GFF3sort: an efficient tool to sort GFF3 files for tabix indexing

Tao Zhu, Chengzhen Liang, Zhigang Meng, Sandui Guo*, and Rui Zhang*

Biotechnology Research Institute, Chinese Academy of Agricultural Sciences, 100081, Beijing, China

Correspondence*

Sandui Guo

guosandui@caas.cn

Rui Zhang

zhangrui@caas.cn

Abstract

Motivation: The traditional method of visualizing gene annotation data in JBrowse is converting GFF3 files to JSON format, which is time-consuming. The latest version of JBrowse supports rendering sorted GFF3 files indexed by tabix, a novel strategy that is more convenient than the original conversion process. However, current tools available for GFF3 file sorting have bugs that would lead to erroneous rendering in JBrowse.

Results: We developed GFF3sort, a script to sort GFF3 files for tabix indexing. GFF3sort can properly deal with the order of features that have the same chromosome and start position. Based on our test datasets from multiple species, GFF3sort produced accurate sorting results while taking significantly less running time compared with currently available tools. We anticipate that GFF3sort will be a useful tool to help with genome annotation data processing and visualization.

Availability: https://github.com/billzt/gff3sort

Keywords: GFF3, JBrowse, Visualization, Tabix

Implementation

Introduction

As a powerful genome browser based on HTML5 and JavaScript, JBrowse has been widely used since released in 2009[1, 2]. According to its configuration document[3], it works by first converting genome annotation data in GFF3 file formats to JSON files by a built-in script “flatfile-to-json.pl”, and then rendering visualized element models such as genes, transcripts, repeat elements, etc. The main problem, however, is that this step is extremely time-consuming. The time is proportional to the number of feature elements in GFF3 files (Figure 1A and Additional file 1). Even for small genomes like yeast (*Saccharomyces cerevisiae*), it takes ~10 seconds to finish the conversion. For large and deeply annotated genomes such as that of humans, the time increases to more than 15 minutes. In addition, through the conversion process, a single GFF3 file is converted to thousands of piecemeal JSON files, thus putting a heavy burden on the ability to back up and store data.

In the recently released JBrowse version (v1.12.3), support for indexed GFF3 files has been added[4]. In this strategy, the GFF3 file is compressed with bgzip and indexed with tabix[5], which generates only two data files: a compressed file (.gz) and an index file (.tbi). Compared with the traditional processing protocol, the whole compression and index process could be finished within a few seconds even for large datasets such as the human genome annotation data (Figure 1A and Additional file 1). The tabix tool requires GFF3 files to be sorted by chromosomes and positions, which could be performed in the GNU sort program or the GenomeTools[6] package (see [7]). However, when dealing with feature lines in the same chromosome and position, both of the tools would sort them in an ambiguous way that usually results in parent features being placed behind their children (Figure 1B), causing erroneous rendering in JBrowse[8] (Figure 1B). An alternative sorting tool is needed to resolve this problem.

Here, we present GFF3sort, an efficient tool to sort GFF3 files for tabix indexing. Compared with GNU sort and GenomeTools, GFF3sort produces sorting results that could be correctly rendered by JBrowse while saving a significant amount of time. We anticipate that GFF3sort will be a useful tool to help with processing and visualizing genome annotation data.

Methods

GFF3sort is a script written in Perl. It uses a single hash table to store the input GFF3 annotation data (Figure 1C). For each feature, the chromosome ID and the start position are stored in the primary and secondary key, respectively. Features with the same chromosome and start position are grouped in an array in the same order of their appearance in the original GFF3 data. After sorting the hash table by chromosome IDs and start positions, GFF3sort implemented two modes to sort features within the array: the default mode and the precise mode (Figure 1C). In most situations, the original GFF3 annotations produced by genome annotation projects have already placed parent features before their children. Therefore, GFF3sort returns the feature lines in their original order, which is the default behavior. In some situations where orders in the input file has already been disturbed (for example, by GNU sort or GenomeTools), GFF3sort would sort them according to the parent-child topology using the sorting algorithm of directed acyclic graph[9], which is the most precise behavior but costs more computational source.

In order to test the performance of GFF3sort, the GFF3 annotation files of seven species (see the Data Sources in Table 1) were downloaded from the ENSEMBL database [10]. All the tests were conducted on a SuperMicro® server equipped with 80 Intel® Xeon® CPUs (2.40GHz), 128 GB RAM, and running the CentOS 6.9 system.

Functionality and Performance

GFF3sort takes a GFF3 file as its input data and returns a sorted GFF3 file as output. An optional parameter is used to turn on the precise mode. It outperforms the GNU sort program and GenomeTools in two aspects: correctness and running speed. In our seven test datasets, GFF3sort produces the sorted GFF3 files with a correct rate of 100% (measured by the percentage of parent features correctly placed before their children), compared with the <50% correct rate for GNU sort or GenomeTools (Table 1). It is able to fix the order of GFF3 files that has been incorrected sorted. Element models sorted by GFF3sort can be correctly rendered by JBrowse (Figure 1D). In addition, GFF3sort runs faster than both of those tools (Figure 1E and Additional file 1). In the default mode, GFF3sort saves ~70% running time. The precise mode takes longer time but still runs faster than traditional tools, especially for large annotation data.

In conclusion, GFF3sort is an efficient tool to sort GFF3 files for tabix indexing and therefore can be used to visualize annotation data. It has a high correct rate and a fast running speed compared with similar, existing tools. We anticipate that GFF3sort will be a useful tool to simplify data processing and visualization.

Figure Legends

Figure 1. **The motivation for, outlines of, and performance of GFF3sort**. A) Comparison of the running time of GFF3-to-JSON conversion and the bgzip-tabix process based on seven GFF3 annotation datasets: *Saccharomyces cerevisiae* (R64-1-1), *Aspergillus nidulans* (ASM1142v1), *Chlamydomonas reinhardtii* (INSDC v3.1), *Drosophila melanogaster* (BDGP6), *Arabidopsis thaliana* (Araport11), *Rattus norvegicus* (Rnor_6.0), and *Homo sapiens* (GRCh38). Feature numbers are measured by counting lines in the GFF3 file. B) An example of incorrectly sorted GFF3 data and its snapshots in JBrowse. The two lines (mRNA) marked in red were placed after their sub-features (exon or UTR). Such incorrect placement leads to losing the first

exon in JBrowse rendering results. C) Outlines of GFF3sort. D) An example of correctly sorted data by GFF3sort and its snapshots in JBrowse. In this example, the two lines (mRNA) marked in red were correctly placed before their sub-features, allowing JBrowse to render them properly. E) Comparison of the running time of GFF3sort (including the default mode and the precise mode) and other sorting tools.

Tables

Table 1. The correct rate[*] of GFF3 sorting results using different tools.

Data Source	**Parent Feature Number**	**GNU sort**	**GenomeTools**	**GFF3sort**
Saccharomyces cerevisiae (R64-1-1)	14,252	47.21%	50.08%	100%
Aspergillus nidulans (ASM1142v1)	21,653	48.94%	37.75%	100%
Chlamydomonas reinhardtii (INSDC v3.1)	29,047	49.68%	36.96%	100%
Drosophila melanogaster (BDGP6)	52,308	27.54%	23.26%	100%
Arabidopsis thaliana (Araport11)	86,846	36.20%	29.70%	100%
Rattus norvegicus (Rnor_6.0)	73,961	34.32%	41.17%	100%
Homo sapiens (GRCh38)	257,467	28.09%	17.41%	100%

[*] The correct rate is measured by the percentage of correctly sorted features that have children ones. Such feature includes coding or non-coding genes and transcripts. If a feature line is placed before all its children features, then it is considered as correctly sorted.

Additional files

Additional file 1: Benchmark data. This file displays: 1) the detailed running time of GFF3-to-JSON conversion and the bgzip-tabix process on our test datasets; 2) the detailed running time of GFF3sort, GNU sort, and GenomeTools on our test datasets. (DOCX)

Availability and requirements

Project name: GFF3sort

Project home page: https://github.com/billzt/gff3sort

Operating system(s): Linux

Programming language: Perl

Other requirements: No

License: No restrictions for academic users.

Any restrictions to use by non-academics: license needed

Declarations

List of abbreviations

JBrowse: JavaScript-based genome browser

GFF3: General Feature Format, version 3

JSON: JavaScript Object Notation

HTML5: HyperText Markup Language, version 5

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Funding

This work is supported by grants from the Ministry of Agriculture of China (Grant Nos. 2016ZX08005004, 2016ZX08009003-003-004).

Authors' contributions

SG, RZ, and TZ initiated the idea of the tool and conceived the project. TZ designed the tool and analyzed the data. CL and ZM helped to test the tool. TZ wrote the paper. All authors read and approved the final manuscript.

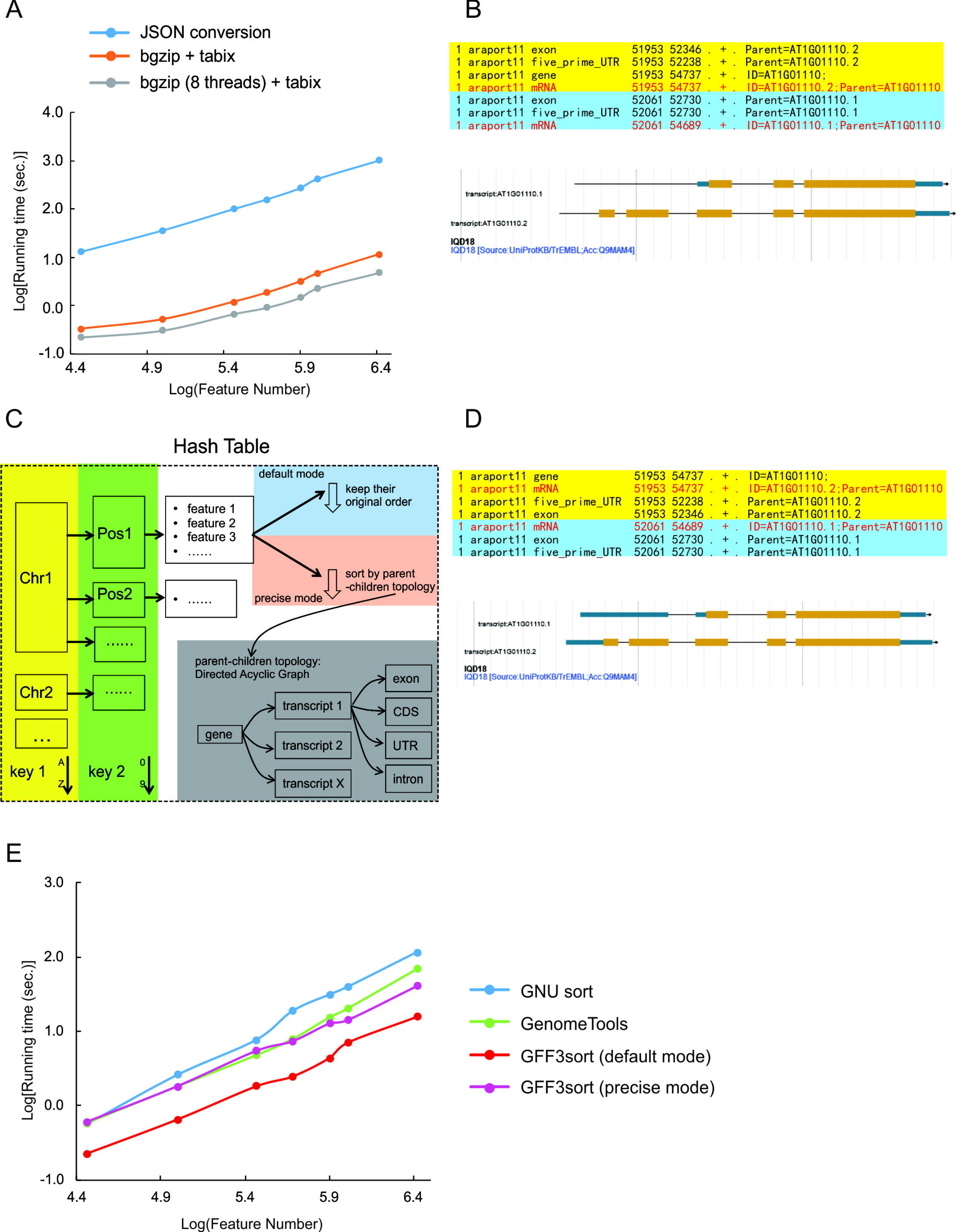
References

1. Skinner ME, Uzilov AV, Stein LD, Mungall CJ, Holmes IH: **JBrowse: A next-generation genome browser**. *Genome Res* 2009, **19**(9):1630-1638.
2. Buels R, Yao E, Diesh CM, Hayes RD, Munoz-Torres M, Helt G, Goodstein DM, Elsik CG, Lewis SE, Stein L *et al*: **JBrowse: a dynamic web platform for genome visualization and analysis**. *Genome Biol* 2016, **17**(1):66.
3. **JBrowse Configuration Guide** [http://gmod.org/wiki/JBrowse_Configuration_Guide] Accessed 26 May 2017
4. **JBrowse-1.12.3: Maintenance Release** [http://jbrowse.org/jbrowse-1-12-3/] Accessed 26 May 2017
5. Li H: **Tabix: fast retrieval of sequence features from generic TAB-delimited files**. *Bioinformatics* 2011, **27**(5):718-719.
6. Gremme G, Steinbiss S, Kurtz S: **GenomeTools: a comprehensive software library for efficient processing of structured genome annotations**. *IEEE/ACM Trans Comput Biol Bioinformatics* 2013, **10**(3):645-656.
7. **JBrowse FAQ** [http://gmod.org/wiki/JBrowse_FAQ] Accessed 26 May 2017
8. **Potential GFF3Tabix issues** [https://github.com/GMOD/jbrowse/issues/780]

Accessed 26 May 2017

9. **Sort::Topological - Topological Sort - metacpan.org** [https://metacpan.org/pod/Sort::Topological] Accessed 4 June 2017

10. Aken BL, Achuthan P, Akanni W, Amode MR, Bernsdorff F, Bhai J, Billis K, Carvalho-Silva D, Cummins C, Clapham P *et al*: **Ensembl 2017**. *Nucleic Acids Res* 2017, **45**(D1):D635-D642.



```
1 araport11 exon            51953 52346 . + . Parent=AT1G01110. 2
1 araport11 five_prime_UTR  51953 52238 . + . Parent=AT1G01110. 2
1 araport11 gene            51953 54737 . + . ID=AT1G01110;
1 araport11 mRNA            51953 54737 . + . ID=AT1G01110. 2;Parent=AT1G01110
1 araport11 exon            52061 52730 . + . Parent=AT1G01110. 1
1 araport11 five_prime_UTR  52061 52730 . + . Parent=AT1G01110. 1
1 araport11 mRNA            52061 54689 . + . ID=AT1G01110. 1;Parent=AT1G01110
```

```
1 araport11 gene            51953 54737 . + . ID=AT1G01110;
1 araport11 mRNA            51953 54737 . + . ID=AT1G01110. 2;Parent=AT1G01110
1 araport11 five_prime_UTR  51953 52238 . + . Parent=AT1G01110. 2
1 araport11 exon            51953 52346 . + . Parent=AT1G01110. 2
1 araport11 mRNA            52061 54689 . + . ID=AT1G01110. 1;Parent=AT1G01110
1 araport11 exon            52061 52730 . + . Parent=AT1G01110. 1
1 araport11 five_prime_UTR  52061 52730 . + . Parent=AT1G01110. 1
```