Processing GBS data

We have a F1 population derived from a cross between Rabiosa and Sikem. The population contains 309 individuals (304 progeny and 2 Sikem parents and 3 Rabiosa parents) and and was sequenced using GBS (Genotype-By-Sequencing).

The general pipeline for processing the GBS data would be:

1. de-multiplex the raw GBS reads
2. quality check the reads, trim adapters and low-quality ends
3. map clean reads to reference
4. call variants
5. filter variants

Previously, Roland de-multiplexed the reads and Jenny did quality check and trimmed the reads. I started from the 3rd step.

On The Dude, activate my environment samtools:

conda activate samtools

Then start mapping using BWA (0.7.17), I used gnu-parallel to speed up the analysis:

Skip the index step as I have copied the indexed files to the working directory

Cleaned reads were copied to input folder

I used the haploid assembly of Rabiosa **hap_polca_flye_42x_miop10.fa** produced in previous blogs as the reference

ls input/ > myreads.txt

cat myreads.txt | parallel -j 6 -k "bwa mem -t 15 rabiosa/hap_polca_flye_42x_miop10.fa input/{} > tmp/{.}.sam"

Then convert sam to bam and sort the bam using samtools (0.1.19), this samtools has a different syntax, I don't know why. Every time we use samtools or other software, we should check their usage in advance.
mkdir tmp_bam_unsorted
ls tmp/ > mysam.txt
cat mysam.txt | parallel -j 90 -k "samtools view -Sbh tmp/{} > tmp_bam_unsorted/{.}.bam"

Next sort bam files
mkdir tmp_bam_sorted
ls tmp_bam_unsorted/ > mybam_unsorted.txt
cat mybam_unsorted.txt | parallel -j 30 -k "samtools sort -@ 3 tmp_bam_unsorted/{} tmp_bam_sorted/{.}"

Now call variant using bcftools (1.12)

mkdir tmp_vcf
bcftools mpileup -f rabiosa/hap_polca_flye_42x_miop10.fa tmp_bam_sorted/*.bam | bcftools call -mv > tmp_vcf/rabiosa.vcf

We found that most genotypes of Sekim parent individuals were missing, this could be due to the low quality of the GBS data (we have two Sikem individuals in the 309 individuals). We want to have genotypes of both parents in the vcf file, but the quality of GBS data of Sikem is low. Luckily, we have some whole-genome-sequencing pair-end reads for both Sikem and Rabiosa, so I align pair-end reads to the reference for both Sikem and Rabiosa to create two BAM and then intersect these two bams with GBS bams to only keep common alignments in the bams of Sikem and Rabiosa. Finally, use the intersected bams with all otehr GBS bams to call variants again.

Align pair-end reads to the reference, here I used the same BWA but a different version of samtools (1.12), so you could see the usage of samtools is different. Be careful of this change.

bwa index -a bwtsw purge_hap_polca_flye_42x_miop10/hap_polca_flye_42x_miop10.fa
bwa mem -t 45 purge_hap_polca_flye_42x_miop10/hap_polca_flye_42x_miop10.fa tmp/Ryegrass_SG700bp_trimmed_1P.fq tmp /Ryegrass_SG700bp_trimmed_2P.fq | samtools sort -m 10G -@ 45 -o purge_hap_polca_flye_42x_miop10/hap_short_sorted.bam
samtools view -b -f 2 -F 2304 -@ 45 purge_hap_polca_flye_42x_miop10/hap_short_sorted.bam > purge_hap_polca_flye_42x_miop10 /hap_short_sortted_filtered.bam

bwa mem -t 45 hap_polca_flye_42x_miop10.fa ../trimm/20201113.B-Sikem_2002_trimmed_1P.fq.gz ../trimm/20201113.B-Sikem_2002_trimmed_2P.fq.gz > Sikem_short.sam

samtools view -Sbh -f 2 -F 2304 -@ 48 Sikem_short.sam | samtools sort -@ 45 -o Sikem_short_sorted.bam

Intersect bams using bedtools (2.30.0), I randomly selected 10 GBS bam files to represent all the GBS bams, I guess 10 files are enough to include all the alignment sites of GBS reads.

**bedtools intersect -abam /home/yutachen/public/Jenny_Peter/NAM_parents/tmp/vcf_yutang/Sikem_short_sorted.bam **
-b tmp_bam_sorted/lmGbsJP_42_*.fastq.trim.bam > tmp_bam_sorted/PSikem.bam

**bedtools intersect -abam /home/yutachen/public/Yutangchen/Rabiosa_data/purge_hap_polca_flye_42x_miop10/hap_short_sortted_filtered.bam **
-b tmp_bam_sorted/lmGbsJP_42_*.fastq.trim.bam > tmp_bam_sorted/PRabiosa.bam

Now call variant again with all the bams:

bcftools mpileup -f rabiosa/hap_polca_flye_42x_miop10.fa tmp_bam_sorted/*.bam | bcftools call -mv > tmp_vcf/rabiosa_ps_pr.vcf

While waiting for the variant calling program to finish, make a sample name list file in order to replace sample names in the vcf file.

ls tmp_bam_sorted/* > sample.txt

cut -f2 -d "/" sample.txt | cut -f1 -d "." > sample_name.txt

It looks like this, one individual per line:

(samtools) -bash-4.2$ head sample_name.txt
lmGbsJP_01_1
lmGbsJP_01_3
lmGbsJP_01_4
lmGbsJP_01_5
lmGbsJP_01_6
lmGbsJP_01_8
lmGbsJP_02_2
lmGbsJP_02_4
lmGbsJP_02_8
lmGbsJP_03_4

Once get the vcf file, reheader the vcf file (replace the sample name with the list made in last step)

bgzip -c tmp_vcf/rabiosa_ps_pr.vcf > tmp_vcf/rabiosa_ps_pr.vcf.gz # here I zipped the vcf file, but I think without zipping the file we can still reheader the vcf file
bcftools reheader -s sample_name.txt tmp_vcf/rabiosa_ps_pr.vcf.gz > tmp_vcf/rabiosa_ps_pr_renamed.vcf.gz

Final step, filter the vcf

**bcftools view -s ^lmGbsJP_M2002_18_1,lmGbsJP_M2002_18_2,lmGbsJP_M2402_16_1,lmGbsJP_M2402_16_2,lmGbsJP_M2402_16_3 tmp_vcf/rabiosa_ps_pr_renamed.vcf.gz | **
bcftools filter -i 'QUAL>30 && DP>100' - | bcftools view -g het -v snps -m2 -M2 -q 0.05 -i 'F_MISSING<0.1' > tmp_vcf/rabiosa_ps_pr_renamed.filtered.vcf

Here I used bcftools view s ^names to exclude the five parent individuals in the vcf file and the filter variants based on mapping quality and read depth and finally only keeps bi-allelic SNPs with minor allele frequency greater than 0.05 and genotype calling rate greater than 0.9.

The filtered vcf file can be directly used as the input for onemap, which is a genetic linkage map builder implemented in R.

If you find there's anything wrong or unclear in the pipeline, please leave a commnet below or feel free to directly contact me via any approaches.

Best wishes,

Yutang