Manual for

Next generation sequencing Pair-end Reads Joining (PERJ) software

version V 1.1

Xuewen Wang, PhD

Email: xwwang@ymail.com

2013-10-12th

Contents

1. Algorithm and function of PERJ

A. What is PERJ

PERJ is the acronym of next generation sequencing (NGS) Pair-End Reads Joining (PERJ) software. PERJ software, version 1.1, will join each pair of pair-end reads and fill in Ns between two reads to construct DNA fragments just like before sequencing. The joined fragments information will output to result files in fasta or fastq format. The processing speed is over 1 million pairs of reads per minute.

Updates in this version:

1. add function to accept stack output-format reads as input files. The option "-stacks 1|0 " was added.

2. add quality value for filled Ns between left and right reads.

The NGS will generate millions of sequence reads which speeds up the genomics and transcriptome research in recent years. Before sequencing, it is necessary to break long DNA fragment to short fragment and add adaptors to make a sequencing library. For each small DNA fragment, the length was decided by the length of the sequencing DNA library. NGS sequencer such as Illumina will sequence both end, also called pair end, of each DNA fragment. The sequenced result of each fragment called read. The read of all DNA fragment of forward orientation will be save in one file and that of reverse orientation will be save in another file. The format of saved reads is in fastq.

In some subsequent processing, the sequences will be needed to join to one simulated DNA fragment like original small DNA fragment with sequenced nt in the both ends. The NGS output is too big to get pair end joined manually. Currently the other softwares/script can't do this kind of tasks due to a fact of complicate problems of fastq data. Currently the fastq function of Bio::SeqIO module http://cpan.uwinnipeg.ca/htdocs/BioPerl/Bio/SeqIO/fastq.pm.html (**Bio::SeqIO::fastq)** from CPAN is not stable regarding parsing the quality file. The software PERJ provided here solved this problem and can accurately processes the quality value of fastq reads.

Most of the NGS sequencing reads are the pair-end reads from Illumina instrument. One is sequenced from the forward direction while the other paired read is sequenced from reverse direction (Figure 1). These raw reads are in fastq format.

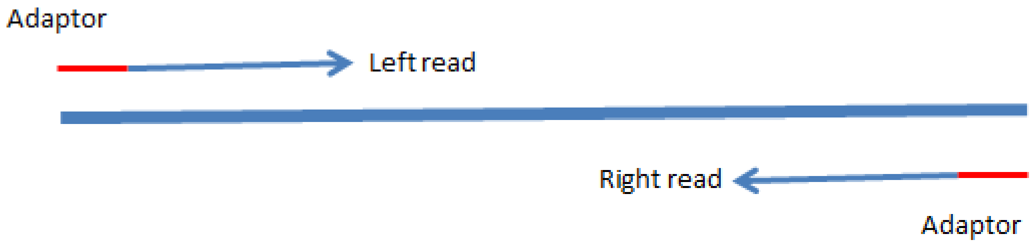
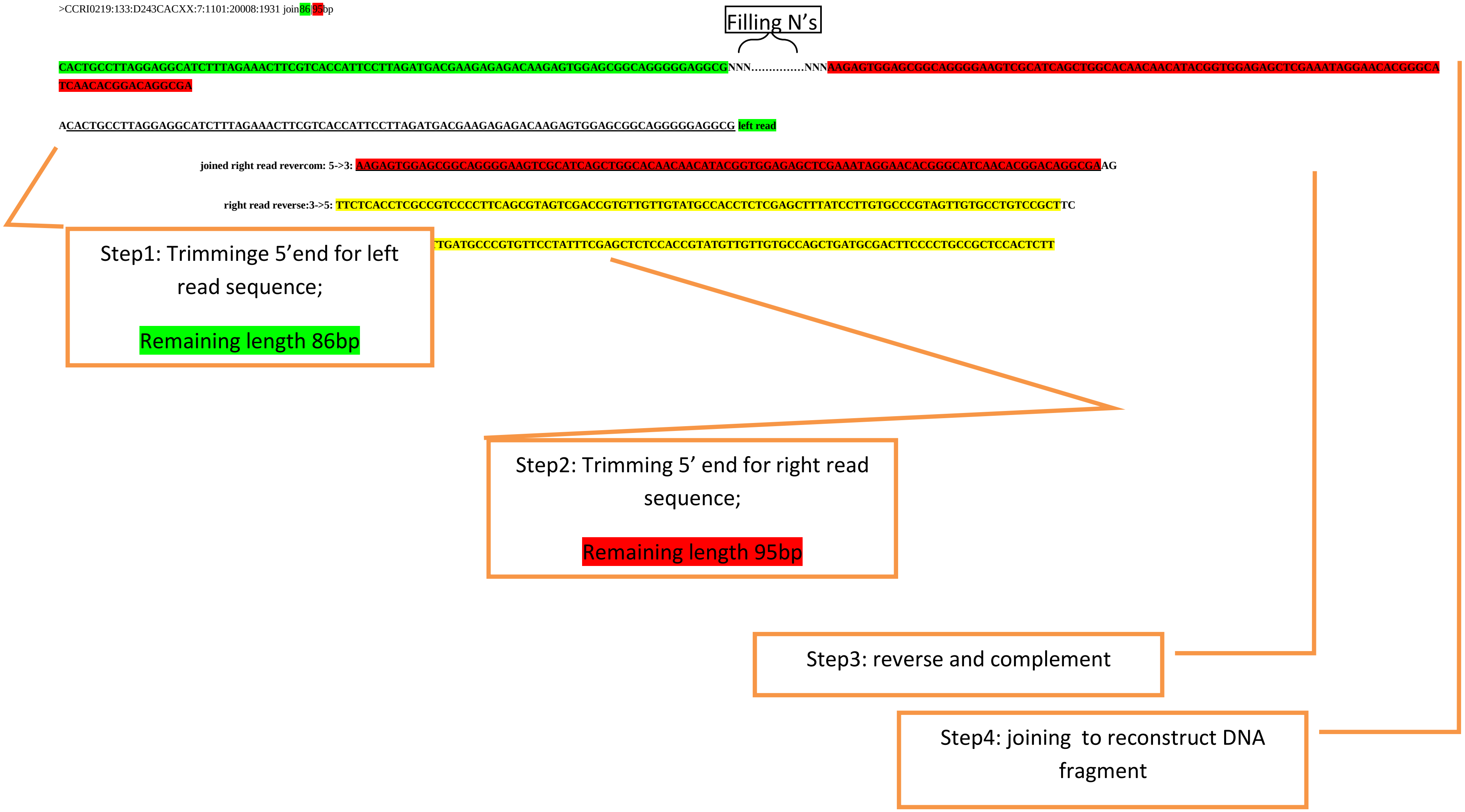


Figure 1 image showing pair-end reads

The novel software PERJ provided here will reverse and complement the right reads and then join the processed right reads to the left reads in matched pair-pattern which is identified by Illumina reads ID. The unknown sequence between two ends will be fill optionally by Ns.

There are more available functions with options in this software including producing the joined reads in standard fasta format or standard fastq format, generating statistic information, trimming the adaptor sequences, calculating the read length after trimming, showing length in the joined read after the read ID, outputting quality value for each joined nr base. Some unknown nr bases (N) can be filled in between left and right reads according to the length of the known sequencing library.

An example showing pair-end reads were joined by PERJ.



>CCRI0219:133:D243CACXX:7:1101:20008:1931 join86|95bp

Filling N's

CACTGCCTTAGGAGGCATCTTTAGAAACTTCGTCACCATTCCTTAGATGACGAAGAGAGACAAGAGTGGAGCGGCAGGGGGAGGCGNNN...............NNNAAGAGTGGAGCGGCAGGGGAAGTCGCATCAGCTGGCACAACAACATACGGTGGAGAGCTCGAAATAGGAACACGGGCATCAACACGGACAGGCGA

ACACTGCCTTAGGAGGCATCTTTAGAAACTTCGTCACCATTCCTTAGATGACGAAGAGAGACAAGAGTGGAGCGGCAGGGGGAGGCG left read

joined right read revercom: 5->3: AAGAGTGGAGCGGCAGGGGAAGTCGCATCAGCTGGCACAACAACATACGGTGGAGAGCTCGAAATAGGAACACGGGCATCAACACGGACAGGCGAAG

right read reverse:3->5: TTCTCACCTCGCCGTCCCCTTCAGCGTAGTCGACCGTGTTGTTGTATGCCACCTCTCGAGCTTTATCCTTGTGCCCGTAGTTGTGCCTGTCCGCTTC

TTGATGCCCGTGTTCCTATTTCGAGCTCTCCACCGTATGTTGTTGTGCCAGCTGATGCGACTTCCCCTGCCGCTCCACTCTT

Step1: Trimminge 5'end for left read sequence;

Remaining length 86bp

Step2: Trimming 5' end for right read sequence;

Remaining length 95bp

Step3: reverse and complement

Step4: joining to reconstruct DNA fragment

B. Workflow of PERJ

The following chart (Figure 1) shows the workflow of PERJ.

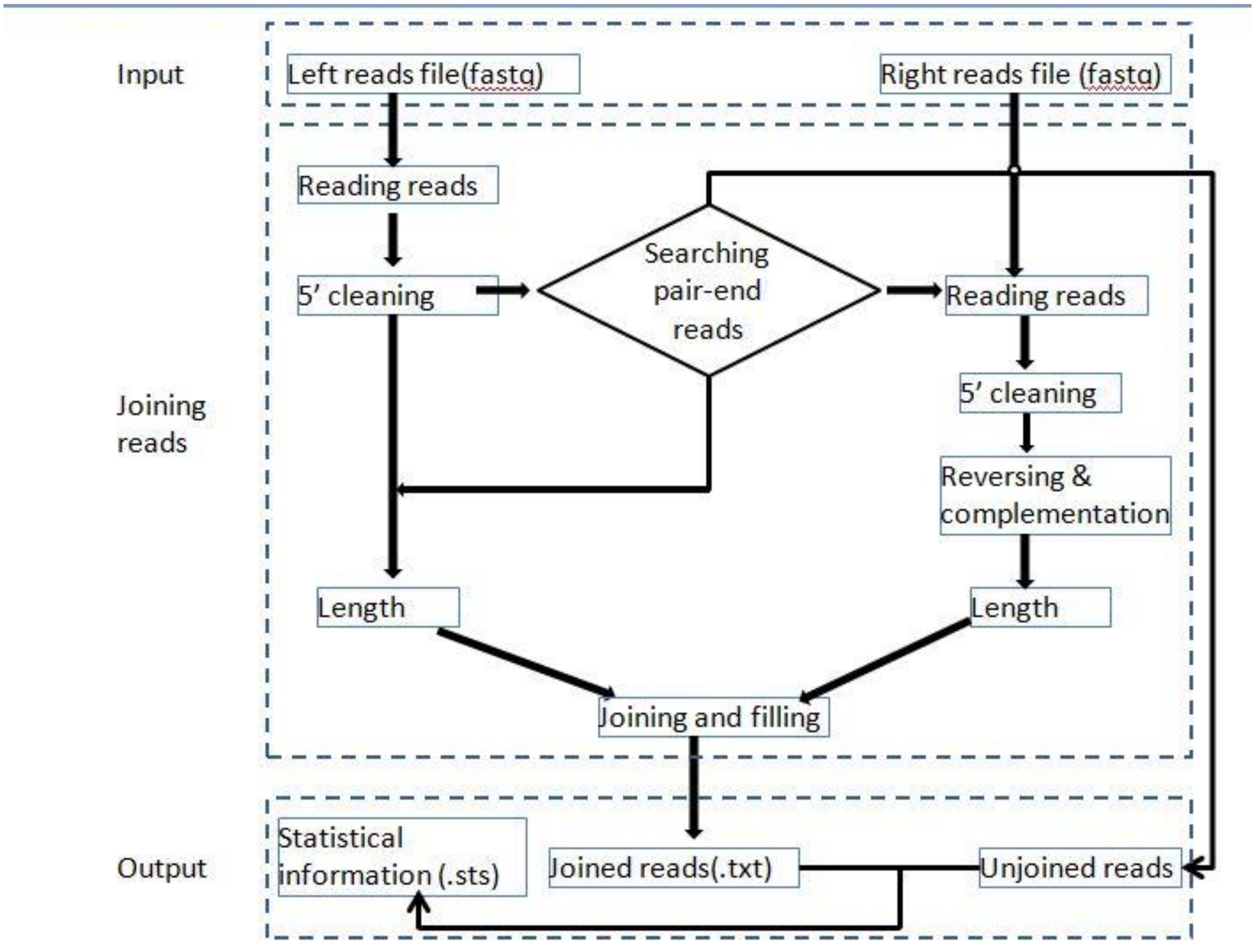


Figure 2 The workflow of PERJ

C. Running requirement

The recommended requirements for software PERJ is minimum 1G RAM, CPU 1G or higher. A minimum 5M disk space is needed. The software PERJ will use around 1.5x memory of the size of the input sequence file. PERJ was tested and running smoothly in platforms including Windows XP, 7, 8； MAC OS 10.7 or higher, Linux 5.5 or higher version. PERJ may work well in other platforms.

The software PERJ was written in Perl 5 and Java 7.0 so its running needs Perl running environment. Perl version 5.8 or higher version is needed. For Microsoft Windows, it is required to test perl installed or not. for other computing system, the Perl is installed by default usually. The Java run time environment is also required which can be download freely from http://www.java.com. However, most computers have already Java run time software installed. Type the following command and if you get the Perl version information it means you have Perl 5 installed. Otherwise go to website http://www.perl.org/, and download the Perl language for free. The command is

```
perl -v
```

```
This is perl, v5.8.8 built for x86_64-linux-thread-multi


Copyright 1987-2006, Larry Wall


Perl may be copied only under the terms of either the Artistic License or the

GNU General Public License, which may be found in the Perl 5 source kit.


Complete documentation for Perl, including FAQ lists, should be found on

this system using "man perl" or "perldoc perl".  If you have access to the

Internet, point your browser at http://www.perl.org/, the Perl Home Page.
```

D. Input files

The input files for this software are two raw or clean fastq files in four lines of standard fastq format. There is no length limitation for each read so the software is suitable for any pair-end reads at any length. The identifier, shaded in green, from the fastq read data should be the standard fastq format or in the following Illumina format or latest format or modified NCBI fastq format. The grey shaded part is optional.

Standard fastq format: @<seqname>\n<seq>\n+[<seqname>]\n<qual>\n

Illumina fastq format:

@CCRI0219:133:D243CACXX:7:1101:20008:1931 1:N:0:

Or Latest Illumina fastq format (casava 1.8 or higer version):

@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG

NCBI/EBI Sequence Read Archive fastq format(- added to remove space):
@SRR001666.1_071112_SLXA-EAS1_s_7:5:1:817:345 length=36

or stacks package output reads format:
@5_1101_3427_2344_1

E. Output files

There are two result files generated by software PERJ . The output file name is given by user through -o option. One output is the joined reads and the other output (.sts) is the statistical information of options, input and output files. For example, if the user set the output file name as **testOUT.txt**. The output files will be:

testOUT.txt

testOUT.txt.sts

All output files are in plain text format so it is easy to use for subsequent scripts or softwares. To view the top 10 lines of output file results, typing the following command in Linux or Mac OS:

```
head testOUT.txt
head testOUT.txt.sts
```

For windows system, just use any text or word processor to view the results in output files.

2. Installation

The software/software PERJ is to join pair-end reads from Illumina sequencing raw data in fastq format. Just a very simple installation, copy and unzip, is needed. Here is an example to do it. Copy or download the zip file, PERJ.zip, to your computer. Simply unzip all files in PERJ.zip. Here are the instructions.

A. Mac OS, Unix system

The followings are the commands to do these. Just simply type the command and press enter in the Linux / Unix command console.

```
unzip PERJ.zip
```

B. Windows system

Double click the zip file, then following instruction to extract all files to the folder prompted automatically.

3. Quick start

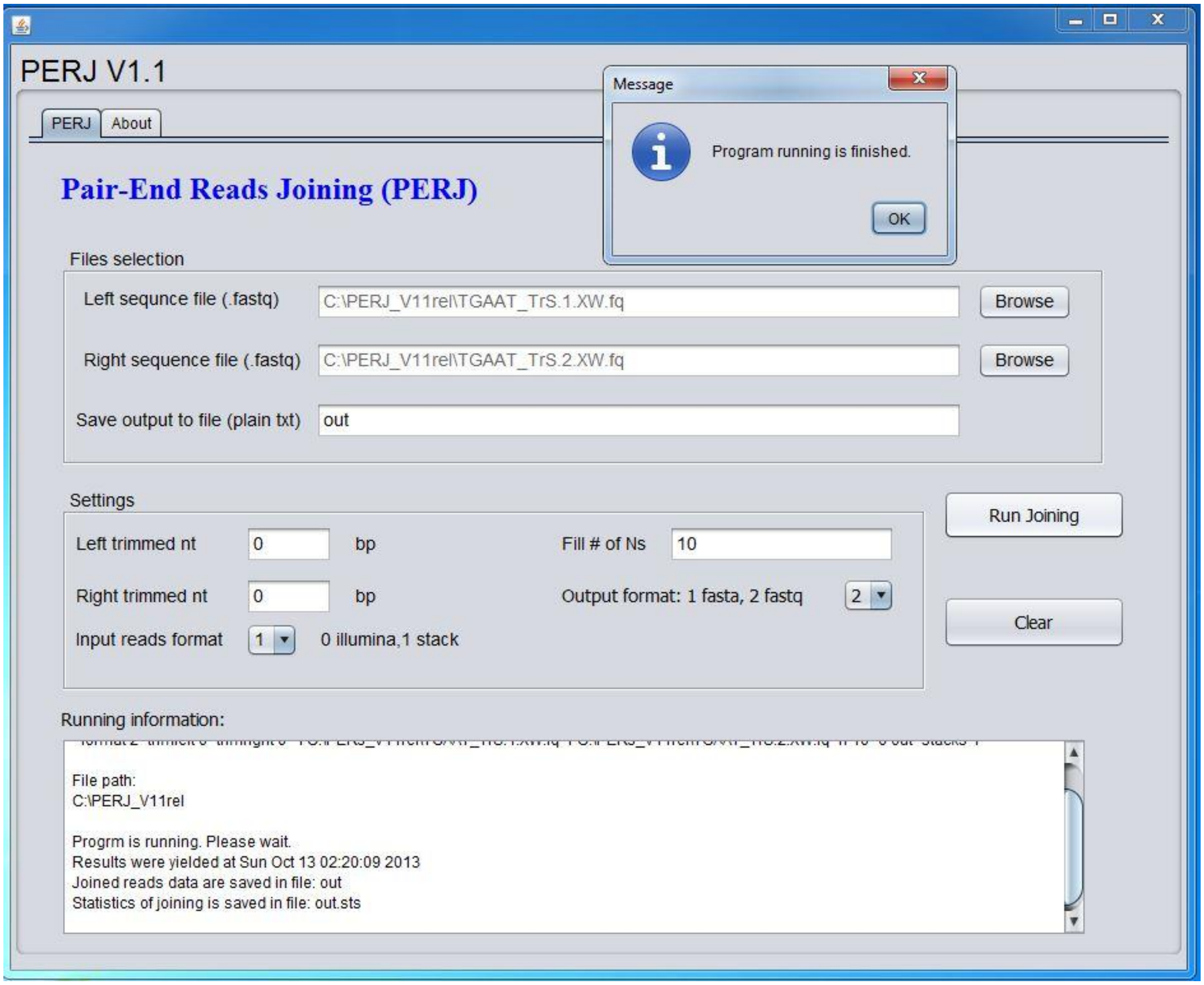
The software PERJ can be run in graphic mode or command mode. Users can choose one of these mode to run PERJ according to their preferences. Both modes can be run in Linux, MAC OS and Windows platforms.

A. Graphic running mode

The software PERJ can be run in graphic mode (Figure 3) through

double clicking the file called PERJ.jar.

The graphic interface allow the user to choose left and sequence files by click "browse" button. The results file name can be given by users or use the default result file name called "out". The other settings are positive integer which allowed the user to decide or use the default value. Once the button "Run joining" is clicked, the software PERJ will join the reads from both input files according to giving settings. A window bellow will give the running information of PERJ including settings and working path and so on. Once PERJ completes the joining, a promotion window will come up and tell the user the joining is finished. More information on how to set the settings is available in subsequent section "Advanced running".



Message

Program running is finished.

Pair-End Reads Joining (PERJ)

Files selection

Left sequnce file (.fastq): C:\PERJ_V11rel\TGAAT_TrS.1.XW.fq

Right sequence file (.fastq): C:\PERJ_V11rel\TGAAT_TrS.2.XW.fq

Save output to file (plain txt): out

Settings

Left trimmed nt: 0 bp

Right trimmed nt: 0 bp

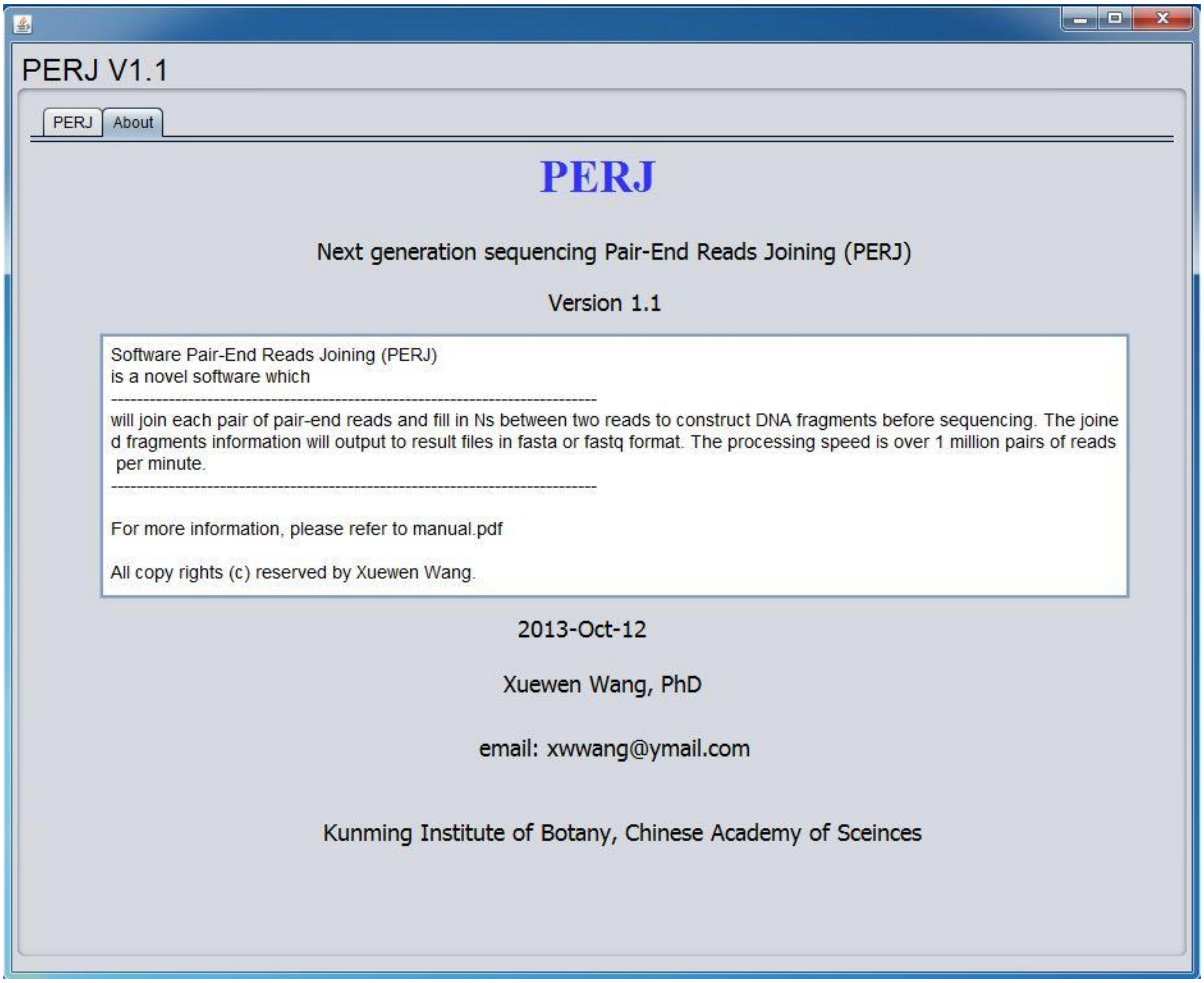
Input reads format: 1 — 0 illumina,1 stack

Fill # of Ns: 10

Output format: 1 fasta, 2 fastq: 2

Running information:

```
-format 2 -trimleft 0 -trimright 0 -f C:\PERJ_V11rel\TGAAT_TrS.1.XW.fq -r C:\PERJ_V11rel\TGAAT_TrS.2.XW.fq -n 10 -o out -stacks 1

File path:
C:\PERJ_V11rel

Progrm is running. Please wait.
Results were yielded at Sun Oct 13 02:20:09 2013
Joined reads data are saved in file: out
Statistics of joining is saved in file: out.sts
```



PERJ

Next generation sequencing Pair-End Reads Joining (PERJ)

Version 1.1

Software Pair-End Reads Joining (PERJ)
is a novel software which
----
will join each pair of pair-end reads and fill in Ns between two reads to construct DNA fragments before sequencing. The joine
d fragments information will output to result files in fasta or fastq format. The processing speed is over 1 million pairs of reads
per minute.

For more information, please refer to manual.pdf

All copy rights (c) reserved by Xuewen Wang.

2013-Oct-12

Xuewen Wang, PhD

email: xwwang@ymail.com

Kunming Institute of Botany, Chinese Academy of Sceinces

Figure 3 The graphic mode of software PERJ

B. Command running mode

This works for any of the Unix, MAC and Windows systems. In the command line console, go to the folder which PERJ is located. Here we go to example folder PERJ. Here is the command:

```
cd PERJ
```

The Illumina pair-end reads raw sequences are stored in two fastq/fastq files. One file is for forward/left reads and the other one is for reverse/right reads. Both paired reads shared a unique illumine ID followed by other tags. For a fast start of this software PERJ, we use these two files as the input files and output the joined reads with/without quality to a result file. Here we use the test sequence files with real Illumina data coming with the software PERJ package. These two example files with fastq (fq) data in standard illumina format are:

TTGGATGG_1_48.fq

TTGGATGG_2_48.fq

We treat **TTGGATGG_1_48.fq** as left input sequence file and **TTGGATGG_2_48.fq** as the right input sequence file.

On addition, there are two other reads files coming with the software which are in stacks output reads format. These files are:

TGAAT_TrS.1.XW.fq

TGAAT_TrS.2.XW.fq

The result after running this software PERJ will be saved in a file we just simply called **testOUT.txt** or any name you like in the working directory. The running command is as simple as the following first line and example running command in the following 2nd line.

```
perl PERJ.pl  -l leftseqFile  -r rightseqFile -o resultFile
```

```
perl PERJ.pl  -l TTGGATGG_1_48.fq  -r TTGGATGG_2_48.fq -o testOUT.txt
```

After that, you will find two new result files were produced by software PERJ. The files were called:

testOUT.txt

testOUT.txt.sts

The results in testOUT.txt are the joined reads data. It looks like:

```
>CCRI0219:133:D243CACXX:7:1101:20008:1931 join87|97bp

ACACTGCCTTAGGAGGCATCTTTAGAAACTTCGTCACCATTCCTTAGATGACGAAGAGAGACAA
GAGTGGAGCGGCAGGGGGAGGCGAAGAGTGGAGCGGCAGGGGAAGTCGCATCAGCTGGCACAAC
AACATACGGTGGAGAGCTCGAAATAGGAACACGGGCATCAACACGGACAGGCGAAG

>CCRI0219:133:D243CACXX:7:1101:1671:2018 join87|97bp

AAACGTCAAGAACATTTTGCCGAACCCGATACACCATCTATCCCAATAGGAATTTTAGTCCACA
AAACCCTGTAGAGAAGAGAGAGACCATCTATCCCAATAGGAATTTTAGTCCACAAAACCCTGTA
GAGAAGAGAGAGAACCGCATGAAATTCTAGCTAGAGCATCCCACCATTTTTTGCTA

…

>CCRI0219:133:D243CACXX:7:1101:3709:2304 join87|97bp

CGCCTCGCCGAGGAGGAGCTCGTTGGTGTGGGGCCCGTAGACATCGGAGGGGTCGAGGAGGGGG
ACGCCGAGATCGGAAGAGCGGGTCGACGCTCTTCCGATCTCCATCCAATGCAGCGCCTCGCCGA
GGAGGAGCTCGTTGGTGTGGGGCCCGTAGACATCGGAGGTGTCGAGGAGGGTGACG
```

The last part starting from join after sequence ID name are the length of trimmed reads from the left input file and then (|) the right input file.

The results in testOUT.txt.sts are the statistic data. It looks like:

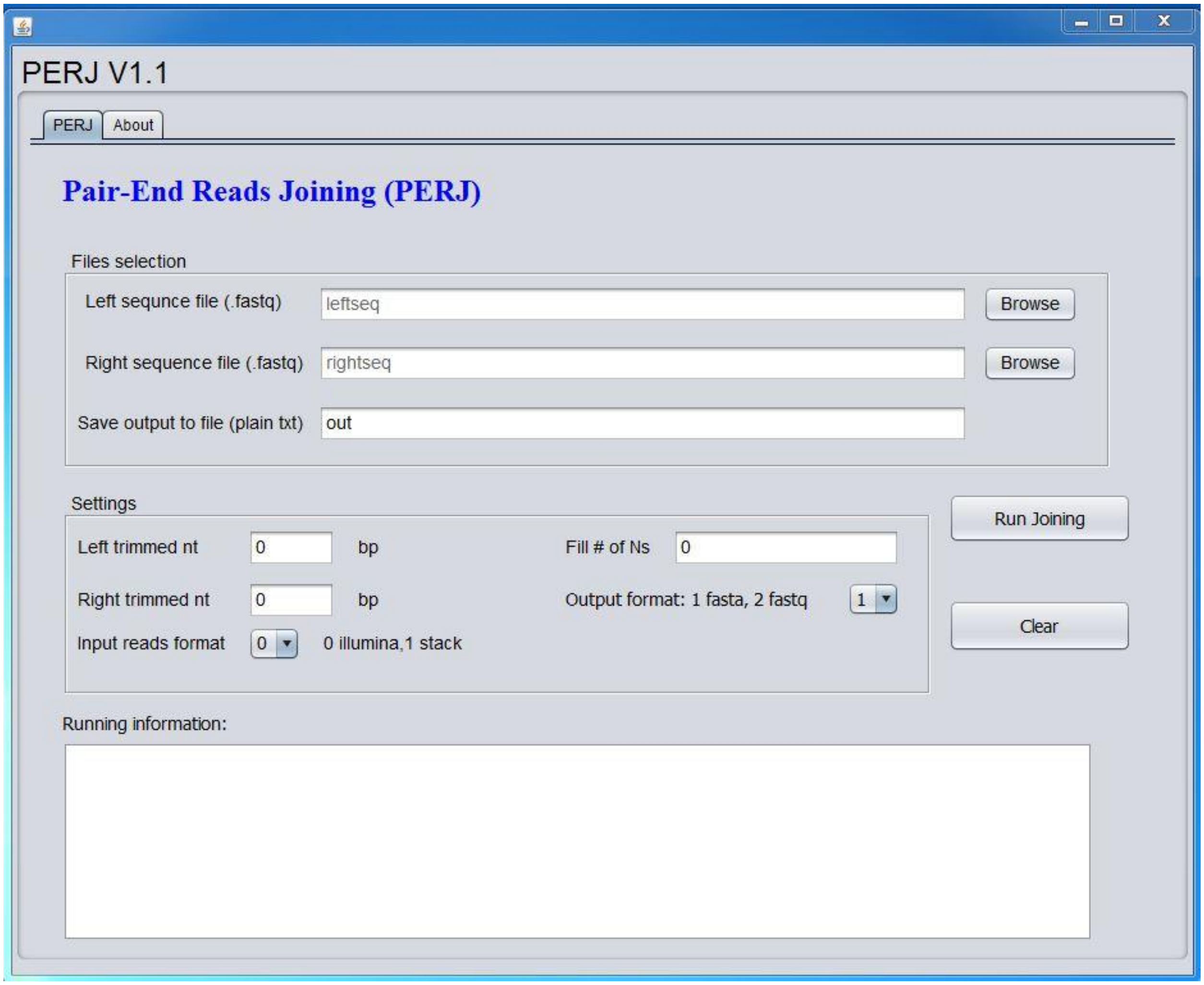
```
PERJ was programmed by Xuewen Wang, for supporting email : xwwang@ymail.com

PERJ was run and results were yielded at Sat Oct  12 15:37:40 2013

options used: -l TTGGATGG_1_48.fq -r TTGGATGG_2_48.fq -trimleft 0 -trimright 0  -n 0 -
stacks 0 -format 1 -o testOUT.txt.

Total reads in TTGGATGG_1_48.fq are 12 .

Total reads in TTGGATGG_2_48.fq are 12 .

Total joined reads in testOUT.txt are 12 .
```

If you want more options to control the joining of reads, please refer to next section called "Advanced running".

4. Advanced running

A. Graphic mode

There are several settings in the graphic mode (Figure 4). The Table 1 gives the meaning and how to set the settings in graphic mode.



PERJ V1.1

Pair-End Reads Joining (PERJ)

Files selection

Left sequnce file (.fastq): leftseq

Right sequence file (.fastq): rightseq

Save output to file (plain txt): out

Settings

Left trimmed nt: 0 bp

Right trimmed nt: 0 bp

Input reads format: 0 (0 illumina,1 stack)

Fill # of Ns: 0

Output format: 1 fasta, 2 fastq: 1

Running information:

Figure 4 Advanced settings for Graphic interface parameters

Table 1 Meaning of parameters in graphic interface

Settings:	Values
Left sequence file	the fastq file name of the left (l)/forward reads. If file name has space, put

	"" to either side of the file name. It can be selected via click button "Browse".
Right sequence file	the fastq file name of the right (r) /reverse reads. It can be selected via click button "Browse".
Left trimmed nt	Integer number, the length of nt to be removed in the 5 end of the left/forward reads. e.g. -trimleft 5. Default is 0. For setting, to type the number in the box.
Right trimmed nt	Integer number, the length of nt to be removed in the 5 end of the right/reverse reads. e.g. -trimright 5. Default is 0. For setting, to type the number in the box.
Save output to file	The value is the file name to store the results of joined reads. For setting, to type the file name in the box.
Output format	The value is 1 or 2. Value 1 will produce joined read in fasta format. Value 2 will produce joined read in fastq format. Default is 1. Select 1 or 2 via click.
Fill # of Ns	Positive integer number, to fill in N for -n value times between left and right reads, e.g. 10 Ns filling, default 0 (no filling). For setting, to type the number in the box.
Input reads format	Value is 1 or 0. Value 1 means input sequence files are from previous 'stacks' output,in 'stacks' output format. Value 0 means the input sequence files are not in stacks output format. Default is 0.

B. Command mode

An advanced run command (Table 2) for PERJ is:

```
perl PERJ.pl -l left_seq.fq -r right_seq.fa.fq -trimleft left_trim_length -trimright
right_trim_length  -format Output_fast1_fastq2 -n No_of_NsFilling -stacks
inputSeq_stackFormat1_not0 -o Results_OutputfileName
```

The option should be [-option value], a space needed between –option and its value, pair. e.g. -trimleft 1 . The order of different command_value pairs is flexible and can be put in any order.

i. standard illumina reads as input files

If the input files of reads are standard illumina read files, the command should look like the following command. Note for illumina reads, the "-stacks 0" should be used or to remove all words of "-stacks 0" because default setting for -stacks is also 0.

```
perl PERJ.pl -l TTGGATGG_1_48.fq -r TTGGATGG_2_48.fq -o testOUT.txt  -trimleft 1 -trimright 2  -format 1 –n 0 -stacks 0
```

or

```
perl PERJ.pl -l TTGGATGG_1_48.fq -r TTGGATGG_2_48.fq -o testOUT.txt  -trimleft 1 -trimright 2  -format 1 –n 0
```

ii. stack format reads as input files

If the input files of reads are from software stack package output, the command should look like the following example of run. Note, the "-stacks 1" should be used for stack format. Otherwise, no reads will be joined. The example input reads files (included in the release) called TGAAT_TrS.1.XW.fq and TGAAT_TrS.2.XW.fq are in stack format (refer to section D Input files in page 7).

1) output joined reads in fastq format

If the expected read output from PERJ should be in fastq format, just set the option value for -format to -format 2 in the command. for example:

```
perl PERJ.pl -l TGAAT_TrS.1.XW.fq -r TGAAT_TrS.2.XW.fq -trimleft 1 -trimright 2  -format 2 -n 100 -stacks 1 -o testOUT_stack.txt
```

The output in testOUT_stack.txt will look like:

```
@5_1101_3427_2344 join80|90bp

ACAAGAACTATCTGTTGGCAAACAAGCGATCAGTGGTTCTACGGCAAGGTTGGAAGAAACCCTC
GGAAAAGAAATTGATGNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTCTTCAAGCGAC
TTAGACCATGTTGTGGATGCACAGATGGCGGAAGCCCATGCTTTGGAGGGACTGCAACTGGCAA
CCCACATAGGGTGT

+

JJJIJGHIHIIIIJJJJJJJJGEIIIDIIGGGICGIFHGGIIIJBH@EHCCCFED;@CEEEDAB
BDDDDD?BDDDDDCC@BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCCCDBDDDDCAD
CCCAADDECCDFDCBEEHHGHGHIIIJGGHGCFIH@JIEJIGHIJJIGEIHIIIJJJJJIIJJH
JJIIJJHHHHGFFF

@5_1101_7543_2367 join80|90bp

TACTCTAGTTTATTGTGACAATGTCAATACTTCATACCTCTCGGCTAATTCTGTCACAAGAAAA
GGGGGTTATATGTAACNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNATTTTCTAATAT
AATTCTTTGTTGCTGCCCTATTGCACTAAAGGTATGATGTTGATGCCTTTATATCGAAAAGAAA
TTTTGAAACTGGTA

+

G>CFHGHIIIIGHEHFAHHHIEEHIGICIBEGGG@>GGIEEGGGHGGCGB4B)8C)=D@FHGDC
>A?#############BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB####AC>;>>>6
666)..?C=;;D@D@>:@@=8.8/.??4<FF9D<AGGEDCFB*F=DD?@EEBF7@>HFCA::DF
EBFC4GHFC<:DD@

(...remaining lines are not shown.)
```

The results in testOUT_stack.txt.sts are the statistic data. It looks like:

```
PERJ.pl was programmed by Xuewen Wang, for supporting email : xwwang@ymail.com

PERJ.pl was run and results were yielded at Sun Oct 13 00:48:26 2013

options used: -l TGAAT_TrS.1.XW.fq -r TGAAT_TrS.2.XW.fq -trimleft 1 -trimright 2 -format 2 -n 100 -stacks 1 -o testOUT_stack.txt.

Total reads in TGAAT_TrS.1.XW.fq are 10 .

Total reads in TGAAT_TrS.2.XW.fq are 10 .

Total joined reads in testOUT_stack.txt are 10 .
```

2) output joined reads in fasta format

If the expected output from PERJ should be in fast format, just set the option value for -format to -format 1 in the command. for example:

```
perl PERJ.pl -l TGAAT_TrS.1.XW.fq -r TGAAT_TrS.2.XW.fq -trimleft 1 -trimright 2  -format 1 -n 100 -stacks 1 -o testOUT_stack.txt
```

The output of the joined reads in file testOUT_stack.txt will look like:

```
>5_1101_3427_2344 join80|90bp

ACAAGAACTATCTGTTGGCAAACAAGCGATCAGTGGTTCTACGGCAAGGTTGGAAGAAACCCTC
GGAAAAGAAATTGATGNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTCTTCAAGCGAC
TTAGACCATGTTGTGGATGCACAGATGGCGGAAGCCCATGCTTTGGAGGGACTGCAACTGGCAA
CCCACATAGGGTGT

>5_1101_7543_2367 join80|90bp
```

```
TACTCTAGTTTATTGTGACAATGTCAATACTTCATACCTCTCGGCTAATTCTGTCACAAGAAAA
GGGGGTTATATGTAACNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNATTTTCTAATAT
AATTCTTTGTTGCTGCCCTATTGCACTAAAGGTATGATGTTGATGCCTTTATATCGAAAAGAAA
TTTTGAAACTGGTA

>5_1101_8272_2434 join80|90bp

CGGCGTGGACAGTGGTGCGGAGGGGCGTCGGCGCTGGCGCCAGGTAAGCGTCGTGATGCTGTGG
CGCTCGTAGCGCCCCGNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTTGCAGGAGTC
GGCGTGGACAGTGGTGCGGAGGGGCGTCGGCGCTGGCGCCAGGTAAGCGTCGTGATGCTGTGGC
GCTCGTAGCGCCCC

>5_1101_9541_2360 join80|90bp

GTCTGCTTGATCCCCGATGCTTACACACAGTGAGGCCGAGATCGGAAGAGCGGTTCAGCAGGAA
TGCCGAGACCGATCTCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCCGAGATCTACA
CTCTTTCCCTACACGACGCTCTTCCGATCTTGAATTGCAGTATCGGTCTGCTTGATCCCCGATG
CTTACACACAGTGA

>5_1101_10886_2255 join80|90bp

AAGAGACGATATGATAGGGAGTTTGGTGAATTTGTGTCGTCCAGGAAGTCTATTTCCACGAAGG
TGCACCCTCATACTTTNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCAAGTGAAAAAA
CATGCGTGCACTTGTCCTTAGAAATCTGACAAAGAGAACAAGGCCAAGAATCACTTGTCATGTG
GGACGAACGGTGGA

>5_1101_12030_2409 join80|90bp

AGACTTAACTACCCTAACCATGGGCAACTAGTTTCAAAGGTTGCCTGCGTGTTTCAAGGTTCGT
GCAGCACCAGCCACTCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCGGTGGATTCAT
```

```
CATCCAAGGTAAGAACCTGCGGTCAATAATTCAAGTCTAGGTCGTCATGGCTCGTGATGAAGTT
GTAGTTGTAGCTAA

>5_1101_12095_2411 join80|90bp

GTACCTCGCGTGGTCTGGTCTCCTCTCTCTTTTTTTTTTCCATAATGAAATGAAATACTCCTCC
TCCTAGTACTGTACTGNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNCGCCACCAATCG
ATCGGCGCCGGAAACATGTTTTCTGCGGTGGTCTAGGGCTGTTTTCCTTCAAAAACCGTCTGTT
TTCGCCAACCCGAC

>5_1101_13433_2380 join80|90bp

GTCGACTGCTGCGAGTACTACGCCGATCCAACTATGATTCAGACACTCCTCCTCCATAAGCTGA
GATCGGGCACAGATAANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTGTTTCTTGTTG
GAAGTTGAGATTTTTTAAGATCGAGTTTGTTGGTATGACTAGGGTGGCGCGATTCAGGATACCG
CTGGATGGTATCCC

>5_1101_17532_2492 join80|90bp

GAGAGATGAAGATTCAATTGCTGCGTTTTGGGGCTTCCATGCATGTTATCTTTTCCCGACTAGT
AGAGTTTGTGGGTGTANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTGTACCTCGAAA
AAATATTGCATCATCATGGTTGCCCACTTCATTCTCTAGCTTCTATCGTGATCAATCTTGCGGA
ACTTTCTAAATCCT

>5_1101_19707_2482 join80|90bp

TTAGCTGGATATGCTTAGCTTACCAGGGGCTGCTTAACTAGCTGGATGTTGTCGGTTCATTAGG
TTGCCTTGTCCGATCGNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNTTGTTAGCTGGA
TATGCTTAGCTTACCAGGTGCTGCTTAACTAGCTGGATGTTGTCGATTCATTAGTTTGCCTTGT
CCGATCGATCGATC
```

The results in testOUT_stack.txt.sts are the statistic data. It looks like:

```
The output of the joined reads will looks like:

PERJ.pl was programmed by Xuewen Wang, for supporting email : xwwang@ymail.com

PERJ.pl was run and results were yielded at Sun Oct 13 00:43:48 2013

options used: -l TGAAT_TrS.1.XW.fq -r TGAAT_TrS.2.XW.fq -trimleft 1 -trimright 2 -format 1 -n 100 -stacks 1 -o testOUT_stack.txt.

Total reads in TGAAT_TrS.1.XW.fq are 10 .

Total reads in TGAAT_TrS.2.XW.fq are 10 .

Total joined reads in testOUT_stack.txt are 10 .
```

iii. shell script for easy run

1) for linux

For easy use, a shell script (.sh) was prepared to run in Linux shell. In the installed folder, there is one file called

linux_run_PERJ.sh

Simply type the following command, you will find two new files **testOUT.txt,**

testOUT.txt.sts generated.

sh **linux_run_PERJ.sh**

After changing the input file names of reads or options value in **linux_run_PERJ.sh,** you can run on your own sequence files data.

2) for windows

For easy use, a shell script (.sh) was prepared to run in Linux shell. In the installed folder, there is one file called

linux_run_PERJ.bat

Simply double click this file or to go the DOS console to run it. You will find two new files **testOUT.txt,**

testOUT.txt.sts generated.

linux_run_PERJ.bat

After changing the input file names of reads or options value in **linux_run_PERJ.bat**

, you can run on your own sequence files data.

iv. command options of PERJ

The following (Table 2) is the meaning of the options for more control on output files. You can also type the following command to see the meaning of each option in the command console.

```
perl PERJ
```

Table 2 meaning of options for software PERJ

Options:	Value
-l	the fastq file name of the left (l)/forward reads. If file name has space, put "" to either side of the file name
-r	the fastq file name of the right (r) /reverse reads
-trimleft	Integer number, the length of nt to be removed in the 5 end of the left/forward reads. e.g. -trimleft 5. Default is 0.
-trimright	Integer number, the length of nt to be removed in the 5 end of the right/reverse reads. e.g. -trimright 5. Default is 0.
-o	value is the file name to store the results of joined reads.
-format	value is 1 or 2. value 1 will produce joined read in fasta format. value 2 will produce joined read in fastq format. default is 1.
-n	Positive integer number, to fill in N for -n value times between left and right reads, e.g. 10 Ns filling, default 0 (no filling).
-stacks	Value is 1 or 0. Value 1 means input sequence files are from previous

	'stacks' output,in 'stacks' output format. Value 0 means the input sequence files are not in stacks output format. Default is 0.

The following is the example of output for option pair: -format 1 for ilumina format reads example files, TTGGATGG_1_48.fq and TTGGATGG_2_48.fq. The setting here is no Ns filling. The joined reads data are in fasta format. Settings are:

```
perl PERJ.pl -l TTGGATGG_1_48.fq -r TTGGATGG_2_48.fq -trimleft 1 -trimright 2  -format 1 -n 0 -stacks 0 -o testOUT.txt
```

The full result data is available in file < testOUT_format1.txt>.

```
>CCRI0219:133:D243CACXX:7:1101:20008:1931 join86|95bp

CACTGCCTTAGGAGGCATCTTTAGAAACTTCGTCACCATTCCTTAGATGACGAAGAGAGACAAGAGTGGAGCGGCAGGGGGAGGCGAAGAGTGGAGCGGCAG
GGGAAGTCGCATCAGCTGGCACAACAACATACGGTGGAGAGCTCGAAATAGGAACACGGGCATCAACACGGACAGGCGA

>CCRI0219:133:D243CACXX:7:1101:1671:2018 join86|95bp

AACGTCAAGAACATTTTGCCGAACCCGATACACCATCTATCCCAATAGGAATTTTAGTCCACAAAACCCTGTAGAGAAGAGAGAGACCATCTATCCCAATAGGAA
TTTTAGTCCACAAAACCCTGTAGAGAAGAGAGAGAACCGCATGAAATTCTAGCTAGAGCATCCCACCATTTTTTGC

>CCRI0219:133:D243CACXX:7:1101:5178:2070 join86|95bp

TGATTCTTTTTTAAGCTACAGGTTAATTGTCGAGTAAAAAAGCTATACCGTGTTTTGTCCTAGCGGTGCGCGGAGTGCGAGCTGGTTTTAAGCTACAGGTTAATT
GTCGAGTAAAAAAGCTATACCGTGTTTTGTCCTAGCGGTGCGCGGAGTGCGAGCTGGTCAAAGGTGAGGCACTCGC

>CCRI0219:133:D243CACXX:7:1101:8477:2127 join86|95bp

GATTGATAAAATTAGTATGTTAATTTGATTCTCTTGTTCCTTGGATACAAGTAACTCTTTTTTTAATGGATACAAGTATCTTTATACTATACCAATTACCAAGTAATA
GTAATGAGGGCCAGACTGAAATCTCCACTGAAGTGCCTAACTAGAGCTAGTAGACAGTTCTAATTAGCATATT

>CCRI0219:133:D243CACXX:7:1101:13266:2024 join86|95bp

GCAAGTTGTATGGTGCGATCTGCATAGTACCTGCTACATGCATTAGGCTGTAGTTGGCCATCTATGCAGTTCTGTACTGTTGTTTCATATATATATACACTTGTGG
TGCTTAGTATTTACTGTTTGGCCATTAGAATGTGTGGTTTATCTAGCATTGGTAACACTTCATATTCTAGTTTTG
```

The following is the example of output for option pair: -format 2 for ilumina format reads example files, TTGGATGG_1_48.fq and TTGGATGG_2_48.fq. The setting here is no Ns filling. The joined reads data are in fastq format. Settings are:

```
perl PERJ.pl -l TTGGATGG_1_48.fq -r TTGGATGG_2_48.fq -trimleft 1 -trimright 2  -format 2 -n 0 -stacks 0 -o testOUT.txt
```

The full data is available in file < testOUT_format2.txt>.

```
@CCRI0219:133:D243CACXX:7:1101:20008:1931 join86|95bp

CACTGCCTTAGGAGGCATCTTTAGAAACTTCGTCACCATTCCTTAGATGACGAAGAGAGACAAGAGTGGAGCGGCAGGGGGAGGCGAAGAGTGGAGCGGCAGGGGAAGTCGCATCAGCTGGCACAAC
AACATACGGTGGAGAGCTCGAAATAGGAACACGGGCATCAACACGGACAGGCGA

+

GGIIJJJIIJJGGJIIIFGIJJFCHIJJJJIFIEGIJJIIIJJIIJIJGIIJHHHEHDFFFFEBC@6>BAACDB=BBBDD######A:CADAB><B>?2+(D@DDDBD@DDDCCAEACFDB?@EEA=FJJIGHGFFCJJGIIFGDBGIHGBG
GIHFGHEGIIGHEGFFF@IHED?FHFDFF

@CCRI0219:133:D243CACXX:7:1101:1671:2018 join86|95bp

AACGTCAAGAACATTTTGCCGAACCCGATACACCATCTATCCCAATAGGAATTTTAGTCCACAAAACCCTGTAGAGAAGAGAGAGACCATCTATCCCAATAGGAATTTTAGTCCACAAAACCCTGTAGAG
AAGAGAGAGAACCGCATGAAATTCTAGCTAGAGCATCCCACCATTTTTTGC

+

HJDFFHGEGIIJJIJJJIGIJIIIIIIIIDIIJ>DE@EGHHHGHFFFFFFEEEEEACCDDCCCDDDD?@BC>>ACCACCC?<??@@CDEDDDDDDDDDDDDCDDEDCDCDDEDCEDFFBBDHGHIJJIHIGG@HGFHE?IIIJI
GGIHIHJGIIIIIIIIGJJIIHJHDCJJIHGHGGFFF

@CCRI0219:133:D243CACXX:7:1101:5178:2070 join86|95bp

TGATTCTTTTTTAAGCTACAGGTTAATTGTCGAGTAAAAAAGCTATACCGTGTTTTGTCCTAGCGGTGCGCGGAGTGCGAGCTGGTTTTAAGCTACAGGTTAATTGTCGAGTAAAAAAGCTATACCGTGTT
TTGTCCTAGCGGTGCGCGGAGTGCGAGCTGGTCAAAGGTGAGGCACTCGC

+

DHHHJEHIJJJIJHGHIJIJJIJIIGGJBHIIIIDFHIIJJJJJIJIJGEHDBFFDA>ACCEDDB=BB@?BD3>8AC@<BDBDD9CDDDDCDDDBDDDCCDDDDDDDCDDDDDDDDDDEEDB@BDDDBDDDDDDDDDDF
FHGIIGGJJIGGGJHIIHGJIGGBHDDEHIJIHDD<CDDBF

...
```

After running, there are two result files generated. They are:

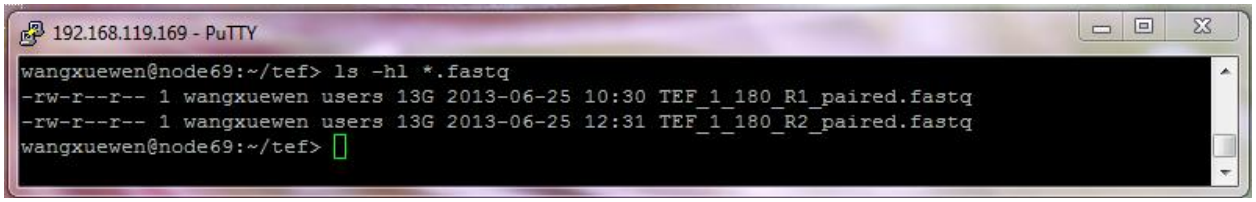
testOUT.txt

testOUT.txt.sts

The file **testOUT.txt** contains the joined reads output results while **testOUT.txt.sts** contains statistical information for all input files and output files , including running options, input file name and total reads, output file name and total joined reads. The content of file **testOUT.txt** will depend on the -format option. -format 1 will produce joined sequence without quality data in fasta format while -format 2 will produced joined fastq output with quality data.

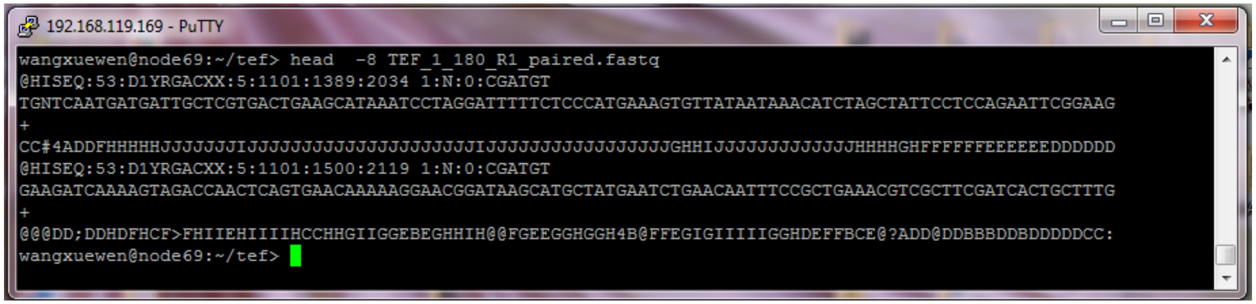
5. A running instance

Here is a real running example for software PERJ. The two next generation sequence files are Illumina hiseq pair-end reads raw data, left reads (Figure 6) and right reads (Figure 7), around 13G bytes in hard disk of each file (Figure 5). There are 53 million reads (Figure 10) in each input file and total running time for joining all reads is around 46.7 min (Figure 11). The joining of pair-end reads is 100% done (Figure 10). The joined reads are saved in two output files, joined reads (Figure 8) and statistical and running log information (Figure 9).



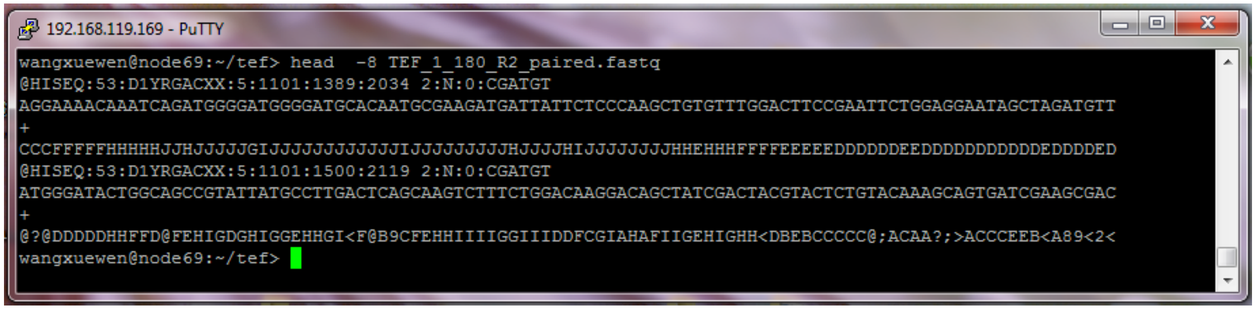
```
wangxuewen@node69:~/tef> ls -hl *.fastq
-rw-r--r-- 1 wangxuewen users 13G 2013-06-25 10:30 TEF_1_180_R1_paired.fastq
-rw-r--r-- 1 wangxuewen users 13G 2013-06-25 12:31 TEF_1_180_R2_paired.fastq
wangxuewen@node69:~/tef> 
```

Figure 5 Two input files containing Illumina hiseq PE100 pair-end reads



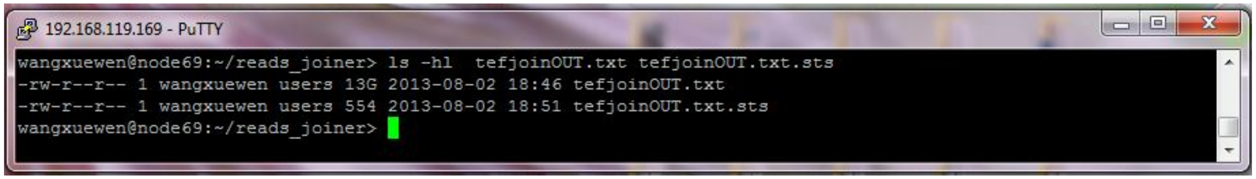
```
wangxuewen@node69:~/tef> head  -8 TEF_1_180_R1_paired.fastq
@HISEQ:53:D1YRGACXX:5:1101:1389:2034 1:N:0:CGATGT
TGNTCAATGATGATTGCTCGTGACTGAAGCATAAATCCTAGGATTTTTCTCCCATGAAAGTGTTATAATAAACATCTAGCTATTCCTCCAGAATTCGGAAG
+
CC#4ADDFHHHHHJJJJJJIJJJJJJJJJJJJJJJJJJJJJIJJJJJJJJJJJJJJJJJJGHHIJJJJJJJJJJJJJHHHHGHFFFFFFEEEEEEEDDDDDD
@HISEQ:53:D1YRGACXX:5:1101:1500:2119 1:N:0:CGATGT
GAAGATCAAAAGTAGACCAACTCAGTGAACAAAAAGGAACGGATAAGCATGCTATGAATCTGAACAATTTCCGCTGAAACGTCGCTTCGATCACTGCTTTG
+
@@@DD;DDHDFHCF>FHIIEHIIIIHCCHHGIIGGEBEGHHIH@@FGEEGGHGGH4B@FFEGIGIIIIIGGHDEFFBCE@?ADD@DDBBBDDBDDDDDCC:
wangxuewen@node69:~/tef> 
```

Figure 6 Left reads showing data format of Illumina hiseq pair-end file



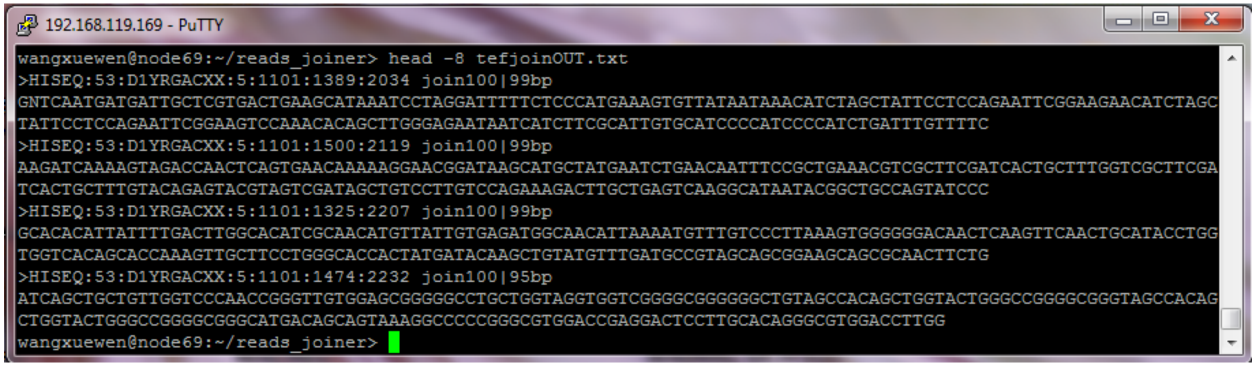
```
wangxuewen@node69:~/tef> head  -8 TEF_1_180_R2_paired.fastq
@HISEQ:53:D1YRGACXX:5:1101:1389:2034 2:N:0:CGATGT
AGGAAAACAAATCAGATGGGGATGGGGATGCACAATGCGAAGATGATTATTCTCCCAAGCTGTGTTTGGACTTCCGAATTCTGGAGGAATAGCTAGATGTT
+
CCCFFFFFHHHHHJJHJJJJJGIJJJJJJJJJJJJIJJJJJJJJJHJJJJHIJJJJJJJJHHEHHHFFFFEEEEEEDDDDDDEEDDDDDDDDDDDDEDDDDED
@HISEQ:53:D1YRGACXX:5:1101:1500:2119 2:N:0:CGATGT
ATGGGATACTGGCAGCCGTATTATGCCTTGACTCAGCAAGTCTTTCTGGACAAGGACAGCTATCGACTACGTACTCTGTACAAAGCAGTGATCGAAGCGAC
+
@?@DDDDDHHFFD@FEHIGDGHIGGEHHGI<F@B9CFEHHIIIIGGIIIDDFCGIAHAFIIGEHIGHH<DBEBCCCCC@;ACAA?;>ACCCEEB<A89<2<
wangxuewen@node69:~/tef> 
```

Figure 7 Right reads showing data format of Illumina hiseq pair-end file



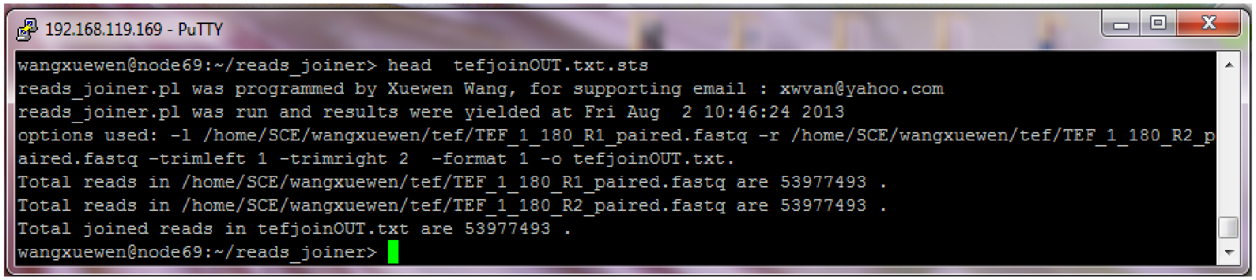
```
wangxuewen@node69:~/reads_joiner> ls -hl  tefjoinOUT.txt tefjoinOUT.txt.sts
-rw-r--r-- 1 wangxuewen users 13G 2013-08-02 18:46 tefjoinOUT.txt
-rw-r--r-- 1 wangxuewen users 554 2013-08-02 18:51 tefjoinOUT.txt.sts
wangxuewen@node69:~/reads_joiner> 
```

Figure 8 Output files generated by the software PERJ



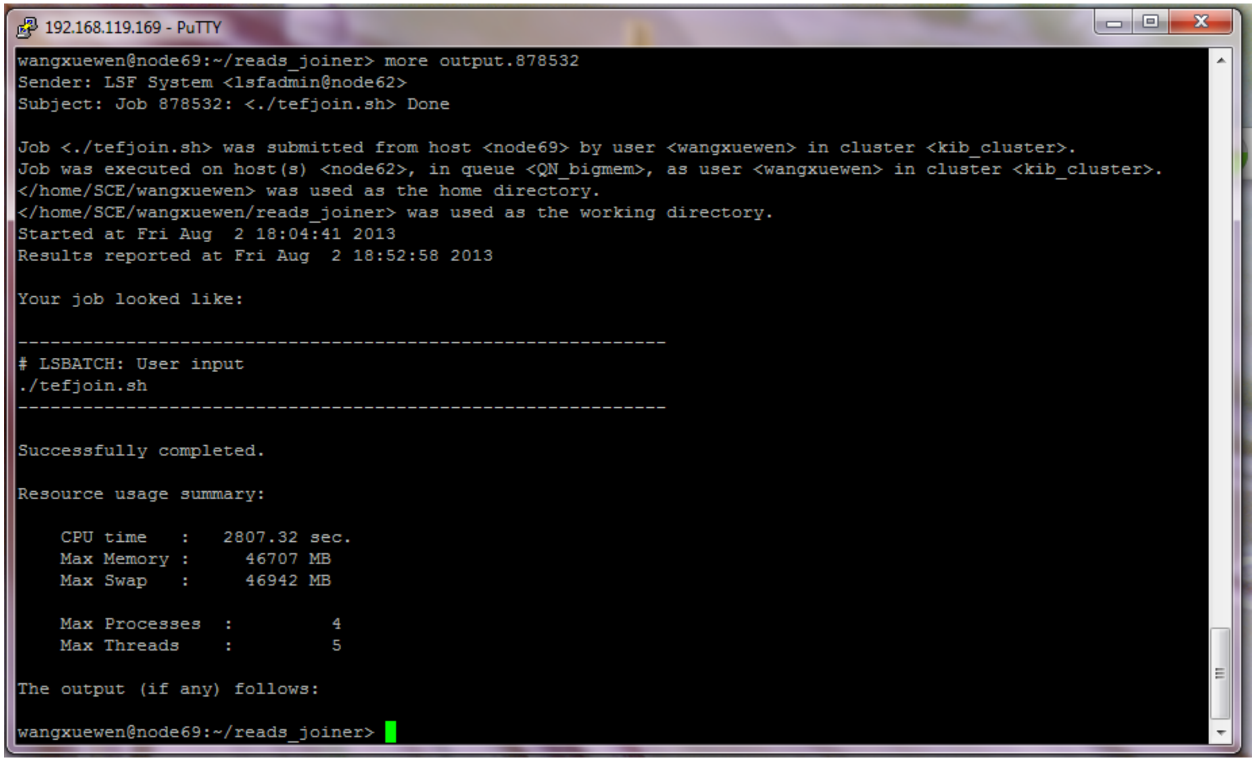
```
wangxuewen@node69:~/reads_joiner> head -8 tefjoinOUT.txt
>HISEQ:53:D1YRGACXX:5:1101:1389:2034 join100|99bp
GNTCAATGATGATTGCTCGTGACTGAAGCATAAATCCTAGGATTTTTCTCCCATGAAAGTGTTATAATAAACATCTAGCTATTCCTCCAGAATTCGGAAGAACATCTAGC
TATTCCTCCAGAATTCGGAAGTCCAAACACAGCTTGGGAGAATAATCATCTTCGCATTGTGCATCCCCATCCCCATCTGATTTGTTTTC
>HISEQ:53:D1YRGACXX:5:1101:1500:2119 join100|99bp
AAGATCAAAAGTAGACCAACTCAGTGAACAAAAAGGAACGGATAAGCATGCTATGAATCTGAACAATTTCCGCTGAAACGTCGCTTCGATCACTGCTTTGGTCGCTTCGA
TCACTGCTTTGTACAGAGTACGTAGTCGATAGCTGTCCTTGTCCAGAAAGACTTGCTGAGTCAAGGCATAATACGGCTGCCAGTATCCC
>HISEQ:53:D1YRGACXX:5:1101:1325:2207 join100|99bp
GCACACATTATTTTGACTTGGCACATCGCAACATGTTATTGTGAGATGGCAACATTAAAATGTTTGTCCCTTAAAGTGGGGGGACAACTCAAGTTCAACTGCATACCTGG
TGGTCACAGCACCAAAGTTGCTTCCTGGGCACCACTATGATACAAGCTGTATGTTTGATGCCGTAGCAGCGGAAGCAGCGCAACTTCTG
>HISEQ:53:D1YRGACXX:5:1101:1474:2232 join100|95bp
ATCAGCTGCTGTTGGTCCCAACCGGGTTGTGGAGCGGGGGCCTGCTGGTAGGTGGTCGGGGCGGGGGGCTGTAGCCACAGCTGGTACTGGGCCGGGGCGGGTAGCCACAG
CTGGTACTGGGCCGGGGCGGGCATGACAGCAGTAAAGGCCCCCGGGCGTGGACCGAGGACTCCTTGCACAGGGCGTGGACCTTGG
wangxuewen@node69:~/reads_joiner> 
```

Figure 9 Joined reads in output file generated by the software PERJ



```
wangxuewen@node69:~/reads_joiner> head  tefjoinOUT.txt.sts
reads_joiner.pl was programmed by Xuewen Wang, for supporting email : xwvan@yahoo.com
reads_joiner.pl was run and results were yielded at Fri Aug  2 10:46:24 2013
options used: -l /home/SCE/wangxuewen/tef/TEF_1_180_R1_paired.fastq -r /home/SCE/wangxuewen/tef/TEF_1_180_R2_p
aired.fastq -trimleft 1 -trimright 2  -format 1 -o tefjoinOUT.txt.
Total reads in /home/SCE/wangxuewen/tef/TEF_1_180_R1_paired.fastq are 53977493 .
Total reads in /home/SCE/wangxuewen/tef/TEF_1_180_R2_paired.fastq are 53977493 .
Total joined reads in tefjoinOUT.txt are 53977493 .
wangxuewen@node69:~/reads_joiner> 
```

Figure 10 Statistical information of output file generated by the software PERJ



```
wangxuewen@node69:~/reads_joiner> more output.878532
Sender: LSF System <lsfadmin@node62>
Subject: Job 878532: <./tefjoin.sh> Done

Job <./tefjoin.sh> was submitted from host <node69> by user <wangxuewen> in cluster <kib_cluster>.
Job was executed on host(s) <node62>, in queue <QN_bigmem>, as user <wangxuewen> in cluster <kib_cluster>.
</home/SCE/wangxuewen> was used as the home directory.
</home/SCE/wangxuewen/reads_joiner> was used as the working directory.
Started at Fri Aug  2 18:04:41 2013
Results reported at Fri Aug  2 18:52:58 2013

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
./tefjoin.sh
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time   :   2807.32 sec.
    Max Memory :     46707 MB
    Max Swap   :     46942 MB

    Max Processes  :         4
    Max Threads    :         5

The output (if any) follows:

wangxuewen@node69:~/reads_joiner>
```

Figure 11 Running time and memory usage for two 13G bytes files

6. Acknowledgement

I wrote the software PERJ which was inspirited from Prof. Katrien Devos. Here I want to thank Prof. Katrien Devos very much for her original idea.

By Xuewen Wang， PhD

Supporting email xwwang@ymail.com

2013-10-12th