ngskit4b Reads Assembler Application Note

Release 0.4.6

CAUTION – this document has not been updated to include new functionality added since original cloning of the BioKanga 4.2.0 documentation

Overview

A common bioinformatics workflow task is the assembly of very large numbers, from hundreds of millions through into billions, of NGS reads into a smaller number of maximally contiguous sequences (contigs) with this task referenced as being a de Novo assembly. The targeted de Novo assembly may be either transcriptomic, using RNA-seq readsets, or genomic from DNA readsets. This de Novo assembly task is normally extremely resource intensive, for both CPU and memory, and may require days or weeks of elapsed processing time. The 'kit4b assemb' sub-process is targeting de Novo assembly only, with the scaffolding of the generated contiguous sequences the responsibility of a separate downstream processor task 'kit4b scaffold' or the researcher may use any other scaffolding application of their choice.

Most contemporary de Novo assemblers utilise de Bruijn graphs with K-mer sizes typically ranging from 25 on up to 70 dependent on the targeted assembly type – genomic or transcriptomic – and the read sizes and expected sequencing error rates. K-mer size is commonly explored, and a K-mer is choosen by the researcher which results in maximisation of some assembly metric with the N50 frequently used.

The ngskit4b assembler does not use de Bruijn graphs and instead uses a multiphase iterative maximal sequence overlap paradigm which incorporates the paired end spatial relationship and results in utilisation of effective K-mer overlaps which are several orders of magnitude larger than that used in the de Bruijn approach. As an example, if using 100bp paired end readsets, then during initial overlap extensions the effective K-mer size will default to 150bp (parameterised so researcher can change according to experimental objectives) and will be progressively reduced to a minimum of 55bp (PE extension plus overlaps) and 25bp (SE overlaps) in the final assembly phases. The overall implemented assembly paradigm is to firstly build paired end read extensions from minimal K-mer exactly matching overlaps with other paired end reads, when the PE1 3' end extensions exactly match overlap with the PE2 5' end extensions by another minimal K-mer then to merge the PE1 and PE2 ends into a single high confidence contig. As the assembly proceeds then minimal required K-mer overlaps are progressively reduced, but only when the overwhelming majority of putative overlaps at the longer K-mers have been explored. It is only in the final assembly phases that exploration of substitutions in the overlaps is allowed, all prior phase processing is for only exact matches in the overlaps.

It is important to note that the readsets processed by the 'ngskit4b assemb' subprocess are expected to have been filtered by 'ngskit4b filter', or some other duplicate and error reduction processing application, which should result in the majority of reads being processed by the de Novo assembler being unique and free of sequencing errors. Also note that 'ngskit4b assemb' will generate contigs only, scaffolding of these contigs is the responsibility of a downstream application.

ngskit4b assemb Parameterisation

The 'ngskit4b assemb' processing module is very flexible in terms of the primary input requirements. It can accept a mixture of input types which may contain filtered reads and/or previously assembled contigs or even just a set of single ended sequences the researcher wishes to extend through overlaps.

Any de Novo assembly is generally a trade-off between generating maximally sized high confidence contiguous sequences (contigs) and minimising the compute resources, a mixture of time and hardware, required to achieve experimental objectives. A number of internal thresholds, most having reasonable empirically derived defaults, are exposed through command line parameters by which the researcher can specify optimal parameterisations by which specific experimental objectives can be best realised.

- -m, --mode=<int>
    - Processing mode can be used by the researcher to balance maximally sized high confidence assembled contigs against available compute resources. Both the K-mer reduction steps and maximal processing passes can be independently overridden by other parameterisations – 'reducethressteps' and '—maxpasses''
        - 0 - standard stringency assembly (default)
        - 1 - high stringency assembly
        - 2 - low stringency quick assembly
        - The stringency is defaulting both number of K-mer reduction steps and maximum number of passes or iterations.
            - Number of K-mer reduction steps
                - standard: 5
                - high: 8
                - low: 3
            - Maximum number of processing passes or iterations
                - standard: 50
                - high: 75
                - low: 30

- -t, --trimends=<int>
    - When loading reads or high confidence seed contigs then trim 5' and 3' ends by this many bases (default 0, range 1..50)

- -X,--minseqlen=<int>
    - Only accept reads or high confidence seed contigs, after any trimming, which are of at least this length (default 90, range 70..500)

- -x, --trimpe2se=<int>
    - Trim PEs both 5' and 3' ends by this many bases when treating these as individual SE sequences (default 10, range 0..50) in the later processing phases

- -w, --senseonly
    - Process sequences as strand specific
        - All overlaps are to be sense onto sense
    - Default is to allow sense insensitive overlaps, e.g. antisense onto sense is allowed

- -e, --singleend
    - Process all paired ends (PEs) as if single ended, spatial relationship between the read ends of a paired end readset will not be utilised
    - Default is to utilise the spatial relationship between paired end reads if assembling a paired end readsets

- -p, --maxpasses=<int>
    - Limit number of de Novo assembly processing passes to this maximum, if this maximum number of processing passes is reached then the assembly is assumed to have completed and current contigs and remaining paired end reads will be written to file.
    - The maximum passes can be specified to be in the range of 20 to 10000
    - Default maximum passes is dependent on the processing mode stringency
        - Standard: 50
        - Stringent: 75
        - Low: 30

- -r, --reducethressteps=<int>
    - Reduce required minimal K-mer overlap thresholds over this many steps from their initial minimal down to their final minimal. Increasing the number of step reductions increases the specificity of the overlaps as there is an increased probability of finding maximal K-mer overlaps exceeding the minimal albeit at the cost of increased compute resource times. Reducing the number of step reductions reduces compute resource times but acts to reduce specificity of the K-mer overlaps thus the quality of the assembled contigs may be adversely impacted.
        - Can be specified to be in the range of 2 to 10
    - Default is dependent on the processing mode stringency
        - Standard: 5
        - High: 8
        - Low: 3 quick
        -
- -P, --passthres=<int>
    - de Novo assembly process pass threshold at which to start writing intermediate checkpoint assemblies
        - When pass threshold is reached then any currently assembled contigs plus remaining paired end sequences are written to files as multifasta.
    - Defaults to 0, which disables writing of intermediate checkpoint assemblies
    - If writing checkpoint assemblies then the names of the assemblies consists of the final assembly file prefix name specified with the '—out' parameter which is then appended with the numerical pass at which the check pointing was processed. E.g. if the final assembly name was 'myassembly', the pass threshold at which to start checkpointing was set to be 4, and paired end assembly, then the checkpoint names will start as 'myassembly.Pass4.SE.fasta', 'myassembly.Pass4.PE1.fasta' and 'myassembly.Pass4.PE2.fasta'.
    - Note that there will be additional checkpoint files generated at all intermediate passes starting from the initial pass threshold until the final completed assembly processing

pass. Check that there is sufficient disk space to hold all the intermediate checkpoint assemblies!
    - Note that the checkpoint files are in fasta format and could be used in a subsequent de Novo assembly processing task perhaps with different parameterisation.

- -s, --maxsubs100bp=<int>
    - Allow max induced substitutions per 100bp overlapping sequence fragments
    - Note that any allowed substitution processing will only occur in the final assembly phases when required overlap K-mer thresholds have been reduced to their minimums.
    - Note that whilst specified per 100bp overlap, internally the maximal allowed substitutions are scaled up to per 1Kbp overlap allowing for acceptance of more clustered substitutions provided that over the 1Kbp the total number of substitutions divided by 10 does not exceed the '—maxsubs100bp' rate.
    - Can be specified to be in the range of 0 to 5
    - Defaults to 1 (or 10 per 1Kbp overlap)

- -S, --maxendsubs=<int>
    - Allow max induced substitutions in overlap 12bp ends
    - Note that any allowed substitution processing will only occur in the final assembly phases when required overlap K-mer thresholds have been reduced to their minimums.
    - Intended to compensate for any hexamer mispriming, especially evident with many RNA-seq datasets
    - These overlap end 12bp allowed substitutions are not combined with any allowed substitutions per 100bp overlapping sequence fragments as may be specified with the '—maxsubs100bp' parameterisation.
    - Can be specified to be in the range 0 to 6
    - Defaults to 0 whereby no substitution processing specific to the 12bp fragment ends is performed.

- -j, --initseovlp=<int>
    - Initial minimal SE overlap required to merge two overlapping SE fragment sequences into a single SE fragment sequence.
    - SE sequences will only be merged with other SE sequences if the overlap is at least this threshold during the initial passes. This threshold will be progressively reduced down to the minimum final SE overlap as may be specified (defaults to 25bp) with the '—finseovlp' parameterisation.
    - Can be specified to be in the range of 20 to 500
    - Defaults to 150

- -J, --finseovlp=<int>
    - Final minimal SE fragment sequences overlap required to merge SEs after progressive reductions in threshold have completed.
    - SE sequences will only be merged with other SE sequences if the overlap is at least this threshold during the final passes. Threshold will be progressively reduced down

to this minimal final SE overlap from the initial required overlap as may have been specified (defaults to 150bp) with the '—initseovlp' parameterisation.
    - Can be specified to be in the range of 20 to the '—initseovlp' setting value.
    - Defaults to 25

- -k, --initpeovlp=<int>
    - Initial minimal total sum of both end overlaps required to merge one paired end read pair (PE) with another PE pair into a single PE, extending the 5' end (PE1) and the 3' end (PE2) as may be required during the initial assembly phases.
        - An individual end overlap may be shorter than the other end overlap but the sum of both end overlaps must be at least this threshold.
    - This threshold will be progressively reduced down to the minimum final PE overlap as may be specified (defaults to 35bp) with the '—finpeovlp parameterisation.
    - Merged PEs will be further checked for overlaps of the PE1 3' end extension against the PE2 5' end extension (any overlap must be at least '--minpe2seovlp') before merging the PE1 and PE2 sequences and accepting as a new SE fragment.
    - Can be specified to be in the range of 35 to 200
    - Defaults to 150

- -K, --finpeovlp=<int>
    - Final minimal PE total sum of PE1 and PE2 end overlaps required before two end overlapping PEs will be merged into a single PE.
    - Threshold will be progressively reduced down to this minimal final PE total sum of PE1 and PE2 overlap from the initial required overlap as may have been specified (defaults to 150bp) with the '—initpeovlp parameterisation.
    - Can be specified to be in the range of 35 to the '—initpeovlp' setting
    - Defaults to 35

- -g, --minpe2seovlp=<int>
    - PE1 3' extensions are expected to eventually overlap with the PE2 5' extensions after a number of PE merge operations. This parameterisation allows the researcher to set the minimal required matching overlap of the PE1 3' extension onto the PE2 5' extension for that PE to henceforth be processed as if a SE sequence.
    - Note that if the extensions contain any missincorporated bases - could be the result of sequencing errors, allelic variations or isoforms if a transcriptome assembly – then the extensions may not overlap. Processing rules check for non-overlapping extensions and if end extensions grow beyond internal thresholded sizes will treat PE1 and PE2 as if two independent SE sequence fragments in the final processing steps ('--pe2sesteps').
    - The minimal overlap of PE1 onto PE2 required to merge as SE can be specified to be in the range of 15 to 100np
    - Defaults to 20bp

- -R, --pe2sesteps=<int>
    - When less or equal to this many remaining threshold steps then check the lengths of PE1 3' extensions and PE2 5' extensions, and if either has been extended to be longer

than an internal threshold derived from the expected insert size range then treat the PE1 and PE2 as individual SE sequences.
    - Set to 0 to disable PE1 and PE2 end extension checks
    - Defaults to 2 threshold reduction steps

- -M, --orientatepe=<int>
    - Use to specify the orientation of the paired end reads; individual end sequences may be either sense or antisense dependent on the library preparation protocols:
        - 0 PE1/PE2 is sense/antisense (PE short insert)
        - 1 PE1/PE2 is sense/sense (MP Roche 454)
        - 2 PE1/PE2 is antisense/sense (MP Illumina circularised)
        - 3 PE1/PE2 is antisense/antisense (MP SOLiD)
    - Default is 0 – sense/antisense as is usual with most Illuminia NGS readsets with insert sizes in the range of 100..500bp

- -a, --inpe1=<file>
    - Load PE1 5' paired end readset which is expected to be in fasta or fastq format:
    - Expected to have been pre-processed with 'biokanga filter', or some other external process, to remove most duplicate and error containing reads.
    - If only SE readsets ('—seedcontigsfile') are to be assembled then this parameter is optional.

- -A, --inpe2=<file>
    - Load 3' paired end readset which is expected to be in fasta or fastq format:
        - The PE1 readset must have been specified with '--inpe1'.
    - Expected to have been pre-processed to remove most duplicate and error containing reads.
    - If only SE readsets ('—seedcontigsfile') are to be assembled then this parameter is optional.

- -c, --seedcontigsfile=<file>
    - Load SE readset which must be in fasta or fastq format:
        - Readset could be NGS readset
        - Readset could be high confidence fasta seed contigs or a previously assembled fasta SE sequences file
    - Optional and need not be specified if only PE assembly processing

- -i, --inartreducfile=<file>
    - Load PE or SE readset from a previously generated 'biokanga filter' artefact reduced packed reads file created with the

- -o, --out=<file>
    - Prefix name of files to which assembled contigs and remaining PE sequences are to be written in multifasta format.
    - The output file containing SE assembled contigs will have been named as the prefix name followed by '.SE.fasta'

- If PE assembling, then the output file containing the 5' PE1 sequences (may have been extended) will have been named as the prefix name followed by '.PE1.fasta', whilst the 3' PE2 sequences will be written to output file having a name suffix of '.PE2.fasta'.
  - For example if the researcher used the parameter '—out=assembled' with a PE readset then 3 output files will be created named as 'assembled.SE.fasta', 'assembled.PE1.fasta' and 'assembled.PE2.fasta'.

- -T, --threads=<int>
  - Number of processing threads 0..n
  - Can be specified to be in the range 0 (uses all CPU cores limited to max 64) to 64
  - Defaults to 0 which sets threads to number of CPU cores but will use no more than 64