Computational Biology I: Quantitative Data Analysis
Frühjahrsemester 2019

Exercise 6

To Submit the exercise

Submit your completed `exercise6.py` file by email to `biocomp1-bioz@unibas.ch`. The submission deadline is Thursday, April 12th, 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 registering at the course.

Mixture-modelling of ChIP-Seq data

Chromatin immunoprecipitation followed by sequencing (ChIP-Seq) is a method used to localise binding sites of DNA-associated proteins. It is most often used to either identify genome-wide *in vivo* transcription factor binding sites, or to reveal condition-specific chromatin modifications, e.g., the tri-methylation of histone H3 on lysine residue K4 (in short, H3K4me3), which is associated with active promoter regions. The protocol is explained in the following diagram:

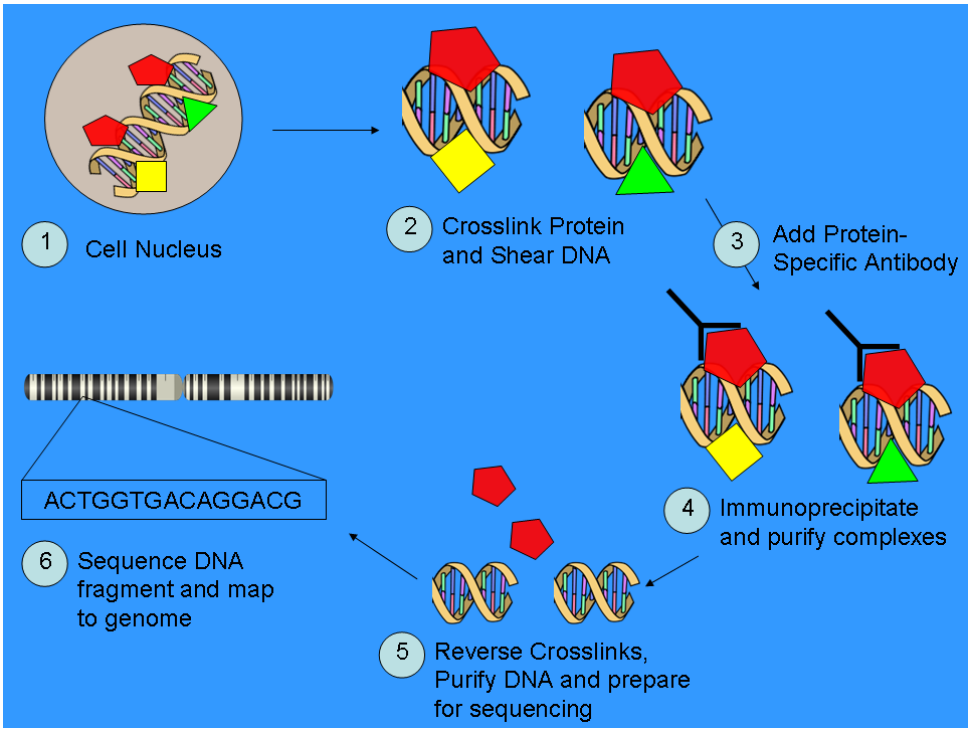

Figure 1: Basic steps of the chromatin immunoprecipitation followed by sequencing (ChIP-Seq) protocol.

1. In the following we will analyse a ChiP-Seq data set which was generated with an antibody against histones H3 that are acetylated at the lysine at position 36 (H3K36ac). It is known that this acetylation mark is present whenever a gene is actively transcribed. The data set you are dealing with summarizes the data by giving the number of reads which are mapped to different genomic regions. Every region is characterized by its size and an integer number of reads that map to it. For example, the first rows of the data file look like this:

```
chr1    121542  121582  region1 200
chr1    121555  121582  region2 9666
chr1    121581  121662  region3 14823
...
```

   Each line gives the location of the region (chromosome, start, end), a name, and the number of sequencing reads mapping to this region. This type of file format is called BED format. It is a common format to store genomic information (see https://genome.ucsc.edu/FAQ/FAQformat.html). Using the function `loadData`, load the region lengths and read counts from the `H3K36ac.bed` file. Try to understand how the function is working. Plot the region lengths versus the number of counts. What do you observe? You might find useful to plot the counts on a log-scale (`plt.semilogy(length,count); plt.show()`.) (answers not to be submitted).

2. Complete the function `transformData` which transforms the data according to the formula:

$$x_i = \log\left(c_i/s_i + p\right)$$

   where c_i is the number of reads in region i, s_i is the region size and p is a pseudo count which we will set to 0.5.

3. Ideally speaking the chromatin immuno-precipitation would only extract genomic regions where H3K36ac nucleosomes were bound, but in reality the anti-bodies have only finite specificity. That is, although regions bound by H3K36ac are selected much more often than those that do not, sequence segments without H3K36ac are still selected (and sequenced) at a certain rate. Thus, the reads we observe across the genome are a mixture of 'foreground' reads from sequence segments truly containing H3K36ac and 'background' reads from segments that did not. In the following we want to use mixture modeling to separate the foreground from the backgound signal. Plot a histogram of the transformed data values (`plt.hist(x,100); plt.show()`) and try to answer the following questions. How does the distribution look like? Why is there a discrete pattern at the lower end of the histogram? What type of mixture model might describe the data? (answers not to be submitted)

4. From the shape of the histogram we might assume that the background is exponentially distributed and the foreground is Gaussian distributed. The likelihood function of the corresponding Gaussian-Exponential mixture model reads:

$$P(D|\rho,\mu,\sigma,\lambda) = \prod_i \left(\rho \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}} + (1-\rho)\lambda e^{-\lambda x_i}\right),$$

   where x_i is the transformed read-count signal in region i. Parameter ρ is the fraction of foreground signal within the data. Parameters (μ,σ) are the mean and standard deviation of the foreground signal, and the parameter λ defines the mean and spread of the exponentially distributed background signal. As already encountered in the previous exercises, it is mathematically more convenient to analyse the log-likelihood function.

$$L(D|\rho,\mu,\sigma,\lambda) = \sum_i \log\left[\rho L_1(x_i|\mu,\sigma) + (1-\rho)L_2(x_i|\lambda)\right], \tag{1}$$

   where,

$$L_1(x_i|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}}, \quad L_2(x_i|\lambda) = \lambda e^{-\lambda x_i}.$$

Complete the function `mixture_logLikelihood` which calculates the log-likelihood of the data (1) given parameters $\rho, \mu, \sigma, \lambda$. Make use of the predefined exponential and Gaussian probability density functions! For implementation purposes note that `lambda` is a reserved keyword in python, so you `cannot` use it as a variable name! Instead, you could use `lam`.

5. The log-likelihood is a function of the unknown parameters $(\rho, \mu, \sigma, \lambda)$. In the following we will derive equations for maximizing the log-likelihood in terms of the unknown parameters. At its maximum, the log-likelihood must simultaneously satisfy the following equations (one for each parameter):

$$\frac{\partial L(D|\rho,\mu,\sigma,\lambda)}{\partial \rho} \stackrel{!}{=} 0, \quad \frac{\partial L(D|\rho,\mu,\sigma,\lambda)}{\partial \mu} \stackrel{!}{=} 0, \quad \frac{\partial L(D|\rho,\mu,\sigma,\lambda)}{\partial \sigma} \stackrel{!}{=} 0, \quad \frac{\partial L(D|\rho,\mu,\sigma,\lambda)}{\partial \lambda} \stackrel{!}{=} 0.$$

These conditions lead to equations which define ρ, μ, σ and λ at the optimum. For example, the condition for μ reads:

$$\begin{aligned}
& \frac{\partial L(D|\rho,\mu,\sigma,\lambda)}{\partial \mu} \stackrel{!}{=} 0 \\
\Leftrightarrow \quad & \sum_i \frac{\rho \frac{\partial L_1(x_i|\mu,\sigma)}{\partial \mu}}{\rho L_1(x_i|\mu,\sigma) + (1-\rho) L_2(x_i|\lambda)} = 0 \\
\Leftrightarrow \quad & \sum_i \frac{-\rho \frac{(x_i-\mu)}{\sigma^2} L_1(x_i|\mu,\sigma)}{\rho L_1(x_i|\mu,\sigma) + (1-\rho) L_2(x_i|\lambda)} = 0 \\
\Leftrightarrow \quad & \sum_i \frac{\rho x_i L_1(x_i|\mu,\sigma)}{\rho L_1(x_i|\mu,\sigma) + (1-\rho) L_2(x_i|\lambda)} = \mu \sum_i \frac{\rho L_1(x_i|\mu,\sigma)}{\rho L_1(x_i|\mu,\sigma) + (1-\rho) L_2(x_i|\lambda)}.
\end{aligned}$$

Defining the probability that x_i comes from the foreground model as,

$$p_{1,i} = \frac{\rho L_1(x_i|\mu,\sigma)}{\rho L_1(x_i|\mu,\sigma) + (1-\rho) L_2(x_i|\lambda)},$$

gives the final equation for μ at the likelihood optimum:

$$\mu = \frac{\sum_i x_i p_{1,i}}{\sum_i p_{1,i}}. \tag{2}$$

In a similar fashion the following equations for ρ, σ and λ at the likelihood optimum hold:

$$\rho = \frac{\sum_i p_{1,i}}{n} \tag{3}$$

$$\sigma^2 = \frac{\sum_i (x_i - \mu)^2 p_{1,i}}{\sum_i p_{1,i}} \tag{4}$$

$$\lambda = \frac{\sum_i p_{2,i}}{\sum_i x_i p_{2,i}} \tag{5}$$

$$p_{2,i} = \frac{(1-\rho) L_2(x_i|\lambda)}{\rho L_1(x_i|\mu,\sigma) + (1-\rho) L_2(x_i|\lambda)}. \tag{6}$$

Try to derive these equations yourself (theoretical exercise). The only way to become confident with these methods is to practice deriving these kinds of expressions.

6. The equations for ρ, μ, σ and λ at the likelihood optimum are all implicit. That is, in 2, the likelihood L_1 of every $p_{1,i}$ term depends again on μ (as well as all the other parameters). Therefore, the maximum likelihood equations don't provide a direct solution for the parameters at the likelihood optimum. However, we can solve these equations iteratively using a two-step procedure. In every step of the iteration the parameter values of the previous step are used to calculate the current likelihood (expectation step). We then maximize this likelihood to get updated parameter solutions (maximization step). Maximization of the likelihood with respect to $\rho, \mu, \sigma, \lambda$ leads to the previously derived update equations :

$$
\begin{aligned}
\rho^{(j+1)} &= \frac{\sum_i p_{1,i}^{(j)}}{n} \\
\mu^{(j+1)} &= \frac{\sum_i x_i p_{1,i}^{(j)}}{\sum_i p_{1,i}^{(j)}} \\
(\sigma^2)^{(j+1)} &= \frac{\sum_i (x_i - \mu^{(j)})^2 p_{1,i}^{(j)}}{\sum_i p_{1,i}^{(j)}} \\
\lambda^{(j+1)} &= \frac{\sum_i p_{2,i}^{(j)}}{\sum_i x_i p_{2,i}^{(j)}},
\end{aligned}
$$

where j refers to the j-th iteration and

$$
\begin{aligned}
p_{1,i}^{(j)} &= \frac{\rho^{(j)} L_1(x_i|\mu^{(j)}, (\sigma)^{(j)})}{\rho^{(j)} L_1(x_i|\mu^{(j)}, (\sigma)^{(j)}) + (1-\rho^{(j)}) L_2(x_i|\lambda^{(j)})} \\
p_{2,i}^{(j)} &= \frac{(1-\rho^{(j)}) L_2(x_i|\lambda^{(j)})}{\rho^{(j)} L_1(x_i|\mu^{(j)}, (\sigma)^{(j)}) + (1-\rho^{(j)}) L_2(x_i|\lambda^{(j)})},
\end{aligned}
$$

uses the likelihood calculated from the previous parameter estimates. In this way, the equations given above for ρ, μ, σ and λ which maximize the likelihood, can be used to update the parameters in each step of the iteration. Of course, $\rho^{(1)}$, $\mu^{(1)}$, $\sigma^{(1)}$ and $\lambda^{(1)}$ have to be initialized at the beginning of the algorithm. The iteration continues until the value of the likelihood converges, i.e. there is no significant change in the likelihood beetween update steps. Briefly, the expectation-maximization (EM) algorithm can be summarised as follows:

```
initialize parameters rho,mu,sigma and lambda
while not converged:
    # estimation step
    for every x[i] calculate likelihoods L1[i] and L2[i] based on current parameter estimates
    # maximization step
    calculate parameter updates using L1 and L2, i.e. solve update equations
    # check for convergence
    converged = (logLik[j] - logLik[j-1]) < threshold
```

Complete the function `EM` which implements the EM algorithm for the data model given in 1.

7. Estimate the parameters ρ, μ, σ and λ by the EM algorithm. The function `plotData` allows you to plot your solution together with the data histogram. How well does your model fit the data?

8. Implement function `probForeground` which calculates for each region the probability that it comes from the Gaussian foreground, given parameters ρ, μ, σ and λ. How many regions in the H3K36ac data set come from foreground, given your optimal parameters? Also plot the transformed data x versus the foreground probabilities (`plt.plot(x,p,'.'); plt.show()`). What do you observe for low and high values of x?

Theoretical exercises

1. Derive equations (3), (4), & (5).

2. The length of the cell-cycle fluctuates from cell to cell. The total length of a cell-cycle t is the sum of the length s of S phase, the length g_1 of G phase, and the length g_2 of $G2$ phase, which all three fluctuate (we will ignore the M phase for this calculation, which means we assume the length of M phase does not fluctuate). We will also assume that the length of these phases fluctuate *independently*.

   Assuming that we know the average lengths $\langle s \rangle$, $\langle g_1 \rangle$ and $\langle g_2 \rangle$, ans also the variances var(s), var(g_1), and var(g_2), what can we now say about the variance if the total cell-cycle length var(t)?

3. The central limit theorem shows that , under some fairly weak conditions, if the quantity X is the *sum* of a number of independent quantities x_i that each have the same distribution, *i.e.* $x = x_1 + x_2 + x_3 + ...$, then the distribution of x becomes a Gaussian':

$$P(x)dx = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx$$

   We now consider a quantity y that is the *product* of a number of quantity that each have the same distribution, *i.e.* $y = y_1 y_2 y_3 ....$ What is the distribution of the quantity y? As a hint, this distribution is known as a *log-normal* distribution. Also, when writing an expression for $P(y)dy$, do not forget the part coming from dy.