Ray M. Marín, Miroslav Šulc, and Jiří Vaníček

PACCMIT-CDS

Concise User's Manual

PACCMIT-CDS is a program searching for microRNA targets within coding sequences. The program is written in C++ and is distributed in source form. The current version can be downloaded in a compressed form from http://lcpt.epfl.ch. Below we describe the basic usage of PACCMIT-CDS assuming a Linux-like operating system running on x86-64 architecture. For more advanced use of the algorithm and for details needed to run the code on other architectures, see PACCMIT-CDS User's Manual. As a courtesy to the authors, if you use this algorithm, please cite Ref. [1].

1. Download the tarball paccmit-cds.tgz from our website http://lcpt.epfl.ch and decompress it by means of the command

   ```
   tar -xzvf paccmit-cds.tgz
   ```

2. Compile the program with the command

   ```
   make
   ```

3. The basic function of PACCMIT-CDS is taking two input files – one with coding sequences of the genes of interest and one with sequences of the miRNAs, and computing for each miRNA-gene pair M-G the probability (P_{SH}) that an interaction between M and G would occur by chance. The smaller this probability, the more likely is gene G the target for miRNA M.

   There are two basic ways to run PACCMIT-CDS:

   - if the conservation of target sites is not required:

     ```
     ./paccmit-cds                                    \
                   -g  ./data/genes_noncons_example.fa \
                   -m ./data/miRNAs_example.fa         \
                   -i 8
     ```

   - if the conservation of target sites is required (*e.g.*, in at least 12 species):

     ```
     ./paccmit-cds                                    \
                   -g  ./data/genes_cons_example.fa    \
                   -m ./data/miRNAs_example.fa         \
                   -i 8                                \
                   -x 27                               \
                   -M">11"
     ```

   The meaning of the program options employed above is:

   - the option `-i 8` requests the P_{SH} values to be evaluated with precision 10^{-8}. Typically, the P_{SH} values are sufficiently resolved if the value of `-i` is set to

     $$\log_{10}\big[\#\mathrm{miRNAs} \times \#\mathrm{genes}\big].$$

This value (or larger) should be used in productions runs. The run time of PACCMIT-CDS for large databases can be lowered by running the code in parallel (see Section 5.2).

- `-g` and `-m` options (required) specify the location of the gene and miRNA input files in FASTA format [see `genes_noncons_example.fa` and Section 3.1 for more detailed description]
- if the conservation is required, the switch `-x` determines the number of aligned sequences while the `-M` switch specifies the conservation mask [see Subsection 3.3.2]. The preceding example requires the conservation of the target sites in at least 12 species. For the precise format of the aligned sequences needed in this case, inspect the file `genes_cons_example.fa` and see Section 3.1.

4. The output of PACCMIT-CDS is printed into files `stat_1e`i`.dat` for each i from 1 to the value set by the `-i` option. Each line of file `stat_1e`i`.dat` corresponds to a unique gene-miRNA pair and contains, in the sixth column, the P_{SH} value of this pair computed with precision 10^{-i}. In order to rank the predictions generated by either of the preceding two examples, simply sort the output file with:

   ```
   sort -n -k6,7 stat_1e8.dat > ranked_predictions
   ```

   The top predictions appear at the top of the file `ranked_predictions`, which is basically the final result of the PACCMIT-CDS algorithm. Note that generating the preliminary results in files with lower resolution of P_{SH}-values is useful since the calculation of final results can take a lot of time for large input files.

   Finally, we note that files `human_mirna_v18.fa` and `human_hg18_28_species.fa` with all human miRNAs and with the aligned coding sequences of all human genes are available at `http://lcpt.epfl.ch` for more detailed tests.

   Results published in Ref. [1] were generated by using these two files as input files in the preceding two examples.

References

[1] R. M. Marín, M. Šulc, J. Vaníček, *RNA* accepted (2012).