Liam Le Goffic Alexis Trang Yann Zhong

GENOM Project report

Evolution of structural signatures in families of soluble protein domains

Introduction

The aim of this project was to work on **Hydrophobic Clusters (HCs)** present in soluble domains of proteins, with the goal of building a suitable **substitution matrix** from information that is derived from HC analysis. HCs can be thought of as structural signals that are defined from sequence information and that match Regular Secondary Structures (RSS).

Hydrophobic Cluster Analysis (HCA) is a technique developed in the late 80s and is able to provide us with protein RSS information from a single amino acid sequence, thus bypassing the need for homologous sequences. This is important because a large number of proteins present in databases such as **Swissprot** exhibit dark or gray regions – biological unknowns which can impede genome annotation and deciphering based on reliable homology-based methods. Additionally, as **structural folds (and structures in general) are better conserved than sequences** [1], the use of HCA is very desirable in this context.

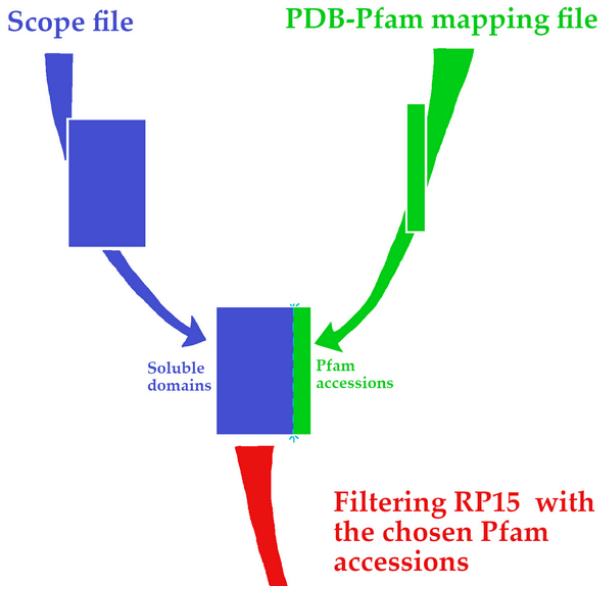
The creation of a substitution matrix for different HCs (there being hundreds upon hundreds of them, depending on how they are defined) is thus a novel approach that would allow a BLOSUM62 type matrix for HCs rather than simply for amino acids, and holds much promise. With such substitution type matrices, scoring would become more accurate and convenient for dynamic programming methods akin to Smith-Waterman as well as heuristics-based methods such as BLAST.

In this short report, we will seek to outline our methods and approach to this problem under the guidance of Elodie Duprat, our supervisor.

Methods

Data acquisition and pre-processing

The data for the project was acquired from three separate links and required a pre-processing function in order to isolate relevant sequences. The three files that were respectively required for this part were obtained at the SCOPe website to do soluble domain extraction [2]and the FTP website of the EBI for Pfam alignments and the mapping file required to extract Pfam IDs[3].



First, we needed to make sure that we were only working on **soluble domains** within the SCOPe classification, a database regrouping all known structures of protein domains.[4] In order to do so, we only considered domains that belonged to classes a (all alpha proteins), b (all beta proteins), c (alpha/beta), and d (alpha+beta). We then created a function that filtered all the domains on the first file based on the identifier on the third column, keeping only those whose columns started with the letters a, b, c or d.

Then, the second tabulated file contained relevant information and notably **Pfam accession numbers** for PDB chains present within UniProt. From our filtered output (for soluble domains) of the first file, we extracted a second filtered list of Pfam accession numbers based on the SCOPe access numbers (1st column) of the soluble domains. That way, we had the Pfam accession numbers of soluble domains that also had known three dimensional sequences, which would enable us to look for relevant alignment identifiers: Pfam identifies sequence families (based on conserved regions) and provides multiple sequence alignments for reference proteins.

Then, the third file that was worked on was *Pfam-A.rp15*. This file is a **Representative Proteome** file containing a large number of sequence alignments per Pfam accession file that fit the threshold value of 15% for CMT (Co-Membership Threshold), which is a metric used in order to reduce redundancy in Pfam alignments. We consequently used the extracted Pfam accession numbers from the second step in order to only keep the alignments in RP15 that both fit the criteria of: 1) Soluble domain, and 2) Known 3D structure.

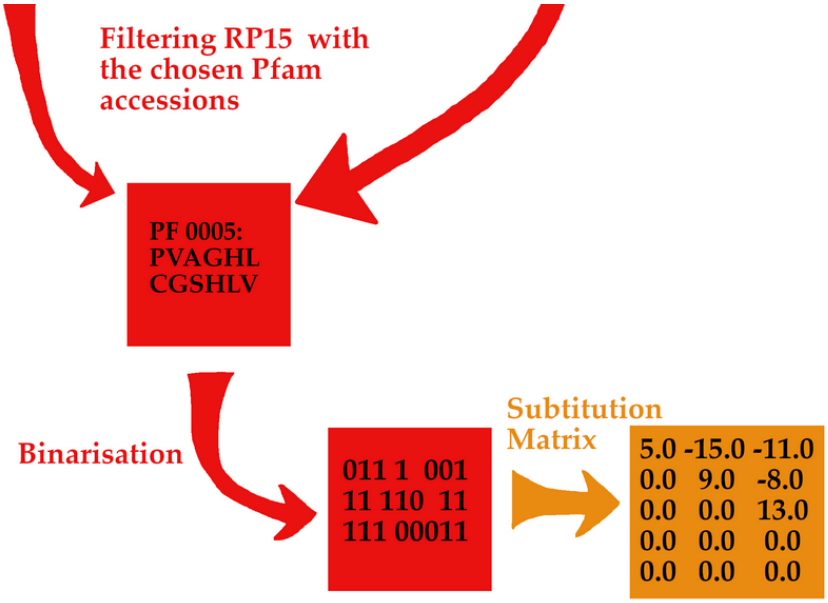
HC – rules of definition, binarization

HCs are strictly defined as a grouping of several strong hydrophobic amino acids, that is to say **V, I, L, F, M, Y or W,** that start with a strong aa and end with a strong aa. In the context of HCA and matrix building, any amino acid that wasn't part of the former group became a **binary 0**, while any that was became a **binary 1.**

A HC also wasn't necessarily only strong amino acids: amino acids represented in binary as 0 may also be present within a HC, with the condition that they do not exceed 4 in length if following each other, as defined by Lamiable et al.[5]

For example: 11, 1101, 10001, 101010101 are all considered acceptable HCs, while 10, 01, 100000001 do not fit the definition of an acceptable HC, so were not identified as such.

Thus, we ran an HC identification code on the filtered RP15 file in order to extract a list of unique HCs by both their raw sequence as well as their alternative representations in P-code and Q-code (though the latter is not used). We identified a large number of unique HCs, some even going as long as having a P-code of 435641765927107293001, but decided to filter the relevant HCs to the **500 most common ones**, mirroring closely the reference csv file we were provided, which also included a list of 400+ most common HCs built from the HC Database[6].



Matrix creation

With a filtered list of the most common HCs in hand, we then proceeded to creating the substitution matrix. In order to do so, we had to consider every single Pfam alignment that remained after having applied all three of our pre-processing steps. Thus, every alignment was able to provide a matrix of counts for each pairing of HCs in our 500 most common list, after which a **"supermatrix"** of counts could be built to create our substitution matrix as the BLOSUM62 matrix [7].

To go into a bit more detail, we first needed a function to binarize our alignments slightly differently than before. Previously, we merely converted strong amino acids to 1 and kept all the other amino acids at 0, while checking for HCs using the method described in the previous section.

The second binarization required was this time to binarize in terms of belonging to an HC: essentially, for a PFAM alignment (of for example 1000 sequences), we needed to have **any amino acid belonging to an HC become a binary 1** and any that did not become a binary 0.

In order to build our substitution matrix, then, we needed to consider (with a pseudo count) the sum of every column (so, every position in a Pfam alignment). Then, if said sum was greater than a minimal acceptance threshold (e.g., **80%**, in our case, where 100% would come from a column full of 1s), we would consider this particular position as strongly conserved in HCs. If a particular position is strongly conserved in HCs, we then built a matrix according to the overlap of HCs in those positions. For example, if at position 25, there was an overlap of HC of P codes 3, 11, and 13, then we would increment 1 on the rows and columns corresponding to those HCs.

This was done for every Pfam alignment that was present in the now filtered RP15 file, which led to the creation of the previously mentioned "supermatrix", which is simply the sum of the 500 by 500 matrices for every Pfam alignment. Finally, from this "supermatrix", we applied **a log odds ratio** as for BLOSUM62 type calculations, following the formula:

$$score(HC1, HC2) = \frac{1}{\lambda} log \frac{p_{HC1\_2}}{f_{HC1} * f_{HC2}}$$

Where p_{HC1_2} corresponds to the frequency of having HC1 and HC2 overlap at least over one amino acid over a strongly conserved HC region, and $f_{HC1} * f_{HC2}$ to the product of their individual frequencies. λ is a scaling factor so as to ensure that our matrix is populated with integers rather than floating point numbers.

Results

The final substitution matrix, as well as the full code include source Python files and tutorial notebooks, is available on our GitHub repository here [8]. Due to lack of time, we have not had the time to evaluate its performance, but we are reasonably confident in our methodology for its construction. Overall, for RP15, the run time is under three hours for pre-processing, creating a dictionary of all unique HCs, and matrix construction (close to 9000 seconds) if we consider that the filtered RP15 file is already at hand. A total of 3287 Pfam alignments were considered, and the top 501 HCs were used in the construction of the 501 by 501 substitution matrix.

In the future, additional work may be done on the other Representative Proteome files such as RP35 or RP55, and the use of the substitution matrix should be tested so that it may see similar use as the popular BLOSUM matrix.

References

[1] A. Goyal, S. Sokalingam, K.-S. Hwang, and S.-G. Lee, 'Identification of an Ideal-like Fingerprint for a Protein Fold using Overlapped Conserved Residues based Approach', *Sci Rep*, vol. 4, no. 1, p. 5643, May 2015, doi: 10.1038/srep05643.

[2] 'EBI FTP for Pfam database'. https://ftp.ebi.ac.uk/pub/databases/Pfam/mappings/pdb_pfam_mapping.txt (accessed Jan. 20, 2022).

[3] 'Index of /pub/databases/Pfam/current_release/'. https://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/ (accessed Jan. 20, 2022).

[4] J.-M. Chandonia, N. K. Fox, and S. E. Brenner, 'SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database', *Nucleic Acids Research*, vol. 47, no. D1, pp. D475–D481, Jan. 2019, doi: 10.1093/nar/gky1134.

[5] A. Lamiable *et al.*, 'A topology-based investigation of protein interaction sites using Hydrophobic Cluster Analysis', *Biochimie*, vol. 167, pp. 68–80, Dec. 2019, doi: 10.1016/j.biochi.2019.09.009.

[6] P. J. Silva, 'Assessing the reliability of sequence similarities detected through hydrophobic cluster analysis', *Proteins*, vol. 70, no. 4, pp. 1588–1594, Oct. 2007, doi: 10.1002/prot.21803.

[7] S. R. Eddy, 'Where did the BLOSUM62 alignment score matrix come from?', *Nat Biotechnol*, vol. 22, no. 8, pp. 1035–1036, Aug. 2004, doi: 10.1038/nbt0804-1035.

[8] 'GENOM_BIM/substitution_matrix.txt at main · yann-zhong/GENOM_BIM'. https://github.com/yann-zhong/GENOM_BIM/blob/main/matrices/RP15/substitution_matrix.txt (accessed Jan. 20, 2022).