Copy Number Analysis using CNVpytor

February, 2022

Table of Contents

# About this Course

Copy number analysis is the process of detecting copy number information from a genome. This course will guide one to get copy number information from alignment file using CNVpytor.

# 1 Introduction

CNVpytor is a Python package and command line tool for copy number variation (CNV)/Copy Number alteration (CNA) analysis from depth-of-coverage by mapped reads developed in Abyzov Lab, Mayo Clinic.

## 1.1 Motivation

## 1.2 Target Audience

The course is intended for …

## 1.3 Curriculum

The course covers…

# 2 Installation and Setting Reference genome

## 2.1 Learning Objectives

This chapter will cover:

* Installation
  + using setuptools
  + using pip
* Steps for setting reference genome
  + Create GC and mask file for new reference genome

## 2.2 Libraries

CNVpytor is written in python and it works on both python 2 and 3. Please install python before proceeding with the installation steps.

The following code can be used to check python version

python --version

## 2.3 Installation

### 2.3.1 Install by cloning from GitHub

The following lines of codes can be used to install directly from GitHub

> git clone https://github.com/abyzovlab/CNVpytor.git  
> cd CNVpytor  
> pip install .

For single user (without admin privileges), can use:

> pip install --user .

### 2.3.2 Install using pip

The following code will download the latest code from GitHub and use pip to install.

pip install git+https://github.com/abyzovlab/CNVpytor.git

## 2.4 Steps for setting reference genome

Commonly used human reference genomes (HG19 and HG38) are integrated and comes with default GitHub installation. Its detects the genome by comparing the chromosome lengths. Although, other reference genomes are also frequently used in practice. This section will guide one to add a new reference genome.

Now, we will create example configuration file for mouse reference genome MGSCv37 which contains a list of chromosomes and chromosome lengths:

# Filename: example\_ref\_genome\_conf.py  
  
import\_reference\_genomes = {  
 "mm9": {  
 "name": "MGSCv37",  
 "species": "Mus musculus",  
 "chromosomes": OrderedDict(  
 [("chr1", (197195432, "A")), ("chr2", (181748087, "A")), ("chr3", (159599783, "A")),  
 ("chr4", (155630120, "A")), ("chr5", (152537259, "A")), ("chr6", (149517037, "A")),  
 ("chr7", (152524553, "A")), ("chr8", (131738871, "A")), ("chr9", (124076172, "A")),  
 ("chr10", (129993255, "A")), ("chr11", (121843856, "A")), ("chr12", (121257530, "A")),  
 ("chr13", (120284312, "A")), ("chr14", (125194864, "A")), ("chr15", (103494974, "A")),  
 ("chr16", (98319150, "A")), ("chr17", (95272651, "A")), ("chr18", (90772031, "A")),  
 ("chr19", (61342430, "A")), ("chrX", (166650296, "S")), ("chrY", (15902555, "S")),  
 ("chrM", (16299, "M"))]),  
 "gc\_file": "/..PATH../MGSCv37\_gc\_file.pytor",  
 "mask\_file": "/..PATH../MGSCv37\_mask\_file.pytor"  
 }  
}

Letter next to chromosome length denote type of a chromosome: A - autosome, S - sex chromosome, M - mitochondria. The instruction for creating MGSCv37\_gc\_file.pytor and MGSCv37\_mask\_file.pytor is in the next section.

To use CNVpytor with new reference genome us -conf option in each cnvpytor command, e.g.

cnvpytor -conf REL\_PATH/example\_ref\_genome\_conf.py -root file.pytor -rd file.bam

CNVpytor will use chromosome lengths from alignment file to detect reference genome. However, if you configured reference genome after you had already run -rd step you could assign reference genome using -rg:

cnvpytor -conf REL\_PATH/example\_ref\_genome\_conf.py -root file.pytor -rg mm9

To avoid typing -conf REL\_PATH/example\_ref\_genome\_conf.py each time you run cnvpytor, you can create an bash alias or make configuration permanent by copying example\_ref\_genome\_conf.py to ~/.cnvpytor/reference\_genomes\_conf.py.

#### 2.4.0.1 Create GC and mask file for new reference genome

CNVpytor also has optional features for GC correction and masking (i.e., commonly known false positive regions). One can setup their reference genome by adding its related content in the gc\_file and mask\_file field of the configuration file.

To create GC file, we need sequence of the reference genome in fasta.gz file:

> cnvpytor -root MGSCv37\_gc\_file.pytor -gc ~/hg19/mouse.fasta.gz -make\_gc\_file

This command will produce MGSCv37\_gc\_file.pytor file that contains information about GC content in 100-base-pair bins. For reference genomes where we have strict mask in the same format as [1000 Genomes Project strict mask](http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/working/20160622_genome_mask_GRCh38/), we can create mask file using command:

> cnvpytor -root MGSCv37\_mask\_file.pytor -mask ~/hg19/mouse.strict\_mask.whole\_genome.fasta.gz -make\_mask\_file

If you do not have mask file, You can skip this step. Mask file contains information about regions of the genome that are more accessible to next generation sequencing methods using short reads. CNVpytor uses P marked positions to filter SNP-s and read depth signal. If reference genome configuration does not contain mask file, CNVpytor will still be fully functional, apart from the filtering step. You may also generate your own mask file by creating fasta file that contains character “P” if corresponding base pair passes the filter and any character different than “P” if not.

### 2.4.1 Code examples

You can demonstrate code like this:

output\_dir <- file.path("resources", "code\_output")  
if (!dir.exists(output\_dir)) {  
 dir.create(output\_dir)  
}

And make plots too:

hist\_plot <- hist(iris$Sepal.Length)



You can also save these plots to file:

png(file.path(output\_dir, "test\_plot.png"))  
hist\_plot

## $breaks  
## [1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0  
##   
## $counts  
## [1] 5 27 27 30 31 18 6 6  
##   
## $density  
## [1] 0.06666667 0.36000000 0.36000000 0.40000000 0.41333333 0.24000000 0.08000000  
## [8] 0.08000000  
##   
## $mids  
## [1] 4.25 4.75 5.25 5.75 6.25 6.75 7.25 7.75  
##   
## $xname  
## [1] "iris$Sepal.Length"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"

dev.off()

## png   
## 2

### 2.4.2 Image example

How to include a Google slide. It’s simplest to use the ottr package:



But if you have the slide or some other image locally downloaded you can also use html like this:

### 2.4.3 Video examples

To show videos in your course, you can use markdown syntax like this:

[A video we want to show](https://www.youtube.com/embed/VOCYL-FNbr0)

Alternatively, you can use knitr::include\_url() like this: Note that we are using echo=FALSE in the code chunk because we don’t want the code part of this to show up. If you are unfamiliar with [how R Markdown code chunks work, read this](https://rmarkdown.rstudio.com/lesson-3.html).

## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

OR this works:

<iframe src=“<https://www.youtube.com/watch?v=RJMQtrD0SuE> width=”672” height=“400px”>

### 2.4.4 Links to files

This works:

Or this:

[This works](https://www.messiah.edu/download/downloads/id/921/Microaggressions_in_the_Classroom.pdf).

Or this:

### 2.4.5 Links to websites

Examples of including a website link.

This works:

OR this:

![Another link](data:text/html; charset=utf-8;base64,)

Figure : Another link

OR this:

### 2.4.6 Citation examples

We can put citations at the end of a sentence like this ([Allaire et al. 2021](#ref-rmarkdown2021)). Or multiple citations Xie, Allaire, and Grolemund ([2018](#ref-Xie2018)).

but they need a ; separator ([Allaire et al. 2021](#ref-rmarkdown2021); [Xie, Allaire, and Grolemund 2018](#ref-Xie2018)).

In text, we can put citations like this Allaire et al. ([2021](#ref-rmarkdown2021)).

### 2.4.7 FYI boxes

Please click on the subsection headers in the left hand navigation bar (e.g., 2.1, 4.3) a second time to expand the table of contents and enable the scroll\_highlight feature ([see more](introduction.html#scroll-highlight)).

### 2.4.8 Dropdown summaries

You can hide additional information in a dropdown menu

Here’s more words that are hidden.

## 2.5 Print out session info

You should print out session info when you have code for [reproducibility purposes](https://jhudatascience.org/Reproducibility_in_Cancer_Informatics/managing-package-versions.html).

devtools::session\_info()

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.3 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2022-02-11   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.3)   
## bookdown 0.24 2022-02-11 [1] Github (rstudio/bookdown@88bc4ea)   
## callr 3.4.4 2020-09-07 [1] RSPM (R 4.0.2)   
## cli 2.0.2 2020-02-28 [1] RSPM (R 4.0.0)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## curl 4.3 2019-12-02 [1] RSPM (R 4.0.3)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.14 2019-05-28 [1] RSPM (R 4.0.3)   
## fansi 0.4.1 2020-01-08 [1] RSPM (R 4.0.0)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.6.1 2022-01-22 [1] CRAN (R 4.0.2)   
## highr 0.8 2019-03-20 [1] RSPM (R 4.0.3)   
## hms 0.5.3 2020-01-08 [1] RSPM (R 4.0.0)   
## htmltools 0.5.0 2020-06-16 [1] RSPM (R 4.0.1)   
## httr 1.4.2 2020-07-20 [1] RSPM (R 4.0.3)   
## knitr 1.33 2022-02-11 [1] Github (yihui/knitr@a1052d1)   
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)   
## magrittr 2.0.2 2022-01-26 [1] CRAN (R 4.0.2)   
## memoise 1.1.0 2017-04-21 [1] RSPM (R 4.0.0)   
## ottr 0.1.2 2022-02-11 [1] Github (jhudsl/ottr@2d03822)   
## pillar 1.4.6 2020-07-10 [1] RSPM (R 4.0.2)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgconfig 2.0.3 2019-09-22 [1] RSPM (R 4.0.3)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## png 0.1-7 2013-12-03 [1] CRAN (R 4.0.2)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.3.4 2020-08-11 [1] RSPM (R 4.0.2)   
## purrr 0.3.4 2020-04-17 [1] RSPM (R 4.0.3)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## readr 1.4.0 2020-10-05 [1] RSPM (R 4.0.2)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 0.4.10 2022-02-11 [1] Github (r-lib/rlang@f0c9be5)   
## rmarkdown 2.10 2022-02-11 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.2 2020-11-15 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2022-02-11 [1] Github (R-lib/testthat@e99155a)   
## tibble 3.0.3 2020-07-10 [1] RSPM (R 4.0.2)   
## usethis 2.1.5.9000 2022-02-11 [1] Github (r-lib/usethis@57b109a)   
## vctrs 0.3.4 2020-08-29 [1] RSPM (R 4.0.2)   
## webshot 0.5.2 2019-11-22 [1] RSPM (R 4.0.3)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2022-02-11 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# 3 Calling CNV from alignment file using terminal

This section will guide to use CNVpytor for calling CNV using Read depth file and incorporating variant information

## 3.1 Learning Objectives

This chapter will cover:

* Steps to process alignment file
* Steps to process variant information.

# 4 Calling CNV from alignment file

The following steps can be used to process read depth information from alignment file

> cnvpytor -root file.pytor -rd file.bam

If the reference genome is human than there is no need to set the reference genome and one can run the following steps.

> cnvpytor -root file.pytor -his 1000 10000 100000  
> cnvpytor -root file.pytor -partition 1000 10000 100000  
> cnvpytor -root file.pytor -call 1000 10000 100000

For non human reference genome, please have a look at the section for setting reference genome.

## 4.1 Importing and using variant information:

> cnvpytor -root file.pytor -snp file.vcf -sample sample\_name  
> cnvpytor -root file.pytor -pileup file.bam # OPTIONAL  
> cnvpytor -root file.pytor -mask\_snps # OPTIONAL   
> cnvpytor -root file.pytor -baf 10000 100000

# About the Authors

These credits are based on our [course contributors table guidelines](https://github.com/jhudsl/OTTR_Template/wiki/How-to-give-credits).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([ottr](https://github.com/jhudsl/ottr)) | [John Muschelli](https://johnmuschelli.com/), [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.3 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2022-02-11   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.3)   
## bookdown 0.24 2022-02-11 [1] Github (rstudio/bookdown@88bc4ea)   
## callr 3.4.4 2020-09-07 [1] RSPM (R 4.0.2)   
## cli 2.0.2 2020-02-28 [1] RSPM (R 4.0.0)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.14 2019-05-28 [1] RSPM (R 4.0.3)   
## fansi 0.4.1 2020-01-08 [1] RSPM (R 4.0.0)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.6.1 2022-01-22 [1] CRAN (R 4.0.2)   
## htmltools 0.5.0 2020-06-16 [1] RSPM (R 4.0.1)   
## knitr 1.33 2022-02-11 [1] Github (yihui/knitr@a1052d1)   
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)   
## magrittr 2.0.2 2022-01-26 [1] CRAN (R 4.0.2)   
## memoise 1.1.0 2017-04-21 [1] RSPM (R 4.0.0)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.3.4 2020-08-11 [1] RSPM (R 4.0.2)   
## purrr 0.3.4 2020-04-17 [1] RSPM (R 4.0.3)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 0.4.10 2022-02-11 [1] Github (r-lib/rlang@f0c9be5)   
## rmarkdown 2.10 2022-02-11 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.2 2020-11-15 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2022-02-11 [1] Github (R-lib/testthat@e99155a)   
## usethis 2.1.5.9000 2022-02-11 [1] Github (r-lib/usethis@57b109a)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2022-02-11 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2021. *Rmarkdown: Dynamic Documents for r*. <https://github.com/rstudio/rmarkdown>.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown>.