Instruction Manual for the Ray *de novo* genome assembler software

Sébastien Boisvert

May 27, 2011

Ray version 1.4.0

http://denovoassembler.sf.net

Reference to cite:

Sébastien Boisvert, François Laviolette & Jacques Corbeil.
Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies.
Journal of Computational Biology (Mary Ann Liebert, Inc. publishers, New York, U.S.A.).
November 2010, Volume 17, Issue 11, Pages 1519-1533.
doi:10.1089/cmb.2009.0238
http://dx.doi.org/doi:10.1089/cmb.2009.0238

Contents

1 Installation

To install Ray, you need a C++ compiler, make and an MPI implementation compliant with MPI standard 2.2.

Ray runs on any POSIX-compliant system. This includes GNU/Linux. Windows users need to install Cygwin although Ray is not tested on Cygwin.

The software below are readily available in most GNU/Linux distributions (Table 1).

Table 1: Suggested software

Software	Name
C++ compiler	GNU g++
MPI implementation	Open-MPI

There are some compilation flags that can be changed. These are listed below in Table 2.

Table 2: Compilation flags

Flag	Effect
HAVE_ZLIB	enable support for .gz files
HAVE_LIBBZ2	enable support for .bz2 files
FORCE_PACKING	enable packing of structures and classes
ASSERT	enable assertions in the code
HAVE_CLOCK_GETTIME	get time at a nanosecond precision – real-time

The source code of Ray can be compiled with a Makefile. You need to have mpic++ in your path, otherwise edit the Makefile to change CC.

```
make
ls -l code/Ray
```

2 Inputs

The input files for Ray contain sequences. The files must be formatted in one of the supported formats. These formats are listed in Table 4. Note that the file extension is obligatory and Ray uses it to select the file format.

Table 3: File formats compatible with the Ray *de novo* genome assembler software

Format	Obligatory extension
Fasta format	.fasta
Fasta format, compressed with GNU zip (gzip)	.fasta.gz
Fasta format, compressed with bzip2	.fasta.bz2
Fastq format	.fastq
Fastq format, compressed with GNU zip (gzip)	.fastq.gz
Fastq format, compressed with bzip2	.fastq.bz2
Standard flowgram format	.sff

3 Parameters

Ray assembles reads (paired or not) to produce an assembly. Paired reads must be on opposite strands (forward & reverse or reverse & forward).

For a paired library, paired reads can be provided as two files (with -p) or as one file containing interleaved sequences (with -i). With both, the average outer distance and the standard deviation can be provided by the user. Otherwise, these values are computed by Ray. The maximum number of paired libraries allowed by Ray is 499.

3.1 k-mer length with -k

Ray builds a distributed catalog of all occuring k-mers in the reads and their reverse-complement. k must be greater or equal to 15 and lower or equal to 31. The k-mer length must be an odd number.

3.2 Output prefix with -o

Output files are named according to the prefix provided by the option -o.

3.3 Single-end reads with -s

```
-s <sequencesFile>
```

3.4 Paired-end reads and mate-pair reads with -p

Average outer distance and standard deviation are computed by Ray if omitted.

```
-p <leftSequencesFile> <rightSequencesFile>
```

```
OR
```

```
-p <leftSequencesFile> <rightSequencesFile>  <averageFragmentLength> <standardDeviation>
```

Example for paired-end reads (the ends of DNA fragments):

```
-p s_200_1.fastq s_200_2.fastq
```

Example for mate-pair reads:

```
-p s_20000_1.fastq s_20000_2.fastq
```

Example with metagenomic data and user-provided average outer distance and standard deviation

```
-p s_1.fastq s_2.fastq 300 30
```

3.5 Paired-end reads and mate-pair reads with -i (interleaved sequences)

Average outer distance and standard deviation are computed by Ray if omitted.

```
-i <sequencesFile>
```

```
OR
```

```
-i <sequencesFile>  <averageFragmentLength> <standardDeviation>
```

In the interleaved file (example is for a fasta file):

```
>200_1_1234/1
ATCGATCGATCGACTCAGACACGTACG
>200_1_1234/2
ACTGACGACGTACGACGTCATGCAACT
...
```

4 Outputs

Table 4: File formats compatible with the Ray *de novo* genome assembler software

File	Description
PREFIX.RayCommand.txt	The exact same command provided
PREFIX.RayVersion.txt	The version of Ray
PREFIX.Contigs.fasta	Contiguous sequences in FASTA format
PREFIX.ContigLengths.txt	The lengths of contiguous sequences
PREFIX.Scaffolds.fasta	The scaffold sequences in FASTA format
PREFIX.ScaffoldComponents.txt	The components of each scaffold
PREFIX.ScaffoldLengths.txt	The length of each scaffold
PREFIX.ScaffoldLinks.txt	Scaffold links
PREFIX.CoverageDistribution.txt	The distribution of coverage values
PREFIX.CoverageDistributionAnalysis.txt	Analysis of the coverage distribution
PREFIX.LibraryStatistics.txt	Number of reads in each file and estimation of outer distances for paired reac
PREFIX.SeedLengthDistribution.txt	The distribution of seed lengths
PREFIX.OutputNumbers.txt	Overall numbers for the assembly
PREFIX.AMOS.afg	Assembly representation in AMOS format

5 Validation of assemblies

In the scripts/ directory, the script ValidateGenomeAssembly.sh can validate an assembly against the corresponding reference.

6 Example

6.1 Bacterial genome with paired-end and mate-pair short reads

The command:

```
mpirun -np 32 ~/Ray/trunk/code/Ray \
-p /home/boiseb01/nuccore/Large-Ecoli/200_1.fastq \
   /home/boiseb01/nuccore/Large-Ecoli/200_2.fastq \
-p /home/boiseb01/nuccore/Large-Ecoli/1000_1.fastq \
   /home/boiseb01/nuccore/Large-Ecoli/1000_2.fastq \
-p /home/boiseb01/nuccore/Large-Ecoli/4000_1.fastq \
   /home/boiseb01/nuccore/Large-Ecoli/4000_2.fastq \
-p /home/boiseb01/nuccore/Large-Ecoli/10000_1.fastq \
   /home/boiseb01/nuccore/Large-Ecoli/10000_2.fastq \
-o BacterialGenome | tee RayLog
```

7 Virtual sequencer & simulations

The Ray package includes a simulator for paired reads.

```
N=6000000
readLength=50
errorRate=0.005
```

```
ref=~/nuccore/Ecoli-k12-mg1655.fasta

g++ code/simulatePairedReads.cpp -O3 -Wall -o Simulator
./Simulator $ref $errorRate 200 20 $N $readLength L1_1.fasta L1_2.fasta
```

8 Exploring the source code

Ray source code is documented with Doxygen standards.

The command below generates code documentation.

```
doxygen DoxygenConfigurationFile
```

To browse it, use this command:

```
firefox DoxygenDocumentation/html/index.html
```

9 Reporting bugs & crashes

```
On the mailing list:
denovoassembler-users # lists DOT sourceforge.net

or personal email:
sebastien.boisvert.3 # ulaval DOT ca  or seb # boisvert DOT info
```

Elements to send:

1. Ray command including mpirun
2. Ray standard output: mpirun -np 32 /path/to/Ray -p p_1.fastq p_2.fastq -o Out |& tee Log; gzip Log; ls -lh Log.gz
3. C++ compiler name (examples: GNU g++, Intel ICC, or other) and version (usually −−version)
4. MPI library name (examples: Open-MPI, MPICH2, or other) and version (usually −−version)
5. Sequencer vendor and model that generated reads
6. Total physical memory of the system and physical memory available for each MPI ranks
7. Job scheduler (examples: Grid Engine, PBS, none or other)

Also, if you have access to compute nodes:

8. Memory page size: getconf PAGESIZE > pagesize
9. Central processing unit: cat /proc/cpuinfo—gzip> cpuinfo.gz
10. Memory information: cat /proc/meminfo—gzip> meminfo.gz
11. Kernel: uname -a> uname-a