基因组与三维基因组分析培训

学习文档

目录

一、调研图

GenomeScope 是一个快速分析未组装短读序列的基因组的工具。它可以从未组装的测序数据中推断出基因组的整体特征，包括基因组大小、重复序列的丰度和杂合率。这些特征对于研究基因组演化趋势非常重要，并且可以在分析过程中作为独立的质量控制，例如评估组装的质量或者测定基因组杂合碱基数目等。

1.1 软件安装

1、Jellyfish

使用 GenomeScope 之前，你需要先计算 k-mer 频率的直方图。推荐使用 Jellyfish 工具来进行计算。

下载地址：https://github.com/gmarcais/Jellyfish/releases

从源代码获取易于编译的打包压缩文件，请执行以下步骤：

./configure --prefix=$PWD

make -j 4

make install

2、GenomeScope

要安装 GenomeScope，您需要安装这两个 R 包 argparse 和 minpack.lm 包。一种实现方法是通过 conda 安装，类似于以下操作：

conda install -c conda-forge r-base r-minpack.lm r-argparse

下载 GenomeScope 2.0 并进入其目录：

git clone https://github.com/tbenavi1/genomescope2.0.git

cd genomescope2.0/

然后我们需要通过运行以下命令来安装 GenomeScope 2.0 软件包：

Rscript install.R

安装完 R 软件包后，命令行用户可以使用 R 脚本 genomescope.R 来运行建模。

3、安装 Smudgeplot

安装 smudgeplot 最简单的方法是使用 conda，其中有可用的 smudgeplot 软件包。可以通过以下命令获取 smudgeplot：

conda install -c bioconda smudgeplot

4、添加环境变量

```
export PATH=/mnt/RAID-5/MD0/bioinformatics/soft/genome/mambaforge/bin/:$PATH
```

1.2 计算 kmer

我们推荐使用 KMC。这个版本的 KMC 有一个额外的 smudge_pairs 程序，它比 hetkmers 的 Python 版本更快速地找到异质 kmer 对，并且使用的内存更少。输入是基因组测序 reads，覆盖度越高越好。该方法设计用于处理大型数据集，可以将所有的单端/双端库合并在一起。对于大多数基因组，我们建议使用 kmer 长度为 21，这个长度足够长以避免大部分 kmer 的重复性，并且对测序误差更加鲁棒。对于非常大的（单倍体大小>>10GB）和/或者重复非常高的基因组，使用较大的 kmer 长度可以增加独特 kmer 的数目。准确的推断需要至少 25 倍的覆盖度，否则模型拟合效果会较差或者无法收

敛。此外，GenomeScope 也需要相对低的测序错误率，例如 Illumina 测序，以确保大部分 kmer 没有错误。例如，2%的错误率对应着平均每 50bp 出现一个错误，这个错误率大于典型的 kmer 大小(k=21)。需要注意的是，目前 Oxford Nanopore 或 Pacific Biosciences 等单分子测序技术的原始测序数据平均错误率为 5-15%，不支持该类型数据，因为平均每 6 到 20 个碱基就会出现一个错误(但 Pacbio HiFi CCS 数据可以用)。

1、KMC 计算 kmer

使用 clean 的 reads 文件给 KMC 计算 kmer 频率，然后生成 kmer 的直方图：

```
mkdir tmp
ls *.fastq.gz > FILES
```

kmer 长度为 21，使用 15 个线程，60G 内存，在 1 到 10000x 的 kmer 覆盖度之间计算 kmer 覆盖度

```
kmc -k21 -t15 -m60 -ci1 -cs10000 @FILES kmcdb tmp
kmc_tools transform kmcdb histogram kmcdb_k21.hist -cx10000
```

其中-k 是 kmer 长度，-m 是要使用的大致内存量（以 GB 为单位，1 至 1024），-ci<value>排除出现次数小于<value>次的 kmer，-cs 是最大值，FILES 是一个包含输入文件列表的文件名，kmcdb 是 KMC 数据库的输出文件名前缀，tmp 是一个临时目录，-cx<value>是要存储在直方图文件中的计数器的最大值。

2、jellyfish 计算

```
jellyfish count -C -m 21 -s 1000000000 -t 1  *.fastq -o reads.jf
```

注意根据你的服务器设置适当的内存(-s)和线程(-t)参数。上面的例子将使用 1 个线程和 1GB 的内存。如果你的测序覆盖度低或者错误率高，可能需要调整 kmer 长度(-m)。在计算 kmer 频率直方图时，应始终使用“canonical kmers”（-C），因为测序 reads 会来自 DNA 的正向链和反向链。

导出 kmer 频率直方图：

```
jellyfish histo -t 10 reads.jf > reads.histo
```

同样，线程数(-t)应根据你的服务器进行调整。当有了 Jellyfish 直方图文件后，你可以在在线的 Web 工具上运行 GenomeScope，或者在命令行中运行。

1.3 运行 GenomeScope

用户可以使用在线版本，它提供了易于使用的 Web 界面，具有与命令行版本相同的功能：http://genomescope.org/，也可以命令行运行 GenomeScope，主要参数如下：

-i INPUT, --input INPUT　输入直方图文件

-o OUTPUT, --output OUTPUT　输出目录名

-p PLOIDY, --ploidy PLOIDY　使用的多倍体数（1、2、3、4、5 或 6）[默认为 2]

-k KMER_LENGTH, --kmer_length KMER_LENGTH　用于计算 kmer 谱的 kmer 长度 [默认为 21]

-n NAME_PREFIX, --name_prefix NAME_PREFIX　输出文件的可选名称前缀

-m MAX_KMERCOV, --max_kmercov MAX_KMERCOV　可选的最大 kmer 覆盖阈值

（覆盖率大于 max_kmercov 的 kmers 将被模型忽略）

--no_unique_sequence 可选标志，关闭绘图中的黄色唯一序列线

--initial_repetitiveness INITIAL_REPETITIVENESS 高级选项：设置重复性的初始值

--initial_heterozygosities INITIAL_HETEROZYGOSITIES 高级选项：设置核苷酸杂合率的初始值

--num_rounds NUM_ROUNDS 高级选项：优化轮数的参数（拟合线不好时候提高，可以设置成 4）

--fitted_hist 高级选项：生成拟合后的 kmer 多重性 0-4 的直方图和概率查找表

--start_shift START_SHIFT 高级选项：在拟合轮次之间排除的覆盖变化

--typical_error TYPICAL_ERROR 高级选项：测序错误的典型水平

命令行用户可以使用 R 脚本 genomescope.R 来运行建模。该脚本估计基因组的大小、杂合性和重复比例。

```
genomescope2 -i kmcdb_k21.hist -k 21  -o results -m 1000 -p 2
```

histogram_file、k-mer_length 和 output_dir 是必需的参数。可选参数 kmer_max 指定从分析中排除高频 kmer 的阈值。建议将 kmer_max 设置为 1000，但具体取决于你的数据特征。输出的图表和推断得到的基因组特征的文本文件将输出到指定的 results 目录中。并输出：

Model converged het:0.00131 kcov:14.4 err:0.00497 model fit:0.748 len:119501447

图表和完整的结果将在 output 目录中显示，显示估计的基因组大小为 119.5Mbp，杂合率为 0.131%（由于建模过程中的随机性，具体数值可能稍有不同）。

1.4 GenomeScope 常见问题

>问：为什么模型没有收敛，或者为什么结果与预期有很大的差异?

答：最常见的问题是**覆盖度太低**，模型无法自信地识别与纯合 kmer 相对应的峰值。这可以通过 kmer 图中**缺乏任何峰值**或者模型拟合与观察到的 kmer 分布不符来识别。要解决此问题，请首先确保已使用规范的 kmer 计数模式（如果使用 jellyfish，则为-C 选项）。如果仍然失败，可以尝试略微**减小 kmer 的大小**，使用 17 或者 19。如果所有尝试都失败，那么很遗憾，您将需要生成额外的测序数据。

> 问：如《补充说明和图 1.3.2 基因组大小估计》所述，单倍体基因组大小的估计方式是：“通过将除了在第 1.3.1 节中鉴定出来的推测测序错误以外的所有 kmer 总数相加，然后除以 2*λ，λ是纯合 kmer 的平均覆盖度”。我理解得是否正确，λ是一个分布的平均值，表示纯合 kmer 的平均覆盖度；而在 GenomeScope 概要文件中，kcov 是杂合 kmer 的估计覆盖度。您能否解释一下 GenomeScope 如何估计单倍体基因组大小，特别是为什么要除以 2 倍的纯合 kmer 的估计覆盖度?

答：首先要注意的是λ和 kcov 指的是同一个值，只是在书面文档中我们使用λ，在代码中使用 kcov。该模型试图识别以λ、2λ、3λ和 4λ为中心的 4 个峰值。这 4 个峰值分

别对应于独特的杂合、纯合、重复的杂合和重复的纯合序列的平均覆盖水平。因此，当估计单倍体基因组大小时，它会除以2λ，即平均纯合覆盖度，而不是您所写的“纯合kmer的估计覆盖度的2倍”。

另一个可能令人困惑的方面是单倍体基因组大小和二倍体基因组大小的含义。我们认为单倍体基因组大小是指一个完整单倍体染色体组的长度范围，而二倍体基因组大小是指两个单倍体拷贝长度（一个二倍体细胞中的总DNA含量）。特别是在人类细胞中，单倍体基因组大小约为3Gbp，而二倍体基因组大小约为6Gbp。如果您对一个人类基因组进行了总共300Gbp的测序，那么母系单倍型的测序量大约为150Gbp（50倍覆盖度），父系单倍型的数据量也是如此。但由于人类的杂合率非常低，分布的主峰将以100倍左右为中心。然而，GenomeScope仍会尝试拟合这4个峰值，因此应该将杂合kmer覆盖度λ设置为50倍，从而将纯合覆盖度设置为2*λ = 100倍。基于此，GenomeScope将计算单倍体基因组大小，即将总测序数据量（300GB）除以纯合覆盖度（100倍），得到预期的3Gbp。具有较高覆盖度的kmer也会自然地被缩放：在kmer分布中出现200或者300次（因此在二倍体基因组中出现2个或者3个拷贝）的kmer仍会以100倍缩放，为估计贡献2个或者3个拷贝的。最后，请注意，如果两个单倍型的长度明显不同，那么报告的单倍体基因组大小将是两者的平均值。

> 问：GenomeScope能够用于估计多倍体性或者具有更高多倍体性的基因组吗?

答：A: 是的，GenomeScope 2.0适用于多倍体水平最高为六倍体（6）。要估计多倍性，您可以使用我们开发的另一种工具Smudgeplot。目前，GenomeScope不支持染色体拷贝数量不均匀的基因组，例如非整倍体的癌症基因组或不等数量的性染色体。在这些情况下，报告的异质性率将代表具有单倍体（拷贝数1）和二倍体（拷贝数2）的碱基比例，以及其他染色体中的任何杂合位点。

> 问：我尝试使用GenomeScope对我正在研究的生物体进行基因组大小估计。使用XXX论文中广泛使用和引用的k-mer方法和计算，得到了869 - 919 Mbp的估计结果，这与其他一项已发表研究中的0.83pg的cDNA值大致一致。然而，使用GenomeScope，我得到了650 Mbp的估计结果。您或您的团队对我为什么会观察到估计结果相差超过200Mbp的偏差有什么见解吗?

答：尽管GenomeScope可以自动化大部分分析，但它有一个**需要过滤高频kmer的关**键参数。由于我们经常在真实样本中看到出现1万次或10万次甚至更多次数的kmer，这些通常是测序的人为引入的测序错误，**或细胞器测序（或其他污染）**。为了处理这些情况，GenomeScope默认情况下会排除出现超过1000次的kmer进行分析，这会导致基因组大小估计为649Mbp。如果将此限制提高到10000，那么新的大小估计结果将为697Mbp。您可能可以将此限制提高得更高，但是看起来您的直方图被截断在10000次（这是jellyfish的默认值）。如果您想包括这些超高频率的kmer，则需要重新生成从jellyfish获得的直方图，并将最大覆盖度阈值设置为100000，甚至可能是1000000。这可能会增加估计的基因组大小，但同时也会使分析对数据中的任何伪像更敏感。不幸的是，每个项目在去除这些伪像时的最佳策略是根据实际样本和研究目的进行调整。您可能需要进行一些试验来找到合适的参数设置，以获得最准确的估计结

果。

此外，不同方法和工具可能会产生略有不同的估计结果，因为它们可能基于不同的模型和算法。因此，在比较不同研究中的基因组大小估计时，需要谨慎考虑方法的选择和参数设置。

最后，基因组大小估计只是一个近似值，并且可能受到多种因素的影响，例如测序深度、杂合性、重复序列等。因此，对于确切的基因组大小，通常需要进行更详细的分析和验证。

二、基因组组装

Hifiasm 是一种快速的单倍型重构 de novo 组装软件，最初设计用于 PacBio HiFi read。其最新版本可以利用超长的 Oxford Nanopore 数据来支持端到端（T2T）组装。Hifiasm 结合了 HiFi、超长和 Hi-C 产生了可能是最好的单个样本 T2T 组装方式，并且在给定父母本二代测序数据情况下是最好的单倍型组装软件之一。对于人类基因组，hifiasm 可以在一天内产生端到端组装。

Hifiasm 能够生成高质量的端到端组装，它倾向于产生更长的 contigs，并能解决其他组装软件产生更多的分段重复的问题。在给定 Hi-C 或父母本二代测序数据的情况下，hifiasm 可以总体上产生迄今为止最好的单倍型基因组组装结果。Human Pangenome Project 对第一批样本的组装选用此软件。Hifiasm 可以默认清除 haplotig 重复，而不依赖于像 purge_dups 这样的第三方工具。Hifiasm 也不需要像 pilon 或 racon 这样的 polishing 工具。这简化了组装流程并节省了运行时间。

Hifiasm 速度快。它可以在半天内组装完人类基因组，并在三天内组装约 30Gb 的红杉基因组。对于 Hifiasm 来说没有太大的基因组是不可以组装的。

2.1 三代 HiFi 组装

Hifiasm 安装简单易用。它不需要 Python、R 或 C++11 编译器，并且可以编译成单个可执行文件。默认设置适用于各种基因组。

1、Hifiasm 安装

```
# Install hifiasm (requiring g++ and zlib)
git clone https://github.com/chhylp123/hifiasm
cd hifiasm && make
```

2、Hifiasm 使用

```
hifiasm -o test -t4 -f0 CRR302668.part_001.fastq.gz
awk '/^S/{print ">"$2;print $3}' test.bp.p_ctg.gfa|seqkit  seq  -w 100 >final.p_ctg.fa  # get primary contigs in FASTA
gaep stat test.p_ctg.fa
```

```
# Assemble inbred/homozygous genomes (-l0 disables duplication purging)
hifiasm -o CHM13.asm -t32 -l0 CHM13-HiFi.fa.gz 2> CHM13.asm.log
# Assemble heterozygous genomes with built-in duplication purging
hifiasm -o HG002.asm -t32 HG002-file1.fq.gz HG002-file2.fq.gz
# Hi-C phasing with paired-end short reads in two FASTQ files
hifiasm -o HG002.asm --h1 read1.fq.gz --h2 read2.fq.gz HG002-HiFi.fq.gz
# Single-sample telomere-to-telomere assembly with HiFi, ultralong and Hi-C reads
hifiasm -o HG002.asm --h1 read1.fq.gz --h2 read2.fq.gz --ul ul.fq.gz HG002-HiFi.fq.gz
```

3、输出文件

通常情况下，hifiasm 生成以下使用 GFA 格式的组装图：

prefix.p_ctg.gfa：primary contigs 的组装图。此图包含一个完整的组装，其中包括长

的相位块。

prefix.a_ctg.gfa：alternate contigs 的组装图。此图包括所有在 primary contigs 图中被丢弃的 contigs。

prefix.hap.p_ctg.gfa：分型的 contigs 组装图。此图保存有相位化 contigs。

Hifiasm 在任何情况下都会输出*.r_utg.gfa 和*.p_utg.gfa。

通过 Hi-C 分区选项，hifiasm 输出：

prefix.hic.p_ctg.gfa：主要 contigs 的组装图。

prefix.hic.hap1.p_ctg.gfa：haplotype1 的完全相位化 contigs 图，其中每个 contigs 都被完全相位化。

prefix.hic.hap2.p_ctg.gfa：haplotype2 的完全相位化 contigs 图，其中每个 contigs 都被完全相位化。

prefix.hic.a_ctg.gfa（使用--primary 可选）：备用 contigs 的组装图。

hifiasm 仅在默认情况下使用 HiFi 读取生成以下组装图：

prefix.bp.p_ctg.gfa：主要 contigs 的组装图。

prefix.bp.hap1.p_ctg.gfa：haplotype1 的部分相位化 contigs 图。

prefix.bp.hap2.p_ctg.gfa：haplotype2 的部分相位化 contigs 图。

如果指定了选项--primary 或-l0，则 hifiasm 会输出：

prefix.p_ctg.gfa：主要 contigs 的组装图。

prefix.a_ctg.gfa：备用 contigs 的组装图。

4、常见问题

1）如何获得 FASTA 格式的 contigs?

可以使用以下命令从 GFA 文件中生成 FASTA 文件：

```
awk '/^S/{print ">"$2;print $3}' test.p_ctg.gfa > test.p_ctg.fa
```

2）应该使用哪种组装类型?

如果有父母本数据，则应始终优先选择在“trio-binning 模式”下生成的 dip.hap.p_ctg.gfa。否则，如果有 Hi-C 数据，则以在 Hi-C 模式下生成的 hic.hap.p_ctg.gfa 为最佳选择。trio-binning 模式和 Hi-C 模式都会生成完全分型的组装结果。

如果您只有 HiFi read，则 hifiasm 默认输出 bp.hap.p_ctg.gfa。通过使用--primary 可以还可以产生主要/备用组装结果。所有这些仅基于 HiFi 的组装结果都不是完全分型的。

3）是否支持自交/同源基因组?

是的，请使用-l 0 选项禁用去除重复步骤。

4）是否支持二倍体基因组?

是的，hifiasm 的大多数模块都设计用于二倍体样本，包括去除重复步骤、部分分型组装和使用 trio-binning 或 Hi-C 进行完全分型组装。

5）是否支持多倍体基因组?

r_utg.gfa 和 p_utg.gfa 文件没有损失，因此它们也适用于多倍体基因组。但是，目前 hifiasm 的 contig 生成模块仅设计用于二倍体样本，这意味着部分分型组装和完全分型组装都不直接支持多倍体基因组。如果将--n-hap 设置为>2，则可以提高多倍体基因组

的主要组装质量。请使用主要组装用于多倍体样本并使用第三方工具（例如purge_dups）运行多轮去重步骤。

6）为什么一个集成了Hi-C数据的组装结果比另一个大?

对于像人类男性这样的一些样本，父本单倍型应该比母本单倍型大。但是，如果一个组装结果比另一个大得多，则这可能是由于hifiasm存在问题。要解决此问题，请为-s（默认值：0.55）设置较小的值。

另一个可能性是hifiasm错误地识别了同源read的覆盖阈值。例如，hifiasm在组装期间输出以下信息：

[M::purge_dups] homozygous read coverage threshold: 36

在此示例中，hifiasm将同源read的覆盖阈值视为36。如果它明显小于同源覆盖峰值，则hifiasm将生成两个不平衡的组装结果。在这种情况下，请将--hom-cov设置为同源覆盖峰值。

7）对于集成了Hi-C数据的组装结果，为什么两个单倍体组装的大小都比估计的基因组大小要大得多?

很可能是因为hifiasm错误地识别了同源read的覆盖阈值。Hifiasm用于调试的输出信息如下：

[M::stat] # heterozygous bases: 645155110; # homozygous bases: 1495396634

如果大部分二倍体样本的碱基都是同种型，则hifiasm错误地确定了覆盖阈值。例如，hifiasm在组装期间输出以下信息：

[M::purge_dups] homozygous read coverage threshold: 36

在此示例中，hifiasm将同源read的覆盖阈值视为36。如果它明显小于同源覆盖峰值，则hifiasm认为大多数read都是同源的，并将它们分配给两个组装结果，从而使两个组装结果比预计的单倍体基因组大得多。在这种情况下，请将--hom-cov设置为同源覆盖峰值。

8）如何调整参数以改善集成了Hi-C数据的组装结果?

与仅基于HiFi的组装结果或trio binning分组组装结果相比，集成了Hi-C数据的组装结果略微复杂，因此需要注意结果。有关如何解决潜在问题的详细信息，请参见“为什么一个集成了Hi-C数据的组装结果比另一个大？”和“对于集成了Hi-C数据的组装结果，为什么两个单倍体组装的大小都比估计的基因组大小要大得多？”。

还有其他几个选项可能会影响集成了Hi-C数据的组装结果。增加--n-weight、--n-perturb和--f-perturb的值可能会改善相位结果，但需要更长时间。然而，调整--l-msjoin就有点棘手了。所有这些选项都不会影响p_utggfa文件，因此可以重复使用hic.bin文件。

9）为什么主要组装版本比完全分型组装和部分分型组装（即*.hap*.p_ctg.gfa）更连续?

对于二倍体样本，主要组装版本通常具有更高的N50值，但代价是备用组装结果高度分段。从方法上看，主要组装具有额外的连接步骤，该步骤将两个单倍型连接起来，从而使主要组装更连续。

在生成完全分型组装和部分分型组装时，hifiasm 旨在保持两个单倍型连续。对于许多下游应用程序（如结构变异调用）而言，这非常重要。

10）我的组装结果过于分段或不够连续，该怎么办?

提高-D 或-N 可能会提高重复区域的分辨率，但需要更长时间。这两个选项影响所有类型的组装结果，并且通常不会对组装质量产生负面影响。相反，--purge-max 仅影响主要组装。为--purge-max 设置较大的值可以使主要组装更连续，但可能会折叠重复区域或片段重复。

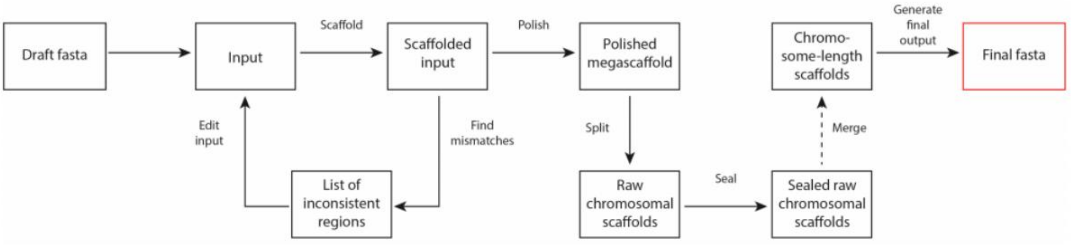
如果组装结果过于分段，则用户应检查 HiFi 数据是否足够好。有关详细信息，请参见“为什么 hifiasm 被卡住或崩溃？”。

11）如何避免组装错误?

将-s、-O 和-s 的值设小，或使用-u 选项可以减少组装错误。

2.2 3D-DNA 组装

3D-DNA 是一种基于 Hi-C 数据的自定义计算流程，用于根据互作数据来修正 DNA 的错误组装、锚定、排序和定向。其工作流程概述如图所示。



首先，我们进行一系列迭代步骤，旨在消除输入片段中的拼接错误。每个步骤都以一个脚手架池开始（最初，该池是输入片段的集合）；使用脚手架算法对这些脚手架进行排序和定向；然后应用拼接错误修正算法来检测脚手架池中的错误。最后，编辑后的脚手架池被用作下一次拼接错误修正算法的输入。这些迭代的最终效果是可靠地检测输入脚手架中的拼接错误，而不会移除正确组装的序列。

完成迭代后，将脚手架算法应用于修订后的输入脚手架，得到一个输出结果，即将所有染色体连接起来的单个“超级脚手架”。然后进行后处理，包括以下四个步骤：(i) 抛光算法；(ii) 染色体分割算法，用于从超级脚手架中提取染色体长度的脚手架；(iii) 密封算法，用于检测拼接错误修正过程中的误报，并从原始脚手架中恢复错误移除的序列；(iv) 合并算法，用于修正输入脚手架中由于未主导杂合性而引起的组装错误。第四步默认情况下不运行，只有在已知草稿中存在大量未主导杂合性的情况下才需要运行。

以上就是 3D-DNA 流程的概述，它可以帮助纠正 Hi-C 数据中的组装错误，并生成经过校正的染色体集合。这些校正后的染色体集合可用于进一步研究和分析基因组的空间结构和功能。

1、软件安装

1)Juice

git clone https://github.com/theaidenlab/juicer.git

cd juicer

ln -s CPU scripts

cd scripts/common

wget https://hicfiles.tc4ga.com/public/juicer/juicer_tools.1.9.9_jcuda.0.8.jar

ln -s juicer_tools.1.9.9_jcuda.0.8.jar juicer_tools.jar

2）3D-DNA

直接下载：https://github.com/aidenlab/3d-dna/releases/tag/201008

2、merged_nodups.txt 制备

```
bwa index AT.fa
python /mnt/RAID-5/MD0/bioinformatics/soft/juicer/misc/generate_site_positions.py MboI AT AT.fa
perl get_Seq_length.pl AT.fa AT.len
mkdir fastq##链接 fastq，注意格式
mkdir splits##可以是拆分的 fastq
sh /mnt/RAID-5/MD0/bioinformatics/soft/juicer/scripts/juicer.sh -g AT -s MboI -z AT.fa -y AT_MboI.txt -p AT.len --assembly -t 30
```

2、3D-DNA 用法

创建一个 fastq 文件夹，并使 Hi-C 数据文件命名为 _*R[1,2]*.fastq and _*R[1,2]*.fastq.gz)，然后执行下述命令：

```
/usr/bin/bash  /mnt/RAID-5/MD0/bioinformatics/soft/3d-dna-201008/run-asm-pipeline.sh -r 1 AT.fa aligned/merged_nodups.txt
```

其中最重要的参数包括：以下是一些调整 3D-DNA 流程性能的参数选项的简要列表：

-i 或 --input：用于定义 scaffold（脚手架）尺寸阈值。小于该阈值的 contigs/scaffold 将被算法忽略，并在输出时不进行任何变化。默认值为 15000。

-r 或 --rounds：迭代纠错的轮数。默认为 2 轮。

-m 或 --merge：指定是否运行合并模块的参数。默认情况下不运行合并模块，只有在预期草稿组装中存在大量未主导杂合性序列时才应使用。

以上是一些可用于调整 3D-DNA 流程性能的参数选项。您可以根据具体需求进行调整。默认情况下，3D-DNA 会将小于 -i 或 --input 参数指定的值的输入片段视为“Tiny”（细小）脚手架，并在分析过程中将它们排除在外。

通过将这些小脚手架排除在分析之外，可以减少噪声和错误，提高结果的可靠性。如果您认为这些小脚手架对您的分析很重要，可以调整 -i 或 --input 参数的值，使其大于默认的 15000，以便包含更小的脚手架。

3、输出。

.fasta 文件：.fasta 是一种用于表示核序列的标准文本格式。输出文件以.fasta 格式呈现，其中包括标记为"FINAL"的染色体长度序列。流水线生成的.fasta 文件还包括作为错误连接检测和合并的一部分生成的初步序列的所有片段（如果处于二倍体模式）。

.hic 文件：.hic 文件是高度压缩的二进制文件，以巧妙的方式存储多个分辨率的接触矩阵，允许随机访问。在这里了解更多关于.hic 格式的信息：https://github.com/theaidenlab/juicer/wiki/Data#hic-files。以.hic 格式输出的文件包括"final" - 接触图，对应于最终的染色体长度输出（在单倍体模式下运行时）。这是在 Juicebox Assembly Tools 中查看的最自然的候选文件（与相应的最终.assembly 文件一起使用）。

.assembly 文件：.assembly 是一种空格分隔的文本格式，以简洁的方式编码对初步序列执行的一系列指令，包括分割、改变顺序、方向和锚定到染色体。对于流水线的所有阶段都生成了这些文件。

.scaffold_track.txt 和.superscaf_track.txt 文件：这些是 Juicebox 2D 注释格式的独立脚手架和超级脚手架（染色体）边界文件。流水线的所有阶段都会生成这些文件。在使用基本的 Juicebox 功能时，可以使用这些文件（基于.assembly 文件自动生成边界注释），但在使用 Juicebox Assembly Tools 时并不是必需的。

其他文件：

edits.for.step..txt、mismatches.at.step..txt、suspect_2D.at.step.*.txt：问题区域的列表（Juicebox 2D 注释格式）。

alignments.txt（仅适用于二倍体模式）：备选单倍型候选的成对比对数据，由 LASTZ 生成。

2.3 JBAT 调图

通常，基因组组装中的错误在 Hi-C 图中是明显可见的。当参考序列正确组装时，线性基因组组装序列相邻的位点在 Hi-C 实验中也会处于紧密的物理邻近，导致 Hi-C 热图中沿着对角线出现亮度较高的接触频率带。相反，当参考序列组装存在错误时，热图中将出现异常的模式。3D-DNA 在其工作流程中生成组装热图。热图通过将基因组组装划分为固定大小（bin)的位点，并且每个热图条目指示两个位点之间的接触频率。热图可以用于检查默认值组装结果的质量，为用户指导如何调整组装参数。实际上，对于大多数问题，默认参数通常能够很好地工作，生成具有极少错误的染色体长度组装，无需任何调整。因此，与其在各种参数范围内进行扫描以确定特定基因组的可靠参数设置，手动识别和修正少数剩余的组装错误更加高效。Assembly Tools 是 Juicebox 桌面应用程序的一个新模块，它扩展了 Juicebox 界面，用于 Hi-C 数据可视化，以便进行交互式染色体组装优化。当发现染色体组装错误时，用户可以通过简单的点选在几秒钟内完成修正。热图和参考基因组会实时更新以反映这些变化。使用 Juicebox Assembly Tools（JBAT），用户可以改进基因组并降低基因组组装的成本。JBAT 还可用于手动组装基因组。详细请查看此教程视频：

https://www.youtube.com/watch?v=Nj7RhQZHM18。

1、环境

Juicebox Assembly Tools 的布局如图 6 所示。用户界面包括：交互热图窗口、菜单栏、视图控制、右侧和注释面板。交互热图窗口包括热图的名称和 Juicebox 发布版信息。

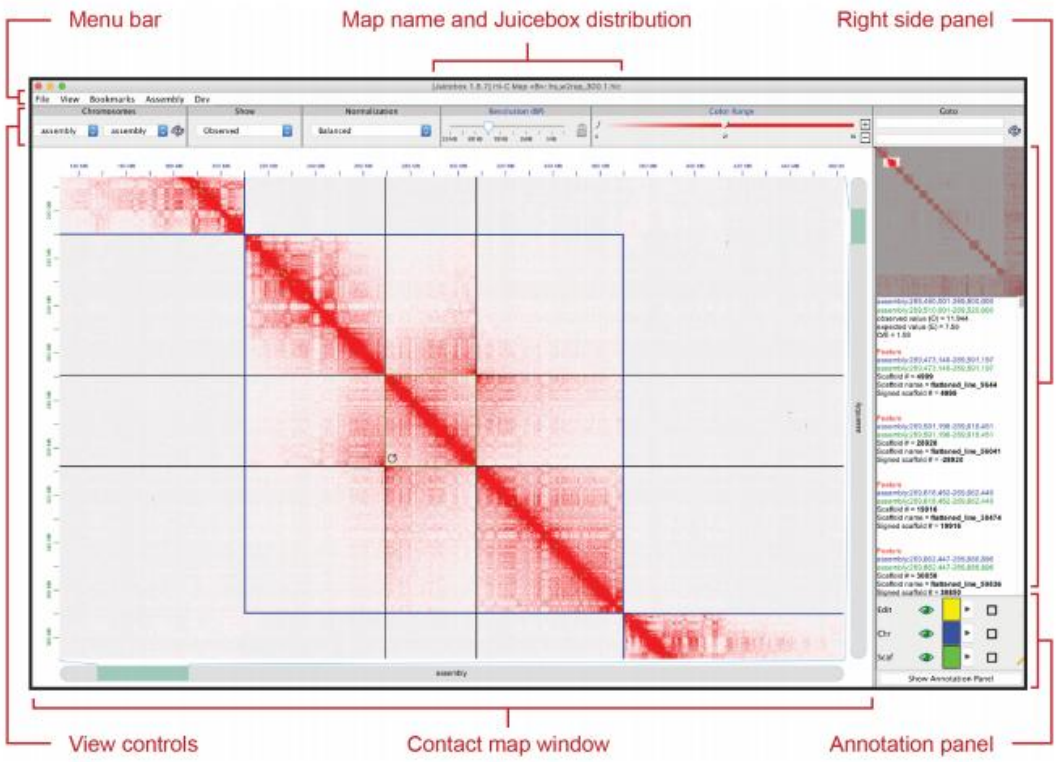


FIGURE 6. Juicebox Assembly Tools environment illustrated with hs-1k genome assembly data, see (Dudchenko et al. 2018). KR-normalized data (Normalization: Balanced) binned at 250kb resolution (Resolution slider: 250kb) is being analyzed. Only a fragment of the map is visible in the contact map window: the position of the fragment with respect to the rest of the map is highlighted on the mini map in the right side panel. Three layers of annotations associated with the assembly are visible in the Annotation panel: green (draft scaffolds), blue (output superscaffolds/chromosomes), and yellow (for holding temporary annotations). The annotations appear as squares superimposed on the contact map along the diagonal. Annotation layers are automatically populated upon loading the 'assembly' *.hic* file (Chromosome: assembly) via the *File* menu and importing the associated *.assembly* file via *Assembly* menu. In this case, the *.hic* and *.assembly* files were generated by 3D-DNA. Only a few green (scaffold) annotations are visible along the diagonal of the map: most of the individual scaffolds in hs-1k draft are too small to be visible at this zoom level. The center of the map is dominated by one blue (chromosome) annotation representing a single chromosome-scale superscaffold. A set of consecutive scaffolds is selected (encompassed by a yellow temporary annotation square with black highlight), and the detailed information associated with the selected scaffolds is displayed in the information panel under the mini map. A mouse cursor prompt for inverting the selected fragment of the assembly is shown in the lower left corner of the selection annotation square. If clicked, the action will result in changes to the contact map as well as to the underlying assembly. Changes to the assembly can be exported as a modified *.assembly* file via *Assembly* menu. The modified *.assembly* can be converted into corresponding changes in the reference *.fasta* sequence (in this case substitution of the selected sequence in the assembly for its reverse complement) using a simple command-line script (distributed as part of the 3D-DNA package). From (Dudchenko, [illegible] with modifications.

主要的 JBAT 组件如下所述。请注意，描述将侧重于与基因组组装相关的新 UI 项。

2、主菜单

在使用 Juicebox Assembly Tools 时，文件和组装菜单是最重要的。文件菜单是主要的入口点，用于将.hic 文件加载到交互图窗口中。请注意，虽然我们正在努力使 JBAT 与所有.hic 文件兼容，但目前只能使用“assembly” .hic 文件进行交互式组装。

组装菜单用于导入.assembly 文件（菜单项“导入组装图谱”）。这些文件是对参

考.fasta 文件的简明表示，并且用于告知 JBAT 底层参考序列中的个别序列如何与交互热图中的位点相关联。因此，非常重要的是使用相同的参考序列生成.assembly 和.hic 文件。

组装菜单还用于保存组装的更改（以修改后的.assembly 文件的形式；菜单项“导出组装”）。默认情况下，导出的文件名将在原始.assembly 文件名后添加'.review'后缀。请注意，.hic 文件在整个审查过程中保持不变。组装菜单还可用于加载保存的更改（菜单项“Import Modified Assembly”）。请注意，在退出应用程序后加载更改时，首先加载原始（未修改的）.assembly 文件通过“Import Map Assembly”菜单项至关重要，然后再加载修改后的.assembly 文件。

视图菜单包含用于自定义交互热图窗口外观的项目。它还用于加载 1D 和 2D 特征文件（包括. bed 和. wig，. bedpe 和. txt 注释文件，以及 3D-DNA 生成的覆盖轨迹等等）以与 Hi-C 图一起查看。

视图控件

视图控件允许选择染色体（在 JBAT 上下文中只有一个“assembly”伪染色体），在观察到的、观察到的/期望的、控制（目前在 JBAT 中不可用）和其他视图之间切换，调整归一化，更改分辨率，微调颜色范围或转到交互热图中的特定位置。

在 Juicebox Assembly Tools 中工作时，“Goto”控件也可用于通过序列名字搜索对应于特定参考序列的交互区域。

交互热图窗口

这是热图加载的地方。可以通过用鼠标拖动和移动或滚动来平移热图：在交互热图顶部滚动将沿对角线移动显示区域，而在染色体图标上滚动（在交互热图区域的底部和右侧可见）将垂直或水平移动显示区域。

双击进行缩放。您还可以按住 ALT 键并绘制感兴趣区域的框来进行缩放。

注释面板

这是 2D 注释的快速访问面板（也可通过视图菜单访问）。在 JBAT 中，2D 注释起着重要作用，因为它们用于建立参考序列与交互热图上的位点之间的对应关系。

组装注释显示为叠加在交互热图上的正方形，沿对角线显示，在组装中由个别序列的线性坐标定义，并且具有与序列长度相等的边长。（如果组装包含多个染色体，则计算“全局”组装坐标，即沿特定染色体的 1D 坐标通过前面染色体的总长度进行移动。）

因此，每个注释正方形都会包围显示给定序列内部位点接触频率的交互热图区域。从正方形的上中和下中部分，以及左中和右中部分，可以看出给定序列上的位点与其他参考序列上的位点的交互频率。

在加载.assembly 文件时，注释面板将自动填充三个图层：Scaffold（用于显示脚手架边界，默认情况下显示为绿色），Chr（用于显示染色体边界，蓝色），以及 Edit（用于临时注释，黄色）。在交互图上由**单个蓝色正方形围住的两个绿色正方形表示应将脚手架序列连接成单个序列为染色体序列**。脚手架是 JBAT 中用户交互的主要焦点。可以使用 **Shift + 单击** 选择单个脚手架，从而可以执行以下操作：**剪切（将脚手架切割成较**

小的片段，用于纠正连接错误），移动（将脚手架移动到组装中的新位置，用于纠正易位）和翻转（用反向互补序列替换原先序列，用于纠正错误组装产生的倒位）。易位和倒位也可以对连续脚手架的或单个脚手架进行操作（**使用 Shift + 拖动以选择连续脚手架**）。选择表示为临时的黄色注释正方形（带有黑色高亮显示），包围选定的脚手架（参见图 6）。除了上述三个操作之外，还可以在任何一对给定的**脚手架之间添加和删除染色体边界**。

右侧面板

右侧面板包含了迷你地图和信息面板。迷你地图用于帮助用户在基因组组装（或者在使用原始 Juicebox 时是染色体）中确定自己的方位。可以通过在迷你地图中移动透明的方块来快速定位到主要交互热图窗口中相应位置。

当鼠标移动到热图上时，面板中的文本会更新，显示有关特定 Hi-C 像素的信息。当 2D 注释等特征叠加在地图上时，当鼠标悬停在特征上方时，会显示有关它们的详细信息。在组装工具的上下文中，这些信息包括 scaffold 的名称和 ID（在.assembly 格式中分配给 scaffold 的数字），scaffold 的方向（以.assembly 文件中表示的“有符号”ID 的形式，正 ID 反映序列应按照其在 draft.fasta 文件中的方向被合并到最终组装中，而负 ID 则需要进行反向互补），以及与注释对应的序列的当前 1D 坐标。还显示染色体（超级 scaffold）的 ID 和坐标。当存在选择活动时，信息面板会被冻结，显示与所选 scaffold 相关的数据（见图 6）。

3、在 JBAT 中修改组装结果

Juicebox 组装工具允许对基因组组装中的错误进行可视化识别和交互式修正。修正以.assembly 文件的形式存储，并生成用户引入的更改的的参考.fasta 文件。在本节中，我们描述了四种常见类型的组装错误以及 JBAT 中相应的纠正操作。

1）错配

JBAT 中的“错配”定义为草图输入的 scaffold 或 contig 中的错配，即不在染色体上紧密相邻的区域（甚至属于不同的染色体）被拼接在一起。错配通常表现为明亮的接触频率带中断，带内有一个绿色方形注释（参见图 8，左图），反映了草稿中暗示了附近序列的物理接近性缺乏。纠正错误的方法与 3D-DNA 类似（参见图 7）：通过切除断点

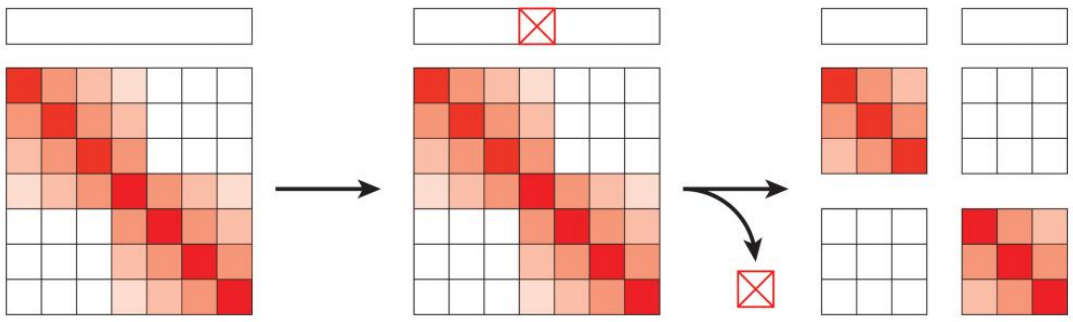


FIGURE 7. Misjoin correction, conceptual scheme. A problematic region that lies inside an input scaffold (a bin marked with an X), gets excised resulting in two internally consistent fragments of the original input scaffold (labeled as 'fragments'). The third fragment that spans a misassembled region is labeled as 'debris'. The debirs fragments are moved to the very end of the genome assembly, where they are kept for future reference. From (Dudchenko et al. 2017).

周围的一个小区域，将原始的 scaffold 切割成三个片段。该过程会得到两个内部一致的原始草稿 scaffold 片段。被切除的片段被标记为“残渣”并移到组装的最末端。残渣片段不能再进行切割。

为了修复错配，通过 **Shift + 单击** 选择有问题的 scaffold（选定的 scaffold 周围会出现黑黄色的高亮显示）。然后，将鼠标**移动到热图的对角线**上，会出现一个**剪刀形状**的情境指针，提示用户“切割”scaffold（参见图 8，中间图）。同时出现的**虚线黄色编辑区域**将显示一旦点击剪刀光标，将从序列中切割掉的区域（**此时分辨率应该大一些，尽量切割少一些**）。使用滚动条增加或减少接近断点的区域的大小。注意，只有在选择只包括一个 scaffold 时才会出现“切割”光标提示。当对预期切割满意时，点击鼠标：这将导致原始 scaffold 变成两个新的 scaffold（相应地，在对角线上原来的一个位置上出现两个新的绿色 scaffold 注释），参见图 8，右图。小的“残渣”scaffold 被移到组装的最末端，并保留以供将来参考。

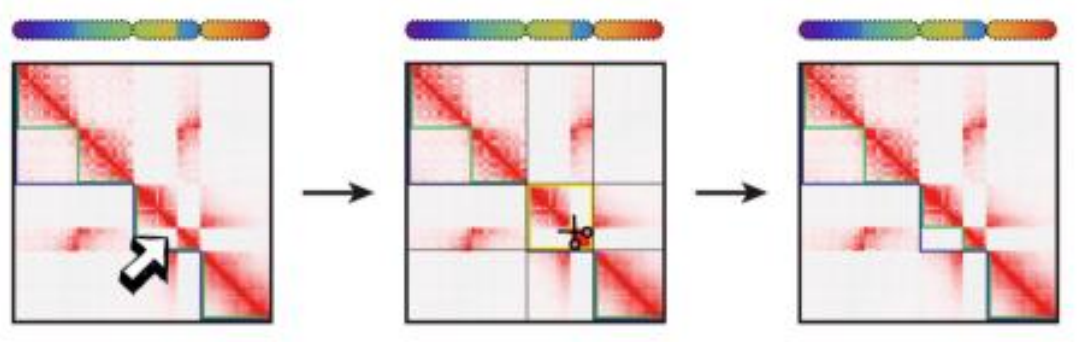


FIGURE 8. Misjoin correction example. Left panel shows a typical anomalous Hi-C signal associated with a misjoin: a draft scaffold containing a point along the diagonal such that the signal to the lower-left and the upper -right from the point is extremely depleted, reflecting the lack of physical proximity between sequences upstream and downstream of the point. In order to correct for the error select the problematic scaffold, move the mouse towards the diagonal and position the appearing situational pointer in such a way that the dotted 'debris' annotation is encompassing the breakpoint, see middle panel. (One can use scroll to adjust the size of the region flanking the breakpoint.) Once the mouse is clicked, the small debris sequence is excised and moved to the very end of the assembly, leaving behind two internally consistent scaffolds in place of the original one, available for downstream user interaction (right panel). The assembly at this point consists of 5 scaffolds joined into 3 chromosomes and a very small 6th debris scaffold. From (Dudchenko et al., 2017) with modifications.

2）易位

基因组组装中常见的错误类型是易位。在 JBAT 的上下文中，修复易位意味着改变参考基因组组装中 scaffold 的顺序。易位错误通常表现为垂直和对称的水平蝴蝶结状富集模式，**远离对角线，表示参考基因组组装中不相邻的两个序列显示出异常高水平的 Hi-C 交互**。蝴蝶结模式中最高富集的点指示需要“组合在一起”的序列（参见图 9，左图）。要重新排序 scaffold，请选择要易位的序列，并将鼠标移动到两个相邻 scaffold 之间的新位置。这将导致**箭头情境指针**的出现（参见图 9，中间图）。（注意，如果两个 scaffold 在组装中不是连续的，或者无法从鼠标提示明确解释建议插入的位置，则不会出现提示。当尝试在两个 scaffold 之间插入具有小元素的元素时，这可能导致看似无响应的行为。在这种情况下，缩放视图以明确指示选定 scaffold 的新位置。）**一旦点击指针，scaffold 将移动到一个新位置（由箭头指示）**（参见图 9，右图）。请注意，**易位可以在由多个 scaffold 组成的选择上执行**。

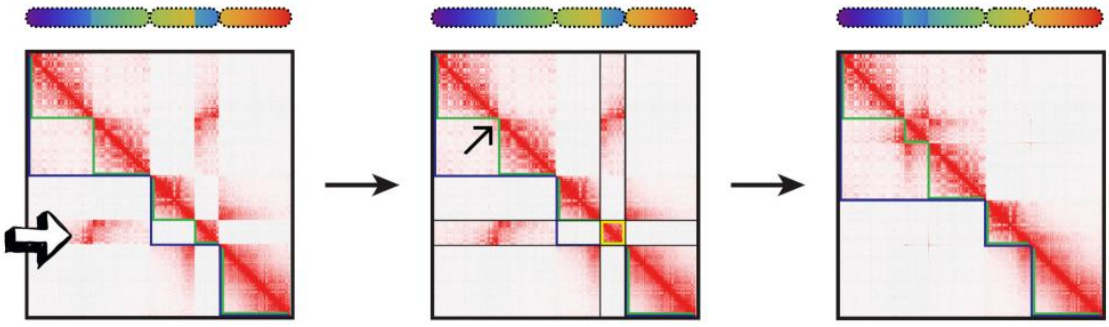


FIGURE 9. Translocation correction example. The left panel shows a typical anomalous Hi-C signal associated with a translocation: a bowtie-ish enrichment motif away from the diagonal indicating that loci not in close 1D proximity according to the reference assembly display unexpectedly high frequency of contact. The middle panel shows a situational pointer indicating a possible correction move: once clicked, the move will result in a relocation of the selected sequence (indicated by the yellow and black highlight) in between two scaffolds as indicated by the arrow. The right panel shows the result of the move, with both the reference and the heatmap up[illegible] the change. The anomalous off-diagonal motif is now gone. From (Dudchenko et al. 2018), with modifications.

3）倒位

基因组组装中的另一种常见错误是倒位错误。在JBAT中，修复倒位是改变由**单个序列或一组**连续序列表示的序列的方向。这种错误通常表现为与对角线平行的**蝴蝶结富集模式**。模式的中点定义了需要倒位的基因组组装区域（见图10，左侧）。为了修复倒位，选择囊括有问题区域的序列，将鼠标移动到选择的**底部左侧或顶部右侧**。在中间面板中会出现用于**翻转序列的情景指针**，如图10所示。鼠标单击后，参考基因组组装中选择的支架序列将被替换为其反向互补序列。交互图实时更新以反映变化（见图10，右侧）。

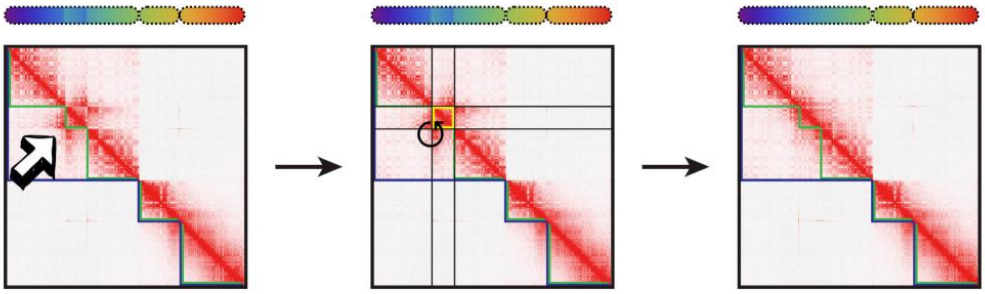


FIGURE 10. Inversion correction example. The arrow in the left panel highlights the anomalous contact motif characteristic of the inversion: a bowtie enrichment motif, parallel to the diagonal, indicating that the sequence at the beginning of the region flanked by the motif is in close proximity with the scaffold immediately downstream from the region and, vice versa, the sequence at the very end of the flanked region is in close proximity with the scaffold immediately upstream. In order to correct for the inversion, select the scaffold or scaffolds flanked by the motif and move the mouse towards either the bottom-left or the top-right corner. A prompt will appear as in the middle panel for inverting the selected sequence. Once clicked, instructions for reverse-complementing the relevant sequence in the reference genome assembly are cached, and the map is updated to reflect the change to the reference (right panel). Th[illegible] bowtie motif is no more. From (Dudchenko et al. 2018), with modifications.

4）染色体边界

调整染色体边界是可以通过Juicebox Assembly Tools进行的另一种组装优化类型。在JBAT中，可以在序列之间**添加或删除染色体边界**，以确定染色体得正确组成。考虑来自图11的示例。在这个例子中，尽管交互图看起来由两个代表两个染色体领域的大型富集方块组成，但组装被分割为三个染色体（三个蓝色注释）。为了移除多余的边界

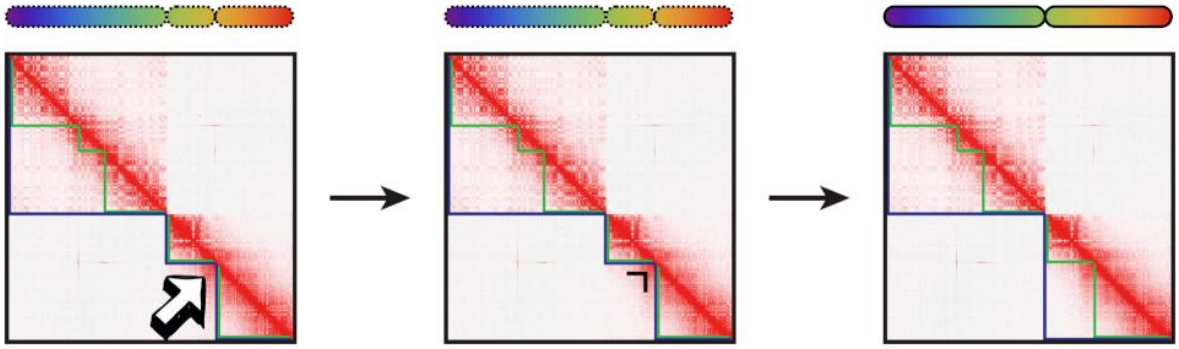


FIGURE 11. Adjusting chromosome boundaries with JBAT. In this case, the tentative assembly (left panel) consists of three chromosomes while the Hi-C signal suggests two chromosomes. The anomalous signal is in some sense opposite to the one observed for misjoin: sequences that belong to two different chromosomes according to the assembly and are hence not in close 1D proximity nonetheless exhibit strong interaction according to the Hi-C signal, indicated by a strong, unbroken diagonal (see arrow in the left panel and compare to the signal associated with the 'true' chromosome boundary upstream from the pointer). The extra chromosome boundary should as such be removed. In order to perform the action move the mouse pointer towards the diagonal, in between the two scaffolds flanking the breakpoint: a situational prompt will appear as shown in the middle panel. On click the chromosome boundary is removed, reflected by the changes in the blue chromosome layer annotations as in the right panel.

（在左侧面板中由箭头指示），将鼠标光标移动到夹在染色体断点两侧的序列之间（不应选择任何支架）：**一个情景指针会出现，用于切换染色体边界**，如**图 11（中间）**所示。点击后，与边界相邻的染色体组中的序列将合并为单个组，可以通过蓝色染色体层注释的变化反映出来（见图 11，右侧面板）。新的组装现在包含两个染色体。请注意，可以在任意两个序列之间添加和移除染色体边界。

5）右键菜单

JBAT 中额外添加了三个菜单项：“移至废料”（Move to debrish)、“撤销”(Undo)和“重做”(Redo)。

“移至废料”菜单项用于将一组序列移动到组装的最后。当处理稀疏或模糊信号或者杂合位点时候，这可能非常有用。

“撤销”菜单项用于撤销最后一次调图操作，“重做”用于撤销上次的撤销。这些操作也可以使用快捷键完成：撤消为 Command-U（Windows 上为 Ctrl-U），重做为 Command-R（Windows 上为 Ctrl-R）。

4、JBAT 调图后生成 Fasta

使用 Assembly 菜单项“导出组装”来保存 JBAT 调图后结果。以修改后的 review.assembly 文件的形式保存。然后转换为 fasta 的最简单方法是使用作为 3D-DNA，运行以下脚本：

```
run-asm-pipeline-post-review.sh -r AT.0.review.assembly AT.fa aligned/merged_nodups.txt -g 100
```

该脚本不仅会生成最终的 fasta 文件（带有“FINAL”后缀），还会重新构建最终的.hic 图进行质量控制。

2.4 AllHIC 组装

多倍体基因组的主要问题在于Hi-C信号通常在等位型间被检测到，而任何现有的Hi-C脚手架化程序都会连接等位型。为了解决这个问题，开发了一种新的Hi-C脚手架化流程，称为ALLHiC，专门针对多倍体或者两套基因组。ALLHiC流程包括共5个步骤：修剪、分区、救援、优化和构建。

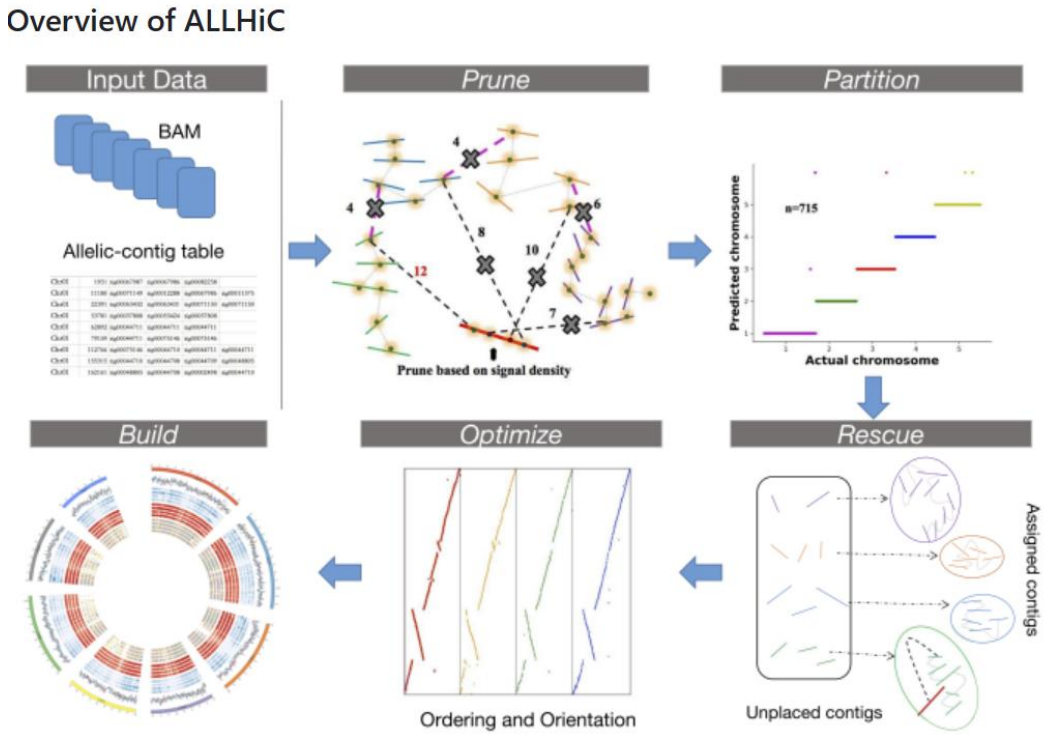


图1. ALLHiC算法的主要步骤概述。新发布的ALLHiC流程包含共5个功能：修剪、分区、救援、优化和构建。简要地说，修剪步骤移除等位连结，使同源染色体更容易单独分离。分区功能根据Hi-C所提示的连结将修剪后的BAM文件中的连结的连续基因组聚类成为同一个预设数量的分区中，这些分区很可能沿着同一个同源染色体。救援功能从原始未修剪的BAM文件中搜索不参与分区步骤的基因组，并根据Hi-C信号密度将它们分配到特定的聚类中。优化步骤对每个分区进行排序和方向优化。最后，构建步骤通过连接基因组装配并在基因组间添加间隔以生成最终的FASTA格式基因组版本。

修剪功能首先允许我们检测等位基因组，可以通过基于良好组装的近缘物种或已组装的单倍体基因组来完成。将等位基因组间的信号（归一化的Hi-C读数）从输入BAM文件中移除。在多倍体基因组组装中，共享高相似性的等位型很可能被折叠。折叠区域与附近的分相等位型之间的信号会导致嵌合的脚手架。在修剪步骤中，只保留折叠的基因组与分相基因组之间的最强连结。

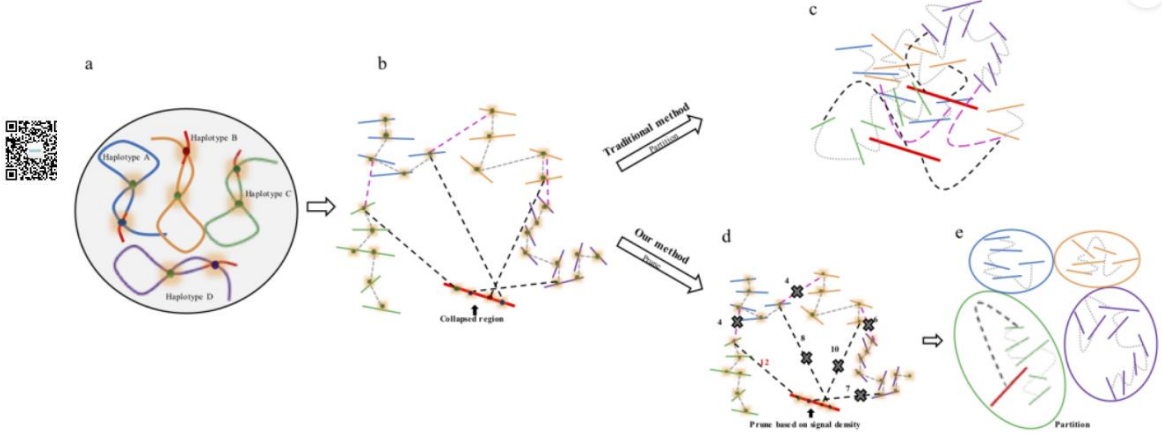


图 2. 多倍体基因组中的 Hi-C 脚手架问题描述及修剪方法的应用。(a) 自动四倍体基因组的示意图。四个同源染色体用不同颜色表示（依次为蓝色、橙色、绿色和紫色）。染色体中红色区域表示具有高相似性的序列。(b) 在自动四倍体基因组中检测 Hi-C 信号。黑色虚线表示折叠区域与未折叠基因组的 Hi-C 信号。粉色虚线表示跨等位型的 Hi-C 连结，灰色虚线表示同一等位型内的 Hi-C 连结。在组装过程中，由于高序列相似性，红色区域将被折叠；而其他区域如果存在丰富的变异，则将被分离成不同的基因组。由于折叠区域与不同等位型的基因组之间有物理联系，Hi-C 信号将在折叠区域与所有其他未折叠基因组之间被检测到。(c) 传统的 Hi-C 脚手架方法将检测到的信号聚类为不同等位型的基因组以及折叠区域。(d) 修剪 Hi-C 信号：1- 移除等位区域间的信号；2- 仅保留折叠区域与未折叠基因组间的最强信号。(e) 基于修剪后的 Hi-C 信息进行分区。根据修剪结果，理想情况下将基因组分为不同的组。

2.4.1 安装

git clone https://github.com/tangerzhang/ALLHiC

cd ALLHiC

chmod +x bin/*

chmod +x scripts/*

export PATH=/your/path/to/ALLHiC/scripts/:/your/path/to/ALLHiC/bin/:$PATH

依赖项

以下是 ALLHiC 流程中将使用的第三方程序列表：

samtools v1.9+

bedtools

matplotlib v2.0+

2.4.2 运行 ALLHiC

数据准备:同一物种或者近源物种染色体级别基因组和基因注释，并且要求该物种必须是二倍体;具有一定覆盖度的 Hi-C reads;具有合理 N50 和基因注释的 contig 级组装结果

1、将 Hi-C reads 映射到 draft 组装结果

使用 bwa index 和 samtools faidx 为 draft 基因组组装结果建立索引：

```
bwa index -a bwtsw draft.asm.fasta
samtools faidx draft.asm.fasta
```

2、将 Hi-C reads 与 draft 组装结果进行比对：

bwa aln -t 24 draft.asm.fasta CRR302669_R1.fastq.gz > sample_R1.sai

bwa aln -t 24 draft.asm.fasta CRR302669_R2.fastq.gz > sample_R2.sai

bwa sampe -r "@RG\tID:sample\tSM:sample" draft.asm.fasta sample_R1.sai sample_R2.sai CRR302669_R1.fastq.gz CRR302669_R2.fastq.gz > sample.bwa_aln.sam

PreprocessSAMs.pl sample.bwa_aln.sam draft.asm.fasta MBOI

filterBAM_forHiC.pl sample.bwa_aln.REduced.paired_only.bam sample.clean.sam

samtools view -bt draft.asm.fasta.fai sample.clean.sam > sample.clean.bam

3、修剪

此步骤是可选的，对于获得单体型的组装结果，必须采用此步骤，进行修剪。如果contig组装结果，只是一套基因组，则可以略过此步骤。使用此步骤必须准备 Allele.ctg.table 文件，来修剪 1）连接等位基因的信号和 2）来自 BAM 文件的弱信号。用法：

ALLHiC_prune -i Allele.ctg.table -b sample.clean.bam -r draft.asm.fasta

参数：

-h：帮助和用法

-i：Allele.ctg.table

-b：输入的 BAM 文件

-r：draft.asm.fasta

提示：关于如何识别等位基因连通子的详细信息可以在以下链接中找到：

https://github.com/tangerzhang/ALLHiC/wiki/ALLHiC:-identify-allelic-contigs

本处以采用 gmap 为例来生成 Allele.ctg.table，它不需要的目标基因组的注释。这个方法适合于大多数人，一般做 Hi-C 辅助基因组组装时肯定没有进行基因预测没有 gff 以及 cds、pep 序列，所以就需要进行下面的操作。

1）运行 gmap 得到一个 gff3 文件

gmap_build -D ./ -d DB draft.asm.fasta

gmap -D . -d DB -t 12 -f 2 -n 2 AT.cds.fa > gmap.gff3

注意：

1.target.genome 是我们组织的多倍体基因组组装的 contig 水平

2.$N 是基因组的倍性，例如 $N=4 如果它是四倍体

3.reference.cds.fasta 是二倍体种的基因组的编码序列

2）生成 allelic.ctg.table

gmap2AlleleTable.pl gmap.gff3

4、分区

将 contig 分配到预定义数量的染色体组中(提高分组准确性，可以将 contig 切成等长来操作）。

用法：

ALLHiC_partition -b prunning.bam -r draft.asm.fasta -e GATC -k 10

参数：

-h：帮助和用法。

-b：修剪后的 BAM 文件（可选，默认为 prunning.bam）

-r：draft.sam.fasta

-e：酶位点（HindIII：AAGCTT；MboI：GATC）

-k：染色体条数（用户定义的 K 值）

-m：最小限制位点数（默认为 25）

5、恢复

将未放置的 contig 分配到分区的聚类中

用法：

```
ALLHiC_rescue -b sample.clean.bam -r draft.asm.fasta -c prunning.clusters.txt -i prunning.counts_GATC.txt
```

参数：

-h：帮助和用法。

-b：样品的未修剪 BAM 文件

-r：draft.sam.fasta

-c：clusters.txt

-i：counts_RE.txt

提示：

clusters.txt（-c）和 counts_RE.txt（-i）可以由 allhic extract 生成。

clusters.txt 的格式如下所示：

```
#Group  nContigs    Contigs
48g1    506         tig0000150 tig0000151 tig0000152
48g2    692         tig0000097 tig0000114 tig0000231
48g3    683         tig0000015 tig0000035 tig0000235
```

第一列是组名。第二列是锚定在该组中的 contig 数量，第三列列出了该组中的 contig 明总。

6、优化

对每个组进行排序和定向。用法：

```
allhic extract sample.clean.bam draft.asm.fasta --RE GATC
allhic optimize group1.txt sample.clean.clm
allhic optimize group2.txt sample.clean.clm
allhic optimize group3.txt sample.clean.clm
allhic optimize group4.txt sample.clean.clm
```

```
allhic optimize group5.txt sample.clean.clm
```

参数：

allhic extract

输入文件：

bam 文件和 contig 序列

选项：

--RE value 限制位点模式（默认为"GATC"）

allhic optimize

输入文件：

**counts_RE.txt - 每个聚类组中每个 contig 的限制位点计数，可以通过 allhic extract 生成。格式如下：

```
#Contig    RECounts    Length
tig0000001      572     157863
tig0000002      143     33000
tig0000003      231     60910
tig0000004      3789    1044000
tig0000005      646     166098
tig0000006      67      15000
tig0000007      1094    319000
```

**clmfile - 记录两个 contig 之间的 Hi-C 连接的基本信息，包括潜在的排序和定向、支持的 read 数量和成对末端 read 的距离。该文件可以由 allhic 提取访问。

选项：

--skipGA 跳过 GA 步骤

--resume 从现有的 tour 文件恢复

--seed value 随机种子（默认为 42）

--npop value 种群大小（默认为 100）

--ngen value 收敛的迭代次数（默认为 5000）

--mutpb value GA 中的突变概率（默认为 0.2）

7、构建

将 tour 格式转换为 fasta 序列和 agp 位置文件。用法：

```
ALLHiC_build draft.asm.fasta
```

输入文件：

draft.asm.fasta - contig 级别的组装结果

此步骤将输出一系列超级 scaffold 和未锚定的 contig。检查 groups.asm.fasta 文件。

8、绘图

我们将生成染色质接触矩阵以评估基因组脚手架。用法：

```
ALLHiC_plot sample.clean.bam groups.agp chrn.list 500k pdf
```

输入文件：

**bam 文件 - 映射的 bam 文件

**agp 文件 - 每个 Hi-C 组中连通子的放置位置，可以由 ALLHiC_build 生成

**chrn.list - 组名和长度的列表。该文件的格式如下：

group1 125000

group2 159000

**bin size - 热图的 bin 大小

**ext - 绘图文件的扩展名，例如 pdf

2.5 基因组组装质量评估

进行回比，BUSCO 评估，完整性评估等！

2.5.1 GAEP 评估

GAEP 是一个用于评估基因组组装的流程。

1、安装

git clone https://github.com/zy-optimistic/GAEP.git

cd GAEP

./gaep

2、使用方法

gaep <command> [options]

pipe (NGS,TGS,trans) 根据输入数据让 GAEP 确定要执行的模块

stat 报告基因组的基本信息

macc (NGS) 基于读取映射的碱基准确性

kacc (NGS) 基于 K-mer 的碱基准确性（merqury）

bkp (TGS) 检测到的错配断点

snvcov (NGS,TGS) SNV 覆盖度点图

busco 运行 busco v5

运行示例

#您可以在 config.txt 中列出您的数据和依赖项。config.txt 的模板在 GAEP/config/目录中。

```
gaep stat test.p_ctg.fa
```

```
gaep pipe --lr TGS.fq.gz --type ccs --sr1 NGS_1.fq.gz --sr2 NGS_2.fq.gz --tr1 RNAseq_1.fq.gz --tr2 RNAseq_2.fq.gz -d results4 -o test1 -t 10 -r test.fa -c /mnt/RAID-5/MD0/bioinformatics/soft/GAEP/config/config.txt -l /mnt/RAID-5/MD0/bioinformatics/soft/busco-5.5.0/busco_downloads/lineages/embryophyta_odb10/
```

2.5.2 BUSCO 分析

https://busco.ezlab.org/busco_userguide.html#manual-installation

1、BUSCO 安装

完整安装 BUSCO 需要 Python 3.3+（从 v4 开始不支持 2.7），BioPython，pandas，BBMap，tBLASTn 2.2+，Augustus 3.2+，Prodigal，Metaeuk，HMMER3.1+，SEPP 和 R + ggplot2 用于绘制辅助脚本。其中一些工具仅用于分析特定类型的物种和输入数据，或者用于特定的运行模式。

(1)建一个新的 conda 环境，环境里装一个 Py37

```
conda create -n Py37 python=3.7
conda activate Py37
```

(2) 安装 augustus，metaeuk（注意添加环境变量）

```
conda install -c bioconda augustus
conda install -c bioconda metaeuk
```

(3) 安装 hmmer

```
conda install -c bioconda hmmer
```

(4) 安装 BBmap

下载即可用 https://sourceforge.net/projects/bbmap/

(5) 安装 biopython1.77

```
conda install -c bioconda biopython=1.77
```

(6)安装 BUSCO

下载自 https://gitlab.com/ezlab/busco/-/releases#5.4.7

2、BUSCO 使用

```
#必选参数
busco -i [SEQUENCE_FILE] -l [LINEAGE] -o [OUTPUT_NAME] -m [MODE] [OTHER OPTIONS]
```

除非在配置文件中提供，否则必选参数

-i 或--in 定义要分析的输入文件，可以是核酸 FASTA 文件或蛋白质 FASTA 文件，具体取决于 BUSCO 模式。从 v5.1.0 开始，输入参数现在还可以是包含 FASTA 文件的目录，以批处理模式运行。

-o 或--out 定义将包含所有结果、日志和中间数据的文件夹。

-m 或--mode 设置评估模式：genome（基因组）、proteins（蛋白质）、transcriptome（转录组）。

-l 或--lineage_dataset

它可以是数据集名称，例如 bacteria_odb10，也可以是路径，例如./bacteria_odb10 或/home/user/bacteria_odb10。在前一种情况下（推荐用法），BUSCO 将自动下载并版本化相应的数据集。在后一种情况下，将使用给定路径中找到的数据集。如果运行自动选择谱系，则可以忽略谱系。

基因组模式的默认基因预测工具是 Metaeuk（从 v5.0.0 开始）。以前的默认值是 Augustus。存在使用 Augustus 或 Miniprot 作为 Metaeuk 替代方案的选项。要使用 Augustus，请使用--augustus 标志。要使用 Miniprot，请使用--miniprot 标志。Miniprot 在

5.5.0 版本中可用于测试。

＃在配置文件中管理运行参数

BUSCO 将尝试使用环境中可用的依赖项。您可以通过传递指定依赖项路径的配置文件来覆盖此设置。在 config/子文件夹中，提供了一个 config.ini 模板文件。在此文件中，您可以声明与您的机器上相匹配的所有第三方组件的路径。要激活此配置文件，请设置环境变量 BUSCO_CONFIG_FILE，并将其设置为文件的路径，如下所示：

export BUSCO_CONFIG_FILE="/path/to/myconfig.ini"

或者，您可以使用--config /path/to/config.ini 命令行选项传递配置文件的路径。这对于在不同配置之间进行切换或在专用文件中管理每次运行的参数非常有用。

＃谱系数据集

BUSCO 利用特定类群的信息在分析的序列中识别 BUSCO 基因。用户可以指定它，也可以在细菌和古菌的情况下自动选择。要打印完整列表：

busco --list-datasets

数据库序列的集合（或称为谱系数据集）是用于 BUSCO 基因识别的重要资源。每个谱系数据集都是针对特定类群（例如细菌、真菌、植物等）进行整理和优化的。这些数据集包含了已知的核心基因组成，用于评估给定序列的完整性和准确性。

通过使用适当的谱系数据集，BUSCO 可以根据输入序列的相似性和匹配度来识别和分类基因，并提供关于序列完整性和质量的评估结果。用户可以自行选择适合其研究对象的谱系数据集，或者在细菌和古菌的情况下，也可以让 BUSCO 自动选择适当的谱系数据集。

总之，谱系数据集是 BUSCO 用于基因识别和序列评估的重要组成部分，它提供了与特定类群相关的关键基因序列，有助于评估给定序列的完整性和准确性。

基因组模式：评估基因组组装

需要使用 Prodigal（非真核生物）或 Metaeuk（真核生物），以及 HMMER。如果输入--augustus 命令行标志，则可以使用 tBLASTn 和 Augustus 的组合替代 Metaeuk。

busco -m genome -i 输入核苷酸文件 -o 输出文件 -l 物种线性参考库

busco -m genome -i final.p_ctg.fa -o result -l embryophyta_odb10 -c 10

#蛋白质模式：评估基因集

需要使用 HMMER。

busco -m protein -i 输入氨基酸文件 -o 输出文件 -l 物种线性参考库

busco -m protein -i AT.longest.protein.fa -o result_protein -embryophyta_odb10 -c 10

#转录组模式：评估组装的转录本

需要使用 tBLASTn（原核生物）或 Metaeuk（真核生物），以及 HMMER。

busco -m transcriptome -i 输入核苷酸文件 -o 输出文件 -l 物种线性参考库

#自动选择线性参考库

busco -m 模式 -i 输入文件 -o 输出文件 --auto-lineage

或者忽略真核生物以减少运行时间，如果与实验目标相符。

busco -m 模式 -i 输入文件 -o 输出文件 --auto-lineage-prok

或者忽略非真核生物以减少运行时间，如果与实验目标相符。

busco -m 模式 -i 输入文件 -o 输出文件 --auto-lineage-euk

3 解读结果

BUSCO 旨在定量评估基因组组装、转录组或注释基因集的完整性，以预期的基因内容为依据。结果被简化为完整且单拷贝、完整且重复、碎片化或丢失的 BUSCO。

BUSCO 完整性结果只有在您的生物学背景下才有意义。您需要了解缺失或重复基因是生物学原因还是技术原因。例如，高水平的重复可能是由最近的整个复制事件（生物学原因）或二倍体组装（存在冗余，技术原因）解释的。未经过滤的转录组和蛋白质集(可变剪接)将导致高比例的重复。因此，在进行 BUSCO 分析之前，应对其进行过滤。最后，在转录组实验中专注于特定组织或特定生命阶段和条件不太可能产生完整的 BUSCO 转录组。在这种情况下，您要追求的是样本之间的一致性。

#完整

如果发现为完整，无论是单拷贝还是重复，BUSCO 匹配得分都在预期范围内，并且长度对齐到 BUSCO profile 的预期范围内。如果实际上在输入数据集中不存在同源基因，或是同源基因只部分存在（高度碎片化），并且存在高相似性的全长同源物，那么这个同源物可能被错误地标识为完整的 BUSCO。得分阈值经过优化，以最小化这种可能性，但仍可能发生。

#碎片化

如果发现为碎片化，BUSCO 匹配得分在得分范围内，但不在长度对齐到 BUSCO profile 的范围内。对于转录组或注释基因集，这表示不完整的转录本或基因模型。对于基因组组装而言，这可能表明基因只部分存在，或者序列搜索和基因预测步骤未能生成全长基因模型，尽管完整基因可能确实存在于组装中。在运行真核数据集时，产生这种碎片化结果的匹配区域会进行“第二轮预测”，使用在已找到完整的 BUSCO 上基因，作为训练集合，然后进行第二轮序列搜索和基因预测，但仍可能无法恢复整个基因。因此，一些基因组组装评估中的碎片化 BUSCO 可能是完整的，但只是过于差异化或具有非常复杂的基因结构，使其很难完整地定位和预测。

#丢失

如果发现为丢失，要么根本没有显著匹配，要么 BUSCO 匹配得分低于 BUSCO profile 的得分范围。对于转录组或注释基因集而言，这表示这些同源物确实丢失，或者转录本或基因模型不完整/碎片化以至于无法满足被视为碎片化的标准。对于基因组组装而言，这可能表明这些同源物确实丢失，或者序列搜索步骤未能识别出任何显著匹配，或者基因预测步骤未能生成即使是部分基因模型，该模型可能已被识别为碎片化的 BUSCO 匹配。与碎片一样，运行真核数据集时，第一轮后仍然丢失的 BUSCO 会进行第二轮，使用在已找到完整的 BUSCO 上训练的参数进行第二轮序列搜索和基因预测，但仍可能无法恢复基因。因此，一些基因组组装评估中丢失的 BUSCO 可能部分存在，甚至可能（但不太可能）完

整，但它们过于差异化或具有非常复杂的基因结构，使其很难正确或部分地定位和预测。

#自动选择：多个领域中的匹配和污染

自动系谱选择过程在域界线数据集上运行 BUSCO，包括古细菌、细菌和真核生物。一旦选择了最佳的域，BUSCO 会自动尝试找到最适合使用的 BUSCO 数据集，该数据集基于系统发育位置进行选择。只有在使用适当的数据集时，即数据集属于要测试的物种系谱，BUSCO 评估才有效。由于多个领域之间存在重叠标记物/虚假匹配，BUSCO 在其他领域中的匹配并不意味着您的基因组/蛋白质组中包含来自该领域的序列。然而，在多个领域中获得较高的 BUSCO 分数可能有助于识别可能的污染

4、绘图

scripts 下面的 **generate_plot.py** 脚本允许用户快速查看他们的 BUSCO 摘要结果(short_summary 文件)，以易于理解的条形图显示。 scrigenerate_plot.py 使用 R（https://www.r-project.org/）和 ggplot2（http://ggplot2.org/）对 BUSCO 运行进行汇总，以便进行并排比较。该脚本生成一个 PNG 图像（如果同时可用 R 和 ggplot2），以及一个 R 源代码文件，可以在另一台同时可用 R 和 ggplot2 的计算机上运行，或者可以编辑该文件以完全自定义生成的条形图（颜色，标签，字体，坐标轴等）。

用法：python3 generate_plot.py -wd [工作目录] [其他选项]

BUSCO 绘图工具。

必填参数：

-wd PATH，--working_directory PATH

定义工作目录的位置

可选参数：

-rt RUN_TYPE，--run_type RUN_TYPE

要使用的摘要类型，“generic”或“specific”

--no_r 避免运行 R。它只会在工作目录中创建 R 脚本文件

要运行 generate_plot.py，请首先创建一个文件夹，例如

```
mkdir BUSCO_summaries
```

然后将每个要绘制的运行的 BUSCO 摘要文件复制到该文件夹中。

然后简单地运行该脚本，将所创建的包含要绘制摘要的文件夹的名称（如果您不在同一个工作目录中，请给出完整路径）作为参数。

```
python3 generate_plot.py  – wd BUSCO_summaries
```

生成的 PNG 图像及其对应的 R 源代码文件将生成在包含 BUSCO 摘要的同一文件夹中。默认情况下，运行名称用作每个绘制结果的标签，这是从摘要文件名中自动提取的：因此，对于 short_summary.generic.lineage_odb10.XX1.txt，标签将为 XX1。您可以根据需要修改此标签，只要保持命名惯例：short_summary.generic.lineage_odb10.[edit_name_here].txt，或者您可以简单地编辑 R 源代码文件以更改任何绘制参数，并在您的 R 环境中手动运行代码生成个性化的条形图。

2.5.3 Merqury

通常，基因组组装项目会提供已经组装好的个体的 illumina 全基因组测序数据。可以利

用该数据的 k-mer 谱图来独立评估基因组组装质量，而无需高质量参考序列。Merqury 提供了一套工具来实现这个目的。

1、软件安装

在新的 conda 环境中运行以下命令：

conda install -c conda-forge -c bioconda merqury

或者，如果已安装不同版本的 jdk 或者想为 merqury 使用一个单独的 conda 环境：

conda create -n merqury -c conda-forge -c bioconda merqury openjdk=11

然后，在使用之前需要激活 merqury 环境：

conda activate merqury

测试运行

Rscript $MERQURY/plot/plot_spectra_cn.R --help

2、运行 Merqury

ln -s $MERQURY/merqury.sh# 链接 merqury

merqury.sh <read-db.meryl> [<mat.meryl> <pat.meryl>] <asm1.fasta> [asm2.fasta] <out>

用法： merqury.sh <read-db.meryl> [<mat.meryl> <pat.meryl>] <asm1.fasta> [asm2.fasta] <out>

<read-db.meryl>: 二代测序 read 的 k-mer 计数

<mat.meryl>: 母源单倍体的 k-mer 计数（例如 mat.only.meryl 或 mat.hapmer.meryl）

<pat.meryl>: 父源单倍体的 k-mer 计数（例如 pat.only.meryl 或 pat.hapmer.meryl）

<asm1.fasta>: 组装的 fasta 文件（例如 pri.fasta、hap1.fasta 或 maternal.fasta）

[asm2.fasta]: 附加的 fasta 文件（例如 alt.fasta、hap2.fasta 或 paternal.fasta）

*将生成 asm1.meryl 和 asm2.meryl，避免使用与 hap-mer 数据库相同的名称

<out>: 输出前缀

1）制作 meryl dbs

```
sh best_k.sh 120000000
meryl k=18 count *.fq.gz output AT.meryl
meryl k=18 count compress *.fq.gz output AT.meryl
```

2）我有一个组装结果（伪单倍体或混合单倍体）

```
# 我没有 hap-mer
merqury.sh AT.meryl/ test.fa  out_prefix
# 我有 hap-mer
merqury.sh read-db.meryl mat.meryl pat.meryl asm1.fasta out_prefix
```

3）我有两个组装结果（二倍体）

```
# 我没有 hap-mer
merqury.sh read-db.meryl asm1.fasta asm2.fasta out_prefix
# 我有 hap-mer
merqury.sh read-db.meryl mat.meryl pat.meryl asm1.fasta asm2.fasta out_prefix
```

注意，无需为每个组装结果单独运行 merqury。只需提供两个 fasta 文件，Merqury 将为每个文件生成统计数据并组合在一起。

3、结果介绍

1） QV estimation

out_prefix.qv 每列显示的内容如下：

仅在组装结果中找到的 k-mer。

在组装结果和 read 集合中都找到的 k-mer。

QV（质量值）。

错误率。

2） k-mer completeness

k-mer 完整性可以通过恢复的可靠 k-mer 的比例来衡量。可靠的 k-mer 是对错误的低复制数 k-mer 进行过滤的不同 k-mer。

k-mer 完整性 = 在组装结果中找到的可靠 k-mer 数量 / 读取集合中的可靠 k-mer 数量

列解释：

组装结果

用于衡量完整性的 k-mer 集合 - all = read set（稍后会扩展为 hap-kmer）

组装结果中的可靠 k-mer 数量

读取集合中的总可靠 k-mer 数量

完整性（%）

2.5.4 LAI

利用下章 EDTA 结果计算：

```
perl /mnt/RAID-5/MD0/bioinformatics/soft/genome/mambaforge/bin/LAI -genome genome.fa.mod -intact genome.fa.mod.EDTA.raw/LTR/genome.fa.mod.pass.list -all genome.fa.mod.EDTA.anno/genome.fa.mod.out
```

三、基因组注释

3.1 EDTA

这个软件包是为了自动化的全基因组去 de novo TE 注释和评估 TE 库的注释性能而开发的。EDTA 软件包旨在过滤原始 TE 候选集中的假阳性结果，并生成一个高质量的非冗余 TE 库，用于全基因组 TE 注释。对于全基因组注释，您需要使用 EDTA 对每个基因组进行注释，生成一个全基因组库，然后使用全基因组库重新对每个基因组进行注释。该软件包还包含了 panEDTA 的顺序版本。

1、安装

conda create -n EDTA

conda activate EDTA

conda config --env --add channels anaconda --add channels conda-forge --add channels bioconda

conda install -n EDTA -y cd-hit repeatmodeler muscle mdust blast java-jdk perl perl-text-soundex multiprocess regex tensorflow=1.14.0 keras=2.2.4 scikit-learn=0.19.0 biopython pandas glob2 python=3.6 tesorter genericrepeatfinder genometools-genometools ltr_retriever=2.8.7 ltr_finder

单独安装：pip install 'h5py<3.0.0' -i https://pypi.tuna.tsinghua.edu.cn/simple

git clone https://github.com/oushujun/EDTA

./EDTA/EDTA.pl

2、用法

加载环境：source activate /mnt/RAID-5/MD0/bioinformatics/soft/EDTA/envs/EDTA

perl /mnt/RAID-5/MD0/bioinformatics/soft/EDTA/EDTA/EDTA.pl [选项]

--genome [文件]基因组 FASTA 文件。必需。

--species [Rice|Maize|others]指定物种以识别 TIR 候选序列。默认值：others

--step [all|filter|final|anno]指定要运行 EDTA 的步骤。

all: 运行整个流程（默认）

filter: 从原始 TE 开始到结束。

final: 从过滤后的 TE 开始到完成运行。

anno: 在 TE 库构建后执行整个基因组注释/分析。

--overwrite [0|1]如果找到以前的结果，决定是否覆盖（1，重新运行）或不覆盖（0，默认）。

--cds [文件]提供一个包含此基因组或其近源物种的的编码序列（无内含子，UTR 或 TE）的 FASTA 文件。

--curatedlib [文件]提供一个经过精心筛选的库，以确保和已知 TE 的命名和分类一致。此文件中的所有 TE 都将被 100%信任，请仅在此处提供手动筛选的 TE。此选项不是强制性的。如果没有提供文件也可以（默认情况）。

--sensitive [0|1]使用 RepeatModeler 来识别剩余的 TE（1）或不使用（0，默认）。此步骤非常慢，可能有助于恢复一些 TE。

--anno [0|1]执行（1）或不执行（0，默认）TE 库构建后的整个基因组 TE 注释。

--rmout [文件]提供自己基于同源性的 TE 注释，而不是使用 EDTA 库进行屏蔽。

文件以 RepeatMasker .out 格式提供。此文件将与基于结构的 TE 注释合并。（--anno 1 必需）。默认值：使用 EDTA 库进行注释。

--evaluate [0|1]评估（1）TE 注释的分类一致性。（--anno 1 必需）。默认值：0。此步骤较慢，不影响注释结果。

--exclude[文件]在 MAKER.masked 输出中排除 TE 屏蔽的区域（bed 格式）。默认值：undef。（--anno 1 必需）。

--u [浮点数]中性突变率，用于计算完整 LTR 的年龄。完整 LTR 年龄在此文件中找到：*EDTA_raw/LTR/*.pass.list。默认值：1.3e-8（每个碱基每年，以水稻为例）。

--threads|-t[整数]运行此脚本的线程数（默认值：4）

例子

```
perl EDTA.pl --genome genome.fa  --overwrite 1 --sensitive 1 --anno 1  --threads 10
```

1)输入

必需：基因组文件 [FASTA]。请确保序列名称短（<=13 个字符）且简单（即，字母、数字和下划线）。

可选参数：

物种或近缘物种的编码序列 [FASTA]。此文件有助于清除 TE 库中的基因序列。

此版本基因组装配的已知基因位置 [BED]。该文件中指定的坐标将被排除在 TE 注释之外，以避免过度屏蔽(可以考虑提供同源注释)。

该物种的筛选的 TE 库 [FASTA]。该文件被完全信任。请确保它是经过筛选的。即使只有几个筛选过的序列，也是可以的。它不需要完整。提供筛选的 TE 序列，尤其是那些被低估的 TE 类型（如 SINEs 和 LINEs），将极大地提高注释质量。

2)输出

一个非冗余的 TE 库：$genome.mod.EDTA.TElib.fa。如果提供了筛选的库，它将被包含在此文件中。如果指定了 --force 1，则水稻库将在部分情况下被包含在内。TEs 被分类为超家族级别，并使用 Wicker 等人（2007）报告的三字母命名系统。每个序列都可以看作是一个 TE 家族。

新的 TE 家族：$genome.mod.EDTA.TElib.novel.fa。此文件包含在筛选库中未包含的 TE 序列（--curatedlib 必需给定）。

全基因组 TE 注释：$genome.mod.EDTA.TEanno.gff3。此文件包含结构完整和碎片化的 TE 注释（--anno 1 必需）。

全基因组 TE 注释的摘要：$genome.mod.EDTA.TEanno.sum（--anno 1 必需）。

低阈值 TE 屏蔽：$genome.mod.MAKER.masked。这是一个只有长的 TE（>=1 kb）被屏蔽的基因组文件。您可以将其用于去 denovo 基因注释。在实践中，这种方法将减少基因区域的过度屏蔽，从而提高基因预测的质量。但是，初始基因模型应包含 TE 并需要进一步过滤（--anno 1 必需）。

简单 TE 的注释不一致性：$genome.mod.EDTA.TE.fa.stat.redun.sum（--evaluate 1 必需）。

嵌套 TE 的注释不一致性：$genome.mod.EDTA.TE.fa.stat.nested.sum（--evaluate 1 必

需）。

整体注释不一致性：$genome.mod.EDTA.TE.fa.stat.all.sum（--evaluate 1 必需）。

识别特定 TE 类型的完整元素

1）从基因组中获取原始 TE（在不同运行中指定 -type ltr|tir|helitron）

perl EDTA_raw.pl [选项]

--genome[文件]基因组 FASTA

--species [Rice|Maize|others]指定物种以识别 TIR 候选序列。默认值：others

--type[ltr|tir|helitron|all]指定要获取的原始 TE 候选类型。默认值：all

--overwrite[0|1]如果找到以前的结果，决定是否覆盖（1，重新运行）或不覆盖（0，默认）。

--threads|-t[整数]运行此脚本的线程数

--help|-h 显示此帮助信息

2）完成其余的 EDTA 分析（指定 -overwrite 0，它将自动获取工作文件夹中的现有结果）

perl EDTA.pl --overwrite 0 [选项]

panEDTA 用法

将按顺序对每个基因组进行注释，然后与 panEDTA 功能结合使用。已识别和重用现有基因组的 EDTA 注释（使用--anno 1 运行的 EDTA）。加快全基因组注释的一种方法是分别并行执行每个基因组的 EDTA 注释，然后执行 panEDTA 以完成其余的运行。您可能希望保存每个基因组的 GFF 文件和 EDTA 结果的总和文件，因为它们将被 panEDTA 覆盖。为了帮助过滤与基因相关的序列，至少需要一个 CDS 文件。您可以查看./test 文件夹中的示例来熟悉。

sh panEDTA.sh -g genome_list.txt -c cds.fasta -t 10

-g 具有路径可从工作目录访问的基因组文件列表。必需：您可以仅在此文件中提供基因组列表（一列，每行一个基因组）。

可选：

您还可以在此文件中提供基因组和 CDS 文件（两列，一列基因组和一列 CDS）。允许缺少 CDS 文件（例如，对于某些或所有基因组）。

-c 必需。以 fasta 格式的编码序列文件。通过此参数提供的 CDS 文件将填充基因组列表中缺少的 CDS 文件。如果基因组列表中没有提供 CDS 文件，则此 CDS 文件将用于所有基因组。

-l 可选。遵循 RepeatMasker 命名格式的手动筛选的非冗余库。

-f 作为 pangenome 候选 TE 保留的个体基因组中完整长度 TE 副本的最小数目.较低的数值包容性更强，并将增加库大小，增加敏感性和增加不一致性。较高的数值则较严格，并将减小库大小，降低敏感性和降低不一致性。默认值：3。

-t 运行 panEDTA 的 CPU 数。默认值：10。

3.2 GeMoMa 预测

- 没有 RNA-seq 数据：

./run.sh <search> <target-genome> <ref-anno> <ref-genome> <out-dir>

- 使用 RNA-seq 数据：

./run.sh <search> <target-genome> <ref-anno> <ref-genome><out-dir> <lib-type> <mapped-reads>

一个应用脚本，允许使用 GeMoMaPipeline 模块以最少的参数输入启动完整的 GeMoMa 工作流程：pipeline.sh。如果您希望使用标准参数且不使用任何加速技巧运行工作流程，只需使用此脚本即可。

- 没有 RNA-seq 数据：

java -jar GeMoMa-1.9.jar CLI GeMoMaPipeline threads=<threads> outdir=<outdir> GeMoMa.Score=ReAlign AnnotationFinalizer.r=NO o=true t=<target_genome> i=<reference_1_id> a=<reference_1_annotation> g=<reference_1_genome>

- 使用 RNA-seq 数据：

./pipeline.sh <search> <target-genome> <ref-anno><ref-genome> <threads> <out-dir><lib-type> <mapped-reads>

其中，

1. search 是用于使用的搜索算法的开关，可以是 tblastn 或 mmseqs。
2. target-genome 是目标物种的基因组（FastA）。
3. ref-anno 是参考物种的注释（GFF/GTF）。
4. ref-genome 是参考物种的基因组（FastA）。
5. threads 是要使用的线程数。
6. out-dir 是输出目录。
7. lib-type 是 RNA-seq 库类型（FR_UNSTRANDED、FR_FIRST_STRAND、FR_SECOND_STRAND）。
8. mapped-reads 是映射的 RNA-seq reads（SAM/BAM）。

四、Hi-C 互作分析

4.1、Hi-C 质控

HiC-Pro 是一个用于处理 Hi-C 数据的流程，它从测序数据（FASTQ 文件）开始，执行多个步骤，最终的输出文件是原始的和标准化的交互矩阵。

HiC-Pro 采用两步对齐策略，使用 Bowtie2。首先，将与基因组上对齐的两条 mate 的 reads 保存在 BAM 文件中。接下来，根据配置文件中指定的连接位点修剪未对齐的 reads，然后在修剪后的 reads 上再次运行 Bowtie2，并将对齐保存在第二个 BAM 文件中。最后，将 BAM 文件合并并存储在 bwt2 文件夹中。

由于两个 mate 是独立对齐的，统计数据分别报告每个 mate 的情况。大部分 reads 在基因组上对齐（平均超过 90%），平均有约 10%的 reads 在修剪后对齐。这意味着有 10%的 reads 是 chimeric，即跨越连接位点。chimeric reads 的比例取决于样本准备过程中选择的片段大小和读长。高比例的未对齐 reads（超过 20%）可能表示文库存在问题，或者输入给 HiC-Pro 的连接位点不正确。

基于对齐和配对对 read pairs 进行过滤。接下来，将 read pairs 映射到限制性片段上，并根据它们在限制性片段上的位置和方向将其分类为有效对和无效对。然后，去除 PCR 重复并将去重后的有效对保存在.allValidPairs 文件中。一般要求有效数据至少 30%，具体取决于 HiC-Pro 过滤参数和文库质量。

1、软件安装

软件下载地址：https://github.com/nservant/HiC-Pro/archive/refs/tags/v3.1.0.tar.gz

解压缩后:

conda env create -f environment.yml -p HICPro

conda activate HICPro

make configure

修改里面的路径，不要采用/usr/bin 那个路径：config-system.txt

make

2、输入和配置文件

要运行 HiC-Pro，需要四个注释文件：Bowtie2 索引，染色体长度文件，消化后的限制性酶切片段的坐标文件，以及（4）配置文件。

1）创建用于每个样本的文件夹，并将原始测序文件移动到相应的子目录中。

mkdir -p rawdata/rep1/

文件夹中放置一对 FQ,格式如下：

rep1_R1.fq.gz

rep1_R2.fq.gz

2）Bowtie2 索引

```
bowtie2-build Arabidopsis_thaliana.TAIR10.genome.fa  Arabidopsis_thaliana.TAIR10.genome.fa
```

3）长度文件

```
samtools faidx Arabidopsis_thaliana.TAIR10.genome.fa
```

```
awk  '{print $1 "\t" $2}'  Arabidopsis_thaliana.TAIR10.genome.fa.fai > AT.chrom.sizes
```

4）生成限制性片段坐标文件。使用`digest_genome.py`脚本生成限制性片段文件，例如DpnII限制性酶的限制性片段文件。

```
digest_genome.py  -r DpnII -o AT.digest  Arabidopsis_thaliana.TAIR10.genome.fa
```

5) 更新HiC-Pro的配置文件`config-hic-pro.txt`。

根据您的实验设置，修改配置文件中的以下字段:

PAIR1_EXT = _1

PAIR2_EXT = _2

BOWTIE2_IDX_PATH = /mnt/RAID-5/MD0/bioinformatics/SXB/10.Genome/03.Hi-C/00.HiCPro

REFERENCE_GENOME = Arabidopsis_thaliana.TAIR10.genome.fa

GENOME_FRAGMENT = /mnt/RAID-5/MD0/bioinformatics/SXB/10.Genome/03.Hi-C/00.HiCPro/AT.digest

LIGATION_SITE = GATCGATC

在`DATA`部分，指定每个mate的FASTQ文件命名方式（在我们的例子中，使用`_1`和`_2`作为后缀）。

在`ALIGNMENT`部分，提供Bowtie2索引的路径。

在`ANNOTATION file`部分，指定参考基因组的名称（必须与Bowtie2索引文件的前缀相同）。

在`DIGESTION`部分，指定限制性片段文件以及连接位点（在我们的例子中是GATCGATC）。

3、执行

```
/mnt/RAID-5/MD0/bioinformatics/soft/HiC-Pro/bin/HiC-Pro -c config-hicpro.txt -i rawdata/ -o Results
```

4、结果介绍

1）映射结果

bowtie_results文件夹包含reads映射的结果。第一步映射的结果在bwt2_glob文件夹中，第二步映射的结果在bwt2_loc文件夹中。最终的BAM文件、reads配对和映射统计结果都可以在bwt2文件夹中找到。

读取映射统计结果以条形图形式表示，显示R1和R2 reads的比对比例。第一根条形代表总体比对reads的比例，第二根条形区分了两个映射步骤。

映射统计结果存储在'.mapstat'文件中。每个样品的总体映射统计结果可以在'SAMPLE_NAME.mmapstat'文件中找到。

通常情况下，预期有很高比例的reads能够对基因组进行比对（80-90%）。其中，我们通常观察到约10%的第二步比对的reads。这些reads是具有连接点的嵌合片段，我们能够检测到其连接位点。异常高水平的嵌合reads可能反映出文库制备过程中的连接问题。

一旦R1和R2 reads在基因组上进行了比对，HiC-Pro会重建配对信息。配对统

计结果汇总在下面的图表中，并可在'.pairstat'文件中找到。每个样品的配对统计结果汇总在'SAMPLE_NAME.mpairstat'文件中。

单端比对或多重比对的比例取决于基因组复杂性和未比对 reads 的比例。单端比例通常接近未比对的 R1 和 R2 reads 之和，因为同一对 mate 都未能比对的情况不太可能发生。

2）有效相互作用产物列表

hic_results 文件夹包含所有经过 Hi-C 处理的数据，并进一步分成几个子文件夹。hic_results/data 文件夹用来存储有效的相互作用结果（'.validPairs'），以及其他统计文件。文本格式：

read 名称 / chr_reads1 / pos_reads1 / strand_reads1 / chr_reads2 / pos_reads2 / strand_reads2 / fragment_size [/allele_specific_tag]

每对 reads 生成一个 validPairs 文件。然后，这些文件被合并为 allValidPairs 文件，并根据配置文件中的设定是否去除重复项。

有关读取配对筛选的统计数据可以在'.RSstat'文件中找到，并在'SAMPLE_NAME.mRSstat'文件中汇总。

通过筛选有效和无效的配对，可以评估连接效率。由于连接是一个随机过程，预期每个有效连接类别的比例为 25%。同样，高的悬空末端或自环配对与低质量的实验有关，并且显示出消化、填充或连接步骤中的问题。

在没有限制酶的 Hi-C 协议中，该分析步骤将被跳过。因此，对齐的配对直接用于生成接触图。建议对短程接触（通常<1kb）进行过滤，因为这些配对很可能是自连接产物。

在进行等位基因特异性分析时，还可以从'assplit.stat'文件中获取附加的统计信息，其中包括分配给亲本基因组的读对比例。需要注意的是，等位基因特异性分配也可以从'.RSstat'文件中获得。'assplit.stat'文件中的统计数据通常与'.RSstat'文件中的数据不同，因为它们是在去除重复项之后对有效对进行计算的，而'.RSstat'文件中的统计数据是在去除重复项之前的每个读块中计算的。

3）染色体内和染色体间接触图

可以从有效相互作用产物列表中提取其他质量控制指标，例如片段大小分布。我们通常希望看到以 300 个碱基为中心的分布，这对应于常用的双端插入大小。还会呈现重复比例。高水平的重复表示分子复杂性较低，可能存在 PCR 偏差问题。最后，一个重要的指标是观察染色体内和染色体间的相互作用比例，以及长程（>20kb）与短程（<20kb）的染色体内相互作用比例。然后，接触图可以在 hic_results/matrix 文件夹中找到。原始交互矩阵位于 raw 文件夹中，归一化后的交互矩阵位于 iced 文件夹中。交互矩阵根据指定的分辨率生成（请参考配置文件）。

一个交互矩阵由以下内容组成：

a 与指定分辨率相关的基因组区间列表（BED 格式）。

b 作为标准三元稀疏格式（即列表格式）存储的矩阵。

4.2、AB 鉴定

计算 Hi-C 矩阵的 PCA 特征向量。

用法: hicPCA --matrix MATRIX --outputFileName OUTPUTFILENAME

[OUTPUTFILENAME ...]

[--whichEigenvectors WHICHEIGENVECTORS [WHICHEIGENVECTORS ...]]

[--format {bedgraph,bigwig}]

[--chromosomes CHROMOSOMES [CHROMOSOMES ...]]

[--method {dist_norm,lieberman}] [--ligation_factor]

[--extraTrack EXTRATRACK] [--histonMarkType HISTONMARKTYPE]

[--pearsonMatrix PEARSONMATRIX] [--obsexpMatrix OBSEXPMATRIX]

[--ignoreMaskedBins] [--help] [--version]

必需参数

--matrix, -m 以 h5 格式存储的 HiCExplorer 矩阵。

--outputFileName, -o 用于存储 PCA 计算结果的文件名。输出文件数量必须和计算的特征向量数量相同。

可选参数

--whichEigenvectors, -we 需要计算的特征向量列表，例如“1 2 5”表示返回第一、第二和第五个特征向量。（默认值：“1 2”）

--format, -f 可选值: bedgraph, bigwig 输出格式。可以是 bedgraph 或 bigwig（默认值：“bigwig”）。

--chromosomes 要包含在相关性计算中的染色体列表。

--method 可选值: dist_norm, lieberman 用于构建观察-期望矩阵的方法。可以选择 dist_norm 或 lieberman（默认值：“dist_norm”）。

--ligation_factor 设置该标志会对预期矩阵的每个条目乘以一个缩放因子，以处理与邻近连接有关的情况，这一点在 Homer 软件中已经解释过。此标志仅对 dist_norm 方法有效，如果选择了 lieberman 方法，则会被忽略。

--extraTrack 需要基因轨迹或组蛋白修饰覆盖文件（最好是广泛的标记），以确定特征向量的值是否需要翻转符号。请注意：bed 文件将被解释为基因轨迹，bigwig 文件将被解释为组蛋白修饰。

--histonMarkType 将其设置为 active 或 inactive。只有在给定附加轨迹时，即组蛋白修饰覆盖文件（默认值：“active”）。

--pearsonMatrix, -pm 内部将输入矩阵按染色体转换为观察-期望矩阵，然后连续转换为 Pearson 矩阵。设置此参数以将 Pearson 矩阵写入文件。

--obsexpMatrix, -oem 内部将输入矩阵按染色体转换为观察-期望矩阵，然后连续转换为 Pearson 矩阵。设置此参数以将观察/期望矩阵写入文件。

--ignoreMaskedBins 掩盖的 bin 通常被设为 0。此选项在计算 PCA 之前移除掩盖的 bin。注意：这将导致空的 PCA 区域。默认值：False

转格式:

hicConvertFormat -m iced/500000/rep1_500000_iced.matrix --bedFileHicpro raw/500000/rep1_500000_abs.bed --inputFormat hicpro --outputFormat h5 -o 50K.h5

hicPCA -m 50K.h5 -o pca1.bedgraph --whichEigenvectors 1 --format bedgraph

hicPCA -m 50K.h5 -o pca1.bw --whichEigenvectors 1 --format bigwig

hicPlotMatrix -m 50K.h5 --bigwig pca1.bw -o pca1.pdf --perChromosome

4.3、TAD 鉴定

使用称为 TAD-separation score 的度量方式，确定在每个 Hi-C 矩阵 bin 上左、右区域之间的分离程度。为了找到 TADs，程序需要首先计算不同窗口大小下的 TAD scores。然后，使用该计算的结果鉴定 TADs。

示例用法如下：

$ hicFindTads -m hic_matrix.h5 – outPrefix TADs – correctForMultipleTesting fdr

此工具生成的 bedgraph 文件可用于绘制基因组或特定区域的所谓的隔离得分（insulation score）。因为 TAD 鉴定可能受测序深度不同以及鉴定 TAD 参数不同而导致不同样本的 TAD 数量或 TADs 的宽度差别很大，结果的可比较性差，而该得分在不同样本之间比较因为更加稳定，而导致结果更加可靠。

使用方法：

hicFindTADs --matrix MATRIX --outPrefix OUTPREFIX

--correctForMultipleTesting {fdr,bonferroni,None}

[--minDepth INT bp] [--maxDepth INT bp] [--step INT bp]

[--TAD_sep_score_prefix TAD_SEP_SCORE_PREFIX]

[--thresholdComparisons THRESHOLDCOMPARISONS]

[--delta DELTA] [--minBoundaryDistance MINBOUNDARYDISTANCE]

[--chromosomes CHROMOSOMES [CHROMOSOMES ...]]

[--numberOfProcessors NUMBEROFPROCESSORS] [--help]

[--version]

必需参数

--matrix, -m 用于计算的校正过的 Hi-C 矩阵。

--outPrefix 保存结果文件的文件前缀：

<prefix>_tad_separation.bm，输出文件的格式为 chrom start end TAD-sep1 TAD-sep2 TAD-sep3 ...等。我们称此格式为 bedgraph 矩阵，并可使用 hicPlotTADs 进行绘制。

<prefix>_zscore_matrix.h5，用于计算 TAD-separation score 的 z-score 矩阵。

<prefix>_boundaries.bed，包含边界的位置。该文件中的基因组坐标对应于所使用的分辨率。因此，对于 10,000bp 的 Hi-C bin，边界位置为 10,000bp 长。对于限制性片段矩阵，边界位置取决于边界处的片段长度。

<prefix>_domains.bed 包含 TADs 的位置。这是一组不重叠的基因组位置。

<prefix>_boundaries.gff 类似于边界 bed 文件，但包含额外的信息（p-value、delta）。

<prefix>_score.bedgraph 文件包含每个 Hi-C bin 坐标处测得的 TAD-separation

score。在基因组浏览器中可视化非常有用。delta 和 p-value 设置将作为名称的一部分保存。

--correctForMultipleTesting 可能的选择：fdr、bonferroni、None，选择 bonferroni 或 FDR（false discovery rate）进行多重比较。Bonferroni 控制家族内错误率（FWER），需要 p 值。FDR 控制 I 型错误的可能性，需要 q 值。第三个选项是不使用任何多重比较方法（默认值：“fdr”）。默认值：“fdr”

可选参数

--minDepth 左侧和右侧的每个 Hi-C bin 上要考虑的最小窗口长度（以 bp 为单位）。此数字应至少为 Hi-C 矩阵的 bin 大小的 3 倍大。

--maxDepth 左侧和右侧的切点处要考虑的最大窗口长度（以 bp 为单位）。该数字应为 Hi-C 矩阵的 bin 大小的 6-10 倍左右。

--step

从 -minDepth 到 -maxDepth 移动时的步长。注意，步长随着 minDeph + (step * int(x)**1.5) for x in [0, 1, …] 以指数方式增长，直到达到 maxDepth。例如，选择 step=10,000、minDepth=20,000 和 maxDepth=150,000 将计算窗口大小为 20,000、30,000、40,000、70,000 和 100,000 的 TAD scores。

--TAD_sep_score_prefix

有时，更改某些参数而无需重新计算 z-score 矩阵和 TAD-separation score 是很有用的。对于这种情况，可以提供包含 TAD separation score 和 z-score 矩阵的前缀。如果给定此选项，将计算新的边界，但不使用 – minDepth、 – maxDepth 和 – step 的值。

--thresholdComparisons

Bonferroni 校正的 p 值阈值/ FDR 的 q 值。通过比较（Wilcoxon 秩和）局部最小值处左侧和右侧区域（菱形）之间的 z-scores 的分布与左侧为 – minDepth 和右侧为 – minDepth 的钻石的矩阵 z-scores，来估计局部最小值为边界的概率。如果 correctForMultipleTesting 为“None”，则将阈值应用于未经任何多重检验校正的原始 p 值。如果不使用阈值，则将其设置为“1”（默认值：0.01）。

--delta

候选边界与周围 bin 的 TAD-separation score 均值之间的差异的最小阈值。delta 值减少了浅层的虚假边界，这些边界通常出现在大 TAD 的中心，当 TAD-sep. score 为平坦时。较高的 delta 阈值产生更保守的边界估计（默认值：0.01）。

--minBoundaryDistance 边界之间的最小距离（以 bp 为单位）。此参数可用于减少噪声引起的虚假边界。

--chromosomes 应绘制的染色体及其顺序。此选项覆盖 – region 和 – region2。

--numberOfProcessors, -p 要使用的处理器数量（默认值：1）。

下面是使用 hicFindTADs 的典型命令行示例：

```
hicFindTADs -m 20K.h5 --outPrefix 20K_thres0.05_delta0.01_fdr --thresholdComparisons 0.05 --delta 0.01 --correctForMultipleTesting fdr -p 1
hicFindTADs -m 20K.h5 --outPrefix 20K_thres0.001_delta0.01_fdr --
```

thresholdComparisons 0.001 --delta 0.01 --correctForMultipleTesting fdr

hicFindTADs -m 20K.h5 --outPrefix 20K_thres0.005_delta0.01_fdr --thresholdComparisons 0.005 --delta 0.01 --correctForMultipleTesting fdr

hicFindTADs -m 20K.h5 --outPrefix 20K_thres0.01_delta0.01_fdr --thresholdComparisons 0.01 --delta 0.01 --correctForMultipleTesting fdr

TAD 边界位置存储在 boundaries 文件中，domains.bed 文件包含 TAD 的位置，score 文件包含 TAD 分离得分或所谓的 TAD 隔离得分，以不同的格式呈现。需要注意的是，tad_score.bm 文件是一个 bedgraph 矩阵，可以用于在 hicPlotTADs 中显示 TAD 分离得分曲线。

要比较多个 TAD 调用结果，可以使用 hicPlotTADs 和以下命令，例如使用以下 tracks.ini 文件：tracks.ini:

hicPlotTADs --tracks tracks.ini --region 1:680000-8500000 -o TAD_calling_comparison.png

注：在_domains.bed 输出文件中，第 5 列包含位于每个 domain 开头的边界的 TAD 分离得分。识别边界的过程如下：调用平均 TAD 分数中的所有局部极小值。每个局部极小值应至少相隔 min_boundary_distance。如果未提供此值，则设置为平均 bin 大小的 4 倍。对于检测到的每个局部极小值，计算其 p 值，然后计算一个 q 值。对于检测到的每个局部极小值，计算“delta”，即最小点之前的 10 个 bin 的平均 TAD 分数与之后的 10 个 bin 的平均 TAD 分数之间的差异（不包括最小点）。仅保留符合以下条件的极小值：p 值（或用户选择的 q 值）应低于给定的阈值，并且 delta 应高于用户定义的阈值。两个连续边界之间的所有内容都是一个 TAD。为了计算 p 值，将局部极小值上方的“菱形”的 z-score 分布与离该局部极小值 min_depth 处的 z-score 分布使用 Wilcoxon 秩和检验进行比较。同样，计算离局部极小值上游的 z-score 的分布。将两个 p 值中最小的值分配给局部极小值。如果未提供 min_depth，则按照以下方式计算：如果 bin 大小小于 1000，则计算为 bin 大小 30；如果 bin 大小介于 1000 和 20,000 之间，则计算为 bin 大小 10；如果 bin 大小大于 20,000，则计算为 bin 大小*5。如果未提供 max_depth，则按照以下方式计算：如果 bin 大小小于 1000，则计算为 bin 大小 60；如果 bin 大小介于 1000 和 20,000 之间，则计算为 bin 大小 40；如果 bin 大小大于 20,000，则计算为 bin 大小*10。

4.4、Loop 鉴定

hicHyperoptDetectLoopsHiCCUPS 是一个用于优化计算的工具，可以根据给定的矩阵计算环，并使用蛋白质文件验证检测到的环。

用法：hicHyperoptDetectLoopsHiCCUPS --matrix MATRIX --proteinFile PROTEINFILE

--maximumNumberOfLoops MAXIMUMNUMBEROFLOOPS

--juicerPath JUICERPATH

[--outputFileName OUTPUTFILENAME]

[--resolution RESOLUTION]

[--chrPrefixLoops {None,add,remove}]

[--runs RUNS] [--threads THREADS]

--normalization NORMALIZATION [--cpu]

[--restricted] [--help] [--version]

必需参数:

--matrix, -m 要计算环的矩阵。

--proteinFile, -p 用于验证检测到的环的蛋白质文件。

--maximumNumberOfLoops, -ml 应用于优化计算的最大环数。

--juicerPath, -juicer.jar 文件的路径。

--outputFileName, -o 用于存储优化结果的输出文件名（默认值：“hyperoptHiCCUPS_result.txt”）。

可选参数:

--resolution, -r 矩阵的分辨率（默认值：10000）。

--chrPrefixLoops, -cl 可能的选择：None、add、remove 对循环的染色体名称添加/删除/不进行“chr”前缀。

--runs hyperopt 运行次数。默认值：100

--threads, -t 线程数（使用 python multiprocessing 模块）（默认值：4）。

--normalization, -k 归一化表名称。

--cpu 使用 CPU 版本。

--restricted 如果使用 GPU 版本，则仅在 8 MB 内搜索。

hicHyperoptDetectLoops 是一个用于优化计算的工具，可以根据给定的矩阵计算环，并使用蛋白质文件验证检测到的环。

用法：hicHyperoptDetectLoops --matrix MATRIX --proteinFile PROTEINFILE

--maximumNumberOfLoops MAXIMUMNUMBEROFLOOPS

[--outputFileName OUTPUTFILENAME]

[--resolution RESOLUTION]

[--chrPrefixLoops {None,add,remove}]

[--threads THREADS] [--runs RUNS] [--help]

[--version]

必需参数:

--matrix, -m 要计算环的矩阵。

--proteinFile, -p 用于验证检测到的环的蛋白质文件。

--maximumNumberOfLoops, -ml 应用于优化计算的最大环数。

--outputFileName, -o 用于存储优化结果的输出文件名（默认值：“hyperopt_result.txt”）。

可选参数:

--resolution, -re 矩阵的分辨率（默认值：10000）。

--chrPrefixLoops, -cl 可能的选择：None、add、remove 对循环的染色体名称添加/删

除/不进行“chr”前缀。

--threads, -t 线程数（使用 python multiprocessing 模块）（默认值：4）。

--runs, -rhyperopt 运行次数（默认值：100）。