

Obrada informacija

Laboratorijska vježba 2

U ovoj vježbi upoznat ćete se s jednom primjenom tehnika obrade informacija u bioinformatici. Ova laboratorijska vježba nosi 4 boda. Izvješće s ove laboratorijske vježbe potrebno je predati u .pdf formatu na *Moodle*. Izvješće koje predajete se mora zvati *PrezimeIme.pdf*.

Osim biblioteka za rad s Fourierovom transformacijom (koristit ćemo samo numpy) koristit ćemo i biblioteku biopython koja sadrži puno korisnih alata iz područja bioinformatike. Mi ćemo je koristiti za jednostavnije baratanje bioinformatičkim tipovima podataka.

Biblioteka biopython dolazi s instalacijom Anaconde, ali ju je potrebno uključiti u okolinu (*environment*) koja se koristi.

Ako vježbu izvodite u Google Colab okruženju, morate instalirati biblioteku biopython. Instalaciju je potrebno izvršiti u sklopu prvog zadatka ove laboratorijske vježbe. Instalaciju izvodite sljedećim kodom:

```
try:
    import google.colab
    !pip install biopython
except ImportError:
    pass
```

Nakon izvođenja ovog koda, možete učitati biopython biblioteku.

1. Zadatak

Python biblioteke potrebne za laboratorijske vježbe su numpy i biopython. Uključite ih ("importirajte") i ispišite verziju svake od njih pomoću [ime_biblioteke].__version__.

UPUTA: Osnovna biopython biblioteka ima naziv Bio.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. Zadatak

Uz laboratorijske vježbe dobili ste dvije datoteke s podacima. Datoteku koja sadrži referentni genom jednog soja bakterije *Escherichia coli* (*escherichia_coli_reference.fasta*) u FASTA formatu i datoteku koja sadrži skup očitavanja dobivenih sekvenciranjem (*ecoli_ILL_small.fastq*) u FASTQ formatu.

Datoteke možete učitati koristeći metodu *parse()* iz biblioteke *Bio.SeqIO*. Metoda *parse()* vraća iterator koji možete pretvoriti u Python listu na sljedeći način:

```
reads = list(parse("ime_datoteke", "tip_datoteke"))
```

Tip datoteke postavite na "fasta" ili "fastq".

Učitajte obje datoteke te ispišite broj zapisa u svakoj od njih (broj elemenata u listi). Datoteka koja sadrži referencu trebala bi imati samo jedan zapis, dok bi datoteka s očitanjima trebala sadržavati veći broj zapisa.

NAPOMENA: Ako niste sigurni kako pronaći datoteke na disku iz Jupyter notebook-a, uvijek možete provjeriti radni direktorij sljedećim naredbama:

```
import os
os.getcwd()
```

i promijeniti ga sa:

```
os.chdir()
```

Ako pak radite u Google Colab okruženju, koristite upute za učitavanje datoteka s Google diska iz prve laboratorijske vježbe.

```
# Ovo je mjesto na kojem možete izvoditi svoj kod.
```

1. Zadatak

Svaki zapis koji ste učitali pomoću metode *Bio.SeqIO.parse()* sadrži Veći broj podataka od kojih su nam bitni samo neki. Naredbom `print` ispišite cijeli prvi zapis iz datoteke s očitanjima i iz datoteke s referencom.

Vidjet ćete da oba zapisa (među ostalim podacima) sadrže identifikator zapisa i sekvencu. Identifikator zapisa možete dohvatiti pomoću

```
zapis.id
```

dok sekvencu možete dohvatiti pomoću

```
zapis.seq
```

Ispišite identifikator i sekvencu za prvo očitavanje te identifikator i prvih 200 znakova za referentni genom *E.coli*.

NAPOMENA: Referentni genom *Escherichia coli* je dugačak oko 4.5 milijuna slova

```
# Ovo je mjesto na kojem možete izvoditi svoj kod.
```

1. Zadatak

Da bismo sekvence DNA analizirali metodama obrade signala, moramo pojedinim nukleotidima (slovima) dodijeliti brojčane vrijednosti. Napišite funkciju u Pythonu koja će primiti slovo koje predstavlja nukleotid i vratiti odgovarajuću brojčanu vrijednost. Vrijednosti dodijelite na sljedeći način:

- A = 3
- G = 2
- C = -2
- T = -3

DNA sekvence mogu sadržavati i neke druge znakove (npr. 'N' koji označava da taj nuklotid nije poznat), njima dodijelite vrijednost 0. Također se može dogoditi da nukleotidi budu označeni i malim slovima, pa vodite računa da vaša funkcija mora vratiti ispravnu vrijednost i u tom slučaju.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. Zadatak

Upotrebite napisanu funkciju da bi od prvog očitavanja i od reference kreirali signal. Izračunajte korelaciju pomoću Fourierove transformacije. Zanimajte imaginarne vrijednosti.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. Zadatak

Ispišite duljinu reference. Koristeći metode biblioteke *numpy*, izračunajte srednju vrijednost i standardnu devijaciju duljine očitavanja (uzmite u obzir sva očitavanja).

Primijetite da su sva očitavanja jednake duljine.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. zadatak

Što ako želimo izračunati korelaciju za veći broj očitavanja i istu referencu? To je tipičan slučaj u bioinformatičari jer uređaji za sekvenciranje proizvode tisuće i desetke tisuća očitavanja koja se potom mapiraju na istu referencu.

Ako korelaciju računamo izravno, potrebno ju je svaki put izračunati iz početka. Ako korelaciju računamo pomoću FFT-a, transformaciju za referencu potrebno je napraviti samo jednom.

Izračunajte korelacije za prvih 10 očitavanja.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. zadatak

Na temelju najveće vrijednosti korelacije između reference i prvog očitavanja pronađite poziciju na referenci koja je najbližija očitavanju. Pozicija odgovara vrijednosti parametra *k* za koji je korelacija najveća.

Napišite metodu koja će primiti dva niza znakova jednake duljine, usporediti znakove na istim pozicijama i vratiti broj razlika (Hammingova udaljenost).

"Izrežite" dio reference koji je najbliži očitavanju (iste duljine kao i očitavanje) i usporedite ga s očitavanjem pomoću napisane funkcije.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. zadatak

U datoteci "ecoli_ILL_small_aln.sam" dana su već izračunata poravnanja svih očitavanja na referencu u SAM formatu. SAM je tekstualni "tab separated" format. U prvom stupcu se nalazi identifikator očitavanja, dok se u četvrtom stupcu nalazi pozicija na referenci na koju je očitavanje najbolje poravnato (ostali stupci nas ne zanimaju). Također, datoteka s poravnanjima sadrži i nekoliko *header*redaka kojima prvi stupac počinje sa znakom '@', njih također možete zanemariti.

Otvorite datoteku s poravnanjima i pronađite poravnanje za prvo očitavanje (identifikator očitavanja u datoteci s očitavanjima i datoteci s poravnanjima mora biti jednak). Usporedite poziciju u datoteci sa pozicijom koju ste dobili pomoću korelacije.

UPOUTA: TSV datoteke možete otvoriti na sljedeći način:

```
tsv_file = open("file_name")
tsv_rows = csv.reader(tsv_file, delimiter="\t")
```

Varijabla *tsv_rows* će sadržavati listu redaka, a svaki redak biti lista vrijednosti (po jedna za svaki stupac).

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. zadatak

Za prvo očitavanje pozicija dobivena pomoću korelacije trebala bi biti 2324486, dok je pozicija u datoteci s poravnanjima 2324487. Razlikuju se samo za 1 pa možemo zaključiti da nam je korelacija dala dobru poziciju za poravnanje.

Prisjetimo se da korelacija ne računa točno poravnanje već ju koristimo samo da bi našli kandidatne pozicije za točno računanje. Tek onda na takvim pozicijama možemo točno poravnanje izračunati pomoću algoritama dinamičkog programiranja. Ako bi primijenili dinamičko programiranje za računanje poravnanja očitavanja s cijelom referencom, postupak bi bio znatno sporiji i zahtijevao bi veliku količinu radne memorije.

Ako želite to možete isprobati pomoću algoritama za poravnanje u biblioteci *bioparser*. Lokalno poravnanje možete izračunati metodom:

```
Bio.pairwise2.align.localxx(seq1, seq2)
```

Za prvih 100 očitavanja izračunajte korelaciju te pomoću korelacije poziciju najveće sličnosti očitavanja i reference. Usporedite rezultat sa podacima u datoteci s poravnanjima. Ispišite broj očitavanja za koja se dvije pozicije razlikuju za najviše 5 mjesta.

Ovo je mjesto na kojem možete izvoditi svoj kod.

1. ZAKLJUČAK

Očekivani broj točno pozicioniranih očitavanja je 50, jer smo za sada uspješno radili samo s očitanjima na jednom lancu DNA.

Prolaskom kroz zadatke u ovoj vježbi dobili ste kratak uvod u rad s bioinformatičkim podacima i tehnikama obrade signala.