

Annotation comparison Vetea Jacot

Generated by Doxygen 1.9.8

1 Namespace Index	1
1.1 Namespace List	1
2 Namespace Documentation	3
2.1 CDScompR Namespace Reference	3
2.1.1 Detailed Description	3
2.1.2 Function Documentation	4
2.1.2.1 build_cluster_list_from_Locus()	4
2.2 CDSmulticompR Namespace Reference	4
2.2.1 Detailed Description	4
2.2.2 Function Documentation	5
2.2.2.1 multicomp()	5
2.2.2.2 result_to_dict()	5
2.2.2.3 write_multi_results()	6
Index	7

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

CDScompR	Created on Thu May 2 11:57:29 2024	3
CDSmulticompR	Created on Fri Jul 5 13:18:44 2024	4

Chapter 2

Namespace Documentation

2.1 CDScompR Namespace Reference

Created on Thu May 2 11:57:29 2024.

Functions

- **format_mismatch_zones** (zones)
Formate une liste de coordonnées de mismatch en chaîne lisible.
- **write_results** (all_results, alt_name, out_dir)
- **build_cluster_list_from_Locus** (ref_loci, alt_loci, dna_mol)
Build clusters of loci from reference and alternative annotations.
- **annotation_comparison** (str ref_path, str alt_path, str out_dir)
Compare two GFF annotations and write results to a file.
- **usage ()**
- **main ()**

Variables

- **script_dir** = os.path.join(os.path.dirname(os.path.abspath(__file__)), "python_util")

2.1.1 Detailed Description

Created on Thu May 2 11:57:29 2024.

This script is used to compute the similarities between two structural annotations of a same genome, one reference and one alternative annotation.

Author

: vetea, ranwez

It expects as input the paths to the annotation files (in GFF format), displays the computed similarities between all annotation pairs, and creates a results CSV file detailing the loci comparisons between the annotations

2.1.2 Function Documentation

2.1.2.1 build_cluster_list_from_Locus()

```
CDScompR.build_cluster_list_from_Locus (
  ref_loci,
  alt_loci,
  dna_mol )
```

Build clusters of loci from reference and alternative annotations.

Args: `read_ref`: Dictionary mapping `chromosome_strand` to list of `Locus` objects from reference annotation `read_ref`
`_alt`: Dictionary mapping `chromosome_strand` to list of `Locus` objects from alternative annotation `dna_mol`
`:Chromosome_strand` key to process

Returns: List of Cluster objects

2.2 CDSmulticompR Namespace Reference

Created on Fri Jul 5 13:18:44 2024.

Functions

- [result_to_dict](#) (`result`)

Extracts the computed identity for each locus of the given result dictionary and add it to a new dictionary.
- [write_multi_results](#) (`multi_results`, `ref_path`, `alt_paths`)

Writes the results returned by the function multicomp into a results synthesis CSV file detailing the loci identity for each alternative.
- [multicomp](#) (`ref_path`, `alt_paths`, `verbose`, `create_strings`, `exon_mode`)

Compares all alternative annotations given (`alt_paths`) to the reference annotation (`ref_path`) by calling `annotation_sort` and appends the loci identities to a list.
- [usage](#) ()
- [main](#) ()

2.2.1 Detailed Description

Created on Fri Jul 5 13:18:44 2024.

This script is used to compute the distance between multiple structural annotations of a same genome, one reference and one or more alternative annotations.

Author

: vetea

It expects as input the paths to the annotation files (in GFF format), displays the computed distances between all annotation pairs, and creates results CSV files for each alternative and one global synthesis file

2.2.2 Function Documentation

2.2.2.1 multicomp()

```
CDSmulticompR.multicomp (
    ref_path,
    alt_paths,
    verbose,
    create_strings,
    exon_mode )
```

Compares all alternative annotations given (`alt_paths`) to the reference annotation (`ref_path`) by calling `annotation_sort` and appends the loci identities to a list.

See also

`annotation_sort()`

Parameters

<code>ref_path</code>	Path to the reference annotation file
<code>alt_path</code>	Path to the alternative annotation file
<code>verbose</code>	If True, triggers display of more information messages. Default is 'False'
<code>create_strings</code>	Boolean indicating whether to use the 'old' comparison function (<code>old_compare_loci</code> , 'True') or the new one (<code>compare_loci</code> , 'False')
<code>exon_mode</code>	Boolean indicating if the main comparison structures read from the file should be coding sequences (CDS, False) or exons (True). Default is 'False' (CDS comparison)

See also

`compare_loci()`
`old_compare_loci()`

Returns

Returns the list of all loci identities (dictionaries) of all alternatives

2.2.2.2 result_to_dict()

```
CDSmulticompR.result_to_dict (
    result )
```

Extracts the computed identity for each locus of the given result dictionary and add it to a new dictionary.

Parameters

<code>result</code>	Result dictionary, as returned by CDScompR's annotation_compare() function
---------------------	--------------------------------------------------------------------------------------------

Returns

Returns a dictionary with the locus name as key, and the comparison identity as value

2.2.2.3 write_multi_results()

```
CDSmulticompR.write_multi_results (
  multi_results,
  ref_path,
  alt_paths )
```

Writes the results returned by the function multicomp into a results synthesis CSV file detailing the loci identity for each alternative.

See also

[multicomp\(\)](#)

Parameters

<i>multi_results</i>	List of loci identities (dictionaries), as returned by multicomp
<i>ref_path</i>	Path to the reference annotation GFF file

Index

build_cluster_list_from_Locus
CDScompR, [4](#)

CDScompR, [3](#)
 build_cluster_list_from_Locus, [4](#)
CDSmulticompR, [4](#)
 multicomp, [5](#)
 result_to_dict, [5](#)
 write_multi_results, [6](#)

multicomp
 CDSmulticompR, [5](#)

result_to_dict
 CDSmulticompR, [5](#)

write_multi_results
 CDSmulticompR, [6](#)