Synthetic genome resturcture by stLFR [Sgrstlfr]

一、前言

合成酵母染色体的SCRaMbLE系统在Cre酶的诱导下，其内含的loxpsym位点可以发生大规模的重排，产生具有大片的插入、缺失、重复和倒位变异的“新”基因组。已有的研究表明，合成酵母的SCRaMbLE系统在高产、抗性菌株的筛选上具有快速而准确的优势，而需要阐明菌株的高产基因和分子机制，离不开对其基因组的研究。目前为了获得完整的SCRaMbLE基因组序列，已有基于二代测序技术开发的组装方法，但是，该方法具有计算资源消耗大，用时久，对于复杂基因组无法精确解析，产生多种基因组组装结果。因此，有必要开发新的方法去解决上述若干问题。

单管长读长测序技术（single tube long fragement sequence techonolgy）,简称为stLFR。该技术利用转座酶在打断的长分子序列中，按照间隔分别插入设计好的相同条形码，再打断为短的序列（下述称为barcode）。随后利用二代测序的方法对打断的序列进行测序。所以，来自于同一个长分子的短测序读长，会包含相同的barcode，在后续处理中，对相应的barcode进行拆分后，就会部分的还原长分子的信息，以此来达到长测序读长的目的。

基于stLFR测序数据的特点。在基因组的组装过程中，如果碱基序列来源与同一长分子，相比于不同位置的重复序列，则其邻接距离会更近。基于SCRaMbLE基因组的特点。对于重构的含义，其实是区分与重测序与从头组装的传统组装方法。Cre-loxpsym系统在重排过程中是以loxpsym位点区分的片段化方式发生“排列组合“，相比与参考基因组，不同区段的碱基序列并没有发生变化，变化在于其发生的删除、插入、重复和倒位等大片段的复杂结构变异。其“新”的SCRaMbLE基因组在某种意义上已经是全新的基因组了，但是相比与参考基因组，只是排列方式发生改变。根据二者特性，我们开发了新的合成酵母SCRaMbLE基因组重构的方法：Synthetic genome restructure by stLFR[Sgrstlfr]。

Sgrstlfr中利用loxpsym序列寻找发生重排的断点，将整个基因组按照给定顺序进行数字排列，不同的区段划分为“边”，根据测序深度估计“边”的数量，而断点则是两个区段的连接方法，称为“点”。而stLFR的长分子共标签信息（下述简称为co-barcode信息），能够将“边”与“点”定位到局部的长分子序列中，进行局部的组装，于是可以减少因为整个基因组的大规模变异产生的基因排列顺序的多个分支。利用多个co-barcode信息进行局部组装成完整的scaffold，随后再利用所有的“边”与“点”的信息获取所有可能的基因组组装结果，利用精确的scaffold结果进行完整基因组的筛选，随后获得了部分准确的基因组组装方式。最后，我们考虑到特殊“点”，即分支，在整个基因组中会有其独属的位置信息，于是我们再利用不同基因组排列的差异“点”，进行全局的筛选，最终获得了少部分的精确SCRaMbLE基因组。

二、软件简介

Sgrstlfr使用perl进行编程，总共包含14个核心脚本，3个包和1个运行脚本。共分为8个模块，分别为：①barcode切分模块，②数据质控模块，③比对模块，④比对结果统计模块，⑤边点提取模块，⑥co-barcode信息提取与质控模块，⑦scaffold和基因组组装模块，⑧基因组翻译模块。8个模块可以分别独立运行，全流程运行需要有以下5个必须文件，以及基本的基因组信息。5个文件包括：reference、fastq1、fastq2、loxpregion、refcoverage

Reference：没有发生SCRaMbLE的参考基因组

Fastq1: PE测序的5’端

Fastq2 PE测序的3’端

Loxpregion：loxpsym位点在基因组中位置信息

Refcoverage： 没有发生SCRaMbLE基因组的测序深度信息

其余具体参数见运行脚本：run\_Sgrstlfr.pl 的帮助信息。 Perl run\_Sgrstlfr.pl -h

Description: This script is used to restructure the synthethic yeast SCRaMbLE genome by stLFR technology !

v1.0 2022.06.12 wendingpang

Usage: perl run\_Sgrstlfr [-option]

--fa [file] : <fasta> the unSCRaMbLE genome FASTA [bwa index must!]

--fq1 [file] : <fastq1> the stLFR FASTQ1

--fq2 [file] : <fastq2> the stLFR FASTQ2

--lxpreg [file] : <loxp\_region> the loxp region file

--blist [file] : <barcode\_list> the barcode\_list from stLFR-seq [default]

--rfcvg [file] : <refcoverage> the seqence of unSCRaMbLE genome's depth

--n : the output median file and directory name [defacult: test]

--o : the output file directory and the median-file directory [defacult: ./]

--step : [all|assembly|others] which steps you want to run Sgrstlfr.

all:run the all [defacult]

assembly: only run the module7 to repair the scaffold and genome assembly

others: you appoint which step to run (likes 4,5,6,7)

<step7-2>

--min\_edge : :the minnum of the barcode include edge [defacult:3]

--max\_edge : the maxnum of the barcode include edge [defacult:7]

--p\_nod : the left and right nod number from diff nod start to stat bs [defacult:1]

--solu\_way : [max | plus ] the solution use barcode support way (max: get the max bs path or plus:get the bs more than 0 path) [defacult:max]

<step7-1>

--start\_nod : the scaffold assembly start type.(defacult auto find uniq edge as start or appoint mutipe nod) [default:undef]

--winfo : [F | T ]in the scaffold assembly if produce the all complete edge-nod link set [defacult:F]

--s\_len : the contig length of start to scaffold assembly [defacult:5000]

--s\_way : [only | all] only:only produce all scaffold set not include filter genome path all:produce all result [defacult:all]

--fix\_dup : if from the info file find the duplication type is erro in the scaffold assembly.

you can direct repair in this paramter likes (0\_1,T,T;2\_3,F,T) [defacult:undef]

--bar\_edge : the process of assembly scaffold you use barcode max edge number (N+this paramter) [defaclut:1]

--s\_bs : the barcode\_support of sub\_path (more than Zero) [defacult:3]

<step6>

--overlap : the alon-reads coverage one edge precentage [deafacult:0.8]

--isz : the PE seqence average insert\_size [defacult:500]

--f\_bar : the number of every edge include reads. [defacult:4].

<step5>

--cutlen : cut each region into subregion for Mahalanobis Distance(MD) analysis [default: 500] #500 model only used

--mcycle : the maximal cycle for estimate copy number [default: 10]

--mdep : the maximal average sequencing deth of copy number = 0 [default: 10]

--minread : the minimal split reads for supporting a breakpoint [default: 2]

<basic inforamtion>

--chrtype : [liner | cycle ]the chrmosome type (liner or cycle) [defacult: cycle]

--chrid : the chrid of want to restructure [defacult:IXR\_BACseq]

--t : the threads about this script run [defacult: 4]

<tools>

--st : the tools path of samtools [defacult]

-soke : the tools path of soapnuke [defacult]

-bwa : the tools path of bwa [defacult]

--h|-help : display this help

三、模块解析

1. barcode拆分模块

barcode的组成：100-10-6-10-6-10

perl split\_barcode.pl <barcode\_list> <fastq1> <fastq2> <name> <prefix>

根据给定的barcode清单将下机数据进行barcode拆分，注：0\_0\_0 类型barcode为拆分失败的barcode。

1. 数据过滤模块

采用soapnuke作为过滤个工具进行数据的过滤，详细见soapnuke使用说明

1. 比对模块

采用bwa进行比对，并用samtools转化为bam文件并建立index

1. 比对统计模块

对比对结果（bam）文件进行统计，包括使用samtools进行单碱基的深度统计，对无法比对的含有loxpsym的reads进行切分重比对，以及对高比对质量的PE进行归类。

* 1. Samtools depth -a \*.sort.bam | grep “IXR\_BACseq” > test.depth
  2. Perl class\_mapping\_result.pl <bam> -n <prefix> -chr <chrid> -s <samtools> -o <outdir>
  3. Perl getvariareads.pl <split.sam> -n <prefix> -s <samtools> -o <outdir>

1. 边点提取模块

边点提取模块采用提取无法正常比对的含loxpsym的reads，获取相应断点，以及利用测序深度估计CNV，来获得“边”，“点”信息。

其中，“点”的筛选根据参数 -minread 控制，该参数的含义是支持该断点的最小reads数，默认为2

其中，“边”的筛选根据参数 -mdep 控制，该参数的含义是设定测序深度小于多少，其CNV=0，默认为10

Perl extnodedge.pl <depth> <bam> <gvr> -refcover <refcoverage> -n <prefix> -o <outdir> -s <samtools> --minread <minread> -mdep <mdep> -mcycle <mcycle> -cutlen <cutlen>

1. Co-barcode信息提取与质控模块

该模块将进行barcode信息的质控以及分别将获得的“边”与“点”的信息锚定到相应的长分子上，将co-barcode信息与“边”，“点”信息结合，为后续局部组装做准备。

* 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。

* 1. 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。

* 1. 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。

* 1. 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。

* 1. 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。

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

利用“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。

1. Scaffold和基因组组装模块

该模块利用有效的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。

* 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。

-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。

-fp ：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。

-I ：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

-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。

-way ：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。

-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

* 1. 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

-max ：用于支持该基因组路径barcode内的最大“边”个数

-wtnod ：不同基因组路径差异断点的左右取值个数

-way ：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

1. 基因组翻译模块

将获得的数字“边”，“点”表示的基因组组装结果翻译为fasta文件

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

四、具体例子

1. 步骤3-8
2. 步骤5
3. 步骤7
4. 步骤1-8