

Winter term 2016/17

## Bioinformatics II

### Assignment Sheet 9

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

*We strongly encourage you to continuously work on the assignments and contact us with questions. However, you will only have to hand in your results (for all sheets of the second project) on January 31.*

### Exercise 1 (Threshold-Free Cluster Enhancement, 10 Points)

In this exercise, you will implement a 2D version of the Threshold-Free Cluster Enhancement (TFCE) algorithm and apply it to the statistical parametric map in `spm.npy` which you can download from the lecture webpage. The 2D map can be seen as an image with a  $t$  score at each pixel. Since  $t$  scores are signed floating-point numbers, we store it in numpy's `.npy` format rather than in a traditional image format.

- Write a function that, for any given threshold value, creates a binary mask and a label image. Labeling means that you should assign a unique identifier to each connected component of the mask. In the label image, the value of each pixel should be the identifier of the component it belongs to. Feel free to make use of the functionality in `scipy.ndimage`. (3P)
- Write a function that, given the label image, creates an extent image which, for each pixel, should contain the extent (i.e., size in number of pixels) of the component it belongs to. (3P)
- Write a loop that iterates over a given number of threshold values (between 0 and the maximum found in the input) and, for each pixel  $p$ , performs numerical integration of the TFCE integral

$$TFCE(p) = \int_{h=0}^{h_p} e(h)^E h^H dh \quad (1)$$

with the commonly used values  $E = 0.5$ ,  $H = 2$ . (3P)

- Try different step sizes for the discretization of the integral (i.e., different number of threshold values). How does the resulting image change? (1P)

Please submit your script and the TFCE-transformed input for at least two different discretizations.

### Exercise 2 (Hemodynamic Response Function, 6 Points)

- Sample the “double gamma” canonical hemodynamic response function at 0.1 s temporal resolution and plot it. (2P)  
*Hint:* You can make use of the gamma distribution available in `scipy`.
- Using the result from a), predict the response to two box-shaped stimuli, which should have unit magnitude and 1s or 2s duration, respectively. Perform the required computations numerically at 0.1s temporal resolution, and plot the results. (2P)  
*Hint:* You can make use of the methods for convolution available in `numpy`.

- c) Downsample your results from b) to the 2.5 s temporal resolution that is more typical of fMRI. Also perform the convolution at 2.5 s resolution and compare the results. (2P)

### Exercise 3 (General Linear Model, 9 Points)

- a) In the following diagram, which line is the best linear regression for the given data points? Why? Which line is the worst? (3P)

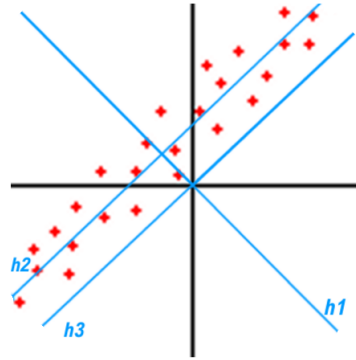


Figure 1: Linear regression.

- b) Given the following data set:  $(x_{1,i}, x_{2,i}, y_i) : \{(2, 5, 2), (1, 4, 7), (3, 1, 11), (4, 2, 4)\}$
- (a) Write down the corresponding matrix  $X$ . (2P)
  - (b) Use GLM to compute the  $\beta$  vector. (4P)

**Good Luck!**