# Obrada mikoloških uzoraka

Mauro Raguzin, Petar Pavlović

Lipanj 2023.

## 1 Uvod

U ovom dokumentu istražit ćemo kako se napisani jezik iz Interpretacije programa koristi, koje ima funkcionalnosti i kako bi se mogao primijenjivati u budućnosti. Jezik se bavi obradom mikoloških uzoraka, to jest omogućava lako rukovanje karakteristikama gljiva: imenima, latinskim imenima, DNA uzorcima, taksonomijom i vremenom pronalaska.
Jezik podržava neke bazične tipove podataka i operacije, poput brojeva, string imena, vektora, računskih operacija, pridruživanja vrijednosti, stvljanja komentara te if-else uvjeta i for petlji. Također je omogućeno baratanje mjernom jedinicom mase, pisanje i čitanje iz datoteka, te stvaranje vlastititih funkcija. Ono što ovaj jezik čini specifičnim za posebnu primjenu su genetski operatori na koje ćemo se u ovom dokumentu malo više osvrnuti.

## 2 Genetski operatori

DNA sekvenciranje je proces određivanja redoslijeda nukleotida kod neke jedinke. Postoji četiri vrste nukleotida: timin (T), adenin (A), citozin (C) i gvanin (G). Timin i adenin, te citozin i gvanin uvijek dolaze „u paru" spojeni jedan na drugoga, tako da je navodenjem nukleotida s jedne strane DNA strukture cijela sekvenca jedinstveno definirana. Odabrali smo tri genetska operatora: mutaciju, križanje i selekciju. Ponašanje tih operacija u prirodi je složeno, i njihove prave verzije bi trebao implementirati neki mikolog koji će koristiti ovaj jezik. Stoga smo se ograničili na neko proizvoljno, ali razumno ponašanje tih operatora kako bi se pokazala njihova osnovna funkcionalnost na kraćim nizovima u demonstrativne svrhe.

### 2.1 Križanje

Operator križanja simulira spajanje genetskog materijala. On prima dvije gljive i vraća njihovo genetsko „dijete". Algoritam nasumično izabire hoće li duljina djetetovog genoma biti kao duljina prve ili druge gljive, te do duljine kraćeg roditelja nasumično

izabire nečiji gen na tom mjestu, a na ostatku uzima gene dužeg ako je tako izabrano. Također je omogućeno križanje na listama jedinki.

## 2.2 Mutacija

Mutacija je promijena redoslijeda nukleotida u genomu. Ovaj operator prima tip podatka gljiva (Fungus) ili listu više gljiva i vraća gljive mutoranog genetskog materijala. Algoritam koji je korišten prolazi kroz DNA sekvencu gljive i zamijeni u prosjeku trećinu nukleotida nekim nasumičnim nukleotidom (možda i istim). U prirodi se kod mutacije sigurno mijenja puno manje od trećine gena, ali ideja je bila da se u našem programu uvijek dogodi nekakva promijena ali da se opet vidi sličnost s originalnom gljivom.

## 2.3 Selekcija

Operator selekcije uzima listu jedinki i po nekom kriteriju bira onu s „najboljim" genetskim kodom i nju vraća. Kriterij koji je izabran je genetska raznolikost, dakle bit će izabrane gljive koje nemaju veliku razliku u broju različitih nukleotida. To se postiže tako da se računa zbroj kvadrata svih razlika između bojeva nukleotida. Istina je da priroda cijeni i nagraduje genetsku raznolikost, ali naravno da je u praksi to kompleksniji algoritam.

# 3 Primjeri programa

**Pogledajmo prvo primjere nekih mogućnosti koje jezik nudi.**

## 3.1 Prvi program

```
let var := "nesto";
if (var) {
   var := "novo";
}
print (var);
var := 5;
let accum := 0;
for var {
   accum := accum + 2;
   print (accum);
}
let v1 := [3, 54g, 200mg, "prvi i ", -1];
let v2 := [1, 1g, 1g, "drugi", 0];
print(v1+v2);
let datum := 1.1.2011.;
let vrijeme := 13.9.2013. 10:10;
print (datum);
print (vrijeme);
let ne := Number("14.22"); # konvertirajuci konstruktor
```

```
print(ne);
```

**Rezultat ovog programa:**

```
novo
2.0
4.0
6.0
8.0
10.0
[4.0, 55.0 g, 1200.0 mg, prvi i drugi, -1.0, ]
1.1.2011.
13.9.2013. 10:10:0
14.22
```

## 3.2 Drugi program

**Kako funkcioniraju funkcije i datoteke:**

```
function mojafun(a, b) {
  let null;
  if(a) {
    null := "a je istinit";
  }
  else {
    null := 0;
  }
  return null;
}
write("datoteka1.txt",mojafun(true, 0));
write("datoteka2.txt",mojafun(false, 0));
let ucitano1 := read("datoteka1.txt");
let ucitano2 := read("datoteka2.txt");
print(ucitano1);
print(ucitano2);
```

**Rezultat ovog programa:**

```
a je istinit
0.0
```

## 3.3 Treći program

Prikažimo rad sa gljivama. Svaka gljiva ima **ime**, **latinsko ime**, **DNA uzorak**, **taksonomiju** i opcionalno **vrijeme pronalaska**. Ako je vrijeme pronalaska nenavedeno, samo se stavlja trenutno vrijeme.

```
let dna1 := DNA(ATCGGTACG);
let dna2 := DNA(GCCGGTCCG);
let tax := Tree();
tax.phyl := "neka filogeneza";
tax.king := "neko kraljevstvo";
```

```
let datum := 25.3.2008.;
let fung1 := Fungus("naziv", "latinski", dna1, tax, datum);
let fung2 := Fungus("naziv2", "latinski2", dna2, tax);
print(fung1);
print(fung2);
```

**Rezultat:**

```
Name: naziv
Latin name: latinski
DNA: ATCGGTACG
Taxonomy:
phylum: neka filogeneza
kingdom: neko kraljevstvo
Time found: 25.3.2008.

Name: naziv2
Latin name: latinski2
DNA: GCCGGTCCG
Taxonomy:
phylum: neka filogeneza
kingdom: neko kraljevstvo
Time found: 18.6.2023. 12:56:19
```

## 3.4 Genetski operatori

```
let dna := DNA(AAAAAAAAAAA);
let tax := Tree();
tax.king := "Fungi";
tax.class := "Agaricomycetes";
tax.ord := "Agaricales";
tax.fam := "Amanitaceae";
tax.gen := "Amanita";
tax.spec := "A. muscaria";

let gljiva := Fungus("muhara", "Amanita muscaria", dna, tax);
let gljiva2 := Fungus("muhara2", "Amanita muscaria", dna2, tax);

let mutirana := mutate gljiva;
let mutirana2:= mutate mutirana;
let dijete := mutirana cross mutirana2;
let najbolja := [gljiva, mutirana, mutirana2]select;
print(gljiva);
print("mutirana");
print(mutirana.dna);
print("mutirana2");
print(mutirana2.dna);
print("dijete");
print(dijete.dna);
print("najbolja");
print(najbolja.dna);
```

**Rezultat:**

```
Name: muhara
Latin name: Amanita muscaria
DNA: AAAAAAAAAAA
Taxonomy:
species: A. muscaria
genus: Amanita
family: Amanitaceae
order: Agaricales
klasa: Agaricomycetes
kingdom: Fungi
Time found: 18.6.2023. 14:22:52

mutirana
AAATAAATAAA
mutirana2
CATTAACTAAA
dijete
CATTAAATAAA
najbolja
CATTAACTAAA
```

# 4 Budućnost

Ovdje prikazan jezik je tek početak, kostur nečega što bi moglo stvarno biti korisno ljudima koji se ovom temom bave profesionalno. Kada bi se uvela još neka poboljšanja i, naravno, genetski operatori učinili realističnima kao u prirodi, ovaj jezik stvarno bi mogao poslužiti u znanstvene svrhe. Imati jezik prilagođen specifičnoj temi olakšalo bi rad jer detalji implementacije i sam programski jezik na kojem je jezik baziran postaju nebitni, sve je konkretno usmjereno na područje znanstvenog rada.

Područje genetike još je uvijek velikim dijelom nepoznato područje otvoreno novim znanstvenim spoznajama, pa bi ovakav alat sigurno našao svoju svrhu. Bilo bi također moguće proširiti jezik i na neke biljke i životinje i tako općenitije istraživati genetiku, simulirati procese u prirodi i uspoređivati simulacije sa stvarnošću. Kada bi se genetski operatori s vremenom učinili dovoljno sličnima onima u prirodi, možda bi bilo moguće čak i pokušati simulirati evoluciju neke vrste. Zvuči ambiciozno, i možda bi to bilo ipak neizvedivo, ali ako bi bilo moguće, ovaj jezik bi sigurno u tome pomogao.

# Literatura

[1] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992. ISBN: 0262111705.

Ova knjiga bi bila korisna pri daljnjem istraživanju mogućih smjerova razvoja ovog jezika. Za sada su korištene samo osnovne ideje triju genetskih operatora, ne i specifičnih detalja njihovih mogućih implementacija iz knjige.