Script P2: Contig Assembly

Hannigan GC, Grice EA, et al.

Abstract

This protocol provides a method for assembly of metagenomic data using the Ray assembly toolkit and the subsequent analysis of contig statistics. Based on the publication:

Hannigan, Geoffrey D., et al. "The Human Skin Double-Stranded DNA Virome: Topographical and Temporal Diversity, Genetic Enrichment, and Dynamic Associations with the Host Microbiome." *mBio* 6.5 (2015): e01578-15.

Citation: Hannigan GC, Grice EA, et al. Script P2: Contig Assembly. **protocols.io**
dx.doi.org/10.17504/protocols.io.ed9ba96
Published: 10 Mar 2016

Guidelines

Required Software:

- Ray-2.3.1
- bowtie2-2.1.0

Relevant Files

Output:

- Virome_Sequence_Counts
- Whole_Microbiome_Sequence_Counts

Perl Scripts: calculate_abundance_from_sam.pl
R Scripts: R1 and R2
Python Scripts: get_trimmed_pairs.py

Before start

Perl scripts and other supplementary information available at:

https://figshare.com/articles/The_Human_Skin_dsDNA_Virome_Topographical_and_Temporal_Diversity_Genetic_Enrichment_and_Dynamic_Associations_with_the_Host_Microbiome/1281248

Protocol

Assembly Process

Step 1.

Contigs were assembled using the program Ray. Make directory for input files, separating based on forward and reverse reads.

cmd COMMAND

```
mkdir ./Ray/R1_for_ray
mkdir ./Ray/R2_for_ray
```

Assembly Process

Step 2.

Copy over the pre-processed fastq files to respective directories.

cmd COMMAND

```
cp ./clean_phix_fastq/*R1* ./Ray/R1_for_ray
cp ./clean_phix_fastq/*R2* ./Ray/R2_for_ray
```

Assembly Process

Step 3.

Then we used a custom script from the Bushman lab to get sequence pairs. Basically this means it went through the corresponding R1 and R2 fastq files for each sample and only kept sequences in each file that has a mate.

cmd COMMAND

```
mkdir ./Ray/R1_for_ray_pairs
mkdir ./Ray/R2_for_ray_pairs
```

Make output directories.

Assembly Process

Step 4.

Search through all of the fastq files.

🔗 LINK:

https://figshare.com/articles/The_Human_Skin_dsDNA_Virome_Topographical_and_Temporal_Diversity_Genetic_Enrichment_and_Dynamic_Associations_with_the_Host_Microbiome/1281248

cmd COMMAND

```
for i in $(ls ./Ray/R1_for_ray); do
for j in $(ls ./Ray/R2_for_ray); do
    # Check to see if the sample ID's match
    for x in ${i::9}; do
    for y in ${j::9}; do
        # If fastq files have the same sample ID, trim pairs
        if [ "$x" == "$y" ]; then
            python get_trimmed_pairs.py -f ./Ray/R1_for_ray/$i -s ./Ray/R2_for_ray/$j -
o ./Ray/R1_for_ray_pairs/${i} -t ./Ray/R2_for_ray_pairs/${j}
        fi
    done
    done
done
done
```

NOTES

Geoffrey Hannigan 12 Jan 2016

The python script get_trimmed_pairs.py is used in this step and is available in the supplementary information.

Assembly Process

Step 5.

We first performed assembly for each individual sample. Generate ouput directory.

cmd COMMAND

```
mkdir ./Ray/ray_contigs_from_cat
```

NOTES

Geoffrey Hannigan 12 Jan 2016

It is important to note that the directory for Ray output should not be created before running Ray. This will halt the program and return an error.

Assembly Process

Step 6.

Write function to run Ray.

SOFTWARE PACKAGE (Unix)

Ray Assembly Toolkit, 2.3.1

Jacques Corbeil
https://github.com/sebhtml/Ray-Releases/

cmd COMMAND

```
runRayAcrossSamples () {
    FILE1=${1}_R1.fa
    FILE2=${1}_R2.fa
    echo $FILE1
    echo $FILE2
    mpiexec -n 9 Ray-2.3.1/Ray -minimum-contig-length 500 -
p ./Ray/R1_for_ray_fasta/${FILE1} ./Ray/R2_for_ray_fasta/${FILE2} -
o ./Ray/ray_contigs_from_cat/${1}
}
export -f runRayAcrossSamples
```

Assembly Process

Step 7.

Run function.

cmd COMMAND

```
ls ./Ray/R1_for_ray_fasta | sed 's/\_R1\.fa//g' | xargs -I {} --max-procs=40 sh -
c 'runRayAcrossSamples {}'
```

Assembly Process

Step 8.

Rename output files so they contain the Sample ID. When they come out of the assembler, they are Contigs.fa- we rename them so they are MG100*_Contigs.fa.

cmd COMMAND

```
ls ./Ray/ray_contigs_from_cat | xargs -
I {} mv ./Ray/ray_contigs_from_cat/{}/Contigs.fasta ./Ray/ray_contigs_from_cat/{}/{}_Contig
s.fa
```

Assembly Process

Step 9.

Additionally, each Contigs.fa has names and contigs in order. We want to add the Sample ID to the end of each contig ID to ensure they are all unique.

cmd COMMAND

```
for file in $(ls ./Ray/ray_contigs_from_cat); do
    #Remove block format in contig fasta file | Next three part of same thing | Replace the
 spaces in the contig names with underscores | Add sample ID to the end of each name
    sed -
r 's/\s/_/g' ./Ray/ray_contigs_from_cat/${file}/${file}_Contigs.fa  | sed 's/^\([A,T,G,C,n]
*\)$/\1\@/g' | sed ':a;N;$!ba;s/\@\n\([A,C,G,T,n]\)/\1/g' | sed 's/\@//g' | sed '/\>/s/ /_/
g' | sed "/>/s/$/\_$file/" > ./Ray/ray_contigs_from_cat/${file}/${file}_Contigs_with_format
.fa
done
```

Assembly Process

Step 10.

We also performed assembly for all of the samples concatenated together. Concatenate all fasta files together.

cmd COMMAND

```
cat ./Ray/R1_for_ray_fasta/* > ./Ray/cat_R1_pairs_for_ray.fa
cat ./Ray/R2_for_ray_fasta/* > ./Ray/cat_R2_pairs_for_ray.fa
```

Assembly Process

Step 11.

Run Ray assembler.

cmd COMMAND

```
mpiexec -n 25 Ray-2.3.1/Ray -minimum-contig-length 500 -
p ./Ray/cat_R1_pairs_for_ray.fa ./Ray/cat_R2_pairs_for_ray.fa -
o ./Ray/ray_contigs_from_total_cat_pairs
```

Assembly Process

Step 12.

The contig output files from Ray are in block fasta format so we need to convert this to standard fasta format.

cmd COMMAND

```
sed -
r 's/\s/_/g' ./Ray/ray_contigs_from_total_cat_pairs/Contigs.fasta  | sed 's/^\([A,T,G,C,n]*
\)$/\1\@/g' | sed ':a;N;$!ba;s/\@\n\([A,C,G,T,n]\)/\1/g' | sed 's/\@//g' > ./Ray/ray_contig
s_from_total_cat_pairs/Contigs_no_block.fasta
```

Assembly Process

Step 13.

Rename the contigs.

cmd COMMAND

```
nl -b p\> -w 1 -
s _ ./Ray/ray_contigs_from_total_cat_pairs/Contigs_no_block.fasta | sed 's/\t//' | sed 's/
 //' | sed 's/>.*//' | sed '/[1-9]/s/^/\>/' > ./Ray/ray_contigs_from_total_cat_pairs/Contig
s_no_block_with_names.fasta
```

Calculating Contig Statistics

Step 14.

Once we have assembled contigs, we want to determine the distribution information for these contigs. First, we calculate the total number of contigs and length of the contigs.

cmd COMMAND

```
mkdir ./Ray/ray_contigs_from_total_cat_pairs_contig_stats
```

Calculating Contig Statistics

Step 15.

Generate table with the sequence length of each contig.

cmd COMMAND

```
awk 'NR % 2 {printf $0"\t"} !(NR % 2) {print length($0)}' ./Ray/ray_contigs_from_total_cat_
pairs/Contigs_no_block_with_names.fasta > ./Ray/ray_contigs_from_total_cat_pairs_contig_sta
ts/contig_length.txt
```

Calculating Contig Statistics

Step 16.

Remove extraneous characters from the table.

cmd COMMAND

```
sed 's/>//g' ./Ray/ray_contigs_from_total_cat_pairs_contig_stats/contig_length.txt > ./Ray/
ray_contigs_from_total_cat_pairs_contig_stats/contig_length_without_greater_sign.txt
```

Calculating Contig Statistics

Step 17.

Additionally, we want to map our quality trimmed, decontaminated sequences against our contigs to determine the coverage of our contigs.

cmd COMMAND

```
mkdir ./Ray/ray_contigs_from_total_cat_pairs_contig_coverage_bowtie2
```

Calculating Contig Statistics

Step 18.

Build bowtie reference from the assembled contigs.

SOFTWARE PACKAGE (Unix)

Bowtie 2, 2.1.0

Langmead B, Salzberg S.
https://github.com/BenLangmead/bowtie2

cmd COMMAND

```
bowtie2-build -
f ./Ray/ray_contigs_from_total_cat_pairs/Contigs_no_block_with_names.fasta ./Ray/ray_contig
s_from_total_cat_pairs_contig_coverage_bowtie2/contig_bowtie2_build
```

Calculating Contig Statistics

Step 19.

Align samples to assembled contigs.

cmd COMMAND

```
bowtie2 -
x ./Ray/ray_contigs_from_total_cat_pairs_contig_coverage_bowtie2/contig_bowtie2_build -
f ./Ray/cat_R1_pairs_for_ray.fa -S ./Ray/cat_R1_pairs_for_ray_bowtie2.sam -p 32 -L 25 -N 1
```

Calculating Contig Statistics

Step 20.

Get abundance data from the bowtie output.

LINK:
https://figshare.com/articles/The_Human_Skin_dsDNA_Virome_Topographical_and_Temporal_Diversity_Genetic_Enrichment_and_Dynamic_Associations_with_the_Host_Microbiome/1281248

cmd COMMAND

```
perl calculate_abundance_from_sam.pl ./Ray/cat_R1_pairs_for_ray_bowtie2.sam ./Ray/cat_R1_pa
irs_for_ray_bowtie2_hit_counts.txt
```

NOTES

Geoffrey Hannigan 12 Jan 2016

The perl script calculate_abundance_from_same.pl is used in this step and is available in the the supplementary information.

Calculating Contig Statistics

Step 21.

Merge the contig length and bowtie hit values into a single tab-delimited file.

cmd COMMAND

```
awk 'FNR==NR { a[$1]=$2; next } $1 in a { print $1"\t"$2"\t"a[$1] }' ./Ray/ray_contigs_from
_total_cat_pairs_contig_stats/contig_length_without_greater_sign_with_header.txt ./Ray/cat_
R1_pairs_for_ray_bowtie2_hit_counts.txt > ./Ray/ray_contigs_from_total_cat_pairs_contig_sta
ts/contig_length_with_coverage_for_graphing.tsv
```

Calculating Contig Statistics

Step 22.

Subsequent analysis of these results were performed in R.