**Minutes**

17/02/23

* Discussed HHblits and PSSMs:
  + Generates multiple sequence alignments (MSAs)
  + Discussed these MSAs:
    - A protein can evolve over time to have subtle changes in the amino acid sequence.
    - The protein with sequence MATYL... could change to
      * MATWL... for chimps. Tyrosine (Y) has changed to Tryptophan (W).
      * ML-WL... for humans. For humans, further change to Leucine (L) from alanine (A) and the deletion of Threonine (T).
      * These sequences are similar to our target sequence inputted to HHblits, therefore the algorithm returns these and other sequences.
  + With these MSAs we can calculate the probability that a specific amino acid exists in this position and encode this in a size 20 vector using proportions.
    - For example, the sequence:
      * MATYL... generates:
      * MATWL...
      * MLTWL... and
      * ML-WL...
      * We have a vector where position M is given 1.0 for our first entry, a vector where A has 0.5 and L has 0.5 for our second entry, and our third entry has T with 1.0, etc.
      * A distribution of the proportion of observed amino acids in position 1 of the sequences.
  + The proportion gives us some value between 0.0 and 1.0. When we put this through a logarithmic function, we get some value between –inf and 0. From inspection of the .hmm file generated by hhblits, they have replaced –inf with a ‘\*’. I was setting this to 0, but I should set it to the largest possible (negative) value (our max value). Finally, they multiply through by (-1), to get values between 0 and \*. More information on this method used by hhblits discussed in: <http://www.cromulentrambling.com/2014/10/getting-hhblits-to-produce-profiles.html>
    - Therefore, the lower values are the more commonly appearing amino acids amongst the sequences. We want our model to value these more.
  + Discussion about why learning models would not favour these larger values:
    - What a model is learning can be on either side of the range. This is why [-1, 1] is an ideal input. It is symmetric and can be flipped easily.
  + Furthermore, PyTorch’s models should work better with these log values.
    - They are simpler to interpret differences between small but differing values.
      * For example, there is very little measurable difference between 0.001 and 0.0001, whereas –log2(0.01)~6.64 and –log2(0.0001)~9.97.
    - If we are working with our proportions, values below 0.1 are usually ignored and can be treated like 0s.
      * With these larger differences, small values can be better represented to be learnt.
    - Large log values will make it clear that these amino acids are unwanted in this position, also helping the model learn.
    - Using logs allow more computational stability as we can sum these values instead of multiplying the tiny proportions as before.
  + The discussion above will create our final input to our neural network.
  + Dynamic programming is used to create multiple sequence alignment.
* Discussed Django server requiring PSSM input instead of just raw sequence:
  + If my model requires PSSM input, then any input sequence (text or FASTA) must be translated to PSSM.
  + Could have a database store:
    - Where sequence ids match to hmm files / the stored pssm.
    - The sequence id would be >sp | accession no. | …
    - This prevents needing to conduct a MSA on every entered sequence. Creates a cache and reduces computing resources as this MSA is quite intensive to be carrying out for every input.
  + Problem is for new entries entered by users (sequences out with the ones I generated) could be misformatted, for example the sequence id could often have the value ‘test’, with a sequence that already exists in the database.
    - Could consider string matching sequences if this is feasible.
* Discussed dataset:
  + There is the possibility of seeping sequences between datasets.
    - Running MSA can return test sequences and involve them within the training data.
      * Problem – this is not an entirely independent test set now.
  + A solution is to ensure test sequences are entirely non-homologous so that they cannot be generated via MSA.
    - This would require a lot of computation to group and separate the homologous sequences. We could generate a new database to separate these homologous sequences.
  + Seeping sequences can be problematic in results and often a reason reviewers flag up published papers.
    - As both my models will have this bias I can still compare my deep learning approaches.
    - Careful consideration would be required when comparing to other papers results.
  + As CASP datasets use newly found proteins, these older CASP datasets may also have seeping sequences, as this protein data may have been entered in Disprot.

Goals for this week:

* Work on dissertation. Finish literature review in background, write up analysis fully and start working on other design and implementation sections.
* Generate validation and test fasta files to create PSSMs.
* Consider feasibility of database with PSSMs. Look at Django models, and what look up method is best.