

SplamiR – Prediction of spliced miRNAs in plants

Christoph J. Thieme, Lydia Gramzow, Dajana Lobbes and Günter Theißen*

Friedrich Schiller University Jena, Department of Genetics, Philosophenweg 12, 07743 Jena, Germany

* guenter.theissen@uni-jena.de

1. Contents of the SplamiR package:	2
2. Installation Instructions & Requirements for Execution	2
3. Execution	2
3.1. Phase 1: Creating the database of complementary sequence pairs	2
3.2. Phase 2: Searching with a Target Gene	4
4. Output	5
5. Requirements for Recompilation	5
6. Known Issues	6

1. Contents of the SplamiR package:

This package includes the SplamiR executables, the full source code as well as examples used for the evaluation of the pipeline. You will find:

generateDatabase.class	Tool for the generation of the database of complementary sequence pairs
generateDatabase.java	Source file
searchMiRNA.class	Tool for searching the generated database with a target mRNA
searchMiRNA.java	Source file
jargs/	
GeneSplicer/	Binaries and training data sets for GeneSplicer
RNAblast/	Pre-compiled G:U-enables BLAST versions
data/osa/	Genome sequence data for <i>Oryza sativa</i>
data/zma/	Genome sequence data for <i>Zea mays</i>
README.pdf	This file

2. Installation Instructions & Requirements for Execution

SplamiR is developed for Linux platforms. All tools are checked to work in this environment. Although the Java Virtual Machine was used, porting SplamiR to other operating systems might result in additional effort (e.g. adaptation of system calls or setting-up path delimiters).

All necessary files for running SplamiR are included in this package and no recompilation is required. It is necessary to have GeneSplicer and the G:U-enabled BLAST version in the same folder where the Java programs, generateDatabase and searchMiRNA, are located.

For running SplamiR you will need:

1. Oracle/SUN Java SE Runtime Environment (tested on build 1.6.0+)
2. For using the miR-abela classifier a fast internet connection is obligatory.

3. Execution

SplamiR consists of two independent phases, the generation of a database of complementary sequence pairs and the scanning by means of a target gene.

3.1. Phase 1: Creating the database of complementary sequence pairs

In order not to run out of memory we recommend using "java -Xms512M -Xmx1024M". The set-up of the database is easily parallelizable. On multicore computers many jobs are executable at the same time by running several tasks using different parameters.

The syntax for executing the database procedure of SplamiR is:

```
java -Xms512M -Xmx1024M generateDatabase [-d,--output-dir] [-i,--input-sequence] [{-e,--e-value}] [{-w,--w-value}] [{-l,--fragment-length}] [{-o,--fragment-overlap}] [{-n,--minHitLength}] [{-m,--maxHitLength}]
```

Mandatory parameters:

output-dir: directory to store the database
input-sequence: input FASTA file

Optional parameters:

-e e-value: BLAST threshold (default: 10.0)
-w w-value: BLAST initial word length (default: 9)
-l fragment-length: size of the fragments, limits pri-u-miRNA length (default: 30,000)
-o fragment-overlap: share of successive fragments (default: 10,000)
-n minHitLength: min length of fragments to be stored in database (default: 16)
-m maxHitLength: max length of fragments to be stored in database (default: 500)

A minimum example configuration is given below. The necessary file (chr01.con) is provided in the file data_osa.zip.

```
java -Xms512M -Xmx1024M generateDatabase -d ./osa-database_chr1 -i ./data/osa/chr01.con
```

The file ./data/osa/chr01.con is processed and the resulting database is written to ./osa-database_chr1/ This produces the subfolders

fragments
blastResults

as well as the database files

chr01.plain //name derived from input file
chr01_database.fasta
chr01_database.fasta.nin
chr01_database.fasta.nhr
chr01_database.fasta.nsq

Additionally to the command-line log and the generated output files, the formatdb.log might be of interest.

3.2. Phase 2: Searching with a Target Gene

In contrast to the generation of the database, it is not recommended to run several instances of phase 2 in parallel. Otherwise the miR-abela server may get overloaded.

The command to execute the search step is as follows

```
java -Xms512M -Xmx1024M searchMiRNA [-d,--output-dir] [-i,--database-file] [-t,--target-sequence] [-s,--GeneSplicer-data] [{-e,--e-value}] [{-w,--w-value}] [{-x,--fiveprimeExtension}] [{-y,--threeprimeExtension}] [{-n,--min-miRNA-length}] [{-m,--max-miRNA-length}]
```

Mandatory parameters:

- d output-dir: directory to store the results
- i database-file: location of blast database file (.nin|.nsg|.nhr)
- t target-sequence: target FASTA file
- s GeneSplicer-data: provided training set (e.g. rice)

Optional parameters:

- e e-value: BLAST threshold (default: 10.0)
- w w-value: BLAST initial word length (default: 9)
- x fiveprimeExtension: number of genomic nt added to the start of precursor (default: 65)
- y threeprimeExtension: number of genomic nt added to the end of precursor (default: 65)
- n min-miRNA-length: minimum size of size of precursor after removal of intron (default: 16)
- m max-miRNA-length: maximum size of size of precursor after removal of intron (default: 1000)

A minimum example configuration is given below. The necessary file (osmads23_utr+protCoding.txt) is provided in the file data_osa.zip.

```
java searchMiRNA -d osa-chr1_23 -i ./osa-database_chr1/chr01_database.fasta.nhr -t data/osa-targets/osmads23_utr+protCoding.txt -s rice
```

Output in ./osa-chr1_23/:

pri-u-miRNAs/	
splicevariants/	
classification/	
databasematch.blast	//results file when BLASTing target to database
results.csv	//final results of miRNA candidates

For the input target file a new sub folder is created, if not present. All intermediate data and final results are stored in this folder. If data is present the corresponding processing step is skipped. In this case, please remove all previous results prior to re-running the calculations. The miR-abela scanning results are given as html output as fetched from the web server. The final results are collected and stored as a csv file (results.csv, tab delimited values) in the specified output folder.

4. Output

The file results.csv lists all candidate pre-miRNAs and can be easily imported into spreadsheet programs. The following information is listed in this file:

Table header	Data
Folder & Filename	Name of miR-abela output file
5' Arm 5' End	Position of the identified complementary sequence pair in reference to the DNA strand for which the sequence is given in the genome files Please note: In case the direction is "RC" (see below) the 5'- and 3'-Arms have to be swapped as well as the 5'- and 3'-Ends have to be swapped
5' Arm 3' End	
3' Arm 5' End	
3' Arm 3' End	
Length in DB	The length of the corresponding complementary sequence pair
BLAST DB Score	E-value of the GU-BLAST search which identified the two sequences of the corresponding complementary sequence pair
Pri-5' End	Position of the pri ^U -miRNA in reference to the DNA strand for which the sequence is given in the genome files Please note: In case the direction is "RC" (see below) the 5'- and 3'-Ends have to be swapped
Pri-3' End	
BLAST Target Score	E-value of the GU-BLAST search of the target mRNA which identified the corresponding complementary sequence pair
Direction	"RC" denotes a candidate pre-miRNA which is encoded by the complementary strand of DNA in reference to the strand for which the sequence is given in the genome files
Intron Start	Start and end of the predicted intron in reference to the DNA strand for which the sequence is given in the genome files Please note: In case the direction is "RC" (see above) start and end have to be swapped
Intron End	
MiR-abela Stem Start MiR-abela Stem End	Location of miRNA as identified by miR-abela – miR-abela may predict more than one robust stem structure for a given sequence, each is given in a separate row in the output file
Score	miR-abela score for candidate pre-miRNA
Length	Length of pre-miRNA as predicted by miR-abela
Energy	Energy of pre-miRNA as predicted by miR-abela
Sequence	Sequence of the pre-miRNA as predicted by miR-abela
Structure	Structure of the pre-miRNA as predicted by miR-abela

5. Requirements for Recompile

When performing changes to the source code, additionally, the Oracle/SUN Java development kit ("JDK") is necessary to compile the Java source code to byte code.

For recompiling BLAST it is necessary to obtain the (current) source files package and to replace the blast_stat.c in the folder "ncbi/algoblast/core/". The GNU build system is required

due to providing the make tools and the gcc compiler. Following the readme, the toolkit is compiled using the command `./ncbi/make/makedis.csh` from the root directory. The generated executables will be located in `ncbi/build`. Details for the compilation are given in the original BLAST source readme file.

To avoid any interference with original BLAST version the three G:U-enabled versions are renamed: `bl2seq` to `bl2rna`, `blastall` to `blastallrna` and `formatdb` to `formatdbrna`. If any of these files is missing the integrity check will block the execution of the prediction pipelining program.

6. Known Issues

Java network problems

"Unable to execute: Network is unreachable" refers to a common bug for some Linux systems. The workaround is as follows:

- Modify in `/etc/sysctl.d/bindv6only.conf` the setting from `net.ipv6.bindv6only = 1` to `net.ipv6.bindv6only = 0`.
- Reset kernel variables: `sudo /etc/init.d/procps restart`

Only 1 fragment is generated

Please use the Java environment provided by Oracle/SUN.