Package 'CODEX'

November 5, 2026

Type Package

Title A Normalization and Copy Number Variation Detection Method for Whole Exome Sequencing

Version 1.45.0

Author Yuchao Jiang, Nancy R. Zhang

Maintainer Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Description A normalization and copy number variation calling procedure for whole exome DNA sequencing data. CODEX relies on the availability of multiple samples processed using the same sequencing pipeline for normalization, and does not require matched controls. The normalization model in CODEX includes terms that specifically remove biases due to GC content, exon length and targeting and amplification efficiency, and latent systemic artifacts. CODEX also includes a Poisson likelihood-based recursive segmentation procedure that explicitly models the count-based exome sequencing data.

License GPL-2

Depends R (>= 3.2.3), Rsamtools, GenomeInfoDb, BSgenome.Hsapiens.UCSC.hg19, IRanges, Biostrings, S4Vectors

Suggests WES.1KG.WUGSC

biocViews ImmunoOncology, ExomeSeq, Normalization, QualityControl, CopyNumberVariation

LazyData yes

git_url https://git.bioconductor.org/packages/CODEX

git_branch devel

git_last_commit 892afc3

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-11-05

Contents

CODEX-package 2
bambedObjDemo 3
choiceofK 3

coverageObjDemo 4
gcDemo 5
getbambed 5
getcoverage 6
getgc 7
getmapp 8
mappability 9
mappDemo 10
mapp_ref 10
normalize 11
normalize2 12
normObjDemo 13
qc 14
qcObjDemo 15
segment 16

Index **18**

`CODEX-package` *A Normalization and Copy Number Variation Detection Method for Whole Exome Sequencing*

Description

CODEX is a normalization and copy number variation calling procedure for whole exome DNA sequencing data. CODEX relies on the availability of multiple samples processed using the same sequencing pipeline for normalization, and does not require matched controls. The normalization model in CODEX includes terms that specifically remove biases due to GC content, exon length and targeting and amplification efficiency, and latent systemic artifacts. CODEX also includes a Poisson likelihood-based recursive segmentation procedure that explicitly models the count-based exome sequencing data.

Details

Package:	CODEX
Type:	Package
Version:	0.99.0
Date:	2015-01-13
License:	GPL-2

CODEX takes as input the bam files/directories for whole exome sequencing datasets and bed files for exonic positions, returns raw and normalized coverage for each exon, and calls copy number variations with genotyping results.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>, Nancy R. Zhang

bambedObjDemo *Demo data pre-stored for bambedObj.*

Description

Pre-stored bambedObj data for demonstration purposes.

Usage

```
data(bambedObjDemo)
```

Details

Pre-computed using whole exome sequencing data of 46 HapMap samples.

Value

bambedObj demo data (list) pre-computed.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
bamdir <- bambedObjDemo$bamdir
sampname <- bambedObjDemo$sampname
ref <- bambedObjDemo$ref
projectname <- bambedObjDemo$projectname
chr <- bambedObjDemo$chr
```

choiceofK *Determine the number of latent factors K.*

Description

Determines the number of latent variables K via AIC, BIC, and deviance reduction. A pdf file containing all three plots is generated.

Usage

```
choiceofK(AIC, BIC, RSS, K, filename)
```

Arguments

AIC	vector of AIC for each K returned from normalize
BIC	vector of BIC for each K returned from normalize
RSS	vector of RSS for each K returned from normalize
K	vector of K returned from normalize
filename	Filename of the output plot of AIC and RSS

Details

AIC: Akaike information criterion, used for model selection; BIC: Bayesian information criterion, used for model selection; RSS: residue sum of squares, used to plot scree plot and determine the 'elbow'.

Value

pdf file with three plots: AIC, BIC, and percentage of variance explained versus the number of latent factors.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

normalize, segment

Examples

```
AIC <- normObjDemo$AIC
BIC <- normObjDemo$BIC
RSS <- normObjDemo$RSS
K <- normObjDemo$K
projectname <- bambedObjDemo$projectname
chr <- bambedObjDemo$chr
choiceofK(AIC, BIC, RSS, K, filename = paste(projectname, "_", chr,
    "_choiceofK", ".pdf", sep = ""))
```

---

coverageObjDemo *Demo data pre-stored for coverageObj.*

Description

Pre-stored coverageObj data for demonstration purposes.

Usage

```
data(coverageObjDemo)
```

Details

Pre-computed using whole exome sequencing data of 46 HapMap samples.

Value

coverageObj demo data (list) pre-computed.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
Y <- coverageObjDemo$Y
readlength <- coverageObjDemo$readlength
```

---

gcDemo *Demo data pre-stored for GC content.*

Description

Pre-stored GC content data for demonstration purposes.

Usage

```
data(gcDemo)
```

Details

Pre-computed using whole exome sequencing data of 46 HapMap samples.

Value

gc demo data (vector) pre-computed.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
head(round(gcDemo, 2))
```

---

getbambed *Get bam file directories, sample names, and exonic positions*

Description

Gets bam file directories, sample names from .txt file, and exonic positions from .bed file.

Usage

```
getbambed(bamdir,bedFile,sampname,projectname,chr)
```

Arguments

bamdir	Column vector. Each line specifies directory of a bam file. Should be in same order as sample names in sampname.
bedFile	Path to bed file specifying exonic targets. Is of type character.
sampname	Column vector. Each line specifies name of a sample corresponding to the bam file. Should be in same order as bam directories in bamdir.
projectname	String specifying the name of the project. Data will be saved using this as prefix.
chr	Chromosome.

Value

bamdir	Bam directories
sampname	Sample names
ref	IRanges object specifying exonic positions
projectname	String specifying the name of the project.
chr	Chromosome

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

References

Lawrence M, Huber W, Pages H, Aboyoun P, Carlson M, Gentleman R, Morgan M and Carey V (2013). "Software for Computing and Annotating Genomic Ranges." PLoS Computational Biology, 9.

See Also

getcoverage

Examples

```
library(WES.1KG.WUGSC)
dirPath <- system.file("extdata", package = "WES.1KG.WUGSC")
bamFile <- list.files(dirPath, pattern = '*.bam$')
bamdir <- file.path(dirPath, bamFile)
sampnameFile <- file.path(dirPath, "sampname")
sampname <- as.matrix(read.table(sampnameFile))
chr <- 22
bambedObj <- getbambed(bamdir = bamdir, bedFile = file.path(dirPath,
    "chr22_400_to_500.bed"), sampname = sampname,
    projectname = "CODEX_demo", chr)
bamdir <- bambedObj$bamdir
sampname <- bambedObj$sampname
ref <- bambedObj$ref
projectname <- bambedObj$projectname
chr <- bambedObj$chr
```

---

getcoverage *Get depth of coverage from whole exome sequencing*

Description

Gets depth of coverage for each exon across all samples from whole exome sequencing files.

Usage

```
getcoverage(bambedObj, mapqthres)
```

Arguments

bambedObj	Object returned from getbambed
mapqthres	Mapping quality threshold hold of reads.

Value

Y	Read depth matrix
readlength	Vector of read length for each sample

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

getbambed

Examples

```
library(WES.1KG.WUGSC)
dirPath <- system.file("extdata", package = "WES.1KG.WUGSC")
bamFile <- list.files(dirPath, pattern = '*.bam$')
bamdir <- file.path(dirPath, bamFile)
sampnameFile <- file.path(dirPath, "sampname")
sampname <- as.matrix(read.table(sampnameFile))
chr <- 22
bambedObj <- getbambed(bamdir = bamdir, bedFile = file.path(dirPath,
    "chr22_400_to_500.bed"), sampname = sampname,
    projectname = "CODEX_demo", chr)
bamdir <- bambedObj$bamdir
sampname <- bambedObj$sampname
ref <- bambedObj$ref
projectname <- bambedObj$projectname
chr <- bambedObj$chr
coverageObj <- getcoverage(bambedObj, mapqthres = 20)
Y <- coverageObj$Y
readlength <- coverageObj$readlength
```

---

getgc *Get GC content for each exonic target*

Description

Computes GC content for each exon. Will be later used in QC procedure and normalization.

Usage

```
getgc(chr, ref)
```

Arguments

chr	Chromosome returned from getbambed
ref	IRanges object returned from getbambed

Value

Vector of GC content for each exon.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

References

Team TBD. BSgenome.Hsapiens.UCSC.hg19: Full genome sequences for Homo sapiens (UCSC version hg19). R package version 1.3.99.

See Also

getbambed, qc, normalize

Examples

```
ref <- IRanges(st = 51207851, end = 51207982)
gc <- getgc(chr = 22, ref)
```

---

getmapp *Get mappability for each exonic target*

Description

Computes mappability for each exon. To save running time, take values from pre-computed results. Will be later used in QC procedure.

Usage

```
getmapp(chr, ref)
```

Arguments

chr	Chromosome returned from getbambed
ref	IRanges object returned from getbambed

Details

To calculate the exonic mappability, we first construct consecutive reads of length 90 that are one base pair apart along the exon. The sequences are taken from the hg19 reference. We then find possible positions across the genome that the reads can map to allowing for two mismatches. We compute the mean of the probabilities that the overlapped reads map to the target places where they are generated and use this as the mappability of the exon.

Value

Vector of mappability for each exon.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

getbambed, qc

Examples

```
ref <- IRanges(st = 51207851, end = 51207982)
mapp <- getmapp(chr = 22, ref)
```

---

mappability *Pre-computed mappabilities*

Description

The results of pre-computed mappabilities to save running time.

Usage

```
data(mappability)
```

Details

Pre-computed mappabilities. Method used is detailed in getmapp.

Value

List of length 24 with pre-computed mappability of each chromosome.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

getmapp

Examples

```
# mappability of chromosome 1
head(round(mappability[[1]], 2))
```

mappDemo *Demo data pre-stored for mappability.*

Description

Pre-stored mappability data for demonstration purposes.

Usage

```
data(mappDemo)
```

Details

Pre-computed using whole exome sequencing data of 46 HapMap samples.

Value

mapp demo data (vector) pre-computed.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
head(round(mappDemo, 2))
```

mapp_ref *Position reference for pre-computed mappability results.*

Description

List consisting of IRanges objects specifying exonic positions whose mappabilities are pre-computed across the genome.

Usage

```
data(mapp_ref)
```

Details

Genomic positions for pre-computed mappabilities. Method used is detailed in getmapp.

Value

List of length 24 with genomic positions of pre-computed mappability of each chromosome.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

getmapp

Examples

```
# mappability exon reference of chromosome 1
mapp_ref[[1]]
```

---

normalize *Normalization of read depth from whole exome sequencing*

Description

Fits a Poisson log-linear model that normalizes the read depth data from whole exome sequencing. Includes terms that specifically remove biases due to GC content, exon capture and amplification efficiency, and latent systemic artifacts.

Usage

```
normalize(Y_qc, gc_qc, K)
```

Arguments

Y_qc	Read depth matrix after quality control procedure returned from qc
gc_qc	Vector of GC content for each exon after quality control procedure returned from qc
K	Number of latent Poisson factors. Can be an integer if optimal solution has been chosen or a vector of integers so that AIC, BIC, and RSS are computed for choice of optimal k.

Value

Yhat	Normalized read depth matrix
AIC	AIC for model selection
BIC	BIC for model selection
RSS	RSS for model selection
K	Number of latent Poisson factors

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

qc, choiceofK

Examples

```
Y_qc <- qcObjDemo$Y_qc
gc_qc <- qcObjDemo$gc_qc
normObj <- normalize(Y_qc, gc_qc, K = 1:5)
Yhat <- normObj$Yhat
AIC <- normObj$AIC
BIC <- normObj$BIC
RSS <- normObj$RSS
K <- normObj$K
```

`normalize2` *Normalization of read depth from whole exome sequencing under the case-control setting*

Description

Fits a Poisson log-linear model that normalizes the read depth data from whole exome sequencing. Includes terms that specifically remove biases due to GC content, exon capture and amplification efficiency, and latent systemic artifacts. If the WES is designed under case-control setting, CODEX estimates the exon-wise Poisson latent factor using only the read depths in the control cohort, and then computes the sample-wise latent factor terms for the case samples by regression.

Usage

```
normalize2(Y_qc, gc_qc, K, normal_index)
```

Arguments

`Y_qc`	Read depth matrix after quality control procedure returned from `qc`
`gc_qc`	Vector of GC content for each exon after quality control procedure returned from `qc`
`K`	Number of latent Poisson factors. Can be an integer if optimal solution has been chosen or a vector of integers so that AIC, BIC, and RSS are computed for choice of optimal k.
`normal_index`	Indices of control samples.

Value

`Yhat`	Normalized read depth matrix
`AIC`	AIC for model selection
`BIC`	BIC for model selection
`RSS`	RSS for model selection
`K`	Number of latent Poisson factors

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

`qc`, `choiceofK`

Examples

```
Y_qc <- qcObjDemo$Y_qc
gc_qc <- qcObjDemo$gc_qc
normObj <- normalize2(Y_qc, gc_qc, K = 1:5, normal_index = seq(1, 45, 2))
Yhat <- normObj$Yhat
AIC <- normObj$AIC
BIC <- normObj$BIC
RSS <- normObj$RSS
K <- normObj$K
```

---

normObjDemo *Demo data pre-stored for normObj.*

Description

Pre-stored normObj data for demonstration purposes.

Usage

```
data(normObjDemo)
```

Details

Pre-computed using whole exome sequencing data of 46 HapMap samples.

Value

normObj demo data (list) pre-computed.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
Yhat <- normObjDemo$Yhat
AIC <- normObjDemo$AIC
BIC <- normObjDemo$BIC
RSS <- normObjDemo$RSS
K <- normObjDemo$K
```

qc *Quality control procedure for depth of coverage*

Description

Applies a quality control procedure to the depth of coverage matrix both sample-wise and exon-wise before normalization.

Usage

```
qc(Y, sampname, chr, ref, mapp, gc,cov_thresh,length_thresh,mapp_thresh,
  gc_thresh)
```

Arguments

Y	Original read depth matrix returned from getcoverage
sampname	Vector of sample names returned from getbambed
chr	Chromosome.
ref	IRanges object specifying exonic positions returned from getbambed
mapp	Vector of mappability for each exon returned from getmapp
gc	Vector of GC content for each exon returned from getgc
cov_thresh	Vector specifying the upper and lower bound of exonic median coverage threshold for QC. 20-4000 recommended.
length_thresh	Vector specifying the upper and lower bound of exonic length threshold for QC. 20-2000 recommended.
mapp_thresh	Scalar variable specifying exonic mappability threshold for QC. 0.9 recommended.
gc_thresh	Vector specifying the upper and lower bound of exonic GC content threshold for QC. 20-80 recommended.

Details

It is suggested that analysis by CODEX be carried out in a batch-wise fashion if multiple batches exist. CODEX further filters out exons that: have extremely low coverage–median read depth across all samples less than 20 or greater than 4000; are extremely short–less than 20 bp; are extremely hard to map– mappability less than 0.9; have extreme GC content–less than 20 or greater than 80. The above filtering thresholds are recommended and can be user-defined to be adapted to different sequencing protocols.

Value

Y_qc	Updated Y after QC
sampname_qc	Updated sampname after QC
gc_qc	Updated gc after QC
mapp_qc	Updated mapp after QC
ref_qc	Updated ref after QC
qcmat	Matrix specifying results of exon-wise QC procedures

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

getbambed, getgc, getmapp

Examples

```
Y <- coverageObjDemo$Y
sampname <- bambedObjDemo$sampname
chr <- bambedObjDemo$chr
ref <- bambedObjDemo$ref
gc <- gcDemo
mapp <- mappDemo
cov_thresh <- c(20, 4000)
length_thresh <- c(20, 2000)
mapp_thresh <- 0.9
gc_thresh <- c(20, 80)
qcObj <- qc(Y, sampname, chr, ref, mapp, gc, cov_thresh, length_thresh,
    mapp_thresh, gc_thresh)
Y_qc <- qcObj$Y_qc
sampname_qc <- qcObj$sampname_qc
gc_qc <- qcObj$gc_qc
mapp_qc <- qcObj$mapp_qc
ref_qc <- qcObj$ref_qc
qcmat <- qcObj$qcmat
```

---

qcObjDemo *Demo data pre-stored for qcObj.*

Description

Pre-stored qcObj data for demonstration purposes.

Usage

```
data(qcObjDemo)
```

Details

Pre-computed using whole exome sequencing data of 46 HapMap samples.

Value

qcObj demo data (list) pre-computed.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
Y_qc <- qcObjDemo$Y_qc
sampname_qc <- qcObjDemo$sampname_qc
gc_qc <- qcObjDemo$gc_qc
mapp_qc <- qcObjDemo$mapp_qc
ref_qc <- qcObjDemo$ref_qc
```

---

segment *Recursive segmentation algorithm for CNV detection and genotyping*

Description

Recursive segmentation algorithm for CNV detection and genotyping, using normalized read depth from whole exome sequencing.

Usage

```
segment(Y_qc, Yhat, optK, K, sampname_qc, ref_qc, chr, lmax, mode)
```

Arguments

Y_qc	Raw read depth matrix after quality control procedure returned from qc
Yhat	Normalized read depth matrix returned from normalize
optK	Optimal value K returned from choiceofK
K	Number of latent Poisson factors. Can be an integer if optimal solution has been chosen or a vector of integers so that AIC, BIC, and RSS are computed for choice of optimal k.
sampname_qc	Vector of sample names after quality control procedure returned from qc
ref_qc	IRanges object of genomic positions of each exon after quality control procedure returned from qc
chr	Chromosome number returned from getbambed
lmax	Maximum CNV length in number of exons returned.
mode	Can be either "integer" or "fraction", which respectively correspond to format of the returned copy numbers.

Value

Final callset of CNVs with genotyping results.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

See Also

normalize, choiceofK

Examples

```
Y_qc <- qcObjDemo$Y_qc
Yhat <- normObjDemo$Yhat
BIC <- normObjDemo$BIC
K <- normObjDemo$K
sampname_qc <- qcObjDemo$sampname_qc
ref_qc <- qcObjDemo$ref_qc
chr <- bambedObjDemo$chr
finalcall <- segment(Y_qc, Yhat, optK = K[which.max(BIC)], K = K, sampname_qc,
    ref_qc, chr, lmax = 200, mode = "integer")
finalcall
```

Index

∗ **datasets**
bambedObjDemo, 3
coverageObjDemo, 4
gcDemo, 5
mapp_ref, 10
mappability, 9
mappDemo, 10
normObjDemo, 13
qcObjDemo, 15
∗ **package**
choiceofK, 3
CODEX-package, 2
getbambed, 5
getcoverage, 6
getgc, 7
getmapp, 8
normalize, 11
normalize2, 12
qc, 14
segment, 16

bambedObjDemo, 3

choiceofK, 3, *11, 12*, *16*
CODEX *(*CODEX-package*)*, 2
CODEX-package, 2
coverageObjDemo, 4

gcDemo, 5
getbambed, 5, *7–9*, *14–16*
getcoverage, *6*, 6, *14*
getgc, 7, *14, 15*
getmapp, 8, *9–11*, *14, 15*

mapp_ref, 10
mappability, 9
mappDemo, 10

normalize, *3, 4*, *8*, 11, *16*
normalize2, 12
normObjDemo, 13

qc, *8, 9*, *11, 12*, 14, *16*
qcObjDemo, 15

segment, *4*, 16