uparma-py Documentation

Release 0.1.0

tbd;

Dec 07, 2018

Contents

CHAPTER 1

uparma-py - the Python framework for the uparma project

UPARMA - Universal Parameter Mapper for Proteomics

1.1 Installation

Use pip to install, use virtualenv for maximum convenience:

```
user@localhost$ pip install -r requirements
```

1.2 Testing

Use pytest to run tests:

```
user@localhost$ pytest
```

1.2.1 Uparma - The Universal parameter mapper class in Python

class uparma.uparma.**UParma**(*refresh_jsons=False*, *source_style='ursgal_style_1'*)

Universal Parameter Mapper Class

Keyword Arguments

- **refresh_jsons** (*bool*) – indicates if uparma jsons should be pulled from central repo (https://github.com/uparma/uparma-lib) or not. Note that if jsons are not available in uparma folder, option is overridden and set to True.
- **source_style** (*str*) – Convenience when mapper is used to map from the same source style every time, in which case self.convert can be used instead of self.translate

convert (*param_dict*, *target_style=None*)
Convenient wrapper to translate params with the source style defined during init.

Calls self.translate with source_style = self.source_style

translate (*param_dict*, *source_style=None*, *target_style=None*)
Translate param_dict from source style into target style.

Keyword Arguments

- **param_dict** (*dict*) – dict containing parameter and value in a given style
- **source_style** (*str*) – style of the input format
- **target_style** (*str*) – style to which the parameters should be translated to.

Returns

dict with the translated key and values for the input dict with additional information in self.details (see below).

Return type translated_params (dict-like)

For example an input in ursgal style:

```
{
    "precursor_mass_tolerance_unit": "ppm",
    "min_pep_length" : 8
}
```

can be converted to msgfplus style, yielding:

```
{
    '-minLength' : 8,
    '-t' : 'ppm'
}
```

the return object is a dict-like structure which holds additional detailed information accessible via self.details. For the example above, self.details looks like:

```
{
    'min_pep_length': {
        'source_key': 'min_pep_length',
        'source_style': 'ursgal_style_1',
        'source_value': 8,
        'target_key': '-minLength',
        'target_style': 'msgfplus_style_1',
        'target_value': 8
    },
    'precursor_mass_tolerance_unit': {
        'source_key': 'precursor_mass_tolerance_unit',
       'source_style': 'ursgal_style_1',
       'source_value': 'ppm',
       'target_key': '-t',
       'target_style': 'msgfplus_style_1',
       'target_value': 'ppm'
    }
}
```

1.2.2 Examples

Get parameter information

get_param_info.**main**(*parameter*)

Get uparma information for a specific parameter name. Description, value transltion and uparma id is extracted.

usage:

./get_param_info.py <parameter_name>

E.g.:

./get_param_info.py 'spectrum, parent monoisotopic mass error units'

The example provides the name of the monoisotopic mass error units of the X!Tandem engines. This name is mapped back to translation styles matching this name.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Latest update Dec 07, 2018!

Index