An algorithm for spatio-temporal clustering of genes along the chromosomes

EPFL Bachelor Project – Computer Science – Spring 2016

Dario Anongba Varela

Supervisors : Saeed Omidi, Felix Naef
Lab : Computational Systems Biology Lab (Felix Naef)

Goals of the research

One:

01 Genes are placed in a fixed order along the chromosomes. The role of this specific positioning, if any, **is yet to be understood**

Two:

02 Are genes placed in **clusters** along the chromosomes ?

Three:

03 If they are, would it suggest a **functional role or evolutionary advantage of the specific positioning** of genes along a chromosome

Partitioning algorithm

—

- How to identify the clusters and partition the genes efficiently ?

- From an O(2^{n-1}) problem to an O(N^2) using dynamic programming

- Example, if n = 1140.
  – Naïve approach : 7.46×10^{342} operations (not computable)
  – Dynamic approach : 1.3×10^6 operations

Partitioning algorithm

—

- A length **n** of a metal rod. Example for n = 4. So $2^{4-1} = 8$ possible solutions
- A table of prices p_i for rods of lengths i = 1,…,n

Partitioning algorithm

—

- A length **n** of a metal rod. Example for n = 4. So 2^3 = 8 possible solutions
- A table of prices p_i for rods of lengths i = 1,…,n

| length i	1	2	3	4
price p_i	1	5	8	9

Partitioning algorithm

—

- A length **n** of a metal rod. Example for n = 4. So 2^3 = 8 possible solutions
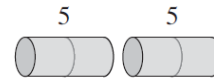- A table of prices p_i for rods of lengths i = 1,…,n

length i	1	2	3	4
price p_i	1	5	8	9

- **Objective** : Decide how to cut the rod into pieces and maximize / minimize the price.

Partitioning algorithm

—

- A length **n** of a metal rod. Example for n = 4. So 2^3 = 8 possible solutions
- A table of prices p_i for rods of lengths i = 1,…,n

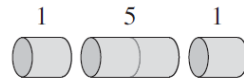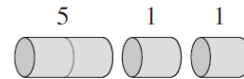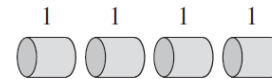| length i	1	2	3	4
price p_i	1	5	8	9
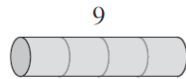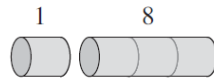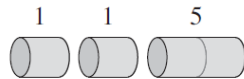
RNA-Seq data

- Reveals the presence and quantity of RNA in a sample at a given moment in time

- We can identify changes in gene expression and partition the genome according to the way genes fit different models.

RNA-Seq data : Sorted and filtered

- **Matrix** : Columns are genes and rows are time point measures (24 time points)

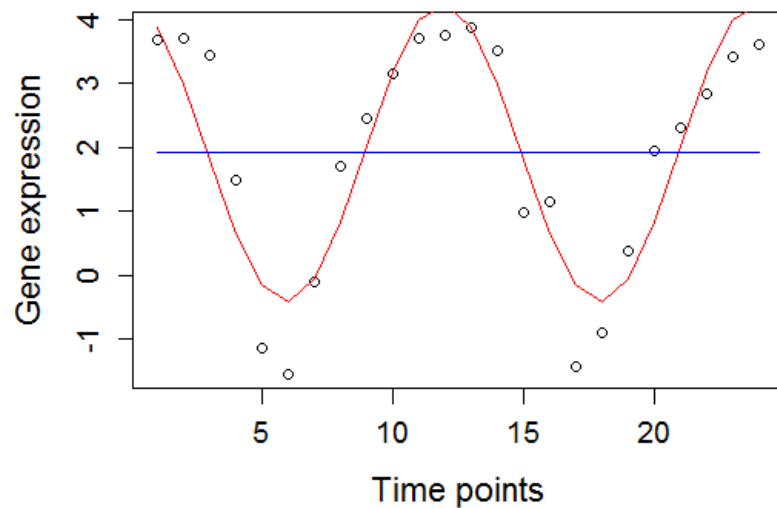|	AI838599	Gm11787	Tmem68	Tgs1	Lyn	Rps20	Plag1	Chchd7	Sdr16c5	Impad1
ZT_0_1_exon	-2.900021	-2.1866640	1.482505	1.703562	3.346512	4.028217	-2.595579	1.265885	-1.427024	3.306969
ZT_2_1_exon	-2.147666	-1.3974544	1.384402	1.742595	3.233279	3.980873	-2.763773	1.706167	-2.078519	3.251865
ZT_4_1_exon	-3.723140	-1.2730027	1.336116	1.667629	3.456569	3.989345	-3.129129	1.470803	-2.275023	3.472580
ZT_6_1_exon	-2.550986	-1.7673547	1.650827	1.780649	3.201119	4.058997	-2.771035	1.678928	-1.844185	3.514598
ZT_8_1_exon	-2.358189	-1.2414010	1.689454	1.760129	3.500712	4.044151	-3.068381	1.708180	-3.042665	3.672099
ZT_10_1_exon	-1.770339	-1.2246844	1.659832	1.981001	3.511658	4.023518	-2.739488	1.966097	-2.085763	3.673638
ZT_12_1_exon	-2.567874	-0.9173816	1.613619	1.890222	3.843681	4.233419	-2.757533	2.092891	-3.130181	3.546923
ZT_14_1_exon	-1.916705	-1.1619536	1.551675	1.805628	3.499936	4.274861	-2.667339	2.152064	-2.103284	3.307825
ZT_16_1_exon	-2.140111	-1.1703514	1.182650	2.001469	3.660696	4.125828	-2.618715	1.822264	-2.746438	3.135911
ZT_18_1_exon	-3.234543	-1.2538391	1.195542	2.058232	3.622647	4.072459	-2.653260	1.628464	-2.784652	3.213773
ZT_20_1_exon	-2.754052	-1.4439207	1.306815	1.805679	3.646546	4.243151	-2.923009	1.322040	-2.029528	3.169621
ZT_22_1_exon	-2.977634	-1.4700764	1.558112	2.019093	3.618602	4.155664	-3.234761	1.467683	-3.131982	3.492495
ZT_0_2_exon	-2.708496	-1.4014085	1.303440	1.664715	3.436299	3.989970	-2.495194	1.665081	-1.605206	3.257282
ZT_2_2_exon	-3.427011	-2.3605478	1.149422	1.506327	3.373362	4.069581	-2.603078	1.584595	-3.493889	3.348353
ZT_4_2_exon	-2.893104	-1.8023181	1.436354	1.783920	3.332191	3.917535	-2.553024	1.598997	-2.528754	3.407981
ZT_6_2_exon	-2.342655	-0.9122524	1.457620	1.688068	3.266939	4.170368	-2.976327	1.667599	-2.990295	3.560779
ZT_8_2_exon	-1.937925	-1.3376977	1.617867	1.947277	3.335588	3.929372	-2.795060	1.742269	-2.527433	3.558703
ZT_10_2_exon	-2.078174	-2.2166813	1.476735	2.012728	3.327091	3.937200	-2.689005	1.917529	-2.503246	3.540805
ZT_12_2_exon	-2.479944	-2.1153123	1.328514	1.898995	3.247362	4.045817	-2.718828	1.990023	-2.547354	3.362967
ZT_14_2_exon	-2.533102	-1.0033291	1.406451	2.006347	3.292311	4.148814	-2.533223	1.903101	-1.218718	3.278278
ZT_16_2_exon	-2.785418	-1.4341554	1.153439	1.907564	3.613436	4.171117	-2.812031	1.845097	-2.174865	3.133933
ZT_18_2_exon	-3.375303	-1.7007105	1.398176	1.990743	3.526876	4.121281	-2.868494	1.636407	-2.645774	3.332045
ZT_20_2_exon	-2.497778	-1.8758133	1.254387	1.808366	3.379053	3.899576	-2.515257	1.442967	-2.329322	3.286607
ZT_22_2_exon	-2.886415	-1.6533395	1.314752	1.975699	3.539485	3.966283	-2.675430	1.531419	-2.124065	3.221432

>

Models

- Two different models, a **flat model** and a **rythmic (circadian) model**

- The flat model is preferred for noisy genes, the circadian model for rythmic genes.

- Flat model : $\hat{Y} = temporal\ mean$

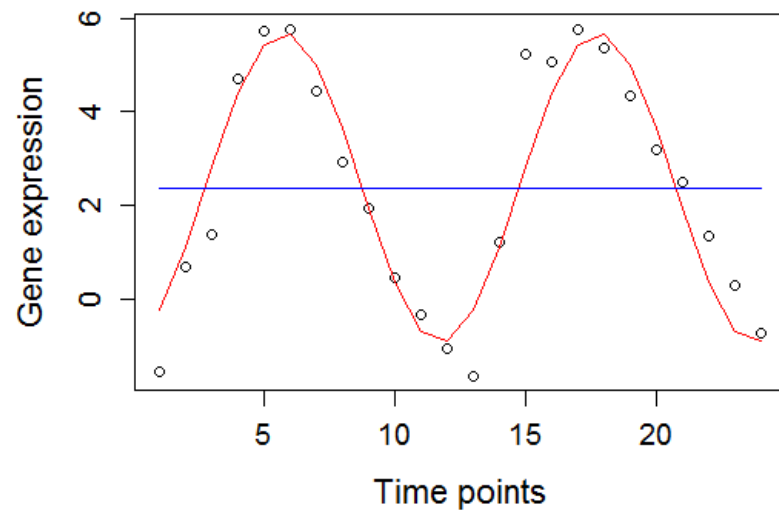- Circadian model : $\hat{Y} = \alpha \sin(2\pi f) + \beta \cos(2\pi f)$

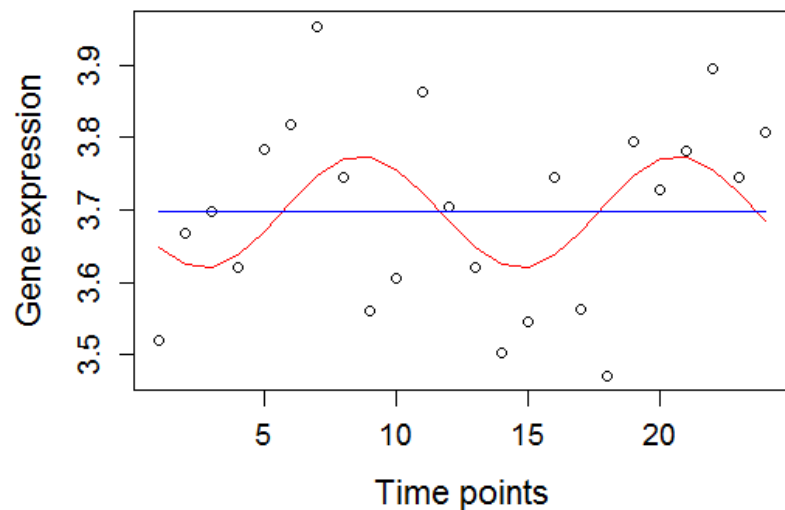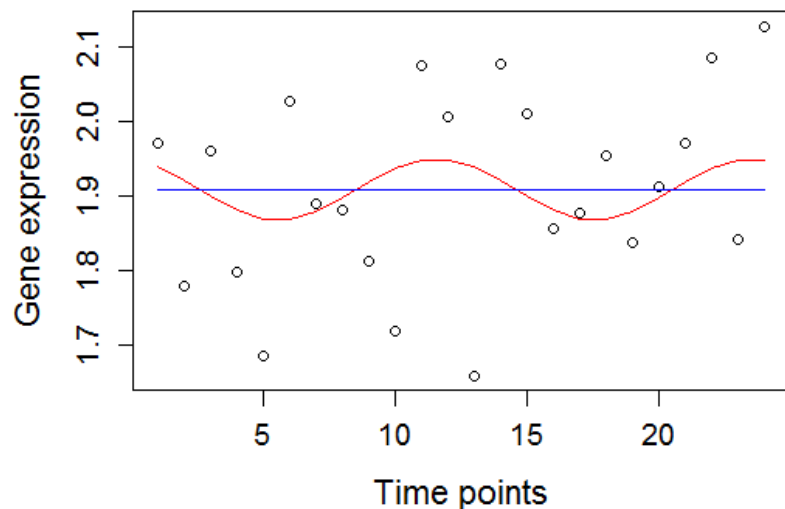# Gene expression and model fitting

Gene expression and model fitting

" Bax " gene expression across time

" Ercc2 " gene expression across time

Partitioning algorithm

—

We are actually doing quite the same than the rod cutting with our RNA-Seq data

- Where **n** is the number of genes of a chromosome we want to partition

- We try to minimize the score instead of maximizing it

What is the equivalent to the **prices** of the rod cutting algorithm ?

Model selection

- The «price» is computed for each block of genes by :
- $\min\{score\ of\ flat\ model,\ score\ of\ circadian\ model\}$

- Where the score is : the residue (square of the errors) + a penalty.

- Model 1 is the flat model and model 2 is a circadian (rythmic model) :

- **Residue** of a block : $\sum_{i=1}^{n}(M_i - \hat{Y}_{ji})^2$,
  for j = 1, 2 (for model 1 and 2), M is the RNA-Seq matrix and $\hat{Y}_j$ is the predictor matrix

Gene expression and model fitting for blocks of genes
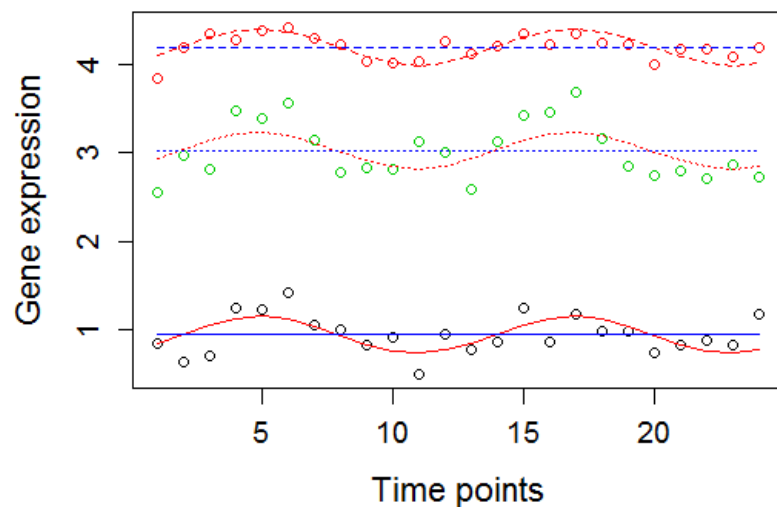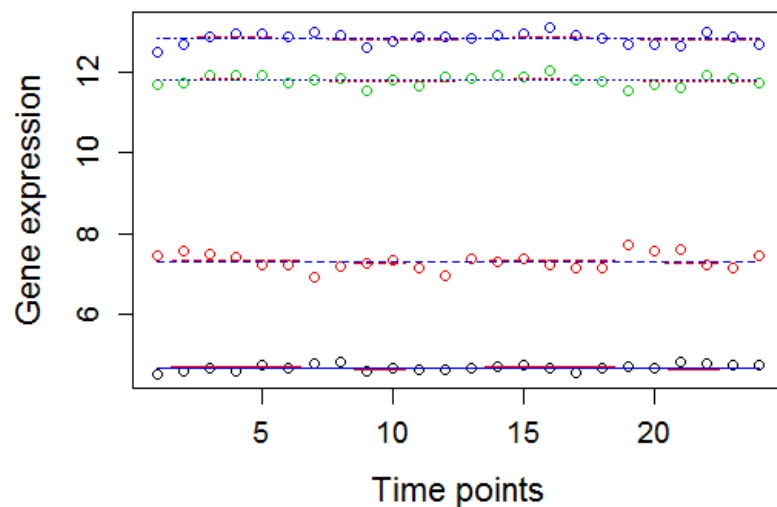
Model selection

—

Circadian model (residues) fits the data at least as bad as the flat model !

Fitting of the models, chr4
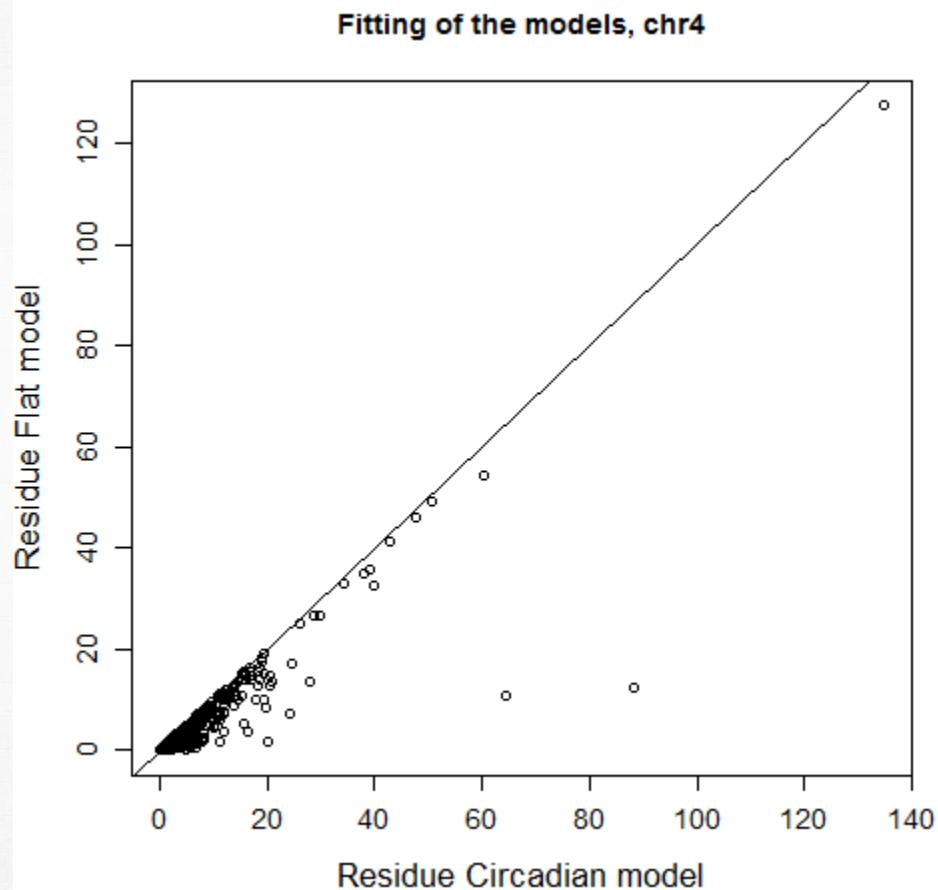
Model selection

- Have to introduce a penalty : Score = residue + penalty

- First attempt : Bayesian information criterion

- $BIC = \sigma^2 \times k \times \log(k * m)$,
  where **k** is the number of parameters and **m** is the number of data points inside the considered block

- Problem : Not additive

Model selection

- Have to introduce a penalty : Score = residue + penalty

- Second attempt

- Penalty = $\sigma^2 * (k + 1) * \log(N),$
  Where **k** is the number of parameters, **N** the total number of data points and σ a customizable value.

- Problem : Solves additive issue of BIC

Gene expression and model fitting

" Arntl " gene expression across time

" Dbp " gene expression across time
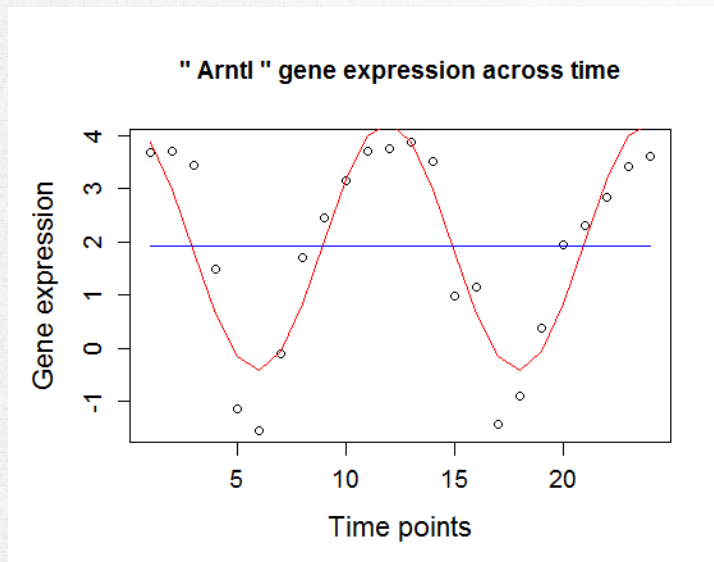
Flat score = 79 / Circadian score = 14.81
Flat score BIC = 78.17/ Circadian score BIC = 13.29
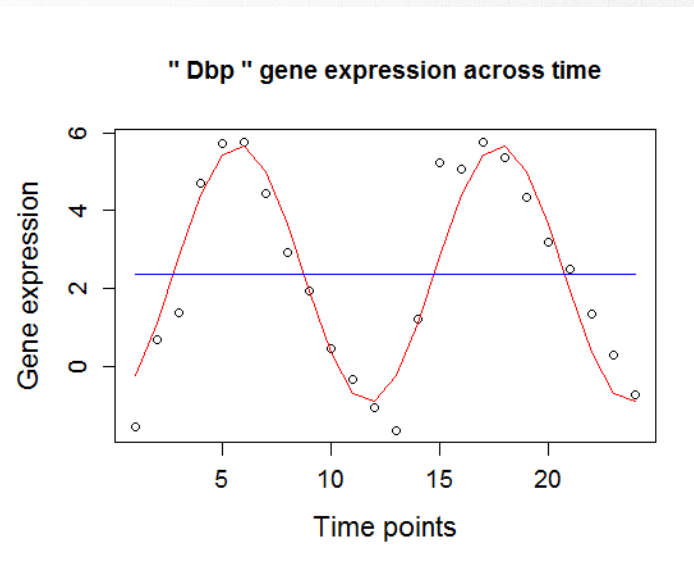
Flat score = 149.1 / Circadian score = 18.63
Flat score BIC = 149.9 / Circadian score BIC = 17.10

Gene expression and model fitting

Flat score = 1.39/ Circadian score = 2.31
Flat score BIC = 0.55/ Circadian score BIC = 0.79
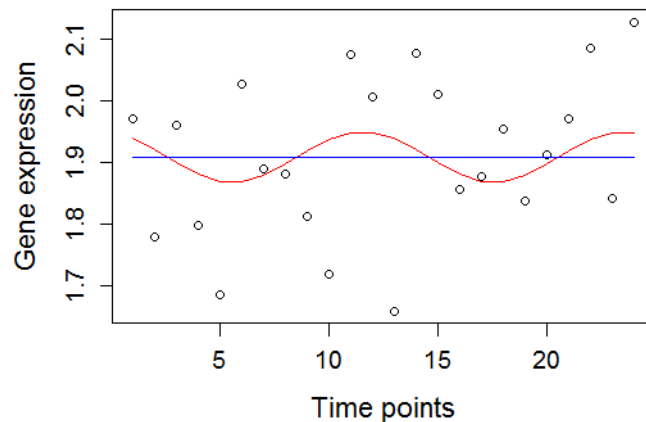
Flat score = 1.38 / Circadian score = 2.35
Flat score BIC = 0.54/ Circadian score BIC = 0.83

Gene expression and model fitting for blocks of genes
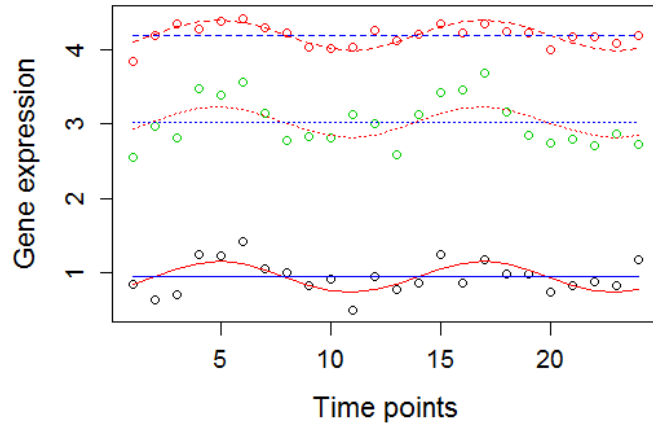
Flat score = 6.01 / Circadian score = 5.47
Flat score BIC = 4.65/ Circadian score BIC = 3.5
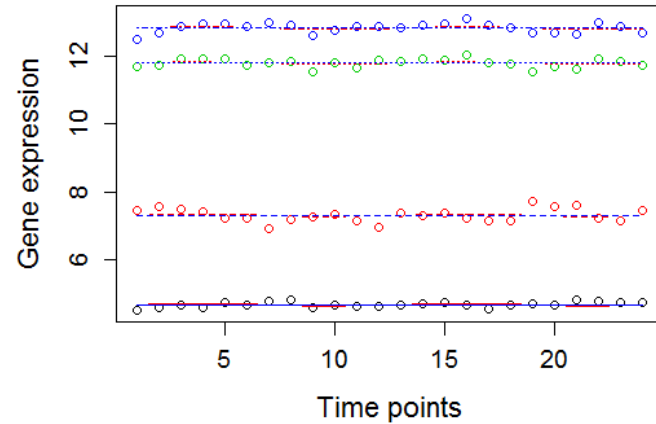
Flat score = 4.37 / Circadian score = 5.3
Flat score BIC = 2.7 / Circadian score BIC = 3.1

Model selection

- Instead of using an abstract sigma value, we transformed the definition of sigma to the **a priori percentage of the rythmic genes**

Model selection

- Instead of using an abstract sigma value, we transformed the definition of sigma to the **a priori percentage of the rythmic genes**

- In order for the circadian model to be chosen over the flat model we need :
- $res.\, 2 +\, \sigma^2(m + 2 + 1)\log(N) < res.\, 1 + \sigma^2(m + 1)\log(N)$

Model selection

- Instead of using an abstract sigma value, we transformed the definition of sigma to the **a priori percentage of the rythmic genes**

- In order for the circadian model to be chosen over the flat model we need :
- $res. 2 + \sigma^2(m + 2 + 1)\log(N) < res. 1 + \sigma^2(m + 1)\log(N)$, where m is the number of genes in the considered block

- Through simple calculations, we obtain :
- $res. 1 - res. 2 > 2\sigma^2\log(N)$

Model selection

- Instead of using an abstract sigma value, we transformed the definition of sigma to the **a priori percentage of the rythmic genes**

- In order for the circadian model to be chosen over the flat model we need :
- $res.\,2 + \sigma^2(m + 2 + 1)\log(N) < res.\,1 + \sigma^2(m + 1)\log(N)$

- Through simple calculations, we obtain :
- $cuttof > 2\sigma^2 \log(N)$

Model selection

- Instead of using an abstract sigma value, we transformed the definition of sigma to the **a priori percentage of the rythmic genes**

- In order for the circadian model to be chosen over the flat model we need :
- $res.\,2 + \sigma^2(m+2+1)\log(N) < res.\,1 + \sigma^2(m+1)\log(N)$

- Through simple calculations, we obtain :
- $cuttof > 2\sigma^2\log(N)$

- So we find the value of sigma squared as : $\sigma^2 = \dfrac{cuttof}{2\log(N)}$

Model selection

- Instead of using an abstract sigma value, we compute a value of sigma given the percentage of expected circadian genes

- We use the inverse cumulative function to find it :
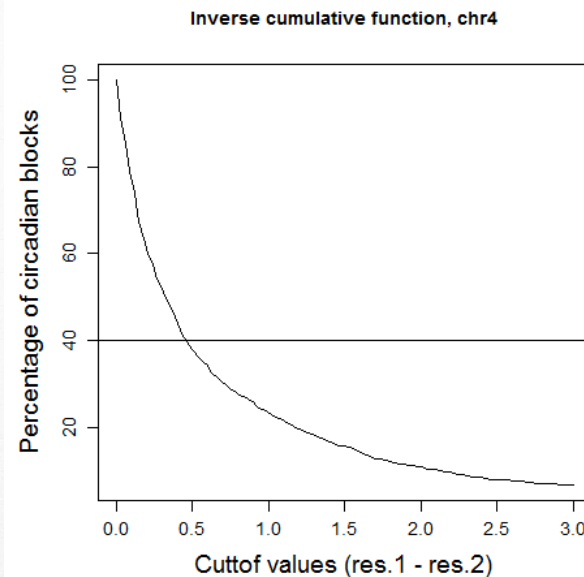
Model selection

- Instead of using an abstract sigma value, we compute a value of sigma given **the percentage of expected circadian genes**

- We use the inverse cumulative function to find it :

- **Cuttof for a percentage of 40%** = 0.4621029

- $\sigma^2 = \dfrac{cuttof}{2\ \log(N)} = 0.02289724$
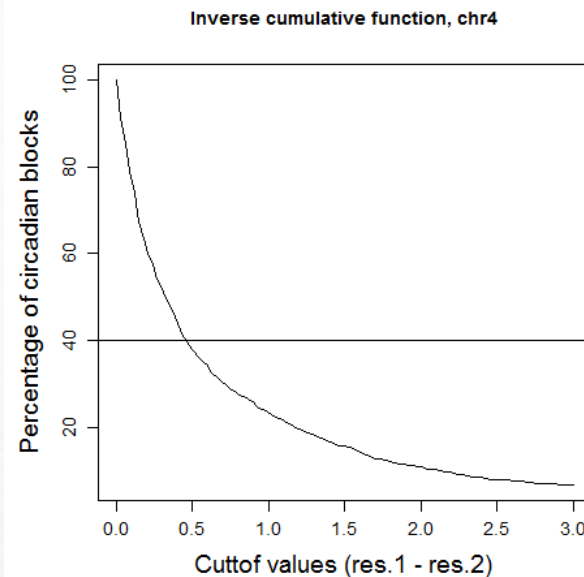
Inverse cumulative function, chr4

Pipeline of computation

Step 1   Input the RNA-Seq data and the parameters

Step 2   Compute the residue of models (this is stored)

Step 3   Compute the cuttof and σ^2

Step 4   Run the algorithm with the given and computed parameters
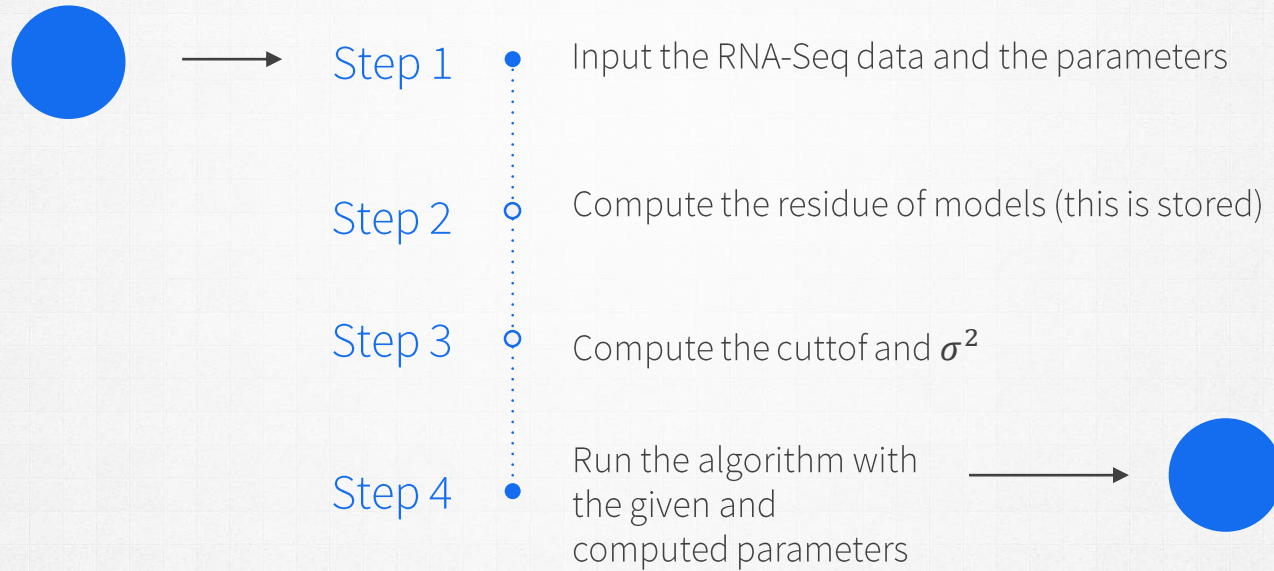
Pipeline of computation

Github repository

Do clusters of genes actually exist ?

- We ran the algorithm for chromosomes with randomly permuted genes
- 50 times each

- Comparison of sizes of circadian partitions
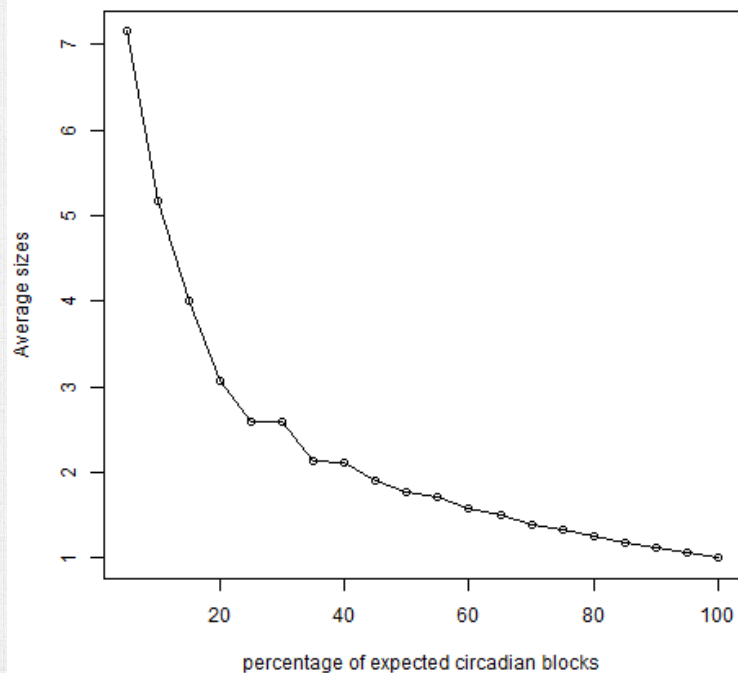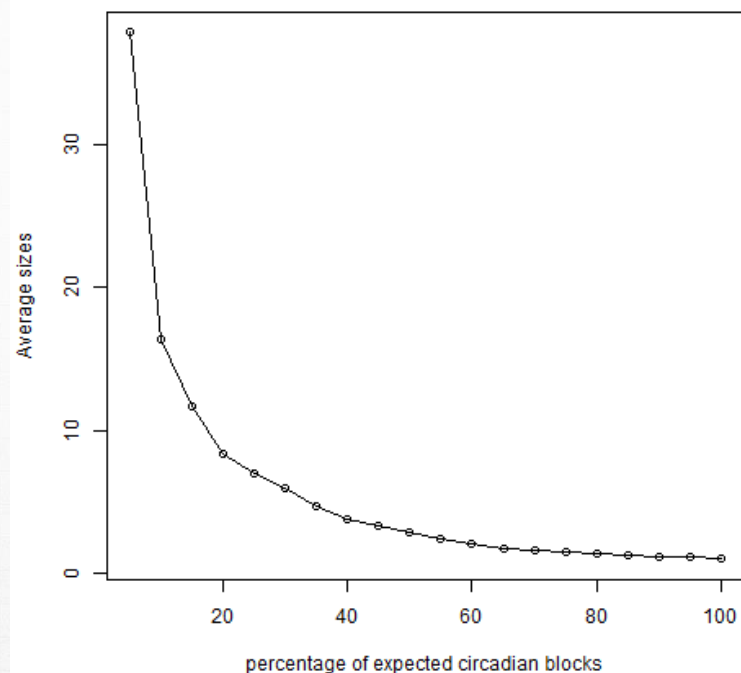
Resulted percentages of circadian blocks

Resulted percentage of circadian blocks, chr4

Average size of partitions
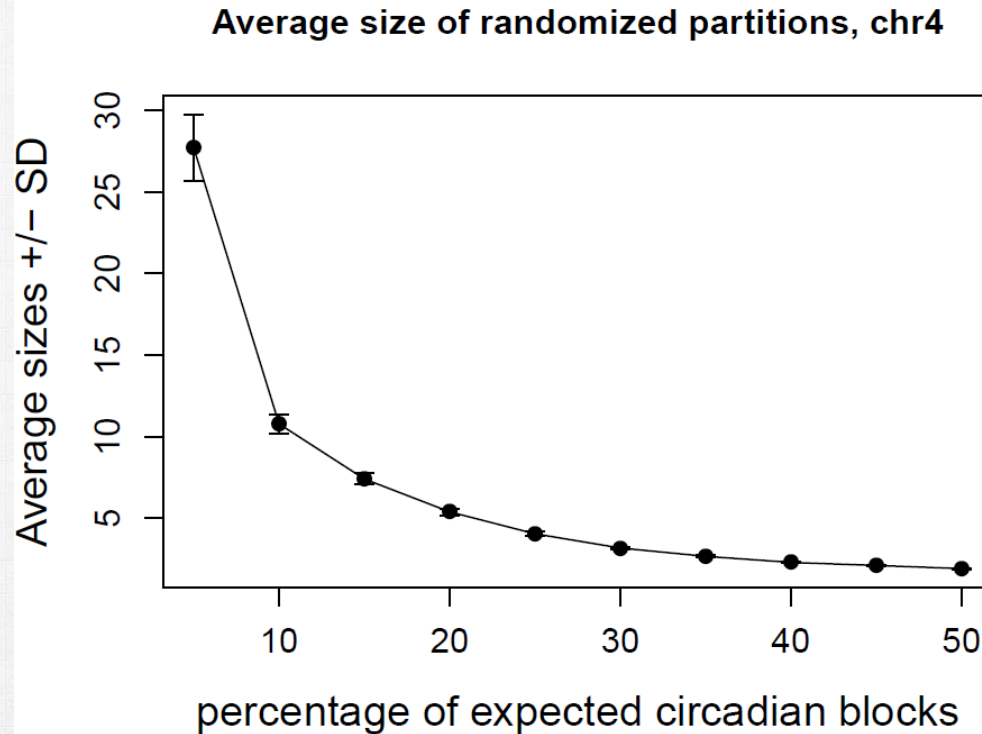
Average size of partitions

Average size of randomized partitions with error bars

—
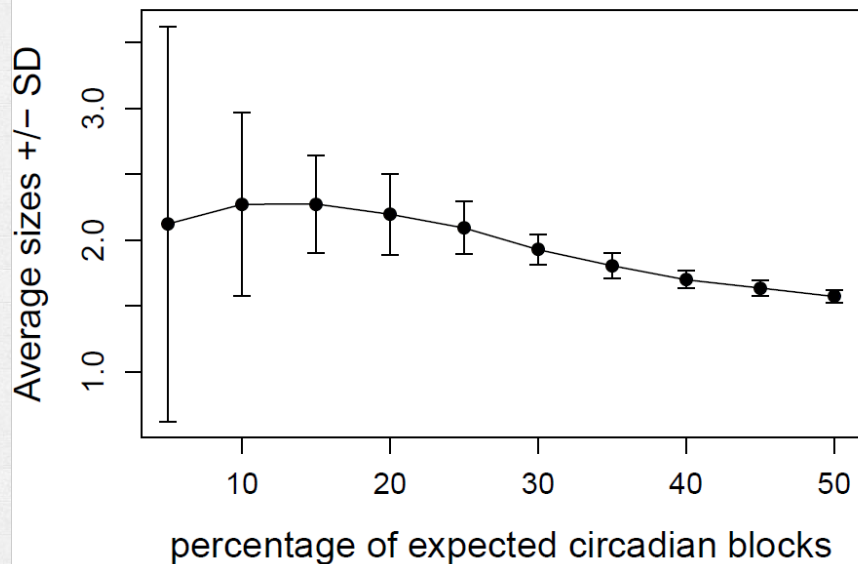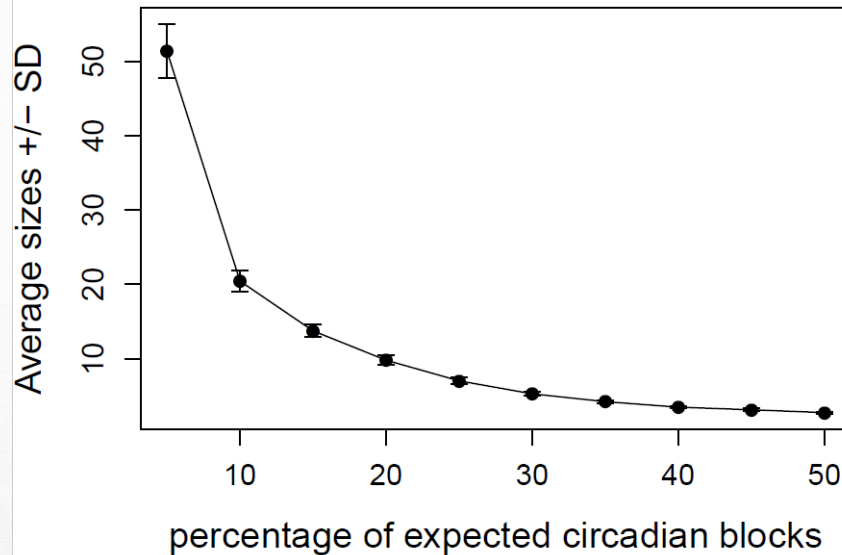
Average size of randomized partitions with error bars

Maximum sizes of partitions

Heatmap percentages 5 to 95
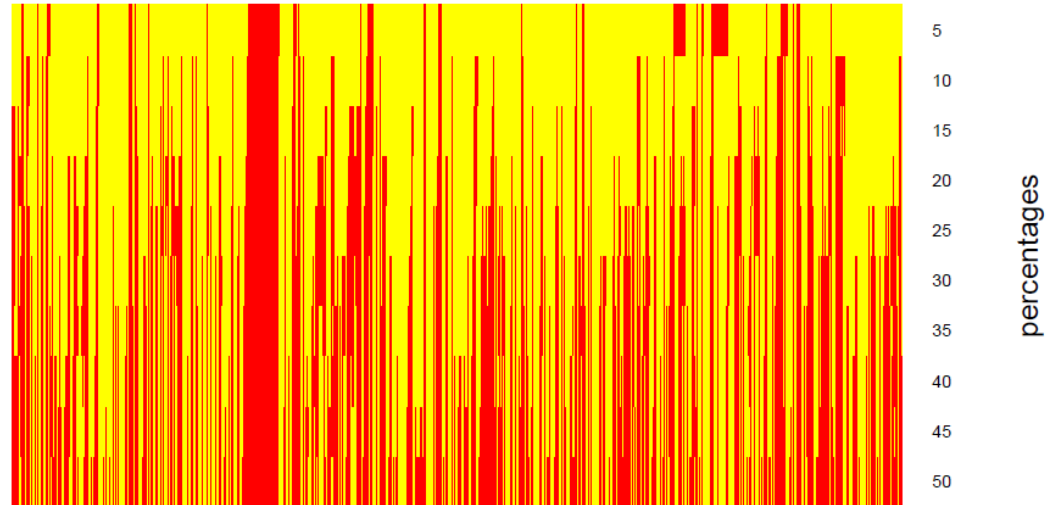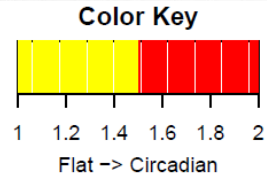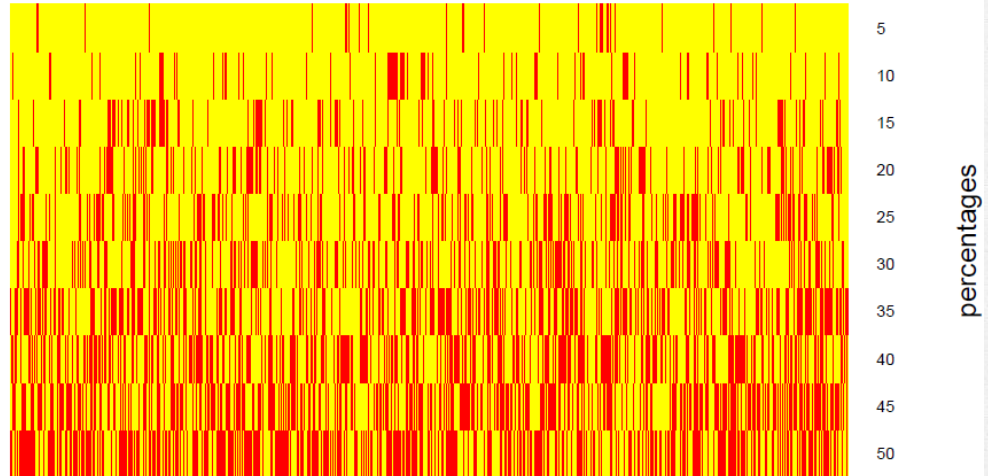
Heatmap percentages 5 to 50
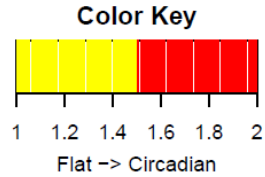
Heatmap randomized chromosome
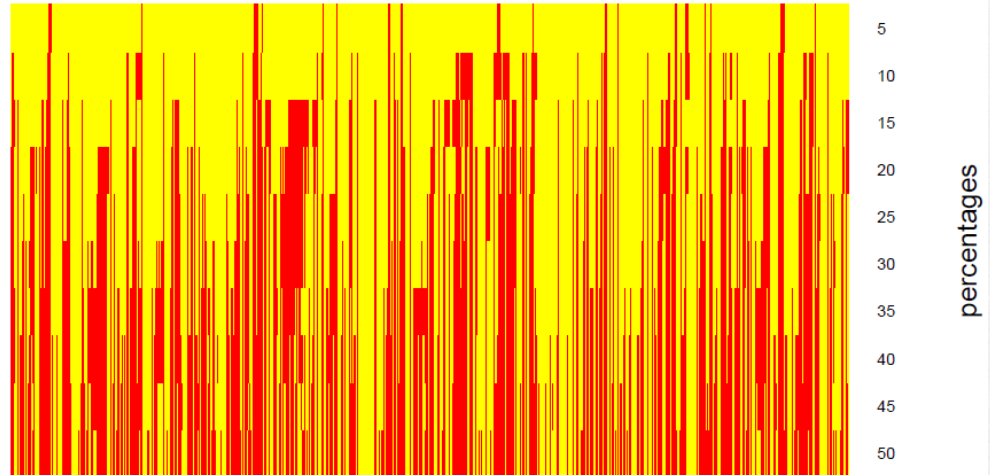
Heatmap percentages 5 to 50

Heatmap randomized chromosome

Heatmap randomized chromosome

Heatmap randomized chromosome

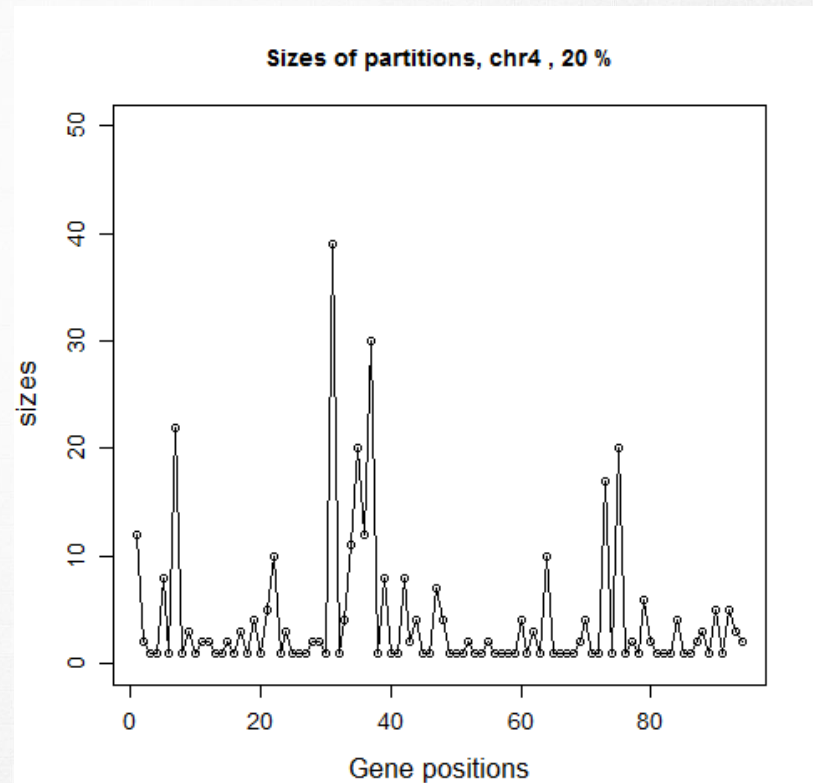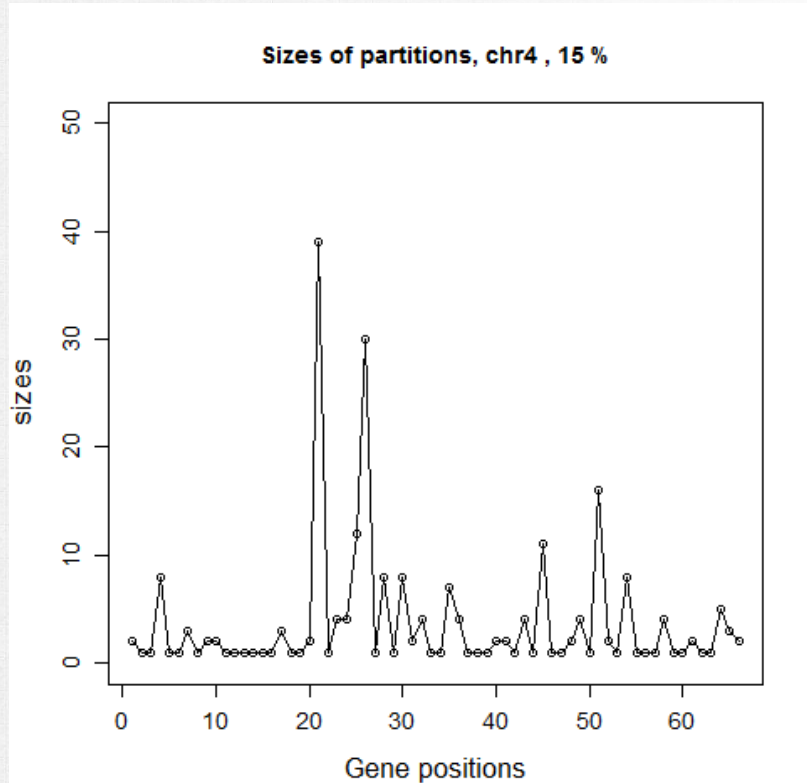# Do clusters of genes actually exist ?

Do clusters of genes actually exist ?
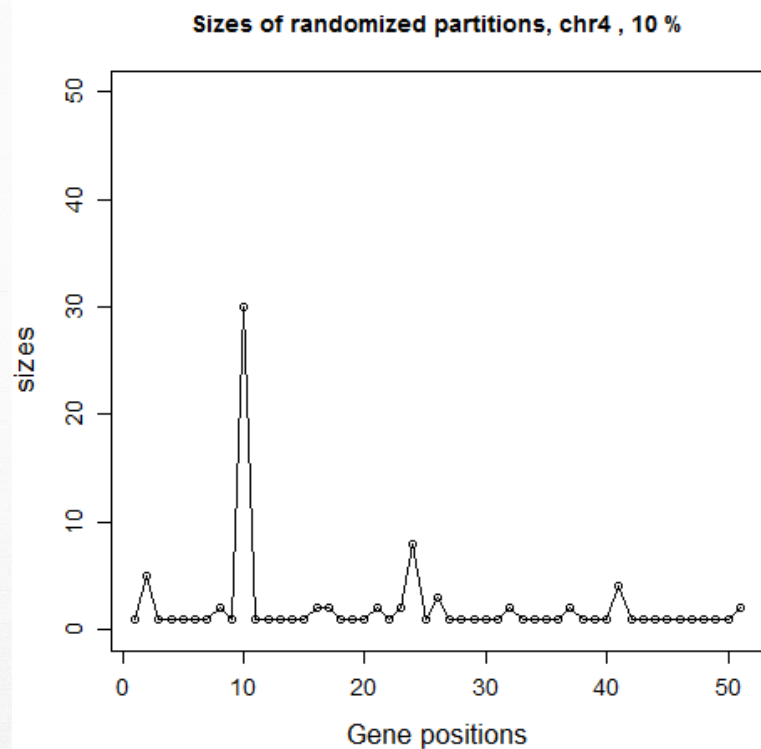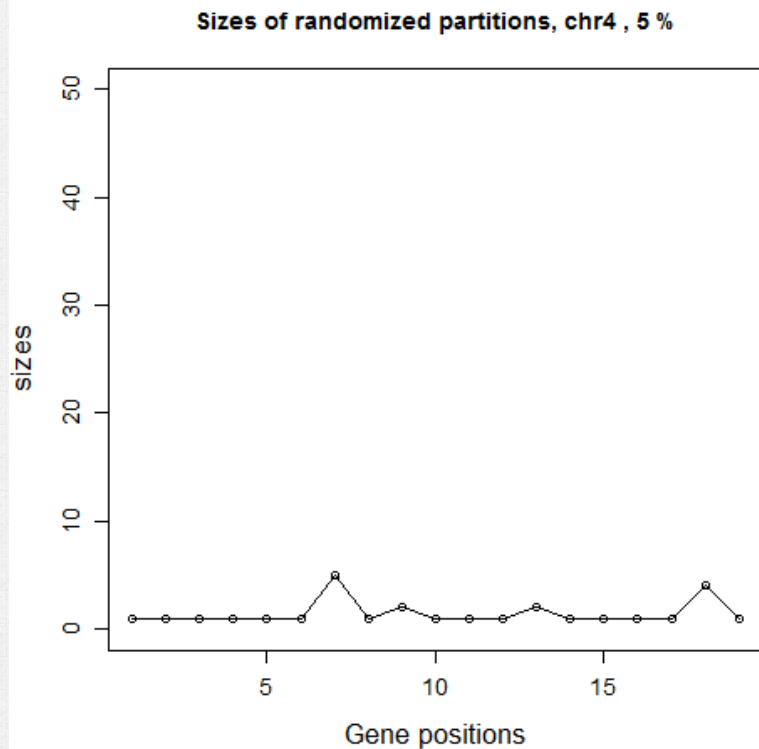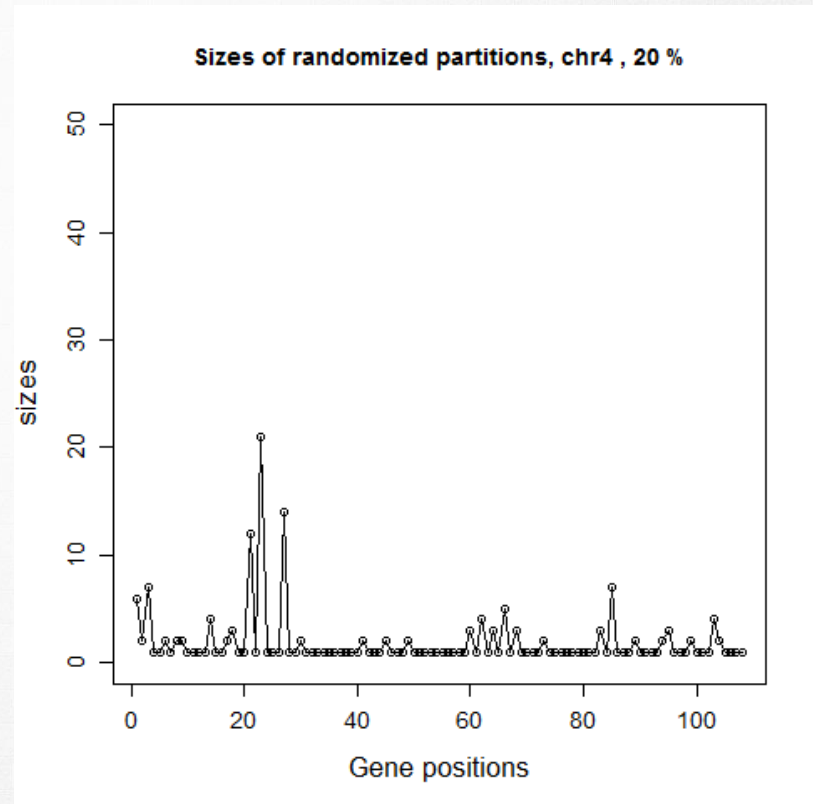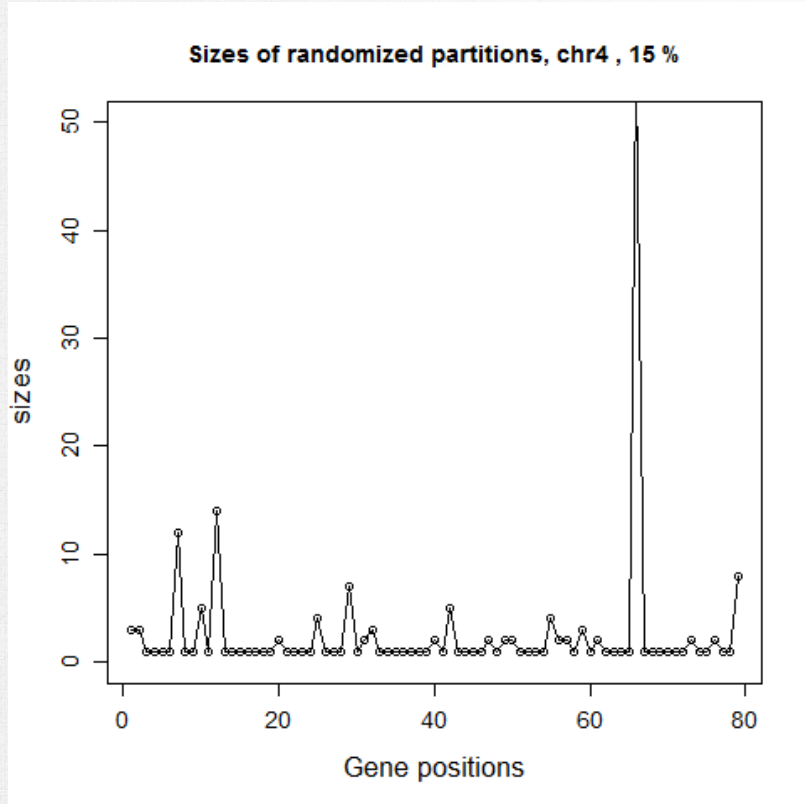
Do clusters of genes actually exist ?

Do clusters of genes actually exist ?

CONCLUSIONS

—

- We adopted an algorithm for partitioning genes along the chromosomes that show the same temporal pattern of expression
- For this specific task, we proposed a new score that solves the issue with the standard BIC approach.
- We identified blocks of genes along the chromosome with a similar temporal pattern.
- We transformed the definition of sigma to the a priori percentage of the rhythmic genes.
- By randomization we can see that the specific positioning of genes are relevant to the expression pattern.

FUTURE DIRECTIONS

—

- Test other models for describing more diverse dynamical patterns of expression

- Using other data set within the same framework and see whether the partitioning is reproducible across data types and conditions.

- Overlay the identified partitions with the chromosome conformation capture (3C) data to introgate the potential interaction between genes placed in the same block type.

- Showing that the genes within the same topological association domain (TAD) are tend to be similar in terms of expression profile.

THANK YOU

Feel free ton clone the github repository

https://github.com/darioAnongba/bachelorProject