**根据Hi-C map调整scaffold到chromosome水平**

bamToBed -i /hicpro\_output/bowtie\_results/bwt2/GD1913\_2/LJM\_GD1913\_2.contigs.bwt2pairs.bam > alignment.bed

**1. 将hifi raw data经过Hicanu 2.1进行组装，**

samtools view filename.bam | awk '{OFS="\t"}; print ">"$1"\n"$10' > GD1913\_2\_all.fasta
/data/Liangjunmin/opt/biosoft/canu-2.1.1/build/bin/canu -p GD1913\_2 -d ./ genomeSize=1600000000 -pacbio-hifi /data/Liangjunmin/GD1913\_2/HIFI/GD1913\_2\_all.fasta

**2. 使用blast比去除低coverage序列、污染序列和线粒体序列**<https://github.com/JanaSperschneider/GenomeAssemblyTools/tree/master/ContaminantScreening>

GD1913 hicanu拼接获得4056contigs,去掉测序深度低于2X的序列剩余2492contigs,根据线粒体库比对，删掉63个比对为线粒体的contig。最终2429contigs用于二倍体拆分

#Map long reads back to the assembly to get coverage for each contig
/data/Liangjunmin/opt/biosoft/minimap2/minimap2 -ax asm20 /data/Liangjunmin/GD1913\_2/HIFI/canu/asm4\_canu2.1/GD1913\_2.contigs.fasta /data/Liangjunmin/GD1913\_2/HIFI/GD1913\_2\_all.fasta --secondary=no -o mapping.sam
#Use BBMap's pileup.sh tool to calculate read coverage and GC content per contig.
/data/Liangjunmin/opt/biosoft/bbmap/pileup.sh in=mapping.sam out=contig\_coverage.txt

**3. 利用purge\_haplotig进行二倍体拆分，**

conda install -y purge\_haplogigs
gzip subreads.fasta
minimap2 -ax map-pb genome.fasta subreads.fasta.gz | samtools view -hF 256 | samtools sort -@ 20 -m 10G - aligned.bam -T tmp.ali
purge\_haplotigs hist -b aligned.bam -g genome.fasta -t 20
#cutoffs for low coverage, low point between the two peaks, and high coverage
purge\_haplotigs contigcov -i aligned.bam.genecov -l 30 -m 80 -h 145 -o coverage\_stats.csv
samtools faidx genome.fasta
purge\_haplotigs purge -g genome.fasta -c coverage\_stats.csv aligned.bam
#生成结果其中curated.fasta是primary haplotig，记作hap1, curated.haplotig.fasta是secondary haplotig,记作hap2
#分别对curated.fasta 和curated.haplotig.fasta进行BUSCO分析，发现两个haplotigBUSCO有重复，需要判断这些重复的归属，由于这种重复只能通过BUSCO有限的1355个同源蛋白发现，所以对两个haplotig得到的重复不全面，需要基于全局比对，因此使用HaploMerge2

**4.使用Haplomerger2利用全局比对，获得准确全局比对结果，对hap1和hap2之间、hap1内部contig之间以及hap2内部contig之间的重复进行判断，原则如下：**

For duplicates between hap1 and hap2,

1. if part1 is far larger than part2 and part2 was assigned to a single scaffold with nothing else, I kept all sequences of this contig to the haplotype holding the longer part.

2. If the two parts have similar length in two haplotypes, such as tig00009080 (1.25M, 604KB and 643KB in each haplotype), I split this contig to two haplotypes

3. However, there are many cases. The part1 is not far longer or shorter than part 2, such as tig00060620 (8719bp in hap1 and 5961bp in hap2). I didn't split them and keep all sequences of this contig to the haplotype having the longer part.

For duplicates within hap1 or hap2

1.  Part1 is far longer than part2 and part 2 was assigned to a single scaffold, I kept all sequences to the hap with longer part;

2. Part 1 is a little longer than part2 and part 1 was assigned to a single scaffold. such as tig00029892\_pilon (6310bp in contig74 vs 15000bp in contig473) , tig00056055 and tig00097736, I also kept all sequences to the scaffold with the longer part. (This dispose can increase scaffold number, Shall we do like this?)

3.  part 1 is not far longer or shorter than part 2, such as tig00047080 (5584bp in contig80 vs 6961bp in contig128) and tig00060358, I didn't spilit them but kept all sequences to the contig with longer part.

In addition, I found HaploMeger2 separated some contigs to 2-3 parts labeled with "\_1","\_2" or "\_3" at the end of old scaffold names. It treated these contigs as misjoins and split them. These contigs are also duplicated between two haplotypes or within each haplotype and were not included in above three sheets of duplicate. I summarized them in a new sheet as  "duplicate by split".

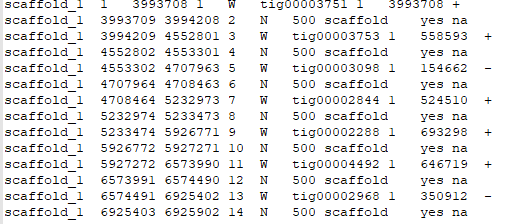
但不建议使用该软件拆分二倍体

wget https://github.com/mapleforest/HaploMerger2 -C ~/opt/biosoft
#利用winmask进行软屏蔽
echo "PATH=$PATH:/data/Liangjunmin/opt/biosoft/HaploMerger2/winmask/windowmasker" >> ~/.bashrc
windowmasker -checkdup true -mk\_counts -in genome.fasta -infmt fasta -out GD1913.count -sformat obinary
windowmasker -ustat GD1913.count -in genome.fasta -out genome.fasta.masked -outfmt fasta -dust true
#利用fanaPolishing.pl对拼好的基因组进行clean品
gunzip -c genome.fasta.masked.gz | /data/Liangjunmin/opt/biosoft/HaploMerger2\_20180603/bin/fanaPolishing.pl --legalizing --maskShortPortion=1 --noleand:q
ingN --removeShortSeq=1 > genome\_cleaned.fa
gzip genome\_clean.fa
#修改模板程序中的路径
cp -r project\_template project\_template\_bck
cd project\_template
grep'\.\.' hm\*#确认要被更改的信息
sed -i 's/\.\./\/data\/Liangjunmin\/opt\/biosoft\/HaploMerger2\_20180603/' hm\*
mkdir -p HaploMerger
cd HaploMerger
cp ~/opt/biosoft/HaploMerger2\_20180603/project\_template/hm\* ./
cp ~/opt/biosoft/HaploMerger2\_20180603/project\_template/\*.ctl ./
#运行主程序
sh ./hm.batchA1.initiation\_and\_all\_lastz pilon1\_cleaned
sh ./hm.batchA2.chainNet\_and\_netToMaf pilon1\_cleaned
sh ./hm.batchA3.misjoin\_processing
#对于hm.batchA1-A3，作者建议迭代2~3次，假如你迭代了三次，那么每次的结果应该是contigs\_wm > contigs\_wm\_A > contigs\_wm\_A\_A > contigs\_wm\_A\_A\_A
#结果在genome\_A\_A\_Ax.result文件夹中，其中hm.new.scaffolds里面包含了全局比对的结果

**4. 使用SALSA对拆分完的单倍体分别用HiC数据链接到scaffold水平**

#使用前对hic数据进行处理
mkdir -p raw\_dir filter\_dir tem\_dir pair\_dir rep\_dir
bwa index haplotig1.fa
bwa mem -t 12 haplotig1.fa /data/Liangjunmin/GD1913\_2/HiC/Clean\_data/LJM4\_R1.fastq.gz | samtools view -@ 10 -Sb > ./raw\_dir/LJM4\_1.bam
bwa mem -t 12 haplotig1.fa /data/Liangjunmin/GD1913\_2/HiC/Clean\_data/LJM4\_R2.fastq.gz | samtools view -@ 10 -Sb > ./raw\_dir/LJM4\_2.bam
samtools view -h ./raw\_dir/LJM4\_1.bam | perl /data/Liangjunmin/opt/biosoft/mapping\_pipeline/filter\_five\_end.pl | samtools view -Sb - > ./filter\_dir/LJM4\_1.bam
samtools view -h ./raw\_dir/LJM4\_2.bam | perl /data/Liangjunmin/opt/biosoft/mapping\_pipeline/filter\_five\_end.pl | samtools view -Sb - > ./filter\_dir/LJM4\_2.bam
perl /data/Liangjunmin/opt/biosoft/mapping\_pipeline/two\_read\_bam\_combiner.pl ./filter\_dir/LJM4\_1.bam ./filter\_dir/LJM4\_2.bam samtools 12 | samtools view -bS -t haplotig1.fa.fai - | sam tools sort -@ 12 -o ./tem\_dir/LJM4.bam
picard AddOrReplaceReadGroups -INPUT ./tem\_dir/LJM4.bam -OUTPUT ./pair\_dir/LJM4.bam -ID LJM4 -LB LJM4 -SM GD1913\_2 -PL ILLUMINA -PU none
picard MarkDuplicates --INPUT ./pair\_dir/LJM4.bam --OUTPUT ./rep\_dir/LJM4\_LABEL.bam --METRICS\_FILE ./rep\_dir/LJM4\_LABEL.metrics --TMP\_DIR ./tem\_dir/ --ASSUME\_SORTED TRUE --VALIDATION\_ST RINGENCY LENIENT --REMOVE\_DUPLICATES TRUE
samtools index ./rep\_dir/LJM4\_LABEL.bam
perl /data/Liangjunmin/opt/biosoft/mapping\_pipeline/get\_stats.pl ./rep\_dir/LJM4\_LABEL.bam > ./rep\_dir/LJM4\_LABEL.bam.stats
echo "Finished Mapping Pipeline through Duplicate Removal"sort -k 4 alignment.bed > tmp && mv tmp alignment.bed
samtools faidx ~/GD1913\_2/HIFI/canu/asm2/GD1913\_2.contigs.fasta
python /data/Liangjunmin/opt/biosoft/SALSA/run\_pipeline.py -a contigs.fasta -l contigs.fasta.fai -b alignment.bed -e GATC -o scaffolds

结果生成hap1\_scaffolds.agp文件，最后一列+ -表示序列的方向



**5. 利用hicpro+hicplotter对SALSA生成的scaffols进行可视化**

cat scaffold1.fasta scaffold2.fasta .....> changed\_total.fasta
#hicpro获得hic data和各scaffold的关联关系，记得修改config\_hicpro.txt中的路径
mkdir -p 01.ref
ln -s /mnt/data/liangjunmin/asm4\_canu2.1/hap1\_2/emboss/hap1/chr\_6th/changed\_total.fasta 01.ref/
perl /data/Liangjunmin/opt/biosoft/HiC-Pro\_2.11.4/bin/utils/digest\_genome.py -r mboi -o changed\_total.MboI.txt ./01.ref/changed\_total.fasta
samtools faidx 01.ref/changed\_total.fasta
awk '{print $1,$2}' 01.ref/changed\_total.fasta.fai > 01.ref/changed\_total.fasta.size
bowtie2-build --threads 40 01.ref/changed\_total.fasta changed\_total
mkdir -p 02.reads/changed\_total
ln -s /data/Liangjunmin/GD1913\_2/HiC/Clean\_data/LJM4\_R\*.fastq.gz ./02.reads/changed\_total
/data/Liangjunmin/opt/biosoft/HiC-Pro\_2.11.4/bin/HiC-Pro -i ./02.reads/ -o hicpro\_output -c config-hicpro.txt
#hicplotter
bash hicplotter.sh

hicplotter.sh 如下

#!/bin/bash

while read line

do

if [[ $line =~ ^# ]];then

continue

fi

python /data/Liangjunmin/opt/biosoft/HiCPlotter/HiCPlotter.py -f /mnt/data/liangjunmin/asm4\_canu2.1/hap1\_2/emboss/hap1/chr\_6th/hicpro1/hicpro\_output/hic\_results/matrix/changed\_total/iced/40000/changed\_total\_40000\_iced.matrix -o ${line} -r 40000 -tri 1 -bed /mnt/data/liangjunmin/asm4\_canu2.1/hap1\_2/emboss/hap1/chr\_6th/hicpro1/hicpro\_output/hic\_results/matrix/changed\_total/raw/40000/changed\_total\_40000\_abs.bed -n ${line} -chr ${line} -ptr 1 -hmc 1

done<readme

**6. 根据hicmap图，hm\_new\_scaffolds, hap1\_scaffolds.agp, 以及序列的telomere情况，对hicmap中的misjoin进行manual correction.**

需要用到的调整脚本包括

#将每个scaffold中的contig生成一个独立fasta文件
conda install -c bioconda emboss
seqretsplit
Read sequences and write them to individual files
Input (gapped) sequence(s):hap1.fasta
Output (gapped) sequence(s):tig00003567.fasta
#reverse fasta
seqtk seq -r ${line}.fasta > ${line}"\_r.fasta"
#将多个调整好的contig组成一个fasta文件，去除换行，见chr.sh批量进行
cat ${line}.fasta gap.fasta >> temp.fasta
c="Chr11"
echo ${c}
awk '/^>/{print s? s"\n"$0:$0;s="";next}{s=s sprintf("%s",$0)}END{if(s)print s}' temp.fasta > temp1.fasta
union -sequence temp1.fasta -outseq temp2.fasta && rm -f temp.fasta temp1.fasta
sed "s/>.\*/>${c}/" temp2.fasta > ${c}.fasta && rm -f temp2.fasta

telomere\_CCCTAAA.pl查找端粒

