VarDict Getting Started Guide

Version Number: 1.2

|  |  |
| --- | --- |
| Document Details | |
| Document Title | VarDict Getting Started Guide |
| Status / Version | Final/1.2 |
| Reference Code |  |

|  |
| --- |
| Disclaimer |
| If you are reading a copy of this document (i.e. print out of controlled electronic document or an electronic copy of a wet ink signed paper document) you may be reading an uncontrolled version. Before making updates, please ensure that you are using the latest version of this document and the associated template. |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Approval / Review | | | |
| Criteria | Name & Role | Signature | Date |
| Author | Viktor Kirst |  | 11-02-2015 |
| Approved |  |  |  |
| Approved |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Version Control | | | |
| Version | Date | Description of Changes | Author |
| 0.1 | 11/02/15 | Initial version | Viktor Kirst |
| 1.0 | 12/02/15 | Review and formatting | Liudmila Simonova |
| 1.1 | 20/02/15 | Added two more options into “Build Instruction” section | Viktor Kirst |
| 1.2 | 27/03/15 | Added a description of default one-thread mode | Viktor Kirst |

| Referenced Documents | | | |
| --- | --- | --- | --- |
| No. | Document Title and ID | Version | Location |
|  |  |  |  |

**Table of Contents**

1. Source Location 3

2. Build Instruction 3

1.1. Distribution Package Structure 3

1.2. Third-Party Libraries 3

3. Launch Instruction 4

# Source Location

The VarDict source code is located at <https://github.com/AstraZeneca-NGS/VarDictJava>.

To load the project, execute the following command:

git clone --recursive https://github.com/AstraZeneca-NGS/VarDictJava.git

**Note:** The original VardDict project is placed in this repository as a submodule and it's contents can be found in the VarDict sub-directory of the VarDictJava working folder. So when you use teststrandbias.R and var2vcf\_valid.pl, you have to add the VarDict prefix: VarDict/teststrandbias.R and VarDict/var2vcf\_valid.pl (see the example in Section 3).

# Build Instruction

The following prerequisites are necessary to build the VarDict project:

* JDK1.7 (or higher)
* Internet access

The project uses Gradle (<http://gradle.org/>) and already includes a gradlew script.

To build the project, in the root folder of the project, run the following command:  
./gradlew clean installApp

To build the project as a distribution archive, in the build/distributions folder, run the following command:

./gradlew clean distZip

This command builds the project and creates an archive of the distribution package in the folder.

To generate Javadoc, in the build/docs/javadoc folder, run the following command:

./gradlew clean javadoc

## Distribution Package Structure

When the build command completes successfully, the build/install/VarDict folder contains the distribution package.

The distribution package has the following structure:

bin/ - contains the launch scripts

lib/ - has the jar file that contains the compiled project code and the jar files of the third-party libraries that the project uses.

You can move the distribution package (the content of the build/install/VarDict folder) to any convenient location.

## Third-Party Libraries

Currently, the project uses the following third-party libraries:

* **JRegex** (<http://jregex.sourceforge.net>, BSD license) is a regular expressions library that is used instead of the standard Java library because its performance is much higher than that of the standard library.
* **Commons CLI** (<http://commons.apache.org/proper/commons-cli>, Apache License) – a library for parsing the command line.
* **HTSJDK** (<http://smtools.github.io/htsjdk/>) is an implementation of a unified Java library for accessing common file formats, such as SAM and VCF.

# Launch Instruction

The following prerequisites are necessary to launch the program:

* Java 1.7 (or higher)

To launch VarDict:

* Run the command <distribution package path>/bin/VarDict or
* Run just VarDict if <distribution package path>/bin/ is in path

The work of the program is identical to the function of the original Perl script and thus can replace it.

For example, the original launch command (from the ***readme*** file of the perl version) is:

vardict -G /path/to/hg19.fa -f 0.01 -N sample\_name -b /path/to/my.bam -z -c 1 -S 2 -E 3 -g 4 /path/to/my.bed | teststrandbias.R | var2vcf\_valid.pl -N sample\_name -E -f 0.01

The same command in VarDictJava is as follows:

<distribution package path>/bin/VarDict -G /path/to/hg19.fa -f 0.01 -N sample\_name -b /path/to/my.bam -z -c 1 -S 2 -E 3 -g 4 /path/to/my.bed | VarDict/teststrandbias.R | VarDict/var2vcf\_valid.pl -N sample\_name -E -f 0.01

The program uses the same parameters as the Perl version. The Java version contains an additional parameter: -th (from ‘threads’). This parameter manages the number of threads that do the work. If this parameter is missing, then the mode is one-thread. If you add the -th parameter, the number of threads equals to the number of processor cores. The parameter -th threads sets the number of threads explicitly.