Biological Data Project
Characterization of the `Phosphomethylpyrimidine kinase` protein domain

Luca Dal Zotto, Francesco Ferretto, Giuliano Squarcina

February 15, 2021

Assignment

This project is about the characterization of a single protein domain. As starting point, each group is provided with a representative domain sequence and the corresponding Pfam identifier. The objective of the first part of the project is to build a sequence model starting from the assigned sequence. In the second part, the actual domain family characterization is performed analyzing different aspects: structure, taxonomy and functions. Along with a PDF report, each group is asked to submit domain models, code, commands and generated data as supplementary material.

---

1 Introduction

According to **Wikipedia**, a **protein domain** is a region of the protein's polypeptide chain that is self-stabilizing and that folds independently from the rest, forming a compact 3-dimensional structure. Therefore, the presence of a certain domain in a protein will determine its **structure** and, most importantly, its **functions**. Indeed, **molecular evolution** uses domains as building blocks studying how they can be recombined in different arrangements. These ideas motivate the characterization of domain families. In this project, we are asked to study the `Phosphomethylpyrimidine kinase` domain, starting from a representative sequence of the domain family. To be precise, we are provided with a UniProt accession, Pfam ID, Pfam name, a domain sequence and domain positions (in our case, the entire UniProt protein corresponds to the domain, so it was not necessary to cut it.

In the first part of the project, we will define a **domain model**. We have considered two different approaches: a **PSSM** model using **PSI-BLAST** and a **HMM** model. These models have been evaluated against Pfam annotations, by considering the ability of retrieving protein sequences having the domain, and also by measuring the capacity of identifying the domain's position in the sequence. After having tuned the parameters of the models, we found two satisfactory configurations, but, for the remaining part of the project, we selected only the model obtained with PSI-BLAST.

In the second part, we looked at the structural and functional aspects and properties of the entire protein family. The analysis has been split into three subsections. Firstly, regarding the **structure**, we performed an all-vs-all pairwise structural alignment, calculated a dendrogram, computed a multiple structural alignment, identified long range conserved contacts and identified the CATH superfamily and family matching our model. Secondly, we collected the **taxonomic** lineage and plotted the taxonomic tree of the family. Finally, we collected **GO annotations** relative to our proteins and performed an enrichment analysis, plotting enriched terms in a word cloud and reporting most significantly enriched branches.

The structure of this report follows the analysis just described: in the first part, we describe our models, while, in the second part, we report the structural characterization, the taxonomic analysis, the functional characterization and we also try to interpret the results. Along with this report, we also provide a Python notebook, with the code used to answer some questions, and a compressed archive where we stored domain models, additional code, commands and generated data. To facilitate the reproducibility of the analysis, we also shared a Drive folder, and we suggest to open the notebook using Google Colab and follow the instructions reported there.

Part I
Domain model definition

1 Building the models

In this first section of the project, the goal is to build a model representing the assigned domain. Its functionality should be two-folds: besides the ability to retrieve the proteins actually having the domain, it is asked to be able to predict the position of the domain itself in the obtained sequences. For this purpose, we considered two different approaches: **PSSM** and **HMM**.

As a preliminary step, we need to define the **ground truth** by finding all proteins in **SwissProt** annotated and not annotated with our domain, and its actual position in the sequences. For this reason, we searched all the manually curated proteins matching the **Pfam** id of our domain using **InterPro**, and downloaded the results in a `.json` format. Here we found the first useful information: our domain matches 150 proteins. Moreover, we also considered the entire SwissProt database, which have been used to compute the confusion matrix and some classification metrics.

The first phase for building both models was to retrieve a list of homologous sequences performing a **BLAST search**. At this point, we could choose which database making the search against to and also how many hits to get. Secondly, starting from these retrieved sequences, we generated a **multiple sequence alignment**. Since the number of retrieved sequences was quite high, we opted for **ClustalOmega** instead of **T-coffee**, which would have been much slower. Then, we visualized the MSA using **JalView**: in almost all cases, the alignment was satisfactory, in the sense that there were not evident misaligned sequences or other form of noise which could have reduced the performances of the models.

Then, we built the PSSM and the HMM model from the MSA and afterwards, we searched for significant hits using **HMM-SEARCH** and **PSI-BLAST**, against SwissProt. This procedure was carried out completely from the command line. Note that in doing so, we had to choose some parameters: in particular we focused on the number of iterations and on the E-value threshold for PSI-BLAST.

Finally, we evaluated both the ability of matching sequences and domain positions of our models. In particular, to evaluate the former, we made a comparison with the ground truth at the protein level, while, for the latter, we made an analysis at the residue level, comparing the positions occupied by the domain according to our model with the true ones.

2 Implementation and results

When performing the BLAST search, we can select different parameters. The first one is the target database. We considered different options: **UniProt**, **UniRef90**, **UniRef50**. In general, Uniref50 provided more false positives than UniRef90, while UniProt performed well with PSI-BLAST but provided more false negatives with HMM. For this reason, we considered only UniRef90 from this point on.

In our first attempts, we used the default values for the other parameters (E-value threshold = 10 and maximum number of hits = 250). With these configurations, the results were not satisfactory. We noticed that our initial models retrieved much more proteins (412 with HMM and 674 with PSI-BLAST) compared to their correct number (150). Examining more in details the results, we found that while the sensitivity was pretty high, the precision was really low. If we give a look at the confusion matrix, it is even more clear which is the problem: our models is very good in limiting the number of false negatives, however, the false positives are so many.

For this reason, we modified the BLAST search parameters reducing the maximum number of hits to 100 and then up to 50, and the E-value threshold to 0.001. In this way, the HMM model's performances increased quite a lot, but the PSI-BLAST model still was not great. Therefore, we changed the default parameters of PSI-BLAST trying less than 4 iterations (to avoid drift problems) and reducing the E-value threshold to 0.0001. The best results were achieved with only 2 iterations.

In the final notebook (`Biological Data project - group 8.ipynb`), we reported just the results obtained with the two best models. However, for completeness, we also saved the results of other trials in the folder "`PROJECT/Other notebooks`", where we also included a README file with a precise description of the files.

Let's now compare the two models. Regarding the ability of matching proteins, PSI-BLAST outperformed HMM, providing the same number of false positives (19) but only 1 false negative (against the 10 of HMM). Besides the confusion matrix (Table 1), a number of classification metrics has been computed (Table 2).

PSI-BLAST

	P	**N**
P*	149	19
N*	1	563,803

HMM

	P	**N**
P*	140	19
N*	10	563,803

Table 1: Confusion matrix of the two best models. (**P** = Actual Positive, **P*** = Positive Predicted)

Metric	**PSI-BLAST**	**HMM**
Weighted accuracy	0.997	0.967
Precision	0.887	0.881
Sensitivity	0.993	0.933
Specificity	1	1
MCC	0.939	0.907
F-score	0.937	0.906

Table 2: Protein matching evaluation.

On the other hand, for what concerns the domain position matching (Table 3), HMM is the winner. We tried to give a possible explanation of this results: maybe that PSI-BLAST is not as good as HMM because when performing new iterations, the model is considering new sequences in the alignment, and if they imply a modification in the successive alignment, this would provide slightly different starting and ending positions, and the results may be affected.

Metric	**PSI-BLAST**	**HMM**
Weighted accuracy	0.86	0.943
Precision	0.871	0.98
Sensitivity	0.981	0.919
Specificity	0.739	0.967
MCC	0.772	0.867
F-score	0.923	0.948

Table 3: Position matching evaluation.

To summarize this part, the PSI-BLAST model is better when it comes to identify the proteins having the domain, while HMM is more precise in finding the correct position of the domain in the sequence.

Part II
Domain family characterization

In the second section we focus on the characterization of the structural/functional properties of the assigned domain.

In order to achieve such goal, it is more important the capacity of the model to identify the proteins which actually contain the *target domain* rather than the capacity to identify the residues' position of the *target domain*.

Hence, referring to the PSI-BLAST's and HMM's performances presented in part I, the PSI-BLAST model should be preferred.

1 Structural characterization

As a starting point we've created two datasets, namely:

- `family_sequences`: all UniRef90 sequences matching our best model (PSI-BLAST);
- `family_structures`: all PDB chains whose sequences significantly match our model and with a minimum overlap of 80% with respect to the location of the domain.

In particular, in order to identify these PDBs, we first tried using the file `pdb_chain_uniprot.tsv.gz`, but since it contains lots of missing values, we decided to use the file `uniprot_segments_observed.tsv.gz`, as suggested by the professor. Essentially, it is a summary of the PDBs to UniProt residue level mapping with all observed data, showing the start and end residues' positions in the PDBs and also with respect to the UniProt sequence. Both these files can be downloaded from the Sifts website.

We parsed this file to extract all PDBs associated with our family of proteins and then, for each protein in our family, we computed the percentage of overlapping with the corresponding PDB, retaining only those PDBs with overlapping above 80%. The result was a list of 40 entries.

In order to evaluate the structural similarity of the PDB database, we performed an **all-vs-all pairwise structural alignment** with TM-align from the command line saving the results in a `.txt` file. Starting from these results, we built

a matrix representing the **pairwise TM-score**[1] and a matrix with the **pairwise RMSD**[2].

To perform a graphical inspection of the information included in such matrices we've generated two **dendrograms**, whose role was to identify clusters of structurally similar sequences according to the different output metrics of TM-align, emerging:

- from the dendrogram built using the RMSD (Figure 2), we remarked that the `2php` PDB constitutes by itself an isolated cluster, with an high different score wrt the rest of the retrieved PDBs;
- according to the TM-score metric (Figure 1), also an additional PDB, `3rm5`, has a structure different from the others.

As suggested in the project assignment, we decided to remove these two outlier PDBs according to the TM-score metric. The resulting dendrograms (Figure 3 and 4) were more satisfactory. In particular, the one based on the RMSD shows a main cluster made of 25 PDBs and a smaller cluster with the remaining 13 PDBs.

We identified conserved position performing **multiple structural alignment** with mTM-align on the web. Since it accepts max 30 PDBs, we selected only the larger cluster of 25 PDBs. Results (including the `.fasta` file) can be found in the folder `"PROJECT/structure/multiple structure alignment"`.

In order to identify long range **conserved contacts** we adopted a two-step approach:

1. identify, for each PDB, all pairs of residues *close* in the structure (<3.5 Å) and simultaneously *far* in the sequence (>12 positions);
2. identify conserved contacts among all PDBs.

The 2°step requires a particular care: after having found the position of long range contacts in each protein sequence, we need to find their corresponding position in the *multiple structural alignment*. In other words, in order to compute conserved contacts among multiple PDBs, we have to align the contact maps of each structure based on the *multiple structural alignment*. For this reason, we built a dictionary mapping the position of each residue from the *original* sequence to the corresponding position in the *multiple structural alignment*.

Done this, it was possible to count the number of long range contacts for each position in the multiple structural alignment, and plotting such information as a bar-plot (Figure 5). In addition to this, we have also computed the positions of long range conserved contacts for different conservation threshold (80%, 90%, 100%), which can be found in the notebook.

The **CATH** *family* and *Clan* were identify generating a `.fasta` file from the 25 PDBs sequences and providing it as input to the web service *HMMSCAN*. All PDBs belong to:

- Clan: CL0118
- Family: Phos_pyr_kin (Phosphomethylpyrimidine kinase)

2 Taxonomy

We collected the **taxonomic lineage** from the Uniprot web interface adding the relevant column *Taxonomic lineage (ALL)* to the output and downloaded the result. Then we parsed the file, filtering it to retain only proteins in `family_sequences` and removing not needed columns.

In order to plot the **taxonomic tree** with nodes' sizes proportional to their relative abundance we needed:

- the tree in a computer readable format (`.newick`);
- dictionary counting the frequency for each term in the *lineage*.

The tree in `.newick` format has been downloaded from *NCBI* website, providing as input the list of proteins in the *Taxonomic Browser* service. Then, the dictionary of frequencies has been used to assign a different weight to each node and finally plotted using the library `ete3.py` (Figure 6).

To facilitate reading, also a schematic tree with only nodes' names has been produced (Figure 7) using `Phylo` from `biopython`. Since the font size is still small, we suggest the reader to give a look at the notebook.

[1] Measure of the global fold similarity that ranges between 0 and 1.

[2] Root Mean Square Deviation.

3 Functional characterization

From UniProt, we selected as columns *Entry* and *GO IDs*, filtered only Swiss-Prot proteins and downloaded the output (`swissprot_goa.tab.gz`). We collected the **GO annotations** for each protein in `family_sequences` parsing this file and storing in an list all the GO-terms associated with every protein. According to Wikipedia, a phosphomethylpyrimidine kinase is an enzyme that catalyzes a particular chemical reaction. Indeed, analyzing the most abundant GO terms, we have found that all proteins are annotated with "catalytic activity" and "kinase activity". However, to make more accurate and meaningful observations, we need to compute those terms that differentiate our set of proteins from the others (enriched terms).

In order to calculate the **enrichment**, we created the dictionary `go_map` storing for each protein in Swiss-Prot the set of associated GO-terms including the *ancestors*. Then we created two frequency dictionaries:

- `terms_set` storing the frequency of each GO-term in `family_sequences`;
- `terms_rest` storing the frequency of each GO-term in the *rest* of proteins.

These dictionaries have been used to compute the **fold increase** and the **Fisher's exact test**. Finally, the statistics computed so far for each GO-term have been stored in the dataset `go_dataset`.

This dataset has been used to compute the dictionary `dict_freq`, having key-value pairs of the form:

$$\text{GO-term} : \log\left(\frac{1}{\text{right p-value}}\right).$$

This information is needed for the **word cloud** representation. The ratio ensure an higher value associated to a lower *right p-value* and the log slows down the decline of the ratio as *right p-value* increases. Hence, words (annotations) with a lower p-value are displayed with an higher font size. Then, we found enriched **branches** (high level terms, so *no* leaves) for each *sub-ontology*, by filtering `go_dataset` in following order:

1. sub-ontology
2. parent nodes

and finally sorting within each sub-ontology the GO-terms by *right p-value* in ascending order.

4 Final considerations

To conclude, we tried to give an **interpretation of the results**, also from a biological point of view. To do so, we compared some information found in internet (especially Wikipedia) with our results.

Firstly, the phosphomethylpyrimidine kinase belongs to the family of transferases, to be specific, those transferring phosphorus-containing groups (phosphotransferases) with a phosphate group as acceptor. Indeed, among the most enriched terms (according to Fisher's exact test) belonging to the molecular function sub-ontology, we found:

- kinase activity;
- pyridoxal kinase activity;
- phosphotransferase activity, alcohol group as acceptor;
- transferase activity, transferring phosphorus-containing groups;
- phosphomethylpyrimidine kinase activity.

Moreover, phosphotransferase activity and kinase activity are also the two most significantly enriched branches, confirming the goodness of the results.

Furthermore, this enzyme participates in thiamine metabolism. In fact, among the most enriched terms (with respect to the fold increase) of the biological process sub-ontology, we found:

- thiamine catabolic process,
- thiamine-containing compound catabolic process,

besides thiaminase activity as molecular function. Moreover, we found thiamine biosynthetic process as one of the most enriched branches of the biological process sub-ontology. Regarding the cellular component namespace, the most enriched term was the cytosol.

To conclude, let's summarize the main results of this project: in the first part, we built a sequence model able to represent the protein family. Then, we characterized our family analyzing their structures, the taxonomic lineage and, most importantly, their functions. Finally, we tried to give an interpretation of our results, comparing them with information available online.

5 Appendix - Figures and Tables

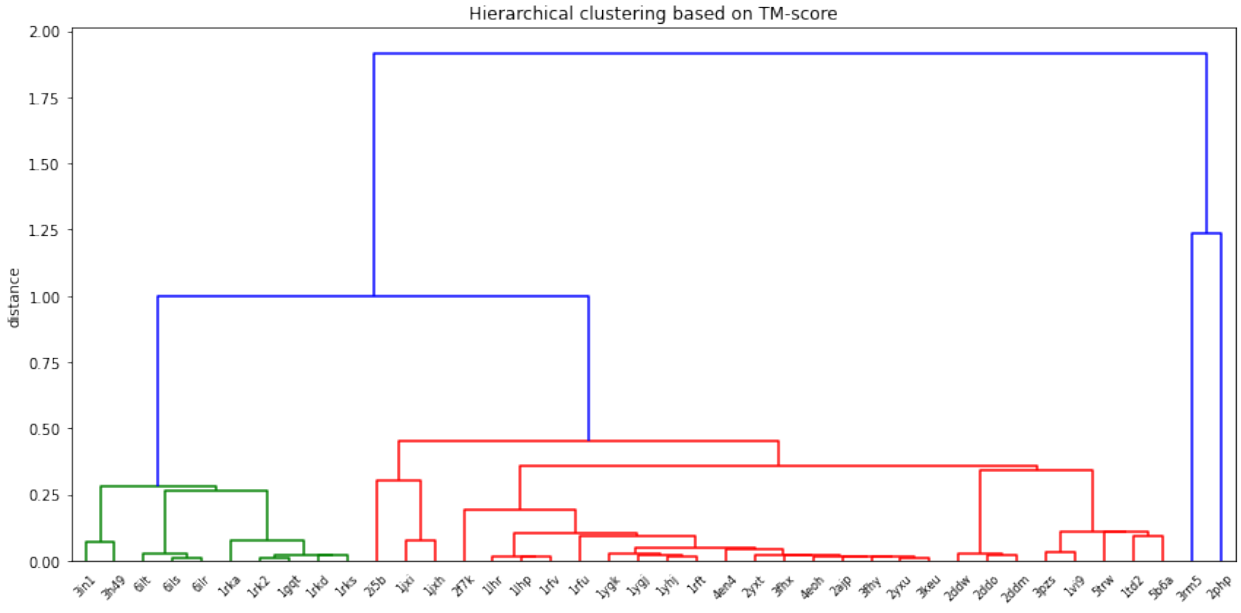

Figure 1: Dendrogram based on the TM-score with outliers

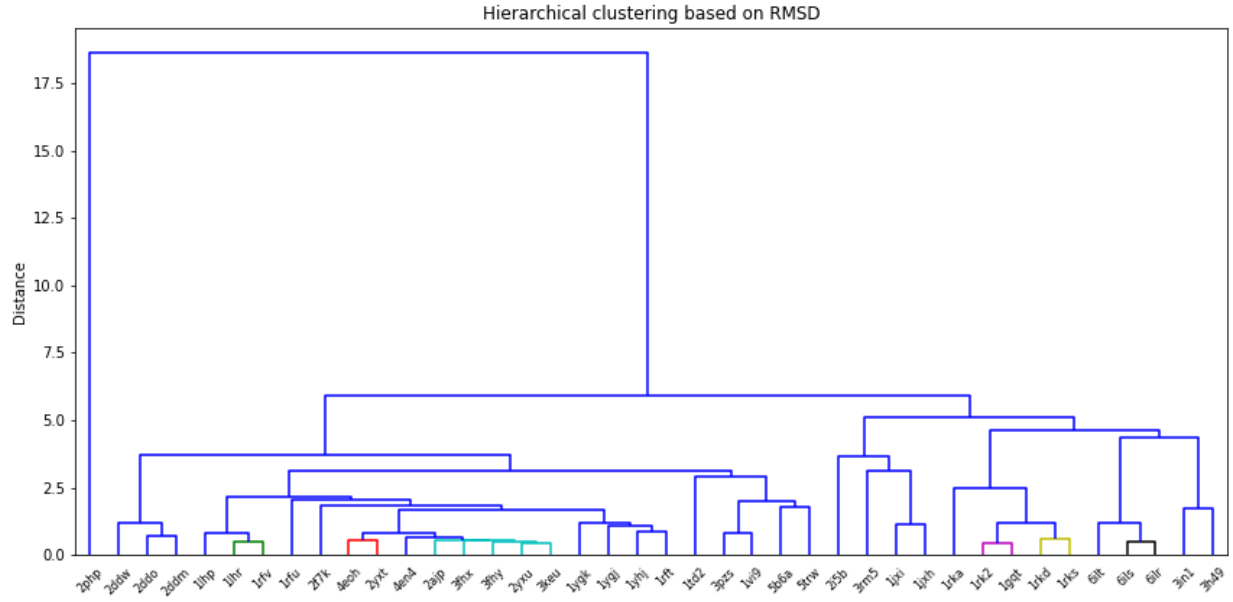

Figure 2: Dendrogram based on the RMSD with outliers

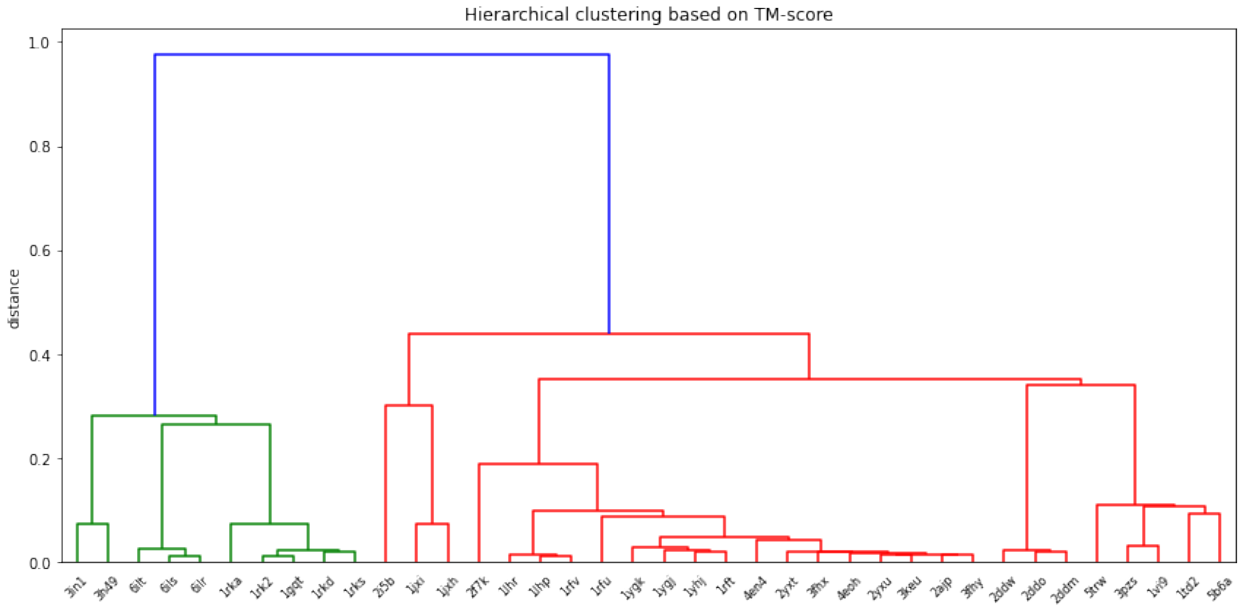

Figure 3: Dendrogram based on the TM-score without outliers

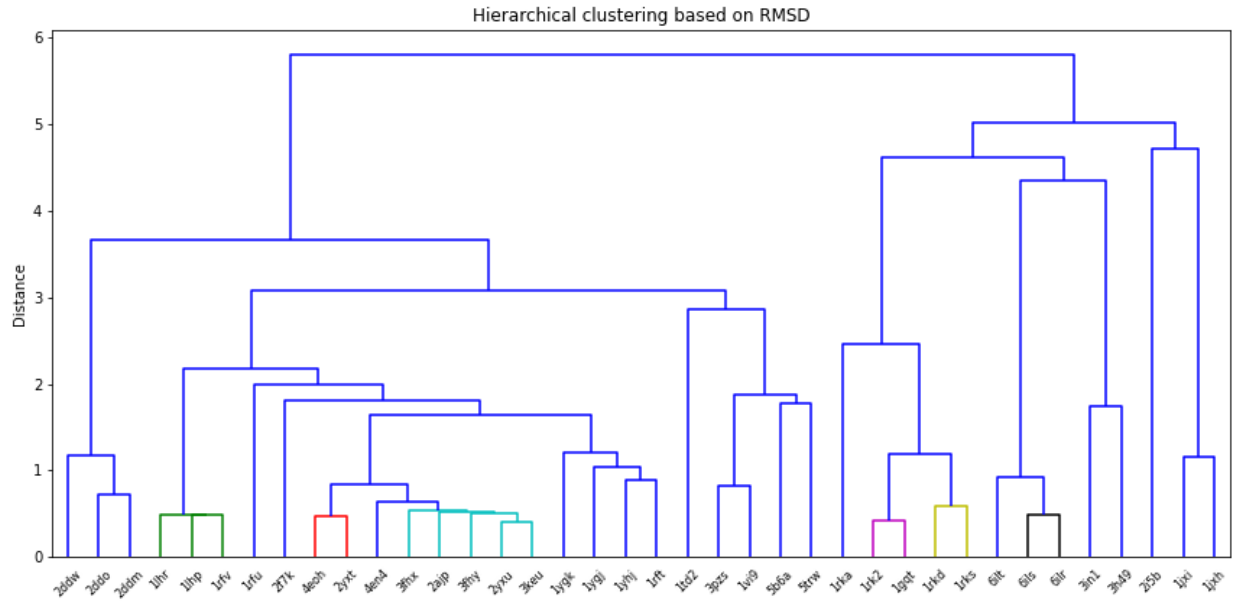

Figure 4: Dendrogram based on the RMSD without outliers

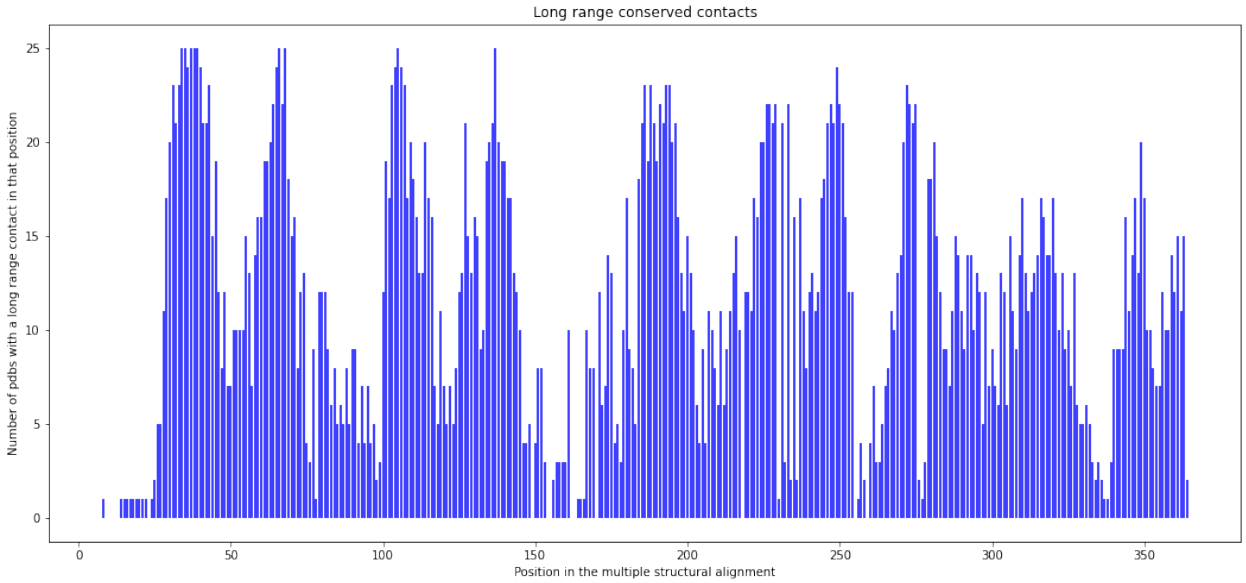

Figure 5: Long range conserved contacts

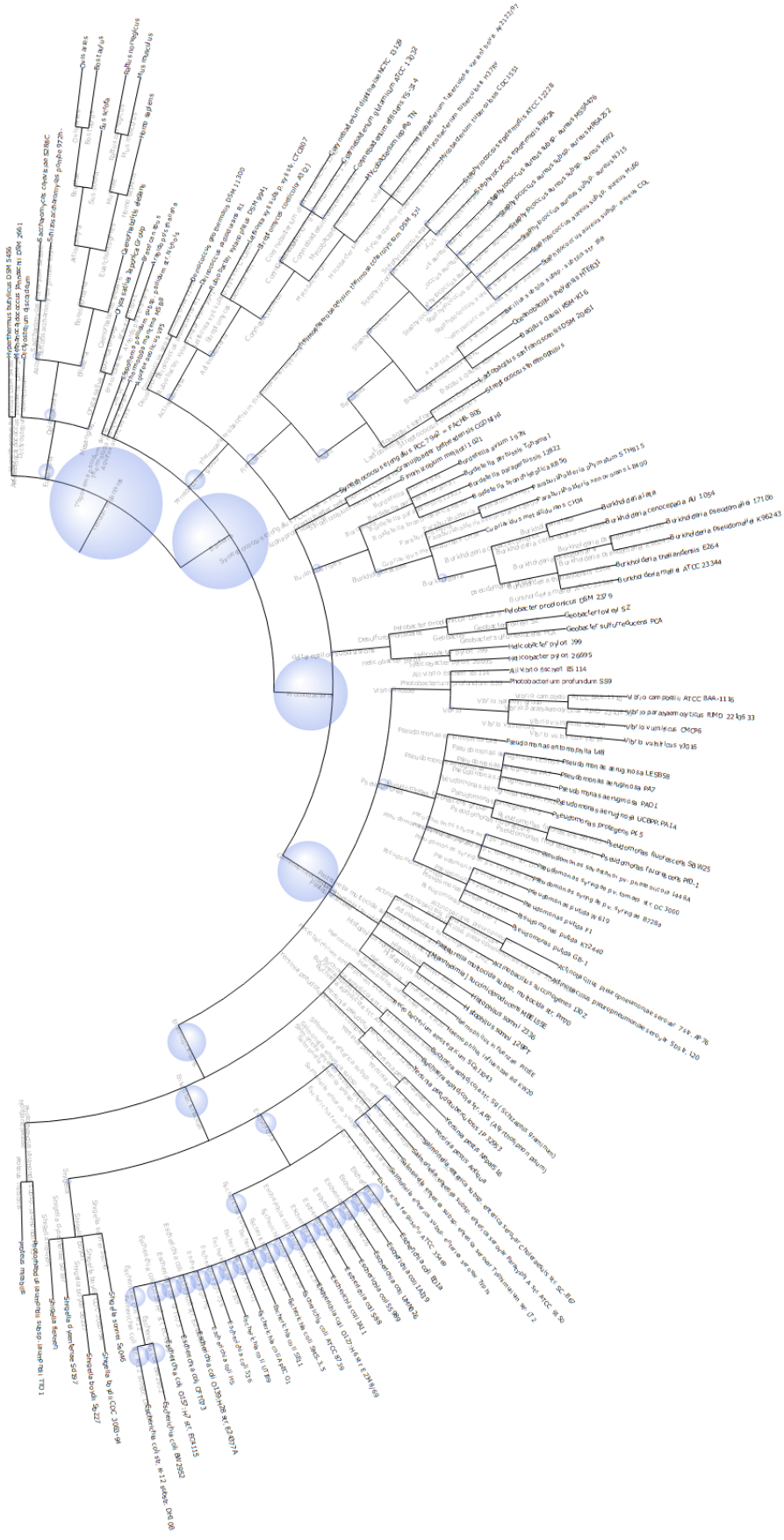

Figure 6: Taxonomic Tree with nodes's sizes proportional to abundance

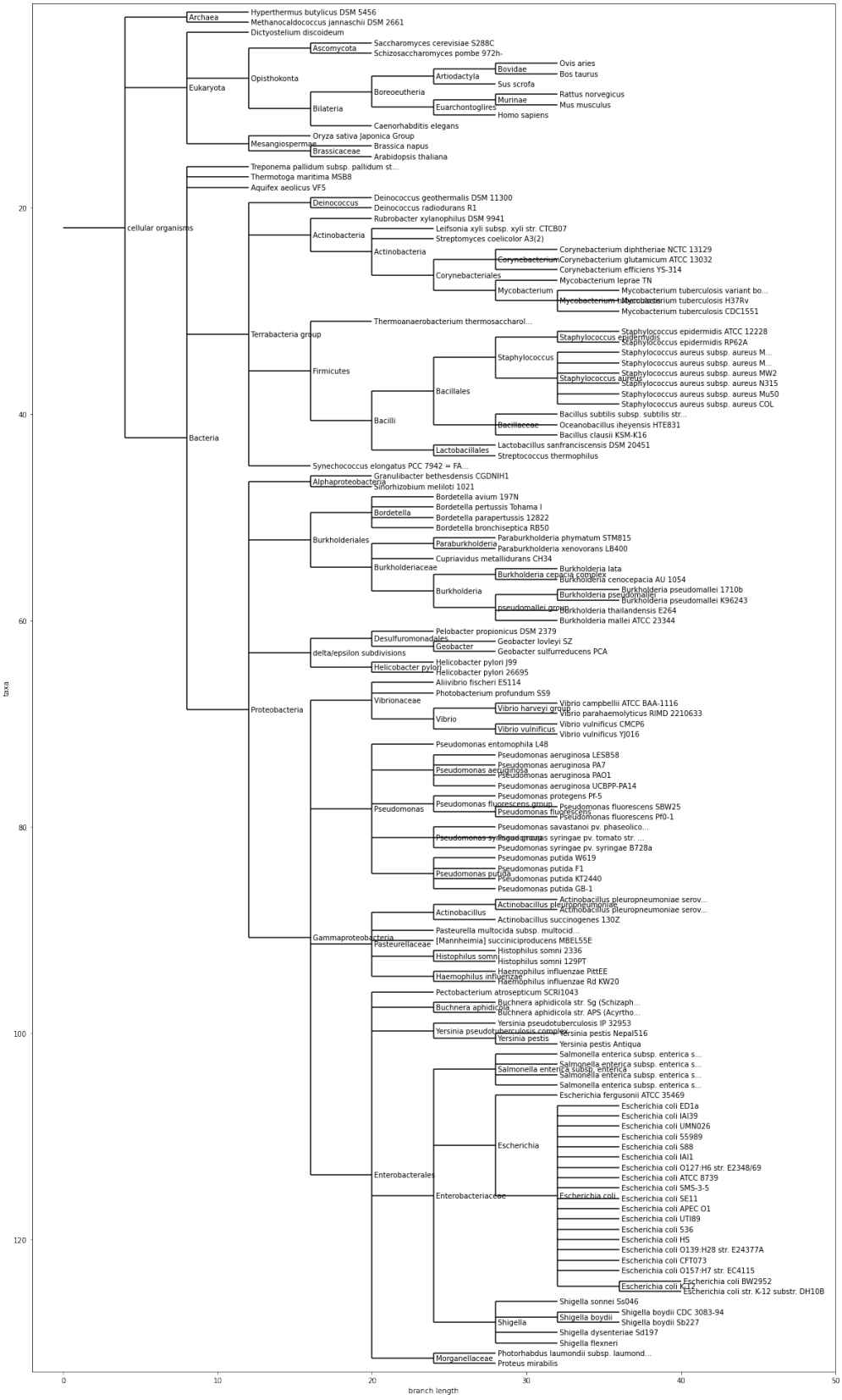

Figure 7: Taxonomic Tree - schematic view

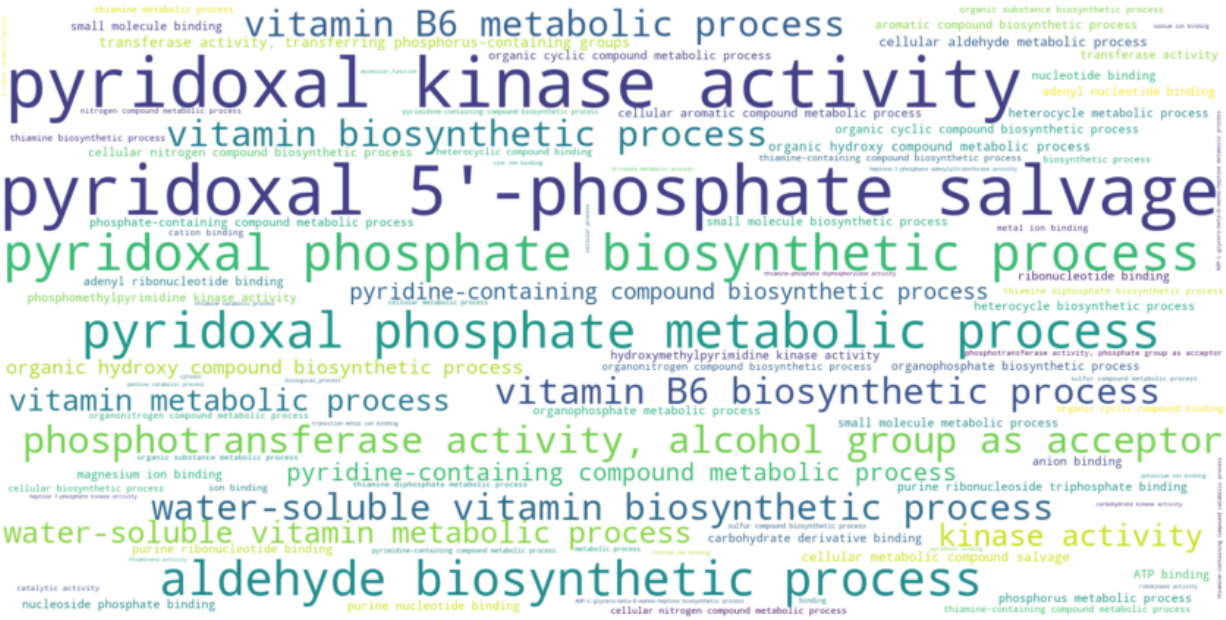

Figure 8: Word Cloud