ECOGEO 'Omics Training: 4.1 Assembly Version 2

Frank Aylward, Daniel Mende

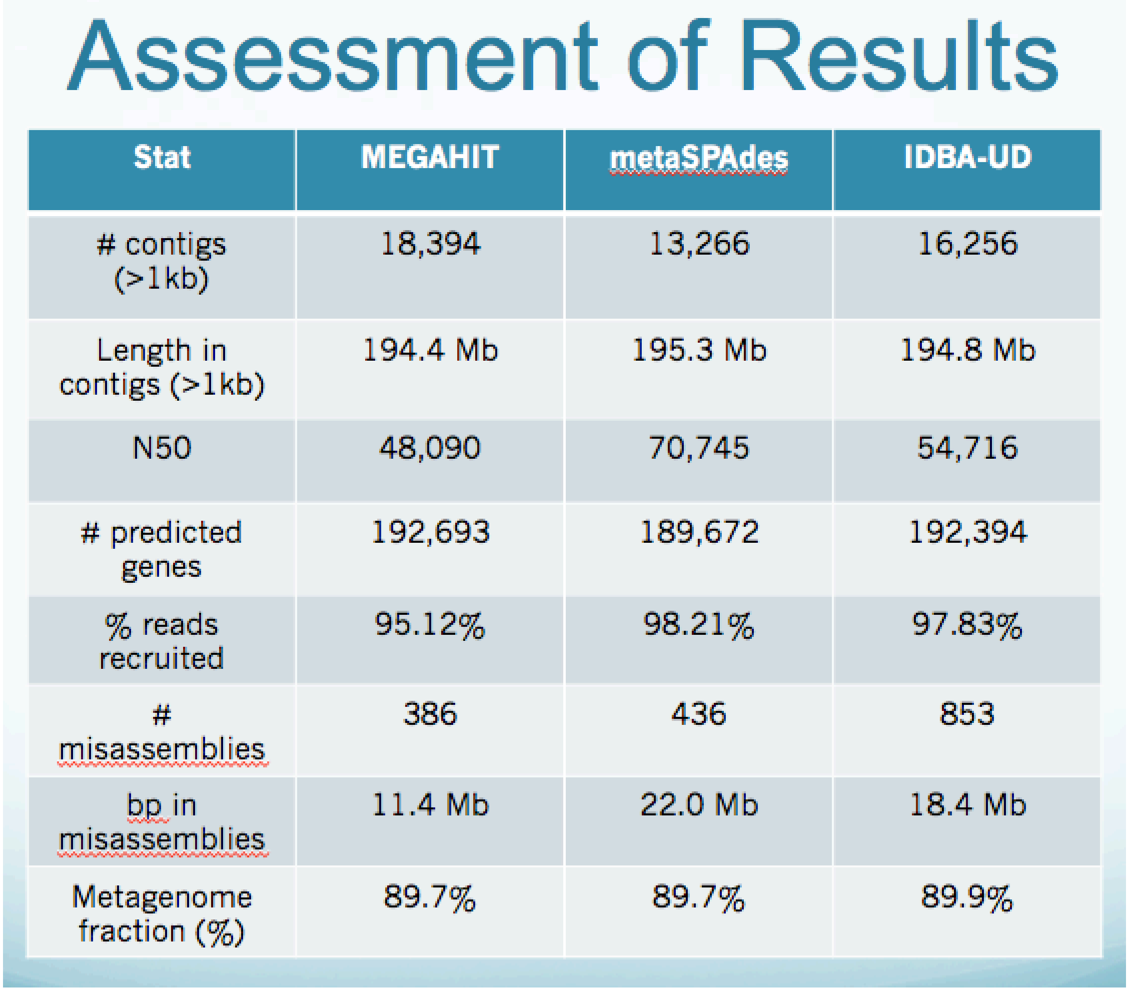
Abstract

Provides a short introduction to MEGAHIT, IDBA-UD, and SPAdes assemblers, a demo on Prodigal Gene Caller, and determining % of reads and contig coverage using Bowtie2 short read aligner.

Open this protocol inside the virtual machine (details in 'Start Instructions') for easy copy, paste of commands into the command line terminal window.

Citation: Frank Aylward, Daniel Mende ECOGEO 'Omics Training: 4.1 Assembly. **protocols.io**
dx.doi.org/10.17504/protocols.io.fi6bkhe
Published: 18 Aug 2016

Guidelines



Assessment of Results

Stat	MEGAHIT	metaSPAdes	IDBA-UD
# contigs (>1kb)	18,394	13,266	16,256
Length in contigs (>1kb)	194.4 Mb	195.3 Mb	194.8 Mb
N50	48,090	70,745	54,716
# predicted genes	192,693	189,672	192,394
% reads recruited	95.12%	98.21%	97.83%
# misassemblies	386	436	853
bp in misassemblies	11.4 Mb	22.0 Mb	18.4 Mb
Metagenome fraction (%)	89.7%	89.7%	89.9%

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Before start

Before starting, please visit the ECOGEO website for more information on this "Introduction to Environmental 'Omics" training series. The site contains a pre-packaged virtual machine that can be downloaded and used to run all of the protocols in this protocols.io collection. In addition to the VM, the website contains video and presentations from our initial "Intro to Env 'Omics" workshop held at the Univ. of Hawai'i at Manoa on 25-26 Jul 2016.

Please email ‘ecogeo-join@earthcube.org’ to join the ECOGEO listserv for future updates.

Protocol

Introduction to assemblers

Step 1.

Move to directory containing assemblers.

cmd COMMAND

```
$ cd /home/c-debi/ecogeo/assembly
```

Introduction to assemblers

Step 2.

View assembler parameters for MEGAHIT v1.0.3, IDBA-UD v1.1.1, and SPAdes v3.7.1

cmd COMMAND

```
$ megahit
$ idba_ud
$ spades.py
```

These commands will show parameters for each assembler.

Introduction to assemblers

Step 3.

Trimmomatic Quality Control:

cmd COMMAND

```
$ java -
jar trimmomatic-0.35.jar PE SRR606249_R1.fastq SRR606249_R2.fastq R1_pe R1_se R2_pe R2_se I
LLUMINACLIP:TruSeq3-PE.fa:2:30:10 SLIDINGWINDOW:10:28 MINLEN:50
```

This step has already been completed for you. PLEASE NOTE: Commands in black on the presentation, video should NOT be executed in the VM (assembly steps require more computational power).

Introduction to assemblers

Step 4.

Assemble with Megahit:

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

cmd COMMAND

```
$ megahit --preset meta-sensitive -1 SRR606249.trim_R1.fastq -2 SRR606249.trim_R2.fastq -o SRR606249.megahit_asm
```

This step has already been completed for you and the command does NOT need to be executed again.

Introduction to assemblers

Step 5.

IDBA-UD: merge FASTQ files to interleaved FASTA files

File_R1: >Seq1   File_R2: >Seq1

File_merged: >Seq1.1

>Seq2.1

cmd COMMAND

```
$ fq2fa --merge --filter SRR606249.trim_R1.fastq SRR606249.trim_R2.fastq SRR606249.trim.merged.fasta
```

This step has already been completed for you and the command does NOT need to be executed again.

Introduction to assemblers

Step 6.

Peform assembly using IDBA-UD:

cmd COMMAND

```
$ idba_ud -r SRR606249.trim.merged.fasta -o SRR606249.idbaud_asm --num_threads 45
```

This step has already been completed for you and the command does NOT need to be executed again.

Introduction to assemblers

Step 7.

Perform assembly using MetaSPAdes:

cmd COMMAND

```
$ spades.py -o ./SRR606249.spades_asm --meta -1 SRR606249.trim_R1.fastq -2 SRR606249.trim_R2.fastq --threads 60 --memory 600
```

This step has already been completed for you and the command does NOT need to be executed again.

Introduction to assemblers

Step 8.

Reference assessment: QUAST can perform comparisons against the reference genomes used to construct artifiial metagenome. Start with a baseline size of contiges (>1kb).

cmd COMMAND

```
$ seqmagick convert --min-length 1000 final.contigs.fa megahit_SRR606249.min1000.fasta
```

Introduction to assemblers

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Step 9.

QUAST against 62 reference genomes:

cmd COMMAND

```
$ metaquast.py megahit_SRR606249.min1000.fasta -R ../Shakya_RefGenomes/
```

This step has already been completed for you and the command does NOT need to be executed again.

Prodigal Gene Caller

Step 10.

First step using prodigal:

File: spades_SRR606249.subset.fasta

(Contains a random subset of contigs from metaSPAdes output.)

cmd COMMAND

```
$ prodigal -a temp1.orfs.faa -d temp1.orfs.fna -i spades_SRR606249.subset.fasta -m -
o temp1.txt -p meta -q
```

-a = output, protein translations -d = output, nucleotide putative coding sequences -i = input -m = treats missing sequence (NNNs) as stop -o = output, genbank format -q = quiet output

Prodigal Gene Caller

Step 11.

Check temp1.orfs.faa output:

cmd COMMAND

```
$ less temp1.orfs.faa
$ grep '>' temp1.orfs.faa | wc -l
```

Number of putative proteins.

Prodigal Gene Caller

Step 12.

Visualize the first 10 header lines:

cmd COMMAND

```
$ grep '>' temp1.orfs.faa | head
```

EXPECTED RESULTS

>NODE_2381_length_13704_cov_7.45857_ID_3071845_1 # 2 # 784 # -1 #
ID=1_1;partial=10;start_type=ATG;rbs_motif=AGGA;rbs_spacer=5-10bp;gc_cont=0.520

Prodigal Gene Caller

Step 13.

Use Unix to simplify the header output:

cmd COMMAND

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

```
$ cut -f1 -d " " temp1.orfs.faa > spades_SRR606249.subset.orfs.faa
```

Prodigal Gene Caller

Step 14.

Repeat for nucleotides:

cmd COMMAND

```
$ cut -f1 -d " " temp1.orfs.fna > spades_SRR606249.subset.orfs.fna
```

Prodigal Gene Caller

Step 15.

Determine putative genes for contigs from SPAdes:

Repeat for Megahit and IDBA-UB

cmd COMMAND

```
$ prodigal -a temp1.orfs.faa -i spades_SRR606249.min1000.fasta -m -o temp1.txt -p meta -q
$ grep “>” temp1.orfs.faa | wc -l
$ cut -f1 -d " " temp1.orfs.faa > spades_SRR606249.min1000.orfs.faa
```

Determining Coverage

Step 16.

Determining % of reads and contig coverage using Bowtie2 short read aligner.

Build index file of assembled contigs:

cmd COMMAND

```
$ bowtie2-build spades_SRR606249.min1000.fasta spades_SRR606249.min1000.bt_index
```

This step has already been completed for you and the command does NOT need to be executed again.

Determining Coverage

Step 17.

Perform alignment with trimmed, high-quality reads from SAM file output:

cmd COMMAND

```
$ bowtie2 -q -1 SRR606249.trim_R1.fastq -2 SRR606249.trim_R2.fastq -
x spades_SRR606249.min1000.bt_index --no-unal -S spades_SRR606249.sam -p 35
```

This step has already been completed for you and the command does NOT need to be executed again.

Determining Coverage

Step 18.

Utilize featureCounts to determine reads aligned to a contig. Requires a pseudo-input file based on FASTA input.

cmd COMMAND

```
$ python fastaToSaf.py < spades_SRR606249.min1000.fasta > spades_SRR606249.min1000.saf
```

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

```
$ featureCounts -F SAF -a spades_SRR606249.min1000.saf -
o spades_SRR606249.min1000.readcount spades_SRR606249.sam
```

The "featureCounts" command has already been executed for you. Go ahead and execute the python script "fastaToSaf.py"

Determining Coverage

Step 19.

Custom made Python script - convertReadcountToCoverage.py → can accept multiple readcount inputs to generate a combined coverage matrix:

cmd COMMAND

```
$ grep ">" spades_SRR606249.min1000.fasta | sed 's/>//' > spades_SRR606249.min1000.ids
$ python convertReadcountToCoverage.py spades_SRR606249.min1000.ids spades_SRR606249.min100
0.coverage
```

This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited