Softwarewerkzeuge der Bioinformatik

Prof. Dr. Volkhard Helms
PD Dr. Michael Hutter, Markus Hollander,
Andreas Denger, Marie Detzler, Velik Velikov
Winter Semester 2021/2022

Saarland University
Center for Bioinformatics

Exercise Sheet 2

Sequence Analysis: Pairwise Alignments

Learning objective: *In this tutorial you will gain practical experience with the Needlemann–Wunsch algorithm and various BLAST searches. Additionally, you will get familiar with Jupyter Notebooks that we will use in the coming tutorials to learn the programming language Python.*

Exercise 2.1: Dynamic Alignment

Compute a **global** alignment of the sequences **ACDEFAFGHI** and **KDELAFG** using the **Needlemann–Wunsch** algorithm.

		A	C	D	E	F	A	F	G	H	I
K											
D											
E											
L											
A											
F											
G											

Global alignment:

Exercise 2.2: ProteinBLAST

Run a **ProteinBLAST** search (http://blast.ncbi.nlm.nih.gov/Blast.cgi) for the protein **P00042**. Choose the **UniProtKB/Swiss–Prot** database to receive non–redundant and high quality results.

Under “Algorithm parameters” there are additional settings that you do not have to change right now. By default, ProteinBLAST uses an E–value threshold of 0.05 and BLOSUM62 as the matrix. These settings are suitable for finding many similar and likely homologous proteins with a medium amount of relatedness.

a) Find the 10 proteins with the highest homology to P00042 and display their sequences.

b) Select all results again. What of proteins are we dealing with?

c) For which types of organisms were results found?

Exercise 2.3: MegaBLAST

Select **human** as the genome on the BLAST main page. Search for the mRNA **NM_175054** of the human gene *HIST4H4* with **megaBLAST** in the database **Genome (GRCh38.p13)**.

a) On which chromsome is *HIST4H4* located?

b) Is there a paralogue?

c) Find two or three directly neighbouring genes of *HIST4H4*.

Exercise 2.4: PSI–BLAST

Use **ProteinBLAST** to search for **many very distantly related homologues** of the protein **Q57997** in the UniProt database. Set the E–value threshold to 0.02 and the “PSI–BLAST Treshold” to 0.01 in order to find homologous proteins, and select the BLOSUM45 or PAM250 matrix to find many distantly related proteins.

a) What is special about the 1. PSI–BLAST iteration?

b) How do the results change with further iterations?

Exercise 2.5: Python and Jupyter Notebooks

In order to become familiar with Python, we will use **Jupyter Notebooks**. A notebook consists of cells that either contain executable code or notes, e.g. for descriptions and explanations. Notes can be formated with *markdown*. Jupyter Notebooks are thus well suited to display code and analyses.

a) **Preparations:** To avoid software installations, we will use the website kaggle.com where you can create notebooks and save and run them in the cloud.

   i. Create a user account on kaggle.com. You can either use a Google account or register with your email address. In the latter case you will receive a confirmation email. Click on the link in the email to activate the account.

   ii. Log into your account and create your first notebook. On the left side of the website is the button **+ Create**. Click on it and select **New Notebook**. New notebooks already use Python as the programming language by default. You can click on the name of the notebook on the top left to rename it. Choose a name that you will remember, e.g. SWW Tutorial 2.

   iii. Your notebook is now saved in your account. When you click on **<>** or **<> code** on the left, you can find all notebooks under **Your Work**.

b) Your notebook already contains a cell. Select it and click on the paper bin icon at the top left to delete its content so that we can begin from scratch.

c) Write $2 + 2$ into an empty cell and click on the arrow button left of the cell. The result will be displayed beneath the cell.

d) Create a new code cell with the **+** button, write *print('Hello World')* into the cell and run the cell. The string *Hello World* should be displayed beneath the cell.

e) You can assign values to variables. These variables are then saved in the notebook and can be reused later. Write $a = 5$ into an empty code cell and run it. Create a new cell and just write a. This will display the value that is saved in a. When the last line contains or returns a value, it will be displayed beneath the cell.

f) Create, edit, run and delete a few cells until you get the impression that you understand the basic concepts of notebooks.

Have fun!