Polistes dominula genome project

Daniel Standage Volker Brendel Amy Toth

November 13, 2014

1 Overview

This documentation is a record of our work for the *Polistes dominula* genome project. It was created to 1) serve as full disclosure of all of the methods, commands, and software used to produce the reported results, and 2) facilitate anonymous replication of those results.

1.1 Data access

Raw instrument data and final data outputs are stored in the iPlant Data Store under the path `/iplant/home/standage/Polistes_dominula/`. All file and directory paths provided in this documentation are relative to that root path, which for the remainder of the documentation will be designated the **Pdom Data Store**.

1.2 Using this documentation

This project is divided into several sections, with each section focusing on a single analysis or small group of related analyses. Each section has a dedicated directory containing code and documentation specific to that section. These resources can be browsed or downloaded at GitHub.

- A `README.md` file (in Markdown format) is included for each section, which provides a prose description of what each set of commands is doing. This file is intended to facilitate interactive replication of results: typing or pasting the commands into the terminal and executing them manually to produce the output. (Note: a single PDF document containing all documentation was produced by concatenating all of the various README files into a single Markdown file and converting to PDF format.)
- Each section also contains a `Makefile` file which includes the same commands as the README file, though without the commentary and in slightly different syntax. The purpose of these files is to facilitate automated replication of each analysis in batch mode. To execute this procedure for a particular analysis, simply change to that directory and execute `make` on the command line.
- Most sections also include additional supplementary files, such as source code, graphics, or configuration files necessary for replicating the results. The purpose of each supplemental file should be clear from the documentation.

If you encounter any problems using this documentation or its associated files, please open a ticket with the Pdom Genome Project issue tracker.

1.3 Authors

- Daniel Standage; Indiana University
- Volker Brendel; Indiana University
- Amy Toth, principal investigator; Iowa State University

2 Genome size estimation

Jellyfish version 2.1.3 was used to count k-mer distributions in the raw genomic short read data. The k-mer coverage C_k was determined for several values of k: 17, 21, 25, and 29. A linear model of C_k as a function of k was fit to compute the estimated nucleotide coverage $C = C_1$ and genome size. The k-mer histogram files have been deposited in the Pdom Data Store at `r1.2/genome-size-est/`.

2.1 Procedure (interactive)

First, designate the number of available processors. This will run multiple jobs/threads at once to speed up computations. For a laptop or a desktop, this will usually be 4, 8, or 16. For server or HPC hardware, you mave have as many as 32 to 64 processors at your disposal.

```
NumThreads=16
```

Next, download short reads using iRODS and decompress.

```
iget -Vr /iplant/home/standage/Polistes_dominula/sequence/genome
ls genome/*.gz | parallel --gnu --jobs $NumThreads gunzip
```

Then, count k-mers and produce k-mer frequency histograms.

```
FastqFiles=$(ls genome/*.fq)
for k in  17 21 25 29
do
  jellyfish count -m $k -s 100M -t $NumThreads -C -o pdom-${k}mers.jf $FastqFiles
  jellyfish histo pdom-${k}mers.jf > pdom-${k}mers.hist
done
```

Finally, estimate k-mer coverage, genome coverage, and genome size.

```
./size-coverage-estimate.R
```

Clean up huge data files.

```
rm -r genome/*.fq *.jf
```

2.2 Procedure (automated)

The same procedure can also be run in batch mode using the following commands (in the `genome-size` directory).

```
make NumThreads=16
make clean
```

2.3 References

- **Marçais G, Kingsford C** (2011) A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* **27**:764-70, doi:10.1093/bioinformatics.

3 Genome assembly

Raw DNA-Seq reads were groomed using Trimmomatic version 0.22, and the groomed reads were then assembled using AllPaths-LG version 43216. The final assembly file has been deposited in the Pdom Data Store at `r1.2/genome-assembly/pdom-scaffolds-unmasked-r1.2.fa.gz`.

3.1 Procedure (interactive)

3.1.1 Short read quality control

First, designate the number of available processors to speed up Trimmomatic's computations. Also, provide the path of the `trimmomatic-0.22.jar` file contained in the Trimmomatic source code distribution.

```
NumThreads=16
TrimJar=/usr/local/src/Trimmomatic-0.22/trimmomatic-0.22.jar
PdomData=/iplant/home/standage/Polistes_dominula
```

Now for the processing. We apply the following filters to each read pair.

- remove adapter contamination
- remove any nucleotides at either end of the read whose quality score is below 3
- trim the read once the average quality in a 5bp sliding window falls below 20
- discard any reads which, after processing, fall below the length threshold (40bp for 100bp reads, 26bp for 35bp reads)

```
for sample in 200bp 500bp 1kb 3kb 8kb
do
  iget ${PdomData}/sequence/genome/pdom-gdnaseq-${sample}-1.fq.gz
  iget ${PdomData}/sequence/genome/pdom-gdnaseq-${sample}-2.fq.gz
  ./run-trim.sh $sample $TrimJar $NumThreads
done
```

3.1.2 Assembly with AllPaths-LG

First, prepare a working directory for the assembly.

```
mkdir -p assembly/Polistes_dominula/data-trim
mv pdom-gdnaseq-*-trim-?.fq assembly/Polistes_dominula/data-trim/.
```

Next, convert the input files into the internal format required by AllPaths-LG.

```
PrepareAllPathsInputs.pl \
      DATA_DIR=assembly/Polistes_dominula/data-trim \
      PLOIDY=2
```

Then, execute the assembly procedure.

```
RunAllPathsLG PRE=assembly \
              REFERENCE_NAME=Polistes_dominula \
              DATA_SUBDIR=data-trim \
              RUN=run01 \
              TARGETS=standard
```

Finally, assign official project IDs to the scaffolds, compress, and clean up intermediate data files.

```
./scaff-ids.pl PdomSCFr1.2- \
    < $PRE/Polistes_dominula/data-trim/run01/ASSEMBLIES/test/final.assembly.fasta \
    > pdom-scaffolds-unmasked-r1.2.fa
gzip pdom-scaffolds-unmasked-r1.2.fa
rm -rf assembly pdom-gdnaseq*.fq*
```

3.2 Procedure (automated)

The same procedure can also be run in batch mode using the following commands (in the `genome-assembly` directory).

```
make NumThreads=16 \
     TrimJar=/usr/local/src/Trimmomatic-0.22/trimmomatic-0.22.jar
make clean
```

3.3 References

- **Lohse M, Bolger AM, Nagel A, Fernie AR, Lunn JE, Stitt M, Usadel B** (2012) RobiNA: a user-friendly, integrated software solution for RNA-Seq-based transcriptomics. *Nucleic Acids Research*, **40**:W622-7, doi:10.1093/nar/gks540.
- **Gnerre S, MacCallum I, Przybylski D, Ribeiro F, Burton J, Walker B, Sharpe T, Hall G, Shea T, Sykes S, Berlin A, Aird D, Costello M, Daza R, Williams L, Nicol R, Gnirke A, Nusbaum C, Lander ES, Jaffe DB** (2010) High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences USA*, **108**(4):1513-1518, doi:10.1073/pnas.1017351108.

4 Transcript assembly

Raw DNA-Seq reads were groomed using Trimmomatic version 0.22, and the groomed reads were then assembled using Trinity version r20131110. Then the assembled transcripts were processed with mRNAmarkup version 10-3-2013 to remove contaminants and correct erroneously assembled chimeric transcripts. The cleaned and annotated transcripts have been deposited in the Pdom Data Store at `r1.2/transcript-assembly/`.

4.1 Procedure (interactive)

4.1.1 Short read quality control

First, designate the number of available processors to speed up Trimmomatic's computations. Also, provide the path of the `trimmomatic-0.22.jar` file contained in the Trimmomatic source code distribution.

```
NumThreads=16
TrimJar=/usr/local/src/Trimmomatic-0.22/trimmomatic-0.22.jar
PdomData=/iplant/home/standage/Polistes_dominula
```

Now for the processing. We apply the following filters to each read pair.

- remove adapter contamination

- remove any nucleotides at either end of the read whose quality score is below 3
- trim the read once the average quality in a 5bp sliding window falls below 20
- discard any reads which, after processing, fall below the 40bp length threshold

```
for caste in q w
do
  for rep in {1..6}
  do
    sample=${caste}${rep}
    iget ${PdomData}/sequence/transcriptome/pdom-rnaseq-${sample}-1.fq.gz
    iget ${PdomData}/sequence/transcriptome/pdom-rnaseq-${sample}-2.fq.gz
    ./run-trim.sh $sample $TrimJar $NumThreads
  done
done
```

4.1.2 Assembly with Trinity

Trinity requires a single input file—or a pair of input files for paired-end data. We need to combine all of the data into a single pair of files.

```
cat pdom-rnaseq-*-trim-1.fq > pdom-rnaseq-all-trim-1.fq
cat pdom-rnaseq-*-trim-2.fq > pdom-rnaseq-all-trim-2.fq
```

We'll then execute the Trinity assember using the `--CuffFly` reconstruction algorithm.

```
Trinity.pl --seqType fq \
           --JM 100G \
           --bflyHeapSpaceMax 50G \
           --output pdom-trinity \
           --CPU $NumThreads \
           --left  pdom-rnaseq-all-trim-1.fq \
           --right pdom-rnaseq-all-trim-2.fq \
           --full_cleanup \
           --jaccard_clip \
           --CuffFly
```

4.1.3 Post-processing with mRNAmarkup

Contaminant, reference protein, and miRNA databases were collected as described in the mRNAmarkup documentation (`db/0README` and `db/0README-hy`). The mRNAmarkup procedure was then run on the Trinity output. Be sure to edit the `mRNAmarkup.conf` file with the correct paths to the databases.

```
mRNAmarkup -c mRNAmarkup.conf \
           -i pdom-trinity/Trinity.fasta \
           -o output-mRNAmarkup
```

4.1.4 Potential *Polistes*-specific genes

Polistes metricus and *P. canadensis* transcript assemblies were groomed and annotated using the same mRNAmarkup procedure. All three transcript sets had a substantial number of TSAs that could not be

annotated by mRNAmarkup. The longest open reading frame translations of at least 80 amino acids derived from these unmatched TSAs were pairwise compared with BLASTp to detect any *Polistes*-conserved and species-specific genes. This procedure relies on a workflow and several scripts available in the supplemental/ directory of the documentation repository.

The workflow to find the *Polistes*-conserved transcripts with no external protein matches can be executed with this command. The workflow depends on the `dnatopro` program (part of the VBTools utilities included in the mRNAmarkup distribution) and the NCBI BLAST+ suite.

```
./venn.make BLASTTHREADS=$NumThreads all
```

Transcripts associated with potential *Polistes*-specific genes will be placed in the `pdom-tsa-r1.2-unmatched-pep.fa` file.

4.2 Procedure (automated)

The same procedure can also be run in batch mode using the following commands (in the `transcript-assembly` directory). Note that this procedure does not automate the mRNAmarkup analysis. After producing the Trinity assembly, it retrieves previously computed mRNAmarkup results for the clade-specific gene analysis.

```
make NumThreads=16 \
     TrimJar=/usr/local/src/Trimmomatic-0.22/trimmomatic-0.22.jar
make clean
```

4.3 References

- **Lohse M, Bolger AM, Nagel A, Fernie AR, Lunn JE, Stitt M, Usadel B** (2012) RobiNA: a user-friendly, integrated software solution for RNA-Seq-based transcriptomics. *Nucleic Acids Research*, **40**:W622-7, doi:10.1093/nar/gks540.
- **Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, Regev A** (2011) Full-length transcriptome assembly from RNA-seq data without a reference genome. *Nature Biotechnology*, **29**(7):644-52, doi:10.1038/nbt.1883.

5 Repeat masking

The genome assembly was screened for known repetitive elements using RepeatMasker version open-4.0.5 and Repbase version 20140131. After masking repeats identified by RepeatMasker, the assembly was screened for additional repeats using Tallymer version 1.5.2. To discriminate *bona fide* repetitive elements from genes occurring in high copy number in the genome, all repeats identified by Tallymer were subjected to a BLASTX search against a database of Hexapod proteins. Any repeats with matches in the database and e-values $<$ 1e-5 were discarded as probable high copy number genes, while the rest were used to mask the genome. The final masked sequence has been deposited in the Pdom data store at `r1.2/genome-assembly/pdom-scaffolds-masked-r1.2.fa.gz`.

5.1 Procedure (interactive)

First, download the unmasked genome sequence.

```
PdomData=/iplant/home/standage/Polistes_dominula
iget ${PdomData}/r1.2/genome-assembly/pdom-scaffolds-unmasked-r1.2.fa.gz
gunzip pdom-scaffolds-unmasked-r1.2.fa.gz
```

5.1.1 Screening with RepeatMasker

Next, identify known repeats with RepeatMasker. By default, RepeatMasker produces soft-masked (lower-case) sequences, so we need to post-process the output to hard mask (N) the sequence.

```
NumThreads=16
GCContent=30.77
RepeatMasker -species insects -parallel $NumThreads -gc $GCContent \
             -frag 4000000 -lcambig -xsmall -gff \
             pdom-scaffolds-unmasked-r1.2.fa \
             > rm.log 2>&1
python lc2n.py < pdom-scaffolds-unmasked-r1.2.fa.masked \
               > pdom-rm-masked.fa
```

5.1.2 Screening with Tallymer

The, do additional *k*-mer based screening for repetitive elements using Tallymer (procedure published by Dan Bolser).

```
gt suffixerator -v \
                -db $IDX \
                -indexname $IDX \
                -tis -suf -lcp -des -ssp -sds -dna \
                > suffixerator.log 2>&1

gt tallymer occratio -v \
                     -minmersize 10 \
                     -maxmersize 45 \
                     -output unique nonunique nonuniquemulti total relative \
                     -esa $IDX \
                     > pdom.occratio.10.45.dump

gt tallymer mkindex -v \
            -mersize 19 \
            -minocc 50 \
            -esa $IDX \
            -counts -pl \
            -indexname pdom.idx.19.50 \
            > mkindex.log 2>&1

gt tallymer search -v \
            -output qseqnum qpos counts \
            -tyr pdom.idx.19.50 \
            -q $IDX \
            > pdom.repeats.19.50.tmer \
            2> tallymer.search.log
```

```
tallymer2gff3.plx -k 19 -s $IDX \
                  pdom.repeats.19.50.tmer \
                  > pdom.repeats.19.50.gff3

gff2fasta.plx -s pdom-rm-masked.fa \
              -f pdom.repeats.19.50.gff3 \
              > pdom.repeats.19.50.fa
```

Do a BLASTx search of repeats found by Tallymer vs known hexapod proteins, and parse out those with hits using MuSeqBox.

```
curl 'http://www.uniprot.org/uniprot/?query=taxonomy%3a6960&force=yes&format=fasta' \
     > hexapoda.fa
makeblastdb -in hexapoda.fa -dbtype prot -parse_seqids
blastx -query pdom.repeats.19.50.fa -db hexapoda.fa \
       -num_alignments 10 -evalue 1e-5 -num_threads 64 \
       -out pdom.repeats.19.50.blastx \
       > pdom.repeats.19.50.log 2>&1

MuSeqBox -i pdom.repeats.19.50.blastx -L 100 \
    | cut -f 1 -d ' ' | sort | uniq \
    | perl -ne 'm/(PdomSCFr1.2-\d+)-\d+\/(\d+)-(\d+)/ and print "$1\t$2\t$3\n"' \
    > pdom.repeats.19.50.hexapodhits.txt
```

Finally, hard mask the Tallymer repeats, excluding any that match Hexapod proteins as probably high-copy-number genes.

```
mask.pl pdom.repeats.19.50.gff3 \
        pdom.repeats.19.50.hexapodhits.txt \
        pdom-rm-masked.fa \
        > pdom-scaffolds-masked-r1.2.fa
gzip pdom-scaffolds-masked-r1.2.fa
```

5.2 Procedure (automated)

The same procedure can also be run in batch mode using the following commands (in the `repeat-masking` directory).

```
make NumThreads=16 GCContent=30.77
make clean
```

5.3 References

- **Smit AFA, Hubley R, Green P**. RepeatMasker Open-3.0. 1996-2010 http://www.repeatmasker.org.
- **Jurka J, Kapitonov VV, Pavlicek A, Klonowski P, Kohany O, Walichiewicz J** (2005) Repbase Update, a database of eukaryotic repetitive elements. *Cytogentic and Genome Research*, **110**(1-4):462-467, doi:10.1159/000084979.
- **Kurtz S, Narechania A, Stein JC, Ware D** (2009) A new method to compute K-mer frequencies and its application to annotate large repetitive plant genomes. *BMC Genomics*, **9**:517, doi:10.1186/1471-2164-9-517.

6 Transcript alignment

Assembled and groomed *P. dominula* transcripts were spliced aligned at high stringency to the repeat-masked genome sequence using GeneSeqer version 2-26-2014. *Polistes canadensis* and *P. metricus* transcripts were also aligned with GeneSeqer using less stringent parameters. The alignments have been deposited in the Pdom Data Store at `r1.2/transcript-alignment`.

6.1 Procedure (interactive)

First, set the `GSQDir` variable to the path of the GeneSeqer source code.

```
GSQDir=/usr/local/src/GENESEQER
```

Next, align *P. dominula* TSAs with stringent parameters.

```
# Download the assembled and masked genome sequence
PdomData=/iplant/home/standage/Polistes_dominula
iget ${PdomData}/r1.2/genome-assembly/pdom-scaffolds-masked-r1.2.fa.gz
gunzip pdom-scaffolds-masked-r1.2.fa.gz

# P. dominula alignments
iget ${PdomData}/r1.2/transcript-assembly/pdom-tsa-r1.2.fa.gz
gunzip pdom-tsa-r1.2.fa.gz
MakeArray pdom-tsa-r1.2.fa
GeneSeqerL -s Arabidopsis \
           -L pdom-scaffolds-masked-r1.2.fa \
           -D pdom-tsa-r1.2.fa.gz \
           -O pdom-tsa-r1.2-masked.gsq \
           -p $GSQDIR/data/prmfileHQ \
           -x 30 -y 45 -z 60 -w 0.8 -m 1000000 \
           > pdom-tsa-r1.2-masked-gsq.log 2>&1
```

Now the *P. canadensis* and *P, metricus* TSAs with less stringent parameters.

```
# P. metricus alignments
iget ${PdomData}/r1.2/transcript-assembly/pmet-tsa-r1.2.fa.gz
gunzip pmet-tsa-r1.2.fa.gz
MakeArray pmet-tsa-r1.2.fa
GeneSeqerL -s Arabidopsis \
           -L pdom-scaffolds-masked-r1.2.fa \
           -d pmet-tsa-r1.2.fa \
           -O pmet-tsa-r1.2-masked.gsq \
           -p $GSQDIR/data/prmfile \
           -x 16 -y 24 -z 48 -w 0.8 -m 1000000 \
           > pmet-tsa-r1.2-masked-gsq.log 2>&1

# P. canadensis alignments
iget ${PdomData}/r1.2/transcript-assembly/pcan-tsa.fa.gz
gunzip pcan-tsa.fa.gz
MakeArray pcan-tsa.fa
GeneSeqerL -s Arabidopsis \
           -L pdom-scaffolds-masked-r1.2.fa \
```

```
            -d pcan-tsa.fa \
            -O pcan-tsa-masked.gsq \
            -p $GSQDIR/data/prmfile \
            -x 16 -y 24 -z 48 -w 0.8 -m 1000000 \
            > pcan-tsa-masked-gsq.log 2>&1
```

6.2 Procedure (automated)

The same procedure can also be run in batch mode using the following commands (in the `transcript-alignment` directory).

```
make GSQDir=/usr/local/src/GENESEQER
```