RGT Documentation

Release 0.0.1

Manuel Allhoff, Joseph Kuo, Eduardo G. Gusmao, Ivan G. Costa

January 12, 2016

Contents

Contents:

class GenomicRegion.**GenomicRegion**(*chrom*, *initial*, *final*, *name=None*, *orientation=None*, *data=None*, *proximity=None*)

This class describes a genomic region.

Define a GenomicRegion on [initial, final) on a particular chromosome.

Methods:

extend(left, right): Extend the region to the left and right side. Negative values are allowed.

overlap(region): Return true if region overlaps with argument-region, otherwise false.

Authors: Ivan G. Costa, Manuel Allhoff

distance(*y*)
Return the distance between two GenomicRegions. If overlapping, return 0; if on different chromosomes, return None

extend(*left*, *right*)
Extend GenomicRegion both-sided

extract_blocks()
Extract the block information in self.data in to GenomicRegionSet

get_data(*as_list=False*)
Return data as string (with special separating character (_$_)) or as list

overlap(*region*)
Return True, if GenomicRegion overlaps with region, else False.

class AnnotationSet.**AnnotationSet**(*gene_source*, *tf_source=None*, *alias_source=None*, *filter_havana=True*, *protein_coding=False*, *known_only=True*)

Annotation of genes and TFs' PWMs.

class DataType
Data type constants.

class AnnotationSet.**GeneField**
Gtf fields constants.

class AnnotationSet.**ReturnType**
Return type constants.

class AnnotationSet.**TfField**
Mtf fields constants.

AnnotationSet.**exact_mapping**(*caps=True*)
Maps (O(n log n)) exact entries of self.gene_list's gene names with self.tf_list's gene names. The mapping populates self.mapping_list with

AnnotationSet.**fix_gene_names**(*gene_set*, *output_dict=False*, *mute_warn=False*)
Checks if all gene names in gene_set are ensembl IDs. If a gene is not in ensembl format, it will be converted using alias_dict. If the gene name cannot be found then it is reported in a separate gene_set

Keyword arguments: gene_set – A GeneSet object.

Return: mapped_gene_list – A list of ensembl IDs unmapped_gene_list – A list of unmapped gene symbols/IDs

AnnotationSet.**get**(*query=None*, *list_type=0*, *return_type=0*)
Gets subsets of either self objects and returns different types.

Keyword arguments: query – A parameter that allows for subsets of self to be fetched. It can be:

- None: All fields/values are going to be returned.
- **A dictionary: Subsets the desired list according to this structure. Each** key must be a field (please refer to AnnotationSet.GeneField or AnnotationSet.TfField) that must point to a single value or a list of values.

list_type – Indicates which list should be subsetted/returned. Please refer to AnnotationSet.DataType. return_type – Indicates what should be returned. Please refer to AnnotationSet.ReturnType.

Return: result_list – A <return_type> containing the requested <list_type> subsetted according to <query>.

AnnotationSet.**get_exons**(*start_site=False*, *end_site=False*, *gene_set=None*)

Gets exons of genes. It returns a GenomicRegionSet with such exons. The id of each gene will be put in the NAME field of each GenomicRegion.

Keyword arguments: gene_set – A set of genes to narrow the search.

Return: result_grs – A GenomicRegionSet containing the exons. unmapped_gene_list – A list of genes that could not be mapped to an ENSEMBL ID.

AnnotationSet.**get_genes**(*gene_set=None*)

Gets regions of genes. It returns a GenomicRegionSet with such genes. The id of each gene will be put in the NAME field of each GenomicRegion.

Keyword arguments: gene_set – A set of genes to narrow the search.

Return: result_grs – A GenomicRegionSet containing the genes. unmapped_gene_list – A list of genes that could not be mapped to an ENSEMBL ID.

AnnotationSet.**get_introns**(*start_site=False*, *end_site=False*, *gene_set=None*)

Gets introns of genes. It returns a GenomicRegionSet with such introns. The id of each gene will be put in the NAME field of each GenomicRegion.

Keyword arguments: gene_set – A set of genes to narrow the search.

Return: result_grs – A GenomicRegionSet containing the introns. unmapped_gene_list – A list of genes that could not be mapped to an ENSEMBL ID.

AnnotationSet.**get_official_symbol**(*gene_name_source*)

Returns the official symbol(s) from gene_name_source.

Keyword arguments: gene_source – It can be a string (single gene name) or a GeneSet (multiple genes).

Return: if gene_source is string then returns the converted string gene name or None if gene name could not be converted. if gene_source is list then returns two lists containing, respectively, converted and not-converted gene names.

AnnotationSet.**get_promoters**(*promoterLength=1000*, *gene_set=None*, *unmaplist=False*)

Gets promoters of genes given a specific promoter length. It returns a GenomicRegionSet with such promoters. The ID of each gene will be put in the NAME field of each GenomicRegion. Each promoter includes also the coordinate of the 5' base pair, therefore each promoter actual length is promoterLength+1.

Keyword arguments: promoterLength – The length of the promoter region. gene_set – A set of genes to narrow the search.

Return: result_grs – A GenomicRegionSet containing the promoters. unmapped_gene_list – A list of genes that could not be mapped to an ENSEMBL ID.

AnnotationSet.**get_tss**(*gene_set=None*)

Gets TSS(Transcription start site) of genes. It returns a GenomicRegionSet with such TSS. The ID of each gene will be put in the NAME field of each GenomicRegion.

Keyword arguments: gene_set – A set of genes to narrow the search.

Return: result_grs – A GenomicRegionSet containing TSS. unmapped_gene_list – A list of genes that could not be mapped to an ENSEMBL ID.

AnnotationSet.**get_tts**(*gene_set=None*)
Gets TTS(Transcription termination site) of genes. It returns a GenomicRegionSet with such TTS. The ID of each gene will be put in the NAME field of each GenomicRegion.

Keyword arguments: gene_set – A set of genes to narrow the search.

Return: result_grs – A GenomicRegionSet containing TTS. unmapped_gene_list – A list of genes that could not be mapped to an ENSEMBL ID.

AnnotationSet.**inexact_mapping**()
TODO

AnnotationSet.**load_alias_dict**(*file_name*)
Reads an alias.txt file and creates a dictionary to translate gene symbols/alternative IDs to ensembl gene ID

Keyword arguments: file_name – Alias file name.

Return: void.

AnnotationSet.**load_gene_list**(*file_name*, *filter_havana=True*, *protein_coding=False*, *known_only=False*)
Reads gene annotation in gtf (gencode) format. It populates self.gene_list with such entries.

Keyword arguments: file_name – The gencode .gtf file name.

Return: void.

AnnotationSet.**load_tf_list**(*file_name_list*)
Reads TF annotation in mtf (internal – check manual) format. It populates self.tf_list with such entries. Everytime a TF annotation is loaded, a mapping with gene list is performed.

Keyword arguments: file_name_list – A list with .mtf files.

Return: void.

AnnotationSet.**map_lists**()
Maps self.gene_list with self.tf_list in various ways.

class Util.**AuxiliaryFunctions**
Class of auxiliary functions.

Authors: Eduardo G. Gusmao.

Methods:

#TODO

static correct_standard_bed_score(*score*)
Makes score between 0 and 1000

static overlap(*t1*, *t2*)
Checks if one interval contains any overlap with another interval.

Keyword arguments: t1 – First tuple. t2 – Second tuple.

Returns: Returns -1 if i1 is before i2; 1 if i1 is after i2; and 0 if there is any overlap.

static string_is_float(*s*)
Verifies if a string is a numeric float

static string_is_int(*s*)
Verifies if a string is a numeric integer

class Util.**ConfigurationFile**
Represent the data path configuration file (data.config).

Authors: Eduardo G. Gusmao.

It serves as a superclass to classes that will contain default variables (such as paths, parameters to tools, etc.) for a certain purpose (genomic data, motif data, etc.).

Variables:

- self.config: Represents the configuration file.
- self.data_dir: Represents the root path to data files.

class Util.**ErrorHandler**
Handles errors in a standardized way.

throw_error(*error_type*, *add_msg=''*)
Throws the specified error type. If the error type does not exist, throws a default error message and exits.

throw_warning(*warning_type*, *add_msg=''*)
Throws the specified warning type. If the warning type does not exist, throws a default warning message and exits.

class Util.**GenomeData**(*organism*)
Represent genomic data.

Authors: Eduardo G. Gusmao.

Inherits ConfigurationFile.

Methods:

get_organism(): Returns the current organism.

get_genome(): Returns the current path to the genome fasta file.

get_chromosome_sizes(): Returns the current path to the chromosome sizes text file.

get_association_file(): Returns the current path to the gene association text file.

get_annotation_dump_dir()
Returns the current path to the gencode annotation gtf file.

get_association_file()
Returns the current path to the gene association text file.

get_chromosome_sizes()
Returns the current path to the chromosome sizes text file.

get_gencode_annotation()
Returns the current path to the gencode annotation gtf file.

get_gene_alias()
Returns the current path to the gene alias txt file.

get_genome()
Returns the current path to the genome fasta file.

get_organism()
Returns the current organism.

class Util.**HelpfulOptionParser**(*usage=None*, *option_list=None*, *option_class=<class optparse.Option>*, *version=None*, *conflict_handler='error'*, *description=None*, *formatter=None*, *add_help_option=True*, *prog=None*, *epilog=None*)

An OptionParser that prints full help on errors.

class Util.**HmmData**

Represent HMM data.

Authors: Eduardo G. Gusmao.

Inherits Co7nfigurationFile.

Methods:

get_default_hmm(): Returns the current repository list.

get_default_bias_table_F()

Returns the current default bias table for the forward strand.

get_default_bias_table_R()

Returns the current default bias table for the reverse strand.

get_default_hmm_dnase()

Returns the current default DNase only hmm.

get_default_hmm_dnase_bc()

Returns the current default DNase only hmm.

get_default_hmm_dnase_histone()

Returns the current default DNase+histone hmm.

get_default_hmm_dnase_histone_bc()

Returns the current default DNase+histone hmm.

get_default_hmm_histone()

Returns the current default Histone only hmm.

class Util.**Html**(*name*, *links_dict*, *fig_dir=None*, *fig_rpath='../fig'*, *cluster_path_fix=''*, *RGT_header=True*, *other_logo=None*, *homepage=None*)

Represent an HTML file.

Authors: Eduardo G. Gusmao.

add_figure(*figure_path*, *notes=None*, *align=50*, *color='black'*, *face='Arial'*, *size=3*, *bold=False*, *width='800'*, *more_images=None*)

Add a figure with notes underneath.

Keyword arguments: figure_path – The path to the figure. notes – A list of strings for further explanation align – Alignment of the heading. Can be either an integer (interpreted as left margin)

or string (interpreted as HTML positional argument). (default 50)

Return: None – Appends the figure to the document.

add_fixed_rank_sortable()

Add jquery for fixing the first column of the sortable table

add_free_content(*content_list*)

Adds free HTML to the document.

Keyword arguments: content_list – List of strings. Each string is interpreted as a line in the HTML document.

Return: None – Appends content to the document.

add_heading(*heading*, *align=50*, *color='black'*, *face='Arial'*, *size=5*, *bold=True*, *idtag=None*)
Creates a heading.

Keyword arguments: heading – The heading title. align – Alignment of the heading. Can be either an integer (interpreted as left margin)

or string (interpreted as HTML positional argument). (default 50)

color – Color of the heading. (default "black") face – Font of the heading. (default "Arial") size – Size of the heading (HTML units [1,7]). (default 5) bold – Whether the heading is bold. (default True) id – Add ID tag in the heading element

Return: None – Appends heading to the document.

add_links()
Adds all the links.

Return: None – Appends links to the document.

add_list(*list_of_items*, *ordered=False*)
Add a list to the document

add_zebra_table(*header_list*, *col_size_list*, *type_list*, *data_table*, *align=50*, *cell_align='center'*, *auto_width=False*, *colorcode=None*, *header_titles=None*, *border_list=None*, *sortable=False*)
Creates a zebra table.

Keyword arguments: header_list – A list with the table headers in correct order. col_size_list – A list with the column sizes (integers). type_list – A string in which each character represents the type of each row.

s = string (regular word or number) i = image l = link

data_table – A table containing the data to be input according to each data type defined. s = string i = tuple containing: ("file name", width) width = an integer l = tuple containing: ("Name","Link")

align – Alignment of the heading. Can be either an integer (interpreted as left margin) or string (interpreted as HTML positional argument). (default 50)

cell_align – Alignment of each cell in the table (default center) auto_width – Adjust the column width by the content automatically regardless of defined col size colorcode – header_titles – Given a list corresponding to the header_list, which defines all the explanation in hint windows border_list – Return: None – Appends table to the document.

create_footer()
Adds footer.

Return: None – Appends footer to the document.

create_header(*relative_dir=None*, *RGT_name=True*, *other_logo=None*)
Creates default document header.

Return: None – Appends content to the document.

write(*file_name*)
Write HTML document to file name.

Keyword arguments: file_name – Complete file name to write this HTML document.

Return: None – Creates file with this HTML document.

class Util.**ImageData**
Represent image data.

Authors: Eduardo G. Gusmao.

Inherits ConfigurationFile.

Methods:

get_rgt_logo(): Returns the rgt logo image file location.

get_css_file(): Returns the css file location.

get_default_motif_logo(): Returns the default motif logo file location.

get_css_file()
Returns the css file location.

get_default_motif_logo()
Returns the default motif logo file location.

get_jquery()
Returns the default sorttable code location.

get_jquery_metadata()
Returns the default sorttable code location.

get_rgt_logo()
Returns the rgt logo image file location.

get_sorttable_file()
Returns the default sorttable code location.

get_tablesorter()
Returns the default sorttable code location.

get_tdf_logo()
Returns the default TDF logo.

get_viz_logo()
Returns the default TDF logo.

class Util.**MotifData**
Represent motif (PWM) data.

Authors: Eduardo G. Gusmao.

Inherits ConfigurationFile.

Methods:

get_repositories_list(): Returns the current repository list.

get_pwm_list(): Returns the list of current paths to the PWM repositories.

get_logo_list(): Returns the list of current paths to the logo images of PWMs in the given repositories.

get_mtf_list(): Returns the list of current paths to the mtf (motif annotation) files in the given repositories.

get_fpr_list(): Returns the list of current paths to the fpr (motif thresholds) files in the given repositories.

get_fpr_list()
Returns the list of current paths to the fpr files.

get_logo_file(*current_repository*)
Returns the path to a specific logo repository.

get_logo_list()
Returns the list of current paths to the logo images of PWMs in the given repositories.

get_mtf_list()
Returns the list of current paths to the mtf files.

get_mtf_path(*current_repository*)
Returns the path to a specific mtf file.

get_pwm_list()
Returns the list of current paths to the PWM repositories.

get_pwm_path(*current_repository*)
Returns the path to a specific motif repository.

get_repositories_list()
Returns the current repository list.

class Util.**OverlapType**
Class of overlap type constants.

Authors: Joseph Kuo.

Constants:

OVERLAP: Return new GenomicRegionSet including only the overlapping regions.

ORIGINAL: Return the regions of original GenomicRegionSet which have any intersections.

COMP_INCL: Return region(s) of the GenomicRegionSet which are 'completely' included.

class Util.**PassThroughOptionParser**(*usage=None*, *option_list=None*, *option_class=<class optparse.Option>*, *version=None*, *conflict_handler='error'*, *description=None*, *formatter=None*, *add_help_option=True*, *prog=None*, *epilog=None*)
An unknown option pass-through implementation of OptionParser. When unknown arguments are encountered, bundle with largs and try again, until rargs is depleted. sys.exit(status) will still be called if a known argument is passed incorrectly (e.g. missing arguments or bad argument types, etc.)

class Util.**SequenceType**
Class of sequence type Author: Joseph Kuo

Constants:

DNA, RNA

CHAPTER 1

Indices and tables

- genindex
- modindex
- search

a

g

m

u

A

C

D

E

F

G

H

I

L

M

O

P

S

T

U

W