BerryGenomics
贝瑞基因

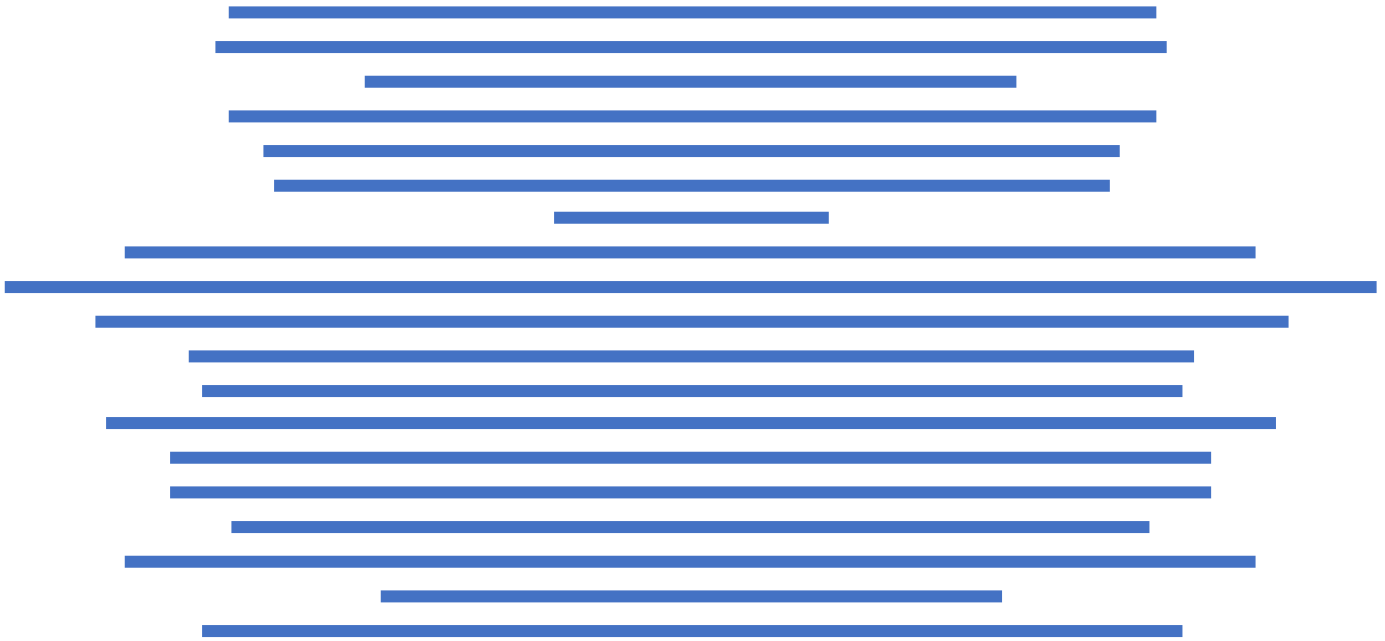
基因组组装

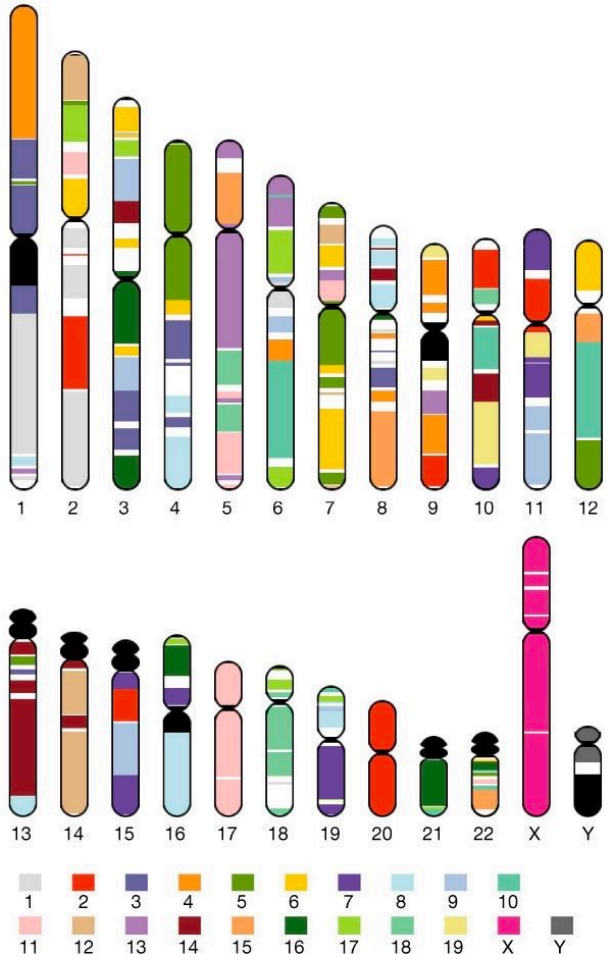
王鹏

仅用于内部学习

1

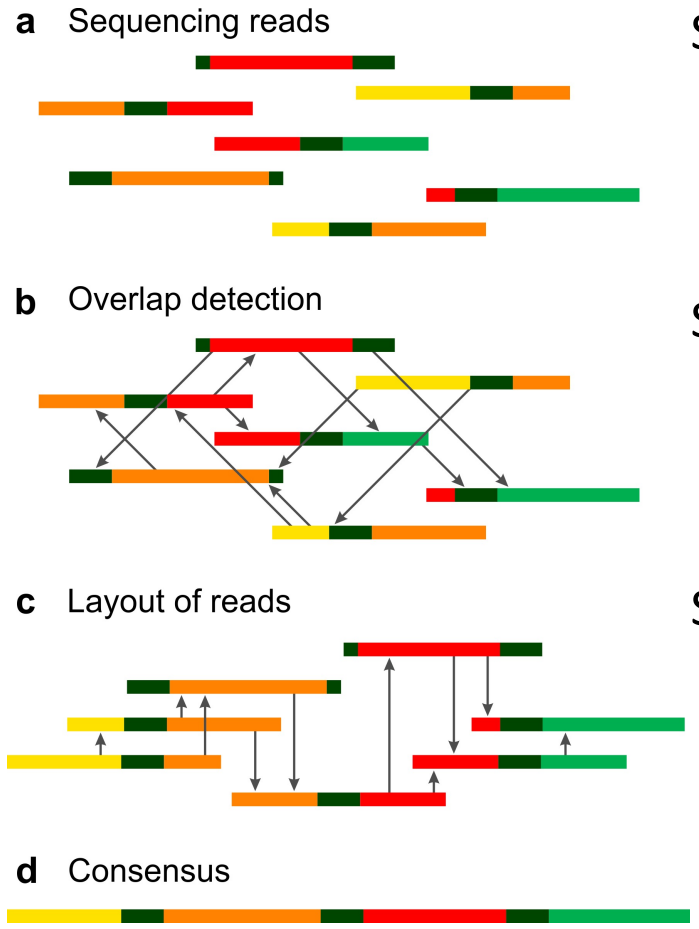
组装方法





Contigs/Scaffolds ≠ Genomes/Chromosome

Overlap layout consensus (OLC)



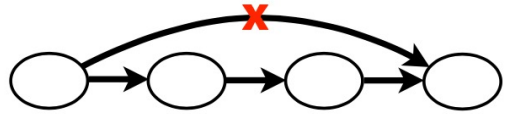
10.1371/journal.pone.0169662

Step1：Finding overlaps of all pairwise reads (overlap graph)

- Find the best alignment of a suffix of X to a prefix of Y
- set minimum overlap length

Step2： Bundle stretches of overlap graph into contigs (simplify graph)

- Remove transitively-inferrible edges
- Remove the non-branching stretches



Step3： Pick most likely nucleotide sequence (consensus contig)

```
TAGATTACACAGATTACTGA TTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAAACTA
TAG TTACACAGATTATTGACTTCATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA
```

✓ Make best use of the complete read information

× Overlap graph is big and messy

Software: Arachne; Celera Assembler; CAP3; Newbler; Phrap

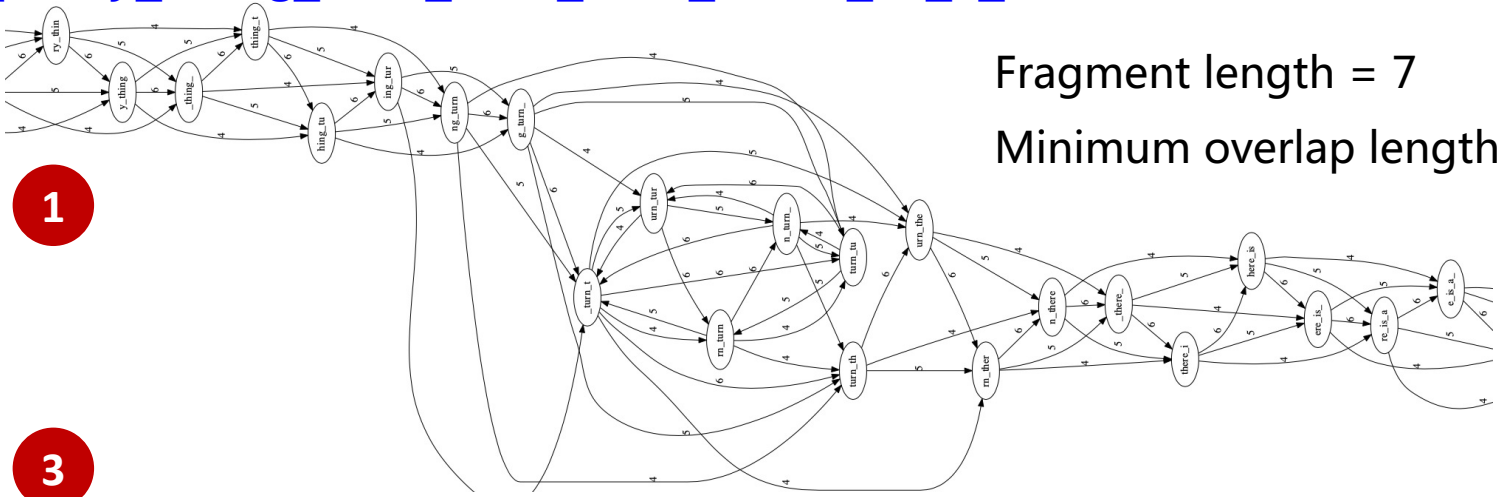
OLC-example

to_every_thing_turn_turn_turn_there_is_a_season

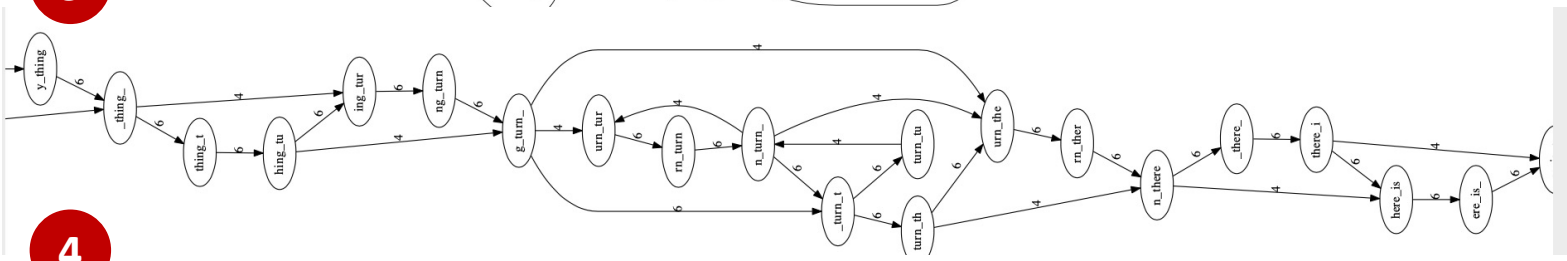
Fragment length = 7

Minimum overlap length = 4

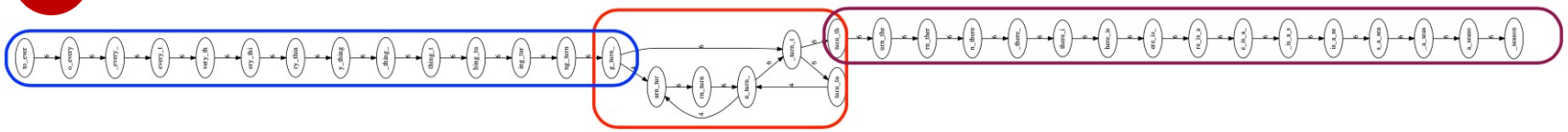
1



3



4



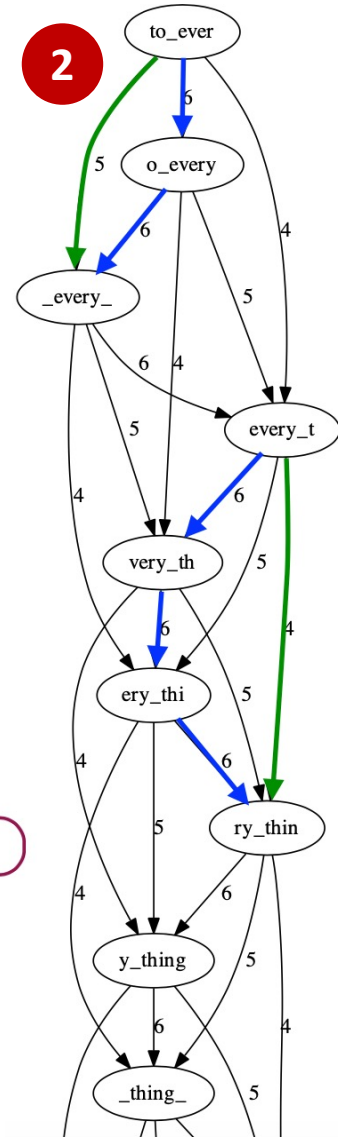
5

Contig 1
to_every_thing_turn_

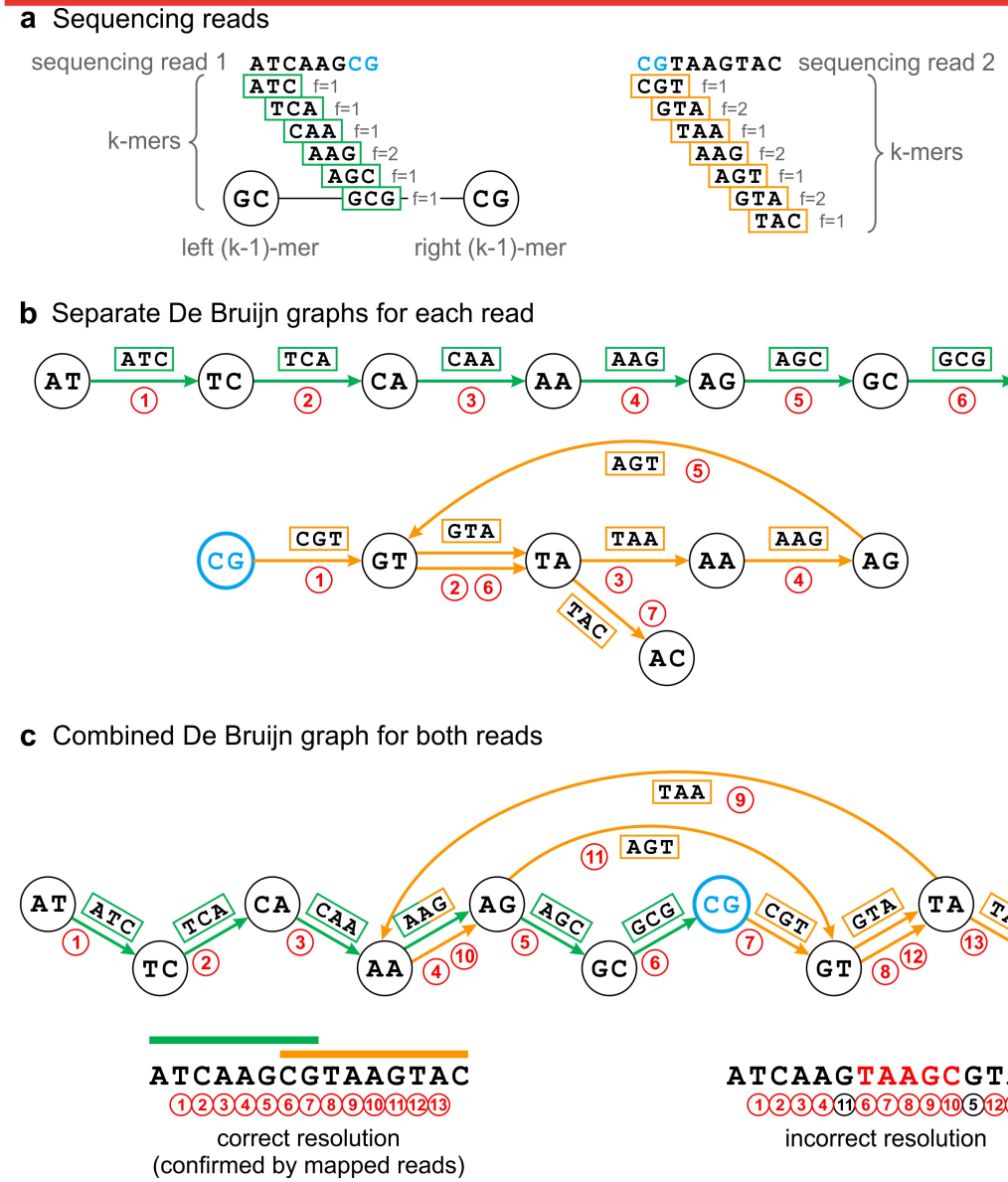
Contig 2
turn_there_is_a_season

Unresolvable repeat

2



De Bruijn graph



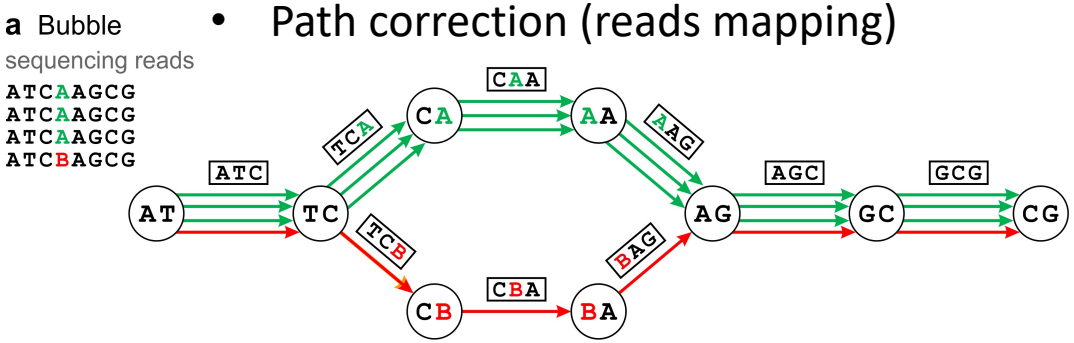
Step1: K-mer frequency (unique k-mer)

Step2: construction of de Bruijn graphs

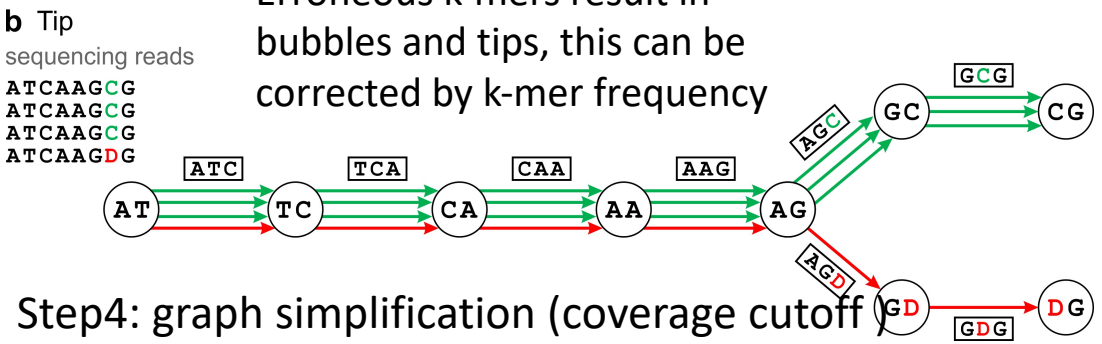
- Nodes: k-mers, edge: overlaps
- Repetitive regions

Step3: combination of de Bruijn graphs

- Path correction (reads mapping)



Erroneous k-mers result in bubbles and tips, this can be corrected by k-mer frequency



Step4: graph simplification (coverage cutoff)

Step5: extract contigs (break down from branch)

De Bruijn graph

De Bruijn graph approach is well suited for the short read lengths of most NGS data, since these are less likely to span all encountered repetitive regions.

☒ High memory (k-mer counting; error correction)

☒ Information stored within the reads is lost

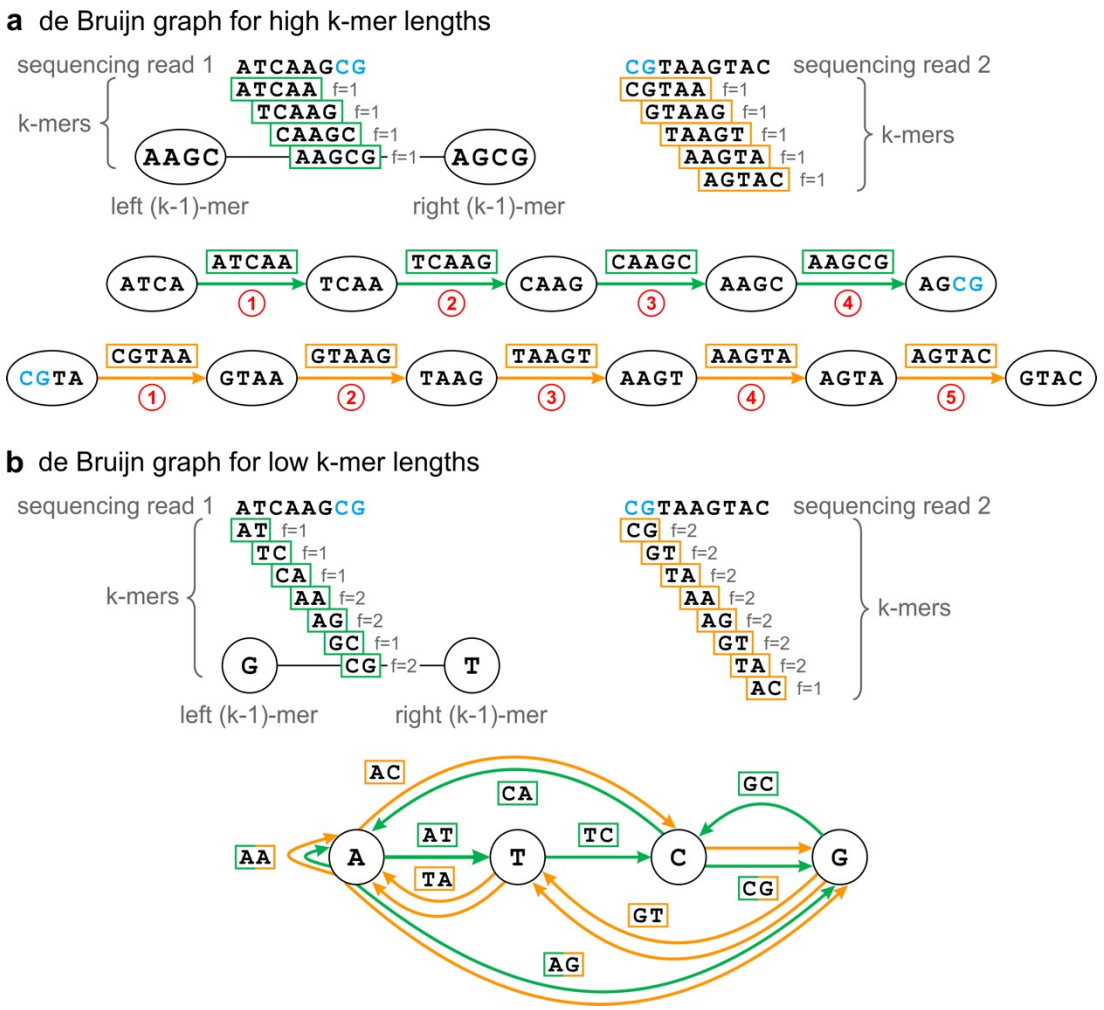
K-mer choice:

☐ Even or **uneven** ?(GAATTC , CTTAAG)

- **Palindromes** introduces additional branches to the graph and gives rise to potential inversions

☐ K-mer length (KMERGENIE; Velvet Advisor)

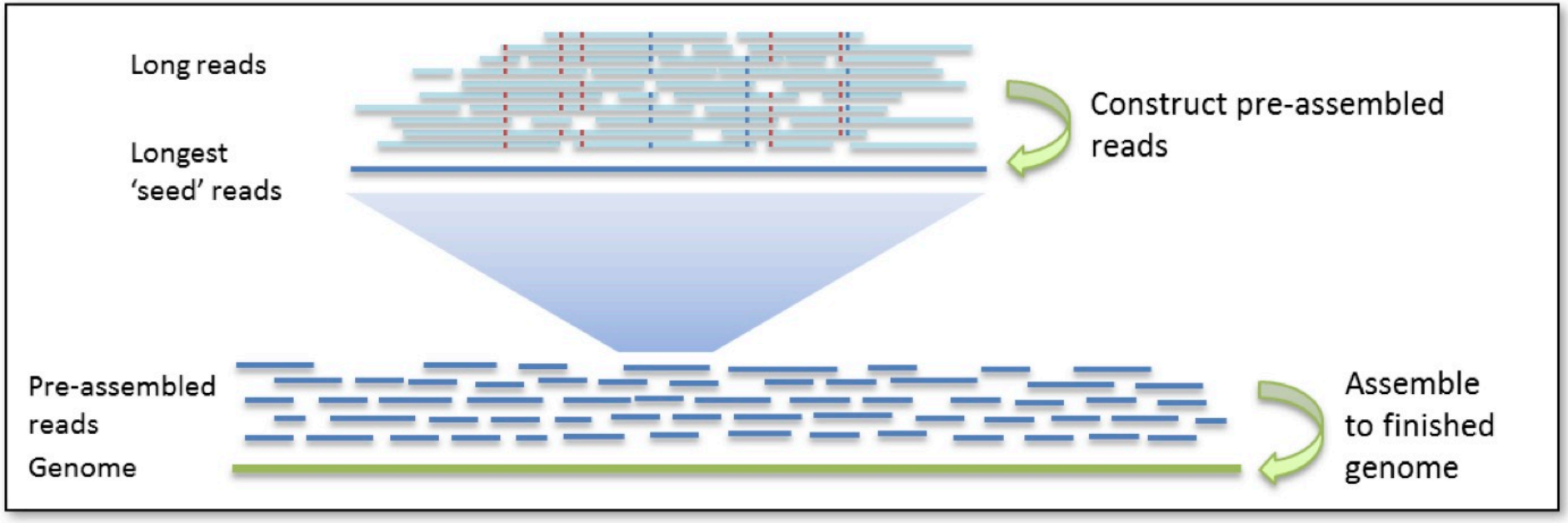
- Too large: less connections; better resolution of small repeats; low k-mer coverage
- Too small: more branches; fewer repetitive regions can be resolved; high k-mer coverage



Software: Velvet; Soapdenovo2; SPAdes; ABySS; IDBA-UD

Software1-Falcon

HGAP（ hierarchical genome assembly process ）

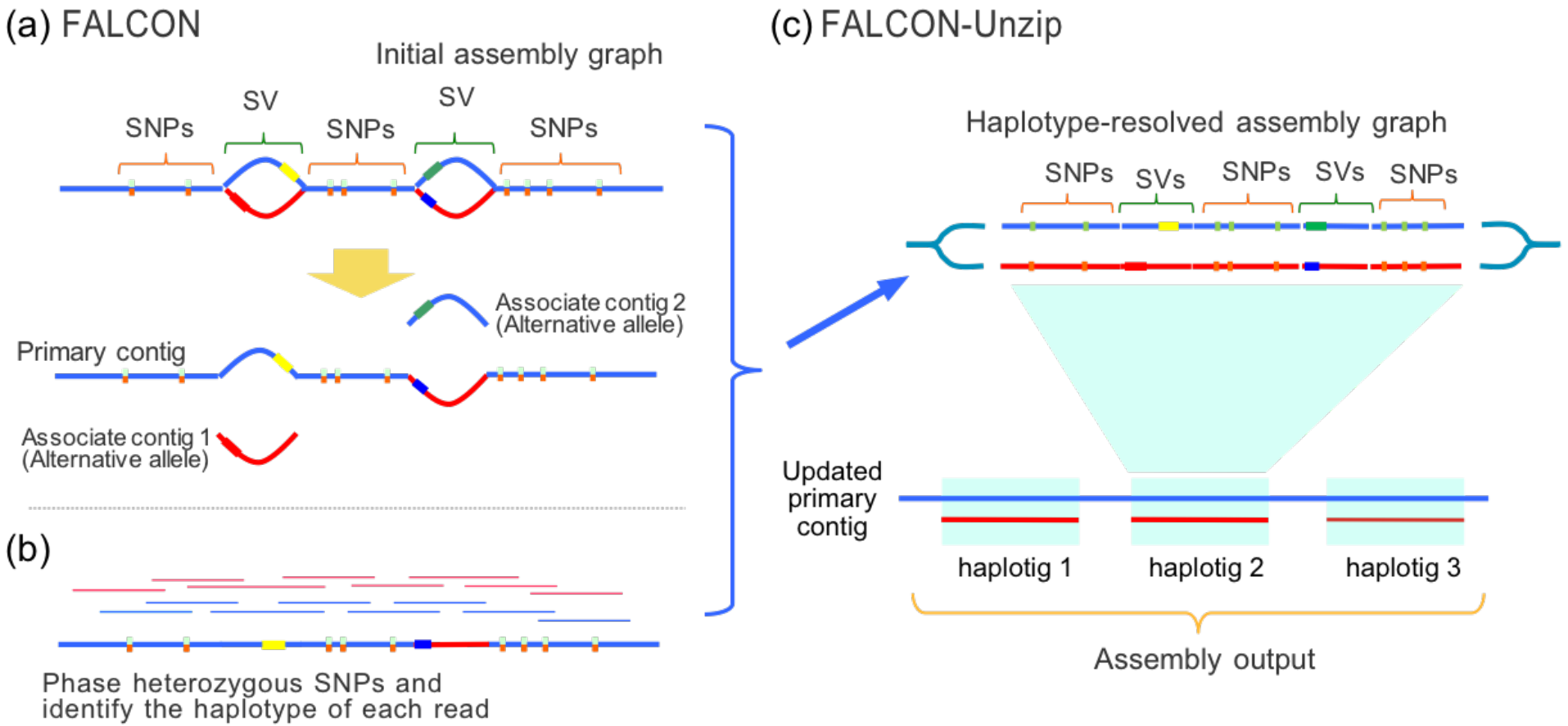


第一轮：选取最长序列作为种子序列，短read与之比对进行校正，得到高准确性的一致序列，对一致序列，根据低覆盖度区域进行split和trim，得到preads（ pre-assembly reads ）

第二轮：preads互相比对，得到基因组的contigs

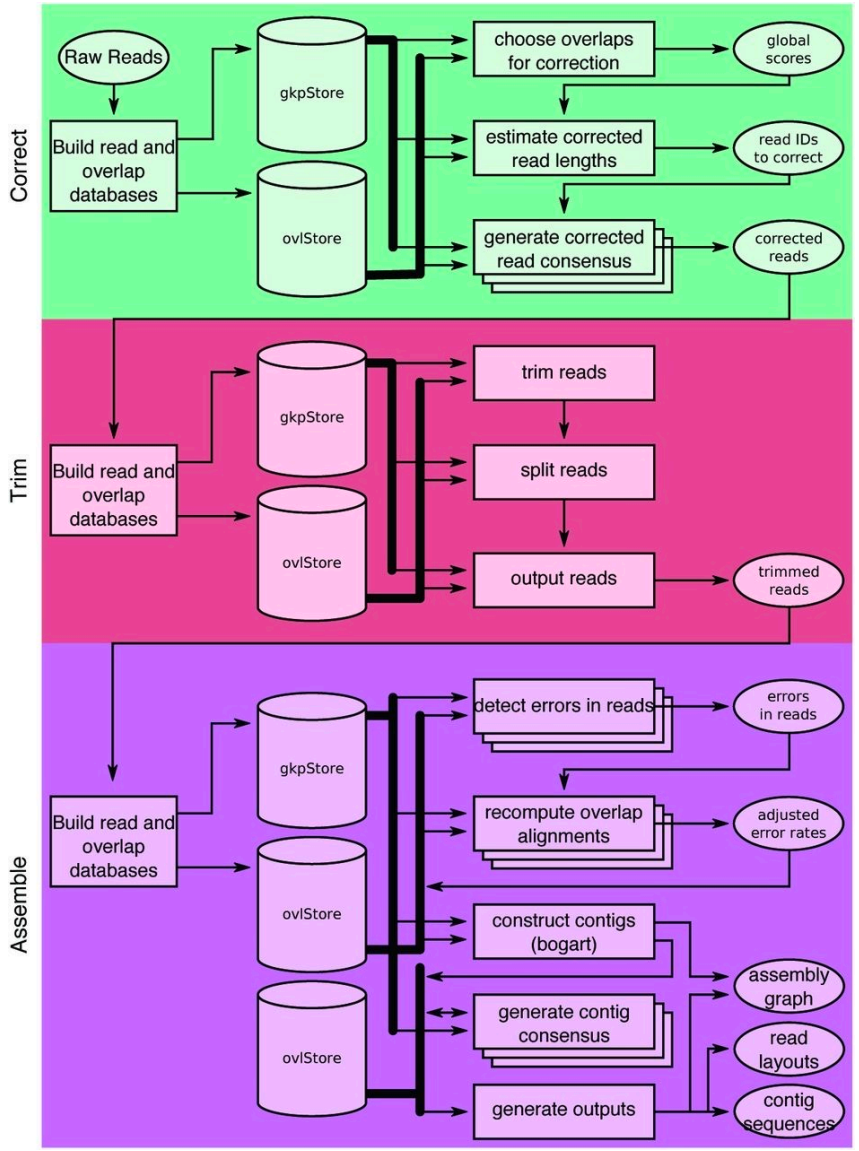
https://pb-falcon.readthedocs.io/en/latest/about.html#overview

Software1-Falcon



- 对于复杂基因组，则根据两套单倍型的SV和SNP变异信息，使用FALCON构建bubbles（图ab）
- 最后为了得到不同haplitig单倍型，对含有bubble的序列进行”解压缩”，即单倍体分型（图c）

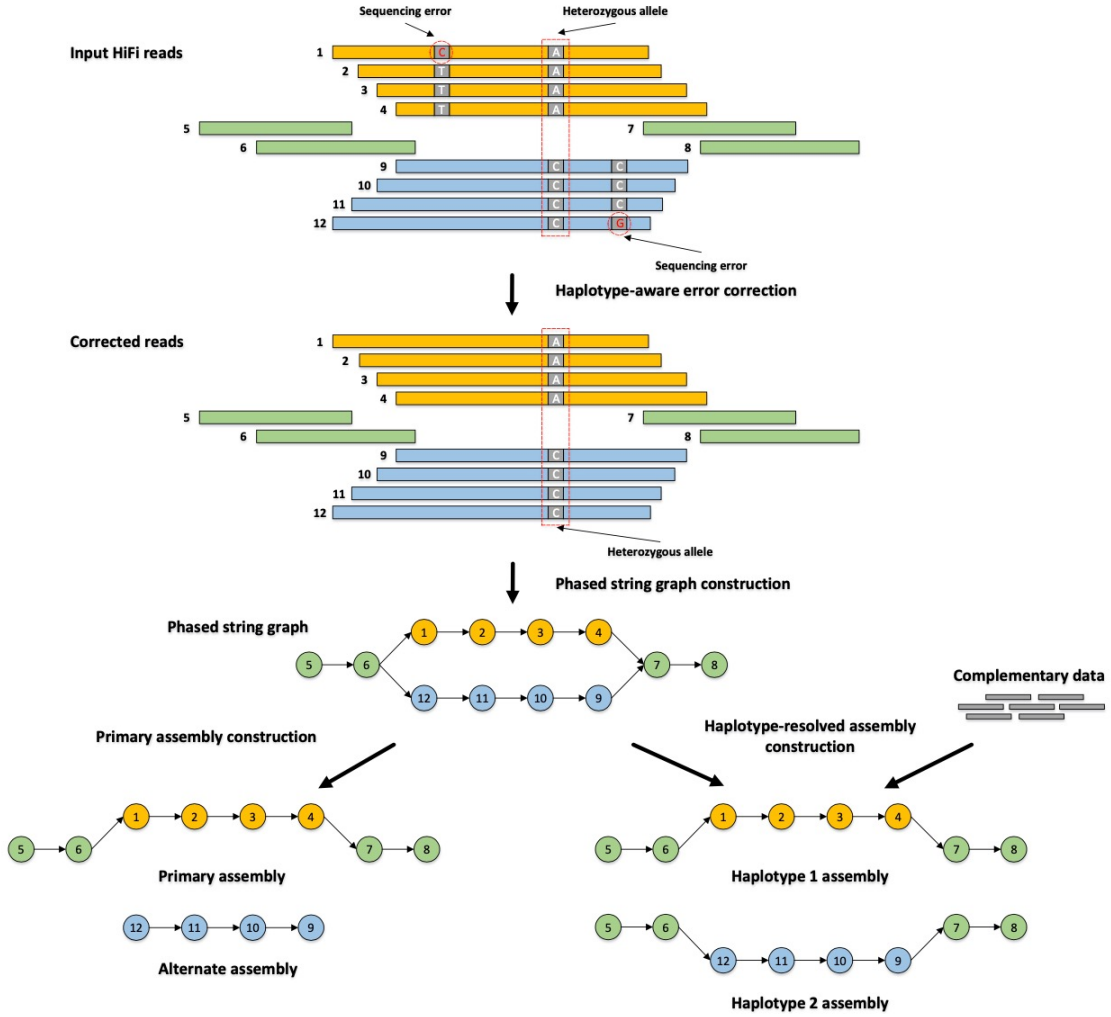
Software2-Canu(Hicanu)



Canu三步走：纠错、修剪和组装

- 纠错：所有的reads放到一起，确定overlap和纠正区域，根据概率对单碱基的错误进行校正；
- 修剪：根据overlap确定高质量区域，修剪低质量序列
- 组装：纠错和修剪后的高质量序列，使用OLC的方法进行组装，得到contigs

Software3-Hifiasm



1）All in all比对对reads进行纠错

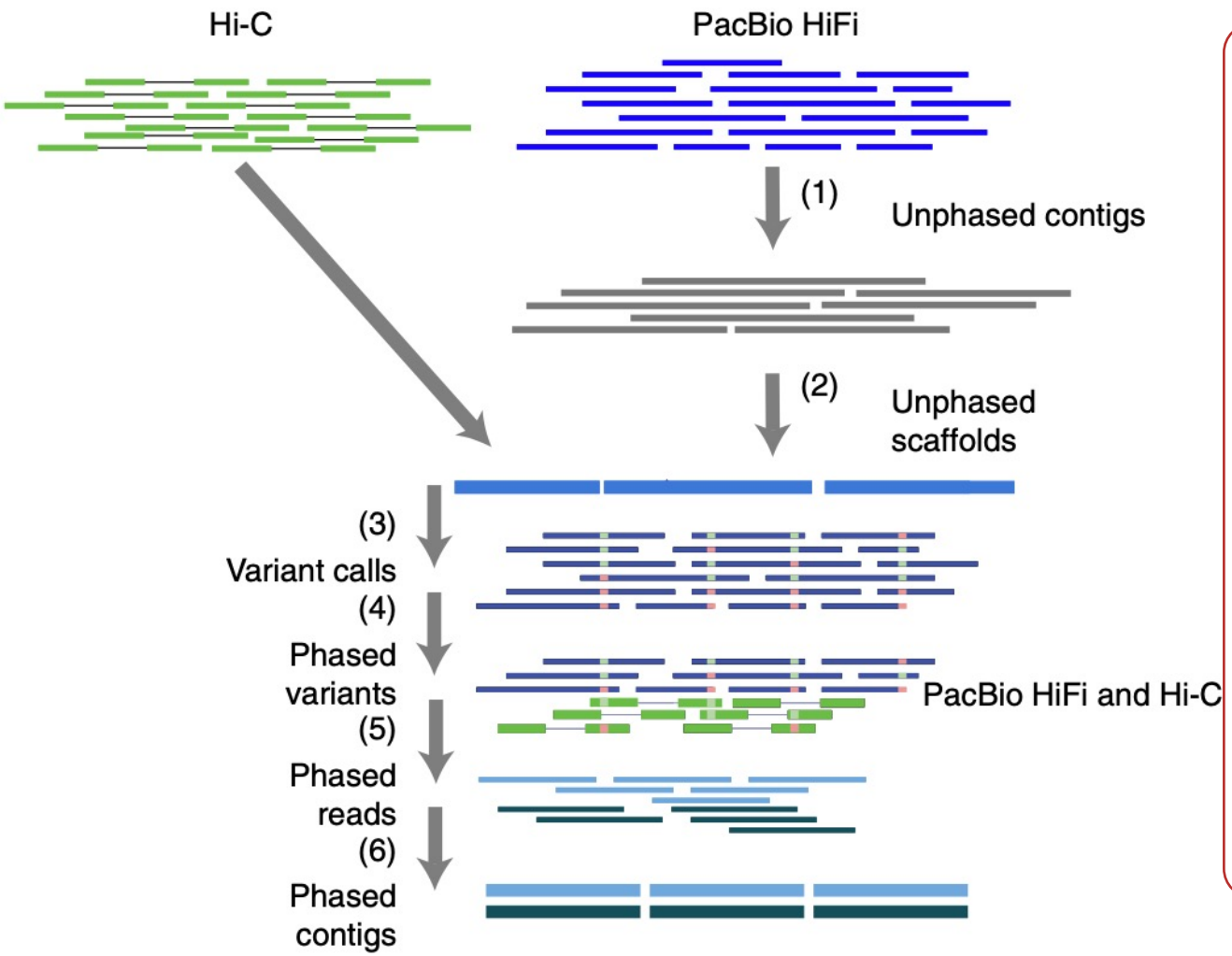
目标read纠错的规则

- 有效目标read：考虑与目标read有overlap的reads
- Informative position：两种碱基类型，且≥3reads支持
- inconsistent reads：与目标read有overlap，且具有informative位点，但该位点的碱基与目标reads不同
- 对某目标read，只选取consistent reads进行校正

2）根据overlap和informative位点构建phased graph（顶点：reads；边：overlap；bubble：杂合位点）

3）根据是否有亲本数据进行单倍体分型

Software4-DipAsm



Step1：使用Peregrine对HiFi数据进行组装，得到unphased contigs

Step2：使用HiRise或3D-DNA，通过HI-C数据对unphased contig分组和排序，得到unphased scaffolds

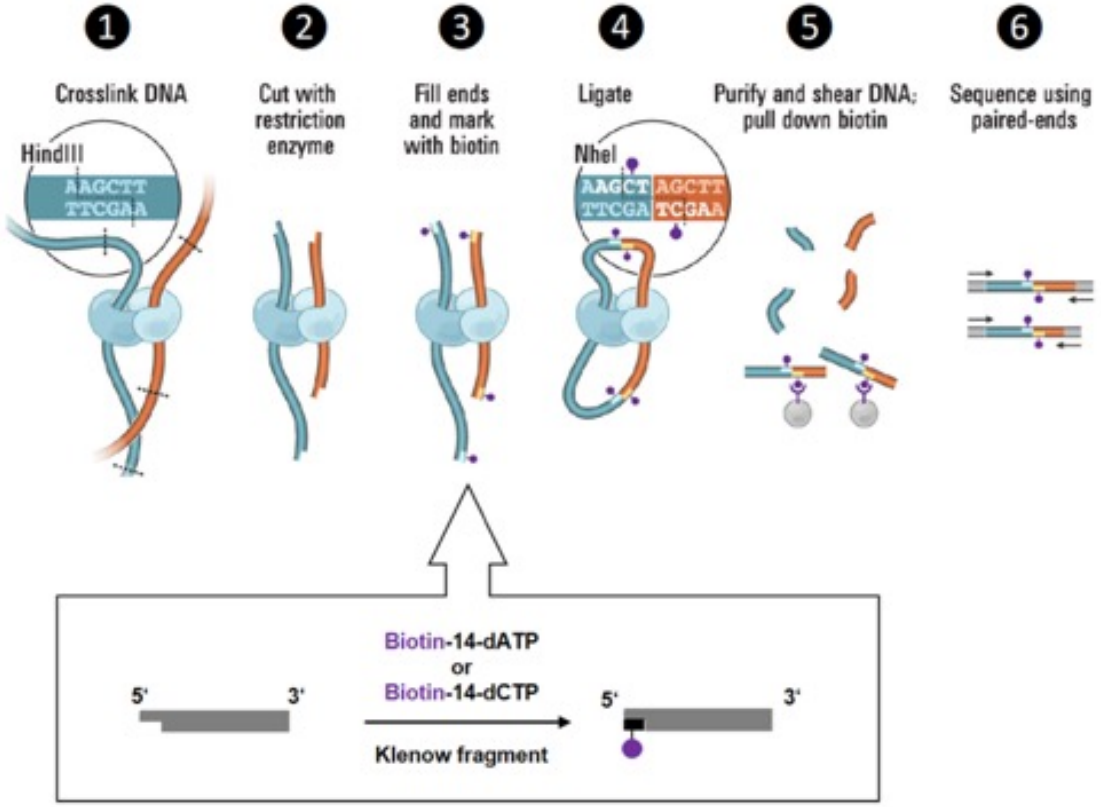
Step3：DeepVariant将HiFi reads映射到scaffolds上，获得杂合的SNPs信息

Step4：WhatsHap和HapCUT2处理HiFi和HI-C数据，获取分型的杂合SNPs

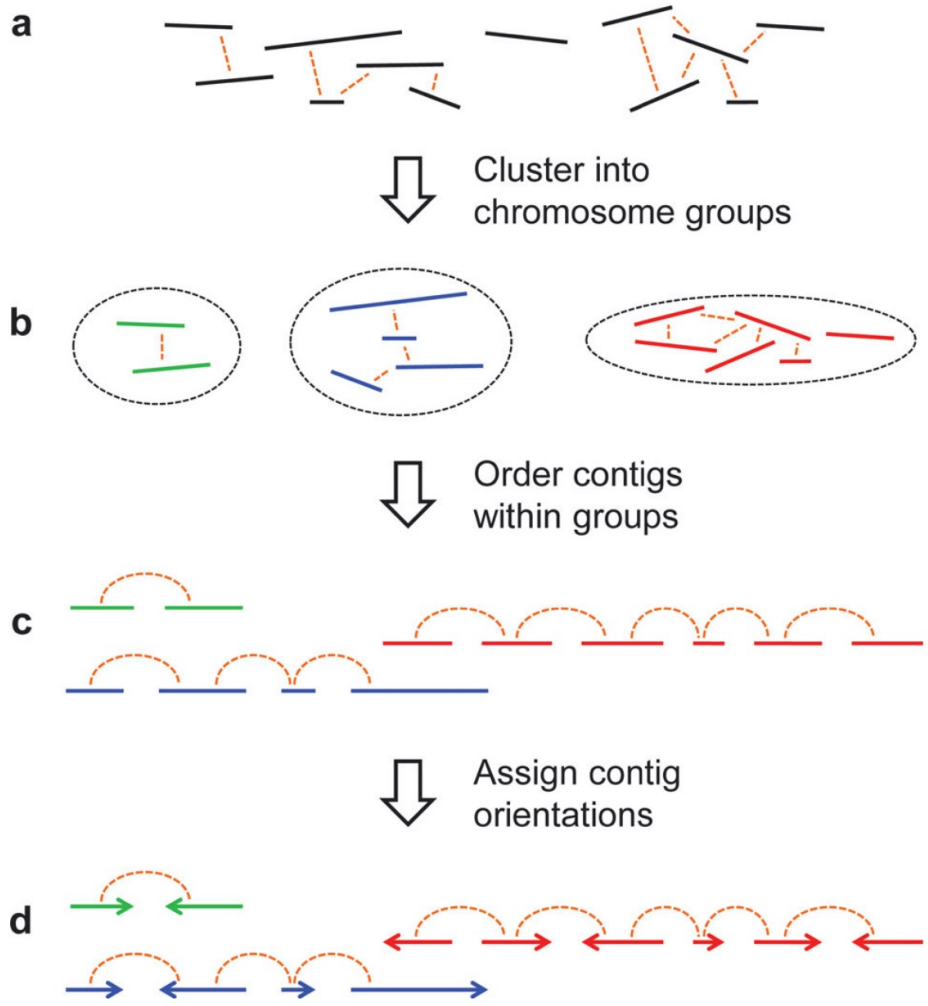
Step5：使用WhatsHap对reads进行分型

Step6：最后使用Peregrine对分型好的reads进行组装

Software5-LACHESIS (HI-C)



HI-C workflow



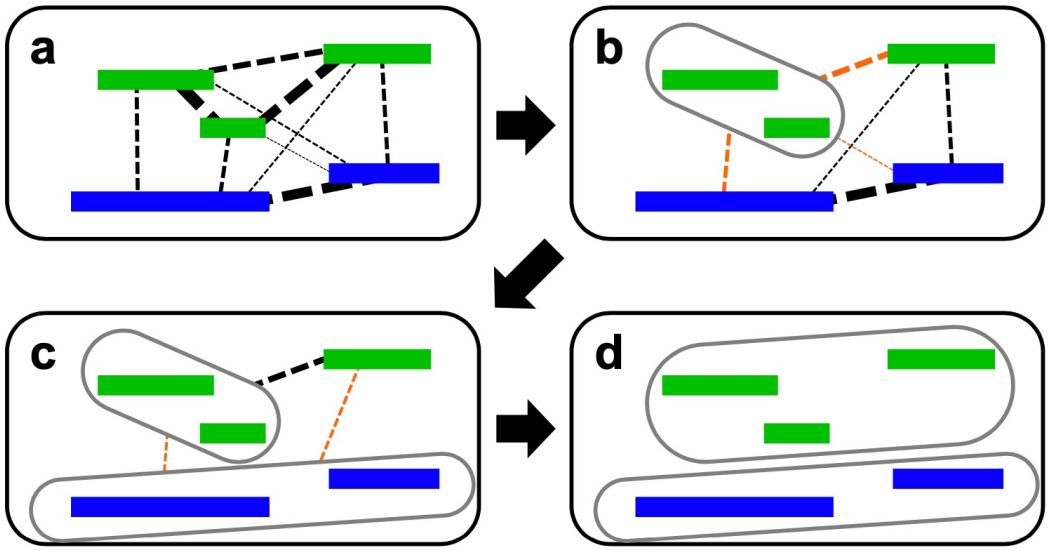
Assembly workflow

Software5-LACHESIS (HI-C)

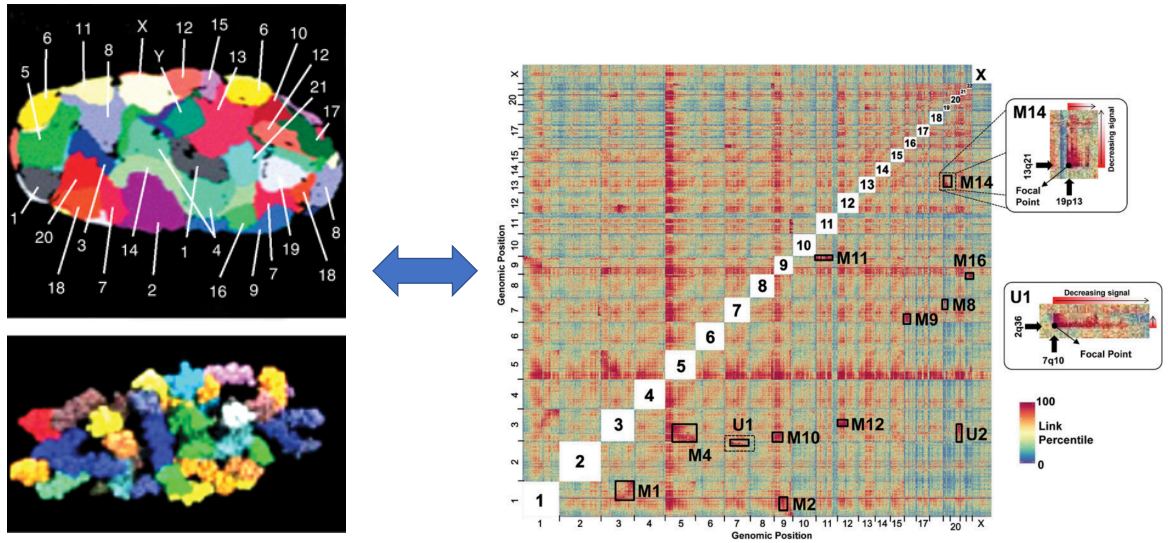
Step1：HI-C reads 比对

- Paired reads 比对在酶切位点的>500bp
- 去掉不唯一的比对reads
- 只保留paired reads比对到contig的reads

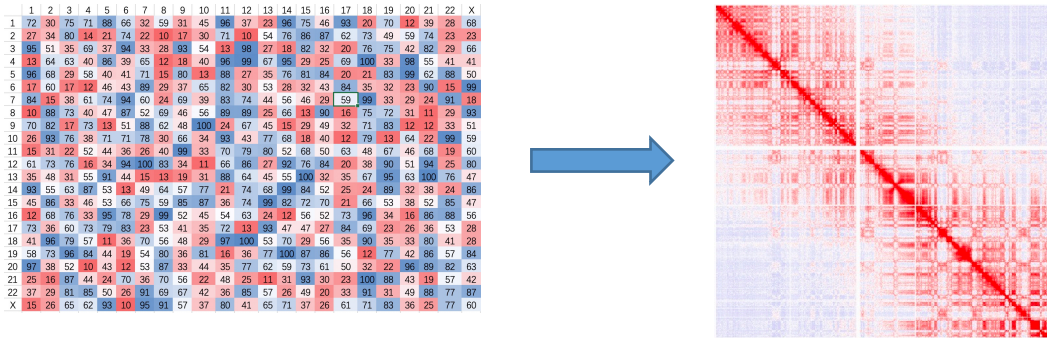
Step2：contigs聚类到染色体水平



!!!染色体内的互作要远大于染色体间的互作!!!

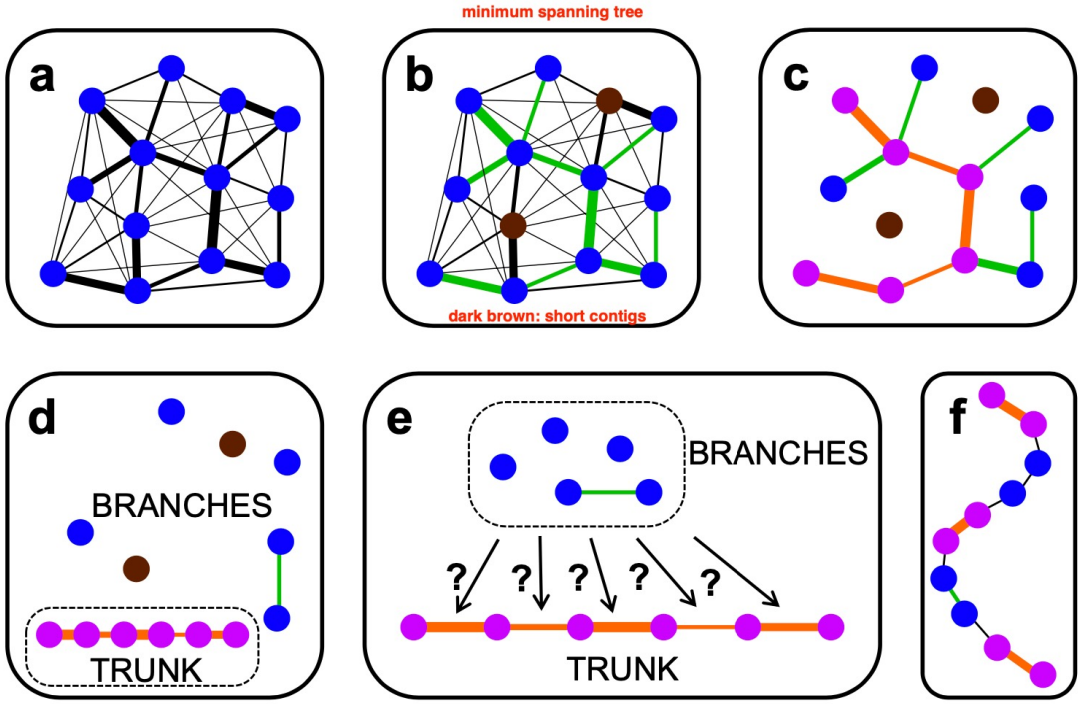


使用层次聚类的算法对互作信息（average-**linkage** metric）进行聚类



Software5-LACHESIS (HI-C)

Step3： 对contigs进行排序



① 构建网络图（节点代表contig，边代表连接，粗细代表权重）

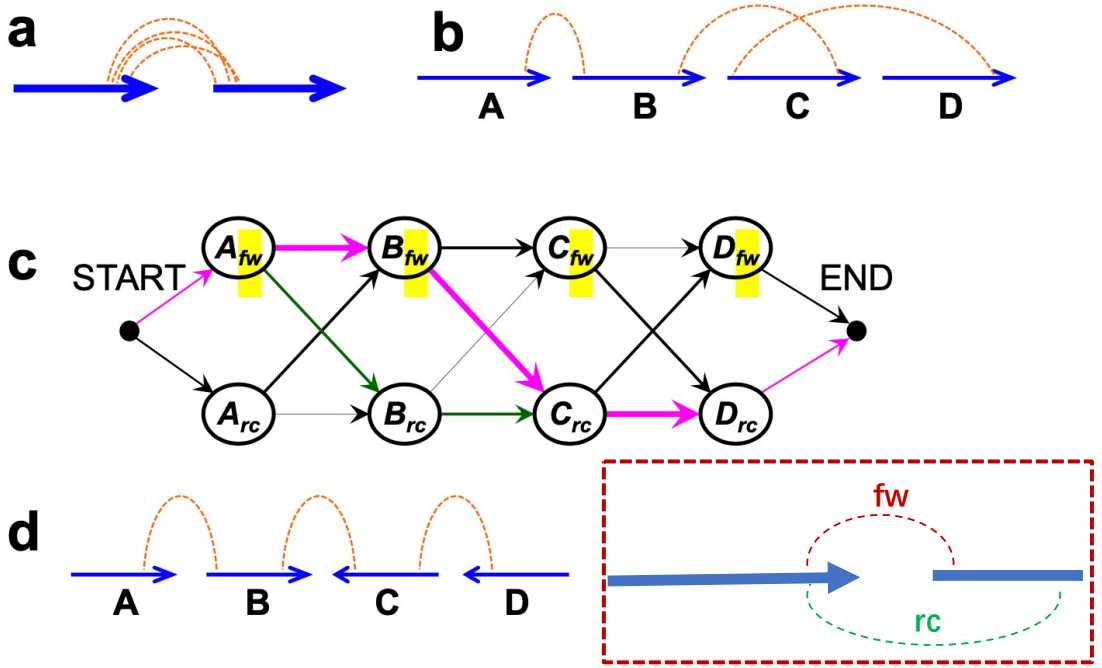
② 简化聚类图，得到生成树（spanning tree）

③ 从生成树中查找最长路径（洋红色），过滤掉较短的contig

④ 分离出最长路径（trunck）和未连接的contig（branches）

⑤ 将branches中的contig按长短依次插入到trunk中

Step4： 对contigs定向



① 构建加权有向无环图（WDAG）

② 针对contig的两种排向，分别计算入离权重差fw}和备选contig的入离权重差 {rc}；如果fw > rc，正向连接，如果rw < rc，反向连接

③ 构建染色体水平的最佳路径

BerryGenomics
贝瑞基因

Thank You!

官 方 网 站

官 方 微 信

www.berrygenomics.com