Comparisons between Different Methods of SNP Calling from Ryegrass Genotyping-by-Sequencing (GBS) Data

Executive Summary

This document serves as the supplementary file to the publication titled "*snpGBS: A Simple and Flexible Bioinformatics Workflow to Identify SNPs from Genotyping-by-Sequencing Data*". Both detailed analytical results and the codes used to generate these outputs are included in this document, with a focus on comparing different methods of SNP calling from ryegrass GBS data.

Data Description

Ninety-six samples were selected from a perennial ryegrass training population as described in [3], for which GBS had been performed using previously established methods [2]. Briefly, DNA was isolated from leaf tissue samples, then digested using the *Ape*KI restriction enzyme (NEB). Each GBS sample was ligated to a unique barcode identifier and a common adapter before merging into a 96-plex library. GBS libraries were each sequenced on two lanes of an Illumina HiSeq 2500 flowcell at AgResearch Invermay, New Zealand.

Bioinformatic Processing and Data Analyses

snpGBS involves demultiplexing raw GBS reads using cutadapt [8], mapping demultiplexed reads back to the same reference genome as described in [3] with bowtie2 [5], and finally, SNP calling using bcftools [6] with default options. For comparison, SNPs had been also identified using UNEAK [7] and TASSEL5 [4]. Genetic analyses of different SNP datasets were carried out using KGD [1].

Summary of KGD Outputs

Item	snpGBS	snpGBS-filtered	TASSEL5	UNEAK
Number of Samples (Pre-KGD filtering)	96	96	96	96
Number of Samples (Post-KGD filtering)	95*	95*	95*	95*

Number of SNPs (Pre-KGD filtering)	1,915,974	837,102	254,079	267,720
Number of SNPs (Post-KGD filtering)**	1,305,406^	830,207	254,079^^^	267,720^^^^
Mean Co-Call Rate	0.371	0. 4618938	0.5591871	0.2531523
Min. Co-Call Rate	0.152	0. 1946683	0.2213551	0.07533991
Proportion of Missing Genotypes	0.481	0. 386157	0.2819255	0.5620006
Call Rates	0.519	0. 613843	0.7180745	0.4379994
Mean Sample Depth	1.922	2. 347947	2.566835	1.019548
Mean Self-Relatedness (G5 Diagonal)	1.018	1.005822	1.001888	1.042407
Number of SNP per GBS Fragment	4.556719	3.753765	2.371545	NA

Note

* 1 sample with maximum depth of 1 and/or mean depth < 0.3 removed

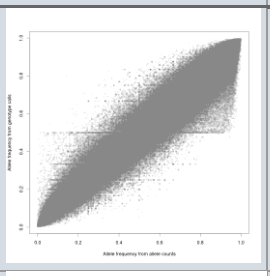
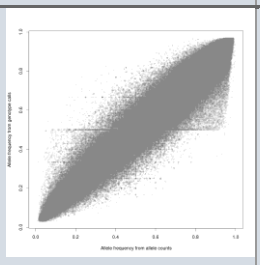
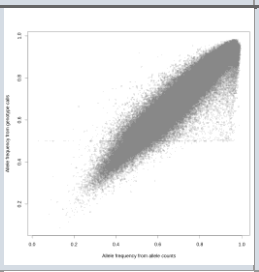
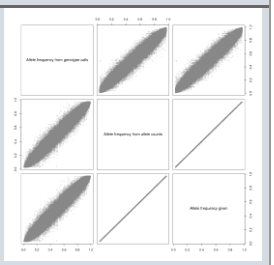
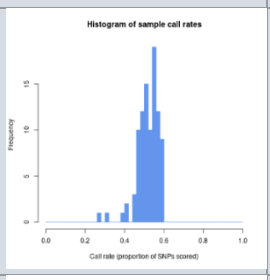
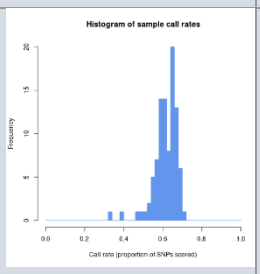
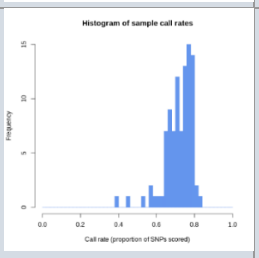
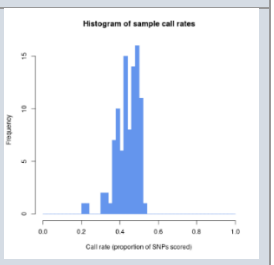
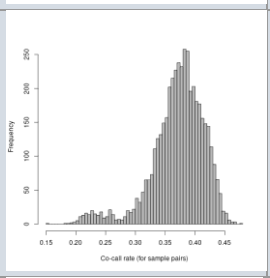
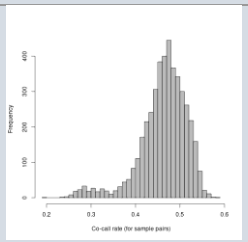
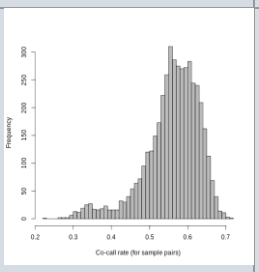
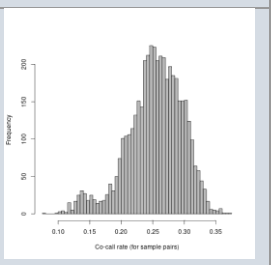
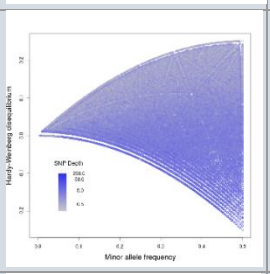
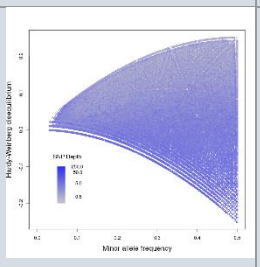
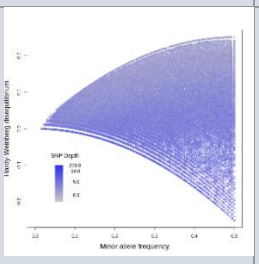
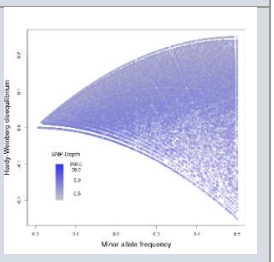
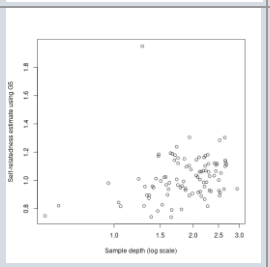
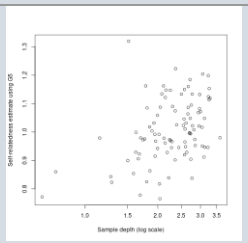
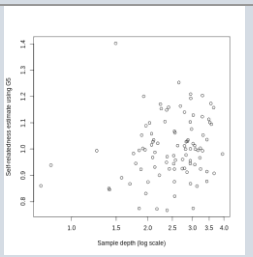
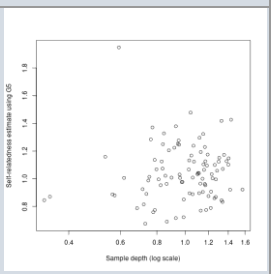
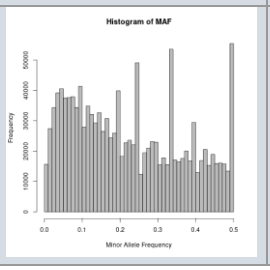
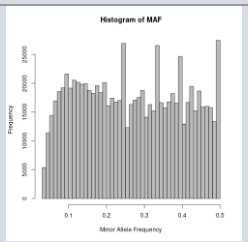
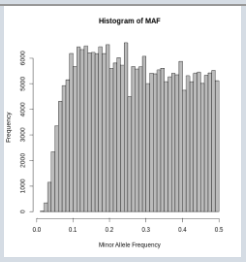
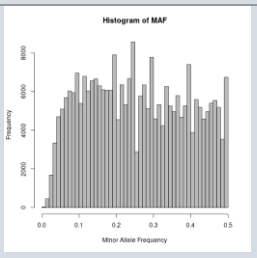
^ 610568 SNPs with MAF=0 or depth < 0.01 removed

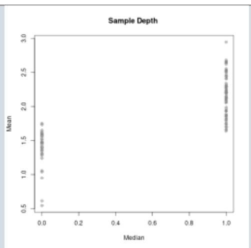
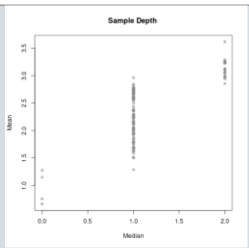
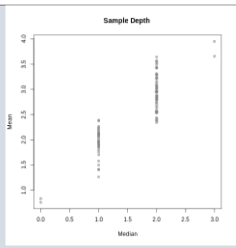
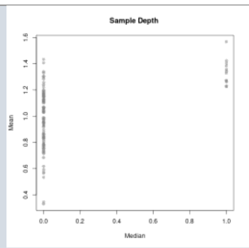
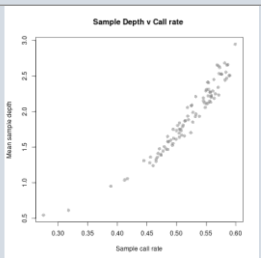
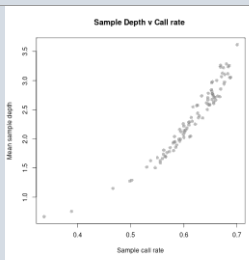
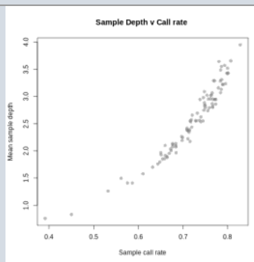
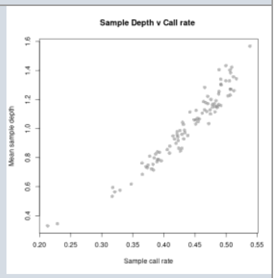
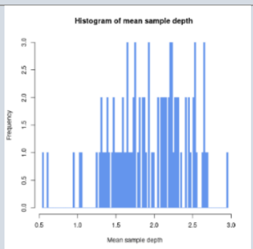
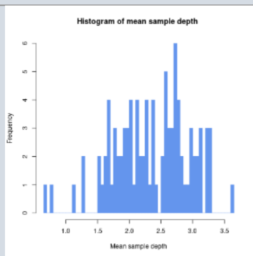
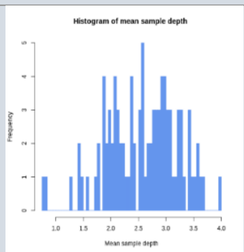
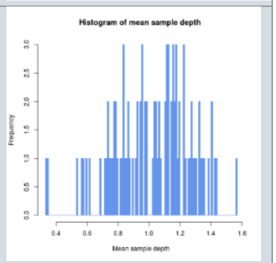
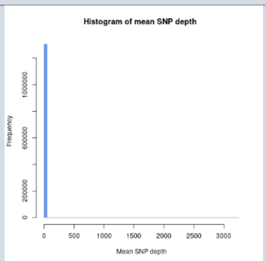
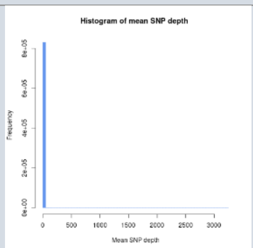
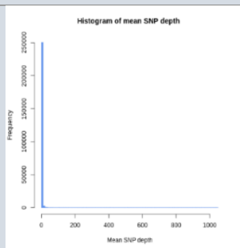
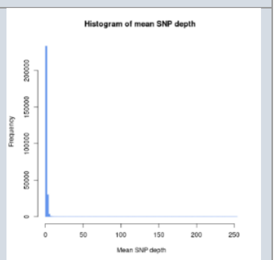
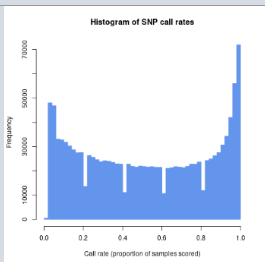
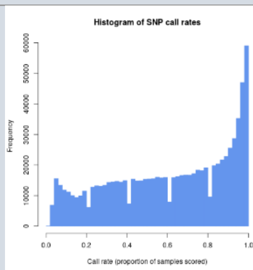
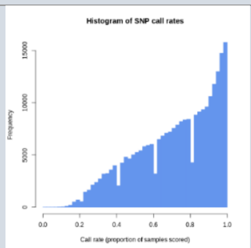
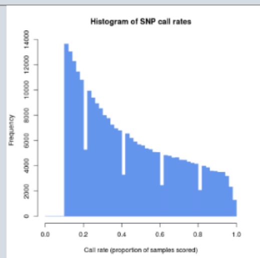
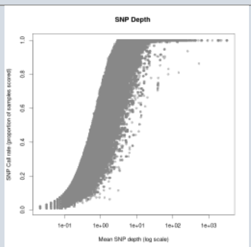
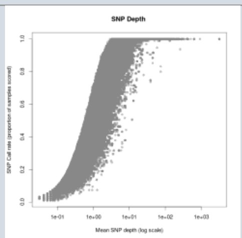
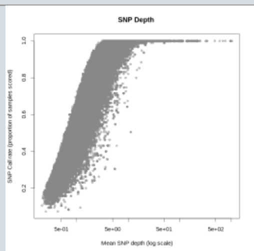
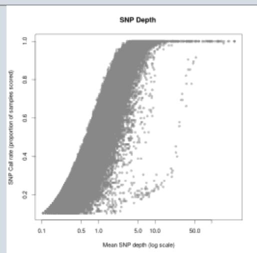
^^ 6895 SNPs with MAF=0 or depth < 0.01 removed

^^^ 0 SNPs with MAF=0 or depth < 0.01 removed

^^^^ 0 SNPs with MAF=0 or depth < 0.01 removed

Graphic Outputs of KGD

Item	snpGBS	snpGBS-filtered	TASSEL5	UNEAK
Allele Frequency				
Call Rates				
Co-call Rates				
Fin Plot				
Diagonal of G5 vs Depth				
Histogram of MAF				

Sample Depth				
Sample Depth vs Call Rate				
Histogram of Sample Depth				
Histogram of SNP Depth				
Histogram of SNP Call Rates				
SNP Call Rates vs Depth				

Reference

1. Dodds, Ken G., et al. "Construction of relatedness matrices using genotyping-by-sequencing data." BMC Genomics 16.1 (2015): 1-15.

2. Elshire, Robert J., et al. "A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species." PloS one 6.5 (2011): e19379.

3. Faville, Marty J., et al. "Predictive ability of genomic selection models in a multi-population perennial ryegrass training set using genotyping-by-sequencing." Theoretical and Applied Genetics 131.3 (2018): 703-720.

4. Glaubitz, Jeffrey C., et al. "TASSEL-GBS: a high capacity genotyping by sequencing analysis pipeline." PloS ONE 9.2 (2014): e90346.

5. Langmead, Ben and Salzberg, Steven L."Fast gapped-read alignment with Bowtie 2." Nature Methods 9.4 (2012): 357.

6. Li, Heng. "A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data." Bioinformatics 27.21 (2011): 2987-2993.

7. Lu, Fei, et al. "Switchgrass genomic diversity, ploidy, and evolution: novel insights from a network-based SNP discovery protocol." PLoS Genet 9.1 (2013): e1003215.

8. Martin, Marcel. "Cutadapt removes adapter sequences from high-throughput sequencing reads." EMBnet. journal 17.1 (2011): 10-12.

Scripts

snpGBS

```
# demultiplexing

cutadapt -j 32 -e 0 --no-indels -g file:barcodes.fasta -o
"demultiplexed_{name}.fastq.gz" ABC12AAXX_1_fastq.txt.gz
>01.demultiplexed.stdout 2>01.demultiplexed.stderr

# mapping

for i in ./demultiplexed_Ind*.fastq.gz

     do     echo $i

     bowtie2 --very-fast-local -x ryegrass -U $i -S ./${i##*/}.sam
2>./${i##*/}.bowtie2.stdout

done

# filtering

for i in *.sam
```

```
   do     echo $i
   samtools view -q 20 -bS $i > "${i%.sam}.bam"
done
# sorting
for i in *.bam
   do     echo $i
   samtools sort $i -o "${i%.bam}.sorted.bam"
done
# generating list of bam files
for i in *.sorted.bam
   do     echo $i
done > bamlist
# SNP calling
bcftools mpileup -I -Ou -f ryegrass_chr1-8.fa -b bamlist -a AD -d
1000000 | bcftools call -cv - | bcftools view -M2 - >
snpGBS_ryegrass_apeki.vcf
# KGD
python vcf2ra.py snpGBS.vcf &>vcf2ra.stdout
Rscript run_KGD_snpGBS.R &> run_KGD_snpGBS.stdout
```

snpGBS-KGD (in R)

```
gform <- "Tassel"   ####Used for HapMap files
genofile <- " snpGBS_ryegrass_apeki.vcf.ra.tab"   ###Add location to
Ra or HapMap file
sampdepth.thresh <- 0.3
source("/home/kangj/git/KGD/GBS-Chip-Gmatrix.R")
Gfull <- calcG()
GHWdgm.05 <- calcG(which(HWdis > -0.05),"HWdgm.05", npc=4)
###To save a G Matrix
writeG(GHWdgm.05, "GHWdgm.05_snpGBS_goat", outtype=c(1,2,3,4,6))
save.image(file = "KGD_snpGBS.image")
#To write out vcf file
# writeVCF(outname="GHWdgm.05_run1")
```

snpGBS-filtered (in bash)

```
# define input and output

VCF_IN=snpGBS_ryegrass_apeki.vcf

VCF_OUT=snpGBS_ryegrass_apeki_filtered.vcf

# define filtering thresholds

MISS=0.9 # MIN_DEPTH=5

QUAL=30 # MAX_DEPTH=50

MAF=0.03

# filtering using vcftools

vcftools --vcf $VCF_IN --max-missing $MISS --maf $MAF --minQ $QUAL --recode --stdout >$VCF_OUT
```

TASSEL5

```
# 01.GBSSeqToTagDBPlugin

run_pipeline.pl -Xms512m -Xmx300g -fork1 -GBSSeqToTagDBPlugin -e ApeKI -i fastq/ -db output/GBSV2.db -k key/key.txt -kmerLength 64 -minKmerL 20 -mnQS 20 -mxKmerNum 100000000 -endPlugin -runfork1

# 02.TagExportToFastqPlugin

run_pipeline.pl -Xms512m -Xmx300g -fork1 -TagExportToFastqPlugin -db output/GBSV2.db -o output/tagsForAlign.fa.gz -c 1 -endPlugin  -runfork1

# 03.Alignment.sh

# Bowtie2 create index from the reference genome

bowtie2-build referenceGenome/ryegrass_chr1-8.fa ryegrass >bowtie2-build.stdout 2>bowtie2-build.stderr

# Bowtie2 Alignment

bowtie2 -p 8 --very-sensitive -x ryegrass -U output/tagsForAlign.fa.gz -S tagsForAlignFullvs.sam  >bowtie2.stdout 2>bowtie2.stderr

# 04.SAMToGBSdbPlugin

run_pipeline.pl -Xms512m -Xmx300g  -fork1 -SAMToGBSdbPlugin -i tagsForAlignFullvs.sam -db output/GBSV2.db -minMAPQ 20 -aProp 0.0 -aLen 0  -endPlugin  -runfork1
```

```
# 05.DiscoverySNPCallerPluginV2

run_pipeline.pl -Xms512m -Xmx300g  -fork1 -
DiscoverySNPCallerPluginV2 -db output/GBSV2.db -mnLCov 0.1 -mnMAF
0.03 -deleteOldData true -endPlugin -runfork1

# 06.ProductionSNPCallerPluginV2

run_pipeline.pl -Xms512m -Xmx300g  -fork1 -
ProductionSNPCallerPluginV2 -db output/GBSV2.db -e ApeKI -i fastq/ -
k key/key.txt -kmerLength 64 -o ryegrass_apeki_tassel5_q20 -
endPlugin -runfork1
```

TASSEL-KGD (in R)

```
gform <- "Tassel"   ####Used for HapMap files

genofile <- "../ryegrass_apeki_tassel5_q20.vcf.ra.tab"   ###Add
location to Ra or HapMap file

sampdepth.thresh <- 0.3

source("/home/kangj/git/KGD/GBS-Chip-Gmatrix.R")

Gfull <- calcG()

GHWdgm.05 <- calcG(which(HWdis > -0.05),"HWdgm.05", npc=4)


###To save a G Matrix

writeG(GHWdgm.05, "GHWdgm.05_tassel5", outtype=c(1,2,3,4,6))

save.image("KGD_apeki_ryegrass_tassel5.image")

#To write out vcf file

writeVCF(outname="GHWdgm.05_tassel5")
```

UNEAK

```
# 01.Create_dirs

run_pipeline.pl -fork1 -UCreatWorkingDirPlugin -w . -endPlugin -
runfork1

# 02.FASTQtoTagCount

run_pipeline.pl -Xms512m -Xmx300g -fork1 -UFastqToTagCountPlugin -w
. -c 1 -e ApeKI -s 400000000 -endPlugin -runfork1

# 03.MergeTaxaTagCounts
```

```
run_pipeline.pl -Xms512m -Xmx300g -fork1 -UMergeTaxaTagCountPlugin -
w . -t n -m 600000000 -x 100000000 -c 3 -endPlugin -runfork1

# 04.TagCountToTagPair

run_pipeline.pl -Xms512m -Xmx300g -fork1 -UTagCountToTagPairPlugin -
w . -e 0.03 -endPlugin -runfork1

# 05.TagPairToTBT

run_pipeline.pl -Xms512m -Xmx300g -fork1 -UTagPairToTBTPlugin -w . -
endPlugin -runfork1

# 06.TBTToMapInfo

run_pipeline.pl -Xms512m -Xmx300g -fork1 -UTBTToMapInfoPlugin -w . -
endPlugin -runfork1

# 07.MapInfoHapMap

run_pipeline.pl -Xms512m -Xmx300g -fork1 -UMapInfoToHapMapPlugin -w
. -mnMAF 0.03 -mxMAF 0.5 -mnC 0.1 -mxC 1 -endPlugin -runfork1
```

UNEAK-KGD (in R)

```
gform <- "uneak"   ####Used for HapMap files

genofile <- "../hapMap/HapMap.hmc.txt"   ###Add location to Ra or
HapMap file

sampdepth.thresh <- 0.3

source("/home/kangj/git/KGD/GBS-Chip-Gmatrix.R")

Gfull <- calcG()

GHWdgm.05 <- calcG(which(HWdis > -0.05),"HWdgm.05", npc=4)

###To save a G Matrix

writeG(GHWdgm.05, "GHWdgm.05_uneak_ryergass_apeki",
outtype=c(1,2,3,4,6))

save.image(file = "KGD_uneak_ryegrass_apeki.image")

#To write out vcf file

# writeVCF(outname="GHWdgm.05_uneak")
```