Michael Christian Hetmann BSc

Clustering of alpha beta hydrolase domain containing proteins based on active site cavity properties

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

Masterstudium Biochemie und Molekulare Biomedizin

eingereicht an der

Technischen Universität Graz

Betreuer

Univ.-Prof. Dr. Karl Gruber

Institut für Molekulare Biowissenschaften

Karl-Franzens-Universität Graz

Graz, Juni 2017

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Contents

1. Abstract

In this work a list of α,β-hydrolase domain containing proteins was analysed using clustering methods. Sequence identity, structural similarity and the similarity of the active site cavity of the proteins were investigated.
Since experimental structure data for this proteins is very rare homology models had to be created. Based on this homology models structural alignment was performed and the cavities of the proteins were calculated. Also the distribution of functions within clusters based on the proteins structure was investigated.

The analysis revealed that there is no correlation between structural similarity clusters and specific protein functions since the functions within a structural similarity cluster vary. Furthermore the analysis of the relation between structural similarity and sequence identity confirmed the presumption that there is only a weak link between sequence identity and structural similarity. Proteins which show only very low sequence identity can still show high structural similarity.
The analysis of the active site cavities show that clusters based on the active site cavity properties do no correlate with the clusters based on structural similarity. This leads to the assumption that although proteins show a similar structure they still can show different active site cavities. A correlation between cavity properties and the protein functions could not be studied because of a lack of information about the protein functions.
Furthermore a group of α,β-hydrolase domain containing proteins was identified, which do not contain the typical catalytic triad.

2. Aim of the thesis

The aim of the thesis was to analyse a list of α,β-hydrolase domain containing proteins to get more functional and structural information about the proteins. Also the relationship between the proteins was studied. To achieve this aim clustering methods based on the sequence identity, the structural similarity and the active site cavity properties of the proteins were used.

3. Background

3.1 α,β-hydrolase domain containing proteins

α,β-hydrolase-domain (ABHD) containing proteins are characterised by their conserved structural motif and are part of the superfamily of α,β-hydrolases. This superfamily is characterised by the α,β-hydrolase fold. [1]
The ABHD containing proteins are present in all reported genomes and are - due to their conserved structural motif - expected to play similar roles in lipid metabolism and signal transduction. Still their physiological roles and roles in metabolic pathways are largely unknown.
Mutations in some ABHD containing proteins are linked to errors in lipid metabolism and studies in cell and animal studies have shown that ABHD containing proteins play an important role in lipid metabolism, lipid signal transduction and metabolic diseases. [2]

3.1.1 Structure of the α,β-hydrolase fold

The canonical α,β-hydrolase fold consist of a 8 stranded β-sheet, with the second β-strand being anti-parallel. The sheet is surrounded by α-helices and loops, connecting the β-strands:

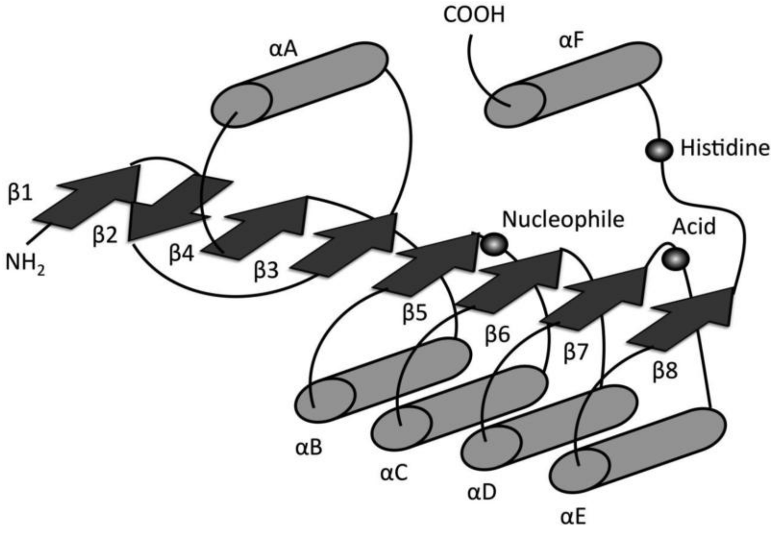

Figure 3.1: Scheme of α,β-hydrolase fold [1]

The fold shows large plasticity. Large insertions can be made, varying from single amino acids to whole domains. This gives the members of the family the ability to adapt and for evolution. The variation of the fold results in enzymes, which are able to operate on substrates, which differ in their chemical and physiological properties therefore allow the proteins to act in different biological contexts. [3]
Here is an example of a protein showing this fold:

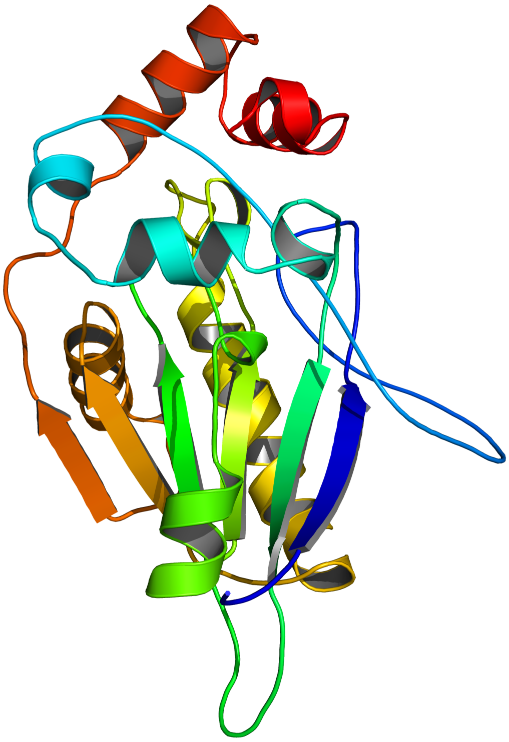

Figure 3.2: α,β-hydrolase fold shown in Neuroligin-2 (Trembl-code:F6VE93)

Although the members of the superfamily show a high structural similarity the sequence identity is low. The functions of the proteins in the superfamily also vary.

3.1.2 Catalytic activity

The catalytic activity derives from a catalytic triad composed of nucleophile-acid-histidine residues, located on loop regions. The nucleophile (Ser, Cys, Asp) is located in a very tight loop, the "nucleophile elbow". [2] It is identified by the consensus sequence Sm-X-Nu-X-Sm (Sm = small residue, X = any residue and Nu = nucleophile). In many ABHD proteins the corresponding motive is GXSXG. Other motifs are GXCXG and GXDXG.

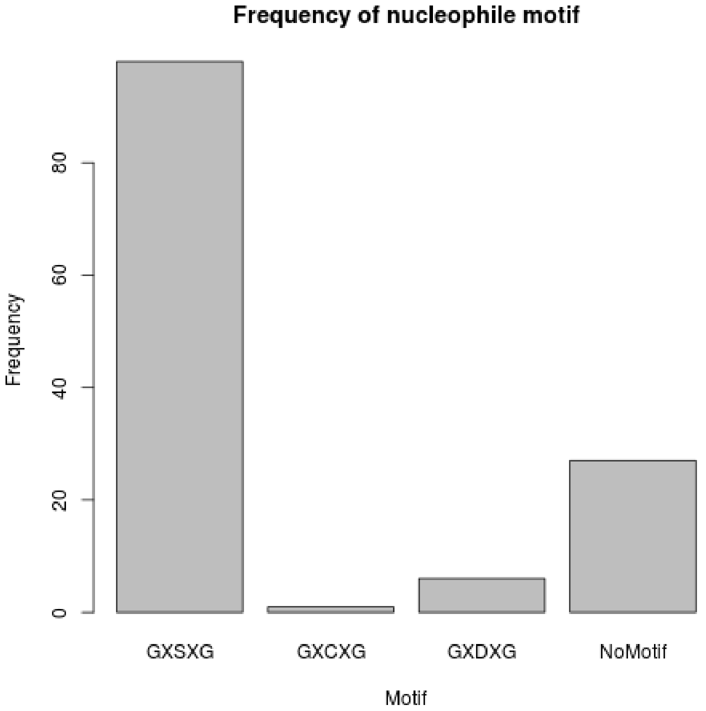

Figure 3.3: Frequency of nucleophile motifs in the analysed ABHD containing proteins

Aspartate or glutamate can be the acid residue in the catalytic triad. The acid is usually located in a loop region after β strand 7. The histidine residue, which is absolutely conserved, is located in a loop following β strand 8. If the sequences are scanned not only for GX(S)(C)(D)XG but also for AX(S)(C)(D)XA only 11 sequences do not show the Sm-X-Nu-X-Sm motif. Most of the proteins, lacking the Sm-X-Nu-X-Sm motif, do not contain the nucleophile residue.

In many ABHD containg proteins a second motif, HXXXXD can be found. This motif is linked to acyltransferase activity [4] [5] . In the used dataset nearly half of the ABHD containing proteins showed this motif:

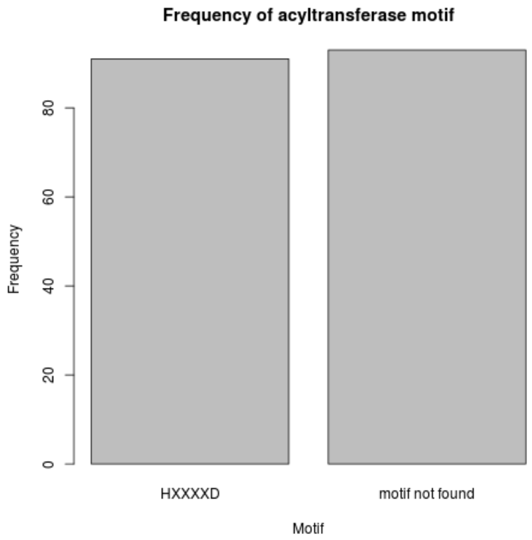

Figure 3.4: Frequency of acyltransferase motif

Several ABHD containing proteins are expected to harbour both hydrolase and acyltransferase activity. Because of this conserved motifs, important roles of ABHD containing proteins in synthesis of small molecules in signal pathways and lipid metabolism are predicted. [2]

Structural analysis of α,β-hydrolase domain containing proteins revealed some common properties of ABHD containing proteins: [3] [6]

- The sequence order of the catalytic triad is nucleophile-acid-histidine
- A "Gly-X-Nu-X-Gly" sequence defines a "nucleophile-elbow" element. The element is located in a tight loop connecting the β-strand and the following α-helix.
- The structure starts at β3-strand and is at least five β-strands long.
- A loop after β7 carries the acid member of the triad, allowing the correct formation of the hydrogen-bond network necessary for catalytic activity.

All α,β fold proteins show the first feature and at least 2 out of the last three features.[3] [6]

3.1.3 Functions of α,β fold proteins

The α,β fold proteins catalyse a variety of reactions. The group contains different types of enzymes, including carboxylic acid ester hydrolase, lipid hydrolase, thioester hydrolase, peptide hydrolase, haloperoxidase, haloalkane dehalogenase, epoxide hydrolase and C-C bond breaking enzymes. [6]

General examples for α,β fold proteins

Acetylcholine Esterase
Acetylcholine Esterase is critical for termination of a nerve impulse. This is achieved by rapid hydrolysis of the neurotransmitter acetylcholine into acetic acid and choline. [6]

Dienelactone Hydrolase
Dienelactone Hydrolase is an enzyme, which occurs in bacteria and fungi. It degrades aromatic compounds such as dienelactone to maleylacetate. [6]

Haloalkane Dehalogenase
Haloalkane Dehalogenase is an enzyme capable of cleaving carbon-halogen bonds. It converts the halogenated alkane substrates into the corresponding alcohols. [7] [6]

Examples of analysed ABHD containing proteins

ABHD1
ABHD1 is a 405 residue protein, which does not contain the HXXXXD motif. The function of the protein is yet not confirmed. But its overexpression in renal cell lines is linked to the reduction of the generation of reactive species by NADH-oxidase. In a oxadative-stress induced mouse model kidney ABHD1

expression is up-regulated.[8] This suggests that ABHD1 is a potential regulator of oxadative stress. [2]

ABHD3
ABHD3 is a 409 residue protein. It does not show a HXXXXD motif.
Studies showed that cells, which overexpress ABHD3, show a higher phospholipase activity toward C14-containing phosphatidylcholine comparing to cells, which overexpress a catalytic dead mutant. Tissue metabolomics of ABHD3 knockout mice show higher levels of tissue metabolomics of ABHD3 knockout [9].

ABHD6
ABHD6 is a 337 residue protein.
2-arachidonylglycerol (2-AG) was the first identified substrate for ABHD6. 2-AG is an endocannabinoid signaling lipid, which plays a key role in neurotransmission and metabolic disease. [2]

3.1.4 Mechanism

Despite the fact that α,β-hydrolase fold proteins catalyse many different reactions, the activities of the enzymes can be categorised in three groups:

1. Cleavage of a peptide-, oxyester or thioester-bond
2. Breaking of a C-Halogen or C-O bond at a sp3 carbon atom with concomitant formation of a C-O bond
3. Breaking of C-C bonds

The chemical strategy for catalysis shares a general principle. In all catalysed reactions the attack of the substrate by the activated nucleophile is necessary. This results in an substate-enzyme ester formation. The oxyanion hole - which is a pocket in the active site, which stabilises a negative charge on oxygen and is present in all α,β-hydrolase fold proteins - plays an important role in catalysis. For reactions belonging to category 1 it is involved in the acylation and deacylation steps, in reactions belonging to category 2 it stabilises the transition state during the cleavage of the substrate-ester bond. [6]

Displayed below is the mechanism for the activation and the initial step in the hydrolase mechansim:

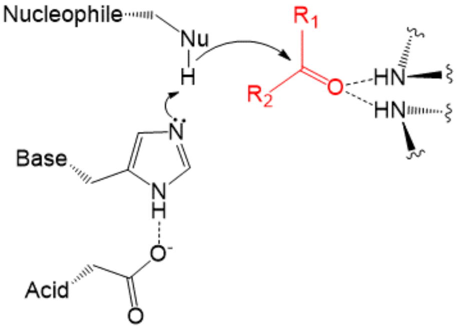

Figure 3.5: Activation mechanism of the catalytic triad. The picture is derived from `https://en.wikipedia.org/wiki/Catalytic_triad` and is usable under the CC BY-SA 3.0 licence [10]

[1]

The nucleophile residue - in ABHD containing proteins a serine, cystein or aspartate - is turned into a very potent nucleophile by deprotonation. The histidine acts as a base and takes up the proton. This is possible because the acid residue - a glutamate or an aspartate - orientates and polarises the histidine.

This activation mechanism of the nucleophile residue allows its attack on the substrate.

[1] In this scheme R1 can not only represent a carbon atom but also an -OR group. The same is true for the reactions scheme below.(This reactions is then a ester hydrolase reaction)

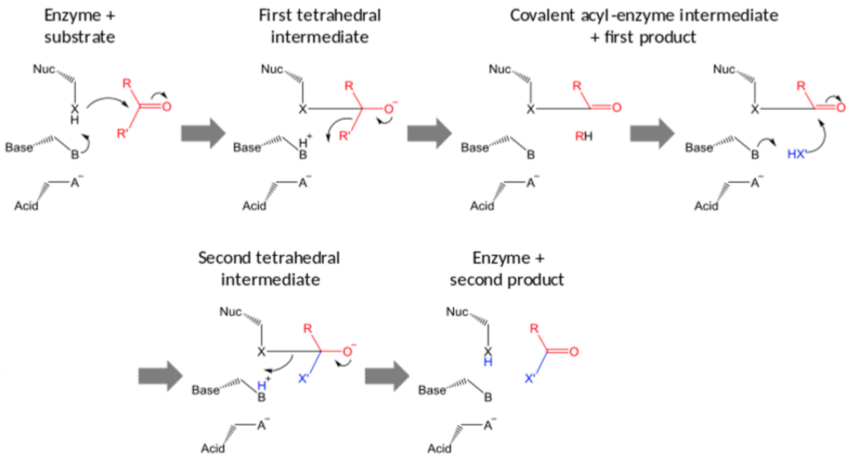
Enzyme + substrate
First tetrahedral intermediate
Covalent acyl-enzyme intermediate + first product
Second tetrahedral intermediate
Enzyme + second product
Nuc
X
H
R
O
R'
Base
B
Acid
A⁻
RH
HX'
X'

Figure 3.6: Hydrolase reaction mechanism. The picture is derived from `https://en.wikipedia.org/wiki/Catalytic_triad` and is usable under the CC BY-SA 4.0 licence [10]

The substrate binding and attack of the nucleophile on the carbonyl-group of the substrate lead to formation of an tetrahedral intermediate. Under splitting off an alcohol group a covalent acyl-enzyme intermediate is formed. The nucleophilic attack of a water molecule on the intermediate yields a second tetrahedral intermediate. After bond-breakage the product is formed. [10]

3.2 Homology Models

3.2.1 General Information

Structural information of proteins is always of great interest. Key functions of a protein are highly dependent and linked to its structural properties. Therefore it is essential to investigate the enzyme's structure to understand enzyme properties like enzymatic activity, ligand binding, protein-protein interaction etc.. Yet there is not always experimental data for a protein's structure available. A possibility to still get information about the protein's structure, is to create a homology model.

Homology models are models for the 3D structure of a protein. The 3D structure is created on basis of the structure of a sequence similar protein.

3.2.2 Creation of a Homology Model

The creation of a homology model involves different steps [11]:

- Identification of template and sequence alignments
- Model building
- Model refinement
- Model validation

Identification of template and alignment

The identification of the template is the first step in the homology model generation. The sequence of the protein with unknown structure is compared to known structures stored in the Protein Database (PDB).

For the sequence alignment different tools like BLAST, PSI-BLAST, ClustalW can be used. A similar protein with at least 30% sequence identity is needed for homology model creation. [11]

Model building

The next step is model building for the target based on the 3D structure of the template. This can be done by different methods. Some often used methods are:

- Segment matching [12]
- Rigid-body assembly [13]
- Spatial restraint [14]
- Artificial evolution [15]

Model refinement

Model refinement is an important task in homology model generation. It involves improving alignment, loop refinement and side-chain orientation refinement. An energy minimisation step, using molecular mechanics force fields [16] [17], is placed at the beginning of the refinement process. After that molecular dynamics techniques like Monte Carlo and genetic algorithm-based sampling can be applied. [18] [19]

Model validation

The model validation is done by comparing the model with a gold standard of trusted reference structures. Since the sequence identity of the reference structure and the model could be as low as 0% they can not be compared directly. The comparison is based on general protein features encoded in knowledge based potentials. This are energy functions, which are created through statistical analysis of known protein structures. Qualitative insights can be converted to quantitative data using the Boltzmann-Formula. [20]
The created energy potentials are normalised, which removes the dependency on size and shape of the protein. A so-called 'Z-score' is created, which indicates how many standard deviations a structure is below the average. Z-scores form the basis for model validation in tools like WHAT-CHECK. [21] [22]

3.3 Catalobase

3.3.1 General workflow

Catalobase is a tool, which analyses protein cavities. It generates a point-cloud, which is dependent on the form and size of the cavity but also on the properties of the amino acids, surrounding the cavity. [23] It also allows the finding of proteins, with a similar active site by comparing the active site cavities based on the belonging property point cloud.
This was used to compare ABHD containing proteins with each other to finally create similarity clusters.
Displayed below is the general workflow of the cavity calculation:

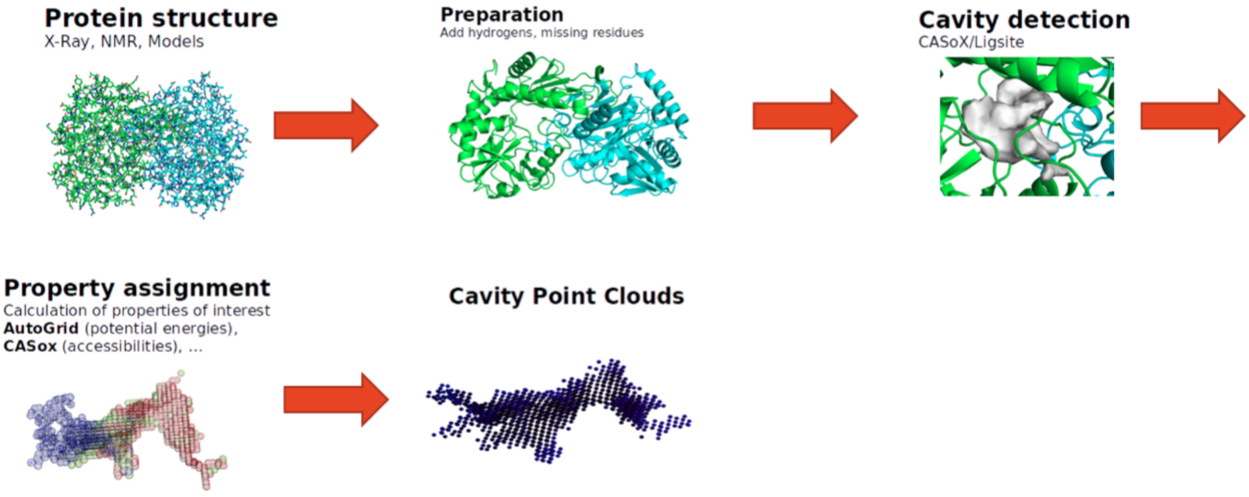
Protein structure
X-Ray, NMR, Models
Preparation
Add hydrogens, missing residues
Cavity detection
CASoX/Ligsite
Property assignment
Calculation of properties of interest
AutoGrid (potential energies),
CASox (accessibilities), ...
Cavity Point Clouds

Figure 3.7: The general workflow of the cavity property point cloud calculation

The starting point is always a 3D-structure of the protein. This can either be an experimental structure (NMR, x-ray) or a homology model. The first step in the calculation of the cavities is a preparation phase. In this phase hydrogen atoms and missing residues are added. After that the cavities are detected. For the detection Ligsite [24] is used.
Using AutoGrid and CASox the properties of the cavity are calculated. Finally the cavity and its properties are represented as a point cloud.

3.3.2 Ligsite

Ligsite is a program used for detection of pockets on the surface of proteins as well as binding pockets for small molecules. The cavity calculation is a multi-step process:

Scanning for cavities

The process starts with generation of a Cartesian grid, all grid points get labelled as solvent and set to a value of 0. Grid points which are inaccessible to solvent are set to a value of -1.
To identify grid points positioned in a cavity the program scans for regions which are enclosed on both sides by solvent-inaccessible grid points along the x axis. There is a high probability for an area of solvent-accessible grid point to be in a cavity if they are surrounded on both sides by solvent-inaccessible grid points. If the grid point is solvent-accessible its value is increased by 1. Such an arrangement (first protein, then solvent, then protein again) is called a PSP event.
The x-axis scan is done for all y and z values and the same procedure is repeated for the y-axis and the z-axis. To make sure that the orientation of the cavity does not have an influence on the correct calculation also the four cubic diagonals are scanned. [24]

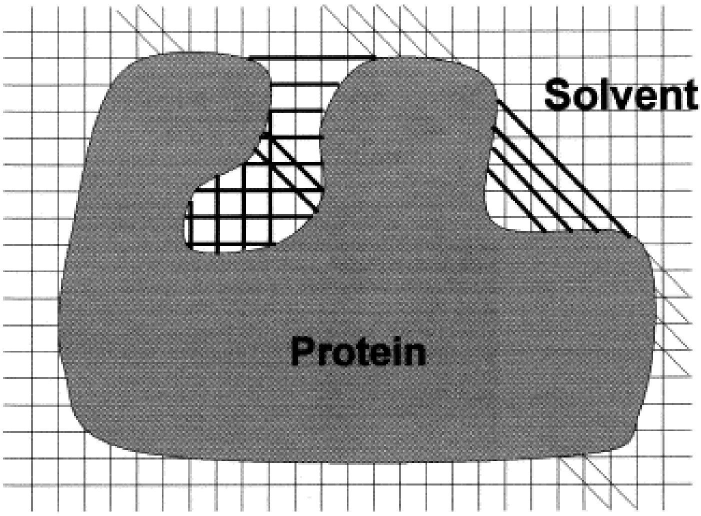

Figure 3.8: Scheme of pocket-searching algorithm of LIGSITE [24]

Recognition of cavities

After the scanning steps all grid points accessible to solvent are assigned values between 0 (no PSP event, the grid point is not located in a cavity) and 7 (the grid point is deeply buried, a PSP event along all axis as well as along the cubic diagonals). A cavity is now defined as a region of grid points with a certain minimum number of PSP event. [24]

Determination of the surface cavities

Now all grid points in which a solvent molecule cloud be placed without overlapping with the proteins are determined. These grid points are then used for determination of the solvent-accessible surface. This works as follows: First the grid points belonging to the same cavity are identified: All grid points within a sphere with the radius of $r_{solvent}$ centred at a grid point in the cavity belong to the cavity. Then the grid points which represent the surface are identified. The grid points are surface points if at least one of the nearest neighbours in the grid is occupied by a protein atom. [24]

3.3.3 AutoGrid

AutoDock is a simulation software for molecular docking. It predicts how small molecules bind to a protein.

AutoGrid is a module for AutoDock which calculates grids describing the protein. The protein is placed on a 3D grid, a probe atom is placed at each grid point. Then the energy of interaction of this single probe atom with the protein is assigned to the grid point. This generates AutoGrid affinity grids. These grids are calculated for different type of atoms (typically carbon, hydrogen, oxygen and nitrogen, but if needed also for other atom types). Also grid for protein properties like desolvation potentials or electrostatic (and others) are calculated. [25]

3.3.4 Cavity Matching using the Iterative Closest Point algorithm

For comparison of the cavity (called cavity matching) the ICP (Iterative Closest Point) algorithm is used. The algorithm is used to minimise the difference between two point clouds. It can be used for 2D and 3D point clouds, in this application it's used for 3D point clouds.

A transformation is calculated to minimise the distance between two point clouds. First all points of the first cloud are matched to the closest points in the second cloud. After that sum of the root mean square distances is minimised by changing the parameters of the transformation. The process is iterative and carries on until an optimum is reached. [26]

3.4 Hierarchical Clustering

Hierarchical clustering is a method for cluster analysis and often used in bioinformatics. It builds a hierarchy of clusters.
The different methods for hierarchical clustering can be assigned to two groups: [27]

- Agglomerative or "bottom up" approach: every object is assigned to each own cluster, then the clusters are merged pairwise until only one cluster is left. The clusters are merged pairwise, starting by the clusters with the smallest distance. The distance between two cluster is calculated differently dependent on the used agglomerative method used for clustering.

- Divisive or "top down" approach: all objects start in one cluster and are then split up into separate clusters.

4. Methods

4.1 Investigated ABHD containing proteins

The following list of ABHD containing proteins was analysed:

mouse-ES10	mouse-acot5	mouse-Ces3a	mouse-Q9DAI6
mouse-ABHD6	mouse-acnt1	mouse-Ces2c	mouse-tmco4
mouse-ABH15	mouse-bat5	mouse-Ces2d-ps	mouse-tmm53
mouse-abhd3	mouse-Q80YU0	mouse-Ces1f	mouse-rbbp9
mouse-abhd1	mouse-abhda	mouse-Ces3b	mouse-Q8C1A9
mouse-LABH2	mouse-abd12	mouse-Ces1d	mouse-CB043
mouse-acche	mouse-SERHL	mouse-Ces1b	mouse-ephx4
mouse-1lipg	mouse-WBSCR21	mouse-Ces1a	mouse-EPHX1
mouse-1llip	mouse-g3uzn6	mouse-Ces4a	mouse-hyes
mouse-LIPM	mouse-buche	mouse-Ces2f	mouse-Abhd8
mouse-LIPK	mouse-CPMac	mouse-abhd5	mouse-ephx3
mouse-Lipo4	mouse-RISC	mouse-abhd4	mouse-OVCA2
mouse-Lipo2	mouse-prtp	mouse-pches	mouse-1hlip
mouse-LIPN	mouse-Ces2b	mouse-c1ib	mouse-kynfo
mouse-Lipo1	mouse-cauxin	mouse-Dorz1	mouse-adcl4
mouse-apeh	mouse-Ces2h	mouse-CMBL	mouse-Q8BUY2
mouse-acot1	mouse-Ces1h	mouse-dpp10	mouse-4930449A18RIK
mouse-acot3	mouse-Ces2a	mouse-dpp4	mouse-adcl3
mouse-acot2	mouse-Ces2g	mouse-DPP6	mouse-hslip
mouse-acot4	mouse-Ces2e	mouse-bphl	mouse-aryla
mouse-BAAT	mouse-Ces1g	mouse-FAP	mouse-Q8BLF1
mouse-Q91XC7	mouse-Ces1e	mouse-dpp9	mouse-q6wqj1
mouse-Q8BYI3	mouse-Ces1c	mouse-F135A	mouse-DGLB
mouse-lipli	mouse-paf2	mouse-thyro	mouse-AI607300
mouse-Lipg	mouse-ndr4	mouse-PPT2	mouse-pafa
mouse-q7m759	mouse-ndr1	mouse-ppt	mouse-FASN
mouse-Q99JW1m	mouse-ndr3	mouse-q3uw77	mouse-psplip
mouse-lypla1	mouse-ndr2	mouse-1plip	mouse-LIPH
mouse-lypla2	mouse-Q923D7	mouse-1plrp	mouse-PPME1
mouse-lypl1	mouse-1neur	mouse-cttli	mouse-tssp
mouse-Q8VCV1	mouse-2neur	mouse-lcat	mouse-pcp
mouse-Q80UX8	mouse-3neur	mouse-C87498	mouse-dpp2
mouse-ACP33	mouse-4neur	mouse-notum	mouse-ppce
mouse-MEST	mouse-Tex30	mouse-q3uuq7	mouse-Q99KJ9
mouse-MGLL	mouse-Kansl3	mouse-srac1	Ensemblmm class=Q9CWI4

Table 4.1: Genes encoding for ABHD containing proteins used for analysis

4.2 Database searches

4.2.1 Retrieval of UniProt Accession numbers and Protein sequence

To be able to analyse the protein structures it was necessary to retrieve their sequence. To achieve that the UniProt accession numbers for the proteins encoded by a ABHD gene locus needs to be acquired. For this task the ESTHER database [28] - a database specialised on α,β hydrolases - was used. For each gene locus there is an entry, containing among other information the UniProt accession number for the protein encoded by the gene. The accession number was used to look up proteins sequences in the Uniprot Database [29]. This was achieved using a python script:

Code 4.1: Python script to acquire the trembl codes of ABHD containing proteins

```
import untangle
import urllib3

liste = open('FILEPATH').read().splitlines() #Opens file with genlist

#This section reads in the gene locus name and generates a list of URLs for each gene locus:
listofurls = [ ]

n=0
while(n < len(liste)):

    a='http://bioweb.ensam.inra.fr/ESTHER/xml?name='
    b=liste[n]
    c=';class=Gene_locus'
    d=a+b+c
    listofurls.append(d)
    n=n+1


#This section opens the url and parses the XML-files. The Uniprot AN is written into a file.
listofuniprotan=[ ]

co=0
while co < len(listofurls):
    xmlpath=listofurls[co]
    obj = untangle.parse(xmlpath)
    listofuniprotan.append([Trembl['value'] for Trembl in obj.Gene_locus.Database.Trembl])
    co=co+1

print(listofuniprotan)
```

The script works as following:

A list is created by reading in a textfile - containing all gene names - storing each gene as an element of the list. Next a URL for the XML-entry of each gene is created. The XML entries are opened and parsed. The entry in the "Trembl['value']" branch is written into a list.

After retrieving a list of UniProt accession numbers the protein sequences were downloaded and saved as FASTA-files.

4.2.2 Structural Information

In the ESTHER database all known structural information for ABHD containing proteins is stored. To find out if there is structural information available for the proteins of interest, the ESTHER database was scanned. To do that a python script was coded:

Code 4.2: Python script to find structural information for a list of ABHD containing proteins in the ESTHER database

```
import urllib.request
import re

genlist = open('FILEPATH').read().splitlines() ##List of Gene

# This section reads in the gene locus name and generates a list of ulrs for each gene locus:
listofurls = []

n=0
while(n < len(genlist)):

    a='http://bioweb.ensam.inra.fr/ESTHER/gene_locus?name='
    b=genelist[n]
    c='&class=Gene_locus'
    d=a+b+c
    listofurls.append(d)
    n=n+1


n=0
nostructure=[]
others=[]

while n < len(listofurls):
    link=listofurls[n]
    a=urllib.request.urlopen(link).read()
    test=a.decode("utf-8")
    abc=re.findall('No structure',test)
    print(abc)

    if len(abc)==0:
        nostructure.append(link)
    else:
        others.append(link)


    print(n)
    n=n+1
```

```
print('nostructure', len(nostructure))

outfile =open('OUTFILEPATH', 'w')
outfile .write("\n".join(nostructure))

print('others', len(others))

outfile2 =open('OUTFILEPATH', 'w')
outfile2 .write("\n".join(others))
```

The script performs the following tasks:
A textfile, which includes all gene names, is opened as a list, every gene is saved as an element of the list. In the next step the corresponding URL is created for each gene entry in the ESTHER database. Then the url is opened and the webpage is searched for the keywords 'No structure', which means that no experimental 3D structure is available for the proteins. There is a experimental 3D structure available if 'No structure' is missing in the text. Only for three proteins a crystal structure could be found.

4.2.3 Annotating already known functions to the proteins

The QuickGo database [30] was scanned for known functions of the ABHD containing proteins. It was only looked for experimental evidence (evidence derived from assay experiments, which is indicated as "IDA" in the database).

4.3 Homology Models

Since not for all proteins structural information is available, homology models had to be created. This was done using YASARA. [31]
The homology model generation was performed using the following parameters:

- Modeling speed (slow = best): Slow
- Number of PSI-BLAST iterations in template search (PsiBLASTs): 1
- Maximum allowed (PSI-)BLAST E-value to consider template (EValue Max): 0.001
- Maximum number of templates to be used (Templates Total): 4

- Maximum number of templates with same sequence (Templates SameSeq): 1
- Maximum oligomerization state (OligoState): 4 (tetrameric)
- Maximum number of alignment variations per template: (Alignments): 5
- Maximum number of conformations tried per loop (LoopSamples): 50
- Maximum number of residues added to the termini (TermExtension): 10

The homology model generation was done automatically using the YASARA build-in macro.

4.4 Cavity Calculation

Cavity determination was done using Catalobase. Catalobase computes the cavity size and properties, saving the information in form of a point cloud. The following property point-clouds got calculated:

- Aromatic carbon point-cloud
- Carbon point-cloud
- Hydrogen-bond donor point-cloud
- Non-hydrogen bonding nitrogen point-cloud
- Nitrogen as H-bond acceptor point-cloud
- Oxygen as H-bond acceptor point-cloud
- Sulfur as H-bond acceptor point-cloud
- Phospor point-cloud
- Desolvation point-cloud
- Accessibility point-cloud
- Electrostatics point-cloud
- Hydrophobicity point-cloud
- Flexibility point-clouid
- Chains point-cloud
- Sulfur point-cloud

- Bromine point-cloud
- Chlorine point-cloud
- Flourine point-cloud
- Iodine point-cloud

4.5 Cavity Matching

After calculating the cavities a cavity matching experiment was performed. All cavities of one protein are compared to cavities of another protein based on the properties listed above. For comparing the cavities the ICP (Iterative Closest Point) algorithm was used. For each property, a score indicating the similarity, is generated. This was done for all proteins.
The overall similarity of cavities is quantified in the total score.

4.6 Reduction of the Dataset

Before the LSQMAN-experiment was performed the relatively large dataset was reduced. The ESTHER database contains more than one protein entry per gene locus. Since many of the proteins are just fragments of other proteins, encoded by the same gene locus the dataset was reduced to one protein per gene locus. If more protein entries per gene locus were available proteins with sequence existence 1 - which means that there is experimental evidence for the proteins existence - and sequence version 1 were preferred.
Here is the list of proteins used for further analysis:

A2A7Z8	P17892	Q6IE26	Q8R197
A2AGI2	P19096	Q6NS59	Q8R1G2
A2AKK5	P23953	Q6NXK7	Q8R2Y0
A2AN96	P28843	Q6P2K2	Q8VCC2
A2RSY1	P34914	Q6P8U6	Q8VCF2
A6H695	P54310	Q6PDB7	Q8VCT4
A7E1Z3	P97321	Q6PE15	Q8VCU1
B0F2B4	P97823	Q6WQJ1	Q8VCV1
B6DQM2	Q03311	Q791M3	Q8VDG7
C0LQ91	Q08ED5	Q7M759	Q8VEB4
D3YU06	Q3TUU5	Q7TMR0	Q8VI78
D3YY49	Q3U213	Q80UX8	Q91WC9
D3YYS6	Q3U3G8	Q80YA7	Q91WG0
D3Z298	Q3U4B4	Q80YU0	Q91WU0
D3Z383	Q3U6J9	Q80Z65	Q91WU4
D3Z5G7	Q3U7M5	Q8BK48	Q91X34
D3Z608	Q3UFF7	Q8BLF1	Q91YQ6
E9PV38	Q3UT41	Q8BM14	Q920A5
E9PW22	Q3UUQ7	Q8BM81	Q99JW1
E9PWH0	Q3UW56	Q8BTG7	Q9CPP7
E9PWK1	Q3V2H7	Q8BVA5	Q9CV37
E9PWX1	Q4VBW7	Q8BVG4	Q9D0Z3
E9PYP1	Q543B9	Q8BVQ5	Q9D3S9
E9QK34	Q544I1	Q8BWN8	Q9D7E3
E9QN99	Q545R3	Q8CGR9	Q9D950
F6Z9B9	Q5BKQ4	Q8CIV3	Q9DAI6
G3INR2	Q5SZ30	Q8K2A6	Q9DBL9
G3UZN6	Q60963	Q8K4F5	Q9ET22
H3BL34	Q63880	Q8QZR3	Q9QYG0
O08710	Q69ZI4	Q8R0P8	Q9QYR7
O35448	Q69ZK2	Q8R0W5	Q9QZC8
O55137	Q69ZK9	Q8R116	Q9R0P3
P11152	Q6AW46	Q8R146	Q9WTL7
Q8R164			

Table 4.2: List of proteins used for further analysis

4.7 LSQMAN

LSQMAN is a tool for protein alignment.[32] It carries out least square alignment of two proteins. The LSQMAN experiment was carried out for all proteins listed in table 4.2.

A pairwise sequence alignment of all proteins is performed, generating a sequence identity score for every protein to every other protein.

A structural alignment is also performed for all proteins. For this the proteins are also compared pairwise. All chains of a protein are compared to all chains of the other protein. The highest similarity match is used as the structural similarity for the two proteins. The structural similarity is indicated in percent.

4.8 Dendrogram generation and Clustering

Based on the sequence alignment and structural alignment data dendrograms were generated. The dendrograms were cut at a specific position to create clusters. This was done by R using the following code:

Code 4.3: R code to generate a dendrogram using the complete linkage method and cut it into clusters

```
# Loading in the Data:

lsq <− read.csv(”LSQman−results−strucid.csv”, sep=”,”)
row.names(lsq) <− lsq$Name
lsq <− lsq[,2:119]
lsq_matrix <− data.matrix(lsq)

#Clustering:

hc=hclust(dist(lsq_matrix), method=”complete”)
hcd = as.dendrogram(hc)

#Colored Dendrogram

labelColors = c(
”red”,
”blue”,
”green”,
”purple”,
”black”,
”orange”
)

# cut dendrogram in 4 clusters
clusMember = cutree(hc, 6)
# function to get color labels
colLab <− function(n) {
    if (is.leaf(n)) {
        a <− attributes(n)
        labCol <− labelColors[clusMember[which(names(clusMember) == a$label)]]
        attr(n, ”nodePar”) <− c(a$nodePar, lab.col = labCol)
    }
    n
}

png(”dendro_complete_6_horiz.png”, # create PNG for the heat map
  width = 20∗113,                   # 5 x 113 pixels
  height = 20∗113,
  res = 113,                        # 113 pixels per inch
  pointsize = 12)

# using dendrapply
clusDendro = dendrapply(hcd, colLab)
# make plot
plot(clusDendro,
main=’Hierarchical Clustering based on Structure Similarity’,
cex.main=3
,horiz = TRUE)

dev.off()
```

The first part of the script reads in the similarity matrix; the range of the matrix and the name of the rows are defined. Then the distance between the elements within the matrix is calculated using the hclust function of R. In the last part the dendrogram is generated and save as an image. It is defined which method should be used for clustering - in this case the complete-linkage method was used, which is an agglomerative clustering method - and it is also defined in how many clusters the dendrogram should be cut into.

It was necessary to make sure that the tree was cut at the right height. Especially for the structural similarity clusters visual control is possible. The structures of proteins of one cluster could be aligned and the quality of the alignment was investigated. For too broad clusters alignments as the following can be found:

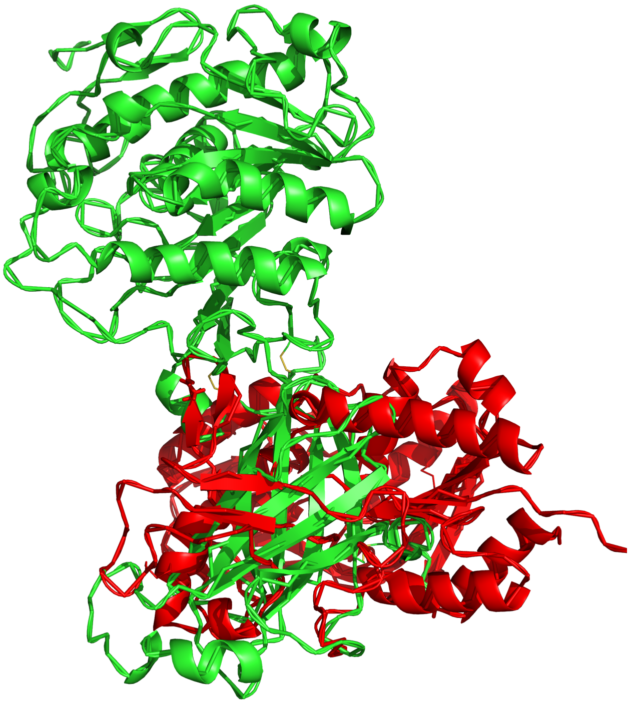

Figure 4.1: Example of aligned proteins of a too broad structural similarity cluster

After introduction of more clusters by cutting the dendrogram at a different height, the structural similarity of proteins - belonging to one cluster - gets higher. This reflects also in the structural alignment:

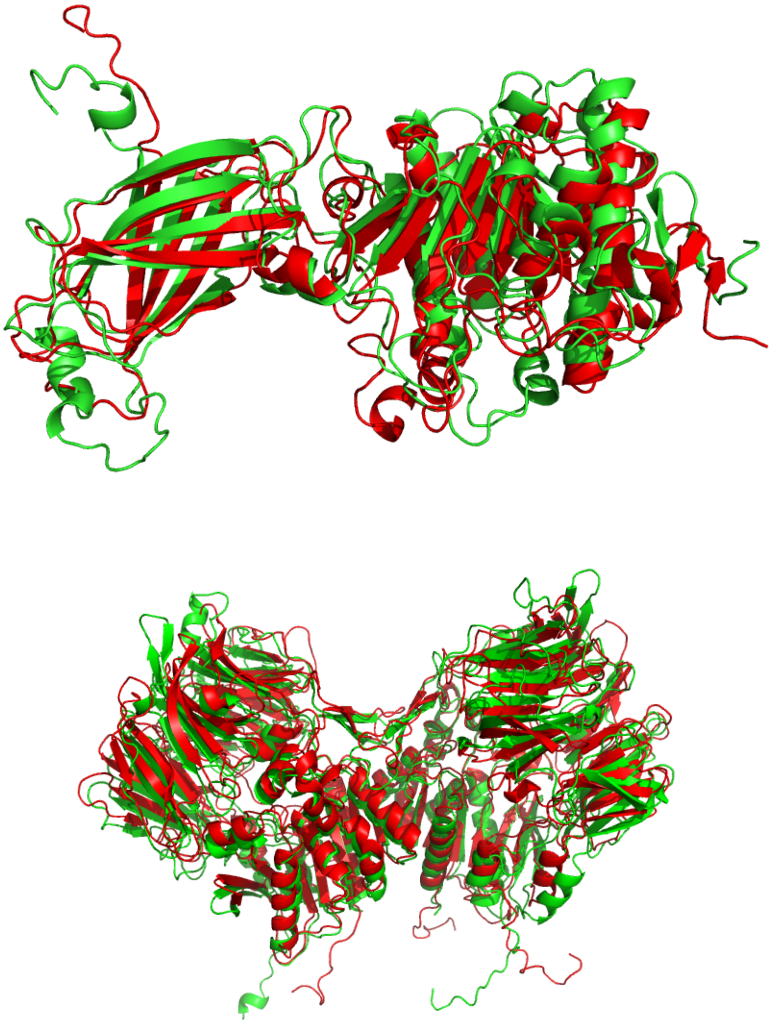

Figure 4.2: Example of protein alignement of proteins belonging to better defined structural clusters

The proteins which belonged into one cluster but did not show high enough structural similarity, are sorted into two different clusters. This leads to higher structural similarity within the clusters.

To gain high structural similarity within a cluster the dendrogram was cut into 6 clusters.

4.9 Correlation of sequence identity and structural similarity

The correlation of sequence identity and structural similarity was investigated. Therefore the sequence identity and structural similarity of all protein pairs were plotted (Figure 5.7 in the results part).

Additionally the sequence identity of proteins within a structural cluster was analysed and visualised in form of histograms. This was done using a python script. The script is not shown here because of its length. It is listed and explained in the appendix in section 8.2.

4.10 Selection of the active site cavities

To determine the active site cavities the proteins structures were visually analysed using PyMol [33]. The cavity nearest to the nucleophile, acid and histidine motif in the α-β hydrolase fold was selected as an active site cavity. An example for an active site cavity, chosen this way, is shown below:

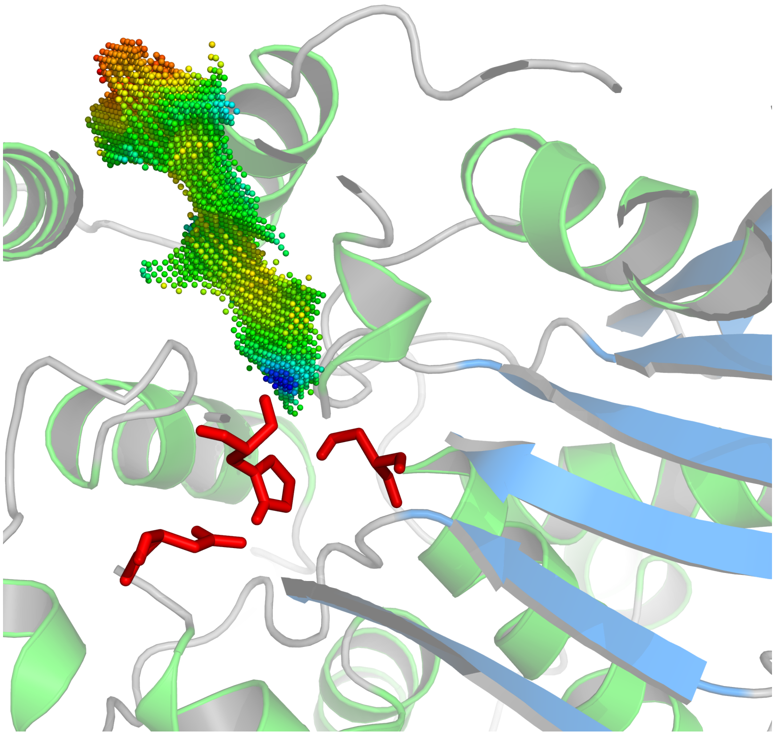

Figure 4.3: Example of active site cavity in Arylacetamide deacetylase-like 3 (Trembl-code:A2A7Z8): The cavity reaches from the surface of the protein to the nucleophile-acid-histidine motif and points at serine and histidine

This was done to be able to compare the proteins based on their active site cavity. For the comparison (clustering) the cavity matching score was used. It was not possible to select an active site cavity for all proteins. In many cases no cavity could be found because no cavity was calculated near the catalytic triad. For these proteins no cavity was selected and they were not used for dendrogram generation and clustering.

For 48 proteins an active site cavity could be identified.

The following active side cavities were used for dendrogram generation:

Protein	Cavity UUID	Protein	Cavity UUID
E0CXH4	F8FE5DAD-BA69-4150-89C0-422BC3164832	Q63880	39A36F77-4AD6-43DF-884D-879D150D3859
Q8R116	45574545-9642-4157-89EA-389C91885439	Q6PE15	36DFDEC4-A531-4977-B0DB-CD151FE4D91A
A2A7Z8	59972D88-2942-46EA-B9D2-B54782B8D971	Q80Z65	8B2C66C6-6F6B-4FD5-931E-C24ECEBA069D
Q8BVG4	63315881-6BF5-4497-88A6-887F4B24E00C	Q8VCC2	45E04564-D8FA-4422-9F3D-5809C8DED11D
A7E1Z3	60362568-F043-4C02-B33A-FF1EEABE985A	Q8VCT4	49190031-99F7-418F-ACBE-B2E02D554DBE
G3UZN6	029A108C-0740-4904-8804-7FFF86BA150E	Q8VCU1	2F9BAC54-9FBA-4C3F-B47B-DE60ED2650C2
P97823	BFE22EEE-B41C-4AC4-8C23-341A4CE8A9A1	Q9CPP7	F46D31D1-F6A1-45BF-AF45-BB5817637BEF
Q3TUU5	6E558B78-D23C-4C0E-ADB4-66AC3699559F	Q9QYR7	68655ACC-B8EA-405F-97C9-9718BC404D30
Q3U5U0	B6A0DD2B-8D52-4D2F-AF27-EF95219C121F	A8JYK8	E73E7FC4-1A3C-40C4-A5B3-B3AB86D44AB6
Q6IE26	54F7D3DE-8043-46BA-A500-EE759F4A2925	A2AN96	CFEB873B-E5EB-44C7-9C05-D4DF86F717C8
Q6P2K2	E36ED8AE-EA3F-4F41-8BFC-8529A0E24789	D3YYS6	959902CB-C681-4317-8336-A229D3CA89D5
Q6Q2Z6	7BF3BA78-446A-441D-8635-C273AAC2D137	D3Z5G7	1F68BAE8-2720-4502-8375-95F91808C846
Q7M759	2B5A60CD-FE4F-4E82-BBF0-BC4E55AB4DF5	E9PV38	6B3BAFA9-C97E-410C-B87D-B69A3036F9E1
Q80UX8	12A67589-F1E8-432D-BC6F-1B6CDEFCDDD8	E9QK34	A0857CDF-D7CE-474F-9162-547785F49BAB
Q8R197	BD5CC167-62B8-43BB-A0D9-49405272E280	O35448	92387321-B021-41F6-8C37-23ECD0F9FD08
D3YU06	7E7CD12A-99DF-4F7F-973B-FD0CE66D8FD7	P23953	27947A1C-0430-4563-8022-B73569172839
E9PYP1	611A2D23-3FA3-4CB3-9A39-B35951A89A42	Q6PDB7	538BA1F1-15EC-44FB-A4D6-C3298C31F221
H3BL34	BA4C0916-2617-48C4-BA19-D17C2112DB7C	Q791M3	C887DABF-1078-487D-A127-617561BC0E3F
P54310	B2E9BEF6-882E-4F70-AF00-7C10AB0E6538	Q91WG0	3EF7EC44-E7A4-4CF2-AA4E-3AD35E82A0AB
Q3U0K4	43E5B0E0-D03B-49C4-A7AB-C5DF89F18A60	Q6AW46	B2F149D6-A4CA-4A73-B54F-926C2898F1C5
Q3U4B4	DB7DC950-2EF6-45F3-8305-9B6312FCA0AF	Q8BM81	5C8E4E7F-A526-4577-8306-BC6460206207
Q3UFF7	CFFA7A88-CB27-4CFB-BD94-B85D33B7CCF6	Q8R0P8	9183405A-4E11-4880-B3D1-F382FF5E6E0D
Q3UW56	B99F535F-3DFE-4C05-95D6-E2B855CB1876	B0F2B4	58F473FD-4464-4707-B42F-7FEEAB1D0FF8
Q60963	5A4E3583-A152-473B-8356-1FE1146BCF90	Q9ET22	AC0FC919-8433-499B-9FB5-82877288BBD1

Table 4.3: List of cavities used for dendrogam generating

5. Results

5.1 Generated Homology Models

As mentioned before one entry for a gene locus contains more than one protein sequence for the same gene. This is because isoforms and also fragments are incluced. In the first step homology models for all proteins were generated.
It was possible to create homology models for 264. For 37 proteins no homology models could be generated. This is the case if no 3D structure of a protein with sufficient sequence identity can be found. So no template for homology model generation is available.
Since only one protein was used per gene 118 homology models were used for the calculations.

5.1.1 Quality

The quality rating of the homology models is based on the Z-scores. The Z-score is in the YASARA report defined as follows:

*"A Z-score describes how many standard deviations the model quality is away from the average high-resolution X-ray structure. Higher values are better, negative values indicate that the homology model looks worse than a high-resolution X-ray structure. The overall Z-scores for all models have been calculated as the weighted averages of the individual Z-scores using the formula Overall = 0.145*Dihedrals + 0.390*Packing1D + 0.465*Packing3D. The overall score thus captures the correctness of backbone- (Ramachandran plot) and side-chain dihedrals, as well as packing interactions. It applies to globular proteins only, and can be mislead by artificial structures like long single alpha helices (which have*

perfect dihedrals and are free of packing errors, since there is no packing)."

The models are grouped into three groups (good, satisfactory and bad) according to their Z-score. Most of the homology models show good or satisfactory quality:

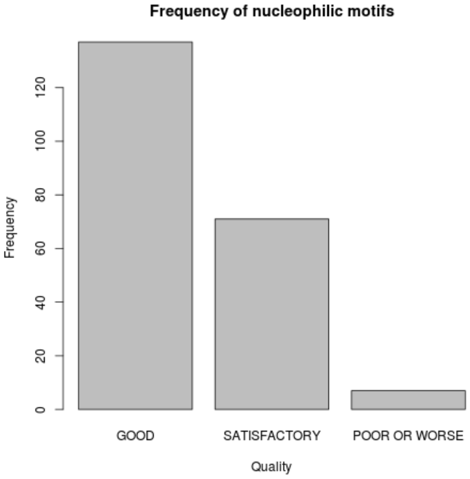

Figure 5.1: Quality of all generated homology models

For one protein the crystal structure is available but still was not used as a template for homology model creation. [1] So it is possible to compare the experimental crystal structure of the protein with the homology model.

[1] The reason that the crystal structure of the protein was not used as a template is most likley that the crystal structure lacks a short variable terminal region, so the sequence identity was not 100 % and another template was used

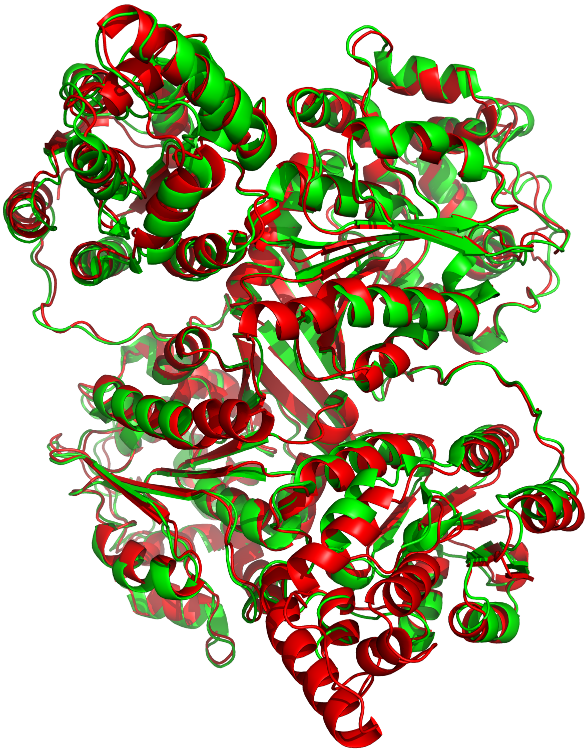

Figure 5.2: Crystal structure of murine soluble epoxide hydrolase from mus musculus (pdb-code:1CR6) aligned with the homology model of murine soluble epoxide hydrolase [based on the template: soluble epoxide hydrolase from homo sapiens (pdb-code:5AM4)] [34]

The x-ray structure and the homology model align very well, indicating good model quality. This alignment could of course only be done for this protein but this result is indicating good homology model quality. Still side chain arrangements could be a problem. The 118 homology models were generated from 67 different templates. This 67 different templates are 61 unique structures.

5.2 Hierarchical Clustering

To identify clusters of similar proteins hierarchical clustering was used. Dendrograms were generated as described in the methods part and cut at a specific height to generate clusters.

5.2.1 Dendrogram based on sequence alignment

Based on the sequence identity a dendrogram was generated:

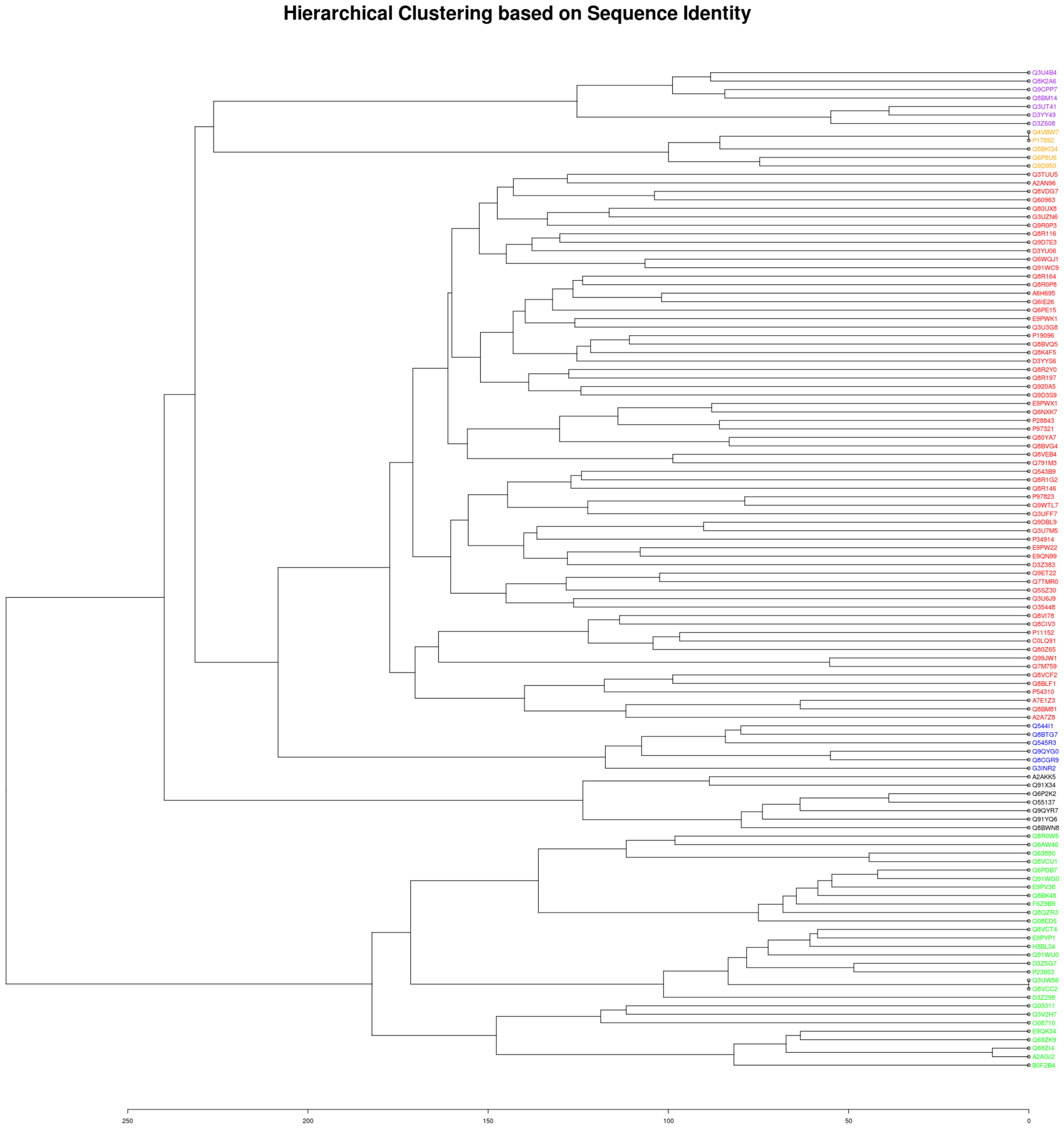

Figure 5.3: Dendrogram of Sequence Identity

The clusters were scanned for proteins with a known functions. This was done using a Python script:

Code 5.1: Python script to identify the function of proteins used for clustering

```
proteins=open(’Filepath to proteinlist ’).read(). splitlines ()
pinc=[]
for n in range(0,len(cluster)):
    with open(’Filepath to function information’) as search:
        for line in search:
            if line [0:6] in proteins[n]:
                pinc.append(line)
print(pinc)
```

This script reads in the protein names and prints out the proteins (which function is known) and their function.

In different cluster the following protein functions can be found:

	Protein	**Function**
Cluster 1	P11152	triglyceride lipase activity
	P19096	fatty acid synthase activity
	P97321	peptidase activity
	Q8R116	palmitoleyl hydrolase activity
	Q8R2Y0	phospholipase activity
	Q8VEB4	calcium-independent phospholipase A2 activity
	Q8VI78	phosphatidylcholine 1-acylhydrolase activity
	Q920A5	serine-type carboxypeptidase activity
	Q9R0P3	hydrolase activity acting on ester bonds
	Q8BLF1	serine hydrolase activity
	Q8VCF2	triglyceride lipase activity
Cluster 3	P23953	carboxylic ester hydrolase activity
	Q03311	acetylcholinesterase activity
	Q8BK48	carboxylic ester hydrolase activity
	Q8QZR3	carboxylic ester hydrolase activity
	Q8VCT4	triglyceride lipase activity
	Q91WG0	acylcarnitine hydrolase activity
Cluster 5	A2AKK5	N-acyltransferase activity
	O55137	acyl-CoA hydrolase activity
	Q8BWN8	acyl-CoA hydrolase activity
	Q91X34	N-acyltransferase activity
	Q9QYR7	acyl-CoA hydrolase activity
Cluster 6	P17892	triglyceride lipase activity
	Q4VBW7	triglyceride lipase activity
	Q6P8U6	triglyceride lipase activity

Table 5.1: Function in clusters of dendrogram based on sequence identity clusters

The proteins belonging to cluster 1 and 3 are rather diverse. Cluster 5 and 6 show proteins with identical functions. The function of the proteins which belong to Cluster 2 and 4 are not yet known.

5.2.2 Dendrogram based on structure alignment

Based on the structure similarity percentage a dendrogram was generated and cut into clusters. The members of one cluster show high structural similarity.

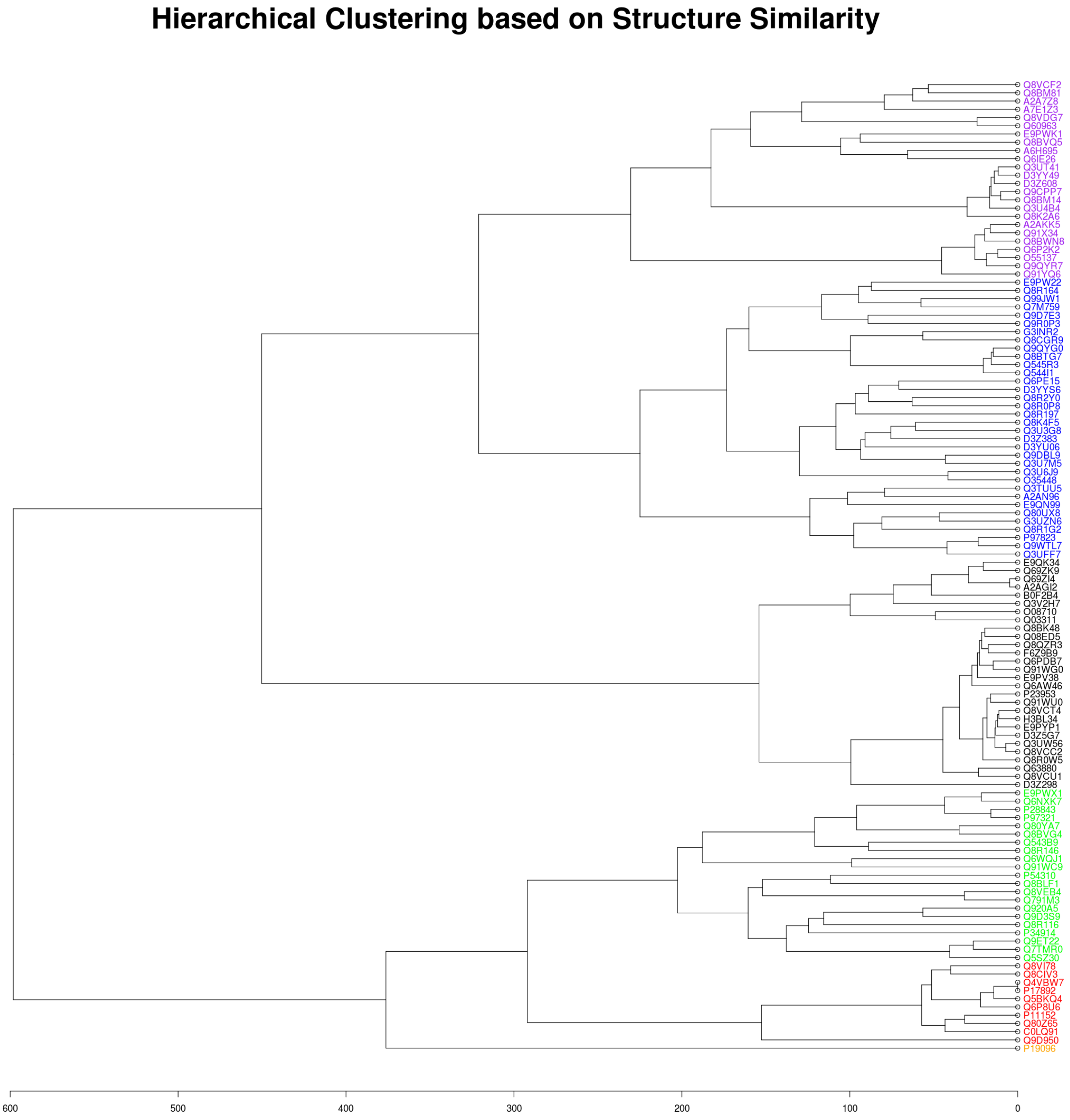

Figure 5.4: Dendrogram of Structure Similarity

Assignment of functions to structure similarity clusters

The clusters were scanned for proteins, with already known functions. This result can be found:

	Protein	**Function**
Cluster 1	A2AKK5	N-acyltransferase activity
	O55137	acyl-CoA hydrolase activity
	Q8BWN8	acyl-CoA hydrolase activity
	Q91X34	N-acyltransferase activity
	Q9QYR7	acyl-CoA hydrolase activity
	Q8VCF2	triglyceride lipase activity
Cluster 2	Q8R2Y0	acylglycerol lipase activity
	Q9R0P3	hydrolase activity acting on ester bonds
Cluster 3	P23953	carboxylic ester hydrolase activity
	Q03311	acetylcholinesterase activity
	Q8BK48	carboxylic ester hydrolase activity
	Q8QZR3	carboxylic ester hydrolase activity
	Q8VCT4	sterol esterase activity
	Q91WG0	acylcarnitine hydrolase activity
Cluster 4	P97321	peptidase activity
	Q8R116	palmitoleyl hydrolase activity
	Q8VEB4	calcium-independent phospholipase A2 activity
	Q920A5	serine-type carboxypeptidase activity
	Q8BLF1	phosphate ion binding
Cluster 5	P11152	triglyceride lipase activity
	P17892	triglyceride lipase activity
	Q4VBW7	triglyceride lipase activity
	Q6P8U6	triglyceride lipase activity
	Q8VI78	phosphatidylcholine 1-acylhydrolase activity
Cluster 6	P19096	fatty acid synthase activity

Table 5.2: Function in clusters of dendrogram based on structure similarity clusters

Templates

Shown below the structure similarity cluster is shown next to the model names the template names are added:

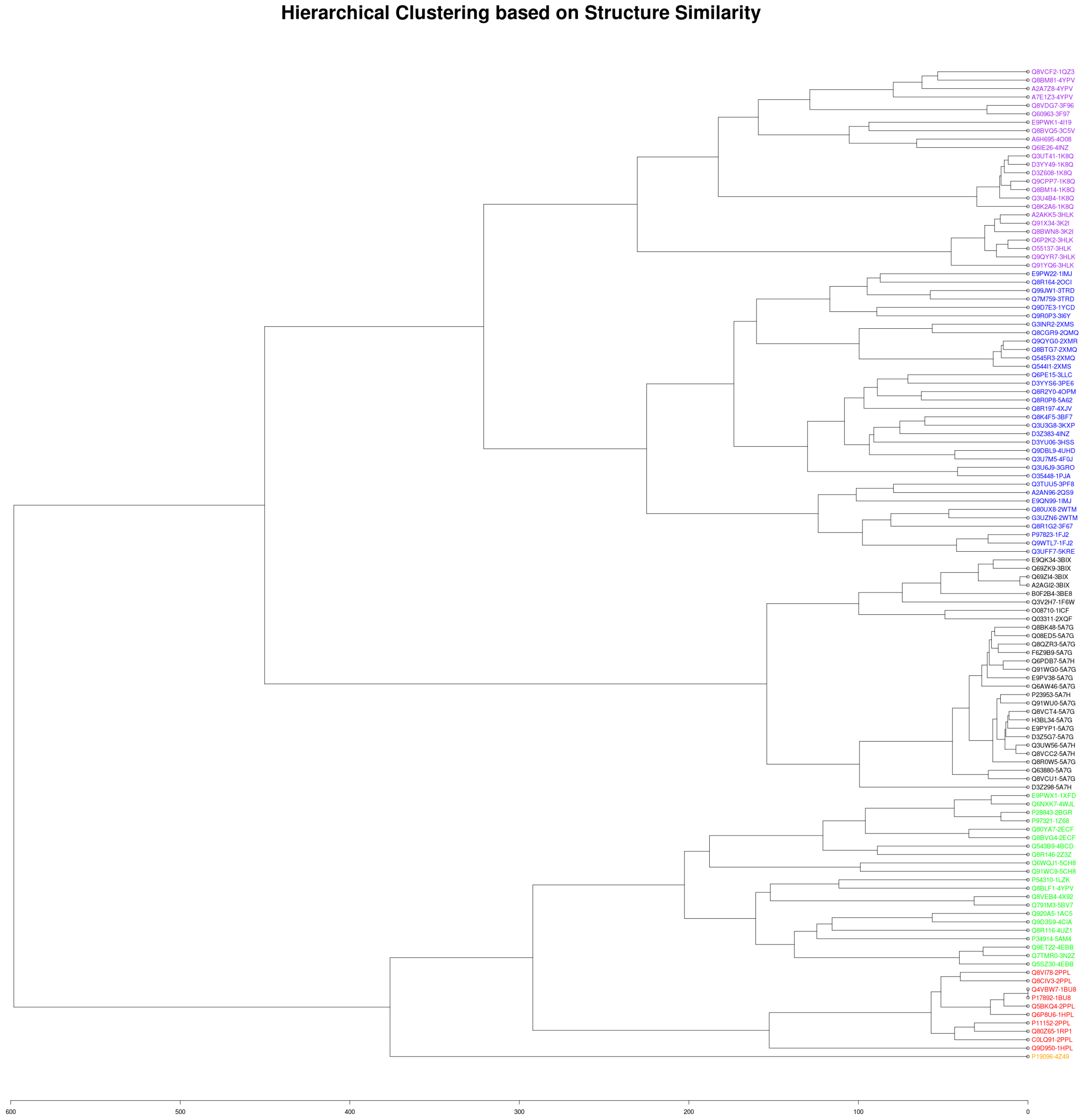

Figure 5.5: Hierarchical Clustering based on Structure Similarity (template names added next to model names)

The frequency of each template within a structure similarity cluster is shown below.

Cluster1		Cluster2		Cluster3	
Template	Frequency	Template	Frequency	Template	Frequency
1K8Q	7	1FJ2	2	5A7G	15
3HLK	5	1IMJ	2	5A7H	5
4YPV	3	2WTM	2	3BIX	4
3K2I	2	2XMQ	2	1F6W	1
1QZ3	1	2XMS	2	1ICF	1
3C5V	1	3TRD	2	2XQF	1
3F96	1	1PJA	1	3BE8	1
3F97	1	1YCD	1		
4I19	1	2OCI	1		
4INZ	1	2QMQ	1		
4O08	1	2QS9	1		
		2XMR	1		
		3BF7	1		
		3F67	1		
		3GRO	1		
		3HSS	1		
		3I6Y	1		
		3KXP	1		
		3LLC	1		
		3PE6	1		
		3PF8	1		
		4F0J	1		
		4INZ	1		
		4OPM	1		
		4UHD	1		
		4XJV	1		
		5A62	1		
		5KRE	1		
Cluster4		Cluster5		Cluster6	
Template	Frequency	Template	Frequency	Template	Frequency
2ECF	2	2PPL	5	4Z49	1
4EBB	2	1BU8	2		
5CH8	2	1HPL	2		
1AC5	1	1RP1	1		
1LZK	1				
1XFD	1				
1Z68	1				
2BGR	1				
2Z3Z	1				
3N2Z	1				
4BCD	1				
4CIA	1				
4UZ1	1				
4WJL	1				
4X92	1				
4YPV	1				
5AM4	1				
5BV7	1				

Table 5.3: Frequency of templates for homology models

Structural alignment of protein within a structure similarity cluster

To investigate the qualtiy of the structure similarity cluster, all proteins within one cluster were aligned. The alignment is shown below:

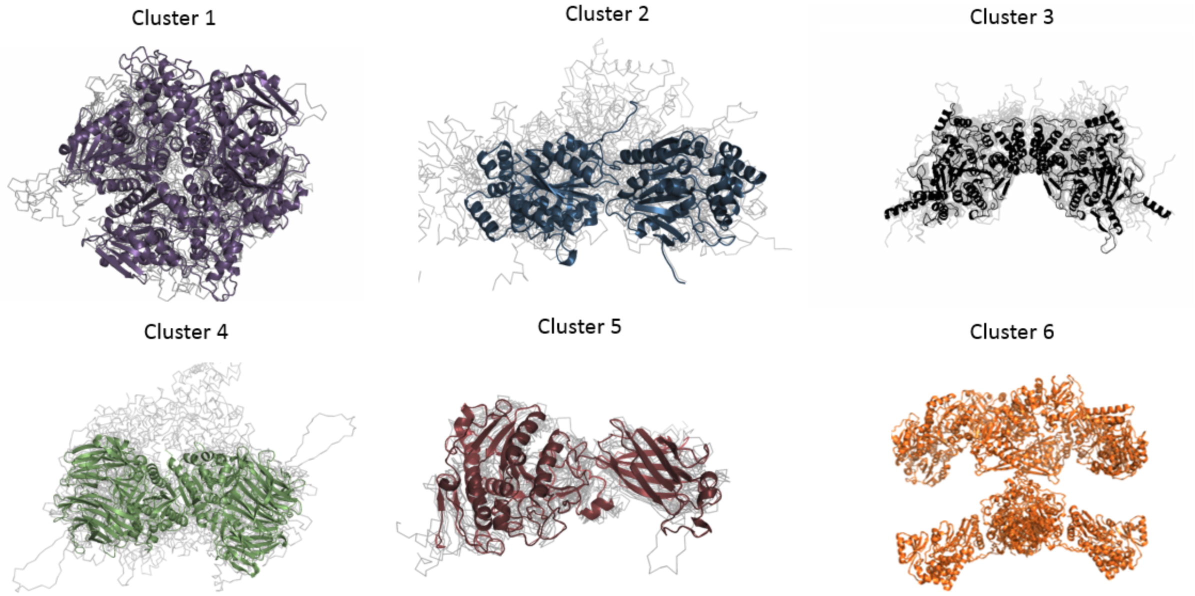

Figure 5.6: Structural alignment of all proteins within a structure similarity cluster generated in Pymol using the cealign algorithm

5.2.3 Relation between structural similarity and sequence identity

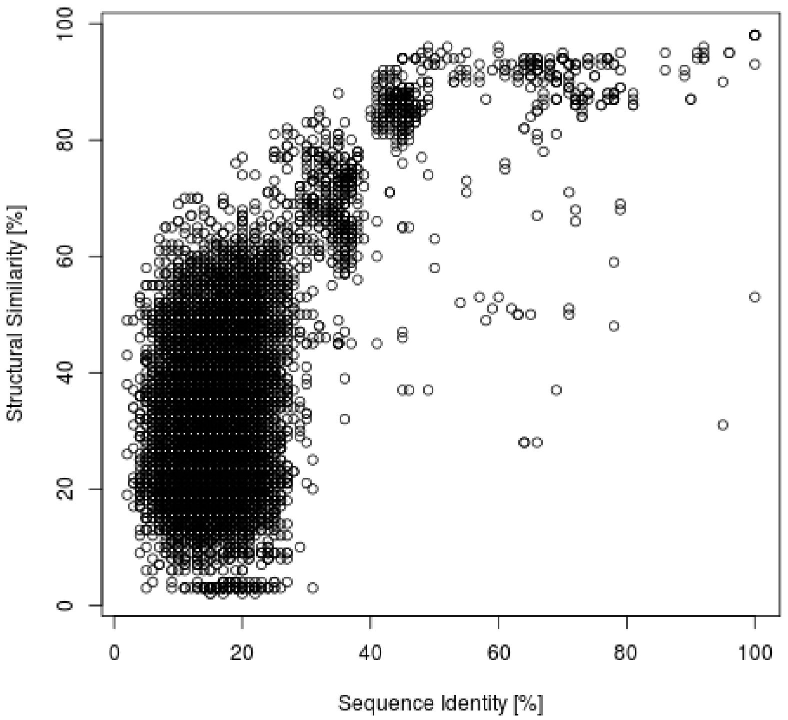

Figure 5.7: Relation of structural similarity and sequence identity

The relation between structural similarity and sequence identity - as shown in figure 5.7 - is not linear. In the area were the sequence identity is lower than 30% the structural similarity can vary between no structural similarity at all and structural similarity up to 70%. At 30% sequence identity the structural similarity increases dramatically.

The relationship between the structural similarity and sequence identity of proteins can also be investigated within the clusters of high structural similarity. Therefore the sequence identity between all proteins belonging to one structural similarity cluster were determined and plotted in histograms:

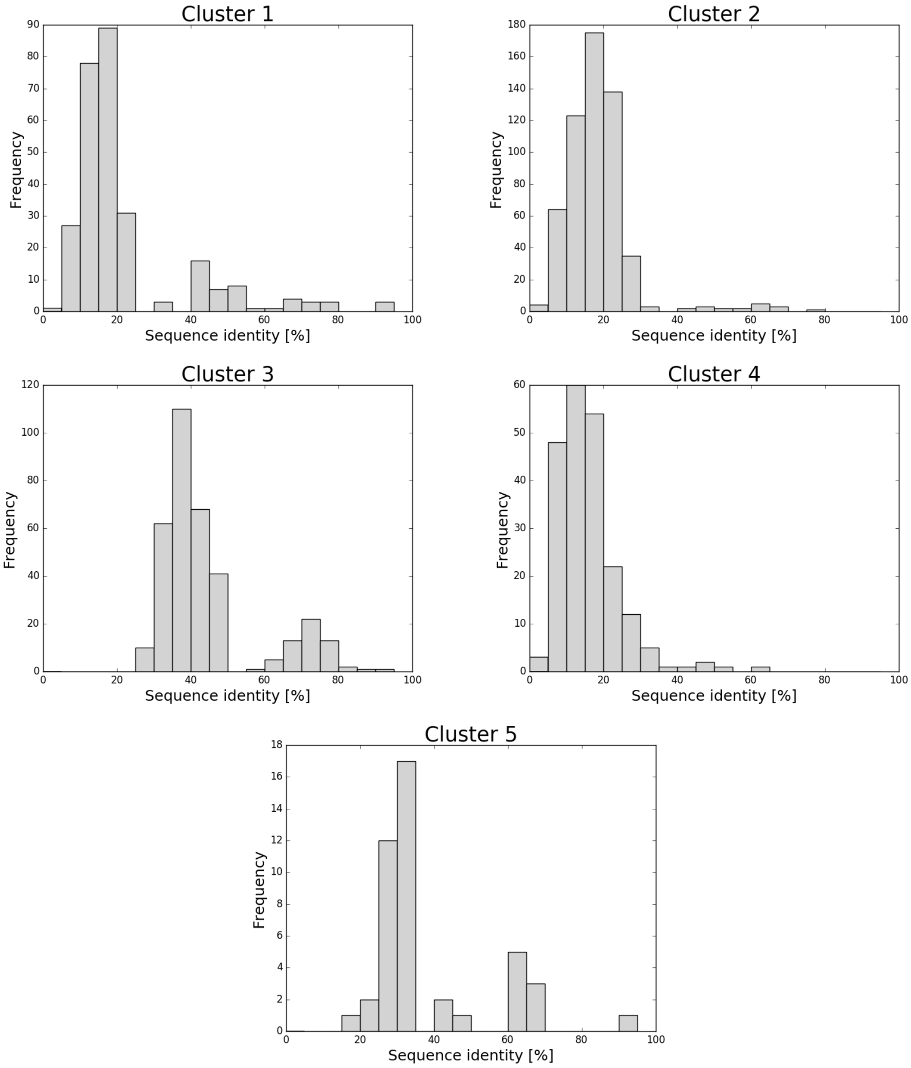

Figure 5.8: Sequence identity of ABHD containing proteins in structure similarity clusters

The distribution of sequence identities within the structural clusters shows that ABHD containing proteins can show high structural similarity despite very low sequence identity. A high sequence identity is therefore no requirement for a high structural similarity.

5.3 Dendrogram based on cavity matching result

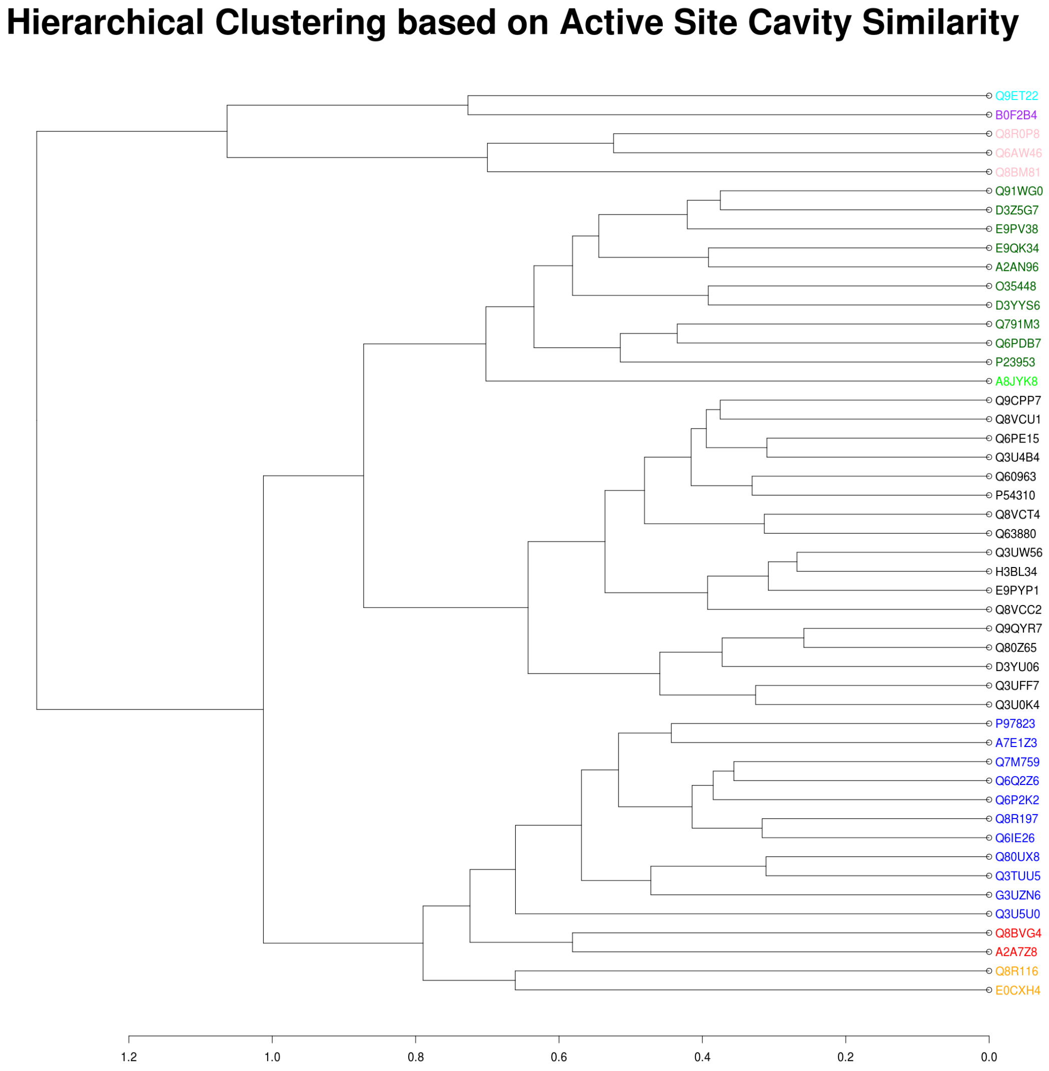

Figure 5.9: Dendrogram of proteins based on cavity matching results

The dendrogram is cut at a specific height to sort the proteins into clusters. 9 clusters were generated. The proteins are coloured according to the clusters they belong to.

Protein	Cluster	Protein	Cluster
E0CXH4	Cluster1	Q63880	Cluster4
Q8R116	Cluster1	Q6PE15	Cluster4
A2A7Z8	Cluster2	Q80Z65	Cluster4
Q8BVG4	Cluster2	Q8VCC2	Cluster4
A7E1Z3	Cluster3	Q8VCT4	Cluster4
G3UZN6	Cluster3	Q8VCU1	Cluster4
P97823	Cluster3	Q9CPP7	Cluster4
Q3TUU5	Cluster3	Q9QYR7	Cluster4
Q3U5U0	Cluster3	A8JYK8	Cluster5
Q6IE26	Cluster3	A2AN96	Cluster6
Q6P2K2	Cluster3	D3YYS6	Cluster6
Q6Q2Z6	Cluster3	D3Z5G7	Cluster6
Q7M759	Cluster3	E9PV38	Cluster6
Q80UX8	Cluster3	E9QK34	Cluster6
Q8R197	Cluster3	O35448	Cluster6
D3YU06	Cluster4	P23953	Cluster6
E9PYP1	Cluster4	Q6PDB7	Cluster6
H3BL34	Cluster4	Q791M3	Cluster6
P54310	Cluster4	Q91WG0	Cluster6
Q3U0K4	Cluster4	Q6AW46	Cluster7
Q3U4B4	Cluster4	Q8BM81	Cluster7
Q3UFF7	Cluster4	Q8R0P8	Cluster7
Q3UW56	Cluster4	B0F2B4	Cluster8
Q60963	Cluster4	Q9ET22	Cluster9

Table 5.4: List of ABHD containing proteins and their assigned cavity similarity cluster

The clusters are compared to the clusters found in the clustering based on structural similarity:

Proteins	Active Site Cavity Similarity Cluster	Strucutral Similarity Cluster
Q8R162	Cluster1	Cluster2
Q8R116	Cluster1	Cluster4
A2A7Z8	Cluster2	Cluster1
Q8BVG4	Cluster2	Cluster4
A7E1Z3	Cluster3	Cluster1
Q6IE26	Cluster3	Cluster1
Q6P2K2	Cluster3	Cluster1
Q6Q2Z6	Cluster3	Cluster1
G3UZN6	Cluster3	Cluster2
P97823	Cluster3	Cluster2
Q3TUU5	Cluster3	Cluster2
Q7M759	Cluster3	Cluster2
Q80UX8	Cluster3	Cluster2
Q8R197	Cluster3	Cluster2
Q60963	Cluster4	Cluster1
Q9QYR7	Cluster4	Cluster1
Q3U4B4	Cluster4	Cluster1
Q9CPP7	Cluster4	Cluster1
Q3U0K4	Cluster4	Cluster1
D3YU06	Cluster4	Cluster2
Q3UFF7	Cluster4	Cluster2
Q6PE15	Cluster4	Cluster2
H3BL34	Cluster4	Cluster3
Q3UW56	Cluster4	Cluster3
Q63880	Cluster4	Cluster3
Q8VCC2	Cluster4	Cluster3
Q8VCT4	Cluster4	Cluster3
Q8VCU1	Cluster4	Cluster3
P54310	Cluster4	Cluster4
Q80Z65	Cluster4	Cluster5
E9PYP1	Cluster4	Cluster3
A8JYK8	Cluster5	Cluster4
A2AN96	Cluster6	Cluster2
D3YYS6	Cluster6	Cluster2
O35448	Cluster6	Cluster2
D3Z5G7	Cluster6	Cluster3
E9PV38	Cluster6	Cluster3
E9QK34	Cluster6	Cluster3
P23953	Cluster6	Cluster3
Q6PDB7	Cluster6	Cluster3
Q91WG0	Cluster6	Cluster3
Q791M3	Cluster6	Cluster4
Q8BM81	Cluster7	Cluster1
Q8R0P8	Cluster7	Cluster2
Q6AW46	Cluster7	Cluster3
B0F2B4	Cluster8	Cluster3
Q9ET22	Cluster9	Cluster4

Table 5.5: List of ABHD containing proteins and their assigned cavity similarity cluster and structural similarity cluster

Proteins	Strucutral Similarity Cluster	Active Site Cavity Similarity Cluster
A2A7Z8	Cluster1	Cluster2
A7E1Z3	Cluster1	Cluster3
Q6IE26	Cluster1	Cluster3
Q6P2K2	Cluster1	Cluster3
Q6Q2Z6	Cluster1	Cluster3
Q60963	Cluster1	Cluster4
Q9QYR7	Cluster1	Cluster4
Q3U4B4	Cluster1	Cluster4
Q9CPP7	Cluster1	Cluster4
Q3U0K4	Cluster1	Cluster4
Q8BM81	Cluster1	Cluster7
Q8R162	Cluster2	Cluster1
G3UZN6	Cluster2	Cluster3
P97823	Cluster2	Cluster3
Q3TUU5	Cluster2	Cluster3
Q7M759	Cluster2	Cluster3
Q80UX8	Cluster2	Cluster3
Q8R197	Cluster2	Cluster3
D3YU06	Cluster2	Cluster4
Q3UFF7	Cluster2	Cluster4
Q6PE15	Cluster2	Cluster4
A2AN96	Cluster2	Cluster6
D3YYS6	Cluster2	Cluster6
O35448	Cluster2	Cluster6
Q8R0P8	Cluster2	Cluster7
E9PYP1	Cluster3	Cluster4
H3BL34	Cluster3	Cluster4
Q3UW56	Cluster3	Cluster4
Q63880	Cluster3	Cluster4
Q8VCC2	Cluster3	Cluster4
Q8VCT4	Cluster3	Cluster4
Q8VCU1	Cluster3	Cluster4
D3Z5G7	Cluster3	Cluster6
E9PV38	Cluster3	Cluster6
E9QK34	Cluster3	Cluster6
P23953	Cluster3	Cluster6
Q6PDB7	Cluster3	Cluster6
Q91WG0	Cluster3	Cluster6
Q6AW46	Cluster3	Cluster7
B0F2B4	Cluster3	Cluster8
Q8R116	Cluster4	Cluster1
Q8BVG4	Cluster4	Cluster2
P54310	Cluster4	Cluster4
A8JYK8	Cluster4	Cluster5
Q791M3	Cluster4	Cluster6
Q9ET22	Cluster4	Cluster9
Q80Z65	Cluster5	Cluster4

Table 5.6: List of ABHD containing proteins and their assigned structural similarity cluster and cavity similarity cluster

The comparison shows that the clustering of the proteins is different using the cavity matching results instead of structural alignment data.

The selection of the active sites revealed proteins, which do not contain a catalytic triade. These proteins are listed here:

Proteins
N-myc downstream regulated gene 1 (Trembl: B7ZWC0)
Protein NDRG4 (Trembl-code:E0CZ50)
Protein NDRG3 (Trembl: Q8CBD0)
Protein NDRG1 (Trembl: E9PVF3)
Inactive dipeptidyl peptidase 10 (Trembl: E9QN98)
Acot5 protein (Trembl: Q91YQ6)
Neuroligin-2 (Trembl: F6VE93)
1-acylglycerol-3-phosphate O-acyltransferase ABHD5 (Trembl: Q9DBL9)
Putative uncharacterized protein (Trembl: Q3TD08)
Protein NDRG2 (Trembl: Q9QYG0)
Sn1-specific diacylglycerol lipase beta (Trembl: Q91WC9)

Table 5.7: List of ABHD containing proteins that lack the catalytic triad

For most of the proteins the reason why they do not show the catalytic triad is that the nucleophile residue and histidine residue are missing. But also a too great distance between the members of the catalytic triad is one reason that a protein does not show the catalytic triad.

Proteins(Trembl-code)	Reason why the protein does not show the catalytic triad
B7ZWC0	The nucleophile and histidine residues are missing
E0CZ50	The nucleophile and histidine residues are missing
Q8CBD0	The nucleophile and histidine residues are missing
E9PVF3	The distance between the residues of the catalytic triad is too great
E9QN98	The nucleophile residue is missing
Q91YQ6	The acid and nucleophile residues are missing
F6VE93	The acid, histidine and nucleophile residues are missing
Q9DBL9	The distance between the residues of the catalytic triad is too great
Q3TD08	The nucleophile and histidine residues are missing
Q9QYG0	The nucleophile and histidine residues are missing
Q91WC9	The distance between the residues of the catalytic triad is too great

Table 5.8: Reasons for lacking the catalytc triad

6. Discussion

6.1 Reasons for not selecting an active site cavity

As mentioned before not for all proteins an active site cavity could be identified. To investigate if using homology models pose a problem the templates for homology models - for which no active site cavities could be identified - were also scanned for an active site cavity.

In homology model generation the side chain orientations are chosen in a way that no clashes occur. Although this is in general useful it can also lead to problems. The side chains are moved into free space so no clashes occur but in this process the cavity gets disrupted. An example is shown below:

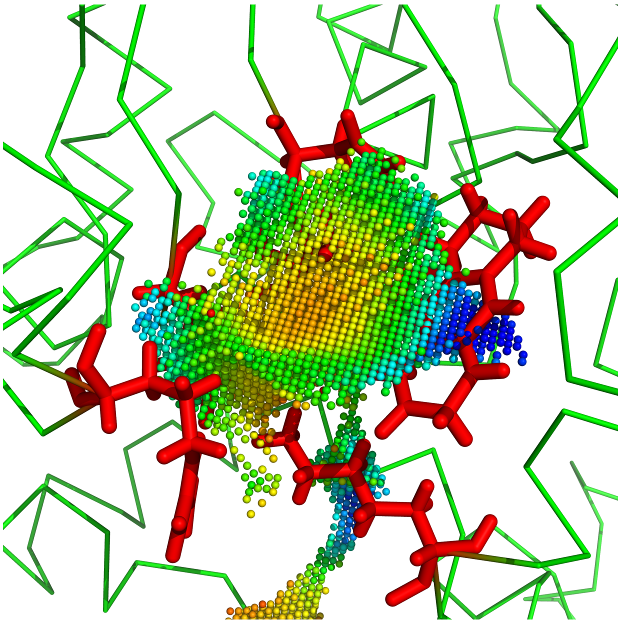

Figure 6.1: In F7AYJ4 no active site cavity could be calculated. To find the reason for this F7AYJ4 was aligned with its template 5EIE which shows an active site cavity. Then the active site cavity was displayed and overlayed with F7AYJ4. This shows that side chains were turned into the free space during homology model generation and that there is no cavity in the homology model.

This is the case for some proteins. But for the most proteins no active site cavity could be identified neither in the homology model nor in the template. This could be dealt by fine tuning the parameters for cavity calculation.

6.2 Relationship of sequence identity and structural similarity

The comparison of the sequence and structural clusters show the expected result. There is no strong correlation between sequence identity and structure similarity. The sequence identity between proteins, with a high structural similarity, can vary. Proteins with very low sequence identity can still show high structural similarity.

6.3 Structural alignment of proteins belonging to one structural similarity cluster

The alignment of the proteins within one structural similarity cluster shows that those proteins show a relative high structural similarity to each other. Cluster 6 only contains one protein, the fatty acid synthase. This is also meaningful since the structure of this proteins varies a lot from all the others.

6.4 Distribution of functions within a structural similarity cluster

The investigation of the distribution of functions within a structural similarity cluster show that the functions within a cluster are diverse. This leads to the assumption that a specific type of structure does not show only one type of function but that proteins showing this type of structure can catalyse different kinds of reactions.

The analysis of the ABHD containing proteins revealed that some ABHD containing proteins do not show the catalytic triad, typical for α-β hydrolases.

6.5 Relationship between structural similarity clusters and active site similarity clusters

The analysis of the clustering of active site cavity matching reveals that one structural similarity cluster can not be assigned to only one cavity similarity cluster and vice versa. There is no correlation between the active site cavity clusters and the structural similarity clusters. Proteins which show a very similar structure can show different active site cavities. And also proteins with different structures can harbour very similar active site cavities.

An analysis concerning the function of the clusters is really difficult since there is a lack of information about the proteins. Only for too few proteins, used for the clustering, the function is known. So no reliable conclusion can be drawn for the correlation between protein clusters and the function of the proteins.

7. Outlook

The current work focused on getting more information about the dataset and to start the studying of the relationship between those proteins. Although the work showed many different interesting results, not all questions are answered yet. So further investigation of the dataset is necessary.
There are still many interesting questions for the future. Studying the correlation between functions and active site cavity clusters is the next step. This might enable assigning functions to ABHD proteins which functions are not yet known.
For this more active site cavities need to be identified and also more functional data must be considered. Since functional assay data for this proteins is relatively rare. So new experimental data need to be added to the dataset constantly and if needed also electronic annotated functions could be used.

Bibliography

[1] Marco Nardini and Bauke W Dijkstra. α/β hydrolase fold enzymes: the family keeps growing. *Current opinion in structural biology*, 9(6):732–737, 1999.

[2] Caleb C Lord, Gwynneth Thomas, and J Mark Brown. Mammalian alpha beta hydrolase domain (abhd) proteins: Lipid metabolizing enzymes at the interface of cell signaling and energy metabolism. *Biochimica et Biophysica Acta (BBA)-Molecular and Cell Biology of Lipids*, 1831(4):792–802, 2013.

[3] Pirkko Heikinheimo, Adrian Goldman, Cy Jeffries, and David L Ollis. Of barn owls and bankers: a lush variety of α/β hydrolases. *Structure*, 7(6):R141–R146, 1999.

[4] Ananda K Ghosh, Geetha Ramakrishnan, Chitraju Chandramohan, and Ram Rajasekharan. Cgi-58, the causative gene for chanarin-dorfman syndrome, mediates acylation of lysophosphatidic acid. *Journal of Biological Chemistry*, 283(36):24525–24533, 2008.

[5] Gabriela Montero-Moran, Jorge M Caviglia, Derek McMahon, Alexis Rothenberg, Vidya Subramanian, Zhi Xu, Samuel Lara-Gonzalez, Judith Storch, George M Carman, and Dawn L Brasaemle. Cgi-58/abhd5 is a coenzyme a-dependent lysophosphatidic acid acyltransferase. *Journal of lipid research*, 51(4):709–719, 2010.

[6] Mats Holmquist. Alpha beta-hydrolase fold enzymes structures, functions and mechanisms. *Current Protein and Peptide Science*, 1(2):209–235, 2000.

[7] Dick B Janssen, Alex Scheper, Lubbert Dijkhuizen, and Bernard Witholt. Degradation of halogenated aliphatic compounds by xanthobacter autotrophicus gj10. *Applied and environmental microbiology*, 49(3):673–677, 1985.

[8] Miriam Stoelting, Marcel Geyer, Stefan Reuter, Rudolf Reichelt, Martin Johannes Bek, and Hermann Pavenstädt. α/β hydrolase 1 is upregulated in d5 dopamine receptor knockout mice and reduces production of nadph oxidase. *Biochemical and biophysical research communications*, 379(1):81–85, 2009.

[9] Jonathan Z Long, Justin S Cisar, David Milliken, Sherry Niessen, Chu Wang, Sunia A Trauger, Gary Siuzdak, and Benjamin F Cravatt. Metabolomics annotates abhd3 as a physiologic regulator of medium-chain phospholipids. *Nature chemical biology*, 7(11):763–765, 2011.

[10] Thomas Shafee. *Evolvability of a viral protease: experimental evolution of catalysis, robustness and specificity.* PhD thesis, University of Cambridge, 2014.

[11] VK Vyas, RD Ukawala, M Ghate, and C Chintha. Homology modeling a fast tool for drug discovery: current perspectives. *Indian journal of pharmaceutical sciences*, 74(1):1, 2012.

[12] Michael Levitt. Accurate modeling of protein conformation by automatic segment matching. *Journal of molecular biology*, 226(2):507–533, 1992.

[13] W J_ Browne, ACT North, DC Phillips, Keith Brew, Thomas C Vanaman, and Robert L Hill. A possible three-dimensional structure of bovine α-lactalbumin based on that of hen's egg-white lysozyme. *Journal of molecular biology*, 42(1):65IN1371–7086, 1969.

[14] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94, 1999.

[15] Zhexin Xiang. Advances in homology protein structure modeling. *Current Protein and Peptide Science*, 7(3):217–227, 2006.

[16] Kenneth M Merz Jr, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc*, 117:5179–5197, 1995.

[17] Jiang Zhu, Hao Fan, Xavier Periole, Barry Honig, and Alan E Mark. Refining homology models by combining replica-exchange molecular dynamics and statistical potentials. *Proteins: Structure, Function, and Bioinformatics*, 72(4):1171–1188, 2008.

[18] Rhiju Das, Bin Qian, Srivatsan Raman, Robert Vernon, James Thompson, Philip Bradley, Sagar Khare, Michael D Tyka, Divya Bhat, Dylan Chivian, et al. Structure prediction for casp7 targets using extensive all-atom refinement with rosetta@ home. *Proteins: Structure, Function, and Bioinformatics*, 69(S8):118–128, 2007.

[19] Rongsheng Han, Alejandra Leo-Macias, Daniel Zerbino, Ugo Bastolla, Bruno Contreras-Moreira, and Angel R Ortiz. An efficient conformational sampling method for homology modeling. *Proteins: Structure, Function, and Bioinformatics*, 71(1):175–188, 2008.

[20] YASARA knowledge-based potentials in yasara. http://www.yasara.org/kbpotentials.htm. Accessed: 04.04.2017.

[21] RW Hooft, Gert Vriend, Chris Sander, Enrique E Abola, et al. Errors in protein structures. *Nature*, 381(6580):272–272, 1996.

[22] structure validation in yasara. http://www.yasara.org/validation.htm. Accessed: 04.04.2017.

[23] Georg Steinkellner, Christian C Gruber, Tea Pavkov-Keller, Alexandra Binter, Kerstin Steiner, Christoph Winkler, Andrzej Łyskowski, Orsolya Schwamberger, Monika Oberer, Helmut Schwab, et al. Identification of promiscuous ene-reductase activity by mining structural databases using active site constellations. *Nature communications*, 5, 2014.

[24] Manfred Hendlich, Friedrich Rippmann, and Gerhard Barnickel. Ligsite: automatic and efficient detection of potential small molecule-binding

sites in proteins. *Journal of Molecular Graphics and Modelling*, 15(6):359–363, 1997.

[25] William E Hart Forli, Scott Halliday, Rik Belew, and Arthur J Olson. Autodock version 4.2. *User Guide*, 2012.

[26] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[27] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[28] Xavier Cousin, Thierry Hotelier, Kurt Giles, Jean Pierre Toutant, and Arnaud Chatonnet. achedb: the database system for esther, the α/β fold family of proteins and the cholinesterase gene server. *Nucleic acids research*, 26(1):226–228, 1998.

[29] Cathy H Wu, Rolf Apweiler, Amos Bairoch, Darren A Natale, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, et al. The universal protein resource (uniprot): an expanding universe of protein information. *Nucleic acids research*, 34(suppl 1):D187–D191, 2006.

[30] David Binns, Emily Dimmer, Rachael Huntley, Daniel Barrell, Claire O'donovan, and Rolf Apweiler. Quickgo: a web-based tool for gene ontology searching. *Bioinformatics*, 25(22):3045–3046, 2009.

[31] YASARA Biosciences. Yasara: Yet another scientific artificial reality application, 2010.

[32] Gerard J Kleywegt and T Alwyn Jones. A super position. *ESF/Ccp4 Newsletter*, 31(9):14, 1994.

[33] Warren L DeLano. Pymol: An open-source molecular graphics tool. *CCP4 Newsletter On Protein Crystallography*, 40:82–92, 2002.

[34] Maria A Argiriadi, Christophe Morisseau, Bruce D Hammock, and David W Christianson. Detoxification of environmental mutagens and

carcinogens: structure, mechanism, and evolution of liver epoxide hydrolase. *Proceedings of the National Academy of Sciences*, 96(19):10637–10642, 1999.

List of Figures

List of Tables

List of Code

Danksagungen

An dieser Stelle möchte ich mich bei Prof. Gruber, für die Möglichkeit der Mitarbeit an einem sehr interessanten Projekt, bedanken. Weiters möchte ich mich bei Christian Gruber und Georg Steinkellner für die Unterstützung während meiner Arbeit und für die sehr angenehme Arbeitsumgebung bedanken.

Meiner Freundin Nadine möchte ich für die moralische Unterstützung und für das Korrekturlesen des Manuskriptes aber auch für die schöne Zeit abseits des Studiums danken.

Ein ganz besonderer Dank gilt meinen Eltern, Ursula und Erich Hetmann. Ohne deren ständige und bedingungslose Unterstützung mein Studium nicht möglich gewesen wäre.

8. Appendix

8.1 Some useful scripts

Here are some useful scripts listed, which were used during the work but aren’t explicitly mentioned in the methods section.

Dictionary

Code 8.1: Python code for a dictionary

```
key = open(’FILEPATH’).read().splitlines()
value = open(’FILEPATH’).read().splitlines()

liste =open(’FILEPATH’).read().splitlines() #List which needs to be translated

dict = dict(zip(key, value))
liste2 =[]

n=0
while(n < len(liste)):
    a=liste[n]
    b=dict[list [n]]
    c=a+’ ’*3+b
    liste2 .append(c)
    n=n+1

outfile =open(’OUTFILE’, ’w’)
outfile .write(”\n”.join(liste2))
```

This script is using the dictionary function of python to translate input. A list of key and a list of the corresponding values is read in. This allows to translate an input which must be in the form of a key.
This was e.g. used to get for each protein the corresponding gene name.

Pattern search

This script was used for scanning the FASTA-files for the typical AB-fold protein pattern.

Code 8.2: Python script for pattern search in a FASTA file

```
import re
import sys

def fasta_read(fastaobj):
     title =””
    seq=””
    datastructure=[]
    for line in fastaobj:
        if line [0]  == ”>”:
            if seq !=””:
                datastructure.append((title,seq))
                seq=””
            title =line.strip()
        else:
            seq=seq+line.strip().upper()
    datastructure.append((title,seq))
    return datastructure

myfasta=open(’FILEPATH’)
fasta=fasta_read(myfasta)

sys.stdout=open(”OUTFILE”,”w”)

pattern = ”G[ˆX][SCD][ˆX]G” ##The search pattern (G must be 1. followed by any AS but not X, then there can be
      a S,C or a D, no X, and at the end there must be a G)

print(’Nucleophe elbow’)
for counter in range(0,156):
    seq=fasta[counter][1]
    if re.search(pattern,seq):
        for match_obj in re.finditer(pattern,seq):
            print(fasta[counter ][0][4:10], match_obj.group(),match_obj.start(), match_obj.end())
    else:
        print(fasta[counter ][0][4:10], ”not found”)
```

Pymol script to highlight amino acids of the catalytic triad

Code 8.3: Pymol script to highlight the amino acids of the catalytic triad

```
select activeas, (ss l and resn ser+cys+asp+glu+his)
color red, activeas
label activeas, resn
```

The script selects serines, cysteins, aspartates, glutamates and histidines , which are located in a loop. The selection is highlighted in red and the residues are labelled.

8.2 Sequence Identity within a Structural Similarity Cluster

Code 8.4: Python script for analysis of sequence identity within a structural similarity cluster

```
import pandas as pd
import numpy as np
import re
import statistics
import matplotlib.pyplot as plt

df = pd.read_csv('LSQman-results-seqid.csv', index_col='Name')

struc1=open('FILEPATH').read().splitlines()
struc2=open('FILEPATH').read().splitlines()
struc3=open('FILEPATH').read().splitlines()
struc4=open('FILEPATH').read().splitlines()
struc5=open('FILEPATH').read().splitlines()
struc6=open('FILEPATH').read().splitlines()

 scoreliststruc1 =[]
n=0
while n < len(struc1):
    a=n+1
    while a < len(struc1):
        e1=struc1[n]
        e2=struc1[a]
         scoreliststruc1 .append(df[e1][e2])
        a=a+1
    n=n+1

 scoreliststruc2 =[]
n=0
while n < len(struc2):
    a=n+1
    while a < len(struc2):
        e1=struc2[n]
        e2=struc2[a]
         scoreliststruc2 .append(df[e1][e2])
        a=a+1
    n=n+1

 scoreliststruc3 =[]
n=0
while n < len(struc3):
    a=n+1
    while a < len(struc3):
        e1=struc3[n]
        e2=struc3[a]
         scoreliststruc3 .append(df[e1][e2])
        a=a+1
    n=n+1

 scoreliststruc4 =[]
n=0
while n < len(struc4):
    a=n+1

    while a < len(struc4):
        e1=struc4[n]
        e2=struc4[a]
         scoreliststruc4 .append(df[e1][e2])
        a=a+1
    n=n+1

scoreliststruc5 =[]
n=0
while n < len(struc5):
    a=n+1
    while a < len(struc5):
        e1=struc5[n]
        e2=struc5[a]
        scoreliststruc5 .append(df[e1][e2])
        a=a+1
    n=n+1

scoreliststruc6 =[]
n=0
while n < len(struc6):
    a=n+1
    while a < len(struc6):
        e1=struc6[n]
        e2=struc6[a]
        scoreliststruc6 .append(df[e1][e2])
        a=a+1
    n=n+1

#Histogram Generation

scoreliststruc1  = list(map(float, scoreliststruc1 ))
n=0
listbin =[]
while n < 100:
    listbin .append(n)
    n=n+5
plt.hist( scoreliststruc1 , bins=listbin, color='lightgray')
plt.show()

scoreliststruc2  = list(map(float, scoreliststruc2 ))
n=0
listbin =[]
while n < 100:
    listbin .append(n)
    n=n+5
plt.hist( scoreliststruc2 , bins=listbin, color='lightgray')
plt.show()

scoreliststruc3  = list(map(float, scoreliststruc3 ))
n=0
listbin =[]
while n < 100:
    listbin .append(n)
    n=n+5
plt.hist( scoreliststruc3 , bins=listbin, color='lightgray')
plt.show()

scoreliststruc4  = list(map(float, scoreliststruc4 ))
n=0
listbin =[]
while n < 100:
    listbin .append(n)
    n=n+5
plt.hist( scoreliststruc4 , bins=listbin, color='lightgray')
plt.show()


scoreliststruc5  = list(map(float, scoreliststruc5 ))
n=0
listbin =[]
while n < 100:
    listbin .append(n)
    n=n+5
plt.hist( scoreliststruc5 , bins=listbin, color='lightgray')
plt.show()

scoreliststruc6 = list(map(float, scoreliststruc6))
n=0
listbin=[]
while n < 100:
    listbin.append(n)
    n=n+5
plt.hist(scoreliststruc5, bins=listbin, color='lightgray')
plt.show()
```

The very first part imports all python packages, which are needed for the analysis. Then the matrix, which contains the sequence identity information of all proteins to each other, is read in. After that, the names of the proteins of each structural similarity cluster are stored in a list.

In the next part all proteins within one cluster are compared to each other and the sequence identity is looked up in the seqid-matrix and stored in a list.

The last part generated a histogram for each cluster.

Assign cavity similarity clusters to structural similarity clusters

The following script reads in two textfiles, in which the protein names and the cavity similarity respectively the structural similarity clusters they are assigned to are saved.

Then a list of the proteins and their belonging structural and cavity similarity clusters are printed out.

Code 8.5: Python script to assign proteins belonging to cavity similarity clusters to structural similarity clusters

```
listecm=open('FILEPATH').read().splitlines()
with open('FILEPATH') as search:
    for line in search:
        for n in range(0, len(listecm)):
            if listecm[n][1:6] in line:
                print(listecm[n] + ' ' + line[7:13])
```