
Manual of VirusRecom

Detecting recombination of viral lineages using information theory

Home page: <https://github.com/ZhijianZhou01/virusrecom>

**Zhi-Jian Zhou, Chen-Hui Yang, Sheng-Bao Ye, Xiao-Wei Yu, Ye Qiu* and
Xing-Yi Ge***

Version 1.0 || May 24, 2022

Content

1.	Introduction	3
1.1.	Background	3
1.2.	Functions.....	3
2.	Download and install	4
3.	Description of parameters	4
3.1.	Getting help.....	4
3.2.	Parameters	5
4.	Example of usage.....	6
4.1.	Unaligned input-sequences.....	6
4.2.	Aligned input-sequences	8
4.3.	Output result	9
5.	A simple running example	11
6.	Bug report	14

1. Introduction

1.1. Background

Recombination is common in viruses. In general, recombination can be divided into two categories: homologous recombination including normal homologous recombination and abnormal homologous recombination, and non-homologous recombination. Homologous recombination occurs between homologous or similar sequences and plays a major role in recombination. The normal homologous recombination was defined as that the recombination region is homologous or similar. However, aberrant homologous generally results in sequence deletion or duplication due to the recombination region is not in the homologous region.

Identifying homologous recombination from highly similar sequences is a challenge, due to the uncertainty of an emerging genomic variance originating from recombination or *in-situ* mutation. Herein, we present VirusRecom, an efficient tool for recombination analysis of viral genome with high similarity based on information theory. The new method evaluates the likelihood of recombination by quantifying recombination contribution using weighted information content (WIC).

1.2. Functions

VirusRecom was written by Python 3, the releases were created by Pyinstaller. The functions are:

- (I) Calculate the recombination contribution of each polymorphic site.
- (II) Identify the recombination events and potential recombination region with p -value.
- (III) Scan the potential recombination breakpoint.

2. Download and install

VirusRecom and all the updated versions are freely available at <https://github.com/ZhijianZhou01/virusrecom>. After obtaining the program, users could directly run the program in Windows, MacOS or Linux (Ubuntu 16.04 or more) systems without installation.

In general, the executable file of VirusRecom is located at the folder of **Main**. Then, just double click the **virusrecom.exe** (windows system) or **virusrecom** (Linux or MacOS system) to start. If you could not get permission to run **virusrecom** on Linux or MacOS system, you could change permissions by **chmod -R 777 virusrecom**.

3. Description of parameters

3.1. Getting help

VirusRecom is a command line interface program, users can get help documentation of the software by entering **virusrecom -h** or **virusrecom --help**.

```
virusrecom [-h] [-a ALIGNMENT] [-q QUERY] [-l LINEAGE] [-g GAP]
[-m METHOD] [-w WINDOW] [-s STEP] [-mr MAX_REGION]
[-cp PERCENTAGE] [-cm CALIBRATE] [-b BREAKPOINT] [-bw BREAKWIN]
[-o OUTDIR] [-t THREAD] [-y Y_START]
```

☆ Example of use ☆

(1) If the input-sequence data was not aligned:

```
virusrecom -q query.fasta -l lineage_dir -g n -m p -w 100 -s 20 -t 2 -o outdir
```

(2) If the input-sequence has been aligned:

```
virusrecom -a alignment.fasta -q query_name -l lineage_name_list.txt -g n -m p -w 100 -s
20 -o outdir
```

Note: sequence file or folder need to enter their absolute path, the above is just a conceptual example.

3.2. Parameters

Parameters	Description
<i>-h, --help</i>	Show this help message and exit.
<i>-a</i>	FilePath of an aligned sequence set (*.fasta format) containing all sequences used for analysis, then the sequence alignment will be skipped. Default value is null. If “-a” parameter was used, the name of each sequence in aligned sequence set requires containing the mark (a unique string) of the lineage.
<i>-q</i>	FilePath of query lineage (usually potential recombinant, *.fasta format). Note, if the ‘-a’ parameter has been used, please enter the mark (a unique string) of query lineage here, such as ‘-q xxxx’, not a FilePath. Using ‘-q auto’ and all lineages will be scanned as potential recombinants in turn.
<i>-l</i>	DirPath of reference lineages. One sequence file (*.fasta format) per lineage, and each lineage could contain multiple sequences. Note, if the ‘-a’ parameter has been used, please enter a file path of a text file containing the mark (a unique string) of lineage here, not a DirPath.
<i>-g</i>	Gaps (-) in the alignment were used in analysis? ‘-g y’ means to reserve gaps, and ‘-g n’ means to delete gaps.
<i>-m</i>	Scanning method of recombination analysis. ‘-m p’ means use polymorphic sites only, ‘-m a’ means all the monomorphic sites and polymorphic sites.
<i>-w</i>	Number of nucleotides sites per sliding window. Note: if the ‘-m p’ has been used, -w refers to the number of polymorphic sites per windows.
<i>-s</i>	Step size of the sliding window. Note: if the ‘-m p’ has been used, -s refers to the number of polymorphic sites per jump.
<i>-mr</i>	The maximum allowed recombination region. Note: if the ‘-m p’

	method has been used, it refers the maximum number of polymorphic sites contained in a recombinant region.
<code>-cp</code>	The cutoff threshold of proportion (<i>cp</i> , default value was 0.9) used for searching recombination regions when $mWIC/EIC \geq cp$, the maximum value of <i>cp</i> is 1.
<code>-cm</code>	Whether to simply use the max cumulative WIC of all sites to identified the major parent. The default value is 'n' and means 'no'. If required, please specify ' <code>-cm y</code> '.
<code>-b</code>	Whether to run the breakpoint scan of recombination. ' <code>-b y</code> ' means yes, ' <code>-b n</code> ' means no. Note: this option only takes effect when ' <code>-m p</code> ' has been specified.
<code>-bw</code>	The window size (polymorphic sites, default value is 200) used for breakpoint scan. The step size is fixed at 1. Note: this option only takes effect when ' <code>-m p -b y</code> ' has been specified.
<code>-t</code>	Number of threads used for the multiple sequence alignments (MSA), the default value is 1.
<code>-y</code>	Specify the starting value of the Y-axis scale in the picture, the default value is 0.
<code>-o</code>	The path of the outdir of results.

4. Example of usage

The sequences data for test in the manual was stored at <https://github.com/ZhijianZhou01/virusrecom/tree/main/example>. Take the `recombination_test_data.zip` provided in the directory `example` as a demonstration.

4.1. Unaligned input-sequences

VirusRecom owns a pipeline built in to handle unaligned sequences. In this case,

multiple sequence alignment is performed by MAFFT (Katoh & Standley, 2013) with alignment strategy of “auto” parameter. In the directory `unaligned_input_sequences` of the compressed file `recombination_test_data.zip`, and the query lineage (simulated recombinant) including multiple sequences was in the file `query_recombinant.fasta`, and these reference lineages were placed in a separate directory named `lineages_dir`. For the reference lineages, sequences for each lineage need to be placed in a separate file, such as `reference_lineage_1.fasta`, `reference_lineage_2.fasta` and `reference_lineage_3.fasta` in the directory `lineages_dir`. Of note, the name of each file is an important label to distinguish different lineages. In fact, the query lineage in the test data was known recombinant because it was from the synthetic data, the major parent was `reference_lineage_1`, the minor parent was `reference_lineage_2` and the recombination region was from site 7333 to 11473 in the genome. However, here we treat the query lineage as a potential recombinant.

Before running the command of VirusRecom, let's think about the search strategy for recombination events. Firstly, we use only polymorphic sites considering that sequences from these lineages are highly similar, which means that the parameter `-m p` needs to be specified. Secondly, we do not consider gap-containing sites in this test and use the parameter `-g n`. Instead, if you consider these gap sites, you need to use the parameter `-g y`. Next, in the first run, let's try first with a window size of 100 and a step size of 20. Of note the value of “size” at this time represents the number of polymorphic sites because the `-m p` parameter has been specified. For the two parameters `-cp` and `-mr`, we use the default value of 0.9 and 1000 in this test. Besides, we also need to specify the number of threads to use for the mafft program. For example, `-t 2` means two threads will be used in the mafft program. Finally, we specify a folder to save the results by parameter `-o`.

Then, we execute the following command to detect the recombination events in query lineage:

```
virusrecom -q query_recombinant.fasta -l lineages_dir -g n -m p -w 100 -s 20 -t 2 -o out_dir
```

Of note, (i) if the current directory is not switched to `unaligned_input_sequences`, the paths of the file `query_recombinant.fasta`, the directory `lineage_dir` and the directory `out_dir` need to use absolute paths instead of relative paths; (ii) the string after each parameter cannot contain spaces.

4.2. Aligned input-sequences

In addition to unaligned input-sequences, users can also provide an independent aligned sequences file which was performed from any other alignment program, including all the sequences from the query lineage and other reference lineages. In the directory `aligned_input_sequences`, we provided an aligned sequence file named `lineages_data_alignment.fas` used for test. In the file `lineages_data_alignment.fas`, each sequence name contained the mark of the lineage name, such as “query_recombinant”, “reference_lineage_1”, “reference_lineage_2” and “reference_lineage_3”. Of note this mark can appear anywhere in the sequence name.

In addition to the aligned input-sequences, a text file containing the names of these reference lineage is required, and an example as shown in the file `reference_lineages_name.txt` in the directory `aligned_input_sequences`:

```
reference_lineage_1  
reference_lineage_2  
reference_lineage_3  
reference_lineage_4  
reference_lineage_5  
reference_lineage_6  
reference_lineage_7  
reference_lineage_8  
reference_lineage_9
```

Of note the mark names should be unique for each lineage.

Then, execute the command to detect recombination events in query lineage, for example:

```
virusrecom -a lineages_data_alignment.fas -q query_recombinant -l  
reference_lineages_name.txt -g n -m p -w 100 -s 20 -o out_dir
```

Of note (i) if the current directory is not switched to `aligned_input_sequences`, the paths of the file `lineages_data_alignment.fas`, the file `reference_lineages_name.txt` and the directory `out_dir` need to use absolute paths instead of relative paths; (ii) the string “query_recombinant” in command is the corresponding mark of query lineage in the file `lineages_data_alignment.fas`.

In fact, we recommend that users use already aligned input-data like the file `lineages_data_alignment.fas` in VirusRecom. A simple reason is that the step of multiple sequence alignment can be omitted when adjusting the parameters in VirusRecom. However, how to add the lineage name into each sequence name when the number of sequences is huge? A common approach is using your own script. However, we recommend completing the addition of mark using VirusRecom for regular users. Firstly, you can prepare the data (unaligned input-sequences) as described in section `4.1. Unaligned input-sequences`, and then submit them to VirusRecom to run and the parameters can be arbitrarily configured first. Then, an intermediate file named `*__merge.fasta` can be found in the directory `run_record` when the log output from the MAFFT software appears. Of note the file `*__merge.fasta` is not aligned. You can use the generated file of aligned `*__merge_mafft.fasta` after MAFFT finishes running, or use other software to align the file `*__merge.fasta`.

4.3. Output result

Take the parameter configuration in the section `4.1. Unaligned input-sequences` above

as an example, and the output directory is `out_dir`. There are three subdirectories and two aggregated reports in the directory `out_dir`. Now, we first introduce three subdirectories, including the directory `run_record`, the directory `WICs of sites` and the directory `WICs of slide_window`.

- (I) In the directory `run_record`, the alignment file created by MAFFT is reserved. If `-g n` is specified, and the file `Record of deleted gap sites_*.txt` containing all the gap sites will be created. Besides, If `-m p` is specified, and the file `Record of same sites in aligned sequence_*.txt` containing all the same sites will be created.
- (II) In the directory `WICs of sites`, the file `*_WIC contribution from lineage in sites.pdf` and the file `*_WIC contribution from lineage in sites.xlsx` are used to record the WIC value for each site.
- (III) In the directory `WICs of slide_window`, the file `*_WIC contribution from lineage in sliding window.pdf` and the file `*_WIC contribution from lineage in sliding window.xlsx` are used to record the mean WIC of each sliding window. The user can fine-tune the window size and step size according to the density of points in the generated graph. In general, very dense points means that the noise is too high and the window size can be increased appropriately in next scan.

In addition to the three subdirectories above, VirusRecom provides two summary files. The file `_Possible recombination event in query_recombinant_conciseness.txt` only retains results of recombination events with p-values less than 0.05, and the file `_Possible recombination event in query_recombinant_detailed.txt` shows those results with p-values greater than 0.05. In fact, recombination events with p-values below 0.001 are less reliable.

Take the file `Possible recombination event in query_recombinant_conciseness.txt` generated by using the test data and the above parameters in section 4.1. Unaligned input-sequences as an example, the output results are as follows:

Possible major parent: reference_lineage_1(global mWIC: 1.8976186779157704)

Other possible parents and significant recombination regions ($p < 0.05$):

reference_lineage_2 7237 to 11539(mWIC: 1.9553354371515168), p_value:
7.831109305531908e-06

In this output report, the major parent of query lineage was `reference_lineage_1` and the minor parent was `reference_lineage_2`, and the recombination region was site 7237 to 11539 and the p-value was 7.83e-06. The identified recombination event was relatively close to the actual, and the error of the recombination boundary is also acceptable.

Besides, if `-b y` is specified, for example, `-b y -bw 200`, then VirusRecom will perform the search of recombination breakpoint. The negative logarithm of p -value in each site is in the file `*_lg(p-value) for potential breakpoint.pdf` and the file `*_lg(p-value) for potential breakpoint.xlsx`.

5. A simple running example

https://github.com/ZhijianZhou01/virusrecom/tree/main/example/recombination_test_data.zip

Take synthetic data above as an example, the directory `aligned_input_sequences` in the compressed file contains two files, one is an aligned sequence dataset and the other is a text file containing the names of different lineages. The sequence dataset is already aligned and contains ten lineages, and each lineage owns 100 sequence samples. Besides, the known recombinant lineage owns the character string of “query_recombinant” in sequence names, and each other lineage has its own label in

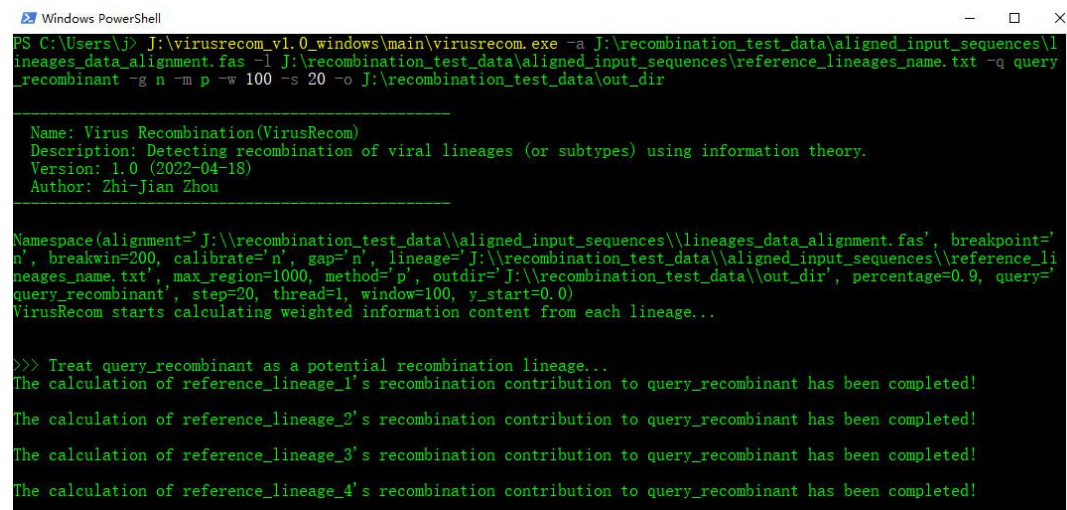
sequence names.

Then, we begin our analysis of potential recombination lineage. We take the windows system as an example, and assume that the directory `aligned_input_sequences` is located in the directory `J:\recombination_test_data`. Besides, assume that the path of the executable of virusrecom is `J:\virusrecom_v1.0_windows\main\virusrecom.exe`

(i) Open the Windows PowerShell, and execute the following command:

```
J:\virusrecom_v1.0_windows\main\virusrecom.exe -a  
J:\recombination_test_data\aligned_input_sequences\lineages_data_alignment.fas -l  
J:\recombination_test_data\aligned_input_sequences\reference_lineages_name.txt -q  
query_recombinant -g n -m p -w 100 -s 20 -o J:\recombination_test_data\out_dir
```

(ii) Then, virusrecom begins calculating the recombination contribution of each reference lineage to the query lineage.



```
PS C:\Users\j> J:\virusrecom_v1.0_windows\main\virusrecom.exe -a J:\recombination_test_data\aligned_input_sequences\l  
ineages_data_alignment.fas -l J:\recombination_test_data\aligned_input_sequences\reference_lineages_name.txt -q query  
_recombinant -g n -m p -w 100 -s 20 -o J:\recombination_test_data\out_dir  
  
-----  
Name: Virus Recombination(VirusRecom)  
Description: Detecting recombination of viral lineages (or subtypes) using information theory.  
Version: 1.0 (2022-04-18)  
Author: Zhi-Jian Zhou  
-----  
  
Namespace(alignment='J:\\recombination_test_data\\aligned_input_sequences\\lineages_data_alignment.fas', breakpoint='  
n', breakwin=200, calibrate='n', gap='n', lineage='J:\\recombination_test_data\\aligned_input_sequences\\reference_li  
neages_name.txt', max_region=1000, method='p', outdir='J:\\recombination_test_data\\out_dir', percentage=0.9, query='  
query_recombinant', step=20, thread=1, window=100, y_start=0.0)  
VirusRecom starts calculating weighted information content from each lineage...  
  
>>> Treat query_recombinant as a potential recombination lineage...  
The calculation of reference_lineage_1's recombination contribution to query_recombinant has been completed!  
The calculation of reference_lineage_2's recombination contribution to query_recombinant has been completed!  
The calculation of reference_lineage_3's recombination contribution to query_recombinant has been completed!  
The calculation of reference_lineage_4's recombination contribution to query_recombinant has been completed!
```

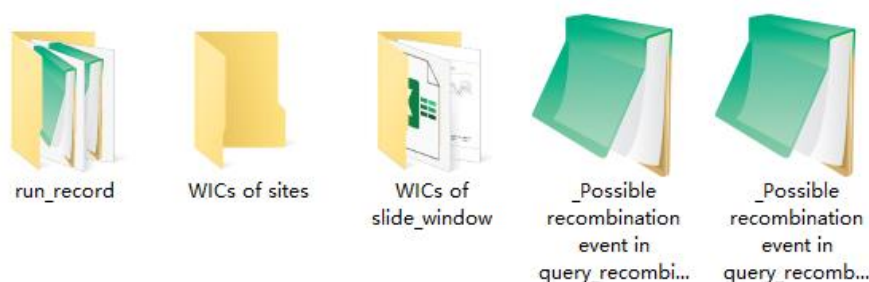
(iii) After the finish of running, a concise report is printed reporting recombination events with p-values less than 0.05.

```
Possible major parent: reference_lineage_1(global mWIC: 1.8976186779157704)  
Other possible parents:  
Possible recombination region map at aligned genomes:  
reference_lineage_2 [['7237 to 11539(mWIC: 1.9553354371515168)', 'p_value: 7.831109305531908e-06']]
```

(iv) The aggregated results is at the file **Possible recombination event in query_recombinant_conciseness.txt**.

```
1 Possible major parent: reference_lineage_1(global mWIC: 1.8976186779157704)
2
3 Other possible parents and significant recombination regions (p<0.05):
4 reference_lineage_2 7237 to 11539(mWIC: 1.9553354371515168), p_value:
5 7.831109305531908e-06
6
6 Significance test of recombinant regions using Mann-Whitney-U test with
two-tailed probabilities, p-value less than 0.05 indicates a significant
difference.
```

The intermediate files generated during the running process are located in the directory **run_record**.



(v) The matrix data generated by the sliding window operation is in the directory **WICs of slide_window**, and users can use these raw data to draw graphs. In fact, virusrecom comes with a drawing function and provides drawn graphics (**WICs of slide_window/_query_recombinant_WIC contribution from lineage in sliding window.pdf**), and they might be as follows:



If the user thinks that the color of this picture is not good, the user can use the original matrix data provided by virusrecom to redraw.

6. Bug report

You can tell us any problems which you encounter in usage, and so that we can further improve VirusRecom. Please submit your question in [GitHub issue](https://github.com/ZhijianZhou01/virusrecom/issues) (<https://github.com/ZhijianZhou01/virusrecom/issues>) of VirusRecom or send email to zjzhou@hnu.edu.cn.

References

- Katoh, K., & Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol Biol Evol*, 30(4), 772-780.
doi:10.1093/molbev/mst010