Method S1 Bioinformatics

2011-2012 De novo Assembly Analysis Pipeline

Total RNA preparations	24 samples
Quality control	Gel Electrophoresis and Agilent 2100 Bioanalyzer
GenXpro	Sample processing, sequencing
Platform	Illumina Hiseq 2000

Hardware and operating systems:

Local hardware	Dell Precision T1500 64bit 8 GB RAM, 4x Intel Core i5 CPU 3.2GHz
Local operating system	Scientific Linux Version 6.1 (Carbon) Kernel Linux 2.6.32-220.7.1el6.x86_64 Bio-Linux6 (Ubuntu lucid 10.04) Kernel Linux 2.6.32-40 generic
RRZN hardware	Cluster System of the Computation Centre Lower Saxony/Hannover (RRZN) at the Leibniz University Hannover Computing power up to 640 GB RAM, 160x CPU (2-3GHz)
RRZN operating system	Various versions of Scientific Linux

A Quality control of GenXpro fastq data files and first assessment

Program: FastQC 0.10.1
Simon Andrews
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
uwe@knut:~ fastqc $in –-nogroup

$in	= variable for data file name
--nogroup	= All reports will show data for every base in the read

Total 100bp single end, strand specific reads in 24 datafiles: 665404362

Data volume in 24 data files: 168 GB
Two-thirds of all 100 bp reads bearing 3`end adapter derived sequences
Read quality drops rapidly after base 50-100

B Adapter trimming (clipping off 3`end adapter sequences)

Program: ea-utils-1.0.5-171-x86_64 fastq processing utilities
Erik Aronesty (2011). ea-utils: "Command-line tools for processing biological sequencing data"
http://code.google.com/p/ea-utils/
uwe@knut:~ fastq-mcf Adapter-Pix2.fa $in -l 20 -o $in-adapter.fastq -q 2

Adapter-Pix2.fa	= adapters sequences in fasta format
$in	= Illumina Hiseq data files in fastq format
-l 20	= minimum length 20 bp after adapter clipping
-q 2	= in this case no quality clipping
-o	= output
$in-adapter.fastq	= Data files in fastq format after adapter clipping

Total 20-100bp sequences in 24 data files: 654462398

C Quality trimming

Program: fastx-tools_0.0.13_binaries_Linux_2.6_amd64
http://hannonlab.cshl.edu/fastx_toolkit
uwe@knut:~ fastq_quality_trimmer -i $in-adapter.fastq -o $in-adapter-qual.fastq -t 3

-i	= input
-o	= output
-t 3	= quality threshold

total 1-100 bp sequences in 24 data files: 653844992

D **Correcting sequencing errors**

Program: coral-1.3, split und cat (Standard linux)
Salmela L, Schroeder J. 2011. Correcting errors in short reads by multiple alignments. Bioinformatics 27: 1455-1461. (Also in HiTSeq 2011).
http://www.cs.helsinki.fi/u/lmsalmel/coral/

```
uwe@knut:~  split -l  16000000  $in-adapter-qual.fastq 0 -d
-l 16000000              = number of lines in each piece
0                        = prefix
-d                       = numeric suffix default 2
uwe@knut:~  for k in {000,001,002,003...};do coral -fq $k -o $k-coral.fastq -p 2  -illumina
                  \ -k 21  -e 0.11 -t 0.75 -a 1000;done
for k ...; do...;done    = loop operation
-fq                      = input is fastq format
-o                       = output
-p 2                     = cpu number for computation
-illumina                = quick option targets to -mr 1 -mm 1 -g 1000
-k 21                    = k-mer length for indexing
-e 0.11                  = maximum error rate in multiple alignments
-t 0.75                  = minimum proportional read consensus
-a 1000                  = maximum size multiple alignments
uwe@knut:~ cat 0*-coral.fastq > $in-coral.fastq #merge corrected reads
```

E **Collapsing reads with identical lenght and 100% identical basepairs**
Reduction datasize and change to fasta format

Program: fastx-tools_0.0.13_binaries_Linux_2.6_amd64

```
uwe@knut:~ fastx_collapser -i $in-coral.fastq -o $in-collapse.fasta
```

F **Collapsing reads with different length and 100% identical base pairs**

Program: USEARCH 5.0.144
Robert Edgar
http://www.drive5.com/usearch/

```
uwe@knut:~     usearch --sort $in-collapse.fasta --output $in-collapse-sort.fasta -minlen 20
--sort                   = read length sort
-minlen 20               = minimum read length 20 bp
uwe@knut:~     usearch -cluster -derep_subseq  $in-collapse-sort.fasta -seedsout $in-
                  \ derep.fasta -w 16 -slots 40000003 -sizeout -minlen 20
-cluster                 = cluster identical reads
-derep_subseq            = independent length
-seedsout                = save consensus sequence
-w 16                    = alignment word size
-slots 40000003          = value for indexing (depends on computation power)
-sizeout                 = shows supporting read count in consensus fasta header
-minlen 20               = minimum read length 20 bp
```

G **Merge all filtered sequences in one data file and rename**

Program: fastx-tools_0.0.13_binaries_Linux_2.6_amd64

```
uwe@knut:~ cat *derep.fasta > ffr.fasta (ffr = final filtered reads)
uwe@knut:~ fastx_renamer -i ffr.fasta -o ffr-re.fasta -n COUNT
-n COUNT                 = delete fasta header and replace with increasing numeric values
```

total 20-100bp sequences in 1 datafiles: 75411191
Median read length 92 bp

H **Denovo assembly (RRZN Cluster system)**
Strand specific multi k-mer assembly

Program: velvet version 1.1.05
Zerbino Dr, Birney E. 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. Genome Research 18: 821-829.
http://www.ebi.ac.uk/~zerbino/velvet/
k-mer size 95-23 in 19 steps with two different

-cov_cutoff (auto oder 2) settings
= 38 different k-mer assemblies

bash script job parameter

```
#!/bin/bash
#PBS -N 20120104_1_velvet
#PBS -M uwe.jonas@obst.uni-hannover.de
#PBS -m ae
#PBS -j oe
#PBS -l nodes=1:ppn=8
#PBS -l walltime=100:00:00
#PBS -l mem=150gb
#PBS -q all
# show which computer the job ran on
echo "Job ran on:" $(hostname)
# initialise the modules environment
source $MODULESHOME/init/bash
# load the relevant modules
module load velvet
# change to work dir:
cd $PBS_O_WORKDIR
# the program to run
export OMP_NUM_THREADS=10
#Platzhalter
#$a = numeric k-mer value
#(95,91,87,83,79,75,71,67,63,59,55,51,47,43,39,35,31,27,23)
#$b = auto oder 2
velveth k_$a $a -short -fasta ffr-re.fasta -strand_specific
velvetg k_$a  -cov_cutoff $b  -min_contig_lgth 200
cat k_$a/contigs.fa > allcontigs.fa
```

I Removal of polyA stretches, low complexity sequences and adapter remnants

Program: seqclean_x86_64
http://compbio.dfci.harvard.edu/tgi/software/
uwe@knut:~ seqclean allcontigs.fa -v Adapter-Pix2.fa

Assembled contigs from Velvet: 789910

Add	22	puplished Prunus avium sequences from Alkio M, Jonas U, Sprink T, Van Nocker S, Knoche M. 2012 Identification of putative candidate genes involved in cuticle formation in Prunus avium (sweet cherry) fruit Annals of Botany 110: 101-112
	3	unpublished Prunus avium sequences from Version 1.3 and
	1	Enterobacteria phage phiX174 complete genome

Programm: renameESTs.pl (perl script aus PAVE 3.0)
uwe@knut:~ renameESTs.pl allcontigs.clean.fasta raw6.fasta
raw6_000001 - raw6_789936
Number of Contigs for PAVE super assembly: 789936

Digested statistics
Program:fasta_summary.pl (perl script Konrad Paszkiewicz)
uwe@knut:~ fasta_summary.pl -i raw6.fasta -o raw6.stat

Statistics for read lengths:

Min read length:	107
Max read length:	7609
Mean read length:	480.94
Standard deviation of read length:	376.04
Median read length:	351
N50 read length:	549

Statistics for numbers of reads:

Number of reads: 789936
Number of reads >=1kb: 61920
Number of reads in N50: 197544

Statistics for bases in the reads:
Number of bases in all reads: 379908465
Number of bases in reads >=1kb: 92311426
GC Content of reads: 43.26 %

Simple dinucleotide repeats:
Number of reads with over 70% dinucleotode repeats: 0.00 % (0 reads)
AT: 0.00 % (0 reads)
CG: 0.00 % (0 reads)
AC: 0.00 % (0 reads)
TG: 0.00 % (0 reads)
AG: 0.00 % (0 reads)
TC: 0.00 % (0 reads)

Simple mononucleotide repeats:
Number of reads with over 50% mononucleotode repeats: 0.00 % (0 reads)
AA: 0.00 % (0 reads)
TT: 0.00 % (0 reads)
CC: 0.00 % (0 reads)
GG: 0.00 % (0 reads)

J Superassembly of assembled Velvet contigs (raw6_000001 - raw6_789936)
Programme: PAVE 3.0, CAP3 and working MySQL-Server

uwe@knut:~ loadLibrary.pl Pa_assembly
Configuration file LIB.cfg

```
PAVE_db          = PAVE_Pa_assembly     # Database name
PAVE_host        = IP                   # Database host IP
PAVE_user        =                      # MySQL read/write user
PAVE_password    =                      # MySQL password

libid            = raw6
seqfile          = raw6.fasta
title            = raw6 contigs Library
organism         = Prunus avium
cultivar         = Regina
tissue           = Ovary Exocarp Mesocarp
stage            = 03 - 94 DAFB weekly sampling
source           = RNA-HiSeq_2010 Velvet raw contigs
default_qual     = 21
```

uwe@knut:~ runPAVE.pl Pa_assembly
Configuration file PAVE.cfg

```
AssemblyID     = Pa     # Assembly name
# Database parameters (same as in LIB.cfg)
PAVE_db          = PAVE_Pa_assembly  # Database name
PAVE_host        = IP                # Database host IP
PAVE_user        =                   # MySQL read/write user
PAVE_password    =                   # MySQL password

CPUs             = 4                 # number of processors to use for assembly
SKIP_ASSEMBLY    = 0
USE_TRANS_NAME   = 0
CLIQUE           = 400 99 20      #400 overlap 99 identity% 20 unfit overlap
TC1              = 300 99 20       #decreasing stringent conditions
TC2              = 200 98 20
```

TC3 = 150 98 20
TC4 = 100 97 40

EXTRA_CONFIRM = 10
SNP_CONFIRM = 10
INDEL_CONFIRM = 5

Number of contigs from PAVE Assembly: 68101

Digested statistics
Program:fasta_summary.pl (perl script Konrad Paszkiewicz)
uwe@knut:~ fasta_summary.pl -i Pa_ctg_sng.fasta -o Pa_ctg_sng.stat

Statistics for contig lengths:

Min contig length:	107
Max contig length:	14149
Mean contig length:	660.80
Standard deviation of contig length:	700.45
Median contig length:	370
N50 contig length:	1070

Statistics for numbers of contigs:

Number of contigs:	68101
Number of contigs >=1kb:	12867
Number of contigs in N50:	11874

Statistics for bases in the contigs:

Number of bases in all contigs:	45000984
Number of bases in contigs >=1kb:	23528097
GC Content of contigs:	42.31 %

Simple dinucleotide repeats:
Number of contigs with over 70% dinucleotode repeats: 0.00 % (0 contigs)
AT: 0.00 % (0 contigs)
CG: 0.00 % (0 contigs)
AC: 0.00 % (0 contigs)
TG: 0.00 % (0 contigs)
AG: 0.00 % (0 contigs)
TC: 0.00 % (0 contigs)

Simple mononucleotide repeats:
Number of contigs with over 50% mononucleotode repeats: 0.00 % (0 contigs)
AA: 0.00 % (0 contigs)
TT: 0.00 % (0 contigs)
CC: 0.00 % (0 contigs)
GG: 0.00 % (0 contigs)

K Read re-mapping to contigs (68101), fpkm calculation

Programs: bowtie, tablet, LibreOffice
bowtie 0.12.7 http://bowtie-bio.sourceforge.net/index.shtml
tablet 1.12.03.26 http://bioinf.scri.ac.uk/tablet
libreoffice 3.5 calc http://de.libreoffice.org
uwe@knut:~ bowtie-build Pa_ctg_sng.fasta
build reference index
uwe@knut:~ bowtie -v 1 -M 2 --trim5 5 --trim3 30 -best -strata Pa_ctg_sng.fasta -q $in -S $in.sam

$in	fastq samples 03G-80M unfiltered, untrimmed
-v 1	allowed mismatch 1
-M 2	random reports of multiple matching reads
--trim5 5	trim 5 bases at 5`end

--trim3 30	trim 30 bases at 3`end
-best	hits guaranteed best stratum
-strata	hits in sub-optimal strata are not reported
$in.sam	reported hits in SAM format

Opening $in.sam in tablet after visual inspection, copy and paste read count/contig to LlibreOffice calc. Calculate total read count and fpkm values.

Filter settings for high and low abundance contigs

Group 1 high abundance contigs	≥ 200 bp length
	≥ 30 total read count per contig in each sample
	≥ 75 total read count per contig in all 24 samples
Group 2 low abundance contigs	< 200 bp length
	< 30 total read count per contig in each sample
	< 75 total read count per contig in all 24 samples