account only one operating point on the receiver operating characteristic (ROC) curve. As a fifth measure, we took the area under the ROC curve (AUC), thereby taking into account a range of operating points. Because the goal is to reduce the amount of lab work, we are mainly interested in low false positive rates, i.e., the left region of the ROC-curve. Therefore, we used the AUC over the range of 0 – 0.3 false positive rate (ROC0.3) as main performance measure.

TRAINING AND VALIDATION PROTOCOL

To avoid overestimation of classification performance, a double 10-fold CV protocol was used, based on the protocol in [108]. We used 10-fold CV feature selection with classifier performance as selection criterion, in which the expected error $((FP/P + FN/N)/2)$ was used as performance measure.

The protocol is shown in Figure 2.3. The dataset D is split into ten equal-sized random

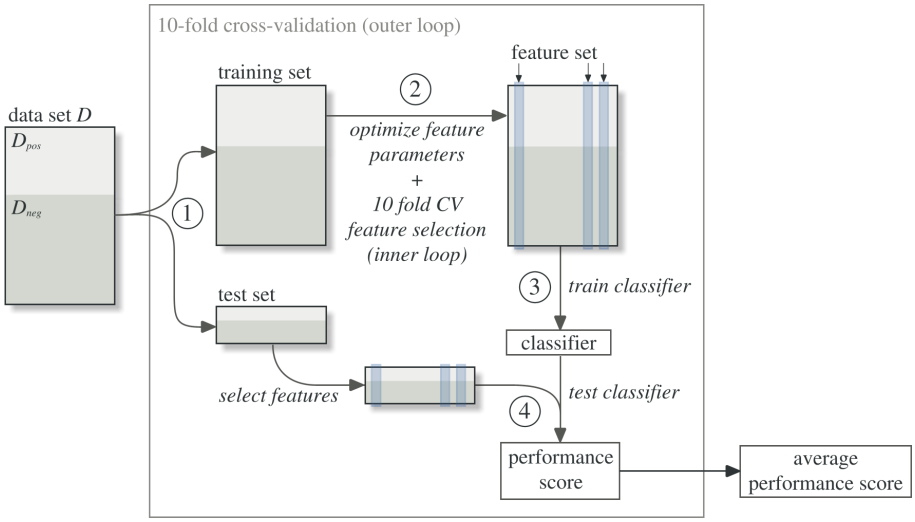


Figure 2.3: Training and validation protocol.

stratified sets. In each outer loop, one of the sets is used as test set, and the remaining nine as the training set (1). An exhaustive search is done to optimize the parameters of the hydropathic peaks features for maximal class separability, and 10-fold CV feature selection (inner loop) is applied on the training set to select an optimal feature set (2). As feature selection methods, we used both forward and backward feature selection. The optimal feature set is used to train a classifier on the entire training set (3). The resulting classifier is applied to the test set that was not employed for training, resulting in a performance score (4). Finally, the performance scores of the 10 CV loops are averaged, resulting in an average performance score.

The training and validation protocol was implemented in Matlab, using the PRTools pattern recognition toolbox [109].

CLASSIFIERS

We tested 8 classifiers: linear and quadratic normal density-based Bayes classifiers (ldc, qdc); nearest mean classifier (nmc); k-nearest neighbor classifier, both with $k = 1$ and with k optimized by leave-one-out CV (1nnc, knnc), naive Bayes classifier (naivebc), Fisher's least square linear classifier (fisherc), and a radial basis support vector machine (svm, $\gamma = 1/\text{number of features}$). We used libsvm [110] for the support vector machine.

2.3. RESULTS

The classifier performance scores are given in Table 2.2. We compared the ROC0.3 scores of the different methods using a paired t-test ($p < 0.05$) on the results of the 10 CV loops.

Table 2.2: Classifier performance scores.

classifier		ROC0.3	sensitivity	specificity	MCC	accuracy
ldc	f[1]	0.232 ±0.03	0.877 ±0.08	0.819 ±0.06	0.691 ±0.08	0.843 ±0.04
	b[2]	**0.236** ±0.03	0.873 ±0.08	0.830 ±0.05	0.700 ±0.07	0.848 ±0.03
svm	f	0.228 ±0.03	0.847 ±0.08	**0.857** ±0.02	**0.701** ±0.07	**0.853** ±0.03
	b	0.232 ±0.02	0.843 ±0.08	0.854 ±0.04	0.695 ±0.09	0.850 ±0.04
fisherc	f	0.234 ±0.03	0.873 ±0.08	0.819 ±0.06	0.688 ±0.08	0.842 ±0.04
	b	0.235 ±0.02	0.881 ±0.09	0.822 ±0.05	0.698 ±0.07	0.846 ±0.03
naivebc	f	0.224 ±0.03	0.854 ±0.08	0.800 ±0.05	0.649 ±0.09	0.823 ±0.04
	b	0.230 ±0.03	0.888 ±0.08	0.803 ±0.03	0.684 ±0.07	0.839 ±0.03
qdc	f	0.221 ±0.03	0.877 ±0.06	0.803 ±0.04	0.674 ±0.06	0.834 ±0.03
	b	0.227 ±0.03	0.884 ±0.05	0.805 ±0.04	0.682 ±0.08	0.838 ±0.04
nmc	f	0.227 ±0.03	**0.910** ±0.07	0.773 ±0.04	0.678 ±0.06	0.831 ±0.02
	b	0.224 ±0.02	0.899 ±0.07	0.773 ±0.04	0.666 ±0.05	0.826 ±0.02
knnc	f	0.218 ±0.03	0.858 ±0.09	0.770 ±0.06	0.624 ±0.10	0.807 ±0.05
	b	0.214 ±0.02	0.862 ±0.06	0.778 ±0.06	0.635 ±0.05	0.813 ±0.03
1nnc	f	0.195 ±0.04	0.798 ±0.09	0.781 ±0.09	0.578 ±0.15	0.788 ±0.07
	b	0.190 ±0.03	0.809 ±0.09	0.749 ±0.08	0.557 ±0.10	0.774 ±0.05

[1] forward feature selection, [2] backward feature selection

This showed that the nearest neighbor classifiers perform significantly worse than all other methods, except for qdc with forward feature selection. The best performance was obtained with ldc and backward feature selection.

Figure 2.4 shows the ROC0.3 scores of ldcs trained on each of the 25 single features, on all 25 features, and on features obtained by backward feature selection. The classifiers are ordered by score. A paired t-test ($p < 0.001$) on the 10 CV loops showed that all single-feature classifiers are significantly outperformed by both multi-feature classifiers. Although using all features provides a higher average score than using backward feature selection, the paired t-test ($p < 0.05$) indicates that the difference is not significant.

Applying WoLF PSORT on our dataset provided a sensitivity of 0.96 and a specificity of 0.49. It appears that WoLF PSORT is too optimistic, providing a large amount of FPs. This could be explained by the difference in the problems we address; WoLF PSORT predicts extracellular proteins, whereas our method also includes successful protein production and secretion. This means that extracellular proteins in D, which are positives for WoLF PSORT, can be part of D_{neg} because of unsuccessful protein production. We used the localization prediction as additional feature. Using ldc with backward feature selection, no significant improvement was observed, probably because the feature contains redundant data.

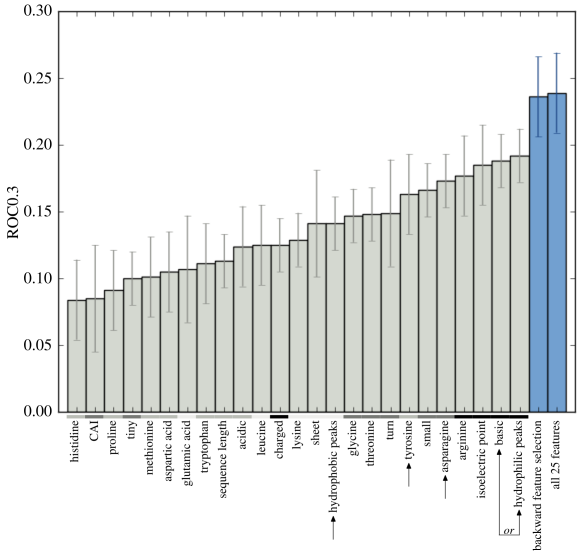


Figure 2.4: Single-feature and multi-feature classification scores.

OPERATING POINT EXAMPLE

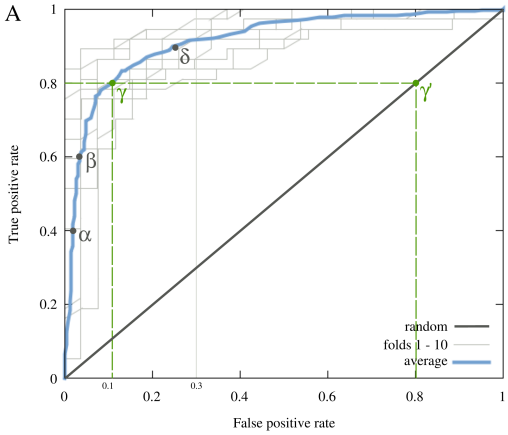
Figure 2.5A shows the ROC of the ldc with backward feature selection. One could use this classifier to screen a set of proteins for potential over-expression candidates. For example, if we have a set S of 100 proteins that we want to screen, containing 42 positives (S_{pos}) and 58 negatives (S_{neg}) (i.e., the same fraction of positives and negatives as D), and if we use γ as operating point, a true positive rate of 0.8 will be obtained. In this case, the classifier will predict 34 true positives and 6 false positives, which means that only 40 lab experiments are needed to identify 34 positives. Without the classifier, to identify 34 positives, both the false and the true positive rate will be 0.8 (operating point γ'). In this case, 80 lab experiments will be needed to identify 34 positives, which means that the classifier could reduce the amount of lab work by a factor two (Figure 2.5B).

FEATURE OPTIMIZATION

Figure 2.6 shows the optimal parameter settings for the hydrophilic and hydrophobic peaks feature as obtained in one of the CV loops. For both features, the same optimum was observed in each CV loop.

Interestingly, when using the optimal parameter settings, the raw signal of the hydrophilic peaks is not smoothed. With $window\ size = 3$ and $edge = 0.0$, the value at a specific location in the sequence is simply the amino acid scale value of the amino acid at that specific location. Therefore, the feature is actually the same as the GRAVY feature, but using an amino acid scale in which all values greater than the threshold are set to zero, and all other values are set to the threshold minus the value. In this case,

A



B

Example different operating points:
set of 100 proteins; 42 positives, 58 negatives

		α	β	γ	δ
	TP-rate	0.40	0.60	***0.80***	0.90
	TP	17	25	***34***	38
				γ	
Classifier	FP-rate	0.02	0.04	***0.10***	0.25
	FP	1	2	***6***	15
	lab tests	18	27	***40***	53
				γ	
Random selection	FP-rate	0.40	0.60	***0.80***	0.90
	FP	23	35	***46***	52
	lab tests	40	60	***80***	90
$\text{lab tests}_{\text{random}}$ / $\text{lab tests}_{\text{classifier}}$		2.22	2.22	***2.00***	1.70

Figure 2.5: ROC-curve. A) Average ROC curve of the ten CV loops (ldc, backward feature selection). The light gray curves are the ROC curves of the separate CV loops. The diagonal line illustrates the random selection ROC curve. **B)** Numeric example that shows the amount of lab work that could be saved for different operating points.

arginine is set to 0.1, lysine to 0.33, and the rest of the amino acids is set to zero. From another perspective, this feature can be seen as an amino acid set composition for the set {arginine, lysine} in which the arginine has a higher weight.

It is questionable if the resulting feature is still related to the proteins hydrophilic character. Since both arginine and lysine are also basic amino acids, it could just as well be related to the proteins basic character. Furthermore, because of the small window size, the feature does not take into account sequence order. However, it could be hypothesized that hydrophilic amino acids will mainly contribute to the proteins hydrophilic character when they have a relatively high occurrence within a larger region.

FEATURE CORRELATION

Figure 2.7 shows a heat map of the hierarchical clustered (complete linkage) feature correlation matrix. The cluster at the top left shows relatively high correlations, which can be explained by the fact that the features contain redundant data: *arginine* is part of both *basic* and *charged*, *basic* is a subset of *charged*, the isoelectric point is derived from a proteins charge and therefore correlated with *charged*, and *hydrophilic peaks* takes into account the amino acids arginine and lysine, that are both in *basic* and *charged*. There is also a high correlation between *small*, *turn*, and *tiny*. This can also be explained by data redundancy: both *turn* and *tiny* are a subset of *small*.

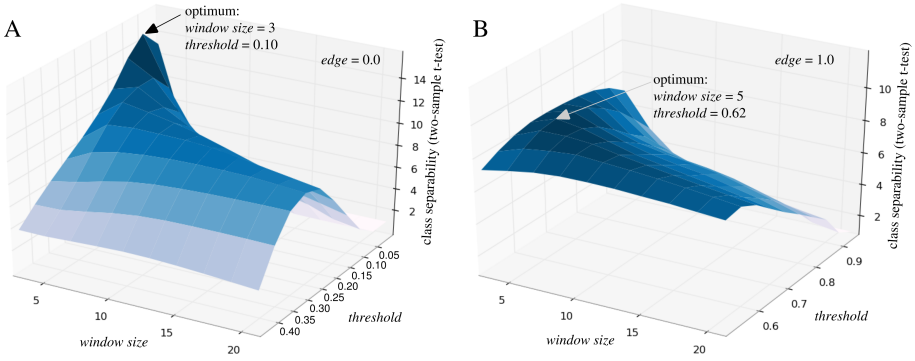


Figure 2.6: Parameter optimization of hydropathic peaks features. A) Class separability scores for the hydrophilic peaks feature plotted against different parameter settings. **B)** The same as in *A*, but for the hydrophobic peaks feature. Both plots show the result for one *edge* value, different *edge* values provided similar plots. Both plots were obtained in one of the CV loops, the same optimum was found in all CV loops.

FEATURE SELECTION

Using ldc with forward feature selection, the feature selection results of the 10 CV loops showed that: *asparagine* was always part of the top-3 selected features (7 times selected first), either *hydrophilic peaks* or *basic* was part of the top-3 selected features 9 times (6 times selected second), *hydrophobic peaks* was part of the top-4 selected features 9 times (7 times selected third), and *tyrosine* was part of the top-4 selected features 6 times (5 times selected fourth).

The high correlation between *hydrophilic peaks* and *basic* (Figure 2.7), together with the fact that both have a high class separability score (Table 2.1), explains their mutual exclusive selection. In Figure 2.4, the colors above the feature names depict what features are in the same correlation cluster and the arrows indicate what features are most often in the top-4 selected features. It shows that these features are in different correlation clusters, and are the best performing ones of their cluster. Therefore, feature selection seems to select individual features that best represent an underlying cluster of related features.

2.4. DISCUSSION

To be useful for large-scale production, a protein should be produced and secreted with high yield. We report a sequence-based approach to classify proteins into *successful* or *unsuccessful* production, which was trained and validated on a set of 638 proteins. We used 10-fold CV for feature selection and classifier training to avoid biased performance results. Since we are mostly interested in the operating points of the first 30 percent of the ROC-curve, we used the AUC of this region as the main performance measure.

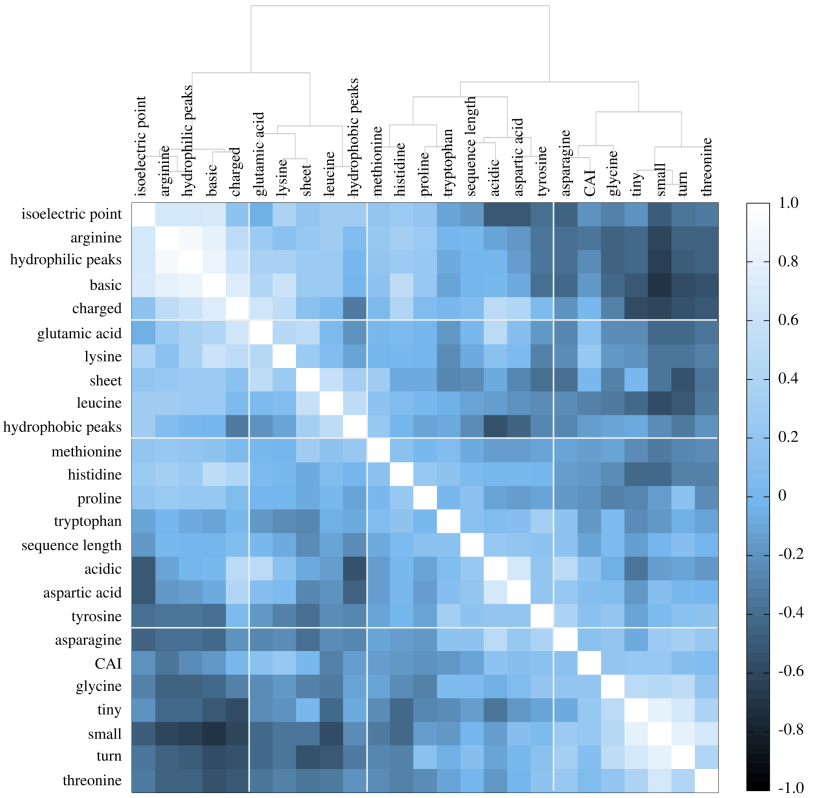


Figure 2.7: Heat map of clustered feature correlation matrix.

We calculated 39 features and used the 25 with highest class separability score for classification. We showed that both a classifier that uses all features and a classifier trained with feature selection, outperform classifiers trained on single features. The classifiers trained with feature selection did not significantly outperform the classifier trained on all 25 features, indicating that all features contribute to the result.

Furthermore, the feature selection results showed that asparagine, the set {arginine, lysine}, and tyrosine, as well as the hydrophobic peaks feature, were most defining in case of the linear discriminant classifier. To get more insight into protein secretion, it would be interesting to link the biological significance of these features to protein secretion mechanisms. For example, the asparagine composition could be related to N-linked glycosylation, a process that in many cases is important for protein folding and stability [111].

3

EXPLORING SEQUENCE CHARACTERISTICS RELATED TO HIGH-LEVEL PRODUCTION OF SECRETED PROTEINS IN *Aspergillus niger*

Bastiaan A van den Berg, Marcel JT Reinders, Marc Hulsman,
Liang Wu, Herman J Pel, Johannes A Roubos, Dick de Ridder

Published in PLOS ONE, Volume 7, Issue 10: page e45869, 2012, doi:10.1371/journal.pone.0045869.

ABSTRACT

Protein sequence features are explored in relation to the production of over-expressed extracellular proteins by fungi. Knowledge on features influencing protein production and secretion could be employed to improve enzyme production levels in industrial bioprocesses via protein engineering. A large set, over 600 homologous and nearly 2,000 heterologous fungal genes, were overexpressed in *Aspergillus niger* using a standardized expression cassette and scored for high versus no production. Subsequently, sequence-based machine learning techniques were applied for identifying relevant DNA and protein sequence features. The amino-acid composition of the protein sequence was found to be most predictive and interpretation revealed that, for both homologous and heterologous gene expression, the same features are important: tyrosine and asparagine composition was found to have a positive correlation with high-level production, whereas for unsuccessful production, contributions were found for methionine and lysine composition. The predictor is available online at `http://bioinformatics.tudelft.nl/hipsec`. Subsequent work aims at validating these findings by protein engineering as a method for increasing expression levels per gene copy.

3.1. INTRODUCTION

In industrial enzyme production, high-level protein production and secretion are key requirements. The commercial market value was estimated to be nearly US$ 5 billion in 2009; roughly half of production is accounted for by filamentous fungi [36]. Interest in industrial enzymes is still growing, driven by the increased demand for sustainable production processes and the need to move from a fossil fuel-based to a bio-based economy. This calls for the exploration of novel enzymes, as well as predictable methods for high-yield production processes. The filamentous fungi *Aspergillus niger*, *Aspergillus oryzae* and *Hypocrea jecorina* are the major fungal workhorses in industrial enzyme production, due to their efficiency in producing polysaccharide-degrading enzymes (particularly amylases, pectinases, lipases and xylanases) in high amounts. The genome sequence of the enzyme producing *A. niger* strain CBS513.88 was published in 2007 [17] and compared with a related citric-acid producing strain ATCC1015 in 2011 [112].

Although rational genetic engineering strategies have been developed [34, 35, 37], including codon optimization, strong promoters etc., protein overexpression is still often an art. Heterologous expression in particular is less successful, often hampered by low production levels [20]. Although protein overexpression, including the secretion process and quality control mechanisms such as UPR-ERAD, has been studied widely [19, 21, 113–115], no generic solution to improve heterologous overexpression is yet available. More successful is the use of fusion proteins, at the cost of reduced overall yield due to the production of the fusion partner. We propose another strategy: to re-engineer proteins to better match the cell's production and secretion machinery. In this paper, we take a first step in this direction.

Our aim is to identify protein characteristics that correlate with the production level of secreted proteins in a library of *A. niger* strains. Ideally, data on protein structure, fold-

ing and even post-translational modification and processing, both intracellular and extracellular, should be exploited to enhance our understanding of the cellular processing of successful and unsuccessful candidates. Such data is however limited and expensive to obtain, unattainable for large sets of non-commercial proteins. On the other hand, some of this information is also captured in the protein sequence as such, which therefore should be informative. Using a large and diverse library of protein sequences should allow focus on generic aspects, ignoring protein-specific aspects.

We constructed a unique library of over 2,600 strains to overexpress a selected protein sequence. After transformation using overexpression cassettes, productivity of each strain was screened by shake-flask growth and analysis of the protein composition of the supernatant on gel. Protein production was scored positive when, compared to the mother strain, an additional band on SDS-PAGE gel was observed in the expected molecular weight range; otherwise it was scored negative. Characteristics found to distinguish between proteins in the positive and negative classes may point to sequence features that could be adapted in optimization schemes to further "streamline" proteins that already show good expression, in analogy to what has been achieved with codon optimization, where gene sequences are adapted to match the translational machinery [30].

Statistically significant associations between sequence features and positive and negative class membership can be obtained relatively easy. However, such analyses are typically univariate, considering only individual features. In contrast, machine learning algorithms can combine large numbers of features and by that achieve more optimal prediction performance. Recently, different machine learning techniques have been applied on sequence data to predict protein localization [59, 68, 116] or protein solubility [74]. A disadvantage of machine learning approaches is that they often result in "black boxes", not easily providing insight into the properties that are defining for the prediction. With few exceptions [117, 118], sequence-based predictors are rarely interpreted.

We developed a sequence-based predictor for extracellular protein production by *A. niger*, with the explicit goal of interpreting which combinations of features are most predictive. We consider a large number of potentially interesting features and develop predictors for both homologous and heterologous gene expression. Sequence data was found to be predictive for both, although less accurate prediction results were obtained for the heterologous data set. Interestingly, interpretation of the underlying model parameters show that for both data sets similar properties are predictive for extracellular protein production. The trained classifier algorithms are made available in a freely accessible online tool (`http://bioinformatics.tudelft.nl/hipsec`).

3.2. METHODS

EXPERIMENTAL SETUP

Proteins were experimentally tested for high-level production in *A. niger*. Binary success scores were obtained by SDS-PAGE of (at least) triplicate shake-flask samples with strains over-expressing the introduced gene as described below. A positive success score was

given when a clear visible band was present, negative otherwise.

Strain - The strain used in this work is a recombinant strain derived from DS03043, a progenitor of CBS 513.88, in which the *gla*A loci (i.e., the promoter and coding sequences) were deleted, creating the so-called Δ*gla*A loci. From this strain, a strain was derived with a strongly reduced production of abundantly secreted proteins by inactivation of the major protease *pep*A and a number of alpha-amylases [119]. This protease- and amylase-reduced strain was used as host strain for over-expression of proteins.

Molecular biology techniques - In order to obtain targeted integration and expression of any desired gene in the above-mentioned host strain, a standard expression unit was used, where the gene of interest was inserted between the host-own glucoamylase promoter (original 2kb 5' *gla*A sequence) and glucoamylase terminator elements (original 2kb 3' *gla*A sequence) in a proprietary *Escherichia coli* vector. The expression unit, a linear piece of DNA, was targeted via single-crossover to the Δ*gla*A locus using the homology in the 2kb 3′- and direct downstream 2kb 3″-*gla*A regions with the identical 2kb-left and 2kb-right flanks of the expression cassette, as described in [119]. All gene sequences were cloned in the *E. coli* vector exactly from start ATG until stop codon.

Shake flask fermentations - *A. niger* strain spores were pre-cultured in 20 ml CSL preculture medium (100 ml flask, baffle). After growth for 18−24 hours at 34° C and 170 rpm, 10 ml of this culture was transferred to Fermentation Medium (FM). Fermentation in FM was performed in 500ml flasks with baffle with 100 ml fermentation broth at 34° C and 170 rpm for the number of days indicated. The CSL medium consisted of (in amount per liter): 100 g Corn Steep Solids (Roquette), 1 g $NaH_2PO_4 \cdot H_20$, 0.5 g $MgSO_4 \cdot 7H_2O$, 10 g glucose$\cdot H_2O$ and 0.25 g Basildon (antifoam). The ingredients were dissolved in demi-water and the pH was adjusted to pH 5.8 with NaOH or H_2SO_4; 100 ml flasks with baffle and foam ball were filled with 20 ml fermentation medium and sterilized for 20 min. at 120° C. The fermentation medium (FM) consisted of (in amount per liter): 150 g maltose$\cdot H_2O$, 60 g Soytone (peptone), 1 g $NaH_2PO_4 \cdot H_2O$, 15 g $MgSO_4 \cdot 7H_2O$, 0.08 g Tween 80, 0.02 g Basildon (antifoam), 20 g MES, 1 g L-arginine. The ingredients were dissolved in demi-water and the pH was adjusted to pH 6.2 with NaOH or H_2SO_4; 500 ml flasks with baffle and foam ball were filled with 100 ml fermentation medium and sterilized for 20 min. at 120° C.

SDS-PAGE electrophoresis - Sample pre-treatment: 30 μl sample was added to 35μl water and 25μl NuPAGETM LDS sample buffer (4×, Invitrogen) and 10μl NuPAGETM Sample Reducing agent (10×, Invitrogen). Samples were heated for ten minutes at 70° C in a thermo mixer. SDS-PAGE was performed in duplicate according to the supplier's instructions (Invitrogen: 4 − 12% Bis-Tris gel, MES SDS running buffer, 35 min. runtime). One of the two gels was used for blotting, 10μl of the sample solutions and 1μl marker M12 (Invitrogen) were applied on the gels (NuPAGETM BisTris, Invitrogen). The gels were run at 200 V, using the XCELL Surelock, with 600 ml 20 times diluted MES-SDS buffer in the outer buffer chamber and 200 ml 20 times diluted MES-SDS buffer, containing 0.5 ml of antioxidant (NuPAGETM Invitrogen) in the inner buffer chamber. After running, the gels were fixed for one hour with 50% methanol / 7% acetic acid (50 ml), rinsed twice with demineralised water and stained with Sypro Ruby (50 ml, Invitrogen) overnight. Images

were made using the Typhoon 9200 (610 BP 30, Green (532 nm), PMT 600 V, 100 micron) after washing the gel for ten minutes with demineralised water. Typical detection limit for the fermentation samples using the described method is around 50 mg/l.

DATA

Two protein data sets were tested for high-level production, one for homologous gene expression (Supplementary Table S1) and one for heterologous gene expression. Proteins in the heterologous data set originated from 14 different fungal donor organisms (Supplementary Table S2–S3). All proteins have a signal peptide (length > 10 amino acids) as predicted by SignalP 3.0 [120], and a total sequence length longer than 100 amino acids. Proteins containing the most common ER retention signal (C-terminal [HK]DEL) and proteins predicted to be transmembrane by both TMHMM [121] and Phobius [122] were filtered out of the data set.

To avoid biasing subsequent analyses, sequence redundancy was reduced using BLASTCLUST [123]. Two sequences were considered redundant when the aligned sequences shared > 40% identity over a length of minimal 90% for at least one of the sequences. From the obtained protein clusters, we selected a representative protein, with the shortest average distance to all other proteins in the cluster, and removed the remainder. If a cluster contained proteins with both positive and negative labels, one positive and one negative protein was selected. This resulted in data sets hom and het containing 345 proteins (178 positives, 167 negatives) and 991 proteins (163 positives, 828 negatives), respectively.

To train a classifier on hom en test it on het, a data set het_{hom} was constructed that contains the het data set without proteins that share > 40% identity with any protein in hom. This data set contained 906 (128 positives, 778 negatives) proteins.

PROTEIN REPRESENTATIONS

Figure 3.1 shows the ten different sequences that were used to represent a protein: r_0) the ORF codon sequence, using a 64 letter codon alphabet; r_1) the N-terminal signal peptide sequence; r_2) the mature protein sequence (excluding the signal peptide); r_3) the predicted solvent accessibility sequence, using B for buried and E for exposed; r_4) the parts of the mature protein sequence predicted to be buried, and r_5) to be exposed, both using the 20 letter amino acid alphabet; r_6) the predicted secondary structure sequence, using H for α-helix, E for β-strand, and C for random coil; r_7) the parts of the mature protein sequence predicted to be in a helix structure; r_8) in a strand structure; and r_9) in a random coil region, all three using the 20 letter amino acid alphabet.

We used randomized versions of the different structural sequences: r'_4) randomized buried sequence, r'_5) randomized exposed sequence, r'_7) randomized helix sequence, r'_8) randomized strand sequence, and r'_9) randomized coil sequence, to test whether their actual amino acid content or just their length is predictive. For example, if for a given protein 50 residues are predicted to be in a helix structure, i.e. the helix sequence has

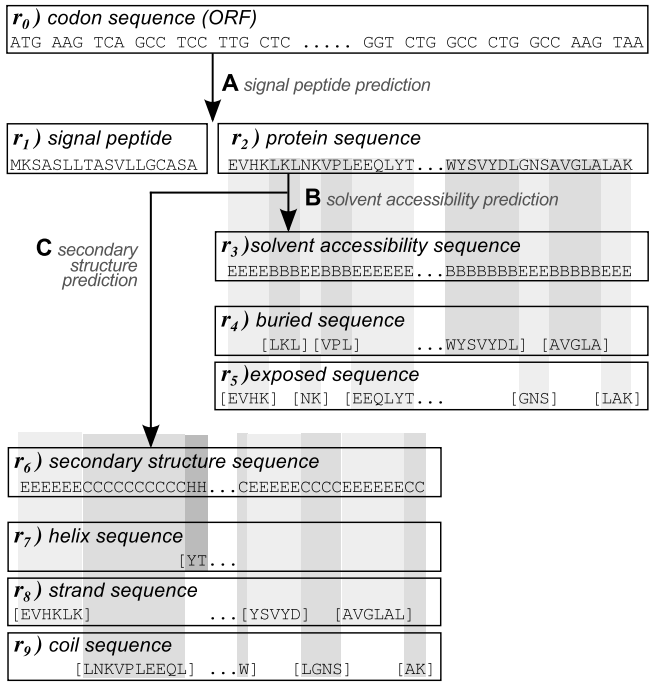


Figure 3.1: Different sequence-based protein representations. The different shades of gray denote predicted buried (B) and exposed (E) regions in case of the the solvent accessibility, and predicted helix (H), strand (E), and random coil (C) region in case of the secondary structure.

length 50, a randomized helix sequence is constructed by selecting 50 residues from the entire protein sequence at random.

STRUCTURAL PREDICTIONS

SignalP 3.0 [120] was used to predict N-terminal signal peptide presence and signal peptide cleavage site. From the neural network output, we used the default D-value threshold (0.43) to decide if a protein contains a signal peptide and used the predicted signal peptide cleavage site to split a protein sequence into a signal peptide part and a mature protein sequence part (Figure 3.1A). NetSurfP 1.0 [124] was used to predict structural location (either buried or exposed) of each amino acid in a mature protein sequence (Figure 3.1B). PsiPred 3.21 [125] was used to predict secondary structure of the mature protein sequence, using UniRef90 as a database (Figure 3.1C).

CLASSIFICATION

A linear support vector machine (LIBSVM [110]) was used for classification [126], in which the prediction y is a weighted combination of kernels $K(s_i, z)$ between the training objects i and a test object z:

$$y = \sum_{s_i \in S} \alpha_i y_i \Phi(s_i)\Phi(s_z) = \sum_{s_i \in S} \alpha_i y_i K(s_i, s_z) \tag{3.1}$$

For each object (protein) i, α_i is the weight assigned to the object as obtained from the trained classifier ($0 < \alpha_i \leq 1$ if the object is a support vector, $\alpha_i = 0$ otherwise), y_i the class label (-1 or 1), s_i the sequence of protein i, and $\Phi(s_i)$ a mapping from sequence to feature space. The SVM is trained by optimizing a quadratic programming problem:

$$\max_{\vec{\alpha}} \sum_{s_i \in S} \alpha_i - \frac{1}{2} \sum_{s_i \in S} \sum_{s_j \in S} y_i y_j \alpha_i \alpha_j K(s_i, s_j) \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C \; \forall i \quad \text{and} \quad \sum_{s_i \in S} \alpha_i y_i = 0 \tag{3.2}$$

The parameter C, controlling the trade-off between training error and classifier complexity, was optimized using a simple grid search over $1.0 \times 10^{-6}, 1.0 \times 10^{-5}, \ldots, 1.0 \times 10^{6}$. Classifier performance on a data set was estimated by running a double 10-fold cross-validation (CV) loop, in which C was optimized in an inner CV-loop on the training set. As performance measure we used the area under the receiver-operator characteristic curve (AUROC) [127]. Classifier performance is defined as the average AUROC over the CV-loops. When separate training and test sets are used, a classifier was trained on the first data set, optimizing C in a 10-fold CV-loop, and tested on the second data set, again using the AUROC as performance measure.

In the cross-validation error estimation procedure, a predictor is repeatedly trained on 90% of the data set and tested on the remaining 10% of the data set. If features derived from a training set that are important for discriminating between the positive and negative class also yield good performance on the test set, then these features apparently allow good generalization. In this sense, a good CV performance can be interpreted as an *in silico* validation of the features found.

CLASSIFIER INTERPRETATION AND COMPARISON

For a given set of sequences S, the feature weight vector $\mathbf{w}$ from a trained SVM classifier was obtained using:

$$\mathbf{w} = \sum_{s_i \in S} \alpha_i y_i \Phi(s_i). \tag{3.3}$$

Classifiers were compared by taking the correlation between $\mathbf{w}$ of both trained classifiers.

A high correlation indicates a high similarity between the classifiers, both assigning similar weights to the same features.

FEATURE SETS

We derived distinct sets of sequence-based features, f_0–f_{22}, which will be described below. A visualization of feature matrices f_0, f_1, f_2, and f_{12} for hom and het are given in Supplementary Figures S1–S8. Features f_1–f_{14} were used in an inner product kernel ($K(\vec{x}, \vec{y}) = \vec{x}^T \vec{y}$); for features f_{15}–f_{22} we used a spectrum kernel (see below).

COMPOSITION-BASED FEATURES:

$f_0 - f_9$) The composition of sequences $r_0 - r_9$ (Figure 3.1). For a sequence s on alphabet A, the composition $\mathbf{c}$ is defined as:

$$\mathbf{c}(s) = \frac{\text{count}(l, s)}{|s|} \quad \forall l \in A, \tag{3.4}$$

in which count(l, s) is a function that counts the number of occurrences of letter l in sequence s, and $|s|$ is the length of the sequence. The size of the feature vector $\mathbf{c}$ depends on the size of alphabet A, e.g. the composition of the codon sequence r_0 results in a feature vector of length 64 and the composition of the protein sequence r_2 results in a feature vector of length 20. This means that f_0 and f_2 consist of 64 and 20 features respectively. f_4', f_5', f_7', f_8' and f_9' are the compositional features of the randomized sequences r_4', r_5', r_7', r_8' and r_9'

f_{10}) Predefined amino acid cluster composition of r_2 using the 11 predefined clusters in Table 3.1. The clusters are based on those defined in [128].[1] For a sequence s and clusters G, the cluster composition vector $\mathbf{cc}$ is defined as:

$$\mathbf{cc}(s, G) = \frac{\sum_{l \in g} \text{count}(l, s)}{|s|} \quad \forall g \in G. \tag{3.5}$$

f_{11}) Optimized amino acid cluster composition of the protein sequence (r_2) using clusters that are optimized for our data set using the method described in the next section (Amino acid clustering).

[1] In this clustering it sometimes occurs that an amino acid is both inside and outside a cluster, based on its state; e.g. a free cysteine is in the polar cluster, while a cysteine that forms a disulfide bridge is outside the polar cluster. Without structural data, amino acid states are unknown. We therefore removed an amino acid from the cluster if it also resides somewhere outside that cluster, i.e. cysteine is not considered to be part of the polar cluster.

Table 3.1: Predefined amino acid clusters.

cluster	amino acids
small	V, C, A, G, T, P, S, D, N
polar uncharged	S, W, N, Q, T, Y
aromatic	F, Y, W, H
acidic	D, E
charged	H, K, R, E, D
basic	K, R, H
hydrophobic	I, L, V, M, F, Y, W, H, C, A, T, K
tiny	A, G, S
nonpolar	A, V, L, I, M, G, F, P
aliphatic	I, L, V
polar	Y, W, H, K, R, D, E, T, S, N, Q

SEQUENCE-DERIVED FEATURES:

f_{12}) Using r_0, codon usage was calculated for the 59 codons that non-uniquely encode for an amino acid. Codon usage is defined as the codon count divided by the amino acid count of the amino acid it encodes for.

f_{13}) Four other sequence-derived features: the signal peptide length, the protein sequence length, the codon adaptation index [104] that was calculated using a codon usage index derived from all *A. niger* genes, and the isoelectric point. The last two values were calculated using the codon sequence (r_0) and the protein sequence (r_2) respectively, both using the Biopython software package [129].

SELECTED FEATURES:

f_{14}) A two-sample t-test (python SciPy package [130]), was applied to a set of 124 features, combining the features from feature sets f_0, f_1, f_2, f_3, f_6, f_{10}, and f_{13}. Features with a p-value $< 1.0 \times 10^{-4}$ were selected for forward feature selection, 36 and 33 features for hom and het respectively (Supplementary Table S4).

In a 10-fold cross-validation loop, forward feature selection was applied on the training set. Features were added one by one, based on their prediction performance as determined using a second inner 10-fold CV-loop, until prediction performance starts to drop. To reduce calculation time, parameter C was not optimized but based on observations fixed to 1.0×10^3 and 1.0×10^{-6} for hom and het, respectively. The selected features per CV-loop for both hom and het are given in Supplementary Table S5.

PATTERN-BASED FEATURES:

f_{15}, f_{16}, ..., f_{22}) We employed spectrum kernels [88], which define similarities between sequences based on fixed-length subsequence (k-mer) counts, as implemented by Shogun [131] to search for predictive patterns. We calculated $k = 2,3,4,5$ spectrum kernels using r_2 (f_{15}, ..., f_{18}) and r_1 (f_{19}, ..., f_{22}).

AMINO ACID CLUSTERING

We developed a method that forms amino acid clusters using our data sets, thereby constructing new features optimized for our data. A cluster is defined as a set of one or more amino acids. For the resulting clusters, each amino acid can be in one cluster only, not every amino acid needs to be in a cluster.

The method starts with selecting the best performing amino acid, i.e. the amino acid that, when used as the only feature, provides the best classification performance, the same as in forward feature selection. For example, if the fraction of lysine in a protein provides the best separation between the positive and negative class, this amino acid will start the first cluster. In the next iteration, for the remaining 19 amino acids, classification performance is tested for two cases: 1) with the amino acid added as new cluster and 2) with the amino acid added to the existing cluster. In case of the example, when adding alanine, classification performance is tested both using the fraction of lysine and the fraction of alanine as two separate features, and using the sum of the fractions of lysine and alanine as a single feature. The case that provides the best classification performance is selected. In the next iteration, with 18 amino acids remaining, the same procedure is applied. This iteration cycle is proceeded until there are no more amino acids left. Finally, the overall best performing clusters are the output of the method. Consequently, it might happen that some amino acids will not be selected at all.

This procedure is implemented in a 10-fold CV-loop, obtaining the best performing clusters on the training set and using them as cluster composition features on the test set. The selection protocol is applied in an inner CV-loop to avoid biases towards the training data. The obtained clusters per CV-loop are given in Supplementary Table S6.

STATISTICAL PATTERN DISCOVERY

The statistical motif finding approach MEME [132] was used to find patterns (described as position-dependent letter-probability matrix) that occur once in every sequence (`oops` mode) of a data set. Discriminative motif discovery was performed using the successful secreted proteins as input with the unsuccessful secreted proteins as negative sequences and vice versa. This was done for both *hom* and *het*. The minimal and maximal motif lengths were set to 2 and 15 respectively.

3.3. RESULTS

SEQUENCE DATA IS PREDICTIVE FOR HIGH-LEVEL PROTEIN PRODUCTION

To test if the sequence data is informative, we used it to predict successful high-level protein production. Classifiers were built using an extensive set of sequence-based features. Performance results (AUROC) of 10-fold CV experiments on both *hom* and *het* are shown in Table 3.2, 0.5 indicating random prediction and 1.0 perfect prediction. Best classification performances of 0.85 and 0.75 AUROC respectively (boldface in Table 3.2) show that sequence data is predictive. As additional support, classifier outcome for the *A. niger* proteome (Supplementary Figure S11) shows an expected result, predicting successful high-level production for only a fraction of the proteome.

Considering the composition-based features, similar results were observed for the codon sequence (f_0) and the protein sequence (f_2), which is expected because of the relation between the two sequences. For *hom*, high performance using protein sequences is in line with results of our previous work [99]. Similarly, results of other previous work, regarding only protein localization and not production rate, reported different amino acid compositions for intra- and extracellular proteins [133, 134]. Although the codon sequence shows a slightly higher score for *hom*, it does not significantly outperform the protein sequence ($p = 0.14$ for a paired t-test on the test scores of the 10 CV-loops).

The predictive power of the amino acid composition of the signal peptide (f_1) proves to be limited, clearly outperformed by both the codon and protein sequence. More advanced methods, taking into account letter/pattern location [135, 136], did not improve prediction results (results not shown).

SIMILAR CHARACTERISTICS ARE IMPORTANT FOR BOTH DATA SETS

Figure 3.2 shows the ROC-curves of the composition-based classifiers discussed thus far. Figure 3.2A and Figure 3.2B show the average result of a 10-fold CV-loop on *hom* and *het* respectively. Figure 3.2C shows the result of a classifier trained on *hom* and tested on *het*. Remarkably, this shows similar results as the classifiers trained on *het*, suggesting that the homologous classifier generalizes well to predict high-level production for *het*. In fact, the classifiers trained on *het* performed even slightly worse. This might be due to the fact that this data set is too heterogeneous, originating from 14 different species, which makes it harder to build a generic classifier and may have caused over-fitting in the CV-loops.

The good generalization of the *hom* classifier on the *het* data set suggests that classifiers trained on *hom* and *het* are similar, i.e. perform their predictions based on the same sequence characteristics. The correlation of 0.65 in Figure 5.3 shows that this is indeed the case. The figure shows the contribution of each amino acid as obtained from the *hom* and *het* classifier, both trained using the protein amino acid composition (f_2). Positive values indicate contribution to successful high-level production and negative values indicate contribution to unsuccessful high-level production.

Table 3.2: Prediction performance scores (AUROC)

features	$hom \rightarrow hom$	$het \rightarrow het$	
Composition-based features			
f_0	**0.85**	0.70	*ORF codon composition*
f_1	0.66	0.51	*signal peptide AA composition*
f_2	0.83	0.70	*mature protein AA composition*
f_3	0.68	0.51	*buried-exposed composition*
f_4 (f_4')	0.80 (0.80)	0.65 (0.64)	*buried AA composition*
f_5 (f_5')	0.82 (0.78)	0.64 (0.65)	*exposed AA composition*
f_6	0.62	0.57	*helix-strand-coil composition*
f_7 (f_7')	0.68 (0.70)	0.60 (0.57)	*helix AA composition*
f_8 (f_8')	0.70 (0.72)	0.61 (0.57)	*strand AA composition*
f_9 (f_9')	0.80 (0.80)	0.65 (0.65)	*coil AA composition*
f_{10}	0.80	0.63	*AA clusters composition*
f_{11}	0.83	0.67	*optimized AA clusters comp.*
Sequence-derived features			
f_{12}	0.64	0.54	*codon usage*
Selected features			
f_{14}	0.84	**0.75**	*feature selection*
Pattern-based features			
f_{15}	0.82	0.63	*2-mer counts protein*
f_{16}	0.77	0.61	*3-mer counts protein*
f_{17}	0.68	0.60	*4-mer counts protein*
f_{18}	0.57	0.47	*5-mer counts protein*
f_{19}	0.63	0.54	*2-mer counts signal peptide*
f_{20}	0.59	0.52	*3-mer counts signal peptide*
f_{21}	0.54	0.51	*4-mer counts signal peptide*
f_{22}	0.56	0.50	*5-mer counts signal peptide*

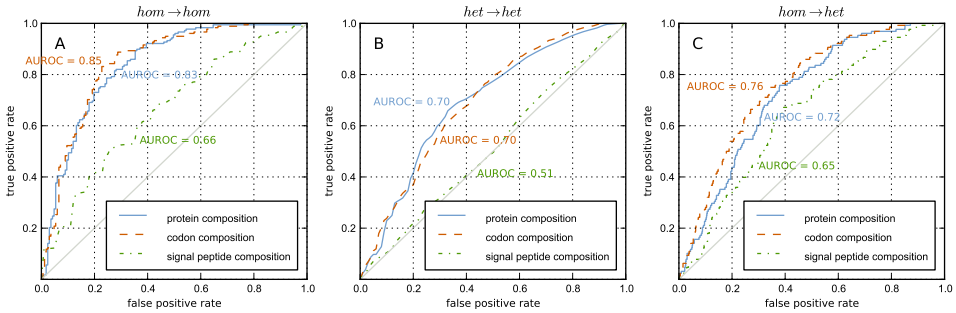


Figure 3.2: Classification performances. ROC-curves of composition-based classifiers using the codon sequence (f_0), the signal peptide sequence (f_1), and the protein sequence (f_2). Performances are shown for classifiers **A)** trained and tested on *hom*, **B)** trained and tested on *het*, and **C)** trained on *hom* and tested on *het*.

For both *hom* and *het*, a remarkable positive and negative contribution of respectively tyrosine (Y) and methionine (M) is apparent. For *hom*, also asparagine (N) and lysine (K) show an outstanding positive and negative contribution respectively. Considering amino acid properties, it is observed that the basic and the sulfur-containing amino acids have a negative contribution whereas the (uncharged) aromatic amino acids have a positive contribution.

Besides comparing the amino acid contributions of the *hom* and *het* classifier, we also compared them to amino acid synthesis costs as defined in [137]. With the exception of the aromatic amino acids, a negative correlation is shown between the *hom* contributions and the amino acid costs (Supplementary Figures S9–S10), suggesting a preference for "cheap" amino acids for high-level secretion.

BASIC AND AROMATIC AMINO ACIDS ARE PREDICTIVE

From a structural and functional perspective, it is often more useful to look at the physicochemical properties of an amino acid, rather than looking at the 20 amino acids as different entities. Therefore, based on physicochemical properties [128], we defined 11 predefined amino acid clusters (Table 3.1), and used these as features (f_{10}). In this case, a correlation of 0.71 was observed between the *hom* and *het* classifier (Figure 5.3B). The aromatic amino acids have a high contribution to high-level production, which, looking back at Figure 5.3A, is similar to the amino acid contributions, except for the positively charged histidine (H). A negative contribution is observed for the basic amino acids, also consistent with the observations in Figure 5.3A.

Since it is unclear which amino acid clusters are suitable for what problem, we developed a novel method that uses the data set to construct clusters. The best performing clusters (f_{11}) obtained in ten CV-loops are jointly shown as a heat map in Figure 3.4. The non-diagonal values show the number of times that two amino acids were found in the same

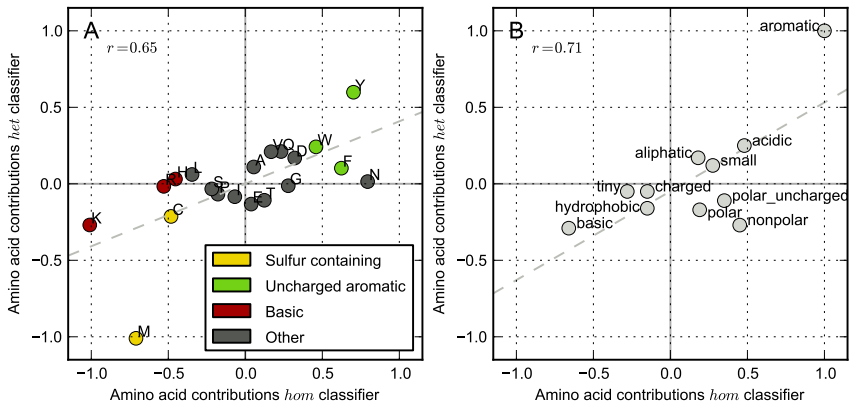


Figure 3.3: Comparing hom and het classifiers. Amino acid contributions obtained from hom and het trained classifiers are the x- and y-values respectively, the correlation is denoted by r. Contributions are normalized per classifier (axis): each contribution is divided by the maximum absolute contribution. The plots show the contributions obtained from classifiers trained using **A)** the protein amino acid composition (f_2) and **B)** the predefined amino acid cluster composition (f_{10}).

cluster. The diagonal values show how often an amino acid was found in any cluster.

The diagonal values correspond to the results observed in Figure 5.3A: highly contributing amino acids were often found in a cluster. For het, noteworthy exceptions are phenylalanine (F) and glycine (G), both of which always ended up in a cluster despite their low contribution.

The non-diagonal values also match the results in Figure 5.3A. As can be observed, amino acids with a positive contribution (green letters) and amino acids with a negative contribution (red letters) often form clusters, whereas amino acids with contradicting contributions rarely do. The occurrence of only few light cells show that not many amino acids consistently form the same cluster. Only clusters with phenylalanine (F), glycine (G), aspartic acid (D), and glutamine (Q) occur relatively often in both data sets, but those do not share an obvious physicochemical property. Despite the high contributions observed for the aromatic amino acids in Figure 5.3B, clustering of these amino acids occurred only a few times.

STRUCTURAL SUBSEQUENCES HAVE LIMITED INFORMATION

The secondary structure composition (f_6) shows to be little predictive, with an AUROC of 0.62 and 0.57 for hom and het respectively. Results using the amino acid composition of the helix (f_7), strand (f_8), and coil sequence (f_9) suggest that the coil sequence is more informative than the helix and strand sequence, however, a similar result was obtained using a randomized version of the sequence (f_9', score between brackets in Ta-

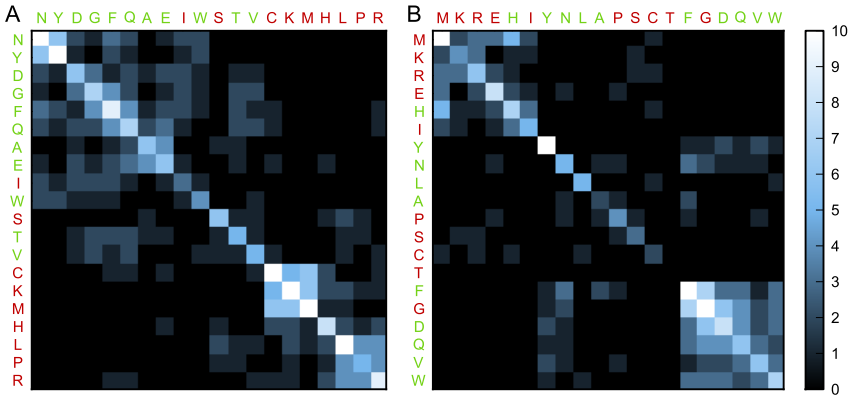


Figure 3.4: Best performing amino acid clusters. The heat maps show the combined result of the best performing clusters obtained in 10 CV-loops for both *hom* (**A**) and *het* (**B**). The values on the diagonals denote how often an amino acid ended up in a cluster (due to selecting the optimal clusters, amino acids might not be selected at all). The colors on the non-diagonal places denote how often two amino acids ended up in the same cluster. Complete linkage hierarchical clustering was used to cluster the heat map, using the euclidean distance as distance measure. The color of the amino acid letters indicates if the amino acid has a positive (green) or negative (red) contribution in Figure 5.3A.

ble 3.2). This indicates that the coil sequence, although it provides higher classification performance, is not more informative than the helix and strand sequence. The better performance can be explained by the length of the sequence, proteins are on average composed of 60% coil, 20% helix, and 20% strand.

Considering the solvent accessibility, the distribution of buried and exposed amino acids (f_3) is only predictive for *hom* (AUROC 0.68). The buried amino acids showed a positive contribution to high-level production (data not shown). Results using the amino acid composition of the buried (f_4) and exposed sequence (f_5), separately, are similar to the randomized buried (f_4') and randomized exposed sequence (f_5'), indicating that neither of the two sequences is more informative than a randomly selected sequence of the same length.

BEST PERFORMANCE WITH ONLY FEW FEATURES

Thus far, all discussed classifiers were trained on a relatively small set of related features. Combining all features results in a large feature set which complicates both classification and interpretation. To resolve this, we used a forward feature selection protocol similar to the one used in previous work [99].

A classification performance of 0.84 AUROC was obtained for *hom* (f_{14} in Table 3.2), similar to the results obtained using the protein's amino acid or codon composition. Interpretation of the selected features shows a similar trend compared to the amino acid

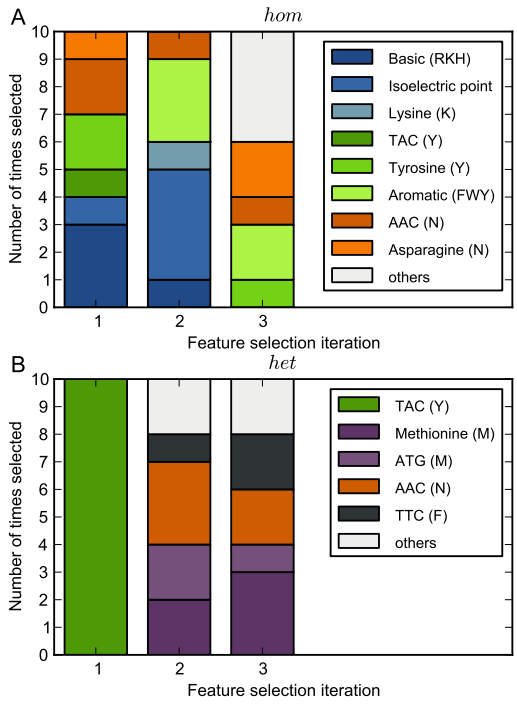


Figure 3.5: Feature selection - For the first three feature selection iterations (x-axis), the bar plot shows how often features were selected in the 10 CV-loops for both hom (**A**) and het (**B**). Features with a different shade of the same color are correlated ($r > 0.65$). The letters between brackets in the legend are amino acids that denote either which amino acids are in the cluster, e.g. the basic cluster contains amino acids R, K, and, H, or for which amino acid a codon encodes, e.g. codon TAC encodes for Y.

contributions observed in Figure 5.3A. As shown in Figure 3.5A, the first three selected features were almost always lysine (K), tyrosine (Y), and asparagine (N), or, as shown by a different shade of the same color, a correlated feature ($r > 0.65$).

For het, feature selection resulted in the best obtained prediction performance of 0.75 AUROC (f_{14}). A relatively low number of features, on average six, were selected each CV-loop, most of which were codons. Remarkably, the codon TAC (Y) was consistently selected first (Figure 3.5B). Methionine (M and ATG) and the codons AAC (N) and TTC (F) were most often selected second and third.

The fact that codons are selected before amino acids suggests the importance of codon usage. However, taking codon usage as features provided an AUROC of only 0.54 (f_{12}). This could be due to the heterogeneous codon usage of the different organisms in het. With an AUROC of 0.64, codon usage in hom appeared to be a little predictive.

A ROLE FOR N-GLYCOSYLATION MOTIFS

Functional patterns, often called (short) linear motifs [138] (SLiMs), have been associated with protein targeting. The most well-known example is the C-terminal [HK]DEL motif that causes ER retention. Also a case with a secretion specific signal has been identified [139].

All proteins in our data sets contain a signal peptide, proteins with an ER-retention signal and proteins with predicted transmembrane regions are filtered out. Still, successful high-level production was observed for only half of the proteins in *hom*. Unsuccessful high-level production could for example be caused by a low production rate or a high degradation rate, resulting in a too low concentration to detect on the gel (i.e. $< 50\,mg/l$). An alternative explanation could be the existence of additional retention or targeting signals. The statistical motif finding approach MEME [132] was used to search for such signals.

For *hom*, the pattern $N[GI]T$, which matches the N-glycosylation pattern $N[\hat{}P][ST]$, was found for successful high-level production. Instead of retention or targeting, this indicates importance of this post-translational modification. No other patterns related to either successful or unsuccessful high-level production were found, indicating the absence of additional generic targeting or retention signals.

SLiMs related to post-translational modifications can occur more than once in a sequence. Therefore we also searched for reoccurring patterns by building classifiers using fixed length pattern (k-mer) counts as features. Using the signal peptide and the protein sequence, results for $k = 2$ to $k = 5$ are shown in Table 3.2 ($f_{15} - f_{22}$). In general, classification performances rapidly drop with increasing pattern length, caused by an explosion of the number of possible k-mers that results in sparse kernels [126] which are difficult to use for classification. Again, the N-glycosylation pattern was identified. Inspection of the 3-mer classifier trained on *hom* showed that six out of the seven 3-mers with the most positive contribution match the N-glycosylation pattern.

The N-glycosylation pattern is much more abundant in *hom* than in *het*, with on average 3.37 N-glycosylation patterns per protein for *hom* compared to an average of 1.42 per protein for *het*. A clear difference is observed between the positive and negative proteins in *hom*, containing an average of 4.71 and 1.95 patterns respectively. Although much smaller, with an average of 1.72 and 1.36 patterns for the positive and negative class, respectively, *het* shows a difference as well, suggesting that the addition of N-glycosylation sites might be useful to improve heterologous secretion [140].

3.4. DISCUSSION

Using machine learning techniques, we explored which combinations of a large number of features best helps predicting successful high-level protein production in *A. niger*. The results show that composition-based features were most predictive, but that the exact representation – by codons, amino acids or amino acid clusters – has little influence.

Taking into account predicted structural location of the amino acids did not further improve prediction results. Although all proteins have a signal peptide and the signal peptide is usually cleaved off in the ER [141], its sequence is still somewhat predictive. This suggests a role for the signal peptide in determining translocation efficiency, possibly due to a higher affinity to the SRP.

Classifiers trained on *hom* and *het* showed similar amino acid contributions, indicating that the properties found important for high-level production are generic in nature. The fact that poorer prediction performance was still obtained for *het* suggests that organism-specific properties may be important for high-level production. However, the heterogeneous nature of the *het* data and the resulting limited number of samples per donor organism hinder the identification of such properties using machine learning.

Feature selection on a larger set of features, including some derived from the sequence, confirmed that mainly composition-based features were selected in the first iterations. In fact, mainly codons and only a few amino acid features were selected for *het*. In the first three iterations, only codons were selected, implying room for production improvement by codon adaptation of heterologous proteins.

Among the composition-based features, a number of individual amino acids stood out as strongly contributing, either positively or negatively, to predicted high-level production:

Tyrosine (Y), tryptophan (W) and phenylalanine (F) contribute positively. These aromatic amino acids are usually found in the protein core; their ability to form stacks can contribute to protein stability. A correlation between protein stability and secretion efficiency has been observed [142–144]. Moreover, improving secretion by increasing the protein stability is shown to be a successful strategy [145, 146]. It is hypothesized that proteins with a high stability more frequently escape from the ER quality control system, since they will more often be in the correctly folded state, which in general is the only state to leave the ER [143, 147].

Asparagine (N) has a high positive contribution for *hom*. Since motif analysis showed the N-glycosylation pattern to be both predictive and abundant in *hom* the contribution of asparagine could be related to this post-translational process in which a specific set of enzymes catalyzes the formation of N-linked glycans. Details are still unknown, but N-linked glycans are known to play an import role in protein folding and quality control [148]. Although N-linked glycosylation is not a prerequisite for secretion [149], there is ample evidence that introduction or modification of glycosylation sites can lead to improved secretion [140, 150, 151].

Methionine (M) shows a strong negative contribution. The fact that it is a sulfur-containing amino acid, and that the other sulfur-containing amino acid, cysteine (C) also has a negative contribution, suggests a negative influence of sulfur-containing amino acids. Another explanation could be that methionine is encoded by the start codon ATG, which could slow down translation due to ribosome reinitiation on alternative start sites [152].

Lysine (K) also has a strongly negative contribution, as do the other basic amino acids

arginine (R) and histidine (H) for *hom*. The positive charge, usually exposed at the protein surface, could facilitate binding to the negatively charged cell membrane, thereby preventing the protein to be filtered out, or could be related to protein thermostability due to charge-charge interactions on the protein surface [153].

In conclusion, we have exploited a large experimental dataset on production of proteins in *A. niger*, using both homologous and heterologous gene expression and employed machine learning algorithms to find combinations of features optimally predictive of presence or absence of high-level production. These features were all derived directly or indirectly from the protein sequences, and could be useful to improve industrial production rates of existing targets and to explore possibilities for new products. In future work, we intend to verify a number of the hypotheses provided here by engineering proteins to better reflect the features found to be related to high production rates.

3.5. SUPPORTING INFORMATION

Supplementary figures and tables are accessible online[2].

[2] http://www.plosone.org/article/info:doi/10.1371/journal.pone.0045869#s5

4

SPICE: A WEB-BASED TOOL FOR SEQUENCE-BASED PROTEIN CLASSIFICATION AND EXPLORATION

Bastiaan A van den Berg, Marcel JT Reinders,
Johannes A Roubos, Dick de Ridder

Published in BMC BioInformatics, Volume 15, Issue 93: page 1–10, 2014, doi:10.1186/1471-2105-15-93.

ABSTRACT

Amino acid sequences and features extracted from such sequences have been used to predict many protein properties, such as subcellular localization or solubility, using classifier algorithms. Although software tools are available for both feature extraction and classifier construction, their application is not straightforward, requiring users to install various packages and to convert data into different formats. This lack of easily accessible software hampers quick, explorative use of sequence-based classification techniques by biologists.

We have developed the web-based software tool SPiCE for exploring sequence-based features of proteins in predefined classes. It offers data upload/download, sequence-based feature calculation, data visualization and protein classifier construction and testing in a single integrated, interactive environment. To illustrate its use, two example datasets are included showing the identification of differences in amino acid composition between proteins yielding low and high production levels in fungi and low and high expression levels in yeast, respectively.

SPiCE is an easy-to-use online tool for extracting and exploring sequence-based features of sets of proteins, allowing non-experts to apply advanced classification techniques. The tool is available at `http://helix.ewi.tudelft.nl/spice`.

4.1. BACKGROUND

The sequence of a protein contains valuable information about its characteristics. Various sequence-based prediction methods exploit this to classify proteins according to specific properties, such as localization [154], function [155], or solubility [72]. This has resulted in relevant and frequently used bioinformatics tools [64] that are offered by a growing number of easily accessible websites and webservices [1,2,3].

Sequence-based protein classifiers assign class labels to proteins based on a set of features, real numbers that capture some sequence property. This process entails three distinct steps. First, *feature extraction* is required to map protein sequences to points in a feature space (Figure 4.1A). Next, a classifier is constructed to optimally separate protein classes in this feature space (*training*, Figure 4.1B), using a set of proteins with known class labels. Finally, the trained classifier can be applied to predict class labels for new proteins (*testing*, Figure 4.1C). Additionally, features and feature distributions can be visualized to explore differences between protein classes by eye.

Software tools are available for each of these three steps. Feature extraction is available as software package [156] and through web services [95, 157–159] and an extensive range of classification software has been developed [131, 160], some of which include feature visualization [161]. However, combined application requires installing different

[1] EBI bioinformatics services: http://www.ebi.ac.uk/services
[2] CBS Prediction Servers: http://www.cbs.dtu.dk/services
[3] PredictProtein: http://ppopen.informatik.tu-muenchen.de

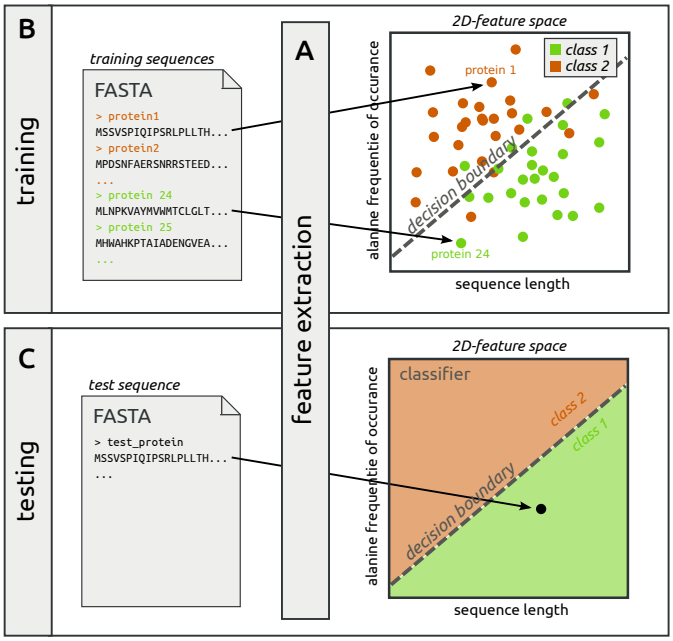


Figure 4.1: Protein classification. *A)* Feature extraction maps protein sequences to feature space. In this case, calculation of the sequence length (x-axis) and the relative frequency of occurrence of alanines (y-axis) map each protein sequence to a point in two-dimensional feature space. *B)* Classifier training using proteins with known class labels: class 1 (orange) and class 2 (green). After mapping to feature space, a classifier is trained to obtain a decision boundary (dashed line) that optimally separates the classes. *C)* Predicting class labels of new proteins using the trained classifier. After mapping to feature space, the point in feature space determines what label is assigned to the protein. Label class 1 will be assigned to the example protein, because of its location on the class 1 side of the decision boundary.

software packages and programming efforts to convert data according to the requirements of each tool. For the construction of specialized high-performance classifiers, the overhead of deploying such a pipeline may be acceptable or even required, because this usually involves extensive exploration of many combinations of (customized) features, types of classifiers, and parameter settings. However, it precludes easy access to these methods for non-expert users.

We set out to offer basic protein classification functionality in a single environment to allow for quick and easy exploration of user-defined protein classes, without the need for any programming, data conversion or software installation. To this end we introduce SPiCE, a web-based tool for Sequence-based Protein Classification and Exploration. SPiCE makes powerful data exploration techniques accessible to non-experts; additionally, expert bioinformaticians can exploit the back-end software to perform customized and/or computationally expensive tasks on a local computer.

Table 4.1: Offered classifiers with corresponding parameter ranges

Classifier	Parameter optimization grid
SVM (linear kernel)	$C = 10^{-3}, 10^{-2}, \ldots, 10^{3}$
SVM (RBF kernel)	$C = 10^{-1}, 10^{0}, 10^{1}$
	$\alpha = 10^{-1}, 10^{0}, 10^{1}$
k-neighbors (unif.[1])	$k = 1, 2, \ldots, 5, 10, 20, \ldots, 50, 100$
k-neighbors (dist.[2])	$k = 1, 2, \ldots, 5, 10, 20, \ldots, 50, 100$
Nearest centroid	$r = 1, 2, \ldots, 10$
LDA[3] classifier	-
QDA[4] classifier	-
Gaussian Naive Bayes	-
Decision Tree	*default scikit-learn parameters*
Random Forest	*default scikit-learn parameters*

[1]uniform resp. [2]distance-based neighbor weights
[3]linear discriminant resp. [4]quadratic discriminant analysis

4.2. IMPLEMENTATION

Before describing the SPiCE functionality, some classification concepts and the offered sequence-based features will be introduced in the following two sections.

CLASSIFICATION

Classifiers are algorithms that assign discrete class labels to objects. These objects are typically represented as vectors of features, real numbers that reflect a property thought to be potentially different for proteins in the different classes. Protein sequences should therefore first be mapped onto such feature vectors, a process called *feature extraction* (Figure 4.1A). This should ideally result in a small number of discriminative features. In SPiCE, feature vectors are always normalized to zero mean and unit standard deviation.

Given a *training set* of proteins with known labels, a classifier can then be trained, i.e. its parameters can be tuned to yield optimal performance (Figure 4.1B). For problems with two classes A and B, performance is often estimated based on a receiver-operator characteristic (ROC) curve. Such a curve represents all possible trade-offs between classifications of proteins in class A as being in class B and vice versa. If class A corresponds to "positive" and class B to "negative", the ROC curve is traditionally drawn as false positive rate vs. true positive rate and the area under the ROC curve (AUC) is used as a measure of classifier performance, with 1 indicating perfect classification and 0.5 random classification. Once trained, the trained classifier can be used to predict the class label for any new protein, a process called *testing* (Figure 4.1C).

To avoid overtraining, i.e. setting the parameters such that the training set is classified

Table 4.2: Sequence-based feature categories

Feature category	Parameters	Number of features
Composition features		
AA composition*	number of segments	20× number of segments
Dipeptide composition	number of segments	400× number of segments
Terminal end amino acid count	N- or C-terminal end, length	20
SS composition*	number of segments	3× number of segments
Per SS class AA composition*	-	3×20
SA composition*	number of segments	2× number of segments
per SA class AA composition*	-	2×20
Codon composition	-	64
Codon usage	-	64
Protein length	-	1
Property profile-based features		
Signal average	AA scale(s), window, edge	1 per AA scale
Signal peaks area	AA scale(s), window, edge, threshold	2 per AA scale
Autocorrelation	type, AA scale(s), distance	1 per AA scale
Pseudo AA composition (type 1)*	AA scale(s), λ	$20 + \lambda$
Pseudo AA composition (type 2)*	AA scale(s), λ	$20 + \lambda$
Amino acid distance-based features		
Property CTD*	property	21
Quasi-sequence-order	AA distance matrix, λ	$20 + \lambda$

*AA: amino acid, SS: secondary structure, SA: solvent accessibility, CTD: composition, transition, distribution

well but test samples will be classified poorly, a stratified cross-validation scheme is used. This entails splitting the training set in k parts reflecting the original class distributions (where the "fold" k is a parameter) and iteratively training classifiers on $k-1$ parts and estimating its performance on the remaining part. The average performance is then an estimate of the performance to be expected on new, unseen data.

A large number of classification algorithms are available, differing in complexity and often applicable to specific problems. SPiCE implements the most well-known classifier types (see Table 4.1). In case the classifier has parameters, they are optimized in an inner k-fold cross-validation loop [108] using the parameter ranges in Table 4.1 as search grid, optimizing for the AUC.

For a more in depth discussion of classification and feature extraction, the reader is referred to relevant reviews [55, 162] or textbooks [163, 164]. Below, an overview of the specific features SPiCE extracts from protein sequences is given.

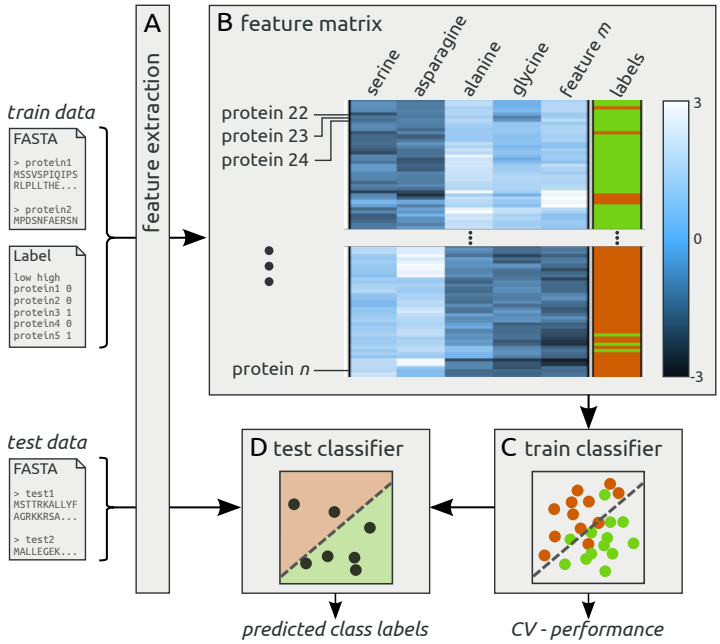


Figure 4.2: Overview of the four main functionalities. *A)* Sequence-based feature extraction, mapping each protein in a FASTA file to a list of feature values (a row in the feature matrix). The uploaded protein labels will be used for classifier construction. *B)* Visual inspection of the calculated feature data, in this example showing (part of) the feature matrix in the form of a clustered heat map with in each row the feature values of one protein and the corresponding protein labels in the rightmost column. *C)* Classifier construction using the calculated feature matrix and the provided labels (train data). A *k*-fold cross-validation protocol is used to assess classification performance. *D)* The trained classifier can be used to predict class labels for a set of new proteins (test data).

SEQUENCE-BASED FEATURES

Table 4.2 lists the feature categories that can be calculated; these categories are briefly discussed below. More details can be found on the SPiCE documentation page [4].

COMPOSITION FEATURES

These features calculate letter counts divided by sequence length for a number of sequence types: amino acid, codon, secondary structure, and solvent accessibility. The 'number of segments' parameter subdivides sequences into equal length parts and returns the composition of each segment separately. For the amino acid sequence, there is also the option to calculate the dipeptide composition, i.e. amino acid pair counts divided by sequence length−1, and the amino acid counts for a given length of the N- or C-terminal end of the protein sequence. For the codon sequence, the codon usage can

[4] http://helix.ewi.tudelft.nl/spice/doc

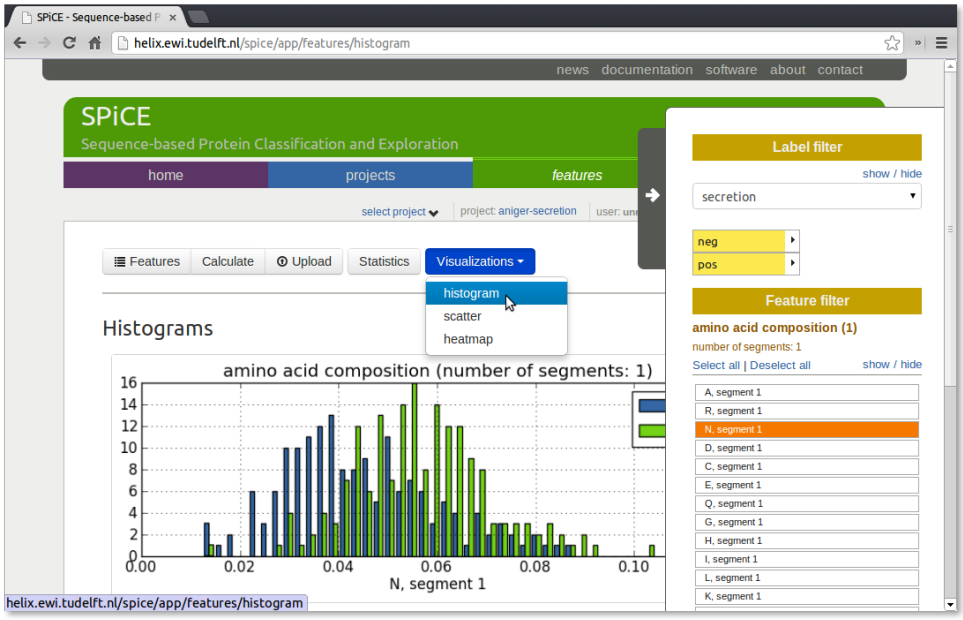


Figure 4.3: SPiCE screenshot.

be calculated.

PROPERTY PROFILE-BASED FEATURES

Amino acid scales map each amino acid to a value that captures a physicochemical or biochemical property, such as hydropathicity or size. These scales are used to obtain a property profile for a protein sequence by mapping all of its residues to the corresponding values. The profiles are in turn used for calculating property profile-based features. The AAIndex data base [97] contains a large collection of scales that can be selected for feature calculation. Because the data base contains many correlated scales, a set of 19 uncorrelated scales derived from the entire AAIndex database [98] can also be selected. Amino acid scales are normalized (zero mean, unit standard deviation) before using them for feature calculation.

Signal average features capture, based on the selected amino acid scale used for generating a property profile, the average property over the entire protein sequence by calculating the average profile value.

Signal peaks area features use the property profiles to capture occurrences of property peaks by calculating the sum of all areas under a protein profile above and below a given threshold. A window and edge parameter define the width and edge weights of a triangular filter with which the profile is convoluted to smooth it before calculating the features [165].

Autocorrelation features employ the property profiles to calculate property correlations between two residues at a given distance over the entire protein sequence. As in PROFEAT, three different types are implemented: normalized Moreau-Broto [92], Moran [93], and Geary [94].

Pseudo-amino acid composition features calculate the amino composition with additional features that include sequence-order information up to a given distance λ. Sequence-order information is incorporated by calculating residue correlation factors between two residues at a given distance over the entire protein sequence, for distances $1, 2, ..., \lambda$. The correlation factors are based on one or multiple user-defined amino acid scales as offered by the PseAAC web server [158]. Both the parallel-correlation type (type 1), as introduced in [89] for predicting protein cellular attributes, and the series-correlation type (type 2), as introduced in [90] for predicting enzyme subfamilies, are offered by SPiCE.

AMINO ACID DISTANCE-BASED FEATURES

These feature categories use amino acid distances for feature calculation, either by using a amino acid distance matrix or by using predefined amino acid clusters.

Property composition, transition, distribution (CTD) features were previously used to predict protein folding classes [96]. Our implementation is based on PROFEAT [159]. The twenty amino acids are subdivided into three groups; A, B, and C, based on a given property. Protein sequences are then mapped to the reduced three-letter alphabet (ABC), which are used to calculate *i*) the property composition, letter counts divided by sequence length, *ii*) property transitions, the number of AB and BA transitions divided by the sequence length - 1 (likewise for AC and BC), and *iii*) the property distribution, relative protein sequence positions of the first occurrence, the 1^{st}, 2^{nd}, and 3^{rd} quantile, and the last occurrence of each property letter. The used properties – hydrophobicity, normalized Van der Waals volume, polarity, polarizibility, charge, secondary structures and solvent accessibility – and corresponding amino acid subdivisions are the same as in PROFEAT.

Quasi-sequence-order descriptors have been used to predict protein subcellular localization [91]. They are comparable to the pseudo amino acid composition, but the Schneider-Wrede amino acid distance matrix [166] is used for calculating correlation factors instead of amino acid scales.

FUNCTIONALITY

SPiCE has four main functionalities, as illustrated in Figure 5.4. First, users can upload a FASTA file with protein sequences for which a range of sequence based features can be calculated (Figure 5.4A). The resulting feature matrix (Figure 5.4B) can then be visually explored using histograms, scatter plots, and heat maps. Classifiers can be trained for a set of user-defined class labels (Figure 5.4C) and the resulting classifier can finally be used to predict class labels of new protein sequences (Figure 5.4D).

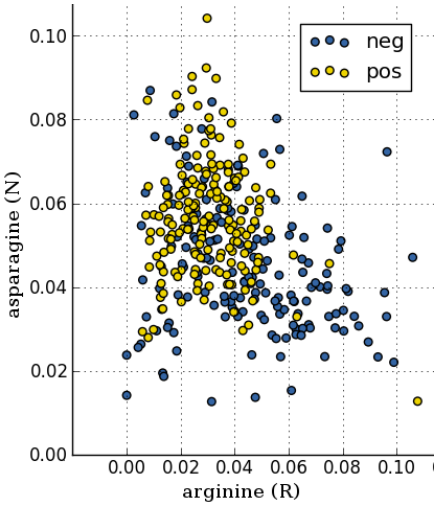


Figure 4.4: Scatter plot showing class separation for the *A. niger* secretion project using the amino acid composition features with the lowest (negative) and highest t-value, arginine and asparagine respectively.

To access these functions, the SPiCE web-based user interface offers four areas: *home*, *projects*, *features*, and *classification*, accessible through the main tabs. The web application can be freely explored without registration. A user account bar – situated directly underneath the main tabs (Figure 4.3) – enables users to login to their account or to create a new account, providing them with a secure personal work space in which their projects will be stored.

Home contains general information and news items. Additional documentation and tutorials can be accessed through the *documentation* link in the header menu at the top of the page (Figure 4.3).

Projects are initiated by uploading a FASTA file with either protein (amino acid) or ORF (nucleotide) sequences. After initiation, one or more labeling files can be uploaded in which each protein is assigned a label, for example its subcellular localization. Users can also upload (predicted) secondary structure and solvent accessibility sequences, which enables the calculation of additional features.

Features can be calculated for all proteins in the project. A list of available sequence-based features is given in Table 4.2. Additionally, users can upload their own calculated features. The resulting feature matrix can be explored using different visualizations. Feature-value distributions and class separation can be explored using histograms (e.g. like in Figure 4.3) and scatter plots. Another way of exploring predictive features is to visually inspect the feature matrix using a hierarchically clustered heat map (Figure 5.4B), in which the protein labels are added as an extra column (not used for clustering).

Classification offers the ability to train classifiers using the proteins in the current project.

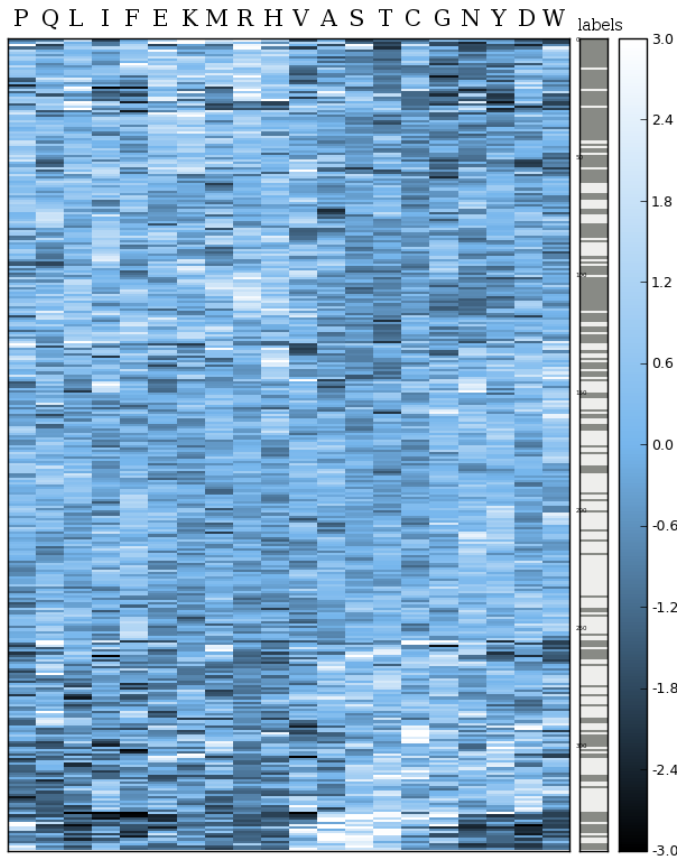


Figure 4.5: Hierarchically clustered feature matrix of the *A. niger* secretion project with the amino acid composition features as columns and the proteins as rows. The corresponding class labels, gray for 'low' and white for 'high', are shown in the column on the right.

Users can select: *i)* the type of classifier to use, *ii)* the classes to train for, *iii)* the features to use for training, and *iv)* the number of cross-validation loops k. A (double) k-fold cross-validation protocol is used to assess classifier performance and to optimize classifier parameters if required. After training, a table with performance measures is reported, together with a receiver operating characteristic (ROC) curve in case of two-class classification. The final classifier is trained on the entire train set using the optimized parameter settings. Trained classifiers can be applied to predict class labels of new proteins by selecting any of the user's projects, in which case class labels will be predicted for each protein in that project.

id	feature category	parameter settings	feature	t-value
aac_1_A1	amino acid composition	number of segments: 1	A, segment 1	16.95
aac_1_V1	amino acid composition	number of segments: 1	V, segment 1	12.60
aac_1_G1	amino acid composition	number of segments: 1	G, segment 1	11.48
aac_1_E1	amino acid composition	number of segments: 1	E, segment 1	5.88
aac_1_D1	amino acid composition	number of segments: 1	D, segment 1	4.11
aac_1_K1	amino acid composition	number of segments: 1	K, segment 1	1.89
aac_1_Y1	amino acid composition	number of segments: 1	Y, segment 1	-0.52
aac_1_W1	amino acid composition	number of segments: 1	W, segment 1	-1.07
aac_1_F1	amino acid composition	number of segments: 1	F, segment 1	-1.12
aac_1_C1	amino acid composition	number of segments: 1	C, segment 1	-1.90
aac_1_I1	amino acid composition	number of segments: 1	I, segment 1	-2.44
aac_1_Q1	amino acid composition	number of segments: 1	Q, segment 1	-2.53
aac_1_L1	amino acid composition	number of segments: 1	L, segment 1	-3.22
aac_1_M1	amino acid composition	number of segments: 1	M, segment 1	-3.22
aac_1_P1	amino acid composition	number of segments: 1	P, segment 1	-3.32
aac_1_T1	amino acid composition	number of segments: 1	T, segment 1	-3.62
aac_1_H1	amino acid composition	number of segments: 1	H, segment 1	-3.68
aac_1_R1	amino acid composition	number of segments: 1	R, segment 1	-4.45
aac_1_N1	amino acid composition	number of segments: 1	N, segment 1	-11.29
aac_1_S1	amino acid composition	number of segments: 1	S, segment 1	-13.54

Figure 4.6: Table with *t*-statistics of the yeast expression-level project. The table shows the *t*-statistics for the amino acid composition features and is ordered by *t*-value. High absolute *t*-values indicate a difference in class means of the two (assumed normal distributed) class distributions.

SOFTWARE FRAMEWORK

The website is developed in Python 2.7.3 [5], using the minimalist python web framework CherryPy 3.2.0 [6]. The back-end uses the Python package *spice* for feature calculation and classification. Within this package, the *featext* module manages feature extraction using a *dataset* module to manage protein sequence data and a *featmat* module to manage the labeled feature matrix. The *classification* module offers a set of classification tasks, which basically is a layer on top of the machine learning software scikit-learn 0.14.1 [160]. Feature extraction and classification tasks are put in a job queue which is handled by a separate compute server.

4.3. RESULTS AND DISCUSSION

To validate the system, we reproduced results of previous work in which a data set was employed to construct classifiers predicting successful high-level production of extracellular proteins in *Aspergillus niger* [167]. The used data set consists of 345 secretory proteins that were over-expressed in *A. niger* and tested for detectable extracellular con-

[5] www.python.org
[6] www.cherrypy.org

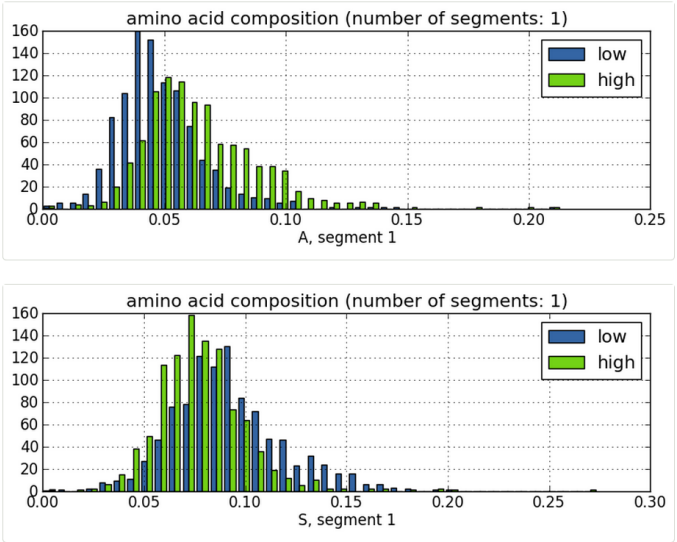


Figure 4.7: Histograms of the yeast protein expression-level project. Histograms are shown for the two amino acid composition features with largest positive and negative t-values (Figure 4.6), alanine and serine respectively, showing different means of the class distributions.

centrations by putting the obtained extracellular medium on a gel after growing the culture in shake flask. A label 'high' was assigned to proteins for which a band on the gel was observed and a label 'low' to the others, resulting in 167 high-level and 178 low-level proteins. This labeled protein set can be loaded as an example project in SPiCE.

The amino acid composition was calculated and used for the construction of a linear support vector machine (10-fold double-loop cross-validation), providing results that are in agreement with the results described earlier [167]. Similar to the observations in that work, the t-statistics reveal strong predictive capacity for the tyrosine, asparagine, arginine, and lysine features (Figure S1), which can also be observed in the histograms (Figure S2). The scatter plot in Figure 4.4 shows the obtained class separation by using the two features with the lowest (negative) and highest t-value respectively. For the hierarchically clustered feature matrix in Figure 4.5, clustering of proteins (rows) with the same label indicate that these features are useful for classification. Classifier construction resulted in a cross-validation performance of 0.837 area under the ROC curve (Figure S4), again similar to results obtained in [167].

Additionally, we used a yeast protein expression data set to illustrate the ease with which one can explore differences between user-defined protein classes. For this data set, yeast proteins were split into low-level and high-level expressed based on data found in [168], in which *Saccharomyces cerevisiae* open reading frames were tagged with a high-affinity epitopes and expressed from their natural chromosomal location after which protein abundances were measured during log-phase growth by immunodetection of the tag. As a pre-processing step, to avoid a bias for sets with highly similar proteins, BLASTCLUST

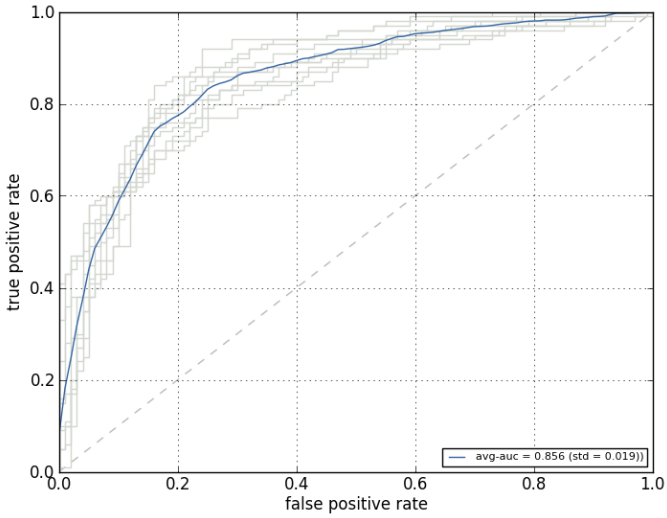


Figure 4.8: Receiver operator characteristic (ROC) curve showing performance of a classifier trained for the yeast expression-level project. The ROC curve shows the performance of a linear support vector machine classifier that was trained using the codon composition as features. Results for the 10 cross-validations are shown in gray, the average performance is shown in blue.

[123] was used to reduce sequence redundancy. After that the list of proteins was ordered by expression level. The top and bottom 1000 proteins were labeled 'high' and 'low' respectively. This data is also available as an example project.

Using the t-statistics table in Figure 4.6, quick exploration of the amino acid composition reveals a preference for alanine, valine, and glycine in the high-expression class, whereas low-expression proteins contain relatively many asparagines and serines. The alanine and serine histograms in Figure 4.7, the features with minimal and maximal t-value respectively, indeed show shifted means of the class distributions. A classification performance, again using a linear support vector machine and 10-fold cross-validation, of 0.794 area under the ROC-curve (Figure S8) showed good predictive capability of the amino acid composition. The predictive capability using the codon composition proved even better, resulting in a performance of 0.856 area under the ROC-curve (Figure 4.8).

For further exploration of the system, two additional example projects can be initiated. One entails protein subcellular localization in human, a data set of 2580 proteins categorized into 14 different subcellular locations as taken from [169]. The other is a solubility data set obtained from [74], consisting of 17.408 yeast proteins that are split into two equal sized classes: *soluble* and *insoluble*.

4.4. CONCLUSION

SPiCE provides easy access to visualization and classification methods for a set of labeled protein sequences. After uploading a FASTA file with protein sequences and a label file with protein labels, the website can be used to calculate sequence-based features, to visualize the resulting feature matrix, and to train and test classifiers for predicting class labels, enabling quick exploration of sets of labeled proteins. The back-end software is made available for expert users to perform customized and computationally demanding tasks on a local computer.

4.5. SUPPORTING INFORMATION

Project name	SPiCE
URL	`http://helix.ewi.tudelft.nl/spice`
Source code python package	`https://github.com/basvandenberg/spice`
Source code web site	`https://github.com/basvandenberg/spiceweb`
Web browsers	Chrome, Firefox, Opera, Safari
Operating system	Platform independent
Programming language	Python 2.7
License	GNU GPL v3

Additional file 1: Showing the use of SPiCE by means of two example projects.[7]

[7] `http://www.biomedcentral.com/content/supplementary/1471-2105-15-93-s1.pdf`

5

PROTEIN REDESIGN BY LEARNING FROM DATA

Bastiaan A van den Berg, Marcel JT Reinders,
Jan-Metske van der Laan, Johannes A Roubos, Dick de Ridder

Published in Protein Design, Engineering & Selection (PEDS), Volume 27, Issue 9: pages 281 – 288, 2014, doi:10.1093/protein/gzu031

ABSTRACT

Protein redesign methods aim to improve a desired property by carefully selecting mutations in relevant regions guided by protein structure. However, often protein structural requirements underlying biological characteristics are not well understood. Here we introduce a methodology that learns relevant mutations from a set of proteins that have the desired property and demonstrate it by successfully improving production levels of two enzymes by *Aspergillus niger*, a relevant host organism for industrial enzyme production. We validated our method on two enzymes, an esterase and an inulinase, creating four redesigns with 5-45 mutations. Up to 10-fold increase in production was obtained with preserved enzyme activity for small numbers of mutations, whereas production levels and activities dropped for too aggressive redesigns. Our results demonstrate the feasibility of protein redesign-by-learning. Such an approach has great potential for improving production levels of many industrial enzymes and could potentially be employed for other design goals.

5

5.1. INTRODUCTION

Proteins are engineered to enhance structural characteristics or to confer new interactions or catalytic functions, with industrial applications in the manufacturing of pharmaceuticals, the processing of food, the composition of detergents, the production of bioplastics and biofuels, and in the bioremediation of waste streams [170]. For the production of industrial enzymes, redesign becomes more and more adopted as an essential tool for attaining economically relevant rates and yields in setting up production processes of high-value proteins [8, 9]. Next to optimization of transcription and translation, e.g. by applying strong and inducible promoters and codon optimization, proteins are redesigned to optimize signal sequences, add N- and C-terminal (solubility) tags, create fusion proteins or co-express with foldases [36]. More recently, there is a growing interest in applying protein redesign for changing properties of the enzyme itself, e.g. to enhance catalytic activity, (thermo)stability, or solubility. In this work, we used a novel protein redesign-by-learning strategy to enhance enzyme production levels.

Over the last decades, protein engineering has moved from the use of directed evolution where large libraries are screened, to rational protein (re)design using computational methods [38, 171, 172]. Proteins have been computationally redesigned to improve folding and stability [51–53, 173], to change binding affinity and specificity [47–49, 174] and even to construct novel enzymatic activities [41, 44, 45]. Redesign methods often start off with a desired backbone and use an energy function to find a corresponding sequence with optimal free energy. This search space is very large and hard to explore using existing search methods [39]. Rational design approaches therefore usually exploit known relationships between a protein's structure and physicochemical properties to target a limited number of residues. For example, Gribenko *et al.* specifically targeted charge-charge interactions on a protein's surface based on the knowledge that optimization of these interactions could enhance its stability [51]. Tian *et al.* targeted glycine-to-proline substitutions in flexible regions, as decreasing conformational entropy is thought to lead

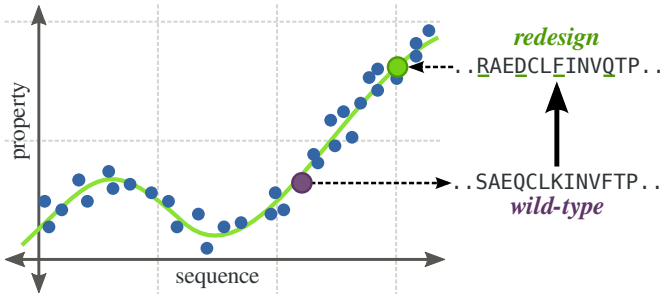


Figure 5.1: Protein redesign-by-learning: a sequence-based predictor is trained on a large set of example proteins with known property and then used as a criterion in a search algorithm that optimizes the desired property. Additional objectives and constraints are required to account for protein structure.

to increased stability [53]. In these cases, targeting a limited number of residues for redesign has the advantage that computationally expensive modeling techniques can be employed to guide the redesign.

In previous work [175], we observed a relation between a global sequence property, the amino acid composition, and high-level production of extracellular proteins by *Aspergillus niger*, a relevant host for industrial enzyme production [17]. Mechanisms by which a protein's characteristic amino acid composition could affect production and secretion processes are unknown, so we cannot target specific structural regions. As a result the number of possible mutations to consider is enormous, rendering the application of computationally expensive optimization methods infeasible. However, given a sufficient number of examples, it is possible to learn global sequence-activity models [176] that can then be used to guide protein redesign. Such approaches were successfully applied for improving thermostability [177] and for removing T-cell epitopes [178]. Datasets sufficiently large to successfully train such models are increasingly available and can be successfully exploited [179].

Therefore, as an alternative to the use of energy functions, we propose a methodology that learns how to redesign proteins from examples (Figure 5.1). First, a predictor is learned from a large set of sequence-based measurements of example proteins, that is indicative for a protein property of interest. Next, this predictor is used as a criterion in an optimization scheme, which iteratively modifies sequences to achieve a certain desired or maximum value for the protein property of interest. We demonstrate this approach by successfully redesigning enzymes for improved production levels by *A. niger*.

5.2. MATERIALS AND METHODS

GENERAL APPROACH

In earlier work, we studied sequence characteristics predictive for high-level production of extracellular proteins by *A. niger* [175]. Briefly, we exploited a large data set of fungal genes, over-expressed in *A. niger*and tested for high versus low (or no) extracellular protein production. Proteins were divided into classes S_{high} and S_{low}, respectively. Subsequently, we used machine learning algorithms to identify DNA and protein sequence features discriminating between these two classes. Extensive analyses indicated the protein amino acid composition to be most predictive: the aromatic amino acid and asparagine fractions are positively correlated with high-level production, the lysine fraction with low production (Figure 5.3, for more details, see [175]). For the purpose of protein redesign, we trained a production-level classifier exploiting amino acid compositions of 345 tested *A. niger*genes (170 in S_{low} and 175 in S_{high}).

The trained classifier (Figure 5.2a) is capable of predicting high-level production given an input protein sequence: the higher the classifier outcome, the higher the predicted probability of high-level production. This outcome is then used as a criterion for optimization. In essence, the classifier is "inverted", allowing us to predict what sequence is most likely to result in a certain desired (maximum) production level. This is the core of our protein redesign-by-learning methodology (Figure 5.2b).

As the classifier is based on sequence data only, optimization of this objective alone is likely to lead to proteins that lose structure, stability or function. Consequently, we use additional constraints in the optimization that prevent the redesigned sequence deviating from these aspects. The final protein redesign strategy thus optimizes a combination of multiple objectives. Next to optimizing the classifier output (Figure 5.4a), a multiple sequence alignment with highly similar proteins is used to avoid mutations at conserved positions (Figure 5.4b). Furthermore, mutations in the protein core and in the vicinity of active sites are not allowed (Figure 5.4e), based on the assumption that these have high risk of affecting function. Finally, the difference between the amino acid composition of the redesigned protein and the average S_{high} amino acid composition is minimized (Figure 5.4c). This avoids repeatedly selecting the same amino acid substitution, which could result in a skewed amino acid composition.

PRODUCTION-LEVEL CLASSIFIER

We set up a support vector machine classifier based on experience gained in previous work [175] and trained it to discriminate between genes for which over-expression resulted in low and high production levels, respectively. The DSM industrial strain used as a host for over-expression of enzymes in *A. niger*is a protease and amylase reduced strain derived from DS03043. A standard expression unit was used for targeted integration of desired genes between the host-own glucoamylase promoter and terminator elements in a proprietary *Escherichia coli* vector [180]. After growing the cultures in shake flask fermentations, extracellular medium was filtered and protein concentrations were

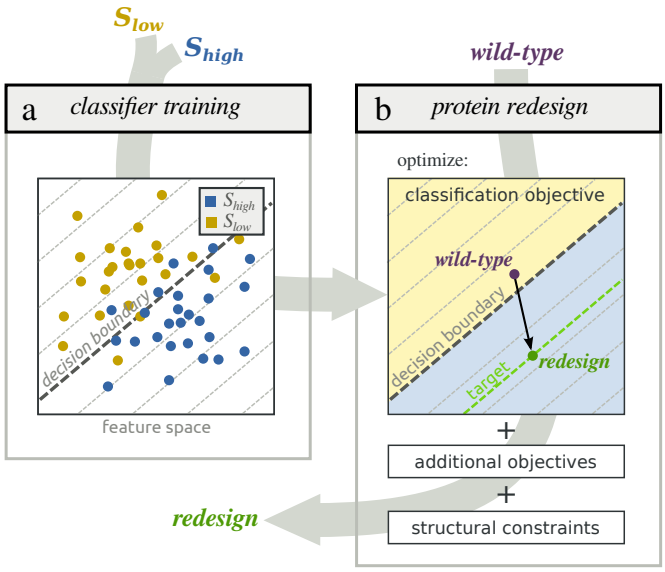


Figure 5.2: Redesigning proteins for improved production levels. **(a)** A classifier is trained to discriminate between two protein classes S_{low} and S_{high}, in our case a set of proteins that has low production levels and a set that has high production levels. **(b)** When redesigning a protein, the trained classifier is used as objective in order to find mutations that moves the classification outcome of the wild-type protein towards the target classification outcome. Additional objectives prevent mutating conserved residues and too much deviation from the average S_{high} amino acid composition. Furthermore, structural constraints prevent mutating buried residues and residues near the active site.

measured using SDS-PAGE electrophoresis. SDS-PAGE was used to evaluate successful high-level production, the label *high-level* was assigned to genes resulting in a visible band, and the label *low-level* was assigned to the remainder. For more details, see [175].

Since we aimed for redesigning a host-own protein, a classifier was trained using a set of *A. niger*genes tested for homologous overexpression. We only selected proteins that were expected to be secreted, based on predicted presence of signal peptides and no ER-retention signals nor transmembrane helices. Signal peptide presence and signal peptide cleavage sites were predicted with SignalP 3.0 [181], transmembrane helices were predicted with TMHMM 2.0 [121]. To avoid a bias for sets with similar proteins, we used BLASTCLUST [123] to remove proteins that share more than 40% sequence identity over a length of 90% with any of the other proteins. This resulted in a set S of 345 proteins, split into 170 low-level proteins (S_{low}) and 175 high-level proteins (S_{high}). Both redesigned enzymes and their high-level paralogs were not in this set.

Using the amino acid sequences (excluding the signal peptide), optimal classification performance was obtained using the amino acid composition as features [175]. This resulted in an area under the receiver-operator characteristic curve of 0.83 (10-fold cross-

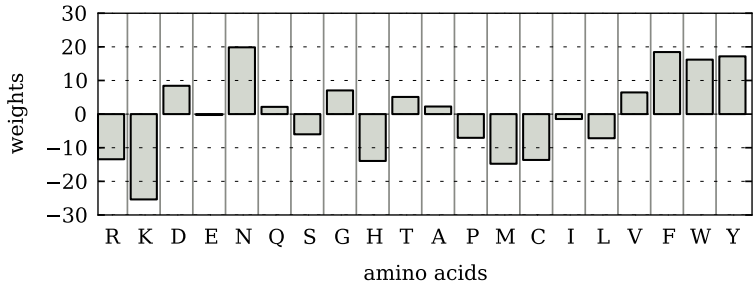


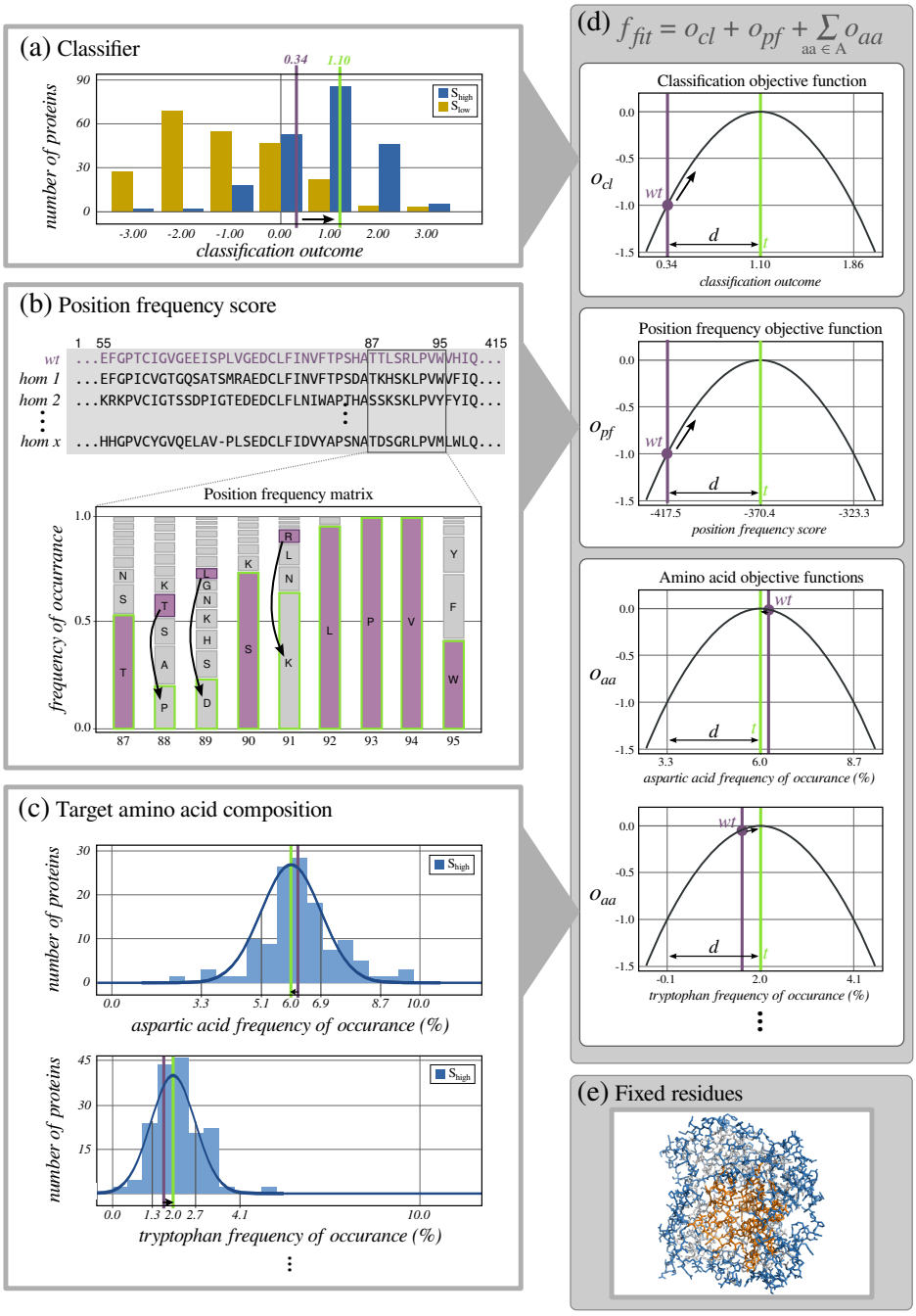
Figure 5.3: Amino acid weights retrieved from a classifier trained for optimal separation of S_{high} and S_{low} based on a protein's amino acid composition. Positive and negative weights indicate importance for high- and low-level production respectively.

validation), indicating good performance of the classifier. Analysis of the classifier provided the feature contribution weights **w** in Figure 5.3, in which negative and positive weights denote negative and positive contributions to predicted high-level production, respectively. The classification outcome for a protein sequence s is defined as:

$$cl(s) = \mathbf{w}^T \cdot \mathbf{c_s} + b, \tag{5.1}$$

where $\mathbf{c_s}$ is the amino acid composition of s, i.e. a vector with the relative frequency of occurrences for the 20 amino acids, and b is a constant.

Figure 5.4: Design method overview showing the three objectives that are combined into a fitness function. The esterase (An08g11860) is used as an example, with original values wt in purple and the target values t in green. **(a)** Histogram of classification outcomes of proteins in S_{low} and S_{high}. This objective will favour mutations that move the original classification outcome (0.34) towards the average classification score of S_{high}. **(b)** Shows a multiple sequence alignment of An08g11860 with a set of similar proteins and a derived position frequency matrix. This objective promotes mutations from infrequently occurring amino acids into frequently occurring ones. **(c)** Histograms and fitted normal distributions of the compositions of S_{high} for two amino acids, aspartic acid and tryptophan. This objective promotes mutations that move the esterase amino acid composition towards the average S_{high} composition. **(d)** The overall fitness function. Each of the objectives is evaluated using a quadratic function with the target score positioned at the top of the parabola. Maximization of the sum of these function results in a combined optimization of the three objectives. **(e)** Predicted structure model with buried residues in white and residues near active sites in orange. Only blue residues are allowed to be mutated.


(a) Classifier
0.34
1.10
number of proteins
classification outcome
(b) Position frequency score
wt ...EFGPTCIGVGEEISPLVGEDCLFINVFTPSHATTLSRLPVWVHIQ...
hom 1 ...EFGPICVGTGQSATSMRAEDCLFINVFTPSDATKHSKLPVWVFIQ...
hom 2 ...KRKPVCIGTSSDPIGTEDEDCLFLNIWAPJHASSKSKLPVYFYIQ...
hom x ...HHGPVCYGVQELAV-PLSEDCLFIDVYAPSNATDSGRLPVMLWLQ...
Position frequency matrix
frequency of occurrance
(c) Target amino acid composition
number of proteins
aspartic acid frequency of occurance (%)
tryptophan frequency of occurance (%)
(d) $f_{fit} = o_{cl} + o_{pf} + \sum_{aa \in A} o_{aa}$
Classification objective function
classification outcome
Position frequency objective function
position frequency score
Amino acid objective functions
aspartic acid frequency of occurance (%)
tryptophan frequency of occurance (%)
(e) Fixed residues

DESIGN METHOD

An overview of the design method is shown in Figure 5.4 in which wt denotes the wild-type protein (excluding signal peptide) to be redesigned. At the core is an algorithm to optimize multiple objectives: a classification objective (o_{cl}), a position frequency objective (o_{pf}), and an amino acid composition objective (o_{aa}) per amino acid (aa) in the amino acid alphabet (A). These objectives are combined into a single fitness function for a sequence (s) using:

$$f_{fit}(s) = o_{cl}(s) + o_{pf}(s) + \sum_{aa \in A} o_{aa}(s). \tag{5.2}$$

Each objective is evaluated with quadratic functions of the form:

$$o(s) = -\frac{1}{d^2} \times (h(s) - t)^2, \tag{5.3}$$

where $h(s)$ is a specific score for sequence s, t is a preset target score, for which the function evaluates to 0 (top of the parabola, maximum fitness) and d is a scale parameter, the distance to the target score at which the objective evaluates to -1 (Figure 5.4d). The latter parameter is chosen for each objective individually and controls its relative contribution to the overall fitness function. In general, d is set such that the wild-type design evaluates to an objective function of -1.

Classification objective $o_{cl}(s)$ uses the outcome of the classification function $cl(s)$ as a score. We do not expect a continuing production-level increase when maximizing the classifier outcome without limits. Therefore, we set the target score t to 1.10, which is a little (0.2) above the average classification outcome of the proteins in S_{high}. The distance d is set to the difference between the target and the wild-type classification outcome (Figure 5.4a).

Position frequency objective $o_{pf}(s)$ optimizes for mutations to amino acids that are often observed at the same position in highly similar proteins from other organisms. Protein similarity was assessed with BLAST using wt as query against the NCBI NR database, using default parameter settings. Only proteins with sequence identity $i > 0.35$ and coverage $c > 0.9$ were selected, assuming that their structures are comparable to wt. To avoid a bias due to multiple occurrences of the same sequence, redundant sequences ($i > 0.9, c > 0.9$) were filtered out using BLASTCLUST [123]. This resulted in redundancy-reduced sets of similar protein sequences H_{wt} (Table TODO-S-tab:msa_ids). A multiple sequence alignment was constructed for $wt \cup H_{wt}$ using Clustal Ω [182], in which all columns with a gap in wt were removed. Subsequently, a $20 \times |wt|$ position frequency matrix was constructed, which is used for calculating position frequency scores $pf(s)$ by taking the sum of logs of the residue frequencies of s (Figure 5.4b). This score is used to calculate the position frequency objective $o_{pf}(s)$ using (5.3). The target score t is set to the maximum possible pf for the given number of mutations m, which means that t differs per design. This is done to render the weight of this objective similar for different

numbers of mutations. The distance d is set to the distance between the target and the wild-type position frequency score.

Amino acid composition objectives $o_{aa}(s), \forall aa \in A$ optimizes for an amino acid composition close to the average composition of proteins in S_{high}. The goal is to prevent repeated selection of certain mutations, which could result in a skewed amino acid composition. For a given amino acid, this objective takes the relative frequency of occurrence of the amino acid as score. Target scores t are set to the mean frequency of occurrence of the amino acids in the S_{high} proteins; d is set to five times the standard deviation (Figure 5.4c). This setting was based on test designs, aiming for the objective to start to have an effect for designs with relatively many mutations ($m > 10$, Supplementary Figure S3), for which the risk of a skewed amino acid composition becomes more relevant.

The fitness function f_{fit} was optimized using a genetic algorithm (see Supplementary Methods for details). The population size and the number of generations were set to 1000 for all designs. The best result of twenty runs (redesign with highest fitness score) was selected for the redesigns with five mutations; the best result of fifty runs was selected for the designs with more than five mutations.

STRUCTURE MODELS AND STRUCTURAL ALIGNMENT

Structure models were predicted for proteins excluding the predicted signal peptide using the I-TASSER webserver [183]. To improve confidence in model accuracy, proteins will only be considered for design if a structure with sequence identity $i > 0.3$ and coverage $c > 0.9$ was present in the Protein Data Base (PDB). Also, predicted structure models will only be considered for redesign if their by I-TASSER predicted TM-score exceed 0.5, indicating a correct topology. For comparing redesigns to their paralogs, TMAlign (version 2012-01-24) was employed for structural alignment [184], obtaining the fraction of aligned residues that are identical as a similarity measure.

FIXED RESIDUES

The I-TASSER output also provides ligand-binding residues as predicted using COFACTOR [185]. Only binding residues predicted with confidence score $C > 0.5$ and binding site score $BS > 1.1$, indicating a good local match with a template binding site, were accepted. Predicted ligand-binding residues were fixed, i.e. no mutations were allowed at these positions. PyMol [186] was used to determine which residues reside within 8Å of any of the ligand-binding residues, by selecting all residues that have an atom within 8Å distance from an atom in a ligand-binding residue; these residues were fixed as well. The fixed (near) ligand-binding residues in the esterase (An08g11860) are shown in orange in Figure 5.4e. Accessible surface areas (ASA) of all residues in a predicted structure were calculated. Relative ASAs were obtained by scaling for the extended states of Ala-X-Ala for every residue X. Residues with relative ASA smaller than 5% were considered buried and therefore fixed. The fixed buried residues in the esterase are shown in grey in Figure 5.4e.

EXPERIMENTAL SETUP

Protein sequences were codon optimized using the method described in [187]. The same as with setting up the learning data set, a protease- and amylase-reduced *A. niger*strain was used as host for protein over-expression and a standard expression unit was used for targeted integration of desired genes between the host-own glucoamylase promoter and terminator elements in a proprietary *Escherichia coli* vector [180]. In this case, resulting enzyme concentrations were measured quantitatively using qSDS-PAGE (see Supplementary Methods for details). For determining inulinase activity, a standard endo-fructanase assay using azo-fructan as substrate (Megazyme assay kit S-AZFRXOI 11/99) was used. Details about the esterase activity measurements are in the Supplementary Methods.

5.3. RESULTS

TWO REDESIGNED ENZYMES

To demonstrate our method we redesigned two enzymes (see Supplementary Table S1): an esterase (An08g11860) and an inulinase (An11g03200). The redesigns were tested in the lab for improved production levels. Both enzymes are expected to be secreted into the extracellular medium: they have a predicted signal peptide, lack predicted transmembrane helices, and are predicted to be extracellular. However, previous work did not yield measurable extracellular concentrations for these enzymes after over-expression [175], i.e. both enzymes are in S_{low}. These enzymes were selected because *i)* their classification outcomes (0.42 and 0.51, respectively) are lower than the target classification outcome (1.10), which leaves room for optimization, and *ii)* accurate predicted structure models are available for both, which enabled us to fix buried and active residues. Additionally, these two enzymes were of particular interest because of available paralogs in S_{high}, which enabled *in silico* validation by comparing redesigned low-level enzymes to their high-level paralogs. To avoid a bias, these paralogs were *not* used during the redesign process, which means that they were removed from S_{high} and from the multiple sequence alignment with similar proteins.

In our redesign, we can decide on the number of mutations m allowed, enabling variation between conservative redesigns with just a few mutations, and liberal ones containing many. To study the influence of the number of mutations, fitness scores – a score that indicate how well a redesign fits our desires – were obtained for redesigns in the range $m = 1, 2, \ldots, 88$ and plotted against m (Supplementary Figure S2). The point where the fitness score saturates was taken as the maximum number of mutations: 45 for the esterase, 30 for the inulinase. Using uniform sampling, esterase redesigns with $m = 15, 30, 45$ mutations were created and inulinase redesigns with $m = 10, 20, 30$ mutations. To test the effect of only a few mutations, an additional redesign with $m = 5$ mutations was created for both enzymes. Sensitivity of the redesigns to the choice of the parameters d was assessed for the inulinase by varying these by ±20%. The resulting redesigns show only limited variation in the selected mutations (Supplementary Fig-

ure S9, S10, and S11). For wet lab testing, all redesigns were translated into DNA, codon-optimized [187] and expressed in *A. niger*.

INCREASED SIMILARITY COMPARED TO HIGH-LEVEL PARALOGS

To verify whether our redesigns resembled known high-level produced paralogs, we calculated sequence identity, i.e. the fraction of identical residues in a structural alignment. The esterase was compared to hydrolase An16g08870, and the inulinase was compared to exo-inulinase An12g08280. Results are given in Table TODO-S-tab:redesign_ident. In general, redesigning increased the similarity to known high-level produced proteins, but not by much. The initial identity between the esterase and An16g08870 was 40.0%. As about half of the mutations in the esterase resulted in the amino acids present in An16g08870, sequence identity increased up to 44.5% for the redesign with 45 mutations. On the other hand, up to five identical residues were lost for the redesigns with more than 5 mutations (Supplementary Figure S4b). The initial inulinase and An12g08280 were 32.6% identical, increasing up to 34.0% for the redesigns. For the conservative redesigns ($m = 5, 10$), half of the residues changed into amino acids identical to those in An12g08280. Fewer identical residues were gained for the more liberal redesigns ($m = 20, 30$). In contrast to the esterase, no identical residues were lost for any of the redesigns (Supplementary Figure S4b).

10-FOLD PRODUCTION INCREASE WITH RETAINED ACTIVITY

Redesigns were tested in triplo for improved extracellular concentrations after overexpression in *A. niger*. A wild-type and codon-optimized version were tested as reference. Resulting extracellular concentrations are shown in Figure 5.5; original qSDS-PAGE results in Supplementary Figures S7 and S8. Concentrations obtained for the wild-types were lower than the detection limit in previous work [175], confirming that both enzymes are in S_{low}. The codon-optimized version resulted in slightly higher concentrations of up to 0.1 mg/ml, i.e. both enzymes were secreted. For the esterase, a redesign with 5 mutations resulted in a 10× concentration increase, whereas redesigns with more than 5 mutations gave no measurable concentrations. For the inulinase, the redesign with 5 mutations gave a 5× concentration increase, and redesigns with 10 and 20 mutations a 10× concentration increase. Only the redesign with 30 mutations failed.

Our method aims for improved production levels, and although constraining mutations to residues away from the active site lowers the risk of affecting enzymatic activity, retained activity is of course not guaranteed. Therefore, redesigns were also tested for retained enzymatic activity. Resulting activities for each inulinase sample are plotted against the corresponding protein concentration for each sample in Figure 5.6. High correlation ($r = 0.96$) between concentration and activity can be observed for the redesigns with 5 and 10 mutations, confirming retained activity. Lower activities with respect to the observed protein concentration for the redesign with 20 mutations indicate affected activity for this redesign. Based on the closest biochemically characterized similar protein, the esterase was expected to accept tributyrin as substrate [188], but no activities

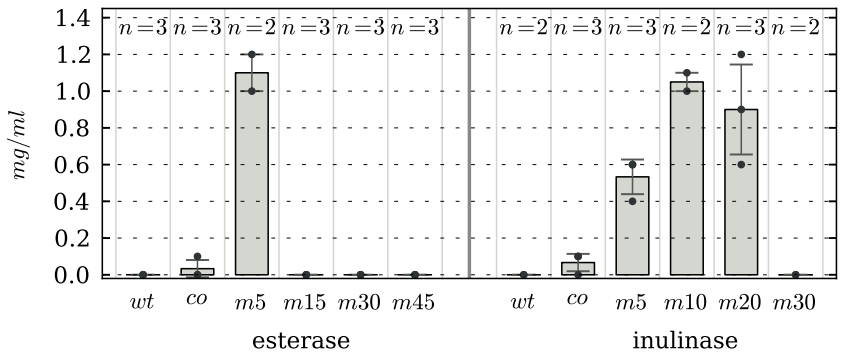


Figure 5.5: Extracellular concentrations for wild-type enzymes and their redesigned versions after over-expression in *A. niger*: left, the esterase (An08g11860); right: the inulinase (An11g03200). In both cases, the first two bars show concentrations measured for the wild-type (wt) and codon-optimized wild-type version (co). The next four bars indicate concentrations obtained for codon-optimized redesigns for increasing numbers of mutations. Most experiments were done in triplicate, some in duplicate (due to failed experiments), as indicated by n. Dots indicate results of each experiment separately, the bars indicate average concentrations with standard deviations.

were observed for any of the samples using a lipase plate assay, including the wild-type and codon optimized version. Additionally, spectrophotometric determination of lipase activity using *p*-nitrophenyl palmitate as substrate and the determination of the esterase activity using pNP-butyrate as substrate did not yield any activity. Therefore it was not possible to test for retained activity for the esterase redesigns.

5.4. DISCUSSION

All redesigned enzymes more closely resembled high-level produced paralogs in terms of sequence similarity, even though these paralogs were not used by the redesign method in any way, i.e. there is no bias in the method to modify the sequence in that direction. This suggests that protein redesign-by-learning is able to generalize well, and that sequence characteristics are identified which correlate with naturally occurring high-level produced proteins. Experimental results confirmed that our redesign method can indeed be successfully applied to improve production levels. While too liberal redesigns failed, it was possible to obtain concentration increases of up to 10× by only 5-20 mutations.

Understanding the relation between improved production and the underlying biology is difficult, since protein production and secretion involves many steps, all of which may influence the obtained extracellular concentrations. It cannot be excluded that the amino acid substitutions affect transcription and translation and thereby influence protein production. However, the large effect with only few mutations (10× in-

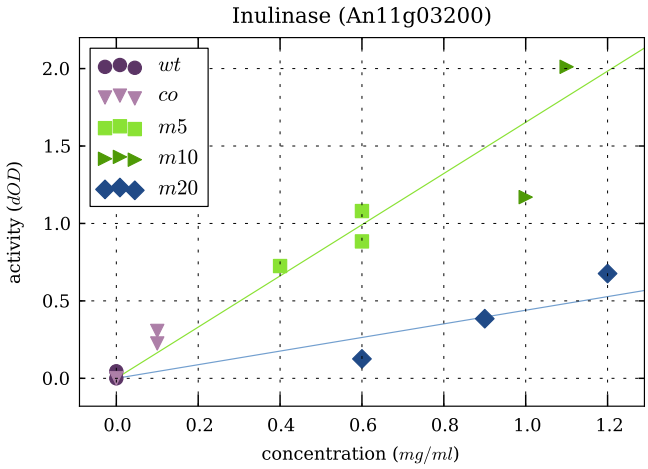


Figure 5.6: Inulase activities plotted against corresponding protein concentrations. This is shown for the wild-type (wt), codon optimized version (co) and the redesigns with 5, 10, and 20 mutations ($m5$, $m10$ and $m20$). Experiments were done in duplo or triplo, each dot represents a single experiment. The green line is a linear fit through the origin for the measurements of all samples except the ones of the 20 mutations redesign, showing that absolute activities linearly increase with concentrations. The blue line is a linear fit through the origin for the samples of the 20 mutations redesign, in this case showing a clear drop in activity.

crease given 5 mutations) indicates that the main effect is not due to changing transcription or translation rates, but most likely due to post-translational effects in the secretion pathway. Interestingly, we observed an increase in the number of potential N-glycosylation sites in our redesigns, due to the introduction of asparagines (Table TODO-S-tab:redesign_numbers). With only a single exception, all mutations to asparagine in the inulinase redesigns introduce a new N-glycosylation pattern. N-glycosylation is a post-translational process that attaches glycans to asparagine side-chains. Although details are unclear, these glycans are thought to play a role in protein folding and quality control [148]. Introduction and modification of N-glycosylation sites has resulted in improved secretion and production before [140, 150, 151]. However, this is not the whole story, as the inulinase redesign with 5 mutations did not introduce any new N-glycosylation pattern and still resulted in a 5× concentration increase, indicating effects of additional mechanisms.

For the esterase, only 5 mutations sufficed for a 10× concentration increase. Redesigns with more than 5 mutations did not result in measurable extracellular concentrations, indicating that some mutations may have adversely affected protein folding or transport. Most likely, proper folding is hampered which usually leads to intracellular clearance by proteolysis. Pinpointing the responsible mutations is difficult because we independently generate redesigns for different numbers of mutations, i.e. a redesign with 5

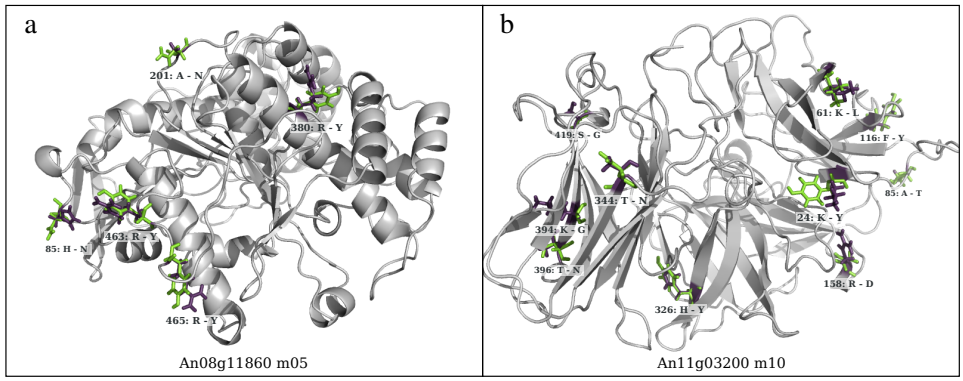


Figure 5.7: Structure models of the most successful redesigns, both providing 10× protein concentrations. Side chains of the mutated residues are shown; the wild-type in purple and the mutant in green. Corresponding annotations give the sequence position and wild-type mutant amino acid pair. **(a)** Esterase redesign with five mutations. **(b)** Inulinase redesign with 10 mutations.

mutations is not used as starting point for a redesign with 10 mutations. We chose for this strategy because in our method, the best 5 mutations are not necessarily all part of the best 10. However, 6 positions (71, 119, 162, 284, 322, and 473) were untouched in the redesign with 5 mutations but mutated in each of the other redesigns (Supplementary Figure S6). This may indicate a relation to the loss of production. Moreover, in all redesigns with more than 5 mutations one or more prolines were substituted, increasing conformational entropy. Since no prolines were substituted in any of the successful inulinase redesigns, these mutations are potential suspects as well.

The inulinase redesigns were successful as well, resulting in improved concentrations for the redesigns with up to 20 mutations. The unsuccessful redesign with 30 mutations carries 14 mutated positions that are unique with respect to any other redesign (Figure 5.8), indicating they may be related to the affected production. Given its proximity to the (predicted) signal peptide cleavage site, the mutation at position 1 in particular is a potential suspect, possibly affecting translocation to the endoplasmic reticulum.

In conclusion, we were able to learn how to redesign proteins from a set of example proteins that showed the desired behavior. Using this approach we were able to successfully increase extracellular enzyme concentrations up to 10× by altering the amino acid composition of a protein. The proposed methodology has great potential for improving production rates of other enzymes, possibly also in other organisms after constructing a organism-specific classifier. While we applied the approach here to improve enzyme production, the methodology itself is generic: given a set of example proteins and measured properties, sequences can be redesigned to achieve certain redesign goals.

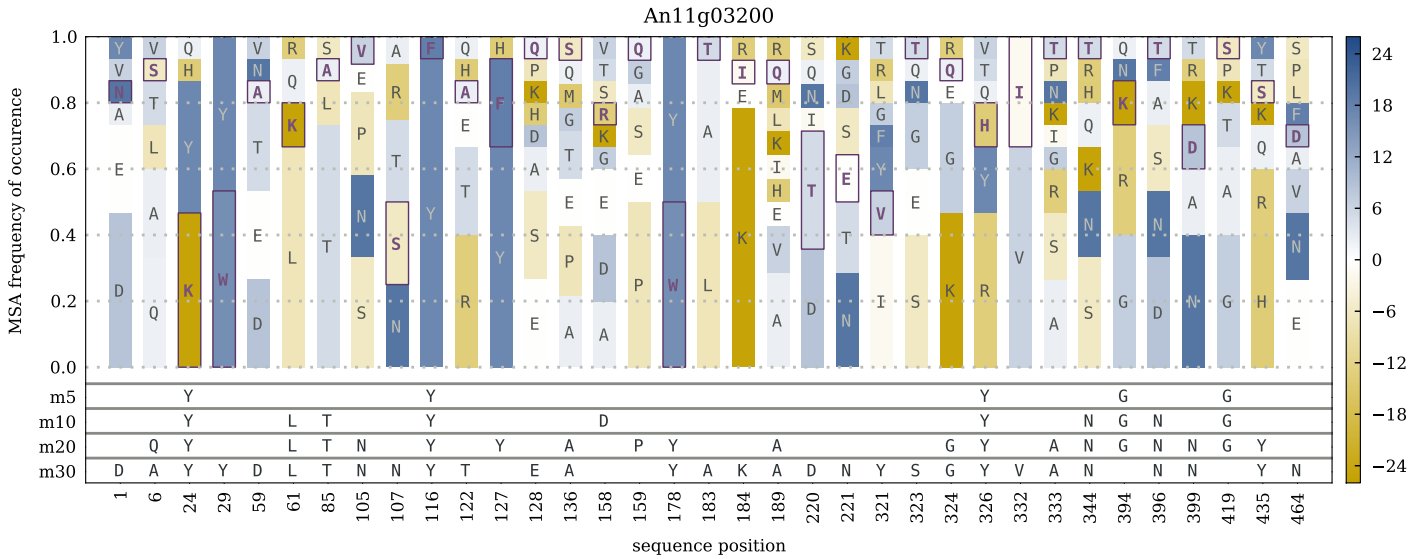


Figure 5.8: Inulinase redesigns. The four rows at the bottom show changed residues and their positions in redesigns with 5, 10, 20 and 30 mutations. The stacked bar plot reflects relative frequencies of amino acid occurrences in the multiple sequence alignment with similar proteins. Bars are sorted, indicating least (top) to most (bottom) frequent occurrence. Purple-bordered bars indicate wild-type amino acids, bar colors reflect amino acid weights in Figure 5.3, corresponding to their importance for high-level production.

5.5. SUPPORTING INFORMATION

The supporting information can be accessed online [1].

[1] http://peds.oxfordjournals.org/content/27/9/281/suppl/DC1

6

INSIGHT INTO NEUTRAL AND DISEASE-ASSOCIATED HUMAN GENETIC VARIANTS THROUGH INTERPRETABLE PREDICTORS

Bastiaan A van den Berg, Marcel JT Reinders, Dick de Ridder, Tjaart AP de Beer

Accepted for publication in PLOS ONE.

ABSTRACT

A variety of methods that predict human nonsynonymous single nucleotide polymorphisms (SNPs) to be neutral or disease-associated have been developed over the last decade. These methods are used for pinpointing disease-associated variants in the many variants obtained with next-generation sequencing technologies. The high performances of current sequence-based predictors indicate that sequence data contains valuable information about a variant being neutral or disease-associated. However, most predictors do not readily disclose this information, and so it remains unclear what sequence properties are most important. Here, we show how we can obtain insight into sequence characteristics of variants and their surroundings by interpreting predictors. We used an extensive range of features derived from the variant itself, its surrounding sequence, sequence conservation, and sequence annotation, and employed linear support vector machine classifiers to enable extracting feature importance from trained predictors. Our approach is useful for providing additional information about what features are most important for the predictions made. Furthermore, for large sets of known variants, it can provide insight into the mechanisms responsible for variants being disease-associated.

6.1. INTRODUCTION

Over the last decade, many predictors have been developed to categorize human nonsynonymous SNPs as disease-associated or neutral [189–204]. Such predictors can be used for identifying the relatively few disease-associated variants in human variation data, a type of data that is rapidly increasing due to the advances in whole genome sequencing techniques [205]. These methods typically employ large sets of known neutral and disease-associated variants to learn how to separate both classes based on variant characteristics, i.e. features. As might be expected, the degree of sequence conservation is highly predictive for disease association of genetic variants. Therefore, all available prediction methods heavily rely on conservation-based features. In fact, several methods, among which the often used method SIFT, predict class labels by thresholding a single conservation-based feature.

A comparative study, however, showed improved prediction results for methods that incorporate additional sequence-derived features [206]. It found two methods, MutPred [196] and SNPs&GO [195] to be most reliable. MutPred builds upon the SIFT score by incorporating gain and loss of structural and functional properties; SNPs&GO calculates several conservation-based features and additionally incorporates features that capture the amino acid substitution, its surrounding sequence, and features based on the functional annotation of the protein in which the substitution occurs. Except for the functional annotation-based features, these and some supplementary features were also used in this work. The more recently developed method CADD, which can be applied to all types of genetic variants, provided good performance by incorporating conservation metrics, regulatory information, transcript information, and protein level scores that are generated with methods like SIFT and PolyPhen [207].

Protein structure-based features are also attractive to further improve classification performance. However, their use is hampered by the limited availability of structural data. Furthermore, regarding the variants that do have available structure data, the fact that relatively many of these variants are disease-associated complicates the use of this type of feature by introducing a strong bias.

The fact that good classification performances can be obtained implies that the used features, which are mostly derived from sequence data, comprise valuable information about the probability of a genetic variant being neutral or disease-associated. However, this information is rarely utilized to provide better insight into what features actually contribute most to classification outcomes, i.e. what sequence characteristics are predictive for the effect of genetic variants. In this work, we show how we can obtain insight in characteristics of variations associated with disease through predictor interpretation.

We used linear support vector machines, allowing us to extract feature weights from trained classifiers. A high weight indicates a strong contribution of a certain feature to the classifier outcome, and its sign indicates if it is predictive for neutral (negative weight) or disease-associated (positive weight) variants (Figure 6.1). To further enhance interpretation potential and performance of the linear classifiers, we trained separate classifiers on subsets that contain variants with the same reference amino acid. This was done based on the assumption that feature importance might be different per type of amino acid substitution. For example, a surrounding sequence with many small amino acids might be a high risk in case of substitutions from small to large amino acids, whereas substitutions from small to other small amino acids in the same surrounding might have a lower risk. Extracting feature importance from classifiers trained on the variant subsets could help in revealing such differences. Although it is not the aim of this paper to introduce a competitive predictor, we demonstrate that classifiers can be made interpretable without significant loss in prediction performance.

6.2. RESULTS AND DISCUSSION

To characterize variants, we used five different sequence-derived feature categories (Figure 6.2a) that were derived from different types of sequence data (Figure 6.2b). Most of these features were inspired by the well performing method SNPs&GO [195]. The *Amino acid substitution category* consists of 20 features that capture the amino acid substitution by setting the reference amino acid to minus one, the mutant amino acid to one, and all other amino acids to zero [195]. These features were added because it is expected that different amino acid substitutions have different probabilities of resulting in a functional effect. *Surrounding sequence features* capture amino acid counts in a window of 19 residues around the substituted amino acid [195], which can be informative for structural surroundings. For example, the features could (implicitly) capture information about backbone disorder, solvent accessibility, and secondary structure. *Conservation features* capture how conserved the mutated position is based on a multiple sequence alignment (MSA) with similar proteins. Two features capture how often the reference and the mutant amino acid occur in the set of amino acids at the mutation

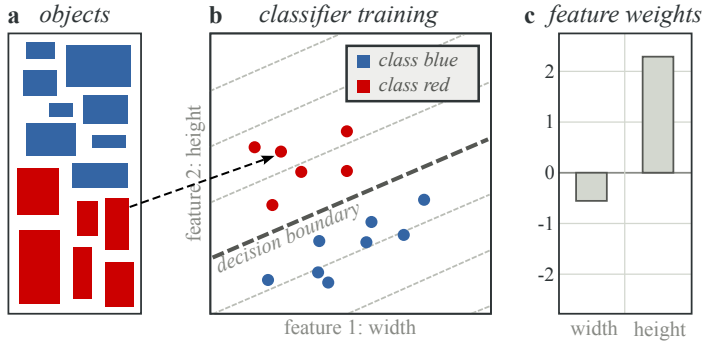


Figure 6.1: Extracting feature weights from trained classifiers *a)* For illustration, objects in two classes (blue and red) are represented by rectangles and characterized by the features "width" and "height". *b)* By measuring widths and heights, objects are mapped to a two dimensional grid (feature space). Classifier training results in the decision boundary that separates the two classes of objects. *c)* Feature importance can be deducted from the slope of the decision boundary. The height is more important than the width, hence the higher (absolute) weight for this feature. The sign indicates for what class the feature is predictive. Blue rectangles are generally wider, hence the negative weight for the width feature. Red rectangles are generally taller, hence the positive weight for the height feature.

position in the MSA (Figure 6.2b). An often occurring reference amino acid indicates strong conservation and therefore a high risk of a functional effect upon mutation. In contrast, a low risk is expected in case of a high occurrence of the mutant amino acid. Two additional features capture the number of proteins in the alignment. These features were added to account for limited availability of homologous sequences, in which case the first two conservation features are expected to be less informative. *Physicochemical conservation features* capture if physicochemical properties of the mutant amino acid are much different compared to those of the amino acids at the mutation position in the MSA. These features were added based on the assumption that, for example, introducing a hydrophobic amino acid at a position where none of the amino acids at that position in the MSA is hydrophobic, might affect protein function. Finally, based on recent work showing an enrichment of deleterious variants in Pfam domains [208], *domain features* capture if a variant resides within a Pfam domain, family, or clan.

For classifier training, we used a set of 171,257 human nonsynonymous SNPs: 149,850 neutral variants from the 1000 Genomes Project and 21,407 disease-associated variants from the SwissProt *humsavar* data base [209, 210] (S1 Information). The variants were split into subsets containing variants with the same reference amino acid. Because the tryptophan, tyrosine, and phenylalanine subsets were too small for classifier training, these were combined into one subset. The resulting variant subsets are listed in Table 6.1. Classifiers were trained on the subsets separately. Afterwards, feature weights were extracted from the trained classifiers (Figure 6.1). This was done using each of the five feature categories separately (Figure 6.2a) and once using all features.

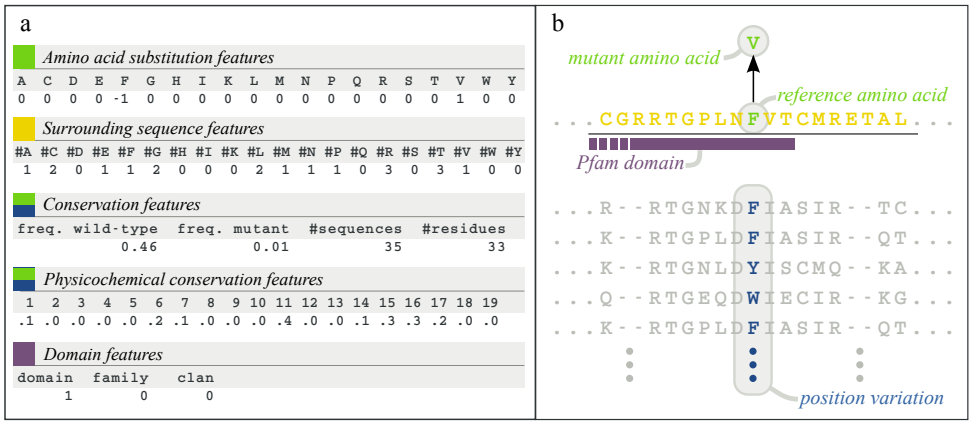


Figure 6.2: Feature categories *a)* Five feature categories and their corresponding features. The colors indicates from which type of sequence data in part B the features were derived. *b)* Sequence and annotation data used to derive variant features; the amino acid substitution (green), the surrounding sequence (yellow), the amino acid variation in similar proteins (blue), and Pfam annotations (purple).

For clarity, practical application of our predictor is different compared to existing methods. In our case there are 18 different classifiers instead of one. Which classifier is applied depends on the variant for which a prediction is desired. For example, if this variant results in an amino acid substitution from Glutamine (reference) to Histidine (mutant), than the classifier that is trained on all variants with reference amino acid Glutamine will be used for prediction.

ENHANCED CLASSIFIER INTERPRETATION

AMINO ACID SUBSTITUTION FEATURES

Extracted feature weights from classifiers trained using the *amino acid substitution features* are visualized using a heat map in Figure 6.3a. Here, each row shows feature weights obtained from one subset classifier, i.e. a classifier trained on one of the variant subsets. For example, the colors in the top row correspond to the weights obtained from the classifier trained on all variants with aspartic acid (D) as reference amino acid. A positive weight (red) indicates that the feature (the mutant amino acid in this case) is predictive for disease-association whereas a negative weight (blue) indicates importance for neutral variants. The higher the (absolute) weight, the higher the feature importance. Using the top row as example again, the low weight of the glutamic acid feature (column E) indicates that a substitution from aspartic acid to glutamic acid is relatively safe, whereas the high weight of the glycine feature (column G) indicates that a substitution from aspartic acid to glycine is relatively dangerous. Gray elements indicate amino acid substitutions that do not occur in our data set, since these require more than one muta-

Table 6.1: Number of variants and proteins per subset

subset		# variants	(%)	# disease	# neutral	# proteins
Alanine	A	14,852	(0.09)	1,294	13,558	7,891
Arginine	R	28,544	(0.17)	3,687	24,857	10,364
Asparagine	N	5,968	(0.03)	624	5,344	4,132
Aspartic acid	D	7,715	(0.05)	1,040	6,675	4,864
Cysteine	C	3,285	(0.02)	1,174	2,111	2,166
Glutamic acid	E	8,618	(0.05)	903	7,715	5,269
Glutamine	Q	4,723	(0.03)	435	4,288	3,467
Glycine	G	12,008	(0.07)	2,648	9,360	6,377
Histidine	H	4,319	(0.03)	532	3,787	3,170
Isoleucine	I	7,985	(0.05)	701	7,284	5,052
Leucine	L	8,206	(0.05)	1,584	6,622	4,988
Lysine	K	5,419	(0.03)	441	4,978	3,793
Methionine	M	4,950	(0.03)	503	4,447	3,598
Proline	P	11,910	(0.07)	1,152	10,758	6,587
Serine	S	11,541	(0.07)	1,165	10,376	6,522
Threonine	T	11,007	(0.06)	891	10,116	6,388
Valine	V	12,771	(0.07)	940	11,831	7,129
WYF		7,436	(0.04)	1,693	5,743	4,593
		171,257	(1.00)	21,407	149,850	16,523

tion at the nucleotide level. Additionally, the feature weights obtained from the classifier that was trained on the entire data set are shown in the single row at the bottom.

The classifier trained on the entire data set (bottom row) only has twenty features to capture the risks of the different amino acid substitutions. For interpretation, the low weight of the methionine feature indicates that substitutions from and to methionine are relatively safe. In contrast, the weights of the subset classifiers offer much richer interpretations. Here it can be observed that substitutions from threonine to methionine are relatively safe, but that substitutions the other way around (from methionine to threonine) are relatively dangerous.

For validation, the heat map in Figure 6.3b shows the log odds ratios between the neutral and disease-associated variants in our data set that were calculated using the amino acid substitution counts. Here, high values indicate relatively dangerous variants, i.e. variants that are relatively often disease related, and low values indicate relatively safe variants. The feature weights of the subset classifiers in Figure 6.3a clearly reflect the log odds ratios, thereby showing that the subset classifiers successfully learned the 'risks' of the different amino acid substitutions.

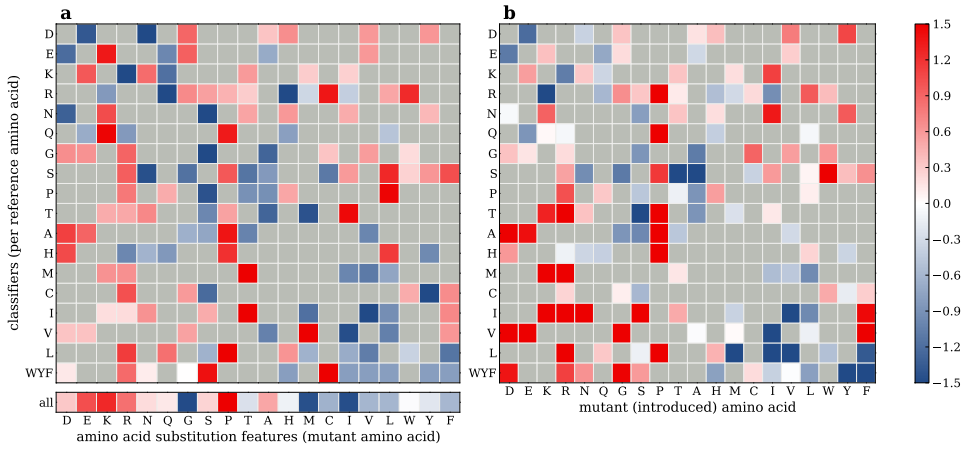


Figure 6.3: Amino acid substitution feature weights. *a)* Heat map showing feature weights obtained from classifiers trained using the amino acid substitution features. The rows show feature weights obtained per variant subset classifier. The single row at the bottom shows feature weights obtained from a classifier trained on the entire set of variants. The rows and columns are ordered based on amino acid properties [211]. Low (blue) and high (red) weights indicate that the feature is predictive for neutral and disease-associated variants respectively. Gray cells indicate amino acid substitutions that do not occur in the data set, because these substitutions require more than one mutation in the reference codon. *b)* Heat map showing log odds ratios between neutral and disease-associated variants that were obtained by counting the amino acid substitutions in our data set. Here, low (blue) and high (red) values indicate that substitutions occur relatively often in the set of neutral and disease-associated variants, respectively.

SURROUNDING SEQUENCE FEATURES

Resulting weights of classifiers trained using the *surrounding sequence features* are shown in Figure 6.4. In this case, most columns show consistently signed weights, which indicates that the same general rules hold for different amino acid substitutions. For example, it is easy to observe that in a serine-rich surrounding (`#S`), any amino acid substitution is relatively safe, independent of what the reference amino acid is. The weights of the classifier trained on the entire set of variants (bottom row) show that the same rules can indeed be learned using the entire set of variants.

For the sequence surrounding features, enhancing interpretation by using the subset approach therefore seems limited. However, some interesting details can still be observed that cannot be derived from the classifier trained on the entire data set. For example, the cysteine subset classifier (`C`) shows a very negative weight for the cysteine count feature (`#C`), indicating that in a cysteine-rich surrounding, substituting cysteines is relatively dangerous. This might be explained by the fact that such variants potentially break disulfide bridges [212]. Similarly, in a glycine-rich surrounding (`#G`), substituting glycines (`G`) shows a relatively high risk of being disease-associated, which might be related to chang-

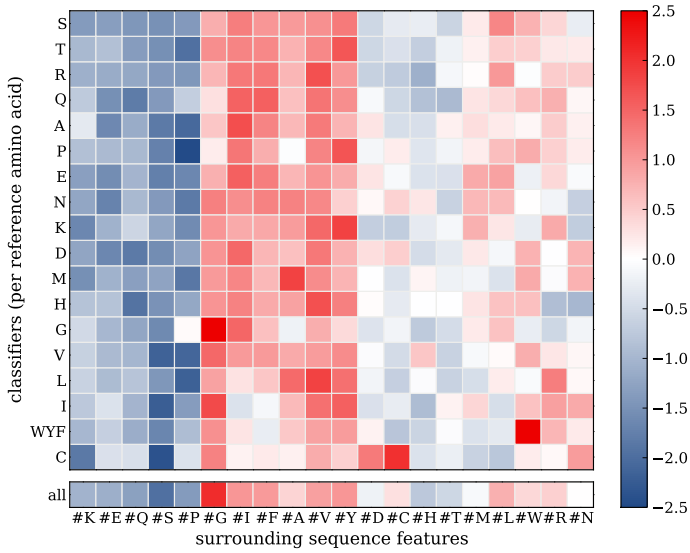


Figure 6.4: Surrounding sequence feature weights. Heat maps showing feature weights obtained from classifiers trained using the surrounding sequence features. The rows show feature weights obtained per variant subset classifier. Both the rows and the columns are hierarchically clustered (complete linkage). The single row at the bottom shows feature weights obtained from a classifier trained on the entire set of variants. Low (blue) and high (red) weights indicate that the feature is predictive for neutral and disease-associated variants respectively.

ing conformational entropy of a flexible region.

The columns were clustered using hierarchical clustering (complete linkage), which reveals a cluster with positive values (red cluster) containing hydrophobic amino acids. This indicates that amino acid substitutions are relatively dangerous in a hydrophobic sequence surrounding, which is consistent with the fact that variants in the hydrophobic protein core have a high-risk of disrupting thermodynamic stability.

PHYSICOCHEMICAL CONSERVATION FEATURES

The *physicochemical conservation features* capture whether there is a large physicochemical distance between the mutant amino acid and the amino acids at the same position in the MSA (Figure 6.2b). For defining physicochemical distances, we used so called amino acid scales: mappings from amino acids to corresponding values that capture some physicochemical property, e.g. hydrophobicity. Many amino acid scales are collected in the AAIndex database [97], but the majority of these scales are highly correlated. We therefore used 19 amino acid scales that were derived from the AAIndex database using VARIMAX [98]. This set contains independent amino acid scales (which is desired for classification performance) of which as many as possible are still closely

Table 6.2: The amino acid scales that corresponds the best to physicochemical properties

Scale	Property
1	Hydrophobicity, β-sheet
2	α-Helix
3	Bulkiness (volume/size/mass)
4	Amino acid composition
7	Isoelectric point
8	β-sheet

related to physicochemical properties (which is desired for interpretation). The amino acid scales that have a strong correlation to physicochemical properties, i.e. the interpretable scales, are given in Table 6.2. The AAIndex scales that best correlate to the derived scales are given in S1 Table.

For calculating these features, the amino acids are first mapped to characterizing values using the amino acid scale, after which the minimal distance between the mutant amino acid and the amino acids at the same position in the MSA is calculated. This is done for all 19 amino acid scales. As an example, a large mutant amino acid on a position where the MSA contains only small amino acids will result in a large bulkiness (scale 3) distance. These features basically capture conservation of physicochemical properties.

This category captures a property of the mutant amino acid, while our classifiers are trained on variants with the same reference amino acid, which complicates interpretation. Theoretically, splitting the variants per amino acid substitution (150 out of the 380 possible substitutions, since we only consider substitutions that are a result of a single mutation in the codon) could improve interpretation possibilities, but these subsets would be too small for classifier training. Still, some intuitive results can be observed (Figure 6.5). For example, cysteines are small and often buried, so replacing these by a large amino acid may disrupt protein core packing. Conversely, a difference in bulkiness when replacing the relatively large amino acids phenylalanine, tyrosine or tryptophan, is found to be relatively safe.

CONSERVATION AND DOMAIN FEATURES

The *conservation features* indicate how conserved a mutated position is. As expected, variants for which the reference amino acid often occur at the same position in homologous sequences have a high risk of being disease-associated, and variants for which the mutant amino acid often occur on the same position in homologous sequences are relatively safe (S1 Figure). These rules hold for all variants, independent of what amino acid substitution they induce. Similarly, considering the *domain features*, it holds for all variants that the risk of being disease-associated is relatively high if it resides in a Pfam domain (S2 Figure). For these features, the classifiers trained on the variant subsets therefore do not provide better interpretations than the classifier trained on all variants.

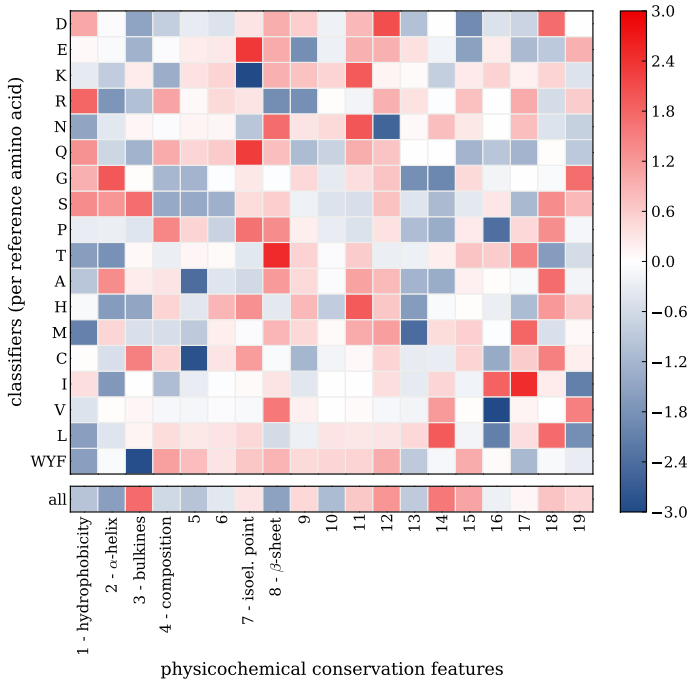


Figure 6.5: Physicochemical conservation feature weights. Heat maps showing feature weights obtained from classifiers trained using the physicochemical conservation features. The rows show feature weights obtained per subset classifier. The single row at the bottom shows feature weights obtained from a classifier trained on the entire set of variants. Low (blue) and high (red) weights indicate that the feature is predictive for neutral and disease-associated variants respectively.

6

ALL FEATURES COMBINED

Considering the classifiers trained using *all features*, the resulting feature weights (S3 Figure) show that the conservation and domain features generally obtain high (absolute) weights, indicating that these feature categories are most predictive. However, high weights for certain features in other categories show that these also contribute to prediction and interpretation. For example, for variants resulting in alanine substitutions, not only high conservation is a strong indicator for disease-association, but also if the alanine is substituted to an aspartic (or glutamic) acid. For the set of variants with phenylalanine, tryptophan, and tyrosine as reference amino acid, it can be observed that substitutions to less bulky amino acids, and especially to cysteines, have a relatively low risk of being disease associated.

Table 6.3: Classification performances (AUC)

Features	C_S*	C_E**	Δ
Linear support vector machine classifiers			
all features	0.833	0.813	0.020
amino acid substitution	0.683	0.587	0.096
surrounding sequence	0.714	0.673	0.041
conservation	0.775	0.765	0.010
physicochemical conservation	0.712	0.633	0.079
domain	0.720	0.676	0.044
RBF support vector machine classifiers			
all features	0.845	0.858	-0.013
Other prediction methods			
SIFT	-	0.803	-
PolyPhen 2	-	0.807	-

*combined subset classifiers, **classifier trained on all variants

CLASSIFIER PERFORMANCES

Interpreting a classifier is only useful if it demonstrates good prediction performance, as otherwise the used features are not predictive and consequently interpretation of their weights is dangerous. To assess classifier performance, we used ten-fold cross-validation using the area under the receiver operator curve (AUC) as performance measure. Again, classifiers were tested using each feature category separately and one classifier was tested using all features. Classifiers were tested for all variant subsets; to obtain a combined subset classifier (C_S) result, all test set predictions were combined to generate an ROC-curve. For comparison, classifiers that were trained on the entire set of variants (C_E) were also tested.

Resulting performances are given in Table 6.3 (more results can be found in S2 Table and S4 Figure). In case of the linear support vector machines, subset classifiers (C_S) consistently outperformed the classifiers trained on the entire set of variations (C_E). The subset approach thus not only improves interpretation, it also results in better classification performances (for linear classifiers). Best performance was obtained using the subset classifier trained on all features, resulting in 0.833 AUC (Figure 6.6).

To compare this result with existing methods, we applied the two often used methods SIFT and PolyPhen2 to our data set. With AUCs of 0.803 and 0.807 respectively, both methods were outperformed by our interpretable classifier (Figure 6.7). We also compared our linear approach to one using a non-linear classifier, which may be better suited for a complex classification problem such as this. With a cross-validation result of 0.858 (Figure 6.6), this was indeed the case for a non-linear SVM (RBF kernel). However, this classifier does not allow for interpretation. By using the subset approach with linear classifiers, we managed to enable interpretation with only a limited loss in performance

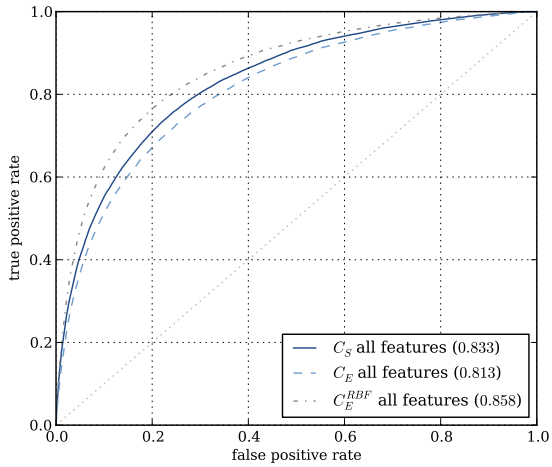


Figure 6.6: ROC-curves showing classifier performances using all features. In blue, performances for linear support machines using the combined subset classifier approach (C_S), and for a classifier trained on the entire set of variants (C_E). In gray the performance of a non-linear support vector machine (RBF kernel) trained on the entire set of variants.

(0.833 vs. 0.858).

6.3. CONCLUSION

In this work, we propose to investigate properties of disease-causing genetic variants, by exploiting predictors trained to distinguish between such variants and neutral mutations. We take a linear classification approach, allowing us to interpret feature weights in a straightforward manner. The results showed that our approach enables interpretation with only limited performance loss compared to the use of non-linear classifiers. This is useful for users that are interested in specific disease-associated variants, providing better understanding about mechanisms potentially responsible for functional effects. Furthermore, when considering large sets of variants, the approach also provides pointers to help find general mechanisms resulting in neutral or disease-associated variants.

6.4. METHODS

HUMAN PROTEIN SEQUENCES

Human protein sequences were obtained from the UniProt website (June 3, 2013) using a query for canonical human proteins with keyword "Complete proteome", and from the Ensembl (version 71) FTP server (see S1 Information for the URLs used). Only the protein sequences that were identical in both sets were selected, thereby providing a one-

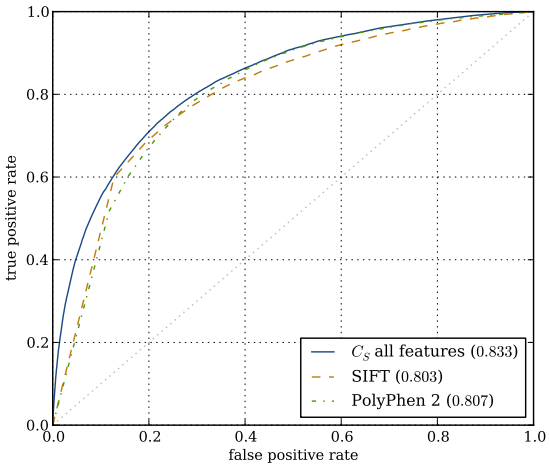


Figure 6.7: ROC-curves showing classifier performance compared to SIFT and PolyPhen 2. In blue, performance using the combined subset classifier approach (C_S). In orange and green, performances of SIFT and Polyphen 2 respectively.

to-one mapping from UniProt to Ensembl proteins (S4 File), which facilitated running other prediction methods on our data. Proteins longer than 10,000 residues were considered outliers and therefore removed. This resulted in a set of 18,162 human protein sequences (S3 File).

HUMAN VARIANTS

Disease-associated variants were obtained from the SwissProt human single amino acid variants data base (humsavar release 2013-07), selecting all variants with annotation disease. Non-disease associated variants from the 1000 Genomes Project were obtained by directly querying the database (Dec 2012). An overlap of 676 variants that were both found in the set of disease-associated variants and the set of neutral variants were assumed to be disease-associated and therefore removed from the neutral set. Synonymous SNPs, duplicate variants, variants in the start codon, and substitutions that included other than the twenty unambiguous amino acids were removed. To prevent a bias caused by an unbalanced occurrence of multiple nucleotide mutations in the different classes [210], amino acid substitutions that require more than one mutation in the codon were removed. This resulted in 23,039 disease-associated variants in 1,941 proteins and 216,697 neutral variants in 17,183 proteins.

The human protein sequences were used to filter out variants that do not “fit” the protein sequence, i.e. variants for which the reference amino acid was not found on the specified position in the sequence. Variants for which no protein sequence was available were removed. Also, variants at different locations with an identical surrounding

sequence in a window of nineteen amino acids around the mutation were removed, assuming a mapping from the same DNA mutation to multiple proteins. The resulting data set consists of 171,257 variants in 16,523 proteins, subdivided into 149,850 neutral and 21,407 disease-associated variants (S1 File and S2 File).

The variants were split into twenty subsets, each containing variants with the same reference amino acid. Due to the low number of substituted tryptophans, tyrosines, and phenylalanines, these subsets were combined into one subset, resulting in a total of eighteen subsets. The number of variants per subset are given in Table 6.1.

FEATURE CATEGORIES

Calculation of the different feature categories is described below. A file with feature matrix data (250 MB) is available on request.

Amino acid substitution features – Amino acid substitutions are represented by twenty features, one per amino acid, in which the reference amino acid (S2 File, column 3) is set to −1, the mutant amino acid (S2 File, column 4) is set to 1, and all other amino acids are set to 0 (Figure 6.2). For each variant subset, some features have the same value for each variant in that set. Therefore, these features do not contribute to the classification and were removed. For example, the serine feature was removed from the variant subset with substitutions from serine to other amino acids, because that feature is −1 for all variants in the subset. Also, the aspartic acid, glutamic acid, histidine, lysine, methionine, glutamine, and valine features were removed, because a substitution from serine to any of these amino acids requires more than one mutation in a serine codon, and such mutations are not present in our data set. These feature values are therefore all 0.

Surrounding sequence features – Twenty features, one per amino acid, capture the surrounding sequence of each variant (S2 File, column 6). These twenty features contain amino acid counts of a sequence window of 19 residues around the variant (Figure 6.2a).

Conservation features – Alignments with similar proteins were obtained for each human protein by running a single HHBlits [213] against the redundancy reduced UniProt20 data base version 2013-03 using default parameter settings (S1 Information). For each variant, four conservation features were derived from the multiple sequence alignment (MSA) column at the mutation position: *i)* the frequency of occurrence of the reference amino acid, *ii)* the frequency of occurrence of the mutant amino acid, *iii)* the total number of aligned proteins, and *iv)* the number of aligned residues in this column.

Physicochemical conservation features – These features employ the MSA to capture minimal physicochemical distances between the mutant amino acid and the set of variant amino acids at the mutation position (Figure 6.2b), in which amino acid scales were used to calculate physicochemical distances between two amino acids. Amino acid scales map each amino acid to a value that captures a physicochemical or biochemical property and the AAIndex data base [97] contains a large collection of these scales, many of which are highly correlated. We therefore used a set of 19 uncorrelated scales

derived from the entire AAIndex database [98]. The uncorrelated scales were derived in such a way that some of the scales remain highly correlated to a set of consensus natural scales: Scale 1 has strong correlation with hydrophobicity and β-sheet scales, scale 2 has strong correlation with α-helix scales, scale 3 has strong correlation with bulkiness scales, scale 4 has strong correlation with amino acid composition scales, and scale 7 has strong correlation with isoelectric point scales (Figure 5b in [98]). This way, all amino acid scales data is captured while interpretation is still possible for some of the resulting uncorrelated scales.

Domain features – Pfam version 27.0 [214] was used to predict Pfam domains on the protein sequences (S1 Information). Resulting annotations (S5 File) were used to construct three binary domain features that are set to 1 if the variant resides within a predicted Pfam family, domain, or clan, respectively, or to 0 otherwise.

CLASSIFICATION

A linear support vector machine (LIBSVM [110]) was employed for classification [160], using a linear and RBF kernel for the linear and non-linear classifiers respectively, and using a 10-fold stratified cross-validation (CV) protocol to asses classifier performance [108]. When using the linear kernel, parameter C was set to 0.1; for the RBF kernel we set $C = 1.0$ and $\gamma = 0.01$. Probability estimates were used as classifier output, so that outcomes of the different subset classifiers could be combined.

Classifiers were trained on the variant subsets separately (C_S). Their combined performance was obtained by combining the outcomes of all CV test sets for all subset classifiers and using these to generate an ROC-curve [127] for the entire data set. The area under the ROC-curve (AUC) was used as performance measure. Classifiers were also trained on the entire set of variants (C_E), in which case the average AUC of the ten CV-loops was used as performance measure. For classifier types, C_E and C_S, a classifier was trained for each of the feature categories, and a classifier was trained on all features. Feature scaling was applied to enable the use of data with varying ranges. All feature values were standardized (the feature value subtracted by the mean of the feature vector and the result divided by the standard deviation of the feature vector) so that all feature vectors have zero-mean and unit-variance.

After cross-validation, classifiers were trained on the entire data sets. These classifiers were used to obtain feature weights. For a given set of variants V, the feature weight vector $\mathbf{w}$ from the trained SVM classifier was obtained using:

$$\mathbf{w} = \sum_{v_i \in V} \alpha_i y_i \Phi(v_i), \tag{6.1}$$

in which α_i are the weights assigned to the objects (variants), y_i are the variant labels (-1 for neutral and 1 for disease-associated) and $\Phi(v_i)$ is a function that maps a variation v_i to its feature representation. For comparison, weight vectors are standardized to zero mean and unit standard deviation.

OTHER PREDICTION METHODS

Predictions for our mutation data set were obtained using the two often used prediction methods SIFT [189] and PolyPhen2 [197]. SIFT predictions were obtained using their website, the resulting SIFT scores were used as prediction outcome. Predictions were missing for a total of 4,208 mutations, either because the protein ID (ENSP) or the requested position in the sequence was not found by the current SIFT predictor. PolyPhen2 predictions were also obtained using their website. Both our list of mutations and the FASTA file with human proteins were supplied to the method, which was run using HumDiv as classifier model, GRCh37/hg19 as genome assembly, canonical transcripts, and missense annotations. The resulting Naive Bayes posterior probabilities were used as prediction outcome. No predictions were given for 647 variants. The area under the ROC-curve was used as performance measure.

6.5. SUPPORTING INFORMATION

The supporting information can be accessed online [1].

6.6. ACKNOWLEDGMENTS

The authors would like to acknowledge Janet Thornton for her valuable contributions to this work.

[1] http://plosone.org

7

Discussion

In this work we identified sequence-based differences between extracellular proteins with low and high production levels in the condition of over-expression in *A. niger*, and used this to improve production levels by making a protein's sequence properties more similar to those of proteins with high production levels. In chapter 2, classifiers were constructed for predicting successful high-level production and secretion. The analysis in chapter 3 showed which sequence properties correlate with either low or high production levels, and in chapter 5 we showed how the trained classifier can be employed to improve production levels through protein redesign. Predicting high-level production and redesigning proteins for improved production levels has great potential for application in the biotechnology industry. Predictors can be used to identify proteins that have potential for large-scale production in *A. niger*. The protein design approach can be used to optimize production levels of industrial enzymes. Also, additional analysis and design iterations could help to increase our understanding of the secretion pathway.

Successful improvement of protein production levels through protein design is a very promising result, but application of this approach depends on several important requirements. First of all, a large set of sample proteins is required to enable classifier training. In our case, this was achieved by gathering relatively easy to measure, but low resolution binary information on low- vs. high-level production, based on the observation of a visible band on a gel. Moreover, we did not restrict ourselves to proteins within a certain family, but included any protein that was expected to be secreted. It is difficult to indicate what a sufficiently large data set size is, because this is very much dependent on the (complexity of the) problem at hand. The set of ≈ 350 *A. niger* proteins (homologous expression) was sufficient for training a well-performing classifier, but the performance obtained on the set of ≈ 1000 heterologous proteins from other organisms was much lower. Yet, increasing the data set size does not guarantee improved classification results.

Besides having enough sample proteins for classifier training, it is also required that there are sequence-based differences between the classes under consideration. This can be verified by inspecting how well classes can be separated by sequence-based classifiers. In chapter 2, good class separation for the set of host-own *A. niger* proteins was shown, while in chapter 3 it was shown that class separability of the set of proteins from other organisms is much worse. We therefore targeted only *A. niger* proteins for redesigns. However, class separability alone is not enough if the aim is to alter production levels by changing sequence properties. It shows that a correlation exists but it does not provide any proof for a causal relation. It was therefore (beforehand) unknown if changing sequence properties would affect production levels. By successfully increasing production levels using the suggested approach in this work, we showed that such a causal relation exists. It is important to notice that the mentioned requirements are necessary for our approach to work but they do not guarantee success.

CLASSIFIER-BASED APPROACH: UNBIASED DESIGN, DIFFICULT INTERPRETATION

We improved protein production levels by making their sequence characteristics more similar to proteins that are known to provide high production levels. For this approach,

no human biological knowledge is required to target specific biological mechanisms or processes that are expected to be related to production speed and secretion levels. The redesign is instead guided by a large set of example proteins, which makes it an unbiased approach. By altering sequence-properties, the underlying biology is implicitly targeted. Moreover, our approach circumvents the complex mathematical modeling of the involved biology, which is often difficult and computationally expensive.

As mentioned before, changing protein sequence properties implicitly changes processes and mechanisms involved in protein production and secretion, but it is unknown how they induce production-level changes. We attempted to get better insight into the relation between sequence properties and production levels by making classifiers, often considered black boxes and usually used for prediction purposes only, more transparent. This enabled us to observe which (combinations of) object features are most discriminating, thereby revealing sequence-based differences between proteins with low and high production levels. However, it is difficult to use these observations to draw firm biological conclusions. This is because the trained classifier captures a mapping from sequence properties to production level, while the relation between sequence properties and the biological processes or mechanisms are often unknown. In other words, there is a knowledge-gap, and although this allows for an unbiased design approach, it complicates biological interpretation.

The interpretations presented in this work therefore do not directly contribute to our understanding of the biology behind the secretion pathway, but the results can be very useful to direct additional biological or bioinformatics research. For example, the overrepresentation of tyrosine and other aromatic amino acids in proteins with high production levels is a highly interesting and novel observation that provides a potentially direction for following research. However, with such observations there is the risk of being biased towards what we already know about the subject under investigation. Moreover, interpreting classifiers brings the risk of reductionism. Showing that oranges are healthy food provides no evidence that vitamin C *per se* is healthy. Similarly, the fact there are relatively few lysines in proteins with high production levels does not necessarily mean that lysines *per se* are bad for production levels. The strength of classifiers is that they can find combinations of features that are important for class separation. One should therefore be careful by drawing conclusions based on single features.

For the analysis described in chapter 3, much effort has been put into finding features that improve classification performance and that are more specific than the amino acid composition, so that it may be easier to link the predictive sequence properties to specific biological processes or mechanisms. In particular, much time has been spent on finding patterns of physicochemical properties, both using existing features, such as autocorrelation, pseudo amino acid composition, and quasi-sequence-order descriptors, and by developing new features. Even though initial results looked promising, classification results never outperformed those of classifiers that used only the amino acid compositions.

PROTEIN STRUCTURE

The results in chapters 2 and 3 describe only sequence-based differences between proteins with low and high production levels. An attempt was made to perform similar analyses with protein structure data, but this provided several difficulties that made further exploration of this research direction infeasible. This direction of research is, however, still interesting for future work, although the current difficulties need to be addressed.

First, the availability of protein structure data is limited compared to that of sequence data. This is because measuring structure is more expensive and time consuming than measuring the sequences, but also because (parts of) proteins lack a fixed structure, i.e. they are intrinsically disordered. The upcoming use of predicted structure models improves this situation, but reliable structures will not be available for all proteins and the issue of intrinsically disordered protein structures remains. Second, the available protein structure data has a strong bias towards much-researched (parts of) proteins. For example, the analysis of sequence-based differences between the sequence surrounding neutral and disease-associated missense mutations in chapter 6, showed that there were relatively many disease-associated variants for which the structure of the associated protein (domain) was available. This bias poses the issue that incorporating any structural feature will lead to a bias in predicting disease-associated missense mutations. Third, deriving features from three-dimensional structure data is much more challenging than for one-dimensional sequence data. This holds for both contriving sensible features as well as computing them. Finally, in contrast to sequence data, one needs to be careful about the fact that protein structure data is dependent on the experimental condition in which it is being measured.

Another issue that needs to be addressed for integrating protein structure data in predictors is data accessibility. The limited strictness of the PDB (Proteid Data Bank) file format, which is the file format used for storing protein structure data, and the lack of updating old records to confirm to updated standards severely complicates their use for computational purposes. Moreover, the most commonly used fixed column text-file format is unsustainable, difficult to maintain, and a potential source of errors. Also, mapping residues to corresponding sequence data is not always straightforward. For future research, it would help to have different access points to protein structure data: one for providing easy access to details of single structures, as is currently provided by the RCSB PDB[1], the EBI PDBe[2], and the PDBj [3]; and one for providing a (programmatic) interface for obtaining specific types of data for protein (sub)sets, including mappings to protein sequence and genomic data.

PROTEIN DESIGN

The Gibbs free energy of proteins is usually just below zero to provide them a stable native fold while remaining a bit flexible. This is what makes proteins such useful tools

[1] http://www.rcsb.org/pdb
[2] http://www.ebi.ac.uk/pdbe
[3] http://pdbj.org/

in a cell, but this also makes them very fragile: a single amino acid substitution can break this equilibrium and hence change the protein. For our design approach we therefore choose to stay on the safe side by restricting ourselves to replacing amino acids at the protein surface and away from the active site. Now that this approach proved successful, an interesting next step would be to see if further production level improvements can be obtained while relaxing this restriction. However, targeting the more dangerous areas, such as the buried residues, would probably require additional tools to asses the risk of the substitution.

Most of these tools predict if substitutions will affect protein structure by using an empirical method to calculate free energy changes based on atomic structure data. Alternatively, a classifier like the one developed in chapter 6 could be useful for making such predictions based on the local sequence and the local protein structure environment of the mutated position. However, training such classifiers requires the availability of many single amino acid substitutions with known effect on the associated protein structure, which may be difficult to obtain. Moreover, such a predictor would only be capable of making predictions about single amino acid substitutions. It cannot take into account the possible combinatorial effect of multiple amino acid substitutions. Still, such a classifier might prove a useful addition to the free energy-based approaches.

Successful improvement of production levels through protein redesign is a very promising result, but useful information might also be hidden in the failed redesigns. Analyzing the structural properties of amino acid substitutions in the unsuccessful redesigns might help improving the redesign method, for example through the addition of restrictions in the allowed amino acid substitutions. The analysis-redesign cycle described in this work could also be the basis of a second cycle that focuses on a single protein. For example for the redesigned inulinase the design method in chapter 5 could be employed to generate a range of redesigns with up to 20 mutations. Obtaining quantitative measurements of the obtained extracellular concentrations of these redesigns would allow for a regression analysis to get a more detailed picture of the relation between sequence properties and production levels.

OTHER POTENTIAL REDESIGN TARGETS

The aim of the analysis and redesign approach was to improve extracellular production levels. The same approach also has potential for two other protein properties: solubility and thermostability. Both are often a limiting factor for commercial protein production and for structural studies. This is especially true in case of heterologous gene expression and in the condition of over-expression. Many strategies are available to improve this, but only few works applied it to redesign proteins. Most of these target specific physicochemical properties that are expected to affect thermostability, such as disulfide bridges or solvent exposed charges. It would be interesting to see whether a less biased approach such as the one described in this work could improve results (higher success rates and greater thermostability increases) and if such results could help indicating which (combination of) sequence (and structure) properties have the greatest effect on protein solubility and thermostability.

Besides aiming for a different goal, the approach could also be applied to other targets than (entire) proteins. One could for example target specific protein parts. In chapter 3 it was shown that the sequence properties of signal peptides are also predictive for high-level production, although much less than the protein properties. If the prediction performance for the signal peptide could be improved, for example by using additional features that capture more location specific properties, then redesigning the signal peptide with a similar approach would provide an additional strategy for optimizing production levels. Other potential targets are small peptides for which support vector machines were developed to predict, for example, peptide binding affinities or the ability of peptides to penetrate cells. Such classifiers might also be reversely applied for redesign purposes. A similar approach might also be applicable to DNA regions, such as promoters or UTRs, or RNA components.

To conclude, this work shows how machine learning algorithms can be applied to modify properties of biological molecules, even without exactly understanding how these modification provide the desired result. The successful results of this first attempt to apply such an approach hold great promise for further optimizations and to expand it to a broader range of problems. The approach adds to our capability to customize cells to our needs. As a future perspective, adding to the currently upcoming field of synthetic biology, this might help in constructing tailor made biological components to customize cells for applications in the food, health, and energy industry.

REFERENCES

[1] Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci USA. 1977;74(12):5463–5467.

[2] Bilofsky HS, Burks C, Fickett JW, Goad WB, Lewitter FI, Rindone WP, et al. The GenBank genetic sequence databank. Nucleic Acids Res. 1986;14(1):1–4.

[3] Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL. GenBank. Nucleic Acids Res. 2008;36(suppl 1):D25–D30.

[4] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970;48(3):443–453.

[5] Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981;147(1):195–197.

[6] Durbin R, Eddy S, Krogh A, Graeme M. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press; 1998.

[7] Samuel D. Investigation of ancient egyptian baking and brewing methods by correlative microscopy. Science. 1996;273(5274):488–490.

[8] Kirk O, Borchert TV, Fuglsang CC. Industrial enzyme applications. Curr Opin Biotech. 2002;13(4):345–351.

[9] van Beilen JB, Li Z. Enzyme technology: an overview. Curr Opin Biotech. 2002;13(4):338–344.

[10] Cox PM, Betts RA, Jones CD, Spall SA, Totterdell IJ. Acceleration of global warming due to carbon-cycle feedbacks in a coupled climate model. Nature. 2000;408(6809):184–187.

[11] Root TL, Price JT, Hall KR, Schneider SH, Rosenzweig C, Pounds JA. Fingerprints of global warming on wild animals and plants. Nature. 2003;421(6918):57–60.

[12] Andrady AL. Microplastics in the marine environment. Mar Pollut Bull. 2011;62(8):1596–1605.

[13] Sims RE, Mabee W, Saddler JN, Taylor M. An overview of second generation biofuel technologies. Bioresource Technol. 2010;101(6):1570–1580.

[14] Peralta-Yahya PP, Zhang F, Del Cardayre SB, Keasling JD. Microbial engineering for the production of advanced biofuels. Nature. 2012;488(7411):320–328.

[15] Shah AA, Hasan F, Hameed A, Ahmed S. Biological degradation of plastics: a comprehensive review. Biotechnol Adv. 2008;26(3):246–265.

[16] Sivan A. New perspectives in plastic biodegradation. Curr Opin Biotech. 2011;22(3):422–426.

[17] Pel HJ, de Winde JH, Archer DB, Dyer PS, Hofmann G, Schaap PJ, et al. Genome sequencing and analysis of the versatile cell factory *Aspergillus niger* CBS 513.88. Nat Biotechnol. 2007;25(2):221–231.

[18] Lodish H, Berk A, Kaiser CA, Krieger M, Bretscher A, Ploegh H, et al. Molecular Cell Biology. 7th ed. W. H. Freeman; 2012.

[19] Archer DB, Peberdy JF. The molecular biology of secreted enzyme production by fungi. Crit Rev Biotechnol. 1997;17(4):273–306.

[20] Gouka RJ, Punt PJ, Van Den Hondel CAMJJ. Efficient production of secreted proteins by Aspergillus: progress, limitations and prospects. Appl Microbiol and Biot. 1997;47(1):1–11.

[21] Conesa A, Punt PJ, van Luijk N, van den Hondel CAMJJ. The secretion pathway in filamentous fungi: a biotechnological view. Fungal Genet Biol. 2001;33(3):155–171.

[22] Verdoes JC, Punt PJ, Van Den Hondel CAMJJ. Molecular genetic strain improvement for the overproduction of fungal proteins by filamentous fungi. Appl Microbiol Biot. 1995;43(2):195–205.

[23] Lee SY, Lee DY, Kim TY. Systems biotechnology for strain improvement. TRENDS in Biotechnology. 2005;23(7):349–358.

[24] Archer DB, MacKenzie DA, Jeenes DJ, Roberts IN. Proteolytic degradation of heterologous proteins expressed in *Aspergillus niger*. Biotechnol Lett. 1992;14(5):357–362.

[25] van den Hombergh JPTW, van de Vondervoort PJI, Fraissinet-Tachet L, Visser J. Aspergillus as a host for heterologous protein production: the problem of proteases. Trends Biotechnol. 1997;15(7):256–263.

[26] Verdoes JC, Punt PJ, Schrickx JM, van Verseveld HW, Stouthamer AH, van den Hondel CA. Glucoamylase overexpression in *Aspergillus niger*: Molecular genetic analysis of strains containing multiple copies of the *glaA* gene. Transgenic Res. 1993;2(2):84–92.

[27] Verdoes JC, Punt PJ, Stouthamer AH, van den Hondel CA. The effect of multiple copies of the upstream region on expression of the *Aspergillus niger* glucoamylase-encoding gene. Gene. 1994;145(2):179–187.

[28] Verdoes JC, van Diepeningen AD, Punt PJ, Debets AJ, Stouthamer AH, van den Hondel CA. Evaluation of molecular and genetic approaches to generate glucoamylase overproducing strains of *Aspergillus niger*. J Biotechnol. 1994;36(2):165–175.

[29] Gustafsson C, Govindarajan S, Minshull J. Codon bias

and heterologous protein expression. Trends Biotechnol. 2004;22(7):346–353.

[30] Plotkin JB, Kudla G. Synonymous but not the same: the causes and consequences of codon bias. Nat Rev Genet. 2010;12(1):32–42.

[31] Tsang A, Butler G, Powlowski J, Panisko EA, Baker SE. Analytical and computational approaches to define the *Aspergillus niger* secretome. Fungal Genet Biol. 2009;46(1):S153.

[32] Lu X, Sun J, Nimtz M, Wissing J, Zeng AP, Rinas U. The intra- and extracellular proteome of *Aspergillus niger* growing on defined medium with xylose or maltose as carbon substrate. Microb Cell Fact. 2010;9(1):1–13.

[33] Braaksma M, Martens-Uzunova ES, Punt PJ, Schaap PJ. An inventory of the *Aspergillus niger* secretome by combining *in silico* predictions with shotgun proteomics data. BMC Genomics. 2010;11(1):584.

[34] Punt PJ, Van Biezen N, Conesa A, Albers A, Mangnus J, Van Den Hondel C. Filamentous fungi as cell factories for heterologous protein production. Trends Biotechnol. 2002;20(5):200–206.

[35] Nevalainen KM, Te'o VSJ, Bergquist PL. Heterologous protein expression in filamentous fungi. Trends Biotechnol. 2005;23(9):468–474.

[36] Lubertozzi D, Keasling JD. Developing *Aspergillus* as a host for heterologous expression. Biotechnol Adv. 2009;27(1):53–75.

[37] Fleißner A, Dersch P. Expression and export: recombinant protein production systems for *Aspergillus*. Appl Microbiol Biot. 2010;87(4):1255–1270.

[38] Pantazes RJ, Grisewood MJ, Maranas CD. Recent advances in computational protein design. Curr Opin Struct Biol. 2011;21(4):467–472.

[39] Samish I, MacDermaid CM, Perez-Aguilar JM, Saven JG. Theoretical and computational protein design. Annu Rev Phys Chem. 2011;62:129–149.

[40] Saven JG. Computational protein design: engineering molecular diversity, nonnatural enzymes, nonbiological cofactor complexes, and membrane proteins. Curr Opin Chem Biol. 2011;15(3):452–457.

[41] Röthlisberger D, Khersonsky O, Wollacott AM, Jiang L, DeChancie J, Betker J, et al. Kemp elimination catalysts by computational enzyme design. Nature. 2008;453(7192):190–195.

[42] Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, Baker D. Design of a novel globular protein fold with atomic-level accuracy. Science. 2003;302(5649):1364–1368.

[43] Butterfoss GL, Kuhlman B. Computer-based design of novel protein structures. Annu Rev Biophys Biomol Struct. 2006;35:49–65.

[44] Privett HK, Kiss G, Lee TM, Blomberg R, Chica RA, Thomas LM, et al. Iterative approach to computational enzyme design. Proc Natl Acad Sci USA. 2012;109(10):3790–3795.

[45] Siegel JB, Zanghellini A, Lovick HM, Kiss G, Lambert AR, Clair JLS, et al. Computational design of an enzyme catalyst for a stereoselective bimolecular Diels-Alder reaction. Science. 2010;329(5989):309–313.

[46] Murphy PM, Bolduc JM, Gallaher JL, Stoddard BL, Baker D. Alteration of enzyme specificity by computational loop remodeling and design. Proc Natl Acad Sci USA. 2009;106(23):9215–9220.

[47] Whitehead TA, Chevalier A, Song Y, Dreyfus C, Fleishman SJ, De Mattos C, et al. Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. Nat Biotechnol. 2012;30(6):543–548.

[48] Kapp GT, Liu S, Stein A, Wong DT, Reményi A, Yeh BJ, et al. Control of protein signaling using a computationally designed GTPase/GEF orthogonal pair. Proc Natl Acad Sci USA. 2012;109(14):5277–5282.

[49] Fleishman SJ, Whitehead TA, Ekiert DC, Dreyfus C, Corn JE, Strauch EM, et al. Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. Science. 2011;332(6031):816–821.

[50] Slovic AM, Kono H, Lear JD, Saven JG, DeGrado WF. Computational design of water-soluble analogues of the potassium channel KcsA. Proc Natl Acad Sci USA. 2004;101(7):1828–1833.

[51] Gribenko AV, Patel MM, Liu J, McCallum SA, Wang C, Makhatadze GI. Rational stabilization of enzymes by computational redesign of surface charge–charge interactions. Proc Natl Acad Sci USA. 2009;106(8):2601–2606.

[52] Joo JC, Pack SP, Kim YH, Yoo YJ. Thermostabilization of *Bacillus circulans* xylanase: Computational optimization of unstable residues based on thermal fluctuation analysis. J Biotechnol. 2011;151(1):56–65.

[53] Tian J, Wang P, Gao S, Chu X, Wu N, Fan Y. Enhanced thermostability of methyl parathion hydrolase from *Ochrobactrum* sp. M231 by rational engineering of a glycine to proline mutation. FEBS J. 2010;277(23):4901–4908.

[54] Webb AR. Statistical pattern recognition. 3rd ed. Wiley; 2011.

[55] de Ridder D, de Ridder J, Reinders MJT. Pattern recognition in bioinformatics. Brief Bioinform. 2013;14(5):633–47.

[56] van't Veer LJ, Dai H, Van De Vijver MJ, He YD, Hart AA, Mao M, et al. Gene expression profiling predicts clinical outcome of breast cancer. Nature. 2002;415(6871):530–536.

[57] Chi SM, Nam D. WegoLoc: accurate prediction of protein subcellular localization using weighted Gene Ontology terms. Bioinformatics. 2012;28(7):1028–1030.

[58] Briesemeister S, Rahnenführer J, Kohlbacher O. YLoc—an interpretable web server for predicting subcellular localization. Nucleic Acids Res. 2010;38(suppl 2):W497–W502.

[59] Blum T, Briesemeister S, Kohlbacher O. MultiLoc2: integrating phylogeny and Gene Ontology terms improves subcellular protein localization prediction. BMC Bioinform. 2009;10(1):274.

[60] Zheng X, Liu T, Wang J. A complexity-based method for

predicting protein subcellular location. Amino Acids. 2009;37(2):427–433.

[61] Chou KC, Shen HB. Cell-PLoc: a package of Web servers for predicting subcellular localization of proteins in various organisms. Nat Protoc. 2008;3(2):153–162.

[62] Garg A, Raghava GP. ESLpred2: improved method for predicting subcellular localization of eukaryotic proteins. BMC Bioinform. 2008;9(1):503.

[63] Tantoso E, Li KB. AAIndexLoc: predicting subcellular localization of proteins based on a new representation of sequences using amino acid indices. Amino Acids. 2008;35(2):345–353.

[64] Emanuelsson O, Brunak S, von Heijne G, Nielsen H. Locating proteins in the cell using TargetP, SignalP and related tools. Nat Protoc. 2007;2(4):953–971.

[65] Horton P, Park KJ, Obayashi T, Fujita N, Harada H, Adams-Collier CJ, et al. WoLF PSORT: protein localization predictor. Nucleic Acids Res. 2007;35(suppl 2):W585–W587.

[66] Su EC, Chiu HS, Lo A, Hwang JK, Sung TY, Hsu WL. Protein subcellular localization prediction based on compartment-specific features and structure conservation. BMC Bioinform. 2007;8(1):330.

[67] Jia P, Qian Z, Zeng Z, Cai Y, Li Y. Prediction of subcellular protein localization based on functional domain composition. Biochem Bioph Res Co. 2007;357(2):366–370.

[68] Pierleoni A, Martelli PL, Fariselli P, Casadio R. BaCelLo: a balanced subcellular localization predictor. Bioinformatics. 2006;22(14):e408–e416.

[69] Hua S, Sun Z. Support vector machine approach for protein subcellular localization prediction. Bioinformatics. 2001;17(8):721–728.

[70] Emanuelsson O, Nielsen H, Brunak S, von Heijne G. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J Mol Biol. 2000;300(4):1005–1016.

[71] Brameier M, Krings A, MacCallum RM. NucPred—predicting nuclear localization of proteins. Bioinformatics. 2007;23(9):1159–1160.

[72] Hirose S, Noguchi T. ESPRESSO: A system for estimating protein expression and solubility in protein expression systems. Proteomics. 2013;13(9):1444–1456.

[73] Smialowski P, Doose G, Torkler P, Kaufmann S, Frishman D. PROSO II–a new method for protein solubility prediction. FEBS J. 2012;279(12):2192–2200.

[74] Magnan CN, Randall A, Baldi P. SOLpro: accurate sequence-based prediction of protein solubility. Bioinformatics. 2009;25(17):2200.

[75] Smialowski P, Martin-Galiano AJ, Mikolajka A, Girschick T, Holak TA, Frishman D. Protein solubility: sequence based prediction and experimental verification. Bioinformatics. 2007;23(19):2536–2542.

[76] Idicula-Thomas S, Kulkarni AJ, Kulkarni BD, Jayaraman VK, Balaji PV. A support vector machine-based method for predicting the propensity of a protein to be soluble or to form inclusion body on overexpression in Escherichia coli. Bioinformatics. 2006;22(3):278–284.

[77] Petersen TN, Brunak S, von Heijne G, Nielsen H. SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat Methods. 2011;8(10):785–786.

[78] Xia XY, Ge M, Wang ZX, Pan XM. Accurate Prediction of Protein Structural Class. PLOS ONE. 2012;7(6):e37653.

[79] Sahu SS, Panda G. A novel feature representation method based on Chou's pseudo amino acid composition for protein structural class prediction. Comp Biol Chem. 2010;34(5):320–327.

[80] Zhang TL, Ding YS, Chou KC. Prediction protein structural classes with pseudo-amino acid composition: approximate entropy and hydrophobicity pattern. J Theor Biol. 2008;250(1):186–193.

[81] Shamim MTA, Anwaruddin M, Nagarajaram HA. Support Vector Machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs. Bioinformatics. 2007;23(24):3320–3327.

[82] Kedarisetti KD, Kurgan L, Dick S. Classifier ensembles for protein structural class prediction with varying homology. Biochem Bioph Res Co. 2006;348(3):981–988.

[83] Rentzsch R, Orengo CA. Protein function prediction–the power of multiplicity. Trends Biotechnol. 2009;27(4):210–219.

[84] Lobley AE, Nugent T, Orengo CA, Jones DT. FFPred: an integrated feature-based function prediction server for vertebrate proteomes. Nucleic Acids Res. 2008;36(suppl 2):W297–W302.

[85] Shen HB, Chou KC. EzyPred: a top–down approach for predicting enzyme functional classes and subclasses. Biochem Bioph Res Co. 2007;364(1):53–59.

[86] Jensen LJ, Gupta R, Staerfeldt HH, Brunak S. Prediction of human protein function according to Gene Ontology categories. Bioinformatics. 2003;19(5):635–642.

[87] Cai C, Han L, Ji ZL, Chen X, Chen YZ. SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res. 2003;31(13):3692–3697.

[88] Leslie C, Eskin E, Noble WS. The spectrum kernel: a string kernel for SVM protein classification. In: Pacific Symposium on Biocomputing. vol. 575. Hawaii, USA.; 2002. p. 564–575.

[89] Chou KC. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins: Structure, Function, and Bioinformatics. 2001;43(3):246–255.

[90] Chou KC. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics. 2005;21(1):10–19.

[91] Chou KC. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. Biochem Bioph Res Co. 2000;278(2):477–483.

[92] Moreau G, Broto P. Autocorrelation of molecular structures, application to SAR studies. Nouv J Chim. 1980;4(12):757–764.

[93] Moran PAP. Notes on continuous stochastic phenom-

ena. Biometrika. 1950;37(1/2):17–23.

[94] Geary RC. The contiguity ratio and statistical mapping. The Incorporated Statistician. 1954;5(3):115–146.

[95] Li ZR, Lin HH, Han LY, Jiang L, Chen X, Chen YZ. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucleic Acids Res. 2006;34(suppl 2):W32–W37.

[96] Dubchak I, Muchnik I, Holbrook SR, Kim SH. Prediction of protein folding class using global description of amino acid sequence. Proc Natl Acad Sci USA. 1995;92(19):8700–8704.

[97] Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M. AAindex: amino acid index database, progress report 2008. Nucleic Acids Res. 2008;36(suppl 1):D202–D205.

[98] Georgiev AG. Interpretable numerical descriptors of amino acid space. J Comp Biol. 2009;16(5):703–723.

[99] van den Berg BA, Nijkamp JF, Reinders MJT, Wu L, Pel HJ, Roubos JA, et al. Sequence-based prediction of protein secretion success in *Aspergillus niger*. In: Proceedings of Pattern Recegnition in Bioinformatics 2010. Springer; 2010. p. 3–14.

[100] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3:1157–1182.

[101] Kurgan L, Razib AA, Aghakhani S, Dick S, Mizianty M, Jahandideh S. CRYSTALP2: sequence-based protein crystallization propensity prediction. BMC Struct Biol. 2009;9:50.

[102] Klee EW, Sosa CP. Computational classification of classically secreted proteins. Drug Discov Today. 2007;12(5-6):234–40.

[103] Nielsen H, Engelbrecht J, Brunak S, Von Heijne G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Eng Des Sel. 1997;10(1):1.

[104] Sharp PM, Li WH. The codon adaptation index-a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Res. 1987;15(3):1281.

[105] Benita Y, Wise MJ, Lok MC, Humphery-Smith I, Oosting RS. Analysis of high throughput protein expression in *Escherichia coli*. Mol Cell Proteomics. 2006;5(9):1567.

[106] Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol. 1982;157(1):105–132.

[107] Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. BBA-Protein Struct M. 1975;405(2):442–451.

[108] Wessels LFA, Reinders MJT, Hart AAM, Veenman CJ, Dai H, He YD, et al. A protocol for building and evaluating predictors of disease state based on microarray data. Bioinformatics. 2005;21(19):3755–62.

[109] Duin RPW, Juszczak P, Paclik P, Pekalska E, de Ridder D, Tax DMJ, et al. A Matlab toolbox for pattern recognition. PRTools version 41. 2000;3.

[110] Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM T Intel Syst Techn. 2011;2(3):27.

[111] Mitra N, Sinha S, Ramya TNC, Surolia A. N-linked oligosaccharides as outfitters for glycoprotein folding, form and function. Trends Biochem Sci. 2006;31(3):156–63.

[112] Andersen MR, Salazar MP, Schaap PJ, van de Vondervoort PJI, Culley D, Thykaer J, et al. Comparative genomics of citric-acid-producing *Aspergillus niger* ATCC 1015 versus enzyme-producing CBS 513.88. Genome Res. 2011;21(6):885–897.

[113] Carvalho NDSP, Arentshorst M, Kooistra R, Stam H, Sagt CM, van den Hondel CAMJJ, et al. Effects of a defective ERAD pathway on growth and heterologous protein production in *Aspergillus niger*. Appl Microbiol Biot. 2011;89(2):357–373.

[114] Jacobs DI, Olsthoorn MA, Maillet I, Akeroyd M, Breestraat S, Donkers S, et al. Effective lead selection for improved protein production in *Aspergillus niger* based on integrated genomics. Fungal Genet Biol. 2009;46(1, Supplement):S141 – S152.

[115] Guillemette T, van Peij NNME, Goosen T, Lanthaler K, Robson GD, van den Hondel CAMJJ, et al. Genomic analysis of the secretion stress response in the enzyme-producing cell factory *Aspergillus niger*. BMC Genomics. 2007;8(1):158.

[116] Horton P, Park KJ, Obayashi T, Nakai K. Protein subcellular localization prediction with WoLF PSORT. In: Proceedings of the 4th annual Asia Pacific bioinformatics conference APBC06, Taipei, Taiwan. vol. 39. Citeseer; 2006. p. 48.

[117] Sonnenburg S, Zien A, Philips P, Rätsch G. POIMs: positional oligomer importance matrices—understanding support vector machine-based signal detectors. Bioinformatics. 2008;24(13):i6–i14.

[118] Briesemeister S, Rahnenführer J, Kohlbacher O. Going from where to why - interpretable prediction of protein subcellular localization. Bioinformatics. 2010;26(9):1232–1238.

[119] van Dijck PWM, Selten G, Hempenius RA. On the safety of a new generation of DSM Aspergillus niger enzyme production strains. Regul Toxicol Pharm. 2003;38(1):27–35.

[120] Dyrløv Bendtsen J, Nielsen H, von Heijne G, Brunak S. Improved prediction of signal peptides: SignalP 3.0. J Mol Biol. 2004;340(4):783–795.

[121] Krogh A, Larsson B, Von Heijne G, Sonnhammer ELL. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J Mol Biol. 2001;305(3):567–580.

[122] Käll L, Krogh A, Sonnhammer ELL. A combined transmembrane topology and signal peptide prediction method. J Mol Biol. 2004;338(5):1027–1036.

[123] Dondoshansky I. Blastclust (NCBI Software Development Toolkit). NCBI, Bethesda, Md. 2002;.

[124] Petersen B, Petersen TN, Andersen P, Nielsen M, Lundegaard C. A generic method for assignment of reliability scores applied to solvent accessibility predictions. BMC Struct Biol. 2009;9(1):51.

[125] Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol. 1999;292(2):195–202.

[126] Ben-Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G. Support vector machines and kernels for computational biology. PLOS Comput Biol. 2008;4(10):e1000173.

[127] Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett. 2006;27(8):861–874.

[128] Taylor WR. The classification of amino acid conservation. J Theor Biol. 1986;119(2):205–218.

[129] Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422.

[130] Jones E, Oliphant T, Peterson P. SciPy: Open source scientific tools for Python; 2001. `http://www.scipy.org/`.

[131] Sonnenburg S, Rätsch G, Henschel S, Widmer C, Behr J, Zien A, et al. The SHOGUN machine learning toolbox. J Mach Learn Res. 2010;99:1799–1802.

[132] Bailey TL, Williams N, Misleh C, Li WW. MEME: discovering and analyzing DNA and protein sequence motifs. Nucleic Acids Res. 2006;34(suppl 2):W369–W373.

[133] Nakashima H, Nishikawa K. Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. J Mol Biol. 1994;238(1):54–61.

[134] Cedano J, Aloy P, Perez-Pons JA, Querol E. Relation between amino acid composition and cellular location of proteins. J Mol Biol. 1997;266(3):594–600.

[135] Rätsch G, Sonnenburg S. Kernel Methods in Computational Biology. The MIT Press; 2004.

[136] Toussaint N, Widmer C, Kohlbacher O, Rätsch G. Exploiting physico-chemical properties in string kernels. BMC Bioinform. 2010;11(Suppl 8):S7.

[137] Akashi H, Gojobori T. Metabolic efficiency and amino acid composition in the proteomes of *Escherichia coli* and *Bacillus subtilis*. Proc Natl Acad Sci USA. 2002;99(6):3695.

[138] Neduva V, Russell RB. Linear motifs: evolutionary interaction switches. FEBS Lett. 2005;579(15):3342–3345.

[139] Hiller NL, Bhattacharjee S, van Ooij C, Liolios K, Harrison T, Lopez-Estraño C, et al. A host-targeting signal in virulence proteins reveals a secretome in malarial infection. Science. 2004;306(5703):1934.

[140] Sagt CMJ, Kleizen B, Verwaal R, De Jong MDM, Muller WH, Smits A, et al. Introduction of an N-glycosylation site increases secretion of heterologous proteins in yeasts. Appl Environ Microbiol. 2000;66(11):4940.

[141] von Heijne G. The signal peptide. J Membrane Biol. 1990;115(3):195–201.

[142] Kowalski JM, Parekh RN, Wittrup KD. Secretion efficiency in Saccharomyces cerevisiae of bovine pancreatic trypsin inhibitor mutants lacking disulfide bonds is correlated with thermodynamic stability. Biochemistry. 1998;37(5):1264–1273.

[143] Kowalski JM, Parekh RN, Mao J, Wittrup KD. Protein folding stability can determine the efficiency of escape from endoplasmic reticulum quality control. J Biol Chem. 1998;273(31):19453.

[144] Whyteside G, Alcocer MJC, Kumita JR, Dobson CM, Lazarou M, Pleass RJ, et al. Native-State Stability Determines the Extent of Degradation Relative to Secretion of Protein Variants from *Pichia pastoris*. PLOS ONE. 2011;6(7):e22692.

[145] Shusta EV, Kieke MC, Parke E, Kranz DM, Wittrup KD. Yeast polypeptide fusion surface display levels predict thermal stability and soluble secretion efficiency1. J Mol Biol. 1999;292(5):949–956.

[146] Kjeldsen T, Ludvigsen S, Diers I, Balschmidt P, Sørensen AR, Kaarsholm NC. Engineering-enhanced protein secretory expression in yeast with application to insulin. J Biol Chem. 2002;277(21):18245–18248.

[147] Ellgaard L, Helenius A. Quality control in the endoplasmic reticulum. Nat Rev Mol Cell Bio. 2003;4(3):181–191.

[148] Helenius A, M A. Intracellular functions of N-linked glycans. Science. 2001;291(5512):2364.

[149] Eriksen SH, Jensen B, Olsen J. Effect of N-linked glycosylation on secretion, activity, and stability of α-amylase from *Aspergillus oryzae*. Curr Microbiol. 1998;37(2):117–122.

[150] van den Brink HJM, Petersen SG, Rahbek-Nielsen H, Hellmuth K, Harboe M. Increased production of chymosin by glycosylation. J Biotechnol. 2006;125(2):304–310.

[151] Liu Y, Nguyen A, Wolfert RL, Zhuo S. Enhancing the secretion of recombinant proteins by engineering N-glycosylation sites. Biotechnol Prog. 2009;25(5):1468–1475.

[152] Kochetov AV. Alternative translation start sites and hidden coding potential of eukaryotic mRNAs. Bioessays. 2008;30(7):683–691.

[153] Strickler SS, Gribenko AV, Gribenko AV, Keiffer TR, Tomlinson J, Reihle T, et al. Protein stability and surface electrostatics: a charged relationship. Biochemistry. 2006;45(9):2761–2766.

[154] Nancy YY, Wagner JR, Laird MR, Melli G, Rey S, Lo R, et al. PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. Bioinformatics. 2010;26(13):1608–1615.

[155] Jensen LJ, Gupta R, Blom N, Devos D, Tamames J, Kesmir C, et al. Prediction of human protein function from post-translational modifications and localization features. J Mol Biol. 2002;319(5):1257–1265.

[156] Cao DS, Xu QS, Liang YZ. propy: a tool to generate various modes of Chou's PseAAC. Bioinformatics. 2013;29(7):960–962.

[157] Gasteiger E, Hoogland C, Gattiker A, Wilkins MR, Appel RD, Bairoch A, et al. Protein identification and analysis tools on the ExPASy server. In: The proteomics protocols handbook. Springer; 2005. p. 571–607.

[158] Shen HB, Chou KC. PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. Anal Biochem. 2008;373(2):386–388.

[159] Rao HB, Zhu F, Yang GB, Li ZR, Chen YZ. Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. Nucleic Acids Res. 2011;39(suppl 2):W385–W390.

[160] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011;12:2825–2830.

[161] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter. 2009;11(1):10–18.

[162] Jain AK, Duin RPW, Mao J. Statistical pattern recognition: a review. IEEE T Pattern Anal. 2000;22(1):4–37.

[163] Duda RO, Hart PE, Stork RG. Pattern classification. Hoboken, NJ: Wiley-Interscience; 2000.

[164] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Berlin, Germany: Springer; 2009.

[165] van den Berg BA, Nijkamp JF, Reinders MJT, Wu L, Pel HJ, Roubos JA, et al. Sequence-based prediction of protein secretion success in *Aspergillus niger*. In: Proceedings of Pattern Recegnition in Bioinformatics 2010. Springer; 2010. p. 3–14.

[166] Schneider G, Wrede P. The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site. Biophys J. 1994;66(2):335–344.

[167] van den Berg BA, Reinders MJT, Hulsman M, Wu L, Pel HJ, Roubos JA, et al. Exploring Sequence Characteristics Related to High-Level Production of Secreted Proteins in *Aspergillus niger*. PLOS ONE. 2012;7(10):e45869.

[168] Ghaemmaghami S, Huh WK, Bower K, Howson RW, Belle A, Dephoure N, et al. Global analysis of protein expression in yeast. Nature. 2003;425(6959):737–741.

[169] Shen HB, Chou KC. A top-down approach to enhance the power of predicting human protein subcellular localization: Hum-mPLoc 2.0. Anal Biochem. 2009;394(2):269–274.

[170] Turanli-Yildiz B, Alkim C, Cakar ZP. Protein engineering methods and applications. In: Kaumaya P, editor. Protein Engineering. Rijeka, Croatia: InTech; 2012. p. 33–58.

[171] Hellinga HW. Rational protein design: combining theory and experiment. Proc Natl Acad Sci USA. 1997;94(19):10015–10017.

[172] Damborsky J, Brezovsky J. Computational tools for designing and engineering enzymes. Curr Opin Chem Biol. 2014;19:8–16.

[173] Wijma HJ, Floor RJ, Jekel PA, Baker D, Marrink SJ, Janssen DB. Computationally designed libraries for rapid enzyme stabilization. Protein Eng Des Sel. 2014;27(2):49–58.

[174] Huang PS, Love JJ, Mayo S. A *de novo* designed protein protein interface. Protein Sci. 2007;16(12):2770–2774.

[175] van den Berg BA, Reinders MJT, Hulsman M, Wu L, Pel HJ, Roubos JA, et al. Exploring sequence characteristics related to high-level production of secreted proteins in *Aspergillus niger*. PLOS ONE. 2012;7(10):e45869.

[176] Jonsson J, Norberg T, Carlsson L, Gustafsson C, Wold S. Quantitative sequence-activity models (QSAM) – tools for sequence design. Nucleic Acids Res. 1993;21(3):733—-739.

[177] Romero PA, Krause A, Arnold FH. Navigating the protein fitness landscape with Gaussian processes. Proc Natl Acad Sci USA. 2013;110(3):E193–E201.

[178] King C, Garza EN, Mazor R, Linehan JL, Pastan I, Pepper M, et al. Removing T-cell epitopes with computational protein design. Proc Natl Acad Sci USA. 2014;p. 201321126.

[179] Zhou P, Tian F, Wu Y, Li Z, Shang Z. Quantitative Sequence-Activity Model (QSAM): applying QSAR Strategy to Model and Predict Bioactivity and Function of Peptides, Proteins and Nucleic Acids. Curr Comput Aided Drug Des. 2008;4(4):311–321.

[180] van Dijck PWM, Selten G, Hempenius RA. On the safety of a new generation of DSM *Aspergillus niger* enzyme production strains. Regul Toxicol and Pharmacol. 2003;38(1):27–35.

[181] Dyrlov Bendtsen J, Nielsen H, von Heijne G, Brunak S. Improved prediction of signal peptides: SignalP 3.0. J Mol Biol. 2004;340(4):783–795.

[182] Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Ω. Mol Syst Biol. 2011;7(1).

[183] Roy A, Kucukural A, Zhang Y. I-TASSER: a unified platform for automated protein structure and function prediction. Nat Protoc. 2010;5(4):725–738.

[184] Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res. 2005;33(7):2302–2309.

[185] Roy A, Xu D, Poisson J, Zhang Y. A Protocol for Computer-Based Protein Structure and Function Prediction. J Vis Exp. 2011;57:e3259.

[186] Schrödinger, L L C . The PyMOL Molecular Graphics System, Version 1.3r1; 2010.

[187] Roubos JA, van Peij NNME. A method for achieving improved polypeptide expression [C12N 15/67 (2006.01)]; 2008.

[188] Bourne Y, Hasper AA, Chahinian H, Juin M, de Graaff LH, Marchot P. *Aspergillus niger* protein EstA defines a new class of fungal esterases within the α/β hydrolase fold superfamily of proteins. Structure. 2004;12(4):677–687.

[189] Ng PC, Henikoff S. SIFT: Predicting amino acid changes that affect protein function. Nucleic Acids Res. 2003;31(13):3812–3814.

[190] Clifford RJ, Edmonson MN, Nguyen C, Buetow KH. Large-scale analysis of non-synonymous coding re-

gion single nucleotide polymorphisms. Bioinformatics. 2004;20(7):1006–1014.

[191] Stone EA, Sidow A. Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity. Genome Res. 2005;15(7):978–986.

[192] Yue P, Li Z, Moult J. Loss of protein structure stability as a major causative factor in monogenic disease. J Mol Biol. 2005;353(2):459–473.

[193] Capriotti E, Calabrese R, Casadio R. Predicting the insurgence of human genetic diseases associated to single point protein mutations with support vector machines and evolutionary information. Bioinformatics. 2006;22(22):2729–2734.

[194] Bromberg Y, Rost B. SNAP: predict effect of non-synonymous polymorphisms on function. Nucleic Acids Res. 2007;35(11):3823–3835.

[195] Calabrese R, Capriotti E, Fariselli P, Martelli PL, Casadio R. Functional annotations improve the predictive score of human disease-related mutations in proteins. Hum Mutat. 2009;30(8):1237–1244.

[196] Li B, Krishnan VG, Mort ME, Xin F, Kamati KK, Cooper DN, et al. Automated inference of molecular mechanisms of disease from amino acid substitutions. Bioinformatics. 2009;25(21):2744–2750.

[197] Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, et al. A method and server for predicting damaging missense mutations. Nat Methods. 2010;7(4):248–249.

[198] Schwarz JM, Rödelsperger C, Schuelke M, Seelow D. MutationTaster evaluates disease-causing potential of sequence alterations. Nat Methods. 2010;7(8):575–576.

[199] Reva B, Antipin Y, Sander C. Predicting the functional impact of protein mutations: application to cancer genomics. Nucleic Acids Res. 2011;39(17):e118–e118.

[200] González-Pérez A, López-Bigas N. Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel. The American Journal of Human Genetics. 2011;88(4):440–449.

[201] Lopes MC, Joyce C, Ritchie GR, John SL, Cunningham F, Asimit J, et al. A combined functional annotation score for non-synonymous variants. Human Heredity. 2012;73(1):47–51.

[202] Olatubosun A, Väliaho J, Härkönen J, Thusberg J, Vihinen M. PON-P: Integrated predictor for pathogenicity of missense variants. Hum Mutat. 2012;33(8):1166–1174.

[203] Li MX, Kwan JS, Bao SY, Yang W, Ho SL, Song YQ, et al. Predicting Mendelian disease-causing non-synonymous single nucleotide variants in exome sequencing studies. PLOS Genetics. 2013;9(1):e1003143.

[204] Shihab HA, Gough J, Cooper DN, Stenson PD, Barker GL, Edwards KJ, et al. Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Hum Mutat. 2013;34(1):57–65.

[205] DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet. 2011;43(5):491–498.

[206] Thusberg J, Olatubosun A, Vihinen M. Performance of mutation pathogenicity prediction methods on missense variants. Hum Mutat. 2011;32(4):358–368.

[207] Kircher M, Witten DM, Jain P, O'Roak BJ, Cooper GM, Shendure J. A general framework for estimating the relative pathogenicity of human genetic variants. Nat Genet. 2014;46(3):310–315.

[208] Yates CM, Sternberg MJ. Proteins and domains vary in their tolerance of non-synonymous single nucleotide polymorphisms (nsSNPs). J Mol Biol. 2013;.

[209] Consortium U, et al. Update on activities at the Universal Protein Resource (UniProt) in 2013. Nucleic Acids Res. 2013;41(D1):D43–D47.

[210] Care MA, Needham CJ, Bulpitt AJ, Westhead DR. Deleterious SNP prediction: be mindful of your training data! Bioinformatics. 2007;23(6):664–672.

[211] Zimmermann K, Gibrat JF. Amino acid "little Big Bang": Representing amino acid substitution matrices as dot products of Euclidian vectors. BMC Bioinform. 2010;11(1):4.

[212] Petersen MTN, Jonson PH, Petersen SB. Amino acid neighbours and detailed conformational analysis of cysteines in proteins. Protein Eng. 1999;12(7):535–548.

[213] Remmert M, Biegert A, Hauser A, Söding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods. 2011;9(2):173–175.

[214] Punta M, Coggill PC, Eberhardt RY, Mistry J, Tate J, Boursnell C, et al. The Pfam protein families database. Nucleic Acids Res. 2012;40(D1):D290–D301.

SUMMARY

The development of high-throughput measurement techniques resulted in rapidly increasing amounts of biological data, which made computational methods essential for biological research. Hence, the field of bioinformatics emerged that since plays an important role in storing, making accessible, integrating, and analysing different types of biological data. Recently, the use of computational methods to (re)design biological molecules is emerging. This thesis describes an approach to determine which protein features influence production and secretion under industrial conditions, and how these results are used to redesign proteins in order to enhance their production levels.

The starting point of this thesis work was a set of measured extracellular concentrations of proteins that were produced under the condition of over-expression in *Aspergillus niger*, a filamentous fungus that is often used for industrial protein production because of its excellent secretion ability. Using this data, classifiers where developed that can predict successful protein over-production based on its sequence. In practice, these classifiers can be used to select proteins that have potential for industrial production.

Subsequent classifier analysis was used to infer what sequence-properties relate to low and high production levels respectively. While taking into account a large set of sequence-derived characteristics, the amino acid composition, i.e. the relative occurrence of the twenty different amino acids in a protein, was found to be best discriminating. Classifier analysis showed the importance of tyrosines for high production levels, whereas lysines and methionines were found to occur relatively often in proteins with low production levels. A similar classifier-analysis method was also used to find characteristic properties of the sequence surrounding neutral and disease-associated missense mutations in humans.

One of the developed classifiers was used in a protein redesign approach to select amino acid substitutions that are expected to have a positive effect on a protein's production level, while additional objectives and restrictions were used to reduce the probability that the amino acid substitution will affect the protein structure. Application of this method resulted in up to ten-fold extracellular concentrations for two *A. niger* enzymes. These results show that this approach is a promising new tool to improve production levels of industrial enzymes. Furthermore, additional research to the redesigned proteins might help to better understand the mechanisms involved in protein production and secretion in *A. niger*.

SAMENVATTING

De ontwikkeling van nieuwe meettechnieken hebben tot een sterke toename van biologische data geleid, waardoor computationele methoden essentieel geworden zijn voor biologisch onderzoek. Daaruit is het onderzoeksgebied bioinformatica ontstaan, dat sindsdien een belangrijke rol speelt bij het opslaan, toegankelijk maken, integreren, en analyseren van verschillende typen data. Tegenwoordig is het gebruik van computationele methoden voor het (her)ontwerpen van biologische moleculen in opkomst. Dit proefschrift beschrijft een methode om eiwiteigenschappen te vinden die van invloed zijn op de productie en secretie van eiwitten die worden geproduceerd onder industriële condities, en beschrijft eveneens hoe deze resultaten zijn gebruikt bij het herontwerpen van eiwitten om hun productieopbrengst te verbeteren.

Het uitgangspunt van dit onderzoek was een dataset van metingen naar extracellulaire concentraties van eiwitten die zijn geproduceerd onder condities waarin overproductie plaatsvindt in *Aspergillus niger*, een filamenteuze schimmel die veel wordt gebruikt voor industriële eiwitproductie vanwege zijn uitzonderlijke secretievermogen. Met behulp van deze data zijn classificatoren ontwikkeld die successvolle overproductie voorspellen op basis van eiwitsequenties. Deze kunnen in de praktijk worden toegepast voor het selecteren van eiwitten die potentieel geschikt zijn voor industriële productie.

Vervolgens zijn deze classificatoren geanalyseerd om te leren welke eigenschappen karakteristiek zijn voor lage danwel hoge productieniveaus. Uit een grote hoeveelheid sequentie-gebaseerde eigenschappen is de aminozuurcompositie, d.w.z. het relatieve voorkomen van elk van de twintig aminozuren in een eiwit, als best onderscheidende eigenschap gevonden. Verdere analyse heeft laten zien dat tyrosines belangrijk zijn voor hoge productieniveaus, terwijl lysines en methionines relatief veel voorkomen in eiwitten met lage productielevels. Eenzelfde methode is ook toegepast om karakteristieke eigenschappen te vinden voor de sequentieomgeving van neutrale en ziektegeassocieerde missense mutaties in mensen.

Een van de ontwikkelde classificatoren is toegepast in een eiwit-herontwerpmethode voor het selecteren van aminozuursubstituties waarvan wordt verwacht dat deze een positief effect hebben op het productieniveau van het eiwit, terwijl tevens extra ontwerpdoelen en restricties zijn ingezet om de kans te verkleinen dat de gekozen aminozuursubstituties de eiwitstructuur aantasten. Het toepassen van deze methode resulteerde in tienmaal hogere extracullulaire concentraties voor twee *A. niger* enzymen. Hiermee introduceert dit proefschrift een nieuwe methode om productieniveaus van industriële enzymen te verbeteren. Daarnaast kunnen de metingen naar de herontworpen eiwitten in vervolgonderzoek leiden tot een beter inzicht in welke mechanismen een rol spelen in eiwit productie en secretie in *A. niger*.

ACKNOWLEDGEMENTS

De totstandkoming van dit proefschrift was niet gelukt zonder de hulp van vele anderen, die ik bij deze graag wil bedanken.

Ten eerste wil ik mijn promotors Marcel Reinders en Dick de Ridder bedanken. Zonder jullie bijdrage was dit resultaat er niet geweest. Jullie expertise, drive en professionaliteit zijn ongekend. Vaak heb ik mij erover verbaasd dat er 's ochtends twee antwoorden in mijn mailbox zaten, terwijl ik pas 's avonds laat een email de deur uit had gedaan. En dan was dat stuk tekst al helemaal doorgenomen, had Dick precies die zinnen waar ik een eeuwigheid aan had zitten sleutelen zo omgevormd dat ze eindelijk wel lekker liepen, en had Marcel maar een paar opmerkingen nodig om precies te benoemen wat er nog aan schort. Daarnaast wil ik Hans Roubos bedanken voor zijn bijdragen als mijn begeleider bij DSM. Je hebt mij de kans gegeven een kijkje te nemen in de toepassingen achter dit onderzoek en daarnaast heb je een belangrijke rol gespeeld in deze samenwerking tussen universiteit en bedrijfsleven. Ook wil ik Saskia en Robbert bedanken voor alle hulp, zonder jullie was alles nooit zo soepel verlopen.

Natuurlijk wil ik ook alle collega's bedanken die mij naast de vele discussies hebben vergezeld bij veel koffie en bier. In Delft, maar ook op de stranden van LA, in de bossen bij Lunteren, bij het kijken naar de Nederland-Spanje finale in Boston, bij pubquizes in Noordwijkerhout, enzovoorts. Ik heb het altijd heel erg naar mijn zin gehad met jullie. Bedankt TU Delft'ers Jurgen, Marc, Alexey, Ahmed, Erdogan, Sepideh, Marcel vb B, Wynand, Thies, Sjoerd, Erik, Chris, Wouter, Ewald, Shipu, Jan, Maarten, Dick, Marcel, Lodewyk, Jeroen, Emile, Emrah, Domenico, en DSM'ers Hans, Jan-Metske, Prescilla, Hilly, Marco, Renger, Liang, Wilbert en Pjotr.

I really enjoyed my three month visit at the EBI in Hinxton, England (despite the freezing cold). I would like to thank Janet Thornton and Tjaart de Beer for the wonderful collaboration, and also all colleagues for introducing me to the great English pub experience. Thanks Tjaart, Sergio, Matthias, Susanna, Dobril, Nidhi, Nick, Roman, and Syed.

Deze jaren had ik ook niet kunnen volbrengen zonder de steun van familie en vrienden.

Ten eerste wil ik diegene bedanken die er altijd voor me was om me door de vele dalen die een promotietrajekt rijk is te helpen, en die mij er op is blijven wijzen dat ik trots mag zijn op de mooie resultaten. Wenda, jouw steun is voor mij onmisbaar.

Ook een speciaal woord van dank aan mijn ouders, Aad en Ans, jullie zijn altijd een enorme steun voor me geweest. Daarnaast wil ik de rest van de (schoon)familie bedanken voor alle steun. Opa en Oma Duyndam, Susanne en Dico, Evelien en Jacco, Lennard en Merel, Marijn, en niet te vergeten de kinderen Iddo, Evi, Boas, Jara, en Lois.

Opa en Oma den Hollander, Oma Koelé, René en Heleen, Linda en Martin, Lara en Dennis, en de kinderen Thijmen, Yenthe en Florian. Bij jullie voel ik me altijd thuis en jullie geven me het besef dat er zoveel meer is dan de wetenschappelijke wereld.

Uiteraard wil ik ook alle vrienden bedanken voor de mentale steun. Twee groepen wil ik er graag uitlichten. Ten eerste de korfballers, tegenwoordig beter bekend als de Bleiswijkbende (ongelofelijk, de invloed van die niet-korfballers), bedankt Thomas en Marianne, Lennard en Merel, Joost en Angela, Nico en Romi. Ook al zien we elkaar niet zo vaak, waar het ook is, wanneer het ook is, het is altijd een feest om met jullie samen te zijn. En ten tweede al die nieuwe vrienden die ik cadeau kreeg bij mijn vrouw: de Doedo's en alle overige aanhangsels. Jullie hebben mij geleerd dat Leiden ook best leuk is en hebben mij de afgelopen jaren de broodnodige afleiding gegeven, waarvoor dank.

En bedankt Puk & Max natuurlijk! Zij die mij altijd weer aan het lachen maakten, de beste ontspanning in stressvolle tijden, hoe kan ik jullie nou vergeten.

CURRICULUM VITÆ

Bas Adriaan van den Berg was born in Bleiswijk, the Netherlands, on July 15th, 1982. He attended pre-university secondary education (VWO) in Schiebroek, Rotterdam at Melanchthon College, where he graduated in 2000. He moved to Delft where he started his study BSc Industrial Design Engineering at the Delft University of Technology. In 2004 he decided to switch to the Computer Science BSc programme at the same university, for which he obtained his degree *cum laude* in 2007. During his Bioinformatics MSc in the Pattern Recognition and Bioinformatics group, he participated in the International Genetically Engineered Machine (IGEM) 2008 competition as part of a team of six students that represented the Delf University of Technology. The team was awarded the best website prize and obtained a gold medal for high-quality contribution. In 2009 he obtained his MSc degree after which he started his PhD project in the same group. The project was part of the BioRange programme of the Netherlands Bioinformatics Center (NBIC) and was performed in collaboration with DSM, under the supervision of Dick de Ridder and Marcel Reinders at the Delft University of Technology, and Hans Roubos at DSM. During his PhD he was a visiting researcher at the European Bioinformatics Institute (EBI) in Hinxton, UK, for three months, where he worked in the group of Janet Thornton under the supervision of Tjaart de Beer. Since March 2014, Bas is working as senior software developer at Biomax Informatics AG in Munich, Germany.

List of Publications

4. BA van den Berg, MJT Reinders, J-M van der Laan, JA Roubos, D de Ridder, *Protein redesign by learning from data*, Protein Engineering Design & Selection (PEDS), **27** (9): 281 – 288, 2014.

3. BA van den Berg, MJT Reinders, JA Roubos, D de Ridder, *SPiCE: a web-based tool for sequence-based protein classification and exploration*, BMC Bioinformatics, **15** (1): 93, 2014

2. BA van den Berg, MJT Reinders, M Hulsman, L Wu, HJ Pel, JA Roubos, D de Ridder, *Exploring Sequence Characteristics Related to High-Level Production of Secreted Proteins in Aspergillus niger*, PLOS ONE, **7** (10): e45869, 2012.

1. BA van den Berg, JF Nijkamp, MJT Reinders, L Wu, HJ Pel, JA Roubos, D de Ridder, *Sequence-based prediction of protein secretion success in Aspergillus niger*, Pattern Recognition in Bioinformatics *5th IAPR International Conference, PRIB 2010, Nijmegen, The Netherlands, September 22-24, 2010. Proceedings*, (6282), 3 – 14, 2010.