Python exercises for the lecture "Models and Algorithms in Bioinformatics"

Organizers: Peter Meinicke, Sina Garazhian
Submission date: 01.02.2025. 12 p.m.

Sheet 5: Self-organizing maps (SOM)

Task 1: Algorithm: Topographic vector quantization

a) Implement a 1D-SOM ($q = 1$) with the TVQ algorithm, as presented in the lecture. Use a fixed number of $K = 20$ prototypes (reference vectors) and test the method on synthetic 2D data with seven normal distribution clusters with centers $(0,0), (1,0), (0,1), (1,1), (0.5,0), (0,0.5), (1,0.5)$ and standard deviation 0.1 with 50 points each, i.e. a total of $N = 350$ 2D data points. In analogy to the K-means algorithm, use the $K \times N$ matrix of all mutual squared Euclidean distances between data vectors and reference vectors. Use for the neighborhood functions (confusion probabilities) the precalculated matrix hmat with

```
h_mat = np.exp(-0.5/sigma2 *
        (np.reshape(grid_vec, (n_nodes,1)) - grid_vec)**2)
```

and the confusion probabilities in

```
prob_mat = h_mat / h_mat.sum(axis=0)
```

Try to calculate the assignments and estimate the reference vectors as far as possible with the given matrix functions and operators. In particular, the (re-)estimation of the matrix of reference vectors can be efficiently implemented as a matrix product! Use the convergence criterion of the K-means algorithm for the termination condition. Repeat the clustering for a sequence of descending σ(sigma) values:

```
sigma_vec = 0.5 * np.logspace(1,0,n_steps)
```

Start with a random initialization of the reference vectors for the largest σ-value and use the resulting reference vectors of the first clustering as initialization of the next initialization of the next σ-level.

b) Visualize the reference vectors for the 2D dataset with points connected by lines using plot. Repeat the stepwise error minimization several times and try to optimize the sequence of decaying sigmas. Can the U-shaped structure of the distribution also be mapped for a shorter sequence?

c) Apply the procedure as above to the gene expression data. Visualize the matrix of reference vectors (e.g. imshow). Output the histogram of the cluster sizes. Is a smaller number of actual clusters recognizable in the data or to be assumed?

Submission instructions

Please submit your solution by **1st February** using the corresponding homework folder in StudIP. Upload a **zip** compressed file containing all documents and data that you use in your solution. The file name should indicate the author, for example *Peter_Meinicke_Sheet5.zip*.
With your solution you should provide the Python source code together with the documentation within a **Jupyter Notebook** answering all the questions above. Use Markdown cells for your answers and explanations well separated from the Python code.
For the programming in Python/numpy do not use any code or functions from special bioinformatics libraries or toolboxes! Make sure that your code is readable and understandable for others, i.e. use descriptive names for the variable definitions according to the conventions on **Sheet 0**, do not use implicit variables (hard-coded values without a variable) and make use of comments!