Programming Lab I
Handout 9

Dr. Martin Vogt
martin.vogt@bit.uni-bonn.de

Jun 30, 2020

Deadline: Jul 15, 2020 **Next Handout: -**

Analyzing RNA sequences

Transcription is the initial step in DNA based gene expression. However transcription is pervasive throughout the genome, which produces not only mRNAs but also a variety of non-coding RNAs, such as long non-coding RNAs and promoter upstream transcripts (PROMPTs)[1]

Unlike mRNAs, those non-coding RNAs are unstable and subject to rapid degradation in the nucleus by a nuclear RNA degradase called RNA exosome. Previous studies on the determinants for rapid exosome degradation by comparing the two classes of RNA, i.e. mRNA (exosome-insensitive) and non-coding RNA (exosome sensitive), suggest that genomic sequences play a critical role.

For our exercise, a set of 1000 RNA transcripts are given. Of these, 500 are exosome-insensitive "stable" transcripts and 500 are exosome-sensitive transcripts. For each transcript, only the first 1000 nucleotides are given. To find the sequence determinants, a first step in the analysis would be to look for differences in the nucleotide composition of the sequences, which is the objective of this exercise.

1. *(20 pts) k-mer distributions*
   A straightforward way for examining sequence composition is the investigation of k-mer frequencies. A k-mer is any subsequence of length k of a transcript.

   (a) *(5 pts)* Write a function `kmer_frequencies(seqs,k)` that takes a set of m RNA sequences `seqs` and a parameter `k` as input and returns the relative frequencies for all k-mers contained in the sequences as a dictionary. Note, that each sequence of length n contains $n-k+1$ k-mers.

   For the calculation of relative frequencies use *pseudocounts*. That is, the frequency of each k-mer is increased by 1, regardless whether it occurs in the sequences or not.

   (b) *(6 pts)* To determine whether a certain k-mer is over- or under-represented one needs to determine the background frequencies of individual nucleotides. If you solved the previous exercise properly, the background frequencies for a set of sequences `seqs` can be obtained form the call `kmer_frequencies(seqs,1)`.

   Similar to the calculations of the BLOSUM scoring matrix odds-ratios can be calculated for the k-mers.

   observed frequency The observed relative frequency (as calculated in (a)) of a k-mer $a = a_1 a_2 \dots a_k$ where each of the a_i is one of the nucleotides A,C,G, or T is

$$o_s = \frac{f_a}{\sum_{\text{k-mer}x} f_x}$$

[1] PROMPTs are RNAs that are produced upstream of the promoters of active protein-coding genes.

where f_x represents the absolute frequency of k-mer x and the sum is taken over all k-mers.

expected frequency The expected frequency of a k-mer $a = a_1 a_2 \dots a_k$ is

$$e_a = p_{a_1} p_{a_2} \dots p_{a_k}$$

where $p_{a_i} = p_A, p_C, p_G, \text{or} p_T$ are the observed frequencies of the nucleotides A,C,G,T, respectively.

log odds ratio The odds-ratio of a k-mer a is

$$r_a = \frac{o_a}{e_a}$$

It is greater than 1 if the k-mer occurs more frequently in the sequences than expected by chance and less than 1 if it occurs less than expected. Taking the logarithm yields the log-odds ratio

$$s_a = \log_2 r_a$$

Write a function `log_odds_ratio(observed,background)` that takes as argument a dictionary of observed frequencies of k-mers and a dictionary containing the background frequencies of the individual nucleotides and calculates the log-odds ratio.

(c) *(3 pts)* For the given data `exosome-sensitive.fa` and `exosome-insensitive.fa` calculate

i. The background frequencies of the nucleotides using the combined set of sequences.

ii. The observed frequencies of k-mers for a) the sensitive sequences and b) the insensitive sequences

iii. Using the common background frequency calculate the log-odds ratio for the set of sensitive and insensitve sequences

Perform these calculations for $k = 2, 3,$ and 4.

(d) *(3 pts)* To discriminate k-mer composition of sensitive and insensitive RNA sequences you could take either

- the difference or
- the ratio

of the log-odds score of sensitive and insensitive sequences for each k-mer. However, only one of the options is correct! Write a small function that determines the correct discriminatory values for each k-mer. Perform these calculations for $k = 2, 3,$ and 4.

(e) *(3 pts)* Sort the k-mers from the lowest to the highest discriminatory values. What are the 16 kmers with highest/lowest scores? Plot the discriminatory values from lowest to highest for all k-mers using matplotlib or a Python plotting library of your choice. The x-axis contains the k-mers in increasing order of their discriminatory value and the y axis is the value. Perform these calculations for $k = 2, 3,$ and 4.

2. *(Optional 6 pts) Building machine learning models*
K-mer frequencies can be used as (simple) features for training machine learning models to distinguish sensitive and insensitive transcripts. To this end a sequence is represented by a feature vector of kmer counts. E.g. There are $4 \cdot 4 = 16$ 2-mers and the sequence

`GATGCGTAATGGATTCGATGCATGCGGCGTGAACTAGTCTAACG`

has the following 2-mer feature vector

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
3	2	1	5	1	0	5	2	4	4	2	3	3	2	5	1

Write code to generate kmer frequency feature vectors for all transcripts.

(a) From the transcripts randomly select a set containing 80% of the data: 400 sensitive and insensitive transcripts each. This set will form the training data. The remaining 20% of the transcripts form the test set

(b) Using the training set, train Support Vector Machines with a linear kernel using cross validation for parameter C selecting C from `[10**x for x in numpy.linspace(-2,2,9)]`, i.e. `[0.01,0.0316,0.1,0.316,1,3.16,10,31.6,100]` for $k = 2, 3$, and 4.

(c) Determine accuracy, precision and recall for the test set for the 3 classifiers corresponding to $k = 2, 3$, and 4.

(d) Repeat (b)-(c) for a Random Forest classifier. Here, you do not need to use cross validation. Instead simply use the default parameters.

Notes:

- Normally one would repeatedly generate different training and test sets according to (a) and determine average performance results (including their variance) to assess the dependence of the performance on the training/test set selection.
- For the models generated, *feature importances* can be assessed. That is, those features that are most relevant for distinguishing sensitive and insensitive transcripts can be determined on a model by model basis. E.g. in scikit-learn the random forest classifiers possess the attribute `feature_importances_` and SVMs (for linear kernels) possess the attribute `coef_` to access the feature importances. Such feature importances can form the basis of further analysis.