InterPro

biopython

InterProFetcher

Biopython package for accessing data deposited in InterPro

InterProFetcher

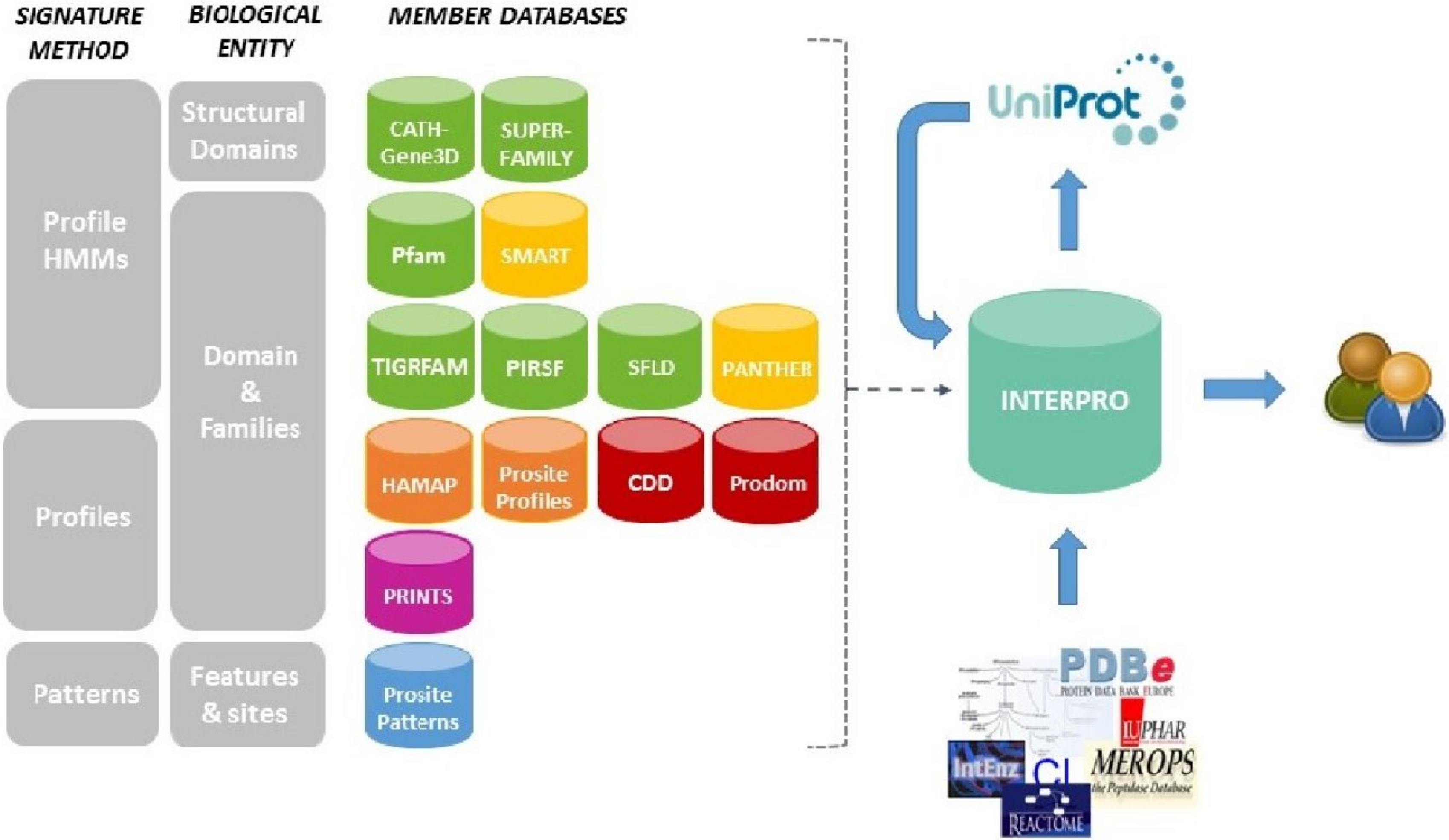
InterPro

- was established in **1999** by a research consortium comprising various institutions, including the European Bioinformatics Institute (EBI) in the United Kingdom, University College London, University of Manchester, and University of Cambridge.

- the aim was to integrate diverse sources of information on proteins and protein domains to provide a more comprehensive analysis and interpretation of protein sequences.

- Currently, it integrates data from multiple databases and tools such as **PROSITE, Pfam, PRINTS, PROSITE, SMART, SUPERFAMILY, and others**.

- it provides advanced tools for the analysis of protein domains, identification of conserved motifs, protein structure prediction, and other aspects related to protein function and evolution.



SIGNATURE METHOD

BIOLOGICAL ENTITY

MEMBER DATABASES

Profile HMMs

Profiles

Patterns

Structural Domains

Domain & Families

Features & sites

CATH-Gene3D

SUPER-FAMILY

Pfam

SMART

TIGRFAM

PIRSF

SFLD

PANTHER

HAMAP

Prosite Profiles

CDD

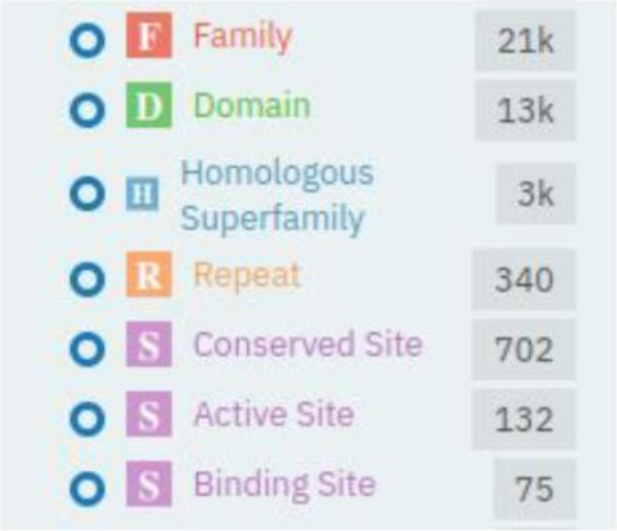
Prodom

PRINTS

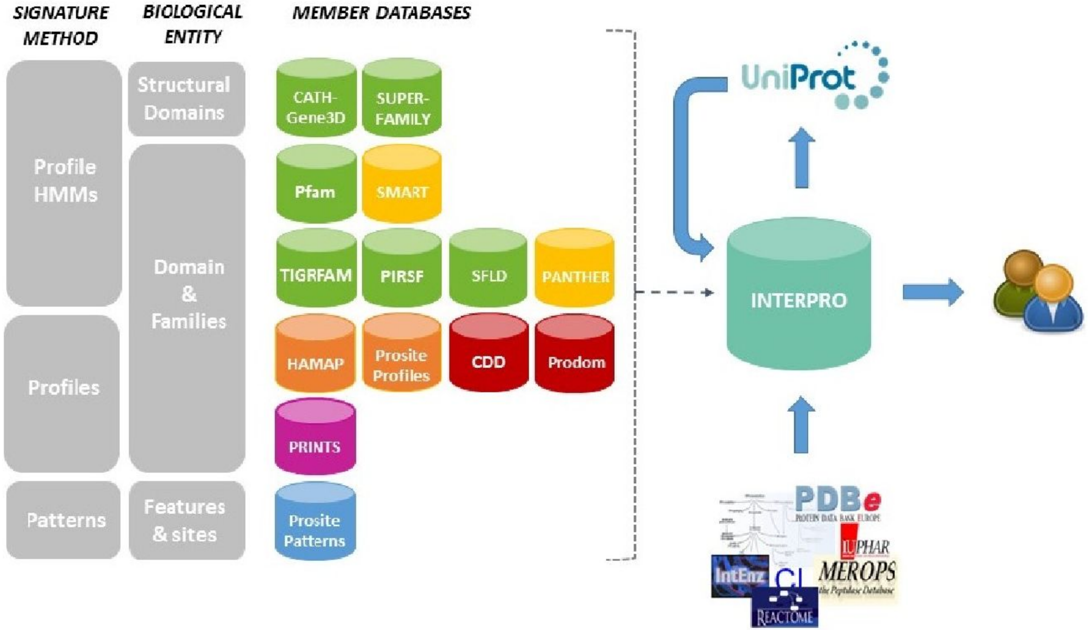
Prosite Patterns

INTERPRO

Types of data:

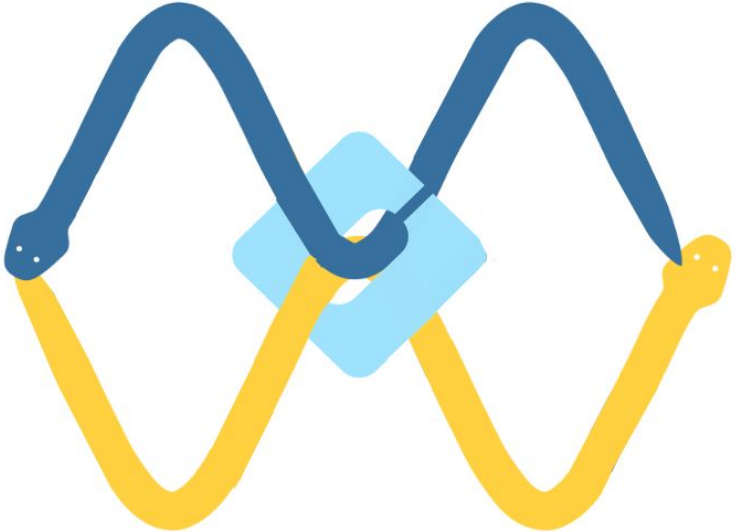


○	F Family	21k
○	D Domain	13k
○	H Homologous Superfamily	3k
○	R Repeat	340
○	S Conserved Site	702
○	S Active Site	132
○	S Binding Site	75



InterProFetcher

BioPython package for accessing data deposited in InterPro

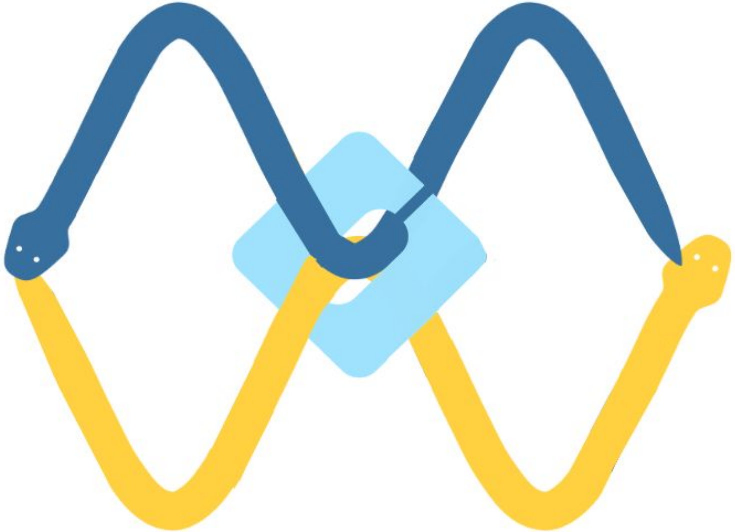


InterProFetcher

BioPython package for accessing data deposited in InterPro

Motivations:

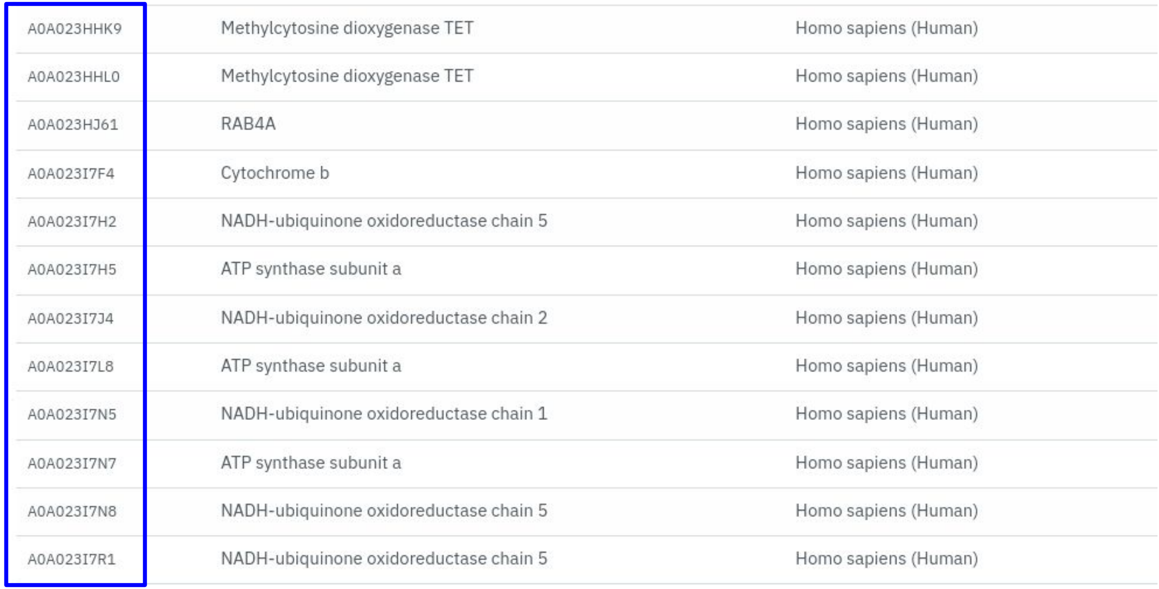
- InterPro is a comprehensive source as it has cross-references
- Complex queries and parameterization pose challenges
- Request limits and speed restrictions may affect usage
- Implementing a library like BioPython simplifies working with biological data



Ideas:

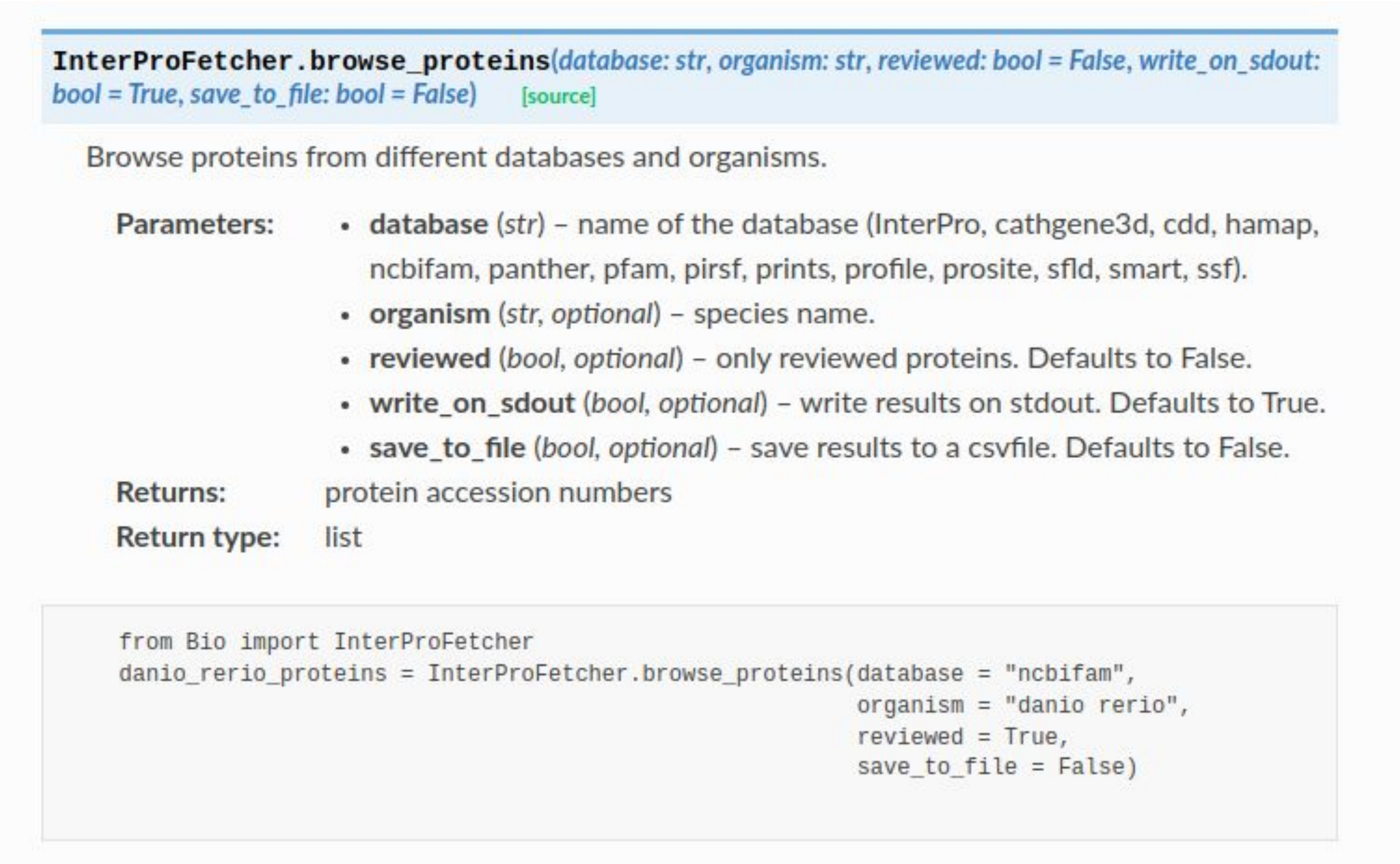
- browsing proteins by database and an organism

InterPro,
homo sapiens



A0A023HHK9	Methylcytosine dioxygenase TET	Homo sapiens (Human)
A0A023HHL0	Methylcytosine dioxygenase TET	Homo sapiens (Human)
A0A023HJ61	RAB4A	Homo sapiens (Human)
A0A023I7F4	Cytochrome b	Homo sapiens (Human)
A0A023I7H2	NADH-ubiquinone oxidoreductase chain 5	Homo sapiens (Human)
A0A023I7H5	ATP synthase subunit a	Homo sapiens (Human)
A0A023I7J4	NADH-ubiquinone oxidoreductase chain 2	Homo sapiens (Human)
A0A023I7L8	ATP synthase subunit a	Homo sapiens (Human)
A0A023I7N5	NADH-ubiquinone oxidoreductase chain 1	Homo sapiens (Human)
A0A023I7N7	ATP synthase subunit a	Homo sapiens (Human)
A0A023I7N8	NADH-ubiquinone oxidoreductase chain 5	Homo sapiens (Human)
A0A023I7R1	NADH-ubiquinone oxidoreductase chain 5	Homo sapiens (Human)

...



InterProFetcher.browse_proteins(*database: str, organism: str, reviewed: bool = False, write_on_sdout: bool = True, save_to_file: bool = False*) [source]

Browse proteins from different databases and organisms.

Parameters:

- **database** (*str*) – name of the database (InterPro, cathgene3d, cdd, hamap, ncbifam, panther, pfam, pirsf, prints, profile, prosite, sfld, smart, ssf).
- **organism** (*str, optional*) – species name.
- **reviewed** (*bool, optional*) – only reviewed proteins. Defaults to False.
- **write_on_sdout** (*bool, optional*) – write results on stdout. Defaults to True.
- **save_to_file** (*bool, optional*) – save results to a csvfile. Defaults to False.

Returns: protein accession numbers

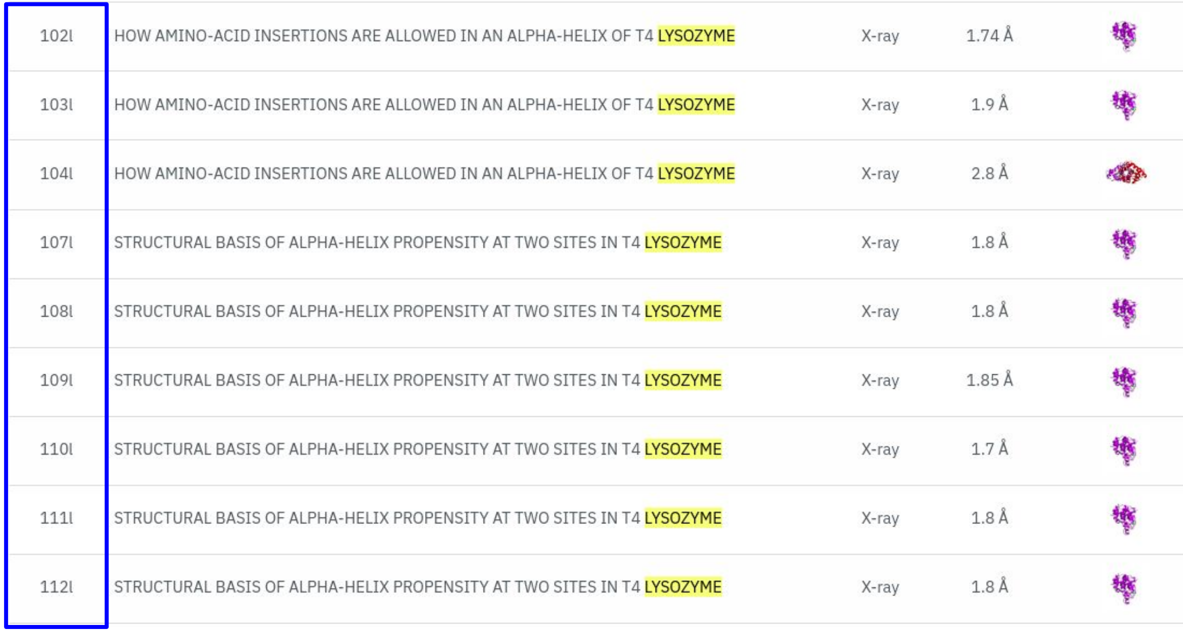
Return type: list

```
from Bio import InterProFetcher
danio_rerio_proteins = InterProFetcher.browse_proteins(database = "ncbifam",
                                                       organism = "danio rerio",
                                                       reviewed = True,
                                                       save_to_file = False)
```

Ideas:

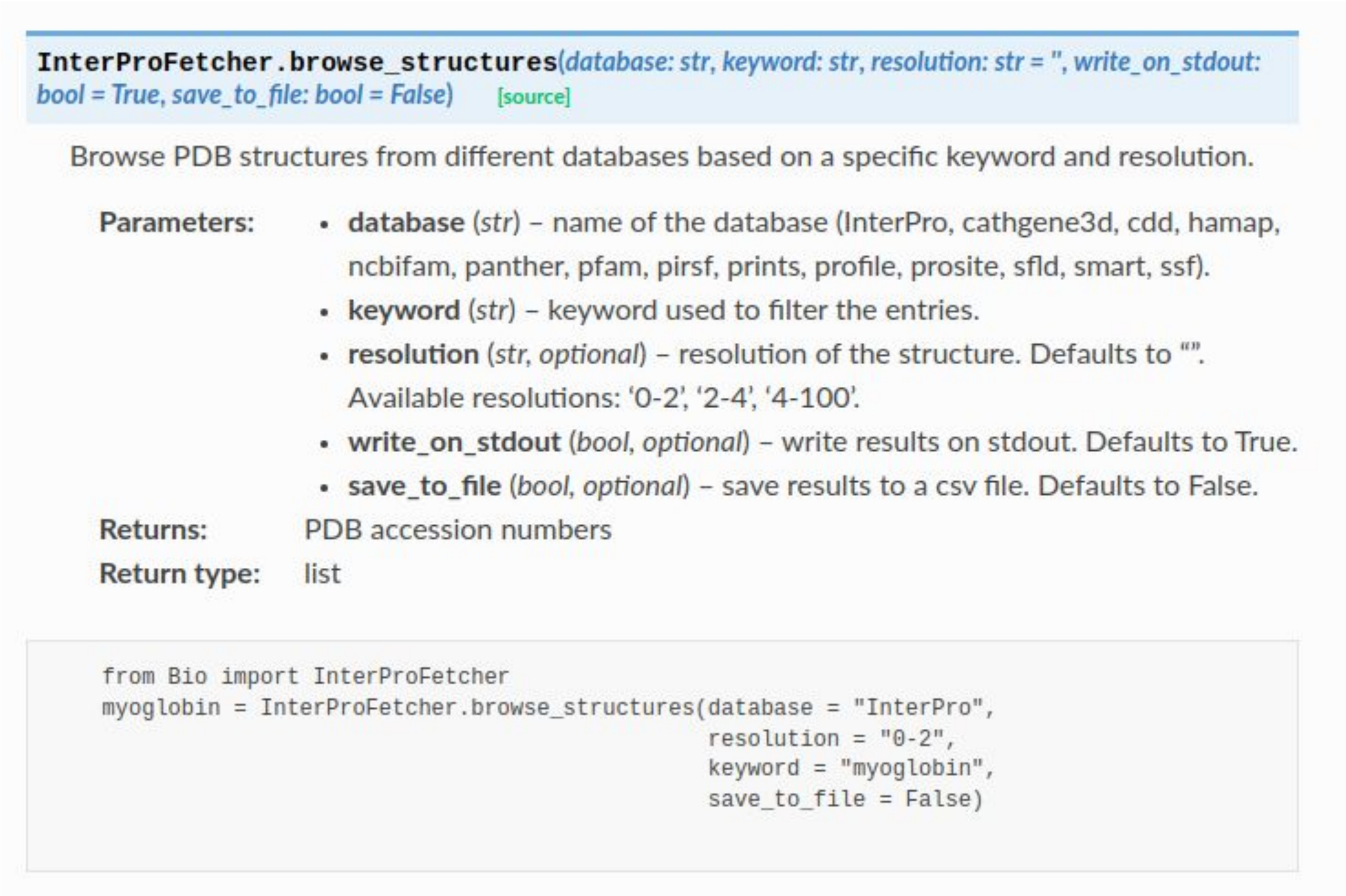
- browsing protein structures by a database and a keyword + downloading them from PDB

InterPro, lysozyme



102l	HOW AMINO-ACID INSERTIONS ARE ALLOWED IN AN ALPHA-HELIX OF T4 LYSOZYME	X-ray	1.74 Å
103l	HOW AMINO-ACID INSERTIONS ARE ALLOWED IN AN ALPHA-HELIX OF T4 LYSOZYME	X-ray	1.9 Å
104l	HOW AMINO-ACID INSERTIONS ARE ALLOWED IN AN ALPHA-HELIX OF T4 LYSOZYME	X-ray	2.8 Å
107l	STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO SITES IN T4 LYSOZYME	X-ray	1.8 Å
108l	STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO SITES IN T4 LYSOZYME	X-ray	1.8 Å
109l	STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO SITES IN T4 LYSOZYME	X-ray	1.85 Å
110l	STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO SITES IN T4 LYSOZYME	X-ray	1.7 Å
111l	STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO SITES IN T4 LYSOZYME	X-ray	1.8 Å
112l	STRUCTURAL BASIS OF ALPHA-HELIX PROPENSITY AT TWO SITES IN T4 LYSOZYME	X-ray	1.8 Å

...



`InterProFetcher.browse_structures`(*database: str, keyword: str, resolution: str = '', write_on_stdout: bool = True, save_to_file: bool = False*) [source]

Browse PDB structures from different databases based on a specific keyword and resolution.

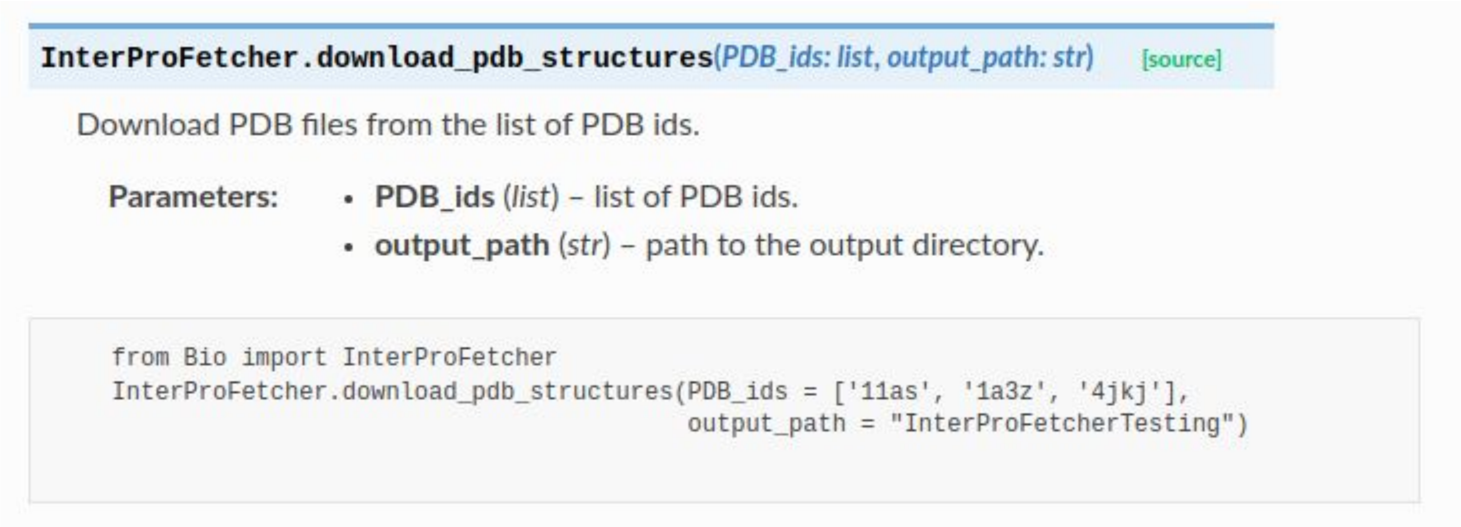
Parameters:

- **database** (*str*) – name of the database (InterPro, cathgene3d, cdd, hamap, ncbifam, panther, pfam, pirsf, prints, profile, prosite, sfld, smart, ssf).
- **keyword** (*str*) – keyword used to filter the entries.
- **resolution** (*str, optional*) – resolution of the structure. Defaults to “”. Available resolutions: ‘0-2’, ‘2-4’, ‘4-100’.
- **write_on_stdout** (*bool, optional*) – write results on stdout. Defaults to True.
- **save_to_file** (*bool, optional*) – save results to a csv file. Defaults to False.

Returns: PDB accession numbers

Return type: list

```
from Bio import InterProFetcher
myoglobin = InterProFetcher.browse_structures(database = "InterPro",
                                              resolution = "0-2",
                                              keyword = "myoglobin",
                                              save_to_file = False)
```



InterProFetcher.download_pdb_structures(*PDB_ids: list, output_path: str*) [source]

Download PDB files from the list of PDB ids.

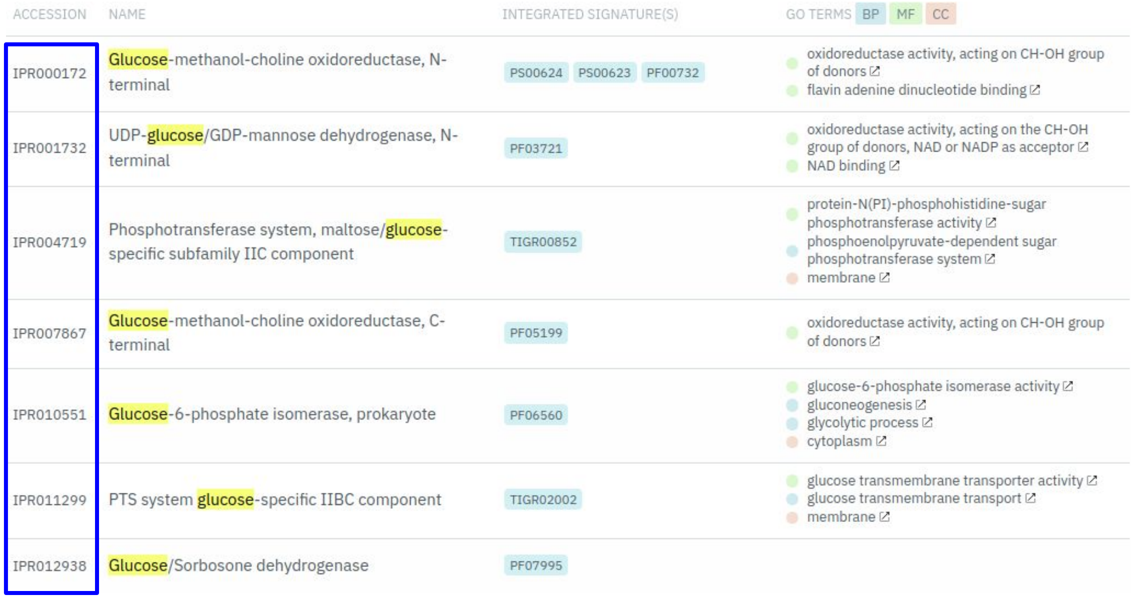
Parameters:
- **PDB_ids** (*list*) – list of PDB ids.
- **output_path** (*str*) – path to the output directory.

```
from Bio import InterProFetcher
InterProFetcher.download_pdb_structures(PDB_ids = ['11as', '1a3z', '4jkj'],
                                        output_path = "InterProFetcherTesting")
```

Ideas:

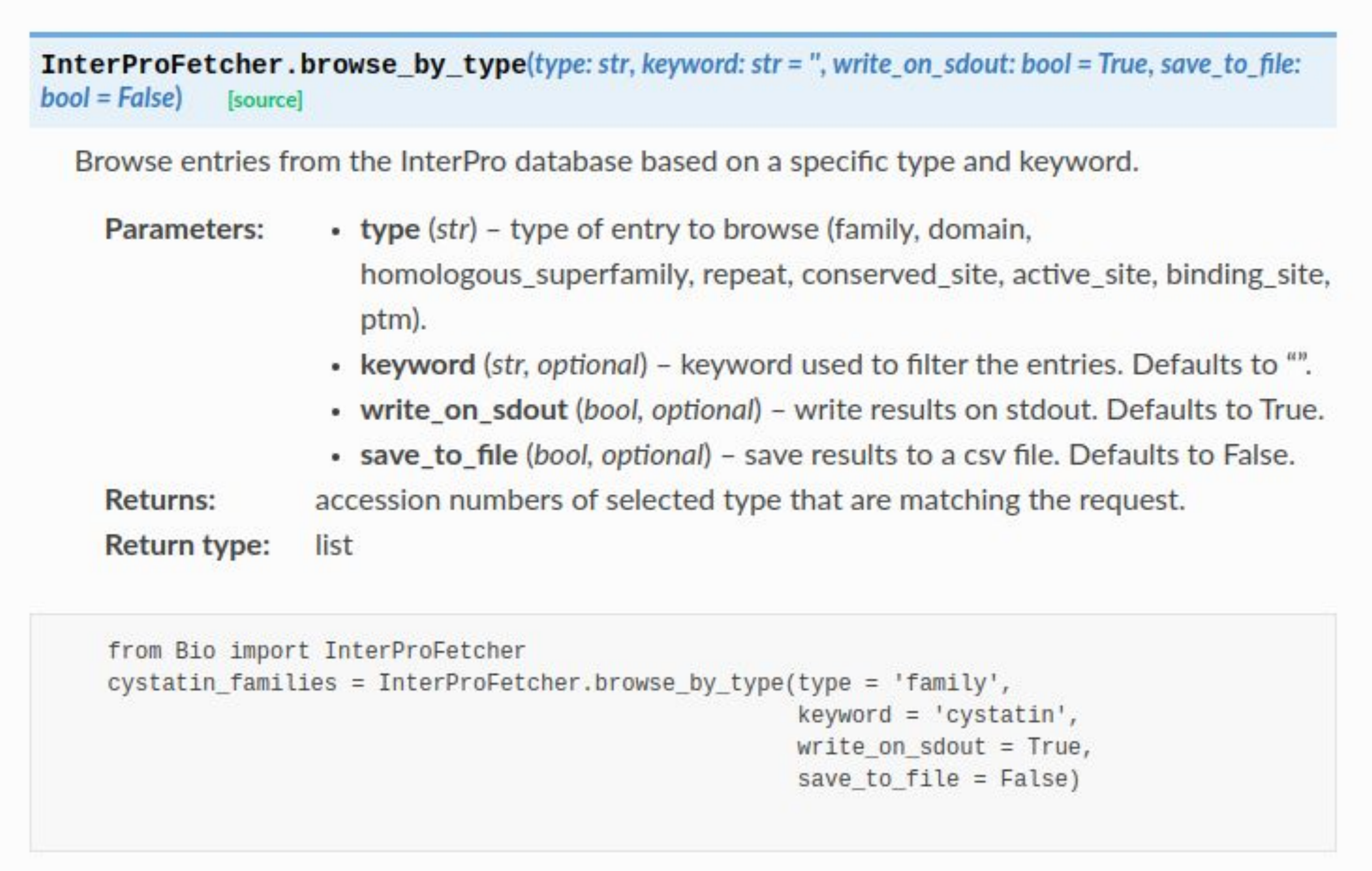
- browsing InterPro database by a keyword and type (e.g. domain)

domain, glucose



ACCESSION	NAME	INTEGRATED SIGNATURE(S)	GO TERMS BP MF CC
IPR000172	Glucose-methanol-choline oxidoreductase, N-terminal	PS00624 PS00623 PF00732	oxidoreductase activity, acting on CH-OH group of donors flavin adenine dinucleotide binding
IPR001732	UDP-glucose/GDP-mannose dehydrogenase, N-terminal	PF03721	oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as acceptor NAD binding
IPR004719	Phosphotransferase system, maltose/glucose-specific subfamily IIC component	TIGR00852	protein-N(PI)-phosphohistidine-sugar phosphotransferase activity phosphoenolpyruvate-dependent sugar phosphotransferase system membrane
IPR007867	Glucose-methanol-choline oxidoreductase, C-terminal	PF05199	oxidoreductase activity, acting on CH-OH group of donors
IPR010551	Glucose-6-phosphate isomerase, prokaryote	PF06560	glucose-6-phosphate isomerase activity gluconeogenesis glycolytic process cytoplasm
IPR011299	PTS system glucose-specific IIBC component	TIGR02002	glucose transmembrane transporter activity glucose transmembrane transport membrane
IPR012938	Glucose/Sorbosone dehydrogenase	PF07995	

...



InterProFetcher.browse_by_type(*type: str, keyword: str = '', write_on_sdout: bool = True, save_to_file: bool = False*) [source]

Browse entries from the InterPro database based on a specific type and keyword.

Parameters:

- **type** (*str*) – type of entry to browse (family, domain, homologous_superfamily, repeat, conserved_site, active_site, binding_site, ptm).
- **keyword** (*str, optional*) – keyword used to filter the entries. Defaults to “”.
- **write_on_sdout** (*bool, optional*) – write results on stdout. Defaults to True.
- **save_to_file** (*bool, optional*) – save results to a csv file. Defaults to False.

Returns: accession numbers of selected type that are matching the request.

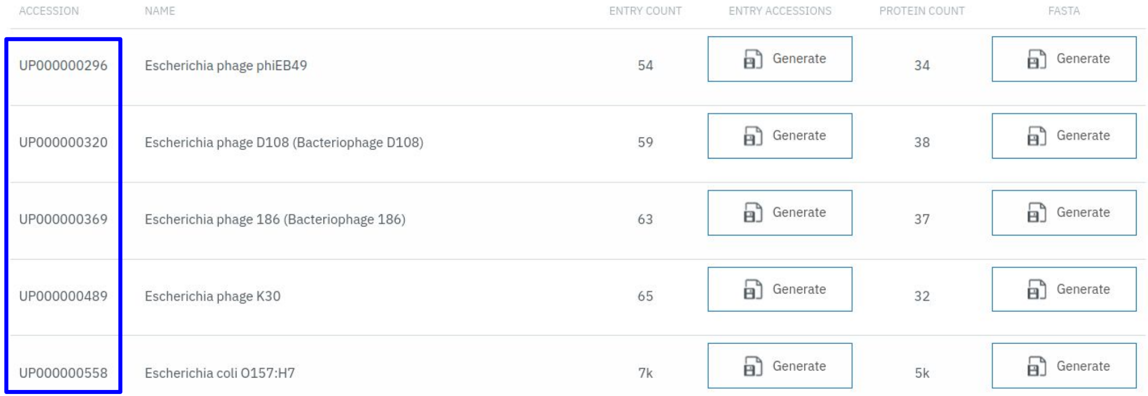
Return type: list

```
from Bio import InterProFetcher
cystatin_families = InterProFetcher.browse_by_type(type = 'family',
                                                   keyword = 'cystatin',
                                                   write_on_sdout = True,
                                                   save_to_file = False)
```

Ideas:

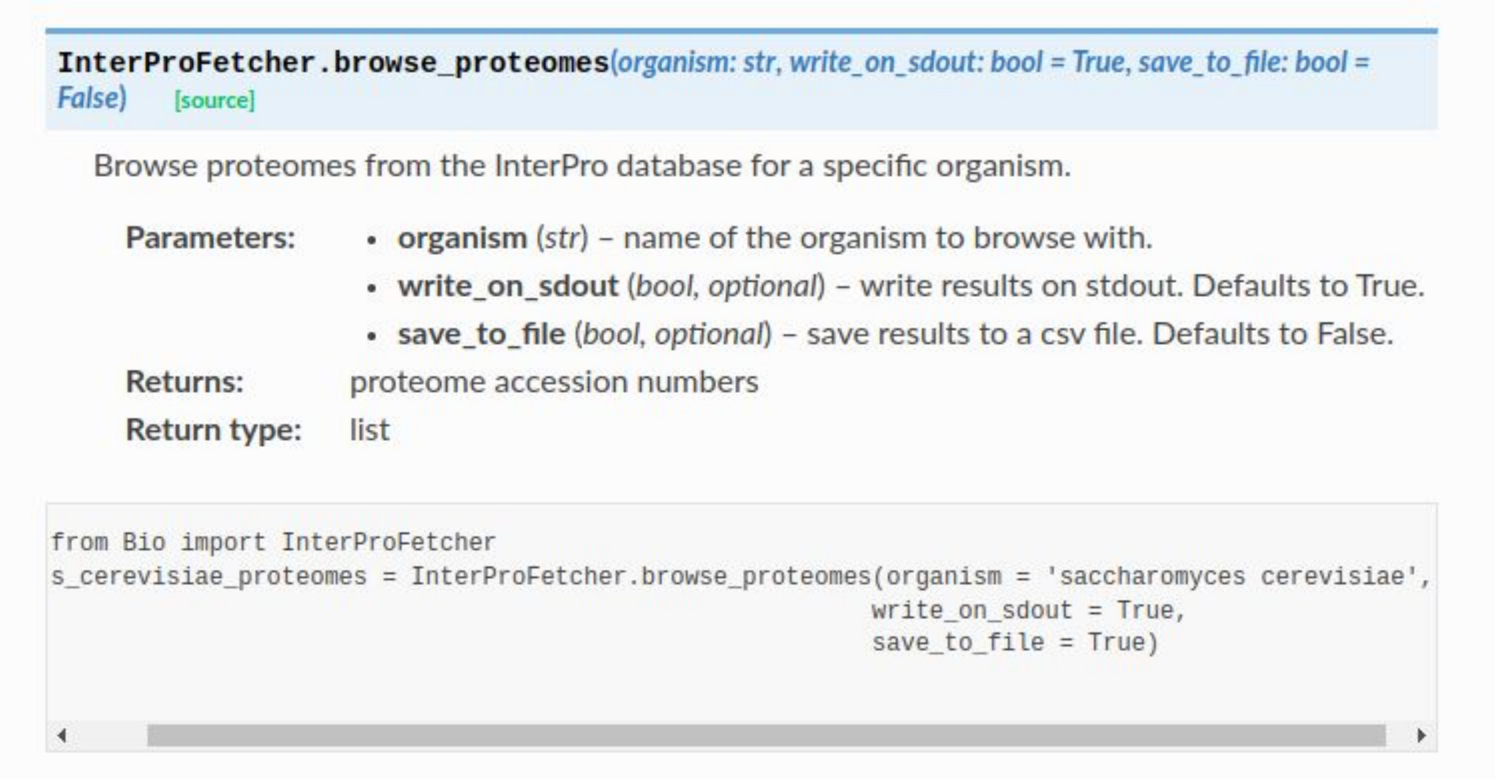
- browsing proteomes by an organism

Escherichia coli



ACCESSION	NAME	ENTRY COUNT	ENTRY ACCESSIONS	PROTEIN COUNT	FASTA
UP000000296	Escherichia phage phiEB49	54	Generate	34	Generate
UP000000320	Escherichia phage D108 (Bacteriophage D108)	59	Generate	38	Generate
UP000000369	Escherichia phage 186 (Bacteriophage 186)	63	Generate	37	Generate
UP000000489	Escherichia phage K30	65	Generate	32	Generate
UP000000558	Escherichia coli O157:H7	7k	Generate	5k	Generate

...



InterProFetcher.browse_proteomes(*organism: str, write_on_sdout: bool = True, save_to_file: bool = False*) [source]

Browse proteomes from the InterPro database for a specific organism.

Parameters:

- **organism** (*str*) – name of the organism to browse with.
- **write_on_sdout** (*bool, optional*) – write results on stdout. Defaults to True.
- **save_to_file** (*bool, optional*) – save results to a csv file. Defaults to False.

Returns: proteome accession numbers

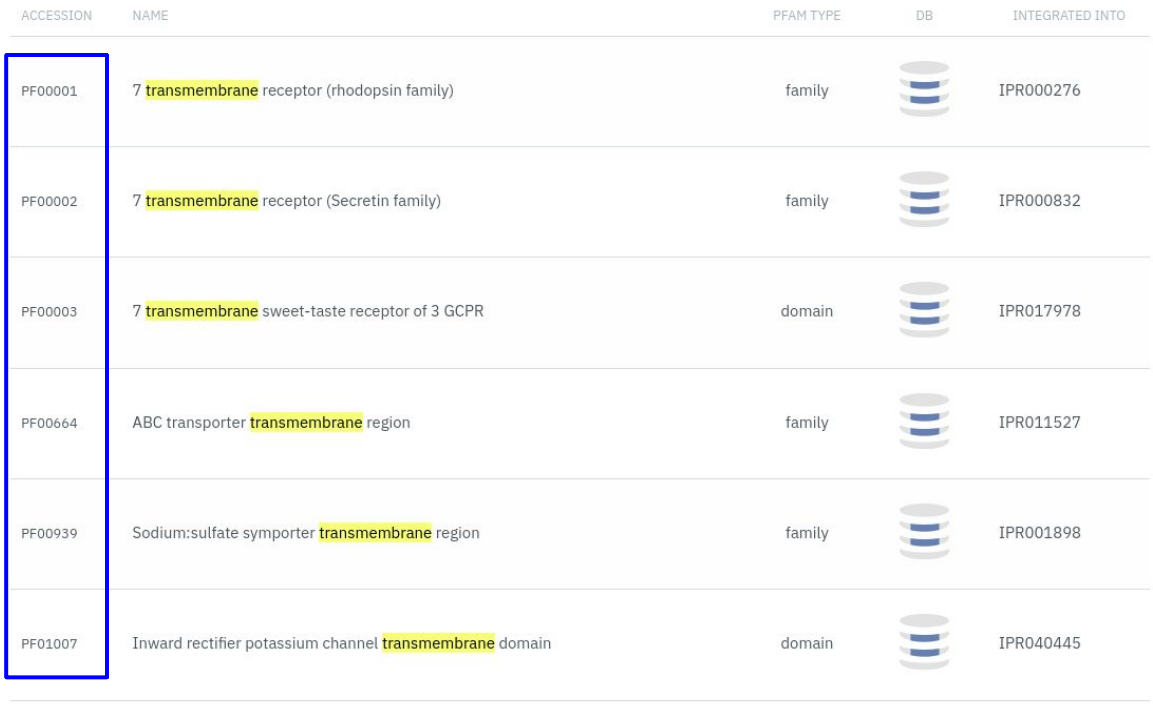
Return type: list

```
from Bio import InterProFetcher
s_cerevisiae_proteomes = InterProFetcher.browse_proteomes(organism = 'saccharomyces cerevisiae',
                                                         write_on_sdout = True,
                                                         save_to_file = True)
```

Ideas:

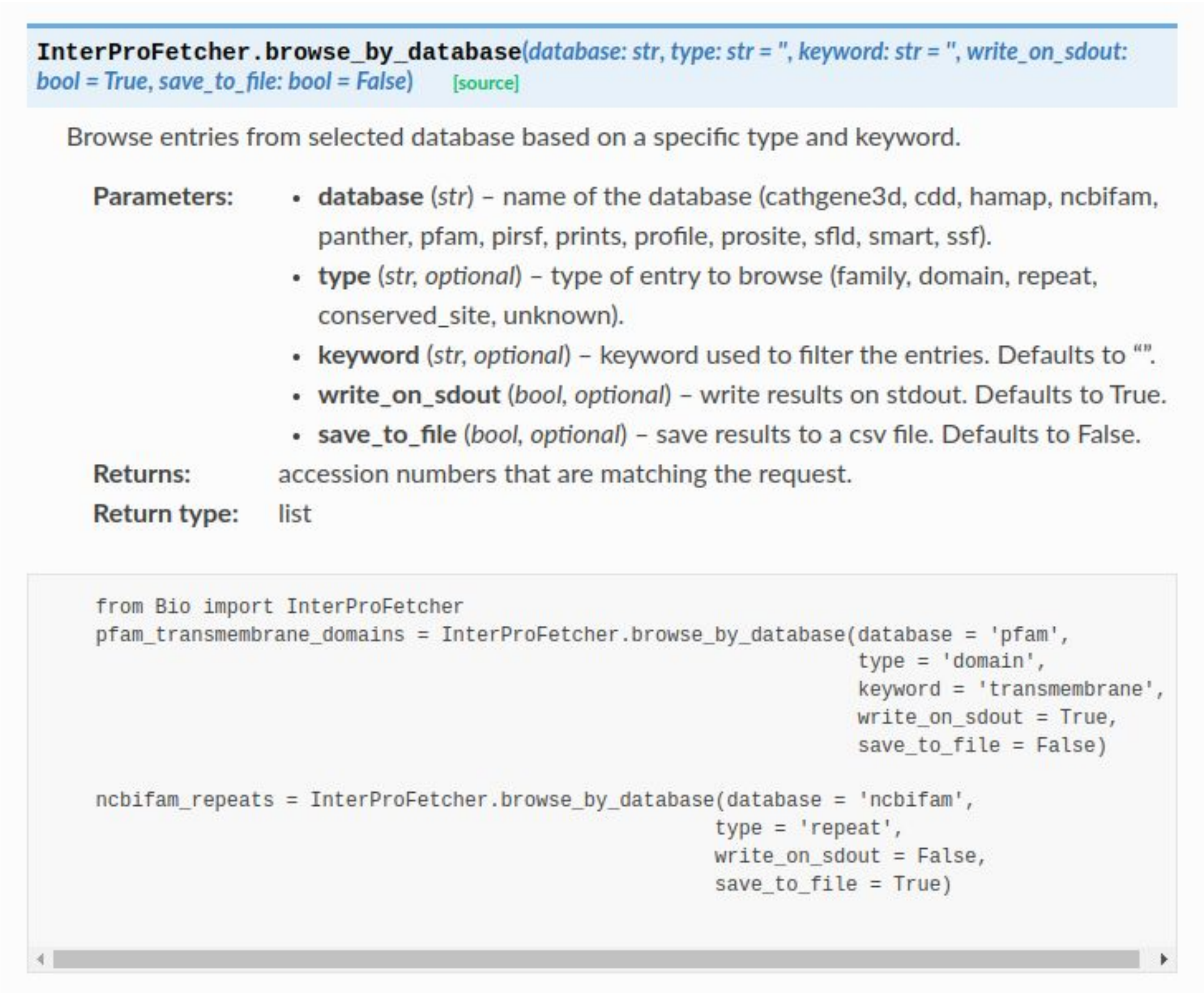
- browsing a database by an entry type and a keyword

Pfam, transmembrane



ACCESSION	NAME	PFAM TYPE	DB	INTEGRATED INTO
PF00001	7 transmembrane receptor (rhodopsin family)	family	🛢	IPR000276
PF00002	7 transmembrane receptor (Secretin family)	family	🛢	IPR000832
PF00003	7 transmembrane sweet-taste receptor of 3 GCPR	domain	🛢	IPR017978
PF00664	ABC transporter transmembrane region	family	🛢	IPR011527
PF00939	Sodium:sulfate symporter transmembrane region	family	🛢	IPR001898
PF01007	Inward rectifier potassium channel transmembrane domain	domain	🛢	IPR040445

...



InterProFetcher.browse_by_database(*database: str, type: str = '', keyword: str = '', write_on_sdout: bool = True, save_to_file: bool = False*) [source]

Browse entries from selected database based on a specific type and keyword.

Parameters:

- **database** (*str*) – name of the database (cathgene3d, cdd, hamap, ncbifam, panther, pfam, pirsf, prints, profile, prosite, sfld, smart, ssf).
- **type** (*str, optional*) – type of entry to browse (family, domain, repeat, conserved_site, unknown).
- **keyword** (*str, optional*) – keyword used to filter the entries. Defaults to “”.
- **write_on_sdout** (*bool, optional*) – write results on stdout. Defaults to True.
- **save_to_file** (*bool, optional*) – save results to a csv file. Defaults to False.

Returns: accession numbers that are matching the request.

Return type: list

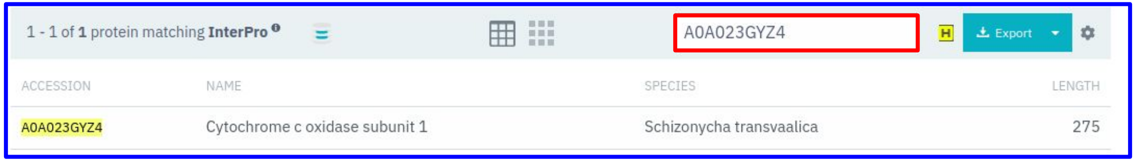
```
from Bio import InterProFetcher
pfam_transmembrane_domains = InterProFetcher.browse_by_database(database = 'pfam',
                                                                type = 'domain',
                                                                keyword = 'transmembrane',
                                                                write_on_sdout = True,
                                                                save_to_file = False)

ncbifam_repeats = InterProFetcher.browse_by_database(database = 'ncbifam',
                                                    type = 'repeat',
                                                    write_on_sdout = False,
                                                    save_to_file = True)
```

Ideas:

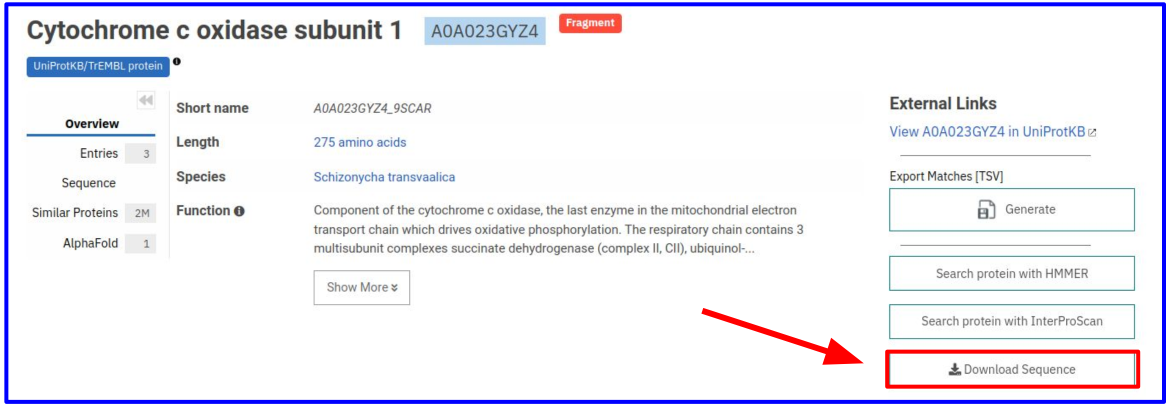
- downloading protein sequences by UniProt accession number

A0A023GYZ4



1 - 1 of 1 protein matching InterPro

ACCESSION	NAME	SPECIES	LENGTH
A0A023GYZ4	Cytochrome c oxidase subunit 1	Schizonycha transvaalica	275



Cytochrome c oxidase subunit 1 A0A023GYZ4 Fragment

UniProtKB/TrEMBL protein

Short name A0A023GYZ4_9SCAR

Length 275 amino acids

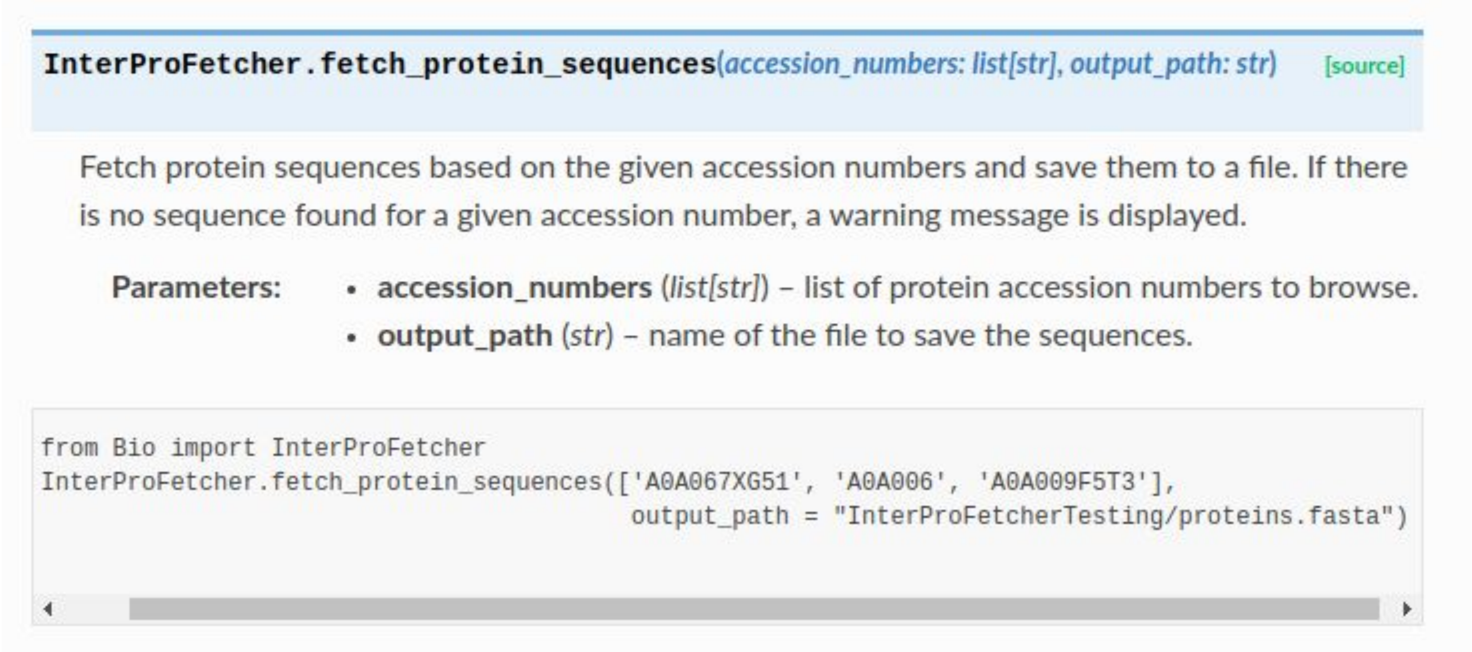
Species Schizonycha transvaalica

Function Component of the cytochrome c oxidase, the last enzyme in the mitochondrial electron transport chain which drives oxidative phosphorylation. The respiratory chain contains 3 multisubunit complexes succinate dehydrogenase (complex II, CII), ubiquinol-...

External Links

View A0A023GYZ4 in UniProtKB

Export Matches [TSV]



InterProFetcher.fetch_protein_sequences(*accession_numbers: list[str], output_path: str*) [source]

Fetch protein sequences based on the given accession numbers and save them to a file. If there is no sequence found for a given accession number, a warning message is displayed.

Parameters:

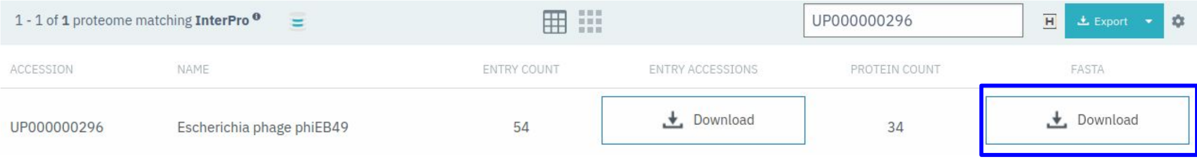
- **accession_numbers** (*list[str]*) – list of protein accession numbers to browse.
- **output_path** (*str*) – name of the file to save the sequences.

```
from Bio import InterProFetcher
InterProFetcher.fetch_protein_sequences(['A0A067XG51', 'A0A006', 'A0A009F5T3'],
                                        output_path = "InterProFetcherTesting/proteins.fasta")
```

Ideas:

- downloading proteomes by InterPro proteome IDs (sequences)

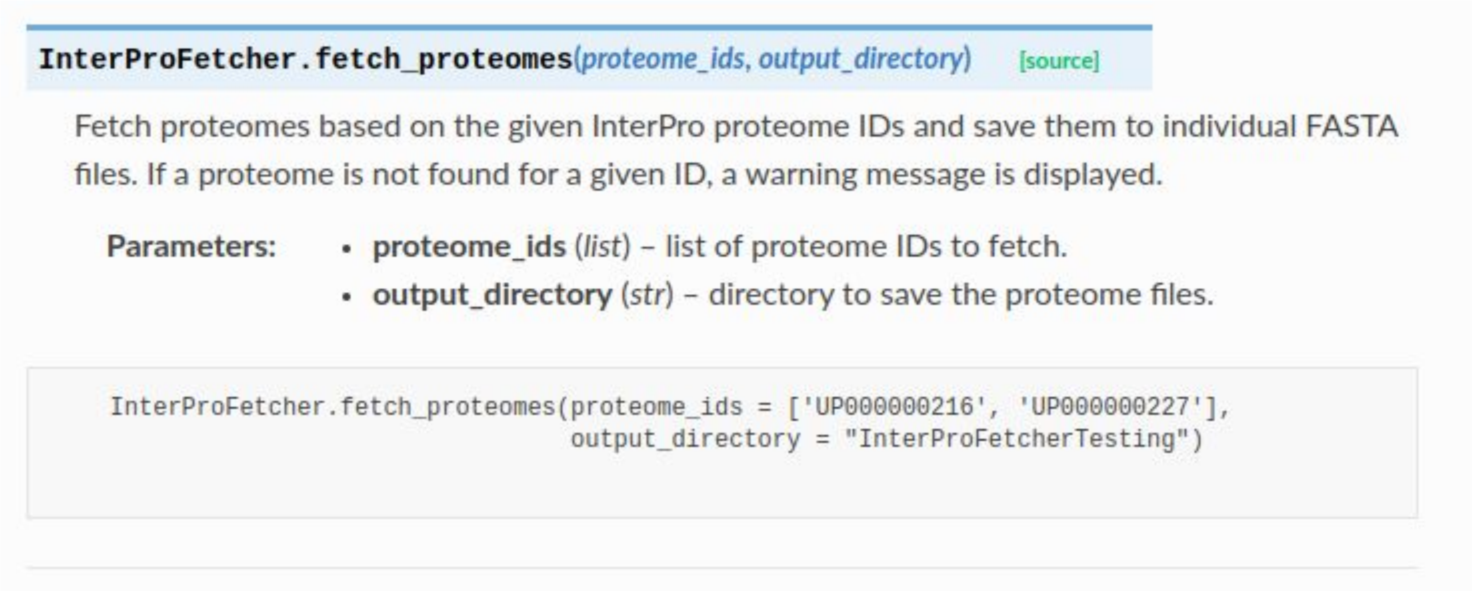
UP000000296



1 - 1 of **1** proteome matching **InterPro**

UP000000296

ACCESSION	NAME	ENTRY COUNT	ENTRY ACCESSIONS	PROTEIN COUNT	FASTA
UP000000296	Escherichia phage phiEB49	54	Download	34	Download



InterProFetcher.fetch_proteomes(*proteome_ids*, *output_directory*) [source]

Fetch proteomes based on the given InterPro proteome IDs and save them to individual FASTA files. If a proteome is not found for a given ID, a warning message is displayed.

Parameters:

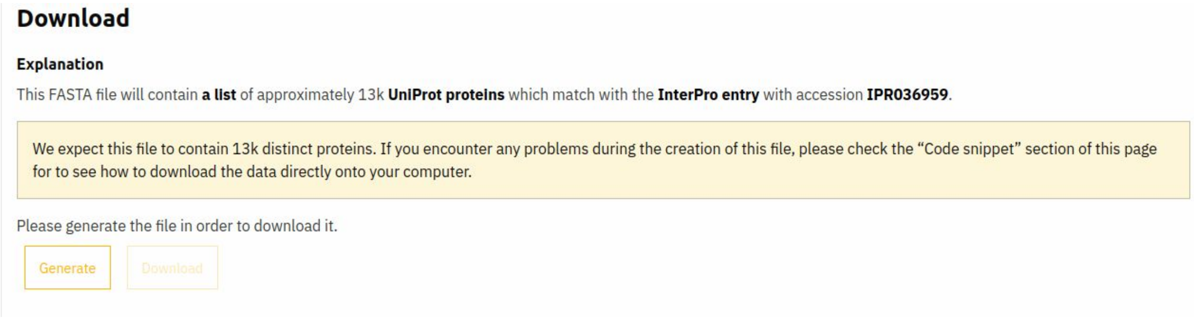
- **proteome_ids** (*list*) – list of proteome IDs to fetch.
- **output_directory** (*str*) – directory to save the proteome files.

```
InterProFetcher.fetch_proteomes(proteome_ids = ['UP000000216', 'UP000000227'],
                                output_directory = "InterProFetcherTesting")
```

Ideas:

- downloading sequences for families/superfamilies and others by ID

IPR036959



Download

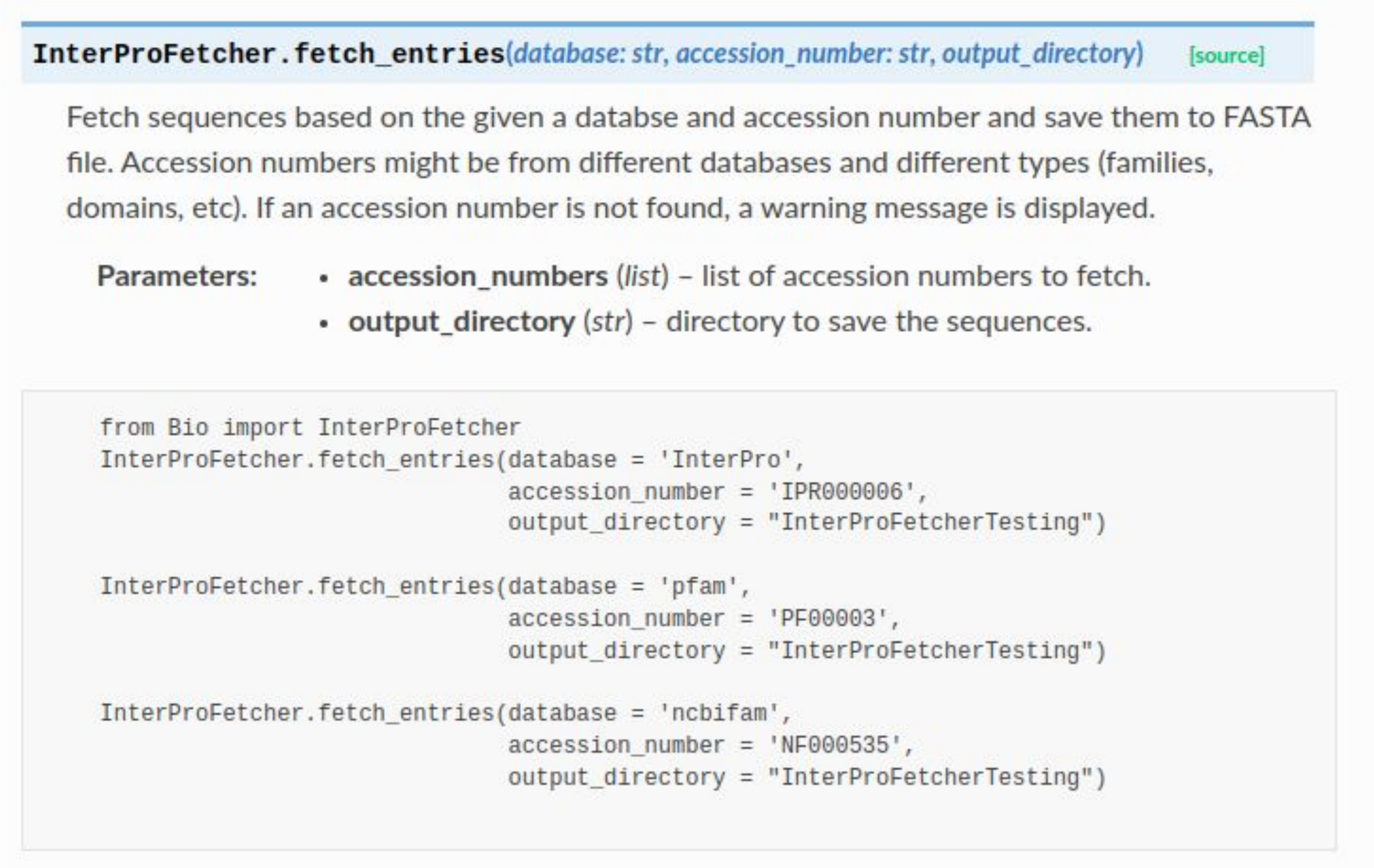
Explanation

This FASTA file will contain **a list** of approximately 13k **UniProt proteins** which match with the **InterPro entry** with accession **IPR036959**.

We expect this file to contain 13k distinct proteins. If you encounter any problems during the creation of this file, please check the “Code snippet” section of this page for to see how to download the data directly onto your computer.

Please generate the file in order to download it.

- online generation and downloading takes a long time and sometimes fails
- other option is to download a script



InterProFetcher.fetch_entries(*database: str, accession_number: str, output_directory*) [source]

Fetch sequences based on the given a databse and accession number and save them to FASTA file. Accession numbers might be from different databases and different types (families, domains, etc). If an accession number is not found, a warning message is displayed.

Parameters:

- **accession_numbers** (*list*) – list of accession numbers to fetch.
- **output_directory** (*str*) – directory to save the sequences.

```
from Bio import InterProFetcher
InterProFetcher.fetch_entries(database = 'InterPro',
                              accession_number = 'IPR000006',
                              output_directory = "InterProFetcherTesting")

InterProFetcher.fetch_entries(database = 'pfam',
                              accession_number = 'PF00003',
                              output_directory = "InterProFetcherTesting")

InterProFetcher.fetch_entries(database = 'ncbifam',
                              accession_number = 'NF000535',
                              output_directory = "InterProFetcherTesting")
```

Thank you for your attention!

Izabela Fedorczyk, Roksana Malinowska, Weronika Trawińska