

The draft genome sequence of a desert tree *Populus pruinosa*

Wenlu Yang

Abstract

Citation: Wenlu Yang The draft genome sequence of a desert tree *Populus pruinosa*. **protocols.io**
dx.doi.org/10.17504/protocols.io.ii5ccg6

Published: 22 Jun 2017

Protocol

Reads Filter

Step 1.

Filter the input raw sequences by using the 01.QualityControlAndMergeReads.pl script we wrote

Reads Filter

Step 2.

Filter the input raw sequences by using the Lighter

```
cmd COMMAND (Lighter - v1.0.7)  
lighter -r *.1.filter.fq.gz -r *.2.filter.fq.gz -k 17 5900000000 alpha -t 10 -trim
```

Reads Filter

Step 3.



Filter the input raw sequences by using the Fastuniq

```
cmd COMMAND (FastUniq - v1.1)  
fastuniq -i reads_list.txt -t q -o *.1.filter.fq -p *.2.filter.fq
```

k-mer analysis

Step 4.

Estimate the genome size with k-mer analysis.

 **SOFTWARE PACKAGE (SOAPec)**
SOAP: short oligonucleotide alignment program, v2.2 
<http://soap.genomics.org.cn/about.html>

Genome Assemble

Step 5.

Run Platanus (version 1.2.1) to assemble our genome.

 **SOFTWARE PACKAGE (linux)**

Platanus Assembler (PLATform for Assembling NUCleotide Sequences, v1.2.1 [↗](#))

http://platanus.bio.titech.ac.jp/?page_id=5

cmd [COMMAND \(Platanus - v1.2.1\)](#)

```
platanus assemble -f xx.1.fq xx.2.fq ... -t 50 -m 290 -o Ppr
platanus scaffold -o Ppr -c Ppr_contig.fa -b Ppr_contigBubble.fa -t 64 -
IP1 158bp.1.fq 158bp.2.fq -IP2 483bp.1.fq 483bp.2.fq -IP3 780bp.1.fq 780bp.2.fq -
OP4 2k.1.fq 2k.2.fq -OP5 5k.1.fq 5k.2.fq -OP6 ...
platanus gap_close -c Ppr_scaffold.fa -o platanus_gapclose -t 64 -
IP1 158bp.1.fq 158bp.2.fq -IP2 483bp.1.fq 483bp.2.fq -IP3 780bp.1.fq 780bp.2.fq -
OP4 2k.1.fq 2k.2.fq -OP5 5k.1.fq 5k.2.fq -OP6 ...
```

Genome Assemble

Step 6.

Perform Gapcloser (version 1.12) to further close gaps in our genome obtained in step3.

 [SOFTWARE PACKAGE \(linux\)](#)

GapCloser, v1.02 [↗](#)

BGI

<http://soap.genomics.org.cn/soapdenovo.html>

cmd [COMMAND \(GapCloser - v1.02\)](#)

```
GapCloser -a scaffold.fa -b soapGapClose.config -o Ppr_GapCloser.fa -t 64 -l 100
```

Genome Assembly Assessment

Step 7.

We first examined the coverage of highly conserved genes using BUSCO to evaluate the completeness of our assembly

 [SOFTWARE PACKAGE \(linux\)](#)

BUSCO, v1.22 [↗](#)

<http://busco.ezlab.org/v2/>

Genome Assembly Assessment

Step 8.

the unigenes obtained in our study and the protein-coding genes predicted in the *P. euphratica* and *P. trichocarpa* genomes were aligned to our genome assembly using the BLAT algorithm with default parameters

Genome Assembly Assessment

Step 9.

we applied the FRC v1.3.0 (Feature-Response Curves) method to evaluate the trade-off between the contiguity and correctness of our assembly.

 [SOFTWARE PACKAGE \(linux\)](#)

FRC (Feature-Response Curves), v1.3.0 [↗](#)

https://github.com/vezzi/FRC_align

cmd [COMMAND \(FRCurve - v1.3.0\)](#)

```
FRC --genome-size 590000000 --pe-sam 483bp.bam --mp-sam 2000bp.bam --pe-max-insert 500 --
mp-max-insert 2000 --out frcurve
```

Call SNP

Step 10.

We mapped the clean reads from the paired-end libraries to the *P. pruinosa* genome using the

Burrows-Wheeler Aligner (BWA) and performed variant calling using the Genome Analysis Toolkit(GATK).

SOFTWARE PACKAGE (linux)

BWA(Burrows-Wheeler Aligner), v0.7.12

<https://sourceforge.net/projects/bio-bwa/files/>

cmd **COMMAND (GATK - v3.5)**

```
java -Xmx10g -jar GenomeAnalysisTK.jar -R Ppr_genome.fa -T RealignerTargetCreator -  
o <sample id>.intervals -I <sample id>.rmdup.bam  
java -Xmx10g -jar GenomeAnalysisTK.jar -R Ppr_genome.fa -T IndelRealigner -  
targetIntervals <sample id>.intervals -o <sample id>.realn.bam -I <sample id>.rmdup.bam
```

Transcriptome Assemble

Step 11.

we assembled these RNA-seq reads using Trinity

SOFTWARE PACKAGE (linux)

Trinity, v2.1.1

<https://github.com/trinityrnaseq/trinityrnaseq/releases>

cmd **COMMAND (Trinity - v2.1.1)**

```
Trinity --seqType fq --max_memory 200G --left reads.1.fq.gz --right reads.2.filter.fq.gz --  
SS_lib_type RF --CPU 30 --no_cleanup --trimmomatic --output <out dir>
```

Transcriptome Assemble

Step 12.

we reduced the redundancy of transcript sequences (>95% similarity) using CD-Hit v4.6.1 next spet 11.

SOFTWARE PACKAGE (linux)

CD-HIT, v4.6.1

<http://weizhongli-lab.org/cd-hit/>

cmd **COMMAND (CD-Hit - v4.6.1)**

```
cd-hit-est -i Trinity_out.fasta -o cd-hit_out.fa -c 0.95 -T 30 -M 0 > cd-hit.log
```

Transcriptome Assemble

Step 13.

we use the software TransDecoder v2.1.0 to identify candidate coding regions within the transcript sequences we filtered at step 12.

SOFTWARE PACKAGE (linux)

TransDecoder, v2.1.0

<https://github.com/TransDecoder/TransDecoder/releases>

cmd **COMMAND (TransDecoder - v4.6.1)**

```
TransDecoder.LongOrfs -t cd-hit_out.fa;  
TransDecoder.Predict -t cd-hit_out.fa
```

Repeat Annotation_homolog

Step 14.

Run RepeatMasker and RepeatProteinMask (version 4.0.6) to identify repeats in the genome at DNA and protein level, respectively, by aligning sequences against existing databases, Repbase TE library and TE protein database.

cmd [COMMAND \(RepeatProteinMask - v4.0.6\)](#)

```
RepeatProteinMask -engine abblast -noLowSimple -pvalue 0.0001 Ppr_genome.fa
```

Tandem Repeats Annotation

Step 15.

we annotated tandem repeats using the software Tandem Repeat Finder (TRF v4.07b)

cmd [COMMAND \(TRF - v4.07b\)](#)

```
trf Ppr_genome.fa 2 7 7 80 10 50 2000 -d -h
```

Repeat Annotation_denovo

Step 16.

Run RepeatModeler, and RepeatScout, respectively, to build de novo library based on the input assembled genome sequence.

cmd [COMMAND \(RepeatMasker - v4.0.6\)](#)

```
RepeatMasker -lib RM*/consensi.fa.classified -pa 30 Ppr_genome.fa
```

Gene prediction_homolog

Step 17.

Download protein sequences of homolog species (*P. euphratica*, *P. trichocarpa*, *Ricinus communis*, *Arabidopsis thaliana*, *Carica papaya* and *Eucalyptus grandis*), then align these against our masked genome sequences with BLAST, and then using BLAST2GENE to obtain the position informations of the BLAST mapping results, run GeneWise (version 2.4.1) to predict genes.

cmd [COMMAND \(GeneWise - v2.4.1\)](#)

```
genewise -u <start> -v <end> -<trev|tfor> -gff query.fa <chr name>.fa > genewise.gff
```

Gene prediction_denovo

Step 18.

Run Augustus (version 3.2.1) and GenScan to de novo predict genes in the repeat-masked genome sequences.

 [SOFTWARE PACKAGE \(linux\)](#)

Augustus, v3.2.1 

<http://bioinf.uni-greifswald.de/augustus/downloads/>

cmd [COMMAND \(GenScan\)](#)

```
genscan genscan/lib/Arabidopsis.smat <chr name>.fa > <chr name>.out
```

Gene prediction_transcripts

Step 19.

For transcriptome-based approach, the *P. pruinosa* transcripts obtained above were aligned to the *P. pruinosa* genome and further assembled using the Program to Assemble Spliced Alignments (PASA v2.0.2) to detect likely protein coding regions.

Gene prediction_EVM

Step 20.

we integrate genes predicted in step 17-19 to obtain the consensus gene set by using EVM v1.1.1.

SOFTWARE PACKAGE (linux)

EVM, v1.1.1

<http://evidencemodeler.github.io/>

cmd **COMMAND (EVM - V1.1.1)**

```
evidence_modeler.pl --genome <chr name>.fa --weights weights.txt --  
gene_predictions ab_initio.gff --protein_alignments homolog.gff --  
transcript_alignments pasa.gff > evm.out; EVM_to_GFF3.pl evm.out <chr name> > evm.out.gff
```

Functional annotation

Step 21.

Map protein sequences of the final gene set to existing databases to identify their functions or motifs, such as SwissProt, TrEMBL, KEGG, InterPro.

Gene Expression

Step 22.

we align the transcriptome reads we obtained from three tissues to the genome assembly using tophat v2.1.1 and then compute the expression of each gene we predicted using cufflinks v2.2.1

SOFTWARE PACKAGE (linux)

Cufflinks, v2.2.1

<http://cole-trapnell-lab.github.io/cufflinks/install/>

Synteny Analysis

Step 23.

Determine the blocks syntenic between *P. pruinosa* and *P. euphratica* by the software MCScanX.

SOFTWARE PACKAGE (linux)

MCScanX

<http://chibba.pgml.uga.edu/mcscan2/>

Synteny Analysis

Step 24.

process the whole-genome alignment using the program 'LAST'.

cmd **COMMAND (last - v802)**

```
lastdb -uNEAR -cR11 ref_db ref.fa  
lastal -P48 -m100 -E 0.05 ref_db Ppr_genome.fa | last-split > query2db.maf  
maf-swap query2db.maf | last-split > query2db.maf.sing.maf
```

Gene Family Clustering Analysis

Step 25.

Cluster the gene family using OrthoMCL v2.0.9 on all the protein-coding genes of *P. pruinosa* and 10 additional species (*P. euphratica*, *P. trichocarpa*, *Salix suchowensis*, *Ricinus communis*, *Arabidopsis thaliana*, *Carica papaya*, *Fragaria vesca*, *Cucumis sativus*, *Eucalyptus Grandis* and *Vitis vinifera*).

SOFTWARE PACKAGE (linux)

orthoMCL, v2.0.9

<http://orthomcl.org/common/downloads/software/>

cmd **COMMAND (orthoMCL - v2.0.9)**

```
orthomclInstallSchema orthomcl.config.template  
orthomclAdjustFasta compliantFasta/species species.pep 1
```

```

orthomclFilterFasta compliantFasta/ 10 20
makeblastdb -in goodProteins.fasta -dbtype prot
blastp -db goodProteins.fasta -query goodProteins.fasta -out all-all.blastp.out -
evalue 1e-5 -outfmt 6 -num_threads 24
orthomclBlastParser all-all.blastp.out compliantFasta > similarSequences.txt
orthomclLoadBlast orthomcl.config.template similarSequences.txt
orthomclPairs orthomcl.config.template orthomcl_pairs.log cleanup=no
orthomclDumpPairsFiles orthomcl.config.template
mcl mclInput --abc -I 1.5 -o mclOutput
orthomclMclToGroups cluster 1 < mclOutput > groups.txt

```

Phylogeny Analysis

Step 26.

Align the single-copy gene families identified in the orthomcl result with MUSCLE v3.8.31 and low quality regions of the alignments were identified and trimmed with Gblocks v0.91b using default parameters. Finally, concatenate these Gblocks results into one 'super gene' for each species to construct a phylogenetic tree using RAXML v8.2.8.

```

cmd COMMAND (RaxML - v8.2.8)
clustalw2 -INFILE=align.part.fa -CONVERT -OUTFILE=alian.part.phy -OUTPUT=PHYLIP
raxmlHPC-PTHREADS-AVX -T 30 -f a -x 12345 -p 12345 -# 100 -m GTRGAMMA -s align.part.phy -
n tre

```

Divergence Time Analysis

Step 27.

Estimate the divergence time based on the phylogenetic relationships by the software of MCMCTree, using fossil calibration times for divergence between *A. thaliana* and *C. papaya* (54-90 million years ago, Mya), *A. thaliana* and *R. communis* (95-109 Mya), *V. vinifera* and *A. thaliana* (106-119 Mya), which were obtained from the TimeTree database (<http://www.timetree.org/>).

SOFTWARE PACKAGE (linux)

PAML, v4.9

<http://abacus.gene.ucl.ac.uk/software/paml.html>

Gene Family Analysis

Step 28.

examine gene family evolution across entire genomes using the CAFÉ (Computational Analysis of gene Family Evolution, v3.1).

SOFTWARE PACKAGE (linux)

CAFÉ (Computational Analysis of gene Family Evolution), v3.1

https://sourceforge.net/projects/cafehahnlab/?source=typ_redirect