Author: Shawn Yost <seyost@ucsd.edu>

Contents

Preface

This document is meant to serve as a guide for the use of multiple Perl scripts to filter false positive variants caused by FFPE DNA damage on next-generation sequencing data. It includes explanations of all command-line options for each command and an idea of basic usage. Input and output file formats are also detailed.

Installation and Prerequisites

The programs are written in Perl and require no installation; they only required that the Perl programming language (http://www.perl.org/get.html), SAMTools (http://samtools.sourceforge.net/), and R (http://www.r-project.org/) are already installed on the machine. To test and see if the programs work properly run "./test_scripts.sh".

Work flow

The programs are designed to work in the following order; however, each script is made to run independently. For example, you can use the calculate_global_nucleotide_mismatch_rate.pl (step 1) program combined with filter_on_globalNucMismatchRate.pl (step 5) to remove variants based on a samples global nucleotide mismatch rate and skip the other three steps.

1. calculate_global_nucleotide_mismatch_rate.pl
    a. This script calculates the global nucleotide mismatch rates across the genome for your sample. See page 4 for details.
2. filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl
    a. This script filters candidate variants based on low read diversity and calculates the local mismatch rate for variants passing the low read diversity filter.
    b. Calculate the local mismatch rate for a set of 'gold standard' variants by specifying to run the 'calculate local mismatch rate' step only. The user can specify to run only one of the two methods or both methods at once (default is to run both methods). These two programs are implemented together because they require the same reads to be extracted from the BAM file.
3. filter_on_local_mismatch_Rates.pl
    a. Next you can filter candidate variants based on their Q score. The Q score is calculated by subtracting the variant's alternate allele frequency by the local mismatch rate calculated in step 2. The program uses the Q score of the 'gold standard' variants to determine threshold used to filter the candidate variants.
4. filter_on_globalNucMismatchRate.pl
    a. This script uses the global nucleotide mismatch rate calculated in step 1 to filter a set of candidate variants.

Basic Usage

Common Options:

-h {} = Print help message.

-v {FILE} = A list of variant sites (FILE TYPE 1).

-o {FILE} = Output file.

-chr {INT} = 0-based coordinate column with the chromosome information [0]

-pos {INT} = 0-based coordinate column with the position information [1]

-ref {INT} = 0-based coordinate column with the reference information [2]

-cons {INT} = 0-based coordinate column with the consensus base information[3]

-cov {INT} = 0-based coordinate column with the coverage information [7]

-seq {INT} = 0-based coordinate column with the read base information [8]

1. calculate_global_nucleotide_mismatch_rate.pl

This program calculates the global nucleotide mismatch rates for a given sample. The global nucleotide mismatch rate profile for sequencing data is calculated across all 6 nucleotide substitution types; A·T>C·G, A·T>G·C, A·T>T·A, C·G>A·T, C·G>G·C, and C·G>T·A. This script first determines a set of high confidence homozygous reference sites derived from a random set of reference loci across the genome. These reference sites are chosen by first removing all potential variant positions in the sample (-v). Next, all sites that are variant in dbSNP132 (-d) and/or the 1000 genomes (-g) project are removed. From the remaining reference loci the script randomly selected 4 sets of '-n' A, T, C, and G sites that have at least -min coverage and at most -max coverage in the sample, making a total of 4n random loci selected per sample. The expected global nucleotide mismatch rate for each substitution type $i \rightarrow j$, $\hat{p}_{(i,j)}$, was then calculated by summing the number of mismatches for a given substitution type and dividing it by the total coverage at the reference site. For example, for the substitution type A·T > C·G, we summed up the number of times we saw an A>C or T>G substitution and then divided by the total coverage obtained by summing over all 200,000 reference A and T sites.

There are 3 required files as inputs and one required option. The first file is a tab-delimited file with a list of sites that are non-reference (-v). The second file is the pileup file (-p). This file is used to determine the number of matching and mismatching bases at a given loci. The third file is the output file (-o) that the global nucleotide mismatch rates will be printed to. The last required option is the percent of the genome covered (-

genCov) option that specifies the percent of the genome that has >= minimum (-min) coverage and <= maximum (-max) coverage. It is highly recommended that you input a dbSNP file and/or the 1000 Genomes project SNPs in order to make the global nucleotide mismatch rates as accurate as possible.

Options:

-h {} = print help message

-v {FILE} = See above, Figure 1 (REQUIRED)

-d {FILE} = dbSNP file in the UCSC genome browser format, Figure 3 (RECOMMENDED)

-g {FILE} = 1000 genomes file, see manual for format, Figure 4 (RECOMMENDED)

-p {FILE} = File in a samtools pileup format, Figure 1 (REQUIRED)

-o {FILE} = Output file for the global nucleotide substitution rates (REQUIRED)

-s {FILE} = A file containing all chromosomes to examine and the total number of bases for that chromosome, see manual for details, Figure 2 (RECOMMENDED) [hg18 chromosomes]

-n {INT} = The number of random As, Ts, Cs, and Gs to examine (OPTIOAL) [100000]

-min {INT} = Minimum coverage for bases used to calculate the global nucleotide substitution rate (OPTIONAL) [3]

-max {INT} = Maximum coverage for bases used to calculate the global nucleotide substitution rate (OPTIONAL) [100]

-genCov {INT} = The percent of the genome with >= ‘min’ and <= ‘max; ranges from 1-100 (REQUIRED)

-chr {INT} = see above (OPTIONAL) [0]

-pos {INT} = see above (OPTIONAL) [1]

-ref {INT} = see above (OPTIONAL) [2]

-cov {INT} = see above (OPTIONAL) [7]

-seq {INT} = see above (OPTIONAL) [8]

2. filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl

This program implements two different methods. The first method removes variants with biased read diversity: Duplicate sequencing reads carrying an error can result in false positive calls. The program removes candidate variants supported by reads with less than X (-minstart) different start positions. The second method of this program calculates the local mismatch rate for a given variant. The local mismatch rate (LMR) = m/(n+m), where (m) is the number of positions matching the reference and (n) the number of mismatched (excluding the candidate variant itself). The LMR is calculated for a given window (-window) around the variant. The user can specify to run only one of the two methods or both methods at once (default is to run both methods). This program requires 4 files; 1) a candidate variant file, 2) a sorted and indexed BAM file that has been processed through SAMTools calmd with the -e option specified, 3) a temporary file that will be deleted with the program finishes, and 4) an output file prefix.

Options:

-h {} = Print out help message

-v {FILE} = Variant file, Figure 1 (REQUIRED)

-b {FILE} = Sorted-indexed BAM file that has been processed with SAMTools calmd -e, Figure 7 (REQUIRED)

-t {FILE} = Temporary file (REQUIRED)

-o {FILE_PREFIX} = Output file prefix (REQUIRED)

-chr {INT} = see above [0]

-pos {INT} = see above [1]

-ref {INT} = see above [2]

-cons {INT} = see above [3]

-window {INT} = +/- Window size used to determine the misMatch rates of a variant [10]

-minstart {INT} = Minimum number of reads with different starting alignment locations needed in order to accept a variant [3]

-rso {} = Specifies to ONLY remove same-start-site variants (do NOT calculate misMatch rate)

-cmro {} = Specifies to ONLY calculate the misMatch rate for all variants (do NOT filter variants)

3. filter_on_local_mismatch_Rates.pl

This program filters variants based on their Q score (Alternate allele frequency - local mismatch rate). The program uses a set of gold standard SNPs' Q scores to determine the minimum Q score for the inputted candidate variants. The minimum Q score is calculated by taking the lower quartile (-q) of the gold standard variants. The higher the quartile the higher the minimum Q score. The program requires 3 files, 1) a variant file with local mismatch rates calculated by filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl, 2) a gold standard variant file with local-mismatch rates calculated by filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl, and 3) an output file for unfiltered variants.

Options:

-h {} = Print this help message

-gold {FILE} = Gold standard set of variants. The file should contain the variants' local-mismatch rates calculated by filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl, Figure 6 (REQUIRED)

-sample {FILE} = Variant file to filter. The file should contain the variants' local-mismatch rates calculated by filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl, Figure 6 (REQUIRED)

-o {FILE} = Output file (REQUIRED)

-mmr {INT} = 0-based column in the variant file (gold/sample) that stores the local mismatch rate calculated by filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl [12]

-seq {INT} = see above [8]

-ref {INT} = see above [2]

-cons {INT} = see above [3]

-q {DOUBLE} = quartile to use on the gold standard variants to calculate the minimum Q score used to filter 'sample' variants [0.10]

4. filter_on_globalNucMismatchRate.pl

This program calculates a false discovery rate according to the Benjemin-Hotchberg method and filters variants below a specified alpha (-a). There are three required input files and one required option. The variant file is in a SAMTools pileup consensus format. The second file is the global nucleotide mismatch rates as calculated by 'calculate_golobal_nucleotide_mismatch_rate.pl'. The third file is a temp file that will be deleted when the program finishes. The last required option is the total possible locations in the genome in which a variant can be called (-tot). This option specifies the *m* variable in the Benjemin-Hotchberg method (http://en.wikipedia.org/wiki/False_discovery_rate).

Options:

-h {} = Print out help message

-gnmr {FILE} = Global nucleotide mismatch rate file, Figure 5 (REQUIRED)

-v {FILE} = Variant file, Figure 1 (REQUIRED)

-o {FILE} = Output file (REQUIRED)

-t {FILE} = tmp file (REQUIRED)

-chr {INT} = see above [0]

-pos {INT} = see above [1]

-ref {INT} = see above [2]

-cons {INT} = see above [3]

-seq {INT} = see above [8]

-a {DOUBLE} = alpha to use in the Benjemin-Hotchberg method, also refers to the FDR [0.05]

-tot {INT} = total possible locations in the genome in which you could possibly call a somatic variant (REQUIRED)

-p {PATH} = The path to the "*calc_binom_pval.r*" file. This file is in the same folder as the filter_on_globalNucMismatchRate.pl program

Extra files

calc_binom_pval.r - calculates the P-value using a Binomial distribution in R.

./test_scripts.sh - tests the Perl programs to see if all prerequisite software is installed.

test_files/ potential_somatic_variants_chr1_1_1000000.final - a set of potential somatic variants (File type 1)

test_files/test_pileup_file.raw - pre-filtered SAMTools pileup consensus file (File type 1)

test_files/ test_genome_size.txt - a file containing the test genome's chromosomes and their size in a BED format (File type 2)

test_files/ test_dbsnp.txt - dbSNP variants from chr1:1-1,000,000 in hg18 coordinates (File type 3)

test_files/ test_1000Genomes.txt - 1000 Genomes project variants from chr1:1-1,000,000 in hg18 coordinates (File type 4)

test_files/ test_gold_set_variants.final - a set of 'gold standard' variants (File type 6)

test_files/ test_bam.bam - a sorted and index BAM, note that this is not File type 7 until it has been processed by samtools calmd -e

test_files/test_chr1_fasta.fa - a fasta file containing the reference genome for hg18 from chr1:1-1,000,000

test_files/test_final_output.txt - what the final output of the test_scripts.sh script should produce

File Formats

Note: All scripts assume that files are tab delimited.

There are 7 different types of files used in the four different programs.

- **FILE TYPE 1**: (v) - Variant file – Figure 1. Required information 1) Chromosome, 2) Position, 3) Reference base 4) Consensus base, 5) Coverage, and 6) Read bases (see SAMTools pileup for more information http://samtools.sourceforge.net/cns0.shtml)
- **FILE TYPE 2:** (s) - Genome Size file – Figure 2. Required information 1) Chromosome, 2) Start position, and 3) End position.
- **FILE TYPE 3:** (d) - dbSNP file – Figure 3. This file is in the format described/downloaded from the UCSC genome browser (http://genome.ucsc.edu/cgi-bin/hgTables).
- **FILE TYPE 4:** (g) - 1000 genomes file – Figure 4. Required information 1) Chromosome and 2) position.
- **FILE TYPE 5:** (gnmr) - Global Nucleotide Mismatch Rates – Figure 5. Required 1) Substitution type and 2) global nucleotide mismatch rate. The substitution types are in the format of ATGC which translates to A>G or T>C.
- **FILE TYPE 6:** (sample/gold) - Variant + Local Mismatch Rate (LMR) file – Figure 6. This file is the same as FILE TYPE 1 except with 3 extra fields. 1) Number of mismatching bases, 2) number of matching bases, and 3) local mismatch rate (LMR).
- **FILE TYPE 7:** (b) - SAM file – Figure 7. This is a normal SAM/BAM format except that it has been processed with SAMTools calmd with the option -e specified (see **Example commands** for details).

#Chr	Pos	Ref	Cons_base	Cons_Qual	SNP_Qual	RMS	Coverage	Read_base	BQ
chr1	856086	C	M	54	54	231	10	.,...a,,AA	Q^Y;$```
chr1	986078	C	S	62	62	246	12	G,,..,...,gg	L``QQ``\^1`]
chr1	1131445	A	W	51	51	186	10	.,tt.,...t	`8````Y`8A
chr1	1223101	G	R	80	80	247	11	,aA,,,a.,,.	``````WO\`S
chr1	1819820	A	M	42	71	195	11	CC..CC.....	HT)+Q]%:^=@
chr1	1865693	C	Y	122	122	244	12	,,t.T.T.T,..	`W_3U```````
chr1	1865704	C	S	130	130	243	11	,g.G.G.G,..	``H[&`````^
chr1	2393176	C	Y	74	74	244	11	.A.T.,...TT	4]```[S``
chr1	2507482	G	K	56	56	216	10	,.t,,tT,,^~.	5``F"JPNX`
chr1	2609998	A	T	65	68	140	10	....,.,ttt	D```W"""`_M

Figure 1: An example of the variant file outputted by SAMTools pileup consensus command (http://samtools.sourceforge.net/cns0.shtml).

chr1	1	247249719
chr2	1	242951149
chr3	1	199501827
chr4	1	191273063
chr5	1	180857866

Figure 2: Genome Size files. This file can be downloaded from the UCSC genomes browser (chromosome Ideogram).

585	chr1	326	327	rs112750067	0	+	T	T	C/T	genomic	single	unknown	0	0	unknown	exact 1
585	chr1	439	440	rs112155239	0	+	C	C	A/C	genomic	single	unknown	0	0	unknown	exact 1
585	chr1	468	469	rs117577454	0	+	C	C	C/G	genomic	single	by-1000genomes	0	0	unknown	exact 1
585	chr1	491	492	rs55998931	0	+	C	C	C/T	genomic	single	by-1000genomes	0	0	unknown	exact 1
585	chr1	518	519	rs62636508	0	+	G	G	C/G	genomic	single	by-1000genomes	0	0	unknown	exact 1
585	chr1	532	533	rs114315702	0	+	G	G	C/G	genomic	single	by-1000genomes	0	0	unknown	exact 1
585	chr1	582	583	rs58108140	0	+	G	G	A/G	genomic	single	unknown	0	0	unknown	exact 1

Figure 3: dbSNP file format from the UCSC genomes browser.

1	533	G	C	0.05
1	41342	T	A	0.242
1	41791	G	A	0.042
1	44449	T	C	0.017
1	44539	C	T	0.017

Figure 4: 1000 genomes file format.

ATTA	0.0050037563772859
ATGC	0.00546448965979923
ATCG	0.00455473665280256
CGTA	0.00847696613843467
CGGC	0.00452029608542015
CGAT	0.00401272654571474

Figure 5: Global Nucleotide Mismatch rate format.

#Chr	Pos	Ref	Cons_base	Cons_Qual	SNP_Qual	RMS	Coverage	Read_base	BQ	Mismatched	Matches	LMR
chr1	856086	C	M	54	54	231	10	.,...a,,AA	Q^Y;$```	20	224	0.081967213
chr1	986078	C	S	62	62	246	12	G,,..,...,gg	L``QQ``\^1`]	2	232	0.008547009
chr1	1131445	A	W	51	51	186	10	.,tt.,...t	`8````Y`8A	7	240	0.028340081
chr1	1223101	G	R	80	80	247	11	,aA,,,a.,,.	``````WO\`S	4	258	0.015267176
chr1	1819820	A	M	42	71	195	11	CC..CC.....	HT)+Q]%:^=@	7	226	0.030042918
chr1	1865693	C	Y	122	122	244	12	,,t.T.T.T,..	`W_3U```````	7	244	0.027888446
chr1	1865704	C	S	130	130	243	11	,g.G.G.G,..	``H[&`````^	18	212	0.07826087
chr1	2393176	C	Y	74	74	244	11	.A.T.,...TT	4]```[S``	2	228	0.008695652
chr1	2507482	G	K	56	56	216	10	,.t,,tT,,^~.	5``F"JPNX`	11	165	0.0625
chr1	2609998	A	T	65	68	140	10	....,.,ttt	D```W""`_M	16	205	0.07239819

Figure 6: SAMTools pileup format with additional Local Mismatch Rate information.

1061_384_918	0	chr1	1	4	8I42M	*	0	0	TTAAACAA==========	``````````````````^^`````	NM:i:8	MD:Z:42
1130_424_1876	0	chr1	1	4	9I41M	*	0	0	CTTCGCAAT==========	B?ANHN``IV````\X\	NM:i:9	MD:Z:41
1198_1841_828	0	chr1	1	10	10I40M	*	0	0	AGAAGCTAAA=========	FCZ``P9GUPM]`]F"	NM:i:10	MD:Z:40

Figure 7: SAM format after being processed with the samtools calmd -e command

Example workflow

Below is an example workflow of how to filter false positive variants caused by FFPE DNA damage.

Example Files:

- VAR = FILE TYPE 1
- BAM = FILE TYPE 7
- GNMR = FILE TYPE 5
- GENOME = FILE TYPE 2
- DBSNP = FILE TYPE 3
- 1000G = FILE TYPE 4
- SAMPLE = FILE TYPE 6 for variant file
- GOLD = FILE TYPE 6 for gold standard variant file
- PILEUP = FILE TYPE 2
- PATH = the path to the calc_binom_pval.r file

> # Calculate Global Nucleotide Mismatch Rate with minimum coverage of 3X and maximum coverage of 100X using 400,000 random high-confidence reference sites (100,000 per base type) with 33% of the genome within the given coverage range.
> perl calculate_global_nucleotide_mismatch_rate.pl –v VAR -d DBSP -g 1000G -p PILEUP -o GNMR -s GENOME -genCov 33
>
>\# Process the BAM file with samtools calmd.
>samtools calmd -e BAM REFERENCE.fa| samtools view -Sbh - > BAM_CALMD
>
>\# Remove variants with low read diversity and calculate local mismatch rate (LMR)
>perl filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl -v VAR -b BAM_CALMD -t TMP.txt -o VAR -window 10 -minstart 3
>
>\# Calculate the LMR for a set of gold standard variants
> perl filter_on_lowReadDiversity_and_or_calc_localMismatchRates.pl -v GOLD -b BAM_CALMD -t TMP.txt -o GOLD -window 10 -cmro
>
>\# Filter variants based on their local mismatch rate
>perl filter_on_local_mismatch_Rates.pl -gold GOLD_mmRate.txt -sample VAR_sameStart_mmRate.txt -o VAR_sameStart_mmRateFiltered.txt
>
>\# Filter variants using the Global Nucleotide Mismatch Rate
>perl filter_on_globalNucMismatchRate.pl -v VAR_sameStart_mmRateFiltered.txt - gnmr GNMR -o VAR_sameStart_mmRateFiltered_GNMR_filtered.txt -t TMP.txt -p PATH -tot 1000000000 -a 0.05
>