iWGS – version 0.4

in silico Whole Genome Sequencer

Xiaofan Zhou

2014/11/10

Contents

Introduction

To be completed

Pre-requisites/Supported tools

This section contains information on all tools supported by iWGS, including the latest supported version and the homepage.

Reads simulator

ART (v03.11.14) [1]: http://www.niehs.nih.gov/research/resources/software/biostatistics/art/

pIRS (v2.0.0) [2]: https://code.google.com/p/pirs/

PBSIM (v1.0.3) [3]: https://code.google.com/p/pbsim/

* The PBSIM package distributed with iWGS has been modified (based on version 1.0.3) to model the distribution of per read quality scores using the Weibull distribution instead of the default normal distribution. It also carries a new quality model profile learned from more recent datasets generated by RSII P5/C3 sequencing chemistry (https://github.com/PacificBiosciences/DevNet/wiki/Arabidopsis-lyrata).

Table 1: iWGS will choose the simulator automatically depending on the target read type and read length.

	SE	PE	MP	PacBio CLR
ART		> 100bp	> 100bp	
pIRS		≤ 100bp	≤ 100bp	
PBSIM				

Color key:

	Supported
	Unsupported

Assembler

ABYSS (v1.5.2) [4]: http://www.bcgsc.ca/platform/bioinfo/software/abyss/

ALLPATHS-LG (r51298) [5, 6]: http://www.broadinstitute.org/software/allpaths-lg/blog/

Celera Assembler (v8.2beta) [7, 8]: http://sourceforge.net/projects/wgs-assembler/

DISCOVAR *de novo* (r51321) [9]:
http://www.broadinstitute.org/software/discovar/blog/

SOAPdenovo2 (v2.0.4) [10]:
http://sourceforge.net/projects/soapdenovo2/files/SOAPdenovo2/

(dip)SPAdes (v3.1.1) [11]: http://bioinf.spbau.ru/spades/

Velvet (v1.2.10) [12]: http://www.ebi.ac.uk/~zerbino/velvet/

Table 2: The compatibility between assemblers and read types.

	SE	PE	MP[a]	PacBio CLR
ABYSS				
ALLPATHS-LG		overlapping		
Celera Assembler[b]				
DISCOVAR *de novo*		(near) overlapping		
SOAPdenovo2				
(dip)SPAdes				
Velvet				

Color key:

	Require all of these
	Require at least one of these
	Supported
	Unsupported

[a]Although ABYSS, SOAPdenovo2, (dip)SPAdes, and Velvet support *de novo* assembly using only MP libraries, it is highly recommended that only high-quality MP data should be used in such cases;
[b]It is recommend to have at least 50x coverage CLR reads for a PacBio only assembly, or at least 20x coverage CLR reads for a hybrid assembly.

Evaluation

QUAST (v2.3): http://bioinf.spbau.ru/quast/

Additional tools

AMOS (v3.1.0) [13]: http://sourceforge.net/projects/amos/

* AMOS is required for Celera Assembler to perform hybrid assembly using both high-quality short reads (e.g. Illumina) and PacBio CLR reads.

BLASR [14]: https://github.com/PacificBiosciences/blasr/

PBDAGCON : https://github.com/PacificBiosciences/pbdagcon/

* BLASR and PBDAGCON are both required for Celera Assembler to run with the slower but more sensitive consensus module "pbdagcon". If either one is missing, the faster module "falcon_sense" will be used instead.

FastqToSam.jar (v1.92) : http://broadinstitute.github.io/picard/

* FastqToSam.jar is part of the Picard package; it is required by DISCOVAR *de novo* to convert FastQ files to BAM format.

KmerGenie (v1.6741) [15]: http://kmergenie.bx.psu.edu/

* KmerGenie can be invoked to estimate the "best" k-mer length for assembly protocols that involve one of these assemblers: ABYSS, SOAPdenovo2, (dip)SPAdes, and Velvet.

Installation

This section contains information on how to install the iWGS pipeline and its pre-requisites.

iWGS

The main iWGS pipeline does not require any installation, and can be run on Linux or Mac OS. To obtain and run iWGS, type:

1) wget [github link missing currently]
2) tar xvf iwgs-[version].tar.gz
3) cd iwgs-[version]
4) ./iWGS

Pre-compiled tools

Pre-compiled binaries for Linux and/or Mac OS are available for some of the supported tools. We thus include these pre-compile tools with the iWGS package so there is no need to install these tools separately:

Reads simulator:

ART: Linux, Mac OS
pIRS: Linux
* The pre-complied binary has a bug, which hopefully will get fixed in future. For now, you will have to compile from source code.
PBSIM: Linux
* The pre-compiled binary is only available for the original version. The source code of the modified version is provided separately.

Assembler:

Celera Assembler: Linux, Mac OS
SOAPdenovo2: Linux, Mac OS
(dip)SPAdes: Linux, Mac OS

Evaluation:

QUAST: Linux, Mac OS

Additional tools:

FastqToSam.jar Linux, Mac OS

Installation reminders/tips for other tools

For other tools, please follow their respective instructions for proper installation. Some reminders/tips are provided here to facilitate a smooth installation (and execution).

Potential dependencies:

Boost: http://www.boost.org/

gtest: https://code.google.com/p/googletest/

HDF5: http://www.hdfgroup.org/HDF5/

jemalloc: http://www.canonware.com/jemalloc/

log4cpp: http://log4cpp.sourceforge.net/

sparsehash: https://code.google.com/p/sparsehash/

zlib: http://www.zlib.net/

Reads simulator:

pIRS:

1) modify the <pirs-dir>/src/pirs/Makfile:
    a. on the first line, replace "/usr/local/share/pirs" with <pirs-dir>/src/Profiles;
    b. requires zlib; if you do not have zlib already, install zlib first, and add the following to the sixth line ("CFLAGS += -static"):
        -L<zlib-lib-dir> –I<zlib-include-dir>.
2) modify the <pirs-dir>/src/stator/gcContCvgBias/Makefile:
    a. requires zlib; if you do not have zlib already, install zlib first, and add the following to the fourth line ("CFLAGS += -static"):
        -L<zlib-lib-dir> –I<zlib-include-dir>.

Assembler:

ABYSS:

1) requires Boost: if you do not have Boost already, install Boost, and add the "--with-boost=<boost-include-dir>" option while running the "configure" script;
2) requires sparsehash: install sparsehash, and add the "CPPFLAGS=-I<sparsehash-include-dir>" option while running the "configure" script;
3) to enable the use of multi-threads, add the "--with-mpi=<openmpi-dir>" option while running the "configure" script; the exact <openmpi-dir> may vary depending on your environment: e.g.

/usr/local/openmpi/latest/x86_64/gcc46 (ACCRE); /usr/lib/openmpi (my PC);

4) by default, ABYSS accepts k-mer sizes up to 64; to allow larger k-mer sizes, add the "--enable-maxk=<value>" (<value> should be a multiple of 32) option while running the "configure" script.

ALLPATHS-LG:

1) requires GCC version ≥ 4.7.0 to compile and **run** (make sure you have the right GCC version when ALLPATHS-LG is executed!);
2) if you do not have access to GCC version ≥ 4.7.0, please use the ALLPATHS-LG version 44837 or earlier;
3) keep in mind that the memory consumption of ALLPATHS-LG will be extremely high when using PacBio data.

DISCOVAR *de novo*:

1) requires GCC version ≥ 4.7.0 to compile and **run** (make sure you have the right GCC version when DISCOVAR *de novo* is executed!);
2) unfortunately, there is no version of DISCOVAR *de novo* that is compatible with GCC version < 4.7.0;
3) requires jemalloc; if you do not have jemalloc already, install jemalloc (version ≥ 3.6.0), and add the "--with-jemalloc=<jemalloc-lib-dir>" option while running the "configure" script.

Velvet:

1) by default, Velvet accepts up to two datasets ("categories"); to allow more than two datasets, add the "CATEGORIES=<value>" option while running the "make" command;
2) by default, Velvet accepts k-mer sizes up to 31; to allow larger k-mer sizes, add the "MAXKMERLENGTH=<value>" option while running the "make" command.
3) to allow the assembly of datasets with more than 2.2 billion reads, add the option "BIGASSEMBLY=1" option while running the "make" command;
4) to enable the use of multi-threads, add the "OPENMP=1" option while running the "make" command.

Evaluation tools:

QUAST:

1) Run "quast.py -test" before the first use.

Additional tools:

AMOS:

1) an error encountered when trying to compile with GCC (version ≥ 4.9.0); earlier GCC versions (e.g. 4.6) seem work fine.

BLASR:

1) requires HDF5:
    a. it is recommended to have a fresh installation of HDF5 (version 1.8.10 or above tested, add the "--enable-cxx" option while running the "configure" script);
    b. modify the $path_to_blasr/common.mk file to change "HDF5INCLUDEDIR" (fifth line) and "HDF5LIBDIR" (sixth line) accordingly;
    c. if errors related to hdf5lib occur, add $path_to_hdf5/lib to the $LD_LIBRARY_PATH environmental variable.

PBDAGCON:

1) use the same GCC version for all the following dependencies (version 4.6.1 tested);
2) requires Boost (version 1.47 tested; version 1.46 is also compatible according to the manual);
3) requires log4cpp (version 1.0 or 1.1);
4) requires gtest (version 1.3.0 or above);
5) do the following:
    a. unzip pbdagcon-master.zip
    b. cd pbdagcon-master
    c. git clone https://github.com/pbjd/alignment.git
    d. git clone https://github.com/pbjd/pbdata.git
    e. modify "src/cpp/pbi.mk" to change "BOOST", "LOG4CPP", "GTEST", "INCDIRS", and "LIBDIRS" accordingly
    f. make
    g. the binary file of pbdagcon will be generated in src/cpp

KmerGenie:

1) by default, KmerGenie evaluates k-mer sizes up to 96; to allow the evaluation of larger k-mer sizes, do the following:
    a. make clean
    b. make k=<value>

Running iWGS

iWGS uses both command line options and a control file to collect all information it needs to carry out an analysis.

Command line options

Basic options:

-s | --settings <string>	the control file that contains all information for a iWGS run (see the "control file" section for detailed explanation)
-g | --genome <string>	the reference genome sequence in FASTA format
-t | --threads <int>	the number of threads to use (default: 1)
-m | --memory <int>	the number of GBs of memory to use (default: 8)
-o | --out_dir <string>	the directory to store all outputs, will be created if not already exists (default: ./)
-c | --clean_up <int>	whether to remove simulated reads and files generated by assemblers after the iWGS run (default: 0 [no])
-r | --overwrite <int>	whether to redo the simulation if some libraries already exists (default: 0 [no])
-v | --verify <int>	verify the settings and quit
-p | --price <string>	provide pricing information to calculate the cost of each library and assembly protocol

Advanced options:

--Mode	iWGS can run in three modes: (default: 1) 1: get parameters from command line / control file, and finish all steps (reads simulation, assembly, and evaluation) in a single run 2: get parameters from command line / control file, and generate three configuration files for library, assembly protocol, and miscellaneous options, respectively, which would allow each library / assembly protocol to have different parameters (see the "advanced configuration files" section for detailed explanation) 3: restart from the configuration files generated in Mode 2
--Conf	the configuration files for library, assembly protocol, and miscellaneous options, separated by comma; to be used in Mode 3
--Real	run iWGS with real datasets

Control file

The control file consists of the following sections (see Appendix for the complete control file):

General options:

These general options in the control file are the same as corresponding command line options, and will override the command line options if both are specified.

genome =
out_dir =
threads =
memory =

Library options:

Each library option line starts with the key word "library" and contains the information for a library to be simulated, in the format of four to six parameters separated by comma. The number and meaning of the parameters needed for a library vary with different read types:

	1st parameter	2nd parameter	3rd parameter	4th parameter	5th parameter	6th parameter
SE	Library name	Read type	Coverage	Read length		
PE					Average insert size	SD of insert size
MP						
CLR				Average accuracy	SD of accurary	

Note: there is no need to specify the reads simulator; iWGS will make the choice automatically based on the read type and other specifics of the library.

Example 1: a 100bp SE library "L001" with 50x coverage.
library = L001,SE,50,100

Example 2: a 100bp PE library "L002" with 50x coverage, average insert size of 180bp, and SD of insert size of 9bp.
library = L002,PE,50,100,180,9

Example 3: a 100bp MP library "L003" with 50x coverage, average insert size of 5000bp, and SD of insert size of 250bp.
library = L003,MP,50,100,5000,250

Example 4: a PacBio CLR library "L004" with 50x coverage, average read accuracy of 0.85, and SD of read accuracy of 0.02.
library = L004,CLR,50,0.85,0.02

Simulator options:

Each simulator option line starts with a keyword in the format of: "[name_of_simulator].[option_of_simulator]". These options are used to set parameters for read simulators. Leave the options blank to use the default values.

Note: these parameters are applied universally to all libraries simulated by the corresponding simulator.

Example 1: set the substitution error rate for pIRS
pIRS.error_rate =

Example 2: set the amount to shift every first-read quality score for ART
ART.qual_shift1 =

Example 3: set the maximum read accuracy for PBSIM
PBSIM.accuracy_max =

Assembly protocol options:

Each assembly protocol option line starts with the key word "protocol" and contains the information for an assembly protocol, in the format of three or more parameters separated by comma. The number and meaning of the parameters needed for an assembly protocol are shown below:

1st parameter	2nd parameter	3rd .. nth parameters
Protocol name	Assembler name	Names of libraries to be included

Example 1: an assembly protocol "P001" using the assembler SOAPdenovo2, and libraries L001 (SE) and L002 (PE).
protocol = P001,SOAPdenovo2,L001,L002

Example 2: an assembly protocol "P002" using the assembler ALLPAHTS-LG, and libraries L002 (overlapping PE), L003 (MP), and L004 (PacBio CLR).
Protocol = P002,ALLPATHS,L002,L003,L004

Assembler options:

Each assembler option line starts with a keyword in the format of: "[name_of_assembler].[option_of_assembler]". These options are used to set parameters for *de novo* genome assemblers. Leave the options blank to use the default values.

Note: these parameters are applied universally to all assembly protocols using the genome assembler.

Example 1: set the k-mer size to be used for SOAPdenovo2; the default value ("0") indicates that the k-mer size is to be estimated.
SOAPdenovo2.kmer = 0

Example 2: set additional options for SPAdes; the text between quotation marks will be passed to the assembler.
SPAdes.option = "--only-assembler"

Evaluation options:

Each evalution option line starts with a keyword in the format of: "[name_of_evaluator].[option_of_evaluator]". These options are used to set parameters for genome assembly assessment tools. Leave the options blank to use the default values.

Example 1: whether the genome to evaluate is eukaryotic
QUAST.eukaryote = 1

Example 2: whether to run QUAST in "GAGE" mode
QUAST.gage = 1

Example 3: to provide a gene annotation file for QUAST (only effective in full evaluation mode)
QUAST.gene =

Executable options:

Each executable option line starts with a keyword in the format of: "bin.[name_of_executable]". These options are used to set path information for needed executables. Leave the options blank if the executables are available in PATH environmental variable.

Example 1: set the path to the pIRS executable, "pirs"
bin.pIRS =

Example 2: set the path to SPAdes executable, "spades.py"
bin.SPAdes =

Example 3: set the path to QUAST executable, "quast.py"
bin.QUAST =

Advanced configuration files

One limitation of the control file is that the simulator/assembler related parameters will be applied universally to all relevant libraries / assembly protocols. iWGS allows advanced users to set parameters specific to each library / assembly protocol in the following way:

1) run iWGS in Mode 2 (advanced option); iWGS will collect information from command line options and/or the control file, and generate three configuration files for library, assembly protocol, and miscellaneous options, respectively;
2) modify the configuration files to set library /assembly protocol specific parameters;
3) run iWGS in Mode 3 to resume the analysis from the modified configuration files.

The three configuration files are explained as below (see Appendix for exemplar configuration files):

Library configuration file: (default name: libraries.conf)

This file contains one section for each library, and all keywords are in the format of: "[library_name].[option_name]".

All libraries have three common options: (using L001 as example)

L001.read_type = SE
L001.simulator = ART
L001.depth = 50

Note: the "read_type" and "simulator" options are for users' reference only, please do not modify.

Other available options vary depend on the read type and simulator; for example:

L001.qual_shift1 =
L001.qual_profile1 =
L001.ins_rate1 =
L001.del_rate1=

Since L001 is a SE library, only the ART parameters for the first-read are available.

Assembly protocol configuration file: (default name: protocols.conf)

This file contains one section for each assembly protocol, and all keywords are in the format of: "[protocol_name].[option_name]".

All protocols have two common options: (using P001 as example)

P001.library = L001,L002
P001.assembler = SOAPdenovo2

Note: the "assembler" options are for users' reference only, please do not modify.

Other available options vary depend on the assembler; for example:

P001.kmer = 0
P001.option = "-F -R -E -w -u"

The default value for "option" (the extra command to pass to the assembler: SOAPdenovo2) is taken from the recipes of GAGE-B project [16].

Miscellaneous option configuration file: (default name: misc.conf)

This file contains all the general options, evaluation options, and executable options, which are in the same format as in the control file.

Output

iWGS will create five directories under the "out_dir" directory:

1) $out_dir/libraries/ - this directory contains all simulated reads; each simulated library will be stored in a separate subdirectory with the same name as the library; this folder will be removed if the "cleanup" command line option is turned on;
2) $out_dir/protocols/ - this directory contains the results of all assembly protocols; files generated by each assembly protocol will be stored in a separate subdirectory with the same name as the protocol; this folder will be removed if the "cleanup" command line option is turned on;
3) $out_dir/logs/ - this directory contains log files for all library simulation, assembly, and evaluation steps;
4) $out_dir/assemblies/ - this directory contains copies of all successfully assembled contig/scaffold files;
5) $out_dir/evaluation/ - this directory contains the results of QUAST evaluations; copies of the main evaluation reports can be found in this directory; two subdirectores, "contigs/" and "scaffolds/", will be created to store the detailed evaluation results on assembled contigs and scaffolds, respectively.

Examples

A quick and simple iWGS analysis:

A quick and simple iWGS analysis can be carried out without using a control file, in which case the only required option is the reference genome ("-g"). iWGS will simulate the following two libraries:

Name	Read type	Simulator	Read length	Coverage	Average insert size	SD of insert size
L001	PE	pIRS	100bp	50x	180bp	9bp
L002	MP	pIRS	100bp	50x	3000bp	150bp

and evaluate the following two assembly protocols:

Name	Assembler	Libraries	Note
P001	SOAPdenovo2	L001, L002	k-mer size to be estimated
P002	ALLPATHS-LG	L001, L002	

iWGS will behave in the same way in cases where a control file is provided but no library and assembly protocol is specified.

Exemplar command:
iWGS -g Kazachestania_africana.fa -t 8 -m 32 -o K_africana

Run with library / assembly protocol specific parameters:

1) collect information from the control file and generate configuration files:
   iWGS -s K_africana.ctl --Mode 2
2) modify the configuration files;
3) resume the analysis:
   iWGS --Mode 3 --Conf libraries.conf,protocols.conf,misc.conf

Using iWGS to analyze real datasets:

iWGS can also be used to analyze real datasets, providing a convenient way for users to try multiple assembler/parameter combinations on their own data and evaluate the resulting assemblies.

1) create the output directory ($out_dir) and the "libraries" subdirectory ($out_dir/libraries);
2) under the "libraries" directory, create a subdirectory for each real dataset to place the corresponding data file(s); the names of the subdirectory the data files should follow the naming convention of iWGS:

Read type	Subdirectory name	Data file name
SE, PacBio CLR	[library_name][a]	[library_name].fq
PE, MP		[library_name]_1.fq, [library_name]_2.fq

[a]the [library_name] can be any legitimate name for the library.

3) in the iWGS control file, create a library option line for each real dataset; real specs (e.g. coverage, read length) of the libraries should be provided;
4) run iWGS with the "Real" option (Note: the "overwrite" and "cleanup" options are automatically turned off with "Real" option; the datasets will be over-written and/or removed if these options are on ; also, always make sure to have backup of the real datasets):
   # Exemplar command:
   iWGS -s real_data.ctl -r 0 -c 0 --Real

Note: similarly, with proper library options in the control file, iWGS can analyze real datasets together with simulated data. The users can use this feature to evaluate the potential improvements additional datasets may have on existing real data.

References

1. Huang, W., et al., *ART: a next-generation sequencing read simulator.* Bioinformatics, 2012. **28**(4): p. 593-4.
2. Hu, X., et al., *pIRS: Profile-based Illumina pair-end reads simulator.* Bioinformatics, 2012. **28**(11): p. 1533-5.
3. Ono, Y., K. Asai, and M. Hamada, *PBSIM: PacBio reads simulator--toward accurate genome assembly.* Bioinformatics, 2013. **29**(1): p. 119-21.
4. Simpson, J.T., et al., *ABySS: a parallel assembler for short read sequence data.* Genome Res, 2009. **19**(6): p. 1117-23.
5. Gnerre, S., et al., *High-quality draft assemblies of mammalian genomes from massively parallel sequence data.* Proc Natl Acad Sci U S A, 2011. **108**(4): p. 1513-8.
6. Ribeiro, F.J., et al., *Finished bacterial genomes from shotgun sequence data.* Genome Res, 2012. **22**(11): p. 2270-7.
7. Myers, E.W., et al., *A whole-genome assembly of Drosophila.* Science, 2000. **287**(5461): p. 2196-204.
8. Koren, S., et al., *Reducing assembly complexity of microbial genomes with single-molecule sequencing.* Genome Biol, 2013. **14**(9): p. R101.
9. Weisenfeld, N.I., et al., *Comprehensive variation discovery in single human genomes.* Nat Genet, 2014.
10. Luo, R., et al., *SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.* Gigascience, 2012. **1**(1): p. 18.
11. Bankevich, A., et al., *SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing.* J Comput Biol, 2012. **19**(5): p. 455-77.
12. Zerbino, D.R. and E. Birney, *Velvet: algorithms for de novo short read assembly using de Bruijn graphs.* Genome Res, 2008. **18**(5): p. 821-9.
13. Schatz, M.C., et al., *Hawkeye and AMOS: visualizing and assessing the quality of genome assemblies.* Brief Bioinform, 2013. **14**(2): p. 213-24.
14. Chaisson, M.J. and G. Tesler, *Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory.* BMC Bioinformatics, 2012. **13**: p. 238.
15. Chikhi, R. and P. Medvedev, *Informed and automated k-mer size selection for genome assembly.* Bioinformatics, 2014. **30**(1): p. 31-7.
16. Magoc, T., et al., *GAGE-B: an evaluation of genome assemblers for bacterial organisms.* Bioinformatics, 2013. **29**(14): p. 1718-25.

Appendix

A complete control file

```
############################
# General options
############################
genome = example/Kazachstania_africana.fa          # the reference genome sequence in fasta format
out_dir = K_africana                               # all outputs will be written to the folder ./K_african
threads = 4                                        # number of CPUs to use (default: 1)
memory = 32                                        # number of GBs of memory to use (default: 8)

############################
# Library options
############################
library = L001,SE,50,100                           # Illumina 100bp SE library "L001" - coverage: 50x
library = L002,PE,50,100,180,9                     # Illumina 100bp PE library "L002" - coverage: 50x;
average insert size: 180bp; SD of insert size: 9bp
library = L003,MP,50,100,5000,250                  # Illumina 100bp MP library "L003" - coverage: 50x;
average insert size: 5000bp; SD of insert size: 250bp
library = L004,CLR,50,0.85,0.02                    # PacBio CLR library "L003" - coverage: 50x; average
read accuracy: 0.85; SD of read accuracy: 0.02
library = L005,PE,50,250,450,23                    # Illumina 250bp PE library "L005" - coverage: 50x;
average insert size: 450bp; SD of insert size: 23bp

############################
# Simulator options
############################
pIRS.error_rate =                                  # substitution error rate: 0, 1, or 0.0001-0.63
(default: 1 - indicate that the default setting of pIRS should be used)
pIRS.error_profile =                               # the base-calling profile for simulating
substitution-error and quality score (default: the default profile of pIRS)
pIRS.gc =                                          # whether to simulate GC bias: 1 - yes; 0 - no
(default: 1)
pIRS.gc_profile =                                  # the GC content-coverage file for simulating GC
bias (default: the default profile of pIRS)
pIRS.indel =                                       # whether to simulate indel errors: 1 - yes; 0 - no
(default: 1)
pIRS.indel_profile =                               # the InDel-error profile for simulating InDel-error
(default: the default profile of pIRS)

ART.qual_shift1 =                                  # the amount to shift the quality score of all first-
reads (default: 0)
ART.qual_shift2 =                                  # the amount to shift the quality score of all second-
reads (default: 0)
ART.qual_profile1 =                                # the quality profile of first-reads (default: the
default profile of ART)
ART.qual_profile2 =                                # the quality profile of second-reads (default: the
default profile of ART)
```

```
ART.ins_rate1 =                          # the insertion rate of first-reads (default: 0.00009)
ART.ins_rate2 =                          # the insertion rate of second-reads (default:
0.00015)
ART.del_rate1 =                          # the deletion rate of first-reads (default: 0.00011)
ART.del_rate2 =                          # the deletion rate of second-reads (default:
0.00023)

PBSIM.model_qc =                         # the model of quality code for simulating read
accuracy (default: the default profile of PBSIM)
PBSIM.ratio =                            # the ratio of substitution:insertion:deletion errors
(default: 10:60:30)
PBSIM.accuracy_max =                     # the maximum read accuracy (default: 0.90)
PBSIM.accuracy_min =                     # the minimum read accuracy (default: 0.75)
PBSIM.length_mean =                      # the mean of read length (default: 3000)
PBSIM.length_sd =                        # the standard deviation of read length (default:
2300)
PBSIM.length_max =                       # the maximum read length (default: 25000)
PBSIM.length_min =                       # the minimum read length (default: 100)

############################
# Assembly protocol options
############################
protocol = P001,ABYSS,L001,L002,L003     # assembly protocol "P001" that uses the ABYSS
assembler and libraries "L001", “L002”, and “L003”
protocol = P002,ALLPATHS,L002,L003       # assembly protocol "P002" that uses the
ALLPATHS-LG assembler and libraries “L002” and “L003”
protocol = P003,CA,L002,L004             # assembly protocol "P003" that uses the Celera
Assembler and libraries “L002” and “L004”
protocol = P004,CA,L004                  # assembly protocol "P004" that uses the Celera
Assembler and the library “L004”
protocol = P005,DISCOVAR,L005            # assembly protocol "P005" that uses the DISCOVAR
de novo assembler and the library “L005”
protocol = P005,SOAPdenovo2,L001,L002,L003  # assembly protocol "P006" that uses the
SOAPdenovo2 assembler and libraries "L001", “L002”, and “L003”
protocol = P005,SPAdes,L001,L002,L003    # assembly protocol "P007" that uses the SPAdes
assembler and libraries "L001", “L002”, and “L003”
protocol = P006,Velvet,L001,L002,L003    # assembly protocol "P008" that uses the Velvet
assembler and libraries "L001", “L002”, and “L003”

############################
# Assembler options
############################
ABYSS.kmer =                             # default: 0 - to be estimated using KmerGenie
ABYSS.option = “l=1 n=5 s=100”           # GAGE-B recipe

ALLPATHS.ploidy =                        # default: 1; set to 2 for heterozygous assembly

CA.sensitive =                           # default: 0; set to 1 for lower-quality PacBio data

SPAdes.kmer =                            # default: 0 - to be estimated using KmerGenie
```

```
SPAdes.multi-kmer =                          # default: 1 - to use multiple k-mer sizes
SPAdes.option = "--only-assembler"           # set it empty (i.e. " ") to enable the error-correction
module

SOAPdenovo2.kmer =                           # default: 0 - to be estimated using KmerGenie
SOAPdenovo2.option = "-F -R -E -w -u"        # GAGE-B recipe

Velvet.kmer =                                # default: 0 - to be estimated using KmerGenie
Velvet.option = "-exp_cov auto -scaffolding yes"   # GAGE-B recipe

############################
# Evaluation options
############################
QUAST.eukaryote =                            # whether the reference genome is eukaryotic: 1 -
yes; 0 - no (default: 1)
QUAST.gage =                                 # whether to generate GAGE report: 1 - yes; 0 - no
(default: 1)
QUAST.gene = example/Kazachstania_africana.genes          # gene annotations to be used for
evaluation (default: NA)

############################
# Executable options
############################
bin.pIRS = /home/xiaofan/iWGS/tools/pIRS/pirs
bin.ART =/home/xiaofan/iWGS/tools/ART/art_illumina
bin.PBSIM = /home/xiaofan/iWGS/tools/PBSIM/bin/pbsim
bin.KmerGenie = /home/xiaofan/iWGS/tools/kmergenie/kmergenie
bin.ABYSS = /home/xiaofan/iWGS/tools/ABYSS/bin/abyss-pe
bin.ALLPATHS = /home/xiaofan/iWGS/tools/ALLPATHS-LG/bin/RunAllPathsLG
bin.CA = /home/xiaofan/iWGS/tools/Linux-amd64/bin/PBcR
bin.DISCOVAR = /home/xiaofan/iWGS/tools/DISCOVAR/bin/DiscovarExp
bin.SOAPdenovo2 = /home/xiaofan/iWGS/tools/SOAPdenovo2
bin.SPAdes = /home/xiaofan/iWGS/tools/SPAdes/bin/spades.py
bin.dipSPAdes = /home/xiaofan/iWGS/tools/SPAdes/bin/dipspades.py
bin.velvetg = /home/xiaofan/iWGS/tools/Velvet/velvetg
bin.velveth = /home/xiaofan/iWGS/tools/Velvet/velveth
bin.FastqToSam = /home/xiaofan/iWGS/tools/FastqToSam.jar
bin.QUAST = /home/xiaofan/iWGS/tools/QUAST/quast.py
```

Exemplar configuration files

Library configuration file: (one example for each read type)

```
############################
# Library options
############################

# parameters for the library L001
L001.read_type = SE
L001.depth = 50
```

```
L001.simulator = ART
L001.read_length = 100
L001.qual_shift1 = 0
L001.qual_profile1 =
L001.ins_rate1 = 0.00009
L001.del_rate1 = 0.00011

# parameters for the library L002
L002.read_type = PE
L002.depth = 50
L002.simulator = pIRS
L002.read_length = 100
L002.frag_mean = 180
L002.frag_sd = 9
L002.error_rate = 1
L002.error_profile =
L002.gc = 1
L002.gc_profile =
L002.indel = 1
L002.indel_profile =

# parameters for the library L003
L003.read_type = MP
L003.depth = 50
L003.simulator = pIRS
L003.read_length = 100
L003.frag_mean = 5000
L003.frag_sd = 250
L003.error_rate = 1
L003.error_profile =
L003.gc = 1
L003.gc_profile =
L003.indel = 1
L003.indel_profile =

# parameters for the library L004
L004.read_type = CLR
L004.depth = 50
L004.simulator = PBSIM
L004.accuracy_mean = 0.85
L004.accuracy_sd = 0.02
L004.accuracy_max = 0.90
L004.accuracy_min = 0.75
L004.model_qc = /home/xiaofan/iWGS/tools/PBSIM/data/model_qc_clr
L004.ratio = 10:60:30
L004.length_mean = 3000
L004.length_sd = 2300
L004.length_max = 25000
L004.length_min = 100
```

```
# parameters for the library L005
L005.read_type = PE
L005.depth = 50
L005.simulator = ART
L005.read_length = 250
L005.frag_mean = 450
L005.frag_sd = 23
L005.qual_shift1 = 0
L005.qual_shift2 = 0
L005.qual_profile1 =
L005.qual_profile2 =
L005.ins_rate1 = 0.00009
L005.ins_rate2 = 0.00015
L005.del_rate1 = 0.00011
L005.del_rate2 = 0.00023
```

Protocol configuration file: (one example for each assembler)

```
##############################
# Assembly protocol options
##############################

# parameters for the assembly protocol P001
P001.library = L001,L002,L003
P001.assembler = ABYSS
P001.kmer = 0
P001.option = "l=1 n=5 s=100"

# parameters for the assembly protocol P002
P002.library = L002,L003
P002.assembler = ALLPATHS
P002.ploidy = 1

# parameters for the assembly protocol P003
P003.library = L002,L004
P003.assembler = CA
P003.sensitive = 0

# parameters for the assembly protocol P004
P004.library = L004
P004.assembler = CA
P004.sensitive = 0

# parameters for the assembly protocol P005
P005.library = L005
P005.assembler = DISCOVAR

# parameters for the assembly protocol P006
P006.library = L001,L002,L003
P006.assembler = SOAPdenovo2
```

P006.kmer = 0
P006.option = "–F –R –E –w –u"

# parameters for the assembly protocol P007
P007.library = L001,L002,L003,L004
P007.assembler = SPAdes
P007.kmer = 0
P007.multi-kmer = 1
P007.option = "--only-assembler"

# parameters for the assembly protocol P008
P008.library = L001,L002,L003
P008.assembler = Velvet
P008.kmer = 0
P008.option = "-exp_cov auto -scaffolding yes"

Miscellaneous options configuration file:

```
##############################
# General options
##############################
genome = /home/xiaofan/iWGS/example/Kazachstania_africana.fa
threads = 4
memory = 32
out_dir = /home/xiaofan/iWGS/example/K_africana

##############################
# Evaluation options
##############################
QUAST.eukaryote = 1
QUAST.gage = 1
QUAST.gene = /home/xiaofan/iWGS /example/Kazachstania_africana.genes

##############################
# Executable options
##############################
bin.pIRS = /home/xiaofan/iWGS/tools/pIRS/pirs
bin.ART =/home/xiaofan/iWGS/tools/ART/art_illumina
bin.PBSIM = /home/xiaofan/iWGS/tools/PBSIM/bin/pbsim
bin.KmerGenie = /home/xiaofan/iWGS/tools/kmergenie/kmergenie
bin.ABYSS = /home/xiaofan/iWGS/tools/ABYSS/bin/abyss-pe
bin.ALLPATHS = /home/zhoux8/usr/bin/allpathslg/bin/RunAllPathsLG
bin.CA = /home/xiaofan/iWGS/tools/Linux-amd64/bin/PBcR
bin.DISCOVAR = /home/xiaofan/iWGS/tools/Discovar/bin/DiscovarExp
bin.SOAPdenovo2 = /home/xiaofan/iWGS/tools/SOAPdenovo2
bin.SPAdes = /home/xiaofan/iWGS/tools/SPAdes/bin/spades.py
bin.dipSPAdes = /home/xiaofan/iWGS/tools/SPAdes/bin/dipspades.py
bin.velvetg = /home/xiaofan/iWGS/tools/Velvet/velvetg
bin.velveth = /home/xiaofan/iWGS/tools/Velvet/velveth
bin.FastqToSam = /home/xiaofan/iWGS/tools/FastqToSam.jar
```

bin.QUAST = /home/xiaofan/iWGS/tools/QUAST/quast.py