Starling-May18
Projects/Katarina Stuart/KStuart.Starling-Aug18/Sv5_AustraliaWGS/Data/2022.11.05.SVsgenomevNA

PDF Version generated by

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

on

Feb 20, 2023 @10:22 AM NZDT

Table of Contents

2022.11.05.SVsgenomevNA

Calling SVs with the vNA genome

```
cd /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/scripts
```

MAPPING WITH BWA

Create a BWA genome database: (completed in batch 1 analysis)

```
#!/bin/bash
#PBS -N 2022-11-05.mapping.pbs
#PBS -V
#PBS -l nodes=1:ppn=16
#PBS -l mem=124gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -M katarina.stuart@student.unsw.edu.au
#PBS -m ae

module load bwa/0.7.17

cd /nesi/nobackup/uoa02613/kstuart_projects/At4_MynaStarling/data/resources

bwa index Svulgaris_genomic.fna
```

Trimming with TrimGalore:

```
#!/bin/bash
#PBS -N 2022-11-05.trim.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=64gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@student.unsw.edu.au
#PBS -m ae
#PBS -J 01-24

module load trimgalore/0.6.5

FILE=$(sed "${SLURM_ARRAY_TASK_ID}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
SAMPLE=$(basename $FILE .1.fq.gz)
echo "working with sample:" $SAMPLE

mkdir /nesi/nobackup/uoa02613/kstuart_projects/At4_MynaStarling/rawdata/starling_AU_wgs_trimmed/${SAMPLE}

OUTPUT_DIR=/nesi/nobackup/uoa02613/kstuart_projects/At4_MynaStarling/rawdata/starling_AU_wgs_trimmed/${SAMPLE}
RAW_DATA_R1=/srv/scratch/canetoad/Stuart.Starling-Feb20/${SAMPLE}*1.fq.gz
RAW_DATA_R2=/srv/scratch/canetoad/Stuart.Starling-Feb20/${SAMPLE}*2.fq.gz

trim_galore -j 16 -o ${OUTPUT_DIR} --fastqc --paired ${RAW_DATA_R1} ${RAW_DATA_R2}
```

Aligning with bwa mem:

```
#!/bin/bash
#PBS -N 2022-11-05.VarCalling_starling_AU_wgs_map.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=64gb
#PBS -l walltime=12:00:00
#PBS -j oe
```

```
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
#PBS -J 01-24

# load modules
module load bwa/0.7.17
module load samtools/1.10

cd /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping

# set paths
FILE=$(sed "${PBS_ARRAY_INDEX}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
SAMPLE=$(basename $FILE .1.fq.gz)
echo "working with sample:" $SAMPLE

GENOME=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/resources/Svulgaris_genomic.fna
TRIM_DATA_R1=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/trimmed/${SAMPLE}*1.fq.gz
TRIM_DATA_R2=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/trimmed/${SAMPLE}*2.fq.gz
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping

# Map the reads
bwa mem -t ${PBS_ARRAY_INDEX} \
-R "@RG\tID:${SAMPLE}\tLB:${SAMPLE}_WGS\tPL:ILLUMINA\tSM:${SAMPLE}" \
-M ${GENOME} ${TRIM_DATA_R1} ${TRIM_DATA_R2} | \
samtools sort | samtools view -O BAM -o ${OUT_DIR}/${SAMPLE}.sorted.bam

# Check output
#samtools flagstat ${OUTPUT}

# Generate index
samtools index -@ ${PBS_ARRAY_INDEX} ${OUT_DIR}/${SAMPLE}.sorted.bam
```

Mark duplicates with picard:

```
#!/bin/bash
#PBS -N 2022-11-07.VarCalling_starling_AU_wgs_dup.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=64gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
#PBS -J 01-24

# load modules
module load samtools/1.10

# set paths
FILE=$(sed "${PBS_ARRAY_INDEX}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
SAMPLE=$(basename $FILE .1.fq.gz)
echo "working with sample:" $SAMPLE

OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping

#Mark Duplicates
export _JAVA_OPTIONS="-Xmx120g"
java -Xmx48g -jar /apps/picard/2.18.26/picard.jar MarkDuplicates INPUT=${OUT_DIR}/${SAMPLE}.sorted.bam OUTPUT=${OUT_DIR}/${SAMPLE}.sorted.dup.bam METRICS_FILE=${OUT

# Generate index
samtools index -@ ${PBS_ARRAY_INDEX} ${OUT_DIR}/${SAMPLE}.sorted.dup.bam
```

Lumpy_SV:

Preprocessing for BWA mem BAM files:

```
#!/bin/bash
#PBS -N 2022-11-10.SVCalling_starlingwgs_lumpy_preprocessing.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=64gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
#PBS -J 01-24

# load modules
module purge
module load samtools/1.10
LUMPY=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/lumpy-sv

SAMPLE=$(sed "${PBS_ARRAY_INDEX}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)

# set paths
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
BAM=${DIR}/${SAMPLE}.sorted.dup.bam

## BWA mem file pre-processing

# Extract the discordant paired-end alignments.
# The samtools view -F1294 option means “do not show reads with flags containing any of these values”, effectively excluding reads with the checked characteristics from the ouput.
samtools view -b -F 1294 ${BAM} > ${DIR}/${SAMPLE}.discordants.unsorted.bam

# Extract the split-read alignments
samtools view -h ${BAM} | \
${LUMPY}/scripts/extractSplitReads_BwaMem -i stdin | \
samtools view -Sb - > ${DIR}/${SAMPLE}.splitters.unsorted.bam

# Sort both alignments
samtools sort ${DIR}/${SAMPLE}.discordants.unsorted.bam -o ${DIR}/${SAMPLE}.discordants.bam
samtools sort ${DIR}/${SAMPLE}.splitters.unsorted.bam -o ${DIR}/${SAMPLE}.splitters.bam

rm ${DIR}/${SAMPLE}.discordants.unsorted.bam
rm ${DIR}/${SAMPLE}.splitters.unsorted.bam
```

Histogram profiling:

```
#!/bin/bash
#PBS -N 2022-11-10.SVCalling_starlingwgs_lumpy_histo.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=10gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
#PBS -J 01-24

# load modules
module purge
module load samtools/1.10
LUMPY=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/lumpy-sv

SAMPLE=$(sed "${PBS_ARRAY_INDEX}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)

# set paths
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy/histo

# Histogram profiling
samtools view ${BAM} | tail -n+100000 |  ${LUMPY}/scripts/pairend_distro.py -r 151 -X 4 -N 10000 -o ${OUT_DIR}/${SAMPLE}.histo | tee ${OUT_DIR}/${SAMPLE}.hist.stdout
```

Running lumpy:

```
#!/bin/bash
#PBS -N 2022-11-10.SVCalling_starlingwgs_lumpy_lumpy.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=120gb
#PBS -l walltime=12:00:00
```

```
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae

# load modules
module purge
module load samtools/1.10
LUMPY=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/lumpy-sv

# set paths
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy

cd ${OUT_DIR}

# create list of input BAM files and other information for LUMPY. First variable creation line creates the "-pe" lines, the second creates the "-sr" lines. We need one for each file

FILE_LIST=""

for SAMPLE_NUMBER in {1..24}
do
SAMPLE=$(sed "${SAMPLE_NUMBER}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
DISC=${DIR}/${SAMPLE}.discordants.bam
SPLIT=${DIR}/${SAMPLE}.splitters.bam
HISTO=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy/histo/${SAMPLE}.histo
MEAN=$(cut -f1 /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy/histo/${SAMPLE}.hist.stdout)
STDV=$(cut -f2 /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy/histo/${SAMPLE}.hist.stdout)
FILE_LIST="${FILE_LIST} "-pe" "id:"${SAMPLE}",bam_file:"${DISC}",read_group:"${SAMPLE}",histo_file:"${HISTO}","${MEAN}","${STDV}",read_length:151,min_non_overlap:151,discordant_z:5,back_dist
FILE_LIST="${FILE_LIST} "-sr" "id:"${SAMPLE}",bam_file:"${SPLIT}",back_distance:10,weight:1,min_mapping_threshold:20""
done

echo ${FILE_LIST}

# Run LUMPY
${LUMPY}/bin/lumpy -mw 4 -tt 0 ${FILE_LIST} > ${OUT_DIR}/starling_wgs_24NAref_lumpy.vcf
```

Running SVtyper:

```
#!/bin/bash
#PBS -N 2022-11-15.SVCalling_starlingwgs_lumpy_svtyper.pbs
#PBS -l nodes=1:ppn=2
#PBS -l mem=120gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae

# load environments
source /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/etc/profile.d/conda.sh
conda activate /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/envs/svtyper

# set paths
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy
cd ${DIR}

# create list of BAM
FILE_LIST_BAM=""
for SAMPLE_NUMBER in {2..24}
do
SAMPLE=$(sed "${SAMPLE_NUMBER}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
FILE_LIST_BAM="${FILE_LIST_BAM}","${BAM}"
done
echo ${FILE_LIST_BAM}

# create list of split reads
FILE_LIST_SPLIT=""
for SAMPLE_NUMBER in {2..24}
do
SAMPLE=$(sed "${SAMPLE_NUMBER}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
BAM2=${DIR}/${SAMPLE}.splitters.bam
```

```
FILE_LIST_SPLIT="${FILE_LIST_SPLIT}","${BAM2}"
done
echo ${FILE_LIST_SPLIT}

# SVtyper
svtyper \
--max_reads 2000 \
-B au01_lem.speedseq.bam,au02_lem.speedseq.bam,au03_mai.speedseq.bam,au04_mai.speedseq.bam,au05_men.speedseq.bam,au06_men.speedse
-S
au01_lem.speedseq.splitters.bam,au02_lem.speedseq.splitters.bam,au03_mai.speedseq.splitters.bam,au04_mai.speedseq.splitters.bam,au05_men.spee
\
-i ${OUT_DIR}/starling_wgs_24NAref_lumpy.vcf > ${OUT_DIR}/starling_wgs_24NAref_lumpy.gt2000.vcf
```

Delly:

Step1: SV calling done separately for each sample

```
#!/bin/bash
#PBS -N 2022-11-10.SVCalling_starlingwgs_delly_step1.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=20gb
#PBS -l walltime=24:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
#PBS -J 01-24

# load environment
source /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/etc/profile.d/conda.sh
conda activate /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/envs/delly

SAMPLE=$(sed "${PBS_ARRAY_INDEX}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
echo "working with sample:" ${SAMPLE}

# set paths
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
BAM=${DIR}/${SAMPLE}.sorted.dup.bam
GENOME=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/resources/Svulgaris_genomic.fna
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/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
#PBS -N 2022-11-10.SVCalling_starlingwgs_delly_step2.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=10gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae

# load environment
source /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/etc/profile.d/conda.sh
conda activate /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/envs/delly

#create sample BCF file input list
BCF_LIST=""
```

```
for SAMPLE in $(cat /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
do
BCF_LIST="${BCF_LIST} /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/delly/${SAMPLE}.bcf"
done

echo $BCF_LIST

# set paths
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/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
#PBS -N 2022-11-10.SVCalling_starlingwgs_delly_step3.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=20gb
#PBS -l walltime=24:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae
#PBS -J 01-24

# load environment
source /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/etc/profile.d/conda.sh
conda activate /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/envs/delly

SAMPLE=$(sed "${PBS_ARRAY_INDEX}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
echo "working with sample:" ${SAMPLE}

# set paths
GENOME=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/resources/Svulgaris_genomic.fna
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/delly
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
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
#PBS -N 2022-11-10.SVCalling_starlingwgs_delly_step4.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=20gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae

# load modules
module purge
module load samtools/1.10

#create sample BCF file input list
BCF_LIST=""

for SAMPLE in $(cat /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
do
BCF_LIST="${BCF_LIST} /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/delly/${SAMPLE}.rep.geno.bcf"
done

echo $BCF_LIST

# set paths
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/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
#PBS -N 2022-11-10.SVCalling_starlingwgs_delly_step5.pbs
#PBS -l nodes=1:ppn=4
#PBS -l mem=20gb
#PBS -l walltime=12:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae

# load modules
module purge
module load samtools/1.10

# set paths
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/delly

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

Manta:

Manta SV calling requires just a single line

```
#!/bin/bash
#PBS -N 2022-11-10.SVCalling_starlingwgs_manta.pbs
#PBS -l nodes=1:ppn=16
#PBS -l mem=120gb
#PBS -l walltime=48:00:00
#PBS -j oe
#PBS -M katarina.stuart@unsw.edu.au
#PBS -m ae

# load environment
source /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/etc/profile.d/conda.sh
conda activate /srv/scratch/z5188231/KStuart.Starling-Aug18/programs/anaconda3/envs/manta

# set paths
GENOME=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/resources/Svulgaris_genomic.fna
OUT_DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/manta

# create list of input BAM files for Manta
FILE_LIST=""
for SAMPLE_NUMBER in {1..24}
do
SAMPLE=$(sed "${SAMPLE_NUMBER}q;d" /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/sample_individual_list.txt)
DIR=/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/mapping
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
/srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/manta/runWorkflow.py
```

Survivor:

linking and filtering SVs

```
cd /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/survivor
ln -s /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/lumpy/starling_wgs_24NAref_lumpy.gt2000.vcf
ln -s /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/delly/merged_rep_geno.vcf
ln -s /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/manta/results/variants/diploidSV.vcf

module load vcftools/0.1.16 samtools/1.10

vcftools --vcf starling_wgs_24NAref_lumpy.gt2000.vcf --min-meanDP 5 --recode --recode-INFO-all --out lumpysv_strvar_repeats.gt.named.pass
vcftools --vcf merged_rep_geno.vcf --remove-filtered-all --recode --recode-INFO-all --out merged_rep_geno.pass
vcftools --vcf diploidSV.vcf --remove-filtered-all --recode --recode-INFO-all --out diploidSV.pass
```

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

```
mkdir split_vcfs

for SAMPLE in $(cat /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/survivor/sampleorder_24indv.txt );
do
echo ${SAMPLE}
mkdir split_vcfs/${SAMPLE}
vcftools --vcf lumpysv_strvar_repeats.gt.named.pass.recode.vcf --indv $SAMPLE --recode --recode-INFO-all --out split_vcfs/${SAMPLE}/lumpysv_strvar.gt.named.${SAMPLE}
vcftools --vcf merged_rep_geno.pass.recode.vcf --indv $SAMPLE --recode --recode-INFO-all --out split_vcfs/${SAMPLE}/merged_geno.${SAMPLE}
vcftools --vcf diploidSV.pass.recode.vcf --indv $SAMPLE --recode --recode-INFO-all --out split_vcfs/${SAMPLE}/diploidSV.${SAMPLE}
done
```

Modifying SURVIVOR pipeline so that we can also incorporate genotype calls into the merging process.

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

```
DIR=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/SURVIVOR/Debug



for SAMPLE in $(cat /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/survivor/sampleorder_24indv.txt );
do

cd split_vcfs/${SAMPLE}/
grep "^#\|0/0" merged_geno.${SAMPLE}.recode.vcf > delly_${SAMPLE}_homref.vcf
grep "^#\|0/1" merged_geno.${SAMPLE}.recode.vcf > delly_${SAMPLE}_het.vcf
grep "^#\|1/1" merged_geno.${SAMPLE}.recode.vcf > delly_${SAMPLE}_homalt.vcf

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

grep "^#\|0/0" lumpysv_strvar.gt.named.${SAMPLE}.recode.vcf > lumpy_${SAMPLE}_homref.vcf
grep "^#\|0/1" lumpysv_strvar.gt.named.${SAMPLE}.recode.vcf > lumpy_${SAMPLE}_mai_het.vcf
grep "^#\|1/1" lumpysv_strvar.gt.named.${SAMPLE}.recode.vcf > lumpy_${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}
```

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

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

cd ../../

done
```

The homref, het, and homalt 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.

Then merge across samples

```
cd /srv/scratch/canetoad/Stuart.Starling-Feb20/SV_calling/survivor/split_vcfs
ls */*_survivor_v2.vcf > allsample_files
DIR=/srv/scratch/z5188231/KStuart.Starling-Aug18/programs/SURVIVOR/Debug
${DIR}/SURVIVOR merge allsample_files 1000 1 1 1 0 30 merged_rep.vcf
```

Seems like GTs were all carried across properly (i.e. SURVIVOR didn't pull over NaN's when there were GT data available).

```
module load vcftools/0.1.16
vcftools --vcf merged_rep.vcf --maf 0.03 --max-missing 0.5 --recode --recode-INFO-all --out merged_rep_filtered

vcftools --vcf merged_rep_filtered.recode.vcf --het --out merged_rep_filtered.hetero

VCF=/srv/scratch/z5188231/KStuart.Starling-Aug18/Sv5_AustraliaWGS/data/snp_variants_processed/wgs_variantsgenotyped_filtered_maf005_r2_noIndel_WithIds_thin.recode.vcf
vcftools --vcf ${VCF} --het --out snps.hetero
```