
barcode

Release 1.0.2

Jeroen F.J. Laros

Nov 21, 2021

CONTENTS:

1	Introduction	3
2	Installation	5
2.1	From source	5
3	Usage	7
4	Command Line Interface	9
4.1	Positional Arguments	9
4.2	Named Arguments	9
4.3	Sub-commands:	9
5	Library	11
6	API documentation	13
7	Contributors	15
	Python Module Index	17
	Index	19

Barcode is a program for the design and validation of sets of sequencing barcodes.
Please see [ReadTheDocs](#) for the latest documentation.

INTRODUCTION

Barcodes are used in NGS to tag samples before pooling. After sequencing, these barcodes are used to *demultiplex* the data, thereby assigning the reads to the originating sample.

The key aspect of a good set of barcodes is robustness against read errors. One read error should not be able to transform one barcode into another. This requirement can be met by selecting barcodes in such a way that the *edit distance* between any pair of barcodes is larger than one. An additional desired property is the ability to *correct* read errors. This can be done by increasing the minimal edit distance between barcodes to at least three. If one read error occurs, the sequenced barcode will have a distance of one to the original barcode and a minimum distance of two to any of the other barcodes. If the read error is high, the minimum edit distance should be increased to a higher (odd) number.

For some sequencers it is important that mononucleotide stretches in barcodes are below a minimum length. An additional filter can be used to remove these barcodes.

INSTALLATION

The software is distributed via [PyPI](#), it can be installed with `pip`:

```
pip install barcode
```

2.1 From source

The source is hosted on [GitHub](#), to install the latest development version, use the following commands.

```
git clone https://github.com/jfjlaros/barcode.git
cd barcode
pip install .
```


USAGE

The barcode program has two subcommands; one for the creation of a set of barcodes and one for the validation of an existing set of barcodes.

To make a set of barcodes and write this set to a file named `barcodes.txt`, use the following command:

```
barcode make barcodes.txt
```

`barcodes.txt` will now contain a list of barcodes that all have length 8, and no barcode will contain a mononucleotide stretch longer than 2.

The length of the barcodes can be controlled with the `-l` parameter, the minimum edit distance is controlled with the `-d` option and the maximum mononucleotide stretch length can be set with the `-s` option. So if we want to make a list of barcodes of length 10, a minimum edit distance of 5 (allowing for the correction of 2 read errors) and a maximum mononucleotide stretch of 1, we use the following command:

```
barcode make -d 5 -l 10 -s 1 barcodes.txt
```

To verify a list of existing barcodes, use the command:

```
barcode test barcode.txt
```

This will check the distance between any pair of barcodes and will tell you how many barcodes violate the distance constraint. Again, the minimum edit distance can be set with the `-d` parameter.

Additionally, a good set of barcodes can be extracted by providing an output file via the `-o` option:

```
barcode test -o good_barcodes.txt barcodes.txt
```


COMMAND LINE INTERFACE

Design and test NGS barcodes.

```
usage: barcode [-h] [-v] {make,test} ...
```

4.1 Positional Arguments

subcommand Possible choices: make, test

4.2 Named Arguments

-v show program's version number and exit

4.3 Sub-commands:

4.3.1 make

Make a set of barcodes, filter them for mononucleotide stretches and for distances with other barcodes.

```
barcode make [-h] [-d DISTANCE] [-H] [-l LENGTH] [-s STRETCH] OUTPUT
```

Positional Arguments

OUTPUT output file

Named Arguments

- d** minimum distance between the barcodes (int default=3)
Default: 3
- H** use Hamming distance
Default: False
- l** length of the barcodes (int default=8)
Default: 8
- s** maximum mononucleotide stretch length (int default=2)
Default: 2

4.3.2 test

Test a set of barcodes.

```
barcode test [-h] [-d DISTANCE] [-H] [-o OUTPUT] INPUT
```

Positional Arguments

- INPUT** input file

Named Arguments

- d** minimum distance between the barcodes (int default=3)
Default: 3
- H** use Hamming distance
Default: False
- o** list of good barcodes

Copyright (c) Jeroen F.J. Laros <J.F.J.Laros@lumc.nl>

LIBRARY

Barcode design via the library is done in three steps. First obtain the full set of permutations with the `all_barcode`s function:

```
>>> from barcode import all_barcode, filter_distance, filter_stretches
>>>
>>> # Generate all barcodes of length 2.
>>> all_barcode(2)
['AA', 'AC', 'AG', 'AT', 'CA', 'CC', 'CG', 'CT', 'GA', 'GC', 'GG', 'GT', 'TA',
'TC', 'TG', 'TT']
```

The resulting list can be filtered with the `filter_distance` and `filter_stretches` functions:

```
>>> # Filter all barcodes of length 3 for a minimal edit distance of 3.
>>> filter_distance(all_barcode(3), 3)
['AAA', 'CCC', 'GGG', 'TTT']
>>>
>>> # Filter all barcodes of length 2 for mononucleotide stretches of length
>>> # longer than 1.
>>> filter_stretches(all_barcode(2), 1)
['AC', 'AG', 'AT', 'CA', 'CG', 'CT', 'GA', 'GC', 'GT', 'TA', 'TC', 'TG']
```

For the best result, apply the `filter_stretches` function before applying the `filter_distance` function:

```
>>> # Make a set of barcodes of length 3, having no mononucleotide stretches
>>> # and a minimum edit distance of 3.
>>> filter_distance(filter_stretches(all_barcode(3), 1), 3)
['ACA', 'CGC', 'GAG']
```


API DOCUMENTATION

`barcode.barcode.all_barcodes(length)`

Generate all possible barcodes of a certain length.

Parameters `length` (*int*) – Length of the barcodes.

Returns list List of barcodes.

`barcode.barcode.filter_distance(barcodes, min_dist, distance=<built-in function distance>)`

Filter a list of barcodes for distance to other barcodes.

Parameters

- **barcodes** (*list*) – List of barcodes.
- **min_dist** (*int*) – Minimum distance between the barcodes.
- **distance** (*function*) – Distance function.

Returns list List of barcodes filtered for distance to other barcodes.

`barcode.barcode.filter_stretches(barcodes, max_stretch)`

Filter a list of barcodes for mononucleotide stretches.

Parameters

- **barcodes** (*list*) – List of barcodes.
- **max_stretch** (*int*) – Maximum mononucleotide stretch length.

Returns list List of barcodes filtered for mononucleotide stretches.

CONTRIBUTORS

- Jeroen F.J. Laros <J.F.J.Laros@lumc.nl> (Original author, maintainer)

Find out who contributed:

```
git shortlog -s -e
```


PYTHON MODULE INDEX

b

`barcode.barcode`, 13

INDEX

A

`all_barcode`s() (*in module barcode.barcode*), 13

B

`barcode.barcode`
module, 13

F

`filter_distance`() (*in module barcode.barcode*), 13
`filter_stretches`() (*in module barcode.barcode*), 13

M

module
`barcode.barcode`, 13