高通量测序质控及可视化工具包RSeQC

原创 2017-06-30 白衣卿相 生信杂谈

> 本期介绍一个高通量测序质控的工具包:RSeQC包,它提供了一系列有用的小工具能够评估高通量测序尤其是RNA-seq数据.比如一些基本模块; `检查序列质量` , `核酸组分偏性` , `PCR偏性` , `GC含量偏性` ,还有RNA-seq特异性模块: `评估测序饱和度` , `映射读数分布` , `覆盖均匀性` , `链特异性` , `转录水平RNA完整性` 等。

安装:

`RSeQC` 是依赖于python的,直接使用 `pip` 进行安装:

```
pip install RSeQC
```

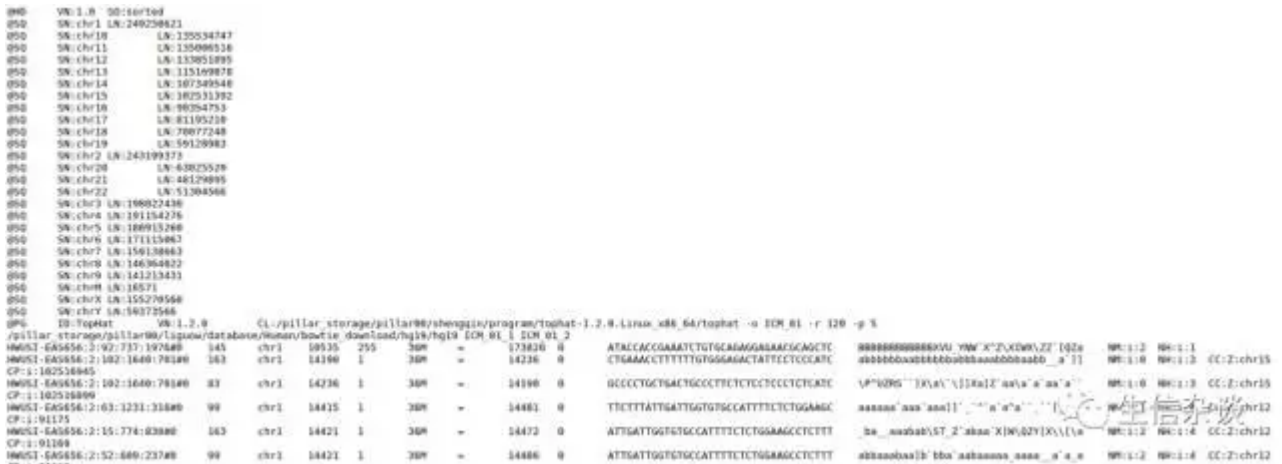
输入数据格式:

RSeQC接受4种文件格式:

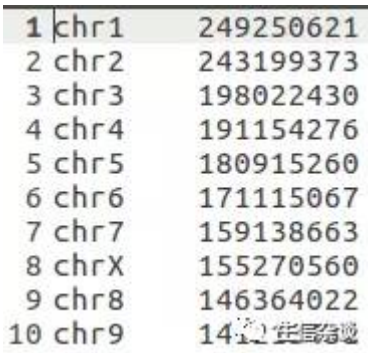
- `BED` 格式: `Tab` 分割, `12列` 的表示基因模型的纯文本文件,例如:

```
chr1    29553   31097   ENST00000473358.1   0   +   29553   31097   0   3   486,104,122,    0,1010,1422,
chr1    30266   31109   ENST00000469289.1   0   +   30266   31109   0   2   401,134,    0,709,
chr1    30365   30503   ENST00000607096.1   0   +   30365   30503   0   1   138,    0,
chr1    35244   36073   ENST00000461467.1   0   -   35244   36073   0   2   237,353,    0,476,
chr1    34553   36081   ENST00000417324.1   0   -   34553   36081   0   3   621,205,361,    0,723,1167,
chr1    52472   53312   ENST00000606857.1   0   +   52472   53312   0   1   840,    0,
chr1    53048   54936   ENST00000594647.1   0   +   53048   54936   0   2   19,107, 0,1781,
chr1    62947   63887   ENST00000492842.1   0   +   62947   63887   0   1   940,    0,
chr1    69090   70008   ENST00000335137.3   0   +   69090   70008   0   1   918,    0,
chr1    89550   91105   ENST00000495576.1   0   -   89550   91105   0   2   500,819,    0,736,
chr1    89294   120932  ENST00000466430.1   0   -   89294   120932  0   4   2335,150,105,158,   0,2796,23405,31480,
chr1    92229   129217  ENST00000477740.1   0   -   92229   129217  0   4   11,105,212,163, 0,20470,28491,36825,
chr1    110952  129173  ENST00000471248.1   0   -   110952  129173  0   3   405,105,119,    0,1747,18102,
chr1    131024  134836  ENST00000442987.3   0   +   131024  134836  0   1   3812,   0,
chr1    129080  133566  ENST00000453576.2   0   -   129080  133566  0   2   143,193,    0,4293,
chr1    135140  135895  ENST00000494149.2   0   -   135140  135895  0   1   755,    0,
chr1    134900  139379  ENST00000423372.3   0   -   138529  139379  0   2   902,1759,   0,2720,
chr1    137681  137965  ENST00000595919.1   0   -   137681  137965  0   1   284,    0,
```


- `SAM` 或 `BAM` 格式: 用来存储 `reads` 比对结果信息. `SAM` 是可读的纯文本文件,然而 `BAM` 是 `SAM` 的二进制文本,一个压缩的可索引的 `reads` 比对文件. 例如:



- 染色体大小文件: 只有两列的纯文本文件,这个之前在生物信息学文本处理大杂烩(一)里已经讲过. `hg19.chrom_24.sizes` 是人基因组hg19版本的size文件，是使用UCSC 的 `fetchChromSizes` （从这里下载：http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/） 下载的.



1	chr1	249250621
2	chr2	243199373
3	chr3	198022430
4	chr4	191154276
5	chr5	180915260
6	chr6	171115067
7	chr7	159138663
8	chrX	155270560
9	chr8	146364022
10	chr9	141

- Fasta文件.

使用方法:

最新版本的 `RSeQC(2.6.4)` 包含以下一些模块,每个模块都可以单独调用进行分析:

- bam2fq.py
- bam2wig.py
- bam_stat.py
- clipping_profile.py
- deletion_profile.py
- divide_bam.py
- FPKM_count.py
- geneBody_coverage.py
- geneBody_coverage2.py
- infer_experiment.py
- inner_distance.py
- insertion_profile.py
- junction_annotation.py
- junction_saturation.py
- mismatch_profile.py
- normalize_bigwig.py
- overlay_bigwig.py
- read_distribution.py
- read_duplication.py
- read_GC.py
- read_hexamer.py
- read_NVC.py
- read_quality.py
- RNA_fragment_size.py
- RPKM_count.py
- RPKM_saturation.py
- spilt_bam.py
- split_paired_bam.py
- tin.py

我们一个一个来看:

> **bam2fq.py:**

将 BAM 或 SAM 格式的文件转为 fastq 格式.(这个感觉一般用不到)

---

bam2wig.py:

将BAM文件转为 wig / bigwig 格式的文件.(这个在作图尤其是信号图的时候很有用.)如果需要转为 bigwig 格式,则需要UCSC的 wigToBigWig 工具,下载地址:
http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/wigToBigWig

---

bam_stat.py:

对比对结果文件 BAM 或 SAM 文件进行统计.其实 samtools 里也有类似工具.结果如下所示:

```
bam_stat.py  -i Pairend_nonStrandSpecific_36mer_Human_hg19.bam

#Output (all numbers are read count)
#==================================================
Total records:                                 41465027
QC failed:                                     0
Optical/PCR duplicate:                         0
Non Primary Hits                               8720455
Unmapped reads:                                0

mapq < mapq_cut (non-unique):                  3127757
mapq >= mapq_cut (unique):                     29616815
Read-1:                                        14841738
Read-2:                                        14775077
Reads map to '+':                              14805391
Reads map to '-':                              14811424
Non-splice reads:                              25455360
Splice reads:                                  4161455
Reads mapped in proper pairs:                  21856[illegible]4生信杂谈
Proper-paired reads map to different chrom:    7648
```

统计结果包括: 总比对记录 , PCR重复数 , Non Primary Hits 表示多匹配位点. 不匹配的reads数 , 比对到+链的reads , 比对到-链的reads , 有剪切位点的reads 等.

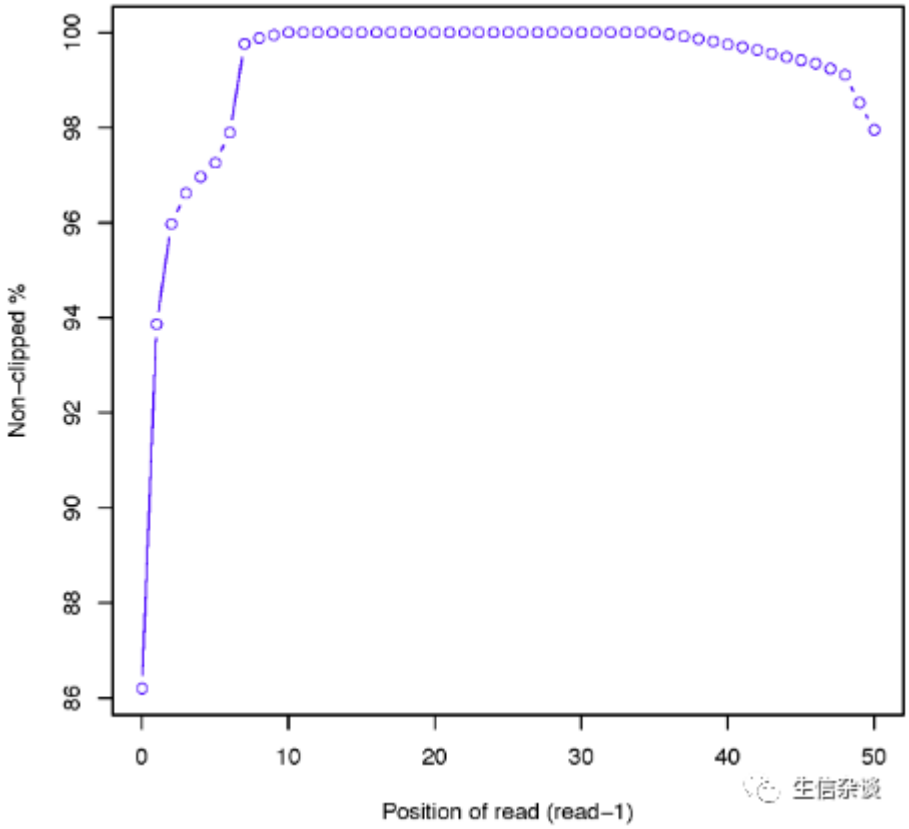
---

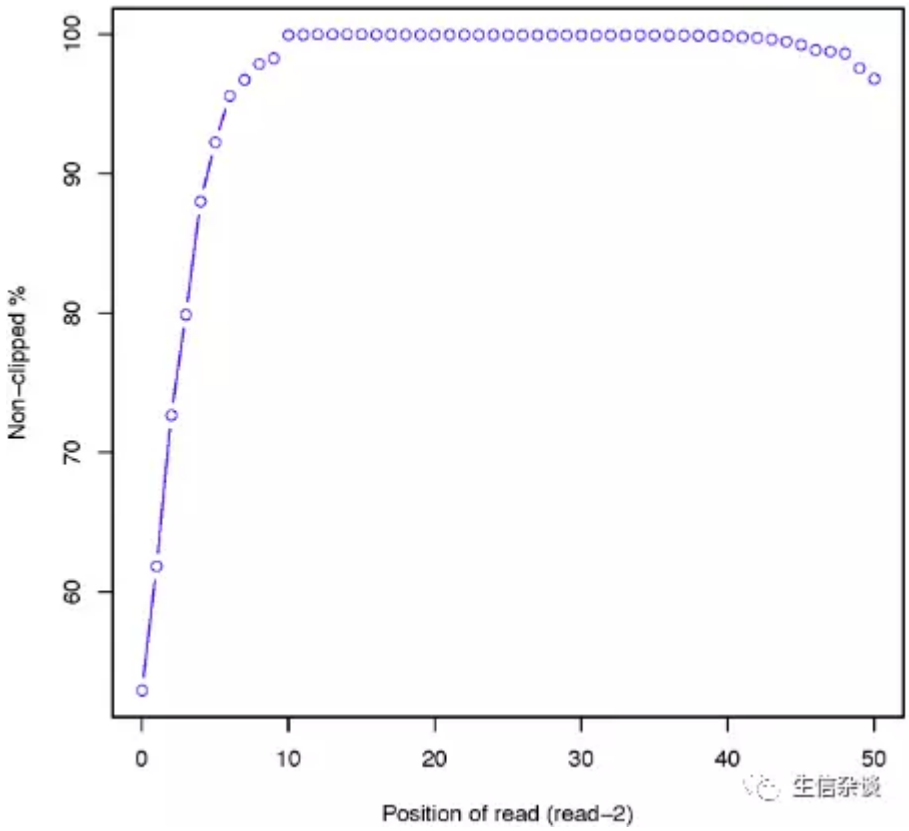
clipping_profile.py:

这个模块用于评估 RNA-seq 的 BAM 或 SAM 文件中的 有切除核苷酸的reads 情况.

clipped reads 有两种,一种是 soft-clipped ,即 reads 5'或3'不能比对到参考基因组;另一种是 hard-clipped ,即 reads 5'或3'不能比对到参考基因组并且被剪切.

这个模块会生成 .r 格式的作图脚本以及 .pdf 格式的报告文件以及 .xls 的数据文件.
例如双端测序 clipping profile 图如下:

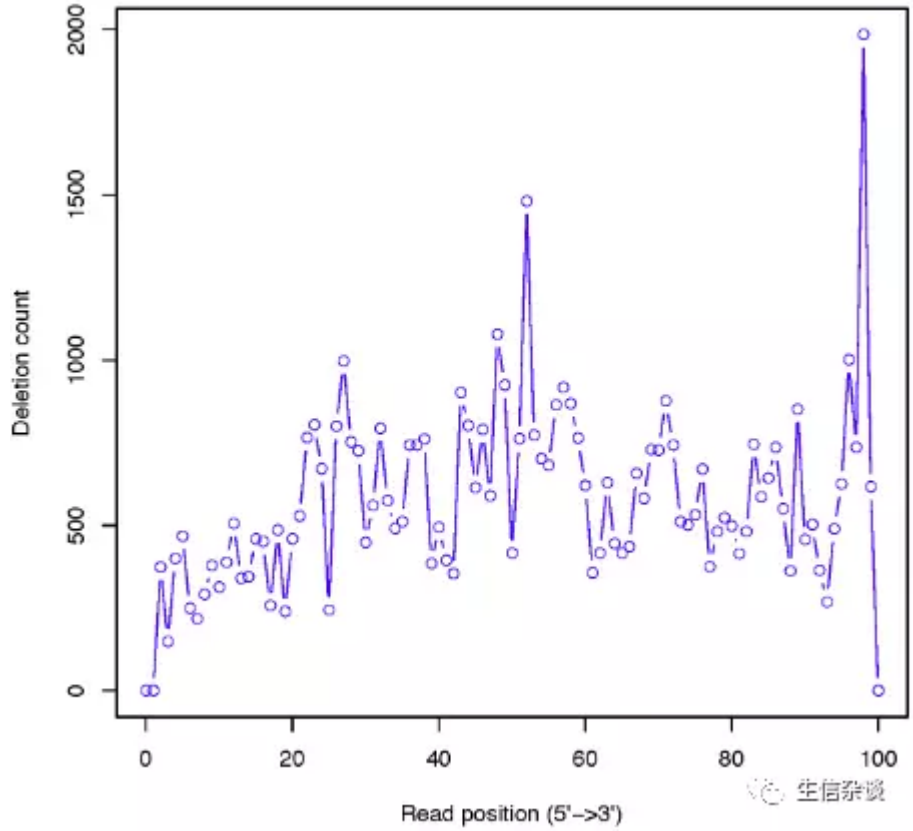




---

deletion_profile.py:

也就是 `reads deletion` 位点的分布.



---

divide_bam.py:

随机分割 `BAM` 文件(m个比对结果)为n个文件,每个文件包含 `m/n` 个比对结果.

FPKM_count.py:

根据 `read count` 和 `gene注释文件` (bed12格式)计算每个基因的 `FPM` (fragment per million)或者 `FPKM` (fragment per million mapped reads per kilobase exon).

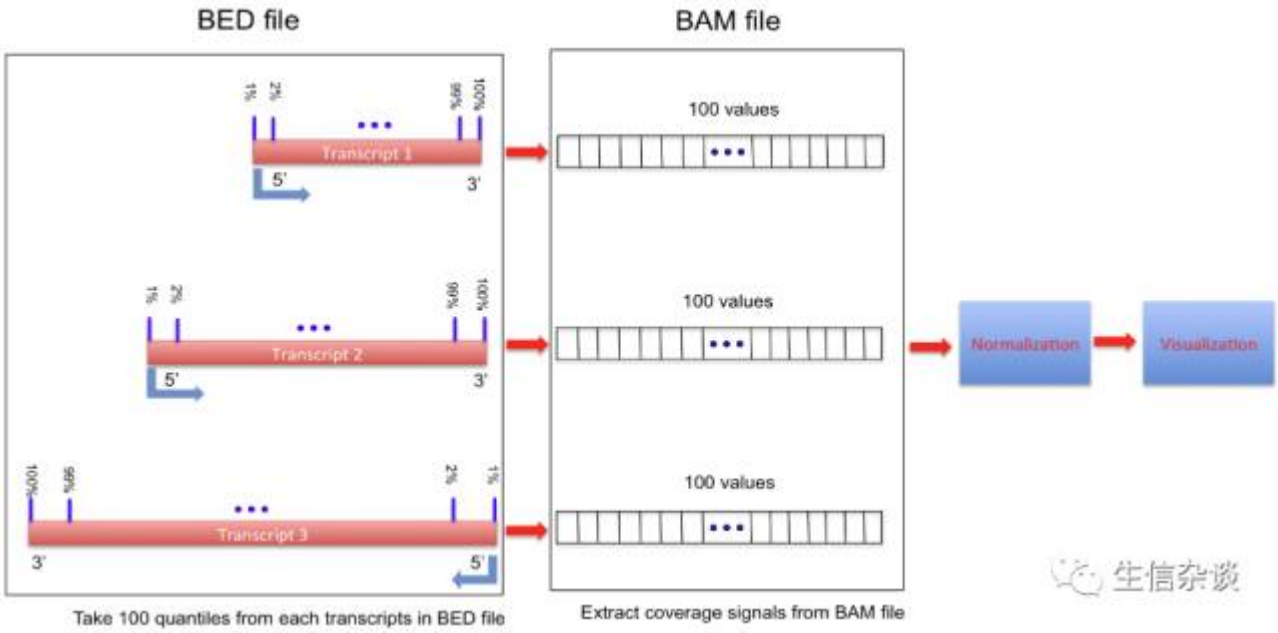
结果类似下面的:

#chrom	st	end	accession	mRNA_size	gene_strand	Frag_count	FPM	FPKM
chr1	100652477	100715409	NM_001918	10815.0	'-'	5498.0	191.73788949	17.728884835
chr1	175913961	176176380	NM_022457	2789.0	'-'	923.0	32.188809021	11.541344217
chr1	150980972	151008189	NM_021222	2977.0	'+'	687.0	23.958517657	8.0478729115
chr1	6281252	6296044	NM_012405	4815.0	'-'	1396.0	48.684265[illegible]	10.[illegible]864
chr1	20959947	20978004	NM_032409	2660.0	'+'	509.0	17.750925018	[illegible]821
chr1	32479294	32509482	NM_006559	2891.0	'+'	2151.0	75.014223408	25.947500314

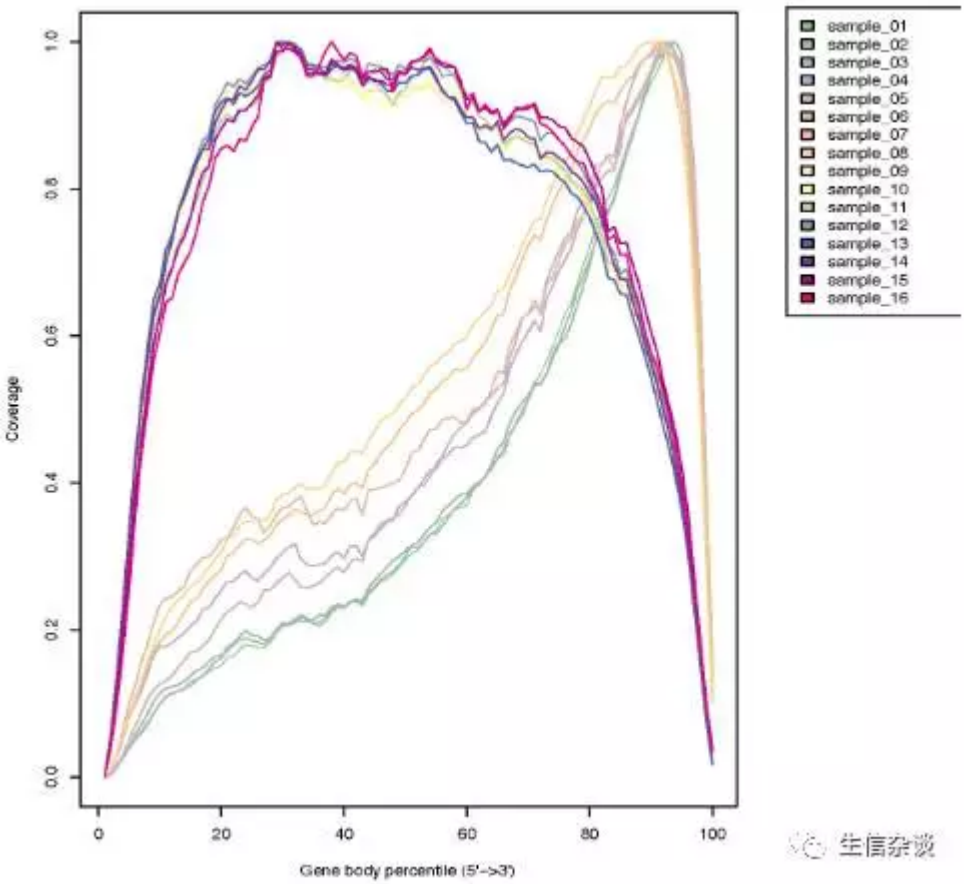
---

geneBody_coverage.py:

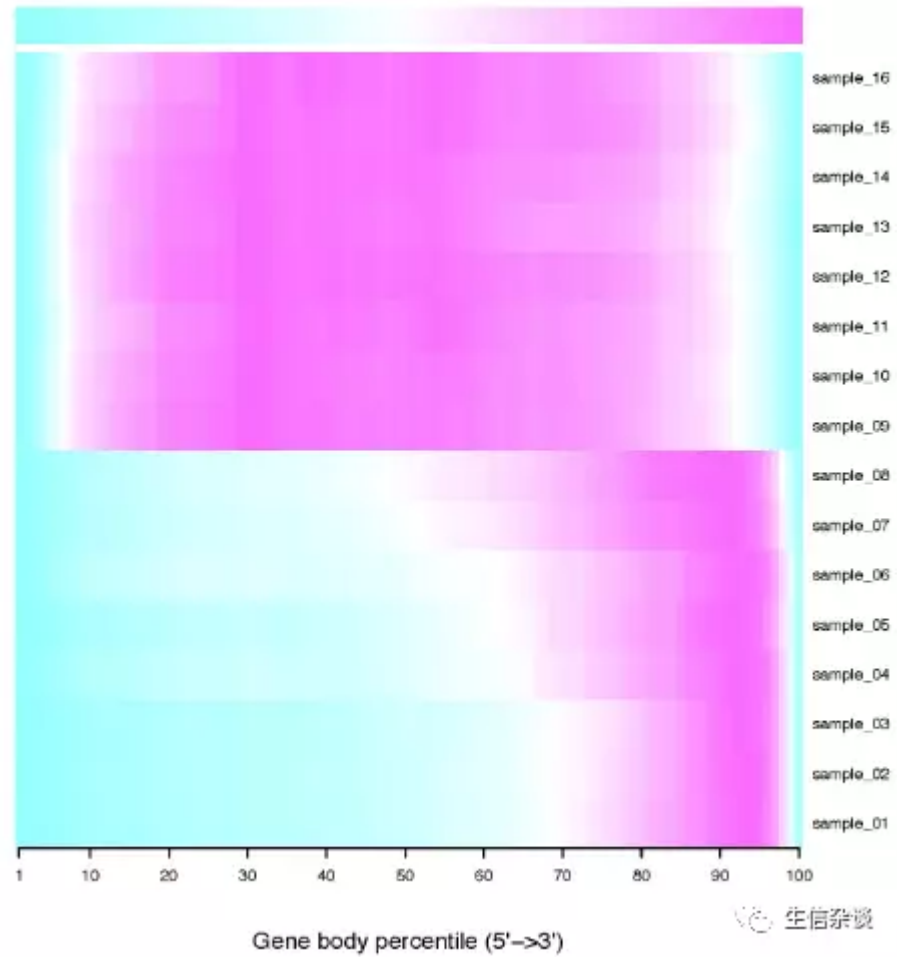
计算RNA-seq `reads` 在基因上的覆盖度.



结果如下:



如果是大于3个的样本,则还会生成热图:



> **geneBody_coverage2.py:**
>
> 功能和上面的 `geneBody_coverage.py` 一样,但输入的是 `bigwig` 格式文件

> **infer_experiment.py:**

1. 这个模块用来”猜”RNA-seq的相关配置信息,针对**链特异性测序**,通过 `reads的链型` 与 `转录本的链型` 来评估 `reads` 是哪一条链的.
2. `reads的链型` 是通过比对结果得到的, `转录本的链型` 是铜鼓注释文件得到的.
3. 对于**非链特异性测序**, `reads的链型` 与 `转录本的链型` 是没有关系的.
4. 对于**链特异性测序**, `reads的链型` 与 `转录本的链型` 是有很大关系的.通过下面的3个例子说明.
5. 在测序前你并不需要知道RNA-seq是不是链特异性的,就当他们是非链特异性的,这个模块可以”猜”到”reads”是哪条链的.

对于双端 `RNA-seq` ,有两种方法来确定reads在哪条链(如illumina ScriptSeq protocol):

(1). 1++,1–,2+-,2-+ :
说明: `1` 和 `2` 表示 `read1` 和 `read2` ,第一个 `+/-` 表示read map 到哪条链,第二个 `+/-` 表示这个 read 所match的基因在哪条链.

那这个 `1++,1–,2+-,2-+` 就表示 `reads` 所match的链和其所在gene的”+/-“是一样的,也就是reads的链型与其基因的链型一样,是不独立的.

(2). 1+-,1-+,2++,2−−:
这个就表示 `reads` 所match的链和其所在gene的"+/-"是不一样的,也就是reads的链型与其基因的链型是不独立的.

对于单端测序:

(1). ++,−−:表示 `read链型` 与其所match的 `gene链型` 一致.
(2). +-,-+:表示 `read链型` 与其所match的 `gene链型` 不一致.

举三个例子说明:

例一:

```
infer_experiment.py -r hg19.refseq.bed12 -i Pairend_nonStrandSpecific_36mer_Human_hg19.bam

#Output::

This is PairEnd Data
Fraction of reads failed to determine: 0.0172
Fraction of reads explained by "1++,1--,2+-,2-+": 0.4903
Fraction of reads explained by "1+-,1-+,2++,2--": 0.4925
```


解释: 总 `reads` 数的 `1.72%` 被映射到基因组区域，我们无法确定这样的 `gene的链型` （比如这个区域两条链都转录）。 对于剩余的 `98.28%` （1 - 0.0172 = 0.9828）的reads，一半是"1 ++，1-，2 + - ，2- +" `reads` 与 `gene` 链型一致的，而另一半是"1 +1 - +2++，2-"不一致的。 我们得出结论，**这不是一个链特异性的测序**，因为 `reads链型` 不依赖于 `gene链型` .

例二:

```
infer_experiment.py -r hg19.refseq.bed12 -i Pairend_StrandSpecific_51mer_Human_hg19.bam

#Output::

This is PairEnd Data
Fraction of reads failed to determine: 0.0072
Fraction of reads explained by "1++,1--,2+-,2-+": 0.9441
Fraction of reads explained by "1+-,1-+,2++,2--": 0.0487
```


解释: `0.72%` 是不能确定链型的. `94.41%` 的 `reads` 与 `gene` 链型一致的.仅有 `4.87%` 是不一致的. 我们得出结论，**这是一个链特异性的测序**，因为 `reads链型` 依赖于 `gene链型` .

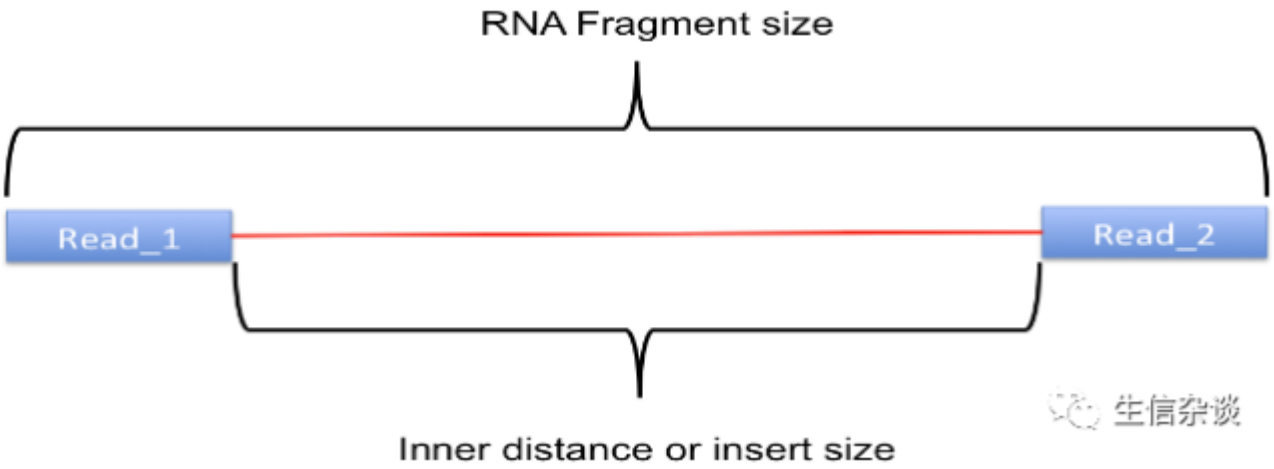
例三,这是个单端测序的例子:

```
infer_experiment.py -r hg19.refseq.bed12 -i SingleEnd_StrandSpecific_36mer_Human_hg19.bam

#Output::

This is SingleEnd Data
Fraction of reads failed to determine: 0.0170
Fraction of reads explained by "++,--": 0.9669
Fraction of reads explained by "+-,-+": 0.0161
```


解释: `reads链型` 依赖于 `gene链型` ,这个是链特异性测序.

inner_distance.py:

针对双端测序,计算 `read pairs` 的 `内部距离` 或者 `插入距离` ,关于 `插入距离` 是什么,看下图:



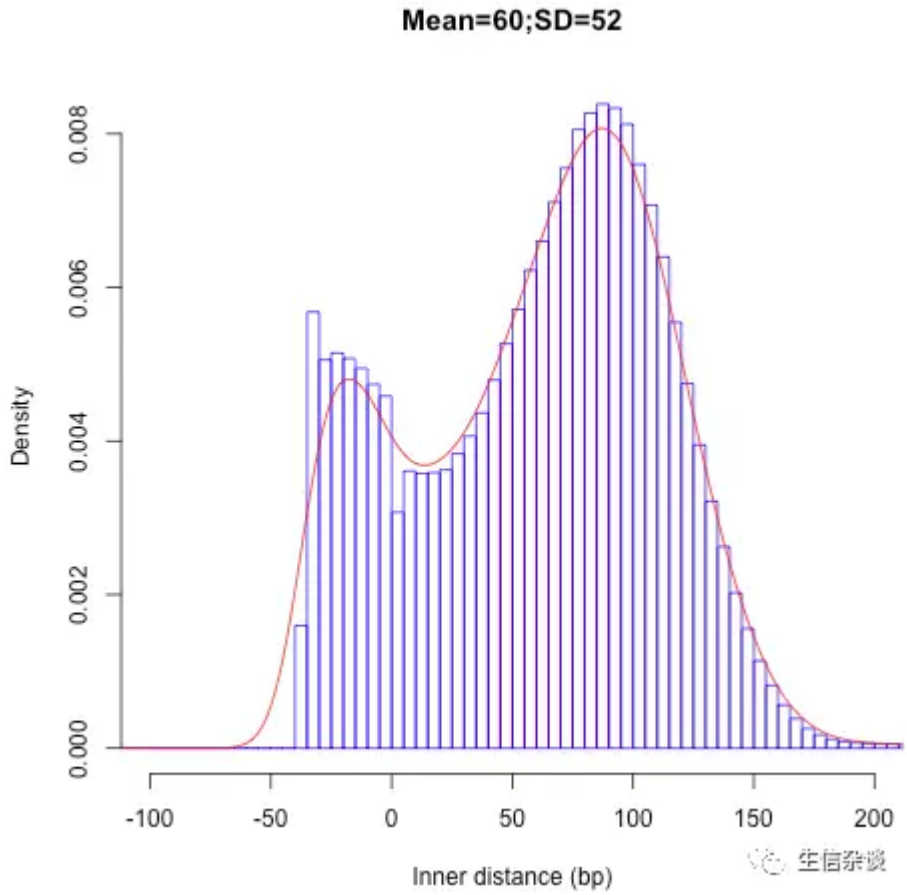
插入距离 `D_size` 计算公式:

D_size = read2_start - read1_end

但是不同条件下计算方法是不一样的:

- `paried reads` 比对到同一个外显子: `插入距离` = D_size
- `paried reads` 比对到不同外显子: `插入距离` = D_size - intron_size
- `paried reads` 比对到非外显子区域(如内含子或者基因外区域): `插入距离` = D_size
- 如果两个 `fragments` 重叠则 `插入距离` 可能为负.

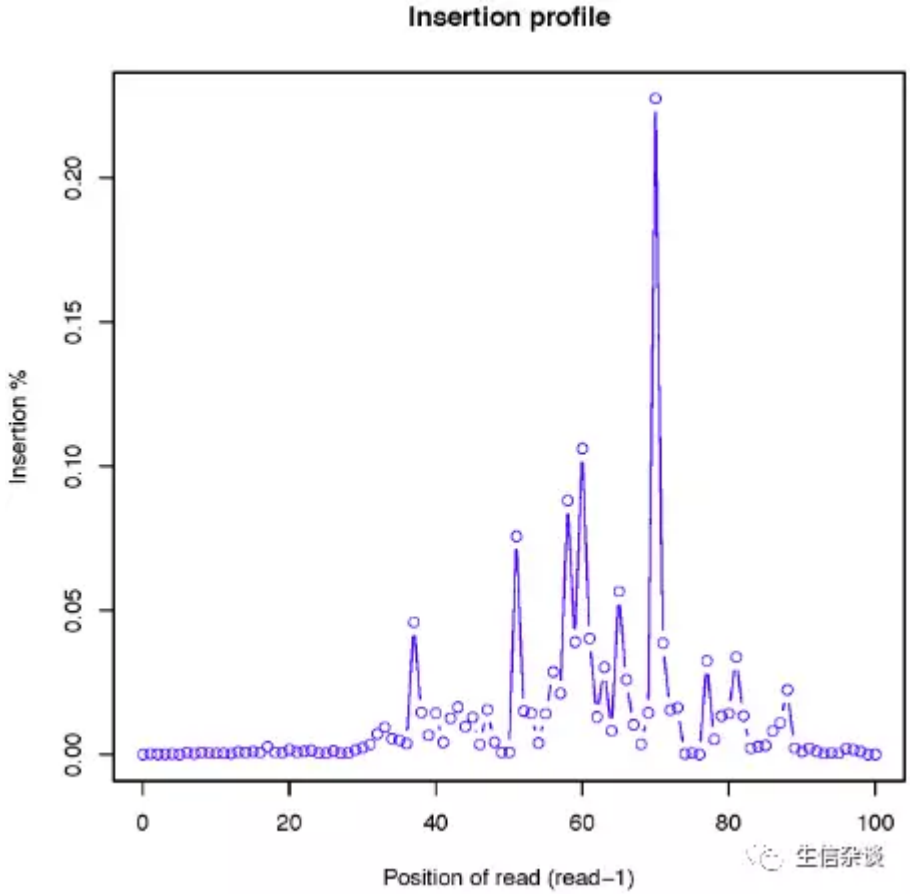
结果举例:

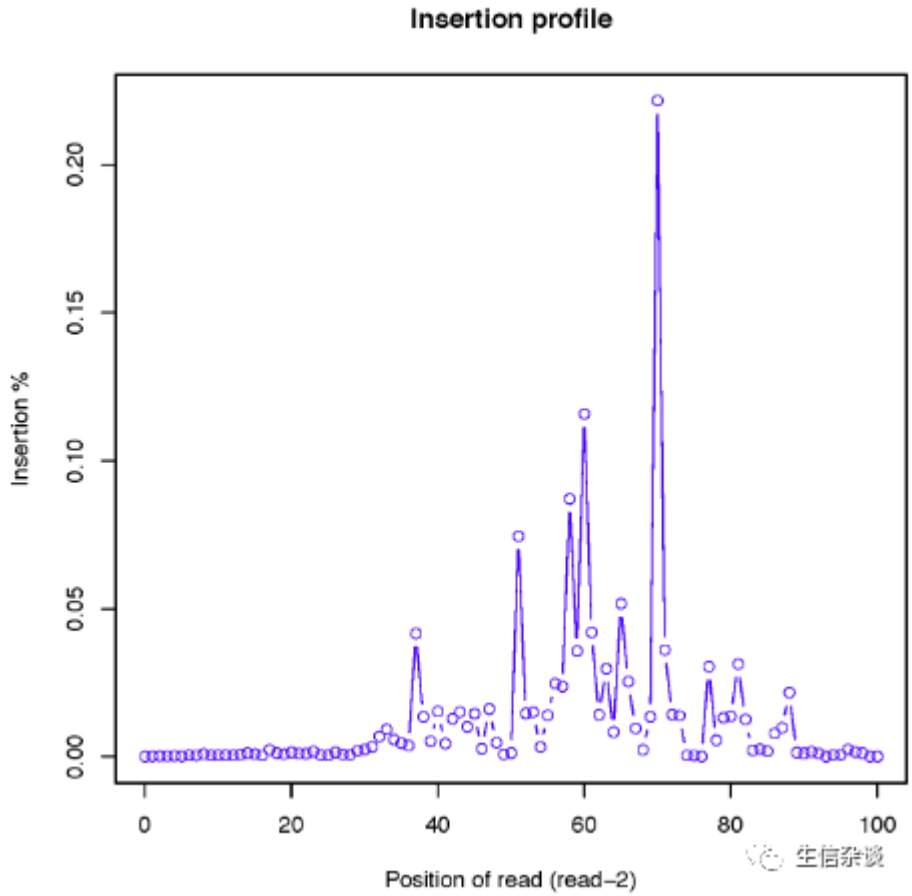


> **insertion_profile.py:**

计算reads上被插入核苷酸的分布.

结果举例:





> junction_annotation.py:

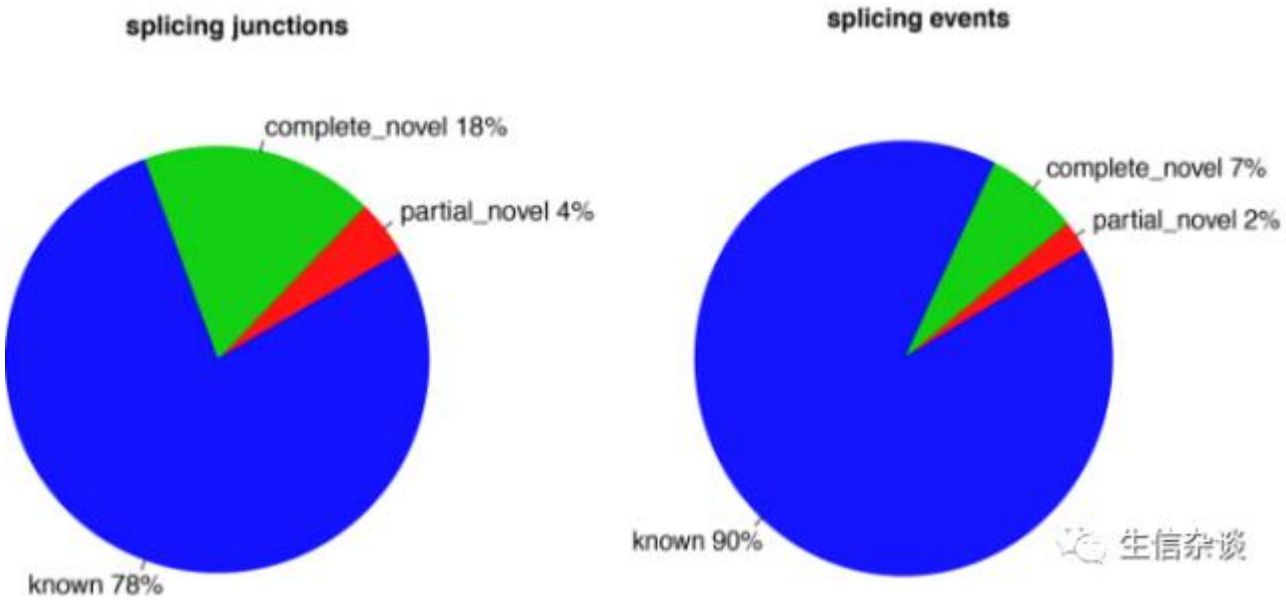
输入一个 `BAM` 或 `SAM` 文件和一个 `bed12` 格式的参考基因文件,这个模块将根据参考基因模型计算剪切融合(splice junctions)事件.

- **splice read**: 一个RNA read,能够被剪切一次或多次,所以100个 `spliced reads` 能够产生>=100个剪切事件.
- **splice junction:**多个跨越同一个内含子的剪切事件能够合并为一个 `splicing junction` .

junction 有三种:

1. 已经被注释的. `5'剪切位点` 和 `3'剪切位点` 已经被注释.
2. 全新的.
3. 部分是新的.

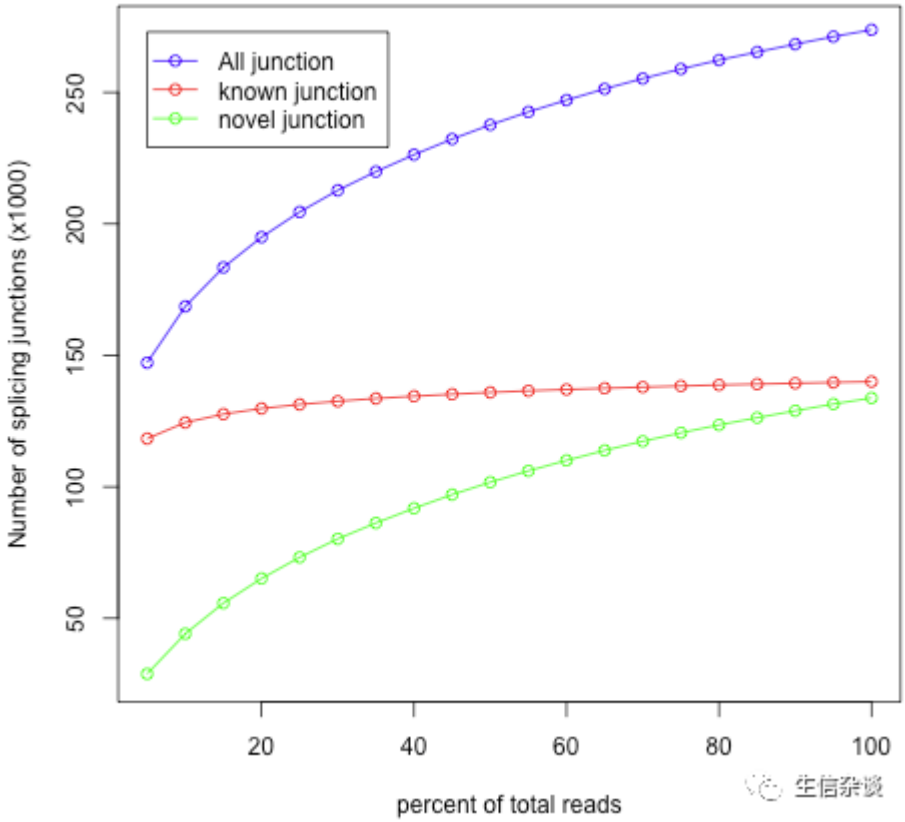
结果示例:



> **junction_saturation.py:**

在剪切位点分析时首先检查当前的**测序深度**是否是足够深的.对于一个已经注释的物种,在某个组织中基因的数量是一定的,所以剪切位点数量也是一定的.可以从参考基因模型(`bed12` 格式的文件)可以提前看出 `splice junctions` 的数量. 一个饱和的RNA-seq能够发现所有已经被注释的 `splice junctioons` ,否则下游剪切位点分析将会出现问题因为将有较少的 `splice juncitons` 将发现不了.这个模块从这个测序结果(`SAM` 或 `BAM` 文件)中进行重抽样,从5%,10%,15%,...,到95%的饱和度来检查每个阶段的 `splice junctions` 并与参考基因组进行比较.

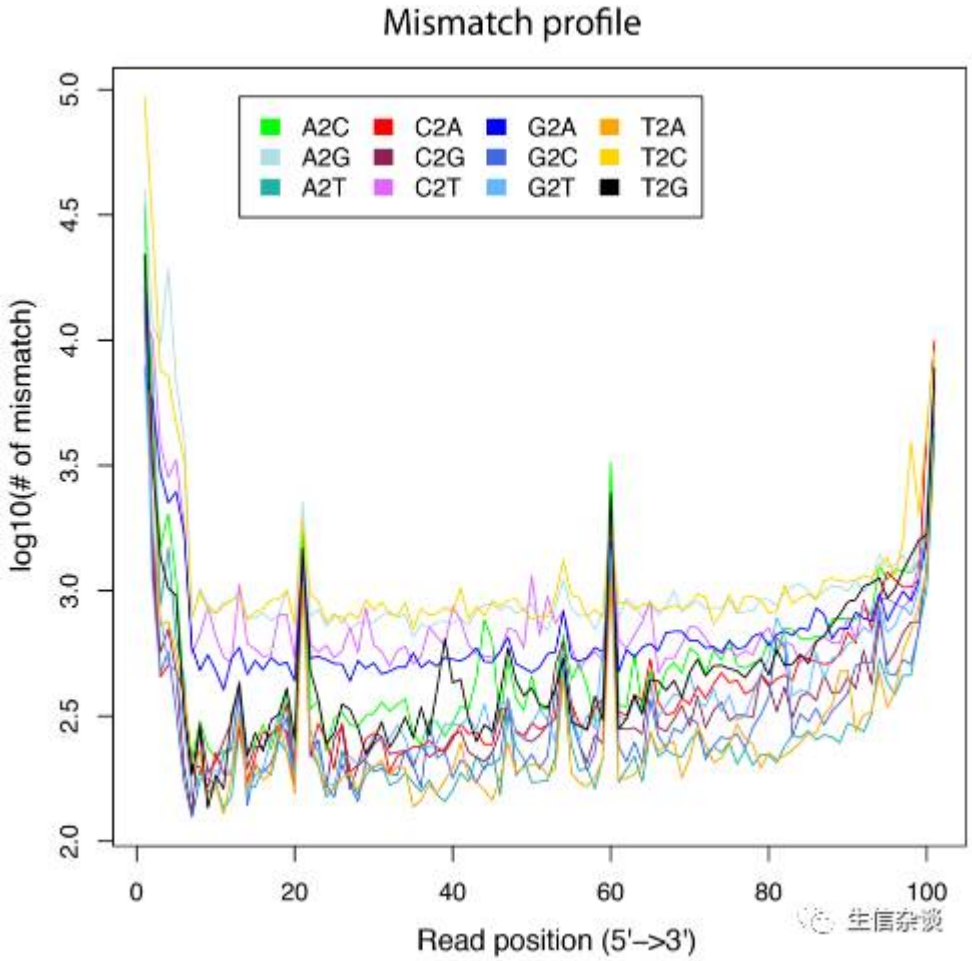
结果示例:



解释: 在这个例子中,每个饱和度下的测序深度的`known junctions`基本都是一致的(**红线**).因为大部分的剪切位点我们基本都发现了.即便加深测序深度也不会发现跟多的`known junctions`,仅会加深junciton的覆盖度(如junction被更多的reads覆盖.).然而当前测序深度(100%的reads)对于发现新的juncitons是不够的(**绿线**)

> **mismatch_profile.py:**
>
> 计算reads的不匹配位点的分布.

结果示例:



上图可以看出5'和3'的不匹配位点最多,这是由于测序本身所决定的.

> **normalize_bigwig.py:**

可视化RNA-seq数据结果是最直接并且高效的质控方式.但在可视化之前我们要保证所有样本的数据是可比较的,这就需要进行**归一化**.信号值文件'wig'或 `bigwig` 文件主要由两个因素决定:**(1)**总reads数,**(2)**read长度.因此,如果两个样本的read长度不一样但仅对"总reads数"归一化是有问题的.这里我们将每个 `bigwig` 文件归一化到相同的 `wigsum` 值. `wigsum` 是对基因组信号值的汇总,例如: `wigsum` =100,000,000等价于**1百万个100nt的reads或2百万个50nt的reads的覆盖度**. 结果生成 `wig` 格式的文件.

> **overlay_bigwig.py**

这个模块让我们操作两个 `bigwig` 文件.可以采取的操作有: **信号值相加,取均值,相除,每个信号值+1,求最大值,求最小值,相乘,相减,求几何平均数**.

> **read_distribution.py:**

这个模块根据提供的 `BAM/SAM` 文件和 `bed12格式的` gene模型文件就按比对上去的 `reads` 在基因组上的分布情况,比如在 `CDS exon` , `5'UTR exon` , `intro` , `基因间区域` 的 `reads` 分布.

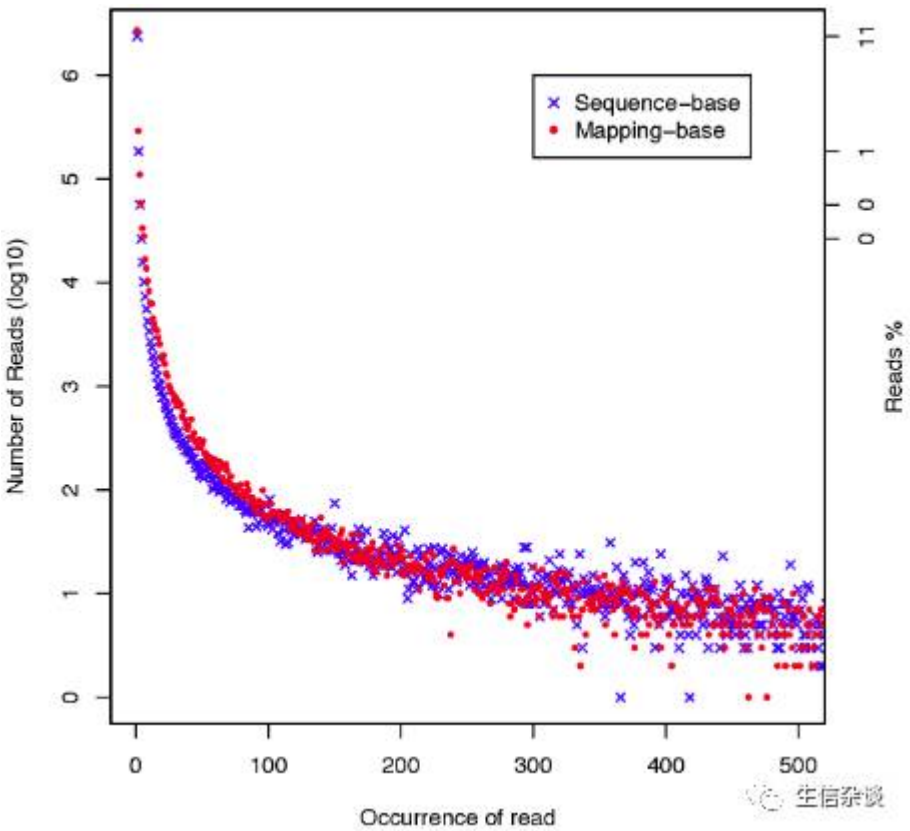
结果示例:

Group	Total_bases	Tag_count	Tags/Kb
CDS_Exons	33302033	20002271	600.63
5'UTR_Exons	21717577	4408991	203.01
3'UTR_Exons	15347845	3643326	237.38
Introns	1132597354	6325392	5.58
TSS_up_1kb	17957047	215331	11.99
TSS_up_5kb	81621382	392296	4.81
TSS_up_10kb	149730983	769231	5.14
TES_down_1kb	18298543	266161	14.55
TES_down_5kb	78900674	729997	9.25
TES_down_10kb	140361190	896882	3生信杂谈

---

read_duplication.py:

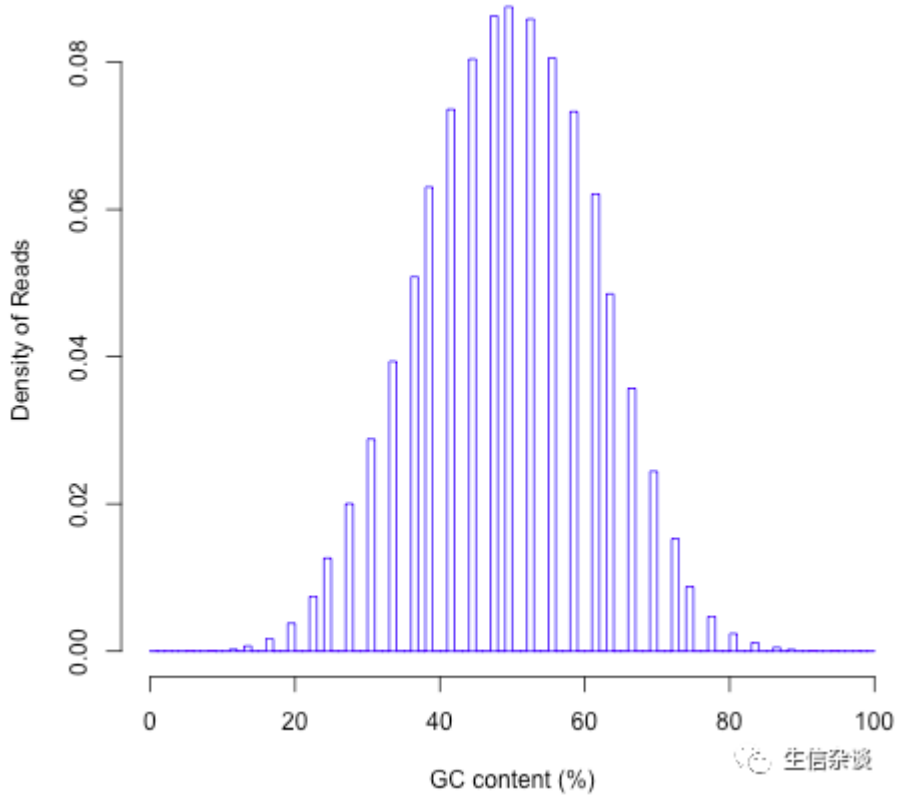
两种用于计算重复率的策略:**(1)** 基于序列的,完全相同序列的reads被视为重复的reads. **(2)** 正好map到同一个基因组位置的reads被视为重复reads. 对于 `splice reads` ,map到同一位置并且以相同方式剪切的视为是重复reads.



---

read_GC.py:

计算reads的GC含量分布.



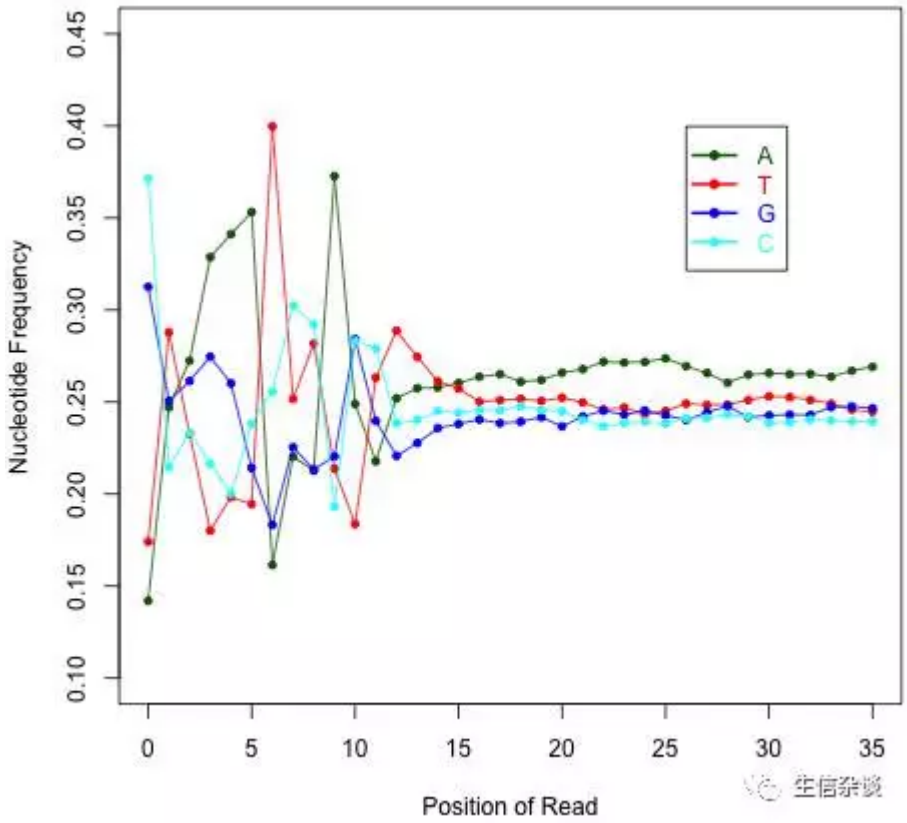
> **read_hexamer.py:**

计算6mer的频率.

> **read_NVC.py:**

这个模块有用来检查核苷酸的碱基组成偏好性.由于随机引物的影响, `reads 5'端` 开始会有某些模式过表达.这种偏好性能够被 `NVC` (Nucleotide versus cycle)画出.**理想状态下, A%=C%=G%=T%=25%**.

结果示例:

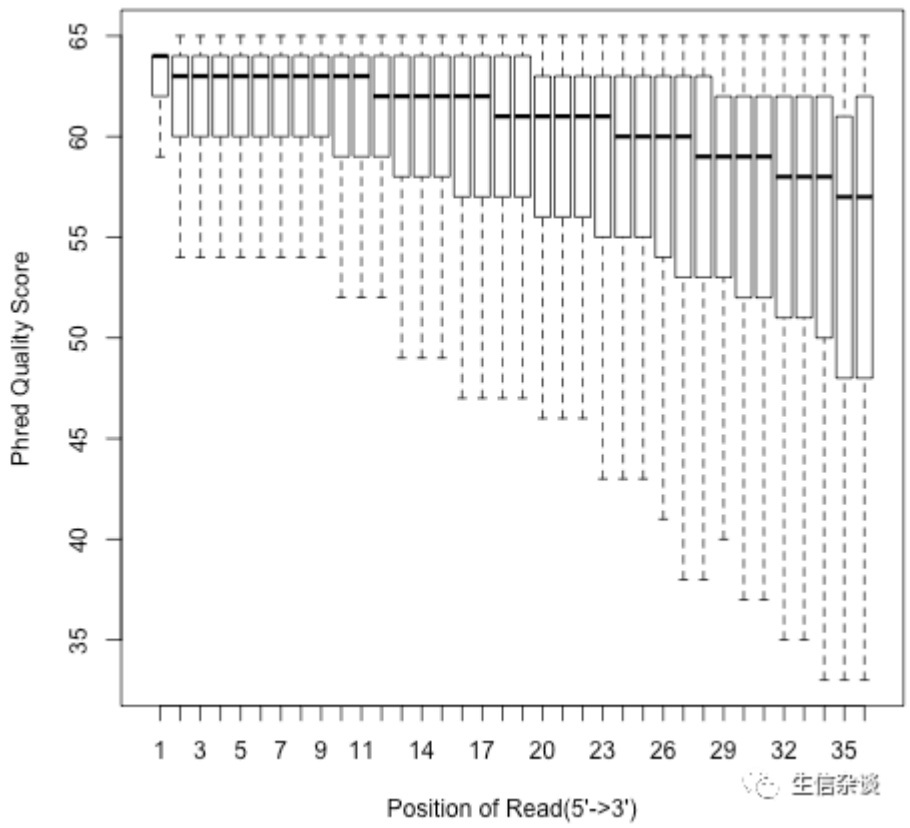


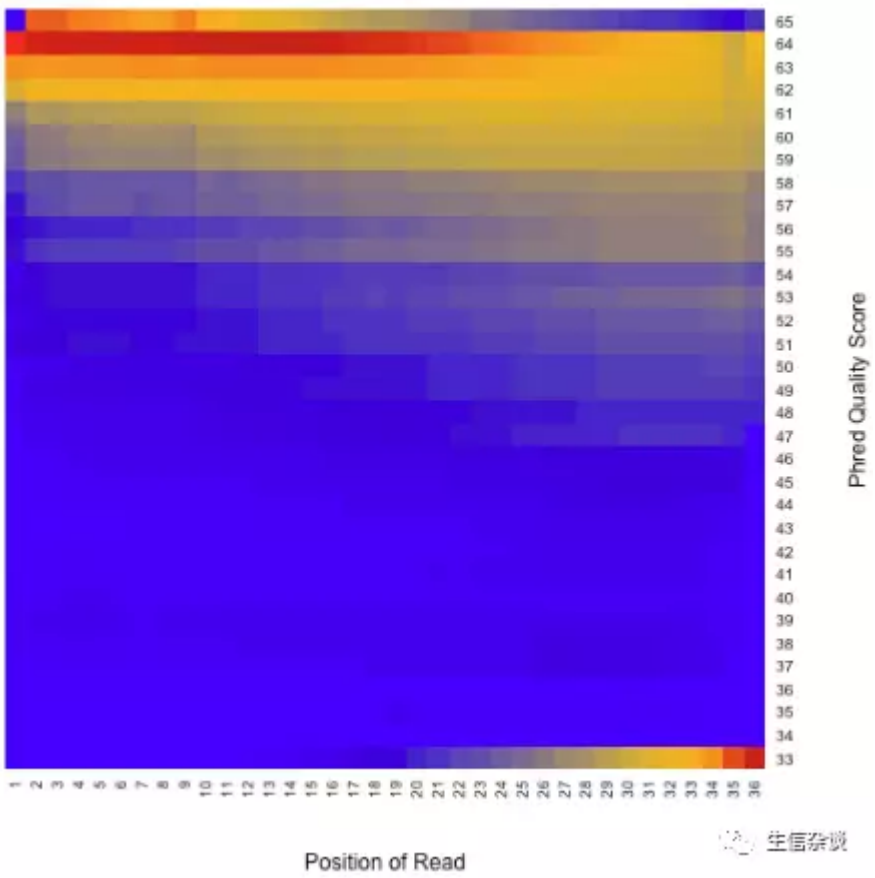
---

read_quality.py:

可视化 `reads` 每个位置的测序质量.

结果示例:





RNA_fragment_size.py:

在map后计算每个gene上的 `fragment` 的大小,包括:每个gene上所有的 `fragment` 的均值,中位数,方差.

结果示例:

chrom	tx_start	tx_end	symbol	frag_count	frag_mean	frag_median	frag_std
chr10	101542354	101611949	ABCC2	87	210.103448276	177.0	[illegible]
chr10	124768428	124817806	ACADSB	769	186.657997399	160.0	87.9515992717
chr10	114133915	114188138	ACSL5	122	183.475409836	157.5	85.1940132118

RPKM_count.py:

这个模块在最新版里已经被废弃,如果想使用可以翻看以前的版本.

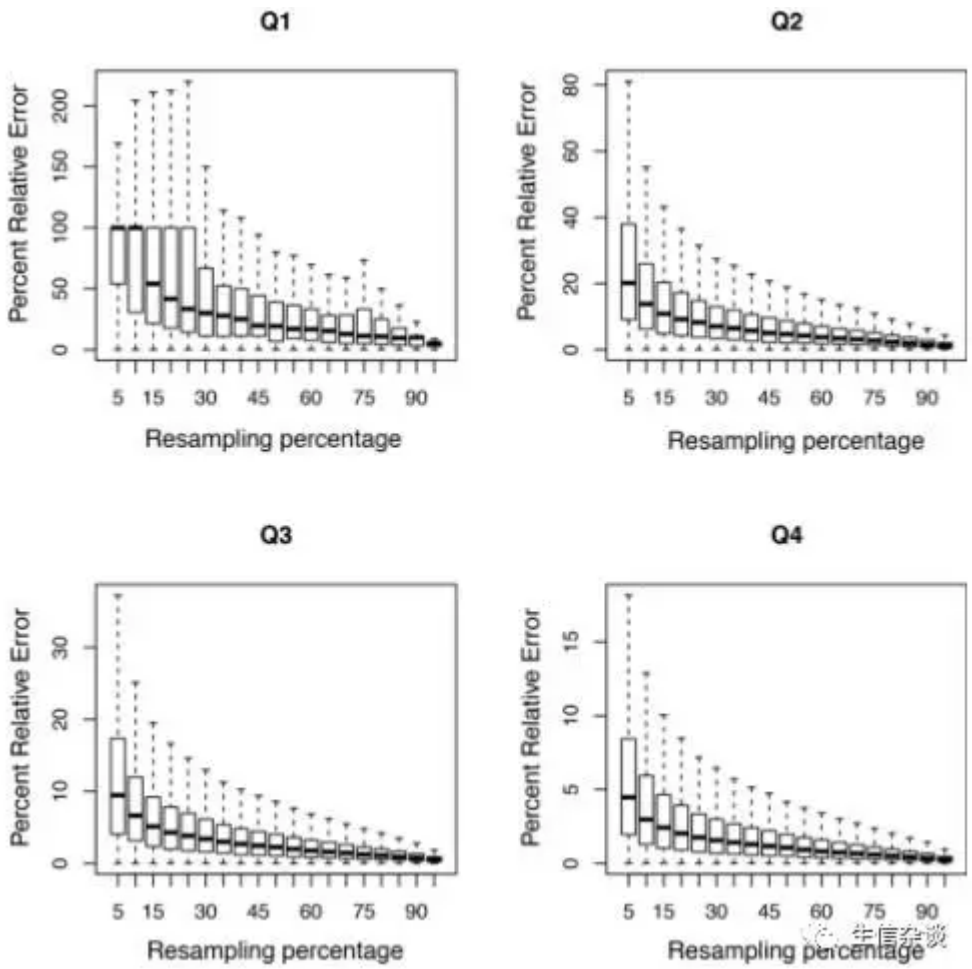
RPKM_saturation.py

任何样本统计（ `RPKM` ）的精度受样本大小（ `测序深度` ）的影响;**重抽样**或**切片**是使用部分数据来评估样本统计量的精度的方法. 这个模块从总的 `RNA reads` 中重抽样并计算每次的 `RPKM` 值. **通过这样我们就能检测当前测序深度是不是够的(如果测序深度不够RPKM的值将不稳定,如果测序深度足够则RPKM值将稳定).**默认情况下,这个模块将计算20个 `RPKM` 值(分别是对个转录本使用5%,10%,...,95%的总 `reads`).

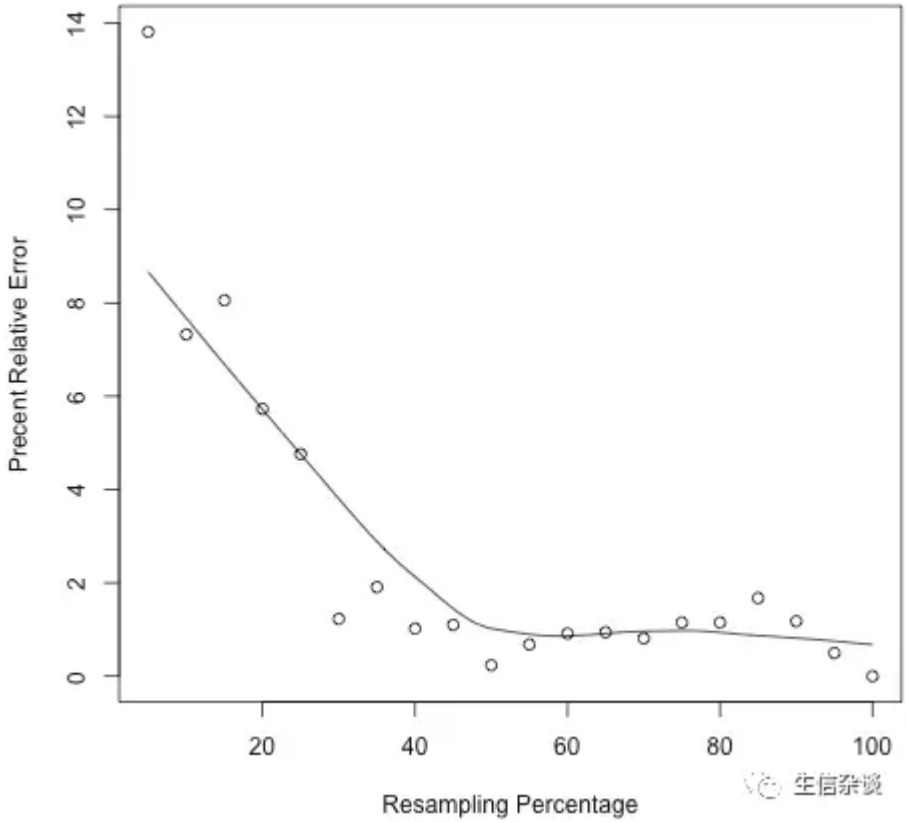
在结果图中,Y轴表示 `"Percent Relative Error"` 或 `"Percent Error"`.用来表示当前样本量下的 `RPKM` 与实际表达量的偏差.计算公式如下:

$$\text{Percent Relative Error} = \frac{|RPKM_{obs} - RPKM_{real}|}{RPKM_{real}} \times 100$$

结果示例:



说明:Q1,Q2,Q3,Q4是按照转录本表达量4分位分开的.Q1表示的是表达量低于25%的转录本,以此类推.**可以看出:**随着样本量升高, `RPKM` 与实际值的偏差也在降低.而且转录本表达量越高这种趋势越明显(Q4最明显).



可以看出,样本量50%之后线条已经趋于平缓,也就是说对于转录本定量来说,当前测序深度是足够的.

> **spilt_bam.py:**

根据提供的 `bed12` 注释文件和 `BAM` 文件拆分为以下三个文件:

1. **XXX.in.bam:** 包含map到外显子趋于的reads.
2. **XXX.ex.bam:**包含map不到外显子趋于的reads.
3. **XXX.junk.bam:**质控失败或者没有map上去的reads.

> **split_paired_bam.py:**

将一个**双端测序**的 `BAM` 文件拆分为两个**单端测序**的 `BAM` 文件.

> **tin.py:**

这个模块用来在转录本级别计算RNA完整性TIN (transcript integrity number)值.

结果示例:

geneID	chrom	tx_start	tx_end	TIN
ABCC2	chr10	101542354	101611949	67.6446525761
IPMK	chr10	59951277	60027694	86.3836184139
RUFY2	chr10	70100863	70167051	43.8967503948

每个模块的详细参数请点击左下角"阅读原文"查看官方文档.

更多原创精彩视频敬请关注**生信杂谈:**

阅读原文