Vicuna User Manual

Xiao Yang, Patrick Charlebois, Michael C. Zody and Matthew Henn

Genome Sequencing and Analysis Program
The Broad Institute of MIT and Harvard

May, 2012

Contents

1 General Description

Vicuna is a program for *de novo* consensus assembly of viral population. It leverages efficient clustering, partitioning and alignment algorithms to make overlap-layout-consensus assembly strategy applicable to next-gen datasets. Vicuna has been used to assemble clinical HIV, RSV, West Nile, and Dengue population, and should be applicable to any other type of retrovirus sample. Vicuna has been applied to both Illumina paired reads and 454 reads. It should be directly applicable for Ion Torrent reads as well.

2 Quick Start – for Broad Institute Users

1. Use bash environment.

   ```
   $ bash
   ```

2. Export NCIB toolkit library path.

   ```
   $ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/seq/viral/analysis/xyang/programs/
   Library/ncbi_cxx--7_0_0/lib
   ```

3. Copy Vicuna template config file to your local directory, where user should specify the path parameter [vicuna_config]

   ```
   $ cp /seq/viral/analysis/xyang/programs/Vicuna/config-miseq-general.txt [vicuna_config]
   ```

4. Modify the parameters in file [vicuna_config]. Note, please uncomment the parameter names, and there should be no space between the beginning of the line and the parameter name.

   - pFqDir – the path of the input folder. This folder should contain a multiple of two paired read fastq file, ending with ".fq" or ".fastq". The user has to make sure that if there are more than 2 paired files, when loaded into the memory, two files that form pairs have to be loaded consecutively. This can be achieved by naming a pair of read files to be i_1.fq and i_2.fq, for instance, where i can be an integer or a string.
   - outputDIR – the path of the output folder.
   - MSAFileName – the path of the file stores the MSA of target genomes, recommended to be used for samples rich in contamination. Set this parameter to be "/seq/viral/analysis/xyang/programs/Vicuna/db/hiv-1B.algn" for HIV, "/seq/viral/analysis/xyang/programs/Vicuna/db/wnv.align" for West Nile virus, "/seq/viral/analysis/xyang/programs/Vicuna/db/denv.align" for Dengue virus, and "/seq/viral/analysis/xyang/programs/Vicuna/db/lasvL.align" for L element of Las virus, and "/seq/viral/analysis/xyang/programs/Vicuna/db/lasvS.align" for S element of Las virus.

5. Run Vicuna.

```
$ OMP_NUM_THREADS=8 /seq/viral/analysis/xyang/programs/Vicuna/bin/vicuna-omp-v8
[vicuna_config]
```

Note:

(1) you can bsub the above command, for example:

```
$ bsub -P ProjName -o {screen_output.txt} -q hour -W 4:00 -R ”rusage[mem=6]” -n 2,8 -R
’span[hosts=1]’ {vicuna} {vicuna_config}.
```

Please replace the parameters between {}.

(2) you can change 8 to the number of cores (CPUs) you wish to use.

The Output can be found in“outputDIR” folder

- “trim.log” – specifies the trimming history of each read.
- “contig.align” – specifies the read alignment to the consensus generated.

6. Parse Vicuna output using analysis script.

   (a) Copy the analysis template config file to your local directory, where the user should specify the path parameter [analysis_config].

   ```
   $ cp /seq/viral/analysis/xyang/programs/VicunAnalysis/config.txt   [analysis_config]
   ```

   (b) Modify parameters in file [analysis_config]

   - trim_log_file – the path of “trim.log” file.
   - aln_file – the path of “contig.align” file.
   - pFqDir – the same as in [vicuna_config].
   - outputDIR – the path to the output directory.

   Instructions for setting other parameters are provided in the config file.

   (c) Run analysis.

   ```
   $ /seq/viral/analysis/xyang/programs/VicunAnalysis/vicunAnalysis   [analysis_config]
   ```

   The Output can be found in“outputDIR” folder.

   - “contig.fa” – contig output in fasta format.
   - “contig.lfv.fasta” – contig output in fasta format, contains low frequent variants.

3 Quick Start – for external users

3.1 Pre-requisite

1. Installation of NCBI Toolkit 7.0.0 (download link `ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/CURRENT/`).

   -
     ```
     $ ./configure --prefix=path_to_install --with-optimization --with-mt --with-dll
     ```

     Note: path_to_install needs to be specified by the user.

   - $ make (note: this is gnu make)
   - $ make install

2. Installation of Perl (recent versions are recommended)

3. g++ compiler (recent versions are recommended)

3.2 Procedure

1. Download the Vicuna package, decompress, and "cd" into the Vicuna folder.

2. Switch to bash environment.

   ```
   $ bash
   ```

3. Export NCIB toolkit library path. Assuming you successfully installed NCBI Toolkit 7.0.0 in directory [path], then you should be able to find the library in directory "[path]/lib"

   ```
   $ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/[path]/lib
   ```

4. Run Vicuna.

   (a) Edit file "Vicuna/src/Makefile" – set the parameter MYPATH to be [path], set the parameter COMPILER to be the path of the g++ compiler you are using (you could use command "$ which g++" to find out this information).

   (b) Compile

   ```
   $ cd src
   $ make
   $ cd ../
   ```

   Note:

   - The executive file can be found in the "Vicuna/bin" folder.
   - By default the executive file is compiled with "-fopenmp" flag and named"vicuna-omp-v1.0". These settings can be changed by modifying the "Makefile".

   (c) Set basic parameters in file: "Vicuna/config/vicuna_config.txt".

   - pFqDir – the path of the input folder. This folder should contain a multiple of two paired read fastq file, ending with ".fq" or ".fastq". The user has to make sure that if there are more than 2 paired files, when loaded into the memory, two files that form pairs have to be loaded consecutively. This can be achieved by naming a pair of read files to be i_1.fq and i_2.fq, for instance, where i can be an integer or a string.
   - outputDIR – the path of the output folder.
   - MSAFileName – the path of the file stores the MSA of target genomes, recommended to be used for samples rich in contamination. We provided this file for HIV ("Vicuna/db/hiv-1B.fa"), West Nile ("Vicuna/db/wnv.fa"), Dengue ("Vicuna/db/denv.fa") and Las virus ("Vicuna/db/lasvL.fa" for L element, "Vicuna/db/lasvS.fa" for S element).

For advanced users, instructions for setting each parameter are provided in the file “Vicuna/config/vicuna_config.txt”.

(d) Execute Vicuna.

```
$ OMP_NUM_THREADS=n   ./bin/vicuna   config/vicuna_config.txt
```

(n is the number of cpus you would like to use, *e.g.* 4)

The Output can be found in“outputDIR” folder

- “trim.log” – specifies the trimming history of each read.
- “contig.align” – specifies the read alignment to the consensus generated.

5. Parsing the output of Vicuna using vicuna_analysis program.

(a) Compile

```
$ cd scripts/VicunAnalysis/
$ make clean
$ make all
$ cd ../../
```

The output will be written as “Vicuna/bin/vicunAnalysis”

(b) Set basic parameters in the config file “Vicuna/config/vanalysis_config.txt”.

- trim_log_file – the path of “trim.log” file.
- aln_file – the path of “contig.align” file.
- pFqDir – the same as in “Vicuna/config/vicuna_config.txt”.
- outputDIR – the path to the output directory.

Instructions for setting other parameters are provided in the config file.

(c) Execute

```
$ ./bin/vicunAnalysis   config/vanalysis_config.txt
```

The Output can be found in“outputDIR” folder

- “contig.fa” – contig output in fasta format.
- “contig.lfv.fasta” – contig output in fasta format, contains low frequent variants.

4 Parameter Setting

Some high level explanations for Vicuna.

1. Vicuna handles both paired and unpaired read files containing reads with variable length. Currently, paired reads have to be present as part of the input (see section **??** on how to handle only unpaired reads), and only fastq and fasta format are handled. This may change in newer version.

2. When reading files from a folder/directory, Vicuna assumes two paired read files are read in consecutively. In order to achieve this, you can give the paired files with the same prefix but different suffix. For example, in a folder we have two sets of paired files (a, b) and (c, d), then we can assign the following names to each of these files: a ← 1.p1.fastq, b ← 1.p2.fastq, c ← 2.p1.fastq, d ← 2.p2.fastq. These files should then be read in comforming alphabetical order.

3. Read ID (rID) assignment. Each read is assigned with a unique ID with the following rules: (1) the first read is assigned with ID 0. (2) for paired reads, (r_1, r_2), $rID_2 = rID_1 + 1$ if r_2 is read in after r_1. (3) if r_2 is read in right after r_1, if both are in the same file, then $rID_2 = rID_1 + 1$ if this file is unpaired, otherwise, $rID_2 = rID_1 + 2$. if they are in different files, then $rID_2 = rID_1 + 1$ if the two files are not paired,

4. For any calculation, *e.g.*, generating consensus of contig, reads are loaded in batches, with user specified batch size. This controls memory usage in case when input consists of a large number of reads.

5. Output files from Vicuna.

   - trim.log – record the trimming information.
   - contig.align – record contig alignment information.
   - contig.lfv.fasta – the fasta format of consensus sequences, retaining any low frequent length polymorphisms. The consensus in this file corresponds to "contig.align".
   - contig.fasta – consensus sequences without low frequent length polymorphisms.

Some high level explanations for VicunAnalysis.

1. Using VicunAnalysis, you can print out alignments of specific region of a specific contig of interest. Particularly, if you are interested in length polymorphic regions or low coverage regions.

2. VicunAnalysis can output the "raw" consensus sequences corresponding to the MSA of each contig output from Vicuna, or you can remove low frequent polymorphisms from the consensus.

3. Output files from VicunAnalysis.

   - contig.n.txt – record the profile of the n^{th} contig.
   - contig.lfv.fasta – the fasta format of consensus sequences, retaining any low frequent length polymorphisms. The consensus in this file corresponds to "contig.align".
   - contig.fa – consensus sequences without low frequent length polymorphisms.

See Table 1 for Vicuna program and Table 2 for VicunAnalysis program.

5 License

Please refer to license folder.

Table 1: Parameter settings for Vicuna program.

Trimming – remove known primer sequences	
vectorFileName	the path of the Fasta file that stores primer/vector sequence(s) to be removed from each read.
minMSize	if the suffix or prefix of a read matches any substring of some primer with length $\geq$ *minMSize*, the matching part is trimmed.
minInternalMSize	if an internal substring of a read matches any substring of some primer with length $\geq$ *minInternalMSize*, the full read is trimmed.
maxOverhangSize	max number of neglect-able bases for a substring in a read to be considered as suffix or prefix.
minReadSize	the min length of read to be retained before or after trimming
Profiling – identify target alike reads	
MSAFileName	the path of the Fasta file storing MSA of previously assembled target genomes
binNumber	the number of bins the MSA is divided into
kmerLength	$k$mer length
maxHD	max Hamming distance tolerated between two $k$mers
minSpan	if $\geq$ minSpan% positions of r is covered by $k$mers from the bin i, r is assigned to bin i.
rMapFileName	if specified, two output files are created, "*rMapFileName*.record.txt" records only mapped rIDs; "*rMapFileName*" is a tab delimited file, each line has three entries: (1) rID, (2) BinID, and (3) isPaired, specifying if the paired end of rID is assigned.
Contig Construction, Validation, and Extension	
$w1$	$k$mer size for the first iteration of min hash
$w2$	$k$mer size for the second iteration of min hash
Divergence	max % of divergence between read & consensus during contig validation
max_read_overhang	number of base pairs that can be ignored towards either end of a read, during contig validation. This number accounts for insufficient trimming, PCR artifacts, sequencing errors, etc.
max_contig_overhang	max length of unreliable region in either end of the consensus to be tolerated during contig merging
min_perc_polymorphism	min frequency of length polymorphic region to be considered to be part of a contig
max_variant_len	max length of any variant that will be removed b4 aligning two contigs
seed_kmer_len	seed $k$mer length for computing overlap between two contigs
min_contig_overlap	min length of overlaps between two contigs for them to be merged
min_contig_links	min number of paired links for attempting to merge two contigs
min_identity	min similarity to merge two contigs
General Parameters	
pFqDir	input folder for paired fastq files
npFqDir	input folder for unpaired fastq files
pFaDir	input folder for paired fasta files
npFaDir	input folder for unpaired fastq files
batchSize	the max number of reads to be stored in the memory
LibSizeUpperBound	upper bound of fragment size
min_output_contig_len	min length of contigs to output
outputDIR	output directory path

Table 2: Parameter settings for VicunAnalysis program.

General Parameters	
pFqDir, npFqDir, pFaDir npFaDir	same as in Vicuna
trim_log_file	“trim.log” output from Vicuna
aln_file	“contig.align” output from Vicuna
outputDIR	output directory for VicunAnalysis

Alignment output for particular region of interest	
num_region	number of regions of interest; for each region, it specifies three tab delimitated fields: contig number, start, and end positions on the contig
lfv_freq	specify low frequency length polymorphism regions % coverage compared to neighboring regions
lfv_max_freq	max length for the low frequency length polymorphism region

6 Citing Vicuna

Xiao Yang, Patrick Charlebois, Sante Gnerre, Matthew G Coole, Niall J. Lennon, Joshua Z. Levin, James Qu, Elizabeth M. Ryan, Michael C. Zody, and Matthew R. Henn (2012) *De novo* assembly of highly diverse viral populations. (in review)

7 Contact

If you have any question, please email Xiao Yang (xiaoyang@broadinstitute.org).