Questions from 26.June 2018

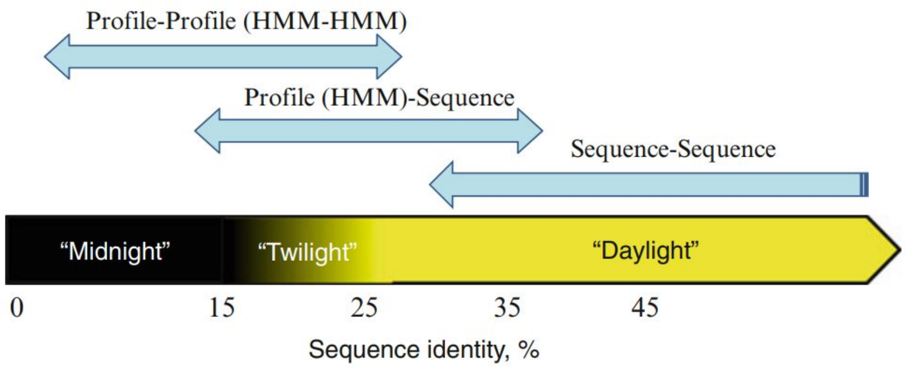
Thanks to All .. Cheers See you on Protein Prediction 2 xd

Thanks guys! @2019 if you read this:
We had a total of 100 points

- **16 points SingleChoice questions: 8 questions of 2 points each, exactly 1 correct answer in 3 choices**
- **64 points from LITERALLY THESE QUESTIONS BELOW 1:1 EVERY WORD**
- **20 points Exercise: Find all ORFs in a given DNA strand (7p), Do Needleman-Wunsch (7p), explain some Neural network stuff (for example backpropagation) (6p)**

Q1: Explain the concepts of sequence comparison zones.

A: This concept indicates how likely we can predict structure from sequence identity, there are three zones:



- **Daylight Zone:** If the sequence identity is around 40 to 100% we can say the structure will be similar. Sequence to Sequence alignment can be used in this zone and proteins are normally homologous.
- **Twilight Zone:** It is not completely clear that we can predict structure from sequence identity, therefore sequence - profile methods such as PSI-Blast, Clustal W/X, MaxHom, SAM/HMMer, T-coffee are required to predict structure from sequence identity. The twilight zone resides in the 20-35% pairwise sequence identity where the signal gets blurred because false positives explode while true positives are slightly increasing.
- **Midnight Zone:** Protein structure cannot be inferred from sequence similarity, profile to profile comparisons [HHblits] are required to get deeper in this zone.

Key words: Sequence Identity, Sequence - Sequence, Sequence - Profile, Profile - Profile, Structure similarity, HMM (Hidden Markov Model)

Oli’s answer (mostly stolen from above):

- **Daylight Zone:** If the sequence identity is around 40%-100%, we can often directly infer structure similarity. Sequence-to-sequence alignment can be used in this zone.
- **Twilight Zone:** It is not clear whether we can predict structure from sequence identity (sometimes we can, sometimes we can’t), therefore sequence-profile methods such as PSI-Blast are required to predict structure. The twilight zone applies to the range of 20%-35% pairwise sequence identity. Plenty of false positives in this zone (i.e. cases where we think we can predict structure similarity but can’t).
- **Midnight Zone:** Structure can’t (or very rarely) be inferred from sequence identity. Profile-to-profile comparisons are required for finding potential structure similarity (coming from “distant relations”). Sequence identity in this zone is 0%-20%.

Q2: How do profiles help in the prediction of protein structure?

Answer: Profiles help in the prediction of protein structure by providing input features to many prediction methods. The input features are the [evolutionary] information condensed by the profile. In addition to that, profiles can also provide information about conserved regions or conserved residues in specific positions. In doing that we can see which segments are the most important for structure/function preservation in the family.

Def:

- “The profile is generated by the calculation of the frequencies of each of the amino acids in all the positions of the alignment. ”
- “A profile shows all the information contained in a multiple sequence alignment.”
- “Used in PSI-BLAST and PSI-search !!! “

– profile = multiple alignment + observed character frequencies at each position

A: -> Contain more info, because they are derived from families
-> families define info that is relevant
-> **Evolutionary information**:
-> Proteins evolve (under constraints =) to maintain a 3D structure, that’s why it is informative
-> machine learning devices help to pick up this evolutionary info

(not directly related to Q2):
PSSM (position specific substitution matrix)
(helps improve alignments(local and global), condense info about the evolution of a protein, helps to find protein homolog in database)
=
profile (tells you at what position you can replace which residue by which residue)

BLOSUM matrix

- accounts for biophysical features (like hydrophobic, positive charged, size) but not position-specific

Oli’s answer:

- Profiles help in predicting secondary structure because they **condense information about the evolution of a protein**
- They also help in **improving alignments** because they take into account **position-specific scoring**, i.e. they tell us about the likelihood of replacing one residue with another at each particular position.
- In general, profiles profit from the **relations between protein families.** This is useful because proteins of the same family often form similar structures.

Q3: What is the difference between BLAST and PSI-BLAST?

A:

Basic **L**ocal **A**lignment **S**earch **T**ool is an algorithm for comparing primary biological sequence information, basically looking for small segments named “words” in both sequences. It‘s faster than dynamic programming methods (Needleman-Wunsch algorithm and Smith-Waterman algorithm), however BLAST cannot guarantee an optimal solution. BLAST uses a position independent system, is less accurate, less sensitive, but faster than PSI-BLAST.

PSI (**P**osition-**S**pecific **I**terative) BLAST is a BLAST with many iterations: the first one is like a normal BLAST and then, after first alignment, the substitution matrix is adjusted to find conservation patterns and run new matches.

PSIBLAST uses position-specific scoring matrices (PSSMs) to score matches between query and database sequences, in contrast to BLAST which uses pre-defined scoring matrices such as BLOSUM62. It is more accurate and provides different scoring for same letters depending on their position.

Alternative:

(from exercise 5:)

Basic Local Alignment Search Tool (BLAST)

– Searches databases for similar protein/nucleotide sequences

– Scores hits based on local alignments and score matrices
(default: BLOSUM62 for proteins)

– High speed due to using seeds for hit determination

s

PSI-BLAST:

• Iterative BLAST

1. Use BLOSUM62 scores for first search against database
2. Build PSSM based on high-scoring hits -> profile
3. Search again using the PSSM (profile)
4. Repeat steps 2 & 3 for a specified number of times

• Can find more distantly related protein sequences

But: false hits can „pollute“ the PSSM

Oli's answer:

- BLAST is an algorithm for comparing amino acid sequences. It finds similar sequences by locating short matches, so-called 'seeds', between 2 (or more, could be an entire database) sequences and further performing local alignments.
- PSI-BLAST uses the same matching principles as BLAST but it performs the matching in several iterations. After the first iteration, a profile is built up and in the subsequent iterations, this profile is updated and used to compare against.

Q4: What is the difference between Hashing and Dynamic Programming?

A: -> database today has 120 million records
-> Hashing doesn't align entire matches but words (of size at least 3, but today it is recommended to use 5)
-> Hashing much faster
-> Hashing more feasible, doable
- Dynamic Programming is great for Global Alignments (see Needleman-Wunsch), Hashing works well for aligning fragments with sequences (i.e. local alignment)

Oli's answer:

- Dynamic Programming solves a problem by consecutively solving sub-problems. Each further sub-problem relies on the solution of preceding sub-problems. This often results in building up a solution matrix such as the one in Needleman-Wunsch. In our context, Dynamic Programming is mostly used for Global Alignment of sequences. Drawbacks are computational expensiveness and massive memory use.
- Hashing relies on smart guessing. This is useful because those guesses often bring us a large step closer to the solution. E.g. with BLAST, we first match a seed (3-letter word) and then "hope" that the sequence similarity continues from there. Hashing is very efficient, fast and computationally cheap.

Q4.5: When to use global and when local alignment?

Answer: When you already know the sequences are closely related and of nearly same lengths, use global alignment, if not use local alignment.

A: global: align two proteins from one end to the 2other, first to last residue
search sequence is a whole protein/
domain = unit that folds on its own, unique protein structure
local: find a match

- Global: all residues aligned: you are forced to compare two proteins from end to end.
- Local: best matches: you just cut out something that matches locally.

We can't use global alignment to detect local regions of high similarity
We can't use global alignment to align a fragment and a sequence
Use Global for: Search for mutations or polymorphisms in a sequence compared to a reference.

Oli’s answer:

- Use local when you align a fragment against a complete sequence
- Use global when your sequences are of roughly the same length and you want to align them end-to-end

Q5: Why is it impractical to dynamic programming align multiple proteins? What is an alternative?

Answer: Dynamic programming for multiple alignments is extremely expensive in terms of CPU time and RAM. Alternatives to using dynamic programming is hashing (BLAST and PSI-BLAST), genetic algorithms (T-Coffee) and HMM-s.

A: - Clustal Omega (ClustalW/ClustalX) does dynamic programming on multiple proteins
- Problem with dyn programming is CPU time
- can’t align >1000 proteins

Hidden Markov models (like HMMER) does alignments faster than Clustal Omega

Oli’s answer:

- Dynamic programming is computationally expensive. Use hashing-based algorithms (e.g. BLAST) instead, or something even fancier such as HMMs.

Q6: How does comparative modeling help us in predicting protein structure?

A: Numbers!

-The term **comparative modeling** refers to modeling a protein 3D structure using a known experimentally determined structure of a homologous protein as a template.

Comparative modeling can model 60 Million (half) protein sequences

- How does it work:
    - Align Query Q against database (PDB) of proteins of which we experimentally know 3D structure
    - Alignment: Sequence-based / String comparison
    - Find Template T similar to Q
    - For T we know the structure because it is experimentally known
    - Comparative modelling: Q and T have similar sequence , the region where they aligned = they have the same 3D structure.

Steps:

1. Do a query to the database in order to find a template protein similar to the query protein.
2. Align the query sequence with your identified template
3. Build a model
4. Assess (e.g. are there any stretched structures? Any clashes?)
5. Redefine.

- (If we are in the twilight zone, we can't find a similar protein.)
- (the twilight zone has no fixed starting point or end)

Oli's answer:

- Evolutionary related proteins (so-called 'homologous' proteins) are likely to have a similar structure **even if** their sequence similarity is minimal
- Therefore Comparative Modelling helps us in **"guessing" the right structure**
- The fact that we already know of proteins related in structure (thank you evolution!) gives us a great starting point and **saves us searching time**. This is the benefit that Comparative Modelling gives us.
- Steps in Comparative Modelling:
  1) Identify template through database search
  2) Align target with template
  3) Build model
  4) Assess model
  5) Refine

Q7: Protein adopt unique 3D structure to function. Explain how some proteins often referred to as proteins with disordered regions do not.

A: Disordered region: Protein is not adopting a unique 3D structure

- Being disordered could be seen as a feature as it allows more flexibility
- Saying: Same sequence adopts same 3D structure, info how it will look is in the sequence
- However here the information is disordered
- some regions have to bind to partners
- disordered region adopts to different regions/partners, dynamic, flexible

I

A: Disordered proteins or proteins with disordered regions have no fixed function or less functionality. Disordered regions can be associated with certain disease, because the protein does not work anymore as intended. Disordered proteins can be positive as well, the disordered regions can bind with other proteins and can therefore act as a bridge between domains. That means that disorder can make a protein more flexible.

Oli's answer:

- Disordered regions are those that don't fall into the regular categories of secondary structure (helix, strand, loop, …).
- However, there's a purpose for disordered regions: often, these are the parts of the protein that help in binding with other proteins (see 'key-lock principle'). If the structure were too regular, it would be hard to "team up" with a partner protein. With disordered regions, the protein can **induce a fit**.

Q8: How do profiles help in the prediction of aspects of protein secondary structure? How can you encode the profile?

A: When aligning protein sequences it is often apparent that certain regions or specific amino acids, are more conserved than others. Such conserved regions are often conserved because they encode a part of the protein that is functionally important. The term *motif* is used to refer to a part of a protein sequence that is associated with a particular biological function.
For example a region of a protein that binds ATP is called an ATP binding *motif*. Since these regions are conserved, they may be recognisable by the presence of a particular sequence of amino acids called a *pattern*. A pattern is thus a qualitative description of a motif in terms of amino acid sequence.
The concept of a *profile* extends this concept, allowing a quantitative description of a motif, by assigning probabilities to the occurrence of a particular amino acid at each position of a motif. Thus profiles can be used to describe very divergent motifs.
The presence of a particular motif within a protein sequence can be used to suggest functions for uncharacterised proteins.
src:
http://homepages.cs.ncl.ac.uk/anil.wipat/home.formal/Bioinfoweb/section3/aboutMotifs.htm

A: Profiles indicates in which position it is possible to replace a residue without changing the structure of the protein. Profiles come from families, it considers the evolution aspect of the protein, where it evolves aiming to preserve structure.

Profiles increase the accuracy of the neural network by increasing the amount of data that can be analyzed.

Oli's answer:

- Profiles help in predicting secondary structure because they **condense information about the evolution of a protein**
- They also help in **improving alignments** because they take into account **position-specific scoring**, i.e. they tell us about the likelihood of replacing one residue with another at each particular position.
- In general, profiles profit from the **relations between protein families.** This is useful because proteins of the same family often form similar structures.

These are my notes of what he said during the recap session for Q8:

- Might be coupled with another one.
- Can be explain 1st generations, 2nd generations methods.
- Bullet point answers:
  - Profiles help because they contain more information, because they are derived from families.
  - Profile is the fraction of particular residues.
  - You can compile profiles in different ways.
  - **Crucial point:** proteins evolve under constraint, and they normally are to maintain the 3d structure.

How can you encode the profile?

A: by assigning probabilities to the occurrence of a particular amino acid at each position of a motif

or

1. Compile the degree given by Blosum62
2. Simply count how many have aa say ‘L’, then replace with the probability vector

Oli’s answer:

- A proper profile is built through a PSSM.
- There are 8 steps for creating a PSSM (see exercise 5):
    1) Calculate sequence weights
    2) Count observed amino acids and gaps
    3) Redistribute gaps according to background frequencies
    4) Add pseudocounts according to amino acid pair frequencies
    5) Normalize to relative frequencies
    6) Divide by background frequencies
    7) Calculate log-score
    8) Remove rows corresponding to gaps in the primary sequence

Q9: Explain the difference between sequence – sequence, sequence-profile and profile – profile comparisons.

-> In pairwise alignment we compare two strings letter-by -letter,

-> In sequence-profile comparison we compare a letter against a vector (which is a family of related sequences)

-> In profile-profile comparison we compare vector against vector (which is hard because there are many alternatives)

- **Sequence - sequence:** pairwise alignments used mostly in the Daylight zone. Letter by letter comparison. -Blast
- **Sequence - Profile:** Increases accuracy, used in the Twilight Zone and part of mid night zone. Letter - vector comparison. PSI Blast would be an example of this method.
- **Profile - Profile:** Compare a profile to the entire database of profiles. Vector-vector comparison. Increases accuracy, used in midnight zone.

Oli’s answer:

- Sequence - Sequence: align two sequences against each other (also called ‘pairwise alignment’). Do this if sequence identity is high (daylight zone). This is letter-letter comparison.
- Sequence - Profile: align a sequence against an entire profile. Do this if sequence identity is semi-low (twilight zone). We use a profile because it gives us additional information on residue replacement. We compensate the lack of sequence similarity with profile-information. This is letter-vector comparison.

- Profile - Profile: Compare two profiles against each other. Do this if sequence identity is low (midnight zone). Even if sequence alignment doesn't show any apparent relation, profiles might still reveal information about "distant relations".

Q10: Explain what a profile – profile comparison is and why it is difficult?

Compare one profile against the entire database of profiles, when building the profile, it is required to define a threshold to apply for new sequence comparisons. If the threshold is too high the relevance of the data is low, but if it is too low, there can be mistakes that lead to wrong results. Defining the right threshold of the profile is difficult. Also it is difficult to do profile - profile comparisons because there are too many free parameters (how to compare 2 vectors)

Oli's answer:

- In a profile-profile comparison, you compare two profiles against each other, i.e. you try to measure how similar they are. This is difficult because there are many free parameters (distinct dimensions, ambiguous residue replacement, …). It is also hard to define a "similarity threshold" (i.e. when does something count as being similar, when does it not?)

Questions from 28.June 2018

Q1: How can you predict per-residue secondary structure with a neural network? Explain in particular how to code for the sequence (<u>keyword:</u> sliding window).

The abstract of the paper from ROST:
….Here, we report a substantial increase in both the accuracy and quality of secondary-structure predictions, using a neural network algorithm. The main improvements come from the use of multiple sequence alignments (better overall accuracy), from balanced training" (better prediction of beta strands), and from "structure context training" (better prediction of helix and strand lengths).

Continued from the paper, this is probably the answer we are looking for :) (The answer of the question could be...)

- A sequence profile of a protein family, not a single sequence, is used as input to a neural network for structure prediction.
- Each sequence position is represented by the amino acid-residue frequencies derived from MSAs as taken from the HSSP database.
- The sequence is coded by sparse coding: for every amino acid there are 21 units (20 + 1 spacer unit)
- The input signal is propagated through a network with one input, one hidden, and one output layer. The output layer has three units corresponding to the three

secondary-structure states, helix, ,-strand, and "loop," at the central position of the input sequence window.

- The error function to be minimized in training is the sum over the squared difference between current output and target output values.
- The first network (sequence-to-structure) is followed by a second network (structure-to-structure) to learn structural context.
- Input to the second network is the predicted state from the first network, plus a fourth spacer unit. Inputs are propagated via a hidden layer to three output nodes for helix, strand, and loop, as in the first network.

This would be my answer:

The secondary structure can be predicted through a sliding window of consecutive residues. The secondary structure is predicted for the central residue. The sequence is coded by sparse coding: for every amino acid there are 21 units (20 + 1 spacer unit), where one of them will take the value of 1 and the other 20 will be 0.

Se

Oli's answer:

- You predict per-residue structure by having a vector of residues as the input to a Neural Network. The vector consists of a central residue (the one whose structure we're trying to predict) as well as his neighbours to the left and right. Once we predicted the structure for this residue, we move one residue further (--> sliding window principle).
- Why do we input a vector? Because structure is never local and mostly your neighbour residues will have the same secondary structure as you. The output of the Neural Network is a structure label, e.g. H (for Helix).
- The sequence is coded through sparse coding, this means for every residue there is a vector of size 21 (representing the 20 amino acids plus a spacer). Of these 21, the residue in question takes the value 1, all the other ones are set to 0.

Q2: What is the reason for a second level structure-to-structure network for predicting secondary structure? How much does this 2nd level improve over the 1st level (sequence-to-structure), and what is the 1st level?(answer to what is 1st level??)

A:

Reason -

- To increase accuracy of prediction
- To do balance training as in 1st level sample of Loops were too high comparing helices and beta strands, so the prediction accuracy for loops were much higher than helices and beta strands

How did it improve in second level-

- by using of multiple sequence alignments (better overall accuracy), so the vector changed from binary to amino acid-residue frequencies derived from MSAs
- through balanced training" (better prediction of beta strands)

How much improvements-

- Accuracy reached above 70% overall
- Accuracy of prediction for beta strands and length of helices increased

First Level:

- The sequence is coded by sparse coding: for every amino acid there are 21 units (20 + 1 spacer unit), where one of them will take the value of 1 and the other 20 will be 0.
- The input signal is propagated through a network with one input, one hidden, and one output layer. The output layer has three units corresponding to the three secondary-structure states, helix, ,-strand, and "loop," at the central position of the input sequence window.
- The error function to be minimized in training is the sum over the squared difference between current output and target output values.

==

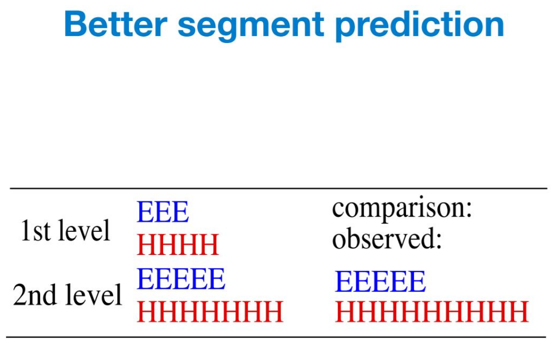
A: In the 1st level, you have no relation between two consecutive residues. For helixes, if one amino acid is helix, then the next three must also be helix otherwise it is predicted wrongly. So, to incorporate this context of neighbouring amino acids, we need a second level network. As we are sliding a window and considering neighbouring amino acids too, so the prediction is more consistent and in longer contiguous blocks, which is more realistic.

1st level predicts on average beta strands that are 3 E's long, helices that are 4 H's long. The second level increases E's to average 5 residues long and H's to 8 long. This looks more similar to the observed levels.

"More Explanation"

The length of the predicted segments are not long enough, that is why we stacked neural networks which is basically the second level.

The first level is the input of 21 sparse matrix of amino acids to the output of secondary structure, which itself is the input to the second level. The second level is trained exactly the same as the first layer. Just different input. In second level we teach the system the local correlation. The second level just refines what first level does.



Better segment prediction

1st level	EEE HHHH	comparison: observed:
2nd level	EEEEE HHHHHHH	EEEEE HHHHHHHHH

Oli’s answer:

- First level is described in the previous question.
- You use a second level to **learn about lengths**. E.g. helices have minimum lengths, so there’s no point in predicting a 2-residue-helix because such thing doesn’t exist in nature. A one-level Neural Network doesn’t know about this. However, if you add a second layer, its input will be a sequence of structure labels which enables us to predict structures of more realistic lengths.

Q3: Applying a method for secondary structure prediction fails for proteins with transmembrane helices (TMH). How could you adapt the known solution to the new problem?

A:

- Retrain with new data (membrane proteins)
- Problem: 2nd level now makes the helices much too long (-- too short I’d say).
- Fix: Just cut them? (More in the lecture…) {**please tell us more :(**)

  Alternative

  Prediction of TMHs fail because these membrane proteins have hydrophobic residues on the outside as well the inside while the non-membrane proteins have hydrophilic residues outside and hydrophobic inside. This change(our professor calls it 'outside-inside swap' if I'm not wrong) is the reason behind the very poor predictions (See slide 28 of the THM lecture).

  How could you adapt the known solution to the new problem?

  1. change the output of the network from H,E,L to helix (T), non-helix (N)
  2. only train on experimental TMH data
  3. Optimize the hydrophobicity index to predict the TM helix.
  4. Add second layer (structure-to-structure) to ensure sufficient residue length for TM helix

Oli’s answer:

- Transmembrane helices are generally longer than normal helices (necessarily, because they need to pass through those lipid membrane bilayer things, back and forth). If we use a Neural Network as our prediction method, we will often predict helices that are too short to be TMHs. Therefore, we retrain with experimental TMH data or simply add a constraint to ignore all predicted helices that are too short. Note: The Neural Network will have to have at least two layers, otherwise it can’t know about lengths.
- (I think I know what caused the confusion regarding too long vs too short:
  - Helices are predicted as too short if we train with normal data and expect TMH predictions
  - Helices are predicted as too long if we train with TMH data and expect normal helices as predictions)

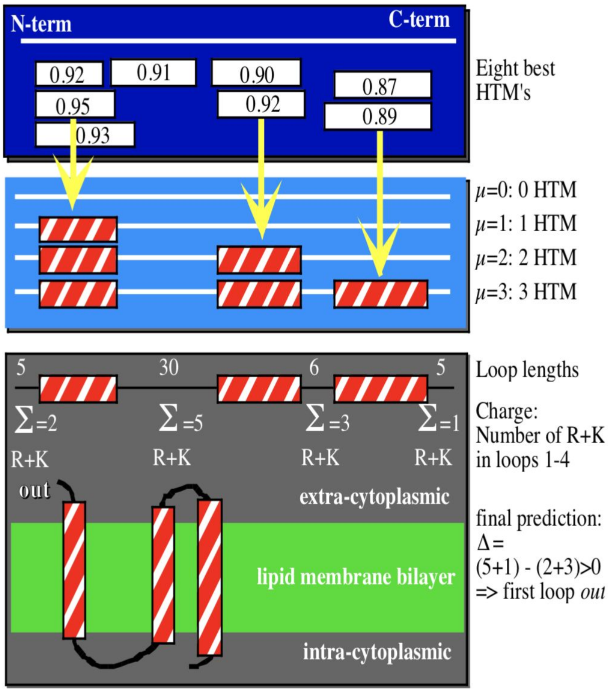
Q4?: For the prediction of transmembrane helices (TMH), the 2nd level network (structure-to-structure) works "too good", i.e. TMHs are predicted twice as long as those observed. How could you fix this problem? Sketch one possible solution.

Short answer based on the video lecture:

In case of TMH we replace the H, E, L in the output with TMH, not-TMH in the the neural network pipeline. Here, in contrast with the normal proteins (that the segmentations were too short and we needed the second level because helixes need to have a minimal length) the second level works “too good”. It means that now the observed helices are twice as long as the average. Or some of them are so longer. The hack solution here is to cut them. That is the simplest solution but the refine solution is to apply dynamic programming solution on that.

The idea is that the membrane helices have between 15 to 25-27 residues. So we will create a pool of all the segments between 15 and 25 that are compatible with membrane potentials and for each we will have an average whether it is a TMH or not. In each step we find the best model which is not conflicting with the previous ones.

PHDhtm



It's the hack to solve the problem of the network predicting TMHs that are twice as long as the ground truth. Say the network predicts T = TMH or N = No TMH, each with a probability. Given that TMHs are usually 15 - 25 residues long, you go through the predictions for the whole sequence and search for every subsequence that is 15-25 residues long, where every single residue was predicted to be T. That gives you the blocks in the dark blue box. For every block, you assign it the average T probability of it's residues. Now, you have all candidates for transmembrane helices. What you do now is you take the most likely candidate, i.e. the one with the highest average T value. In the box above it is the 0.95 box (left yellow arrow). Then, you pick the one with the next highest average T value that is not overlapping with the boxes you've already picked, because that would give a clash. That is the 0.92 block. Then again do it and you'll get the 0.89 block. The prof didn't specify a stop criterion for this procedure, he said you can stop, when the next block you add has an average T-value of < 0.5. Or you can stop of course, when there are no blocks left, like in the figure above. The result is that you have the TMH blocks and their exact positions.

How to fix: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3535531/

- If the predicted TMHs is between 30 and 50 residues, which means the TMHs is too long and is not factual, then the TMHs is expanded 10 amino acid residues from the two sides respectively and further we cut this TMHs into two equal parts to seek for potential TMHs.

- If the length of the predicted TMHs is greater than 50 residues, then the TMHs is cut into three equal parts using the same method above.

Oli's answer:

- We train the Neural Network with TMH data. TMHs are generally longer than normal helices, so if we attach a 2nd level (structure-to-structure), the Neural Net will tend to predict long helices and possibly miss out shorter ones.
- Solution: Cutting or Dynamic Programming (<-- not sure how this would work)

Q5: Profiles improve the prediction of protein secondary structure. Given two multiple alignments for a family, how can we assess which one is better (assume these two exist for proteins for which you know the secondary structure; suggest one way to check)? How can you find the better of two if you do NOT know the secondary structure? What aspects of the alignment are relevant to improve the prediction? ["more diversity and coverage" is this part of the last question or the 2nd last? please confirm]

A: The interesting part of a protein is the structure and function. We can judge which multiple alignment is better by finding out which multiple alignment contains more motifs. If the secondary structure is NOT known we can use profile-profile to find a secondary structure which is similar, which then can be used as a template. Then by comparing the motifs one can compare which multiple alignment is closer to the function of the template.
[does this make sense?]

A:

1. The one, that has the lower validation score [what validation score, DOPE, DP?]
2. Number of sequences & More diversity & Coverage

Potential Answer: Profiles need to be able to show evolutionary information from the family. This means that the profiles need to have a meaningful number or patterns for conserved AA segments. So we judge the goodness of the profile by looking at how much conservation is going on, how relevant are the positions of important amino acids. If we have the 3D structure from some members of the profile, can we see why certain AA are preserved in certain positions? (they might be far away in sequence but physically they interact to shape the protein) Do rare AA get conserved in certain positions or are they slightly overpowered by more common ones? (the presence - or lack thereof of rare AA is important to notice).

Oli's answer:

- Not sure what he's asking for but you can improve a profile by making it more "informative". You make it more informative by adding more aligned sequences and especially by adding sequences that are "diverse" i.e. sufficiently distinct from the ones that are already there.

There some part in the membrane lecture which he said that we need to apply the inside out rule and optimize it by putting the thresholds. So basically we do the optimization for every single protein without knowing the structure. And the aspect that is relevant is Hydrophobicity [Does this make sense?]

Q6: Give an example for a method that predicts protein disorder through machine learning without using any experimental data about disorder. This method uses no positive (disorder) "like" what it is supposed to predict and uses many negatives (not disorder) that incorrectly labelled. How can it still work?[not complete yet]
A: See absence of order (constant signal?), absence of a clear signal.

Only signal is consistent. In dataset with more than 70 residues, there is something about loopy disorder, which are conservation, solvent accessibility etc.

I understood that even if you predict the long residues when you want smaller ones, you can see the conservation of smaller parts in your true dataset.

One example is the method that predicts short NORS regions or distinguishes between unstructured and well-structured loops. For it we need two datasets, the true one, containing everything that has been already predicted that have no regular structure, and the false one, that contains the entire PDB, where every structure is ordered. We train the machine learning algorithm to learn common features inside each dataset, that distinguishes the datasets with each other, and it works because of the Signal, which is the only consistent thing. The mistakes are not consistent. Therefore, by capturing the signal, it can distinguish between 30 residue loops that are well structured and 30 residue loops that are disordered.

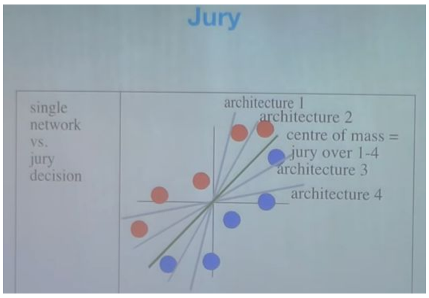
Q7: For secondary structure prediction, there are neural networks trained in "balanced" and in "unbalanced" fashion. What is the difference? What is the advantage of "balanced", what the disadvantage? Why could it help to compute a "jury" (average) over balanced AND unbalanced predictions? What type of error can be improved by averaging over many methods?[not complete yet]
A:
Difference:

- Balanced fashion, shares data equally between helices, strands and others, whereas unbalanced gives more data to one of the classes.
- Balanced fashion gives equal accuracies. Unbalanced data gives better accuracy when the majority class predicted correctly.

Advantage: Balanced training can predict beta-strands with higher accuracy as opposed to unbalanced.
Disadvantage: Requires data adjustment / replication for predicting classes having lower number of samples to train
Jury- Different Methods draw different separation lines to predict and in each of these lines there remains some white noise (errors in the prediction that are not systematic). By

averaging the these lines (“jury” calculation) reduces (removes?) the white noise and performs better.

What type of error can be improved by averaging over many methods? - white noise (non-systematic errors)



Oli’s answer:

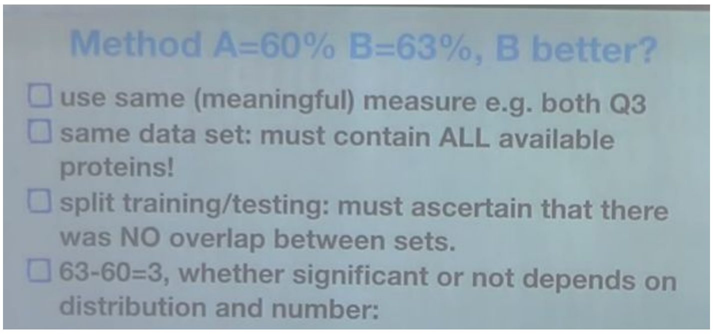
- Unbalanced training: feeding input data to our Neural Network with an “as it is” secondary structure distribution. Problem: Loops occur more often than helices, helices occur more often than strands, so there will be a tendency to e.g. predict helices more often than strands.
- Balanced training: choose input data in such a way that the distribution of secondary structure is equal (e.g. throw in more beta strands because they occur less often). Benefit: get rid of prediction bias

Q8: Define five different criteria to measure the performance of secondary structure prediction.

The success of a prediction is determined by comparing it to the results of the DSSP algorithm (or similar e.g. STRIDE) applied to the crystal structure of the protein.

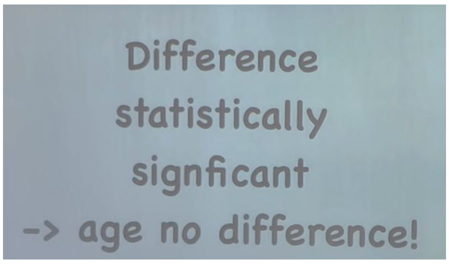
$$Q3 = \frac{\textit{number of correctly predicted residues in states helix, strand, other}}{\textit{number of residues in protein}}$$

I’ve thought that maybe he wants the 5 criteria to decide whether method A or B is better



Method A=60% B=63%, B better?

□ use same (meaningful) measure e.g. both Q3
□ same data set: must contain ALL available proteins!
□ split training/testing: must ascertain that there was NO overlap between sets.
□ 63-60=3, whether significant or not depends on distribution and number:



Difference statistically signficant -> age no difference!

1. Do they use the same measure? (for example Q3)
2. Do they use the same ~~test~~ data?
3. Is it ensured that there is no overlap between test data and train data?
4. Is DeltaQ3 significant? (stderror: sigma /sqrt(#ofInstances)
5. Cross-training
6. ~~How old is A and B?~~
7. Use all of the available protein data and split it between training and testing. [as per slides, part of this statement is part of second point, i think]

I also think 1-3 describe scientific significance, whilst 4 describes statistical significance.

scientific significance depends on “what matters” in targeted task to be accomplished

Q9: Statistical significance? What are statistically significant digits? What data set size is relevant? Standard Varianz?

A: **statistical significance:** Method A = 60%, Method B = 63% => deltaQ3 = 63-60 = 3
stderr = sigma/sqrt(# of proteins).
If the delta between the new method and existing method is larger than the stderr of the new method then it is statistically significant.

Oli’s answer:

- Delta = difference between accuracies of two methods
- Delta is statistically significant if it is larger than the standard error
- Standard error = sigma / sqrt(number of proteins)

Q10: What is the positive-inside rule? How can it be used to improve transmembrane helix (TMH) prediction?

A: The "positive inside" rule is an observation made by Gunnar von Heijne where he noticed that in membrane proteins, the collective charge of the intracellular residues tend to be more positive as compared to the extracellular. This occurrence can be used to try to predict the normal orientation of the membrane protein in the membrane.
[https://www.quora.com/For-membrane-proteins-what-is-the-positive-inside-rule]

Alternative Answer:
(G von Heijne looked at all regions that connect the membrane helices and looked at the difference in their biophysical features.)
-> Positive-inside rule: There is an excess of positive charged residues inside of a cell. With this we can determine which region is inside and which is outside of the cell by analyzing their charge.

Oli’s answer:

- The positive-inside rule states that “the distribution of positively charged residues in bacterial inner membrane proteins correlates with the transmembrane topology”.

- In other words: TMHs are always oriented in such a way that there are more positively charged residues inside the cytoplasm than outside. By counting the charges inside and outside, we can infer the orientation of the TMHs.

Q11: Why do we need more “global” information for the prediction of beta-strands than for alpha-helices? Answer what most people (as e.g. represented by Wikipedia) believe. What did you learn in the lecture that tells you that this answer is incorrect?
A: Strands are not local: beta strands span over a larger window than our sliding window which makes it more global - that’s why you need more global information => wrong, because through neural networks and balanced training we can predict both equally well.

Oli’s answer:

- Helices are local in the sense that the hydrogen bond partners that stabilize the structure are very close to each other (only every 4 residues away)
- Strands form hydrogen bonds between residues that might be very far away in sequence, that’s why they’re called ‘non-local’. The hydrogen bond partners are still physically close but they are far away **in sequence**.
- Because of this non-locality, strand prediction needs more “global” information. “Global” in the sense that we need to have (shared) information about residues that are many “sequence-steps” away from each other.
- We can get rid of this distinction by predicting structure through a Neural Network that underwent balanced training. With balanced training, we can predict helices and strands equally well (and ignore the local-vs-non-local issue).

Q12: How can you use the amino acid proline to predict helices?
A: proline causes a fold so if any helix precedes it, it will always end at proline, the latest proline breaks helices because it is a ring form, hence can be used to predict helix/non helix

Oli’s Answer:

- Proline induces a change in the angle of the protein
- → proline bends main chain
- → proline breaks a helix
- So: whenever you encounter Proline, interpret this as a potential end of a helix
- Constraint: There have to be sufficient residues in between the Prolines because of the i--i+3--rule and the fact that helices span several turns

Q13: UCON, NORSnet, MetaDisorder - how can we predict disorder?
A: no regular secondary structure
unstructured regions from contact prediction
meta disorder predictor

I

MD (Meta-Disorder predictor)

A neural-network based meta-predictor. MD significantly outperformed its constituents. MD is capable of predicting disordered regions of all "flavors", and identifying new ones that are not captured by other predictors.

NORSnet

A neural network based method that focuses on the identification of unstructured loops. NORSnet was trained to distinguish between very long contiguous segments with non-regular secondary structure. One disadvantage of this approach is that it is not optimal for the identification of the "average" disordered region.

Ucon (prediction of natively unstructured regions through contacts)

A method that combines protein-specific internal contacts with generic pairwise energy potentials to accurately predict long and functional unstructured regions. One advantage of Ucon over statistical-potential based methods is that it incorporates the contribution of the specific order of the amino-acids rather than the amino acid composition alone.

ADDITIONAL QUESTIONS (e.g. during lectures,..):

Q1: Where to put the threshold of B-value (B-Factor?) in backbone flexibility? For rigidity or flexibility?

A: Why not at peak? Any error has the strongest impact on the peak. It’s always possible to have error! keep threshold at the place, where the effect is minimum.

Q2: What is a dark-proteome?

A: No 3D-structure is known. Some part is disordered, transmembrane, but rest is unknown.

Q3: Through what is a secondary structure stabilized?

A: Hydrogen-bonds

Last Years’s Question List :

(https://docs.google.com/document/d/1N4FvlcFZbG5m8KSEK3RDQRTC1c8jjcAh1yYcDEKZ5jk

I took the freedom to append last years questions to this document at the end (afaik most exam questions from last year are in there)

Quick Numbers:

How many proteins are known: 116 Million (UniProt: https://www.uniprot.org/)

For how many proteins do we know structure: 140 000 (PDB: https://www.rcsb.org/)

How many secondary structures can we predict through Comparative Modeling?: roughly half => +- 50 Million

GOOD LUCK EVERYONE!

STEAL HIS LOOK
RUSSIAN SCARF "LYUBAVA" 50€
OUR LEGACY ORIGINAL BUTTON DOWN SHIRT 169€
STAN RAY SLIM PAINTER PANTS 95€
POST OVERALLS ROYAL TRAVELLER VEST 415€

https://www.youtube.com/watch?v=mqPsNazEipA

And believe in “proteinism”!

Here’s all of Oli’s answers:
https://docs.google.com/document/d/1DZMffmwNbuvA1SpEMkkB6C8VfYoxiaW25FkGRWtYpD0/edit?usp=sharing

Oli 4 President!
Give Oli a cookie! :D
Or better a Beer :D
-- lol, cheers

! There you go Oli! -- Yummy! For Oli! Cheers!

Haha Chiemseer is good choice! It's Tegernseer :v LOL Haha ok -- cheers! :D

Guys there is more next page…

Last year's Recap Questions

(mostly used in last year's exam)

1) What is the assumption behind almost all alignment methods that is incorrect and nevertheless seems to work? Give a method that aligns without needing to use this assumption.
   Answer: (i, i+k) -independent; T-coffee
   Oli's Answer:
   - they all make the independence assumption, i.e. they assume that alignment at position j is independent of alignment at position i (where j > i)
   - Problem: this is far from reality
   - Solution: Use a genetic algorithm

2) When I build a profile on a family: do they share the same structure? Should I verify that they do? How could I do that?

3) How many protein sequences are known (roughly)? How many experimental 3D structures are known (roughly)? Why is the difference between the two numbers so high? Guess how it will be in 10 years from now?

4) What is homology modeling (comparative modeling)? What are possible limitations?

5) When you run a profile-profile comparison between profile P1 and P2: What is the assumption for the proteins within each of the two profiles? What the assumption between the two profiles (if the "match")?

6) Explain the difference between pairwise alignments and sequence-profile alignments? What can you achieve/gain? What is the risk?

7) We discovered a sequence: what do we do next? First, how do you get a name?

8) What is an E-value? Given an alignment of a pair of proteins with an E-value = 10-3, what does this number signify? How do we adjust E-values in a BLAST search?

9)You want to develop a new method to predict B-values. How do you prepare your data?

10) What are structural domains? How can we deal with the fact that they occur at different positions in the protein?

11) What is the difference between a BLOSUM substitution matrix and a position specific substitution matrix?

12) What is the most successful method to predict 3D structure?

13) How can you predict structure in the daylight, twilight, and midnight zone?

14) Say the 3D structure were known for N thousand proteins and these serve as a base for a method predicting 1D structure (e.g secondary structure) How can you define the value for "sequence-unique" that you have to apply to create an unbiased data set? Why do you need an unbiased data set?

15) What is the significance of using information from protein families (also referred to as evolutionary information) as input to the machine learning device predicting 1D structure?

16) How can you get from 1D structure prediction (e.g, secondary structure) to reconstructing 3D structures? What can you do with secondary structure prediction?

17) Explain the concept between the notation 3D,2D,1D structure. What is in the PDB? What does DSSP give?

18) Why do we need separate methods to predict secondary structure for membrane and water-soluble/non-membrane proteins?
What is needed for membrane prediction beyond secondary structure?

19) What is the principle difference between PSI-BLAST and CLUSTALW (or any other multiple sequence alignment method)?
Possible answer: Dynamic programming find the alignment but is too slow. So, you try to speed up. The main difference is the way in what they compare statistics.

20) In the first iteration PSI-BLAST finds the most low-hanging fruits through pairwise comparisons, What does it do in iteration 2? Why can this work better?
What could happen that makes iteration n+1 not find more distant relatives than iteration n (for all n=1 ...N)?

- Convergency

Say n+1 find many more hits than n: all green?

21) What do per-residue scores poorly reflect the performances of transmembrane prediction? Invent an alternative method to score TMH (transmembrane helix) prediction methods.
Ans: Answer for this is in the TMSEG method video. Michael gives the answer which I don't remember.

22) Method A is published to predict solvent accessibility at Q2=61%, Method B in another publication claims to achieve Q2=63%. What do you have to check to ascertain that method B is really a better method? (address the terms statistical significance and scientific significance in your response)

- Ideally you would test A and B on new proteins.

23) What features can be used to predict secondary structures from sequence? Argue why?.

Most important feature is PSSM (profile)

24) What is the most accurate way to predict protein 3D protein structure? (explain the idea behind the method: naming it is good but not enough) Why does this method hardly work for membrane proteins and even less well for disorder proteins?

25) UniProt currently holds about 85 million protein sequences: Do we have any idea about the structure of any of those? Roughly for how many? Do we have any idea about how many of the 85M are membrane/disorder?

26) You want to use a regular neural network (input/hidden/output) to solve a certain prediction problem (e.g. predict disordered regions). How can you find how many hidden units you need? How waht the best input is? How to best code the output (here disorder)?

27) You want to develop a method to predict secondary structure in 3 states(HEL). You use DSSP to convert the 3D structure in the PDB into HEL. What do you have to watch? Can you use the entire PDB? Once you have N proteins in your data set: how can you assess prediction performance? (How to measure “right”, name a few scores that are relevant, how to measure statistical significance, how to measure scientific significance?)

28) TMH prediction: how can you predict the direction the of a helix? What assumption does comparative modeling make? Why do proteins always have to adopt the same 3D structure? Do different organisms use different proteins? How much does it cost (time/money) to experimentally determine the 3D structure of average protein?

29) How would you define life? How are proteins crucial to maintain it?

30) why do we need membranes around cells? What are they made of? Why do we need proteins that pass through membranes?