The code associated with this package are intended free-of-charge for non-profit usages.

Please contact the authors for commercial usages, modifications and re-distributions.

Please contact the author Xiaotu Ma at Xiaotu.Ma@stjude.org for questions, bugs.

==========

This documentation has 5 sections. (A), Downloading and Testing CleanDeepSeq; (B), Running CleanDeepSeq on your own data; (C), Using your own parameter for allele counting; (D), Using your own parameter for visualization; (E), Running deepSNV

A. Downloading and Testing CleanDeepSeq

A1. Download CleanDeepSeq package at https://www.stjuderesearch.org/site/lab/zhang/cleandeepseq

A2. Unzip the package, and go to the folder: cd ./CleanDeepSeq

A3. Make sure you have samtools working by running below command.

$ samtools view bamdir/Colo829Normal_S5_L001.bam 1:154002369-154002469 | head -n10

You should see reads print on your screen. Consult your system administrator if you have difficulty in this.

A4. Make sure you have R installed by running below command:

$ Rscript code/add_mappability_score.r

You should see below information.

[1] "Usage: Rscript ~ <geno.ifn> <mappability.ifn> <ofn>"

Also make sure you have R package “gtools” installed.

(https://cran.r-project.org/web/packages/gtools/index.html).

$ Rscript code/do_plot_1flank_slim.R

You should see below information:

[1] "Usage: Rscript ~ <marker.ifn> <ifn> <minCvg | 100000; 10000>"

A5. Make sure you have a working cat by running below command:

$ cat /dev/urandom | tr -dc 'a-z0-9' | head -c 30

You should see a random string of length 30.

A6. With steps A3,A4,A5 passed, you can now run below command:

$ sh go.sh

You should see the progress being print out:

Step 0

Step 1

…

A7. After a few hours, you should see below two files:

cpg.mapp.geno_Q30.counts.Colo829Normal_S5_L001.bam.txt
cpg.mapp.geno_Q30.counts.Colo829_TtoN_1to1000_Rep1_S1_L001.bam.txt

shown below are a few rows from file (opened using Microsoft excel)
"*cpg.mapp.geno_Q30.counts.Colo829_TtoN_1to1000_Rep1_S1_L001.bam.txt*"

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Chr	Pos	A	C	G	T	N	geno	Apcnt	Cpcnt	Gpcnt	Tpcnt	N5	N4	N3	N2	N1	P1	P2	P3	P4	P5	Mappability	CpGisland
2	chr1	150208240	1	188	0	0	187	C	0.535	99.465	0	0						G	C	A	A	A	1	-1
3	chr1	150208241	7	0	26346	2	26355	G	0.027	0	99.966	0.008					C	C	A	A	A	T	1	-1
4	chr1	150208242	1	26648	1	1	26651	C	0.004	99.989	0.004	0.004				C	G	A	A	A	T	A	1	-1
5	chr1	150208243	26700	55	2	0	26757	A	99.787	0.206	0.007	0			C	G	C	A	A	T	A	T	1	-1
6	chr1	150208244	26861	0	4	3	26868	A	99.974	0	0.015	0.011		C	G	C	A	A	T	A	T	A	1	-1
7	chr1	150208245	26875	0	1	1	26877	A	99.993	0	0.004	0.004	C	G	C	A	A	T	A	T	A	A	1	-1
8	chr1	150208246	4	1	3	27033	27041	T	0.015	0.004	0.011	99.97	G	C	A	A	A	A	T	A	A	A	1	-1
9	chr1	150208247	27152	0	1	1	27154	A	99.993	0	0.004	0.004	C	A	A	A	T	T	A	A	A	C	1	-1
10	chr1	150208248	3	1	2	27124	27130	T	0.011	0.004	0.007	99.978	A	A	A	T	A	A	A	A	C	G	1	-1
11	chr1	150208249	27138	0	5	0	27143	A	99.982	0	0.018	0	A	A	T	A	T	A	A	C	G	C	1	-1
12	chr1	150208250	27135	2	4	3	27144	A	99.967	0.007	0.015	0.011	A	T	A	T	A	A	C	G	C	C	1	-1
13	chr1	150208251	27191	0	2	0	27193	A	99.993	0	0.007	0	T	A	T	A	A	C	G	C	C	C	1	-1
14	chr1	150208252	0	27156	0	4	27160	C	0	99.985	0	0.015	A	T	A	A	A	G	C	C	C	A	1	-1
15	chr1	150208253	2	1	27165	1	27169	G	0.007	0.004	99.985	0.004	T	A	A	A	C	C	C	C	A	C	1	-1
16	chr1	150208254	8	26925	2	3	26938	C	0.03	99.952	0.007	0.011	A	A	A	C	G	C	C	A	C	T	1	-1
17	chr1	150208255	1	27030	1	1	27033	C	0.004	99.989	0.004	0.004	A	A	C	G	C	C	A	C	T	A	1	-1
18	chr1	150208256	1	27041	0	5	27047	C	0.004	99.978	0	0.018	A	C	G	C	C	A	C	T	A	C	1	-1

Columns with header "A", "C","G","T" are the allele counts; "N" means coverage. "geno" means the genotype, called using 5% as cutoff. Also listed are percentage of A (Apcnt), C (Cpcnt), G (Gpcnt), T (Tpcnt); flanking bases from upstream (N5 through N1) and downstream (P1 through P5). Mappability (1: uniquely mapped) and CpGisland (1: in CpGisland; -1: outside CpGisland) information are also listed.

A8. Different statistical analysis (such as deepSNV analysis) can start from these output files.

A9. You should also see 4 pdf files that look like **Fig. 2** and **Fig. 3** in the manuscript. These are the visualization of error rates, broken down to 1 flanking bases and 3 flanking bases, respectively.

B. Run CleanDeepSeq on your own data

Please run below command:

$ more go.sh

You should see below:

sh ./code/countbam_slim.sh ./bamdir/Colo829Normal_S5_L001.bam 100000

sh ./code/countbam_slim.sh ./bamdir/Colo829_TtoN_1to1000_Rep1_S1_L001.bam 500000

B.1 So the actual wrapper of the code is ./code/countbam_slim.sh; and it is operating on bam files ./bamdir/Colo829Normal_S5_L001.bam and ./bamdir/Colo829_TtoN_1to1000_Rep1_S1_L001.bam

Note that these bam files have been indexed (i.e., having .bai files; using command samtools index). **You need to make sure your own bam files are also indexed.**

B.2 The parameters 100000 (for COLO829BL, normal cell line) and 500000 (for 1:1000 dilution) are visualization coverage cutoff (see main text "**Deep sequencing data analysis**"). Also see A9 above.

B.3 So you can replace the bam file path to your own bam files with above command. You probably want to adjust the visualization coverage cutoff based on your own depth.

C. Using your own parameter for allele counting

Please run below command:

$ head -n29 code/countbam_slim.sh | tail -n10

You should see below output:

QCUTS=30

mCVG=30

readQcut=20

readFcut=0.05

TRIMLEN=5

C1. Here QCUTS means the Phred score cutoff to count alleles. We used 30 here. For HiSeq2500, we suggest using 38, although in our paper we used 30 for simplicity. At cutoff 38, HiSeq2500 had a similar performance as NovaSeq.

C2. mCVG means the minimum coverage that a base pair to be printed to the output file. If you set this parameter to 1, then all covered bases will be output, so that the output file could be huge (such as whole genome data).

C3. readQcut. This is the parameter used to define "poor quality bases", used in **Supplementary Fig. 2d,** orange curve**.**

C4. readFcut. This is the parameter used to define "LQRead" based on "number of poor quality bases" in C3. It is used in **Supplementary Fig. 2d,** orange curve.

C5. TRIMLEN. This is the parameter to do read end trimming. Related to **Fig. 1b**.

You can adjust these parameter to do allele counting.

D. Using your own parameter for visualization

Please run below command:

$ tail -n18 code/countbam_slim.sh | head -7

You should see below output:

GenoCvgCut=50

GenoMinMAF=0.05

GenoMinMut=5

Flank=5

oCut=100

Qcut=30

D1. GenoCvgCut means we will do genotyping on bases with >50 coverage.

D2. GenoMinMAF=0.05 means we will make calls whenever an allele has frequncy >0.05, without a testing.

D3. GenoMinMut=5 means there must be at least 5 mutant alleles to make a call.

D4 Flank=5 means we will provide the upstream 5 bases and downstream 5 bases to the output file. Related to the **Fig. 3** that used flanking bases.

D5. oCut=100 means we only want to output the bases with coverage >100.

D6. Qcut=30 means we have been using Phred score 30 as cutoff.

E. Running deepSNV

You will need to install R package "deepSNV" from https://bioconductor.org/packages/release/bioc/html/deepSNV.html

We have successfully installed the package using R 3.5.0 under Windows 7 Enterprise.

Please run below command:

$ cd ./proc_deepSNV

$ Rscript go_deepSNV.R

In a few minutes you should be able to see below file:

pdf.test_1v1000.txt.pdf

This will be the testing result of our example data.

The actual program that does testing is below file

test_with_deepSNV.R in folder ./proc_deepSNV