**Manual of Fasta2Gnashy PairedV2 Version2/3**

**JINCHUN ZHANG**

**Last updated 2016-07-10**

I am writing a module to analyze the mammalian Calling Cards with the following protocol.

* **THE PROTOCOL DESCRIPITON:**

Forward primer (One of the following):

OM-PB-ACG:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTACGTTTACGCAGACTATCTTTCTAG

OM-PB-CTA:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTCTATTTACGCAGACTATCTTTCTAG

OM-PB-GAT:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTGATTTTACGCAGACTATCTTTCTAG

OM-PB-TGC:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTTGCTTTACGCAGACTATCTTTCTAG

OM-PB-TAG:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTTAGTTTACGCAGACTATCTTTCTAG

OM-PB-ATC:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTATCTTTACGCAGACTATCTTTCTAG

OM-PB-CGT:

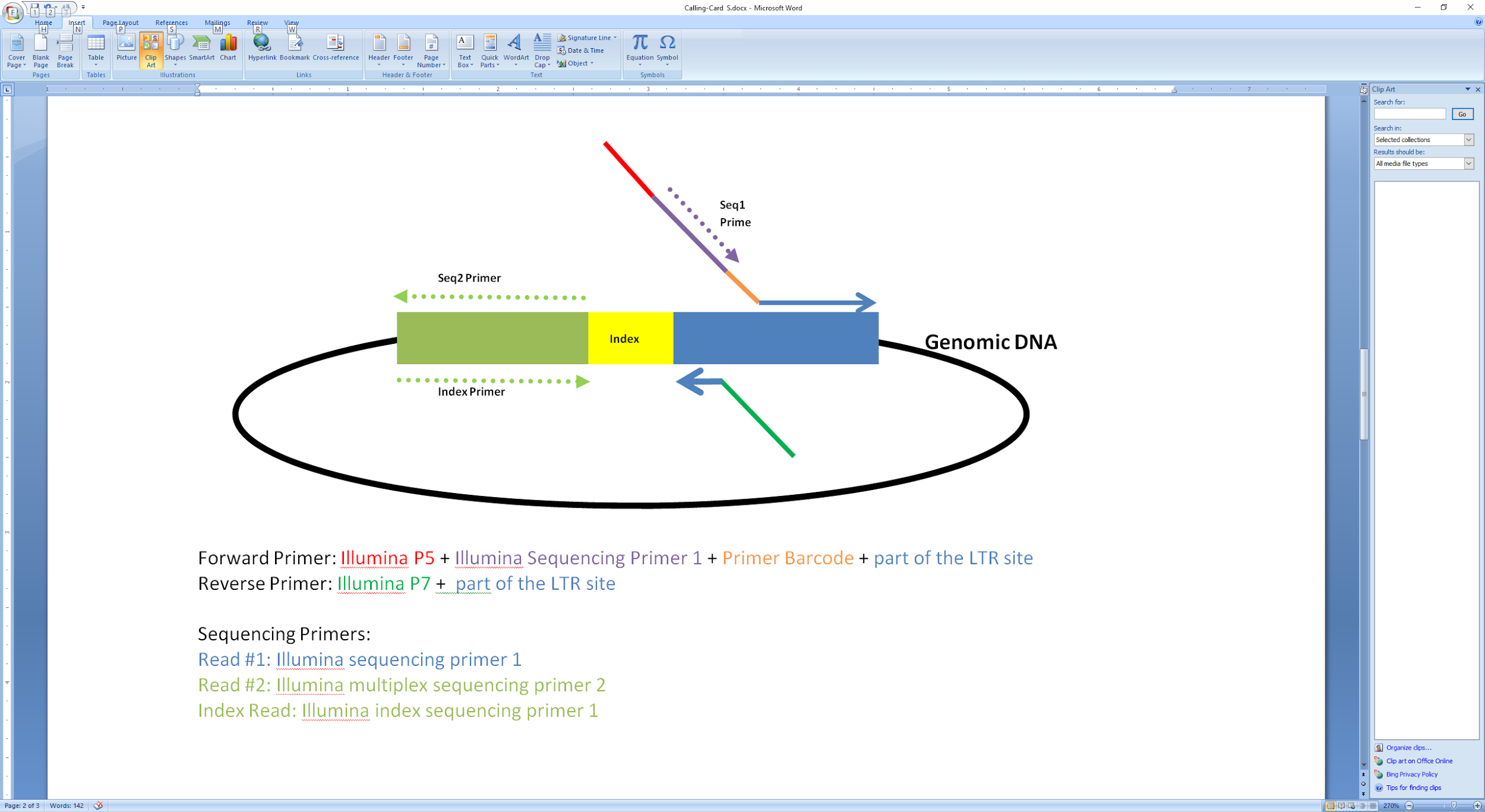
AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTCGTTTTACGCAGACTATCTTTCTAG

OM-PB-GCA:

AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCTGCATTTACGCAGACTATCTTTCTAG

Reverse Primer: OM8951:

CAAGCAGAAGACGGCATACGAGATCTTGTTATAGATAGCTTATCGCGATC



* **SCRIPT 1: Fasta2Gnashy\_PairedV2\_version2.py & Fasta2Gnashy\_PairedV2\_version3.py**

Depends on weather filter the reads by mapping the raw sequences back to plasmid reference genome, there are two versions scripts. Fasta2Gnashy\_PairedV2\_version3.py contains the plasmid mapping step, while the Fasta2Gnashy\_PairedV2\_version2.py dosen’t.

**Location:**

/scratch/rmlab/code/jczhang/Fasta\_to\_Gnashy/PairV2/Fasta2Gnashy\_PairV2\_version3.py

/scratch/rmlab/code/jczhang/Fasta\_to\_Gnashy/PairV2/Fasta2Gnashy\_PairV2\_version2.py

**The module required for the script:**

bowtie2, biopython, numpy, pandas, samtools, bedtools, os

**The overall workflow takes four files as input:**

1. A index file that matches the transposon barcode and primer barcode to the experiments. The format is: <experiment\_name>,<primer\_barcode>,<transposon\_barcode1>,<transposon\_barcode2>,...
2. A .fastq file with the reads from the 1st paired end.
3. A .fastq file with the reads from the 2nd paired end.
4. A .fastq file with the reads from the index read.

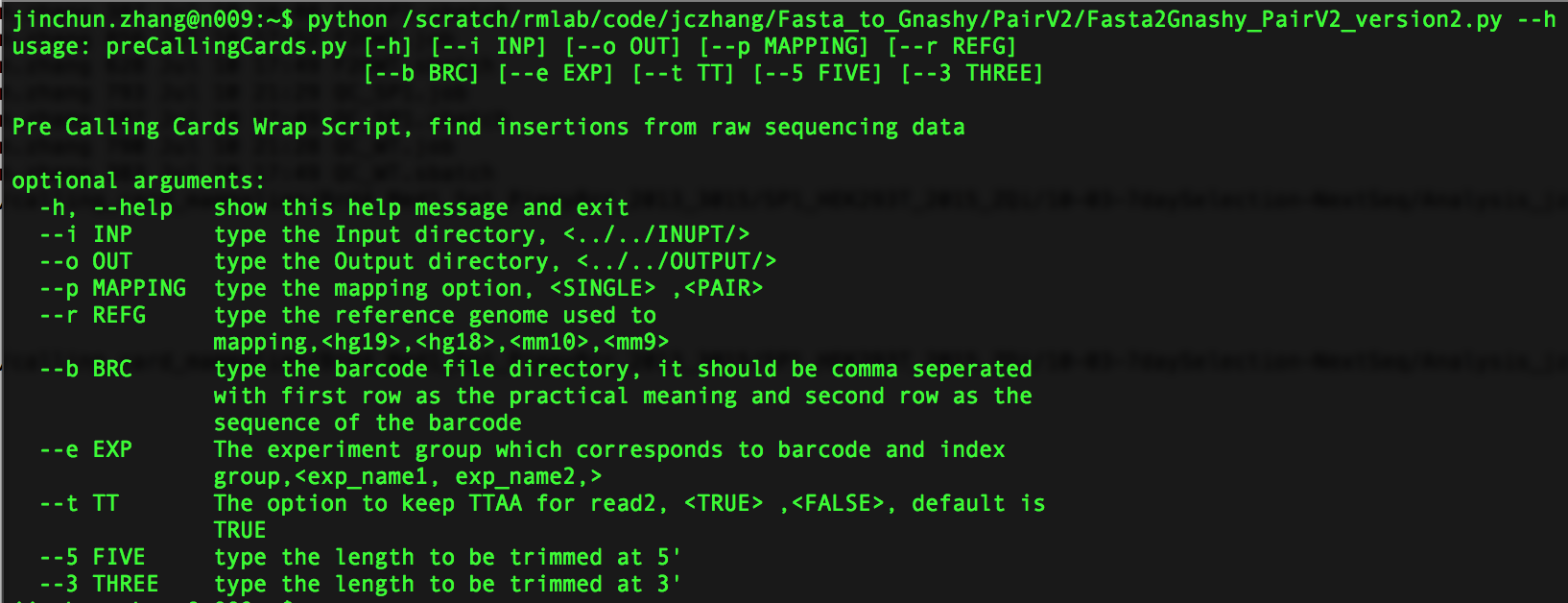
Note: All read should be named in this fashion:

EXP\_<experimentname>\_BRD\_<barcodeseq>\_IDX\_<indexsequence>\_Read1.fastq

EXP\_<experimentname>\_BRD\_<barcodeseq>\_IDX\_<indexsequence>\_Read2.fastq

EXP\_<experimentname>\_BRD\_<barcodeseq>\_IDX\_<indexsequence>\_Index.fastq

The module can be loaded as a module or called from the command line. If called from the command line, the usage is:



**The sample command line is:**

python /scratch/rmlab/code/jczhang/Fasta\_to\_Gnashy/PairV2/Fasta2Gnashy\_PairV2\_version2.py --i /scratch/rmlab/calling\_card\_mammalian/Brd4\_Med1\_Sp1\_PiggyBrc\_2013\_3015/SP1\_HEK293T\_2015\_ZQi/10-03-7daySelection-NextSeq/name-changed/ --o /scratch/rmlab/calling\_card\_mammalian/Brd4\_Med1\_Sp1\_PiggyBrc\_2013\_3015/SP1\_HEK293T\_2015\_ZQi/10-03-7daySelection-NextSeq/Analysis\_jzhang/Analysis\_Output/ --b /scratch/rmlab/calling\_card\_mammalian/Brd4\_Med1\_Sp1\_PiggyBrc\_2013\_3015/SP1\_HEK293T\_2015\_ZQi/10-03-7daySelection-NextSeq/bfile --r hg19 --p SINGLE --e SP1 --t TRUE --5 0 --3 0'

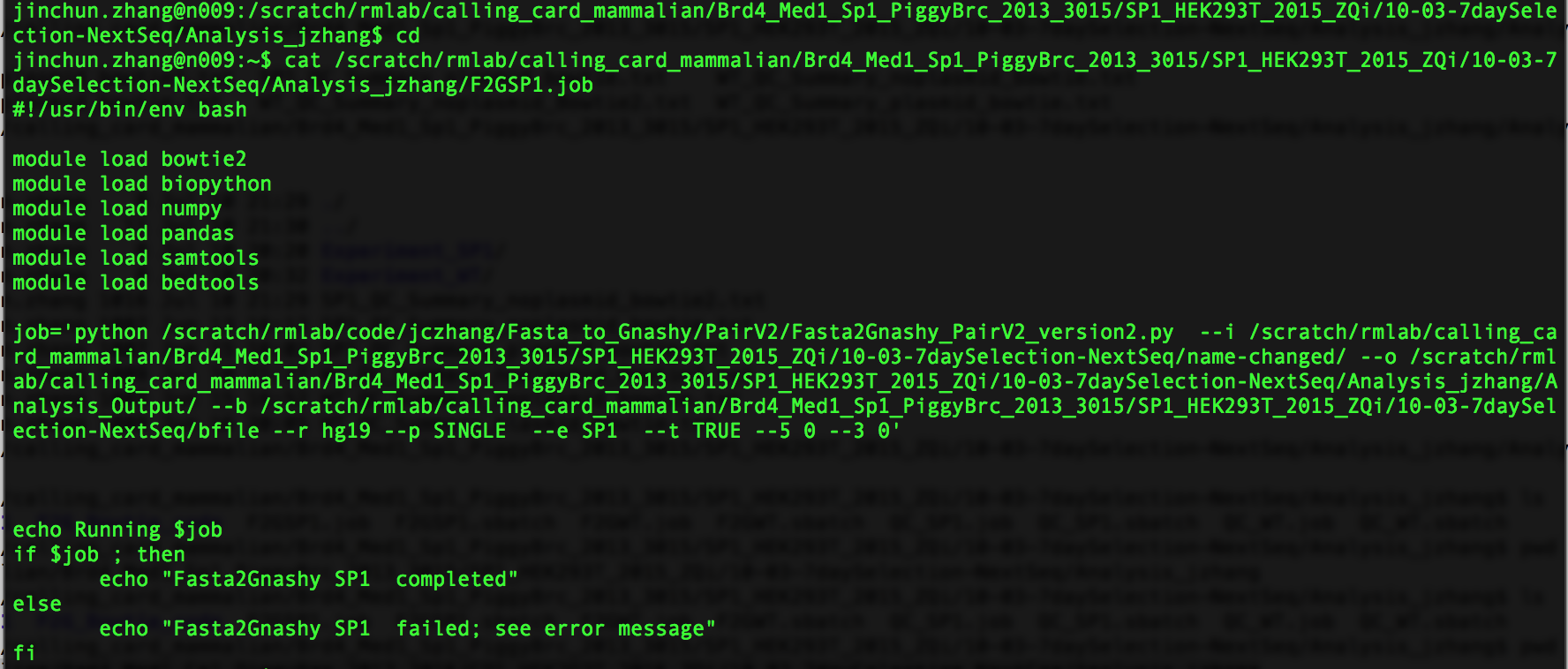
**The program works as follows:**

1. Open barcode file.
2. Read in barcodes and experiment names into hash
3. Loop through each experiment,
4. Loop through each index within that experiment, analyze reads in <read1\_filename>, <read2\_filename>,<indexread\_filename> separately.
   1. For read1, check if the read has a Primer Barcode and TTAA following the Primer Barcode. For read2, check if the read has cutter. For index read, check if it matches the index passed. If so, save the read’s ID to a temporary list for each file.
   2. Print out reads that whose ID are in all three lists above.
   3. Align the reads for each index read with bowtie to plasmid\_ref seq.
   4. For the reads not mapped to plasmid reference sequence, align the reads for each index read with bowtie.
5. Compile the bowtie output files into gnashy format.

**The output diagram of the program:**

Note for running scripts on HTCF cluster

Since the scripts need to be submitted using sbatch, we need a .sbatch and a .job script to do so.

Here is a example for .job script:

Here is a example for .sbatch script,

In order to submit the sbatch, use the command ‘sbatch Brd4-HyPB.sbatch’

In order to check the scripts, use the command ‘squeue -u <usr\_name>’

* **SCRIPT 2, Quality Control Summary.**

The scripts from Fasta2Gnashy\_PairedV2\_version3/2.py analyze the input sequencing data and output the gnashy file through output and mapping two major steps. And there is a summary .txt file for each step.

Under the Filter\_Output/ folder for each index, the summary file named, ‘Filter\_Summary.csv’. And under the Map\_Out/ folder for each index, the summary file named, ‘stderr.txt’ which is actually the standard err, ouptut from bowtie.

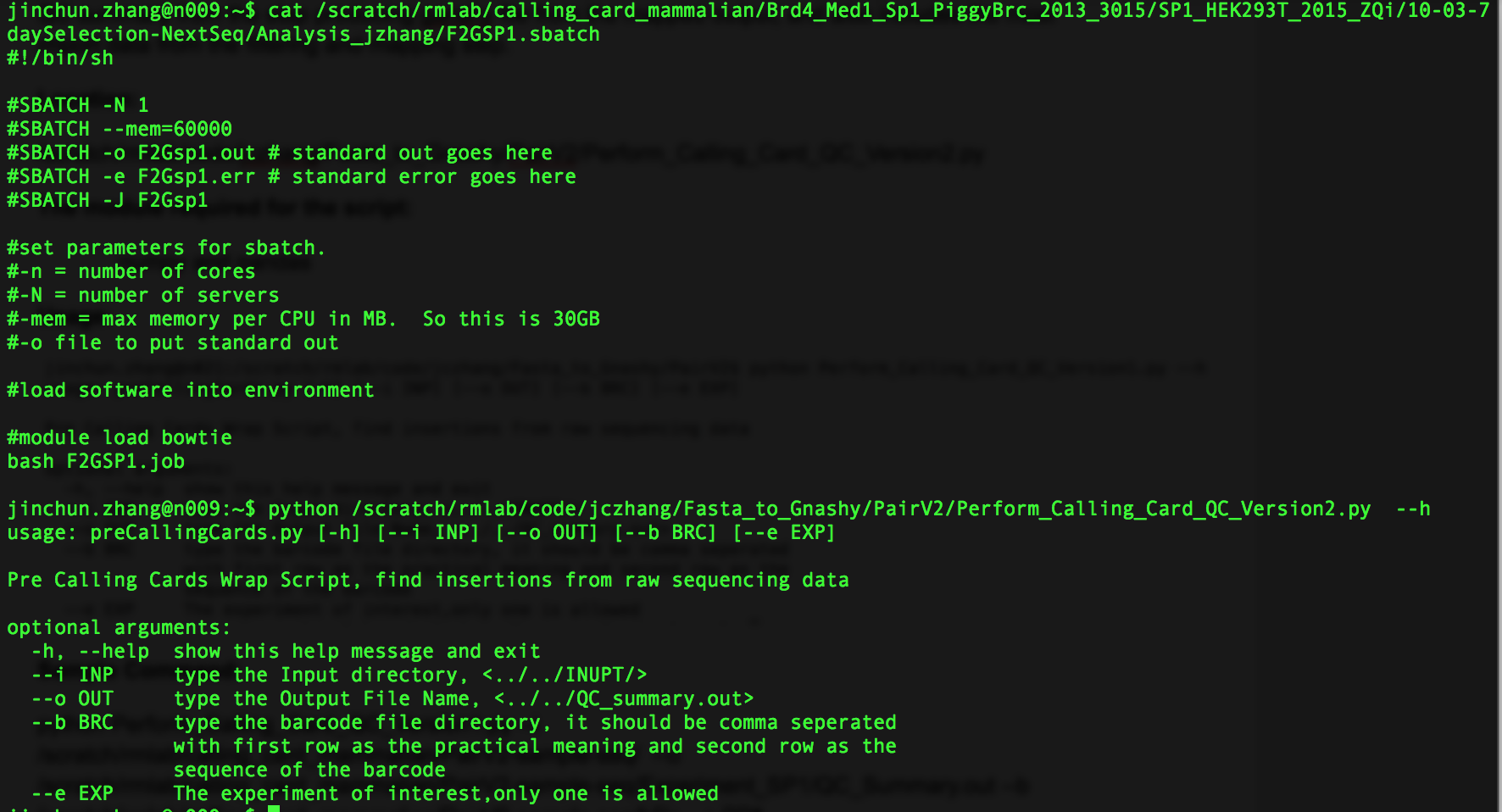
Furthermore, in order to have **a summary of the whole analysis**, I wrote another QC scripts. It produces the QC summary from the step producing ganshy file from the mapped output, and also includes the most import data from the filtering and mapping step.

**Location:**

/scratch/rmlab/code/jczhang/Fasta\_to\_Gnashy/PairV2/Perform\_Calling\_Card\_QC\_Version2.py

**The module required for the script:**

numpy and pandas

**Usage**:

**Sample Command:**

python Perform\_Calling\_Card\_QC\_Version2.py --i /scratch/rmlab/calling\_card\_mammalian/PairV2-sample-seq/ --o /scratch/rmlab/calling\_card\_mammalian/PairV2-sample-seq/Experiment\_SP1/QC\_Summary.out --b /scratch/rmlab/calling\_card\_mammalian/PairV2-sample-seq/bfile --e SP1

**Sample Output:**