Točnost pretraživanja, clustering i klasifikacija

Kapec, Ivan

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Ivan Kapec

TOČNOST PRETRAŽIVANJA, CLUSTERING I KLASIFIKACIJA

Diplomski rad

Voditelj rada:
doc. dr. sc. Pavle Goldstein

Zagreb, veljača, 2021.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Ocu, jer mi je pomogao kad god sam ga to tražio.
Majci, jer je uvijek bila tu kad sam je trebao.
Petri, jer je sa mnom prošla kroz moje poraze.
Zahvaljujem se svom mentoru, čije su se strpljenje, trud i vodstvo u svakom pogledu pokazali ključnima.

Sadržaj

Uvod

Problem traženja proteina koji pripadaju istoj proteinskoj familiji još uvijek je otvoreno pitanje u molekularnoj biologiji. Zbog velikog povećanja podataka o proteinima dobivenih sekvenciranjem genoma, postoji potreba za pouzdanim, automatskim metodama za takvu klasifikaciju proteina, s obzirom na to da su ručne metode skupe i dugotrajne. Bioinformatika je znanstvena disciplina na križanju računarstva, biologije i statistike koja se između ostalog bavi upravo istraživanjem novih metoda za klasifikaciju proteina u proteinske familije.

U proteinskoj familiji nalaze se proteini koji imaju zajedničko evolucijsko podrijetlo. Proteom je skup svih proteina u nekom organizmu. Protein je izuzetno složena molekula, prisutna u svim živim bićima, koja se sastoji od aminokiselina. Upravo sličnost između nizova aminokiselina može ukazivati na pripadnost proteina nekoj proteinskoj familiji. Iterativno pretraživanje proteoma jedan je od standardnih modela koji se koristi za dobivanje skupa proteina koji pripadaju istoj proteinskoj familiji, međutim on koristi koncept sličnosti, pa ima određenu uspješnost.

U ovom radu istražuje se metoda koja bi nadopunom modela iterativnog pretraživanja poboljšala njegovu uspješnost, a temelji se na pridruživanju grafa dobivenom skupu proteina, odnosno skupu kraćih nizova aminokiselina (tzv. motiva). U takvom grafu, egzaktni algoritam za traženje najveće klike već se pokazao korisnim u povećavanju uspješnosti cijelog modela. Međutim, takvo rješenje je iznimno sporo za veći broj podataka, stoga će se ovdje promatrati učinkovitost algoritma za traženje aproksimativne najveće klike u problemu klasifikacije proteina.

U prvom poglavlju ovog rada definirane su mjere uspješnosti i osnovni pojmovi iz teorije vjerojatnosti i matematičke statistike koji su nužni za razumijevanje rada. U drugom poglavlju izloženi su korišteni algoritmi za traženje klike nakon definicija potrebnih pojmova iz teorije grafova. U trećem poglavlju predstavljena je biološka pozadina, iterativno pretraživanje proteoma, način i cilj pridruživanja grafa ovom problemu. U četvrtom poglavlju prikazani su dobiveni rezultati na proteomima talijinog uročnjaka, krumpira, rajčice i šećerne repe. Konačno, u posljednjem poglavlju, temeljem dobivenih rezultata razmotrena je učinkovitost upotrebe algoritma za traženje aproksimativne najveće klike u ovom problemu.

Poglavlje 1

Vjerojatnost i statistika

1.1 Vjerojatnosni prostor

Definicija 1.1.1. ***Slučajni pokus*** *ili* ***slučajni eksperiment*** *je pokus čiji ishodi, tj. rezultati nisu jednoznačno određeni uvjetima u kojima izvodimo pokus.*

Definicija 1.1.2. ***Prostor elementarnih događaja*** Ω *je neprazan skup koji reprezentira skup svih ishoda slučajnog pokusa. Elemente* ω *skupa* Ω *nazivamo* ***elementarni događaji****.*

Definicija 1.1.3. *Familija* $\mathcal{F}$ *podskupova od* Ω ($\mathcal{F} \subset \mathcal{P}(\Omega)$) *je* σ***-algebra skupova*** *na* Ω *ako je:*

$$(i) \quad \emptyset \in \mathcal{F}$$
$$(ii) \quad A \in \mathcal{F} \implies A^c \in \mathcal{F}$$
$$(iii) \quad A_i \in \mathcal{F}, i \in \mathbb{N} \implies \bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$$

Definicija 1.1.4. *Neka je* $\mathcal{F}$ σ*-algebra na skupu* Ω*. Uređen par* $(\Omega, \mathcal{F})$ *zove se* ***izmjeriv prostor****.*

Definicija 1.1.5. *Neka je* $(\Omega, \mathcal{F})$ *izmjeriv prostor. Funkcija* $\mathbb{P} : \mathcal{F} \to \mathbb{R}$ *je* ***vjerojatnost*** *(na* $\mathcal{F}$*, na* Ω*) ako vrijedi:*

(i) $\mathbb{P}(A) \geq 0, \forall A \in \mathcal{F}$ (*nenegativnost*)

(ii) $\mathbb{P}(\Omega) = 1$ (*normiranost*)

(iii) $A_i \in \mathcal{F},\ i \in \mathbb{N}\ i\ A_i \cap A_j = \emptyset\ za\ i \neq j \implies \mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$ (σ*-aditivnost*)

Definicija 1.1.6. *Uređena trojka* $(\Omega, \mathcal{F}, \mathbb{P})$*, gdje je* $\mathcal{F}$ σ*-algebra na* Ω*, a* $\mathbb{P}$ *je vjerojatnost na* $\mathcal{F}$*, zove se* ***vjerojatnosni prostor****.*

1.2 Uvjetna vjerojatnost i nezavisnost

Definicija 1.2.1. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i $A \in \mathcal{F}$ takav da je $\mathbb{P}(A) > 0$. Definirajmo funkciju $\mathbb{P}_A : \mathcal{F} \to [0,1]$ na sljedeći način:*

$$\mathbb{P}_A(B) = \mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}, \quad B \in \mathcal{F} \tag{1.1}$$

*$\mathbb{P}_A$ je vjerojatnost na $\mathcal{F}$ i zovemo je **uvjetna vjerojatnost uz uvjet A**. Broj $\mathbb{P}(B|A)$ zovemo **vjerojatnost od B uz uvjet A**.*

Definicija 1.2.2. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i $A_i \in \mathcal{F}$, $i \in I$ proizvoljna familija događaja. Kažemo da je to **familija nezavisnih događaja** ako za svaki konačan podskup različitih indeksa $i_1, i_2, \ldots, i_k \in I$ vrijedi:*

$$\mathbb{P}(\bigcap_{j=1}^{k} A_{i_j}) = \prod_{j=1}^{k} \mathbb{P}(A_{i_j}) \tag{1.2}$$

1.3 Slučajna varijabla i funkcija distribucije

Definicija 1.3.1. *Neka je S proizvoljan neprazan skup i $\mathcal{A}$ familija podskupova od S ($\mathcal{A} \subset \mathcal{P}(S)$). Sa $\sigma(\mathcal{A})$ označimo najmanju σ-algebru podskupova od S koja sadrži $\mathcal{A}$. Nju nazivamo **σ-algebra generirana sa $\mathcal{A}$**.*

Definicija 1.3.2. *Neka je sa $\mathcal{B}$ označena σ-algebra generirana familijom svih otvorenih skupova na $\mathbb{R}$. $\mathcal{B}$ zovemo **σ-algebra Borelovih skupova na $\mathbb{R}$**, a elemente σ-algebre $\mathcal{B}$ zovemo **Borelovi skupovi**.*

Definicija 1.3.3. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor. Funkcija $X : \Omega \to \mathbb{R}$ je **slučajna varijabla** (na Ω) ako je $X^{-1}(B) \in \mathcal{F}$ za proizvoljno $B \in \mathcal{B}$, tj. $X^{-1}(\mathcal{B}) \subset \mathcal{F}$.*

Definicija 1.3.4. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i X slučajna varijabla na Ω. Za $B \in \mathcal{B}$ definiramo funkciju $\mathbb{P}_X : \mathcal{B} \to [0,1]$ relacijom:*

$$\mathbb{P}_X(B) = \mathbb{P}(X^{-1}(B)) = \mathbb{P}\{\omega \in \Omega : X(\omega) \in B\} = \mathbb{P}\{X \in B\} \tag{1.3}$$

*$\mathbb{P}_X$ zovemo **vjerojatnosna mjera inducirana sa X**, a vjerojatnosni prostor $(\mathbb{R}, \mathcal{B}, \mathbb{P}_X)$ zovemo **vjerojatnosni prostor induciran sa X**. $\mathbb{P}_X$ često zovemo i **zakon razdiobe od X**.*

Definicija 1.3.5. *Neka je X slučajna varijabla na Ω. **Funkcija distribucije od X** je funkcija $F_X : \mathbb{R} \to [0,1]$ definirana sa:*

$$F_X(x) = \mathbb{P}_X(\langle -\infty, x]) = \mathbb{P}(X^{-1}(\langle -\infty, x])) = \mathbb{P}\{\omega \in \Omega : X(\omega) \leq x\} = \mathbb{P}\{X \leq x\}, \quad x \in \mathbb{R}$$

Napomena 1.3.6. *Ako je jasno o kojoj se slučajnoj varijabli radi, piše se F umjesto F_X.*

Teorem 1.3.7. *Funkcija distribucije F slučajne varijable X je rastuća i neprekidna zdesna na $\mathbb{R}$, te zadovoljava:*

$$\begin{aligned} F(-\infty) &= \lim_{x\to-\infty} F(x) = 0 \\ F(+\infty) &= \lim_{x\to+\infty} F(x) = 1. \end{aligned} \tag{1.4}$$

Funkciju $F : \mathbb{R} \to [0, 1]$ koja ima prethodna svojstva zovemo **vjerojatnosna funkcija distribucije** (na $\mathbb{R}$) ili kraće, **funkcija distribucije**.

Definicija 1.3.8. *Funkcija $g : \mathbb{R} \to \mathbb{R}$ je **Borelova funkcija** ako je $g^{-1}(B) \in \mathcal{B}$ za svako $B \in \mathcal{B}$, tj. ako je $g^{-1}(\mathcal{B}) \subset \mathcal{B}$.*

Definicija 1.3.9. *Slučajna varijabla X je **diskretna**, ako postoji konačan ili prebrojiv skup $D \subset \mathbb{R}$ takav da je $\mathbb{P}\{X \in D\} = 1$.*

Definicija 1.3.10. *Neka je X slučajna varijabla na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ i neka je F_X njezina funkcija distribucije. Kažemo da je X **apsolutno neprekidna** ili kraće, **neprekidna slučajna varijabla** ako postoji nenegativna realna Borelova funkcija f na $\mathbb{R}$ ($f : \mathbb{R} \to \mathbb{R}_+$) takva da je*

$$F_X(x) = \int_{-\infty}^{x} f(t)d\lambda(t), \quad x \in \mathbb{R} \tag{1.5}$$

Ako je X neprekidna slučajna varijabla, tada se funkcija f iz (1.5) zove **funkcija gustoće vjerojatnosti od X**, tj. od njezine funkcije distribucije F_X ili kraće, **gustoća od X** i ponekad je označavamo sa f_X.

1.4 Matematičko očekivanje i varijanca

Definicija matematičkog očekivanja provodi se u 3 koraka. Prvo se definira matematičko očekivanje jednostavne slučajne varijable, zatim nenegativne slučajne varijable i na kraju općenite slučajne varijable. Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor.

Definicija 1.4.1. *Neka je X slučajna varijabla na $(\Omega, \mathcal{F}, \mathbb{P})$. X je **jednostavna slučajna varijabla** ako je njezino područje vrijednosti konačan skup.*

Označimo sa $\mathcal{K}$ skup svih jednostavnih slučajnih varijabli definiranih na Ω, a sa $\mathcal{K}_+$ skup svih nenegativnih funkcija iz $\mathcal{K}$.

Neka je $X \in \mathcal{K}$, $X = \sum_{k=1}^{n} x_k \mathcal{K}_{A_k}$, gdje su $A_1, A_2, \ldots, A_n \in \mathcal{F}$ međusobno disjunktni.

Definicija 1.4.2. ***Matematičko očekivanje*** *od X ili kraće,* ***očekivanje*** *od X označavamo sa* $\mathbb{E}[X]$ *i definira se sa:*

$$\mathbb{E}[X] = \sum_{k=1}^{n} x_k \mathbb{P}(A_k). \tag{1.6}$$

Neka je sada X **nenegativna slučajna varijabla** definirana na Ω. Tada postoji rastući niz $(X_n)_{n\in\mathbb{N}}$ nenegativnih jednostavnih slučajnih varijabli takav da je $X = \lim_{n\to\infty} X_n$. Niz $(\mathbb{E}[X_n])_{n\in\mathbb{N}}$ je rastući niz u $\mathbb{R}_+$, dakle postoji $\lim_{n\to\infty} \mathbb{E}[X_n]$ koji može biti jednak i $+\infty$.

Definicija 1.4.3. ***Matematičko očekivanje*** *od X ili kraće,* ***očekivanje*** *od X definira se sa*

$$\mathbb{E}[X] = \lim_{n\to\infty} X_n. \tag{1.7}$$

Neka je sada napokon X **proizvoljna slučajna varijabla** na Ω. Vrijedi $X = X^+ - X^-$, gdje su X^+, X^- slučajne varijable i $X^+, X^- \geq 0$.

Definicija 1.4.4. *Kažemo da* ***matematičko očekivanje*** *od X ili kraće,* ***očekivanje*** *od X* ***postoji*** *(ili da je definirano) ako je barem jedna od veličina* $\mathbb{E}[X^+], \mathbb{E}[X^-]$ *konačna, tj. vrijedi* $min\{\mathbb{E}[X^+], \mathbb{E}[X^-]\} < +\infty$. *Tada po definiciji stavljamo*

$$\mathbb{E}[X] = \mathbb{E}[X^+] + \mathbb{E}[X^-]. \tag{1.8}$$

Definicija 1.4.5. *Neka je X slučajna varijabla na* $(\Omega, \mathcal{F}, \mathbb{P})$ *i neka je* $\mathbb{E}[X]$ *konačno. Tada definiramo* ***varijancu*** *od X koju označavamo sa* $Var(X)$ *ili* σ_X^2 *na sljedeći način:*

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]. \tag{1.9}$$

Napomena 1.4.6. *Pozitivan drugi korijen iz varijance nazivamo* ***standardna devijacija*** *i označavamo sa* σ_X.

1.5 Primjeri slučajnih varijabli

Eksponencijalna distribucija

Neprekidna slučajna varijabla X ima **eksponencijalnu distribuciju** s parametrom $\lambda > 0$ ako joj je funkcija gustoće f zadana sa:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \leqslant 0. \end{cases} \tag{1.10}$$

Logistička distribucija

Neka su $\alpha, \beta \in \mathbb{R}, \beta > 0$. Neprekidna slučajna varijabla X ima **logističku distribuciju** s parametrima α i β ako joj je funkcija gustoće f zadana sa:

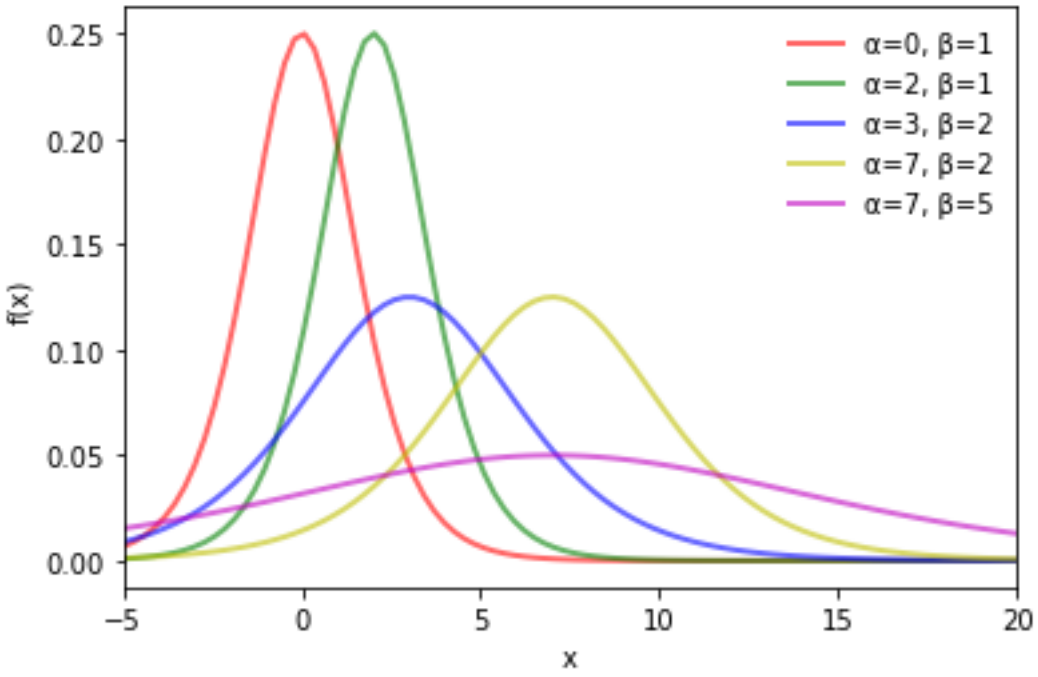
$$f(x) = \frac{e^{-\frac{x-\alpha}{\beta}}}{\beta\left(1 + e^{-\frac{x-\alpha}{\beta}}\right)^2}, \quad x \in \mathbb{R}. \tag{1.11}$$

Kažemo da X ima **standardnu logističku distribuciju** ako je $\alpha = 0$ i $\beta = 1$.

Neka su $\alpha, \beta > 0$. Neprekidna slučajna varijabla X ima **generaliziranu logističku distribuciju** ako joj je funkcija gustoće f zadana sa:

$$f(x) = \frac{1}{B(\alpha,\beta)} \frac{e^{-\beta x}}{(1 + e^{-x})^{\alpha+\beta}}, \quad x \in \mathbb{R}, \tag{1.12}$$

gdje je funkcija B definirana sa $B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt, \quad x, y > 0.$



Slika 1.1: Graf funkcije gustoće logističke distribucije za razne vrijednosti parametara

Teorija ekstremnih vrijednosti

Neka je $\alpha \in \mathbb{R}$ i $\beta > 0$. Slučajna varijabla X ima **Gumbelovu distribuciju** s parametrima α i β ako joj je funkcija gustoće f zadana sa:

$$f(x) = \frac{1}{\beta} e^{-\frac{x-\alpha}{\beta} - e^{-\frac{x-\alpha}{\beta}}}, \quad x \in \mathbb{R}. \tag{1.13}$$

Neka je $p > 0$. Slučajna varijabla X ima **generaliziranu Gumbelovu distribuciju** ako joj je funkcija gustoće f zadana sa:

$$f(x) = \frac{1}{\Gamma(p)} e^{-px} e^{-e^{-x}}, \quad x \in \mathbb{R}, \tag{1.14}$$

gdje je funkcija Γ definirana sa: $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \quad x > 0.$

Teorem 1.5.1. *Neka su X_1 i X_2 nezavisne slučajne varijable s generaliziranom Gumbelovom distribucijom s parametrima p i q, respektivno. Tada, slučajna varijabla $Y = X_1 - X_2$ ima generaliziranu logističku distribuciju (iz 1.12) s parametrima p i q.*

Teorem 1.5.2 (Fisher-Tippett (1928.), Gnedenko (1943.))**.** *Neka su $X_1, X_2, \ldots, X_n$ nezavisne, jednako distribuirane slučajne varijable i neka je $M_n = max(X_1, X_2, \ldots, X_n)$. Ako postoje konstante $a_n \in \mathbb{R}$, $b_n > 0$ i nedegenerirana funkcija distribucije H takva da je:*

$$\lim_{n \to +\infty} P\left(\frac{M_n - a_n}{b_n} \leqslant x\right) = H(x), \tag{1.15}$$

odnosno:

$$\frac{M_n - a_n}{b_n} \xrightarrow{\mathcal{D}} H, \quad n \to +\infty, \tag{1.16}$$

tada granična distribucija H pripada jednoj od tri distribucije ekstremnih vrijednosti: Gumbelovoj, Fréchetovoj ili Weibullovoj distribuciji.

1.6 Mjere uspješnosti modela klasifikacije

Klasifikacija

U statistici, **klasifikacija** je problem identificiranja kojoj od skupa kategorija (klasa) pripada neka nova opservacija, na temelju skupa poznatih podataka koji sadrže opservacije (ili instance) čija je kategorija već određena. Primjer bi bio klasifikacija proteina u određenu proteinsku familiju, gdje postoje samo 2 kategorije (klase) označene brojevima 0 i 1:

0: Novi protein nije dovoljno sličan proteinima iz proteinske familije
1: Novi protein je dovoljno sličan proteinima iz proteinske familije

Na temelju poznatih podataka (proteina), model u ovom primjeru pomoću određene funkcije sličnosti označava nove podatke (proteine) oznakama 0 ili 1 sa određenom uspješnošću.

Mjere uspješnosti

Da bi se ocijenila uspješnost nekog modela, definirane su mjere uspješnosti modela. One se temelje na pojmovima iz matrice uspješnosti (eng. *confusion matrix*) prikazanoj sljedećom tablicom.

		Predviđeno stanje		
	Ukupna populacija	Ocijenjeni pozitivno	Ocijenjeni negativno	
Stvarno stanje	Pozitivno stanje (CP)	TP (stvarno pozitivni)	FN (lažno negativni)	Osjetljivost (TPR)
	Negativno stanje (CN)	FP (lažno pozitivni)	TN (stvarno negativni)	Specifičnost (TNR)
		Preciznost (PPV)	Negativna prediktivna vrijednost (NPV)	

Tablica 1.1: Tablica uspješnosti

Napomena 1.6.1. *U ovom radu će se provjera broja TP (eng. True Positives) i ostalih brojeva iz matrice uspješnosti (FP, FN, TN) vršiti pomoću liste CP (eng. Condition Positive). Lista CP sadrži sve proteine za koje je pripadnost određenoj familiji već utvrđena. Dakle, u savršenom modelu bi svi proteini sa liste CP imali oznaku 1, a svi proteini koji nisu na listi CP bi imali oznaku 0.*

Slijede definicije nekih od mjera uspješnosti modela za binarnu klasifikaciju:

Osjetljivost ili **TPR** (eng. *True Positive Rate*) je postotak pozitivnih elemenata uzorka u odnosu na određeno stanje, odnosno CP elemenata uzorka, koji su ispravno prepoznati kao pozitivni.

$$\text{TPR} = \frac{\text{broj stvarno pozitivnih}}{\text{broj stvarno pozitivnih} \ + \ \text{broj lažno negativnih}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{CP}} \tag{1.17}$$

Specifičnost ili **TNR** (eng. *True Negative Rate*) je postotak negativnih elemenata uzorka u odnosu na određeno stanje, odnosno CN (eng. *Condition Negative*) elemenata uzorka, koji su ispravno prepoznati kao negativni.

$$\text{TNR} = \frac{\text{broj stvarno negativnih}}{\text{broj stvarno negativnih } + \text{ broj lažno pozitivnih}} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{TN}}{\text{CN}} \tag{1.18}$$

Preciznost ili **PPV** (eng. *Positive Predictive Value*) je omjer broja stvarno pozitivnih elemenata uzorka i broja elemenata uzorka koji su modelom prepoznati kao pozitivni.

$$\text{PPV} = \frac{\text{broj stvarno pozitivnih}}{\text{broj stvarno pozitivnih } + \text{ broj lažno pozitivnih}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1.19}$$

Negativna prediktivna vrijednost ili **NPV** (eng. *Negative Predictive Value*) je omjer broja stvarno negativnih elemenata uzorka i broja elemenata uzorka koji su modelom prepoznati kao negativni.

$$\text{NPV} = \frac{\text{broj stvarno negativnih}}{\text{broj stvarno negativnih } + \text{ broj lažno negativnih}} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{1.20}$$

F_β**-score** je mjera uspješnosti modela koja povezuje osjetljivost i preciznost. Dobiva se kao harmonijska sredina osjetljivosti i preciznosti modela, uz težinski faktor β.

$$F_\beta = \frac{\left(\beta^2 + 1\right) \cdot PPV \cdot TPR}{\beta^2 \cdot PPV + TPR} \tag{1.21}$$

Najčešće se koristi F_1**-score** ($\beta = 1$):

$$F_1 = \frac{2 \cdot PPV \cdot TPR}{PPV + TPR} \tag{1.22}$$

Napomena 1.6.2. *Sve navedene mjere postižu vrijednosti isključivo na intervalu [0, 1]. Model je uspješniji po nekoj od navedenih mjera, što je ta mjera bliže broju 1. Model sa mjerom 0 bi bio loše ocijenjen model.*

Pojmovi iz ovog poglavlja preuzeti su iz izvora [7], [5], [8] i [4].

Poglavlje 2

Teorija grafova

2.1 Graf

Definicija 2.1.1. ***Graf*** *G je uređeni par $G = (V, E)$, gdje je $V \neq \emptyset$ skup vrhova, a E je skup 2-podskupova od V, koje nazivamo* ***bridovi****.*

Definicija 2.1.2. *Kažemo da su vrhovi $u, v \in V$ u grafu $G = (V, E)$* ***susjedni*** *ako postoji brid $e = \{u, v\} \in E$.*

Definicija 2.1.3. ***Težinski graf*** *$G = (V, E)$ je graf s težinskom funkcijom $f : E \to \mathbb{R}$ ili $f : E \to \mathbb{R}_0^+$ na skupu bridova E.*

Definicija 2.1.4. *Kažemo da je graf $G' = (V', E')$* ***podgraf*** *grafa $G = (V, E)$ ako je $V' \subseteq V$ i $E' \subseteq E$.*

Definicija 2.1.5. *Ako je $G' = (V', E')$ podgraf grafa $G = (V, E)$ i vrijedi da se skup E' sastoji od svih bridova iz G čija oba kraja leže u V', tada graf G' zovemo* ***inducirani podgraf*** *grafa G.*

Definicija 2.1.6. ***Šetnja*** *od v_0 do v_n u grafu $G = (V, E)$ je niz $(v_0, e_1, v_1, e_2, v_2, \ldots, e_n, v_n)$, gdje je $e_i \in E$ oznaka za brid $\{v_{i-1}, v_i\}$, a $v_i \in V$, za $i = 1, 2, \ldots, n$.*

Definicija 2.1.7. ***Put*** *je šetnja u kojoj su svi vrhovi različiti (osim eventualno prvog i zadnjeg).*

Definicija 2.1.8. *Kažemo da su vrhovi $x, y \in V$ grafa $G = (V, E)$* ***u relaciji****, odnosno $x \equiv y$, ako postoji put u grafu G od x do y. Time je definirana* ***relacija ekvivalencije*** *$\equiv$ na skupu vrhova V grafa G.*

Definicija 2.1.9. ***Komponenta povezanosti*** *grafa $G = (V, E)$ je podgraf induciran klasom ekvivalencije po $\equiv$ iz prethodne definicije.*

Definicija 2.1.10. *Kažemo da je graf $G = (V, E)$* ***povezan*** *ako postoji samo jedna komponenta povezanosti u tom grafu.*

Definicija 2.1.11. *Kažemo da je graf G* ***usmjereni graf*** *ili* ***digraf*** *ako ima* ***usmjerene bridove*** *(bridove s orijentacijom tako da idu od jednog vrha prema drugome). Usmjereni bridovi reprezentiraju se uređenim parovima iz E, umjesto 2-podskupovima iz E.*

Definicija 2.1.12. *Kažemo da je graf G* ***multigraf*** *ako ima* ***višestruke bridove*** *(više bridova između jednog para vrhova). Kod višestrukih bridova E postaje multiskup.*

Napomena 2.1.13. *U ovom radu, neusmjereni graf bez višestrukih bridova i petlji (bridova koji spajaju vrh sa samim sobom) naziva se* ***0-1 graf****.*

2.2 Problem traženja najveće klike

Definicija 2.2.1. *Kažemo da je graf $G = (V, E)$* ***potpun*** *ako svaki par vrhova iz V čini brid.*

Definicija 2.2.2. ***Klika*** *u grafu $G = (V, E)$ je potpun podgraf grafa G koji se sastoji od barem dva vrha.*

Definicija 2.2.3. ***Maksimalna klika*** *u grafu $G = (V, E)$ je klika koja nije sadržana ni u jednoj većoj kliki, tj. dodavanjem nekog vrha, ona prestaje biti klika.*

Definicija 2.2.4. ***Najveća klika*** *u grafu $G = (V, E)$ je klika koja ima najveći broj vrhova.*

Bron-Kerbosch algoritam

Nizozemski programeri Coenraad Bron i Joep Kerbosch osmislili su egzaktni algoritam za računanje svih maksimalnih klika u neusmjerenom grafu. Bron-Kerbosch algoritam objavljen je 1973. godine, a u ovom radu koristi se verzija sa pivotiranjem. Budući da je najveća maksimalna klika ujedno i najveća klika, pomoću ovog algoritma moguće je pronaći najveći potpun podgraf zadanog grafa. Bron-Kerbosch algoritam je rekurzivan i prikazan je sljedećim pseudokodom:

```
bronkerbosch(R, P, X):
    if P = ∅ and X = ∅:
        return R
    odaberi pivotni vrh p ∈ P ∪ X
    for v ∈ P \ N(p): #N(p) su susjedni vrhovi vrha p
        bronkerbosch(R ∪ {v}, P ∩ N(v), X ∩ N(v))
        P := P \ {v}
```

$X := X \cup \{v\}$

Listing 2.1: Bron-Kerbosch pseudokod

Pokazano je u [9] da je vremenska složenost Bron-Kerbosch algoritma za generiranje svih maksimalnih klika u najgorem slučaju $O(3^{n/3})$, gdje je n broj vrhova u zadanom grafu.

2.3 Traženje aproksimativne najveće klike

Budući da je prethodno prikazani egzaktni algoritam za traženje najveće klike iznimno spor za veće n, u ovom radu istražit će se učinkovitost algoritma za traženje aproksimativne najveće klike.

Definicija 2.3.1. *Skup vrhova $S \subseteq V$, u grafu $G = (V, E)$ zovemo **nezavisni skup** ako vrijedi da za svaka dva vrha u S ne postoji brid koji ih spaja.*

Definicija 2.3.2. ***Najveći nezavisni skup** u grafu $G = (V, E)$ je nezavisni skup koji ima najveći broj vrhova.*

Definicija 2.3.3. *Neka je $G = (V, E)$ graf. Skup $(V \times V) \setminus E$ zovemo **komplementom skupa bridova E** i označavamo ga sa $\bar{E}$.*

Lema 2.3.4. *Neka je $G = (V, E)$ graf. Postoji 1-1 korespondencija između k-klika u G i nezavisnih skupova veličine k u grafu $\bar{G} = (V, \bar{E})$, gdje je $k \in \mathbb{N}$.*

Dokaz. Očito. □

Problem traženja najvećeg nezavisnog skupa u grafu je NP-težak, kao i problem traženja najveće klike u grafu. Algoritam za traženje najveće aproksimativne klike korišten u ovom radu upotrebljava ekvivalenciju problema najveće klike i najvećeg nezavisnog skupa procjenjujući prvo najveći nezavisni skup.

2.4 Traženje aproksimativnog najvećeg nezavisnog skupa

Neka je sada $G = (V, E)$ graf u kojem se traži aproksimativni najveći nezavisni skup i n broj vrhova u G. Neka je $N(v)$ podgraf od G induciran svim vrhovima koji su susjedni sa proizvoljnim vrhom $v \in V$ (eng. *neighborhood*). Slično, neka je $\bar{N}(v)$ podgraf od G induciran svim vrhovima koji nisu susjedni sa v. $N(v)$ zovemo **susjedstvo vrha** v, a $\bar{N}(v)$ zovemo **nesusjedstvo vrha** v.

Pretpostavimo sada da želimo smjestiti vrh $v \in V$ u nezavisni skup. Mogli bismo samo pretražiti nesusjedstvo od v ($\bar{N}(v)$) da bismo našli ostale vrhove u tom nezavisnom

skupu. Takvim pristupom nastaje sljedeća heuristika, tzv. pohlepna metoda (eng. *greedy*) prikazana pseudokodom.

1. Izaberi $v \in V(G)$
2. $I(G) \leftarrow \{v\} \cup I(\bar{N}(v))$

Listing 2.2: Pseudokod početne heuristike

Naravno, kod ovakvog pristupa nastaje problem. Susjedstvo pivotnog elementa v uopće nije uzeto u obzir, a ono može sadržavati puno veći nezavisni skup, pa uspješnost uvelike ovisi o izboru pivotnog elementa v. To nas navodi na drugu metodu pronalaska nezavisnog skupa. Kao i prije, uzmimo $v \in V$ i pretražimo nesusjedstvo vrha v. Međutim, ovaj puta pretražimo i susjedstvo od v i od ta dva skupa, uzmimo veći. Naravno, dualna metoda se može primijeniti i za traženje klika u grafu G, a to je objedinjeno sljedećim pseudokodom.

1. Izaberi $v \in V$
2. $I(G) \leftarrow max(\{v\} \cup I(\bar{N}(v)), I(N(v)))$
3. $C(G) \leftarrow max(\{v\} \cup C(\bar{N}(v)), C(N(v)))$

Listing 2.3: Pseudokod poboljšane heuristike

Takvim pristupom dobiven je sljedeći algoritam nazvan po engleskom matematičaru imena Frank P. Ramsey (1903.-1930.).

Ramsey(G)
if $G = \emptyset$ **then return**$(\emptyset, \emptyset)$
Choose some $v \in G$
$(C_1, I_1) \leftarrow$ **Ramsey**$(N(v))$
$(C_2, I_2) \leftarrow$ **Ramsey**$(\bar{N}(v))$
return $(max(C_1 \cup \{v\}, C_2), max(I_1, I_2 \cup \{v\}))$

Listing 2.4: Ramsey pseudokod

Može se pokazati da gornji algoritam ima dobar učinak ako zadani graf nema velikih klika. Međutim, ako postoje dovoljno velike klike u grafu G, ništa se ne može ustvrditi o učinku ovog algoritma. Prema tome, ako bismo se nekako mogli riješiti velikih klika, imali bismo dobar učinak na ostatku grafa. To nas navodi na novu, jednostavnu metodu:

1. Ukloni maksimalni skup disjunktnih k-klika iz G, za neki $k \in \mathbb{N}$.
2. Primijeni Ramsey na preostali graf.

Prvo pitanje koje se javlja kod ovakvog pristupa je hoće li išta ostati u grafu nakon što uklonimo sve vrhove u disjunktnim k-klikama? Za proizvoljni graf stvarno je moguće da ništa ne ostane. Međutim, ako imamo graf koji sadrži dovoljno velik nezavisni skup, može se pokazati da će postojati preostali graf (dapače, bit će prilično velik).

Drugi problem je to što je pronalazak k-klika u grafu poprilično skupo, no ne moramo uklanjati klike na koje ne naiđemo, samo one koje nam se nađu na putu. Dakle, dovoljno je da uklanjamo klike dok prolazimo. Sjetimo se da Ramsey pronalazi aproksimaciju i za kliku i za nezavisni skup. Ako je pronađena klika mala, onda nezavisni skup mora biti velik. Ako je pak pronađena klika velika, onda je možemo ukloniti i ponoviti postupak. Na taj način dobijemo konačan postupak algoritma za traženje najvećeg nezavisnog skupa.

Clique Removal(G)
$i \leftarrow 1$
$(C_i, I_i) \leftarrow$ **Ramsey**(G)
while $G \neq \emptyset$:
    $G \leftarrow G - C_i$
    $i \leftarrow i + 1$
    $(C_i, I_i) \leftarrow$ **Ramsey**(G)
endwhile
return $((max^{i}_{j=1} I_j), \{C_1, C_2, \ldots, C_i\})$

Listing 2.5: Algoritam za pronalazak aproksimativnog najvećeg nezavisnog skupa

Gore navedeni algoritam opetovano zove funkciju **Ramsey** i uklanja nađene klike iz grafa, sve dok se graf ne iscrpi. Zatim, algoritam vrati najveći od svih pronađenih nezavisnih skupova zajedno sa nizom svih pronađenih klika.

Primjena na traženje najveće klike

U kontekstu leme 2.3.4, jednostavno je primijeniti gornji algoritam na problem traženja najveće klike. Dovoljno je primijeniti isti algoritam na graf komplementaran grafu u kojem želimo pronaći najveću kliku. Konačna implementacija cijelog algoritma za traženje najveće aproksimativne klike prikazana je sljedećim kodom (u programskom jeziku Python).

```
import networkx as nx
from networkx.algorithms.approximation import ramsey
def max_clique(G):
    cgraph = nx.complement(G)
    iset, _ = clique_removal(cgraph)
    return iset

def clique_removal(G):
    #Repeatedly remove cliques from the graph.
    graph = G.copy()
    c_i, i_i = ramsey.ramsey_R2(graph)
```

```
    cliques = [c_i]
    isets = [i_i]
    while graph:
        graph.remove_nodes_from(c_i)
        c_i, i_i = ramsey.ramsey_R2(graph)
        if c_i:
            cliques.append(c_i)
        if i_i:
            isets.append(i_i)
    # Determine the largest independent set as measured by cardinality.
    maxiset = max(isets, key=len)
    return maxiset, cliques
```

Listing 2.6: Implementacija algoritma u Pythonu

Vremenska složenost gore prikazanog algoritma je u najgorem slučaju $O(n/(\log(n))^2)$, gdje je n broj vrhova u zadanom grafu. Vidljivo je kako je to značajno bolja vremenska složenost od složenosti Bron-Kerbosch algoritma ($O(3^{n/3})$), no ipak se ovdje radi o aproksimaciji. Detalji o ovoj implementaciji nalaze se u [2], odnosno u dokumentaciji „NetworkX“ paketa za programski jezik Python, a više o učinku i porijeklu ove aproksimacije nalazi se u [1].

Pojmovi iz ovog poglavlja preuzeti su iz izvora [1], [2], [5] i [10].

Poglavlje 3

Bioinformatika

3.1 Biološki pojmovi

Protein ili bjelančevina je izuzetno složena molekula, prisutna u svim živim bićima, a sastoji se od aminokiselina. Proteom je skup svih proteina u nekom organizmu, stanici ili vlaknu. Proteini su sastavni dijelovi svake stanice, što ih čini jednom od osnova života na Zemlji. Aminokiseline unutar proteina povezane su peptidnim vezama i zajedno tvore duge lance, kao kuglice nanizane na žici. Aminokiseline su zapravo organski spojevi sastavljeni od karboksilne skupine, amino skupine i bočnog lanca po kojemu se one međusobno razlikuju. Proteini su izgrađeni od 20 standardnih aminokiselina prikazanih na sljedećoj tablici.

Kratica	Naziv	Kratica	Naziv
A	Alanin	M	Metionin
C	Cistein	N	Asparagin
D	Asparaginska kiselina	P	Prolin
E	Glutaminska kiselina	Q	Glutamin
F	Fenilalanin	R	Arginin
G	Glicin	S	Serin
H	Histidin	T	Treonin
I	Izoleucin	V	Valin
K	Lizin	W	Triptofan
L	Leucin	Y	Tirozin

Tablica 3.1: Standardne aminokiseline

GDSL lipaze

Lipaze su enzimi koji sudjeluju kao katalizatori u hidrolizi lipida (masti). Hidroliza lipida je razgradnja (rastavljanje) molekula lipida u reakciji s vodom. **GDSL lipaze** jedan su od primjera lipaza, a njihova posebnost je u tome što imaju fleksibilno katalitičko mjesto (niz aminokiselina koji kataliziraju reakciju supstrata) koje mijenja svoj strukturni raspored u prisutnosti različitih supstrata. To bi moglo objasniti njihovu katalitičku multifunkcionalnost i to ih čini vrlo pogodnim za istraživanja i primjene.

GDSL lipaze nađene su u biljkama, životinjama, gljivama i bakterijama, a najviše ih ima u kopnenim biljkama. Upravo bi biljke mogle biti dobar izvor obećavajućih enzima. Takvi enzimi bi se mogli koristiti u hidrolizi i sintezi spojeva koji su od velikog interesa u biotehnologiji. Stoga je traženje novih GDSL lipaza u biljkama iznimno važno.

3.2 Iterativno pretraživanje proteoma

Za pronalazak proteina koji pripadaju istoj proteinskoj familiji, često se koristi iterativno pretraživanje proteoma. Cilj iterativnog pretraživanja je da se za određeni upit dobiju svi nizovi aminokiselina koji su dovoljno slični zadanom upitu (s obzirom na određenu funkciju sličnosti). Time se dobije niz proteina koji su iz iste proteinske familije uz određenu uspješnost. U ovom radu za iterativno pretraživanje proteoma koristi se IGLOSS server, opisan u izvoru [6]. Slijede definicije upita i odgovora koje utvrđuju kako se ti termini koriste u ovom radu.

Definicija 3.2.1. ***Upit*** *(ili motiv) definira se kao niz standardnih aminokiselina, odnosno niz slova, najčešće duljine od 5 do 20.*

Definicija 3.2.2. *Skup svih nizova aminokiselina dobivenih iterativnim pretraživanjem proteoma naziva se* ***odgovor****.*

Definicija 3.2.3. *BLOSUM matrica B je 20×20 matrica, $B = (b_{ij}) \in M_{20}(\mathbb{Z})$, koja na (i, j)-tom mjestu sadrži koeficijente sličnosti i-te i j-te aminokiseline. (više o BLOSUM matrici u ....). Ukratko, bazirana je na sljedećoj formuli:*

$$B(i,j) = \left\lfloor \log \frac{\mathbb{P}\left(a_i \leftrightarrow b_j \mid M\right)}{\mathbb{P}\left(a_i, b_j \mid R\right)} \right\rfloor, \quad a_i, b_j \in \mathcal{A}, \tag{3.1}$$

gdje su a_i i b_j aminokiseline pridružene, respektivno, i-tom i j-tom mjestu, a $\mathcal{A}$ je skup svih standardnih aminokiselina. M je model koji pretpostavlja da aminokiseline a_i i b_j imaju zajedničkog pretka, a R je random model koji pretpostavlja nezavisnost aminokiselina, pa vrijedi $\mathbb{P}(a_i, b_j \mid R) = \mathbb{P}(a_i \mid R) \cdot \mathbb{P}(b_j \mid R)$. Distribucija standardnih aminokiselina uz model

R dana je sa:

A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
0.078	0.051	0.043	0.053	0.019	0.043	0.063	0.072	0.023	0.053	0.091	0.059	0.022	0.039	0.052	0.068	0.059	0.014	0.032	0.066

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	−2	−1	−2	−1	−1	−1	0	−2	−1	−2	−1	−1	−3	−1	1	0	−3	−2	0
R	−2	7	−1	−2	−4	1	0	−3	0	−4	−3	3	−2	−3	−3	−1	−1	−3	−1	−3
N	−1	−1	7	2	−2	0	0	0	1	−3	−4	0	−2	−4	−2	1	0	−4	−2	−3
D	−2	−2	2	8	−4	0	2	−1	−1	−4	−4	−1	−4	−5	−1	0	−1	−5	−3	−4
C	−1	−4	−2	−4	13	−3	−3	−3	−3	−2	−2	−3	−2	−2	−4	−1	−1	−5	−3	−1
Q	−1	1	0	0	−3	7	2	−2	1	−3	−2	2	0	−4	−1	0	−1	−1	−1	−3
E	−1	0	0	2	−3	2	6	−3	0	−4	−3	1	−2	−3	−1	−1	−1	−3	−2	−3
G	0	−3	0	−1	−3	−2	−3	8	−2	−4	−4	−2	−3	−4	−2	0	−2	−3	−3	−4
H	−2	0	1	−1	−3	1	0	−2	10	−4	−3	0	−1	−1	−2	−1	−2	−3	2	−4
I	−1	−4	−3	−4	−2	−3	−4	−4	−4	5	2	−3	2	0	−3	−3	−1	−3	−1	4
L	−2	−3	−4	−4	−2	−2	−3	−4	−3	2	5	−3	3	1	−4	−3	−1	−2	−1	1
K	−1	3	0	−1	−3	2	1	−2	0	−3	−3	6	−2	−4	−1	0	−1	−3	−2	−3
M	−1	−2	−2	−4	−2	0	−2	−3	−1	2	3	−2	7	0	−3	−2	−1	−1	0	1
F	−3	−3	−4	−5	−2	−4	−3	−4	−1	0	1	−4	0	8	−4	−3	−2	1	4	−1
P	−1	−3	−2	−1	−4	−1	−1	−2	−2	−3	−4	−1	−3	−4	10	−1	−1	−4	−3	−3
S	1	−1	1	0	−1	0	−1	0	−1	−3	−3	0	−2	−3	−1	5	2	−4	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	2	5	−3	−2	0
W	−3	−3	−4	−5	−5	−1	−3	−3	−3	−3	−2	−3	−1	1	−4	−4	−3	15	2	−3
Y	−2	−1	−2	−3	−3	−1	−2	−3	2	−1	−1	−2	0	4	−3	−2	−2	2	8	−1
V	0	−3	−3	−4	−1	−3	−3	−4	−4	4	1	−3	1	−1	−3	−2	0	−3	−1	5

Slika 3.1: BLOSUM matrica

Definicija 3.2.4. *BLOSUM score s je rezultat koji odgovara sličnosti (ili povezanosti) dvaju nizova aminokiselina. Što je BLOSUM score veći, nizovi aminokiselina su sličniji. BLOSUM score s dvaju nizova standardnih aminokiselina dobiva se zbrajanjem sličnosti između pojedinih aminokiselina u tim nizovima, gdje su te sličnosti prethodno definirane BLOSUM matricom.*

Za ocjenu sličnosti IGLOSS koristi funkciju LLR (log likelihood ratio) koja je ocijenjena pomoću logističke distribucije. Za odgovor se uzimaju u obzir nizovi čija je ocjena veća ili jednaka od skale pretraživanja.

Skala pretraživanja je parametar koji postavlja granicu „dovoljne“ sličnosti s obzirom na funkciju sličnosti. Što je veća skala, sličniji nizovi su odabrani. Dakle, prirodno je očekivati više podataka u odgovoru za manje skale, a manje podataka u odgovoru za veće skale.

3.3 Pridruživanje grafa

Odgovoru dobivenom iterativnim pretraživanjem (uz zadani upit i skalu) pridružuje se **težinski graf** na sljedeći način. Svakom nizu aminokiselina iz odgovora pridružen je jedan vrh grafa. Sličnosti između tih nizova aminokiselina (BLOSUM score) postaju bridovi

grafa. Preciznije, težina brida između dva vrha u grafu je BLOSUM score dvaju nizova aminokiselina kojima su ta dva vrha pridružena.

Zbog načina pridruživanja, dobiveni težinski graf je potpun. Međutim, sličnosti između tih nizova aminokiselina ne moraju biti dovoljno velike da se na temelju njih utvrdi pripadnost određenoj proteinskoj familiji. Da bi se isključili nedovoljno slični nizovi aminokiselina, postavlja se određeni prag. **Prag** je najmanja težina koja se smatra dovoljno velikom da bi sličnosti veće ili jednake pragu mogle predstavljati biološku značajnost. Optimalna vrijednost praga može se izračunati i iznosi $2.5 \cdot l$, gdje je l duljina zadanog upita.

Prema tome, u dobivenom težinskom grafu isključuju se svi bridovi čije su težine manje od praga ($2.5l$) i na taj način dobije se $0-1$ graf, koji vrlo vjerojatno više nije potpun.

Primjena algoritama za traženje najveće klike

Cilj svega prethodnog je da se traženjem najvećeg potpunog podgrafa u dobivenom $0-1$ grafu dobije skup nizova aminokiselina, odnosno skup proteina koji sadrži što veći udio onih proteina koji pripadaju istoj proteinskoj familiji. Dakle, ovdje se koriste algoritmi za traženje najveće klike da bi se smanjio broj lažno pozitivnih (FP) primjera u odgovoru.

Naravno, postoje rizici kod korištenja ovakvog pristupa. Jedan od rizika je mogući gubitak prevelikog broja stvarno pozitivnih (TP) primjera iz odgovora, čime se onda značajno kvari uspješnost modela. Testiranje ovih tvrdnji dano je sljedećim poglavljem koje sadrži rezultate modela na stvarnim podacima.

Pojmovi iz ovog poglavlja preuzeti su iz izvora [6], [3] i [8].

Poglavlje 4

Primjeri

4.1 Proteomi

U ovom radu, uspješnost tri različita modela ispitana je na četiri različita proteoma:

- Talijin uročnjak (lat. *Arabidopsis thaliana*)
- Krumpir (lat. *Solanum tuberosum*)
- Rajčica (lat. *Solanum lycopersicum*)
- Šećerna repa (lat. *Beta vulgaris*)

Slika 4.1: *Arabidopsis thaliana*

Kod svih proteoma korišten je upit FVFGDSLSDA za iterativno pretraživanje proteoma. Taj upit sadrži niz aminokiselina GDSL, koji je tipičan za GDSL lipaze. Smisao

korištenja takvog upita je upravo pronalazak proteina koji su iz familije GDSL lipaza u danom proteomu. Mjere uspješnosti izračunate su usporedbom rezultata određenog modela sa listom CP (eng. *Condition Positive*), odnosno listom svih proteina za koje je već utvrđeno da pripadaju familiji GDSL lipaza. Svi ostali proteini koji nisu na listi CP, smatraju se CN (eng. *Condition Negative*). Svi proteini koje neki od modela vrati kao rezultat označeni su sa P (**pozitivni**, eng. *Positives*), dok su svi ostali proteini iz danog proteoma koji nisu u rezultatu označeni sa N (**negativni**, eng. *Negatives*). Za ilustraciju, slijede odnosi između definiranih pojmova i pojmova iz tablice uspješnosti:

$$TP = P \cap CP, \quad FP = P \cap CN,$$
$$TN = N \cap CN, \quad FN = N \cap CP.$$

Ispitana je uspješnost tri modela koji su označeni na sljedeći način:

- IGLOSS
- IGLOSS+Exact Clique
- IGLOSS+Appr Clique

IGLOSS je iterativno pretraživanje zadanog proteoma, kako je ranije opisano u cjelini 3.2. IGLOSS+Exact Clique je izvršavanje Bron-Kerbosch (egzaktnog) algoritma za traženje najveće klike u 0-1 grafu koji je pridružen odgovoru IGLOSS-a kako je opisano u 3.3. IGLOSS+Appr Clique je izvršavanje algoritma za traženje aproksimativne najveće klike u 0-1 grafu koji je pridružen odgovoru IGLOSS-a kako je opisano u 3.3.

Kod pridruživanja 0-1 grafa za sve proteome korišten je prag 25 ($2.5 \cdot 10$), kako je navedeno u 3.3, jer je duljina upita 10. Korištene su različite vrijednosti skale pretraživanja u IGLOSS-u, da bi se učinak svih modela vidio na manjem i na većem skupu podataka. Dakle, vrijednosti skale pretraživanja su: 5, 4, 3.5 i 3 i to na svakom od četiri proteoma. U konačnici je dobiveno 16($= 4$ skale $\cdot$ 4 proteoma) rezultata za modele IGLOSS i IGLOSS+Appr Clique. Dobiveno je 8($= 2 \cdot 4$) rezultata za model IGLOSS+Exact Clique. Za taj model korištene su samo skale 5 i 4 zbog prevelike vremenske složenosti za veći broj podataka kakav se dobije za niže skale pretraživanja.

4.2 Rezultati i usporedba

U tablicama su prikazane mjere uspješnosti sva tri navedena modela. Sa n je označen broj vrhova u 0-1 grafu, odnosno broj nizova aminokiselina koje IGLOSS vrati u odgovoru.

Arabidopsis Thaliana

Duljina liste CP za talijin uročnjak je 103, a rezultati su prikazani ispod, gdje Vrijeme označava vrijeme u kojem se određeni algoritam za klike izvršio (nakon iterativnog pretraživanja), što je bitno zbog usporedbe dva algoritma za klike. Prema tome, informacija o vremenu izvršavanja IGLOSS-a nije dostupna.

1. Skala pretraživanja je 5 ($n = 421$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.67	0.84	0.913	5s	69
IGLOSS+Exact Clique	0.86	0.87	0.932	148s	89
IGLOSS	0.90	0.26	0.417	.	93

2. Skala pretraživanja je 4 ($n = 882$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.796	0.837	0.911	36s	82
IGLOSS+Exact Clique	0.874	0.849	0.918	3337s	90
IGLOSS	0.951	0.139	0.245	.	98

Već iz prve dvije tablice vidljivo je da egzaktni algoritam postiže bolje mjere uspješnosti od aproksimativnog. Međutim, vrijeme izvršavanja egzaktnog algoritma za $n = 882$ je skoro sat vremena, dok se aproksimativni algoritam izvrši već za 36 sekundi.

3. Skala pretraživanja je 3.5 ($n = 1339$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.699	0.818	0.899	128s	72
IGLOSS	0.951	0.092	0.168	.	98

4. Skala pretraživanja je 3 ($n = 1993$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.757	0.838	0.912	480s	78
IGLOSS	0.932	0.061	0.115	.	96

Gornje četiri tablice pokazuju da IGLOSS nalazi više TP od ostalih modela. To se moglo i očekivati, s obzirom na to da ostali modeli rade na principu isključivanja podataka koji su dobiveni IGLOSS-om, pa očito isključe i neke TP. Međutim, vidi se iz osjetljivosti (TPR), preciznosti (PPV) i visokog F1-scorea da su ostali modeli dosta uspješni

u isključivanju FP dobivenih IGLOSS-om, bez da isključe previše TP. Početni F1-score IGLOSS-a je 0.417 i on ovdje pada sve do 0.115 kako se povećava broj podataka (kako pada skala pretraživanja).

U sljedećoj tablici je prikaz postotaka zadržanih TP u odnosu na IGLOSS za aproksimativni algoritam.

IGLOSS+Appr Clique Udio zadržanih TP u odnosu na IGLOSS			
Skala 5	Skala 4	Skala 3.5	Skala 3
74.19%	83.67%	73.47%	81.25%

Aproksimativni algoritam je ovdje u najgorem slučaju zadržao 73.47% stvarno pozitivnih dobivenih IGLOSS-om.

Krumpir

Duljina liste CP za krumpir je 123, a rezultati su prikazani ispod.

1. Skala pretraživanja je 5 (n = 389).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.593	0.849	0.918	4s	73
IGLOSS+Exact Clique	0.707	0.897	0.945	99s	87
IGLOSS	0.772	0.245	0.393	.	95

2. Skala pretraživanja je 4 (n = 893).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.602	0.949	0.973	40s	74
IGLOSS+Exact Clique	0.707	0.907	0.950	2122s	87
IGLOSS	0.772	0.109	0.197	.	95

U prve dvije tablice vide se odlični rezultati egzaktnog algoritma u usporedbi s IGLOSS-om. Aproksimativni algoritam ovdje ima nešto lošiju osjetljivost od egzaktnog, ali preciznost mu je dobra, pa mu je visok i F1-score. Također, opet se za veće podatke vidi velika razlika u brzini.

3. Skala pretraživanja je 3.5 (n = 1334).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.659	0.890	0.941	143s	81
IGLOSS	0.780	0.075	0.139	.	96

4. Skala pretraživanja je 3 (n = 1983).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.634	0.929	0.962	510s	78
IGLOSS	0.797	0.052	0.099	.	98

Aproksimativni algoritam zadržao je visoku preciznost i F1-score i na većem broju podataka.

IGLOSS+Appr Clique Udio zadržanih TP u odnosu na IGLOSS			
Skala 5	Skala 4	Skala 3.5	Skala 3
76.84%	77.89%	84.38%	79.59%

Aproksimativni algoritam je ovdje u najgorem slučaju zadržao 76.84% stvarno pozitivnih dobivenih IGLOSS-om.

Rajčica

Duljina liste CP za rajčicu je 108, a rezultati su prikazani ispod.

1. Skala pretraživanja je 5 (n = 387).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.648	0.9459	0.97170	4s	70
IGLOSS+Exact Clique	0.806	0.9456	0.97178	88s	87
IGLOSS	0.877	0.249	0.399	.	95

2. Skala pretraživanja je 4 (n = 882).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.704	0.938	0.968	39s	76
IGLOSS+Exact Clique	0.806	0.9456	0.97178	1405s	87
IGLOSS	0.870	0.110	0.198	.	94

U gornje dvije tablice zanimljivo je primijetiti da je egzaktni algoritam oba puta našao istu kliku (u grafu sa 387 vrhova i u grafu sa 882 vrha). Prema tome, egzaktni algoritam pokazuje iste mjere uspješnosti u oba slučaja. Također, ovdje su razlike u preciznosti i F1-scoreu između aproksimativnog i egzaktnog algoritma toliko male da su vidljive tek u četvrtoj, odnosno petoj decimali.

3. Skala pretraživanja je 3.5 ($n = 1292$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.676	0.948	0.973	127s	73
IGLOSS	0.889	0.078	0.145	.	96

4. Skala pretraživanja je 3 ($n = 2014$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.657	0.922	0.959	523s	71
IGLOSS	0.898	0.051	0.097	.	97

Aproksimativni algoritam zadržao je visok F1-score i za veći broj podataka. Za $n = 2014$ trebalo mu je oko 9 minuta za izvršavanje.

IGLOSS+Appr Clique Udio zadržanih TP u odnosu na IGLOSS			
Skala 5	Skala 4	Skala 3.5	Skala 3
73.68%	80.85%	76.04%	73.20%

Aproksimativni algoritam je ovdje u najgorem slučaju zadržao 73.20% stvarno pozitivnih dobivenih IGLOSS-om.

Šećerna repa

Duljina liste CP za šećernu repu je 82, a rezultati su prikazani ispod.

1. Skala pretraživanja je 5 ($n = 306$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.463	0.864	0.926	2s	38
IGLOSS+Exact Clique	0.707	0.906	0.950	11s	58
IGLOSS	0.756	0.219	0.359	.	62

2. Skala pretraživanja je 4 ($n = 692$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.707	0.906	0.950	34s	58
IGLOSS+Exact Clique	0.707	0.906	0.950	114s	58
IGLOSS	0.756	0.098	0.179	.	62

Iz gornjih tablica vidi se da je egzaktni algoritam opet u oba slučaja pronašao istu najveću kliku. Dodatno, vidi se da je za $n = 692$ i aproksimativni algoritam pronašao tu istu najveću kliku. Također, aproksimativni algoritam pronalazi najviše TP za $n = 692$.

3. Skala pretraživanja je 3.5 ($n = 1074$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.524	0.915	0.955	83s	43
IGLOSS	0.780	0.067	0.125	.	64

4. Skala pretraživanja je 3 ($n = 1741$).

Model	TPR	PPV	F1-score	Vrijeme	Broj TP
IGLOSS+Appr Clique	0.561	0.868	0.929	392s	46
IGLOSS	0.780	0.042	0.080	.	64

Iako svugdje ima visok F1-score, aproksimativni algoritam ima najvišu osjetljivost upravo u slučaju u kojem je pronašao stvarnu najveću kliku.

IGLOSS+Appr Clique Udio zadržanih TP u odnosu na IGLOSS			
Skala 5	Skala 4	Skala 3.5	Skala 3
65.52%	93.55%	67.19%	71.19%

Aproksimativni algoritam ovdje je u najgorem slučaju zadržao 65.52% stvarno pozitivnih dobivenih IGLOSS-om.

Pojmovi iz ovog poglavlja preuzeti su iz [6].

Poglavlje 5

Zaključak

Nakon usporedbe uspješnosti tri različita modela na proteomima talijinog uročnjaka, krumpira, rajčice i šećerne repe za upit FVFGDSLSDA i za različite vrijednosti skale pretraživanja, može se uočiti sljedeće.

Modeli koji su koristili klike (IGLOSS+Appr Clique i IGLOSS+Exact Clique) u svim slučajevima imaju znatno bolju uspješnost (u smislu mjere F1-score) od osnovnog modela iterativnog pretraživanja (IGLOSS). Međutim, ta dva modela u svim slučajevima imaju i manji broj stvarno pozitivnih od modela iterativnog pretraživanja. Konkretno, aproksimativni algoritam (IGLOSS+Appr Clique) je u najgorem slučaju zadržao 65.52% stvarno pozitivnih dobivenih IGLOSS-om, a u najboljem slučaju zadržao je 93.55% stvarno pozitivnih.

Također, egzaktni algoritam (IGLOSS+Exact Clique) samo u jednom slučaju nije uspješniji od aproksimativnog algoritma, a u svim slučajevima su razlike u F1-score-u između ta dva modela iznimno male. Usto, egzaktni algoritam zadrži veći udio stvarno pozitivnih dobivenih IGLOSS-om od aproksimativnog algoritma, osim u slučajevima u kojima aproksimativni algoritam nađe stvarnu najveću kliku. Tada je udio zadržanih stvarno pozitivnih jednak za oba modela.

Ako se F1-score postavi kao jedina mjera uspješnosti modela, ukupni rezultati u ovom radu pokazuju da je isplativije koristiti aproksimativni algoritam na većim skupovima podataka, a egzaktni algoritam na manjim skupovima podataka. Razlog je to što postižu vrlo sličan F1-score, a aproksimativni model je puno brži na većim skupovima podataka. Međutim, ukoliko se uz F1-score razmatra i osjetljivost ili broj stvarno pozitivnih, mora se uzeti u obzir da aproksimativni algoritam postiže manju osjetljivost od ostala dva modela u gotovo svim slučajevima, pa prethodna preporuka više ne vrijedi.

Bibliografija

[1] R. Boppana i M. M. Halldórsson, *Approximating maximum independent sets by excluding subgraphs*, BIT Numerical Mathematics **32** (1992), ISSN 1572-9125, https://doi.org/10.1007/BF01994876.

[2] A. A. Hagberg, D. A. Schult i P. J. Swart, *Exploring Network Structure, Dynamics, and Function using NetworkX*, Proceedings of the 7th Python in Science Conference (Pasadena, CA USA) (Gaël Varoquaux, Travis Vaught i Jarrod Millman, ur.), 2008, str. 11 – 15.

[3] S. Henikoff i J. G. Henikoff, *Amino acid substitution matrices from protein blocks*, Proceedings of the National Academy of Sciences **89** (1992), br. 22, 10915–10919, ISSN 0027-8424, https://www.pnas.org/content/89/22/10915.

[4] M. Kobovac, *Neki aspekti iterativnog pretraživanja proteoma*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet (Matematički odsjek), 2017.

[5] K. Martinić, *Maksimalne klike u analizi sličnosti proteinskih motiva*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet (Matematički odsjek), 2018.

[6] B. Rabar, M. Zagorščak, S. Ristov, M. Rosenzweig i P. Goldstein, *IGLOSS: iterative gapless local similarity search*, Bioinformatics **35** (2019), br. 18, 3491–3492, ISSN 1367-4803, https://doi.org/10.1093/bioinformatics/btz086.

[7] N. Sarapa, *Teorija Vjerojatnosti*, Školska knjiga, 2002.

[8] D. Strmečki, *Analiza točnosti pretraživanja*, Diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet (Matematički odsjek), 2020.

[9] E. Tomita, A. Tanaka i H. Takahashi, *The worst-case time complexity for generating all maximal cliques and computational experiments*, Theoretical Computer Science **363** (2006), br. 1, 28–42, ISSN 0304-3975, https://www.sciencedirect.com/science/article/pii/S0304397506003586, Computing and Combinatorics.

[10] D. Veljan, *Kombinatorna i diskretna matematika*, Algoritam, 2001.

Sažetak

U ovom radu promatrao se problem traženja proteina iz iste proteinske familije i uspješnost primjene algoritama iz teorije grafova na taj problem. Konkretno, analizirala se isplativost korištenja algoritma za traženje najveće aproksimativne klike koji djeluje kao nadopuna iterativnom pretraživanju u klasifikaciji proteina. Rezultati korištenja navedenog aproksimativnog pristupa uspoređeni su sa rezultatima algoritma za traženje egzaktne najveće klike i posebno, sa rezultatima početnog iterativnog pretraživanja proteoma. Jedna od glavnih mjera koja je korištena za usporedbu uspješnosti modela je F1-score, a negdje je u obzir uzeta i vremenska složenost. Postupak je proveden na proteomima talijinog uročnjaka, krumpira, šećerne repe i rajčice. Promatrao se učinak algoritama na manjim i na većim skupovima podataka. Dobiveni rezultati pokazuju da korištenje aproksimativnog algoritma bitno poboljšava uspješnost modela (F1-score), iako nalazi manji broj stvarno pozitivnih (TP) u odnosu na početno iterativno pretraživanje. Također, egzaktni algoritam očekivano postiže nešto bolje rezultate od aproksimativnog, ali je na većem broju podataka puno sporiji u izvršavanju.

Summary

This paper covers the problem of identifying proteins belonging to the same protein family and explores the use of applying known graph theory algorithms to this problem. Specifically, it examines the benefits of maximum clique approximation, which serves as a supplement to iterative searching in protein classification. Results of using the aforementioned approximation approach were compared with the results of using the exact maximum clique algorithm, as well as with the results of the initial proteome iterative search. F1-score was the primary metric used for comparing model performance, while in some cases time complexity was also considered. The procedure was performed on several proteomes, namely thale cress, potato, tomato and sugar beet. Algorithmic efficiency was measured on both small and large data sets. Results obtained demonstrate that the use of the approximation algorithm greatly improves model performance (F1-score), although it does retrieve a smaller number of true positives (TP) compared with the initial iterative search. Furthermore, the exact algorithm achieves somewhat better results than the approximation algorithm, but it is considerably slower on large data sets.

Životopis

Rođen sam 24. ožujka 1996. godine u Zagrebu. Školovanje sam započeo u Osnovnoj školi bana Josipa Jelačića u Zagrebu i nastavio u Gimnaziji Lucijana Vranjanina, također u Zagrebu. Nakon završetka srednjoškolskog obrazovanja 2014. godine, upisao sam preddiplomski studij Matematika na Prirodoslovno-matematičkom fakultetu u Zagrebu, kojeg sam završio 2017. godine. Iste godine upisao sam diplomski studij Matematička statistika, također na PMF-u u Zagrebu.

Glazbeno obrazovanje stekao sam u Osnovnoj glazbenoj školi Ivana Zajca i u Glazbenoj školi Blagoja Berse. U slobodno vrijeme volim trenirati, igrati šah i družiti se s prijateljima uz gitaru i pjesmu.