TRANSCRIPTOMICS

Annotation

Day 03

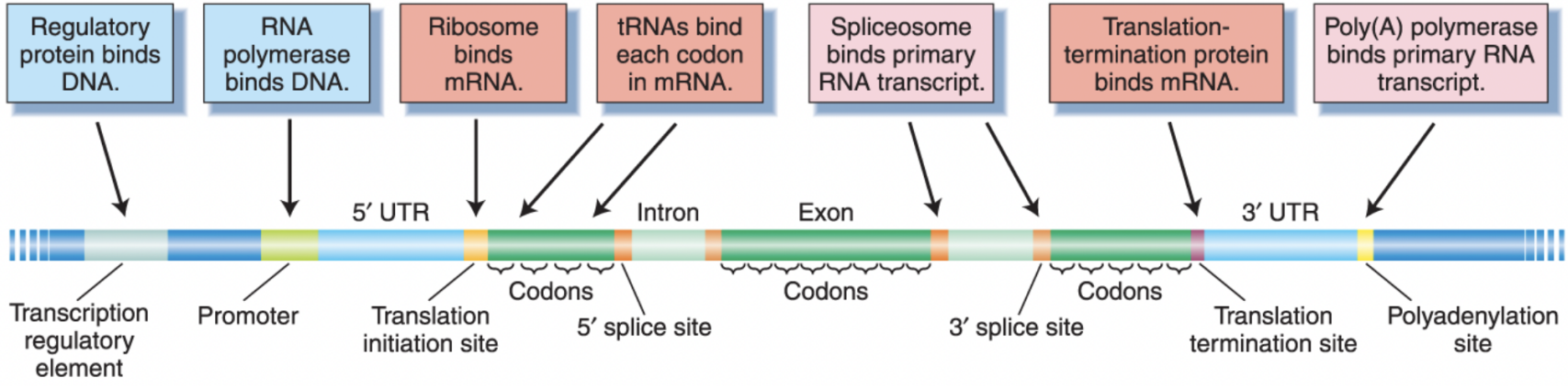
https://tttorres.github.io/transcriptomics/

Transcriptome Annotation

Meaning from seguences

Transcriptome Annotation

Meaning from seguences



RNA-Seq: Annotation

Transcriptome Annotation

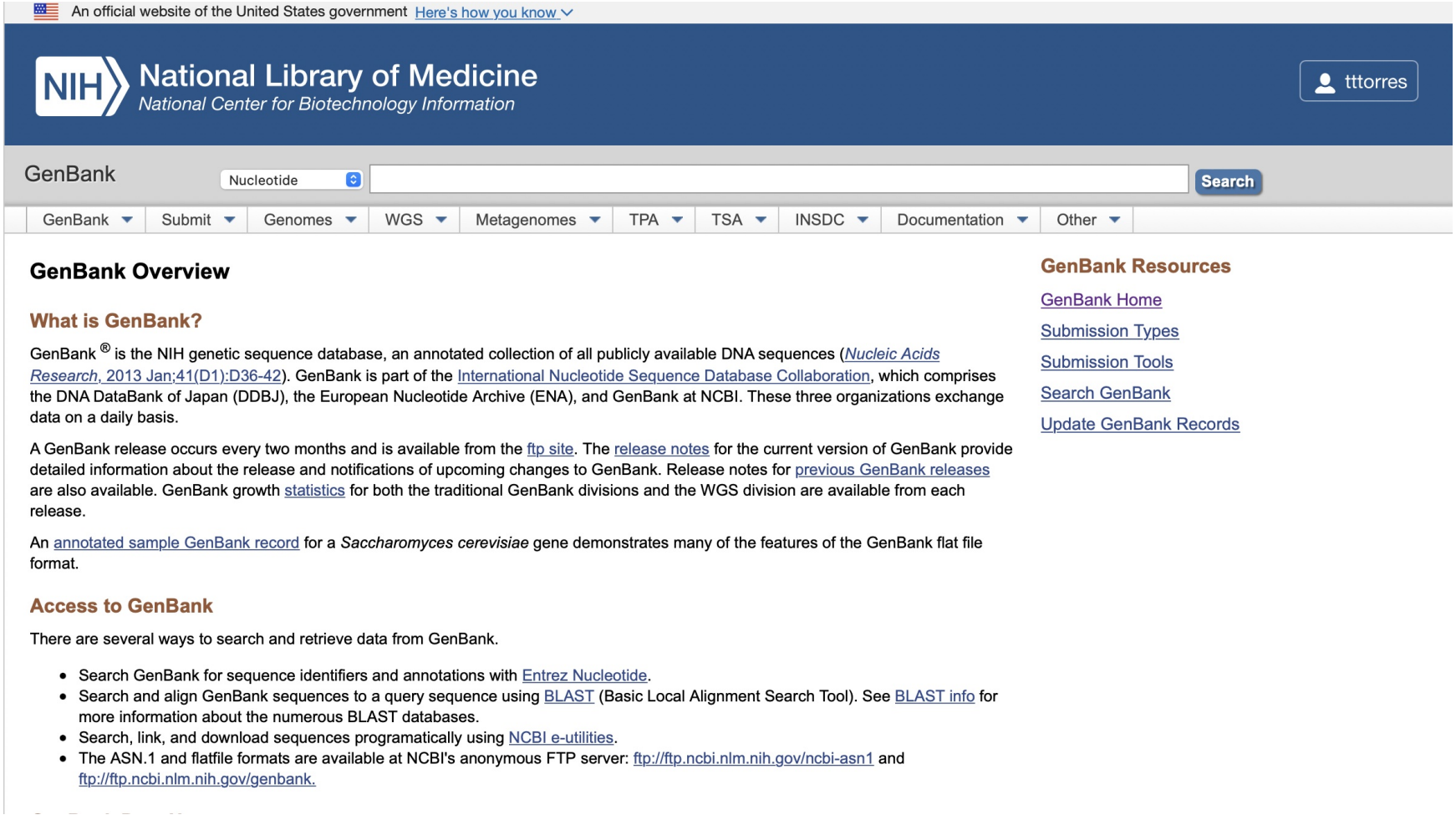
> 📝 Assigning biological meaning to assembled sequences by identifying coding regions, functional domains, and associated biological processes.

Why is it Important?

- To understand gene functions.
- To identify potential coding sequences and functional domains.
- To classify transcripts based on biological processes, molecular functions, and cellular components.

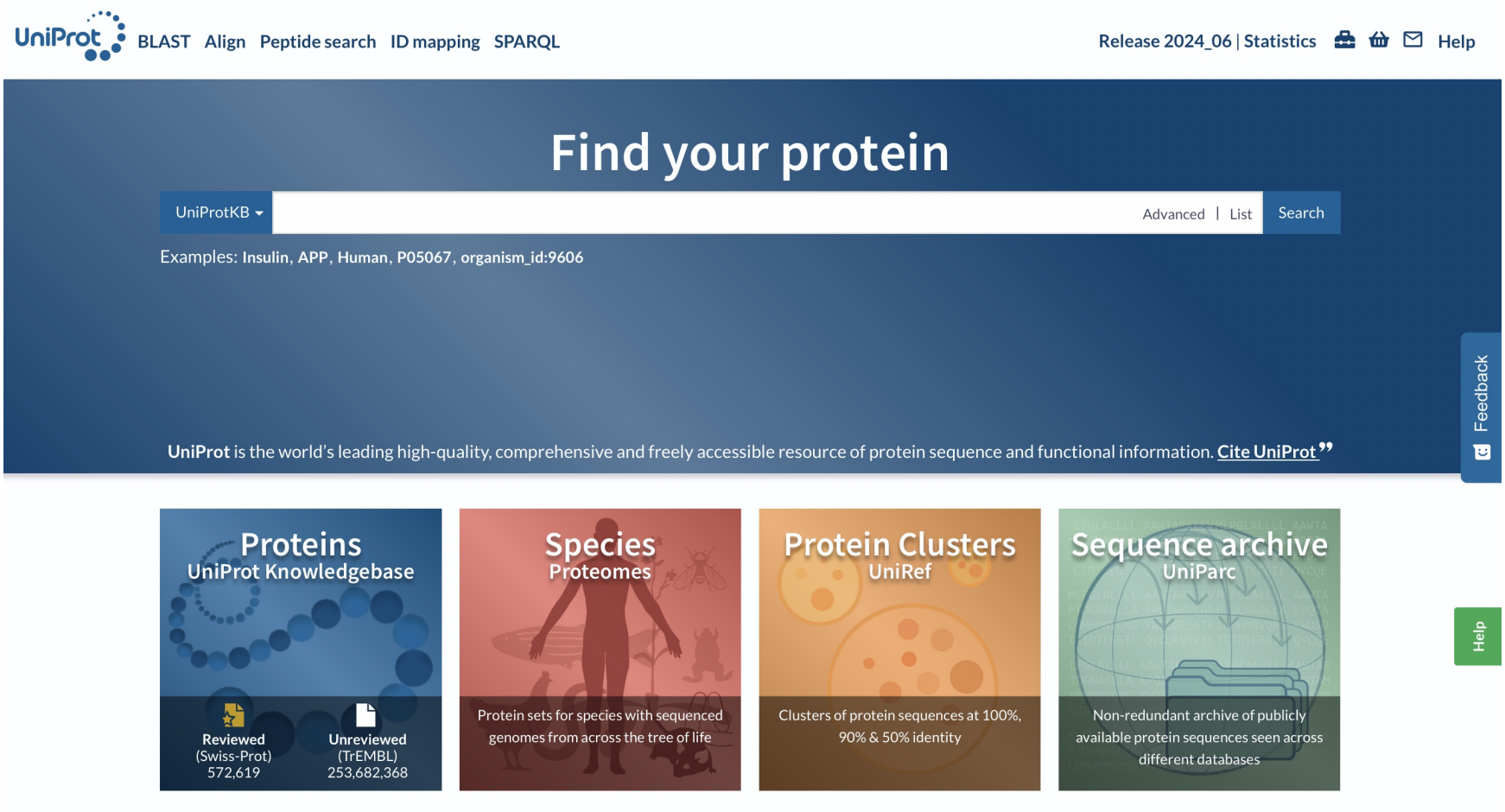
Predicting function from sequence

Sequence similarity



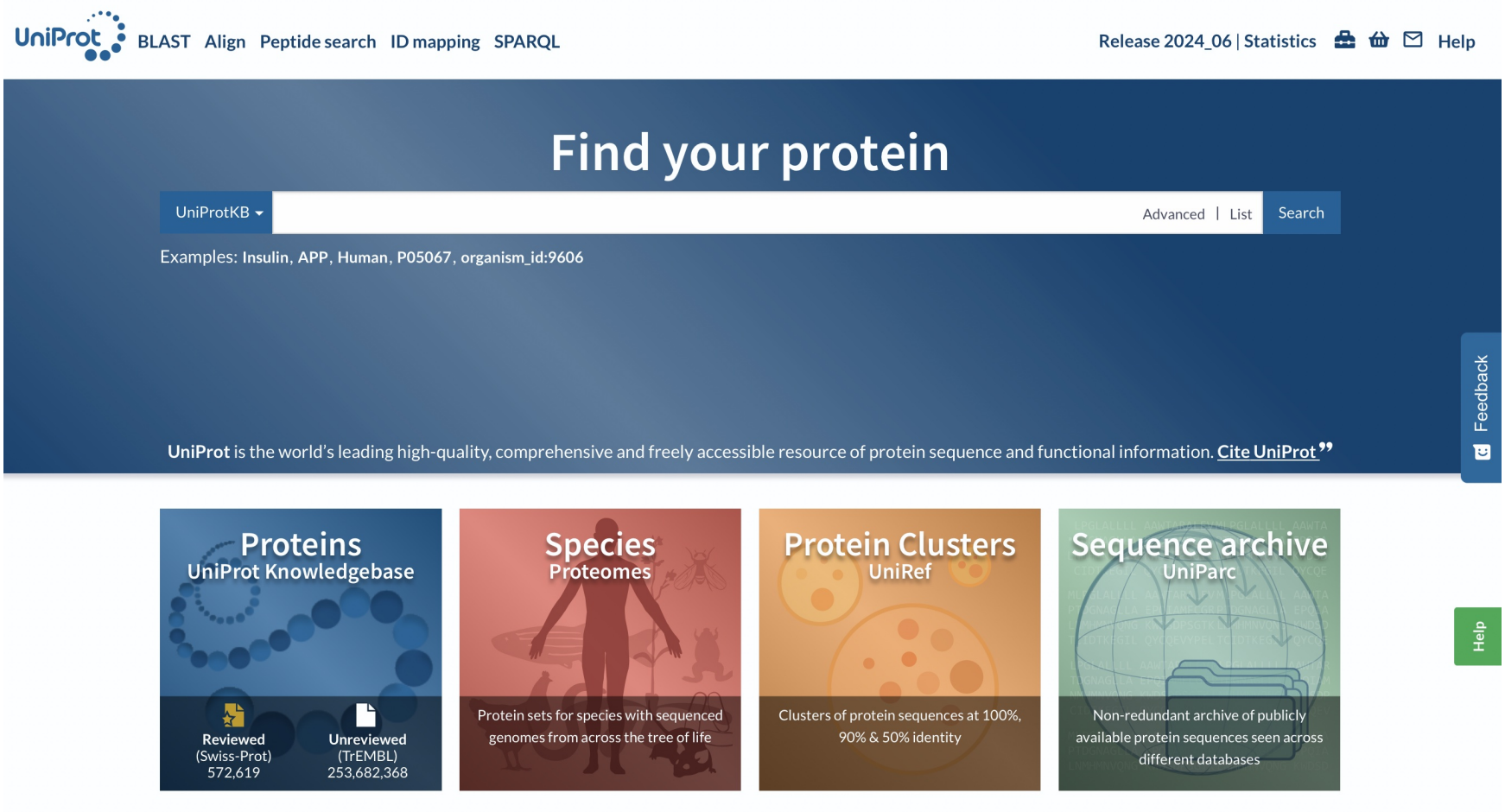
Predicting function from sequence

Sequence similarity



Predicting function from sequence

Sequence similarity



Predicting function from sequence

No sequence similarity!

Predicting function from sequence

No sequence similarity!

Is there an ORF for a potential Coding Region?

GGAGCTGGAGGCCCCCAGGCAACTACACCGTCCACGTACCCAGAGGGGCTGGGCCCTCCC
ACCAGAGACCACGCCCTGGTGTGCCTTAGGGGCCCTGGTTTGTTAGTCTCTGAGTGTGCA
GTTGCTGCAC**ATGGGGCCCTGGCGCTTGCTGCACCAACTTCCTGTTGGGCCCGTGGTCCT**
TGGAGGCATGCAGTTCAGCAGACAGTGACTCAGCCATCCACCCAACATGCGGAACGTGTC
TCTTCTGCAGGTCCCGGTCCACAGCAGGATTCCCCCTCTGTGAAAAGGCACGCTGATCTG
TCTGGATAAGTGTGGCCGGCCCCATGTATCCGGAATCAACCACGGGGTCCCCAGCTCGAC
TCTCCCTGCGGCAGACAGGCTCCCCCGGGATGATCTACAGTACTCGTTATGGGAGTCCCA
AAAGACAGCTCCAGTTTTACAGGAATCTGGGCAAATCTGGCCTTCGGGTCTCCTGCCTGG
GGCTTGGAACATGGGTGACCTTCGGGGGCCAGATCACGGATGAGATGGCAGAGCACCTAA
TGACCTTGGCCTACGATAATGGCATCAACCTGTTCGATACGGCGGAGGTCTACGCTGCTG
GAAAAGCTGAAGTGGTATTAGGGAACATCATTAAGAAGAAGGGATGGAGACGGTCCAGCC

Predicting function from sequence

ORF Finder

Finds all open reading frames and provides translations

NIH National Library of Medicine
National Center for Biotechnology Information

tttorres

Open Reading Frame Finder

ORF finder searches for open reading frames (ORFs) in the DNA sequence you enter. The program returns the range of each ORF, along with its protein translation. Use ORF finder to search newly sequenced DNA for potential protein encoding segments, verify predicted protein using newly developed SMART BLAST or regular BLASTP.

This web version of the ORF finder is limited to the subrange of the query sequence up to 50 kb long. Stand-alone version, which doesn't have query sequence length limitation, is available for Linux x64.

Examples (click to set values, then click Submit button) :

- NC_011604 Salmonella enterica plasmid pWES-1; genetic code: 11; 'ATG' and alternative initiation codons; minimal ORF length: 300 nt
- NM_000059; genetic code: 1; start codon: 'ATG only'; minimal ORF length: 150 nt

Enter Query Sequence

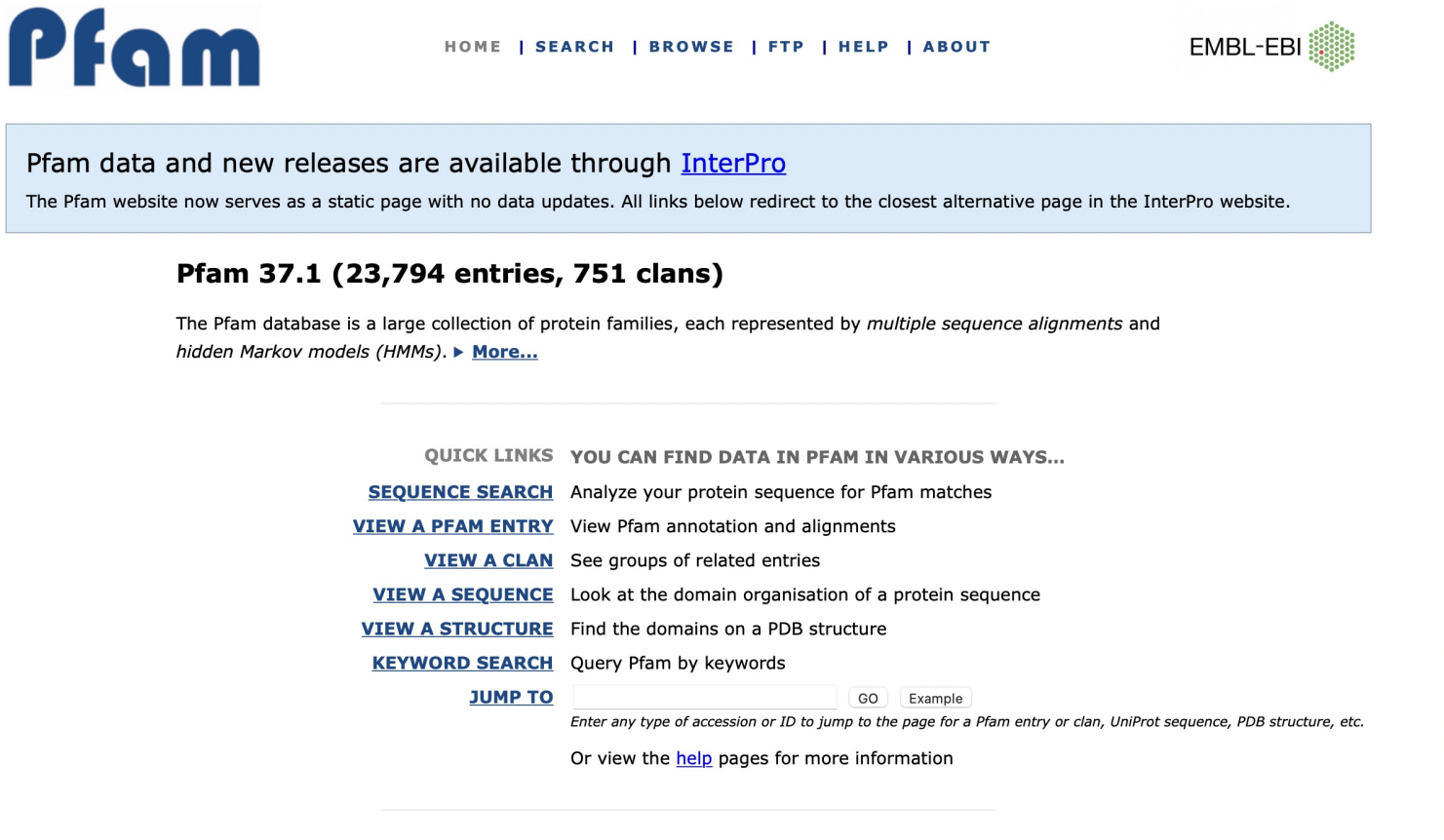
Enter accession number, gi, or nucleotide sequence in FASTA format:

From: **To:**

Predicting function from sequence

Pfam database

Large collection of protein families, each represented by multiple sequence alignments and hidden Markov models

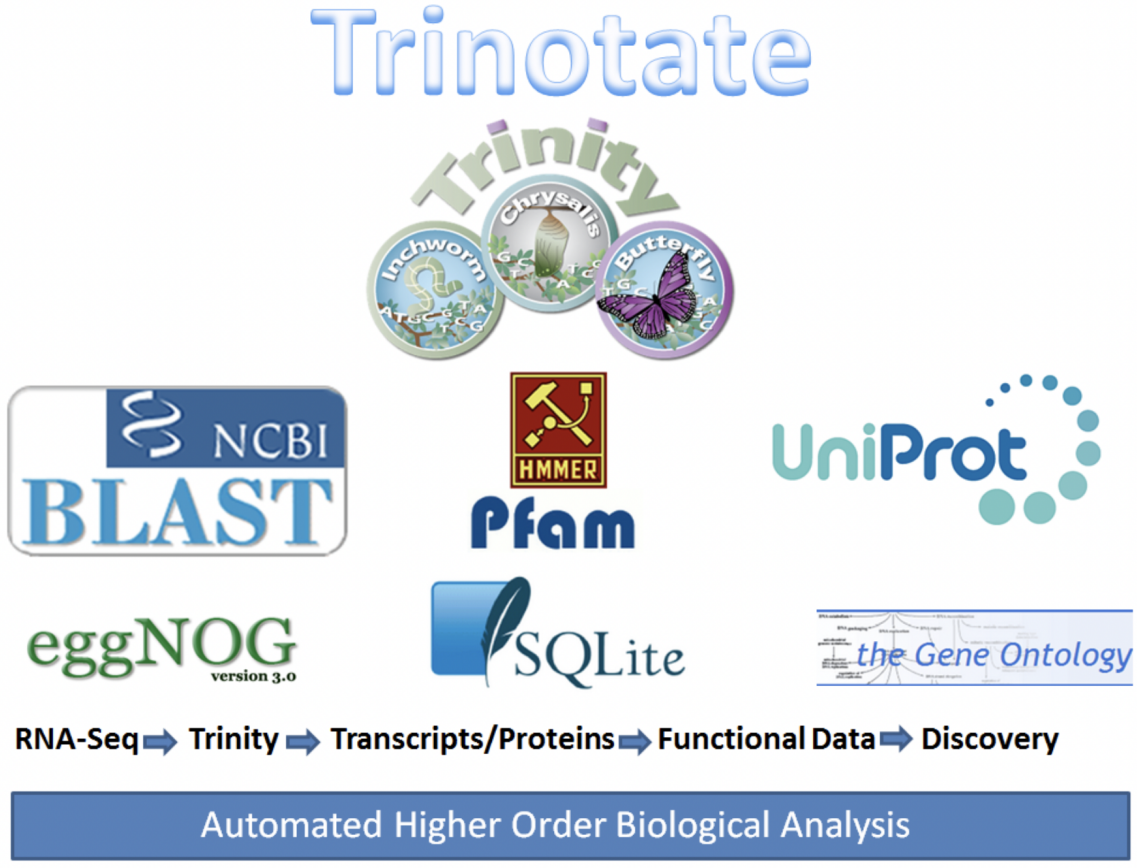


RNA-Seq: Annotation

Methods to predict gene function

Tool for organizing all this data

Trinotate: Transcriptome Functional Annotation and Analysis



RNA-Seq: Annotation

Transcriptome Annotation Workflow

1. **Coding Region Identification:** Detecting open reading frames (ORFs).
2. **Functional Annotation:** Link sequences to known proteins.
3. **Prediction of sequence features:** Identifying conserved functional domains and signal peptides
4. **Integration:** Compiling results into a single database
5. **GO Term Association:** Assigning Gene Ontology terms for functional classification.

RNA-Seq: Annotation

Transcriptome Annotation Workflow

1. **Coding Region Identification:** Transdecoder
2. **Functional Annotation:** Blast and HMMER.
3. **Domain Search:** signalP and tmhmm.
4. **Integration:** Compiling results into a single database
5. **GO Term Association:** Assigning Gene Ontology terms for functional classification.

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

> TransDecoder predicts open reading frames (ORFs) in transcript sequences. These ORFs represent potential protein-coding regions.

Key Steps:

1. **LongOrfs:** Identifies likely coding regions based on sequence properties.
2. **Predict:** Refines ORF predictions by incorporating similaty and coding potential.

https://github.com/TransDecoder/TransDecoder/wiki

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

1. Create and navigate to a new folder

```
mkdir ~/rnaseq/04-Annotation/
cd ~/rnaseq/04-Annotation/
```

2. Activate environment:

```
conda activate annotation
```

3. Run TransDecoder.LongOrfs:

```
TransDecoder.LongOrfs -t ${SPECIES}.Trinity.fasta
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `base_freqs.dat`

```
cd ${SPECIES}.Trinity.fasta.transdecoder_dir
head base_freqs.dat
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `base_freqs.dat`

```
cd ${SPECIES}.Trinity.fasta.transdecoder_dir
head base_freqs.dat
A       552053  0.306
C       348809  0.194
G       348809  0.194
T       552053  0.306
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `longest_orfs.cds`

```
head longest_orfs.cds
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `longest_orfs.cds`

```
>TRINITY_DN1811_c0_g1_i1.p1 type:5prime_partial \
TRINITY_DN1811_c0_g1_i1:1-639(+)
CAGGATGTCTATAAAATTGGTGGTATTGGTACAGTACCCGTGGGTCGTGTTGAAACT
>TRINITY_DN1811_c0_g1_i1.p2 type:3prime_partial \
TRINITY_DN1811_c0_g1_i1:351-1(-)
ATGAGCAGTATGACAATCCAAAACTGGAGTATAACCATTGGCAATTTGTCCAGGATG
>TRINITY_DN1803_c0_g1_i1.p1 type:internal \
TRINITY_DN1803_c0_g1_i1:1-444(+)
ATCGTACTCCACTTGACAAGCAACATGAAATTTTTAATTGTTTTCGCTTTGGCTTTG
>TRINITY_DN1854_c0_g1_i1.p1 type:internal \
TRINITY_DN1854_c0_g1_i1:311-3(-)
GAAAAACGTAAACTGAAATTGGTATGGCGTAATATAATTGCATTCGGTTATTTGCAT
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `longest_orfs.gff3`

```
head longest_orfs.gff3
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `longest_orfs.gff3`

```
TRINITY_DN1811_c0_g1_i1 transdecoder    gene    1       7
TRINITY_DN1811_c0_g1_i1 transdecoder    mRNA    1       7
TRINITY_DN1811_c0_g1_i1 transdecoder    exon    1       7
TRINITY_DN1811_c0_g1_i1 transdecoder    CDS     1       6
TRINITY_DN1811_c0_g1_i1 transdecoder    three_prime_UTR 6
TRINITY_DN1811_c0_g1_i1 transdecoder    gene    1       7
TRINITY_DN1811_c0_g1_i1 transdecoder    mRNA    1       7
TRINITY_DN1811_c0_g1_i1 transdecoder    five_prime_UTR  3
TRINITY_DN1811_c0_g1_i1 transdecoder    exon    1       7
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `longest_orfs.pep`

```
head longest_orfs.pep
```

RNA-Seq: Annotation

Identifying Coding Regions with TransDecoder

4. Check files: `longest_orfs.pep`

```
>TRINITY_DN1811_c0_g1_i1.p1 type:5prime_partial gc:univers
QDVYKIGGIGTVPVGRVETGILKPGMVVNFAPVNLVTEVKSVEMHHEALSEAMPGDN
>TRINITY_DN1811_c0_g1_i1.p2 type:3prime_partial gc:univers
MSSMTIQNWSITIGNLSRMIKNNYLSCKISSSLRRTIFGITGNITATQFLNRNVFNI
>TRINITY_DN1803_c0_g1_i1.p1 type:internal gc:universal TRI
IVLHLTSNMKFLIVFALALATSASAELVSRSVVVPVLENEGRITNGQTASVGQFPYQ
>TRINITY_DN1854_c0_g1_i1.p1 type:internal gc:universal TRI
EKRKLKLVWRNIIAFGYLHLAALYGAYLLFTSAKWQTIAFAFGLYVVSGLGITAGAH
>TRINITY_DN1821_c0_g1_i1.p1 type:internal gc:universal TRI
SEVKVKVRINHHGIVLISSASLVDKKELEEPQTPQPPEQQINTEQPASGEGQPANAG
```

RNA-Seq: Annotation

Identify ORFs with similarity to known proteins

BLAST

> Search a protein database such as Swissprot (fast) or Uniref90 (slow but more comprehensive) using BLAST+

RNA-Seq: Annotation

Identify ORFs with similarity to known proteins

4a. Download databases (Linux)

```
wget https://ftp.uniprot.org/pub/databases/uniprot/curren
wget hhttps://ftp.ebi.ac.uk/pub/databases/Pfam/current_re
```

4b. Download databases (MacOs)

```
curl -O https://ftp.uniprot.org/pub/databases/uniprot/cur
curl -O https://ftp.ebi.ac.uk/pub/databases/Pfam/current_
```

RNA-Seq: Annotation

Identify ORFs with similarity to known proteins

5. Prepare databases

```
gunzip uniprot_sprot.fasta.gz
gunzip Pfam-A.hmm.gz
makeblastdb -in uniprot_sprot.fasta -dbtype prot
hmmpress Pfam-A.hmm
```

RNA-Seq: Annotation

Identify ORFs with similarity to known proteins

5. Search a protein database such as Swissprot (fast) or Uniref90 (slow but more comprehensive) using BLAST+

```
blastp -query transdecoder_dir/longest_orfs.pep  \
       -db uniprot_sprot.fasta  -max_target_seqs 1 \
       -outfmt 6 -evalue 1e-5 -num_threads 8 > blastp.out
```

- -query: Input file name
- -db: BLAST database name
- -max_target_seqs: Maximum number of aligned seqs to keep
- -outfmt: alignment view options (6 = Tabular)
- -evalue: Expectation value (E) threshold for saving hits
- -num_threads
- >

RNA-Seq: Annotation

Identify ORFs with similarity to known proteins

BLAST results

```
head blastp.out
TRINITY_DN1811_c0_g1_i1.p1      sp|P05303|EF1A2_DROME    9
TRINITY_DN1803_c0_g1_i1.p1      sp|P08897|COGS_HYPLI     4
TRINITY_DN1854_c0_g1_i1.p1      sp|Q9BH41|FAD9_ACHDO     7
TRINITY_DN1805_c0_g1_i1.p1      sp|C0HK92|Y5076_DROME    7
TRINITY_DN1864_c0_g1_i1.p1      sp|Q9VHG4|RENR_DROME     5
TRINITY_DN1834_c0_g1_i1.p1      sp|Q00449|MDR49_DROME    6
TRINITY_DN1855_c0_g1_i1.p1      sp|Q27331|VATA2_DROME    9
TRINITY_DN1826_c0_g1_i1.p1      sp|Q7KN62|TERA_DROME     9
TRINITY_DN1826_c0_g1_i1.p1      sp|Q7KN62|TERA_DROME     4
TRINITY_DN1832_c0_g1_i1.p1      sp|P55828|RS20_DROME     9
```

RNA-Seq: Annotation

Identify Functional Domains

HMMER

> Functional domains are conserved regions within proteins responsible for specific biochemical activities. HMMER uses Hidden Markov Models (HMMs) to identify these domains.

RNA-Seq: Annotation

Identify Functional Domains

HMMER

6. Search for functional domains using HMMER

```
hmmscan --cpu 8 --domtblout pfam.domtblout Pfam-A.hmm \
Chom.Trinity.fasta.transdecoder_dir/longest_orfs.pep
```

➡️ -cpu: number of threads

➡️ -domtblout: save parseable table of per-domain hits to file

RNA-Seq: Annotation

Identify Functional Domains

HMMER

6. Search for functional domains using HMMER

```
hmmscan --cpu 8 --domtblout hmmscanPFAM.out Pfam-A.hmm \Cl
```

➡️ -cpu: number of threads

➡️ -domtblout: save parseable table of per-domain hits to file

RNA-Seq: Annotation

Identify Functional Domains

HMMER

```
head hmmscanPFAM.out
#
# target name         accession   tlen query name
#-------------------- ---------- ----- ------------
GTP-eEF1A_C           PF22594.2    101 TRINITY_DN1811_c0_g
GTP_EFTU_D3           PF03143.23   106 TRINITY_DN1811_c0_g
GTP_EFTU_D3           PF03143.23   106 TRINITY_DN1811_c0_g
GTP_EFTU_D2           PF03144.31    73 TRINITY_DN1811_c0_g
GTP_EFTU_D2           PF03144.31    73 TRINITY_DN1811_c0_g
GTP_EFTU_D4           PF14578.12    86 TRINITY_DN1811_c0_g
GTP_EFTU_D4           PF14578.12    86 TRINITY_DN1811_c0_g
```

RNA-Seq: Annotation

Identify signal peptides

> Identifying signal peptides during genome or transcriptome annotation is crucial for understanding the functionality and localization of proteins. Signal peptides are short amino acid sequences at the N-terminal of a protein that direct the protein to specific cellular compartments, often for secretion or localization to organelles like the endoplasmic reticulum (ER), Golgi apparatus, or lysosomes.

RNA-Seq: Annotation

Identify signal peptides

SignalP, this step will not be done during the course

1. Run signalp

```
signalp6 --fastafile longest_orfs.pep \
         --output_dir ${SPECIES}_signalp \
         --organism eukarya
         -f short \
         -n signalp.out
```

RNA-Seq: Annotation

Annotating Gene Ontology (GO) Terms

Gene Ontology (GO) terms classify gene functions into three main categories:

1. **Biological Processes:** Pathways and processes.
2. **Molecular Functions:** Activities performed by gene products.
3. **Cellular Components:** Locations within the cell.

RNA-Seq: Annotation

Annotating Gene Ontology (GO) Terms

Eggnog-Mapper

> EggNOG-mapper is a tool for fast functional annotation of novel sequences. It uses precomputed Orthologous Groups (OGs) and phylogenies from the EggNOG database (http://eggnog5.embl.de) to transfer functional information from fine-grained orthologs only.

RNA-Seq: Annotation

Annotating Gene Ontology (GO) Terms

Eggnog-Mapper

1. Run

```
emapper.py -i longest_orfs.pep \
           -o eggnog \
           --output_dir ~/rnaseq/04-Annotation/${SPECIES}/
           --cpu 8 \
           --tax_scope 6656
```

- -i: input file for annotation
- -o: output file
- --output_dir: output directory
- --cpu 8: number of threads
- --tax_scope: taxon (6656 for Arthropoda)

RNA-Seq: Annotation

Integrating Results with Trinotate

What is Trinotate?

> Trinotate is a comprehensive annotation platform that integrates multiple data sources into a single SQLite database, simplifying downstream analyses.

RNA-Seq: Annotation

Integrating Results with Trinotate

1a. Activante conda environment

```
conda activate annotation
```

1b. Activate docker container

```
docker run --rm -it \
>                  -v `pwd`:/data \
>                  -v /tmp:/tmp \
>                  -e TRINOTATE_HOME=/usr/local/src/Trino
>                  trinityrnaseq/trinotate bash
```

RNA-Seq: Annotation

Integrating Results with Trinotate

2a. Prepare the Trinotate SQLite database in conda

```
Trinotate --create \
          --db ${SPECIES}_Trinotate.sqlite \
          --trinotate_data_dir ~/rnaseq/04-Annotation/\
          ${SPECIES}/Trinotate
```

2b. Prepare the Trinotate SQLite database in docker

```
$TRINOTATE_HOME/Trinotate --create \
          --db ${SPECIES}_Trinotate.sqlite \
          --trinotate_data_dir /data/rnaseq/04-Annotation/
          ${SPECIES}/Trinotate
```

RNA-Seq: Annotation

Integrating Results with Trinotate

Trinotate SQLite database

```
Trinotate --create \
          --db ${SPECIES}_Trinotate.sqlite \
          --trinotate_data_dir ~/rnaseq/04-Annotation/\
          ${SPECIES}/Trinotate
```

- --create: create database
- --db: database name
- --trinotate_data_dir: output directory

RNA-Seq: Annotation

Integrating Results with Trinotate

3a. Integrating results (conda)

```
Trinotate --db ${SPECIES}_Trinotate.sqlite --init \
--gene_trans_map /data/rnaseq/03-Assembly/${SPECIES}/${SP
--transcript_fasta /data/rnaseq/03-Assembly/${SPECIES}/${
--transdecoder_pep /data/rnaseq/04-Annotation/${SPECIES}/
```

3b. Integrating results (docker)

```
$TRINOTATE_HOME/Trinotate --db ${SPECIES}_Trinotate.sqlit
--gene_trans_map /data/rnaseq/03-Assembly/${SPECIES}/${SP
--transcript_fasta /data/rnaseq/03-Assembly/${SPECIES}/${
--transdecoder_pep /data/rnaseq/04-Annotation/${SPECIES}/
```

RNA-Seq: Annotation

Integrating Results with Trinotate

Integrating results

```
$TRINOTATE_HOME/Trinotate --db ${SPECIES}_Trinotate.sqlit
--gene_trans_map /data/rnaseq/03-Assembly/${SPECIES}/${SP
--transcript_fasta /data/rnaseq/03-Assembly/${SPECIES}/${
--transdecoder_pep /data/rnaseq/04-Annotation/${SPECIES}/
```

- --db: database name
- --gene_trans_map: Trinity file map isoforms/genes
- --transcript_fasta: Trinity assembled transcripts
- -transdecoder_pep: Transdecoder ORFs

RNA-Seq: Annotation

Integrating Results with Trinotate

Integrating results

With the database set, we can load several data:

```
Trinotate --db <sqlite.db> --LOAD_swissprot_blastp <file.outfmt6>
Trinotate --db <sqlite.db> --LOAD_pfam <file>
Trinotate --db <sqlite.db> --LOAD_signalp <file>
Trinotate --db <sqlite.db> --LOAD_EggnogMapper <file>
Trinotate --db <sqlite.db> --LOAD_tmhmmv2 <file>
Trinotate --db <sqlite.db> --LOAD_deeptmhmm <file.gff3
```

Expression data can also be loaded in Trinotate database

RNA-Seq: Annotation

Integrating Results with Trinotate

Loading BLAST results against SwissProt database

```
$TRINOTATE_HOME/Trinotate \
             --db ${SPECIES}_Trinotate.sqlite \
             --LOAD_swissprot_blastp blastp.out
```

RNA-Seq: Annotation

Integrating Results with Trinotate

Loading HMMScan results

```
$TRINOTATE_HOME/Trinotate \
             --db ${SPECIES}_Trinotate.sqlite \
             --LOAD_pfam hmmscanPFAM.out
```

RNA-Seq: Annotation

Integrating Results with Trinotate

Loading signalp results

```
$TRINOTATE_HOME/Trinotate \
             --db ${SPECIES}_Trinotate.sqlite \
             --LOAD_signalp signalp/prediction_results.tx
```

RNA-Seq: Annotation

Integrating Results with Trinotate

Loading EggnogMapper results

```
$TRINOTATE_HOME/Trinotate \
              --db ${SPECIES}_Trinotate.sqlite \
              --LOAD_EggnogMapper eggnog.emapper.annotatio
```

RNA-Seq: Annotation

Generate the Trinotate Annotation Report

Check if the database was populated

```
$TRINOTATE_HOME/Trinotate \
             --db ${SPECIES}_Trinotate.sqlite \
             --report >${SPECIES}_TrinotateReport.xls
```

RNA-Seq: Annotation

The Trinotate Annotation Report

The report is a tab-delimited output with the following columns:

```
0       #gene_id
1       transcript_id
2       sprot_Top_BLASTX_hit
3       infernal
4       prot_id
5       prot_coords
6       sprot_Top_BLASTP_hit
7       Pfam
8       SignalP
9       TmHMM
10      eggnog
11      Kegg
12      gene_ontology_BLASTX
13      gene_ontology_BLASTP
14      gene_ontology_Pfam
15      transcript # optional, use --incl_trans
16      peptide # optional, use --incl_pep
```

Slide 10: Annotating Gene Ontology (GO) Terms

What are GO Terms?

Commands:

```
# Extract GO terms from the Trinotate report
Trinotate Trinotate.sqlite report > trinotate_annotation_

# Parse GO terms
go-slim-parse.pl trinotate_annotation_report.xls > GO_ter
```

Slide 11: Visualizing and Interpreting Results

Why Visualize Results?

Visualization tools like WEGO and REVIGO help interpret annotation data by highlighting enriched GO terms and patterns.

Commands:

```
# Example for preparing GO input for WEGO
cut -f2 GO_terms_summary.txt | sort | uniq -c > GO_input.t
```