UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Mari-Liis Kruup

Finding motifs from short peptides

Bachelor’s Thesis (6 ECTS)

Supervisors: Meelis Kull
Balaji Rajashekar
Sven Laur

Author: . . . . " . . . . . . " May 2013

Supervisor: . . . . " . . . . . . " May 2013

Supervisor: . . . . " . . . . . . " May 2013

Supervisor: . . . . " . . . . . . " May 2013

Approved for defense

Professor: . . . . " . . . . . . " May 2013

TARTU 2013

Table of contents

Introduction

Bioinformatics is a field that joins biology and computer science. It mainly deals with providing and using suitable tools to analyze and work with biological data. One part of bioinformatics is analyzing biological material gathered from an individual. This is done to get some specific information about the individual. For example, there are techniques to sequence the human genome to get all genetic information about the person. This can help in developing better treatments and improve personalized health care. There are also a lot of other techniques and methods to get information about the person. They all consist of two main parts. First, it is necessary to actually do the experiment in order to get the biological data. Then, after the experiment, there are data, which are somehow stored in the computer - for example nucleic or amino acid sequences presented as strings. Since the amount of data is almost always very big, it is not humanly possible to analyze it without the help of computers. Computational methods are used to analyze the data in order to extract the necessary information.

In this thesis, a workflow is assembled to extract important information from one individual’s data that have been acquired by doing a certain experiment. The interest in doing that comes from an actual project, where the goal is to find similarities between individuals with the same disease. To achieve that goal, biological experiments had been done with many individuals who have different diseases. As a result, there are a lot of data about every individual. One way to achieve the goal of the project is to analyze the data one individual at a time and extract the most important information from it. Then, individuals with the same disease can be compared to see, if they have similarities.

In the first chapter of this thesis the most important definitions alongside the clear definition of the goal of this thesis are provided. In the second chapter, a literature overview is given about similar methods that have been developed in the past. In the third chapter, data that are analyzed in this thesis are explained in detail. In the fourth chapter, commonly known methods that are used to develop the workflow are introduced. In the fifth chapter, the developed workflow is explained in detail. In the sixth chapter, the results of running the workflow are shown. Finally, in the seventh chapter, conclusions about the developed workflow are made. All the peptide sequences and motifs that are presented in this thesis have been modified in order not to reveal any confidential information. The exact content of the biological experiment done with the individuals is also confidential information and is not explained in this thesis.

1 Preliminaries

1.1 Peptide

In this thesis, data of one individual are represented as peptides. Peptides are short sequences that usually consist of 50 or less amino acids. In the dataset that is being analyzed, all peptides have an equal length of 12 as shown in Figure 1.1. The number of different peptides that one individual has is around 100 000.

```
C I M P R I L Y W F E D
G K I H F M N R M H F W
I E A G P V D R K T S L
```

Figure 1.1: Example of peptides.

All peptides in this dataset also have a count representing the number of times this peptide occurs, or in other words, how many duplicates of this peptide exist in the dataset. This means that the data are represented as shown in Figure 1.2. If all peptides are counted with their duplicates, the size of the dataset of one individual can increase 2 to 3 times.

```
G K I H F M N R M H F W  10
C I M P R I L Y W F E D   1
I E A G P V D R K T S L  84
```

Figure 1.2: Example of peptides with the numbers representing their duplicates.

1.2 Motif

The information that has to be extracted from the peptides of one individual is represented as motifs. Motif is a pattern that exists in peptide sequences. Motifs can be represented in various ways. There are two representations used in this thesis: position weight matrices and regular expressions. The latter is from now on referred to as simple motif.

1.2.1 Simple motif

Simple motif is a representation of the motif in a regular expression like format. Not all possible components of regular expression format are used to generate simple motifs. In every position of the motif can be one of the three following components: a certain amino acid, a group of amino acids (between square brackets) or a dot representing any amino acid. There is also a restriction that dots can only be in the middle of the sequences, a simple motif cannot start or end with a dot. It is said that a peptide matches to the motif when the motif fits somewhere on the peptide. Example of a simple motif and two peptides that match to it are shown in Figure 1.3.

[IFV]M.R...W

C**IM**P**R**ILY**W**FED

GKIH**FM**N**R**MHF**W**

Figure 1.3: Example of a simple motif and two peptides that match to that motif.

1.2.2 Position weight matrix

A position weight matrix is one of the most precise ways to represent a motif. This is a matrix that shows the weight of every element from an alphabet in every position of the motif. Since this work is being conducted with peptides, the alphabet consists of one letter abbreviations of the 20 amino acids. The weight can be interpreted as importance of the amino acid in a certain position and it can be calculated in various ways.

	1	2	3	4	5	6	7	8	9	10	11	12
A	-2.45	0.00	0.86	-2.39	-2.45	-2.45	0.00	-2.45	-2.45	-2.45	-2.36	-2.32
C	-2.45	-2.45	-2.45	-2.39	-2.45	-2.45	-2.45	-2.45	-2.45	-2.45	1.06	-2.32
D	0.00	-2.45	-2.45	0.12	-2.45	-2.45	-2.45	0.00	-2.45	-2.45	1.06	-2.32
E	0.00	-2.45	0.00	-2.39	-2.45	-2.45	-2.45	-2.45	-2.45	0.86	1.06	-2.32
F	-2.45	-2.45	0.86	0.12	3.06	0.00	-2.45	-2.45	0.86	-2.45	1.06	-2.32
G	0.86	-2.45	0.00	-2.39	-2.45	0.00	-2.45	-2.45	-2.45	-2.45	-2.36	0.26
H	-2.45	-2.45	-2.45	0.12	0.86	-2.45	-2.45	-2.45	-2.45	3.06	-2.36	2.38
I	-2.45	-2.45	-2.45	-2.39	-2.45	2.35	-2.45	-2.45	-2.45	-2.45	-2.36	1.68
K	1.40	3.90	-2.45	0.12	-2.45	-2.45	-2.45	-2.45	-2.45	-2.45	-2.36	-2.32
L	2.35	-2.45	0.86	1.92	-2.45	-2.45	0.00	-2.45	2.75	-2.45	-2.36	-2.32
M	-2.45	-2.45	0.00	0.12	-2.45	1.40	-2.45	-2.45	2.91	2.09	-2.36	-2.32
N	-2.45	-2.45	1.40	0.99	1.40	-2.45	3.90	-2.45	-2.45	-2.45	-2.36	-2.32
P	-2.45	-2.45	2.35	1.53	-2.45	-2.45	-2.45	-2.45	-2.45	-2.45	-2.36	-2.32
Q	0.86	-2.45	-2.45	-2.39	-2.45	2.35	-2.45	-2.45	-2.45	-2.45	-2.36	-2.32
R	1.40	-2.45	-2.45	-2.39	0.00	-2.45	-2.45	-2.45	-2.45	1.40	1.06	-2.32
S	0.86	-2.45	0.00	1.92	0.00	0.00	-2.45	3.97	-2.45	-2.45	1.99	-2.32
T	-2.45	0.00	0.00	-2.39	0.00	-2.45	-2.45	-2.45	-2.45	-2.45	1.60	-2.32
V	-2.45	-2.45	-2.45	-2.39	-2.45	0.00	-2.45	-2.45	-2.45	-2.45	-2.36	-2.32
W	-2.45	-2.45	-2.45	-2.39	0.00	-2.45	-2.45	-2.45	-2.45	-2.45	-2.36	2.63
Y	-2.45	-2.45	-2.45	-2.39	0.00	0.00	-2.45	-2.45	0.00	-2.45	-2.36	0.26

CIMPRILYWFED = -9.63 **GKIHFMNRMHFW** = 18.00

Figure 1.4: Position weight matrix and peptides that are matched to it.
Cells of the matrix are colored accordingly to the peptide's amino acids.

To evaluate how well a peptide matches to a position weight matrix, a score is calculated. This is done by placing the peptide on the position weight matrix and adding the weights of the peptide's amino acids in the positions they are placed. The higher the score, the better the peptide matches to the motif. Since sometimes the peptide is shorter than the motif, the peptide is placed in all possible positions and all scores are calculated. The final score is then the biggest one. In Figure 1.4 is an example of a position weight matrix and two peptides with scores saying how well they match to the matrix.

The matrix itself is not very easy for humans to interpret, so a visualization of the matrix has to be generated. This visualization is called a sequence logo. To generate sequence logos, a tool called WebLogo [1] is used. Example of a sequence logo generated with WebLogo is shown in Figure 1.5. The *x*-axis represents the positions of the motif and the *y*-axis represents the amount of information present in every position, measured in bits. If a position would only consist of one amino acid, the amount of information would be the maximum amount of bits, which is $\log_2(20) \approx 4.32$. If a position would contain all the possible 20 amino acids equally, the amount of information would be 0 bits, because there is no information about which amino acid is preferred. So the higher a stack of amino acids is, the more important that position is. Amino acids in one position are ordered so, that the most important amino acid is on the top and the least important in the bottom. Sizes of amino acids in the stack represent their relative frequency among all the amino acids in that position. The colors of the amino acids are based on the chemical properties of the amino acids. Amino acids with similar properties have the same color. From that representation, it is possible to see, that K in the second position, N in the ninth position and S in the eight are describing the motif in Figure 1.5 very well.

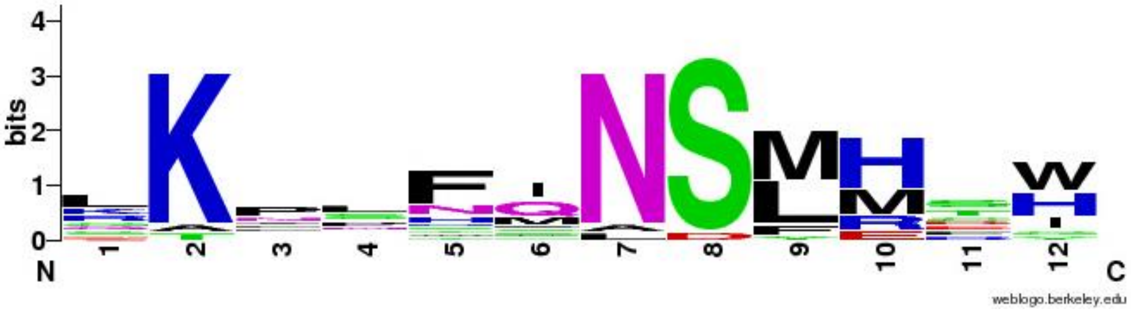
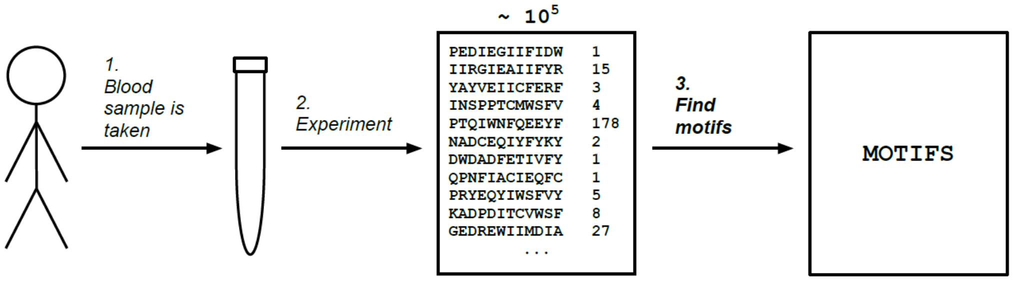


Figure 1.5: Example of a sequence logo. The *x*-axis represents positions of the motif. The size of each letter represents the importance of the amino acid in that position.

1.3 Goal

The goal of this thesis is to develop a method that finds motifs from a set of peptides and represents these motifs as simple motifs and sequence logos. This is the last step of the workflow of describing an individual as shown in Figure 1.6.



PEDIEGIIFIDW	1
IIRGIEAIIFYR	15
YAYVEIICFERF	3
INSPPTCMWSFV	4
PTQIWNFQEEYF	178
NADCEQIYFYKY	2
DWDADFETIVFY	1
QPNFIACIEQFC	1
PRYEQYIWSFVY	5
KADPDITCVWSF	8
GEDREWIIMDIA	27
...	

Figure 1.6: Workflow of describing an individual. The goal of this thesis is to develop a workflow for doing the third step.

All assumptions about the data analyzed and conclusions about the results generated in this thesis are done in collaboration with biologists. It is already known that the peptides of an individual are not random but can be divided into groups that share a common motif. These groups and the motifs that they contain are not known. The challenge is to not define these motifs too narrowly or too widely. This means that if the motif were defined in too much detail, it would not cover the group of all the similar peptides. If the motif were described too generally, it would cover peptides from different groups. The workflow that is developed should be able to identify the motifs that have a clear and strong signal. This means that motifs that are found should be present in a reasonably high amount of peptides. Motifs that are just in a few peptides are not the ones to look for. In conclusion, the main idea is to represent the peptides as a set of motifs that describe the main clusters that the peptides of an individual have.

2 Literature overview

The task of finding motifs from a set of peptides is not novel and there already are some existing tools that are able to do that. There are tools like MUSI [2], MEME [3] and SPEXS [4] that can find motifs from a set of sequences. These tools are all a little different and they all have slightly different goals. None of these goals exactly match the goal of this thesis and therefore these tools cannot be used.

MUSI is a tool that finds position weight matrices from a set of short sequences. It does it by aligning all the sequences and extracting different parts from the alignment which are then represented as sequence logos. It can consider duplicate sequences and it is very fast. MUSI can analyze datasets with hundreds of thousands of sequences within a reasonable time. The downside of MUSI is that it is best suitable for finding sub-motifs from already similar sequences. The peptides in this analysis can be very dissimilar because there are quite many different motif groups. MUSI treats these distant peptides as noise and groups them in one cluster. Therefore, MUSI is not capable of finding the actual motifs and cannot be used.

MEME is also a tool for finding position weight matrices from a set of sequences. MEME should work very well on this dataset because it assumes that the sequences are ungapped, which is true in this case. MEME can also work well with distant peptides that this set contains. MEME’s downside is the high time complexity which means that it is too slow for large datasets. If there were a need to analyze only one individual, it would be acceptable, but since this analysis is eventually going to be run on hundreds of individuals, it would be too time consuming. This is the reason why MEME is not suitable for the analysis.

SPEXS is a tool that finds motifs from a set of sequences. It has to be provided with two sets of sequences – the query sequences, which contain the motifs of interest, and reference sequences, which should not contain these motifs. SPEXS exhaustively scans through the space of all the motifs generated from the query sequences to find the ones that satisfy the search criteria set by the user. It outputs the motifs that are significantly more represented in the query sequences than in the reference sequences or fit some other user criteria. This tool can handle the size of this dataset in a reasonable amount of time but there are also downsides. Firstly, SPEXS requires a reference set. Composing this set is quite difficult because this set can influence the results (motifs to find) quite a bit.

Secondly, SPEXS gives out a large number of motifs. These motifs can be overlapping and very similar. There has to be some post-processing of the motifs in order to really get the right ones. Another disadvantage of SPEXS is that it is difficult to get groups of amino acids (for example [AFT]) as part of the simple motifs. It is possible, but not very convenient and when these groups are added, the performance time increases significantly.

Because of all these reasons these tools are not suitable for this analysis and another approach has to be found.

3 Data from experiments

From every individual, a blood sample was taken and a biological experiment was done on that as can be seen from Figure 1.6. The output from each experiment was a set of short nucleotide sequences which were translated to peptides. The peptides of one individual are the input data for this analysis.

The input data consists of 12mer peptides. The potential size of all the 12mer peptides is 4.096×10^{15} and however the experiment conducted with the blood samples can find around 1×10^{9} 12mers, we can imagine each peptide of one individual's data to situate somewhere in this space. There is a hypothesis that there are clusters or groups of peptides in the data that are closer to each other – clustered, see Figure 3.1. As said in section 1.3, the goal is to find these clusters of similar peptides that describe one individual and represent them as motifs. There should be a way to represent these clusters as motifs because a cluster is formed when similar peptides group together, so they should have something in common.

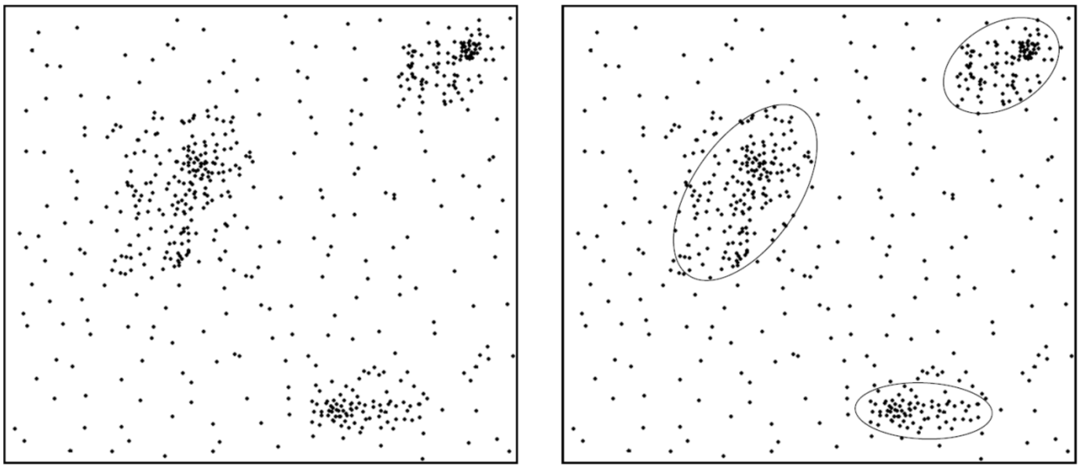


Figure 3.1: Artistic representation of the peptide space. Peptides that are closer together have something similar and should be grouped together. The figure is illustrative and does not correspond to real peptides.

There is also noise in the data. This noise should be interpreted as peptides that are randomly there and do not belong to any of the clusters, like dots that are outside the cluster borders in Figure 3.1. The percentage of noise is thought to be bigger among peptides that have just a few duplicates. Another assumption is that in the center of these clusters should be peptides with high numbers of duplicates. More towards the border of the cluster are peptides with smaller counts. Peptides that are assumed to be the noise should have very small counts. The aim is to find these clusters, draw a line to separate the clusters from the noise and present each cluster with the motif that the peptides in this cluster have in common.

The size of the peptide set that one individual has is around 100 000 peptides without considering duplicates and about 2 or 3 times more with them. From Figure 3.2 it is possible to see, that most of these peptides have only 1 or 2 duplicates. The percentage of noise should be quite high among these peptides. Peptides that have a really high number of duplicates are quite rare. The example individual has only 15 peptides that have more than 300 duplicates as can be seen from Figure 3.2. These peptides are very important in describing these clusters and should be positioned in the middle of the clusters.

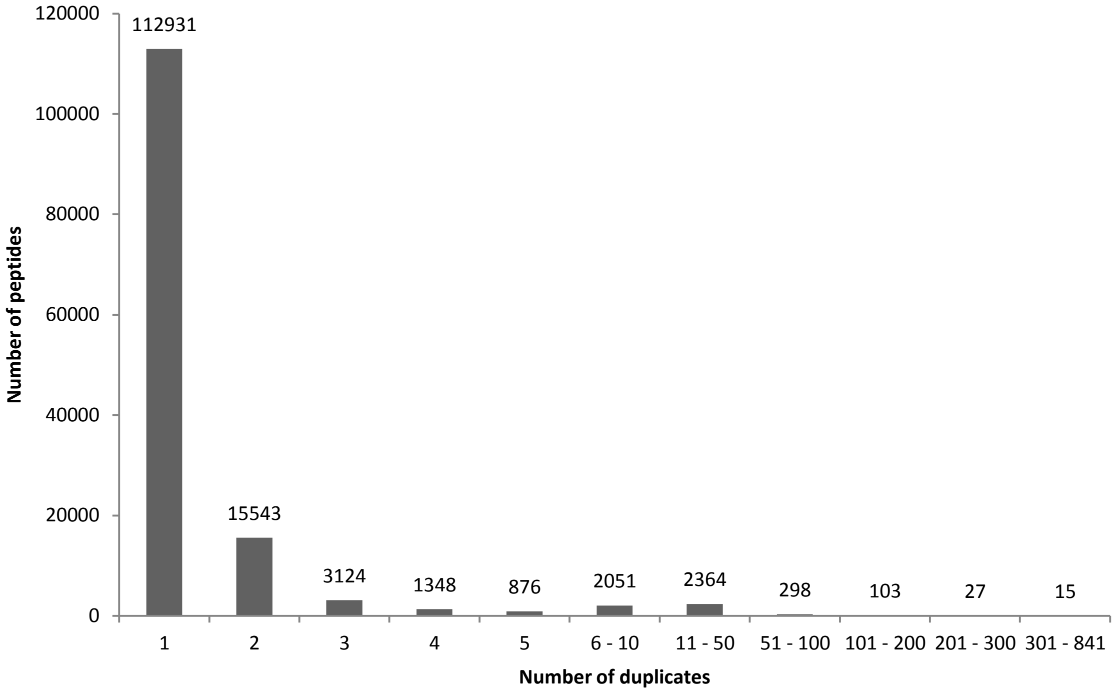


Figure 3.2: Example of peptide distribution of an individual. Figure shows how many different peptides are there considering different number of duplicates.

4 Methods used in the workflow

As a solution for finding motifs from peptides, a workflow based on hierarchical clustering is composed. Hierarchical clustering is selected because this method generates a tree that resembles to the assumed structure of actual clusters. There should be big clusters with more general motif and sub-clusters that have some minor differences in the motif. Hierarchical clustering looks through the peptide space and calculates all distances between all peptides and then joins the peptides into clusters. When the clusters are extracted from the output of hierarchical clustering, each cluster is represented with a motif. That is the main idea of how to find motifs from the peptides of one individual. Since this solution is based on combining several commonly known methods, these methods, hierarchical clustering and multiple sequence alignment, will be described first. A description about how to combine these methods into a workflow will be provided in chapter 5.

4.1 Hierarchical clustering

The method that is used for clustering is called agglomerative hierarchical clustering [5]. This method starts combining peptides together until there is one big cluster. The algorithm is explained with an example in Figure 4.1. The left part of the figure represents the space of all peptides. Similar peptides are closer to each other. The right part of the figure represents the hierarchical tree that this algorithm generates. First, all peptides form a cluster on their own (peptides A-F). Two closest peptides are found (E and F) and they are joined in a new cluster (cluster 1). Then the next closest clusters are found (A and B) and joined (cluster 2). Next, cluster 1 is joined with peptide D and they form cluster 3. This process is repeated until finally clusters 2 and 4 are joined in cluster 5, which contains all the peptides. The result of this clustering is a hierarchical tree of clusters. From this tree it is possible to see, which peptides are closer to each other and form a cluster. It is also possible to see how one cluster can be divided into sub-clusters. As can be seen from Figure 4.1, the algorithm does not automatically generate clusters, but it generates a tree, where these clusters can be extracted from. Extracting clusters from the tree is another step that is going to be described in section 5.3.

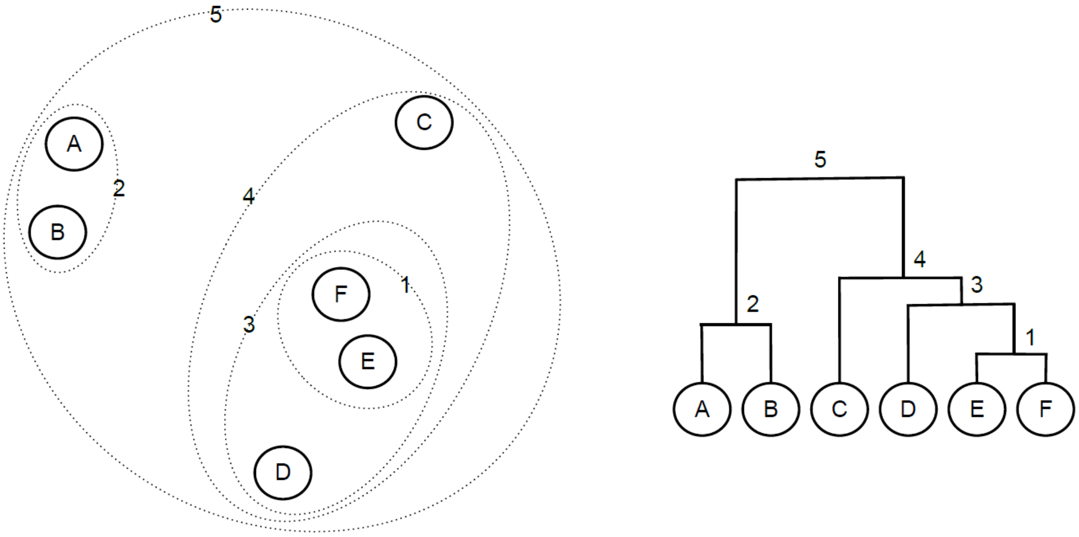


Figure 4.1: Agglomerative hierarchical clustering.

In the algorithm of hierarchical clustering, the distances between peptides and clusters are the basis of joining them. In the left part of Figure 4.1 the closest clusters and their distance can be visually seen. This distance can be calculated in different ways. First, a measure how to calculate the distance between two peptides has to be set. Then, a measure how to calculate the distance between clusters that have many peptides has to be set. Methods chosen for calculating those distances are explained in sections 4.1.1 and 4.1.2.

4.1.1 Levenshtein distance

The distance between two peptides should be interpreted as the difference between two peptides. Since peptides are represented with a 20 letter amino acid alphabet, they are basically just strings. There are two main ways to calculate a distance between strings. First of them, that only works for equal length words, is Hamming distance [6]. It gives the number of places where two strings differ. Since in this case, the similar part of the peptides can be in different positions in the peptides, this method is not suitable. A more flexible method is Levenshtein distance [6]. Given two strings, Levenshtein distance gives the number of edits necessary to turn one string into another. There are three possible edits: deletion, insertion and substitution, which mean that it is possible to delete a character from a string, insert a new character in the string or replace one character of the string with another. An example of calculating Levenshtein distance between two strings (ASMKR and SDKRL) is shown in Figure 4.2. The first step is the deletion of A from the first position. The second step is the substitution of M with D in the third position. The third step is the insertion of L after the last position. With a minimum of three steps it was possible to convert ASMKR to SDKRL. This means that the distance between these two strings is 3.

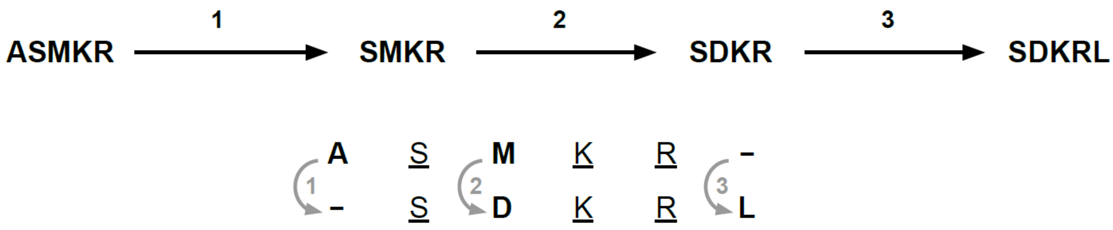


Figure 4.2: Steps to turn one string into another.

One reason why Levenshtein distance is not ideal for these peptides is that it allows insertions and deletions in the middle of the peptide. This should not be allowed because of the nature of these peptides. Insertions and deletions should only occur in the beginning or in the end of the peptide. The reason why this method was still chosen will be explained in the section 5.2.

4.1.2 Linkage criterion

Levenshtein distance covered the problem of how to calculate the distance between two peptides. When calculating the distance between clusters, that contain more than one peptide, it is necessary to use another distance measure. This is called the linkage criterion. There are several ways to calculate the distance between clusters and the three main criteria are shown in Figure 4.3. The first of them is single linkage [5]. With single linkage, the distance between two clusters is calculated by first finding all distances between peptide pairs across the clusters. Then the distance is taken as the minimum distance among those calculated distances. The downside of this criterion is that some clusters may be forced together when they only have a few very similar peptides but most of them are distant. So clusters may be too eagerly joined. Another possible method is complete linkage [5]. This is similar to single linkage but has a difference in that not the minimum but the maximum of all distances between the peptides of two clusters is selected as the distance. This is not ideal as well because a single peptide can again affect the joining too much. The third option is average linkage [5]. This means that all pairwise distances between two clusters are calculated and the mean of these distances is taken. This is more suitable for these data because this compromises between single and complete linkage by minimizing the effect of a single peptide.

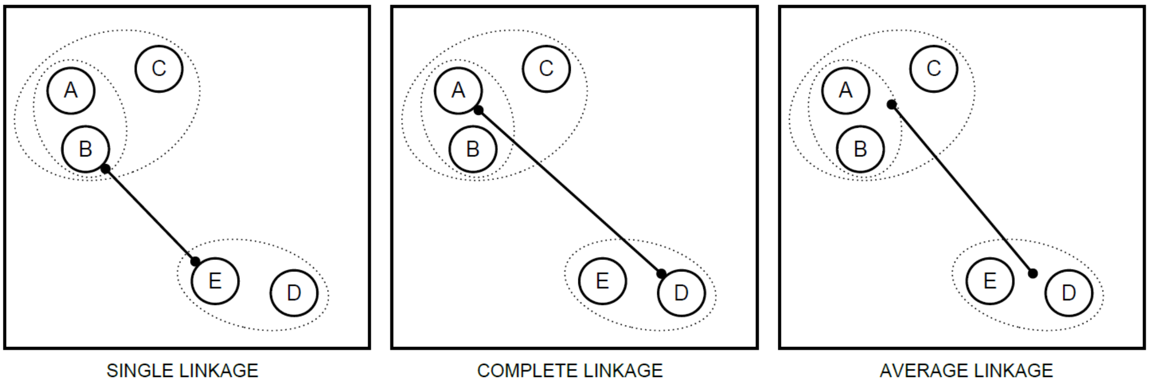


Figure 4.3: Example of different linkage criteria.

4.2 Multiple sequence alignment

Multiple sequence alignment is a method for placing a set of sequences so, that similar parts of them would be in the same positions. This is done in order to see the similarities between some sequences, in this case, to see the similar parts of the peptides that are in one cluster. Most multiple sequence alignment methods and tools generate gapped alignments. Gapped alignment means that there can be insertions or deletions done in the middle of the alignment. Ungapped alignment does not allow these gaps. An example between gapped and ungapped alignment is in Figure 4.4.

```
--GIVKRETDPLAS        --GIVKRETDPLAS
LPGI--RKVDSKEA        LPGIRKVDSKEA--
```

Figure 4.4: Gapped and ungapped alignments.

When generating multiple sequence alignment, scores between sequences are calculated. The main idea of alignment is to set the sequences so that the scores between them would be as small as possible. The score is smaller when the sequences are placed so, that there are as many similar parts as possible. This is done with insertions and deletions in the beginning, end and middle of the sequence. To generate ungapped alignment with a tool that usually allows gaps, a parameter called gap penalty can be used. When gap penalty is set to very high, an additional penalty is added to the score when a gap is introduced in the middle of some sequence. By setting this penalty to very high, these gaps in the middle of the sequences can be avoided because ungapped alignment would always have a better score. There are also methods that do not require setting the gap penalty, but only allow shifting the sequences to align them, but tools that implement these methods are not very common.

5 Workflow

The main idea of the thesis, as mentioned in section 1.3, is to develop a method that would divide a set of peptides into clusters that each represent one motif and then extract these motifs from the clusters. This way a large set of peptides will be described by a much smaller set of motifs. To reach that goal, a workflow is assembled. This workflow consists of many different bioinformatics tools, algorithms and additional scripts that in combination should give the expected result. The main steps of the workflow are shown in Figure 5.1.

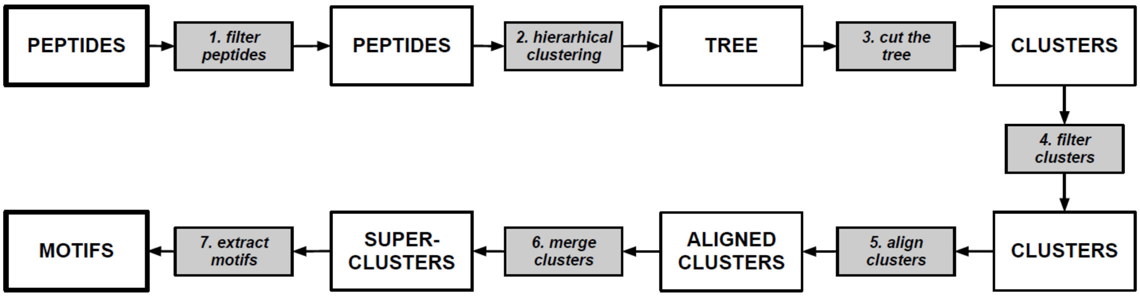


Figure 5.1: Workflow for finding motifs from a set of peptides.

5.1 Filtering the peptides

The first step of the workflow, before doing hierarchical clustering, is to reduce the amount of peptides. This has to be done because the algorithm of hierarchical clustering requires calculating all pairwise distances between given peptides. This means that lots of memory is required to cluster a large number of peptides. Since the size of the peptide set can be up to a few hundred thousand peptides, it is necessary to throw out some of the peptides to be able to do the clustering in a reasonable time. Most of the peptides, that one individual has, have only a small number of duplicates, so eliminating them will decrease the peptide set size multiple times. This elimination can be done because peptides with just a few duplicates are not so important in describing this individual. Also a large percentage of them are thought to be noise and can therefore be excluded without loss of the most important information. For the threshold of eliminating peptides, a value of 3 was selected. This means that peptides that had only 1 or 2 duplicates were thrown out. These peptides are represented by the first two columns in Figure 3.2. The value of 3 was selected because it was the smallest threshold that generated a peptide set small enough for doing hierarchical clustering. Eliminating these peptides removed about 90% of the unique peptides but only 50% of the peptides when considering duplicates as can be seen from Table 5.1.

Table 5.1: Average sizes of peptide sets of 40 different individuals before and after filtering.

	Before filtering	After filtering	Percentage of remaining peptides
Unique peptides	110 428	11 503	10,42%
Duplicate peptides	231 367	119 039	51,45%

The ideal way of doing the clustering would be to include the duplicates. This cannot be done due to the fact that after filtering, the peptide set is still too big when considering the duplicates. Because of this, the clustering is done without the duplicates and duplicates are taken into use again after the clustering.

5.2 Clustering the peptides

The next step of the workflow is hierarchical clustering. This will generate a tree of peptides based on their similarity. From that tree it is possible to extract the clusters. After the filtering, the size of the peptide set is small enough to apply hierarchical clustering on it. To do the clustering, a tool called HappieClust [7] was chosen. This tool was chosen firstly because of its fitness to purpose. It calculates the hierarchical tree from the input peptides. There is no need to combine many different tools to achieve that goal. One limitation of this tool is that it provides only two measures for calculating the distance between peptides: Hamming distance and Levenshtein distance. As said before, Levenshtein distance is quite suitable but not completely ideal because it allows gaps in the middle of the sequences. Regardless, this measure was still used because generated clusters seemed reasonable and since HappieClust was otherwise suitable, this shortcoming was compromised on.

5.3 Cutting the tree

Hierarchical clustering does not automatically generate clusters of peptides. This method generates a tree that represents the hierarchy of clusters. Every node of this tree has a value representing the distance between its two child nodes (sub-clusters). The tree has a root, which represents a cluster of all the peptides. Since it is necessary to get just the right clusters out, the tree has to be cut from the right distance. This is one of the most complicated steps because there is no knowledge about which distance is the right one or how many clusters should one individual have. From Figure 5.2 it is possible to see how the cutting value can affect generated clusters. Since this is a crucial step, the tree has to be analyzed carefully to get an idea where to cut it. To do that, a test individual was analyzed in depth. The hierarchical tree generated from this individual's peptides was cut from different distances to see how the clusters change.

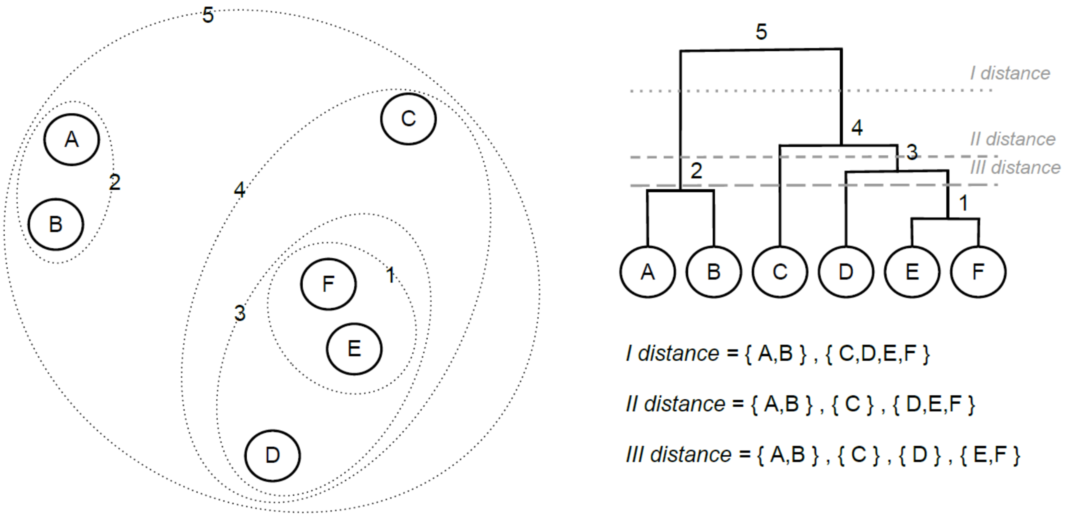

5
A
B
2
4
C
F
1
3
E
D
5
I distance
4
II distance
3
III distance
2
1
A
B
C
D
E
F
I distance = { A,B } , { C,D,E,F }
II distance = { A,B } , { C } , { D,E,F }
III distance = { A,B } , { C } , { D } , { E,F }

Figure 5.2: Cutting the tree from different distances can result in very different clusters.

Since the maximum possible distance between two strings is the length of the longest one, tree nodes have values between 0 and 12. The tree has to be cut somewhere in between this range. To find the right cutting value, tree of the test individual was cut from different distances between 6 and 11. The generated clusters were aligned and visualized as sequence logos to have some information about the generated clusters and the motifs that they contain. Generating alignments and visualizations will be described in detail in sections 5.5 and 5.7.2.

The results of the cutting were reviewed in collaboration with biologists. From the lower distances it was clearly visible that there were very small clusters and one motif was present in many different clusters, as shown in example in Figure 5.3.

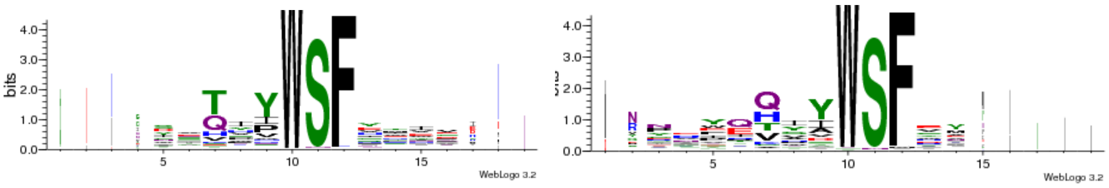


Figure 5.3: Example of sequence logos of two clusters that should be clustered together but are not.

With larger distances, the clusters were very big. There were clusters that were formed from many clusters that should not have clustered together. An example of this kind of clusters is shown in Figure 5.4. At the same time there were still some clusters that had too similar motifs that should have been clustered together as shown in example in Figure 5.3.

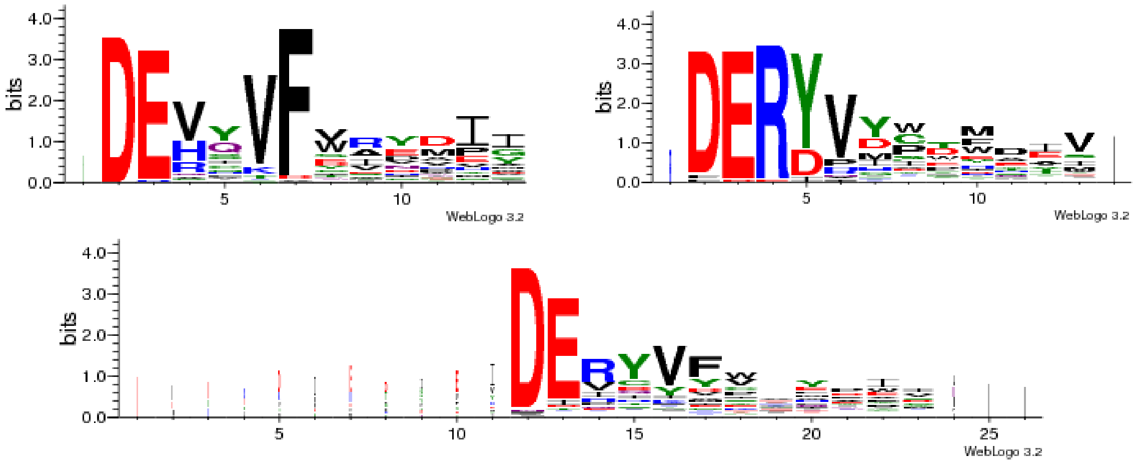


Figure 5.4: Example of sequence logos of two clusters that are clustered together when they should not.

This analysis made it clear that it is not possible to select the perfect cut value. Instead, the tree has to be cut somewhere, where different motifs have not clustered together and later, these clusters should be merged with other methods. The distance of 9 was finally selected as the cutting value because with this distance there were no clusters in the test individual that contained multiple motifs.

5.4 Filtering the clusters

After cutting the tree, there were clusters with different sizes varying from just 1 unique peptide to about 1000 unique peptides. To take duplicates back into use, the peptides in every cluster were multiplied by the number of their duplicates in the original set. This resulted in clusters varying from 3 peptides to about 20 000 peptides. Since these clusters are not ideal and the same motif can exist in many different clusters, similar clusters have to be merged together. But in order to do the merging correctly, some clusters have to be thrown out. This has to be done because the tree was cut from a quite low distance, and because of that, there were some clusters that did not contain very many peptides. When a cluster consists of only a few peptides, the real motif group it belongs to cannot be very well identified. This is because these few peptides could be very similar to many different bigger motif groups which could result in the cluster being merged with the wrong cluster. Also this cluster could not be similar to any of the bigger groups, so it forms a separate cluster when it should not. Because of that, small clusters are hard to merge with other clusters and have to be eliminated to be able to do the merging correctly. There are two parameters set to filter out the small clusters. Firstly, a cluster has to have at least 100 duplicate peptides in it. This is required because when there would be fewer peptides, then the motif would not be clear enough. Secondly, a cluster has to have at least 15 unique peptides. This prevents from having clusters with just a few very similar peptides that have many duplicates. This would be bad because then the motif would consist of the whole peptide sequence and it would be hard to cluster. After this filtering, only bigger clusters with clearer motifs are left.

5.5 Aligning the clusters

The peptides that are in the same cluster have to have something in common so that they were clustered together. This common motif can be in different positions in different peptides. In order to see that motif, the peptides should be placed so, that those similar parts would be in the same positions among the peptides. This will be done with multiple sequence alignment.

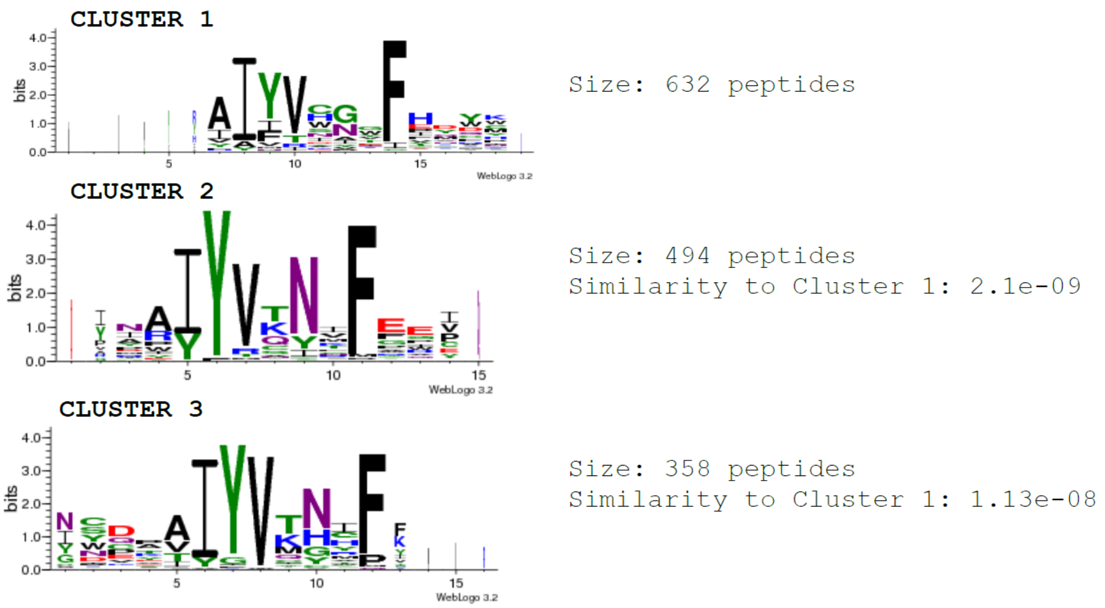
There are many different tools for doing multiple sequence alignment, but there are two big limitations when selecting the tool. Firstly, the tool has to allow ungapped alignment because of the nature of the peptides. There are not many good and available ungapped alignment tools. This means that other tools, that generate gapped alignments, have to be used in a way that results in ungapped alignment. This could be achieved by setting the gap penalty to very high as explained in section 4.2. Since not all tools allow setting this parameter, it limits the selection of finding the right alignment tool. Another criterion for finding the tool is the speed of aligning large sets of sequences. There is a need to align tens of thousands of peptides, and not many tools can do it in a reasonable amount of time.

One of the free and available tools that matched all these requirements was MAFFT [8]. This tool was therefore chosen for doing multiple sequence alignment. MAFFT is a tool that as one function can do multiple sequence alignment. It uses progressive method for aligning the sequences. It takes a set of sequences as input and outputs an alignment of these peptides which can among other things be used to generate position weight matrices.

5.6 Merging the clusters

Cutting the hierarchical tree generated some clusters that share a common motif. To repair that so, that there would be only one cluster per one motif, these clusters with similar motifs have to be merged. In order to do that, there should be some measure saying how similar two clusters are. This similarity could be measured in different ways. One possibility would be to extract simple motifs or position weight matrices from the alignments and compare them. Comparing simple motifs is not selected because it is not sensitive enough. There is no information about the importance of the amino acids. Comparing position weight matrices would be better but would still require additional effort of building the matrices. Another option is to compare the alignments. Since the alignments are already present and there are tools to compare them, this option was selected. A tool called Compass [9] was chosen for comparing the alignments.

Compass is a tool that is able to compare two alignments at a time. It builds profiles from alignments and then calculates an e-value saying how similar the two profiles are. The smaller the value, the more similar the two alignments are. A profile is a similar data structure to the position weight matrix. It contains information about every position of the alignment. Compass is used to find the similarity between two clusters.



Size: 632 peptides

Size: 494 peptides
Similarity to Cluster 1: 2.1e-09

Size: 358 peptides
Similarity to Cluster 1: 1.13e-08

Figure 5.5: Example of three clusters that are merged together.

The entire merging process is conducted as follows. First, clusters are ordered by their size considering duplicate peptides. Then the biggest cluster is taken and all other clusters similar enough to this cluster are merged and they form a new cluster. An example of three clusters that were merged together is shown in Figure 5.5. Those clusters that were merged are removed so they cannot be in any other cluster. Then the next biggest cluster is taken and the same process is repeated. Since the similarity is evaluated with the e-value, there has to be some threshold set saying when clusters are similar enough to join them. A value of 1 × 10^{-4} was selected as this threshold after experimenting with different thresholds on the test individual.

5.7 Extracting the motifs

After the merging, there are groups of clusters, from now on called as super-clusters, that should each contain one motif that is only present in that super-cluster. Now it is necessary to analyze the super-clusters and extract the motifs from them. Steps of this analysis are described in Figure 5.6. Two results are extracted from every super-cluster's peptides: a sequence logo and a simple motif with scores saying how well this simple motif describes specifically that cluster.

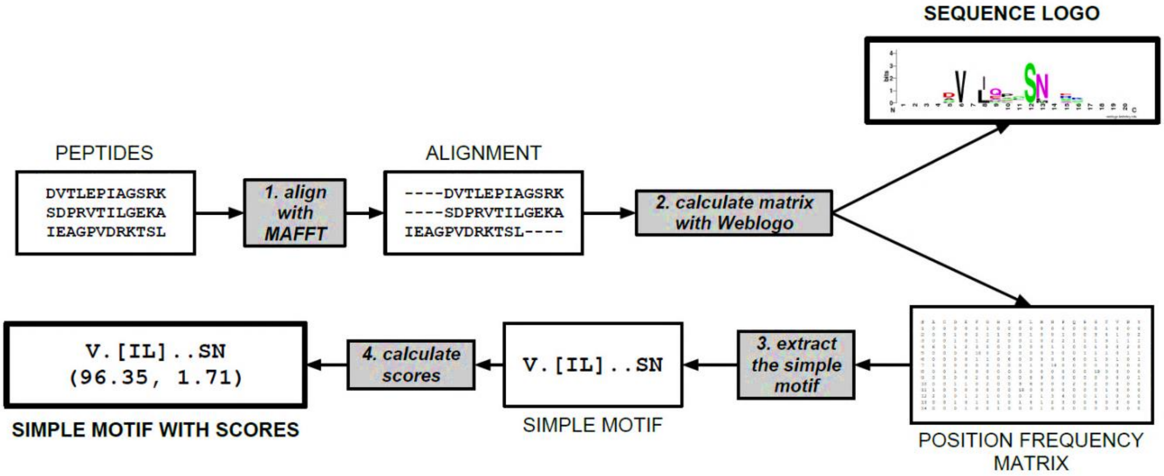


Figure 5.6: Workflow of extracting sequence logo and simple motif from a cluster.

5.7.1 Aligning the super-clusters

The merging resulted in super-clusters that consist of a set of aligned clusters from now on called as sub-clusters. To extract the motif from the super-cluster, peptides of all the sub-clusters have to be aligned together. This is also done with MAFFT. The requirement, that the alignment tool should be able to align very large sets of peptides is especially important in this stage, because the super-clusters are bigger than the sub-clusters and their size can be up to tens of thousands of peptides.

5.7.2 Building position weight matrices

Information about the amino acids present in every position can be read from the alignments of the super-clusters. This information should be now presented as sequence logos and simple motifs. In order to extract them, position weight matrices have to be calculated from the alignments. A tool called WebLogo is used for that. This tool is able to do two things. Firstly, it generates a visualization of a position weight matrix, a sequence logo. This sequence logo is generated for every super-cluster to give an overview of the cluster. By generating these logos, one representation of the clusters is provided. Secondly, it generates a position frequency matrix that is used to extract the simple motifs. Simple motifs will be the second representation of every cluster.

5.7.3 Extracting simple motifs

The visualization of the position weight matrix is good for humans to interpret but not for computers. That is why there is a need to represent a motif in another way. Since in the future motifs of different individuals have to be compared, they could be represented just as alignments or position weight matrices, because there are tools available for comparing them. These options are all good but due to the nature of experiments that are done with the found motifs, they have to be represented as simple motifs. This means that each motif should be read from the position frequency matrix as follows. If only one amino acid is important in some position, this amino acid is added to the motif. If a group of amino acids is important, this group is added to the motif. If no amino acid is important enough, a dot representing any amino acid is added to the motif. In this way a motif like the one shown in Figure 5.7 is generated.

```
.[IFV]M.R...W..
```

Figure 5.7: Motif with dots in the beginning and end.

Since it is not important where in the peptide this motif is positioned, the dots can be eliminated from the beginning and end as shown in Figure 5.8. A motif that is generated after eliminating the dots is called the simple motif.

```
[IFV]M.R...W
```

Figure 5.8: Motif without dots in the beginning and end – a simple motif.

As said, WebLogo generates a visualization of the position weight matrix and a simple motif cannot be read out from that. WebLogo also generates position frequency matrices. A position frequency matrix is like a position weight matrix only that in every position is a number representing the frequency of the amino acid in that position. For extracting the simple motifs, this frequency matrix is used.

To extract information from the position frequency matrix about one position, these frequencies should be converted into percentages to be able to tell which amino acids are more probable in different positions. This is done by first adding the value of 1 to every matrix element for smoothing the values. That is especially important when there are just a few peptides present. Then, every frequency is calculated into a percentage using formula (5.1).

$$\boldsymbol{percentage} = \frac{frequency\ of\ the\ amino\ acid\ in\ the\ position}{sum\ of\ the\ frequences\ in\ that\ position}$$

(5.1)

With this formula, the matrix of frequencies is translated into a matrix of percentages. Every position in the motif is then represented as a vector like the one shown in Figure 5.9. This vector shows the percentages of amino acids in one position.

A	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**
0.07	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.12	0.00

M	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**
0.07	0.02	0.00	0.20	0.00	0.26	0.07	0.01	0.00	0.13

Figure 5.9: Example of amino acid percentages in one position.

In order to represent a position in a simple motif format, the workflow shown in Figure 5.10 has to be run. This workflow will take one vector of percentages as one input. A weight representing the percentage of information present in that position is given as the second input. This weight is generated by WebLogo and it can be interpreted as the percentage of amino acids in that position, or in other words, number elements that are not dashes ("-") in that position of the alignment. If the number is small, then there are a lot of dashes in that position. This weight also reflects in the sequence logos as the width of a column. Firstly, the value of this weight is checked. If it is too low, under 90%, there is a weak signal in that position and a dot is added to the motif. If the weight is high enough, the maximum percentage of amino acids in the vector is found. If some single amino acid already covers 80% of the sequences in that position, only that amino acid is added to the motif. If there is not a strong single amino acid signal, the top three are taken and their percentages are summed. If this sum is over 80%, then a new group is going to be formed. The amino acids are added to the group as follows. The amino acid with the maximum percentage is taken. If its percentage is higher than 10%, it is added to the group. Then the next biggest amino acid is taken. If the group size is smaller than three, the same process is done with that amino acid. If some of these conditions are not true anymore, the formed group is added to the motif. If even the three top amino acids combined did not give a percentage over 80%, a dot is added to the motif. This workflow is run on all of the position vectors to find the simple motif.

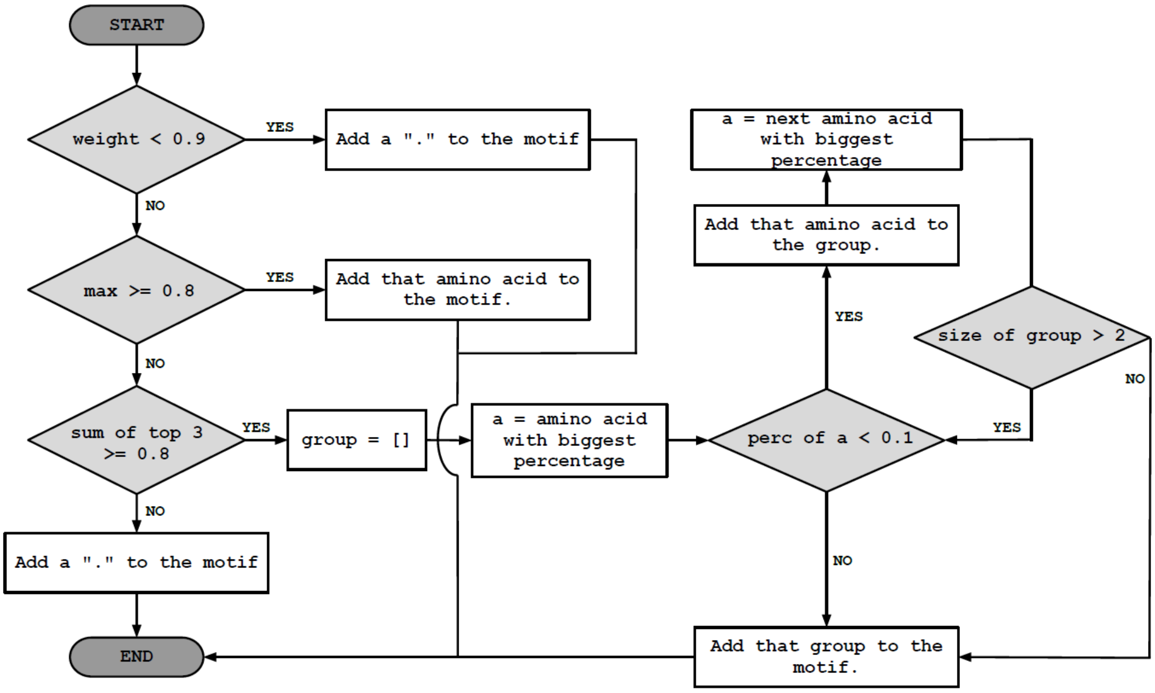

START
weight < 0.9
YES
Add a "." to the motif
NO
max >= 0.8
YES
Add that amino acid to the motif.
NO
sum of top 3 >= 0.8
YES
group = []
a = amino acid with biggest percentage
perc of a < 0.1
YES
Add that amino acid to the group.
a = next amino acid with biggest percentage
size of group > 2
YES
NO
NO
Add that group to the motif.
NO
Add a "." to the motif
END

Figure 5.10: Flowchart of finding one position of simple motif.

When applying this workflow on the position vector shown in Figure 5.9 and taking the weight of the position higher than 0.9, the result would be a dot. This is because no amino acid has a percentage higher or equal to 0.8, the biggest one is S with 0.26. The sum of the three best percentages is 0.59 (S + Q + Y), so there is not a good group present as well. Because of that, a dot is added to the motif in that example.

5.7.4 Calculating scores

The simple motifs that are generated in the previous step are the final outcome of the workflow. To make sure the generated super-clusters are good clusters, they should be evaluated. This evaluation should show how well the found simple motif describes the specific peptide cluster. This evaluation is necessary in order to see how well some individual was described with the simple motifs found with this workflow. To do that, two percentages are calculated. One for saying how well this found motif covers peptides from that cluster. The other is for saying how well that motif covers all other peptides that do not belong to this cluster. A good motif would be the one that covers a large percentage of the cluster's peptides and a small percentage of the rest of the peptides. Since the simple motif can sometimes be very precise, the motif does not have to match the peptide exactly to count as a match. One mistake is allowed. This mistake means that one of the motif's fixed parts (positions that have a certain amino acid or a group of amino acids) could be substituted with any other amino acid. An example of two peptides matching a motif with one mistake is in Figure 5.11.

[IFV]M.R...W

TT**FM**S**S**RNK**W**LP

AMS**R**IFV**W**LKKR

Figure 5.11: Example of peptides that match to the motif with one mistake.

This mistake is allowed only for motifs that have more than three fixed positions. This is because when a motif has only three fixed parts, allowing one mistake would result in a motif with only two fixed parts. This would be so general that there would be too many random matches.

The two scores of every cluster are calculated using formulas (5.2) and (5.3).

$$\boldsymbol{coverage\ of\ cluster's\ peptides} = 100 * \left(\frac{nr\ of\ matching\ peptides\ in\ the\ cluster\ considering\ duplicates}{nr\ of\ all\ peptides\ in\ the\ cluster\ considering\ duplicates}\right) \quad (5.2)$$

$$\boldsymbol{coverage\ of\ other\ super\ cluster's\ peptides} = 100 * \left(\frac{nr\ of\ matches\ from\ other\ super\ cluster's\ peptides\ considering\ duplicates}{nr\ of\ all\ other\ super\ cluster's\ peptides\ considering\ duplicates}\right) \quad (5.3)$$

After these scores have been calculated for every simple motif, the workflow is finished. With this workflow a set of peptides is represented as a set of motifs where every motif is represented in two ways: as a sequence logo and as a simple motif.

6 Results

This workflow was developed and run on one test individual who had 138680 (10206) unique and 277166 (133149) duplicate peptides. The numbers in parentheses represent the size of this individual’s peptide set after filtering out peptides with 2 or less duplicates. The results of running the workflow on the test individual will be described in this chapter.

Table 6.1: Changes in test individual’s peptide set in in different stages of the workflow.

	Original count	Peptides >= 3	Clustered peptides
Unique peptides	138680	10206 (7.36%)	5102 (3.68%; 49.99%)
Duplicate peptides	277166	133149 (48.04%)	94790 (34.20%; 71.19%)

From Table 6.1, it is possible to see that 7.36% of the original unique peptides and 48.04% of the duplicate peptides had more than 2 duplicates and remained in analysis after the filtering of the peptides. After clustering, there were some very small clusters that were thrown out because they were hard to merge with other clusters. The percentage of the original peptides that were actually clustered was 3.68% considering unique peptides and 34.20% considering duplicates. When compared to all the peptides that were taken into clustering (3 or more duplicates), these percentages were 49.99% and 71.19%. Since duplicate peptides are the ones that are most important, these percentages are quite satisfying. Especially considering that from the peptides that were actually clustered, 71.19% were eventually divided into clusters.

Cutting the hierarchical tree of the test individual from distance 9 resulted in 1221 clusters. During the filtering of the clusters, only 99 of them had more than 15 unique or 100 duplicate peptides and remained in the analysis. The sizes of these 99 clusters are shown in Figure 6.1. From the peptides that were inserted into hierarchical clustering, 94790 duplicate and 5102 unique peptides were covered by these 99 clusters.

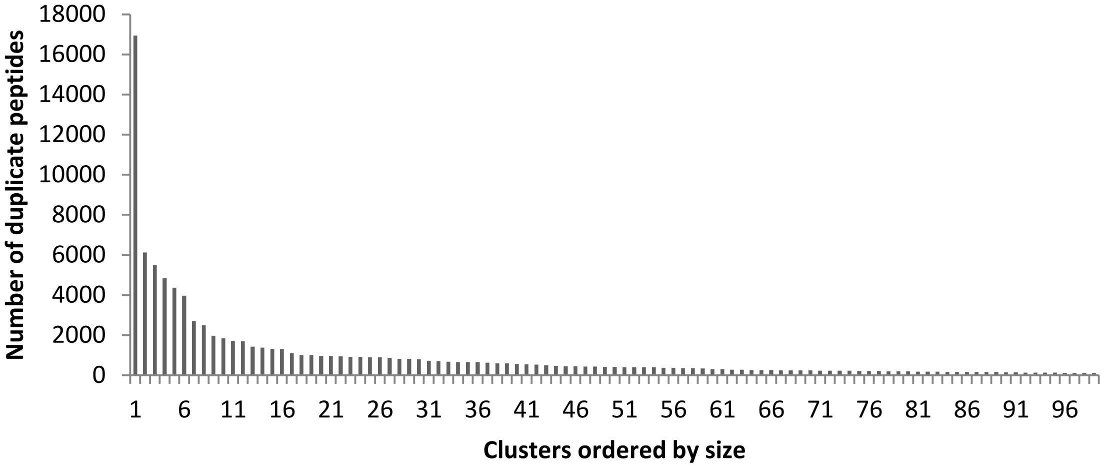


Figure 6.1: Cluster sizes of test individual after filtering of bigger clusters and before merging them into super-clusters.

These 99 clusters were then merged to make sure that there is exactly one cluster per one motif. This merging resulted in 46 super-clusters. The sizes of those super-clusters are shown in Figure 6.2.

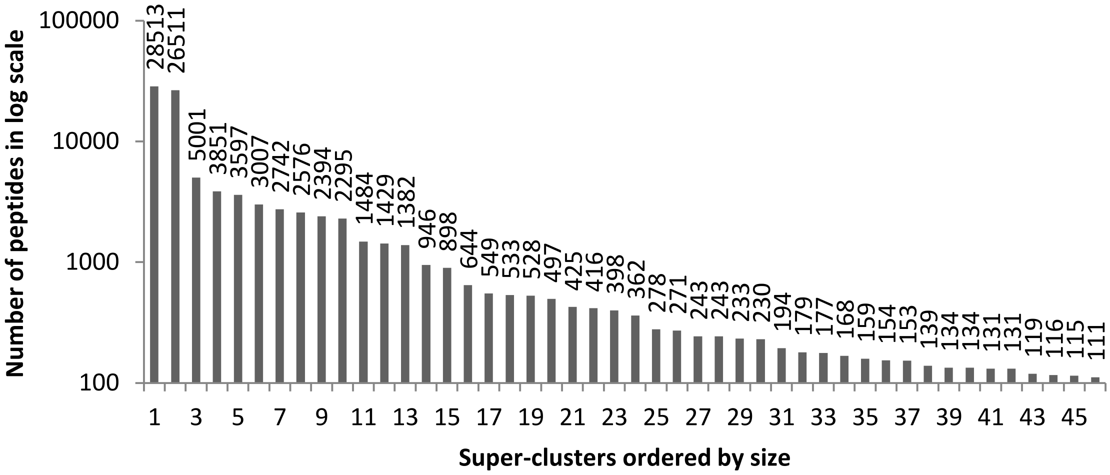


Figure 6.2: Super-cluster sizes of test individual after merging the sub-clusters.

The generated super-clusters had different numbers of sub-clusters. Most of the bigger clusters had a larger number of sub-clusters. The biggest super-cluster, considering the number of sub-clusters, consisted of 13 sub-clusters. Most of the super-clusters, especially the smaller ones, only consisted of one sub-cluster. From Figure 6.3 it is possible to see the number of sub-clusters that each super-cluster contained.

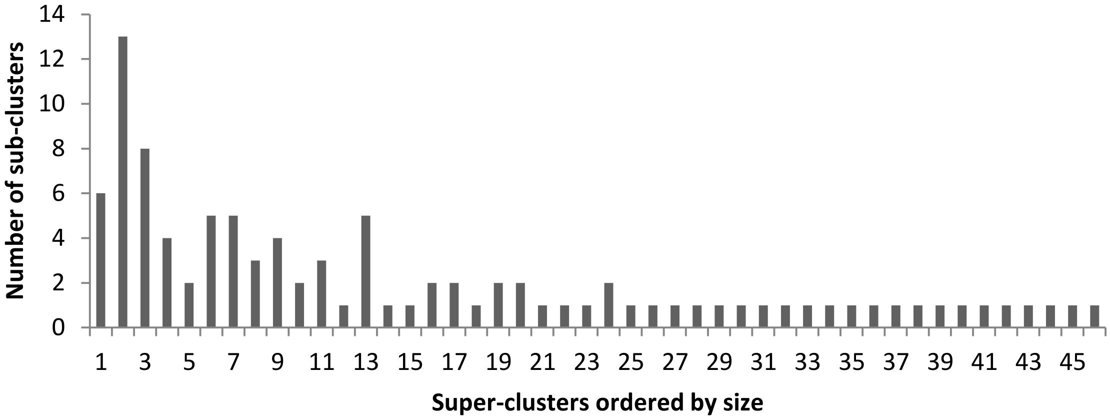
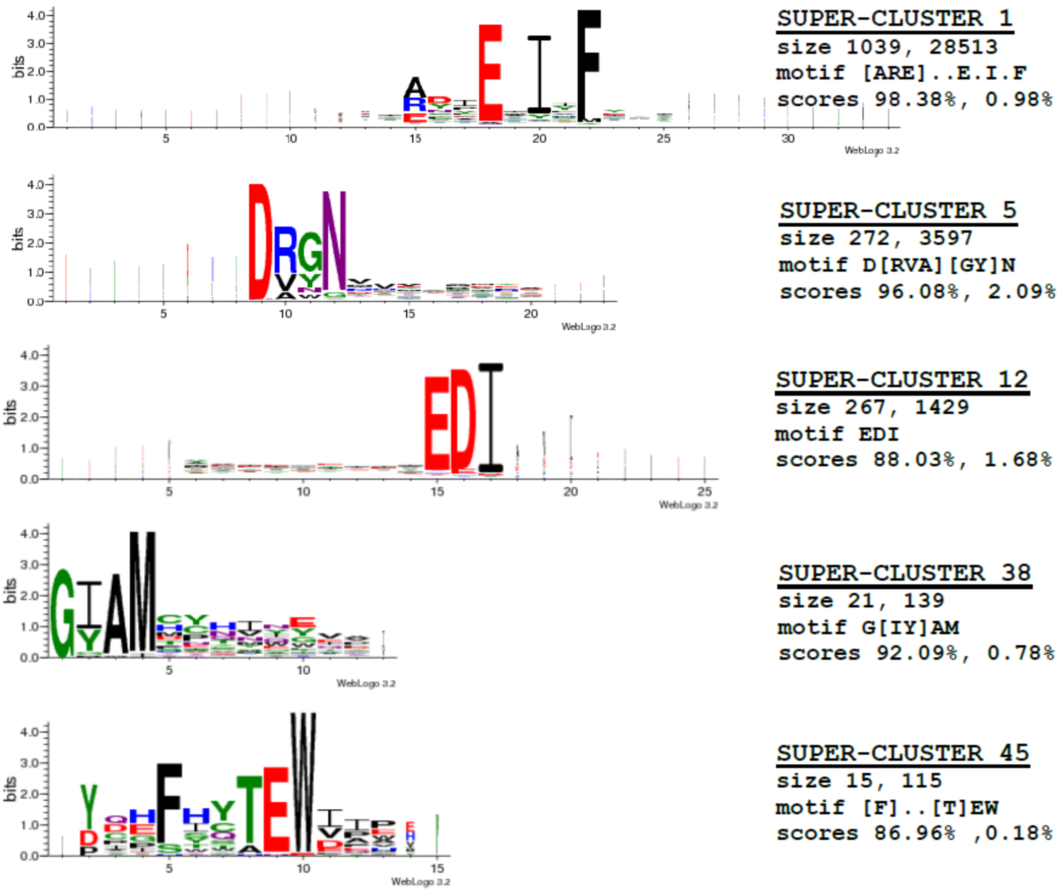


Figure 6.3: Number of sub-clusters in every super-cluster of the test individual.

Every super-cluster was then represented as a sequence logo and simple motif. The simple motifs were also scored for saying how well they describe the specific clusters. Examples of some super-clusters generated from the test individual are shown in Figure 6.4.



SUPER-CLUSTER 1
size 1039, 28513
motif [ARE]..E.I.F
scores 98.38%, 0.98%

SUPER-CLUSTER 5
size 272, 3597
motif D[RVA][GY]N
scores 96.08%, 2.09%

SUPER-CLUSTER 12
size 267, 1429
motif EDI
scores 88.03%, 1.68%

SUPER-CLUSTER 38
size 21, 139
motif G[IY]AM
scores 92.09%, 0.78%

SUPER-CLUSTER 45
size 15, 115
motif [F]..[T]EW
scores 86.96% ,0.18%

Figure 6.4: Example of some super-clusters of the test individual.

From Figure 6.4 it is possible to see the variety of motifs found from one individual. The scores that were given to simple motifs are giving a good indication of how clear these generated clusters were and also how well these simple motifs described the peptides of specifically those clusters. Scores of all super-clusters of the test individual are shown in Figure 6.5.

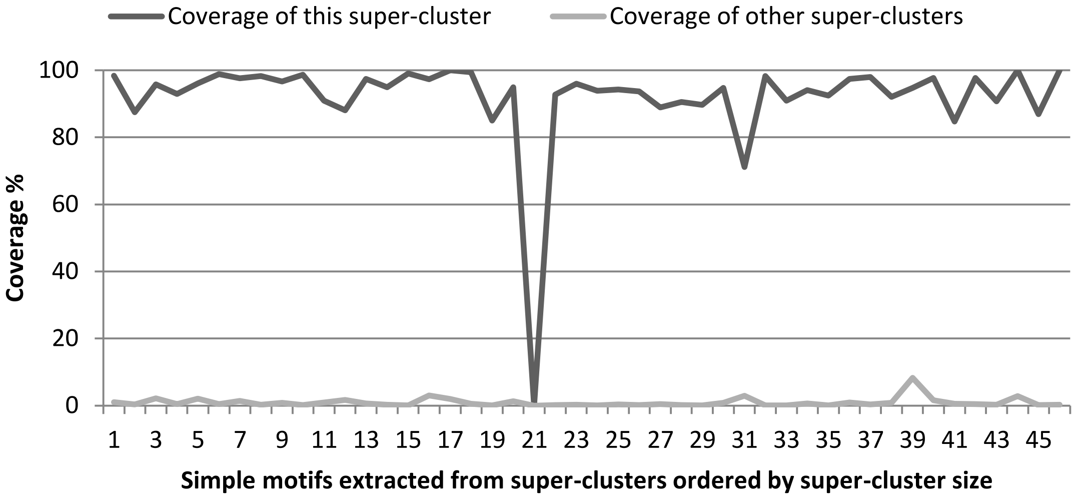


Figure 6.5: Scores of the simple motifs of the test individual.

Almost all of the simple motifs showed very good scores. The second lowest coverage percentage of a super-cluster was 71.13%, but most of them were over 80%. The highest coverage percentage a motif had according to other clusters was 8.29%, but most of them were under 3%. There is one exception, where the signal from the super-cluster was so weak, that the simple motif that was extracted from that consisted of only dots. Since these were eliminated from the beginning and end, there was no simple motif representation of that cluster. Therefore this motif had only zeros as scores. The fact, that only one cluster was so unclear, that it did not show any motif, is another indicator that the generated clusters were quite good. Overall good scores show that the requirement that there is one motif per one cluster is very nicely followed and that the extracted simple motifs are good enough to describe precisely those clusters.

7 Summary

In this thesis, a workflow was developed to extract motifs from one individual's peptides by combining different bioinformatics tools and additional scripts. This workflow was based on hierarchical clustering, which generated clusters that were then post-processed to extract the motifs that they contained. Those motifs were then represented as sequence logos and simple motifs with scores saying how well the motifs represent precisely those clusters.

The workflow was developed on one individual's peptides. The results of applying the workflow on the test individual were satisfying. There were 46 motifs found from the test individual and these motifs described 34.20% of the original peptides and 71.19% of the peptides that were submitted to hierarchical clustering. The latter percentage is especially important because it shows that a high percentage of peptides that were actually the input of the hierarchical clustering ended up in super-clusters.

With the developed workflow, it is possible to extract motifs from one individual's peptides. In the future, this workflow can be applied to different individuals who have different diseases. Then, motifs of individuals with the same disease can be compared to see if there are some similarities. These similarities could be used to learn more about the disease or give some other biological meaning to the results.

Motiivide otsimine lühikestest peptiididest

Bakalaureusetöö (6 EAP)

Mari-Liis Kruup

Resümee

Käesoleva töö eesmärgiks on arendada töövoog, mis leiaks etteantud lühikestest peptiididest sarnaste peptiidide grupid ning esitaks need grupid motiividena. Sellist töövoogu oleks hiljem võimalik kasutada motiivide avastamiseks erinevate indiviidide peptiididest, et leida sarnasusi sama diagnoosiga haigete vahel. Peptiididest motiivide leidmise töövoo koostamiseks kombineeritakse erinevaid üldtuntud meetodeid, bioinformaatika tööriistu ning lisaskripte.

Koostatud töövoog põhineb hierarhilisel klasterdamisel, mille abil jagatakse etteantud peptiidid sarnasuse alusel gruppidesse. Leitud gruppe modifitseeritakse, et koostada just sellised grupid, millest igaüks sisaldaks ühte unikaalset motiivi. Lõplikest gruppidest leitakse motiivid, mis visualiseeritakse logodena ning esitatakse ka regulaaravaldise kujul. Leitud motiividele lisatakse skoorid, mis annaksid infot selle kohta, kui hästi iga motiiv just oma peptiidigruppi kirjeldab.

Valminud töövoog koostati ning rakendati ühe testindiviidi peal. Töövoo rakendamine oli edukas ning etteantud 277166 peptiidist suudeti 71.19% jagada 46 motiivigruppi, millest 43 said ka väga head skoorid. Selle töövoo abil on võimalik edaspidi analüüsida erinevaid indiviide, et leida sama diagnoosiga haigetel ühiseid motiive.

Bibliography

[1] T. D. Schneider and R. Stephens, "Sequence logos: a new way to display consensus sequences," *Nucleic Acids Research*, vol. 18, no. 20, pp. 6097-6100, 1990.

[2] K. TaeHyung, T. M. S., H. Huang, S. S. Sidhu, G. D. Bader, D. Gfeller and P. M. Kim, "MUSI: an integrated system for identifying multiple specificity from very large peptide or nucleic acid data sets," *Nucleic Acids Research,* vol. 40, no. 6, p. e47, 2012.

[3] T. L. Bailey, N. Williams, W. W. Li and C. Misleh, "MEME: discovering and analyzing DNA and protein sequence motifs," *Nucleic Acids Research,* vol. 34, no. suppl 2, pp. W369-W373, 2006.

[4] J. Vilo, "7.1.1 SPEXS - Sequence Pattern EXhaustive Search," in *Pattern Discovery from Biosequences*, Helsinki, Helsinki University Printing House, 2002, pp. 118-121.

[5] T. Hastie, R. Tibshirani and J. Friedman, "14.3.12 Hierarchical clustering," in *The Elements of Statistical Learning*, 2nd ed., New York, Springer, 2009, pp. 520-528.

[6] S. Hosangadi, "Distance Measures for Sequences," *CoRR,* vol. abs/1208.5713, p. 16, 2012.

[7] M. Kull and J. Vilo, "Fast approximate hierarchical clustering using similarity heuristics," *BioData Mining,* vol. 1, 2008.

[8] K. Katoh, K. Misawa, K.-i. Kuma and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," *Nucleic Acids Research,* vol. 30, no. 14, pp. 3059-3066, 2002.

[9] R. Sadreyev and N. Grishin, "COMPASS: A Tool for Comparison of Multiple Protein Alignments with Assessment of Statistical Significance," *Journal of Molecular Biology,* vol. 326, no. 1, pp. 317-336, 2003.

Non-exclusive licence to reproduce thesis and make thesis public

I, Mari-Liis Kruup (date of birth: 14.10.1991),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

“Finding motifs from short peptides”,

supervised by Meelis Kull, Balaji Rajashekar and Sven Laur,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **13.05.2013**