Final Project

DeepPurpose: a Deep Learning Library for Drug-Target Interaction Prediction

Piyar Ali , Ali

CSCI E-104 Advanced Deep Learning, 2023

Harvard University Extension School

Prof. Zoran B. Djordjević

Contents

Project Summary:

DeepPurpose is a powerful deep learning library that specializes in drug discovery and drug repurposing tasks. The library employs various deep learning models such as GCN and RNN to learn representations of molecular structures and predict their properties. DeepPurpose is equipped with a variety of features that include drug-target interaction prediction, compound property prediction, protein-protein interaction prediction, and protein function prediction.

However, it does not provide protein to protein amino acid sequence similarity. Protein sequence pairwise similarity is a valuable tool that can aid in the identification and characterization of drug targets, as well as optimizing drug design and minimizing off-target effects. Our project aims to leverage the built-in target encoders in DeepPurpose to encode protein sequence data from uniport to train a model to calculate the similarity pairwise scores.

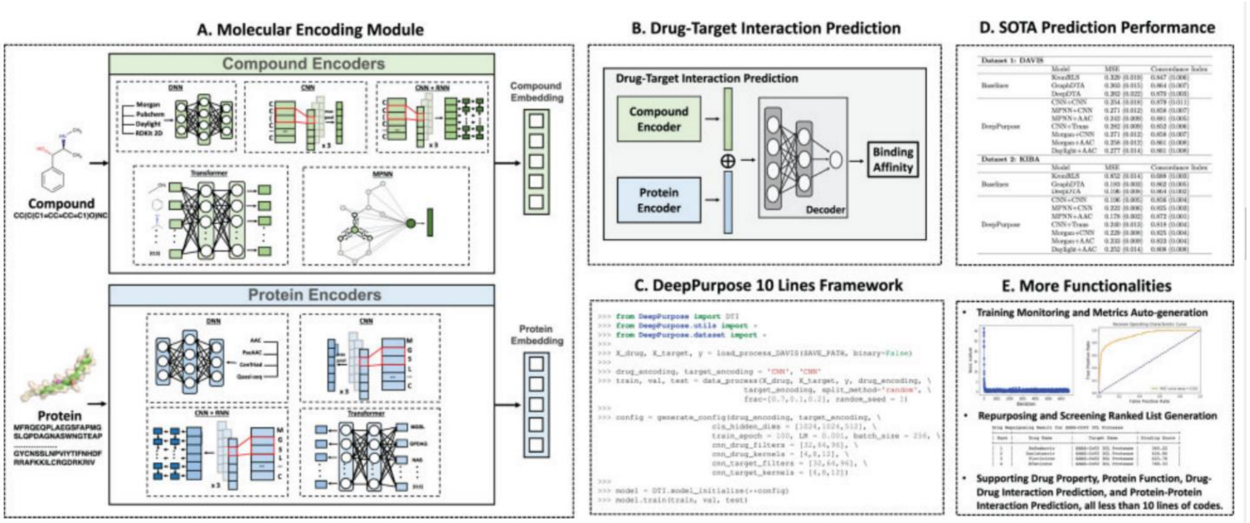
DeepPurpose protein to protein interaction prediction utilizes affinity scores. Protein affinity scores, or binding affinity scores, measure the attraction between proteins and ligands and help us understand how proteins and ligands interact with each other. In contrast, pairwise similarity is more focused on comparing the amino acid sequence and inferring evolutionary relationships between proteins. While both protein affinity scores and protein pairwise similarity provide information about proteins, they measure different aspects of protein biology.

To verify the protein to protein pairwise scores generated by our model, we can use NCBI BLAST, which is a widely used algorithm developed and maintained by the National Center for Biotechnology Information (NCBI) for comparing and analyzing biological sequences. BLAST can provide alignment scores if the two proteins are aligned. Overall, DeepPurpose is a powerful tool for drug discovery and drug repurposing tasks, and it offers a wide range of functionalities and tools for various aspects of drug discovery.

Problem Statement:

DeepPurpose is a powerful deep learning library for drug discovery, but it lacks protein to protein amino acid sequence similarity, which is an essential tool for identifying and characterizing drug targets, optimizing drug design, and minimizing off-target effects. The objective of the project is to leverage the built-in target encoders in DeepPurpose to encode protein sequence data from uniport and train a model to calculate the similarity pairwise scores. To verify the protein to protein pairwise scores generated by the model, the project will use NCBI BLAST, which is a widely used algorithm for comparing, analyzing, and scoring biological sequences.

DeepPurpose Library


A. Molecular Encoding Module
Compound Encoders
DNN
Morgan
Pubchem
Daylight
RDKit 2D
CNN
CNN + RNN
Transformer
MPNN
Compound Embedding
Compound
CC(C(C1=CC=CC=C1)O)NC
Protein Encoders
DNN
AAC
PseAAC
Conjtriad
Quasi-seq
CNN
CNN + RNN
Transformer
Protein Embedding
Protein
MFRQEQPLAEGSFAPMG
SLQPDAGNASWNGTEAP
GYCNSSLNPVIYTIFNHDF
RRAFKKILCRGDRKRIV
B. Drug-Target Interaction Prediction
Drug-Target Interaction Prediction
Compound Encoder
Protein Encoder
Decoder
Binding Affinity
C. DeepPurpose 10 Lines Framework
D. SOTA Prediction Performance
E. More Functionalities
Training Monitoring and Metrics Auto-generation
Repurposing and Screening Ranked List Generation
Supporting Drug Property, Protein Function, Drug-Drug Interaction Prediction, and Protein-Protein Interaction Prediction, all less than 10 lines of codes.

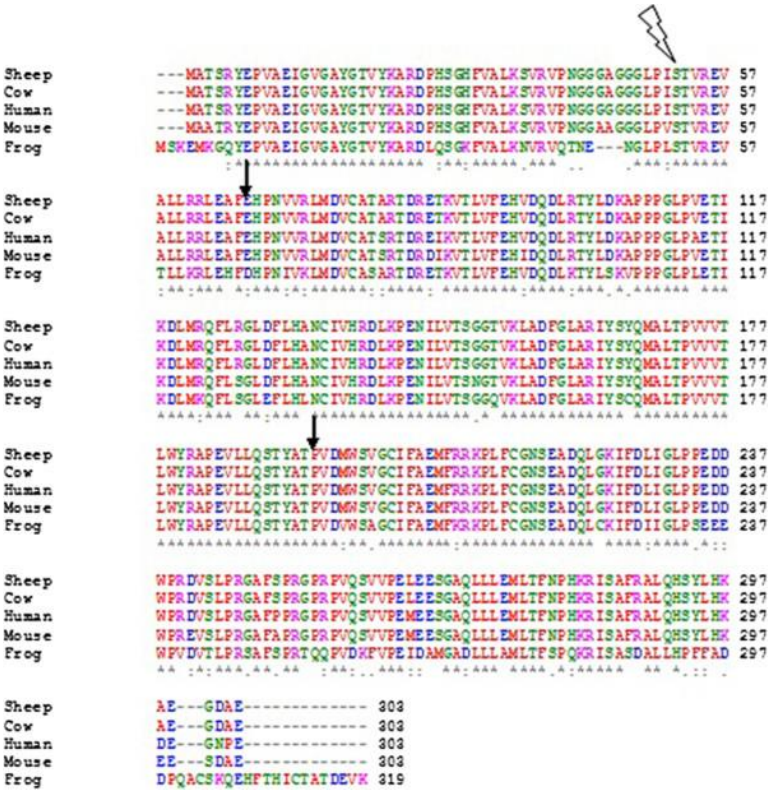
```
>>> from DeepPurpose import DTI
>>> from DeepPurpose.utils import *
>>> from DeepPurpose.dataset import *
>>>
>>> X_drug, X_target, y = load_process_DAVIS(SAVE_PATH, binary=False)
>>>
>>> drug_encoding, target_encoding = 'CNN', 'CNN'
>>> train, val, test = data_process(X_drug, X_target, y, drug_encoding, \
                                    target_encoding, split_method='random', \
                                    frac=[0.7,0.1,0.2], random_seed = 1)
>>>
>>> config = generate_config(drug_encoding, target_encoding, \
                             cls_hidden_dims = [1024,1024,512], \
                             train_epoch = 100, LR = 0.001, batch_size = 256, \
                             cnn_drug_filters = [32,64,96], \
                             cnn_drug_kernels = [4,8,12], \
                             cnn_target_filters = [32,64,96], \
                             cnn_target_kernels = [4,8,12])
>>>
>>> model = DTI.model_initialize(**config)
>>> model.train(train, val, test)
```

Dataset 1: DAVIS			
	Model	MSE	Concordance Index
Baselines	KronRLS	0.329 (0.019)	0.847 (0.006)
	GraphDTA	0.263 (0.015)	0.864 (0.007)
	DeepDTA	0.262 (0.022)	0.870 (0.003)
DeepPurpose	CNN+CNN	0.254 (0.018)	0.879 (0.011)
	MPNN+CNN	0.271 (0.012)	0.858 (0.007)
	MPNN+AAC	0.242 (0.009)	0.881 (0.005)
	CNN+Trans	0.292 (0.009)	0.852 (0.006)
	Morgan+CNN	0.271 (0.012)	0.858 (0.007)
	Morgan+AAC	0.258 (0.012)	0.861 (0.008)
	Daylight+AAC	0.277 (0.014)	0.862 (0.008)
Dataset 2: KIBA			
	Model	MSE	Concordance Index
Baselines	KronRLS	0.852 (0.014)	0.688 (0.003)
	GraphDTA	0.183 (0.003)	0.862 (0.005)
	DeepDTA	0.196 (0.008)	0.864 (0.002)
DeepPurpose	CNN+CNN	0.196 (0.005)	0.858 (0.004)
	MPNN+CNN	0.222 (0.006)	0.825 (0.003)
	MPNN+AAC	0.178 (0.002)	0.872 (0.001)
	CNN+Trans	0.240 (0.013)	0.818 (0.004)
	Morgan+CNN	0.229 (0.008)	0.825 (0.004)
	Morgan+AAC	0.233 (0.009)	0.823 (0.004)
	Daylight+AAC	0.252 (0.014)	0.808 (0.008)

Image Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8016467/pdf/btaa1005.pdf

Fig. 1. Overview of DeepPurpose library. (A) DeepPurpose takes as input the SMILES of a compound and a protein's amino acid sequence and then generates embeddings for them. (B) The learned embeddings are then concatenated and fed into a decoder to predict DTI binding affinity. (C) DeepPurpose provides a simple but flexible programming framework that implements over 50 state-of-the-art DL models for DTI prediction. (D) DeepPurpose models achieve comparable performance with three other DTI prediction algorithms on two benchmark datasets. (E) Finally, DeepPurpose has many functionalities, including monitoring the training process, debugging and generation ranked lists for repurposing and screening. Further, DeepPurpose supports other downstream prediction tasks (e.g., drug–drug interaction prediction, compound property prediction)

Protein Sequence Similarity



```
Sheep           ---MATSRYEPVAEIGVGAYGTVYKARDPHSGHFVALKSVRVPNGGGAGGGLPISTVREV 57
Cow             ---MATSRYEPVAEIGVGAYGTVYKARDPHSGHFVALKSVRVPNGGGAGGGLPISTVREV 57
Human           ---MATSRYEPVAEIGVGAYGTVYKARDPHSGHFVALKSVRVPNGGGGGGGLPISTVREV 57
Mouse           ---MAATRYEPVAEIGVGAYGTVYKARDPHSGHFVALKSVRVPNGGAAGGGLPVSTVREV 57
Frog            MSKEMKGQYEPVAEIGVGAYGTVYKARDLQSGKFVALKNVRVQTNE---NGLPLSTVREV 57
                       :******************** :**:*****.*** ..    .***:******

Sheep           ALLRRLEAFEHPNVVRLMDVCATARTDRETKVTLVFEHVDQDLRTYLDKAPPPGLPVETI 117
Cow             ALLRRLEAFEHPNVVRLMDVCATARTDRETKVTLVFEHVDQDLRTYLDKAPPPGLPVETI 117
Human           ALLRRLEAFEHPNVVRLMDVCATSRTDREIKVTLVFEHVDQDLRTYLDKAPPPGLPAETI 117
Mouse           ALLRRLEAFEHPNVVRLMDVCATSRTDRDIKVTLVFEHIDQDLRTYLDKAPPPGLPVETI 117
Frog            TLLKRLEHFDHPNIVKLMDVCASARTDRETKVTLVFEHVDQDLKTYLSKVPPPGLPLETI 117
                :**:*** *:***:*:******::****: ********:****:***.*.****** ***

Sheep           KDLMRQFLRGLDFLHANCIVHRDLKPENILVTSGGTVKLADFGLARIYSYQMALTPVVVT 177
Cow             KDLMRQFLRGLDFLHANCIVHRDLKPENILVTSGGTVKLADFGLARIYSYQMALTPVVVT 177
Human           KDLMRQFLRGLDFLHANCIVHRDLKPENILVTSGGTVKLADFGLARIYSYQMALTPVVVT 177
Mouse           KDLMRQFLSGLDFLHANCIVHRDLKPENILVTSNGTVKLADFGLARIYSYQMALTPVVVT 177
Frog            KDLMKQFLSGLEFLHLNCIVHRDLKPENILVTSGGQVKLADFGLARIYSCQMALTPVVVT 177
                ****:*** **:*** *****************.* ************* **********

Sheep           LWYRAPEVLLQSTYATPVDMWSVGCIFAEMFRRKPLFCGNSEADQLGKIFDLIGLPPEDD 237
Cow             LWYRAPEVLLQSTYATPVDMWSVGCIFAEMFRRKPLFCGNSEADQLGKIFDLIGLPPEDD 237
Human           LWYRAPEVLLQSTYATPVDMWSVGCIFAEMFRRKPLFCGNSEADQLGKIFDLIGLPPEDD 237
Mouse           LWYRAPEVLLQSTYATPVDMWSVGCIFAEMFRRKPLFCGNSEADQLGKIFDLIGLPPEDD 237
Frog            LWYRAPEVLLQSTYATPVDVWSAGCIFAEMFKRKPLFCGNSEADQLCKIFDIIGLPSEEE 237
                *******************:**.********:************** ****:****.*::

Sheep           WPRDVSLPRGAFSPRGPRFVQSVVPELEESGAQLLLEMLTFNPHKRISAFRALQHSYLHK 297
Cow             WPRDVSLPRGAFSPRGPRFVQSVVPELEESGAQLLLEMLTFNPHKRISAFRALQHSYLHK 297
Human           WPRDVSLPRGAFPPRGPRFVQSVVPEMEESGAQLLLEMLTFNPHKRISAFRALQHSYLHK 297
Mouse           WPREVSLPRGAFAPRGPRFVQSVVPEMEESGAQLLLEMLTFNPHKRISAFRALQHSYLHK 297
Frog            WPVDVTLPRSAFSPRTQQFVDKFVPEIDAMGADLLLAMLTFSPQKRISASDALLHPFFAD 297
                ** :*:***.**.**  :**:..***::  **:*** ****.*:*****  ** *.:: .

Sheep           AE---GDAE------------- 303
Cow             AE---GDAE------------- 303
Human           DE---GNPE------------- 303
Mouse           EE---SDAE------------- 303
Frog            DPQACSKQEHFTHICTATDEVK 319
```

Image Source: https://www.wikiwand.com/en/Clustal

Dataset:

DeepPurpose

Public Drug-Target Binding Benchmark Dataset

Data	Function
BindingDB	`download_BindingDB()` to download the data and `process_BindingDB()` to process the data
DAVIS	`load_process_DAVIS()` to download and process the data
KIBA	`load_process_KIBA()` to download and process the data

Repurposing Dataset

Data	Function
Curated Antiviral Drugs Library	`load_antiviral_drugs()` to load and process the data
Broad Repurposing Hub	`load_broad_repurposing_hub()` downloads and process the data

Bioassay Data for COVID-19 (Thanks to MIT AI Cures)

Data	Function
AID1706	`load_AID1706_SARS_CoV_3CL()` to load and process

Protein Sequence Data

Data	Function
Isoform sequences	`list(SeqIO.parse(seq_path, 'fasta'))`

Installation/configuration

The project is setup in Python Notebook using Google Colab

Install Libraries

```
!pip install rdkit
!pip install git+https://github.com/bp-kelley/descriptastorus
!pip install DeepPurpose
!pip install Bio
!pip install PySpark
!pip install wget
```

Import Libraries

```
from DeepPurpose import DTI as modelsdti
from DeepPurpose import PPI as modelsppi
from DeepPurpose.utils import *
from DeepPurpose.dataset import *
from DeepPurpose import dataset, CompoundPred, utils, encoders

import Bio
from Bio.Blast import NCBIWWW, NCBIXML
from Bio import SeqIO
from Bio.SeqRecord import SeqRecord
from Bio.Seq import Seq

import torch
import torch.nn as nn
import torch.optim as optim

import pandas as pd
import numpy as np
import time
import random
import csv
import os
import wget

import multiprocessing
from multiprocessing import pool

import matplotlib.pyplot as plt
```

Download Dataset

```
# Download the affinity scores file

!wget -P /content http://staff.cs.utu.fi/~aatapa/data/DrugTarget/drug-target_interaction_affinities_Kd__Davis_et_al.2011.txt

# Get the fasta (protein sequence data) from https://www.uniprot.org/help/downloads

!wget -P /content https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot_varsplic.fasta.gz

# extract/unzip the file

!gunzip /content/uniprot_sprot_varsplic.fasta.gz

# set the fasta file path

seq_path = '/content/uniprot_sprot_varsplic.fasta'
```

Load Dataset

```
# Read and extract protein records using Bio library

seq_records = list(SeqIO.parse(seq_path, 'fasta'))
```

Note: the dataset is large total records 40,957, possible Paris 40,957 x 40,957 = 1,677,475,849

for model training purpose, use only first 200 records.

```
seq_records = list(SeqIO.parse(seq_path, 'fasta'))[:200]
```

```
# Extract Sequence Pair
seqs = []
for i in range(len(seq_records)):
  seqs.append(str(seq_records[i].seq))
```

DeepPurpose Encoders

```
# Encode the protein sequence pair using DeepPurpose protein encoders

X = utils.encode_protein(pd.DataFrame(seq, columns=['Target Sequence']),
'AAC')
```

Note: Encoding AAC takes time. Time Reference: 24s for ~100 sequences in a CPU

DeepPurpose provides following different encoders, you can use any of them to improve the model:

Target Encodings	**Description**
AAC	Amino acid composition up to 3-mers
PseudoAAC	Pseudo amino acid composition
Conjoint_triad	Conjoint triad features
Quasi-seq	Quasi-sequence order descriptor
ESPF	Explainable Substructure Partition Fingerprint

CNN	Convolutional Neural Network on target seq
CNN_RNN	A GRU/LSTM on top of a CNN on target seq
Transformer	Transformer Encoder on ESPF

```
# Convert the target encoding to a tensor
X_tensor = torch.tensor(X['target_encoding'].values.tolist(),
dtype=torch.float)
```

Similarity Model

```
# Instantiate the similarity scoring model
model = nn.Sequential(
    nn.Flatten(),
    nn.Linear(16840, 256),
    nn.ReLU(),
    nn.Linear(256, 64),
    nn.ReLU(),
    nn.Linear(64, 1),
    nn.Sigmoid()
)
```

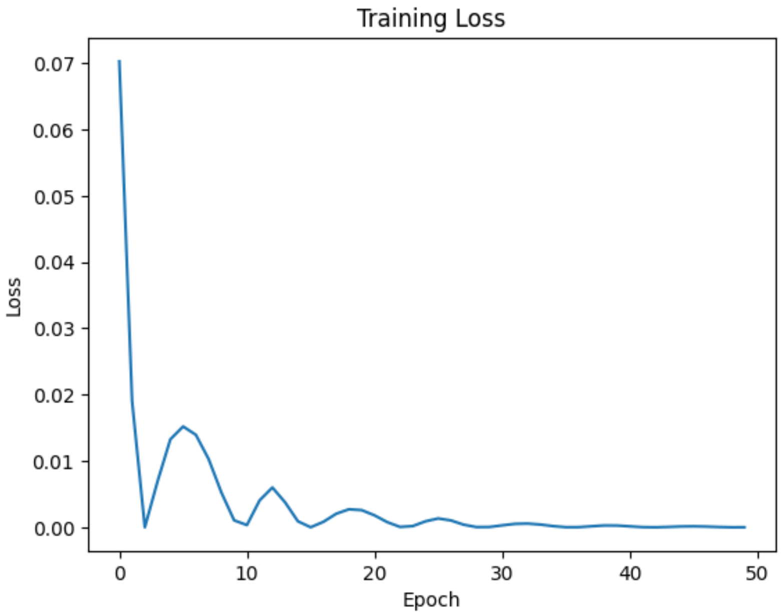
```
# Define the optimization algorithm and loss function
optimizer = optim.Adam(model.parameters(), lr=0.001)
criterion = nn.MSELoss()
```

```
# Train the model on a pair of protein sequences
x1 = X_tensor[0].unsqueeze(0)
x2 = X_tensor[1].unsqueeze(0)
y = torch.tensor([[0.8]], dtype=torch.float)
```

```
losses = []

for epoch in range(50):
    optimizer.zero_grad()

    # Concatenate x1 and x2 along the column axis
    x = torch.cat((x1, x2), dim=1)

    output = model(x)
    loss = criterion(output, y)
    loss.backward()
    optimizer.step()

    losses.append(loss.item())

    print('Epoch {}: Loss = {:.4f}'.format(epoch+1, loss.item()))
```

Results

Training Loss



Similarity Score

For the first two pairs

```
# Calculate the similarity score for the pair of protein sequences
```

```
score = model(torch.cat((x1, x2), dim=1)).item()
print('Similarity score: {:.4f}'.format(score))
```

Result: `Similarity score: 0.7929`

x1 = MENERAKQVYLAKLNEQAERYDEMVEAMKKVAALDVELTIEERNLLSVGYKNVIGARRASWRILSSIEQKEESKGNEQNAKRIKDYRTKVEEELSKICYDILAVIDKHLVPFATSGESTVFYYKMKGDYFRYLAEFKSGADREEAADLSLKAYEAATSSASTELSTTHPIRLGLALNFSVFYYEILNSPERACHLAKRAFDEAIAELDSLNEDSYKDSTLIMQLLRDNLTLWTSDLEEGGK

x2 = MENERAKQVYLAKLNEQAERYDEMVEAMKKVAALDVELTIEERNLLSVGYKNVIGARRASWRILSSIEQKEESKGNEQNAKRIKDYRTKVEEELSKICYDILAVIDKHLVPFATSGESTVFYYKMKGDYFRYLAEFKSGADREEAADLSLKAYEAATSSASTELSTTHPIRLGLALNFSVFYYEILNSPERACHLAKRAFDEAIAELDSLNEDSYKDSTLIMQLLRDNLTLWTSDLEEGGEQSKGHNQQDEVNKI

Verification

The scores can be verified using NCBI Blast: https://blast.ncbi.nlm.nih.gov/Blast.cgi

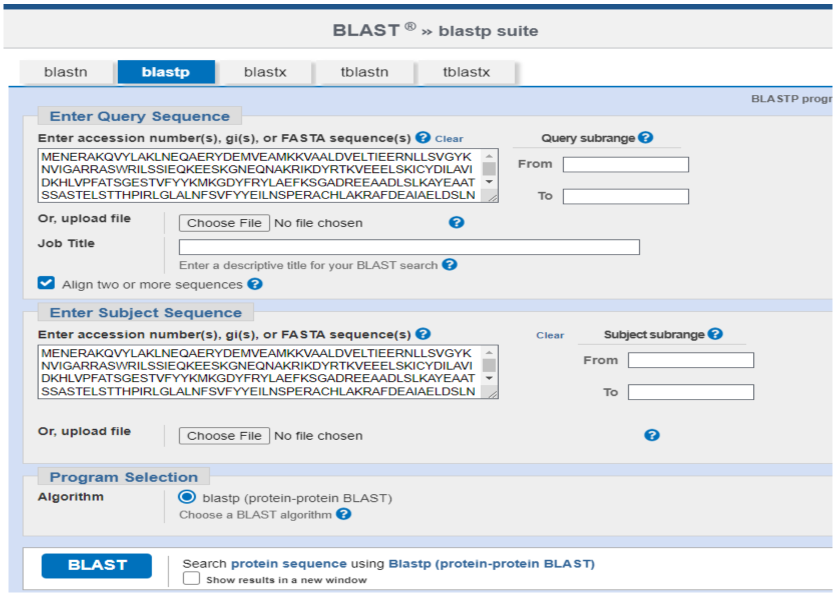
NCBI BLAST (Basic Local Alignment Search Tool) is a widely used algorithm developed and maintained by the National Center for Biotechnology Information (NCBI) for comparing and analyzing biological sequences, such as DNA, RNA, and protein sequences. Blast can provide alignment scores, if the two proteins are aligned. Blast can be access via web interface or using python library.

Using python library is very slow if verifying large dataset of protein-to-protein pairwise similarity sequence as the API has a limit of 15 seconds wait after each protein-to-protein pair request. Each request response takes approximately 30 seconds. For example, sending an API request for 200 protein pairs can take approximately 1.5 to 2 hours.

Python notebook provides the code for accessing the NCBI Blast using python. For the purposes of this project, we used the web interface.

Web Interface Verification

Enter the sequence in the NCBI Blast Search Interface



Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

```
MENERAKQVYLAKLNEQAERYDEMVEAMKKVAALDVELTIEERNLLSVGYK
NVIGARRASWRILSSIEQKEESKGNEQNAKRIKDYRTKVEEELSKICYDILAVI
DKHLVPFATSGESTVFYYKMKGDYFRYLAEFKSGADREEAADLSLKAYEAAT
SSASTELSTTHPIRLGLALNFSVFYYEILNSPERACHLAKRAFDEAIAELDSLN
```

Query subrange: From [ ] To [ ]

Or, upload file: No file chosen

Job Title: Enter a descriptive title for your BLAST search

☑ Align two or more sequences

Enter Subject Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

```
MENERAKQVYLAKLNEQAERYDEMVEAMKKVAALDVELTIEERNLLSVGYK
NVIGARRASWRILSSIEQKEESKGNEQNAKRIKDYRTKVEEELSKICYDILAVI
DKHLVPFATSGESTVFYYKMKGDYFRYLAEFKSGADREEAADLSLKAYEAAT
SSASTELSTTHPIRLGLALNFSVFYYEILNSPERACHLAKRAFDEAIAELDSLN
```

Subject subrange: From [ ] To [ ]

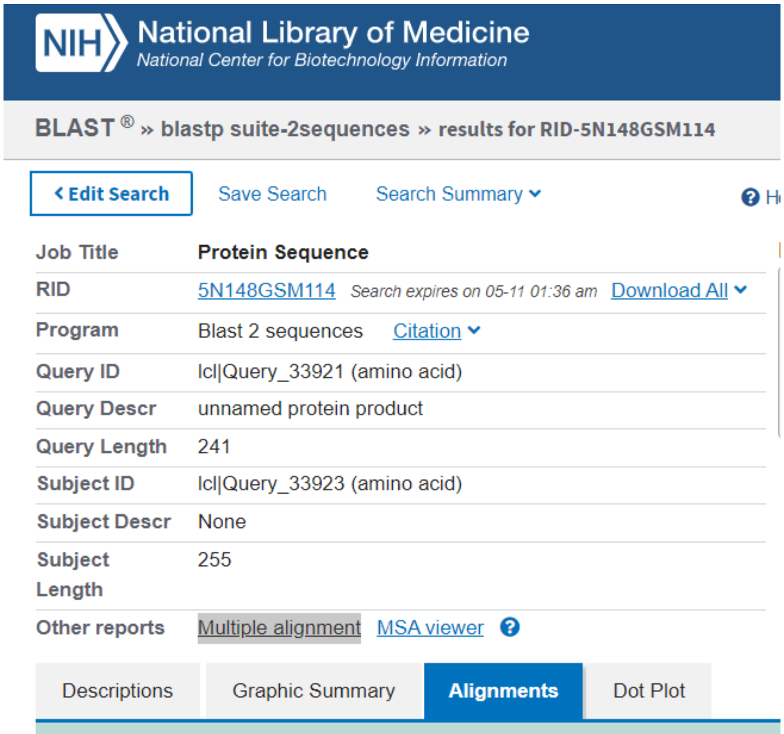
Or, upload file: No file chosen

Program Selection

Algorithm: ● blastp (protein-protein BLAST)
Choose a BLAST algorithm

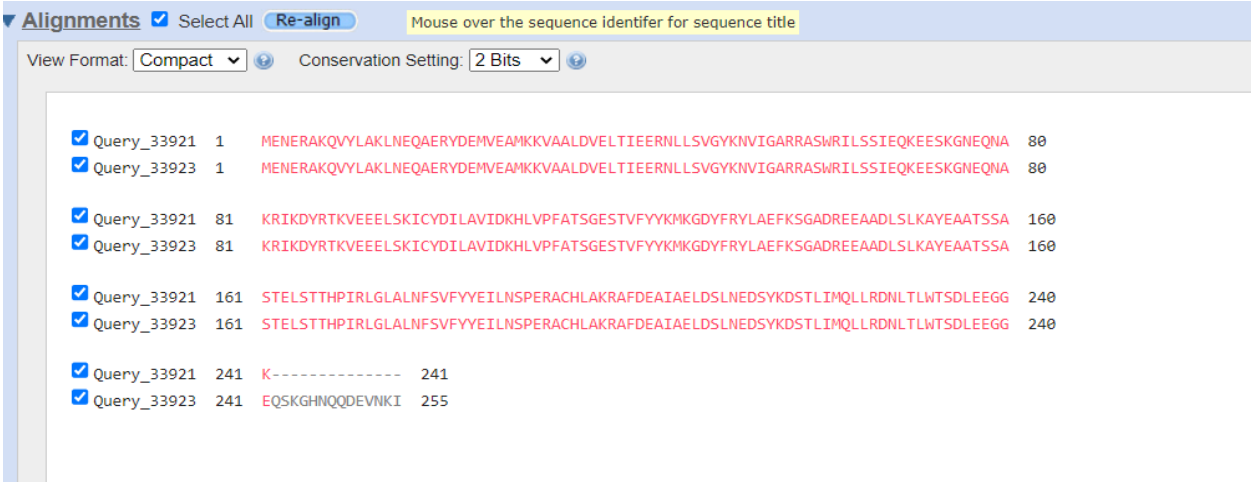
Search protein sequence using Blastp (protein-protein BLAST)
□ Show results in a new window

Get the results.


NIH
National Library of Medicine
National Center for Biotechnology Information
BLAST ® » blastp suite-2sequences » results for RID-5N148GSM114
< Edit Search
Save Search
Search Summary
H
Descriptions
Graphic Summary
Alignments
Dot Plot

Job Title	Protein Sequence
RID	5N148GSM114 *Search expires on 05-11 01:36 am* Download All
Program	Blast 2 sequences Citation
Query ID	lcl|Query_33921 (amino acid)
Query Descr	unnamed protein product
Query Length	241
Subject ID	lcl|Query_33923 (amino acid)
Subject Descr	None
Subject Length	255
Other reports	Multiple alignment MSA viewer

Analyze the Results


Alignments ☑ Select All Re-align
Mouse over the sequence identifer for sequence title

View Format: Compact   Conservation Setting: 2 Bits

```
☑ Query_33921  1    MENERAKQVYLAKLNEQAERYDEMVEAMKKVAALDVELTIEERNLLSVGYKNVIGARRASWRILSSIEQKEESKGNEQNA  80
☑ Query_33923  1    MENERAKQVYLAKLNEQAERYDEMVEAMKKVAALDVELTIEERNLLSVGYKNVIGARRASWRILSSIEQKEESKGNEQNA  80

☑ Query_33921  81   KRIKDYRTKVEEELSKICYDILAVIDKHLVPFATSGESTVFYYKMKGDYFRYLAEFKSGADREEAADLSLKAYEAATSSA  160
☑ Query_33923  81   KRIKDYRTKVEEELSKICYDILAVIDKHLVPFATSGESTVFYYKMKGDYFRYLAEFKSGADREEAADLSLKAYEAATSSA  160

☑ Query_33921  161  STELSTTHPIRLGLALNFSVFYYEILNSPERACHLAKRAFDEAIAELDSLNEDSYKDSTLIMQLLRDNLTLWTSDLEEGG  240
☑ Query_33923  161  STELSTTHPIRLGLALNFSVFYYEILNSPERACHLAKRAFDEAIAELDSLNEDSYKDSTLIMQLLRDNLTLWTSDLEEGG  240

☑ Query_33921  241  K--------------  241
☑ Query_33923  241  EQSKGHNQQDEVNKI  255
```

Our model predicted the similarity score of 0.7929

The NCBI Blast database score is 241/255 = 0.9450

Our model did very good. The model can be improved by experimenting with increasing epochs or changing the model hyper parameters or using different DeepPurpose available encoders.

Challenges

1. Data format challenge: DeepPurpose requires data in a specific format, but the documentation does not provide clear guidelines on this format. Users must either inspect their dataset or look at the code to determine the required format.
2. Large dataset challenge: The Uniprot protein sequence data is massive, with over 40,000 records, resulting in a possible 1.6 billion protein to protein pairs. Running a model on a good sample size requires a computer with high RAM, even when utilizing big data technologies such as Spark or Multiprocessing. The provided multiprocessing code in Python Notebook can help apply the similarity model on the full dataset, but most computers will run out of memory. This requires either a high-powered machine or chunking the data into batches and running the model on each batch.

3. Verification challenge: Our model similarity scores can be verified using NCBI Blast, either via a web interface or a Python library. However, using the Python library to verify large datasets of protein to protein pairwise similarity is very slow, as the API has a 15-second wait time after each protein-to-protein pair request. Each request response can take approximately 30 seconds. For example, sending an API request for 200 protein pairs can take approximately 1.5 to 2 hours.