Starling-May18
Projects/Katarina Stuart/KStuart.Starling-Aug18/Nc3_HihiSV/Data/2023-01-23.SVcalling

PDF Version generated by

Katarina Stuart (z5188231@ad.unsw.edu.au)

on

Aug 15, 2024 @03:06 PM NZST

Table of Contents

2023-01-23.SVcalling

Katarina Stuart (z5188231@ad.unsw.edu.au) - Mar 27, 2024, 12:11 PM GMT+13

SV Var calling: 82 myna WGR

Choosing individuals that have high enough coverage:
13-28G dup mapped data for these samples

```
cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling
ls -lh /nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/mapped_reads/*sorted.dup.bam | sed 's/G / /g' | sort -t' ' -k6,6n -k6,6V | cut -d' ' -f 10 | sed 's/.sorted.dup.bam//g' | tail -n 31 | sort | sed 's/\/\t/g' | cut -f 9 > sample_file_ids_highcov.txt
```

Smoove:

For when you want to use lumpy on a large cohort use smoove: https://github.com/brentp/smoove

Step1: SV calling done separately for each sample

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_smoove_step1.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-12:00:00
#SBATCH --mem=5GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --profile task
#SBATCH --array=1-31
#SBATCH --partition=milan

# load modules
module purge
module load smoove/0.2.8-Miniconda3

SAMPLE=$(sed
"${SLURM_ARRAY_TASK_ID}q;d" /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt)
echo "working with sample:" ${SAMPLE}

# set paths
DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/mapped_reads/
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove
```

```
#Run smoove step 1
smoove call --outdir ${OUT_DIR} --name ${SAMPLE} --fasta ${GENOME} -p 1 --genotype ${BAM}
```

Step2: Get the union of sites across all samples

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_smoove_step2.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-48:00:00
#SBATCH --mem=5GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --profile task
#SBATCH --partition=milan

# load modules
module purge
module load smoove/0.2.8-Miniconda3

# set paths
GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove

#Run smoove step 2
smoove merge --name merged -f ${GENOME} --outdir ${OUT_DIR} ${OUT_DIR}/*.genotyped.vcf.gz
# this will create ./merged.sites.vcf.gz
```

Step3: genotype each sample at those sites

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_smoove_step3.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-12:00:00
#SBATCH --mem=5GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --profile task
#SBATCH --array=1-31
#SBATCH --partition=milan

# load modules
module purge
module load smoove/0.2.8-Miniconda3
module load duphold/0.2.3

SAMPLE=$(sed
```

```
"${SLURM_ARRAY_TASK_ID}q;d" /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt)
echo "working with sample:" ${SAMPLE}

# set paths
DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/mapped_reads/
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove

#Run smoove step 3
smoove genotype -d -x -p 4 --name ${SAMPLE}_joint --outdir ${OUT_DIR} --fasta ${GENOME} --vcf ${OUT_DIR}/merged.sites.vcf.gz ${BAM}
```

Step4: paste all the single sample VCFs with the same number of variants to get a single, squared, joint-called file

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_smoove_step4.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-12:00:00
#SBATCH --mem=5GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --profile task
#SBATCH --partition=milan

# load modules
module purge
module load smoove/0.2.8-Miniconda3

# set paths
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove

cd ${OUT_DIR}

#Run smoove step 4
smoove paste --name smoove_hihi ${OUT_DIR}/*_joint-smoove.genotyped.vcf.gz
```

Step5: annotate the variants with exons, UTRs that overlap from a GFF and annotate high-quality heterozygotes:

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_31.SVCalling_hihiwgs_smoove_step5.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-12:00:00
#SBATCH --mem=5GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
```

```
#SBATCH --profile task
#SBATCH --partition=milan

# load modules
module purge
module load smoove/0.2.8-Miniconda3

# set paths
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove

cd ${OUT_DIR}

module load AGAT/1.0.0-gimkl-2022a-Perl-5.34.1-R-4.2.1
GFF=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/annotation/Ncf_H98617_scaffolded_liftoff.gff
agat_convert_sp_gxf2gxf.pl --gff $GFF -o tsebra_fix.gff

#my annotation file didn't have a 'Name=" column and thus kept giving me an error message
#ERROR: no records found with 'gene' type in gff
#https://github.com/brentp/smoove/issues/184
#Manually add this field to the end of each gene line

awk -v search="gene" 'BEGIN { OFS="\t" } {
    if ($3 == search) {
        $9 = $9";Name=dummy"
    }
    print
}' tsebra_fix.gff > tsebra_fix2.gff

#Run smoove step 5
smoove annotate --gff tsebra_fix2.gff smoove_hihi.smoove.square.vcf.gz | bgzip -c > smoove_hihi.smoove.square.anno.vcf.gz
```

Step6: Filtering

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_31.SVCalling_starlingwgs_smoove_step6.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-12:00:00
#SBATCH --mem=1GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --profile task
#SBATCH --partition=milan

# load modules
module purge
module load BCFtools/1.13-GCC-9.2.0
module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1

# set paths
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove
GVCF=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove/smoove_hihi.smoove.square.anno.vcf.gz

cd ${OUT_DIR}
```

```
#filter out z and w chrom contigs
SEX=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/annotation/Ncf_sex_linked_contigs.txt

LIST=

for CHROM in $(cut -f 1 $SEX)
do
LIST="${LIST} "--not-chr" $CHROM"
done

echo $LIST

vcftools --gzvcf $GVCF $LIST --keep-INFO-all --recode --out smoove_autosomes

#Filter to just the superscaffolds
cat <(zgrep "^#" smoove_autosomes.recode.vcf) <(zgrep -v "^#" smoove_autosomes.recode.vcf | grep -v "Ncf_contig" ) > smoove_chroms.vcf

#filtering variant quality
bcftools view -i '(SVTYPE = "DEL" & FMT/DHFFC[0-30] < 0.7) | (SVTYPE = "DUP" & FMT/DHBFC[0-30] > 1.3) | (SVTYPE = "INV")' -O v -o
smoove_hihi_SVfiltered.vcf smoove_chroms.vcf

#filter individual genotype quality
bcftools view -i '(MSHQ>=3)' -O v -o smoove_hihi_genofiltered.vcf smoove_hihi_SVfiltered.vcf
```

Delly:

Step1: SV calling done separately for each sample

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_delly_step1.sl
#SBATCH --account=00338
#SBATCH --time=00-24:00:00
#SBATCH --mem=10GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --profile task
#SBATCH --array=1-31

# load modules
module purge
module load Delly/1.1.3

SAMPLE=$(sed "${SLURM_ARRAY_TASK_ID}q;d"
 /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt)
echo "working with sample:" ${SAMPLE}

# set paths
```

```
DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/mapped_reads/
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly

#Run delly step 1
delly call -o ${OUT_DIR}/${SAMPLE}.bcf -g ${GENOME} ${BAM}
```

Step2: Merge SV sites into a unified site list

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_delly_step2.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-24:00:00
#SBATCH --mem=10GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --profile task

# load modules
module purge
module load Delly/1.1.3

#create sample BCF file input list
BCF_LIST=""

for SAMPLE in $(cat /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt )
do
BCF_LIST="${BCF_LIST} /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly/${SAMPLE}.bcf"
done

echo $BCF_LIST

# set paths
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly

#Run delly step2
delly merge -o ${OUT_DIR}/merged_sites.bcf ${BCF_LIST}
```

Step3: Genotype this merged SV site list across all samples

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_delly_step3.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-24:00:00
#SBATCH --mem=20GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
```

```
#SBATCH --cpus-per-task=2
#SBATCH --profile task
#SBATCH --array=1-31

# load modules
module purge
module load Delly/1.1.3

SAMPLE=$(sed
"${SLURM_ARRAY_TASK_ID}q;d" /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt)
echo "working with sample:" ${SAMPLE}

# set paths
GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly
DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/mapped_reads/
BAM=${DIR}/${SAMPLE}.sorted.dup.bam

#Run delly step3
delly call -g ${GENOME} -v ${OUT_DIR}/merged_sites.bcf -o ${OUT_DIR}/${SAMPLE}.rep.geno.bcf ${BAM}
```

Step4: Merge all genotyped samples to get a single VCF/BCF using BCFtools merge

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_delly_step4.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-04:00:00
#SBATCH --mem=10GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --profile task

# load modules
module purge
module load BCFtools/1.13-GCC-9.2.0

#create sample BCF file input list
BCF_LIST=""

for SAMPLE in $(cat /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt )
do
BCF_LIST="${BCF_LIST} /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly/${SAMPLE}.rep.geno.bcf"
done

echo $BCF_LIST

# set paths
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly

#Run delly step4
bcftools merge -m id -O b -o ${OUT_DIR}/merged_rep_geno.bcf ${BCF_LIST}
```

Step5: Convert BCF to VCF

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihiwgs_delly_step5.sl
#SBATCH --account=uoa00338
#SBATCH --time=00-04:00:00
#SBATCH --mem=10GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --profile task

# load modules
module purge
module load BCFtools/1.13-GCC-9.2.0

# set paths
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly

#Run delly step5
bcftools view ${OUT_DIR}/merged_rep_geno.bcf -o ${OUT_DIR}/merged_rep_geno.vcf
```

Step 6: filtering

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_31.SVCalling_hihiwgs_delly_step6.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-12:00:00
#SBATCH --mem=1GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --profile task

# load modules
module purge
module load BCFtools/1.13-GCC-9.2.0
module load PLINK/1.09b6.16
module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1

cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly

VCF=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly/merged_rep_geno.vcf

bcftools view -f PASS ${VCF} > merged_delly_pass.vcf

#generate file with info (like length and type)
grep -v "^#" merged_delly_pass.vcf | cut -f1-8 | sed 's/;/\t/g' | cut -f1,2,3,5,8,11 | sed 's/END=//g' | awk '{print $0,"\t",$6-$2}'
> merged_delly_pass_info.txt
```

```
#count number of (genotype) PASS per SNP
grep -v "^#" merged_delly_pass.vcf | grep -o -n 'PASS'  | cut -d : -f 1 | uniq -c > merged_delly_pass_infopass.txt

#final info file has the final column with a count of how many snps had a pass
paste merged_delly_pass_info.txt <(sed 's/[[:space:]]\+/\t/g' merged_delly_pass_infopass.txt | cut -f 2 ) > merged_delly_pass_infoall.txt

#generate list of SNPs to keep (75% + PASS for samples)
awk '$8>24' merged_delly_pass_infoall.txt | cut -f3 > merged_delly_pass_VARs.snpID

vcftools --vcf merged_delly_pass.vcf --snps merged_delly_pass_VARs.snpID --keep-INFO-all --recode --out merged_delly_SNPfilt

#filter out z and w chrom contigs
SEX=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/annotation/Ncf_sex_linked_contigs.txt

LIST=

for CHROM in $(cut -f 1 $SEX)
do
LIST="${LIST} "--not-chr" $CHROM"
done

echo $LIST

vcftools --vcf merged_delly_SNPfilt.recode.vcf $LIST --keep-INFO-all --recode --out merged_delly_autosomes

#Filter to just the superscaffolds
cat <(zgrep "^#" merged_delly_autosomes.recode.vcf) <(zgrep -v "^#" merged_delly_autosomes.recode.vcf | grep -v "Ncf_contig" ) >
merged_delly_chroms.vcf
```

Manta:

Manta SV calling requires just a single line (and a bit of patience)

```
#!/bin/bash -e

#SBATCH --job-name=2024_01_26.SVCalling_hihi_manta.sl
#SBATCH --account=uoa02613
#SBATCH --time=00-150:00:00
#SBATCH --mem=150GB
#SBATCH --output=%x_%j.errout
#SBATCH --mail-user=katarina.stuart@auckland.ac.nz
#SBATCH --mail-type=ALL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --profile task

# load modules
module purge
module load manta/1.6.0-gimkl-2020a-Python-2.7.18
```

```
# set paths
GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
OUT_DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_manta

# create list of input BAM files for Manta
FILE_LIST=""
for SAMPLE_NUMBER in {1..31}
do
SAMPLE=$(sed "${SAMPLE_NUMBER}q;d"
/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt )
DIR=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/mapped_reads
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
FILE_LIST="${FILE_LIST} "--bam" ${BAM} "
done
echo ${FILE_LIST}

# This created a runWorkflow.py file for the job
configManta.py ${FILE_LIST} --referenceFasta ${GENOME} --runDir ${OUT_DIR}

# run manta
/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_manta/runWorkflow.py
```

Fixing inversions

```
cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_manta/results/variants

GENOME=/nesi/nobackup/uoa00338/kstuart_projects/Nc2_HihiWGS/data/resources/Ncf_H98617_scaffolded_genome.fa
SAMTOOLS=/opt/nesi/CS400_centos7_bdw/SAMtools/0.1.19-gimkl-2017a/bin/samtools

cp diploidSV.vcf.gz diploidSV2.vcf.gz
gunzip diploidSV2.vcf.gz

module purge
module load Python/2.7.18-gimkl-2020a

python convertInversion.py ${SAMTOOLS} ${GENOME} diploidSV.vcf > diploidSV_inversions.vcf
```

Filtering

```
# load modules
module purge
module load BCFtools/1.13-GCC-9.2.0
module load PLINK/1.09b6.16
module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1

cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_manta/results/variants

VCF=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_manta/results/variants/diploidSV_inversions.vcf

bcftools view -f PASS ${VCF} > diploidSV_pass.vcf

#filter out z and w chrom contigs
SEX=/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/annotation/Ncf_sex_linked_contigs.txt

LIST=

for CHROM in $(cut -f 1 $SEX)
```

```
do
LIST="${LIST} "--not-chr" $CHROM"
done

echo $LIST

vcftools --vcf diploidSV_pass.vcf $LIST --keep-INFO-all --recode --out manta_autosomes

#Filter to just the superscaffolds
cat <(zgrep "^#" manta_autosomes.recode.vcf) <(zgrep -v "^#" manta_autosomes.recode.vcf | grep -v "Ncf_contig" ) > manta_chroms.vcf
```

SURVIVOR

Installing

```
cd /nesi/nobackup/uoa02613/kstuart_projects/programs
git clone https://github.com/fritzsedlazeck/SURVIVOR.git
cd SURVIVOR/Debug
make
```

```
cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_survivor
ln -s /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_smoove/smoove_hihi_genofiltered.vcf
ln -s /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_delly/merged_delly_chroms.vcf
ln -s /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_manta/results/variants/manta_chroms.vcf
```

Splitting up the currnet SVCF files so we have 1 file per individual PER SVcaller, so I can work/merge with them individually.

```
module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1

mkdir split_vcfs

for SAMPLE in $(cat /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt);
do
echo ${SAMPLE}
mkdir split_vcfs/${SAMPLE}
vcftools --vcf smoove_hihi_genofiltered.vcf --indv $SAMPLE --recode --recode-INFO-all --out
split_vcfs/${SAMPLE}/smoove_autosomes.${SAMPLE}
vcftools --vcf merged_delly_chroms.vcf --indv $SAMPLE --recode --recode-INFO-all --out
split_vcfs/${SAMPLE}/delly_autosomes.${SAMPLE}
vcftools --vcf manta_chroms.vcf --indv $SAMPLE --recode --recode-INFO-all --out
split_vcfs/${SAMPLE}/manta_autosomes.${SAMPLE}
done
```

Merging with genotype info

Splitting up each individual sample's 3 VCF files into het, homref, and homalt & merging across tools with SURVIVOR (but within samples)

```
module load BCFtools/1.16-GCC-11.3.0

cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_survivor

DIR=/nesi/nobackup/uoa02613/kstuart_projects/programs/SURVIVOR/Debug/

for SAMPLE in $(cat  /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/sample_file_ids_highcov.txt);
do
cd split_vcfs/${SAMPLE}/
grep "^#\|0/0" delly_autosomes.${SAMPLE}.recode.vcf > delly_${SAMPLE}_homref.vcf
grep "^#\|0/1" delly_autosomes.${SAMPLE}.recode.vcf > delly_${SAMPLE}_het.vcf
grep "^#\|1/1" delly_autosomes.${SAMPLE}.recode.vcf > delly_${SAMPLE}_homalt.vcf

grep "^#\|0/0" manta_autosomes.${SAMPLE}.recode.vcf > manta_${SAMPLE}_homref.vcf
grep "^#\|0/1" manta_autosomes.${SAMPLE}.recode.vcf > manta_${SAMPLE}_het.vcf
grep "^#\|1/1" manta_autosomes.${SAMPLE}.recode.vcf > manta_${SAMPLE}_homalt.vcf

grep "^#\|0/0" smoove_autosomes.${SAMPLE}.recode.vcf > lumpy_${SAMPLE}_homref.vcf
grep "^#\|0/1" smoove_autosomes.${SAMPLE}.recode.vcf > lumpy_${SAMPLE}_het.vcf
grep "^#\|1/1" smoove_autosomes.${SAMPLE}.recode.vcf > lumpy_${SAMPLE}_homalt.vcf

ls *${SAMPLE}*homref.vcf > homref_${SAMPLE}
ls *${SAMPLE}*het.vcf > het_${SAMPLE}
ls *${SAMPLE}*homalt.vcf > homalt_${SAMPLE}

#merging WITHIN genotype to make sure genotype is also in consensus (because SURVIVOR doesn't have a genotype option)

${DIR}/SURVIVOR merge homref_${SAMPLE} 1000 2 1 1 0 30 ${SAMPLE}_survivor_homref.vcf

${DIR}/SURVIVOR merge het_${SAMPLE} 1000 2 1 1 0 30 ${SAMPLE}_survivor_het.vcf

${DIR}/SURVIVOR merge homalt_${SAMPLE} 1000 2 1 1 0 30 ${SAMPLE}_survivor_homalt.vcf

grep "^#" ${SAMPLE}_survivor_homref.vcf > ${SAMPLE}_survivor_header
grep -v "^#" ${SAMPLE}_survivor_homref.vcf > ${SAMPLE}_survivor_homref_SNPs.vcf
grep -v "^#" ${SAMPLE}_survivor_het.vcf > ${SAMPLE}_survivor_het_SNPs.vcf
grep -v "^#" ${SAMPLE}_survivor_homalt.vcf > ${SAMPLE}_survivor_homalt_SNPs.vcf

cat ${SAMPLE}_survivor_header ${SAMPLE}_survivor_homref_SNPs.vcf  ${SAMPLE}_survivor_het_SNPs.vcf
 ${SAMPLE}_survivor_homalt_SNPs.vcf > ${SAMPLE}_survivor_unsorted.vcf

bcftools sort ${SAMPLE}_survivor_unsorted.vcf > ${SAMPLE}_survivor_v2.vcf
```

```
echo ${SAMPLE}
grep -v "^#" ${SAMPLE}_survivor_v2.vcf | wc -l

#Make sure the final column contains GT info
awk -F'\t' 'BEGIN{OFS="\t"} { if ($12 == "./.:NaN:0:0,0:--:NaN:NaN:NaN:NAN:NAN:NAN") $12 = $11; print }' ${SAMPLE}_survivor_v2.vcf
> ${SAMPLE}_survivor_v3.vcf

cd ../../

done
```

The absent and present VCF files should have at least 2 genotypes (that are the same).
The final per-sample VCF should have only one genotype in it, because we merged across genotypes.

What to do about insertions?

At this stage the dataset may not have many insertions (INS). This is because LUMPY doesn't pick them up, and so it depends on the consensus between MANTA and DELLY as to how many you will have in your final data set. The way I see it, your obvious choices at this stage are:

1) Not care about it, and just proceed with the SVs you have (many papers do this). **DONE**

2) Choose to use a 4th program that picks up INS. You could even swap out LUMPY. Other programs I have looked into are GRIDDS, didn't work but I didn't troubleshoot it for too long.

3) Carry some of the DELLY and MANTA insertions back into the dataset using some other level of criteria

Then merge across samples

```
cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_survivor
ls */*/*_survivor_v3.vcf > allsample_files
DIR=/nesi/nobackup/uoa02613/kstuart_projects/programs/SURVIVOR/Debug/
${DIR}/SURVIVOR merge allsample_files 1000 1 1 1 0 30 merged_rep_new.vcf

#exclude long SVs and high missingness
grep -v "^#" merged_rep_new.vcf | sed 's/;/\t/g' | cut -f 1,2,3,10 | sed 's/SVLEN=\|-//g' | awk '$4 > 50000' | cut -f3 > snpid_longSvs.txt

module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1
vcftools --vcf merged_rep_new.vcf --exclude snpid_longSvs.txt --max-missing 0.5 --mac 1 --recode --recode-INFO-all --out hihi_size
#remove TRA
grep -v "SVTYPE=TRA"  hihi_size.recode.vcf > hihi_size2.recode.vcf

#remove duplicates
grep -v "^#" hihi_size2.recode.vcf | cut -f 3 | sort | uniq -d > snpid_duplicatedSvs.txt
vcftools --vcf hihi_size2.recode.vcf --exclude snpid_duplicatedSvs.txt --recode --recode-INFO-all --out hihi_filtered

#THINKING: not included yet - probably will
#Remove variants without  (https://onlinelibrary.wiley.com/doi/10.1111/eva.13652)
##

#count number of genotypes for each SNP
```

```
grep -v "^#" hihi_filtered.recode.vcf | cut -f3 > genofilter_SNPID.txt
grep -v "^#" hihi_filtered.recode.vcf | awk '{print gsub(/0\/0/, "&")}' > genofilter_homr.txt
grep -v "^#" hihi_filtered.recode.vcf | awk '{print gsub(/0\/1/, "&")}' > genofilter_het.txt
grep -v "^#" hihi_filtered.recode.vcf | awk '{print gsub(/1\/1/, "&")}'  > genofilter_hom.txt

#final info file has the final column with a count of how many snps had a pass
paste genofilter_SNPID.txt <(sed 's/[[:space:]]\+/\t/g' genofilter_homr.txt | cut -f 2 ) <(sed 's/[[:space:]]\+/\t/g' genofilter_het.txt | cut -f 2 ) <(sed
's/[[:space:]]\+/\t/g' genofilter_hom.txt | cut -f 2 ) > genofilter_both.txt

###old method:
#needs at least 2 hets and 2 refs or alts
awk '$3 > 1 && ($2 > 1 || $4 > 1)' genofilter_both.txt | cut -f1 > genofilter1_snplist.txt
vcftools --vcf hihi_filtered.recode.vcf --snps genofilter1_snplist.txt --recode --recode-INFO-all --out hihi_genofiltered

#decided to actually reinclude these, and exclude non-het SNPs let
#so for now I had to generate a new VCF files of the ones I gnored in the first round of manual curation, so I can make their plots using my
code on the curation page
vcftools --vcf hihi_filtered.recode.vcf --exclude genofilter1_snplist.txt --recode --recode-INFO-all --out hihi_genofilteredextra

###current method: make list of SNPs to exclude later
# and exclude ones that don't have any het (it's just 22), and 8 that remain in the final dataset.
awk '$3 < 1' genofilter_both.txt | cut -f1 > SVs_to_remove.txt
```

Plotting PCA and looking at het

```
module load PLINK/1.09b6.16
module load VCFtools/0.1.15-GCC-9.2.0-Perl-5.30.1
cd /nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_survivor

vcftools --vcf hihi_filtered.recode.vcf --het --out hihi_filtered

vcftools --vcf hihi_filtered.recode.vcf --plink --out hihi_filtered.plink
plink --file hihi_filtered.plink --pca --out hihi_filtered --make-rel  --allow-extra-chr --chr-set 27

module load R/4.1.0-gimkl-2020a
R
library(ggplot2)

setwd("/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/variant_calling/SV_survivor")

# PASS
pca.eigenvec <- read.table("hihi_filtered.eigenvec", sep=" ")
pca_g1 <- data.frame(PC1 = pca.eigenvec$V3,    # the first eigenvector
                     PC2 = pca.eigenvec$V4,    # the second eigenvector
                     PC3 = pca.eigenvec$V5,    # the second eigenvector
                     stringsAsFactors = FALSE)

population <- read.table("/nesi/nobackup/uoa00338/kstuart_projects/Nc3_HihiSV/data/sample_metadata.txt", header = T, sep = "\t")
pca_plot <- cbind(population,pca_g1)
```

```
ggplot(pca_plot ,aes(x=PC1,y=PC2, col = BAM_SIZE_G))+
geom_point(size=5,alpha=1)+
theme_classic(base_size = 18)

summary(lm(PC1 ~ BAM_SIZE_G, data=pca_plot )) #nonsig
summary(lm(PC2 ~ BAM_SIZE_G, data=pca_plot )) #nonsig

#need to also test insert size: https://accio.github.io/bioinformatics/2020/03/10/filter-bam-by-insert-size.html
```