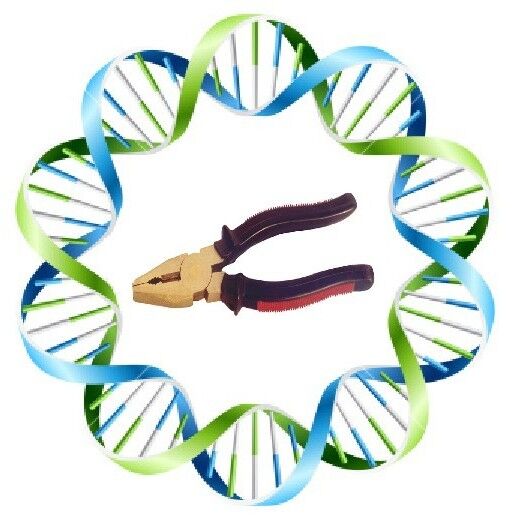
**BamDeal**

****

**V 0.2.3**

**2020-05-12**

**hewm**

|  |
| --- |
| **$ BamDeal convert soap2bam**  **Convert SOAP to SAM/BAM format**  Usage: soap2bam -i <in.soap> -s <out.sam>  Usage: soap2bam -i <in.soap> -b <out.bam> -d Ref.fa  -i <str> input SOAP file  -b <str> output BAM file  -s <str> output SAM file  -d <str> input reference FASTA to get header for BAM  -p if soap is PairOut, for flag  -Q <int> shift sequence quality score by [+31] or [-31] or [0], default [0]  -g all read in memory to search mate information  -h show more details for help [hewm2008 v1.04]   1. soap2bam -i <in.soap> -s AAA   This will convert the SOAP to SAM and output to a compressed file named AAA in current directory.   1. soap2bam -i <in.soap> -b AAA -d Ref.fa   This will convert the SOAP to BAM with the header from reference FASTA and output to a compressed file named AAA in current directory. |
| **$ BamDeal convert bam2soap**  **convert BAM/SAM to SOAP format**  Usage: bam2Soap -i <in.bam> -o <out.soap>  -i <str> input SAM/BAM file  -o <str> output SOAP file  -Q <int> shift sequence quality score by [+31] or [-31] or [0], default [0]  -h show more details for help   1. bam2soap -i <in.bam> -s AAA   This will convert the BAM to SOAP and output to a compressed file named AAA in current directory.   1. bam2sap -i <in.bam> -s AAA -Q -31   This will convert the BAM to SOAP with quality score updated from Solexa to Sanger and output to a compressed file named AAA in current directory. |
| **$ BamDeal convert bam2fq**  **convert BAM to FASTQ format**  Usage: bam2fq -i <in.bam> -o <out.fq>  -i <str> input SAM/BAM file  -o <str> output FASTQ file  -u only output unmapped reads  -h show more details for help   1. Bam2fq -i <in.bam> -o AAA   This will convert the all the BAM to FASTQ and output to a compressed file named AAA in current directory.   1. bam2fq -i <in.bam> -o AAA -u   This will convert the unmapped part of BAM to FASTQ and output to a compressed file named AAA in current directory. If all the reads are mapped, this will give nothing. |
| **$ BamDeal convert bam2fa**  **convert BAM to FASTA format**  Usage: bam2fq -i <in.bam> -o <out.fa>  -i <str> input SAM/BAM file  -o <str> output FASTA file  -u only output unmapped reads  -h show more details for help   1. bam2fa -i <in.bam> -o AAA   This will convert the all the BAM to FASTA and output to a compressed file named AAA in current directory.   1. bam2fa -i <in.bam> -o AAA -u   This will convert the unmapped part of BAM to FASTA and output to a compressed file named AAA in current directory. If all the reads are mapped, this will give nothing. |
| **$ BamDeal modify bamFilter**  **filter low quality read in BAM**  Usage: BamFilter -i <in.bam> -o <out.bam>  -i <str> input SAM/BAM file  -o <str> output BAM file  -q <int> the quality to filter reads, default [15]  -l <int> the length to filter reads, default [30]  -s <int> the beginning of interval containing the 1-based leftmost mapping position of first matching base, default [0]  -e <int> the end of interval containing the 1-based leftmost mapping position of first matching base, default [1e9]  -c <str> specify the chromosome to output, default [all chromosomes]  -d <str> remove the duplicate read  -h show more details for help   1. BamFilter -i <in.bam> -o AAA   This will remove the aligned reads whose quality lower than 15 or length shorter than 30bp and output the reads left to the file named AAA in current directory.   1. BamFilter -i <in.bam> -o AAA -q Q -l L -s S -e E -c ChrX   This will remove the aligned reads whose quality lower than Q or length shorter than Lbp or the 1-based leftmost mapping position of first matching base does not locate within [S,E], and output the reads left of ChrX to the file named AAA in current directory. |
| **$ BamDeal modify bamSplit**  **split the SAM/BAM by chromosome**  Usage: bamSplit -l <bam.list>  Usage: bamSplit -i <A.bam B.bam>  -i <str> input SAM/BAM files, delimited by space  -l <str> Input list of SAM/BAM files  -o <str> output directory, default [PWD]  -s to set the output files in SAM format, default output is in BAM format.  -q <int> reads with quality lower than this would be classified to unmap.bam, default [10]  -r reset output files headers by remove the chromosomes not in the output files  -h show more details for help   1. bmSplit -i A.bam B.bam -q Q   This will split A.bam and B.bam by chromosome and output the splitting BAM files to current directory.   * 1. A.bam and B.bam should have same header.   2. The splitting files would keep the same headers as input files.   3. Reads in A.bam and B.bam with quality lower than Q would be outputted to the file unmap.bam.  1. bamSplit -l bam.list -r -s   This will split the files in bam.list by chromosome and output the splitting SAM files to current directory.  (2.1) files in bam.list should have same header.  For example, if user has two BAM files A.bam and B.bam to split, bam.list should be formatted as:  ./A.bam  ./B.bam  (2.2) with -s added, the output files would be compressed SAM files.  (2.3) with -r added, the header of each output file would keep its own chromosome. |
| **$ BamDeal modify bamAssign**  **assign chromosomes in SAM/BAM to different files**  Usage: bamAssign -i <A.bam B.bam> -a <assign.list> -r  Usage: bamAssign -l <bam.list> -a <assign.list>  -i <str> input SAM/BAM, delimited by space  -l <str> input list of SAM/BAM files  -a <str> list indicating how to assign chromosomes to outputs  -o <str> output directory, default [PWD]  -q <int> reads with quality lower than this would be classified to unmap.bam, default [10]  -r reset output files headers by remove the chromosomes not in the output files  -h show more details for help   1. bamAssign -i <A.bam B.bam> -a <assign.list> -q Q -r   bamAssign -l <bam.list> -a <assign.list> -q Q -r  This will assign the chromosomes in the input SAM/BAM to different files according to the <assign.list> and output the results to current directory.   * 1. -l lists the input files. For example, if user has two input files A.bam and B.bam, bam.list should be formatted as:   ./A.bam  ./B.bam   * 1. Headers of the input SAM/BAMs should be the same.   2. The <assign.list> indicates how to assign chromosomes to outputs.   For example, if there are 5 chromosomes (Chr1, Chr2, Chr3, Chr4, Chr5) in the input, and the <assign.list> is formatted as:  Chr1 AAA  Chr2 AAA  Chr3 BBB  Chr4 BBB  bamAssign would output the reads of Chr1 and Chr2 to the file named AAA and output the reads of Chr3 and Chr4 to the file BBB.   * 1. Chromosomes not in the <assign.list> (like Chr5 in this example) would be outputted to a file named NaAss.bam.   2. Both unmapped reads and reads with quality lower than Q (default value of Q is 10) would be outputted to file named UnMap.bam.   3. With -r added, the header of the output SAM/BAM will keep its own chromosomes. As the example above, header of AAA would contain only Chr1 and Chr2. Without -r, the header of output would be the same as the header of input. |
| **$ BamDeal modify bamCat**  **capture multiple SAM/BAM files into one**  Usage: bamCat -l <.sort.bam.list> -o <out.sort.bam> -s  Usage: bamCat -i <A.bam B.bam> -o C.bam  -i <str> input SAM/BAM file, delimited by space  -l <str> input list of SAM/BAM files  -o <str> output BAM file  -s output sort bam file when all inputs were sorted  -h show more details for help   1. bamCat -i <A.bam B.bam> -o AAA -s   bamCat -i <A.sort.bam B.sort.sam> -o AAA -s  bamCat -l <bam.list> -o AAA  bamCat -l <bam.sort.list> -o AAA -s  This will capture the input files into one file and output the result to a sorted BAM file named AAA in current directory.   * 1. -i input files could be all BAM, all SAM or the mix of two.   2. -l lists the input files. For example, if user has two input files A.bam and B.bam, bam.list should be formatted as:   ./A.bam  ./B.bam   * 1. Input files could have different headers. If the input files have the same chromosome with different length, warning would be given and the longest length would be kept in the header for that chromosome.   2. -s only works when the input files are sorted. Otherwise, the input files would be simply captured together. |
| **$ BamDeal modify bamRand**  **randomly sample SAM/BAM by proportion**  Usage: bamRand -i <in.bam> -o <out.bam>  -i <str> input SAM/BAM file  -o <str> output BAM file  -p <float> probability with which each read would be written into output, default [0.1]  -s <int> random seed, default [time]  -h show more details for help   1. bamRand -i <in.bam> -p X -o AAA   For each read in input SAM/BAM, they would be output into a file named AAA with a probability X (default 0.1) in current directory. |
| **$ BamDeal modify bamSubChr**  **extract or remove chromosome(s) from SAM/BAM**  Usage: bamSubChr -i <in.bam> -d <list> -o <out.bam>  bamSubChr -i <in.bam> -k <list> -o <out.bam>  -i <str> input SAM/BAM file  -o <str> output BAM file  -k <str> list of chromosomes to be kept  -d <str> list of chromosomes to be deleted  -u remove unmapped reads  -r reset output headers by remove the chr(s) not in the out files  -h show more details for help   1. bamSubChr -i <in.bam> -d <delete.list> -o AAA -r   This will remove the chromosome(s) in the delete.list from input SAM/BAM and output the reads left to a file named AAA in current directory. For example, if user would like to remove Chr1 and Chr2 from input BAM file, the delete.list would be formatted as:  Chr1  Chr2  The header of the output SAM/BAM will also remove the name(s) of the chromosome(s) in the delete.list. As the example above, Chr1 and Chr2 would not show up in the header of output file.   1. bamSubChr -i <in.bam> -k <keep.list> -o AAA -r   This will extract the chromosome(s) in the keep.list from input SAM/BAM and output them to a file named AAA in current directory. keep.list is formatted in the same way as delete.list shown above.   1. bamSubChr -i <in.bam> -o AAA -u   This will remove the unmapped reads from input SAM/BAM and output the reads left to a file named AAA in current directory. |
| **$ BamDeal modify bamShiftQ**  **set the Phred quality in the output file**  Usage: BamShiftQ -i <in.bam> -o <out.bam>  -i <str> input SAM/BAM file  -o <str> output BAM file  -p <int> phred quality in output BAM, [1]: ASCII+33 or [2]: ASCII+64, default [1]  -q <int> the quality to filter reads, default [10]  -l <int> the length to filter reads, default [30]  -h show more details for help   1. bamShiftQ -i <in.bam> -o AAA -p 1   This will set the phred quality of output file to ASCII 33 and output the results to a file named AAA in current directory. If the input file is already in ASCII 33 quality system, ShiftQ would ask the user if they still want to continue. |
| **$ BamDeal modify bamLimit**  **split the SAM/BAM by read number**  Usage: bamLimit -i <in.bam> -o ./  -i <str> input SAM/BAM file  -o <str> output directory, default [PWD]  -n <int> max read number for each bam[1000000000]  -h show more details for help   1. bamLimit -i <in.bam> -o ./ -n N   This will split the input BAM file and output the splitting files in current directory. Each output has at most N reads, default value of N is 1000000000. |
| **$ BamDeal statistics Coverage**  **calculate genome coverage/depth/GC**  Usage: Coverage -l <bam.list> -r <Ref.fa> -o <outPrefix>  Usage: Coverage -i <A.bam B.bam> -r <Ref.fa> -o <outPrefix>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -r <str> input reference FASTA to get Depth-GC wig  -w <int> windows size for Depth-GC wig, default [10000]  -b <str> list of the regions of which the coverage and mean of depth would be given  -q <int> the quality to filter reads, default [10]  -d <int> Filter the duplicated read  -h show more details for help   1. Coverage -i <A.bam B.bam> -r <Ref.fa> -o AAA -q Q   Coverage -l <bam.list> -r <Ref.fa> -o AAA -q Q  This will generate four files (GC with depth, depth frequency, depth along reads in reference FASTA and basic statistics) and output the result to current directory. Output files are named with the prefix AAA.  (1.1) the reads with quality lower than Q will be removed from analysis, default value of Q is 10.   1. Coverage -i <A.bam B.bam> -r <Ref.fa> -o AAA -b <bed.region>   Coverage -l <bam.list> -r <Ref.fa> -o AAA -b <bed.region>  This will generate the same outputs as the example above with an extra file giving coverage and mean of depth information for the regions listed in the <bed.region>. The <bed.region> is formatted as [Chr\_name Start\_site End\_site]. For example:  Chr1 1 100  Chr1 1000 2000  Chr2 3 50  … |
| **$ BamDeal statistics BasesCount**  **calculate Genome every Site's four base depth**  Usage：BasesCount -l <bam.list> -o < outPrefix >  Usage：BasesCount -I < A.bam B.bam > -o <outPrefix>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -b <str> list of the regions of which the coverage and mean of depth would be given  -q <int> the quality to filter reads, default [10]  -d <int> Filter the duplicated read  -h show more details for help   1. BasesCount -i <A.bam B.bam> -o AAA -q Q   BasesCount -l <bam,list> -o AAA -q Q  This will generate three files (ATCG four base depth,depth frequency and basic statistics) and output the result to current directory. Output files are named with the prefix AAA.  (1.1) the reads with quality lower than Q will be removed from analysis, default value of Q is 10.   1. BasesCount -i <A.bam B.bam> -o AAA -b <bed.region>   BasesCount -l <bam,list> -o AAA -b <bed.region>  This will generate the same outputs as the example above with an extra file giving  coverage and mean of depth information for the regions listed in the <bed.region>.  The <bed.region> is formatted as [Chr\_name Start\_site End\_site]. For example:  Chr1 1 100  Chr1 1000 2000  Chr2 3 50  … |
| **$ BamDeal statistics DeteCNV**  **detect CNV/deletion region**  Usage: DeteCNV -l <bam.list> -r <Ref.fa> -o <outPrefix>  Usage: DeteCNV -i <A.bam B.bam> -r <Ref.fa> -o <outPrefix>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -r <str> input reference FASTA to get N-base ratio  -f <float> depthRatio to judge breakpoint of merge adjacent[0.45]  -c for each chromosome, use its own mean of depth into calculation, default would use the mean of depth of the whole genome  -m <int> set the minimum length of CNV, default [1800]  -p <float> p-value of CNV depth bias, default [0.02]  -q <int> the quality to filter reads, default [10]  -h show more details for help   1. DeteCNV -i <A.bam B.bam> -r <Ref.fa> -o AAA -q Q -m M   DeteCNV -l <bam.list> -r <Ref.fa> -o AAA -q Q -m M  This will generate two files (raw cnv file and filtered cnv file) and output the result to current directory. Output files are named with the prefix AAA.   * 1. the reads with quality lower than Q will be removed from analysis, default value of Q is 10.   2. it will not be considered a CNV with the minimum length less than M, default value of M is 1800. |
| **$ BamDeal statistics DeteSV**  **detect SV by pair end read**  Usage: DeteSV -l <bam.list> -o <outPrefix>  Usage: DeteSV -I < A.bam B.bam > -o <outPrefix>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -s <int> set the insert value,default[auto]  -d <int> set the insert standard deviation value,default[auto]  -r <str> input reference FASTA to get gap information  -m <int> set the minimum count of paired-end read in one cluster,default[6]  -h show more details for help   1. DeteSV -i <A.bam B.bam> -r <Ref.fa> -m M -o AAA   DeteSV -i <bam.list> -r <Ref.fa> -o -m M AAA  This will generate two files and output the result to current directory. Output files are named with the prefix AAA.   * 1. -m number of paired-end read which support this structure variation,if it is lower than M,this structure variation will be filtered. |
| **$ BamDeal statistics LowDepth**  **detect low depth region**  Usage: LowDepth -l <bam.list> -o <out.bed>  Usage: LowDepth -i < A.bam B.bam > -o <out.bed>  Usage: LowDepth -d <Depth.fa.gz> -o <out.bed>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -d <str> depth along site in reference FASTA  -o <str> output bed region file  -x <int> set the minimum value of low depth,default[2]  -s <int> the length to filter short region, default [1000]  -q <int> ignore too low mapQ read, default [10]  -h show more details for help   1. LowDepth -i <A.bam B.bam> -o <out.bed> -q Q -s S   LowDepth -l <bam.list> -o <out.bed> -q Q -s S  This will generate one compressed file (bed file of low depth region) named out.bed.gz in current directory.   * 1. the reads with quality lower than Q will be removed from analysis, default value of Q is 10.   2. the bed region length lower than S will be filtered.  1. LowDepth -d <Depth.fa.gz> -o <out.bed> -q Q   This operation is the same with the example above but with different input format.  (2.1) the input file <Depth.fa.gz> shows the depth along the reference FASTA. User could get this file with the function Coverage in the Statistics module. Below is an example of the usage of this command. More details could be found with the -h in this command.  bamdeal statistics Coverage -i <in.bam> -o AAA -r <Ref.fa> |
| **$ BamDeal visualize StatQC**  **generate plots for quality control**  Usage: StatQC -i <A.bam B.bam> -o ./  Usage: StatQC -l <bam.list>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> output directory, default [PWD]  -k output the Rscript used to generate plots  -h show more details for help   1. StatQC -i <A.bam B.bam> -k   StatQC -l <bam.list> -k  This will generate four plots (GC with depth, insert size, base distribution and quality distribution) and output the result to current directory. Output files are named with the prefix bamQC.  (1.1) -l lists the input files. For example, if user has two input files A.bam and B.bam, bam.list should be formatted as:  ./A.bam  ./B.bam |
| **$ BamDeal visualize DepthCov**  **generate plots for depth v.s coverage and base proportion**  Usage: DepthCov -i <A.bam B.bam> -o <outprefix>  Usage: DepthCov -d <ref.depthsite.fa.gz> -o <outprefix>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -d <str> depth along site in reference FASTA  -m <int> x-axis of the plot, default [4\*meanDepth]  -q <int> the quality to filter reads, default [10]  -k output the Rscript used to generate plots  -h show more details for help   1. DepthCov -i <A.bam B.bam> -o AAA -k -q Q   DepthCov -l <bam.list> -o AAA -k -q Q  This will generate a plot with sequencing depth in x-axis and two y-axes on different sides (left and right): base proportion and accumulative coverage. The output file will be named with the prefix AAA and output to current directory.  (1.1) -l lists the input files. For example, if user has two input files A.bam and B.bam, bam.list should be formatted as:  ./A.bam  ./B.bam  (1.2) the reads with quality lower than Q will be removed, default value of Q is 10.   1. DepthCov -d <ref.depthsite.fa.gz> -o AAA -k -q Q   This operation is the same with the example above but with different input format.  (2.1) the input file <Depth.fa.gz> shows the depth along the reference FASTA. User could get this file with the function Coverage in the Statistics module. Below is an example of the usage of this command. More details could be found with the -h in this command.  bamdeal statistics Coverage -i <in.bam> -o AAA -r <Ref.fa> |
| **$ BamDeal visualize DepthGC**  **generate plots for depth v.s GC content**  Usage: DepthGC -l <bam.list> -r <Ref.fa> -o <outPrefix>  Usage: DepthGC -f <DepthGC.wig.gz> -o <outPrefix>  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -r <str> input reference FASTA  -f <str> file containing depth and GC content in each window. This file is one of the output files of Bamdeal statistics Coverage.  -w <int> window size to calculate base frequency, default [10000]  -q <int> reads with quality lower than this will be filtered, default [10]  -y <int> maximum of y axis of the plot, default [3\*mean of depth]  -k output the Rscript used to generate plots  -h show more details for help   1. DepthGC -i <A.bam B.bam> -r <Ref.fa> -o AAA -k -q Q   DepthGC -l <bam.list> -r <Ref.fa> -o AAA -k -q Q  This will generate the plot of depth v.s GC content by window. The output file will be named with the prefix AAA and output to current directory.  (1.1) -l lists the input files. For example, if user has two input files A.bam and B.bam, bam.list should be formatted as:  ./A.bam  ./B.bam  (1.2) the reads with quality lower than Q will be removed, default value of Q is 10.   1. DepthGC -f <DepthGC.wig.gz> -o AAA -k -q Q   This operation is the same with the example above but with different input format.  (2.1) the input file <DepthGC.wig.gz> shows depth and GC content by window. User could get this file with the function Coverage in the Statistics module. Below is an example of the usage of this command. More details could be found with the -h in this command.  bamdeal statistics Coverage -i <in.bam> -o AAA -r <Ref.fa> |
| **bamdeal visualize DepthSlide**  Usage: DepthSlide -l <bam.list> -r <Ref.fa> -o <outPrefix>  Usage: DepthSlide -f <DepthGC.wig.gz> -o <outPrefix> -c chr1,chr2  -i <str> input SAM/BAM files, delimited by space  -l <str> input list of SAM/BAM files  -o <str> prefix of output file  -r <str> input reference FASTA  -f <str> file containing depth and GC content in each window. This file is one of the output files of Bamdeal statistics Coverage.  -w <int> window size to calculate base frequency, default [10000]  -s <float> windows sliding ratio (0,1], default [1]  -q <int> reads with quality lower than this will be filtered, default [10]  -c <str> chromosome(s) to draw, delimited by comma. default [all chromosomes]  -y <int> maximum of y axis of the plot, default [4\*mean of depth]  -k output the Rscript used to generate plots  -h show more details for help   1. DepthSlide -i <A.bam B.bam> -r <Ref.fa> -o AAA -k -q Q   DepthSlide -l <bam.list> -r <Ref.fa> -o AAA -k -q Q  This will generate the plot of depth v.s GC content by window. The output file will be named with the prefix AAA and output to current directory.  (1.1) -l lists the input files. For example, if user has two input files A.bam and B.bam, bam.list should be formatted as:  ./A.bam  ./B.bam  (1.2) the reads with quality lower than Q will be removed, default value of Q is 10.   1. DepthGC -f <DepthGC.wig.gz> -o AAA -k -q Q   This operation is the same with the example above but with different input format.  (2.1) the input file <DepthGC.wig.gz> shows depth and GC content by window. User could get this file with the function Coverage in the Statistics module. Below is an example of the usage of this command. More details could be found with the -h in this command.  bamdeal statistics Coverage -i <in.bam> -o AAA -r <Ref.fa>   1. DepthSlide -i <A.bam B.bam> -r <Ref.fa> -o AAA -c Chr1,Chr3 -s 0.2   This will generate the plot of depth v.s GC content of Chr1 and Chr3 by window. The output file will be named with the prefix AAA and output to current directory.  (3.1) with -s set as 0.2, every time it would move forward r\*w (window size) bases to begin a new window. For example, if window size is 10000 by default, the first window would start at site 1, the second window would start at site 2001 (1+10000\*0.2=2001), the third window would start at site 4001 (2001+10000\*0.2=4001), and so on. |