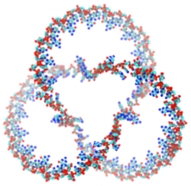
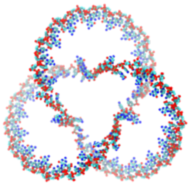


TargetRNA3



User Manual

TargetRNA3 uses machine learning to identify targets of small regulatory RNAs (sRNAs) throughout a genome. At its core, TargetRNA3 employs a gradient boosting classification algorithm that has been trained on thousands of evinced interactions between sRNAs and their regulatory targets in various prokaryotes. When making target identifications, the machine learning algorithm uses a variety of features indicative of regulatory interactions, including the thermodynamics of the interaction and potential homologous interactions in related organisms, if available.

The quickest and easiest way to use TargetRNA3 is via the webserver:
https://cs.wellesley.edu/~btjaden/TargetRNA3

However, some users may prefer to use TargetRNA3 on their own machine. This user manual provides details on downloading TargetRNA3 and using it locally.

Table of Contents

REQUIREMENTS

Source code and data files for TargetRNA3 are available from GitHub (shown in Table 1):
https://github.com/btjaden/TargetRNA3

TargetRNA3 is written in Python and requires at least version 3 of Python. Beyond the Python standard library, TargetRNA3 uses three libraries that are available from the Python Package Index:

- `numpy`
- `pandas`
- `scipy`

TargetRNA3 uses three applications from the BLAST+ suite provided by NCBI:

- `blastn`
- `blastp`
- `blastdb_aliastool`

TargetRNA3 uses two applications from the ViennaRNA Package:

- `RNAplfold`
- `RNAplex`

The BLAST+ and ViennaRNA applications should be available to TargetRNA3 for execution in the current path, or else the executables for these applications should be in the same directory as the TargetRNA3 source code.

The contents of two further data files are necessary. Because of their larger size (8 GBs and 4 GBs), they are not provided on GitHub. The two files are located at:

- https://cs.wellesley.edu/~btjaden/TargetRNA3/DB_FNA.tar
- https://cs.wellesley.edu/~btjaden/TargetRNA3/DB_FAA.tar

These two files should be placed in the `DataFiles/` directory and their contents extracted:

```
tar xf DB_FAA.tar        tar xf DB_FNA.tar
```

Each of these two files contains 15 files within it. Once extracted, the `DataFiles/` directory should contain 45 files (15 from each of these two files and 15 from GitHub as shown in Table 1).

Table 1. Source code and data files from GitHub

File	Description
TargetRNA3.py	Source code
AddGenome.py	Source code for adding a new genome
SgrS.fa	FASTA file containing an example sRNA sequence
Prrf1.fa	FASTA file containing an example sRNA sequence
DataFiles/model.pickle	Trained machine learning model
DataFiles/gene_genome.pickle	Mapping genes to genomes
DataFiles/genome_IDs.pickle	Mapping genomes to unique identifiers
DataFiles/assembly_summary.txt.archaea.gz	Used for adding a new archaeal genome (optional)
DataFiles/assembly_summary.txt.bacteria.gz	Used for adding a new bacterial genome (optional)
DataFiles/16S.fna.ndb	BLAST database file
DataFiles/16S.fna.nhr	BLAST database file
DataFiles/16S.fna.nin	BLAST database file
DataFiles/16S.fna.njs	BLAST database file
DataFiles/16S.fna.nog	BLAST database file
DataFiles/16S.fna.nos	BLAST database file
DataFiles/16S.fna.not	BLAST database file
DataFiles/16S.fna.nsq	BLAST database file
DataFiles/16S.fna.ntf	BLAST database file
DataFiles/16S.fna.nto	BLAST database file
Genomes/GCF_000005845.2/closest_relatives.txt	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/GCF_000005845.2_ASM584v2_feature_table.txt.gz	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/GCF_000005845.2_ASM584v2_genomic.fna.gz	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/GCF_000005845.2_ASM584v2_genomic.gff.gz	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/GCF_000005845.2_ASM584v2_protein.faa.gz	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/GCF_000005845.2_ASM584v2_rna_from_genomic.fna.gz	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/mRNA.homologs	*E. coli* genome files (optional)
Genomes/GCF_000005845.2/RNAplfold_results/*	Many *E. coli* genome files (optional)
Genomes/GCF_000006765.1/closest_relatives.txt	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/GCF_000006765.1_ASM676v1_feature_table.txt.gz	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/GCF_000006765.1_ ASM676v1_genomic.fna.gz	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/GCF_000006765.1_ ASM676v1_genomic.gff.gz	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/GCF_000006765.1_ ASM676v1_protein.faa.gz	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/GCF_000006765.1_ ASM676v1_rna_from_genomic.fna.gz	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/mRNA.homologs	*P. aeruginosa* genome files (optional)
Genomes/GCF_000006765.1/RNAplfold_results/*	Many *P. aeruginosa* genome files (optional)

USAGE: TargetRNA3

As input, TargetRNA3 requires genome information and a sRNA sequence. Two example sets of genome files (`Genomes/GCF_000005845.2/` for *E. coli* and `Genomes/GCF_0000006765.1/` for *P. aeruginosa*) and two example sRNA sequences (`SgrS.fa` and `Prrf1.fa`) have been provided.

To search for targets of the sRNA SgrS in *E. coli*:

```
    python TargetRNA3.py -s SgrS.fa -g Genomes/GCG_000005845.2
```

To search for targets of the sRNA Prrf1 in *P. aeruginosa*:

```
    python TargetRNA3.py -s Prrf1.fa -g Genomes/GCG_000006765.1
```

Command line arguments are shown in Table 2.

Table 2. Command line arguments for `TargetRNA3.py`

Flag	Value	Description	Required?
-s	String	File in FASTA format containing sRNA sequence	Required
-g	String	Path to directory containing genome information including files `genomic.fna`, `protein.faa`, `rna_from_genomic.fna`, and `feature_table.txt` (files may be gzipped or not)	Required
-o	String	File to which results should be output (default is standard out)	Optional
-prob	Decimal	Probability above which a gene is identified as a target (default is 0.5)	Optional
-pval	Decimal	*P*-value below which a gene is identified as a targe (default is 0.05)	Optional
-n_threads	Integer	Number of threads (default is based on self-identification of number of processors)	Optional
-db	String	Path to BLAST database (default is `DataFiles/combined.fna`)	Optional
-model	String	Path to ML model file (default is `DataFiles/model.pickle`)	Optional
-h		Print usage and description, ignore all other flags	Optional
-help		Print usage and description, ignore all other flags	Optional

As output, TargetRNA3 produces a ranked list of candidate regulatory targets that it identifies for a sRNA. The main measure of the likelihood that a sRNA interacts with a candidate target is the *probability*. Probabilities greater than 0.5 indicate that a candidate is more likely than not to be a target, as determined by the machine learning algorithm. Probabilities less than 0.5 indicate that a candidate is more likely *not* to be a target, as determined by the machine learning algorithm.

If TargetRNA3 reports no significant targets (or too few significant targets) for a sRNA, the sensitivity can be increased so as to identify more targets by lowering the probability threshold with the `-prob` command line argument.

For identified targets of a sRNA, TargetRNA3 outputs a variety of information for each target, including the name and annotation of the target, a predicted structure and energy (based on thermodynamics of hybridization and structural accessibility of the sRNA and target) for the sRNA:mRNA target interaction, the location of the interaction within the sRNA and within the target (relative to the start of the target), the probability that there is an interaction between the sRNA and target as determined by the machine learning algorithm, and a corresponding *p*-value for the interaction.

USAGE: Adding a New Genome

Two example sets of genome files (`Genomes/GCF_000005845.2/` for *E. coli* and `Genomes/GCF_0000006765.1/` for *P. aeruginosa*) have been provided. Using `AddGenome.py`, new genomes can be added for TargetRNA3 to search. `AddGenomes.py` provides two different ways to add a new genome: (1) with the assembly accession identifier or (2) with manual download.

(1) Adding a genome with the assembly accession identifier

First, determine the assembly accession identifier for the genome of interest. For *Escherichia coli* str. K-12 substr. MG1655, the identifier is GCF_000005845.2. For *Pseudomonas aeruginosa* PAO1, the identifier is GCG_000006765.1.

For bacteria, assembly accession identifiers can be found in the file `DataFiles/assembly_summary.txt.bacteria.gz` or online at:
https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/assembly_summary.txt

For archaea, assembly accession identifiers can be found in the file `DataFiles/assembly_summary.txt.archaea.gz` or online at:
https://ftp.ncbi.nlm.nih.gov/genomes/refseq/archaea/assembly_summary.txt

To add a new genome (e.g., *Bacillus subtilis* subsp. subtilis str. 168 with assembly accession identifier GCF_000009045.1) using the assembly accession identifier:

```
python AddGenome.py -a GCF_000009045.1
```

(2) Adding a genome with manual download

In case a genome cannot be added with its assembly accession identifier, e.g., because the genome assembly is not available from RefSeq, a genome can be added manually. Four genome files are needed: the genome sequence in FASTA format (`genomic.fna`), protein sequences in FASTA format (`protein.faa`), RNA sequences in FASTA format (`rna_from_genomic.fna`), and the feature table (`feature_table.txt`). These four files can be gzipped or not.

To add a new genome (e.g., assuming the four genome files are located in a directory `Genomes/MyNewGenome`) based on the provided genome files:

```
python AddGenome.py -g Genomes/MyNewGenome
```