Predikcija neuređenosti regiona/proteina korišćenjem jezičkih modela

Seminarski rad u okviru kursa
Uvod u bioinformatiku
Matematički fakultet

Anđela Damnjanović
mi19059@alas.matf.bg.ac.rs

5. jul 2024.

Sažetak

Još od otkrića prvih proteina tokom 18-og veka, naučnici teže da otkriju i ekstrahuju što veći broj ovih molekula kako bi bolje razumeli koje sve funkcije oni vrše u organizmu. Do sada su poznate mnoge funkcije proteina – gradivna, hormoni, enzimi, katalizatori samo su neke od njih. No, u poslednjih par decenija otkriveno je da neuređeni delovi proteina mogu da se dovedu u vezu sa nastajanjem kancera i mnogih neurodegenerativnih bolesti poput Parkinsonove i Alchajmerove bolesti. Stoga je veoma bitno posvetiti pažnju nalaženju ovakvih regiona u proteinima kako bi se mogao napraviti efikasan lek. Programeri aktivno pokušavaju da osmisle različite načine kako bi rešili ovaj problem, ali se nijedan metod do sada nije pokazao kao dovoljno precizan i efikasan. Mnogi od novijih pristupa podrazumevaju korišćenje neuronskih mreža i dubokog učenja. Na sličnom pristupu zasnovan je i metod koji će detaljnije biti obrađen u ovom radu. U pitanju je predikcija neuređenih regiona velikim jezičkim modelima.

Sadržaj

1 Uvod

Otkriće proteina i drugih makromolekula bitnih za funkcionisanje organizma odškrinulo je vrata koja vode boljem razumevanju procesa koji se svakodnevno odigravaju u ljudskom telu. Prvi proteini pronađeni su čak u 18. veku (gluten, albumin, fibrin...) i od tada se na tom polju neprestano dolazi do novih otkrića. Sa napretkom tehnika i tehnologije, uspešno je izolovano oko 20000 proteina koji ulaze u sastav ljudskog organizma, poput hemoglobina i nekih enzima, koji su pomno proučavani u laboratorijama. Od posebnog značaja za ovaj rad su proteini koji ne podležu zakonima uređenosti jer oni mogu biti indikatori za mnoge bolesti kod čoveka. Stoga je veoma bitno ispravno predvideti potencijalne regione neuređenosti i posmatrati sekvence aminokiselina koje se u njima nalaze kako bi se predupredila bolest. Upravo je predikcija neuređenih regiona centralna tema ovog rada. No, pre nego što se pređe na detaljniji opis problema, sledi upoznavanje sa osnovnim biološkim pojmovima u poglavlju 1.1, te sa velikim jezičkim modelima koji vrše predikciju neuređenih regiona u poglavlju 1.2.

1.1 Biološki okviri

Proteini (takođe poznati i pod nazivom *belančevine*) su prirodni makromolekuli nastali kondenzacijom aminokiselina, čime su stvorene *peptidne veze* – veze između amino grupe jedne i atoma ugljenika druge aminokiseline [3]. Zbog svoje kompleksnosti proteini su našli uloge u mnogim ćelijskim procesima: neki su tu da daju strukturu ćeliji, neki imaju uloge antitela koja štite organizam, drugi su pak enzimi koji potpomažu hemijske reakcije i stvaranje novih molekula, treći služe da za sebe vežu manje molekule i transportuju ih kroz ćeliju (ili čak između ćelija), četvrti služe za koordinaciju bioloških procesa i tako dalje [9].

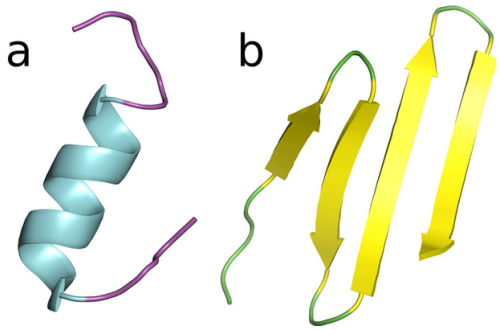
U sastav svakog proteina ulazi od nekoliko stotina do nekoliko hiljada manjih jedinica (gorepomenutih aminokiselina) koje su poređane u lance. Sam redosled u kome će se aminokiseline naći unutar jednog proteina zapisan je u genima. Genetički kod određuje 20 *osnovnih* aminokiselina koje se mogu naći u proteinima (pored osnovnih, postoje još 2 aminokiseline koje se mogu kodirati, ali se one ne smatraju esencijalnim). Svaka aminokiselina kodirana je sa 3 *nukleotida* koji su određeni sekvencom gena. Dakle, sve potrebne informacije potrebne za sintezu aminokiselina i kasnije proteina zapisane su unutar DNK.

Upravo redosled, broj i niz aminokiselina u polipeptidnom lancu čini *primarnu strukturu proteina*. Za proteine koji imaju međusobno sličnu primarnu strukturu ustanovljeno je da dele zajedničko evoluciono poreklo. Grupa proteina koji su svi međusobno slični po primarnoj strukturi naziva se *familija proteina* i trenutno je poznato preko 60000 proteinskih familija.

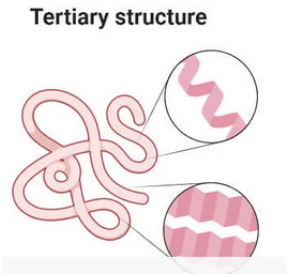
Redosled aminokiselina u lancu bitno utiče i na *sekundarnu strukturu proteina*. Ova struktura određena je diedralnim uglovima ϕ i ψ koji su nastali kao rezultat formiranja vodoničnih veza između CO i NH grupa aminokiselina iz polipeptidnog lanca. Najčešći oblik sekundarne strukture su α-*heliks* i β-*ravan*, koji se mogu videti na slici 1.

Međutim, bočni lanci aminokiselina nastavljaju da interaguju međusobno i savijaju se u još kompaktniji oblik, čime protein dobija svoju trodimenzionalnu strukturu koja predstavlja *tercijarnu strukturu proteina*.Primer tercijarne strukture proteina može se videti na slici 2.

Tokom šezdesetih godina prošlog veka američki biohemičar Kristijan Anfinsen izneo je teoriju (koja će po njemu kasnije i dobiti naziv *Anfin-*



Slika 1: Primer savijanja proteina u a) α-heliks i b) β-ravan [8]

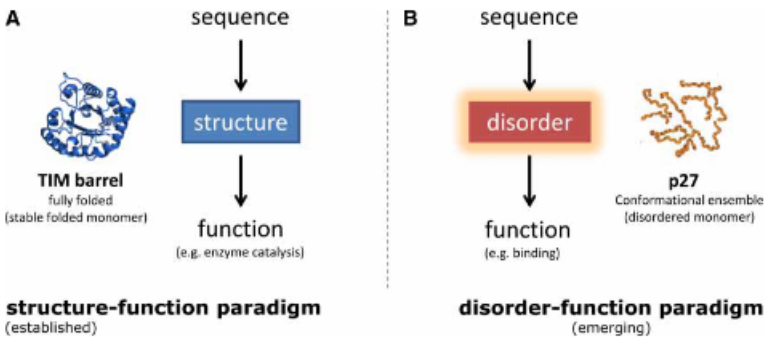


Slika 2: Primer tercijarne strukture proteina [1]

senova dogma) da upravo sekvence aminokiselina određuju trodimenzioni izgled proteina. Dalje, Anfinsen i kolege koje su radile na istraživanju tvrde da trodimenzioni oblik proteina uslovljava njegovu funkciju. Na osnovu rezultata koji su dobijeni kristalografijom hemoglobina i još nekih enzima, teorija je potvrđena. Ova paradigma prikladno je dobila naziv *sekvenca-struktura-funkcija* jer DNK sekvenca određuje strukturu proteina koja dalje određuje funkciju koju će dati protein obavljati u organizmu [2].

Kako je tehnologija napredovala, sve više i više proteina je otkriveno. Ipak, iako se većina novootkrivenih proteina povinovala zakonima koje je Anfinsen ustanovio, postoje i oni proteini čiji regioni (nekad čak i celi proteini) odudaraju od predviđene tercijarne strukture. Naime, postoje proteini nemaju stabilnu sekunsarnu strukturu, tj. može se desiti da ceo protein ili samo neki njegov region (deo proteina) ne zauzima neki od strukturalnih oblika [1]. Postoji mogućnost da se ovi regioni (ili čitavi proteini) sve vreme nalaze u stanju neuređenosti, dok se takođe može desiti da iz neuređenog oblika pređu u uređeni, pa opet u neuređeni. Da bi se opisalo ovakvo ponašanje proteina, uvedena je paradigma *sekvenca-neuređenost-funkcija* [2]. Na slici 3 prikazan je izgled po jednog uređenog i jednog neuređenog proteina. Iako proteini i dalje mogu da vrše svoju funkciju uprkos neuređenosti, pokazuje se da neuređeni regioni mogu biti povezani sa raznim bolestima poput neurodegeneracije i raka, te je potrebno obratiti pažnju na njih [10, 2].

[1] Primeri strukturalnih oblika su gorepomenuti α-heliks ili β-ravan.



Slika 3: Prikaz paradigme sekvenca-struktura-funkcija i sekvenca-neuređenost-funkcija [2]

1.2 Veliki jezički modeli

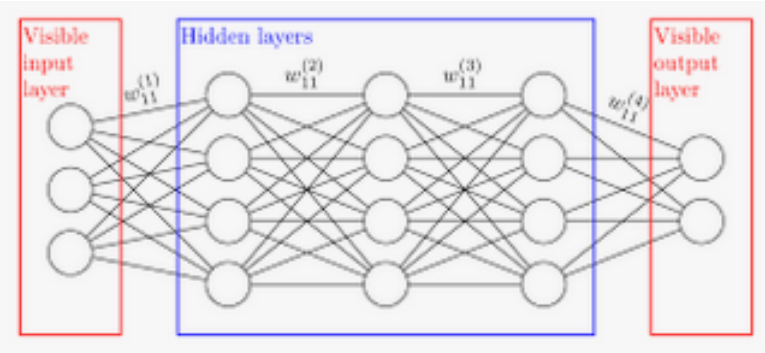
Veliki jezički modeli (eng. LLM) predstavljaju algoritam dubokog učenja zasnovan na posebnom tipu neuronskih mreža (takozvanim *transformerima*) i treniran na ogromnim količinama podataka sa ciljem razumevanja prirodnog jezika ili drugih kompleksnih tipova podataka [4, 6, 7].

Ovi modeli trenirani su na ogromnim skupovima podataka (radi se o milionima ili čak milijardama rečenica) da bi pravilnosti koje treba da nauče što bolje reprezentovale jezik koji treba naučiti i njegova pravila. Stoga je veoma bitno trenirati model na *reprezentativnom* skupu podataka kako bi bile otkrivene pravilnosti koje zaista postoje u jeziku. Nakon što su trenirani, ovi modeli su u mogućnosti da predvide kako treba završiti rečenicu, pa čak imaju i sposobnost da sami generišu rečenice [4].

Da bi ovakvo učenje bilo moguće, koriste se neuronske mreže koje su osmišljene sa ciljem da oponašaju ponašanje ljudskih neurona. Svaka neuronska mreža se sastoji od *ulaznog sloja* (koji obezbeđuje podatke koji će biti obrađivani) i *izlaznog sloja* (koji vraća rezultat izračunavanja). Pored ova dva sloja, u neuronskim mrežama mogu se naći i takozvani *skriveni slojevi*. Njih može biti jedan ili više u zavisnosti od arhitekture mreže. Skriveni slojevi imaju ulogu da transformišu ulaz u nešto što izlazni sloj može da razume i upravo oni omogućavaju mreži da nauči kompleksne reprezentacije podataka. Primer jedne neuronske mreže sa skrivenim slojevima može se videti na slici 4.

Kao što je već napomenuto, LLM za učenje koriste poseban tip neuronskih mreža – transformere. Ono što transformere izdvaja od ostalih tipova neuronskih mreža jeste njihova sposobnost da na osnovu sekvence nauče kontekst, tj. kako elementi zavise jedni od drugih. Oni funkcionišu tako što najpre dobijeni ulaz razdele na *tokene* nad kojim se zatim primenjuju matematičke opercije kako bi se došlo do zaključka u kakvom su odnosu tokeni [6]. Upravo je to razlog što su transformeri u stanju da, nakon završenog procesa treniranja, na osnovu verovatnoće uspešno predvide koja je sledeća reč u rečenici ili koji je sledeći karakter u reči [4, 7].

Veliki jezički modeli našli su svoju primenu u raznim sferama: mogu se koristiti za dobijanje informacija (npr. Google), za prevođenje, za generi-



Slika 4: Primer neuronske mreže sa skrivenim slojevima [5]

sanje tekstualnog sadržaja (ChatGPT), za analizu povratnih informacija (na osnovu ocene usluge i tekstualnog objašnjenja uz pomoć jezičkih modela može se lako utvrditi koje su reči povezane sa pozitivnim, a koje sa negativnim emocijama), za istraživanje DNK i proteina, za marketing i tako dalje [4, 6, 7].

2 Opis problema i dosadašnji rezultati

Nakon upoznavanja sa osnovnim pojmovima, može se pristupiti definisanju problema. Kao što je u poglavlju 1 naznačeno, predikcija neuređenih regiona unutar proteina može biti veoma važna pogotovo u poljima kao što su proizvodnja lekova i inženjering proteina te je veoma bitno razviti tačne i efikasne algoritme koji rešavaju ovaj problem. Međutim, iako se o postojanju neuređenih regiona zna nekoliko decenija, i dalje ne postoje precizne metode za njihovo određivanje [10].

Do sada je napravljeno desetine prediktora koji pokušavaju da reše ovaj problem. No, ono što je zajedničko onim prediktorima koji su se do sada najbolje pokazali jeste činjenica da su svi zasnovani na dubokom učenju. Na primer, AUCpreD je implementiran koristeći duboke konvolutivne neuronske mreže i uslovna nasumična polja, SPOT-Disorder2 uz pomoć dugotrajne kratkoročne memorije, ESpritz preko dvosmernih rekurzivnih neuronskih mreža, a fIDPnn (koji se 2018. godine najbolje pokazao) takođe koristi duboke neuronske mreže kao svoju osnovu [10].

Međutim, ukoliko se obrati pažnja može se videti da jezik proteina ima dosta sličnosti sa ljudskim jezikom: kao što svaki jezik na planeti ima ograničeni broj slova u azbuci, tako i proteini imaju 20 ”slova” svoje "azbuke"(20 aminokisenlina koje mogu da uđu u njihov sastav). I kao što slova neke azbuke poređana u odgovarajućem redosledu daju validne reči jezika, tako i određen redosled aminokiselina daje validne reči jezika proteina (tj. same proteine). Stoga nije teško uvideti da se na proteinima može koristiti metoda *obrade prirodnih jezika* (eng. Natural Language Processing, skraćeno NLP). Jedan od poznatijih alata zasnovan na velikim jezičkim modelima je Evolutionary Scale Modeling (ESM) koji ima ulogu da predviđa strukturu proteina (poput redosleda aminokiselina u polipeptidnom lancu) i čija se konvolutivna mreža sastoji od 34 sloja i koja je obučena na 86 milijardi aminokiselina i više od 250 miliona proteinskih sekvenci [10]. Autori ovog pristupa su otkrili da je ESM veoma

koristan i za predviđanje sekundarne i tercijarne strukture proteina. Zbog ovih važnih otkrića, informatičari sa Univerziteta u Nju Orleansu odlučili su da nastave da se bave efektima koje veliki jezički modeli imaju na predviđanje neuređenih regiona u proteinima. Upravo je s tim ciljem razvijen DisPredict3.0, kojim se ovaj rad bavi.

3 Implementacija

Sam DisPredict3.0 se oslanja na rezultate dobijene fIDPnn metodom na za prikupljanje informacija o proteinu na nivou aminokiselina (primarne strukture) i proteinskih sekvenci. Takođe, nad proteinima je pokrenut i gorepomenuti ESM (koji je prethodno istreniran na milijerdama sekvenci) u cilju dobijanja evolucionih podataka o proteinima. U toku rada ESM-a, prikupljeni su podaci sa sva 34 sloja (jer je eksperimentalno dokazano da se dobijaju bolji rezultati predviđanja ukoliko se u obzir uzmu informacije iz više slojeva umesto samo iz jednog) i rezultati su stavljeni u matricu dimenzija 34x1281. Međutim, pri ekstrakciji podataka javljaju se 2 problema: velika dimenzionalnost podataka i ograničenje ESM metoda na ulaze dužine manje od 1024 tokena. Prvi izazov je premošćen primenom Principal Component Analysis (skraćeno PCA) [2], dok je drugi problem rešen jednostavnom podelom dužih sekvenci na više kraćih delova od kojih je svaki najpre obrađen, pa su dobijeni rezultati ukombinovani u jedan.

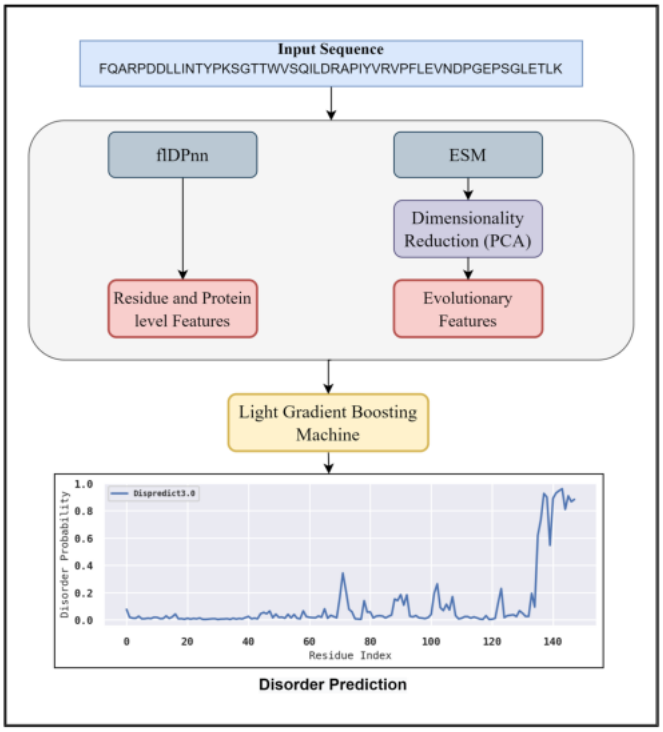
Nakon dobijanja svih potrebnih podataka pokrenut je optimizovan Light Gradient Boosting Machine (LightGBM) [3] sa ciljem da se smanji količina iskorišćene memorije i poveća efikasnost. Na slici 5 mogu se ujedno videti i arhitektura modela i primer dobijenog rezultata za određenu sekvencu.

Zatim, postojeći model je potrebno istrenirati. Model je istreniran i validiran na skupu proteina koji se mogu naći u DisProt bazi podataka (verzija iz juna 2022). Data baza je odabrana jer važi za "zlatni standard" kada su informacije o neuređenim regionima u pitanju [10]. Takođe treba napomenuti da je isti skup podataka korišćen i za treniranje fIDPnn metoda [4]. Podaci iz DisProt baze podeljeni su na trening, test i validacioni skup. No, svaki od njih sadrži dosta više uređenih nego neuređenih regiona, tako da se mora voditi računa o nebalansiranosti klasa. Ovde važnu ulogu igraju izbori vrednosti pragova i hiperparametara. U cilju prevazilaženja problema nebalansiranosti klasa, vrednosti koje treba uzeti kao prag se optimizuju tako da površina ispod dobijene ROC krive bude što veća. Na osnovu vrednosti dobijenih treningom i validacijom, zaključeno je da je optimalna vrednost praga 0.382. Takođe, uporedo su pokrenuti i eksperimenti koji imaju za cilj da utvrde i koji je optimalni broj stabala koje LightGBM generiše da bi predviđanje neuređenosti bilo optimalno. Dobijeni rezultati pokazuju da je to 1000 drveta. Vrednosti za sve ostale parametre su predefinisane vrednosti koje su predložene u scikit-learn biblioteci.

[2] PCA je standardna metoda za smanjivanje dimenzionalnosti podataka uz očuvanje varijanse

[3] Algoritam zasnovan na drvetima odlučivanja koji funkcioniše tako što formira stablo i bira one instance sa velikim gradijentom.

[4] Jedini izuzetak je sekvenca sa oznakom DP01026 koja je uključena u fIDPnn, ali ne i u trening skup DisPredict3.0 metode jer sadrži u sebi jednu neidentifikovanu aminokiselinu.



Slika 5: Primer arhitekture DisPredict3.0 i rezultata izvršavanja za jednu sekvencu [10]

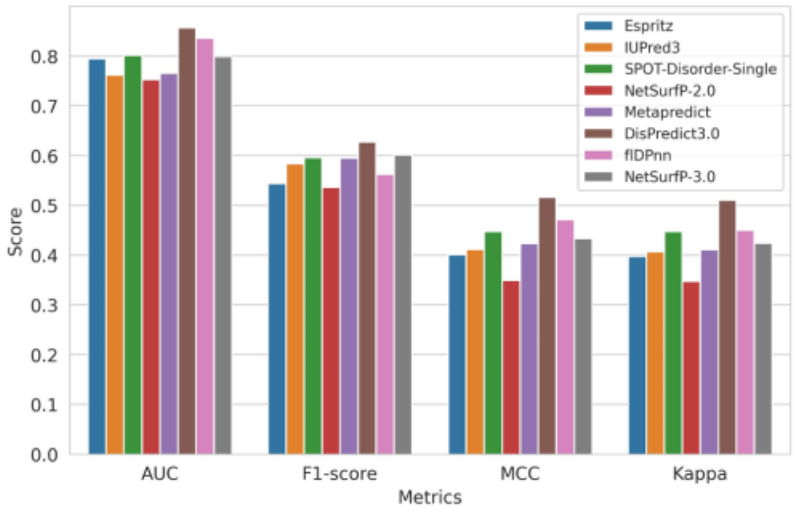
4 Eksperimenti

Naravno, sav rad i trud oko implementacije nove metode pao bi u vodu kada ona ne bi bila dovoljno vremenski efikasna ili ukoliko ne bi davala zadovoljavajuće rezultate. Stoga je potrebno uporediti rezultate dobijene korišćenjem DisPredict3.0 sa ostalim metodama koje se koriste u iste svrhe. Za merenje kvaliteta dobijenih rezultata korišćene su sledeće metrike: AUC (površina ispod ROC krive – što je ona veća, to je model bolji), F1-skor (harmonijska sredina preciznosti [5] i odziva [6]), MCC (metrika za ocenjivanje performansi binarnih klasifikatora, daje bolju sliku od F1-skora) i kappa koeficijent (koji ocenjuje slaganje između 2 klasifikatora). Rezultati poređenja DisPredict3.0 sa ostalim popularnim prediktorima dati su na slici 6. Kao što se sa slike jasno može videti, DisPredict3.0 postigao je najbolje rezultate u svakoj posmatranoj kategoriji. Samim tim možemo zaključiti da je ovaj prediktor bolji od svih ostalih posmatranih.

Takođe treba napomenuti da je DisPredict3.0 učestvovao na proceni prediktora nazvanoj CAID2 2022. godine. Dobri rezultati ni ovde nisu izostali – DisPredict3.0 zauzeo je prvo mesto u predviđanju neuređenih

[5] Preciznost je sposobnost klasifikatora da identifikuje samo relevantne podatke. Meri se kao TP/ (TP + FP), gde TP označava true positive, a FP false positive vrednost

[6] Odziv je udeo relevantnih podataka koji je pronađen i meri se kao TP/(TP + FN), gde je TP true positive, a FN false negative.



Slika 6: Metrike koje ocenjuju kvalitet prediktora [10]

regiona proteina koji su pripadali NOX (No X-ray) skupu podataka i visoko šesto mesto (od preko 40 prediktora) u predviđanju povezanosti aminokiselina. Sa druge strane, prediktor se nije proslavio u predviđanju neuređenosti proteina iz PDB baze podataka, no to je i bilo za očekivati s obzirom da je treniran na DisProt skupu podataka, a testiran na PDB skupu podataka koji sadrži drugačiju anotaciju.

Ostalo je još ispitati kako DisProt funkcioniše u slučajevima kada je veći deo proteinske sekvence neuređen i kada je ceo protein neuređen. Dobijeni rezultati upoređeni su sa rezultatima koje ostvaruje fIDPnn metoda (jer se ona do sada pokazala najboljom). Dobijeni rezultati su i više nego zadovoljavajući – DisPredict3.0 se dosta bolje pokazao u slučajevima kada je protein bio većinski neuređen, dok su se oba prediktora dobro pokazala u slučajevima potpune neuređenosti.

Ono što je takođe veoma značajo jeste vreme izvršavanja. Svi eksperimenti obavljeni su na operativnom sistemu Linux, na 64 jezgra i 768GB RAM memorije. Najveći deo vremena potrošen je za učitavanje velikog jezičkog modela u memoriju da bi se model istrenirao (sveukupno trajanje oko 6 minuta), dok se proteini koji se zatim testiraju relativno brzo obrađuju (oko 3 minuta po proteinu). No, ovi rezultati se mogu poboljšati paralelizacijom, tj. korišćenjem više procesora istovremeno. Eksperimentima sa brojem korišćenih procesora ustanovljeno je da se značajno ubrzanje pojavljuje kada se koristi 10 procesora (tada se vreme obrade jednog proteina svede na svega 7 sekundi), dok se daljim povećanjem broja procesora ne dobijaju značajno bolji rezultati.

5 Pokušaj instalacije i pokretanja

Nakon upoznavanja sa implementacionim detaljima alata i njegovim dosadašnjim rezultatima, moguće je preći na proces instalacije i pokretanja sa ciljem da se demonstrira rad alata na praktičnom primeru.

5.1 Github

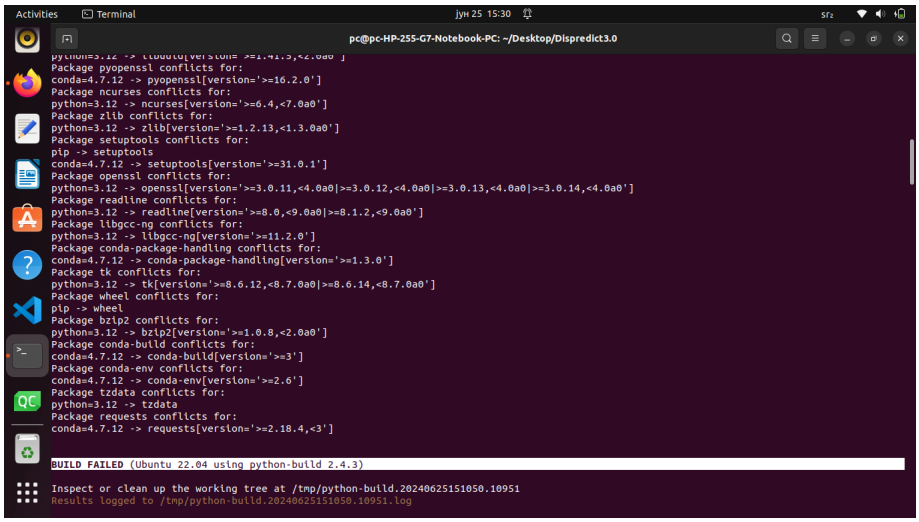
Za pokretanje koda koji je dostupan na Github stranici autora, najpre je potrebno preuzeti kod komandom

```
1000 git clone https://github.com/wasicse/Dispredict3.0.git
```

Nakon uspešnog kloniranja, na lokalnom računaru se sada nalazi kopija alata. Najpre je potrebno pozicionirati se u taj direktorijum i nakon toga instalirati sve potrebne zavisnosti komandom

```
1000 ./install_dependencies.sh
```

Međutim, može se videti da komande koje se nalaze u datoj skripti proizvode grešku. Uzrok tome je što je skripta pokrenuta na Linux-ovom Ubuntu 22.04 operativnom sistemu, dok je rad testiran takođe na Linux operativnom sistemu, ali na verziji 20.04 (slika 7). Razlika između ove dve verzije Linux-a je verzija Pajtona na kojoj su zasnovane, zbog čega i dolazi do greške.



Slika 7: Ispis greške prilikom instalacije zavisnosti na Ubuntu 22.04

Stoga je, kao rešenje problema, pokušana instalacija ovih zavisnosti na verziji koja je i korišćena u svrhe pisanja rada – Ubuntu 20.04. Ovoga puta nije bilo problema. Sada je samo potrebno pokrenuti obezbeđeni test primer koji je došao uz alat. To se radi komandama

```
1000 cd script
../.venv/bin/poetry run python Dispredict3.0.py -f "../example/
     sample.fasta" -o "../output/"
```

Međutim, ovaj put se dobija greška o nepostojanju biblioteke PyTorch. Zaista, ako se baci pogled na pyproject.toml fajl lako se uočava da među

zavisnostima zaista nema PyTorch modula. Sadržaj pomenute datoteke može se videti na slici 8. Da bi se rešio ovaj problem, dodat je pomenuti modul u listu zavisnosti (slika 9).

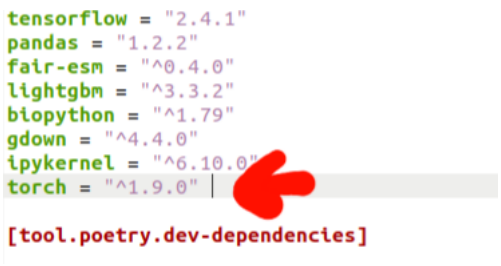
```
[tool.poetry]
name = "dispredict3.0"
version = "0.1.0"
description = "Dispredict3.0: Prediction of Intrinsically Disordered Proteins with Protein Language Model"
authors = ["Md Wasi Ul Kabir <mkabir3@uno.edu>"]
license = "MIT"

[tool.poetry.dependencies]
python = ">=3.7.1"
plotly = "4.14.3"
scikit-learn = "0.23.1"
keras = "2.4.3"
tensorflow = "2.4.1"
pandas = "1.2.2"
fair-esm = "^0.4.0"
lightgbm = "^3.3.2"
biopython = "^1.79"
gdown = "^4.4.0"
ipykernel = "^6.10.0"

[tool.poetry.dev-dependencies]

[build-system]
requires = ["poetry-core>=1.0.0"]
build-backend = "poetry.core.masonry.api"
```

Slika 8: Sadržaj pyproject.toml datoteke



Slika 9: Sadržaj pyproject.toml datoteke nakon dodavanja novih uslovnosti

Gorenavedene komande pokrenute su ponovo i sada ne dolazi do greške sa nepostojećim modulima, ali se javljeju novi problemi – server sa kog se dovlače podaci ne radi (slika 10). Zaista, pri pokušaju povezivanja na server preko veba, dobija se ista poruka (slika 11).

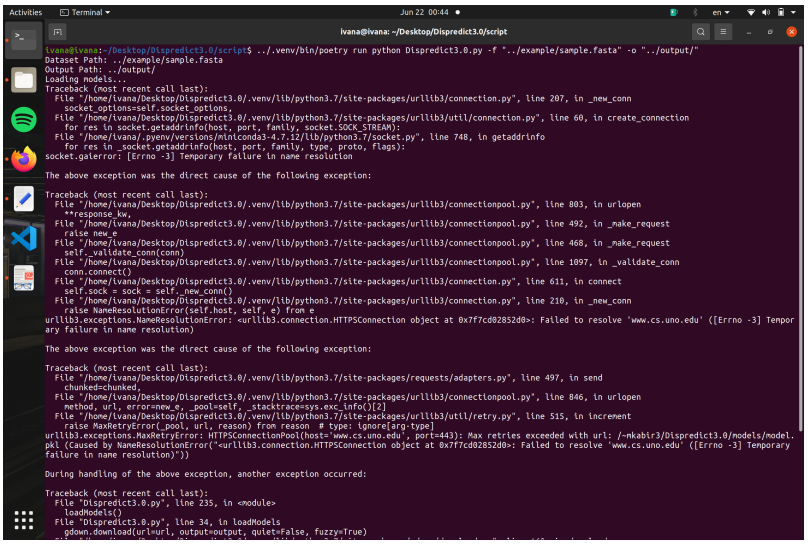
5.2 Docker

Nakon neuspešnog pokretanja koda koji je javno dostupan, sledeća moguća opcija je Docker. Instalacija pomenutog alata vrši se komandom

```
1000 sudo apt install docker
```

Nakon instalacije Docker-a, može se preuzeti kod komandom

```
1000 docker run -ti --name dispredict3.0  wasicse/dispredict3.0:latest
```



Slika 10: Greška nastala zbog nedostupnosti servera

Slika 11: Provera dostupnosti servera

Međutim, vidimo da i ona proizvodi grešku (slika 12).

Slika 12: Greška prilikom pokretanja alata Docker

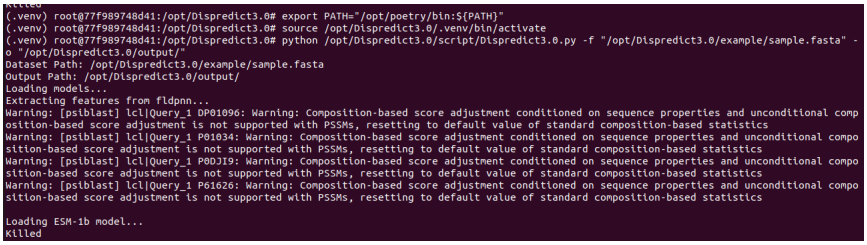
Za otklanjanje problema potrebno je pokrenuti sledeće komande redom:

```
sudo groupadd docker
sudo usermod -aG docker ${USER}
sudo chmod 666 /var/run/docker.sock
sudo systemctl restart docker
docker run -ti --name dispredict3.0  wasicse/dispredict3.0:latest
```

Sada je docker spreman za korišćenje. Za pokretanje alata sada je još potrebno izvršiti sledeće komande:

```
export PATH="/opt/poetry/bin:${PATH}"
source /opt/Dispredict3.0/.venv/bin/activate
python /opt/Dispredict3.0/script/Dispredict3.0.py -f "/opt/
    Dispredict3.0/example/sample.fasta" -o "/opt/Dispredict3.0/
    output/"
```

No, i ovaj pristup dovodi do greške – proces biva ubijen tokom izvršavanja (slika 13). Ista greška dobija se na oba računara.



Slika 13: Pokrenuti proces biva terminiran

5.3 Singularity

Poslednji mogući način za preuzimanje i pokretanje alata DisPredict3.0 jeste uz pomoć Singularity alata. Najpre je potrebno preuzeti Singularity i Go programski jezik. To je moguće uraditi komandama:

```
sudo yum update -y && \
    sudo yum groupinstall -y 'Development Tools' && \
    sudo yum install -y \
    openssl-devel \
    libuuid-devel \
    libseccomp-devel \
    wget \
    squashfs-tools
// instalacija programskog jezika Go
sudo apt-get update
wget https://go.dev/dl/go1.21.0.linux-amd64.tar.gz
sudo tar -xvf go1.21.0.linux-amd64.tar.gz
sudo mv go /usr/local
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
source ~/.profile

// instalacija alata Singularity
echo 'export GOPATH=${HOME}/go' >> ~/.bashrc && \
    echo 'export PATH=/usr/local/go/bin:${PATH}:${GOPATH}/bin' >>
    ~/.bashrc && \
    source ~/.bashrc
go get -u github.com/golang/dep/cmd/dep
export VERSION=v3.0.3 # or another tag or branch if you like && \
    cd $GOPATH/src/github.com/sylabs/singularity && \
    git fetch

./mconfig && \
    make -C ./builddir && \
    sudo make -C ./builddir install

// pokretanje alata DisPredict3.0
singularity pull dispredict3.sif docker://wasicse/dispredict3.0
singularity run --writable-tmpfs dispredict3.sif

export PATH="/opt/poetry/bin:${PATH}"
source /opt/Dispredict3.0/.venv/bin/activate
python /opt/Dispredict3.0/script/Dispredict3.0.py -f "/opt/
    Dispredict3.0/example/sample.fasta" -o "/opt/Dispredict3.0/
    output/"
```

Tokom pokretanja alata DisPredict3.0 dobija se ista greška kao i u slučaju sa alatom Docker (vidi sliku 13). Ista greška ponovo se dobija na oba računara.

6 Zaključak

Problem nalaženja neuređenih sekvenci je veoma važno pitanje u oblasti proteina. Njegovo uspešno rešavanje značajno bi doprinelo u oblasti proizvodnje lekova. DisPredict3.0 upravo predstavlja pristup rešavanju navedenog problema. Iako se dosta prediktora u poslednjih par godina zasnivalo na neuronskim mrežama i dubokom učenju, veliki jezički modeli i transformatori (koje koristi DisPredict3.0) predstavljaju novinu u pokušajima predviđanja neuređenosti proteina. Sprovedeni eksperimenti su pokazali da je ovaj metod superioran u odnosu na većinu dosad postojećih modela. Za neke skupove podataka čak se pokazao i kao najbolji. Takođe, sve bitne metrike ukazuju da je pouzdaniji od ostalih modela. Jedina slaba tačka jeste vreme izvršavanja jer se dosta vremena troši na samo učitavanje velikog jezičkog modela. No, paralelizacijom se i ovaj problem rešava. Za dalji razvoj i eventualno dobijanje još boljih rezultata postoji mogućnost podešavanja parametara velikog jezičkog modela, pa je to jedan od mogućih pravaca daljeg razvoja u ovoj oblasti.

Literatura

[1] Sagar Aryal. Protein Structure- Primary, Secondary, Tertiary, and Quaternary, 2022. on-line at: https://microbenotes.com/protein-structure-primary-secondary-tertiary-and-quaternary/.

[2] Madan M. Babu. The contribution of intrinsically disordered regions to protein function, cellular complexity, and human disease, 2016. on-line at: https://sci-hub.se/10.1042/bst20160172.

[3] Engelbert Buxbaum. *Fundamentals of Protein Structure and Function*. Springer, 2015.

[4] Cloudflare. What is a large language model (LLM)? on-line at: https://www.cloudflare.com/learning/ai/what-is-large-language-model/.

[5] DeepAI. Hidden Layer. on-line at: https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning.

[6] elastic. What is a large language model (llm)? on-line at: https://www.elastic.co/what-is/large-language-models.

[7] IBM. What are LLMs? on-line at: https://www.ibm.com/topics/large-language-models.

[8] Greg Lever. *Large Scale Quantum Mechanical Enzymology*. University of Cambridge, 2014.

[9] Jeffrey Skolnick and Jacquelyn S. Fetrow. From genes to protein structure and function: novel applications of computational approaches in the genomic era, 2019. on-line at: https://sci-hub.se/10.1016/s0167-7799(99)01398-0.

[10] Wasi Ul Kabir and Tamjidul Hoque. Dispredict3.0: Prediction of intrinsically disordered regions/proteins using protein language model. *Applied Mathematics and Computation*, 472, 2024.