Generativni model za traženje terapeutskih peptidnih sekvenci

Antunović, Luka

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij računarstva

Završni rad

GENERATIVNI MODEL ZA TRAŽENJE TERAPEUTSKIH PEPTIDNIH SEKVENCI

Rijeka, rujan 2024.

Luka Antunović

0069090935

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Sveučilišni prijediplomski studij računarstva

Završni rad

GENERATIVNI MODEL ZA TRAŽENJE TERAPEUTSKIH PEPTIDNIH SEKVENCI

Mentor: doc. dr. sc. Goran Mauša

Rijeka, rujan 2024.

Luka Antunović

0069090935

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 17.03.2024.

Zavod: Zavod za računarstvo

Predmet: Programsko inženjerstvo

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Luka Antunović (0069090935)**

Studij: Sveučilišni prijediplomski studij računarstva (1035)

Zadatak: **Generativni model za traženje terapeutskih peptidnih sekvenci / Generative model for searching therapeutic peptide sequences**

Opis zadatka:

Proučiti postojeće generativne modele, s naglaskom na one korištene u kemiji peptida. Objasniti njihove konceptualne razlike te prednosti i nedostatke. Preuzeti postojeći generativni model zasnovan na sprezi genetskog algoritma i strojnog učenja te ga prilagoditi generiranju peptidnih sekvenci terapeutske namjene. Analizirati konvergenciju algoritma pretraživanja za generiranje peptidnih sekvenci kod ponovljenih mjerenja u nasumično odabranim početnim uzorcima.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
izv. prof. Goran Mauša

Predsjednik povjerenstva za
završni ispit:
prof. dr. sc. Miroslav Joler

Izjava o samostalnoj izradi rada

Izjavljujem da sam rad izradio samostalno uz stručnu pomoć mentora izv. prof. dr. sc. Gorana Mauše.

Rijeka, rujan 2024.

_____

Luka Antunović

Zahvala

Zahvaljujem se mentoru izv. prof. dr. sc. Goranu Mauši i asistentu Marku Njirjaku na stručnoj pomoći, te obitelji, prijateljima i djevojci na podršci tokom izrade rada.

Sadržaj

1. Uvod

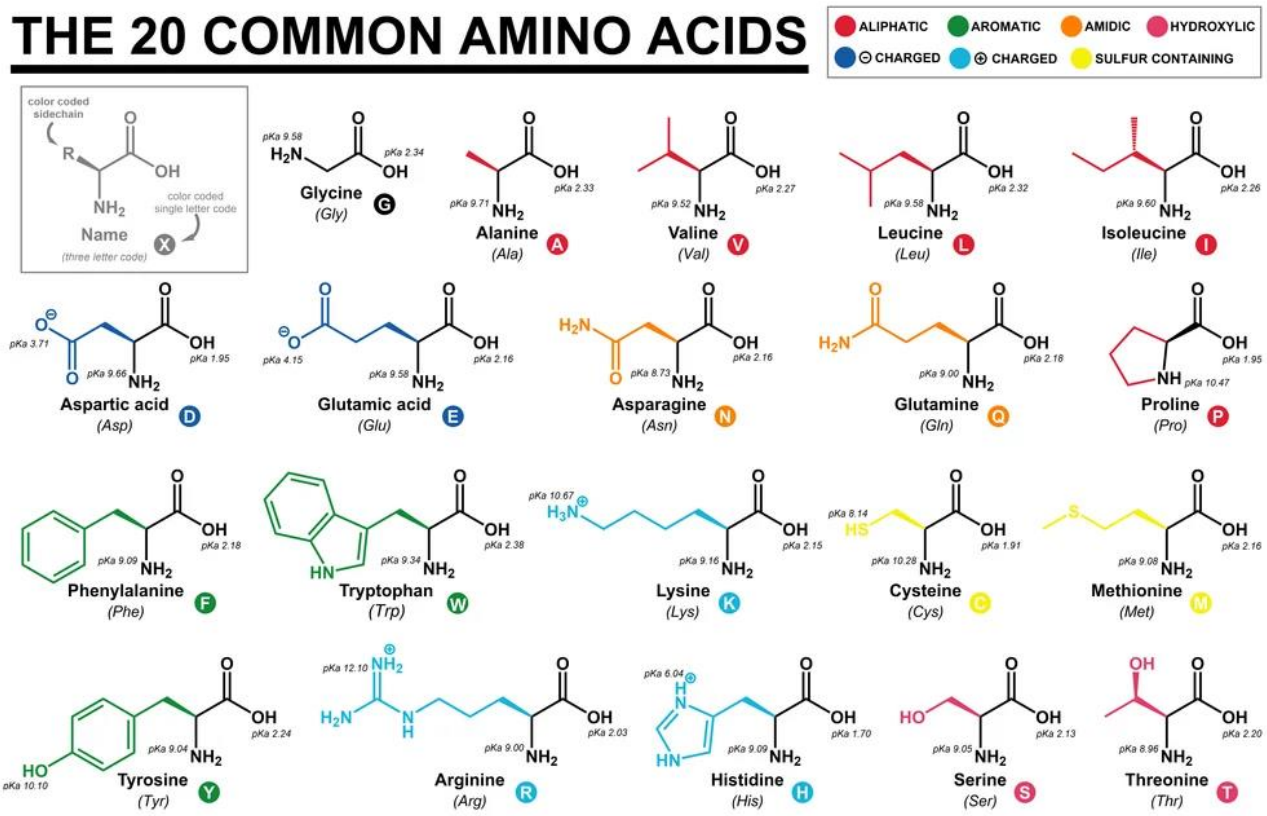
Ovaj završni rad izrađen je u okviru istraživačkih projekata pod naslovom "Predviđanje terapeutskih peptida zasnovano na dubokom učenju" (UNIRI-23-78) te "Dizajn pametnih peptidnih nanostruktura sa ispoljavajućim funkcijama" (UNIRI-23-16). Cilj ovog rada bio je istražiti generativni model zasnovan na sprezi genetskog algoritma i strojnog učenja te ga prilagoditi generiranju peptidnih sekvenci terapeutskih namjena. Nadalje, u ovom radu predstavit će se prednosti i nedostatci kod ponovljenog postupka generiranja peptida u različitim početnim uvjetima, proučit će se svojstva tako dobivenih populacija peptida, analizirat će se konvergencija algoritma pretraživanja za generiranje peptidnih sekvenci kod ponovljenih mjerenja u nasumično odabranim početnim uzorcima te objasniti zaključci koji su iz navedenog dobiveni. Također će se predložiti mogući načini unaprjeđenja i nadogradnje postojećeg generativnog modela za daljnje istraživanje i proučavanje šireg peptidnog prostora. Istraživanje terapeutskih peptida u modrenom dobu od posebne je važnosti jer omogućuje razvijanje novih načina borbe protiv patogena koji danas brže nego ikad razvijaju otpornost na postojeće lijekove, antibiotike i cjepiva [1]. Potkrepljeno činjenicom da terapeutski peptidi u prirodi već postoje i takve je potrebno raspoznati kako bi se mogli upotrijebiti u ljudsku korist, a na tržištu antibiotika i dalje nema puno primjera, ali i željom za pronalaskom još neistraženih sintetičkih spojeva, generativni model omogućava istraživanje širokog peptidnog prostora s ciljem pronalaska novih terapeutskih peptidnih sekvenci.

2. Peptidi

Peptidi su definirani kao kratki, kontinuirani i nerazgranati lanci aminokiselina koji su u pravilu veličine dvije do pedeset aminokiselina [2]. Lanci manji od dvadeset aminokiselina nazivaju se oligopeptidi, dok lance veličine pedeset i više aminokiselina često nazivamo polipeptidima, iako taj izraz strogo tehnički opisuje bilo koji lanac više aminokiselina. Proteini, molekule koje igraju ključnu ulogu u funkcioniraju živih organizama te s drugim proteinima i makromolekulama grade kompleksnije strukture kao npr. DNA i RNA, građeni od jednog ili više polipeptida [3]. Peptidi spadaju u široku skupinu biopolimera, zajedno s nukleinskim kiselinama, oligosaharidima i polisaharidima.

2.1 Aminokiseline

Aminokiseline su organski spojevi koji sadrže amino skupinu ($-NH_2$), karboksilnu skupinu (-COOH) te bočni ogranak (R-skupinu) koji je jedinstven za svaku aminokiselinu [4]. Kako je već i ranije spomenuto, njihova glavna biološka uloga je izgradnja polipeptida iz kojih se grade proteini, iako vrijedi napomenuti da postoje i aminokiseline koje ne ulaze u sastav bjelančevina, a nazivaju se neproteinske aminokiseline, kao što su npr. ß-alanin, ornitin i citrulin. U prirodi je pronađeno više od 500 različitih aminokiselina, no samo njih 22 može se pronaći u genetskom kodu [5] (α-aminokiseline). Od navedene 22 aminokiseline, njih 20, poznate i kao "osnovne" ili prirodne aminokiseline, standardno se može pronaći u genetskom kodu, dok se 2 (selenocistein i pirolizin) pojavljuju kod specifičnih načina translacije kodona [6]. 20 osnovnih aminokiselina može se dalje podijeliti na esencijalne i neesencijalne. Esencijalnih aminokiselina je 9 te ih je u organizam potrebno unijeti iz vanjskih izvora, a su one: fenilalanin (F), valin (V), treonin (T), triptofan (W), metionin (M), leucin (L), izoleucin (I), lizin (K) i histidin (H). Preostalih 11 mogu se sintetizirati u ljudskom tijelu te ih kao takve nije nužno dobivati iz vanjskih izvora. One su: alanin (A), arginin (R), aspargin (N), asparaginska kiselina (D), cistein (C), glicin (G), glutamin (Q), glutaminska kiselina (E), prolin (P), serin (S) i tirozin (Y). Kemijske strukture, skupine kojima bočni ogranci pripadaju te troslovne i jednoslovne nazive navedenih aminokiselina prikazuje *Slika 2.1.*



Slika 2.1. Prikaz kemijskih struktura, troslovnih i jednoslovnih oznaka te bojom naznačenih skupina bočnih ogranaka 20 osnovnih aminokiselina, preuzeto iz [7]

2.2 Antimikrobni peptidi

Antimikrobni peptidi (eng. *antimicrobial peptides*, AMPs), zvani i obrambeni peptidi domaćina (eng. *host defence peptides*), dio su urođenog imunološkog odziva prisutnog u svim vrstama živih organizama [8]. Za iste je demonstrirana uspješnost u borbi protiv Gram-pozitivnih i Gram-negativnih bakterija [9], omotanih virusa i gljivica, te čak i kancerogenih stanica [10]. Za razliku od većine konvencionalnih antibiotika, za antimikrobne peptide čini se da često destabiliziraju stanične membrane, mogu formirati transmembranske kanale, a također mogu imati sposobnost jačanja imuniteta djelovanjem kao imunomodulatori. Antimikrobne peptide možemo nazivati i terapeutskim peptidima.

Antimikrobni peptidi široka su i raznolika skupina peptida, te se unutar sebe dodatno mogu klasificirati. Glavne klasifikacije antimikrobnih peptida [11] su:

- Prema strukturi: razlikujemo α-heliks, β-lanac, β-ukosnica (β-lanac povezan petljom) te izduženu peptidnu strukturu,
- Prema porijeklu: dijelimo ih na peptide životinjskog, biljnog, mikrobnog te sintetičkog porijekla,
- Prema mehanizmu djelovanja: podjela na peptide koji destabiliziraju staničnu membranu, one koji ciljaju unutarstanične strukture te imunomodulatorne peptide,
- Prema funkciji: antimikrobni peptidi uskog i širokog spektra djelovanja,
- Prema izvoru unutar tijela: i konačno razlikujemo konstitutivno izražene peptide (stalno prisutni u tijelu, pružajući kontinuiranu zaštitu) i inducibilne peptide (proizvode se kao odgovor imunološkog sustava na infekciju ili upalu).

U ovom radu koristit će skup podataka o antimikrobnim peptidima koji je već istraživan u srodnim studijama [12], [13] i [14], a prikupljen je iz baza podataka DRAMP [15] i Uniprot [16]. Baza DRAMP unutar sebe AMP-ove dodatno dijeli na antibakterijske, antiviralne, antifungalne, antiparazitske i antiprotozoalne, ali za potrebe ovog rada njih se sve zajedno promatralo kao jednu kategoriju, AMP.

3. Generativni model zasnovan na sprezi genetskog algoritma i strojnog učenja

Glavni fokus ovog rada bit će generativni model zasnovan na sprezi genetskog algoritma i strojnog učenja (u nastavku rada "GA + ML model") razvijen na Tehničkom fakultetu u Rijeci [17]. Osnovna ideja genetskog algoritma inspirirana je biološkim procesom evolucije i opstanka najsposobnijih jedinki te njihovog usavršavanja kroz generacije. Potpomognuto strojnim učenjem, čija je glavna primjena u ovom modelu bila treniranje i testiranje modela za predviđanje antimikrobnih peptida, ovakav pristup temeljen na mekom računarstvu *(eng. soft computing)* nudi adaptivan i široko primjenjiv model sposoban za daljnje istraživanje svojstva peptida i peptidnog prostora.

Meko računarstvo je skupina metoda u računarstvu koje oponašaju način na koji ljudski mozak rješava složene probleme, s naglaskom na fleksibilnost, prilagodljivost i toleranciju na netočnosti ili nesigurnosti [18]. Za razliku od tradicionalnog ili "tvrdog računarstva", (eng. *hard computing*), koje zahtijeva precizne i strogo definirane podatke i pravila, meko računarstvo koristi tehnike poput neuronskih mreža, genetskih algoritama, neizrazite logike (eng. *fuzzy logic*) i strojnog učenja kako bi omogućio rješavanje problema u kojima su podaci nejasni, promjenjivi ili nepotpuni. Glavna prednost mekog računarstva je njegova sposobnost da obrađuje neegzaktne ili približne podatke, što ga čini idealnim za aplikacije u stvarnom svijetu gdje je takva neizvjesnost uobičajena i očekivana.

3.1 Genetski algoritam

Općenito, genetski algoritam (GA) metoda je optimizacije inspirirana procesima evolucije i prirodnog odabira. Algoritam koristi principe evolucijske biologije kao što su odabir (selekcija), križanje (rekombinacija) te mutacija kako bi pronašao rješenja za kompleksne probleme [19]. Proces generalno započinje s populacijom slučajno generiranih mogućih rješenja, koje nazivamo jedinkama. Svaka jedinka predstavlja potencijalno rješenje i ocjenjuje se pomoću funkcije dobrote (eng. *fitness function*), koja mjeri kvalitetu rješenja u odnosu na cilj optimizacije. Najbolje jedinke, one s većom dobrotom, imaju veću vjerojatnost da budu odabrane za reprodukciju. Kroz procese križanja, njihove se karakteristike (geni) kombiniraju kako bi se stvorile nove jedinke. Tijekom procesa povremeno se

primjenjuje mutacija kako bi se u populaciju unijele nove varijacije i spriječila stagnacija na lokalnim optimumima. Ovaj ciklus ponavlja se kroz više generacija, dok algoritam ne pronađe optimalno ili zadovoljavajuće rješenje problema. Genetski algoritmi koriste se u različitim područjima gdje tradicionalne metode optimizacije mogu biti neučinkovite.

3.1.1 Objašnjenje rada genetskog algoritma

Genetski algoritam ovog modela napisan je u programskom jeziku Python uz pomoć paketa Numpy [20]. Rad samog genetskog algoritma može se podijeliti u 2 faze; inicijalizaciju te proces generiranja i optimizacije jedinki kroz generacije. Također treba pojasniti pojam jedinke u opsegu genetskog algoritma, gdje je jedinka definirana kao klasa koja se sastoji od dva atributa. Prvi atribut je niz znakova (eng. *string*) u kojem svaki znak predstavlja jednu od 20 osnovnih aminokiselina koristeći njihovu jednoslovnu oznaku. Nizu su određena minimalna i maksimalna duljina, u ovom slučaju to su 3 (najmanja duljina) i 50 (najveća duljina). Takav niz znakova, s definiranom strukturom te gornjom i donjom granicom duljine, može se interpretirati i kao niz aminokiselina koji zadovoljava ranije navedene uvijete peptida, pa će se u nastavku rada prema tom nizu referirati kao peptidnoj sekvenci jedinke ili samo peptidu jedinke. Drugi atribut klase je realni broj implementiran kao broj s pomičnom zarezom (eng. *floating-point number*, *float*) čija je vrijednost inicijalizirana na 0, dok će tijekom izvršavanja algoritma poprimiti povratnu vrijednost funkcije dobrote. Vrijednost funkcije dobrote nalazi se u rasponu [0, 1] i predstavlja razinu antimikrobnosti peptida jedinke. U nastavku rada prema ovom atributu referirat će se i kao dobrota jedinke. Više o funkciji dobrote, njenoj implementaciji i primjeni u potpoglavlju 3.1.2.

Pri inicijalizaciji GA potrebno je odrediti ulazne parametre:

- Veličina populacije: definira maksimalan broj jedinki u konačnoj populaciji svake generacije,
- Broj potomaka: definira broj potomaka koji će biti uveden u populaciju u svakoj generaciji,
- Broj generacija: definira broj generacija nakon kojeg, u nedostatku drugog razloga, genetski algoritam prestaje svoje izvođenje,
- Funkcija dobrote: algoritam koji na temelju određenih parametara predviđa antimikrobna svojstva peptida jedinke.

Nakon što je GA uspješno inicijaliziran može pobliže promotriti što se dešava njegovim pokretanjem.

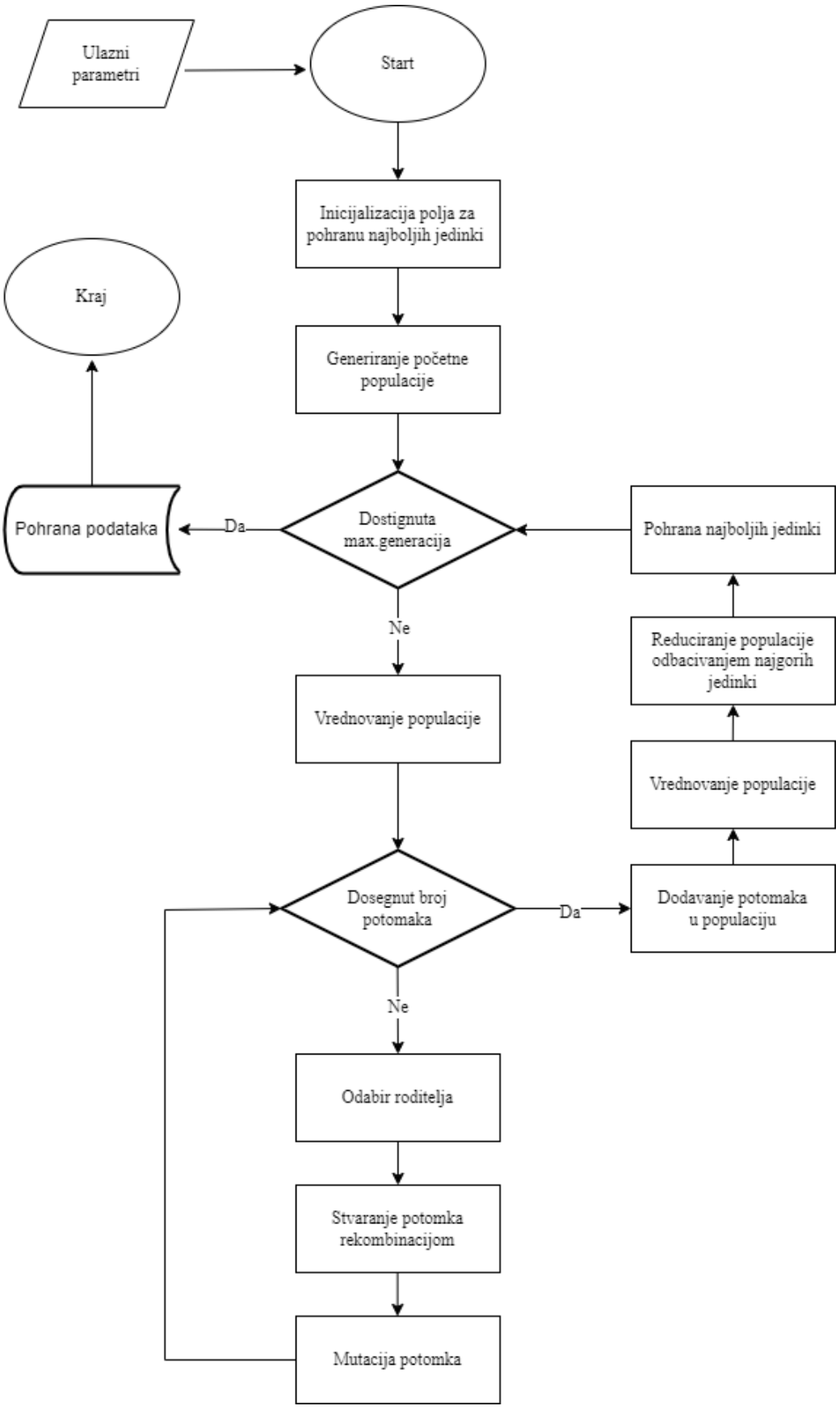
1. Prvi korak je inicijalizacija 2D polja kojem je svaki element niz u koji će se spremati najbolja rješenja, odnosno peptidi, iz svake generacije koje je ujedno i izlazna vrijednost genetskog algoritma.
2. Nakon toga generira se početna, slučajna populacija. Stvara se polje u koje će se spremati jedinke trenutne populacije. Svakom peptidu jedinke nasumično se određuje duljina, naravno unutar ranije utvrđenih raspona, te se svakom elementu novonastalog znakovnog niza nasumično dodjeljuje jedan od 20 mogućih znakova, svaki od kojih predstavlja jednu od osnovnih aminokiselina. Brojač generacija postavlja se na 1.
3. Nakon čega ulazimo u logičku petlju koja se ponavlja dok god brojač generacija ne preraste ranije određen maksimalan broj generacija.

Unutar petlje koja predstavlja napredovanje generacije genetskog algoritma događa se sljedeće:

1. Vrši se vrednovanje trenutke populacije. Svaki peptid koji je zapisan u jedinki šalje se kao ulazni podatak modelu strojnog učenja, čija se povratna vrijednost koristi kao vrijednost dobrote.
2. Zatim započinje proces generiranja potomaka iz trenutke populacije ovisan o ranije određenom broju potomaka. Proces započinje odabirom roditelja. Biraju se dva roditelja, svaki od kojih je produkt provjere i usporedbe dobrote jedinke 10 nasumično izabranih jedinki te odabirom one čija je vrijednost dobrote jedinke najveća, ne bi li se na taj način stimuliralo razmnožavanje boljih jedinki bez da se kompletno izuzme mogućnost razmnožavanja onih lošijih.
3. Nakon što su roditelji odabrani započinje proces rekombinacije. Nasumičnim odabirom dobiva se vrijednost p u rasponu [0, 1] te se početnih (p*100)% aminokiselina prvog roditelja kombinira s posljednjih ((1-p)*100)% aminokiselina drugog roditelja. Nad novonastalim peptidom se zatim provodi provjera; u slučaju da duljina peptida potomka nadilazi maksimalnu dozvoljenu duljinu posljednja aminokiselina u nizu se odbacuje, dok se u slučaju da je duljina peptida potomka manja od minimalne dozvoljene na kraj niza dodaje se nasumično odabrana aminokiselina. Isto se ponavlja dok uvjeti nisu zadovoljeni. Time osiguravamo da niti jedan novonastali peptid potomak ne krši u početku postavljena pravila duljina peptidnih sekvenci.

4. Prije nego li se novonastali potomci dodaju u populaciju, svaki peptid jedinke potomaka prolazi i proces mutacije. Ovisno, još jednom, o nasumično odabranoj vrijednosti i duljini peptida potomka proces mutacije rezultira na 4 moguća načina: jedna aminokiselina na nasumično određenoj poziciji biva zamijenjena s drugom (nasumično odabranom) aminokiselinom, aminokiselina na posljednjoj poziciji se odbacuje, na kraj niza dodaje se jedna nasumično odabrana aminokiselina te posljednja mogućnost u kojoj peptid ostaje nepromijenjen. Na ovaj način otvaramo mogućnost stvaranja novih peptida koji inače ne bi bili mogući samo rekombinacijom peptida slučajno generirane početne populacije.
5. Potomci se zatim dodaju u trenutnu populaciju te se populacija ponovno vrednuje. Populacija se potom sortira u ovisnosti o dobroti jedinke te se reducira, odbacujući pritom one jedinke s najmanjom dobrotom, na pri inicijalizaciji određenu veličinu populacije.
6. Konačno, trenutna populacija (najbolje jedinke te generacije) pohranjuje se kao element u početku stvorenog 2D polja te se brojač generacija inkrementira za 1.

Slika 3.1. donekle je pojednostavljen shematski prikazuje rad genetskog algoritma. Na slici se jasnije vidi tijek izvršavanja genetskog algoritma. Od postavljanja ulaznih parametara i početne inicijalizacije polja za pohranu najboljih jedinki i stvaranja početne slučajne populacije, potom ulaska u petlju koja predstavlja napredak generacija GA, procese vrednovanja populacije, postupak odabira roditelja te generiranja potomaka metodama rekombinacije i mutacije, dodavanje novonastalih potomaka u populaciju te njeno ponovno vrednovanje zatim reduciranje populacije odbacivanjem najgorih jedinki i pohrane preostalih (najboljih) jedinki te generacije, sve do konačnog dostizanja posljednje generacije genetskog algoritma i povrata polja u kojem su zapisane sve najbolje jedinke svih generacije i obustavljanjem izvršavanja rada genetskog algoritma.



Slika 3.1. Shematski prikaz rada genetskog algoritma

3.1.2 Funkcija dobrote

Funkcija dobrote je ključni koncept u evolucijskim algoritmima koji se koristi za ocjenjivanje kvalitete rješenja u okviru problema optimizacije, i to kroz singularan čimbenik kakvoće (eng. *figure of merit*) [21]. Funkcija dobrote mjeri "prilagodljivost" jedinke (rješenja) u populaciji, odražavajući koliko je to rješenje blizu optimalnog rješenja za zadani problem. U genetskim algoritmima, funkcija dobrote igra ulogu odabirnog kriterija, gdje se bolje prilagođena rješenja s većom dobrotom imaju veću šansu za reprodukciju i prijenos svojih svojstava na sljedeće generacije, čime se populacija vodi prema optimalnom rješenju kroz iterativne procese. U konkretnom modelu funkcija dobrote ima dva glavana djela. Prvi dio funkcije dobrote ispituje 9 svojstava peptida te vraća ukupno 28 značajki i njihovih vrijednosti. Te se značajke potom proslijede istreniranom ML modelu koji na temelju istih procjenjuje vjerojatnost antimikrobne aktivnosti jedinke, što je ujedno i povratna vrijednost funkcije. Svojstva koje prvi dio funkcije dobrote ispituje su:

- Sastav aminokiselina: ispituje kompoziciju aminokiselina u peptidu, daje relativnu učestalost svake aminokiseline unutar sekvence,
- Cruciani svojstva: ispituje tri specifična svojstva (PP1, PP2, PP3) prema Cruciani pristupu, koji povezuje kemijska svojstva aminokiselina s njihovom sposobnošću interakcije s otapalima. Ova svojstva pomažu u razumijevanju topljivosti, hidrofobnosti i ukupnog interakcijskog profila peptida,
- Indeks nestabilnosti: predviđa je li peptid stabilan u epruveti ili je sklon brzoj razgradnji,
- Bomanov indeks: ispituje sposobnost peptida da se veže za proteine,
- Indeks hidrofobnosti: ispituje hidrofobnost peptida koristeći Eisenberg skalu te je ključan faktor u presavijanju proteina i stabilnosti peptida,
- Hidrofobni moment: mjeri amfipatičnost peptida - kako su hidrofobne i hidrofilne regije raspoređene duž peptidnog lanca,
- Alifatski indeks: mjera relativnog volumena alifatskih bočnih lanaca (alanin, valin, izoleucin i leucin) u peptidu, ukazuje na termičku stabilnost peptida,
- Izoelektričnu točka(pI): pH vrijednosti pri kojoj peptid nema neto električni naboj,
- Neto naboj peptida: izračunava neto naboj peptida pri određenom pH (konkretno za 7,4 pH).

Značajke koja funkcija dobrote prosljeđuje istreniranom ML modelu te na temelju kojih se vrši izračun vrijednosti dobrote jedinke su: Tiny-Number, Tiny-Mole%, Small-Number, Small-Mole%, Aliphatic-Number, Aliphatic-Mole%, Aromatic-Number, Aromatic-Mole%, NonPolar-Number, NonPolar-Mole%, Polar-Number, Polar-Mole%, Charged-Number, Charged-Mole%, Basic-Number, Basic-Mole%, Acidic-Number, Acidic-Mole%, Cruciani_PP1, Cruciani_PP2, Cruciani_PP3, InstabilityIndex, Boman, Hydrophobicity-Eisenberg, HydrophobicMoment, AliphaticIndex, IsoelecticPoint-Lehninger i NetCharge-7.4-Lehninger. Razlozi za odabir tih svojstava i značajki opisani su u istraživačkom radu [22] te u ranije navedenim znanstvenim radovima [12] i [13].

Što se tehničke implementacije tiče, prvi dio funkcije dobrote, koji peptidnu sekvencu "pretvara" u skup svojstava napisan je kao skripta u programskom jeziku R te koristi javno dostupan paket Peptides [23] za provođenje ispitivanja ranije navedenih svojstava peptida, kao ulazni parametar prima peptidnu sekvencu, a kao izlazni vraća listu značajki i njihovih vrijednosti u JSON formatu, koji se po primitku raspakirava za daljnje korištenje od strane modela strojnog učenja. Kao računalni zapis peptida koriste se navedene značajke, te iste čine ulaz u ML model, a povratka vrijednost je vjerojatnost kojom se iskazuje sklonost antimikrobnoj aktivnosti. Što je vjerojatnost veća, to je veća i dobrota peptidne sekvence. Ta konačna vrijednost funkcije dobrote prosljeđuje GA se te se pohranjuje kao dobrota ispitane jedinke. U nastavku rada peptid jedinke smatrat će se antimikrobnim ako je dobrota te jedinke, odnosno povratna vrijednost funkcija dobrote, veća ili jednaka 0.95. Ovaj naizgled strog uvjet omogućit će različite perspektive sagledavanja na genetski algoritam, populacije generiranih peptida i zaključe koji će iz navedenih proizaći.

3.2 Strojno učenje

Strojno učenje (eng. *machine learning,* ML) je područje računalne znanosti koje se bavi razvojem algoritama i modela koji omogućuju računalima da uče iz podataka i donose odluke ili predviđanja bez potrebe eksplicitnog programiranja za svaki zadatak [24]. Modeli strojnog učenja analiziraju uzorke u podacima, prepoznaju pravila i koriste stečeno znanje za donošenje odluka ili predviđanje novih rezultata. Korištenjem algoritama strojnog učenja, računala mogu poboljšavati svoje performanse s više iskustva i podataka, čineći ih vrlo korisnima u širokom spektru primjena, od prepoznavanja slika i jezika, do predviđanja u raznim disciplinama medicine [25], ekonomije, prometa [26], poljoprivrede [27] itd.

3.2.1 Primjena strojnog učenja u radu generativnog modela

U generativnom modelu koristi se model strojnog učenja temeljen na povratnoj neuronskoj mreži (eng. *recurrent neural network*), koja koristi sekvencijalne značajke peptida kao format njihova računalnog zapisa. Takav model naziva se i *sequential properties* model te je razvijen na Tehničkom Fakultetu u Rijeci [12, 13] i korišten u ranije navedenom znanstvenom radu [17] uz ključnu razliku da je ovaj model treniran na setu podataka antimikrobnih peptida (AMP). Takav model pokazao se kao najučinkovitiji i pouzdaniji način hibridnog pristupa prepoznavanju i predviđanju antimikrobnih peptida, uz sve to istovremeno služeći i kao dio funkcije dobrote tijekom rada genetskog algoritma. *Tablica 3.1.* prikazuje rezultate treniranja ML modela na više drugačijih setova podataka različitim pristupima iz čega se set podataka AMP pokazuje kao onaj konzistentno najpouzdaniji.

*Tablica 3.1. Rezultati treniranja modela na više setova podataka različitim pristupima iz čega se set podataka **AMP** pokazuje kao onaj s konzistentno najvišim rezultatima, preuzeto iz [12]*

	F1 score	MCC	GM	recall	precision	AUC	data set
peptide properties	0.833	0.588	0.782	**0.870**[b]	0.800	0.867[b]	
one-hot	0.829	0.610	0.805	0.814	0.845[b]	0.869[b]	
embedding	0.806	0.547	0.772	0.804	0.808	0.837	AVPdb
sequential properties	**0.858**[b]	**0.671**[b]	**0.834**[b]	0.855[b]	**0.863**[b]	**0.882**[b]	
Friedman p-value	2.55e-5	3.56e-5	5.34e-5	3.1e-6	1.73e-6	3.52e-4	
peptide properties	0.188	0.256	0.293	0.117	0.643[b,c]	0.888	
one-hot	0.396[b]	**0.440**[b]	0.524[b]	0.282[b]	**0.743**[b]	0.935[b]	
embedding	0.168	0.205	0.237	0.108	0.427[b,c]	0.918	AVDRAMP
sequential properties	**0.413**[b]	**0.440**[b]	**0.550**[b]	**0.322**[b]	0.662[b]	**0.947**[b]	
Friedman p-value	1.5e-4	1.09e-3	6.94e-5	1.59e-5	4.05e-2	5.12e-8	
peptide properties	0.838	0.547	0.749	0.885[b]	0.797	0.849	
one-hot	0.847[b]	0.577	0.766	0.887[b]	0.813	0.855	
embedding	0.827	0.502	0.711	**0.889**[b]	0.775	0.830	AVMerged
sequential properties	**0.869**[b]	**0.652**[b]	**0.817**[b]	0.885[b]	**0.856**[b]	**0.885**[b]	
Friedman p-value	9.63e-6	2.7e-7	3.62e-7	0.82	5.59e-8	8.48e-8	
peptide properties	0.828	0.723	0.855	0.801	0.856	0.942	
one-hot	0.891	0.822	0.909	0.885[b]	0.898	0.973	
embedding	0.866	0.782	0.887	0.849	0.883	0.959	AMP
sequential properties	**0.901**[b]	**0.839**[b]	**0.917**[b]	**0.890**[b]	**0.913**[b]	**0.977**[b]	
Friedman p-value	2.64e-11	2.64e-11	3.87e-11	1.58e-9	2.66e-9	1.38e-11	

S obzirnom da je cilj ovog rada istraživanje i prilagodba generativnog modela za potrebu generiranja terapeutskih peptidnih sekvenci te ispitivanje prednosti i nedostataka kod ponovljenog postupka generiranja peptida i analiza konvergencije genetskog algoritma korištenjem gotovih ML modela, u nastavku rada veća pažnja će se posvetiti analizi rezultata dobivenih radom genetskog algoritma.

4. Studija slučaja

Kako bi dobivene podatke mogli smisleno analizirati bilo je potrebno pomno odabrati ulazne parametre genetskog algoritma takve da bi generativni model ostvario željene rezultate za što je moguće manje utrošenog vremena, ali i kako bi se ostvarilo bolje razumijevanje rada modela u uvjetima kada model ili ne dostigne većinski antimikrobnu populaciju ili se duži broj generacija zadržava u većinski antimikrobnim populacijama. U tu svrhu podaci dobiveni radom generativnog modela podijeljeni su u skupove generiranih peptida, neki od koji se unutar sebe još dodatno dijele u podskupove. Svaka instanca rada generativnog modela unutar generiranog skupa u nastavku rada nazivat će se iteracija generativnog modela, ili jednostavnije samo iteracija. Nomenklatura generiranih podataka pratit pravilo u kojem prvi broj naznačuje generirani skup, a drugi iteraciju po principu [*skup*]-[*iteracija*], npr. *2-1* označuje prvu iteraciju generativnog modela u drugom generiranom skupu podataka. Iteracije po završetku izvršavanja najbolje jedinke svojih populacija kroz generacije pohranjuju u binarnu numpy datoteku pod nazivom "jedinke[*skup*]-[*iteracija*].npy" za daljnje korištenje u obradi podataka i analizi rezultata.

Za svaku iteraciju pratio se sastav aminokiselina (učestalost pojave aminokiselina u peptidnim sekvencama jedinki populacije) u ključnim generacijama, poglavito u prvoj i posljednjoj generaciji te nekoliko puta između ovisno o broju generacija kako bi se dobio uvid u to kako se udio aminokiselina mijenja i koje aminokiseline bi mogle doprinositi antimikrobnosti peptida. Nadalje, za svaku se iteraciju pratila i progresija antimikrobnosti generacija, odnosno udio antimikrobnih jedinki naspram ukupne populacije. Sastav aminokiselina također se pratio i na razini skupova/podskupova podataka uzimajući u obzir sastave svih iteracija skupa/podskupa te računanjem njihovih prosjeka ne bi li se ublažio ili čak kompletno otklonio djelomično nasumičan rad genetskog algoritma. Pratila se i duljina peptidnih sekvenci s ciljem provjere konvergira li duljina sekvenci ka određenoj, optimalnoj duljini. Za početne i konačne populacije svake iteracije dodatno je bila promatrana i sličnost među peptidima koristeći Needleman–Wunsch algoritam [28] ne bi li se ustvrdilo konvergiraju li peptidne sekvence unutar populacije, ali i kako bi se provjerilo konvergiraju li konačne populacije različitih iteracija ka istim ili sličnim rješenjima. Više o Needleman–Wunsch algoritmu u potpoglavlju 4.2

4.1 Generirani skupovi podataka

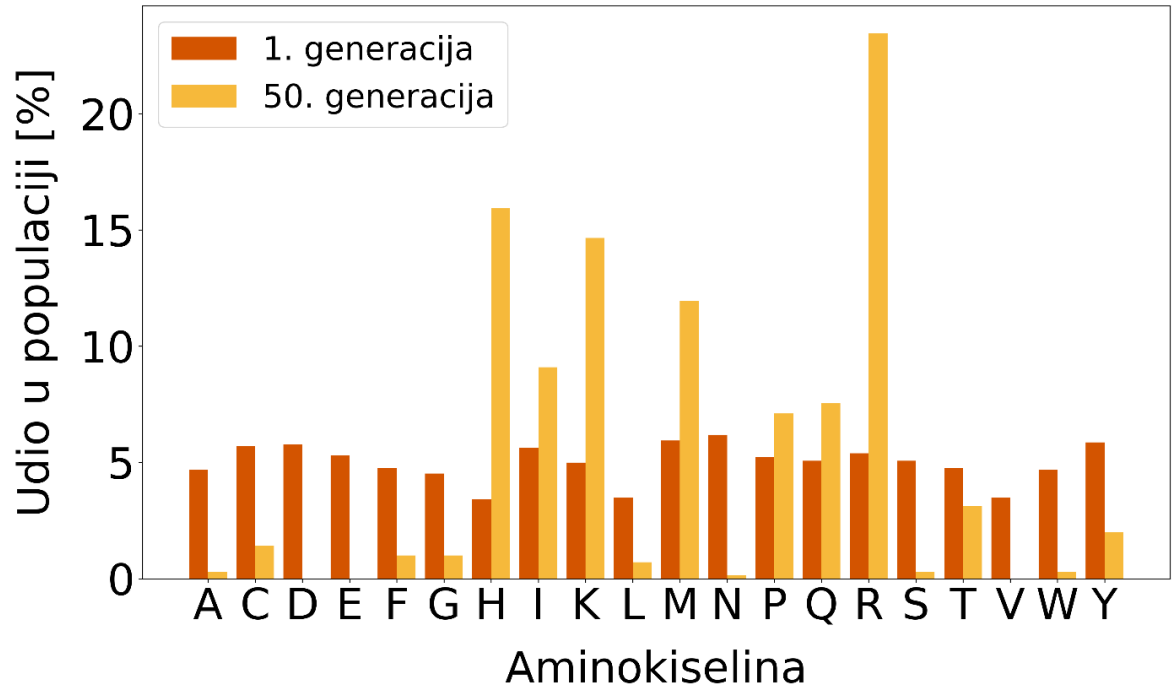
Kako ulazne parametre prvog generiranog skupa podataka u ovom koraku nije još bilo s čime usporediti, isti su odabrani slobodnom procjenom, s ciljem da pritom odabrani parametri mogu ukazati na potencijalne nedostatke modela na kojima bi se u idućim generiranim skupovima dalo poraditi. Stoga su parametri odabrani za prvi generirani skup peptida sljedeći:

- *Skup 1*
    - Broj iteracija: 10
    - Veličina populacije: 20
    - Broj potomaka: 2
    - Broj generacija: 50

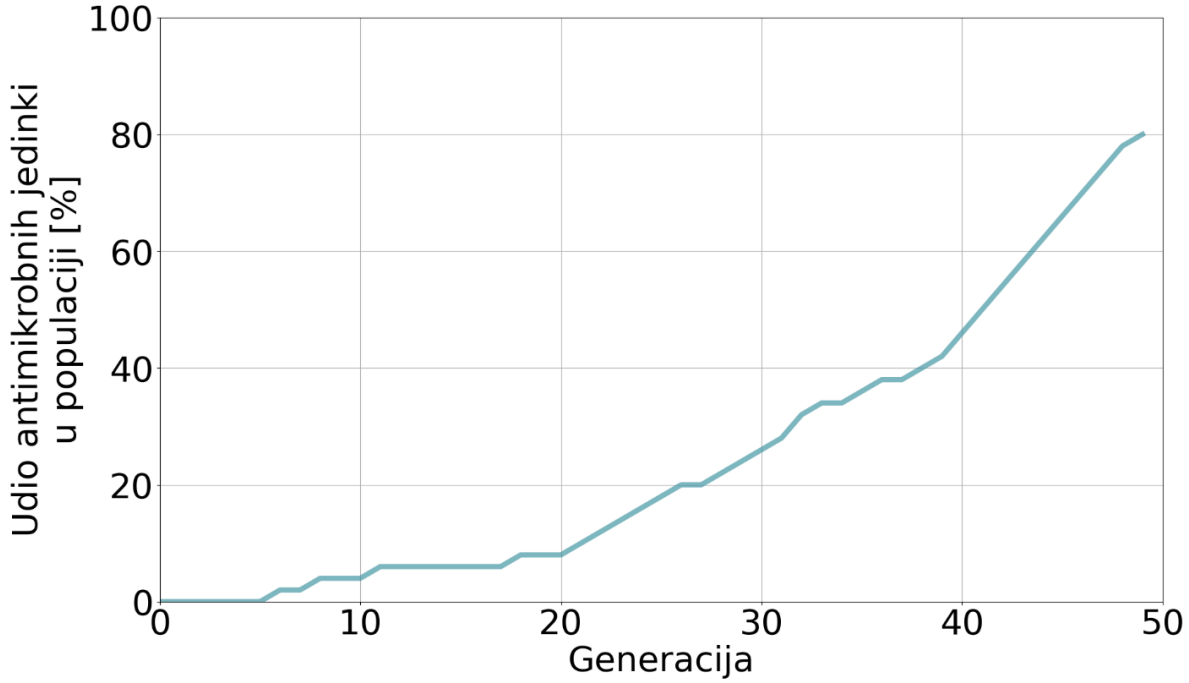
Iako se u kao najistaknutije aminokiseline u konačnim populacijama pojavljuju lizin (K) i arginin (R), s prosječnim udjelima od 14.66% i 13.83%, za kojima slijede histidin (H), glicin (G) i izoleucin (I) s prosječnim udjelima od ~7%, njihovi udjeli uvelike variraju između iteracija. Najmanje istaknute aminokiseline, asparaginska kiselina (D) i glutaminska kiselina (E) pojavljuju se s prosječnim udjelom od ~0.1% (prosječni udjeli svih aminokiselina vidljivi su na *Slici 5.1.*). Tu se zamjećuje razlika između najčešćih aminokiselina u peptidima za trening, koje su alanin (A), cistein (C), glicin (G), lizin (K) i leucin (L), i to ponajviše u promjeni udjela aminokiseline arginin (R). Vrijedi napomenuti i da niti jedna iteracija ovog generiranog skupa ne postiže potpuno antimikrobnu populaciju, dio iteracija, njih 2/10, ne uspijeva producirati niti jedan antimikroban peptid, a većina, njih 7/10, u konačnoj populaciji ima udio antimikrobnih peptida manji od 50%.

Slika 4.1. grafički prikazuje sastav aminokiselina u početnoj i konačnoj generaciji najuspješnije iteracije prvog generiranog skupa, iteracije *1-7*. Na-x-osi jednoslovnom se oznakom prikazuju aminokiseline, a na y-osi njihov udio u ukupnoj populaciji. *Slika 4.2*. prikazuje progresiju udjela antimikrobnih jedinki u populaciji kroz generacije iste iteracije (*1-7*), gdje se na x-osi prikazuje generacija, a na y-osi udio antimikrobnih peptida te generacije.

Sve ovo razlozi su za osmišljanje i stvaranje novih, većih generiranih skupova podataka. No, prije nego li ispitamo kako model radi s većim brojem iteracija, jedinki i generacija, potrebno je i proučiti kakav utjecaj povećanje broja potomaka ima na rezultate. U tu svrhu stvoren je idući generirani skup peptida.



Slika 4.1. Sastav aminokiselina u početnoj i konačnoj generaciji najuspješnije iteracije prvog generiranog skupa, iteraciji 1-7



Slika 4.2. Progresija udjela antimikrobnih jedinki u populaciji kroz generacije najuspješnije iteracije prvog generiranog skupa, iteraciji 1-7

- *Skup 2*
    - Broj iteracija: 10
    - Veličina populacije: 20
    - Broj potomaka: 10
    - Broj generacija: 50

Razlike u odnosu na prvi generirani skup su primjetne (*Slike 5.1 i 5.2*). Sastav aminokiselina razlikuje se po udjelima i poretku najistaknutijih aminokiselina (arginin (R) preuzima vodeće mjesto prestižući lizin (K) po prosječnom udjelu s gotovo dvostruko većim udjelom, metionin (M) se iskazuje kao treća najučestalija aminokiselina) dok se broj iteracija koje postižu potpuno antimikrobnu populaciju znatno povećava, toliko da samo jedna od deset iteracija ne dostiže taj cilj. To je također i iteracija koja ne uspijeva producirati niti jedan antimikroban peptid. Iako definitivno unaprjeđenje u odnosu na prijašnji, ovaj generirani skup naslijedio je i nedostatke prvog skupa, poglavito malen broj iteracija, jedinki i generacija. Sljedeći generirani skup osvrnut će se na te nedostatke.

- Skup 3
    - Broj iteracija: 20 (4 * 5 iteracija svakog podskupa)
    - Veličina populacije: 100
    - Broj potomaka: {5, 10, 20, 30} (ovisno o podskupu)
    - Broj generacija: 100

U ovom generiranom skupu podataka po prvi puta koriste se i podskupovi, i to iz dosad nespomenutog razloga, a to je vrijeme potrebno za izvršavanje (engl. *execution time*) generativnog algoritma. Dosadašnje iteracije GA svoje bi izvršavanje dovršile u svega nekoliko minuta, dok je vrijeme izvršavanja iteracija s ovoliko povećanom veličinom populacije, broja potomaka i broja generacija preraslo u sate. Svaki od ovih podskupova u idealnom svijetu mogao bi biti zaseban skup, no radi ograničenosti vremenom i procesorskom snagom smanjen je njihov obujam te su grupirani u jedan veći generirani skup podataka.

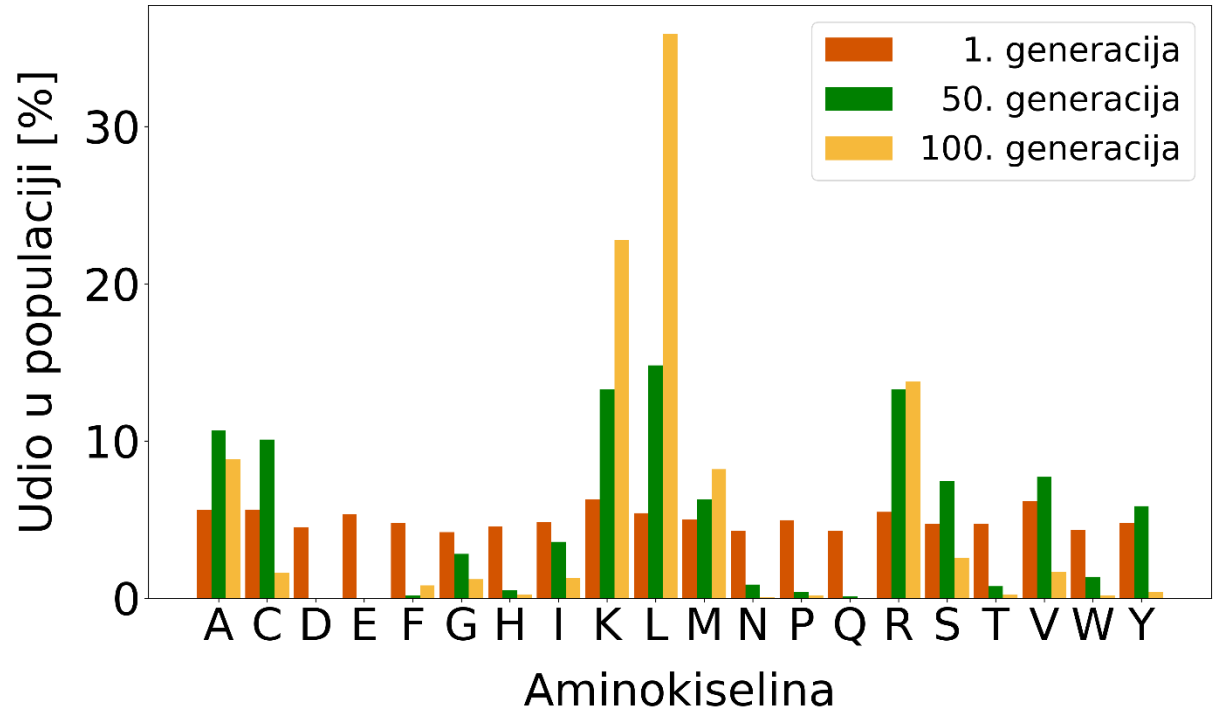
Kada je riječ o samim karakteristikama generiranog skupa, varijacije sastava aminokiselina između iteracija još uvijek postoje, ali su manje nego kod prijašnjih skupova generiranih peptida (*Slika 5.3.*). Varijacije sastava aminokiselina prisutne su i među podskupovima, ali to se može pridonijeti relativno maloj veličini svakog od podskupova. U globalu, generativni model jasnije konvergira ka peptidnim sekvencama pretežno građenim od aminokiselina arginin (R) i lizin (K), te sve iteracije dostižu

potpunu antimikrobnu populaciju u broju iteracija ovisnom u broju potomaka. Podskupovi s većim brojem potomaka u prosjeku potpunu antimikrobnu populaciju dostižu u manjem broju iteracija nego oni s manjim brojem potomaka. Primjer grafičkog prikaza sastava aminokiselina u početnoj, pedesetoj (srednjoj) i konačnoj generaciji te progresije antimikrobnosti jedinki jedne od iteracija ovog generiranog skupa prikazani su na *Slikama 4.3. i 4.4.* Što je bilo zanimljivo za zamijetiti u ovom skupu podataka (*Skup 3*) jest da se sastav aminokiselina mijenja čak i nakon dostizanja potpune antimikrobne populacije. Kao pokušaj savladavanja i ovog novonastalog izazova stvoren je još jedan generirani skup podataka gotovo identičan trenutnom uz jednu ključnu razliku. Broj generacija za sljedeći generirani skup je udvostručen.

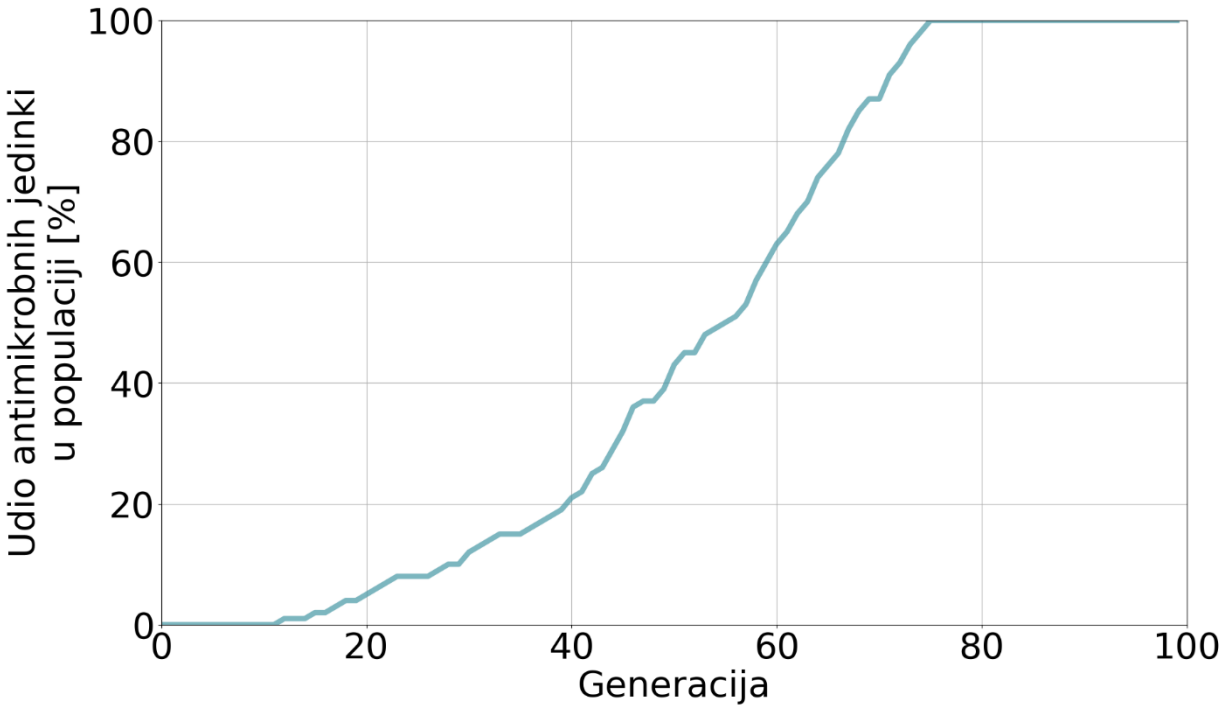
- *Skup 4*
    - Broj iteracija: 20 (4 * 5 iteracija svakog podskupa)
    - Veličina populacije: 100
    - Broj potomaka: {5, 10, 20, 30} (ovisno o podskupu)
    - Broj generacija: 200

Karakteristike ovog generiranog skupa u velikoj mjeri nalikuju onima prijašnjeg. Varijacije sastava aminokiselina još uvijek su prisutne (*Slika 5.4.*), kao i varijacije u sastavima aminokiselina među podskupovima što se ponovno može pridonijeti maloj veličini svakog od podskupova. Prosječni sastavi aminokiselina približno su jednaki onima iz prethodno navedenog *Skupa 3*. Primjer grafičkog prikaza sastava aminokiselina u početnoj, pedesetoj, stotoj i konačnoj generaciji te progresije antimikrobnosti jedinki jedne od iteracija ovog generiranog skupa prikazani su na *Slikama 4.5. i 4.6.* Iako je ovom promjenom znatno umanjena promjena prosječnog sastava aminokiselina između stote i dvjestote (konačne) generacije generiranog skupa (*Slika 5.6.*), razlike u pojedinim iteracijama preostaju. Ovime se otvara mogućnost teoretskog broja iteracija nakon kojeg genetski algoritam stagnira, no kako ni s brojem generacija postavljenim na 500 taj cilj nije dostignut (*Slika 4.7.*) takav maksimum u opsegu ovog rada ostaje nepoznat.

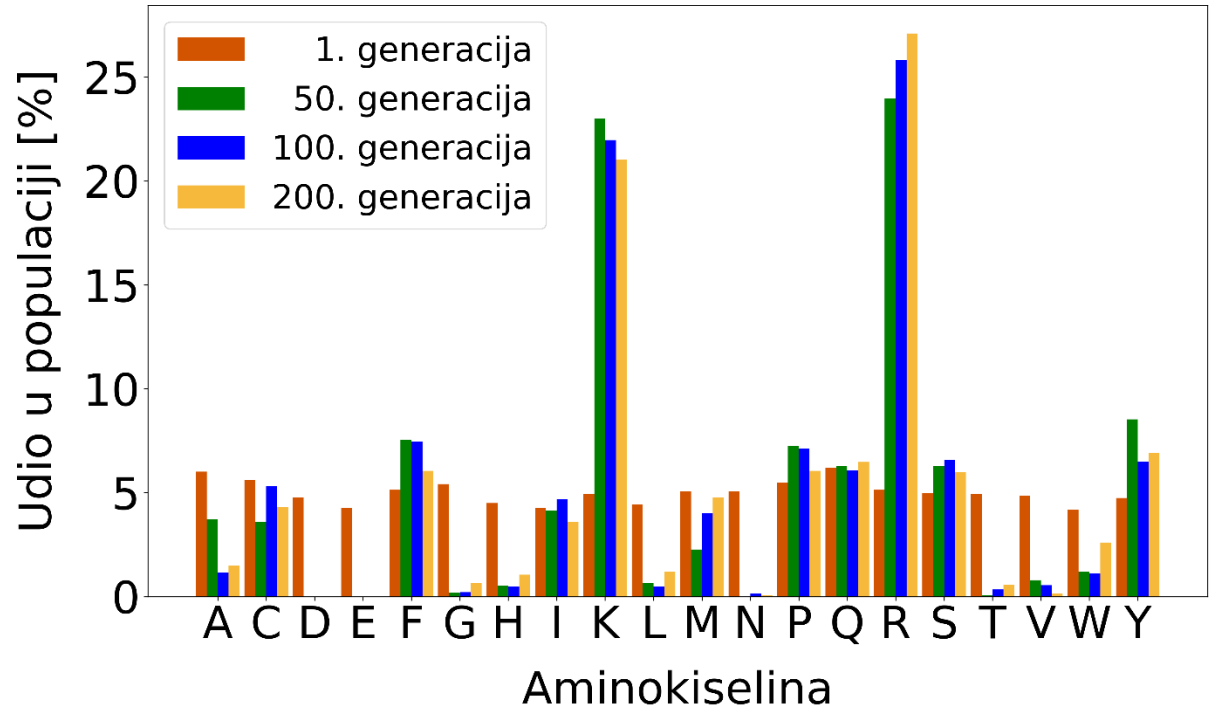
Preostaje pitanje što ako generativni model zaustavimo točno u trenutku kada prvi put dostigne potpuno antimikrobnu populaciju. Za to je potrebno modificirati genetski algoritam na način da nakon svake generacije prebrojava vlastite jedinke, sumirajući pritom one antimikrobne te uspoređujući tu sumu s veličinom populacije. U slučaju u kojem su te dvije vrijednosti jednake genetski algoritam prestaje sa svojim radom. S tom izmjenom na umu stvoren je sljedeći generirani skup podataka.



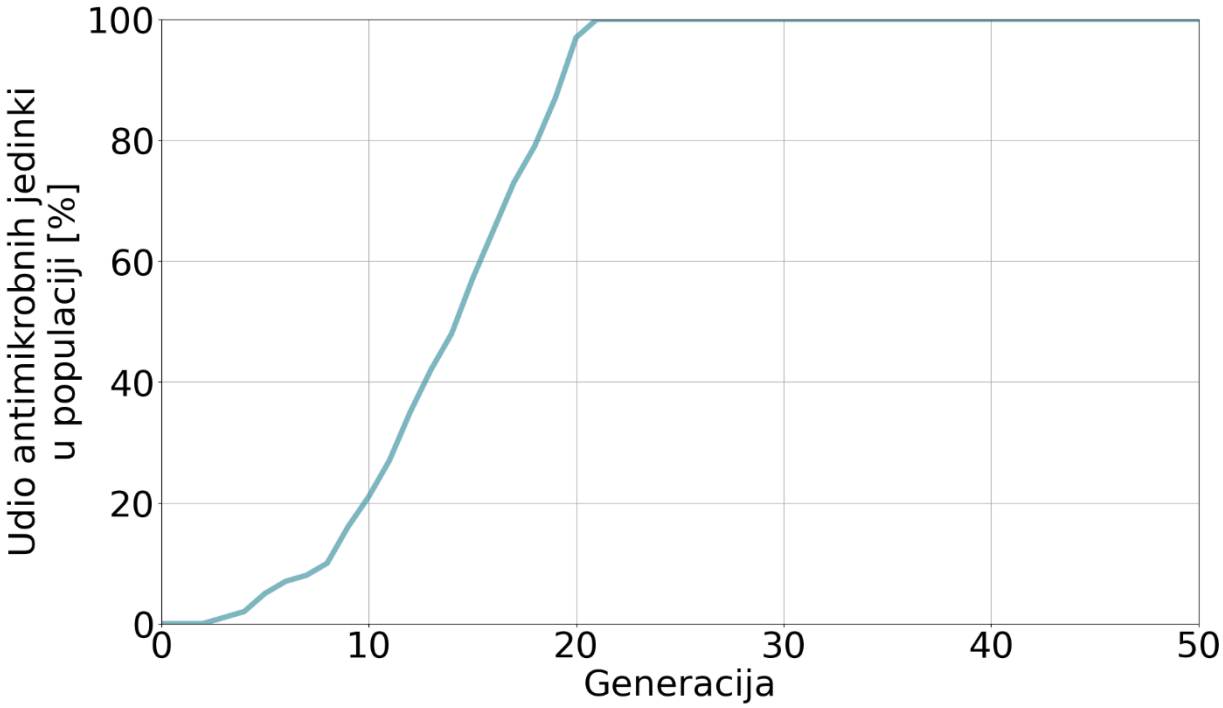
Slika 4.3. Sastav aminokiselina u početnoj, pedesetoj i konačnoj generaciji, iteracija 3-4



Slika 4.4. Progresija udjela antimikrobnih jedinki u populaciji kroz generacije, iteracija 3-4



Slika 4.5. Sastav aminokiselina u početnoj, pedesetoj, stotoj i konačnoj generaciji, iteracija 4-7



Slika 4.6. Progresija udjela antimikrobnih jedinki u populaciji kroz generacije, iteracija 4-7

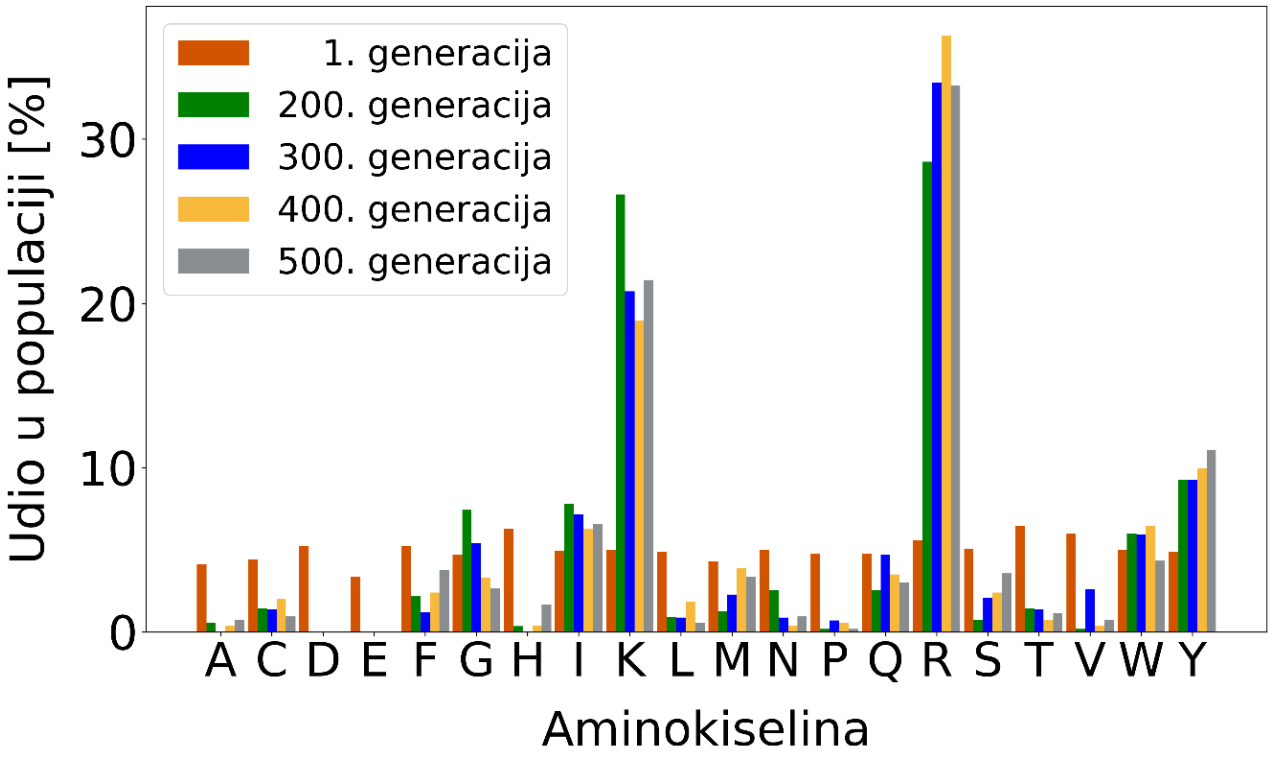
- *Skup 5*
    - Broj iteracija: 30 (3 * 10 iteracija svakog podskupa)
    - Veličina populacije: 50
    - Broj potomaka: {10, 20, 30} (ovisno o podskupu)
    - Broj generacija: 50
    - Modificirani genetski algoritam

Varijacije sastava aminokiselina, kako konačnih populacija tako i njihovih prosjeka kroz podskupove naglašenije su nego li u prijašnja dva skupa (*Slika 5.5.*), no gledajući cjelokupni generirani skup one aminokiseline najizraženije (arginin (R) i lizin (K)) i najmanje izražene u prijašnjim to bivaju i u ovom. Broj generacija svake iteracije, u ovom generiranom skupu različit od iteracije do iteracije radi modifikacije genetskog algoritma, prati ranije uspostavljeni trend u kojem iteracije s većim broj potomaka u prosjeku prije dostižu potpuno antimikrobnu populaciju nego one s manjim brojem potomaka, pa time i genetski algoritam istih prije završava sa svojim radom. Niti jedna iteracija ne dostiže predefinirani maksimalan broj generacija (50) iako jedna dolazi vrlo blizu, čak do 48. generacije.

- *Skup 6*
    - Broj iteracija: 5
    - Veličina populacije: 50
    - Broj potomaka: 5
    - Broj generacija: 500

Ovaj generirani skup podataka osmišljen je kao provjera stagnira li genetski algoritam nakon većeg broja generacija te radi izrazito malog broja iteracija, njih samo 5, prouzročenog sve većim vremenom potrebnim za izvršavanje svake neće biti uziman u obzir tijekom analize podataka, osim ako suprotno nije eksplicitno navedeno. Varijacije u sastavima aminokiselina u kasnijim generacijama još uvijek postoje iz čega se da zaključiti da algoritam ne stagnira čak ni nakon uvećanja broja generacija.

Primjer grafičkog prikaza sastava aminokiselina u početnoj, stotoj, dvjestotoj, tristotoj, četiristotoj i petstotoj (konačnoj) generaciji jedne od iteracija ovog generiranog skupa prikazan je na *Slici 4.7.* Na slici se vidi da iako se već u dvjestotoj generaciji aminokiseline arginin(R) i lizin(K) iskazuju kao one s najvećim udjelom, taj udio varira u budućim generacijama, što vrijedi i za ostale manje zastupljene aminokiseline.



Slika 4.7. Sastav aminokiselina u početnoj, stotoj, dvjestotoj, tristotoj, četiristotoj i petstotoj(konačnoj) generaciji, iteracija 6-1

4.2 Usporedba peptidnih sekvenci

Za potrebe usporedbe peptidni sekvenci u ovom radu koristit će se Needleman-Wunsch algoritam. Needleman-Wunsch algoritam je klasična metoda za globalno poravnanje dviju sekvenci, najčešće korištena u bioinformatici za usporedbu DNA, RNA ili proteinskih sekvenci. Algoritam je razvijen 1970. godine od strane Saula B. Needleman-a i Christiana D. Wunsch-a te se temelji se na dinamičkom programiranju, što omogućuje optimalno poravnanje cijele duljine dviju sekvenci. Needleman-Wunsch algoritam koristi se za određivanje sličnosti između dviju sekvenci kroz cijelu njihovu duljinu. To je vrlo korisno kada se želi pronaći optimalno poravnanje cijelih genoma, proteina ili drugih bioloških sekvenci. Algoritam je temelj za mnoge druge algoritme u bioinformatici, uključujući algoritme za lokalno poravnanje, kao što je Smith-Waterman algoritam. Prednosti Needleman-Wunsch algoritma su to što garantira globalno optimalno poravnanje, jednostavan je za implementaciju, te se može prilagoditi različitim supstitucijskim matricama, dok su njegovi nedostaci činjenica da je relativno spor za vrlo duge sekvence zbog svoje vremenske i prostorne složenosti O(n*m), gdje su n i m duljine dviju sekvenci.

Što se tehničke implementacije tiče korišten je algoritam iz javno dostupnog Python paketa scikit-bio [29], dok je za određivanje bodova supstitucijske matrice korištena je BLOSUM50 matrica [30]. Povratna vrijednost algoritma je u rasponu [0, 1] gdje 1 naznačuje da su dvije uspoređene peptidne sekvence identične, a opadanjem sličnosti među sekvencama smanjuje se i povratna vrijednost algoritma. U opsegu ovog rada Needleman-Wunsch algoritam bit će korišten kao sredstvo za usporedbu te procjenu sličnosti dviju peptidnih sekvenci, ali i procjenu sličnosti konačnih populacija peptida ne bi li se time pokušalo ustvrditi ne samo konvergiraju li rješenja pojedinih iteracija ka određenim peptidnim sekvencama, nego i postoji li sličnost među najboljim rješenjima različitih iteracija i skupova generiranih peptida.

5. Analiza rezultata

5. 1 Sastav aminokiselina

O sastavu aminokiselina već je ponešto rečeno u prethodnom poglavlju, dok će u ovom isto biti detaljnije analizirano. Dok sve iteracije genetskog algoritma svoj rad započinju s približno jednakom distribucijom osnovnih aminokiselina (svaka čini ~5% ukupne početne populacije), njihov udio značajno se mijenja u konačnim populacijama. Aminokiseline arginin (R) i lizin (K) se kroz sve skupove generiranih peptida pokazuju kao značajno najzastupljenije (s prosječnom zastupljenošću 21.64% i 16.52%), sugerirajući njihov doprinos antimikrobnim svojstvima peptida, za kojima slijede metionin (M), histidin (H), alanin (A) i glicin (G) s prosječnom zastupljenošću većom od 5%. Suprotno tome, aminokiseline asparaginska kiselina (D), glutaminska kiselina (E) gotovo i da nisu prisutne u generiranim peptidima, za kojima s nešto većim ali još uvijek ispod prosječnim udjelima slijede aspargin (N), serin (S) i glutamin (Q). Što se isto da zamijetiti jest da u skupu generiranih peptida u kojemu se ne postiže potpuna antimikrobna populacija (*Skup 1*) i u onom čije se izvršavanje zaustavlja u trenutku kada se prvi put dostigne potpuno antimikrobna populacija (*Skup 5*) opažamo podjednak udio aminokiselina arginin (R) i lizin (K) (13.83% i 14.66% u *Skupu 1*; 17.02% i 16.79% u *Skupu 5*), dok u onim generiranim skupovima čije se izvršavanje nastavlja i nakon postizanja potpune antimikrobne populacije (*Skupovi 2, 3 i 4*) udio arginina (R) u odnosu na lizin (K) raste (24.12% i 14.34% u *Skupu 2*; 26.76% i 17.35% u *Skupu 3*; 26.47% i 19.44% u *Skupu 4*). Iz toga se može zaključiti da dok i arginin (R) i lizin (K) u sličnoj mjeri doprinose ranom napretku antimikrobnosti jedinki, u konačnom dijelu optimizacije istih arginin se pokazuje kao bitniji čimbenik.

Varijabilnost sastava aminokiselina mjerila se kao prosječno kvaratno odstupanje podataka od srednje vrijednosti (standardna devijacija). Očekivano, varijabilnost podatka opada povećanjem broja iteracija, veličine populacije, broja potomaka i broja generacija danog generiranog skupa iz razloga što se povećanjem navedenih parametara generativnom modelu dalje više vremena (generacija) u kojem može konvergirati ka optimalnom rješenju. Jasno je onda da u generiranim skupovima peptida za koje to je slučaj (*Skupovi 3 i 4*) bolje uočavamo i očekivane raspone u kojima se, u prosjeku, svaka aminokiselina pojavljuje u antimikrobnim peptidima.

Nadalje, za neke aminokiseline, poglavito izoleucin (I), leucin (L) i valin (V), zamjećujemo visoku stopu odstupanja unatoč relativno malom prosječnom udjelu, što bi ukazalo na to da, iako se rijetko pojavljuju kao dijelovi antimikrobnih peptida, kada se pojave to čine u visokom udjelu. Daljnje istraživanje potrebno je da se ustvrdi postoji li veza između pojavljivanja određenih, u prosjeku manje zastupljenih aminokiselina u ovisnosti u drugim aminokiselinama u antimikrobnim peptidnim sekvencama.

Prosječni udio svake od aminokiselina u konačnim generacijama svih iteracija pojedinog generiranog skupa podatka prikazuje *Tablica 5.1.*, u kojoj svaki redak opisuje jednu od aminokiselina, a svaki stupac jedan od generiranih skupova peptida dobivenih radom genetskog algoritma. Brojevi predstavljaju postotne vrijednosti. Udio zastupljenosti potpomognut je i intenzitetom obojenja tablice radi veće vizualne jasnoće.

Tablica 5.2. na sličan način prikazuje standardnu devijaciju svake od aminokiselina u konačnim generacijama svih iteracija pojedinog generiranog skupa podatka. Svaki redak i u ovoj tablici opisuje jednu od aminokiselina, a svaki stupac jedan od generiranih skupova peptida dobivenih radom genetskog algoritma. Brojevi predstavljaju postotne vrijednosti. Razina odstupanja potpomognuta je i intenzitetom obojenja tablice radi veće vizualne jasnoće.

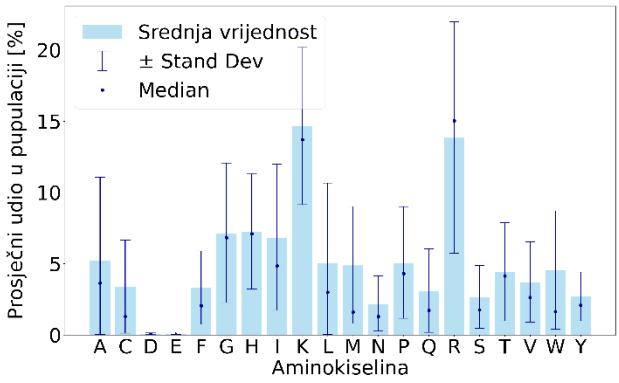
Slika 5.1. - *Slika 5.5.* daje grafički prikaz navedenih mjera. Na x-osi nalaze se aminokiseline, a na y-osi prosječni udio u konačnoj populaciji, gdje svaki stupac predstavlja srednju vrijednost prosječnog udjela svake od aminokiselina u konačnim populacijama svih iteracija, točkom je naznačen medijan vrijednost istog, a stupcima pogreške (engl. *error bars*) standardna devijacija. Svaka slika svojim rednim brojem odgovara skupu generiranih peptida na koji se odnosi.

Tablica 5.1. Prosječni udio aminokiselina u konačnim generacijama svih iteracija kroz skupove

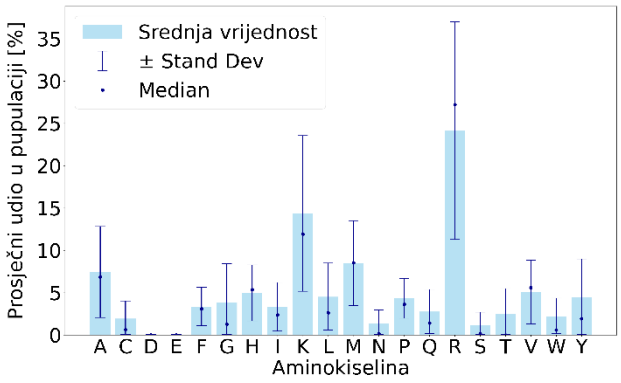
	Skup				
	1	2	3	4	5
A	5.24	7.42	3.38	3.71	5.5
C	3.37	1.93	3.45	3.06	3.88
D	0.08	0	0	0	0.04
E	0.08	0	0	0	0.02
F	3.3	3.35	4.35	5.8	4.52
G	7.14	3.85	4.78	3.22	6.18
H	7.24	4.95	5.02	2.98	6.24
I	6.84	3.31	3.52	4.57	5.23
K	14.66	14.34	17.35	19.44	16.79
L	5.05	4.52	6.1	2.91	4
M	4.88	8.46	5	6.68	5.2
N	2.18	1.34	0.95	0.79	2.27
P	5.06	4.31	1.66	1.9	2.66
Q	3.09	2.75	3.15	2.53	1.8
R	13.83	24.12	26.76	26.47	17.02
S	2.64	1.12	2.12	2.46	3.39
T	4.41	2.45	2.84	2.46	3.73
V	3.69	5.05	2.17	3.4	3.73
W	4.54	2.23	2.83	4.08	4.16
Y	2.68	4.49	4.54	3.55	3.64

Tablica 5.2. Standardna devijacija od srednje vrijednosti aminokiselina u konačnim generacijama svih iteracija kroz skupove

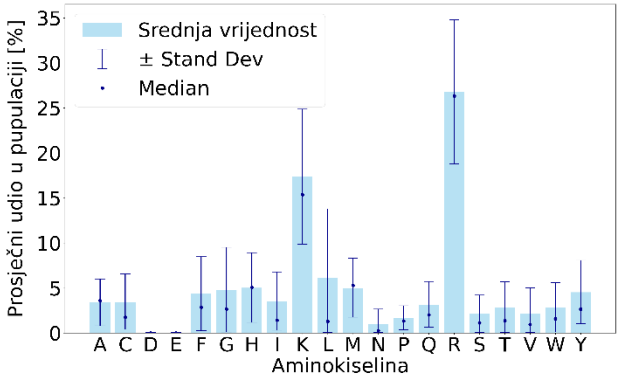
Skup				
1	2	3	4	5
7.51	6.99	3.01	5.48	4.99
4.25	2.7	3.55	3.34	3.48
0.09	0	0	0	0.17
0.15	0	0	0	0.05
3.35	2.98	4.75	4.54	4.65
6.35	5.9	5.43	4.03	5.45
5.25	4.3	4.48	3.07	5.05
6.67	3.74	3.75	6.47	6.37
7.17	11.92	8.64	9.09	6
7.24	5.14	8.83	2.78	3.47
5.33	6.52	3.78	5.11	4.85
2.53	2.09	1.95	1.42	3.44
5.09	3.07	1.58	2.26	3.14
3.81	3.39	2.94	2.51	2
10.51	16.61	9.21	10.54	9.26
2.88	2.03	2.4	3.75	3.42
4.47	3.92	3.26	2.66	3.89
3.67	4.91	3.26	5.2	4.1
5.39	2.75	3.17	4.01	4.3
2.24	5.81	4.06	2.63	3.36



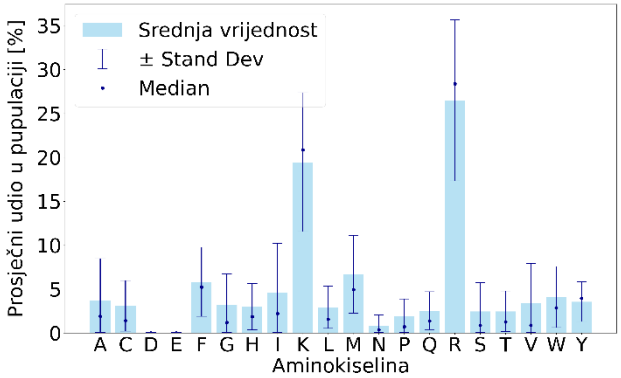
Slika 5.1. Prosječni sastav aminokiselina konačnih generacija - Skup 1



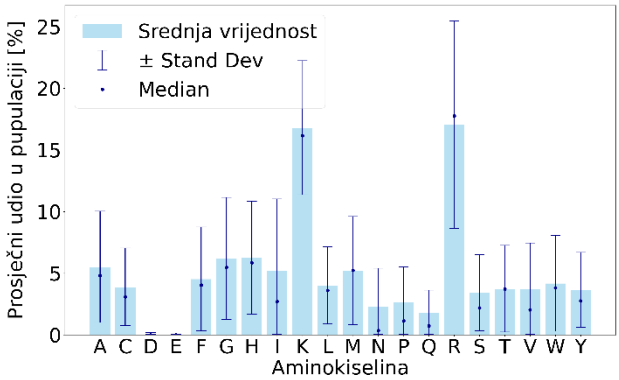
Slika 5.2. Prosječni sastav aminokiselina konačnih generacija - Skup 2



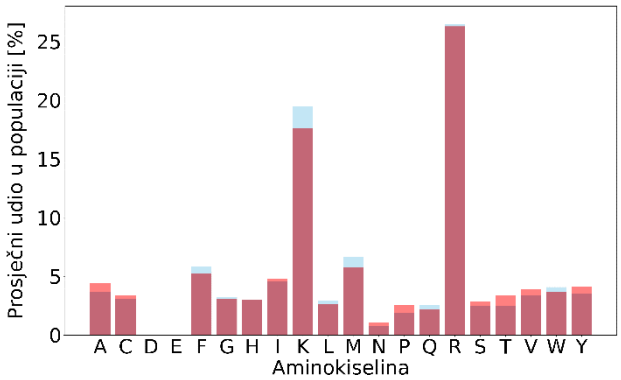
Slika 5.3. Prosječni sastav aminokiselina konačnih generacija - Skup 3



Slika 5.4. Prosječni sastav aminokiselina konačnih generacija - Skup 4



Slika 5.5. Prosječni sastav aminokiselina konačnih generacija - Skup 5



Slika 5.6. Presjek prosječnih sastava aminokiselina stote(crveno) i dvjestote(plavo) generacije - Skup 4

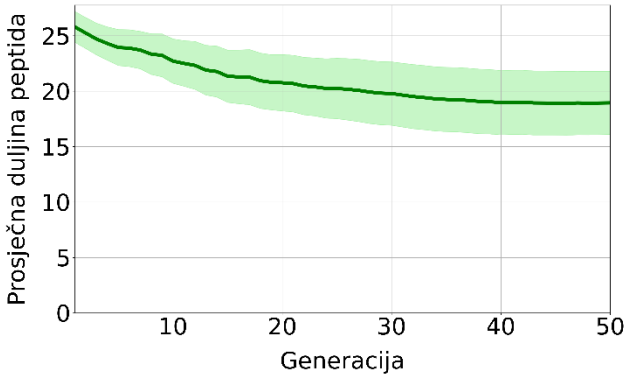
5. 2 Duljina peptidnih sekvenci

Duljina peptidnih sekvenci bitan je čimbenik u razumijevanju svojstva samog peptida. Sve iteracije rada genetskog algoritma svoje izvršavanje kreću iz slučajne početne populacije. Kako je i svakom od peptida u tim početnim populacijama nasumično određena duljina, sve iteracije očekivano započinju s prosječnom duljinom peptida ~26.5 (podsjetimo se, najmanje i najveće dozvoljene duljine peptida u genetskom algoritmu su 3 i 50, (50+3) / 2 = 26.5). No, ta se duljina brzo mijenja. U broju generacija obrnuto proporcionalnom broju potomaka te iteracije (veći broj potomaka – manji broj generacija i *vice versa*) prosječna duljina peptidne sekvence opada. Ponovno je vidljiva razlika između skupova generiranih peptida koji ili ne dostižu potpunu antimikrobnu populaciju ili se zaustavljaju čim to postignu (*Skupovi 1 i 5*) u kojima prosječne duljine peptida u konačnim populacijama iznose 18.94 i 17.73, te onih generiranih skupova čije se izvršavanje nastavlja i nakon postizanja potpune antimikrobne populacije (*Skupovi 2, 3 i 4*) u kojima prosječne duljine peptida u konačnim populacijama iznose 14.83, 14.65 i 14.68. Iz navedenog može se zaključiti da, iako postoje male razlike u prosječnim odstupanjima od srednje vrijednosti, prosječna duljina antimikrobnih peptida nedvojbeno konvergira ka vrijednosti između 14 i 15. No, pošto je duljina svakog peptida cjelobrojna vrijednost i dobiveni rezultati su po svojoj vrijednosti bliži 15, zaključno se utvrđuje da sekvence antimikrobnih peptida generiranih ovim generativnim modelom konvergiraju ka duljini 15 aminokiselina.

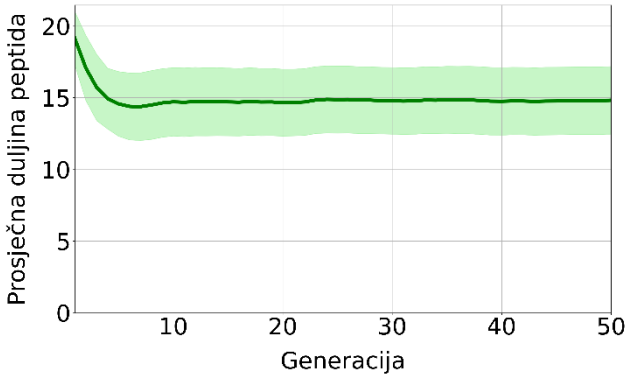
Na slikama *5.7.* - *5.10.* prikazana je promjena u prosječnoj duljini peptida za određeni skup generiranih peptida. Na x-osi prikazane su generacije, a na y-osi prosječna duljina peptidnih sekvenci u toj generaciji. Tamnozelenom crtom naznačena je srednja vrijednost duljine peptida, dok područje označenom svijetlozelenom bojom predstavlja standardnu devijaciju. Svaka slika odnosi se na zaseban skup generiranih peptida.

Zanimljiva problematika proizašla je i kod izračuna prosječne duljine peptida onog skupa generiranih peptida s modificiranim genetski algoritmom (*Skup 5*). U ovom generiranom skupu posljednja generacija svake iteracija varirala je i to većinski u rasponu od 5. do 15. generacije, što je rezultiralo malim brojem iteracija sa znatno većim brojem generacija koje su u prosjeku bile sastavljene od dužih peptidnih sekvenci zbog čega i *Slika 5.11.* izgleda zavaravajuće. Isto se može i uzeti kao potvrda pretpostavke da duljina peptida utječe na njegova antimikrobna svojstva te postojanja optimalnog

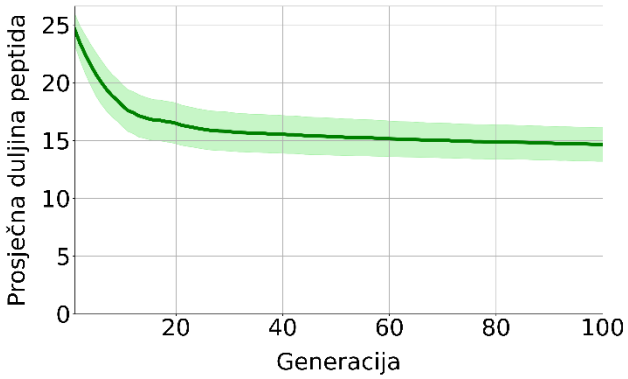
raspona duljina u koji antimikrobni peptidi spadaju, iz razloga što iteracije modificiranog genetskog algoritma s peptidima kraćih duljina prije završavaju sa svojim izvršavanjem, što se događa kad iteracija dostigne potpuno antimikrobnu populaciju. Dakle, iteracije s kraćim peptidima prije (u manjem broju generacija) postižu potpuno antimikrobnu populaciju i kao takve prije "ispadaju" iz izračuna, dok iteracije s pretežno dužim peptidima znatno kasnije (u većem broju potrebnih generacija) postižu potpuno antimikrobnu populacija te duljine njihovih peptida umjetno uvećavaju prosjek kasnijih generacija. U svrhu točnijeg izračuna prosječne duljine peptida za one iteracije kraće od najdulje (one s najvećim brojem generacija) tijekom računanja prosjeka uzimala se u obzir duljina njihove posljednje generacije ukoliko iteracije generaciju za koju se prosjek računao nikada nisu dostigle. Tako dobivenu (prepravljenu) progresiju duljina peptidnih sekvenci prikazuje *Slika 5.12.*



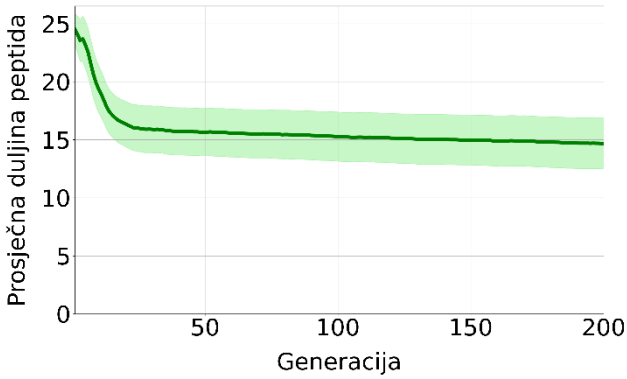
Slika 5.7. Progresija prosječnih duljina peptidnih sekvenci kroz generacije – Skup 1



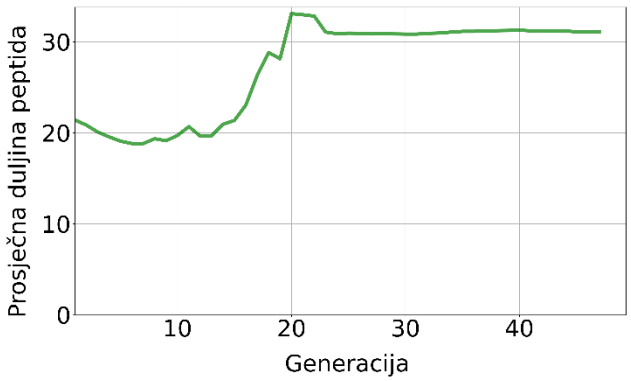
Slika 5.8. Progresija prosječnih duljina peptidnih sekvenci kroz generacije – Skup 2



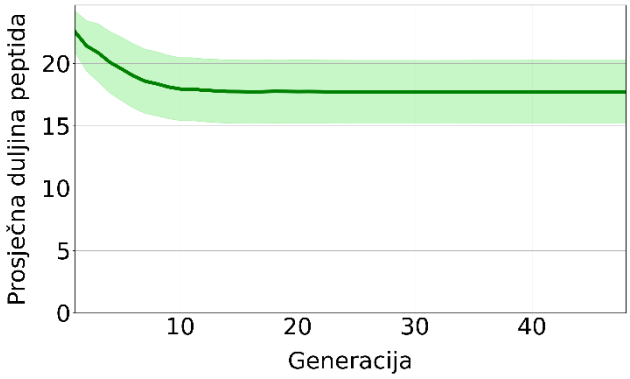
Slika 5.9. Progresija prosječnih duljina peptidnih sekvenci kroz generacije – Skup 3



Slika 5.10. Progresija prosječnih duljina peptidnih sekvenci kroz generacije – Skup 4



Slika 5.11. Zavaravajuća progresija prosječnih duljina peptidnih sekvenci kroz generacije Skup 5



Slika 5.12. Progresija prosječnih duljina peptidnih sekvenci kroz generacije Skup 5

5. 3 Progresija antimikrobnosti jedinki

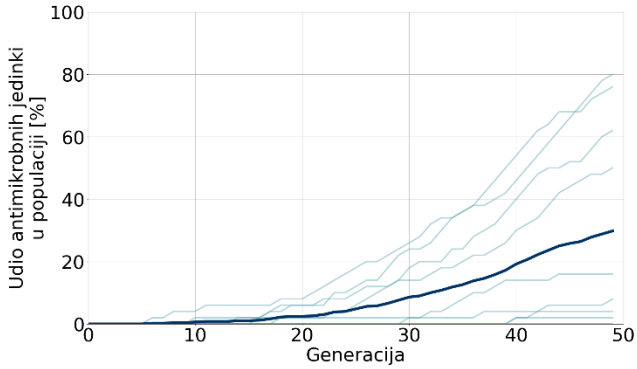
Sve iteracije rada genetskog algoritma svoje izvršavanje započinju iz nasumično generirane populacije peptida. Takvi peptidi toga imaju i širok opseg svojstava, no rijetki posjeduju antimikrobna svojstva, i to u dovoljnom intenzitetu da bi ih funkcija dobrote prepoznala, koja se generativnim modelom ciljaju iskazati. Sva tri ulazna parametra generativnog modela u nekoj mjeri utječu na brzinu, odnosno broj generacija, potrebnu za dostizanje potpune antimikrobne populacije dane iteracije, ali kao dakako najrelevantniji pokazao se broj potomaka. Takav zaključak najlakše je i potkrijepiti pogledamo li one skupove generiranih peptida koji se unutar sebe dodatno dijele na podskupove među kojima je jedina razlika broj potomaka, dok veličina populacije i broj generacija ostaju nepromijenjeni (*Skupovi 3, 4 i 5*). U tim generiranim skupovima jasno se vidi trend opadanja broja generacija u obrnuto proporcionalnoj ovisnosti od broja potomaka, gdje oni podskupovi s većim brojem potomaka u prosjeku u manjem broju generaciju postižu potpuno antimikrobne populacije. Vrijednosti prosječnog broja iteracija potrebnog za dostizanje potpuno antimikrobnih populacija za skupove i podskupove generiranih peptida zajedno s brojem potomaka za svaki od generiranih skupova i podskupova prikazuje *Tablica 5.3*.

Navedeno je i očekivano jer se uvećanjem broja potomaka u svaku generaciju uvodi ne samo veći broj peptida koji su rezultat rekombinacije dvaju, u prosjeku, boljih peptida (onih čija je dobrota veća), nego i jer se veći broj onih lošijih iz svake generacije izbacuje. Time uvećanje broja potomaka rezultira ne samo pozitivnom promjenom u udjelu antimikrobnih peptida u populaciji, nego i ubrzanjem te promjene.

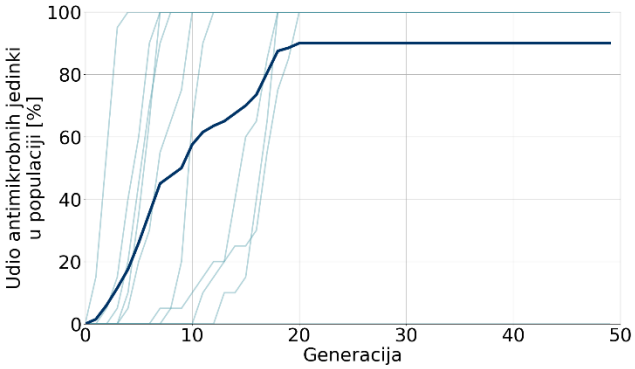
Tablica 5.3. Prosječan broj iteracija potreban za dostizanje potpuno antimikrobne populacije za skupove i podskupove generiranih peptida zajedno s brojem potomaka

	Skup												
	1	2	3				4				5		
Prosječan broj iteracija skupa	0	16.3	28.4				27.9				13.6		
Broj potomaka	2	10	5	10	20	30	5	10	20	30	10	20	30
Prosječan broj iteracija podskupa	0	16.3	65.4	30.8	16	10.4	49.6	32.6	16	13.4	19.1	12.9	9

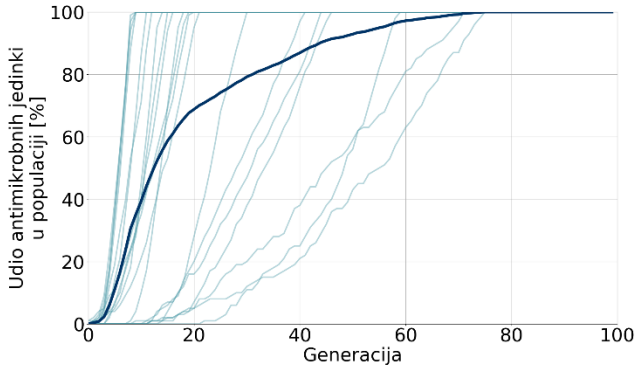
Slika 5.13. - *Slika 5.17.* grafički prikazuje promjenu u udjelu antimikrobnih peptida u populaciji za određeni skup generiranih peptidnih sekvenci. Na x-osi prikazane su generacije, a na y-osi udio antimikrobnih peptidnih sekvenci u toj generaciji. Tanjim svijetloplavim linijama naznačene su progresije udjela antimikrobnosti peptida svake iteracije unutar generiranog skupa, dok je debljom tamnoplavom linijom prikazana njihova srednja vrijednost. Svaka slika odnosi se na zaseban skup generiranih peptida, naznačeno ispod slike.



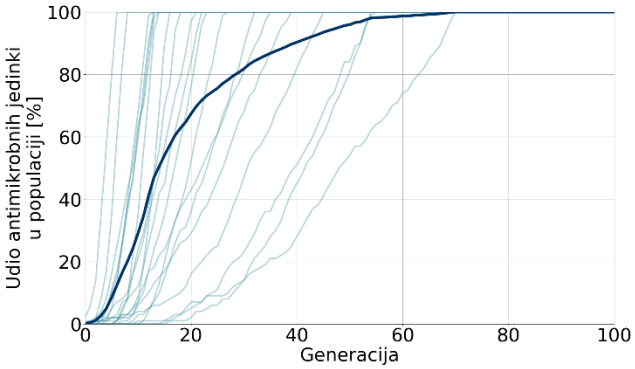
Slika 5.13. Progresija udjela antimikrobnih peptidnih sekvenci u populaciji kroz generacije Skup 1



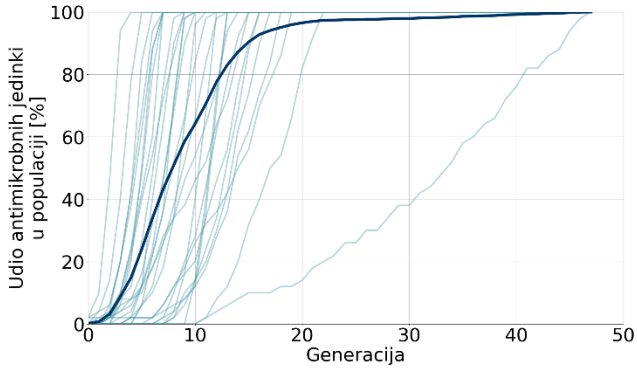
Slika 5.14. Progresija udjela antimikrobnih peptidnih sekvenci u populaciji kroz generacije Skup 2



Slika 5.15. Progresija udjela antimikrobnih peptidnih sekvenci u populaciji kroz generacije Skup 3



Slika 5.16. Progresija udjela antimikrobnih peptidnih sekvenci u populaciji kroz generacije Skup 4



Slika 5.17 Progresija udjela antimikrobnih peptidnih sekvenci u populaciji kroz generacije Skup 5

5. 4 Sličnost peptidnih sekvenci temeljena na Needleman-Wunsch algoritmu

Sličnost se prvo mjerila na razini iteracije, gdje je svaki od peptida konačne generacije sa svakim od preostalih peptida proslijeđen Needleman-Wunsch algoritmu na procjenu sličnosti dviju peptidnih sekvenci (peptidi su uspoređeni "svaki sa svakim", eng. *pairwise comparison*). Prosjek svih tako dobivenih vrijednosti uzet je kao ukupna prosječna sličnost peptidnih sekvenci dane iteracije. Taj proces je potom ponovljen za svaku iteraciju unutar generiranog skupa te se prosjek tih vrijednosti može očitati kao prvi redak *Tablice 5.4.*

Za svaku iteraciju također je pohranjen i "najbolji", najreprezentativniji peptid, odnosno peptid najsličniji svim ostalim peptidima te iteracije (onaj s najvećom prosječnom povratnom vrijednosti Needleman-Wunsch algoritma). Tako dobiveni "najbolji" peptidi unutar generiranog skupa ponovno su uspoređeni "svaki sa svakim" te se prosjek tih vrijednosti može očitati kao drugi redak *Tablice 5.4*.

Isto je ponovljeno i za početne generacije svih iteracija te su rezultati grupirani po generira skupovima generiranih peptida, vrijednosti čega se mogu očitati kao treći i četvrti redak *Tablice 5.4.*

Tablica 5.4. Sličnost peptidnih sekvenci temeljena na Needleman-Wunsch algoritmu u konačnim i početnim populacijama peptida generiranih genetskim algoritmom

	Skup					
	1	2	3	4 (stota generacija)	4	5
Prosjek sličnosti svih jedinki konačnih populacija	0.67651	0.77181	0.59901	0.61247	0.58178	0.70207
Prosjek sličnosti najboljih jedinki konačnih populacija	0.09348	0.13193	0.18231	0.16342	0.18368	0.12298
Prosjek sličnosti svih jedinki početnih populacija	0.07648	0.08148	0.07924	0.07971		0.09907
Prosjek sličnosti najboljih jedinki početnih populacija	0.09823	0.11893	0.10212	0.10725		0.09801

Očekivano, sličnost populacija peptida završnih generacija svih iteracija znatno je viša od ostalih vrijednosti, što se uvelike može doprinijeti međusobnoj srodnosti peptida unutar svake od iteracija.

Novi peptidi koji se uvode u populaciju, nakon njenog inicijalnog slučajnog stvaranja, proizvod su odabira i rekombinacije gena najboljih roditelja, zbog čega jedinke konačne populacije međusobno toliko nalikuju. Čini se i da veće populacije peptida kojima je dano više vremena (generacija) produciraju u prosjeku različitije skupine peptida.

No, isto se ne može reći i za usporedbu sličnosti najboljih jedinki svake konačne generacije pojedine iteracije s drugim najboljim jedinkama ostalih konačnih generacija iteracija unutar generiranog skupa. Dok je ta vrijednost u skupovima koji uspijevaju generirati zadovoljavajući udio antimikrobnih peptida (svi generirani skupovi izuzev *Skupa 1*) veća nego prosječna vrijednost nasumično generiranih peptida pa čak i usporedbe najboljih tako dobivenih peptida (*Tablica 5.4*, redovi 3 i 4), takvo uvećanje u sličnosti može se pridonijeti tome što su antimikrobni peptidi već inherentno sličniji jedni drugima po sastavu aminokiselina i duljini sekvence, čimbenicima koji doprinose procjeni sličnosti korištenjem Needleman-Wunsch algoritma, u odnosu na nasumično generirane peptide.

Razlika između sličnosti jedinki unutar iteracije i sličnosti najboljih jedinki svake iteracije unutar generiranog skupa može se objasniti time da, iako generativni model svakim svojim pokretanjem pretražuje i nalazi antimikrobne peptidne sekvence, zbog razlika u početnim populacijama svako izvršavanje uzima drugačiji "put" kroz peptidni prostor i pronalazi drugačiju, možda i jedinstvenu skupinu antimikrobnih peptida.

6. Zaključak

Napredak tehnologije i medicine ono je što ljudima omogućilo lakše i bolje živote. Dok tradicionalne metode istraživanja i ispitivanja peptida imaju svoje prednosti, njihovi nedostaci otvaraju vrata novim i inovativnim načinima daljnjeg istraživanja peptidnog prostora. Generativni model i modeli poput njega radi svoje brzine i skalabilnosti upravo su takav sljedeći korak u pretrazi peptidnog prostora za novim, dosad nepronađenim rješenjima. U ovom radu pokazano je da uz dobro razumijevanje i poznavanje generativnog modela isti je moguće prilagoditi generiranju peptidnih sekvenci s traženim svojstvima. Kako se ovaj rad primarno fokusirao na traženje antimikrobnih, odnosno terapeutskih peptidnih sekvenci, daljnje tvrdnje odnosit će se na navedene. Terapeutske peptidne sekvence dobivene radom generativnog modela udjelom pretežno sadrže aminokiseline, redom, arginin (R), lizin (K), metionin (M), histidin (H), alanin (A) i glicin (G), što bi sugeriralo da prisutnost baš tih aminokiseline pridonosi antimikrobnim svojstvima peptida, dok aminokiseline asparaginska kiselina (D) i glutaminska kiselina (E) naizgled imaju suprotan učinak. Nadalje, antimikrobne peptidne sekvence dobivene radom generativnog modela konvergiraju ka onima duljine 15 aminokiselina. Također vrijedno spomena jest i da model svakim svojim izvršavanjem proizvodi različite populacije antimikrobnih peptida, čineći ga idealnim za pretragu prostranog i još neistraženog peptidnog prostora. Ovakve zaključke trebalo bi dodatno empirijski i eksperimentalno dokazati, što bi generativnom modelu pridalo dodatnu razinu vjerodostojnosti. Zamijećen je i jedan mogući način napretka generativnog algoritma u cilju poboljšanja njegove efikasnosti. Naime, tijekom procesa generiranja novih populacija peptida jedinke se vrednuju dva puta, prije i nakon stvaranja i dodavanja potomaka u populaciju. Dok je prvo vrednovanje neizbježno i nužno za proces odabira roditelja, ono iduće moglo bi se prilagoditi da u obzir uzima samo novonastale potomke, a ne ponovno cijelu populaciju, izbjegavajući time dvostruko vrednovanje onih jedinki koji su u populaciji već vrednovani. Takva promjena ne bi promijenila niti način rada ni rezultate generativnog modela, ali bi ubrzala njegovo izvršavanje, što bi moglo biti pogotovo bitno ako bi se isti koristi za generiranje većeg skupa podataka.

7. Literatura

[1] Magana, M. i dr. (2020). The value of antimicrobial peptides in the age of resistance. In The Lancet Infectious Diseases (Vol. 20, Issue 9, pp. e216–e230). Elsevier BV. https://doi.org/10.1016/s1473-3099(20)30327-3

[2] Friedberg, F., Winnick, T. i Greenberg, D. M. (1947). Peptide synthesis in vivo. The Journal of biological chemistry, 169(3), 763.

[3] Ardejani, M. S. i Orner, B. P. (2013). Obey the Peptide Assembly Rules. In Science (Vol. 340, Issue 6132, pp. 561–562). American Association for the Advancement of Science (AAAS). https://doi.org/10.1126/science.1237708

[4] Lehninger, A.L., Nelson, D.L. i Cox,M.M. (2005). Lehninger principles of biochemistry. United Kingdom: W. H. Freeman.

[5] OUP accepted manuscript. (2019). In Nucleic Acids Research. Oxford University Press (OUP). https://doi.org/10.1093/nar/gkz1000

[6] Ambrogelly, A., Palioura, S. i Söll, D. (2006). Natural expansion of the genetic code. In Nature Chemical Biology (Vol. 3, Issue 1, pp. 29–35). Springer Science and Business Media LLC. https://doi.org/10.1038/nchembio847

[7] Hombalka, S.: What Are The Two Rare Amino Acids?, https://www.scienceabc.com/pure-sciences/what-are-the-two-rare-amino-acids.html, s Interneta, 06.08.2024.

[8] Huan, Y., Kong, Q., Mou, H. i Yi, H. (2020). Antimicrobial Peptides: Classification, Design, Application and Research Progress in Multiple Fields. In Frontiers in Microbiology (Vol. 11). Frontiers Media SA. https://doi.org/10.3389/fmicb.2020.582779

[9] Ageitos, J. M., Sánchez-Pérez, A., Calo-Mata, P., i Villa, T. G. (2017). Antimicrobial peptides (AMPs): Ancient compounds that represent novel weapons in the fight against bacteria. In Biochemical Pharmacology (Vol. 133, pp. 117–138). Elsevier BV. https://doi.org/10.1016/j.bcp.2016.09.018

[10] Reddy, K. V. R., Yedery, R. D. i Aranha, C. (2004). Antimicrobial peptides: premises and promises. In International Journal of Antimicrobial Agents (Vol. 24, Issue 6, pp. 536–547). Elsevier BV. https://doi.org/10.1016/j.ijantimicag.2004.09.005

[11] Wang, G. (2017). Antimicrobial peptides: Discovery, design and novel therapeutic strategies. CABI.

[12] Otović, E., Njirjak, M., Kalafatovic, D. i Mauša, G. (2022). Sequential Properties Representation Scheme for Recurrent Neural Network-Based Prediction of Therapeutic Peptides. In Journal of Chemical Information and Modeling (Vol. 62, Issue 12, pp. 2961–2972). American Chemical Society (ACS). https://doi.org/10.1021/acs.jcim.2c00526

[13] Erjavac, I., Kalafatovic, D. i Mauša, G. (2022). Coupled encoding methods for antimicrobial peptide prediction: How sensitive is a highly accurate model? In Artificial Intelligence in the Life Sciences (Vol. 2, p. 100034). Elsevier BV. https://doi.org/10.1016/j.ailsci.2022.100034

[14] Negovetić, M., Otović, E., Kalafatovic, D., i Mauša, G. (2024). Efficiently solving the curse of feature-space dimensionality for improved peptide classification. In Digital Discovery (Vol. 3, Issue 6, pp. 1182–1193). Royal Society of Chemistry (RSC). https://doi.org/10.1039/d4dd00079j

[15] G. Shi, X. Kang, F. Dong, Y. Liu, N. Zhu, Y. Hu, H. Xu, X. Lao, and H. Zheng. Welcome to dramp database. http://dramp.cpu-bioinfor.org

[16] S Interneta, https://www.uniprot.org, 04.09.2024.

[17] Mauša, G., Njirjak, M., Otović, E. i Kalafatovic, D. (2023). Configurable soft computing-based generative model: The search for catalytic peptides. In MRS Advances (Vol. 8, Issue 19, pp. 1068–1074). Springer Science and Business Media LLC. https://doi.org/10.1557/s43580-023-00629-8

[18] Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. In Communications of the ACM (Vol. 37, Issue 3, pp. 77–84). Association for Computing Machinery (ACM). https://doi.org/10.1145/175247.175255

[19] Mitchell, M. (1998). An introduction to genetic algorithms. MIT Press.

[20] S Interneta, https://numpy.org/doc/stable, 18.08.2024.

[21] Eiben, A. E. i Smith, J. E. (2015). Introduction to Evolutionary Computing. In Natural Computing Series. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44874-8

[22] Osorio, D., Rondón-Villarreal, P., i Torres, R. (2015). Peptides: A Package for Data Mining of Antimicrobial Peptides. In The R Journal (Vol. 7, Issue 1, p. 4). The R Foundation. https://doi.org/10.32614/rj-2015-001

[23] Osorio, D., Rondon-Villarreal, P. i Torres, R. (2014). Peptides: Calculate Indices and Theoretical Physicochemical Properties of Protein Sequences [Dataset]. In CRAN: Contributed Packages. The R Foundation. https://doi.org/10.32614/cran.package.peptides

[24] Koza, J. R., Bennett, F. H., III, Andre, D. i Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In Artificial Intelligence in Design '96 (pp. 151–170). Springer Netherlands. https://doi.org/10.1007/978-94-009-0279-4_9

[25] Amisha, Malik, P., Pathania, M. i Rathaur, V. K. (2019). Overview of artificial intelligence in medicine. Journal of family medicine and primary care, 8(7), 2328–2331. https://doi.org/10.4103/jfmpc.jfmpc_440_19

[26] Hu, J., Niu, H., Carrasco, J., Lennox, B. i Arvin, F. (2020). Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning. In IEEE Transactions on Vehicular Technology (Vol. 69, Issue 12, pp. 14413–14423). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/tvt.2020.3034800

[27] Yoosefzadeh-Najafabadi, M., Earl, H. J., Tulpan, D., Sulik, J. i Eskandari, M. (2021). Application of Machine Learning Algorithms in Plant Breeding: Predicting Yield From Hyperspectral Reflectance in Soybean. In Frontiers in Plant Science (Vol. 11). Frontiers Media SA. https://doi.org/10.3389/fpls.2020.624273

[28] Needleman, S. B. i Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. In Journal of Molecular Biology (Vol. 48, Issue 3, pp. 443–453). Elsevier BV. https://doi.org/10.1016/0022-2836(70)90057-4

[29] S Interneta, https://scikit.bio/docs/latest/generated/skbio.alignment.global_pairwise_align_protein.html#skbio.alignment.global_pairwise_align_protein, 20.08.2024.

[30] S Interneta, https://www.ncbi.nlm.nih.gov/IEB/ToolBox/C_DOC/lxr/source/data/BLOSUM50, 22.08.2024.

Sažetak

U ovom radu analiziran je generativni model zasnovan na sprezi genetskog algoritma i strojnog učenja u svrhu pretraživanja terapeutskih peptidnih sekvenci. Detaljno je objašnjen rad genetskog algoritma, istaknute su prednosti i nedostaci odabira različitih ulaznih parametara te je analizirana konvergencija algoritma pretraživanja za generiranje peptidnih sekvenci kod ponovljenih mjerenja u nasumično odabranim početnim uzorcima. Analizom je utvrđeno da su terapeutske peptidne sekvence dobivene radom generativnog modela pretežno građene od aminokiselina arginin (R) i lizin (K) te konvergiraju ka duljini 15 aminokiselina. Generativni model svakim svojim izvršavanjem proizvodi različite populacije terapeutskih peptida, čineći ga idealnim za pretragu prostranog peptidnog prostora.

Ključne riječi: generativni model, genetski algoritam, antimikrobni peptidi, aminokiseline

Abstract

This paper presents a generative model based on the combination of a genetic algorithm and machine learning for the purpose of searching for therapeutic peptide sequences. The workings of the genetic algorithm are explained in detail, the advantages and disadvantages of selecting different input parameters are highlighted, and the convergence of the search algorithm for generating peptide sequences is analyzed across repeated measurements in randomly selected initial samples. The analysis determined that the therapeutic peptide sequences created by the generative model are predominantly composed of the amino acids arginine (R) and lysine (K) and tend to converge towards a length of 15 amino acids. The generative model produces different populations of therapeutic peptides with each execution, making it ideal for searching through the vast peptide space.

Keywords: generative model, genetic algorithm, antimicrobial peptides, amino acids