Command Line BLAST in the Pringle Lab

Lisl Esherick, January 15, 2015open

**Genomic and Transcriptomic Resources for *Aiptasia***

The Pringle Lab and the Voolstra Lab at KAUST have generated several transcriptomes for *Aiptasia* as well as the *Aiptasia* genome. Each of these resources is stored in a separate FASTA file, with unique identifiers to describe each sequence in the file. For the most part they were generated from different sequencing libraries, using slightly different assembly algorithms, and all have their own advantages and disadvantages. Here is a table summarizing some key information about the datasets that I think are the most useful.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | File Name | GeneIDs | Assembly Algorithm | Published In |
| Erik’s Transcriptome | SymTranscriptsClustered… | Locus#\_Transcript#/# | Velvet/Oases | Lehnert *et al.* 2014 |
| Seb’s Transcriptome | aiptasia\_transcriptome\_final.fa | comp#\_c#\_seq# | Trinity | Baumgarten *et al.* 2015 |
| Genomic gene models (AA) | aiptasia.genome-models.aa.fa | AIPGENE# | mapping/  *ab initio* prediction | Baumgarten *et al.* 2015 |
| Genomic gene models (mRNA) | aiptasia.genome-models.mRNA.fa | AIPGENE# | mapping/  *ab initio* prediction | Baumgarten *et al.* 2015 |
| The Genome | aiptasia\_genome\_final\_clean.fa | scaffold#|size# | AllPath | Baumgarten *et al.* 2015 |
| Cawa’s larval CC7 x H2 apo Day 3 | LarvalTranscriptomeFinal.fa |  |  | Not published |

Erik’s transcriptome was assembled from Illumina paired-end reads generated from symbiotic adult CC7 anemones.

Seb’s transcriptome was assembled from Illumina paired-end reads generated from 10 day old aposymbiotic larvae, 10 day old symbiotic larvae, aposymbiotic adults, intermediately infected adults (day12), and symbiotic adult anemones – all CC7. Reads generated from symbiotic anemones were mapped to the *Symbiodinium* reference transcriptome to eliminate contamination.

The genomic gene models were created by mapping RNA-seq reads to the genome to generate a training set for *ab initio* gene prediction software.

The genome was assembled from Illumina paired-end reads generated from aposymbiotic CC7 anemones.

**Formatting BLAST databases.**

Before being able to search against a transcriptomic or genomic database, you have to format the database so it is searchable by the BLAST software. This, and all other BLAST related activity will happen in Terminal.

1. Using “cd” navigate to the directory (folder) that the database is in. For me, that is:

cd Documents/Pringle\ Lab/BLAST/

2. Use the command “makeblastdb” to format the database. “makeblastdb” requires two arguments: -in to specify your input file (the transcriptome/genome .fa file) and –dbtype to specify whether it is an amino acid or nucleotide file (“prot” or “nucl”). For the genomic mRNA gene models you would type:

makeblastdb -in aiptasia.genome-models.mRNA.fa -dbtype nucl

That’s it you’re done!

**Searching BLAST databases.**

To search BLAST databases you will need a FASTA file of the sequence you are searching with – your query sequence, a formatted database you are searching against, and the name of the file you want your output stored in (the file doesn’t have to exist already, BLAST will create it for you).

1. If you haven’t already got a FASTA file of your query sequence, download or create one. For practice we’ll use Human beta-actin, available here:

[http://www.ncbi.nlm.nih.gov/nuccore/168480144?report=FASTA](http://www.ncbi.nlm.nih.gov/nuccore/168480144?report=fasta)

2. In general, when I am BLAST-ing an aiptasia nucleotide database with an aiptasia nucleotide query sequence I use blastn (searches nucleotide vs. nucleotide). When using a query sequence from another organism, especially a distantly related one like human, I use tblastx (translates the query sequence and database to amino acid sequence in all 6 frames, then searches).

For human actin we will use the command “tblastx.” I always give at least three arguments –query followed by my query sequence FASTA file, -db followed by the database FASTA file, -out followed by my output file. For example:

-query hsa\_actin.fa -db aiptasia\_transcriptome\_final.fa -out hsa\_actin.blastout.txt

3. The regular output file will contain the following information:

The version of BLAST you used

The database you searched against

The query sequence(s) you used followed by a list of hits with BLAST scores and E-values

Alignments for each hit.

You can open your output file in TextEdit either through Finder, or by typing the command:

open hsa\_actin.blastout.txt

To open in Terminal, I use:

less hsa\_actin.blastout.txt

**Reciprocal BLAST.**

Often when searching the *Aiptasia* databases you will get hits with strong similarity to your gene of interest, but they may not be true paralogues of that gene (for the sake of simplicity I am going to say we are looking for true paralogues, though you may not always be).

To confirm that a hit I have is really what I am looking for, I take the sequence of the hit from the database file (if it’s just one or two hits I copy and paste), and search it against a publicly available database at NCBI. I prefer to use BLASTX or BLASTP against nr or swissprot/uniprot.

**Slightly Fancier BLAST**

There will be times when you want to BLAST multiple sequences against an *Aiptasia* database at the same time. To do this, I create a FASTA file containing all of my query sequences and use that to search against the database, as above. The more sequences you use, the more tiresome it becomes to parse the regular output by reading through and copy/pasting the sequences your interested in.

You can limit the BLAST output in several ways. The ones I use most frequently are the arguments

-evalue “E-value cutoff” i.e. 1e-5

-max\_target\_seqs “Number of maximum hits to return” i.e. 10

You can also change the formatting of the BLAST output to make it easier to parse using Perl or Python scripts. I use -outfmt 6 for tab delimited output (outputs the query sequence, hit, BLAST score etc. on a single line separated by tabs). The order of the output is:

query id, database sequence (subject) id, percent identity, alignment length, number of mismatches, number of gap openings, query start, query end, subject start, subject end, Expect value, HSP bit score

You can learn all about the arguments you can supply to BLAST by typing

blastn -help

Once you have a tab-delimited output, you can write or use custom Perl or Python scripts to analyze the output in anyway, or pull or genes of interest. Here is a protocol I used while searching for paralogues to certain genes of interest for the genome paper. As an example let’s search for Caspases 3, 7, 8, and 9.

1. I created a FASTA file of the human Caspases downloaded from Uniprot. Because they are protein sequences we will use tblastn (protein query, translated nucleotide database). We will impose some cutoffs: E-value cutoff of 1e-5, only 5 hits per query sequence.

tblastn -query hsa\_caspases.fa -db aiptasia\_transcriptome\_final.fa -out hsa\_caspases.blastout.txt -outfmt 6 -evalue 1e-5 -max\_target\_seqs 5

2. Rather than search through the transcriptome, copying and pasting, for all of our hits, we’ll use a Python script I wrote to create a FASTA file from a tab-delimited BLAST output file. (Note: I wrote this script quickly, and for a very specific purpose, you may have different purposes and may want to write your own script to suit them!).

The script takes three arguments: the blast output file, the query fasta file, the database fasta file. Use the “>” operator to send the output to a file of your choice.

python fasta\_from\_blastout.py hsa\_caspases.blastout.txt hsa\_caspases.fa aiptasia\_transcriptome\_final.fa > aip\_v\_hsa\_caspases.fa

3. Now you have a FASTA file containing *Aiptasia* sequences with similarity to Caspases 3, 7, 8, 9. Do with it what you will! (I would recommend starting with reciprocal BLAST again).