Appendix 2. Transcript Annotation and Seasonal Gene Expression in Kidney Tissue of the Australian Pygmy Bluetongue Skink, *Tiliqua adelaidensis*

Carmel Maher

2023

Contents

1 Analysis Workflow Outline

2 Assembly

3 Annotation

---

1 Analysis Workflow Outline

Transcript Annotation and Seasonal Gene Expression in Kidney Tissue of the Australian Pygmy Bluetongue Skink, *Tiliqua adelaidensis*

This document outlines the pipeline methods for assembly of poly-a selected transcripts extracted from kidney tissues of three male and five female Australian pygmy bluetongue skinks collected in South Australia during spring (September) and autumn (March or April). Long-read transcripts were sequenced from one individual "G6" using Pacific Biosciences' IsoSeq Sequel platform, and short-reads for all eight individuals were sequenced on an Illumina HiSeq. The wet-lab library preparation up to sequencing is outlined in Thesis Methods Chapter 2.

Sample ID	Experimental factors
G1	September Female
G2	September Female
G3	March/April Female
G4	March/April Male
G5	September Male
G6	September Male
G7	March/April Female
G8	March/April Female

There are five main sections to this document, which are referenced in three separate chapters of the thesis:

1. Analysis Workflow Outline

2. Assembly: Assembly, cleaning and clustering of transcript isoforms from **both** short-read and long-read sequencing formats to form a pseudo-reference for *T. adelaidensis* and to compare these methods (Chapter 3).

3. Annotation: Annotation and further analysis of the full length transcript isoforms derived from the long-read analysis (Chapter 3).

4. Comparison of Short-read assembly and Long-read transcripts: Comparison of BUSCO scores and exploratory alignment and BLASTn comparison of generated transcript isoforms between datasets (Chapter 3).

5. Gene Expression: Gene expression analysis using the short-read data aligned to the long-read pseudo-reference produced in Chapter 3, to assess seasonal variation of gene expression (Chapter 4).

Analysis was done using a combination of computing and HPC machines with specifications as below:

- Long-read cleaning, read frame prediction and blast searches were run on the Australian National eResearch Collaboration Tools and Resources project (NECTAR) cloud, using an 'm1.xlarge flavour' allocation with 8 AMD Opteron 63xx class CPUs, 32GB RAM, 10GB root disk, and 240GB of secondary disk storage.
- The ANGEL program which required a higher memory allocation was run on a Dell PowerEdge server with 40 cores and 512G RAM.
- Short-read QC, trimming, initial trinity assembly, and alignments were completed using Flinders University's high-performance computer "Deep thought" with 16 x 256Gb (4TB RAM), 1024 AMD x86 CPU cores at 2.0GHz across 14 standard compute and 2 data science nodes, and 100 TiB of usable storage via the Dell EMC Network Storage reference architecture.
- Downstream gene expression analysis and manual manipulation of annotation files was completed on a local Intel core i7 machine using RStudio or in-house bash scripts applied in Git Bash for Windows.

All larger scripts included in this document have also been uploaded to Github separately and are linked here at the relevant steps.
All scripts and commands in this pipeline depend on specific directory structure, and may show inconsistencies in this structure between steps where files have been transferred between computing resources. The input and output files of note are highlighted in the text between steps to bridge these gaps.

---

2 Assembly

A genome from a closely related species was unavailable for these data. Gene expression analysis of short-read data was initially intended to be carried out using a *de-novo* assembly of this same data as a pseudo-reference for expression counts.
When long-read data became available, the assembly of short read data, and the processed full length transcript datasets were compared. The long-read assembly then provided the reference for expression counts in Chapter 4.

This section is divided into the two methods of processing; using the long-read IsoSeq data, and using the short-read HiSeq data. Both these sections contain titles referencing the main program in use at each step.

2.1 *Long-read data*

Cleaning and clustering of long-read IsoSeq data sequenced from poly-a selected RNA extracted from a single kidney from september male G6. These sequences were be used to annotate genes found in kidney tissue of T. adelaidensis and create a set of reference transcripts for gene expression analysis.

Sequences were downloaded from Pacific Biosciences & checksumms verified.

2.1.1 SMRT Tools - IsoSeq3

SMRT Tools provides various programs for appropriate treatment of Pacific Biosciences sequencing outputs. IsoSeq3 was used to collapse circular sequences, initial polishing, and to characterise full length transcript reads. This analysis was conducted in an environment on the eRSA NECTAR research cloud.

A primer reference file was created manually for reference in the Isoseq3 script using the primers listed in the Clontech SMARTer cDNA Library prep kit which was used during the synthesis of cDNA for these samples.

primers.fasta:

```
>primer_5p
AAGCAGTGGTATCAACGCAGAGTACATGGG
>primer_3p
GTACTCTGCGTTGATACCACTGCTT
```

A single script was used to complete the Pacific Biosciences Isoseq3 pipeline isoseq3.sh:

```
#!/bin/bash
#
# This script is used to characterise full length transcripts de novo using
↪  smrttools and bioconda as specified by PacBio workflow
# see: https://github.com/PacificBiosciences/IsoSeq3/blob/master/README_v3.0.md and
↪  follow links for bioconda references
#
# The paths in the script assume that a specific directory structure has been set
↪  up.
# This directory structure is for the Nectar cloud VM

# open a screen
# usage from /mnt
```

```
# bash isoseq3.sh <ctrl-a ctrl-d>
#
# Carmel Maher
# April 2019


#---------Installed Programs---------

# smrttools               -> smrttools-release_6.0.0.47835
  # help:
  #
  ↪  /pacbio/smrtlink/install/smrtlink-release_6.0.0.47841/bundles/smrttools/install/
  ↪  smrttools-release_6.0.0.47835/smrtcmds/bin/isoseq3 -h
  #
  ↪  /pacbio/smrtlink/install/smrtlink-release_6.0.0.47841/bundles/smrttools/install/
  ↪  smrttools-release_6.0.0.47835/smrtcmds/bin/isoseq3 --version
  # Version: isoseq3 3.0.0 (commit v3.0.0-7-gcc6cddd)

# python 2.7          -> to run miniconda - automatically installed on nectar
# miniconda           -> recommended python 2.7 compatible version by PacBio
↪  workflow. Miniconda2 | VER: 4.6.14
# ccs                     -> installed through conda, inside pbccs package | VER:
↪  pbccs-3.4.1
# lima                    -> installed through conda | VER: lima-1.9.0


#------adjust these for your run------

#as per the local installation, the smrttools program directories are (in /mnt).
↪  Notable directories listed in installation:
SMRT_ROOT=/mnt/pacbio/smrtlink

# Isoseq3 executeable location (from /mnt)
Iso3_DIR=$SMRT_ROOT/install/smrtlink-release_6.0.0.47841/bundles/smrttools/install/
↪  smrttools-release_6.0.0.47835/smrtcmds/bin

DATA=/mnt/data
MOVIE="MAH6260A1_m54196_190204_223227"
THREADS=0     #0 for the Iseseq3 tools =autodetection of threads and is the default


#------------------------------------

function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}
```

```
# go to the working data directory
cd $DATA || error_exit "$LINENO: directory error"

# Generate consensus sequnces (ccs file) from raw subread data
# NOTE: this automatically runs with default autodetection of threads =  8
ccs $MOVIE.subreads.bam $MOVIE.ccs.bam --noPolish --minPasses 1 || error_exit
↪  "$LINENO: consensus sequence ccs error"

echo " - - - consensus sequnces generated"

# The primers.fasta as per the recommended Clontech SMARTer cDNA library prep - no
↪  barcodes used for this sample
# primers.fasta has been uploaded to /mnt/data and does not need to be created
↪  here:
   # primers.fasta
   # >primer_5p
   # AAGCAGTGGTATCAACGCAGAGTACATGGG
   # >primer_3p
   # GTACTCTGCGTTGATACCACTGCTT

# Remove primers and demultiplex:
lima --isoseq --dump-clips --no-pbi $MOVIE.ccs.bam primers.fasta demux.bam ||
↪  error_exit "$LINENO: Primer demultiplex lima error"

echo " - - - primers removed and demultiplexed"

       #!#!#!#!#
       # CHECK #  Note: A search using Git bash on the output files for any
       ↪  of the primers listed for Isoseq by Pacific biosciences returns
       ↪  no results after this script
       #!#!#!#!#

# ********************
# From here on, execute the following steps for each output BAM file. -- only one
↪  in this case.
# isoseq3 tool usage: isoseq3 <tool>
# due to nectar permissions and installation location, usage: $Iso3_DIR/isoseq3
↪  <tool>
# ********************

### cluster Tool: Cluster CCS reads and generate unpolished transcripts.
# recommended to give this step as many cores as possible
# Usage
# isoseq3 cluster [options] input output
# Example
# isoseq3 cluster <two types of potential inputfile> demux.bam <OR>
↪  $MOVIE.consensusreadset.xml  unpolished.bam

# Cluster consensus sequences to generate unpolished transcripts:
# note the demux.bam is named after the headers used in primers.fasta
$Iso3_DIR/isoseq3 cluster --verbose -j $THREADS demux.primer_5p--primer_3p.bam
↪  unpolished.bam || error_exit "$LINENO: Isoseq3 cluster error"
```

```
echo " - - - css reads clustered"

### polish Tool: Polish transcripts using subreads.
# Usage
# isoseq3 polish [options] input_1 input_2 output

# Polish transcripts using subreads:
$Iso3_DIR/isoseq3 polish --verbose -j $THREADS unpolished.bam $MOVIE.subreads.bam
↪  polished.bam || error_exit "$LINENO: Isoseq3 polish error"

echo " - - - transcripts polished"

### summarize Tool: Create a .csv-format barcode overview from transcripts.
# Usage
# isoseq3 summarize [options] input output
$Iso3_DIR/isoseq3 summarize --verbose polished.bam summary.csv || error_exit
↪  "$LINENO: Isoseq3 summary error"

echo " - - - summary file created"
echo " - - - done"
```

All of the above output was moved into a folder ~/data/2019-04-30_isoseq3_out

The output from this script, the polished.hq.fasta.gz file was unzipped and then used as the input for the program Cogent.

2.1.2 Cogent

Transcript clustering and collapse of redundant isoforms. Line by line instructions for running Cogent for clustering of transcript isoforms in a screen on the NECTAR machine as below. Includes instructions for Cupcake and minimap to collapse redundant transcript isoforms.

Some notes and commentary are based on personal communication with Tessa Bradford & Terry Bertozzi at the SA Museum, as well as the Cogent GitHub Tutorial page

Environments were used for installations (most often achieved with mini conda) on the NEXTAR machine. Activate screen, activate the environment, make Cogent output directory, and go to analysis working directory

```
screen -s cogent
conda activate anaCogent
mkdir /mnt/IsoSeq-analysis/data/Cogent
cd /mnt/IsoSeq-analysis/data/Cogent
```

From the analysis directory the input data directory is ../2019-04-30_out/polished.hq.fasta. The first output will be in this original data directory. Check paths (During installation, all paths were put into .profile)

2.1.2.0.1 Family Finding Running Family Finding for a small dataset

Create a k-mer profile of the input and calculate pairwise distances
Note: default K-mer sixe =30

```
python /mnt/Prog/Cogent/Cogent/run_mash.py --cpus=7
 ↪  /mnt/IsoSeq-analysis/data/2019-04-30_out/polished.hq.fasta
```

The output will be <fasta_filename>.k<sketch_size>.dist
Output written to: /mnt/IsoSeq-analysis/data/2019-04-30_out/polished.hq.fasta.s1000k30.dist

Process the distance file and create the partitions for each gene family

```
process_kmer_to_graph.py ../2019-04-30_out/polished.hq.fasta
 ↪  ../2019-04-30_out/polished.hq.fasta.s1000k30.dist ./hq hq
```

This generates an output log file hq.partition.txt and for each partition (isoform set), a subdirectory called hq/<partition_number> which contains the subset of fasta sequences belonging to that isoform set. Note that sequences that don't belong to any partition (ones with no similarity with other sequences) will be "unassigned" and noted in the partition log file.

2.1.2.0.2 Coding Genome Reconstruction Each isoform set family must be reconstructed individually for coding region.
Reconstructed contigs will contain the whole coding region. Reconstructed file is called cogent2.renamed.fasta

The script for an individual folder is:

```
reconstruct_contig.py -S T.adelaidensis hq/hq_0
```

The following script was written to loop this procedure and process hundreds of isoform set folders with sequences requiring coding region identification reconstructContig_1.sh:

```
#!/bin/bash
#
## this script is for coding region reconstruction of Cogent gene families as these must be completed individually.
# This script assumes you are following Running-Cogent.txt and using the same directory structure
# This script to be used after the process_kmer_to_graph.py step

# script in /mnt/IsoSeq-analysis/src
# usage from /mnt/IsoSeq-analysis/data/Cogent/hq
## bash ../../../src/reconstructContig.sh

# Carmel Maher
# August 2019

#Terry's error exit
function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message
    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}

cd ~/mnt/IsoSeq-analysis/data/Cogent/hq

for d in */ ; do

  DIR=${d%*}
  echo $DIR

  reconstruct_contig.py -S T.adelaidensis ./$DIR || error_exit "$LINENO: Error "$DIR""

done
echo compete
```

An error message prompted further checking of folder completion as below

Number of gene families:

```
ls ./hq/*/cogent2.renamed.fasta | wc -1
4454
```

Finding failed reconstruction jobs (pers. comm. Terry Bertozzi, SA Museum)

Number of hq gene folders:

```
ls hq | wc -l #number of folders
4459
```

Number of /hq folders completed:

```
ls ./hq/*/COGENT.DONE | wc -l
4454
```

This matches the total gene family count so all outputs have been recorded correctly, but does not match the total /hq folder count = 5 folders not completed.

The following will list the folders that have not finished:

```
comm -3 <(find /mnt/IsoSeq-analysis/data/Cogent/hq -iname 'COGENT.DONE' -printf
↪  '%h\n' | sort -u) <(find /mnt/IsoSeq-analysis/data/Cogent/hq -maxdepth 1
↪  -mindepth 1 -type d | sort) | sed -e 's/^.*hq\///'

hq_2605
hq_2738
hq_320
hq_6391
hq_9030
```

Some testing was required to create a modified script with a loop for failed folders, in order to include 5 folders which failed to run with kmer length 30 informed by this issue.
Due to how the loop functions inputting kmer size of 27 on execution will actually run all folders at the desired kmer size 30.

```
bash /mnt/IsoSeq-analysis/src/TBreconstructContig-edit.sh
↪  /mnt/IsoSeq-analysis/data/Cogent/hq 27 T.adelaidensis
```

The modified script TBreconstructContig-edit.sh:

```
#!/bin/bash

# This script runs reconstruct_contig.py on every subdirectory contained in the top
↪  directory. reruns with a kmer parameter of +3 (to a max of 99) for failed
↪  directories.
# This script assumes directories and files exist as up until the Coding Genome
↪  Reconstruction step in the Running Cogent GitHub wiki
#
# usage:  bash <path/script> <full directory containing hq_folders> <starting kmer
↪  value> <Species Name>
# e.g.        bash /mnt/IsoSeq-analysis/src/TBreconstructContig-edit.sh
↪  /mnt/IsoSeq-analysis/data/Cogent/hq 27 T.adelaidensis

# DIRNAME=$1
# species=$3

kmerSize=$2

find $1 -mindepth 1 -maxdepth 1 -type d | while read line; do #lists off all the
↪  /hq_* folders within the input directory and passes them through the script one
↪  by one (by line)
  FILE=$line/cogent2.fa
```

```
 if [ ! -f $FILE ]; then
   echo "failed no cogent2.fa Increase K-mer size" #Note all jobs will "fail" on
   ↪ first attempt unless you are running this in a previously reconstructed
   ↪ directory. In the latter case only the failed directories will continue
   ↪ to rerun.
   while [ ! -f "$FILE" ];do
     kmerSize=$((kmerSize + 3))
     rerunCMD="reconstruct_contig.py -k ${kmerSize} -S $3 $line"
     ↪ #reconstruction script increasing kmer parameter by 3 -note because
     ↪ all directories fail initially the input kmer value should be 3 below
     ↪ the desired actual value.
     echo ${rerunCMD}
     eval ${rerunCMD}
     if [ -f $FILE ]; then
       touch $1/hq_kmerSize.txt #This text file will give a list of the used
       ↪ kmerSize per hq_* folder
       echo "SUCCESS! The increased Kmer size $kmerSize was successful for
       ↪ $line." >> $1/hq_kmerSize.txt
     fi
     if [ ${kmerSize} -gt 99 ];then
       cp $line/in.fa $line/cogent2.fa # if a folder reruns until 99kmer
       ↪ parameter length is reached the input is simply copied into
       ↪ cogent2.fa
       touch $1/failed-jobs.txt #This text file will remain empty or give a
       ↪ list of all failed folders
       echo "Failed to succeed reconstruction with largest Kmer on ${line}.
       ↪ Copied input transcripts to cogent2.fa as output." >>
       ↪ $1/failed-jobs.txt
     fi
   done
   kmerSize=$2
 fi
done

if [ -s $1/failed-jobs.txt ]; then
  echo "failed-jobs.txt not empty. Some jobs failed. Please re-run them."
else
  echo "failed-jobs.txt empty or doesnt exist. All jobs completed."
fi
```

Re-check output.

Number of completed gene families:

```
ls ./hq/*/COGENT.DONE | wc -l
4459
```

OR:

```
ls ./hq/*/cogent2.renamed.fasta | wc -l
4459
```

Finding failed reconstruction jobs.

Number of items in ./hq:

```
ls hq | wc -l
4460
```

The added 1 is the text file reporting the successful kmer size for each "gene" family folder. Therefore all have completed.

List the names of folders that have not finished:

```
comm -3 <(find /mnt/IsoSeq-analysis/data/Cogent/hq -iname 'COGENT.DONE' -printf
↪  '%h\n' | sort -u) <(find /mnt/IsoSeq-analysis/data/Cogent/hq -maxdepth 1
↪  -mindepth 1 -type d | sort) | sed -e 's/^.*hq\///'
```

(this command listed no folders, indicating the script has completed all folders successfully)

See failed-jobs.txt and/or hq_kmerSize.txt. Note: all completed at kmer size 30 this time, and filed-jobs.txt does not exist as none failed.

2.1.2.0.3 Using Cogent to collapse redundant transcripts in absence of genome Creating the "fake genome"

List and number of unassigned sequences
Make sure you are in the correct Cogent directory and can see the file hq.partition.txt:

```
tail -n 1 hq.partition.txt |tr ',' '\n' > unassigned.list
```

Create the unassigned.list file in the same directory:

```
tail -n 1 hq.partition.txt |tr ',' '\n' | wc -l
3193
```

Note: this number did not change with the addition of the five failed reconstruction folders. i.e. they failed completely and did not end up in the unassigned dataset.

The unassigned hq sequences were made into a fasta file.

```
export PATH=$PATH:/mnt/Prog/cDNA_Cupcake/sequence
get_seqs_from_list.py ../2019-04-30_out/polished.hq.fasta unassigned.list >
↪  unassigned.fasta
```

Unassigned sequences were concatenated with the Cogent reconstructed contigs by putting the reconstructed genes plus the unassigned single hq isoforms into a single fasta file:

```
mkdir collected
cd collected
cat ../hq/*/cogent2.renamed.fasta ../unassigned.fasta > cogent_fake_genome.fasta
```

Redundant isoforms were collapsed:

Create a SAM alignment with minimap2 (This could also be done with GMAP).
Obtain a final set of unique (non-redundant) transcript Isoforms that can be used as a reference gene set.
As there is natural 5' degradation in RNA some sequences will represent identical isoforms which may not all be identified in clustering.
These parameters are default or suggested in the Cogent tutorial page.

Create an aligned, sorted SAM file:

```
export PATH=$PATH:/mnt/Prog/minimap2
minimap2 -ax splice -t 30  -uf --secondary=no cogent_fake_genome.fasta
↪  ../../2019-04-30_out/polished.hq.fasta > hq.fasta.sam
```

Output:

```
[M::mm_idx_gen::2.392*1.00] collected minimizers
[M::mm_idx_gen::3.526*1.32] sorted minimizers
[M::main::3.528*1.32] loaded/built the index for 9960 target sequence(s)
[M::mm_mapopt_update::3.708*1.30] mid_occ = 31
[M::mm_idx_stat] kmer size: 15; skip: 5; is_hpc: 0; #seq: 9960
[M::mm_idx_stat::3.810*1.29] distinct minimizers: 6267388 (82.12% are singletons);
↪  average occurrences: 1.346; average spacing: 2.846
[M::worker_pipeline::13.901*5.65] mapped 25117 sequences
[M::main] Version: 2.11-r797
[M::main] CMD:
↪  /mnt/Prog/pacbio/smrtlink/install/smrtlink-release_6.0.0.47841/bundles/smrttools/
↪  install/smrttools-release_6.0.0.47835/private/thirdparty/minimap2/minimap2_2.11/binwrap/../../
↪  private/thirdparty/minimap2/minimap2_2.11/bin/minimap2 -ax splice -t 30 -uf
↪  --secondary=no cogent_fake_genome.fasta ../../2019-04-30_out/polished.hq.fasta
[M::main] Real time: 13.942 sec; CPU: 78.558 sec
```

Then follow the collapse tutorial from Cupcake
There is another cupcake page with information but the example scripts are provided on the cogent tutorial page.

Sort the SAM file:

```
sort -k 3,3 -k 4,4n hq.fasta.sam > hq.fasta.sorted.sam
```

Colapse identical isoforms to obtain a list of full-length, unique, hq isoforms to use as reference transcripts:

```
cd /mnt/Prog/cDNA_Cupcake/sequence
which collapse_isoforms_by_sam.py
/home/ubuntu/miniconda2/envs/anaCogent/bin/collapse_isoforms_by_sam.py

cd /mnt/IsoSeq-analysis/data/Cogent/collected
```

Usage is:

```
# usage: collapse_isoforms_by_sam.py [-h] [--input INPUT] [--fq] -s SAM -o
#                                    PREFIX [-c MIN_ALN_COVERAGE]
```

```
#                                     [-i MIN_ALN_IDENTITY]
#                                     [--max_fuzzy_junction MAX_FUZZY_JUNCTION]
#                                     [--flnc_coverage FLNC_COVERAGE]
#                                     [--dun-merge-5-shorter]
```

Collapse:

```
collapse_isoforms_by_sam.py --input ../../2019-04-30_out/polished.hq.fasta -s
↪  hq.fasta.sorted.sam -c 0.94 -i 0.85 --dun-merge-5-shorter -o hq.fasta.no5merge
```

These parameters taken from pers. comm. Tessa Bradford, SA Museum.
Output is the name 'stem' the collapsed.rep.fa is for annotation.
The output files are <-o>.collapsed.gff, <-o>.collapsed.rep.fq, and <-o>.collapsed.group.txt i.e. hq.fasta.no5merge.collapsed.g
hq.fasta.no5merge.collapsed.rep.fq, and hq.fasta.no5merge.collapsed.group.txt

The naming system for the post-collapse isoform is PB.<loci_index>.<isoform_index>

More terminal output from the above has been excluded. This is an example:

```
Ignored IDs written to: hq.fasta.no5merge.ignored_ids.txt
Output written to:
hq.fasta.no5merge.collapsed.gff
hq.fasta.no5merge.collapsed.group.txt
hq.fasta.no5merge.collapsed.rep.fa
Namespace(allow_extra_5exon=False, flnc_coverage=-1, fq=False,
 ↪  input='../../2019-04-30_out/polished.hq.fasta', max_3_diff=100,
 ↪  max_5_diff=1000, max_fuzzy_junction=5, min_aln_coverage=0.94,
 ↪  min_aln_identity=0.85, prefix='hq.fasta.no5merge', sam='hq.fasta.sorted.sam')
```

Check outputs:

```
ls

cogent_fake_genome.fasta                        hq.fasta.no5merge.collapsed.rep.fa
hq.fasta.no5merge.collapsed.gff                 hq.fasta.no5merge.ignored_ids.txt
hq.fasta.no5merge.collapsed.gff.unfuzzy         hq.fasta.sam
hq.fasta.no5merge.collapsed.group.txt           hq.fasta.sorted.sam
hq.fasta.no5merge.collapsed.group.txt.unfuzzy
```

The file hq.fasta.no5merge.collapsed.group.txt names the isoforms as PB.<loci_index>.<isoform_index> and lists the collapsed identical isoforms.
Each 'locus' consists of a strand-specific locus with isoforms that overlap by at least 1 bp. So PB.11.1 and PB.11.2 means this locus has two isoforms.

Count the 'loci' (transcripts) found:

```
wc -l hq.fasta.no5merge.collapsed.group.txt
15729 hq.fasta.no5merge.collapsed.group.txt
```

Write file statistics:

```
get_abundance_post_collapse.py hq.fasta.no5merge.collapsed
↪  /mnt/IsoSeq-analysis/data/2019-04-30_out/polished.cluster_report.csv

WARNING: isoseq3 format detected. Output `length` column will be `NA`.

Read stat file written to hq.fasta.no5merge.collapsed.read_stat.txt
Abundance file written to hq.fasta.no5merge.collapsed.abundance.txt
```

Note: this is true, hq.fasta.no5merge.collapsed.read_stat.txt has no length information and a whole column of NA values, however the headers of hq.fasta.no5merge.collapsed.rep.fa do have length information. This is how IsoSeq3 format data is processed and does not affect other aspects of this script.

Get count information from the abundance and group .txt files

Add the headers:

```
echo -e "pbid\tcount_fl" > output.collapsed.abundance.txt
```

Add the PB.loci_index.isoform_index information and number of isoforms:

```
paste <(awk '{print $1}' hq.fasta.no5merge.collapsed.group.txt) <(awk -F , '{print
↪  NF}' hq.fasta.no5merge.collapsed.group.txt) >> output.collapsed.abundance.txt
```

All these commands except reconstructContig_1.sh and TBreconstructContig.edit.sh are provided in the one document Running-Cogent-2.sh on Github.

The longest transcript for each set of isoforms is in hq.fasta.no5merge.collapsed.rep.fa. This file is used in the following step and also provides the full transcripts to be sub-set as a reference based on further clustering.

2.1.3 ANGEL

ANGEL: Robust Open Reading Frame prediction

Line by line instructions for running ANGEL in order to predict open reading frame and peptide sequences of transcript Isoforms. Also outputs 5' and 3' untranslated regions and scores transcript completeness.

This analysis was tested on an environment on the eRSA NECTAR research cloud, final data run was completed by Terry Bertozzi on a Dell PowerEdge server with 40 cores and 512G RAM due to memory limitations of the NECTAR allocation.

All steps and script parameters are accurate here, however, this analysis was run on a different HPC machine due to memory constraints. Directories in these commands are consistent with input data and where output was then moved back to, and serve as an example only. All results were immediately placed back in ~IsoSeq-analysis/data/ANGEL

```
#The python dependencies for ANGEL are:

numpy
Biopython
scikit-learn
CD-HIT version 4.8.1
conda install -n anaCogent scikit-learn

mkdir /mnt/IsoSeq-analysis/data/ANGEL
cd /mnt/IsoSeq-analysis/data/ANGEL
```

2.1.3.0.1 Dumb ORF prediction dumb_predict.py takes as input a FASTA file. It outputs the longest ORF in all frames - use to create a top training dataset.

Usage:

```
dumb_predict.py <fasta_filename> <output_prefix>
        [--min_aa_length MIN_AA_LENGTH]
        [--use_firstORF]
        [--use_rev_strand] [--cpus CPUS]
```

By default, only the forward strand is used. This is especially true for PacBio transcriptome sequencing output.

```
dumb_predict.py
↪  /mnt/IsoSeq-analysis/data/Cogent/collected/hq.fasta.no5merge.collapsed.rep.fa
↪  pygmy.dumb --min_aa_length 100 --cpus 24

# ------
13335   finished        7611   clusters
```

2.1.3.0.2 Creating a non-redundant training dataset ANGEL classifier training:

```
angel_make_training_set.py pygmy.dumb.final pygmy.dumb.final.training --random
↪  --cpus 24

angel_train.py pygmy.dumb.final.training.cds pygmy.dumb.final.training.utr
↪  pygmy.dumb.final.classifier.pickle --cpus 12
```

2.1.3.0.3 Robust ORF prediction Based on both the ANGEL Classifier training and the dumb ORF prediction.

Sequences are output as fasta files with whether they are complete, 5' partial, 3' partial, or internal in the header information, which also includes the aa length.
Preidctions are tagged as confident, likely, or suspicious, and dumb ORF predictions as dumb.
The proportion of each isoform that is untranslated, as well as the position of the cds sequence is also output.

```
angel_predict.py ../Cogent/collected/hq.fasta.no5merge.collapsed.rep.fa
↪  pygmy.dumb.final.classifier.pickle pygmy --output_mode=best
↪  --min_angel_aa_length 100 --min_dumb_aa_length 100 --cpus 48
```

Output is written to:

- pygmy.ANGEL.cds
- pygmy.ANGEL.pep
- pygmy.ANGEL.utr

These outputs are used further below for additional clustering and BLASTx searches.

Note: The reverse strand options within ANGEL are not necessary on Isoseq data.

All these commands are provided in the one document Running_ANGEL.sh on Github.

2.1.4 CD-HIT

Clustering of transcript isoforms. Notes and output recorded for CD-HIT analysis of peptide sequences output from ANGEL ORF prediction. This analysis was conducted on the CD-HIT web server.

CD-HIT was run by uploading pygmy.ANGEL.pep to the server and selecting a cut-off of 0.99.
CD-HIT notes and recorded output are in CDHit notes.pep99.bash and below.

```
cd-hit - run on pygmy.ANGEL.pep

separate run cut-off at .99

***
Your job 1598936109 is finished.
Program you ran: cd-hit
You input file is pygmy.ANGEL.pep and we named it as 1598936109.fas.0
Summary information for 1598936109.fas.0 included in 1598936109.fas.0.stat
You required 1 runs for sequence clustering
     1. Fasta file for representative sequences at 99% identity is 1598936109.fas.1
         Summary information for 1598936109.fas.1 included in 1598936109.fas.1.stat
         Corresponding cluster file is1598936109.fas.1.clstr
         Sorted cluster file by size is 1598936109.fas.1.clstr.sorted
Generated shell script is run-1598936109.sh

faa_stat.pl 1598936109.fas.0
faa_stat.pl 1598936109.fas.1
/data5/data/NGS-ann-project/apps/cd-hit/clstr_sort_by.pl no <
↪  1598936109.fas.1.clstr > 1598936109.fas.1.clstr.sorted
/data5/data/NGS-ann-project/apps/cd-hit/clstr_list.pl 1598936109.fas.1.clstr
↪  1598936109.clstr.dump
gnuplot1.pl < 1598936109.fas.1.clstr > 1598936109.fas.1.clstr.1; gnuplot2.pl
↪  1598936109.fas.1.clstr.1 1598936109.fas.1.clstr.1.png
/data5/data/NGS-ann-project/apps/cd-hit/clstr_list_sort.pl 1598936109.clstr.dump
↪  1598936109.clstr_no.dump
/data5/data/NGS-ann-project/apps/cd-hit/clstr_list_sort.pl 1598936109.clstr.dump
↪  1598936109.clstr_len.dump len
/data5/data/NGS-ann-project/apps/cd-hit/clstr_list_sort.pl 1598936109.clstr.dump
↪  1598936109.clstr_des.dump des
```

The generated CD-HIT script is uploaded to github as run-1598936109.sh

The full output metadata file is uploaded to github as 1598936109.out and the summary data provided below:

```
================================================================
Program: CD-HIT, V4.7 (+OpenMP), Sep 06 2018, 09:36:42
Command: /data5/data/NGS-ann-project/apps/cd-hit/cd-hit -i
         1598936109.fas.0 -d 0 -o 1598936109.fas.1 -c 0.99 -n 5
         -G 1 -g 1 -b 20 -l 10 -s 0.0 -aL 0.0 -aS 0.0 -T 4 -M
         32000

Started: Mon Aug 31 21:55:20 2020
================================================================
                            Output
----------------------------------------------------------------
```

```
      total seq: 13882
      longest and shortest : 2513 and 99
      Total letters: 4952779
      Sequences have been sorted

      ...


      13882  finished       9907  clusters
```

The output 1598936109.fas.1.clstr.sorted contains the sorted cluster groups generated by CD-Hit and indicates the longest representative transcript for each cluster.

2.1.5 Reading CD-HIT output in R

dplyr was used to manipulate the CD-Hit output to create a list of unique transcript IDs of the longest representative transcript for each cluster. This list file is required in the following step to subset from the full-length transcript file which contains the untranslated regions.

Load libraries and set the working directory

Import 1598936109.fas.1.clstr.sorted, the CD-Hit output file with all transcript IDs assigned to a sorted cluster (note: ./ReferenceClusters/ has been continually included due to setwd errors).

```
clstr <- read.csv("./ReferenceClusters/1598936109.fas.1.clstr.sorted",
    sep = "\t", row.names = NULL, header = FALSE, stringsAsFactors = FALSE)
# head(clstr)
```

Initial file had a blank line with ">Cluster #" in between new clusters. Extend the >Cluster # down the column so that all transcripts are directly associated with a cluster number in their row.

```
clstr2 <- clstr
n = nrow(clstr)
x = 0
numbers_only <- function(x) !grepl("\\D", x)
for (row in c(1:n)) {
    if (numbers_only(clstr2[row, 1]) == TRUE) {
        clstr2[row, 1] <- x
    } else {
        NULL
    }
    x <- clstr2[row, 1]
}
# head(clstr2,20)
```

Count the number of transcript members for each cluster.

```
clstr.sums <- data.frame(dplyr::count(clstr2, V1))
write.csv(clstr.sums, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/clstr.sums.csv",
    quote = FALSE, row.names = FALSE, col.names = FALSE)
head(clstr.sums)
```

```
##                V1  n
## 1    >Cluster 0 19
## 2    >Cluster 1 16
## 3   >Cluster 10 12
## 4  >Cluster 100  6
## 5 >Cluster 1000  3
## 6 >Cluster 1001  3
```

Remove the additional blank row without a transcript name from between each cluster group.

```
switch <- clstr.sums[1, 2]
clstr3 <- cbind(clstr2[1], clstr)
clstr3[c((switch - 5):(switch + 5)), ]
```

```
clstr4 <- clstr2[-which(clstr2$V2 == ""), ]
clstr4[c(1:5, (switch - 5):(switch + 5)), ]
```

Separate CD-Hit information into columns by delimiters:

- Remove > symbol from transcript names
- Move number of amino acids into a new column by separating “aa”
- Move statistics of % match into a new column byt separating “…”
- Give the columns headers

```
clstr5 <- clstr4
clstr5[] <- lapply(clstr5, gsub, pattern = ">", replacement = "")
clstr5.2 <- data.frame(str_split_fixed(clstr5$V2, "aa, ", 2))
clstr5.3 <- data.frame(str_split_fixed(clstr5.2$X2, "... ", 2))
clstr6 <- cbind(clstr5[1], clstr5.2[1], clstr5.3[1:2])
colnames(clstr6) <- c("cluster", "aa", "TranscriptID", "stat")
# head(clstr6)
```

Filter based on rows that have a “*” in the stat column. These indicate the representative sequences for each cluster determined by CD-Hit and will act as the main reference.

```
clstr7 <- filter(clstr6, stat == "*")
head(clstr7, 50)
```

Pull out the column “TranscriptID” and export into a txt file so it can be used as a list to extract these sequences.

```
clstrTranscriptID <- pull(clstr7, var = TranscriptID)

write.csv(clstrTranscriptID, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/clstrTranscriptID.csv",
    quote = FALSE, row.names = FALSE, col.names = FALSE)
head(clstrTranscriptID)
```

```
## [1] PB.6088.11|e88f16|path1:164-2792(+)|transcript/22345|m.11362
## [2] PB.2919.1|6c1fb9|path4:13-2574(+)|transcript/10624|m.5342
```

```
## [3] PB.535.2|136fa8|path0:1-2775(+)|transcript/4312|m.913
## [4] PB.4689.2|afa3de|path19:22-2096(+)|transcript/11491|m.8666
## [5] PB.6357.1|f27f14|path2:4-2108(+)|transcript/12511|m.11846
## [6] PB.715.4|1b393a|path6:6-8031(+)|transcript/8|m.1216
## 13882 Levels: PB.10.1|004815|path1:5-1713(+)|transcript/17534|m.3 ...
```

Pull out the first segment before the first | from column "TranscriptID" and export into a txt file. this reduced the ID to the initial PB.<loci_index>.<isoform_index> identifier.

```
WholeTranscriptID <- word(clstrTranscriptID, 1, sep = "\\|")
write.csv(WholeTranscriptID, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/WholeTranscriptID.csv",
    quote = FALSE, row.names = FALSE, col.names = FALSE)
head(WholeTranscriptID)
```

```
## [1] "PB.6088.11" "PB.2919.1"  "PB.535.2"   "PB.4689.2"  "PB.6357.1"
## [6] "PB.715.4"
```

Note there will be "duplicates" if the name is truncated all the way back to the PB.<loci_index>.<isoform_index> identifier due to multiple isoforms that are retained. A longer, but still truncated version of this ID (excluding the information appended by ANGEL), was required to subset from the hq.fasta.no5merge.collapsed.rep.fa file.

The clstrTranscriptID.csv is now the full list of representative transcripts based on clustering of the translated proteins in predicted read frame. This file was then used in Seqtk on the NECTAR cloud to extract a .fasta of all the reference transcripts representative of these putative gene clusters.

2.1.6 Seqtk

Seqtk was used on the NECTAR machine to manipulate the fasta files and subset based on list files of transcript names.

Installation:

```
#Carmel July 2020

#Used on NECTAR research cloud

#Relies on hard-coded paths & filenames
#run line by line in shell

----------------

cd /mnt/Prog
conda install -c bioconda seqtk

#___
The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    certifi-2020.6.20          |   py37hc8dfbb8_0         151 KB  conda-forge
    conda-4.8.3                |   py37hc8dfbb8_1         3.0 MB  conda-forge
    openssl-1.1.1g             |       h516909a_1         2.1 MB  conda-forge
    python_abi-3.7             |          1_cp37m           4 KB  conda-forge
    seqtk-1.3                  |       hed695b0_2          39 KB  bioconda
    ------------------------------------------------------------
                                           Total:         5.3 MB

The following NEW packages will be INSTALLED:

  python_abi         conda-forge/linux-64::python_abi-3.7-1_cp37m
  seqtk              bioconda/linux-64::seqtk-1.3-hed695b0_2
#___

Conda packages are located in /mnt/Prog/miniconda3/bin/ if you dont set a path
----------------
```

2.1.6.1 Subsetting fasta file of cluster representatives

Manually convert clstrTranscriptID.csv to an clstrTranscriptID.lst file.

Seqtk was used to subset the pygmy.ANGEL.cds fasta file based on the R cluster list outputs: Moved files into /mnt/IsoSeq-analysis/data/Seqtk

```
screen        #to be safe
cd /mnt/IsoSeq-analysis/data/Seqtk
/mnt/Prog/miniconda3/bin/seqtk subseq ../ANGEL/pygmy.ANGEL.cds
↪  clstrTranscriptID.lst > clstrTranscriptID.fa
```

check:

```
wc -l clstrTranscriptID.lst
# 9907
wc -l clstrTranscriptID.fa
# 19814

# 19814/2 = 9907. So it is correct
```

The hq.fasta.no5merge.collapsed.rep.fa file is a fasta file where sequence information ran over multiple "lines" within the file.

Seqtk was used to convert this file into a .fasta with sequence on a single line.

```
17/11/2020
# just converting hq.fasta.no5merge.collapsed.rep.fa into a fasta with sequence all
↪  on one line
# command seq is basic fasta/q conversion

cd /mnt/IsoSeq-analysis/data/Seqtk
/mnt/Prog/miniconda3/bin/seqtk seq
↪  ../Cogent/collected/hq.fasta.no5merge.collapsed.rep.fa >
↪  hq.fasta.no5merge.collapsed.rep.1L.fa
```

Checked that the file still has the correct number of sequences:

```
$ grep -c ">" hq.fasta.no5merge.collapsed.rep.fa
15729

grep -c ">" hq.fasta.no5merge.collapsed.rep.1L.fa
15729
```

2.1.6.2 Cluster and BLAST output exploration and visualisation using seqtk Exploration of long read cluster members

Clusters used above, before filtering by representative sequences:

```
head(clstr6)
```

```
##     cluster  aa                                               TranscriptID
## 2 Cluster 0 108    PB.6088.1|e88f16|path1:1-2798(+)|transcript/5100|m.11352
## 3 Cluster 0 108 PB.6088.3|e88f16|path1:161-2678(+)|transcript/16260|m.11354
## 4 Cluster 0 108 PB.6088.4|e88f16|path1:161-2798(+)|transcript/22174|m.11355
## 5 Cluster 0 108 PB.6088.13|e88f16|path1:164-2688(+)|transcript/6074|m.11357
## 6 Cluster 0 108 PB.6088.8|e88f16|path1:164-2220(+)|transcript/11457|m.11359
## 7 Cluster 0 108 PB.6088.9|e88f16|path1:164-2792(+)|transcript/13989|m.11360
##        stat
## 2 at 99.07%
## 3 at 99.07%
## 4 at 99.07%
## 5 at 99.07%
## 6 at 99.07%
## 7 at 99.07%
```

Reminder of header format for Reference transcripts TranscriptID

```
head reference_transcripts.1Lv.clean.fasta

>PB.2.1|002537|path0:1-1624(+)|transcript/18304 transcript/18304
↪  full_length_coverage=2;length=1627;num_subreads=26

↪  GACCACGCGACCGTTTGGATTTCCTTTTTCCGCTGCCCTCTTGATGTCCTCCTGAGGCCATTCTCTGCCTGCGCGGTGGTGATACGTGGGGTC
>PB.3.2|00361c|path0:43-2240(+)|transcript/10564 transcript/10564
↪  full_length_coverage=2;length=2200;num_subreads=27

↪  GGGAGTTGCGTGGTGACGTCCGTTTCTATATGAGCGCGAGCGGCGGGAGCCCCGGTCTTTCTCCCCGCCAGCCAGCCAGCCGCGCGGCGCTCC
```

Note the headers cut off before the |m### added by ANGEL which are present in the Cluster file Identifiers. All sequences are on one line

Example clusters Transcript IDs were extracted manually into text files names by Cluster number ID This was completed using Seqtk at the same time as BLASTn outputs were extracted from the *de novo* assembly.

Clusters 0, 4 and 5 are discussed as examples; cluster 0 having one of the longer canonical transcripts and being a larger cluster, Remembering that Isoseq output non-redundant transcripts with putative labels "PB.<Transcript#>.<Isoform#> Cluster 4 is visualised as an example of a cluster which has included multiple"Transcript numbers", and Cluster 5 is an example of a cluster with many transcripts with the same "transcript number" (though not all) but different "isoform numbers"

As expected, as these clusters were generated using the coding sequence file and clustered based on protein translation, the sequences in this cluster are near identical in central regions which represent the predicted coding region of ech transcript. Much of the differences occur at the the 5' and 3' ends and in the untranlsated regions.

2.1.7 Visualisation of clusters using Seqtk and MEGAX

```
#!/bin/bash
# ### (SLURM submission truncated for brevity) Important parameters are kept:

module load Miniconda3/4.9.2

#################################################################
# Enter the command-line arguments that you job needs to run.

source ~/.bashrc
conda activate SeqConda

#Script input:

# This script is used to sort BLASTn outputs and to use Seqtk to extract example
↪  clusters of transcripts which had a best match to the same target sequence, as
↪  well as extract example Clustered isoforms from the full reference database for
↪  visualisation of transcript clusters and alignments.
# The paths in the script assume that a specific directory structure has been set
↪  up.
#y
# Modules required:
```

```
# module load Miniconda3/4.9.2

# usage through slurm
#
# Carmel Maher
# Aug 2023

#------------------------------------
#Run from within BwaConda environment

#module load Miniconda3/4.9.2
#conda create --name SeqConda
#conda activate SeqConda (this is input in every SLURM script)

#Required Installations:
#conda install -c bioconda seqtk
# seqtk-1.4

#------------------------------------
function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}


#Create output folder
cd /scratch/user/mahe0050/DE-analysis/2_alignedData
#mkdir ./seqtk || error_exit "$LINENO: Directory Error 0"
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtk

# lists of sequence IDs for clusters and target sequences were created manually in
↪  notepad:


#Extract sequences with names in file name.lst, one sequence name per line: seqtk
↪  subseq in.fq name.lst > out.fq


#ID files to extract were placeed in
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtk

# Files to sub-set
#
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/hq.fasta.no5merge.collapsed.rep.1L.fa
#
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.1L.Trinity.fas
↪  #as created below
```

```
#Cluster List files:
# /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtkBLAST
# /scratch/user/mahe0050/IsoSeq-analysis/data/seqtkClusters/Clstr0.lst
# /scratch/user/mahe0050/IsoSeq-analysis/data/seqtkClusters/Clstr0.4st
# /scratch/user/mahe0050/IsoSeq-analysis/data/seqtkClusters/Clstr0.5st


# ----------
# To subset Cluster examples

cd /scratch/user/mahe0050/IsoSeq-analysis/data/seqtkClusters

#List files:
seqtk subseq
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/hq.fasta.no5merge.collapsed.rep.1L.fa
↪  Clstr0.lst > Clstr0.fa || error_exit "$LINENO: Cluster 0 error"

seqtk subseq
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/hq.fasta.no5merge.collapsed.rep.1L.fa
↪  Clstr4.lst > Clstr4.fa || error_exit "$LINENO: Cluster 4 error"

seqtk subseq
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/hq.fasta.no5merge.collapsed.rep.1L.fa
↪  Clstr5.lst > Clstr5.fa || error_exit "$LINENO: Cluster 5 error"

echo "clusters subseq done"

echo "job complete"
```

BLASTx output by gene

Seqtk was also used to search for transcripts which returned BLAST results for some genes of interest as data exploration. The reference list files for this were extracted from the BLASTx output were created using find and sort functions in notepad++ and excel, and then applied to the pygmy.ANGEL.cds fasta file to extract sequences.

```
#Usage from /mnt/IsoSeq-analysis/data/ANGEL

cd /mnt/IsoSeq-analysis/data/Seqtk
/mnt/Prog/miniconda3/bin/seqtk subseq ../ANGEL/pygmy.ANGEL.cds AQP.lst > AQP.fa

/mnt/Prog/miniconda3/bin/seqtk subseq ../ANGEL/pygmy.ANGEL.cds HSP.lst > HSP.fa

/mnt/Prog/miniconda3/bin/seqtk subseq ../ANGEL/pygmy.ANGEL.cds SLC.lst > SLC.fa

/mnt/Prog/miniconda3/bin/seqtk subseq ../ANGEL/pygmy.ANGEL.cds MAPK.lst > MAPK.fa
```

2.1.8 Subsetting Full-length Transcripts Which Represent each Putative Gene

Creation of a subset of the longest representative transcripts for each cluster as per CD-HIT and the previous two to form the reference for counts in the following Chapter 4.

As only the peptide coding sequence of transcripts was used in the BLASTx search and the CD-HIT clustering in order to cluster similarly expressed transcripts, the untranslated regions of transcripts are absent. Our goal is to include these untranslated regions in the reference to be used for gene expression counts.
Rather than attempting to piece pygmy.ANGEL.cds and pygmy.ANGEL.utr back together, a truncated version of the transcript names was used to pull the full length unaltered transcript out of the earlier hq.fasta.no5merge.collapsed.rep.fa file.

Initial checks of clstrTranscriptID.csv were carried out. Files are continually checked for transcript ID or sequence counts to ensure manual manipulation did not remove any data that was not being targeted. Exploration of discrepancies below identified a number of transcripts that with clustering of multiple isoforms, have seeded more than one cluster. This means that there were duplicates in the list file once coding region identifiers were removed and sequences had to be extracted manually and not by applying the clstrTranscriptID list to hq.fasta.no5merge.collapsed.rep.1L.fa in seqtk.

The following was done on a local Intel core i7 machine using RStudio or in-house bash scripts applied in Git Bash for Windows. Some later steps were carried out on the NECTAR allocation.

All files copied to "~/IsoSeq-analysis/data/Seqtk/reference COPIES 2020-11" and contents checked

The following file contains all of the below:

```
Carmel@CarmelASUS MINGW64 ~
cd "E:/@DATA/@@PBT_KidneyIsoSeq_Working
 ↪  Data_C.Maher/IsoSeq-analysis/data/Seqtk/reference COPIES 2020-11"

2020-11-17

cd "C:\Users\Carmel\Desktop\reference COPIES 2020-11"

-----

19814 clstrTranscriptID.fa                    /2 = 9907
9907 clstrTranscriptID.lst
```

```
hq.fasta.no5merge.collapsed.rep.fa
hq.fasta.no5merge.collapsed.rep.1L.fa
pygmy.ANGEL.cds
pygmy.ANGEL.pep
pygmy.ANGEL.utr
```

Sort and filter clstrTranscriptID.lst by unique values to make sure there are no dupicates

```
sort clstrTranscriptID.lst | uniq > clstrUniq.txt
wc -l clstrUniq.txt
9907 clstrUniq.txt
```

List duplicates if there are any found (there were not)

```
sort clstrTranscriptID.lst | uniq -d > clstrUniqD.txt
wc -l clstrUniqD.txt
0 clstrUniqD.txt
```

```
Carmel@CarmelASUS MINGW64 /e/@DATA/@@PBT_KidneyIsoSeq_Working
↪  Data_C.Maher/IsoSeq-analysis/data/Seqtk/reference COPIES 2020-11
```

---> 1. Isoseq HQ output file: hq.fasta.no5merge.collapsed.rep.fa

header format

```
head -n 2 hq.fasta.no5merge.collapsed.rep.fa
>PB.1.1|000643|path0:1-1879(+)|transcript/14742 transcript/14742
↪  full_length_coverage=3;length=1887;num_subreads=52
ATAGAGTACAAGTACTAGAGTTTTAGTGTGGTCGAAGAGAAATCCAAGATCCACCATGGG
```

no of sequences

```
grep -c ">" hq.fasta.no5merge.collapsed.rep.fa
15729
```

---> 2. Angel CDS output from Isoseq HQ file: pygmy.ANGEL.cds

header format

```
head -n 2 pygmy.ANGEL.cds
>PB.2.1|002537|path0:1-1624(+)|transcript/18304|m.1 type:likely-NA len:135 strand:+
 ↪  pos:282-686

 ↪  ATGTCCTACTGCAGGCAAGAAGGAAAGGACAGAATTATATTTGTGACAAAAGAGGACCATGAAACTCCAAGTAGTGCTGAGTTAGTAGCGGAT
```

Note Sequence is on a single line

no of sequences

```
grep -c ">" pygmy.ANGEL.cds
13882
```

---> Angel CDS file clusterd with CD-HIT: 1598936109.fas.1.clstr

header format

```
cd cd-hit\ 99
head 1598936109.fas.1.clstr
>Cluster 0
0       2513aa, >PB.715.4|1b393a|path6:6-8031(+)|transcript/8|m.1216... *
1       2250aa, >PB.715.5|1b393a|path6:278-8030(+)|transcript/13|m.1217... at
 ↪  99.56%
2       2369aa, >PB.715.6|1b393a|path6:524-8029(+)|transcript/20|m.1218... at
 ↪  99.58%
3       2148aa, >PB.715.7|1b393a|path6:1187-8030(+)|transcript/49|m.1219... at
 ↪  99.53%
4       1913aa, >PB.715.9|1b393a|path6:1893-8028(+)|transcript/118|m.1221... at
 ↪  99.79%
5       1688aa, >PB.715.11|1b393a|path6:2596-8054(+)|transcript/279|m.1223... at
 ↪  99.59%
6       1452aa, >PB.715.14|1b393a|path6:3304-8029(+)|transcript/591|m.1226... at
 ↪  100.00%
7       1333aa, >PB.715.15|1b393a|path6:3629-8031(+)|transcript/935|m.1227... at
 ↪  99.77%
8       1314aa, >PB.715.16|1b393a|path6:3916-8078(+)|transcript/962|m.1228... at
 ↪  100.00%
```

File contains truncated headers from the Angel CDS file, clustered acoording to similarity. Truncation appears to be at the first whitespace no of sequences

```
grep -c ', >' 1598936109.fas.1.clstr     #search pattern changed to avoid the ">"
 ↪  in front of each cluster label eg >Cluster 0
13882
```

Sequence number matches Angel CDS file

no of clusters

```
grep -c '>C' 1598936109.fas.1.clstr
9907
```

```
grep -c '>C' 1598936109.fas.1.clstr.sorted
9907
```

both match

```
tail 1598936109.fas.1.clstr
>Cluster 9902
0       99aa, >PB.8346.1|transcript/22300:1-1143(+)|transcript/22300|m.14112... *
>Cluster 9903
0       99aa, >PB.8372.1|transcript/22508:1-1103(+)|transcript/22508|m.14137... *
>Cluster 9904
0       99aa, >PB.8489.1|transcript/23315:1-855(+)|transcript/23315|m.14244... *
>Cluster 9905
0       99aa, >PB.8706.1|transcript/25054:1-372(+)|transcript/25054|m.14469... *
>Cluster 9906
0       99aa, >PB.9028.1|transcript/4242:1-2955(+)|transcript/4242|m.14776... *
```

```
tail 1598936109.fas.1.clstr.sorted
>Cluster 9902
0       99aa, >PB.8346.1|transcript/22300:1-1143(+)|transcript/22300|m.14112... *
>Cluster 9903
0       99aa, >PB.8372.1|transcript/22508:1-1103(+)|transcript/22508|m.14137... *
>Cluster 9904
0       99aa, >PB.8489.1|transcript/23315:1-855(+)|transcript/23315|m.14244... *
>Cluster 9905
0       99aa, >PB.8706.1|transcript/25054:1-372(+)|transcript/25054|m.14469... *
>Cluster 9906
0       99aa, >PB.9028.1|transcript/4242:1-2955(+)|transcript/4242|m.14776... *
```

Cluster numbers also match both files:Cluster 9906 + Cluster 0 = 9907

Check if the header changed from Isoseq HQ after processing with Angel?

```
cd ../
grep '>PB.715.4|1b393a|path6:6-8031(+)|transcript/8'
 ↪  hq.fasta.no5merge.collapsed.rep.fa
>PB.715.4|1b393a|path6:6-8031(+)|transcript/8 transcript/8
 ↪  full_length_coverage=43;length=7837;num_subreads=60
```

note: the “|m.xxxxx” suffix in the ANGEL output file is added at the first whitespace

---> 3. The longest seed sequence from the CD-HIT output: clstrUniq.txt

header format

```
head clstrUniq.txt
PB.10.1|004815|path1:5-1713(+)|transcript/17534|m.3
PB.100.1|049515|path5:1-1563(+)|transcript/19243|m.132
PB.1000.1|26d4b2|path0:1-2889(+)|transcript/4521|m.1797
PB.1000.2|26d4b2|path0:1287-2889(+)|transcript/18232|m.1798
PB.1002.1|26ea71|path2:1-2673(+)|transcript/6271|m.1799
PB.1002.2|26ea71|path2:11-2691(+)|transcript/8089|m.1800
PB.1003.1|26fc93|path2:1-3987(+)|transcript/1197|m.1801
PB.1005.1|2718e2|path0:1-5294(+)|transcript/258|m.1804
PB.1006.1|272202|path0:1-2713(+)|transcript/5814|m.1805
PB.1007.1|272202|path1:1-3370(+)|transcript/2554|m.1806
```

Prefix ">" and suffix "… *" have been removed in R Studio

no of strings

```
grep -c 'PB.' clstrUniq.txt
9907
```

matches number of clusters above

strip the "|m.xxxx" from the strings in clstrUniq.txt

```
sed 's/|m.*//' clstrUniq.txt > new_cluster.txt
```

check

```
head new_cluster.txt
PB.10.1|004815|path1:5-1713(+)|transcript/17534
PB.100.1|049515|path5:1-1563(+)|transcript/19243
PB.1000.1|26d4b2|path0:1-2889(+)|transcript/4521
PB.1000.2|26d4b2|path0:1287-2889(+)|transcript/18232
PB.1002.1|26ea71|path2:1-2673(+)|transcript/6271
PB.1002.2|26ea71|path2:11-2691(+)|transcript/8089
PB.1003.1|26fc93|path2:1-3987(+)|transcript/1197
PB.1005.1|2718e2|path0:1-5294(+)|transcript/258
PB.1006.1|272202|path0:1-2713(+)|transcript/5814
PB.1007.1|272202|path1:1-3370(+)|transcript/2554
```

```
grep -c 'PB.' new_cluster.txt
9907
```

How many of thes strings are in the original ONE LINE isoseq file?

```
grep -c -Ff new_cluster.txt hq.fasta.no5merge.collapsed.rep.1L.fa
9813
```

94 strings are missing

---> 4. find the missing strings

convert the isoform hq file into the same format as new_cluster.txt

```
grep '>' hq.fasta.no5merge.collapsed.rep.1L.fa| sed s'/ tra.*//' | sed s'/>//' >
↪  hq_headers
head hq_headers
PB.1.1|000643|path0:1-1879(+)|transcript/14742
PB.2.1|002537|path0:1-1624(+)|transcript/18304
PB.3.2|00361c|path0:43-2240(+)|transcript/10564
PB.4.1|004079|path1:6-745(+)|transcript/23781
PB.4.2|004079|path1:6-612(+)|transcript/24311
PB.5.1|004079|path2:209-642(+)|transcript/24981
PB.6.1|004079|path3:273-616(+)|transcript/25193
PB.7.1|004079|path5:6-664(+)|transcript/24242
PB.8.1|004079|path6:1-607(+)|transcript/24477
PB.9.1|004079|path7:1-725(+)|transcript/23873
```

```
cat hq_headers |wc -l
15729
```

Checking for the same number of matches

```
grep -Ff new_cluster.txt hq_headers > matching_strings
cat matching_strings |wc -l
9813
```

Checking for duplicates

```
cat hq_headers | sort |uniq -cd
0

cat new_cluster.txt | sort |uniq -cd
      2 PB.1206.1|2e8181|path5:1-4617(+)|transcript/551
      2 PB.1261.1|30ba06|path3:633-1922(+)|transcript/21324
      2 PB.1570.2|3bd5a8|path2:20-1934(+)|transcript/13977
      2 PB.1719.1|404d72|path0:1-3429(+)|transcript/2407
      2 PB.1764.3|429969|path8:44-2062(+)|transcript/12739
      2 PB.1813.1|446aed|path0:1-2871(+)|transcript/4320
      2 PB.1865.1|46bd4e|path0:1-2536(+)|transcript/6726
      2 PB.1869.1|46d663|path2:1-1320(+)|transcript/21375
      2 PB.220.1|08884f|path0:1-4120(+)|transcript/991
      2 PB.2360.1|5843a6|path32:1-2062(+)|transcript/11326
      2 PB.2541.3|5e93d1|path0:5-2495(+)|transcript/7381
      2 PB.2560.1|5f09e2|path0:1-1196(+)|transcript/21957
      2 PB.2618.1|612418|path1:1-2013(+)|transcript/12200
      2 PB.2713.3|65abaa|path0:37-2495(+)|transcript/7693
      2 PB.2892.1|6b359a|path12:1-4597(+)|transcript/526
      2 PB.3037.2|702e9b|path5:2-3018(+)|transcript/3923
      2 PB.3243.1|784f79|path0:1-2683(+)|transcript/5903
      2 PB.3259.1|792211|path3:1-5233(+)|transcript/268
      2 PB.3286.3|7a2645|path3:103-3087(+)|transcript/3780
      2 PB.3405.1|7f2f96|path0:5-2081(+)|transcript/11903
      3 PB.3430.1|8055ae|path8:1-3589(+)|transcript/2043
      2 PB.3461.1|81f072|path5:1-2492(+)|transcript/7285
```

```
2 PB.351.4|0cccb8|path5:27-4293(+)|transcript/775
2 PB.3562.1|864d1b|path0:1-3492(+)|transcript/2174
2 PB.3615.1|883e51|path3:1-4044(+)|transcript/1347
2 PB.3836.3|8ff0e9|path0:4-3972(+)|transcript/946
2 PB.3934.1|93b547|path0:1-3353(+)|transcript/2676
2 PB.4042.2|97fba8|path2:34-2096(+)|transcript/13603
2 PB.4219.2|9d8e09|path3:9-3566(+)|transcript/2045
2 PB.4442.1|a6620e|path0:1-2906(+)|transcript/4555
2 PB.4458.1|a6e2a5|path0:1-3673(+)|transcript/1739
2 PB.4822.1|b4950f|path1:1-3315(+)|transcript/2177
2 PB.4822.2|b4950f|path1:1240-3315(+)|transcript/10982
2 PB.4919.1|b7da5b|path0:1-3327(+)|transcript/2652
2 PB.4923.1|b7fed9|path0:1-2394(+)|transcript/8643
2 PB.5080.1|bde542|path3:1-3060(+)|transcript/3777
2 PB.5096.2|be8c26|path4:70-2754(+)|transcript/5782
2 PB.5149.1|c15864|path0:1-1628(+)|transcript/17934
2 PB.5308.1|c8b5b5|path2:35-3805(+)|transcript/1487
2 PB.5395.1|cbd93f|path0:1-1699(+)|transcript/17107
2 PB.5483.2|cf1828|path1:23-2344(+)|transcript/9126
2 PB.5619.1|d4d2ea|path0:1-2677(+)|transcript/6673
2 PB.5663.1|d5e71b|path0:1-5101(+)|transcript/303
2 PB.5939.1|e23f0e|path0:1-2960(+)|transcript/3907
2 PB.5939.2|e23f0e|path0:5-4542(+)|transcript/622
2 PB.613.1|16d05c|path0:1-1558(+)|transcript/18673
2 PB.6133.1|ea032e|path9:1-2052(+)|transcript/12270
2 PB.6276.1|ee992b|path0:1-1487(+)|transcript/19762
2 PB.6283.1|eee48e|path1:1-3454(+)|transcript/2331
2 PB.6286.1|ef0b10|path12:1-2744(+)|transcript/5222
2 PB.6303.1|efc514|path0:1-3222(+)|transcript/2889
2 PB.6378.3|f31894|path9:80-2754(+)|transcript/5975
2 PB.6385.1|f32c31|path0:1-1500(+)|transcript/12840
2 PB.6426.1|f4ded6|path0:1-1796(+)|transcript/16785
2 PB.6576.2|facceb|path2:1-3314(+)|transcript/3157
2 PB.659.1|19437d|path1:1-3483(+)|transcript/2210
2 PB.6763.1|transcript/10186:1-2230(+)|transcript/10186
2 PB.6877.1|transcript/1099:1-4113(+)|transcript/1099
2 PB.6966.1|transcript/11636:1-2103(+)|transcript/11636
2 PB.713.1|1b043a|path0:1-965(+)|transcript/23154
2 PB.7227.1|transcript/138:1-5869(+)|transcript/138
2 PB.7287.1|transcript/14227:1-1898(+)|transcript/14227
3 PB.7342.1|transcript/14710:1-1879(+)|transcript/14710
2 PB.7349.1|transcript/14730:1-1903(+)|transcript/14730
2 PB.7353.1|transcript/14772:1-1860(+)|transcript/14772
2 PB.74.1|02ebe4|path0:1-2606(+)|transcript/6537
2 PB.7501.1|transcript/1596:1-3759(+)|transcript/1596
2 PB.7528.1|transcript/16151:1-1797(+)|transcript/16151
2 PB.7678.1|transcript/1728:1-3672(+)|transcript/1728
2 PB.7707.1|transcript/17496:1-1683(+)|transcript/17496
2 PB.7868.1|transcript/188:1-5558(+)|transcript/188
2 PB.7973.1|transcript/19628:1-1476(+)|transcript/19628
2 PB.8082.1|transcript/20433:1-1412(+)|transcript/20433
2 PB.816.1|1e334e|path4:1-2040(+)|transcript/12316
2 PB.8325.1|transcript/22158:1-1191(+)|transcript/22158
```

```
      2 PB.834.2|1ec273|path2:1291-2750(+)|transcript/19945
      2 PB.8389.1|transcript/22625:1-1060(+)|transcript/22625
      2 PB.8709.1|transcript/252:1-5299(+)|transcript/252
      2 PB.8883.1|transcript/3393:1-3152(+)|transcript/3393
      2 PB.8886.1|transcript/3402:1-3146(+)|transcript/3402
      2 PB.8920.1|transcript/3588:1-3076(+)|transcript/3588
      2 PB.9068.1|transcript/4442:1-2918(+)|transcript/4442
      2 PB.9162.1|transcript/4983:1-2815(+)|transcript/4983
      2 PB.9209.1|transcript/5238:1-2771(+)|transcript/5238
      2 PB.924.1|22d3cf|path3:1-1494(+)|transcript/19025
      2 PB.9277.1|transcript/5650:1-2707(+)|transcript/5650
      2 PB.944.1|241290|path4:2-2629(+)|transcript/6557
      2 PB.9535.1|transcript/7343:1-2508(+)|transcript/7343
      2 PB.9581.1|transcript/7647:1-2463(+)|transcript/7647
      2 PB.9683.1|transcript/8355:1-2402(+)|transcript/8355
      2 PB.9766.1|transcript/9024:1-2324(+)|transcript/9024
      2 PB.9784.1|transcript/915:1-4197(+)|transcript/915
```

During the clustering it appears that several of the sequences have been made seed sequences (denoted with a *) for more than one cluster.

The counts above equate to how many strings which is the discrepancy noted above.

```
cat new_cluster.txt | sort |uniq -cd | cut -d' ' -f7 | awk '{sum+=$1-1}
↪  END{printf("%d\n",sum)}'
94
```

Both hq.fasta.no5merge.collapsed.rep.fa and hq.fasta.no5merge.collapsed.rep.1L.fa. were checked in this way but the single line file is required in later steps so only those commands have been included.

Final pulling out of transcripts sequences into a .fasta on a single line, also removes the skipped lines that grep outputs as '--'.
note: NOT using seqtk anymore:

```
grep -Ff new_cluster.txt -A 1 hq.fasta.no5merge.collapsed.rep.1L.fa | grep -v ^-- >
↪  reference_transcripts.1Lv.fasta

grep -c ">" reference_transcripts.1Lv.fasta
9813
```

---> **next step**

Persisting poly-a tails needed to be trimmed from the following file:

```
reference_transcripts.1Lv.fasta
```

Note: The Trim polya command in bbduk leaves many tails intact due to single non 'A' bases and does not facilitate hamming distance command at the same time, however specifying Ktrim=r literal or "trimming twice" (poly-a command and THEN ktrim) sequences become very short even with different K values.

Terry Bertozzi provided perl script as below to manually trim:

```
# Terry Bertozzi
#command line to remove poly A tails

#for use on /mnt/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.fasta

cd /mnt/IsoSeq-analysis/data/Seqtk

perl -pe 's/(A{5,}.{0,2})+$//gm' reference_transcripts.1L.fasta >
↪  reference_transcripts.1L.clean.fasta

perl -pe 's/(A{5,}.{0,2})+$//gm' reference_transcripts.1Lv.fasta >
↪  reference_transcripts.1Lv.clean.fasta

grep -c ">" reference_transcripts.1Lv.clean.fasta
9813

head reference_transcripts.1Lv.clean.fasta
#also looks good.
```

reference_transcripts.1Lv.clean.fasta - IS THE REFERENCE

2.2 *Short-read Data*

De-novo assembly of short-read Transcripts:
Short-read transcripts used for gene expression analysis in Chater 6 were initially intended for *de-novo* assembly and analysis due to a lack of closely related genomic references for *T. adelaidensis*.

This section outlines the methods for assembly of short-read datasets in the absence of a reference or genome for the purposes of comparing these final transcript datasets to the assembly generated from the long-reads as above.

Other notes: Initial plans for gene expression analysis did not include the availability of long-read sequencing to supplement a reference transcript list. Therefore, a pooled transcript assembly was created by concatenating reads from all eight samples in order to use this larger dataset as a pseudo-reference for other programs. This longer assembly is included in the BUSCO analysis below to illustrate the difference in assembly completeness and duplication.

Sequences were downloaded from the Australian Genomics Research Facility server & checksumms verified. This analysis was conducted on the Flinders University 'Deep Thought' HPC.

2.2.1 FASTQC Quality Assessment

Quality of sequencing output was quantified using the following script and FastQC v0.11.8 (note there are two commands because there were two sequencing run data folders AGRF_CAGRF13871_HCGYYBCXY and AGRF_CAGRF20022_CDNJBANXX).

```
!/bin/bash

# November 2019
# Carmel Maher
# This script is to submit a FastQC analysis on raw reads through Flinders'
↪  DeepThought HPC Slurm
# This script therefore assumes submission to Flinders Deepthought with an assumed
↪  directory structure and use of available modules


#-------------------------------------
# Required Modules:
# module add fastqc/0.11.8


#-------------------------------------
# Terry's error exit

function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}

#-------------------------------------
```

```
cd /scratch/user/mahe0050/DE-analysis/0_rawData/AGRF_CAGRF13871_HCGYYBCXY ||
↪  error_exit "$LINENO: directory error 1"
for file in *.fastq.gz
do
  FILESTEM=${file%.*}
  echo $FILESTEM

  fastqc $FILESTEM.gz --extract --outdir
  ↪  /scratch/user/mahe0050/DE-analysis/0_rawData/FastQC || error_exit "$LINENO:
  ↪  Error with FastQC at "$FILESTEM""
done

cd /scratch/user/mahe0050/DE-analysis/0_rawData/AGRF_CAGRF20022_CDNJBANXX ||
↪  error_exit "$LINENO: directory error 2"
for file in *.fastq.gz
do
  FILESTEM=${file%.*}
  echo $FILESTEM

  fastqc $FILESTEM.gz --extract --outdir
  ↪  /scratch/user/mahe0050/DE-analysis/0_rawData/FastQC || error_exit "$LINENO:
  ↪  Error with FastQC at "$FILESTEM""
done
```

The FastQC Reports were then combined for visualisation using NGSReports The full FastQC output summary .html file of all eight samples PRIOR to trimming is available here.

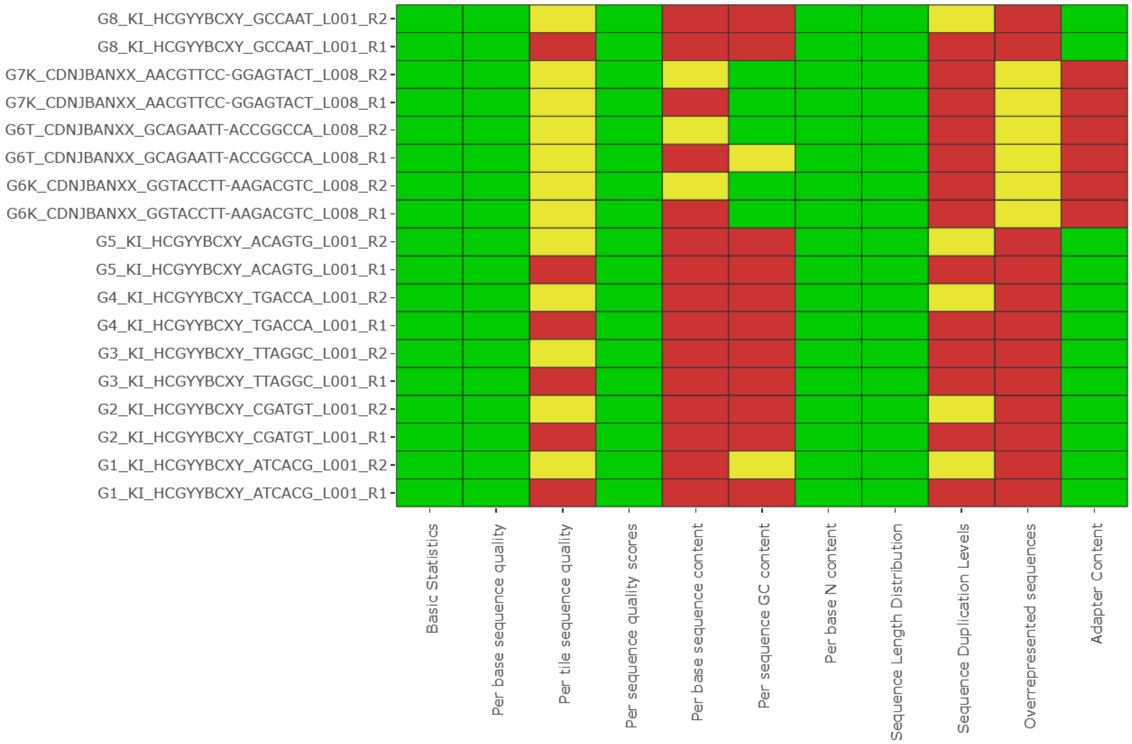


Figure 1: Summary of FastQC assessment of short-read samples prior to adapter removal or quality trimming. Colours indicate assessment of scores by FastQC for each overall category: Green=Pass, Yellow=Warning, Red=Fail. Poly(A) selected mRNA, isolated from kidney tissue from six *Tiliqua adelaidensis* skinks, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing.

Samples G6 and G7 were run on a second sequencing run, and have different overall lengths. This is the main source of variation in sequencing scores.

2.2.2 Adapter Trimming

Trimming to remove Illumina TruSeq adapters and SMARTer PCR tags as well as trimming based on quality scores was done using Cutadapt version 4.1,

This was set up on the Flinders Deep thought machine as below

```
module load Miniconda3/4.9.2
conda create -n ShortConda
source ~/.bashrc
conda activate ShortConda
conda install -c bioconda cutadapt
```

Primers were initially checked using a simple grep command

```
# ***
#Trimming Primers
# ***
```

```
# from within
cd /scratch/user/mahe0050/0_rawData:

### G6 and G7 files renamed from Prefix "G6k_" to "G6_KI_" to match the rest of the
↪  samples
head check zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | head -n 5


#Illumina primers check:
#~~~~~~~~~~~~~~~~~~~~~~~~

 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | grep
 ↪  "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA" -m10
#         =        (found on RHS)
 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | grep
 ↪  "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT" -m10
#         =        (none found)
 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R2.fastq.gz | grep
 ↪  "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA" -m10
#         =        (none found)
 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R2.fastq.gz | grep
 ↪  "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT" -m10
#         =        (found on RHS)


## Check of G6 as G6&7 were sequenced separately:
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R1.fastq.gz | grep
 ↪  "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA" -m10
#         =        (found on RHS)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R2.fastq.gz | grep
 ↪  "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT" -m10
#         =        (found on RHS)


#Clontech primers check (unknown bases removed for grep)
#~~~~~~~~~~~~~~~~~~~~~~~~

 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | grep "AAGCAGTGGTATCAACGCAGAGTAC"
 ↪  -m10
#         =        (found on LHS)
 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | grep
 ↪  "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGTACTCTGCGTTGATACCACTGCTT" -m10
#         =        (none found, found if the number of A is reduced)
zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | grep
↪  "AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT" -m10
#         =        (found on LHS)
 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R1.fastq.gz | grep "GTACTCTGCGTTGATACCACTGCTT"
 ↪  -m10
#         =        (found on RHS)

 zcat G1_KI_HCGYYBCXY_ATCACG_L001_R2.fastq.gz | grep "AAGCAGTGGTATCAACGCAGAGTAC"
 ↪  -m10
```

```
#          =        (found on LHS)
zcat G1_KI_HCGYYBCXY_ATCACG_L001_R2.fastq.gz | grep
↪  "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGTACTCTGCGTTGATACCACTGCTT" -m10
#          =        (none found, found if the number of A is reduced)
zcat G1_KI_HCGYYBCXY_ATCACG_L001_R2.fastq.gz | grep
↪  "AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT" -m10
#          =        (found on LHS)
zcat G1_KI_HCGYYBCXY_ATCACG_L001_R2.fastq.gz | grep "GTACTCTGCGTTGATACCACTGCTT"
↪  -m10
#          =        (found on RHS)

#note that in the absence of the longer string of unknown bases (or in the presence
↪  of poly-a tails) the forward and reverse primers have a sequence that is
↪  identical so there are instances where they are both found on the LHS of R1 and
↪  R2

## Check of G6 as G6&7 were prepared separately [not expected here]:
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R1.fastq.gz | grep
 ↪  "AAGCAGTGGTATCAACGCAGAGTAC" -m10
#          =        (none found)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R1.fastq.gz | grep
 ↪  "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGTACTCTGCGTTGATACCACTGCTT" -m10
#          =        (none found)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R1.fastq.gz | grep
 ↪  "AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT" -m10
#          =        (none found)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R1.fastq.gz | grep
 ↪  "GTACTCTGCGTTGATACCACTGCTT" -m10
#          =        (none found)

 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R2.fastq.gz | grep
 ↪  "AAGCAGTGGTATCAACGCAGAGTAC" -m10
#          =        (none found)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R2.fastq.gz | grep
 ↪  "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGTACTCTGCGTTGATACCACTGCTT" -m10
#          =        (none found)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R2.fastq.gz | grep
 ↪  "AAGCAGTGGTATCAACGCAGAGTACTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT" -m10
#          =        (none found)
 zcat G6_KI_CDNJBANXX_GGTACCTT-AAGACGTC_L008_R2.fastq.gz | grep
 ↪  "GTACTCTGCGTTGATACCACTGCTT" -m10
#          =        (none found)


# ***
# Proceed with script file removeAdadpters2.sh using SLURM queueing
# ***

cd /scratch/user/mahe0050/DE-analysis/bash

#test first
sbatch --test-only Slurm-Trim_DT-2022.sh
```

```
#run
# sbatch Slurm-Trim_DT-2022.sh        #a test run without quality trimming
sbatch Slurm-Trimq_DT-2022.sh         #run with quality trimming 5

#list all current jobs for a user
squeue -u mahe0050
```

Trimming was completed within a conda environment, using the following Slurm script to submit to Flinder's Deep thought machine, and to call the script removeAdapters2q.sh to remove primers and quality trim paired reads to a phredd 33 score of 5 (as below).

```
#!/bin/bash

# October 2022
# Carmel Maher
# This script is to submit a Cutadapt script to cut adapters and quality on raw
↪  reads through Flinders' DeepThought HPC Slurm
# This script therefore assumes submission to Flinders Deepthought with an assumed
↪  directory structure and use of available modules


#------------------------------------
#Run from within ShortConda environment
module load Miniconda3/4.9.2
conda activate ShortConda #(this is input in every SLURM script)

#Required Installations
#cutadapt (version 4.1)
#fastQC (v0.11.9)

#------------------------------------

# Illumina Truseq Primers:
# AGATCGGAAGAGCACACGTCTGAACTCCAGTCA (to remove from R1)
# AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT (to remove from R2)

# Clontech SMARTer Primers:
# SMARTer II A Oligonucleotide
# 5'- AAGCAGTGGTATCAACGCAGAGTACXXXXX -3' (X = undisclosed base in the proprietary
↪  SMARTer oligo sequence)
# 3' SMART CDS Primer II A
# 5’- AAGCAGTGGTATCAACGCAGAGTACT(30)N-1N -3’ (N = A, C, G, or T; N-1 = A, G, or C)
#
# FWDPRIMER   =   AAGCAGTGGTATCAACGCAGAGTACNNNNN
# RCFWDPRIMER     =   NNNNNGTACTCTGCGTTGATACCACTGCTT
# REVPRIMER   =   AAGCAGTGGTATCAACGCAGAGTACT
# RCREVPRIMER     =   AGTACTCTGCGTTGATACCACTGCTT


#------------------------------------
# Terry's error exit

function error_exit
```

```
{
  # Exit function due to fatal error
  # Accepts 1 arg:
  # string - descriptive error message

  echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
  exit 1
}

# in = /scratch/user/mahe0050/DE-analysis/0_rawData
# out = /scratch/user/mahe0050/DE-analysis/1_trimmedData

#-------------------------------------

#this script assumes you start IN the clean data folder as per DeepThought's
↪  paralell file system
#cd $BGFS


cd /scratch/user/mahe0050/DE-analysis/0_rawData || error_exit "$LINENO: directory
↪  error 1"


for file in *R1.fastq.gz
do
echo $file
FILESTEM=${file%_*}
echo $FILESTEM
FILESTEM2=`echo $file | cut -d "_" -f1,2 --output-delimiter="_"`
echo $FILESTEM2

#Check whether a FASTQ file is properly formatted
#    cutadapt -o /dev/null $file || error_exit "$LINENO: Fastq format error"

#Remove Illumina Truseq adapters from all paired data using the 'regular 3'  paired
↪  adapters' function (these adapters are on the RHS of both R1 and R2 reads in
↪  all 8 samples)
#Reads are also trimmed to 5 phred-33
cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT
↪  --cores=12 -q 5 -o ../1_trimmedData/q/$FILESTEM2"_R1_Icleanq.fq.gz" -p
↪  ../1_trimmedData/q/$FILESTEM2"_R2_Icleanq.fq.gz" $file $FILESTEM"_R2.fastq.gz"
↪  || error_exit "$LINENO: Error removing Illumina R1 adapters in $file"


#Then remove Clontech SMARTer adapters from all paired data using 'linked adapters'
↪  function for paired end reads (No filtering for "anchored" adapters at this
↪  step - see notes, also not expected in G6 or G7)

#as the length of target sequences is varied and unknown the possibility for the
↪  reverse completment of the second adapter is allowed by trimming as linked
↪  adapters.
#For linked adapters even though -a and -A are used (which indicate 3' adapters in
↪  the single line above) linked adapters assume the 5' is FIRST, and the 3' is
↪  SECOND.
```

```
#Therefore to trim FORWARD R1 reads, they are called as <forward primer in 5'-3'
↪  orientation> ... <reverse complement of reverse primer>
#to trim REVERSE R2 reads, they are called as <reverse primer in 5'-3' orientation>
↪  ... <reverse complement of forward primer>
↪  [https://cutadapt.readthedocs.io/en/stable/guide.html#linked-adapters]
#Reads shorter than 30bp are filtered out
cutadapt -a AAGCAGTGGTATCAACGCAGAGTACNNNNN...AGTACTCTGCGTTGATACCACTGCTT -A
↪  AAGCAGTGGTATCAACGCAGAGTACT...NNNNNGTACTCTGCGTTGATACCACTGCTT --cores=12
↪  --minimum-length 30 -o ../1_trimmedData/q/$FILESTEM2"_R1_cleanq.fq.gz" -p
↪  ../1_trimmedData/q/$FILESTEM2"_R2_cleanq.fq.gz"
↪  ../1_trimmedData/q/$FILESTEM2"_R1_Icleanq.fq.gz"
↪  ../1_trimmedData/q/$FILESTEM2"_R2_Icleanq.fq.gz" || error_exit "$LINENO: Error
↪  removing SMARTer adapters in $file"
done


#run cleaned data through fastqc
```

The trimmed files were than assessed again using FastQC version 0.11.9 submitted to Flinder's Deep Thought as below

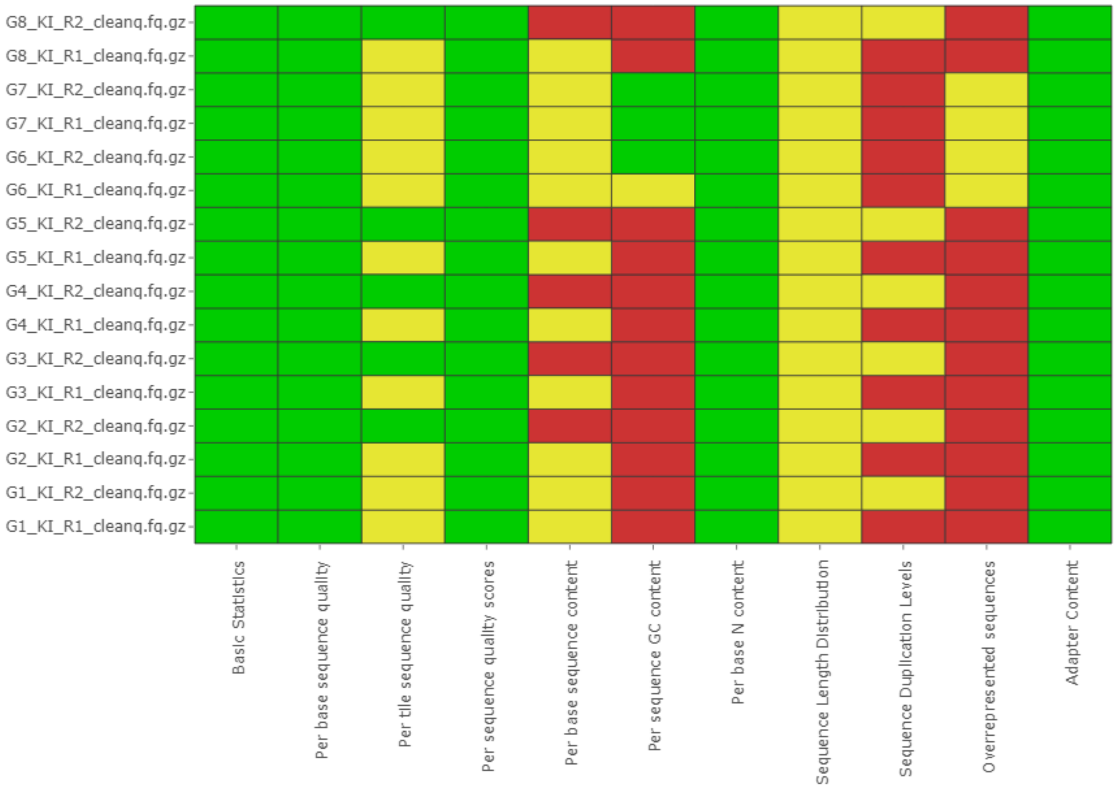
```
#FastQC installed on environment ShortConda does not work. This is an issue with
↪  the conda channels used to install
#error with fonts on report creation.
#Create a new environment calling jdk will make it install from ::conda-forge so
↪  that it works.

cd /scratch/user/mahe0050/DE-analysis/bash

conda create -n FQConda fastqc openjdk
source ~/.bashrc
conda activate FQConda

#File: FastQC-cl-run.bash
#FastQC generated for trimmed reads min length 30 with no quality trimming
# AND
#trimmed reads min length 30 with quality trimming 5 (phred -33)

# called in SLURM script:
sbatch Slurm-FQC_DT-2022.sh

# outputs summarised in NGSReports in R
```

To call the following script FastQC-cl-run.bash

```
!/bin/bash

# Oct 2022
# Carmel Maher
# This script is to submit a FastQC analysis on trimmed reads through Flinders'
↪  DeepThought HPC Slurm
# This script therefore assumes submission to Flinders Deepthought with an assumed
↪  directory structure and use of available modules
```

```
#------------------------------------
# Required Modules:

#Required Installations
#fastQC (v0.11.9)
#openjdk::conda-forge/linux-64::openjdk-17.0.3-h85293d2_2


#------------------------------------
# Terry's error exit

function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}

#------------------------------------

cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/q/ || error_exit "$LINENO:
↪   directory error 1"

for file in *R1_cleanq.fq.gz
do
  echo $file
  FILESTEM2=`echo $file | cut -d "_" -f1,2 --output-delimiter="_"`
  echo "FILESTEM2" $FILESTEM2

  fastqc --noextract --threads 8 -o ./FastQC_Cleanq ./$FILESTEM2"_R1_cleanq.fq.gz"
  ↪   ./$FILESTEM2"_R2_cleanq.fq.gz" || error_exit "$LINENO: Error with FastQC-q at
  ↪   "$FILESTEM2""
done


cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/ || error_exit "$LINENO:
↪   directory error 2"

for file in *R1_clean.fq.gz
do
  echo $file
  FILESTEM2=`echo $file | cut -d "_" -f1,2 --output-delimiter="_"`
  echo "FILESTEM2" $FILESTEM2

  fastqc --noextract --threads 8 -o ./FastQC_Clean ./$FILESTEM2"_R1_clean.fq.gz"
  ↪   ./$FILESTEM2"_R2_clean.fq.gz" || error_exit "$LINENO: Error with FastQC at
  ↪   "$FILESTEM2""
done
```

The output html file of all eight samples FastQC reports AFTER trimming is available here.



	Basic Statistics	Per base sequence quality	Per tile sequence quality	Per sequence quality scores	Per base sequence content	Per sequence GC content	Per base N content	Sequence Length Distribution	Sequence Duplication Levels	Overrepresented sequences	Adapter Content
G8_KI_R2_cleanq.fq.gz	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟨	🟨	🟥	🟩
G8_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟥	🟥	🟩
G7_KI_R2_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟩	🟩	🟨	🟥	🟨	🟩
G7_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟩	🟩	🟨	🟥	🟨	🟩
G6_KI_R2_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟩	🟩	🟨	🟥	🟨	🟩
G6_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟨	🟩	🟨	🟥	🟨	🟩
G5_KI_R2_cleanq.fq.gz	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟨	🟨	🟥	🟩
G5_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟥	🟥	🟩
G4_KI_R2_cleanq.fq.gz	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟨	🟨	🟥	🟩
G4_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟥	🟥	🟩
G3_KI_R2_cleanq.fq.gz	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟨	🟨	🟥	🟩
G3_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟥	🟥	🟩
G2_KI_R2_cleanq.fq.gz	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟨	🟨	🟥	🟩
G2_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟥	🟥	🟩
G1_KI_R2_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟨	🟥	🟩
G1_KI_R1_cleanq.fq.gz	🟩	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟥	🟥	🟩

Figure 2: Summary of FastQC assessment of short-read samples after adapter removal and quality trimming. Colours indicate assessment of scores by FastQC for each overall category: Green=Pass, Yellow=Warning, Red=Fail. Poly(A) selected mRNA, isolated from kidney tissue from six *Tiliqua adelaidensis* skinks, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing.

2.2.3 *De-novo* Assembly with TRINITY

De-novo assembly of transcripts.
Short-reads were assembled *de novo* using the program TRINITY v2.14.0. through the Singularity container available here

Sequencing run AGRF_CAGRF13871_HCGYYBCXY involved the synthesis of a non-stranded cDNA library prior to sequencing, while sequencing run AGRF_CAGRF20022_CDNJBANXX was completed with a different kit. All samples were treated as non-stranded for consistency.

Assembly was run on each sample individually as well as on a pooled (concatenated) file of all reads.

Individual samples were assembled using the following script on Flinders University's Deep Thought with a module of singularity v.3.6.3: Slurm-Trinity_DT, as below.

The full slurm submission script is shown here once. In following scripts the submission SLURM lines have been removed for brevity.

```
 #!/bin/bash
# Please note that you need to adapt this script to your job
# Submitting as is will fail cause the job to fail
# The keyword command for SLURM is #SBATCH --option
# Anything starting with a # is a comment and will be ignored
# ##SBATCH is a commented-out #SBATCH command
# SBATCH and sbatch are identical, SLURM is not case-sensitive
##################################################################
# Change FAN to your fan account name
# Change JOBNAME to what you want to call the job
# This is what is shows when attempting to Monitor / interrogate the job,
# So make sure it is something pertinent!
#
#SBATCH --job-name=mahe0050_Trinity
#
##################################################################
# If you want email updates form SLURM for your job.
# Change MYEMAIL to your email address
#SBATCH --mail-user=carmel.maher@flinders.edu.au
#SBATCH --mail-type=ALL
#
# Valid 'points of notification are':
# BEGIN, END, FAIL, REQUEUE.
# ALL means all of these
##################################################################
# Tell SLURM where to put the Job 'Output Log' text file.
# This will aid you in debugging crashed or stalled jobs.
# You can capture both Standard Error and Standard Out
# %j will append the 'Job ID' from SLURM.
# %x will append the 'Job Name' from SLURM
# %
#SBATCH --output=/home/mahe0050/%x-%j.out.txt
#SBATCH --error=/home/mahe0050/%x-%j.err.txt
##################################################################
# The default partition is 'general'.
# Valid partitions are general, gpu and melfu
##SBATCH --partition=PARTITIONNAME
#
```

```
##################################################################
# Tell SLURM how long your job should run for as a hard limit.
# My setting a shorter time limit, it is more likely that your
# job will be scheduled when attempting to backfill jobs.
#
# The current cluster-wide limit is 14 Days from Start of Execution.
# The timer is only active while your job runs, so if you suspend
# or pause the job, it will stop the timer.
#
# The command format is as follows: #SBATCH --time=DAYS-HOURS
# There are many ways to specify time, see the SchedMD Slurm
# manual pages for more.
#SBATCH --time=14-0
#
##################################################################
# How many tasks is your job going to run?
# Unless you are running something that is Parallel / Modular or
# pipelined, leave this as 1. Think of each task as a 'bucket of
# resources' that stand alone. Without MPI / IPC you can't talk to
# another bucket!
#
#SBATCH --ntasks=1
#
# If each task will need more that a single CPU, then alter this
# value. Remember, this is multiplicative, so if you ask for
# 4 Tasks and 4 CPU's per Task, you will be allocated 16 CPU's
#SBATCH --cpus-per-task=16
##################################################################
# Set the memory requirements for the job in MB. Your job will be
# allocated exclusive access to that amount of RAM. In the case it
# overuses that amount, Slurm will kill the job. The default value is
# around 2GB per CPU you ask for.
#
# Note that the lower the requested memory, the higher the
# chances to get scheduled to 'fill in the gaps' between other
# jobs. Pick ONE of the below options. They are Mutually Exclusive.
# You can ask for X Amount of RAM per CPU (MB by default).
# Slurm understands K/M/G/T For Kilo/Mega/Giga/Tera Bytes.
#
##SBATCH --mem-per-cpu=12G
# Or, you can ask for a 'total amount of RAM'. If you have multiple
# tasks and ask for a 'total amount' like below, then SLURM will
# split the total amount to each task evenly for you.
#SBATCH --mem=128G
##################################################################
# Change the number of GPU's required for you job. The most GPU's that can be
# requested is 2 per node. As there are limited GPU slots, they are heavily
# weighted against for Fairshare Score calculations.
# You can request either a 'gpu:telsa_v100:X' or a 'gpu:x'
#
# You can either request 0, or omit this line entirely if you
# a GPU is not needed.
#
```

```
#SBATCH --gres="gpu:0"
#################################################################
# Load any modules that are required. This is exactly the same as
# loading them manually, with a space-separated list, or you can
# write multiple lines.
# You will need to uncomment these.

module load singularity/3.6.3

#################################################################
# This example script assumes that you have already moved your
# dataset to /scratch as part of your HPC Pre-Job preparations.
# Its best to use the $TMP/$TMPDIR setup for you here
# to allow for the HPC to auto-clean anything you
# leave behind by accident.
# If you have a job-array and need a shared directory for
# data on /local, you will need to manually cleanup that
# directory as a part of your job script.

# Example using the SLURM $BGFS Variable (the Parallel Filesystem)

cd $BGFS
mkdir $BGFS/data/

cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/q

#################################################################
# Enter the command-line arguments that you job needs to run.


# Carmel Maher & Terry Bertozzi
# Sep 2017 - last edited oct 2022


function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}

#Slurm script places you in a temporary directory, this shall be treated as the
↪  same 'level' as /scratch/user/mahe0050/DE-analysis/

# go to the working directory - working from $BGFS/data
pwd

# *** #

# To run trinity separately on each sample
```

```
 # assemble transcripts from a single paired sample file
 #note trinity requires output directory with "trinity" in name

# ***G1***

 singularity exec --home $BGFS/data:/home -B
 ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
 ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
     --seqType fq \
     --left /data/G1_KI_R1_cleanq.fq.gz  \
     --right /data/G1_KI_R2_cleanq.fq.gz \
     --verbose --CPU 16 --max_memory 128G \
     --output G1_KI_trinity-sep || error_exit "$LINENO: Error running
     ↪  trinity-sep at G1_KI_"

echo "Trinity G1 complete"


# ***G2***

 singularity exec --home $BGFS/data:/home -B
 ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
 ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
     --seqType fq \
     --left /data/G2_KI_R1_cleanq.fq.gz  \
     --right /data/G2_KI_R2_cleanq.fq.gz \
     --verbose --CPU 16 --max_memory 128G \
     --output G2_KI_trinity-sep || error_exit "$LINENO: Error running
     ↪  trinity-sep at G2_KI_"

echo "Trinity G2 complete"


# ***G3***

 singularity exec --home $BGFS/data:/home -B
 ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
 ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
     --seqType fq \
     --left /data/G3_KI_R1_cleanq.fq.gz  \
     --right /data/G3_KI_R2_cleanq.fq.gz \
     --verbose --CPU 16 --max_memory 128G \
     --output G3_KI_trinity-sep || error_exit "$LINENO: Error running
     ↪  trinity-sep at G3_KI_"

echo "Trinity G3 complete"


# ***G4***

 singularity exec --home $BGFS/data:/home -B
 ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
 ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
```

```
          --seqType fq \
          --left /data/G4_KI_R1_cleanq.fq.gz  \
          --right /data/G4_KI_R2_cleanq.fq.gz \
          --verbose --CPU 16 --max_memory 128G \
          --output G4_KI_trinity-sep || error_exit "$LINENO: Error running
          ↪  trinity-sep at G4_KI_"

echo "Trinity G4 complete"


# ***G5***

  singularity exec --home $BGFS/data:/home -B
  ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
  ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
          --seqType fq \
          --left /data/G5_KI_R1_cleanq.fq.gz  \
          --right /data/G5_KI_R2_cleanq.fq.gz \
          --verbose --CPU 16 --max_memory 128G \
          --output G5_KI_trinity-sep || error_exit "$LINENO: Error running
          ↪  trinity-sep at G5_KI_"

echo "Trinity G5 complete"


# ***G6***

  singularity exec --home $BGFS/data:/home -B
  ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
  ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
          --seqType fq \
          --left /data/G6_KI_R1_cleanq.fq.gz  \
          --right /data/G6_KI_R2_cleanq.fq.gz \
          --verbose --CPU 16 --max_memory 128G \
          --output G6_KI_trinity-sep || error_exit "$LINENO: Error running
          ↪  trinity-sep at G6_KI_"

echo "Trinity G6 complete"


# ***G7***

  singularity exec --home $BGFS/data:/home -B
  ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
  ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
          --seqType fq \
          --left /data/G7_KI_R1_cleanq.fq.gz  \
          --right /data/G7_KI_R2_cleanq.fq.gz \
          --verbose --CPU 16 --max_memory 128G \
          --output G7_KI_trinity-sep || error_exit "$LINENO: Error running
          ↪  trinity-sep at G7_KI_"

echo "Trinity G7 complete"
```

```
# ***G8***

  singularity exec --home $BGFS/data:/home -B
  ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
  ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
          --seqType fq \
          --left /data/G8_KI_R1_cleanq.fq.gz  \
          --right /data/G8_KI_R2_cleanq.fq.gz \
          --verbose --CPU 16 --max_memory 128G \
          --output G8_KI_trinity-sep || error_exit "$LINENO: Error running
          ↪  trinity-sep at G8_KI_"

echo "Trinity G8 complete"


##############
# REMOVE all * , relative file positions, FILESTEM & other variables, they are not
↪  read within the container.
##############


##################################################################
# Once you job has finished its processing, copy back your results
# and ONLY the results to /scratch, then clean-up the temporary
# working directory
# This command assumes that the destination exists

mkdir /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-sep

cp -r /$BGFS/data /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-sep
```

All kidney samples were run through Trinity together using the same programs and parameters as above, in another script Slurm-TrinityAll_DT, detailed below.

Sample files to be concatenated and named appropriately

```
#!/bin/bash
# ### (SLURM submission truncated for brevity) Important parameters are kept:
#
#SBATCH --job-name=mahe0050_Trinity
#
#SBATCH --time=14-0
#
#SBATCH --ntasks=1
#
#SBATCH --cpus-per-task=16
#
##SBATCH --mem-per-cpu=12G
#SBATCH --mem=256G

##################################################################
```

```
# Load any modules that are required.

module load singularity/3.6.3

#################################################################

# using the SLURM $BGFS Variable (the Parallel Filesystem)

cd $BGFS
mkdir $BGFS/data2/

cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/q

#################################################################
# Enter the command-line arguments that you job needs to run.


# Carmel Maher & Terry Bertozzi
# Sep 2017 - last edited oct 2022


function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}

#Slurm script places you in a temporary directory, this shall be treated as the
↪  same 'level' as /scratch/user/mahe0050/DE-analysis/

# go to the working directory - working from $BGFS/data
pwd

# *** #


##############
# REMOVE all * , relative file positions, FILESTEM & other variables, they are not
↪  read within the container.
##############


# *** #

#To create one master kidney transcript set, sample files to be concatenated and
↪  named appropriately

cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/q
```

```
  #concatenate relevant sample files for one transcriptome - add singletons to R1
  ↪  as its not stranded anyway?
  cat ./*_R1_cleanq.fq.gz >> ./PBTKI_cat_R1.fq.gz || error_exit "$LINENO: Error
  ↪  concatenating R1"
  cat ./*_R2_cleanq.fq.gz >> ./PBTKI_cat_R2.fq.gz || error_exit "$LINENO: Error
  ↪  concatenating R2"

  singularity exec --home $BGFS/data2:/home -B
  ↪  /scratch/user/mahe0050/DE-analysis/1_trimmedData/q:/data -e
  ↪  /scratch/user/mahe0050/DE-analysis/trinityrnaseq.v2.14.0.simg  Trinity \
        --seqType fq \
        --left /data/PBTKI_cat_R1.fq.gz  \
        --right /data/PBTKI_cat_R2.fq.gz \
        --verbose --CPU 16 --max_memory 256G \
        --output trinity-all || error_exit "$LINENO: Error running Trinity-all"

echo "Trinity concatenated kidney samples complete"

#clean up concatenated files

rm -rf /scratch/user/mahe0050/DE-analysis/1_trimmedData/q/PBTKI_cat_R1.fq.gz ||
↪  error_exit "$LINENO: Error cleanup concatenated R1"
rm -rf /scratch/user/mahe0050/DE-analysis/1_trimmedData/q/PBTKI_cat_R2.fq.gz ||
↪  error_exit "$LINENO: Error cleanup concatenated R1"


##################################################################
# Once you job has finished its processing, copy back your results
# and ONLY the results to /scratch, then clean-up the temporary
# working directory
# This command assumes that the destination exists

mkdir /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all
cp -r /$BGFS/data2 /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all

# No need to cleanup $BGFS, SLURM handles the cleanup for you.
# Just dont forget to copy out your results, or you will lose them!

##################################################################
```

2.3 *Summary / Quantification*

2.3.1 Long-read Sequences Data Summary

15,729 total transcript isoforms retained in initial cleaning and clustering of redundant isoforms.

13,882 sequences in a predicted open reading frame.

9,907 clusters identified based on translated proteins of predicted open reading frame.

9,813 full length transcripts were subset into a reference file as the longest representatives of one or more transcript clusters.

2.3.2 Short-read Sequences Data Summary

Total Transcript count of each TRINITY generated dataset was simply conducted using shell commands. Total transcript count for long-read files are included in the BUSCO usage file below as headers required manipulation for input into BUSCO and totals were checked at this step.

Sample ID	Experimental factors	File name	Total transcripts/sequences assembled
G1	September Female	G1_KI_trinity-sep	48161
G2	September Female	G2_KI_trinity-sep	45075
G3	March/April Female	G3_KI_trinity-sep	37039
G4	March/April Male	G4_KI_trinity-sep	40893
G5	September Male	G5_KI_trinity-sep	38719
G6	September Male	G6_KI_trinity-sep	185868
G7	March/April Female	G7_KI_trinity-sep	229145
G8	March/April Female	G8_KI_trinity-sep	43621
All Samples combined	n/a	trinity-all	359197

3 Annotation

Completed on the long-read data above

- pygmy.ANGEL.cds
- pygmy.ANGEL.pep
- pygmy.ANGEL.utr

3.1 BLASTx

BLASTx searches were conducted on the dataset at this stage to include isoforms that are collapsed later on.

BLASTx was run on the pygmy.ANGEL.cds file as it contains the predicted open reading frame sequence of nucleotides for translateable proteins. This was completed on the NECTAR machine.

```
# Package: blast 2.9.0, build Mar 11 2019 15:20:05
# Uniprot sprot & trembl databases downloaded: 28/5/19     (note: trembl database
↪  not indexed or used)
# Anolis .pep fasta files downloaded 12/11/19
      Anolis_carolinensis.AnoCar2.0.pep.all.fa
      Anolis_carolinensis.AnoCar2.0.pep.abinitio.fa

# To format a database: #
# makeblastdb -in mydb.fsa -dbtype nucl -parse_seqids
```

UniProt Swiss-Prot database:

```
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/makeblastdb -in uniprot_sprot.fasta
↪  -parse_seqids -blastdb_version 5 -title "sprot" -dbtype prot -out sprot
```

Anolis carolinensis proteins database:

```
#AnoCar2.0.pep.all
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/makeblastdb -in
↪  Anolis_carolinensis.AnoCar2.0.pep.all.fa -parse_seqids -blastdb_version 5 -title
↪  "sprot" -dbtype prot -out AnoCar2.0.pep.all
```

Anolis Carolinensis database *ab initio*:

```
#AnoCar2.0.pep.abinitio
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/makeblastdb -in
↪  Anolis_carolinensis.AnoCar2.0.pep.abinitio.fa -parse_seqids -blastdb_version 5
↪  -title "sprot" -dbtype prot -out AnoCar2.0.pep.abinitio
```

Various searches were run in a screen to examine the effects of parameters and databases. Final searches are as listed below.

Searches included all databases listed above and variations of the parameters: -max_target_seqs 5 or -max_target_seqs 1 -max_hsps 5 or -max_hsps 1 -evalue 0.00001 or -evalue 1e-10

The NCBI Blast results most often referenced in the thesis are final parameters blastx -fmt6 -maxtarget1 -maxhsps1 -eval0.00001 against the UniProt Swiss-Prot database.

```
# uniprot sprot
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/blastx -db
↪  /mnt/Prog/blast/blastdb/sprot/sprot -query
↪  /mnt/IsoSeq-analysis/data/ANGEL/pygmy.ANGEL.cds -out
↪  pygmy.ANGEL_blastx_sprot-maxtarg1-maxhsp1.out -outfmt 6 -max_target_seqs 1
↪  -max_hsps 1 -evalue 0.00001 -num_threads 3
```

```
#AnoCar2.0.pep.all
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/blastx -db
↪  /mnt/Prog/blast/blastdb/AcarProt/AnoCar2.0.pep.all/AnoCar2.0.pep.all -query
↪  /mnt/IsoSeq-analysis/data/ANGEL/pygmy.ANGEL.cds -out
↪  pygmy.ANGEL_blastx_AnoCar.pep.all-maxtarg1-maxhsp1.out -outfmt 6
↪  -max_target_seqs 1 -max_hsps 1 -evalue 0.00001 -num_threads 3
```

All searches except the below were output in tab separated -outfmt 6. Due to import requirements an -outfmt 5 .xml file was required for results to be visualised in the program BLAST2GO (-max_target_seqs 5 -max_hsps 5 -evalue 0.00001).

```
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/blastx -db
↪  /mnt/Prog/blast/blastdb/AcarProt/AnoCar2.0.pep.all/AnoCar2.0.pep.all -query
↪  /mnt/IsoSeq-analysis/data/ANGEL/pygmy.ANGEL.cds -out
↪  pygmy.ANGEL_blastx_AnoCar.pep.all-BLAST2GO -outfmt 5 -max_target_seqs 5
↪  -max_hsps 5 -evalue 0.00001 -num_threads 4

/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/blastx -db
↪  /mnt/Prog/blast/blastdb/sprot/sprot -query
↪  /mnt/IsoSeq-analysis/data/ANGEL/pygmy.ANGEL.cds -out
↪  pygmy.ANGEL_blastx_sprot-BLAST2GO -outfmt 5 -max_target_seqs 5 -max_hsps 5
↪  -evalue 0.00001 -num_threads 4
```

The above and other trial searches are all listed in the file IsoSeq analysis post-ANGEL blast notes.sh

3.2 Gene Name IDs Assigned to Transcript ID

The ANGEL predicted open reading frame transcripts were searched against the UniProt Swiss-Prot database using BLASTx with parameters: -max_target_seqs 1 -max_hsps 1 -evalue 0.00001).
The BLASTx results are uploaded here.

Protein IDs from the BLASTx results were uploaded to the UniProt Retrieve/ID mapping tool and mapped to Gene names. This produced a list of unique UniProt Swiss-Prot identifiers and their corresponsing gene name ID.

3.2.1 Full Transcript Putative Gene Reference File

Using dplyr the BLAST results and these gene ID mappings were joined and filtered to create a reference file of transcript IDs assigned both a UniProt Swiss-Prot database protein ID, and a gene name.

The BLASTx results were imported:

```
setwd("~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters")

Unisprot.BLASTx.out <- read.csv("pygmy.ANGEL_blastx_sprot-maxtarg1-maxhsp1.out",
    sep = "\t", header = FALSE)
head(Unisprot.BLASTx.out)
```

```
##                                                       V1     V2     V3  V4  V5
## 1     PB.2.1|002537|path0:1-1624(+)|transcript/18304|m.1 Q6PBC3 95.000 100   5
## 2    PB.3.2|00361c|path0:43-2240(+)|transcript/10564|m.2 P55209 88.995 209  22
## 3     PB.11.1|004815|path2:1-3039(+)|transcript/3426|m.4 P36633 61.892 719 258
## 4 PB.11.2|004815|path2:1047-3035(+)|transcript/13401|m.5 Q8JZQ5 67.010 194  63
## 5    PB.12.1|004815|path5:1-1665(+)|transcript/17167|m.6 Q8JZQ5 61.712 222  80
## 6     PB.13.1|004c10|path2:1-2811(+)|transcript/4993|m.7 Q9H063 83.984 256  38
##   V6  V7   V8  V9 V10       V11 V12
## 1  0   1  300   1 100  2.49e-69 207
## 2  1   1  627 184 391  5.86e-61 198
## 3  8  46 2175  32 743  0.00e+00 855
## 4  1 196  777 513 705  3.11e-92 246
## 5  3   1  666 532 748  3.96e-77 249
## 6  1   1  759   1 256 4.61e-156 437
```

```
nrow(Unisprot.BLASTx.out)
```

```
## [1] 12894
```

Column Headers Renamed to match BLAST output format 6:

```
Unisprot.BLASTx.out <- Unisprot.BLASTx.out %>%
    rename(TranscriptID = V1, UniprotID = V2, pident = V3, length = V4,
        mismatch = V5, gapopen = V6, qstart = V7, qend = V8,
        sstart = V9, send = V10, evalue = V11, bitscore = V12)

colnames(Unisprot.BLASTx.out)
```

```
##  [1] "TranscriptID" "UniprotID"    "pident"       "length"       "mismatch"
##  [6] "gapopen"      "qstart"       "qend"         "sstart"       "send"
## [11] "evalue"       "bitscore"
```

```
# head(Unisprot.BLASTx.out) tail(Unisprot.BLASTx.out)
```

Gene ID mapping list for Uniprot IDs were imported:

```
BLASTx.to.gene <- read.csv("~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/pygmy.ANGEL_blastx_sprot-maxtarg1-maxhsp1==G
↪   Name IDs.txt",
    sep = "\t", header = TRUE)

head(BLASTx.to.gene)
```

```
##     From     To
## 1 Q6PBC3 chchd4
## 2 P55209 NAP1L1
## 3 P36633   Aoc1
## 4 Q8JZQ5   Aoc1
## 5 Q9H063   MAF1
## 6 A1XBS5 FAM92A
```

Column Headers were renamed:

```
BLASTx.to.gene <- BLASTx.to.gene %>%
    rename(UniprotID = From, Gene = To)
colnames(BLASTx.to.gene)
```

```
## [1] "UniprotID" "Gene"
```

Although The UniProt database returns a non-redundant, unique list, conventions of gene and protein names of different species means there is some duplication when treated as case sensitive. Here all protein IDs will be made upper case, and all gene IDs will be made lower case.

```
BLASTx.to.gene <- BLASTx.to.gene %>%
    mutate(UniprotID = toupper(UniprotID))

BLASTx.to.gene <- BLASTx.to.gene %>%
    mutate(Gene = tolower(Gene))

head(BLASTx.to.gene)
```

```
##   UniprotID   Gene
## 1    Q6PBC3 chchd4
## 2    P55209 nap1l1
## 3    P36633   aoc1
## 4    Q8JZQ5   aoc1
## 5    Q9H063   maf1
## 6    A1XBS5 fam92a
```

```r
nrow(BLASTx.to.gene)
```

```
## [1] 7221
```

Note that nrow = 7221, now the duplicated rows will be filtered out:

```r
BLASTx.to.gene %>%
    count(UniprotID) %>%
    filter(n > 1)
```

```
##   UniprotID n
## 1    P32969 4
```

UniProt ID P32969 corresponds to gene IDs: RPL9, RPL9P7, RPL9P8, and RPL9P9. As these data are not specifically assigned to any one of these 4 genes as they all have the same protein ID, and to avoid duplication of transcripts on joining, this will be simplified to one entry of RPL9.

```r
BLASTx.to.gene <- BLASTx.to.gene %>%
    distinct(UniprotID, .keep_all = TRUE)
nrow(BLASTx.to.gene)
```

```
## [1] 7218
```

```r
BLASTx.to.gene %>%
    group_by(Gene) %>%
    summarize(n = n())
```

```
## # A tibble: 6,343 x 2
##    Gene        n
##    <chr>   <int>
##  1 aaas        1
##  2 aadacl4     1
##  3 aadat       2
##  4 aagab       1
##  5 aamp        1
##  6 aanat       1
##  7 aar2        1
##  8 aars1       1
##  9 aarsd1      1
## 10 aass        1
## # ... with 6,333 more rows
```

There are now 7218 Unique Uniprot IDs listed and after making case consistent, 6343 unique Gene IDs listed. Multiple Protein IDs map to the same Gene ID. This may be due to data sourced from a variety of species' and particular transcripts returning a BLAST hit for gene orthologues, or gene haplotypes. These putative Gene IDs are for reference only, and protein IDs will not be removed from transcript data, so no information will be lost.

Assign gene ID to the BLAST result by joining based on UniProt ID

```
BLASTx.gene.join <- left_join(Unisprot.BLASTx.out, BLASTx.to.gene,
    by = "UniprotID")
# head(BLASTx.gene.join)
```

```
BLASTx.gene.join <- relocate(BLASTx.gene.join, Gene, .before = pident)
# head(BLASTx.gene.join) tail(BLASTx.gene.join)
```

3.2.1.1 Manipulation of 'BLASTx.gene.join_filt_clustered' to Match Reference Files

- The fasta file which contains *only* protein coding regions (and sequences which were part of the input for the BLASTx search) has the characters "|m.*" at the end of transcript IDs
- The fasta file containing the full length of these transcripts corresponsing to the clustered protein coding regions (above), and which was used as the reference file for Kallisto does not have this name extension

In order to be able to filter this file based on either naming convention and compare to BLASTx outputs, genes of interest lists, and gene expression results: Here a corresponding column with the "|m.*" removed is added

Duplicate the ID column

```
BLASTx.gene.join_split <- cbind(BLASTx.gene.join, replicate(1,
    BLASTx.gene.join$TranscriptID))
BLASTx.gene.join_split <- rename(BLASTx.gene.join_split, TranscriptID_2 = "replicate(1,
↪   BLASTx.gene.join$TranscriptID)")
names(BLASTx.gene.join_split)
```

```
##  [1] "TranscriptID"   "UniprotID"      "Gene"           "pident"
##  [5] "length"         "mismatch"       "gapopen"        "qstart"
##  [9] "qend"           "sstart"         "send"           "evalue"
## [13] "bitscore"       "TranscriptID_2"
```

```
BLASTx.gene.join_split <- relocate(BLASTx.gene.join_split, "TranscriptID_2",
    .before = "UniprotID")
# head(BLASTx.gene.join_split)
```

Split the Transcript ID column based on the last "|" and return as a data frame

```
BLASTx.gene.join_split2 <-
↪   lapply(stri_split_regex(stri_reverse(BLASTx.gene.join_split$TranscriptID_2),
    pattern = "[|\\s]+", n = 2), stri_reverse)
BLASTx.gene.join_split2 <- setNames(data.table::transpose(BLASTx.gene.join_split2)[2:1],
    c("output1", "output2"))
BLASTx.gene.join_split2 <- as.data.frame(c(list(input =
↪   BLASTx.gene.join_split$TranscriptID_2),
    BLASTx.gene.join_split2))
# head(BLASTx.gene.join_split2)
```

Rename Columns and Join the truncated columns back into the original dataframe based on TranscriptID_2

```r
names(BLASTx.gene.join_split2)
```

```
## [1] "input"   "output1" "output2"
```

```r
colnames(BLASTx.gene.join_split2) <- c("TranscriptID_2", "TranscriptID_3",
    "TranscriptID_4")
# head(BLASTx.gene.join_split2)

BLASTx.gene.join_split3 <- full_join(BLASTx.gene.join_split,
    BLASTx.gene.join_split2, by = "TranscriptID_2")
# head(BLASTx.gene.join_split3)
```

Check: (they all have the same numbers of rows and no data was lost)

```r
nrow(BLASTx.gene.join)
```

```
## [1] 12894
```

```r
nrow(BLASTx.gene.join_split)
```

```
## [1] 12894
```

```r
nrow(BLASTx.gene.join_split2)
```

```
## [1] 12894
```

```r
nrow(BLASTx.gene.join_split3)
```

```
## [1] 12894
```

Remove the now unneeded duplicate 'TranscriptID_2' column and 'TranscriptID_4' column Reorder columns so that TranscriptID_3 appears on the left next to the full TranscriptID

```r
BLASTx.gene.join_split4 <- subset(BLASTx.gene.join_split3, select = -c(TranscriptID_2,
    TranscriptID_4))
BLASTx.gene.join_split4 <- relocate(BLASTx.gene.join_split4,
    TranscriptID_3, .before = UniprotID)
# head(BLASTx.gene.join_split4)
```

Reorder the rows so that the Data is arranged by Gene in alphabetical order (and so that transcripts coresponding to the same gene are listed together).

```r
BLASTx.gene.join_split4 <- BLASTx.gene.join_split4 %>%
    arrange(Gene)
# head(BLASTx.gene.join_split4)
nrow(BLASTx.gene.join_split4)  #confirming that number of rows remains consistent. The
↪  initial BLASTx result has 12894 rows, and 12894 rows remain.
```

```
## [1] 12894
```

The aforementioned 6343 genes are all present in this dataset, with the additional record representing rows with no Gene ID assigned to Protein ID.

```
BLASTx.gene.join_split4 %>%
    group_by(Gene) %>%
    summarize(n = n())
```

```
## # A tibble: 6,344 x 2
##    Gene        n
##    <chr>   <int>
##  1 aaas        1
##  2 aadacl4     2
##  3 aadat       9
##  4 aagab       3
##  5 aamp        2
##  6 aanat       2
##  7 aar2        2
##  8 aars1       1
##  9 aarsd1      1
## 10 aass        1
## # ... with 6,334 more rows
```

Write the output to .csv

```
write.csv(BLASTx.gene.join_split4, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/BLASTx.gene.join_split4.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)
```

Remove all of the rows where Gene = NA

```
BLASTx.gene.join_filt <- BLASTx.gene.join_split4 %>%
    filter(!is.na(Gene))
head(BLASTx.gene.join_filt)
```

```
##                                                     TranscriptID
## 1 PB.7010.1|transcript/12012:1-2076(+)|transcript/12012|m.12781
## 2      PB.2837.1|6a2a91|path0:1-1784(+)|transcript/15955|m.5151
## 3      PB.2838.1|6a2a91|path1:1-2161(+)|transcript/11047|m.5152
## 4      PB.2778.1|67914a|path1:1-1875(+)|transcript/14733|m.5057
## 5     PB.2778.2|67914a|path1:22-1698(+)|transcript/16857|m.5058
## 6      PB.6589.1|fb4549|path0:1-2510(+)|transcript/7231|m.12232
##                                           TranscriptID_3 UniprotID    Gene
## 1 PB.7010.1|transcript/12012:1-2076(+)|transcript/12012    Q9NRG9    aaas
## 2     PB.2837.1|6a2a91|path0:1-1784(+)|transcript/15955    Q5VUY2 aadacl4
## 3     PB.2838.1|6a2a91|path1:1-2161(+)|transcript/11047    Q5VUY2 aadacl4
## 4     PB.2778.1|67914a|path1:1-1875(+)|transcript/14733    Q5E9N4   aadat
## 5    PB.2778.2|67914a|path1:22-1698(+)|transcript/16857    Q5E9N4   aadat
## 6      PB.6589.1|fb4549|path0:1-2510(+)|transcript/7231    Q8N5Z0   aadat
##   pident length mismatch gapopen qstart qend sstart send    evalue bitscore
```

```
## 1 66.048    539       162        2     127 1737       1  520  0.00e+00      684
## 2 44.477    344       188        3       1 1026      63  405  8.03e-98      298
## 3 42.820    383       217        2      79 1224      24  405 6.29e-108      327
## 4 68.235    425       135        0       1 1275       1  425  0.00e+00      647
## 5 68.235    425       135        0       1 1275       1  425  0.00e+00      647
## 6 58.824    425       174        1       1 1272       1  425  0.00e+00      543
```

```
tail(BLASTx.gene.join_filt)
```

```
##                                                     TranscriptID
## 12527   PB.2713.1|65abaa|path0:1-2495(+)|transcript/7761|m.4957
## 12528 PB.2713.2|65abaa|path0:37-2312(+)|transcript/10678|m.4958
## 12529  PB.2713.3|65abaa|path0:37-2495(+)|transcript/7693|m.4959
## 12530  PB.2713.3|65abaa|path0:37-2495(+)|transcript/7693|m.4960
## 12531   PB.2714.1|65abaa|path1:1-2323(+)|transcript/9134|m.4961
## 12532  PB.2714.2|65abaa|path1:4-2323(+)|transcript/10134|m.4962
##                                            TranscriptID_3 UniprotID Gene pident
## 12527   PB.2713.1|65abaa|path0:1-2495(+)|transcript/7761    Q04584  zyx 64.286
## 12528 PB.2713.2|65abaa|path0:37-2312(+)|transcript/10678    Q04584  zyx 81.985
## 12529  PB.2713.3|65abaa|path0:37-2495(+)|transcript/7693    Q04584  zyx 92.857
## 12530  PB.2713.3|65abaa|path0:37-2495(+)|transcript/7693    Q04584  zyx 66.055
## 12531   PB.2714.1|65abaa|path1:1-2323(+)|transcript/9134    Q04584  zyx 81.724
## 12532  PB.2714.2|65abaa|path1:4-2323(+)|transcript/10134    Q04584  zyx 81.724
##       length mismatch gapopen qstart qend sstart send    evalue bitscore
## 12527     84       24       1    514  765    285  362  7.93e-29      107
## 12528    272       43       1    832 1647    258  523 1.66e-163      479
## 12529    126        9       0      3  380    375  500  7.10e-85      260
## 12530    109       31       1    931 1257    258  360  3.96e-37      144
## 12531    290       47       1   1090 1959    258  541 6.38e-175      512
## 12532    290       47       1    988 1857    258  541 2.28e-175      511
```

```
nrow(BLASTx.gene.join_filt)
```

```
## [1] 12532
```

See above, 12532 transcripts remain which were mapped to one of the 6343 Gene IDs.

Write the output to .csv

```
write.csv(BLASTx.gene.join_filt, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/BLASTx.gene.join_filt.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)
```

3.2.2 Putative Genes for Clustered Transcripts used as the Gene Expression Analysis Reference

The above list of transcripts represent all transcript isoforms assembled for *T. adelaidensis* with a predicted open reading frame determined by ANGEL. After further clustering using translated protein sequences in CD-HIT, a smaller reference set of transcripts were used as the reference for gene expresion analysis.

The above list will be filtered based on the same list of cluster representatives to create a smaller list of all BLAST results relating to the transcripts used as gene expression references.

Import the list of unique clustered transcripts that were used as the reference for gene expression analysis. Note that in this case, data after the ANGEL step are being sub-set, as noted when coding regions were predicted, a few full length transcripts give rise to more than one transcript isoform listed here, and thus may be assigned more than one putative gene ID.

ClstrTranscriptID is already loaded:

```r
head(clstrTranscriptID)
```

```
## [1] PB.6088.11|e88f16|path1:164-2792(+)|transcript/22345|m.11362
## [2] PB.2919.1|6c1fb9|path4:13-2574(+)|transcript/10624|m.5342
## [3] PB.535.2|136fa8|path0:1-2775(+)|transcript/4312|m.913
## [4] PB.4689.2|afa3de|path19:22-2096(+)|transcript/11491|m.8666
## [5] PB.6357.1|f27f14|path2:4-2108(+)|transcript/12511|m.11846
## [6] PB.715.4|1b393a|path6:6-8031(+)|transcript/8|m.1216
## 13882 Levels: PB.10.1|004815|path1:5-1713(+)|transcript/17534|m.3 ...
```

Filter the BLASTx results with gene name based on the transcript references used for the expression analysis (i.e. the clustered transcript file).

```r
BLASTx.gene.join_filt_clustered <- BLASTx.gene.join_filt %>%
    filter(TranscriptID %in% clstrTranscriptID)
# head(BLASTx.gene.join_filt_clustered)
nrow(BLASTx.gene.join_filt_clustered)
```

```
## [1] 8861
```

Note: of the initial 13882 sequences in the predicted open read frame set submitted for a BLASTx search, only 12602 query sequences returned a match, and not all protein ID matches returned a Gene ID match. Coupled with some transcripts that may have been identified to putative gene ID being removed at the CD-HIT clustering step, the total remaining number of transcripts with a coresponding Protein ID and Gene ID here is expected to be less than the clustered reference transcript dataset containing 9813 unique full-length transcripts.
Ninety four full length transcripts produced more than one isoform in predicted open read frame which was retained at the clustering step, so there is also the possibility that duplicate entries here apply to a single full length 'master' transcript.

Here 8861 transcripts in predicted open read frame have been assigned both a protein ID and putative gene ID, from a total list of 9813 corresponding full length transcripts which were used as a reference in later gene expression analysis.

3.2.3 Presence of Identified "Genes of Interest" in the *T. adelaidensis* Transcript Set

In order to generate a list of potential "genes of interest" and narrow down the focus of analysis, an NCBI gene database search was conducted aiming for genes identified in reptiles associated with renal function, water homeostasis and heat regulation.

The final NCBI Gene database search was accessed on on June 14, 2020 including the following terms:
- (Sauria[Organism]) AND (Aquaporin OR Bile OR Dehydration OR Diffus* OR Excretion OR Filt* OR Fluid OR (Heat AND Stress) OR Heat Shock OR Heat Stress OR Hibernation OR Homeostasis OR Ion Channel OR Ion Transport OR Kidney OR Membrane And (Potential OR Permeability OR Pore) OR Metabolic OR Osmo* OR Permeability OR Ph Balance OR Renal OR Solute OR Stress OR Temperature OR Thermal OR Uric OR Water OR Water Retention OR Water Permeability OR Water Transport)

These full search results were exported and refined in excel with care taken to manually preserve gene symbol data fields. Note: summary counts of genes IDs assigned to transcripts was initially conducted in excel (Appendix 3). Manual manipulation of the exported database data was conducted to summarise genes identified, and to create a list of unique genes returned by the search term. Duplicate gene entries were prioritised by taxa and a list of 993 genes of particular interest to this study were identified.

Excel was used to create initial summary statistics of the taxa and genes returned, as well as record information retained at each step of manual filtering.

Excel is not ideal for the manipulation of lists of gene names this represents preliminary data exploration, so great care was taken to ensure gene IDs integrity was maintained and all further filtering was conducted in R below. The following analysis was conducted in to subset the sequenced *T. adelaidensis* transcript lists based on putative gene annotation.

Import the reference "genes of interest" list built from an NCBI database search (Created for reference in Chapter 3, outlined in Appendix 3). This imported list is after initial duplicate filtering in excel as well as adding a column for gene sumbol/ID in all lower case. It contains 993 unique gene symbols for the most favoured taxa (as per the list in Appendix 3).

```
Sauria_2020.06.14 <- read.csv("~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/Returned organisms 2020-06-15
↪   Sauria-RM-dup-loc FULL.csv",
    sep = ",", header = TRUE, )
Sauria_2020.06.14 <- Sauria_2020.06.14 %>%
    select("tax_id", "Org_name.new_name", "Org_name", "GeneID",
        "Symbol", "Symbol.CAPS", "Aliases", "description", "other_designations")
nrow(Sauria_2020.06.14)
```

```
## [1] 993
```

```
head(Sauria_2020.06.14)
```

```
##   tax_id Org_name.new_name            Org_name    GeneID Symbol Symbol.CAPS
## 1   9031                 J       Gallus gallus    416811   AACS        aacs
## 2  38654                 I  Alligator sinensis 102374204  ABCA5       abca5
## 3   9031                 J       Gallus gallus    418791  ABCC4       abcc4
## 4   9031                 J       Gallus gallus    423767  ABCG2       abcg2
## 5  28377                 A Anolis carolinensis 100565298   ace2        ace2
## 6   9031                 J       Gallus gallus    373916   ACO1        aco1
##        Aliases                                                    description
## 1                                                  acetoacetyl-CoA synthetase
## 2                                   ATP binding cassette subfamily A member 5
## 3                                   ATP binding cassette subfamily C member 4
## 4              ATP binding cassette subfamily G member 2 (Junior blood group)
## 5                                           angiotensin I converting enzyme 2
## 6 IREB1- IREBP                                           aconitase 1- soluble
##
## 1
## 2                                                                                          ATP-bindi
## 3                                                        multidrug resistance-associated protein 4|A
## 4 broad substrate specificity ATP-binding cassette transporter ABCG2|ATP-binding cassette sub-family
## 5
## 6                    cytoplasmic aconitate hydratase|IRE-BP 1|IRE-binding protein 1|Iron responsive
```

The “Gene Symbol” column output by the NCBI database matches the format of “gene names” output by the conversion of UniProt Protein IDs generated

Check Gene ID title case for non-unique entries

```
Sauria_2020.06.14 %>%
    count(Symbol.CAPS) %>%
    filter(n > 1)
```

```
## [1] Symbol.CAPS n
## <0 rows> (or 0-length row.names)
```

Rename column Symbol.CAPS to Gene so column can be compared to the Gene column in BLASTx.gene.join_filt, make this column characters to allow joins

```
Sauria_2020.06.14 <- rename(Sauria_2020.06.14, Gene = "Symbol.CAPS")
Sauria_2020.06.14$Gene <- as.character(Sauria_2020.06.14$Gene)
# head(Sauria_2020.06.14)
```

Create a larger summary file of the transcripts which have a BLASTx result. All BLASTx results will be retained, but there is no need to retain information for ‘genes of interest’ which were not identified in this dataset. This will be used later to identify information on transcripts which may be identified in the gene expression analysis.

```
BLASTx.gene.join_Database.Full.PBT.Summary <- left_join(BLASTx.gene.join_filt,
    Sauria_2020.06.14, by = "Gene")
nrow(BLASTx.gene.join_Database.Full.PBT.Summary)
```

```
## [1] 12532
```

```
head(BLASTx.gene.join_Database.Full.PBT.Summary)
```

```
##                                                      TranscriptID
## 1 PB.7010.1|transcript/12012:1-2076(+)|transcript/12012|m.12781
## 2      PB.2837.1|6a2a91|path0:1-1784(+)|transcript/15955|m.5151
## 3      PB.2838.1|6a2a91|path1:1-2161(+)|transcript/11047|m.5152
## 4      PB.2778.1|67914a|path1:1-1875(+)|transcript/14733|m.5057
## 5     PB.2778.2|67914a|path1:22-1698(+)|transcript/16857|m.5058
## 6      PB.6589.1|fb4549|path0:1-2510(+)|transcript/7231|m.12232
##                                         TranscriptID_3 UniprotID    Gene
## 1 PB.7010.1|transcript/12012:1-2076(+)|transcript/12012    Q9NRG9    aaas
## 2     PB.2837.1|6a2a91|path0:1-1784(+)|transcript/15955    Q5VUY2 aadacl4
## 3     PB.2838.1|6a2a91|path1:1-2161(+)|transcript/11047    Q5VUY2 aadacl4
## 4     PB.2778.1|67914a|path1:1-1875(+)|transcript/14733    Q5E9N4   aadat
## 5    PB.2778.2|67914a|path1:22-1698(+)|transcript/16857    Q5E9N4   aadat
## 6      PB.6589.1|fb4549|path0:1-2510(+)|transcript/7231    Q8N5Z0   aadat
##   pident length mismatch gapopen qstart qend sstart send    evalue bitscore
## 1 66.048    539      162       2    127 1737      1  520  0.00e+00      684
## 2 44.477    344      188       3      1 1026     63  405  8.03e-98      298
## 3 42.820    383      217       2     79 1224     24  405 6.29e-108      327
## 4 68.235    425      135       0      1 1275      1  425  0.00e+00      647
```

```
## 5 68.235    425      135       0      1 1275      1  425  0.00e+00      647
## 6 58.824    425      174       1      1 1272      1  425  0.00e+00      543
##   tax_id Org_name.new_name Org_name GeneID Symbol Aliases description
## 1     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 2     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 3     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 4     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 5     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 6     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
##   other_designations
## 1               <NA>
## 2               <NA>
## 3               <NA>
## 4               <NA>
## 5               <NA>
## 6               <NA>
```

Export this file as a .csv

```
write.csv(BLASTx.gene.join_Database.Full.PBT.Summary, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/BLASTx.gene.join_Database.Full.PBT.Summary.c
    quote = FALSE, row.names = FALSE, col.names = TRUE)
```

The number of transcripts which returned a BLASTx result for each gene was already counted in a previous section. Make this a dataframe.

```
Gene_transcript_count <- BLASTx.gene.join_split4 %>%
    group_by(Gene) %>%
    summarize(n = n())
head(Gene_transcript_count)
```

```
## # A tibble: 6 x 2
##   Gene        n
##   <chr>   <int>
## 1 aaas        1
## 2 aadacl4     2
## 3 aadat       9
## 4 aagab       3
## 5 aamp        2
## 6 aanat       2
```

Filter the genes of interest list based on Gene IDs which match to a BLASTx result from the full transcript list

```
Sauria_2020.06.14_Transcripts.Present.Full <- semi_join(Sauria_2020.06.14,
    BLASTx.gene.join_filt, by = "Gene")
# head(Sauria_2020.06.14_Transcripts.Present.Full)
nrow(Sauria_2020.06.14_Transcripts.Present.Full)
```

```
## [1] 393
```

Add the counts of number of transcripts matching each 'gene of interest' as a new column to the genes of interest table, reorder columns.

```
Sauria_2020.06.14_Transcripts.Present.Count <-
↪   left_join(Sauria_2020.06.14_Transcripts.Present.Full,
    Gene_transcript_count, by = "Gene")
Sauria_2020.06.14_Transcripts.Present.Count <-
↪   rename(Sauria_2020.06.14_Transcripts.Present.Count,
    Transcript_Count = "n")

Sauria_2020.06.14_Transcripts.Present.Count <-
↪   relocate(Sauria_2020.06.14_Transcripts.Present.Count,
    description, .before = Aliases)
Sauria_2020.06.14_Transcripts.Present.Count <-
↪   relocate(Sauria_2020.06.14_Transcripts.Present.Count,
    Transcript_Count, .before = Aliases)
Sauria_2020.06.14_Transcripts.Present.Count <-
↪   select(Sauria_2020.06.14_Transcripts.Present.Count,
    -Org_name.new_name)  #this column was used as alphabetical factors in excel to sort
     ↪  the organisms by category so that the desired filtering based on taxa group could
     ↪  be achieved when duplicate gene database results were removed. It is no longer
     ↪  needed
head(Sauria_2020.06.14_Transcripts.Present.Count)
```

```
##   tax_id      Org_name GeneID Symbol   Gene
## 1   9031 Gallus gallus 418791  ABCC4  abcc4
## 2   9031 Gallus gallus 373916   ACO1   aco1
## 3   9031 Gallus gallus 420090 ACSBG2 acsbg2
## 4   9031 Gallus gallus 421534  ACTA1  acta1
## 5   9031 Gallus gallus 373918  ACTN1  actn1
## 6   9031 Gallus gallus 419194    ADA    ada
##                                     description Transcript_Count      Aliases
## 1     ATP binding cassette subfamily C member 4                2
## 2                          aconitase 1- soluble                4 IREB1- IREBP
## 3 acyl-CoA synthetase bubblegum family member 2                3
## 4                actin alpha 1- skeletal muscle                2
## 5                              actinin- alpha 1                2
## 6                           adenosine deaminase                1
##
## 1                                     multidrug resistance-associated protein 4|ATP-binding cassette-
## 2 cytoplasmic aconitate hydratase|IRE-BP 1|IRE-binding protein 1|Iron responsive element binding pro
## 3                                                                                             long-c
## 4
## 5                                                          alpha-actinin-1|F-actin cross-linking pro
## 6
```

Export this faile as a .csv

```
write.csv(Sauria_2020.06.14_Transcripts.Present.Full, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/Sauria_2020.06.14_Transcripts.Present.Count.
    quote = FALSE, row.names = FALSE, col.names = TRUE)
```

Filter the genes of interest list based on Gene IDs which match to a BLASTx result from the clustered transcript list used as a reference for gene expression analysis

```
Sauria_2020.06.14_Transcripts.Present.Clustered <- semi_join(Sauria_2020.06.14,
    BLASTx.gene.join_filt_clustered, by = "Gene")
nrow(Sauria_2020.06.14_Transcripts.Present.Clustered)
```

```
## [1] 392
```

Note: only a single gene that was both identified through the BLASTx search *and* identified as a 'gene of interest' in the database search was lost in this filtering step.

filter the Full transcript list based on Gene symbol identified in NCBI database search

```
nrow(BLASTx.gene.join_filt)
```

```
## [1] 12532
```

```
BLASTx.gene.join_Database.Present <- semi_join(BLASTx.gene.join_filt,
    Sauria_2020.06.14, by = "Gene")
nrow(BLASTx.gene.join_Database.Present)
```

```
## [1] 955
```

Of the 12532 transcripts which returned a BLASTx result for their predicted coding region, 955 are represented in the 'genes of interest' dataset created from an NCBI gene database search of terms relevant to renal function and water homeostasis.

Export this faile as a .csv

```
write.csv(BLASTx.gene.join_Database.Present, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/BLASTx.gene.join_Database.Present.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)
```

3.3 BLAST 2 GO

BLAST2GO (Within OmicsBox – Bioinformatics Made Easy, BioBam Bioinformatics) was used to perform a gene ontology analysis of the above BLASTx results.

Götz S., Garcia-Gomez JM., Terol J., Williams TD., Nagaraj SH., Nueda MJ., Robles M., Talon M., Dopazo J. and Conesa A. (2008). High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic acids research, 36(10), 3420-35.

For import into the program a BLASTx search with the output set to .xml was performed using the fasta file containing the predicted open reading frame compared to the *Anolis carolinensis* proteins database, and a second search to the UniProt Swiss-Prot database (-max_target_seqs 5 -max_hsps 5 -evalue 0.00001).

3.3.1 UniProt Swiss-Prot Protein Database BLASTx

Final GO data analysis as included in Chapter 3 was performed on the BLASTx search using the Swiss-Prot proteins database in order to remain consistent with other annotation comparisons performed here, and due to a larger number of returned results. This analysis was completed within a Windows x64 v2.0.36 build of the OmicsBox program and figures were exported as below.

BLASTx was run on the eRSA NECTAR machine, using BLAST 2.9.0 against the UniProt Swiss-Prot database downloaded on 28/05/19.
Due to import requirements for Blast2GO output format 5 was used instead of output format 6 which is used in all other sections above. The BLASTx query was:

```
/mnt/Prog/blast/ncbi-blast-2.9.0+/bin/blastx -db
↪  /mnt/Prog/blast/blastdb/sprot/sprot -query
↪  /mnt/IsoSeq-analysis/data/ANGEL/pygmy.ANGEL.cds -out
↪  pygmy.ANGEL_blastx_sprot-BLAST2GO -outfmt 5 -max_target_seqs 5 -max_hsps 5
↪  -evalue 0.00001 -num_threads 4
```

With maximum target sequences = 5, Maximum hits per sequence = 5, and a required e-value of 0.00001. Note this limits the results less than the search used above, which limited maximum target sequences to 1, and maximum hits per sequence to 1 so that the top result could easily be retrieved manually. As evident in the statistics below, OmicsBox is capable of interpreting multiple hits per query and mapping accordingly without duplicating data beyond the input 13,882 input sequences.

Blastx results were imported into the program Omics Box, where results were then mapped and annotated to GO terms.

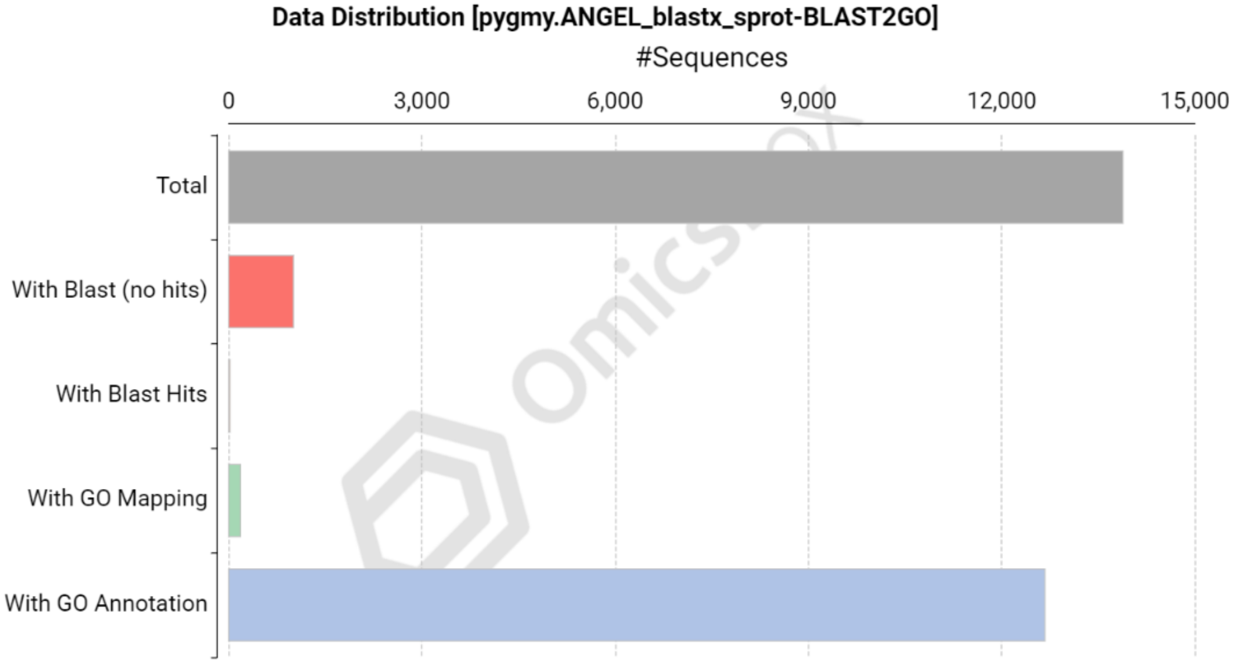


Figure 3: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Summary data distribution using the UniProt Swiss-Prot protein BLAST database. The file of predicted *T. adelaidensis* transcripts in open reading frame (pygmy.ANGEL) contains 13,882 unique sequences representing predicted open reading frame of transcripts: 12672 of these sequences produced >1 BLASTx hit with Mapping and GO Annotation, 186 produced a BLASTx hit and mapping only, 19 produced a BLAST hit only, and 1005 did not produce BLASTx hits

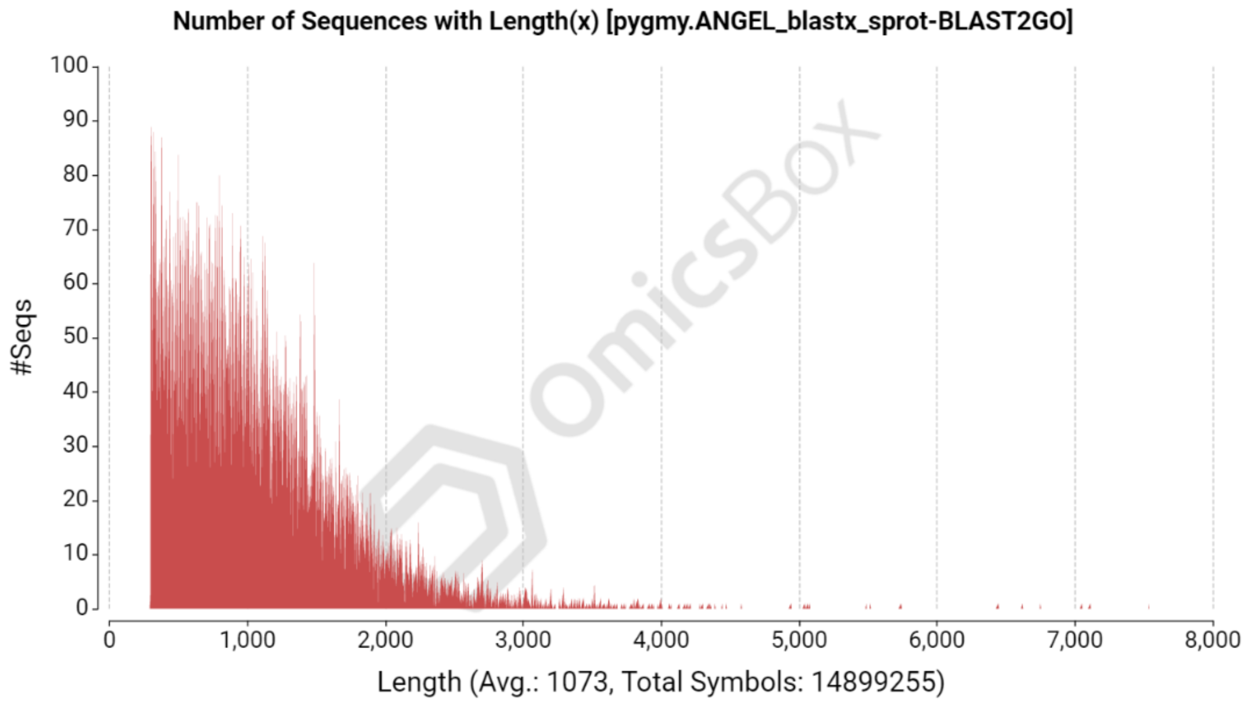


Figure 4: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Gene ontology conducted using B2GO. Summary of number of sequences with length (x) The file of predicted *T. adelaidensis* transcripts in open reading frame (pygmy.ANGEL) has an average sequence length of 1073bp

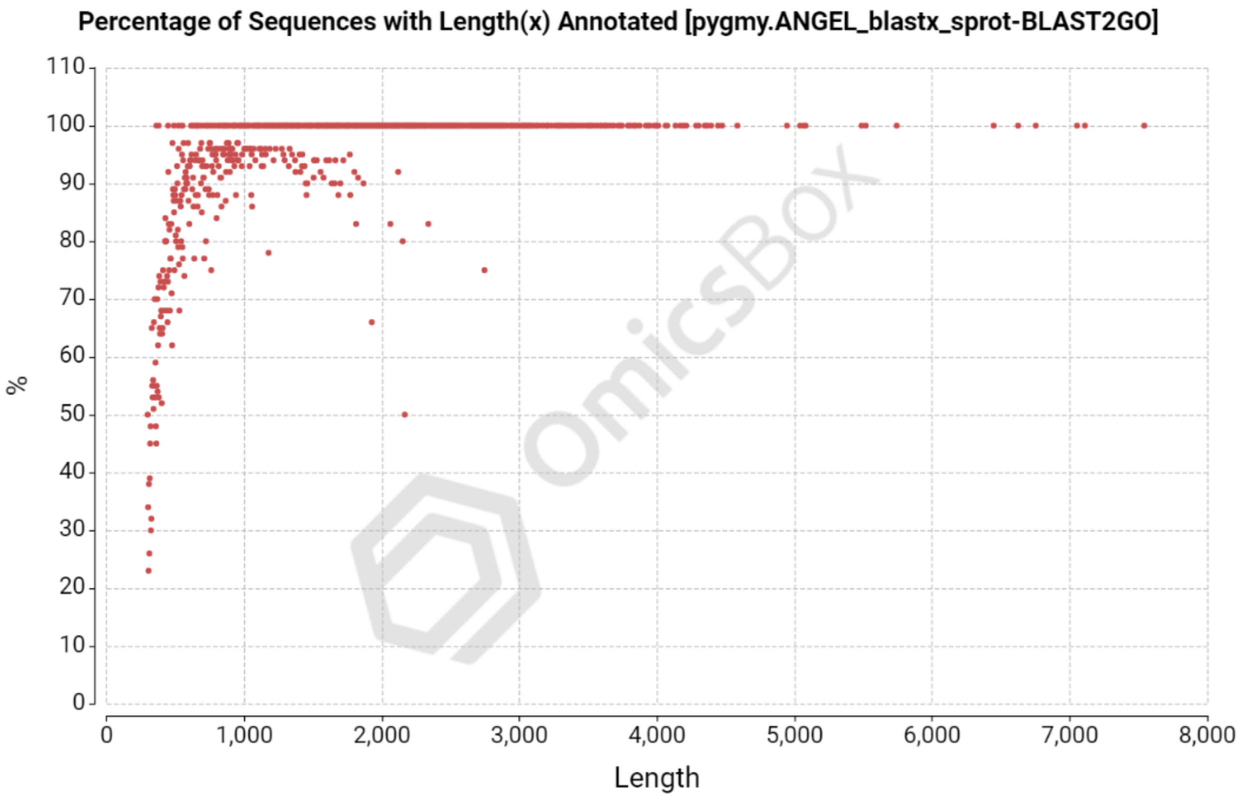


Figure 5: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Gene ontology conducted using B2GO. Percentage of sequences with length (x) (Predicted *T. adelaidensis* transcripts in open reading frame (pygmy.ANGEL) as BLAST query against the UniProt Swiss-Prot database formatted to a BLAST database))

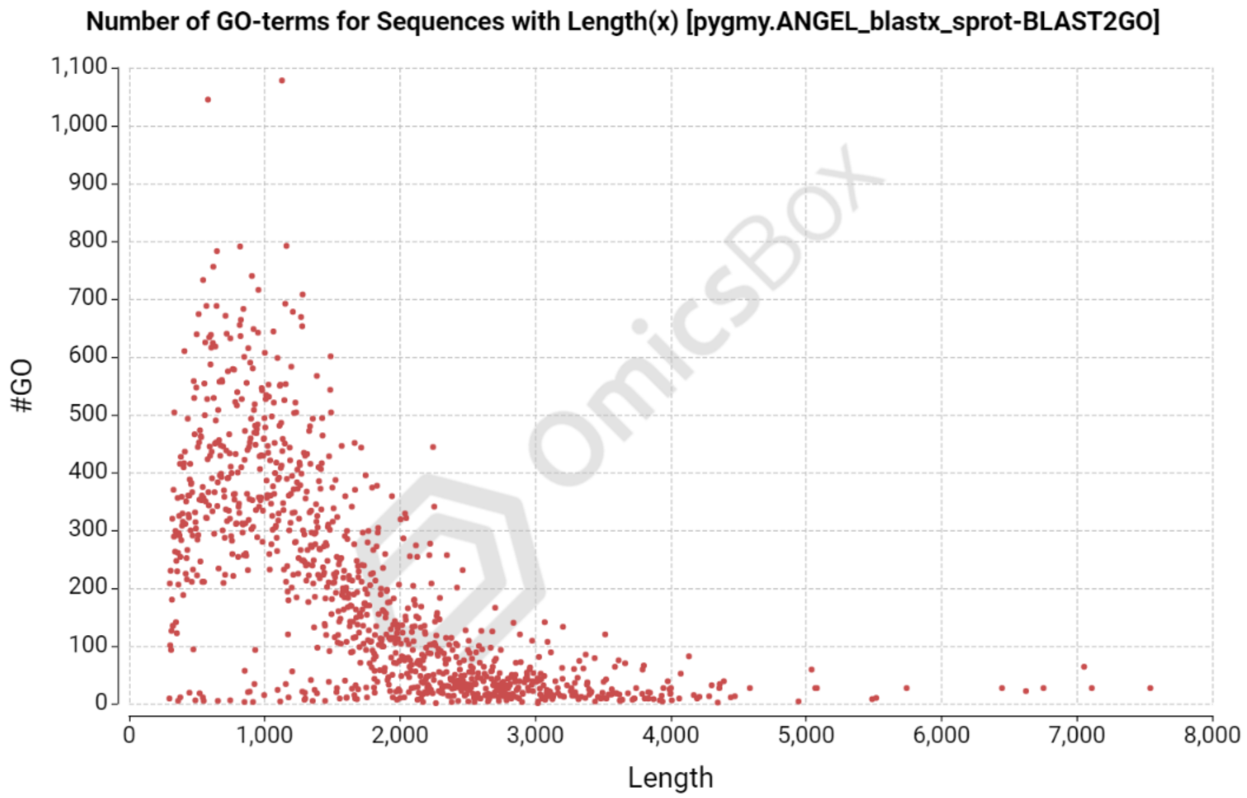


Figure 6: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Gene ontology conducted using B2GO. Number of GO terms annotated to sequences with length (x) (Predicted *T. adelaidensis* transcripts in open reading frame (pygmy.ANGEL) as BLAST query against the UniProt Swiss-Prot database formatted to a BLAST database)

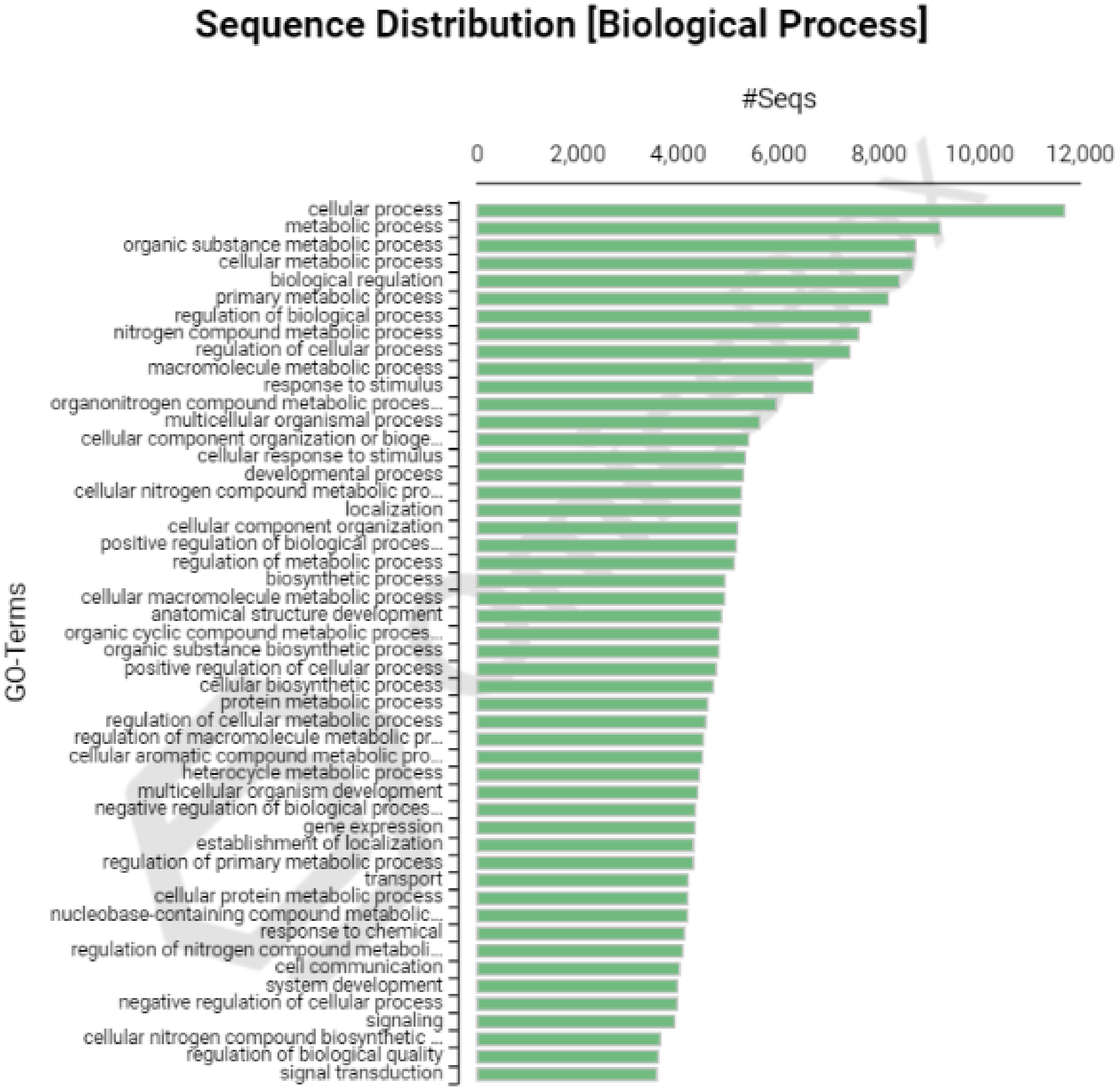


Figure 7: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Gene ontology conducted using B2GO. Top gene ontology terms by number of annotated sequences for all GO levels in the biological process category for predicted Tiliqua adelaidensis transcripts in open reading frame. (pygmy.ANGEL)

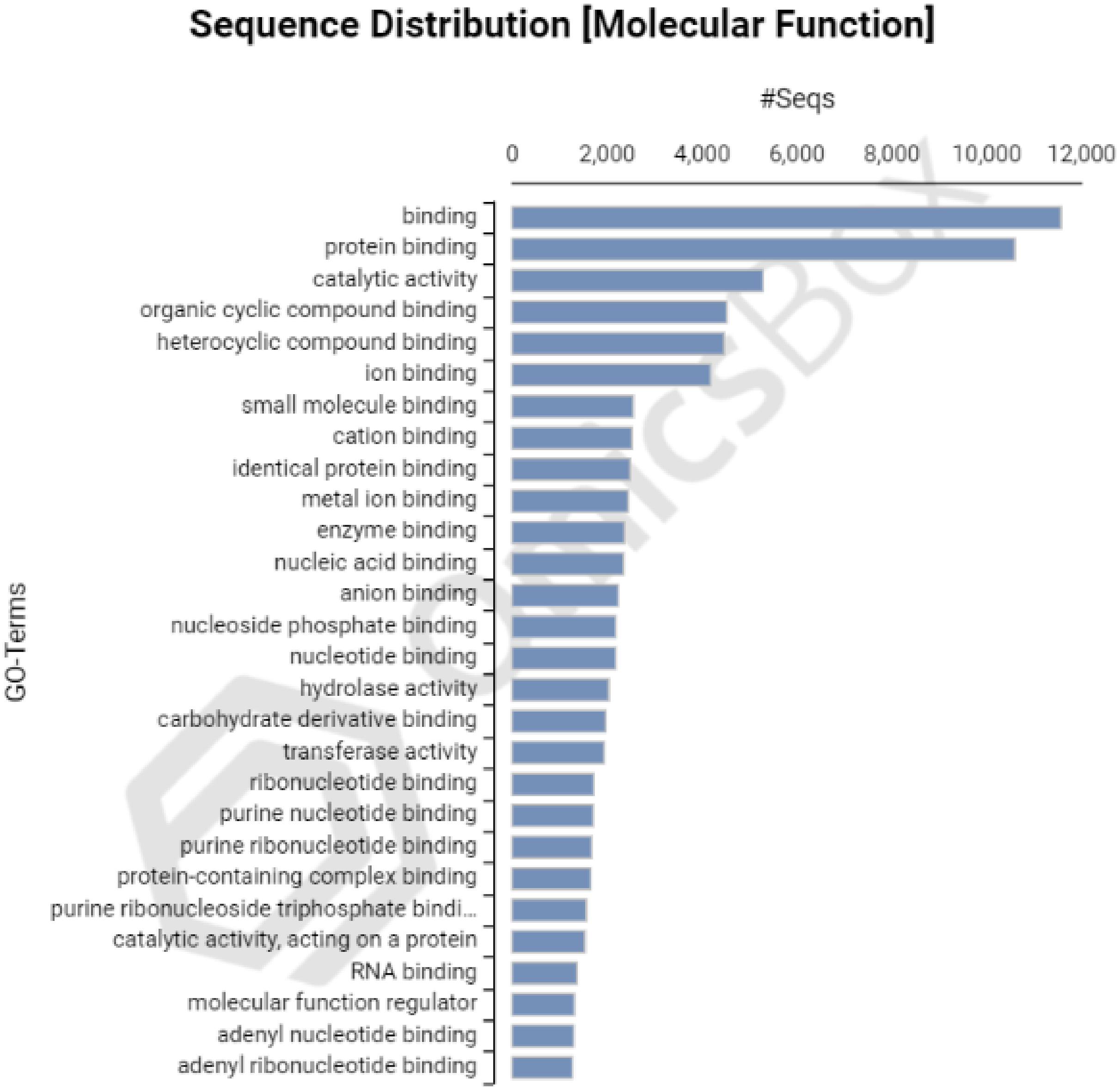


Figure 8: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Gene ontology conducted using B2GO. Top gene ontology terms by number of annotated sequences for all GO levels in the molecular function category for predicted Tiliqua adelaidensis transcripts in open reading frame. (pygmy.ANGEL)

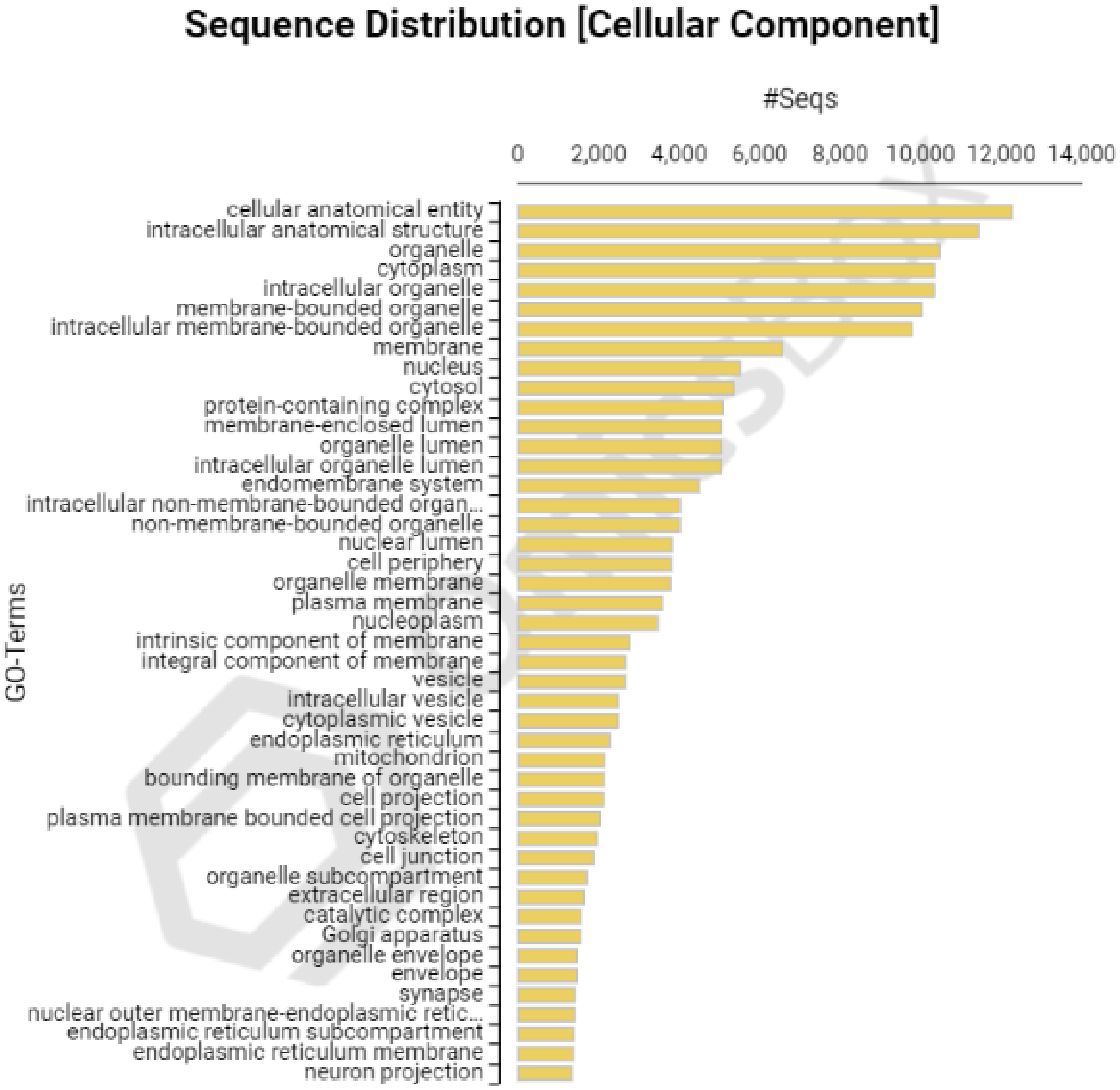


Figure 9: Poly(A) selected mRNA, isolated from kidney tissue from one *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Gene ontology conducted using B2GO. Top gene ontology terms by number of annotated sequences for all GO levels in the cellular component category for predicted Tiliqua adelaidensis transcripts in open reading frame. (pygmy.ANGEL)

3.4 Summary

15,729 Total transcript isoforms retained in initial cleaning and clustering of redundant isoforms.

13,882 sequences in a predicted open reading frame.

9,907 clusters identified based on translated proteins of predicted open reading frame.

9,813 full length transcripts have been subset into a reference file as the longest representatives of one or more transcript clusters.

12,602 of the 13,882 sequences in a predicted open reading frame were successfully annotated to a BLASTx result, protein ID, and gene ontology category when compared to the Anolis protein database.

12,672 of the 13,882 sequences in a predicted open reading frame were successfully annotated to a BLASTx result, protein ID, and gene ontology category when compared to the UniProt Swiss-Prot protein database

4 Comparison of Short-read assembly and Long-read transcripts

4.1 BUSCO

Completeness of assemblies were assessed using Benchmarking Universal Single-Copy Orthologs BUSCO version 5.4.2 (BUSCO) and the dataset vertebrata_odb10.

Moving finalised assembly files into the BUSCO Directory & Editing Headers

BUSCO requires simplified header formats as the symbols included in appended names and quality information in the provided pasta files interfere with its processing. Long read reference files were copied into the BUSCO directory and their headers altered, and the locations for where to call the trinity assemblies from are outlined in the bash document output-BUSCO__Name.sh:

```
#!/bin/bash
#
#
# These command lines were used to rename and copy assembled trinity & long read
↪  transcript datasets into the /BUSCO directory so all files have similar naming
↪  convention for input into the BUSCO script.
#
# The paths assume that a specific directory structure has been set up.
#
# Modules required: none
#
#
# Carmel Maher
# August 2020 edited Oct 2022




#to grab the trinity output for BUSCO:
#-------------------------------------


###SHORT_READ ASSEMBLIES

# go to the working directory - working from /clean
# cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/q  || error_exit "$LINENO:
↪  Directory Error 1"

#for file in *R1_cleanq.fq.gz
# do
#
#     FILESTEM=${file%_*}
#     #this FILESTEM only cuts to _clean, the _R1 is included in stem
#     FILESTEM=${FILESTEM/R1/}
#     #removes R1 from FILESTEM (FILESTEM ends in _ therefore not needed in "text"
↪  names)

#All individual samples are named as below:
#
#
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-sep/data/$FILESTEM"trinity-sep.Trini
```

```
#Concatenated assembly of all samples is named:
#
#
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
↪  /scratch/user/mahe0050/BUSCO/trinity-all_assembled.fasta  || error_exit
↪  "$LINENO: Error copying concatenated trinity-all output"


#####
#Note that using the above method all single sample trinity outputs can be called
↪  without editing or moving them.
#####


#--------------------------------------
### LONG READS (manually moved/renamed line by line)


cp -i
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Cogent/collected/hq.fasta.no5merge.collapsed.rep.
↪  /scratch/user/mahe0050/BUSCO/hq.fasta.no5merge.collapsed.rep.fasta

      # -> one long-read file should end up in
      ↪  /scratch/user/mahe0050/IsoSeq-analysis/BUSCO named
      ↪  hq.fasta.no5merge.collapsed.rep_assembled.fasta This contains the full
      ↪  length of all non-redundant high quality transcripts

cp -i /scratch/user/mahe0050/IsoSeq-analysis/data/ANGEL/pygmy.ANGEL.cds
↪  /scratch/user/mahe0050/BUSCO/pygmy.ANGEL.cds.fasta

      # -> one long-read file should end up in /scratch/user/mahe0050/BUSCO named
      ↪  pygmy.ANGEL_assembled.fasta This contains the predicted coding sequence
      ↪  of non-redundant unique transcripts before protein clustering

cp -i
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.clean.fasta
↪  /scratch/user/mahe0050/BUSCO/reference_transcripts.1Lv.clean.fasta

      # -> one long-read file should end up in
      ↪  /scratch/user/mahe0050/IsoSeq-analysis/BUSCO named
      ↪  reference_transcripts.1Lv.clean_assembled.fasta This contains the full
      ↪  length transcripts corresponding to the representative for clustered
      ↪  proteins translated from predicted open read frame


# header formats:
head /scratch/user/mahe0050/BUSCO/hq.fasta.no5merge.collapsed.rep.fasta
>PB.1.1|000643|path0:1-1879(+)|transcript/14742 transcript/14742
↪  full_length_coverage=3;length=1887;num_subreads=52

head /scratch/user/mahe0050/BUSCO/pygmy.ANGEL.cds.fasta
>PB.2.1|002537|path0:1-1624(+)|transcript/18304|m.1 type:likely-NA len:135 strand:+
↪  pos:282-686
```

```
head /scratch/user/mahe0050/BUSCO/reference_transcripts.1Lv.clean.fasta
>PB.2.1|002537|path0:1-1624(+)|transcript/18304 transcript/18304
↪  full_length_coverage=2;length=1627;num_subreads=26


grep -c ">" /scratch/user/mahe0050/BUSCO/hq.fasta.no5merge.collapsed.rep.fasta
15729
grep -c ">" /scratch/user/mahe0050/BUSCO/pygmy.ANGEL.cds.fasta
13882
grep -c ">" /scratch/user/mahe0050/BUSCO/reference_transcripts.1Lv.clean.fasta
9813


# NOTE the files have symbols in faste headers such as "/" and "+" before
↪  transcript IDs which need to be removed for BUSCO

# after sed to remove all "/" resulted in an error at metaeuk headers "ValueError:
↪  could not convert string to float: '+'"
# after sed to remove all "+" same error

# Goal is a PERCENT score of alignments to BUSCOs and which specific transcripts
↪  match is no explored further. Therefore all headers were grossly simplified &
↪  truncated to remove symbols

cut -d '|' -f1 /scratch/user/mahe0050/BUSCO/hq.fasta.no5merge.collapsed.rep.fasta >
↪  hq.fasta.no5merge.collapsed.rep_.fasta

cut -d '|' -f1 /scratch/user/mahe0050/BUSCO/pygmy.ANGEL.cds.fasta >
↪  pygmy.ANGEL.cds_.fasta

cut -d '|' -f1 /scratch/user/mahe0050/BUSCO/reference_transcripts.1Lv.clean.fasta >
↪  reference_transcripts.1Lv.clean_.fasta


# total number of sequences retained as counted above
grep -c ">" /scratch/user/mahe0050/BUSCO/hq.fasta.no5merge.collapsed.rep_.fasta
15729
grep -c ">" /scratch/user/mahe0050/BUSCO/pygmy.ANGEL.cds_.fasta
13882
grep -c ">" /scratch/user/mahe0050/BUSCO/reference_transcripts.1Lv.clean_.fasta
9813


# two of these files will have "duplicate" headers in this format as we know that
↪  some transcripts gave rise to >1 predicted cds. numbers were appended as below
↪  to make all sequence headers unique:
awk '/^>/{$0=$0"_"(++i)}1'  pygmy.ANGEL.cds_.fasta > pygmy.ANGEL.cds__.fasta
awk '/^>/{$0=$0"_"(++i)}1'  hq.fasta.no5merge.collapsed.rep_.fasta >
↪  hq.fasta.no5merge.collapsed.rep__.fasta

grep -c ">" hq.fasta.no5merge.collapsed.rep__.fasta
15729
```

```
grep -c ">" pygmy.ANGEL.cds__.fasta
13882
```

BUSCO Installation & Usage:

BUSCO was installed on Flinders Deep Thought machine using a Miniconda environment created as below

```
# ***
# BUSCO: Comparison of 3 stages of long read filtering, to de-novo assembled
↪  transcriptomes
# ***

cd /scratch/user/mahe0050/DE-analysis/bash

source ~/.bashrc
conda create -n BUSCOConda python=3.7
conda activate BUSCOConda
conda install -c conda-forge -c bioconda busco=5.4.2

busco --v
BUSCO 5.4.2
```

The script Slurm-BUSCO_DT was used to run the program BUSCO on all samples on the NECTAR machine.
Run BUSCO ->

```
#!/bin/bash
# ### (SLURM submission truncated for brevity) Important parameters are kept:
#
#SBATCH --job-name=mahe0050_BUSCO
#
#SBATCH --time=5-0
#
#SBATCH --ntasks=1
#
#SBATCH --cpus-per-task=12
#
#SBATCH --mem-per-cpu=8G

##################################################################
# Load any modules that are required.

module load Miniconda3/4.9.2
module load singularity/3.6.3


##################################################################
# Enter the command-line arguments that you job needs to run.

source ~/.bashrc
conda activate BUSCOConda

# See notes file associated with usage: Output-BUSCO_Name2020.sh and
↪  BUSCO_all_v5.0.sh
```

```
# Inputs include:
#     Short-read Trinity outputs for all 8 Kidney samples assembled from short
↪  reads
#     Short-read Trinity output for one file of all 8 kidney sequencing
↪  concatenated and assembled into a single transcript file

#     Isoseq3 long-read full list of non-redundant hq transcripts
#     Isoseq3 long-read full list of transcript predicted open read frame coding
↪  regions
#     Isoseq3 long-read representative transcripts of 'putative genes' based on
↪  coding sequence protein clustering, but including full-length including UTRs,
↪  with additional poly-a tail trimming


#------adjust these for your run-----

LINEAGE="vertebrata_odb10"

#-----------------------------------

function error_exit
{
    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}


#go to the directory containing trimmed files to pull ID names
cd /scratch/user/mahe0050/DE-analysis/1_trimmedData/q

for file in *R1_cleanq.fq.gz
  do
      FILESTEM=${file%_*}
      #this FILESTEM only cuts to _clean, the _R1 is included in stem
      FILESTEM=${FILESTEM/R1/}
      #removes R1 from FILESTEM (FILESTEM ends in _ therefore not needed in "text"
      ↪  names)

      mkdir /scratch/user/mahe0050/BUSCO/$FILESTEM"BUSCOout" || error_exit
      ↪  "$LINENO: Error creating trinity-sep output directory at $FILESTEM"

      cd /scratch/user/mahe0050/BUSCO/

      busco -i
      ↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-sep/data/$FILESTEM"trinity-sep
      ↪  -l $LINEAGE -f -c 12 -o $FILESTEM"BUSCOout" -m transcriptome ||
      ↪  error_exit "$LINENO: Error running BUSCO at $FILESTEM"

  done

echo "trinity-sep BUSCOs complete"
```

```
#
# ***
#


cd /scratch/user/mahe0050/BUSCO/

mkdir ./trinity-all_BUSCOout  || error_exit "$LINENO: directory error at
↪  trinity-all"

busco -i
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
↪  -l $LINEAGE -f -c 12 -o trinity-all_BUSCOout -m transcriptome || error_exit
↪  "$LINENO: Error running BUSCO at trinity-all"

echo "trinity-all BUSCOs complete"


#
# ***
#


 cd /scratch/user/mahe0050/BUSCO/

 mkdir -p ./hq-fasta_BUSCOout || error_exit "$LINENO: directory error at hq-fasta"

 /home/mahe0050/.conda/envs/BUSCOConda/bin/busco -i
 ↪  hq.fasta.no5merge.collapsed.rep__.fasta -l $LINEAGE -f -c 12 -o
 ↪  hq-fasta_BUSCOout -m transcriptome || error_exit "$LINENO: Error running BUSCO
 ↪  at hq-fasta"

#

 mkdir -p ./ANGEL.cds_BUSCOout || error_exit "$LINENO: directory error at
 ↪  reference-transcripts"

 /home/mahe0050/.conda/envs/BUSCOConda/bin/busco -i pygmy.ANGEL.cds__.fasta -l
 ↪  $LINEAGE -f -c 12 -o ANGEL.cds_BUSCOout -m transcriptome || error_exit
 ↪  "$LINENO: Error running BUSCO at ANGEL.cds"

#

 mkdir -p ./reference-transcripts_BUSCOout || error_exit "$LINENO: directory error
 ↪  at reference-transcripts"

 /home/mahe0050/.conda/envs/BUSCOConda/bin/busco -i
 ↪  reference_transcripts.1Lv.clean_.fasta -l $LINEAGE -f -c 12 -o
 ↪  reference-transcripts_BUSCOout -m transcriptome || error_exit "$LINENO: Error
 ↪  running BUSCO at reference-transcripts"

#
```

```
echo "all BUSCOs complete"
```

Visualisation

BUSCO outputs a file named <-o>/short_summary.specific.vertebrata_odb10.<-o>.txt These are used to plot a summary figure of all samples.

These summary files were manually copied into ./BUSCO_summaries to group the outputs. The python script generate_plot.py included in the BUSCO installation was then used using the following script to convert these scores into an R script to generate a visualisation of the output.

```
#!/bin/bash
# ### (SLURM submission truncated for brevity) Important parameters are kept:
#
#SBATCH --job-name=mahe0050_BUSCO-summmary
#
#SBATCH --time=2-0
#
#SBATCH --ntasks=1
#
#SBATCH --cpus-per-task=8
#
#SBATCH --mem-per-cpu=2G

##################################################################
# Load any modules that are required.

module load Miniconda3/4.9.2
module load python/3.9.6

##################################################################
# Enter the command-line arguments that you job needs to run.

source ~/.bashrc
conda activate BUSCOConda

module load Miniconda3/4.9.2
module load python/3.9.6


# See notes file associated with usage: Output-BUSCO_Name2020.sh and
↪  BUSCO_all_v5.0.sh
# (directory creation and summary copied manually executed before this script)


# Inputs include:
#     Short-read Trinity outputs for all 8 Kidney samples assembled from short
↪  reads
#     Short-read Trinity output for one file of all 8 kidney sequencing
↪  concatenated and assembled into a single transcript file

#     Isoseq3 long-read full list of non-redundant hq transcripts
#     Isoseq3 long-read full list of transcript predicted open read frame coding
↪  regions
```

```
#     Isoseq3 long-read representative transcripts of 'putative genes' based on
↪  coding sequence protein clustering, but including full-length including UTRs,
↪  with additional poly-a tail trimming


#------adjust these for your run-----

LINEAGE="vertebrata_odb10"

#-----------------------------------

function error_exit
{
echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
exit 1
}


#go to the directory containing trimmed files to pull ID names
cd /scratch/user/mahe0050/BUSCO/

#mkdir BUSCO_summaries  || error_exit "$LINENO: dir error 1"

#cp G1_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G1_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G2_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G2_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G3_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G3_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G4_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G4_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G5_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G5_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G6_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G6_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G7_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G7_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp G8_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G8_KI_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp
↪  reference-transcripts_BUSCOout/short_summary.specific.vertebrata_odb10.reference-transcripts_
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp
↪  ANGEL.cds_BUSCOout/short_summary.specific.vertebrata_odb10.ANGEL.cds_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp hq-fasta_BUSCOout/short_summary.specific.vertebrata_odb10.hq-fasta_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"
#cp
↪  trinity-all_BUSCOout/short_summary.specific.vertebrata_odb10.trinity-all_BUSCOout.txt
↪  BUSCO_summaries/.  || error_exit "$LINENO: copy error"

# (The path below is where the config files for conda-BUSCO environment are
↪  located)
```

```
python3 /home/mahe0050/.conda/envs/BUSCOConda/bin/generate_plot.py -wd
 ↪  /scratch/user/mahe0050/BUSCO/BUSCO_summaries  || error_exit "$LINENO: BUSCO
 ↪  error all"

echo "'all' BUSCO summary complete"

# *** *** ***

#mkdir BUSCO_SR_summaries  || error_exit "$LINENO: dir error 2"

#cp G1_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G1_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G2_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G2_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G3_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G3_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G4_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G4_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G5_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G5_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G6_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G6_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G7_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G7_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp G8_KI_BUSCOout/short_summary.specific.vertebrata_odb10.G8_KI_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"
#cp
 ↪  trinity-all_BUSCOout/short_summary.specific.vertebrata_odb10.trinity-all_BUSCOout.txt
 ↪  BUSCO_SR_summaries/.  || error_exit "$LINENO: copy error"

python3 /home/mahe0050/.conda/envs/BUSCOConda/bin/generate_plot.py -wd
 ↪  /scratch/user/mahe0050/BUSCO/BUSCO_SR_summaries  || error_exit "$LINENO: BUSCO
 ↪  error SR"

echo "short read BUSCO summary complete"

# *** *** ***

#mkdir BUSCO_LR_summaries  || error_exit "$LINENO: dir error 3"

#cp
 ↪  reference-transcripts_BUSCOout/short_summary.specific.vertebrata_odb10.reference-transcripts_
 ↪  BUSCO_LR_summaries/.  || error_exit "$LINENO: copy error"
#cp
 ↪  ANGEL.cds_BUSCOout/short_summary.specific.vertebrata_odb10.ANGEL.cds_BUSCOout.txt
 ↪  BUSCO_LR_summaries/.  || error_exit "$LINENO: copy error"
#cp hq-fasta_BUSCOout/short_summary.specific.vertebrata_odb10.hq-fasta_BUSCOout.txt
 ↪  BUSCO_LR_summaries/.  || error_exit "$LINENO: copy error"

python3 /home/mahe0050/.conda/envs/BUSCOConda/bin/generate_plot.py -wd
 ↪  /scratch/user/mahe0050/BUSCO/BUSCO_LR_summaries  || error_exit "$LINENO: BUSCO
 ↪  error LR"
```

```
echo "long read BUSCO summary complete"
```

(the ./BUSCO_SR_summaries and ./BUSCO_LR_summaries were generated so that short reads and long read data could be visualised separately but were ultimately not used)

These files were downloaded moved into an R working directory to generate the figure.
The R script produced by the python command above was edited to allow for local directory structure and streamline the labelling and order of sample names.

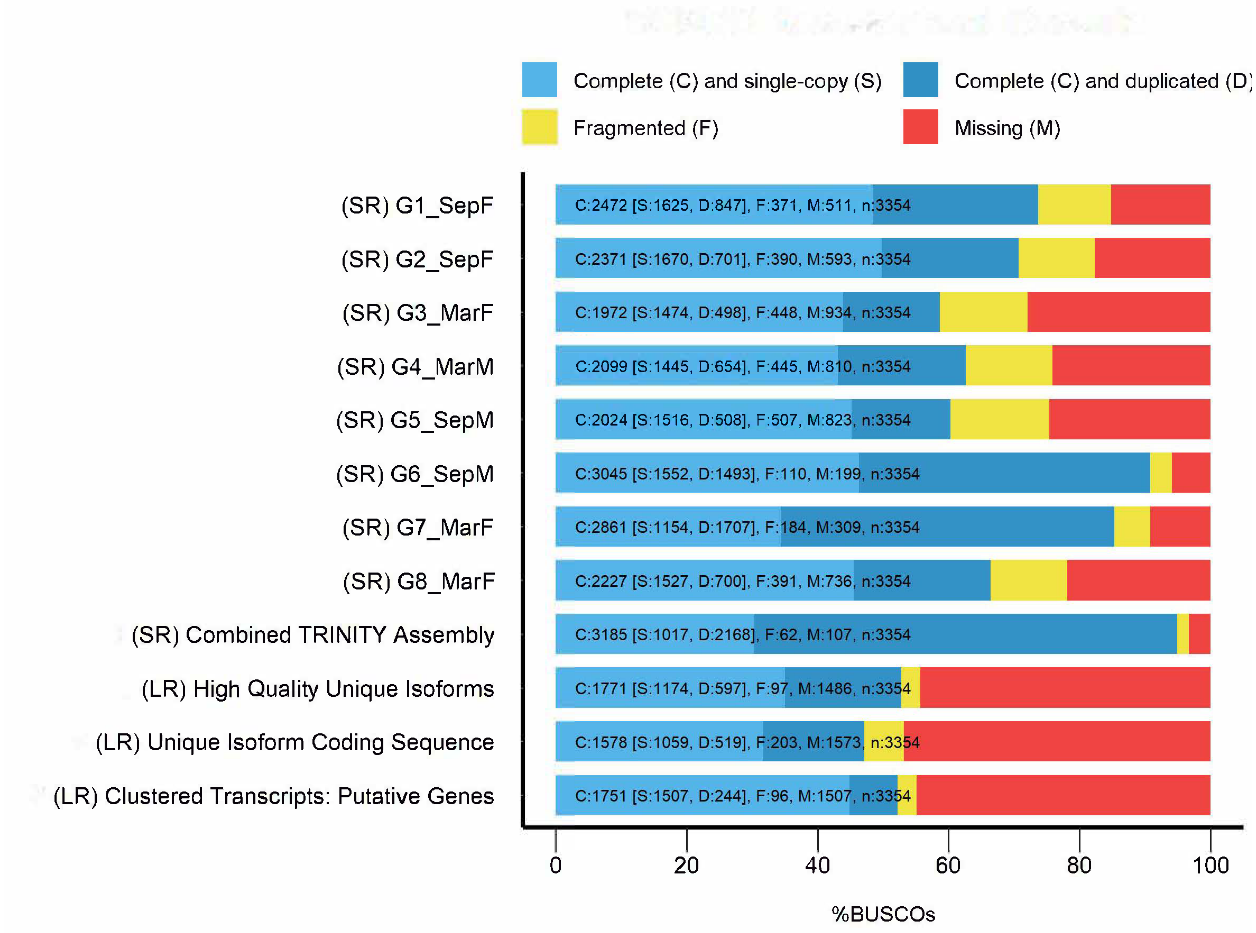


Figure 10: Benchmarking Universal Single-Copy Orthologs (BUSCO) as a measure of completeness for transcript assemblies of *Tiliqua adelaidensis* kidney tissue mRNA. Bar proportions are shown in % of BUSCO library, true values are provided in text. Transcripts compared against lineage dataset vertebrata_odb10 which includes 3354 BUSCOs for vertebrate species. Short-read de novo TRINITY RNA transcript assemblies are indicated by (SR) and long-read data by (LR). BUSCO sequence scores C=completed, S=single, D=duplicated, F=fragmented, M=missing, n=total number of BUSCOs in database. Short-read individual samples are identified based on lizard ID, Season, and Sex: Sep=September (spring), Mar=March (autumn).

4.2 BWA Alignment of combined TRINITY short-read assembly of all individuals to long-read reference transcripts was run using the following script on Flinders Deep Thought HPC

```
#!/bin/bash
# ### (SLURM submission truncated for brevity) Important parameters are kept:
#
#SBATCH --job-name=mahe0050_bwa
#
#SBATCH --time=0-12
#
#SBATCH --ntasks=1
#
#SBATCH --cpus-per-task=1
#
#SBATCH --mem-per-cpu=4G

##################################################################
# Load any modules that are required.

module load Miniconda3/4.9.2
module load SAMtools/1.12-GCC-10.3.0


##################################################################
# Enter the command-line arguments that you job needs to run.

source ~/.bashrc
conda activate BwaConda

#Script input:

# This script is used to run BWA-MEM2 to align the concatenated TRINITY assembly to
↪  the long-read Iso-seq transcripts generated for T adelaidensis.
# The paths in the script assume that a specific directory structure has been set
↪  up.
#
# Modules required:
# module load Miniconda3/4.9.2
# module load SAMtools/1.12-GCC-10.3.0

# usage through slurm
#
# Carmel Maher
# Aug 2023

#-------------------------------------
#Run from within BwaConda environment

#module load Miniconda3/4.9.2
#conda create --name BwaConda
#conda activate BwaConda (this is input in every SLURM script)
```

```
#Required Installations:
#conda install -c bioconda bwa-mem2


#-----------------------------------
function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}


#Create output folder
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/
mkdir ./BWA || error_exit "$LINENO: Directory Error 0"
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/BWA


#
# Indexing the reference sequence (Requires 28N GB memory where N is the size of
↪  the reference sequence).
/home/mahe0050/.conda/envs/BwaConda/bin/bwa-mem2 index -p IsoRef
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.clean.fasta
↪  || error_exit "$LINENO: Index error"


# Mapping
# Run "./bwa-mem2 mem" to get all options
/home/mahe0050/.conda/envs/BwaConda/bin/bwa-mem2 mem -t 8 IsoRef
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
↪  | samtools sort -@8 -o trinity-all_sortedbam.bam - || error_exit "$LINENO:
↪  bwa-mem error"

samtools flagstat trinity-all_sortedbam.bam

samtools index trinity-all_sortedbam.bam

echo "job complete"
```

Inspection of the output of the Burrows Wheeler Alignment:

```
head trinity-all_out.sam

tail trinity-all_out.sam
```

```
#the middle:
sed -n 18000,18010p trinity-all_out.sam
```

flagstat file shows 46.11% mapped

```
402933 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
43736 + 0 supplementary
0 + 0 duplicates
185795 + 0 mapped (46.11% : N/A)
0 + 0 paired in sequencing
0 + 0 read1
0 + 0 read2
0 + 0 properly paired (N/A : N/A)
0 + 0 with itself and mate mapped
0 + 0 singletons (N/A : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
job complete
```

4.3 BLASTn comparison of combined TRINITY short-read assembly of all individuals to long-read reference transcripts

This BLASTn used the short-read assembled transcripts as the query sequences and the long-read reference transcripts as the BLAST database. BLASTn searches were conducted using the following script, and the best hit for the query with parameters max hits per sequence 5, and max target sequences 5 was extracted manually.

```
#!/bin/bash
# ### (SLURM submission truncated for brevity) Important parameters are kept:
#
#SBATCH --job-name=mahe0050_BLAST-iso
#
#SBATCH --time=0-12
#
#SBATCH --ntasks=1
#
#SBATCH --cpus-per-task=1
#
#SBATCH --mem-per-cpu=4G

####################################################################
# Load any modules that are required.

module load BLAST+/2.12 #BLAST+:


####################################################################
# Enter the command-line arguments that you job needs to run.

source ~/.bashrc

#Script input:

# This script is used to run BLASTn to compare the concatenated TRINITY assembly
↪  (as the query) to the long-read Iso-seq transcripts (as a database) generated
↪  for T adelaidensis.
# The paths in the script assume that a specific directory structure has been set
↪  up.
#
# Modules required:
# module load BLAST+/2.12 #BLAST+:

# usage through slurm
#
# Carmel Maher
# Aug 2023


#-------------------------------------
function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
```

```
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}


#Create output folder
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/
#mkdir ./BLAST || error_exit "$LINENO: Directory Error 0"
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST

#reference to index
#
↪   /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.clean.fasta

#assembled transcripts to query
#
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta


# BLAST does not like '|' in FASTA database headers
# simplify the headers
sed 's/|/-/g'
↪   /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.clean.fasta
↪   >
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST/reference_transcripts.1Lv.c-nopipe.fas
↪   || error_exit "$LINENO: sed error1"
sed 's/path/p/g'
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST/reference_transcripts.1Lv.c-nopipe.fas
↪   >
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST/reference_transcripts.1Lv.c-nopipep.fa
↪   || error_exit "$LINENO: sed error2"
sed 's/transcript/t/g'
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST/reference_transcripts.1Lv.c-nopipep.fa
↪   >
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST/reference_transcripts.1Lv.c-nopipept.f
↪   || error_exit "$LINENO: sed error3"

# Indexing the reference sequence
# makeblastdb -in [input database] -out [output database] -dbtype [database type]

makeblastdb -in
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST/reference_transcripts.1Lv.c-nopipept.f
↪   -parse_seqids -dbtype nucl -out Reference_cluster_db || error_exit "$LINENO:
↪   Index error"


# BLASTn
# blastn -query [query file] -db [database file] -out [output file] -num_threads 4
↪   -outfmt 6 -evalue 1e-5 -max_target_seqs 1 -max_hsps 5

blastn -query
↪   /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
↪   -db Reference_cluster_db -out Trinity-all_IsoSeq_BLASTn_t5-h5 -num_threads 4
↪   -outfmt 6 -evalue 1e-5 -max_target_seqs 5 -max_hsps 5 || error_exit "$LINENO:
↪   BLAST error"
```

```
blastn -query
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
↪  -db Reference_cluster_db -out Trinity-all_IsoSeq_BLASTn_t5-h5 -num_threads 4
↪  -outfmt 6 -evalue 1e-5 -max_target_seqs 1 -max_hsps 5 || error_exit "$LINENO:
↪  BLAST error"

blastn -query
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
↪  -db Reference_cluster_db -out Trinity-all_IsoSeq_BLASTn_t1-h1 -num_threads 4
↪  -outfmt 6 -evalue 1e-5 -max_target_seqs 1 -max_hsps 1 || error_exit "$LINENO:
↪  BLAST error"

echo "job complete"


# ----------
#Ran separately by commenting parts of code out:

#To sort BLASTn output for single best hit per query decided to use output from
↪  -evalue 1e-5 -max_target_seqs 5 -max_hsps 5 and sort manually
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/BLAST
sort -k1,1 -k11,11g Trinity-all_IsoSeq_BLASTn_t5-h5 | sort --merge -u  -k1,1 -o
↪  Trinity-all_IsoSeq_BLASTn_t5-h5_Efilt.txt || error_exit "$LINENO: sort 1 error"
#This will sort by query sequence and p-value (assuming default format 6 output),
↪  then will select all first unique results (lowest p-value) for each query

#Then sort by target sequence, in order to easily pull out groups of transcripts
↪  that match the same reference for visualisation
sort -k2,2 Trinity-all_IsoSeq_BLASTn_t5-h5_Efilt.txt -o
↪  Trinity-all_IsoSeq_BLASTn_t5-h5_Efilt-Tsort.txt || error_exit "$LINENO: sort 2
↪  error"

To determine how many target sequences received a hit:
cut -f 2 Trinity-all_IsoSeq_BLASTn_t5-h5 | sort | uniq | wc -l

# ----------
```

4.4 Extraction of query (assembled short-read) sequences with common BLASTn target hits (from long-read set)

The following script is an example of extraction of BLASTn hits and reference sequences using Seqtk. Note that the headers had to be truncated for nput into BLAST and these were manually corrected in the .lst file when groups were manually extracted.

This job was also run on Flinders Deep thought HPC, with 1 task per CPU, 1 CPU requested with 2GB memory.

```
module load Miniconda3/4.9.2

#------------------------------------
#Run from within BwaConda environment

#module load Miniconda3/4.9.2
#conda create --name SeqConda
#conda activate SeqConda (this is input in every SLURM script)

#Required Installations:
#conda install -c bioconda seqtk
# seqtk-1.4

#------------------------------------

##################################################################
# Enter the command-line arguments that you job needs to run.

source ~/.bashrc
conda activate SeqConda

#Script input:

# This script is used to sort BLASTn outputs and to use Seqtk to extract example
↪  clusters of transcripts which had a best match to the same target sequence, as
↪  well as extract example Clustered isoforms from the full reference database for
↪  visualisation of transcript clusters and alignments.
# The paths in the script assume that a specific directory structure has been set
↪  up.
#y
# Modules required:
# module load Miniconda3/4.9.2

# usage through slurm
#
# Carmel Maher
# Aug 2023

#------------------------------------
function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message
```

```
    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}


#Create output folder
#cd /scratch/user/mahe0050/DE-analysis/2_alignedData
#mkdir ./seqtk || error_exit "$LINENO: Directory Error 0"
#cd /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtk

# lists of sequence IDs for clusters and target sequences were created manually in
↪  notepad:


#Extract sequences with names in file name.lst, one sequence name per line: seqtk
↪  subseq in.fq name.lst > out.fq


#ID files to extract were placeed in
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtk

# Files to sub-set
#
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/hq.fasta.no5merge.collapsed.rep.1L.fa
#
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.1L.Trinity.fas
↪  #as created below

# ----------
#BLASTn List file (manually created - note fasta headers were simplified for
↪  BLASTn, these were also manually changed back)
#
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtkBLAST/PB.100.1-049515-p5.lst
# ----------



#To subseq BLASTn comparison of Trinity & Isoseq - The Isoseq reference must be
↪  pulled out, as well as the matching Trinity query sequences

# NOTE the existing Trinity-Concatenated output is on multiple lines fasta
# head
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.Trinity.fasta
# >TRINITY_DN8_c0_g1_i1 len=9122 path=[0:0-6445 1:6446-7041 2:7042-7418
↪  4:7419-9121]
#TTTTTTGAGGTACACTGGCTTTTATTCCATGCATCTGTTCATTTTGATAGTCAGTCTTGT
#CCCCCATTGTTACACACTGTTCAGTACAAATAAAAAAAATCAAGTTTGCCTCAGGAAGCC
#ATAGAGTAGGAAGGAGAGAGAAAGACAGCACAGGAGATTAAATGATAATTTCACATGATA
#TGAAACCAATAATCGCCATTAATTGCAACAATGATTGTGCTATACTGGTCCCAGTAGTTA
#AGAGTAGGTATGAGTTATGAGTAATGAAAGTATCGCAAAGACGCTACTTGCCTCCAGAGC
#ATCTTATCCAGTTTGCATTGCTGATTCTAAGCATGCTCCAATTCATTGTGCCCTTATTG
#CAAACCCTTAATCAGGCAGTTATAATAATTGTAATGCCCTTACATAATAGTTAAACAGT
```

```
#AGATTCACCATTGATAACTGCACACTTGGGCAATACTGTCTCAATCTGAATGAGATGAT
#TTGTTTGGTTTATGGTTTTCAACTAAAAAGCATGAGATTTGTTCTAATAGACTCTGCAAT

#To convert to single line:
cd /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2
seqtk seq trinity-all.Trinity.fasta > trinity-all.1L.Trinity.fasta  || error_exit
↪  "$LINENO: Trinity1L error"

# To subset Trinity BLASTn examples

cd /scratch/user/mahe0050/DE-analysis/2_alignedData/seqtkBLAST

#Reference

seqtk subseq
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/hq.fasta.no5merge.collapsed.rep.1L.fa
↪  PB.100.1-049515-p5_Ref.lst > PB.100.1-049515-p5_Ref.fa || error_exit "$LINENO:
↪  PB.100.1-049515-p5_Ref error"

#Query hits

seqtk subseq
↪  /scratch/user/mahe0050/DE-analysis/2_alignedData/trinity-all/data2/trinity-all.1L.Trinity.fas
↪  PB.100.1-049515-p5.lst > PB.100.1-049515-p5.fa || error_exit "$LINENO:
↪  PB.100.1-049515-p5 error"

#they can be imported together into a program for visualisation later as separate
↪  files

echo "trinity subseq done"

# ----------
```

5 Gene Expression Analysis

5.0.0.0.1 Long-reads (Reference Index) - Previous Analysis:

- Isoseq3 pipe
- Collapsing of isoforms & generating fasta of longest transcripts in predicted ORFs (Cogent/ANGEL)
- BLASTx search of coding sequence file for the longest transcripts in predicted ORFs
- Grouping of transcript clusters/putative gene families (CD-HIT)
- Longest representative transcript collapsed into a reference .fasta (R & Shell scripts)
- As this longest transcript was taken from earlier in the pipeline, persisting poly-a tails were trimmed from the reference .fasta file (Perl script)

5.0.0.0.2 Short-reads - Previous Analysis:

- FASTQC quality assessment
- Cutadapt adapter and quality trimming
- FASTQC quality assessment (note: further TRINITY assembly not used here for counts)

5.0.0.0.3 Both Datasets are Combined in this Analysis:

- Transcript counts per million estimated using Kallisto
    - Clean short-reads (above) counted against reference index generated from long-read .fasta (above)
- Kallisto output imported into R for analysis using EdgeR

5.1 Kallisto

Short-reads have already been trimmed. These cleaned sequences were imported directly into Kallisto.

Kallisto was run on the Flinders' HPC Deep Thought and files were moved accordingly, keeping directories relative locations intact.

Kallisto version version 0.48.0 was installed using Miniconda3/4.9.2 (in the same environment as Cutadapt used previously) as below

```
# Open Deep Thought and navigate to the /scratch disc where all data and scripts
↪  are located.
cd /scratch/user/mahe0050

#Load Miniconda V4.9.2
module load Miniconda3/4.9.2

# Create a conda environment "ShortConda"
#conda create -n ShortConda

## Package Plan ##
# environment location: /home/mahe0050/.conda/envs/ShortConda
#
# To resolve init error initialise conda with bash (already an available shell,
↪  when completed states no change to paths etc.
#     $ conda init bash
#
```

```
# To activate this environment, use
#     $ conda activate ShortConda
#
# To deactivate an active environment, use
#     $ conda deactivate


#Activate this environment
source ~/.bashrc
conda activate ShortConda

#Check bioconda channels:
conda config --add channels bioconda
conda config --add channels conda-forge
conda config --add channels defaults

# cutadapt version 4.1
conda install -c bioconda cutadapt

# kallisto version 0.48.0
conda install kallisto

# ***
# Proceed with Kallisto
# ***

cd /scratch/user/mahe0050/DE-analysis/bash
source ~/.bashrc
conda activate ShortConda

#File: kallisto-clustr-clean.sh
#uses the trimmed reads to q 5

# called in SLURM script:
sbatch Slurm-Kallisto_DT-2022.sh


# ***
# Kallisto outputs are then analysed in R studio
# ***
```

The file reference_transcripts.1Lv.clean.fasta was used as the index.
All of this was achieved using the following

The following Slurm script was used to run kallisto-clstr-clean.sh as below.

Creation of the index is 'commented out' in the below script as it only needs to be run once and was completed during troubleshooting.

Run kallisto:

```
#!/bin/bash
#
#
```

```
# This script is used to estimate transcript counts using Kallisto - using HiSeq
↪  reads against an IsoSeq reference.
# This reference was created from Iso-Seq data, clustered by CDHit (based on
↪  pygmy.ANGEL.pep, cutoff at .99, job 1598936109) and subset to include the
↪  representative transcript for each cluster using dplyR and seqtk.
# The paths in the script assume that a specific directory structure has been set
↪  up.
# Kallisto must be run on paired samples separately
#
# usage on flinder's Deep Thought Machine via SLURM
# usage from within: conda activate ShortConda
#
# Carmel Maher & Terry Bertozzi
# Dec 2019, last altered Oct 2022


#------adjust these for your run-----

# module load Miniconda3/4.9.2
# conda activate ShortConda
# conda install kallisto
# kallisto, version 0.48.1


# ReferenceDIR=/scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk #contains
↪  reference_transcripts.1Lvclean.fasta
# KallistoEXE=/mnt/Prog/miniconda3/envs/anaKallisto/bin
KallistoDIR=/scratch/user/mahe0050/DE-analysis/2_alignedData/kallisto-clstr-cleanq
ReadDIR=/scratch/user/mahe0050/DE-analysis/1_trimmedData/q #contains trimmed R1, R2

#------------------------------------

function error_exit
{
    # Exit function due to fatal error
    # Accepts 1 arg:
    # string - descriptive error message

    echo "${PROGNAME}: ${1:-"Unknown error"}" 1>&2
    exit 1
}

# *** make the index: ***
 cd /scratch/user/mahe0050/DE-analysis/2_alignedData/kallisto-clstr-cleanq

#create the index
#the below contains the representative transcript for clustered isoforms with
↪  (most) of the poly-a tails trimmed:
# kallisto index -i reference_transcripts.1Lv.clean.idx
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.clean.fasta
↪  || error_exit "$LINENO: kallisto index error"

# ------------
```

```
## [build] loading fasta file
↪  /scratch/user/mahe0050/IsoSeq-analysis/data/Seqtk/reference_transcripts.1Lv.clean.fasta
## [build] k-mer length: 31
## [build] counting k-mers ... done.
## [build] building target de Bruijn graph ...  done
## [build] creating equivalence classes ...  done
## [build] target de Bruijn graph has 59186 contigs and contains 17770125 k-mers

# ------------


# need to be in the input directory for for file
cd $ReadDIR

for file in *R1_cleanq.fq.gz
do
  FILESTEM=${file%_*}
  #this FILESTEM only cuts to _clean, the _R1 is included
  FILESTEM=${FILESTEM/R1/}
  #removes R1 from FILESTEM (FILESTEM ends in _ therefore not needed in "text"
  ↪  names)

echo $FILESTEM

kallisto quant -i $KallistoDIR/reference_transcripts.1Lv.clean.idx -o
↪  $KallistoDIR/$FILESTEM"kallisto-out" -b, --bootstrap-samples=100 --threads=12
↪  --pseudobam $file $FILESTEM"R2_cleanq.fq.gz" || error_exit "$LINENO: kallisto
↪  error at $FILESTEM"


done

echo "kallisto-clstr-cleanq done"
```

5.2 EdgeR Expression Analysis

Kallisto outputs were downloaded and placed into three separate R working directories for each analysis of the different number of samples as outlined in the three sections below. /RWorkingDir/R-kallisto-clstr-6&7/kallisto-clstr/ #contained folder for final 6 analysed samples /RWorkingDir/R-kallisto-clstr/kallisto-clstr/ #contained folder for all eight samples names /RWorkingDir/R-kallisto-clstr-4/kallisto-clstr/ #contained folder for four female samples, two from each season group

All sample folders within these ~/kallisto-clstr/ folders were named as the following: “G1_SepF” “G2_SepF” “G3_MarF” “G4_MarM” “G5_SepM” “G6_SepM” “G7_MarF” “G8_MarF” So that names and labels are consistent throughout this document and in figures.

Exploration for these data has been included for different groupings in order to explore the effects of sample removal, or potential effect of sex as a confounding factor. Samples G6 and G7 have ultimately been removed due to batch effects caused by sequencing runs.

The first analysis below shows the analysis across season of collection for the remaining 6 sampes and represents data included in thesis chapter 4.

An additional exploration of only the four female samples are included for comparison to explore the effect of the divergent G5 and potential effect of sexual differences - possibly influenced by the reproductive function of the squamate kidney.

Preliminary analyses of all original eight samples is included to show the reasoning behind the ultimate removal of the short reads of samples G6 and G7. Remembering that Sample G6 is the sample used to create the long-read reference.

5.3 Six *T. adelaidensis* Individuals Collected Between Two Seasonal Periods.

(These methods exclude samples G6 and G7).

This document separates samples by Seasonal group factors AND Sex Group Factors separately to explore the effects on the data.

Import the data:

```
# create paths
setwd("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr-6&7/kallisto-clstr/")
DIR <- ("~/@Uni/@Flinders University
↪  PhD/RWorkingDir/R-kallisto-clstr-6&7/kallisto-clstr/")

head(DIR)
```

```
## [1] "~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr-6&7/kallisto-clstr/"
```

```
paths <- list.dirs(path = DIR, full.names = FALSE, recursive = FALSE)
# the working dir only includes kallisto output for the
# kidneys at this time
head(paths)
```

```
## [1] "G1_SepF" "G2_SepF" "G3_MarF" "G4_MarM" "G5_SepM" "G8_MarF"
```

```
Kcaught <- catchKallisto(paths, verbose = TRUE)
```

```
## Reading G1_SepF, 9813 transcripts, 100 bootstraps
## Reading G2_SepF, 9813 transcripts, 100 bootstraps
## Reading G3_MarF, 9813 transcripts, 100 bootstraps
## Reading G4_MarM, 9813 transcripts, 100 bootstraps
## Reading G5_SepM, 9813 transcripts, 100 bootstraps
## Reading G8_MarF, 9813 transcripts, 100 bootstraps
```

```
setwd("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr-6&7")
# View(Kcaught)
```

Set the group factors. Input Sample files are listed in number order G1-G8 excluding G6 and G7. Group factors are applied to the DGElist object and are the point of comparison for expression analyses.

Season of Collection: 1 = September, 2 = March/April For the purposes of consistent labelling per group factors some April collections are referred to in the March group. Accurate collection information and dates are outlined in the methods chapter.

Individual Sex: 1 = Female, 2 = Male

```
# these factors correlate to season of collection. 1 =
# September, 2 = March.
season_group <- factor(c(1, 1, 2, 2, 1, 2))

# these factors correlate to individuals sex. 1 = Female, 2
# = Male
sex_group <- factor(c(1, 1, 1, 2, 2, 1))
```

5.3.1 Group Factor: Season

Create the EdgeR DGE list for use in subsequent analyses

```
dge_Season <- DGEList(counts = Kcaught$counts/Kcaught$annotation$Overdispersion,
    genes = Kcaught$annotation, group = season_group)
# View(dge_Season) names(dge_Season)
dge_Season
```

```
## An object of class "DGEList"
## $counts
##                                                        G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       0.00000    5.27176
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       42.09258   53.72594
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.00000    0.00000
##                                                        G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      2.475519    1.944563
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      19.475009   16.355109
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
##                                                        G5_SepM     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     131.000000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    652.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      5.315287    4.370939
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      55.741267   14.183114
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
## 9808 more rows ...
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10088007            1
```

```
## G2_SepF     1 10028560           1
## G3_MarF     2 10002306           1
## G4_MarM     2 10118645           1
## G5_SepM     1  8854300           1
## G8_MarF     2 10310696           1
##
## $genes
##                                                      Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304        1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564       2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426        2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    1965        1642.711
##                                                      Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304             1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564            1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534            2.924537
## PB.11.1|004815|path2:1-3039(+)|transcript/3426             1.425430
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401         1.000000
## 9808 more rows ...
```

Obtain Counts Per Million to standardise count data comparison

```r
myTPM_Season <- dge_Season$counts
# head(myTPM_Season)
```

5.3.2 Initial Data Exploration

Which values in myCPM are greater than 0.5? This produces a logical matrix with TRUEs and FALSEs. TRUE values are samples with > 0.5 counts in that sample per million

```r
thresh_Season <- myTPM_Season > 0.5
# This produces a logical matrix with TRUEs and FALSEs
head(thresh_Season)
```

```
##                                                      G1_SepF G2_SepF G3_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304          TRUE    TRUE    TRUE
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564         TRUE    TRUE    TRUE
## PB.10.1|004815|path1:5-1713(+)|transcript/17534        FALSE    TRUE    TRUE
## PB.11.1|004815|path2:1-3039(+)|transcript/3426          TRUE    TRUE    TRUE
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401     FALSE   FALSE   FALSE
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993          TRUE    TRUE    TRUE
##                                                      G4_MarM G5_SepM G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304          TRUE    TRUE    TRUE
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564         TRUE    TRUE    TRUE
## PB.10.1|004815|path1:5-1713(+)|transcript/17534         TRUE    TRUE    TRUE
## PB.11.1|004815|path2:1-3039(+)|transcript/3426          TRUE    TRUE    TRUE
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401     FALSE   FALSE   FALSE
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993          TRUE    TRUE    TRUE
```

Insert colour palette for following visualisations consistency

```
library(RColorBrewer)
myPalette <- c("#999999", "#F0E442", "#56B4E9", "#009E73", "#D55E00",
    "#CC79A7")
```

Plot CPM distribution with logged CPM

```
# export the plot as png
png("plotDensities_Season.png", width = 600)
unfilteredExpr_Season <- cpm(dge_Season, log = T)
plotDensities(unfilteredExpr_Season, col = myPalette, legend = TRUE)
```

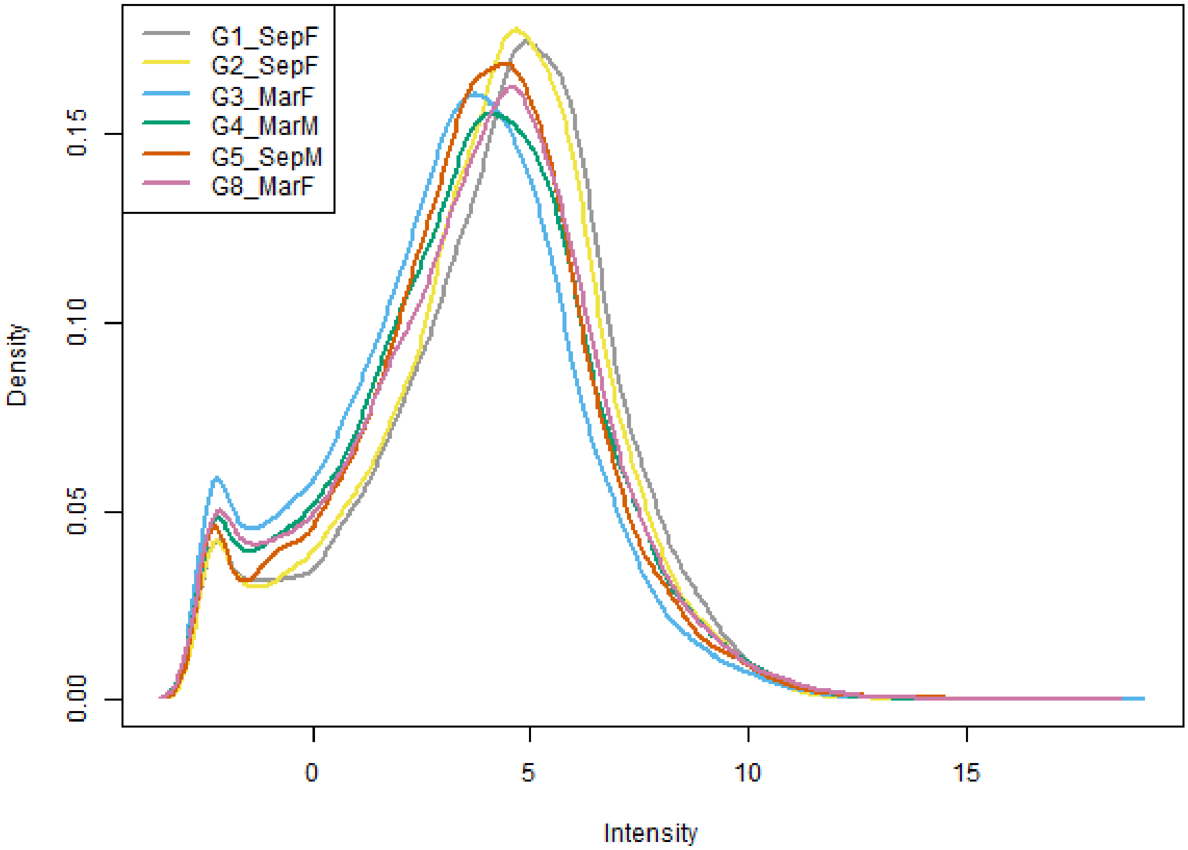


Figure 11: Log Counts per million Densities calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Visualise library sizes per sample

```
# export the plot as png
png("LibSize_Season.png", width = 600)
# The names argument tells the barplot to use the sample
# names on the x-axis The last argument rotates the axis
# names
barplot(dge_Season$samples$lib.size, names = colnames(dge_Season),
    las = 2, col = myPalette, legend = TRUE)
# title('Library size per sample')
```

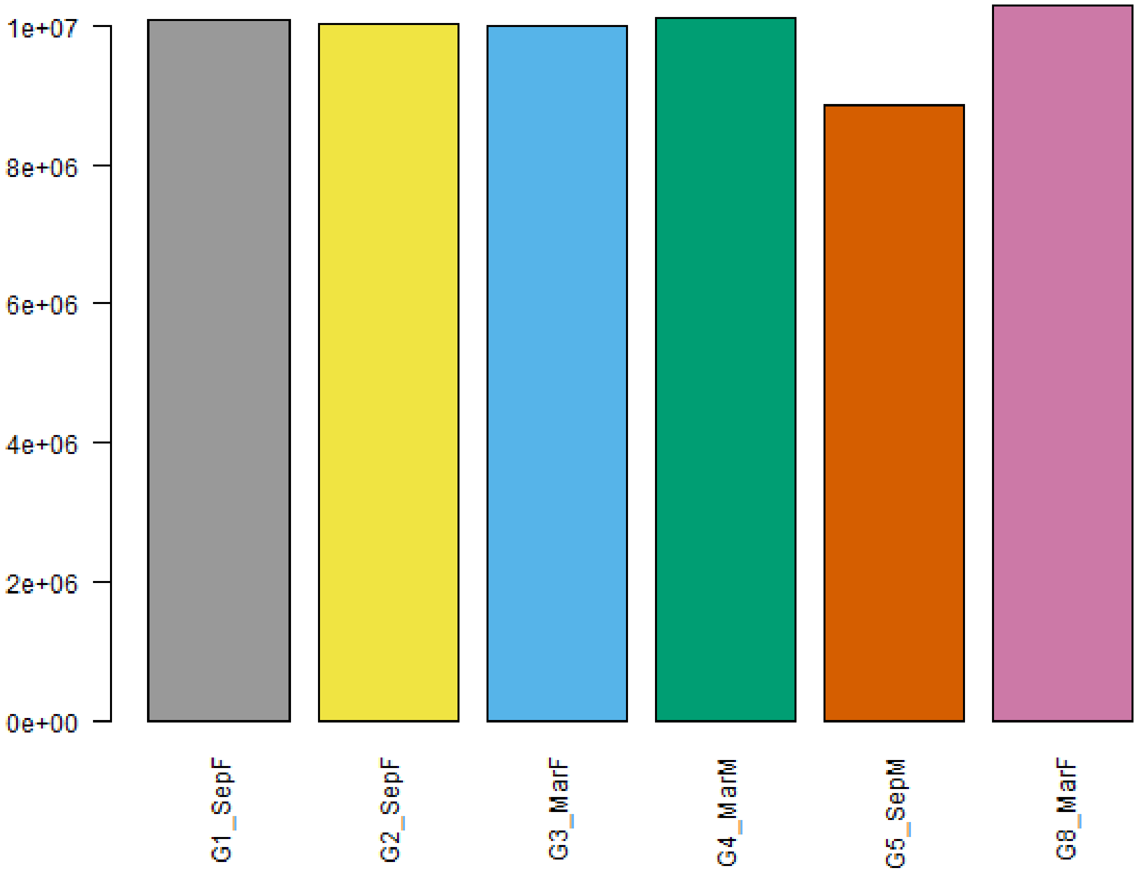


Figure 12: Library size of read counts calculated using Kallisto for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Visualise total estimated transcript counts per sample

```r
# export the plot as png
png("boxplot_Season.png", width = 600)
boxplot(dge_Season$counts, col = myPalette, las = 2, legend = TRUE)
# title('Boxplot of transcript total estimated counts')
```

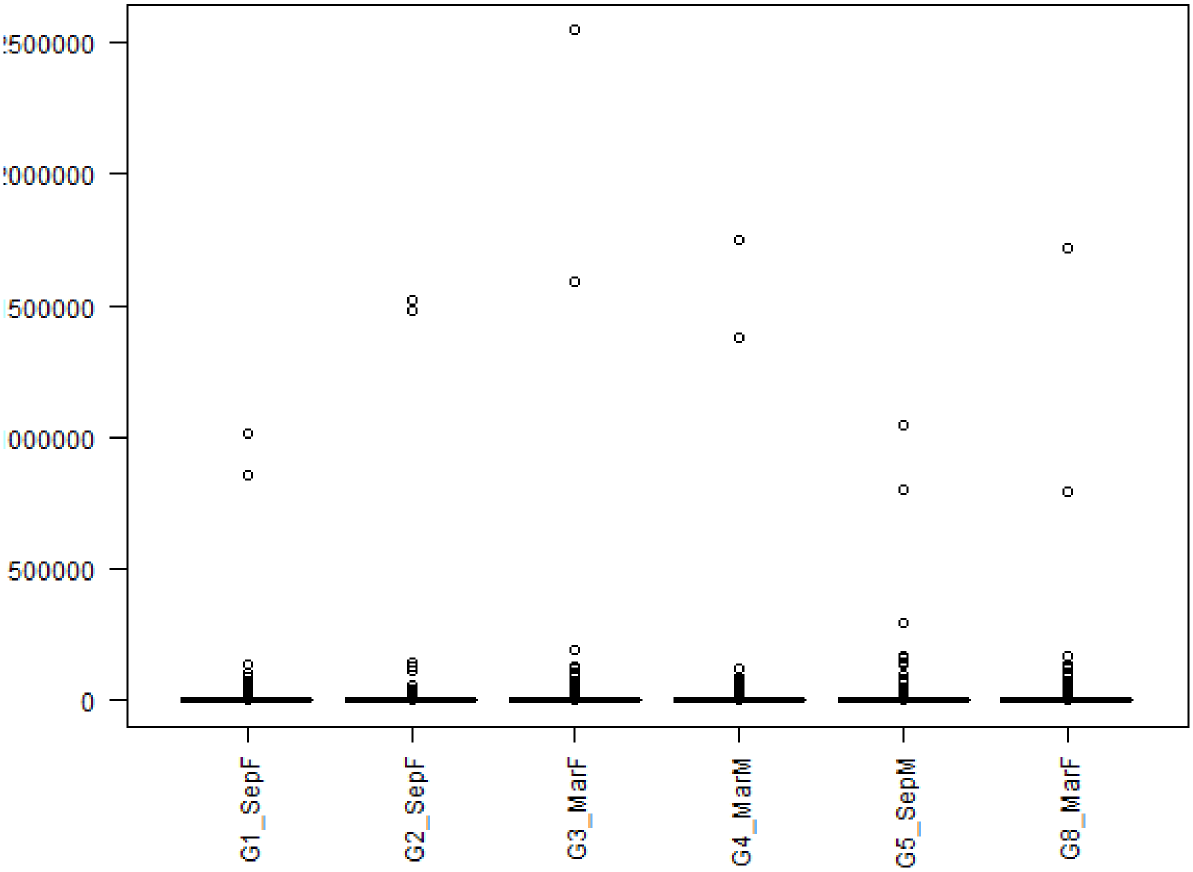


Figure 13: Total estimated Kallisto counts for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Visualise logged total estimated transcript counts per sample

```
# Get log2 counts per million
logcounts_Season <- cpm(dge_Season, log = TRUE)
# Check distributions of samples using boxplots export the
# plot as png
png("boxplot-logcounts_Season.png", width = 600)
boxplot(logcounts_Season, xlab = "", ylab = "Log2 counts per million",
    las = 2, col = myPalette, legend = TRUE)
# Let's add a blue horizontal line that corresponds to the
# median logCPM
abline(h = median(logcounts_Season), col = myPalette, legend = TRUE)
# title('Boxplots of log Counts Per Million')
```

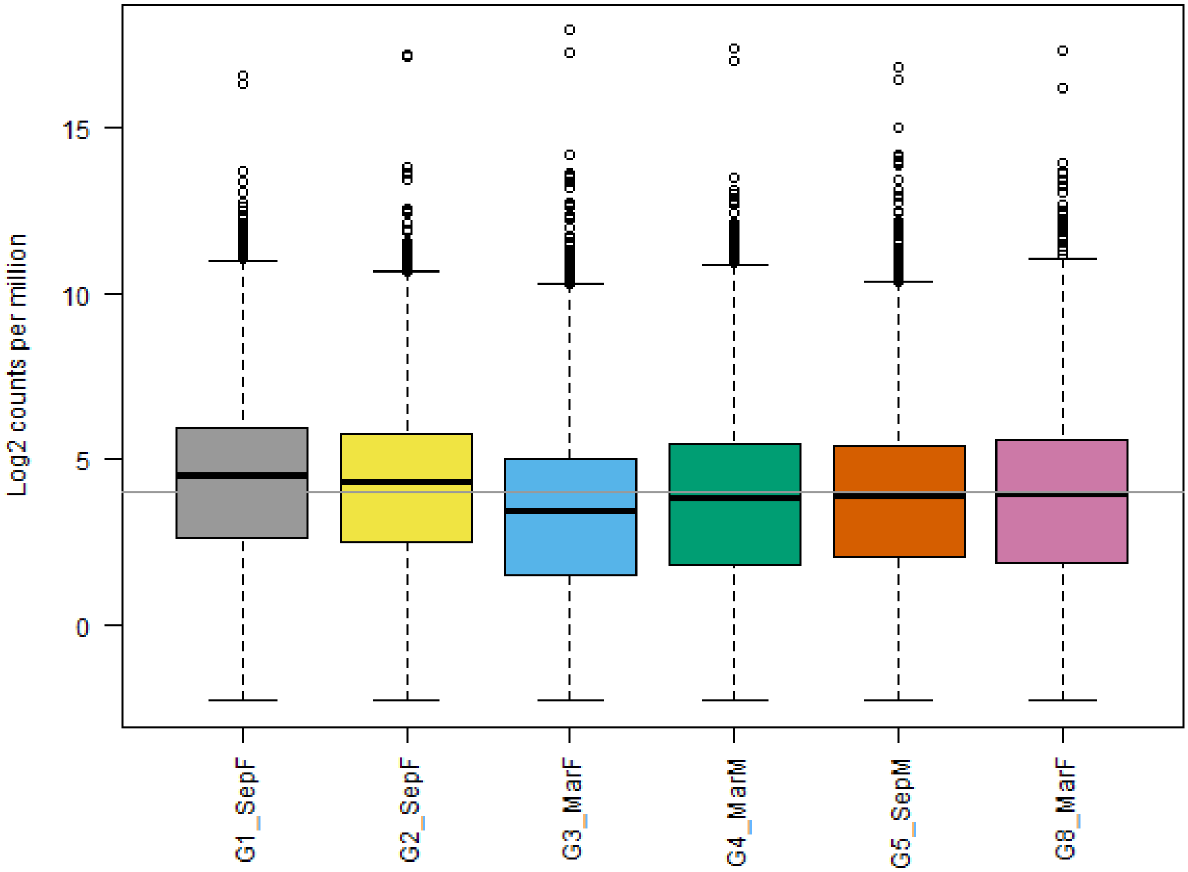


Figure 14: Log2 of estimated counts per million calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Visualise sample variation using an MDS plot

```
# export the plot as png
png("MDS_Season.png", width = 600)
plotMDS(dge_Season, col = myPalette)
```

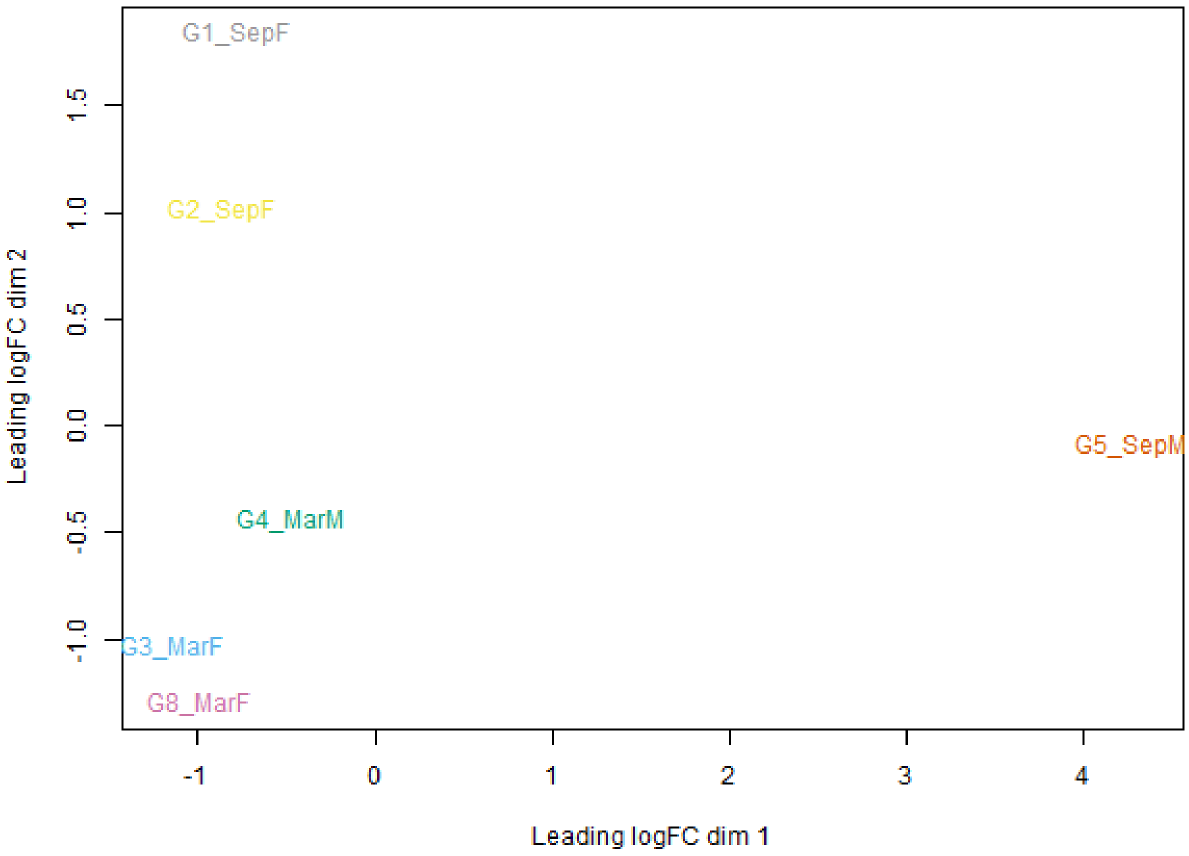


Figure 15: MDS plot of variation among six *T. adelaidensis* individuals, Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Prepare data for a HeatMap (based off of this document)

Estimate the variance for each row in the logcounts matrix

```r
var_genes_season <- apply(logcounts_Season, 1, var)
head(var_genes_season)
```

```
##     PB.2.1|002537|path0:1-1624(+)|transcript/18304
##                                       2.655830e-01
##    PB.3.2|00361c|path0:43-2240(+)|transcript/10564
##                                       2.898334e-01
##    PB.10.1|004815|path1:5-1713(+)|transcript/17534
##                                       5.406861e-01
##     PB.11.1|004815|path2:1-3039(+)|transcript/3426
##                                       7.830272e-01
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401
##                                       5.679799e-30
##     PB.13.1|004c10|path2:1-2811(+)|transcript/4993
##                                       7.170383e-01
```

Transcript IDs for the top 250 most variable transcripts

```r
select_var_decT_season <- names(sort(var_genes_season, decreasing = TRUE))[1:250]
head(select_var_decT_season)
```

```
## [1] "PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183"
## [2] "PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227"
## [3] "PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628"
## [4] "PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129"
## [5] "PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746"
## [6] "PB.225.3|089810|path3:70-2159(+)|transcript/17955"
```

Subset logcounts matrix

```r
highly_variable_lcpm_season <- logcounts_Season[select_var_decT_season,
    ]
dim(highly_variable_lcpm_season)
```

```
## [1] 250   6
```

```r
head(highly_variable_lcpm_season)
```

```
##                                                         G1_SepF     G2_SepF
## PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183   -2.3074902 -2.30749015
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227   0.4358760 -0.04576065
## PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628    -0.4974824 -0.79112517
## PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129   1.8226083  1.33598440
## PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746  7.2925047 -1.35300458
## PB.225.3|089810|path3:70-2159(+)|transcript/17955    1.2764962  1.41657483
##                                                         G3_MarF    G4_MarM
```

```
## PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183   -2.3074902 -2.3074902
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227  -0.8297537 -0.2700215
## PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628    -0.3820039 -1.1500597
## PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129   1.6575834  0.8374349
## PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746 -1.4534031  5.6538634
## PB.225.3|089810|path3:70-2159(+)|transcript/17955    1.4289610  1.4493353
##                                                        G5_SepM    G8_MarF
## PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183    9.638618 -2.3074902
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227  10.284891 -0.7899420
## PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628     9.944293 -0.1842004
## PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129  11.938968  1.7547910
## PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746  6.011154  7.6252538
## PB.225.3|089810|path3:70-2159(+)|transcript/17955   11.556590  1.0016701
```

Reorder columns to cluster Season factor groups, and sexes

```
col.order_season <- c("G1_SepF", "G2_SepF", "G5_SepM", "G3_MarF",
    "G8_MarF", "G4_MarM")
highly_variable_lcpm_season <- highly_variable_lcpm_season[,
    col.order_season]
```

```
head(highly_variable_lcpm_season)
```

```
##                                                         G1_SepF     G2_SepF
## PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183   -2.3074902 -2.30749015
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227   0.4358760 -0.04576065
## PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628    -0.4974824 -0.79112517
## PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129   1.8226083  1.33598440
## PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746  7.2925047 -1.35300458
## PB.225.3|089810|path3:70-2159(+)|transcript/17955    1.2764962  1.41657483
##                                                        G5_SepM    G3_MarF
## PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183    9.638618 -2.3074902
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227  10.284891 -0.8297537
## PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628     9.944293 -0.3820039
## PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129  11.938968  1.6575834
## PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746  6.011154 -1.4534031
## PB.225.3|089810|path3:70-2159(+)|transcript/17955   11.556590  1.4289610
##                                                         G8_MarF    G4_MarM
## PB.4280.1|a059b0|path16:1-2684(+)|transcript/6183   -2.3074902 -2.3074902
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227  -0.7899420 -0.2700215
## PB.3026.2|6fad6f|path6:3-799(+)|transcript/23628    -0.1842004 -1.1500597
## PB.2910.2|6c046b|path2:22-1784(+)|transcript/16129   1.7547910  0.8374349
## PB.4554.2|aa4354|path10:39-2063(+)|transcript/13746  7.6252538  5.6538634
## PB.225.3|089810|path3:70-2159(+)|transcript/17955    1.0016701  1.4493353
```

Colour pallete

```
mypalette2 <- brewer.pal(11, "RdYlBu")
morecols <- colorRampPalette(mypalette2)
```

Colour vector for season_group factor variable: note - factor groups changed to match column reordering in highly_variable_lcpm September=1 March=2

```r
Reordered_season_group <- factor(c(1, 1, 1, 2, 2, 2))
col.cell_season <- c("chartreuse3", "orange")[Reordered_season_group]
```

Create the HeatMap

```r
png(file = "High_var_genes.heatmap_season.png", width = 7.5 *
    300, height = 5 * 300, res = 300, pointsize = 8)
# 5 x 300 pixels height = 5*300, res = 300,# 300 pixels per
# inch
heatmap.2(highly_variable_lcpm_season, col = rev(morecols(50)),
    Colv = FALSE, trace = "none", main = "Top 250 most variable genes across samples",
    ColSideColors = col.cell_season, scale = "row", margins = c(8,
        4), srtCol = 45, labRow = "", dendrogram = c("none"))
```

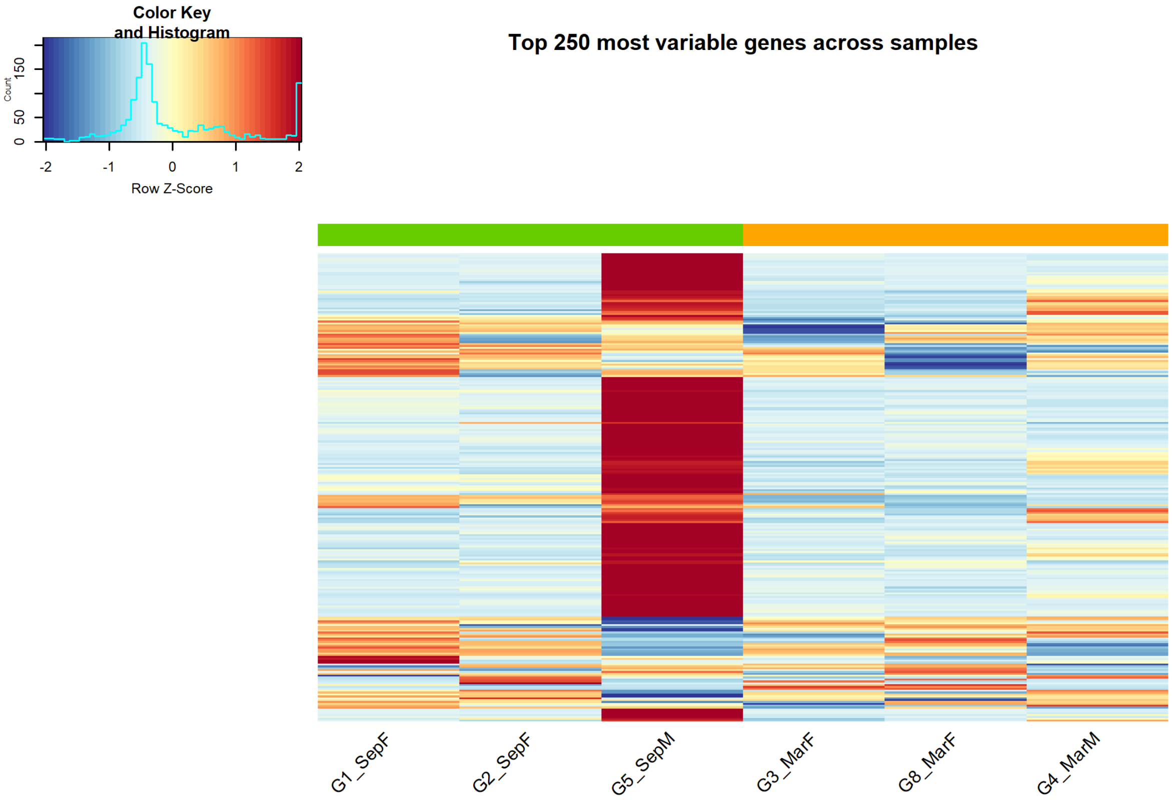


Figure 16: Top 250 most variable transcripts among six *T. adelaidensis* individuals, based on total counts per million values calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Transcript IDs for the bottom 250 most variable transcripts = i.e the least variable transcripts

```
select_var_decF_season <- names(sort(var_genes_season, decreasing = FALSE))[1:250]
head(select_var_decF_season)
```

```
## [1] "PB.11.2|004815|path2:1047-3035(+)|transcript/13401"
## [2] "PB.130.4|056771|path1:13-5910(+)|transcript/2903"
## [3] "PB.159.1|068d5c|path0:1-2199(+)|transcript/10689"
## [4] "PB.253.3|092e23|path2:4-1403(+)|transcript/20322"
## [5] "PB.281.1|0a5144|path2:1-1582(+)|transcript/18660"
## [6] "PB.347.1|0ca129|path6:1-490(+)|transcript/24817"
```

Subset logcounts matrix

```
low_variable_lcpm_season <- logcounts_Season[select_var_decF_season,
    ]
dim(low_variable_lcpm_season)
```

```
## [1] 250   6
```

```
head(low_variable_lcpm_season)
```

```
##                                                     G1_SepF  G2_SepF  G3_MarF
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401 -2.30749 -2.30749 -2.30749
## PB.130.4|056771|path1:13-5910(+)|transcript/2903   -2.30749 -2.30749 -2.30749
## PB.159.1|068d5c|path0:1-2199(+)|transcript/10689   -2.30749 -2.30749 -2.30749
## PB.253.3|092e23|path2:4-1403(+)|transcript/20322   -2.30749 -2.30749 -2.30749
## PB.281.1|0a5144|path2:1-1582(+)|transcript/18660   -2.30749 -2.30749 -2.30749
## PB.347.1|0ca129|path6:1-490(+)|transcript/24817    -2.30749 -2.30749 -2.30749
##                                                     G4_MarM  G5_SepM  G8_MarF
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401 -2.30749 -2.30749 -2.30749
## PB.130.4|056771|path1:13-5910(+)|transcript/2903   -2.30749 -2.30749 -2.30749
## PB.159.1|068d5c|path0:1-2199(+)|transcript/10689   -2.30749 -2.30749 -2.30749
## PB.253.3|092e23|path2:4-1403(+)|transcript/20322   -2.30749 -2.30749 -2.30749
## PB.281.1|0a5144|path2:1-1582(+)|transcript/18660   -2.30749 -2.30749 -2.30749
## PB.347.1|0ca129|path6:1-490(+)|transcript/24817    -2.30749 -2.30749 -2.30749
```

Reorder columns to cluster Season factor groups, and sexes

```
col.order_season <- c("G1_SepF", "G2_SepF", "G5_SepM", "G3_MarF",
    "G8_MarF", "G4_MarM")
low_variable_lcpm_season <- low_variable_lcpm_season[, col.order_season]
```

```
head(low_variable_lcpm_season)
```

```
##                                                     G1_SepF  G2_SepF  G5_SepM
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401 -2.30749 -2.30749 -2.30749
## PB.130.4|056771|path1:13-5910(+)|transcript/2903   -2.30749 -2.30749 -2.30749
## PB.159.1|068d5c|path0:1-2199(+)|transcript/10689   -2.30749 -2.30749 -2.30749
```

```
## PB.253.3|092e23|path2:4-1403(+)|transcript/20322   -2.30749 -2.30749 -2.30749
## PB.281.1|0a5144|path2:1-1582(+)|transcript/18660   -2.30749 -2.30749 -2.30749
## PB.347.1|0ca129|path6:1-490(+)|transcript/24817    -2.30749 -2.30749 -2.30749
##                                                     G3_MarF  G8_MarF  G4_MarM
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401 -2.30749 -2.30749 -2.30749
## PB.130.4|056771|path1:13-5910(+)|transcript/2903   -2.30749 -2.30749 -2.30749
## PB.159.1|068d5c|path0:1-2199(+)|transcript/10689   -2.30749 -2.30749 -2.30749
## PB.253.3|092e23|path2:4-1403(+)|transcript/20322   -2.30749 -2.30749 -2.30749
## PB.281.1|0a5144|path2:1-1582(+)|transcript/18660   -2.30749 -2.30749 -2.30749
## PB.347.1|0ca129|path6:1-490(+)|transcript/24817    -2.30749 -2.30749 -2.30749
```

Colour pallete

```
mypalette2 <- brewer.pal(11, "RdYlBu")
morecols <- colorRampPalette(mypalette2)
```

Colour vector for season_group factor variable: note - factor groups changed to match column reordering in low_variable_lcpm September=1 March=2

```
Reordered_season_group <- factor(c(1, 1, 1, 2, 2, 2))
col.cell_season <- c("chartreuse3", "orange")[Reordered_season_group]
```

Create the HeatMap

```
png(file = "Low_var_genes.heatmap-season.png", width = 7.5 *
    300, height = 5 * 300, res = 300, pointsize = 8)
# 5 x 300 pixels height = 5*300, res = 300,# 300 pixels per
# inch
heatmap.2(low_variable_lcpm_season, col = rev(morecols(50)),
    Colv = FALSE, trace = "none", main = "Bottom 250 most variable genes across samples",
    ColSideColors = col.cell_season, scale = "row", margins = c(8,
        4), srtCol = 45, labRow = "", dendrogram = c("none"))
```

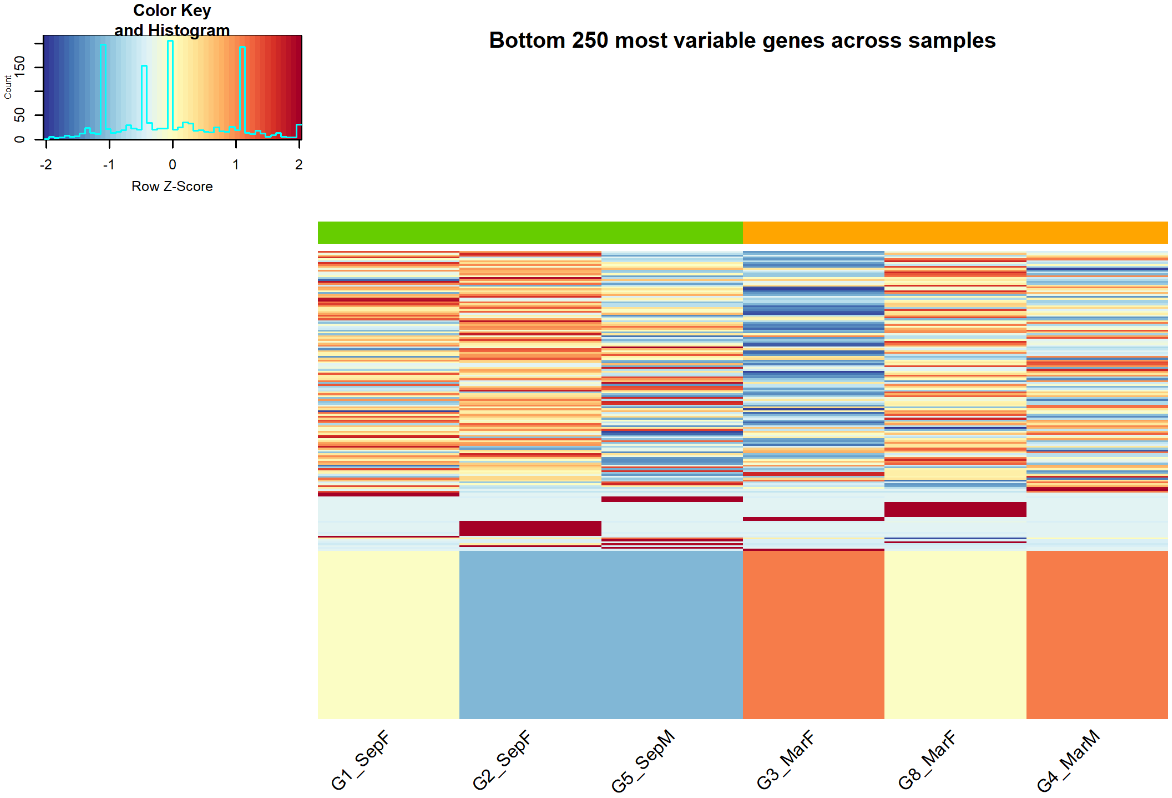


Figure 17: Bottom 250 most variable transcripts among six *T. adelaidensis* individuals, based on total counts per million values calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

5.3.3 Calculations

```r
# Data with actual counts, not TPM = 'dge'
head(dge_Season)
```

```
## An object of class "DGEList"
## $counts
##                                                        G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564     1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534        0.00000    5.27176
## PB.11.1|004815|path2:1-3039(+)|transcript/3426        42.09258   53.72594
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401     0.00000    0.00000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993       108.03236   68.57706
##                                                        G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564     990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       2.475519    1.944563
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       19.475009   16.355109
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.000000    0.000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993       18.788237   30.061179
##                                                        G5_SepM     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      131.000000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564     652.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       5.315287    4.370939
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       55.741267   14.183114
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.000000    0.000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993       33.818826   40.394709
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10088007            1
## G2_SepF     1 10028560            1
## G3_MarF     2 10002306            1
## G4_MarM     2 10118645            1
## G5_SepM     1  8854300            1
## G8_MarF     2 10310696            1
##
## $genes
##                                                    Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564      2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   1965        1642.711
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993       2714        2391.711
##                                                    Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304           1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564          1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534          2.924537
## PB.11.1|004815|path2:1-3039(+)|transcript/3426           1.425430
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401       1.000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993           1.064496
```

```
head(myTPM_Season)
```

```
##                                                        G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       0.00000    5.27176
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       42.09258   53.72594
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.00000    0.00000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993      108.03236   68.57706
##                                                        G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      2.475519    1.944563
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      19.475009   16.355109
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993      18.788237   30.061179
##                                                        G5_SepM     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     131.000000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    652.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      5.315287    4.370939
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      55.741267   14.183114
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993      33.818826   40.394709
```

Filter out genes with less than 5 reads in 2 samples for each transcript

```
filter_Season <- apply(dge_Season, 1, function(x) length(x[x >
    5]) >= 2)
filtered_Season <- dge_Season[filter_Season, ]
head(filtered_Season)
```

```
## An object of class "DGEList"
## $counts
##                                                     G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304   278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534    0.00000    5.27176
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    42.09258   53.72594
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   108.03236   68.57706
## PB.14.1|006283|path0:1-1955(+)|transcript/13821  988.84009  671.75142
##                                                     G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304   99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534   2.475519    1.944563
## PB.11.1|004815|path2:1-3039(+)|transcript/3426   19.475009   16.355109
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   18.788237   30.061179
## PB.14.1|006283|path0:1-1955(+)|transcript/13821 467.385028  700.161101
##                                                     G5_SepM     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304  131.000000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 652.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534   5.315287    4.370939
## PB.11.1|004815|path2:1-3039(+)|transcript/3426   55.741267   14.183114
```

```
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   33.818826   40.394709
## PB.14.1|006283|path0:1-1955(+)|transcript/13821 441.724674  734.985868
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10088007            1
## G2_SepF     1 10028560            1
## G3_MarF     2 10002306            1
## G4_MarM     2 10118645            1
## G5_SepM     1  8854300            1
## G8_MarF     2 10310696            1
##
## $genes
##                                                  Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304    1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564   2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534   1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    2999        2676.711
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993    2714        2391.711
## PB.14.1|006283|path0:1-1955(+)|transcript/13821   1920        1597.711
##                                                  Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304         1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564        1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534        2.924537
## PB.11.1|004815|path2:1-3039(+)|transcript/3426         1.425430
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993         1.064496
## PB.14.1|006283|path0:1-1955(+)|transcript/13821        1.091177
```

Run calculations

```
dge_Season <- calcNormFactors(dge_Season)
dge_Season <- estimateCommonDisp(dge_Season)
dge_Season <- estimateTagwiseDisp(dge_Season)
```

```
dge_Season$common.dispersion
```

```
## [1] 0.2581349
```

```
head(dge_Season$tagwise.dispersion)
```

```
## [1] 0.1528773 0.2153296 0.3857439 0.2599781 0.4338756 0.2145087
```

5.3.4 Differentially Expressed Genes:

View the top ten ‘differentially expressed genes’

```
results_Season <- exactTest(dge_Season)
topTags(results_Season)
```

```
## Comparison of groups:  2-1
##                                                        Length EffectiveLength
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418      2000       1677.7106
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242      2278       1955.7106
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825    1608       1285.7106
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882         2640       2317.7106
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548    1750       1427.7106
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462      3086       2763.7106
## PB.8300.1|transcript/22013:1-1186(+)|transcript/22013    1153        830.7106
## PB.499.1|124cf9|path4:1-2080(+)|transcript/11611         2071       1748.7106
## PB.5332.1|c9b505|path1:1-1445(+)|transcript/19992        1414       1091.7106
## PB.2003.2|4ab8e8|path0:1458-4018(+)|transcript/6743      2535       2212.7106
##                                                        Overdispersion     logFC
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418          1.634228 -2.967037
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242          3.962404  5.820065
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825        1.010844 -5.011832
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882             8.951643 -3.700669
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548        1.032928 -4.423886
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462          1.062696 -2.866295
## PB.8300.1|transcript/22013:1-1186(+)|transcript/22013        1.000000 -4.211046
## PB.499.1|124cf9|path4:1-2080(+)|transcript/11611             1.414303 -2.555325
## PB.5332.1|c9b505|path1:1-1445(+)|transcript/19992            3.495611 -3.467388
## PB.2003.2|4ab8e8|path0:1458-4018(+)|transcript/6743          1.550899 -2.478772
##                                                           logCPM       PValue
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418      4.6334826 4.421016e-09
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242      0.8151181 5.550732e-08
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825    2.8031450 7.503333e-08
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882         3.9451660 9.978139e-08
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548    2.7361115 3.118402e-07
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462      4.1681199 3.544003e-07
## PB.8300.1|transcript/22013:1-1186(+)|transcript/22013    2.2007363 7.750384e-07
## PB.499.1|124cf9|path4:1-2080(+)|transcript/11611         5.0358251 8.944590e-07
## PB.5332.1|c9b505|path1:1-1445(+)|transcript/19992        2.5037815 1.657704e-06
## PB.2003.2|4ab8e8|path0:1458-4018(+)|transcript/6743      4.4605648 2.316459e-06
##                                                                 FDR
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418      4.338343e-05
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242      2.447887e-04
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825    2.447887e-04
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882         2.447887e-04
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548    5.796216e-04
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462      5.796216e-04
## PB.8300.1|transcript/22013:1-1186(+)|transcript/22013    1.086493e-03
## PB.499.1|124cf9|path4:1-2080(+)|transcript/11611         1.097166e-03
## PB.5332.1|c9b505|path1:1-1445(+)|transcript/19992        1.807449e-03
## PB.2003.2|4ab8e8|path0:1458-4018(+)|transcript/6743      2.187699e-03
```

Output the top 25 table of values based on calculated PValue

```r
tab_Season <- topTags(results_Season, n = 25, sort.by = "PValue")
head(tab_Season)
```

```
## Comparison of groups:  2-1
##                                                        Length EffectiveLength
```

```
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418     2000        1677.711
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242     2278        1955.711
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825   1608        1285.711
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882        2640        2317.711
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548   1750        1427.711
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462     3086        2763.711
##                                                       Overdispersion     logFC
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418         1.634228 -2.967037
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242         3.962404  5.820065
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825       1.010844 -5.011832
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882            8.951643 -3.700669
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548       1.032928 -4.423886
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462         1.062696 -2.866295
##                                                          logCPM       PValue
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418   4.6334826 4.421016e-09
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242   0.8151181 5.550732e-08
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825 2.8031450 7.503333e-08
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882      3.9451660 9.978139e-08
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548 2.7361115 3.118402e-07
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462   4.1681199 3.544003e-07
##                                                                FDR
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418   4.338343e-05
## PB.3885.3|918572|path1:1406-3716(+)|transcript/9242   2.447887e-04
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825 2.447887e-04
## PB.3681.1|8a3a64|path3:1-2713(+)|transcript/5882      2.447887e-04
## PB.7443.1|transcript/15548:1-1797(+)|transcript/15548 5.796216e-04
## PB.8897.1|transcript/3462:1-3118(+)|transcript/3462   5.796216e-04
```

```
write.table(tab_Season, file = "Kgenelist-Top25_Season.txt")
```

Join this table of the top 25 differentially expressed genes with the annotation data

```
Kgenelist_Top25_Season <- read.csv("./Kgenelist-Top25_Season.txt",
    sep = " ", row.names = NULL, header = TRUE, stringsAsFactors = FALSE)
Kgenelist_Top25_Season <- Kgenelist_Top25_Season %>%
    rename(TranscriptID_3 = row.names)  #rename column to join with

Kgenelist_Top25_Season_Gene <- left_join(Kgenelist_Top25_Season,
    BLASTx.gene.join_Database.Full.PBT.Summary, by = "TranscriptID_3")  #add annotation
↪  info
# head (Kgenelist_Top25_Season_Gene)
colSums(!is.na(Kgenelist_Top25_Season_Gene))  #count column entries that are not NA
```

```
##     TranscriptID_3             Length    EffectiveLength     Overdispersion
##                 25                 25                 25                 25
##              logFC             logCPM             PValue                FDR
##                 25                 25                 25                 25
##       TranscriptID          UniprotID               Gene             pident
##                 16                 16                 16                 16
##             length           mismatch            gapopen             qstart
##                 16                 16                 16                 16
##               qend             sstart               send             evalue
```

```
##                 16                  16                  16                  16
##           bitscore              tax_id  Org_name.new_name            Org_name
##                 16                   3                   3                   3
##             GeneID              Symbol             Aliases         description
##                  3                   3                   3                   3
## other_designations
##                  3
```

```
write.csv(Kgenelist_Top25_Season_Gene, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/Kgenelist_Top25_Season_Gene.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(Note: the column "TranscriptID_3" indicates the full list of 25 output top differentially expressed genes. "TranscriptID" marks the start of the joined BLASTx data: 16 of these 25 transcripts returned a BLASTx result and gene ID under "Gene". "tax_id" marks the start of the joined genes of interest dataframe: only three of these 16 BLASTx results were also identified in the NCBI gene database search as of particular interest (under "Symbol" genes slc25a15, mcm3 and tk1).

Output the entire table of values

```
tab_Season_all <- topTags(results_Season, n = Inf, sort.by = "PValue")
write.table(tab_Season_all, file = "Kgenelist_Season.txt")
```

Output summary expression data

```
dim(results_Season)
```

```
## [1] 9813    3
```

```
summary(de_Season <- decideTests(results_Season))
```

```
##          2-1
## Down     189
## NotSig 9599
## Up        25
```

```
summary_Season <- (de_Season <- decideTests(results_Season))
head(summary_Season)
```

```
## TestResults matrix
##                                                   2-1
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      0
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564     0
## PB.10.1|004815|path1:5-1713(+)|transcript/17534     0
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      0
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401  0
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993      0
```

```r
write.table(summary_Season, file = "summary_Season.txt")

detags_Season <- rownames(dge_Season)[as.logical(de_Season)]
head(detags_Season)
```

```
## [1] "PB.193.6|07b09f|path0:5-2520(+)|transcript/22831"
## [2] "PB.193.7|07b09f|path0:5-2523(+)|transcript/22910"
## [3] "PB.221.2|0889ce|path1:2-3692(+)|transcript/1643"
## [4] "PB.222.2|0889ce|path13:1-894(+)|transcript/23126"
## [5] "PB.225.1|089810|path3:1-2255(+)|transcript/24270"
## [6] "PB.225.2|089810|path3:44-2167(+)|transcript/11385"
```

```r
head(results_Season)
```

```
## An object of class "DGEExact"
## $table
##                                                            logFC     logCPM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     -2.929926e-02  4.1846367
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564     9.644061e-01  7.0198583
## PB.10.1|004815|path1:5-1713(+)|transcript/17534    -9.401921e-02 -0.9102426
## PB.11.1|004815|path2:1-3039(+)|transcript/3426     -1.324846e+00  1.8464118
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401  5.125482e-15 -2.3498774
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993     -8.638145e-01  2.2548713
##                                                        PValue
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     0.95430801
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    0.08379776
## PB.10.1|004815|path1:5-1713(+)|transcript/17534    1.00000000
## PB.11.1|004815|path2:1-3039(+)|transcript/3426     0.04797759
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401 1.00000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993     0.14302160
##
## $comparison
## [1] "1" "2"
##
## $genes
##                                                    Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564      2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   1965        1642.711
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993       2714        2391.711
##                                                    Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304           1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564          1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534          2.924537
## PB.11.1|004815|path2:1-3039(+)|transcript/3426           1.425430
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401       1.000000
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993           1.064496
```

Filter and sort this summary to only include upregulated and downregulated transcripts

```
summary_Season.sort <- read.csv("./summary_Season.txt", sep = " ",
    row.names = NULL, header = TRUE, stringsAsFactors = FALSE)
summary_Season.sort <- summary_Season.sort %>%
    rename(TranscriptID_3 = row.names, Regulation = X2.1)  #the name 'TranscriptID_3' is
     ↪  used for consistency and joining as seen in the next step
```

Get the Expression statistics for all transcripts together with the BLASTx statistics in one file

```
# Header for column 1 manually changed to 'TranscriptID_3'
# in 'Kgenelist_Season.txt'
All_Season.sort <- read.csv("./Kgenelist_Season.txt", sep = " ",
    header = TRUE, stringsAsFactors = TRUE)
All_Season.sort$TranscriptID_3 <- rownames(All_Season.sort)
# head(All_Season.sort)
# head(BLASTx.gene.join_Database.Full.PBT.Summary)
BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp <-
↪   left_join(BLASTx.gene.join_Database.Full.PBT.Summary,
    All_Season.sort, by = "TranscriptID_3")
head(BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp)
```

```
##                                                     TranscriptID
## 1 PB.7010.1|transcript/12012:1-2076(+)|transcript/12012|m.12781
## 2      PB.2837.1|6a2a91|path0:1-1784(+)|transcript/15955|m.5151
## 3      PB.2838.1|6a2a91|path1:1-2161(+)|transcript/11047|m.5152
## 4      PB.2778.1|67914a|path1:1-1875(+)|transcript/14733|m.5057
## 5     PB.2778.2|67914a|path1:22-1698(+)|transcript/16857|m.5058
## 6      PB.6589.1|fb4549|path0:1-2510(+)|transcript/7231|m.12232
##                                           TranscriptID_3 UniprotID    Gene
## 1 PB.7010.1|transcript/12012:1-2076(+)|transcript/12012    Q9NRG9    aaas
## 2     PB.2837.1|6a2a91|path0:1-1784(+)|transcript/15955    Q5VUY2 aadacl4
## 3     PB.2838.1|6a2a91|path1:1-2161(+)|transcript/11047    Q5VUY2 aadacl4
## 4     PB.2778.1|67914a|path1:1-1875(+)|transcript/14733    Q5E9N4   aadat
## 5    PB.2778.2|67914a|path1:22-1698(+)|transcript/16857    Q5E9N4   aadat
## 6      PB.6589.1|fb4549|path0:1-2510(+)|transcript/7231    Q8N5Z0   aadat
##   pident length mismatch gapopen qstart qend sstart send    evalue bitscore
## 1 66.048    539      162       2    127 1737      1  520  0.00e+00      684
## 2 44.477    344      188       3      1 1026     63  405  8.03e-98      298
## 3 42.820    383      217       2     79 1224     24  405 6.29e-108      327
## 4 68.235    425      135       0      1 1275      1  425  0.00e+00      647
## 5 68.235    425      135       0      1 1275      1  425  0.00e+00      647
## 6 58.824    425      174       1      1 1272      1  425  0.00e+00      543
##   tax_id Org_name.new_name Org_name GeneID Symbol Aliases description
## 1     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 2     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 3     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 4     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 5     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
## 6     NA              <NA>     <NA>     NA   <NA>    <NA>        <NA>
##   other_designations Length EffectiveLength Overdispersion       logFC   logCPM
## 1               <NA>   2046        1723.711       1.000000 -0.09969098 3.715385
## 2               <NA>     NA              NA             NA          NA       NA
## 3               <NA>   2028        1705.711       1.009976  1.12455852 6.934186
```

```
## 4                <NA>   1841        1518.711       1.000000  0.91270230 6.429255
## 5                <NA>     NA              NA             NA          NA       NA
## 6                <NA>   2476        2153.711       2.533499  0.30344590 6.038945
##        PValue       FDR
## 1 0.84758404 1.0000000
## 2         NA        NA
## 3 0.06132159 0.5765864
## 4 0.09708727 0.7003254
## 5         NA        NA
## 6 0.71520870 1.0000000
```

Export this file as a .csv

```
BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp$other_designations <- gsub(",",
    "-",
    ↪  as.character(BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp$other_designations))
BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp$description <- gsub(",",
    "-", as.character(BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp$description))
# removing commas to retain the integrity of column
# delimiters
write.csv(BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp,
    "~/@Uni/@Flinders University
    ↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/BLASTx.gene.join_Database.Full.PBT.Summa
    quote = FALSE, row.names = FALSE, col.names = TRUE)
```

filter out the transcripts which are significantly upregulated when compared by season

```
summary_Season.sort_UP <- filter(summary_Season.sort, Regulation ==
    "1")  #upregulated transcripts denoted as +1
head(summary_Season.sort_UP)
```

```
##                                     TranscriptID_3 Regulation
## 1 PB.876.2|206df6|path10:20-1995(+)|transcript/12799          1
## 2 PB.876.5|206df6|path10:31-1974(+)|transcript/14832          1
## 3  PB.879.1|206df6|path7:41-2171(+)|transcript/12507          1
## 4 PB.1305.2|31fdc7|path1:93-2420(+)|transcript/10029          1
## 5 PB.1391.1|34b6f8|path1:88-1988(+)|transcript/14257          1
## 6   PB.1436.1|365ff0|path0:1-3058(+)|transcript/6006          1
```

```
nrow(summary_Season.sort_UP)
```

```
## [1] 25
```

Extract the full information about the putative identiry of these transcripts from the Summary file

```
summary_Season.sort_UP_info <-
↪  semi_join(BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp,
    summary_Season.sort_UP, by = "TranscriptID_3")
# head (summary_Season.sort_UP_info)
nrow(summary_Season.sort_UP_info)
```

```
## [1] 19
```

```
write.csv(summary_Season.sort_UP_info, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Season.sort_UP_info.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: although there are 25 transcripts significantly upregulated, only 19 of these returned a BLASTx result for their corresponding putative coding region)

Filter for Transcripts in this list which ALSO have information from the 'genes of interest' list

```
summary_Season.sort_UP_info.GOI <- summary_Season.sort_UP_info %>%
    filter(!is.na(GeneID))

nrow(summary_Season.sort_UP_info.GOI)
```

```
## [1] 1
```

```
write.csv(summary_Season.sort_UP_info.GOI, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Season.sort_UP_info.GOI.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: 1 of the 19 upregulated transcripts that returned a BLASTx result, are represented in the 'genes of interest' list)

Do the same for the Seasonally deownregulated transcripts

```
summary_Season.sort_DOWN <- filter(summary_Season.sort, Regulation ==
    "-1")  #downregulated transcripts denoted as -1
head(summary_Season.sort_DOWN)
```

```
##                                       TranscriptID_3 Regulation
## 1  PB.193.6|07b09f|path0:5-2520(+)|transcript/22831         -1
## 2  PB.193.7|07b09f|path0:5-2523(+)|transcript/22910         -1
## 3   PB.221.2|0889ce|path1:2-3692(+)|transcript/1643         -1
## 4  PB.222.2|0889ce|path13:1-894(+)|transcript/23126         -1
## 5  PB.225.1|089810|path3:1-2255(+)|transcript/24270         -1
## 6 PB.225.2|089810|path3:44-2167(+)|transcript/11385         -1
```

```
nrow(summary_Season.sort_DOWN)
```

```
## [1] 189
```

Extract the full information about the putative identity of these transcripts from the Summary file

```
summary_Season.sort_DOWN_info <-
↪  semi_join(BLASTx.gene.join_Database.Full.PBT.Summary_SeasonExp,
    summary_Season.sort_DOWN, by = "TranscriptID_3")
# head (summary_Season.sort_DOWN_info)
nrow(summary_Season.sort_DOWN_info)
```

```
## [1] 99
```

```
write.csv(summary_Season.sort_DOWN_info, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Season.sort_DOWN_info.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: although there are 189 transcripts significantly downregulated, only 99 of these returned a BLASTx result for their corresponding putative coding region)

Filter for Transcripts in this list which ALSO have information from the 'genes of interest' list

```
summary_Season.sort_DOWN_info.GOI <- summary_Season.sort_DOWN_info %>%
    filter(!is.na(GeneID))

nrow(summary_Season.sort_DOWN_info.GOI)
```

```
## [1] 7
```

```
write.csv(summary_Season.sort_DOWN_info.GOI, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Season.sort_DOWN_info.GOI.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: 7 of the 99 downregulated transcripts that returned a BLASTx result, are represented in the 'genes of interest' list)

Visualise the logged Fold Change against Average logged counts per million, in expression of transcripts between seasons using plotsmear: red indicates PValue < 0.05

```
png("plotsmear_Season.png", width = 600)
plotSmear(results_Season, de.tags = detags_Season)
```

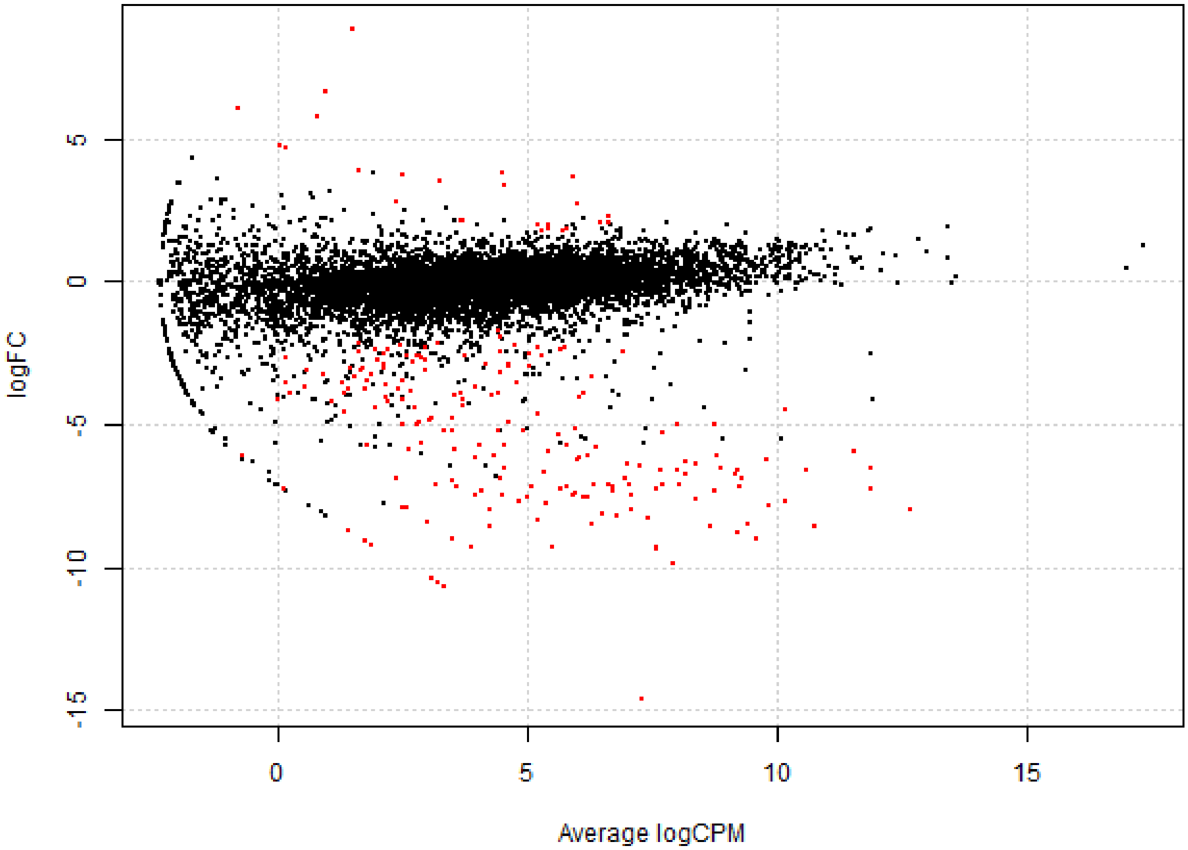


Figure 18: Log Fold Change vs average Log counts per million for expression level comparing March to September season variables among six *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change over season, with p-values <0.05. Counts calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts.

Visualise differential expression of transcripts between season 1 (September) and 2 (March) with negative Logged PValue against Logged Fold Change in expression: Red indicates a transcript with PValue < 0.05, and Log Fold change > 2

```
volcanoData_Season <- cbind(results_Season$table$logFC,
↪  -log10(results_Season$table$PValue))
colnames(volcanoData_Season) <- c("logFC", "negLogPval")
DEGs_Season <- results_Season$table$PValue < 0.05 & abs(results_Season$table$logFC) >
    2

point.col_Season <- ifelse(DEGs_Season, "red", "black")
png("volcanoData_Season.png", width = 600)
plot(volcanoData_Season, pch = 16, col = point.col_Season, cex = 0.5)
# pch is shape 1-26 are vectors, cex is size
```

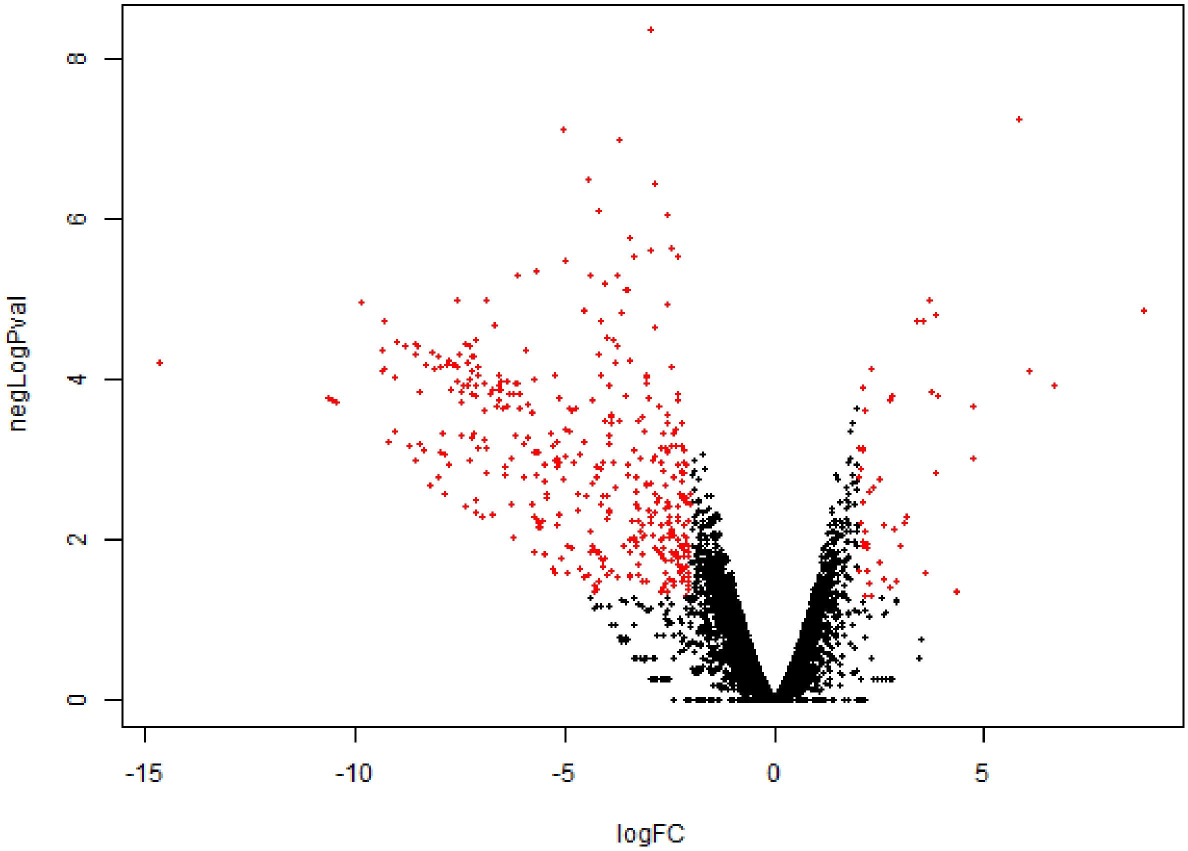


Figure 19: Negative Log of the p-value against Log Fold Change in expression level comparing March to September season variables among six *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change over season, with p-values <0.05 and *also* a log fold change >2. Counts calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts.

5.3.5 Group Factor: Sex

Create the EdgeR DGE list for use in subsequent analyses

```
dge_Sex <- DGEList(counts = Kcaught$counts/Kcaught$annotation$Overdispersion,
    genes = Kcaught$annotation, group = sex_group)
# View(dge_Sex) names(dge_Sex)
dge_Sex
```

```
## An object of class "DGEList"
## $counts
##                                                       G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       0.00000    5.27176
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       42.09258   53.72594
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.00000    0.00000
##                                                       G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      2.475519    1.944563
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      19.475009   16.355109
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
##                                                       G5_SepM     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     131.000000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    652.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      5.315287    4.370939
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      55.741267   14.183114
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
## 9808 more rows ...
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10088007            1
## G2_SepF     1 10028560            1
## G3_MarF     1 10002306            1
## G4_MarM     2 10118645            1
## G5_SepM     2  8854300            1
## G8_MarF     1 10310696            1
##
## $genes
##                                                    Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564      2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   1965        1642.711
##                                                    Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304           1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564          1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534          2.924537
## PB.11.1|004815|path2:1-3039(+)|transcript/3426           1.425430
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401       1.000000
## 9808 more rows ...
```

Obtain Counts Per Million to standardise count data comparison

```
myTPM_Sex <- dge_Sex$counts
# head(myTPM_Sex)
```

5.3.6 Initial Data Exploration

Which values in myCPM are greater than 0.5? This produces a logical matrix with TRUEs and FALSEs. TRUE values are samples with > 0.5 counts in that sample per million

```
thresh_Sex <- myTPM_Sex > 0.5
# This produces a logical matrix with TRUEs and FALSEs
head(thresh_Sex)
```

```
##                                                     G1_SepF G2_SepF G3_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304         TRUE    TRUE    TRUE
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564        TRUE    TRUE    TRUE
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       FALSE    TRUE    TRUE
## PB.11.1|004815|path2:1-3039(+)|transcript/3426         TRUE    TRUE    TRUE
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    FALSE   FALSE   FALSE
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993         TRUE    TRUE    TRUE
##                                                     G4_MarM G5_SepM G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304         TRUE    TRUE    TRUE
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564        TRUE    TRUE    TRUE
## PB.10.1|004815|path1:5-1713(+)|transcript/17534        TRUE    TRUE    TRUE
## PB.11.1|004815|path2:1-3039(+)|transcript/3426         TRUE    TRUE    TRUE
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    FALSE   FALSE   FALSE
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993         TRUE    TRUE    TRUE
```

Raw Counts per million and library sizes will be consistent with the Season comparison above. There is no need to plot them again however scripts are included to create separate R objects for downstream application.

CPM distribution with logged CPM

```
# export the plot as png
png("plotDensities_Sex.png", width = 600)
unfilteredExpr_Sex <- cpm(dge_Sex, log = T)
plotDensities(unfilteredExpr_Sex, col = myPalette, legend = TRUE)
```

Library sizes per sample

```
# export the plot as png
png("LibSize_Sex.png", width = 600)
# The names argument tells the barplot to use the sample
# names on the x-axis The last argument rotates the axis
# names
barplot(dge_Sex$samples$lib.size, names = colnames(dge_Sex),
    las = 2, col = myPalette, legend = TRUE)
```

Total estimated transcript counts per sample

```
# export the plot as png
png("boxplot_Sex.png", width = 600)
boxplot(dge_Sex$counts, col = myPalette, las = 2, legend = TRUE)
# title('Boxplot of transcript total estimated counts')
```

Logged total estimated transcript counts per sample

```
# Get log2 counts per million
logcounts_Sex <- cpm(dge_Sex, log = TRUE)
# Check distributions of samples using boxplots export the
# plot as png
png("boxplot-logcounts_Sex.png", width = 600)
boxplot(logcounts_Sex, xlab = "", ylab = "Log2 counts per million",
    las = 2, col = myPalette, legend = TRUE)
# Let's add a blue horizontal line that corresponds to the
# median logCPM
abline(h = median(logcounts_Sex), col = myPalette, legend = TRUE)
# title('Boxplots of log Counts Per Million')
```

5.3.7 Calculations

```
# Data with actual counts, not TPM = 'dge' head(dge_Sex)
# head(myTPM_Sex)
```

Filter out genes with less than 5 reads in 2 samples for each transcript

```
filter_Sex <- apply(dge_Sex, 1, function(x) length(x[x > 5]) >=
    2)
filtered_Sex <- dge_Sex[filter_Sex, ]
head(filtered_Sex)
```

```
## An object of class "DGEList"
## $counts
##                                                  G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304   278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534    0.00000    5.27176
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    42.09258   53.72594
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   108.03236   68.57706
## PB.14.1|006283|path0:1-1955(+)|transcript/13821  988.84009  671.75142
##                                                  G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304   99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534   2.475519    1.944563
## PB.11.1|004815|path2:1-3039(+)|transcript/3426   19.475009   16.355109
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   18.788237   30.061179
## PB.14.1|006283|path0:1-1955(+)|transcript/13821 467.385028  700.161101
##                                                  G5_SepM     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304  131.000000  225.000000
```

```
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 652.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534   5.315287    4.370939
## PB.11.1|004815|path2:1-3039(+)|transcript/3426   55.741267   14.183114
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   33.818826   40.394709
## PB.14.1|006283|path0:1-1955(+)|transcript/13821 441.724674  734.985868
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10088007            1
## G2_SepF     1 10028560            1
## G3_MarF     1 10002306            1
## G4_MarM     2 10118645            1
## G5_SepM     2  8854300            1
## G8_MarF     1 10310696            1
##
## $genes
##                                                  Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304    1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564   2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534   1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    2999        2676.711
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993    2714        2391.711
## PB.14.1|006283|path0:1-1955(+)|transcript/13821   1920        1597.711
##                                                  Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304         1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564        1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534        2.924537
## PB.11.1|004815|path2:1-3039(+)|transcript/3426         1.425430
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993         1.064496
## PB.14.1|006283|path0:1-1955(+)|transcript/13821        1.091177
```

Run calculations

```r
dge_Sex <- calcNormFactors(dge_Sex)
dge_Sex <- estimateCommonDisp(dge_Sex)
dge_Sex <- estimateTagwiseDisp(dge_Sex)
```

```r
dge_Sex$common.dispersion
```

```
## [1] 0.2178922
```

```r
head(dge_Sex$tagwise.dispersion)
```

```
## [1] 0.1319103 0.2089947 0.3706877 0.3181733 0.4299255 0.2331927
```

5.3.8 Differentially Expressed Genes:

View the top ten ‘differentially expressed genes’

```r
results_Sex <- exactTest(dge_Sex)
topTags(results_Sex)
```

```
## Comparison of groups:  2-1
##                                                    Length EffectiveLength
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021   1103        780.7106
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195   1136        813.7106
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575     3066       2743.7106
## PB.3135.1|74403d|path0:1-863(+)|transcript/23390     825        502.7858
## PB.5826.1|dd0edc|path1:1-2280(+)|transcript/9146    2280       1957.7106
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860     905        582.7631
## PB.2180.1|51f17e|path0:1-673(+)|transcript/24169     469        155.3070
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901     896        573.7631
## PB.725.1|1b70a4|path2:1-1745(+)|transcript/16546    1705       1382.7106
## PB.193.9|07b09f|path0:5-2700(+)|transcript/23445     647        325.3615
##                                                    Overdispersion    logFC
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021       2.871719 7.965249
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195       1.766850 8.089864
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575         1.024380 8.364555
## PB.3135.1|74403d|path0:1-863(+)|transcript/23390        1.037547 7.812164
## PB.5826.1|dd0edc|path1:1-2280(+)|transcript/9146        1.000000 7.091370
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860        1.173526 7.639690
## PB.2180.1|51f17e|path0:1-673(+)|transcript/24169        1.000000 7.617367
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901        1.309409 8.194551
## PB.725.1|1b70a4|path2:1-1745(+)|transcript/16546        1.000000 4.156773
## PB.193.9|07b09f|path0:5-2700(+)|transcript/23445        1.618587 6.510361
##                                                       logCPM       PValue
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021  5.666695 2.550012e-43
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195  8.292470 5.483625e-36
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575    4.081070 1.328872e-25
## PB.3135.1|74403d|path0:1-863(+)|transcript/23390   2.953564 1.640197e-24
## PB.5826.1|dd0edc|path1:1-2280(+)|transcript/9146   3.164706 1.013885e-23
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860   7.524509 2.471147e-20
## PB.2180.1|51f17e|path0:1-673(+)|transcript/24169   9.391923 3.726989e-20
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901   6.841911 6.409448e-20
## PB.725.1|1b70a4|path2:1-1745(+)|transcript/16546   5.897672 2.689880e-19
## PB.193.9|07b09f|path0:5-2700(+)|transcript/23445  11.876401 3.730622e-18
##                                                             FDR
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021 2.502327e-39
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195 2.690541e-32
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575   4.346740e-22
## PB.3135.1|74403d|path0:1-863(+)|transcript/23390  4.023813e-21
## PB.5826.1|dd0edc|path1:1-2280(+)|transcript/9146  1.989850e-20
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860  4.041561e-17
## PB.2180.1|51f17e|path0:1-673(+)|transcript/24169  5.224707e-17
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901  7.861989e-17
## PB.725.1|1b70a4|path2:1-1745(+)|transcript/16546  2.932865e-16
## PB.193.9|07b09f|path0:5-2700(+)|transcript/23445  3.660859e-15
```

Output the top 25 table of values based on calculated PValue

```
tab_Sex <- topTags(results_Sex, n = 25, sort.by = "PValue")
# head(tab_Sex)
write.table(tab_Sex, file = "Kgenelist-Top25_Sex.txt")
```

Output the entire table of values

```
tab_Sex_all <- topTags(results_Sex, n = Inf, sort.by = "PValue")
write.table(tab_Sex_all, file = "Kgenelist_Sex.txt")
```

Output summary expression data

```
dim(results_Sex)
```

```
## [1] 9813    3
```

```
summary(de_Sex <- decideTests(results_Sex))
```

```
##         2-1
## Down     27
## NotSig 9422
## Up      364
```

```
summary_Sex <- (de_Sex <- decideTests(results_Sex))

write.table(summary_Sex, file = "summary_Sex.txt")

detags_Sex <- rownames(dge_Sex)[as.logical(de_Sex)]
# head(detags_Sex) head(results_Sex)
```

Filter and sort this summary to only include upregulated and downregulated transcripts

```
summary_Sex.sort <- read.csv("./summary_Sex.txt", sep = " ",
    row.names = NULL, header = TRUE, stringsAsFactors = FALSE)
summary_Sex.sort <- summary_Sex.sort %>%
    rename(TranscriptID_3 = row.names, Regulation = X2.1)
```

filter out the transcripts which are significantly upregulated when compared by sex

```
summary_Sex.sort_UP <- filter(summary_Sex.sort, Regulation ==
    "1")  #upregulated transcripts denoted as +1
head(summary_Sex.sort_UP)
```

```
##                                     TranscriptID_3 Regulation
## 1   PB.48.1|01e359|path4:1-3511(+)|transcript/2176          1
## 2   PB.48.2|01e359|path4:4-3376(+)|transcript/2493          1
## 3 PB.147.1|0620ba|path0:1-1756(+)|transcript/23387          1
## 4 PB.147.3|0620ba|path0:1-1628(+)|transcript/24333          1
## 5 PB.193.1|07b09f|path0:1-2724(+)|transcript/21409          1
## 6 PB.193.2|07b09f|path0:3-2528(+)|transcript/11304          1
```

```
nrow(summary_Sex.sort_UP)
```

```
## [1] 364
```

Extract the full information about the putative identiry of these transcripts from the Summary file

```
summary_Sex.sort_UP_info <- semi_join(BLASTx.gene.join_Database.Full.PBT.Summary,
    summary_Sex.sort_UP, by = "TranscriptID_3")
# head (summary_Sex.sort_UP_info)
nrow(summary_Sex.sort_UP_info)
```

```
## [1] 210
```

```
write.csv(summary_Sex.sort_UP_info, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Sex.sort_UP_info.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: although there are 364 transcripts significantly upregulated, only 210 of these returned a BLASTx result for their corresponding putative coding region)

Filter for Transcripts in this list which ALSO have information from the 'genes of interest' list

```
summary_Sex.sort_UP_info.GOI <- summary_Sex.sort_UP_info %>%
    filter(!is.na(GeneID))

nrow(summary_Sex.sort_UP_info.GOI)
```

```
## [1] 15
```

```
write.csv(summary_Sex.sort_UP_info.GOI, "~/@Uni/@Flinders University
↪   PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Sex.sort_UP_info.GOI.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: 15 of the 210 upregaulted transcripts that returned a BLASTx result, are represented in the 'genes of interest' list)

Do the same for the deownregulated transcripts

```
summary_Sex.sort_DOWN <- filter(summary_Sex.sort, Regulation ==
    "-1")  #downregulated transcripts denoted as -1
head(summary_Sex.sort_DOWN)
```

```
##                                         TranscriptID_3 Regulation
## 1   PB.687.3|1a3601|path0:17-1696(+)|transcript/17363         -1
## 2  PB.871.3|206114|path0:182-1900(+)|transcript/16557         -1
## 3     PB.953.1|24d6a9|path0:1-2789(+)|transcript/5211         -1
## 4    PB.1210.1|2ea8f7|path0:1-2553(+)|transcript/7797         -1
## 5  PB.1762.2|4279e1|path0:18-2114(+)|transcript/11272         -1
## 6 PB.1863.3|46a928|path18:46-1843(+)|transcript/14672         -1
```

```r
nrow(summary_Sex.sort_DOWN)
```

```
## [1] 27
```

Extract the full information about the putative identiry of these transcripts from the Summary file

```r
summary_Sex.sort_DOWN_info <- semi_join(BLASTx.gene.join_Database.Full.PBT.Summary,
    summary_Sex.sort_DOWN, by = "TranscriptID_3")
# head (summary_Sex.sort_DOWN_info)
nrow(summary_Sex.sort_DOWN_info)
```

```
## [1] 24
```

```r
write.csv(summary_Sex.sort_DOWN_info, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Sex.sort_DOWN_info.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: although there are 27 transcripts significantly downregulated, 24 of these returned a BLASTx result for their corresponding putative coding region)

Filter for Transcripts in this list which ALSO have information from the 'genes of interest' list

```r
summary_Sex.sort_DOWN_info.GOI <- summary_Sex.sort_DOWN_info %>%
    filter(!is.na(GeneID))

nrow(summary_Sex.sort_DOWN_info.GOI)
```

```
## [1] 3
```

```r
write.csv(summary_Sex.sort_DOWN_info.GOI, "~/@Uni/@Flinders University
↪  PhD/RWorkingDir/Workflow-readthedown/ReferenceClusters/summary_Sex.sort_DOWN_info.GOI.csv",
    quote = FALSE, row.names = FALSE, col.names = TRUE)  #write the file to a .csv
```

(note: 3 of the 24 downregaulted transcripts that returned a BLASTx result, are represented in the 'genes of interest' list)

Visualise the logged Fold Change against Average logged counts per million, in expression of transcripts between sex 1 (Female) and 2 (Male) using plotsmear: red indicates PValue < 0.05

```r
png("plotsmear_Sex.png", width = 600)
plotSmear(results_Sex, de.tags = detags_Sex)
```

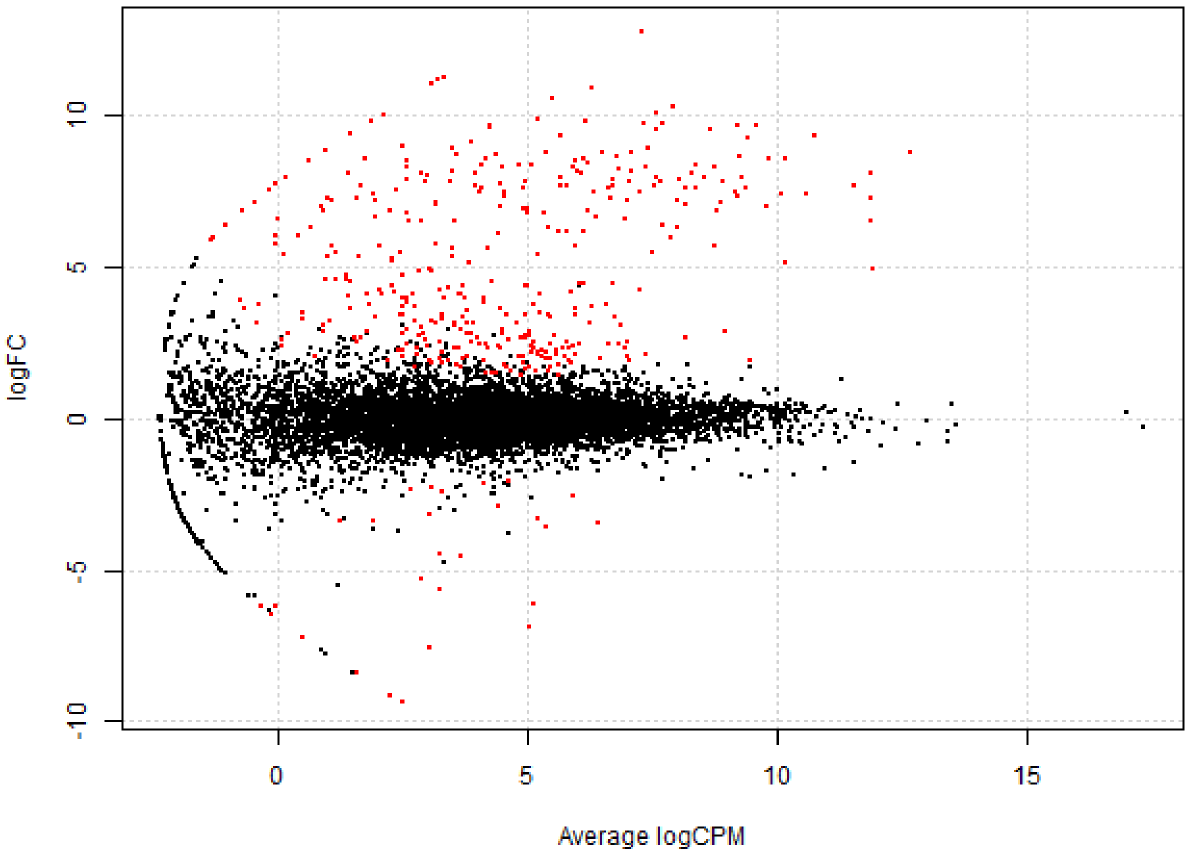


Figure 20: Log Fold Change vs average Log counts per million for expression level comparing Male to Female sex variables among six *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change between sexes, with p-values <0.05. Counts calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts.

Visualise differential expression of transcripts between sex 1 (Female) and 2 (Male) with negative Logged PValue against Logged Fold Change in expression: Red indicates a transcript with PValue < 0.05, and Log Fold change > 2

```r
volcanoData_Sex <- cbind(results_Sex$table$logFC, -log10(results_Sex$table$PValue))
colnames(volcanoData_Sex) <- c("logFC", "negLogPval")
DEGs_Sex <- results_Sex$table$PValue < 0.05 & abs(results_Sex$table$logFC) >
    2

point.col_Sex <- ifelse(DEGs_Sex, "red", "black")
png("volcanoData_Sex.png", width = 600)
plot(volcanoData_Sex, pch = 16, col = point.col_Sex, cex = 0.5)
# pch is shape 1-26 are vectors, cex is size
```

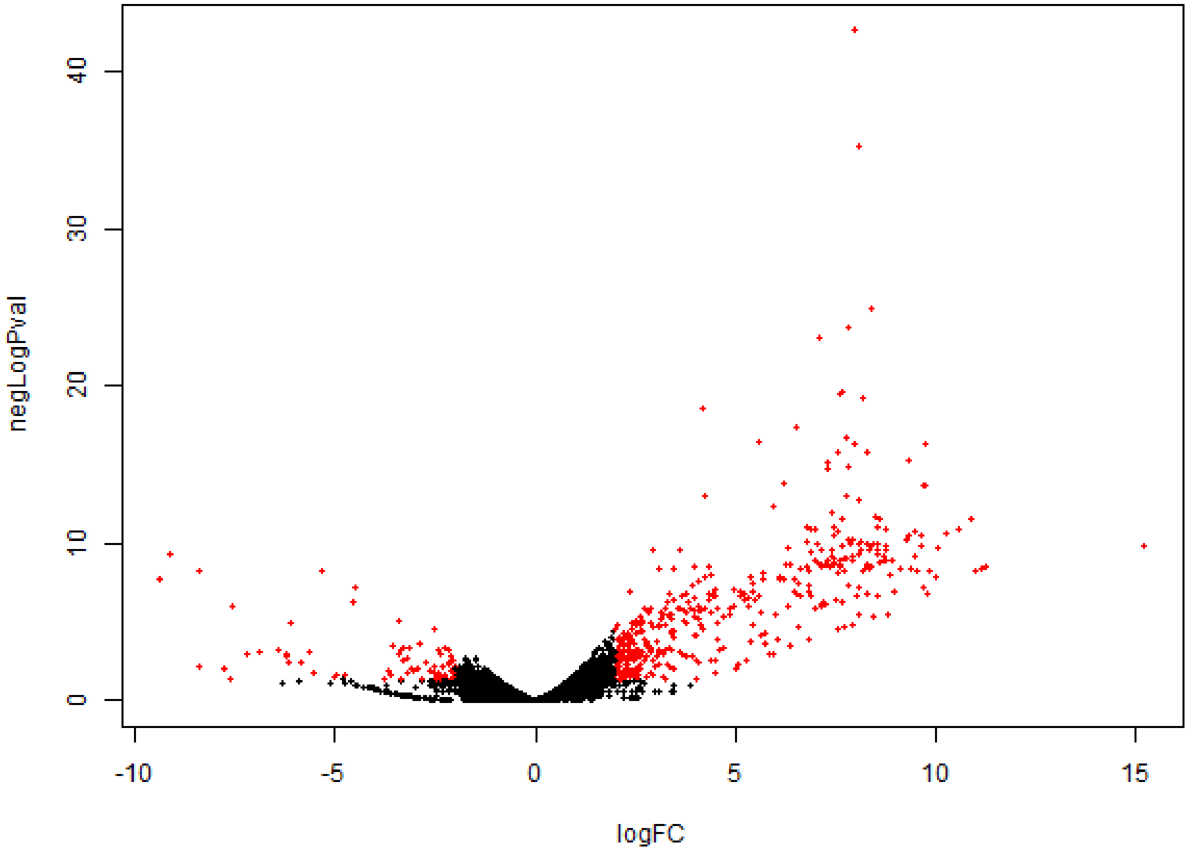


Figure 21: Negative Log of the p-value against Log Fold Change in expression level comparing Male to Female sex variables among six *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change between sexes, with p-values <0.05 and *also* a log fold change >2. Counts calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of six *Tiliqua adelaidensis* individuals, three collected in autumn and three collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts.

5.4 Sequencing Batch Effects

Note G1, G2, G3, G4, G5 and G8 were all sequenced together on an Illumina Hiseq and library prep was done with the same kit. G6 and G7 were later prepared in a separate batch using a different kit and sequenced on an Illumina Hiseq with fewer samples, and have a much higher read depth. These 2 represent a September Male, and March Female, respectively.

The removal of G6 and G7 from the dataset results in two groups of 3 per season, each with 2 females and 1 male, as analysed above. The below analyses have been provided for comparison of how group manipulation and inclusion of these samples affects results.

5.5 Eight *T. adelaidensis* Individuals Collected Between Two Seasonal Periods.

Exploration of all Eight samples by each group factor:

Import the data:

```
# create paths
setwd("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr/kallisto-clstr/")
DIR8 <- ("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr/kallisto-clstr/")
head(DIR8)
```

```
## [1] "~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr/kallisto-clstr/"
```

```
paths8 <- list.dirs(path = DIR8, full.names = FALSE, recursive = FALSE)
# the working dir only includes kallisto output for the
# kidneys at this time View(paths)
head(paths8)
```

```
## [1] "G1_SepF" "G2_SepF" "G3_MarF" "G4_MarM" "G5_SepM" "G6_SepM"
```

```
Kcaught8 <- catchKallisto(paths8, verbose = TRUE)
```

```
## Reading G1_SepF, 9813 transcripts, 100 bootstraps
## Reading G2_SepF, 9813 transcripts, 100 bootstraps
## Reading G3_MarF, 9813 transcripts, 100 bootstraps
## Reading G4_MarM, 9813 transcripts, 100 bootstraps
## Reading G5_SepM, 9813 transcripts, 100 bootstraps
## Reading G6_SepM, 9813 transcripts, 100 bootstraps
## Reading G7_MarF, 9813 transcripts, 100 bootstraps
## Reading G8_MarF, 9813 transcripts, 100 bootstraps
```

```
setwd("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr")
# View(Kcaught)
```

Set the group factors. Input Sample files are listed G1-G8 Group factors are applied to the DGElist object and are the point of comparison for expression analyses.

Season of Collection: 1 = September, 2 = March For the purposes of consistent labelling per group factors some April collections are referred to in the March group. Accurate collection information and dates are outlined in the methods chapter.

Individual Sex: 1 = Female, 2 = Male

```
# these factors correlate to season of collection. 1 =
# March, 2 = Sep.
season8_group <- factor(c(1, 1, 2, 2, 1, 1, 2, 2))

# these factors correlate to individuals sex. 1 = Female, 2
# = Male
sex8_group <- factor(c(1, 1, 1, 2, 2, 2, 1, 1))
```

5.5.1 Group Factor: Season

Create the EdgeR DGE list

```
dge_Season8 <- DGEList(counts = Kcaught8$counts/Kcaught8$annotation$Overdispersion,
    genes = Kcaught8$annotation, group = season8_group)
# View(dge_Season8)
names(dge_Season8)
```

```
## [1] "counts"  "samples" "genes"
```

```
dge_Season8
```

```
## An object of class "DGEList"
## $counts
##                                                        G1_SepF     G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      278.00000  218.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    1208.00000 1143.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       0.00000    5.206985
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       44.12913   56.325347
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.00000    0.000000
##                                                        G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      2.445102    1.920669
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      20.417263   17.146414
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
##                                                        G5_SepM    G6_SepM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     131.000000  793.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    652.000000 9810.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      5.249977   75.81151
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      58.438180 6815.29409
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000   62.63286
##                                                         G7_MarF     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       561.00000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    12443.00000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       57.12165    4.317232
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      3215.92340   14.869331
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    55.19076    0.000000
## 9808 more rows ...
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1  9760098            1
## G2_SepF     1  9760939            1
## G3_MarF     2  9745167            1
## G4_MarM     2  9855867            1
## G5_SepM     1  8598755            1
## G6_SepM     1 41290113            1
## G7_MarF     2 43632769            1
## G8_MarF     2 10009376            1
##
```

```
## $genes
##                                                     Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304        1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564       2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426        2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    1965        1642.711
##                                                     Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304            1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564           1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534           2.960918
## PB.11.1|004815|path2:1-3039(+)|transcript/3426            1.359646
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401        2.524871
## 9808 more rows ...
```

Obtain Counts Per Million

```
myTPM_Season8 <- dge_Season8$counts
# head(myTPM_Season8)
```

5.5.2 Initial Data Exploration

```
# Which values in myCPM are greater than 0.5?
thresh_Season8 <- myTPM_Season8 > 0.5
# This produces a logical matrix with TRUEs and FALSEs
# head(thresh_Season8)
```

Insert colour pallate for following visualisations consistency

```
myPalette8 <- c("#999999", "#F0E442", "#56B4E9", "#009E73", "#D55E00",
    "#0072B2", "#E69F00", "#CC79A7")
```

Plot CPM distribution with logged CPM

```
# export the plot as png
png("plotDensities_Season8.png", width = 600)
unfilteredExpr_Season8 <- cpm(dge_Season8, log = T)
plotDensities(unfilteredExpr_Season8, col = myPalette8, legend = TRUE)
```

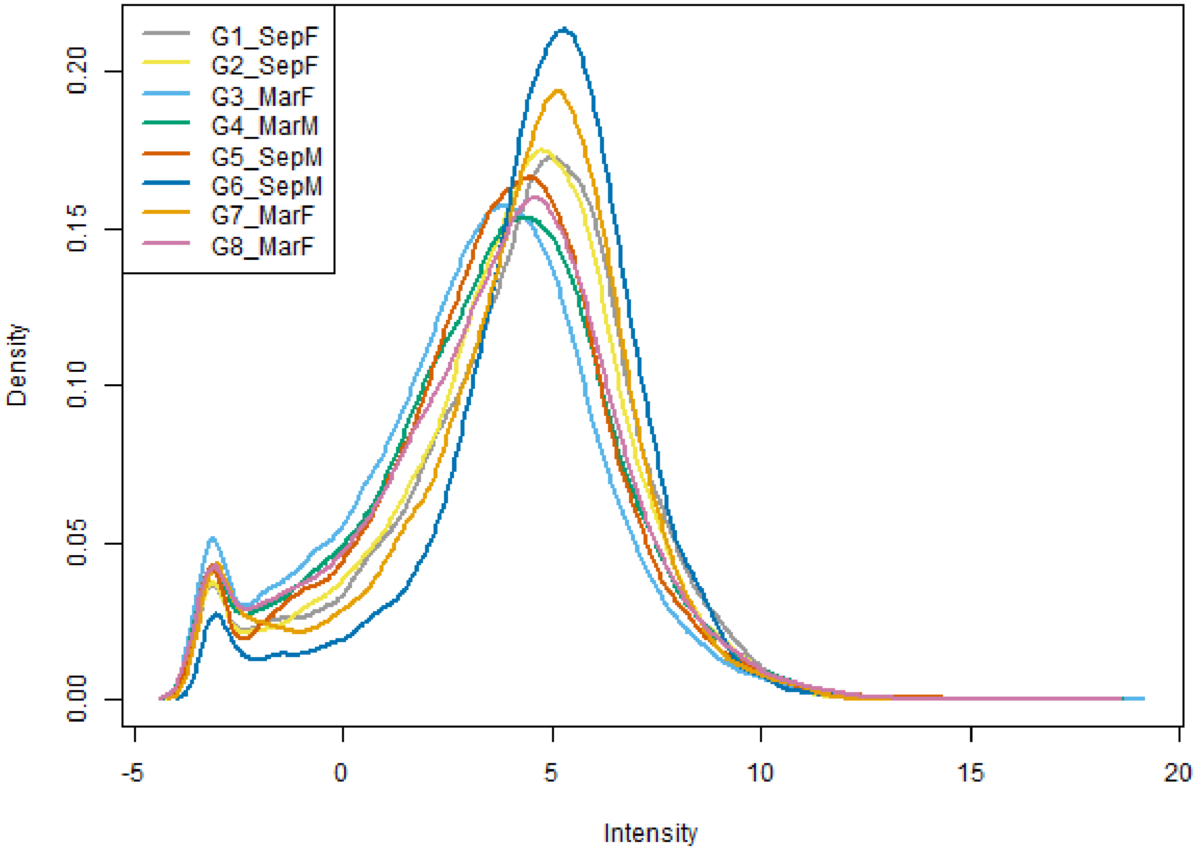


Figure 22: Log Counts per million Densities, calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

Visualise library sizes per sample

```
# The names argument tells the barplot to use the sample
# names on the x-axis The las argument rotates the axis
# names export the plot as png
png("Barplot.LibSize_Season8.png", width = 600)
barplot(dge_Season8$samples$lib.size, names = colnames(dge_Season8),
    las = 2, col = myPalette8)
# title('Barplot of library sizes')
```

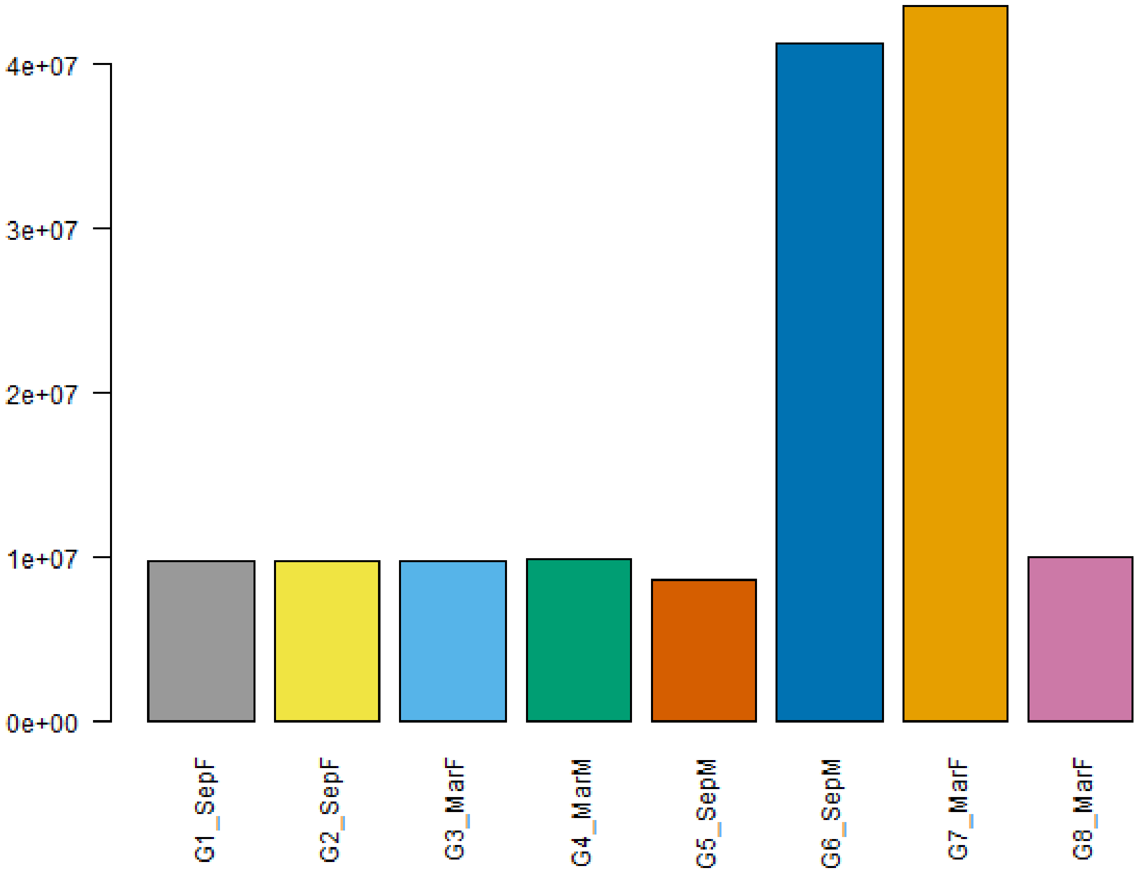


Figure 23: Library size of read counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

Visualise transcript total estimated counts per sample

```r
# export the plot as png
png("Boxplot.Counts_Season8.png", width = 600)
boxplot(dge_Season8$counts, col = myPalette8, las = 2, legend = TRUE)
# title('Boxplot of total unfiltered Counts')
```

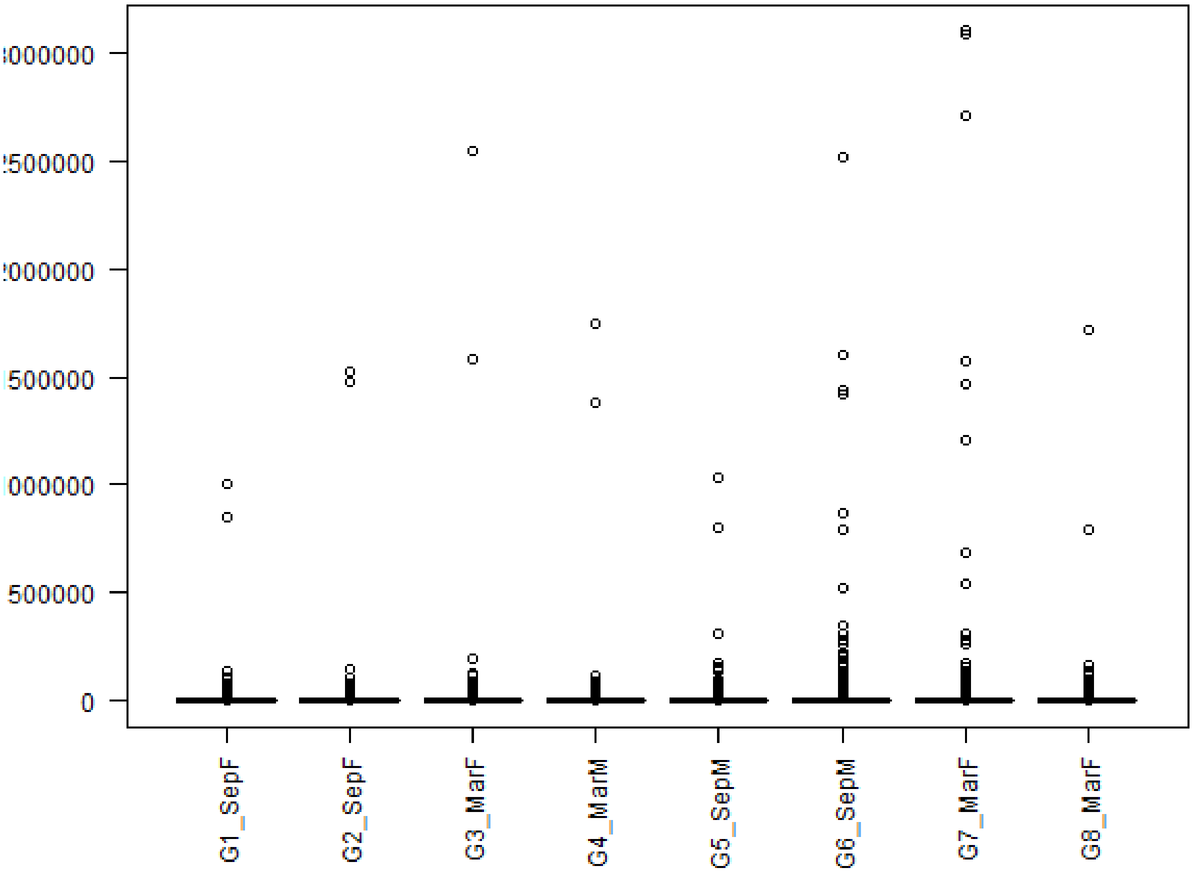


Figure 24: Total estimated counts, calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

```r
# Get log2 counts per million
logcounts_Season8 <- cpm(dge_Season8, log = TRUE)
# Check distributions of samples using boxplots export the
# plot as png
png("boxplot.CPM_Season8.png", width = 600)
boxplot(logcounts_Season8, xlab = "", ylab = "Log2 counts per million",
    las = 2, col = myPalette8, legend = TRUE)
# Let's add a blue horizontal line that corresponds to the
# median logCPM
abline(h = median(logcounts_Season8), col = myPalette8, legend = TRUE)
# title('Boxplots of log Counts Per Million')
```

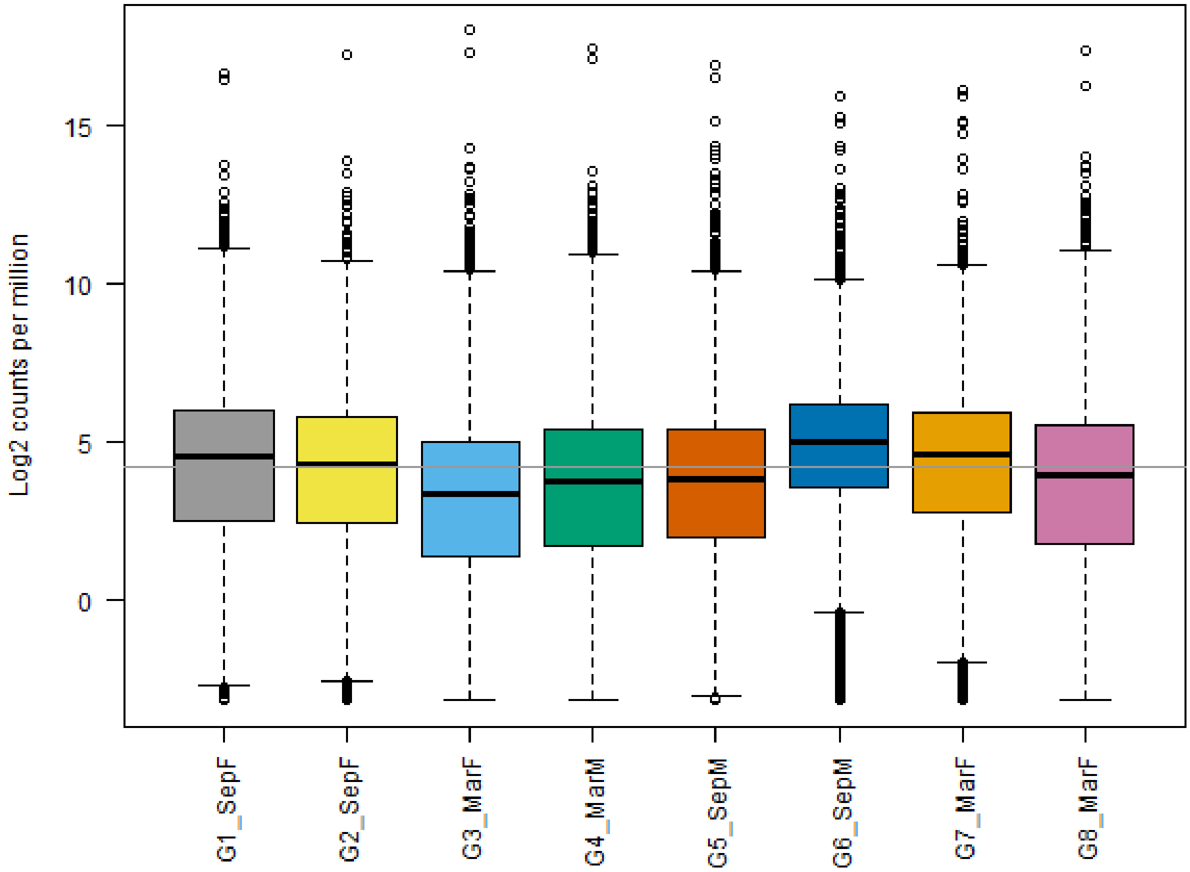


Figure 25: Log2 of estimated counts per million, calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

Visualise sample variation using an MDS plot

```r
# export the plot as png
png("MDS_Season8.png", width = 600)
plotMDS(dge_Season8, col = myPalette8)
# title('MDS plot')
```

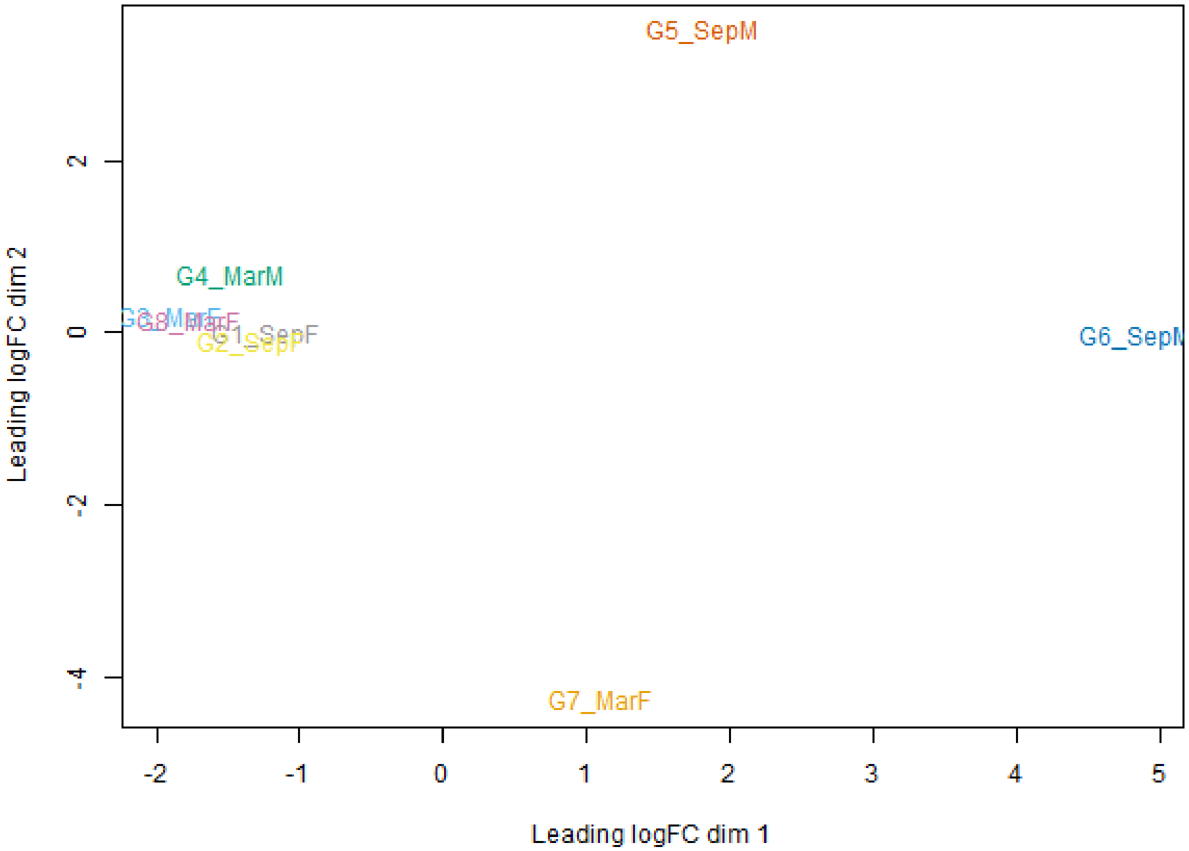


Figure 26: MDS plot of variation among eight *T. adelaidensis* individuals. Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

5.5.3 Calculations

```
# Data with actual counts, not TPM = 'dge'
# head(dge_Season8) head(myTPM_Season8)
```

filter out genes with less than 5 reads in 2 samples for each transcript

```
filter_Season8 <- apply(dge_Season8, 1, function(x) length(x[x >
    5]) >= 2)
filtered_Season8 <- dge_Season8[filter_Season8, ]
# head(filtered_Season8)
```

Run calculations

```
dge_Season8 <- calcNormFactors(dge_Season8)
dge_Season8 <- estimateCommonDisp(dge_Season8)
dge_Season8 <- estimateTagwiseDisp(dge_Season8)
```

```
dge_Season8$common.dispersion
```

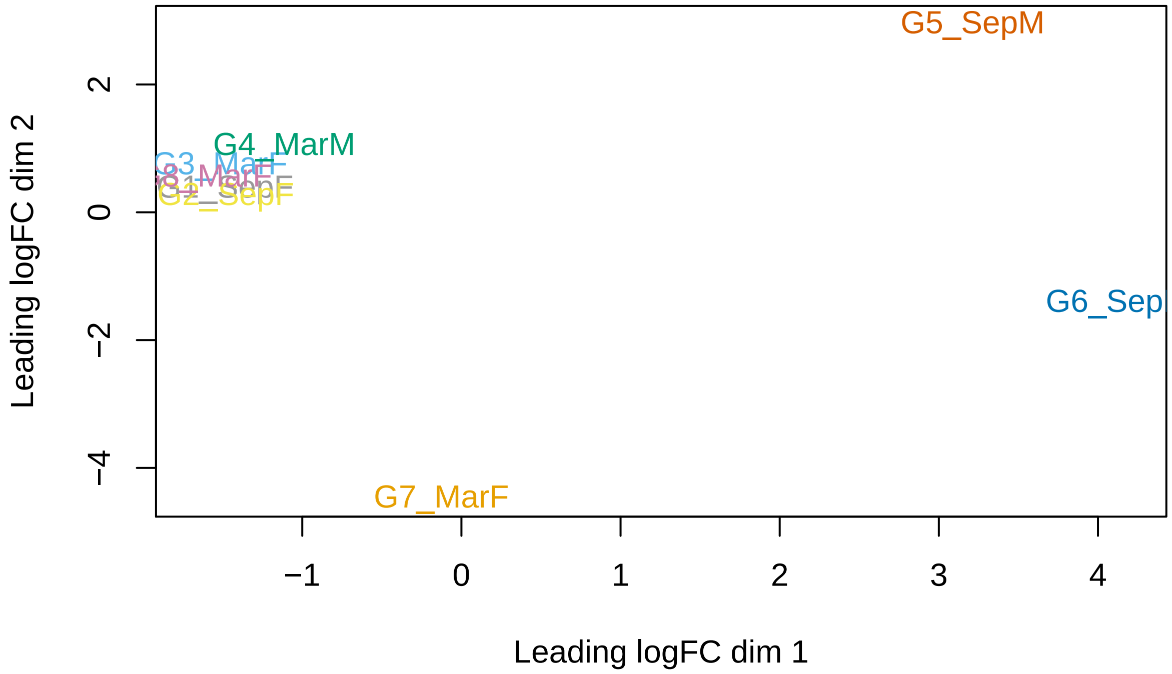
```
## [1] 0.5201665
```

```
head(dge_Season8$tagwise.dispersion)
```

```
## [1] 0.3175575 0.2989024 0.8628780 1.1229636 2.0589261 0.8799119
```

Note that filtering did not change the appearance of the MDS plot

```
plotMDS(dge_Season8, col = myPalette8)
```



5.5.4 Differentially Expressed Genes:

View the top ten ‘differentially expressed genes’

```
results_Season8 <- exactTest(dge_Season8)
topTags(results_Season8)
```

```
## Comparison of groups:  2-1
##                                                        Length EffectiveLength
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418      2000       1677.7106
## PB.3353.1|7cf053|path5:1-7730(+)|transcript/10           7701       7378.7106
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825    1608       1285.7106
## PB.1055.1|295311|path2:1-1979(+)|transcript/12857        1976       1653.7106
## PB.3353.2|7cf053|path5:2504-7730(+)|transcript/274       5192       4869.7106
## PB.3029.5|6fe444|path1:5-2780(+)|transcript/5123         2524       2201.7106
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227       1771       1448.7106
## PB.6088.16|e88f16|path1:1012-2792(+)|transcript/15476    1780       1457.7106
## PB.621.2|172e3f|path6:3-1224(+)|transcript/21583         1227        904.7106
## PB.840.1|1eedc5|path0:1-2091(+)|transcript/11690         2059       1736.7106
##                                                        Overdispersion     logFC
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418          1.767895 -3.568347
## PB.3353.1|7cf053|path5:1-7730(+)|transcript/10               2.543874 -6.995545
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825        1.000000 -5.135802
## PB.1055.1|295311|path2:1-1979(+)|transcript/12857            1.000000 -6.242521
```

```
## PB.3353.2|7cf053|path5:2504-7730(+)|transcript/274          4.234974  -7.381847
## PB.3029.5|6fe444|path1:5-2780(+)|transcript/5123            1.124468  -9.597014
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227          1.416655 -10.043618
## PB.6088.16|e88f16|path1:1012-2792(+)|transcript/15476       2.382893  -8.787069
## PB.621.2|172e3f|path6:3-1224(+)|transcript/21583            1.363178 -13.764707
## PB.840.1|1eedc5|path0:1-2091(+)|transcript/11690            1.000000  -7.383704
##                                                              logCPM       PValue
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418   5.173178 4.544680e-09
## PB.3353.1|7cf053|path5:1-7730(+)|transcript/10        5.885474 2.200857e-07
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825 2.901392 2.927072e-07
## PB.1055.1|295311|path2:1-1979(+)|transcript/12857     4.710647 6.060042e-07
## PB.3353.2|7cf053|path5:2504-7730(+)|transcript/274    4.735994 6.095150e-07
## PB.3029.5|6fe444|path1:5-2780(+)|transcript/5123      7.625550 6.300014e-07
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227    7.822567 6.708151e-07
## PB.6088.16|e88f16|path1:1012-2792(+)|transcript/15476 6.838386 7.879104e-07
## PB.621.2|172e3f|path6:3-1224(+)|transcript/21583      5.355191 9.360938e-07
## PB.840.1|1eedc5|path0:1-2091(+)|transcript/11690      5.001310 1.026803e-06
##                                                                FDR
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418   4.459695e-05
## PB.3353.1|7cf053|path5:1-7730(+)|transcript/10        9.286326e-04
## PB.7745.1|transcript/17825:1-1637(+)|transcript/17825 9.286326e-04
## PB.1055.1|295311|path2:1-1979(+)|transcript/12857     9.286326e-04
## PB.3353.2|7cf053|path5:2504-7730(+)|transcript/274    9.286326e-04
## PB.3029.5|6fe444|path1:5-2780(+)|transcript/5123      9.286326e-04
## PB.4788.1|b34d56|path10:2-1859(+)|transcript/15227    9.286326e-04
## PB.6088.16|e88f16|path1:1012-2792(+)|transcript/15476 9.286326e-04
## PB.621.2|172e3f|path6:3-1224(+)|transcript/21583      9.286326e-04
## PB.840.1|1eedc5|path0:1-2091(+)|transcript/11690      9.286326e-04
```

Output summary expression data

```r
dim(results_Season8)
```

```
## [1] 9813    3
```

```r
summary(de_Season8 <- decideTests(results_Season8))
```

```
##         2-1
## Down    210
## NotSig 9598
## Up        5
```

```r
Summary_Season8 <- (de_Season8 <- decideTests(results_Season8))

write.table(Summary_Season8, file = "summary_Season8.txt")

detags_Season8 <- rownames(dge_Season8)[as.logical(de_Season8)]
```

Visualise with a plotsmear

```
detags_Season8 <- rownames(dge_Season8)[as.logical(de_Season8)]
# export the plot as png
png("plotSmear_Season8.png", width = 600)
plotSmear(results_Season8, de.tags = detags_Season8)
```

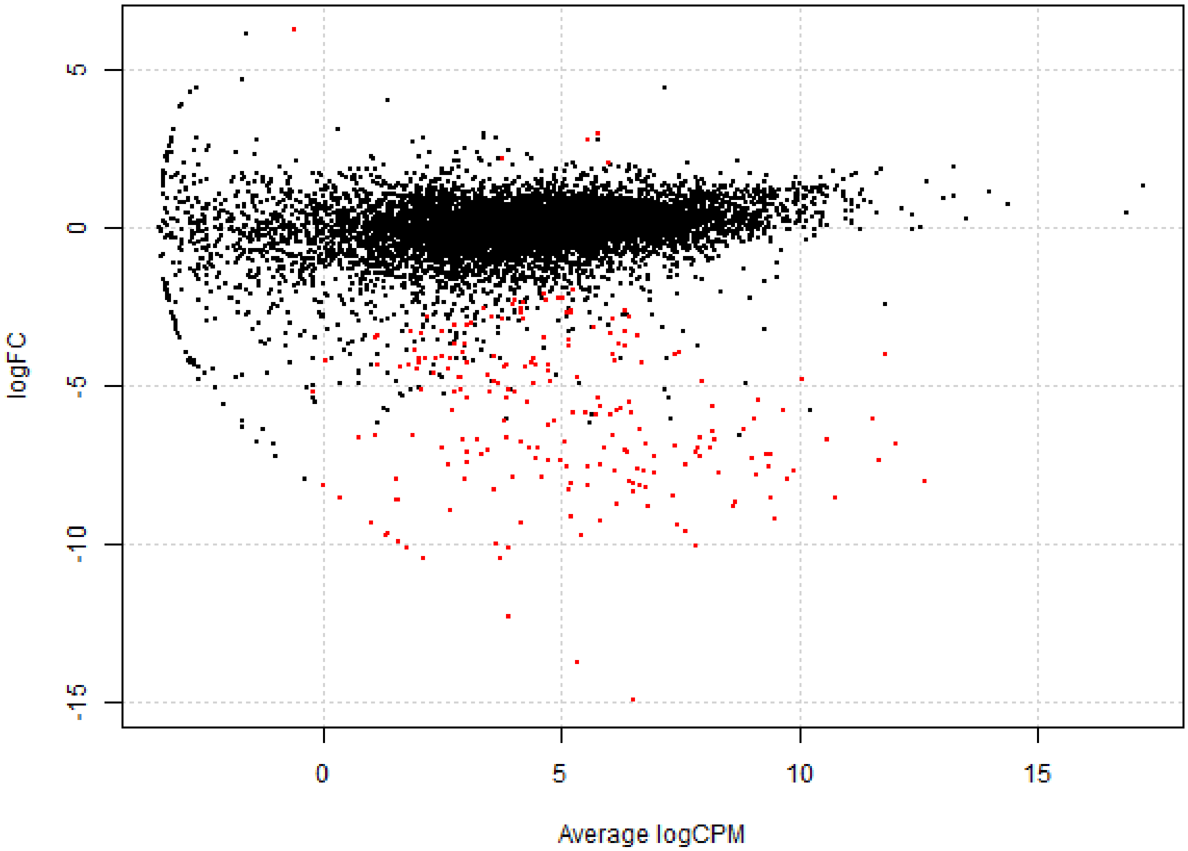


Figure 27: Log Fold Change vs average Log counts per million for expression level comparing March to September among eight *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change over season, with p-values <0.05. Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

Visualise with a volcano plot

```
# export the plot as png
png("volcanoData_Season8.png", width = 600)
volcanoData_Season8 <- cbind(results_Season8$table$logFC,
↪  -log10(results_Season8$table$PValue))
colnames(volcanoData_Season8) <- c("logFC", "negLogPval")
DEGs_Season8 <- results_Season8$table$PValue < 0.05 & abs(results_Season8$table$logFC) >
    2
point.col_Season8 <- ifelse(DEGs_Season8, "red", "black")
plot(volcanoData_Season8, pch = 16, col = point.col_Season8,
    cex = 0.5)
```

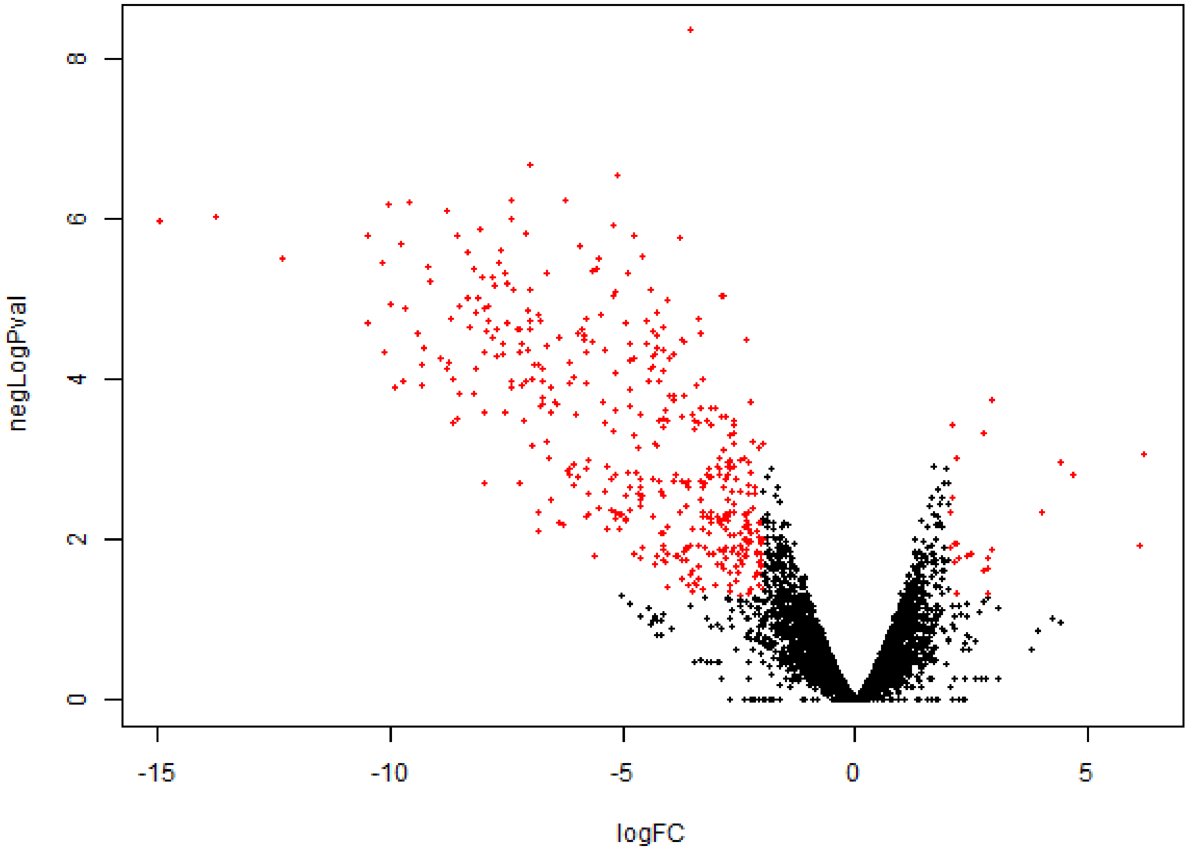


Figure 28: Negative Log of the p-value against Log Fold Change in expression level comparing March to September among eight *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change over season, with p-values <0.05 and *also* a log fold change >2. Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

5.5.5 Group Factor: Sex

Create the EdgeR DGE list

```
dge_Sex8 <- DGEList(counts = Kcaught8$counts/Kcaught8$annotation$Overdispersion,
    genes = Kcaught8$annotation, group = sex8_group)
# View(dge_Sex8)
names(dge_Sex8)
```

```
## [1] "counts"  "samples" "genes"
```

```
dge_Sex8
```

```
## An object of class "DGEList"
## $counts
##                                                       G1_SepF     G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      278.00000  218.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    1208.00000 1143.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       0.00000    5.206985
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       44.12913   56.325347
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.00000    0.000000
##                                                       G3_MarF     G4_MarM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      99.000000  170.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    990.000000 1467.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      2.445102    1.920669
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      20.417263   17.146414
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
##                                                       G5_SepM    G6_SepM
## PB.2.1|002537|path0:1-1624(+)|transcript/18304     131.000000  793.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    652.000000 9810.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      5.249977   75.81151
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      58.438180 6815.29409
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000   62.63286
##                                                        G7_MarF     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       561.00000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    12443.00000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       57.12165    4.317232
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      3215.92340   14.869331
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    55.19076    0.000000
## 9808 more rows ...
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1  9760098            1
## G2_SepF     1  9760939            1
## G3_MarF     1  9745167            1
## G4_MarM     2  9855867            1
## G5_SepM     2  8598755            1
## G6_SepM     2 41290113            1
## G7_MarF     1 43632769            1
## G8_MarF     1 10009376            1
##
```

```
## $genes
##                                                       Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304          1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564         2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534         1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426          2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401      1965        1642.711
##                                                       Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304              1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564             1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534             2.960918
## PB.11.1|004815|path2:1-3039(+)|transcript/3426              1.359646
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401          2.524871
## 9808 more rows ...
```

```
# Obtain Counts Per Million
myTPM_Sex8 <- dge_Sex8$counts
# Have a look at the output head(myTPM_Sex8)
```

5.5.6 Initial Data Exploration

```
# Which values in myCPM are greater than 0.5?
thresh_Sex8 <- myTPM_Sex8 > 0.5
# This produces a logical matrix with TRUEs and FALSEs
# head(thresh_Sex8)
```

Raw Counts per million and library sizes will be consistent with the Season comparison above. There is no need to plot them again however scripts are included to create separate R objects for downstream application.

CPM distribution with logged CPM

```
# export the plot as png
png("plotDensities_Sex8.png", width = 600)
unfilteredExpr_Sex8 <- cpm(dge_Sex8, log = T)
plotDensities(unfilteredExpr_Sex8, col = myPalette8, legend = TRUE)
```

Library sizes per sample

```
# The names argument tells the barplot to use the sample
# names on the x-axis The las argument rotates the axis
# names export the plot as png
png("Barplot.LibSize_Sex8.png", width = 600)
barplot(dge_Sex8$samples$lib.size, names = colnames(dge_Sex8),
    las = 2, col = myPalette8)
# title('Barplot of library sizes')
```

Transcript total estimated counts per sample

```
# export the plot as png
png("Boxplot.Counts_Sex8.png", width = 600)
boxplot(dge_Sex8$counts, col = myPalette8, las = 2, legend = TRUE)
# title('Boxplot of total unfiltered Counts')
```

Logged total estimated transcript counts per sample

```
# Get log2 counts per million
logcounts_Sex8 <- cpm(dge_Sex8, log = TRUE)
# Check distributions of samples using boxplots export the
# plot as png
png("boxplot.CPM_Sex8.png", width = 600)
boxplot(logcounts_Sex8, xlab = "", ylab = "Log2 counts per million",
    las = 2, col = myPalette8, legend = TRUE)
# Let's add a blue horizontal line that corresponds to the
# median logCPM
abline(h = median(logcounts_Sex8), col = myPalette8, legend = TRUE)
# title('Boxplots of log Counts Per Million')
```

5.5.7 Calculations

```r
# Data with actual counts, not TPM = 'dge' head(dge_Sex8)
# head(myTPM_Sex8)
```

filter out genes with less than 5 reads in 2 samples for each transcript

```r
filter_Sex8 <- apply(dge_Sex8, 1, function(x) length(x[x > 5]) >=
    2)
filtered_Sex8 <- dge_Sex8[filter_Sex8, ]
# head(filtered_Sex8)
```

Run calculations

```r
dge_Sex8 <- calcNormFactors(dge_Sex8)
dge_Sex8 <- estimateCommonDisp(dge_Sex8)
dge_Sex8 <- estimateTagwiseDisp(dge_Sex8)
```

```r
dge_Sex8$common.dispersion
```

```
## [1] 0.4957059
```

```r
head(dge_Sex8$tagwise.dispersion)
```

```
## [1] 0.3077159 0.3081643 0.8345605 1.0635014 2.0240359 0.8801331
```

5.5.8 Differentially Expressed Genes:

View the top ten 'differentially expressed genes'

```r
results_Sex8 <- exactTest(dge_Sex8)
topTags(results_Sex8)
```

```
## Comparison of groups:  2-1
##                                                       Length EffectiveLength
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021    1103        780.7106
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195    1136        813.7106
## PB.3895.1|91ba83|path6:1-1594(+)|transcript/18563    1376       1053.7106
## PB.6451.1|f62586|path2:1-2617(+)|transcript/23694     736        413.9638
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575      3066       2743.7106
## PB.4153.1|9a8a98|path29:1-1042(+)|transcript/22949    862        539.7749
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901      896        573.7631
## PB.2493.6|5d0eeb|path5:27-5305(+)|transcript/20823   1337       1014.7106
## PB.6452.1|f62586|path3:1-2621(+)|transcript/24182     633        311.4422
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860      905        582.7631
##                                                       Overdispersion    logFC
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021        3.909377 8.427189
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195        1.778063 8.498502
## PB.3895.1|91ba83|path6:1-1594(+)|transcript/18563        3.281779 9.877587
## PB.6451.1|f62586|path2:1-2617(+)|transcript/23694        2.206536 8.379631
```

```
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575          1.000000 8.825922
## PB.4153.1|9a8a98|path29:1-1042(+)|transcript/22949       2.017518 7.786646
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901         1.851161 8.076026
## PB.2493.6|5d0eeb|path5:27-5305(+)|transcript/20823       7.349539 9.470585
## PB.6452.1|f62586|path3:1-2621(+)|transcript/24182        2.348219 8.535088
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860         1.155099 7.594306
##                                                       logCPM       PValue
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021  5.581957 5.041529e-32
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195  8.808743 4.881766e-27
## PB.3895.1|91ba83|path6:1-1594(+)|transcript/18563  7.290776 8.269517e-19
## PB.6451.1|f62586|path2:1-2617(+)|transcript/23694  5.379825 1.467057e-18
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575    4.283730 3.452531e-18
## PB.4153.1|9a8a98|path29:1-1042(+)|transcript/22949 4.904003 3.111298e-17
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901   6.272961 8.781115e-17
## PB.2493.6|5d0eeb|path5:27-5305(+)|transcript/20823 5.652139 7.496662e-16
## PB.6452.1|f62586|path3:1-2621(+)|transcript/24182  3.542180 1.769362e-15
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860   7.555428 3.678419e-15
##                                                             FDR
## PB.2770.1|6751e7|path1:1-1210(+)|transcript/22021  4.947252e-28
## PB.2769.1|6751e7|path0:4-1327(+)|transcript/21195  2.395239e-23
## PB.3895.1|91ba83|path6:1-1594(+)|transcript/18563  2.704959e-15
## PB.6451.1|f62586|path2:1-2617(+)|transcript/23694  3.599057e-15
## PB.979.1|25edbb|path0:1-3100(+)|transcript/3575    6.775938e-15
## PB.4153.1|9a8a98|path29:1-1042(+)|transcript/22949 5.088528e-14
## PB.4152.1|9a8a98|path2:1-992(+)|transcript/22901   1.230987e-13
## PB.2493.6|5d0eeb|path5:27-5305(+)|transcript/20823 9.195593e-13
## PB.6452.1|f62586|path3:1-2621(+)|transcript/24182  1.929195e-12
## PB.637.1|1826ff|path3:1-1155(+)|transcript/22860   3.609632e-12
```

Output summary expression data

```r
dim(results_Sex8)
```

```
## [1] 9813    3
```

```r
summary(de_Sex8 <- decideTests(results_Sex8))
```

```
##         2-1
## Down      2
## NotSig 9490
## Up      321
```

```r
Summary_Sex8 <- (de_Sex8 <- decideTests(results_Sex8))

write.table(Summary_Sex8, file = "summary_Sex8.txt")

detags_Sex8 <- rownames(dge_Sex8)[as.logical(de_Sex8)]
```

Visualise with a plotsmear

```
# export the plot as png
png("plotSmear_Sex8.png", width = 600)
plotSmear(results_Sex8, de.tags = detags_Sex8)
```

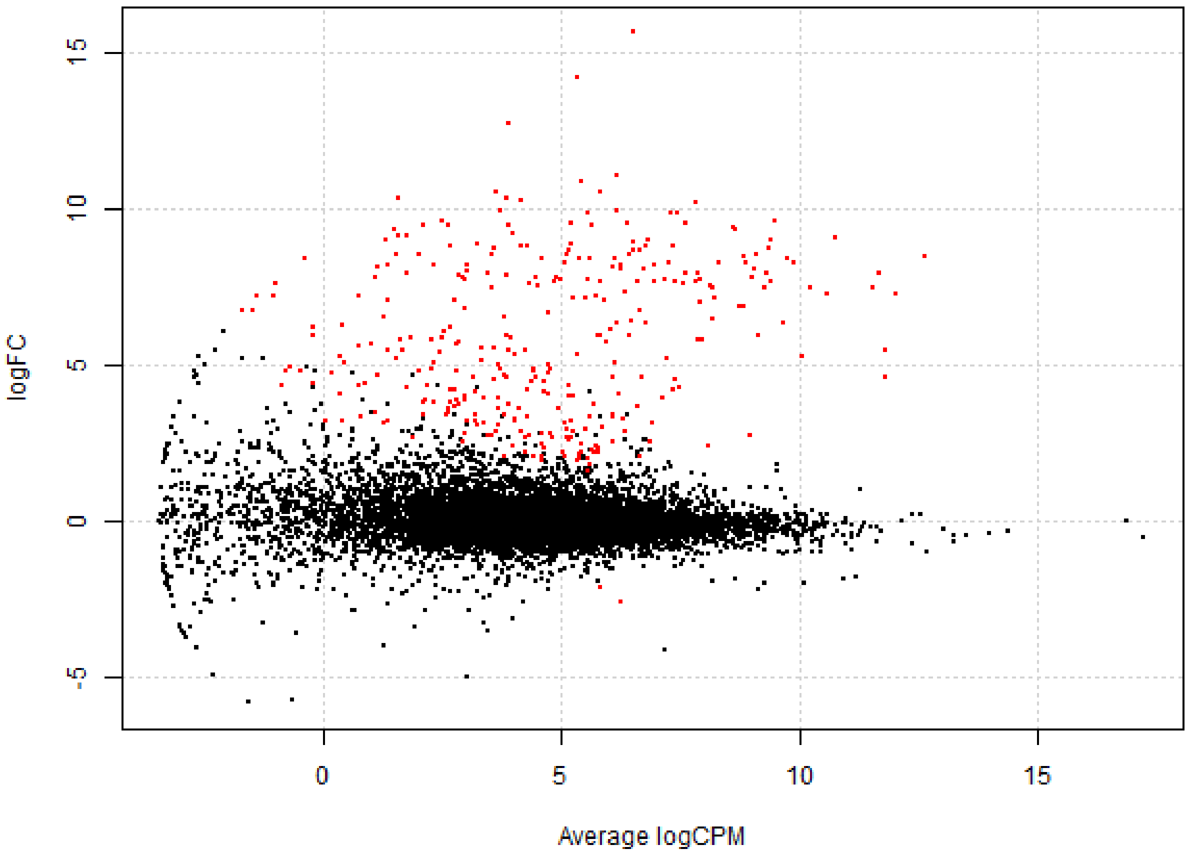


Figure 29: Log Fold Change vs average Log counts per million for expression level comparing Male to Female as variables among eight *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change between sexes, with p-values <0.05. Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

Visualise with a volcano plot

```
# export the plot as png
png("volcanoData_Sex8.png", width = 600)
volcanoData_Sex8 <- cbind(results_Sex8$table$logFC, -log10(results_Sex8$table$PValue))
colnames(volcanoData_Sex8) <- c("logFC", "negLogPval")
DEGs_Sex8 <- results_Sex8$table$PValue < 0.05 & abs(results_Sex8$table$logFC) >
    2
point.col_Sex8 <- ifelse(DEGs_Sex8, "red", "black")
plot(volcanoData_Sex8, pch = 16, col = point.col_Sex8, cex = 0.5)
```

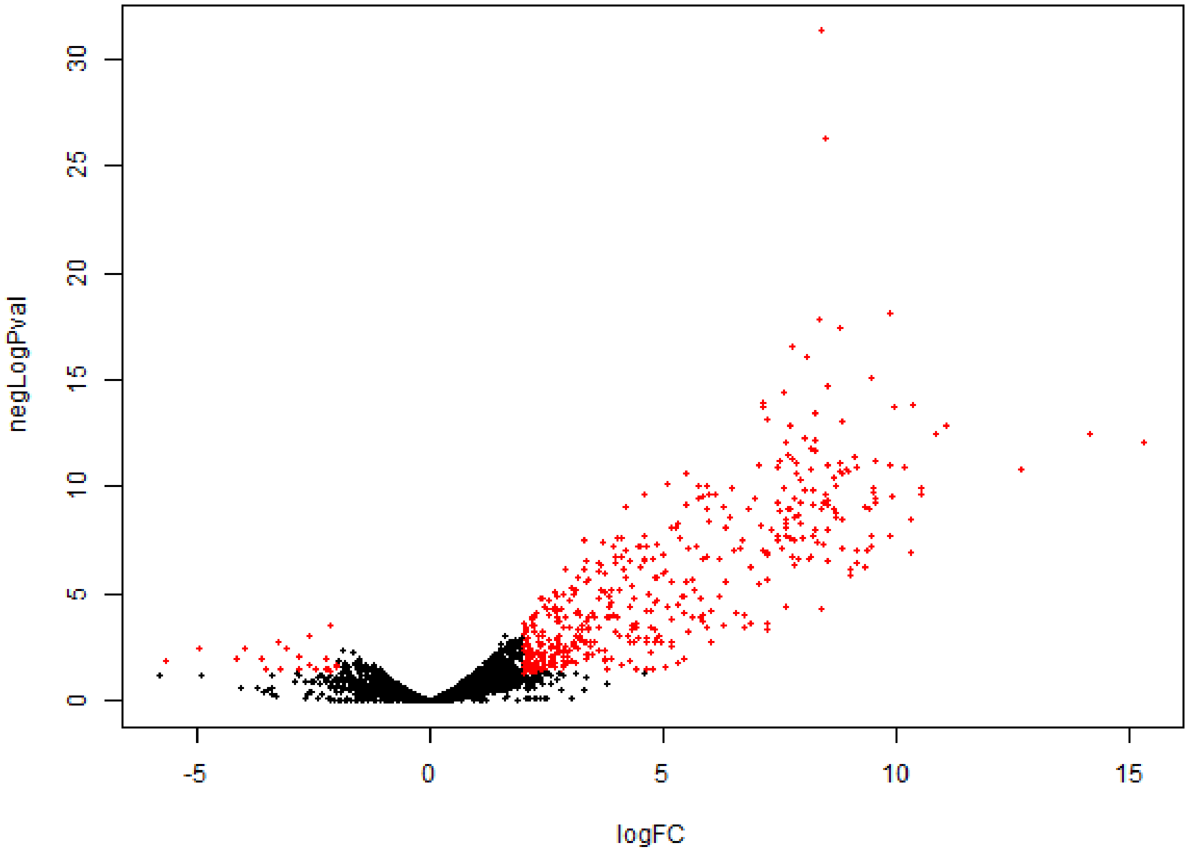


Figure 30: Negative Log of the p-value against Log Fold Change in expression level comparing Male to Female as variables among eight *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change between sexes, with p-values <0.05 and *also* a log fold change >2. Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of eight *Tiliqua adelaidensis* individuals, four collected in autumn and four collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts. Samples G6 and G7 are subject to batch effects due to different cDNA synthesis methods and sequencing runs.

5.6 Four *T. adelaidensis* individuals collected between two seasonal periods.

Exploration of 2:2 Female samples by Season group factor:

Skink G5 represents a September Male, and is also markedly different from the remaining 5 samples on the MDS plot when 6 samples are compared in groups of 3. There is a chance G5 is responsible for a large amount of the observed variation in expression, and the skew in the data towards higher expressed genes. These Four females represent two from September, and two from March/April, and were sequenced on the same run. This section is purely exploratory on how results may have remained consistent or markedly different on the removal of sample G5 (and on balance, also the other male G4).

Import the Data:

```
# create paths
setwd("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr-4/kallisto-clstr/")
DIR4 <- ("~/@Uni/@Flinders University
↪  PhD/RWorkingDir/R-kallisto-clstr-4/kallisto-clstr/")
head(DIR4)
```

```
## [1] "~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr-4/kallisto-clstr/"
```

```
paths4F <- list.dirs(path = DIR4, full.names = FALSE, recursive = FALSE)
# the working dir only includes kallisto output for the
# kidneys at this time View(paths4F)

Kcaught4F <- catchKallisto(paths4F, verbose = TRUE)
```

```
## Reading G1_SepF, 9813 transcripts, 100 bootstraps
## Reading G2_SepF, 9813 transcripts, 100 bootstraps
## Reading G3_MarF, 9813 transcripts, 100 bootstraps
## Reading G8_MarF, 9813 transcripts, 100 bootstraps
```

```
setwd("~/@Uni/@Flinders University PhD/RWorkingDir/R-kallisto-clstr-4")
# View(Kcaught4F)
```

Set the group factors. Input Sample files are listed G1-G8 Group factors are applied to the DGElist object and are the point of comparison for expression analyses.

Season of Collection: 1 = September, 2 = March For the purposes of consistent labelling per group factors some April collections are referred to in the March group. Accurate collection information and dates are outlined in the methods chapter.

Individual Sex: 1 = Female, 2 = Male

```
# these factors correlate to season of collection. 1 =
# March, 2 = Sep.
Season4F_group <- factor(c(1, 1, 2, 2))
```

5.6.1 Group Factor: Season

Create the EdgeR DGE list

```r
dge_Season4F <- DGEList(counts = Kcaught4F$counts/Kcaught4F$annotation$Overdispersion,
    genes = Kcaught4F$annotation, group = Season4F_group)
# View(dge_Season4F)
names(dge_Season4F)
```

```
## [1] "counts"  "samples" "genes"
```

```r
dge_Season4F
```

```
## An object of class "DGEList"
## $counts
##                                                      G1_SepF    G2_SepF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      278.00000  218.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    1208.00000 1143.00000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534       0.00000    4.88473
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       41.70711   53.23394
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401    0.00000    0.00000
##                                                      G3_MarF     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304      99.000000  225.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564    990.000000 2302.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      2.293777    4.050043
## PB.11.1|004815|path2:1-3039(+)|transcript/3426      19.296666   14.053231
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   0.000000    0.000000
## 9808 more rows ...
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10091726            1
## G2_SepF     1 10026494            1
## G3_MarF     2  9999768            1
## G8_MarF     2 10309597            1
##
## $genes
##                                                    Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304       1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564      2065        1742.711
## PB.10.1|004815|path1:5-1713(+)|transcript/17534      1612        1289.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426       2999        2676.711
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401   1965        1642.711
##                                                    Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304           1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564          1.000000
## PB.10.1|004815|path1:5-1713(+)|transcript/17534          3.156256
## PB.11.1|004815|path2:1-3039(+)|transcript/3426           1.438604
## PB.11.2|004815|path2:1047-3035(+)|transcript/13401       1.000000
## 9808 more rows ...
```

Obtain Counts Per Million

```r
myTPM_Season4F <- dge_Season4F$counts
# head(myTPM_Season4F)
```

5.6.2 Initial Data Exploration

```
# Which values in myCPM are greater than 0.5?
thresh_Season4F <- myTPM_Season4F > 0.5
# This produces a logical matrix with TRUEs and FALSEs
# head(thresh_Season4F)
```

Insert colour pallate for following visualisations consistency

```
myPalette4 <- c("#999999", "#F0E442", "#56B4E9", "#CC79A7")
```

Plot CPM distribution with logged CPM

```
# export the plot as png
png("plotDensities_Season4F.png", width = 600)
unfilteredExpr_Season4F <- cpm(dge_Season4F, log = T)
plotDensities(unfilteredExpr_Season4F, col = myPalette4, legend = TRUE)
```

Visualise library sizes per sample

```
# The names argument tells the barplot to use the sample
# names on the x-axis The las argument rotates the axis
# names export the plot as png
png("Barplot.LibSize_Season4F.png", width = 600)
barplot(dge_Season4F$samples$lib.size, names = colnames(dge_Season4F),
    las = 2, col = myPalette4)
# title('Barplot of library sizes')
```

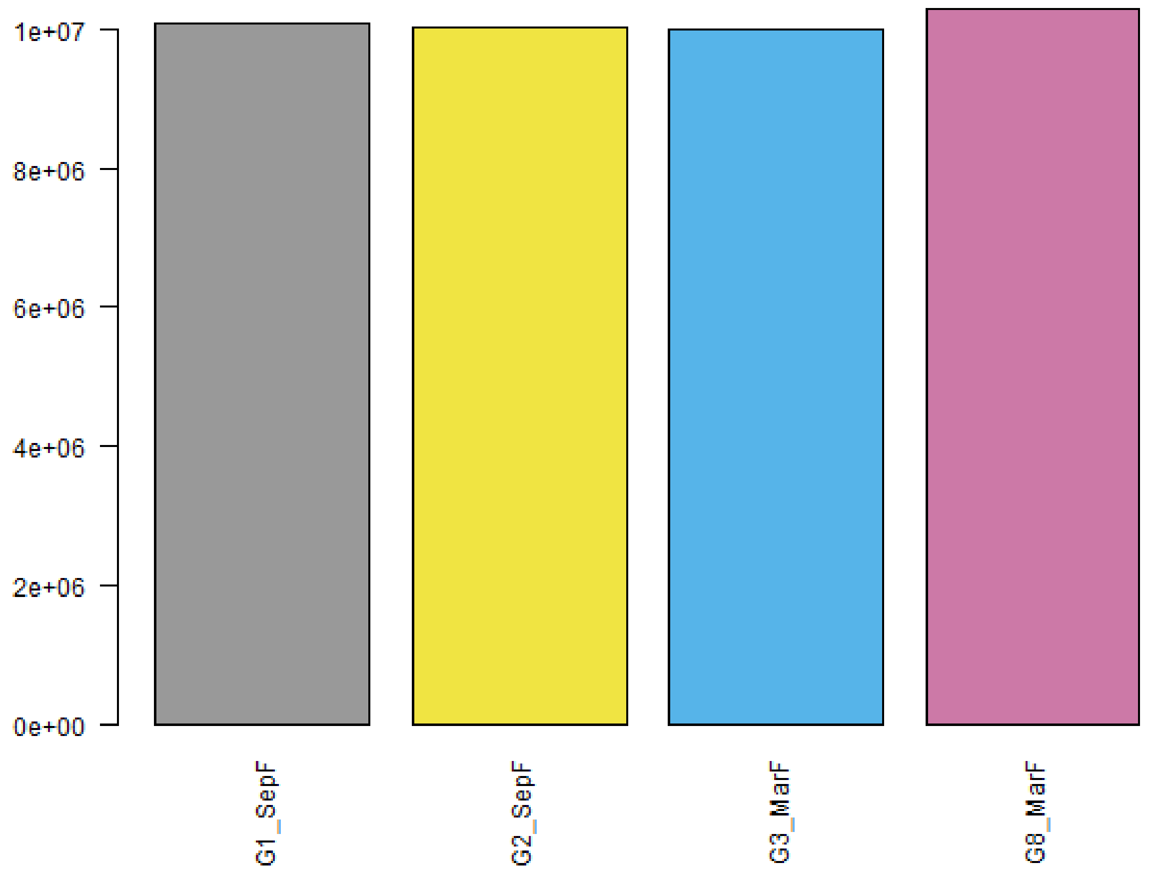


Figure 31: Library size of read counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of four female *Tiliqua adelaidensis* individuals, two collected in autumn and two collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts (note G6 was male). Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Visualise transcript total estimated counts per sample

```r
# export the plot as png
png("Boxplot.Counts_Season4F.png", width = 600)
boxplot(dge_Season4F$counts, col = myPalette4, las = 2, legend = TRUE)
# title('Boxplot of total unfiltered Counts')
```

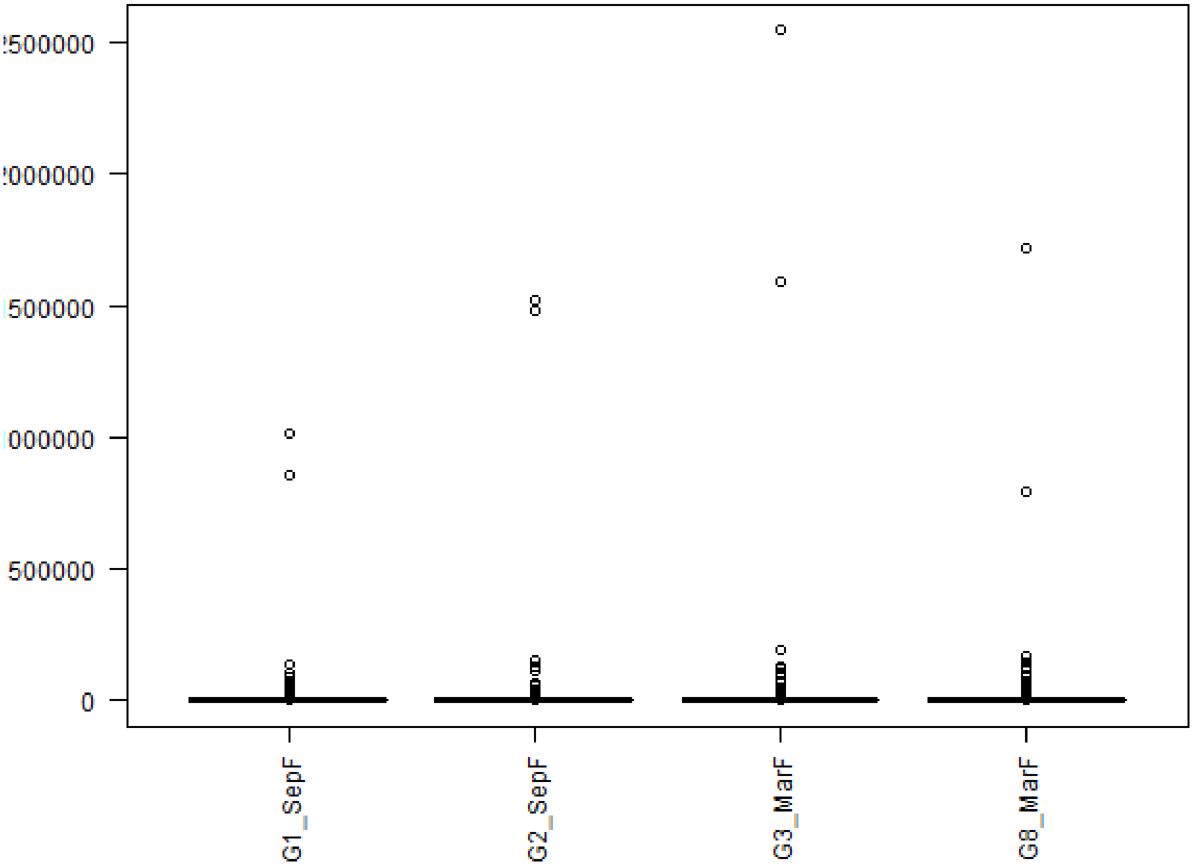


Figure 32: Total estimated counts, calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of four female *Tiliqua adelaidensis* individuals, two collected in autumn and two collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts (note G6 was male). Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

```r
# Get log2 counts per million
logcounts_Season4F <- cpm(dge_Season4F, log = TRUE)
# Check distributions of samples using boxplots export the
# plot as png
png("boxplot.CPM_Season4F.png", width = 600)
boxplot(logcounts_Season4F, xlab = "", ylab = "Log2 counts per million",
    las = 2, col = myPalette4, legend = TRUE)
# Let's add a blue horizontal line that corresponds to the
# median logCPM
abline(h = median(logcounts_Season4F), col = myPalette4, legend = TRUE)
# title('Boxplots of log Counts Per Million')
```

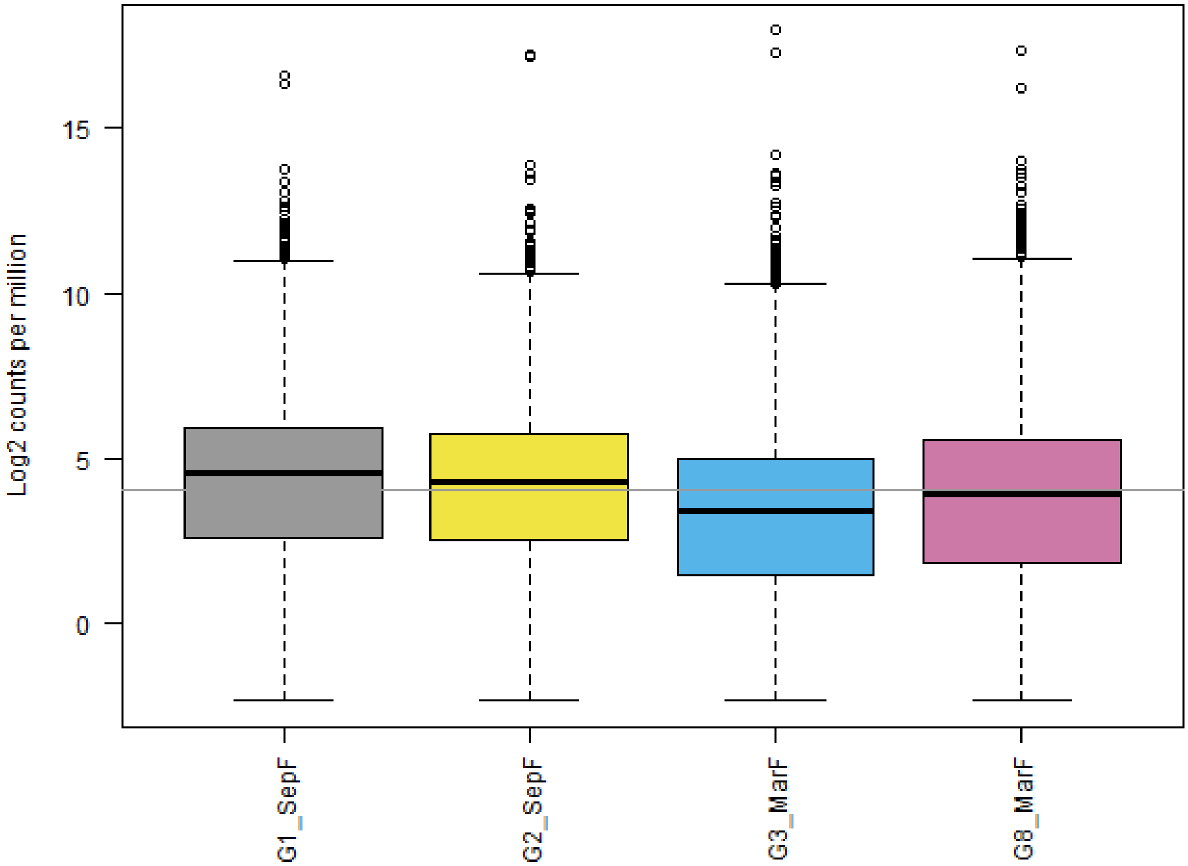


Figure 33: Log2 of estimated counts per million, calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of four female *Tiliqua adelaidensis* individuals, two collected in autumn and two collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts (note G6 was male). Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

Visualise sample variation using an MDS plot

```
# export the plot as png
png("MDS_Season4F.png", width = 600)
plotMDS(dge_Season4F, col = myPalette4)
# title('MDS plot')
```

Figure 34: MDS plot of variation among four *T. adelaidensis* individuals. Counts calculated using Kallisto and displayed in EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of four female *Tiliqua adelaidensis* individuals, two collected in autumn and two collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts (note G6 was male). Season and Sex are indicated in labels: Sep = September (spring), Mar = March (autumn), F =Female, M=Male.

5.6.3 Calculations

```
# Data with actual counts, not TPM = 'dge'
# head(dge_Season4F) head(myTPM_Season4F)
```

filter out genes with less than 5 reads in 2 samples for each transcript

```
filter_Season4F <- apply(dge_Season4F, 1, function(x) length(x[x >
    5]) >= 2)
filtered_Season4F <- dge_Season4F[filter_Season4F, ]
head(filtered_Season4F)
```

```
## An object of class "DGEList"
## $counts
##                                                     G1_SepF    G2_SepF   G3_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304   278.00000  218.00000  99.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 1208.00000 1143.00000 990.00000
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    41.70711   53.23394  19.29667
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993   108.66377   68.97787  18.89805
## PB.14.1|006283|path0:1-1955(+)|transcript/13821  964.49077  655.21013 455.87608
## PB.15.1|006e23|path0:1-3614(+)|transcript/1774   725.36411  637.30364 427.59261
##                                                     G8_MarF
## PB.2.1|002537|path0:1-1624(+)|transcript/18304   225.00000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564 2302.00000
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    14.05323
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993    40.63080
## PB.14.1|006283|path0:1-1955(+)|transcript/13821  716.88749
## PB.15.1|006e23|path0:1-3614(+)|transcript/1774   839.75194
##
## $samples
##         group lib.size norm.factors
## G1_SepF     1 10091726            1
## G2_SepF     1 10026494            1
## G3_MarF     2  9999768            1
## G8_MarF     2 10309597            1
##
## $genes
##                                                 Length EffectiveLength
## PB.2.1|002537|path0:1-1624(+)|transcript/18304    1530        1207.711
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564   2065        1742.711
## PB.11.1|004815|path2:1-3039(+)|transcript/3426    2999        2676.711
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993    2714        2391.711
## PB.14.1|006283|path0:1-1955(+)|transcript/13821   1920        1597.711
## PB.15.1|006e23|path0:1-3614(+)|transcript/1774    3583        3260.711
##                                                 Overdispersion
## PB.2.1|002537|path0:1-1624(+)|transcript/18304        1.000000
## PB.3.2|00361c|path0:43-2240(+)|transcript/10564       1.000000
## PB.11.1|004815|path2:1-3039(+)|transcript/3426        1.438604
## PB.13.1|004c10|path2:1-2811(+)|transcript/4993        1.058310
## PB.14.1|006283|path0:1-1955(+)|transcript/13821       1.118725
## PB.15.1|006e23|path0:1-3614(+)|transcript/1774        1.101516
```

Run calculations

```
dge_Season4F <- calcNormFactors(dge_Season4F)
dge_Season4F <- estimateCommonDisp(dge_Season4F)
dge_Season4F <- estimateTagwiseDisp(dge_Season4F)
```

```
dge_Season4F$common.dispersion
```

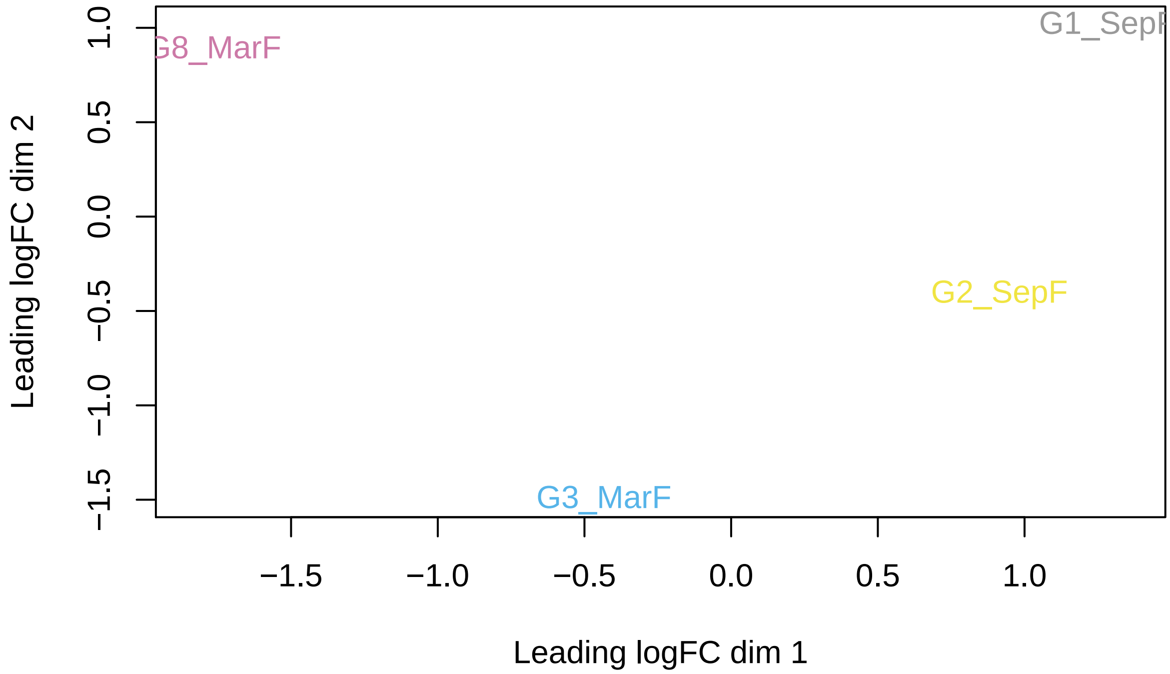
```
## [1] 0.1222519
```

```
head(dge_Season4F$tagwise.dispersion)
```

```
## [1] 0.09703715 0.10226045 0.23568294 0.16190164 0.26578585 0.12841076
```

Note that at this smaller scale filtering *does* change the appearance of the MDS plot

```
plotMDS(dge_Season4F, col = myPalette4)
```



5.6.4 Differentially Expressed Genes:

View the top ten 'differentially expressed genes'

```r
results_Season4F <- exactTest(dge_Season4F)
topTags(results_Season4F)
```

```
## Comparison of groups:  2-1
##                                                    Length EffectiveLength
## PB.4138.1|9a0bd1|path6:1-1549(+)|transcript/20634    1416       1093.7106
## PB.5398.2|cbf34d|path5:582-2207(+)|transcript/17939   1601       1278.7106
## PB.2869.1|6b0af0|path2:1-1117(+)|transcript/22387    1086        763.7106
## PB.4931.3|b8571c|path20:38-3608(+)|transcript/1966   3487       3164.7106
## PB.9559.1|transcript/7471:1-2422(+)|transcript/7471   2391       2068.7106
## PB.876.2|206df6|path10:20-1995(+)|transcript/12799   2024       1701.7106
## PB.3850.1|906dad|path20:1-2023(+)|transcript/12300   1942       1619.7106
## PB.879.1|206df6|path7:41-2171(+)|transcript/12507    1871       1548.7106
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418   2000       1677.7106
## PB.4804.1|b3ae52|path0:150-1779(+)|transcript/17853   1574       1251.7106
##                                                    Overdispersion     logFC
## PB.4138.1|9a0bd1|path6:1-1549(+)|transcript/20634         2.095136  8.181660
## PB.5398.2|cbf34d|path5:582-2207(+)|transcript/17939       2.388134 -7.324372
## PB.2869.1|6b0af0|path2:1-1117(+)|transcript/22387         1.000000 -3.952345
## PB.4931.3|b8571c|path20:38-3608(+)|transcript/1966        5.321449  7.096853
## PB.9559.1|transcript/7471:1-2422(+)|transcript/7471       1.078024 -3.886336
## PB.876.2|206df6|path10:20-1995(+)|transcript/12799        7.027709  3.950243
## PB.3850.1|906dad|path20:1-2023(+)|transcript/12300        1.019013 -4.330238
## PB.879.1|206df6|path7:41-2171(+)|transcript/12507        16.000678  9.507562
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418       1.582753 -3.241687
## PB.4804.1|b3ae52|path0:150-1779(+)|transcript/17853       1.137449 -3.511822
##                                                      logCPM       PValue
## PB.4138.1|9a0bd1|path6:1-1549(+)|transcript/20634   4.329581 2.337337e-25
## PB.5398.2|cbf34d|path5:582-2207(+)|transcript/17939 4.355781 6.015218e-15
## PB.2869.1|6b0af0|path2:1-1117(+)|transcript/22387   5.424124 1.182776e-13
## PB.4931.3|b8571c|path20:38-3608(+)|transcript/1966  1.923894 3.949696e-13
## PB.9559.1|transcript/7471:1-2422(+)|transcript/7471 4.131625 4.351701e-13
## PB.876.2|206df6|path10:20-1995(+)|transcript/12799  3.719088 1.367118e-12
## PB.3850.1|906dad|path20:1-2023(+)|transcript/12300  3.312113 1.460823e-11
## PB.879.1|206df6|path7:41-2171(+)|transcript/12507   2.115742 3.494145e-11
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418 4.879667 4.564669e-11
## PB.4804.1|b3ae52|path0:150-1779(+)|transcript/17853 6.485933 1.011724e-10
##                                                              FDR
## PB.4138.1|9a0bd1|path6:1-1549(+)|transcript/20634   2.293628e-21
## PB.5398.2|cbf34d|path5:582-2207(+)|transcript/17939 2.951367e-11
## PB.2869.1|6b0af0|path2:1-1117(+)|transcript/22387   3.868859e-10
## PB.4931.3|b8571c|path20:38-3608(+)|transcript/1966  8.540648e-10
## PB.9559.1|transcript/7471:1-2422(+)|transcript/7471 8.540648e-10
## PB.876.2|206df6|path10:20-1995(+)|transcript/12799  2.235922e-09
## PB.3850.1|906dad|path20:1-2023(+)|transcript/12300  2.047865e-08
## PB.879.1|206df6|path7:41-2171(+)|transcript/12507   4.286006e-08
## PB.499.11|124cf9|path4:2108-4109(+)|transcript/7418 4.977011e-08
## PB.4804.1|b3ae52|path0:150-1779(+)|transcript/17853 9.928051e-08
```

```r
tab_Season4F <- topTags(results_Season4F)
write.table(tab_Season4F, file = "Kgenelist-Top25_Season4F.txt")
```

Output summary expression data

```r
dim(results_Season4F)
```

```
## [1] 9813    3
```

```r
summary(de_Season4F <- decideTests(results_Season4F))
```

```
##         2-1
## Down    133
## NotSig 9612
## Up       68
```

```r
Summary_Season4F <- (de_Season4F <- decideTests(results_Season4F))

write.table(Summary_Season4F, file = "summary_Season4F.txt")

detags_Season4F <- rownames(dge_Season4F)[as.logical(de_Season4F)]
```

Visualise with a plotsmear

```
detags_Season4F <- rownames(dge_Season4F)[as.logical(de_Season4F)]
# export the plot as png
png("plotSmear_Season4F.png", width = 600)
plotSmear(results_Season4F, de.tags = detags_Season4F)
```

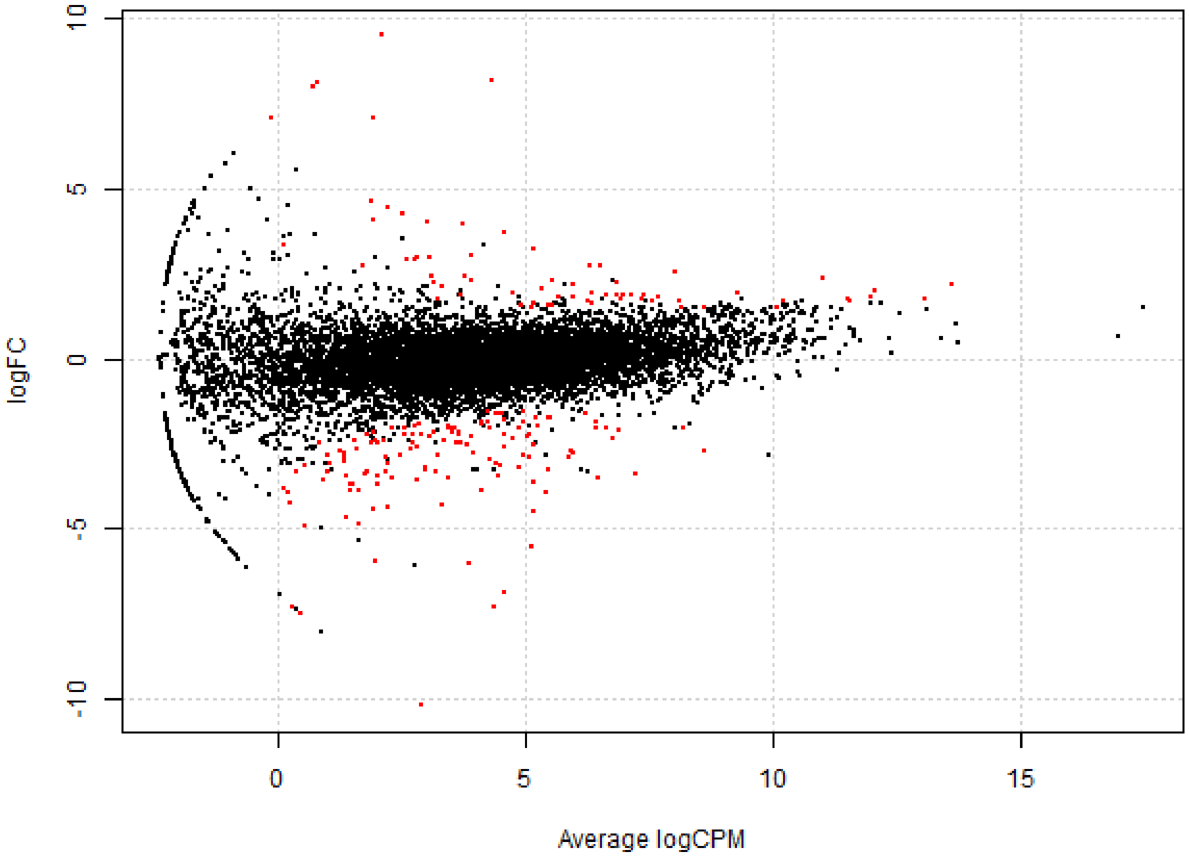


Figure 35: Log Fold Change vs average Log counts per million for expression level comparing March to September among four *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change over season, with p-values <0.05. Counts calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of four female *Tiliqua adelaidensis* individuals, two collected in autumn and two collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts (note G6 was male).

Visualise with a volcano plot

```
# export the plot as png
png("volcanoData_Season4F.png", width = 600)
volcanoData_Season4F <- cbind(results_Season4F$table$logFC,
↪  -log10(results_Season4F$table$PValue))
colnames(volcanoData_Season4F) <- c("logFC", "negLogPval")
DEGs_Season4F <- results_Season4F$table$PValue < 0.05 & abs(results_Season4F$table$logFC)
↪  >
    2
point.col_Season4F <- ifelse(DEGs_Season4F, "red", "black")
plot(volcanoData_Season4F, pch = 16, col = point.col_Season4F,
    cex = 0.5)
```

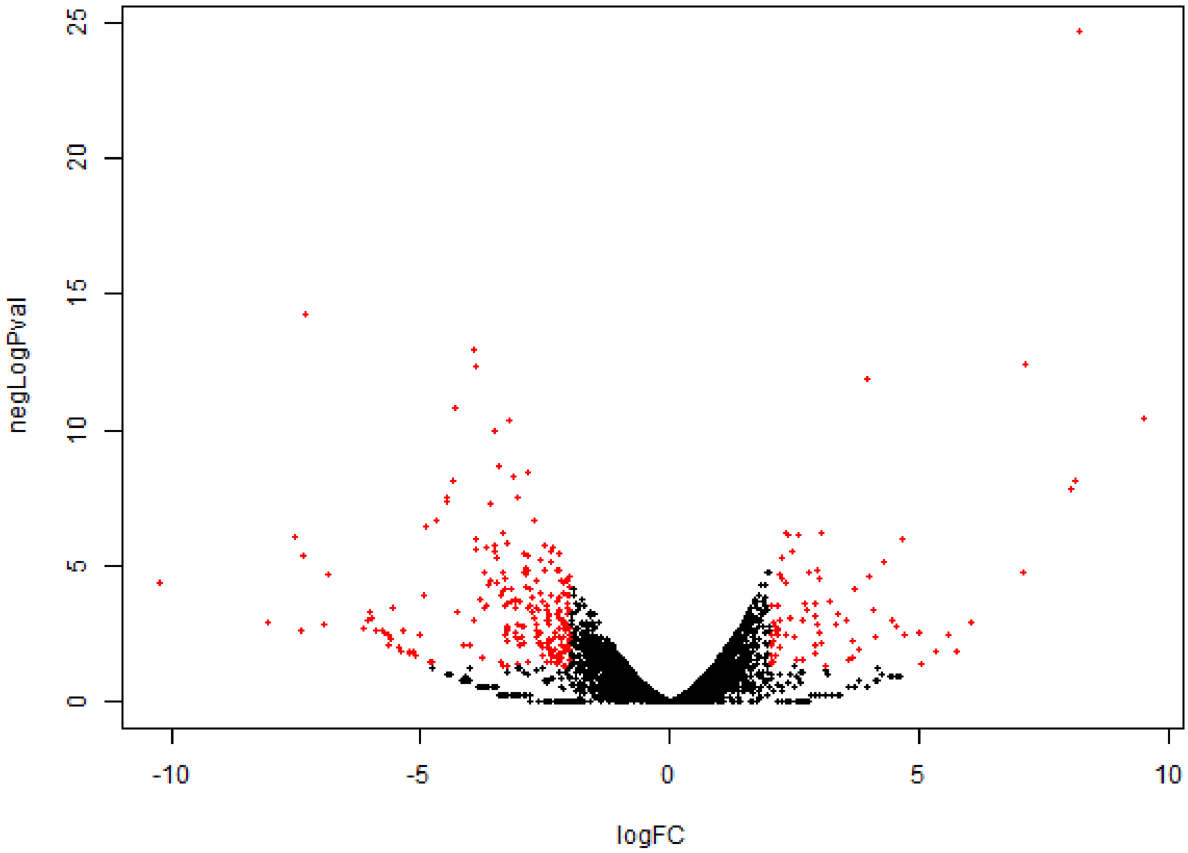


Figure 36: Negative Log of the p-value against Log Fold Change in expression level comparing March to September among four *T. adelaidensis* individuals. Red dots indicate transcripts with a significant change over season, with p-values <0.05 and *also* a log fold change >2. Counts calculated using Kallisto and EdgeR, for short-read poly(A) selected mRNA sequenced from kidney tissue of four female *Tiliqua adelaidensis* individuals, two collected in autumn and two collected in spring, sequenced using Illumina Hi-Seq PE short-read sequencing. Reference sequence isolated from *Tiliqua adelaidensis* skink ID: G6, a representative individual for Pacific Biosciences Iso-Seq long-read transcript sequencing and annotation to generate full length reference transcripts (note G6 was male).