NCGAS Makes Robust Transcriptome Assembly Easier with a Readily Usable Workflow Following *de novo* Assembly Best Practices

Sheri Sanders
Bioinformatics Analyst
NCGAS @ IU
ss93@iu.edu

NATIONAL CENTER FOR GENOME ANALYSIS SUPPORT

Problem

Many users new to *de novo* assemblies gravitate toward Trinity for its ease of use - but…

- One assembler is going to miss some things and be biased in one way or another
- Trinity can give large numbers of false positives – which is great if you have a good idea on how to filter/curate
- It is generally a good idea to use multiple kmers to capture different information, which one trinity run will not accomplish!
- **Degree of ease should not dictate analysis for a project!**

Assembler

CDTA – Combined *de novo* Transcriptome Assembly

- Multiple assemblers, multiple parameters (kmers)
- Best of all worlds
- Get as much data as possible and look for concordance between the different assemblers.
  - It is less likely that different assembly algorithms will experience the same biases/errors in assembly.
  - Not always needed…

https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-014-1192-7

Why we generally do this…

- In several projects (particularly in large or polyploid systems), we were not recovering transcripts we knew were expressed – we had qPCR to back them up! No one assembler got all the target transcripts – the CDTA did!
- We've seen quality increases in the transcriptome when we run this pipeline.
- It has been published in best practices for RNA-seq to use multiple parameters at least.
- It's easier to defend in publication!

Workflow Overview

- Trinity (k=25)
- SOAPdenovo (k=35,45,55,65,75,85)
- Velvet (k=35,45,55,65,75,85)
- TransAbyss (k=45,55,65,76,85)

19 different assemblies!

- Combine with Evigenes

All these Kmers – why?

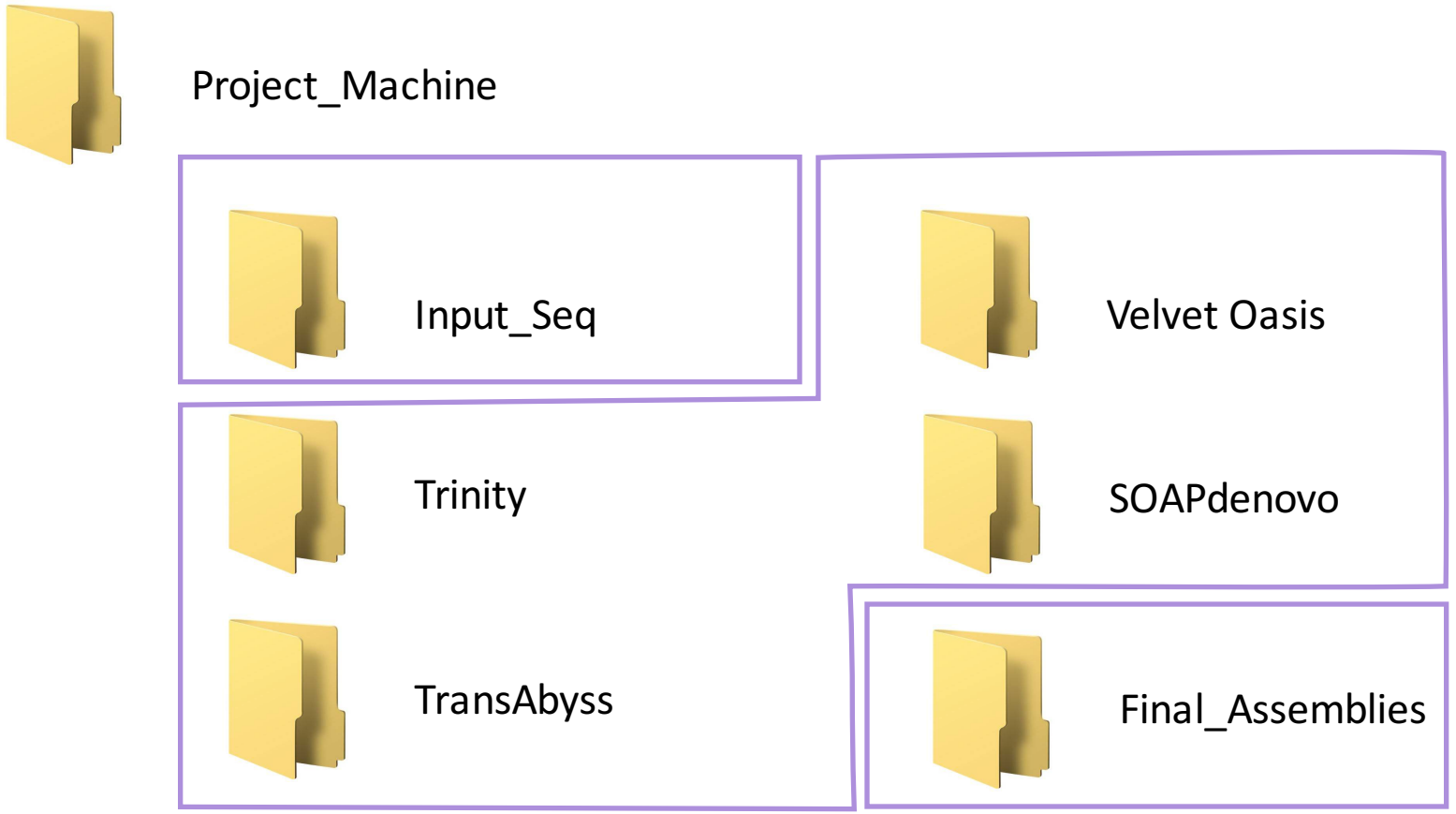
The structure of an assembly graph is highly dependent on the k-mer size used for assembly. Small k-mers result in shorter contigs with lots of connections, while large k-mers can result in longer contigs with fewer connections.

- longer reads and/or higher read depth → you can use larger k-mers which are useful in resolving complex areas of the graph (i.e. advantages of PacBio vs Illumina in genome assembly).
- shorter reads and/or lower read depth → you may have to use shorter k-mers to build a more complex graph.

Often we use several and combine to gain information from a range of kmers, because estimating an optimum can be difficult.

See https://github.com/rrwick/Bandage/wiki/Effect-of-kmer-size for a great write up on this.

What this looks like in practice



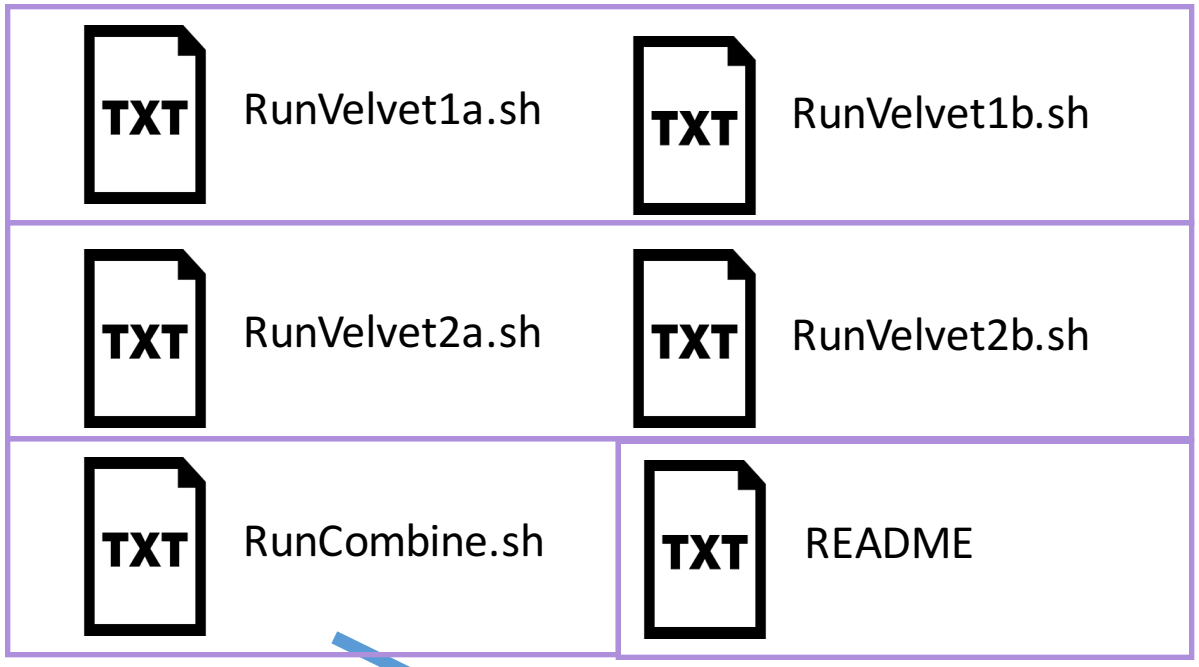
Two commands to run out of the README to set up:

A. Correct the email in each job file
B. Fill in the absolute path where necessary

1. Place input files
2. Run all assemblies
3. Combine

Structure

Velvet



Final_Assemblies directory

In each directory, there are job files for each assembler.

Available for SLURM (PSC Bridges) and Torque (IU Machines).

Each set of numbers can be run in parallel.

After all kmers are complete, output is labeled and compiled.

Because of set directory structure, no need to specify where inputs are, where output goes etc.

READMEs have instructions, documentation, and tips.

Commands

```
Step 0:
You need to put your email as the point of contact for all the scripts.  This can be done en masse with the following command
(replace youremail@wherever.com with your actual email) from the Project directory:
for f in */Run*; do sed -i 's/YOUREMAILHERE/youremail@wherever.com/g' $f; done

You will also need to map your project directory to the run files.  To do this, run the following from the Project directory:
for f in *; do p=`pwd`; sed -i "s|PWDHERE|$p|g" $f/* ; done
```

```
Step 1:
Put all your reads into input_files
Read the README in input_files to get instructions for combining reads properly into input files.
You can do this with symlink (use command "man ln" if you are unfamiliar with this command).
```

Commands

```
Step 2: SOAP
Run RunSOAP1.sh and RunSOAP1b.sh at the same time.
Command: qsub RunSOAP1*
When they finish, run ./Combine.sh
Command: ./Combine.sh

Step 2b: Velvet
Run RunVelvet1.sh and RunVelvet1b.sh at the same time.
Command: qsub RunVelvet1*
When BOTH above are complete, run RunVelvet2.sh and RunVelvet2b.sh at the same time.
Command: qsub RunVelvet2*
When BOTH above are complete, run RunVelvet3.sh and RunVelvet3b.sh at the same time.  When they finish, run ./Combine.sh (no n
eed to submit to queue).
Command: qsub RunVelvet3*
When they finish, run ./Combine.sh
Command: ./Combine.sh

Step 2c: TransAbyss
Run RunTransAb1.sh and RunTransAb1b.sh at the same time.
Command: qsub RunTransAbyss1*
When they finish, run ./Combine.sh
Command: ./Combine.sh

Step 2d: Trinity
Run RunTrinity.sh, there is no combine script for this assembler.
Command: qsub RunTrinity.sh
```

Cleaning it up with Evigenes

Evidential Genes

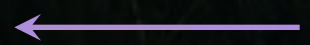
- Leverages fastanrdb, CD-hit, Cd-hit-est, and blast
  - Removes perfect redundancy (fastanrdb)
  - Removes perfect fragments (cd-hit-est)
  - Uses blastn to find 98% identity, exon sized alignments (blast)

- Identifies full length cds and identifies transcript quality to identify main ("okay"), alternative ("okalt"), and dropped ("drop") sets.

- See eugenes website for more details!
- NOTE: This is not what I call a totally friendly program to use…

Commands

```
Step 3: Combine all outputs
The outputs for each combined set will be placed automatically in final_assembly.
Run ./Combine.sh FIRST to get one input for Evigenes
Run RunEviGene.sh
Command: ./Combine.sh; qsub RunEviGene.sh
```



```
OUTPUT:
In final_assemblies, you will see the following directories:
        okayset - where the good files are
        dropset - where dropped files are

within okayset, you will set two sets of files:
        okay.fa/aa/cds - these are the highest quality transcripts
                anything labeled "complete" is a full cds
        okalt.fa/aa/cds - these are the alternative versions of the transcripts in the okay file (alleles, isoforms, etc).
                anything labeled "complete" is a full cds.

SEE http://arthropods.eugenes.org/EvidentialGene/trassembly.html for documentaiton!
```

```
dropset   litF.fa              litF.tr2aacds.log  okayset     SOAP.fa   TransAb.fa  Velvet.fa
inputset  litFnrcd1_db.perf    litF.trclass       RunEviGene  tmpfiles  Trinity.fa
```

Benefits

- Pretty much filters for you – usually I end up with the expected 20-30k transcripts in the “okay complete” set.
- You get a separate file with all the alternatives, tagged with which gene in the okay set they are associated with. This is nice if you want this data!
- Automatically gives you cds, aa, and fa formats
- Replicability is high for a filtering paradigm

- **You start with working scripts that you can easily change, with documentation.**

Output Numbers

Polyploid plant – input 62,847,654 paired end
- Okay – 50,091
- Okay complete - 28,021
- Okalt - 135,248
- Okalt complete - 56,538

Polyploid plant – input 37,609,484 paired end
- Okay - 44,274
- Okay complete - 25,321
- Okalt - 132,894
- Okalt complete - 58,483

Salamander: 83,468,758 paired end input

- Just trinity: 110,973

- Okay – 45,816
- Okay complete – 28,792
- Okalt – 30,053
- Okalt complete –15,384

Early user reception

Great for us – NCGAS does a LOT of transcriptome assemblies

- Sweet potato, coffee, peanut, fly, frog, daphnia…

Working well for our users

- “The use of this pipeline has **saved me tons of time** from having to figure out the script for each assembly program and it is VERY easy to use, especially for a **person like myself who barely understands Linux!**”

- “If this pipeline was not available, I would have **most likely used only only package and at one kmer size** for my assembly, and it would have have **probably taken me just as long** to figure out and run. ”

How do you get all this?

- Github – Torque and SLURM versions available
- Tutorial will be on our website soon.
- Contact me (help@ncgas.org)!

Acknowledgements

NCGAS Staff
- Carrie Ganote (IU)
- Bhavya Papudeshi (IU)
- Tom Doak (IU)
- Phil Blood (PSC)

Collaborators
- Keithanne Mockaitis
- Heather Walsh
- Juliet Wong

Others
- Don Gilbert

WORKSHOP!

We are hosting a two (full) day workshop on doing *de novo* Transcriptome Assembly using HPC resources in March!

When: March 19-20th
How to Apply: TBA, email ss93@iu.edu