bowtie program manual

File Name: bowtie program manual.pdf
Size: 1641 KB
Type: PDF, ePub, eBook
Category: Book
Uploaded: 30 May 2019, 18:52 PM
Rating: 4.6/5 from 593 votes.

Status: AVAILABLE

Last checked: 4 Minutes ago!

In order to read or download bowtie program manual ebook, you need to create a FREE account.

Download Now!

eBook includes PDF, ePub and Kindle version

Book Descriptions:

We have made it easy for you to find a PDF Ebooks without any digging. And by having access to our ebooks online or by storing it on your computer, you have convenient answers with bowtie program manual . To get started finding bowtie program manual , you are right to find our website which has a comprehensive collection of manuals listed.
Our library is the biggest of these that have literally hundreds of thousands of different products represented.

Book Descriptions:

bowtie program manual

Nature Methods. 2012, 9357359. Bioinformatics. bty648. Nature Methods. 2012, 9357359. Ultrafast and memoryefficient alignment of short DNA sequences to the human genome. Genome Biology 10R25. It is particularly good at aligning reads of about 50 up to 100s of characters to relatively long e.g. mammalian genomes. Bowtie 2 indexes the genome with an FM Index based on the BurrowsWheeler Transform or BWT to keep its memory footprint small for the human genome, its memory footprint is typically around 3.2 gigabytes of RAM. Bowtie 2 supports gapped, local, and pairedend alignment modes. Multiple processors can be used simultaneously to achieve greater alignment speed. Bowtie 2 is distributed under the GPLv3 license, and it runs on the command line under Windows, Mac OS X and Linux. Bowtie 2 and Bowtie also called " Bowtie 1 " here are also tightly integrated into many other tools, some of which are listed here. Papers describing Bowtie 2 are Fast gappedread alignment with Bowtie 2.Since then, technology has improved both sequencing throughput more nucleotides produced per sequencer per day and read length more nucleotides per read. Number of gaps and gap lengths are not restricted, except by way of the configurable scoring scheme. Bowtie 1 finds just ungapped alignments. Local alignments might be "trimmed" "soft clipped" at one or both extremes in a way that optimizes alignment score. Bowtie 2 also supports endtoend alignment which, like Bowtie 1, requires that the read align entirely. Bowtie 1 does not. In Bowtie 2 all alignments lie along a continuous spectrum of alignment scores where the scoring scheme, similar to NeedlemanWunsch and SmithWaterman. That said, it handles arbitrarily small reference sequences e.g. amplicons and very long reads i.e. upwards of 10s or 100s of kilobases, though it is slower in those settings. It is optimized for the read lengths and error modes yielded by typical Illumina sequencers.http://freshandgleam.com/userfiles/dish-vip-922-manual-pdf.xml

- **bowtie program manual, bow tie program manual pdf, bow tie program manual download, bow tie program manual free, bow tie program manual template.**

With Bioconda installed, you should be able to install Bowtie 2 with conda install bowtie2. If you plan to compile Bowtie 2 yourself, make sure to get the source package, i.e., the filename that ends in "source.zip". It should be possible to build Bowtie 2 on most vanilla Linux installations or on a Mac installation with Xcode installed. But see note about the TBB library below. Bowtie 2 can also be built on Windows using a 64bit MinGW distribution and MSYS. In order to simplify the MinGW setup it might be worth investigating popular MinGW personal builds since these are coming already prepared with most of the toolchains needed. Make sure you're getting the source package; the file downloaded should end in source.zip. Unzip the file, change to the unzipped directory, and build the Bowtie 2 tools by running GNU make usually with the command make, but sometimes with gmake with no arguments. If building with MinGW, run make from the MSYS environment. By default, Bowtie 2 uses the Threading Building Blocks library TBB for this. The table below list some of the commands for a few of the more popular operating systems. Prepackaged builds will include a package that supports SRA. If you're building bowtie2 from source please make sure that the Java runtime is available on your system.This is recommended for most users. To do this, follow your operating system's instructions for adding the directory to your PATH. An "alignment" is a result from this process, specifically an alignment is a way of "lining up" some or all of the characters in the read with some characters from the reference in a way that reveals how they're similar. For example It's not always possible to determine this with certainty. For instance, if the reference genome contains several long stretches of As AAAAAAAAA etc. and the read sequence is a short stretch of As AAAAAAA , we cannot know for certain exactly where in the sea of A s the read

originated.http://www.nextwebdesign.co.uk/pictures/dish-vip211k-owners-manual.xml

That is, it searches for alignments involving all of the read characters. This is also called an “untrimmed” or “unclipped” alignment. In this mode, Bowtie 2 might “trim” or “clip” some read characters from one or both ends of the alignment if doing so maximizes the alignment score. Such an alignment can be produced by Bowtie 2 in either endtoend mode or in local mode. Reference GACTGCGATCTCGACATCG. AlignmentIn this case, 4 characters are omitted or “soft trimmed” or “soft clipped” from the beginning and 3 characters are omitted from the end. This sort of alignment can be produced by Bowtie 2 only in local mode. Reference TAACTTGCGTTAAATCCGCCTGG. AlignmentThe higher the score, the more similar they are. A score is calculated by subtracting penalties for each difference mismatch, gap, etc. and, in local alignment mode, adding bonuses for each match. A length2 read gap receives a penalty of 11 by default 5 for the gap open, 3 for the first extension, 3 for the second extension.A length2 read gap receives a penalty of 11 by default 5 for the gap open, 3 for the first extension, 3 for the second extension.This happens when there are no differences between the read and the reference. The threshold is configurable and is expressed as a function of the read length. This can be configured with the scoremin option. For details on how to set options like scoremin that correspond to functions, see the section on setting function options. For instance, a read that originated inside a repeat element might align equally well to many occurrences of the element throughout the genome, leaving the aligner with no basis for preferring one over the others. Mapping quality is sometimes abbreviated MAPQ, and is recorded in the SAM MAPQ field. The bigger the gap between the best alignment’s score and the secondbest alignment’s score, the more unique the best alignment, and the higher its mapping quality should be.

For instance, a variant caller might choose to ignore evidence from alignments with mapping quality less than, say, 10. A mapping quality of 10 or less indicates that there is at least a 1 in 10 chance that the read truly originated elsewhere. Exactly what expectations hold for a given dataset depends on the lab procedures used to generate the data. For example, a common lab procedure for producing pairs is Illumina’s Pairedend Sequencing Assay, which yields pairs with a relative orientation of FR “forward, reverse” meaning that if mate 1 came from the Watson strand, mate 2 very likely came from the Crick strand and vice versa. Also, this protocol yields pairs where the expected genomic distance from end to end is about 200500 base pairs. Depending on the protocol, these might actually be referred to as “pairedend” or “matepaired.” Also, we always refer to the individual sequences making up the pair as “mates.” The first mate in the file for mate 1 forms a pair with the first mate in the file for mate 2, the second with the second, and so on. When aligning pairs with Bowtie 2, specify the file with the mate 1s mates using the 1 argument and the file with the mate 2s using the 2 argument. This causes Bowtie 2 to take the paired nature of the reads into account when aligning them. The first record describes the alignment for mate 1 and the second record describes the alignment for mate 2. In both records, some of the fields of the SAM record describe various properties of the alignment; for instance, the 7th and 8th fields RNEXT and PNEXT respectively indicate the reference name and position where the other mate aligned, and the 9th field indicates the inferred length of the DNA fragment from which the two mates were sequenced. See the SAM specification for more details regarding these fields. If both mates have unique alignments, but the alignments do not match pairedend expectations i.e.

the mates aren’t in the expected relative orientation, or aren’t within the expected distance range, or both, the pair is said to align “discordantly”. Discordant alignments may be of particular interest, for instance, when seeking structural variants. The expected range of intermates distances as measured from the furthest extremes of the mates; also called “outer distance” is set with the I and X options. Note that setting I and X far apart makes Bowtie 2 slower. See documentation for I and X. This is a conservative threshold, but this is often desirable when seeking structural variants. This is called “mixed mode.” To disable mixed mode, set the nomixed option. The first least significant bit 1

in decimal, 0x1 in hexadecimal is set if the read is part of a pair. The second bit 2 in decimal, 0x2 in hexadecimal is set if the read is part of a pair that aligned in a pairedend fashion. The fourth bit 8 in decimal, 0x8 in hexadecimal is set if the read is part of a pair and the other mate in the pair had at least one valid alignment. The sixth bit 32 in decimal, 0x20 in hexadecimal is set if the read is part of a pair and the other mate in the pair aligned to the Crick strand or, equivalently, if the reverse complement of the other mate aligned to the Watson strand. The seventh bit 64 in decimal, 0x40 in hexadecimal is set if the read is mate 1 in a pair. The eighth bit 128 in decimal, 0x80 in hexadecimal is set if the read is mate 2 in a pair. See the SAM specification for a more detailed description of the FLAGS field. A SAM optional field is formatted like this “XPi1” where “XP” is the TAG, “i” is the TYPE “integer” in this case, and “1” is the VALUE. See the SAM specification for details regarding SAM optional fields. Consider this example Mate 2 TGTTTGGGGTGACACATTACGCGTCTTTGAC. Reference GCAGATTATATGAGTCAGCTACGATATTGTTTGGGGTGACACATTACGCGTCTTTGAC Mate 2 TGTTTGGGGTGACACATTACGC. Reference GCAGATTATATGAGTCAGCTACGATATTGTTTGGGGTGACACATTACGCGTCTTTGAC.

Mate 1 CAGCTACGATATTGTTTGGGGTGACACATTACGC. Mate 2 CTACGATATTGTTTGGGGTGAC. Reference GCAGATTATATGAGTCAGCTACGATATTGTTTGGGGTGACACATTACGCGTCTTTGAC Mate 2 TATGAGTCAGCTACGATATTGTTTGGGGTGACACAT. Reference GCAGATTATATGAGTCAGCTACGATATTGTTTGGGGTGACACATTACGCGTCTTTGAC Bowtie 2’s default behavior is to consider overlapping and containing as being consistent with concordant alignment. By default, dovetailing is considered inconsistent with concordant alignment. Setting nooverlap causes Bowtie 2 to consider overlapping mates as nonconcordant. Setting nocontain causes Bowtie 2 to consider cases where one mate alignment contains the other as nonconcordant. Setting dovetail causes Bowtie 2 to consider cases where the mate alignments dovetail as concordant. Bowtie 2 has three distinct reporting modes. The default reporting mode is similar to the default reporting mode of many other read alignment tools, including BWA. It is also similar to Bowtie 1’s M alignment mode. When we say that a read has multiple alignments, we mean that it has multiple alignments that are valid and distinct from one another. Specifically, we say that two alignments are distinct if there are no alignment positions where a particular read offset is aligned opposite a particular reference offset in both alignments with the same orientation. E.g. if the first alignment is in the forward orientation and aligns the read character at read offset 10 to the reference character at chromosome 3, offset 3,445,245, and the second alignment is also in the forward orientation and also aligns the read character at read offset 10 to the reference character at chromosome 3, offset 3,445,245, they are not distinct alignments. When it finds a valid alignment, it generally will continue to look for alignments that are nearly as good or better. It will eventually stop looking, either because it exceeded a limit placed on search effort see D and R or because it already knows all it needs to know to report an alignment.

Information from the best alignments are used to estimate mapping quality the MAPQ SAM field and to set SAM optional fields, such as ASi and XSi. Bowtie 2 does not guarantee that the alignment reported is the best possible in terms of alignment score. Increasing D makes Bowtie 2 slower, but increases the likelihood that it will report the correct alignment for a read that aligns many places. Increasing R makes Bowtie 2 slower, but increases the likelihood that it will report the correct alignment for a read that aligns many places. That is, if k 2 is specified, Bowtie 2 will search for at most 2 distinct alignments. It reports all alignments found, in descending order by alignment score. The alignment score for a pairedend alignment equals the sum of the alignment scores of the individual mates. Supplementary alignments will also be assigned a MAPQ of 255. See the SAM specification for details. Still, this mode can be effective and fast in situations where the user cares more about whether a read aligns or aligns a certain number of times than where exactly it originated. Alignments are reported in descending order by alignment score. Supplementary alignments will be assigned a MAPQ of 255. See the SAM specification for details. Bowtie 2 is not!

For very large genomes, this mode is very slow. For example, if Bowtie 2 discovers a set of 3 equallygood alignments and wants to decide which to report, it picks a pseudorandom integer 0, 1 or 2 and reports the corresponding alignment. Arbitrary choices can crop up at various points during alignment. If you run the same version of Bowtie 2 on two reads with identical names, nucleotide strings, and quality strings, and if seed is set the same for both runs, Bowtie 2 will produce the same output; i.e., it will align the read to the same place, even if there are multiple equally good alignments. This is intuitive and desirable in most cases. Most users expect Bowtie to produce the same output when run twice on the same input.

When this is specified, Bowtie 2 might report different alignments for identical reads. This is counterintuitive for some users, but might be more appropriate in situations where the input consists of many identical reads. This is “multiseed alignment” and it is similar to what Bowtie 1 does, except Bowtie 1 attempts to align the entire read this way. For instance, it is possible for a read to have a valid overall alignment but to have no valid seed alignments because each potential seed alignment is interrupted by too many mismatches or gaps. You can adjust these options onebyone, though Bowtie 2 comes with some useful combinations of options prepackaged as “ preset options.” This step accounts for the bulk of Bowtie 2’s memory footprint, as the FM Index itself is typically the largest data structure used. For instance, the memory footprint of the FM Index for the human genome is about 3.2 gigabytes of RAM. Bowtie 2 considers all ambiguous characters in the reference including IUPAC nucleotide codes to be Ns. An alignment position that contains an ambiguous character in the read, reference, or both, is penalized according to np. nceil sets an upper limit on the number of positions that may contain ambiguous reference characters in a valid alignment. The optional field XNi reports the number of ambiguous reference characters overlapped by an alignment. For an alignment overlapping an ambiguous reference character to be found, it must have one or more seed alignments that do not overlap ambiguous reference characters. See the documentation for the preset options for details. Bowtie 2 will still print a SAM record for such a read, but no alignment will be reported and the YFi SAM optional field will be set to indicate the reason the read was filtered. This only happens when the input is in Illumina’s QSEQ format i.e. when qseq is specified and the last 11th field of the read’s QSEQ record contains 1.

These messages are printed to the “standard error” “stderr” filehandle. For datasets consisting of unpaired reads, the summary might look like this The wrappers shield users from having to distinguish between “small” and “large” index formats, discussed briefly in the following section. Also, the bowtie2 wrapper provides some key functionality, like the ability to handle compressed inputs, and the functionality for un, al and related options. For genomes less than about 4 billion nucleotides in length, bowtie2build builds a “small” index using 32bit numbers in various parts of the index. When the genome is longer, bowtie2build builds a “large” index using 64bit numbers. Small indexes are stored in files with the.bt2 extension, and large indexes are stored in files with the.bt2l extension. The user need not worry about whether a particular index is small or large; the wrapper scripts will automatically build and use the appropriate index. A denser SA sample yields a larger index, but is also particularly effective at speeding up alignment when many alignments are reported per read. This decreases the memory footprint of the index. In these cases the user specifies three parameters a a function type F, b a constant term B, and c a coefficient A. The available function types are constant C , linear L , squareroot S , and natural log G . The parameters are specified as F,B,A that is, the function type, the constant term, and the coefficient are separated by commas with no whitespace.For example, in the case if the scoremin option, the function fx sets the minimum alignment score necessary for an alignment to be considered valid, and x is the read length. If is specified, bowtie2 will read the mate 1s from the “standard in” or “stdin” filehandle. If is specified, bowtie2 will read the mate 2s from the “standard in” or “stdin” filehandle. If is specified, bowtie2 gets the reads from the “standard in” or “stdin” filehandle.

If the accession provided cannot be found in local storage it will be fetched from the NCBI database.See Obtaining Bowtie 2 for details. The alignpairedreads and preservetags options affect the way Bowtie 2 processes records. By default, alignments are written to the “standard out” or “stdout” filehandle i.e. the console. FASTQ files usually have extension.fq or.fastq. FASTQ is the default format. See also solexaquals and intquals. An input file can be a mix of unpaired and pairedend reads and Bowtie 2 recognizes each according to the number of fields, handling each as it should. FASTA files usually have extension.fa,.fasta,.mfa,.fna or similar. FASTA files do not have a way of specifying quality values, so when f is set, the result is as if ignorequals is also set. When r is set, the result is as if ignorequals is also set. Each kmer is aligned as a separate read. Quality values are set to all Is 40 on Phred scale. Only single kmers, i.e. unpaired reads, can be aligned in this way.There is no way to specify read names or qualities, so c also implies ignorequals. Bases will be trimmed from either the 3’ right or 5’ left end of the read.This scheme was used in older Illumina GA Pipeline versions prior to 1.3. Default off. Can be set to 0 or 1. Setting this higher makes alignment slower often much slower but increases sensitivity. Default 0. Smaller values make alignment slower but more sensitive. Default the sensitive preset is used by default, which sets L to 22 and 20 in endtoend mode and in local mode. For instance, if the read has 30 characters, and seed length is 10, and the seed interval is 6, the seeds extracted will be Seed 1 fw TAGCTACGCT. Seed 1 rc AGCGTAGCTA. Seed 2 fw CGCTCTACGC. Seed 2 rc GCGTAGAGCG. Seed 3 fw ACGCTATCAT. Seed 3 rc ATGATAGCGT. Seed 4 fw TCATGCATAA. Seed 4 rc TTATGCATGA See also setting function options. If the function returns a result less than 1, it is rounded up to 1.

Default the sensitive preset is used by default, which sets i to S,1,1.15 in endtoend mode to i S,1,0.75 in local mode. See also setting function options. Reads exceeding this ceiling are filtered out. Default L,0,0.15. Default 15. This is also the default behavior when the input doesn’t specify quality values e.g. in f, r, or c modes. If norc is specified, bowtie2 will not attempt to align unpaired reads against the reversecomplement Crick reference strand. In pairedend mode, nofw and norc pertain to the fragments; i.e. specifying nofw causes bowtie2 to explore only those pairedend configurations corresponding to fragments from the reversecomplement Crick strand. Default both strands enabled. Such alignments can be found very quickly, and many short read alignments have exact or nearexact endtoend alignments. However, this can lead to unexpected alignments when the user also sets options governing the multiseed heuristic, like L and N. For instance, if the user specifies N 0 and L equal to the length of the read, the user will be surprised to find 1mismatch alignments reported. This option prevents Bowtie 2 from searching for 1mismatch endtoend alignments before using the multiseed heuristic, which leads to the expected behavior when combined with options such as L and N. This comes at the expense of speed. The match bonus ma always equals 0 in this mode, so all alignment scores are less than or equal to 0, and the greatest possible alignment score is 0. This is mutually exclusive with local. endtoend is the default mode. Rather, some characters may be omitted “soft clipped” from the ends in order to achieve the greatest possible alignment score. The match bonus ma is used in this mode, and the best possible alignment score is equal to the match bonus ma times the length of the read. Specifying local and one of the presets e.g. local veryfast is equivalent to specifying the local version of the preset veryfastlocal . This is mutually exclusive with endtoend.

endtoend is the default mode. In local mode is added to the alignment score for each position where a read character aligns to a reference character and the characters match. Not used in endtoend mode. Default 2. A number less than or equal to MX and greater than or equal to MN is subtracted from the alignment score for each position where a read character aligns to a reference character, the characters do not match, and neither is an N. If ignorequals is specified, the number subtracted quals MX.Default 5, 3. Default 5, 3. This is a function of read length. See also setting function options. The default in endtoend mode is L,0.6,0.6 and the default in local mode is G,20,8. When it finds a valid alignment, it continues looking for alignments that are nearly as good or better. The

best alignment found is reported randomly selected from among best if tied. Information about the best alignments is used to estimate mapping quality and to set SAM optional fields, such as ASi and XSi. Instead, it searches for at most distinct, valid alignments for each read. The search terminates when it can't find more distinct valid alignments, or when it finds, whichever happens first. All alignments found are reported in descending order by alignment score. For reads that have more than distinct, valid alignments, bowtie2 does not guarantee that the alignments reported are the best possible in terms of alignment score. k is mutually exclusive with a. A seed extension "fails" if it does not yield a new best or a new secondbest alignment. This limit is automatically adjusted up when k or a are specified. Default 15. When "reseeding," Bowtie 2 simply chooses a new set of reads same length, same number of mismatches allowed at different offsets and searches for more alignments. A read is considered to have repetitive seeds if the total number of seed hits divided by the number of seeds that aligned at least once is greater than 300. Default 2. A 19bp gap would not be valid in that case.

If trimming options 3 or 5 are also used, the I constraint is applied with respect to the untrimmed mates. This is because larger differences between I and X require that Bowtie 2 scan a larger window to determine if a concordant alignment exists. For typical fragment length ranges 200 to 400 nucleotides, Bowtie 2 is very efficient. A 61bp gap would not be valid in that case. If trimming options 3 or 5 are also used, the X constraint is applied with respect to the untrimmed mates, not the trimmed mates. This is because larger differences between I and X require that Bowtie 2 scan a larger window to determine if a concordant alignment exists. For typical fragment length ranges 200 to 400 nucleotides, Bowtie 2 is very efficient. Default fr appropriate for Illumina's Pairedend Sequencing Assay. This option disables that behavior. This option disables that behavior. See also Mates can overlap, contain or dovetail each other. Default mates cannot dovetail in a concordant alignment. See also Mates can overlap, contain or dovetail each other. Default a mate can contain the other in a concordant alignment. See also Mates can overlap, contain or dovetail each other. Default mates can overlap in a concordant alignment. Use this option to align pairedend reads instead. This is printed to the "standard error" "stderr" filehandle. Default off. These reads correspond to the SAM records with the FLAGS 0x4 bit set and neither the 0x40 nor 0x80 bits set. If ungz is specified, output will be gzip compressed. If unbz2 or unlz4 is specified, output will be bzip2 or lz4 compressed. Reads written in this way will appear exactly as they did in the input file, without any modification same sequence, same name, same quality string, same quality encoding. Reads will not necessarily appear in the same order as they did in the input. These reads correspond to the SAM records with the FLAGS 0x4, 0x40, and 0x80 bits unset. If algz is specified, output will be gzip compressed.

If albz2 is specified, output will be bzip2 compressed. Similarly if allz4 is specified, output will be lz4 compressed. Reads will not necessarily appear in the same order as they did in the input. Otherwise,.1 or.2 are added before the final dot in to make the permate filenames. Reads written in this way will appear exactly as they did in the input files, without any modification same sequence, same name, same quality string, same quality encoding. Reads will not necessarily appear in the same order as they did in the inputs. Otherwise,.1 or.2 are added before the final dot in to make the permate filenames. Reads will not necessarily appear in the same order as they did in the inputs. Having alignment metric can be useful for debugging certain problems, especially performance issues. See also met. Default metrics disabled. This is not mutually exclusive with metfile. Having alignment metric can be useful for debugging certain problems, especially performance issues. See also met. Default metrics disabled. Only matters if either metstderr or metfile are specified. Default 1. It also causes the RGZ extra field to be attached to each SAM output record, with value set to. Specify rg multiple times to set multiple fields. See the SAM Spec for details about what fields are legal. Specifying this option causes Bowtie 2 to print an asterisk in those fields instead. Only available in local mode. If is greater than the offrate used to build the index, then some row

markings are discarded when the index is read into memory. This reduces the memory footprint of the aligner but requires more time to calculate text offsets.Searching for alignments is highly parallel, and speedup is close to linear. Increasing p increases Bowtie 2’s memory footprint. E.g. when aligning to a human genome index, increasing p from 1 to 8 increases the memory footprint by a few hundred megabytes.Has no effect if p is set to 1, since output order will naturally correspond to input order in that case.

Only has an effect when read format is qseq. Default off. It seeds the generator with a number derived from a the read name, b the nucleotide sequence, c the quality sequence, d the value of the seed option. This means that if two reads are identical same name, same nucleotides, same qualities Bowtie 2 will find and report the same alignments for both, even if there was ambiguity. When nondeterministic is specified, Bowtie 2 reinitializes its pseudorandom generator for each read using the current time. This means that Bowtie 2 will not necessarily report the same alignment for two identical reads. This is counterintuitive for some users, but might be more appropriate in situations where the input consists of many identical reads. For more details, see the SAM format specification. Each line is a collection of at least 12 fields separated by tabs; from left to right, the fields are If the read name contains any whitespace characters, Bowtie 2 will truncate the name at the first whitespace character. This is similar to the behavior of other tools. The standard behavior of truncating at the first whitespace can be suppressed with samnoqnametrunc at the expense of generating nonstandard SAM. Flags relevant to Bowtie are Offset is 0 if there is no mate. Size is negative if the mate’s alignment occurs upstream of this alignment. Size is 0 if the mates did not align concordantly. However, size is non0 if the mates aligned discordantly to the same chromosome. The encoded quality values are on the Phred quality scale and the encoding is ASCIIoffset by 33 ASCII char ! , similarly to a FASTQ file. Only present if SAM record is for an aligned read. Can be negative. Can be greater than 0 in local mode but not in endtoend mode. Only present if the SAM record is for an aligned read and more than one alignment was found for the read. Note that, when the read is part of a concordantlyaligned pair, this score could be greater than ASi.