Meerkat User Manual

Version: 0.189

I. Introduction

Meerkat is designed to identify structural variations (SVs) from paired end high throughput sequencing data. It predicts SVs from discordant read pairs (pairs that mapped to reference genome in unexpected way). Then it looks for reads that cover the predicted breakpoints junctions (split read support), refines breakpoints by local alignments and predicts mechanisms that SVs are formed. It is more sensitive, with remapping of unmapped and partially mapped reads, especially when the insert size of sequencing library is small (i.e. read length is 100bp and insert size is 200bp), since the SV breakpoint has to be in-between the paired end reads to form discordant read pair. With discordant read pair, split read support and some filtering steps, it has low false positive rate. It can also take into account of reads from repetitive regions (non-uniquely mapped), combine discordant read pair clusters to predict complex events, and select the most supported and smallest events. In a word, it's a great algorithm. Of course it is, who would develop a bad algorithm any way.
Current version is tested for Illumina reads only. See reference for detailed description of methods and performances. This document is only a guide for how to use it.
Note, in this version, BWA mem alignment is fully supported. We have made some improvements of the code and suggested some new parameter sets that perform slightly better in high coverage genomes from our own experience, since the latest Illumina reads are of better quality and genomes are often sequenced at higher coverage than it used to be. Please pay attention to the parameter section. The Meerkat package can now be used to call SVs in whole exome sequencing (WES) data too. Check the parameter section for how to use it properly.

II. Prerequisites

1. **Software environment:**
    1. Unix/Linux system.
    2. CMake version 2.6.4 or above.
    3. PERL 5.8.1 or above.
    4. BioPERL 1.5.0 or above.
    5. R 2.6.1 or above (you must be able to run Rscript directly from command line).

6. Samtools 0.1.5 to 0.1.19. Note the latest version 1.x won't work. Use 0.1.x version.
7. BWA 0.6.2. You can use the latest version to align the reads and generate a bam file as input of Meerkat, but use version 0.6.2 in Meerkat for split read alignment.
8. NCBI blast 2.2.10 or above, can be downloaded at: ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/2.2.24/blast-2.2.24-x64-linux.tar.gz.
9. Blat 2.2.24 or above.
10. Primer3 2.2.0 or above.

2. Reference file and other files:

1. Reference genome sequence in fasta format and indexed by PERL module Bio::DB::Fasta. The fasta file of reference must be in a folder alone without any other files, see Bio::DB::Fasta manual for more detail. Reference sequence in 2bit format is required to setup blat server.
2. BWA index of reference file generated by "bwa index" command.
3. Fai index of reference file generated by "samtools faidx" command.
4. Gene annotation of reference genome which can be downloaded from UCSC. The file needs to be sorted by chromosome and coordinate by command "sort refGene.txt -k 3,3 -k 5,5n > refGene_sorted.txt".
5. Repeat annotation (RepeatMasker output) of reference genome which can be downloaded from UCSC.

3. Examples:

The required programs/files are configured as following for example:

1. Samtools in folder /opt/samtools/.
2. BWA in folder /opt/bwa/.
3. NCBI blast in folder /opt/blast/. Binaries of blastall and formatdb are in /opt/blast/bin/.
4. Blat and gfServer in folder /opt/blat/.
5. Reference hg18.fasta in folder /db/hg18/hg18_fasta/. Bio::DB::Fasta will index the entire folder and generate a file "directory.index" in the same folder.
6. BWA index files hg18.fasta.* of reference in folder /db/hg18/hg18_bwa_idx/.
7. Fai index file hg18.fasta.fai of reference in folder /db/hg18/.
8. Gene annotation file refGene_hg18_sorted.txt in folder /db/hg18/.
9. RepeatMasker file rmsk-hg18.txt in folder /db/hg18/.

4. Input file:

Paired end reads alignment in BAM format. The function of adjusting non-uniquely mapped reads depends on BWA aligned BAM file with XT tag to determine if a read is uniquely mapped or non-uniquely mapped, and XA tag to extract all alternative mapping positions. Meerkat can handle multiple read groups with different insert size, different read length. Alignment for multiple read groups shall be merged into one single BAM file with RG tag properly labeled. Note, "/" is not allowed in RG string. For human 75bp or 100bp reads, recommend "bwa aln" command to use "-l 40 -k 2" and "bwa sampe" command to use "-N 100" parameters. BAM file must be sorted and indexed. If the genome is aligned by BWA mem, it has to be processed by Picard or other tools to mark duplicates, see section IV and VI.

If certain read groups are to be left out of analysis due to low quality, a read group blacklist file shall be generated with one read group ID per line. See section IV.1 for details.

Note, Meerkat will generate a large number of intermediate files with same prefix as the BAM

file, it is recommended to create one folder per BAM file, place the BAM file in the folder or make soft link to the BAM file, so that all the intermediate files are generated in the same folder.

Important note, the latest version of required softwares listed in section II.1 may have new features that cause problem. If so, try older versions. For example, occasionally BWA 0.7x will abort or report CIGAR string inconsistent error in Meerkat run, in this case, use BWA 0.6.2. For another example, the new suite BLAST+ will not work for Meerkat, use legacy BLAST suite.

III. Installation

To compile mybamtools library and 3 binary files bamreader, dre and sclus, follow these steps.

1. **Build mybamtools:**
   cd ./src/
   tar xjvf mybamtools.tbz
   cd mybamtools
   mkdir build
   cd build
   cmake ..
   make
   This will populate the lib folder with the requisite shared libraries the src folder IS still necessary--header files stored here are needed to compile other binaries.

2. **Build bamreader (must have mybamtools compiled):**
   tar xjvf bamreader.tbz
   cd bamreader
   Edit Makefile and set BTROOT to the path to which mybamtools was extracted.
   vi Makefile
   BTROOT = /path/to/mybamtools
   make
   mv ./bamreader ../../bin/
   Add path to mybamtools' compiled libraries to LD_LIBRARY_PATH.
   in bash: export LD_LIBRARY_PATH=/path/to/mybamtools/lib
   in csh: setenv LD_LIBRARY_PATH /path/to/mybamtools/lib

3. **Build dre (must have mybamtools compiled):**
   tar xjvf dre.tbz
   cd dre
   Edit Makefile and set BTROOT to the path to which mybamtools was extracted.
   vi Makefile
   BTROOT = /path/to/mybamtools
   make
   mv ./dre ../../bin/

4. **Build sclus:**
   tar xjvf sclus.tbz

cd sclus
make
mv ./sclus ../../bin/

IV. Work flow

Meerkat is composed of 3 steps: pre_process, meerkat and mechanism. Each step is executed by a PERL script. Most parameters are self explainable. There are multiple sub-steps for pre_process.pl and meerkat.pl. For high coverage sequencing, one shall run sub-steps one by one, check outputs and adjust parameters if necessary. An example of run time and memory usage can be found in section IV.4. More details of the parameters are provided below.

You should try to decide what parameters are the best for your own data, or use the suggested parameters we provided for different types of data, using default parameters is not a good idea.

1. **pre_process.pl**

Usage:
perl ./scripts/pre_process.pl [options]
-b FILE sorted and indexed bam file, required
-k INT min coverage required for a nucleotide to be ignored (ignore position that covered by too many reads), 0 turn this function off, default 500
-r INT range of insert size to be plotted, insert size larger than such won't be used calculating distribution, default 1000
-n INT number of reads per read group to be used calculating insert size distribution, 0 use all reads, default 10000
-l INT [0/1], extract paired soft clipped reads and pairs with one read mapped, the other unmapped, or both unmapped, re-map mate pair, default 1
-q INT Read trimming parameter. Equivalent to BWA's -q option, default 15
-c INT bp to be cut off from beginning and end of unmapped and soft clipped reads, such reads were used to find smaller events, default 35
-s INT bp to be cut off from beginning and end of unmapped and soft clipped reads, such reads were used in split reads mapping, must be same as -s in meerkat.pl, default 20
-u INT [0/1], process uu pair (both reads unmapped in a pair) into 4 pairs, default 0
-f INT number of alternative mappings to print in XA tag for clipped alignments, default 100
-N INT Clipped reads and split reads must have <= INT Ns, default 5
-t INT number of threads used in bwa alignment, default 1
-R STR file name of a list of blacked read groups, one read group ID per line
-I STR /path/to/reference/bwa_index for bwa alignment, required if -l 1
-A STR /path/to/reference/fasta.fai for bwa alignment, required if -l 1
-S STR /path/to/samtools, path only, not the command, no need to specify if samtools is in PATH
-W STR /path/to/bwa, path only, not the command, no need to specify if bwa is in PATH
-P STR specify step to run, [all|is|cl], default all
is: extract unmapped, soft clipped reads, calculate insert size distribution
cl: map soft clipped mate pairs to reference genome, if you are using large clusters to

run, you may split cl step into two stages, cl1 and cl2, cl1 can run on multiple threads controlled by t parameter, cl2 stage only runs on single thread.

all: run all above steps

-h help

-k option is aimed to filter out reads mapped to telomeres or centromeres by filtering out positions with unusually high coverage. In cancer genomes, it's common to observe copy gain events. One shall use higher value such as 1500 in order not to filter out such events.

-l option is to control whether to re-map some unmapped reads and partially mapped reads or not. If insert size is small, it's common to have a read cover a SV break point. Such read would be unmapped or partially mapped. For read pair that one end is mapped, the other end is partially mapped, the unmapped part of partially mapped read is extracted and pair with mapped read to form an artificial read pair. For read pair that one end is mapped, the other end is unmapped, a portion from beginning and a portion from the end of the unmapped read are extracted and form two artificial read pairs with the mapped read. The portion to be extracted is controlled by -c option. Such artificial read pairs are re-mapped to reference. It can greatly increase sensitivity if insert size is small and read is long (75bp or 100bp). For short read length (50bp or shorter), set this option to 0. For BWA mem aligned genomes, re-mapping is unnecessary, so turn this function off and also turn it off in meerkat.pl.

-q option is to control trimming off poor quality bases from the end of a read. Don't set to 0 unless you know what you are doing.

-u option is to control generating 4 artificial pairs from both end unmapped read pairs. It is useful to identify small events if the sequencing quality is very good and genome is not too repetitive. For human genome, recommend to turn this function off.

-c option is to control how much to extract from unmapped or partially mapped reads to form artificial read pairs, see also -l option. It should be slightly small than ½ read length. The extracted part will be aligned against the original reference genome. So one should consider mappability when selecting this parameter. Default is good for human genome with long read length. One can use smaller value if the genome is less repetitive.

-s option is to control how much to extract from unmapped or partially mapped reads to generate split reads. Since split reads will be mapped to break point regions of SVs predicted from discordant read pairs, the value can be set small without sacrificing mappability. It should be between 1/5 and 1/3 of read length.

-R specifies the file that contains the list of read group IDs that shall be left out of analysis. Recommend not to process a read group if the uniquely mapped reads are less than 30%. If all the read groups are of high quality, you don't need to specify this option.

For example, for 50bp reads, <10x TCGA genomes, suggest to use "-s 18 -l 0 -q 0". For 75-101bp reads, 20-30x and 60-80x TCGA genomes, suggest to use "-s 20 -k 1500 -q 15". For BWA mem aligned genomes, use "-s 20 -k 1500 -q 15 -l 0". For NA18507 run in our original publication, use all defaults. For TCGA WES data, use "-s 20 -k 10000 -q 5".

2. **meerkat.pl**

Usage:

perl ./scripts/meerkat.pl [options]

-b FILE sorted and indexed bam file, required

-k INT [0/1], use black list generated in pre_process.pl, default 1

-d FLT standard deviation cutoff to call discordant mate pairs, default 3

-c FLT standard deviation cutoff to cluster discordant mate pairs, default equal to -d, it is recommended to use same -c as -d if -d<=5, -c 5 if -d > 5

-p INT number of supporting mate pairs required for an event to be called, default 2

-o INT number of supporting full length read pairs, default 0, specify this option will decrease sensitivity on small complex events

-q INT number of supporting split reads required for an event to be called, default 1

-z INT event size cutoff, default 1,000,000,000

-s INT bp to be cut off from beginning and end of unmapped reads, must be same as -s in pre_process.pl, default 20

-m INT [0/1], if set to 1, use Meerkat to remove duplicates; if set to 0, use flag 'd' marked by Picard or other tools to remove duplicates. If the bam file is aligned by bwa mem, it has to be processed by Picard to mark duplicates and use 0 option. The bwa mem aligned bam file won't work with option 1. Default 1

-a INT [0/1], adjust non-uniq mapped reads, default 1

-u INT [0/1], use all alignments in the BAM file, turn this option on if the BAM file is not generated by BWA, turn on this option will force turning off option a, default 0

-Q INT minimum mapping quality for reads to be used, default 0

-g INT number of alternative mappings to consider in main bam file, number of alternative mappings printed out in XA tag by bwa is controlled by -N, default use all in bam file

-f INT number of alternative mappings to consider in clipped alignments, default use all in bam file

-l INT [0/1], consider clipped alignments, default 1

-t INT number of threads used in bwa alignment, default 1

-R STR file name of a list of blacked read groups, one read group ID per line

-F STR /path/to/reference/fasta/files, path only, not the files, required

-S STR /path/to/samtools, path only, not the command, no need to specify if samtools is in PATH

-W STR /path/to/bwa, path only, not the command, no need to specify if bwa is in PATH

-B STR /path/to/blastall and formatdb, path only, not the command, no need to specify if blastall and formatdb is in PATH

-P STR specify step to run, dc|cl|mpd|alg|srd|rf|all, default 'all', each step require results from previous steps

dc: extract discordant read pairs

cl: construct clusters of discordant read pairs

mpd: call events based on read pairs

alg: align split reads to candidate break point regions, if you are using large clusters to run, you may split cl step into two stages, alg1 and alg2, cl1 can run on multiple threads controlled by t parameter, alg2 stage only runs on single thread.

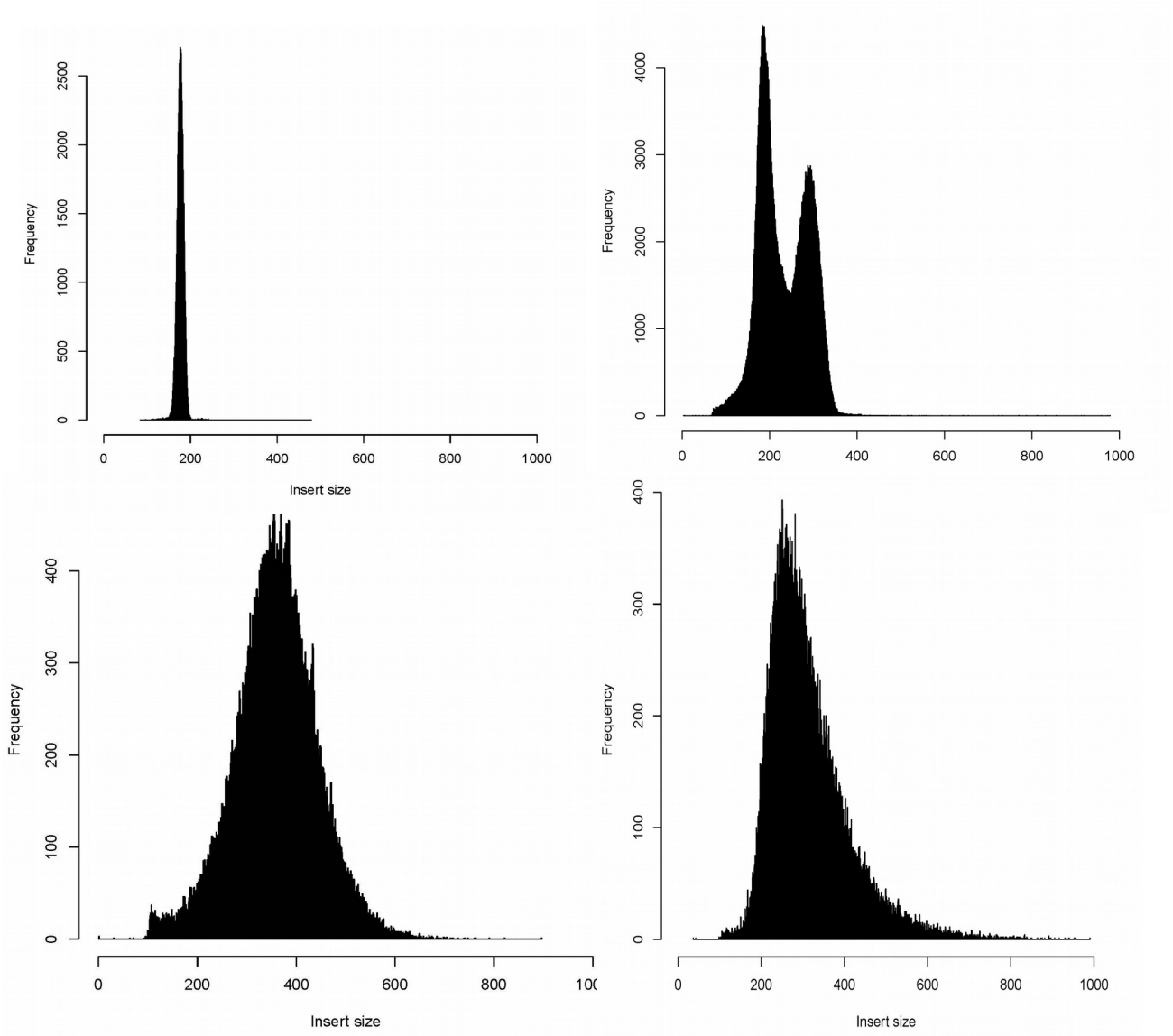
srd: confirm events based on split reads and filter results

rf: refine break points by local alignments

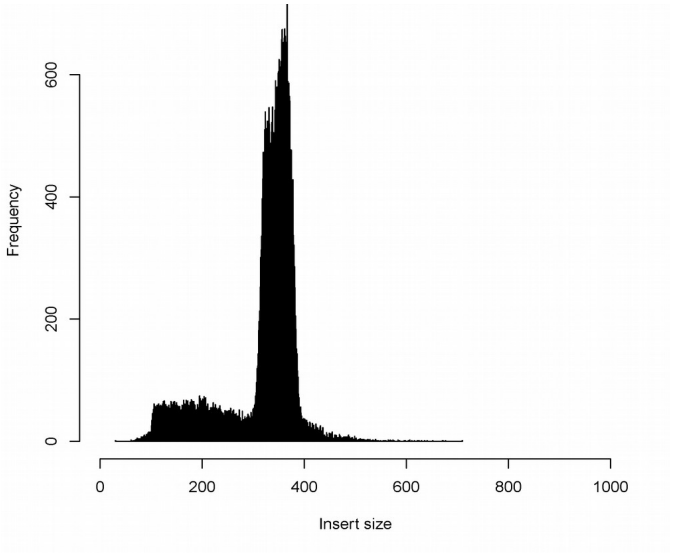
all: run all above steps

-h help

-d option is to control how to call discordant read pairs. It is equivalent to define what is the largest fragment (median insert size + d x sd) that could be concordant. Use default if the insert size distribution is narrow and symmetric (top 2 plots in next page), and use 5 if it's wide or skewed with long tail (bottom 2 plots in next page). Don't use value less than 3. For deep coverage (>30x), even with narrow peak using 5 may give slightly better performance.



If the insert size distribution looks like the plot below, the peak is narrow but there is still a tail



on the right side, which is the case for most TCGA genomes, use -d 5 in meerkat.pl will be better than -d 3.

-c option is to control how to merge discordant read pairs into clusters and construct confidence interval of break points. It is equivalent to define what is the largest fragment (median insert size + c x sd) that could cover a break point. If -d is 5 or less, use same c. If use very large d, such as 10, use smaller value of c, such as 5. If the coverage is really high, or there are extensive copy gain events, use 5 rather than 3 to avoid no confidence interval of break points can be constructed.

-a g and f options are to control how to handle non-uniquely mapped reads. Set a to 0 to turn such function off if sequencing quality is not good or low coverage sequencing. If turning off, both reads in a pair are required to be uniquely aligned. It will depend on "XT" tag of uniqueness of mapping generated by BWA. If no XT tag, use option Q. Option g and f are to control number of alternative mapping positions to consider. The number of alternative mapping positions BWA printed out into XA tag is controlled by -N parameter in "bwa sampe" command.

-u option is to control if to use all alignments in BAM file or not. Usually, BAM file generated by BWA is recommended. For BAM file that is not aligned by BWA or aligned by BWA but there is no "XT" tag, you shall turn on this option. When you turn on this option, all reads will be used and treated as uniquely mapped, you shall use option Q to specify a minimum allowed mapping quality to get rid of poor mapping reads. For option Q, recommend to use 10. For BWA aligned BAM, you can still use option Q to remove poor mapping reads.

-p o and q options are to control the stringency of calling SVs. If using -a 1 -l 1, it's recommended to use -o 1 to avoid too many artifacts arisen from repeats. However, specifying o option will decrease the sensitivity for small events, especially small complex events.

For example, for 50bp reads, <10x TCGA genomes, suggest to use "-s 18 -d 5 -a 0 -l 0 -q 1". For 75-101bp reads, 30-40x TCGA genomes, suggest to use "-s 20 -d 5 -p 3 -o 1 -a 0 -u 1 -Q 10" ("-a 0 -u 1 -Q 10" is used because most of these genomes lack XT and XA tags so the -a 1 option won't work). If aligned by BWA mem, use "-s 20 -d 5 -p 3 -o 1 -m 0 -l 0". For 101bp reads, 60-80x TCGA genomes, suggest to use "-s 20 -d 5 -p 5 -o 3". If tumor genome is sequenced at 60x and normal genome at 30x, use 60x parameters for both tumor and normal. Always replace the *.blacklist.gz file for cancer genome by the file generated from the matched normal genome (discard the cancer.blacklist.gz file, put the normal.blacklist.gz file into the cancer folder and rename it to cancer.blacklist.gz). For NA18507 run, use "-s 20 -p 3 -o 1". For TCGA WES, use same parameter as 30-40x whole genome.

3. **mechanism.pl**

   Usage:
   perl ./scripts/mechanism.pl [options]
   -b FILE sorted and indexed bam file, required
   -o INT [0/1], include rmsk type \"Other\" in TE, default 1
   -t INT max size of TE, default 100,000
   -z INT size limit of SVs to be processed, default 1,000,000,000
   -R STR /path/to/repeat_mask_file, required, can be downloaded from UCSC
   -h help

   Example for NA18507 run:
   perl ./scripts/mechanism.pl -R /db/hg18/rmsk-hg18.txt -b na18507.sorted.bam

4. **Benchmark**
   For HapMap individual NA18507 (42x coverage, 100bp read length, 500bp insert size), it took about 1.5 days and 10GB memory when using 10 threads for BWA alignment. For 30x coverage cancer genome, it can take more than 2 days and more than 30GB of memory. If the sequencing quality is not very good, such as many chimeric reads or many non-uniq mapped reads, it can take a lot longer and a lot more memory.

5. **Example**
   There is an example bam file at example/example.sorted.bam, and 3 output files. After running pre_process.pl and meerkat.pl, you should get 2 files, example.intra.refined.typ.sorted and example.inter.refined.typ.sorted. After running mechanism.pl, you should get example.variants file. Otherwise, you should go back and make sure you have everything set up correctly.

6. **Special attention**

When you run Meerkat on cancer genome, it would be better to use the blacklist.gz file generated from its matched normal genome rather than use the blacklist.gz file generated from the cancer genome. These files are generated by pre_process.pl.

Sometimes, you will get error message "differing read lengths" in pre_process.pl, you can ignore this message. It's only telling you the read lengths are not the same in some read groups.

V. Outputs

Output files:

prefix/*_1/2.fq.gz: artificial read pairs per read group.

prefix/*.is: sample insert size per read group (used to generate prefix.pdf).

prefix.sr/*.srout: split read mapping of each breakpoint, file name is discordant cluster ID. The top line is reference seq on one side of the breakpoint, the bottom line is reference seq on the other side of the breakpoint. The lines in between are reads spanning the breakpoint junction.

prefix.blacklist.gz: positions that shall not be processed.

prefix.bp.fasta: sequences of confidence intervals of break points.

prefix.bp.info: table of cluster id and corresponding break points.

prefix.bp_reads: table of break points and supporting read names.

prefix.cl.disc.sorted.bam: discordant reads in re-mapped artificial read pairs.

prefix.cl.dup.bam: duplicates in re-mapped artificial read pairs.

prefix.cl.sorted.bam: BAM file for alignments of re-mapped artificial read pairs.

prefix.clusters: all discordant clusters sorted by weight (one discordant read pair per line).

prefix.disc.sorted.bam: discordant reads in original BAM file.

prefix.discord: summary of discordant clusters (one discordant cluster per line).

prefix.dre.log: log file to extract discordant read pairs.

prefix.dup.bam: duplicates in original BAM file.

prefix.inter/intra.refined.typ.sorted: variants after refine break points by local alignments, sorted by event type.

prefix.isinfo: read length and insert size statistics per read group.

prefix.mapping.raw: clusters with all possible mapping positions, sorted by coordinates.

prefix.mp.inter/intra.out: variants predicted by discordant read pair clusters.

prefix.pdf: plots of insert size distribution and quality of reads.

prefix.pre.log: log file for extracting unmapped, partially mapped reads and generating artificial read pairs and split reads.

prefix.softclips.fq.gz: partially mapped reads after trimming poor quality basepairs from end.

prefix.softclips.rdist: base quality distribution of partially mapped reads (used to generate prefix.pdf).

prefix.sr.1/2.fq.gz: split read sequences in fastq format.

prefix.sr.inter/intra.filtered: filtered variants of prefix.sr.inter/intra.out (remove variants that use same cluster).

prefix.sr.inter/intra.out: variants have split reads support.

prefix.sr.sorted.bam: split reads alignments.

prefix.unmapped.fq.gz: unmapped reads.

prefix.unmapped.rdist: base quality distribution of unmapped reads (used to generate prefix.pdf).

prefix.variants: final variants with mechanism prediction. **This is the result file you should look at.**

Event types:

del: simple deletion, no insertion at break point.

del_ins: deletion with insertion at the break point with unknown source.

del_inssd: deletion with insertion at the break point, insertion comes from the same chromosome, same orientation and downstream of deletion.

del_inssu: deletion with insertion at the break point, insertion comes from the same chromosome, same orientation and upstream of deletion.

del_insod: deletion with insertion at the break point, insertion comes from the same chromosome, opposite orientation and downstream of deletion.

del_insou: deletion with insertion at the break point, insertion comes from the same chromosome, opposite orientation and upstream of deletion.

del_inss: deletion with insertion at the break point, insertion comes from a different chromosome, same orientation.

del_inso: deletion with insertion at the break point, insertion comes from a different chromosome, opposite orientation.

del_invers: deletion with inversion at the break point, inversion comes from deleted part.

inssd: insertion, insertion comes from the same chromosome, same orientation and downstream of deletion.

inssu: insertion, insertion comes from the same chromosome, same orientation and upstream of deletion.

insod: insertion, insertion comes from the same chromosome, opposite orientation and downstream of deletion.

insou: insertion, insertion comes from the same chromosome, opposite orientation and upstream of deletion.

inss: insertion, insertion comes from a different chromosome, same orientation.

inso: insertion, insertion comes from a different chromosome, opposite orientation.

invers: inversion with reciprocal discordant read pair cluster support

invers_f: unpaired cluster, both end of read pairs mapped to same chromosome, both on forward strand.

invers_r: unpaired cluster, both end of read pairs mapped to same chromosome, both on reverse

strand.

tandem_dup: tandem duplication.

transl_inter: inter chromosomal translocation.

Note that del, del_ins, invers_f, inver_r, tandem_dup are all predicted from unpaired clusters. They can arise from other complex events while only one cluster is called, the other cluster is not. They can also arise from intra chromosomal translocations rather than a true deletions or tandem duplications.

Types of mechanisms:

TEI: transposable element insertion, single or multiple TE insertion.

TEA: Alternative TE, usually a deletion with insertion event is called, deletion is a TE in reference genome and insertion is the same type of TE in reference genome. It should be arisen from sequence divergence of TE.

VNTR: variable number of tandem repeat, deletion or insertion of satellite repeat, simple repeat or low complexity repeat.

NAHR: non-allellic homologous recombination, >100bp homology.

alt-EJ: alternative end joining, 3-100bp homology.

NHEJ: non-homologous end joining, 0-2bp homology or 1-10bp insertion at deletion break point.

FoSTeS: fork stalling and template switching, template switch, >10bp insertion at deletion break points.

NA: unclassified.

Format for final variants (prefix.variants):

del, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of deletion (2 col), deletion size, homology sizes, annotation of break points

del_ins, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of deletion (2 col), deletion size, chr (donor), range of insertion (2 col), insert size, annotation of break points

del_inss/o*, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of deletion (2 col), deletion size, chr (donor), range of insertion (2 col), insert size, distance of deletion and insertion, homology at break points, annotation of break points

del_invers, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of deletion (2 col), deletion size, chr (donor), range of inversion (2 col), inversion size, distance of inverstion and deletion (2 col), homology at break points, annotation of break points

inss/o*, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of deletion (2 col), deletion size, rchr (donor), ange of insertion (2 col), insert size, distance of deletion and insertion, homology at break points, annotation of break points

invers, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, inversion left boundary, inversion right boundary, inversion size, homology at break points, annotation of break points

invers_*, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, inversion left boundary, inversion right boundary, inversion size, homology at break points, annotation of break points

tandem_dup, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, tandem duplication boundary 1, tandem duplication boundary 2, tandem duplication size, homology at break points, annotation of break points

del_inss/o, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of deletion (2 col), deletion size, chr of insertion donor, range of insertion (2 col), insert size, homology at break points, annotation of break points

inss/o, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr, range of insert site, insert site size, chr of insertion donor, range of insertion (2 col), insert size, homology at break points, annotation of break points

transl_inter, mechanism, cluster id, number of supporting read pairs, number of supporting split reads, chr of 1st cluster, boundary of 1st cluster, orientation of 1st cluster, chr of 2nd cluster, boundary of 2nd cluster, orientation of 2nd cluster, homology at break points, annotation of break points

Note1: For homology at break points, positive number means homology, negative number means additional nt at break point. For annotation of break points, break points are annotated sequentially by repeat contents seperated by "_". SR: satellite repeat, simple repeat or low complexity repeat. TE: transposable elements. For example, BP:_TE__SR means the second break point is TE and fourth break point is SR, first and third break points are neither TE nor SR.

Note2: For complex events, such as deletion with insertion, deletion with inversion, typically, 2 clusters are paired to call one event, thus, 2 cluster IDs are given. However, it is possible that there is a third cluster span the deletion depend on event size and insert size of library. In such case, 3 cluster IDs are given.

Examples:

del TEI_SINE_AluY 2790 33 12 chr1 17578190 17578503 312 15 BP:TE_TE

del_ins NHEJ 621 84 48 chr4 173225214 173229507 4292 - - - 1 BP:TE_TE

del_inssu FoSTeS 1434_0/1005_0 52/62 6/6 chr6 56012024 56017028 5003 chr6 55938323 55939086 764 72938 2/3 BP:_TE__TE

del_inssd FoSTeS 416_0/662_0/2774 99/82/33 9/18 chr3 147867879 147873095 5215 chr3 147877553 147877855 303 4458 2/-9 BP:TE_TE_TE_TE

insod NA 1146_0/2090_0 58/42 10/20 chr1 104382016 104382013 -4 chr1 104383501 104384051 551 1488 6/-4 BP:___

inss NA 5424_2/6914_0 42/22 20/12 chr9 67959791 67959796 4 chr3 84522750 84522975 226 -5/0 BP:___TE

invers NA 941_0/2146_0 65/41 12/6 chr6 130889878 130893988 4110 1 BP:_TE

tandem_dup NA 1171 57 5 chr9 118553273 118555818 2545 1 BP:TE_

invers_r NA 1727 24 4 chr17 7528656 12039589 4510933 2 BP:TE_SR

transl_inter NA 3649 10 1 chr17 7528046 -1 chr5 138794678 1 0 BP:TE_

Format for variants with refined break points (prefix.inter/intra.refined.typ.sorted):

Same as prefix.variants except without mechanism and annotation of break points columns.

Format for variants with split read support (prefix.sr.inter/intra.out/filtered):

Same as prefix.inter/intra.refine.typ.sorted except without level of homology at break points column.

Format for variants with discordant read pair support (prefix.mp.inter/intra.out):

Same as prefix.sr.inter/intra.out except without number of split read support column. The last one or two columns are confidence interval of break points separated by “:”.

Format for discordant read pair clusters (prefix.discord):

primary cluster ID, secondary cluster ID, number of discordant read pairs, chrA, posA, strandA, chrB, posB, strandB, distance

Format for detailed discordant read pair clusters (prefix.clusters):

primary cluster ID, secondary cluster ID, ith cluster, weight, number of mismatches, read name, read group, chrA, strandA, posA, read lengthA, chrB, strandB, posB, read lengthB, distance

Note:

All nucleotides of given coordinates in the final output are retained at the breakpoint in the donor genome. For example, a deletion of chrA:1234-5678 will be as the following in the donor:

```
          1234|              |5678
      ---------------|              |----------------
```

The deleted part is 1235-5677. A tandem duplication of chrA:1234-5678 will be:

```
              1234      5678|1234       5678
      -------------==========|==========---------------
```

If you have short reads (i.e. read length 36bp) and reference genome is quite repetitive, it’s not recommended to use split read to filter variants. You should stop after running “meerkat.pl -P mpd” and check prefix.mp.inter/intra.out for results.

VI. Filtering Variants

The variants reported in file prefix.variants are raw calls. It should be filtered to produce more reliable calls. If you are calling somatic SVs in cancer genomes, use somatic_sv.pl to filter out germline events and other artifacts. It works for both high coverage (>20x) and low coverage genoms (<10x). You can also use somatic_sv.pl to get high confidence germline SVs in normal genomes (>20x). If the BAM file is not generated by BWA or XT tag is missing, use option Q for filtering based on mapping quality.

1. somatic_sv.pl

Usage:

perl ./scripts/somatic_sv.pl [options]

-i FILE input file, required

-o FILE output file, required

-D FLT standard deviation cutoff to call discordant read pairs, use the same value as used in d option of meerkat.pl, default 3

-I FILE isinfo file from Meerkat run, required if n or e option is turned on

-K FILE file name of read group to be ignored, one read group ID per line, same as R option in pre_process.pl and meerkat.pl. If all the read groups are of high quality, you don’t need to specify this option. If there is any blacklist rg and any of the options n, u, f, e is enabled, K option is required.

-F STR name of folder contains all *.discord files from normal genomes to filter germline events, we recommend to filter against all normal genomes from one tumor type

-x INT number of discordant read pairs supporting the same event to be used as filter from *.discord file, default 2

-l INT distance of breakpoints to filter germline events from *.discord file, default 500

-s INT minimum size of simple events, default 100

-E INT filter TEI for deletions, these are typically germline events, default 1

-d INT max homology allowed for deletion and intra-chr events, 0 to disable, default 100

-t INT max homology allowed for inter chromosomal translocation events, 0 to disable, default 100

-m INT filter events that both breakpoints fall in satellite or simple repeats, default 1

-n INT filter by total number of discordant read pairs in matched normal genome, if certain number of discordant read pairs are observed in the given genome, discard the event, if enable this option, B and I options are required, default 0

-y INT maximum fraction of discordant pairs in normal bam, default 0.1

-u INT filter by non-uniq mapped reads in matched normal genome, determined by XT tag or mapping quality, if too many non-uniq mapped reads are observed in given genome, discard the event, if enable this option, B option is required, default 0

-v INT window size to look for non-uniq mapped reads in normal bam file, default 100

-r FLT cutoff of ratio of non-uniq mapped reads to all mappable reads, default 0.25

-f INT filter by soft-clipped reads in matched normal genome, if certain number of soft-clipped reads are observed in the given genome, discard the event, if enable this option, B option is required, default 0

-g INT window size to look for soft-clipped reads in normal bam file, default 10

-j INT cutoff for number of soft-clipped reads in normal bam file, default 3

-e INT filter by discordant read pairs in tumor genome, if certain breakpoint has no discordant read pair, it is an artifact of split read mapping, discard the event, if enable this option, B and I options are required, default 0

-B FILE bam file

-k INT window size to look for discordant read pairs in tumor bam file, default 800

-z INT to enable parameter p, q and P, default 0

-p INT filter by number of supporting discordant read pairs, default 3

-q INT filter by number of supporting split reads, default 1

-P INT filter by sum of supporting discordant read pairs and supporting split reads, default 6

-M INT filter by mate position of split reads, for reads covering breakpoint junctions, their mate should map near breakpoints, if enable this option, B option is required, default 0

-N INT window size to look for mate in tumor bam file, default 2000

-C FILE bp_reads file from Meerkat output, required if M option is enabled

-Q INT minimum mapping quality for reads to be used, default 0, if use the Q option in meerkat.pl, use the same value here

-b INT allele frequency cutoff

-V STR blat server, if enable option V and T, will filter by blat the split reads against whole genome, and require both side of breakpoints to be best hit. One needs to set up a blat server before using this filter (i.e. gfServer start 10.11.240.76 17777 /reference/hg18/hg18.2bit -stepSize=5)

-T STR blat port

-L STR /path/to/blat, path only, not the command, no need to specify if blat is in PATH

-A STR location of prefix.sr folder from Meerkat run.

-S STR /path/to/samtools, path only, not the command, no need to specify if samtools is in PATH

Please note that the b option in previous versions has been merged into B option, and the w option is removed.

Example of command to get high confidence somatic SVs from either 30-40x or 60x TCGA genomes aligned by either BWA sampe or mem and TCGA WES, all of the following steps need to be run sequentially:

perl ./scripts/somatic_sv.pl -i $variantsfile -o $somatica -F all_normal_disc/ -l 1000 -R /db/hg18/rmsk-hg18.txt

perl ./scripts/somatic_sv.pl -S /home/ly55/opt/samtools/ -i $somatica -o $somaticb -R /db/hg18/rmsk-hg18.txt -n 1 -D 5 -Q 10 -B $normal_bam -I $normal_isinfo -K $normal_blacklistrg

perl ./scripts/somatic_sv.pl -S /home/ly55/opt/samtools/ -i $somaticb -o $somaticc -R /db/hg18/rmsk-hg18.txt -u 1 -Q 10 -B $normal_bam -K $normal_blacklistrg

perl ./scripts/somatic_sv.pl -S /home/ly55/opt/samtools/ -i $somaticc -o $somaticd -R /db/hg18/rmsk-hg18.txt -f 1 -Q 10 -B $normal_bam -K $normal_blacklistrg

perl ./scripts/somatic_sv.pl -S /home/ly55/opt/samtools/ -i $somaticd -o $somatice -R /db/hg18/rmsk-hg18.txt -e 1 -D 5 -Q 10 -B $tumor_bam -I $tumor_isinfo -K $tumor_blacklistrg

perl ./scripts/somatic_sv.pl -S /home/ly55/opt/samtools/ -i $somatice -o $somaticf -R /db/hg18/rmsk-hg18.txt -z 1

perl ./scripts/somatic_sv.pl -S /home/ly55/opt/samtools/ -i $somaticf -o $somaticg -R /db/hg18/rmsk-hg18.txt -d 40 -t 20

The all_normal_disc folder contains all the prefix.discord files from normal genomes. Don’t use -F option in somatic_sv.pl with n, u, f, e or z options, it will be extremely slow. Run those filters in separate steps. The isinfo file is generated by pre_process.pl. When using option n and e, isinfo file needs to be provided.

Example of command to get high confidence somatic SVs from 6-8x genomes, all of the following steps need to be run sequentially:

perl ./scripts/somatic_sv.pl -i $variantsfile -o $somatica -F all_normal_disc/ -R /db/hg19/rmsk-hg19.txt -x 1

perl ./scripts/somatic_sv.pl -i $somatica -o $somaticb -R /db/hg19/rmsk-hg19.txt -M 1 -B $tumor_bam -C $tumor_bp_reads

perl ./scripts/somatic_sv.pl -i $somatica -o $somaticc -R /db/hg19/rmsk-hg19.txt -L /opt/blat/ -V "10.11.240.76" -T 17777 -A $srout_dir

perl ./scripts/merge_sv.pl $somaticd $somaticb $somaticc

We found the above c and d filters are too stringent and will miss some real events, but merging the events together performs better. The $somaticd file is the final output.

Example of command to get high confidence normal or germline SVs from 30-40x genomes, all of the following steps need to be run sequentially:

perl ./scripts/somatic_sv.pl -i $variantsfile -o $germa -R /db/hg19_bi/rmsk-hg19_bi.txt -E 0

perl ./scripts/somatic_sv.pl -i $germa -o $germb -R /db/hg19_bi/rmsk-hg19_bi.txt -E 0 -e 1 -D 5 -Q 10 -B $normal_bam -I $normal_isinfo -K $normal_blacklistrg

perl ./scripts/somatic_sv.pl -i $germb -o $germc -R /db/hg19_bi/rmsk-hg19_bi.txt -E 0 -u 1 -Q

10 -B $normal_bam -K $normal_blacklistrg

perl ./scripts/somatic_sv.pl -i $germc -o $germd -R /db/hg19_bi/rmsk-hg19_bi.txt -E 0 -d 40 -t 40

perl ./scripts/somatic_sv.pl -i $germd -o $germe -R /db/hg19_bi/rmsk-hg19_bi.txt -E 0 -z 1 -p 5 -P 10

Eventually, one might consider to manually inspect the discordant read pair alignments with IGV and inspect the split read alignments from prefix.sr folder to make sure the variants are convincing. For split read alignments, one may need to manually edit the alignment a little bit. For the reads spanning the breakpoint junction, part of the reads should align to the left-hand side of reference seq on the top, the remaining of the reads should align to the right-hand side of reference seq on the bottom.

For TCGA WES, extra filtering steps are needed: 1. discard samples that have >=100 somatic fusions. 2. apply filter_fusions.pl as following:

perl ./scripts/filter_fusions.pl -i $somaticg_fusion -o $somatich_fusion -m 4 -s 20000

The input file $somaticg_fusion is generated by fusions.pl from the final somatic variants (see section VII.2 for fusions.pl usage).

VII. Utilities

1. **Convert Meerkat to VCF format (meerkat2vcf.pl).**
   Meerkat output can be converted to VCF format by meerkat2vcf.pl.

   Usage example:
   perl ./scripts/meerkat2vcf.pl -i variantfile -o vcffile -H headerfile -F /db/hg18/hg18_fasta/

   You have to generate a header file by yourself and the -F option is the same as -F in meerkat.pl. An example header file can be found in Meerkat.example package.

2. **Fusion annotation (fusions.pl).**
   For functional analysis, you can use fusions.pl to annotate fusion events. The output has uniformed format for all breakpoints (1 breakpoint per line). The Meerkat output has different format for different event type. Therefore, it's easier to use this output for certain purposes such as draw circos plot, compare with other calls, etc. Gene annotation of reference genome can be downloaded from UCSC. The file needs to be sorted by chromosome and coordinate by command "sort refGene.txt -k 3,3 -k 5,5n > refGene_sorted.txt".

   Usage example:
   perl ./scripts/fusions.pl -i variantfile -G /db/hg18/refGene_hg18_sorted.txt

   Output format:
   type1 type2 type3 chrA posA oriA geneA exon_intronA chrB posB oriB geneB exon_intronB event_type mechanism event_id disc_pair split_read homology partners

gene-gene head-tail in_frame chr21 38778477 1 ERG I1 chr21 41797322 -1 TMPRSS2 I1 del NHEJ 2284_0 16 6 0 3-5

The fusion types predicted should be self explainable. When the fusion is annotated as “no impact”, it means it’s an intronic deletion that would not alter gene function. For the above example, the 5’ of TMPRSS2 and 3’ of ERG is fused together and formed an in frame fusion. The 5’ and 3’ partners information is encoded in the partners column. The “in frame” and “out of frame” prediction is only based on reference, a predicted out of frame fusion can actually be an in frame fusion if spliced differently.

3. **Design PCR primers for validation (primers.pl).**
   The fusions generated by the fusions.pl shall be used as input.

   Usage:
   -i FILE input file, required, fusions file generated by fusions.pl
   -o FILE output file, required
   -p STR prefix of primers
   -c INT column offset, default 0, if you have sample name in column 1 for fusion events, set to 1
   -f INT flanking region, default 500, the region to design primers in
   -s STR primer sizes seperated by ",", default 20,23,25,27
   -m STR primer min, opt, max Tm seperated by ",", default 50,60,65
   -m STR primer min, max GC content seperated by ",", default 40,60
   -n INT number of primers to design for each primer size, default 5
   -r INT mask repeats, default 0
   -q INT print out flanking sequences to design primers
   -F STR /path/to/reference/fasta/files, path only, not the files, required, same as option F in meerkat.pl
   -P STR /path/to/primer3_core, path only, not the command, no need to specify if primer3_core is in PATH
   -L STR /path/to/blat, path only, not the command, no need to specify if blat is in PATH
   -V STR blat server (e.g. run server as: “gfServer start 10.11.240.76 17777 /reference/hg18/hg18.2bit -stepSize=5”, the server name will be 10.11.240.76)
   -T STR blat port (17777 in above example)
   -h help

   Output format:
   ID, primer seq, primer position, primer size, tm, gc content, number of blat hit in whole genome, distance to breakpoint

   All primers are designed by Primer3. For each event, pick one primer from both .1 and .2. The orientation has been considered for different types of events, so when ordering primers, just copy and paste the seq, no need to reverse compliment. If the seq is in lower case, that means the primer is in a repeat region. Ideally, you should pick upper case primer with 1 blat hit. If number of blat hit is 0, that means there are too many hits (don't pick such primer). Sometimes, even the primer is in repeat region (lower case), it's still unique in the genome (1 blat hit) because it's a quite diverged repeat element. It's

OK to pick such primer. Based on my experience, a primer with less than 20 blat hit will work fine if there is no uniq one available. As some primers may not work well, you can pick 2 forward primers and 2 reverse primers to perform 4 PCR reactions at the same time. General rules of primer design still apply, for example, you should pick primers that Tms are not too different and GC contents are not too extreme.

Example, for the following event (an EML4-ALK gene fusion from a lung adeno carcinoma patient):

```
gene-gene	head-tail	in_frame	2	29446548	1	ALK	I19	2
	42507907	1	EML4 I5	invers_f	NA	1511021_0	3	1	4
	3-5
```

You will get output like this:

```
LY20131231_1.1	GAGTCTGCGGTGCTGTGATA	84,20	60.016	55.000	1	416
LY20131231_1.1	TGAAGCACTACACAGGCCAC	393,20	59.905	55.000	1	107
LY20131231_1.1	ATTCAGCCCCTACACTGCAC	106,20	60.142	55.000	1	394
LY20131231_1.1	ACATTCAGCCCCTACACTGC	104,20	60.142	55.000	1	396
LY20131231_1.1	TGCGGTGCTGTGATAACATT	89,20	60.142	45.000	1	411
...
LY20131231_1.2	ATCCACAGATTCAGCCAACC	160,20	59.934	50.000	4	340
LY20131231_1.2	CCCGTGGATACAGGACAACT	416,20	59.844	55.000	1	84
LY20131231_1.2	TGGGTTGAAAATATTTGGGG	181,20	59.497	40.000	1	319
LY20131231_1.2	TCCTAAAATCAGTCCCCCGT	401,20	60.683	50.000	1	99
LY20131231_1.2	CATCCACAGATTCAGCCAAC	159,20	59.090	50.000	4	341
...
```

I would pick the following 2 primers, and the expected PCR product will be (394+84) bp.

```
LY20131231_1.1	ATTCAGCCCCTACACTGCAC	106,20	60.142	55.000	1	394
LY20131231_1.2	CCCGTGGATACAGGACAACT	416,20	59.844	55.000	1	84
```

4. Calculate allele frequency (discon.pl).

This script will give you the number of discordant and concordant read pair across all breakpoints.

Usage:

-i FILE input variants file, required

-o FILE output file, required

-D INT count number of discordant pairs from bam, [0/1], turn this function on for genotyping, default 0

-B FILE bam file, required

-C FILE cluster file generated by Meerkat, required

-I FILE isinfo file from Meerkat run, required

-K FILE file name of read group to be ignored, one read group ID per line, same as R option in meerkat.pl

-S STR /path/to/samtools, path only, not the command, no need to specify if samtools is in PATH

-d FLT standard deviation cutoff to call discordant read pairs, default 3

-Q INT minimum mapping quality for reads to be used, default 0

-h help

Usage example for TCGA genomes:

perl ./scripts/discon.pl -d 5 -Q 10 -i $somaticg -o $somaticg_rp -B $cancer_bam -C $cancer_cluster -I $cancer_isinfo -K $cancer_blacklistrg -S /home/ly55/opt/samtools/

Each event is given an RP tag with 4 numbers in A_B_C_D format.

A: number of full length discordant read pairs

B: number of discordant read pairs from partially mapped reads (clipped reads)

C: number of concordant read pairs at the first breakpoint

D: number of concordant read pairs at the second breakpoint

The number of A+B should be equal to the total number of discordant read pairs given by Meerkat.

This tool can be used to count the number of concordant read pairs of any given genomic location. For example, if there is a non-reference L1 insertion in the genome, we can format the event similar to a del event as following:

tei L1 NA 16 3 8 136006375 136006390 14 BP:NA-NA

With following command:

perl discon.pl -i $inputfile -o $outputfile -B $bam -I $isinfo -S /home/ly55/opt/samtools/

We will get output:

tei L1 NA 16 3 8 136006375 136006390 14 BP:NA-NA RP:16__29_30

The L1 insertion is at chr8:136006375-136006390 with 14bp target site duplication and 16 supporting discordant read pairs. There are 29 concordant read pairs spanning chr8:136006375 and 30 concordant pairs spanning chr8:136006390. Therefore, the allele fraction for this event is about 0.35.

To use this function, give a non-Meerkat event type (e.g. tei) in column 1, give genomic locations in col 6,7,8, the contents of other fields don't matter.

VIII. Reference

Yang, L., L. J. Luquette, N. Gehlenborg, R. Xi, P. S. Haseley, C. Hsieh, C. Zhang, X. Ren, A. Protopopov, L. Chin, R. Kucherlapati, C. Lee, and P. J. Park. 2013. Diverse Mechanisms of Somatic Structural Variations in Human Cancer Genomes. *Cell*. 153: 919-929.

Yang, L., M. Lee, H. Lu, D. Oh, Y. J. Kim, D. Park, G. Park, X. Ren, C. A. Bristow, P. S. Haseley, S. Lee, A. Pantazi, R. Kucherlapati, W. Park, K. Scott, Y. Choi, P. J. Park. 2016. Analysis of somatic genome rearrangements in human cancers by using whole-exome sequencing. *Am J Hum Genet*, 98:843-856.

IX. Contact

Questions and comments are welcome. Before asking us questions, please read the user manual carefully, there are a lot of contents in this manual, your question may already be answered in the manual.

When you contact us, please provide the following:

Are you able to run ./bin/bamreader from commandline?
Are you able to run example.bam?
The output of "ls -l" for run folder.
pre.log file if it's generated.
isinfo file if it's generated.
dre.log file if it's generated.
Error message if there is any.

Authors:
Lixing Yang, PhD
Lovelace Luquette
The Department of Biomedical Informatics, Harvard Medical School
Boston, MA, 02115, USA
Email: lixing_yang@hms.harvard.edu or ylixing@gmail.com