SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Mateja Vragović

STATISTIKA VIŠESTRUKOG PORAVNANJA

Diplomski rad

Voditelj rada:
doc. dr. sc. Pavle Goldstein

Zagreb, srpanj, 2015

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Veliko hvala mentoru doc. dr. sc. Pavlu Golsteinu na ukazanom povjerenju, strpljenju, korisnim savjetima i smjernicama. Također, hvala mojoj obitelji i prijateljima na podršci i razumijevanju.

Sadržaj

Uvod

Bioinformatika je najmlađa znanost koja je nastala spajanjem informatike i biologije, a bavi se analizom bioloških nizova uz pomoć tehnika iz matematike, statistike i računarstva. Neke od tema ovog područja su poravnanje nizova proteina ili nukleinskih kiselina, predviđanje strukture proteina, analiza genskih, proteinskih i metaboličkih mreža i slično.

Od navedenih područja usredotočiti ćemo se na višestruko poravnanje nizova proteina. Pod pojmom višestrukog poravnanja podrazumijeva se poravnanje tri ili više nizova proteina. Konkretno, zanima nas kako opisati profil familije već poravnatih nizova proteina, da bi se kasnije s njom mogli poravnati neki novi nizovi ili ustanoviti pripadnost familiji.

U ovom diplomskom radu, nakon osnovnih matematičkih definicija i biološkog uvoda, slijedi opis skrivenog Markovljevog modela (to je profil familije). Nakon toga opisujemo metode i algoritme koji su korišteni za analize: Viterbijev algoritam za poravnanje nizova, Forward algoritam za računanje ukupne vjerojatnosti niza u odnosu na model, procjena parametara proširenog skrivenog Markovljevog modela i simulacija nove familije proteina. Ideja je na temelju poravnate familije procijeniti parametre i ponovno poravnati familiju proteina. Želimo, na temelju statističkih podataka, koje ćemo dobiti Forward algoritmom ocijeniti model i pokušati bolje procijeniti parametre simulacijom nove familije za dani model. Očekujemo da će model biti osjetljiv na promjene koliko god one bile male. Na kraju slijede rezultati i zaključak.

Poglavlje 1

Teorija vjerojatnosti

1.1 Uvod u vjerojatnost i statistiku

Definicija 1.1.1. ***Slučajni pokus*** *ili* ***slučajni eksperiment*** *takav je pokus čiji ishodi, tj. rezultati nisu jednoznačno određeni uvjetima u kojima izvodimo pokus.*

Kod svakog slučajnog pokusa osnovno je da se ustanovi odnos između uzorka i posljedice. Poznavanje tog odnosa omogućuje definiranje **uvjeta pokusa** i predviđanje **ishoda** pri svakom realiziranju pokusa. Najčešći primjer slučajnog pokusa bacanja je igraće (simetrične) kocke.

Definicija 1.1.2. ***Prostor elementarnih događaja*** *neprazan je skup* Ω *koji reprezentira skup svih ishoda slučajnog pokusa. Elemente od* Ω *označavamo sa* ω *i zovemo* ***elementarni događaji****.*

Definicija 1.1.3. *Familija* $\mathcal{A}$ *podskupova od* Ω *jest* ***algebra skupova*** *(na* Ω*) ako je:*

1. $\emptyset \in \mathcal{A}$
2. $A \in \mathcal{A} \Longrightarrow A^c \in \mathcal{A}$
3. $A_1, A_2, \ldots, A_n \in \mathcal{A} \Longrightarrow \bigcup_{i=1}^{n} A_i \in \mathcal{A}$

Definicija 1.1.4. *Familija* $\mathcal{F}$ *podskupova od* Ω *(*$\mathcal{F} \subset \mathcal{P}(\Omega)$*) je* $\sigma-$ ***algebra skupova*** *na* Ω *ako je :*

1. $\emptyset \in \mathcal{A}$
2. $A \in \mathcal{A} \Longrightarrow A^c \in \mathcal{A}$
3. $A_i \in \mathcal{A} \Longrightarrow \bigcup_{i=1}^{n} A_i \in \mathcal{A}$

Definicija 1.1.5. *Neka je $\mathcal{F}$ $\sigma-$ algebra na skupu Ω. Uređeni par $(\Omega, \mathcal{F})$ zove se izmjeriv prostor.*

Definicija 1.1.6. *Neka je $(\Omega, \mathcal{F})$ izmjeriv prostor. Funkcija $P\colon \mathcal{F} \to \mathbb{R}$ je **vjerojatnost** (na $\mathcal{F}$, na Ω) ako vrijedi:*

1. *$P(A) \leq 0$, $A \in \mathcal{F}$ (nenegativnost)*
2. *$P(\Omega) = 1$ (normiranost)*
3. *Za svaki niz $(A_n,\ n \in \mathbb{N})$, $A_n \in \mathcal{F}$, takav da je $A_n \cap A_m = \emptyset$ za $m \neq n$, vrijedi*

$$\mathbb{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbb{P}(A_n) \text{ } (\sigma - \textit{aditivnost ili prebrojiva aditivnost})$$

Definicija 1.1.7. *Uređena trojka $(\Omega, \mathcal{F}, \mathbb{P})$, gdje je $\mathbb{F}$ σ-algebra na Ω i $\mathbb{P}$ vjerojatnost na $\mathbb{F}$, zove se **vjerojatnosni prostor.***

Ako je Ω konačan ili prebrojiv skup, onda $(\Omega, \mathcal{F}, \mathbb{P})$ zovemo **diskretni vjerojatnosni prostor**.

Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor. Elemente $\sigma-$algebre $\mathcal{F}$ zovemo **događaji**, a broj $\mathbb{P}(A)$, $A \in \mathcal{F}$ zove se **vjerojatnost događaja A**.

Definicija 1.1.8. *Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ proizvoljan vjerojatnosni prostor i $A \in \mathcal{F}$ takav da je $P(A) > 0$. Definiramo funkciju $\mathbb{P}_A\colon \mathcal{F} \to [0, 1]$:*

$$\mathbb{P}_A(B) = \mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)},\ B \in \mathcal{F}. \tag{1.1}$$

*$\mathbb{P}_A$ je vjerojatnost na $\mathcal{F}$ i zovemo je **uvjetna vjerojatnost uz uvjet A**. Broj $\mathbb{P}(B \mid A)$ zovemo **vjerojatnost od B uz uvjet A.***

Definicija 1.1.9. *Konačna ili prebrojiva familija $(H_i,\ i = 1, 2, \ldots)$ događaja u vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ jest **potpun sistem događaja** ako je $H_i \neq 0$ za svako i, $H_i \cap H_j = \emptyset$ za $i \neq j$ (tj. događaji se uzajamno isključuju) i $\bigcup_i H_i = \Omega$*

Drugim riječima, potpun sistem događaja konačna je ili prebrojiva particija skupa Ω s tim da su elementi particije događaji.

Teorem 1.1.10. *(Formula potpune vjerojatnosti) Neka je $(H_i,\ i = 1, 2, \ldots)$ potpun sistem događaja u vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$. Tada za proizvoljno $A \in \mathcal{F}$ vrijedi*

$$\mathbb{P}(A) = \sum_i \mathbb{P}(H_i)\mathbb{P}(A \mid H_i) \tag{1.2}$$

Teorem 1.1.11. *(Bayesova formula) Neka je* $(H_i,\ i = 1, 2, \ldots)$ *potpun sistem događaja u vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$ *i* $A \in \mathcal{F}$ *takav da je* $\mathbb{P}(A) > 0$*. Tada za svako i vrijedi*

$$\mathbb{P}(H_i \mid A) = \frac{\mathbb{P}(H_i)\mathbb{P}(A \mid H_i)}{\sum_j \mathbb{P}(H_j)\mathbb{P}(A \mid H_j)} \tag{1.3}$$

Definicija 1.1.12. *Neka je* $(\Omega, \mathcal{F}, \mathbb{P})$ *vjerojatnosni prostor i* $A, B \in \mathcal{F}$*. Događaji A i B su* ***nezavisni*** *ako vrijedi*

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B) \tag{1.4}$$

Iz ove definicije slijedi da ako je $A \in \mathcal{F}$ takav da je $\mathbb{P}(A) = 0$ tada su A i B nezavisni događaji za svako $B \in \mathcal{F}$. Također Ω i B su nezavisni događaji za svako $B \in \mathcal{F}$.

Definicija 1.1.13. *Neka je* $(\Omega, \mathcal{F}, \mathbb{P})$ *vjerojatnosni prostor i* $A_i \in \mathcal{F}$, $i \in I$ *proizvoljna familija događaja. Kažemo da je to* ***familija nezavisnih događaja*** *ako za svaki konačan podskup indekasa* $i_1, i_2, \ldots, i_k \in I$ *vrijedi*

$$\mathbb{P}\left(\bigcap_{j=1}^{k} A_{i_j}\right) = \prod_{j=1}^{k} \mathbb{P}(A_{i_j}) \tag{1.5}$$

Ako je $\mathcal{A}$ neka familija nezavisnih događaja, tada je očigledno svaka potfamilija od $\mathcal{A}$ također familija nezavisnih događaja.

1.1.1 Slučajna varijabla

Neka je $\mathbb{R}$ skup realnih brojeva. Sa $\mathcal{B}$ označimo σ-algebru generiranu familijom otvorenih skupova u $\mathbb{R}$. $\mathcal{B}$ zovemo **σ-algebra Borelovih skupova na** $\mathbb{R}$, a elemente σ-algebre $\mathcal{B}$ zovemo **Borelovi skupovi**.

Definicija 1.1.14. *Neka je* $(\Omega, \mathcal{F}, \mathbb{P})$ *vjerojatnosni prostor. Funkcija* $X : \Omega \to \mathbb{R}$ *jest* ***slučajna varijabla*** *(na* Ω*) ako je* $X^{-1}(B) \in \mathcal{F}$ *za proizvoljno* $B \in \mathcal{B}$*, tj.* $X^{-1}(B) \subset \mathcal{F}$*.*

Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i X slučajna varijabla na Ω. Za $B \in \mathcal{B}$ stavimo

$$\mathbb{P}(B) = \mathbb{P}(X^{-1}(B)) = \mathbb{P}\{\omega \in \Omega\,;\, X(\omega) \in B\} = \mathbb{P}\{X \in B\}. \tag{1.6}$$

Relacijom 1.6 definirana je funkcija $\mathbb{P}_X \colon \mathcal{B} \to [0, 1]$ koja je vjerojatnost, odnosno vjerojatnosna mjera na $\mathcal{B}$. $\mathbb{P}_X$ zovemo **vjerojatnosna mjera inducirana sa X**, a vjerojatnosni prostor $(\mathbb{R}, \mathcal{B}, \mathbb{P}_X)$ zovemo **vjerojatnosni prostor induciran sa X**. $\mathbb{P}_X$ često zovemo i **zakon razdiobe od X**.

Način na koji računamo vjerojatnosti, što ćemo vidjeti kasnije, ovisi o tipu slučajne varijable. U teoriji vjerojatnosti postoje dva glavna tipa : diskretne i neprekidne slučajne varijable.

Definicija 1.1.15. *Slučajna varijabla X je* ***diskretna*** *ako postoji konačan ili prebrojiv skup* $D \in \mathbb{R}$ *takav da je* $\mathbb{P}\{X \in D\} = 1$.

Iz navedene definicije slijedi da, ako je X slučajna varijabla takva da je skup svih vrijednosti od X konačan ili prebrojiv, tada je X diskretna. Odavde sada imamo da ako je $(\Omega, \mathcal{F}, \mathbb{P})$ diskretni vjerojatnosni prostor, tada je svaka realna funkcija na Ω diskretna slučajna varijabla.

Definiciju neprekidne slučajne varijable dat ćemo u idućem poglavlju, nakon što damo definiciju funkcije distribucije.

1.1.2 Funkcija distribucije

Definicija 1.1.16. *Neka je X slučajna varijabla na* Ω. ***Funkcija distribucije od X*** *jest funkcija* $F_X \colon \mathbb{R} \to [0, 1]$ *definirana sa*

$$F_X(x) = \mathbb{P}_X(< -\infty, x]) = \mathbb{P}(X^{-1}\langle -\infty, x]) = \mathbb{P}\{\omega \in \Omega\,;\, X(\omega) \leq x\} = \mathbb{P}\{X \leq x\},\ x \in \mathbb{R} \quad (1.7)$$

Stavljat ćemo da je $F_X = F$, ako je jasno o kojoj se slučajnoj varijabli, odnosno funkciji distribuciji radi.

Definicija 1.1.17. *Neka je X slučajna varijabla na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$ *i neka je* F_X *njezina funkcija distribucije. Kažemo da je X* ***apsolutno neprekidna*** *ili, kraće,* ***neprekidna slučajna varijabla*** *ako postoji nenegativna realna Borelova funkcija* f *na* $\mathbb{R}$ $(f \colon \mathbb{R} \to \mathbb{R}_+)$ *takva da je*

$$F_X(x) = \int_{-\infty}^{x} f(t)\, d\lambda(t),\ x \in \mathbb{R}. \quad (1.8)$$

Integral u 1.8 Lebesgueov je integral funkcije f u odnosu na Lebesgueovu mjeru λ. Ako je X neprekidna slučajna varijabla, tada se funkcija f iz 1.8 zove **funkcija gustoće vjerojatnosti od X**, tj. od njezine funkcije distribucije F_X ili, kraće **gustoća od X**, ponekad je i označujemo sa f_X.

Teorem 1.1.18. *Funkcija distribucije F slučajne varijable X je rastuća i neprekidna zdesna na* $\mathbb{R}$ *i zadovoljava*

$$F(-\infty) = lim_{x \to -\infty} F(x) = 0$$
$$F(+\infty) = lim_{x \to +\infty} F(x) = 1$$

Funkciju iz teorema 1.1.18 zovemo **vjerojatnosna funkcija distribucije (na** $\mathbb{R}$**)**.

1.2 Matematičko očekivanje

Definicija 1.2.1. *Neka je X slučajna varijabla na* $(\Omega, \mathcal{F}, \mathbb{P})$*. X je* ***jednostavna slučajna varijabla*** *ako je njezino područje vrijednosti konačan skup.*

Definicija matematičkog očekivanja provodi se u tri koraka. Prvo za jednostavne slučajne varijable, zatim za nenegativne slučajne varijable i na kraju za opće slučajne varijable.

Označimo sa $\mathcal{K}$ skup svih jednostavnih slučajnih varijabli definiranih na Ω, a sa $\mathcal{K}_+$ skup svih nenegativnih funkcija iz $\mathcal{K}$. Neka je $X \in \mathcal{K}$, $X = \sum_{k=1}^{n} x_k K_{A_k}$, gdje su $A_1, \ldots, A_n \in \mathcal{F}$ međusobno disjunktni.

Definicija 1.2.2. ***Matematičko očekivanje od X*** *ili, kraće,* ***očekivanje od X*** *koje označujemo sa EX definira se sa*

$$EX = \sum_{k=1}^{n} x_k P A_k$$

Neka je X nenegativna slučajna varijabla definirana na Ω. Tada postoji rastući niz $(X_n,\ n \in \mathbb{N})$ nenegativnih jednostavnih slučajni varijabli takav da je $X = \lim_{n\to\infty} X_n$. Niz $(EX_n,\ n \in \mathbb{N})$ rastući niz u $\mathbb{R}_+$, pa postoji $\lim_{n\to\infty} EX_n$ koji može biti jednak i $+\infty$.

Definicija 1.2.3. ***Matematičko očekivanje od X*** *ili, kraće,* ***očekivanje od X*** *definira se sa*

$$EX = \lim_{n\to\infty} EX_n\,.$$

Neka je sada X proizvoljna slučajna varijabla na Ω. Vrijedi $X = X^+ - X^-$, za X^+ i X^- su slučajne varijable i $X^+,\ X^- \geq 0$.

Definicija 1.2.4. *Kažemo da* ***matematičko očekivanje od*** *X ili, kraće,* ***očekivanje od*** *X,* ***postoji*** *ili da je* ***definirano*** *ako je barem jedna od veličina* EX^+ *ili* EX^- *konačna, tj. vrijedi*

$$\min\{EX^+,\ EX^-\} < \infty\,.$$

Tada po definiciji stavljamo

$$EX = EX^+ - EX^-\,.$$

1.3 Varijanca

Neka je X slučajna varijabla na $(\Omega, \mathcal{F}, \mathbb{P})$ i $r > 0$.

Definicija 1.3.1. $E[\,X^r\,]$ *zovemo* ***r-ti moment od*** *X,* $E[\,|X|^r\,]$ *zovemo* ***r-ti apsolutni moment od*** *X.*

Po dogovoru stavljamo da je $E[\,X^0\,] = E[\,|X|^0\,] = 1$.

Definicija 1.3.2. *Neka je EX postoji (tj. konačno je). Tada $E[\,(X - EX)^r\,]$ zovemo* ***r-ti centralni moment*** *X, a $E[\,|X - EX|^r\,]$ zovemo* ***r-ti apsolutni centralni moment*** *X*

Definicija 1.3.3. ***Varijanca od*** *X koju označavamo sa $Var\,X$ ili σ_X^2 jest drugi centralni moment od X, dakle je*

$$VarX = E[(X - EX)^2]\,.$$

Pozitivan drugi korijen iz varijance zovemo **standardna devijacija od** X i označavamo sa σ_X.

1.4 Normalna distribucija

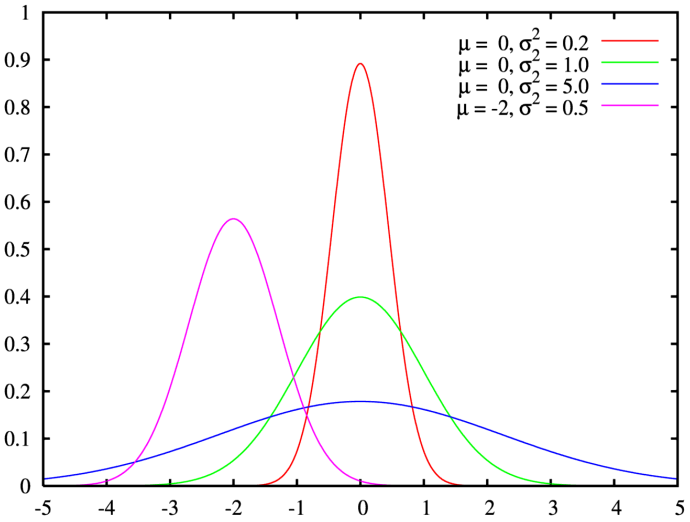
Definicija 1.4.1. *Neka su $\mu, \sigma \in \mathbb{R}$, $\sigma > 0$. Neprekidna slučajna varijabla X ima* ***normalnu distribuciju s parametrima*** *μ i σ^2 ako joj je gustoća f dana s*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},\ x \in \mathbb{R}. \tag{1.9}$$

To ćemo označavati sa $X \sim N(\mu, \sigma^2)$

Očekivanje i varijanca normalne distribucije: $EX = \mu$ i $VarX = \sigma^2$.

X je ***jedinična normalna distribucija*** *ako je $X \sim N(0, 1)$, dakle $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, $x \in \mathbb{R}$*



Slika 1.1: Normalna distribucija za različite parametre

Poglavlje 2

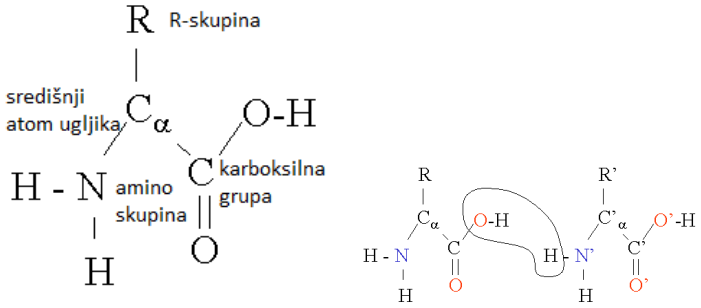
Proteini

Proteini ili ***bjelačevine*** su organski spojevi. Protein je niz aminokiselina koje su međusobno povezane peptidnom vezom.

2.1 Aminokiseline

Aminokiselina (eng. *aminoacid*) je organski spoj koji se sastoji od središnjeg atoma ugljika (C_α atom) te na njega vezanih atoma vodika, amino skupine, karboksilne skupine i bočnog lanca. (Slika 2.1a)

Dvije aminokiseline mogu se međusobno vezati tako da reakcijom između karboksilne skupine jedne i amino skupine druge aminokiseline nastane *peptid*. Veza između njih zove se *peptidna veza*, (Slika 2.1b) , u kojoj se atom ugljika veže uz atom dušika uz oslobađanje molekule vode.



(a) Struktura aminokiselina (b) Peptidna veza

Slika 2.1

U standardni sastav ulazi ukupno 20 aminokiselina. Redoslijed tih aminokiselina u proteinu određuje funkciju proteina. Što povlači činjenicu, čim se promijeni redoslijed aminokiselina u lancu (nizu), lanac dobije nove karakteristike. Kemijska svojstva aminokiseline ovise o bočnom lancu (radikalu, R), pa ih onda i dijelimo s obzirom na tu R-skupinu na : polarne i nepolarne, bazične i nebazične. Za naziv aminokiseline obično se koriste skraćenice od jednog ili tri slova, ali ovdje ćemo koristiti samo jedno slovo (simbol), zbog lakšeg rada s podacima. Naš alfabet glasi ovako:

$$\mathcal{A} = \{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}. \tag{2.1}$$

Nazivi aminokiselina, skraćenice, simboli koje ćemo koristiti, te kojoj skupini pripada možemo vidjeti u tablici 2.1.

Tablica 2.1: Podjela aminokiselina

nepolarne-narančasta, polarne-zelena, bazične-plava, nebazične(kisele)-crvena

Aminokiselina	**Kratice**	
Alanin	Ala	A
Arginin	Arg	R
Asparagin	Asn	N
Aspartat	Asp	D
Cistein	Cys	C
Glutamin	Gln	Q
Glutamat	Glu	E
Glicin	Gly	G
Histidin	His	H
Izoleucin	Ile	I
Leucin	Leu	L
Lizin	Lys	K
Metionin	Met	M
Fenilalanin	Phe	F
Prolin	Pro	P
Serin	Ser	S
Treonin	Thr	T
Triptofan	Trp	W
Tirozin	Tyr	Y
Valin	Val	V

2.2 Struktura proteina

Niz više aminokiselina povezanih peptidnim vezama je polipeptid, odnosno protein. Postoji nekoliko struktura proteina i to su: primarna, sekundarna, tercijarna i kvartarna struktura.

Primarna struktura proteina (eng. *primary structure*, 1D) je točan niz aminokiselina u proteinu.

Sekundarna struktura proteina (eng. *secondary structure*, 2D) je prostorna organizacija aminokiselina koje su blizu u primarnoj strukturu. Elementi sekundarne strukture su $\alpha-$**zavojnice** (eng. α-helix), β**-ploče** (eng. β-sheet) i **okreti** (eng. turn).

- U $\alpha-$zavojnici polipeptidni lanac se zavija u smjeru kazaljke na satu (desni navoj) i stvara strukturu čvrsto upakiranog valjka. Iako postoje i desni i lijevi navoji, lijevi su energetski nepovoljniji, pa su gotovo sve $\alpha-$zavojnice pronađene u proteinima desnog navoja.

- U $\beta-$lancu, polipeptidni lanac je gotovo sasvim izdužen. Dva ili više $\beta-$lanca koji su povezani vodikovim vezama stvaraju $\beta-$nabranu ploču, koju kraće zovemo $\beta-$ploča. Ti lanci mogu biti *paralelni* ili *antiparalelni* ili mogu biti *kombinacija* paralelnih i antiparalelnih. Iako $\beta-$ploče mogu biti ravne, većina $\beta-$ploča prirodno zavija udesno.

- $\beta-$zavoji i petlje polipeptidnog lanca nalaze se na površini strukture proteina i često reagiraju s drugim molekulama ili proteinima.

Tercijarna struktura (eng. *tertiary structure*, 3D) je prostorna struktura svih atoma jednog polipeptidnog lanca. Struktura nastala interakcijama među aminokiselinama koje nisu blizu u primarnoj strukturi. Nakon što protein poprimi specifičnu tercijarnu strukturu tek tada može obavljati svoju funkciju. Stoga poznavanje tercijarne strukture proteina je važno za proučavanje njegove evolucije i funkcije. Predviđanje tercijarne strukture naziva se predikcija strukture proteina, poznatije kao "protein folding problem".

Kvartarna struktura (eng. *quaternary structure*, 4D) proteina je spajanje više polipeptida u jednu strukturnu jedinicu. Dakle, razlika u odnosu na tercijarnu strukturu je ta što ovdje imamo više polipeptidnih lanaca.

Na slici 2.2 možemo vidjeti strukturu proteina.

2.3 Evolucija proteina

Proteini su zaslužni za gotovo sve funkcije koje su potrebne stanicama, te bilo kakva promjena na njima, mijenja njihov oblik i funkciju. Proteini ne nastaju *de novo* već su posljedica evolucije postojećih proteina, stoga uvodimo pojam familije proteina. Konkretno,

familija proteina definira se kao skup proteina koji potječu od istog pretka. Pod evolucijom ćemo promatrati mutacije događaja na slučajnom mjestu u proteinskom nizu. Iako postoji više mutacijskih događaja, ovdje ćemo se bazirati na slijedećim:

- **insercija** - ubacivanje jedne ili više aminokiselina
- **delecija** - izostavljanje jedne ili više aminokiselina
- **supstiticija** - zamjena jedne aminokiseline drugom

Kako bismo označili mjesto delecije ili insercije u nizu, uvodimo simbol $'-'$ koji zovemo prazninom (eng. gap), te njime označavamo slučaj kada simbol nije pridružen niti jednom simbolu drugog niza. Pri čemu slučaj pridruživanja praznine simbolu zovemo *delecija*, a slučaj pridruživanja simbola praznini *insercija*. Objasnimo to na primjeru.

Primjer 2.3.1. *Neka je zadan konsenzusni niz **HPEW**. Konsenzusni niz je predak iz kojeg su mutacijom na slučajnim mjestima nastali novi nizovi. Neka su mutacijama nastali slijedeći nizovi:*

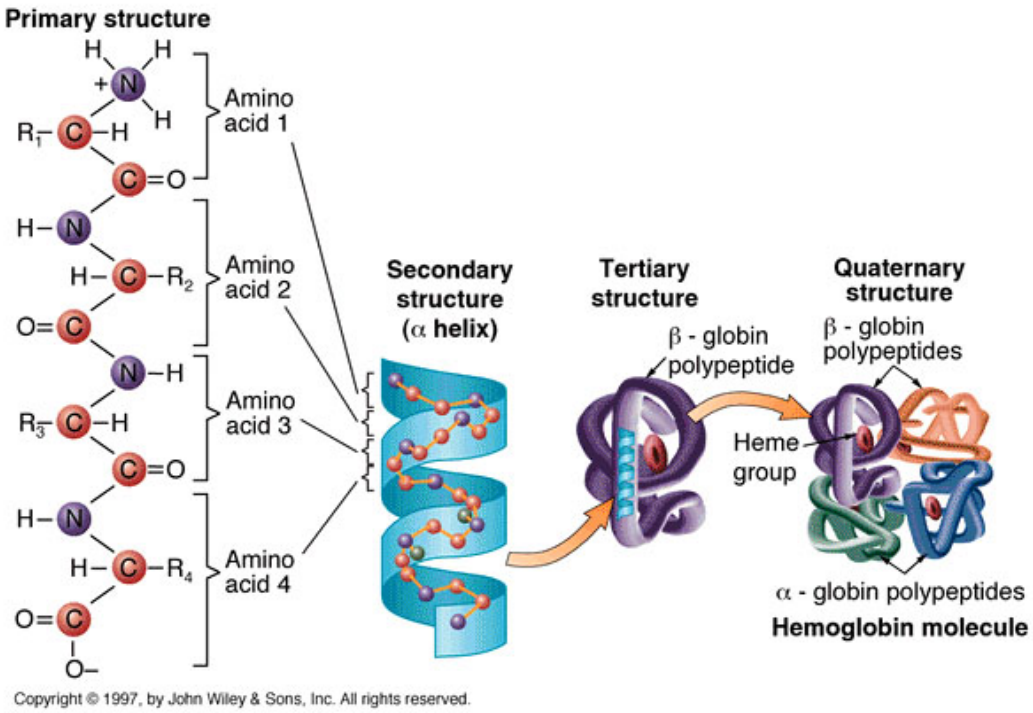
PW - aminokiseline H i E su izbačene iz niza (delecija)

HPAW - aminikiselina E zamijenjena je aminokiselinom A (supstitucija)

HPEWL - aminokiselina L dodana je na kraj niza (insercija)

Kada je poznata primarna struktura pretka onda ove nizove možemo poravnati. Poravnanje je slijedeće:

```
- P - W -
H P A W -
H P E W L
```



Slika 2.2: Strukture proteina

Poglavlje 3

Skriveni Markovljev model-HMM

3.1 Uvjetno matematičko očekivanje

Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor i neka je $A \in \mathcal{F}$ takav da je $P(A) > 0$. U poglavlju 1 definirali smo uvjetnu vjerojatnost formulom 1.1 koja je potrebna za definiciju uvjetnog matematičkog očekivanja.

Ako je X slučajna varijabla na Ω takva da postoji EX, tada je X očigledno integrabilna i u odnosu na P_A i **uvjetno očekivanje od X za dano A** definiramo sa

$$E(X \mid A) = \int_{\Omega} X \, dP_X.$$

Ako iskoristim definiciju 1.1 u prethodnoj jednakosti dobivamo slijedeće

$$E(X \mid A) = \frac{1}{P(A)} \int_{\Omega} X \, dP = \frac{1}{P(A)} E(XK_A) \tag{3.1}$$

Ako u 3.1 stavimo da je $X = K_B$, $B \in \mathcal{F}$, dobijemo

$$E(K_B \mid A) = \frac{P(A \cap B)}{P(A)} = P(B \mid A). \tag{3.2}$$

3.2 Markovljev lanac

Definicija 3.2.1. *Neka je S skup. Slučajan proces s diskretnim vremenom i prostorom stanja S je familija $X = (X_n : n \geq 0)$ slučajnih varijabli(ili elemenata) definiranih na nekom vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ s vrijednostima i S. Dakle, za svaki $n \geq 0$, je $X_n : \Omega \to S$ slučajna varijabla.*

Definicija 3.2.2. *Neka je S prebrojiv skup. Slučajni proces* $X = (X_n : n \geq 0)$ *definiran na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$ *s vrijednostima u skupu S je Markovljev lanac(eng. Markov chains) ako vrijedi*

$$\mathbb{P}(X_{n+1} = j \,|\, X_n = i,\, X_{n-1} = i_{n-1},\, \ldots,\, X_0 = i_0) = \mathbb{P}(X_{n+1} = j \,|\, X_n = i) \tag{3.3}$$

za svaki $n \geq 0$ *i za sve* $i_0, \ldots, i_{n-1}, i, j \in S$ *za koje su obje uvjetne vjerojatnosti dobro definirane.*

Svojstvo u relaciji 3.3 naziva se **Markovljevo svojstvo**.

Pretpostavimo da se nalazimo u trenutku n. Tada vrijeme $n+1$ predstavlja neposrednu budućnost, a 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. Drugi način iskaza je

$$\begin{aligned}&\mathbb{P}(X_{n+1} = j,\, X_{n-1} = i_{n-1},\, \ldots,\, X_0 = i_0 \,|\, X_n = i) = \\ &= \mathbb{P}(X_{n+1} = j \,|\, X_n = i)\mathbb{P}(X_{n-1} = i_{n-1},\, \ldots,\, X_0 = i_0 \,|\, X_n = i)\end{aligned}$$

3.3 Skriveni Markovljev model

Definicija 3.3.1. ***Skriveni Markovljev model*** *(eng. Hidden Markov Model, HMM) zadajemo sa dva niza,* π *i* x

- $\pi = \pi_1, \pi_2, \ldots, \pi_N$ *- niz slučajnih varijabli koje poprimaju diskretne vrijednosti*
- $x = x_1, x_2, \ldots, x_N$ *- niz slučajnih varijabli koje poprimaju diskretne ili kontinuirane vrijednosti*

pri čemu te varijable zadovoljavaju slijedeće uvjete

$$\mathbb{P}(\pi_t \,|\, \pi_{t-1},\, x_{t-1},\, \ldots,\, \pi_1,\, x_1) = \mathbb{P}(\pi_t \,|\, \pi_{t-1}) \tag{3.4}$$

$$\mathbb{P}(\pi_t \,|\, \pi_N,\, x_N,\, \pi_{N-1},\, x_{N-1},\, \ldots,\, \pi_{t+1},\, x_{t+1},\, \pi_t,\, x_t,\, \pi_{t-1},\, x_{t-1},\, \ldots,\, \pi_1,\, x_1) = \mathbb{P}(\pi_t \,|\, \pi_{t-1}) \tag{3.5}$$

Skriveni Markovljev model možemo zadati slijedećim parametrima :

- M − broj mogućih opažanja

$$B = \{b_1, b_2, \ldots, b_M\}\,, \text{ gdje je B - skup svih opažanja}$$

- L – duljina opaženog niza

$$x = (x_1, x_2, \ldots, x_L)\,, \text{ gdje je x - opaženi niz}$$

- N – broj stanja u kojima se proces može nalaziti

$$S = \{1, 2, \ldots, N\}\,, \text{ gdje je S - skup svih stanja procesa}$$

- tranzicijske vjerojatnosti
- emisijske vjerojatnosti

Budući da ćemo promatrati aminokiseline, onda ćemo u tom kontekstu i definirati navedene parametre.

3.3.1 Alfabet modela

Alfabet modela sastoji se od 20 standardnih aminokiselina (vidi niz 2.1), pa je onda broj mogućih opažanja jednak 20. Oznaka :

$$\sum = b_1, b_2, \ldots, b_{20}\,.$$

3.3.2 Tipovi stanja

Neka je $x = (x_1, x_2, \ldots, x_L)$ opaženi niz aminokiselina. Stanja u kojima se proces može nalaziti označavamo sa π_i. Model se sastoji od tri tipa stanja : *match, insert i delete* stanje. *Match* stanja označena su kvadratićima i predstavljaju konzervirane stupce u poravnanju (što je poravnanje vidjeti ćemo kasnije), *insert* stanja označena su rombovima i predstavljaju insercije, a *delete* stanja su označena krugovima i predstavljaju delecije u primarnoj strukturi. Osim navedenih stanja još postoje stanja koja označavaju početak i kraj modela i to označavamo sa *begin i end*. Ukoliko model ima *m match* stanja,tada on ima $m + 1$ *insert* stanja, te *m delete* stanja. To znači da je $3m + 3$ broj stanja u kojima se proces može nalaziti. Duljina modela zapravo je broj *match* stanja. Prolaz kroz model označavamo sa π i on počinje sa *begin* stanjem, te dopuštenim tranzicijama dolazi do *end* stanja. Prolaz niza kroz model nije jedinstven, te onda ni prolazi nisu jednako vjerojatni. Uvedimo oznake za stanja.

- *Match* stanja : $M_1, M_2, \ldots, M_m$
- *Insert* stanja : $I_0, I_1, \ldots, I_m$
- *Delete* stanja : $D_1, D_2, \ldots, D_m$
- *Begin* i *end* : B, E

3.3.3 Emisijske vjerojatnosti

Match i *insert* stanja su stanja koja emitiraju simbole te pri prolasku niza kroz model definiramo vjerojatnost kojom će se neka aminokiselina iz alfabeta emitirati ovisno o tome u kojem se stanju model nalazi. Uvedimo oznaku:

$$e_k(b_i) = \mathbb{P}(x_i = b_i \,|\, \pi_i = k) \text{ vjerojatnost da stanje k emitira simbol } b_i. \tag{3.6}$$

Stanje k može emitirati bilo koji simbol iz alfabeta, pa za stanje k dobivamo neki vektor dimenzije 20. Nastali vektor emisija je vjerojatnosni vektor i za svako stanje k vrijedi $\sum_{i=1}^{20} e_k(b_i) = 1$. Takav vjerojatnosni vektor naziva se distribucijom aminokiseline u nekom stanju. Vektori za *match* stanje procjenjuju se iz nekog uzorka za treniranje, dok vektori za *insert* stanja imaju prosječnu distribuciju aminokiselina (vjerojatnost emisije za bilo koje slovo alfabeta u bilo kojem insert stanju je $\frac{1}{20}$). *Delete* stanja, *begin* i *end* stanja su stanja koje ne emitiraju simbole (šutljiva stanja).

3.3.4 Tranzicijske vjerojatnosti

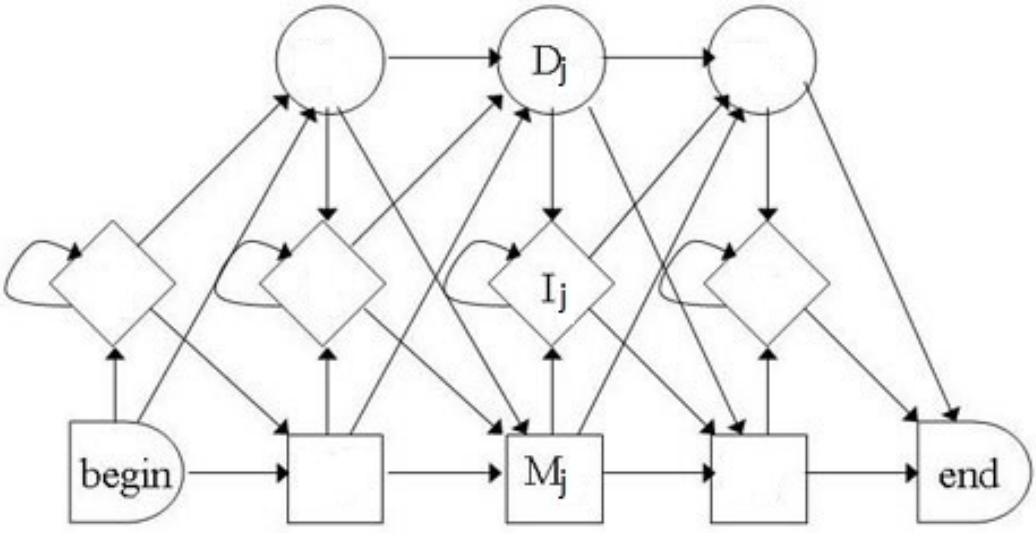
Tranzicijske vjerojatnosti su vjerojatnosti prelaska iz stanje u stanje. No pri tome moramo paziti jer ne možemo baš iz svakog stanja u svako stanje već postoji pravilo. Uvedimo oznaku:

$$a_{kl} = \mathbb{P}(\pi_i = l \,|\, \pi_{i-1} = k) \text{ vjerojatnost prelaska iz stanja k u stanje l} \tag{3.7}$$

Također vrijedi da je suma vjerojatnosti prelaska iz stanja u tri moguća stanja uvijek 1. Neka je m duljina modela i neka se nalazimo u trenutku t, $0 < t < m$. Mogućnosti prelaska su slijedeće :

* iz B ili I_0 možemo u M_1, I_0 ili D_1
* iz E ne možemo više nikud kad je to završno stanje
* iz M_t, I_t ili D_t tada možemo u M_{t+1}, I_t, D_{t+1}
* iz M_m, I_m ili D_m tada možemo samo u I_m, E

Opisanu strukturu možemo vidjeti na slici 3.1.



Slika 3.1: Skriveni Markovljev model (profil familije)

Poglavlje 4

Algoritmi

Put nekog niza kroz model nije jedinstven. Prelasci iz stanja u stanje nisu jednako vjerojatni, a niti vjerojatnosti da se u nekom stanju emitira neki simbol. Da bismo uopće mogli govoriti o bilo kakvom prolasku nekog niza kroz model, moramo imati zadani model. Budući da nemamo zadani model moramo ga prvo procijeniti. Jedino što imamo zadano je familija poravnatih nizova. Što je poravnanje? Kako procijeniti model? Kako dobiti najbolji put nekog niza kroz model? To su neka od pitanja na koja ćemo odgovoriti u ovom poglavlju.

Neka je $x = x_1 x_2 \dots x_n$ proizvoljan niz, te neka su emisijske i tranzicijske vjerojatnosti dane oznakama 3.6 i 3.7. Neka su *match*, *insert* i *delete* stanja označena slovima M, I i D, te neka su sva stanja u modelu označena kao u poglavlju 3.3.2.

Svi algoritmi pisani su u programskom jeziku *Python*, a sve statističke analize provedene u *R*-u.

4.1 Viterbijev algoritam

Viterbijev algoritam je algoritam dinamičkog programiranja koji pronalazi optimalan prolaz niza kroz model. Tražimo

$$\pi^* = arg\max_{\pi} \mathbb{P}(x, \pi) = arg\max_{x} \mathbb{P}(\pi \mid x) \tag{4.1}$$

gdje $\mathbb{P}(x, \pi)$ definiramo kao

$$\mathbb{P}(x, \pi) = a_{0\pi_1} \prod_{i=1}^{n} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}} \tag{4.2}$$

pri čemu za modeliranje početka i kraja stavimo da je $a_{0\pi_1} = a_{\pi_i \pi_{i+1}} = 1$.

Označimo sa $v_k(i)$ vjerojatnost najvjerojatnijeg prolaza π^* koji završava u stanju k pri čemu su emitirani simboli $x_1, x_2, \ldots, x_i$. Niz stanja π^* dobivamo rekurzijom i nazivamo ga *Viterbijev put*. Algoritam glasi ovako:

$$V_j^M(i) = log\frac{e_{M_j}(x_i)}{q_{x_i}} + max\begin{cases} V_{j-1}^M(i-1) + log\, a_{M_{j-1}M_j} \\ V_{j-1}^I(i-1) + log\, a_{I_{j-1}M_j} \\ V_{j-1}^D(i-1) + log\, a_{D_{j-1}M_j} \end{cases}$$

$$V_j^I(i) = log\frac{e_{I_j}(x_i)}{q_{x_i}} + max\begin{cases} V_j^M(i-1) + log\, a_{M_jI_j} \\ V_j^I(i-1) + log\, a_{I_jI_j} \\ V_j^D(i-1) + log\, a_{D_jI_j} \end{cases}$$

$$V_j^D(i) = max\begin{cases} V_{j-1}^M(i) + log\, a_{M_{j-1}D_j} \\ V_{j-1}^I(i) + log\, a_{I_{j-1}D_j} \\ V_{j-1}^D(i) + log\, a_{D_{j-1}D_j} \end{cases}$$

U slučaju kada bi produkt emisijske i tranzicijske vjerojatnosti bio 0.1, vjerojatnost Viterbijevog puta bi bila $10^{-100000}$, a tada bi se javio problem rada računala s malim vrijednostima, tj. *underflow*, a da izbjegnemo takav problem uvodimo logaritamsku transformaciju koju ćemo kraće zvati *log space*. Ona neće mijenjati vjerojatnosti, jer je logaritam produkata jednak zbroju logaritama. Budući da logaritam po definiciji ima bazu 10, znači da će vjerojatnost Viterbijevog puta biti -100000, a takve vrijednosti su dovoljno velike da ne stvaraju probleme. U prilogu se mogu vidjeti oba koda, tj. sa i bez logaritma pod nazivima **viterbilog.py** i **viterbi.py**.

4.2 Forward algoritam

Forward algoritam računa ukupnu vjerojatnost niza u odnosu na model, tu vjerojatnost kraće ćemo označavati sa *score*, $\mathbb{P}(x \mid M)$, formalno:

$$\mathbb{P}(x \mid M) \;=\; \sum_{\pi} \mathbb{P}(x, \pi \mid M)\,, \text{ gdje je M oznaka za model}$$

Taj score dobijemo kao sumu po svim putevima vjerojatnosti niza kroz model. Označimo sa $f_k(i)$ vjerojatnost da niz $x_1, x_2, \ldots, x_i$ završava u stanju k. Formalno:

$$f_k(i) = \mathbb{P}(x_1, x_2, \ldots, x_i \mid \pi_i = k) \tag{4.3}$$

Rekurzija je slična onoj Viterbijevom algoritmu i glasi:

$$\begin{aligned}
\text{inicijalizacija: } & f_B(0) = 1 \\
& f_{M_k}(i) = e_{M_k}(i)\left[f_{M_{k-1}}(i-1)a_{M_{k-1}M_k} + f_{I_{k-1}}(i-1)a_{I_{k-1}M_k} + f_{D_{k-1}}(i-1)a_{D_{k-1}M_k}\right] \\
& f_{I_k}(i) = e_{I_k}(i)\left[f_{M_k}(i-1)a_{M_kI_k} + f_{I_k}(i-1)a_{I_kI_k} + f_{D_k}(i-1)a_{D_kI_k}\right] \\
& f_{D_k}(i) = f_{M_{k-1}}(i)a_{M_{k-1}D_k} + f_{I_{k-1}}(i)a_{I_{k-1}D_k} + f_{D_{k-1}}(i)a_{D_{k-1}D_k}
\end{aligned}$$

Napomena 4.2.1. *Emisijske vjerojatnosti skalirane su s* $\frac{1}{7}$*, a tranzicijske vjerojatnosti s* $\frac{1}{3}$.

Kod se može vidjeti prilogu **forward.py** .

4.3 Višestruko poravnanje HMM-a

4.3.1 Biološko značenje

Višestruko poravnanje nizova (eng. multiple sequence alignment, MSA) je poravnanje tri ili više nizova simbola dobivenih sa ili bez umetanja simbola $'-'$ (eng. gap) što rezultira činjenicom da svi poravnati nizovi imaju jednaku duljinu.

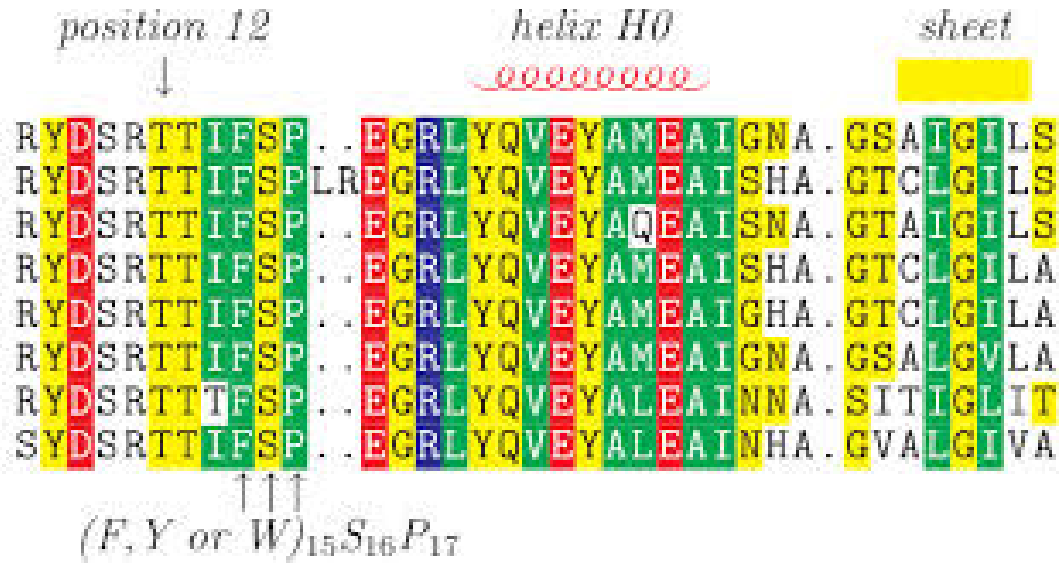
Problem višestrukog poravnanja javlja se u nekoliko područja kao što su molekularna biologija, geologija i računalna znanost. U biologiji osobito je važno za konstrukciju evolucijskog stabla baziranog na DNA i za analizu strukture proteina. Nama su ovdje važni proteini, pa ćemo se usredotočiti na to područje. Zanima nas evolucijska povijest proteina. Pretpostavljamo da proteini nastaju evolucijom pretka, ali zapravo nama nije poznat predak iz kojeg su nastali homologni proteini.

Višestruko poravnanje nizova spada u klasu optimizacijskog problema s eksponencijalnim vremenom složenosti, zvanog kombinatorni problem. Složenost je reda $O(L^N)$, gdje je L prosječna duljina niza koji treba biti poravnat, a N je broj nizova koje treba poravnati, $N \geq 3$. Dakle, višestrukim poravnanjem dobivamo matricu reda $N \times M$, gdje je M duljina poravnanja. Najčešće se poravnavaju cijeli proteinski nizovi i taj rezultat zovemo **globalnim poravnanjem**, no ponekad nas zanima da li dva ili više proteina iz različitih familija imaju zajedničku domenu i to zovemo **lokalno poravnanje**.

4.3.2 Algoritam

Neka je zadana familija nizova, te neka je zadan profil familije (HMM za opis MSA) i alfabet $\mathcal{A} \cup \{-\}$. Da bismo mogli poravnati nizove iz te familije prvo moramo odrediti optimalan prolaz svakog niza kroz model. Za to nam koristi prethodno definiran Viterbijev algoritam. Nakon što su svi nizovi prošli kroz algoritam i zapamtili stanja prolaska kroz model, pri čemu su oznake za stanja kao u poglavlju 3.3.2, vršimo poravnanje stanja koristeći uređaj na skupu stanja modela. (Parcijalni) uređaj na skupu stanja se može odrediti iz

tranzicijskih vjerojatnosti (stanje A je "manje" od stanja B ako je tranzicija iz A u B strogo veća od 0).
Begin i *end* stanje ne poravnavamo njih mičemo. Ostala stanja poravnavamo po uređaju da $I_0 \to (M_1\,,\, D_1) \to I_1 \to (M_2\,,\, D_2) \to \ldots$. Kada su sva stanja poravnata treba umjesto oznaka stanja vratiti oznake alfabeta $\mathcal{A}$. Za emitirajuća stanja pridruži se znak iz alfabeta $\mathcal{A}$, dok se stanju *delete* pridruži znak $-$. Primjer nekog poravnanja možemo vidjeti na slici 4.1. Detaljan kod za višestruko poravnanje zajedno sa Viterbijevim algoritmom možemo vidjeti u prilogu **poravnanje.py**.



Slika 4.1: Višestruko poravnanje - umjesto '-' korišteno '.'

4.4 Procjena modela i simulacija nizova

- **Procjena *match* stanja**

  Poravnanje se promatra stupac po stupac, a svaki stupac prikazuje evoluciju jedne aminokiseline u pretku. Pretpostavljamo da su stupci nezavisni. Procjenu *match* stanja izvodimo tako da brojimo one stupce u poravnatoj familiji za koje vrijedi da je postotak emitiranih simbola iz alfabeta $\mathcal{A}$ veći od 70% u odnosu na cijeli stupac.

- **Procjena duljine modela**

  Duljina modela definira se kao broj *match* stanja i označimo je sa m.

- **Pseudozbroj**

  Pseudozbroj metoda je metoda koja dodaje konstantu u svaki broj, čime se izbjegava problem s vjerojatnošću 0. Ukoliko radimo sa uzorkom koji je nedovoljno velik, moguće su situacije da su emisijske ili tranzicijske vjerojatnosti jednake 0. Broj koji dodajemo emisijama i tranzicijama zove se **pseudozbroj**. Konkretno, u slučaju emisijskih vjerojatnosti vrlo je moguće da simbol nije emitiran u stupcu koji gledamo, što povlači da će vjerojatnost njegovog emitiranja biti 0.

- **Procjena emisijskih vjerojatnosti**

  Emisijske vjerojatnosti su vjerojatnosti emitiranja simbola iz alfabeta $\mathcal{A}$, a *match* i *insert* stanja emitiraju simbole pa samo njih trebamo procijeniti. Vjerojatnost pojavljivanja pojedinog simbola u svakom *insert* stanju stavimo da je

  $$e_{I_i}(a) = \frac{1}{20}, \text{ gdje je } a \in \mathcal{A},\ i \in \{0,\ 1,\ 2,\ \ldots,\ m\}$$

  Za svako stanje koje smo proglasili *match* stanjem radimo slijedeće :

  - brojimo frekvencije pojedinog simbola, e_a
  - zbog izbjegavanja situacije da nam neki simbol nije emitiran u stanju uvodimo pseudo:
    $$f_a = \frac{e_a + 0.1}{1.2}, \text{ gdje je } a \in \mathcal{A}$$
  - procijenjenu emisijsku vjerojatnost dobivamo računajući relativne frekvencije iz prethodnog koraka
    $$e_{M_i}(a) = \frac{f_a}{\sum_{a \in \mathcal{A}} f_a}, \text{ gdje je } a \in \mathcal{A}$$

- **Procjena tranzicijskih vjerojatnosti**

  Procjena tranzicijskih vjerojatnosti je malo teža. Lako smo procijenili koja stanja su *match* stanja. Za svako stanje treba procijeniti slijedeće tranzicije :

  $$\begin{gathered} M_i \to M_{i+1},\ M_i \to I_i,\ M_i \to D_{i+1} \\ I_i \to M_{i+1},\ I_i \to I_i,\ I_i \to D_{i+1} \\ D_i \to M_{i+1},\ D_i \to I_i,\ D_i \to D_{i+1} \end{gathered}$$

  Vjerojatnost prelaska iz *begin* i *insert,* I_0 stanja stavimo fiksno:

  $$\begin{aligned} [a_{BM_1},\ a_{BI_0},\ a_{BD_1}] &= [0.9999,\ 0.00005,\ 0.00005] \\ [a_{I_0M_1},\ a_{I_0I_0},\ a_{I_0D_1}] &= [0.015,\ 0.97,\ 0.015] \end{aligned}$$

Vjerojatnost dolaska u *end* stanje, zapravo taj zadnji korak prelazaka u modelu također fiksiramo:

$$[a_{M_mE},\, a_{M_mI_m}] = [0.999,\, 0.001]$$
$$[a_{I_mE},\, a_{I_mI_m}] = [0.05,\, 0.95]$$
$$[a_{D_mE},\, a_{D_mI_m}] = [0.999,\, 0.001]$$

Ukoliko neko stanje nismo uspjeli procijeniti stavljamo fiksnu vjerojatnost. Neka je i stanje koje nije procijenjeno.

$$[a_{M_iM_{i+1}},\, a_{M_iI_i},\, a_{M_iD_{i+1}}] = [0.999,\, 0.00025,\, 0.00075]$$
$$[a_{I_iM_{i+1}},\, a_{I_iI_i},\, a_{I_iD_{i+1}}] = [0.70,\, 0.15,\, 0.015]$$
$$[a_{D_iM_{i+1}},\, a_{D_iI_i},\, a_{D_iD_{i+1}}] = [0.999,\, 0.00025,\, 0.00075]$$

Neka je $\boldsymbol{j}$ stupac u poravnatoj familiji koji nije *match* stanje, te neka je $e_{j,a}$ frekvencija simbola a u nekom stupcu j .
Prelaske iz *delete* stanja fiksiramo:

$$[a_{D_jM_{j+1}},\, a_{D_jI_j},\, a_{D_jD_{j+1}}] = [0.999,\, 0.00025,\, 0.0075]$$

Ostaje procijeniti prelaske iz *match* i *insert* stanja.

$$j-1 \in M \qquad\qquad j-1 \in I$$
$$a_{MI} = \frac{\sum_{a\in\mathcal{A}} e_{j,a} + 1}{\sum_{a\in\mathcal{A}} e_{j-1,a} + 3} \qquad a_{II} = \frac{\sum_{a\in\mathcal{A}} e_{j,a} + 1}{\sum_{a\in\mathcal{A}} e_{j-1,a} + 3}$$
$$a_{MD} = \frac{1}{\sum_{a\in\mathcal{A}} e_{j-1,a} + 3} \qquad a_{ID} = \frac{1}{\sum_{a\in\mathcal{A}} e_{j-1,a} + 3}$$
$$a_{MM} = 1 - a_{MI} - a_{MD} \qquad a_{IM} = 1 - a_{II} - a_{ID}$$

Ako je $j-1 \in M$, procjena ostaje takva. Transformacije radimo za $j-1 \in I$, zbog višestrukih povrataka *inserta* u *insert*. U tom slučaju računamo aritmetičku sredinu onih stanja koja su više puta procijenjena. Detaljan kod može se vidjeti u prilogu **procjena.py**.

- **Simulacija modela**

  Budući da smo procijenili model iz poravnate familije, s tim modelom bismo htjeli dobiti veći broj članova familije, odnosno uzorak za treniranje, sa kojim bismo mogli donositi statističke zaključke što o poravnanju, što o još boljoj procjeni modela. Dobiveni uzorak za treniranje mora biti reprezentativan uzorak poravnate familije, što znači da mora imati dovoljno članova familije i mora sadržavati karakteristična

svojstva familije. Prvi uvjet zadovoljavamo tako što ćemo simulirati 1000 nizova, možemo i bolje od toga, naravno, ali pretpostavit ćemo da je to dovoljno. Drugi uvjet je također ispunjen, s obzirom da radimo sa procijenjenim modelom. Opisati ćemo simulaciju za jedan niz.

* iz tranzicijskih vjerojatnosti odredimo kumulativne tranzicijske vjerojatnosti

$$\tilde{a}_1 = a_{MM} \ , \tilde{a}_2 = a_{MM} + a_{MI} \ , \tilde{a}_3 = 1$$

* za svako stanje generiramo slučajan broj $\alpha \in \langle 0\,, 1\rangle$
* provjeravamo koji od navedenih uvjeta je zadovoljen

$$\alpha < \tilde{a}_1 \text{ ili } \alpha < \tilde{a}_2 \text{ ili } \alpha < 1$$

i ovisno o tome gdje se α nalazi, pod napomenom ako je neki slučaj zadovoljen prekidamo provjeru, tako se mičemo po stanjima. Naravno ako dođemo u *delete* stanje ne emitiramo simbol, nego nastavljamo dalje sa uvjetima. Ukoliko smo u *match* ili *insert* stanju onda emitiramo simbol, ali kako? Po istom principu kao i kod tranzicijskih vjerojatnosti, transformacija emisijskih vjerojatnosti u kumulativne emisijske vjerojatnosti, dobivamo koji simbol iz alfabeta $\mathcal{A}$.

– Kumulativne emisije za *insert* stanja

$$\tilde{e}_{I_1} = \frac{1}{20}\,, \tilde{e}_{I_2} = \frac{2}{20}\,, \ldots\,, \tilde{e}_{I_{19}} = \frac{19}{20}\,, \tilde{e}_{I_{20}} = \frac{20}{20} = 1$$

– Kumulativne emisije za *match* stanja

$$\tilde{e}_{M_1} = e_{M_1}\,, \tilde{e}_{M_2} = e_{M_1} + e_{M_2}\,, \ldots\,, \tilde{e}_{M_{20}} = 1$$

Detaljan kod može se vidjeti u prilogu **simulacija.py**.

Poglavlje 5

Rezultati

Dana je familija 176 poravnatih nizova **at.aln**, a duljina poravnatih je 324. Za danu familiju procjenu vršimo po opisanom postupku u poglavlju 4.4. Dobiveni model zapisujemo u datoteku **model70.txt**. Mana te procjene jest da fiksiramo neke vjerojatnosti, ne procjenjujemo ih, što i nije najsretniji način, ali ipak poravnanje početnog uzorka tim modelom je dovoljno dobro. Druga stvar koja ranije nije navedana jest zašto smo za procjenu *match* stanja odabrali da je postotak emitiranih simbola u odnosu na stupac 70%. Duljina modela za 70% je $m = 283$, povećanjem postotka smanjuje se duljina modela, a smanjivanjem postotka povećava se duljina modela. Kada početnu familiju (pri čemu u poravnatoj familiji mičemo praznine $'-'$, ostaju samo simboli iz alfabeta) poravnamo u odnosu na taj model, najbliže originalnom poravnanju je onaj model za koji je postotak na 70%. Poravnanje iz procjene zapisano je u **poravnanje70.txt**.

ORIGINALNI PODACI - Dio originalnog poravnanja - **at.aln**

```
PHLDHPLLPLLTQNDNDN-----EDAAALLQQTRYAQPALFAFQVALHRLLTDGYHITPH
PHLDHPLLPLLTQNDNDNDN---EDAAALLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PHLDHPLLPLLTQDPNTQDTTTLEEAAALLQQTRYAQPALFAFQVALHRLLTDGYHITPH
PHLDHPLLPLLTQDPNTQDTTTLEEAAALLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PHLDHPLLPLLTQDPNTQDTTTLEEAAALLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PLMDVDLLT--------LVLDAGAASDSYLQQTRYAQPAIFAVEYALARLWMHWG-VAAD
PVLNESLLE---------VLYGG--KGHLLEQSGVSQPALFAVEYALAQLWKSWG-VKPA
PVLNESLLE---------VLYGG--KGHLLEQSGVSQPALFAVEYALAQLWKSWG-VKPA
GKWAESLLS---------VMHG---TGSRIDDTEYTQPALFALEYALAELWRSWG-VEPW
-ANPISLLS---------VLYPEPGVPTPLDETEFTQPALFALQCALAKLWRSWG-IEPS
PILPKPLLS---------VLYPEVGQESPIDETEYTQPALFAVEYALATLWRSWG-IEPS
PWLGRSLLS---------VIYPESGATSPLDETLFTQPALFAIEYALAELWRSWG-VTPS
PLLGRSILS---------VIYPEAGQASPIDETAFTQPALFAFEWALAELWRSWG-VVPT
PHLGRSLLS---------VLYPEPGSRTPLDETAFTQPALFAIEYALAMLWQSWG-IQPT
AEAGWSLLAELAADEGSSQ----------IERIDVVQPVLFALAVAFAALWRSWG-VGPD
```

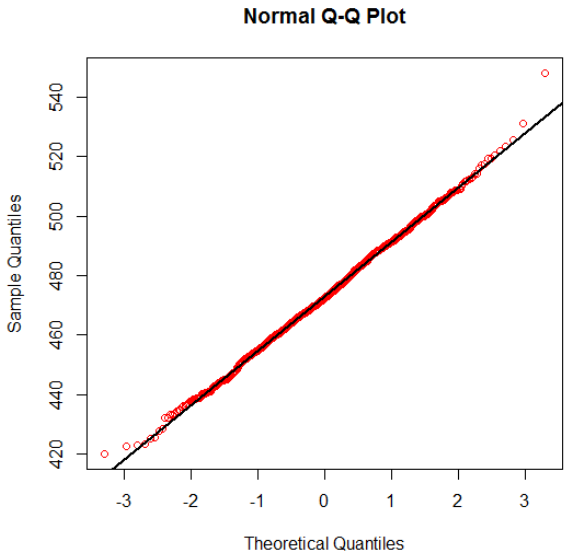
Usporedimo sad originalno poravnanje i poravnanje s procijenjenim modelom. Prvo što možemo uočiti je duljina poravnatih, kod originalnog poravnanja je 324, a procijenjenog 361. Očito su negdje upala neka *insert* stanja, koja su produljila poravnanje. No to nije dovoljno da bismo zaključili da je model dobar. Moramo vidjeti jesu li simboli relativno dobro poravnati. Vidimo da se na početku ovog dijela poravnanja simboli ne poklapaju, očito ima više *insert* stanja, dok se od sredine pa do kraja poklapa.

PROCIJENJENI MODEL - Dio poravnanja s novim modelom - **poravnanje70.txt**

```
PHL-D--HP----LLPLLTQNDNDN-------EDAAA---------LLQQTRYAQPALFAFQVALHRLLTDGYHITPH
PHL-D--HP----LLPLLTQNDNDNDN-----EDAAA---------LLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PHL-D--HP----LLPLLTQDPNTQDTTTLE-EAAA----------LLQQTRYAQPALFAFQVALHRLLTDGYHITPH
PHL-D--HP----LLPLLTQDPNTQDTTTLE-EAAA----------LLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PHL-D--HP----LLPLLTQDPNTQDTTTLE-EAAA----------LLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PLM-D--VD----LLTLVLD------------AGAASDSY-------LQQTRYAQPAIFAVEYALARLWMHWG-VAAD
PVL-N--ES----LLE---------------VLYGGKGH-------LLEQSGVSQPALFAVEYALAQLWKSWG-VKPA
PVL-N--ES----LLE---------------VLYGGKGH-------LLEQSGVSQPALFAVEYALAQLWKSWG-VKPA
GKW-A--ES----LLS---------------VMHGTGS--------RIDDTEYTQPALFALEYALAELWRSWG-VEPW
A-N-P--IS----LLS---------------VLYPEPGVPT-----PLDETEFTQPALFALQCALAKLWRSWG-IEPS
PIL-P--KP----LLSVLYPEV----------GQES----------PIDETEYTQPALFAVEYALATLWRSWG-IEPS
PWL-G--RS----LLSVIYPE----------SGATS---------PLDETLFTQPALFAIEYALAELWRSWG-VTPS
PLL-G--RS----ILSVIYPEA---------GQAS----------PIDETAFTQPALFAFEWALAELWRSWG-VVPT
PHL-G--RS----LLSVLYPE----------PGSRT---------PLDETAFTQPALFAIEYALAMLWQSWG-IQPT
AEA-G--WS----LLA---------------ELAADEGSSQ------IERIDVVQPVLFALAVAFAALWRSWG-VGPD
```

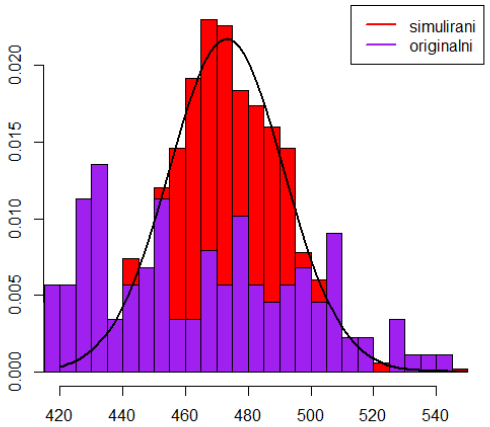
Sljedeći korak je simulirati uzorak za treniranje po opisanom postupku u poglavlju 4.4. Simuliranu familiju možemo vidjeti u prilogu u datoteci **simulirani70.txt**. Problem simulacije je što nizovi nisu jednake duljine. Duljina najvećeg niza je 1647, a najkraćeg 281.

Daljnja ideja je da pokušamo što bolje procijeniti model pomoću simulacije. Tako ćemo *forward algoritmom* izračunati scorove za simulirani uzorak i za orginalne podatke. Scorove smo prije prikazivanja u histogramu još i logaritmirali (log-scorovi), te u *R*-u ispitujemo pripadnost normalnoj distribuciji gdje smo μ i σ procijenili i vrijedi $\tilde{\mu} = 473.1203$ i $\tilde{\sigma} = 18.36406$.



Slika 5.1: QQplot log-scorova za simulirane podatke

Na slici 5.1 prikazan je *qqplot* log-scorova, koji ukazuje da podaci relativno dobro aproksimiraju pravac, što zapravo znači da su log-scorovi simuliranih nizova aproksimativno normalno distribuirani.



Slika 5.2: Histogram log-scorova i pripadnost normalnoj distribuciji

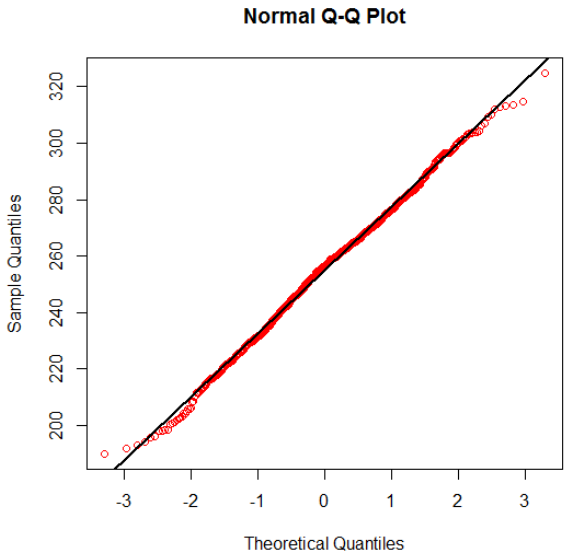
Na slici 5.2 prikazan je histogram log-scorova originalnih i simuliranih podataka, te je nacrtana normalna vjerojatnosna distibucija, gdje su μ i σ procijenjeni iz simuliranih log-socorova. Taj graf nam potvrđuje zaključak sa slike 5.1, odnosno log-scorovi dolaze iz normalne distribucije, uz to histogram lijepo prikazuje kako simulirani log-scorovi dijele originalne log-scorove.

Kako na temelju histograma odlučiti koji nizovi će poboljšati procjenu modela? Gledamo desni rep normalne distribucije, te odabiremo proizvoljan log-score koji se nalazi u tom repu. Procjena novog modela zasniva se na nizovima čiji su log-scorovi veći od 510. Takvih nizova ima 3. Duljina novog modela je 271.

PROCIJENJENI MODEL - Dio poravnanja sa novim modelom - **poravnanje701.txt**

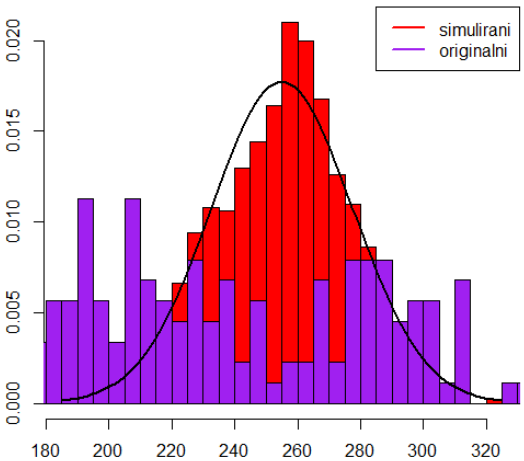
```
PHLD--HPLLPLLTQNDNDNEDAAAL-----LQQTRYAQPALFAFQVALHRLLTDGYHITPH
PHLD--HPLLPLLTQNDNDNDNEDAAAL---LQQTPYAQPALFAFQVALHRLLTDGYHITPH
PHLD--HPLLPLLTQDPNTQDTTTLEEAAALLQQTRYAQPALFAFQVALHRLLTDGYHITPH
PHLD--HPLLPLLTQDPNTQDTTTLEEAAALLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PHLD--HPLLPLLTQDPNTQDTTTLEEAAALLQQTPYAQPALFAFQVALHRLLTDGYHITPH
PLMD--VDLLTLVLDAGAASDSY--------LQQTRYAQPAIFAVEYALAR-LWMHWGVAAD
PVLN--ESLLEVLYGGKGHL-----------LEQSGVSQPALFAVEYALAQ-LWKSWGVKPA
PVLN--ESLLEVLYGGKGHL-----------LEQSGVSQPALFAVEYALAQ-LWKSWGVKPA
GKWA--ESLLSVMHGTGSR------------IDDTEYTQPALFALEYALAE-LWRSWGVEPW
GANP--ISLLSVLYPEPGVPTP---------LDETEFTQPALFALQCALAK-LWRSWGIEPS
PILP--KPLLSVLYPEVGQESP---------IDETEYTQPALFAVEYALAT-LWRSWGIEPS
PWLG--RSLLSVIYPESGATSP---------LDETLFTQPALFAIEYALAE-LWRSWGVTPS
PLLG--RSILSVIYPEAGQASP---------IDETAFTQPALFAFEWALAE-LWRSWGVVPT
PHLG--RSLLSVLYPEPGSRTP---------LDETAFTQPALFAIEYALAM-LWQSWGIQPT
AEAG--WSLLAELAADEGSSQ----------IERIDVVQPVLFALAVAFAA-LWRSWGVGPD
```

Usporedivši s prethodnim poravnanjem, novi model je bolji. Poravnanje je bliže onom originalnom (**poravnanje701.txt**). To vidimo i iz duljine koja se sa 361 smanjila na 353. Simuliramo novi uzorak za treniranje. Niti sad nisu duljine simuliranih nizova jednake ali više nema tako velike razlike među njima. Najduži niz je duljine 350, a najkraći je duljine 268. Očekujemo da će log-scorovi simuliranih biti aproksimativno normalno distribuirani pa to i provjeravamo. Procijenjeni $\mu = 255.0104$ i $\sigma = 22.5034$.



Slika 5.3: QQplot log-scorova za simulirane podatke s novim modelom

Sa slike 5.3 vidimo da log-scorovi relativno dobro aproksimiraju normalni vjerojatnosni graf, dakle log-scorovi su normalno distribuirani.



Slika 5.4: Histogram log-scorova i pripadnost normalnoj distribuciji

Sa slike 5.4 potvrđujemo da su log-scorovi simuliranih normalno distribuirani. Također, simulirani dijele originalne log-scorove.

Dakle, gore opisanim postupkom, iterativno bismo mogli dobiti model koji bi originalne nizove poravnao bolje nego svaki prethodni model. U jednom trenutku kad bi nam se poravnanje relativno dobro poklapalo sa originalnim, svjesni činjenice da nećemo dobiti potpuno poklapanje, tada bismo stali. Pri tom postupku zanemarili smo da stupci u poravnanju možda i ovise jedni o drugima, te smo ih promatrali kao statistički nezavisne. Prvo što pokušavamo je procijeniti model, a kao što je već navedeno ima puno fiksiranja vjerojatnosti i uvijek možemo uzeti veći ili manji postotak simbola u nekom stupcu. Pa ukoliko promijenimo samo jednu vrijednost model će se promijeniti. Što je dobro poravnanje ostaje na nama da sami procijenimo, ali svakako smo dobili iterativni proces kojim svaki put dobijemo sve bolji model.

Bibliografija

[1] Durbin, R.: *Biological sequence analysis : Probabilistic Models of Proteins and Nucleic*. Cambridge University Press, 1998.

[2] Rudman, M.: *Kompleksnost skrivenih Markovljevih modela*. PMF-MO, skripta, 2014.

[3] Sarapa, N.: *Teorija vjerojatnosti*. Školska knjiga, Zagreb, 2002.

[4] Vondraček, Z.: *Markovljevi lanci*. PMF-MO, skripta, 2008.

Sažetak

U ovom diplomskom radu bavimo se opisom profila familije proteina. Opisati profil familije znači procijeniti parametre pridruženog skrivenog Markovljevog modela. Profil familije omogućuje da se novi članovi familije mogu poravnati s već poravnatom familijom.

Procjena parametara modela nije jednostavan proces, neke parametre procijenimo, druge fiksiramo. Promijenimo li samo jednu vrijednost u procjeni, promijenit će se poravnanje, dakle možemo reći da procjena nije robusna. Iz poravnate familije proteina procijenimo inicijalne parametre, na temelju kojih iterativno želimo dobiti parametre pomoću kojih će poravnanje familije biti najsličnije originalnom. Proceduru za procjenu parametara ponovili smo dva puta i sa svakom procjenom dobili bolje poravnanje.

Iako smo napravili samo dvije procjene, ta procedura može biti iterirana. S druge strane ali moramo imati na umu da ne moramo dobiti potpuno isto poravnanje.

Summary

In this thesis, we want to describe a protein family profile. To describe the family profile means to estimate the parameters of the Hidden Markov Model associated to an alignment. Family profile enables aligning of newly discoverd family members with an already existing alignment.

Estimation of the model parameters is not a simple procedure, some parameters are estimated, others are given. If we change a single value in the estimate, the resulting alignment might change, therefore, our estimate is not robust. From a given protein family we estimate the initial parameters, and we would like to get parameters so that the alignment of the family will be the most similar to original alignment. We repeated the estimation procedure twice and that yielded a better alignment.

Although we have made only two model estimations, this procedure can, clearly, be iterated. However, we do not expect to obtain an alignment identical to the one we started with.

Životopis

Rođena sam 29.07.1990 godine u Zaboku. U razdoblju od 1997. do 2005. godine pohađam Osnovnu školu Mače u Maču. Od 2005. do 2009. godine pohađam XVIII. gimnaziju u Zagrebu. 2009. godine upisujem preddiplomski sveučilišni studij na PMF-MO u Zagrebu, koji završavam 2013. godine. Iste godine na PMF-MO nastavljam školovanje upisivanjem diplomskog studija Matematičke statistike.