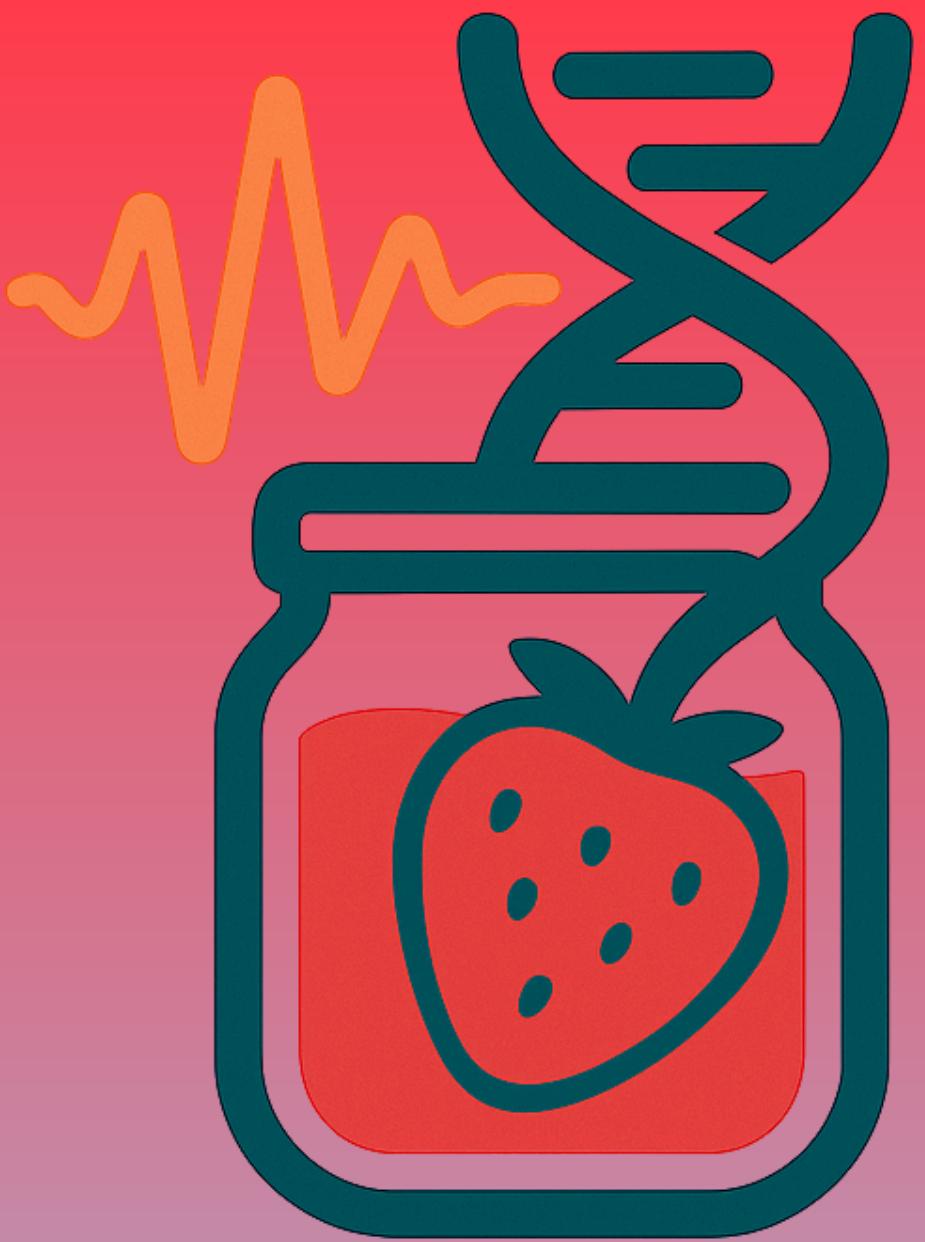


JAMSFetch

Joint Automated Multi-source Sequence Fetcher



Joanna Dąbrowska
Anna Szymik
Michał Stanowski
Stanisław Gołębiewski

Architecture of large projects in bioinformatics 2025



JAMSFetch

INTRODUCTION

GOALS

- O1** Automate the retrieval of sequence and structure data from major bioinformatics databases, reducing manual effort
- O2** Create a unified Python interface that intelligently handles multiple data sources based on ID type
- O3** Simplify bioinformatics workflows by automatically organizing downloaded data into standard formats and user-defined directories for easy access and management.



WHY?

O1

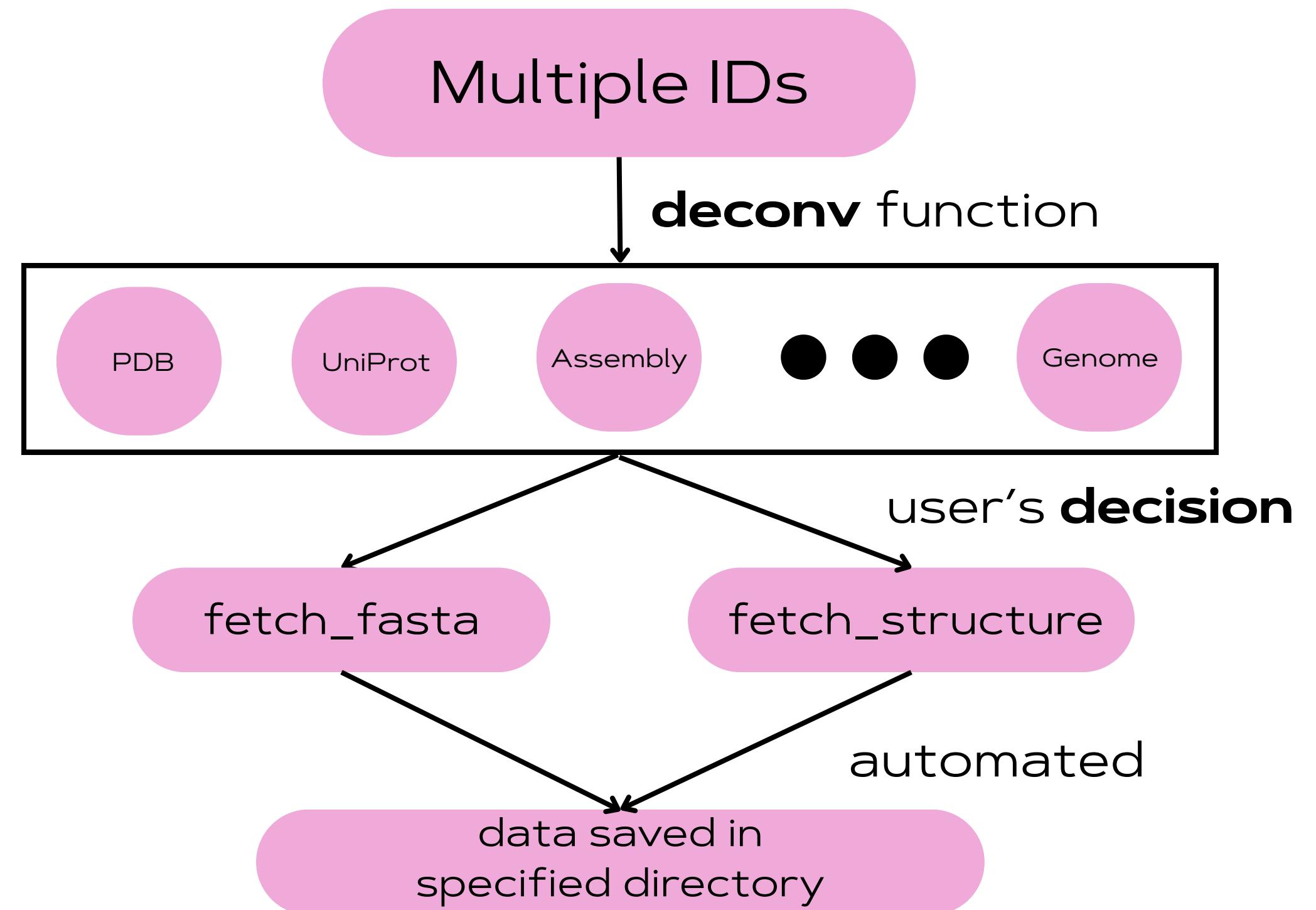
Managing and downloading biological sequence and structure data from multiple databases manually is time-consuming and error-prone, so an automated tool greatly improves efficiency and accuracy.

O2

Researchers need a streamlined, unified solution to easily access diverse data sources without dealing with different formats and interfaces, enabling faster and more reproducible analyses.

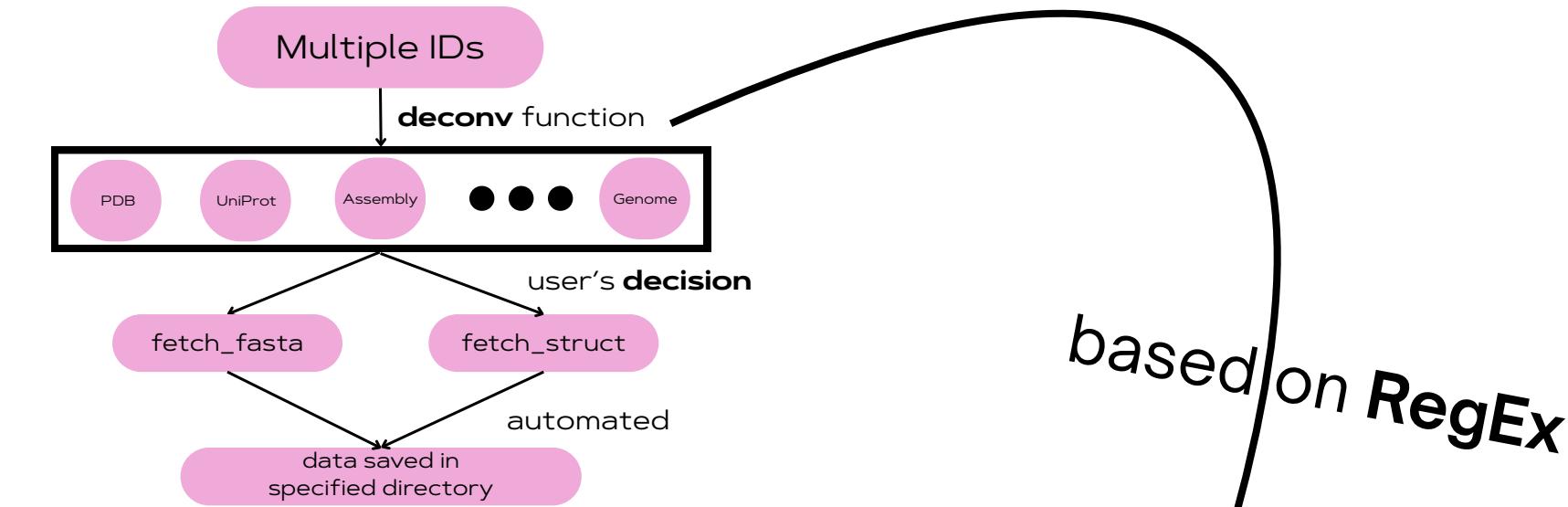
JAMSFetch

Design of the tool



JAMSFetch

Deconvolute function



Nucleotide

```
?:[A-Z]{2}_[0-9]+(?:\.[0-9]+)?|(?:[A-Z]{1,2}[0-9]{5,6}(?:\.[0-9]+)?|
```

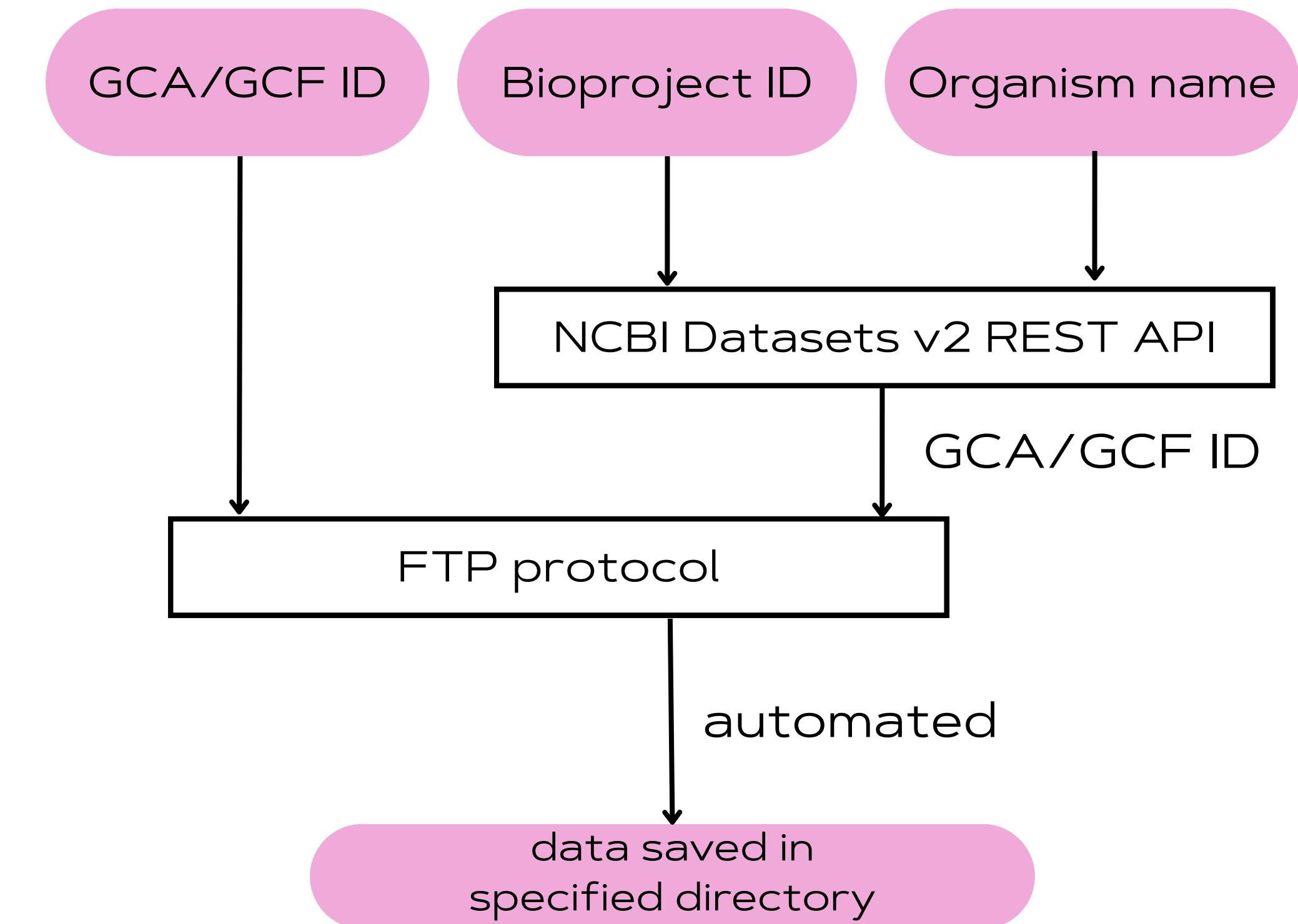
UniProt

```
[OPQ][0-9][A-Z0-9]{3}[0-9]|([A-NR-Z][0-9](?:[A-Z][A-Z0-9]{2}[0-9])?{1,2}
```

PDB

```
A-Za-z0-9{4}
```

JAMSFETCH. UTILS. GET_ASSEMBLY



JAMSFetch

Usage example – fetching sequences/structures from **unspecified** databases

```
>>> from jamsfetch import fetch_fasta
>>> ids = ["P12345", "NM_001200.2", "GCF_000006945"]
... fetch_fasta(
...     id_list=ids,
...     output_dir="downloads/",
...     assembly_data_type="genomic" # or "protein"
... )
...

[ Sorted identifiers by category:

Uniprot (1):
- P12345

Pdb (0):

Nucleotide (1):
- NM_001200.2

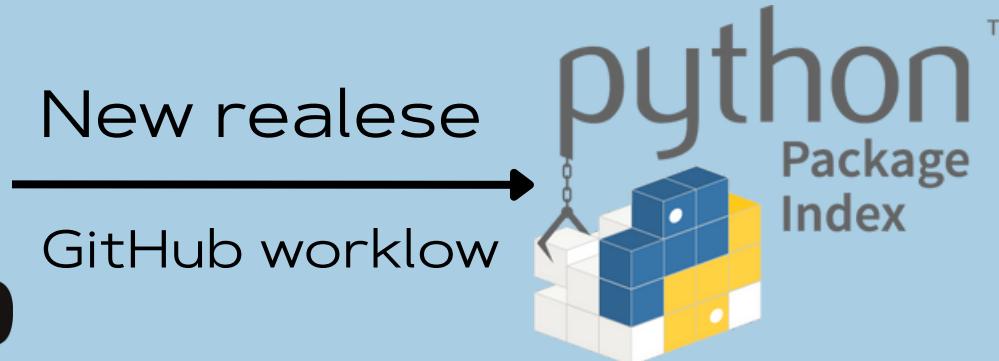
Assembly (1):
- GCF_000006945
→ Downloading amino acid sequences from Uniprot for: P12345
Saved: downloads/P12345.fasta
→ Downloading nucleotide sequences from NCBI:nucleotide for: NM_001200.2
→ Downloading genome from Genome Assembly for: GCF_000006945
--2025-06-09 18:22:41-- https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/006/945/GCF_000006945.2_ASM694v2/GCF_000006945.2_ASM694v2_genomic.fna.gz
Loaded CA certificate '/etc/ssl/certs/ca-certificates.crt'
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.31, 130.14.250.10, 130.14.250.7, ...
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.31|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1471540 (1.4M) [application/x-gzip]
Saving to: 'downloads/GCF_000006945.2_ASM694v2_genomic.fna.gz'

GCF_000006945.2_ASM694v2_genomic.f 100%[=====] 1,40M 312KB/s in 4,8s

2025-06-09 18:22:46 (302 KB/s) - 'downloads/GCF_000006945.2_ASM694v2_genomic.fna.gz' saved [1471540/1471540]

✓ Finished fetching all FASTA sequences.
```

Back-end



```
platform linux -- Python 3.12.3, pytest-8.4.0, pluggy-1.6.0
rootdir: /home/lambi/Desktop/studia/ArchProj/JAMS
configfile: pyproject.toml
collected 15 items

tests/test_alphaFold.py ...
tests/test_assembly.py ...
tests/test_auto.py ..
tests/test_deconvolute.py .
tests/test_esm.py ..
tests/test_nucleotide.py ...
tests/test_pdb.py ..
tests/test_uniprot.py ..
```

JAMSFetch

How to install?

The screenshot shows the PyPI project page for "jamsfetch 0.0.2". The top navigation bar includes links for Help, Docs, Sponsors, Log in, and Register. A search bar is at the top right. The main title is "jamsfetch 0.0.2" with a "Latest version" button. Below the title is a pip installation command: "pip install jamsfetch". The release date is listed as "Released: Jun 8, 2025". The project description states: "JAMS-Fetch (Joint Automated Multi-source Sequence Fetcher) is a Python package. It automates the retrieval of sequence and structure data from major bioinformatics databases using a unified, user-friendly interface." On the left, there's a "Navigation" sidebar with "Project description" (selected), "Release history", and "Download files". Below the sidebar are sections for "Verified details" (with a green checkmark), "Maintainers" (listing "D4S1"), and "Supported databases" (listing NCBI Nucleotide, NCBI Genome Assembly, UniProt, and RCSB PDB). A descriptive paragraph at the bottom says: "Whether you're working with DNA sequences, protein sequences, or molecular structures, JAMSFetch simplifies the download process and saves files in standard formats."

Search projects

Help Docs Sponsors Log in Register

jamsfetch 0.0.2

✓ Latest version

Released: Jun 8, 2025

platform linux -- Python 3.12.3, pytest-8.4.0, pluggy-1.6.0
rootdir: /home/lambi/Desktop/studia/ArchProj/JAMS
configfile: pyproject.toml
collected 15 items

tests/test_alphaFold.py ...
tests/test_assembly.py ...
tests/test_auto.py ..
tests/test_deconvolute.py .
tests/test_esm.py ..
tests/test_nucleotide.py ...
tests/test_pdb.py ..
tests/test_uniprot.py ..

Navigation

Project description

JAMSFetch

JAMSFetch (Joint Automated Multi-source Sequence Fetcher) is a Python package. It automates the retrieval of sequence and structure data from major bioinformatics databases using a unified, user-friendly interface.

Supported databases:

- NCBI Nucleotide
- NCBI Genome Assembly
- UniProt
- RCSB PDB

Whether you're working with DNA sequences, protein sequences, or molecular structures, JAMSFetch simplifies the download process and saves files in standard formats.

JAMSFetch

Which databases are supported?



UniProt



NCBI Nucleotide



NCBI Genome Assembly



PDB



AlphaFold DB



ESM Metagenomic Atlas



PDB

Download desired protein structures:

```
from jamsfetch.utils import get_pdb  
  
get_pdb(  
    pdb_ids=['1TUP', '9E2J'],  
    output_dir="structures/",  
    file_format="pdb",  
    unzip=True  
)
```



The package also supports automatic fetching
of identifiers from specific databases.

SUMMARY

1. One tool, all major sources

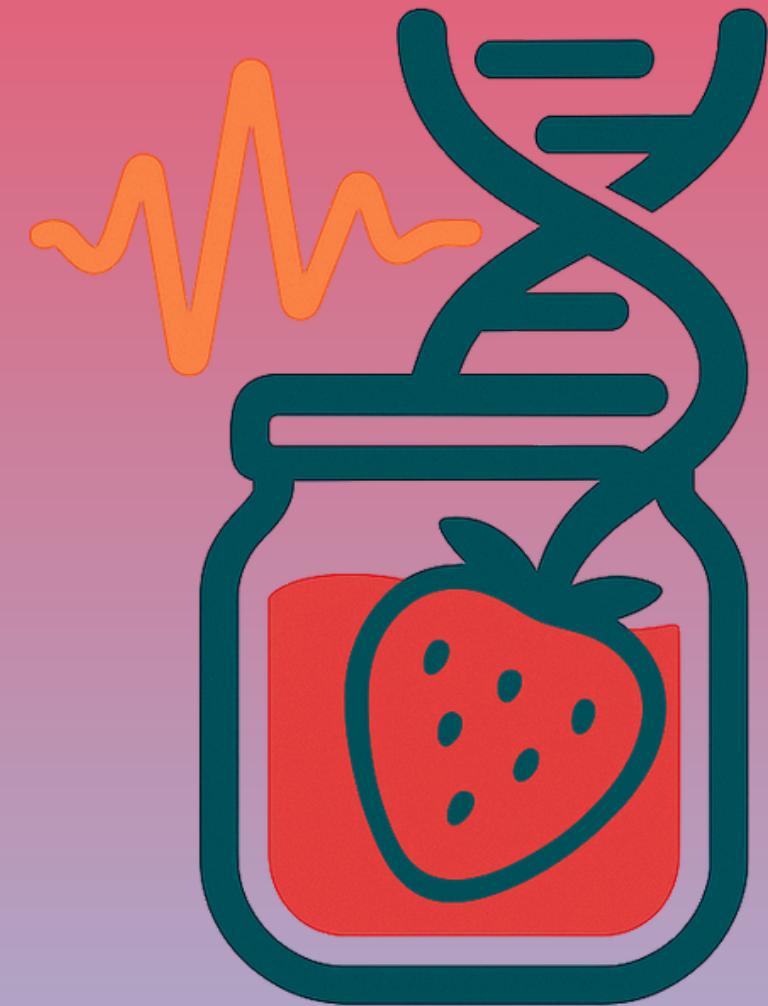
Automatically detects and fetches data from UniProt, PDB, NCBI, AlphaFold, and ESM - no manual API juggling needed.

2. Streamlined batch workflows

Download sequences and structures in bulk with minimal code — perfect for pipelines, scripts, or notebooks.

3. Standardized formats, organized output

Saves data in FASTA, PDB, or CIF formats and neatly organizes it into user-defined directories.



JAMS-Fetch