CHARITÉ / HUMBOLDT UNIVERSITÄT ZU BERL
INSTITUTE FOR THEORETICAL BIOLOGY

Prof. Hanspeter Herzel
Dr. Grigory Bordyugov
Institute for Theoretical Biology
D-10115 Berlin

h.herzel@biologie.hu-berlin.de
grigory.bordyugov@hu-berlin.de
Tel: +49 30 2903 9101 / 9121
Fax: +49 30 2903 8801

MODULE IV - BIOINFORMATICS: ASSIGNMENT 1

Please return your solution to G Bordyugov (office #201 in House 4, Charité Campus) in hard copy until April 7th, 3pm.

0. Installing R

Install RSTudio on your laptop and/or home desktop. Find out how to enter a program, how to save it in a file on your hard disk and restore from the previously saved file. Write a function that prints `Hello world!`.

1. R basics

1. Try it in an R session and describe what happens: If you add a number and a string? If you combine (using the `c()` function) a number and a string? If you try to calculate the square root of a negative number?

2. Try to execute the three following R snippets and explain why they generate errors:

```
# Example 1
add <- function(x,y) {
  mysum <- x+y
  return(mysum)
}
x <- add(3, 4)
print(x)
print(mysum)
```

```
# Example 2
x <- "1"
y <- "2"
z <- x + y

# Example 3
nucleotides <- c("C", "A")
if (nucleotides[2] == A) {
  print("we've got an A!")
}
```

3. Write a function `mysqrt(x)` that computes and returns the square root of the number `x` if `x` is non-negative and the square root of `-x` otherwise. Can you implement it without using an `if` statement? Can you think of more than one way to do so?

4. Generate a vector of N equidistant points on the interval $[0; 2\pi]$ and compute the values of the function $f(x) = \sin^2 x + \cos^2 x$ for `x` from the generated vector. Implement both a version using a `for` statement and a version using the `sapply()` function. Double check that the result corresponds to your mathematical intuition about what should have come out. Compare the performance (i.e. how fast the computations were) of the version with `sapply()` against the performance of the version with the `for` statement for N large enough (millions or tens of millions).

5. Implement a function `longest(strings)` that accepts a vector `strings` of strings and returns the longest string in `strings`. What string does the function return if there are two distinct strings with the largest length? *Hints:* The built-in function `nchar()` may be helpful here. Do a little research on how to find the largest element of a vector in R.

6. *Bonus problem:* Implement a function `secondmax(ns)` that finds and returns the second-to-largest element in the vector `ns` of numbers.

2. Flipping coins in R

1. Write a function `flip(p)` that generates and returns a result of a single random flip of a loaded coin with probability of a heads flip given by `p`. The function should return either “heads” or “tails” as a string.

2. Write a function `flips(N,p)` that generates and returns a sequence of `N` random flips (each being either “heads” or “tails”) of a loaded coin, given `N`, the number of flips, and `p`, the probability of a heads in a single flip. Think about how to use the previously written function `flip(p)` in order to implement the function `flips(N,p)`.

3. Write a function `seqprob(s,p)` that computes and returns the likelihood of observing the sequence `s` of coin flips in an experiment with a loaded coin with the heads probability `p`. Make sure that you can feed the result of the function `flips(N,p)` directly into the function `seqprob(s,p)` and get a correct answer. Does the function `seqprob(s,p)` behave correctly as `s` becomes larger? Try it with `s` of length 100, 1000, 10000, 100000, 1000000. How would you remedy your function?

4. For three different values of p of your choice, generate sequences of 10, 100, 1000, and 10000 coin flips. Then, by performing the χ^2-test on each of the sequences, test the hypothesis that the probability of a heads is equal to the one you chose.

3. Sequence statistics

1. Consider a plasmid with 10000 base pairs and assume no sequential correlation of nucleotide probabilities. One probability is known: $p_T = 0.2$. Determine the probabilities p_A, p_C, and p_G. How many restrictions sites `AACGTT` and `CGTACG` can be expected in the given plasmid on average? What is the standard deviation of the number of restriction sites? Calculate the probability that no site `CGTACG` is found.

2. Let us assume $p_i = 1/4$ independence of nucleotides and subsequent triplets. What is the probability to observe for $N = 100$ triplets $k = 0, 1, 2, \ldots, 6$ stop codons (`TGA`, `TAG`, `TAA`)? How many random ORFs with 100 codons can be expected in a mammalian genome?

4. Positional weight matrix

An example position weight matrix is given at `http://itb.biologie.hu-berlin.de/~bordyugov/tut/mmm2016/table.txt`

1. Calculate the Position Count Matrix (PCM) from the example data.
2. Calculate the position weight matrix (PWM) with elements W_{ij} derived from the PCM assuming $p_T = 0.3$.

   Hint: *Before* calculating the PCM, add 1 pseudo-count in each position of the PCM to avoid $\log_2 0$ (please also explain why should this be avoided?). Do not forget to modify N accordingly.
3. What does a positive W_{ij} mean for a letter i at position j in the matrix? How the weight of a new binding site might be interpreted biologically?
4. Suggest a possible consensus sequence.
5. Calculate the scores for the following putative binding sites:
   (a) `GTGGATT`
   (b) `GATGAAG`
   (c) `GGTGGAA`

   What cut-off value would you stick to and why? Which of the three sequences would be therefore predicted as a potential new binding site?
6. Suggest a hypothetical high-score binding site.
7. Create a sequence logo for the weight matrix. Use the WebLogo `http://weblogo.threeplusone.com/`
8. *Bonus problem:* Implement all steps above in R in a way that the your implementation can work on alignment tables with thousands or even millions of alignments.

5. A little bit of analysis

1. Determine the zeroes, extrema and asymptotic behaviour for the function:

$$f(x) = xe^{-x}.$$

   Draw the function in the interval $(0, 3)$.

2. What are the zeroes of the function:

$$g(t) = 4\cos^2(2t).$$

Draw the function in the interval $(0, 6\pi)$.