Biokanga Aligner Application Note

Covers release 3.7.6

11th May 2015

Overview

A major task in most NGS dataset dependent bioinformatics workflows is the alignments of the reads against targeted sequences. Such targets may be in the form of a reference fully assembled genome comprising multiple chromosomal sized sequences, partially assembled genomes comprising scaffolded contigs, transcribed gene sequences, or in fact any arbitrary set of sequences having some experimental relevance. The NGS datasets may themselves have been generated using a variety of technological resources with different library preparation protocols for which there will be different optimal best alignment strategies required. The biokanga alignment module exposes a large number of internal thresholds and alignment strategies allowing the experimenter to select the most optimal set of parameters which maximise their chances of meeting or exceeding experimental objectives. It must be noted that there are defaults for most of the parameterisation thresholds which have been chosen empirically and in most cases can deliver alignments which are consistent with experimental objectives.

The 'biokanga align' module can efficiently process up to 100, either single ended or paired end, readsets containing 4 billion reads aligned against a maximum of 100 million target sequences totalling 120Gbp in size. Input readsets can be in fasta, fastq, or as SOLiD colorspace, and there is no need to decompress as biokanga will natively accept gzip'd input readsets. Alignments can be performed in either genomic or RNA transcriptomic space, SNP calling is integrated as are microInDel detection and prioritised region alignments. A range of allowed substitutions can be requested, and a 5' end region differential substitution rate is another option. Output alignment results can be requested to be generated as CSV, BED, SAM or BAM with associated index file. Optional summary statistics can be generated for Phred score distributions, alignment read pair insert sizes, required substitution rates along read lengths, and substitution neighbourhood compositions to name a few.

Biokanga Align Parameterisation

For the 'biokanga align' processing module, the primary inputs are expected to be the NGS readsets, which may specified as either single ended or paired ended, plus the targeted sequences as a pre-indexed suffix array generated with the 'biokanga index' module. The NGS readsets would normally be as raw fastq or SOLiD, but may be provided as pre-filtered multifasta when the experimenter is targeting some specific experimental objective. Although the following list of parameterisation options may seem daunting it is worth noting that most have reasonable defaults which should only require overriding in order to meet specific experimental outcome objectives.

- -m, --mode=<int>
    - This allows the user to trade off sensitivity against alignment throughput. In the less sensitive mode larger anchor core K-mer windows are used and thus reads with more mismatches may not be aligned, in the more sensitive mode smaller anchor core K-mer windows are used resulting in reads with more mismatches becoming alignable but at the cost of reducing processing throughput. It must be remembered that mismatches are not usually uniformly distributed along reads and tend to proximally cluster so the vast majority of reads are alignable even at the lower sensitivities.
    - Four levels of sensitivity are selectable, ranging from low (-m3), default (-m0), more sensitive (-m1), and ultra-sensitive (-m2).
    - The default sensitivity (-m0) is targeting a reasonable balance between sensitivity and throughput for most alignment tasks.

- -Q, --alignstrand=<int>
    - Use to specify the relative sequenced strand
        - 0 sequenced from both sense and antisense, thus alignments expected to be equally distributed between target relative sense and antisense.
        - 1 sense Watson '+', reads were sense strand specific sequenced, alignments only attempted target relative sense
        - 2 antisense Crick '-', reads were antisense strand specific sequenced, alignments only attempted target relative antisense
    - Default (0) is to accept read alignments from either strand

- -a, --microindellen=<int>
    - MicroInDels will be detected and reported with up to the maximum microInDel length as specified by this option. Processing for microInDels will be on those reads which are otherwise unalignable with up to the maximum specified aligner induced substitutions. To be accepted as a microInDel read there must be suport from at least one other read at the same loci also determined as containing a microInDel.
    - The acceptable maximum InDel length can be specified to be in the range of 1 to 20.
    - The default microInDel length is 0 which disables InDel processing and identification.

- -A, --splicejunctlen=<int>
    - RNA-seq reads are likely to include many which span splice junctions. This option allows for the exploration of splice junction spanning reads on those reads which are unalignable using either standard or micoInDel processing. The user specifies the maximum allowed junction separation. Potential junctions are scored on both separation distance and presence of canonical donor/acceptor sites, and the highest scored junction is reported provided there are at least two reads covering that same splice junction.
    - Splice junction separation can be specified to be in the range of 25 to 100000bp.
    - The default is 0 which disables splice junction processing.

- -C, --colorspace
    - When aligning SOLiD colorspace readsets then this parameter option must be explicitly specified.

  - Additionally, the target genome suffix array must have been generated with the ‘biokanga index' module requesting colorspace indexing mode with the '-C' option.
  - By default alignments are in basespace.

- -k, --pcrwin=<int>
  - An identified issue with some NGS readsets is the presence of significant numbers of near identical read copies which stack align non-uniformly to the target. These are observed more commonly in RNA-seq experiments where a common experimental objective is the determination of transcriptomic differential expression. The origin of these stacking reads is likely to be from differential amplification of fragments during the PCR library preparation protocol (PCR artefacts), and the objective of the ‘pcrwin’ threshold processing is to use the number of reads aligning within a specified window bracketing a loci being processed as an indicator of stacking potential at that loci and to adjust the aligned read counts accordingly. All loci in the target with reads aligning are processed for PCR artefact count reductions.
  - The PCR differential amplification artefact reduction window length can be specified to be in the range of 1 to 250.
  - Default is 0 for no PCR differential amplification artefact reduction

- -g, --quality=<int>
  - Although Kanga does not use quality scores as it's alignment discriminant, the distribution of aligner induced substitutions relative to quality scores can be generated in output format 3 ('-M3') provided an output statistics file was also specified with the '-O<file>' option.
  - The '-g' option parameter specifies the input reads fastq quality score type which can be one of the following:
    - 0 for Sanger or Illumina 1.8+
    - 1 for Illumia 1.3+
    - 2 for Solexa < 1.3
    - 3 (default) specifying that quality scores are not of interest and not to be saved
  - Default (3) is not to process any readset associated quality scores

- -r, --mlmode=<int>
  - If the user has requested that reads mapping to multiple loci are to be treated as aligned then this option specifies which of these multiple mapped loci are to be reported as that reads accepted aligned to loci.
  - The user may choose from the following option parameter values:
    - 0 - (default) reads mapping to multiple loci are not accepted as aligned
    - 1 - report only the statistics of multialigned reads, not their loci
    - 2 - request that the accepted reported loci be simply randomly selected from the range of mapped to loci

        - 3 - the accepted loci will be a loci which is overlapped by at least one uniquely aligned read. These overlaps (usually multiple overlaps for any given read) are scored according to both the degree of overlap and the number of uniquely aligned reads overlapped at that loci. The highest scoring overlap loci will be the reported accepted loci. If no overlaps then the read is not accepted as being aligned.
        - 4 - multiple loci mapped reads are proximally clustered, and the cluster loci which is highest scoring is the reported loci accepted for that read. Reads which cluster with no other read will not be accepted as aligned.
        - 5 - output all loci to which all reads aligned including multialigned reads CAUTION: this may result in a huge result file being generated as some reads may align to many thousands of loci.
    - Note: If proximal clustering with other multiple loci mapped reads (-r4) is specified then overlaps with reads which are uniquely aligned will first be explored before the proximal clustering.
    - Default (0) is to report only uniquely aligning reads.

- -R, --maxmulti=<int>
    - A limit can be set on the number of loci any multialigned read may align to, and only those reads aligning within this limit will then be processed according to the mlmode as specified by the '-r' option.
    - The upper limit for multialigned read alignment loci depends on the mlmode-
        - 500 for mlmodes 1 through 4
        - 100000 for mlmode 5
    - Reads aligning to more than the requested maximum number of loci will be not reported as accepted aligned. Note that 'clampmaxmulti' may be used to override, reads will be then reported as if matching the maximum number of loci.
    - The default is for mlmode 0 (reporting only uniques) is 1, and for all other mlmodes it is 5.

- -y, --trim5=<int>
    - Trim this number of bases from 5' end of reads when loading raw reads
    - Up to 50bp may be requested to be trimmed
    - Default is for no 5' trimming
    - Note that if remaining sequence length after end trimming is less than 15bp then that read will be discarded from further processing.

- -Y, --trim3=<int>
    - Trim this number of bases from 3' end of reads when loading raw reads
    - Up to 50bp may be requested to be trimmed
    - Default is for no 3' trimming
    - Note that if remaining sequence length after end trimming is less than 15bp then that read will be discarded from further processing.

- -X, --clampmaxmulti

    - Use to override the default reporting behaviour when processing for multialigned reads where the number of aligned to loci is more than the limit set with ‘maxmulti ‘. If ‘clampmaxmulti’ specified then biokanga will treat reads mapping to more than the limit set with '-R<n>' as if there were exactly <n> matches and report as if accepted aligned
    - The default is not to report as accepted aligned reads multialigned to more the ‘maxmulti’ threshold.

- -b, --bisulfite
    - If the reads were generated as a result of bisulfite library processing protocols then this option must be specified. Additionally the target genome suffix array must have been generated with ‘biokanga index’ requesting bisulfite methylation patterning mode ‘—mode=1’.
    - The default is not to process for bisulfite readsets.

- -e, --editdelta=<int>
    - The user can specify that accepted matches must be at least this Hamming distance from the next best match. Because Biokanga is utilising near exhaustive searches when locating putative alignment loci then it can determine if a putative alignment is unique and the edit distance, or Hamming, to the next best alignment.
    - This may be specified to be either 1 or 2 Hammings
    - If not specified then the default distance is 1.

- -s, --substitutions=<int>
    - This option is used to specify the maximum number of aligner induced substitutions in any accepted alignment. If not specified then a default of 1/10th of the individual read length is used, e.g. if the read length is 75 then the maximum allowed substitutions for that read will be set to 7

- -n, --maxns=<int>
    - Use to specify the maximum number of indeterminate 'N's in reads before treating that read as unalignable. If not specified then will be defaulted to only one indeterminate 'N' allowed.

- -x, --minflankexacts=<int>
    - With current sequencing technologies most errors tend to be towards the read flanks, This option allows the user to request that aligned reads will be 5' and 3' flank trimmed towards the centre of the read until at least this number of exactly matching bases are encountered.
    - The default is for no flank trimming.
    - Note that if a read after flank trimming has been reduced to less than 50% of it's original length then that read is no longer accepted as being aligned. This option is also very useful in RNA-seq processing as it allows for splice site boundaries to be more clearly defined as reads which extend over an exon boundary into the intron are likely to have large numbers of errors in the intron aligned flank, and this flank will be trimmed off.

  - Note that if remaining sequence length after flank trimming is less than 15bp then that read will be discarded from further processing.

- -p, --snpreadsmin=<int>
  - If this option is specified then SNP detection is enabled. The option parameter is used to specify the required minimum read coverage at any loci before processing that loci for SNP determination. The default is for no SNP processing

- -K, --markerlen=<int>
  - Output marker sequences of this length with centralised SNP, output to SNP file name with '.marker' apended (default no markers, range 25..500)

- -G, --markerpolythres=<dbl>
  - Maximum allowed marker sequence base polymorphism independent of centroid SNP (default 0.333, range 0.0 to 0.5)

- -P, --qvalue=<dbl>
  - QValue controlling FDR, using Benjamini-Hochberg, on reported SNPs. Alignment loci are processed for putative SNPs using binominal CDF with an expected error rate set to the maximum of either 0.01 or the observed error rate (number of substitutions/total aligned bases) for the targeted sequence.

- -1, --snpnonrefpcnt=<int>
  - Min percentage of non-ref bases at any putative SNP loci (defaults to 25, range 1 to 35)

- -M, --format=<int>
  - Results can be generated in a number of output formats. This option can be used to specify output as either CSV (with, or without sequences), UCSC BED, or as SAM/BAM. SAM can be specified '-M5' to only include those reads accepted as aligned, or with '-M6' to include all reads. If not specified then SAM output format is used. If SAM output specified and the output alignment file specified with '-o<file' has the extension '.bam', then BAM (compressed with bgzf) and it's associated BAI index file will be generated.

- -t, --title=<string>
  - When generating results in BED format, this option parameter may be used to specify the track title to be written in the resultant BED file.

- -B, --priorityregionfile=<file>
  - Prioritise alignments to loci regions in this BED file
  - The BED file is loaded and the contained loci regions are given priority for all alignments of equal alignment potential in non-prioritised regions.

    - Thus, if a read is aligning to a prioritised region with at most 2 substitutions but that read is aligning elsewhere also with at most 2 substitutions then that read will be processed as if uniquely aligning to the prioritised region.

- -V, --nofiltpriority
    - If priority regions have been specified with the '—priorityRegionfile' parameter then prioritise the alignments, but report all alignments including those to non-prioritised regions.
    - Default in prioritised region processing is to only report alignments into regions which were specified as being priority, and not to report alignments which were elsewhere.

- -N, --bestmatches
    - Accept the best '-R<N>' multiple alignments with alignments having at most '-s<max subs>'
    - Exercise caution – overriding this option parameter may result in excessively large alignment files being generated.
    - If there are more potential alignments than that specified then the 1st N alignments in order of ascending required substitutions will be reported.
    - Default is for the single best only alignment to be accepted.

- -i, --in=<file>
    - Input single end reads, or the 5' PE1 reads if paired end, from these raw sequencer read files
        - wildcards allowed if only single ended, e.g. '-i*.fastq' will be accepted if single end processing but if paired end processing then the 5' PE1 file must be explicitly specified with the corresponding 3' PE2 file specified using the '-u' parameter. Note further that the correspondence between PE1 and PE2 files is implicit in the order in which the PE1 files are specified; the Nth PE1 file specified on the command line is expected to pair with the Nth PE2 file specified.
    - Sequencer read files may be in fasta, fastq, or colorspace if SOLiD, format
    - Sequencer read files may be supplied as compressed, if the extension is '.gz' then the file contents will be automatically decompressed utilising integrated zlib capability.
    - Note that if the read sequence length is less than 15bp then that read, and the corresponding 3' PE2 if paired end, will be discarded from further processing.

- -u, --pair=<file>
    - When processing paired end readsets then specify the 3' PE2 read files with this option parameter.
    - Wildcards in the filename are not allowed.
    - Note that the correspondence between PE1 and PE2 files is implicit in the order in which the PE1 and PE2 files are specified; the Nth PE2 file specified on the command line is expected to pair with the Nth PE1 file specified.
    - Sequencer read files may be in fasta, fastq, or colorspace if SOLiD, format
    - Sequencer read files may be supplied as compressed, if the extension is '.gz' then the file contents will be automatically decompressed utilising integrated zlib capability.

    - Note that if the read sequence length is less than 15bp then that read, and the corresponding 5’ PE1 read, will be discarded from further processing.

- -U, --pemode=<int>
    - Paired end processing mode, user can request that aligned reads which have no globally unique aligned paired read (orphans) be associated with paired reads which are locally unique within the insert size range specified with the -d and -D parameters.
        - U1 - paired end with recover orphan ends
        - U2 - paired end no orphan recovery
        - U3 - paired end with recover orphan ends treating orphan ends as SE
        - U4 - paired end no orphan recovery treating orphan ends as SE
    - There is no default, and this option parameter must be explicitly specified for all paired end read processing.

- -d, --pairminlen=<int>
    - Accept paired end alignments with apparent insert sizes of at least this length
    - Note that in RNA-seq experiments it is very common to process readsets for which this minimum insert size should be set much lower than the expected. If a specific RNA-seq paired end readset has lower than expected alignment rates then try setting the minimum insert size down to just above the read length and examine the insert size length distributions reported with the ‘-O<statsfile>’ option parameter.
    - Defaults to 100bp

- -D, --pairmaxlen=<int>
    - Accept paired end alignments with apparent insert size of at most this length
    - Defaults to 1000bp
    - Note that this may need to be increased if the paired end insert size distribution is long tailed.

- -E, --pairstrand
    - Process paired end reads as if the 5' PE1 and 3' PE2 reads are orientated sense/sense
    - The default is for paired end reads to be aligned with the relative 5’ PE1 sense and 3’ PE2 reads as antisense

- -I, --sfx=<file>
    - This option specifies the suffix array file (must have been pre-generated with ‘biokanga index’) containing the target genome or sequences against which reads are to be aligned.
    - If the '-C' colorspace option or the '-b' bisulphite processing option is specified then Biokanga will check that the suffix array file was appropriately generated for that specific sequence library type.

- -S, --snpfile=<file>

    - Output SNPs (CSV format) to this file (default is to output file name with '.snp' appended)
    - di-SNPs and tri-SNPs will be written to files named with the SNP filename suffixed with ".disnp.csv" and ".trisnp.csv" respectively.

- -c, --minchimeric=<int>
    - Some reads may be chimeric as a result of cross over during the library preparation, most likely the PCR phase. Or alternately there may be partially retained adaptors. This option allows for dynamic trimming from both ends of the read during alignment down to a minimum length.
    - minimum chimeric length as a percentage of probe length (default is 0 to disable, otherwise 50..99)
    - Allowed substitutions will be pro-rata reduced to length actually aligned.

- -o, --out=<file>
    - Primary alignment results will be written to this file which should be named appropriately for the requested output format as specified with the '-M' option parameter. If user has requested microInDel processing then these will be written to a separate results file having the same root name as the primary results file and having the extension '.ind'.
    - If user has requested splice junction processing then these will be written to a separate results file having the same root name as the primary results file and having the extension '.jct'. If user has requested SNP processing then these will be written to a separate results file having the same root name as the primary results file and having the extension '.snp'.
    - If the specified output file has file extension of '.gz' then the file will be generated in gzip compressed format. If SAM output specified (either '-M5' or '-M6') and the output file has the extension '.bam', then BAM (compressed with bgzf) and associated BAI or CSI index will be generated.

- -j, --nonealign=<file>
    - User may specify that all unalignable reads be written to this file for post-processing
    - The format of this generated file will be multifasta with the descriptor line containing a copy of the read identifier as parsed from the original input reads file.
-
- -J, --multialign=<file>
    - User may specify that all reads which align to more than one loci be written to this file
    - The format of this generated file will be multifasta with the descriptor line containing a copy of the read identifier as parsed from the original input reads file.
    -

- -O, --stats=<file>
    - Output aligner induced substitution summary distribution statistics (not supported for '-M6' output mode) or paired end length distributions to this CSV file
        - Substitution summary statistics give the number of substitutions required at each base offset along the length of reads together with the proportion of reads requiring multiple substitutions.
        - Paired end length distributions give the apparent lengths of paired end read insert sizes.
    - Default is for no induced substitution distribution stats or paired end length distributions to be reported

- -L, --siteprefs=<file>
    - output aligned reads start site octamer preferencing to this file
    - If read start sites are randomly sampled then it would be expected that octamer preferencing would have the same distribution as if randomly sampled from the targeted assembly or transcriptomic sequences.
    - Comparing the generated octamer preferencing to the expected preferencing distributions may inform as to potential fragmentation or sequencing biases.

- -l, --siteprefsofs=<int>
    - offset read start sites when processing site octamer preferencing, range -100..100 (default is -4)
    - Any read start site preferencing may be more evidenced up or down stream of the actual read start site.

- -H, --contaminants=<file>
    - Optional input multifasta file containing putative contaminant sequences (perhaps adaptor or primers) which may be overlapping onto the reads. If overlaps are detected then the overlapped reads will be appropriately trimmed to remove the contaminant sequence.
    - Later section in this document describes the format of the contaminates file.

- -Z, --chromexclude=<string>
    - High priority - regular expressions defining chromosomes to exclude
        - These chromosomes are processed for target exclusion at a higher priority than any specified inclusion chromosomes
    - If specified then a given read aligning to an excluded chromosome will not be accepted as aligned to that chromosome
    - The default is for no chromosomes to be excluded

- -z, --chromeinclude=<string>
    - Low priority - regular expressions defining chromosomes to include
        - Inclusion chromosomes are processed for target acceptance after processing for explicit exclusion chromosomes
    - The experimenter can explicitly specify chromosomes for which reads aligning within alignment constraints are to be accepted as aligned
    - The default is for all chromosomes to be included

- -2, --pecircularised
    - Experimental - set true if processing for RNA-seq PE circularised transcript fragments. Normally alignment of a PE paired 3' end is expected to be antisense relative to, and downstream, from the alignment of the corresponding 5' paired end. If a RNA-seq fragment is from a circularised transcript then it would be expected that there will be read pairs in which the paired 3' end is antisense relative to, but now upstream, from the 5' end.
    - This parameterisation option enables processing for alignment acceptance of read pairs which evidence circularisation of the RNA-seq transcript.

- -3, --petranslendist
    - Experimental - include PE length distributions for each transcript in the output summary statistics file.
    - It is observed that it is common for RNA-seq experiments to have paired end alignments for which the insert sizes are much shorter than those expected. If the accepted

- -4, --rptsamseqsthres=<int>
    - Write all SAM reference sequence names to the SAM (BAM) file header if number of reference sequences <= this limit (defaults to 10000). If number of reference sequences above this threshold then only those reference sequences to which there are alignments will be written to the SAM (BAM) file header.

- -5, --lociconstraints=<file>
    - A major problem with alignments to incomplete targets is the reliable characterisation of DE and SNPs. This is because reads are mistakenly aligned to available target sequences even though they may have originated elsewhere, perhaps from homologous sequences, ortholog or paralog, whose sequence is not currently present as a target. However the non-target potential source sequences may have some known polymorphic variations relative to a target sequence even though the complete non-target source sequence is unknown. Biokanga can utilise any known polymorphic variation as a discriminate when aligning reads to a target sequence by filtering out reads which show that variation at specific loci when aligned after applying the usual alignment constraints. Thus if it is known that at loci ChromA.1077 the target reference base is a 'A' and there is likely to be some polymorphic variation in the non-target sequence then all reads potentially accepted as aligned covering the loci ChromA.1077 will be rejected if the base in reads covering that loci are not 'A'. Polymorphic variation elsewhere on ChromA in reads will be accepted. To provide the above functionality, a CSV file is created by the user, and specified to 'biokanga align' with parameter '-5constraints.csv', which is expected to contain four comma separated fields per row:
    - "TargSeq",StartLoci,EndLoci,AcceptBases
        - Whereby -
            - "TargSeq" is the name of the target sequence, must match one of the sequence names in the target.
            - StartLoci is the 0 based offset on the target sequence of the first loci constrained base
            - EndLoci is the 0 bases offset on the target sequence of the last loci constrained base (can be equal to StartLoci for a single base constrained loci or after StartLoci for a constrained loci range)
            - AcceptBases are those bases which will be accepted in reads covering constrained loci. Bases may be explicitly specified as 'A','C','G','T' or to represent the base in the underlying target then as 'R'.
        - A combination may be specified , for example as "AC" in which case only reads with bases 'A' or 'C' within the loci range will be alignment accepted.

    - There is an internal limit of 64 different "TargSeq"s with a maximum total of 6400 loci base constraints accepted for processing.

- -6, --pcrprimersubs=<primersubs>
    - Many readsets have significant numbers of mismatches over the 5' first 12 bases of each read which are independent of their Phred score, and may be caused by hexamer primer binding errors during the sample library preparation. If aligning with very stringent allowed mismatches then these 5' errors may prevent reads from aligning even though the remainder of the read meets the alignment critera. If the <primersubs> is set to be more than 0 then an initial alignment is made allowing a total of <primersubs> + <substitutions>. Subsequent processing then attempts to bring the total number of required subsitutions in the read down to the allowed <substitutions> by iteratively correcting up to <primersubs> mismatches in the 5' 1st 12 bases.
    - The read alignment will only be putatively accepted if the 5' 12bp required no more than 'pcrprimersubs' and the remainder of the read required no more than 'substitutions'. This is a putative alignment only, and will be further processed to check if meeting the 'editdelta' requirements.

- -7, --snpcentroid=<file>
    - Output SNP centroid distributions (CSV format) to this file (default is for no centroid processing)

- -T, --threads=<int>
    - Number of processing threads 0..n (defaults to 0 which sets threads to actual number of CPU cores, max 64)

- -q, --sumrslts=<file>
    - Output results summary to this SQLite3 database file

- -w, --experimentname=<str>
    - Specifies experiment name to use in SQLite3 database file

- -W, --experimentdescr=<str>
    - Specifies experiment description to use in SQLite3 database file

Biokanga Align Internal Processing

The alignment subprocess has many different parameterisation options by which the user can customise both internal thresholding and utilised functionality for delivering optimal experimental outcomes. In the following explanations of the internal processing it is assumed that the experimenter is performing a common alignment task whereby there is a set of 100bp paired end reads which are to be aligned to a reference genome assembly allowing at most 3 mismatches. A later section explains any SNP processing, and a further section covers region prioritorisation capability.

For the outlined simple alignment task scenario the internal processing flow will be –

- Parse and validate the experimenter's thresholding parameterisations and associate appropriate defaults to those option parameters not explicitly specified by the experimenter.
- Initiate on a separate thread the loading and parsing of the regional alignment prioritisation file.
- Initiate on a separate thread the loading of the pre-indexed suffix array file containing the targeted genomic sequences. If an RNA-seq experiment then these could be known transcriptome sequences.
- Initiate on a separate thread the loading and parsing of the paired end readsets.
    - Each related read pair is loaded and parsed as an integral pair of sequences
        - In the outlined scenario, the only checks are for canonical bases, number of indeterminate 'N's, and read sequence length. Other scenarios could have involved trimming etc.
        - If either end of the read pair is unacceptable then both ends of that pair are discarded and not processed further.
        - Reads accepted as parsed are loaded into on-demand dynamically allocated internal structures preserving read descriptors, packed (3 bits per base) associated quality scores (packed into same byte as the base, 5 bits per normalised Phred score) and the spatial relationship between the pair ends.
- Initiate the threads which will be responsible for performing the individual read alignments, these will initially wait for notification that loading of the indexed suffix array has completed, and then iteratively sync/wait on blocks of reads loaded by the readset load/parsing thread to become available for alignment.
    - Each alignment thread will be given at most 4096 reads in a block, and will align these before waiting on the next block of reads to become available. The block size will be progressively reduced so that the processing load can be more uniformly distributed amongst the threads as the remaining reads to be aligned reduces.
    - Alignment threads will independently process each end of the paired end reads –
        - A given read sequence is non-overlap cored, with the size of a core (K) dependent on the read length (RL) and number of substitutions (S) allowed with K = RL / (S + 1)
            - For 100bp reads allowing 3 subs then K = 25
            - Core is used as an anchor, cores exactly matching a subsequence of same K length in the target are flank extended until same length as read is matching, perhaps with substitutions.
        - The size of an anchor core is adjusted according to the requested alignment stringency specified by the experimenter. High stringency reduces the core size, lower stringency increases.
            - Default sensitivity, no change to core size K
        - Number of non-overlapping cores is NC = RL/K
                - NC = 100/4 or 4
            - For each anchor NC cores in the read, an exact match of length K is sought with a binary search against all K-mer length subsequences in the targeted genome
            - Exactly matching target K-mers are then extended, 5' and 3' relative to the anchor core, in the flanking sequences counting the number of substitutions required to continue matching the target sequence.

        - If the total number of required substitutions to match over the full length of the read is less than the current lowest putatively accepted match loci for that read then the just matched loci becomes the lowest putatively accepted match loci.
    - The given read is then reverse complemented, anchor cored, core exact matched and flank extended as per the forgoing when aligning with the original read sense.
    - The finally accepted alignment will be the lowest putatively accepted alignment provided it is at least 'editdelta' better than the next best alignment for that read.
- When reporting alignments in the default SAM or BAM format, the reported alignments will be sorted and ready for post-alignment analytics which accept these formats without requiring any additional intermediate processing steps.

Biokanga SNP Processing

When aligning, Biokanga can also process alignments for single nucleotide polymorphic (SNP) calling using several different experimenter specified thresholds. These are :

- Minimum coverage (number of reads with alignments covering a potential SNP loci)
- Minimum percentage of bases at a putative SNP loci which are non-reference
- Max allowed P-value derived by summing Pr(k=k) accounting for the local sequencing error rate
- Using the Benjamini-Hochberg QValue to rank the called SNPs as a FDR control

The processing flow for SNP detection and acceptance is as follows:

- Aligned reads are stacked in sense orientation (antisense aligned reads are reverse complemented) by ascending alignment loci, so that at any given loci both the coverage and stacked base composition counts can be easily determined.
- For each chromosome a global sequencing error rate (GSER) is calculated as being the total number of read alignment required substitutions (TotChromReadBasesSubs) divided by the total length of all aligned reads (TotChromReadBases) to that chromosome with 0.01 as the floor.
- Each loci along the length of the chromosome is then iterated and the following processing on that loci is executed:
    - If the coverage at the loci currently being processed is less than that specified by the experimenter then that loci is skipped and next loci will be processed.
    - If the proportion of non-reference bases at the loci currently being processed is less than that specified by the experimenter than that loci will be skipped.
    - A local sequencing error rate (LSER) is calculated over a 101bp window bracketing (50bp 5' and 50bp 3' relative to loci being processed for SNP but excluding the putative SNP loci). This LSER is calculated by dividing the total number of read alignment required substitutions in the window by the total length of all aligned bases within the window (excluding bases stacking at the putative SNP loci).

- If the LSER is more than 0.2 then the currently processed loci is skipped (local context too noisy) and next loci will be processed.
- The P-value (1.0 – binomial(n,k,p)) is then calculated for the current loci using the sum of Pr(K = k) as nCk * p^k * q^(n-k) for K = 0 up to K = k where k = number of non-reference bases, n = total bases, and p = LSER.
- If the P-value is above that specified by the user then that loci is skipped and the next loci will be processed.
- Putative SNP loci which meet the forgoing criteria are then deemed as accepted and will be reported

When all SNPs have been accepted for a chromosome, these SNPs are then ranked using Benjamini-Hochberg with the highest rank = 999 and the lowest = 1.

SNP calls are reported in the following CSV format.

Column Header	Meaning	Example
“SNP_ID”	Monotonically increasing unique integer identifier	1234
“ElType”	Element Type – currently always “SNP”	“SNP”
“Species”	Targeted species	“GSS Wheat”
“Chrom”	Name of chrom/contig/sequence on which SNP was identified	“Chr1AL_5678”
“StartLoci”	0 based loci on chrom/contig/sequence on at which SNP was identified	13579
“EndLoci”	Same as StartLoci for SNPs	13579
“Len”	SNP so length always 1	1
“Strand”	SNP is reported relative to sense strand	“+”
“Rank”	Benjamini-Hochberg rank, highest confidence assigned rank 999 and lowest confidence assigned rank 1	679
“PValue”	Probability of SNP as false positive using P-value = 1.0 – binomial(n,k,p)	0.000000
“Bases”	Total number of bases stacking at the SNP loci from all reads covering that loci	2091
“Mismatches”	Of the bases stacking, the number which were not matching the targeted reference sequence	941
“RefBase”	The nucleotide in the reference sequence at the SNP loci	“T”
“MMBaseA”	Number of mismatch stacking A bases	1
“MMBaseC”	Number of mismatch stacking C bases	934
“MMBaseG”	Number of mismatch stacking G bases	6
“MMBaseT”	Number of mismatch stacking T bases - in this example “T” is the reference but still reported as if mismatches	1150
“MMBaseN”	Number of mismatch stacking N bases	0
“BackgroundSubRate”	Local sequencing error rate (LSER)	0.015647
“TotWinBases”	Total number of bases over which LSER was calculated	113054
"TotWinMismatches"	Total number of mismatch bases in LSER	1769
“MarkerID"	Not used in SNP reporting	0

“NumPolymorphicSites”	Not used in SNP reporting	0

Contaminate Sequence File Format

The contaminate sequence file is multifasta and contains those sequences for which the experimenter is interested in counts of the number of times contaminates are observed as overlapping read sequences in NGS readsets. Contaminants may be adaptor sequences or any other artefact sequences, the presence of which may impact on the degree of filtering required in subsequent processing of the effected NGS readsets. Contaminate sequences must be in the range of 4 to 128bp in length.

Ordering priority is determined by the order of contaminate sequences in the contaminate file; earlier occurring sequences have higher priority than later occurring sequences. Higher priority contaminate sequences are processed for putative overlaps on to read sequences before lower priority contaminate sequences with overlap counts accrued to the first contaminate with matching overlap.

The fasta descriptor names are expected to conform to the following convention and will be parsed as such, the convention allows the experimenter to specify the read sequence context for which putative overlaps by that contaminate sequence may be explored. An example fasta contaminate sequence may be:

>contamABC@12

ACGATAGGATTTTTT

The above descriptor line will be parsed and interpreted as meaning that the sequence identified as ‘contamABC’ is only to be processed for sense overlaps onto the 5’ end of a single end read, or the 5’ ends of either end of a paired end pair.

The contaminate sequence naming convention is that sequence names are suffixed with a ‘@” followed by contaminant overlay codes. If not present, then the default is to process for 5’ end sense overlaps only as in the forgoing example.

Contaminate overlay codes are one or more combinations of the following numeric characters and must be prefixed by the ‘@’ character:

Numeric Code	Accepting Putative Overlaps
1	Sense 5’ end of SE, or 5’ PE1 of PE
2	Sense 5’ PE2 of PE
3	Sense 3’ end of SE, or 3’ PE1 of PE
4	Sense 3’ PE2 of PE
5	Antisense 5’ end of SE, or 5’ PE1 of PE
6	Antisense 5’ PE2 of PE
7	Antisense 3’ end of SE, or 3’ PE1 of PE
8	Antisense 3’ PE2 of PE