The Browser Extensible Data (BED) format

Jeffrey Niu, Danielle Denisko, Michael M. Hoffman

May 19, 2021

1 Specification

Browser Extensible Data (BED) is a whitespace-delimited file format, where each **file** consists of one or more **lines**.[1] Each **line** describes discrete genomic **features** by physical start and end position on a linear **chromosome**. The file extension for the BED format is `.bed`.

1.1 Typographic conventions

This document uses the following typographic conventions:

Style	Meaning	Examples
Bold	Terms defined in subsections 1.2–1.3	**chromosome** **file**
Sans serif	Names of **fields**	chrom chromStart chromEnd
Fixed-width	Literals or regexes[2]	`.bed` `grep` `[[:alnum:]]+` `ATCG`

1.2 Terminology 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`.

BEDn: A **file** with the first n **field**s of the BED format. For example, **BED3** means a **file** with only the first 3 **fields**; **BED12** means a **file** with all 12 **field**s.

BEDn+: A **file** that has n **field**s of the BED format, followed by any number of **field**s of custom data defined by a user.

BEDn+m: A **file** that has a custom tab-delimited format starting with the first n **field**s of the BED format, followed by m **field**s of custom data defined by a user. For example, **BED6+4** means a **file** with the first 6 **field**s of the BED format, followed by 4 user-defined **field**s.

block: Linear subfeatures within a **feature**. Usually used to designate exons.

[1] "Frequently Asked Questions: Data File Formats." University of California, Santa Cruz (UCSC) Genome Browser FAQ, `https://genome.ucsc.edu/FAQ/FAQformat.html`

[2] POSIX/IEEE 1003.1–2017 Extended Regular Expressions, for the "C" locale. *IEEE Standard for Information Technology—Portable Operating System Interface (POSIX) Base Specifications*, IEEE 1003.1–2017, 2017

chromosome: A sequence of nucleobases with a name. In this specification, "chromosome" may also describe a named scaffold that does not fit the biological definition of a chromosome. Often, chromosomes are numbered starting from `1`. There are also often sex chromosomes such as `W`, `X`, `Y`, and `Z`, mitochondrial chromosomes such as `M`, and possibly scaffolds from an unknown chromosome, often labeled `Un`. The name of each chromosome is often prefixed with `chr`. Examples of chromosome names include `chr1`, `21`, `chrX`, `chrM`, `chrUn`, `chr19_KI270914v1_alt`, and `chrUn_KI270435v1`.

feature: A linear region of a **chromosome** with specified properties. For example, a **file**'s **feature**s might all be peaks called from ChIP-seq data, or transcript.

field: Data stored as non-tab text. All **field**s are 7-bit US ASCII.

file: Sequence of one or more **line**s.

line: String terminated by a **line separator**, in one of the following classes. Either a **data line**, a **comment line**, or a **blank line**. Discussed more fully in subsection 1.3

line separator: Either carriage return, line feed, or carriage return followed by line feed. The same **line separator** must be used throughout the **file**.

1.3 Lines

1.3.1 Data lines

Data lines contain **feature** information. A data line is composed of **field**s separated by whitespace. The whitespace must match the regex `[[:space:]]+`[3].

1.3.2 Comment lines and blank lines

Both comment lines and blank lines provide no **feature** data.

Comment lines start with `#` with no whitespace beforehand. A `#` appearing anywhere else in a line is treated as **feature** data, not a comment.

Blank lines consist entirely of whitespace. Both comment and blank lines may appear as any line in a **file**, at the beginning, middle, or end of the file. They may appear in any quantity.

1.4 BED fields

Each **data line** contains between 3 and 12 whitespace-delimited **field**s. The first 3 **field**s are mandatory, and the last 9 **field**s are optional. In optional **field**s, the order is binding—if 1 **field** is filled, then all previous **field**s must also be filled. However, **BED10** and **BED11** are prohibited.

In a BED **file**, each **data line** must have the same number of **field**s. The positions in BED **field**s are all described in the **0-based, half-open coordinate system**.

[3] `[[:space:]]` includes the following characters: space, form-feed, newline, carriage-return, tab, and vertical-tab

Col	Field	Type	Regex or range	Brief description
1	chrom	String	`[[:alnum:]_]{1,255}`[4]	**Chromosome** name
2	chromStart	Int	$[0, 2^{64} - 1]$	**Feature** start position
3	chromEnd	Int	$[0, 2^{64} - 1]$	**Feature** end position
4	name	String	`[^\t]{0,255}`	**Feature** description
5	score	Int	$[0, 1000]$	A numerical value
6	strand	String	`[-+.]`	**Feature** strand
7	thickStart	Int	$[0, 2^{64} - 1]$	Thick start position
8	thickEnd	Int	$[0, 2^{64} - 1]$	Thick end position
9	itemRgb	Int,Int,Int	$([0, 255], [0, 255], [0, 255])$ `| 0`	Display color
10	blockCount	Int	$[0, \text{chromEnd} - \text{chromStart}]$[5]	Number of **block**s
11	blockSizes	List[Int]	`([[:digit:]]+,){blockCount−1}[[:digit:]]+,?`[6]	**Block** sizes
12	blockStarts	List[Int]	`([[:digit:]]+,){blockCount−1}[[:digit:]]+,?`	**Block** start positions

1.5 Coordinates

1. chrom: The name of the **chromosome** or scaffold where the **feature** is present. Limiting only to word characters only, instead of all non-whitespace characters, makes BED **file**s more portable to varying environments which may make different assumptions about allowed characters. The name must be between 1 and 255 characters long, inclusive.

2. chromStart: Start position of the **feature** on the **chromosome** or scaffold. chromStart must be an integer greater than or equal to 0 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 $2^{64} - 1$, which is the maximum size of an unsigned 64-bit integer.

3. chromEnd: End position of the **feature** on the **chromosome** or scaffold. chromEnd 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 $2^{64} - 1$, the maximum size of an unsigned 64-bit integer.

1.6 Simple attributes

4. name: String that describes the **feature**. The name must be 0 to 255 non-tab characters. The name must not be empty or contain whitespace, unless all fields in file are delimited exclusively using single tab characters. A visual representation of the BED format may display the name next to the **feature**.

5. score: Integer between 0 and 1000, inclusive. If the **feature** has no score information, then `0` should be used as a default value. A visual representation of the BED format may shade features differently depending on their score.

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

[5] `[[:alnum:]_]` is equivalent to the regex `[A-Za-z0-9_]`. It is also equivalent to the Perl extension `[[:word:]]`.

[6] chromEnd-chromStart is the maximum number of **block**s that may exist without overlaps.

[6] For example, if blockCount = 4, then the allowed regex would be `([[:digit:]]+,){3}[[:digit:]]+,?`

1.7 Display attributes

7. thickStart: Start position at which the **feature** is visualized with a thicker or accented display. This value must be an integer between chromStart and chromEnd, inclusive. There is no specified default value for thickStart.

8. thickEnd: End position at which the **feature** is visualized with a thicker or accented display. This value must be an integer greater than or equal to thickStart and less than or equal to chromEnd, inclusive. In BED files with fewer than 7 **field**s, the whole **feature** has thick display. In **BED7+** files, to achieve the same effect, set thickStart equal to chromStart and thickEnd equal to chromEnd. If this **field** is not specified but thickStart is, then the entire **feature** has thick display. There is no specified default value for thickEnd.

9. itemRgb: A triple of integers that determines the color of this **feature** when visualized. The triple is three integers separated by commas. Each integer is between 0 and 255, inclusive. To make a **feature** black, itemRgb may be a single `0`, which is visualized identically to a **feature** with itemRgb of `0,0,0`.

1.8 Blocks

10. blockCount: Number of **block**s in the **feature**. blockCount must be an integer greater than 0. blockCount is mandatory in **BED12+** files. Null or empty blockCount are not allowed, because blockSizes and blockStarts rely on blockCount. A visual representation of the BED format may have blocks appear thicker than the rest of the **feature**.

11. blockSizes: Comma-separated list of length blockCount containing the size of each **block**. There must be no spaces before or after commas. There may be a trailing comma after the last element of the list. blockSizes is mandatory in **BED12+** files. Null or empty blockSizes is not allowed, because blockStarts cannot be verified without blockSizes.

12. blockStarts: Comma-separated list of length blockCount containing each **block**'s start position, relative to chromStart. There must not be spaces before or after the commas. There may be a trailing comma after the last element of the list. Each element in blockStarts is paired with the corresponding element in blockSizes. Each blockStarts element must be an integer between 0 and $\mathsf{chromEnd} - \mathsf{chromStart}$, inclusive. For each couple i of $(\mathsf{blockStarts}_i, \mathsf{blockSizes}_i)$, the quantity $\mathsf{chromStart} + \mathsf{blockStarts}_i + \mathsf{blockSizes}_i$ must be less or equal to chromEnd. These conditions enforce that each **block** is contained within the **feature**. The first **block** must start at chromStart and the last **block** must end at chromEnd. Moreover, the **block**s must not overlap. The list must be sorted in ascending order. blockStarts is mandatory in **BED12+** files. Null or empty blockStarts is not allowed.

2 Examples

2.1 Example BED6 file from the UCSC Genome Browser FAQ[7]

```
chr7  127471196  127472363  Pos1  0  +
chr7  127472363  127473530  Pos2  0  +
chr7  127473530  127474697  Pos3  0  +
```

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

```
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  -
```

2.2 Example BED12 file from the UCSC Genome Browser FAQ

```
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
```

The **block**s in this example satisfy the required constraints. The first **block** starts at chromStart since the first blockStarts element is 0. The last **block** ends at chromEnd since the last **block** starts at position 4512 (1000+3512) with size 488, and therefore ends at position 5000 (4512+488).

3 Recommended practice for the BED format

3.1 Mandatory fields

- chrom: The name of each **chromosome** should also match the names from a reference genome, if applicable. For example, in the human genome, the chromosomes may be named `chr1` to `chr22`, `chrX`, `chrY`, and `chrM`. Names should be consistent within a **file**. For example, one should not use both `17` and `chr17` to represent the same **chromosome** in the same **file**.

3.2 Optional fields

- name: If a feature has no name, then a dot (`.`) should be used. Names should avoid using the space character even if the file is exclusively delimited with single tab characters because parsers may interpret a space as a delimiter.

- itemRgb: Eight or fewer colors should be used as too many colors may slow down visualizations and are difficult for humans to distinguish.[8]

3.3 User-defined fields

Custom data **fields** may contain any non-tab 7-bit US ASCII character. Definitions of a custom BED format should restrict the type of each **field** to the extent possible. Each custom **field** should contain either one of the following data types or a comma-separated list of values of the same type:

Type	Definition
Integer	String representation of 64-bit signed integer[9]
Unsigned	String representation of 64-bit unsigned integer[10]
Float	String representation of 64-bit floating point number[10]
Character	One character, other than tab
String	One or more characters, other than tab

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

[9] *IEEE 754–1985 IEEE Standard for Binary Floating-Point Arithmetic.* IEEE 754–1985, 1985

This specification does not contain a means for interchanging custom BED format definitions. The AutoSQL format[10] provides one method for defining custom BED formats in a separate file.

3.4 Sorting

BED **file**s should be sorted by chrom, then by chromStart numerically, and finally by chromEnd numerically. chrom may be sorted using any scheme (such as lexicographic or numeric order), but all lines with the same chrom value should occur consecutively. For example, the lexicographic order of `chr1`, `chr10`, `chr11`, `chr12`, . . . , `chr2`, `chr20`, `chr21`, . . . , `chr3`, . . . , `chrX`, `chrY`, `chrM` is an acceptable sorting. The numeric order of `chr1`, `chr2`, . . . , `chr21`, `chr22`, `chrM`, `chrX`, `chrY` is also acceptable. Regardless of the chromosome sorting scheme, lines for two features on the same chromosome should not have any lines for features on other chromosomes between them.

3.5 Whitespace

Though lines may use any kind of whitespace as a delimiter between **field**s, a single tab (`\t`) should be used. This is because almost all tools support tabs while some tools do not support other kinds of whitespace. Also, whitespace within the name **field** may be used only if the **field** delimiter is tab throughout the **file**.

3.6 Large BED files

If a **file** intended for visualization is over 50 MiB in size, the **file** should be converted to `bigBed` format, which is an indexed binary format.[11] The `bedToBigBed` program may perform this conversion.[12]

4 UCSC track files

Track files are files that contain additional information intended for a visualization tool such as the UCSC Genome Browser.[13] Track files contain browser lines and track lines that precede lines from a file format supported by the Genome Browser.[14] Track files are not valid BED files — valid BED files must not have any browser or track lines. To distinguish between BED files and track files, track files should use the file extension `.track`.

5 Acronyms

ASCII	American Standard Code for Information Interchange
BED	Browser Extensible Data
GA4GH	Global Alliance for Genomics and Health
regex	regular expression

[10] Kent, W. James. (2000) “AutoSQL.” https://hgwdev.gi.ucsc.edu/~kent/exe/doc/autoSql.doc

[11] Kent, W. James et al. (2010) “BigWig and BigBed: enabling browsing of large distributed datasets.” *Bioinformatics* 26(17):2204–2207. https://doi.org/10.1093/bioinformatics/btq351

[12] “bigBed Track Format.” UCSC Genome Browser FAQ, https://genome.ucsc.edu/goldenPath/help/bigBed.html

[13] Haeussler, Maximilian et al. (2019) “The University of California, Santa Cruz Genome Browser database: 2019 update.” *Nucleic Acids Research* 47(D1):D853–D858. https://doi.org/10.1093/nar/gky1095

[14] “Displaying your own annotations in the Genome Browser.” UCSC Genome Browser FAQ, https://genome.ucsc.edu/goldenPath/help/customTrack.html#lines

UCSC University of California, Santa Cruz

6 Acknowledgments

We thank W. James Kent and the UCSC Genome Browser team for creating the BED format. We thank W. James Kent and Hiram Clawson (UCSC); Eric Roberts (Princess Margaret Cancer Centre); John Marshall (University of Glasgow); Aaron R. Quinlan and Brent S. Pedersen (University of Utah); Ting Wang (Washington University in St. Louis); and the Global Alliance for Genomics and Health (GA4GH) File Formats Task Team for comments on this specification.