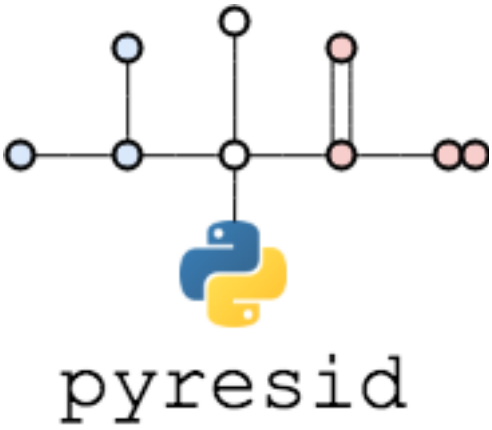

pyresid Documentation

Release 0.4.1

Rob Firth

Apr 27, 2018

CONTENTS:

Python tools for mining Protein Residues from Fulltext articles using PMC number, ePMC and PDB. Identify sentences in structural publications that refer to local features of a protein.

CHAPTER

ONE

HIGH-LEVEL FUNCTIONS

pyresid.**identify_residues**(*fulltext*, *verbose=False*)

Uses Regular Expressions to identify and locate usages of residues within the supplied *fulltext*. Returns a list of MatchClass objects that contain the start and end of the match within the text, and the matched string. For compound matches, a list of positions and residues is included in the match, which needs decomposing before further use. Based on Ravikumar+ 2012.

Parameters

- **fulltext** (*string*) – text to be searched for residues.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **matches** – The matches found within *fulltext*.

Return type List of *MatchClass* objects

See also:

- *locate_residues()*
- *process()*

pyresid.**locate_residues**(*source*, *matches*, *decompose=True*, *nlp=None*, *cifscantype='flex'*, *verbose=True*)

This function takes the raw list of matches from *identify_residues()* and augments them with contextual information and matches them against protein matches also found within the text.

This is a reincarnation of locate_residues2 (despite the name).

Parameters

- **source** (*SourceClass()*) – Class instance containing the source id and fulltext (and possibly the *spaCy* doc)
- **matches** (*List*) – a list of *MatchClass* objects
- **decompose** (*Bool, optional, default: True.*) – Whether to turn the "compound" mentions matched into actual residues.
- **nlp** (spaCy model - https://spacy.io/usage/models, optional, default: None) – The text model to use to turn the *source.fulltext* into *source.doc*
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **matches** – The matches found within *fulltext*, augmented with contextual information - sentence, prefix postfix, protein accession id.

Return type List of *MatchClass* objects

pyresid.**process**(*ext_id_list*, *outdir*, *filename='pyresid_output.json'*, *provider='West-Life'*, *cifscantype='flex'*, *save=True*, *overwrite=False*, *return_dict=False*, *decompose=True*, *verbose=False*)

This wraps the main workhorse functions, taking a list of PMC IDs and mining the resulting fulltext. output is a json structure (the encoded output of _locate_residues saved with MyEncoder), to match the EBI specifications.

Parameters

- **ext_id_list** (*List of strings*) – List containing ePMC identifiers used to retrieve the relevant entry. Format is prefix of 'PMC' followed by an integer.
- **outdir** (*String or Path*) – Directory that will contain the output file.
- **filename** (*String*) – The structured output JSON file containing the annotations, one line per *ext_id*.
- **save** (*Bool, optional, default: True*) – Flag to turn off writing the JSON. Good for debugging whenm combined with *return_dict*.
- **overwrite** (*Bool, optional, default: False*) – Flag to determine whether to append (default) or overwrite the json file
- **return_dict** (*Bool, optional, default: False*) – Flag to return the output as a dictionary
- **cifscantype** (*{"flex", "standard"}, default: "flex"*) – Flag passed to *pycifrw* via `_locate_residues2()`; *scantype* can be *standard* or *flex*. *standard* provides pure Python parsing at the cost of a factor of 10 or so in speed. *flex* will tokenise the input CIF file using fast C routines. Note that running PyCIFRW in Jython uses native Java regular expressions to provide a speedup regardless of this argument.
- **context** (*String, optional, default: "sent"*) – Flag passed to `_locate_residues2()` to determine the type of context added to annotations, "sent" uses the spaCy parsed sentences, anything else will use *x* characters either side of the matched tokens.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **(optional) outdict** – Dictionary containing the annotations. Good for debugging.

Return type OrderedDict

See also:

- *locate_residues()*

CHAPTER

TWO

CANDIDATE IDENTIFICATION FUNCTIONS

pyresid.**identify_protein_ID**(*fulltext*, *simple=False*, *infile=None*, *locdir=None*, *verbose=False*)

Uses Regular Expressions to find Protein IDs in the, and then cross-checks against the PDB list of entities.

Parameters

- **fulltext** (*string*) – text snippet to be searched for residues.
- **simple** (*Bool, optional, default: False*) – Flag that, if set, will skip the check against the PDB. Generally a bad idea.
- **infile** (*String or Path, optional, None*) – File to load in order to check candidate PDB entries against. contains the PDB entries - if *None* defaults to "PDBID.list"
- **locdir** (*String or Path, optional, default: None*) – Directory that contains the infile. If None, default is assumed to be *PDB_dir*, if this is not found then will be the user home.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **unique_protiens** – Matches found within *fulltext*

Return type Set

See also:

`_identify_residues()` *load_protein_IDs()*

pyresid.**check_residue_candidate_validity**(*cand*, *pattern='[a-zA-Z]{3}\\d+/\\d+|[a-zA-Z]{3}\\d+'*, *verbose=False*)

Parameters

- **cand** (*String*) – Candidate residue to check for validity
- **pattern** (*String, optional, default : "[a-zA-Z]{3}d+/d+|[a-zA-Z]{3}d+".*) – Regular Expression with which to match the candidate.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **match_bool_list** – List of True/False booleans reflecting the validity of the input candidates.

Return type List of Bool.

CHAPTER

THREE

LOCATION FUNCTIONS

pyresid.**locate_proteins**(*fulltext*, *simple=False*, *infile=None*, *locdir=None*, *verbose=False*)

Uses Regular Expressions to find Protein IDs in the, and then cross-checks against the PDB list of entities. Returns a list of MatchClass objects, rather than a list of strings, as in

Parameters

- **fulltext** (*string*) – text snippet to be searched for residues.
- **simple** (*Bool, optional, default: False*) – Flag that, if set, will skip the check against the PDB. Generally a bad idea.
- **infile** (*String or Path, optional, None*) – File to load in order to check candidate PDB entries against. contains the PDB entries - if *None* defaults to "PDBID.list"
- **locdir** (*String or Path, optional, default: None*) – Directory that contains the infile. If None, default is assumed to be *PDB_dir*, if this is not found then will be the user home.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **matches** – The matches found within *fulltext*.

Return type List of *MatchClass* objects

See also:

- *identify_residues()*
- *load_protein_IDs()*

pyresid.**locate_proteins**(*fulltext*, *simple=False*, *infile=None*, *locdir=None*, *verbose=False*)

Uses Regular Expressions to find Protein IDs in the, and then cross-checks against the PDB list of entities. Returns a list of MatchClass objects, rather than a list of strings, as in

Parameters

- **fulltext** (*string*) – text snippet to be searched for residues.
- **simple** (*Bool, optional, default: False*) – Flag that, if set, will skip the check against the PDB. Generally a bad idea.
- **infile** (*String or Path, optional, None*) – File to load in order to check candidate PDB entries against. contains the PDB entries - if *None* defaults to "PDBID.list"
- **locdir** (*String or Path, optional, default: None*) – Directory that contains the infile. If None, default is assumed to be *PDB_dir*, if this is not found then will be the user home.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **matches** – The matches found within *fulltext*.

Return type List of *MatchClass* objects

See also:

- *identify_residues()*
- *load_protein_IDs()*

CHAPTER

FOUR

API ITERACTION FUNCTIONS

pyresid.**request_fulltextXML**(*ext_id*)

Requests a fulltext XML document from the ePMC REST API. Raises a warning if this is not possible

Parameters **ext_id** (*String*) – ePMC identifier used to retrieve the relevant entry. Format is prefix of 'PMC' followed by an integer.

Returns **r** – The response to the query served up by the requests package.

Return type Requests.Response

pyresid.**parse_request**(*ext_id*)

Wrapper for *request_fulltextXML()* that returns a *BeautifulSoup* XML object

Parameters **ext_id** (*String*) – ePMC identifier used to retrieve the relevant entry. Format is prefix of 'PMC' followed by an integer.

Returns **soup** – BeautifulSoup XML object created from the text response from *pyresid.request_fulltextXML()*

Return type BeautifulSoup

See also:

request_fulltextXML()

CHAPTER

FIVE

OTHER FUNCTIONS

pyresid.**get_sections_text**(*ext_id*, *remove_tables=True*, *fulltext=False*, *verbose=False*)

Requests fulltext XML from the EBI ePMC web REST API, parses the response into a dict.

Parameters

- **ext_id** (*String*) – ePMC identifier used to retrieve the relevant entry. Format is prefix of 'PMC' followed by an integer.
- **remove_tables** (*Bool, optional, default: True*) – Flag to ignore the text found within tables.
- **fulltext** (*Bool, optional, default: False*) – Flag used to return a 'dumb' fulltext (rather than that from *reconstruct_fulltext()*)
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **text_dict** – Dictionary containing the parsed XML. Each entry corresponds to a Section in the XML (technically a child of the <*body*>).

Return type OrderedDict

See also:

- *request_fulltextXML()*
- *parse_request()*
- *reconstruct_fulltext()*

pyresid.**reconstruct_fulltext**(*text_dict*, *tokenise=True*, *verbose=False*)

Converts a *text_dict* into a single string or series of tokens in a list of strings.

Parameters

- **text_dict** (*Dict or OrderedDict*) – Dictionary containing the parsed XML. Each entry corresponds to a Section in the XML. Usually an output from *get_sections_text()*
- **tokenise** (*Bool, optional, default: False*) – Flag to enable or disable the reconstructed text being returned as spaCy tokens rather than a string
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns

- **fulltext** (*string*) – text snippet to be searched for residues.
- *OR*

- **fulltext_tokens** (*List of strings*) – Array of tokens that can be used, for example, to search for *residue_mentions*.

See also:

get_sections_text()

pyresid.**load_protein_IDs**(*infile=None*, *locdir=None*)
Reads in list of valid (approved and pending) PDB entries.

Parameters

- **infile** (*String or Path, optional, default: None*) – File that contains the PDB entries - defaults to "PDBID.list"
- **locdir** (*String or Path, optional, default: None*) – Directory that contains the infile. If None, default is assumed to be *PDB_dir*, if this is not found then will be the user home.

Returns **pdb_arr** – List of valid (approved and pending) PDB entries

Return type List of strings

See also:

- pyresid.combine_compound_IDs()
- pyresid.get_compound_IDfiles()

pyresid.**reconstruct_fulltext**(*text_dict*, *tokenise=True*, *verbose=False*)
Converts a *text_dict* into a single string or series of tokens in a list of strings.

Parameters

- **text_dict** (*Dict or OrderedDict*) – Dictionary containing the parsed XML. Each entry corresponds to a Section in the XML. Usually an output from *get_sections_text()*
- **tokenise** (*Bool, optional, default: False*) – Flag to enable or disable the reconstructed text being returned as spaCy tokens rather than a string
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns

- **fulltext** (*string*) – text snippet to be searched for residues.
- *OR*
- **fulltext_tokens** (*List of strings*) – Array of tokens that can be used, for example, to search for *residue_mentions*.

See also:

get_sections_text()

pyresid.**get_text**(*ext_id*, *verbose=False*)
A wrapper for *get_sections_text()* that adds additional information to the *text_dict*.

Parameters

- **ext_id** (*String*) – ePMC identifier used to retrieve the relevant entry. Format is prefix of 'PMC' followed by an integer.
- **verbose** (*Bool, optional, default: False*) – Flag to turn on verbose output

Returns **text_dict** – Dictionary containing the parsed XML. Each entry corresponds to a Section in the XML (technically a child of the <*body*>). An augmented version of the *text_dict* returned by *get_sections_text()* - containing additional information including spaCy tokens, length in characters and starting offset.

Return type OrderedDict

See also:

- *get_sections_text()*

pyresid.**setup_plot_defaults**()
Sets up default plot settings for figures.

CHAPTER

SIX

CLASSES

class pyresid.**SourceClass**
Class for handling sources

class pyresid.**MatchClass**(*start*, *end*, *string*)
Class for handling residue matches.

class pyresid.**ProteinMatchClass**(*start*, *end*, *string*)
Class for handling protein structure matches

class pyresid.**MyEncoder**(*, *skipkeys=False*, *ensure_ascii=True*, *check_circular=True*, *allow_nan=True*, *sort_keys=False*, *indent=None*, *separators=None*, *default=None*)

CHAPTER

SEVEN

LICENSE

BSD 3-Clause License

Copyright (c) 2018, Robert Elliot Firth (Science and Techology Faclities Council) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- *License*
- search

CHAPTER

NINE

STRUCTURE

INDEX