MIA -- Mapping Iterativ Assembler V 1.0

A tool for creating short read assemblies.

Report bugs to <green@eva.mpg.de>.

==========++++++++++++++==

Usage:

mia -r <reference sequence>

-f <fasta or fastq file of fragments to align>

-s <substitution matrix file> (if not supplied an default matrix is used)

-m <root file name for maln output file(s)> (assembly.maln.iter)

FILTER parameters:

-u fasta database has repeat sequences, keep one based on alignment score

-U fasta database has repeat sequences, keep one based on sum of q-scores

-C<tolerance> collapse sequences with same start, end, strand info into a single sequence

Allow <tolerance> bases difference for start and end coordinates

Important: keep NO SPACE between parameter and value: e.g. -C3

-A use adapter presence and coordinate information to more aggressively

remove repeat sequences - suitable only for 454 sequences that have not

already been adapter trimmed

-T fasta database has adapters, trim these

-a <adapter sequence or code>

-k <use kmer filter with kmers of this length>

-I <filename of list of sequence IDs to use, ignoring all others>

ALIGNMENT parameters:

-p <consensus calling code; default = 1>

-c means reference/assembly is circular

-i iterate assembly until convergence (default)

-n do not iterate assembly until convergence

-F <only output the FINAL assembly, not each iteration>

-D <distantly related reference sequence>

-h give special discount for homopolymer gaps

-M <use lower-case soft-masking of kmers>

-H <do not do dynamic score cutoff, instead use this Hard score cutoff>

-S <slope of length/score cutoff line>

-N <intercept of length/score cutoff line>

The default substitution matrix used the following parameters:

MATCH=200, MISMATCH=-600, N=-100 for all positions

The procedure for removing bad-scoring alignments from the assembly is:

Default: fit a line to length versus score and remove reads that are

less that SCORE_CUTOFF_BUFFER than the average score for its length.

If -H is specified then this hard score cutoff is applied to all reads.

This is preferable if all reads are the same length.

If -S or -N are specified, then these are used as the slope and intercept

of a length/score line. Reads must score above this line to be included.

If only one of -S or -N is specified then the default values are used for

the other (default S = 200.0; default N = 0.0)

The kmer filter requires that a sequence fragment have at least one

kmer of the specified length in common with the reference sequence in

order to align it. For 36nt Solexa data, a value of 12 works well.

The -p option specifies how the new consensus assembly sequence is called

at each iteration:

1 => Any base whose aggregate score is MIN_SC_DIFF_CONS better than all

others is the assembly base. If none is, then N is the assembly base.

2 => The best scoring base whose aggregate score is better than MIN_SCORE_CONS

is the assembly base. If none is, then N is the assembly base.

If -T is specified, mia will attempt to find and trim adapters on each sequence. The adapter sequence itself can be specified by a one letter code as argument to -a. N or n => Neandertal adapter

any other single letter => Standard GS FLX adapter

sequence (less than 127 nt) => user-specified adapter

ma -M <maln input file>

-c <consensus code>

-f <output format>

-R <REGION_START:REGION_END>

-I <ID to assign to assembly sequence>

ma reports information from a maln assembly file as generated by mia

How the assembly calls each base can be determined by the consensus code. 1 = highest, positive aggregate score base (if any)

2 = highest aggregate score base if it is 2400 higher than second highest

The output format can be specified through -f as one of the following. More complete descriptions of these output formats is below, under FORMATS

1 => clustalw

2 => line format; one line each for consensus, reference and coverage

3 => column format; one line per base, one column for consensus, reference, and coverage; includes header with summary info

4 => columns description of all assembly data for positions that differ between consensus and CURRENT reference sequence (see FORMATS, below)

41 => same as above, but for ALL positions

5 => fasta format output of assembled sequence only

6 => show all fragments in a region specified by -R

-C Color format 6 output -> don't pipe this output to file!

7 => ACE

FORMATS (option f):

1 => clustalw

2 => line format; first line is "Consensus, chrM, coverage:"
second line is the entire, assembled, aligned consensus sequence
third line is the entire aligned reference sequence to which the
consensus is aligned
fourth line is the sequence coverage at each position in a space-
separated list of integers

3 => column format; header shows summary statistics; table has one row
per position; columns are described in the output

4 => alternative column format with one row per base that differes between
the consensus assembly and the reference of this iteration.
Note that in the FINAL iteration reference and consensus are equal!
So there won't be any output. Each row has the following
columns: (1)position on reference; 0-based coordinates, (2) reference
base, (3)consensus assembly base, (4)coverage, (5)A's, (6)C's, (7)G's,
(8)T's, (9)gaps; columns 5 through 9 should add up to column 4
(10) aggregate score for A, (11) aggregate score for C
(12) aggregate score for G, (13) aggregate score for T

41=> same as above, but for every position

5 => fasta format using ID "Consensus" for the assembly

6 => region; shows the reference sequence, the consensus sequence, and then
all assembled fragments in a region specified by option -R

61=> same as above, but in multi-fasta format for viewing in Bioedit, e.g.

(also requires a region as specified by the option -R

7 => ACE format

Usage: ccheck [-r <ref.fa>] [-a] [-t] [-s M-N] [-v] <aln.maln>

Reads a maln file and tries to quantify contained contamination.

Options:

-r, --reference FILE FASTA file with the likely contaminant (default: builtin mt311)

-a, --ancient Treat DNA as ancient (i.e. likely deaminated)

-t, --transversions Treat only transversions as diagnostic

-s, --span M-N Look only at range from M to N

-n, --numpos N Require N diagnostic sites in a single read (default: 1)

-f, --force Do not look for a higher numbered .maln

-T, --table Output as tables (easier for scripts, harder on the eyes)

-v, --verbose Increase verbosity level (can be repeated)

-h, --help Print this help message