barseqcount

Release 0.1.5

Damien Marsic

May 16, 2023

CONTENTS:

1 Overview **1**

2 Latest news **3**

3 Installation **5**

4 Update **7**

5 Usage **9**
- 5.1 barseqcount count . . . 9
- 5.2 barseqcount analyze . . . 11

6 Functions **13**
- 6.1 main() . . . 13
- 6.2 count(args) . . . 13
- 6.3 analyze(args) . . . 13
- 6.4 anaconf(fname,args) . . . 13
- 6.5 countconf(fname,args) . . . 14
- 6.6 find_bc(l,templ,bcr,cl,ctempl,cbcr) . . . 14
- 6.7 fb(l,templ,i,bcr) . . . 14
- 6.8 maxmatch(sample,target,probe) . . . 15
- 6.9 override(func) . . . 15
- 6.10 version() . . . 15

CHAPTER

ONE

OVERVIEW

`barseqcount` is a tool for the analysis of DNA barcode sequencing experiments, typically used for biodistribution studies. It has 2 main commands, `count` and `analyze`. `barseqcount count` counts barcodes combinations from NGS read files. `barseqcount analyze` analyzes the data and displays the results as a collection of plots.

Source code:
https://github.com/damienmarsic/barseqcount

Python package:
https://pypi.org/project/barseqcount/

Bioconda package:
https://bioconda.github.io/recipes/barseqcount/README.html

Bug report / feature requests:
https://github.com/damienmarsic/barseqcount/issues/new/choose

CHAPTER

TWO

LATEST NEWS

2023-02-23: update to 0.1.4. Improved detection of errors in the count configuration file. 2023-05-16: update to 0.1.5. Updated some function calls in order to be compatible with latest version of dmbiolib.

CHAPTER

THREE

INSTALLATION

It is recommended to install `barseqcount` as a bioconda package under a Python 3 conda environment. If you don't have such environment, first install Miniconda3 from here (exists for Linux, MacOSX and Windows).

To install the `barseqcount` bioconda package, at the conda prompt type:

```
conda install -c bioconda barseqcount
```

`barseqcount` can be also installed using pip:

```
pip install barseqcount
```

If using pip intall, note that dependencies (numpy, matplotlib and regex) might need to be installed individually if not already present.

Windows conda users: note that Bioconda does not support Windows and does not allow the automatic creation of a bat file. At the conda prompt, please type the following:

```
(echo @echo off & echo python -m barseqcount %%*) > %CONDA_PREFIX%\Scripts\barseqcount.
↪bat
```

Windows non-conda users: follow the instruction above except that you need to replace `%CONDA_PREFIX%` with the full path to a directory that is part of your PATH system variable.

Linux and MacOSX non-conda users: create a file named `barseqcount` in a directory that is part of your PATH environment variable, with the following content:

```
#!/bin/sh
python -m barseqcount $@
```

Then type the following to make the file executable:

```
chmod +x barseqcount
```

CHAPTER

FOUR

UPDATE

To update `barseqcount` to the latest version:

Using conda:

```
conda update barseqcount
```

or even better:

```
conda update --all
```

Using pip:

```
pip install barseqcount -U
```

CHAPTER

FIVE

USAGE

```
barseqcount [-h] [-v] {count,analyze} ...

or

python -m barseqcount [-h] [-v] {count,analyze} ...
```

The optional arguments `-h/--help` and `-v/--version` allow to show a help message and version information respectively.

Positional arguments `count` and `analyze` are the two commands that `barseqcount` can run. The former processes read files, collects barcode data and saves a barcode distribution file. The latter analyzes the barcode distribution data and displays the results as a collection of plots.

`barseqcount count` has optional arguments `-c/--configuration_file` and `-n/--new`. The `-c` argument (followed by a file name) specifies which configuration file to use, or which to create if it does not exist yet. The `-n` argument allows to ignore an existing configuration file and to create a new one.

`barseqcount analyze` has the same `-c` and `-n` arguments, but also a third one: `-f/--file_format`, allowing to chose a file format to save the plots individually. If the `-f` argument is not used, then all plots will be saved in a single multipage pdf file.

5.1 barseqcount count

5.1.1 Barcodes

Although `barseqcount count` works with any number of barcodes, typically, each read contains one variant barcode, as well as possibly one or more sample barcodes. The sample barcodes are introduced by barcoded primers which are used to amplify the variant barcode sequence. In the case of Illumina sequencing, if indexes (barcodes introduced by PCR or ligation) are used for different samples, demultiplexing is typically performed by the NGS provider, in which case multiple files will be provided (1 file or 1 file pair per index), with their reads only containing the variant barcode. In that case, and in any case where multiple read files are present, file name prefixes will be used as sample barcodes.

5.1.2 Paired-end reads

Any common read file format (fasta or fastq, either uncompressed or gzipped) can be read by `barseqcount count`. However, paired-end reads must be merged before use. Many merger programs can be used, for example `NGmerge` from the `ngmerge` package is recommanded. Example of installation in a conda environment:

```
conda install ngmerge
```

Example of merging read files Reads_1.fq.gz and Reads_2.fq.gz into Merged_reads.fq.gz:

```
NGmerge -1 Reads_1.fq.gz -2 Reads_2.fq.gz -o Merged_reads.fq.gz
```

If you get the error message `Error! Quality scores outside of set range`, then add the `-u 41` and `-g` arguments (see NGmerge documentation for more information):

```
NGmerge -1 Reads_1.fq.gz -2 Reads_2.fq.gz -o Merged_reads.fq.gz -u 41 -g
```

5.1.3 Configuration file

If no configuration file (barseqcount_count.conf by default or any file name entered after the `-c` argument) exists in the current directory, or if the `-n` argument is used, the command `barseqcount count` will create a new configuration file (named barseqcount_count.conf by default if the `-c` argument is not used). If the `-n` argument is used, the existing configuration file will be renamed by adding a unique string of numbers before the file extension. The configuration file needs to be edited by the user and each section needs to be filled out with appropriate information before it can be used. Some sections are populated automatically if the program detects the most plausible content: Project name (working directory name), Read file(s) (any gzipped files present in the current directory) and Template sequence (if a single fasta file is present in the current directory). All other sections must be populated by the user according to the instructions provided within the configuration file. When the configuration file is ready, running `barseqcount count` will open it and check its contents. If errors are detected, the program will exit with an explanatory message. Otherwise, it will proceed with processing the read file(s).

5.1.4 Error correction

Whether error correction is performed is determind automatically by `barseqcount count` by analyzing the barcode sequences in the configuration file. If all barcodes within the same barcode location differ by at least 3 nucleotide substitutions from any other barcode, then single substitution error correction will be activated for that location, which means that if an unknown barcode is obtained which can be converted to a know barcode by a single substitution, it will be converted to that known barcode. The other type of error correction corrects for indels within homopolymers of the sequences surrounding the barcode and for homopolymer insertions within the barcode sequence, and is only activated if homopolymers are absent from all expected barcodes in the barcode locationand if the ends of the barcodes are different from the nucleotide next to them.

5.1.5 Read file processing

Barcodes combinations are collected, error corrected when applicable, converted to variant names and sample names whenever possible, and saved into a barcode distribution csv file, which can later be used by the `barseqcount analyze` program. A result summary is also displayed and added to a report file.

5.2 barseqcount analyze

5.2.1 Configuration file

If no configuration file (barseqcount_analyze.conf by default or any file name entered after the `-c` argument) exists in the current directory, or if the `-n` argument is used, the command `barseqcount analyze` will create a new configuration file (named barseqcount_analyze.conf by default if the `-c` argument is not used). If the `-n` argument is used, the existing configuration file will be renamed by adding a unique string of numbers before the file extension. The configuration file will only be created if a count report file can be found in the current directory (if more than one is present, the most recent will be used), from which relevant information (such as the barcode distribution file name and the definitions) will be used to prepopulate some sections of the configuration file. The configuration file needs to be edited by the user and each section needs to be filled out with appropriate information before it can be used. Most sections are actually populated automatically by `barseqcount analyze` (but should still be edited by the user according to their preferences) except for the global genome and expression titers which need to be entered manually (although simplified analysis can still be performed if these sections are empty). When the configuration file is ready, running `barseqcount analyze` will open it and check its contents. If errors are detected, the program will exit with an explanatory message.

5.2.2 Analysis

`barseqcount analyze` analyzes the data from the barcode distribution file according to the settings in the configuration files, and displays the results as a collection of configurable bar plots and heat maps. For each plot, the data is also saved as a csv file, so the user also has the option of creating their own plots.

5.2.3 Variant mix composition

If a variant mix exists in the sample definitions, its composition is displayed as a bar plot, with the variants in the x-axis and the deviation from equimolar frequency in the y-axis. If some variants have a frequency below a threshold defined in the configuration file, they will be removed from all subsequent analyses.

5.2.4 Global read count per sample

Total read counts per sample are displayed as a bar plot, allowing to verify that each sample is represented by a sufficient number of reads.

5.2.5 Global variant enrichment

Enrichment of each variant between the variant mix (if present) and each sample is displayed as a heat map, with colors indicating enrichment factors in Log scale. If mix is absent, equimolar variant mix is assumed.

5.2.6 Global biodistributions

If both Global titers and Combine data sections exist (and are not empty) in the configuration file, a global biodistribution plot will be displayed for each group in the Combine data section.

5.2.7 Detailed biodistributions

If the Combine data section exists and is not empty, detailed biodistribution plots will be displayed for each group in the section. In these plots, data from biological replicates are combined. If Global titers exist in the configuration file, biodistribution is expressed as titers in the appropriate unit, otherwise it is shown as enrichment factors. Each group is represented by two plots: a heat map and a bar plot. In the bar plots, individual data points corresponding to biological replicates can be overlaid in a choice of shapes, and error bars can be shown as range, standard deviation or standard error, according to settings in the configuration file.

CHAPTER

SIX

FUNCTIONS

Many of the functions used in `barseqcount` are also used in other projects and have been included in the dmbiolib package.

6.1 main()

The `main()` function uses `argparse` to read and process the command line arguments.

6.2 count(args)

- args: optional arguments following the `count` command

Creates a new configuration file if none exists or if -n/–new argument is present. Otherwise, processes the read file(s) according to instructions in the configuration file. Saves the barcode distribution in a csv file, and a report in a txt file.

6.3 analyze(args)

- args: optional arguments following the `analyze` command

Creates a new configuration file if none exists or if -n/–new argument is present. Otherwise, analyzes the data according to instructions in the configuration file. Creates a series of plots and saves results in csv files.

6.4 anaconf(fname,args)

- fname: name of the configuration file to be created
- args: arguments

Creates a configuration file for the `barseqcount analyze` program

6.5 countconf(fname,args)

- fname: name of the configuration file to be created
- args: arguments

Creates a configuration file for the `barseqcount count` program

6.6 find_bc(l,templ,bcr,cl,ctempl,cbcr)

- l: read
- templ: template
- bcr: dictionary containing information about barcode locations and error correction
- cl: compressed read (using compress function from `dmbiolib`)
- ctempl: compressed template
- cbcr: dictionary containing information about barcode locations based on compressed template

Identifies all barcodes in a read and perfoems error correction as appropriate.

Returns a dictionary of barcode positionsa / barcode sequences, a number indicating whether the read was corrected (>0) or not (0), and a list containing error correction counters.

6.7 fb(l,templ,i,bcr)

- l: read (nucleotide sequence)
- templ: template
- i: barcode index
- bcr: dictionary containing information about barcode locations and error correction

Determines bacode sequence by mapping read sequence to template, using information about barcode locations and error correction.

Returns barcode sequence.

6.8 maxmatch(sample,target,probe)

- sample: nucleotide sequence of primer
- target: nucleotide sequence of template
- probe: initial probe size

Determines largest part of the primer that matches the template.

Returns (a,x,b,y) where a is the maximum extent of the primer from its right end that matches the template, b is the maximum extent of the primer from its left end that matches the template, x is the template index of sample[-a:], and y is the template index of sample[:b].

6.9 override(func)

Allows argparse to handle the `-v/--version` argument correctly.

6.10 version()

Displays version and other information:

```
python -m barseqcount -v
  Project: barseqcount
  Version: 0.1.4
  Latest update: 2023-02-23
  Author: Damien Marsic, damien.marsic@aliyun.com
  License: GNU General Public v3 (GPLv3)
```

- search