BED Format Specification

Jeffrey Niu, Danielle Denisko, Michael Hoffman

August 6, 2019

1 The BED Format Specification

BED (Browser Extensible Data) is a whitespace-delimited file format consisting of one or more BED lines. The BED lines describe features of a genome in terms of a starting and ending position. The file extension for the BED format is `.bed`

The BED format encodes all its fields using 7-bit US ASCII. The regular expressions in this document follow POSIX / IEEE 1003.1 extended syntax.

1.1 Terminologies and Concepts

0-start, half-open coordinate system: A coordinate system where the first base starts at position 0 and the start of the interval is included but the end is not. For example, for a sequence of bases ACTGCG, the bases given by the interval [2, 4) are TG.

BED3, BED4, ..., BED12: Designates a file that has the first # fields of the BED format. e.g. BED3 means a file with only the first three fields; BED12 means a file with all 12 fields.

Feature: A sequence of bases on a chromosome that are significant. For example, it could be peaks that are discovered during peak calling.

Block: Another term for exon, a coding region of the strand.

1.2 UCSC Track Files

Track files are files that can be viewed with the UCSC Genome Browser.[1] Track files contain browser lines and track lines (explained here) that precede lines that are of one of the following formats: bedGraph, GTF, PSL, BED, bigBed, WIG, bigGenePred, bigNarrowPeak, bigMaf, bigChain, bigPsl, barChart, bigBarChart, interact, bigInteract, bigWig, BAM, CRAM, VCF, MAF, BED detail, Personal Genome SNP, broadPeak, narrowPeak, and microarray.
Track files are not valid BED files. This means that the BED format must not have any browser or track lines.

[1] Haeussler, Maximilian et al. "The UCSC Genome Browser database: 2019 update." Nucleic acids research vol. 47,D1 (2019): D853-D858. doi:10.1093/nar/gky1095

1.3 Comments and Blank Lines

Comments can be inserted by placing a # before the comment's text. Comments must appear on their own lines. A comment line can be anywhere and in any quantity.

Blank lines are also permitted and can appear anywhere and in any quantity.

1.4 Examples

An example from the UCSC Genome Browser FAQ[2] except the track/browser lines are removed:

```
chr7 127471196  127472363  Pos1  0  +
chr7 127472363  127473530  Pos2  0  +
chr7 127473530  127474697  Pos3  0  +
chr7 127474697  127475864  Pos4  0  +
chr7 127475864  127477031  Neg1  0  -
chr7 127477031  127478198  Neg2  0  -
chr7 127478198  127479365  Neg3  0  -
chr7 127479365  127480532  Pos5  0  +
chr7 127480532  127481699  Neg4  0  -
```

Another example from the UCSC Genome Browser FAQ that uses all 12 BED fields:

```
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

1.5 BED fields

There are 12 whitespace-delimited fields. The first three fields are mandatory, and the last nine fields are optional. In the optional fields, the order is binding; if one field is filled, then all previous fields must also be filled. However, BED10 is not allowed.[3]

In a BED file, each row must have the same number of fields.

The positions in BED fields are all described in the 0-based, half-open coordinate system.

Col	Field	Type	Regexp/Range	Brief Description
1	chrom	String	`\w+`[4]	Name of the chromosome/scaffold
2	chromStart	Int	$[0, 2^{32} - 1]$	Starting position of feature
3	chromEnd	Int	$[0, 2^{32} - 1]$	Ending position of feature
4	name	String	`[^\s]+`	Name of the data line
5	score	Int	$[0, 1000]$	A numerical value
6	strand	String	`[+-.]`	The feature's strand (plus/minus)
7	thickStart	Int	$[0, 2^{32} - 1]$	Starting position for thick display
8	thickEnd	Int	$[0, 2^{32} - 1]$	Ending position for thick display
9	itemRgb	Int,Int,Int	$[0, 255], [0, 255], [0, 255]$	Color of data when displayed

[2] Frequently Asked Questions: Data File Formats." Genome Browser FAQ, genome.ucsc.edu/FAQ/FAQformat.html

[3] Knowing only the number of blocks has almost no use cases.

[4] Though restrictive, this limitation makes them portable to varying environments.

10	blockCount	Int	$[0,$ chromEnd-chromStart$]$[5]	Number of exons
11	blockSizes	List[Int]	`(\d+,){blockCount-1}\d+,?`[6]	List of exon sizes
12	blockCount	List[Int]	`(\d+,){blockCount-1}\d+,?`	List of exon start positions

1. **chrom**: The name of the chromosome or scaffold. The name that the chromosome or scaffold is given should align with the genome that it is associated with. Additionally, the manner in which chromosomes are named should remain consistent in a file. e.g. There should not be a mix of 17 and chr17 to represent the same chromosome.

2. **chromStart**: The starting position of the feature in the chromosome or scaffold. This value must be an integer greater than or equal to zero and less than the total number of bases of the chromosome to which it belongs. If the size of the chromosome is unknown, then `chromStart` must be less than or equal to $2^{32} - 1$, which is the maximum size of an unsigned integer.

3. **chromEnd**: The ending position of the feature in the chromosome or scaffold. This value must be an integer greater than or equal to the value of `chromStart` and less than or equal to the total number of bases in the chromosome to which it belongs. If the size of the chromosome is unknown, then `chromEnd` must be less than or equal to $2^{32} - 1$, which is the maximum size of an unsigned integer.

4. **name**: A string that describes the feature. The name must be one or more characters without whitespace. The name will be displayed next to the feature in the Genome Browser. There is no default value for name. If the feature does not need a name, there is a recommendation for the default value in Section 2.2.

5. **score**: An integer between 0 and 1000, inclusive. The score field is used by the Genome Browser to shade features where features with higher scores get darker shades. If the feature has no score, then "0" should be used.

6. **strand**: The strand that the feature appears on. The strand can either refer to the + (sense/coding) strand or the - (antisense/complementary) strand. If the feature has no strand information, then a dot "." must be used.

7. **thickStart**: The starting position at which the feature is drawn thickly. This field is for indicating the position where the Genome Browser will display the feature with a thicker line. This value must be an integer between `chromStart` and `chromEnd`, inclusive. There is no specified default value for `thickStart`. Instead, if the feature does not need a thick outline, there is a recommendation for the default value in Section 2.2.

8. **thickEnd**: The ending position at which the feature is drawn thickly. This field is for indicating the position where the Genome Browser will stop displaying the feature with a thicker line. This value must be an integer greater than or equal to `thickStart` and less than or equal to `chromEnd`, inclusive. If this field is not specified but `thickStart` is, then the entire feature will have thick display. There is no specified default value for `thickEnd`. Instead, if the feature does not need a thick outline, there is a recommendation for the default value in Section 2.2.

[5] `chromEnd-chromStart` is the maximum number of blocks that can exist without having overlaps.

[6] `blockCount-1` is meant to be evaluated. e.g. If `blockCount=4`, then the regexp would be `(\d+,){3}\d+,?`

9. **itemRgb**: A RGB value that determines the color that will be displayed with this feature. The RGB value must be in the format of three integers separated by commas, each integer between the values of 0 and 255, inclusive. If the feature does not need to be colored, then a single zero "0" should be used, though any non-negative integer can also be used.

10. **blockCount**: The number of blocks in the data line. In the Genome Browser, the blocks will be thicker than the rest of the sequence. `blockCount` must be an integer greater than zero. There is no default value for `blockCount` because `blockSizes` and `blockStarts` cannot be defined without `blockCount`.

11. **blockSizes**: A comma-separated list of length `blockCount` that contains the size of each block. There must be no spaces before or after the comma(s). There may be a trailing comma after the last element of the list. Each size must be an integer greater than or equal to zero but less than or equal to the total number of bases in the feature. There is no default value for `blockSizes` because `blockStarts` cannot be verified without `blockSizes`.

12. **blockStarts**: A comma-separated list of length `blockCount` containing the positions of where each block starts, relative to `chromStart`. There should be no spaces before or after the comma(s). There may be a trailing comma after the last element of the list. Each index in this list corresponds to the corresponding index in `blockSizes`. Each block start must be an integer between zero and `chromEnd - chromStart`, and `chromStart + blockStart + blockSize` must be less or equal to `chromEnd`. These conditions enforce that the block is fully contained within the feature. The first element in `blockStarts` must be zero, and the last element of `blockStarts` must satisfy `blockStart + blockSize = chromEnd`. The list must be sorted in ascending order. There is no default value for `blockStarts`.

2 Recommended Practice for the BED Format

1. **Mandatory Fields**:

   - `chrom`: The name of each chromosome should match the tool that is being used or match other files being used such as a chromosome sizes file.

2. **Optional Fields**:

   - `name`: A recommended default value for this field is a dot "."
   - `itemRgb`: The UCSC Genome Browser recommends that eight or fewer colors should be used as too many colors may slow down the browser.
   - `thickStart` and `thickEnd`: It is recommended that if there is no need for features to be drawn thickly, then both `thickStart` and `thickEnd` should be set to `chromStart`.

3. **Sorting**: BED files should be sorted by `chrom` alphabetically, then by `chromStart` numerically. e.g. the order that chromosomes should appear should be chr1, chr10, chr11, chr12, ..., chr2, chr20, chr21, ..., chr3, ...

4. **Whitespacing**: Though any kind of whitespacing is allowed as a delimiter for the BED fields, it is recommended that TABs be used because almost all tools support TABs while some tools do not support other kinds of whitespace.

5. **Very large BED files**: For files that are greater than 50 MB in size, it is recommended that the file be converted to bigBed format, which is an indexed binary format. The conversion is done by a program called `bedToBigBed`[7].

3 Custom BED definitions

Custom BED definitions are BED files where the first fields follow the standard BED specification and the last fields are custom for the specific user. These BED files are denoted by BEDx+y where x is the number of fields taken from standard BED (between 3 and 12 inclusive), and y is the number of fields that are custom. There is no limit to the number of custom fields that can be added. e.g. BED6+3 refers to a file with the first six fields of BED (chrom, chromStart, chromEnd, name, score, strand), followed by three fields that are defined by the user.

[7] Kent, W J et al. âĂIJBigWig and BigBed: enabling browsing of large distributed datasets.âĂİ Bioinformatics (Oxford, England) vol. 26,17 (2010): 2204-7. doi:10.1093/bioinformatics/btq351