jpredapi Documentation

Release 1.5.6

Andrey Smelter

Sep 23, 2018

Contents

CHAPTER 1

jpredapi

The *jpredapi* package provides a simple Python interface for submitting and retrieving jobs from JPred: A Protein Secondary Structure Prediction Server (JPred).

This is unofficial Python port of *jpredapi* perl script that can be found here: http://www.compbio.dundee.ac.uk/jpred4/api.shtml

1.1 Links

- jpredapi @ GitHub
- jpredapi @ PyPI
- Documentation @ ReadTheDocs

1.2 Installation

1.2.1 Install on Linux, Mac OS X

```
python3 -m pip install jpredapi
```

1.2.2 Install on Windows

```
py -3 -m pip install jpredapi
```

Note: Read the User Guide and The jpredapi Tutorial on ReadTheDocs to learn more and to see code examples on using the *jpredapi* as a library and as a command-line tool.

1.3 License

This package is distributed under the MIT *license*.

Contents:

1.3.1 User Guide

Description

The *jpredapi* package provides a simple Python interface for submitting and retrieving jobs from JPRED: A Protein Secondary Structure Prediction Server (JPRED).

Installation

The *jpredapi* package runs under Python 2.7 and Python 3.4+. Starting with Python 3.4 pip is included by default. To install system-wide with pip run the following:

Install on Linux, Mac OS X

```
python3 -m pip install jpredapi
```

Install on Windows

```
py -3 -m pip install jpredapi
```

Install inside virtualenv

For an isolated install, you can run the same inside a virtualenv.

```
$ virtualenv -p /usr/bin/python3 venv  # create virtual environment, use python3␣
↪interpreter

$ source venv/bin/activate             # activate virtual environment

$ python3 -m pip install jpredapi      # install jpredapi as usually

$ deactivate                           # if you are done working in the virtual␣
↪environment
```

Dependencies

jpredapi depends on several Python libraries, it will install its dependencies automatically, but if you wish to install them manually, then run commands below:

- **docopt for creating *jpredapi* command-line interface.**
  - To install docopt run the following:

```
python3 -m pip install docopt  # On Linux, Mac OS X
py -3 -m pip install docopt    # On Windows
```

- **requests for sending HTTP/1.1 requesrs to JPRED server.**
  - To install requests Python library run the following:

```
python3 -m pip install requests  # On Linux, Mac OS X
py -3 -m pip install requests    # On Windows
```

- **retrying for controlling status requests.**
  - To install retrying Python library run the following:

```
python3 -m pip install retrying  # On Linux, Mac OS X
py -3 -m pip install retrying    # On Windows
```

Basic usage

jpredapi can be used in several ways:

- As a library within interactive Python shell or Python script and as a command-line tool to:
  - Submit JPRED job.
  - Check status of JPRED job.
  - Retrieve results of JPRED job.

Note: Read *The jpredapi Tutorial* to learn more and see code examples on using `jpred`.

1.3.2 The jpredapi Tutorial

The *jpredapi* package provides functions to submit, check status, and retrieve results from JPred: A Secondary Structure Prediction Server (JPred).

Command Line Interface

```
jpredapi command-line interface

The RESTful API allows JPred users to submit jobs from the command-line.

Usage:
    jpredapi submit (--mode=<mode> --format=<format>) (--file=<filename> | --seq=
↪<sequence>)
                    [--email=<name@domain.com>] [--name=<name>] [--rest=<address>] [--
↪skipPDB] [--silent]
    jpredapi status (--jobid=<id>) [--results=<path>] [--wait=<interval>] [--attempts=
↪<max>]
                    [--rest=<address>] [--jpred4=<address>] [--extract] [--silent]
    jpredapi get_results (--jobid=<id>) [--results=<path>] [--wait=<interval>] [--
↪attempts=<max>]
                         [--rest=<address>] [--jpred4=<address>] [--extract] [--
↪silent]
```

(continues on next page)

(continued from previous page)

```
    jpredapi quota (--email=<name@domain.com>) [--silent]
    jpredapi check_rest_version [--rest=<address>] [--silent]
    jpredapi -h | --help
    jpredapi -v | --version

Options:
    -h, --help                  Show this help message.
    -v, --version               Show jpredapi package version.
    --silent                    Do not print messages.
    --extract                   Extract results tar.gz archive.
    --skipPDB                   PDB check.
    --mode=<mode>               Submission mode, possible values: single, batch, msa.
    --format=<format>           Submission format, possible values: raw, fasta, msf,␣
↪blc.
    --file=<filename>           Filename of a file with the job input (sequence(s)).
    --seq=<sequence>            Instead of passing input file, for single-sequence␣
↪submission.
    --email=<name@domain.com>   E-mail address where job report will be sent (optional␣
↪for all but batch submissions).
    --name=<name>               Job name.
    --jobid=<id>                Job id.
    --results=<path>            Path to directory where to save archive with results.
    --rest=<address>            REST address of server [default: http://www.compbio.
↪dundee.ac.uk/jpred4/cgi-bin/rest].
    --jpred4=<address>          Address of Jpred4 server [default: http://www.compbio.
↪dundee.ac.uk/jpred4].
    --wait=<interval>           Wait interval before retrying to check job status in␣
↪seconds [default: 60].
    --attempts=<max>            Maximum number of attempts to check job status␣
↪[default: 10].
```

Print jpredapi help message

```
$ python3 -m jpredapi --help
```

Print jpredapi version

```
$ python3 -m jpredapi --version
```

Submit jobs to JPred server

Submit single sequence in `raw` format using `--seq` parameter:

```
python3 -m jpredapi submit --mode=single --format=raw --seq=MQVWPIEGIKKFETLSYLPP
```

Submit single sequence in `raw` format using `--file` parameter:

```
python3 -m jpredapi submit --mode=single --format=raw --file=tests/example_data/
↪single_raw.example
```

Content of `single_raw.example` file:

```
MQVWPIEGIKKFETLSYLPPLTVEDLLKQIEYLLRSKWVPCLEFSKVGFVYRENHRSPGYYDGRYWTMWKLPMFGCTDATQVLKELEEAKKAYPDAFVRI
```

Submit single sequence in `fasta` format using `--file` parameter:

```
python3 -m jpredapi submit --mode=single --format=fasta --file=tests/example_data/
↪single_fasta.example
```

Content of `single_fasta.example` file:

```
>my test sequence
MQVWPIEGIKKFETLSYLPPLTVEDLLKQIEYLLRSKWVPCLEFSKVGFVYRENHRSPGYYDGRYWTMWKLPMFGCTDATQVLKELEEAKKAYPDAFVRI
```

Submit multiple sequences in `fasta` format using `--file` parameter:

```
python3 -m jpredapi submit --mode=batch --format=fasta --file=tests/example_data/
↪batch_fasta.example --email=name@domain.com
```

Content of `batch_fasta.example` file:

```
>my_seq1
MKFLVLLFNILCLFPILGADELVMSPIPTTDVQPKVTFDINSEVSSGPLYLNPVEMAGVK
YLQLQRQPGVQVHKVVEGDIVIWENEEMPLYTCAIVTQNEVPYMAYVELLEDPDLIFFLK
EGDQWAPIPEDQYLARLQQLRQQIHTESFFSLNLSFQHENYKYEMVSSFQHSIKMVVFTP
KNGHICKMVYDKNIRIFKALYNEYVTSVIGFFRGLKLLLLNIFVIDDRGMIGNKYFQLLD
DKYAPISVQGYVATIPKLKDFAEPYHPIILDISDIDYVNFYLGDATYHDPGFKIVPKTPQ
CITKVVDGNEVIYESSNPSVECVYKVTYYDKKNESMLRLDLNHSPPSYTSYYAKREGVWV
TSTYIDLEEKIEELQDHRSTELDVMFMSDKDLNVVPLTNGNLEYFMVTPKPHRDIIIVFD
GSEVLWYYEGLENHLVCTWIYVTEGAPRLVHLRVKDRIPQNTDIYMVKFGEYWVRISKTQ
>my_seq2
MASVKSSSSSSSSFISLLLLILLVIVLQSQVIECQPQQSCTASLTGLNVCAPFLVPGSP
TASTECCNAVQSINHDCMCNTMRIAAQIPAQCNLPPLSCSAN
>my_seq3
MEKKSIAGLCFLFLVLFVAQEVVVQSEAKTCENLVDTYRGPCFTTGSCDDHCKNKEHLLS
GRCRDDVRCWCTRNC
```

Submit multiple sequence alignment files in `fasta` format:

```
python3 -m jpredapi submit --mode=msa --format=fasta --file=tests/example_data/msa_
↪fasta.example --email=name@domain.com
```

Content of `msa_fasta.example` file:

```
>QUERY_1
MQVWPIEGIKKFETLSYLPPLTVEDLLKQIEYLLRSKWVPCLEFSKVGFVYRENHRSPGYYDGRYWTMWKLP
MFGCTDATQVLKELEEAKKAYPDAFVRIIGFDNVRQVQLISFIAYKPPGC
>UniRef90_Q40250_2
MKVWPPIGLKKYETLSYLPPLSDEALSKEIDYLIRNKWIPCLEFEEHGFVYREHHHSPGYYDGRYWTMWKLP
MFGCTDSAQVMKEVGECKKEYPNAFIRVIGFDNIRQVQCISFIVAKPPGV
>UniRef90_A7YVW5_3
MQVWPPLGKRKFETLSYLPPLPVDALLKQIDYLIRSGWIPCIEFTVEGFVYREHHHSPGYYDGRYWTMWKLP
MYGCTDSTQVLAEVEANKKEYPNSYIRIIGFDNKRQVQCVSFIVHTPPS-
>UniRef90_P04714_4
MQVWPPYGKKKYETLSYLPDLTDEQLLKEIEYLLNKGWVPCLEFTEHGFVYREYHASPRYYDGRYWTMWKLP
MFGCTDATQVLGELQEAKKAYPNAWIRIIGFDNVRQVQCISFIAYKPPG-
>UniRef90_W9RUU9_5
MQVWPPRGKLKFETLSYLPDLTDEQLLKEIDYLLRSNWIPCLEFEVKAHIYRENNRSPGYYDGRYWTMWKLP
MFGCTDATQVLAEVQETKKAYPDAHVRIIGFDNNRQVQCISFIAYKPPA-
```

Submit multiple sequence alignment files in `msf` format:

```
python3 -m jpredapi submit --mode=msa --format=msf --file=tests/example_data/msa_msf.
↪example --email=name@domain.com
```

Content of `msa_msf.example` file:

```
/tmp/filelPdICy  MSF: 108  Type: N  January 01, 1776  12:00  Check: 2741 ..

Name: 0_1a             Len:   108  Check:  4063  Weight:  1.00
Name: 1_MA             Len:   108  Check:  4875  Weight:  1.00
Name: 2_KE             Len:   108  Check:   449  Weight:  1.00
Name: 3_NC             Len:   108  Check:  3354  Weight:  1.00

//

           0_1a  APAFSVSPAS GASDGQSVSV SVAAAGETYY IAQCAPVGGQ DACNPATATS
           1_MA  APGVTVTPAT GLSNGQTVTV SATTPGTVYH VGQCAVVEGV IGCDATTSTD
           2_KE  SAAVSVSPAT GLADGATVTV SASATSTSAT ALQCAILAGR GACNVAEFHD
           3_NC  APTATVTPSS GLSDGTVVKV AGAQAGTAYD VGQCAWVDGV LACNPADFSS

           0_1a  FTTDASGAAS FSFTVRKSYA GQTPSGTPVG SVDCATDACN LGAGNSGLNL
           1_MA  VTADAAGKIT AQLKVHSSFQ AVVANGTPWG TVNCKVVSCS AGLGSDSGEG
           2_KE  FSLSG.GEGT TSVVVRRSFT GYVPDGPEVG AVDCDTAPCE IVVGGNTGEY
           3_NC  VTADANGSAS TSLTVRRSFE GFLFDGTRWG TVDCTTAACQ VGLSDAAGNG

           0_1a  GHVALTFG
           1_MA  AAQAITFA
           2_KE  GNAAISFG
           3_NC  PGVAISFN
```

Submit multiple sequence alignment files in `blc` format:

```
python3 -m jpredapi submit --mode=msa --format=blc --file=tests/example_data/msa_blc.
↪example --email=name@domain.com
```

Content of `msa_blc.example` file:

```
>0_1a  Name
>1_MA  Name
>2_KE  Name
>3_NC  Name
* iteration 1
AASA
PPAP
AGAT
FVVA
STST
VVVV
STST
PPPP
AAAS
*
```

Check job status on JPred server

Check single job status using `jobid`:

```
python3 -m jpredapi status --jobid=jp_K46D05A
```

Check single job status using `jobid` and retrieve results:

```
python3 -m jpredapi status --jobid=jp_K46D05A --results=jpred_sspred/results
```

Check single job status using `jobid`, retrieve results, and decompress archive:

```
python3 -m jpredapi status --jobid=jp_K46D05A --results=jpred_sspred/results --extract
```

Retrieve results from JPred server

Retrieve results using `jobid`:

```
python3 -m jpredapi get_results --jobid=jp_K46D05A --results=jpred_sspred/results
```

Retrieve results using `jobid` and decompress archive:

```
python3 -m jpredapi get_results --jobid=jp_K46D05A --results=jpred_sspred/results --
↪extract
```

Check how many jobs you have already submitted on a given day:

```
python3 -m jpredapi quota --email=name@domain.com
```

Using jpredapi as a library

Importing jpredapi module

If *jpredapi* package is installed on the system, it can be imported:

```
>>> import jpredapi
>>>
```

Submit jobs to JPred server

Submit single sequence in `raw` format using `seq` parameter:

```
>>> import jpredapi
>>>
>>> jpredapi.submit(mode="single", user_format="raw", seq="MQVWPIEGIKKFETLSYLPP")
>>>
```

Submit single sequence in `raw` format using `file` parameter:

```
>>> jpredapi.submit(mode="single", user_format="raw", file="tests/example_data/single_
↪raw.example")
>>>
```

Submit single sequence in `fasta` format using `file` parameter:

```
>>> jpredapi.submit(mode="single", user_format="fasta", file="tests/example_data/
↪single_fasta.example")
>>>
```

Submit multiple sequences in `fasta` format using `file` parameter:

```
>>> jpredapi.submit(mode="batch", user_format="fasta", file="tests/example_data/batch_
↪fasta.example", email="name@domain.com")
>>>
```

Submit multiple sequence alignment files in `fasta` format:

```
>>> jpredapi.submit(mode="msa", user_format="fasta", file="tests/example_data/msa_
↪fasta.example", email="name@domain.com")
>>>
```

Submit multiple sequence alignment files in `msf` format:

```
>>> jpredapi.submit(mode="msa", user_format="msf", file="tests/example_data/msa_msf.
↪example", email="name@domain.com")
>>>
```

Submit multiple sequence alignment files in `blc` format:

```
>>> jpredapi.submit(mode="msa", user_format="blc", file="tests/example_data/msa_blc.
↪example", email="name@domain.com")
>>>
```

Check job status on JPred server

Check single job status using `jobid`:

```
>>> import jpredapi
>>>
>>> jpredapi.status(jobid="jp_K46D05A")
>>>
```

Check single job status using `jobid` and retrieve results:

```
>>> jpredapi.status(jobid="jp_K46D05A", results_dir_path="jpred_sspred/results")
>>>
```

Check single job status using `jobid`, retrieve results, and decompress archive:

```
>>> jpredapi.status(jobid="jp_K46D05A", results_dir_path="jpred_sspred/results",␣
↪extract=True)
>>>
```

Retrieve results from JPred server

Retrieve results using `jobid`:

```
>>> import jpredapi
>>>
>>> jpredapi.get_results(jobid="jp_K46D05A", results_dir_path="jpred_sspred/results")
>>>
```

Retrieve results using `jobid` and decompress archive:

```
>>> jpredapi.get_results(jobid="jp_K46D05A", results_dir_path="jpred_sspred/results",␣
↪extract=True)
>>>
```

Check how many jobs you have already submitted on a given day:

```
>>> import jpredapi
>>>
>>> jpredapi.quota(email="name@domain.com")
>>>
```

1.3.3 The jpredapi API Reference

jpredapi Python library

The JPred API allows users to submit jobs from the command-line.

Usage example for command-line:

```
python3 -m jpredapi --help
python3 -m jpredapi --version
python3 -m jpredapi submit --mode=single --format=raw --
↪seq=MQVWPIEGIKKFETLSYLPP
python3 -m jpredapi status --jobid=jp_K46D05A
python3 -m jpredapi get_results --jobid=jp_K46D05A --results=jpred_sspred/
↪results
python3 -m jpredapi quota --email=name@domain.com
python3 -m jpredapi check_rest_version
```

Usage example for interactive Python shell:

```
>>> import jpredapi
>>>
>>> jpredapi.submit(mode="single", user_format="raw", seq="MQVWPIEGIKKFETLSYLPP")
>>>
>>> jpredapi.status(jobid="jp_K46D05A")
>>>
>>> jpredapi.get_results(jobid="jp_K46D05A", results_dir_path="jpred_sspred/results")
>>>
>>> jpredapi.quota(email="name@domain.com")
>>>
>>> jpredapi.check_rest_version()
>>>
```

jpredapi.api.**check_rest_version**(*host='http://www.compbio.dundee.ac.uk/jpred4/cgi-bin/rest'*, *suffix='version'*, *silent=False*)

Check version of JPred REST interface.

Parameters

- **host** (*str*) – JPred host address.
- **suffix** (*str*) – Host address suffix.

- **silent** (True or False) – Should the work be done silently?

Returns Version of JPred REST API.

Return type str

jpredapi.api.**quota**(*email*, *host='http://www.compbio.dundee.ac.uk/jpred4/cgi-bin/rest'*, *suffix='quota'*, *silent=False*)

Check how many jobs you have already submitted on a given day (out of 1000 maximum allowed jobs per user per day).

Parameters

- **email** (*str*) – E-mail address.
- **host** (*str*) – JPred host address.
- **suffix** (*str*) – Host address suffix.
- **silent** (True or False) – Should the work be done silently?

Returns Response.

Return type requests.Response

jpredapi.api.**submit**(*mode*, *user_format*, *file=None*, *seq=None*, *skipPDB=True*, *email=None*, *name=None*, *silent=False*, *host='http://www.compbio.dundee.ac.uk/jpred4/cgi-bin/rest'*)

Submit job to JPred REST API.

Parameters

- **mode** (*str*) – Submission mode, possible values: *single*, *batch*, *msa*.
- **user_format** (*str*) – Submission format, possible values: *raw*, *fasta*, *msf*, *blc*.
- **file** (*str*) – File path to a file with the job input (sequence or msa).
- **seq** (*str*) – Alternatively, amino acid sequence passed as string of single-letter code without spaces, e.g. –seq=ATWFGTHY
- **skipPDB** (True or False) – Should the PDB query be skipped?
- **email** (*str*) – For a batch job submission, where to send the results?
- **name** (*str*) – A name for the job.
- **silent** (True or False) – Should the work be done silently?

Returns Response.

Return type requests.Response

jpredapi.api.**status**(**args*, ***kw*)

Check status of the submitted job.

Parameters

- **jobid** (*str*) – Job id.
- **results_dir_path** (*str*) – Directory path where to save results if job is finished.
- **extract** (True or False) – Extract (True) or not (False) results into directory.
- **silent** (True or False) – Should the work be done silently?
- **host** (*str*) – JPred host address.
- **jpred4** (*str*) – JPred address for results retrieval.

Returns Response.

Return type requests.Response

jpredapi.api.**get_results**(*jobid*, *results_dir_path=None*, *extract=False*, *silent=False*, *host='http://www.compbio.dundee.ac.uk/jpred4/cgi-bin/rest'*, *jpred4='http://www.compbio.dundee.ac.uk/jpred4'*)

Download results from JPred server.

Parameters

- **jobid** (*str*) – Job id.
- **results_dir_path** (*str*) – Directory path where to save results if job is finished.
- **extract** (True or False) – Extract (True) or not (False) results into directory.
- **silent** (True or False) – Should the work be done silently?
- **host** (*str*) – JPred host address.
- **jpred4** (*str*) – JPred address for results retrieval.

Returns Response.

Return type requests.Response

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

j

Index