genome.gov
National Human Genome Research Institute
National Institutes of Health

User Guide

For the best print results, we recommend you download the acompanying User Guide PDF.

Requirements

SubmiRine is composed of two modules:

- ***SubmiRine_Search***: an open-source Python package, which is distributed with a modified version of the miRmap (Vejnar et al. 2012) code base and a copy of the py-burrows-wheeler package. The *SubmiRine_Search* module was designed to be run with Python version 2.7. It may be compatible with other versions of Python, but no other version has been tested. Additionally, *SubmiRine_Search* requires the SeqIO module within the BioPython framework, and this package must be recognized by the Python include path (environment variable `PYTHONPATH`). For example if you install the Bio module directory in /usr/bin/Bio/, run the following shell command (Bash) prior to running SubmiRine:

```
> export PYTHONPATH=${PYTHONPATH}:/usr/bin/Bio
```

- ***SubmiRine_Compare***: is a standalone R script, and should be compatible with a standard R installation.

The memory requirements of SubmiRine are dependent upon the size of the data set(s) being analyzed, with *SubmiRine_Search* representing the main bottleneck. *SubmiRine_Search* stores all target sites identified in memory so that target sites can be compared across 3'UTR alleles. Thus, the number of target sites identified is the main factor in RAM usage, particularly with larger data sets. While the exact memory requirements will vary depending upon characteristics of the input data, we have provided a function to compute the expected memory usage of a particular *SubmiRine_Search* job based upon the size of the input data sets and runtime arguments. Based on the data sets we have used to test SubmiRine, the RAM usage is roughly 3-5 KB per expected target site, where the number of expected target sites, $E[|\mathrm{T}|]$, can be computed by:

$$E[|\mathrm{T}|] = \frac{M}{4^S} \cdot [D - U(S-1)]$$

where M is the number of input miRNA sequences, S is the minimum target site seed length, D is the amount of DNA sequence (in bases) from the input 3'UTR sequences, and U is the number of UTR records. From our experience, the pre-processed COPD clinical genomic data set presented in the SubmiRine manuscript contained 93 3'UTRs in the clinically relevant (test) model and 1608 3'UTRs in the background model, and each used the same set of input miRNAs that were expressed above 10 reads per million (RPM), representing 418 miRNAs. These two models required 264 MB and 2.4 GB RAM, respectively.

Usage

Input File Formats

3'UTR Sequences (FASTA file)

```
>UNIQUE_GENE_IDENTIFIER_1|other_information_allele1_1:C   Expression_Value
CAACACACAGGACATCAGCA
CACAGTTTTT

>UNIQUE_GENE_IDENTIFIER_1|other_information_allele2_1:A   Expression_Value
AAACACACAGGACATCAGCA
CACAGTTTTT

>UNIQUE_GENE_IDENTIFIER_2|other_information_allele1_24-25:deletion   Expression_Value
TCGTCGAACACAGTACGCTA
ACA--GTTAGTTA
```

```
>UNIQUE_GENE_IDENTIFIER_2|other_information_allele2_24-25:GG   Expression_Value
TCGTCGAACACAGTACGCTA
ACAGGGTTAGTTA
...
```

Important Notes:

- Different alleles of the same 3'UTR are combined according to the `UNIQUE_GENE_IDENTIFIER` key immediately following the `>` character, and immediately followed by a pipe (`|`) character. All information following the first pipe character is auxilary information and will be ignored in the file parse.
- The defline should have exactly two fields, separated by white space. The first field defines the UTR sequence information, and the second field contains a numeric value representing the expression of the 3'UTR. Thus, **the sequence information in the defline (field #1) should not contain white space**.
- Different alleles of the same 3'UTR must be aligned. Therefore in/dels should be denoted with '`-`' characters.

MicroRNA Mature Sequences (FASTA file)

```
>MicroRNA_identifier_1   Expression_Value
ACTACAGTGGACAGAGTTTACT

>MicroRNA_identifier_2   Expression_Value
CGTCACATACGGGACATATAGG
...
```

Important Notes:

- The defline should have exactly two fields, separated by white space. The first field defines the miRNA sequence information, and the second field contains a numeric value representing the expression of the miRNA. Thus, **the sequence information in the defline (field #1) should not contain white space**.

Command Line

Standard Workflow (Test v. Background Model)

Ex. Identify and score miR-TSVs in `test_UTRs.fa` relative to the background model `background_UTRs.fa`, using all miRNAs in `miRNAs.fa`. All canonical miRNA target sites will be considered, therefore having minimum seed length = 6.

```
> ./SubmiRine_Search.py -u test_UTRs.fa -m miRNAs.fa -s 6 -o test_output_dir
> ./SubmiRine_Search.py -u background_UTRs.fa -m miRNAs.fa -s 6 -o bg_output_dir
> Rscript SubmiRine_Compare.R test_output_dir bg_output_dir
```

Target Prediction-Only Runs

Ex. Generate all target site predictions on `test_UTRs.fa` using miRNAs from `miRNAs.fa`. This is called "one allele" mode in SubmiRine because it assumes each 3'UTR is represented only once, therefore not predicting miR-TSVs. This will generate both raw TargetScan6 context+ scores ("binary scores") and miRNA-abundance weighted scores ("empirical scores").

```
> ./SubmiRine_Search.py -u test_UTRs.fa -m miRNAs.fa -s 6 -1 -o test_output_dir
```

Output

SubmiRine_Search generates an output directory containing seven output files. The output below is based on the example data in the code directory `/data/examples/`.

- **sitewise_all_targets.tsv:** A dump of all target sites identified. Each target is given a unique Target_ID, and different copies of the same target site on different 3'UTR alleles are assigned to the same Target_Group.

Target_ID	Target_Group	MicroRNA	MicroRNA_Seq	UTR	UTR_Seq	UTR_Aligned_Index	Mer_Type	Binary_Score	Empirical_Score
5	1	hsa-let-7a-5p	GAGGUAG	NM_001004317|...site1_C	CUACCUC	2480	7mer-m8	-0.0750561225124	-0.508880510634
12	1	hsa-let-7a-5p	GAGGUAG	NM_001004317|...site1_G	UACCUC	2483	6mer	0.0228096044831	0.154649118396

- **sitewise_uniq_targets.tsv:** A subset of rows from sitewise_all_targets.tsv representing only targets from Target_Groups that have allele-specific scoring differences. (In this case, same as above).

Target_ID	Target_Group	MicroRNA	MicroRNA_Seq	UTR	UTR_Seq	UTR_Aligned_Index	Mer_Type	Binary_Score	Empirical_Score
5	1	hsa-let-7a-5p	GAGGUAG	NM_001004317|...site1_C	CUACCUC	2480	7mer-m8	-0.0750561225124	-0.508880510634
12	1	hsa-let-7a-5p	GAGGUAG	NM_001004317|...site1_G	UACCUC	2483	6mer	0.0228096044831	0.154649118396

- **sitewise_deltas.tsv:** For each Target_Group with allele-specific scoring differences, this file contains information about the scoring differences. One record will be presented for each 3'UTR group, and the deltas will be computed between the highest and lowest scoring target sites within the group. Also included are the total binary and empirical scores for all target sites on the UTR.

Target_Group	MicroRNA	UTR	Delta_Binary	Min_Binary	UTR_Binary	Delta_Empirical	Min_Empirical	UTR_Empirical
1	hsa-let-7a-5p	NM_001004317|...site1_C	-0.0978657269955	-0.0750561225124	-0.811300772265	-0.663529629029	-0.508880510634	-5.50061923596

- **mirwise_all_targets.tsv:** The "mirwise" files are treated the same as the "sitewise" files, except that all targets of the same miRNA on the same 3'UTR allele are grouped (scores summed). In the example case, there are five total let-7a sites on this UTR, one of which is detailed above in the "sitewise" files.

MicroRNA	UTR	Target_Groups	Binary_Score	Empirical_Score
hsa-let-7a-5p	NM_001004317|...site1_C|site2_C	1,2,3,4,5	-0.90916649926	-6.16414886499
hsa-let-7a-5p	NM_001004317|...site1_G|site2_G	1,2,3,4,5	-0.368366371297	-2.49752399739

- **mirwise_uniq_targets.tsv:** Similar to the sitewise_uniq_targets file, this produces the subset of records for which a scoring difference occurs.

MicroRNA	UTR	Target_Groups	Binary_Score	Empirical_Score
hsa-let-7a-5p	NM_001004317|...site1_C|site2_C	1,2,3,4,5	-0.90916649926	-6.16414886499
hsa-let-7a-5p	NM_001004317|...site1_G|site2_G	1,2,3,4,5	-0.368366371297	-2.49752399739

- **mirwise_deltas.tsv:** Similar to the sitewise_deltas file, this indicates the delta scores.

MicroRNA	UTR	Delta_Binary	Min_Binary	UTR_Binary	Delta_Empirical	Min_Empirical	UTR_Empirical
hsa-let-7a-5p	NM_001004317|...site1_C|site2_C	-0.540800127964	-0.90916649926	-0.368366371297	-3.66662486759	-6.16414886499	-2.49752399739

- **UTR_summaries.tsv:** Provides a summary of the total set of miR-TSV predictions for each 3'UTR, and highlights the Unique_Targets on each, in order by delta score. The UTR_Rank and Rank_Score are not currently implemented.

UTR_Rank	Rank_Score	UTR	Expression	Total_Binary	Total_Empirical	Unique_Targets
.	.	NM_001004317|...site1_C|site2_C	4.510000	-0.90916649926	-6.16414886499	3,5
.	.	NM_001004317|...site1_G|site2_G	5.510000	-0.368366371297	-2.49752399739	12

SubmiRine_Compare assumes that *SubmiRine_Search* has been run twice on a clinically relevant "Test Model" and a separate "Background Model". As input, it accepts a path to the output

directories for each of these runs, and reads the data from the files described above for each model. *SubmiRine_Compare* generates two output files that, by default, are placed in the output directory of the "Test Model".

- **sitewise_scored.tsv:** Contains a list of candidate miR-TSVs, their underlying binary, empirical, Δbinary, and Δempirical scores, the six empirical probabilities (P_1-P_6), and the SubmiRine SLP score. The records are sorted by SLP score, with the highest scoring (most negative) first.
- **mirwise_scored.tsv:** Same as sitewise_scored, except that miR-TSVs are handled in the "mirwise" fashion.

Reproducing Results from the SubmiRine Paper

In the examples below, `SUBMIRINEPATH` represents the location of the SubmiRine/ base directory where SubmiRine is installed.

1. Test COPD Data Set:
```
> /SUBMIRINEPATH/src/SubmiRine_Search.py -u /SUBMIRINEPATH/data/COPD/background_UTRs.fa -m /SUBMIRINEPATH/data/COPD/mirs_gt10rpm.fa -l 2 -o
/SUBMIRINEPATH/data/COPD/bg_model
> /SUBMIRINEPATH/src/SubmiRine_Search.py -u /SUBMIRINEPATH/data/COPD/COPD_UTRs.fa -m /SUBMIRINEPATH/data/COPD/mirs_gt10rpm.fa -l 2 -o
/SUBMIRINEPATH/data/COPD/test_model
> Rscript /SUBMIRINEPATH/src/SubmiRine_Compare.R /SUBMIRINEPATH/data/COPD/test_model /SUBMIRINEPATH/data/COPD/bg_model
```
2. Test Known SNPs:
```
> /SUBMIRINEPATH/src/SubmiRine_Search.py -u /SUBMIRINEPATH/data/known_snps/background_UTRs.fa -m
/SUBMIRINEPATH/data/known_snps/COPD_miRs_plus_simulated.fa -l 2 -o /SUBMIRINEPATH/data/known_snps/bg_model
> /SUBMIRINEPATH/src/SubmiRine_Search.py -u /SUBMIRINEPATH/data/known_snps/known_snp_UTRs.fa -m
/SUBMIRINEPATH/data/known_snps/COPD_miRs_plus_simulated.fa -l 2 -o /SUBMIRINEPATH/data/known_snps/test_model
> Rscript /SUBMIRINEPATH/src/SubmiRine_Compare.R /SUBMIRINEPATH/data/known_snps/test_model /SUBMIRINEPATH/data/known_snps/bg_model
```

Last Modified: Monday, 02-Feb-2015 14:15:37 EST