[Synthetic Genomeic Restructure by Single Tube Long Fragement Sequence Software]

[v1.0]

设计说明

（如文档为软件设计说明，请把“使用手册”更改为“设计说明”）

注：使用说明须含有详细的登录界面、操作步骤截图及文字描述。设计说明须含有详细的软件结构图、各个功能的流程图、逻辑框图，介绍软件总体、接口设计，模块名称功能，函数名称功能，算法，数据结构、数据字典、运行设计等内容。文档一般建议15页以上。

目录

[一、引言 3](#_Toc108023998)

[1.1 编写目的 3](#_Toc108023999)

[1.2 项目背景 3](#_Toc108024000)

[1.3 定义 3](#_Toc108024001)

[1.4 参考资料 3](#_Toc108024002)

[二、软件概述 4](#_Toc108024003)

[2.1 软件简介 4](#_Toc108024004)

[2.2 软件结构 4](#_Toc108024005)

[2.3 运行环境 5](#_Toc108024006)

[三、操作说明 6](#_Toc108024007)

[3.1 软件参数 6](#_Toc108024008)

[3.2 输入文件 7](#_Toc108024009)

[3.3 运行软件举例 7](#_Toc108024010)

[四、模块说明 8](#_Toc108024011)

[4.1 barcode拆分模块 8](#_Toc108024012)

[4.1.1 描述 8](#_Toc108024013)

[4.1.2 输入文件 8](#_Toc108024014)

[4.1.3 输出文件 8](#_Toc108024015)

[4.1.4 用法举例 8](#_Toc108024016)

[4.2 过滤模块 8](#_Toc108024017)

[4.2.1 描述 8](#_Toc108024018)

[4.2.2 输入文件 8](#_Toc108024019)

[4.2.3 输出文件 9](#_Toc108024020)

[4.2.4 参数说明 9](#_Toc108024021)

[4.3 比对模块 9](#_Toc108024022)

[4.3.1 描述 9](#_Toc108024023)

[4.3.2 输入文件 9](#_Toc108024024)

[4.3.3 输出文件 9](#_Toc108024025)

[4.3.5 命令详情 9](#_Toc108024026)

[4.4 统计模块 10](#_Toc108024027)

[4.4.1 描述 10](#_Toc108024028)

[4.4.2 输入文件 10](#_Toc108024029)

[4.4.3 输出文件 10](#_Toc108024030)

[4.4.4 过程说明 10](#_Toc108024031)

[4.5 边点提取模块 11](#_Toc108024032)

[4.5.1 描述 11](#_Toc108024033)

[4.5.2 输入文件 11](#_Toc108024034)

[4.5.3 输出文件 11](#_Toc108024035)

[4.5.4 使用说明 12](#_Toc108024036)

[4.6 barcode信息整合与质控模块 12](#_Toc108024037)

[4.6.1 描述 12](#_Toc108024038)

[4.6.2 输入文件 12](#_Toc108024039)

[4.6.3 输出文件 12](#_Toc108024040)

[4.6.4 过程说明 13](#_Toc108024041)

[4.6.3 参数说明 14](#_Toc108024042)

[4.7 精准scaffold组装模块和全局筛选 14](#_Toc108024043)

[4.7.1 描述 14](#_Toc108024044)

[4.7.2 输入文件 14](#_Toc108024045)

[4.7.3 输出文件 14](#_Toc108024046)

[4.7.2 参数说明 15](#_Toc108024047)

[4.8 序列重建模块 16](#_Toc108024048)

[4.8.1 描述 16](#_Toc108024049)

[4.8.2 输入文件 16](#_Toc108024050)

[4.8.3 输出文件 16](#_Toc108024051)

[4.8.4 使用说明 16](#_Toc108024052)

[五、运行设计 17](#_Toc108024053)

[5.1 集成 17](#_Toc108024054)

[5.2 分块 17](#_Toc108024055)

[六、常见问题与解决方法 17](#_Toc108024056)

[6.1 常见问题 17](#_Toc108024057)

# 一、引言

## 1.1 编写目的

基因组是生命系统的指令中枢，对基因组的研究是生命科学的的核心内容。人工合成酵母基因组计划（Sc2.0）在设计时，在非必须基因下游的插入正反向回文序列的loxp位点，此处称为loxpsym。以loxpsym位点介导的合成染色体的重排与修饰称为SCRaMbLE系统（Synthetic Chromosome Rearrangement and Modification by LoxP-mediated Evolution) ，在Cre酶的诱导下基因组可以发生大规模的重排，产生具有插入、缺失、重复和倒位等打片段变异的“新”基因组。已有研究表面，SCRaMbLE系统在筛选酵母高产、抗性菌株上具有快速而准确的优势，而需要阐明菌株背后的分子机制，需要对其基因组进行研究。常规的基因组组装方法具有耗时长、花费高，组装效果差等缺点，而已有的基于二代测序开发的合成酵母基因组重构方法（Syngenor）对于近60%的复杂变异SCRaMbLE基因组的组装多样性高，无法获得准确的基因组组装结果。于是基于华大基因新报道的stLFR（single tube long fragement sequence）测序技术，开发了Sgrstlfr。

## 1.2 项目背景

基于合成酵母SCRaMbLE基因组研究中，基因组组装困难，成本高，不利于大规模筛选菌株，于是开发了本软件。

## 1.3 定义

1. Sc2.0：人工合成酵母基因组2.0计划

2. Loxpsym回文序列的loxp重组元件（“ATAACTTCGTATAATGTACATTATACGAAGTTAT”，34bp），在重组酶的作用下，可以产生基因组重排。

3. SCRaMbLE：由loxp位点介导的合成染色体的重排与修饰(Synthetic Chromosome Rearrangement and Modification by LoxP-mediated Evolution)，简称为SCRaMbLE。

4. stLFR：单管长读长测序技术（single tube long fragement seqence），简称为stLFR。

5. pair end nod：通过双端测序的信息获取的基因组断点。

6. split read nod：通过切分无法比对到基因组的含有loxpysm序列的reads，进行重比对获取的断点。

7. 边（edge）：通过获取的新断点将基因组切分为不同的区段，每个区段简称为“边”。

8. 点（nod）：通过切分无法比对到基因组的含有loxpsym序列的reads，进行重比对获取的断点。

## 1.4 参考资料

[1] Dymond, J. & Boeke, J. The Saccharomyces cerevisiae SCRaMbLE system and genome minimization. Bioeng. Bugs 3, 168–171 (2012).

[2] Shen, Y., Stracquadanio, G., Wang, Y., Yang, K., Mitchell, L.A., Xue, Y., Cai, Y., Chen, T., Dymond, J.S., Kang, K., et al. (2016). SCRaMbLE generates designed combinatorial stochastic diversity in synthetic chromosomes. Genome Res 26, 36-49.

[3] Wang, O., Chin, R., Cheng, X., Wu, M.K.Y., Mao, Q., Tang, J., Sun, Y., Anderson, E., Lam, H.K., Chen, D., et al. (2019). Efficient and unique cobarcoding of second-generation sequencing reads from long DNA molecules enabling cost-effective and accurate sequencing, haplotyping, and de novo assembly. Genome Res 29, 798-808.

# 二、软件概述

## 2.1 软件简介

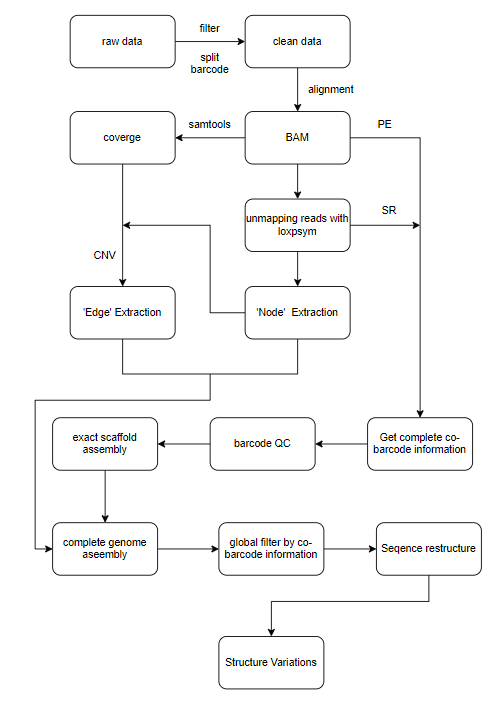
本软件全称：基于stLFR的合成基因组重构软件（Synthetic Genomeic Restructure by Single Tube Long Fragement Sequence Software），简称Sgrstlfr。本软件主要用于以loxpsym位点构成的SCRaMbLE基因组的组装，通过该软件，可以获得少部分可能的基因组组装结果，相比于原有方法(Syngenor)，其基因组组装结果大大减少，获取准确的基因组组装结果，服务于下游的研究。

## 2.2 软件结构

本软件主要由以下八个模块组成，分别对应step1-8，并按顺序运行：

模块1：barcode拆分模块

模块2：过滤模块  
  
模块3：比对模块  
  
模块4：统计模块  
  
模块5：边点提取模块  
  
模块6：barcode信息整合与质控模块  
  
模块7：精准scaffold组装模块和全局筛选模块  
  
模块8：序列重建模块



## 2.3 运行环境

硬件环境

1. 32/64位中央处理器。

2. 内存：4G以上。

3. 硬盘空间：无具体需求。

软件环境

1. Linux操作系统( 内核版本号不低于2.6)。

2. perl(版本不低于5.1)。

3. samtools(版本不低于1.11)。

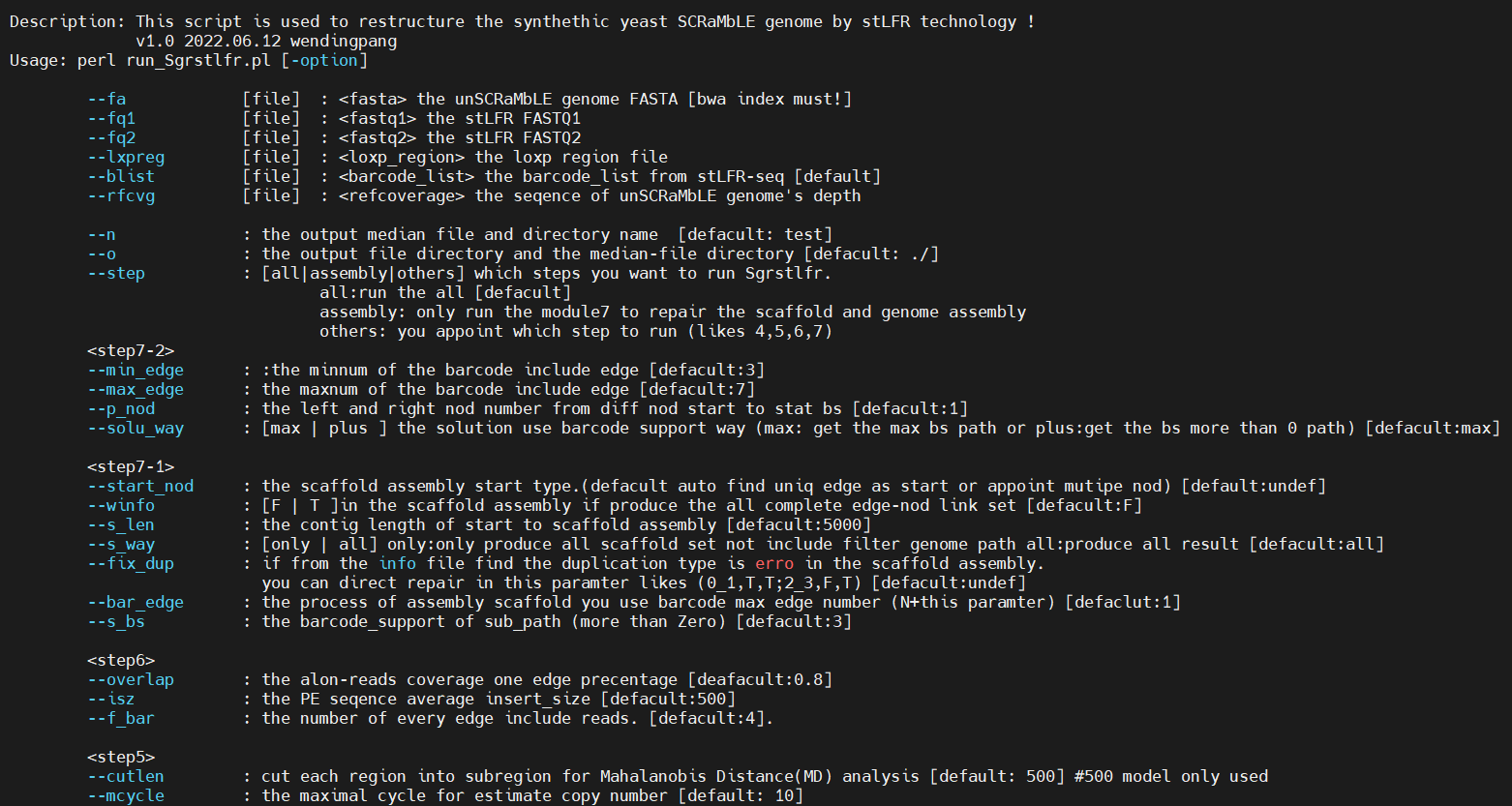
4. soapnuke（版本不低于2.1.0）。

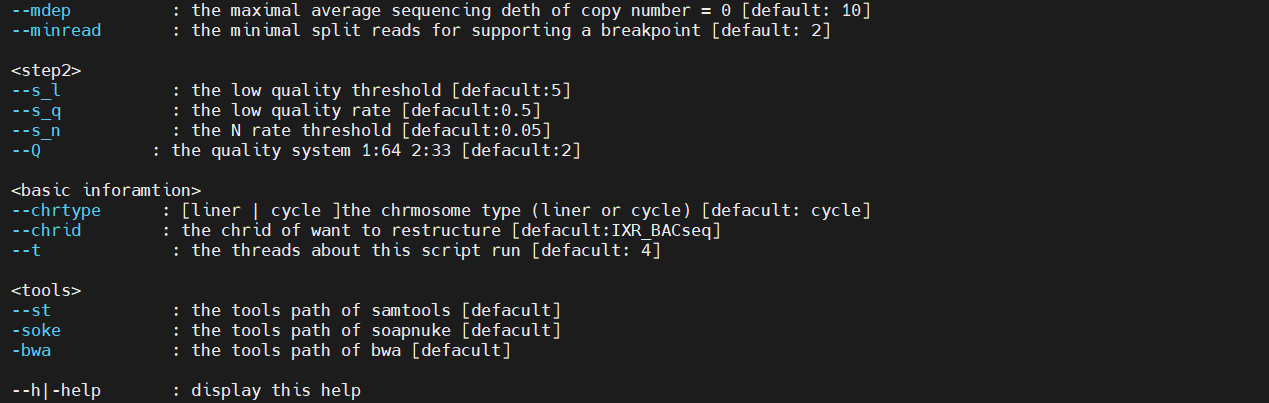
5 bwa（版本不低于0.7.17）。

# 三、操作说明

## 3.1 软件参数

本软件的运行环境为 Unix/Linux操作系统，通过Unix/Linux命令运行。通过 perl run\_Sgrstlfr.pl -h 获取软件的使用信息。详情如下：





基本参数详细说明：

--fa 已经建立index的unSCRaMbLE基因组的fasta文件。

--fq1 stLFR双端测序的1端的fastq文件。

--fq2 stLFR双端测序的2端的fastq文件。

--lxpreg 记录loxp位点位置信息的文件（\*.list.loxpreg）

--blist 记录所有barcode序列信息的清单文件。

--rfcvg unSCRaMbLE基因组的测序深度文件

--n 输出的过程文件以及结果的文件名。【注：过程文件寻找的前置文件名】

--o 输出的过程文件以及最终结果路径。【注：过程文件寻找的路径】

--step 程序运行的步骤，分为三种[all | assembly |others].

all: 运行所有步骤 [defacult]；assembly: 仅仅运行模块7，需要含有模块4-6的过程文件！！！；others: 指定相应的运行步骤 (likes 4,5,6,7)

--chrtype 合成染色体的类型，线性或者环状 [defacult: cycle]

--chrid 合成染色体名称 [defacult:IXR\_BACseq]

--t 软件运行的线程数 [defacult: 4]

--st 给定samtools软件的运行路径 [defacult]

--soke 给定soapnuke软件的运行路径 [defacult]

--bwa 给定bwa软件的运行路径 [defacult]

--h 展示软件使用具体的参数说明。

其余参数解释，见第四章“模块说明”。

## 3.2 输入文件

输入文件主要包括4个文件。主要是母本参考基因组fasta文件，必须包含bwa的index、stLFR双端测序的1端fastq1文件，stLFR双端测序的2端fastq2，包含loxp位点的起始位点信息的loxpreg文件。具体参数如下：

--fa 已经建立index的unSCRaMbLE基因组的fasta文件。

--fq1 stLFR双端测序的1端的fastq文件。

--fq2 stLFR双端测序的2端的fastq文件。

--lxpreg 记录loxp位点位置信息的文件（\*.list.loxpreg）

## 3.3 运行软件举例

1. 采用所有默认参数运行所有步骤：

Perl run\_Sgrstlfr.pl -fa /../\*.fasta -fq1 /../\*.fq1.gz -fq2 /../\*.fq2.gz -n test -o outdr -chrid SynIII -chrtype liner

2. 只运行模块5修改边点提取条件（必须包含过程文件,且文件名相同！！！）

Perl run\_Sgrstlfr.pl -step 5 -n test -o outdir -chrid SynIII -chrtype liner -minread Num1 -mdep -Num2

3. 只运行模块7修改基因组组装条件（必须包含过程文件,且文件名相同！！！）

Perl run\_Sgrstlfr.pl -step 5 -n test -o outdir -chrid SynIII -chrtype liner -bar\_edge Num1 -s\_bs Num2 -p\_nod Num3

4.其他自由组合的步骤。

# 四、模块说明

## 4.1 barcode拆分模块

### 4.1.1 描述

<step1> stLFR的下机数据fq1为100bp，fq2为142bp，其中fq2的42bp为barcode序列。其中barcode的数量为1536种，42bp的barcode序列组成为10-6-10-6-10,10bp为1种类型的barcode，则其总barcode数为1536 X 1536 X 1536种。于是本模块利用给定的barcode清单，进行下机数据barcode的拆分。

### 4.1.2 输入文件

stLFR测序数据双端测序的fq1与fq2、以及记录barcode信息的barcode list。

### 4.1.3 输出文件

过程记录文件：Split\_read\_stat.log、barcode\_freq.txt、split\_barcode.log

Barcode切分结果：split\_reads.1.fq.gz、split\_reads.2.fq.gz

### 4.1.4 用法举例

Perl split\_barcode.pl <barcode\_list> <fastq1> <fastq2> <prefix> <outdir> >split\_barcode.log 2>split\_barcode.err

注：关于barcode切分引用连接所述脚本，详情见：(https://github.com/BGI-Qingdao/stLFR\_barcode\_split)

## 4.2 过滤模块

### 4.2.1 描述

<step2> 本模块采用SOAPnuke软件对barcode拆分数据进行过滤

### 4.2.2 输入文件

Split\_reads.1.fq.gz、split\_reads.2.fq.gz

### 4.2.3 输出文件

过程记录文件若干：Statistics\_of\_Filtered\_Reads.txt、Basic\_Statistics\_of\_Sequencing\_Quality.txt等。

过滤数据：split\_clean\_reads.1.fq.gz、split\_clean\_reads.2.fq.gz

### 4.2.4 参数说明

|  |  |
| --- | --- |
| 参数 | 参数解释 |
| -l （s\_l） | 低质量碱基值 [默认：5] |
| -q (s\_q) | 低质量碱基率 [默认：0.5] |
| -n (s\_n) | N占比率 [默认：0.05] |
| -Q (Q) | 质量系统，1:64,2:33 [默认：2] |

## 4.3 比对模块

### 4.3.1 描述

本模块采用BWA软件对过滤数据进行比对，并使用SAMTOOLS排序转化为BAM文件。需提前使用BWA为reference建立index。

### 4.3.2 输入文件

Split\_reads.1.fq.gz、split\_reads.2.fq.gz

### 4.3.3 输出文件

Test.sort.bam

### 4.3.5 命令详情

bwa mem Test.fa Test.fq1 Testfq2 | samtools view sort -> Test.sort.bam

## 4.4 统计模块

### 4.4.1 描述

<step4> 本模块采用SAMTOOLS对BAM文件进行单碱基的测序深度统计。使用脚本class\_mapping\_result.pl脚本筛选高质量的PE，并且将指定合成染色体上无法比对的含有loxpsym的reads进行切分，再次运用BWA进行重比对产生sam文件（不可sort）。使用脚本getvariareads.pl对重比对结果（sam文件）进行比对方法和位置的统计归类。

### 4.4.2 输入文件

Test.sort.bam

### 4.4.3 输出文件

Test.split.sam、Test.synmap.gz、Test.gvr、Test.depth

### 4.4.4 过程说明

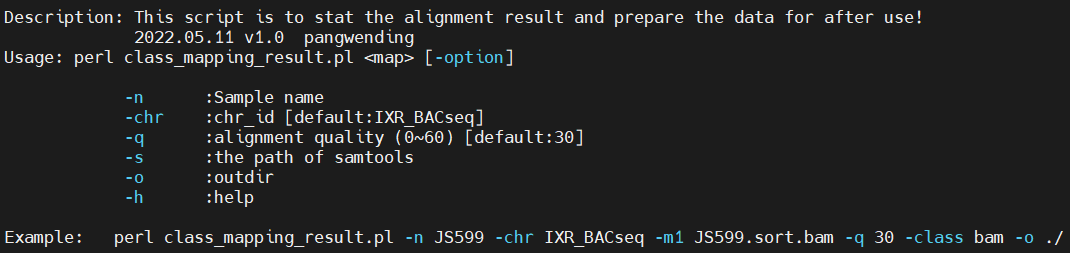
① Samtools depth -a Test.sort.bam | grep “IXR\_BACseq” > Test.depth

使用samtools depth -a 命令统计全染色体的单碱基测序深度，并用grep命令抓取指定染色体的单碱基测序深度。

生成Test.depth 文件

② Perl class\_mapping\_result.pl <bam> -n <prefix> -chr <chrid> -s <samtools> -o <outdir>

使用脚本class\_mapping\_result.pl 筛选bam文件比对质量高的PE reads，同时将无法比对的含有loxpsym位点的reads进行切分。

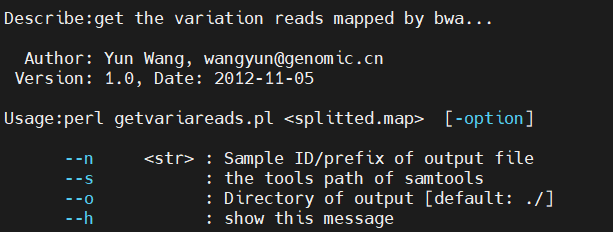


生成 Test.synmap.gz Test.split.fq

③ bwa mem Test.fa Test.split.fq > Test.split.sam

将切分的reads进行重比对，生成新的比对文件 Test.split.sam

④ Perl getvariareads.pl <split.sam> -n <prefix> -s <samtools> -o <outdir>



使用脚本getvariareads.pl 统计重比对的PE reads位置。

## 4.5 边点提取模块

### 4.5.1 描述

<step5> 本模块利用提取的断点，将合成基因组分割为不同的区段，再利用测序深度估计不同区段的拷贝数， 最后获得关于基因组的“边”，“点”关系。

### 4.5.2 输入文件

Test.sort.bam、ref.coverage.depthsingle、Test.gvr、Test.depth

|  |  |
| --- | --- |
| 输入文件 | 解释 |
| <refcoverage> | UnSCRaMbLE基因组的标准测序深度 |
| <bam> | 比对结果的bam文件 |
| <gvr> | 提取的split-reads比对关系 |
| <depth> | 单碱基测序深度 |

### 4.5.3 输出文件

Test.edg、Test.nod

|  |  |
| --- | --- |
| Edg | 包含切分的“边”的基本信息 |
| Nod | 包含筛选的可能“点”的基本信息 |

4.5.4 参数说明

|  |  |
| --- | --- |
| 参数 | 参数解释 |
| -minread | 支持nod的关系的最小reads数。[默认：10] |
| -mdep | 拷贝数为0的最小测序深度。[默认：2] |
| -mcycle | 拷贝数估计过程的循环次数。[默认：10] |
| -cutlen | 估计MD距离时，选取的区段长度。[默认：500] |

注：

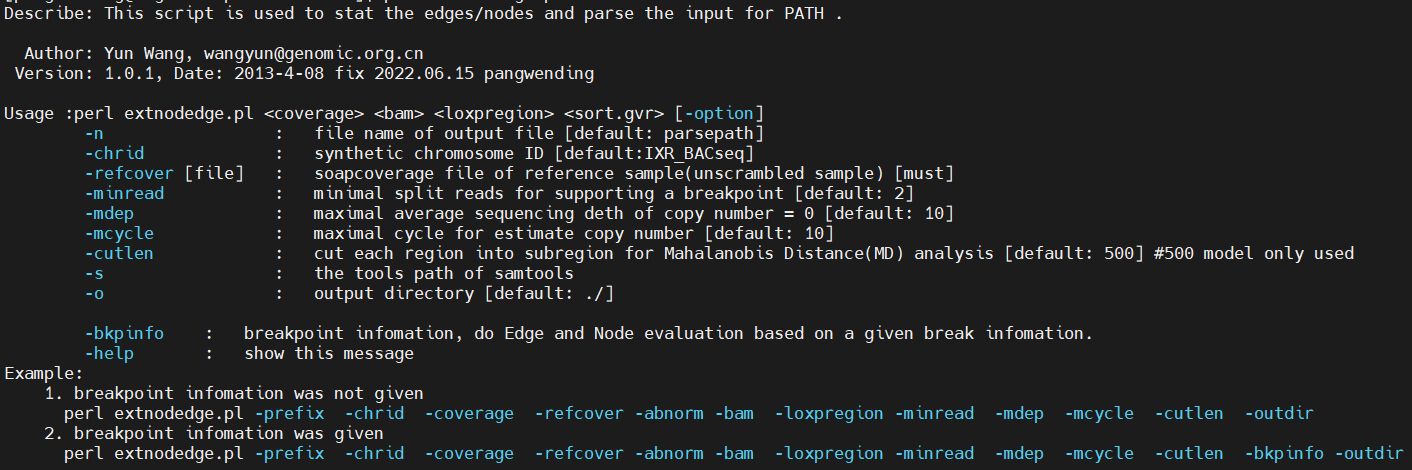
-minread 该参数通过控制支持新断点的最小reads数筛选的点的类型，将影响边的范围已经后续组装的准确性。

-mdep 该参数通过控制最小测序深度的值的边的拷贝数为0，进而影响所有边的拷贝数估计，也将会影响后续组装的准确性。

！！！边点提取模块十分重要，正确的边点关系是组装结果准确的基础。

### 4.5.4 使用说明

Perl extnodedge.pl Test.depth Test.sort.bam Test.list.loxpreg Test.gvr -n Test -chrid IXR\_BACseq -refcover Ref.coverage.depthsingle -minread 10 -mdep 2 -mcycle 10 -cutlen 500 -o ./



## 4.6 barcode信息整合与质控模块

### 4.6.1 描述

<step6> 本模块根据co barcode reads之间的邻近关系，提取测序的长分子上的“边”，“点”关系。因此在有限长度长分子内，“边”，“点”关系是有限的，这将为我们的局部组装提供支撑。

### 4.6.2 输入文件

Test.edg、Test.nod、Test.list.loxpreg、Test.depth、Test.split.sam、Test.synmap.gz

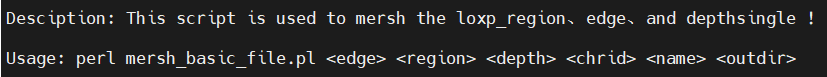
### 4.6.3 输出文件

Test.sort.barcode.stat

### 4.6.4 过程说明

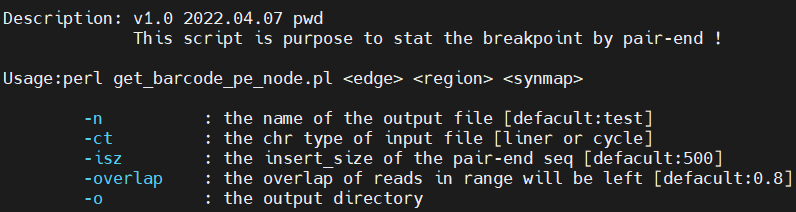
① Perl mersh\_basic\_file.pl <edge> <loxpregion> <depth> <chrid> <prefix> <outdir>

将边、loxpsym位置与深度信息做一个汇总。



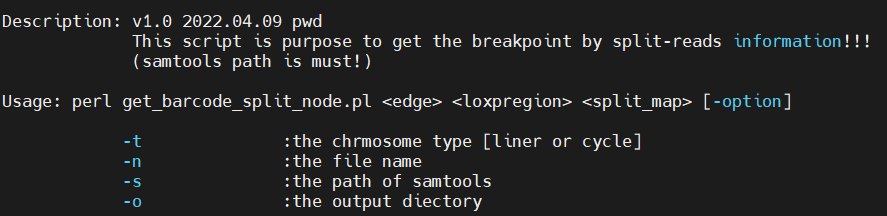
② Perl get\_barcode\_pe\_node.pl <edge> <loxpregion> <synmap> -n <prefix> -o <outdir> -ct <chr\_type> -overlap <overlap> -isz <insert size>

根据PE信息获取co-barcode上的pe.node。其中PE的插入信息和“边”的长度将会影响其准确性。



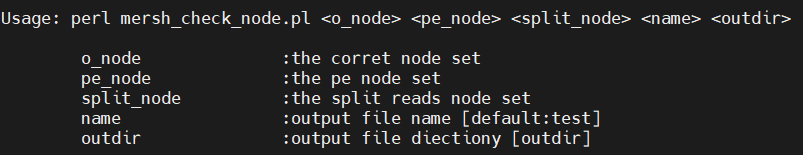
③ Perl get\_barcode\_split\_node.pl <edge> <loxpregion> <split.sam> -s <samtools> -t <chr\_type> -n <prefix> -o <outdir>

根据split reads信息获取长分子上的split.nod。



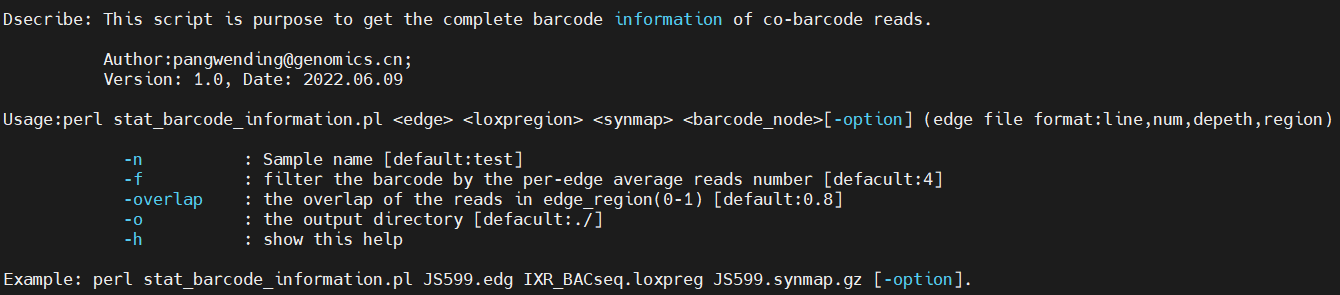
④ Perl mersh\_check\_node.pl <node> <pe.node> <split.node> <name> <outdir>

根据提取的node检查获得的split.nod和pe.node的正确性。



⑤ Perl stat\_barcode\_inforamtion.pl <edge> <synmap> <mersh.nod> -n <prefix> -overlap <overlap> -f <f\_bar>

根据高质量比对的位置，获取长分子包含的“边”类型，并且提取的node对应到相应的barcode上，利用RPKM指标估计长分子内部，“边”的数量比，以及统计相应边上比对的reads数及位置。获取综合的barcode信息。



⑥ Perl arrange\_barcode\_inforamtion.pl <node.barcode> <name> <outdir>

根据“边”的数量，筛选长分子中的“点”数量为：node number > edge number – 1的barcode。长分子中包含的“点”数量大于“边”数量减一即为高质量barcode，通过这些barcode，我们可以进行局部的组装。



### 4.6.3 参数说明

|  |  |
| --- | --- |
| -overlap | 一个reads在一个edge上的覆盖度。[默认：0.8] |
| -isz | PE测序插入序列的大小。[默认：500] |
| -f\_bar | barcode获得的edge类型需要至少包含barcode的多少条reads。[默认：4] |

## 4.7 精准scaffold组装模块和全局筛选

### 4.7.1 描述

<step7> 本模块利用有效的co-barcode信息，选取在基因组中位置唯一的CNV=1的“边”，作为起始，在连接过程中根据长分子的邻接关系，边点的变异信息来实现在局部区域内的精确组装，以此来将多种基因组组装路径进行修剪分支，达到最终基因组组装的目的。

### 4.7.2 输入文件

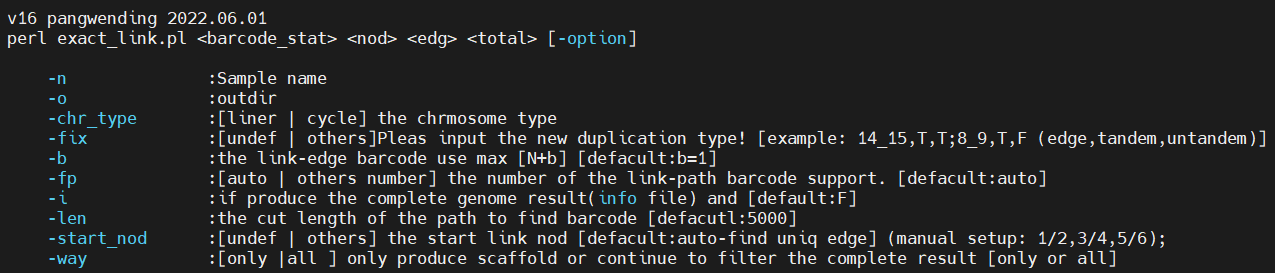
Test.sort.barcode.stat、Test.nod、Test.edg、Test.total

### 4.7.3 输出文件

Test.scaffold、Test.info、Test.result、Test.scaf.log、Test.final、Test.solu.log

### 4.7.2 参数说明

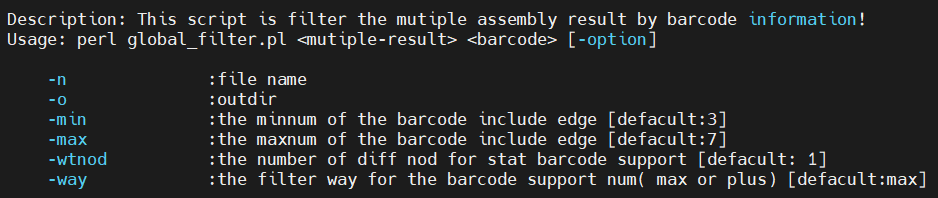
① <step7-1> perl exact\_link.pl <sort.barcode.stat> <nod> <edg> <total> -n <prefix> -o <outdir> -chr\_type <chrtype> -b <bar\_edge\_num> -fix <fix\_dup> -fp <f\_barcode\_num> -I <winfo> -len <start\_len> -way <scaf\_way>



参数解释：

|  |  |
| --- | --- |
| -b | 该参数通过控制局部组装过程中，所利用支持的barcode的边点与子路径的边与点的差值，参数越大，准确度越高，但其局部组装路径越短。[默认：1] |
| -fix | 在进行串联重复与局部重复变异类型的判断时，出现判断错误时，可以根据info文件的结果，来通过该参数进行相应的修正。[默认：undef] |
| -fp | [auto |others]该参数利用barcode的数量来支持局部组装路径的准确性，参数越大，准确性越高，但其局部组装路径越短。一般为根据测序深度自动选择合适参数也可自由指定[默认：auto] |
| -I | [T|F]该参数为该连接过程中，是否要额外生成全部连接路径文件。[默认：F] |
| -len | 该参数通过控制子路径的长度来确定通过2个还是3个边的子路径作为局部组装的起始。[默认：5000] |
| -way | [all|only] 该参数提供两种方式控制程序运行过程，一种为只生成scaffold组装结果，一种为即生成scaffold组装结果又生成基因组组装结果。该参数的作用在于通过调整参数debug生成的scaffold结果的准确性。[默认：all] |
| -chr\_type | 线性染色体还是环状染色体，二者差异在于是否首尾相连，是否包含首尾相连点。[默认：cycle] |

② <step7-2> perl global\_filter.pl <result> <sort.barcode.stat> -n <prefix> -o <outdir> -min <min\_barcode\_edg> -max <max\_barcode\_edge> -wtnod <bs\_edge\_num> -way <solu\_way>



参数解释：

|  |  |
| --- | --- |
| -min | 用于支持该基因组路径barcode内的最小“边”个数。[默认：3] |
| -max | 用于支持该基因组路径barcode内的最大“边”个数。[默认：7] |
| -wtnod | 不同基因组路径差异断点的左右取值个数。[默认：1] |
| -way | [max|plus]筛选基因组路径的方式，取barcode支持最大或是删除barcode支持数小于0的路径。[默认：max] |

## 4.8 序列重建模块

### 4.8.1 描述

<step8> 本模块将边点表示的基因组路径转化为碱基序列（fasta文件）。

### 4.8.2 输入文件

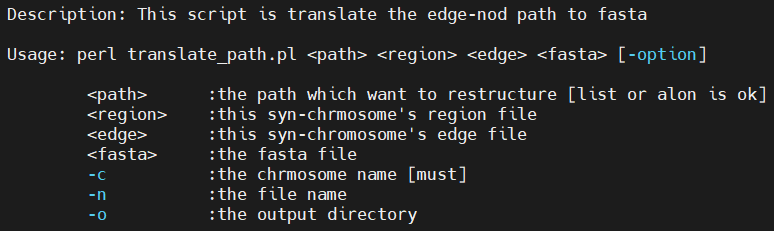
Test.final

### 4.8.3 输出文件

Test.fasta

### 4.8.4 使用说明

Perl translate\_path.pl <final> <loxpregion> <edg> -n <prefix> -o <outdir> -c <chrid>



# 五、运行设计

## 5.1 集成

所有模块与各模块参数集成于独立的perl程序，run\_Sgrstlfr.pl 中，用户可以一键运行该脚本完成所有的模块运行，并且大部分参数已有默认值，只需要根据具体样本进行实际调整。同时，将程序所需包包含，方便快捷的解决运行环境依赖的问题。

## 5.2 分块

所有的模块都可以单独的运行，可以根据模块中间与过程文件，检查程序运行的结果错误。

注：不用的模块错误信息由perl的标准变量（$?）捕获，当该默认变量不为0时，原则上，该模块运行错误。

# 六、常见问题与解决方法

## 6.1 常见问题

①没有edg和nod文件产生

根据extnodedge.log文件，修改-minread与-mdep参数，再重新运行模块5。

②没有result和final文件的结果

根据scaf.log文件，修改模块7的参数，具体解释见4.。