SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Ivana Kurolt

PORAVNANJE PROTEINSKIH STRUKTURA

Diplomski rad

Voditelj rada:
doc. dr. sc. Pavle Goldstein

Zagreb, rujan, 2014.

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. _____

Sadržaj

Uvod

Proteini su vrlo važan dio organizma jer sudjeluju u svim unutarstaničnim procesima. Mnoge metode predviđanja funkcija pojedinih proteina se oslanjaju na identificiranje sličnosti među strukturama proteina nepoznate funkcije i proteina poznate funkcije. Smatra se da dijelovi proteina sličnih struktura imaju slično djelovanje odnosno ponašanje. Eksperimentalne metode, kao što su nuklearna magnetska rezonanca i spektroskopija, su vrlo skupe i vremenski zahtjevne, stoga računalni algoritmi koji prepoznaju sličnosti u proteinima višestruko ubrzavaju i pojeftinju postupak.

Cilj ovog rada je prikazati primjer jednog takvog algoritma za poravnanje proteinskih struktura.

Ovaj rad je podjeljen u tri poglavlja. U prvom poglavlju je opisana struktura proteina kao i njihova funkcija. Isto tako navode se dva oblika poravnanja proteina a to su poravnanje nizova i strukturalno poravnanje. U drugom poglavlju su detaljno opisane matrice udaljenosti dvaju proteina, jednog proteina i matrice udaljenosti za parove sekundarnih struktura. Ove matrice nam daju uvid u lokalna poravnanja atoma ugljika u proteinima i pomoću njih uviđamo podudaranja struktura. U posljednjem poglavlju detaljno je opisan algoritam za poravnanje dvaju proteinskih struktura.

Poglavlje 1

Proteini

1.1 Struktura proteina

Aminokiselina je osnovna građevna jedinica proteina. Sastoji se od središnjeg atoma ugljika (C_α atom) i na njega vezanih atoma vodika, amino skupine, karboksilne skupine i bočnog lanca. O bočnom lancu ovise kemijska svojstva aminokiseline. Niz aminokiselina u proteinu je određen nizom kodona A, C, G, T, koji je je zapisan u genetskom kodu. Općenito, genetski kod precizira 20 standardnih aminokiselina (slika 1.1). Poznato ih je mnogo više, ali ne izgrađuju sve proteine u živim bićima ili se javljaju u neznačajnim postocima. Aminokiseline se označavaju skraćenicama od jednog ili tri slova.

Primarna

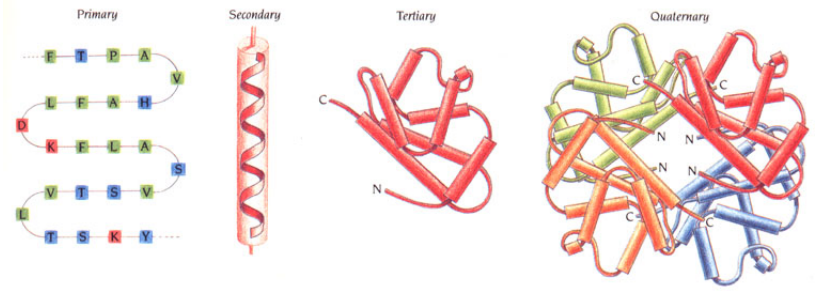
Niz sastavljen od više aminokiselina povezanih peptidnim vezama je polipeptid, odnosno protein. Taj polipetidni niz je zapravo primarna struktura proteina i većina ih sadrži od 100 do 1000 aminokiselina.

Sekundarna

Sekundarna struktura proteina je prostorna organizacija uvjetovana interakcijama, najčešće vodikovim vezama, među atomima aminokiselina bliskih u primarnoj strukturi. U obzir se uzima samo okosnica dok se bočni ogranci zanemaruju. Elementi sekundarne strukture su α-zavojnice, β-ploče koje su povezani β-lanci. U ovu strukturu ubrajaju se jos i zavoji i petlje.

AMINOKISELINA	KRATICA
alanin	Ala-A
arginin	Arg-R
asparagin	Asn-N
asparaginska kiselina	Asp-D
cistein	Cys-C
fenilalanin	Phe-F
glutamin	Gln-Q
glutaminska kiselina	Glu-E
glicin	Gly-G
histidin	His-H
izoleucin	Ile-I
leucin	Leu-L
lizin	Lys-K
metionin	Met-M
prolin	Pro-P
serin	Ser-S
tirozin	Tyr-Y
treonin	Thr-T
triptofan	Trp-W

Slika 1.1: 20 standardnih aminokiselina



Slika 1.2: Razine proteinske strukture

Tercijarna

Tercijarna struktura je prostorna organizacija koja je nastala interakcijama među aminokiselinama koje nisu blizu u primarnoj strukturi. Protein, nakon što poprimi tercijarnu strukturu, može obavljati svoju funkciju. Stoga je poznavanje tercijarne strukture proteina vrlo važno za proučavanje njegove evolucije i funkcije.

Kvaterna

Udruživanjem više proteina u veće agregate nastaje proteinski kompleks koji predstavlja kvaternu strukturu.

1.2 Funkcija proteina

Osnovna funkcija proteina je sudjelovanje u procesu rasta i razvoja. Za svaki dio našeg tijela koji prolazi kroz procese rasta ili regeneracije, stvaraju se nove tjelesne stanice kojima su potrebni proteini za njihovu izgradnju i uspostavljanje odgovarajuće funkcije u tijelu. Bitna funkcija proteina je nadomjestak oštećenih i odmrlih stanica kao što su stanice krvi, bubrega, jetre i mišića. Također i stanica kose, noktiju, zubi i kosti. Proteini su enzimi koji ubrzavaju biokemijske procese i zaslužni su za oblik života kakav mi poznajemo danas. Omogućuju komunikaciju i usklađivanje biokemijskih procesa između različitih tkiva i organa, takve proteine nazivamo hormonima. Bitni su i za obranu organizma od bakterija i virusa.

1.3 Poravnanje proteina

Poravnanje nizova

Definicija 1.3.1. *Neka je $\mathcal{A}$ alfabet i $W(\mathcal{A})$ skup svih konačnih riječi nad $\mathcal{A}$. Neka je $\mathcal{A}' = \mathcal{A} \cup \{-\}$ gdje $-$ označava prazninu, odnosno situaciju kada simbol iz jednog niza nije pridružen niti jednom simbolu iz drugog niza. $W(\mathcal{A}')$ je skup svih konačnih riječi nad alfabetom $\mathcal{A}'$. Neka su nizovi $a = (a_1, a_2, \ldots, a_n)$, $b = (b_1, b_2, \ldots, b_n) \in W(\mathcal{A})$; $a' = (a'_1, a'_2, \ldots, a'_n)$, $b' = (b'_1, b'_2, \ldots, b'_n) \in W(\mathcal{A}')$. Poravnanje od a i b označavamo sa $A(a, b)$ i definiramo kao bilo koje preslikavanje $a \mapsto a'$ i $b \mapsto b'$ tako da vrijedi:*

1. $a'|_{\mathcal{A}} = a,\ b'|_{\mathcal{A}} = b$
2. $|a'| = |b'| = k$
3. $\forall i = 1, \ldots, k,\ a'_i \neq -\ \ ili\ \ b'_i \neq -$

Ako su a i b nizovi aminokiselina, $A(a, b)$ njihovo poravnanje i zadovoljeno je $|a'| = |b'| = k$, za simbole a_i i b_j kažemo da su poravnati ako postoji $l \in 1, \ldots, k$ takav da vrijedi $a'_l = a_i$ i $b'_l = b_j$.

Poravnanje nizova je način uspoređivanja nizova DNA, RNA ili proteina koji omogućuje prepoznavanje sličnih cjelina u proteinu koje mogu biti posljedica funkcionalne, strukturne ili evolucijske veze između tih nizova. Poravnanje se najčešće prikazuje matricom, čiji reci

npr. predstavljaju isti protein kod različitog organizma ili dva proteina sa sličnim ili istim svojstvima kod istog organizma. Stupce te matrice čine poravnati simboli. Ti simboli su slova alfabeta koja predstavljaju određenu aminokiselinu od njih 20. Kod poravnanja nizova, slučaj pridruživanja praznine simbolu zovemo delecija, a slučaj pridruživanja simbola praznini zovemo insercija. Ako dva niza dijele zajedničkog pretka, nepodudarnosti se mogu interpretirati kao mutacije, a praznine kao insercijske ili delecijske mutacije.

Strukturalno poravnanje

Strukturno poravnanje pokušava uspostaviti sličnost između dva ili više polimerskih struktura na temelju njihovog oblika tj. trodimenzionalne strukture. Ovaj postupak se obično primjenjuje na protein tercijarne strukture. Za ulazne podatke uzimaju se setovi prostornih koordinata atoma dva ili više nizova aminokiselina. Pretpostavimo da imamo dvije proteinske strukture $\mathcal{X}$ i $\mathcal{Y}$ iste duljine n, tj. $\mathcal{X} = [x_1, x_2, \ldots, x_n]$, $\mathcal{Y} = [y_1, y_2, \ldots, y_n]$. Svaka aminokiselina u proteinu je reprezentirana nizom od tri koordinate u prostoru $x_i = (x_i^1, x_i^2, x_i^3)^T$ odnosno $y_i = (y_i^1, y_i^2, y_i^3)^T \ \forall i = 1, \ldots, n$. Da bi se proteinske strukture poravnale, treba pronaći matricu rotacije $A \in M_3(\mathbb{R})$ i vektor translacije $B \in \mathbb{R}^3$ tako da je udaljenost transformirane proteinske strukture $A\mathcal{X} + B$ i strukture $\mathcal{Y}$ minimalna moguća. Odnosno, dovoljno je minimizirati kvadrat udaljenosti

$$\sum_{i=1}^{n} d(Ax_i + B, y_i)^2 \to min$$

gdje d označava 2-normu razlike vektora.

Definicija 1.3.2. *Neka su* $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ *i* $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ *vektori u* $\mathbb{R}^n$. *Euklidska metrika ili 2-norma vektora je dana sa:*

$$d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}|| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$$

.

Izlazna informacija uspješnog strukturalnog poravnanja je set superponiranih trodimenzionalnih koordinata svake ulazne strukture primjenom metoda kao što su metoda minimizacije greške RMSD (*engl. Root Mean Square Distance*), metoda globalne udaljenosti GDT (*engl. Global Distance Test*), DALI (*Distance Alignment Matrix*) i drugo. Zajedničko svim tipovima strukturalnog poravnanja je uzimanje u obzir atoma okosnice strukture, tj. koordinata atoma uključenih u peptidnu vezu. Udaljenijim atomima se smanjuje značaj kako bi se izbjegli negativni efekti nastali zbog mobilnosti omči, torzije zavojnica i ostalih manjih strukturnih varijacija. Izlazni podaci obično uključuju globalno optimalno poravnanje te vrijednost koja predstavlja stupanj sličnosti.

Poglavlje 2

Matrice udaljenosti

2.1 Uvod

Matrica udaljenosti dva proteina je matrica lokalnog poravnanja C^α-atoma jednog proteina sa C^α-atomima drugog proteina. U DALI algoritmu radimo superpoziciju struktura po svim koordinatama atoma okosnice, vertikalnim i horizontalnim pomacima za jednu poziciju. Podudaranje struktura tražimo metodom klizajućih prozora veličina koje su fiksne. Na primjer za prozor duljine 5 A.K. dobivamo izračun sličnosti pentapeptida, a za prozor duljine 10 A.K. izračun dekapeptida.

Definicija 2.1.1. *Označimo sa k duljinu prozora. Zapisujemo uređenu k-torku sastavljenu od liste koordinata atoma okosnice prvog promatranog proteina p^A duljine $|p^A|$ kao $\boldsymbol{v} = (v_1, v_2, \ldots, v_i, \ldots, v_{|p^A|})$. Analogno za protein p^B pišemo $\boldsymbol{u} = (u_1, u_2, \ldots, u_i, \ldots, u_{|p^B|})$. Vektor $wl_i^k, i \in (1, \ldots, |p| - k + 1)$ zovemo **vektor udaljenosti** C_i^α-atoma za prozor k, pri čemu je i bilo koja pozicija u proteinu. Poziciji i u p^A za neki k pridružujemo vektor udaljenosti $wl_i^k(p^A)$ na sljedeći način:*

$$wl_i^k(p^A) = \begin{pmatrix} d(v_i, v_{i+1}) & d(v_i, v_{i+2}) & d(v_i, v_{i+3}) & \ldots & d(v_i, v_{i+k-1}) \\ & d(v_{i+1}, v_{i+2}) & d(v_{i+1}, v_{i+3}) & \ldots & d(v_{i+1}, v_{i+k-1}) \\ & & & \ddots & \vdots \\ & & & & d(v_{i+k-2}, v_{i+k-1}) \end{pmatrix}$$

Analogno tome, poziciji i u p^B za neki k pridružujemo vektor udaljenosti $wl_i^k(p^B)$ na sljedeći način:

$$wl_i^k(p^B) = \begin{pmatrix} d(u_i, u_{i+1}) & d(u_i, u_{i+2}) & d(u_i, u_{i+3}) & \ldots & d(u_i, u_{i+k-1}) \\ & d(u_{i+1}, u_{i+2}) & d(u_{i+1}, u_{i+3}) & \ldots & d(u_{i+1}, u_{i+k-1}) \\ & & & \ddots & \vdots \\ & & & & d(u_{i+k-2}, u_{i+k-1}) \end{pmatrix}$$

Preglednosti radi zapisali smo gornje vektore u obliku gornje trokutaste matrice, duljina vektora je $\frac{k(k-1)}{2}$.

2.2 Matrica udaljenosti dvaju proteina

Označimo listu svih vektora udaljenosti $wl_i^k(p)$ sa $WL^k(p)$ i zapišemo je kao matricu dimenzije $(\frac{k(k-1)}{2}) \times (|p| - k + 1)$:

$$WL^k(p) = \begin{pmatrix} wl_1^k(p) \\ wl_2^k(p) \\ \vdots \\ wl_{|p|-k+1}^k(p) \end{pmatrix}$$

Definicija 2.2.1. *Za neki k uspoređujemo dva pridružena vektora udaljenosti $wl_i^k(p)$ i $wl_j^k(p)$ oko odgovarajućih pozicija i i j; $wl_i^k(p)$, $wl_j^k(p) \in WL^k(p)$. Matrica definirana s $M(p^A, p^B) = (m_{ij}) = d(wl_i^k(p^A), wl_j^k(p^B))$, gdje je d 2-norma razlike vektora, zovemo* ***matrica udaljenosti proteina*** *p^A* ***i*** *p^B.*

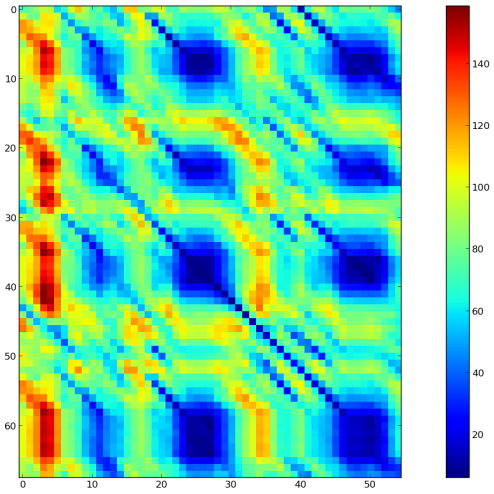
U matrici udaljenosti na *i*-toj i *j*-toj poziciji se nalazi Euklidska udaljenost vektora $wl_i^k(p^A)$ i $wl_j^k(p^B)$, tj. matrica je oblika:

$$M(p^A, p^B) = \begin{pmatrix} d(wl_1^k(p^A), wl_1^k(p^B)) & \dots & d(wl_1^k(p^A), wl_{|p^B|-k+1}^k(p^B)) \\ \vdots & d(wl_i^k(p^A), wl_j^k(p^B)) & \vdots \\ d(wl_{|p^A|-k+1}^k(p^A), wl_1^k(p^B)) & \dots & d(wl_{|p^A|-k+1}^k(p^A), wl_{|p^B|-k+1}^k(p^B)) \end{pmatrix}$$

Iz matrice udaljenosti otkrivamo lokalne sličnosti u 3D strukturi oko odgovarajućih točaka, odnosno točaka oko kojih smo gledali vektore. Ovakva matrica koristi se za traženje parova jako sličnih segmenata sekundarne strukture (motiva) određene duljine (duljina prozora) dvaju promatranih proteina.

2.3 Matrica udaljenosti (jednog) proteina

Tražimo lokalne sličnosti u proteinu računajući geometrijske sličnosti intramolekularnih udaljenosti, kao što smo vidjeli ranije, to činimo oduzimanjem dvaju vektora udaljenosti.



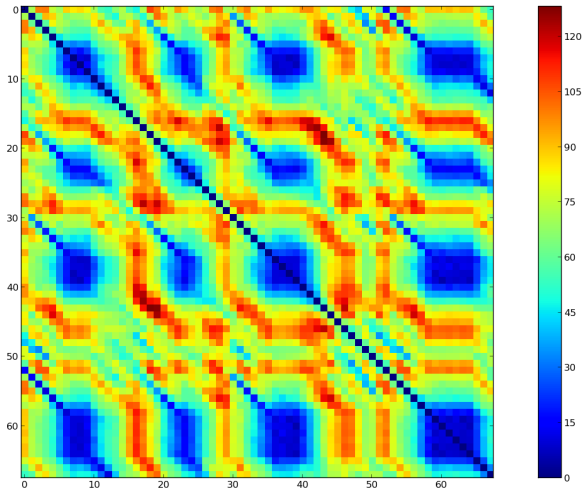
Slika 2.1: Matrica udaljenosti proteina 1mu5A02 i 2gybM01 za prozor duljine 10

Za neki k, izračunamo $wl_i^k(p), \forall i \in (1, 2, \ldots, |p| - k + 1)$ prema definiciji 2.1.1. Zatim izračunamo $d(wl_i^k(p), wl_j^k(p)), \forall i, j \in WL^k(p)$ prema definiciji 1.3.2. Uzimamo u obzir i kada je $i == j$ te tako na dijagonali matrice udaljenosti dobijemo nule. **Matrica proteina** $M(p)$ je dimenzije $(|p| - k + 1) \times (|p| - k + 1)$ ovisno koju vrijednost k uzmemo.

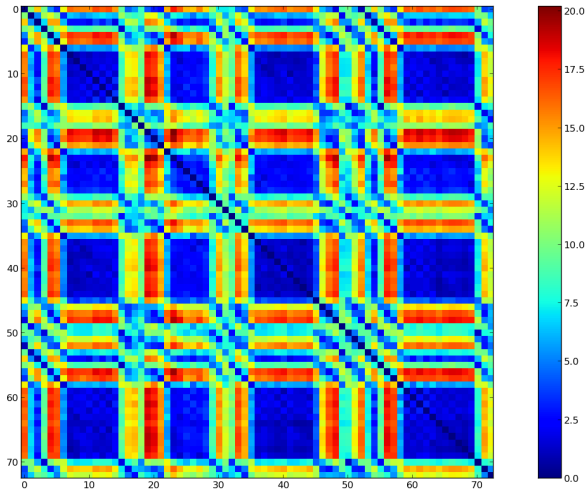
Zavojnice i ploče u proteinu su periodične strukture sa konstantnim (gotovo identičnim) kutevima. Zbog toga na pozicijama gdje se slične strukture *poklope* u matrici $M(p)$ uz dijagonalu očekujemo submatrice malih vrijednosti. Submatrice su zapravo razlike vektora udaljenosti oko pozicija koje se nalaze unutar elemenata sekundarne strukture proteina. Pomoću računa matrice udaljenosti zapravo pronalazimo pozicije i duljine elemenata sekundarne strukture polipeptidnog lanca. Točnu poziciju i duljinu elementa određujemo usporedbom rezultata dobivenih pomoću različitih velicina duljina prozora k i izračunom udaljenosti $d(C_i^\alpha, C_{i+3}^\alpha), \forall i \in (1, 2, \ldots, |p| - 3)$. Znamo i da su petlje obično kraće od zavojnica i strukturu kraću od 4 A.K. ne smatramo zavojnicom.

2.4 Matrica udaljenosti za parove sekundarne strukture

Na poslijetku ćemo promatrati matrice udaljenosti za parove motiva. Vidjeli smo da pomoću matrice udaljenosti proteina određujemo položaje i duljinu elemenata sekundarne struk-



Slika 2.2: Matrica udaljenosti proteina 1mu5A02 samim sa sobom za prozor duljine 10



Slika 2.3: Matrica udaljenosti proteina 1mu5A02 samim sa sobom za prozor duljine 5

ture, a pomoću matrice udaljenosti dva proteina dobijamo na uvid pozicije i duljine segmenata lokalnih podudarnosti.

Definicija 2.4.1. *Neka je k duljina prozora i neka uređena k-torka* $(C_i^\alpha, \ldots, C_{i+k-1}^\alpha)$ *označava prvi motiv prteina* p^A*, a* $(C_j^\alpha, \ldots, C_{j+k-1}^\alpha)$ *drugi motiv istog proteina. Analogno za protein*

p^B prvi motiv označimo sa $(C^\alpha_h, \ldots, C^\alpha_{h+k-1})$, a drugi sa $(C^\alpha_l, \ldots, C^\alpha_{l+k-1})$. Definiramo ***matricu udaljenosti parova motiva*** *$S^k_{i,j,h,l} = (s_{x,y})$ za neke pozicije i,j,h,l. k je duljina prozora. Matrica je oblika:*

$$S^k_{i,j,h,l} = \begin{pmatrix} (d(C^\alpha_j, C^\alpha_i) - d(C^\alpha_l, C^\alpha_h))^2 & (d(C^\alpha_j, C^\alpha_{i+1}) - d(C^\alpha_l, C^\alpha_{h+1}))^2 & \ldots & (d(C^\alpha_j, C^\alpha_{i+k-1}) \\ & & & -d(C^\alpha_l, C^\alpha_{h+k-1}))^2 \\ \vdots & \vdots & \vdots & \vdots \\ (d(C^\alpha_{j+k-1}, C^\alpha_i) - d(C^\alpha_{l+k-1}, C^\alpha_h))^2 & (d(C^\alpha_{j+k-1}, C^\alpha_{i+1}) - d(C^\alpha_{l+k-1}, C^\alpha_{h+1}))^2 & \ldots & (d(C^\alpha_{j+k-1}, C^\alpha_{i+k-1}) \\ & & & -d(C^\alpha_{l+k-1}, C^\alpha_{h+k-1}))^2 \end{pmatrix}$$

tražimo argmin $\sum^k_{x,y}(s_{x,y})$.

Računamo preklapanje submatrica fiksirane duljine, tražimo parove sličnih motiva na način da izračunamo poravnanja prvih motiva od oba proteina uz uvjet da su drugi motivi poravnati, tj. iterativno hodamo po lancima gdje smo pronašli slične motive. Od prvog do zadnjeg C^α atoma svakog motiva poravnavamo svaki sa svakim uz uvjet da su susjedni poravnati.

Tražimo preklapanje na najmanje 5 pozicija (za svaki motiv). Preklapanje rastežemo do duljine najkraćeg elementa sek. strukture. Metaheuristički je određeno kako rezultat $min(s_{x,y})$ treba rasti, čim je manji broj, bolje je preklapanje tj. veća je sličnost. Dakle, imamo maksimizaciju broja C^α-atoma u ekvivalentnim motivima uz minimizaciju devijacije strukture. Ovim uvjetovanjem poravnanja kombinacije dva susjedna elementa sekundarne strukture aproksimiramo lokalno poravnanje supra-sekundarnih struktura dva proteina. Ista metoda se može korstiti i za poravnanje trećeg motiva uz uvjet da neka dva bliska već poravnata.

Poglavlje 3

Algoritam

Algoritam ćemo opisati u nekoliko koraka. Implementacija algoritma je rađena u Matlabu i korištene su proteinske strukture 1.10.8.50_2gybM01 i 1.10.8.50_1mu5A02. Označimo sa $d : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ Euklidsku metriku na $\mathbb{R}^3$. Parametri koje koristimo u algoritmu su *wind*,*trash*,*penal*,*koliko*,*dlanac* i bit će naknadno objašnjeni.

1. Već smo ranije spomenuli kako je svaka aminokiselina u proteinskoj strukturi prezentirana nizom od tri koordinate u prostoru, stoga na početku kao ulazne podatke uzimamo dva polja točaka u prostoru duljina n_1 i n_2:

   $$\mathbf{tocke}_p : \{1, 2, \ldots, n_p\} \to \mathbb{R}^3, \quad p = 1, 2$$

   Koordinate točaka za oba proteina su zapisane u datotekama ekstenzija .txt i Matlab ih otvara pomoću funkcije open('ime_datoteke.txt','r') i sprema u vektor. Konačno, koordinate aminokiselina proteinskih stuktura su zapisane u obliku:

   $$\mathbf{tocke}_1 = [[-94.58, -55.634, 2.171], \ldots, [-93.656, -47.757, -13.499]]$$

   $$\mathbf{tocke}_2 = [[25.748, 73.306, -2.544], \ldots, [25.927, 86.172, -1.707]]$$

2. Nakon toga definiramo parametar *wind* koji označava širinu prozora u kojemu ćemo uspoređivati *wind*-torke točaka iz oba niza. U radu je korištena duljina prozora 10. Na slikama 2.2 i 2.3 su prikazane matrice udaljenosti proteina za prozore duljina 10 i 5.

3. Definiramo dva polja matrica dimenzije $wind \times wind$:

   $$\mathbf{matrice}_p : \{1, \ldots, n_p - wind + 1\} \to M_{wind \times wind}(\mathbb{R}), \ p = 1, 2$$

   tako da je $matrice_p(i)$, $i = 1, \ldots, n_p - wind + 1$ matrica međusobnih udaljenosti točaka $tocke_p(i), \ldots, tocke_p(i + wind - 1)$. Dakle na mjestima (j, k) u matricama se nalaze

euklidske udaljenosti točaka $tocke_p(i + j - 1)$ i $tocke_p(i + k - 1)$ koje se računaju kao u definiciji 1.3.2, odnosno:

$$matrice_p(i)_{j,k} = d(tocke_p(i + j - 1), tocke_p(i + k - 1)),\ \ j, k = 1, \ldots, wind$$

Tako će na primjer prvih par članova prvog retka matrice $matrice_1$ biti

$$[0.0, 3.813842812702168, 6.526780906388695, \ldots]$$

jer su u

$$tocke_1 = [[25.748, 73.306, -2.544], [28.522, 74.395, -0.164], [29.226, 78.099, 0.2], \ldots]$$

pa je prvi član u matrici jednak nula jer nema udaljenosti između prve aminokiseline sa njom samom, ostale se udaljenosti računaju po formuli 1.3.2.

4. Sada kada smo izračunali oba polja matrica $matrice_1$ i $matrice_2$, uspoređujemo ih. Definiramo novu matricu **usporedi** $\in M_{n_1-wind+1, n_2-wind+1}(\mathbb{R})$ sa:

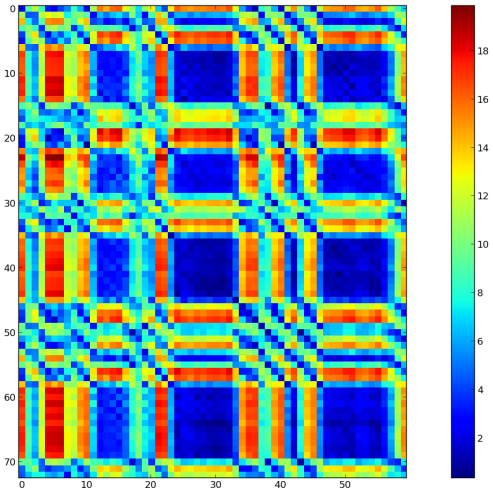
$$usporedi_{i_1,i_2} = \frac{1}{2} \sum_{j,k=1}^{wind} |matrice_1(i_1)_{j,k} - matrice_2(i_2)_{j,k}|.$$

Na slikama 3.1 i 2.1 su prikazane matrice *usporedi* za proteine koje mi promatramo za prozore duljine 5 i 10. Zapravo, to je matrica koja nam na mjestu (i_1, i_2) pokazuje koliko su *wind*-torke sa početkom na mjestima i_1 u polju $tocke_1$ i i_2 u polju $tocke_2$ bliske.

5. Postavlja se pitanje kako odrediti što je relativno blisko. U tu svrhu, navodimo dva načina.

   a) Pomoću postotka relevantnih: zadamo proizvoljan postotak, npr. 10% relevantno bliskih i odaberemo toliko najmanjih elemenata matrice *usporedi*.

   b) Drugi način, koji smo mi koristili u algoritmu, jest pomoću apsolutne vrijednosti. Kažemo da su neke točke relativno bliske ako je vrijednost elemenata u matrici *usporedi* manja od neke fiksne vrijednosti **trash** koju zadamo. U radu je korištena vrijednost $trash = 25$ za dva različita proteina, dok je prilikom testiranja na dva jednaka proteina bilo prikladnije uzeti vrijednost $trash = 10$. Intuitivno govoreći, što uzmemo manju vrijednost varijable *trash*, to će rezultati poravnanja kasnije biti bolji za proteine velike sličnosti, odnosno odgovarat će očekivanim rezultatima.



Slika 3.1: Matrica *usporedi* dva proteina za prozor duljine 5. Plavom bojom su označene male vrijednosti elemenata, a crvenom velike vrijednosti

Pomoću dosad navedenog, definiramo novu matricu pomoću matrice *usporedi* koja se sastoji od nula i jedinica **ma01** $\in M_{n_1-wind+1,n_2-wind+1}(\{0,1\})$ sa:

$$ma01_{i_1,i_2} = \begin{cases} 1, & usporedi_{i_1,i_2} \leq trash \\ 0, & usporedi_{i_1,i_2} > trash. \end{cases}$$

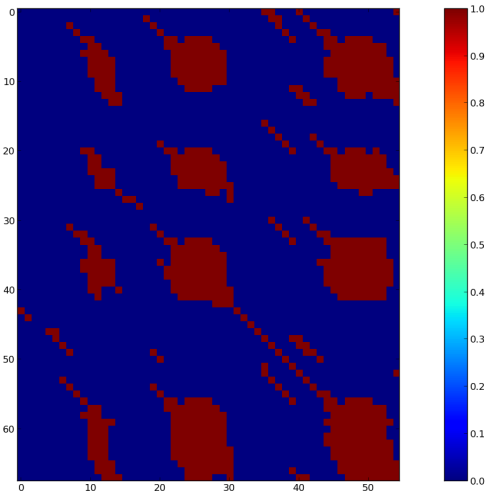
Na slikama 3.2 i 3.3 su prikazane matrice *ma*01 za različite vrijednosti parametra *trash*. Crvenom bojom su označene vrijednosti jedan, a sa plavom nula.

6. Sljedeće što želimo je identificirati uzastopne točke iz dva skupa točaka, i to one koje si međusobnim udaljenostima odgovaraju. Ono što tražimo su zapravo lanci jedinica u matrici *ma*01 pa defininiramo novi parametar **dlanac** koji označava duljinu. Tražimo nizove parova indeksa te zadane duljine.

$$(i_1, i_2), (i_1 + 1, i_2 + 1), \ldots, (i_1 + dlanac - 1, i_2 + dlanac - 1).$$

Ostale jedinice brišemo iz matrice *ma*01 i tako dobijamo novu matricu **Lanci**. U radu je korištena vrijednost $dlanac = 15$.

7. Slijedi korak u kojemu sve parove indeksa koje u matrici *Lanci*, koji imaju vrijednost jedan, spremamo u novo polje **matchhh**. Pretpostavimo da takvih parova ima N.



Slika 3.2: Matrica *ma*01 dva proteina za vrijednost parametra $trash = 50$. Jedinica ima 792

Znači da je:

$$matchhh : \{1, \ldots, N\} \to \{1, \ldots, n_1 - wind + 1\} \times \{1, \ldots, n_2 - wind + 1\}$$

Iz ovog polja parova želimo izdvojiti najdulji niz konzistentnih parova. Kasnije ćemo iz ovog polja izbacivati parove koji su najlošiji prema ostalima.

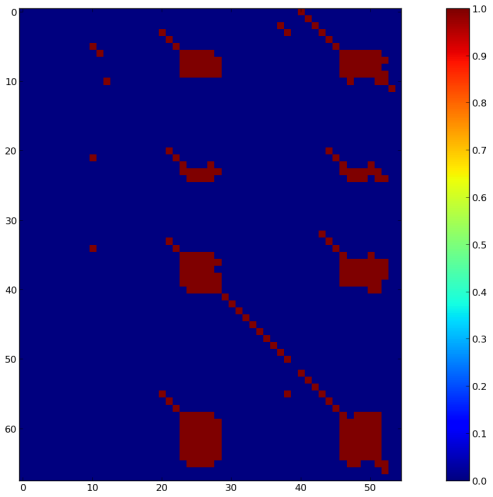
8. Definiramo novu matricu **matkand** $\in M_{N\times N}$ sa:

$$matkand_{i,j} = \begin{cases} 0, & i = j \\ penal, & i \neq j \ \& \ \min\{d_{1,i,j}, d_{2,i,j}\} = 0 \\ \dfrac{|d_{1,i,j} - d_{2,i,j}|}{\min\{d_{1,i,j}, d_{2,i,j}\}}, & i \neq j \ \& \ \min\{d_{1,i,j}, d_{2,i,j}\} \neq 0 \end{cases}$$

gdje smo označili:

$$d_{1,i,j} = d(tocke_1(matchhh(i)_1), tocke_1(matchhh(j)_1)),$$

$$d_{2,i,j} = d(tocke_2(matchhh(i)_2), tocke_1(matchhh(j)_2))$$



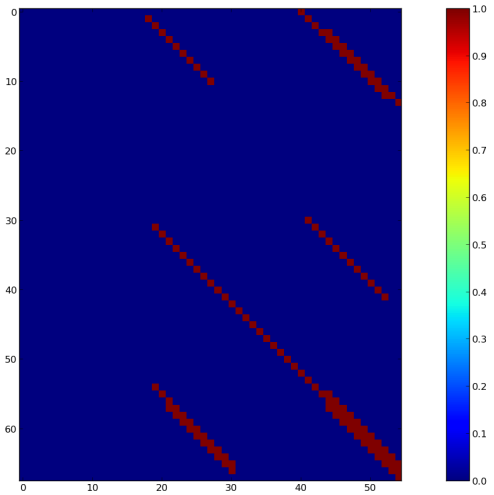
Slika 3.3: Matrica *ma*01 dva proteina za vrijednost parametra $trash = 25$. Jedinica ima 278

a *penal* je neka dovoljno velika vrijednost. U radu smo koristili $penal = 10$. Element (i, j) matrice *matkand* nam govori koliko su kompatibilne točke koje odgovaraju $i-$tom paru $(matchhh(i)_1, matchhh(i)_2)$ i $j-$tom paru $(matchhh(i)_1, matchhh(i)_2)$ indeksa. $machhh(i)_1$ i $matchhh(j)_1$ odgovaraju indeksima u polju $tocke_1$, a $matchhh(i)_2$ i $matchhh(j)_2$ u polju točaka $tocke_2$. Ako te točke međusobno odgovaraju onda je udaljenost dviju točaka iz prvog polja $tocke_1$ bliska udaljenosti dviju točaka iz polja $tocke_2$.

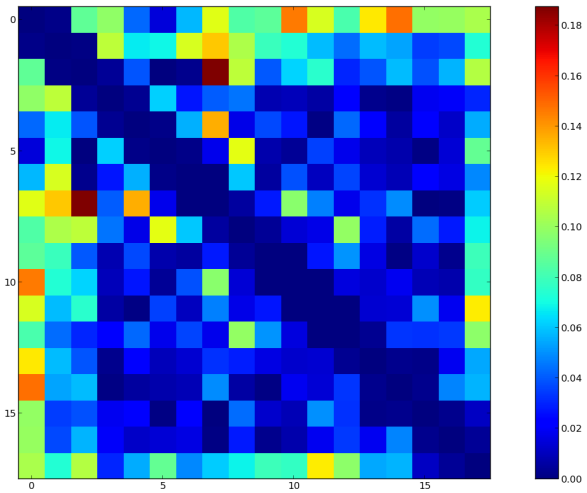
9. Kako smo najavili ranije, sada želimo iz polja *matchhh* izbaciti onaj par indeksa koji je najlošiji prema ostalima. Takav par odredimo tako da u matrici *matkand* prosumiramo stupce, a zatim podijelimo sa trenutnom duljinom polja *matchhh*. Tako definiramo polje realnih brojeva **kvaliteta** duljine N.

$$kvaliteta_i = \frac{\sum_{j=1}^{\#(matchhh)} matkand_{j,i}}{\#(matchhh)}.$$

Indeks najvećeg elementa u tom polju je indeks para u polju *matchhh* kojeg izbacujemo. Nakon toga dobijemo polje parova indeksa duljine $N - 1$ pa ponavljamo postupak. Postupak ponavljamo dok nismo zadovoljni rezultatom. Zapravo, brišemo

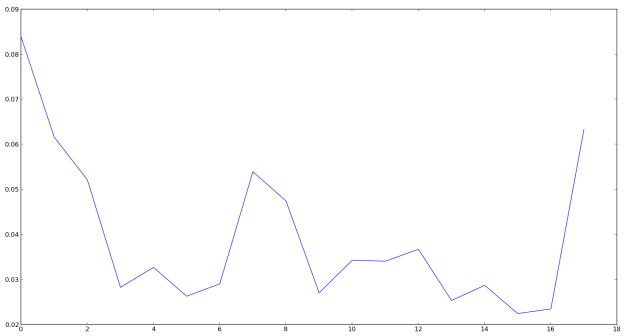


Slika 3.4: Matrica *Lanci* dva proteina za vrijednost parametra $dlanac = 10$

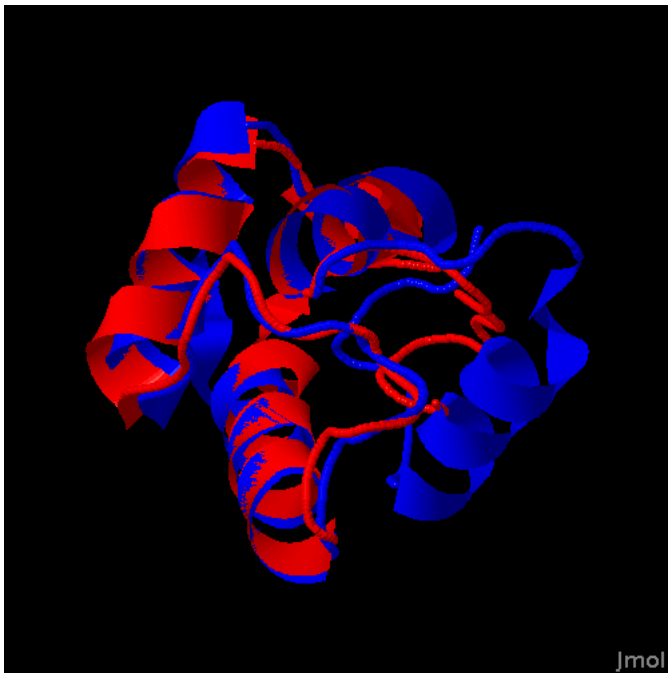


Slika 3.5: Matrica *matkand* za vrijednost parametara $trash = 25$ i $dlanac = 15$

par iz polja *matchhh* a u matrici *matkand* brišemo redak i stupac koji odgovara tom paru. Još uvijek je otvoreno pitanje koliko želimo parova da nam ostane na kraju.



Slika 3.6: Polje *kvaliteta*



Slika 3.7: Ilustracija poravnatih 1.10.8.50_2gybM01 i 1.10.8.50_1mu5A02 proteina

Bibliografija

[1] Barešić A.,(2005.) Skriveni Markovljevi modeli i biblioteka I-sites. Diplomski rad. Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Biološki odsjek.

[2] http://en.wikipedia.org/wiki/Structural_alignment

[3] http://zhanglab.ccmb.med.umich.edu/TM-align/tmp/846666.html (rujan 2014.)

[4] Tambača J., izbacivanje.pdf dostupno na priloženom CD-u.

[5] Vlahović R.,(2014.)Matematičko modeliranje u biologiji. Diplomski rad. Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Matematički odsjek.

[6] Zagorščak M.,(2012.) Matrice udaljenosti, reducirani skriveni Markovljevi modeli i usporedba sekundarnih struktura proteina. Diplomski rad. Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Biološki odsjek.

Sažetak

U ovom radu opisan je jedan algoritam za poravnanje proteinske strukture različitih duljina. Na početku smo definirali matrice udaljenosti točaka za dvije proteinske strukture zasebno. Pomoću tih matrica smo napravili novu koja uspoređuje njihove vektore određene duljine (koju smo sami zadali parametrom *wind*) i govori koliko su bliski.

Kako bismo odredili što je relativno blisko, uveli smo novi parametar (*trash*), te sve vrijednosti koje su manje od parametra postavili na 1, a ostale na 0. Nakon toga smo tražili uzastopne pozicije, koje u matrici imaju vrijednost 1, i koje zapravo predstavljaju točke iz dva skupa točaka sa početka. Pozicije najduljih nizova smo spremili u polje iz kojeg smo kasnije izbacivali one koji su najlošiji prema ostalima.

Tim postupkom smo došli do pozicija u proteinima koje međusobno imaju najmanje udaljenosti iz kojih zaključujemo na kojim dijelovima se proteini podudaraju, odnosno na kojim dijelovima su poravnati.

Summary

This paper describes an algorithm for protein structure alignment of different lengths. At the beginning, we define the matrix of distances between points for two protein structures separately. Using these matrices, we made a new one that compares their vectors of a certain length (which we have set with parameter *wind*) and tells us how close they are.

To determine which is relatively close, we introduced a new parameter ($trash$), and all values that are less than the parameter set to 1, and the rest to 0. After that we were looking for consecutive positions, which in the matrix have a value 1, and that actually represent points from the two sets of points from the beginning. Positions of longest strings are stored in the field, from which we later evicted those who are worst to others.

With this process, we come to positions in proteins that have the least distance from each other. From that, we conclude what parts of the proteins coincide, ie which parts are aligned.

Životopis

Ivana Kurolt rođena je 21. rujna 1988. u Banja Luci, BiH. U Varždinu pohađa 1. Osnovnu školu, a nakon toga Prvu gimnaziju Varaždin po programu prirodoslovno-matematičke gimnazije. Studij matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu upisala je 2007. godine. Na istom fakultetu 2012. godine upisala je diplomski studij Primijenjena matematika.