



BerryGenomics  
贝瑞基因



# 基因组组装

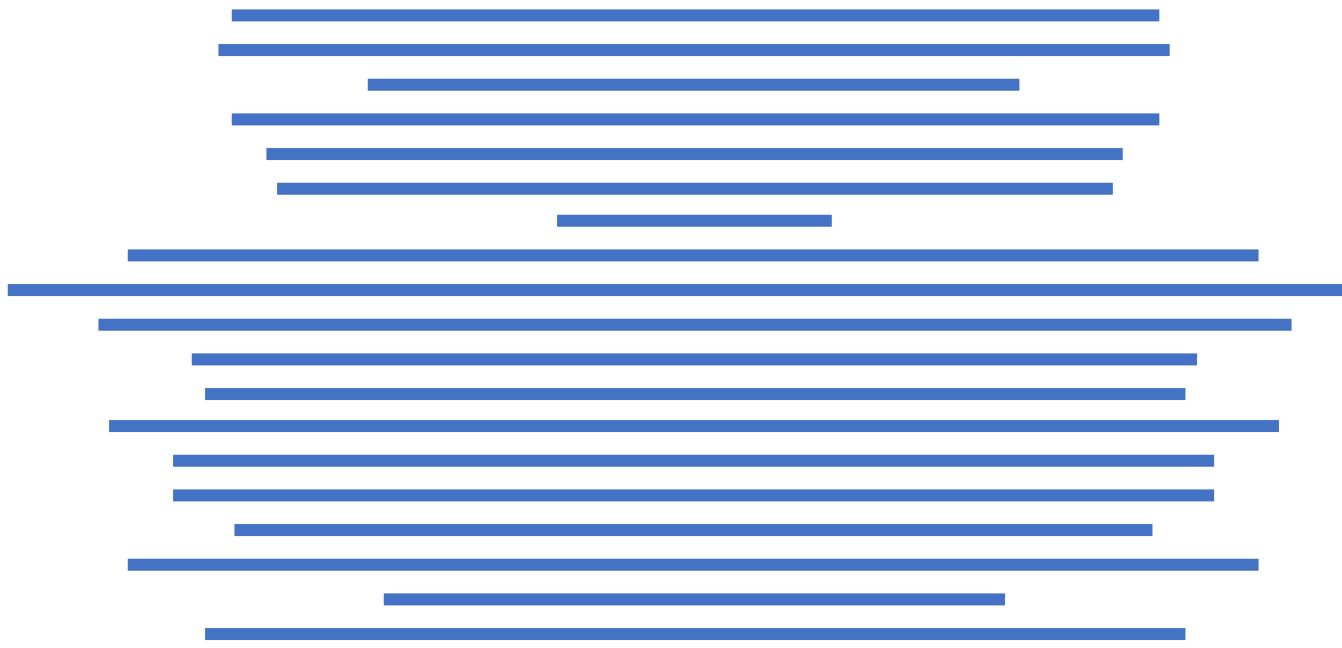
王鹏

仅用于内部学习

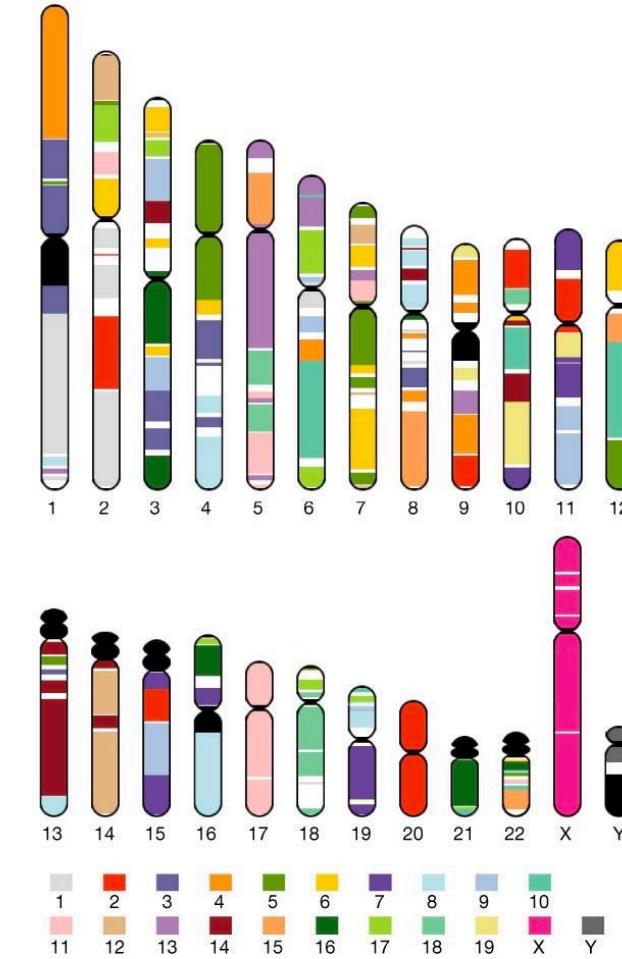
1



# 组装方法

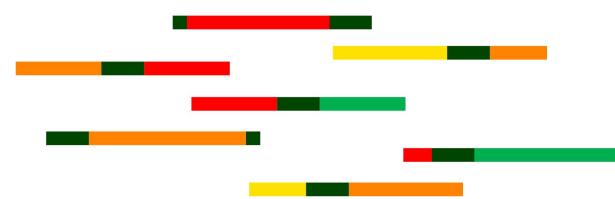


**Contigs/Scaffolds ≠ Genomes/Chromosome**

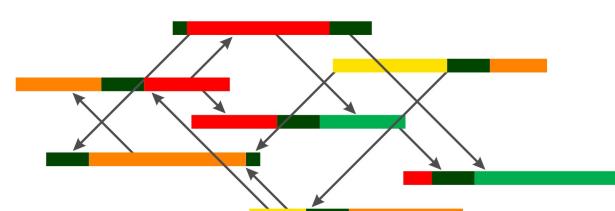


# Overlap layout consensus (OLC)

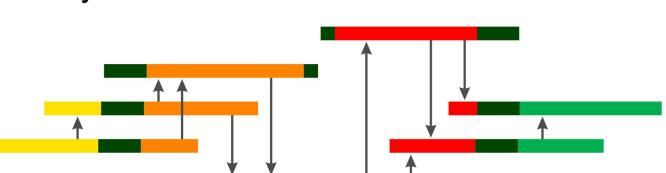
a Sequencing reads



b Overlap detection



c Layout of reads



d Consensus

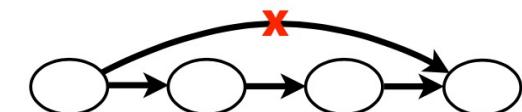


Step1 : Finding overlaps of all pairwise reads (overlap graph)

- 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  
TAGATTACACAGATTACTGACTTGATGGCGTAACTA  
TAG TTACACAGATTATGACTTCATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA  
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA

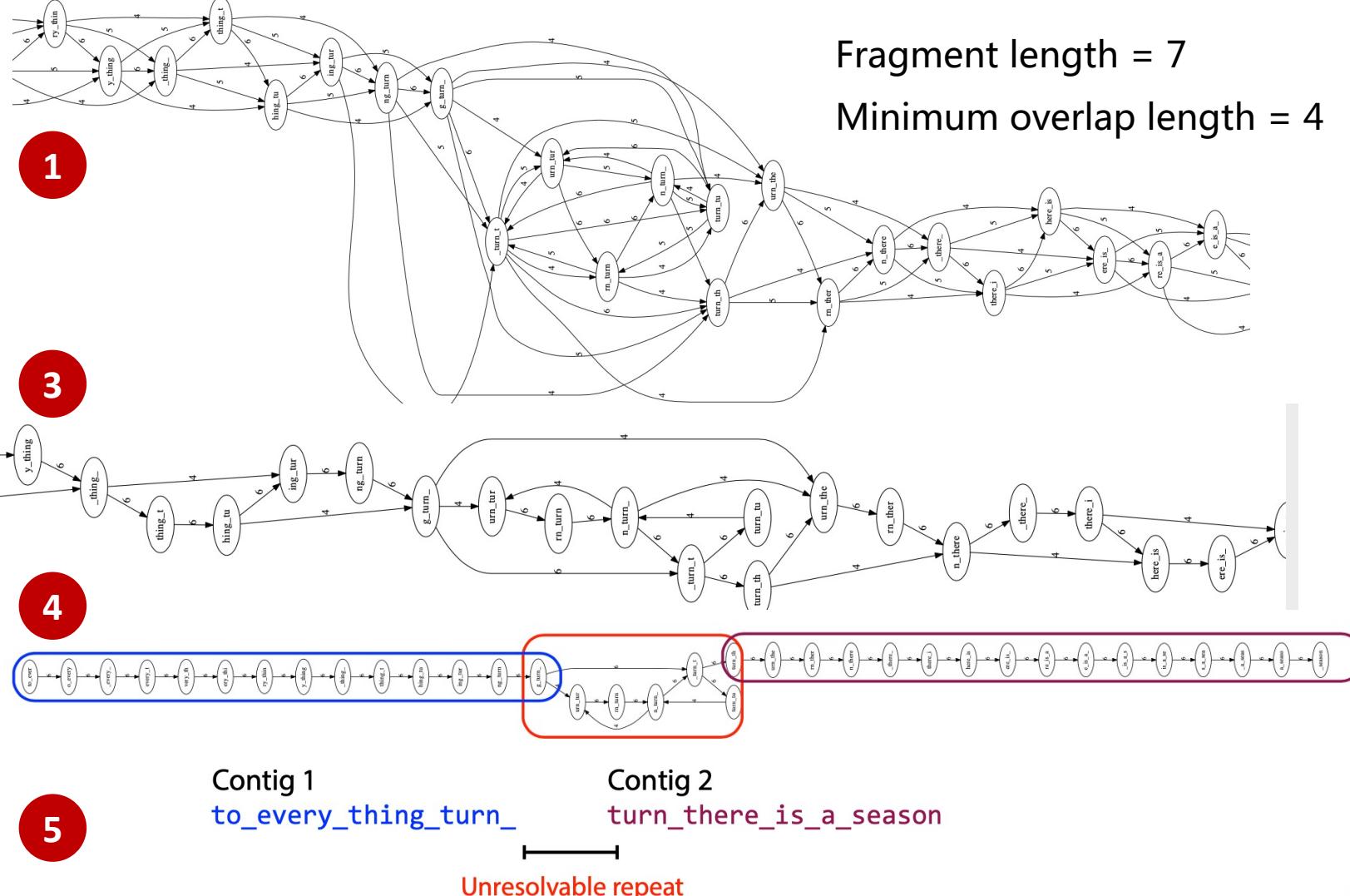
TAGATTACACAGATTACTGACTTGATGGCGTAA CTA

- ✓ Make best use of the complete read information
- ✗ Overlap graph is big and messy

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

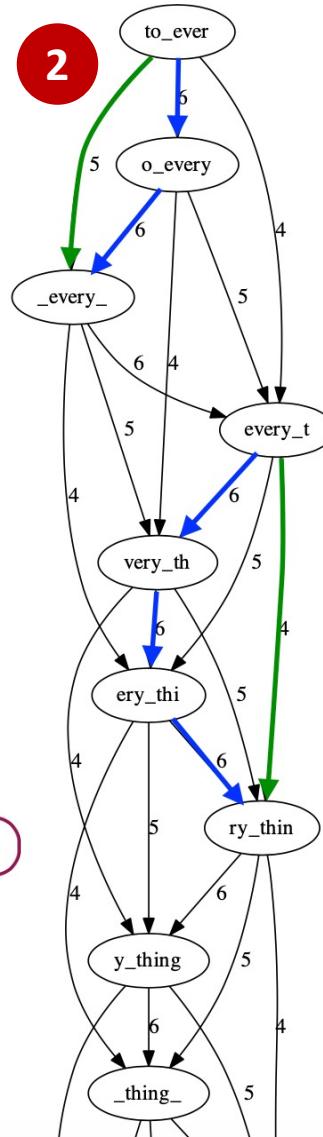
# OLC-example

to\_every\_thing\_turn\_turn\_turn\_there\_is\_a\_season



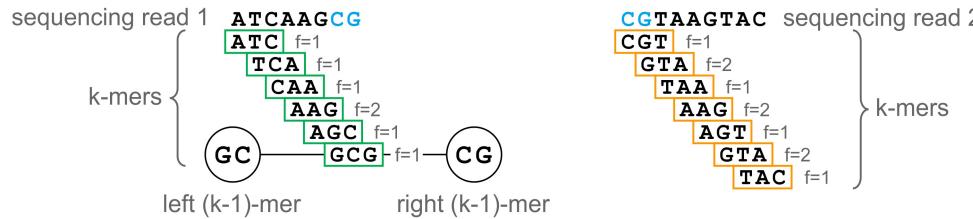
Fragment length = 7

Minimum overlap length = 4

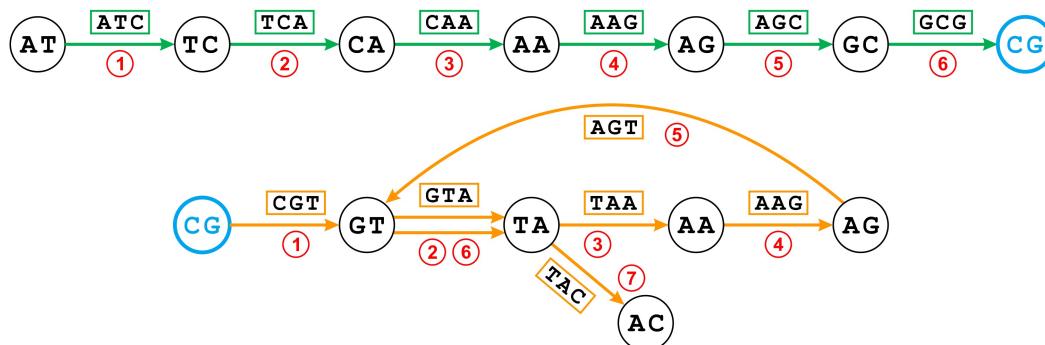


# De Bruijn graph

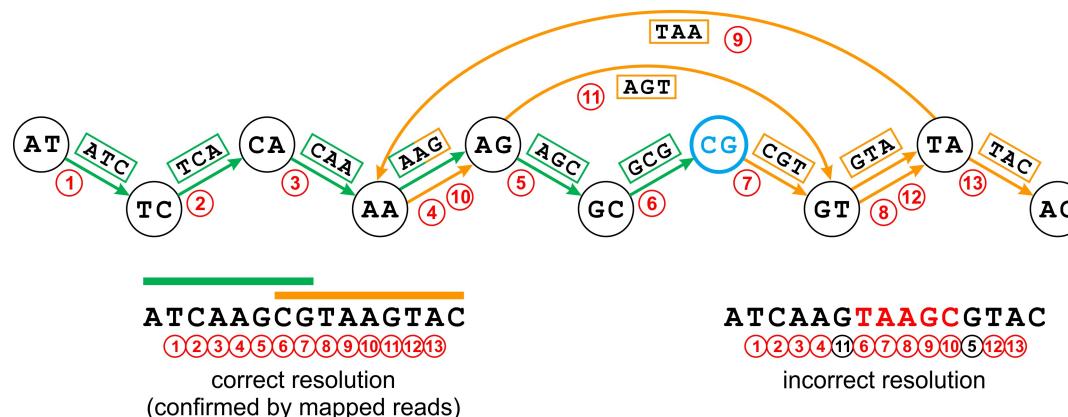
## a Sequencing reads



## b Separate De Bruijn graphs for each read



## c Combined De Bruijn graph for both reads



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

Step2: construction of de Bruijn graphs

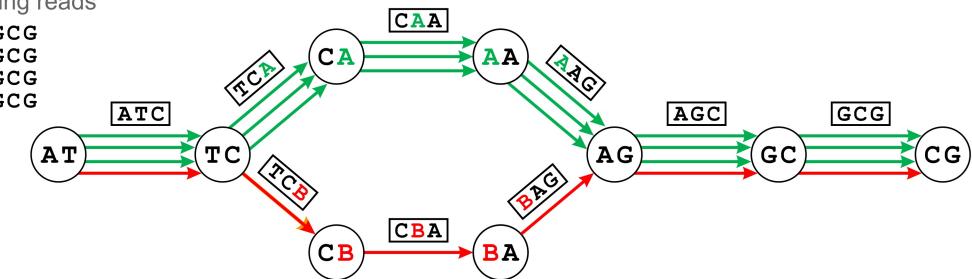
- Nodes: k-mers, edge: overlaps
- Repetitive regions

Step3: combination of de Bruijn graphs

- Path correction (reads mapping)

## a Bubble

sequencing reads:  
 ATCAAGCG  
 ATCAAGCG  
 ATCAAGCG  
 ATCBAGCG

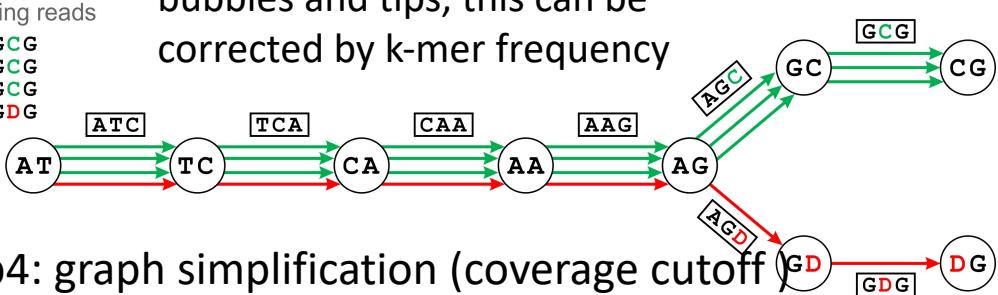


Erroneous k-mers result in

bubbles and tips, this can be  
 corrected by k-mer frequency

## b Tip

sequencing reads:  
 ATCAAGCG  
 ATCAAGCG  
 ATCAAGCG  
 ATCAAGDG



Step4: graph simplification (coverage cutoff)

Step5: extract contigs (break down from branch)

# De Bruijn graph

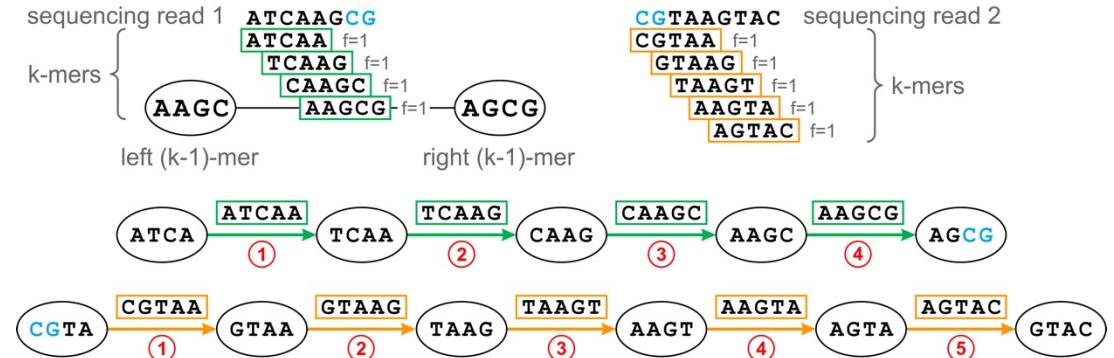
De Bruijn graph approach is well suited for the short read lengths of most NGS data, since these are less likely to span all encountered repetitive regions.

- ☒ High memory (k-mer counting; error correction)
- ☒ Information stored within the reads is lost

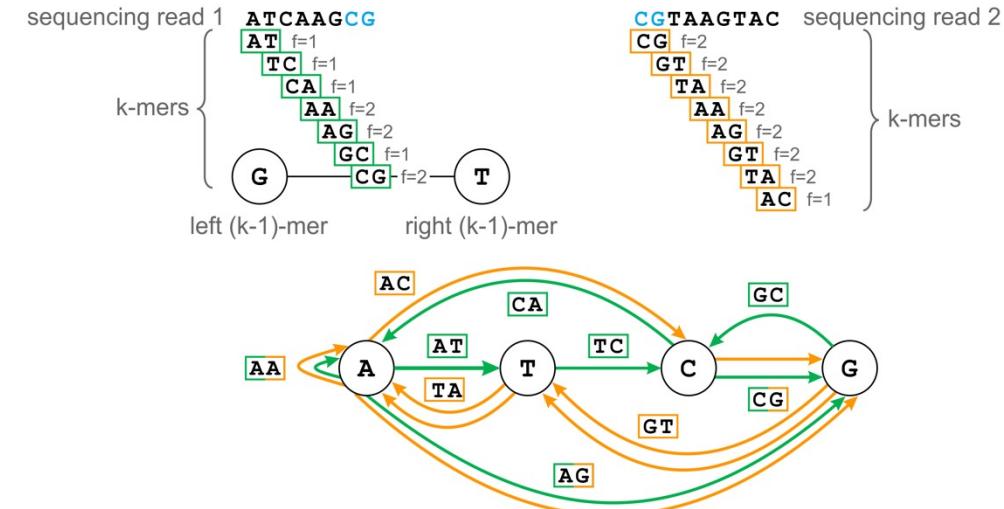
## K-mer choice:

- ☐ Even or **uneven** ?(GAATTCC , 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

a de Bruijn graph for high k-mer lengths

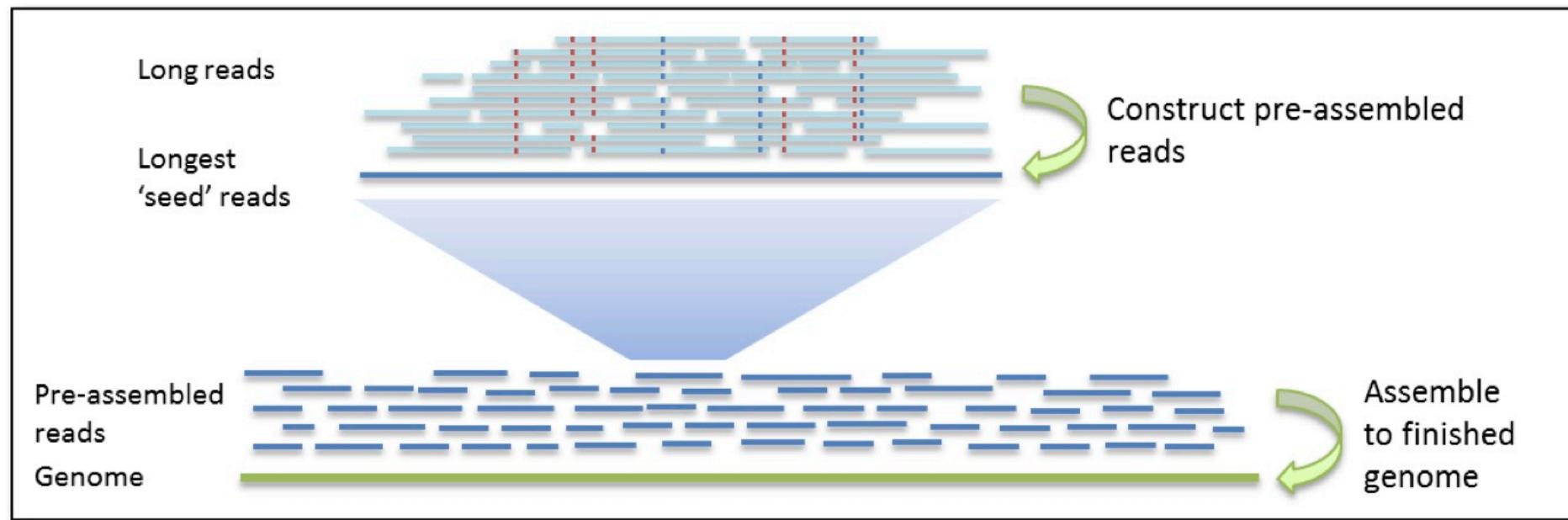


b de Bruijn graph for low k-mer lengths



**Software:** Velvet; Soapdenovo2; SPAdes; ABYSS; IDBA-UD

# Software1-Falcon

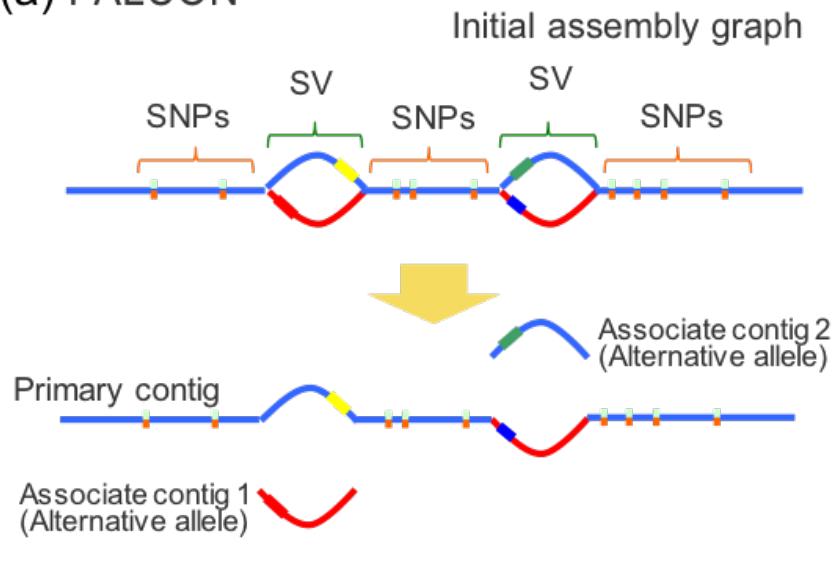


第一轮：选取最长序列作为种子序列，短read与之比对进行校正，得到高准确性的一致序列，对一致序列，根据低覆盖度区域进行split和trim，得到preads ( pre-assembly reads )

第二轮：preads互相比对，得到基因组的contigs

# Software1-Falcon

(a) FALCON

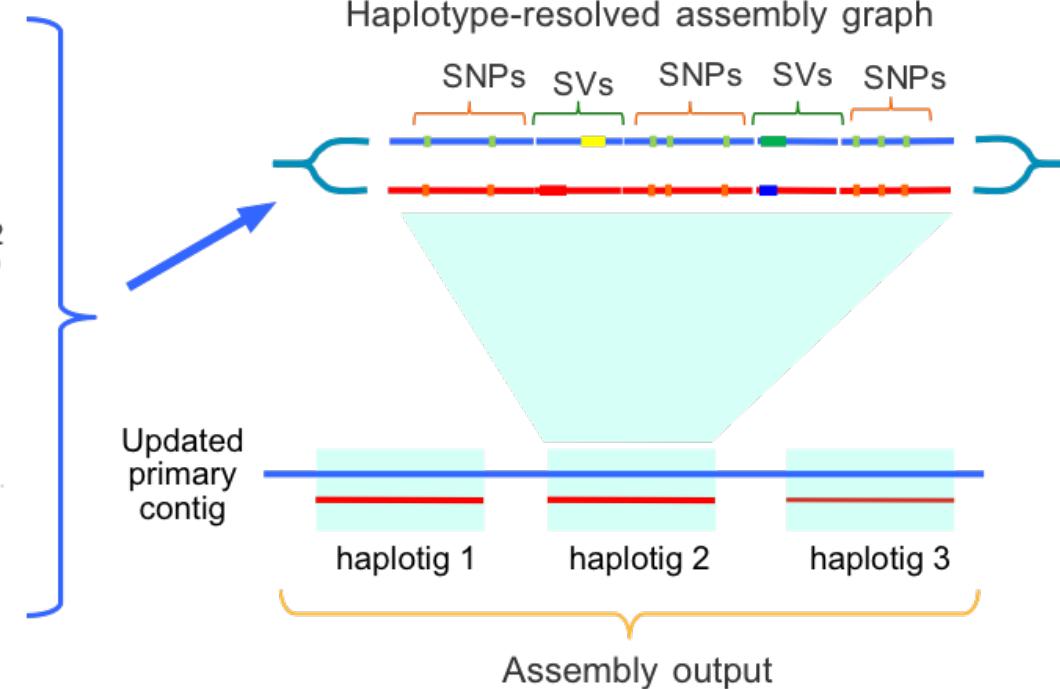


(b)



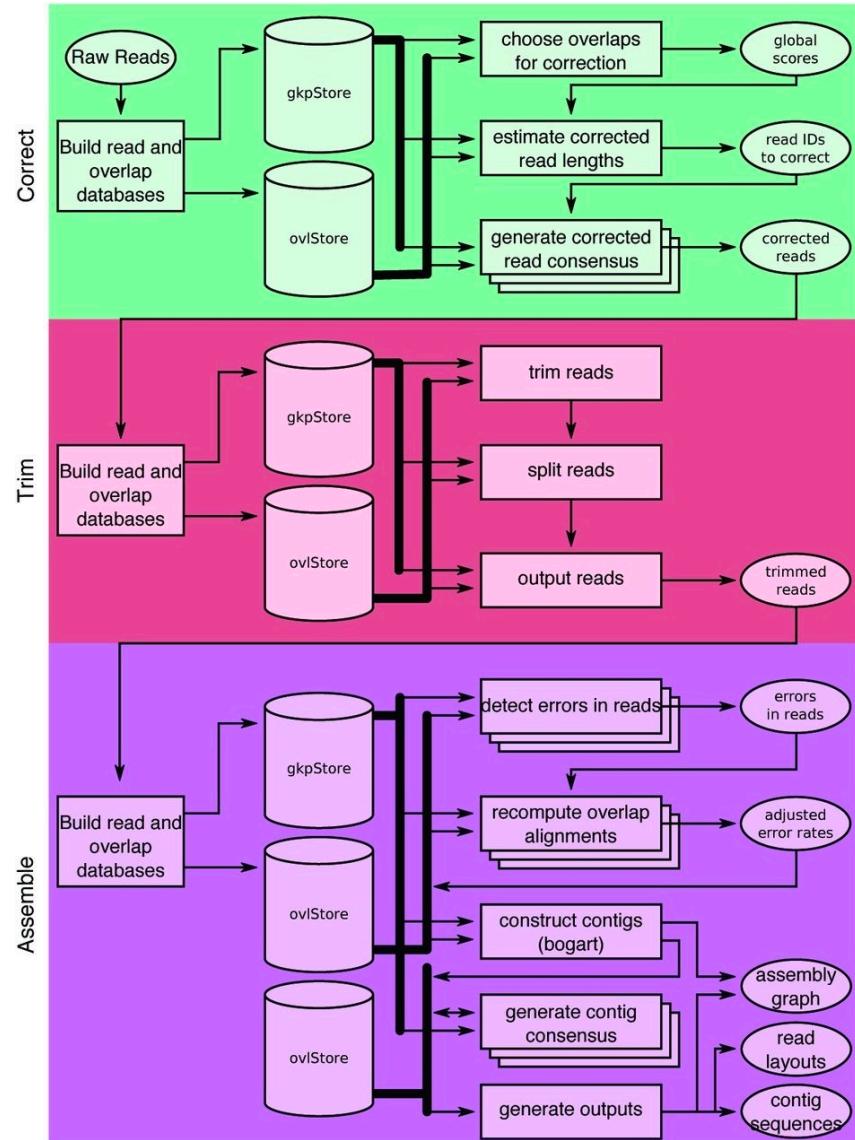
Phase heterozygous SNPs and  
identify the haplotype of each read

(c) FALCON-Unzip



- 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）

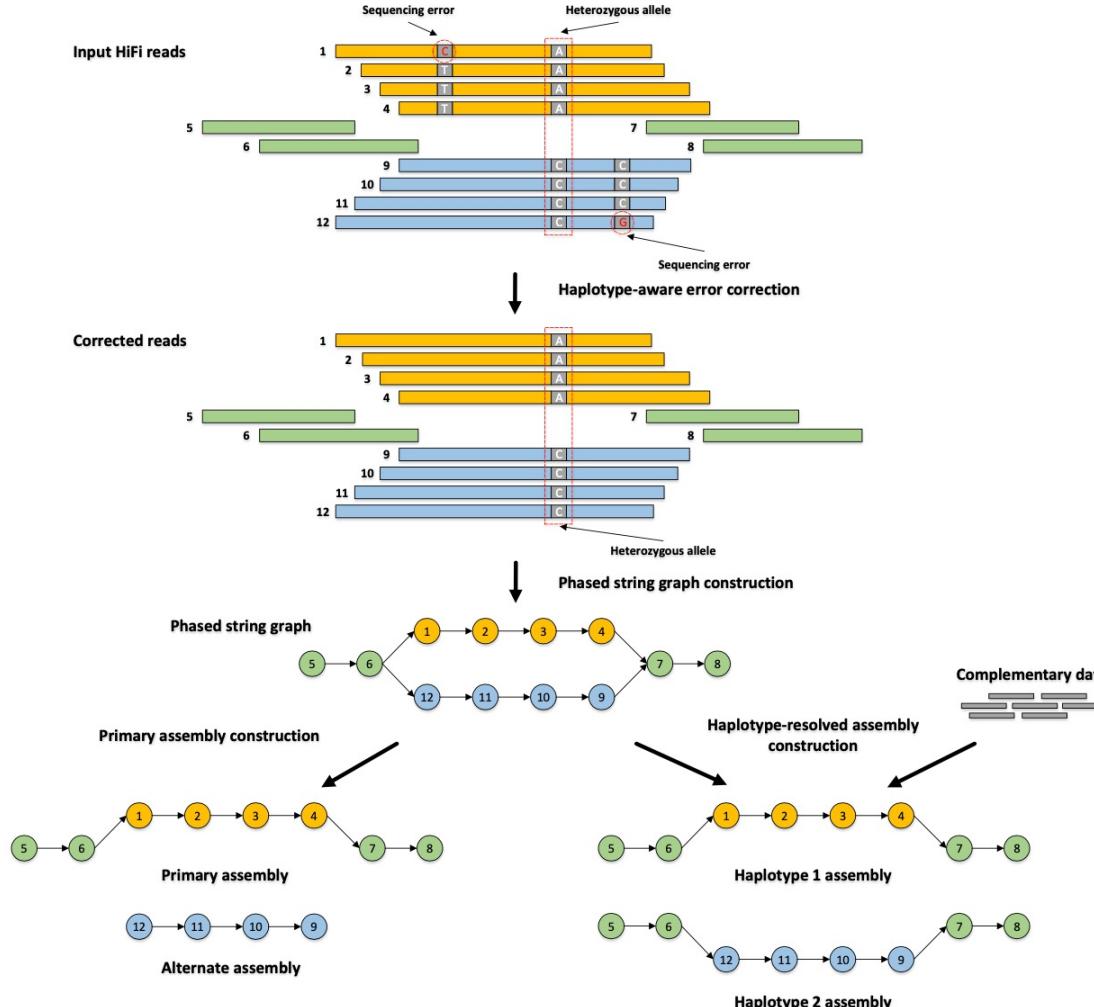
# Software2-Canu(Hicanu)



## 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

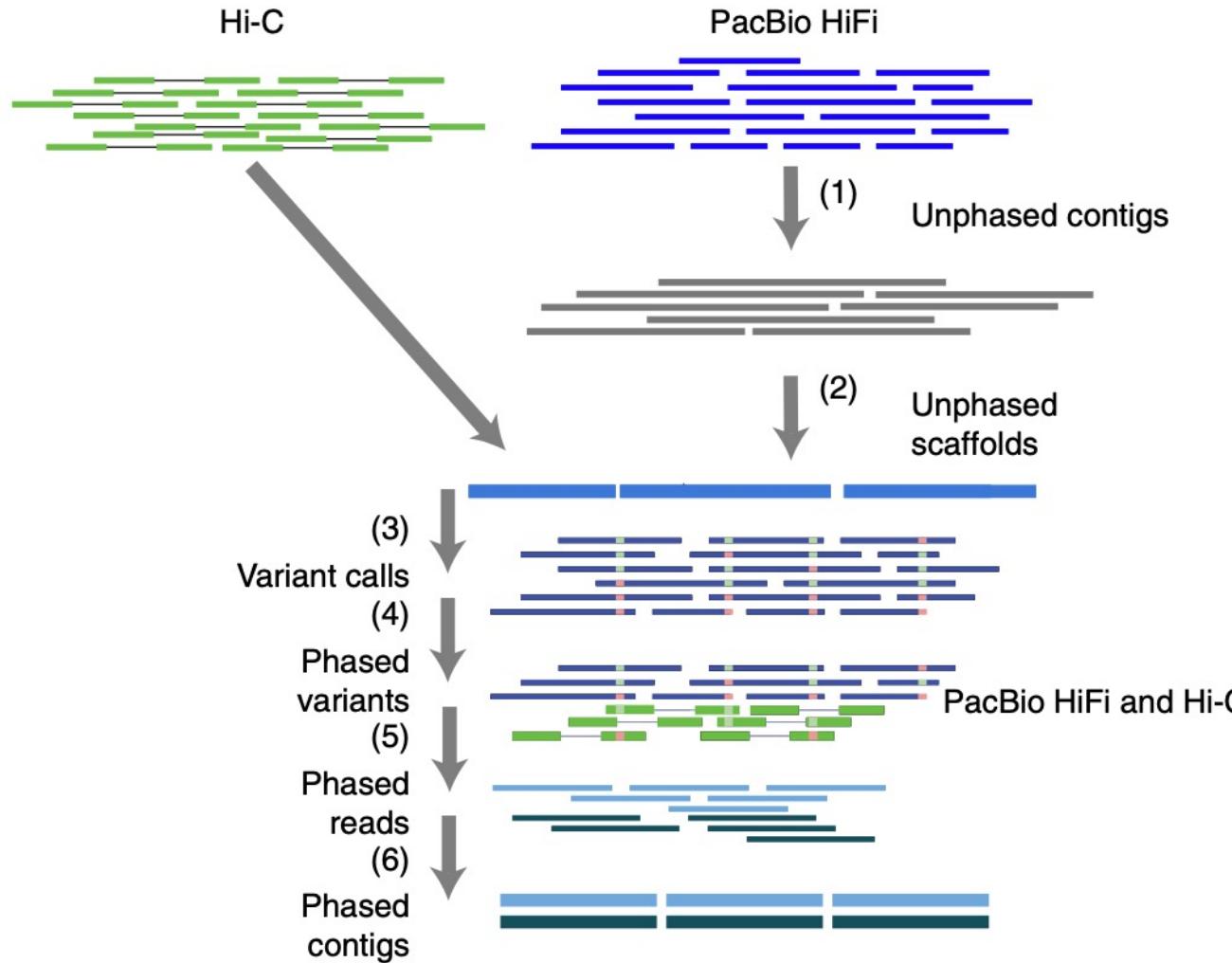
目标read纠错的规则

- 有效目标read：考虑与目标read有overlap的reads
- Informative position：两种碱基类型，且 $\geq 3$  reads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

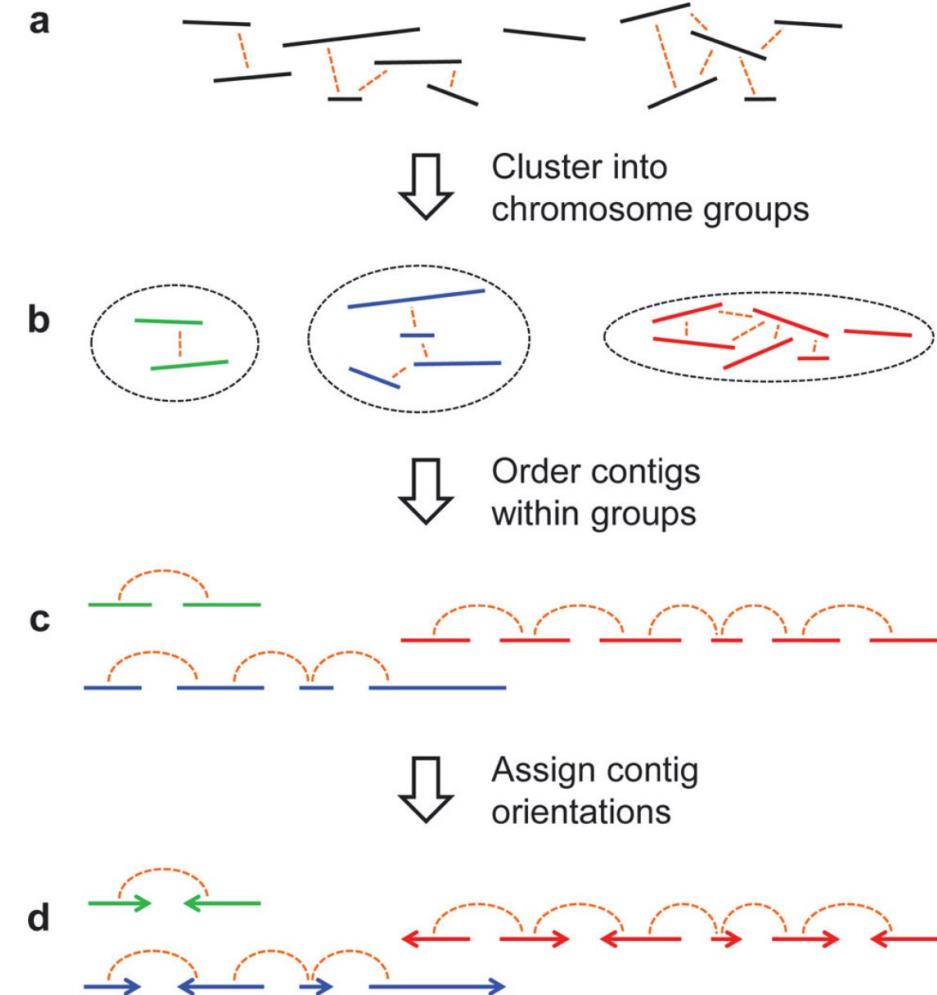
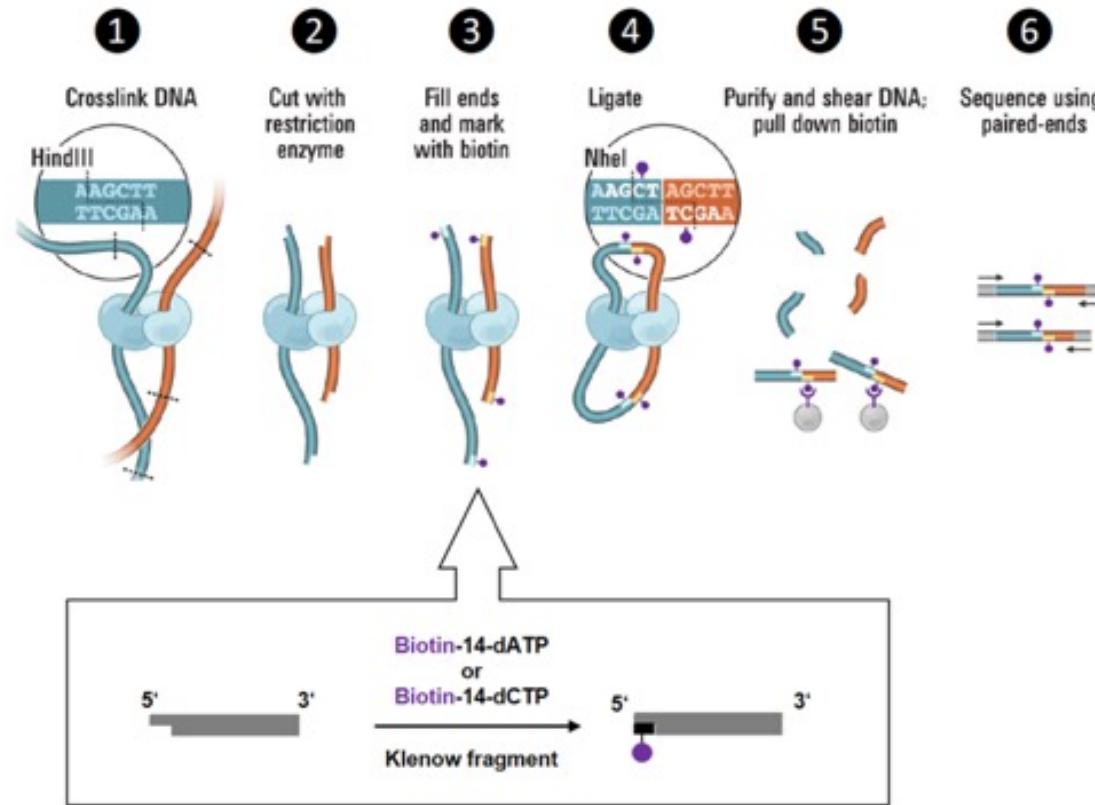
Step3：DeepVariant将HiFi reads映射到scaffolds上，获得杂合的SNPs信息

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

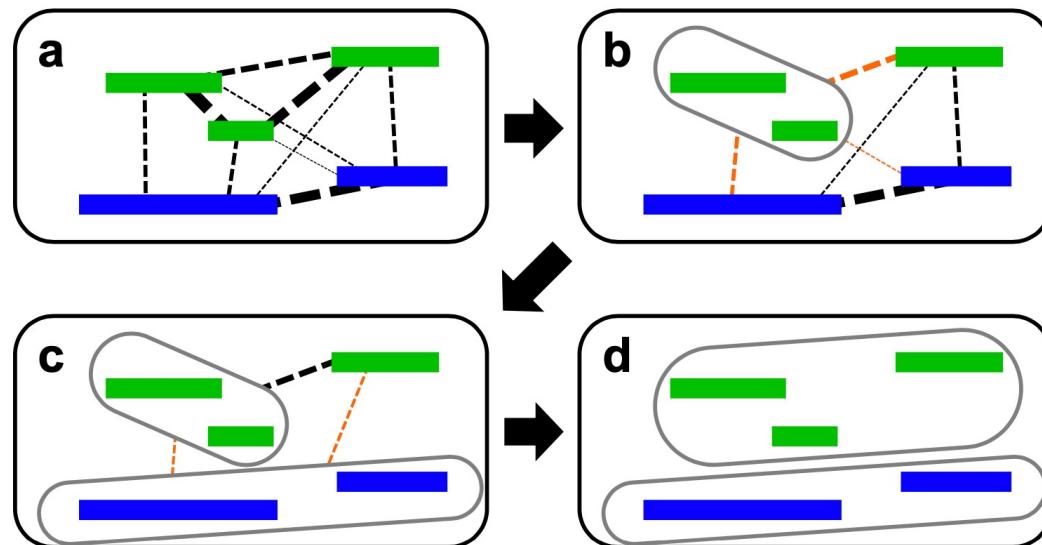
Assembly workflow

# Software5-LACHESIS (HI-C)

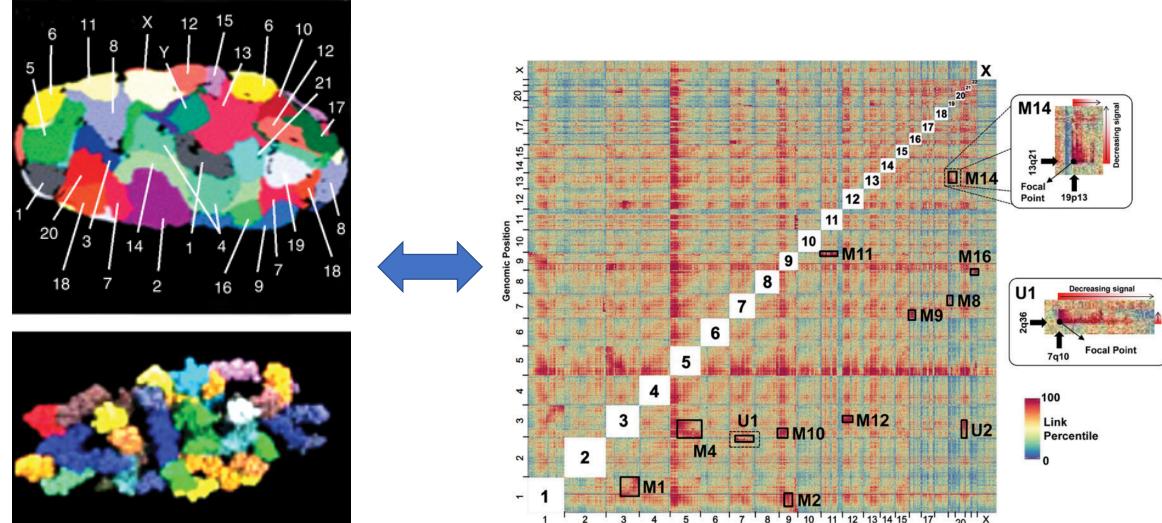
## Step1 : HI-C reads 比对

- 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

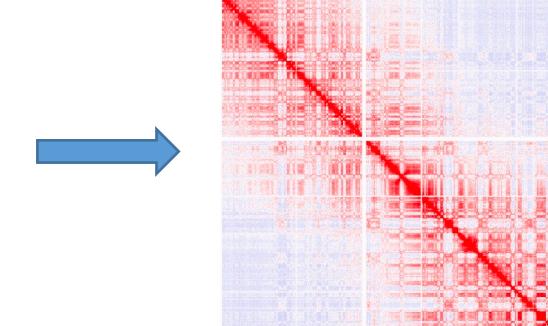


!!!染色体内的互作要远大于染色体间的互作!!!



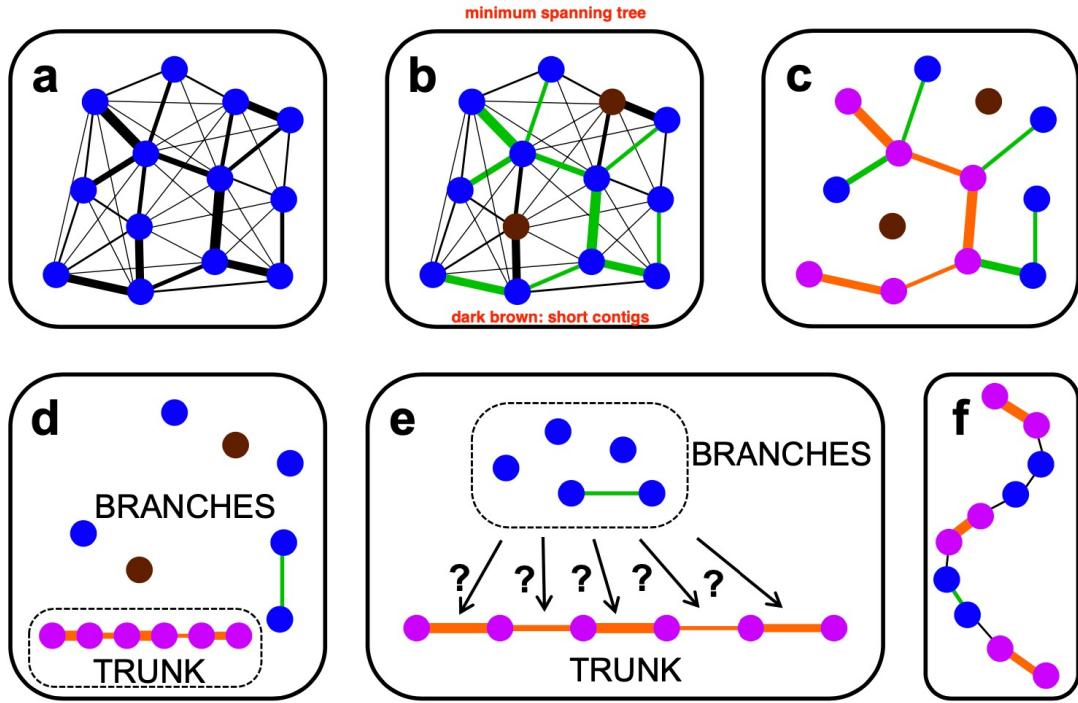
使用层次聚类的算法对互作信息 ( average-linkage metric ) 进行聚类

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	X
1	72	30	75	71	88	66	32	59	31	45	96	37	23	96	75	46	93	20	70	12	39	28	68
2	27	34	80	14	21	74	22	28	17	54	10	54	76	86	87	62	73	49	59	74	23	23	
3	95	51	35	69	37	94	33	28	93	54	13	98	27	18	82	32	20	76	75	42	82	29	66
4	13	64	63	40	86	39	65	12	18	40	99	99	67	95	29	25	69	100	33	99	55	41	41
5	96	68	29	58	40	41	71	15	80	13	88	27	35	76	81	84	20	21	83	99	62	88	50
6	84	15	38	61	46	52	29	37	54	52	53	54	52	53	54	38	32	29	24	24	24	24	18
7	10	88	72	40	47	87	52	69	46	56	89	89	25	16	75	72	31	11	29	08			
8	70	82	17	73	13	51	88	62	48	100	24	67	45	15	29	49	32	71	83	12	12	33	51
9	26	93	76	38	71	71	78	30	66	34	93	43	77	68	18	40	12	79	13	64	22	99	59
10	15	31	22	52	44	36	26	40	99	33	70	79	80	52	68	50	63	48	67	46	68	19	60
11	61	73	76	16	34	94	100	83	34	11	66	86	27	92	76	84	20	38	90	51	94	25	80
12	35	48	31	55	91	44	15	13	19	31	88	64	45	55	100	32	35	67	95	69	40	76	47
13	93	55	63	87	53	13	49	64	57	77	21	74	68	99	84	52	25	24	89	32	38	24	86
14	45	86	33	46	53	66	75	59	85	87	36	74	99	82	72	79	21	66	53	39	52	85	47
15	12	68	76	33	79	78	29	52	45	54	63	24	56	52	73	96	34	16	85	9			
16	41	38	79	57	11	36	70	56	48	29	97	100	43	70	29	56	35	80	33	80	41	28	
17	68	93	84	44	19	54	89	36	81	16	36	77	100	87	86	56	12	77	42	86	57	84	
18	97	38	52	10	43	12	53	87	33	44	35	77	62	59	73	61	50	32	22	96	89	82	63
19	25	16	87	44	24	70	36	70	56	22	45	25	11	31	93	30	23	100	88	43	19	57	42
20	37	29	81	85	50	26	91	69	67	42	36	85	57	26	49	20	33	90	31	43	88	77	87
X	15	26	65	62	93	19	91	57	37	89	41	65	71	37	26	61	71	83	35	25	77	60	



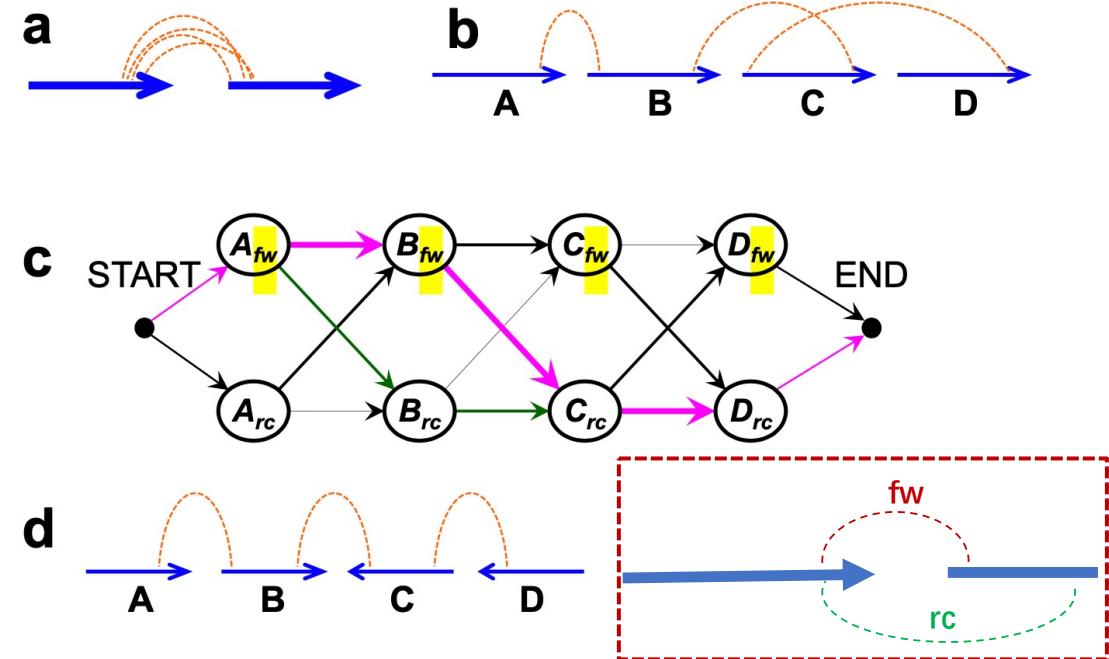
# Software5-LACHESIS (HI-C)

## 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（trunk）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



# Thank You!



官方网站



官方微信

TCGATCGA GATCGATCGATCGATCGATCG  
GATCGATCGATCGATCGATCGATCGATCG  
CGATCGATCGATCGATCGATCGATCGATCG

[www.berrygenomics.com](http://www.berrygenomics.com)