

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2324

**Orthobalancer: aplikacija za
kreiranje skupova bioloških vrsta
usporedive taksonomijske širine**

Ivan Slijepčević

Zagreb, svibanj 2012.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da biste uklonili ovu stranicu obrišite naredbu \izvornik.*

zahvale

SADRŽAJ

Popis slika	v
Popis tablica	vi
1. Uvod	1
2. Teoretski uvod	2
2.1. Homologija proteina	2
3. Podaci	3
4. Metode	4
5. Implementacija	5
5.1. Cjevovod	5
5.2. tax	7
5.3. server	9
6. Rezultati	10
7. Zaključak	11
Literatura	12

POPIS SLIKA

POPIS TABLICA

1. Uvod

2. Teoretski uvod

ključna uloga proteina

srodnost / evolucija / otkrivanje

2.1. Homologija proteina

Homologija u biološkom smislu predstavlja slične osobine među vrstama na različitim razinama organizacije života, poput organa, tkiva, stnice ili molekule. Homologne osobine uočene među jedinkama različitih vrsta obično upućuju na zajedničke pretke tih vrsta u evoluciji. Međutim, u molekularnoj biologiji termin homolog se često koristi i za naznačavanje sličnosti, bez obzira na genetsko srodstvo [1]

Za homologne sekvence proteina kažemo da su ortologne kad su direktni potomci neke sekvence u zajedničkom pretku, bez da su prošle duplikaciju gena. Drugim riječima, ortologne sekvence se mogu naći u jedinkama različitih vrsta, a obavljaju istu funkciju u svim tim vrstama. Paralogne sekvence su homologne sekvence koje su nastale od dvije različite kopije nekog gena koji je prošao kroz proces duplikacije gena u nekom zajedničkom evolucijskom pretku. Paralozi se mogu naći u jedinkama jedne ili više vrsta te obavljaju slične funkcije.

chart ortho-para

ideja...

3. Podaci

s FASTA formatu

blast nr baza

taxonomy baza

ulaz

izlaz

4. Metode

pipeline neki dijagram za pipeline (sequence / activity / state)

tax dio pseudokod

neki sequence dijagram za sve

izlaz

(prebaciti u neki teex file za server) server slike (ulaz, zamjenjivi, izvršavanje, kraj, error)

5. Implementacija

Aplikacija je pisana u programskom jeziku Python verzije 2.7. Aplikacija se dijeli u nekoliko zasebnih cjelina. U središtu aplikacije nalazi se cjevovod koji poziva alate poput BLAST-a i fastacmd-a za komuniciranje sa NCBI-jevom neredundantnom bazom, zatim dio aplikacije za odabir i balansiranje vrsta na taksonomskom stablu te alat mafit za poravnanje sekvenci. Pored cjevovoda implementirana je web aplikacija kao korisničko sučelje za cijeli program. Web aplikacija je implementirana koristeći Flask microframework, dok su operacije na klijentskoj strani implementirane u javascriptu uz korištenje biblioteke jQuery.

5.1. Cjevovod

Cjevovod je arhitektonski programski obrazac u kojem prolaze kroz filtre koji su postavljeni jedan za drugim. Time se simulira jedan tok koji ulazne podatke transformacijom kroz filtre generira izlazne podatke. U ovome projektu cjevovodna arhitektura je samo logički kostur koji se enkapsulira unutar razreda *Pipeline*. Iako je u začetku razvoja aplikacije svaki filter bio zaseban proces, vrlo ubrzo je ustanovljeno kako većina filtera generira podatke koji su potrebni na raznim mjestima u cijeloj aplikaciji te se činilo lakše imati sve podatke u memoriji pojedinog cjevovoda. To je omogućilo da razred *Pipeline* naslijedi razred *Thread* iz modula *threading* te se može pozivati kao zasebna dretva.

Tok cjevovoda se može vidjeti na slici 5.1. Ulaz u cjevovod predstavljaju paralogni proteini u FASTA formatu koje zadaje korisnik. Ti se podaci zadaju pri stvaranju objekta *Pipeline* kako bi se mogli zapisati na disk u direktorij vezan za instancu *Pipeline-a*. Stvarni objekt kojeg prima konstruktor *Pipeline-a* je rječnik prilagođen uporabi servera, što je detaljnije opisano u sekciji 5.3.

Pri pokretanju cjevovoda za svaku se od unesenih sekvenci stvara objekt razreda *ProteinHolder* prilikom čega se obavljaju pozivi filtera nezavisnih za svaku pojedinu sekvencu. Prvi filter koji se koristi je alat *BLAST* te je izveden kao poziv zasebnog

izvršnog programa *blastall* na sljedeći način:

```
blastall -p blastp -i <ulaz-FASTA> -d <nr-baza> -m 8 -a <broj-dretvi>
```

Argument *p* s parametrom *blastp* označava programu da se koristi algoritam za uspoređivanje jedne ulazne sekvence amino kiselina sa bazom proteinskih sekvenci. S argumentom *i* se zadaje put do ulazne datoteke s FASTA sekvencom. Nadalje, argument *d* prima put na disku do NCBI-jeve neredundantne baze sekvenci koja je prethodno formatirana za pretraživanje sekvenci u FASTA formatu. Argument *m* određuje format ispisa koji generira *blastall*, a parametar 8 označava tabularni ispis bez dodatnih komentara koji je pogodan za parsiranje. Konačno, argument *a* upućuje *blastall* na broj dretvi koji treba koristiti kako bi se ubrzalo njegovo izvođenje. Broj dretvi se u Orthobalanceru može postaviti prilikom instalacije.

Izlaz koji generira *BLAST* predstavlja informacije o sekvencama sličnim ulaznoj sekvenci, odnosno informacije o potencijalnim ortolozima za ulazni paralog. Svaki redak, između ostalih, sadrži dvije bitne informacije: jedinstveni ključ sekvence u neredundantnoj bazi — gi-broj — te ocjenu sličnosti ulaznome paralogu. Nakon parsiranja tog izlaza ocjene sličnosti se spremaju za kasniju upotrebu, a gi-brojevi se zapisuju u privremenu datoteku za sljedeći korak u cjevovodu.

Sljedeći filter je izvršni program *fastacmd* koji za svaki gi-broj pronalazi i ispisuje sekvencu u FASTA formatu, također koristeći NCBI-jevu neredundantnu bazu. Program se poziva ovako:

```
fastacmd -i <ulaz> -d <nr-baza>
```

Budući da program *fastacmd* dolazi u paketu zajedno sa *BLAST* alatima, argumenti imaju sličnu konotaciju kao i za *blastall*: s argumentom *i* se zadaje ulazna datoteka, a s *d* put do neredundantne baze.

Izlaz *fastacmd*-a se parsira pomoću razreda FastaRecord. Svaka dobivena sekvenca kao kandidat za ortologa dobiva instancu razreda FastaRecord u kojoj je sadržana sekvenca te sve bitne informacije iz zaglavlja pojedinog FASTA zapisa. Dodatno, instanci FastaRecord se pridružuje ocjena sličnosti sekvence koju opisuje prema ulaznom paralogu. Kako bi bilo lakše objasniti što je preuzeto iz zaglavlja pojedinog FASTA zapisa, bitno je imati na umu strukturu neredundantne baze što je objašnjeno u poglavlju 3. Naime, ako je za nekoliko proteina zabilježeno da imaju identičnu sekvencu, tada će zaglavlje takve FASTA sekvence u neredundantnoj bazi sadržavati spojene podatke o navedenim proteinima. Zato objekt razreda FastaRecord sadrži listu elemenata zaglavlja, gdje se svaki od tih elemenata opisuje n-torkom (gi-broj elementa, ime vrste, ime proteina, zastavica: najbolja sekvenca za ovaj gi-broj). Iako je poznato da nere-

dundantna baza sadrži sekvence sakupljene iz raznih baza što implicira činjenicu da zaglavlja pojedinih sekvenci ne moraju imati identičan oblik, uočeno je određeno pravilo po kojem *fastacmd* ispisuje sekvence te se ono koristi kao heuristika ugrađena u parser. Pretpostavljeni oblik pojedinog elementa zaglavlja je sljedeći:

```
>gi|"gi-broj"|nebitne-informacije IME PROTEINA  
[IME VRSTE]nebitna-informacija.
```

Na primjer:

```
>gi|344243907|gb|EGW00011.1| Cofilin-1 [Cricetulus griseus]
```

Na kraju obrade podataka za pojedini paralog s ulaza prolazi se listom svih objekata FastaRecord te se za svaku pronađenu vrstu odabire sekvenca s najboljom ocjenom sličnosti. Time nastaje jedan podskup sekvenci za koje možemo reći da predstavljaju grupu ortologa za dani paralog.

Nakon što je svaki paralog opisan jednim objektom razreda *ProteinHolder* potrebno je odrediti postoji li koji gi-broj koji se može pronaći u grupama ortologa različitih paralog. Donesena je odluka kako to svojstvo nije poželjno jer jedan te isti protein ne želimo imati kao predstavnika neke vrste u različitim grupama ortologa. Za potrebe ove funkcionalnosti dodani su razredi *BestScore* i *BestScoreCollection*. Razred *BestScore* sadrži informaciju koja upućuje u kojoj se grupi ortologa, na kojem FASTA zapisu te sa kojim elementom zaglavlja FASTA zapisa nalazi najbolje ocijenjena sekvenca za dani gi-broj identifikator. Razred *BestScoreCollection* je sučelje za korištenje rječnika najboljih sekvenci, a nudi metode za ažuriranje rječnika kandidatima za najbolje sekvence te metodu za dohvat trenutno postavljene najbolje instance razreda *BestScore*. Skica algoritma za pronalazak najbolje ocijenjenih dana je algoritmom 1.

Ovdje bi se još mogla dodati funkcionalnost kojom bi se za odbačenu vrstu pronašla sljedeća najbolja nezauzeta sekvenca, no to nije razmatrano jer predstavlja rubni slučaj osnovne funkcionalnosti odbacivanja sekvenci, što je već samo po sebi rubni slučaj i vrlo se rijetko dešava.

SLIKA CJEVOVODA (prikazuje podakte a ne filtre radi boljeg shvaćanja)

tekst o objektima

SLIKA objekata (class dijagram)

5.2. tax

reći da se pazi na scientific-name

Algorithm 1 Nalaženje najbolje ocijenjenih proteina

function PROTEINHOLDER::IZRAČUNAJNAJOCJENE*ocjene* \leftarrow *BestScoreCollection*()**for all** *fastaRecord* \in *this.records* **do****for all** *element* \in *fastaRecord.elementiZaglavlja* **do***ocjena* \leftarrow *BestScore*(*element*)*ocjene.auriraj*(*ocjena*, *fastaRecord.ocjena*)**end for****end for** *return ocjene***end function****function** PROTEINHOLDER::PROPAGIRAJ(*najOcjene*)**for all** *fastaRecord* \in *this.records* **do****for all** *element* \in *fastaRecord.elementiZaglavlja* **do***najOcjena* \leftarrow *najOcjene.dohvati*(*element*['gid'])**if** \neg *najOcjena.provjeriElement*(*element*) **then***element*['zastavicaNajbolji'] \leftarrow **false****end if****end for****end for****end function****function** PRONAĐINAJOCJENE(*proteinHolders*) \triangleright funkcija je isječak iz

cjevovoda

najOcjene \leftarrow *newBestScoreCollection*()**for all** *protein* \in *proteinHolders* **do***ocjene* \leftarrow *protein.izraunajNajOcjene*()*najOcjene.aurirajKolekcijom*(*ocjene*)**end for****for all** *protein* \in *proteinHolders* **do** \triangleright propagacija najboljih ocjena*protein.propagiraj*(*najOcjene*)**end for****end function**

5.3. server

dio po dio

jQuery

komunikacija pipelinea i klijenta, log, dekoratori

automatizacija (posla kroz cjevovod->treba ići u neko uvodno poglavlje o namjeri)

robusnost

6. Rezultati

7. Zaključak

LITERATURA

- [1] Andreas D. Baxevanis. *Bioinformatics and The Internet*. John Wiley & Sons, Inc., 2002. ISBN 9780471223924.

**Orthobalancer: aplikacija za kreiranje skupova bioloških vrsta usporedive
taksonomijske širine**

Sažetak

Ključne riječi:

**Orthobalancer: web application for creation of taxonomically balanced sets of
orthologous protein sequences**

Abstract

Keywords: