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
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.

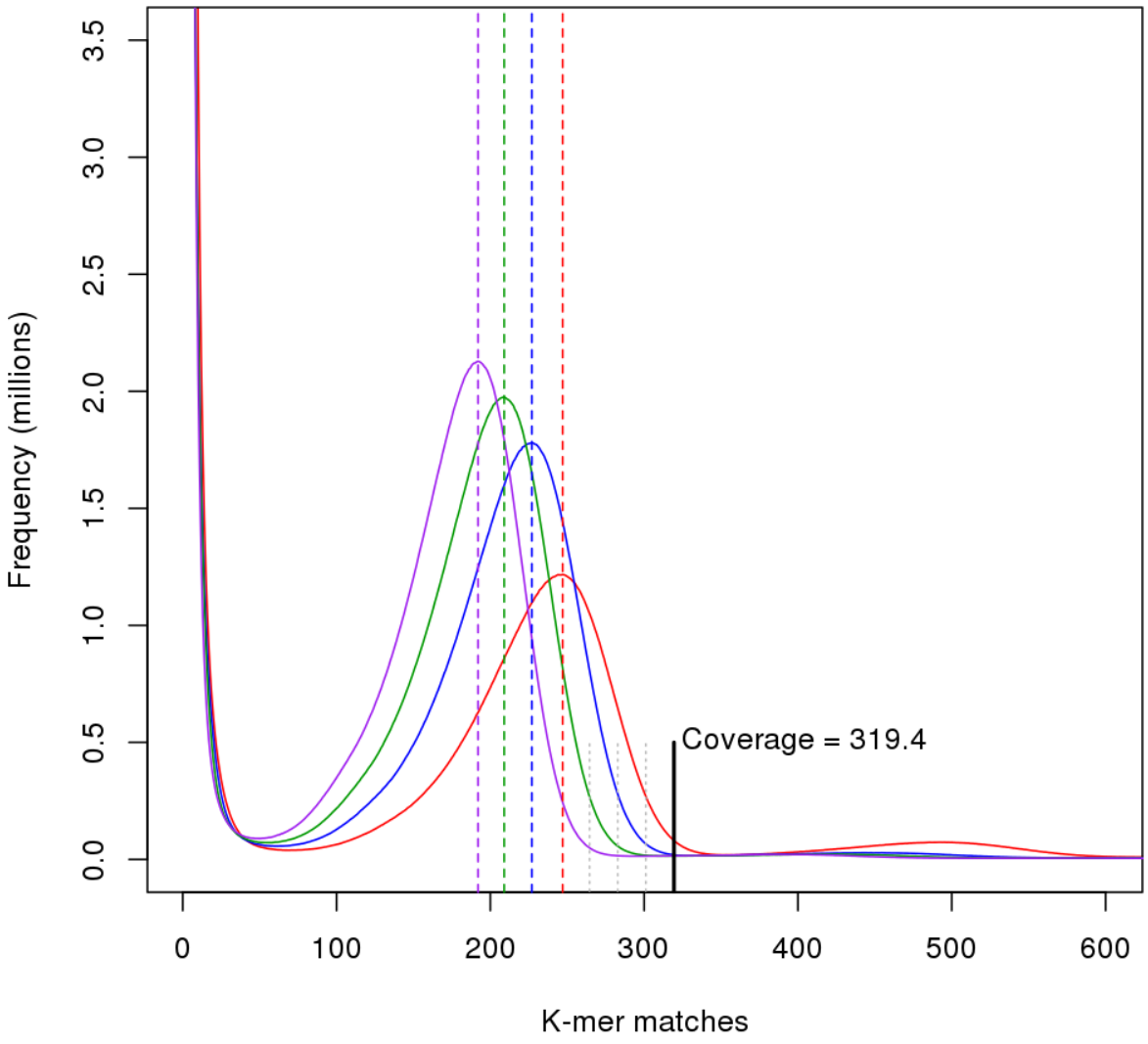

Figure 1: We can estimate genome size by observing coverage C_k for different values of k and interpolating to find C_1.

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
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.