Gene prediction and annotation of non-model organisms using the Trinity-Transdecoder-Trinotate Pipeline

Cheryl Lewis Ames
Peter Buck Predoctoral Fellow, Smithsonian , NMNH
Ph.D. Candidate, University of Maryland, Biological Sciences Graduate Program
Laboratories of Analytical Biology, Smithsonian, NMNH
May 11, 2015

“RNA sequencing (RNA-Seq), one application of NGS, is a powerful tool, providing information not only about the expression level of genes but also further about the structure of transcripts as it enables to unequivocally identify splicing events, RNA editing products, and mutations in expressed coding sequences within a single experiment.”

Borodina et al. 2011

Annotation

dictionary definition of "to annotate":

– "to make or furnish critical or explanatory notes or comment"

- **some of what this includes for genomics/transcriptomics/proteomics**

– gene product names

– functional characteristics of gene products

– physical characteristics of gene/protein/genome

– overall metabolic profile of the organism

- **elements of the annotation process**

– gene finding

– homology searches

– functional assignment

– ORF management

– data availability

- **manual vs. automatic**

– automatic = computer makes the decisions

- good on easy ones
- bad on hard ones

– manual = human makes the decisions

- highest quality

**Due to the VOLUMES of genome data today, most genome projects are annotated primarily using automated methods with limited manual annotation

Source: http://cpgr.plantbiology.msu.edu/training/workshop_mar07/Lecture3_GenomeAnnotation.pdf

My work: A summary

Functional annotation and differential expression of genes involved in venom, vision and sex in the venomous box jellyfish *Alatina alata* (Cnidaria: Cubozoa)

The phylum Cnidaria: an intriguingly diverse group

corals **jellyfish** **anemones** **hydroids**

sea fans **zoanthids** **siphonophores** **stalked jellies**

Source: Google Images

Cubozoans aka Box jellyfish

elaborate sex

vertebrate-like eyes

Carybdeida

venom

Chirodropida

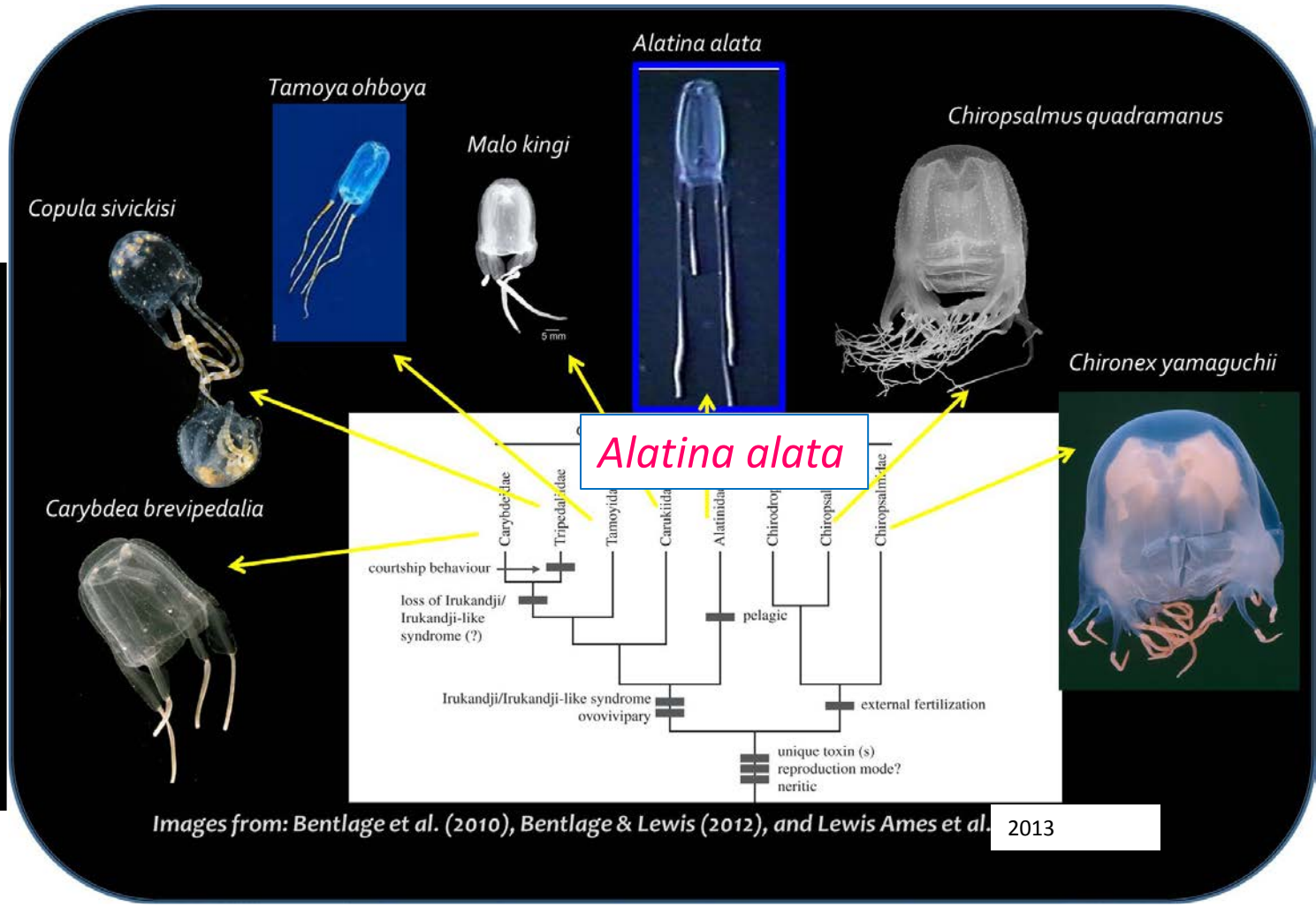


Image sources: Bentlage et al. 2010, Flickr, NOAA

Bioinformatics

- All my major jobs (qsubs) were submitted in Lattice: pool/genomics/my/home/dir

- All Trinity suite software is installed at: pool/genomics/lattice

- All steps vary in time and memory requirements

RNA-Seq De novo Assembly Using Trinity

Trinity, developed at the Broad Institute and the Hebrew University of Jerusalem, represents a novel method (de Bruijn graphs) for the efficient and robust de novo reconstruction of transcriptomes from RNA-seq data.

Three independent software modules: Inchworm, Chrysalis, and Butterfly, are applied sequentially to process large volumes of RNA-seq reads.

Trinity was published in Nature Biotechnology. Protocol for transcriptome assembly and downstream analysis is now published in Nature Protocols

The Trinity software package can be downloaded here.

Runtime and transcript reconstruction performance stats are available for current and previous releases.

Screencast videos are available to introduce you to Trinity and its various components. Also, hands-on tutorials for Trinity and Tuxedo are available as part of our RNA-Seq Workshop.

RNA-Seq De novo Assembly Using Trinity

Things to consider & current bugs:

- Strandedness (during library prep)
- Fastqc of raw RNASeq reads
- Trimming – outside of, or within trinity
- Paired or single end reads, or both
- Combining reads from various runs
- Pre-assembly error correction of raw reads
- Ribosomal removal
- NGS platform – Illumina, and no 454 and Pac bio can be done
- In silico normalization
- Genome-guided Trinity assembly

Trinity: Frequently Asked Questions

Trinity-home

- There are too many transcripts! What do I do?
- What computing resources are required?
- How long should this take?
- How can I run this in parallel on a computing grid?
- How do I identify the specific reads that were incorporated into the transcript assemblies?
- How do I combine multiple libraries in a single Trinity run? Or how do I combine paired and single reads?
- Trinity process died due to *std::bad_alloc*
- Butterfly fails with java Error: Cannot create GC thread. Out of system resources.
- How do I change the K-mer size?

Trinity – Assembly

Note

Breaking News for 2015

- Jan, 2015: Trinity moves to GITHUB. The new Trinity website is now trinityrnaseq.github.io
- Jan, 2015: Trinity user support and announcements will now occur through our Trinityrnaseq-users Google group: https://groups.google.com/forum/#!forum/trinityrnaseq-users.
- Jan, 2015: Trinity assembly services are now publicly accessible via the Trinity NCGAS Galaxy Portal. Simply upload your fastq files and run! This is our initial offering as part of a larger effort to build and release a Cancer Transcriptome Analysis Toolkit using Trinity, as described in the short video below:

Trinity Link: http://trinityrnaseq.github.io/

Trinity – Assembly of RNASeq data

Quick Guide for the Impatient

Trinity assembles transcript sequences from Illumina RNA-Seq data.

Download Trinity here.

Build Trinity by typing *make* in the base installation directory.

Assemble RNA-Seq data like so:

```
Trinity --seqType fq --left reads_1.fq --right reads_2.fq --CPU 6 --max_memory 20G
```

Find assembled transcripts as: *trinity_out_dir/Trinity.fasta*

Trinity Link: http://trinityrnaseq.github.io/

Trinity – Assembly of RNASeq data

Output of Trinity

When Trinity completes, it will create a *Trinity.fasta* output file in the *trinity_out_dir/* output directory (or output directory you specify).

Trinity groups transcripts into clusters based on shared sequence content. Such a transcript cluster is very loosely referred to as a *gene*. This information is encoded in the Trinity fasta accession. An example Fasta entry for one of the transcripts is formatted like so:

```
>TR1000|c115_g5_i1 len=247 path=[31015:0-148 23018:149-246]
AATCTTTTTTGGTATTGGCAGTACTGTGCTCTGGGTAGTGATTAGGGCAAAAGAAGACAC
ACAATAAAGAACCAGGTGTTAGACGTCAGCAAGTCAAGGCCTTGGTTCTCAGCAGACAGA
AGACAGCCCTTCTCAATCCTCATCCCTTCCCTGAACAGACATGTCTTCTGCAAGCTTCTC
CAAGTCAGTTGTTCACAGGAACATCATCAGAATAAATTTGAAATTATGATTAGTATCTGA
TAAAGCA
```

Trinity.fasta

The accession encodes the Trinity *gene* and *isoform* information. In the example above, the accession *TR1000|c115_g5_i1* indicates Trinity read cluster *TR1000|c115*, gene *g5*, and isoform *i1*. Because a given run of trinity involves many many clusters of reads, each of which are assembled separately, and because the *gene* numberings are unique within a given processed read cluster, the *gene* identifier should be considered an aggregate of the read cluster and corresponding gene identifier, which in this case would be *TR1000|c115_g5*.

So, in summary, the above example corresponds to *gene id: TR1000|c115_g5* encoding *isoform id: TR1000|c115_g5_i1*.

Obtain basic stats for the number of *genes* and *isoforms* and contiguity of the assembly by running:

Trinity Link: http://trinityrnaseq.github.io/

Trinity – Assembly of RNASeq data

There are so many output files…

What do I need for downstream analyses?

- Raw reads – the original Fastq trimmed reads
- Trinity assembly output Trinity.fasta file

Trinity Link: http://trinityrnaseq.github.io/

Trinity – Post-Assembly Transcriptome Analysis

- Abundance Estimation using RSEM or eXpress, and Visualization using IGV
- Differential Expression Analysis using Bioconductor
- Protein-coding Region Identification Using TransDecoder
- Functional Annotation Using Trinotate
- Gene Ontology functional category enrichment using GOseq and Trinotate
- Full-length Transcript Analysis

- Advanced Guide to Trinity
- Frequently Asked Questions

Post-Assembly Transcriptome Analysis Link: http://trinityrnaseq.github.io/

Trinity – Post-Assembly Transcriptome Analysis

Things to consider:

- Trinity genes vs Trinity transcripts/isoforms
- Biological replicates or none
- Interested in all or just a subset of genes/transcripts
- Cut offs and p-values
- Downstream annotation or just quantification of expression differences

---

Post-Assembly Transcriptome Analysis Link: http://trinityrnaseq.github.io/

Trinity: Abundance Estimation Using RSEM or eXpress, and Visualization Using IGV

Trinity-home

Trinity supports the use of either RSEM or eXpress for transcript abundance estimation (ie. computing expression values). Be sure to download the software and make it available via our PATH setting.

To compute abundance estimates, the original reads (not in silico normalized) are aligned to the Trinity transcripts using either bowtie(1) or bowtie2. Then, either RSEM or eXpress are executed to estimate expression values based on the resulting alignments.

Estimates & IGV link: http://trinityrnaseq.github.io/analysis/abundance_estimation.html

Typical assembly and abundance estimates stats

C14

	A	B	C	D	E	F	G	H	I
1	Table 1 Assembly Statistics	Total	Reduced (fpkm=1.5)			Table 2			
2	No. of raw trimmed reads	264505922 (~265M)				Length (bp)	Reduced (fpkm=1.5)		
3	No. of transcripts	126484	84124			20172	Genes	%	
4	No. of genes	31776	20172			200 - 500	6409	31.77	
5	Total assembled bases	125647941	48556932			501 - 1000	4351	21.57	
6	Avg (mean) transcript length (bp)	993.39	1528.1			1001 - 1500	2655	13.16	
7	Median transcript length (bp)	456	989			1501 - 2000	2022	10.02	
8	N50	1994	2545			2001 - 2500	1417	7.02	
9	GC content (%)	39.4	40.26			2501 - 3000	970	4.81	
10	Percent proper pairs	76.7	77.8			>3000	2349	11.64	
11	Samtools percent mapped and paire	75.1	76.55						
12						Length (bp)	Reduced (fpkm=1.5)		
13						31776	Genes	%	
14						200 - 500	9257	29.13	
15						501 - 1000	6745	21.23	
16						1001 - 1500	4348	13.68	
17						1501 - 2000	3212	10.11	
18						2001 - 2500	2282	7.18	
19						2501 - 3000	1637	5.15	
20						>3000	4296	13.52	

Trinity Link: http://trinityrnaseq.github.io/

Trinity: Identification and Analysis of Differentially Expressed Trinity Genes and Transcripts

EdgeR – No Biological Replicates
Bioconductor – biological Replicates

Trinity-home

Our current system for identifying differentially expressed transcripts relies on using the EdgeR Bioconductor package. We have a protocol and scripts described below for identifying differentially expressed transcripts and clustering transcripts according to expression profiles. This process is somewhat interactive, and described are automated approaches as well as manual approaches to refining gene clusters and examining their corresponding expression patterns.

We recommend generating a single Trinity assembly based on combining all reads across all samples as inputs. Then, reads are separately aligned back to the single Trinity assembly for downstream analyses of differential expression, according to our abundance estimation protocol. If you decide to assemble each sample separately, then you'll likely have difficulty comparing the results across the different samples due to differences in assembled transcript lengths and contiguity.

Note If you have biological replicates, align each replicate set of reads and estimate abundance values for the Trinity contigs independently.

Heat maps, volcano plots, median expression clusters

Differential Expression analyses:
http://trinityrnaseq.github.io/analysis/diff_expression_analysis.html

Trinity: Full-length transcript analysis for model and non-model organisms using BLAST+

One metric for evaluating the quality of a transcriptome assembly is to examine the number of transcripts that were assembled that appear to be full-length or nearly full-length. Such an analysis with a reference transcript set, such as from human or mouse, is relatively straightforward, since one can align the assembled transcripts to the reference transcripts and examine the length coverage. For non-model organisms, no such reference transcript set is available. If a high quality annotation exists for a closely related organism, then one might compare the assembled transcripts to that closely related transcriptome to examine full-length coverage. In other cases, a more general analysis to perform is to align the assembled transcripts against all known proteins and to determine the number of unique top matching proteins that align across more than X% of its length.

Trinity supports these analyses using BLAST+.

Useful protein databases to search include SwissProt and TrEMBL.

To examine the extent of top-matching BLAST alignments, first run BLAST, and then run the included analysis script below:

examine the percent of the target being aligned to by the best matching Trinity transcript

Build a blastable database:

```
%  makeblastdb -in uniprot_sprot.fasta -dbtype prot
```

Perform the blast search, reporting only the top alignment:

```
% blastx -query Trinity.fasta -db uniprot_sprot.fasta -out blastx.outfmt6 \
         -evalue 1e-20 -num_threads 6 -max_target_seqs 1 -outfmt 6
```

Full-length transcript analysis:
http://trinityrnaseq.github.io/analysis/full_length_transcript_analysis.html

Trinity: Post-Assembly Transcriptome Analysis

Things to consider & current bugs:

- Tree cut off – percentage or K-mer clusters
- P values
- Problems running some DE analysis on hicpu.q
- Updates to perl or R software in newer versions of Trinity

Differential Expression analyses:
http://trinityrnaseq.github.io/analysis/diff_expression_analysis.html

Trinity: Post-Assembly Transcriptome Analysis

There are so many output files…

What do I need for downstream analyses?

- Trinity assembly output: Trinity.fasta file

Transdecoder: Extract Likely Coding Sequences from Trinity Transcripts

Likely coding regions can be extracted from Trinity transcripts using TransDecoder, which comes included in the Trinity software distribution. The system works as follows:

- the longest ORF is identified within the Trinity transcript (if strand-specific, this is restricted to the top strand).
- of all the longest ORFs extracted, a subset corresponding to the very longest ones (and most likely to be genuine) are identified and used to parameterize a Markov model based on hexamers. These likely coding sequences are randomized to provide a sequence composition corresponding to non-coding sequence.
- all longest ORFs found are scored according to the Markov Model (log likelihood ratio based on coding/noncoding) in each of the six possible reading frames. If the putative ORF proper coding frame scores positive and is highest of the other presumed wrong reading frames, then that ORF is reported.
- if a high-scoring ORF is eclipsed by (fully contained within the span of) a longer ORF in a different reading frame, it is excluded.

Original link:
http://trinityrnaseq.github.io/analysis/extract_proteins_from_trinity_transcripts.html

Transdecoder – annotation and visualization

Output files explained

A working directory (ex. transcripts.transdecoder_dir/) is created to run and store intermediate parts of the pipeline, and contains:

```
longest_orfs.pep   : all ORFs meeting the minimum length criteria, regardless of coding potential.
longest_orfs.gff3  : positions of all ORFs as found in the target transcripts
longest_orfs.cds   : the nucleotide coding sequence for all detected ORFs

longest_orfs.cds.top_500_longest   : the top 500 longest ORFs, used for training a Markov model for coding
sequences.

hexamer.scores                     : log likelihood score for each k-mer  (coding/random)

longest_orfs.cds.scores            : the log likelihood sum scores for each ORF across each of the 6 reading
frames
longest_orfs.cds.scores.selected   : the accessions of the ORFs that were selected based on the scoring
criteria (described at top)
longest_orfs.cds.best_candidates.gff3              : the positions of the selected ORFs in transcripts
```

Many applications for gene discovery

New Link: http://transdecoder.github.io/

Transdecoder – for downstream analyses using Trinotate

Then, the final outputs are reported in your current working directory:

```
transcripts.fasta.transdecoder.pep : peptide sequences for the final candidate ORFs; all shorter candidates
within longer ORFs were removed.
transcripts.fasta.transdecoder.cds  : nucleotide sequences for coding regions of the final candidate ORFs
transcripts.fasta.transdecoder.gff3 : positions within the target transcripts of the final selected ORFs
transcripts.fasta.transdecoder.bed  : bed-formatted file describing ORF positions, best for viewing using
GenomeView or IGV.
```

View ORFs in the context of the transcript structures in the reference genome:
IGV link: https://www.broadinstitute.org/software/igv/home

If you lack a genome sequence and are working exclusively with the target transcripts, you can load the transcript fasta file and the ORF predictions (bed file) into GenomeView link: http://genomeview.org/

Eg. http://transdecoder.github.io/

Transdecoder – for downstream analyses using Trinotate

There are so many output files…

What do I need for downstream analyses?

- Trinity assembly output: Trinity.fasta file
- Peptide sequences for final candidate ORFS: Trinity.fasta.transdecoder.pep file

Trinotate: Transcriptome Functional Annotation & Analysis

Automated Higher Order Biological Analysis

Note **Trinotate Breaking News**

- Feb, 2015: Trinotate moves to Github
- Feb, 2015: Trinotate release v2.0 is now available (see below)
- Feb, 2015: TrinotateWeb is becoming more useful, the interface was given a facelift including other various performance improvements.

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/Uniref90), protein domain identification (HMMER/PFAM), protein signal peptide and transmembrane domain prediction (singalP/tmHMM), and comparison to currently curated annotation databases (EMBL Uniprot eggNOG/GO Pathways 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 Link: http://trinotate.github.io/

Trinotate: Transcriptome Functional Annotation & Analysis

All functional annotation data derived from the analysis of transcripts is integrated into a SQLite database which allows:

1) fast efficient searching for terms with specific qualities related to a desired scientific hypothesis

1) or a means to create a whole annotation report for a transcriptome.

Trinotate Link: http://trinotate.github.io/

Trinotate: Transcriptome Functional Annotation & Analysis

NCBI BLAST

HMMER

Pfam

UniProt

eggNOG version 3.0

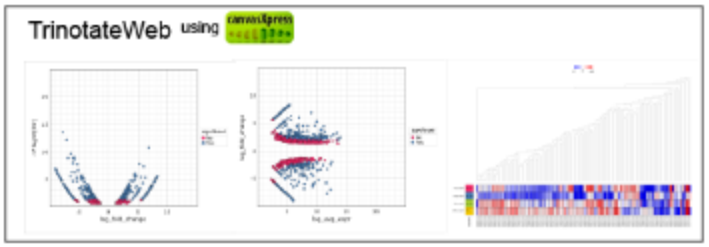
SQLite

the Gene Ontology

RNA-Seq ➡ Trinity ➡ Transcripts/Proteins ➡ Functional Data ➡ Discovery

Trinotate Link: http://trinotate.github.io/

Trinotate: Transcriptome Functional Annotation & Analysis



1. Table of Contents

Software and database: required and optional:
Trinotate Link: http://trinotate.github.io/

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 ar
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_trembl_blastp <file>
  LOAD_pfam <file>
  LOAD_tmhmm <file>
  LOAD_signalp <file>
  ```

- Trinity transcript search results:

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

- 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.

Trinotate Link: http://trinotate.github.io/

Trinotate: Transcriptome Functional Annotation & Analysis

Things to consider & current bugs:

- What functional annotations are you interested in?
- P-values and cutoffs
- Can't run the Trinotate sqlite database from a temporary dir, only a root
- Only NCBI is available in the Trinity package on lattice
- Need to download several of the other db locally
- Hmmpress doesn't work – get error
- Most steps take 1 – 5 hours
- Blastp and Blastx steps take anywhere from 5 hours to >5 days

Trinotate Link: http://trinotate.github.io/

Trinotate: Output an Annotation Report

The output has the following column headers:

```
0       #gene_id
1       transcript_id
2       Top_BLASTX_hit
3       RNAMMER
4       prot_id
5       prot_coords
6       Top_BLASTP_hit
7       Pfam
8       SignalP
9       TmHMM
10      eggnog
11      gene_ontology
```

and the data are formatted like so:

```
# example protein-coding transcript
0       comp66_c0
1       comp66_c0_seq1
2       sp|Q09739|MCP7_SCHPO^Q09739^Q:1-519,H:15-187^100%ID^E:5e-95^RecName: Full=Meiotic coiled-coil protein
regulated gene 32 protein;^Eukaryota; Fungi; Dikarya; Ascomycota; Taphrinomycotina; Schizosaccharomycetes; Sch
Schizosaccharomycetaceae; Schizosaccharomyces.
3       .
4       m.4
5       1-522[+]
6       sp|Q5XGY9|MND1_XENLA^Q5XGY9^Q:4-169,H:19-181^37.13%ID^E:7e-24^RecName: Full=Meiotic nuclear division p
Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Amphibia; Batrachia; Anura; Pipoidea; Pipidae; Xenopodi
Xenopus.`sp|Q9BWT6|MND1_HUMAN^Q9BWT6^Q:4-169,H:19-181^38.32%ID^E:1e-23^RecName: Full=Meiotic nuclear division
```

Trinotate Link: http://trinotate.github.io/

Typical output: Trinotate.annotation.report.xls

Q2 fx .

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	#gene_	transcri	sprot_T	TrEMBL	RNAMM	prot_id	prot_co	sprot_T	TrEMBL	Pfam	SignalP	TmHMM	eggnog	gene_o	gene_
2	c5102_g1	c5102_g1_	sp|U3JAG	UniRef90_	.	c5102_g1_	65-1981[-]	sp|U3JAG	UniRef90_	.	.	.	.	GO:00602	.
3	c29780_g2	c29780_g2	sp|Q9ZQ8	UniRef90_	.	c29780_g2	519-1826[-	sp|Q5MAI	UniRef90_	PF09423.5	.	.	COG1409^	GO:00055	GO:00
4	c29780_g2	c29780_g2	sp|Q9ZQ8	UniRef90_	.	c29780_g2	59-586[+]	sp|Q9LMC	UniRef90_	.	sigP:1^29	.	.	GO:00095	.
5	c15084_g1	c15084_g1	sp|Q9ZQ8	UniRef90_	.	c15084_g1	164-1918[-	sp|Q9ZQ8	UniRef90_	PF00149.2	sigP:1^20	.	COG1409^	GO:00055	GO:00
6	c791_g1	c791_g1_i	sp|Q9ZPY	UniRef90_	.	c791_g1_i	2-364[-]	sp|Q9ZPY	UniRef90_	PF07731.9	.	.	COG2132^	GO:00480	GO:00
7	c25103_g1	c25103_g1	sp|Q9ZNV	UniRef90_	.	c25103_g1	36-584[+]	sp|Q9ZNV	UniRef90_	PF01161.1	sigP:1^19	.	.	GO:00057	.
8	c21449_g1	c21449_g1	sp|Q9ZD4	UniRef90_	.	.	.	.	.	.	.	.	COG0517^	GO:00055	.
9	c20979_g1	c20979_g1	sp|Q9ZAE	UniRef90_	.	c20979_g1	1-1017[+]	sp|D4GU7	UniRef90_	PF04321.1	.	.	.	GO:00506	GO:00
10	c18122_g1	c18122_g1	sp|Q9Z35	UniRef90_	.	c18122_g1	140-586[+	sp|P5669	UniRef90_	PF03520.9	.	.	COG1226^	GO:00099	.
11	c28763_g1	c28763_g1	sp|Q9Z34	UniRef90_	.	c28763_g1	1800-6086	sp|Q9Z34	UniRef90_	PF12053.3	.	.	.	GO:00451	GO:00
12	c24020_g1	c24020_g1	sp|Q9Z30	UniRef90_	.	c24020_g1	4506-5093	sp|Q9Z30	UniRef90_	PF13833.1	.	.	COG5126^	GO:00059	GO:00
13	c24020_g1	c24020_g1	sp|Q9Z30	UniRef90_	.	c24020_g1	4491-5078	sp|Q9Z30	UniRef90_	PF13833.1	.	.	COG5126^	GO:00059	GO:00
14	c29561_g1	c29561_g1	sp|Q9Z2Z	UniRef90_	.	c29561_g1	1524-3575	sp|Q9Z2Z	UniRef90_	PF00310.1	.	.	COG0449^	GO:00302	GO:00
15	c29561_g1	c29561_g1	sp|Q9Z2Z	UniRef90_	.	c29561_g1	1516-3567	sp|Q9Z2Z	UniRef90_	PF00310.1	.	.	COG0449^	GO:00302	GO:00
16	c29561_g1	c29561_g1	sp|Q9Z2Z	UniRef90_	.	c29561_g1	1511-3562	sp|Q9Z2Z	UniRef90_	PF00310.1	.	.	COG0449^	GO:00302	GO:00
17	c29561_g1	c29561_g1	sp|Q9Z2Z	UniRef90_	.	c29561_g1	1502-3553	sp|Q9Z2Z	UniRef90_	PF00310.1	.	.	COG0449^	GO:00302	GO:00
18	c29489_g1	c29489_g1	sp|Q9Z2Z	UniRef90_	.	c29489_g1	239-1207[-	sp|Q9Z2Z	UniRef90_	PF00153.2	.	.	.	GO:00160	.
19	c27689_g1	c27689_g1	sp|Q9Z2X	UniRef90_	.	c27689_g1	548-2263[-	sp|Q9Z2X	UniRef90_	PF00651.2	.	.	.	GO:00058	GO:00
20	c23982_g1	c23982_g1	sp|Q9Z2X	UniRef90_	.	c23982_g1	379-1095[-	sp|Q9Z2X	UniRef90_	PF13637.1	.	.	COG0666^	GO:00156	GO:00
21	c29515_g1	c29515_g1	sp|Q9Z2W	UniRef90_	.	c29515_g1	496-2946[-	sp|Q9Z2W	UniRef90_	PF10613.4	.	.	.	GO:00322	GO:00
22	c18846_g1	c18846_g1	sp|Q9Z2U	UniRef90_	.	c18846_g1	177-908[-]	sp|Q9Z2U	UniRef90_	PF10584.4	.	.	COG0638^	GO:00058	GO:00
23	c19109_g1	c19109_g1	sp|Q9Z2Q	UniRef90_	.	c19109_g1	301-897[-]	sp|Q9Z2Q	UniRef90_	PF09812.4	.	.	.	GO:00057	.
24	c30037_g1	c30037_g1	sp|Q9Z2Q	UniRef90_	.	c30037_g1	3-2006[+]	sp|Q9Z2Q	UniRef90_	PF02607.1	.	.	COG1410^	GO:00057	GO:00
25	c13372_g1	c13372_g1	sp|Q9Z2L	UniRef90_	.	c13372_g1	3-920[-]	sp|Q9Z2L	UniRef90_	PF01442.1	.	.	NOG7141	GO:00057	GO:00

annotation_report_r2r3r4_f

Ready

Post-Trinotate Analysis

So, now what to I do with this huge Trinotate.annotation.report.xls file???

It depends on what/how you want to annotate

dictionary definition of "to annotate":

– "to make or furnish critical or explanatory notes or comment"

- **some of what this includes for genomics/transcriptomics/proteomics**

– gene product names

– functional characteristics of gene products

– physical characteristics of gene/protein/genome

– overall metabolic profile of the organism

- **elements of the annotation process**

– gene finding

– homology searches

– functional assignment

– ORF management

– data availability

- **manual vs. automatic**

– automatic = computer makes the decisions

- good on easy ones
- bad on hard ones

– manual = human makes the decisions

- highest quality

**Due to the VOLUMES of genome data today, most genome projects are annotated primarily using automated methods with limited manual annotation

Source: http://cpgr.plantbiology.msu.edu/training/workshop_mar07/Lecture3_GenomeAnnotation.pdf

But in the absence of a genome…

You are mostly on your own …

Publications citing Trinotate:

Brehkman et al. 2015. *Transcriptome profiling of the dynamic life cycle of the scypohozoan jellyfish* Aurelia aurita
- With coauthor Brian Haas (Trinity inventor)

Ghaffari et al. 2014. *Novel transcriptome assembly and improved annotation of the whiteleg shrimp (*Litopenaeus vannamei), *a dominant crustacean in global seafood mariculture*

If you know any others please let me know.

Gene Ontology Enrichment using Trinotate and GOseq

Extract GO assignments per gene

For biological replicates only

Extract all GO assignments for each gene feature, including all parent terms within the GO DAG, using a script included in Trinotate (not Trinity) like so:

```
${TRINOTATE_HOME}/util/extract_GO_assignments_from_Trinotate_xls.pl --Trinotate_xls trinotate.xls \
                                                                    -G --include_ancestral_terms \
                                                                  > go_annotations.txt
```

Run GOseq

The Bioconductor package GOseq can then be used to perform functional enrichment tests, like so:

```
${TRINITY_HOME}/Analysis/DifferentialExpression/run_GOseq.pl --factor_labeling  factor_labeling.txt \
                                                             --GO_assignments go_annotations.txt \
                                                             --lengths gene.lengths.txt
```

Note, the above script is here in the Trinity software package (not Trinotate).

The *factor_labeling.txt* file should be of format:

```
gene_id (tab) factor
```

where factor is a string describing that subset of genes. (Note, future versions will require formatting *factor (tab) gene_id* instead).

For example:

```
my_gene_A (tab) diff_expressed_cond_X_Y
my_gene_B (tab) diff_expressed_cond_X_Y
...
my_gene_M (tab) diff_cond_W_Z
my_gene_N (tab) diff_cond_W_Z
...
```

GO enrichment using Trinotate and GOSeq link:
http://trinityrnaseq.sourceforge.net/analysis/run_GOseq.html

Address questions about how a specific set of candidate genes is expressed across multiple samples:

- Biological replicates
- Various anatomical structures or cell types
- Environmental replicates
- Time course data

How to reference Trinity Suite software

- Visit the website for each component
- E.g., Trinotate

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.

Trinotate Link: http://trinotate.github.io/

Some lasts bits of advice (speaking from experience…):

- Organize your files from the start
- Keep track of your files
- Back up your files regularly
- Get rid of unnecessary files
- Check in with Paul Frandsen on a regular basis
- Learn Unix and basic bash scripting
- Become a polyglot: learn Python or Perl, and R languages

Where to go for help:

1. Google
2. SEQanswers forum: http://seqanswers.com/
3. Online Forums related to the Trinity suite software:

Trinity user support and announcements will now occur through our Trinityrnaseq-users Google group: https://groups.google.com/forum/#!forum/trinityrnaseq-users.

Technical Support and Project Announcements
Join our TransDecoder google group at
https://groups.google.com/forum/#!forum/transdecoder-users

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

4. OCIO - Office of Research Information Services

Thanks for your help:

Vanessa Gonzalez

DJ Ding

Paul Frandsen

QUESTIONS???