

Barcodes Designer (v1.0)

Generate and optimize barcode sets

Description

Barcodes Designer is a toolkit that has two major components for creating and optimizing barcode sets. These short oligonucleotide sequences are often used as unique molecular identifiers (UMIs) or sample indices in next-generation sequencing workflows, for example, to improve multiplexing or to simplify qualitative and quantitative analysis.

The optimization of the barcode sets is based on a renowned genetic algorithm (NSGA-II [1]) with the aim to improve barcode count, nucleotide diversity and minimal pairwise distance.

The parameters of the barcode set generation and optimization can further be modified by the user.

Features

- Generate a barcode set from scratch based on your specific lab set-up (e.g., sequencing platform)
- Import files containing barcodes
- Optimize subsets of given barcodes considering minimal pairwise distance, color balance while also keeping subset size high
- Export and view the obtained barcode sets

Prerequisites

To run and customize the procedures, our toolkit includes both a command-line interface and a user-friendly JavaFX user interface. The toolkit is implemented in Java 16 for both interfaces.

If you prefer to utilize the GUI, keep in mind that JavaFX is no longer included in the recent Java SDKs, thus you'll need to specify the JavaFX modules as mentioned in the **Usage** section.

- Java 16
- JavaFX SDK (only for GUI)

*If you're looking for an easier way to open the user interface, we've got you covered with our **Quick Start** scripts. With **Quick Start**, there is no need to install Java or JavaFX, also you don't have to worry about finding the correct module paths.*

Usage

Java 16 is required to run our toolkit; Additionally, the GUI requires JavaFX. The executable jar file is available for download. It is not essential to set this version as your default (See **Other Info**).

Skip to **Quick Start** if you do not have the required versions installed.

Command Line Interface

Mode 1: To run the default generation of a barcode set, open a terminal inside the directory containing the jar file, and execute:

```
$ java -cp barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain --type  
"generate"
```

Optional parameters are:

```
$ java -cp barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain --type
"generate"
[-nbarcodes, -nb] [-length, -l] [-pattern, -p] [-gcmin] [-gcmax]
[-hamming={true|false}, -h={true|false}] [-mindist, -md]
[-popsize, -ps] [-niter, -ni] [-nrun, -nr]
[-outtype={"text"|"json"}, -ot={"text"|"json"}] [-outputfile=<file2>, -of=<file2>]
[-distMetric={"hamming"|"levenshtein"}, -dm={"hamming"|"levenshtein"}] [-parallel, -par]
[-quiet, -q]
```

Mode 2: For the default optimization of subsets from a newly generated barcode set, open a terminal inside the directory containing the jar file, and execute:

```
$ java -cp barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain --type
"select"
```

Here, the default parameters generate an initial barcode set from which subsets are optimized.

Optional parameters are:

```
$ java -cp barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain
--type "select"
[[-file=<file1>, -f=<file1>], [-nbarcodes, -nb] [-length, -l] [-pattern, -p]
[-gcmin] [-gcmax] [-hamming, -h] ]
[-popsize, -ps] [-niter, -ni] [-nrun, -nr]
[-outtype={"text"|"json"}, -ot={"text"|"json"}] [-outputfile=<file2>, -of=<file2>]
[-distMetric={"hamming"|"levenshtein"}, -dm={"hamming"|"levenshtein"}]
[-parallel, -par] [-balancecolors, -bc]
[-quiet, -q]
```

Please refer to **Computational Settings** for more in-depth information about these parameters and their default values.

Graphical User Interface

The GUI requires JavaFX, hence the paths of the JavaFX modules have to be stated.

If you have **JavaFX SDK and Java 16 installed**, to open the GUI, run:

```
$ java --module-path <PATH/TO/JAVAFX/SDK/lib> --add-
modules=javafx.controls,javafx.fxml,javafx.graphics -jar barcodesDesigner_v1.0.jar
```

Example:

```
$ java --module-path Documents/javafx-sdk-16/lib --add-
modules=javafx.controls,javafx.fxml,javafx.graphics -jar barcodesDesigner_v1.0.jar
```

Quick Start (GUI)

We provide easy scripts to setup Java 16 and JavaFX without having to install these versions on your computer. The scripts download the necessary files from <https://jdk.java.net/archive/> and <https://gluonhq.com/products/javafx/>. The quick start for Linux encompasses the manual download of the 2 folders with an easy run script.

GUI on MacOS

1) Obtain the directory "quickStartMacOS". Initially, this directory should have the following structure:

```
.
├─ quickStartMacOS          #
│   ├── barcodesDesigner_v1.0.jar # The jar of the tool
│   ├── setup.sh              # Run this to setup Java and JavaFX
└─ └─ run.sh                  # Run this to open GUI
```

Open the terminal/bash inside quickStartMacOS, and run the following scripts:

2) `$./setup.sh` This script should download the files and organize them under a folder `java` within quickStartMacOS. This will download ~415Mb and should not take longer than 2 minutes (depending on your internet connection.). 3) `$./run.sh` This script should finally open the user interface.

GUI on Windows

1) Obtain the directory "quickStartWindows". Initially, this directory should have the following structure:

```
.
├─ quickStartWindows
│   ├── barcodesDesigner_v1.0.jar # The jar of the tool
│   ├── setup.bat                 # Run this to setup Java and JavaFX
└─ └─ run.bat                     # Run this to open GUI
```

Open the CMD inside quickStartWindows, and run the following scripts: 2) `$ setup.bat` This script should download the files and organize them under a folder `java` within quickStartMacOS. This will download ~450Mb and should not take longer than 2 minutes (depending on your internet connection.). 3) `$ run.bat` This script should finally open the user interface.

GUI on Linux

1) Get **JavaFX**

Download the JavaFX SDK 16 from https://download2.gluonhq.com/openjfx/16/openjfx-16_linux-x64_bin-sdk.zip and extract it. Put the resulting folder `javafx-sdk-16` inside the folder `quickStartLinux/java`.

2) Get **Java 16**

Similarly, download the openJDK 16 from https://download.java.net/java/GA/jdk16/7863447f0ab643c585b9bdebf67c69db/36/GPL/openjdk-16_linux-x64_bin.tar.gz, extract it and also put the resulting folder `jdk-16` in `quickStartLinux/java`.

This should result in the following structure:

```
.
├─ quickStartLinux
│   ├── barcodesDesigner_v1.0.jar # ... already exists
│   ├── java                      # ... already exists
│   │   ├── javafx-sdk-16        # Obtained by extracting 1st zip; put it here
│   │   └─ jdk-16                # Obtained by extracting 2nd zip; put it here
└─ └─ run.sh                     # Run this to start GUI
```

3) Open a bash inside `quickStartLinux` and execute:

```
$ ./run.sh
```

This script should open the user interface.

Command Line Interface

1. Download the openJDK 16 for your operating system:

- **MacOS:**

https://download.java.net/java/GA/jdk16/7863447f0ab643c585b9bdebf67c69db/36/GPL/openjdk-16_osx-x64_bin.tar.gz

- **Linux:**

https://download.java.net/java/GA/jdk16/7863447f0ab643c585b9bdebf67c69db/36/GPL/openjdk-16_linux-x64_bin.tar.gz

- **Windows:**

https://download.java.net/java/GA/jdk16/7863447f0ab643c585b9bdebf67c69db/36/GPL/openjdk-16_windows-x64_bin.zip

2. Extract the downloaded directory - you should obtain the folder `jdk-16`.
3. Put this folder in the same directory as the `barcodeDesigner_v1.0.jar`. Your folder structure should resemble the following:

```
.
├─ theFolder                               #
│   └─ barcodesDesigner_v1.0.jar           # ... already exists
└─ jdk-16 (or jdk-16.jdk on MacOS)        # Obtained by extracting 2nd zip; put it here
```

4. Open a terminal/bash/CMD inside the top directory (here: `theFolder`), and execute the jar in the preferred manner (See **Usage - Command Line Interface**) Instead of `$ java (...)`, specify the java executable inside of the downloaded folder.

Example (on MacOS):

```
$ jdk-16.jdk/Contents/Home/bin/java -cp barcodesDesigner_v1.0.jar
main.code.commandLineUse.CommandLineMain --type "select"
```

Example (on Linux):

```
$ jdk-16/bin/java -cp barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain --
type "select"
```

Example (on Windows):

```
$ jdk-16\bin\java.exe -cp barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain
--type "select"
```

Other info

Java version:

Run the following in your terminal/bash/CMD:

```
$ java -version
```

You should see a specification of your installed default Java version, e.g.,:

```
$ java version "16.0.2" 2021-07-20
Java(TM) SE Runtime Environment (build 16.0.2+7-67)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.2+7-67, mixed mode, sharing)
```

If Java 16 is not your default version, you can simply specify the installed JDK 16 on your machine.

Instead of `$ java (...)`, be sure to point to the executable of Java. In the following are the three default paths depending on the operating system.

Example (on MacOS):

```
$ /Library/Java/JavaVirtualMachines/jdk-16.jdk/Contents/Home/bin/java -cp
barcodesDesigner_v1.0.jar main.code.commandLineUse.CommandLineMain --type "select"
```

Example (on Linux):

```
$ /usr/lib/jvm/jdk-16/bin/java -cp barcodesDesigner_v1.0.jar
main.code.commandLineUse.CommandLineMain --type "select"
```

Example (on Windows):

```
$ C:\Program Files\Java\jdk-16\bin\java.exe -cp barcodesDesigner_v1.0.jar
main.code.commandLineUse.CommandLineMain --type "select"
```

Trouble shooting:

- Command not found, e.g.,:

```
$ zsh: command not found: java
```

- -> Either you do not have Java installed, or your JAVA_HOME (Java environment) variable is not properly set.

- A Module related error occurs, e.g.,:

```
Error occurred during initialization of boot layer
java.lang.module.FindException: Module javafx.controls not found
```

- -> Be sure to separately have the JavaFX SDK installed, since it is no longer included in Java >= 11. Also if JavaFX SDK is installed, check again for the correct module path(s).

- A `LinkageError` occurs, e.g.,:

```
$ Error: LinkageError occurred while loading main class
main.code.commandLineUse.CommandLineMain
    java.lang.UnsupportedClassVersionError: main/code/commandLineUse/CommandLineMain has
    been compiled by a more recent version of the Java Runtime (class file version 60.0),
    this version of the Java Runtime only recognizes class file versions up to 59.0
```

- -> You tried to execute BarcodesDesigner with a Java version other than 16. Here the 60.0 refers to the correct version 16, e.g., 59.0 would mean that you tried to execute the tool with Java 15.

- A parameter related error occurs, e.g.,:

```
$ Error: --type must be either "select" or "generate"!
```

- -> There is possibly a typo within the value of this parameter (here `-type`).

```
$ Error: Unknown argument: -lengthOfBarcode
```

- -> This parameter you added (here `-lengthOfBarcode`) does not exist.

Parameters

Barcode Generation

- *Barcode length*: The length of the generated barcodes (values: 1-1000; default: 12; integer)
- *Number of barcodes*: The size of the barcode set (values: 2-1000000; default: 100; integer)
- *Barcode pattern*: A user-defined pattern specifying fixed and free nucleotide positions (must match: `[ACGT_]{length}`)
- *G/C-percentage*: The range of allowed GC content of each barcode (note: specify 60 instead of 0.6 for 60%; values: 0-100 (minGC <= maxGC); default: 40 (minGC) - 60 (maxGC); integer)
- *Hamming*: Optional initialization with Hamming codes (guarantees minimal pairwise distance >= 3, reduces set of feasible barcodes)

Genetic Algorithm/Optimization

- *Evaluated solutions*: Population size (values: 10-500; default: 100; integer)
- *Number of iterations*: Generation count (values: 1-1000000; default: 1000; integer)
- *Minimal distance*: If specified, the computation ends when the minimal distance reaches a given threshold (Only in first mode)
- *Number of restarts*: Count of runs performed (May improve results - but also computation time) (values: 1-10; default: 4; integer)
- *Number of parallel streams*: Degree of parallelization of specific stages during GA (parent selection, recombination, mutation, fitness evaluation) (values: 1-10; default: 4)
- *Try to balance colors at each base position*: Option to in-/exclude color balance as third objective (only in second mode)
- *Output type*: Specifies format of output (text: each barcode is printed in newline, json: more hierarchical representation) (default: "text", string)
- *Output file*: Specifies where the file should be saved
- Quiet mode

References

[1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, April 2002, doi: 10.1109/4235.996017.

Contributing

License

Citation

Contact