PBGL CNV-seq Analysis v1.0

A Laboratory Manual

Anibal E. Morales-Zambrana

Plant Breeding and Genetics Laboratory
FAO/IAEA Joint Division
Seibersdorf, Austria

Created: March, 2021
Last updated: 25 March 2021

Please note: *This is not an official IAEA publication but is made available as working material. The material has not undergone an official review by the IAEA. The views expressed do not necessarily reflect those of the International Atomic Energy Agency or its Member States and remain the responsibility of the contributors. The use of particular designations of countries or territories does not imply any judgement by the publisher, the IAEA, as to the legal status of such countries or territories, of their authorities and institutions or of the delimitation of their boundaries. The mention of names of specific companies or products (whether or not indicated as registered) does not imply any intention to infringe proprietary rights, nor should it be construed as an endorsement or recommendation on the part of the IAEA.*

Contents

1 Background 2

2 Installations 3
2.1 Miniconda3 (conda) . . . 3
2.2 Git with conda . . . 3
2.3 Required Libraries with conda . . . 3
2.3.1 Automatically (slower) . . . 4
2.3.2 Manually (faster) . . . 4

3 Running a Jupyter Notebook 5

4 Editing the Configuration File 6

5 Running the RCNV_seq-template Jupyter Notebook 8

6 References 9

1 Background

[DRAFT]

Copy number variation (CNV) analysis using CNV-seq, R, Jupyter Notebooks, Miniconda3, and Git.

Note: This is not an official IAEA publication but is made available as working material. The material has not undergone an official review by the IAEA. The views expressed do not necessarily reflect those of the International Atomic Energy Agency or its Member States and remain the responsibility of the contributors. The use of particular designations of countries or territories does not imply any judgement by the publisher, the IAEA, as to the legal status of such countries or territories, of their authorities and institutions or of the delimitation of their boundaries. The mention of names of specific companies or products (whether or not indicated as registered) does not imply any intention to infringe proprietary rights, nor should it be construed as an endorsement or recommendation on the part of the IAEA.

---

2 Installations

Before installing any necessary software, it is recommended to check if the computer is running 32-bit or 64-bit for downloading Miniconda3. Run the following to verify the system:

```
$ uname -m
```

2.1 Miniconda3 (conda)

Download the Miniconda3, or simply "conda", installer:

- Miniconda3 installer for Linux

Run the downloaded installer (for a 64-bit system):

```
$ bash Miniconda3-latest-Linux-x86_64.sh
```

Open a new terminal window for conda to take effect. Verify the installation in new terminal window with:

```
$ conda list
```

2.2 Git with conda

Git will be installed first to clone locally (download a copy to your local computer) the CNV-seq repository from GitHub. To do so, run the following:

```
$ conda install -c anaconda git
```

After the installation, clone the CNV-seq repository to the local computer in the desired directory.

```
$ git clone https://github.com/amora197/copy-number-analysis.git
```

Verify that the installation is complete by listing the files in the directory.

```
$ ls -l
```

A folder called **copy-number-analysis** should be listed in the directory.

2.3 Required Libraries with conda

CNV-seq has multiple dependencies, listed below:

- Git
- R
  - configr
  - ggplot2
  - BiocManager
  - Bioconductor-GenomicAlignments
- Jupyter Notebook

- IRkernel

There are two ways to install the rest of the necessary libraries to run CNV-seq: automatically or manually. The former is slower, providing a long coffee break while the conda installations run. The latter proves a faster way to get the tool up-and-running.

2.3.1 Automatically (slower)

Two YAML files, **environment.yml** and **libraries.yml**, are provided to automatically create a conda environment and install the dependent libraries. The former creates the conda environment, along R, Jupyter Notebook, and the R-kernel in Jupyter. The latter installs dependent R libraries. Run **environment.yml**:

```
$ conda env create --file envs/environment.yml
```

Once done, the created environment can be verified running:

```
$ conda env list
```

Activate the created environment (**cnv-seq**) and run **libraries.yml**:

```
$ conda activate cnv-seq
$ conda env update --file envs/libraries.yml
```

Once done, all the necessary packages should be installed. This can be verified with:

```
$ conda list
```

2.3.2 Manually (faster)

To manually create and activate an environment, run:

```
$ conda create --name cnv-seq
```

Once done, the created environment can be verified running:

```
$ conda env list
```

Activate the virtual environment with:

```
$ conda activate cnv-seq
```

Start running the installations of the necessary libraries, paying attention to the prompts for each one:

```
$ conda install -c conda-forge r-base=4.0
$ conda install -c anaconda jupyter
$ conda install -c r r-irkernel
$ conda install -c conda-forge r-biocmanager
$ conda install -c bioconda bioconductor-genomicalignments
$ conda install -c pcgr r-configr
$ conda install -c r r-ggplot2
```

Once done, all the necessary packages should be installed. This can be verified with:

```
$ conda list
```

3 Running a Jupyter Notebook

To access the Jupyter Notebooks, run:

```
$ jupyter notebook
```

This command will start a Jupyter Notebook session inside the directory the command is run. The user can navigate between directories, visualize files, and edit files in the browser by clicking on directories or files, respectively.

Look for the directory **copy-number-analysis** and click on it. Click on **jupyter-notebooks** directory, which contains four directories and two Jupyter Notebooks. Here is a breakdown of each:

- *config*:
  - directory containing configuration files
- *helper-functions*:
  - directory containing R scripts with functions to calculate and plot CNVs
- *images*:
  - directory that will contain output CNV plots after running CNV-seq
- *tab-files*:
  - directory that will contain two types of output tab-delimited files:
    - hits used to calculate CNVs
    - CNVs per chromosome per comparison
- two Jupyter Notebooks:
  - RCNV-seq-example.ipynb
    - example analysis of sorghum
  - RCNV-seq-template.ipynb
    - template for the user

Note: Jupyter lets the user duplicate, rename, move, download, view, or edit files in a web browser. This can be done by clicking the box next to a file and choosing accordingly.

4 Editing the Configuration File

In order to run the CNV-seq Jupyter Notebook, the user needs to feed it with a configuration file (**config-CNVseq.yml**) that specifies the paths to the bam files, comparisons to be done, chromosomes to analyze, and parameter definitions for calculating and plotting CNVs.

The configuration file **config-CNVseq.yml** can be found in the **copy-number-analysis/jupyter-notebooks/config** directory.

Note: The user needs to edit **config-CNVseq.yml** to point towards bam/bed files; specify comparisons and chromosomes to analyze; and define the parameters to calculate/plot CNVs.

Two example configuration files are provided (**example1-config-CNVseq-coffee.yml** and **example2-config-CNVseq-sorghum.yml**). The configuration file **config-CNVseq.yml** contains multiple parameters to be defined by the user:

- *paths*:
  - sample names and their respective paths to **.bam** files
  - samples can be named as desired but the sample name must be repeated after the colon and prefixed with a & sign
  - the & prefix sign is used to reference the sample's path in different places of the same configuration file
  - example use:

```
paths:
  mysample: &mysample /home/john/bam_files/mysample.bam
  XYZ-123: &XYZ-123 /home/john/bam_files/XYZ-123.bam
  potato95: &potato95 /home/john/bam_files/potato95.bam
```

- *bed_path*:
  - path to bed file if using varying window sizes
- *comparisons*:
  - comparison names with respective control and mutant samples per comparison
  - each comparison can be named as desired
  - the sample names to be used as *control* and *mutant* need to be prefixed by a * sign
  - the * prefixed sign is used to extract the sample's path defined in the *paths* section
  - example:

```
comparisons:
  comparison-1:
    control: *mysample
    mutant: *potato95
  a-different-comparison-278asd:
    control: *mysample
    mutant: *XYZ-123
```

- *chromosomes*:
  - list of chromosome names to analyze separated by commas
  - chromosome names can be extracted from a bam file's header
- *parameters*:

- parameters used to create window sizes, thresholds, and plots
- the parameter defaults are provided

5 Running the RCNV_seq-template Jupyter Notebook

Note: It is recommended to duplicate the **RCNV-seq-template** notebook and then renaming the copy before doing any edits to the notebook.

In the **copy-number-analysis/jupyter-notebooks** directory, click on **RCNV-seq-template** and a new tab will open the notebook.

The notebook contains cells that are populated by text or code. Instructions are provided in the notebook to guide the user.

The notebook consists of 5 sections:

1. User Input (MANDATORY)
2. Installing Required Libraries (optional)
3. Loading Required Libraries (MANDATORY)
4. CNV Calculations and Plotting (MANDATORY)
5. Plotting Specific Window of One Chromosome (optional)

To run a cell, click on the corresponding cell and press **Ctrl + Enter** or **Shift + Enter**.

6 References

BMC Bioinformatics Publication:

- CNV-seq, a new method to detect copy number variation using high-throughput sequencing [LINK]

GitHub repositories:

- hliang/cnv-seq
- Bioconductor/copy-number-analysis
- amora197/copy-number-analysis