

# Package ‘Rgff’

May 7, 2026

**Title** R Utilities for GFF Files

**Version** 0.1.7

**Author** Juan Antonio Garcia-Martin [cre, aut] (ORCID:  
<<https://orcid.org/0000-0003-0993-4064>>),  
Juan Carlos Oliveros [aut, ctb] (ORCID:  
<<https://orcid.org/0000-0002-4520-0853>>),  
Rafael Torres-Perez [aut, ctb] (ORCID:  
<<https://orcid.org/0000-0002-3696-4720>>)

**Maintainer** Juan Antonio Garcia-Martin <[ja.garcia@cnb.csic.es](mailto:ja.garcia@cnb.csic.es)>

## Description

R utilities for gff files, either general feature format (GFF3) or gene transfer format (GTF) formatted files. This package includes functions for producing summary stats, check for consistency and sorting errors, conversion from GTF to GFF3 format, file sorting, visualization and plotting of feature hierarchy, and exporting user defined feature subsets to SAF format. This tool was developed by the BioinfoGP core facility at CNB-CSIC.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Imports** withr (>= 2.4.3), rlang (>= 0.4.12), stringi (>= 1.7.6),  
data.tree (>= 1.0.0), tibble (>= 3.1.6), dplyr (>= 1.0.7),  
jsonlite (>= 1.7.2), magrittr (>= 2.0.1)

**Suggests** DiagrammeR (>= 1.0.6.1), DiagrammeRsvg (>= 0.1), rsvg (>= 2.2.0), rmarkdown (>= 2.11), knitr (>= 1.36), GenomicRanges (>= 1.46.1), rtracklayer (>= 1.54.0), S4Vectors (>= 0.32.3)

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-05-09 09:00:02 UTC

## Contents

check\_gff . . . . . 2

|                            |   |
|----------------------------|---|
| get_features . . . . .     | 4 |
| gff_stats . . . . .        | 4 |
| gff_stats_by_chr . . . . . | 5 |
| gff_to_gff3 . . . . .      | 6 |
| plot_features . . . . .    | 6 |
| saf_from_gff . . . . .     | 7 |
| sort_gff . . . . .         | 8 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>10</b> |
|--------------|-----------|

---

|           |   |
|-----------|---|
| check_gff | <i>Test consistency and order of a GFF file</i> |
|-----------|---|

---

### Description

This function tests the consistency and order of a GFF file.

### Usage

```
check_gff(inFile, fileType = c("AUTO", "GFF3", "GTF"))
```

### Arguments

|          |  |
|----------|--|
| inFile   | Path to the input GFF file   |
| fileType | Version of the input file (GTF/GFF3). Default AUTO: determined from the file name. |

### Details

The following list indicates the code and description of the issues detected in GFF3 files

**NCOLUMNS\_EXCEEDED** Input file contains lines with more than 9 fields

**NCOLUMNS\_INFERIOR** Input file contains lines with less than 9 fields

**TOO\_MANY\_FEATURE\_TYPES** Input file contains too many (more than 100) different feature types

**NO\_IDS** ID attribute not found in any feature

**DUPLICATED\_IDS** There are duplicated IDs

**ID\_IN\_MULTIPLE\_CHR** The same ID has been found in more than one chromosome

**NO\_PARENTs** Parent attribute not found in any feature

**MISSING\_PARENT\_IDS** There are missing Parent IDs

**PARENT\_IN DIFFERENT CHR** There are features whose Parent is located in a different chromosome

**PARENT\_DEFINED\_BEFORE\_ID** Feature ids referenced in Parent attribute before being defined as ID

**NOT\_GROUPED\_BY\_CHR** Features are not grouped by chromosome

**NOT\_SORTED\_BY\_COORDINATE** Features are not sorted by start coordinate

**NOT\_VALID\_WARNING** File cannot be recognized as valid GFF3. Parsing warnings.

**NOT\_VALID\_ERROR** File cannot be recognized as valid GFF3. Parsing errors.

The following list indicates the code and description of the issues detected in GTF files

**NCOLUMNS\_EXCEEDED** Input file contains lines with more than 9 fields

**NCOLUMNS\_INFERIOR** Input file contains lines with less than 9 fields

**TOO\_MANY\_FEATURE\_TYPES** Input file contains too many (more than 100) different feature types

**NO\_GENE\_ID\_ATTRIBUTE** gene\_id attribute not found in any feature

**MISSING\_GENE\_IDS** There are features without gene\_id attribute

**NO\_GENE\_FEATURES** Gene features are not included in this GTF file

**DUPLICATED\_GENE\_IDS** There are duplicated gene\_ids

**GENE\_ID\_IN\_MULTIPLE\_CHR** The same gene\_id has been found in more than one chromosome

**NO\_TRANSCRIPT\_ID\_ATTRIBUTE** transcript\_id attribute not found in any feature There are no elements with transcript\_id attribute

**MISSING\_TRANSCRIPT\_IDS** There are features without transcript\_id attribute

**NO\_TRANSCRIPT\_FEATURES** Transcript features are not included in this GTF file

**DUPLICATED\_TRANSCRIPT\_IDS** There are duplicated transcript\_ids

**TRANSCRIPT\_ID\_IN\_MULTIPLE\_CHR** The same transcript\_id has been found in more than one chromosome

**DUPLICATED\_GENE\_AND\_TRANSCRIPT\_IDS** Same id has been defined as gene\_id and transcript\_id

**NOT\_GROUPED\_BY\_CHR** Features are not grouped by chromosome

**NOT\_SORTED\_BY\_COORDINATE** Features are not sorted by start coordinate

**NOT\_VALID\_WARNING** File cannot be recognized as valid GTF. Parsing warnings.

**NOT\_VALID\_ERROR** File cannot be recognized as valid GTF. Parsing errors.

### Value

A data frame of detected issues, including a short code name, a description and estimated severity each. In no issues are detected the function will return an empty data frame.

### Examples

```
test_gff3<-system.file("extdata", "eden.gff3", package="Rgff")
check_gff(test_gff3)
```

---

|              |  |
|--------------|--|
| get_features | <i>Analyses the feature type hierarchy of a GFF file</i> |
|--------------|--|

---

### Description

Based on the feature type hierarchy a GFF file, this function creates and returns a feature tree or a feature dependency table.

### Usage

```
get_features(  
  inFile,  
  includeCounts = FALSE,  
  outFormat = c("tree", "data.frame", "JSON"),  
  fileType = c("AUTO", "GFF3", "GTF")  
)
```

### Arguments

|                            |  |
|----------------------------|--|
| <code>inFile</code>        | Path to the input GTF/GFF3 features file   |
| <code>includeCounts</code> | Include number of occurrences of each feature and subfeature                               |
| <code>outFormat</code>     | Output format of the function. Available formats are: tree (DEFAULT), data.frame and JSON. |
| <code>fileType</code>      | Version of the input file (GTF/GFF3). Default AUTO: determined from the file name.         |

### Value

Depending on the `outFormat` selected returns a feature tree (tree), a feature dependency table as data.frame (data.frame) or a feature dependency table as JSON object (JSON)

### Examples

```
test_gff3<-system.file("extdata", "AthSmall.gff3", package="Rgff")  
get_features(test_gff3)
```

---

|           |   |
|-----------|---|
| gff_stats | <i>Summarizes the number of features of each type in a GFF file</i> |
|-----------|---|

---

### Description

This function summarizes the number of features of each type in a GFF file and returns the statistics

### Usage

```
gff_stats(inFile)
```

**Arguments**

inFile            Path to the input GFF file

**Value**

A tibble with the summary data

**Examples**

```
test_gff3<-system.file("extdata", "AthSmall.gff3", package="Rgff")
gff_stats(test_gff3)
```

---

|                  |  |
|------------------|--|
| gff_stats_by_chr | <i>Summarizes the number of elements of each type in each chromosome of a GFF file</i> |
|------------------|--|

---

**Description**

This function summarizes the number of features of each type in each chromosome of a GFF file and returns the statistics

**Usage**

```
gff_stats_by_chr(inFile)
```

**Arguments**

inFile            Path to the input GFF file

**Value**

A tibble with the summary data

**Examples**

```
test_gff3<-system.file("extdata", "AthSmall.gff3", package="Rgff")
gff_stats_by_chr(test_gff3)
```

---

|             |   |
|-------------|---|
| gtf_to_gff3 | <i>Converts a GTF file into a GFF3 file</i> |
|-------------|---|

---

### Description

This function converts a GTF file into a GFF3 file maintaining the feature hierarchy defined by the `gene_id` and `transcript_id` attributes. The remaining attributes of each feature will be kept with the same name and value.

### Usage

```
gtf_to_gff3(gtffile, outfile, forceOverwrite = FALSE)
```

### Arguments

|                             |  |
|-----------------------------|--|
| <code>gtffile</code>        | Path to the input GTF file   |
| <code>outfile</code>        | Path to the output GFF3 file, if not provided the output will be <code>gtffile.gff3</code> |
| <code>forceOverwrite</code> | If output file exists, overwrite the existing file. (default FALSE)                        |

### Value

Path to the generated GFF3 file

### Examples

```
## Not run:  
test_gtf<-system.file("extdata", "AthSmall.gtf", package="Rgff")  
gtf_to_gff3(test_gtf)  
  
## End(Not run)
```

---

|               |  |
|---------------|--|
| plot_features | <i>Plots or exports an image of the feature tree from a GFF file</i> |
|---------------|--|

---

### Description

This function plots the feature tree from a GFF file or, if an output file name is provided, exports an image of in the desired format ("png", "pdf" or "svg"). Packages "DiagrammeR", "DiagrammeRsvg" and "rsvg" must be installed to use this function.

**Usage**

```
plot_features(
  inFile,
  outFile,
  includeCounts = FALSE,
  fileType = c("AUTO", "GFF3", "GTF"),
  exportFormat = c("png", "pdf", "svg")
)
```

**Arguments**

|                            |   |
|----------------------------|---|
| <code>inFile</code>        | Path to the input GFF file  |
| <code>outFile</code>       | Path to the output features image file, if not provided the tree will be plotted  |
| <code>includeCounts</code> | Include number of occurrences of each subfeature  |
| <code>fileType</code>      | Version of the input file (GTF/GFF3). If not provided it will be determined from the file name.   |
| <code>exportFormat</code>  | Output image format when it is not possible to deduce it from the extension of <code>outFile</code> ("png", "pdf" or "svg"). Default, "png" |

**Value**

Path of the output features image file

**Examples**

```
if(requireNamespace("DiagrammeR")){
  test_gff3<-system.file("extdata", "AthSmall.gff3", package="Rgff")
  plot_features(test_gff3)
}
```

---

|                           |   |
|---------------------------|---|
| <code>saf_from_gff</code> | <i>Creates a SAF file from a GTF/GFF3 features for the given pairs of blocks and features</i> |
|---------------------------|---|

---

**Description**

This function creates a SAF file from a GTF/GFF3 features for the given blocks and features

**Usage**

```
saf_from_gff(
  inFile,
  outFile,
  fileType = c("AUTO", "GFF3", "GTF"),
  forceOverwrite = FALSE,
  features = c("gene > exon"),
  sep = ">"
)
```

**Arguments**

|                |   |
|----------------|---|
| inFile         | Path to the input GFF file  |
| outFile        | Path to the output SAF file, if not provided the output path will be the input path with the suffix ".feature1-block1.feature2-block2(...).saf"                   |
| fileType       | Version of the input file (GTF/GFF3). Default AUTO: determined from the file name.  |
| forceOverwrite | If output file exists, overwrite the existing file. (default FALSE)   |
| features       | Vector of pairs of features/blocks, separated by '>' (see sep argument). In the case of features without defined blocks, only the feature is needed (see example) |
| sep            | Separator of each "feature" and "block" provided in the feature argument (default '>')  |

**Value**

Path to the generated SAF file

**Examples**

```
test_gff3<-system.file("extdata", "AthSmall.gff3", package="Rgff")
## Default usage, extract gene features by exon blocks
saf_from_gff(test_gff3)
## Define only feature without block to count reads within the whole genomic locus
saf_from_gff(test_gff3, features=c("gene"))
## Define multiple features for counting readoverlapping only in exonic regions
saf_from_gff(test_gff3, features=c("gene > exon", "ncRNA_gene > exon"))
```

---

sort\_gff

*Sorts a GTF/GFF3 file*

---

**Description**

This function produces a sorted GFF file from an unsorted GFF file. The default order is by Chromosome, Start, End (reverse) and feature (based on the precedence in feature tree)

**Usage**

```
sort_gff(
  inFile,
  outFile,
  fileType = c("AUTO", "GFF3", "GTF"),
  forceOverwrite = FALSE
)
```

**Arguments**

|                             |  |
|-----------------------------|--|
| <code>inFile</code>         | Path to the input GFF file   |
| <code>outFile</code>        | Path to the output sorted file, if not provided the output will be the input path (without extension) with the suffix <code>sorted.gtf/gff3</code> |
| <code>fileType</code>       | Version of the input file (GTF/GFF3). Default AUTO: determined from the file name.   |
| <code>forceOverwrite</code> | If output file exists, overwrite the existing file. (default FALSE)  |

**Value**

Path to the sorted feature file

**Examples**

```
test_gff3<-system.file("extdata", "eden.gff3", package="Rgff")
sort_gff(test_gff3)
```

# Index

[check\\_gff](#), [2](#)

[get\\_features](#), [4](#)

[gff\\_stats](#), [4](#)

[gff\\_stats\\_by\\_chr](#), [5](#)

[gtf\\_to\\_gff3](#), [6](#)

[plot\\_features](#), [6](#)

[saf\\_from\\_gff](#), [7](#)

[sort\\_gff](#), [8](#)