Internal Report CS Bioinformatics Track 13-03 June 2013

Leiden University

Master Computer Science

A Voyage Through Protein Sequence Space

ing. Youri Hoogstrate

MASTER'S THESIS

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

Contents

Abstract

The search for novel proteins is complicated by the diversity of amino acids, the non-linear relationship between amino acid sequence and folding and the huge size of sequence space. The goal of this research is to discover patterns in amino acid composition and sequence that relate to known or novel folds, thereby helping to constrain the search space for new functional proteins. This approach is referred to as sequenomics. First, in the construction phase, we demonstrate how sequence space can be meaningfully described and what kind of algorithms and sequence characteristics can be taken into account. Second, in the analytic phase, we describe characteristics of natural and artificial sequences in the sequence space. We find that a coordinate system based on substitution rates can reveal hidden patterns for α-helices and β-sheets in biological sequences. Furthermore, multiple local variance measurements suggest that biological sequences have a complex information signature that distinguishes them from randomly generated sequences. These observations pose new hypotheses about the protein sequence-structure-function relationship that can help to focus the search for new functional proteins.

Keywords: Sequence Space, Maximally Distant Sequence, Multi Dimensional Scaling, Local Variance, Discrete Fourier Transform

1 Introduction

Amino acids are small molecules, consisting of three groups: the amine (NH_2), the carboxylic acid (-COOH) and the functional group which is also called the side chain. In most forms of life there are twenty different amino acids, although some rare exceptional amino acids exist as well. The only thing that differs between these amino acids are the functional groups. The amino and carboxylic acid group are able to form covalent bonds with each other, these connections allow them to form a chain.
Proteins are polymer molecules that consist of a chain of amino acids. A description of what amino acid is located at which position is called the protein *sequence*. Proteins can fulfil many different functions in living cells, like for example: giving cells structure, degrading toxins and cell replication. The number of functions that one protein fulfils is usually limited; it is the combination of and interaction between thousands of proteins that control a cell.
A proteins function is related to its structure which in turn is determined by its sequence. The number of different protein sequences as they appear in nature, the biological sequences, is limited compared to the number of possible sequences. They are just a fraction of the entire *sequence space* (definition 2 and 1).

Definition 1. *Biological sequences are the subset of sequence space that are found in nature.*
In contrast, artificial sequences are the subset of sequence space that are not found in nature.

Definition 2. *The protein sequence space is the collection of all theoretical possible protein sequences.*

Amino acids are able to form bonds beyond the backbone bonds as well. As result, proteins are folded in particular 3D orientations called 3D structures. The observed structures in proteins consist of a limited number of folding motifs, called folds. The *Structural Classification Of Proteins*, or SCOP, has defined 7 structural classes and 3902 different folds (release 1.75; February 23, 2009) [25].

In brief, in nature only a fraction of sequence space has been found, producing an even more limited number of folds or structures.
Concerned with the question whether or not beyond biological sequences novel structures, folds or functions might exist, are questions related to *sequenomics* (definition 3).

Definition 3. *The field of studying sequence space is called sequenomics.*

The words sequence space and sequenomics are not restricted to protein sequences only. Previous sequenomics studies have proven that sequenomics research in nucleic acid sequences have been successful [30, 31, 33, 34, 46].

Biological proteins have scientific potential in biomedical and biochemical applications; for instance as drugs or catalysts. Searching for novel proteins could be simplified by narrowing down the search space. This is where the sequenomics analysis comes in. Using a sequence space, characteristics of sequences can be pointed out and these features could potentially be used to narrow down the search space. The goal of this project is to create a sequence space that consists of biological as well as artificial sequences, which function as observations to apply statistical methods on. The strength of sequenomics research will be illustrated by visualizing the found characteristics.

1.1 Sequence Space

A trivial calculation tells us that the number of possible sequences is $20^{300} = 2.037 \cdot 10^{390}$, if only the average length in humans of $\sim$ 300 amino acids is taken into account. According to currently available computational resources it is important to realize that the entire sequence space is too large for bulk analysis. It says that it is not even possible to construct the entire sequence space. Instead, because of this enormous size, it is necessary to find landmarks or directions in the sequence space that carry information that answers biological questions. The creation of the landmarks in a sequence space could be compared with designing an atlas. An atlas does not contain every possible paving stone. Its resolution is attuned to the demand. For travelling large distances, the resolution of corresponding maps is lower and for travelling short distances, higher resolution maps are available. For sequence space a similar construction would be convenient; to have those sequences that are important to answer a specific question, with a resolution that meets the demands. In brief, the most important questions for creating a sequence space is: which landmarks at which resolution can answer a biological question? Individual maps of sequences are referred to as datasets of which the description is given in definition 4.

Definition 4. *A dataset is a set of protein sequences, either biological or artificial, that reflects a certain direction or location in sequence space. The sequences should have some relationship with each other.*

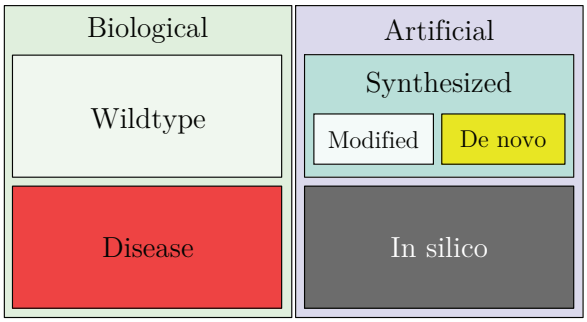


Figure 1: A schematic representation of the nomenclature of types of sequences based on their origin.

2 Methods: Construction

Section overview In this section it will be explained what classes of sequences there are and how they can be used as a dataset to construct a varied sequence space. The first treated datasets belong to biological sequences and the following datasets belong to the artificial. Additionally, some of the sections will illustrate what properties can be controlled in order to preserve biological characteristics. The last given method will focus on the construction of a trajectory of sequences that have an increased evolutionary distance with respect to biological sequences. All these chapters together form the methodology of the construction of a sequence space. An overview of the datasets which recapitulates this chapter briefly, is given at the end of the section in table 1.

2.1 Nomenclature

To avoid misunderstandings between different terminologies, a proposed nomenclature of the types of sequences has been given in figure 1. The major incentive for the nomenclature is the origin of a sequence, because that defines the context of a dataset.
The construction of sequence space is concerned with primarily two types of sequences, the biological and the artificial. The biological sequences, as they appear in

nature, also include malfunctioning or disease causing proteins. For this reason the biological sequences are classified into two subclasses; the wild-type and the disease causing sequences.
Artificial sequences have been created with some technical human influence. They often originate from other sequences, with the purpose to enhance a reaction or prefer different chemical environmental conditions for instance. However, they are not restricted to sequences that correspond to synthesized proteins. Sequences with some theoretical meaning or predicted property can also carry information, although the corresponding protein has never been actually synthesized. Hence, the artificial sequences are partitioned into two subclasses: the *synthesized-* and *in silico* artificial proteins.
Since synthesized artificial sequences are often designed with the purpose to enhance existing proteins, they can have a different meaning than proteins created from scratch. Synthesized artificial proteins are separated in two branches: sequences derived from other sequences, referred to as *modified* sequences, and those that are generated from scratch, referred to as *de novo* sequences.

2.2 Biological Sequences

The space of biological sequences has an enormous diversity and is classified in various ways. Common strategies for classification are:

- By function, like the enzyme commission (EC) numbering [2].
- By sequence, like PFAM [28].
- By structure and folds, like SCOP [25].

Because the SCOP classification is based protein structure, it was used for the construction of the biological datasets. The following SCOP classes have been implemented as a dataset:

A α-helix only proteins.

B β-sheet only proteins.

C α/β proteins.
Mainly parallel β-sheets (β-α-β units).

D $\alpha + \beta$ proteins.
Mainly anti-parallel β-sheets (segregated α and β units).

E Multi-domain proteins ($\alpha + \beta$).
Folds consisting of two or more domains that belong to a different class.

F Surface proteins and peptides.
Membrane and cell surface proteins are (partially) located inside the hydrophobic cell membrane [44].

G Small proteins.

Two types of sequences are not considered to be a separate class according to SCOP. These are the *fibrous-* and the *intrinsically unstructured* proteins. On the top of the SCOP classes the following datasets have been added:

I Fibrous proteins.
They are bar or wire shaped proteins, often giving structure to cells [26]. Because no corresponding maintained public database was found, the dataset was constructed of one single sequence. It comes from the protein Collagen and its sequence is given in supplementary section 7.1. Collagen forms a triple helix [4] which is a repetitive structure that interconnects every three amino acids. Its structure is illustrated in figure 2.

II Intrinsically unstructured proteins.
Intrinsically unstructured proteins (IUP) have no clear classified structure and look like a set of many undirected coils. Still, many of these proteins are found to be functional [41]. The *DisProt* database [36] is a public

database that contains annotations of IUPs. The content is referred to as the intrinsically unstructured protein dataset. It comes with a list of sequences and an additional annotation per sequence describing which regions (sub-sequences) are unstructured. It must be mentioned that for some sequences the annotation is poor; sometimes no annotation is given, sometimes multiple, contradictory, annotations exist.

III IUP: merged.
The DisProt database contains protein sequences and annotations that indicate what regions are actually unstructured. In order to perform analysis on unstructured proteins, it would be most useful to only have the unstructured regions and to take away the structured. For this reason a merged sequences of all unstructured regions was created as follows:

- From all DisProt sequences, the unstructured annotated regions have been taken.
- For only these regions, the composition is taken into account.

All these compositions together have been merged into one composition which is further referred to as the *merged IUP regions*. Notice that this not an actual sequence but only a description the composition of the unstructured regions.

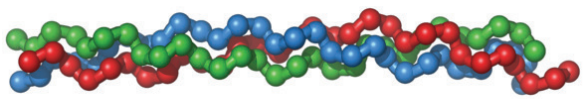


Figure 2: A schematic representation of the triple helix structure of Collagen. Source: `http://en.wikipedia.org/wiki/Collagen_helix`

2.3 Artificial Sequences

In biotechnology, novel proteins are created in order to obtain new or enhance existing functionality. The construction process is different from evolution, which could lead to sequences with different characteristics than the biological. From a sequenomics point of view it is an interesting question whether these sequences are in some way different from the biological, and what these differences are. Another sequenomics related question is why human intervention did, and evolution did not find those enhanced proteins. Is this because the function or enhancement is not essential for survival, evolution simply did not reach this point (yet), or is it impossible to reach it by evolution anyway?

2.3.1 Synthesized Proteins

In laboratories artificial sequences are often synthesized into real proteins. A common example is the enhanced, commercially available, luciferase protein. The group of synthesized proteins is diverse and includes for instance proteins with a medicinal background, like recombinant human, bacterial or viral proteins that function as drug components [40]. Also artificially constructed antibodies [5] or proteins using amino acids beyond the 20 found in human life [15] belong to this group.
In the nomenclature a distinction was made between modified and de novo synthesized sequences. However, it is not easy to draw the boundary between them, as the following example will illustrate. In previous research a method was designed where existing sequences have been used as a template. These sequences carry certain desired properties (stability, solubility) but lack an actual function. These recombinant sequences are continuously randomly mutated at specific regions, until a novel functional protein with the desired properties is discovered [40]. This method uses a evolution-

ary process as well as technical pre-selection by humans. To avoid confusion, synthesized proteins are classified de novo, if no clear reference sequence can be defined.
In previous research, the construction of de novo artificial proteins from a library of $6 \cdot 10^{12}$ randomly generated sequences, indicated that functional proteins can be created by chance [18]. They found 79 sequences, selected for ATP binding affinity, which are functional and unrelated to other sequences. These sequences are further referred to as the *artificial synthesized de novo* dataset.

2.3.2 In Silico Sequences

Entropy trajectories In previous research a method for sampling a compositionally widespread view of sequence space has been proposed [46]. It creates sequences that vary in their uniformity of amino acid composition. This property, called entropy, is further explained in section 3.1.5. Their proposed construction protocol is as follows:

- Choose a desired sequence length: 40, 80 160 and 320 have been chosen.
- Create for every amino acid a sequence of the chosen length, consisting only of this amino acid. Each of these 20 sequences is called a homo-polymer and will initialize an amino acid specific direction in sequence space.
- For every one of these 20 directions, the sequence is systematically mutated with one single amino acid per time. The amino acid is chosen such that the frequency of amino acids of the sequence becomes more uniform. This process is repeated until equal frequencies (maximal entropy) are obtained.

Because every amino acid forms a trajectory over entropy, the dataset is referred to as the *entropy trajectory* dataset.

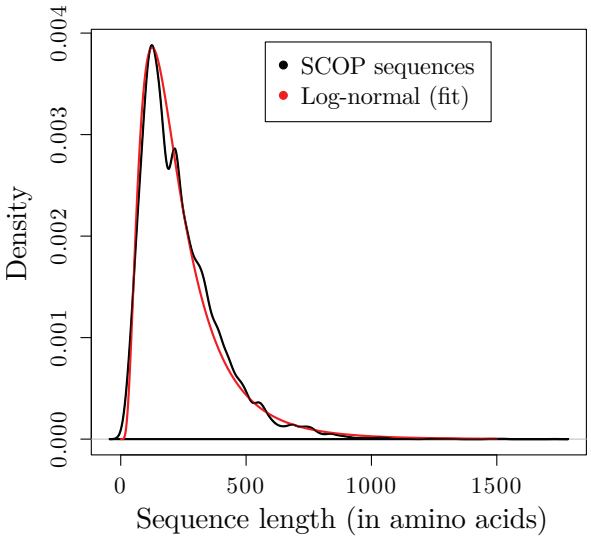


Figure 3: For the biological sequences from the SCOP database, the lengths of the sequences have been used to fit a log-normal distribution using the MASS package in R. The corresponding distributions are illustrated above, where the actual SCOP lengths are indicated black and the fitted log-normal distribution red.

Sequence parameters For randomly sampling sequences it is convenient to use distributions that resemble biological property distributions. The distribution of sequence lengths of the SCOP sequences has been fitted to a log-normal distribution, given in figure 3. The fit has the corresponding parameters: $\mu = 194.42$ and $\sigma = 1.93$ Previous research supports the finding that sequence lengths are log-normal distributed. [47] For constructing sequences, 25.000 lengths have been randomly sampled using the log-normal fitted distribution.
With these sampled lengths two datasets have been constructed. Thus, both datasets consist of 25.000 sequences, sampled using compositions based on:

- Uniform amino acid frequencies ($\frac{1}{20}$).
- Observed frequencies in biological sequences (from SCOP, given in figure 6).

The former dataset is referred to as the *artificial in silico* and the latter as the *artificial in silico: biological composition* dataset.

Shuffled biological sequences Additionally, the biological (SCOP) sequences have been shuffled with the purpose to ensure that composition remains constant while the internal sequential order is disrupted. It is further referred to as the *shuffled* dataset.

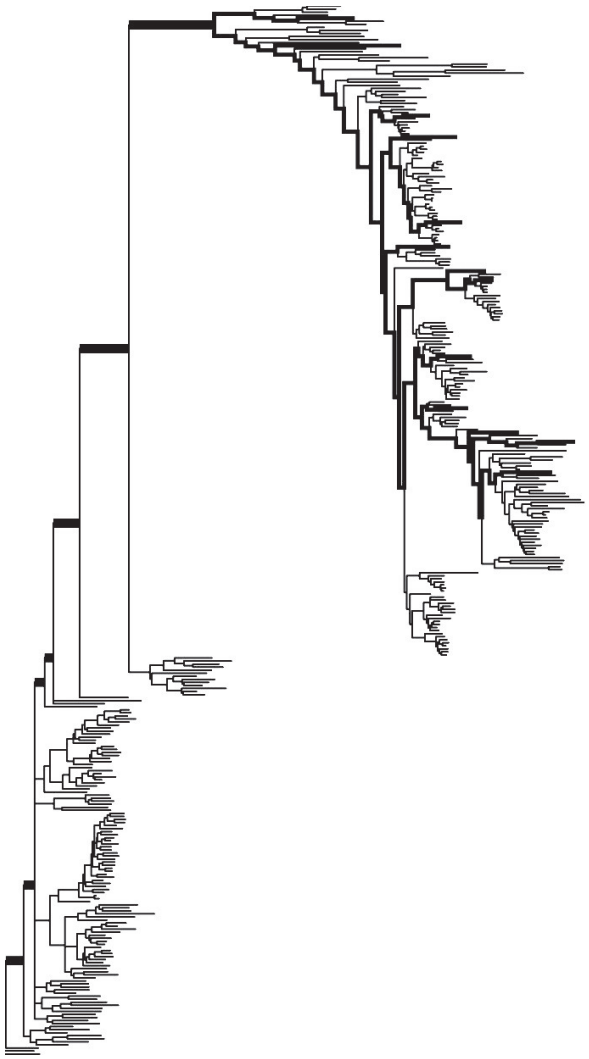


Figure 4: This figure, taken from [8], shows a phylogenetic tree of the ribosomal RNA of different organisms. It illustrates that evolution takes certain directions through (nucleic) sequence space. It could be that once a particular direction has been "chosen", it is unlikely evolve towards another direction because of its surroundings.

2.3.3 Maximally Distant Sequence

To better understand what is special about biological sequences it might be convenient to take a look at sequences that are the furthest away from them. The corresponding concept of the *maximally distant sequence* is given in definition 5.

Definition 5. *Maximally distant sequences have the largest possible evolutionary distance to all biological sequences.*

Evolution has taken certain directions and might, because of this, be restricted to its surroundings. Previous research pointed out that in ribosomal RNA evolution certain directions have been chosen [8]. Their phylogenetic tree is given in in figure 4.
By finding sequences that are the furthest away from any biological sequence, details about how they differ from biological sequences might be found. If possible, insights could be obtained about whether beyond the biological sequences there might be isolated island of sequences; locations in sequence space that cannot be reached by evolution because the limited accessibility of the surroundings. The properties of the sequences that are far away from biological sequences might be useful for understanding what is necessary for sequences to be biologically (ir)relevant.

Heuristic repair method Here an algorithm able to find maximally distant sequences will be proposed. Its goal is to find a trajectory of fixed length sequences, T_i, where i is the trajectory iterator, that become gradually more distant with respect to a given library of biological sequences $\mathbf{L}$. Assuming that $f(\bullet)$ is a fitness function, a trajectory must by satisfy:
$f(T_0) < f(T_1) < \ldots < f(T_i) < \ldots < f(T_n)$.
The proposed algorithm solves the combinatorial search problem by searching for a fit-

ness optimization (maximizing evolutionary distance with respect to its nearest neighbour) which is dependent on many different variables (amino acid homology for all biological sequences). The algorithm is initialized with library $\mathbf{L}$ and an initial sequence x, located at T_0. This sequence is used for step-wise mutations in order to improve fitness. Sequence T_0 can be any sequence; random in silico generated, homo-polymer or an element in $\mathbf{L}$. More precisely, the proposed algorithm is referred to as a *heuristic repair* method because of the following step plan [23, 24]:

1 Estimate fitness of the target.

Biological context: find the evolutionary distance of sequence x to all sequences in $\mathbf{L}$.

2 Repair target based on heuristics.

Biological context: according to the estimated evolutionary distance, mutate sequence x into x' such that it is likely that x' will improve the fitness over x.

3 Estimate fitness of the repaired target. If fitness has improved, use the repaired target instead. Return to step **1**.

Biological context: find the evolutionary distance of x' with all sequences in $\mathbf{L}$. If sequence x' improves fitness; $f(x') > f(x)$: $x \leftarrow x'$ and $T_i \leftarrow x'$ and $i = i + 1$. Return to step **1**.

Evolutionary distance Although definition 5 states that a maximally distant sequence must have the largest evolutionary distance, the explicit definition of evolutionary distance is not given. Evolutionary distance between two arbitrary sequences can be hard to estimate since it describes the relative distance to evolve from one sequence to another, without knowing the intermediate and ancestor sequences. It is usually expressed in relative evolutionary time or in the number of substitutions, where the former also considers the likelihood of individual amino acid substitutions.

There exist different algorithms for finding sequence similarity. The majority of algorithms have the purpose to find homology between sequences. Most of these algorithms, called sequence alignment algorithms, calculate a score for similarity and produce an alignment indicating where precisely the similarities and dissimilarities are located.

Of course, similarity is the opposite of the evolutionary distance. In contrast, minimizing the similarity is an approach which is to some extend similar to maximizing the evolutionary distance. The additional advantage of using sequence alignments is that the alignment indicates where similarities are located. In turn, these locations can be used for targeted heuristic reparations; they can be used to define exactly those amino acids that should be mutated in order to obtain a higher likelihood for obtaining optimization.

Definition 6. *The optimization criterion for evolutionary distance only allows optimization if:*

- *The maximum alignment score of x' with the nearest neighbour in L is be lower than for x with L.*
- *Or, the maximum alignment score of x' with the nearest neighbour in L is identical to x with L, but the corresponding maximal number of possibilities is lower.*

The implemented alignment method which defines the evolutionary distance, is an adapted implementation of the original end-space free alignment [35], extended with the ability to find the number of optimal solutions. To maintain the structure of this re-

port, supplementary information about this algorithm is given in section 7.2.
Because of the relevance of the number of optimal solutions on top of the alignment score, the optimization criterion is implemented as defined in definition 6.

Mutation procedure The mutation procedure involves the following three parameters:

- The number of substitutions per iteration.
- The positions of substitutions (relative to the target sequence).
- The choice of amino acids (per position).

The most straightforward approach is mutating the target sequence according to these parameters completely random. Although this would explore the entire search space, convergence is expected to go slower since no heuristics are taken into account. Instead, using the results of the alignments, faster convergence is expected. The proposed mutation procedure goes as follows:

1. Find all sequences in $\mathbf{L}$ that have the smallest evolutionary distance to sequence x and put this subset in $\mathbf{K}$.
2. According to the alignments of x with the sequences in $\mathbf{K}$, create a vector $\mathbf{C}$ that contains position of x that have a match with a sequence in $\mathbf{K}$.
3. Create a second matrix $\mathbf{D}$ that includes only the unique values of $\mathbf{C}$.
4. Estimate the first parameter ζ, the number of substitutions per iterations. This number has been estimated to be optimal for $\zeta = 1$.
5. Estimate the second parameter, the positions of the substitutions. Pick randomly a number of ζ unique items from $\mathbf{C}$. After every draw, remove all elements in $\mathbf{C}$ that have the found value to preserve draws with unique values.
6. Estimate the third parameter, the actual replacements. For every position that has to be substituted, the amino acids that are unlikely to optimize will be discarded. From the remaining amino acids, one will randomly be chosen. This goes as follows:
   - Per mutation-location j, the algorithm starts with a library of all 20 amino acids, minus the original amino acid at x_j.
   - For every alignment i in $\mathbf{K}$, if at the position aligned to x_j a match is found, the corresponding amino acid in sequence K_i is removed from the library.
   - The eventual substitution will be the replacement of x_j for an amino acid randomly chosen from the library. If the library is empty, it will be re-initiated and a random element will be chosen.

It must be mentioned that the larger the choice of ζ, the larger the distance between the found sequences will be. If a smooth trajectory is desired, where the evolutionary distance between two sequences should be close, ζ should be small.
Notice that using the non-unique matrix $\mathbf{C}$ for the estimation of the locations that have to be mutated, replacements of locations that are found in multiple sequences in $\mathbf{K}$, obtain a higher likelihood.
The mutation process systematically tries to mutate similar amino acids. From this perspective, similar amino acids can be seen as conflicts, since they decrease the evolutionary distance between two sequences. Under the assumption that similar amino acids are conflicts, the algorithm is classified as a min-conflicts heuristic repair method [23, 24].

Optimization Comparing every iteration sequence x with all biological sequences in order to obtain the distances results in a high computational demand. Given that the time complexity of the alignment algorithm is in $O(n \cdot m)$ (where n and m are the lengths of the sequences), the complexity of one iteration finding distances is in $O(n \cdot m \cdot \langle \mathbf{L} \rangle)$ just to find out if a possible novel target sequence indeed improves the fitness function. Here $\langle \bullet \rangle$ is the vectors length operator.

In every iteration the heuristic repair procedure wants to find $\mathbf{K}$, the most similar sequences to x, in order to mutate x into x'. After the first iteration all alignments between x and $\mathbf{L}$ have been calculated. Sequences that now have a large evolutionary distance to x should not be taken into account the next iteration, because only a limited number of mutations take place. For the next iteration, it can be estimated on beforehand which sequences will always have a larger evolutionary distance than x' to $\mathbf{K}$. Therefore they can be left out that iteration and will therefore improve the performance since a number of alignments can be skipped.

Assume that $s(\bullet)$ finds the similarity between two sequences, then $s(x, \mathbf{K})$ has the largest similarity of x to the biological sequences. Every sequence i in $\mathbf{L}$ of which $s(x', L_i) - s(x, \mathbf{K}) > \zeta \cdot \max(\textit{match score})$ can be left out for analysis because their evolutionary distance is large. Here ζ represents the number of substitutions that have taken place between the comparisons of $s(x, \mathbf{K})$ and $s(x', L_i)$.

Notice that if a certain sequence is left out for analysis one iteration, its ζ increases with the number of mutations that have taken place that iteration.

The subset of the $\mathbf{L}$ that are not left out a particular iteration, because they could potentially be the most similar sequence to x, are called *the scope*. Notice that every iteration that the x' has optimized x, the scope changes. The illustration of the entire algorithm, including the scope, is illustrated in figure 5.

Analysis For finding the maximally distant sequence, 1500 SCOP sequences with a length of about 300 amino acids have been used as library of biological sequences. These sequences can be found at the repository, given in section 4.1. The algorithm has been executed using different settings:

- Homogeneous scoring $\times$ biological initiator sequence*
- Homogeneous scoring $\times$ homo-polymer `W` of length 300 as initiator sequence
- Biological scoring $\times$ biological initiator sequence*
- Biological scoring $\times$ homo-polymer `W` of length 300 as initiator sequence

* The following initiator sequence was taken from the SCOP database because of its average length and average composition:

```
GDVQN AVEGA MVRVA DTVQT SATNS ERVPN
LTAVE TGHTS QAVPG DTMQT RHVIN NHVRS
ESTIE NFLAR SACVF YLEYK TGTKE DSNSF
NNWVI TTRRV AQLRR KLEMF TYLRF DMEIT
VVITS SQDQS TSQNQ NAPVL THQIM YVPPG
GPIPV SVDDY SWQTS TNPSI FWTEG NAPAR
MSIPF ISIGN AYSNF YDGWS HFSQA GVYGF
TTLNN MGQLF FRHVN KPNPA AITSV ARIYF
KPKHV RAWVP RPPRL CPYIN STNVN FEPKP
VTEVR TNIITT
```

Because of the large complexity of the algorithm, the programs runtime has been set to maximally 10 days.

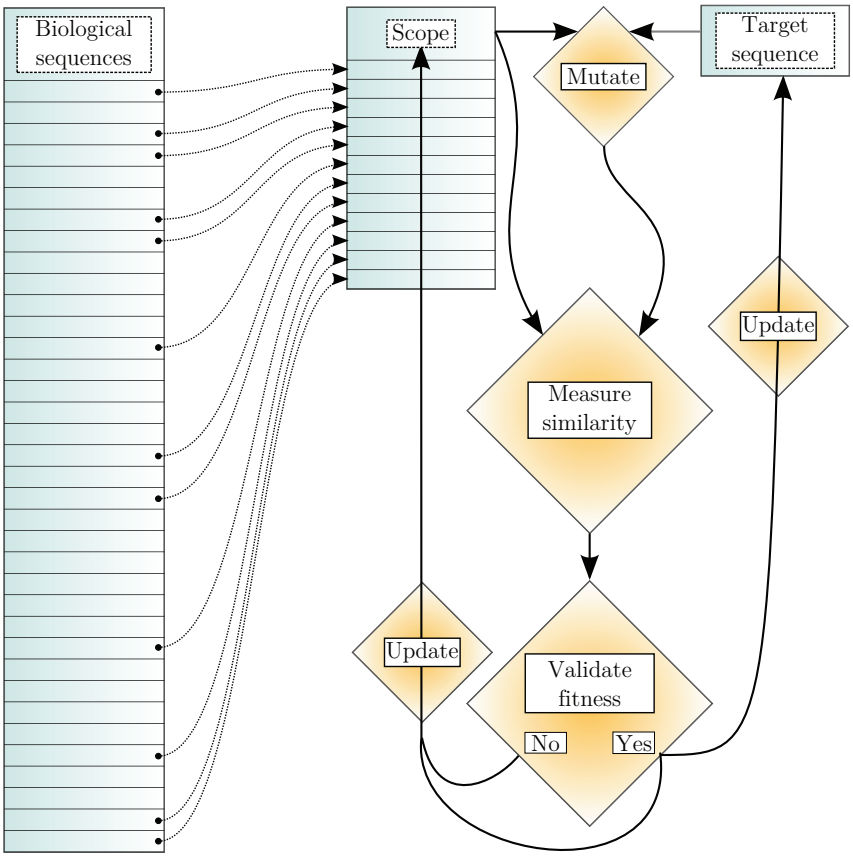


Figure 5: This is the schematic view of the maximally distant sequence algorithm. The variables are given in the boxes, including the target sequence, the library biological of sequences and the scope. The diamonds indicate the processes. The target sequence is mutated and only if the similarity measurement indicates that fitness with respect to the scope has been improved, the sequence is updated.

Dataset	Biological	⟨Sequences⟩
SCOP (unique)	Yes	22709
SCOP Shuffled	No	22709
SCOP class A	Yes	46456
SCOP class B	Yes	48724
SCOP class C	Yes	51349
SCOP class D	Yes	53931
SCOP class E	Yes	56572
SCOP class F	Yes	56835
SCOP class G	Yes	56992
Fibrous proteins	Yes	1
IUP	Yes	684
Merged IUP regions	No	1
Artificial in silico (uniform)	No	25000
Artificial in silico (preserved)	No	25000
Entropy walk length 40	No	76000
Entropy walk length 80	No	152000
Entropy walk length 160	No	304000
Entropy walk length 320	No	608000
Random (uniform amino acid composition)	No	25000
Random (biological amino acid composition)	No	25000
Most distant sequence trajectory (homogeneous)	No	1217
Most distant sequence trajectory (uniform)	No	569

Table 1: The different datasets that form the proposed landmarks in protein sequence space.

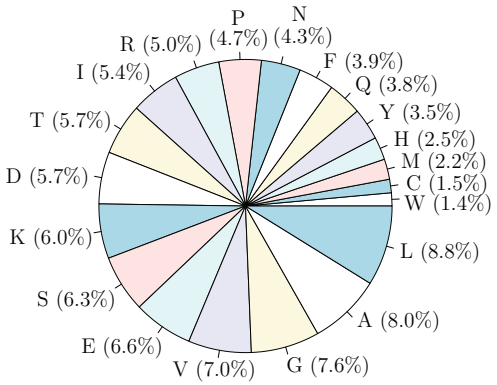


Figure 6: A pie-chart of the average composition of all SCOP sequences. The composition deviates from a uniform (5% per amino acid) distribution.

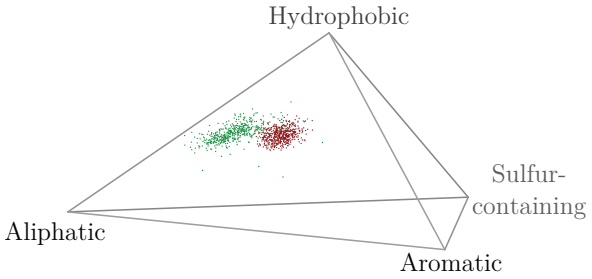


Figure 7: This figure illustrates that using a simplex and mutually exclusive properties, the dimensionality can be reduced by 1. In this simplex the composition for 500 biological (green) and 500 artificial in silico sequences with a uniform amino acid composition (red) have been project.

3 Methods: Data Mining

Section overview The main goal behind this research is to determine characteristics for sequences that correspond to particular classes. There are many, if not endless, characteristics or properties of sequences. The challenge is to find those that are able to make a distinction between classes of sequences. For this research, the properties are classified into two types:

- Global properties, which carry information over the entire sequence without taking sub-sequences into account, like length and composition.
- Local properties, which carry information over sub-sequences or other local regions of a sequence.

This section tries to explain in which way sequences can be examined in order to find specific patterns and characteristics. Additionally, different ways of visualizing sequences and their properties will be explained.

3.1 Global Measurements

One of the most basal properties that a protein sequence has is its composition. This is the relative frequency of each amino acid in a sequence, regardless of the order in which they are found. Previous research pointed out that, in non-homologue nucleic acid sequences, for specific functions certain compositions are preferred, probably due to intrinsic structure constraints [30, 33]. Because of the observation in nucleic acids, similar behaviour can be expected for proteins sequences.

If for protein sequences the composition is examined, it could be projected in a barplot or a pie-chart like figure 6. It shows that the amino acid composition of protein sequence is not uniformly distributed but that there are certain overrepresented and rare amino acids. Because such barplots and pie-charts construct a composition space (where individual compositions are drawn instead of individual sequences), it is difficult to resolve the characteristics of individual sequences. The major problem that makes it hard to create a visualization where individual sequences are drawn, is that the protein sequence alphabet consists of 20 letters, what makes the composition a 20-dimensional space. This makes it impossible to draw protein sequence composition in two or three dimensions [46]. Instead, the composition could be projected in a property space; where the properties of amino acids are used as axes.

There are many different published lists of numerical amino acid properties (544 in AAindex 9.1, March 2013), even more than the size of the protein alphabet squared [17]. On top of that there exist classification schemes used for categorization of amino acids. Despite this enormous number of amino acid properties, the desired number of properties for visualizing sequence composition should be as low as possible with a maximum of three. What is important to realize is that when a composition is drawn in a property space, it will give the contribution per property instead of per amino acid. The consequence is that this might result in an incomplete or simplified view of the entire 20-dimensional amino acid composition. The properties that can make a space with the lowest level of loss of information with respect to the original composition can form the ideal space to draw the sequence composition in.

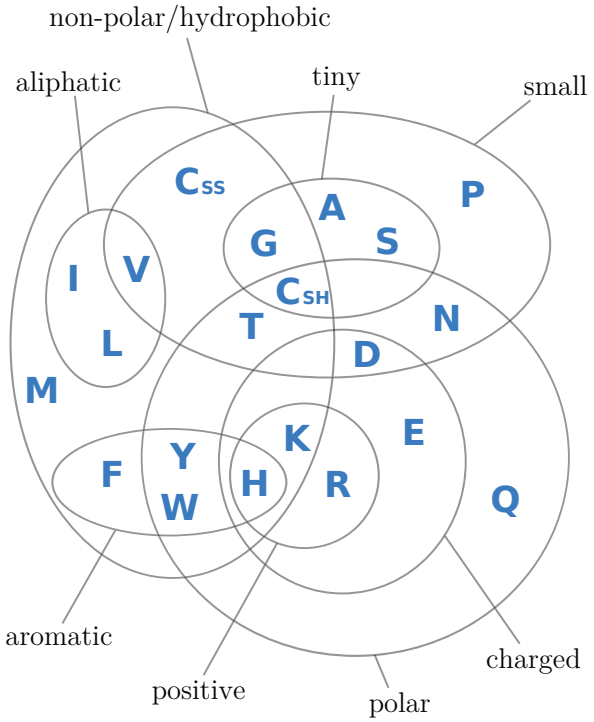


Figure 8: The amino acids clustering based on their most important properties [39]. The figure has been adapted from: `www.embl.de/~seqanal/courses/commonCourseContent/commonProteinStructureFunctionExercises.html`

3.1.1 Amino Acid Classes

A particular type of amino acid properties are the nominal or discrete properties. In previous research the amino acids have been classified by their most important properties, which resulted in the classification scheme illustrated in figure 8 [39]. This scheme shows some classes which are descriptive in a functional or chemical sense. The basis for this classification scheme is the evolutionary distance (substitution frequencies) and the highest conserved physicochemical properties. In the scheme cysteine is exceptional since it may contain two forms of sulphur. According to the type of sulphur the behaviour in cysteine differs such that the amino acid is classified twice.

The major point of the scheme is that there is *redundancy* in the amino acid properties. The main assumption is that *the illustrated amino acid properties in figure 8 are the actual properties that give a protein its structure and accordingly its function. Thus, substitutions for amino acids with similar properties are more likely to preserve structure and function.* It is because of this assumption and the redundancy that a property space might be a strong solution to draw the amino acid composition, since the redundancy suggests that composition can be simplified without losing information.

There is a subset of class-type properties that have a major advantage for visualization purposes. Such discrete properties have to be *mutually exclusive*: if an amino acid falls in one class it excludes to fall in another class. In mathematical terms, the intersection of the classes must be empty. Take for example the hydrophobic amino acids. With the exception of threonine, they can be clustered in the following three classes:

- Aliphatic: `I, L, V`
- Aromatic: `F, H, W, Y`

- Sulfur-containing: C, M

If an amino acid is sulphur-containing, like for example methionine, it excludes that the amino acid is aliphatic or aromatic. This mutually exclusiveness allows to draw n amino acids properties in $n-1$ dimensions. This is achieved by translating two properties into their gradient, the ratio between the properties. The gradient of each property can be drawn as a point in a space. Considering the given example, the four classes hydrophilic, aliphatic aromatic and sulphur-containing can be drawn in a so called tetrahedron as illustrated in figure 7.
This immediately leads to the weakness of discrete and nominal values. If such a value origins from a quantitative value, like mass for example, accuracy will be lost if only the discrete properties *tiny*, *small* and *large* are considered because the average of the mass could have been calculated instead. Because of this reason and the assumption that a property space should lose information about the true composition as few as possible, nominal properties are less suitable for analysing sequence composition. However, this does not mean that nominal properties are not suitable for a sequenomics type of analysis or visualizations.

3.1.2 Numerical Properties

The properties published in AAindex are numerical and have an ordinal or quantitative meaning [17]. They can validly be used for ranking and in certain situations they can also be used for calculations. The domain of ordinal and quantitative amino acid properties consist for the majority of physico-chemical properties.
Physico-chemical means that they must have something to do with either physics or chemistry and must be measurable or observable in some way. Common examples are mass and hydropohibicity. Because certain properties are relatively difficult to measure or can be labelled in multiple ways, redundancy in properties is common. An example are three commonly used schemes for hydrophobicity [12, 14, 20]. Despite the high number of properties in the AAindex, a clustering indicated that there are 6 major branches of properties [17].
Other numerical types of properties can be focussing on the structural level of a protein. They include probabilities to be found in helix, sheet or coil for example.
In comparison with nominal values, averages of quantitative properties have a biological and accurate meaning and are for this reason more suitable for drawing a property space since they lose less information about the action composition than nominal values. However, it is difficult estimate what the exact level of information loss is. This depends on each individual property and on their combinations in a property space.
To get an impression of the properties, the values of mass, electron-ion interaction potential and three scales of hydrophobicity are given in table 2.

3.1.3 Dimension Reduction

There are many different numerical amino acids properties but it is difficult to estimate which are most useful. A possible solution is applying dimension reduction methods like principal component analysis on such data. It searches for those linear directions that have most variance in a multi-dimensional space. However, besides that the data is not guaranteed to be linear nor scaled, the principal component analysis does not give the most biologically important properties but those that have most variance (probably biassed towards those that are studied most intensively). Using such properties with a study related preference is called *preferential attachment* [46]. For this reason super properties by mass scale analysis of AAin-

1-letter	3-letter	Mass	EIIP	Hydroph*[1]	Hydroph*[2]	Hydroph*[3]
A	Ala	71.0788	0.0373	-0.5	1.8	1.6
R	Arg	156.1875	0.0959	3.0	-4.5	-12.3
N	Asn	114.1038	0.0036	0.2	-3.5	-4.8
D	Asp	115.0886	0.1263	3.0	-3.5	-9.2
C	Cys	103.1388	0.0829	-1.0	2.5	2.0
E	Glu	129.1155	0.0058	3.0	-3.5	-8.2
Q	Gln	128.1307	0.0761	0.2	-3.5	-4.1
G	Gly	57.0519	0.0050	0.0	-0.4	1.0
H	His	137.1411	0.0242	-0.5	-3.2	-3.0
I	Ile	113.1594	0.0000	-0.8	4.5	3.1
L	Leu	113.1594	0.0000	-1.8	3.8	2.8
K	Lys	128.1741	0.0371	3.0	-3.9	-8.8
M	Met	131.1926	0.0823	-1.3	1.9	3.4
F	Phe	147.1766	0.0946	-2.5	2.8	3.7
P	Pro	97.1167	0.0198	0.0	-1.6	-0.2
S	Ser	87.0782	0.0829	0.3	-0.8	0.6
T	Thr	101.1051	0.0941	-0.4	-0.7	1.2
W	Trp	186.2132	0.0548	-3.4	-0.9	1.9
Y	Tyr	163.1760	0.0516	-2.3	-1.3	-0.7
V	Val	99.1326	0.0057	-1.5	4.2	2.6

Table 2: This table contains several physico-chemical amino acid properties. [1] Hydrophilicity: Hopp and Woods [14], [2] Hydrophobicity: Kyte and Doolittle [20]., [3] Hydrophobicity: Engleman et al. [12], Electron-ion interaction potential (EIIP) [45]

dex seem to offer no solution for a property space.

Yet on the other hand, the classification scheme in figure 8 shows that properties based on substitutions are expected to be most important for proteins structures and functions. In other words, such properties maintain the most complete view of the amino acid composition.

Because the amino acid classification indicated what properties are biologically important, and numerical properties are technically most convenient, a method that finds a solution for this will be given. A family of techniques that allows to reduce dimensions in data is *multi-dimensional scaling* (MDS). They derive novel coordinates, with a lower number of dimensions, from a matrix of distances, trying to preserve the original distances. For input MDS requires a chosen number of dimensions and a distance matrix of a set of entities. A distance matrix has the following constraints:

- Every diagonal value must be zero.
- The matrix is diagonal symmetrical.
- Every distance is non-negative.
- Every distance is representing a certain amount of dissimilarity.
- Every distance meets the triangle inequality.

Given distance matrix $\boldsymbol{\delta}$ of size $n \times n$ containing distances between entities i to j, the algorithmic problem is to find for each entity a vector where the distance between the entities ($d(\boldsymbol{x}_i - \boldsymbol{x}_j)$) is as close as possible to the original distance $\boldsymbol{\delta}_{i,j}$. Thus, for each i-th entity ($1 \leq i \leq n$) the goal is to find m coordinates $\omega_1, \ldots, \omega_m$, which are put in vector $\mathbf{x}_i$ as follows:

$$\boldsymbol{x}_i = \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_m \end{bmatrix}$$

The problem of MDS is to solve the minimization problem given in equation 1, where $d(\bullet)$ is a distance function, e.g. the Euclidean distance, n the number of objects and m the chosen number of dimensions.

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n} \sum_{i<j} (d(\boldsymbol{x}_i - \boldsymbol{x}_j) - \boldsymbol{\delta}_{i,j})^2 \qquad (1)$$

Important properties The redundancy found in amino acid properties are presumably the constraints that allow certain substitutions more often than others, and, to some extend, guide evolution. Take for example glutamine and asparagine which are about equally hydrophobic. In certain situations a substitution of one amino acids for the other might have (almost) no effect on the eventual protein since the hydrophobicity is preserved. For this reason a substitution of glutamine to aparagine might be more probable than glutamine to valine. Indeed, this would not only hold for hydrophobicity: no matter what property it is, if it is really important for protein functioning, evolution will have ensured that the substitution for amino acids that lack this property occurred as few as possible. To visualise protein sequences in a property space, these evolutionary most important properties are useful because they create a space that does not lose information about the composition in a protein-functional context. For this reason they are the key candidates for a property space.

The substitution matrix Substitution matrices are tables that contain the rate of substitution of amino acids to each other, transformed to a score. They are generally used for calculating the similarity between sequences in alignment algorithms. The input for calculating a substitution matrix is a set of homologue sequences which are aligned using a multiple sequence alignment (MSA) [7]. This step requires, ironically, another substitution matrix. To illustrate the calculation of a substitution matrix, consider the MSA given in table 3. A substitution score is a log-odd ratio between the observed and expected substitution rate of two amino acids [11, 13]. The score $S(a_i, a_j)$ for substituting amino acid a_i for amino acid a_j, is calculated using equation 2 where p_{a_i} and p_{a_j} are the probabilities of finding amino acids a_i and a_j in any sequence, and p_{a_i,a_j} is the observed probability of amino acid i to be substituted for amino acid j. Variable λ is only a scaling factor.

Species	Sequence
Human	`TTNYLIVSLAVADLLVATLVMPWVV`
Mouse	`TTNYLIVSLANADLLVATLVWPWVV`
Carp	`TTNYLIVSLAVQDLQVATLVMPWSV`
Ant	`WTNYLIVSLANQDLLVATLVMPWVV`
Worm	`TTNYLIVSEAVADLQVATLVMPWVV`

Table 3: An example multi sequence alignment of a homologue sequence in $n = 5$ different organisms.

$$S_{a_i,a_j} = \frac{1}{\lambda} log \frac{p_{a_i,a_j}}{p_{a_i} \cdot p_{a_j}} \qquad (2)$$

The process of the estimation of p_{a_i,a_j}, considers each column of the MSA separately since substitutions take place vertically (further illustrated in table 4). The procedure uses a count matrix to count the observed substitutions. Initially the count-matrix C_{a_i,a_j}, of length 20×20, is filled with zeros. For every column the following procedure is applied to obtain the complete count matrix and find the corresponding substitution probabilities:

- *Find the amino acids that belong to the target column.*

  For the first column in the example, substitution of `T`, `T`, `T`, `W` and `T` are considered. Since the ancestral amino acid

Substitutions	Counts	
T: .TTWT	T→T: 3	T→W: 1
T: ..TWT	T→T: 2	T→W: 1
T: ...WT	T→T: 1	T→W: 1
W: ....T		T→W: 1
T:		

	T	W	...	S	V
T	3	1		0	0
W		0		0	0
⋮					
S				0	0
V					0

	T	W	...	S	V
T	6	4		0	0
W		0		0	0
⋮					
S				0	0
V					0

Table 4: The summation matrix is constructed by iteratively walking over the columns of the MSA. **Bottom-left**: in the first column, for the the first amino acid there are $n-1$ possible pairs, where n equals the number of organisms (and rows in the MSA). **Bottom-right**: after the first column is processed there are $n(n-1)/2 = 10$ possible pairs added to the matrix. **Top**: for the first column in the MSA, the matrix is filled with these observed substitution.

is unknown, substitutions are undirected.

- *Find all possible substitution combinations of the column.*

  For column 1, the following combinations are observed:

  - The 1^{st} T could have been substituted by $\{\mathtt{T}, \mathtt{T}, \mathtt{W}, \mathtt{T}\}$.
  - The 2^{nd} T could have been substituted by $\{\mathtt{T}, \mathtt{W}, \mathtt{T}\}$.
  - The 3^{rd} T could have been substituted by $\{\mathtt{W}, \mathtt{T}\}$.
  - The W could have been substituted by $\{\mathtt{T}\}$.

- *Estimate the corresponding number of substitution counts.*

  For the first column the following counts are observed:

  - $3 \cdot [\mathtt{T} \to \mathtt{T}]$ $\quad$ $1 \cdot [\mathtt{T} \to \mathtt{W}]$
  - $2 \cdot [\mathtt{T} \to \mathtt{T}]$ $\quad$ $1 \cdot [\mathtt{T} \to \mathtt{W}]$
  - $1 \cdot [\mathtt{T} \to \mathtt{T}]$ $\quad$ $1 \cdot [\mathtt{T} \to \mathtt{W}]$
  - $1 \cdot [\mathtt{W} \to \mathtt{T}]$

- *Update count-matrix C_{a_i,a_j} by adding the new observed substitution counts.*

  For the first column the matrix is updated as follows:

  - $C_{\mathtt{T},\mathtt{T}} = C_{\mathtt{T},\mathtt{T}} + 3$
  - $C_{\mathtt{T},\mathtt{W}} = C_{\mathtt{T},\mathtt{W}} + 1$

  - $C_{\mathtt{T},\mathtt{T}} = C_{\mathtt{T},\mathtt{T}} + 2$
  - $C_{\mathtt{T},\mathtt{W}} = C_{\mathtt{T},\mathtt{W}} + 1$

  - $C_{\mathtt{T},\mathtt{T}} = C_{\mathtt{T},\mathtt{T}} + 2$
  - $C_{\mathtt{T},\mathtt{W}} = C_{\mathtt{T},\mathtt{W}} + 1$

  - $C_{\mathtt{W},\mathtt{T}} = C_{\mathtt{W},\mathtt{T}} + 1$

- *When all columns have been analysed, estimate the probabilities by dividing the number of observed substitutions by the arithmetic sum the matrix:*

  $p_{a_i,a_j} = \frac{C_{a_i,a_j}}{\sum \mathbf{C}}$

MDS on BLOSUM62 Because substitution matrices carry substitution rates, and MDS is able to preserve distances in a lower dimensionality, it is assumed that applying MDS on a substitution matrix will result in the evolutionary most important amino acid properties.
The BLOSUM62 matrix is probably the most popular substitution matrix because of its good performance in sequence alignments [11] and previous research has pointed out that the matrix can be used for MDS applications [16, 48]. For sequence alignment the matrix with rounded values is often used, which has explicitly not been done here, in order to preserve accuracy.

This raw matrix is given in section 7.4, in table 6 [13].
Substitution matrices do not meet the constraints of a distance matrix since they can be negative and the diagonals are typically not 0. They can therefore not directly be used for MDS. To achieve this the matrix has been transformed into a distance matrix using Euclidean distance. Its formal description is given in equation 3 where δ_{a_i,a_j} is the distance transformed matrix and B_{a_i,a_j} is the BLOSUM62 score for the substitution of amino acid a_i for a_j.

$$\delta_{a_i,a_j} = \sqrt{\sum_{i=1}^{n}(B_{a_i,a_j} - B_{a_j,a_i})^2} \qquad (3)$$

There are multiple solutions that try to find the objective of multi-dimension scaling. Therefore two have been implemented to apply upon the transformed BLOSUM62 matrix; *classical MDS* [42] and *Sammon's non-linear mapping* [29].
The preliminary results are given in advance, because they are essential for understanding upcoming methods. By visual interpretation, the results of the Sammon's non-linear mapping seem comparable to the results of previous research [16,48] although a different distance transformation has been used. The corresponding error levels of MDS are illustrated in figure 9a. Because the error levels of Sammon's non-linear mapping outperform classical MDS, the coordinates by Sammon's non-linear mapping will be used as property space axes. The corresponding coordinates are given in table 5.

	Coord. 1	Coord. 2	Coord. 3
A	0.118	1.305	-1.371
R	-2.327	-1.313	3.358
N	-3.453	-1.794	-2.118
D	-5.172	0.516	-1.788
C	3.183	4.039	-3.375
Q	-2.995	-2.147	1.385
E	-4.564	-0.711	0.700
G	-1.351	0.451	-4.456
H	-1.419	-4.729	-0.239
I	4.152	2.514	1.639
L	4.069	0.829	2.104
K	-3.161	0.063	2.302
M	2.436	0.387	3.405
F	5.181	-2.165	0.445
P	-2.969	4.759	0.183
S	-1.929	0.919	-0.994
T	-0.699	2.092	0.784
W	4.904	-3.642	-4.087
Y	3.104	-4.200	0.483
V	2.892	2.828	1.640

Table 5: The coordinates produced by Sammon's non-linear mapping derived from the BLOSUM62 matrix.

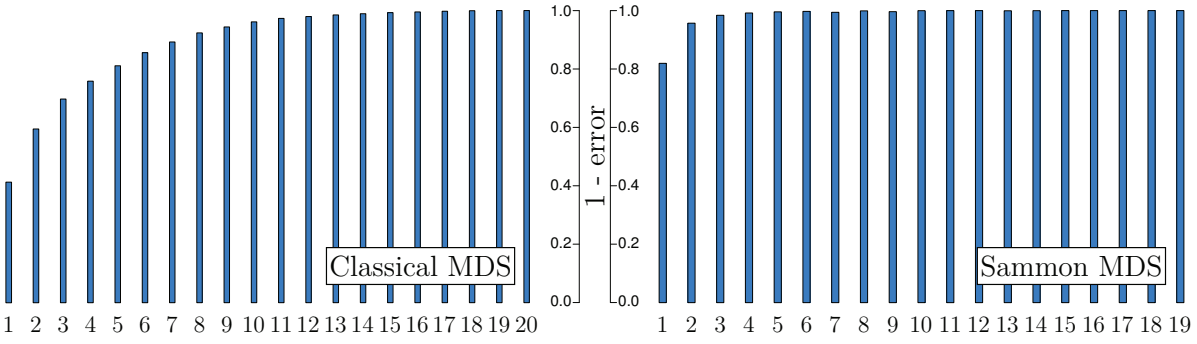
To understand what these properties actually mean in a biological context, the coordinates have been correlated known important amino acid properties using the Pearson correlation, as illustrated in figure 9b. This indicated the following correlations:

- Coordinate 1: Hydrophilicity (Hopp and Woods) [14]: -0.8370398
- Coordinate 2: Mass: -0.68

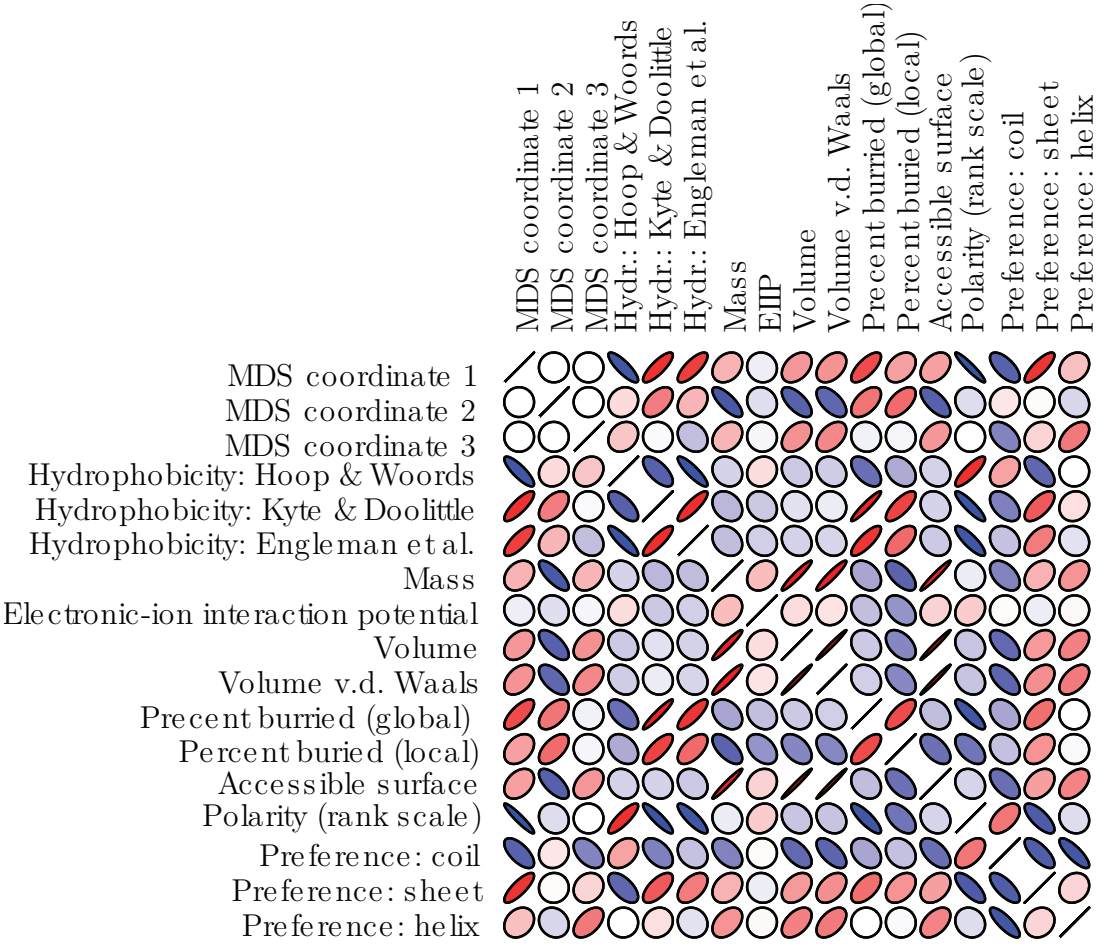
Because the coordinate 1 has a high correlation with hydrophobicity, it is referred to as the hydrophobic coordinate. Similarly coordinate 2 is referred to as the mass coordinate. Because the coordinate 3 is not having high correlation , it remains unidentified. The high correlation for the first two coordinates indicated that the method was indeed able to find properties that seem biologically relevant. Because of their biological meaning they can be used as coordinate system for a property space. In figure 10 is explained how to get an impression and how to interpret the space.

3.1.4 Composition

For the sequence composition analysis the composition has been estimated in terms of the multi-dimensional scaling (MDS) coordinates. Thus, a sequence is translated into



(a) The bar-plots indicate the cumulative success ratios per used number of MDS coordinates. The higher the success, the lower the amount of loss of information. **Top:** using classical MDS. **Bottom:** using Sammon's non-linear mapping.



(b) A projection of the Pearson correlation matrix of several amino acid properties and the MDS coordinates. High correlation is indicated with red ovals turned to the right. Anti-correlation is indicated with blue ovals turned to the left. The thickness and the amount of color indicate the amount of correlation. The MDS coordinates show correlation with properties like hydrophobicity, polarity and mass.

Figure 9: Results on the MDS analysis.

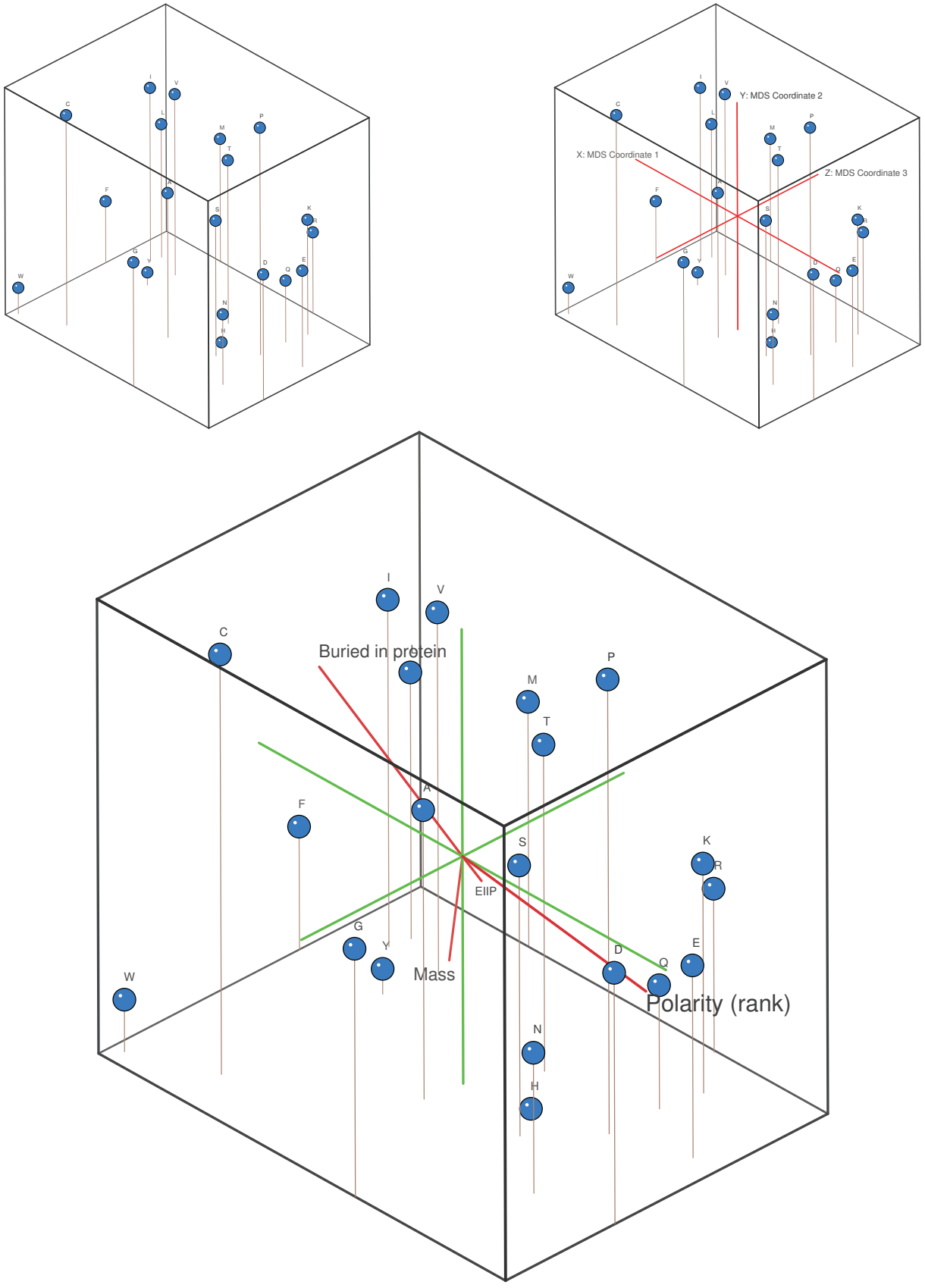


Figure 10: In this figure the property space is explained. In each plot the MDS coordinates 1, 2 and 3 represent x, y and z axes respectively. **Top-left:** the locations of the amino acids are drawn in the space. **Top-right:** the axes are included to support orientation. **Bottom:** also the correlating biological directions are included, to give the space a biological context.

three vectors consisting of the MDS coordinate values that correspond to the amino acids. For each of the vectors the average is calculated, which indicates the x, y and z-axis.
First, to get an impression of how the space looks like, the composition of every sequence in the entropy trajectory and the entire SCOP dataset is calculated and a 3D projection has been constructed.
Second, to get a more detailed impression about individual datasets, the average composition per class has been estimated for the following classes:

- SCOP classes A, . . . ,G (individually)
- SCOP (full)
- PFAM
- Merged IUP regions
- Artificial in silico (uniform)

3.1.5 Entropy

Shannon entropy can find the density of the usage of characters from a given alphabet in a string. The larger the entropy, the more dispersed the usage of characters is. In a textual context, a homo-polymer has the lowest entropy (zero) because the composition of characters is the furthest away from uniform. On the other hand, a amino acid composition with a uniform distribution has the highest entropy because the distribution of amino acids is dense. The formula for Shannon entropy is given in equation 4, where x is a amino acid of alphabet χ of which sequence $\mathbf{s}$ consists.

$$H(\mathbf{s}) = -\sum_{x \in \chi} p(x) \cdot log_2(p(x)) \qquad (4)$$

Previously it was addressed that the composition of amino acids in biological protein sequences is not uniform because of their biological context; amino acids `C` and `W` are relatively rare while e.g. `A` is not. This section will define Shannon entropy for protein sequences in two ways: in a textual- and the biological context. The formulation of entropy stays similar, but the formulation of the probability of finding amino acids differs. For textual entropy, the probability of finding char x in sequence $\mathbf{s}$ is denoted as p_T and is estimated by finding the frequency that it appears in the sequence. Thus, for appliance in Shannon entropy, $p(x)$ has to be replaced for $p_T(x)$. Assume that $f(\bullet)$ is a function that finds the frequency of observing a given char in a given string, the formal description of textual entropy is given in equation 5.

$$p_T(x) = f(x, \mathbf{s}) \qquad (5)$$

The probability of finding a specific composition in biology differs; every amino acid has its corresponding frequency in biological sequences. For biological entropy, the probability of finding char x in sequence $\mathbf{s}$ requires taking the frequency of x in biological sequences into account. The frequency x in all biological sequences $\mathbf{b}$ is denoted by $f(x, \mathbf{b})$ and the biological probability of finding x in sequence $\mathbf{s}$, $p_B(x)$, is formulated in equation 6, where $\langle \bullet \rangle$ is a vectors length operator. Thus, for the biological entropy, $p(x)$ has to be replace for $p_B(x)$ in the Shannon entropy formula. In fact, the formula finds a maximal entropy only when the composition is similar to the composition observed in $\mathbf{b}$.

$$p_B(x) = \frac{1}{\langle \chi \rangle} \cdot p_T(x) \cdot f(x, \mathbf{b}) \qquad (6)$$

For the analysis the entropy has been estimated for the following datasets:

- Biological sequences: entire SCOP dataset.
- Artificial in silico sequences.
- Artificial in silico sequences (preserved composition).

3.2 Local Measurements

The functions of proteins are based on their 3D structure which on their turn are formed by their sequences. Therefore it is plausible that important characteristics will be found at the local level of a sequence, like repetitive motifs, patterns in amino acid properties or some complex behaviour. The following sections will introduce methods that narrow the resolution down and focus on the local, also called internal, level of sequences.

3.2.1 Linguistic Complexity

A measurement which might find characteristics in sequences is linguistic complexity (LC). It finds the complexity of a string by estimating the ratio between the observed unique sub-sequences and the possible number of unique sub-sequences [43]. Notice that different sequences with the same entropy can still vary in their linguistic complexity because LC takes the internal sequence structure into account.
Under the assumption that word-lengths of different sizes have an equal contribution, the linguistic complexity finds in a sequence of length n the observed number of unique sub-sequences for all word-lengths k where $1 \leq k \leq n$, by counting them. The number of unique sub-sequences for all lengths is denoted as $\mathrm{o}(\mathbf{S})$.
The number of possible unique sub-sequences for all k is given in equation 7, where l is the alphabet size. The linguistic complexity of a sequence is the ratio between these two, given in equation 8.

$$\mathrm{p}(\mathbf{S}) = \sum_{k=1}^{n} min(l^k, n-k+1) \qquad (7)$$

$$LC(\mathbf{S}) = \frac{\mathrm{o}(\mathbf{S})}{\mathrm{p}(\mathbf{S})} \qquad (8)$$

It is important to realize that the function does not find the total number of possible sub-sequences according to the alphabet (permutations), but, the total number of possible sub-sequences of that size that can be found in the specific sequence length. For sequence $\mathbf{S}$ of length $n = 4$ amino acids, there are maximally two possible sub-sequences of length 3 that fit this length, namely $\{S_1,S_2,S_3\}$ and $\{S_2,S_3,S_4\}$.
The following example illustrates how the linguistic complexity is calculated, only for a word-length of $k = 2$. Imagine the following two sequences:

1 `AAABBBCCC`

2 `BACCABCBA`

The corresponding unique sub-sequences (only for length $k = 2$) are:

1 `AA AB BC CC`

2 `BA AC CC CA AB BC CB`

Sequence 1 contains four unique sub-sequences and sequence 2 contains seven unique sub-sequences. Assuming that both sequences may only consist of chars `A`, `B` or `C`, the alphabet size $l = 3$. The total number of possible sub-sequences for both sequences (only for $k = 2$) $\mathrm{p}(4, 3, k = 2) = 8$. The corresponding LC for sequence 1 is: 4/8 and for sequence: 7/8. Since both the sequences have a similar composition; $\langle \mathtt{A} \rangle = 3, \langle \mathtt{B} \rangle = 3, \langle \mathtt{C} \rangle = 3$, their corresponding entropy is identical, showing that sequences with a similar entropy can have a different LC.
The analysis is divided into two steps:

- The estimation of the distribution of the LC of biological and artificial sequences.
- The estimation of the likelihood of the LC with respect to other sequences with a similar composition.

For the first step, the distributions of LC for the biological (SCOP) and artificial (in silico with preserved composition) datasets have been estimated.

For the second step, for every biological sequence S from a given dataset, the likelihood of its LC has been estimated using the following protocol:

- The LC of S has been calculated; $X \leftarrow \mathrm{LC}(S)$.
- Sequence S has been shuffled 100 times, resulting in sequences $S'_1, \ldots, S'_{100}$.
- The LC for S'_i $(1 \leq i \leq 100)$ has been estimated; $X'_i \leftarrow \mathrm{LC}(S'_i)$.
- A corresponding normal distribution has been estimated on $\mathbf{X}'$, by using $\mu = median(\mathbf{S}')$ and $\sigma = variance(\mathbf{S}')$.
- The probability of finding the X in the distribution of $\mathbf{X}'$ has been calculated.
- To get an impression of the distribution of probabilities, a kernel density estimation has been projected.

Optimized software has been obtained upon request [43] and has been used for the calculations. Notice that the program was modified only such that the number of floating point significant digits was increased from 2 to 10.

3.2.2 Local Entropy Variance

Entropy is limited in finding sequence characteristics because it does not take the internal level of a sequence into account; it only acts on the composition level. In figure 11 is illustrated that images, which seem complex to humans, do not have a maximal entropy like pure chaotic images. Instead, complex figures consist of local high and low entropy regions which are essential for the formation of structures. Because similar behaviour could also be expected for protein sequences, it would be convenient also focus at different resolution rather than the entire sequence. The proposed solution for finding variation in entropy is *local entropy variance* (LEV), which measures for a sequence the entropy of every sub-sequence of one particular size and calculates the corresponding variance. Thus, for sequence $\mathbf{S}$ of length n, the LEV is calculated over all windows φ of window-size α as given in equation 9, where $E[\bullet]$ is the expected operator. Notice that every sequence contains $n - \alpha + 1$ windows which are overlapping.

$$\varphi_i = \{S_i, \ldots, S_{i+\alpha}\}$$

$$X_i = H(\varphi_i), \quad 0 \leq i \leq n - \alpha$$

$$LEV^2 = E[(\mathbf{X} - \mu)^2] \tag{9}$$

Since a biological and textual entropy have been proposed, both types haven been used for the LEV analysis. The analysis consists of two main steps:

- Estimation of the optimal window size.
- Comparison of LEV between biological and artificial sequences.

The rational behind the first part of the analysis, the optimal window size (α) estimation, is that the resolution containing the most information is yet unknown. The expectation is that the window size that contains most differences between artificial and biological sequences carries the most important information. The analysis compares the means of their distributions in order to find the largest difference. For two matrices, dataset a (artificial) and b (biological), consisting of vectors that contain the LEV for all sequences per window size α, the relative difference between means of the distributions of the datasets is estimated using equation 10, where σ_α is the overall variance of the LEV for both classes for the same α. In the equation, $\hat{b}_\alpha$ represents the mean LEV for biological sequences using a window size of α. Similarly, $\hat{a}_\alpha$ represents the mean LEV for artificial sequences.

$$\Delta(\mathbf{a}, \mathbf{b}) = \frac{(\hat{b}_\alpha - \hat{a}_\alpha)^2}{\sigma_\alpha} \tag{10}$$

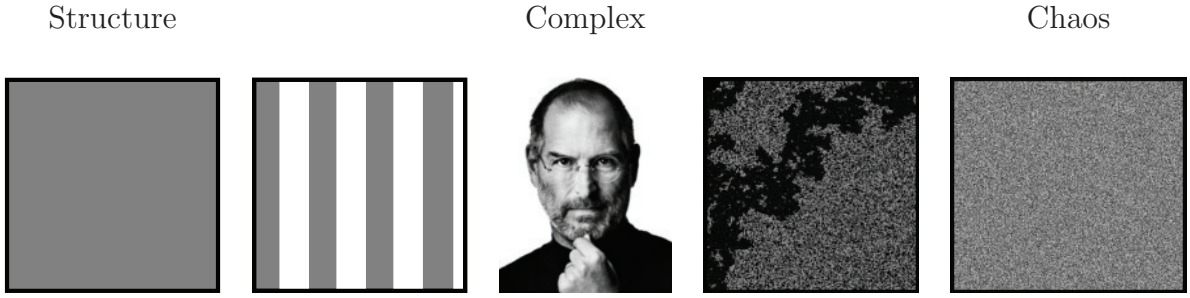


Figure 11: These figures have from left to right an increasing amount of entropy. However, the most complex picture between them, a picture of a human, seems not to be the figure with the highest amount of entropy but with the highest variation in entropy. This variation in entropy is essential for the formation of structures, like eyes, beard and forehead. In contrast, the figure with the highest amount of entropy seems to be pure chaos.

To avoid an unnecessary amount of calculations only α with the values $\{2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 25, 35, 50\}$ have been analysed. The Δ-estimations per α have been projected in a scatter-plot and the α with the largest Δ, belongs to the optimal window size.
The LEV has been calculated for a biological (SCOP) and artificial (SCOP shuffled) dataset. The corresponding LEV values have been projected in a scatter-plot for further analysis.

3.2.3 Local Variance Variance

The entropy of a protein sequence is not measuring variations in (biologically relevant) amino acid properties, like physicochemical properties for instance. Instead, it only measures the topology of local composition. In this section the local variance variance (LVV) will be introduced, which takes the variation of a particular amino acid property into account.
By focussing on individual amino acid property variation changes, complexity might be explained at multiple scales, like e.g. hydrophobicity or mass as well as on multiple resolutions. The idea behind LVV is similar to LEV; find the local variation in a certain enrichment measurement. The major difference with LEV is that instead of a sequence consisting of chars, a vector consisting of numerical properties is used. Entropy finds the sparsity of alphabet usage, whereas variance does something similar, but with a numerical vector. Therefore the entropy function is replaced for the variance function. If a high LVV is found, the window variances are heterogeneous and therefore such a sequences is referred to as heteroscedastic. In contrast, a low LVV means that the window variances are homogeneous and are therefore referred to as homoscedastic.
The properties that perfectly suit this application are the coordinates produced by multi-dimensional scaling (MDS) since they are assumed to be biologically important because they are derived from substitution rates. The formal description of LVV is given in equation 11, where $\mathbf{S}$ is a vector that replaces a sequence for a vector that consists of the corresponding amino acid properties.

$$\mu_i = \frac{\sum_{j=i}^{i+\alpha} S_j}{\alpha}$$

$$X_i = E[(S_{i,j} - \mu_i)^2], \quad 0 \leq j < \alpha$$

$$LVV^2 = E[(\mathbf{X} - \mu)^2], \quad 0 \leq i \leq n - \alpha \quad (11)$$

Symmetry The proposed implementation of LVV calculates the variance of a vector that contains variances. This is tricky

because the variance function assumes symmetrical data with respect to its expected value for input. This assumption does not hold because variance functions are χ^2-distributed [19]. The proposed solution is the symmetrical implementation of LVV, which is further referred to as *Symmetrical Local Variance Variance* (SLVV) and has been calculated as follows:

- Fit the vector of window variances to a χ^2-distribution.
- Find for every element in the vector the probability to find it in the fitted χ^2-distribution.
- Find for every χ^2-probability the corresponding observed value in a (symmetric) standard normal distribution.

Analysis The analyis conists of 3 major steps:

- The estimation of the optimal window size.
- The analysis on the differences in (S)LVV between artificial and biological sequences.
- The likelihood estimation of a particular (S)LVV value.

The first step, the estimation of the optimal window size, is similar to the method used in the LEV analyses. The window-size α has been estimated using the difference in means formulated in 10 and a corresponding scatter-plot projection has been used to indicate the optimal α.
In the actual (S)LVV analysis, for each MDS coordinate, the corresponding LVV distributions have been projected as a function of the overall variance of the property. This data has been used to indicate the differences between biological and artificial sequences.

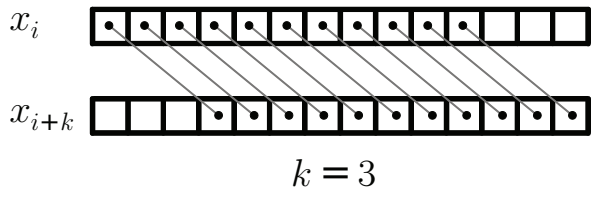


Figure 12: This figure illustrates the autocorrelation of numerical vector x. In principle it finds the correlation between sub-sequence $S_1, \ldots, S_{n-k}$ (top) with $S_{1+k}, \cdots, S_n$ (bottom).

The last step is the estimation of the likelihood of a particular (S)LVV value in comparison with other sequences having a similar composition. This has been done using the following protocol:

- Calculate for a sequence for every window φ_i its variances and put it in vector V_i.
- Create 100 shuffled sequences from the original sequence.
- For every shuffled sequences, calculate for every window φ_i its variance and add it to vector $\mathbf{W}$. Notice that vector $\mathbf{W}$ will be 100 times larger than $\mathbf{V}$.
- Estimate using the F-test, which assumes χ^2-distributions, the likelihood to find the variances in $\mathbf{V}$ in the distribution of $\mathbf{W}$.
- Estimate using the Bartlett-test, which is especially designed for comparing variances [3], the likelihood of finding the variances in $\mathbf{V}$ in the distribution of $\mathbf{W}$.
- To evaluate the results, for every α a corresponding density estimation of the probabilities has been constructed.

3.2.4 Autocorrelation

Autocorrelation finds the correlation of a sequence with itself. Between the start positions of two sub-sequences is a difference of k arbitrary chosen elements. This difference

is called the lag. Thus, the autocorrelation of vector $\mathbf{S}$ of length n is calculated between $S_1, \ldots, S_{n-k}$ and $S_{1+k}, \ldots, S_n$ [6, 22]. In figure 12 is illustrated how the autocorrelation works. The formal description is given in equation 12. In this formula $\mathbf{X}$ is the numerical vector of which the autocorrelation will be estimated and μ its expected value.

$$r_k = \frac{\sum_{i=1}^{n-k}(X_i - \mu)(X_{i+k} - \mu)}{\sum_{i=1}^{n}(X_i - \mu)^2} \qquad (12)$$

Its strength is that repeating patterns of a specific amino acid length can be detected without having a priori knowledge about the shape of the pattern. Since the multi-dimensional scaling (MDS) coordinates are assumed to be biologically important, the autocorrelation has been measured on all three coordinates for lag values from $k = 1, \ldots, 50$. In order to find out whether specific types of sequences have different correlation patterns. it was applied on the following classes of sequences:

- SCOP classes A, . . . ,G (individually)
- SCOP (full)
- Intrinsically unstructured proteins
- Artificial in silico (preserved composition)
- Fibrous proteins (Collagen)
- Functional artificial in silico de novo

The number of data-points for visualizing the autocorrelation is huge. Therefore projections have been constructed that aggregate the data, called "heatmaps".

3.2.5 Discrete Fourier Transform

The discrete Fourier transformation is a mathematical transformation that transforms a finite length vector consisting of numerical values into the frequency domain. In the frequency domain, using complex numbers, the vector is expressed as a combination of sines and cosines, represented by amplitudes for different frequencies. Thus, if data inside a vector contains an emphasized sine pattern of a particular frequency, that frequency shall obtain a higher amplitude in the transformation. It is useful for analysis because it can detect multiple frequency-specific sinusoid patterns with corresponding amplitudes. The formal expression of the discrete Fourier transform is given in equation 13. Here x is the transformed numerical vector of length N, e the base of the log transform and i the imaginary unit.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi/Nkn} \qquad k = 0, \ldots, N-1. \qquad (13)$$

The multi-dimensional scaling (MDS) coordinates are suitable for this operation since they represent biological important properties. The following datasets have been used for the Fourier transformation:

- SCOP classes A, . . . ,G (individually)
- SCOP (full)
- Intrinsically unstructured proteins
- Artificial in silico (preserved composition)
- Fibrous proteins (Collagen)
- Functional artificial in silico de novo

For each of the datasets the discrete Fourier transform has been applied to every individual sequence. Besides present patterns in properties, different lengths and compositions have an additional effect on the amplitudes. It is expected that the raw amplitudes lead to obscure results and therefore a normalization was applied. For every transformation, the amplitudes are divided through the standard deviation of the frequency domain.

Because the method provides a large number of data-points, the normalized data has been aggregated in so called "heatmaps". On top of the heatmaps the averages per binned frequencies are indicated. Because the averages can be noisy, they are smoothed with a LOESS regression using a span parameter of 0.1.

4 Results

Section overview The purpose of this chapter is to guide the reader through the answers of a journey through sequence space. This journey has two main questions:

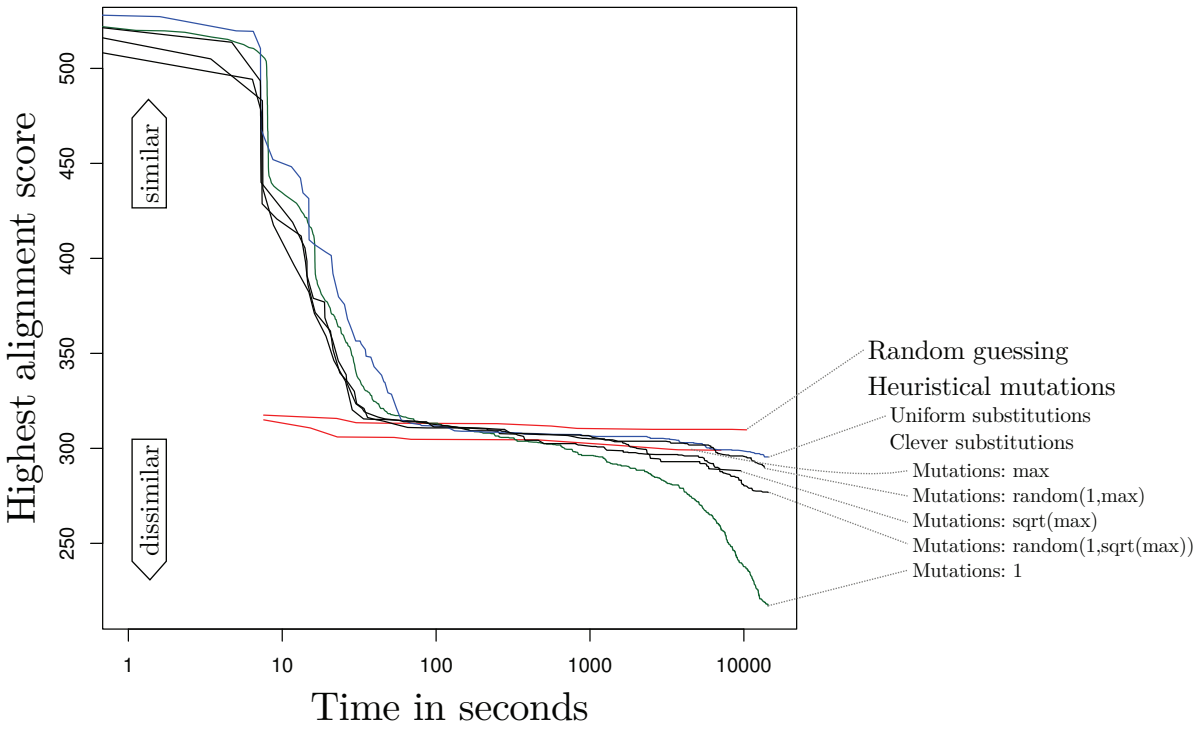
- What are convenient methods to find important locations in sequences space?
- What characteristics do specific groups of sequences have?

To answer the first question the section starts with the presentation of the results of the maximally distant sequence algorithm. To answer the second question, the section will present the results in two main analysis directions. This involves the results of global sequence analysis, followed by the local measurements that try to find patterns at lower resolutions.

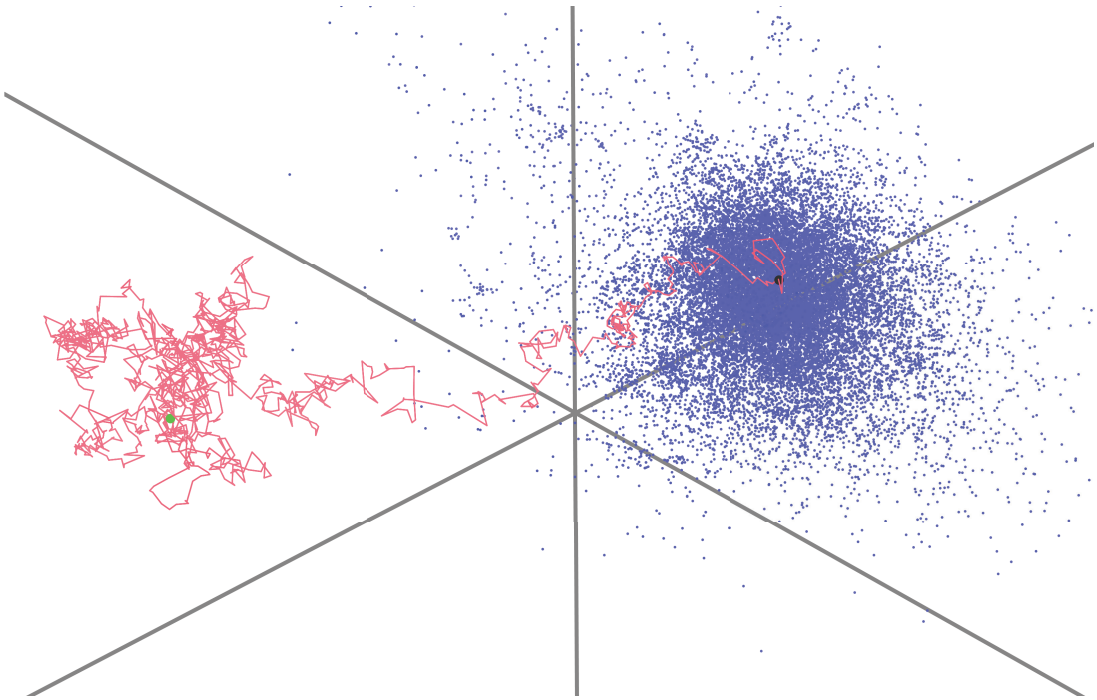
4.1 Maximally Distant Sequence

The performance of the maximally distance sequence has been estimated by measuring the alignment score of the target sequence as a function of time, illustrated in figure 13b. It shows that the algorithm outperforms a model that randomly generates sequences, without using the heuristic repair procedure. On top of that the heuristic repair procedure has been split up into two groups:

- A method that does not use the knowledge of the alignment. Therefore it does not take the precise mutation location and the most likely choice of amino acid into account. Instead, these two parameters are uniform randomly chosen. The results indicate that this is slower than using the information provided by the alignments.



(a) The algorithm performance of the maximally distant sequences estimates the alignment score of the target sequence as a function of time. The *random guessing* method randomly generates sequences without using heuristics. The other methods use heuristics and mutate the target sequence. The *uniform substitutions* method does not take the alignment locations into account but estimates them randomly. The remaining methods do take alignments into account, with a difference in number of mutations per iteration.



(b) The trajectory of most distant sequences (pink) in the property space using homogeneous scoring. The black dot is the point of the initial sequence and the green dot is the maximally distant sequence. The blue dots are the SCOP sequences.

Figure 13: The results of the maximally distant sequence algorithm.

- The methods that do use the knowledge of the alignment. For this analysis the number of mutations that take place each iteration, defined as ζ, has been varied. In the figure, max means the maximal number of possible unique mutations. This is the number of amino acids found to be similar to the target sequence in the alignment, in all sequences with the smallest evolutionary distance (defined as $\mathbf{K}$). The analysis indicates that the larger the number of substitutions per iteration is, the slower the algorithm finds more distant sequences. Its optimum is found using only 1 mutation per iteration. A small remark is that in the very early stage of the algorithm it seems to be an advantage to use a larger number of substitutions.

The found maximally distant sequences are given in supplementary section 7.3. They show, using both the homogeneous and BLOSUM62 type of scoring, that the sequences leave the biological composition and head towards the rare amino acids. Also, their entropy decreases the more distant the sequences become. The hypothesis was that, using a homogeneous scoring, the algorithm would find sequences that are heading towards the homo-polymeric `W` sequence. To support this finding, the algorithm has been initiated with this sequences but the results indicated that it is not a maximally distant sequence; using both the scoring procedures, the algorithm was able to find more distant sequences.

The implementation of the maximally distant sequence, called *yh-maximally-distant-sequence*, is free software and published under the open-source MIT license. It has been written in Python 2 and includes the proposed end-space free sequence alignment method. The code is publicly available at: `https://code.google.com/p/yh-maximally-distant-sequence/`

4.2 Global Measurements

4.2.1 Composition

It was previously addressed that the composition of sequences is difficult to visualize because the alphabet size makes the composition a high dimensional problem. Instead, the composition of sequences has been drawn in a property space using the biologically relevant MDS coordinates.

To get an impression of what the property space looks like, the entropy trajectory and SCOP sequences have been projected into it which is illustrated in figure 14. The balls represent locations in the space that can only be accessed by homo-polymers, which are only found in the entropy trajectory dataset. The SCOP sequences have a composition that is somewhat shifted towards the center, meaning that their entropy is large and that homo-polymeric sequences are rarely observed.

To get a more detailed impression of specific class compositions, the average contribution per coordinate has been calculated for several datasets. The corresponding results are illustrated in figure 15.

It illustrates that, with the exception of two, all biological datasets consist of a similar amino acid compositions. With respect to sequences that have a uniform distribution, their average compositions are shifted towards smaller and hydrophobic (and polar) amino acids. In contrast, membrane and surface proteins deviate from this composition. Their average amino acid size stays more or less similar but the preference for polarity (and hydrophobic) has changed into a preference for hydrophilic (and apolar) amino acids, even more than for uniform random sequences. The other dataset with a deviating composition are the intrinsically unstructured proteins. Their composition consists of somewhat smaller amino acids than the other biological sequences, but the preference for polarity (and hydrophobicity) is even larger than for the

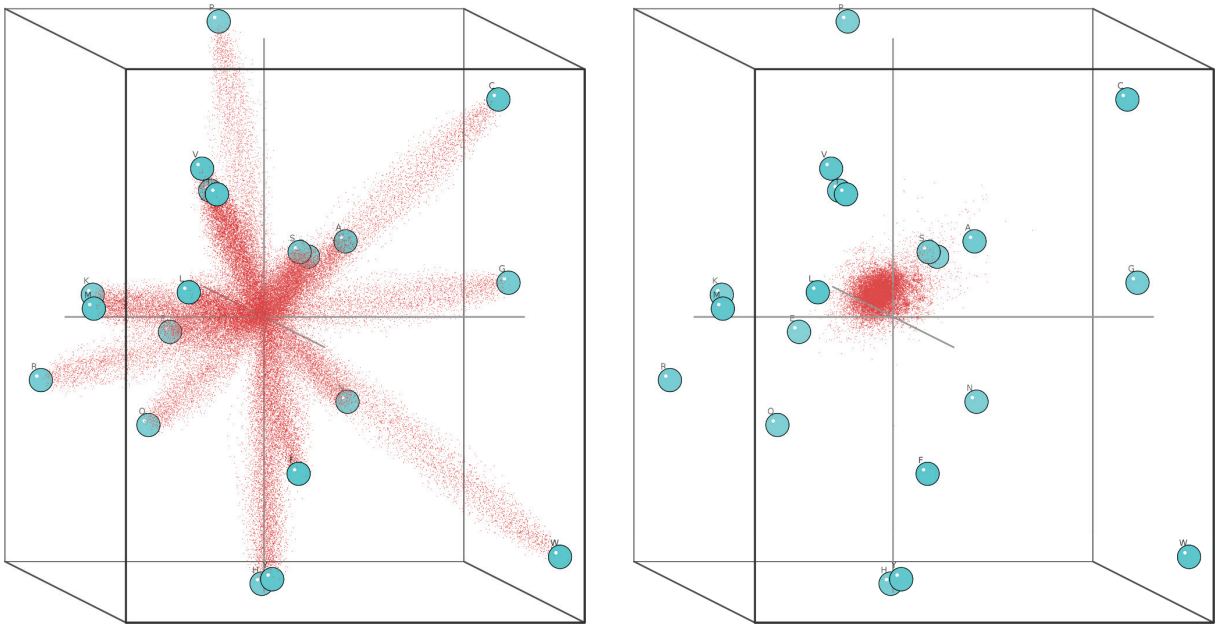


Figure 14: In these figures the 3D space of the MDS coordinates is shown. A blue ball represents an amino acid in the space, a red dot represents a sequence. For every sequence the average of a coordinates is used to find its location in the space. **Left:** the sequences of the entropy-walk dataset, indicating the boundary of the space. **Right:** the sequences of the SCOP dataset. In the figures the lowest entropy sequences are found near the balls representing amino acids since they are homopolymers. The highest entropy sequences are found where the x-, y- and z-axes cross each other. Clearly, the SCOP sequences are located near the higher entropy sequences compared to the entropy walk dataset.

other biological sequences.
To summarize, the composition analysis does indicate that proteins prefer certain compositions in the MDS property space.

4.2.2 Entropy

In addition to the analysis of sequence composition the entropy analysis finds the sparsity of the usage of amino acids in sequences. For biological and artificial (uniform and preserved composition) sequences, the entropy has been estimated. Because the composition analysis showed preferences towards certain compositions, two implementations of entropy have been used. The textual entropy, which assumes that the highest entropy is reached with a uniform composition and the biological entropy which assumes that the highest entropy is reached with the composition as found in biological sequences (see figure 6). Their distributions are given in figure 16, where the following has been observed:

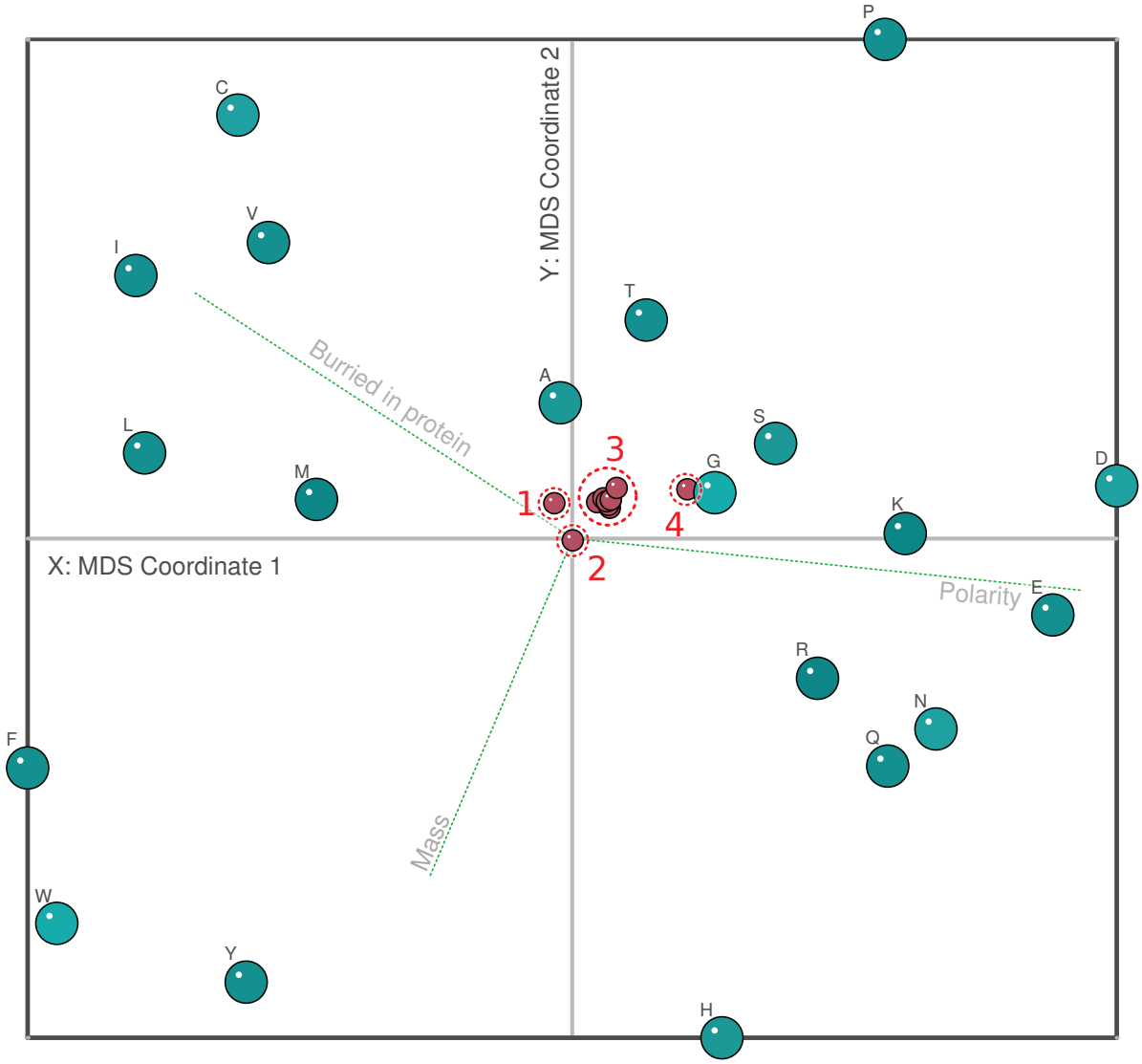


Figure 15: In this figure the x-axis represents the first MDS coordinate, which correlates highly with polarity and hydrophobicity. The second MDS coordinate is given on the y-axis and correlates with the mass of amino acids. Accordingly, every blue ball represents the position of an amino acid in this property space. The red balls represent the average composition of a dataset. In dashed circles are highlighted: **1:** SCOP class F (membrane proteins), **2:** random sequences with a uniform amino acid distribution, **3:** PFAM and all individual SCOP classes (except for SCOP class F) and **4:** the intrinsically unstructured proteins.

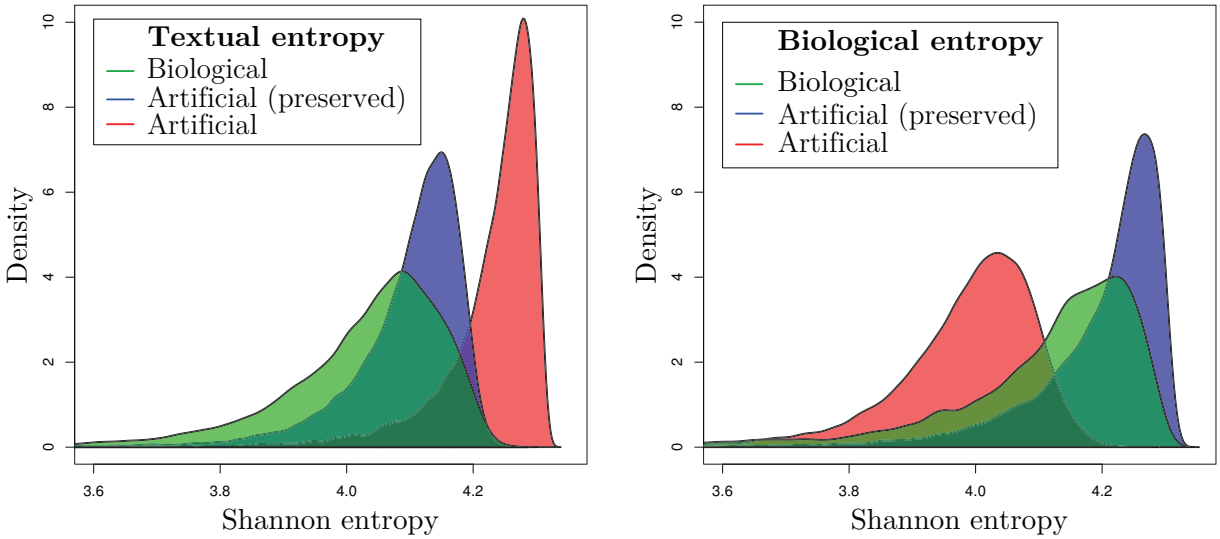


Figure 16: The Shannon entropy has been calculated for three datasets. The corresponding densities are illustrated in the figures above. **Left:** the textual entropy (calculated using uniform amino acid probabilities). **Right:** biological entropy where amino acid probabilities are normalized for their frequencies in nature. In both cases the entropy is in the range of 0 to ${}^2\log(20) = 4.32$.

- The textual entropy was found in the following order: artificial (uniform) > artificial (preserved) > biological. As expected, the artificial (uniform) sequences have the highest entropy, since their composition equals the uniform distribution that textual entropy expects. That the biological sequences have a lower textual entropy than artificial sequences with biological composition can be interpreted as follows: although the two datasets have an overall similar composition, the composition of individual biological sequences differs more than the artificial, suggesting the presence of redundancy in amino acid usage.

- The biological entropy was found in the following order: artificial (preserved) > biological > artificial (uniform). Similarly to textual entropy, this means that individual biological sequences deviate more often from the overall biological composition than artificial sequences do. And, as expected, since the biological entropy expects a biological composition, artificial (uniform) sequences differ from that composition such that their entropy is the lowest.

In brief, the entropy of biological sequences is high, in particularly biological entropy. However, their entropy is lower than for artificial sequences. This means that there is redundancy in amino acid usage in biological sequences. A part of this redundancy can be explained by the compositional subsets that have been indicated in the composition analysis On top of that, this also indicates that there is some selective redundancy; if a particular amino acid is found in a certain biological sequence it is more likely to find it multiple times than by chance.

4.3 Local Measurements

4.3.1 Linguistic Complexity

The first local measurement is the linguistic complexity, measuring the ratio between the number of unique and possible unique sub-sequences. For biological and artificial sequences the distributions have been estimated and are given in figure 17 (left). It

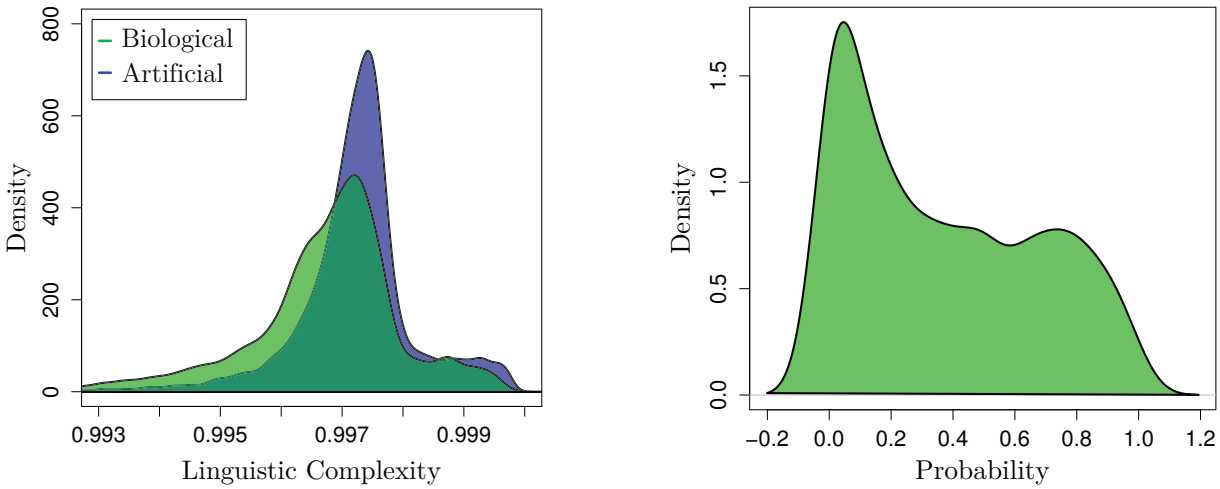


Figure 17: The results of the linguistic complexity analysis. **Left:** The kernel density estimation of the Linguistic Complexity of the entire set of SCOP proteins and the 25.000 randomly generated sequences with a preserved amino acid composition similar to the one of SCOP. **Right:** the kernel density estimation of likelihood of finding the LC of a biological (SCOP) sequence compared to 100 artificial sequences with a similar composition. Both the images show that the linguistic complexity of biological sequences is lower than random sequences with an equal composition of amino acids.

illustrates that the average distribution for artificial sequences is higher than for biological sequences.

Notice that the actual differences are relatively small and the the majority of all linguistic complexity values fall in the range $0.996 \sim 0.998$. A possible reason for this is that linguistic complexity assumes that sub-sequences of different lengths have an equal contribution to the measure. As result of this, linguistic complexity values of different sized sequences are expected to have different meaning, making it unreliable to compare them with each other. Therefore, an additional analysis has been done which tries to overcome this problem. In this analysis the linguistic complexity has been estimation for every biological sequence individually, and its likelihood of finding it in artificial sequences with a similar composition has been estimated. The estimated distribution of the corresponding probabilities is given in figure 17 (right). The figure indicates that there is a considerable deviation towards low probabilities, meaning that a majority of biological sequences have a significantly lower linguistic complexity than artificial sequences with a similar composition.

A summary of the results:

- Biological sequences have a lower linguistic complexity than artificial sequences while both the datasets have similar amino acid distribution.
- Biological sequences often have a considerably lower linguistic complexity than artificial sequences with a similar composition.

Taken the results together, it indicates that biological sequences are linguistically less complex than artificial sequences. This means that the number of sub-sequences in biological sequences are more redundant than can be expected by chance.

4.3.2 Local Entropy Variance

Because entropy only measures on the compositional level of a sequence, and sequences correspond to complex structures at lower resolutions, the LEV method was designed.

Window size The first step in its analysis is the estimation of the optimal window size. This parameter, α, is assumed to be optimal when it is able to separate biological from artificial sequence maximally. The results are illustrated in figure 18 (left) and point out that the estimated window size for textual LEV: $\alpha = 5$ amino acids, and for biological LEV: $\alpha = 6$ amino acids.

Analysis Using the estimated optimal window size the LEV was calculated for biological and artificial sequences. Their corresponding distributions are given in 18 (right). It shows that biological sequences are often found to have a higher LEV than artificial sequences, but a complete separation between the classes is not present. Thus, biological sequences more often change in their entropy than by chance. In contrast, sequences of which the entropy is at its maximum, LEV is approximately equal regardless whether its a biological or artificial sequence.

4.3.3 Local Variance Variance

Because LEV lacks to have a chemical meaning, the LVV was introduced. It finds for a sequence the variance of the window variance of an amino acid property.

Window size The first step in its analysis is the estimation of the optimal window size, α. It is assumed that window size that results in the maximal separation between biological and artificial sequences is optimal. The analysis indicated that the differences is about ~ 5 times lower for LVV than for SLVV (data not shown). For this reason, LVV was left out for further analysis. The optimal window size estimation for SLVV is given in figure 19. Per used property, the following optima have been estimated:

- For MDS coordinate 1: $\alpha = 6$
- For MDS coordinate 2: $\alpha = 6$
- For MDS coordinate 3: $\alpha = 3$

Analysis The next step in the analysis is the SLVV estimation using the optimal window sizes per property. The corresponding data has been projected as a function of the overall variance of a sequence, given in figure 20. Similarly to LEV, it indicates that biological often have a larger local variation than artificial sequences. Similarly, the separation between the two datasets is not complete, there is a subset of the biological sequences with a comparable SLVV to artificial sequences, in all three MDS coordinates. The best separation has been found in the following order: MDS coordinate 1 > MDS coordinate 3 > MDS coordinate 2.
Additional analysis indicated that the SLVV measurements using different properties are uncorrelated to each other (data not shown).

Likelihood The last step in the analysis is the likelihood estimation of finding a certain local variance variance with respect to other sequences having a similar composition. In both the F-test and Bartlett-test the following has been observed: when the length of the window sizes increase, the probabilities for both the biological and artificial dataset (null-hypothesis) decrease. This suggests that, for artificial sequences, the likelihood of finding its local variance variance would be significantly different compared to other artificial sequences. Because this suggestion can not be true, the method is considered to be unreliable. To avoid misinterpretations, the data has been moved to supplementary section 7.5, in figure 40.

4.3.4 Autocorrelation

To find repetitive patterns in sequence properties, an autocorrelation analysis was applied. For several datasets, the sequences

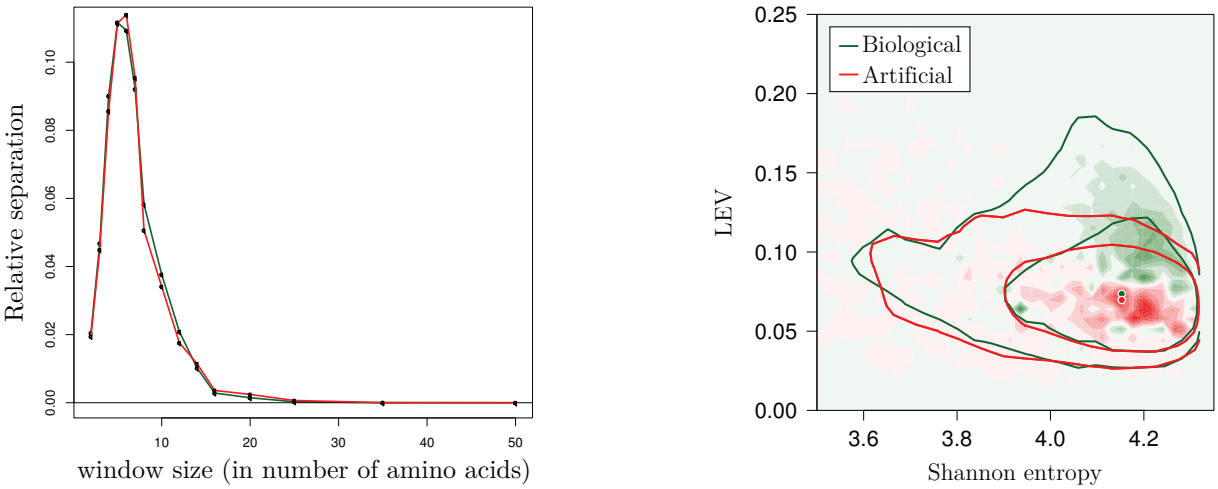


Figure 18: The results for LEV are given in this figure. **Left:** to find the optimal window size, the difference in means ($\Delta(\mathbf{a}, \mathbf{b})$; equation 10) between biological (SCOP) sequences and artificial (shuffled SCOP) sequences are indicated. The difference is the largest using an α of 5 (textual; red line) and 6 (biological; green line) amino acids. **Right:** for all biological (SCOP) and artificial (shuffled SCOP) sequences the textual entropy and the textual LEV for $\alpha = 5$ has been projected. On the x-axis the textual entropy (in bits) is drawn, on the y-axis the LEV. Because of the high number of data points the data was aggregated into a contour plot. The lines prepresent the contours at 25% (inner contour) and 1.5% (outer contour), with their medians in the middle. In the background the colors illustrate what the majority of type of sequences in that area is; the more green; enriched with biological sequences, the more red; enriched with artificial sequences, white; equilibrium. The figure shows that the LEV for biological sequences is considerably higher.

have been translated into vectors of properties. Similarly to the SLVV analysis, the MDS coordinates have been used because of the assumption that they are biologically relevant. The autocorrelation per dataset per property, has been estimated using lag values in the range from 1 to 50 amino acids. The analysis of a dataset per property has been projected in a heatmap. In such a heatmap every column represents a lag value and every row represents the amount of autocorrelation ($-1 \leq r \leq 1$). The surface has been split into a fixed number of bins where the colour of a bin represents the amount of sequences that have that particular amount of autocorrelation per lag value. Red indicates a low and yellow a high presence of sequences.

The results of SCOP classes [A & B], [C & D], [E & F] and [G] are given in figures 21,22,23 and 24 respectively. In figure 24 also the autocorrelation for intrinsically unstructured proteins is given. The autocorrelation of sequences from the entire SCOP database and the artificial in silico sequences (preserved composition) are given in figure 25. In figure 26 the autocorrelation for fibrous protein Collagen and the artificial synthesized proteins are given.

At first glance the figures show that there is no high autocorrelation in any type of sequence. This means that a high level of uncorrelated fluctuation in these properties is observed. The autocorrelation for Collagen (using MDS coordinate 3) is an exception. It has an overall high correlation for all lag values that are a multiple of 3. The second, though less clear, exceptions are SCOP class G (smaller proteins) and the artificial functional sequences. They seem to have a more scattered amount of autocorrelation for all 3 types of coordinates. Notice that both these

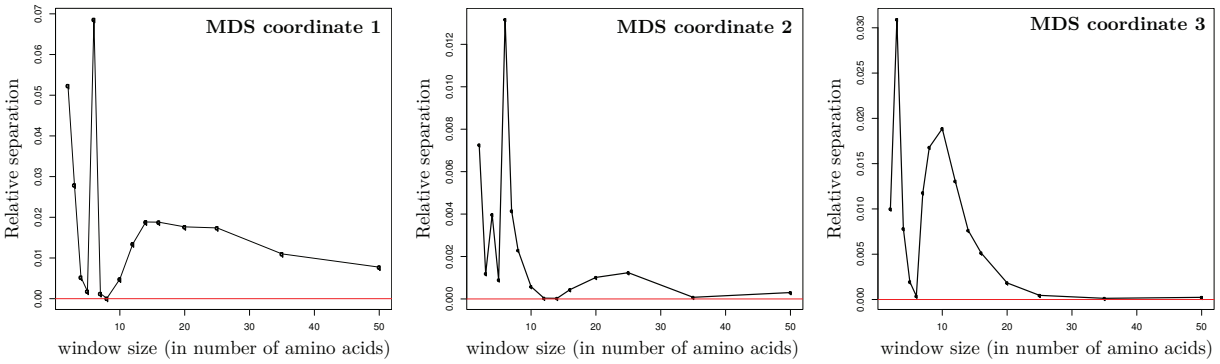


Figure 19: The estimation of the optimal window size using function 10, which finds the difference the means between the biological and artificial sequences. The window size with the highest separation is considered the optimal window size. The following optimal window sizes are observed: MDS coordinate 1 has $\alpha = 6$, MDS coordinate 2 has $\alpha = 6$ and MDS coordinate 3 has $\alpha = 3$.

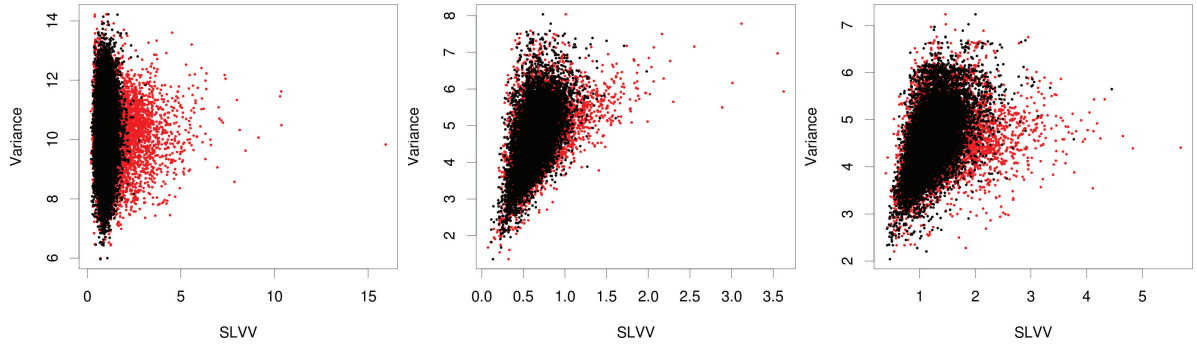


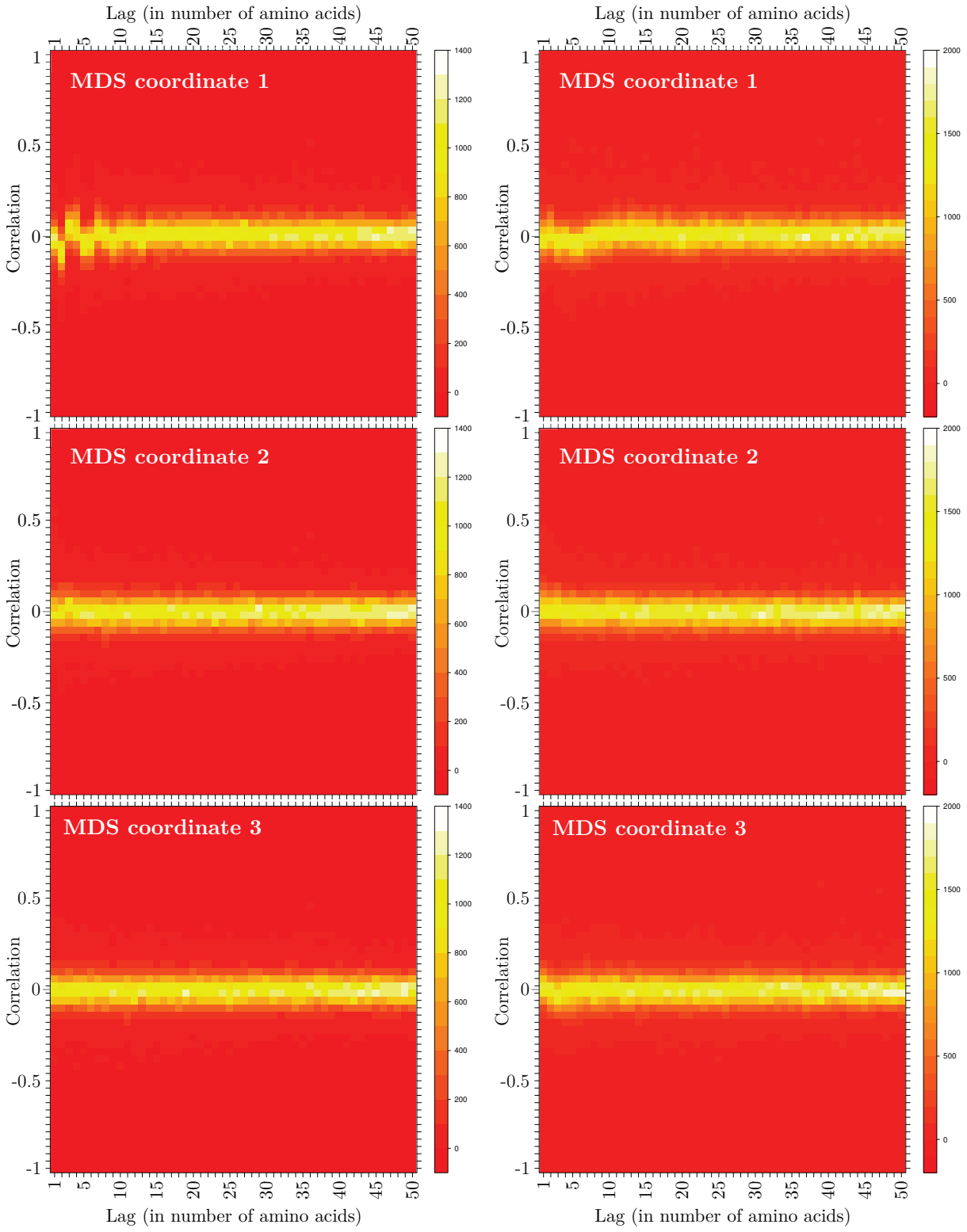
Figure 20: The SLVV data values for biological (red) and artificial sequences (black) are drawn as a function of the overall variance. **Left:** LVV for MDS coordinate 1 using an α of 6 amino acids. **Center:** LVV for MDS coordinate 2 using an α of 6 amino acids. **Right:** LVV for MDS coordinate 3 using an α of 3 amino acids.

types of sequences are relatively small.

If the figures are studied in more depth, and individual classes of sequences are compared with the results of artificial in silico sequences, group specific patterns become visible. The autocorrelation using the MDS coordinate 1 (hydrophobicity) of SCOP classes A, C, D and E, the entire SCOP dataset and also the intrinsically unstructured proteins, all containing α-helices, show a sinusoid like wave of correlation with a peak around a lag of $\sim$3 to $\sim$4 amino acids.

In contrast, SCOP class B, consisting of β-sheet proteins, shows a little wave-like shape, only with a peak near $14 \sim 15$ amino acids and a lower frequency. Notice that this sinusoid shape is less obvious than the previous.

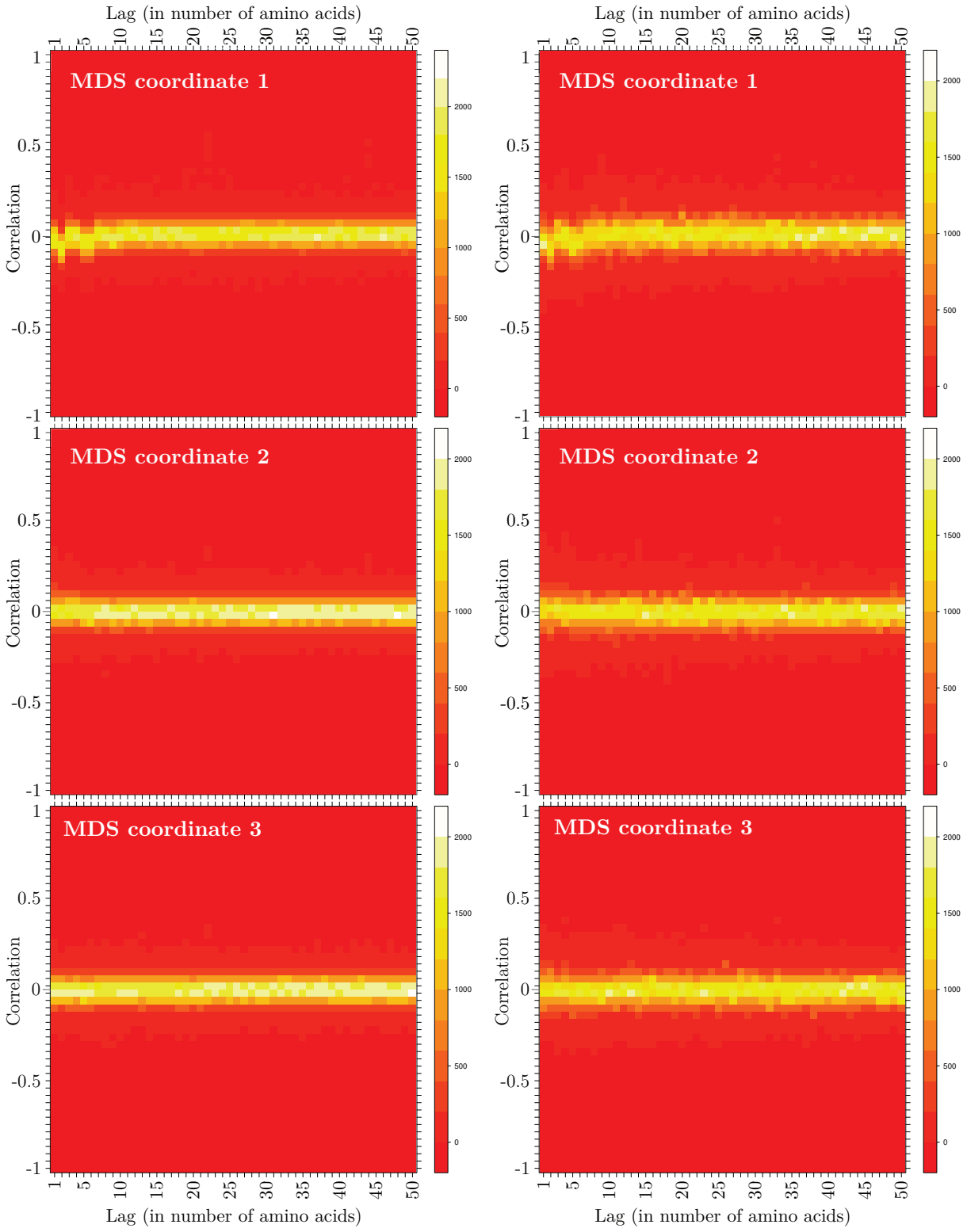
MDS coordinate 2 does not indicate a correlation in any dataset. Neither does MDS coordinate 3, with the exception for Collagen.



(a) SCOP class A

(b) SCOP class B

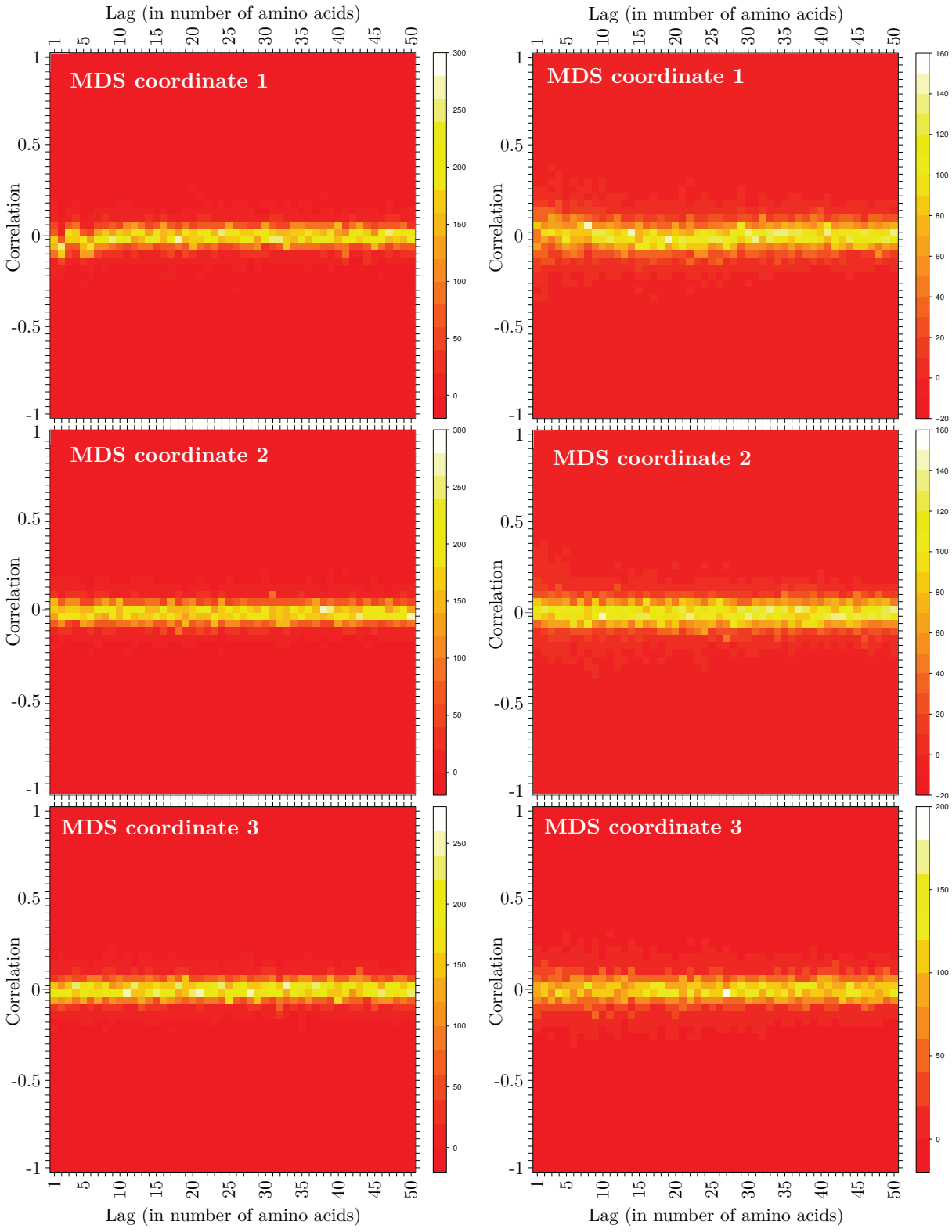
Figure 21: Heatmaps of the autocorrelation analysis of SCOP classes A (left) & B (right). From top to bottom: MDS coordinate 1, 2 and 3. MDS coordinate 1 shows a sinusoid pattern in both classes. However, the frequency of this pattern differs; for class A this frequency is higher than for class B. The peak for class A seems to be near a lag value of $3 \sim 4$ amino acids, and for class B this is around ~ 15 amino acids. For class B this seems to be near although the frequency for class A is higher than for class B. For coordinates 2 and 3, class B has a small elevation in the autocorrelation for lag of 2 and 3 amino acids.



(a) SCOP class C

(b) SCOP class D

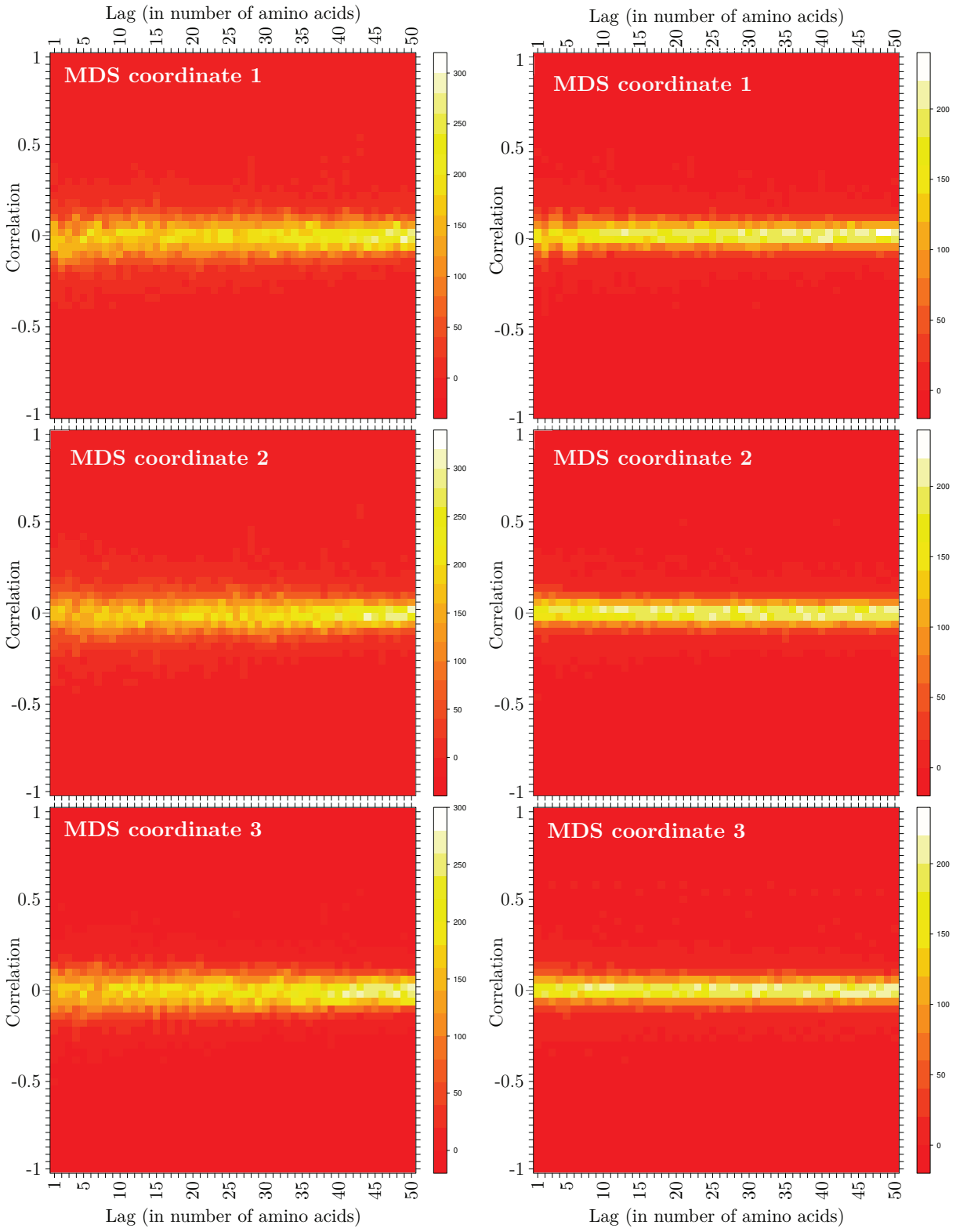
Figure 22: Heatmaps of the autocorrelation analysis of SCOP classes C (left) & D (right). From top to bottom: MDS coordinate 1, 2 and 3. Both classes C and D show a sinusoid pattern equal to the observation inf class A: in MDS coordinate 1 there is a sine-wave like pattern with a peak and frequency near $3 \sim 4$ amino acids. However, in these datasets the observation is less clear than for SCOP class A. The pattern that was observed in class B is not visible in class C. The larger variance in class D makes it difficult to judge whether that patterns is present there. The remaining MDS coordinates seem to have no patterns.



(a) SCOP class E

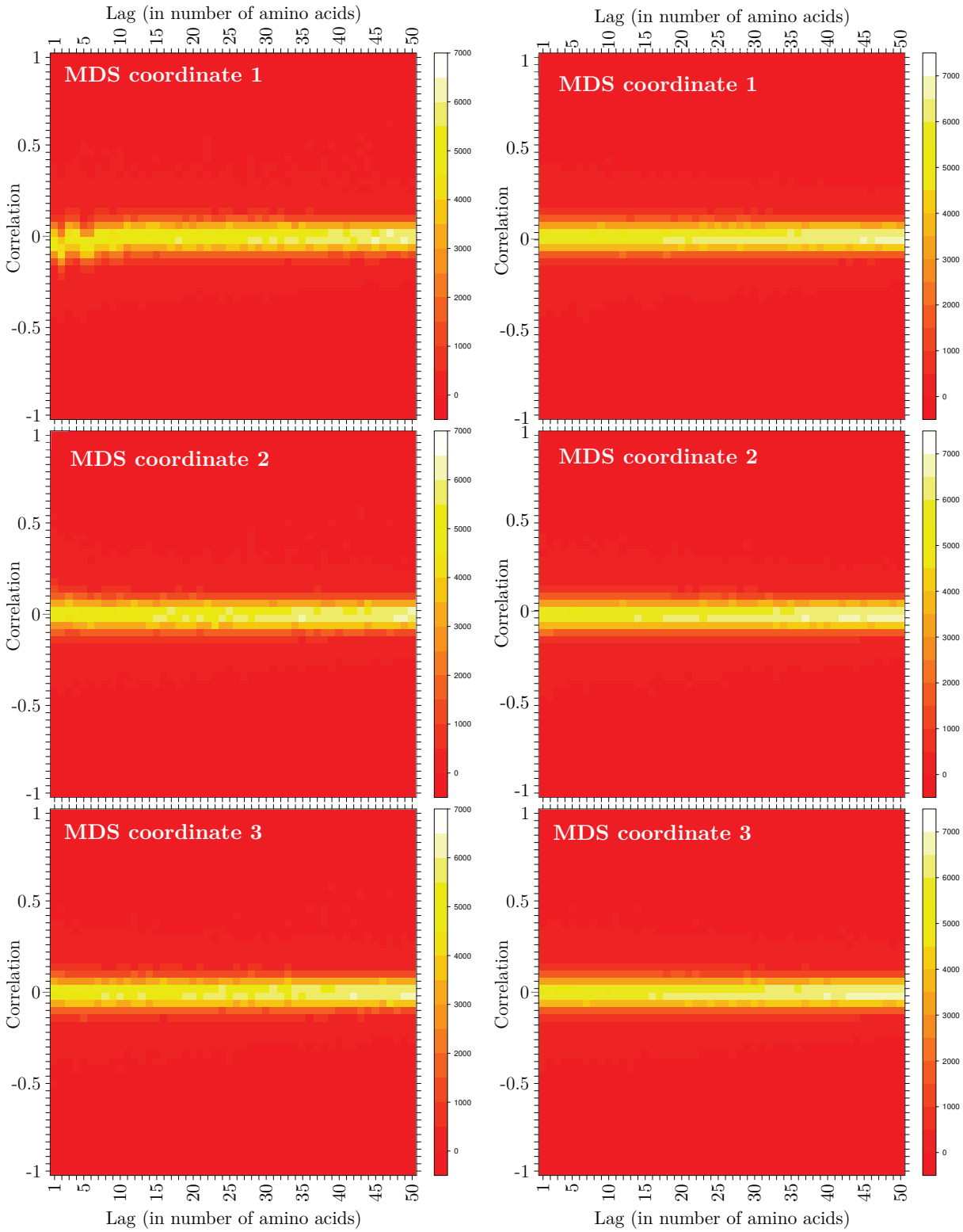
(b) SCOP class F

Figure 23: Heatmaps of the autocorrelation analysis of SCOP classes E (left) & F (right). From top to bottom: MDS coordinate 1, 2 and 3. SCOP class E shows a similar pattern to what has been observed in class A: a sinusoid pattern with an peak value near a lag of 3 or 4 in MDS coordinate 1. Class F shows in MDS coordinate a higher correlation for lower lag values, although a sinusoid wave is not observed here. The larger variation or the reduced sample size makes the illustrates a little fuzzy, which makes it difficult to interpret the figure. The other two MDS coordinates do not indicate correlation patterns.



(a) SCOP class G (b) IUP

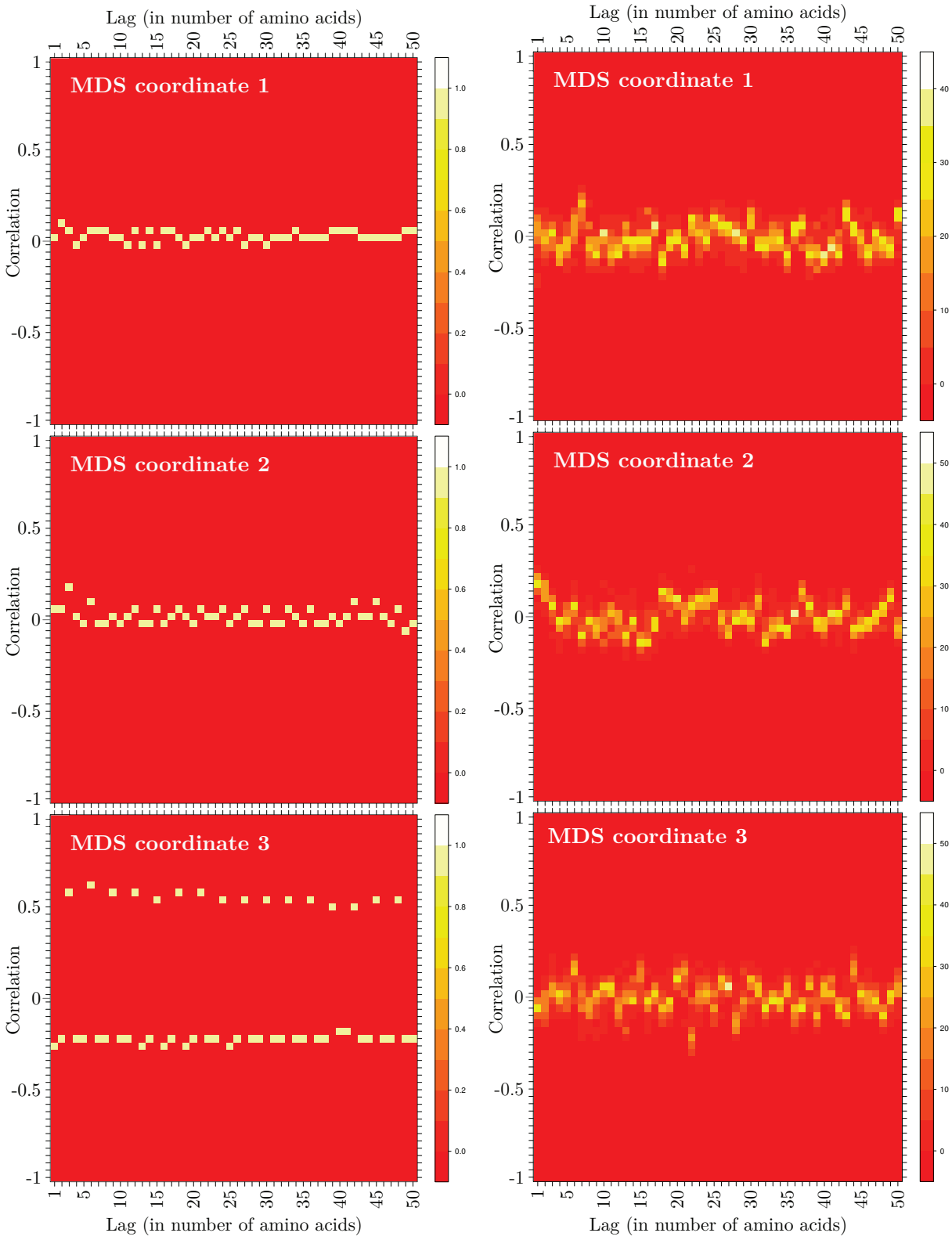
Figure 24: Heatmaps of the autocorrelation analysis of SCOP classes G (left) & the intrinsically unstructured proteins (right). From top to bottom: MDS coordinate 1, 2 and 3. The IUP show, similar to the observation of SCOP class A, a sinusoid pattern in correlation with a peak near $3 \sim 4$ amino acids. For SCOP class G the large variation makes it is hard to estimate any pattern. For class G the overall correlation is higher and even more scattered near lower lag values.



(a) all SCOP classes

(b) Artificial In Silico

Figure 25: Heatmaps of the autocorrelation analysis of all SCOP- (left) & artificial in silico sequences with a peserved composition (right). From top to bottom: MDS coordinate 1, 2 and 3. The suoerset of SCOP sequences show in MDS coordinate 1 a correlation pattern similar to the observation of class A but lack the correlation pattern as observed in class B. The in silico sequences do not show any patterns and have a relatively low overall correlation.



(a) Collagen

(b) Artificial functional

Figure 26: Heatmaps of the autocorrelation analysis Collagen (left) & artificial functional sequences (right). From top to bottom: MDS coordinate 1, 2 and 3. For MDS coordinate 1 and 2, Collage does not show clear correlation patterns. For MDS coordinate 3 however, a high correlation is observed for every lag value that is a multiple of 3. The artificial functional sequences have for all MDS coordinates a relatively high autocorrelation. On the other hand, there are no obvious repetitive patterns observed.

4.3.5 Discrete Fourier Transform

To overcome the shortcoming of the autocorrelation, the discrete Fourier transform has been applied on several datasets the MDS coordinates.
The analysis of SCOP classes A,...,G are given in figures 27, 28, 29, 30, 31, 32 and 33 respectively. Analysis on the superset of all SCOP classes can be found in figure 34, and the IUP results are given in figure 35. The artificial in silico sequences are given in figures 36 and 37 (preserved composition). The results for Collagen are given in figure 38. The figures can be explained as follows:

- The x-axis represents the frequency of a found pattern. The frequencies are translated into numbers of amino acids, to give them a biological context. A frequency of 10 amino acids indicates a sinusoid pattern in an MDS coordinate, that forms a sine within exactly 10 amino acids.

- The y-axis represents the amplitude in the frequency domain. The higher this amplitude, the more a pattern is present. Because the sequences differ in length and in composition, the frequency domains have been normalized. The y-axis represents the normalized amplitudes.

- To aggregate the large number of datapoints, the surface per plot has been discretized into bins, where colors represent the presence of sequences. Yellow indicates a higher number of sequences.

- Notice that the datasets differ in numbers of sequences and therefore larger variances are observed for smaller datasets. For this reason, on top of each heatmap, the raw- and LOESS regressed averages are drawn to reduce noise.

MDS Coordinate 1 The Fourier transformation MDS coordinate 1 (hydrophobicity) has indicated several dataset specific trends. An increase in the amplitude for frequencies around $3.5 \sim 4$ amino acids is observed in:

- SCOP classes A, C, D, E and F (vague).
- SCOP superset.
- Intrinsically unstructured proteins.

An increase in the amplitude for frequencies around ~ 15 amino acids is observed in:

- SCOP classes B, C, D (vague) and E.
- SCOP superset
- Intrinsically unstructured proteins (vague)

For the following datasets the amplitude for low frequencies (> 50 amino acids) drops:

- SCOP classes A, B, C, D, E and G.
- SCOP superset

For the following datasets the amplitude for low frequencies (> 50 amino acids) rises:

- SCOP class F
- Intrinsically unstructured proteins
- Collagen (1 sequence)

MDS Coordinate 2 MDS coordinate 2 has a small pattern that returns for every dataset with biological sequences except for SCOP class A. For frequencies that are low, say > 50 amino acids, the amplitude is elevated. This observation is the most obvious in SCOP class F, SCOP class G and the intrinsically unstructured proteins.

MDS Coordinate 3 With the exception of SCOP class E and Collagen, MDS coordinate 3 always follows the behaviour of MDS coordinate 2. For class E, for lower frequencies of > 50 amino acids, MDS coordinate 3 drops in amplitude. For Collagen, there is a clear elevation in amplitude for a frequency of exactly 3 amino acids.

Artificial in silico The artificial sequences that are constructed by pure randomization schemes of a computer do not indicate any overall regular pattern in any of the MDS coordinates.

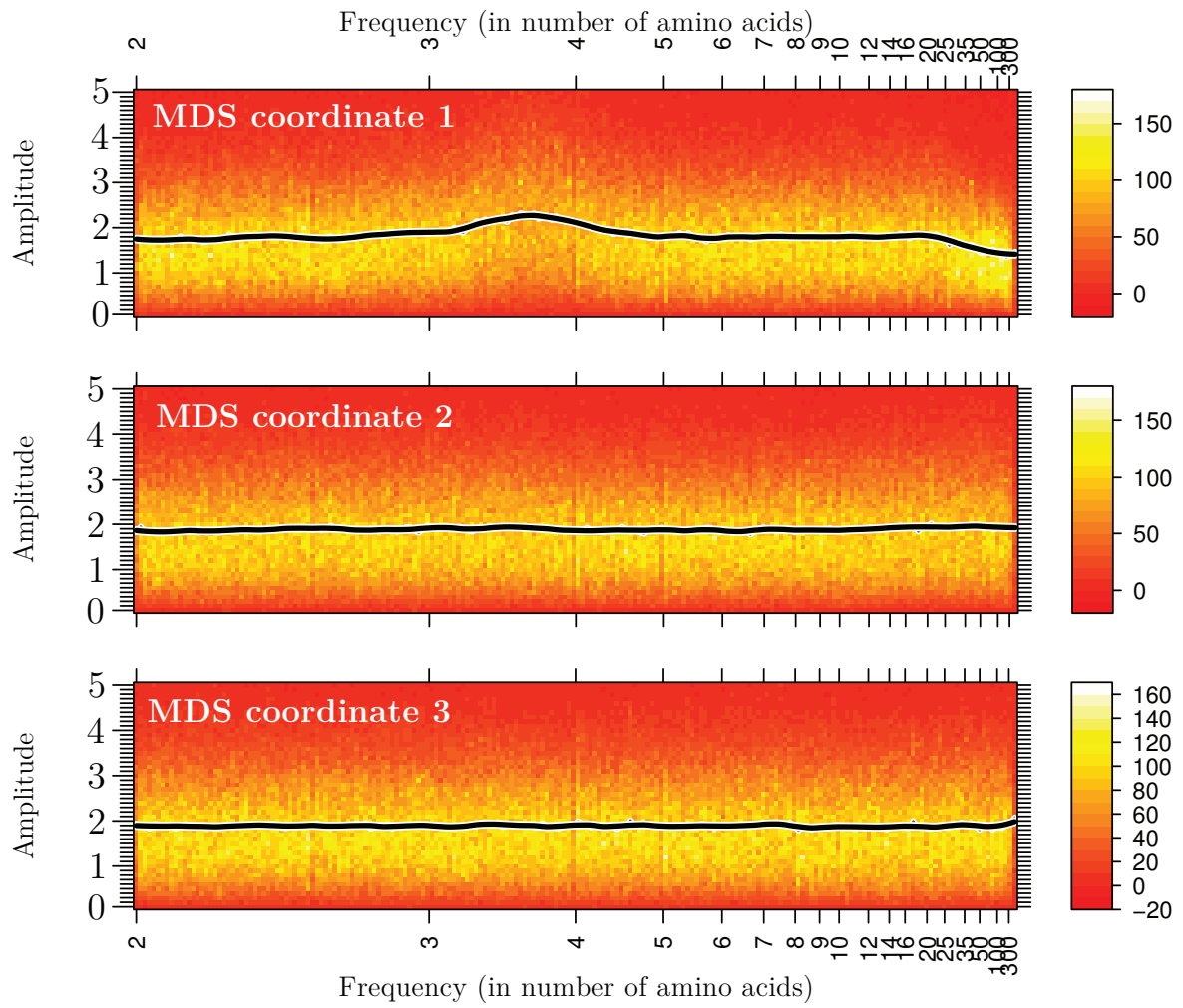


Figure 27: Discrete Fourier transform on SCOP class A. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). In MDS coordinate 1 an elevation is observed for frequencies near $3.5 \sim 4$ amino acids and a drop for > 25 amino acids. MDS coordinate 3 has a small increase in amplitude for frequencies > 100 amino acids.

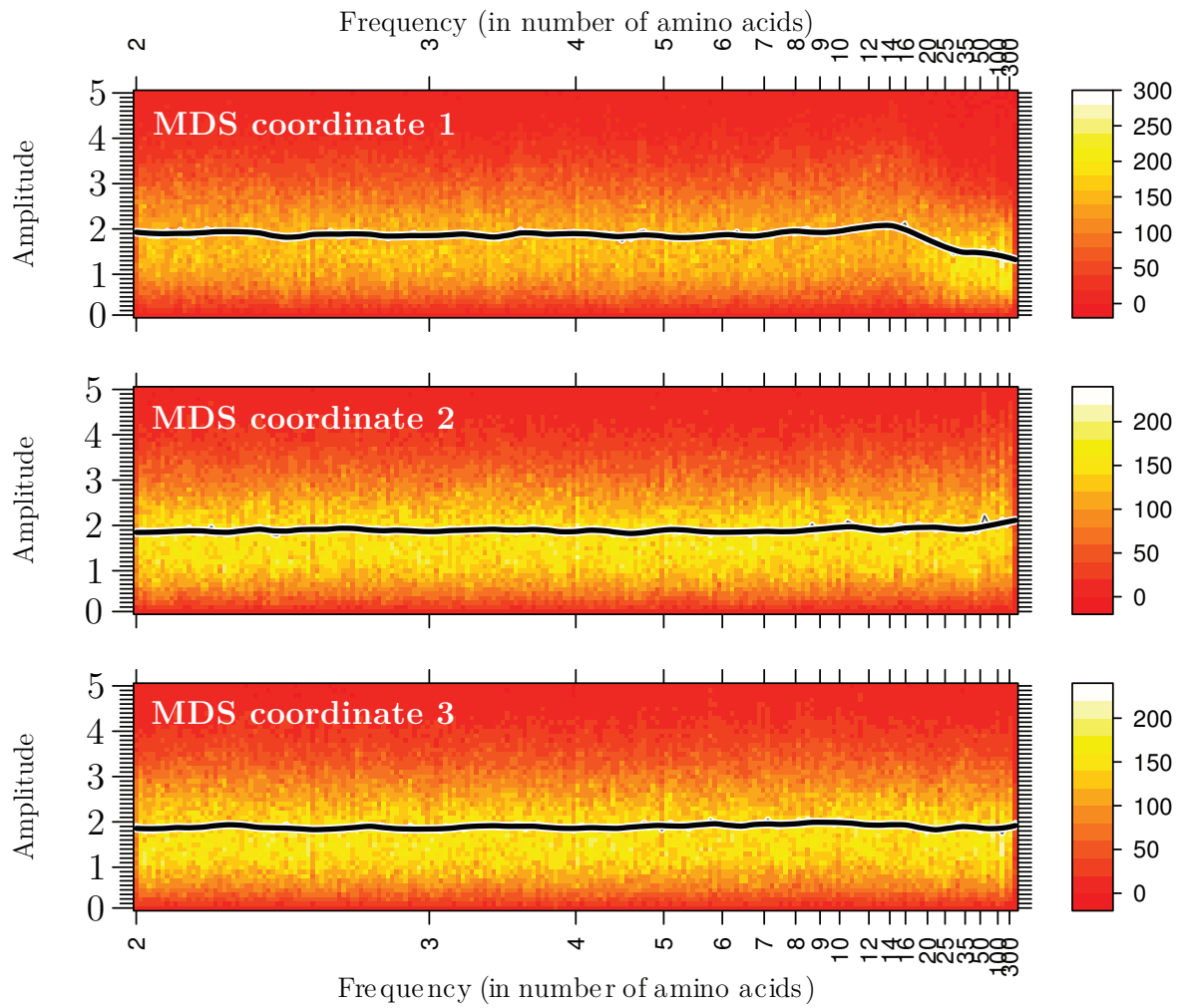


Figure 28: Discrete Fourier transform on SCOP class B. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). MDS coordinate 1 has an increased amplitude for frequencies of ~ 15 amino acids. MDS coordinate 2 has a small increase in amplitude for frequencies > 100 amino acids.

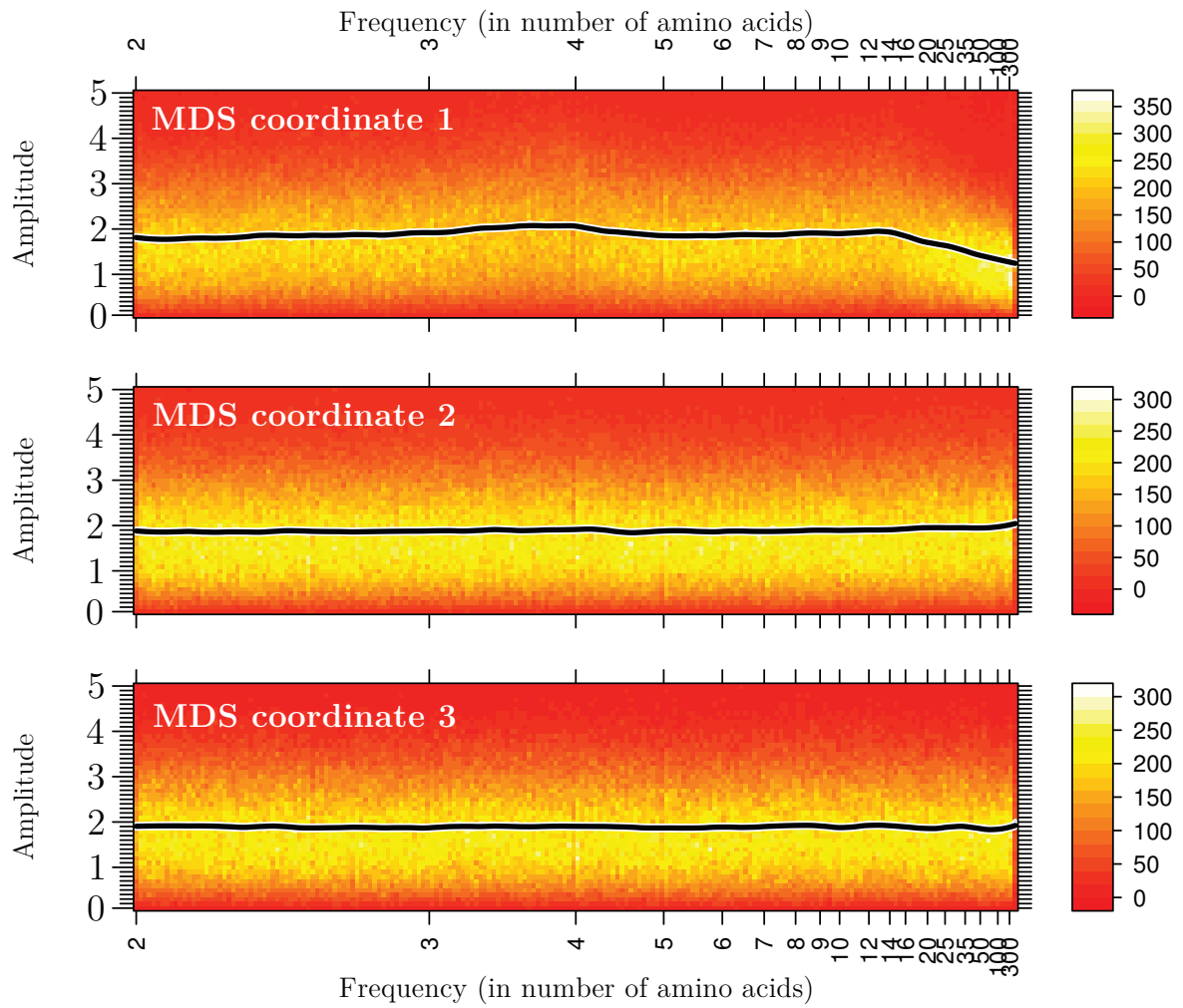


Figure 29: Discrete Fourier transform on SCOP class C. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). In MDS coordinate 1 an elevation is observed for frequencies near $3.5 \sim 4$ amino acids and a drop for > 25 amino acids. It also has an increased amplitude for frequencies of ~ 15 amino acids. Although it is not obvious, it looks like MDS coordinates 2 and 3 have a tiny increase in amplitude for frequencies > 100 amino acids.

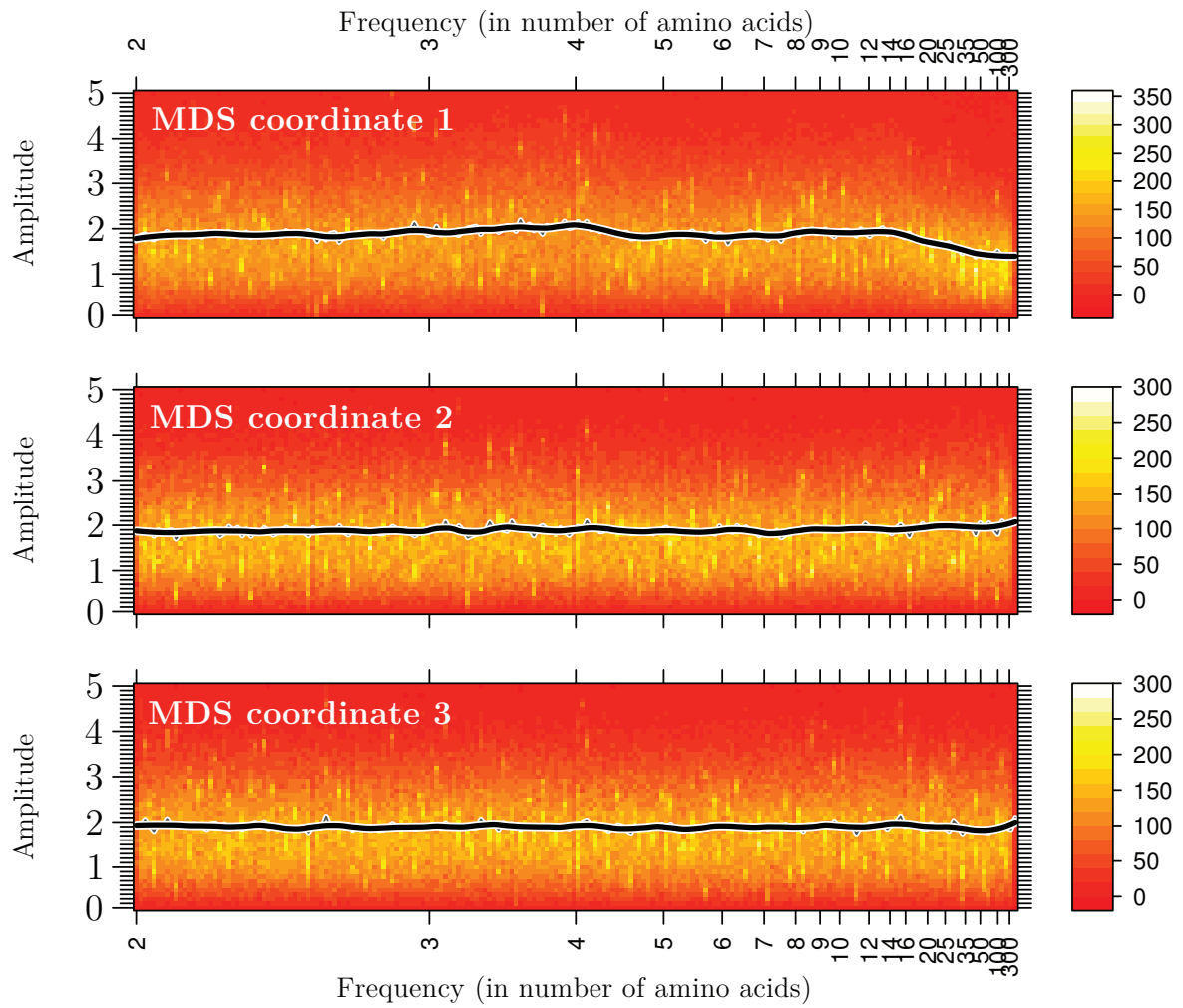


Figure 30: Discrete Fourier transform on SCOP class D. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). In MDS coordinate 1 a minor elevation is observed for frequencies near $\sim$ 4 amino acids and a drop for $>$ 25 amino acids. The amplitude for frequencies of $\sim$ 15 amino acids is also slightly elevated. MDS coordinates 2 and 3 have a tiny increase in amplitude for frequencies $>$ 100 amino acids.

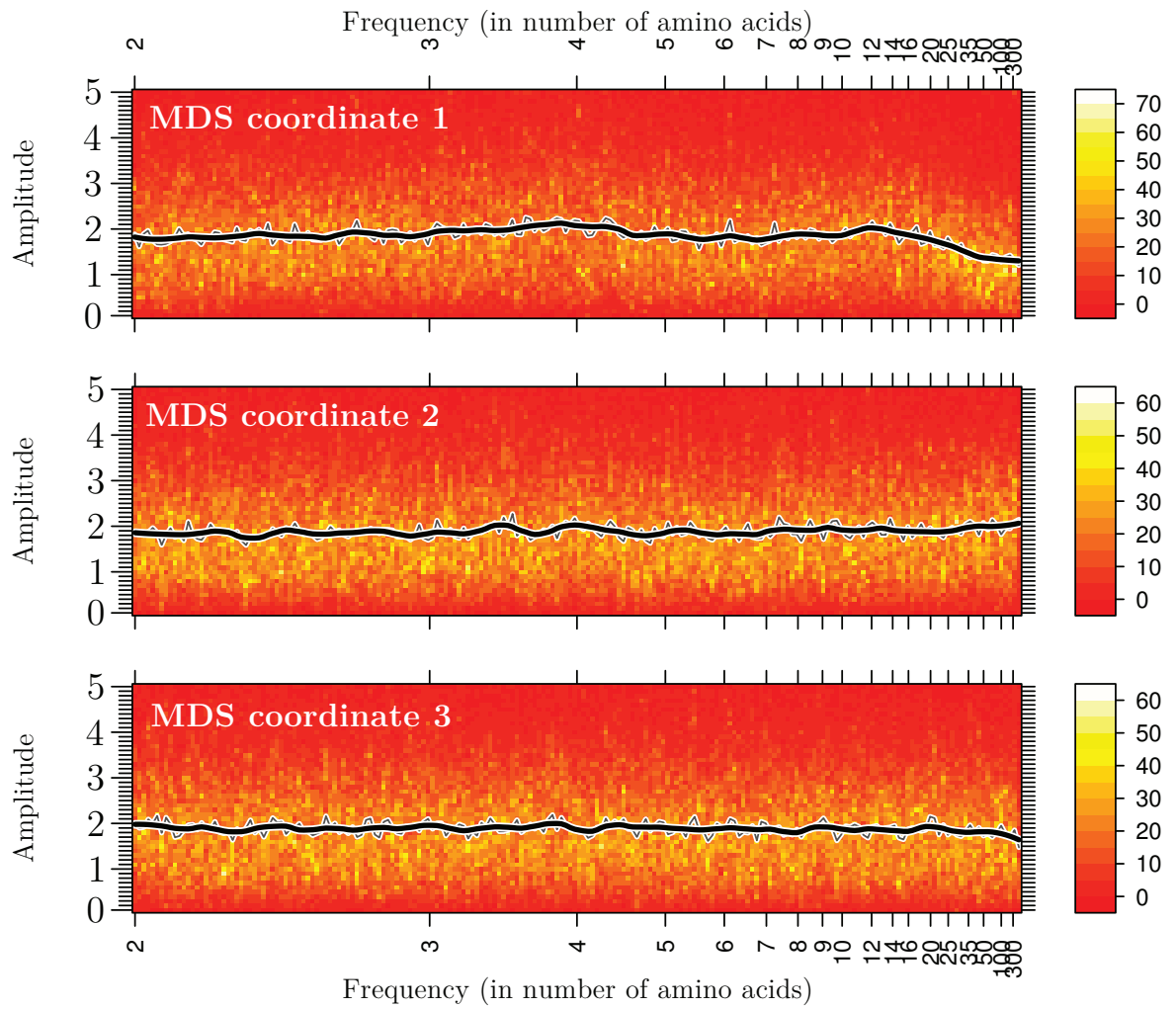


Figure 31: Discrete Fourier transform on SCOP class E. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). In MDS coordinate 1 an elevation is observed for frequencies near $3.5 \sim 4$ amino acids and a drop for > 50 amino acids. It also has an increased amplitude for frequencies of ~ 12 amino acids. MDS coordinates 2 has a small increase in amplitude for frequencies > 50 amino acids. MDS cooridnate 3 has a small drop in amplitude for frequencies < 50 amino acids.

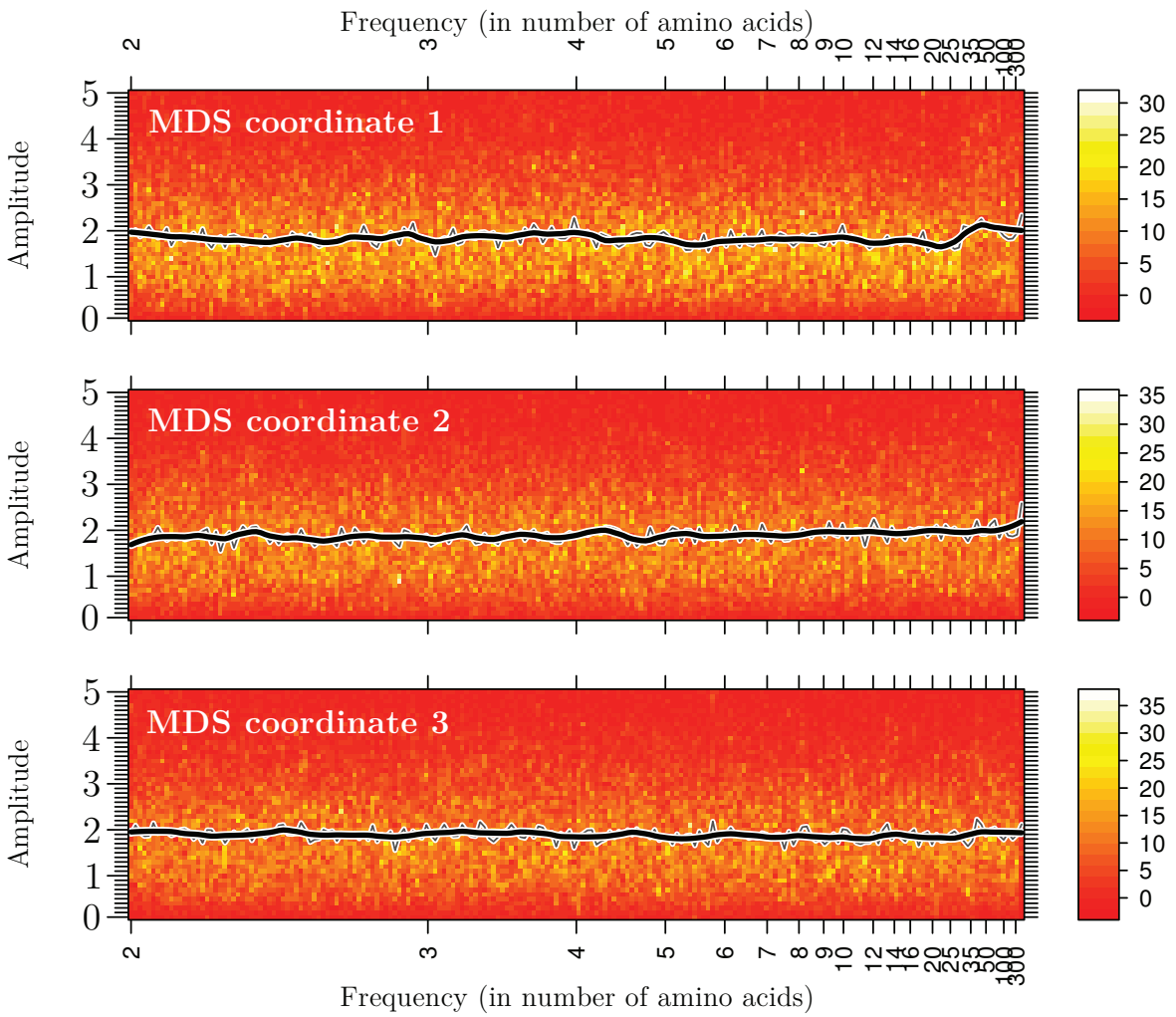


Figure 32: Discrete Fourier transform on SCOP class F. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). In MDS coordinate 1 a vague elevation is observed for frequencies near $3.5 \sim 4$ amino acids. For small frequencies of > 25 amino acids an elevation in amplitude is observed. MDS coordinates 1 and 2 do not show clear patterns.

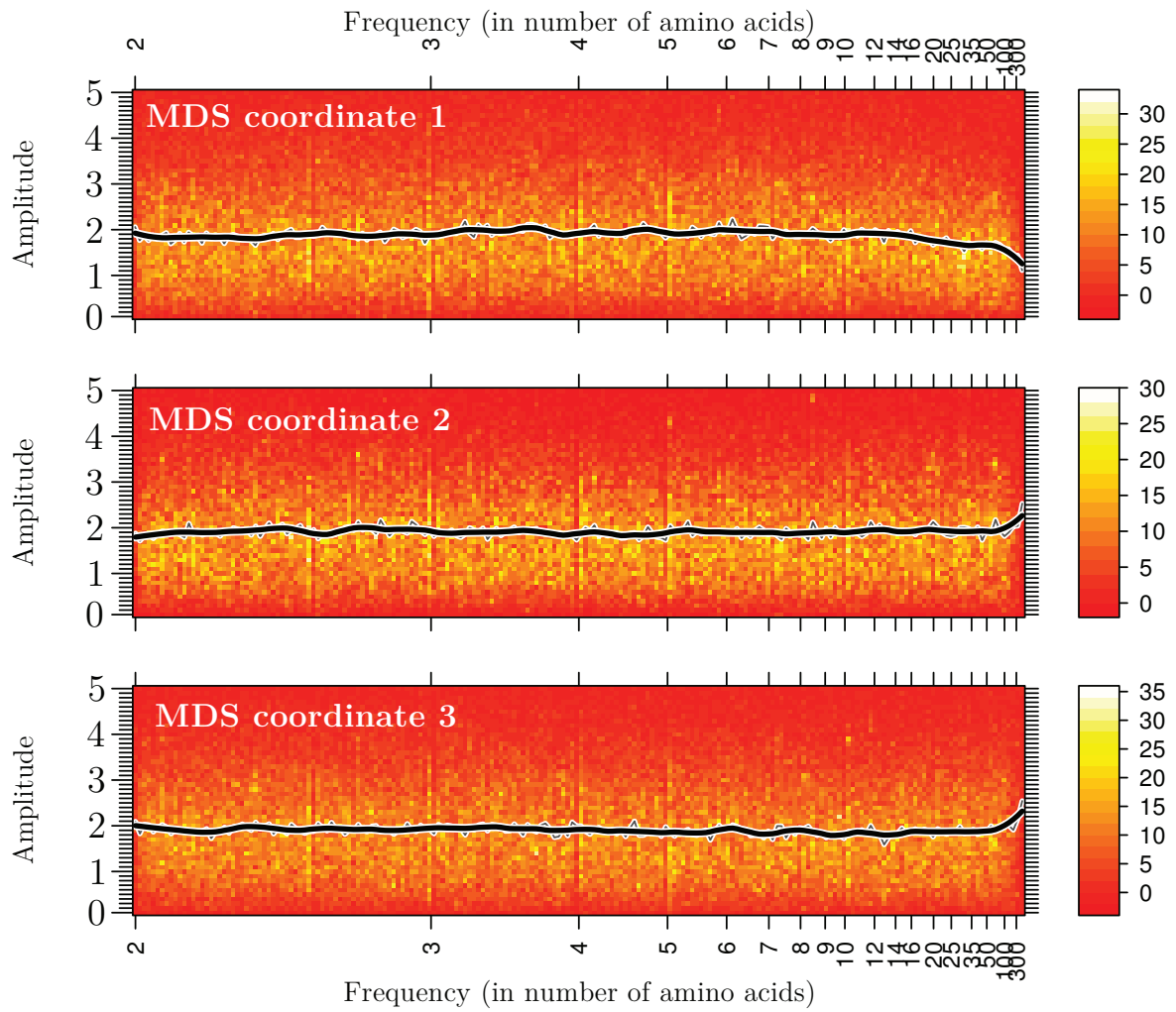


Figure 33: Discrete Fourier transform on SCOP class G. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). MDS cooridnate 1 shows no clear elevations. A drop for low frequencies > 50 amino acids is observed. For MDS coordinate 2 and 3, an elevation for low frequencies > 50 amino acids is observed.

Discrete Fourier transform on entire SCOP dataset

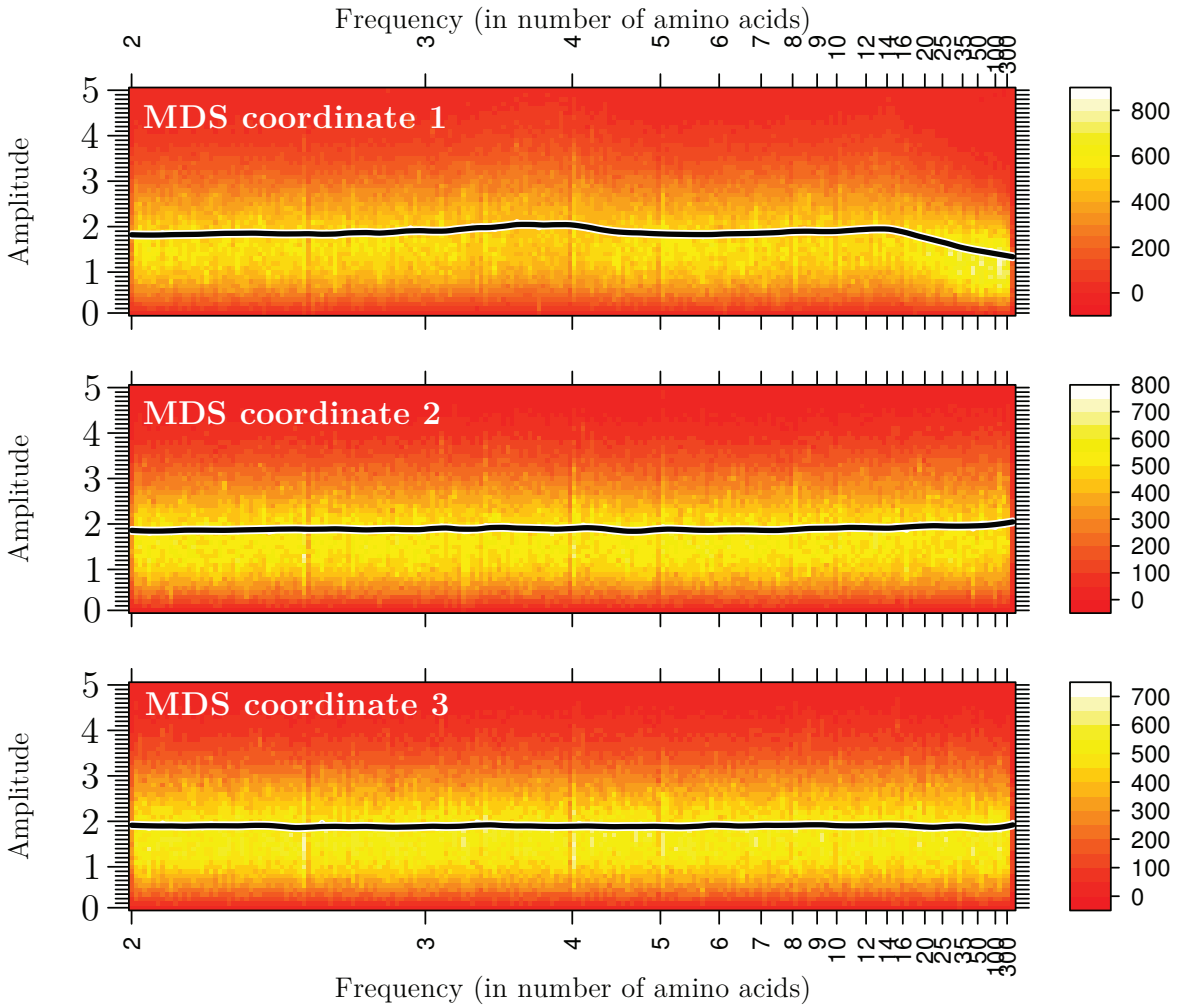


Figure 34: Discrete Fourier transform on entire SCOP set. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). MDS coordinate 1 shows that there is an elevation near $3.5 \sim 4$ amino acids and near ~ 15 amino acids. A drop is observed for frequencies > 20 amino acids. In both MDS coordinate 2 and 3 a tiny, unclear, elevation is observed for low frequencies of > 100 amino acids.

Discrete Fourier transform on Intrinsically Unstructured Proteins

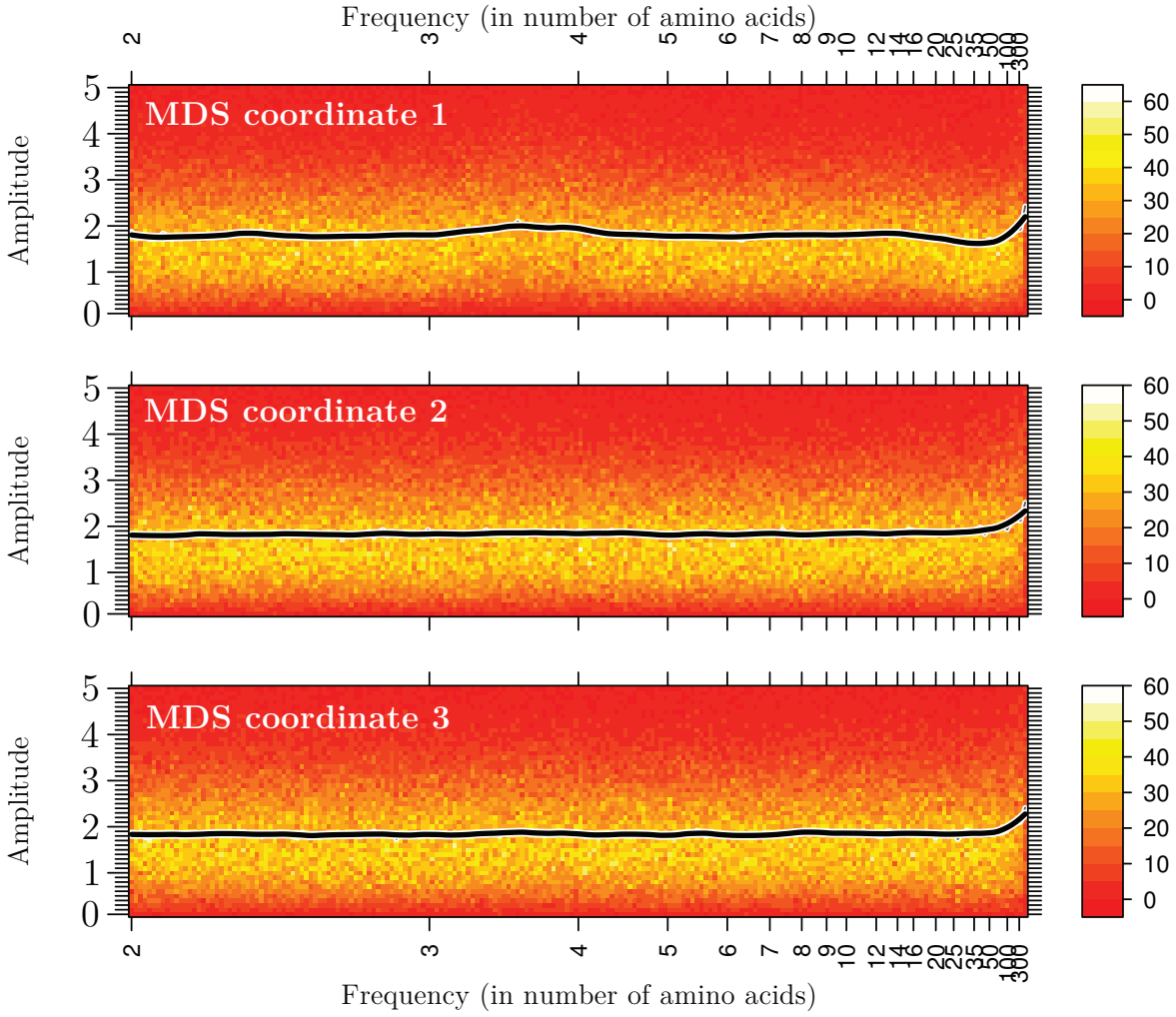


Figure 35: Discrete Fourier transform on Intrinsically Unstructured Proteins. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). MDS coordinate 1 shows that there is an elevation near 3.5 $\sim$ 4 amino acids and near $\sim$ 15 amino acids. Another elevation is observed for frequencies > 50 amino acids. MDS coordinate 2 and 3 also show an elevation for frequencies > 50 amino acids.

Discrete Fourier transform on artificial in silico sequences

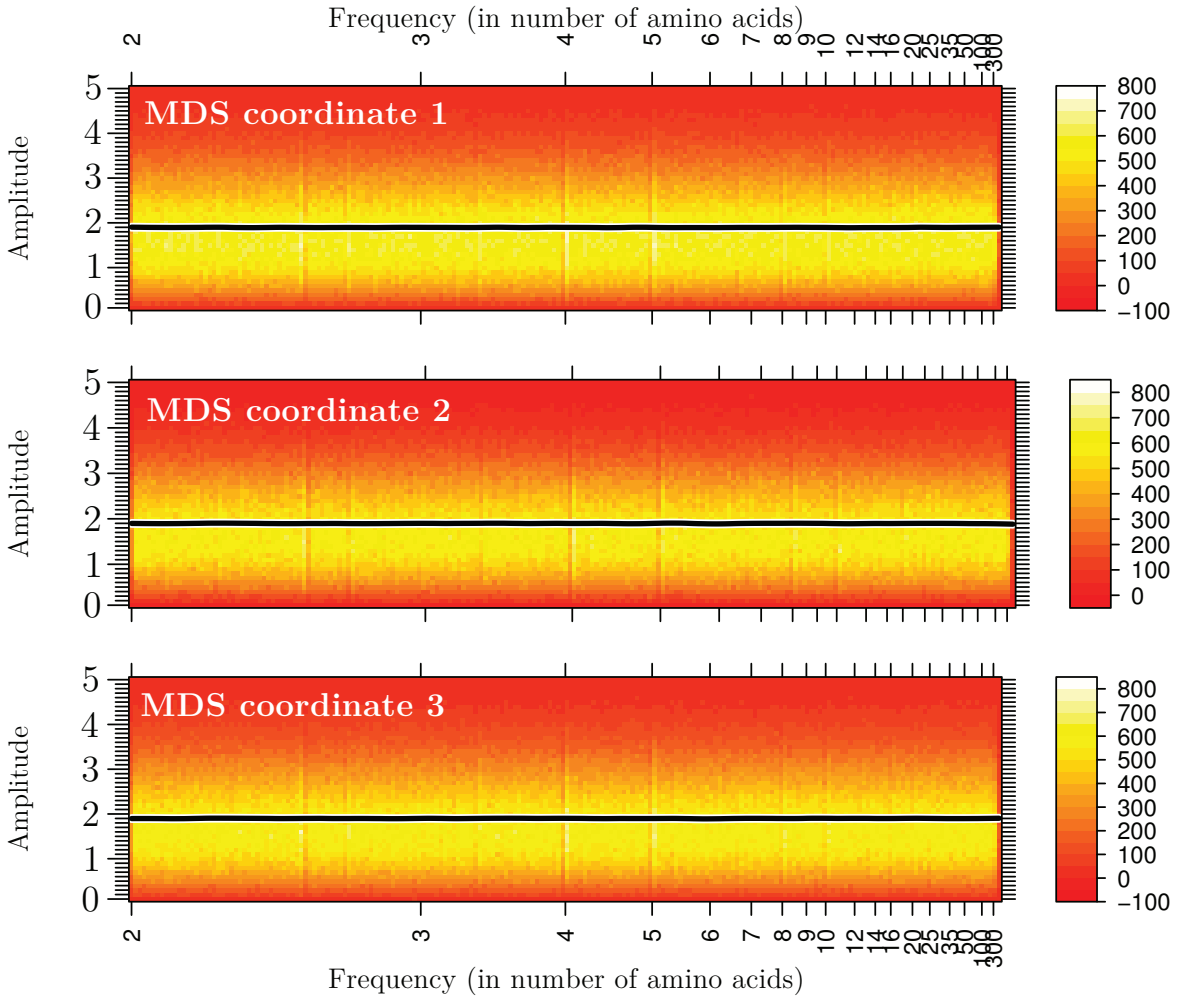


Figure 36: Discrete Fourier transform on artificial in silico sequences. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). No patterns are observed in all three MDS coordinates.

Discrete Fourier transform on artificial in silico sequences (preserved composition)

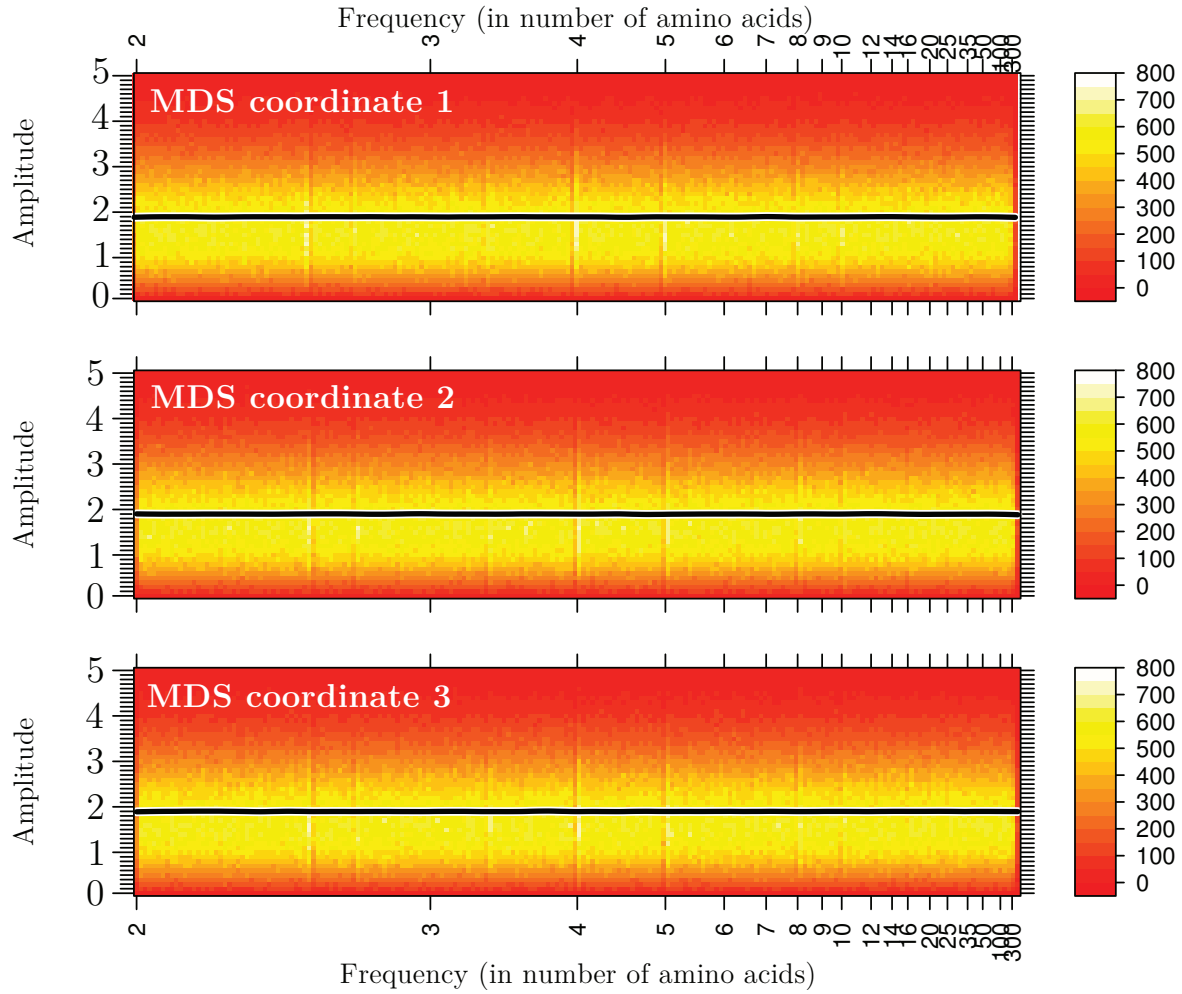


Figure 37: Discrete Fourier transform on artificial in silico sequences with a biologically preserved amino acid composition. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). No patterns are observed in all three MDS coordinates.

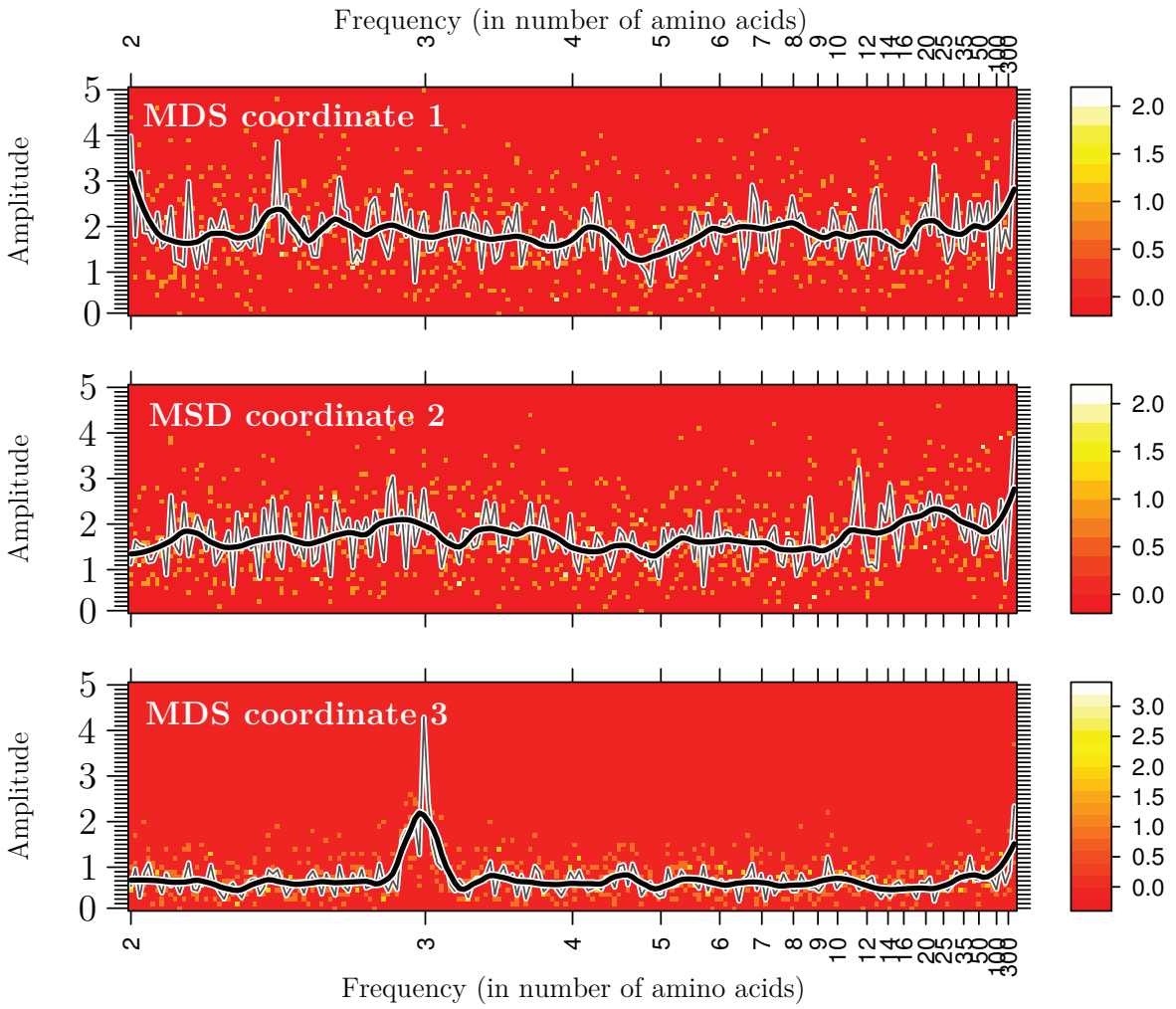


Figure 38: Discrete Fourier transform on collagen alpha-1. The Fourier transform was applied to vectors consisting of values from MDS coordinate 1 (top), MDS coordinate 2 (middle) and MDS coordinate 3 (bottom). MDS coordinate has a fuzzy increase for amplitudes near $15 \sim 35$ amino acids. MDS coordinate 3 shows a clear elevation in amplitude for a frequency of exactly 3 amino acids.

5 Discussion

Section overview In this section a small recapitulation of the work will be given. On top of that, it will be discuss why certain decisions have been made and what could have been done different. Additionally, possible steps for future work will be given.

Sequence Space The construction of sequence space, where sequences represent landmarks and datasets represent maps, has shown to be able to successfully guide a journey through sequence space by asking and answering specific sequenomics related questions. For the construction of the space, sequence classification is a very important preprocessing step because it defines the meaning of directions and locations in sequence space.
Biological sequences have been classified using the SCOP classification scheme. Although this classification is large and extensive, it is also overdue. SCOP also contains unofficial classes, including the *designed proteins*. Using the proposed nomenclature these sequences would have been classified as artificial synthesized. It has to be mentioned that the classification does not distinguish modified from de novo sequences. Only after this sub classification is applied, it should be included in sequence space. They are expected to be useful for future research, because once compared to biological sequences they could indicate potential differences between human protein engineering and natural selection.
Artificial sequences are designed by some human influence. A subset of the artificial sequences are the artificial in silico sequences. They are generated by the computer and have not been synthesized in a laboratory. Three of those datasets have been constructed using a computational randomization scheme, the shuffled SCOP and artificial sequences sampled from length- and uniform or biologically preserved composition distributions. Their purpose is to function as null hypothesis dataset for several analyses. They have shown to be useful since the analyses were able to successfully indicate biological sequence characteristics that differ from artificial. Only the parameters composition and length have been used for sampling sequences. However, it is a challenge for the future to extend this by controlling more parameters, preferably related to sequential order.
Although these in silico sequences are just "blind" guesses by a computer which might include biological sequences just by chance. Therefore they do not necessarily correspond to proteins that lack function or structure. It is necessary to address because it comes with the assumption that they are different from biological sequences and therefore they have been used as null hypothesis group. Of course, the chances of finding biological sequences by these methods are low since the number of possible sequences is huge.
The entropy walk dataset is not spanning a region in sequence space with a chemical meaning. Instead, it has shown to be useful to visualize the boundaries of the space and thereby to explain the concept of a property space. Also, previous research has pointed out that it can be useful for highlighting differences in structure prediction algorithms [46].

Maximally Distant Sequences The maximally distant sequences are examples of in silico artificial sequences, which have been generated using an optimization criterion for a specific property; evolutionary distance.
Using a homogeneous scoring scheme, the algorithm was successfully able to find sequences that have an increasing amount of evolutionary distance to a given set of biological sequences. However, because the scoring method does not correct for likelihood of substitutions, the algorithm gradu-

ally finds sequences that contain more and more rare amino acids, as expected. To some extent the composition seems to become mirrored; the biologically rare amino acids become more common while the common amino acids become more rare. Because it was expected that the maximally distant sequence would be a homo-polymer of the rarest amino acid, `W`, an additional run of the algorithm was initiated with this sequence. The reason behind this is that, if there is no sequence with a larger evolutionary distance than the homo-polymer, it will not find a more distant sequence. In contrast to the hypothesis, the algorithm was able to find more distant sequences. Thus, a homo-polymer sequence consisting of `W` is not the maximally distant sequence.

To overcome the preference for rare amino acids, thus to find maximally distant sequences with a composition similar to biological sequences, a transformed BLOSUM62 scoring function especially designed for the adapted ends-space free alignment has been used. Unexpectedly, the results are comparable to those using the homogeneous scoring matrix. It also found sequence with a preference for rare amino acids. This is unexpected because the alignment algorithm should contain constraints that force to mutate towards biologically preferred compositions. The reason for this is that similar rare amino acids get a higher score by the alignment algorithm and should therefore be mutated in an early stage. The following reason for the results is suggested:

- The arbitrary chosen biological sequences are not the sequences that have been used to built the BLOSUM62 matrix. If the underlying substitution frequencies of the matrix and the biological sequences do not agree with each other, the measurement of the evolutionary distance may become unreliable and undesired mutations can take place.

This suggestion is supported by an additional analysis where the scoring function has been modified such that scores for `W` towards other amino acids became higher, and an analysis where the substitution score for `W` to `W` became higher. The results indicated that the frequency of `W` in the target sequences decreased (data not shown), indicating that the choice of the substitution matrix is indeed determinant for the outcome, and that the proposed transformation of the BLOSUM62 matrix is not sufficient. Therefore it is recommended to dedicate possible future work to the enhancement of the biological scoring method in order to find maximally distant sequences with a biological composition.

The analysis with respect to the algorithms performance indicate that the heuristic repair procedure gives a boost in performance; more distant sequences are found faster. Consequentially, this reduces the search space and induces the possibility to end up in a saddle point. Therefore, the method is classified as a "greedy" search method. The speed analysis also indicated that the number of mutations (reparations) that are applied each iteration are determinant for the performance. A large number seems to increase the performance only in the very early stage while a small number seems to take this performance over in a later stage. It might be valuable to design a hybrid model, where the number of mutations decreases over time. Otherwise, it is recommended to use only 1 mutation per iteration. The reason that a small number of mutations is advantageous is because the smaller the number of mutations per iteration, the more unnecessary calculations can be skipped by the optimization procedure called the scope.

The algorithm did not yet find sequences that seem to be useful for this research. However, a similar algorithm with a different optimization criterion could be helpful for sequenomics research. A candidate could

be a predicted structure, where sequences could be mutated until a preferred predicted structure appears. Another example could be to find the most similar sequence with respect to two sets of sequences. Imagine a family of sequences that have a common ancestor, which have been separated by evolution into two branches, e.g. enzymes that synthesize either product A or B. Then the most similar sequence to these two families, potentially an isolated island, could theoretically be able to synthesize both the products [32]. The strength of the method would be that the algorithm would preserve the functional constraints of both the families of sequences.

Multi-Dimensional Scaling Multi dimensional scaling (MDS) is a family of methods that map a multi dimensional distance matrix into a lower dimensional coordinate system, based on the preservation of distances. The hypothesis has been that if it is applied upon a distance transformation of a substitution matrix, the biologically most important amino acid properties shall form the novel coordinate system. Two solutions that solve the MDS problem have been applied; classical MDS and Sammon's non-linear mapping. Because the latter found coordinates with a lower amount of error, it is assumed to have the lowest amount of loss of information (also for indicating sequence composition) and therefore it seems to be the most suitable candidate for a property space. To get an impression of what these coordinates actually mean and to validate their biological importance, they have been correlated with known biologically important amino acid properties. The analysis indicated that the coordinates have a high correlation with like hydrophobicity and mass, similar to the most important properties as proposed in the amino acid classification scheme (figure 8). This indicates that the method is indeed able to find amino acid properties that are important to be preserved.

Because there are other solutions that solve the MDS problem beyond those that have been used in this research, it is recommended to focus future work on the choice of other, or the improvement of the current MDS methods. This goes in parallel with the choice of the distance transformation. On top of that there are some points of discussion with respect to the usage of the chosen substitution matrix BLOSUM62:

- Previous research has pointed out that although the BLOSUM62 performance for sequence alignment is good, this is probably partially because of programming errors [38]. Therefore this its reliability can be questioned.
- Because at the time it was designed, 1992, clearly not even the majority of all sequences were discovered. It has been built upon 2000 alignment blocks originating from 500 groups of related proteins [13]. This limited number might be outdated and on top of that it is possible that there is a selection for sequences which have studied more intensively by that time (preferential attachment).

Hence, it is also recommended to spend eventual work on the choice of other substitution matrices. Most upcoming analyses rely on the coordinates produced by this method, which addresses its importance. On the other hand, because the correlation with known properties is rather high, it is not expected that novel MDS coordinates will change radically.

Previous research pointed out that with the electron-ion interaction potential (EIIP) [45] it is possible to apply sequence comparison using the wavelet transform [9]. Unexpectedly the correlation analysis did not indicate high correlation with any MDS coordinate. In contrast with their findings [9], this suggests that the EIIP is not a biologically important amino acid property.

Further research should point out what the value of EIIP is and whether or not, using different properties, their proposed method could be enhanced.

Composition The composition of protein sequences is known to be different from uniform random; alignment algorithms incorporated substitution matrices for this reason two decades ago. To find out whether there are classes of sequences with preferences for certain compositions, their average composition has been projected in the property space. In the property space the axes are formed by the multi-dimensional scaling (MDS) coordinates. This results indicate that most biological proteins have a rather similar amino acid composition, with the exception of two classes. The majority of the biological sequences have an enrichment in hydrophobic and smaller amino acids. The exceptional classes are the intrinsically unstructured- and membrane proteins. With respect to the majority of the biological sequences, their average amino acid size does not differ much, but their preference for hydrophobicity does:

- Membrane proteins prefer hydrophilic amino acids.
- Intrinsically unstructured proteins prefer even more hydrophobic amino acids than the majority of the biological sequences.

Since membrane proteins are partially inside the (fatty) cell-membrane, this agrees with their environment. Also, since unstructured regions often include large turns that fall outside the structured domains, it is not surprisingly that their composition prefers water. Even without knowing the exact reasons for compositional preferences, they will most likely give an advantage in artificially designing proteins.
For composition analysis, biological sequence compositions have been compared with uniform amino acid compositions. Because uniform compositions can easily be interpreted, they can explain differences in composition understandably. However, using a uniform distribution as null hypothesis actually relies on the assumption that protein sequences evolve without a preference towards certain amino acids. This assumption is unrealistic because the translation of nucleic acid to amino acid is based on a 3-letter codon system and the ratio of amino acids to number of codons is not uniformly distributed. If a protein coding RNA sequence has uniformly random distributed composition of nucleic acids, the corresponding amino acid distribution is most likely not, since certain amino acids are translated by a higher number of codons. It is plausible that through the dynamic systems in cell, the synthesis ratios of amino acids also differ from the codon rates. The ideal null hypothesis would be the synthesis rates of amino acids, although they probably fluctuate because of specific protein synthesis demand. Thus, instead of using uniform compositions as null hypothesis, it might also be convenient to visualize the composition from other angles like the codon rates or amino acid synthesis rates.

Entropy On top of the known compositional preferences of different classes of protein sequences, an entropy analysis has been performed to get an impression of the redundancy in composition. Because of the non-uniform compositions, it was known beforehand that entropy for biological sequences can not be maximal. Therefore an addition biological entropy has been introduced, assuming that maximal entropy is reached once the sequences has a composition similar to that observed in the entire SCOP dataset.
It is expected that using the biological entropy, biological sequences have a higher entropy than using textual entropy. Also, using sequences with a uniform distribution,

it is to be expected that the textual entropy will be maximal. The results agreed on both the expectations. More interestingly are the differences between the biological artificial in silico sequences. They indicate that using textual entropy, the artificial in silico sequences with uniform compositions have the highest entropy, followed by the artificial in silico sequences with a preserved composition and the lowest textual entropy was found for biological sequences. In contrast, the biological entropy was highest for artificial in silico sequences with a preserved composition, followed by the biological sequences and the lowest amount of biological entropy was found by artificial in silico sequences with a uniform distribution.
On the one hand, the results indicate that the entropy for biological sequences is rather high. To illustrate this, low-entropy sequences like homo-polymer are rarely found. This suggests that the redundancy in chemical characteristics of amino acids is large enough to allow a diverse composition.
On the other hand, it indicates that artificial sequences with a similar overall composition generally have a higher entropy than biological sequences. Thus, although biological sequences do not have a low entropy, they still have more redundancy (duplicate amino acids) than can be expected by chance This suggests the presence of a functional, structural or environmental preference for certain amino acids. Thereby repetitive sequences could play a role in this as well.
However, it has to be mentioned that the previous composition analysis indicated different classes of composition in biological sequences. This ensures that the biological entropy for at least those classes is lower since their overall entropy differs from the majority of the biological sequences.
The following example illustrates how a lower entropy can be interpreted. Assume that the following sequences are biological:

`CBDBADCA`

`GFHEEHFG`

And following sequences are artificial.

`ABCDEFGH`
`BHCEAFGD`

The overall composition of the entire datasets is identical, but the composition of the individual sequences differs.

Linguistic Complexity The linguistic complexity estimates repetition at the subsequence level of sequences, able to indicate repetition at the level of sub-sequences.
The goal was to find these differences between artificial from biological sequences. Therefore their distributions have been compared and the results indicated that biological sequences have a lower linguistic complexity than artificial sequences, although the differences are marginal.
It has to be addressed that the meaning of the linguistic complexity scale is ambiguous, and relies on assumptions that make it unreliable to compare linguistic complexity values of different size sequences with each other. Imagine a sequence of length $n = 2$: `AA`. If the linguistic complexity for a sequence of length n is calculated, the number of observed sub-sequences of length n (which is always 1) will be divided through the number of possible sequences (also 1). Then for a length $n - 1$, the number of observed sequences will be either 1 or 2, divided by possible the number of subsequences, 2, and so on. In the example, since `A` occurs twice, the LC will be 2/3. In contrast, a similar sequence of length $n = 3$: `AAA` has a LC of 3/6. This illustrates that the length of a sequence gives a different meaning to a linguistic complexity value. The underlying problem is that the linguistic complexity does not correct for the expected number of sub-sequences but for the number of possible sub-sequences. Consequently, sub-sequences of different lengths

are weighted equally. Thus, to compare linguistic complexity that correspond to sequences of different lengths, it must be corrected for the expected number of subsequences, which would also solve the problem of finding extremely high values only.

The second part of the analysis tried to overcome this problem without adjustments of the algorithm. The likelihood of a linguistic complexity corresponding to a sequence was estimated by comparing it to a distribution of linguistic complexity values of sequences with similar length and composition. It indicated that biological sequences are often significantly linguistically less complex than the artificial sequences.

This means that there biological sequences are more redundant in terms of subsequences; thus that there is more repetition in sub-sequences than can be expected by chances. A possible reason could be the presence of folds or with a repetitive nature, like β-sheet for instance.

As expected, the usage of likelihood estimations indicated more clearly that there are differences between biological and artificial sequences than using the linguistic complexity values of sequences with different lengths. A remark on the applied methodology is that the probabilities have been calculated using a normal distribution. Although the linguistic complexity distributions have some characteristics similar to the normal distribution (bell shaped and symmetrical to some extend) it surely differs. Therefore the probabilities do not precisely answer the asked questions and could be improved by using more elaborated statistics.

A more general remark on the usage of linguistic complexity in protein sequences is that, as the name already indicates, it only considers the textual context of amino acids instead of the biological. Whereas the linguistic complexity uses counts of subsequences, a biological context could be included by using sequence alignment scores between sub-sequences (e.g. like global sequence alignment using a BLOSUM62 scoring matrix) .

Local Entropy Variance Complex figures showed they typically do not have a maximal entropy. Instead, they show a variance in local entropy. This is necessary for the formation of structures at multiple levels. Because similar behaviour can be expected for protein sequences, the local entropy variance has been estimated. It finds the variation in entropy for all possible subsequences, called windows, of one arbitrary chosen length. Since two different estimations for entropy have been used (textual and biological), also two estimations for LEV were designed.

The estimation of the optimal window size has been done by finding the differences in means of the distributions. The results indicated that window sizes of $\alpha = 5$ or 6 amino acids are optimal for the artificial- and biological LEV respectively. In contrast with the entropy analysis, the LEV analysis did not indicate clear differences in using a biological- or textual entropy. A small remark on the method of the optimal window size estimations is that this method is does not takes local class variances into account. A potential solution is a function as proposed by [10], which finds the ratio of between-group to within-group sums of squares. Although the analysis method will become more accurate, the results in combination with visual interpretation of the data are so straightforward that it is unlikely that the window sizes shall differ.

Using an optimal window size, the LEV shows that there are clearly differences between artificial and biological sequences. More specifically, a majority of the biological sequences have a larger LEV than artificial sequences. This suggests that in a literal context the sequences seem to have a more complex nature than shuffled sequences and seem to distinguish themselves from pure chaos, like the complexity images given in

figure 11. It is also an indication for the presence of an internal structure inside biological sequences.
Biological and artificial sequences that do have a maximal entropy, also have a comparable LEV. This is because the entropy of the entire sequence is maximal and because of that the entropy of every sub-sequence has to be maximal as well. If every subsequence has a maximal entropy its variance will be minimal because all entropy values will be (nearly) identical.
Initially, a function was proposed which estimates the likelihood of finding a LEV value compared to shuffled sequences with a similar composition. The problem with this method is that the smaller the window becomes, the smaller the number of possible outcomes for entropy values becomes. Using entropy values, especially of small windows, is therefore tricky to apply statistics on since the number of possible outcomes becomes limited. Therefore the method was left out. Here a solution will be suggested. For this analysis the number of times that the LEV of a target sequence is smaller than the average of the distribution of shuffled sequences, should be counted. Using a χ^2-test, the likelihood of how often the LEV for a sequence is lower than for other sequences with a similar composition can be estimated.
In summary, using the right window size there are indeed differences in LEV between biological and artificial sequences. The majority of the biological sequences have a higher LEV than artificial sequences, suggesting that biological sequences have a more complex nature.
The most important things that last are:

- Finding out what regions in sequences (with respect to protein structure and function) contribute to a larger entropy variance in biological sequences.

- Finding out what types of sequences are enriched in LEV and which are not. This also includes finding what proteins have a maximal entropy (like chaos in images).

Local Variance Variance Because entropy does not take physico-chemical amino acid properties into account, its biological relevance can be questioned. The following examples will illustrate this. It is possible to find a sequence with a high entropy while in a chemical context all amino acids in the sequence are identical. From the perspective of this chemical property it can be convenient to classify this example sequence as having a low diversity, whereas entropy would have classified it as having a high diversity. The local variance variance (LVV) has been introduced with the goal to serve as a method similar to local entropy variance, but able to take the biological context into account. The main difference between LVV en LEV is the usage of variance on a vector with chemical properties instead of entropy on a string. The variance gives high values for high diversity low values for low diversity, which is to some extend similar to entropy. Because variances do not result in a symmetrical distribution, an additional normalization scheme assuming a χ^2-distribution has been used and this method is referred to as SLVV.
Because the first three MDS coordinates are assumed to be the biologically most relevant properties, they have been used for local variance estimation. Accordingly, sequences have been translated into vectors of these coordinates. Such vectors can thus exist of maximally 20 unique values, which is rather limited for variance estimation.
The first step in the LVV analysis is the estimation of the optimal window size α for all three MDS coordinates, by finding the separation in the distributions of biological and artificial sequences. The corresponding window sizes are $\alpha = 6$ amino acids for MDS coordinate 1 and 2, and $\alpha = 3$ amino acids for MDS coordinate 3. The best separation

has been found in the following order, using: MDS coordinate 1 > MDS coordinate 3 > MDS coordinate 2. That MDS coordinate 3 is able to distinguish better than MDS coordinate 2 is surprising, because MDS coordinate 2 is expected to be biologically more important. A possible explanation could be because MDS coordinate 2 correlates with mass, protein sequences do not prefer as much variance in mass as they do in MDS coordinate 3.

There is room for improvement for the optimal window size estimation method. Similarly to LEV, the selection method described by [10] which finds the ratio of between-group to within-group sums of squares, could enhance the current method of finding the optimal window size since it also takes local group variances into account. However, the optimal window size is expected to stay similar.

The SLVV procedure contains an interim step where local variances are converted using a χ^2-distribution into standard normal distribution observations. This conversion uses a limited number of observations, and therefore some accuracy will be lost in this process. However, despite this conversion, only SLVV is able to distinguish artificial from biological sequences. The unsymmetrical implementation of LVV is unreliable because it relies on wrong assumptions.

A 3D projection of the data indicated that the SLVV analysis on the different MDS coordinates is uncorrelated. In contrast to LEV, the strength of SLVV is that local property variability can independently be explained at multiple scales (amino acid properties) at different resolutions (window sizes). Taken this together, both the LEV and LVV indicate that protein sequences have more local variation than by chance. This can be interpreted, similarly to complex figures, as evidence for internal complexity.

To get an indication of the likelihood of finding a SLVV value with respect to other sequences with a similar composition, a probability has been calculated using the F-test and the Bartlett test. However, whereas the artificial sequences are expected to have a uniform distribution, their average probability polarizes towards 0 as the α parameters increases. Because those artificial sequences have been compared with other artificial sequences, this polarization is an unexpected outcome. The following causes for the contradictory results have been taken into account:

- The local variations are compared with 10 times more local variations, because the sequence is shuffled 10 times. It could be the difference in vector length has some influence on the distribution estimation. However, it is assumed that the higher the number of observations the better the estimation of a distribution would be. Still, it could be that the implementation failed for some technical reason.

- The usage of overlapping windows affects the independence of individual observations (the variances per windows); a high variance for a large windows also ensures a (rather) high variance for the overlapping neighbour window. This hypothesis is supported by the fact that the larger the windows become the more this effect takes place; the larger the windows become the more dependent the observations become since their overlap becomes larger.

Because the second reason is most plausible, it is assumed that this influences the tests. A possible solution would be to use non-overlapping windows for both the LEV and LVV analysis. The drawback will be that the number of windows will reduce, which affects the accuracy of the distribution estimation. Further research on LVV should focus on:

- The vectors used for (S)LVV are the

MDS coordinates because they are expected to be biologically relevant. However, this does not mean that other properties can not contain strong local variances. Instead, to enhance (S)LVV analysis it is recommended to spend eventual future work to the choice of other amino acid properties or the optimization of the current.

- Because the LEV indicated strong separation results, it could be convenient to translate the MDS properties in to probabilities and apply a corresponding LEV on it. This could indicate whether the usage of entropy outperforms variance, still preserving the chemical context of the amino acids.

In brief, although there is room for improvement the analysis indicated that biological sequences contain often higher LEV values. This evidence for complexity of a sequences at different resolutions and scales is uncorrelated. Currently no explanation can be given on why the optimal window sizes are 6 and 3 amino acids. It might be possible that, because α-helices have a turnover rate of 3.6 amino acids and β-sheets near 15, the optimal window size of 6 is an composited average. Be aware, the addressed problem with overlapping windows might also play a role in the optimal window size.
It is recommended to spend future work on the analysis of what regions in sequence contribute to these local variances in order to explain the complexity. Another important direction for future work would be finding out which of the sequences have higher local variances (e.g. multi domain proteins?) and which have not (e.g. fibrous proteins?).

Autocorrelation The previous methods have been focussing mainly on irregularities in sequences. In contrast, autocorrelations focusses on the regularities in sequences. Using vectors that replace sequences with the MDS coordinates, the autocorrelation has been applied on most datasets.
The results indicated that at first glance, for all MDS coordinates, the overall autocorrelation for all lag values is low. This means that entire sequences do not have a common repetition of a specific length in the MDS coordinates. The fibrous protein Collagen is herein exceptional. Using MDS coordinate 3 it has a high correlation for all lag values that are a multitude of 3. Collagen is a rod shaped protein that forms a so called triple helix. These helices interconnect with a rate of exactly 3 amino acids. It is plausible that the high correlation is a result of the presence of the triple-helix in Collagen.
If the results are examined in more depth, specific patterns become visible. For all classes of sequences that contain α-helices, there is an increased correlation for lag values of 3 and 4 amino acids using in MDS coordinate 1. This finding is comparable to the turnover-rate of an α-helix which is 3.6 amino acids. Notice that autocorrelation can only find correlation patterns with rounded lag values.
Similarly, but less intense, the β-sheet sequences have an increased correlation for lag values of $\sim$ 15 amino acids, also found in MDS coordinate 1. The length of a β-sheets is typically 3 to 10 amino acids long, with an average of 6 amino acids. Since two sheets and connecting amino acids are essential to form a turnover, their rates are indeed close to 15 amino acids.
The average autocorrelation values, in both the analyses of the α-helix and β-sheet sequences, behave like a sinusoid function over the increasing lag-axis. This is probably because the underlying patterns in the MDS coordinates, caused by the α-helices and β-sheets, have a repetitive nature and the lag values go in parallel with the phase shift.
Because the autocorrelation values are low, even for α-helix and β-sheet sequences, additional statistics could indicate the degree of correlation with respect to artificial se-

quences more precisely. Therefore the probabilities of the distribution of autocorrelations using a particular lag value, for all sequences that contain α-helices, could be compared with artificial sequences with a similar composition.

Although the presence of α-helices and β-sheets could be observed, it was hardly visible. Only the correlation for a homogeneous structured protein was able to showed a high degree of correlation. An explanation for this is that the the former include typically more complex proteins since they usually have multiple structures (on top of the helices and sheets). As effect of this, the correlation patterns of the α-helices or β-sheets are faded out by the presence of amino acids that contribute to other structures. Thus, autocorrelation seems to be a decent approach for low complexity, or homogeneous structured proteins but is not ideal for complex structured protein sequences.

Discrete Fourier Transform To overcome the shortcomings in the autocorrelation analysis, like having the ability to find patterns at unrounded frequencies and the ability to decompose patterns at multiple resolutions, the Fourier transformation was applied using vectors that replace sequences with the MDS coordinates for most datasets.

The results indicated that datasets with sequences that contain α-helices, have an elevated amplitude for frequencies near $3.5 \sim 4$ amino acids in MDS coordinate 1. These findings agree with the actual turnover rate of 3.6 amino acids. Similarly, datasets with sequences that contain β-sheets, have an elevated amplitude for frequencies near ~ 15 amino acids in MDS coordinate 1. These findings agree with the average turnover rates that are expected to be near 15 amino acids. Thus, the discrete Fourier transform is able to find patterns in the property space of sequences that correspond to known structures in proteins.

Most biological sequences have a drop in amplitude for frequencies that are larger than ~ 25 amino acids in MDS coordinate 1. A suggested explanation is that most structures and folds have a limited size; smaller than $\sim$ 25 amino acids for instance, because of the proteins folded 3D structure. Folding reduces the physical length of proteins, thus the corresponding structures and fold should be limited to those surroundings. The suggestion is supported by the contrary observation, that intrinsically unstructured proteins have a higher amplitude for those frequencies. Because they are generally unstructured, long, random coils, or the regions that interconnect domains, it is plausible that they are not restricted to any size. Therefore it is possible to find patterns that span a large number of amino acids.

However, analysis on MDS coordinate 2 and 3 shows that low frequencies > 50 amino acids often have an increased amplitude. No clear explanation for this can be given and is therefore an interesting lead for future research.

For fibrous protein Collagen, analysis on MDS coordinate 3 has indicated an enormous increase in the amplitude for a frequency that corresponds to exactly 3 amino acids. This agrees precisely with the triple helix structure that interconnects amino acids every 3 amino acids. MDS coordinates 1 and 2 previously received an identity because of their correlation with the physico-chemical properties hydrophobicity and mass respectively. MDS coordinate 3 received the identity of *triple helix* because it seems to be important for constraints in substitutions involved in triple helix regions in protein sequences like Collagen. This supported by the following findings:

- The discrete Fourier transform analysis indicates high amplitudes for MDS coordinate 3 at a frequency that corresponds to three amino acids.

- The autocorrelation indicates high cor-

relations in MDS coordinate 3 for all lag values that are a multitude of three amino acids.

- The SLVV has an optimal window size in MDS coordinate 3 of three amino acids, where the other optimal window sizes are 6 amino acids long.

Although the results are promising, there is room for improvement and eventual work in the future:

- Although the MDS coordinates have shown to be able to indicate structure related patterns, improvement of these properties and novel properties might be able to find novel or more clear patterns.

- The Fourier transformed data was normalized because of length- and composition differences. The amplitudes have been divided through the σ^2. This normalization can be improved by taking the exact length and composition into account.

- Whereas the discrete Fourier transform transforms the data to sinuses and cosines, other transformation are able to transform into other functions. For the round structure of a helix the sine and cosine might be expected, but it is plausible that β-sheets for instance, might emerge with a different transformation more clearly.

- It has to be addressed that the heatmaps of the frequency domain give frequencies in number of amino acids on a reciprocal scale. Therefore it is difficult to visualise amplitude differences for lower frequencies. However, if the data is transformed to a linear scale, the number of observations per frequency drop on a reciprocal scale. A solution for this problem could be a topic of future research.

6 Conclusion

The journey has demonstrated the usefulness of the abstract idea of sequence space in providing new ways to understand protein structure and evolution. Locations in sequence space can be demarcated and navigated. Furthermore, the properties of sequences and structures in those locations can be shown to vary in systematic ways from place to place. These properties include the presence or absence of structural patterns in corresponding proteins while others have indicated differences in complexity of information density. This shows that the construction of a sequence space as being a map or atlas of the geography of protein folding and function. The diversity in sizes of the datasets used herein indicate that for certain biological questions low resolution maps are sufficient, while for other questions it is necessary to have high resolution samples. In turn, the hills and valleys of sequence space can be exploited when designing novel protein sequences for new applications. Because only a fraction of the immense space has yet been explored, more journeys have to follow in order to refine the atlas of protein landscapes.

References

[1] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. Journal of molecular biology, 215(3):403–10, October 1990.

[2] A J Barrett. Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme Nomenclature. Recommendations 1992. Supplement 4: corrections and additions (1997). European Journal of Biochemistry, 250(1):1–1, November 1997.

[3] M. S. Bartlett. Properties of Sufficiency and Statistical Tests. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 160(901):268–282, May 1937.

[4] Rita Berisio, Luigi Vitagliano, Lelio Mazzarella, and Adriana Zagari. Crystal structure of the collagen triple helix model. pages 262–270, 2002.

[5] H Kaspar Binz, Patrick Amstutz, and Andreas Plückthun. Engineering novel binding proteins from nonimmunoglobulin domains. Nature biotechnology, 23(10):1257–68, October 2005.

[6] GEP Box, GM Jenkins, and GC Reinsel. Time series analysis: forecasting and control, volume 3rd editio. Prentice Hall, Englewood Cliffs, NJ, 3 edition, 1993.

[7] Dawn J Brooks, Jacques R Fresco, Arthur M Lesk, and Mona Singh. Evolution of amino acid frequencies in proteins over deep time: inferred order of introduction of amino acids into the genetic code. Molecular biology and evolution, 19(10):1645–55, October 2002.

[8] In-Geol Choi and Sung-Hou Kim. Evolution of protein structural classes and protein sequence families. Proceedings of the National Academy of Sciences of the United States of America, 103(38):14056–61, September 2006.

[9] Chafia Hejase de Trad, Qiang Fang, and Irena Cosic. Protein sequence comparison based on the wavelet transform approach. Protein engineering, 15(3):193–203, March 2002.

[10] Sandrine Dudoit, Jane Fridlyand, and P Terence. Comparison of Discrimination Methods for the Classi ' cation of Tumors Using Gene Expression Data. 97(457), 2002.

[11] Sean R Eddy. Where did the BLOSUM62 alignment score matrix come from? Nature biotechnology, 22(8):1035–6, August 2004.

[12] D M Engelman, T A Steitz, and A Goldman. Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins. Annual review of biophysics and biophysical chemistry, 15:321–53, January 1986.

[13] S Henikoff and J G Henikoff. Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences of the United States of America, 89(22):10915–9, November 1992.

[14] T P Hopp and K R Woods. Prediction of protein antigenic determinants from amino acid sequences. Proceedings of the National Academy of Sciences of the United States of America, 78(6):3824–8, June 1981.

[15] Kristopher Josephson, Matthew C T Hartman, and Jack W Szostak. Ribosomal synthesis of unnatural peptides. Journal of the American Chemical Society, 127(33):11727–35, August 2005.

[16] M Juan, Antonio Falc, and Javier Lorenzo. N-Dimensional Mapping of Amino Acid Substitution Matrices (Unpublished). Technical report, Instituto de Sistemas Inteligentes. IUSIANI Univ. Las Palmas de Gran Canaria, Spain.

[17] Shuichi Kawashima, Piotr Pokarowski, Maria Pokarowska, Andrzej Kolinski, Toshiaki Katayama, and Minoru Kanehisa. AAindex: amino acid index database, progress report 2008. Nucleic acids research, 36(Database issue):D202–5, January 2008.

[18] A D Keefe and J W Szostak. Functional proteins from a random-sequence library. Nature, 410(6829):715–8, April 2001.

[19] Keith Knight. Mathematical statistics, volume 1. Chapman & Hall/CRC Texts in Statistical Science, Toronto, 2000.

[20] J Kyte and R F Doolittle. A simple method for displaying the hydropathic character of a protein. Journal of molecular biology, 157(1):105–32, May 1982.

[21] D. Lipman and W. Pearson. Rapid and sensitive protein similarity searches. Science, 227(4693):1435–1441, March 1985.

[22] LM Liu, GB Hudak, GEP Box, ME Muller, and GC Tiao. Forecasting and time series analysis using the SCA statistical system, volume 1. 1992.

[23] S Minton, MD Johnston, AB Philips, and P Laird. Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. Proceedings of the eighth ..., pages 17–24, 1990.

[24] S Minton, MD Johnston, AB Philips, and P Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. Artificial Intelligence, 1992.

[25] A G Murzin, S E Brenner, T Hubbard, and C Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of molecular biology, 247(4):536–40, April 1995.

[26] G. S. Nadiger, N. V. Bhat, and M. R. Padhye. Investigation of amino acid composition in the crystalline region of silk fibroin. Journal of Applied Polymer Science, 30(1):221–225, January 1985.

[27] S B Needleman and C D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3):443–53, March 1970.

[28] Marco Punta, Penny C Coggill, Ruth Y Eberhardt, Jaina Mistry, John Tate, Chris Boursnell, Ningze Pang, Kristoffer Forslund, Goran Ceric, Jody Clements, Andreas Heger, Liisa Holm, Erik L L Sonnhammer, Sean R Eddy, Alex Bateman, and Robert D Finn. The Pfam protein families database. Nucleic acids research, 40(Database issue):D290–301, January 2012.

[29] J.W. Sammon. A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers, C-18(5):401–409, May 1969.

[30] E a Schultes, P T Hraber, and T H LaBean. Estimating the contributions of selection and self-organization in RNA secondary structure. Journal of molecular evolution, 49(1):76–83, July 1999.

[31] Erik Schultes, Peter T. Hraber, and Thomas H. LaBean. A parameterization of RNA sequence space. Complexity, 4(4):61–71, March 1999.

[32] Erik A. Schultes and David P. Bartel. One sequence, two ribozymes: implications for the emergence of new ribozyme folds. Science, 289(5478):448–452, July 2000.

[33] Erik A Schultes, Peter T Hraber, and Thomas H Labean. No Molecule Is an Island: Molecular Evolution and the Study of Sequence Space. Natural Computing Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[34] Erik a Schultes, Alexander Spasic, Udayan Mohanty, and David P Bartel. Compact and ordered collapse of randomly generated RNA sequences. Nature structural & molecular biology, 12(12):1130–6, December 2005.

[35] Ron Shamir, Doron Yaary, and Ami Pelded. Scribe of Lecture 2 : November 1 , 2001. 2001.

[36] Megan Sickmeier, Justin a Hamilton, Tanguy LeGall, Vladimir Vacic, Marc S Cortese, Agnes Tantos, Beata Szabo, Peter Tompa, Jake Chen, Vladimir N Uversky, Zoran Obradovic, and a Keith Dunker. DisProt: the Database of Disordered Proteins. Nucleic acids research, 35(Database issue):D786–93, January 2007.

[37] T F Smith and M S Waterman. Identification of common molecular subsequences. Journal of molecular biology, 147(1):195–7, March 1981.

[38] Mark P Styczynski, Kyle L Jensen, Isidore Rigoutsos, and Gregory Stephanopoulos. BLOSUM62 miscalculations improve search performance. Nature biotechnology, 26(3):274–5, March 2008.

[39] William Ramsay Taylor. The classification of amino acid conservation. Journal of Theoretical Biology, 119(2):205–218, March 1986.

[40] Ian M Tomlinson. Next-generation protein drugs. Nature biotechnology, 22(5):521–2, May 2004.

[41] Peter Tompa. Intrinsically disordered proteins: a 10-year recap. Trends in biochemical sciences, 37(12):509–16, December 2012.

[42] Warren S. Torgerson. Multidimensional scaling: I. Theory and method. Psychometrika, 17(4):401–419, December 1952.

[43] Olga G Troyanskaya, Ora Arbell, Yair Koren, Gad M Landau, and Alexander Bolshoy. Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity. Bioinformatics, 18(5):679–688, May 2002.

[44] G E Tusnády and I Simon. Principles governing amino acid composition of integral membrane proteins: application to topology prediction. Journal of molecular biology, 283(2):489–506, October 1998.

[45] V. Veljković and I. Slavić. Simple General-Model Pseudopotential. Physical Review Letters, 29(2):105–107, July 1972.

[46] Ward L. Weistra. A first look at the Protein Sequence Space. (September), 2012.

[47] J Zhang. Protein-length distributions for the three domains of life. Trends in genetics : TIG, 16(3):107–9, March 2000.

[48] Karel Zimmermann and Jean-François Gibrat. Amino acid "little Big Bang": representing amino acid substitution matrices as dot products of Euclidian vectors. BMC bioinformatics, 11:4, January 2010.

7 Appendix

7.1 Collagen Alpha-1(I)

Gene: COL1A1, RefSeq sequence: OI4NP_000079.

MFSFVDLRLL LLLAATALLT HGQEEGQVEG QDEDIPPITC VQNGLRYHDR DVWKPEPCRI
CVCDNGKVLC DDVICDETKN CPGAEVPEGE CCPVCPDGSE SPTDQETTGV EGPKGDTGPR
GPRGPAGPPG RDGIPGQPGL PGPPGPPGPP GPPGLGGNFA PQLSYGYDEK STGGISVPGP
MGPSGPRGLP GPPGAPGPQG FQGPPGEPGE PGASGPMGPR GPPGPPGKNG DDGEAGKPGR
PGERGPPGPQ GARGLPGTAG LPGMKGHRGF SGLDGAKGDA GPAGPKGEPG SPGENGAPGQ
MGPRGLPGER GRPGAPGPAG ARGNDGATGA AGPPGPTGPA GPPGFPGAVG AKGEAGPQGP
RGSEGPQGVR GEPGPPGPAG AAGPAGNPGA DGQPGAKGAN GAPGIAGAPG FPGARGPSGP
QGPGGPPGPK GNSGEPGAPG SKGDTGAKGE PGPVGVQGPP GPAGEEGKRG ARGEPGPTGL
PGPPGERGGP GSRGFPGADG VAGPKGPAGE RGSPGPAGPK GSPGEAGRPG EAGLPGAKGL
TGSPGSPGPD GKTGPPGPAG QDGRPGPPGP PGARGQAGVM GFPGPKGAAG EPGKAGERGV
PGPPGAVGPA GKDGEAGAQG PPGPAGPAGE RGEQGPAGSP GFQGLPGPAG PPGEAGKPGE
QGVPGDLGAP GPSGARGERG FPGERGVQGP PGPAGPRGAN GAPGNDGAKG DAGAPGAPGS
QGAPGLQGMP GERGAAGLPG PKGDRGDAGP KGADGSPGKD GVRGLTGPIG PPGPAGAPGD
KGESGPSGPA GPTGARGAPG DRGEPGPPGP AGFAGPPGAD GQPGAKGEPG DAGAKGDAGP
PGPAGPAGPP GPIGNVGAPG AKGARGSAGP PGATGFPGAA GRVGPPGPSG NAGPPGPPGP
AGKEGGKGPR GETGPAGRPG EVGPPGPPGP AGEKGSPGAD GPAGAPGTPG PQGIAGQRGV
VGLPGQRGER GFPGLPGPSG EPGKQGPSGA SGERGPPGPM GPPGLAGPPG ESGREGAPGA
EGSPGRDGSP GAKGDRGETG PAGPPGAPGA PGAPGPVGPA GKSGDRGETG PAGPAGPVGP
VGARGPAGPQ GPRGDKGETG EQGDRGIKGH RGFSGLQGPP GPPGSPGEQG PSGASGPAGP
RGPPGSAGAP GKDGLNGLPG PIGPPGPRGR TGDAGPVGPP GPPGPPGPPG PPSAGFDFSF
LPQPPQEKAH DGGRYYRADD ANVVRDRDLE VDTTLKSLSQ QIENIRSPEG SRKNPARTCR
DLKMCHSDWK SGEYWIDPNQ GCNLDAIKVF CNMETGETCV YPTQPSVAQK NWYISKNPKD
KRHVWFGESM TDGFQFEYGG QGSDPADVAI QLTFLRLMST EASQNITYHC KNSVAYMDQQ
TGNLKKALLL QGSNEIEIRA EGNSRFTYSV TVDGCTSHTG AWGKTVIEYK TTKTSRLPII
DVAPLDVGAP DQEFGFDVGP VCFL

7.2 Alignment Method

Sequence alignment methods often use a dynamic programming algorithm. Such algorithms align two sequences, A of length m and B of length n with each other. Initially, matrix $\mathbf{F}$ of length $m+1 \times n+1$ is constructed for finding an optimal alignment score. The function $S(\bullet)$ scores the similarity of individual chars a and b. A scoring scheme is given in equation 14.

$$S(a,b) = \begin{cases} -1, & \text{if } \mathbf{gap} \\ 2, & \text{if } a = b \\ -1, & \text{if } a \neq b \end{cases} \tag{14}$$

The following three branches of the algorithms are often used (modifications of these algorithms exist, e.g. in order to reduce the complexity [1,21]):

7.2.1 Global Alignment

The global alignment solves the following problem: *find the maximum similarity between two sequences* [27,35]. The global alignment searches for the best overall alignment of two sequences. It assumes that two sequences originate from exactly the same ancestor and consequentially identical sequence lengths are expected. It fills $\mathbf{F}$ as follows:

for $0 \leq i \leq m$ **do**
  $F_{i,0} \leftarrow i \cdot S(\mathbf{gap})$
end for
for $0 \leq j \leq n$ **do**
  $F_{0,j} \leftarrow j \cdot S(\mathbf{gap})$
end for
for $1 \leq i \leq m$ **do**
  for $1 \leq j \leq n$ **do**
    Match $\leftarrow F_{i-1,j-1} + S(A_i, B_j)$
    Delete $\leftarrow F_{i-1,j} + S(\mathbf{gap})$
    Insert $\leftarrow F_{i,j-1} + S(\mathbf{gap})$
    $F_{i,j} \leftarrow max(\text{Match}, \text{Insert}, \text{Delete})$
  end for
end for

The optimal alignment score is found at $F_{i',j'} = F_{m,n}$. The corresponding alignment can be found by tracing the route back that was used in the filling phase, from $F_{i',j'}$ until $F_{0,0}$.

7.2.2 Local Alignment

The local alignment solves the following problem: *find the maximum similarity between a sub-sequence of a sequence and a sub-sequence of another sequence* [35,37]. The sequences are expected to have a shared similar sub-sequence, or one sequence is expected to be the entire sub-sequence of the other. Therefore it assumes that two sequences are not entirely similar in the sense that length differences are not taken into account. It fills $\mathbf{F}$ as follows:

for $0 \leq i \leq m$ **do**
  $F_{i,0} \leftarrow 0$
end for

for $0 \leq j \leq n$ **do**
  $F_{0,j} \leftarrow 0$
end for
for $1 \leq i \leq m$ **do**
  for $1 \leq j \leq n$ **do**
    Match $\leftarrow F_{i-1,j-1} + S(A_i, B_j)$
    Delete $\leftarrow F_{i-1,j} + S(\mathbf{gap})$
    Insert $\leftarrow F_{i,j-1} + S(\mathbf{gap})$
    $F_{i,j} \leftarrow max(\text{Match}, \text{Insert}, \text{Delete}, 0)$
  end for
end for

The alignment score is found at $F_{i',j'} = \max_{1 \leq i \leq m, 1 \leq j \leq n} F_{i,j}$. The optimal alignment can be found by tracing the route back used in the filling phase, from $F_{i',j'}$, as long as $F_{i,j} \geq 0$.

7.2.3 End-space Free Alignment

The end-space free alignment solves the following problem: *find the maximum similarity between two prefixes and suffixes two sequences* [35]. It assumes that the sequences are both derived from one longer broken up ancestor, like products of shotgun sequencing. Because broken pieces can be each others sub-sequence or each others pre- or suffix, no equal lengths are assumed. It fills **F** as follows:

for $0 \leq i \leq m$ **do**
  $F_{i,0} \leftarrow 0$
end for
for $0 \leq j \leq n$ **do**
  $F_{0,j} \leftarrow 0$
end for
for $1 \leq i \leq m$ **do**
  for $1 \leq j \leq n$ **do**
    Match $\leftarrow F_{i-1,j-1} + S(A_i, B_j)$
    Delete $\leftarrow F_{i-1,j} + S(\mathbf{gap})$
    Insert $\leftarrow F_{i,j-1} + S(\mathbf{gap})$
    $F_{i,j} \leftarrow max(\text{Match}, \text{Insert}, \text{Delete})$
  end for
end for

The alignment score is found at $F_{i',j'} = \max(\max_{1 \leq i \leq m,n} F_{i,j}, \max_{m,1 \leq j \leq n} F_{i,j})$. The optimal alignment can be found by tracing the route back used in the filling phase, from $F_{i',j'}$, until $F_{0,0}$.

7.2.4 Choice

The choice of the alignment method is determinant for the results. To illustrate the possible obstacles, alignments for four example sets of two sequences are given in figure 39. A description of the actual sequence similarity of examples 1 upto 4:

1. The sequences have many amino acids in common, spread out over the entire two sequences. Different sized gaps are found in between.

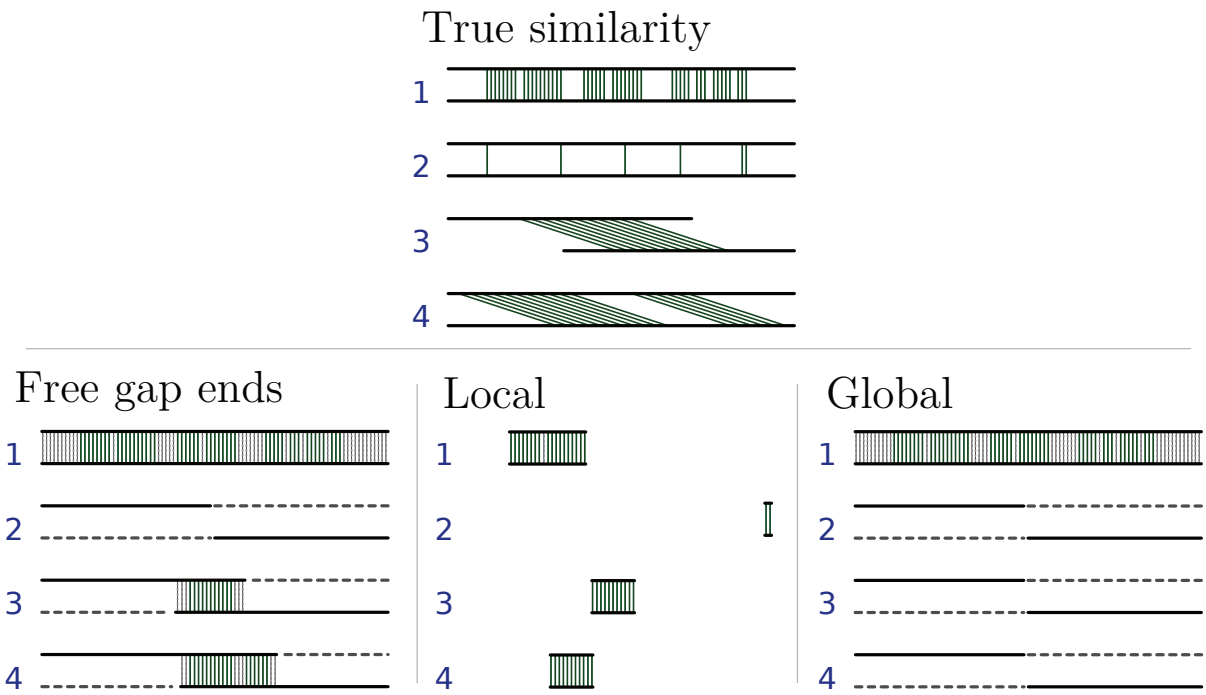


Figure 39: A schematic illustration of the results of three types of alignments using the scoring function in equation 14. **Top:** of the figure is the actual similarity between four sets of sequences illustrated: the black lines indicate the sequences, the green lines indicate identical amino acids; dissimilar amino acids are not indicated as such. **Bottom:** the alignments are illustrated. Matches are indicated with vertical green lines, mismatches are indicated with vertical dashed gray lines and gaps are indicated with horizontal gray lines.

2. The sequences have only a few similar amino acids, spread out over the whole length.

3. The sequences contain one similar sub-sequence. Its start position differs per sequence and the regions jutting out on both sides have no similarity. In a biological context, the two proteins could share a functional motif.

4. The sequences contain two similar sub-sequences both of different length. The start position of the similar sub-sequence differs per sequence. In a biological context, sequences could share two functional motifs.

Global Alignment The global alignment only finds similarity in the first example. In the other examples it does not find any similarity. The reason for that is that global alignments expect a high overall similarity in order to align, because it assumes that two sequences come from the same ancestor. Sequences of which the overall similarity is low will de-align (indicate no similarity), like examples 2, 3 and 4.
This alignment method is sensitive for non-similar regions and herewith it impedes the detection of local similarity, which is essential for the estimation of evolutionary distance. A solution which overcomes this is using a mismatch penalty of 0 instead of -1. Accordingly, non-similar regions will not contribute to the score any more; only gaps and matches will be taken into account.
In examples 3 and 4 the regions that jut out also decrease the alignment score because they introduce gaps instead of mismatches. The problem behind this is that the algorithm

assumes similar sequence lengths and therefore penalizes for continuous initial (at the beginning of the alignment) and terminal (at the end of the alignment) gaps that are only caused by length differences. This causes a so called length bias, meaning that if two sequences differ in length, that difference times the gap penalty will be subtracted from the score. The more two compared sequences differ in length, the more drastic this effect will be. A solution for the length-bias is to correct the score afterwards by subtracting the absolute length bias: score = score$-(|\text{length}_1 - \text{length}_2| \cdot$ gap penalty), where $|\bullet|$ is the absolute value operator.

Local Alignment In contrast, the local alignment method is able to indicate similarity in all four examples. However, in all examples that contain multiple similar sub-sequences (1, 2 and 4) only the longest similar sub-sequence is aligned. This is because it assumes that one sequence can be a sub-sequence of the other. Therefore the trace-back function starts at the maximum value of the **F** matrix (instead of $F_{m,n}$), and stops whenever $F_{i,j}$ is smaller than 0 (instead of $F_{0,0}$) [35, 37]. Similarly to the global alignment, using a mismatch score of 0 instead of -1 can overcome this.
The second problem in local alignments is that initial and terminal gaps will always be removed from the alignment. Accordingly, incomplete alignments can appear which make it hard to estimate the locations of matches and mismatches. This problem can not easily be solved without adapting characteristics from the global alignment.

End-space Free Alignment The end-space free alignment is able to find the similarity for examples 1, 3 and 4. It is some kind of hybrid type of alignment: it initializes like local- (using 0 instead of gap penalties), and fills like global alignments (using a trace back that goes back to $F_{0,0}$). Using a mismatch penalty of 0 allows to indicate the similarity for example 2. This method does not suffer from the listed problems of the previous two methods; the length bias (global alignment) and the incomplete alignments (local alignment). Additionally, this algorithms allows to find similarity at the prefix- and suffix level of sequences. Because of these advantages, the end-space free alignment has been chosen for estimating the evolutionary distance for the maximally distant sequence problem.

7.2.5 Number Of Solutions

The alignment algorithms can find multiple optimal solutions for a single alignment. If an alignment is applied only to obtain the alignment score it is not a great deal; all optimal solutions have the same score. However, if the actual alignment does matter, it can become problematic because the algorithm finds only one of all possible solutions. For the maximally distant sequence problem a sequence is repaired (mutated) based on the alignment with the purpose to decrease the alignment score. It is possible that by repairing based on 1 of the multiple optimal solutions, only a subset of the optimal solutions will decease in alignment score. Then the alignment with the repaired sequence will have an identical alignment score, but a lower number of optimal solutions. The cause of this may be that, if the number of optimal solutions is not taken into account in the maximally distant sequence algorithm, the repaired sequence is not considered as an optimization of the previous. The algorithm might because of this end up in a saddle point. Thus, not only the alignment score but also the number of optimal solutions is important for estimating the optimization criterion of the maximally distant sequence.

This problem can explained most easily using a global alignment example. By gradually mutating the sequence, the importance of the number of optimal solutions becomes visible. Consider sequence x to be the target sequence which has to be repaired in order to become more distant to the constant library sequence L_i. The repair process is as follows:

`WWWWWWWWWW` (sequence x; target)
`AAAWAAAWAA` (sequence L_i; constant biological library sequences element i)

After applying the global alignment (mismatch: 0, match: 2, gap: -1) the following alignment is obtained, with a corresponding alignment score of 4 (2·match):

```
WWWWWWWWWW
:::|:::|::
AAAWAAAWAA
```

To reduce the similarity score of this alignment and keeping sequence L_i constant, sequence x has to be mutated such that x_4 and x_8 will become something else than `W`. If the following replacements take place: $x_4 \leftarrow$ `X` and $x_8 \leftarrow$ `X`, sequence x becomes:

`WWWXWWWXWW` (sequence x)

Aligning this novel sequence x with sequence L_i, results in 17 optimal alignment. Their corresponding score is 2 (2·match, 2·gap). A random subset of 6 of the solutions is given below:

```
WW-WXWWWXWW  W-WWXWWWXWW  -WWWXWWWXWW  WWWXWWWXWW-  WWWXWWWXWW-  WWWXWWWXWW-
:::|:::|:::  :::|:::|:::  :::|:::|:::  ::::|:::|::  ::::|:::|::  ::::|:::|::
AAAWAAAWAA-  AAAWAAAWAA-  AAAWAAAWAA-  -AAAWAAAWAA  A-AAWAAAWAA  AA-AWAAAWAA
```

The novel sequence x can again be repaired at the matches in one of these 17 alignments such that either:

- $x_5 \leftarrow$`X` and $x_9 \leftarrow$`X`, such that $x^I \leftarrow$`WWWXXWWXXW`
- $x_3 \leftarrow$`X` and $x_7 \leftarrow$`X`, such that $x^{II} \leftarrow$`WWXXWWXXWW`

Intuitively, the novel sequence should be more distant to L_i than the previous. But, the alignment tells us that (no matter whether x^I or x^{II} is chosen,) the novel sequences have an identical alignment score as the previous. The score is still 2 (2·match, 2·gap):

```
W-WWXXWWXXW   -WWWXXWWXXW
:::|:::|:::   :::|:::|:::
AAAWAAAWAA-   AAAWAAAW-AA
```

To indicate that minimizing the alignment score is insufficient, notice that according to the scores the novel sequences do not optimize the previous. However, the number of optimal solutions has become smaller (x^I has 9 optimal solutions, x^{II} 8 optimal solutions). To illustrate why its evolutionary distance has become larger, notice that the sequence

has become two mutations closer to sequence XXXXXXXXXX which is maximally distant to sequence L_i.
Thus, the optimization objective should not only be minimizing the alignment score, but also minimizing the number of optimal solutions per alignment score. The alignment method should thus be modified such that it is also able to calculate the number of optimal solutions. Therefore an alternative implementation of the end-space free alignment has been designed. It uses an additional matrix **P** of similar size as **F** to find the number of possible optimal solutions. The used alignment algorithm is defined in equations 15, 16 and 17.

$$S(a,b) = \begin{cases} 2, & \text{if } a = b \\ 0, & \text{if } a \neq b \\ -1, & \text{if } \textbf{gap} \end{cases} \tag{15}$$

$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + w(a_i, b_j) \\ \begin{cases} F_{i,j-1} + 0 & \text{if } i = m \\ F_{i,j-1} + S(a_i, -) & \text{otherwise} \end{cases} \\ \begin{cases} F_{i-1,j} + 0 & \text{if } j = n \\ F_{i-1,j} + S(-, b_i) & \text{otherwise} \end{cases} \end{cases} \tag{16}$$

$$P_{i,j} = \sum \begin{cases} P_{i-1,j-1} & \text{if } F_{i,j} = F_{i-1,j-1} + w(a_i, b_j) \\ P_{i,j-1} & \text{if } F_{i,j} = \begin{cases} F_{i,j-1} + 0 & \text{if } i = m \\ F_{i,j-1} + S(a_i, -) & \text{otherwise} \end{cases} \\ P_{i-1,j} & \text{if } F_{i,j} = \begin{cases} F_{i-1,j} + 0 & \text{if } j = n \\ F_{i-1,j} + S(-, b_j) & \text{otherwise} \end{cases} \end{cases} \tag{17}$$

for $0 \leq i \leq m$ **do**
    $F(i,0) \leftarrow 0$
    $P(i,0) \leftarrow 1$
end for
for $0 \leq j \leq n$ **do**
    $F(0,j) \leftarrow 0$
    $P(0,j) \leftarrow 1$
end for
for $1 \leq i \leq m$ **do**
    for $1 \leq j \leq n$ **do**
        $F_{i,j} \leftarrow$ (equation 16)
        $P_{i,j} \leftarrow$ (equation 17)
    end for
end for

In contrast to the original implementation of the end-space free alignment as proposed by [35], the trace back algorithm of the global alignment is required. Thus, the optimal route from $F_{m,n}$ until $F_{0,0}$ leads to the alignment. The alignment score can be found at $F_{m,n}$ and the number of optimal solutions at point $P_{m,n}$. The reason for the alternative implementation is its convenience with respect to calculating the number of optimal solutions.

7.2.6 Biological Scoring

Because the homogeneous scoring function given in 15 is not biologically relevant, an additional scoring function using the BLOSUM62 matrix has been proposed. The objection is that the BLOSUM62 matrix does not meet the constraints of the given substitution matrix. In the given substitution matrix the mismatch penalty has explicitly been set to 0 to overcome certain problems. For proper alignment a biologically relevant substitution matrix may also not have values lower than 0. Since values in the BLOSUM62 matrix larger than 0 represent substitution that occurs more often than by chance, it can be considered to be a match. Therefore, all values in the BLOSUM62 matrix that are 0 or higher, should in the novel matrix be 1 or higher. As result, the BLOSUM62 matrix, $\mathbf{B}$, has been transformed into $\mathbf{C}$ as given in equation 18, where a and b represent the substituted amino acids. This transformed matrix has been included in the biological scoring function $S_B(a, b)$ as given in equation 19.

$$C_{a,b} = \frac{B_{a,b}}{-\min \mathbf{B}} + 1 \tag{18}$$

$$S_B(a, b) = \begin{cases} -1, & \text{if } \textbf{gap} \\ C_{a,b}, & \text{if } \textit{not } \textbf{gap} \end{cases} \tag{19}$$

7.3 Maximally Distant Sequences

Using a heterogeneous scoring, the analysis of the most distant sequence gradually iterated towards the following sequence (runtime is 6 days):

```
MQHWMCCKCE MWWWYMCYFT WHWCLCRCCC FMWEHCMMMM WMTCYPQMWA WEMIWWWRPR
HCHWRICWWW KHKWCNWWCC FWICMWWWEW PKHHCFTWQD FFGPHHVHQQ HMMHMWCWWW
RCWWWCWARH HSWWMMCWMC FCQWQCCWCW TMCHTCNWCC CMMWWWWCWR CWWCHQCHCC
CCDHCCMRQM MHMQQCWNWC CSWYMCHHQI CWHHCRMRPN WCCQQRKWMM RMWWCWWDDD
MWYMFYHFFC CCCCCCSMMM MENNMCCYCT CTYWYEWVCC E
```

Using a heterogeneous scoring, the analysis of the most distant sequence gradually iterated towards the following sequence (runtime is 10 days):

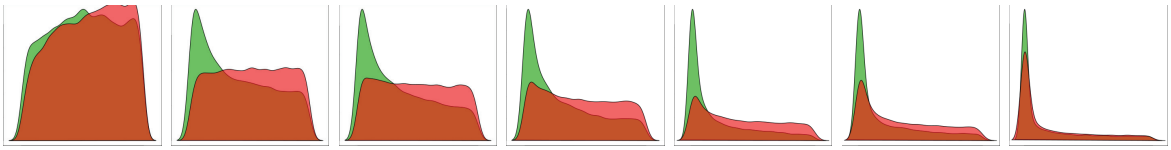
```
WCWWWCWWCC WYWFWWWWYY WWWWWWWWCC CCCWCCCWWW WWWWWCWCYC CCCCCFCCYC
CCCCCCWCCC CCCCCCCCCC CCCWWCWAPW CWWHHHRIHK FYWWWWWWWW WWYLWWEWWH
HRRRRPYRQW WWWWWWWWYY HHHHGCCCKC CFWWWWWCCC CCWWCCCCCC CCCCVCCCWC
WWWWWWWWWW WWWWCWCCCV CCPPPCPPPC CPCCKPPPPP PPPCCCPPPP PCQRNRRRRR
RTPPRPRPPH HHPPPPPPPP PPPPPPHCCC CCCCWWCWWW W
```

7.4 BLOSUM62 Matrix

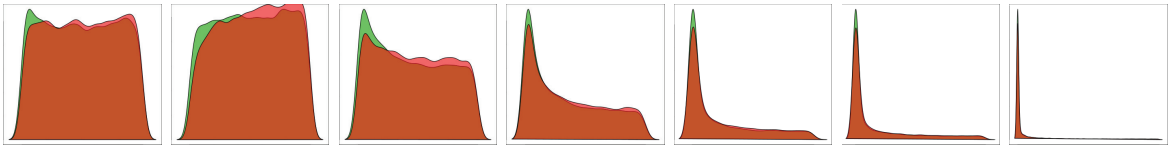
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	1.9646	-0.7068	-0.7654	-0.8767	-0.2043	-0.402	-0.4319	0.0798	-0.8126	-0.6609	-0.7323	-0.367	-0.4676	-1.105	-0.4071	0.5579	-0.0227	-1.2634	-0.882	-0.0947
R	-0.7068	2.7367	-0.2199	-0.8029	-1.6946	0.4914	-0.0577	-1.1521	-0.1249	-1.4951	-1.0773	1.0544	-0.6836	-1.3932	-1.0543	-0.3824	-0.5612	-1.3397	-0.8469	-1.2513
N	-0.7654	-0.2199	2.8266	0.6358	-1.3299	8e-04	-0.134	-0.2114	0.2892	-1.6085	-1.6895	-0.0895	-1.0754	-1.497	-1.0002	0.3005	-0.023	-1.848	-1.0409	-1.4382
D	-0.8767	-0.8029	0.6358	2.8871	-1.73	-0.1567	0.7552	-0.6568	-0.5595	-1.5606	-1.8028	-0.3509	-1.5293	-1.7419	-0.7401	-0.1305	-0.5254	-2.1072	-1.5325	-1.5713
C	-0.2043	-1.6946	-1.3299	-1.73	4.2911	-1.4509	-1.8062	-1.2502	-1.4939	-0.6138	-0.6387	-1.5182	-0.7099	-1.1877	-1.3976	-0.4375	-0.4333	-1.1521	-1.2036	-0.4038
Q	-0.402	0.4914	8e-04	-0.1567	-1.4509	2.6426	0.9273	-0.8926	0.224	-1.3848	-1.067	0.6363	-0.2105	-1.5822	-0.641	-0.0506	-0.3377	-0.9732	-0.7105	-1.0992
E	-0.4319	-0.0577	-0.134	0.7552	-1.8062	0.9273	2.4514	-1.0551	-0.0588	-1.5972	-1.4232	0.3877	-0.999	-1.5962	-0.5581	-0.0735	-0.4316	-1.4177	-1.0102	-1.2211
G	0.0798	-1.1521	-0.2114	-0.6568	-1.2502	-0.8926	-1.0551	2.7816	-1.0204	-1.8624	-1.8135	-0.764	-1.3383	-1.5537	-1.0668	-0.1462	-0.7877	-1.2457	-1.5199	-1.5694
H	-0.8126	-0.1249	0.2892	-0.5595	-1.4939	0.224	-0.0588	-1.0204	3.7555	-1.6158	-1.3934	-0.3605	-0.7756	-0.6171	-1.0805	-0.4408	-0.8429	-1.1711	0.8463	-1.5587
I	-0.6609	-1.4951	-1.6085	-1.5606	-0.6138	-1.3848	-1.5972	-1.8624	-1.6158	1.9993	0.7608	-1.3351	0.5634	-0.0804	-1.3783	-1.1741	-0.3588	-1.2903	-0.6657	1.2735
L	-0.7323	-1.0773	-1.6895	-1.8028	-0.6387	-1.067	-1.4232	-1.8135	-1.3934	0.7608	1.9247	-1.2234	0.9959	0.2074	-1.43	-1.2213	-0.5987	-0.8159	-0.531	0.3942
K	-0.367	1.0544	-0.0895	-0.3509	-1.5182	0.6363	0.3877	-0.764	-0.3605	-1.3351	-1.2234	2.2523	-0.6774	-1.5393	-0.5068	-0.1017	-0.3348	-1.4782	-0.91	-1.1312
M	-0.4676	-0.6836	-1.0754	-1.5293	-0.7099	-0.2105	-0.999	-1.3383	-0.7756	0.5634	0.9959	-0.6774	2.6963	0.0063	-1.2382	-0.7404	-0.3331	-0.7124	-0.4974	0.3436
F	-1.105	-1.3932	-1.497	-1.7419	-1.1877	-1.5822	-1.5962	-1.5537	-0.6171	-0.0804	0.2074	-1.5393	0.0063	3.023	-1.7986	-1.1845	-1.0538	0.4588	1.4696	-0.4245
P	-0.4071	-1.0543	-1.0002	-0.7401	-1.3976	-0.641	-0.5581	-1.0668	-1.0805	-1.3783	-1.43	-0.5068	-1.2382	-1.7986	3.6823	-0.4045	-0.5376	-1.8271	-1.4599	-1.1744
S	0.5579	-0.3824	0.3005	-0.1305	-0.4375	-0.0506	-0.0735	-0.1462	-0.4408	-1.1741	-1.2213	-0.1017	-0.7404	-1.1845	-0.4045	1.9422	0.6906	-1.3759	-0.8429	-0.8231
T	-0.0227	-0.5612	-0.023	-0.5254	-0.4333	-0.3377	-0.4316	-0.7877	-0.8429	-0.3588	-0.5987	-0.3348	-0.3331	-1.0538	-0.5376	0.6906	2.2727	-1.2145	-0.803	-0.0278
W	-1.2634	-1.3397	-1.848	-2.1072	-1.1521	-0.9732	-1.4177	-1.2457	-1.1711	-1.2903	-0.8159	-1.4782	-0.7124	0.4588	-1.8271	-1.3759	-1.2145	5.252	1.0771	-1.4171
Y	-0.882	-0.8469	-1.0409	-1.5325	-1.2036	-0.7105	-1.0102	-1.5199	0.8463	-0.6657	-0.531	-0.91	-0.4974	1.4696	-1.4599	-0.8429	-0.803	1.0771	3.2975	-0.6038
V	-0.0947	-1.2513	-1.4382	-1.5713	-0.4038	-1.0992	-1.2211	-1.5694	-1.5587	1.2735	0.3942	-1.1312	0.3436	-0.4245	-1.1744	-0.8231	-0.0278	-1.4171	-0.6038	1.8845

Table 6: The BLOSUM62 matrix (unrounded).

7.5 LVV Likelihood



(a) Results for the F-test; the larger the windows become (the more to the right) the lower the probabilities become. Since this happens for both the biological (green) as the artificial sequences (red), the test does not answer precisely the biological question.



(b) Results for the Bartlett-test; the larger the windows become (the more to the right) the lower the probabilities become. Since this happens for both the biological (green) as the artificial sequences (red), the test does not answer precisely the biological question.

Figure 40: For both the F-test (top) and Bartlett-test (bottom) the likelihood of a LVV value with respect to the distribution of LVV values for shuffled sequences has been calculated. The density distributions of the corresponding probabilities for biological (green) and artificial sequences (red) is illustrated. The figures represent from left to right the distributions using window sizes in the ranges 2,3,4,6,12,20 and 50. On the x-axis the probability is drawn in the domain $0 \leq p \leq 1$, on the y-axis the relative probability density.

Acknowledgements

I would like to thank the following people for supervision, support, IT facilitation and bright ideas:

- Dr. Erik Schultes
- Dr. Jeroen Laros
- Dr. Herman van Haagen
- Reinout van Schouwen
- Dr. Michael Emmerich
- Dr. Alexander Bolshoy

Similarly, I would like to thank all involved people, of course including my parents and girlfriend, but also Prof. Dr. Guido Jenster for helping and supporting my studies. Also I would like to thank Dr. Alexander Bolshoy for sending me the code with the optimized implementation of linguistic complexity. Last but not least, I want to thank Dr. Erik Schultes for putting a lot of effort in me and never letting me down when I was asking him favours. Once you read this, good luck with future sequenomics studies and an possible professorship.
I also should not forget to thank all people who have contributed to open-source (more specifically free) software, which allowed me to do my work properly. The following figure is dedicated to all the free software and all time that people have spend on it: