

# CNproScan Vignette

Robin Jugas

08/08/2022

## Contents

|  |          |
|--|----------|
| <b>CNproScan Vignette</b>                    | <b>1</b> |
| 1. Website and issues                        | 1        |
| 2. CNproScan workflow                        | 1        |
| 2.1 Install CNproScan                        | 1        |
| 2.2 Reads alignment and coverage calculation | 1        |
| 2.3 CNV detection                            | 2        |
| 2.4 The output of CNproScanCNV               | 2        |
| 3. Session information                       | 2        |

## CNproScan Vignette

**CNproScan** is R package developed for CNV detection in bacterial genomes. It employs Generalized Extreme Studentized Deviate test for outliers to detect CNVs in read-depth data with discordant reads detection to annotate the CNVs. It was tested and proven to be able to detect short CNVs. Following text is a workflow showcase. The **CNproScan** consist of a single function *CNproScanCNV* which carries the whole procedure. The steps necessary to get input files is also explained at the GitHub repository.

### 1. Website and issues

CNproScan github repository and issues reporting: <https://github.com/robinjugas/CNproScan>

### 2. CNproScan workflow

#### 2.1 Install CNproScan

Install CNproScan from Bioconductor:

```
## try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
biocLite("CNproScan")
```

or install the most current release from Github:

```
install.packages("devtools")
library(devtools)
install_github("robinjugas/CNproScan")
```

#### 2.2 Reads alignment and coverage calculation

Apply the following steps to get the files needed to CNV detection.

```

# Prerequisite: reference fasta file, samtools, bwa aligner
# Alignment
bwa index -a is reference.fasta
samtools faidx reference.fasta
bwa mem reference.fasta read1.fq read2.fq > file.sam
# BAM processing
samtools view -b -F 4 file.sam > file.bam # mapped reads only
samtools sort -o file.bam file1.bam
samtools index file.bam
# Calculate coverage with zero coverage reported
samtools depth -a file.bam > file.coverage
# Genome mappability file by GENMAP (https://github.com/cpockrandt/genmap)
# only for mappability normalization
genmap index -F reference.fasta -I mapp_index
genmap map -K 30 -E 2 -I mapp_index -O mapp_genmap -t -w -bg

```

## 2.3 CNV detection

```

library("CNproScan")
# Working directory with files
setwd("workdir")
# File paths
fasta_file <- "reference.fasta"
bam_file <- "file.bam"
coverage_file <- "file.coverage"
bedgraph_file <- "mapp_genmap.bedgraph"

# For only GC normalization
DF <- CNproScanCNV(coverage_file, bam_file, fasta_file,
                  GCnorm=TRUE, MAPnorm=FALSE, cores=4)
# Without any normalization
DF <- CNproScanCNV(coverage_file, bam_file, fasta_file,
                  GCnorm=FALSE, MAPnorm=FALSE, cores=4)
# Both GC normalization and mappability normalization
DF <- CNproScanCNV(coverage_file, bam_file, fasta_file,
                  GCnorm=TRUE, MAPnorm=TRUE, bedgraph_file, cores=4)

```

## 2.4 The output of CNproScanCNV

The output is a data.frame object with several descriptive columns. Also, the VCF file is written into the working directory.

## 3. Session information

```

sessionInfo()

## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.1 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0

```

```
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_GB.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_GB.UTF-8  LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_GB.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## loaded via a namespace (and not attached):
## [1] compiler_4.1.3  magrittr_2.0.3  fastmap_1.1.0  cli_3.3.0
## [5] tools_4.1.3     htmltools_0.5.3 rstudioapi_0.13 yaml_2.3.5
## [9] stringi_1.7.8   rmarkdown_2.14 knitr_1.39     stringr_1.4.0
## [13] xfun_0.31       digest_0.6.29  rlang_1.0.4    evaluate_0.15
```