SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br.000

Fraktalna vizualizacija evolucijskim algoritmima

Mirjam Škarica

Zagreb, svibanj 2014.

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

Sadržaj

1. Uvod....1
2. Uvod u bioinformatiku....2
3. Uvod u evolucijske algoritme....4
4 . IFS fraktali 5
4.1. IFS fraktali generirani ulaznim podacima 6
4.2. Sintetiziranje DNA podataka Markovljevim modelom....8
4.3. Struktura jedinke....11
4.4. IFS fraktali generirani evolucijskim algoritmima....12
4.5. Korištene evolucijske strategije....12
5. Rezultati....13
6. Zaključak....14
Literatura....15

1. Uvod

2. Uvod u bioinformatiku

Bioinformatika je širok pojam, sljedeća definicija je predložena od strane NCBI-a (*The National Center for Biotechnology Information).*

> *Bioinformatika svodi pojmove biologije na razinu makromolekula te potom primjenjuje informatičke tehnike (izvedene iz disciplina kao što su primijenjena matematika, računarska znanost i statistika) kako bi se razumjele i organizirale informacije povezane s tim molekulama, u velikim razmjerima.* [1]

Svaka stanica je zapravo dinamičan sustav koji se sastoji od molekula, kemijskih reakcija i kopije genetskog materijala, odnosno genoma tog organizma. Makromolekule nukleinskih kiselina, proteina i ugljikohidrata su presudne za funkcioniranje svih poznatih živih organizama.

DNA (eng. *deoxyribonucleic acid*) i RNA (engl. *ribonucleic acid*) su nukleinske kiseline. DNA kontrolira aktivnosti u stanici određivanjem enzima i drugih proteina koji će se sintetizirati.

DNA se sastoji od dvostruke uzvojnice koje se sastoje o manjih gradivih jedinica nukleotida. Svaki nukleotid se sastoji od fosfata, šećera i nukleinskih baza (adenin, citozin, gvanin i timin) koje označavamo slovima A, C, G i T. DNA je organizirana upakiran u strukture koje nazivamo kromosomima.

RNA obično ima samo jedan lanac iako postoje i RNA s dva lanca. Uz šećer i fosfatne, nukleinske baze koje grade lanac su adenin, citozin, gvanin i uracil koje označavamo sa slovima A, C, G i T.

Proteini su makromolekule sastavljene od jednoga ili više lanaca aminokiselina, čine većinu biomase organizma. Obavljaju različite funkcije npr. ubrzavanje metaboličkih reakcija, umnažanje i prepisivanje DNA, i mnoge druge.

---

1 *„Bioinformatics is conceptualizing biology in terms of macromolecules (in the sense of physical-chemistry) and then applying "informatics" techniques (derived from disciplines such as applied maths, computer science, and statistics) to understand and organize the information associated with these molecules, on a large-scale."* [1]

Objašnjene toka genetskih informacija između DNA, RNA i proteina naziva se centralna dogma molekularne biologije.[2] Opći prijenos je umnažanje DNA, kopiranje informacije iz DNA u RNA. Ova sekvenca RNA dalje nosi informaciju u obliku, tzv. kodona. Kodon je slijed 3 uzastopne nukleinske baze koje određuju umetanje određene aminokiseline u polipeptidni lanac tijekom sinteze proteina, ili signaliziraju početak i prestanak sinteze istih. Postoji 64 različitih kodona, npr. AGU, CUC, GUU.

3. Uvod u evolucijske algoritme

Evolucijski algoritmi traže rješenje problema simulirajući proces evolucije. Ideju EA je predstavio I. Rechenberg 1960-tih godina u svom djelu „*Evolution strategie*" („Evolutionsstrategie"). Danas postoje mnoge verzije EA, ali je većina bazirana na istim principima.

Algoritam započinje stvaranjem određenog broja jedinki. Jedinke su izgenerirane nasumično i svaka od njih predstavlja moguće rješenje problema. Skup svih jedinki jedne generacije čini populaciju. Sljedeći korak je evaluacija populacije koja se radi pomoću funkcije dobrote gdje se svakoj jedinki pridružuje faktor dobrote, bolje jedinke imaju veći faktor dobrote. Izdvajajući određeni broj rješenja (jedinki), formira se nova populacija. Početna nova populacija mijenja se operatorima križanja i mutacije. Algoritam ponavlja ovaj proces sve dok nije pronađeno rješenje problema, odnosno dok jedna jedinka nije zadovoljila uvjete evolucije.

4 . IFS fraktali

Fraktal, kao pojam, skovao je Benoit Mandelbrot 1975. Fraktal je prirodna tvorevina, a opisujemo ga kao geometrijski uzorak koji se ponavlja (barem približno) na svim skalama umanjenosti tvoreći nepravilne oblike i površine koje se ne mogu predstaviti klasičnom geometrijom. Odnosno, fraktali su samoslični bilo da ih gledamo iz bliza ili iz daleka. Osobito se koriste u računalnom modeliranju nepravilnih uzoraka i struktura u prirodi.

IFS (engl. *iterated function systems*) su metode konstrukcije fraktala. Rezultirajuće konstrukcije su uvijek samoslične. IFS fraktali mogu biti bilo koje dimenzije, ali se najčešće računaju i crtaju u 2D. U principu, koristi se skupina jednostavnih transformacija kao što su rotacija, skaliranje i translatiranje kako bi se pomicala točka. Orbita koju dobijemo iterativnom primjenom definiranih transformacija na neku početnu točku je upravo IFS fraktal.

Ako želimo omeđenu orbitu, odnosno fraktal konačne površine, ne mogu se koristiti bilo kave transformacije već samo one za koje vrijedi da se za bilo koji par točaka njihova međusobna udaljenost smanjuje s primjenom te transformacije. Takve transformacije nazivamo kontrakcijskim mapama (engl. *contraction maps*). Formalno, $f : \mathbb{R}^2 \to \mathbb{R}^2$ je kontrakcijska mapa ako vrijedi:

$$d(p,q) \geqslant d(f(p), f(q)) \; \forall \, p, q \in \mathbb{R} \qquad (4.1)$$

Jedan od najpoznatijih primjera je IFS za trokut Sierpinskog. Neka su dane sljedeće transformacije (svaki redak predstavlja jednu od transformacija).

a	b	c	d	e	f	p_i
0.5	0.0	0.0	0.5	0.0	0.0	1/3
0.5	0.0	0.0	0.5	1.28	0.0	1/3
0.5	0.0	0.0	0.5	0.64	0.8	1/3

Tablica 4.1: transformacije za trokut Sierpinskog

Ove transformacije točku $T_i = (x_i , y_i)$ preslikavaju u točku $T_{i+1} = (x_{i+1} , y_{i+1})$, gdje

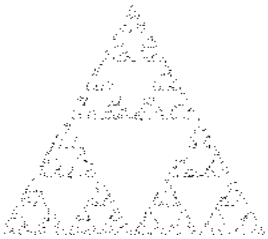
se x_{i+1} , y_{i+1} računaju pomoću funkcija:

$$x_{i+1} = a \cdot x_i + b \cdot y_i + e \tag{4.2}$$

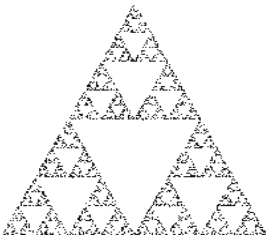
$$y_{i+1} = c \cdot x_i + d \cdot y_i + f \tag{4.3}$$

Fraktal, u ovom slučaju trokut Sierpinskog, dobijemo tako da na početnu točku iterativno primjenjujemo jednu od *3* transformacije iz tablice 4.1, s vjerojatnošću p_i. Općenito mora vrijediti $\sum_{i=1}^{n} p_i = 1$. Počevši od T_o= *(0,0),* i ponavljajući postupak preslikavanja velik broj puta dobiju se rezultati na slikama 4.1- 4.4.

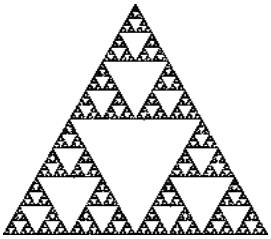
Slika 4.1: 200. iteracija



Slika 4.2: 800. iteracija



Slika 4.3: 3000. iteracija



Slika 4.4: 20000. iteracija

Lako se da zaključiti kako se korištenjem drukčijih vrijednosti koeficijenata, vjerojatnosti ili broja transformacija mogu dobiti vrlo različiti rezultati, pa time i fraktali.

4.1. IFS fraktali generirani ulaznim podacima

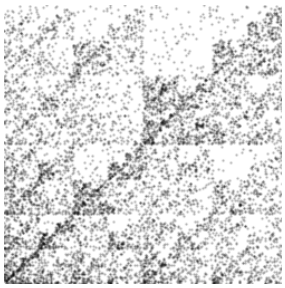
Na primjeru IFS za trokut Sierpinskog je pokazan način generiranja fraktala pomoću transformacija čiji izbor određuje slučajnost, odnosno njihove vjerojatnosti. Ovaj rad se na dalje bavi generiranjem fraktala pomoću transformacija čiji izbor određuju ulazni podaci. Da povučemo paralelu s DNA podacima, najjednostavniji primjer bio bi da imamo definirane neke 4 transformacije, da nam je ulazni niz upravo niz nukleinskih baza (A, C, G, T) te da pojava svake od baza uvijek rezultira primjenom iste transformacije. Algoritam bi tad slijedno prolazio kroz niz te na točku primjenjivao transformaciju koja odgovara zadnjem pročitanom znaku, $znak \in \{A, C, T, G\}$.

Primjer[2] ovih transformacija prikazan je tablicom *4.2*.

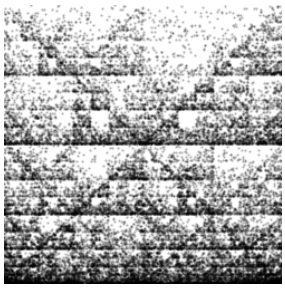
a	*b*	*c*	*d*	*e*	*f*	*znak*
0.5	0.0	0.0	0.5	0.0	0.5	A
0.5	0.0	0.0	0.5	0.5	0.5	T
0.5	0.0	0.0	0.5	0.5	0.0	G
0.5	0.0	0.0	0.5	0.0	0.0	C

Tablica 4.2: Transformacije za IFS fraktal generiran DNA nizom

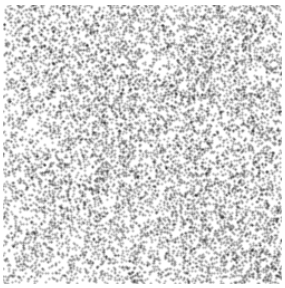
Što ove transformacije rade možemo lakše predočiti slikom *4.8*. Svaka nukleinska baza "privlači" točku u svoj kut, tako da trenutnu točku preslika u novu na pola puta između "svog" kuta i trenutne pozicije točke. Na slikama *4.5-4.6* prikazani su IFS fraktali generirani ulaznim podacima *HIV* genoma, *Methanococcus jannaschii* genoma te s nasumično generiranim DNA podacima, radi usporedbe.



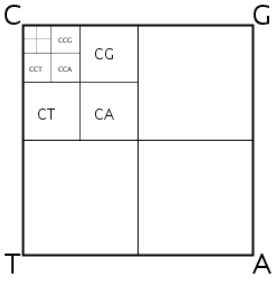
Slika 4.5 HIV



Slika 4.6 Methanococcus jannaschii



Slika 4.7 nasumično generirani podaci



Slika 4.8 legenda

Zanimljivo je primijetiti uzorke dijagonala, izostanak istih, te tzv. *double scoop* koji se može lako primijetiti na slici 4.5 kao relativno prazan prostor u svakom podkvadrantu GC. Pojava *double scoop* uzorka je prvi put zabilježena u ljudskoj beta-globin regiji i ukazuje na relativnu rijetkost uzastopne pojave gvanina i citozina.[3]

Upravo ovo brzo i jednostavno dolaženje do spoznaja je glavni motiv vizualiziranja DNA podataka. To je također i motiv za daljnji pokušaj pronalaska drugih korisnih skupova transformacija koje bi nam s lakoćom otkrivale različite informacije.

2 U literaturi se može pronaći pod nazivima *DNA driven four-cornered chaos game* ili c*haos game representation algorithm.*

4.2. Sintetiziranje DNA podataka Markovljevim modelom

Jedan od mogućih problema koji se mogu javiti je kratak ulazni DNA niz. Naravno, mogli bismo proći kroz dani niz više puta i spremati broj pojavljivanja svake dobivene točke te zatim tu informaciju koristiti kao mjeru učestalosti nekog uzorka. Ali, u praksi se u ovu svrhu koriste Markovljevi lanci.

Markovljevi procesi su oni stohastički (slučajni) procesi čije buduće stanje ovisi samo o trenutnom stanju. To svojstvo zovemo svojstvo odsustva pamćenja. Markovljevi procesi mogu imati diskretan ili kontinuiran skup stanja. Procesi s diskretnim stanjima nazivamo lancima.

Neka Markovljev lanac $X_0, X_1, X_2, \ldots, X_n$ može poprimiti vrijednosti iz skupa diskretnih stanja $S = \{s_0, s_1, s_2 \ldots\}$. Skup stanja može biti beskonačan ili konačan. Oznaka $X_n = i,\ i \in S$ neka znači da se Markovljev lanac u *n*-tom koraku nalazi u stanju *i*. Za opći Markovljev lanac vrijedi sljedeće:

$$P\left(X_{n+1} = j \mid X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n\right) = P\left(X_{n+1} = j \mid X_n = i_n\right) \tag{4.4}$$

za svaki $n \geqslant 0$ i za sve $i_0, \ldots, i_{n-1}, i, j \in S$.

Svojstvo u relaciji (4.4) je upravo svojstvo odsustva pamćenja, odnosno Markovljevo svojstvo. Pretpostavimo da se nalazimo u vremenskom trenutku *n*. Tada vrijeme $n+1$ predstavlja neposrednu budućnost, dok vremena $0,1,\ldots,n-1$ predstavljaju prošlost. Markovljevo svojstvo nam govori da je ponašanje Markovljevog lanca u neposrednoj budućnosti, uvjetno na sadašnjost i prošlost, jednako ponašanju Markovljevog lanca u neposrednoj budućnosti, uvjetno samo na sadašnjost. Prijelaz iz stanja *i* u koraku $n-1$ u stanje *j* u koraku n opisan je prijelaznom vjerojatnošću:

$$p_{ij}^{(n)} = P\left(X_{n+1} = j \mid X_n = i_n\right) \tag{4.5}$$

U općem slučaju ta se vjerojatnost mijenja ovisno o koraku *n*.

Nas će zanimati samo homogeni Markovljevi lanci. To su oni za koje prijelazne vjerojatnosti ne ovise o koraku, odnosno vremenu $n \geqslant 1$. Vrijedi da je:

$$p_{ij}^{(n)} = p_{ij}, \quad \forall i, j, n \tag{4.6}$$

U vezi s tim, uvodimo pojam stohastičke matrice. Matrica $P = (p_{ij} : i, j \in S)$ se naziva stohastičkom matricom ako je $p_{ij} \geqslant 0,\ \forall i, j \in S$ te ako vrijedi:

$$\sum_{j \in S} p_{ij} = 1, \quad \forall i \in S \tag{4.7}$$

Ako je broj stanja u *S* konačan, tada je *P* „prava" (konačna) matrica. S druge strane, ako je *S* beskonačan skup, tada će *P* biti beskonačna matrica.

Ideja je pomoću stohastičke matrice opisati vjerojatnosti pojavljivanja pojedinih uzorka u ulaznom DNA nizu te pomoću tih vjerojatnosti generirati DNA niz proizvoljne duljine koji bi bio vjerna reprezentacija ulaznog niza. U mnogim slučajevima, kao što je i naš, nije lako uočiti i odrediti prijelazne vrijednosti između stanja. U tu svrhu koristimo skrivene Markovljeve modele koji opisuju statističku vezu između promatranog niza i skupa stanja S.

Markovljev model *k*-tog reda DNA niza se dobije tako što se za svaki mogući podniz duljine *k* izračuna kolika je vjerojatnost da ga slijedi baza C, G, A, odnosno T.[4] To se radi na sljedeći način. Ulazni niz se čita redom, za svaki podniz duljine *k*, računa se koliko puta je taj podniz slijedila svaka od četiri baze. Vjerojatnosti se dobiju dijeljenjem tih brojeva s ukupnim brojem pojavljivanja pojedinog podniza. U slučaju da se neki od podnizova ne pojavi, koriste se jednake vjerojatnosti za sve četiri baze, odnosno $p_{podnizC} = p_{podnizG} = p_{podnizA} = p_{podnizT} = 0.25$.

Ilustrirajmo na primjeru ulaznog niza:

AGCAACTAGGCCACCCGGACTACTACCTGCAGGTCCCTAGCATGTATCAA

Kako bi izračunali Markovljev model 2.reda (*k*=2) prolazimo slijedno kroz niz koristeći klizeći prozor veličine 2 kako je prikazano na ilustraciji 4.1

AG CAACTAGGCCACCCGGACTACTACCTGCAGGTCCCTAGCATGTATCAA
A GC AACTAGGCCACCCGGACTACTACCTGCAGGTCCCTAGCATGTATCAA
AG CA ACTAGGCCACCCGGACTACTACCTGCAGGTCCCTAGCATGTATCAA

Ilustracija 4.1

Za svaki trenutni podniz unutar klizećeg prozora broje se ponavljanja sljedeće nukleinske baze. Rezultat toga za ovaj konkretan primjer prikazan je tablicom *4.4*. A sve vjerojatnosti prijelaza prikazane su tablicom *4.5*.
Sada, uz izračunati Markovljev model, možemo generirati tzv. sintetičke DNA nizove proizvoljne duljine po algoritmu:

```
Za početnu vrijednost klizećeg prozora izabrati
  nasumičan podniz duljine k iz ulaznog DNA niza

Dok novi niz nije željene duljine:
     generiraj novu nukleinsku bazu s obzirom na
     trenutni sadržaj klizećeg prozora koristeći
     prijelazne vjerojatnosti

     pomakni klizeći prozor za jedno mjesto
```

Primjer generiranog DNA niza duljine 100 za dani ulazni primjer i klizeći prozor veličine 2:

ATCATCATCAGGACTACCCTAGGTCACCCAACTAGCAGGCCACTAGGCAGG
ACTAGGCATCCTGCCCGGCCGGTCAACCAGGTAGGACCTACCCGGACCT

Radi vjerodostojnosti podataka, dalje u radu koriste se Markovljevi modeli reda 6.

podniz	N_C	N_G	N_A	N_T
CC	2	1	1	2
CG	0	1	0	0
CA	1	1	2	1
CT	0	1	4	0
GC	1	0	3	0
GG	1	0	1	1
GA	1	0	0	0
GT	1	0	1	0
AC	2	0	0	3
AG	1	2	0	0
AA	1	0	0	0
AT	1	1	0	0
TC	1	0	1	0
TG	1	0	0	1
TA	2	2	0	1
TT	0	0	0	0

Tablica 4.3

podniz	P_C	P_G	P_A	P_T
CC	0.333	0.167	0.167	0.333
CG	0.0	1.0	0.0	0.0
CA	0.2	0.2	0.4	0.2
CT	0.0	0.2	0.8	0.0
GC	0.25	0.0	0.75	0.0
GG	0.333	0.0	0.333	0.333
GA	1.0	0.0	0.0	0.0
GTs	0.5	0.0	0.5	0.0
AC	0.4	0.0	0.0	0.6
AG	0.333	0.667	0.0	0.0
AA	1.0	0.0	0.0	0.0
AT	0.5	0.5	0.0	0.0
TC	0.5	0.0	0.5	0.0
TG	0.5	0.0	0.0	0.5
TA	0.4	0.4	0.0	0.2
TT	0.25	0.25	0.25	0.25

Tablica 4.4
Markovljev model 2. reda za dani primjer

4.3. Struktura jedinke

Spomenimo još jednom kako svaka jedinka u evolucijskim algoritmima predstavlja jedno moguće rješenje problema. Zato je izbor i modeliranje strukture jedinke jedno od najvažnijih koraka u EA.

Idealno bi bilo kada bi ista fraktalna reprezentacija radila ujedno s DNA, proteinima i kodonima. Iako se svi ovi oblici mogu dobiti iz bilo kojeg drugog, svaki od njih se pojavljuje u drugom stadiju biološkog procesa. Sirov DNA ima najviše informacija, ali najmanji stupanj interpretabilnosti, dok dijeljenjem DNA podataka u kodone, ona postaje je interpretabilnija (npr. kodon sadrži informaciju o termalnoj stabilnosti DNA)[5]. Dalje u radu, po prijedlogu [Ashclock, Golden], izbor transformacije će ovisiti o kodonima koji dijele DNA niz u 64 moguće trojke.
Jedinka se sastoji od 2 strukture podataka:

- liste transformacija,
- liste duljine 64 čiji elementi sadrže redne brojeve transformacija

Lista transformacija je prikazana tablicom *4.5*. Transformacije su definirane rednim brojem i s još 4 realna broja:

- kutom rotacije Θ izraženog u radijanima,
- translacijom po x-osi Δx,
- translacijom po y-osi Δy,
- faktorom skaliranja s ($0 < s < 1$).

Tako se točka $T_i = (x_i , y_i)$ preslikava u točku $T_{i+1} = (x_{i+1} , y_{i+1})$, gdje se x_{i+1} , y_{i+1} računaju pomoću funkcija:

$$x_{i+1} = s \cdot (x_i \cdot \cos\theta - y_i \cdot \sin\theta + \Delta x) \tag{4.8}$$

$$y_{i+1} = s \cdot (x_i \cdot \sin\theta + y_i \cdot \cos\theta + \Delta y) \tag{4.9}$$

Druga struktura jedinke je prikazana tablicom *4.6*. Lista je duljine 64 jer svaki element odgovara točno jednom kodonu, a sadržaj svakog elementa je redni broj transformacije koja će se primijeniti u slučaju nailaska na taj specifičan kodon. Npr. gledajući tablicu *4.6*. ako algoritam pročita kodon "CCG", na trenutnu točku primijenit će se 2. transformacija.

rbr.	kut rotacije u radijanima (Θ)	translatacija po x-osi (Δx)	translatacija po y-osi (Δy)	skaliranje (s)
1	6.0176	0.8754	0.5239	0.9086
2	1.0304	0.2503	-0.657	0.9729
3	5.9167	-0.525	0.1340	0.6029
4	0.1123	0.3145	0.5260	0.6262
5	0.3586	0.5329	0.3257	0.5669
6	5.7666	0.4117	-0.817	0.9305
7	0.4247	0.9795	0.9789	0.9910
8	3.0488	-0.055	-0.881	0.9690

Tablica 4.5 Primjer liste transformacija

CCC	CCG	CCA	CCT	CGC	CGG	CGA	CGT	...	TTA	TTT
8	2	2	7	4	4	1	3	...	5	3

Tablica 4.6 Primjer liste s koja određuje odnos kodon→transformacija

4.4. IFS fraktali generirani evolucijskim algoritmima

4.5. Korištene evolucijske strategije

5. Rezultati

6. Zaključak

Literatura

1. N. M. Luscombe, D. Greenbaum, M. Gerstein (2001) *What is Bioinformatics? A Proposed Definition and Overview of the Field*. *http://www.ebi.ac.uk/luscombe/docs/imia_review.pdf*
2. F. Crick (1958), *Central Dogma of Molecular Biology* *http://cs.brynmawr.edu/Courses/cs380/fall2012/CrickCentralDogma1970.pdf*
3. Achuthsankar S. Nair, Vrinda V. Nair, Arun K. S., *Bio-sequence Signatures Using Chaos Game Representation* *http://deity.gov.in/hindi/sites/upload_files/dithindi/files/Bio-sewuence_AlpanaDey.pdf*
4. D. Ashlock (2003) *Application to Bioinformatics: Chapter 15* *https://orion.math.iastate.edu/danwell/ma378/chapter15.pdf*
5. D. Ashlock, J. Golden, *Evolutionary Computation and Fractal Visualization of Sequence Data* *http://eldar.mathstat.uoguelph.ca/dashlock/eprints/biochapter.pdf*

Fraktalna vizualizacija evolucijskim algoritmima

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: ključne riječi, odvojene zarezima.

Fractal visualization with evolutionary algorithms

Abstract

Abstract.

Keywords: Keywords.