

Giorgio Gonnella

RGFATools library - API documentation

Version 1.1

Table of Contents

Table of Contents	2
Documentation by YARD 0.8.7.6	4
Top Level Namespace	5
Defined Under Namespace	5
Constant Summary	5
Module: RGFATools	6
Overview	6
Defined Under Namespace	6
Constant Summary	6
Constant Summary	6
Constants included from Multiplication	6
Method Summary	6
Methods included from PBubbles	6
Methods included from LinearPaths	6
Methods included from SuperfluousLinks	6
Methods included from Multiplication	6
Methods included from InvertibleSegments	6
Methods included from CopyNumber	6
Methods included from Artifacts	6
Module: RGFATools::Artifacts	7
Overview	7
Instance Method Summary (collapse)	7
Instance Method Details	7
- (RGFA) remove_dead_ends(minlen)	7
- (RGFA) remove_small_components(minlen)	7
Module: RGFATools::PBubbles	8
Overview	8
Instance Method Summary (collapse)	8
Instance Method Details	8
- (RGFA) remove_p_bubble(segment_end1, segment_end2, count_tag: @default[:count_tag], unit_length: @default[:unit_length])	8
- (RGFA) remove_p_bubbles	8
Module: RGFATools::CopyNumber	9
Overview	9
Instance Method Summary (collapse)	9
Instance Method Details	9
- (RGFA) apply_copy_number(segment, count_tag: :cn, distribute: :auto, copy_names_suffix: :lowercase, origin_tag: :or, conserve_components: true)	9
- (RGFA) apply_copy_numbers(count_tag: :cn, distribute: :auto, copy_names_suffix: :lowercase, origin_tag: :or, conserve_components: true)	9
- (RGFA) compute_copy_numbers(single_copy_coverage, mincov: single_copy_coverage * 0.25, count_tag: @default[:count_tag], cn_tag: :cn, unit_length: @default[:unit_length])	10
- (RGFA) delete_low_coverage_segments(mincov, count_tag: @default[:count_tag], unit_length: @default[:unit_length])	10
- (RGFA) set_count_unit_length(unit_length)	10
- (RGFA) set_default_count_tag(tag)	11
Module: RGFATools::LinearPaths	12
Overview	12
Constant Summary	12
Instance Method Summary (collapse)	12
Instance Method Details	12
- (RGFA) merge_linear_path(segpath, **options)	12
Module: RGFATools::Multiplication	14

Overview	14
Constant Summary	14
Instance Method Summary (collapse)	14
Instance Method Details	14
- (RGFA) multiply(segment, factor, copy_names::lowercase, distribute::auto, conserve_components:true, origin_tag::or)	14
- (RGFA) multiply_without_rgfatools(segment, factor, copy_names::lowercase, conserve_components:true)	15
Module: RGFATools::SuperfluousLinks	16
Overview	16
Instance Method Summary (collapse)	16
Instance Method Details	16
- (RGFA) enforce_all_mandatory_links(conserve_components: true)	16
- (RGFA) enforce_segment_mandatory_links(segment, conserve_components: true)	16
- (RGFA) remove_self_link(segment)	16
- (RGFA) remove_self_links	17
Module: RGFATools::InvertibleSegments	18
Overview	18
Instance Method Summary (collapse)	18
Instance Method Details	18
- (RGFA) randomly_orient_invertible(segment)	18
- (RGFA) randomly_orient_invertibles	18
Class: RGFA	19
Overview	19
Constant Summary	19
Constant Summary	19
Constants included from RGFATools::Multiplication	19
Method Summary	19
Methods included from RGFATools::PBubbles	19
Methods included from RGFATools::LinearPaths	19
Methods included from RGFATools::SuperfluousLinks	19
Methods included from RGFATools::Multiplication	19
Methods included from RGFATools::InvertibleSegments	19
Methods included from RGFATools::CopyNumber	19
Methods included from RGFATools::Artifacts	19
Exception: RGFATools::Error	20
Overview	20

Documentation by YARD 0.8.7.6

The Graphical Fragment Assembly (GFA) described under github.com/pmelsted/GFA-spec/blob/master/GFA-spec.md is a proposed format which allows to describe the product of sequence assembly and is implemented in the RGFA class defined in the rgfa gem.

RGFATools is an extension to the RGFA library, which allow to perform further operations. Thereby additional conventions are required, with respect to the GFA specification, which are compatible with it.

The methods defined here allow, e.g., to randomly orient a segment which has the same connections on both sides, to compute copy numbers and multiply or delete segments according to them, to distribute the links of copies after multiplying a segment, or to eliminate edges in the graph which are incompatible with an hamiltonian path.

Custom optional fields are defined, such as "cn" for the copy number of a segment, "or" for the original segment(s) of a duplicated or merged segment, "mp" for the starting position of original segments in a merged segment, "rp" for the position of possible inversions due to arbitrary orientation of some segments by the program.

Furthermore a convention for the naming of the segments is introduced, which gives a special meaning to the characters "_^()".

This gem is depends on the "rgfa" gem. Please install the rgfa gem first (github.com/ggonnella/rgfa).

The API documentation is available as pdf under github.com/ggonnella/rgfatools/blob/master/pdfdoc/rgfatools-api-1.1.pdf or in HTML format (www.rubydoc.info/github/ggonnella/rgfatools/master/RGFA).

References

Giorgio Gonnella, Stefan Kurtz, "RGFA: powerful and convenient handling of assembly graphs" (2016)

The manuscript describing the library has been accepted for presentation at the German Conference on Bioinformatics 2016. The PeerJ preprint will be linked here, as soon as available.

Top Level Namespace

Defined Under Namespace

Classes: [RGFA](#)

Constant Summary

RGFATools =

[Module.new](#)

Module: RGFATools

Includes:	Artifacts , CopyNumber , InvertibleSegments , LinearPaths , Multiplication , PBubbles , SuperfluousLinks
Included in:	RGFA
Defined in:	lib/rgfatools.rb

Overview

Module defining additional methods for the RGFA class. In the main file is only the method redefinition infrastructure (private methods). The public methods are in the included modules.

Defined Under Namespace

Modules: [Artifacts](#), [CopyNumber](#), [InvertibleSegments](#), [LinearPaths](#), [Multiplication](#), [PBubbles](#), [SuperfluousLinks](#) **Classes:** [Error](#)

Constant Summary

Constant Summary

Constants included from [Multiplication](#)

[Multiplication::LINKS_DISTRIBUTION_POLICY](#)

Method Summary

Methods included from [PBubbles](#)

[#remove_p_bubble](#), [#remove_p_bubbles](#)

Methods included from [LinearPaths](#)

[#merge_linear_path](#)

Methods included from [SuperfluousLinks](#)

[#enforce_all_mandatory_links](#), [#enforce_segment_mandatory_links](#),
[#remove_self_link](#), [#remove_self_links](#)

Methods included from [Multiplication](#)

[#multiply_with_rgfatools](#), [#multiply_without_rgfatools](#)

Methods included from [InvertibleSegments](#)

[#randomly_orient_invertible](#), [#randomly_orient_invertibles](#)

Methods included from [CopyNumber](#)

[#apply_copy_number](#), [#apply_copy_numbers](#), [#compute_copy_numbers](#),
[#delete_low_coverage_segments](#), [#set_count_unit_length](#), [#set_default_count_tag](#)

Methods included from [Artifacts](#)

[#remove_dead_ends](#), [#remove_small_components](#)

Module: RGFATools::Artifacts

Included in:	RGFATools
Defined in:	lib/rgfatools/artifacts.rb

Overview

Methods which edit the graph components without traversal

Instance Method Summary

(collapse)

- (RGFA) `remove_dead_ends(minlen)`

Remove end segments, whose sequence length is under a specified value.

- (RGFA) `remove_small_components(minlen)`

Remove connected components whose sum of lengths of the segments is under a specified value.

Instance Method Details

- (RGFA) `remove_dead_ends(minlen)`

Remove end segments, whose sequence length is under a specified value.

Parameters:

- `minlen` (Integer) — the minimum length

Returns:

- (RGFA) — self
-

- (RGFA) `remove_small_components(minlen)`

Remove connected components whose sum of lengths of the segments is under a specified value.

Parameters:

- `minlen` (Integer) — the minimum length

Returns:

- (RGFA) — self

Module: RGFATools::PBubbles

Included in:	RGFATools
Defined in:	lib/rgfatools/p_bubbles.rb

Overview

Methods for the RGFA class, which involve a traversal of the graph following links

Instance Method Summary

(collapse)

```
- (RGFA) remove_p_bubble(segment_end1, segment_end2, count_tag:
@default[:count_tag], unit_length: @default[:unit_length])
```

Removes a p-bubble between segment_end1 and segment_end2.

```
- (RGFA) remove_p_bubbles
```

Removes all p-bubbles in the graph.

Instance Method Details

```
- (RGFA) remove_p_bubble(segment_end1, segment_end2, count_tag:
@default[:count_tag], unit_length: @default[:unit_length])
```

Removes a p-bubble between segment_end1 and segment_end2

Parameters:

- **segment_end1** (RGFA::SegmentEnd) — a segment end
- **segment_end2** (RGFA::SegmentEnd) — another segment end
- **count_tag** (Symbol) — *(defaults to: :RC or the value set by CopyNumber#set_default_count_tag)* the count tag to use for coverage computation
- **unit_length** (Integer) — *(defaults to: 1 or the value set by CopyNumber#set_count_unit_length)* the unit length to use for coverage computation

Returns:

- (RGFA) — self
-

```
- (RGFA) remove_p_bubbles
```

Removes all p-bubbles in the graph

Returns:

- (RGFA) — self

Module: RGFATools::CopyNumber

Included in:	RGFATools
Defined in:	lib/rgfatools/copy_number.rb

Overview

Methods which edit the graph components without traversal

Instance Method Summary

(collapse)

- (RGFA) **apply_copy_number**(segment, count_tag: :cn, distribute: :auto, copy_names_suffix: :lowercase, origin_tag: :or, conserve_components: true)
Applies the computed copy number to a segment.
- (RGFA) **apply_copy_numbers**(count_tag: :cn, distribute: :auto, copy_names_suffix: :lowercase, origin_tag: :or, conserve_components: true)
Applies the computed copy number to all segments.
- (RGFA) **compute_copy_numbers**(single_copy_coverage, mincov: single_copy_coverage * 0.25, count_tag: @default[:count_tag], cn_tag: :cn, unit_length: @default[:unit_length])
Self.
- (RGFA) **delete_low_coverage_segments**(mincov, count_tag: @default[:count_tag], unit_length: @default[:unit_length])
Delete segments which have a coverage under a specified value.
- (RGFA) **set_count_unit_length**(unit_length)
Sets the unit length (k-mer size, average read length or average fragment length) to use for coverage computation (defaults to: 1).
- (RGFA) **set_default_count_tag**(tag)
Sets the count tag to use as default by coverage computations (defaults to: :RC).

Instance Method Details

```
- (RGFA) apply_copy_number(segment, count_tag: :cn, distribute: :auto, copy_names_suffix: :lowercase, origin_tag: :or, conserve_components: true)
```

Applies the computed copy number to a segment

Parameters:

- **count_tag** (Symbol) — (defaults to: :RC or the value set by `#set_default_count_tag`) the count tag to use for coverage computation
- **copy_names_suffix** (:lowercase, :uppercase, :number, :copy) — (Defaults to: :lowercase) Symbol representing a system to compute the names from the name of the original segment. See "Automatic computation of the copy names".

Returns:

- (RGFA) — self

```
- (RGFA) apply_copy_numbers(count_tag: :cn, distribute: :auto,
```

```
copy_names_suffix: :lowercase, origin_tag: :or, conserve_components: true)
```

Applies the computed copy number to all segments

Parameters:

- **count_tag** (Symbol) — (defaults to: *:RC* or the value set by [#set_default_count_tag](#)) the count tag to use for coverage computation
- **copy_names_suffix** (:lowercase, :uppercase, :number, :copy) — (Defaults to: *:lowercase*) Symbol representing a system to compute the names from the name of the original segment. See "Automatic computation of the copy names".

Returns:

- (RGFA) — self

```
- (RGFA) compute_copy_numbers(single_copy_coverage, mincov:  
single_copy_coverage * 0.25, count_tag: @default[:count_tag], cn_tag: :cn,  
unit_length: @default[:unit_length])
```

Returns self

Parameters:

- **mincov** (Integer) — (defaults to: *1/4 of single_copy_coverage*) the minimum coverage, cn for segments under this value is set to 0
- **single_copy_coverage** (Integer) — the coverage that shall be considered to be single copy
- **cn_tag** (Symbol) — (defaults to: *:cn*) the tag to use for storing the copy number
- **count_tag** (Symbol) — (defaults to: *:RC* or the value set by [#set_default_count_tag](#)) the count tag to use for coverage computation
- **unit_length** (Integer) — (defaults to: *1* or the value set by [#set_count_unit_length](#)) the unit length to use for coverage computation

Returns:

- (RGFA) — self

```
- (RGFA) delete_low_coverage_segments(mincov, count_tag:  
@default[:count_tag], unit_length: @default[:unit_length])
```

Delete segments which have a coverage under a specified value.

Parameters:

- **mincov** (Integer) — the minimum coverage
- **count_tag** (Symbol) — (defaults to: *:RC* or the value set by [#set_default_count_tag](#)) the count tag to use for coverage computation
- **unit_length** (Integer) — (defaults to: *1* or the value set by [#set_count_unit_length](#)) the unit length to use for coverage computation

Returns:

- (RGFA) — self

```
- (RGFA) set_count_unit_length(unit_length)
```

Sets the unit length (k-mer size, average read length or average fragment length) to use for coverage computation (defaults to: 1).

Parameters:

- `unit_length (Integer)` — the unit length to use

Returns:

- `(RGFA)` — self
-

```
- (RGFA) set_default_count_tag(tag)
```

Sets the count tag to use as default by coverage computations (*defaults to: :RC*).

Parameters:

- `tag (Symbol)` — the tag to use

Returns:

- `(RGFA)` — self

Module: RGFATools::LinearPaths

Included in:	RGFATools
Defined in:	lib/rgfatools/linear_paths.rb

Overview

Methods for the RGFA class, which involve a traversal of the graph following links

Constant Summary

Instance Method Summary

(collapse)

– (RGFA) `merge_linear_path(segpath, **options)`

Merge a linear path, i.e.

Instance Method Details

– (RGFA) `merge_linear_path(segpath, **options)`

Merge a linear path, i.e. a path of segments without extra-branches. Extends the RGFA method, with additional functionality:

- `name`: the name of the merged segment is set to the name of the single segments joined by underscore (`_`). If a name already contained an underscore, it is splitted before merging. Whenever a segment is reversed complemented, its name (or the name of all its components) is suffixed with a `^`; if the last letter was already `^`, it is removed; if it contained `_` the name is splitted, the elements reversed and joined back using `_`; round parentheses are removed from the name before processing and added back after it.
- `:or`: keeps track of the origin of the merged segment; the origin tag is set to an array of `:or` or name (if no `:or` available) tags of the segment which have been merged; the character `^` is assigned the same meaning as in `name`
- `:rn`: tag used to store possible inversion positions and it is updated by this method; i.e. it is passed from the single segments to the merged segment, and the coordinates updated
- `:mp`: tag used to store the position of the single segments in the merged segment; it is created or updated by this method

Limitations: all containments und paths involving merged segments are deleted.

Parameters:

- `segpath` (Array<RGFA::SegmentEnd>) — a linear path, such as that retrieved by `#linear_path` (see RGFA API documentation)
- `options` (Hash) — optional keyword arguments

Options Hash (`**options`):

- `:merged_name` (String, :short, nil) — default: nil — if nil, the merged_name is automatically computed; if :short, a name is computed starting with "merged1" and calling next until an available name is found; if String, the name to use
- `:cut_counts` (Boolean) — default: false — if true, total count in merged segment `m`, composed of segments `s` of set `S` is multiplied by the factor $\text{Sum}(|s|)/|m|$

- **:disable_tracking** (Boolean) — default: `false` — if true, the original `#multiply` of RGFA without `RGFATools` is called.

Returns:

- ([RGFA](#)) — self

See Also:

- `#merge_linear_paths`

Module: RGFATools::Multiplication

Included in:	RGFATools
Defined in:	lib/rgfatools/multiplication.rb

Overview

Methods which edit the graph components without traversal

Constant Summary

LINKS_DISTRIBUTION_POLICY =

Allowed values for the links_distribution_policy option

`[:off, :auto, :equal, :E, :B]`

Instance Method Summary

(collapse)

- (RGFA) **multiply**(segment, factor, copy_names::lowcase, distribute::auto, conserve_components:true, origin_tag::or)

Create multiple copies of a segment.

- (RGFA) **multiply_without_rgfatools**(segment, factor, copy_names::lowcase, conserve_components:true)

Original multiply method of RGFA.

Instance Method Details

- (RGFA) **multiply**(segment, factor, copy_names::lowcase, distribute::auto, conserve_components:true, origin_tag::or)

Create multiple copies of a segment.

Complements the multiply method of gfatools with additional functionality. To call the original method use `#multiply_without_rgfatools`.

Automatic computation of the copy names:

- First, it is checked if the name of the original segment ends with a relevant string, i.e. a lower case letter (for `:lowcase`), an upper case letter (for `:upcase`), a digit (for `:number`), or the string `"_copy"` plus one or more optional digits (for `:copy`).
- If so, it is assumed, it was already a copy, and it is not altered.
- If not, then a (for `:lowcase`), A (for `:upcase`), 1 (for `:number`), `_copy` (for `:copy`) is appended to the string.
- Then, in all cases, next (*) is called on the string, until a valid, non-existent name is found for each of the segment copies
- (*) = except for `:copy`, where for the first copy no digit is present, but for the following is, i.e. the segment names will be `:copy`, `:copy2`, `:copy3`, etc.
- Can be overridden, by providing an array of copy names.

Links distribution policy

Depending on the value of the option `distribute`, an end is eventually selected for

distribution of the links.

- `:off`: no distribution performed
- `:E`: links of the E end are distributed
- `:B`: links of the B end are distributed
- `:equal`: select an end for which the number of links is equal to `factor`, if any; if both, then the E end is selected
- `:auto`: automatically select E or B, trying to maximize the number of links which can be deleted

Parameters:

- **factor** (Integer) — multiplication factor; if 0, delete the segment; if 1; do nothing; if > 1; number of copies to create
- **segment** (String, RGFA::Line::Segment) — segment name or instance
- **copy_names** (:lowercase, :uppercase, :number, :copy, Array<String>) — (Defaults to: :lowercase) Array of names for the copies of the segment, or a symbol, which defines a system to compute the names from the name of the original segment. See "Automatic computation of the copy names".
- **conserve_components** (Boolean) — (Defaults to: true) If factor == 0 (i.e. deletion), delete segment only if `#cut_segment?(segment)` is false (see RGFA API).
- **distribute** (RGFATools::Multiplication::LINKS_DISTRIBUTION_POLICY) — (Defaults to: :auto) Determines if and for which end of the segment, links are distributed among the copies. See "Links distribution policy".
- **origin_tag** (Symbol) — (Defaults to: :or) Name of the custom tag to use for storing origin information.

Returns:

- (RGFA) — self

```
- (RGFA) multiply_without_rgfatools(segment, factor, copy_names::lowercase,
conserve_components:true)
```

Original multiply method of RGFA. See the RGFA API documentation for detail.

Returns:

- (RGFA) — self

Module: RGFATools::SuperfluousLinks

Included in:	RGFATools
Defined in:	lib/rgfatools/superfluous_links.rb

Overview

Methods which edit the graph components without traversal

Instance Method Summary

(collapse)

- (RGFA) **enforce_all_mandatory_links**(*conserve_components: true*)
Remove superfluous links in the presence of mandatory links in the entire graph.
- (RGFA) **enforce_segment_mandatory_links**(*segment, conserve_components: true*)
Remove superfluous links in the presence of mandatory links for a single segment.
- (RGFA) **remove_self_link**(*segment*)
Remove links of segment to itself.
- (RGFA) **remove_self_links**
Remove all links of segments to themselves.

Instance Method Details

- (RGFA) **enforce_all_mandatory_links**(*conserve_components: true*)

Remove superfluous links in the presence of mandatory links in the entire graph

Parameters:

- **conserve_components** (Boolean) — (*Defaults to: true*) delete links only if #cut_link?(link) is false (see RGFA API).

Returns:

- (RGFA) — self

- (RGFA) **enforce_segment_mandatory_links**(*segment, conserve_components: true*)

Remove superfluous links in the presence of mandatory links for a single segment

Parameters:

- **segment** (String, RGFA::Line::Segment) — segment name or instance
- **conserve_components** (Boolean) — (*Defaults to: true*) delete links only if #cut_link?(link) is false (see RGFA API).

Returns:

- (RGFA) — self

- (RGFA) **remove_self_link**(*segment*)

Remove links of segment to itself

Parameters:

- **segment** (`String`, `RGFA::Line::Segment`) — segment name or instance

Returns:

- (`RGFA`) — self
-

- (`RGFA`) `remove_self_links`

Remove all links of segments to themselves

Returns:

- (`RGFA`) — self

Module: RGFATools::InvertibleSegments

Included in:	RGFATools
Defined in:	lib/rgfatools/invertible_segments.rb

Overview

Methods which edit the graph components without traversal

Instance Method Summary

(collapse)

- (RGFA) `randomly_orient_invertible(segment)`

Selects a random orientation for an invertible segment.

- (RGFA) `randomly_orient_invertibles`

Selects a random orientation for all invertible segments.

Instance Method Details

- (RGFA) `randomly_orient_invertible(segment)`

Selects a random orientation for an invertible segment

Parameters:

- **segment** (String, RGFA::Line::Segment) — segment name or instance

Returns:

- (RGFA) — self
-

- (RGFA) `randomly_orient_invertibles`

Selects a random orientation for all invertible segments

Returns:

- (RGFA) — self

Class: RGFA

Inherits:	Object	show all
Includes:	RGFATools	
Defined in:	lib/rgfatools.rb	

Overview

The main class of RGFA. See the RGFA API documentation.

Constant Summary

Constant Summary

Constants included from *RGFATools::Multiplication*

`RGFATools::Multiplication::LINKS_DISTRIBUTION_POLICY`

Method Summary

Methods included from *RGFATools::PBubbles*

`#remove_p_bubble`, `#remove_p_bubbles`

Methods included from *RGFATools::LinearPaths*

`#merge_linear_path`

Methods included from *RGFATools::SuperfluousLinks*

`#enforce_all_mandatory_links`, `#enforce_segment_mandatory_links`,
`#remove_self_link`, `#remove_self_links`

Methods included from *RGFATools::Multiplication*

`#multiply_with_rgfatools`, `#multiply_without_rgfatools`

Methods included from *RGFATools::InvertibleSegments*

`#randomly_orient_invertible`, `#randomly_orient_invertibles`

Methods included from *RGFATools::CopyNumber*

`#apply_copy_number`, `#apply_copy_numbers`, `#compute_copy_numbers`,
`#delete_low_coverage_segments`, `#set_count_unit_length`, `#set_default_count_tag`

Methods included from *RGFATools::Artifacts*

`#remove_dead_ends`, `#remove_small_components`

Exception: RGFATools::Error

Inherits:	StandardError	show all
Defined in:	lib/rgfatools/error.rb	

Overview

Parent class for errors raised by the RGFATools library

Generated on Fri Aug 5 05:35:44 2016 by [yard](#) 0.8.7.6 (ruby-2.0.0).