Instrukcja do ćwiczenia nr 4 z MBI
Analiza danych sekwencyjnych człowieka

Tomasz Gambin, Wiktor Kuśmirek, Robert Nowak

2024Z

1 Wstęp

Zmiany liczby kopii (delecje i duplikacje) mogą być wykryte na podstawie analizy zmian głębokości pokrycia. Zmniejszone pokrycie w zadanym regionie genomu wskazuje na delecje, natomiast zwiększone na duplikacje. Ze względu na duże fluktuacje głębokości pokrycia wynikające z występowania artefaktów technologicznych, do wykrywania zmian liczby kopii stosuje się jednoczesną analizę wielu próbek. Analiza składa się z następujących etapów:

- obliczenia głębokości pokrycia dla kolejnych regionów genomu (np. eksonów);
- kontrola jakości (usunięcie słabo pokrytych regionów oraz odstających próbek);
- normalizacji głębokości pokrycia w każdym eksonie (względem innych próbek);
- segmentacji i identyfikacji zmian liczby kopii.

2 Instalacja narzędzi i pobranie danych wejściowych

Instrukcja pokazuje poszczególne kroki na Ubuntu 20.04 w wersji 64 bitowej. Na samym początku ćwiczenia utwórz katalog roboczy i pobierz dane:

```
mkdir ~/mbi_cwiczenie_4/
cd mbi_cwiczenie_4/
mkdir data
cd data

wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/DGV.tar.gz
wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/TGP_SV.tar.gz
wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/bed.tar.gz
wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/codex_output_all.tar.gz
wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/coverage.tar.gz
wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/refFlat.tar.gz
wget https://github.com/NGSchoolEU/2017/raw/master/CNV_detection/data/segDups.tar.gz
```

Rozpakuj pobrane pliki, następnie przejdź do katalogu roboczego:

```
cd ~/mbi_cwiczenie_4/
```

Opis zawartości plików w folderze data:

- bed - współrzędne regionów sekwencjonowanych metodą WES w projekcie 1000 Genomes (pobrane z `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/exome_pull...`)
- coverage - przeliczone pokrycie dla 99 próbek z projektu 1000 Genomes
- codex_output - wyniki analizy za pomocą programu CODEX dla 99 próbek
- segDups - współrzędne segmentalnych duplikacji w genomie człowieka
- TGP_SV - zmiany liczby kopii wyznaczone przez konsorcjum 1000 Genomes w wyniku jednoczesnej analizy danych z WES i WGS (pobrane z `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/integrated_sv_map/ALL.w...`)
- refFlat - współrzędne genów (pobrane z `ftp://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/refFlat.txt.gz`)
- DGV - zmiany liczby kopii z repozytorium "Database of Genomic Variants"(pobrane z `ftp://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/dgvSupporting.tx..`)

Uruchom przygotowany do ćwiczenia kontener docker:

```
sudo docker run --rm -it -v ~/mbi_cwiczenie_4:/mbi -w /mbi wkusmirek/mbi-lab-4 R
```

Załaduj wymagane biblioteki, ustaw katalog roboczy:

```
library(data.table)
library(parallel)
library(RCurl)
library(gdata)
library(matrixStats)
library(DNAcopy)
library(GenomicRanges)
library(Rsubread)
library(WES.1KG.WUGSC)
library(CODEX)

# set working directory to workDir
workDir <- "/mbi/"
setwd(workDir)
#set number of available cores
cores <- 4
```

3 Liczenie pokrycia

Załaduj przykładowe fragmenty plików BAM z WES oraz policz pokrycie wykorzystując metodę getcoverage() z pakietu CODEX. W pakiecie "WES.1KG.WUGSC" znajduje się 46 plików BAM z odczytami pochodzącego z niewielkiego fragmentu chromosomu 22. Odpowiadający plik bed zawiera definicję dla 100 reginów z tego chromosomu (podejrzyj zawartość pliku poleceniem fread(bedFile)).

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

Objekt Y zawiera dane na temat liczby odczytów dla każdego regionu; próbki są zdefiniowane w kolejnych kolumnach; regiony w wierszach.

```
Y_ac <- apply(Y, 2,function(x){(100*x)/width(ref)})
colnames(Y_ac) <- sampname
```

Policz statystyki głębokości pokrycia:

```
summary(apply(Y_ac, 2, median))
```

Która próbka ma najmniejsze, a która największą medianę głębokości pokrycia?

4 Wykrywanie zmian liczby kopii DNA przy użyciu narzędzia CODEX

Zdefiniuj nazwę projektu oraz utwórz katalog wyjściowy:

```
projectname <- "TGP99"
outputDir <- paste0(workDir,"codex_output/")
dir.create(outputDir)
```

Załaduj wstępnie przeliczone dane o głębokości pokrycia dla 99 próbek z 1000 Genomes i przygotuj dane dla chromosomu 20:

```
cfiles <- dir(paste0(workDir, "data/coverage/"), "*bam.coverage*")
cdf <- rbindlist(mclapply(cfiles, function(f){
  print(f);
  df <- fread(paste0(workDir, "data/coverage/",f));
  df$SampleName <- strsplit(f, "\\.")[[1]][1];df
},mc.cores=cores))
colnames(cdf) <- c("Chr", "Start", "Stop", "ReadCount", "SampleName")
cdf <- cdf[order(cdf$SampleName, cdf$Chr, cdf$Start, cdf$Stop),]
dim(cdf)
#[1] 465498        5
head(cdf)
#   Chr  Start   Stop ReadCount SampleName
#1:  20  68319  68439       103    NA06985
#2:  20  76611  77091       129    NA06985
#3:  20 123208 123358        69    NA06985
#4:  20 125995 126389       105    NA06985
#5:  20 138119 138269        37    NA06985
#6:  20 139359 139719       156    NA06985
bedFile <- paste0(workDir, "data/bed/20130108.exome.targets.bed")
sampname <- unique(cdf$SampleName)

chr <- "20"
targetsChr <-  cdf[which(cdf$Chr==chr & cdf$SampleName == cdf$SampleName[1]),
                   c("Chr", "Start",  "Stop")]
selChr <- cdf[which(cdf$Chr==chr),]
Y <- t(do.call(rbind,lapply(sampname,
               function(s){selChr$ReadCount [which(selChr$SampleName == s)]})))
colnames(Y) <- sampname
rownames(Y) <- 1:nrow(Y)
dim(Y)
#[1] 4702   99
dim(targetsChr)
#[1] 4702    3
ref <- IRanges(start = targetsChr$Start, end = targetsChr$Stop)
gc <- getgc(chr, ref)
mapp <- getmapp(chr, ref)
```

Przeprowadź kontrolę jakości:

```
mapp_thresh <- 0.9 # remove exons with mapability < 0.9
cov_thresh_from <- 20 # remove exons covered by less than 20 reads
cov_thresh_to <- 4000 #  remove exons covered by more than 4000 reads
length_thresh_from <- 20 # remove exons of size < 20
length_thresh_to <- 2000 # remove exons of size > 2000
gc_thresh_from <- 20 # remove exons with GC < 20
gc_thresh_to <- 80 # or GC > 80
```

```
qcObj <- qc(Y, sampname, chr, ref, mapp, gc,
            cov_thresh = c(cov_thresh_from, cov_thresh_to),
            length_thresh = c(length_thresh_from, length_thresh_to),
            mapp_thresh = mapp_thresh,
            gc_thresh = c(gc_thresh_from, gc_thresh_to))
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
```

Ile eksonów (regionów) zostało usuniętych w wyniku kontroli jakości?

Ile eksonów (regionów) będzie usuniętych jeżeli zmienimy progi odcięcia zawartości GC (gc_thresh_from, gc_thresh_to)? Podaj wynik dla wybranych przez Ciebie progów.

Przeprowadź normalizację głębokości pokrycia oraz segmentację:

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

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

finalcall <- CODEX::segment(Y_qc, Yhat, optK = optK,
                            K = K, sampname_qc,
                            ref_qc, chr, lmax = 200,
                            mode = "integer")
finalcall <- data.frame(finalcall, stringsAsFactors=F)
finalcall$targetCount <- as.numeric(finalcall$ed_exon) - as.numeric(finalcall$st_exon)
```

Przejrzyj zawatość obiektu finalcall. Poniżej informacja co zawierają kolejne kolumny:

- sampl_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 qcObjref_qc), ed_exon (the last exon after QC in the cnv, integer value numbered in qcObjref_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)

Ile zmian liczby kopii wykrył CODEX?
Ile wśród nich jest delecji a ile duplikacji?
Czy występują jakiekolwiek homozygotyczne delecje (copy_no ==0)?
Załaduj funkcję plotCall:

```
plotCall <- function(calls, i, Y_qc, Yhat_opt){
  startIdx <- as.numeric(calls$st_exon[i])
  stopIdx <- as.numeric(calls$ed_exon[i])
  sampleName <- calls$sample_name[i]
  wd <- 20
  startPos <- max(1,(startIdx-wd))
  stopPos <- min((stopIdx+wd), nrow(Y_qc))
  selQC <- Y_qc[startPos:stopPos,]
  selQC[selQC ==0] <- 0.00001
  selYhat <- Yhat_opt[startPos:stopPos,]
  png(file="cnv.png")
  matplot(matrix(rep(startPos:stopPos, ncol(selQC)),
           ncol=ncol(selQC)), log(selQC/selYhat,2),
           type="l",lty=1, col="dimgrey",  lwd=1,
           xlab="exon nr", ylab="logratio(Y/Yhat)")
  lines(startPos:stopPos,log( selQC[,sampleName]/ selYhat[,sampleName],2), lwd=3, col="red")
  dev.off()
}
```

i wykonaj wykres dla wybranej zmiany, np:

```
cnvId <- 1    # indeks zmiany dla której zostanie sporządzony wykres
plotCall(finalcall, cnvId, Y_qc, Yhat[[optK]])
```

Uzyskany wykres załącz do sprawozdania.

5 Zadanie implementacyjne

- Zapoznaj się z publicznie dostępną bazą danych wariantów strukturalnych Database of Genomic Variants (`http://dgv.tcag.ca/dgv/app/downloads`).
- Pobierz dane o wariantach strukturalnych a następnie wykorzystaj je aby zaadnotować zmiany liczby kopii, które zostały wykryte w tym ćwiczeniu.
- Dla każdego CNV wykrytego w poprzednich krokach tego ćwiczenia podaj informację:
  - Ile delecji z DGV ma jakąkolwiek część wspólną z zadanym CNV?
  - Ile duplikacji z DGV ma jakąkolwiek część wspólną z zadanym CNV?

  - Ile dowolnych zmian z DGV ma 80-procentową część wspólną z zadanym CNV?

- Dane zaprezentuj w formie tabelarycznej. Należy dokonać implementacji w języku R lub python.