

Winter term 2016/17

Bioinformatics II

Assignment Sheet 1

If you have questions concerning the exercises, please write to our mailing list:
vl-bioinf@lists.iai.uni-bonn.de.

This week, we ask you to install the software that will be required for the practical projects. We encourage you to read the Python tutorial and try a very simple exercise in order to become familiar with the language. This sheet will not be part of the project itself; you will not have to hand in anything from it.

Exercise 1 (Install and Try Python, 0 Points)

Throughout the semester, we will use the programming language Python to perform visualization and image processing. You will need the following open source software, which is available for Linux, Windows, and Macs:

- The Python interpreter itself, which is the basis for running any Python program.
- The Python packages NumPy, SciPy, matplotlib, pandas, and scikit-image. Please note that these packages may not be supported by all version of Python. We test our own model solutions using Python 2.7.
- It's very helpful to use an integrated development environment that supports Python. One possible choice is [Spyder](#).

You can find more information on installing these packages at <http://scipy.org/install.html>. Alternatively, for the first time this semester, we are offering VisCoSiTea, a special Linux live distribution that you can simply run in a virtual machine or boot from a USB stick and that contains all software required for the assignments. If you would like to try it, please find further instructions at <https://cg.cs.uni-bonn.de/en/activities/visualization-and-medical-image-analysis-group/viscositea/>.

Once you have the required software, we encourage you to do the following simple exercise, especially if you have not worked with Python before:

Create an $m \times n$ array, assign a unique integer from $[0, \dots, m \times n - 1]$ to each element, and output the result. The top left element should get index 0, and indices should grow first from the left to the right, then from top to bottom, for example:

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \end{pmatrix}$$

Please proceed in the following steps and submit your final script:

- a) Define a function that, given indices (i, j) and the size of the matrix, returns the corresponding integer.

- b) Read the desired matrix size (m, n) from the command line. You may use the getopt package for this.
- c) Create a matrix of the desired size. Use the function from a) to fill it.
- d) Output the matrix to the terminal.

Hint: If you are used to C++ or Java, you should be aware of the fact that Python does not use brackets to group statements. Instead, code with exactly the same indentation (be careful not to confuse spaces and tabs!) forms a block, e.g., the body of a loop. For a more detailed introduction to Python, please refer to <https://docs.python.org/2/tutorial/>.

Good Luck!