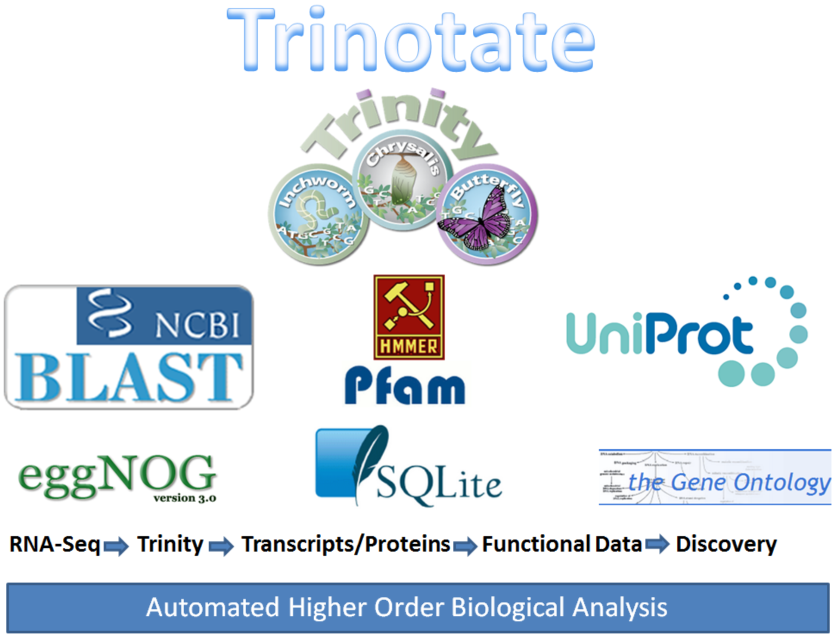
Trinotate: Transcriptome Functional Annotation and Analysis



Note

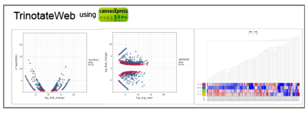
Trinotate Breaking News

- March, 2016: Trinity V3.0 is released. New features include:
- include custom protein database search results
- autoTrinotate script for automating all computations and loading the database.
- Kegg annotations are now reported.
- Make your own *fresh* boilerplate database anytime, or use a pre-generated one. Your choice.
- Lighttpd webserver support for TrinotateWeb
- Boilerplate database is now tiny compared to previous versions.
- Uniprot/Trembl/Uniref90 not integrated directly, but searched as a custom database if desired.

Background

Trinotate is a comprehensive annotation suite designed for automatic functional annotation of transcriptomes, particularly de novo assembled transcriptomes, from model or non-model organisms. Trinotate makes use of a number of different well referenced methods for functional annotation including homology search to known sequence data (BLAST+/SwissProt), protein domain identification (HMMER/PFAM), protein signal peptide and transmembrane domain prediction (signalP/tmHMM), and leveraging various annotation databases (eggNOG/GO/Kegg databases). All functional annotation data derived from the analysis of transcripts is integrated into a SQLite database which allows fast efficient searching for terms with specific qualities related to a desired scientific hypothesis or a means to create a whole annotation report for a transcriptome.

Trinotate includes TrinotateWeb, which provides a locally-driven web-based graphical interface for navigating transcriptome annotations and analyzing transcript expression and differential expression using the Trinity/RSEM/Bioconductor analysis framework.



1. Table of Contents

Software and Data Required

1. Software Required

- Trinotate *http://trinotate.github.io* download Trinotate
- Trinity (includes support for expression and DE analysis using RSEM and Bioconductor): *http://trinityrnaseq.github.io/* download Trinity. >Note, Trinity is not absolutely required. It is possible to use Trinotate with other sources of transcript data as long as suitable inputs are available.
- TransDecoder for predicting coding regions in transcripts *http://transdecoder.github.io* download TransDecoder.
- sqlite (required for database integration): http://www.sqlite.org/
- NCBI BLAST+: Blast database Homology Search: http://www.ncbi.nlm.nih.gov/books/NBK52640/

- HMMER/PFAM Protein Domain Identification: http://hmmer.janelia.org/download.html

Below are optional but recommended:

- signalP v4 (free academic download) http://www.cbs.dtu.dk/cgi-bin/nph-sw_request?signalp

  ```
  #You should edit the following line to read like so, increasing the max number of entries that can be processed:
  my $MAX_ALLOWED_ENTRIES=2000000;  # default is only 10000

  #also update the path to where you have the signalP software installed eg.:
  $ENV{SIGNALP} = '/usr/local/src/signalp-4.1';
  ```

- tmhmm v2 (free academic download) http://www.cbs.dtu.dk/cgi-bin/nph-sw_request?tmhmm

  ```
  You might need to edit the header lines of the scripts 'tmhmm' and 'tmhmmformat.pl' to read:
  #!/usr/bin/env perl
  ```

- RNAMMER (free academic download) http://www.cbs.dtu.dk/cgi-bin/sw_request?rnammer

  ```
  Installation notes:

      Installing RNAMMER requires a little bit of hacking, unfortunately, and if you follow the instructions below you will likely get it
  working.
  When you obtain the software bundle from the above website, be sure to untar it in a new directory. For example:

  mkdir RNAMMER
  cd RNAMMER
  mv /path/to/rnammer-1.2.src.tar.Z .
  tar zxvf rnammer-1.2.src.tar.Z
  ```

  a. RNAMMER requires the older version of hmmsearch (v2). You can obtain the hmmsearch_v2 at http://eddylab.org/software/hmmer/2.3/hmmer-2.3.tar.gz. After building the software, rename this version of hmmsearch as *hmmsearch2*.

  b. Hack the *rnammer* script like so; In the *rnammer* software configuration, edit the rnammer script to point

  ```
  $HMMSEARCH_BINARY = "/path/to/hmmsearch2";
      # be sure to give the complete path to where you installed hmmsearch2.

      # update the INSTALL_PATH setting:
  $INSTALL_PATH = "/dir/where/you/installed/RNAMMER";
  ```

  c. Hack the *core-rnammer* script like so:

  ```
  There are two places where you'll find '--cpu 1 --compat'.  Remove the '--cpu1' at each of these places, and retain the '--compat'.
  ```

  d. Be sure that rnammer functions correctly by executing it on their provided sample data. RNAMMER is quite useful, but the current implementation is not robust to error, so check carefully.

  ```
      # visit the example directory included in rnammer
  cd RNAMMER/example

      # now run the example command like so:
  ../rnammer -S bac -m lsu,ssu,tsu -xml ecoli.xml -gff ecoli.gff -h ecoli.hmmreport < ecoli.fsa

  If it runs without error AND generates new 'ecoli.xml', 'ecoli.gff', and 'ecoli.hmmreport' files (check the datestamps on the files
  via 'ls -ltr'), then congratulate yourself for successfully installing rnammer.
  ```

2. Sequence Databases Required

Trinotate **relies heavily on SwissProt and Pfam**, and custom protein files are generated as described below to be specifically used with Trinotate. You can obtain the protein database files by running this Trinotate build process. This step will download several data resources including the latest version of swissprot, pfam, and other companion resources, create and populate a Trinotate boilerplate sqlite database (Trinotate.sqlite), and yield *uniprot_sprot.pep* file to be used with BLAST, and the *Pfam-A.hmm.gz* file to be used for Pfam searches. Run the build process like so:

```
$TRINOTATE_HOME/admin/Build_Trinotate_Boilerplate_SQLite_db.pl  Trinotate
```

and once it completes, it will provide to you:

```
Trinotate.sqlite
uniprot_sprot.pep
Pfam-A.hmm.gz
```

Prepare the protein database for blast searches by:

```
makeblastdb -in uniprot_sprot.pep -dbtype prot
```

Uncompress and prepare the Pfam database for use with *hmmscan* like so:

```
gunzip Pfam-A.hmm.gz
hmmpress Pfam-A.hmm
```

Running Sequence Analyses

Both the transcripts and any predicted protein-coding regions are subject to analysis using the various strategies below.

Trinity transcripts are searched for sequence homologies using BLASTX, and RNAMMER is used to identify potential rRNA transcripts.

Transdecoder-predicted coding regions are also searched for sequence homologies using BLASTP, protein domains identified via a Pfam search, and additional properties such as signal peptides and likely transmembrane-spanning regions are explored.

1. Files needed for execution

- *Trinity.fasta* - Final product containing all the transcripts assembled by Trinity
- *Trinity.fasta.transdecoder.pep* - Most likely Longest-ORF peptide candidates generated from the Trinity Assembly. Instructions for generation of this file can be found here: http://transdecoder.github.io/

2. Capturing BLAST Homologies

NCBI BLAST

BLAST information Instructions for installation of command line stand alone blast can be found here: http://www.ncbi.nlm.nih.gov/books/NBK52640/ NOTE: This step will undoubtedly take the longest, for very large files execution on a multi-cpu server HPC environment is highly recommended, and your thread count should be equal to the number of CPU’s present on the node the job is run on.

Blast Commands

```
# search Trinity transcripts
blastx -query Trinity.fasta -db uniprot_sprot.pep -num_threads 8 -max_target_seqs 1 -outfmt 6 > blastx.outfmt6

# search Transdecoder-predicted proteins

blastp -query transdecoder.pep -db uniprot_sprot.pep -num_threads 8 -max_target_seqs 1 -outfmt 6 > blastp.outfmt6
```

In addition to searching uniprot_sprot.pep, you can search any other protein database and load the results in as a custom protein database. Searching Swissprot, however, is critical to Trinotate, because that’s where it retrieves the various Kegg, GO, and Eggnog, etc., annotations from.

num_threads should be equal to the amount of cores available

Note If you have access to a compute farm running LSF, SGE, PBS, or SLURM, consider using HPC GridRunner to maximally parallelize your blast searches.

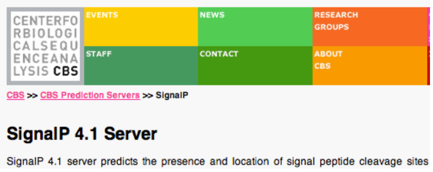
3. Running HMMER to identify protein domains

hmmscan (HMMER) command:

```
hmmscan --cpu 8 --domtblout TrinotatePFAM.out Pfam-A.hmm transdecoder.pep > pfam.log
```

Note num_threads should be equal to the number of cores available

4. Running signalP to predict signal peptides



CBS >> CBS Prediction Servers >> SignalP

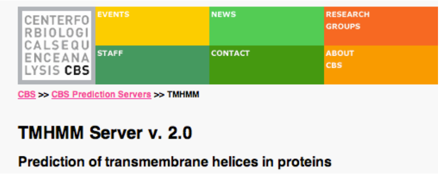
SignalP 4.1 Server

SignalP 4.1 server predicts the presence and location of signal peptide cleavage sites

signalP command:

```
signalp -f short -n signalp.out transdecoder.pep
```

5. Running tmHMM to predict transmembrane regions



CBS >> CBS Prediction Servers >> TMHMM

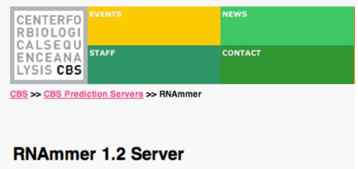
TMHMM Server v. 2.0

Prediction of transmembrane helices in proteins

tmhmm command:

```
tmhmm --short < transdecoder.pep > tmhmm.out
```

6. Running RNAMMER to identify rRNA transcripts



CBS >> CBS Prediction Servers >> RNAmmer

RNAmmer 1.2 Server

RNAMMER was originally developed to identify rRNA genes in genomic sequences. To have it identify rRNA sequences among our large sets of transcriptome sequences, we first concatenate all the transcripts together into a single super-scaffold, run RNAMMER to identify rRNA homologies, and then transform the rRNA feature coordinates in the super-scaffold back to the transcriptome reference coordinates. The following script will perform all of these operations:

```
$TRINOTATE_HOME/util/rnammer_support/RnammerTranscriptome.pl

############################################################################
```

```
#
#  --transcriptome <string>      Transcriptome assembly fasta file
#
#  --path_to_rnammer <string>    Path to the rnammer software
#                                (ie.  /usr/bin/software/rnammer_v1.2/rnammer)
#
#  Optional:
#
#  --org_type <string>           arc|bac|euk   (default: euk)
#
################################################################################
```

And so, you might execute it like so:

```
$TRINOTATE_HOME/util/rnammer_support/RnammerTranscriptome.pl --transcriptome Trinity.fasta --path_to_rnammer /usr/bin/software/rnammer_v1.2
/rnammer
```

Once complete, it will have generated a file: *Trinity.fasta.rnammer.gff*, which can be loaded into Trinotate as described in sections below.

Trinotate: Loading Above Results into a Trinotate SQLite Database

The following commands will import the results from the bioinformatic computes performed above into a Trinotate SQLite database. All operations are performed using the included *Trinotate* utility. Usage is like so:

```
usage: Trinotate <sqlite.db> <command> <input> [...]

<commands>:
```

- Initial import of transcriptome and protein data:

  ```
  init --gene_trans_map <file> --transcript_fasta <file> --transdecoder_pep <file>
  ```

- Transdecoder protein search results:

  ```
  LOAD_swissprot_blastp <file>
  LOAD_pfam <file>
  LOAD_tmhmm <file>
  LOAD_signalp <file>
  ```

- Trinity transcript search results:

  ```
  LOAD_swissprot_blastx <file>
  LOAD_rnammer <file>
  ```

- Load custom blast results using any searchable database

  ```
  LOAD_custom_blast --outfmt6 <file> --prog <blastp|blastx> --dbtype <database_name>
  ```

- report generation:

  ```
  report [ -E (default: 1e-5) ] [--pfam_cutoff DNC|DGC|DTC|SNC|SGC|STC (default: DNC=domain noise cutoff)]
  ```

Follow the steps below to obtain a boilerplate Trinotate sqlite database and populate it with your data.

1. Load transcripts and coding regions

Begin populating the sqlite database by loading three data types:

- Transcript sequences (de novo assembled transcripts or reference transcripts)
- Protein sequences (currently as defined by TransDecoder)
- Gene/Transcript relationships (tab delimited format: "gene_id(tab)transcript_id", same as used by the RSEM software). If you are using Trinity assemblies, you can generate this file like so:

  ```
  $TRINITY_HOME/util/support_scripts/get_Trinity_gene_to_trans_map.pl Trinity.fasta >  Trinity.fasta.gene_trans_map
  ```

> **Note** If you’re not using Trinity transcript assemblies, then it’s up to you to provide the correspinding gene-to-transcript mapping file.

Load these info into the Trinotate sqlite database like so (example, using Trinity assemblies):

```
Trinotate Trinotate.sqlite init --gene_trans_map Trinity.fasta.gene_trans_map --transcript_fasta Trinity.fasta --transdecoder_pep
transdecoder.pep
```

2. Loading BLAST homologies

```
# load protein hits
Trinotate Trinotate.sqlite LOAD_swissprot_blastp blastp.outfmt6

# load transcript hits
Trinotate Trinotate.sqlite LOAD_swissprot_blastx blastx.outfmt6
```

Optional: load custom database blast hits:

```
# load protein hits
Trinotate Trinotate.sqlite LOAD_custom_blast --outfmt6 custom_db.blastp.outfmt6 --prog blastp --dbtype custom_db_name

# load transcript hits
Trinotate Trinotate.sqlite LOAD_custom_blast --outfmt6 custom_db.blastx.outfmt6 --prog blastx --dbtype custom_db_name
```

3. Load Pfam domain entries

```
Trinotate Trinotate.sqlite LOAD_pfam TrinotatePFAM.out
```

4. Load transmembrane domains

```
Trinotate Trinotate.sqlite LOAD_tmhmm tmhmm.out
```

5. Load signal peptide predictions

```
Trinotate Trinotate.sqlite LOAD_signalp signalp.out
```

Trinotate: Output an Annotation Report

```
Trinotate Trinotate.sqlite report [opts] > trinotate_annotation_report.xls
```

Note, you can threshold the blast and pfam results to be reported by including the options below:

```
#################################################################
#
#  -E <float>                 maximum E-value for reporting best blast hit
#                             and associated annotations.
#
#  --pfam_cutoff <string>     'DNC' : domain noise cutoff (default)
#                             'DGC' : domain gathering cutoff
#                             'DTC' : domain trusted cutoff
#                             'SNC' : sequence noise cutoff
#                             'SGC' : sequence gathering cutoff
#                             'STC' : sequence trusted cutoff
#
#################################################################
```

The output has the following column headers:

```
0       #gene_id
1       transcript_id
2       sprot_Top_BLASTX_hit
3       RNAMMER
4       prot_id
5       prot_coords
6       sprot_Top_BLASTP_hit
7       custom_pombe_pep_BLASTX
8       custom_pombe_pep_BLASTP
9       Pfam
10      SignalP
11      TmHMM
12      eggnog
13      Kegg
14      gene_ontology_blast
15      gene_ontology_pfam
16      transcript
17      peptide
```

and the data are formatted like so:

```
0       TRINITY_DN179_c0_g1
1       TRINITY_DN179_c0_g1_i1
2       GCS1_SCHPO^GCS1_SCHPO^Q:53-2476,H:1-808^100%ID^E:0^RecName: Full=Probable mannosyl-oligosaccharide glucosidase;^Eukaryota; Fungi;
Dikarya; Ascomycota; Taphrinomycotina; Schizosaccharomycetes; Schizosaccharomycetales; Schizosaccharomycetaceae; Schizosaccharomyces
3       .
4       TRINITY_DN179_c0_g1_i1|m.1
5       2-2479[+]
6       GCS1_SCHPO^GCS1_SCHPO^Q:18-825,H:1-808^100%ID^E:0^RecName: Full=Probable mannosyl-oligosaccharide glucosidase;^Eukaryota; Fungi;
Dikarya; Ascomycota; Taphrinomycotina; Schizosaccharomycetes; Schizosaccharomycetales; Schizosaccharomycetaceae; Schizosaccharomyces
7
SPAC6G10_09_SPAC6G10_09_I_alpha_glucosidase_I_Gls1_predicte^SPAC6G10_09_SPAC6G10_09_I_alpha_glucosidase_I_Gls1_predicte^Q:53-2476,H:1-808^100%ID^E:0^
8
SPAC6G10_09_SPAC6G10_09_I_alpha_glucosidase_I_Gls1_predicte^SPAC6G10_09_SPAC6G10_09_I_alpha_glucosidase_I_Gls1_predicte^Q:18-825,H:1-808^100%ID^E:0^.
9       PF16923.2^Glyco_hydro_63N^Glycosyl hydrolase family 63 N-terminal domain^58-275^E:6.9e-60`PF03200.13^Glyco_hydro_63^Glycosyl
hydrolase family 63 C-terminal domain^315-823^E:5.1e-187
10      .
11      .
12      .
13      KEGG:spo:SPAC6G10.09`KO:K01228
14      GO:0005783^cellular_component^endoplasmic reticulum`GO:0005789^cellular_component^endoplasmic reticulum
membrane`GO:0016021^cellular_component^integral component of membrane`GO:0004573^molecular_function^mannosyl-oligosaccharide glucosidase
activity`GO:0009272^biological_process^fungal-type cell wall biogenesis`GO:0009311^biological_process^oligosaccharide metabolic
process`GO:0006487^biological_process^protein N-linked glycosylation
15      .
16      .
17      .
```

> **Note** Include options *report --incl_pep --incl_trans* to add the protein and transcript sequence data in the above tab delimited report.

```
# Example rRNA entry


 0       TRINITY_DN2464_c0_g1
 1       TRINITY_DN2464_c0_g1_i1
 2       ART2_YEAST^ART2_YEAST^Q:6813-6646,H:1-56^85.71%ID^E:2e-23^RecName: Full=Putative uncharacterized protein ART2;^Eukaryota; Fungi;
Dikarya; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycetales; Saccharomycetaceae; Saccharo
myces
 3       18s_rRNA^1258-3098`28s_rRNA^3521-7502
 4       TRINITY_DN2464_c0_g1_i1|m.606
 5       6628-6960[-]
 6       ART2_YEAST^ART2_YEAST^Q:50-105,H:1-56^85.71%ID^E:3e-28^RecName: Full=Putative uncharacterized protein ART2;^Eukaryota; Fungi;
Dikarya ; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycetales; Saccharomycetaceae; Saccharomyces
 7       .
 8       .
 9       .
 10      .
 11      .
 12      .
 13      .
 14      .
 15      .
 16      .
 17      .
```

> **Note** The Trinity-assembled 18S/28S S. pombe rRNA region includes a TransDecoer predicted ORF with a blast match to an S. cerevisiae protein "Antisense to ribosomal RNA transcript protein 2 (ART2).

Backticks (`) and carets (^) are used as delimiters for data packed within an individual field, such as separating E-values, percent identity, and taxonomic info for best matches. When there are multiple assignments in a given field, the assignments are separated by (\`) and the fields within an assignment are separated by (\^). In a future release (post Feb-2013), the backticks and carets will be used more uniformly than above, such as carets as BLAST field separators, and including more than the top hit.

Automated Execution of Trinotate: Running computes and loading results

Trinotate now comes with a script that automates the process of running all the above computes and loading the results. You'll find it as:

```
$TRINOTATE_HOME/auto/autoTrinotate.pl
```

```
##############################################################################
#
# Required:
#
#  --Trinotate_sqlite <string>                Trinotate.sqlite boilerplate database
#
#  --transcripts <string>                     transcripts.fasta
#
#  --gene_to_trans_map <string>               gene-to-transcript mapping file
#
#  --conf <string>                            config file
#
#  --CPU <int>                                number of threads to use.
#
#
##############################################################################
```

The configuration file given to the *--conf* parameter describes the pipeline execution. An example config file is provided as:

`$TRINOTATE_HOME/auto/conf.txt`

Simply update the header portion of the conf.txt file to indicate the path to the programs you have installed and the paths to the resources indicated, and then run the script to process the data.

Trinotate: Sample data and execution

Sample data and a *runMe.sh* script are available at $TRINOTATE_HOME/sample_data/

Executing the *runMe.sh* script will pull down the Trinotate sqlite boilerplate database, populate with the provided bioinformatics computes, and generate the final Trinotate annotation report. In addition, pre-computed expression and DE analyses will be loaded for use with TrinotateWeb.

Literature references for software used for functional annotation

- [Trinity]Full-length transcriptome assembly from RNA-Seq data without a reference genome. Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, Regev A. Nature Biotechnology 29, 644�652 (2011)
- [HMMER]HMMER web server: interactive sequence similarity searching R.D. Finn, J. Clements, S.R. Eddy Nucleic Acids Research (2011) Web Server Issue 39:W29-W37.
- [PFAM] The Pfam protein families database Punta, P.C. Coggill, R.Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, Ceric, J. Clements, A. Heger, L. Holm, E.L.L. Sonnhammer, S.R. Eddy, A. Bateman, R.D. Finn Nucleic Acids Research (2012) Database Issue 40:D290-D301
- [SignalP]SignalP 4.0: discriminating signal peptides from transmembrane regions Thomas Nordahl Petersen, Soren Brunak, Gunnar von Heijne & Henrik Nielsen Nature Methods, 8:785-786, 2011
- [tmHMM]Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. Krogh A, Larsson B, von Heijne G, Sonnhammer EL. J Mol Biol. 2001 Jan 19;305(3):567-80.
- [BLAST]Basic local alignment search tool. Altschul SF; Gish W; Miller W; Myers EW; Lipman DJ J Mol Biol 215: 403-10 (1990)
- [KEGG]KEGG for integration and interpretation of large-scale molecular datasets. Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M.; Nucleic Acids Res. 40, D109-D114 (2012).
- [GO]Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium Nature Genet. 25: 25-29 (2000)
- [eggNOG]eggNOG v3.0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges. Powell S, Szklarczyk D, Trachana K, Roth A, Kuhn M, Muller J, Arnold R, Rattei T, Letunic I, Doerks T, Jensen LJ, von Mering C, Bork P. Nucleic Acids Res. 2012 Jan;40(Database issue):D284-9. Epub 2011 Nov 16.
- [RNAMMER] RNammer: consistent annotation of rRNA genes in genomic sequences Lagesen K, Hallin PF, Rodland E, Staerfeldt HH, Rognes T Ussery DW Nucleic Acids Res. 2007 Apr 22.

Contact Us

User support for Trinotate is provided at the Trinotate Google Group: https://groups.google.com/forum/?hl=en#!forum/trinotate-users

Last updated 2017-01-03 21:45:02 EST