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.

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*
```

3.1.3 Comparison of genome composition

Compare the nucleotide composition of *Polistes dominula* with that of other Hymenoptera. Only consider scaffolds/chromosomes/linkage groups 1 Mbp in length or greater.

```
# Import data
am <- read.table("amel.seq.dstbn.csv", sep=",", header=FALSE)
bt <- read.table("bter.seq.dstbn.csv", sep=",", header=FALSE)
hs <- read.table("hsal.seq.dstbn.csv", sep=",", header=FALSE)
mr <- read.table("mrot.seq.dstbn.csv", sep=",", header=FALSE)
nv <- read.table("nvit.seq.dstbn.csv", sep=",", header=FALSE)
pd <- read.table("pdom.seq.dstbn.csv", sep=",", header=FALSE)

# Calculate GC content
am$GCcontent <- 100 * am$V3 / am$V2
bt$GCcontent <- 100 * bt$V3 / bt$V2
hs$GCcontent <- 100 * hs$V3 / hs$V2
mr$GCcontent <- 100 * mr$V3 / mr$V2
nv$GCcontent <- 100 * nv$V3 / nv$V2
pd$GCcontent <- 100 * pd$V3 / pd$V2

# Filter out sequences shorter than 1 Mbp
amel <- am[am$V4 >= 1000000,]
bter <- bt[bt$V4 >= 1000000,]
hsal <- hs[hs$V4 >= 1000000,]
mrot <- mr[mr$V4 >= 1000000,]
nvit <- nv[nv$V4 >= 1000000,]
pdom <- pd[pd$V4 >= 1000000,]

# Compute histograms of GC content
amel.h <- hist(amel$GCcontent, plot=FALSE)
bter.h <- hist(bter$GCcontent, plot=FALSE)
hsal.h <- hist(hsal$GCcontent, plot=FALSE)
mrot.h <- hist(mrot$GCcontent, plot=FALSE)
nvit.h <- hist(nvit$GCcontent, plot=FALSE)
pdom.h <- hist(pdom$GCcontent, plot=FALSE)

png("hym-gc-dists.png", height=1000, width=1000, res=150)
plot(0, yaxt="n", ylab="", ylim=c(0,1), xlim=c(20, 60), xlab="%GC content", bty='n')
rug(amel$GCcontent, col="gold",    side=3, ticksize=.1, line=0)
rug(bter$GCcontent, col="#999900", side=3, ticksize=.1, line=-4)
rug(hsal$GCcontent, col="red",     side=3, ticksize=.1, line=-8)
rug(mrot$GCcontent, col="#009900", side=3, ticksize=.1, line=-12)
rug(nvit$GCcontent, col="blue",    side=3, ticksize=.1, line=-16)
rug(pdom$GCcontent, col="black",   side=3, ticksize=.1, line=-20)
text(40, 0.975, "A. mellifera",   col="gold",    pos=4)
text(44, 0.81,  "B. terrestris",  col="#999900", pos=4)
```

```
text(37, 0.63,  "H. saltator",    col="red",      pos=2)
text(47, 0.45,  "M. rotundata",   col="#009900", pos=4)
text(45, 0.27,  "N. vitripennis", col="blue",     pos=4)
text(35, 0.095, "P. dominula",    col="black",    pos=4)
d <- dev.off()
```

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
```

Lastly, convert all of the alignments into GFF3 format, filtering out alignments with GeneSeqer similarity or coverage scores < 0.8.

```
./gsq2makergff3.py < pdom-tsa-r1.2-masked.gsq > pdom-tsa-r1.2-masked.gff3
./gsq2makergff3.py < pmet-tsa-r1.2-masked.gsq > pmet-tsa-r1.2-masked.gff3
./gsq2makergff3.py < pcan-tsa-r1.2-masked.gsq > pcan-tsa-masked.gff3
```

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
```

6.3 References

- **Brendel V, Xing L, Zhu W** (2004) Gene structure prediction from consensus spliced alignment of multiple ESTs matching the same genomic locus. *Bioinformatics*, **20**:1157-1169, doi:10.1093/bioinformatics/bth058.
- **Berens *et al***, in preparation.
- **Ferreira P, Patalano S, Chauhan R, Ffrench-Constant R, Gabaldon T, Guigo R, Sumner S** (2013) Transcriptome analyses of primitively eusocial wasps reveal novel insights into the evolution of sociality and the origin of alternative phenotypes. *Genome Biology*, **14**(2):R20, doi:10.1186/gb-2013-14-2-r20.

7 Reference protein alignment

Reference protein sequences from Apis mellifera (OGS 3.2 & NCBI GNOMON) and Drosophila melanogaster (FlyBase r5.55) were splice-aligned to the genome using GenomeThreader version 1.6.0. The alignments have been deposited in the Pdom Data Store at `r1.2/protein-alignment`.

7.1 Procedure (interactive)

First, 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
```

Next, download the reference proteins.

```
BeeData=http://hymenopteragenome.org/beebase/sites/hymenopteragenome.org.beebase
curl ${BeeData}/files/data/consortium_data/amel_OGSv3.2_pep.fa.gz \
    | zcat \
    | sed 's/gnl|Amel_4.5|//g' \
    > amel-ogs-prot.fa
curl ftp://ftp.ncbi.nih.gov/genomes/Apis_mellifera/protein/protein.fa.gz \
    | zcat \
    | perl -ne 's/>gi\|(\d+)\|ref\|([^|]+)\|\S*/>$1 $2/; print' \
    > amel-ncbi-prot.fa
FlyData=ftp://ftp.flybase.net/releases/FB2014_01/dmel_r5.55
curl ${FlyData}/fasta/dmel-all-translation-r5.55.fasta.gz \
    | zcat \
    > dmel-flybase-prot.fa
```

Then perform the alignments.

```
# The $GTHBIN variable should point to the bin directory in the GenomeThreader
# distribution.
export BSSMDIR=$GTHBIN/bssm
export GTHDATADIR=$GTHBIN/gthdata

for prot in amel-ogs amel-ncbi dmel-flybase
do
  $GTHBIN/gth -genomic pdom-scaffolds-masked-r1.2.fa \
              -protein ${prot}-prot.fa \
              -species arabidopsis \
              -gcmaxgapwidth 20000 \
              -gcmincoverage 25 \
              -prhdist 6 \
              -prminmatchlen 18 \
              -prseedlength 6 \
              -o ${prot}-prot-masked.gth \
              -force \
              > ${prot}-prot-masked.log 2>&1
done
```

Finally, convert the GenomeThreader alignments into GFF3 format, filtering out alignments with similarity or coverage scores < 0.5.

```
./gth2makergff3.py < amel-ogs-prot-masked.gth > amel-ogs-prot-masked.gff3
./gth2makergff3.py < amel-ncbi-prot-masked.gth > amel-ncbi-prot-masked.gff3
./gth2makergff3.py < dmel-flybase-prot-masked.gth > dmel-flybase-prot-masked.gff3
```

7.2 Procedure (automated)

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

```
make GTHbin=/usr/local/bin/GTH
```

7.3 References

- **Gremme G, Brendel V, Sparks ME, Kurtz S** (2005) Engineering a software tool for gene structure prediction in higher organisms. *Information and Software Technology*, **47**(15):965-978, doi:10.1016/j.infsof.2005.09.005.

8 Genome annotation

The Maker annotation pipeline version 2.31.6 was used to annotate protein-coding genes and tRNA genes in the *P. dominula* genome. The final, cleaned up annotation file and corresponding sequence files have been deposited in the Pdom Data Store at `r1.2/genome-annotation/`.

8.1 Gene predictor training

Approximately 3000 genes were selected from annotations whose reliability was confirmed by strict measures of conservation with several other Hymenopteran species. These genes were then used to train 3 ab initio gene predictors (SNAP, Augustus, and GeneMark) for use with the Maker pipeline. For Augustus and SNAP, the procedures documented in their respective source code distributions were followed. For GeneMark, a model built by the self-training version of GeneMark was used with Maker. All models have been deposited in the Pdom Data Store at `r1.2/genome-annotation/models`.

8.2 Procedure (interactive)

8.2.1 Data files

First, there are *a lot* of data files to download. Make sure to set the `AUGUSTUS_CONFIG_DIR` variable appropriately.

```
PdomData=/iplant/home/standage/Polistes_dominula
AUGUSTUS_CONFIG_DIR=/usr/local/src/augustus/config
export AUGUSTUS_CONFIG_DIR

# Genome sequence
iget ${PdomData}/genome-assembly/pdom-scaffolds-masked-r1.2.fa.gz
gunzip pdom-scaffolds-masked-r1.2.fa.gz

# Transcript alignments
iget ${PdomData}/r1.2/transcript-alignment/pdom-r1.2-trans-align.tar.gz
tar -xzf pdom-r1.2-trans-align.tar.gz
mv pdom-r1.2-trans-align/*.gff3 .

# Protein alignments
iget ${PdomData}/r1.2/protein-alignment/pdom-r1.2-refrprot-align.tar.gz
tar -xzf pdom-r1.2-refrprot-align.tar.gz
mv pdom-r1.2-refrprot-align/*.gff3 .

# Highly conserved and manually curated annotations
iget ${PdomData}/r1.2/genome-annotation/pdom-viga-improved.gff3.gz
iget ${PdomData}/r1.2/genome-annotation/pdom-yrgate.gff3.gz
gunzip pdom-viga-improved.gff3.gz
```

```
gunzip pdom-yrgate.gff3.gz

# Download species-specific parameter settings for ab initio gene predictors
iget ${PdomData}/r1.2/genome-annotation/models/pdom.snap.hmm
iget ${PdomData}/r1.2/genome-annotation/models/pdom.genemark.mod
iget ${PdomData}/r1.2/genome-annotation/models/pdom.augustus.tar.gz
tar -xzf pdom.augustus.tar.gz
mv pdom ${AUGUSTUS_CONFIG_DIR}/species
```

8.2.2 Configuration files

Next, we need to create the Maker control files. If all of the supplementary programs are the in the path, the `maker_exe.ctl` file will be populated automatically. If not, you will need to manually fill it in with the location of all the programs.

```
maker -CTL
# Replace the empty `maker_opts.ctl` file with our file,
# which has all the values filled in.
cp pdom_maker_opts.ctl maker_opts.ctl
```

8.2.3 Annotation with Maker

After all of the data files and control files are in place, running Maker is trivial.

```
NumThreads=16
maker -genome pdom-scaffolds-masked-r1.2.fa \
      -fix_nucleotides \
      -nodatastore \
      -RM_off \
      -cpus $NumThreads \
      -base pdom \
      > pdom.maker.log 2>&1
```

8.2.4 Post-processing

The feature IDs created internally by Maker are pretty unwieldy, so we use a few scripts to clean them up.

```
# Merge and clean up data
gff3_merge -o pdom-annot-p1.2-raw.gff3 pdom.maker.output/pdom_master_datastore_index.log
bash clean.sh pdom-annot-p1.2-raw.gff3 > pdom-annot-p1.2-cleaned.gff3
# Fixes strange output artifact for some VIGA-based predictions
perl -n fix.pl < pdom-annot-p1.2-cleaned.gff3 > pdom-annot-p1.2-fixed.gff3

# Create a polished GFF3 file with official IDs for the annotations and proper
# ##sequence-region pragmas.
gt gff3 -retainids -sort -tidy pdom-annot-p1.2-fixed.gff3 2> gt.log \
    | python annot-ids.py --idfmt='Pdom%sr1.2-%05lu' -n --rnamap=rnaids.txt --dbxref=MAKER - \
    | python fix-regions.py pdom-scaffolds-masked-r1.2.fa > pdom-annot-r1.2.gff3

# Create transcript and protein sequence files with proper feature IDs
cat pdom.maker.output/pdom_datastore/PdomSCFr1.2-*/PdomSCFr1%2E2-*.maker.transcripts.fasta \
```

```
    | python seq-ids.py rnaids.txt > pdom-annot-r1.2-transcripts.fasta
cat pdom.maker.output/pdom_datastore/PdomSCFr1.2-*/PdomSCFr1%2E2-*.maker.proteins.fasta \
    | python seq-ids.py rnaids.txt > pdom-annot-r1.2-proteins.fasta
```

8.2.5 Functional annotation by BLASTp search

Translation products of all gene models predicted by Maker were searched against all animal proteins in the NCBI nr database. This BLAST search provides a preliminary functional annotation for the gene models, but also identified gene models without any matches to known proteins.

```
# NCBI nr database installed using update_blastdb.pl script
# distributed with BLAST.
blastdb_aliastool -gilist anm.p.gil -dbtype prot -db nr -out anm

# BLAST search
blastp -db anm \
       -query pdom-annot-r1.2-proteins.fa \
       -evalue 1e-4 \
       -num_threads $NumThreads \
       -outfmt 5 \
       -out pdom-r1.2-vs-anm-blastp.xml
```

Using the NCBI Taxonomy data, we can examine the best BLAST hit for each gene model and tally these up according to the family of the best hit.

```
# Download NCBI taxonomy data
mkdir taxonomy
cd taxonomy
curl -O ftp://ftp.ncbi.nih.gov/pub/taxonomy/taxdump.tar.gz
tar xzf taxdump.tar.gz
cd -

perl blastxml-to-besthit.pl taxonomy/names.dmp \
    < pdom-r1.2-vs-anm-blastp.xml \
    > pdom-annot-r1.2-besthit-gis.txt

# Breakdown of hits to Hymenoptera proteins, by family
./taxtrav --rank family \
        --filter 'order=Hymenoptera' \
        --nodes taxonomy/nodes.dmp \
        --names taxonomy/names.dmp \
        <(cut -f 2 pdom-annot-r1.2-besthit-gis.txt)
    > pdom-hym-hits-tax.csv
cut -f 3 -d , pdom-hym-hits-tax.csv | sort | uniq -c | sort -rn

# Number of hits to non-Hymenoptera proteins
./taxtrav --rank family \
        --filter 'order=Hymenoptera' \
        --nodes taxonomy/nodes.dmp \
        --names taxonomy/names.dmp \
        --complement \
        <(cut -f 2 pdom-annot-r1.2-besthit-gis.txt)
```

```
    > pdom-nothym-hits-tax.csv
wc -l pdom-nothym-hits-tax.csv
```

Finally, let's pull out the gene models with no matches in the database. Unmatched gene models tend to be short, but we can also pull out gene models whose length (in amino acids) exceeds the median length for all gene models and set these aside as for further examination.

```
comm -13 <(cut -f 1 pdom-annot-r1.2-besthit-gis.txt | sort) \
         <(perl -ne 'm/>(\S+)/ and print "$1\n"' < pdom-annot-r1.2-proteins.fa) \
    > pdom-r1.2-no-anm-hits.ids
./select-seq pdom-r1.2-no-anm-hits.ids \
             pdom-annot-r1.2-proteins.fa \
    > pdom-r1.2-no-anm-hits.fa
bash fasta-median-plus.sh pdom-annot-r1.2-proteins.fa \
                          pdom-r1.2-no-anm-hits.fa \
    > pdom-annot-r1.2-long-unmatched.fa
```

8.3 Procedure (automated)

The same procedure can also be run in batch mode using the following commands (in the `genome-annotation` directory). Keep in mind that unless all of the supplemental programs are in your path, you will still need to run `maker -CTL` and edit the `maker_exe.ctl` file manually.

```
make NumThreads=16
```

8.4 References

- **Cantarel BL, Korf I, Robb SMC, Parra G, Ross E, Moore B, Holt C, Alvarado AS, Yandell M** (2008) MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Research*, 18:88-196, doi:10.1101/gr.6743907.

9 Interval loci

The LocusPocus program (AEGeAn Toolkit version 53d092091c) was used to compute interval loci (iLoci) from the genome annotation. The xtractore program was used to extract the iLocus sequences from the genome sequence. The iLocus sequences and annotations have been deposited in the Pdom Data Store at `/r1.2/interval-loci`.

9.1 Procedure (interactive)

We first need the genome sequence and corresponding annotations.

```
PdomData=/iplant/home/standage/Polistes_dominula/r1.2
iget ${PdomData}/genome-annotation/pdom-annot-r1.2.gff3
iget ${PdomData}/genome-assembly/pdom-scaffolds-unmasked-r1.2.fa.gz
gunzip pdom-scaffolds-unmasked-r1.2.fa.gz
```

Then we compute the iLocus coordinates, excluding gene-less iLoci if they are at the end of a scaffold.

```
locuspocus --intloci --skipends \
           --delta=500 --verbose \
           --outfile=pdom-loci-r1.2.gff3 \
           pdom-annot-r1.2.gff3
```

The xtractore program gives us the iLocus sequences.

```
xtractore --type=locus --outfile=pdom-loci-r1.2.fa \
          pdom-loci-r1.2.gff3 \
          pdom-scaffolds-unmasked-r1.2.fa
```

For some analyses, we want gene coordinates expressed in reference to the iLocus to which they belong, as opposed to the entire scaffold. We transformed the annotations to iLocus-based coordinates and created two files, one including tRNA genes and one without.

```
perl glb2lcl.pl < pdom-loci-r1.2.gff3 \
    | gt gff3 -retainids -sort -tidy -o pdom-annot-r1.2-iloci.gff3

perl glb2lcl.pl < pdom-loci-r1.2.gff3 \
    | grep -v -e trnascan -e PdomTRNA \
    | gt gff3 -retainids -sort -tidy -o pdom-annot-r1.2-iloci-sanstrna.gff3
```

9.1.1 Comparison of genome composition

At the level of large genomic sequences (> 1Mbp), *P. dominula* has the most AT-rich genome observed in Hymenoptera. For comparative evaluation of genome composition at the level of gene loci, we collected data for 13 Hymenopteran species: see `hymenoptera-ilocus-stats-2014.tsv`. We applied several filters to the data, selecting loci within the 10% and 90% quantiles in length, containing no more than a single gene and no more than 25% ambiguous nucleotides. We then compared the nucleotide composition of *P. dominula* to that of *Apis mellifera*, noting that the trend is reversed at the resolution of individual genes.

```
# Import data
data <- read.table("hymenoptera-ilocus-stats-2014.tsv", header=TRUE, sep="\t")

# Filter by number of genes
data.sub <- data[data$GeneCount == 1,]

# Filter by N content
data.sub <- data.sub[data.sub$Ncontent < 0.25,]

# Filter by length quantiles
qnt <- quantile(data.sub$Length, probs=c(0.1, 0.9))
data.sub <- data.sub[data.sub$Length > qnt[1] & data.sub$Length < qnt[2],]

# Plot Amel vs Pdom
pdom <- data.sub[substr(data.sub$ID, 1, 4) == "Pdom",]
amel <- data.sub[substr(data.sub$ID, 1, 4) == "Amel",]
pdom.h <- hist(pdom$GCcontent, breaks=25, plot=FALSE)
amel.h <- hist(amel$GCcontent, breaks=25, plot=FALSE)
png("amel-pdom-iloci-gc.png", height=1000, width=1000, res=150)
plot(pdom.h$mids, pdom.h$counts, type="l", xlab="%GC content", ylab="Frequency", xlim=c(0,0.65))
lines(amel.h$mids, amel.h$counts, col="orange")
```

```
abline(v=median(pdom$GCcontent), lty=3)
abline(v=median(amel$GCcontent), lty=3, col="orange")
d <- dev.off()
```

9.2 Procedure (automated)

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

```
make
make clean
```

10 Novel genes

Two previous analyses produced potential sources of novel genes warranting further investigation.

- The first potential source of novel genes are gene models predicted by Maker that have no matches against animal proteins in the NCBI nr database. These are described in the genome annotation documentation and are referred to hereafter as *unmatched gene models*.
- The second potential source of novel genes are assembled transcripts that have no matches in protein databases but are conserved among the 3 *Polistes* species. These are described in the transcript assembly documentation and are referred to hereafter as *unmatched TSAs*.

Here, we look for interval loci (iLoci) that have support from both of these sources of evidence. Relevant data has been uploaded to the Pdom Data Store at `r1.2/novel-genes`.

10.1 Procedure (interactive)

First we need to download the relevant data.

```
PdomData=/iplant/home/standage/Polistes_dominula/r1.2
iget ${PdomData}/transcript-assembly/pdom-tsa-r1.2-unmatched-pep.txt
iget ${PdomData}/transcript-alignment/pdom-tsa-masked-filtered.gff3
iget ${PdomData}/interval-loci/pdom-loci-r1.2.gff3
iget ${PdomData}/interval-loci/pdom-loci-r1.2-mrnamap.txt
iget ${PdomData}/genome-annotation/pdom-r1.2-no-anm-hits.ids
iget ${PdomData}/transcript-assembly/pdom-tsa-r1.2.fa.gz
gunzip pdom-tsa-r1.2.fa.gz
```

Next, identify iLoci to which unmatched TSAs align.

```
# Pull out alignment coordinates for unmatched TSAs
python align-keep.py pdom-tsa-r1.2-unmatched-pep.txt \
                     pdom-tsa-masked-filtered.gff3 \
    > pdom-tsa-aligned-unmatched.gff3

# Find the corresponding iLoci
bedtools intersect -a <(grep $'\tlocus\t' pdom-loci-r1.2.gff3) \
                   -b pdom-tsa-aligned-unmatched.gff3 -wa -u \
    | perl -ne 'm/(PdomILCr1.2-\d+)/ and print "$1\n"' \
    | sort | uniq > pdom-tsa-aligned-unmatched-loci.txt
```

Then look at unmatched gene models and determine the iLoci to which they correspond.

```
./selex --out=1 pdom-r1.2-no-anm-hits.ids pdom-loci-r1.2-mrnamap.txt \
    | sort | uniq > pdom-r1.2-no-anm-hits-iloci.txt
```

Finally, determine which iLoci appear in both lists (iLoci to which unmatched TSAs align and iLoci containing unmatched gene models).

```
comm -12 pdom-r1.2-no-anm-hits-iloci.txt pdom-tsa-aligned-unmatched-loci.txt \
    > pdom-loci-r1.2-pot-trgs.txt
```

10.2 Procedure (automated)

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

```
make
make clean
```

11 Differential expression analysis

iLocus expression levels were determined for each replicate using RSEM version 1.2.7, which in turn utilizes Bowtie version 1.0.0 to perform read alignments. EBSeq version 1.1.5 (bundled with RSEM) was used to identify genes that are differentially expressed between queens and workers. The tk-rnaseq toolkit version 5ce0f8c075 was used at various stages to filter, process, and visualize the data. Various data tables, sequence files, and graphics described here have been deposited in the Pdom Data Store at `r1.2/diff-exp`.

11.1 Procedure (interactive)

The differential expression analysis took place in 3 stages. In the first stage we conducted the full analysis to collect information about each iLocus and each sample.

```
# Ensure the RSEM directory is in the $PATH variable.
# Download and trim RNA-Seq data as described in the
# transcriptome assembly.

# Download the iLoci
mkdir iloci
PdomData=/iplant/home/standage/Polistes_dominula
iget ${PdomData}/r.1/interval-loci/pdom-loci-r1.2.fa iloci/

# Prepare/index the iLocus sequences
mkdir -p logs/unfiltered
rsem-prepare-reference iloci/pdom-loci-r1.2.fa \
                       iloci/pdom-loci-r1.2 \
                       > logs/unfiltered/rsem-prep.log 2>&1

# Align the RNA-Seq reads, calculate expression estimates
mkdir -p abundances/unfiltered tmp/unfiltered
for caste in q w
```

```
do
  for rep in {1..6}
  do
    sample=${caste}${rep}
    rsem-calculate-expression --paired-end \
                              --temporary-folder=tmp/unfiltered/rsem-calc-${sample}-temp \
                              --time \
                              --num-threads=32 \
                              rnaseq-clean/pdom-rnaseq-${sample}-trim-paired-1.fq \
                              rnaseq-clean/pdom-rnaseq-${sample}-trim-paired-2.fq \
                              iloci/pdom-loci-r1.2 \
                              abundances/unfiltered/pdom-${sample} \
                              > logs/unfiltered/rsem-calc-${sample}.log 2>&1
  done
  wait
done

# Compile expression data matrix: 1 row per iLocus, 1 column per sample
rsem-generate-data-matrix abundances/unfiltered/pdom-q1.genes.results \
                          abundances/unfiltered/pdom-q2.genes.results \
                          abundances/unfiltered/pdom-q3.genes.results \
                          abundances/unfiltered/pdom-q4.genes.results \
                          abundances/unfiltered/pdom-q5.genes.results \
                          abundances/unfiltered/pdom-q6.genes.results \
                          abundances/unfiltered/pdom-w1.genes.results \
                          abundances/unfiltered/pdom-w2.genes.results \
                          abundances/unfiltered/pdom-w3.genes.results \
                          abundances/unfiltered/pdom-w4.genes.results \
                          abundances/unfiltered/pdom-w5.genes.results \
                          abundances/unfiltered/pdom-w6.genes.results \
                          > abundances/unfiltered/pdom.genes.results

# Determine differentially expressed genes, FDR < 0.05
rsem-run-ebseq abundances/unfiltered/pdom.genes.results \
               6,6 pdom.genes.diffexp > logs/unfiltered/ebseq.log 2>&1
rsem-control-fdr pdom.genes.diffexp \
                 0.05 pdom.genes.diffexp.sig.05

# Generate a heatmap to visualize DEGs
git clone https://github.com/standage/tk-rnaseq.git
tk-rnaseq/de-viz ebseqinfile=abundances/unfiltered/pdom.genes.results \
                 ebseqoutfile=pdom.genes.diffexp \
                 workdir=expression-data \
                 samples="Q1,Q2,Q3,Q4,Q5,Q6,W1,W2,W3,W4,W5,W6" \
                 all
```

In the second stage, we filtered the iLoci to remove outliers and sequences that exhibited too much inconsistency across replicated, and then re-ran the entire analysis workflow.

```
# Build a table containing various data about each iLocus, most notably:
# the fold change; the number of raw reads mapping for each sample; and
# the unnormalized and normalized expression estimates for each sample.
make -f tk-rnaseq/build-table -j 16 \
     ebseqinfile=abundances/unfiltered/pdom.genes.results \
```

```
     ebseqoutfile=pdom.genes.diffexp \
     bamfilepattern=abundances/unfiltered/pdom-*.transcript.sorted.bam \
     fasta=iloci/pdom-loci-r1.2.fa \
     workdir=expression-data \
     all

# Filter the iLoci based on the specified criteria.
tk-rnaseq/de-filter --density=0.01,10 \
                    --nonzeros=5 \
                    --varcoef=1.0 \
                    --noheader \
                    --numsamples=6,6 \
                    < expression-data/expression-data.txt \
                    > expression-data/expression-data-filtered.txt

# Create a new fasta file containing only the iLoci that satisfy
# the filtering criteria.
tk-rnaseq/select-seq <(cut -f 1 expression-data/expression-data-filtered.txt) \
                     iloci/pdom-loci-r1.2.fa \
                     > iloci/pdom-loci-r1.2-filtered.fa

# Prepare/index the filtered iLocus sequences
mkdir -p logs/filtered
rsem-prepare-reference iloci/pdom-loci-r1.2-filtered.fa \
                       iloci/pdom-loci-r1.2-filtered \
                       > logs/filtered/rsem-prep.log 2>&1

# Align the RNA-Seq reads, calculate expression estimates
mkdir -p abundances/filtered tmp/filtered
for caste in q w
do
  for rep in {1..6}
  do
    sample=${caste}${rep}
    rsem-calculate-expression --paired-end \
                              --temporary-folder=tmp/filtered/rsem-calc-${sample}-temp \
                              --time \
                              --num-threads=48 \
                              rnaseq-clean/pdom-rnaseq-${sample}-trim-paired-1.fq \
                              rnaseq-clean/pdom-rnaseq-${sample}-trim-paired-2.fq \
                              iloci/pdom-loci-r1.2-filtered \
                              abundances/filtered/pdom-${sample} \
                              > logs/filtered/rsem-calc-${sample}.log 2>&1
  done
  wait
done

# Compile expression data matrix
rsem-generate-data-matrix abundances/filtered/pdom-q1.genes.results \
                          abundances/filtered/pdom-q2.genes.results \
                          abundances/filtered/pdom-q3.genes.results \
                          abundances/filtered/pdom-q4.genes.results \
                          abundances/filtered/pdom-q5.genes.results \
```

```
                          abundances/filtered/pdom-q6.genes.results \
                          abundances/filtered/pdom-w1.genes.results \
                          abundances/filtered/pdom-w2.genes.results \
                          abundances/filtered/pdom-w3.genes.results \
                          abundances/filtered/pdom-w4.genes.results \
                          abundances/filtered/pdom-w5.genes.results \
                          abundances/filtered/pdom-w6.genes.results \
                          > abundances/filtered/pdom.filtered.genes.alllibs.results

# Determine DEGs
rsem-run-ebseq abundances/filtered/pdom.filtered.genes.alllibs.results \
               6,6 pdom.filtered.genes.alllibs.diffexp > logs/filtered/ebseq-alllibs.log 2>&1
rsem-control-fdr pdom.filtered.genes.alllibs.diffexp \
                 0.05 pdom.filtered.genes.alllibs.diffexp.sig.05

# Visualize DEGs
tk-rnaseq/de-viz ebseqinfile=abundances/filtered/pdom.filtered.genes.alllibs.results \
                 ebseqoutfile=pdom.filtered.genes.alllibs.diffexp \
                 workdir=expression-data-filtered \
                 samples="Q1,Q2,Q3,Q4,Q5,Q6,W1,W2,W3,W4,W5,W6" \
                 all
```

In the final stage, we identified the least characteristic replicate from each condition (caste), removed those replicates, and re-ran the final step of the analysis workflow.

```
# Determine which replicate from each caste is the most similar to the other caste
tk-rnaseq/bully --samples="Q1,Q2,Q3,Q4,Q5,Q6,W1,W2,W3,W4,W5,W6" --numreps=6,6 \
                --norm=abundances/filtered/pdom.filtered.genes.alllibs.results \
                < expression-data-filtered/de.expr.dat.sorted

# Compile expression data matrix, this time sans outlier replicates
rsem-generate-data-matrix abundances/filtered/pdom-q1.genes.results \
                          abundances/filtered/pdom-q2.genes.results \
                          abundances/filtered/pdom-q3.genes.results \
                          abundances/filtered/pdom-q5.genes.results \
                          abundances/filtered/pdom-q6.genes.results \
                          abundances/filtered/pdom-w1.genes.results \
                          abundances/filtered/pdom-w2.genes.results \
                          abundances/filtered/pdom-w3.genes.results \
                          abundances/filtered/pdom-w4.genes.results \
                          abundances/filtered/pdom-w5.genes.results \
                          > abundances/filtered/pdom.filtered.genes.results

# Determine DEGs
rsem-run-ebseq abundances/filtered/pdom.filtered.genes.results \
               5,5 pdom.filtered.genes.diffexp > logs/filtered/ebseq.log 2>&1
rsem-control-fdr pdom.filtered.genes.diffexp \
                 0.05 pdom.filtered.genes.diffexp.sig.05

# Visualize DEGs
tk-rnaseq/de-viz ebseqinfile=abundances/filtered/pdom.filtered.genes.results \
                 ebseqoutfile=pdom.filtered.genes.diffexp \
                 workdir=expression-data-filtered-final \
```

```
                samples="Q1,Q2,Q3,Q5,Q6,W1,W2,W3,W4,W5" \
                all
```

11.2 References

- **Li B, Dewey CN** (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, **12**:323, doi:10.1186/1471-2105-12-323.
- **Langmead B, Trapnell C, Pop M, Salzberg SL** (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, **10**:R25, doi:10.1186/gb-2009-10-3-r25.
- **Leng N, Dawson JA, Thomson JA, Ruotti V, Rissman AI, Smits BMG, Haag JD, Gould MN, Stewart RN, Kendziorski C** (2013) EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, **29**(8):1035-1043, doi:10.1093/bioinformatics/btt087.

12 Analysis of alternative and differential splicing

The Tuxedo suite and the AEGeAn Toolkit were used to catalog alternative splicing events and identify genes that are differentially spliced between queens and workers. RNA-seq reads (trimmed for quality control*) were mapped to the filtered iLoci* using Tophat version 2.0.12 and Bowtie2 version 2.2.3. Cufflinks version 2.2.1 was then used to assemble the aligned reads perform differential splicing tests, while the `asinspect` program (AEGeAn version 4fc8909d9b) was used to construct a catalog of alternative splicing events. The output has been deposited in the Pdom Data Store at `r1.2/splicing`.

*Described in the section on differential expression.

12.1 Procedure

12.1.1 RNA-Seq alignments

First we align the RNA-Seq reads to the genome (iLoci) sample-by-sample. We start by downloading and indexing the iLocus sequences.

```
mkdir -p genome
PdomData=/iplant/home/standage/Polistes_dominula/
IlocusData=${PdomData}/r1.2/interval-loci
iget ${IlocusData}/pdom-loci-r1.2-filtered.fa genome/
iget ${IlocusData}/pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 genome/

cd genome
bowtie2-build pdom-loci-r1.2-filtered.fa pdom-loci-r1.2
cd -
```

Then we perform the alignments. Make sure the groomed RNA-Seq data* is in the `rnaseq-clean` directory.

```
NumThreads=16
mkdir -p alignments
for caste in q w
do
  for rep in {1..6}
  do
    sample=${caste}${rep}
```

```
    tophat --output-dir alignments/${sample} \
           --GTF genome/pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 \
           --mate-inner-dist 70 \
           --min-anchor 6 \
           --num-threads $NumThreads \
           genome/pdom-loci-r1.2 \
           rnaseq-clean/pdom-rnaseq-${sample}-trim-paired-1.fq \
           rnaseq-clean/pdom-rnaseq-${sample}-trim-paired-2.fq
  done
done
```

12.1.2 Assemblies

Next we need to assemble mapped reads to construct transcripts. We first assemble each sample individually.

```
mkdir -p assemblies
rm -f assemblies/all.txt
for caste in q w
do
  for rep in {1..6}
  do
    sample=${caste}${rep}
    echo assemblies/${sample}/transcripts.gtf >> assemblies/all.txt
    cufflinks --output-dir assemblies/${sample} \
              --num-threads $NumThreads \
              --multi-read-correct \
              --GTF-guide genome/pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 \
              alignments/${sample}/accepted_hits.bam
  done
done
```

Then we combine all assemblies into a single aggregate assembly.

```
cuffmerge -o assemblies/merged \
          --num-threads $NumThreads \
          --ref-sequence genome/pdom-loci-r1.2-filtered.fa \
          --ref-gtf genome/pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 \
          assemblies/all.txt
```

For tools that expect transcript structures in GFF3 format, we convert the final assembly from GTF to GFF3 (removing strandless single exon features).

```
gt gtf_to_gff3 -tidy assemblies/merged/merged.gtf \
    | perl -ne '@f = split(/\t/); next if($f[6] eq "."); print' \
    | gt gff3 -sort -tidy -o pdom-annot-r1.2-tuxedo.gff3 -force
```

12.1.3 Differential splicing

With a combined assembly, we can analyze the queen and worker data and look for genes showing caste-dependent alternative splicing patterns.

```
qaligns=$(ls alignments/q?/accepted_hits.bam | tr '\n' ',')
waligns=$(ls alignments/w?/accepted_hits.bam | tr '\n' ',')
qaligns=${qaligns%?}
waligns=${waligns%?}
cuffdiff --output-dir diff \
         --labels Queen,Worker \
         --num-threads $NP \
         --multi-read-correct \
         assemblies/merged/merged.gtf \
         $qaligns $waligns
```

12.1.4 Catalog alternative splicing events

Cuffdiff examines differential splicing patterns, but we also want to compose a catalog of alternative splicing events, ignoring whether the castes exhibit any difference in isoform preference.

```
asinspect --refr genome/pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 \
          --gtf \
          --out pdom-as.tsv \
          assemblies/merged/merged.gtf \
```

The `asinspect` program reports the following classes of alternative splicing.

- *Cassette exons*: an exon from the reference annotation is designated a cassette exon (CE) if its flanking exons are spliced together in any of the isoforms present in the Cufflinks assembly. The program also reports novel exons from the Cufflinks assembly that are present between 2 exons included in the reference assembly.
- *Retained introns*: an intron from the reference annotation is designated a retained intron (RI) if a single exon from the Cufflinks assembly matches the outer boundaries of the intron's flanking exons. The program also reports novel introns from the Cufflinks assembly that split a single exon from the Maker annotation into two new exons sharing the same outer boundaries.

We are particularly interested in further investigating conserved CE events. First we identify cassette exons that are entirely coding exons (i.e. contain no start or stop codon).

```
python coding-cassette-exons.py \
       pdom-as.tsv pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 \
       > pdom-as-ce-coding.tsv
```

Next we create a GFF3 and Fasta file containing both the exon-skipped isoform and the exon-contained isoform.

```
python ce-isoforms.py \
       pdom-as-ce-coding.tsv pdom-annot-r1.2-iloci-sanstrna-filtered.gff3 \
    | gt gff3 -retainids -sort -tidy \
    > pdom-annot-r1.2-withceisoforms.gff3

gt extractfeat -matchdescstart -join -retainids -translate \
               -seqfile pdom-loci-r1.2.fa -type CDS \
               pdom-annot-r1.2-withceisoforms.gff3 \
    | perl get_ce_iso.pl \
    > pdom-annot-r1.2-ce-isoforms.fa
```

Finally, we do a BLASTp search of these isoforms against *Apis mellifera* proteins. To identify high-confidence conserved CE events, we used MuSeqBox to filter the BLAST output and select the genes satisfying the following criteria.

- both isoforms have a match with expect value < 1e-20
- >= 90% reciprocal coverage between each query and match
- the CE-containing isoform has a different best match than the CE-skipped isoform

```
curl -O ftp://ftp.ncbi.nih.gov/genomes/Apis_mellifera/protein/protein.fa.gz
gunzip protein.fa.gz
mv protein.fa amel-ncbi-prot.fa
makeblastdb -in amel-ncbi-prot.fa -dbytype prot -parse_seqids
blastp -db amel-ncbi-prot.fa -query pdom-annot-r1.2-ce-isoforms.fa \
       -evalue 1e-6 -num_threads $NumThreads -out pdom-vs-amel.blastp

MuSeqBox -i pdom-vs-amel.blastp -n 1 -s 1 -l 25 -d 4 -L 128 -c ce-crtfile \
    > pdom-vs-amel-blastp.msb

python ce-filter.py < pdom-vs-amel-blastp.msb > pdom-vs-amel-conserved.txt
cut -f 1 -d '=' pdom-vs-amel-conserved.txt | sort | uniq \
    > pdom-annot-r1.2-ce-list.txt
```

12.2 Procedure (automated)

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

```
bash procedure.sh . 16
```

12.3 References

- **Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL** (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology*, **14**:R36, doi:10.1186/gb-2013-14-4-r36.
- **Langmead B, Salzberg SL** (2012) Fast gapped-read alignment with Bowtie 2. *Nature Methods*, **9**:357-359, doi:10.1038/nmeth.1923.
- **Trapnell C, Hendrickson D, Sauvageau S, Goff L, Rinn JL, Pachter L** (2013) Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology*, **31**:46-53, doi:10.1038/nbt.2450.