

Practical – k-mer distributions

Probabilities and statistics for modelling 1 (STAT1)

Jacques van Helden

2020-02-20

Contents

Introduction	1
Organisms	1
Tutorial for the first steps	2
Working directory	2
Counting k-mer occurrences in each promoter of a model organism	2
Download the count table from RSAT	2
Load the k-mer count table in R	3
Homework	4
Compute marginal statistics	4
Draw the distributions	5
Exploring k-mer count distributions in promoter sequences	5
Fitting distributions	5

Introduction

In this practical, we will count k-mer occurrences in DNA sequences of different organisms (one organism per student), fit different theoretical distributions of probabilities onto the empirical distributions of counts, and test the goodness of fit for these alternative distributions.

We will run the beginning of this practical in tutorial mode, by providing the solutions for

- computing k-mer distributions in the upstream sequences of all the genes of an organism of interest on the RSAT server;
- downloading the resulting k-mer count table;
- loading this table in and R data frame.

We will then pursue with exercises to - explore the data (descriptive statistics) - fit different theoretical distributions on the observed k-mer counts - assess the goodness of the fit.

Organisms

Each student will choose one organism of interest among the following ones.

Taxon	Organism
Fungi	Saccharomyces_cerevisiae
Bacteria	Escherichia coli GCF_000005845.2_ASM584v2
Mammalian	Homo sapiens GRCh38
Mammalian	Mus musculus GRCm38
Bird	Gallus gallus EnsEMBL
Fish	Danio_rerio_EnsEMBL
Insect	Drosophila melanogaster
Worm	
Plant	Arabidopsis thaliana.TAIR10.29
Plant	Zea mays.AGPv3.29
Apicomplexa	

Tutorial for the first steps

Working directory

On your computer, create a directory for this practical. I suggest to use a consistent naming for the different practicals of this course.

We will further create a sub-folder with the name of our organism of interest.

```
## Define your organism of interest
org <- "Homo_sapiens"

## Define a working directory
work.dir <- file.path("~", "CMB-STAT2_practicals", "kmer_distrib", org)
dir.create(work.dir, recursive = TRUE, showWarnings = FALSE)

## Print a message with the result directory
message("Result directory\t", work.dir)
```

Counting k-mer occurrences in each promoter of a model organism

1. Open a connection to the Regulatory Sequence Analysis Tools (RSAT) teaching server : <http://teaching.rsat.eu/>
2. In the tool search box, type “retrieve sequence” and click on the corresponding tool.
3. In the *retrieve-sequence* form,
 - click *Mandatory inputs*, enter the name your organism of interest, and check the option *all genes of this organism*;
 - in *Mandatory options*, select *upstream*, and set the sequence limits from -1 to -500
 - in *Advanced options*, make sure that this option is **unchecked**: Prevent overlap with neighbour genes (noorf)*¹
 - click *Run analysis* and *GO*.

After a few seconds (or minutes) the result is displayed. Right-click on the sequence file (extension fasta) and open it in a separate tab to check its content.

4. Come back to the result page of retrieve-sequence. In the *Next Step* box below the result, click on the link to *oligo-analysis*. This will transfer your sequences to the oligo-analysis form.
 - In the *Sequence* section, inactivate the option *purge sequence*.
 - In the *Oligmer counting mode*, **uncheck** the option *prevent overlapping matches*.
 - Select *Count on single strand*.
 - For *oligomer lengths*, select 2 and **uncheck the other lengths**.
 - In *Results*, check the option *Occurrence table*.
 - Type your email address and select the mail output.
 - Click *GO*.

After a few seconds (minutes), the RSAT server should display the result page, with links to the k-mer count table. Copy the URL of the result file.

Download the count table from RSAT

Let us define the name we will give to the local copy of the k-mer count table generated on the RSAT server in the previous steps.

¹Note: normally it is recommended to check this option, but we intently inactivate it in order to get sequences of the same sizes.

```
## Define the path and the name of the local file
kmer.file <- file.path(work.dir, "2nt-ovlp-1str_Homo_sapiens.tab")
```

One solution is to download manually the k-mer count table generated on the RSAT server, move it to the work directory, and rename it 2nt-ovlp-1str_Homo_sapiens.tab (to be adapted depending on your organism of interest).

Another possibility is to use R command `download.file()` download it from the URL of the result file on the RSAT server.

```
## Note: this will only work for a few days, because the temporary files are removed from the server
temp.url <- "http://pedagogix-tagc.univ-mrs.fr/rsat/tmp/www-data/2020/02/17/oligo-analysis_2020-02-17.1

## Provide the arguments in the order of the function definition
download.file(temp.url, kmer.file)

## Equivalent : name the arguments
# download.file(url = temp.url, destfile = kmer.file)

## Note: named arguments can be provided in a different order without problem
# download.file(destfile = kmer.file, url = temp.url)
```

Whichever method was chosen, check that the file is at the right place on your computer.

```
## List the files in the working directory
list.files(work.dir)
```

```
[1] "2nt-ovlp-1str_Homo_sapiens.tab"
```

```
## Send a message with the k-mer file location
message("K-mer count table file:\t", kmer.file)
```

Load the k-mer count table in R

Use the function `read.table()` to load the k-mer count table in a variable named `kmer.table`.

```
## Call the help for read.table()
# ?read.table

## Load the k-mer count table in a variable
kmer.table <- read.table(
  file = kmer.file,
  comment.char = ";", ## comment lines start with ";" in RSAT
  header = TRUE, # the first row (after the comments) contain the column headers
  row.names = 1, ## the first column contains row names but there might be homonyms
  sep = "\t" ## column separator is the tabulation
)
```

Check the dimensions of this table.

```
## Check the dimensions of the k-mer count table
dim(kmer.table)
```

```
[1] 60675    16
```

```
## Number of k-mers
m <- ncol(kmer.table)

## Number of genes
```

```
n <- nrow(kmer.table)

## Print the result
print(paste0("Number of rows: ", n))
```

```
[1] "Number of rows: 60675"
```

```
print(paste0("Number of columns: ", m))
```

```
[1] "Number of columns: 16"
```

Check the column names

```
## Check the column names
names(kmer.table)
```

```
[1] "aa" "ac" "ag" "at" "ca" "cc" "cg" "ct" "ga" "gc" "gg" "gt" "ta" "tc" "tg" "tt"
```

```
colnames(kmer.table) # equivalent
```

```
[1] "aa" "ac" "ag" "at" "ca" "cc" "cg" "ct" "ga" "gc" "gg" "gt" "ta" "tc" "tg" "tt"
```

Display the first and last 10 lines of the k-mer count table.

```
## Show the first 10 lines of the k-mer count table
head(kmer.table)
```

	aa	ac	ag	at	ca	cc	cg	ct	ga	gc	gg	gt	ta	tc	tg	tt
ENSG00000210049 Homo_sapiens_GRCh38 MT-TF	48	54	17	36	58	69	14	32	14	24	7	11	35	26	18	36
ENSG00000211459 Homo_sapiens_GRCh38 MT-RNR1	56	58	16	36	62	69	9	31	10	19	7	11	38	25	15	37
ENSG00000210077 Homo_sapiens_GRCh38 MT-TV	57	42	41	22	37	38	18	35	28	26	22	27	39	23	22	22
ENSG00000210082 Homo_sapiens_GRCh38 MT-RNR2	48	42	42	22	35	39	17	36	30	25	22	28	41	21	24	27
ENSG00000209082 Homo_sapiens_GRCh38 MT-TL1	52	41	32	34	40	40	19	29	28	16	23	23	38	32	16	34
ENSG00000198888 Homo_sapiens_GRCh38 MT-ND1	48	38	35	32	39	37	21	29	30	16	24	25	36	37	15	35

```
tail(kmer.table)
```

	aa	ac	ag	at	ca	cc	cg	ct	ga	gc	gg	gt	ta	tc	tg	tt
ENSG00000128973 Homo_sapiens_GRCh38 CLN6	55	24	34	31	31	31	15	36	30	34	29	19	27	24	34	45
ENSG00000272269 Homo_sapiens_GRCh38 RP11-500C11.3	41	29	33	14	43	90	25	33	21	38	32	15	11	35	16	23
ENSG00000267091 Homo_sapiens_GRCh38 CTBP2P7	21	22	38	25	40	34	9	37	30	35	28	32	15	29	50	54
ENSG00000151655 Homo_sapiens_GRCh38 ITIH2	62	30	35	33	48	23	2	41	27	22	17	19	23	39	30	48
ENSG00000234159 Homo_sapiens_GRCh38 RBPM5LP	112	30	39	36	41	34	4	24	32	20	18	14	33	19	23	20
ENSG00000141338 Homo_sapiens_GRCh38 ABCA8	74	22	39	48	31	13	4	30	43	21	19	16	35	22	37	45

Homework

- Read the tutorial first steps with R
- Compute marginal statistics
- Draw histograms of a given k-mer of your choice

Compute marginal statistics

Tips: use the R function `apply()`.

Draw the distributions

Exploring k-mer count distributions in promoter sequences

1. Load the k-mer count table generated in the previous step.
2. Draw an histogram with the distribution of counts for a given k-mer. After having fine-tuned the representation, generate a pdf file with 4 x 4 panels to depict the histograms of the 16 k-mers.
3. Use other graphical representations to get an insight of the k-mer count distributions (boxplots, violin plots)
4. Compute summary statistics for each column of the count table, including the following estimators
 - min and max
 - mean
 - percentiles 05, 25 (=Q1), 50 (=median), 75 (=Q3), 95
 - variance and standard deviation
 - sum
5. Compute a vector with the relative frequency of each k-mer in all the sequences.
6. Compute a table with the relative frequencies of k-mers per sequence, and compute similar summary statistics per column on this relative frequency table.
7. Write a brief interpretation of the results.

Fitting distributions

1. Fit a Poisson distribution on each empirical distribution of k-mer counts.
 - How do you choose the parameters?
 - Draw the fitted Poisson distribution over the histogram of empirical distribution (observed k-mer occurrences)

Tips:

 - in order to add some plot over an existing plot, you can use the `lines()` function
 - you can also use specific options to draw histogram-like lines: `'lines(x, y, type = "h")`.
2. Do the same with the following distributions :
 - a. binomial
 - b. hypergeometric
 - c. normal
 - d. negative binomial
3. Estimate the goodness of fit for these distributions.