Integrative Transcriptomics
Prof. K. Nieselt,
Institute for Bioinformatics and Medical Informatics Tübingen
Prof. S. Nahnsen,
Institute for Bioinformatics and Medical Informatics Tübingen

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Lecture: Grundlagen der Bioinformatik SoSe 2022

Assignment 12 - the last one! (20 points)

Notice: We upload the last assignment already one week before the official start for you to decide how you would like to distribute your time. Consider that some topics will be first handled on the lectures of the 18th and 20th of July.

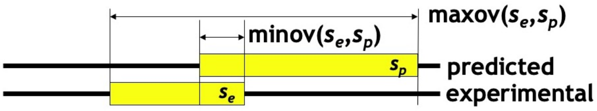
Hand out:	Thursday, July 14
Official start:	Thursday, July 21
Hand in due:	Thursday, July 28 18:00

Direct inquiries via the ILIAS forum or to your respective tutor at:
Mathias Witte Paz: iizwi01@uni-tuebingen.de
theresa-anisja.harbig@uni-tuebingen.de
meret.haeusler@student.uni-tuebingen.de
jules.kreuer@student.uni-tuebingen.de
simon.heumos@qbic.uni-tuebingen.de

1. **Fractional Overlap of Segments (SOV)** (8P)

   Background information: An alternative measure for Q_3 to compute the quality of a secondary structure prediction is the segment overlap score (SOV) ([1],[2]), that computes an average overlap between the experimentally determined and the computationally predicted segments instead of the average per-residue accuracy as it is done in Q_3. SOV is based on the comparison of pairs of secondary structure segments, and is defined as follows:

   Let $S_e(k)$ and $S_p(k)$ be the experimentally and computationally predicted segment of type $k \in \{H, E, C\}$, respectively. Define minov(S_e, S_p) to be the length of the intersection of S_e and S_p, and similarly define maxov(S_e, S_p) to be the length of the union of S_e and S_p (see illustrating figure).



   Then

$$SOV = \left[\frac{100}{N} \sum_{k \in \{H,E,C\}} \sum_{(s_e,s_p) \in S(k)} \frac{minov(s_e,s_p) + \delta(s_e,s_p)}{maxov(s_e,s_p)} \cdot len(s_e) \right]$$

   where $S(k)$ is equal to the set of all overlapping pairs (s_e, s_p) of S_e and S_p where the segments are in the same state k and

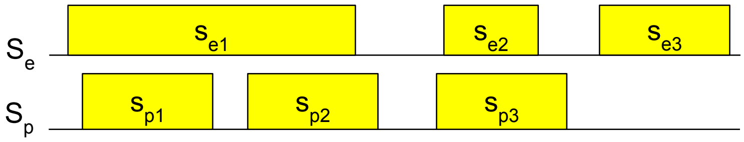
$$\delta(s_e, s_p) = \min \begin{pmatrix} maxov(s_e, s_p) - minov(s_e, s_p) \\ minov(s_e, s_p) \\ \lfloor (len(s_e)/2) \rfloor \\ \lfloor (len(s_p)/2) \rfloor \end{pmatrix}.$$

Note that $\lfloor x \rfloor$ denotes the lower integer truncation of x.

Furthermore, $N = \sum_{k \in \{H,E,C\}} N(k)$ with

$$N(k) = \sum_{(s_e, s_p) \in S(k)} len(s_e) + \sum_{s_e \in S'(k)} len(s_e)$$

with $S'(k)$ is equal to the set of segments in S_e without any overlap with another segment with the state k. The following figure can help you to understand the computation of $S(k)$ and $S'(k)$. Consider all elements of the same color to be from the same state k.



$S(k) = \{(s_{e1}, s_{p1}), (s_{e1}, s_{p2}), (s_{e2}, s_{p3})\}$ $S'(k) = \{s_{e3}\}$

Task: For the following experimentally derived secondary structure:

$S_{\text{obs}} =$ `CCCEEEEEECCCEEEEEECC`

and the two predicted structures

$S1_{\text{pred}} =$ `CCCCEEECCCCCEEEEECCC`

$S2_{\text{pred}} =$ `CCCECEEECCCEEECECECC`

compute the respective SOV as well as Q_3. For the computation of the SOV, please also hand in the intermediate steps of your calculations.

Compare the two values for each predicted structure. What do you conclude?

References:

[1] Rost B, Sander C, Schneider R: Redefining the goals of protein secondary structure prediction. J. Mol. Biol. 1994, 235:13-26.

[2] Zemla A, Venclovas C, Fidelis K, Rost B: A modified definition of SOV, a segment- based measure for protein secondary structure prediction assessment. Proteins 1999, 34:220-223.

Practical Assignments

2. Predict secondary structure of proteins (8P)

In this task, you will predict the secondary structure of a protein using two different programs and afterwards compare them with the true secondary structure using your own Python script. For this:

(a) Use the secondary structure prediction program `PHD`[1] to predict the secondary structure for the amino acid sequence found in the file `secStructure.fasta`. `PHD` does not provide you

[1] `http://www.predictprotein.org`

directly with the secondary structure sequence. The easiest way to extract this sequence is to look at the XML file and locate the `featureString` element under the `PROFsec`. You are allowed to only copy-paste this string into a `txt`-file for the usage in the future steps.

(b) Next use the program `Proteus2`[2] to also predict the secondary structure for the same amino acid sequence. Here you need to save the sequence in the section *Predicted Complete Secondary Structure* into a `txt`-file for the next steps.

(c) Create a Python that parses your files containing the predicted structures and computes the `Q3` value using the true secondary structure (`trueSecStructure.txt`).

(d) Conclude: Which prediction method is more accurate?

3. 3D protein structure prediction (4P)

For this task we ask you to predict the 3D structure for the amino acid sequence found in the file `3DStructure.fasta`. For this please use three different programs:

(a) phyre-Server: estimated run time ≈ 7-8 hours (`http://www.sbg.bio.ic.ac.uk/phyre2`)

(b) iTasser: estimated run time ≈ 16+ hours (`http://zhanglab.ccmb.med.umich.edu/I-TASSER/`). This program requires a registration, but you can use it for free.

(c) AlphaFold2: estimated run time ≈ 1 hour (`https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb`)

Compare the results of the three servers by visualizing them. You can use the `RCSB PDB 3D Viwer`[3] if your tool only returns a PDB file. Do the predictions of the tools agree with each other? Include a screenshot for each prediction in your PDF. From the results try to deduce the function of the protein.

Note: the computation may take some time so be sure to start the jobs early.

Please read the questions carefully. If there are any questions, you may ask them during the tutorial session or in the forum of ILIAS. You will usually get an answer in time, but late e-mails (e.g. the evening of the hand-in) might not be answered in time. Please upload all your solutions to ILIAS. Don't forget to put your names on every sheet **and** in your source code files. Please pack both your source code as well as the theoretical part into one single archive file and give it a name using this scheme: `<name1>_<name2>_<Assignment>_<#>.zip`. The program should run without any modification needed.

[2] `http://www.proteus2.ca/proteus2/index.jsp`

[3] `https://www.rcsb.org/3d-view`