Cross Verification of Protein Information

Group 2

Hassan Elsayed
Aya Abdelbaky
Daria Podorskaja
Mostafa Elhawwary

9th February 2021

Outline

- Introduction (Hassan)
- Methodology (Aya)
- Software Overview (Daria)
- Discussion (Mostafa)

1. Introduction

Project overview

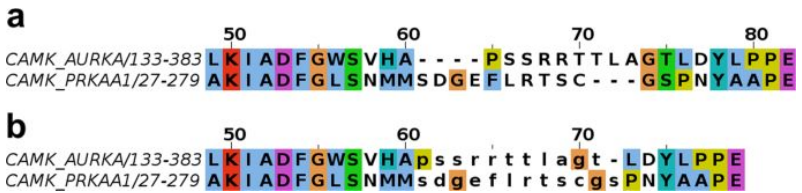
Cross Verification of Protein Information

- Investigate whether protein information available on two different "gold standard databases" ; **Uniprot** and **NCBI refseq** match up

- Check whether the extracted comparable informations from the two databases are properly **updated**

Scientific significance

- Generate a **user friendly visual interface** of the data from Uniprot and Refseq
- To provide a **trustworthy package** for scientists & researchers to be able to retrieve non redundant and an up to date information about human proteins



Project problem/task

- Databases are not the same / uncomparable
  - query for the same protein in two different database will not get the same response
- Different research groups update or submit protein structure data independently
  - Difference can be in annotation, sequence, variant or PTM
- Compare data for all human genome across databases is a challenge

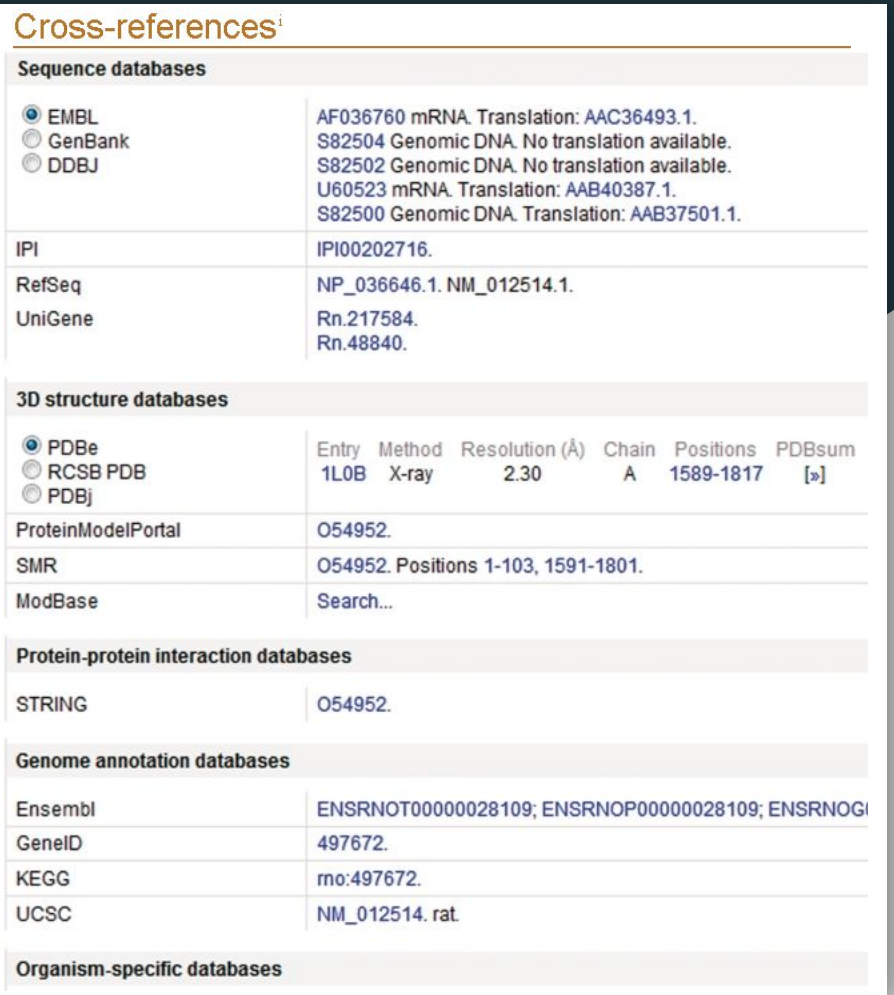
Background

- **Uniprot** is a protein sequence and annotation database., it provide a high-quality and freely accessible resource for most of the publicly available proteins of the world
- **Uniprot** is so-called gold standard“ as it provide a highly curated and annotated information about proteins using their two part system (Swiss-Prot and TrEMBL)

- **Refseq** “is an integrated, non-redundant, well-annotated set of reference sequences including genomic, transcript, and **protein”**.
- Refseq built by NCBI

Uniprot cross reference

- Uniprot cross reference with many databases.
- Each database is described by its name and abbreviation and a link to its web server is provided, as well as literature references where available.



Cross-references

Sequence databases

EMBL / GenBank / DDBJ	AF036760 mRNA. Translation: AAC36493.1. S82504 Genomic DNA. No translation available. S82502 Genomic DNA. No translation available. U60523 mRNA. Translation: AAB40387.1. S82500 Genomic DNA. Translation: AAB37501.1.
IPI	IPI00202716.
RefSeq	NP_036646.1. NM_012514.1.
UniGene	Rn.217584. Rn.48840.

3D structure databases

PDBe / RCSB PDB / PDBj

Entry	Method	Resolution (Å)	Chain	Positions	PDBsum
1L0B	X-ray	2.30	A	1589-1817	[»]

ProteinModelPortal	O54952.
SMR	O54952. Positions 1-103, 1591-1801.
ModBase	Search...

Protein-protein interaction databases

STRING	O54952.

Genome annotation databases

Ensembl	ENSRNOT00000028109; ENSRNOP00000028109; ENSRNOG
GeneID	497672.
KEGG	rno:497672.
UCSC	NM_012514. rat.

Organism-specific databases

Cross-references

Web resources

IL2RAbase
IL2RA mutation db
SeattleSNPs

Sequence databases

Select the link destinations:
EMBL
GenBank
DDBJ

X01057 mRNA Translation: CAA25525.1
X03131 » X03138 Genomic DNA Translation: CAA26906.1
K03122 mRNA Translation: AAB59535.1 Sequence problems.
M11066 » M11065 Genomic DNA Translation: AAA67527.1
AY563103 Genomic DNA Translation: AAS55572.1
AL157395 Genomic DNA No translation available.
AL137186 Genomic DNA No translation available.
CH471072 Genomic DNA Translation: EAW86414.1
M15864 Genomic DNA Translation: AAA59162.1
BN000945 Genomic DNA Translation: CAK26553.1

CCDS: CCDS7076.1
PIR: A44186, UHHU2
RefSeq: NP_000408.1, NM_000417.2
NP_001295171.1, NM_001308242.1
NP_001295172.1, NM_001308243.1

3D structure databases

Select the link destinations:
PDBe
RCSB PDB
PDBj

PDB entry	Method	Resolution (Å)
1ILM	model	-
1ILN	model	-
1Z92	X-ray	2.80
2B5I	X-ray	2.30
2ERJ	X-ray	3.00
3IU3	X-ray	2.90

interleukin-2 receptor subunit alpha isoform 1 precursor [Homo sapiens]

NCBI Reference Sequence: NP_000408.1

Identical Proteins FASTA Graphics

Go to:

```
LOCUS       NP_000408               272 aa            linear   PRI 15-DEC-2020
DEFINITION  interleukin-2 receptor subunit alpha isoform 1 precursor [Homo
            sapiens].
ACCESSION   NP_000408
VERSION     NP_000408.1
DBSOURCE    REFSEQ: accession NM_000417.3
KEYWORDS    RefSeq; MANE Select.
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
REFERENCE   1  (residues 1 to 272)
  AUTHORS   Ni M, Tian FB, Xiang DD and Yu B.
  TITLE     Characteristics of inflammatory factors and lymphocyte subsets in
            patients with severe COVID-19
  JOURNAL   J Med Virol 92 (11), 2600-2606 (2020)
   PUBMED   32470153
REFERENCE   2  (residues 1 to 272)
  AUTHORS   Stefanovic M, Zivotic I, Stojkovic L, Dincic E, Stankovic A and
            Zivkovic M.
  TITLE     The association of genetic variants IL2RA rs2104286, IFI30
            rs11554159 and IKZF3 rs12946510 with multiple sclerosis onset and
```

Program implementation/Solve tasks

- ★ Uniprot and refseq are cross reference to each others
- ★ Download all human proteins sequence from Uniprot and refseq
  - ○ Send API query to extract data needed from databases
  - ○ Ensure gene symbol, accession identifier, amino acid sequence are found in the downloaded data
- ★ Map human proteins from the two databases (choose a column)
- ★ Compare between uniprot and refseq data (comparable value)
  - ○ accession IDs , gene symbol, amino acid sequence
- ★ Show the data in a web interface displaying a comparison between the two databases.

2. Methodology

Strategy

- **Task 1**
  - Database Download (Daria)
- **Task 2**
  - Mapping the Metadata (Hassan)
- **Task 3**
  - Compare the Data (Aya)
- **Task 4**
  - GUI (Mostafa)

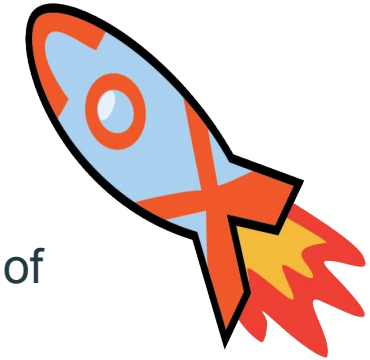
Implementation

- Step 1
  - Download the entire Human Proteome from UniProt and NCBI RefSeq databases
  - Write the downloaded metadata to separate “.tsv” files
- Step 2
  - Merge both files using “Accession Number Version”
  - Write the merged metadata to a new “.tsv” file
- Step 3
  - Generate a dataframe with the specified protein metadata obtained from both databases
  - Generate a dataframe with the percentage similarity for across the whole Human Proteome
- Step 4
  - View Table of Summary Statistics for comparing the databases
  - View Table for the metadata of a protein given by the user

Competitors

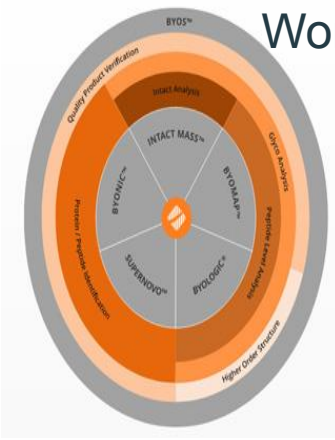
Software Tool for Researching Annotations of Proteins (STRAP)

LAB / MaxQuant



customProDB / R Package

Protein Metrics / Featured Byos® Workflows



PROTEIN METRICS

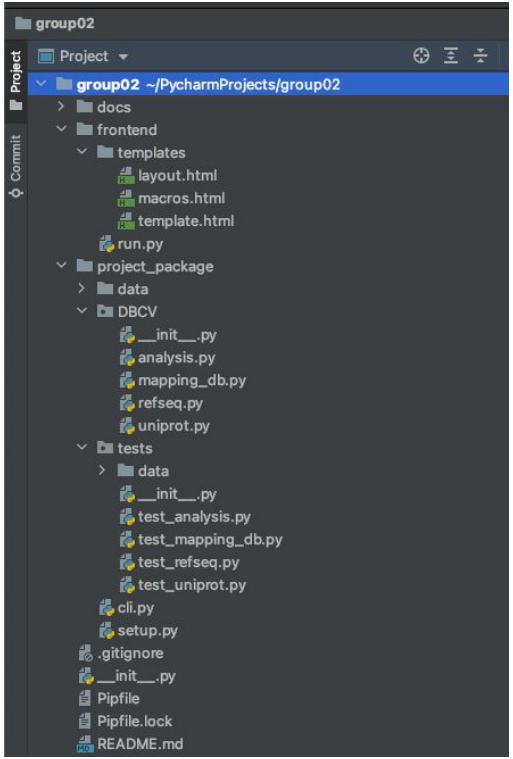
Olson Lab / Optide-Hunter

DBCV

3. Software Overview

Project structure

- Frontend
  - Templates
  - run.py
- Project package
  - Data
  - DBCV
  - Tests
    - Test data
- cli.py
- setup.py
- pipfile
- README.md



Download Data from RefSeq (refseq.py)

```python
def download_refseq(output_path: str, entries_number: int = 120_000):
    """
    Fetch entries form NCBI protein database using Entrez API functions.
    API query: "Homo sapiens[Orgn] AND refseq[filter]"
    GenBank parse is used for reading the fetched data.
    https://biopython.readthedocs.io/en/latest/chapter_entrez.html
    :param entries_number: number of entries
    :param output_path: path to output file
    :return:
    """
    if os.path.exists(output_path):
        print(f"File {output_path} already exists")
        return

    add_headers(output_path=output_path)
    handle = Entrez.esearch(db="protein", term="Homo sapiens[Orgn] AND refseq[filter]", retmax=entries_number)
    record = Entrez.read(handle)
    for i in range(0, entries_number, 101):

        handle = Entrez.efetch(db="protein", id=record['IdList'][i:i + 100], rettype="gp", retmode="text")
        records = GenBank.parse(handle)

        for r in records:
            data = augment_data(r)
            write_data(data, output_path)
        handle.close()
```

Download Data from Uniprot (uniprot.py)

```python
def generate_payload():
    """
    Payload for the API query "https://www.uniprot.org/uniprot/?query=proteome:UP000005640"
    Documentation can be found here:
    https://bioservices.readthedocs.io/en/master/_modules/bioservices/uniprot.html
    :return: payload for the API query
    """
    return {
        'query': 'organism :"Homo sapiens (Human) [9606]" AND proteome:up000005640',
        'format': 'tab',
        'columns': ','.join(['genes', 'id', 'database(HGNC)', 'database(RefSeq)', 'sequence']),
    }


def download_uniprot(payload: dict, output_path: str = "../data/uniprot.tsv"):
    """
    Download information from UniProt based on the payload.
    :param payload: payload for API query
    :param output_path: path for the output file
    :return:
    """

    if os.path.exists(output_path):
        print(f"File {output_path} already exists")
        return
    result = requests.get(BASE + KB_ENDPOINT, params=payload)
    if result.ok:
        with open(output_path, 'a') as f:
            f.write(result.text)
    else:
        print(result.status_code)
```

Mapping of the Data (mapping_db.py)

```
60      # Merging the modified Uniprot dataframe and the RefSeq dataframes based on the  ReSeq accession number
61
62      df4_mergefiles = pd.merge(refseq_df, df3_newuniprot, on='accession_number_version')
63
64      # Rearraging the columns of the dataframe for better interpretation
65
66      df4_mergefiles = df4_mergefiles[
67          ['accession_number_version', 'Refseq_accession_number_Refseq', 'db_source_accession_Refseq', 'Entry_Uniprot',
68           'Entry_Uniprot_Refseq',
69           'HGNC_Uniprot', 'HGNC_Refseq', 'MimID_Refseq', 'Gene_Names_Uniprot', 'Gene_Name_Refseq',
70           'Gene_synonyms_Refseq',
71           'Gene_ID_Refseq', 'Sequence_Uniprot', 'Sequence_Refseq']]
72
73      # Writing the dataframe to an output file
74      df4_mergefiles.to_csv(out_path, sep='\t', index=False, header=True)
```

Analysis of the Data (analysis.py)

```
protein_info_data = [
    ["Protein's gene symbol", str(df.iloc[row, df.columns.get_loc('Gene_Names_Uniprot')]).split(' ')[0],
     df.iloc[row, df.columns.get_loc('Gene_Name_Refseq')],
     str(df.iloc[row, df.columns.get_loc('Gene_Names_Uniprot')]).split(' ')[0] == df.iloc[
         row, df.columns.get_loc('Gene_Name_Refseq')]],
    ["UniProt accession IDs", df.iloc[row, df.columns.get_loc('Entry_Uniprot')],
     df.iloc[row, df.columns.get_loc('Entry_Uniprot_Refseq')],
     df.iloc[row, df.columns.get_loc('Entry_Uniprot')] == df.iloc[
         row, df.columns.get_loc('Entry_Uniprot_Refseq')]],
    ["RefSeq accession IDs", df.iloc[row, df.columns.get_loc('accession_number_version')],
     df.iloc[row, df.columns.get_loc('Refseq_accession_number_Refseq')],
     df.iloc[row, df.columns.get_loc('accession_number_version')].split('.')[0] == df.iloc[
         row, df.columns.get_loc('Refseq_accession_number_Refseq')]],
    ["Amino Acid sequences lengths", len(df.iloc[row, df.columns.get_loc('Sequence_Uniprot')]),
     len(df.iloc[row, df.columns.get_loc('Sequence_Refseq')]),
     len(df.iloc[row, df.columns.get_loc('Sequence_Uniprot')]) == len(
         df.iloc[row, df.columns.get_loc('Sequence_Refseq')])],
    ["Amino Acid sequences alignments", uniprot_seq,
     refseq_seq,
     df.iloc[row, df.columns.get_loc('Sequence_Uniprot')] == df.iloc[
         row, df.columns.get_loc('Sequence_Refseq')]]
]
protein_info = pd.DataFrame(data=protein_info_data, columns=protein_info_columns)
```

Frontend (run.py)

```python
@app.route("/")
def home():
    """
    This method loads the home page with the comparison table between the two databases

    """
    df = compare_data('new_out.tsv')
    df = df.to_html(index=False)
    return render_template('template.html', DF=df)


@app.route("/protein", methods=['POST'])
def protein():
    """
    This method is responsible for displaying the comparison between the two databases for a specific protein written
    in the textbox of the GUI

    """
    summary = compare_data('new_out.tsv')
    summary = summary.to_html(index=False)
    protein_name = request.form['textbox']
    df = compare_data('new_out.tsv', protein_name)
    if df is not None:
        df = df.to_html(index=False)

    return render_template('template.html', comp=df, DF=summary)
```

Encountered Problems

- Finding the right NCBI database
- Size of the downloaded files from the database
- Sequence is too long to display on frontend for comparison

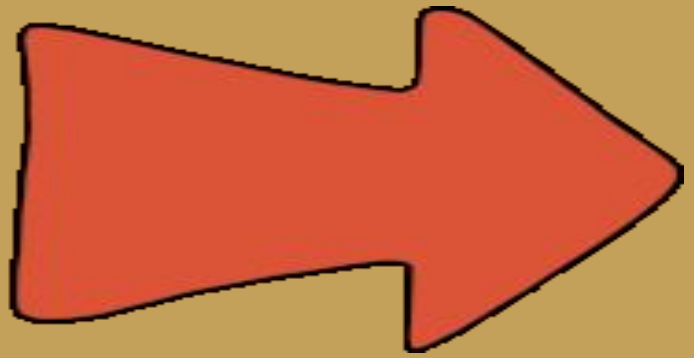
4. Discussion

Use case:

- Researchers working on protein data

- Used to decide whether the info they have is consistent or not
  - Source?
  - Matching or not!?

Live Demo.



References

Google search. (n.d.). Retrieved February 06, 2021, from https://www.google.com/search?q=c%23%2B&tbm=isch&ved=2ahUKEwjMw6TCp9buAhUOy6QKHTWHAnwQ2-cCegQIABAA&oq=c%23%2B&gs_lcp=CgNpbWcQAzIECCMQJzIECCMQJzICCAAyAggAMgQIABBDMgIIADICCAAyAggAMgIIADICCABQ_cgBWP3IAWCMywFoAHAAeACAAT-IAT-SAQExmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=SBgfYMz_IY6WkwW1jorgBw&bih=754&biw=1536&rlz=1C1SQJL_enDE872DE872#imgrc=F-b4ROxPF0wVoM&imgdii=s4dhvB95W_OXs

Google search. (n.d.). Retrieved February 06, 2021, from https://www.google.com/search?q=featured%2Bbyos%C2%AE%2Bworkflows&tbm=isch&ved=2ahUKEwiJjujEptbuAhUlJcUKHerZDlUQ2-cCegQIABAA&oq&gs_lcp=CgNpbWcQARgFMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnUImuA1iJrgNgw-wDaAFwAHgAgAEAiAEAkgEAmAEAoAEBqgELZ3dzLXdpei1pbWewAQrAAQE&sclient=img&ei=QRcfYMnpGaXKlAbqs7uoBQ&bih=754&biw=1536&rlz=1C1SQJL_enDE872DE872#imgrc=S6rnROdy6V6jwM&imgdii=2dJoZFhcnJx0dM

Google search. (n.d.). Retrieved February 06, 2021, from https://www.google.com/search?q=maxquant&tbm=isch&ved=2ahUKEwirhbz5pdbuAhVL2KQKHeGjDwgQ2-cCegQIABAA&oq&gs_lcp=CgNpbWcQARgFMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnUOuOCVjSlAlgycgJaAFwAHgEgAGvC4gB3CuSAQM3LTSYAQCgAQGqAQtnd3Mtd2l6LWltZ7ABCsABAQ&sclient=img&ei=oxYfYOunGcuwkwXhx75A&bih=754&biw=1536&rlz=1C1SQJL_enDE872DE872#imgrc=d7n_bNnJl8fH2M

Google search. (n.d.). Retrieved February 06, 2021, from https://www.google.com/search?q=optide-hunter&tbm=isch&ved=2ahUKEwik47rjptbuAhUegaQKHe8UAdkQ2-cCegQIABAA&oq&gs_lcp=CgNpbWcQARgFMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnMgcIIxDqAhAnUMW_AljFvwJgttwCaAFwAHgAgAEAiAEAkgEAmAEAoAEBqgELZ3dzLXdpei1pbWewAQrAAQE&sclient=img&ei=gRcfYOSeKp6CkgXvqYTIDQ&bih=754&biw=1536&rlz=1C1SQJL_enDE872DE872#imgrc=wXnxoCJZ0IbDBM

Google search. (n.d.). Retrieved February 06, 2021, from https://www.google.com/search?q=r%2Bpackage&tbm=isch&ved=2ahUKEwjo1Y7spdbuAhUHrKQKHYd1DAsQ2-cCegQIABAA&oq=r%2Bpackage&gs_lcp=CgNpbWcQAzIECCMQJzICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyAggAUN_QAVjf0AFgvdMBaABwAHgAgAE6iAE6kgEBMZgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=hxYfYKj2GIfYkgWH67FY&bih=754&biw=1536&rlz=1C1SQJL_enDE872DE872#imgrc=01ba1L3iPuiLiM&imgdii=rAjGJIScdLSbBM