CODEX vignette

Yuchao Jiang
yuchaoj@wharton.upenn.edu

June 17, 2024

This is a demo for using the `CODEX` package in R. `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 capture 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. Below is an example on calling copy number variation using whole-exome sequencing data of 46 HapMap samples sequenced at the Washington University Genome Sequencing Center. Only the 401-500 exons from chromosome 22 are analysed for illustration purposes. R packages are available at Bioconductor for CODEX and the toy dataset WES.1KG.WUGSC.

1. Website and online forum

CODEX's website with usage and installation information: https://github.com/yuchaojiang/CODEX.
Online Q&A forum: https://groups.google.com/d/forum/codex_wes_cnv.
If you've any questions regarding the software, please don't hesitate emailing us at codex_wes_cnv@googlegroups.com.

2. CODEX workflow:

2.1 Install CODEX.

Install the current release from Bioconductor:

```
> ## try http:// if https:// URLs are not supported
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("CODEX")
```

Install the devel version from GitHub:

```
> install.packages("devtools")
> library(devtools)
> install_github("yuchaojiang/CODEX/package")
```

2.2 Get directories of .bam files, read in exon target positions from .bed files, and get sample names.

The direct input of `CODEX` include: `bamdir`, which is a vector indicating the directories of all .bam files; `sampname`, which is a column vector with row entries of sample names; `bedFile`, which indicates the directory of the .bed file (WES bait file, no header, sorted by start and end positions); and `chr`, which specifies the chromosome. `CODEX` processes the entire genome chromosome by chromosome; make sure the chromosome formats are consistent between the .bed and the .bam files.

```
> library(CODEX)
> library(WES.1KG.WUGSC) # Load Toy data from the 1000 Genomes Project.
> dirPath <- system.file("extdata", package = "WES.1KG.WUGSC")
> bamFile <- list.files(dirPath, pattern = '*.bam$')
> bamdir <- file.path(dirPath, bamFile)
> sampname <- as.matrix(read.table(file.path(dirPath, "sampname")))
> bedFile <- file.path(dirPath, "chr22_400_to_500.bed")
> chr <- 22
> bambedObj <- getbambed(bamdir = bamdir, bedFile = bedFile,
+                        sampname = sampname, projectname = "CODEX_demo", chr)
> bamdir <- bambedObj$bamdir; sampname <- bambedObj$sampname
> ref <- bambedObj$ref; projectname <- bambedObj$projectname; chr <- bambedObj$chr
```

2.3 Get raw read depth from the .bam files. Read lengths across all samples are also returned.

```
> coverageObj <- getcoverage(bambedObj, mapqthres = 20)
> Y <- coverageObj$Y; readlength <- coverageObj$readlength
```

2.4 Compute GC content and mappability for each exon target.

```
> gc <- getgc(chr, ref)
> mapp <- getmapp(chr, ref)
```

2.5 Take a sample-wise and exon-wise quality control procedure on the depth of coverage matrix.

```
> qcObj <- qc(Y, sampname, chr, ref, mapp, gc, cov_thresh = c(20, 4000),
+     length_thresh = c(20, 2000), mapp_thresh = 0.9, gc_thresh = c(20, 80))
> 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
> write.table(qcmat, file = paste(projectname, '_', chr, '_qcmat', '.txt', sep=''),
+             sep='\t', quote=FALSE, row.names=FALSE)
```

2.6 Fit Poisson latent factor model for normalization of the read depth data.

```
> normObj <- normalize(Y_qc, gc_qc, K = 1:9)
> Yhat <- normObj$Yhat; AIC <- normObj$AIC; BIC <- normObj$BIC
> RSS <- normObj$RSS; K <- normObj$K
```

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. `normal_index` specifies the indices of normal samples and the normalization function to use under this setting is `normalize2`.

```
> normObj <- normalize2(Y_qc, gc_qc, K = 1:9, normal_index=seq(1,45,2))
> Yhat <- normObj$Yhat; AIC <- normObj$AIC; BIC <- normObj$BIC
> RSS <- normObj$RSS; K <- normObj$K
```

2.7 Determine the number of latent factors. AIC, BIC, and deviance reduction plots are generated in a .pdf file.

CODEX reports all three statistical metrics (AIC, BIC, percent of Variance explained) and uses BIC as the default method to determine the number of Poisson factors. Since false positives can be screened out through a closer examination of the post-segmentation data, whereas CNV signals removed in the normalization step

cannot be recovered, CODEX opts for a more conservative normalization that, when in doubt, uses a smaller value of K.

```
> choiceofK(AIC, BIC, RSS, K, filename = paste(projectname, "_", chr,
+     "_choiceofK", ".pdf", sep = ""))
```

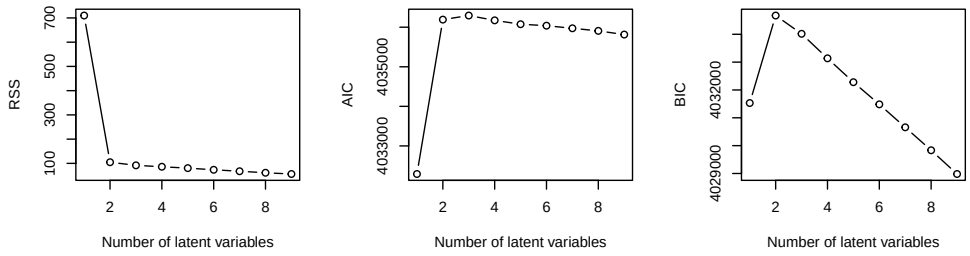

Figure 1: Determination of K via AIC, BIC, and deviance reduction. Optimal K is set at 2.

2.8 Fit the Poisson log-likelihood ratio based segmentation procedure to determine CNV regions across all samples.

For germline CNV detection, `CODEX` uses the "integer" mode; for CNV detection involving large recurrent chromosomal aberrations in mixture populations (e.g. somatic CNV detection in cancer), `CODEX` opts to use the "fraction" mode.
The output file is tab delimited and has 13 columns with rows corresponding to CNV events. The columns include sample_name (sample names), chr (chromosome), cnv (deletion or duplication), st_bp (cnv start position in base pair, the start position of the first exon in the cnv), ed_bp (cnv end position in base pair, the end position of the last exon in the cnv), length_kb (CNV length in kb), st_exon (the first exon after QC in the cnv, integer value numbered in qcObj$ref_qc), ed_exon (the last exon after QC in the cnv, integer value numbered in qcObj$ref_qc), raw_cov (raw coverage), norm_cov (normalized coverage), copy_no (copy number estimate), lratio (likelihood ratio of CNV event versus copy neutral event), mBIC (modified BIC value, used to determine the stop point of segmentation), pvalue (p-values computed by the Wilk's theorem from the likelihood ratio test).
For the "fraction" mode post segmentation thresholding is necessary to filter out long CNV events with factional copy numbers close to 2.

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

 sample_name chr  cnv   st_bp      ed_bp      length_kb st_exon ed_exon raw_cov
 "NA18990"   "22" "dup" "22312814" "22326373" "13.56"   "60"    "72"    "1382"
 norm_cov copy_no lratio  mBIC
 "1000"   "3"     "60.33" "49.728"

> write.table(finalcall, file = paste(projectname, '_', chr, '_', optK,
+             '_CODEX_frac.txt', sep=''), sep='\t', quote=FALSE, row.names=FALSE)
> save.image(file = paste(projectname, '_', chr, '_image', '.rda', sep=''),
+      compress='xz')
```

3. Citation

CODEX: a normalization and copy number variation detection method for whole exome sequencing Yuchao Jiang; Derek A. Oldridge; Sharon J. Diskin; Nancy R. Zhang Nucleic Acids Research 2015; doi: 10.1093/nar/gku1363 (html, pdf).

4. Session information:

Output of sessionInfo on the system on which this document was compiled:

- R version 4.4.0 (2024-04-24), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Time zone: `Etc/UTC`
- TZcode source: `system (glibc)`
- Running under: `Ubuntu 24.04 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3`
- LAPACK: `/usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.26.so` ; LAPACK version3.12.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, stats4, utils
- Other packages: BSgenome 1.73.0, BSgenome.Hsapiens.UCSC.hg19 1.4.3, BiocGenerics 0.51.0, BiocIO 1.15.0, Biostrings 2.73.1, CODEX 1.37.0, GenomeInfoDb 1.41.1, GenomicRanges 1.57.1, IRanges 2.39.0, Rsamtools 2.21.0, S4Vectors 0.43.0, WES.1KG.WUGSC 1.37.0, XVector 0.45.0, rtracklayer 1.65.0
- Loaded via a namespace (and not attached): Biobase 2.65.0, BiocParallel 1.39.0, DelayedArray 0.31.1, GenomeInfoDbData 1.2.12, GenomicAlignments 1.41.0, Matrix 1.7-0, MatrixGenerics 1.17.0, R6 2.5.1, RCurl 1.98-1.14, S4Arrays 1.5.1, SparseArray 1.5.8, SummarizedExperiment 1.35.0, UCSC.utils 1.1.0, XML 3.99-0.16.1, abind 1.4-5, bitops 1.0-7, buildtools 1.0.0, codetools 0.2-20, compiler 4.4.0, crayon 1.5.2, curl 5.2.1, grid 4.4.0, httr 1.4.7, jsonlite 1.8.8, knitr 1.47, lattice 0.22-6, maketools 1.3.0, matrixStats 1.3.0, parallel 4.4.0, restfulr 0.0.15, rjson 0.2.21, sys 3.4.2, tools 4.4.0, xfun 0.44, yaml 2.3.8, zlibbioc 1.51.1