siRNA off-target Discovery Pipeline v1.0: User manual

Ferhat Alkan [1,2], Anne Wenzel [1,2], Oana Palasca [1,2,3], and Jan Gorodkin [1,2,*]

[1]Center for non-coding RNA in Technology and Health,

[2]Department of Veterinary Clinical and Animal Science,

University of Copenhagen, Grønnegårdsvej 3, 1870 Frederiksberg C, Denmark,

[3]Novo Nordisk Foundation Center for Protein Research,

University of Copenhagen, Blegdamsvej 3B, 2200 Copenhagen N, Denmark,

January 16, 2017

Contents

*To whom correspondence should be addressed. Tel: +45 353 33578; Fax: +45 353 34704; Email: gorodkin@rth.dk

WARNING

Please read the whole document before running any off-target predictions with this pipeline.

1 Introduction

This siRNA off-target discovery pipeline has been designed to predict individual off-targets of a given siRNA together with its overall off-targeting potential. It is an application of `RIsearch2`, an RNA–RNA interaction prediction tool. The pipeline combines `RIsearch2` interaction predictions with accessibility and abundance information of predicted binding sites to increase its accuracy to predict the true off-targets. Before running the pipeline, please read the complete manual.

2 Pre-computations

In order to make the pipeline more flexible and efficient, it is designed in a way that several preprocessing steps must be completed before running the pipeline itself. Prior to the pre-computations it is highly recommended to create a workspace directory for the pipeline and carry out all the computations within this workspace.

```
$ mkdir /path/to/your/workspace/
```

2.1 Pre-computing the `RIsearch2` interaction predictions

For a given siRNA, interaction predictions between the siRNA and target sequences (could be whole genome/transcriptome, set of transcript sequences, etc.) must be completed with `RIsearch2` prior to the execution of the pipeline. Note that in order to compute the overall off-targeting potential of a given siRNA, it is highly recommended to obtain `RIsearch2` predictions by screening the siRNA against the whole genome (or transcriptome). In this run, `RIsearch2` parameters can be specified by the user, however, `-p1` option of `RIsearch2` is not allowed. `RIsearch2` off-target interaction predictions must be reported either with its default reporting option or `-p2` extended option. (This is mainly because the pipeline requires "interaction per line" output.) To learn more about how to run and decide parameters for `RIsearch2`, please read the `RIsearch2` user manual. Recommended parameter settings for siRNA off-target discovery are as follows: `-s 1:12/6 -e -10 -l 20`.

```
$ cd /path/to/your/workspace/
$ mkdir RIsearch2_results
$ cd RIsearch2_results/
$ risearch2.x -c /input_path/target.fa -o target.pksuf
$ risearch2.x -q /input_path/sirna.fa -i target.pksuf -s 1:12/6 -e -10 -l 20
```

2.2 Precomputing the RNAplfold opening energies

This is an optional pre-computation. Nonetheless, it is strongly recommended. In order to take the opening energies of predicted binding sites into account, these energies must be pre-computed with the `run_RNAplfold_and_pack_results.py` script that is part of this code package. To run this script, RNAplfold from ViennaRNA package (https://www.tbi.univie.ac.at/RNA/) must already be installed

and running on your system. When running this script, RNAplfold parameters `W` and `P` must be given as parameters. In case you would like to compute opening energies for the reverse strand of your target sequences, strand information also needs to be given.

```
$ cd /path/to/your/workspace/
$ mkdir accessibility_results
$ cd accessibility_results/
$ /path/to/pipeline_package/src/run_RNAplfold_and_pack_results.py \
/input_path/target.fa 80 40 +
$ /path/to/pipeline_package/src/run_RNAplfold_and_pack_results.py \
/input_path/target.fa 80 40 -
```

3 Executing the pipeline

You can always access the help page of the siRNA off-target discovery pipeline by giving `-h` or `--help` parameter during execution. After all pre-computations are done, execution requires several essential parameter settings together with plenty of other optional parameters. In the following, we explain when they are needed and what they are useful for.

```
$ /path/to/pipeline_package/src/pipeline.py -h
$ /path/to/pipeline_package/src/pipeline.py --help
```

3.1 siRNA–target interaction predictions (`-r <file>, -type <gw/tw>`)

siRNA–target interaction predictions precomputed by `RIsearch2` are fed into the pipeline by the `-r` parameter. Since overall off-targeting potential measure is computed with these interactions, it is very important that these predictions have been made on genome- or transcriptome-wide level. When running the pipeline, you need to set up the `-type` to "gw" or "tw", depending on your predictions, whether it is genome- or transcriptome-wide respectively. Note that `-r` is a mandatory parameter and the default for `-type` is "gw".

```
$ ... -r /path/to/your/workspace/RIsearch2_results/risearch_siRNAID.out.gz ...
$ ... -r /path/to/your/workspace/RIsearch2_results/risearch_siRNAID.out.gz -type tw ...
```

3.2 Transcriptome intersection (`-t <file>`, `-feature <str>`, `-expmetric <str>`)

Transcriptomic data providing the abundance estimates of the predicted binding sites is passed into the pipeline with `-t` parameter. Note that accepted file formats for transcriptomic data are BED, GFF and GTF or their gzipped versions. If a BED file is used, expression abundance levels must be passed within the score field. If GFF or GTF files are used, `-feature` parameter needs to be set to either *exon*, *transcript* or *gene*. On default settings, this is set to *exon*. This setting determines which lines are actually read from the given GFF or GTF file and determines if the expression levels are exon, transcript or gene specific. The last parameter `-expmetric` determines the expression unit to be read from GFF or GTF files. On default, it is set to *FPKM*. Depending on your input data it is crucial to set up these two `-feature` and `-expmetric` parameters. Besides that, if *transcript_id* or *gene_id* fields are not set within the given GTF or GFF file, or name field within the BED file, transcript (gene) specific off-targeting

probabilities cannot be computed. Parameter `-t` is an optional parameter and in case it is not set, the pipeline assigns an expression level of 1 to every predicted binding site.

```
$ ... -t /input_path/expression_data.bed ...
$ ... -t /input_path/expression_data.gff2 ...
$ ... -t /input_path/expression_data.gtf.gz ...
$ ... -t /input_path/expression_data.gtf -feature gene ...
$ ... -t /input_path/expression_data.gff.gz -feature transcript -expmetric RPKM ...
```

3.3 Passing the precomputed opening energies (`-p <path>`)

Path to the folder that contains precomputed accessibility files must be passed to the pipeline with `-p` parameter. Unless it is set, the pipeline does not take accessibilities into account. When set, the pipeline makes two computations, with and without accessibility information, and reports results regarding both cases.

```
$ ... -p /path/to/your/workspace/accessibility_results/ ...
```

3.4 Information of the given siRNA (`-q <str>, -os <file>`)

You have to pass an siRNA ID and sense-strand sequence file (FASTA-format) to the pipeline for the siRNA you are running the pipeline for. This siRNA ID, given with `-q` parameter, must be consistent with the sequence ID present in the FASTA file, given by `-os` parameter. However, this FASTA file can contain more than one sequence as long as it contains the sequence with given siRNA ID. These are mandatory parameters.

```
$ ... -q siRNAID -os /input_path/sirna.fa ...
```

3.5 Reporting results (`-o <file>, --offPs`)

Parameter `-o` is a mandatory parameter. A path to an output file must be given in order to report the overall off-targeting potential of the given siRNA. Additionally, if you would also like to get the chromosome/gene/transcript-specific off-targeting probabilities for this siRNA, you need to execute the program by also passing the `--offPs` parameter.

```
$ ... -o siRNA_off_targets_output.txt ...
$ ... -o siRNA_off_targets_output.txt --offPs ...
```

3.6 `RIsearch2` executable (`-rx <command>`)

In case RIsearch2 is not executable with the default `risearch2.x` command, you have to pass the RIsearch2 executable to the pipeline to be able to compute the minimum hybridization energy of the given siRNA.

```
$ ... -rx /path/to/RIsearch2_executable ...
```

3.7 `--sort` parameter

When computing the off-targeting potential on genome-wide level, it is highly recommended to pass the `--sort` parameter to the pipeline, so that pipeline can sort the RIsearch2 target prediction file which consequently makes the pipeline faster due to more structured reading of the opening energies.

```
$ ... --sort ...
```

3.8 α and γ parameters (`-alpha <float>, -gamma <float>`)

If you would like to change the α and γ parameters for partition function computation, you can pass them with `-alpha` and `-gamma` parameters. These parameters are used when detecting obvious off-targets in order to unify their hybridization energies. Please read the manuscript to understand it better. Note that multiple values are allowed if seperated by "`;`", and in this case, results for all the combinations of α and γ where $\alpha \leq \gamma$ will be reported. Default settings are equal to passing `-alpha 0.8;1 -gamma 0.8;1`.

```
$ ... -alpha 0.6;0.7;1 -gamma 0.7;1 ...
```

3.9 Specifying the on-target

Since off-targeting potential is also based on interactions of the siRNA with its on-target transcript, on-target transcript information must be fed into the pipeline in one of the following three formats.

3.9.1 By ID or genomic location (`-oi <str>, -on <chr;sp;ep;str>`)

It is most likely the case that the on-target transcript/gene of the given siRNA is part of the input target sequence, and, abundance information of this transcript is present in the given transcriptome data. Also, since this target file is used in siRNA–target predictions by RIsearch2, possible siRNA–on-target interactions (including intended perfect(near-perfect)-complementary interaction) are already fed into the pipeline. Therefore, the on-target needs to be given to the pipeline to differentiate on-target interactions from off-targets. This is simply done by providing the ID or genomic location of the on-target. Note that in order to use the ID option, transcriptome data fed into the pipeline has to contain this on-target ID.

```
$ ... -oi ENSG000XXXXXX ...
$ ... -on "chrX;1234567;3456789;+" ...
$ ... -oi ENST000XXXXXX -on "chrX;1234567;3456789;+" ...
```

3.9.2 By giving the on-target files (`-of <file>`, `-rp <s,X;l,X;e,X>`, `-ap <L,X;W,X>`, `-oexp <float>`)

If the on-target transcript is not part of the target sequence used in RIsearch2 pre-computations or transcriptomic data fed into the pipeline, the on-target sequence must be fed into the pipeline in FASTA format. Our default parameter settings for RIsearch2 and RNAplfold can also be fed into the pipeline with `-rp` and `-ap` parameters. In addition, abundance estimate of the on-target needs to be set with `-oexp` parameter.

```
$ ... -of /input_path/on_target.fa -rp "s,1:12/6;l,20;e,-10" -ap "L,40;W,80" -oexp 123 ...
```

3.9.3 By precomputed on-target data (`-op <file>, -oa <file>, -oexp2 <float>`)

In case you have already pre-computed the on-target interactions with RIsearch2 and accessibility info for the on-target transcript with `run_RNAplfold_and_pack_results.py` script, you can pass them into the pipeline with `-op, -oa` parameters. Note that the abundance estimate of the on-target still needs to be set with `-oexp2` parameter.

```
$ ... -op risearch_onTarget.out.gz -oa onTarget.acc.bin -oexp2 123 ...
```

3.10 Other experimental parameters

There are also a few experimental parameters that might come in handy when parallelizing your workflow or filtering target predictions of the siRNA.

Filtering interactions (`--less`, `-chr <chrID>`, `-loc <chrID;spos;epos>`, `-thr <float>`, `-tp <float>`, `--thrAfterOpEn`)

If you would like to filter some of the target interactions predicted on the given target sequence by RIsearch2, these parameter settings are what you need. First of all, in case you have performed transcriptome-wide predictions with RIsearch2, you might want to filter out the interactions predicted by RIsearch2 on the antisense strand of the given transcripts. This can be achieved by passing the `--less` parameter. If you would like to filter out the interactions based on genomic location, you can set the `-chr` and `-loc` parameters. To filter the interactions based on hybridization energy or free energy of the interaction you have to use one or more of the `-thr`, `-tp` and `--thrAfterOpEn` parameters. `-thr` sets a fixed energy threshold, `-tp` sets this threshold as some percentage of the minimum hybridization energy, achieved by perfect complementary interaction of the siRNA, and `--thrAfterOpEn` switches the whole approach from hybridization energy to free energy (hybridization + opening energy).

Save intermediate files (`-sa <file>, --intersection`)

If you would like to save some of the intermediate files generated by the pipeline, you have to use the `-sa` and `--intersection` parameters. If you would like to save the opening energies of the expressed binding sites you have to specify a file path with `-sa <file>` option. And, to save the intersection file that contains the expressed RIsearch2 binding sites, you have to pass the `--intersection` parameter when running the pipeline.

Experimental θ parameter (`-theta <float>`)

This parameter setting is experimental and it replaces the computation with α and γ parameters when detecting obvious off-targets. When set, all hybridization energies are adjusted as follows.

$$E' = ((E + 10) * \theta) - 10$$

4 Example runs

Please read the whole document before running any off-target prediction. Here we present two examples; genome-wide and transcriptome-wide prediction of off-targets, that you can easily run with the dummy data provided by the code package.

In the following, you can find the commands in order, that we encourage you to run, as an example genome-wide off-target prediction for the dummy siRNA on the dummy genome, with the dummy transcriptome data, all part of the code package. Note that RIsearch2 and RNAplfold must already be installed and running on your system to be able to run some of these commands.

```
$ cd test_suite/
$ mkdir gw_test
$ cd gw_test
$ mkdir RIsearch2_results
$ cd RIsearch2_results
$ risearch2.x -c ../../genome.fa -o genome.pksuf --verbose
$ risearch2.x -q ../../sirna.fa -i genome.pksuf -s 1:12/6 -e -10 -l 20 --verbose
$ cd ..
$ mkdir accessibility_results
$ cd accessibility_results
$ ../../../src/run_RNAplfold_and_pack_results.py ../../genome.fa 80 40 +
$ ../../../src/run_RNAplfold_and_pack_results.py ../../genome.fa 80 40 -
$ cd ..
$ ../../src/pipeline.py -r RIsearch2_results/risearch_siRNAID.out.gz \
  -t ../expression_data.gtf -feature transcript -expmetric RPKM \
  -p accessibility_results/ -q siRNAID -os ../sirna.fa \
  -o sirna_gw_off_targets.txt --offPs -oi onTarget --sort
```

In the following, you can find an example run for transcriptome-wide off-target prediction. Note that `--less` and `-type tw` parameters are crucially important here. We encourage you to try out running this transcriptome-wide off-target prediction by executing the commands given in order.

```
$ mkdir tw_test
$ cd tw_test
$ mkdir RIsearch2_results
$ cd RIsearch2_results
$ risearch2.x -c ../../transcripts.fa -o transcripts.pksuf --verbose
$ risearch2.x -q ../../sirna.fa -i transcripts.pksuf -s 1:12/6 -e -10 -l 20 --verbose
$ cd ..
$ mkdir accessibility_results
$ cd accessibility_results/
$ ../../../src/run_RNAplfold_and_pack_results.py ../../transcripts.fa 80 40 +
$ cd ..
$ ../../src/pipeline.py -r RIsearch2_results/risearch_siRNAID.out.gz --less \
  -type tw -t ../expression_data.bed -p accessibility_results/ -q siRNAID \
  -os ../sirna.fa -o sirna_tw_off_targets.txt --offPs \
  -of ../onTarget.fa -oexp 1000
```

Same transcriptome-wide off-target prediction can also be run with precomputed onTarget interaction predictions and accessibility info. In that case, you should be able to run the following command.

```
$ .../../src/pipeline.py -r RIsearch2_results/risearch_siRNAID.out.gz --less \
  -type tw -t ../expression_data.bed -p accessibility_results/ -q siRNAID \
  -os ../sirna.fa -o sirna_tw_off_targets_2.txt --offPs \
  -op risearch_siRNAID_only.out.gz -oa onTarget.open.acc.bin -oexp2 1000
```