Computational Biology I: Quantitative Data Analysis
Frühjahrsemester 2019

Exercise 2

Pascal Grobecker and Athos Fiori

General Remarks

- Pay attention to the Input/Output variable type.
- Do not modify file name and function name.
- If you have questions write to qbiocomp1-bioz@unibas.ch

Working with numerical python

Python has a specialized module for numerical computations called Numpy (http://www.numpy.org/) and it's usually imported with the following comand

```
import numpy as np
```

Numpy can represent vectors and matrices by the array data type. Arrays allow you to do mathematical computations conviniently.

```
In [1]: x = np.arange(10) # array with numbers from 0 to 9
In [2]: x + 3
Out[2]: array([ 3,  4,  5,  6,  7,  8,  9, 10, 11, 12])
```

Thus, in the example above, you don't need to iterate over every element of the array in a for-loop to add the number 3. Instead the `+`-operator is defined *element-wise* for an array object. To train a bit yourself you can, as a first exercise, find out which other operators (like `-`, `*`, etc.) work element-wise on arrays. When can you add up two arrays? You might want to consult the Numpy reference manual (http://docs.scipy.org/doc/numpy/reference/) for that. Simple array constructors are:

`np.array([[1,2,3],[4,5,6]],dtype=float)`	2-by-3 matrix with floats 1.0 to 6.0
`np.zeros(n,m)`	n-by m matrix of zeros
`np.ones(n,m)`	n-by-m matrix of ones
`np.arange(start,end,step)`	array of evenly space numbers in the given interval

Besides, other mathematical functions like `sum`, `cumsum`, `min`, `max`, etc. can be directly applied to arrays. Indexing of arrays is very similar to indexing of lists, e.g., `x[0]` is first element of array `x`, `Y[2,3]` is the element of the 3rd row and 4th column of matrix `Y`. Additionally, you can define array slices by the ':' operator.

`x[0:4]`	elements 1 to 4 in `x`
`Y[:,0]`	the first column of matrix `Y`
`Y[3,2:4]`	the 3rd till 4th element of the 4th row of matrix `Y`

Sometimes you are interested in those elements of an array that match a certain condition. In this case logical indexing is the method of choice.

```
In [1]: x>3
Out[1]: array([False, False, False, False,  True,  True,  True,  True,  True,  True], dtype=bool)

In [2]: x[x>3]
Out[2]: array([4, 5, 6, 7, 8, 9])
```

Homework

We strongly encourage you to solve the exercises in order. In fact, many functions that you are going to implement are needed in questions that follow them. Before implementing a new function **think** about the functions you already implemented and how (if it's possible) can you use them in the current task. Read carefully the "Working with numerical python" paragraph since it can save you a lot of time.

1. *Log binomial coefficient*
   In *Exercise 1* you worked out how to implement the factorial using a loop. An alternative way can be implemented using the property of gamma function:

   $$n! = \Gamma(n+1) \quad \text{for} \quad n \in \mathbf{N}$$

   Using the log-gamma function provided by Scipy we can easily implement the log-factorial as

   ```
   from scipy.special import gammaln

   def log_factorial(n):
       return(gammaln(n+1.0)) # return the natural log of n!
   ```

   We already provide you this function and we suggest you to use it. Your task is now to

   - Implement the function `log_binomial_coeff` which returns the logarithm of the binomial coefficient $\left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$.
   - Implement the function `log_hypergeometricPMF` which returns the logarithm of the Hypergeometric Probability Mass Function $\dfrac{\left(\begin{array}{c} K \\ k \end{array} \right)\left(\begin{array}{c} N-K \\ n-k \end{array} \right)}{\left(\begin{array}{c} N \\ n \end{array} \right)}$.

   NB: For both functions use the properties $\log(ab) = \log a + \log b$ *and* $\log(\frac{a}{b}) = \log a - \log b$ *in order to reduce the computational complexity. By "default" we always consider the natural logarithm.*

2. *Descriptive statistics.*

   (a) The cumulative distribution function (CDF) for a discrete probability distribution (PMF) with parameter θ is defined as

   $$\text{CDF}(x|\theta) = \text{PMF}(k \leq x|\theta) = \sum_{k=0}^{x} \text{PMF}(k|\theta)$$

   Implement the CDF of the hypergeometric distribution in function `hypergeometricCDF`.

   (b) The q-th percentile is defined as x such that $\min_x CDF(x) \geq q$. The function `hypergeometricPPF` calculates the q-th percentile of the hypergeometric distribution with given parameters.

   (c) The interquartile range (IQR) is defined as the difference between the upper and lower quartiles. It is a measure of dispersion like the standard deviation. The function `hypergeometricIQR` calculates the IQR of the hypergeometric distribution with given parameters.

3. *Comparing Hypergeometric and Binomial distributions.* As was argued in the lectures, the larger the 'pool' of objects we are sampling from, the closer the hypergeometric distribution comes to a binomial distribution. Thus, we can use the approximation

   $$P(k|N,K,n) = \frac{\binom{N-K}{n-k}\binom{K}{k}}{\binom{N}{n}} \approx \binom{n}{k}\left(\frac{K}{N}\right)^k \left(1-\frac{K}{N}\right)^{n-k}.$$

   (a) The function `approximateHypergeometric` returns $P(k), 0 \leq k \leq n$ under the hypergeometric distribution and the corresponding approximation by the binomial distribution.

* Optional : Compare your results by plotting both PMFs for different parameter combinations. We already provide you `plot_cellCulture1`.

4. *Cell culture problem.* We have a set of $N = 1000$ cell cultures of which an unknown number M express a certain protein. We are interested in estimating the number M of cultures that express this protein. However, the only way that we can check whether the cells within a given cell culture express the protein is to stain the cells, and this procedure kills the cells in the culture. Thus, the more cultures we test, the less living cultures we have left for doing other experiments with. We thus want to test as few cultures as possible but enough to reach a certain level of accuracy in our estimate. In this exercise we will work out how many of our cultures we need to test.

   (a) First we want to derive the posterior probability distribution of M. Assume we sample $n = 10$ cultures and obtain $m = 3$ that express the protein. We know the likelihood of the data we observed is $P(D|I) \sim \text{Hypergeometric}(m = 3|N = 1000, M, n = 10)$. Assuming a uniform prior probability for any possible value of M, i.e. $\forall x : P(x = M) = \text{constant}$, and applying Bayes formula we can derive the posterior of $x = M$ as:

$$P(x = M|N = 1000, m = 3, n = 10) = \frac{P(m = 3|N = 1000, x = M, n = 10)}{\sum_{x=3}^{993} P(m = 3|N = 1000, x, n = 10)} \tag{1}$$

   where we have made use of our current knowledge that $3 \leq x \leq 993$, i.e., we already sampled 10 cultures of which 3 express the protein and 7 don't. The function `cellCulture1` calculates the posterior distribution of M given $N = 1000$, n and m. Tip: get a feeling for the results by plotting P for different values of n and m `plot_cellCulture1` .

   (b) The function `cellCulture2` extends the previous example by calculating a minimal value $M_{\min}$ such that $P(M < M_{\min}|N = 1000, n, m) > 0.025$ and a maximum value $M_{\max}$ such that $P(M > M_{\max}|N = 1000, n, m) \leq 0.975$. That is, it returns the lower and upper limits of the 95% posterior probability interval of M.
   Hint: In (2.) you did something similar for the likelihood, here you should do it for the posterior.

   (c) We can define the relative error e in M, associated with this interval as the size of the interval relative to the middle of the interval, i.e.

$$e = \frac{M_{\max} - M_{\min}}{0.5(M_{\min} + M_{\max})}. \tag{2}$$

   The function `cellCulture3` calculates this error as a function of $N = 1000$, n and m.

   (d) Let's assume that we start testing cultures one by one and calculate e after each experiment. We will stop doing experiments when $e < 0.1$, that is when the relative error in the data-set is less than a threshold of 10%. Assume for simplicity that all our data-sets always have the form[1] $m = n/2$ that express the protein, i.e. we imagine data-sets such that either $n = 2, m = 1$ or $n = 4, m = 2$, or $n = 10, m = 5$, or $n = 100, m = 50$, etcetera. The function `cellCulture4` calulates the minimal number of experiments we have to do in order to achieve a relative error smaller than a given threshold ϵ.

Theoretical questions (optional)

- Could you intuitively explain why the hypergeometric distribution has the form

$$P(k|N, K, n) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

  Does it represents "drawing with replacement" or "drawing without replacement"? What about the binomial distribution?

[1] Clearly it's very unlikely that in reality we obtain all data-sets of this form.

- We have a box with 10 balls of which 7 are red and 3 are blue. We now will draw 2 balls from this box. Without looking , and without replacement. First, we want to know what is the probability P that the second ball we draw is blue[2]? And second, assuming that the second ball is indeed blue, what is the probability P' that the first ball was red?
- How likely is, in Bridge, that one player misses out completely one color? Let's look at a specific example. In Bridge the 52 cards are randomly distributed over 4 players. Imagine you are one of the 4. What is the probability P that you end up with a hand (i.e. random selection of 13 cards) in which there are no hearts at all?
- What's the mean $\bar{x}$ of the data set $D = (x_1, x_2, x_3, x_4, x_5, x_6) = (2, 2, 4, 2, 8, 6)$?
- What's the definition of the interquartile range $[x_{min}, x_{max}]$ of a continuous distribution $P(x)$?
- Assume the continuous quantity x follows a uniform distribution over the range $x_{min} \leq x \leq x_{max}$. We are interested in the quantiles of this distribution. What's the interquantile range $[x_q, x_{1-q}]$? That is, for $q = 1/4$ this would correspond to the *interquartile* range $[x_{0.25}, x_{0.75}]$, and for $q = 0.005$ to the range from the 5th to the 95th percentile $[x_{0.05}, x_{0.95}]$. Write an expression for $[x_q, x_{1-q}]$ in terms of $q,\ x_{min},\ x_{max}$.

How to submit the exercise

Submit your completed `Exercise2.py` file by email to biocomp1-bioz@unibas.ch. The submission deadline is Thursday, midnight.

1. Check that your functions are working properly before submitting them!
2. Don't forget to attach your exercise file!
3. Don't change the name of the file or the name of any function within the file!
4. Please send your solutions from the same email address that you used to register for the course.

[2]This question is easy if you think probability in terms of information. Do you think that we have additional information, more than the prior information, in knowing that it's the second ball that we are drawing? Nota Bene: This is a good example of distiction between physical causationa and logical implication.