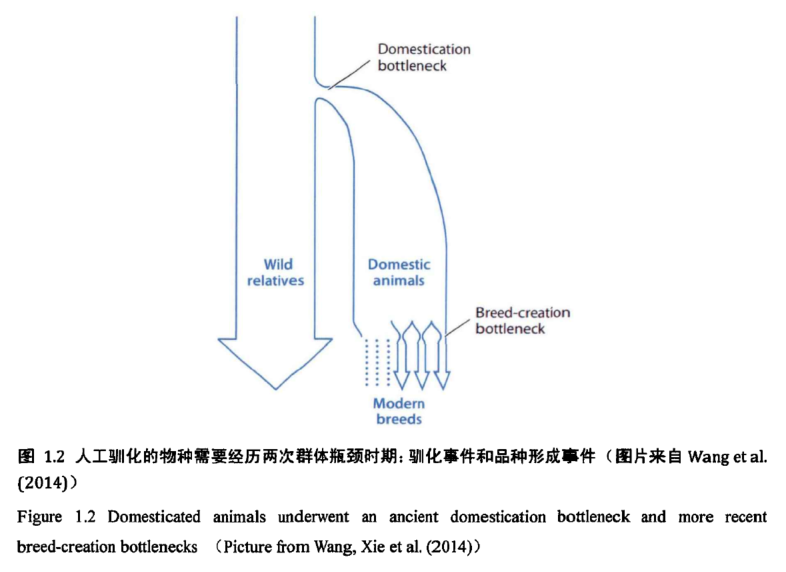
## 动物驯化及群体历史研究：

与驯化相关的研究热点主要集中在：家养动物何时何地被驯化？驯化途径是什么？驯化表型是什么？与驯化表型相关的基因有哪些？家养动物驯化地点大致主要集中在三个地方：近东地区、中国中部和安第斯山脉。几乎所有家养动物都是在近几千年或几万年内被驯化。

驯化的途径主要分为三种：共生、捕食和直接驯化。共生：指人类与被驯化物种的野外物种长期处在相同环境下，逐渐与人和谐共处达到驯化的目的，猫和狗是通过该方式驯化的；捕食：指人类相对被驯化动物是捕食者，当人类狩猎捕食的个体太多的时候，会将一些个体圈养起来，进而达到驯化的目的，牛与猪是通过该方式驯化：直接驯化：指人类为了某项具体的目的直接改造物种的方式，例如人类养蚕是为了丝绸织布，驯化马和骆能是为了运输。

人工驯化的家养动物通常需要经历两次群体瓶颈时期：驯化事件和品种形成事件。驯化事件指的是从野生状态到家养状态的过程，一般发生在一万年左右，例如灰狼驯化成狗的过程。品种形成事件则是指现代品种的形成，一般发生在最近几百年左右，例如各个不同品种狗的形成。驯化会导致一系列表型的显著改变，在家养动物中，这种表型的选择通常包括温顺程度、行为、体型大小与形态、皮肤和毛发颜色以及与繁殖相关性状等方面的变化。

### 等位基因频谱（SFS）：

基于 SFS 重构群体历史的常用软件有***∂a∂i***与 ***Fastsimcoal2***

***∂a∂i***：使用扩散近似法（diffusion approximation）来推断最多 3 个群体的种群大小变化、种群分化及迁移率等历史事件；

***Fastsimcoal2***：使用复合似然值（composite likelihood）来推断群体演化历程。

两者对于少于 4 个群体的历史推测有着相似的运算效率，但当群体数超过 3 个时，只能利用 ***Fastsimcoal2*** 进行计算。

计算过程中，***∂a∂i*** 与 ***Fastsimcoal2***都需要从高质量且独立的位点获得群体之间的频谱信息：即在计算 SFS 之前，必须要去掉变异位点之间的连锁不平衡，且尽可能利用高覆盖度高质量的全基因组，测序数据去生成 SFS。

相比 ***PSMC、MSMC***，***SFS*** 方法可以获得更为近期的群体历史动态变化，而且可以利用更大的群体信息得到更为精确的结果，同时不会增加计算负担。SFS 方法需要在计算时提供先验模型，然后从预先已知的先验分布中重复抽样来估计未知群体历史参数的后验分布。基于 SFS 的方法的另一个优点是可以重塑非常复杂的多群体历史动态，而且可以推断群体之间的迁移事件。

SFS 的方法需要用户提供可靠的先验模型才能得到具有生物学意义的结果。但在很多情况下，研究者并没有这样一个可靠的先验模型，所以大多数用户都会结合群体遗传结构分析的结果选择多个可能的模型分别进行计算，而且每个模型需要进行多次的bootstrap 并通过比较似然值来提高结果的准确性（似然值越大，所得到的参数越贴近实际数据）。不同模型之间，可以通过赤池信息量准则（Akaike Information Criterion, AIC）和贝叶斯信息准则（Bayesian Information Criterion, BIC）综合考虑自由参数与似然值进而选择最佳的模型。

针对每一个分离位点，如果我们不知道哪一个碱基是祖系状态，哪一个是后天得到的，那么这个时候 我们可以用**minor allele frequency(MAF)来描述，也就是“少数等位基因”频率。MAF的范围在1/n - 0.5之间。它也被称为site frequency spectrum(SFS)**。因为不能确定祖系状态，只能在“少数等位基因”，因而也被称为folded spectrum.

如果我们通过其他物种的基因组推测出了每一个位点的祖系状态，这个时候就可以用\*\*derived allele frequency(DAF)\*\*来表示位点变异情况，DAF的范围在1/n 到 (n-1)/n之间，相对应的，也被称为unfolded spectrum

在基因组数据分析中，我们在不知道祖系状态的情况下，通常将MAF<0.05的位点舍弃，因为这可能是测序错误或者有害突变，并不一定代表基因的多态性。但是这也有可能包含了DAF>0.95的位点，也就是会丢失掉很多进化上很重要的位点。

### 群体历史模拟中最优模型选择准则：

赤池信息量准则：

Akaike information criterion、简称AIC，是衡量统计模型拟合优良性的一种标准，是由日本统计学家赤池弘次创立和发展的。赤池信息量准则建立在熵的概念基础上，可以权衡所估计模型的复杂度和此模型拟合数据的优良性。

增加自由参数的数目提高了拟合的优良性，AIC鼓励数据拟合的优良性但尽量避免出现过度拟合（Overfitting）的情况。所以优先考虑的模型应是AIC值最小的那一个。赤池信息量准则的方法是寻找可以最好地解释数据但包含最少自由参数的模型

贝叶斯信息准则：

也称为Bayesian Information Criterion（BIC）。贝叶斯决策理论是主观贝叶斯派归纳理论的重要组成部分。是在不完全情报下，对部分未知的状态用主观概率估计，然后用贝叶斯公式对发生概率进行修正，最后再利用期望值和修正概率做出最优决策。

与AIC相似，训练模型时，增加参数数量，也就是增加模型复杂度，会增大似然函数，但是也会导致过拟合现象，针对该问题，AIC和BIC均引入了与模型参数个数相关的惩罚项，BIC的惩罚项比AIC的大，考虑了样本数量，样本数量过多时，可有效防止模型精度过高造成的模型复杂度过高。

AIC和BIC的原理是不同的，AIC是从预测角度，选择一个好的模型用来预测，BIC是从拟合角度，选择一个对现有数据拟合最好的模型，从贝叶斯因子的解释来讲，就是边际似然最大的那个模型。

共性

构造这些统计量所遵循的统计思想是一致的，就是在考虑拟合残差的同时，依自变量个数施加“惩罚”。

不同点

BIC的惩罚项比AIC大，考虑了样本个数，样本数量多，可以防止模型精度过高造成的模型复杂度过高。

AIC和BIC前半部分是一样的，BIC考虑了样本数量，样本数量过多时，可有效防止模型精度过高造成的模型复杂度过高。

今天我们所观察到的基因组遗传变异是一系列复杂演化过程的产物，不仅受突变、随机漂变、选择（自然选择、人工选择）、重组等的影响，也与参考基因组的组装质量及遗传变异的质量有关。在诸多的影响因素下，为了能够更精确的完成群体之间的溯祖，我们需要结合多方面的分析结果来确认一个最稳健的进化模型，包括群体遗传分析（系统发育树分析，主成分分析，聚类分析，连锁不平衡分析及群体遗传参数统计等）、不依赖先验模型的分析（PSMC、SMC++和 MSMC）及模拟分析（ms、Ma CS）等。

## 基因组渗透：

适应性基因渗透包含两个过程：基因渗透和适应。基因渗透是指遗传成分从一个群体通过有

性生殖的方式进入到另一个群体。产生杂交的前提是两个群体之间没有生殖隔离或只有不完全生殖隔离。只有这样才能通过与可育Ｆ１代回交达到遗传成份垂直传递的目的。通过基因渗透得到的区域，如果在该群体内没有受到选择，那么会因为随机遗传漂变的影响，逐渐在基因组中消失。只有当基因渗透得到的区域受到自然或人工选择，它才能在基因组中保留一定的频率。

## 变异检测：

一般来说我们使用GATK HaplotypeCaller模块进行变异检测。

由于古 DNA 中存在的 5’端 C 到 T 和 3’端 G到 A 的损伤模式会造成大量假阳性 SNP，因此只保留在现代样本中能够检测到的变异，排除了古代样本中的特有变异。由于古代样本中内源性 DNA 含量通常较低，最终得到的测序深度也很低 ，因此针对古代样本还增加了--minPruning 1 和--minDanglingBranchLength 1 的设置以提高低深度位点变异检测的灵敏度。

而变异检测还可以使用bcftools和ANGSD等软件。

若进行的是那种top期刊的研究，推荐使用两种变异检测软件进行变异检测以降低检测到的SNP的假阳性。

性别鉴定，在动物中可以由性染色体的测序深度来确定，由于雄性为XY，雌性为XX，因此雌性的性染色体测序深度应为雄性的二倍。分别统计每个样本在常染色体、X 染色体和 Y 染色体上 SNP 位点的平均 reads 覆盖深度，以常染色体的深度为分母，分别计算了 X 和 Y 染色体的相对覆盖深度。预期观察到雄性个体中的 X 染色体相对覆盖深度为 0.5 且 Y 染色体也为 0.5，而雌性个体中 X 染色体相对覆盖率为 1 且Y 染色体为 0。

## 前期处理

### 数据

/home/sll/2022-DNA-cattle-graduate/20220906-cattle-hebing-vcf-chr1-30-all-total

20220907\_cattle\_204\_hebing\_Chr1\_30\_genotype.vcf.gz

### 提取SNP位点并进行硬过滤以及去多等位基因、染色体

**## 1.提取SNPs位点**

**nohup /home/software/gatk-4.1.4.0/gatk SelectVariants --select-type SNP -V /home/sll/2022-DNA-cattle-graduate/20220906-cattle-hebing-vcf-chr1-30-all-total/20220907\_cattle\_204\_hebing\_Chr1\_30\_genotype.vcf.gz -O 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.vcf.gz &**

**## 2. SNPs位点过滤--还是挺快的**

**nohup /home/software/gatk-4.1.4.0/gatk VariantFiltration -V 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.vcf.gz --filter-expression "QD < 2.0 || FS > 60.0 || MQ < 40.0 || SOR > 3.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0" --filter-name "SNP\_FILTER" -O** **20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.filter.vcf.gz &**

**bcftools index -t 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.filter.vcf.gz**

**##3. 提取PASS的位点**

**nohup /home/software/gatk-4.1.4.0/gatk SelectVariants -V 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.filter.vcf.gz -O 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.filter.pass.vcf.gz -select "vc.isNotFiltered()" &**

**## 4.删除多等位基因位点**

**bcftools view -m 2 -M 2 --type "snps" 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.filter.pass.vcf.gz -Ov -o 20220907\_cattle\_204\_hebing\_Chr1\_30\_snps.genotype.filter.pass.2allele.vcf**

### 过滤（--indep-pairwise 50 25 0.2, --geno 0.05 , --maf 0.03）

解压：gzip -d -c

#### 转换染色体号，去除多余染色体

23773

nohup perl /home/sll/replace\_chr/replace.pl cattle\_204\_hebing\_Chr1\_30\_genotype.vcf /home/sll/replace\_chr/NC-chr1.txt cattle\_204\_hebing\_Chr1\_29\_genotype.nchr.vcf &

转格式

plink --allow-extra-chr --chr-set 29 -vcf cattle\_204\_hebing\_Chr1\_29\_genotype.nchr.vcf --make-bed --double-id --out cattle\_204\_hebing\_Chr1\_29\_genotype.nchr

删除旧的vcf文件cattle\_204\_hebing\_Chr1\_30\_genotype.vcf

#### 去除SNP位点缺失率 > 0.05 ,最小等位基因频率 < 0.03

nohup plink --allow-extra-chr --chr-set 29 -bfile cattle\_204\_hebing\_Chr1\_29\_genotype.nchr --geno 0.05 --maf 0.03 --make-bed --out QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003 &

#### 连锁不平衡过滤，50kb窗口，25kb步长，R2阈值为0.2

注意：进行LD过滤之前先将map文件的第二列填充，否则，LD不会过滤SNP

#转为map、ped格式

plink --allow-extra-chr --chr-set 29 -bfile QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003 --recode --out QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003

#填充map文件第二列，

nohup awk -F '\t' '{print $1"\t"$1\_$4"\t"$3"\t"$4}' QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003.map > QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003.correction.map &

#删除原来的map文件, 将生成的map文件改为原来的map文件名称

rm QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003.map

mv QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003.correction.map QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003.map

#转为二进制格式

nohup plink --allow-extra-chr --chr-set 29 --file QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003 --make-bed --out QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003 &

#进行LD过滤

nohup plink --allow-extra-chr --chr-set 29 -bfile QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003 --indep-pairwise 50 25 0.2 --out QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502 &

nohup plink --allow-extra-chr --chr-set 29 -bfile QC.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003 --extract QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502.prune.in --make-bed --out QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502 &

## 计算群特异性位点

#### 1、提取

bcftools提取和牛与其他群体vcf文件(大的vcf文件用的是未过滤maf 和geno的)，然后用Maf和geno过滤后的SNP即这个群体获得的SNP数。

#### 2、取特异性位点

再把bim文件进行sort a.txt b.txt b.txt | uniq -u操作得到a群体特异性位点。

sort QC.01.10.JPBC-geno005-maf003.bim QC.01.10.sample115-geno005-maf003.bim QC.01.10.sample115-geno005-maf003.bim | uniq -u > JPBC.only.txt

uniq参数说明：

-d 仅显示重复出现的行列;

-u 仅显示出一次的行列。

## 群体结构

### PCA

纯数学的运算方法，可以将多个相关变量经过线形转换选出较少个数的重要变量。

#### GCTA软件

##### ## make germ

/home/software/gcta\_1.92.3beta3/gcta64 --bfile QC.common\_89\_cattle\_851\_ASIA-geno005-maf003 --make-grm --autosome-num 29 --out QC.common\_89\_cattle\_851\_ASIA-geno005-maf003.gcta

##### ## PCA

/home/software/gcta\_1.92.3beta3/gcta64 --grm QC.common\_89\_cattle\_851\_ASIA-geno005-maf003.gcta --pca 4 --out QC.common\_89\_cattle\_851\_ASIA-geno005-maf003.gcta.out

##### PCA.sh(233个体)

touch PCA.sh

nohup bash PCA.sh &

## 提取：

bcftools view -S id.txt common\_89\_cattle\_851\_ASIA.vcf -Ov > sample-select.vcf

## 转格式：

plink --allow-extra-chr --chr-set 29 -vcf sample-select.vcf --make-bed --double-id --out sample-select

## 过滤：

plink --allow-extra-chr --chr-set 29 -bfile sample-select --geno 0.05 --maf 0.03 --make-bed --out QC.sample-select-geno005-maf003

## pca

## make germ

/home/software/gcta\_1.92.3beta3/gcta64 --bfile QC.sample-select-geno005-maf003 --make-grm --autosome-num 29 --out QC.sample-select-geno005-maf003.gcta

## PCA

/home/software/gcta\_1.92.3beta3/gcta64 --grm QC.sample-select-geno005-maf003.gcta --pca 4 --out QC.sample-select-geno005-maf003.gcta.out

#### EIGENSOFT软件

##### 1、转换格式：

使用EIGENSOFT中内置的convertf 文件转化为smartpca的输入文件

convertf -p transfer.conf

需要一个config file，将文件的输入输出写进去。

##config file

genotypename: myplink\_test.ped

snpname: myplink\_test.map # or example.map, either works

indivname: myplink\_test.ped # or example.ped, either works

outputformat: EIGENSTRAT

genotypeoutname: example.eigenstratgeno

snpoutname: example.snp

indivoutname: example.ind

familynames: NO

产生三个pca所需的输入文件 example.eigenstrat example.snp example.ind

##### 2、smartpca做PCA

smartpca -p runningpca.conf

##config file

genotypename: example.geno

snpname: example.snp

indivname: example.ind

evecoutname: example.pca.evec

evaloutname: example.eval

altnormstyle: NO

numoutevec: 10

numoutlieriter: 5

outliersigmathresh: 6.0

### Admixture

群体结构分析的算法模型为：假设所有群体拥有K个祖先，每个群体的特征由其每个位点的等位基因频率决定，在哈代-温伯格平衡下，使用最大似然估计算法将实际群体中的每个个体以概率的形式，计算每个个体的基因组变异来源，用Q值表示，Q值越大，表明该位点来自这个祖先群体的可能性越大，从而将每个位点归类到不同的祖先成分。

#### # geno和maf、LD过滤

#### # admixture

for K in 2 3 4 5 6 7 8 9 10 11 12 13 14 15; do /home/software/admixture\_linux-1.3.0/admixture --cv QC.sample-northeast\_Asia-geno005-maf003.bed $K | tee log${K}.out; done

#### # 提取CV值

CV error最小的为最佳K值

grep -h CV log\*.out

#### Admixture.sh

touch admixture.sh

nohup bash admixture,sh &

## 提取：

bcftools view -S id.txt common\_89\_cattle\_851\_ASIA.vcf -Ov > sample-select.vcf

## 转格式：

plink --allow-extra-chr --chr-set 29 -vcf sample-select.vcf --make-bed --double-id --out sample-select

## 过滤：

plink --allow-extra-chr --chr-set 29 -bfile sample-select --geno 0.05 --maf 0.03 --make-bed --out QC.sample-select-geno005-maf003

## admixture

for K in 2 3 4 5 6 7 8 9 10 11 12 13 14 15; do admixture --cv QC.sample-select-geno005-maf003.bed $K | tee log${K}.out; done

## 提取CV值

## CV error最小的为最佳K值

grep -h CV log\*.out

### Treemix（加了外群）:

TreeMix基于全基因组多态性模拟遗传漂变来推断种群之间的关系。首先估计样本数量之间关系的系统树图，然后来比较系统树模拟构建与观察到的群体之间的变异结构。当群体比通过分杈树建模表现出更密切的关系，则说明群体之间在历史上有过杂合过程。TreeMix在系统法语中增加边线使之成为一个系统网络，这些边缘的位置和方向都是有信息的；如果一个边缘产生更多的基底系统网络这表明杂交发生事件比较早或者来源于另一个分化的群体。

#### 1.计算等位基因频率

# 转换为tped格式，生成sample.tped和sample.tfam文件。

vcftools --vcf QC.sample-select-geno005-maf003.vcf --plink-tped --out sample

# sample.tfam修改第一列数据为breed ID。

# 提取文件sample.pop.cov，格式为：共三列，前两列与修改后的sample.tfam前两列一样，为群体ID和样本ID，第三列和第一列一致，tab分隔。

cat sample.tfam |awk '{print $1"\t"$2"\t"$1}' >sample.pop.cov

# 计算等位基因组的频率，生成plink.frq.strat和plink.nosex文件

plink --threads 12 --tfile sample --freq --allow-extra-chr --chr-set 29 --within sample.pop.cov

#压缩等位基因频率文件

gzip plink.frq.strat

#转换格式【耗时小时计】

#用treemix自带脚本进行格式转换，notes：输入输出都为压缩文件，plink2treemix.py使用python2并需要绝对路径（否则报错）。

python2 /home/sll/miniconda3/bin/plink2treemix.py plink.frq.strat.gz sample.treemix.in.gz

#### 2.treemix推断基因流

# 多次分析以评估最佳m值

比如m取1-10(常用1-5,1-10)，每个m值重复5次(至少两次)

for m in {1..5}

do

{

for i in {1..5}

do

{

treemix -i sample.treemix.in.gz -o sample.${m}.${i} -bootstrap 100 -root wBGU -m ${m} -k 500 -noss

}

done

}

done

-i 指定基因频率输入文件

-o 指定输出文件前缀

-tf 【可选】指定树文件，指定后就使用指定树的拓扑结构(最好把支持率和其他无关信息都删除，只留最简单的Newick格式)，否则treemix会推断拓扑

-root指定外类群(指定的是居群名称)，多个用逗号分隔；最好指定，否则后面plot\_tree画树没找到更换外类群的参数会很麻烦

-m 为the number of migration edges即基因渗入的次数

-k 1000 因为SNP之间不是独立位点，为了避免连锁不平衡，用k参数指定SNP数量有连锁，比如这里指定用1000个SNP组成的blocks评估协方差矩阵

-se 计算迁移权重的标准误差(计算成本高)，如果想省时间可以不用这个参数

-bootstrap 为了判断给定树拓扑的可信度，在blocks运行bootstrap重复

-global 在增加所有种群后做一轮全局重组。

-noss 关闭样本量校正。TreeMix计算协方差会考虑每个种群的样本量，有些情况(如果有种群的样本只有1个)会过度校正，可以关闭。

-g old.vertices.gz old.edges.gz #使用之前生成的树和图结果，用-g指定之前的两个结果文件

-cor\_mig known\_events and -climb #合并已知的迁移事件

nohup bash treemix.sh &

##### 2.2.最佳迁移边缘个数选择

将生成的llik、cov、modelcov文件放置在同一文件夹，使用R包OptM分析：

library(OptM)

linear = optM("./data") ##文件夹名

plot\_optM(linear)

生成图中，当Δm值最小时的migration edges为最佳迁移边缘个数

##### 2.3.基因渗入作图

使用m=最佳迁移边缘个数结果文件做图

source("plotting\_funcs.R") #treemix scr文件夹中R脚本

plot\_tree("TreeMix") #TreeMix为结果文件前缀

##=====5.2 绘制混合树====

prefix="sample.3" ## treemix产生的结果文件前缀

library(RColorBrewer)

library(R.utils)

par(mfrow=c(2,3))

for(edge in 1:3){

plot\_tree(cex=0.8,paste0(prefix,".",edge))

title(paste(edge,"repetition"))

} # 放的是m012345，则0:5

#### treemix系统发育分析（ML）

群体间的系统发育树，每个分支代表一个群体。

treemix -i sample.treemix.in.gz -o sample.ML.tree -k 1000 -global

结果文件sample.treemix.treeout.gz解压后可用ITOL进行可视化

注意：若使用NJ法构建进化树，则不能过滤LD，而使用ML法构建进化树时则可过滤LD！！！

### MEGA(进化树)

##### #转格式

plink --allow-extra-chr --chr-set 29 -vcf QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502.vcf --make-bed --double-id --out QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502

##### #计算genome

plink --bfile QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502 --allow-extra-chr --chr-set 29 --genome

##### #计算MEGA

PLINK\_genome\_MEGA.pl

第一个open里改成自己的genome文件

第二个open里改成自己的fam文件

第三个open里将>后面的改成输出的文件名.meg

在下面的sample\_size里，将数量改成自己使用的数量

#!usr/bin/perl

# define array of input and output files

open (AAA,"plink.genome") || die "can't open AAA";

open (BBB," QC.ld.cattle\_204\_hebing\_Chr1\_29\_genotype.nchr-geno005-maf003-502502.fam") || die "can't open BBB";

open (CCC,"> cattle\_204\_hebing\_Chr1\_29.meg");

my @aa=<AAA>;

my @bb=<BBB>;

$sample\_size=204; ### 涓綋鏁扮洰

print CCC "#mega\n!Title: $sample\_size pigs;\n!Format DataType=Distance DataFormat=UpperRight NTaxa=$sample\_size;\n\n";

foreach ($num1=0;$num1<=$#bb;$num1++){

chomp $bb[$num1];

@arraynum1=split(/\s+/,$bb[$num1]);

print CCC "#$arraynum1[1]\n"; ##涓綋鐨処D鍚嶇О

}

print CCC "\n";

@array=();

foreach ($num2=1;$num2<=$#aa;$num2++){

chomp $aa[$num2];

@arraynum1=split(/\s+/,$aa[$num2]);

push(@array,1-$arraynum1[12]);

}

@array2=(0);

$i=$sample\_size;

while ($i>0){

push(@array2,$array2[$#array2]+$i);

$i=$i-1;

}

print "@array2";

for ($i=($sample\_size-1); $i>=0; $i=$i-1){

print CCC " " x ($sample\_size-($i+1));

for ($j=$array2[$sample\_size-$i-1]; $j<=$array2[$sample\_size-$i]-1; $j++){

print CCC "$array[$j] ";

}

print CCC "\n";

}

close AAA;

close BBB;

close CCC;

### RAxML(进化树):

##### 1、转为phy格式：

python /home/sll/software/vcf2phylip.py --input FASN.vcf.recode.vcf

##### 2、建树：

raxmlHPC-PTHREADS-SSE3 -f a -m GTRGAMMA -p 12345 -x 12345 -# 100 -s FASN.min4.phy -n raxml -T 30

-f a此参数用于选择 RAxML 运算的算法。可以设定的值非常之多。 a 表示执行快速 Bootstrap 分析并搜索最佳得分的 ML 树。

-x 12345指定一个 int 数作为随机种子，以启用快速 Bootstrap 算法。

-p 12345指定一个随机数作为 parsimony inferences 的种子。

-# 100指定 bootstrap 的次数。

-m PROTGAMMALGX 指定核苷酸或氨基酸替代模型。PROTGAMMALGX 的解释： "PROT" 表示氨基酸替代模型； GAMMA 表示使用 GAMMA 模型； X 表示使用最大似然法估计碱基频率。

-s ex.phy指定输入文件。phy 格式的多序列比对结果。软件包中包含一个程序来将 fasta 格式转换为 phy 格式。

-n ex输出文件的后缀为 .ex 。

-T 20指定多线程运行的 CPUs 。

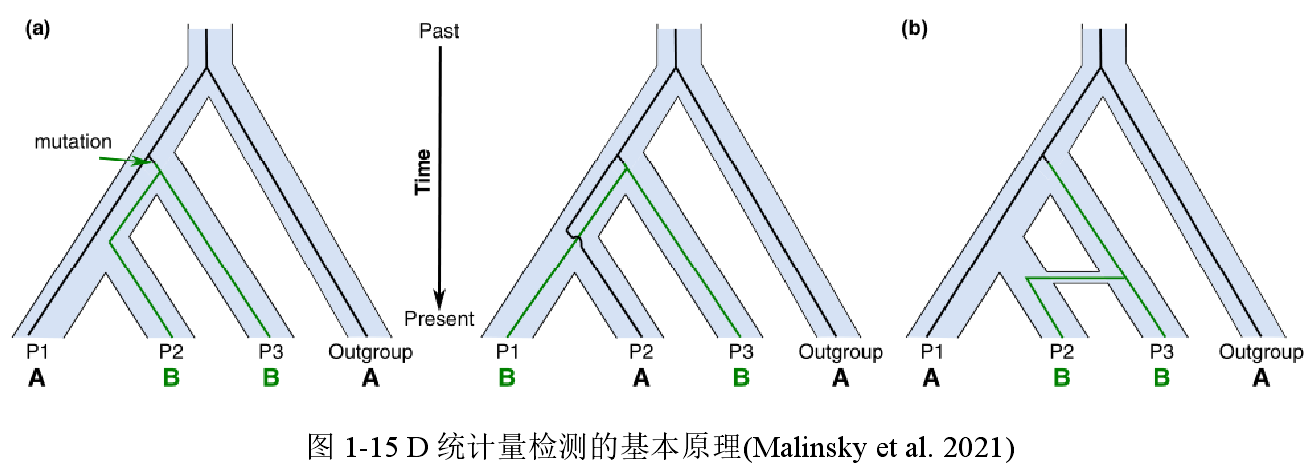
## 基因流、群体历史、选择信号

### f3、f4检验以及D检验

D 统计量（Patterson’s D statistic）是目前最广为使用的渗入检测方法之一。D 统计量需要四个群体，这里称之为 P1、P2、P3 和 O。他们的系统发育关系如图 1-15 所示，为(((P1,P2)P3,)O)。其中，P1 和 P2 为姐妹群，O 为外群。对于这四个群体中存在的双等位 SNP，以外群O 中的类型作为祖先型 A（即 ancestral allele），另一种则为衍生型 B（即 derived allele）。在已知的系统发育关系的情况下，基因组上绝大多数 SNP 在这四个群体中的排布模式应该为 BBAA，即 P1 和 P2 同为衍生型等位基因，而 P3 和 O 同为祖先型的

等位基因。BBAA 的模式是符合系统发育关系的，但如果 P2 和 P3 之间发生了基因流，

那么则会产生大量 ABBA 模式的位点，即 P2 和 P3 共享了同一种等位基因。反之，如

果 P1 和 P3 之间发生了渗入，则会产生大量 BABA 模式的位点。

但实际上，由于 P1、P2 和 P3 的共同祖先在某些位点上可能是存在多态性的，这

种多态性可能会随机分配给这三个群体，从而导致 ABBA 或者 BABA 的情况。这种

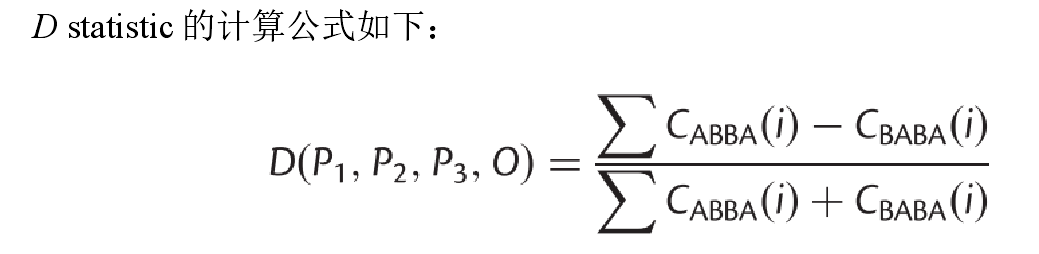
现象通常被称为不完全谱系分选（incomplete lineage sorting，ILS）。因此单独去检测

ABBA 或者 BABA 模式的位点数量无法判断它们是由于 ILS 还是渗入导致的。但由于

ILS 是不受选择的，所以它产生的 ABBA 和 BABA 模式的位点数量应该是大致相当的。

所以我们可以通过比较 ABBA 和 BABA 的数量是否有显著的差异来判断是单纯的 ILS

还是发生了渗入。



其中，CABBA表示 ABBA 模式位点的数量。D 统计为符合 ABBA 模式的位点数量与 BABA 模式的位点数量之差，除以 ABBA 模式和 BABA 模式位点数量之和。故 D统计量为正时，ABBA 的数量更多，因此可能是 P2 和 P3 之间发生了渗入。反之，D统计量为负时，BABA 数量更多，可能是 P1 和 P3 之间发生了渗入。由于 D 统计量的计算是基于对 ABBA 和 BABA 模式位点的计数，因此它也被称之为 ABBA-BABA test。需要注意的是，D statistic 中四个群体的排列顺序以及正负值代表的含义在不同的软件中可能并不相同，如 ADMIXTOOLS中D为正值时可能是 P1 和 P3 之间发生了渗入、ANGSD doAbbababa2和 Dsuite

**不完全谱系分选**(ILS,Incomplete Lineage sorting):位点树（基因树）和物种树不一致的现象。例如ABC物种分化前某一位点存在多态性，为0/1。随着C分化出去，而此多态性的位点在C中发生了固定，为1，而在AB祖先中是以多态存在。当AB发生分化时，此等位以随机的方式分别进入AB物种中。当B的位点和C相同时，即发生BC关系更近的现象。我们以为是BC之间存在基因流的现象，而实际上为不完全谱系分选。

外群在 D statistic 中的作用主要是为了判断祖先型等位基因，但是对于 D statistic来说，判断哪个是祖先型，哪个是衍生型，其实并不重要，因为 ABBA 和 BAAB 是等效的。所以也就有研究人员尝试只使用 3 个群体判断渗入，其统计量称为 D3

AdmixTools需要特征（eigenstrat）文件，要将vcf文件转换为eigenstrat。

bash convertVCFtoEigenstrat.sh QC.sample-select-geno005-maf003.vcf

#### convertVCFtoEigenstrat.sh

#!/bin/bash

# Script to convert vcf to eigenstrat format for ADMIXTOOLS

# Written by Joana Meier

# It takes a single argument: the vcf file (can be gzipped) and

# optionally you can specify --renameScaff if you have scaffold names (not chr1, chr2...)

# Here, you can change the recombination rate which is currently set to 2 cM/Mb

rec=2

# It requires vcftools and admixtools

# for some clusters, it is needed to load these modules:

# module load gcc/4.8.2 vcftools openblas/0.2.13\_seq perl/5.18.4 admixtools

renameScaff="FALSE"

# If help is requested

if [[ $1 == "-h" ]]

then

echo "Please provide the vcf file to parse, and optionally add --renameScaff if you have scaffolds instead of chromosomes"

echo "Usage: convertVCFtoEigenstrat.sh <vcf file> --renameScaff (note, the second argument is optional)"

exit 1

# If the second argument renameScaff is given, set it to True

elif [[ $2 == "--renameScaff" ]]

then

renameScaff="TRUE"

# If no argument is given or the second one is not -removeChr, give information and quit

elif [ $# -ne 1 ]

then

echo "Please provide the vcf file to parse, and optionally add --renameScaff if you have scaffolds instead of chromosomes"

echo "Usage: ./convertVCFtoEigenstrat.sh <vcf file> --renameScaff (note, the second argument is optional)"

exit 1

fi

# Set the first argument to the file name

file=$1

file=${file%.gz}

file=${file%.vcf}

# if the vcf file is gzipped:

if [ -s $file.vcf.gz ]

then

# If renaming of scaffolds is requested, set all chromosome/scaffold names to 1

if [ $renameScaff == "TRUE" ]

then

echo "setting scaffold names to 1 and positions to additive numbers"

zcat $file".vcf.gz" | awk 'BEGIN {OFS = "\t";add=0;lastPos=0;scaff=""}{

if($1!~/^#/){

if($1!=scaff){add=lastPos;scaff=$1}

$1="1"

$2=$2+add

lastPos=$2

}

print $0}' | gzip > $file.renamedScaff.vcf.gz

# Get a .map and .ped file (remove multiallelic SNPs, monomorphic sites and indels)

vcftools --gzvcf $file".renamedScaff.vcf.gz" \

--plink --mac 1.0 --remove-indels --max-alleles 2 --out $file

else

# Get a .map and .ped file (remove multiallelic SNPs, monomorphic sites and indels)

vcftools --gzvcf $file".vcf.gz" \

--plink --mac 1.0 --remove-indels --max-alleles 2 --out $file

fi

# if the file is not gzipped

else

# If renaming of scaffolds is requested, set all chromosome/scaffold names to 1

if [ $renameScaff == "TRUE" ]

then

echo "setting scaffold names to 1 and positions to additive numbers"

awk 'BEGIN {OFS = "\t";add=0;lastPos=0;scaff=""}{

if($1!~/^#/){

if($1!=scaff){add=lastPos;scaff=$1}

$1="1"

$2=$2+add

lastPos=$2

}

print $0}' $file.vcf | gzip > $file.renamedScaff.vcf.gz

# Get a .map and .ped file (remove multiallelic SNPs, monomorphic sites and indels)

vcftools --gzvcf $file".renamedScaff.vcf.gz" \

--plink --mac 1.0 --remove-indels --max-alleles 2 --out $file

else

vcftools --vcf $file".vcf" --plink --mac 1.0 --remove-indels --max-alleles 2 --out $file

fi

fi

# Change the .map file to match the requirements of ADMIXTOOLS by adding fake Morgan positions (assuming a recombination rate of 2 cM/Mbp)

awk -F"\t" -v rec=$rec 'BEGIN{scaff="";add=0}{

split($2,newScaff,":")

if(!match(newScaff[1],scaff)){

scaff=newScaff[1]

add=lastPos

}

pos=add+$4

count+=0.00000001\*rec\*(pos-lastPos)

print newScaff[1]"\t"$2"\t"count"\t"pos

lastPos=pos

}' ${file}.map | sed 's/^chr//' > better.map

mv better.map ${file}.map

# Change the .ped file to match the ADMIXTOOLS requirements

awk 'BEGIN{ind=1}{printf ind"\t"$2"\t0\t0\t0\t1\t";

for(i=7;i<=NF;++i) printf $i"\t";ind++;printf "\n"}' ${file}.ped > tmp.ped

mv tmp.ped ${file}.ped

# create an inputfile for convertf

echo "genotypename: ${file}.ped" > par.PED.EIGENSTRAT.${file}

echo "snpname: ${file}.map" >> par.PED.EIGENSTRAT.${file}

echo "indivname: ${file}.ped" >> par.PED.EIGENSTRAT.${file}

echo "outputformat: EIGENSTRAT" >> par.PED.EIGENSTRAT.${file}

echo "genotypeoutname: ${file}.eigenstratgeno" >> par.PED.EIGENSTRAT.${file}

echo "snpoutname: ${file}.snp" >> par.PED.EIGENSTRAT.${file}

echo "indivoutname: ${file}.ind" >> par.PED.EIGENSTRAT.${file}

echo "familynames: NO" >> par.PED.EIGENSTRAT.${file}

# Use CONVERTF to parse PED to eigenstrat

convertf -p par.PED.EIGENSTRAT.${file}

# change the snp file for ADMIXTOOLS:

awk 'BEGIN{i=0}{i=i+1; print $1"\t"$2"\t"$3"\t"i"\t"$5"\t"$6}' $file.snp > $file.snp.tmp

mv $file.snp.tmp $file.snp

#### f3统计量及Outgroup-f3

f3统计又称admixture检验，常用于判断一个群体（或其祖先）是否来自两个群体的混和。由于混合后的群体存在遗传漂变，随着时间的推移其混合的分子学特征会逐渐减弱，因此f3检验比较适合近期的基因流信号检测。

目前能做f3检验的只有ADMIXTOOLS软件包中的*qp3Pop*和TreeMix中的*threepop*, 且需要精确的基因型信息。

##### AdmixTools中的qp3Pop

1、修改.ind文件的第三列为品种id

2、提供A、B、C群体的pop文件，3列（outgroup f3时，C为外群，判断A和B的关系）

3、修改脚本文件par.PED.EIGENSTRAT.QC.sample-select-out-geno005-maf003，里的东西为：

genotypename: QC.sample-select-out-geno005-maf003.eigenstratgeno

snpname: QC.sample-select-out-geno005-maf003.snp

indivname: QC.sample-select-out-geno005-maf003.ind

popfilename: pop.txt

inbreed: YES ##(做outgroup f3应删除这一行，目标群体存在近交，则加上这行)

4、qp3Pop分析

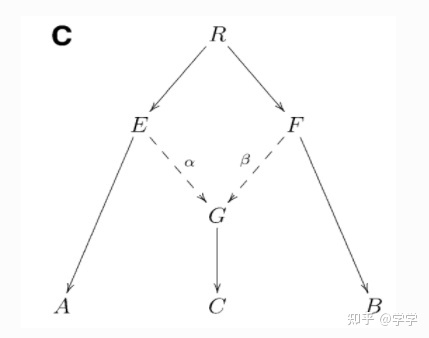
qp3Pop -p par.PED.EIGENSTRAT.QC.sample-select-out-geno005-maf003 > 3pop\_qp3pop

##### Treemix中的threepop

threepop -i sample.treemix.in.gz -k 500 > 3pop

cat 3pop |grep -v Estimating |grep -v nsnp|tr ';' ' ' > 3pop.txt

z < -3为显著

一个物种与另外两个物种之间的等位基因频率的相关性

F3中的3其实表示了三个物种之间的基因流

如果一个物种（C）与另外两个物种（A，B）之间的等位基因频率相关性越高，则说明该物种C可能混合了A，B的基因。

A, B为参考群体，C为测试群体

统计量F3检验的内容有：

（i）C是否混合A，B的遗传变异；

（ii）A，B是否共同获得了C的遗传漂变。

如果C来自于A，B的混合，C的等位基因频率c应该趋向于a和b之间，所以，如果F3算出来的值是一个负值的话（同时满足统计学检验的标准，Z-scores=F3/SE(F3)要小于-3），认为F3具有显著的统计学效应，即C中的一些基因来自于A和B的混合。

值得注意的是，上面反过来不成立，F3大于0，并不认为C中不含有A和B的混合

如果令C物种为外来物种的话，那么C其实无论如何都不可能是AB的混合，而F3值越大，可以用来表征A,B之间的相似性越强(outgroup f3)

#### f4统计量

判断A,B与C,D之间是否存在基因流

若B,C之间发生基因流，B,C等位基因频率趋同，介于A,D之间，f4<0

若A,D之间发生基因流，A,D等位基因频率趋同，介于B,C之间，f4<0

A,C或B,D之间发生基因流，f4>0

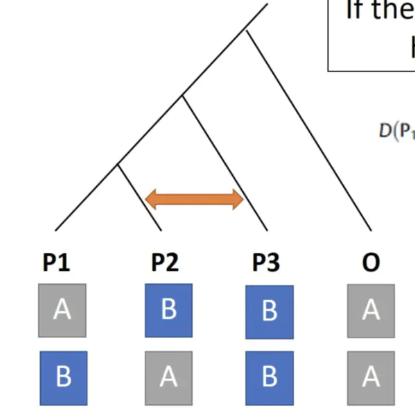
将A当作外来物种，那么我们就可以直接通过F4的正负来判断基因流了。若F4>0，则B与D之间有基因流；若F4<0，则B与C之间有基因流

##### Treemix中的fourpop

treemix -i sample.treemix.in.gz -k 500 > 4pop

cat 4pop |grep -v Estimatin|grep -v nsnp |tr ',' ' '|tr ';' ' ' > 4pop.txt

#### D-statistics

 也叫ABBA-BABA检验，用于检测四个群体是否存在由“Admixture”造成的显著不符合拓扑树结构事件的方法，在正确的拓扑结构下，又可以判断其中两个群体是否存在基因交流，假定系统发育拓扑结构（（（W,X）;Y）,Outgroup），D统计通过对符合ABBA，BABA，BBAA，BBBA模式的位点数目进行计数，并根据计数结果进行基因流检测，其中A表示祖先型等位基因，B表示衍生型等位基因。

正常情况下，由于拓扑结构的固定，BBBA于BBAA模式的位点应是最常见的。而ABBA与BABA模式的位点不支持四个群体的拓扑结构，从概率上讲两种模式的位点数应该一致。D统计通过计算ABBA与BABA模式位点数的差值与两种模式位点数的和的比值判断X与Y或W之间是否存在基因流。使用Z检验作为显著性检验算法（（（W,X）;Y）,Outgroup）

X,Y是姊妹类群(多为一个物种内的两个种群)，W是潜在的基因渐渗来源物种，Outgroup是外类群。

4 populations (W, X, Y, Z)

num = (w − x)(y − z )

den = (w + x − 2wx)(y + z − 2yz )

D = num/ den （这个是官网上给的公式）

Z为外群的情况下，D > 0，W与Y之间存在基因流

D < 0，X与Y之间存在基因流

##### AdmixTools中的qpDstat

1、修改.ind文件的第三列为品种id

2、提供A、B、C、D群体的pop文件，4列

3、修改脚本文件par.PED.EIGENSTRAT.QC.sample-select-out-geno005-maf003，里的东西为：

genotypename: QC.sample-select-out-geno005-maf003.eigenstratgeno

snpname: QC.sample-select-out-geno005-maf003.snp

indivname: QC.sample-select-out-geno005-maf003.ind

popfilename: pop.txt

f4mode: YES ##此选项为进行f4检验，默认是NO

4、qpDstat分析

qpDstat -p par.PED.EIGENSTRAT.QC.sample-select-out-geno005-maf003 > 4pop\_admixtools

D统计量=0，说明总体ABBA和BABA的数量相同，不存在明显的渗入；如果不等于0，则或许存在渗入。

Z=D/ standard\_error

Z>3和<-3分别是正负显著

也就是用于判断X,Y与W之间是否发生基因流



### 渗入片段的检测（单倍型）

基因渗入指两个物种之间通过杂交方式进行遗传物质的交换，一个群体的遗传物质通过杂交转移至另一个群体。基因流的研究证明它是影响群体演化的一个重要因素。一个群体可以通过基因交流快速获得新的等位基因，增加群体的适应性变异。

基因组上的哪些片段时通过基因交流获得的以及它们的生物学功能

与HAPMIX, LAM-LD, RFmix需要设置多个参数，HAPMIX需要提供遗传图谱、充足率、突变率和平均代时等较难获得的生物学参数。Loter除了单倍型数据以外不需要其他任何生物学参数即可完成祖先血统推断。与HAPMIX, LAM-LD, RFmix软件相比，在考虑人工模拟和人工混群的情况下，随着基因渗入次数的增加，Loter软件受到的影响较小。

提供足够的祖先物种的基因组数据来作为来源群体（source populations），同时也需要发生渗入样本的基因组数据。对于每一个渗入个体的每一个变异位点，loter均会进行祖先来源判断，从所提供的来源群体中选择一个可能性最大的群体作为其来源。

需要提供一个几乎没有发生渗入或者渗入比例极低的群体作为对照。

##### rIBD分析--IBD(beagle4.0版本，赖伟宁师姐找到的脚本，快跑到她面前夸她)

血缘同源（IBD）是指两个或两个以上的等位基因单倍型遗传自同意祖先且未发生重组事件。

一般为50Kb窗口，25kb步长，计算每个区段中两群体间的共享IBD单倍型数量。

而由于群体间两两比较的总数不同，故对共享IBD单倍型进行标准化，获得nIBD值，标准化后的nIBD值分布在0（未检测到共享IBD）——1

nIBD=cIBD/tIBD

cIBD表示两群体间所有共享IBD单倍型数量，tIBD表示两群体所有单倍型数量。为了能直接体现不同群体间共享IBD单倍型的差异，可以计算rIBD

rIBD= nIBDgroup1- nIBDgroup2

nIBDgroup1是指某一群体与群体1的共享IBD，nIBDgroup2是指某一群体与群体2的共享IBD。

java -jar -Xmn12G -Xms24G -Xmx48G /home/sll/software/beagle.r1399.jar ibd=true impute=false window=50000 overlap=25000 gt=out.beagle.vcf.gz out=out.beagle.ibd

结果文件：

1) First sample identifier 第一个样本ID

2) First sample haplotype index (1 or 2) 第一个样本的单倍型

3) Second sample identifier 第二个样本ID

4) Second sample haplotype index (1 or 2) 第二个样本的单倍型

5) Chromosome 染色体

6) Starting genomic position (inclusive) 起始位置

7) Ending genomic position (inclusive) 终止位置

8) LOD score (cM length of IBD segment）值越大越好，IBD段的cM长度

##### Loter(结果看不懂)

1、过滤 （去多等位基因位点，保留最大缺失率（max-missing）<10% 即检出率>90%的SNP位点，（MAF）>0.03（哈代温伯格平衡检验的P值大于10^-6 -hwe）

vcftools --vcf common\_89\_cattle\_851\_ASIA.latter.vcf --max-missing 0.9 --maf 0.03 --remove-indels --min-alleles 2 --max-alleles 2 --recode --recode-INFO-all --out common\_89\_cattle\_851\_ASIA.filter

2、bagle填充并转为单倍型格式

java -jar -Xmn12G -Xms24G -Xmx48G /home/software/beagle.25Nov19.28d.jar gt=common\_89\_cattle\_851\_ASIA.filter.recode.vcf out=out.beagle ne=233

loter\_cli -r ANG.1.test.recode.vcf MSU.1.test.recode.vcf -a BC.1.test.recode.vcf -f vcf -o interval.txt -n 8 -pc -v

-r REF [REF ...], --ref REF [REF ...] files storing input reference haplotypes.可以是多个，空格分开输入

-a ADM, --adm ADM file storing input admixed haplotypes.

-o OUTPUT, --output OUTPUT

file to store the infered ancestries of admixed haplotypes, saved as a numpy array by default, or as a text CSV file if the file extension is '.txt'.

-f 输入文件的格式；有npy(默认), txt（csv）, vcf可选，所有的输入文件格式一致

-n CPU数量

-pc Run the phase correction module after the inference algorithm (only available for data from diploid organisms)

西农一个师姐的，好像用不了

####第一步：分染色体分区域计算渗入区间

####01.loter.sh

source /stor9000/apps/users/NWSUAF/20120109540/.bashrc

export PYTHONPATH="/stor9000/apps/users/NWSUAF/20120109540/.local/lib/python3.8/site-packages:$PYTHONPATH"

export PYTHONPATH="/stor9000/apps/users/NWSUAF/20120109540/Software/Anaconda3\_2020\_11/lib/python3.8/site-packages:$PYTHONPATH"

export LD\_LIBRARY\_PATH="/stor9000/apps/users/NWSUAF/20120109540/Software/Anaconda3\_2020\_11/lib::$LD\_LIBRARY\_PATH"

chr=`cat ./regionfile/region${1} |awk '{print $1}' `

#echo $chr

/stor9000/apps/users/NWSUAF/20120109540/Software/Anaconda3\_2020\_11/bin/python /stor9000/apps/users/NWSUAF/20120109540/Script/bioCal/loter\_multiRegions.py \

/stor9000/apps/users/NWSUAF/2019055262/sheep.Y/14.loter/01.vcf/Sheep278.chr${chr}.imp.phase.vcf.gz \

out\_regions2/regions\_${1} \

--groupfile group.txt \

--regionfile /stor9000/apps/users/NWSUAF/20120109540/fangwenwen/01.loter/02.whole\_genome/run02/regionfile/region${1} \

--refpops EU\_Mou,Mid\_MOU \ #渗入来源群体

--querypops AFR,Americas,EUR,ME,NEC,SouthAsia,YKP \ #被渗入群体

--threads 4

####第二步：整合每个个体的渗入区域

####02.indseg.sh

#!/bin/sh

python /stor9000/apps/users/NWSUAF/20120109540/Script/bioCal/merge\_loter\_multiRegions.py \

./out\_regions2/regions\_${1}.anc.tsv.gz \

./out\_regions2/regions\_${1}.indseg.tsv.gz

##### RFMix

需提供比较完善的参考群体，对目标群体的遗传背景应清楚

1、过滤（去除多等位基因位点）

2、beagle填充

java -jar -Xmn12G -Xms24G -Xmx48G /home/software/beagle.25Nov19.28d.jar gt=tibetan-36.filter-nchr.recode.vcf out=tibetan-36.filter-nchr.recode.vcf.beagle ne=36

3、RFMix推断渗入区域（分染色体推断）

rfmix -f BC.test.recode.vcf -r ANG.test.recode.vcf -m map.txt -g genetic.map -o outer --chromosome=1

-f 目标群体

-r 参考群体(多个群体在一个vcf文件里即可)

-m map文件包含两列；1：参考群体样本ID，2：参考群体品种ID（也是主要设定ID的文件）

-g genetic.map包含三列；1:染色体号，2:物理距离(bp)，3:遗传距离（cM）, 1cM=1Mb(10-6)

--chromosome= 选择需要分析的染色体（和vcf文件一致）

为了获得某一祖先源比例显著高于全基因组水平的片段（定义为过量片段），对所有片段进行卡方检验

应该是看msp.tsv文件，可规定渗入单倍型为多少为渗入区域，用于构建进化树验证渗入事件以及计算渗入时间

验证：可利用渗入区域对其进行进化树分析，若两群体之间存在基因渗入，两个群体内的个体在进化树上会聚集在一起。

方法大致应该是先将渗入区域根据位置提取出来，再对其进行进化树分析（有些文献是选取群体中的某个个体来做的）

并且可以将非渗入区域也进行建树作为对照，通过对比，基因渗入更加直接明了

### 有效群体大小（Ne）-- smc++

与实际群体有相同基因频率方差或相同的杂合度衰减率的理想群体含量***Ne***。理想群体指群体含量在世代间保持恒定、群体内随机交配、无世代重叠、无其他遗传效应影响。

越近的时间点，尤其1千年以内，只有SMC++才有足够精确的解析能力。

SMC++分析的总体思路和MSMC类似，也是可以整合多个样本进行Ne估测。但通过算法的改进，SMC++的优点体现在:

1)可拓展性

运算效率高，可以一次分析多个样本，突破了MSMC一般只能同时分析4个样本的限制。

2)结果更准确

由于整合多个样本的信息，因此结果更加准备，且对近期事件(1万年以内的历史)解析精度大大提高。

3)不需要phasing，基础的重测序数据即可

4)结果以拟合线的形式展示，更顺滑直观;

#### 1、Convert your VCF(s) to the SMC++ input format with vcf2smc:

smc++ vcf2smc my.data.vcf.gz out/chr1.smc.gz chr1 Pop1:S1,S2

### LD衰减 -- PopLDdecay

连锁不平衡是指位于某一座位的特定等位基因与另一座位的某一等位基因同时出现的概率大于群体中因随机分布的两个等位基因同时出现的概率。

LD 的衰减是受重组率和重组代数影响；研究 LD 的衰减可以揭示群体重组的历史，与等位基因的频率相结合，LD 衰减也可以用于检测正向选择

LD的衰减指位点间由连锁不平衡到连锁平衡的演变过程；LD衰减的速度在不同物种间或同物种的不同亚群间，往往差异非常大。所以，通常会使用1个标准——“LD衰减距离”来描述LD衰减速度的快慢。

LD衰减距离通常指的是：当平均LD系数r2 衰减到一定大小的时候，对应的物理距离。“一定大小”是这个定义的关键点，但没有特别统一的标准，在不同文章中标准不同。常见的标准包括：

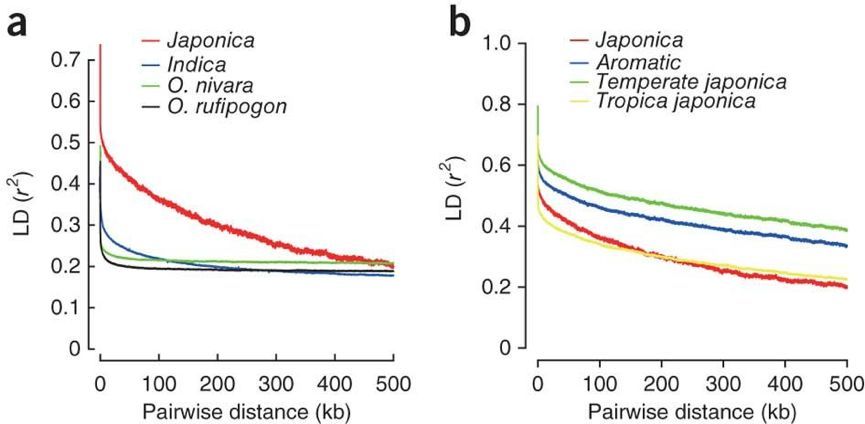
a）LD系数降低到最大值的一半；

b）LD系数降低到0.5以下；

c）LD系数降低到0.1以下；

d）LD系数降低到基线水平（注意，不同物种的基线值是不同的）。

所以，下次你在文章中看到“LDdecay distance is XXkb”的时候，不要忘了看看文章使用的标准是什么。



值的获取：成对计算指定距离范围内的所有SNP的r2 值，按区间取平均。

LD衰减距离的应用

1、判断GWAS所需标记量，决定GWAS的检测效力以及精度；

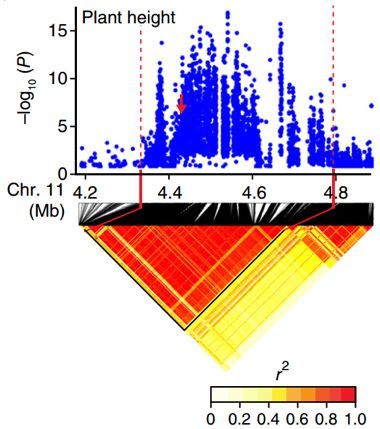
GWAS标记量 = 基因组大小/LD衰减距离

2、辅助分析进化与选择

在同一个连锁群上，LD衰减的慢说明该群体受到选择。一般来说，野生群体比驯化改良群体LD衰减快，异花授粉植物比自花授粉植物LD衰减快。比如玉米：地方品种1kb，自交系2kb，商用自交系100kb。

Haplotype Block（单体型块）

单体型块，即连锁不平衡区域，是指同一条染色体上处于连锁不平衡状态的一段连续的区域。单体型块分析可以用于筛选tag SNP、确定候选基因的范围等。



如果GWAS检测到显著关联的区间，可以通过进一步绘制局部的LD单体型块图，来进一步判断显著相关的SNP和目标基因间是否存在强LD关系。

#### 1、计算LDdeacy

##### 整个vcf文件中群体的LDdeacy

PopLDdecay -InVCF QC.sample-select-out-geno005-maf003.vcf -MaxDist 1000 -OutType 3 -OutStat overlap.all.stat

##### 计算vcf文件中的某个群体的LDdeacy(一般用这个)

示例：

PopLDdecay -InVCF QC.sample-select-out-geno005-maf003.vcf -OutStat GroupA\_sample.stat -MaxDist 1000 -SubPop GroupA\_sample.list

实操：

制作一列的ID的每个群体txt文件

(保证当前目录没有其他txt文件的情况下，运行以下脚本）

ls \*.txt | cut -d '.' -f 1 | sort -u | while read id; do (nohup PopLDdecay -InVCF QC.sample-select-out-geno005-maf003.vcf -OutStat ${id}\_sample.stat -MaxDist 1000 -SubPop ${id}.txt &); done

-InVCF：输入vcf文件。

-MaxDist：最长Decay距离。

-OutType：输出文件格式。

-OutStat：输出文件前缀。

#### 2、可视化LDdeacy

##### 单个群体的可视化：

Plot\_OnePop.pl -inFile overlap.all.stat.gz -bin1 500 -bin2 7000 -break 5000 -output overlap

##### 多个群体的可视化（一张图）：

Plot\_MultiPop.pl -inList Pop.ResultPath.txt -bin1 500 -bin2 7000 -break 5000 -output Fig

-bin1 -bin2 -break 可以看结果来加大，若结果不太平滑的话

- List Format :[Pop.ResultPath PopID ]

制作一个第一列为上面生成群体对应的路径，第二列为群体ID的txt文件

### 遗传多样性

##### 连续纯和片段（ROH）

参数设置plink --homozyg --homozyg-window-snp 50 --homozyg-density 50 --homozyg-window-het 3 --homozyg-window-missing 5 --out

#.indiv文件中NSEG为个体中ROH的个数，KB为长度，KBAVG为平均长度。

#计数时使用.hom文件，按区间统计

##### 近交系数

vcftools计算所有基因作为杂合型频率观察值（Ho）与预期值（He）的平均偏差，从而得出每个个体的近交系数值（F）

也有基于ROH计算的近交系数，以及PLINK中--het直接计算的近交系数

##### 核苷酸多样性

vcftools，滑窗50kb计算每个个体的核苷酸多样性π。

ANGSD计算π值，检测不同测序深度的个体π值的影响，参数为 -doThetas 。

vcftools --vcf test.vcf --keep id.txt --window-pi 10000 --out id\_pi

cat id\_pi|awk '{sum+=$5} END {print "Average = ", sum/NR}'

##### 亲缘系数（IBD）

plink --allow-extra-chr --chr-set 29 --file jiaxi1\_noinclude0-502502-geno02-maf03 --genome

结果文件解读：

PI\_HAT：为IBD比例 , 即 P(IBD=2) + 0.5\*P(IBD=1)，PI\_HAT的值与对应关系如下所示：

PI\_HAT＝0 无亲缘关系

PI\_HAT＝0.25 表兄弟

PI\_HAT＝0.5 亲子或兄弟姐妹

PI\_HAT＝1 本人或同卵双胞胎

RT:推断的关系类型

EZ:IBD期望共享值

Z0、Z1、Z2：P（IBD=0、IBD=1、IBD=2）

DST: IBS distance

### 选择信号

##### 概念：

选择信号：生物在自然选择或人工选择过程中，由于选择作用在基因组上留下的遗传多态性降低、连锁不平衡等选择印记。

选择性清除（selective sweep）：受选择基因座位的遗传多样性随着选择作用的发生而降低，并且由于相邻基因间的高度连锁不平衡，多态性一并降低的过程。

计算品种特异性的选择信号时，可用它与其他品种进行选择信号检测出，最终取交集

搭车效应：受到正向选择作用的基因座位不仅本身基因频率会升高，并且会借助连锁不平衡的作用使得附近位点的基因频率一并升高的现象。为去除这种效应，可比较相关基因上、下游各50Kb长的区域，以及邻近基因的FST，pi和XPEHH值。

为了排除这个效应，在对选择信号分析过程中，通常需要使用基于单倍体连锁不平衡的检验方法来校验受选择的位点。

选择压力放松会导致有害突变大量积累。

选择信号方法检测出的功能基因中，不同群体的某方面功能候选基因的非同义突变和同义突变的比值的比较可用于排除选择压力的放松。

##### 正选择信号的检测方法

Tajima's D：基于中性突变理论，首先计算群体多态性为点数S和平均非匹配数π。Tajima's D检验的统计量计算公式：

π表示平均非匹配数，S表示多态性为点数，a为 。

当Tajima's D = 0时，表明群体处于中性进化状态。

当Tajima's D < 0 时，提示低频变异位点多于中性预期，表明群体经历了负选择事件或者群体扩张。当某个点因为周围的位点受到选择而出现基因搭车效应时，也会表现为负值。

当Tajima's D > 0 时，提示高频变异位点多于中性预期，表明群体处于平衡选择。

#### ##Fst分析

FST：（群体间）基于等位基因频率

基于等位基因频率的常用检测方法，衡量群体间的分化程度进而推测可能存在的正选择。但是无法判断受选择方向。

全基因组 FST检验可以通过比较单个位点FST值找到 局部受到选择的基因，但由于一些牨种的基因组比 较大，在研究中会使用滑框(大动牨常用 50 k 的滑框) 进行 FST 值的计算。在中性突变下，FST 的大小主要受迁移和遗传漂变等因素的影响。但当一个突变受 到人工选择或自然选择时，其频率的升高就会增大 种群的分化水平，FST 值也越大(0＜FST＜1，1 表示该区域种群完全分化)。通过揭示变异的“离群”模式、多样性的缺失以及连锁标记受到选择的影响， 就可以实现对选择痕迹的检测。FST 的计算具有多种方式，目前最常用的方式是通过均方误差，计算公式如下：

其中 MSG 是群体内的均方误差，MSP 是群体间的均方误差， 是校正后整个群体的平均样本大小，此方法主要适用于不同群体之间选择信号的检测。

##### ##1 Fst(50K窗口，50K步长)

vcftools --vcf QC.JBC\_EU.filter-geno005-maf005.vcf --weir-fst-pop JBC.txt --weir-fst-pop EU.txt --out Fst\_JBC-EU --fst-window-size 50000 --fst-window-step 50000

##### ##2 根据fst值排序

sort -k 5 -g Fst\_JBC-EU.windowed.weir.fst > Fst\_JBC-EU.windowed.weir.sorted.fst

##### ##3 提取前5% 1%

tail -n 2487 Fst\_JBC-EU.windowed.weir.sorted.fst > Fst\_JBC-EU.sort.5%

EHH：扩展单倍型纯和度

当群体处于正选择作用下时，致因突变及其连锁位点在正选择的作用下，在短时间内达到较高频率，形成大片段的纯合单倍型。扩展单倍型纯合度(EHH)检验正是基于这样的特征来筛选受选择基因。

其中表示第i个扩展单倍型在样本中的数目，表示第t个核心单倍型在样本中的数目，S是独立单倍型的数目。

局限性：

EHH 检验也存在局限性，即由于忽略了不同染色体片段之间重组率的差异，而幵不适用于基因组范围内的选择信号强度的检测。认识到 EHH 检测的局限性后，Sabeti 等[14]又对其迚行了改迚，研发出了交叉群体扩展的单倍型纯合度(XP-EHH)检验。

#### ##XP-EHH分析

XP-EHH（群体间）:基于连锁不平衡和单倍型

优势与局限：

XPEHH值具有方向性，正的较大XPEHH值表明A群体可能在该位点发生了选择事件，负的较大XPEHH值表明B群体可能在改位点发生了选择事件。

只能筛选正选择信号

具有方向性，xpehh>2表示目标群体的受选择方向，xpehh<-2表示参考群体的受选择方向

##### ## beagle填充，还有所谓的phasing

java -jar -Xmn12G -Xms24G -Xmx48G /home/software/beagle.25Nov19.28d.jar gt=QC.JBC\_EU.filter-geno005-maf005.vcf out=QC.JBC\_EU.filter-geno005-maf005.beagle ne=82

##### ## 提取群体样品

bcftools view -S JBC.txt QC.JBC\_EU.filter-geno005-maf005.bagle.vcf.gz -Ov > JBC.beagle.vcf (不太行，好像是因为得用bgzip压缩并建索引，但是解压后bgzip用不了)

bcftools view -S EU.txt QC.JBC\_EU.filter-geno005-maf005.bagle.vcf.gz -Ov > EU.beagle.vcf

vcftools --gzvcf QC.JBC\_EU.filter-geno005-maf005.beagle.vcf.gz --keep JBC.txt --recode --recode-INFO-all --out JBC.beagle (头文件不对，记得改，否则会导致提出的文件没有基因型信息)

vcftools --gzvcf QC.JBC\_EU.filter-geno005-maf005.beagle.vcf.gz --keep EU.txt --recode --recode-INFO-all --out EU.beagle.

##### ##1 拆分染色体(1\_JBC.chr.sh 1\_EU.chr.sh)

for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf JBC.beagle.vcf --recode --recode-INFO-all --chr ${i} --stdout > JBC.chr${i}.vcf;

done

for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf EU.beagle.vcf --recode --recode-INFO-all --chr ${i} --stdout > EU.chr${i}.vcf;

done

##### ##2准备分染色体的map文件(2\_map.sh)

用的上一步的生成文件，也就是每个染色体的vcf的单倍型文件

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf EU.chr${k}.vcf --plink --out chr${k}.MT;

done

##### ##3 map计算遗传距离(3\_map.distance.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29; do awk 'BEGIN{OFS=" "} {print 1,".",$4/1000000,$4}' chr${k}.MT.map > chr${k}.MT.map.distance;

done

2和3用其中一个群体的做就行，distance文件时公用的

##### ##4 计算XPEHH(4\_xpehh.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29; do /home/software/selscan/bin/linux/selscan --xpehh --vcf JBC.chr${k}.vcf --vcf-ref EU.chr${k}.vcf --map chr${k}.MT.map.distance --threads 10 --out chr${k}.EU\_JBC;

done

# 得到 chr${k}.EU\_JBC文件29个 好像速度巨慢

# 重测序数据直接计算XPEHH（对每个100K窗口单倍型进行打分，标准化使其正态分布，并统计值>2的SNP所占的比例） 相对芯片来说步骤少点

# 最终得到的标准化的xpehh值表示的是xpehh>2的SNP在此区间所占的比例

# 具有方向性，xpehh>2表示目标群体的受选择方向，xpehh<-2表示参考群体的受选择方向

##### ##5 第一列改为染色体号(5\_add.chr.sh)

并将空格改为tab键分割

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do awk '{print '${k}',$2,$3,$4,$5,$6,$7,$8}' chr${k}.EU\_JBC.xpehh.out > Chr${k}.EU\_JBC.xpehh.out;

sed -i 's/ /\t/g' Chr${k}.EU\_JBC.xpehh.out;

done

##### ##6 加50kb窗口且标准化(6\_norm.xpehh.sh)

可按需求修改窗口大小

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do /home/software/selscan/bin/linux/norm --xpehh --files Chr${k}.EU\_JBC.xpehh.out --bp-win --winsize 50000;

done

# 产生两种文件.100bins.norm文件和.100bins.norm.50kb.windows文件，后面这个文件缺少第一列也就是染色体号

.100bins.norm.50kb.windows文件的每一列表头为

#<win start> <win end> <scores in win> <gt the fraction of XP-EHH scores >2> <lt the fraction of XP-EHH scores < -2> <approx percentile for gt threshold wins> <approx percentile for lt threshold wins> <max score> <min score>

##### ##7 提取并合并结果文件(7\_extract.merge.all.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do awk '{print '${k}',$1,$2,$4,$5,$8,$9}' Chr${k}.EU\_JBC.xpehh.out.norm.50kb.windows > Chr${k}.EU\_JBC.chart.xpehh.out.norm.50kb.windows;

cat ./\*.EU\_JBC.chart.xpehh.out.norm.50kb.windows > all.xpehh.out.norm.50kb.windows;

done

#### ##XP-CLR分析（XP-CLR软件）

XP-CLR 是陈华老师、Nick Patterson 和 David Reich 在 2010 年发表的方法，全称叫 the cross-population composite likelihood ratio test（跨群体复合似然比检验），是一种是基于选择扫荡（selective sweeep）的似然方法。

选择扫荡可以增加群体之间的遗传分化，导致等位基因频率偏离中性条件下的预期值。XP-CLR 利用了两个群体之间的多基因座等位基因频率差异建立模型，使用布朗运动来模拟中性下的遗传漂移，并使用确定性模型来近似地对附近的单核苷酸多态性（SNPs）进行选择性扫描。

该方法利用已有的数据集通过极大似然法估计等位基因频率等群体参数，然后预测等位基因频率在中性模型下的“失真”程度进而判断是否有选择发生。极大似然法估计群体参数的一个明显缺点就是过度依赖现有的数据集，可能会造成假阳性的选择信号，作者利用成对SNP连锁不平衡的方法，降低了可能存在连锁的SNP的权重。

##### ##1 拆分染色体(1\_JBC\_EU.chr.sh)

for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf QC.JBC\_EU.filter-geno005-maf005.beagle.vcf.gz --recode --recode-INFO-all --chr ${i} --out QC.JBC\_EU.chr${i};

done

##### ##2准备分染色体的map文件(2\_map.sh)

用的上一步的生成文件

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf QC.JBC\_EU.chr${k}.recode.vcf --plink --out chr${k}.MT;

done

##### ##3 map计算遗传距离(3\_map.distance.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29; do awk 'BEGIN{OFS=" "} {print 1,".",$4/1000000,$4}' chr${k}.MT.map > chr${k}.MT.map.distance;

done

##### ##4 计算XPCLR(4\_xpclr.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do xpclr --out chr${k} --format vcf --input QC.JBC\_EU.chr${k}.recode.vcf --samplesA EU.txt --samplesB JBC.txt --map chr${k}.MT.map.distance --chr ${k} --gdistkey None --phased --size 50000 --step 50000;

done

很慢，等个三天才能做出来

xpclr --out chr9 --format vcf --input QC.JBC\_EU.chr9.recode.vcf --samplesA EU.txt --samplesB JBC.txt --map chr9.MT.map.distance --chr 9 --gdistkey None --size 50000 --step 25000

A为参考群体，B为目标群体

--out: 输出文件

--format： 输入格式，vcf，hdf5，zarr，txt（对应2种基因型，和一个snp位点文件）

--input：输入vcf，或者hdf5， 选择txt时，不选择，

--samplesA： 样本A名称文件； txt时，不选择

--samplesB：样本B名称文件； txt时，不选择

--map: 基因位点文件

--popA： 样本A基因型文件，txt时使用

--popB： 样本B基因型文件，txt时使用

--chr： 染色体，和输入文件一致即可，每次分析一个

--ld： LD值，进行权重分析

--maxsnps： 一个窗口最大的SNP

--minsnps： 一个窗口最小的SNP

--size： 窗口大小

--step： 步长

##### ##5 提取并合并结果文件(5\_extract.merge.all.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do awk '{print $2,$3,$4,$12,$13}' Chr${k} > Chr${k}.EU\_JBC.chart.xpclr.50kb.windows;

cat ./\*.EU\_JBC.chart.xpclr.50kb.windows > all.xpclr.50kb.windows;

done

##### ##6 结果处理

没有norm\_xpclr值的按照0处理（个人认为）

#### 群体内选择信号

有时候由于群体间的分化程度过大，使用群体间选择信号难以衡量

#### ##iHS分析

iHS：（群体内）基于连锁不平衡和单倍型

标准化：

iHS=0时，表明该位点处于平衡选择状态。

当 iHS 值为较大的正值时，表示单倍型可能与祖先一致，当 iHS 值为较大的负值时，表示可能具有新衍生的单倍型。

重测序数据进行计算的时候，最后标准化后产生的iHS表示的是每个窗口内**|iHS| > 2**的比例，因此最高不超过1。

优势与局限：

iHS 检验 是基于祖先等位基因的一种检验方法，其优势是通 过祖先的信息能更准确地观察到迚化中受选择的情 况。而其劣势是现实中很难获得祖先的群体信息， 因为很多古老的品种已经灭绝或野生品种很难获得。

##过滤省略

QC.JBC.filter-geno005-maf005.vcf

##### ## beagle填充，还有所谓的phasing

java -jar -Xmn12G -Xms24G -Xmx48G /home/software/beagle.25Nov19.28d.jar gt=QC.JBC.filter-geno005-maf005.vcf out=QC.JBC.filter-geno005-maf005.beagle ne=11

##### ##1拆分染色体(1\_chr.sh)

for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --gzvcf QC.JBC.filter-geno005-maf005.beagle.vcf.gz --recode --recode-INFO-all --chr ${i} --stdout > chr${i}.vcf;

done

##### ##2准备分染色体的map文件(2\_map.sh)

用的上一步的生成文件，也就是每个染色体的vcf的单倍型文件

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf chr${k}.vcf --plink --out chr${k}.MT;

done

##### ##3 map计算遗传距离(3\_map.distance.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29; do awk 'BEGIN{OFS=" "} {print 1,".",$4/1000000,$4}' chr${k}.MT.map > chr${k}.MT.map.distance;

done

##### ##4 计算iHS(4\_ihs.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29; do /home/software/selscan/bin/linux/selscan --ihs --vcf chr${k}.vcf --map chr${k}.MT.map.distance --out chr${k}.iHS;

done

# 得到 ${k}.iHS.ihs.out文件29个 好像速度巨慢

# 重测序数据直接计算iHS（对每个100K窗口单倍型进行打分，标准化使其正态分布，并统计绝对值>2的SNP所占的比例） 相对芯片来说步骤少点

# 最终得到的ihs值表示的是|iHS|>2的SNP在此区间所占的比例

##### ##5 第一列改为染色体号(5\_add.chr.sh)

并将空格改为tab键分割

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do awk '{print '${k}',$2,$3,$4,$5,$6}' chr${k}.iHS.ihs.out > Chr${k}.ihs.out ;

sed -i 's/ /\t/g' Chr${k}.ihs.out;

done

##### ##6 加50kb窗口且标准化(6\_norm.ihs.sh)

可按需求修改窗口大小

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do /home/software/selscan/bin/linux/norm --ihs --files Chr${k}.ihs.out --bp-win --winsize 50000;

done

# 产生两种文件.100bins.norm文件和.100bins.norm.50kb.windows文件，后面这个文件缺少第一列也就是染色体号

##### ##7 提取并合并结果文件(7\_extract.merge.all.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do awk '{print '${k}',$1,$2,$4}' Chr${k}.ihs.out.100bins.norm.50kb.windows > Chr${k}.chart.ihs.out.50kb.windows;

cat ./\*.chart.ihs.out.500kb.windows > all.ihs.out.50kb.windows;

done

###按照第四列排序

sort -k 4n,4 all.ihs.out.50kb.windows > all.ihs.out.50kb.windows.sort

筛选出绝对值大的前5%注释

#### ## CLR分析(SweeD软件)

##### ##1拆分染色体(1\_chr.sh)

nohup bash chr.sh &

for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do vcftools --vcf QC.JBC.filter-geno005-maf005.vcf --recode --recode-INFO-all --chr ${i} --stdout > QC.JBC.filter-geno005-maf005.chr${i}.vcf;

done

##### ##2 CLR(2\_CLR.sh)

sed -i 's/\t/,/g' chr\_grid.txt

##############

for line in `cat chr\_grid.txt`

do

chr=${line%%,\*}

grid=${line##\*,}

/home/ywf/software/sweed/SweeD -name ${chr}.100kb \

-input QC.JBC.filter-geno005-maf005.chr${chr}.vcf \

-grid ${grid} \

-minsnps 200 \

-maf 0.05 \

-missing 0.1;

done

单个染色体做

nohup /home/ywf/software/sweed/SweeD -name JBC.chr1.100kb-2 -input JBC-geno005-maf005.chr1.recode.vcf -grid 15800 -maf 0.05 -missing 0.1 &

-name 指定程序运行名和输出结果文件的后缀

-input 输入需要分析的数据文件

-sampleList 指定参与分析的样本文件

-grid 每条染色体应分割成多少个网格，数值越大，分割的片段越多

-minsnps 参与计算的窗口中最少的SNP数量

-maf 小于该频率的次等位基因将被过滤掉

-missing 缺失率小于该值的位点将被过滤

##### ## 3对结果第一列加上染色体(3\_add.chr.delet.merge.sh)

for k in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29;

do awk '{print '${k}',$1,$2,$3,$4,$5}' SweeD\_Report.${k}.100kb > SweeD\_Report.chr.${k}.100kb;

sed -i 's/ /\t/g' SweeD\_Report.chr.${k}.100kb;

# 4删除结果中前三行

sed -i '1,3d' SweeD\_Report.chr.\*.100kb;

# 5 合并结果文件

cat ./SweeD\_Report.chr.\*.100kb > all.CLR.100k.txt;

done

其中，第二列为位置，第三列为似然法计算出的值

##### ##4 按照第三列排序，取前5%，并对其位置进行取整（使用excel表格处理）

24710\*0.05 = 1236

取整：=ROUND(B1,0)

注意：注释前请换染色体号

**加50k窗口**

**perl /home/sll/replace\_chr/replace.pl 5%.50k.txt /home/sll/replace\_chr/chr2-NC.txt 5%.50kNC.bed**

#### 基因注释

从4.0版本开始，SnpEff自动下载和安装所有数据库，且为数字染色体形式。

snpEff:

下载最新的安装包并解压

wget https://snpeff.blob.core.windows.net/versions/snpEff\_latest\_core.zip

unzip snpEff\_latest\_core.zip

查询对应物种的数据库

java -jar /home/sll/software/snpEff/snpEff.jar databases | grep "Bos"

注释：

java -Xmx4g -jar /home/sll/software/snpEff/snpEff.jar sheepv1.0 XXX.windowed.weir.sorted.1%.vcf > XXX.windowed.weir.sorted.1%.anno.vcf

## 拷贝数变异分析

#### CNV及其形成机制

在基因组水平上，CNV 不像单核苷酸多态性（SNPs）那样高频发生，但是它们对生物的表型、功能、适应性等有重要作用，且其产生的影响比 SNP 要更加强烈(Alkan et al. 2011; Zhang et al. 2009; Ho et al. 2020; Henrichsen et al. 2009a; Henrichsen et al. 2009b)。CNVs 是驱动基因和基因组进化的重要机制(Zhang et al. 2009)。越来越多的研究表明，CNVs 通过各种分子机制（如基因剂量、基因融合）对家畜的孟德尔性状和数量性状的表型变异具有重要影响。如果CNV 存在于蛋白编码区，则改变蛋白功能，如果在调控区，则改变基因表达水平(Liu and Bickhart 2012)。

CNV为基因组序列中长度为 50 bp - 5 Mb 的重复和缺失变异，在群体中，不同个体间重叠区域 CNV 的整合结果称作 CNV 区域（CNV Region, CNVR）。当这些 CNVs 被发现在蛋白质编码基因（CNV 基因）或 miRNA（CNV-miRNA）的基因组区域时，该遗传变异就可能作用于分子调控机制，并影响着生物的表型多样性和对疾病的易感性。

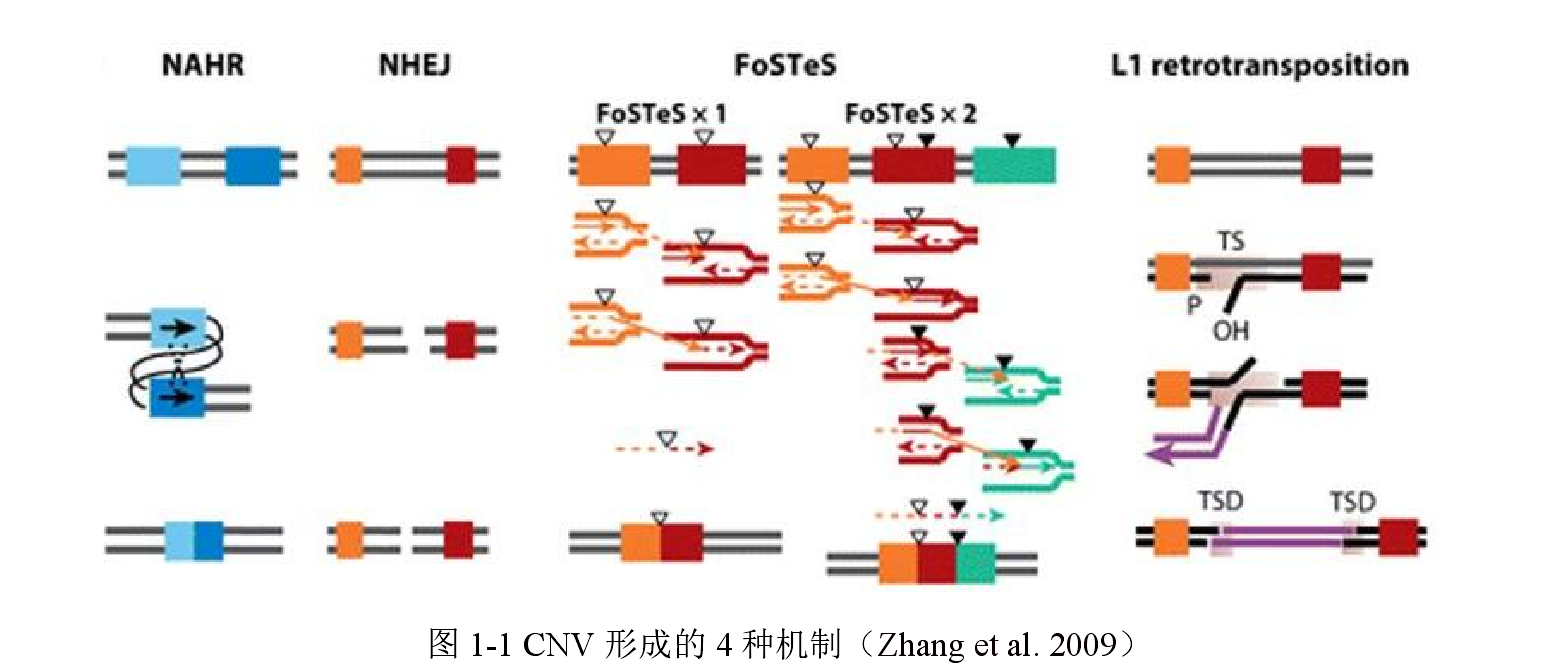
非等位同源重组（Non-allelic homologous recombination, NAHR）、非同源末端连接（Non-homologous end joining, NHEJ）、复制叉停滞与模板交换（Fork stalling and template switching, FoSTeS）、L1 介导的反转录转座（L1-mediated retrotranspositio, LINE1）是基因组中形成重排的主要机制，同时也是大部分 CNV 产生的原因。

由于非同源染色体之间的两个相似序列区域容易发生重组，所以 NAHR常发生在减数分裂和有丝分裂过程。姐妹染色单体之间发生交叉的过程可能增加 DNA片段或丢失另一个 DNA 片段，从而导致染色体片段的重复、缺失和倒位。

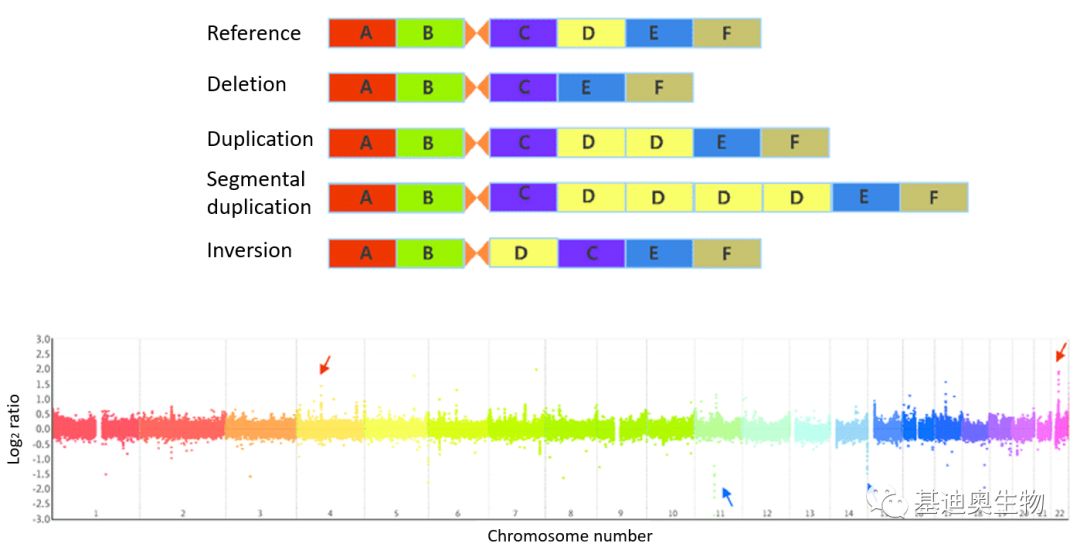
NHEJ 机制是细胞用来修复由电离辐射或活性氧物质引起的 DNA 双链断裂（Double-strand breaks, DSBs）的生理形式。

FoSTeS 是一种基于 DNA 复制的机制，可以解释复杂的基因组重排和 CNV。

L1 转座通过逆转录和整合方式引发 CNV 的产生。基因组中倒位和易位等组合事件的鉴别在技术上仍然具有挑战性，要完全鉴别其变异类型和成因的成本也很高，相对而言，CNV 更容易被鉴别，且可以为分析复杂变异提供强有力的遗传学证据。



Zhou 等（2022）把新疆褐牛基因组上的 CNV 分别比对到 UMD 3.1 和 ARS-UCD 1.2 两个参考基因组，结果表明新发布的 ARS-UCD 1.2 参考基因组更适合于 CNV 检测。



#### 检测手段

芯片法和测序法

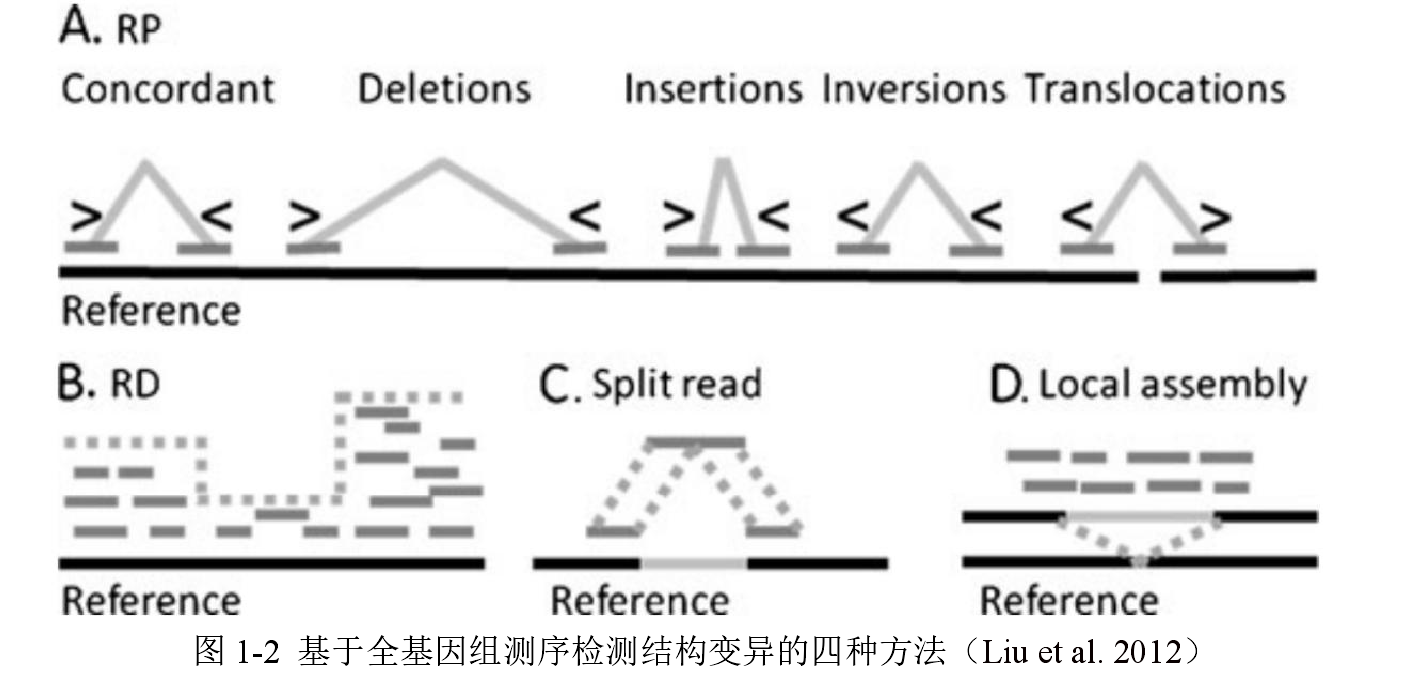
芯片法主要包括比较基因组杂交芯片（Array comparative genomic hybridization, a CGH）和 SNP 芯片（Singlenucleotide polymorphism arrays）。

DNA 测序法主要包括全基因组测序（Whole genome sequnecing, WGS）和单分子测序。

Paired-end mapping（PEM）方法、Split read（SR）方法、Read depth（RD）方法和 De novo 组装是检测基因组拷贝数变异的主要策略。

RD 方法通过利用短读取序列数据与参考基因组进行比对，标准化每个区域的读取深度，其中低或零深度的区域被解释为缺失，深度增加的区域被解释为拷贝数的重复或扩增。RD 方法相较于 PEM 和 SR 方法有着较高的检出率和准确性。

现在基于二代测序数据分析 CNV的研究还存在着样本少、测序深度较低、选用参考基因组质量差异大等不足之处，这些因素均会对 CNV 检测结果产生不利影响



#### matchclips2（刘正喜师姐的脚本，还得是师姐）

基于long soft clips的CNV断点计算方法

##### 每个样品分别进行matchclips2计算：

很快，结果也挺准确的

/home/software/matchclips2/matchclips -t 4 -f /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna -b BC-448.sorted.addhead.markdup.bam -o CNV.BC-448.txt

ls \*bam|cut -d"." -f 1 | sort -u | while read id; do /home/software/matchclips2/matchclips -t 4 -f /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna -b ${id}.sorted.addhead.markdup.bam -o ${id}.cnv.txt; done

##### 结果文件：

1、CHROM

2、起始位置

3、中止位置

4、DEL或DUP

For DEL, the 5' break point is BEGIN and 3' is END,

For DUP, the 5' break point is END and 3' is BEGIN.、

5、区域长度：3' break point - 5' break point

6、UN，碱基数量This number tells how many bases are repeated at the two break points

7、RD:n1;n2;n3:s

##### 合并结果：

#保证当前目录没有其他txt文件的情况下

find -name '\*txt' -exec basename {} \; > cnvf.txt

#进行合并

/home/software/matchclips2/cnvtable -L 10000 -cnvf cnvf.txt -O 0.5 -o overlap.txt

cnvf.txt为含有上一步结果文件名称的txt文件，一列一个

-cnv STR file names separated by white spaces

-i STR IDs for the files separated by white spaces

-cnvf STR a file with list of file names and(or) ids

-l INT minimum length to include, INT=3

-L INT maximum length to include, INT=10000000

-t STR only process STR type of CNVs

-R STR subset cnvs in region STR

-chr 1-22XY are treated as chr[1-22XY]

-O FLOAT minimum reciprocal overlap ratio [0.0, 1.0], FLOAT=0.5

-o STR outputfile, STR=STDOUT

##### matchclips2.sh：

touch matchclips2.sh

bash matchclips2

##运行

ls \*bam|cut -d"." -f 1 | sort -u | while read id; do /home/software/matchclips2/matchclips -t 4 -f /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna -b ${id}.sorted.addhead.markdup.bam -o ${id}.cnv.txt; done

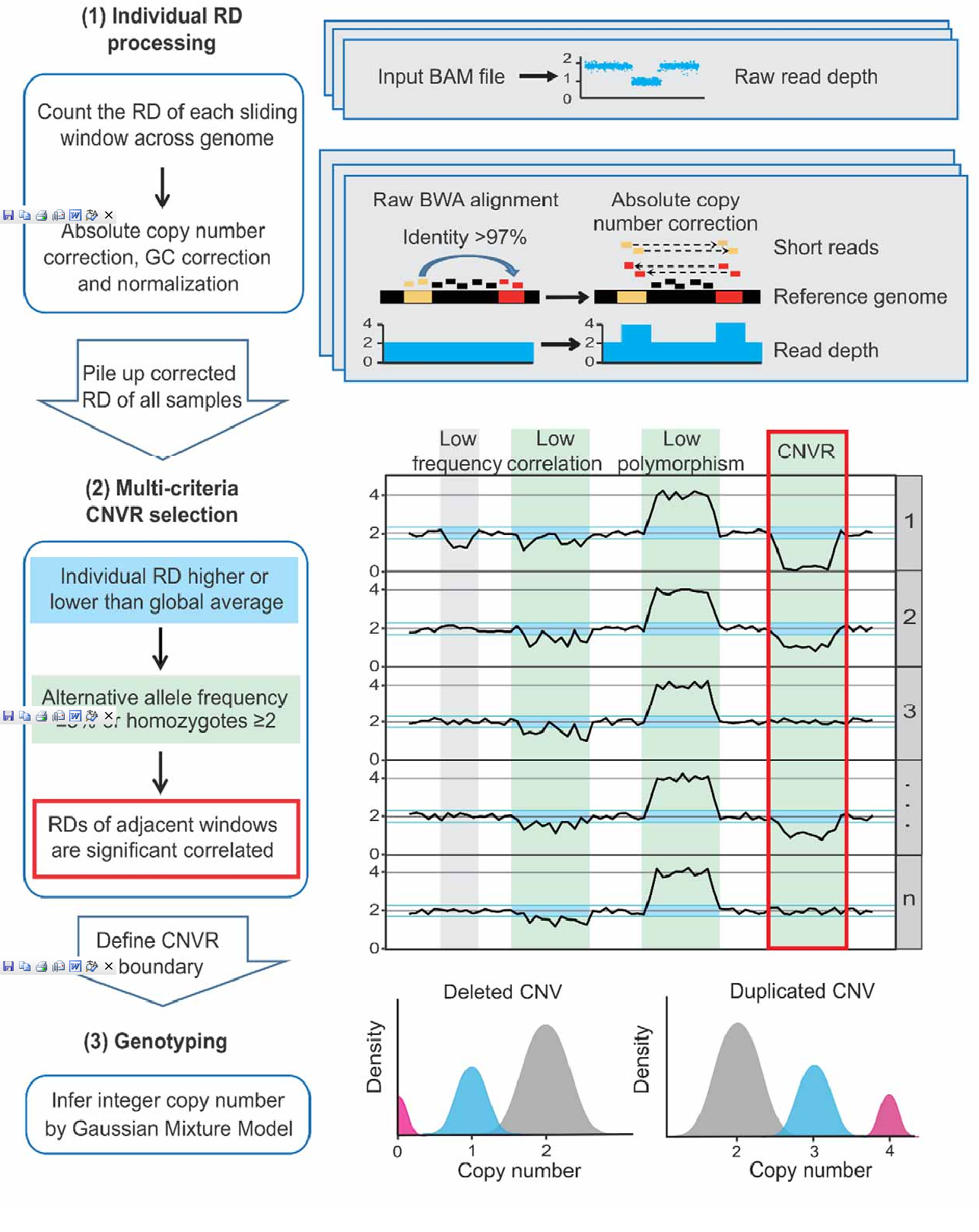
#保证当前目录没有其他txt文件的情况下

find -name '\*txt' -exec basename {} \; > cnvf.txt

#进行合并

/home/software/matchclips2/cnvtable -L 10000 -cnvf cnvf.txt -O 0.5 -o overlap.txt

#### CNVcaller（输入文件用绝对路径）



怎么定义一个窗口是否为拷贝数变异窗口？

在定义一个窗口是否为拷贝数变异窗口时，CNVcaller 首先需要对各个样本设定阈值。考虑

到测序过程中，可能会有一定比例的样本由于各种系统偏差导致测序 reads 出现系统偏斜，

从而造成窗口读段数不是随机分布，即窗口读段数的标准差会明显升高。CNVcaller 默认群

体中至少 75%的样本是测序质量较高的样本。根据经验，校正后滑动窗口 reads 数的变异系数（coefficient of variation，CV）一般要低于 0.2。对于测序质量较高（或者 CV 值较低）的前75%的样本，CNVcaller 默认拷贝数缺失与重复的阈值分别设为：1-0.35 和 1+0.35；针对测序质量相对差（或者 CV 值较大）的后 25%的样本，CNVcaller 会通过样本的 CV 与所有样本 CV 的四分之三分位数的比值计算得到该样本的矫正系数γ，并将拷贝数缺失与重复的阈值分别设为：1-0.35\*γ和 1+0.35\*γ。对于不符合哈迪温伯格平衡的群体（如大豆、小麦杂合变异比例极低的纯系群体），CNVcaller 默认拷贝数缺失与重复的阈值分别为：1-0.75 和1+0.75。

关于频率，CNVcaller 通过分别统计基因组每个窗口杂合变异和纯合变异的样本数，默认杂

合变异或纯合变异样本占所有样本 10%以上，即CNV在二倍体群体中的变异频率超过5%，

该窗口就被判定为拷贝数变异窗口。对于纯系群体，CNVcaller默认纯合变异个体数超过 2

的窗口为拷贝数变异窗口。

CNV区域（CNV Region, CNVR）是指不同个体间检测得到的CNV具有一部分重叠区域，把重叠的CNV进行整合，合并成为一个CNVR

##### 1.在当前目录创建referenceDB.windowsize文件，构建参考基因组数据库

perl CNVReferenceDB.pl ref.fa -w 400

perl /home/sll/miniconda3/CNVcaller/bin/CNVReferenceDB.pl /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna -w 800

作用：将参考基因组按用户指定大小（-w）的滑动窗口及一定的步长（内置是滑动窗口

大小的一半）分别统计基因组上每个窗口的 GC、repeat 及 gap 含量。

-w the window size (bp) for all samples [default=800] 滑动窗口大小及步长，内置是滑动窗口大小的一半， >10x使用400-1000，<10x使用1000-2000

-l the lower limit of GC content [default=0.2] 窗口可能含有的最低GC比例，低于该比例的窗口不会进入后续计算。

-u the upper limit of GC content [default=0.7] 窗口可能含有的最高GC比例

-g the upper limit of gap content [default=0.5] 窗口可能含有的最高gap比例

生成一个“referenceDB.800”的文件。

第一列，染色体名称 第二列，窗口在对应染色体上的序号

第三列，窗口实际起始位置 第四列，GC 含量

第五列，重复序列含量 第六列，gap 比例

##### 2.计算每个窗口的绝对拷贝数

设置Individual.Process.sh中的环境变量

export CNVcaller=/home/sll/miniconda3/CNVcaller

bash /home/sll/miniconda3/CNVcaller/Individual.Process.sh -b `pwd`/SRR8588100.sorted.addhead.markdup.bam -h SRR8588100 -d /home/sll/miniconda3/CNVcaller/Btau5.0.1\_800\_link -s none

-h BAM 文件的标头信息，需要与 BAM 文件中 SM 标签保持一致（用户可以通过 samtools view -H BAM 查看）。

-d 校正所需要的dup文件。查看物种dup文件[JiangYuLab/CNVcaller (github.com)](https://github.com/JiangYuLab/CNVcaller/tree/master#generate-your-own-duplicated-window-record-file)。

-s 性染色体名称。CNVcaller 会根据给定的性染色体所有窗口读段数的中位数与所有常染色体窗口读段数的中位数比例来确定该个体的性别。

运行结束后，三个默认文件夹（RD\_raw、RD\_absolute、RD\_normalized）将会在当前目录下被创建，分别包含了每个样本的全基因组所有窗口原始读段数、通过 link 文件合并后的读段数，以及每个样本经 GC 测序偏斜校正和标准化后的绝对拷贝数。标准化的文件名显示了该个体基因组所有窗口的读段数平均值、标准差与性别（1 为 XX 或 ZZ，2 为 XY 或ZW），其中平均值和标准差可用于样本的质控。

绝对拷贝数为 1 时，表示为正常的拷贝数，即正常二倍体；0.5 表示杂合缺失；0 表示纯合缺失；1.5 表示杂合重复；2 表示纯合重复；绝对拷贝数超过 2 表示复杂的多次重复。

创建自己所需要的dup文件:

1、Split genome into short kmer sequences

python 0.1.Kmer\_Generate.py [OPTIONS] FAFILE WINSIZE OUTFILE

<FAFILE> Reference sequence in FASTA format

<WINSIZE> The size of the window to use for CNV calling

<OUTFILE> Output kmer file in FASTA format

python /home/sll/miniconda3/CNVcaller/bin/0.1.Kmer\_Generate.py /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna 800 kmer.fa

2: Align the kmer FASTA (from step 1) to reference genome using blasr.

sawriter 创建sa索引文件

/home/sll/miniconda3/CNVcaller/sawriter /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna.sa /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna

blasr kmer.fa /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna --sa /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna.sa --out kmer.aln -m 5 --noSplitSubreads --minMatch 15 --maxMatch 20 --advanceHalf --advanceExactMatches 10 --fastMaxInterval --fastSDP --aggressiveIntervalCut --bestn 10

3: Generate duplicated window record file.

python 0.2.Kmer\_Link.py [OPTIONS] BLASR WINSIZE OUTFILE

<BLASR> blasr results (-m 5 format)

<WINSIZE> The size of the window to use for CNV calling

<OUTFILE> Output genome duplicated window record file

python /home/sll/miniconda3/CNVcaller/bin/0.2.Kmer\_Link.py kmer.aln 1000 Bos\_ARS1.2\_window.link

##### 3.拷贝数变异区域的确定（多样本合并，单个样本会报错）

通过综合考虑绝对拷贝数的分布、变异的频率及相邻窗口的显著相关性来初步确定 CNVR 的边界（primaryCNVR）。最后，将相邻且拷贝数在群体中分布显著相关的 CNVR进一步合并得到最终的拷贝数变异检测结果（mergedCNVR）。

cd RD\_normalized

cp referenceDB.800 RD\_normalized

记得把第一步生成的referenceDB.800放进去,输入文件用绝对路径！！！

list 上一步输出的文件名的列表，一行一个，多个样本，需包含绝对路径，RD\_normalized文件夹中的文件。

ls -R `pwd`/\*sex\_1 > list.txt

touch exclude\_list

exclude\_list 为空文件，提前创建

运行完成后，会在当前目录下产生两个文件名为primaryCNVR，mergeCNVR 的文本文件。

bash /home/sll/miniconda3/CNVcaller/CNV.Discovery.sh -l `pwd`/list.txt -e `pwd`/exclude\_list -f 0.1 -h 1 -r 0.1 -p primaryCNVR -m mergeCNVR

-l 个体经绝对拷贝数校正后的结果文件列表

-e 该列表中的样本不被用于 CNVR 的检测，但结果文件会记录这些样本的绝对拷贝数。exclude\_list 为空文件代表所有个体将被用于CNVR检测。

-f 当一个窗口有超过该频率的个体绝对拷贝数与正常拷贝（“1”）显著差异（杂合删除或者

杂合复制）时，就定义该窗口为候选拷贝数变异窗口。

-h 当一个窗口有超过该频数的个体绝对拷贝数与正常拷贝（“1”）显著差异（纯合删除或者

纯合复制）时，就定义该窗口为候选拷贝数变异窗口。

##只需要满足-f 与-h 中的任意一个即可

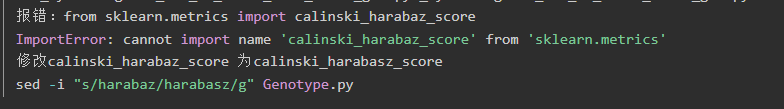
-r 在定义 CNVR 时，如果相邻（没有 overlap）候选拷贝数变异窗口的绝对拷贝数的相关系数高于该值将被合并。

####推荐使用显著水平为 0.01 的 pearson 相关系数。该值越高拷贝数检测的准确性越高，但获得的拷贝数变异区域越碎片化。

##### 4.基因型判定

利用混合高斯模型将每个样本的拷贝数归入不同的基因型分类，并以 VCF 格式输出，

以方便后续通过全基因组关联分析挖掘与重要经济性状有关的拷贝数变异。

python /home/sll/miniconda3/CNVcaller/Genotype.py --cnvfile mergeCNVR --outprefix genotypeCNVR --nproc 4

sed -i "s/harabaz/harabasz/g" /home/sll/miniconda3/CNVcaller/Genotype.py

--cnvfile CNVR 结果文件，包含全部样本的拷贝数信息，上一步的输出结果（mergeCNVR）。--outprefix 输出结果文件的前缀，默认会输出两个文件，其中，后缀为 tsv 的文件记录了分

类结果的基本统计信息，方便后续过滤低质量的 CNVR；后缀为 vcf 的文件为常规 VCF 格

式。

--nproc 程序使用的进程数，默认为单进程，使用此参数可显著减少程序运行时间，但会增

加内存消耗。

--merge （若想使用CNVR进行后续的群体遗传学分析，则使用这个参数）为了得到更多的双等位 CNVR 用于后续分析，可以使用--merge 选项。使用后，会增加一个后缀为 \_merge.vcf 的 输 出 文 件 ， 类 似 <outprefix>.vcf 文 件 ， 区 别 在 于<outprefix>\_merge.vcf 中把所有重复算作一种变异类型

##### 5.结果文件

genotypeCNVR.vcf文件内容

CHROM: 所在染色体

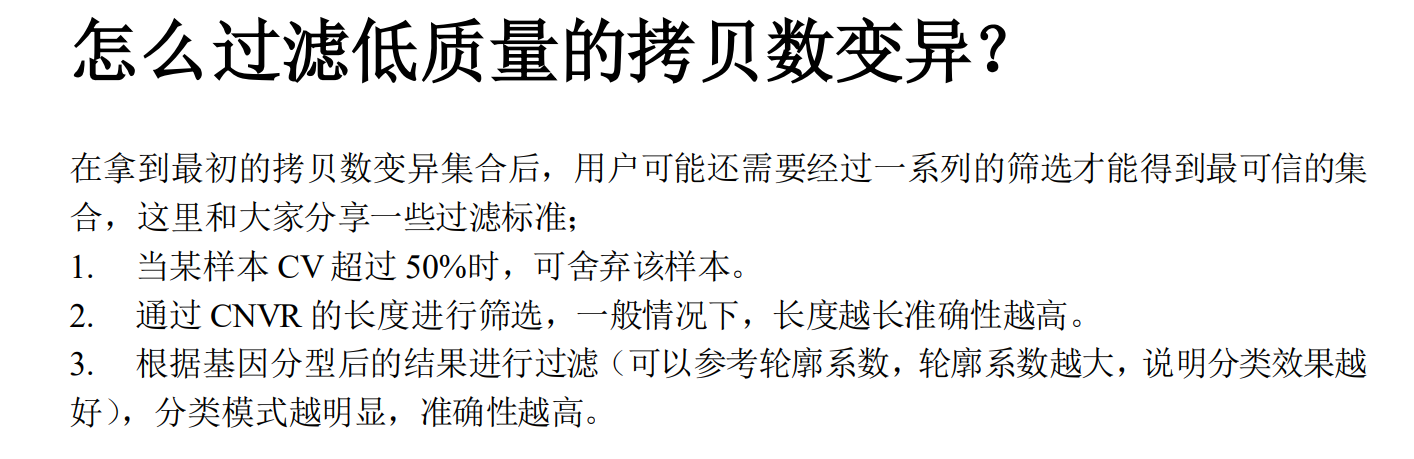
POS：CNVR的起始坐标位置

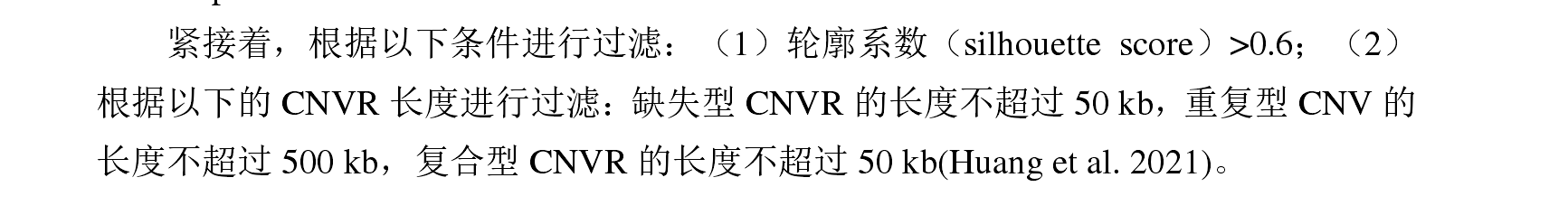
ID：CNVR的编号，格式为：序列坐标:起始位置-终止位置

ALT: 变异类型，包括CN0、CN1、CN2和CNH，分别代表0个拷贝、1个拷贝、2个拷贝和超过2个的拷贝

INFO：包含CNVR的终止位置（END），CNVR的变异类型（SVTYPE），基因分型的对数自然值（LOGLIKELIHOOD）和轮廓系数（SILHOUETTESCORE）

FORMAT：每个个体基因分型结果的输出格式，GT和CP分别代表个体的基因分型结果和绝对拷贝数。



genotypeCNVR.vcf文件中：

number：表示当前CNVR包含的窗口数

gap：gap（含N的比例）

repeat：重复序列（依据基因组序列小写字母）

gc：gc的比例

kmer：代表该CNVR的所有kmer在基因组上出现次数的中位数还是平均数，用来表示该CNVR的重复性。

og\_likelihood：表示该CNVR的所有样本绝对拷贝数的分型的效果；

average：该CNVR所有样本绝对拷贝数的平均值

sd（standard deviation）：表示该CNVR所有样本绝对拷贝数的标准差；

CNVcaller 默认拷贝数缺失与重复的阈值分别设为：1-0.35 和 1+0.35

M：纯和重复，基因型为1/1或2/2，cp>4

AA：基因型为0/0，cp在2左右

Ad：缺失，基因型为0/1

dd：纯和缺失，基因型为1/1，cp<0.5

AB：重复，基因型为0/2或0/1，cp在3左右

BB：纯和重复，基因型为2/2或1/1

BC：基因型为2/3或1/2

B、C、M代表重复，A代表正常，d代表缺失

##### 6.CNVCaller.sh

2199

bash CNVCaller.sh

# 对基因组创建窗口文件（以后直接用即可）

perl /home/sll/miniconda3/CNVcaller/bin/CNVReferenceDB.pl /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna -w 800

# 计算每个窗口的绝对拷贝数

ls \*bam|cut -d"." -f 1 | sort -u | while read id; do bash /home/sll/miniconda3/CNVcaller/Individual.Process.sh -b `pwd`/${id}.sorted.addhead.markdup.bam -h ${id} -d /home/sll/miniconda3/CNVcaller/Btau5.0.1\_800\_link -s none; done

# 将referenceDB.800文件复制到RD\_normalized目录下

cp referenceDB.800 RD\_normalized

# 进入RD\_normalized目录

cd RD\_normalized

# 将RD\_normalized目录下新生成的sex\_1结尾的文件名，以绝对路径的形式写入list.txt中

ls -R `pwd`/\*sex\_1 > list.txt

# 新建exclude\_list文件

touch exclude\_list

# 拷贝数变异区域的确定

bash /home/sll/miniconda3/CNVcaller/CNV.Discovery.sh -l `pwd`/list.txt -e `pwd`/exclude\_list -f 0.1 -h 1 -r 0.1 -p primaryCNVR -m mergeCNVR

# 基因型判定

python /home/sll/miniconda3/CNVcaller/Genotype.py --cnvfile mergeCNVR --outprefix genotypeCNVR --nproc 8

#### cnvnator（拉跨）

检测出的片段长度很长，假阳性结果率较低，还算比较准确的一个软件

##### 1.提取mapping信息

/home/software/CNVnator\_v0.4.1/src/cnvnator

cnvnator -root SAMN05788539.root -tree SAMN05788539.sorted.addhead.markdup.bam -unique

nohup /home/software/CNVnator\_v0.4.1/src/cnvnator -root BC-448.root -tree BC-448.sorted.addhead.markdup.bam -unique &

#提取比对后的reads，构建树，如果程序之前运行失败，已经生成了一个root文件，在下次重新运行时一定要删除该root文件，如果不删除，新的分析结果会追加到错误的root文件中，影响后续分析。另外，可以添加 -chrom 函数指定染色体 如 -chrom 1 2 代表选择1，2号染色体。

##### 2.生成质量分布图HISTOGRAM

cnvnator -root SAMN05788539.root -d -his 100 -d /ref

nohup /home/software/CNVnator\_v0.4.1/src/cnvnator -root BC-448.root -d -his 100 &

cnvnator -root SAMN05788539.root -d -his 100 -fasta ref

(将参考基因组放于当前文件夹时)

# 100指的是bin size，可以通过-eval参数进行筛选，也可以根据经验值进行确定，一般测序深度20-30x选取bin size大小100，2-3x选取500，100x选取30。

# -d指定目录，内部存放给染色体的fasta文件，该参数指针对报错信息显示不能解析基因组文件，此时需要指定参考基因组的位置，且各染色体需要拆分成单独的fasta文件（目录下有其文件也可以，只要有所有染色体的序列就好，软件会自动识别）

##### 3.生成统计结果

cnvnator -root SAMN05788539.root -stat 100

/home/software/CNVnator\_v0.4.1/src/cnvnator -root BC-448.root -stat 100

##### 4.RD（reads depth）信息分割partipition

cnvnator -root SAMN05788539.root -partition 100

/home/software/CNVnator\_v0.4.1/src/cnvnator -root BC-448.root -partition 100

##cnvnator在进行变异检测时，以提供的bin size对整个基因组进行切割，之后按照RD(read-depth)为基准进行cnv的检测。

##### 5.变异检出

/home/software/CNVnator\_v0.4.1/src/cnvnator -root BC-448.root -call 100 > BC-448-cnvout.txt

运行之后输出结果如下：

#第一列变异类型

#第二列位点信息

#第三列CNV大小

#第四列为标准化参数normalized\_RD

#第5-8列为e-value值，其中第五列越小，说明结果越准确

#第九列q0质量值

##### 6.转为vcf格式

/home/software/CNVnator\_v0.4.1/src/cnvnator2VCF.pl cnv.call.txt > cnv.vcf

#### Delly

delly call -g /home/sll/genome-cattle/ARS-UCD1.2/GCF\_002263795.1\_ARS-UCD1.2\_genomic.fna BC-448.sorted.addhead.markdup.bam > BC-448.cnv.vcf

Delly

主要检测sv。

·deletions

·tandem duplications

·inversions

·translocations

joint call or 分开

specificity > sensitivity : call separately

sensitivity > specificity : joint call

cnv

#分开

delly -t DUP -o sample1.out -g ref.fa sample1.bam

delly -t DUP -o sample2.out -g ref.fa sample2.bam

#...

delly -t DUP -o sample6.out -g ref.fa sample6.bam

# and then merge the output vcfs

# joint call

delly -t DUP -o allSamples.out -g ref.fa sample1.bam sample2.bam sample3.bam sample4.bam sample5.

Delly v0.7.8 可以一次性call 所有类型的SVs；低版本的需要通过-t指定SV类型

输入文件

1、bam文件：Bam files need to be sorted, indexed and ideally duplicate marked. If multiple libraries are present for a single sample these need to be merged in a single bam file with unique ReadGroup tags.

2、.excl: a file include information of telomere and centromere regions and also all unplaced contigs, those regions will be removed from calling.

有群体时，建议将所有个体的.bcf文件合并后，再进行genotyping。

#install

git clone --recursive https://github.com/dellytools/delly.git

cd delly/

make all

#也可以直接下载编译后的二进制文件，直接使用，不需安装 https://github.com/dellytools/delly/releases/

#Running

DB="/root/cc/DB"

i="SXY"

samtools index $i.recal.bam （建索引）

~/cc/biosoft/delly/delly\_v0.7.8\_linux\_x86\_64bit call -g $DB/hg19\_chr.fa -x exc1.excl -o delly\_out/$i.exc1.bcf $i.recal.bam

#bcftools view $i.target.bcf >$i.target.vcf （bcf转vcf格式）

#Merge SV sites into a unified site list

delly merge -o sites.bcf s1.bcf s2.bcf ... sN.bcf

#Genotype this merged SV site list across all samples. This can be run in parallel for each sample.

delly call -g $DB/target-ref.fa -v sites.bcf -o s1.geno.bcf s1.bam

#Merge all genotyped samples to get a single VCF/BCF using bcftools merge

bcftools merge -m id -O b -o merged.bcf s1.geno.bcf s2.geno.bcf ... sN.geno.bcf

#Apply the germline SV filter which requires at least 20 unrelated samples

delly filter -f germline -o germline.bcf merged.bcf

bcftools view germline.bcf >germline.vcf

脚本记录：

/GS01/project/wudd\_group/wudd2t4/project\_lzx/software/delly\_v0.8.3\_linux\_x86\_64bit call -g /GS01/project/wudd\_group/wudd2t4/project\_lzx/sheep/GCF\_002742125.1\_Oar\_rambouillet\_v1.0\_genomic.fna -o SRR11657523.bcf SRR11657523.sorted.addhead.markdup.bam

插入（Insertion, INS）

缺失（Deletion, DEL）

反转（Inversion, INV）

染色体内部易位（Intra-chromosomal Translocation, ITX）

染色体间易位（Inter-chromosomal Translocation, CTX)

#### LUMPY（挺牛的一个软件）

每种算法都要其优势和不足之处，综合运用多种策略有助于提高检测的灵敏度，lumpy就是这样一款软件，集合了read-pair，split-read，read-depth, 等多种策略来预测CNV

##### 1. mapping

推荐采用bwa-mem算法将双端序列比对到参考基因组上，为了加快运行速度，这里用samblaster软件进行markduplicate, 用法如下

samblaster --excludeDups \

--addMateTags \

--maxSplitCount 2 \

--minNonOverlap 20 \

samtools view -Sb - > sample.bam

##### 2. extract discordant paired-end alignments

discordant reads指的是R1和R2端比对之间的距离超过了期望的插入片段长度或者比对到了不同链的reads,

**samtools view -b -F 1294 \**

**sample.bam \**

**> sample.discordants.unsorted.bam**

##### 3. extract split-reads alignments

split-reads指的是覆盖了断裂点的单端reads,这些reads根据断裂点拆分成subreads后可以正确的比多到参考基因组上。在软件的安装目录，自带了一个名为extractSplitReads\_BwaMem的脚本，用于提取split-reads, 用法如下

**samtools view -h sample.bam \**

**| scripts/extractSplitReads\_BwaMem -i stdin \**

**| samtools view -Sb - \**

**> sample.splitters.unsorted.bam**

##### 4. sort bams

软件要求输入的bam文件必须是排序之后的文件，所以对提取的两个子bam进行排序，用法如下

**samtools sort \**

**sample.discordants.unsorted.bam \**

**sample.discordants**

**samtools sort \**

**sample.splitters.unsorted.bam \**

**sample.splitters**

##### 5. run lumpy

lumpyexpress是lumpy的一个封装脚本，使用起来更加方便，基本用法如下

lumpyexpress \

-B sample.bam \

-S sample.splitters.bam \

-D sample.discordants.bam \

-o sample.vcf

##### 6. genotype

检测到的CNV, 可以用svtyper这个软件预测在样本中的分型结果，用法如下

**svtyper \**

**-B sample.bam \**

**-S sample.splitters.bam \**

**-i sample.vcf**

**> sample.gt.vcf**

#### vst分析：

Vst分析是类似于Fst的一个指标，用来衡量群体间每个CNVR差异大小的统计量。

Vst＝（Vt－Vs）/Vt

Vt表示所有样本该区域拷贝数大小的方差，Vs表示两个群体各自的方差根据各自群体大小加权之后的值。

标准差函数： ———— STDEV(A1:A6)

方差函数： ———— VAR(A1:A6)

Vst的值介于0-1之间，值越大表示群体间该区域拷贝数变异差异越大，反之则越小。

## ANNOVAR变异注释

##### 1、下载地址：

http://www.openbioinformatics.org/annovar/download/0wgxR2rIVP/annovar.latest.tar.gz

##### 2、解压：

tar -zxvf annovar.latest.tar.gz

##### 3、构建注释数据库：

# 下载注释gtf和fa文件:从UCSC下载。最好是

bosTau9.fa

bosTau9.refGene.gtf

##### 4、下载安装gtfToGenePred工具

wget http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86\_64.v369/gtfToGenePred

chmod +x gtfToGenePred

##### 5 用 gtfToGenePred 工具将 GTF file 转换 GenePred file

/home/sll/software/gtfToGenePred -genePredExt bosTau9.refGene.gtf bos1.2\_refGene.txt

/home/sll/software/gtfToGenePred -genePredExt Bos\_taurus.ARS-UCD1.2.108.gtf Bos1.2\_refGene.txt

NCBI下载的第一次可能会能报错：使用下面脚本解决

位置：/home/sll/software/annovar/cattle

replacegtf.py （记得改内容）

/home/sll/software/gtfToGenePred -genePredExt GCF\_002263795.1\_ARS-UCD1.2\_genomic\_replace.gtf ARS-UCD1.2\_refGene.txt

##### 6 生成转录组信息文件

UCSC

perl /home/sll/software/annovar/retrieve\_seq\_from\_fasta.pl --format refGene --seqfile bosTau9.fa bos1.2\_refGene.txt -outfile ARS1.2\_refGeneMrna.fa

Ensembl

perl /home/sll/software/annovar/retrieve\_seq\_from\_fasta.pl --format refGene --seqfile Bos\_taurus.ARS-UCD1.2.dna.toplevel.fa Bos1.2\_refGene.txt -outfile Bos1.2\_refGeneMrna.fa

**# -format指定gene definition file格式**

**# -seqfile 指定基因组序列文件名称**

**# -outfile 指定输出mRNA序列文件的名称**

建库完成

##### 注释示例：

###### 1、vcf转为适宜的格式

perl /home/sll/software/annovar/convert2annovar.pl -format vcf4old QC.JPBC-geno005-maf003.bed.vcf -outfile QC.JPBC-geno005-maf003.avinput

#关于-format vcf4,并没有保留全部位点：

#WARNING to old ANNOVAR users: this program no longer does line-to-line conversion for multi-sample VCF files. If you want to include all variants in output, use '-format vcf4old' or use '-format vcf4 -allsample -withfreq' instead.

###### 2、annotate\_variation注释

数字染色体

第一种，使用annotate\_variation.pl ，三种类型分开：

perl /home/sll/software/annovar/annotate\_variation.pl -out jpbc.anno -dbtype refGene -buildver Bos1.2 QC.JPBC-geno005-maf003.avinput /home/sll/software/annovar/cattle/ -csvout

# -geneanno 表示使用基于基因的注释 一般是默认的

# -dbtype refGene 表示使用"refGene"类型的数据库

# -out jpbc.anno 表示输出以jpbc.anno为前缀的结果文件

# 基于基因

annotate\_variation.pl -geneanno -dbtype refGene -buildver hg19 example/ex1.avinput humandb/

#基于区域

annotate\_variation.pl -regionanno -dbtype cytoBand -buildver hg19 example/ex1.avinput humandb/

#基于筛选

annotate\_variation.pl -filter -dbtype exac03 -buildver hg19 example/ex1.avinput humandb/

第二种，方便一点，基于table\_annovar.pl，直接注释三种类型：

table\_annovar.pl是ANNOVAR多个脚本的封装，可以一次性完成三种类型的注释

perl /home/sll/software/annovar/table\_annovar.pl QC.JPBC-geno005-maf003.avinput /home/sll/software/annovar/cattle/ -buildver Bos1.2 -out jpbc.anno -remove -operation g -protocol refGene -nastring NA -csvout

# -buildver Bos1.2 表示使用的参考基因组版本为Bos1.2

# -out jpbc.anno 指定输出文件前缀为jpbc.anno

# -remove 表示删除中间文件

# -protocol 后跟注释来源数据库名称，每个protocal名称或注释类型之间只有一个逗号，并且没有空白

# -operation 后跟指定的注释类型，和protocol指定的数据库顺序是一致的，g代表gene-based、r代表region-based、f代表filter-based

# -nastring NA 表示用NA替代缺省值

# -csvout 表示最后输出.csv文件

