R'MES

Finding Exceptional Motifs

Version 3.1.0

Mark HOEBEKE[1] and Sophie SCHBATH[2]

November 23, 2009

`http://genome.jouy.inra.fr/ssb/rmes/`

Institut National de la Recherche Agronomique
[1] Laboratoire Statistique et Génome, Evry, France
[2] Unité Mathématique, Informatique & Génome, Jouy-en-Josas, France

Contents

Chapter 1

Introduction

1.1 Aim

The main question R'MES addresses is "does this motif occur in a DNA[1] sequence with an expected frequency?" In other words, can we observe it so many times, or so few times, just by chance? Usually, when the answer is no, such a motif is a candidate to have a particular biological meaning; Only a candidate because statistical significance is not equivalent to biological significance.

1.2 Statistical method

Here is a brief presentation of the statistical method used in R'MES to evaluate the significance of a motif frequency in a sequence. For more details about the methodology, please refer to the tutorial available on the R'MES home page (`http://genome.jouy.inra.fr/ssb/rmes/`) or to the book *DNA, Words and Models* by Robin, Rodolphe and Schbath published by CUP in 2005 (or by BELIN in 2003 for the French version).

A model as reference. The key idea is to compare the observed count of the motif with the expected one given some knowledge about the sequence. To decide if a word count is indeed unexpected, we need to know what to expect. This will be defined by a probabilistic model, i.e. by the description of what "random" means. In practice, Markovian models are used because a Markov chain model of order m fits the observed

[1] In this manual the explanations focus on DNA sequences, although R'MES is capable of handling other kinds of sequences.

counts of all oligonucleotides of length 1 up to $(m+1)$ of the observed sequence. Let us denote by Mm such model.

Choice of the model. Choosing model Mm means to take the base, the dinucleotide, the trinucleotide, ..., the $(m+1)$-mer compositions of the sequence into account to determine what to expect. However, the sequence should be long enough to correctly estimate the 3×4^m parameters of the model (the transition probabilities). Note that a motif of size ℓ can be only analyzed in M0 up to M$(\ell-2)$ because higher models would fit the motif count itself (the motif will then be expected by definition). Remember that the model determines the reference; So, changing the reference may change the exceptionality feature of a motif. A word can be exceptionally frequent in one model but expected in another one which, for instance, takes more information on the sequence composition into account. Therefore, when claiming that an observation is statistically significant, do not forget to mention your a priori, your reference, your model.

p-value. To evaluate the significance of the difference between observed and expected counts, we need to evaluate the p-value which is the probability, under our model, to observe as much (or as few) occurrences of our motif of interest. It requires the statistical distribution of the count of a motif. Several methods exist either to calculate this p-value exactly (not tractable for long sequences) or to approximate it. Two kinds of approximations exist: a direct approximation using large deviation techniques or an approximation of the motif count distribution. R'MES uses the latter, namely a Gaussian approximation which is suitable for expectedly frequent motifs or a compound Poisson approximation adapted for expectedly rare motifs.

Score of exceptionality. R'MES converts the p-values into scores of exceptionality using the standard one-to-one probit transformation: for a given probability $p \in [0,1]$, the associated score $u \in \mathbb{R}$ is such that $\mathbb{P}(\mathcal{N}(0,1) \geq u) = p$. Therefore, exceptionally frequent motifs will have high positive scores, whereas exceptionally rare motifs will have high negative scores. When using the Gaussian approximation, R'MES directly calculates the scores which is much faster. When using the compound Poisson approximation, the p-values are first calculated, either directly with an efficient algorithm or, for word families, by summing up probabilities of the form $\mathbb{P}(\text{count} = x)$ (numerical problems may happen for frequent families).

Chapter 2

Installing R'MES

R'MES comes as a source distribution only, and needs to be compiled before use. The following sections describe the requirements and the procedure to get and to install R'MES properly.

2.1 System Requirements

Installation Requirements. R'MES is written in C and C++. Our distribution was specifically designed to be compiled with the GNU GCC compiler. It has been tested on a variety of Unix platforms (Linux, MacOS X).

Run-time Requirements. The amount of RAM needed by R'MES primarily depends on the length of the words or motifs under study. For words of length ℓ, the amount of memory grows with a factor of n^ℓ (where n is the size of the alphabet, 4 in the case of DNA). To study words of size 10 in DNA sequences, the number of words that have to be stored is $4^{10} = 2^{20} = 1\text{M}$. Each word requires about 30 bytes to store its parameters (depending on the approximation method). So all in all, R'MES will need $1\text{M} \times 30\text{bytes} = 30\text{Mb}$ to store word information (added to that, are the counts of all words whose lengths are in the requested Markov order interval and the sequence amongst others, but these quantities tend to be small relative to word information when word lengths get large).

2.2 Getting R'MES

R'MES is a free software package available under the GNU General Public License. It can be downloaded from its home web page or directly from `https://mulcyber.toulouse.`

inra.fr/projects/rmes/.

R'MES has a companion tool, RMESPlot which is available at https://mulcyber.toulouse.inra.fr/projects/rmesplot/ and provides a graphical user interface for the visualization of R'MES generated results.

2.3 Compiling and Installing R'MES

R'MES´ installation procedure follows the GNU package distribution standards. So, after downloading rmes-<version>.tar.gz (where <version> stands for the version number) here is the list of steps to perform.

1. Extract the archive and change to the extracted directory:

```
$tar zxvf rmes-<version>.tar.gz
$cd rmes-<version>
```

2. Configure the compilation/installation:

```
$./configure
```

   This will install the R'MES programs in the /usr/local/bin directory. If R'MES is to be installed elsewhere, the --prefix option to configure must be used. For instance, to install R'MES in user johndoe's home directory :

```
$./configure --prefix=/home/johndoe
```

   The programs will then be installed in /home/johndoe/bin. The list of options provided by compile and their meanings can be obtained as follows:

```
$./configure --help
```

3. Compile the programs:

```
$make
```

4. Check the compiled programs (optional):

```
$make check
```

This runs the freshly compiled programs and checks their results against a standard set of results, reporting any inconsistency. If the checks fail, it might be because of numerical discrepancies between platforms with no measurable consequences on the results produced by R'MES. In doubt, please mail the R'MES team list (`rmes@jouy.inra.fr`) for support.

5. Install the programs. This step needs write access to the directories where the package will be installed (by default `/usr/local` and its subdirectories) and may need to be performed as `root`.

```
#make install
```

For more details, refer to the `INSTALL` file included in the source distribution.

Chapter 3

Running R'MES

R'MES has to be run via a command line which looks like:

```
$rmes [options] -s <filename> -o <string>
```

All the options can be obtained by typing:

```
$rmes --help
```

In this chapter, we start by giving the most basic use case of R'MES and then we describe the other possible cases with the associated options.

3.1 Basic Command

The most basic use case of R'MES consists in analyzing all the oligonucleotides of a given length in a given sequence. Naturally, the input parameters are

- *the sequence file*: it is provided after the `-s <filename>` option; The sequence should be in FASTA or Genbank format,
- *the word length*: it is provided after the `-l <int>` option,
- *the order of the model*: it is provided after the `-m <int>` option,
- *the approximation method*: it is provided either by the `--gauss` option for the Gaussian approximation or by the `--compoundpoisson` option for the compound Poisson approximation.

An additional option `-o <string>` is required to specify the prefix of the output files. Then, the basic commands look like:

```
$rmes --gauss -s <filename> -l <int> -m <int> -o <string>
```

or

```
$rmes --compoundpoisson -s <filename> -l <int> -m <int> -o <string>
```

Output file The above commands will produce a unique output file with the `'.0'` suffix. This file is not intended to be read by the user but only to store the numerical values of each of the quantities of interest (observed counts, estimated expected counts, scores etc.). To obtain user readable output, this file needs to be formatted. Two tools are available for this (see Section 3.7): the `rmes.format` program included in the source distribution or the java interface `RMESPlot`. Moreover the output file will be compressed if the `-z` option is specified.

Simultaneously analyzing several word lengths It is possible to simultaneously analyze several word lengths with a single command. For this, the `-l <int>` option should be replaced with options `--lmin <int>` and `--lmax <int>` which specify the minimal and maximal word lengths.

Using the maximal model When using the maximal model, i.e. when $m = \ell - 2$, the `-m <int>` option can be replaced with `--max`[1]. Moreover, the `--max` option allows the simultaneous analysis of several word lengths, each one in the associated maximal model (see previous paragraph).

Analyzing concatenated sequences R'MES can consider a concatenation of several sequences as a single sequence. In order to avoid introducing non-existing words at the boundaries of each piece of the concatenated sequence, the latter must be separated by a specific character which depends on the sequence type or alphabet. For DNA, this separator is the letter "Z", and for amino acids the separator is the letter "X". Separators can be either upper or lower case. For user specified alphabets, the separator must be part of the alphabet definition (see Section 3.4).
Warning: the sequence file should however look like a unique sequence file in FASTA or GenBank format. In FASTA format, for instance, the sequence file should contain a unique title line and a unique sequence of concatenated bases.

[1]When using the maximal model, be it explicitly with the `--max` option or implicitly when $m = \ell - 2$, a very efficient algorithm is used to calculate the score from the Gaussian approximation (option `--gauss`).

3.2 Finding exceptional word families or degenerated motifs.

To analyze families of oligonucleotides, for instance degenerated oligonucleotides of length 6 with an `'n'` in second position, or starting with a purine, or oligonucleotides with their reverse complementary, ..., the `-l <int>` option must be replaced by the `-f <filename>` option, in which `<filename>` represents the file which enumerates the families. Both approximations can be used. The basic commands become:

```
$rmes --gauss -s <filename> -f <filename> -m <int> -o <string>
```

or

```
$rmes --compoundpoisson -s <filename> -f <filename> -m <int> -o <string>
```

Compatibility with other options The `-f <filename>` option can be used with the `--max` (maximal model) option but is not compatible with the word length options `-l <int>`, `--lmin <int>` and `--lmax <int>`.

Format of the family file A word family has to be composed of words of the same length, say ℓ. All the families contained in a family file have to be composed of the same number of words, say d. The structure of the associated family file is as follows:

- the first line is a title (character string) ended by the `#` character,
- the second line contains the number of families,
- the third line contains the number d of words in each family,
- the fourth line contains the length ℓ of the words,
- then, each family is listed as follows: the family name followed by all the d words of this family

An example can be found on the R'MES home page.

Output file The above commands will produce a unique output file with the `'.0'` suffix. It is a storage file but is human readable since each of its lines successively contains the word family name, its observed count, its estimated expected count, either the normalizing factor for the Gaussian approximation or again its estimated expected count for the compound Poisson approximation, and finally its score. However, this output file can also be formatted thanks to `rmes.format` or visualized with the graphical interface `RMESPlot`. The output file will be compressed if the `-z` option is specified.

Note The `-f <filename>` option can also be used to study a restricted list of oligonucleotides.

3.3 Analyzing coding DNA sequences.

When analyzing coding DNA sequences, it is usually more relevant to use a Markov model that takes the phase into account. More generally, one may be interested to take into account some periodicity of the sequence. For this purpose, the `--phases <int>` option can be used to specify the period. Note that only the Gaussian approximation is available with phased models. The basic command becomes:

```
$rmes --gauss -s <filename> -l <int> -m <int> --phases <int> -o <string>
```

Compatibility with other options The `--phases <int>` option can be used with the `--max` (maximal model) option, with the word length options `--lmin <int>` and `--lmax <int>` and with the family option `-f <filename>`. However, it cannot be used with the `--compoundpoisson` (compound Poisson approximation) option.

Output files If a phased model with r phases is used, the above command will produce $r+1$ output files with suffixes `'.1'`, `'.2'`, ..., `'.`$(r+1)$`'`. The file suffixed by `'.`i`'`, $1 \leq i \leq r$, corresponds to the number of occurrences ending exclusively on phase i, while the file suffixed by `'.`$(r+1)$`'` corresponds to the number of all occurrences (like for a non phased model). Each of these files has the same structure than the one obtained with a non phased model (see previous paragraphs) and can be also automatically compressed.

3.4 Using different alphabets

Character case in sequences The default alphabets included in R'MES are case insensitive.

Nucleotide alphabet By default, the nucleotide alphabet is used assuming that the analyzed sequence is a DNA sequence.

Amino acid alphabet When analyzing protein sequences, the amino acid alphabet should be used with the `--aa` option.

Other alphabets The user can specify his/her own alphabet with the `--alphabet <character string>` option. The character string explicitly describes the alphabet used in the sequence according to a particular format:

`validchars:interruptchars:jokerchar`

where

- `valdichars` is the list of allowed characters of the sequence,
- `interruptchars` is the list of characters used to separate pieces in a concatenated sequence,
- `jokerchar` is a single character standing for any of the valid characters.

For instance, if sequences of protein secondary structures are to be analyzed, the alphabet definition could be: `ABC:X:N`. Where `A` could stand for alpha-helix, `B` for beta-sheet and `C` for coil. `X` would be used to separate pieces of sequences and `N` could replace any of `A`, `B` or `C`.

3.5 Finding exceptional skewed motifs

When analyzing a DNA sequence, one can be interested to know if an oligonucleotide has a significant skew. The skew of an oligonucleotide is usually defined like the ratio between the oligonucleotide count and the count of its reverse complementary. The skew is used to describe the strand bias. The p-value associated with an oligonucleotide skew can be easily approximated using a Gaussian approximation of the word counts. With R'MES it is then possible to get an exceptionality score of the skew of an oligonucleotide or of a

word family. To do this, the `--gauss` option from the basic command must be replaced with the `--skew` option:

```
$rmes --skew -s <filename> -l <int> -m <int> -o <string>
```

Compatibility with other options The `--skew` option can be used with the `--max` (maximal model) option , with the word length options `--lmin <int>` and `--lmax <int>` and with the family option `-f <filename>`. However, it cannot be used with the options `--compoundpoisson` (compound Poisson approximation) and `--phases <int>` (phased models), neither with the alphabet options `--aa` and `--alphabet <character string>`.

Additional output file The quantities associated with the skew and its significance are stored in an additional file with the `'.skew'` suffix. This file is presented like a table: the columns successively correspond to the word (or word family), the observed count, the observed count of the reverse complementary, the observed skew and the score of exceptionality of the skew.

3.6 Words with an exceptional number of clumps

Overlapping words, i.e. words whose occurrences can overlap in a sequence, may form clumps of overlapping occurrences in a sequence. Instead of studying the significance of the number of occurrences of a word, one may be interested in studying the significance of the number of clumps of a given word. This question has been partially addressed in R'MES because few theoretical results exist in the literature on the number of clumps. The score of exceptionality provided by R'MES is based on a Poisson approximation of the number of clumps and is only available for exact oligonucleotides under model M1. The command is as follows

```
$rmes  --poisson -s <filename> -l <int> -o <string>
```

where the options `-s`, `-l` and `-o` are similar to the basic command (see Section 3.1).

Compatibility with other options The `--poisson` option is not compatible with the Gaussian nor the compound Poisson approximation options (`--compoundpoisson` and `--gauss`), with the model order options (`-m <int>` and `--max`) neither with the family option (`-f <filename>`).

Output file The above command will produce a unique output file with the `'.0'` suffix. This file is not intended to be read by the user but only to store the numerical values of each of the quantities of interest (observed number of clumps, estimated expected counts, scores etc.). This file will then need to be formatted with the `rmes.format` program or loaded into `RMESPlot` (see Section 3.7). Moreover the output file will be compressed if the option `-z` is specified.

3.7 Utilities

3.7.1 rmes.format

This program displays the results contained in an output file (`<rmesfilename>`) generated by the `rmes` command. It produces a table with the motifs sorted according to their exceptionality scores. The basic command is

```
$rmes.format < <rmesfilename> > <tablefilename>
```

The meaning of the different columns of the output table differs from the approximation used, i.e. from the options `--gauss`, `--compoundpoisson`, `--poisson` or `--skew`.

- Gaussian approximation (options `--gauss` or `--skew`): the 6 columns successively correspond to the motifs, their observed count, their estimated expected count, their estimated limiting variance, their score of exceptionality and their rank when all motif scores are sorted by increasing order.

- Compound Poisson approximation (option `--compoundpoisson`): the 7 columns successively correspond to the motifs, their observed count, their estimated expected count, the estimation of their expected number of clumps, their overlap probability, their score of exceptionality and their rank when all motif scores are sorted by increasing order.

- Poisson approximation (options `--poisson`): the 6 columns successively correspond to the motifs, their observed number of clumps, the estimation of their expected number of clumps (twice because the variance of a Poisson variable is equal to its expectation), their score of exceptionality and their rank when all motif scores are sorted by increasing order.

Analyzing several word lengths If the input file contains results for more than one word length, then as many tables as the considered word lengths will be produced. However, if only a subset of word lengths is of interest, the `-l <int>`, `--lmin <int>` or `--lmax <int>` options can be used.

Score thresholds The significance associated with a given score is obtained thanks to the standard Gaussian distribution with mean 0 and variance 1. In particular, scores ranging from -3 to 3 are not really significant (p-value greater than 0.00135). Therefore, motifs with such scores are removed by default from the tables. To specify other thresholds, one can use the `--tmax <float value>` and `--tmin <float value>` options; In this case, only motifs with a score greater than the maximal threshold or less than the minimal threshold will be displayed in the table.

3.7.2 rmes.gfam

This program allows to generate family files when the corresponding families are degenerated DNA motifs which can be written thanks to the bases `a`, `c`, `g`, `t` and `n`. The basic command is:

```
$rmes.gfam  -t <label> -p <string>
```

The `-t <label>` option just specifies the title of the resulting family file (this title will be the first line of the family file).

The pattern specified by the `-p <string>` option is the template to generate the degenerated motifs (families). Its length ℓ will be the length of the words in the families. Each of its ℓ characters can take a value among `#`, `a`, `c`, `g`, `t` and `n`.

- If the i-th character of the template is `a`, `c`, `g`, `t` or `n`, then the i-th character of all the degenerated motifs will be set to this character.
- If the i-th character is a `#`, then the i-th character of the degenerated motifs will be successively `a`, `c`, `g` and `t`.

For instance, the template `#n##` will produce the 64 degenerated tetranucleotides having an `n` in the second position, whereas the template `gnta` will produce the unique degenerated motif `gnta`. The number of families will then be 4^b where b is the number of `#`'s in the template.

3.7.3 rmes.composition

This program allows to know the length of a sequence (number of valid characters + number of separators + number of joker characters) and its composition. The basic command

```
$rmes.composition  -s <filename> -l <int>
```

gives the composition of the input sequence in words whose length has been specified by the `-l <int>` option. If this option is missing, the letter composition will be given by default. Note that the `-l <int>` option can be replaced by the `--lmin <int>` and `--lmax <int>` options if several compositions are requested.

Alphabet By default, the alphabet is the 4-letter DNA alphabet. The option `--aa` can be used if the sequence is composed of amino acids. In the current release, only the DNA or amino acid alphabets are supported.

3.7.4 RMESPlot

RMESPlot is a stand-alone Java application to explore result files generated with R'MES. It is available separately at `https://mulcyber.toulouse.inra.fr/projects/rmesplot/`. RMESPlot comes with its own user guide.

Chapter 4

Frequently asked questions

Is there a limit for the sequence length? There is absolutely no limit.

Is there a limit for the word length? There are two limits. The first one is due to the fact that all the words of a given length have to be analyzed (several quantities per word have to be stored) and the number of such words has to be smaller than the largest integer in the computer. For a 4-letter alphabet, for instance, this limit is 14. The second limit is a practical one and depends on the memory available on the computer (see Section 2.1).

Which Markov model to use? The choice of the Markov model, in particular of the order m, really depends on the sequence composition one wants to fit. Higher the order, better the fit. But we have the constraint that $m \leq \ell - 2$ where ℓ is the word length (see Introduction) and that the number of model parameters should not be too high compared to the sequence length; As a crude rule, be sure that $n \geq 100\,(|\mathcal{A}| - 1)|\mathcal{A}|^m$, where n is the sequence length and $|\mathcal{A}|$ is the size of the alphabet. The maximal model (order $\ell - 2$) is probably the most interesting one when performing a systematic analysis of exceptional oligonucleotides of a sequence, starting from dinucleotides under M0, then trinucleotides under M1, etc. Clearly, in the maximal model, all exceptional words have a frequency that cannot be explained by sub-word frequencies.

Which approximation to use? It is quite well known now that a Gaussian approximation is not good for the count of expectedly rare words (see Robin and Schbath, *J. Comp. Biol.* 2001 for instance). In this case, a compound Poisson approximation is much better. The frontier is not easy to determine, but if the question is to detect the most

exceptional words, then both approximations will give the same list of words. similar results. Since the scores obtained by the Gaussian approximation are faster to get, it is the method to try first. If the question is really to get accurate p-values, then the compound Poisson approximation is better. Note that numerical problems may arise when using the compound Poisson approximation to compute the p-values of frequent and expectedly frequent word families (a warning message will appear in such cases). This problem does not exist for (single) words because the tail of the corresponding compound Poisson distribution can be directly and efficiently calculated. As a crude rule, calculate the naive expected count of a word of length ℓ in a sequence of length n namely $n/(|\mathcal{A}|^\ell)$; If it is smaller than 10 then use the compound Poisson approximation, if it is greater than 100 then use the Gaussian approximation (in between is the twilight zone).

Mailing list `rmes-users` If you want to be informed about new developments related to R'MES (new releases, new functions, bugs and their correction etc.), join the `rmes-users` mailing list. To subscribe, just send an email to `sympa@listes.inra.fr` with
`Subject`: empty
`Message`: `SUB rmes-users` Firstname Lastname
or to the address below.

Contact information Please address questions and bug reports via Email to:
`rmes@jouy.inra.fr`