[ChIPpeakAnno] [http://bioconductor.org/packages/release/bioc/html/ChIPpeakAnno.html]

1. Introduction

该包可用于发现富集峰最近的基因, 外显子, miRNA或定制的features(例如用户提供的保守单元或其他转录因子结合位点), 查询peak附近的序列, 获得富集的GO或通路.

ChIPpeakAnno一个重要功能就是根据已知的基因组特征注释peaks, 例如TSS, 5'UTR, 3'UTR等. 因此构建和选择合适的注释数据至关重要.

针对常见模式生物, 已经构建了一系列的转录起点注释信息, 例如TSS.human.NCBI36, TSS.human.GRCh37... 对于峰注释其他基因组信息, 可使用`getAnnotation`选择对应的featuretye, 'Exon'用于最近的外显子, 'miRNA'用于最近的miRNA, '5utr','3utr'来定位'5UTR','3UTR'的重叠.

此外, 针对自定义注释数据, 例如GRanges, 可用于`annotatePeakInBath`, 这里通过`toGRanges`函数将定义的注释数据转换为其他格式, 例如USCS BED/GFF格式. GRanges对象可通过`toGRanges`从EnsDB或TxDb对象构建而来.

而TxDb/EnsDB对象可通过GenomicFeature包从UCSC Genome Bioinformatics/BioMart下载, 或使用`makeTxDbFromGRanges`/`makeTxDbFromGFF`创建.

2. Quick start

```
library(ChIPpeakAnno)
```

```
macs <- system.file("extdata","MACS_peaks.xls",package="ChIPpeakAnno")
```

```
macsOutput <- toGRanges(macs, format="MACS")
```

使用ensembl 注释

```
data(TSS.human.GRCh38)
```

```
macs.anno <- annotatePeakInBatch(macsOutput, AnnotationData=TSS.human.GRCh.38)
```

加入基因symbol

```
library(org.Hs.eg.db)
```

```
macs.anno <- addGeneIDs(annotatedPeak=macs.anno, orgAnn="org.Hs.eg.db",
IDs2Add="symbol")
```

3. An examle of ChIP-seq analysis workflow using ChIPpeakAnno

输入为一系列来自ChIP-seq实验识别的峰. 在ChIPpeakAnno中, 峰是以GRanges的格式表示的. 使用函数`toGRanges`将峰文件格式, 例如BED, GFF或MACS格式转换为Granges.

该流程用于将BED/GFF格式转换为GRanges, 然后在两组峰中查询重叠的峰, 使用Venn图查看.

读取峰文件

`bed <- system.file("extdata","MACS_output.bed",package="ChIPpeakAnno")`

`gr1 <- toGRanges(bed, format="BED",header=FALSE)`

也可使用`rtracklayer`包的`import`函数转换格式为GRanges

`library(rtracklayer)`

`gr1.import <- import(bed, format="BED")`

`identical(start(gr1), start(gr1.import))`

`gff <- system.file("extdata","GFF_peaks.gff",package="ChIPpeakAnno")`

`gr2 <- toGRanges(gff, format="GFF",header=FALSE,skip=3)`

```
> gr2 <- toGRanges(gff, format="GFF",header=FALSE,skip=3)
If you are importing files downloaded from ensembl,
         it will be better to import the files into a TxDb object,
         and then convert to GRanges by toGRanges. Here is the sample code:
         library(GenomicFeatures)
         txdb <- makeTxDbFromGFF('/Library/Frameworks/R.framework/Versions/3.6/Resources/library/ChIPpeakAnn
o/extdata/GFF_peaks.gff')
         anno <- toGRanges(txdb, format='gene')
```

针对GFF文件, 建议先导入为TxDb对象, 再使用toGRanges转换

查询重叠区域, 绘制文式图和饼图

`ol <- findOverlapsOfPeaks(gr1, gr2)`

`makeVenDiagram(ol, fill=c("#009E73","#F0E442"), col=c("#D55E00","#0072B2"), cat.col=c("#D55E00","#0072B2"))`

`pie1(table(ol$overlappingPeaks[["gr1///gr2"]]$overlapFeatures))`

`findOverlapsOfPeaks`返回7个值的列表

`venn_cnt` VennCounts对象

`peaklist` 包含重叠峰或独立峰的列表

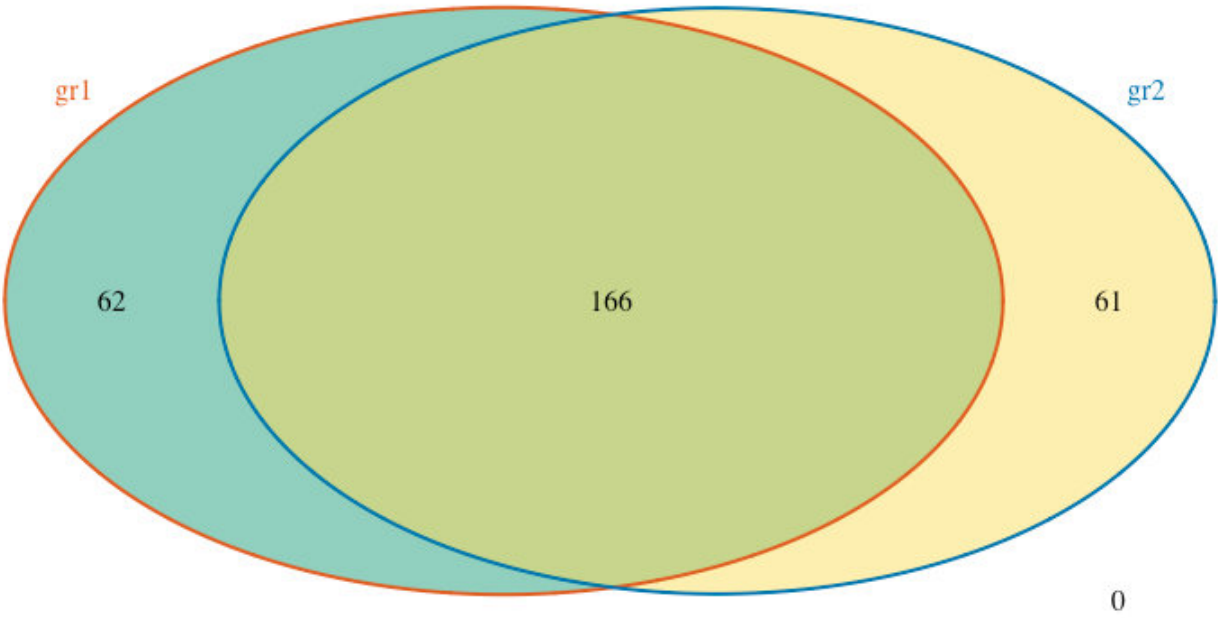
`uniquePeaks` 包含所有独立峰的GRanges对象

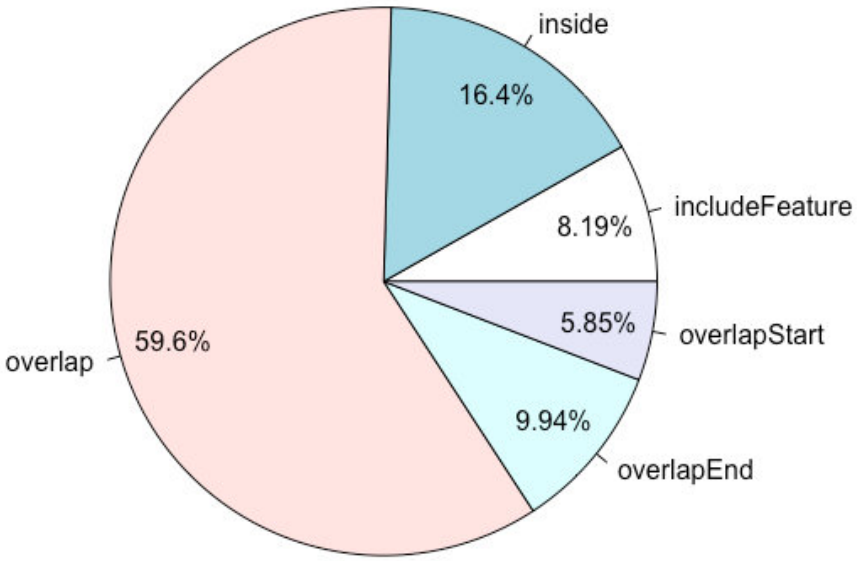
`mergedPeaks` 包含所有合并了的重叠峰的GRanges对象

`peaksInMergedPeaks` 包含每个样本中涉及到重叠峰的GRanges对象

`overlappingPeaks` 包含所有重叠峰的注释的数据框

`all.peaks` 所有输入峰的GRanges对象





查询到重叠峰后, 根据AnnotationData中的基因组信息, 使用`annotatePeakInBatch`注释重叠的峰其5000bp内的特征信息, with certain distance away specified by maxgap, which is 5kb in the following example.

```
overlaps <- ol$peaklist[["gr1///gr2"]]
```

```
library(EnsDb.Hsapiens.v75)
```

使用EnsDb/TxDb构建注释文件

```
annoData <- toGRanges(EnsDb.Hsapiens.v75, feature="gene")
```

```
overlaps.anno <- annotatePeakInBatch(overlaps, AnnotationData=annoData,
output="overlapping", maxgap=5000L)
```

`overlaps.anno$gene_name <- annoData$gene_name[match(overlaps.anno$feature, names(annoData))]`

`maxgap`为最大两峰之间对gap距离为5000bp的注释

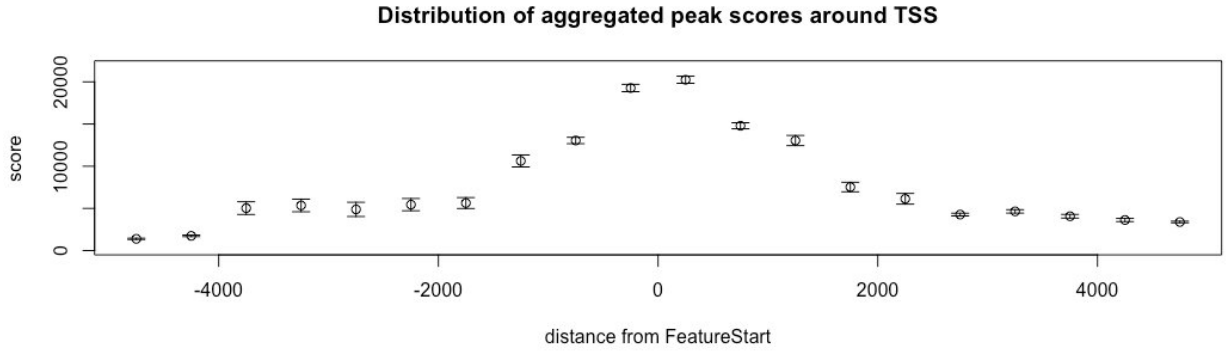
```
> head(overlaps.anno)
GRanges object with 6 ranges and 11 metadata columns:
                     seqnames        ranges strand |                      peakNames        peak
                        <Rle>     <IRanges>  <Rle> |                    <CharacterList> <character>
  X001.ENSG00000228327     chr1 713791-715578      * | gr1__MACS_peak_13,gr2__001,gr2__002         001
  X001.ENSG00000237491     chr1 713791-715578      * | gr1__MACS_peak_13,gr2__001,gr2__002         001
  X002.ENSG00000237491     chr1 724851-727191      * |          gr2__003,gr1__MACS_peak_14         002
  X003.ENSG00000272438     chr1 839467-840090      * |          gr1__MACS_peak_16,gr2__004         003
  X004.ENSG00000223764     chr1 856361-856999      * |          gr1__MACS_peak_17,gr2__005         004
  X004.ENSG00000187634     chr1 856361-856999      * |          gr1__MACS_peak_17,gr2__005         004
                         feature start_position end_position feature_strand insideFeature
                     <character>      <integer>    <integer>    <character>      <factor>
  X001.ENSG00000228327 ENSG00000228327         700237       714006             -  overlapStart
  X001.ENSG00000237491 ENSG00000237491         714150       745440             +  overlapStart
  X002.ENSG00000237491 ENSG00000237491         714150       745440             +         inside
  X003.ENSG00000272438 ENSG00000272438         840214       851356             +       upstream
  X004.ENSG00000223764 ENSG00000223764         852245       856396             -  overlapStart
  X004.ENSG00000187634 ENSG00000187634         860260       879955             +       upstream
                     distancetoFeature shortestDistance fromOverlappingOrNearest     gene_name
                             <numeric>       <integer>              <character>   <character>
  X001.ENSG00000228327             215             215              Overlapping RP11-206L10.2
  X001.ENSG00000237491            -359             359              Overlapping RP11-206L10.9
  X002.ENSG00000237491           10701           10701              Overlapping RP11-206L10.9
  X003.ENSG00000272438            -747             124              Overlapping  RP11-5407.16
  X004.ENSG00000223764              35              35              Overlapping   RP11-5407.3
  X004.ENSG00000187634           -3899            3261              Overlapping        SAMD11
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

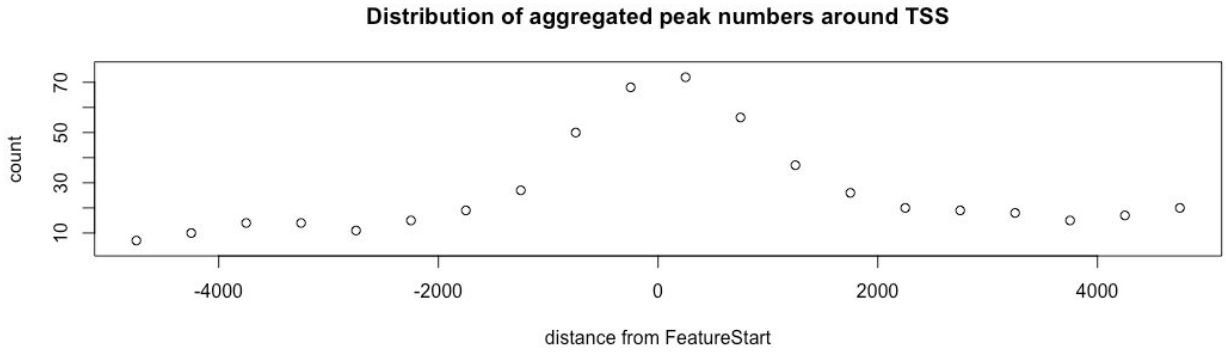
完成峰的注释后, **距离最近的基因组特征信息, 例如转录起始点(TSS)可绘制**

`gr1.copy <- gr1`

`gr1.copy$score <- 1`

`binOverFeature(gr1, gr1.copy, annotationData=annoData, radius=5000, nbins=10, FUN=c(sum, length), ylab=c("score","sum"),main=c("Distribution of aggregated peak score around TSS", "Distribution of aggregated peak numbers around TSS'))`

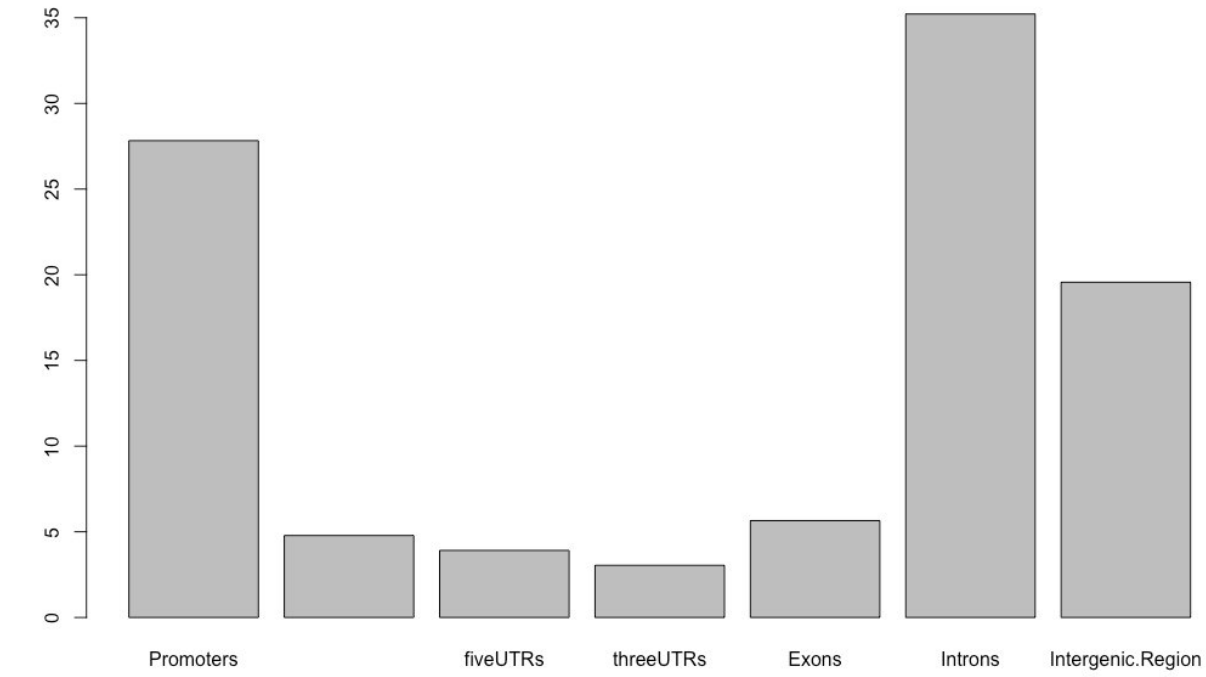




绘制峰跨越外显子, 内含子, 增强子(enhancer), proximal promoter, 5' UTR, 3' UTR的分布图

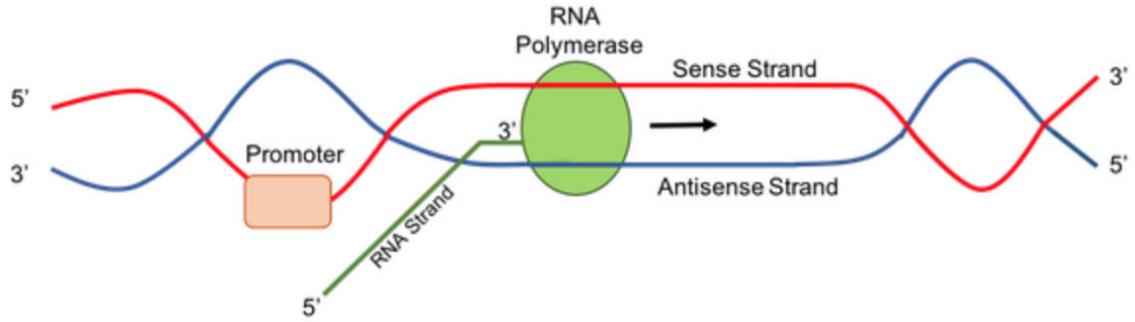
```
if(require(TxDb.Hsapiens.UCSC.hg19.knownGene)){aCR <- assignChromosomeRegion(gr1,
nucleotideLevel=FALSE,
precedence=c("Promoters","immediateDownstream","fiveUTRs","threeUTRs","Exons","Int
rons"),TxDb=TxDb.Hsapiens.UCSC.hg19.knownGene)

barplot(aCR$percentage)}
```



4. Detailed Use Cases and Scenarios

获得峰周围序列




```
library(BSgenome.H10407.NCBI.01)
```

```
seq <- getAllPeakSequence(overlaps, upstream=20,downstream=20,genome=Hsapiens)
```

```
write2FASTA(seq, "test.fa")
```

需要注意的是, `overlaps` 中的strand不包含方向:

```
GRanges object with 312 ranges and 6 metadata columns:
                 seqnames      ranges strand |    length abs_summit    pileup -log10(pvalue) fold_enrichment
                    <Rle>   <IRanges>  <Rle> | <integer>  <integer> <numeric>      <numeric>       <numeric>
   H_NS_peak_1 NC_017633.1 20000-20200      * |       201      20068     11.58        3.71098         2.89335
   H_NS_peak_2 NC_017633.1 21125-21430      * |       306      21311     20.46        9.75584         4.78185
   H_NS_peak_3 NC_017633.1 21931-22306      * |       376      22135      19.3        9.68985         4.87758
   H_NS_peak_4 NC_017633.1 22536-22809      * |       274      22775     18.91        7.23803         3.95914
   H_NS_peak_5 NC_017633.1 23001-23331      * |       331      23061     21.23        7.73977         3.70509
```

因此可以先简单修改strand值, 在获得序列(或使用 `getSeq`):

```
strand(h_ns_macs2_anno) <- h_ns_macs2_anno$feature_strand
```

```
seq1 <-
getAllPeakSequence(h_ns_macs2_anno,genome=BSgenome.H10407.NCBI.01,upstream =
50,downstream = 50)
```

or

```
> getSeq(BSgenome.H10407.NCBI.01,h_ns_macs2_anno)
  A DNAStringSet instance of length 313
      width seq                                               names
  [1]   201 GGAACTCCTTTTCCTTTGTTGTTCG...CAATAATGAAAATTATCAGTTTCAT H_NS_peak_1.ETEC_...
  [2]   306 ACTGGGGGGATGATATTGCCAAAAC...GTTGTCTGGTCCGGTGGTATTAATA H_NS_peak_2.ETEC_...
  [3]   376 TGCACGATTACGACAAAACATCCCC...TTTTATCCGATAACTATAATGCTAT H_NS_peak_3.ETEC_...
  [4]   274 ATTTATACTTTTATTACCCTATAAT...AAGCAACATATTTGGAGCATCATTC H_NS_peak_4.ETEC_...
  [5]   331 ATTGGATGTTTTTGATGACAGAACA...TAAAATCCCCTCCTCATCGAAAGAT H_NS_peak_5.ETEC_...
  ...   ... ...
```

最后输出

```
write2FASTA(seq1, "test1.fa")
```

```
writeXStringSet(seq2, "test2.fa")
```

Mischellaneous

- makeTxDbFromGFF

`library(GenomicFeatures)`

`hs11286_txdb <- makeTxDbFromGFF("GCF_000240185.1_ASM24018v2_genomic.gff",organism="Klebsiella_pneumonia_hs11286",taxonomyId = 573,dataSource="NCBI Klebseilla pneumonia HS11286 gff file",dbxrefTag = "locus_tag",circ_seqs = c("NC_016838.1","NC_016839.1","NC_016840.1","NC_016841.1","NC_016845.1","NC_016846.1","NC_016847.1"))`

`select(hs11286_txdb,keys=keys(hs11286_txdb),columns=columns(hs11286_txdb),keytype = "GENEID")`

- Forge a BSgenome Data

1. 来源数据文件:1) 包含序列的文件; 2) 包含mask数据的文件(可选)

序列数据必须为单个twoBit文件(e.g. musFur1.2bit)或者为FASTA文件汇总(可能为gzip压缩). 假如是后者, 需满足一个序列一个fasta文件, 同时每个fasta文件的名称必须为`<prefix><seqname><suffix>`格式, `<seqname>`为该文件中的序列名称, 并且所有fasta文件中的prefix和suffix(可选)要求相同.

同时可以使用`Biostrings`包中的`fasta.seqlengths`函数来获得fasta文件的长度

`library(Biostrings)`

`fasta.seqlengths(file)`

```
Package: BSgenome.Rnorvegicus.UCSC.rn4
Title: Full genome sequences for Rattus norvegicus (UCSC version rn4)
Description: Full genome sequences for Rattus norvegicus (Rat) as provided by UCSC (rn4, Nov. 2004) an
Version: 1.4.2
Suggests: TxDb.Rnorvegicus.UCSC.rn4.ensGene
organism: Rattus norvegicus
common_name: Rat
provider: UCSC
provider_version: rn4
release_date: Nov. 2004
release_name: Baylor College of Medicine HGSC v3.4
source_url: http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/
organism_biocview: Rattus_norvegicus
BSgenomeObjname: Rnorvegicus
seqnames: paste("chr", c(1:20, "X", "M", "Un", paste(c(1:20, "X", "Un"), "_random", sep="")), sep="")
circ_seqs: "chrM"
SrcDataFiles: chromFa.tar.gz from http://hgdownload.cse.ucsc.edu/goldenPath/rn4/bigZips/
PkgExamples: genome$chr1  # same as genome[["chr1"]]
        .
        ## ------------------------------------------------------------------------
        ## Upstream sequences
        ## ------------------------------------------------------------------------
        ## Starting with BioC 3.0, the upstream1000, upstream2000, and
        ## upstream5000 sequences for rn4 are not included in the BSgenome data
        ## package anymore. However they can easily be extracted from the full
        ## genome sequences with something like:
```

2. `BSgenome`数据包seed文件包含了`forgeBSgenomeDataPkg`函数构建目的包的所有信息, 该seed文件格式为DCF(Debian Control File), 同时也是用来DESCRIPTION任何R包的文件格式. seed文件包含3个有效的分类域:

- Standard DESCRIPTION fields, 为任何DESCRIPTION文件中必须包含的内容, 将会直接复制到目的包中:
    - Package, 目的包名称, 一般名称有点分开的4部分, BSgenome.abbreviated_name_organism.organisation_provided_genome.release_string

_number_version
    - Title, 目的包的title, e.g. Full genome sequences for Rattus norvegicus(UCSC version rn4)
    - Description, Version, Author, Maintainer, License, 和前两个一样, 为固定必须内容
    - Suggests, [OPTIONAL], 例如给出例子
- Non-standard DESCRIPTION fields, 为seed文件特异性的fields, 也将复制到到目的包中, 此外, 这些fields的值也将被包含在目的包中
    - organism, 物种Genus species subspecies的科学名称, e.g. Home sapiens neanderthalensis
    - common_name, 物种的通用名称, e.g. Rat或Human
    - provider, 序列数据的提供者, e.g. UCSC, NCBI, BDGP, FlyBase
    - provider_version, 基因组的provider-side 版本
    - release_date, 基因组公布日期
    - release_name, 基因组公布的名称后构建数目
    - source_url, 测序数据文件永久的URL
    - organism_biocview, 该物种的官方biocViews项目
- Additional fields, don't fall in the first 2 categories
    - BSgenomeObjname, 应匹配package名称的第二部分内容
    - seqnames, [OPTIONAL]序列名称, 假如使用序列数据文件的汇总, 此时应为用于构建的单个序列名称. e.g. `paste("chr",c(1:20),"X","M","Un",paste(c(1:20),"X","Un"),"_random",sep=""),sep="")`
    - circ_seqs, [OPTIONAL]为环状序列的名称, 同上, 默认为NULL
    - ...

3. forge the targe package

`forgeBSgenomeDataPkg`函数根据`seed`文件构建BSgenome包

构建完成后, 忽略所有的warnings, 退出R, 构建源包的(tarball)

```
R CMD build <pkgdir>
```

`<pkgdir>` is the path to the source tree of the package

然后检查构建好的包

```
R CMD check <tarball>
```

`<tarball>`为R CMD build构建的tarball路径(tarball, 压缩包)

最后安装该包

```
R CMD INSTALL <tarball>
```

4. forge a BSgenome data package with masked sequences

BSgenomeForge当前支持4种 built-in masks

- the masks of assembly gaps, aka "the AGAPS masks"
- the masks of intra-contig ambiguities, aka "the AMB masks"
- the masks of repeat regions that were determine by the RepeatMasker software, aka "the RM masks"
- the masks of repeat regions that were determined by the Tandem Repeats Finder

software(where only repeats with period less than or equal to 12 were kept), aka "the TRF masks"

对于AGAPS masks, 需要UCSC 'gap' or NCBI 'agp'文件. 每条染色体一个文件或单个大文件包含所有染色体的组装gap信息...

对于AMB masks, 无需任何额外的文件

对于RM masks, 需要RepeatMasker .out文件, 同AGAPS masks, 可以是一个染色体一个文件或单个文件包含所有染色体的RepeatMasker信息, 对于前者文件名称需为`<prefix><seqname><suffix>`

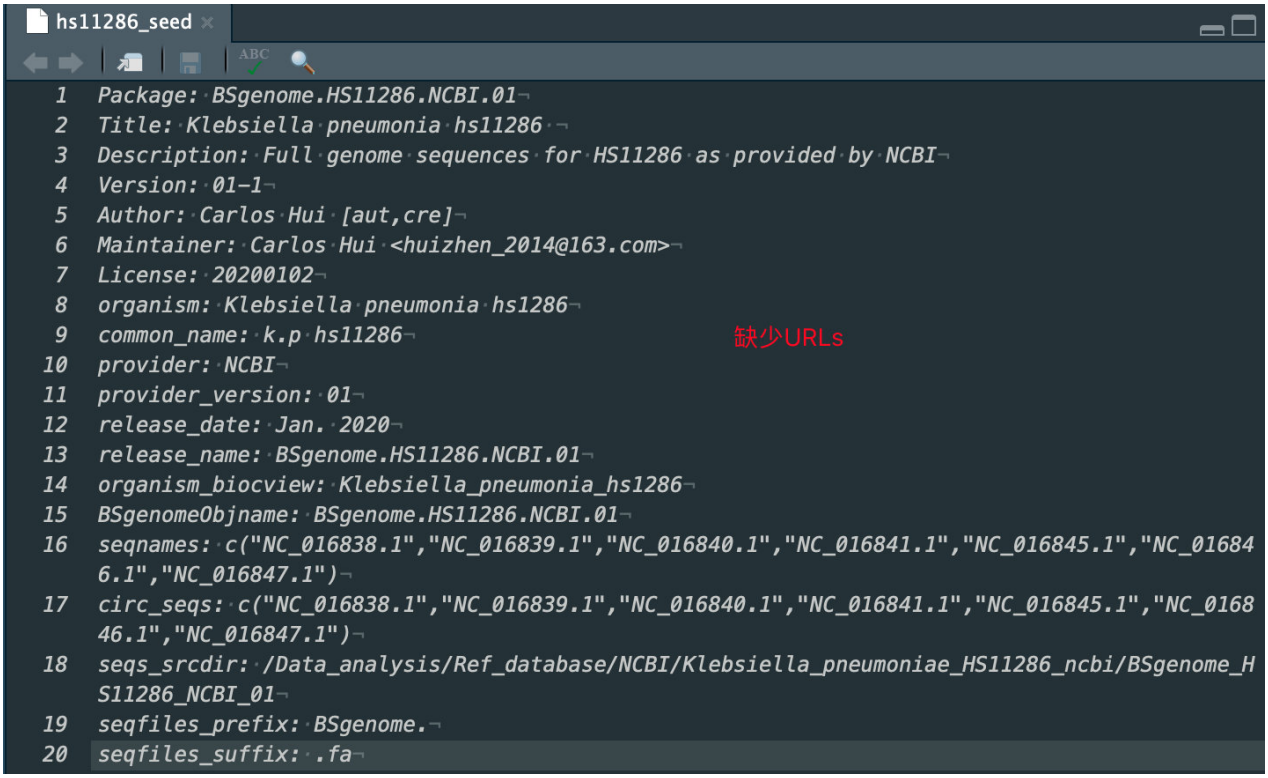
对于TRF masks, 需要Tandem Repeats Finder .bed文件. 同样可以是一个染色体一个文件或者单个大的文件. 对于前者文件名称需为`<prefix><seqname><suffix>`

seed文件(the masked BSgenome data package, 2nd targe package)和包含纯序列文件的BSgenome数据包的seed文件类似. 包含所有用于`forgeMaskedBSgenomeDataPkg`函数构建2nd target package的信息

The DESCRIPTION file contains basic information about the package in the following format:

```
Package: pkgname
Version: 0.5-1
Date: 2015-01-01
Title: My First Collection of Functions
Authors@R: c(person("Joe", "Developer", role = c("aut", "cre"),
                    email = "Joe.Developer@some.domain.net"),
             person("Pat", "Developer", role = "aut"),
             person("A.", "User", role = "ctb",
                    email = "A.User@whereever.net"))
Author: Joe Developer [aut, cre],
  Pat Developer [aut],
  A. User [ctb]
Maintainer: Joe Developer <Joe.Developer@some.domain.net>
Depends: R (>= 3.1.0), nlme
Suggests: MASS
Description: A (one paragraph) description of what
  the package does and why it may be useful.
License: GPL (>= 2)
URL: https://www.r-project.org, http://www.another.url
BugReports: https://pkgname.bugtracker.url
```

- Standard DESCRIPTION fileds
  - Package, 2nd target pacakge的名称, 推荐使用同reference target package名称, 后缀为.masked
  - Title, 包的title. e.g. Full masked genome sequences for Rattus norvegicus(UCSC version rn4)
  - Description, Version, Author, Maintainer, License
- Non-standard DESCRIPTION fields
  - organism_biocview, 同reference target
  - source_url, 用于构建该对象的永久性的mask数据文件URL
- Other fields
  - RefPkgname, 参考target包的名称
  - nmask_per_seq, 每个序列mask的数目(1到4)
  - PkgDetails, PkgExamples, 和之前一样
  - ...



或直接通过从NCBI注释信息提取的注释data.frame, 手动构建用于ChIPpeakAnno的GRange文件