SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3257

Evaluacija aplikacija za pretraživanje baze proteinskih sljedova

Robert Vaser

Zagreb, lipanj 2013.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

Veliko hvala Mili Šikiću i Matiji Korparu na znanju, trudu i strpljenju!

SADRŽAJ

1. Uvod

Velike količine informacija, koje su sadržane u genima živih organizama, zbližile su biologiju i računarstvo, učinivši izazove u biologiji novim područjem interesa u računarstvu. Takva bliska povezanost pridonijela je postepenom razvoju interdisciplinarnog područja koje obuhvaća biologiju, računarsku znanost, matematiku i statistiku te se jednim imenom naziva bioinformatika [1]. Danas je bioinformatika važan dio u mnogim područjima biologije kao što su molekularna biologija, genetika i genomika, a prikupljanje i obrada podataka iz tih područja iziskuje mnogo računalnih resursa. Glavna zadaća bioinformatike jest izrada algoritama koji ekonomično spremaju prikupljene podatke te ih obrađuju u što kraćem vremenu.

Jedna od zanimljivijih grana biologije koja je isprepletena s bioinformatikom jest genetika, znanost o genima, nasljedstvu te raznolikosti živih organizama. Važni, a možda i najstariji algoritmi genetike su algoritmi za poravnanje, tj. izračun sličnosti dvaju ili više sljedova. Oni opisuju načine grupiranja sljedova DNA, RNA i proteina kako bi se odredila područja sličnosti koja su posljedica funkcionalnih, strukturnih ili evolucijskih odnosa. Nastali su kao posljedica prevelike količine vremena potrebne za pronalazak svih mogućih poravnanja dviju sekvenci te izbor najboljih, tj. optimalnih. Generalni algoritmi poravnanja sljedova temeljeni su na dinamičkom programiranju, tj. na metodi programiranja u kojoj se rješavanje kompleksnih problema sastoji od rastavljanja na manje, jednostavnije probleme što garantira matematički optimalno rješenje [2]. Takav postupak troši mnogo vremena i resursa te zbog toga većina najpoznatijih aplikacija za pretraživanje baza DNA, RNA i proteinskih slijedova koristi heurističke metode koje su brže, ali čiji rezultati nisu optimalni.

Pronalazak sličnih sljedova s poznatim svojstvima, osim što služi za određivanje evolucijskog srodstva, pomaže u procjeni svojstava mutiranih slijedova. Vrlo zanimljiva činjenica je da samo jedna izmjena aminokiseline u proteinu može promijeniti funkciju cijelog proteina. Najkorištenija aplikacija za procjenu utjecaja supstitucija aminokiselina na funkciju proteina je SIFT (engl. *Sorting Intolerant From Tolerant*) [3] koja koristi aplikaciju BLAST, točnije PSI-BLAST algoritam za poravnanje pro-

teinskih slijedova. SIFT je veliku popularnost stekao zbog baze neutralnih i negativnih supstitucija za svaki protein u ljudskom organizmu. Kako potražnja baza za proteine drugih vrsta organizama raste, a najveća količina vremena se troši da PSI-BLAST pronađe poravnanja, javlja se potreba za bržom i efikasnijom aplikacijom koja će zamijeniti PSI-BLAST. Stoga je glavna tema ovog rada evaluacija aplikacija za pretraživanje baza proteinskih slijedova kako bi se ubrzao proces poravnanja slijedova uz ograničenje da se točnost aplikacije SIFT, koju ima kad je kombinirana s PSI-BLAST-om, ne umanji.

2. Aplikacija SIFT

U poglavlju se daje kratki opis rada aplikacije SIFT, od kojih koraka se sastoji te na koji način se predočava izlaz koji stvara. S obzirom da je aplikacija za poravnanje proteinskih sljedova prvi korak SIFT-a, prvo će se pojasniti što su to poravnanja, koje vrste postoje te koji su generalni algoritmi za pojedinu vrstu.

2.1. Poravnanje proteinskih sljedova

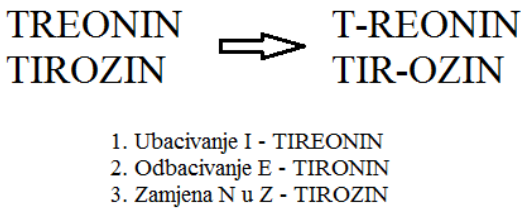
Proteini ili bjelančevine su makro molekule građene od dvadesetak različitih aminokiselina povezanih peptidnim vezama u duže ili kraće lance. Redoslijed i broj aminokiselina u lancu te prostorni raspored lanaca uvjetuje funkciju proteina. Stoga proteini koji obavljaju jednaku ili sličnu funkciju u različitim vrstama živih organizama imaju međusobno slične strukture.

Kada se sekvenciranjem proteina[1] neke vrste organizama pronađu nepoznati sljedovi aminokiselina, prvi korak je pronaći sličnosti s ostalim proteinima drugih vrsta. Ako su poznate funkcije strukturno sličnih proteina, velika je vjerojatnost da će se nepoznati protein podudarati u funkciji. Najjednostavniji način ocjene sličnosti bio bi pronalazak minimalnog broja promjena potrebnih da se iz jednog proteina dobije drugi, ne uzimajući u obzir vrstu promjene (supstitucija, ubacivanje ili odbacivanje dijelova proteina). Kako je danas većina poznatih proteina spremljena u velikim bazama podataka, najčešće u FASTA formatu[2], lako je povući analogiju s prirodnim jezikom. Na slici 2.1 prikazan je primjer poravnanja riječi Treonin i Tirozin, koje rezultira s 5 od 7 identitčnih slova.

Poravnanje se može definirati kao način grupiranja slova kako bi se identifici-

[1] Proces u kojem se precizno određuje položaj svih aminokiselina u proteinu. Otkrivanje funkcija i struktura proteina važan je postupak za razumijevanje staničnih procesa.

[2] Tekstualni format za predstavljanje sljedova nukleotida ili aminokiselina, gdje je svaki nukleotid ili aminokiselina predočena jednim slovom engleske abecede (npr. nukleotid Timidin se označava sa slovom T).



Slika 2.1: Primjer poravnanja riječi Treonin i Tirozin. Također su prikazane potrebne izmjene (ubacivanje, izbacivanje i supstitucija) kako bi iz riječi Treonin dobili Tirozin.

rala mjesta preklapanja u zadanim riječima. Isto vrijedi za grupiranje aminokiselina gdje područja preklapanja ukazuju na moguće podudaranje funkcija proteina. Navedena metoda nije vremenski učinkovita za pronalazak optimalnog poravnanja proteina jer su proteini nekoliko redova veličina dulji od prosječne rečenice prirodnog jezika. Broj mogućih poravnanja dviju sekvenci duljina N moguće je pronaći pomoću furmule (2.1).

$$2^{2*N}/\sqrt{2*\pi*N} \tag{2.1}$$

Za duljine od 150 aminokiselina, broj različitih poravnanja je 10^{90} [2]. Stoga se koriste algoritmi temeljeni na dinamičkom programiranju, metodi koja rješavanje kompleksnih problema rastavlja na manje, jednostavnije probleme što garantira matematički optimalno rješenje.

Algoritmi poravnanja optimiraju ocjenjivanje sličnosti sljedova na način da uparuju aminokiseline te ubacuju praznine di su potrebne. Kako redoslijed originalnih sljedova mora biti očuvan, dozovljene su samo operacije pomaka cijele sekvence te stvaranje procjepa. Dijele se na:

- *algoritme globalnog poravnanja* - optimizacija poravnanja duž cijele duljine proteina. Generalni algoritam je Needleman-Wunsch [4].
- *algoritme lokalnog poravnanja* - pretraživanje proteina, koji mogu biti različitih duljina, za područjima visoke podudarnosti. Generalni algoritam je Smith-Waterman [5].

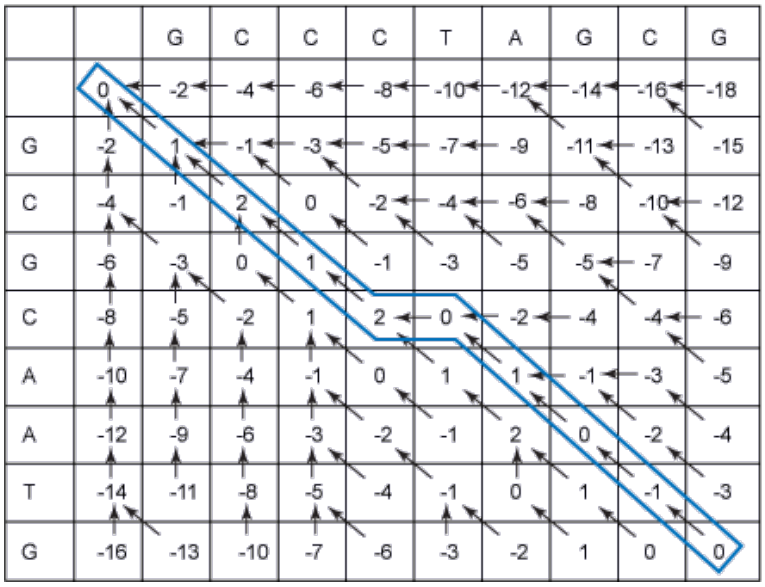
U nastavku su opisani osnovni postupci i ideje algoritama Needleman-Wunsch, Smith-Waterman (oba s linearnom kaznom za procijep) te heurističnog algoritma kojeg koristi alat BLAST.

2.1.1. Needleman-Wunsch algoritam

Glavna ideja algoritma jest da je poravnanje dviju sekvenci duljina n i m moguće reducirati na poravnanje dviju sekvenca duljina n-1 i m-1, što je posljedica pogodka ili supstitucije, ili n-1 i m ili n i m-1, što predstavlja ubacivanje praznine u drugu ili prvu sekvencu. Tako dobivamo rekurzivnu formulu (2.2) koja generira optimalno rješenje. Parametar d predstavlja kaznu za stvaranje procijepa, a S(x, y) je funkcija koja vraća vrijednost iz supstitucijske matrice[3] za aminokiseline x i y. Početni uvjeti su $H(0,0) = 0$, $H(0,j) = -j * d$ te $H(i,0) = -i * d$.

$$H(i,j) = max \begin{cases} H(i-1,j-1) + S(x_i, y_i) \\ H(i-1,j) - d \\ H(i,j-1) - d \end{cases} \tag{2.2}$$

Vrijednosti ove rekurzivne funkcije upisuju se u matricu dimenzija $n * m$ od gornjeg lijevog do donje desnog kuta, na čijim osima su postavljene sekvence koje se uspoređuju. Kako bi se pronašlo optimalno poravnanje potrebno je rekonstruirati put krenuvši iz donjeg desnog kuta matrice prema gornjem lijevom. Odabir smjera kretanja temelji se na maksimalnoj vrijednosti lijeve, gornje te dijagonalne ćelije kao što je prikazano na slici 2.2. Vremenska složenost ovog algoritma je kvadratna $O(nm)$.



Slika 2.2: Rekonstrukcija optimalnog poravnanja pomoću Needleman-Wunsch algoritma.

[3]Supstitucijske matrice sadrže ocjenu supstitucije svakog para amino kiselina. Najpoznatije matrice su PAM i BLOSUM, a danas najkorištenija je BLOSUM62 [6].

2.1.2. Smith-Waterman algoritam

Globalno poravnanje ne uzima u obzir dijelove sljedova s jako visokom podudarnošću, koje je važno kod homolognih proteina[4]. Trenutno najviše odnosa proteina pronađeno je pomoću lokalnog poravnanja. Ovaj algoritam je modifikacija Needleman-Wunsch algoritma. Izostavlja se negativno bodovanje koje je onemogućavalo lokalna poravnanja. Formula 2.2 prelazi u formulu 2.3 s početnim uvjetima $H(0,0) = 0$ i $H(i,0) = H(0,j) = 0$.

$$H(i,j) = max \begin{cases} 0 \\ H(i-1,j-1) + S(x_i, y_i) \\ H(i-1,j) - d \\ H(i,j-1) - d \end{cases} \tag{2.3}$$

Rekonstrukcija počinje od polja s najvećom ocjenom te je izbor smjera kretanja jednak kao kod Needleman Wunsch algoritma. Postupak završava kad rekonstrukcija naiđe na polje s vrijednošću 0 dobivši najbolje ocijenjeno lokalno poravnanje. Ovaj algoritam također ima kvadratnu vremensku složenost $O(nm)$.

		G	C	C	C	T	A	G	C	G
	0	0	0	0	0	0	0	0	0	0
G	0	1	0	0	0	0	0	1	0	1
C	0	0	2	1	1	0	0	0	2	0
G	0	1	0	1	0	0	0	1	0	3
C	0	0	2	1	2	0	0	0	2	1
A	0	0	0	1	0	1	1	0	0	1
A	0	0	0	0	0	0	2	0	0	0
T	0	0	0	0	0	1	0	1	0	0
G	0	1	0	0	0	0	0	1	0	1

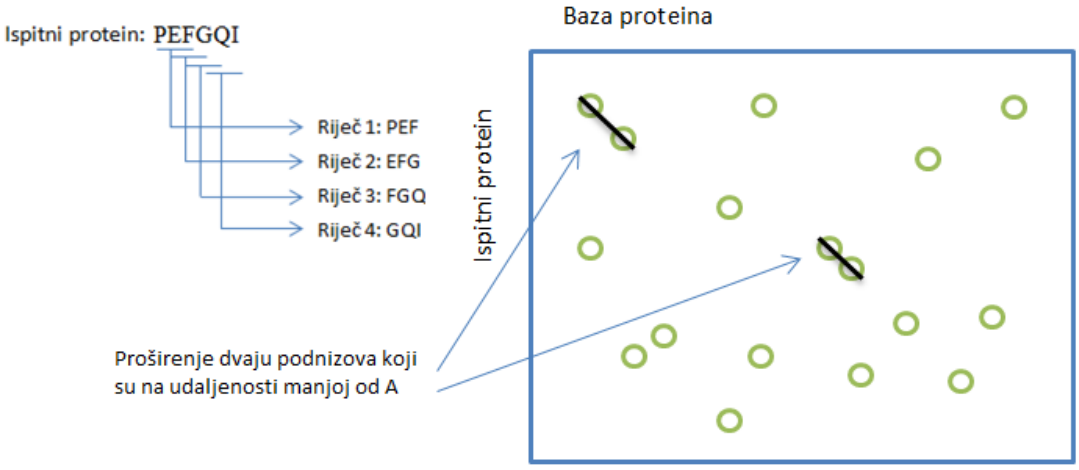
Slika 2.3: Rekonstrukcija lokalnog poravnanja pomoću Smith-Waterman algoritma.

[4]Proteini koji su divergirali od zajedničkog "pretka". Pojam homologija definirao je Richar Owen 1843. godine kao isti organ različitih organizama u raznovrsnim oblicima i funkcijama.

2.1.3. BLAST

BLAST (engl. *Basic Local Alignment Search Tool*) je najkorištenija i najcitiranija aplikacija za pronalaženje poravnanja. Algoritam kojeg koristi je heuristični algoritam za pronalaženje lokalnog poravnanja koji aproksimira Smith-Watermanovo rješenje. Zbog heurističnosti rezultati nisu potpuno točni, ali prednost algoritma je veliko ubrzanje u odnosu na Smith-Watermana (do 50 puta). Temelji se na pretpostavci da su podnizovi velikih sličnosti statistički često sadržani u značajnim poravnanjima [7]. Ideja algoritma može se prikazati kroz sljedeće korake:

- *sastavljanje liste podnizova* - protein za kojeg tražimo poravnanja rastavlja se na nizove od n znakova (engl. *seed*) kako je prikazano lijevo na slici 2.4.
- *pretraživanje podnizova u bazi* - traže se sekvence koje sadrže podnizove duljina n te čija ocjena sličnosti s nekim od podnizova zadanog proteina nije ispod granice T (engl. *threshold*). Koriste se supstitucijske matrice BLOSUM.
- *proširivanje pronađenih podnizova* - nakon što se napravi stablo pogodaka, slijedi poravnanje između dva ili više pogotka koji su na udaljenosti manjoj od A. Rezultat su lokalna poravnanja od kojih se na kraju odabere ono koje ima najvišu ocjenu. Postupak proširivaranja vidljiv je desno na slici 2.4. Ovaj korak mnogo je složeniji nego je ovdje opisano te uzima preko 90% vremena aplikacije.



Slika 2.4: Prikaz sastavljanja liste podnizova duljine 3 (lijevo) te proširivanja pogodaka u bazi proteina (desno).

2.2. SIFT

Aplikacija SIFT (engl. *Sorting Intolerant From Tolerant*) pomoću homolognih proteina predviđa kakvo djelovanje supstitucije aminokiselina imaju na funkcije proteina, koje potencijalno mogu promijeniti čak i fenotip[5]. Rad aplikacije baziran je na pretpostavci da redoslijed aminokiselina važan za funkciju treba biti očuvan kod homolognih proteina [3]. Tako će susptitucije na dobro očuvanim područjima homologa biti proglašene štetnim, dok će se neutralne supstitucije pronalaziti na područjima manje podudarnosti.

Za zadani protein, supstituciju na nekoj poziciji te bazu proteina SIFT će u nekoliko koraka predvidjeti kakav utjecaj dotična izmjena ima na funkciju proteina:

- *pronaći će poravnanja sličnih proteinima* - pri tome mu pomaže PSI-BLAST algoritam i to u 2 iteracije (umjesto nekad korištenih 4 iteracija). Prvi korak jednak je BLAST algoritmu opisanom u prethodnom potpoglavlju. Drugi korak je opet BLAST algoritam, ali sa supstitucijskom matricom specifičnih pozicija dobivene iz poravnanja prvog koraka[6]. Kako je PSI-BLAST korak vremenski jako zahtjevan, često se zamjenjuje s BLAST-om. Postoji i opcija da se na ulaz u SIFT stavi sekvenca te njoj slične već poravnate sekvence što višestruko ubrzava aplikaciju.
- *izračunati vjerojatnosti supstitucija* - na svakoj poziciji poravnanja, svaka aminokiselina i pojavljuje se s frekvencijom f_i. Iz frekvencija f_i pomoću Dirichletovog procesa određuju se vjerojatnosti pojavljivanja p_i. SIFT koristi vjerojatnosti od 0 do 1 koje nisu potpuno matematički ispravne jer ne vrijedi formula (2.4).

$$\sum_{i=1}^{20} p_{aminokiselina_i} = 1. \tag{2.4}$$

  Vjerojatnosti od 0 do 0.05 znače loše djelovanje dok veće od 0.05 predstavljaju neutralno djelovanje supstitucija. Treba naglasiti da vrijednosti između 0.05 do 0.1 nisu jako pouzdane, pa se preporuča uzimanje granične vjerojatnosti od 0.1.

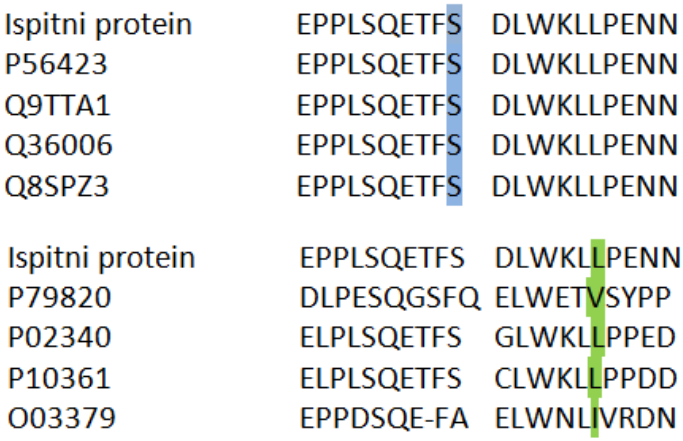
Ako se ne navede željena supstitucija, SIFT će predvidjeti utjecaje supstitucija svih 20 aminokiselina na svim pozicijama ispitnog proteina.

Raznolikost pronađenih poravnanja utječe na pouzdanost alata. Veća raznolikost

[5]Fenotip su vidljive karakteristike organizma kao što je morfologija i ponašanje.

[6]PSSM (engl. *Position Specific Score-Matrix*) je matrica u kojoj se na vertikalnoj osi nalaze aminokiseline, a na horizontalnoj pozicije u proteinu. Svaka pozicija u poravnanju koja je visoko očuvana dobiva velik broj bodova, dok one slabo očuvane dobivaju bodove blizu nule.

povlaći više podataka što je demonstrirano na slici 2.5. Prvi red prikazuje protein na kojem ispitujemo supstitucije, a ostali redovi su pronađena poravnanja. Na lijevoj

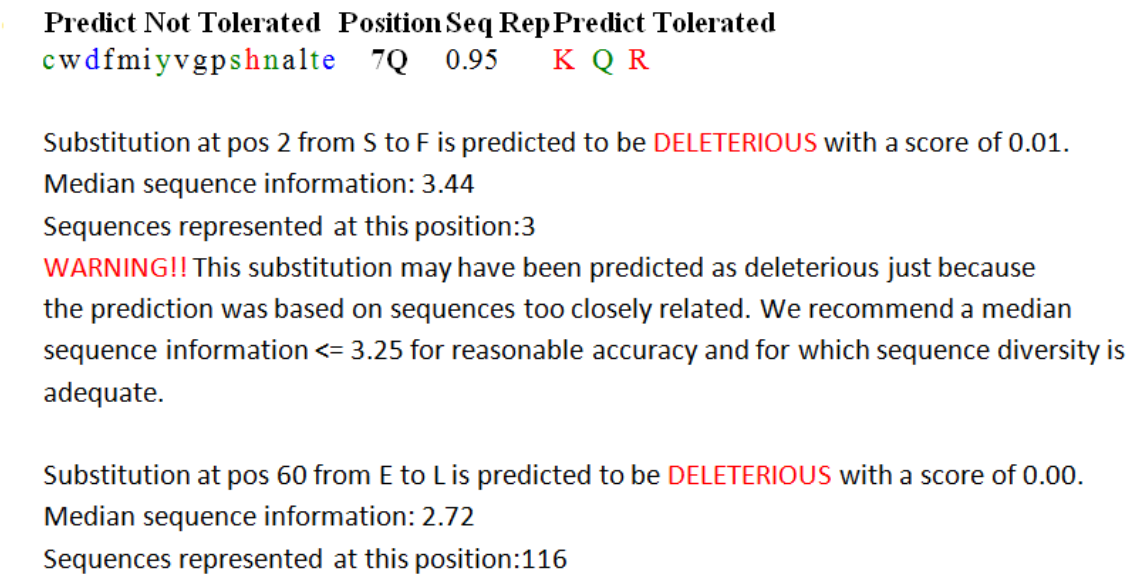


Slika 2.5: Utjecaj raznolikosti homolognih proteina na predikciju aplikacije SIFT.

polovici slike prikazane su identične sekvence što može biti biološka posljedica ili artefakt baze. SIFT će odrediti sve supstitucije na svim pozicijama štetnim jer zbog manjka raznolikosti smatra da su navedene aminokiseline važne za funkciju. Kako bi povećali pouzdanost predikcija, traženi protein treba usporediti s više ne potpuno identitčnih proteina. Na desnoj polovici slike prikazana su različita poravnanja s drugim proteinima. Na poziciji 16 nalaze se aminokiseline Glicin, Valin i Izoleucin što ukazuje da će SIFT tolerirati supstitucije hidrofobnih aminokiselina. Sličnosti dobivenih poravnanja koriste se za izračun medijana očuvanja sekvenci. Njegove vrijednosti protežu se od 0 do 4.32 (gornja granica dobivena je od $\log_2 20$, jer postoji 20 aminokiselina). Vrijednosti između 2.75 i 3.25 smatraju se pouzdanim za predviđanje utjecaja supstitucija. Po iskustvu korisnika vrijednosti do 3.5 također prolaze kao pouzdane. Sve veće vrijednosti indiciraju na premalu raznolikost između pronađenih poravnanja što može biti posljedica baze proteina.

Primjeri izlaza web aplikacije dani su na slici 2.6. Prvi primjer prikazuje rad SIFT-a kada ispituje vjerojatnosti pojavljivanja svih 20 aminokiselina na poziciji 7 gdje se nalazi Glutamin (Q), Arginin (R) i Lizin (K) imaju vjerojatnosti veće od 0.05 (nisu prikazane) te su predočene kao neutralne supstitucije. Broj 0.95 ukazauje da 95% ispitanih poravnanja na poziciji 7 ima aminokiselinu. Svojstva aminokiselina su kodirana bojom: crna označuje nepolarne, zelena polarne, crvena bazične te plava kisele. Drugi i treći primjer prikazuju predikcije za određenu poziciju i željenu supstituciju. Razlika je u medijanu očuvanja sekvenci. U drugom primjeru on iznosi 3.44 što ukazuje na

manjak poravnanja (vidljivo ja da samo 3 poravnanja imaju aminokiselinu na poziciji 2). U trećem primjeru medijan iznosi 2.72 što znači da je predikcija štetnog djelovanja opravdana (116 aminokiselina ima aminokiselinu na poziciji 60).



Slika 2.6: Različiti oblici izlaza aplikacije SIFT.

3. Metode

Svakodnevni porast veličine genomskih i proteinskih baza potiče na razvoj sve bržih aplikacija za pretraživanje sličnih sljedova. Pojava paralelnog programiranja na grafičkim karticama pomogla je razvoju aplikacija te omogućila ponovnu uporabu generalnih algoritama (osobito za lokalno poravnanje). Najpoznatije aplikacije današnjice uz BLAST su GPU-BLAST, CUDA-BLAST, BLAT te Tachyon. Evaluacijom aplikacija GPU-BLAST, BLAT te SW# (engl. *Smith-Waterman Sharp*) pokušat će se zamijeniti PSI-BLAST korak aplikacije SIFT. Razlozi odbacivanja aplikacija CUDA-BLAST i Tachyon su sljedeći: trenutna verzija CUDA-BLASTp aplikacije ne podržava opciju pretraživanja baza za više ispitnih proteina odjednom što će usporiti proces ispitivanja te umanjiti performanse same aplikacije [8]; aplikacija Tachyon ne pokazuje velika ubrzanja na manjim bazama proteina te ju nije moguće dobiti za lokalnu upotrebu [9]. U nastavku poglavlja ukratko su opisane odabrane aplikacije te sam postupak evaluacije dotičnih.

3.1. GPU-BLAST

Aplikacija implementira BLAST algoritam pomoću paralelnog programiranja te programiranja s CUDA[1] tehnologijom. Pažljivo raspoređivanje dužih i kraćih sljedova između grafičke kartice i memorije računala rezultira s ubrzanjem do maksimalno 4 puta s identičnim rezultatima [10]. Takav rad ostvaren je s višedretvenim izvršavanjem na centralnoj procesnoj jedinici (CPU) paralelno s jednom grafičkom procesnom jedinicom (GPU). Trenutna verzija GPU-BLAST-a podržava samo pronalaženje proteinskih poravnanja i to za više ispitnih proteina odjednom, ali nastoji se izraditi verzija za poravnanje sljedova nukleotida kao i PSI-BLAST algoritam koji je senzitivniji kod pronalaženja manje sličnih sljedova.

[1]CUDA (engl. *Compute Unified Device Architecture*)je platforma za paralelno programiranje na grafičkim procesorima koja drastično povećava računalne performanse. Proizvod je kompanije grafičkih kartica nVIDIA-e.

3.2. BLAT

BLAT (engl. *BLAST-Like Alignment Tool*) je alat za pretraživanje sličan BLAST-u. Dizajniran je za brzo pronalaženje sljedova DNA sa sličnošću većom od 95% te duljina većih od 40 nukleotida. Za proteine pronalazi one s više od 80% sličnosti te duljina većih od 20 aminokiselina. Razlike između BLAT-a i BLAST-a su sljedeće:

- BLAST generira listu podnizova iz ispitnog proteina te sljedno pretražuje bazu podataka. BLAT u radnu memoriju računala sprema indeksiranu ispitnu bazu. Indeksi predstavljaju podnizove od 11 nukleotida ili 4 aminokiselina te sljedno pretražuje ispitni protein.
- BLAST traži podnizove čija sličnost ne pada ispod nekog praga. BLAT traži samo savršene podnizove, one koji imaju najviše jednu ili dvije točkaste supstitucije.
- U koraku proširenja podnizova BLAT spaja više od dva najbliža.
- BLAT spaja visoko ocjenjena lokalna poravnjana.

Takav način pretraživanja daje ubrzanje od 50 puta uz cijenu manjeg broja poravnanja u odnosu na BLAST.

3.3. SW#

SW# (engl. *Smith-Waterman Sharp*) je aplikacija koja implementira Smith-Waterman algoritam pomoću CUDA tehnologije. Kako bi se smanjila memorijska složenost s polinomne na linearnu (s $O(mn)$ na $O(min(m,n))$) upotrebljen je Hirschbergov algoritam [11]. Za veće proteine ubrzanja se kreću od 10 puta sve do 400 puta za manje sljedove nukleotida u odnosu na neparalelizirani algoritam [12]. Parametri CUDA tehnologije, broj blokova i broj dretvi po bloku, dobiveni su uporabom genetskog algoritma[2] koji cijeli proces pronalaska poravnanja dodatno ubrzava za 15-20% [12]. Aplikacija podržava rad s više dretva i više kartica gdje dretve pronalazak poravnanja izvršavaju ili na centralnoj procesnoj jedinici ili na jednoj od dostupnih grafičkih procesnih jedinica. Također je omogućen rad s više računala pomoću tehnologije MPI[3] što ovisno o broju računala daje daljna ubrzanja [13].

[2] Optimizacijski algoritam inspiriran evolucijom.

[3] MPI (engl. *Message Passing Interface*) je protokol koji omogućuje izvršavanje aplikacija na više računala neovisno o programskom jeziku u kojem su izrađene.

3.3.1. Dodatne optimizacije

Aplikacija SW# uspoređuje svaki ispitni protein sa svakom sekvencom u bazi što je vremenski zahtjevan posao. Da bi se postiglo dodatno ubrzanje, uveli smo novi početni korak koji bi trebao smanjiti prostor pretraživanja [13]. Korak je sličan pronalasku podnizova kod aplikacija BLAT i BLAST. Najprije se baza proteina spremi u tablicu raspršenog adresiranja. Spremaju se parovi koji sadrže podniz duljine 4 ili 5 aminokiselina (ovisno koja implementacija se koristi) te polje brojeva. Svaki broj polja označava poziciju sekvence u bazi koja sadrži dotični podniz. Takva struktura podataka omogućuje operaciju dohvata u konstantnom vremenu (vremenska složenost je $O(1)$). Nakon stvaranja tablice generiraju se podnizovi od ispitnog proteina. Za svaki podniz traže se indeksi potencijalno značajnih sekvenca koji se zatim predaju Smith-Waterman dijelu. Na opisan način smanjuje se prostor pretraživanja baze od 40% s podnizovima duljine 4 do 85% s podnizovima duljine 5. Smanjenje prostora ubrzava rad aplikacije do najviše 2.5 puta [13]. Zbog smanjenja broja pronađenih poravnanja dodana je opcija koja modificira traženje podnizova u tablici. Umjesto traženja potpuno identičnih podnizova traže se podnizovi koji se razlikuju za najviše jednu aminokiselinu što umanjuje dobivena ubrzanja.

3.4. Evaluacija aplikacija za pretraživanje sljedova

Aplikacija SIFT pomoću aplikacije PSI-BLAST, koja troši veći dio vremena, predviđa da li neka supstitucija aminokiseline djeluje štetno ili neutralno na funkciju proteina. Zbog rasta potražnje baza proteinskih supstitucija drugih vrsta organizama, javlja se potreba za bržim i efikasnijim alatom. Da bi pronašao najprikladniji alat mjerit ću vrijeme izvođenja aplikacija GPU-BLAST, BLAT, SW# te SW# uz indeksiranje baze proteina. Alati ne smiju loše utjecati na predikcije SIFT-a koje daje kad je kombiniran s PSI-BLAST algoritmom. Iz tog razloga koristit ću binarnu klasifikaciju za zadani ispitni skup kako bi odredio točnost, preciznost, osjetljivost, specifičnost te pokrivenost pojedinih alata.

3.4.1. Binarna klasifikacija

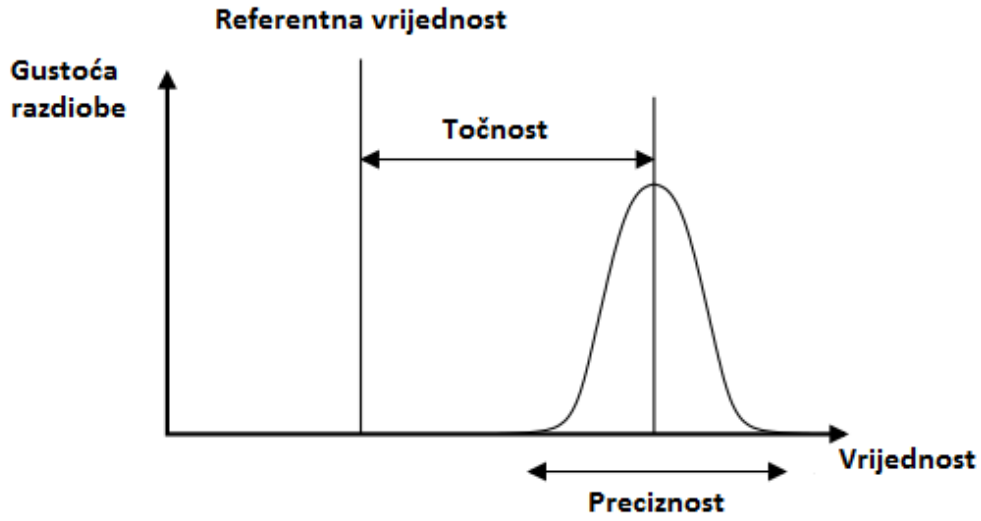
Razvrstavanje objekata na temelju informacije posjeduje li objekt neko svojstvo ili ne naziva se binarna klasifikacija. Primjer je testiranje pacijenata na neku bolest gdje je klasifikacijsko svojstvo prisustvo bolesti. Rezultati testova mogu biti ili pozitivni

ili negativni bez obzira na točnu situaciju pacijenta. Kombinacija stvarnih i testom dobivenih rezultata daje sljedeću podjelu:

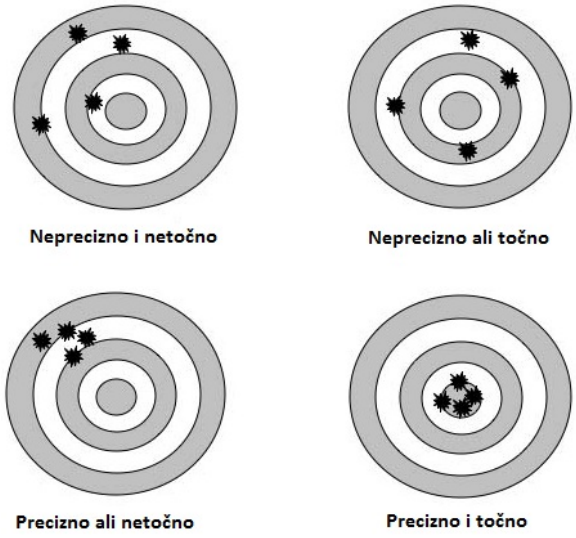
- ispravan pozitiv (engl. *True Positive, TP*) - dijagnosticirana bolest kod bolesnog čovjeka (ispravno identificiran).
- pogrešni pozitiv (engl. *False Positive, FP*) - dijagnosticirana bolest kod zdravog čovjeka (pogrešno identificiran).
- ispravan negativ (engl. *True Negative, TN*) - zdrav čovjek proglašen zdrav (ispravno odbijen).
- pogrešni negativ (engl. *False Negative, FN*) - bolestan čovjek proglašen zdrav (pogrešno odbijen).

Pomoću ispravnih/pogrešnih pozitiva/negativa određuju se točnost, preciznost, osjetljivost i specifičnost.

Točnost (engl. *accuracy*) alata definira se kao mjera bliskosti između izmjerenih vrijednosti i točnih vrijednosti dok je preciznost (engl. *precision*) mjera bliskosti između više ponovljenih mjerenja u nepromijenjenoj okolini [14]. Razlika se može vidjeti na slici 3.1. Mjerenje može biti točno, ali neprecizno ili bilo koja kombinacija ovih mjera. Zbog lakšeg razumijevanja i razlučivanja često se povlači analogija sa streljaštvom. Na slici 3.2. prikazane su četiri kombinacije mjera točnosti i preciznosti.



Slika 3.1: Grafički prikaz mjera točnosti i preciznosti.



Slika 3.2: Prikaz kombinacija mjera točnosti i preciznosti u streljaštvu.

U binarnoj klasifikaciji točnost i preciznost računaju se pomoću formula (3.1) i (3.2), gdje oznaze TP, TN, FP i FN označavaju broj ispravnih pozitiva, ispravnih negativa, pogrešnih pozitiva te pogrešnih negativa.

$$točnost = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

$$preciznost = \frac{TP}{TP + FP} \tag{3.2}$$

Osjetljivost (engl. *sensitivity*) je mjera koja iskazuje svojstvo alata da prepozna pozitivne rezultate [14]. Na prethodnom primjeru osjetljivost bi bila broj točno dijagnosticiranih bolesti na skupu bolesnih ljudi. Specifičnost (engl. *specificity*) je suprotna mjera, tj. broj ljudi ispravno proglašenih zdravim na skupu zdravih ljudi. Formalnije, to je mjera koja ikazuje svojstvo alata da prepoznaje negativne alate. Kako bi izračunali mjere osjetljivosti i specifičnosti koriste se formule (3.3) i (3.4).

$$osjetljivost = \frac{TP}{TP + FN} \tag{3.3}$$

$$specifičnost = \frac{TN}{TN + FP} \tag{3.4}$$

Pokrivenost (engl. *coverage*) alata također je mjera za ocjenu alata ali ona ne spada u binarnu klasifikaciju te se jednostavno može izračunati pomoću formule (3.5).

$$pokrivenost = \frac{broj \quad uspješnih \quad testova}{ukupni \quad broj \quad testova} \tag{3.5}$$

3.4.2. Ispitni skupovi

Skup podataka za ispitivanje sastoji se od ljudskih proteina te njihovih aminokiselinskih supstitucija. Dijeli se u dvije grupe:

- engl. *human divergence* (humDiv) - sadrži popis štetnih supstitucija za koje se smatra da uzrokuju Mendelovu bolest kod ljudi. Neutralna lista supstitucija dobivena je usporedbom ljudskih proteina s homolognim proteinima usko povezanih sisavaca [15].
- engl. *human variation* (humVar) - sadrži štetne supstitucije koje uzrokuju bilo kakve ljudske bolesti, dok su neutralne one koje ne uzrokuju [15].

Analogno s primjerom testiranja bolesti možemo odrediti ispravne/pogrešne pozitive/ negative kod supstitucija. Broj proteina, njihova ukupna duljina i broj supstitucija iz skupina humDiv i humVar prikazani su u tablici 3.1.

Tablica 3.1: Ispitni skup

Grupa proteina	Vrsta supstitucije	Broj supstitucija	Broj proteina	Ukupna duljina
humDiv	Neutralna	6027	315	235534
	Štetna	3055	493	363110
humVar	Štetna	12598	1101	801600
	Neutralna	8638	3400	2267306

Za pronalaženje poravnanja koristit ću bazu Swiss-Prot [16] koja sadrži 518415 proteina ukupne duljine od 372270693 aminokiselina (veličine $250MB$). Kako bi izmjerio vrijeme izvođenja koristit ću dvije baze na dva različita stroja. Baze su Swiss-prot i UniRef90 [16] koja sadrži 14650230 proteina ukupne duljine 5110404254 (veličine $6.2GB$).

U tablici 3.2 prikazane su referentne vrijednosti parametara binarne klasifikacije za kombinaciju alata SIFT i PSI-BLAST. Cilj je pronaći alat koji će u kombinaciji sa SIFT-om imati upravo takve ili bolje parametre.

Tablica 3.2: Vrijednosti parametara binarne klasifikacije za kombinaciju aplikacija SIFT i PSI-BLAST.

Grupa proteina	Točnost	Preciznost	Osjetljivost	Specifičnost	Pokrivenost
humDiv	0.842	0.798	0.714	0.907	0.911
humVar	0.718	0.794	0.712	0.726	0.827

4. Rezultati

Rezultati provedenog mjerenja su brzine izvođenja te parametri binarne klasifikacije (točnost, preciznost itd.) za alate BLAT, GPU-BLAST, SW# te SW# s indeksiranom bazom. Kako bi se demonstrirala moć paralelnog programiranja na grafičkim karticama dodana su i vremena aplikacije BLAST. Točna vremena algoritma PSI-BLAST nisu sadržana u ispitnom skupu, ali zasigurno su veća od vremena izvođenja aplikacije BLAST (PSI-BLAST 2 puta izvodi BLAST). U nastavku je opisano sklopovlje na kojem su izvršena mjerenja, način pokretanja aplikacija te brzine izvođenja. Prikazan je utjecaj smanjenja prostora pretraživanja na brzinu i točnost izvođenja aplikacije SW#. Mjere točnosti, preciznosti, osjetljivosti, specifičnosti i pokrivenosti dane su na kraju poglavlja za kombinacije navedenih alata sa SIFT-om.

4.1. Vrijeme izvođenja

Sklopovlje na kojem se izvode mjerenja ima najveći utjecaj na rezultate mjerenja. Za aplikacije GPU-BLAST i SW# osobito su važne grafičke kartice, dok je za aplikacije BLAT i BLAST relevantna snaga centralne procesne jedinice (CPU). Alati su tražili maksimalno 400 poravnanja uz ograničenje maksimalne E-vrijednosti[1] koja iznosi 10^{-4}. Kako aplikacija BLAT nema opciju za E-vrijednost, postavio sam minimalnu sličnost na 30% (što će se pokazati dosta nezgodno). Vrijeme je mjereno na *Linux* okruženju pomoću naredbe *time* [17] na dva različita stroja. Na manjoj bazi (Swiss-Prot) mjerena su vremena za svaku skupinu proteina kako bi se kasnije mogle izračunati potrebne vrijednosti za ocjenu točnosti i ostalih parametara. Ova mjerenja izvršena su na stroju čije su karakteristike:

- *Intel Core™2 Quad CPU Q6600 @ 2.40GHz*
- *nVIDIA™Corporation GF110 [Geforce GTX 570]*

[1] E-vrijednost je očekivanje, tj. parametar koji opisuje očekivani broj pogodaka kod pretraživanja baze proteina određene veličine. Sve niža E-vrijednost (sve bliža nuli) označuje sve značajniji pogodak. Pošto u obzir uzima duljine ispitnih sekvenci, kraća poravnanja će imati veće E-vrijednosti.

– *8 GB radne memorije*

Zbog veličine baze UniRef90, mjerenja su izvršena samo na skupu od 315 proteina iz skupine humDiv i to na sklopovlju boljih karakteristika:

– *Intel Core™i7-3770 @ 2.40GHz*

– *nVIDIA™Corporation GK104 [GeForce GTX 690] x 2*

– *16 GB radne memorije*

Vremena izvođenja aplikacija za bazu Swiss-prot prikazana su u tablici 4.1. Utjecaj prostora pretraživanja na vrijeme aplikacije SW# prikazan je u tablici 4.2. Vremena izvođenja na bazi UniRef90 prikazana su u tablici 4.3.

Tablica 4.1: Vrijeme izvođenja aplikacija s bazom Swiss-Prot.

Skup proteina	Broj	BLAST	BLAT	GPU-BLAST	SW#
humDiv	315	$46m17.885s$	$0m38.306s$	$31m27.689s$	$26m32.460s$
	493	$66m39.264s$	$0m51.738s$	$43m56.203s$	$40m55.935s$
humVar	1101	$145m50.906s$	$1m49.702s$	$97m9.046s$	$89m12.645s$
	3400	$415m44.097s$	$4m34.045s$	$271m38.197s$	$264m51.385s$

Tablica 4.2: Vrijeme izvođenja aplikacije SW# s indeksiranom bazom Swiss-Prot. Vrijednosti za svaku inaćicu alata su broj kandidata - vrijeme.

Skup proteina	Broj	SW# s4	SW# s5	SW# s5 p
humDiv	315	388448-$28m44.303s$	62680-$14m33.341s$	256257-$23m46.271s$
	493	388770-$43m16.044s$	62110-$20m56.995s$	257197-$35m53.141s$
humVar	1101	386108-$95m10.601s$	61750-$47m5.602s$	254756-$79m2.103s$
	3400	375470-$270m43.804s$	56614-$131m26.837s$	242009-$221m46.495s$

4.2. Ocjena parametara binarne klasifikacije

Poravnjanja ispitnih proteina sa Swiss-prot bazom predao sam SIFT-u kako bi dobio predikcije za sve supstitucije iz tablice 3.1. Za izračun mjere točnosti i ostalih parametara pojedine aplikacije koristio sam formule (3.1) do (3.5). Brojevi ispravnih/pogrešnih pozitiva/negativa iz oba skupa supstitucija su u tablicama 4.4 i 4.6. Parametri binarne klasifikacije pojedine aplikacije prikazani su u tablicama 4.5 i 4.7. Zbog preglednosti prikaza, dodana su dva stupčasta dijagrama (slika 4.1 i slika 4.2) koji prikazuju odnose parametara kandidatnih aplikacija s parametrima PSI-BLAST algoritma.

Tablica 4.3: Vrijeme izvođenja aplikacija s bazom UniRef90 te ispitnim skupom od 315 proteina.

Aplikacija	Vrijeme
BLAST	$612m16.067s$
BLAT	$neuspjeh$
GPU-BLAST	$578m51.831s$
SW#	$232m9.275s$
SW s5	$neuspjeh$

Tablica 4.4: Brojevi ispravnih/pogrešnih pozitiva/negativa za humDiv skup supstitucija.

Aplikacija	TP	FN	TN	FP
BLAT	2613	153	3764	1984
GPU-BLAST	2095	773	4789	639
SW#	2032	854	5308	486
SW# s5	2089	811	5290	483

Tablica 4.5: Vrijednosti parametara binarne klasifikacije za kombinaciju ispitnih aplikacija sa SIFT-om. Skup supstitucija je humDiv.

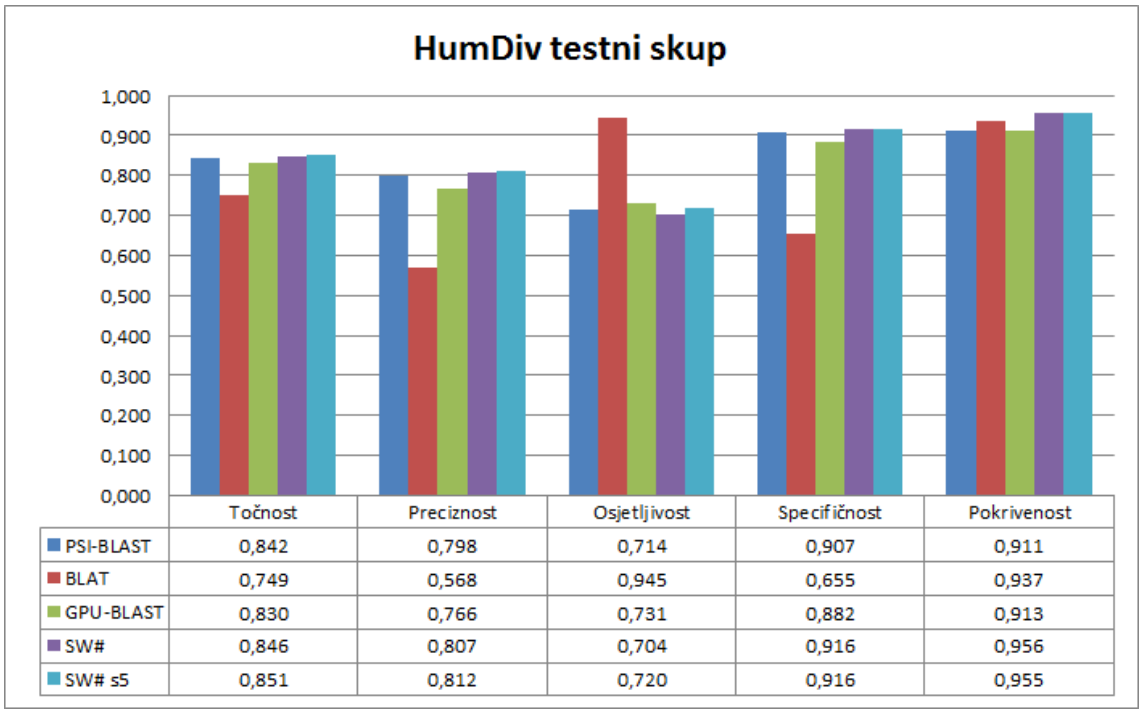
Aplikacija	Točnost	Preciznost	Osjetljivost	Specifičnost	Pokrivenost
PSI-BLAST	0.842	0.798	0.714	0.907	0.911
BLAT	0.749	0.568	0.945	0.655	0.937
GPU-BLAST	0.830	0.766	0.731	0.882	0.913
SW#	0.846	0.807	0.704	0.916	0.956
SW# s5	0.851	0.812	0.720	0.916	0.955

Tablica 4.6: Brojevi ispravnih/pogrešnih pozitiva/negativa za humVar skup supstitucija.

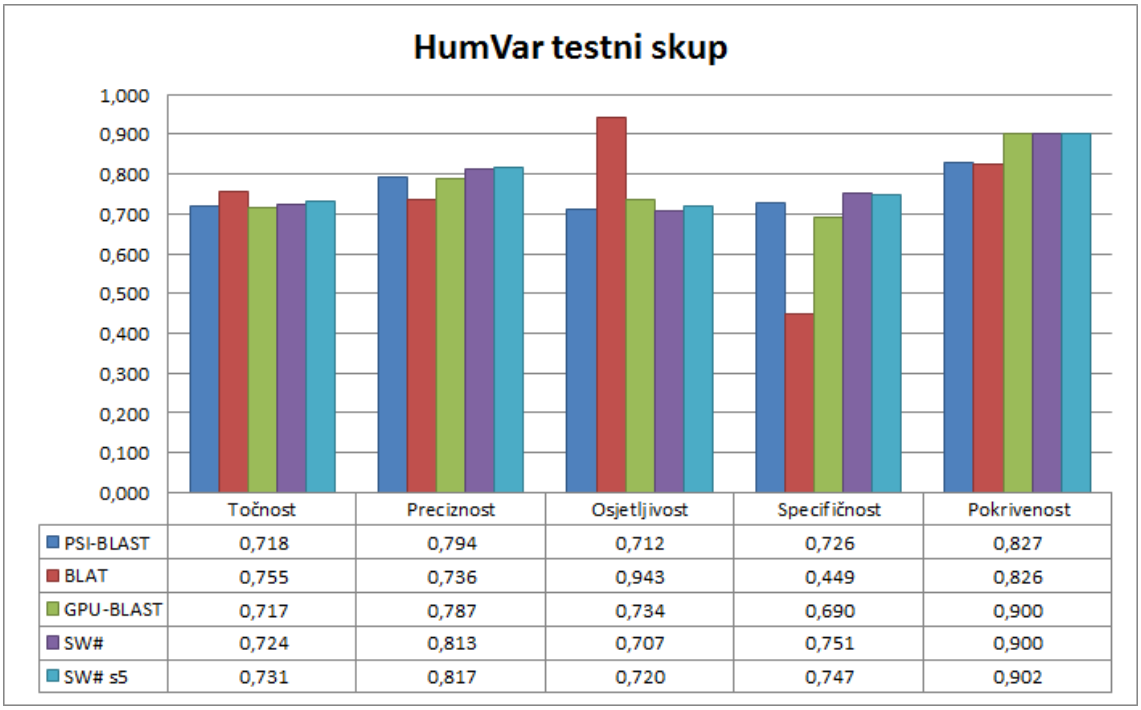
Aplikacija	TP	FN	TN	FP
BLAT	10246	619	3004	3681
GPU-BLAST	8560	3101	5142	2311
SW#	8185	3386	5659	1877
SW# s5	8422	3266	5577	1885

Tablica 4.7: Vrijednosti parametara binarne klasifikacije za kombinaciju ispitnih aplikacija sa SIFT-om. Skup supstitucija jest humVar.

Aplikacija	Točnost	Preciznost	Osjetljivost	Specifičnost	Pokrivenost
PSI-BLAST	0.718	0.794	0.712	0.726	0.827
BLAT	0.755	0.736	0.943	0.449	0.826
GPU-BLAST	0.717	0.787	0.734	0.690	0.900
SW#	0.724	0.813	0.707	0.751	0.900
SW# s5	0.731	0.817	0.720	0.747	0.902



Slika 4.1: Grafički prikaz parametara binarne klasifikacije za skup humDiv.



	Točnost	Preciznost	Osjetljivost	Specifičnost	Pokrivenost
PSI-BLAST	0,718	0,794	0,712	0,726	0,827
BLAT	0,755	0,736	0,943	0,449	0,826
GPU-BLAST	0,717	0,787	0,734	0,690	0,900
SW#	0,724	0,813	0,707	0,751	0,900
SW# s5	0,731	0,817	0,720	0,747	0,902

Slika 4.2: Grafički prikaz parametara binarne klasifikacije za skup humVar.

5. Diskusija

Dobiveni rezultati vrlo su zadovoljavajući. Sve aplikacije višestruko su brže u odnosu na BLAST algoritam te ujedno i na PSI-BLAST algoritam. To je bilo i očekivano pošto je u pitanju paralelno programiranje na grafičkim karticama. Koja aplikacija najviše odgovara SIFT-u objasnit ću u nastavku.

Strahovito ubrzanje pokazala je aplikacija BLAT koja sprema indeksiranu bazu u radnu memoriju. Razlog takvom ubrzanju je veličina izlaza, tj. broj generiranih poravnanja. Naime, BLAT se koristi za pronalaženje sekvenci koje su slične 80% ili više. Veličina izlaza kojeg generira za 3400 ispitnih proteina iznosi $340MB$, dok aplikacije GPU-BLAST i SW# generiraju izlaze od $980MB$ i $1.2GB$, respektivno. Kako nisam bio u mogućnosti namjestiti broj poravnanja niti željeni prag E-vrijednosti, poravnanja koja je BLAT pronašao nisu bila dovoljna za pouzdan rad SIFT-a. SIFT-u je potreban što veći skup različitih poravnanja kako bi na temelju više informacija predvidio utjecaj supstitucije. Medijani skoro svih predikcija prelaze vrijednost od 3.9. Velika vrijednost osjetljivosti (mjera točno predviđenih štetnih utjecaja iz skupa štetnih mutacija) te mala specifičnosti (mjera točno predviđenih neutralnih utjecaja iz skupa neutralnih mutacija) posljedica su činjenice da SIFT s malo informacija sve supstitucije proziva štetnim. Još jedna mana BLAT-a jest neuspješno izvržavanje na većoj bazi (UniRef90). Moguć uzrok jest manjak radne memorije, ali nisam se upuštao u daljna istraživanja. Iz navedenih razloga aplikacija BLAT ne zadovoljava kriterije za zamjenu PSI-BLAST algoritma.

Aplikacija GPU-BLAST pokazuje značajno ubrzanje u odnosu na BLAST algoritam. Maksimalno ubrzanje koje postiže je 1.6 puta na skupu od 1101 proteina. Zanimljivo je malo ubrzanje na jačem sklopovlju što ukazuje na nedostatak korištenja samo jedne grafičke kartice. Velika mana algoritma su preniske vrijednosti mjera točnosti, preciznosti i ostalih parametara. To je vidljivo na dijagramima sa slika 4.1 i 4.2. Razlog tome je senzitivnost PSI-BLAST algoritma u pronalasku manje sličnih sljedova. Kako on koristi BLAST za izradu supstitucijske matrice specifičnih pozicija te zatim pomoću nje i BLAST-a ponovo pretražuje bazu, rezultati nisu toliko neočekivani.

Na temelju dobivenih vrijednosti mjera možemo zaključiti da SIFT-u poravnanja koja prolanazi BLAST algoritam nisu dovoljna. Zbog navedenih razloga niti GPU-BLAST nije aplikacija dovoljno pouzdana za rad SIFT-a.

Očigledni pobjednik je aplikacija SW#. Broj poravnanja te njihova točnost posljedica su optimalnosti Smith-Waterman algoritma. Ubrzanje postignuto pomoću CUDA tehnologije doseže i do 2.67 puta na stroju s dvije grafičke kartice. SW# ima najveće vrijednosti točnosti, preciznostim, specifičnosti i pokrivenosti. Jedino GPU-BLAST nadmašuje mjeru osjetljivosti na skupu humDiv što možemo lako zanemariti s obzirom na sveukupne prednosti. Novi početni korak, koji dodaje heuristično ponašanje SW#-u, u mogućnosti je dodatno povećati ubrzanje. Ispitivanjem podnizova različite duljine s mogućom uporabom permutacija kako bi se dobilo više kandidata, pronašao sam najbolje rješenje za dane ispitne skupove. To su podnizovi duljine 5 bez permutacija. Prostor pretraživanja se ovakvim postavkama smanjuje do 85% što rezultira u ubrzanju do 2 puta. Očekivano ubrzanje za ovakvu redukciju prostora pretraživanja je dosta veće, ali način rada aplikacije SW# ne omogućuje bolju izvedbu. Mjere točnosti, preciznosti te ostalih parametara čak su više od mjera SW#-a što je posljedica manjeg broja pronađenih poravnanja (do 10% manje poravnanja na svim ispitnim skupovima). Naime, SIFT zadaje ograničeno vrijeme za pronalazak predikcija te se događa situacija da SIFT prekine rad kada SW# pronađe 400 poravnanja za proteine duljina većih od 2000 aminokiselina. Jedina mana indeksiranja je neuspjeh izvođenja za bazu UniRef90. Razlog je veličina tablice raspršenog adresiranja koja nije uspješno pohranjena u radnu memoriju. MPI tehnologija, koja u ovom radu nije ispitana, otvara vrata dodatnim ubrzanjima.

6. Zaključak

U radu je opisan problem aplikacije SIFT, na koji način ga riješiti te je razrađen postupak evaluacije najpoznatijih alata za pretraživanje baza proteinskih sljedova. Kao potencijalno rješenje odabrana je aplikacija Smith-Waterman Sharp ili kraće SW#, koja ubrzava pronalazak sličnih proteinskih sljedova do 2.6 puta na strojevima s više grafičkih kartica. Veća točnost, preciznost, osjetljivost, specifičnost i pokrivenost na skupu proteina ljudskog organizma, velika su odlika odabrane aplikacije. Dodatno ubrzanje moguće je postići uvođenjem dodatnog koraka koji pomaže u smanjenju prostora pretraživanja, tj. dodaje heuristični element već optimalnom algoritmu. Metoda pomoću indeksirane baze proteina smanjuje broj kandidata, koji ulaze u Smith-Waterman dio, od 40% do 85%. Maksimalno dobiveno ubrzanje je 2 puta za podnizove od 5 aminokiselina uz najviše 10% izgubljenih poravnanja u odnosu na regularni SW#. Manji broj poravnanja čak je poboljšao parametre binarne klasifikacije. Mogućnost izvođenja aplikacije na više računala pomoću MPI tehnologije otvara vrata dodatnim ubrzanjima.

LITERATURA

[1] N.M. Luscombe, D. GreenBaum, i M. Gernstein, *What is bioinformatics? An introduction and overview*, 2001.

[2] Sean R. Eddy, *What is dynamic programming?*, 2004.

[3] Pauline C. Ng i Steven Henikoff, *SIFT: predicting amino acid changes that affect protein function*, 2003.

[4] Saul B. Needleman i Christian D. Wunsch, *A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins*, 1969.

[5] T.F. Smith i M.S. Waterman, *Identification of Common Molecular Subsequences*, 1981.

[6] M.O. Dayhoff, R.M. Schwartz, i B.C. Orcutt, *A Model of Evolutionary Change in Proteins*, 1978.

[7] S.F. Altschul, W. Gish, i W. Miller, *Basic Local Alignment Search Tool*, 1990.

[8] W. Liu, B. Schmidt, i W. Muller-Witting, *CUDA-BLASTP: Accelerating BLASP on CUDA-Enabled Graphics Hardware*, 2011.

[9] J. Tan, D. Kuchibhatla, i S. Maurer-Stroh, *Tacyhon search speeds up retrieval of similar sequences by several orders of magnitude*, 2012.

[10] P.D. Vouzis i N.V. Sahinidis, *GPU-BLAST: using graphics processors to accelerate protein sequence alignment*, 2010.

[11] D.S. Hirschberg, *A linear space algorithm for computing maximal common subsequences*, 1975.

[12] Matija Korpar, *Implementacija Smith Waterman algoritma koristeći grafičke kartice s CUDA arhitekturom*, 2011.

[13] D. Pavlović, R. Vaser, M. Korpar, i M. Šikić, *Protein database search optimization based on CUDA and MPI*, MIPRO 2013 - 36th International Convention.

[14] International vocabulary of metrology, *Basic and general concepts and associate terms (VIM)*, 2008.

[15] NL Sim, Kumar P, Hu J, Henikoff S, Schneider G, i Ng PC, *SIFT web server: predicting effects of amino acid substitutions on proteins*, 2012.

[16] Protein knowledgebase, `http://www.uniprot.org/`, 2011.

[17] Time manual, `http://ss64.com/bash/time.html`, 2011.

Evaluacija aplikacija za pretraživanje baze proteinskih sljedova

Sažetak

Aplikacija SIFT najkorišteniji je alat za predviđanje utjecaja aminokiselinskih supstitucija na funkcije proteina. Njen glavni problem je PSI-BLAST korak koji troši veliku količinu vremena kako bi pronašao slične sekvence ispitnog proteina. U ovom radu razrađena je evaluacija najpoznatijih aplikacija za pretraživanje baze proteinskih sljedova: BLAT, GPU-BLAST te SW#. Kriterij odabira aplikacije, koja će zamijenit PSI-BLAST, temelji se na brzini izvođenja te na vrijednostima parametara binarne klasifikacije, tj. mjere točnosti, preciznosti, osjetljivosti, specifičnosti i pokrivenosti alata. Najbolje karakteristike pokazala je aplikacija SW# s ubrzanjem do 2.6 puta (u odnosu na BLAST) na strojevima s više grafičkih kartica te boljim vrijednostima točnosti, preciznosti i ostalih parametara. Razmatrana je dodatna optimiziacija SW#-a, tj. uvođenje heurističnog koraka koji će smanjiti prostor pretraživanja. Pomoću podnizova duljine 5 dobiveno je dodatno ubrzanje do 2 puta te poboljšanje parametara binarne klasifikacije.

Ključne riječi: SIFT, aminokiselinske supstitucije, poravnanje sljedova, BLAST, SW#, bioinformatika

The evaluation of protein database searching tools

Abstract

SIFT is the most widely used tool to predict whether an amino acid substitution has neutral or deleterious effect on a protein function. Its main problem is the time-consuming PSI-BLAST step which generates sequence alignments. Therefore, this paper shows the evaluation of the best-known protein database searching tools: BLAT, GPU-BLAST and SW#. The criterion whether a tool colaborates fine with SIFT or not consists of time needed to generate sequence alignments and values of the binary classification parameters which are the following: accuracy, precision, sensitivity, specificity and coverage. The best solution for the given problem is SW#, a tool which can speed up the alignment stage up to 2.6 times on more graphic cards with even better values for accuracy, precision and the other parameters. To get an even better boost in speed, the paper shows an optimisation that adds a new step to SW# which uses heuristics to reduce the database search space. With seeds of length 5, SW# gets a speed boost up to 2 times and has higher values of the binary classification parameters. The basic speed ups are compared to the BLAST algorithm.

Keywords: SIFT, amino acid substitutions, sequence alignments, BLAST, SW#, bioinformatics