

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1877

VIZUALIZACIJA STRUKTURE PROTEINA

Toni Sente

Zagreb, lipanj 2019.

Sažetak

Na ovu stranicu treba upisati tekst sažetka diplomskog rada ne duži od 150 riječi.

Tekst diplomskog rada piše se u fontu Arial veličine 12 s proredom od 1,5 linije. Poravnat je s obje strane (justified). Lijeva margina namještena je na 3,5 cm, a ostale na 3 cm. Format papira je A4.

Sadržaj

Popis oznaka i kratica

B	magnetska indukcija
engl	engleski
EUROSTAT	Statistički ured Europske Unije
FER	Fakultet elektrotehnike i računarstva, Zagreb
f_{er}	funkcija doza-učinak (engl. Function Exposition Response)
KS	kratki spoj
TE	termoelektrana
US DOE	Ministarstvo energetike SAD-a (od engl. naziva United States Department of Energy)
API	Application Programming Interface
“spline” funkcija sastoji se od polinoma definiranih na podintervalima, funkcije su povezane preko granica intervala koristeći različite relacije neprekidnosti (https://www.pmf.unizg.hr/download/repository/nmmm8.pdf)	

Popis tablica

Po završetku rada treba generirati popis svih tablica:

Insert ⇒ Reference ⇒ Index and Tables ⇒ Table of Figures
(potrebno je koristiti *Caption label*: **Tablica**)

Popis slika

Po završetku rada treba generirati popis svih slika:

Insert ⇒ Reference ⇒ Index and Tables ⇒ Table of Figures
(potrebno je koristiti *Caption label*: **Slika**)

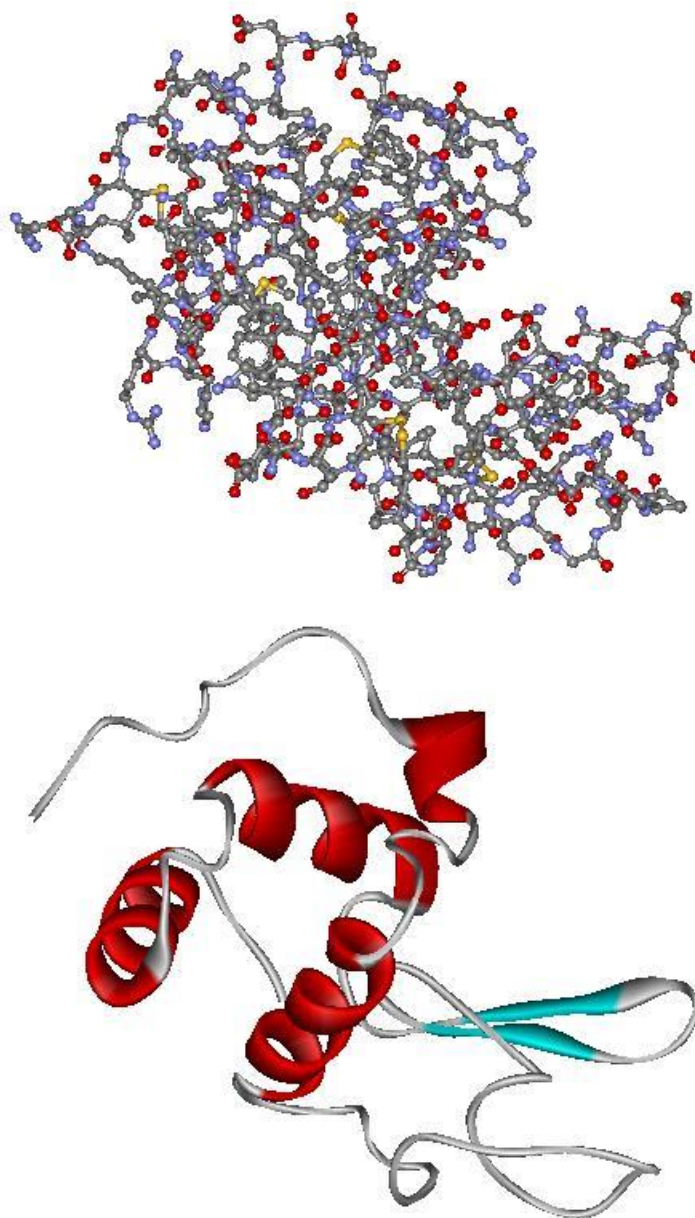
1. Uvod

Proteini su jedni od glavnih makromolekula prisutnih u organizmu koje sudjeluju u gotovo svakom biološkom procesu kao što su prijenos i spremanje kemijskih spojeva, kataliziranje kemijskih reakcija, prevođenje i signalizacija informacija od strane drugih proteina, očuvanje strukture stanica i tkiva, pretvorba kemijske energije u mehaničku (pokretanje mišića) te mnogi drugi. (ct: <https://neos-guide.org/content/protein-structure>) Funkcija proteina striktno je određena njegovom 3D strukturom(ct: essential knjiga), stoga je područje istraživanja utjecaja 3D strukture na funkcionalnost proteina trenutno vrlo aktivno područje u bioinformatici. Jednom kada nam je jasan utjecaj 3D strukture na funkciju proteina, otvara se mogućnost ručnog dizajniranja proteina koji bi na točno određeni način djelovali na druge proteine što bi u konačnici dovelo do lakšeg i jednostavnijeg razvoja raznih lijekova(ct: proteinShader).

Strukturu (pozicije atoma) i izgled proteina je moguće odrediti korištenjem nekoliko različitih metoda (X-ray kristalografija, nmr spektroskopja, elektron mikroskopija). Te informacije je zatim moguće iskoristiti za 3D rekonstrukciju proteina u nekom od alata za računalnu vizualizaciju proteina što nam u konačnici omogućava da paralelno uspoređujemo više proteina. Ovaj način istraživanja je vrlo čest jer poznate funkcionalnosti jednog proteina možemo preslikati na slične ili identične dijelove drugog proteina i na kraju djelomično ili u potpunosti shvatiti funkciju novog proteina. Dobar alat za vizualizaciju je ovdje od velike važnosti jer mala vizualna razlika može uvelike utjecati na konačnu funkcionalnost proteina.

Prije računalnih programa za vizualizaciju, molekule su se najčešće prikazivale kao fizikalni modeli sa kuglama i štapićima,[slika](ct: essential) međutim razvoj računalne grafike omogućio je puno sofisticiraniju vizualizaciju u kojoj korisnik može interaktivno pregledavati i analizirati pojedinačne dijelove kompliciranijih molekula u 3D prostoru. Uz to, programi za vizualizaciju najčešće omogućuju različite stilove vizualnog prikaza proteina koji olakšavaju

proučavanje funkcionalnosti. Jedan od najčešće korištenih stilova je korištenjem vrpca koji olakšavaju prikaz tercijarne strukture proteina.[slika]



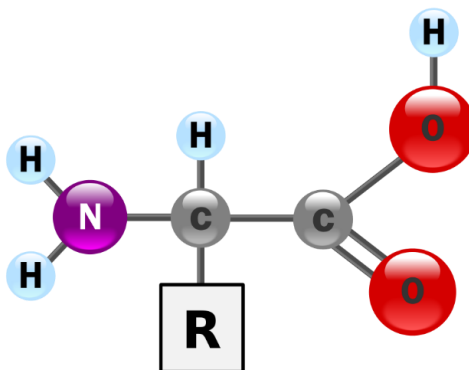
Tema ovog rada bit će prikazati izvedbu programa koji na ulazu prima zapis strukture proteina na temelju kojeg generira interaktivni 3D model. U 2. poglavlju će za lakše razumijevanje najprije biti objašnjene biološke i kemijske osnove proteina. Potom će u poglavlju 3. ukratko biti opisani različiti zapisi strukture proteina od kojih će detaljnije biti objašnjen najpopularniji, .PDB format. U poglavlju 4. će biti razrađeni osnovni dijelovi programa potrebni za

generiranje i prikaz modela proteina, nakon čega će u 5. poglavlju biti prikazana usporedba dobivenih rezultata sa rezultatima već postojećih alata. U poglavlju 6 su opisane kratke upute za korištenje programa dok su zaključak, literatura i sažetak dani u poglavljima 7., 8. i 9.

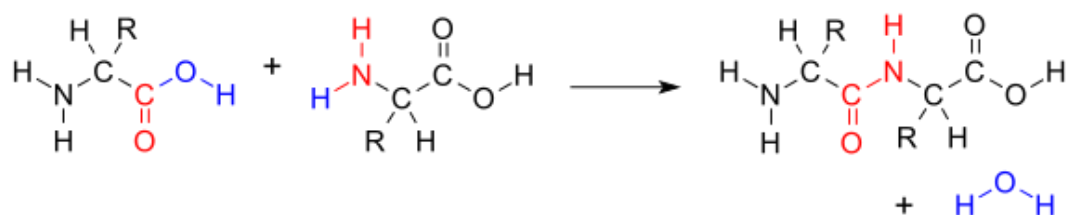
2. Proteini

Proteini ili bjelančevine su makromolekule koje se sastoje od niza aminokiselina, međusobno povezanih poput karika u lancu. Protein se može sastojati od više takvih lanaca, a redoslijed i broj aminokiselina u pojedinom lancu određuje specifične osobine svakog proteina. Njihova prva i osnovna zadaća je proces rasta i razvoja, a odgovorni su i za nadomještanje oštećenih i odumrlih stanica te služe kao enzimi za ubrzavanje biokemijskih procesa. (ct: <https://hr.wikipedia.org/wiki/Bjelan%C4%8Devine>)

Aminokiseline su molekule koje imaju slobodnu amino grupu (NH_2) i karboksilnu skupinu (COOH) gdje su obje skupine vezane na središnji atom ugljika ($\text{C}\alpha$) na koji je još vezan i bočni lanac R koji određuje svojstva svake aminokiseline (slika).



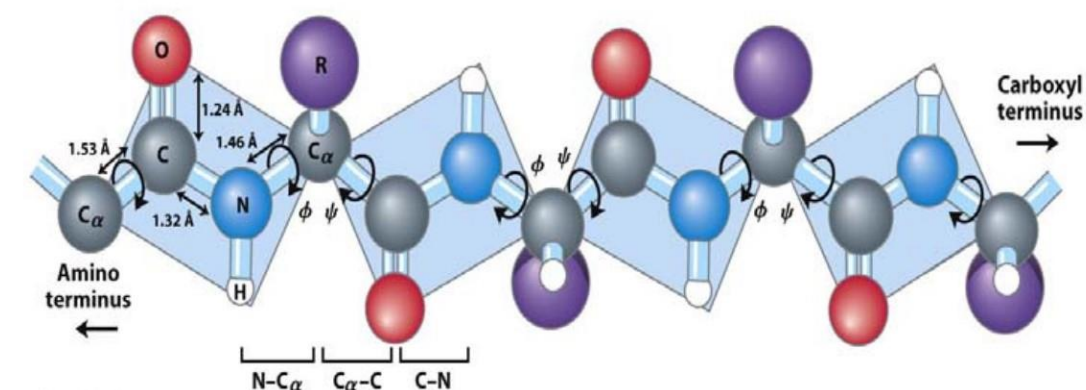
Aminokiseline se mogu međusobno povezati formiranjem peptidne veze te tako stvoriti polipeptidni lanac - protein. Peptidna veza koja spaja dvije aminokiseline je kovalentna veza koja nastaje između karboksilne skupine jedne i amino skupine druge aminokiseline te se pri tome oslobađa jedna molekula vode.



Zbog lakšeg proučavanja, struktura proteina podijeljena je na 4 različite razine: primarnu, sekundarnu, tercijarnu te kvartarnu.

2.1. Primarna struktura

Primarna struktura proučava redoslijed aminokiselina u pojedinom lancu proteina. Lanac je usmjeren, a započinje amino krajem te završava karboksilnim (C) krajem peptida. Tri veze odvajaju susjedne $C\alpha$ atome. $C\alpha - C$ i $N - C\alpha$ mogu rotirati, dok peptidna veza $CO = NH$ ne (slika). Ove rotacije omogućavaju proteinima da se nabiru na različite načine što omogućava velik broj različitih oblika. (ct: 14obk-p4)



2.2. Sekundarna struktura

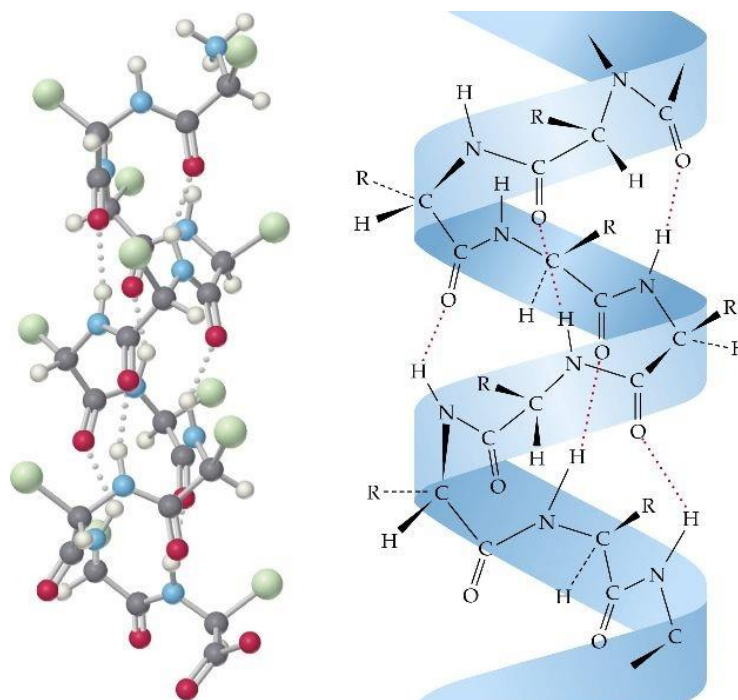
Sekundarna struktura opisuje prostorni raspored glavnih atoma okosnice ($C\alpha$), a da se pri tome ne uzima u obzir konformacija bočnih ogranaka aminokiselina.

Veliki broj načina rotacija peptidnog lanca pridonosi stvaranju nekih pravilnih tvorevina od kojih su najbitnije α -zavojnice te β -lanci. Uz njih, u sekundarnu strukturu se još ubrajaju petlje te oštri zavoji (β -zavoji).

2.2.1. α -zavojnica

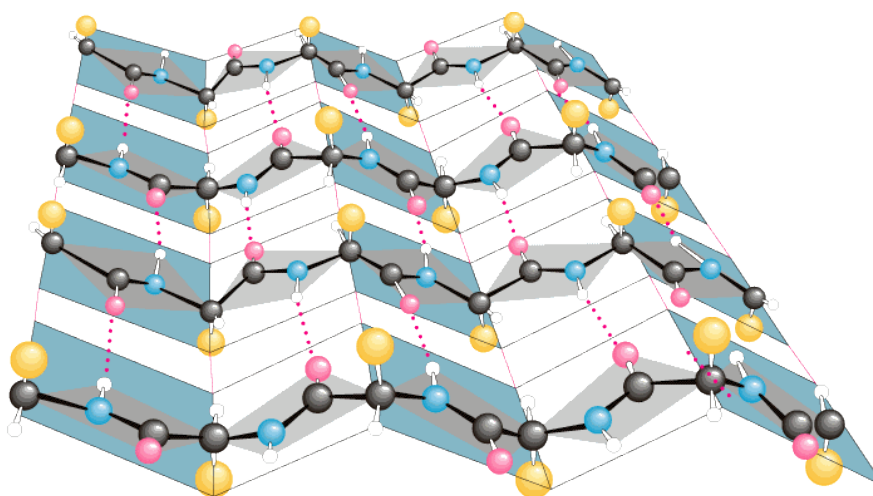
U α -zavojnici, polipeptidni lanac zavija u lijevu ili desnu stranu i stvara strukturu čvrsto pakiranog valjka (slika). Do takve strukture dolazi zbog stvaranja vodikovih veza između svakog četvrtog centralnog, alfa atoma ugljika ($C\alpha$)(ct::essential). Smjer zavojnice može biti i lijevi i desni, međutim

desni smjer je energetski puno povoljniji te su zavojnice sa lijevim smjerom iznimno rijetke.(ct: 14obk-p5)



2.2.2. β -naborana ploča

Za razliku od α -zavojnice, β -naborana ploča je izdužena tvorevina koju čine dva ili više β -lanca, međusobno povezani vodikovom vezom između amino i karboksilne skupine.(slika) Lanci aminokiselina koji su međusobno povezani u β -naboranoj ploči mogu biti paralelni, antiparalelni ili mogu biti kombinacija paralelnih i antiparalelnih lanaca.



2.3. Tercijarna struktura

Cjelokupni 3D raspored svih sekundarnih struktura jednog polipeptidnog lanca definira tercijarnu strukturu te se pri tome se najviše gleda interakcija između R-grupa svih aminokiselina koje čine protein. U tu se interakciju ubrajaju vodikove veze, ionske veze, dipol-dipol interakcije te londonove disperzijske sile. Također, ovdje su vrlo bitne i hidrofilne te hidrofobne interakcije koje određuju na koji će način protein reagirati s vodom. (cit: <https://www.khanacademy.org/science/biology/macromolecules/proteins-and-amino-acids/a/orders-of-protein-structure>).

2.4. Kvarterna struktura

Proteini koji se sastoje od samo jednog polipeptidnog lanca imaju samo prve tri strukture, dok proteini koji su sačinjeni od dva ili više lanca imaju još i kvarternu strukturu. Općenito, u kvarternoj strukturi se gledaju iste interakcije i stvaranje veza kao i u tercijarnoj strukturi, no umjesto na jednom lancu, ovdje se promatraju međudjelovanja jednog lanca na drugi.

2.5. Stabilnost proteina

Kako je cjelokupna 3D struktura (sekundarna, tercijarna te kvarterna) proteina određena slabim silama, proteini su vrlo osjetljivi te su lako podložni poremećaju strukture ili denaturaciji. Ukoliko se naruši njihova struktura, tada oni više nisu u stanju obavljati svoju osnovnu funkciju što dovodi do poremećaja stanične aktivnosti te moguće smrti stanice. (cit: [https://hr.wikipedia.org/wiki/Denaturacija_\(biokemija\)](https://hr.wikipedia.org/wiki/Denaturacija_(biokemija))).

2.6. Određivanje 3D strukture proteina

Određivanje strukture proteina najčešće se vrši dvjema popularnim metodama: rendgenskom kristalografijom (engl. *x-ray crystallography*) te nuklearnom magnetskom rezonancijom (engl. *Nuclear Magnetic Resonance Spectroscopy, NMR*).

Analiza rendgenskom kristalografijom radi se na način da se protein najprije zamrzne kako bi fiksirao položaj atoma. Potom taj uzorak bombardiramo X zrakama koje se odbijaju od elektronskih oblaka obližnjih atoma te prilikom difrakcije proizvode regularne uzorke koje potom zabilježimo. Iz tih podataka, korištenjem matematičkih alata te heurističkih metoda, možemo iz dobivenih rezultata očitati položaj atoma u uzorku. (ct: essential)

Drugi najčešći način određivanja strukture je spektroskopija nuklearnom magnetskom rezonancijom. Ova metoda mjeri promjene magnetskog momenta jezgre u vanjskom magnetskom polju na temelju kojih se mogu dobiti podaci o strukturnim i dinamičkim svojstvima molekula bilo da su slobodne ili vezane (ct: <http://physics.mef.hr/Predavanja/nmr/prva.html>, 11.6)

3. Zapisi proteina

Jednom kada se odredi struktura proteina, sve sakupljene informacije potrebno je zapisati u neku formatiranu datoteku. Trenutno su u uporabi koriste dva formata: stariji PDB te noviji mmCIF.

Protein Data Bank (skraćeno PDB) format je osmišljen ranih 1970ih godina te je inicijalno bio napravljen da bude kompatibilan sa FORTRAN programskim jezikom. Format ima strogu strukturu od 80 znakova u jednoj liniji gdje je svaka linija poseban zapis. Datoteka započinje zaglavljem u kojem su opisane razne informacije o proteinu kao što su ime molekule, podrijetlo organizma, način određivanja strukture, kristalografski parametri i rezolucija, položaj sekundarnih struktura i sl. (ct: essential) Svaki redak započinje nekom od ključnih riječi nakon kojeg slijede podaci koji su strukturirani ovisno o toj ključnoj riječi. Neke od ključnih riječi i njihovo objašnjenje dano je u nastavku (ct: tablica)

Ključna riječ	Značenje
HEADER	Zaglavlje
TITLE	Naziv molekule
REMARK	Općenite informacije u slobodnom obliku

MODEL	Označava početak jednog modela proteina (format dozvoljava više zapisa iste molekule)
ATOM	Informacije o jednom atomu u molekuli
HETATM	Heterogeni atom koji ne pripadaju standardnim grupama atoma, npr. za opis atoma molekule vode koja ne pripada molekuli proteina
TER	Kraj zapisa atoma jednog lanca aminokiseline
ENDMDL	Kraj informacija za jedan model proteina (dolazi u paru sa MODEL oznakom)

Stupac	Sadržaj	Tip podatka
1-4	„ATOM“	Znakovi
7-11	Serijski broj atoma	Cijeli broj (integer)
13-16	Ime atoma	Znakovi
17	Indikator alternativne lokacije	Znak
18-20	Ime člana (aminokiseline)	Znakovi
22	Identifikator lanca	Znak
23-26	Redni broj člana (aminokiseline)	Cijeli broj (integer)
27	Kod za umetanje člana	Znak
31-38	X koordinata atoma	Realni broj (float 8.3)
39-46	Y koordinata atoma	Realni broj (float 8.3)
47-54	Z koordinata atoma	Realni broj (float 8.3)
55-60	Zauzeće	Realni broj (float 6.2)
61-66	Temperaturni faktor	Realni broj (6.2)
73-76	Identifikator segmenta	Znakovi

77-78	Simbol elementa	Znakovi
-------	-----------------	---------

PDB format već je dugo u uporabi te ga je lako čitati i koristiti, međutim, format nije dizajniran za računalno izvlačenje podataka koje je sve češće potrebno prilikom pretraživanja baze podataka. Nadalje, pojedine restrikcije su podosta zakomplicirale njegovo korištenje, npr. u formatu se navode samo koordinatne pozicije atoma ali ne i njihove veze što stvara problem prilikom određivanja disulfidnih veza koje nisu zapisane u datoteci već ih program mora sam odrediti (neki programi ni ne uspiju). Još neke negativne strane formata su npr. veličina stupca za serijski broj atoma koji iznosi 5 brojeva čime se stavlja ograničenje da jedan model proteina može imati maksimalno 99 999 atoma. Također, identifikator lanca je definiran samo sa jednim znakom što znači da cjelokupna molekula proteina može imati najviše 26 lanaca (Identifikator lanca je određen velikim slovom engleske abecede)(ct: essential). Sva ova ograničenja stvaraju probleme prilikom rada sa velikim proteinskim kompleksima koji zbog tih ograničenja moraju biti podijeljeni na više datoteka stoga je daljnji razvoj ovog formata zaustavljen 2012 godine te se podaci u PDB formatu više ni ne prihvataju u Protein Data Bank bazi podataka(ct: <http://www.wwpdb.org/documentation/file-formats-and-the-pdb>)

Ograničenja PDB dovela su do razvoja novih formata od kojih je najpopularniji *macromolecule crystallographic information file* ili kraće mmCIF. Format se temelji na *crystallographic information file* (CIF) formatu koji se koristi za zapis manjih molekula te je proširen informacijama potrebnim za prikaz makro molekula(ct: https://www.rcsb.org/pdb/static.do?p=file_formats/mmcif/index.html). mmCIF podržava razne vrste informacija koje su definirane u mmCIF rječniku, te atributi i njihove vrijednosti, za razliku od PDB formata, mogu biti dugački i deskriptivni.

Jednom kada je struktura proteina određena i zapisana u mmCIF format, datoteka se može objaviti na *Protein Data Bank* (skraćeno PDB) repozitoriju – jedinom svjetskom repozitorij za procesiranje i distribuciju podataka o trodimenzionalnim strukturama velikih molekula (makromolekula) i nukleinskih

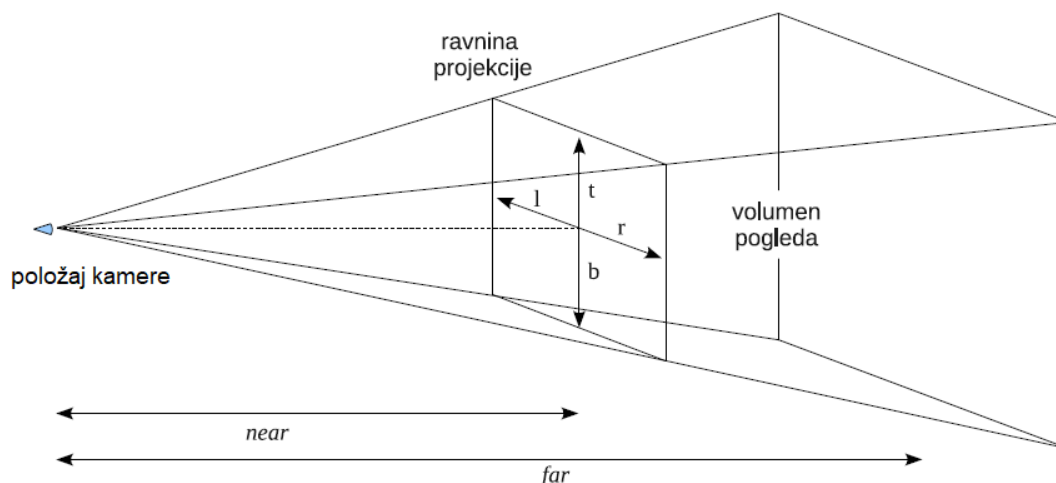
kiselina. Repozitorij u trenutku pisanja broji više od 152 000 struktura, te su svi podaci besplatno dostupni javnosti (cit: https://www.rcsb.org/pdb/static.do?p=general_information/about_pdb/index.html, 11.06).

4. Program za vizualizaciju proteina

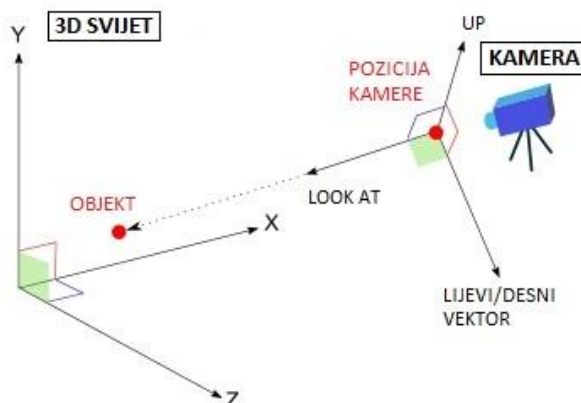
Cjelokupni program sastoji se od mnogo dijelova, a ovdje će biti opisani samo najvažniji i najkompliciraniji dijelovi programa.

4.1. Kamera

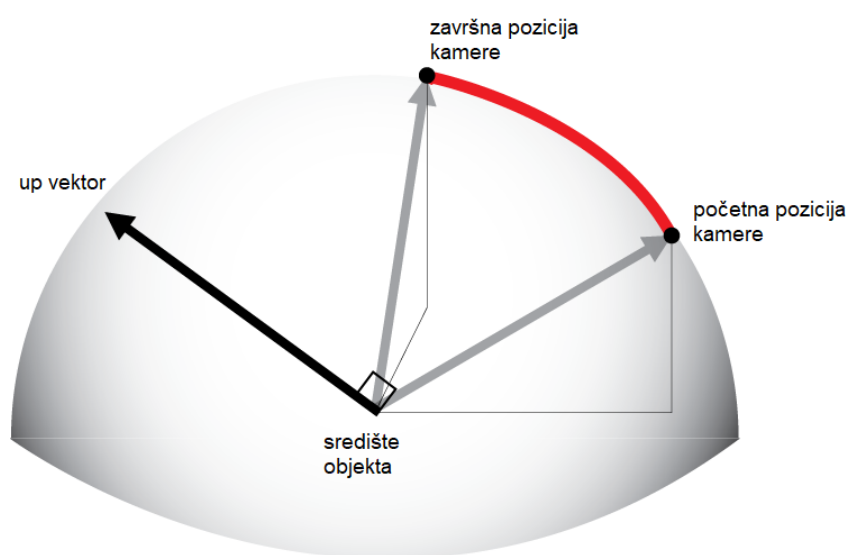
Kamera u 3D grafici jedan od glavnih aspekata programa. Isto kao i ostali objekti, kamera ima svoj položaj ali i orijentaciju te neke druge, trenutno manje važne, parametre. Osnove 3D kamere u računalnoj grafici prikazane su na slikama



Orijentacija kamere je jednako je važna kao i pozicija, a određena je pomoću dva vektora. Prvi vektor je „look at“ vektor koji je usmjeren od položaja kamere prema nekom objektu. Taj vektor prolazi kroz središte bliže i daljnje ravnine odsijecanja koje određuju međuprostor koji se u konačnici prikazuje na zaslonu. Drugi vektor je takozvani „up“ vektor čija je svrha dati pravilnu orijentaciju kamere. Vektorskim produktom „up“ i „look at“ vektora dobije se još i treći bočni vektor koji je usmjeren bočno od kamere. Ukoliko se koristi lijevi koordinatni sustav, bočni vektor će pokazivati desno od kamere, a ukoliko se koristi desni koordinatni sustav, bočni vektor će pokazivati lijevo od kamere.

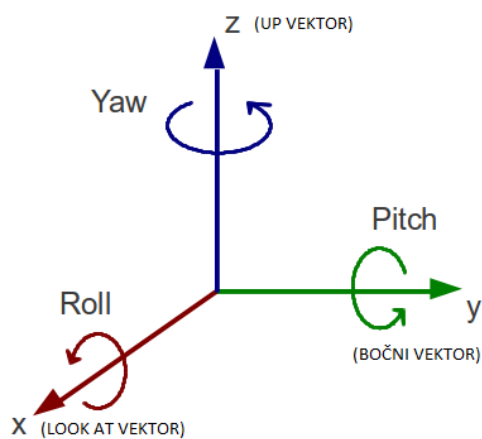


Za detaljniju analizu i proučavanje pojedinačnog objekta, kao što su u ovom slučaju proteini, najčešće se koristiti tehnika u kojoj se objekt istraživanja stavlja u središte prostora, a kamera se rotira oko njega po zamišljenoj sferi (slika). Ekvivalentni prikaz mogli bismo dobiti ako, umjesto pozicije kamere, rotiramo model oko njegovog središta



Kontrola kamera se najčešće i najlakše kontrolira korištenjem miša. Kako bi realizirali ranije opisanu tehniku, program prvo treba registrirati određen pomak miša na temelju kojeg se potom određuje smjer i količina rotacije. Kada miš pomičemo lijevo ili desno, očekujemo da se kamera zarotira ulijevo odnosno udesno oko proteina (tj. da se protein zarotira u suprotnom smjeru, oko svog središta). Međutim, kako je kamera objekt u prostoru koji može imati

proizvoljnu poziciju i koordinate, moramo biti pažljivi da ne radimo rotaciju korištenjem globalnih koordinatnih osi. Ispravan način je zapravo koristiti koordinatne osi kamere („up“, „look at“ i desni vektor). Tako horizontalnu rotaciju ulijevo i udesno dobijemo na sljedeći način: vektor koji pokazuje od središta objekta prema poziciji kamere, nazovimo ga ObjektKamera vektor, koji je kolinearne sa „look at“ vektorom te ujedno i okomit na „up“ vektor, rotiramo oko „up“ vektora postavljenog u središte objekta. Ova rotacija naziva se još i yaw rotacija (slika).



Prilikom te bočne, odnosno yaw, rotacije, smijer „up“ vektor se neće promijeniti (jer on čini rotacijsku os), no bočni vektor hoće, međutim kako bočni vektor u svakom trenutku možemo dobiti vektorskim produktom „up“ i „look at“ vektora, nema potrebe da radimo dodatnu rotaciju.

Ekvivalentno, vertikalnu rotaciju dobijemo rotacijom ObjektKamera vektora oko bočnog vektora kamere postavljenog u središte objekta. Ova rotacija se još i naziva pitch rotacija. Međutim, prilikom ove rotacije moramo biti oprezni jer, osim rotacije ObjektKamera vektora moramo, korištenjem iste rotacijske matrice, rotirati i up vektor kako bi osi kamere ostale konzistentne i okomite jedne na drugu.

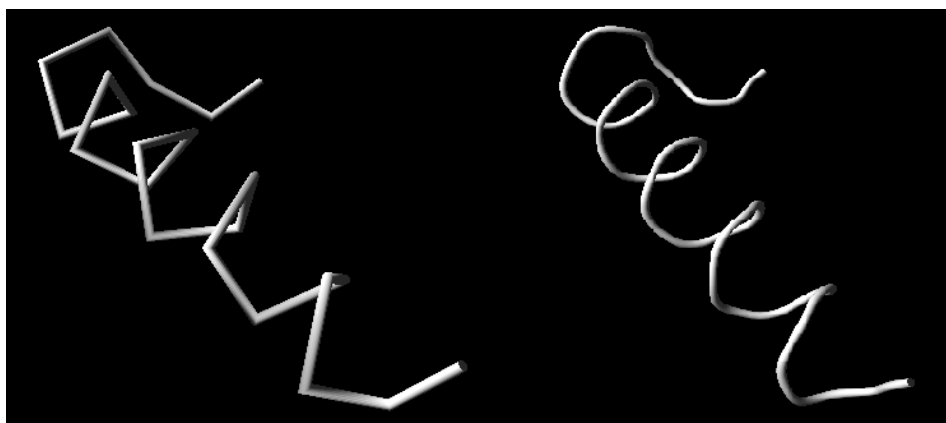
Kako su rotacije u 3D grafici poprilično opširna i kompleksna tema, da ne ulazimo preduboko, spomenut ćemo samo kako gotovo sva API sučelja za računalnu grafiku imaju ugrađene metode kojima je moguće dobiti potrebnu rotacijsku matricu za rotaciju jednog vektora oko drugog.

Osim rotacije, koja čini najkompleksniji dio kamere, potrebno je još i realizirati zumiranje te pomicanje odnosno translaciju objekta. Zumiranje je vrlo jednostavno izvesti. Kako je ranije rečeno, kamera se miče po zamišljenoj sferi čiji je središte u središtu objekta, a polumjer određen duljinom ObjektKamera vektora. Ukoliko se želimo približiti ili udaljiti, potrebno je jednostavno smanjiti odnosno povećati polumjer sfere.

Na početku pokretanja programa, središte objekta ujedno je i fokus kamere, međutim, ukoliko cijeli objekt želimo pomaknut u stranu, tada središte objekta više neće biti fokus kamere. Ako cijeli objekt želimo pomaknuti horizontalno, to možemo učiniti tako da točku fokusa pomaknemo korištenjem bočnog vektora kamere, a ako želimo pomaknut vertikalno, to činimo korištenjem „up“ vektora.

4.2. Izrada 3D modela proteina

Kada bi prikazivali svaki pojedini atom proteina korištenjem metode kuglica i štapića, prikaz bi vrlo brzo postao prekompleksan, stoga se danas najčešće koristi prikaz okosnice zajedno sa vizualno izraženijim sekundarnim strukturama. Okosnica je sačinjena od alfa atoma ugljika svake aminokiseline u lancu, a kako bi cijela struktura izgledala glađe i prirodnije, umjesto direktnog spajanja atoma okosnice, kroz njih se često provlači krivulja(slika). Osi za okosnicu, ista se krivulja koristi i kod generiranja alfa zavojnice i beta lanaca



4.2.1. Krivulje

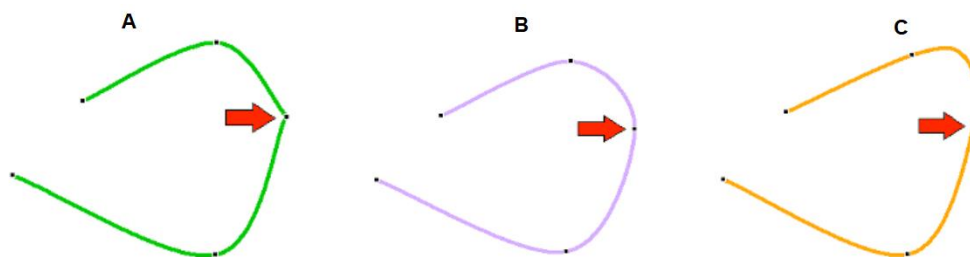
Matematički gledano, krivulja je neprekidna crta, ili točnije rečeno, jednodimenzionalni skup točaka(ct: <https://hr.wikipedia.org/wiki/Krivulja>).

Krivulje se obično zadaju analitički te imamo tri osnovne kategorije zapisa (ct: irg):

- Eksplicitnom jednačinom $y = f(x)$
- Implicitnom jednačinom $f(x, y) = 0$
- Parametarskim zapisom, npr. $x = \cos(t)$, $y = \sin(t)$

Pojedini zapisi imaju određene probleme pa tako naprimjer eksplicitnom jednačinom $y = ax + b$ ne možemo prikazati pravac paralelan s y-osi. Još jedan problem je nemogućnost prikazivanja funkcija s višestrukim vrijednostima. Implicitne jednačine rješavaju ove nedostatke, no one pak imaju problem što ne mogu prikazivati djelomičan prikaz

Jedno od važnijih svojstava krivulja je svojstvo neprekinutosti pa tako postoje C kontinuiteti krivulja: C^0 kontinuitet zahtjeva neprekinutost u koordinata, C^1 kontinuitet zahtjeva neprekinutost prve derivacije – drugim riječima, ne smije biti šiljaka, već krivulja mora biti glatka, C^2 zahtjeva neprekinutost druge derivacije u svim točkama – to nam osigurava neprekinutost zakrivljenosti, C^3 zahtjeva neprekinutost treće derivacije u svim točkama itd. (ct: irg)(slika)

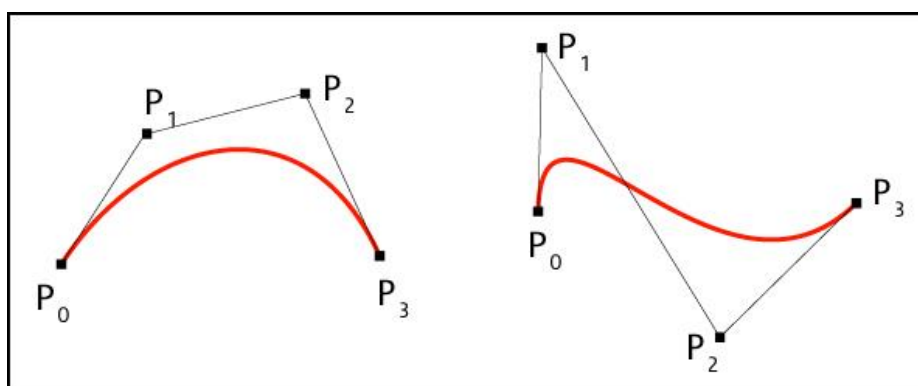


Krivulje se u računalnoj grafici najčešće konstruiraju korištenjem kontrolnih točaka i težinskih polinoma. Ideja je sljedeća: svaka kontrolna točka doprinosit će ukupnom obliku krivulje svojim koordinatama pomnoženim s pripadajućom težinskom funkcijom (koja daje vrijednost od 0 do 1):

$$T_K = \sum_{i=0}^n f_i(t) \cdot T_i$$

gdje je T_K točka krivulje, T_i i -ta zadana kontrolna točka, $f_i(t)$ težinska funkcija za i -tu točku, a $t \in [0, 1]$.

Ovisno o potrebi, postoje razni načini kako definirati težinske funkcije, međutim ovako zadana krivulja neće prolaziti kroz kontrolne točke (osim prve i zadnje). Na primjer, na slici (slika), crvenom bojom prikazane su dvije Bezierove krivulje, dok P_0 , P_1 , P_2 i P_3 predstavljaju kontrolne točke. Na slici se jasno vidi kako kontrolne točke „privlače“ smjer krivulje u određenu stranu. Težinske funkcije pak služe da definiraju jačinu utjecaja pojedinih kontrolnih točaka.

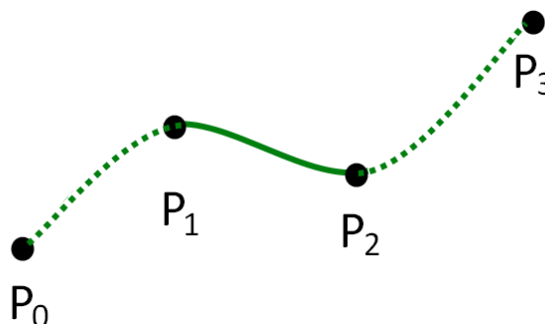


Krivulje koje ne prolaze kroz dane točke nazivaju se aproksimacijskim krivuljama te nisu dobar odabir za naše potrebe. Još jedna velika mana ovako zadanih krivulja je u tome što su težinske funkcije polinomi čiji stupa ovisi o broju kontrolnih točaka. Ukoliko imamo četiri kontrolne točke, kao na slici (slika), tada težinske funkcije trebaju biti trećeg stupnja, odnosno ukoliko imamo n kontrolnih točaka, težinske funkcije trebaju biti $n-1$ stupnja što može biti vrlo nepovoljno ako imamo puno točaka.

Uz pomoć algebarskih operacija, moguće je iz ranije navedenog zapisa doći do funkcije koja generira krivulju koja prolazi danim točkama. Općenito, takve krivulje se nazivaju interpolacijskim te su upravo ono što trebamo za generiranje okosnice proteina. Međutim, i dalje ostaje problem polinoma velikog stupnja. Problem bi bilo najbolje riješiti kada bismo spojili više manjih dijelova krivulje u jednu veću krivulju. Međutim, tu treba biti pažljiv da spajanje krivulja bude glatko tako da krivulja zadrži (barem) C^1 kontinuitet. Srećom, postoji upravo takva vrsta krivulja, a radi se o Catmull-Rom splajnu.

Catmull-Rom splajn definirali su Edwin Catmull i Raphael Rom (cit: https://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline).

Radi se o interpolirajućoj krivulji definiranoj sa 4 kontrolne točke P_0 , P_1 , P_2 i P_3 gdje je krivulja definirana samo između P_1 i P_2 (slika)



Krivulja je definirana sljedećom funkcijom:

$$P(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -S & 2-S & S-2 & S \\ 2S & S-3 & 3-2S & -S \\ -S & 0 & S & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}, t \in [0,1]$$

gdje parametar $S \in [0,1]$ predstavlja napetost krivulje. Za $S = 0$, krivulja izgleda kao ravan pravac između dvije točke dok za $S = 1$ krivulja postaje vrlo zaobljena. Za optimalne rezultate, najbolje je koristiti vrijednost između 0.5 i 1.

Odlično svojstvo Catmull-Rom krivulje je to što dvije susjedne krivulje zadržavaju C^1 kontinuitet što u konačnici omogućuje postepenu izgradnju proizvoljno velike krivulje koja na cijelom području zadržava C^1 kontinuitet. (cit: <http://algorithmist.net/docs/catmullrom.pdf>). Međutim, i ova vrsta krivulja ima svoje probleme. Ako koristimo samo dani skup kontrolnih točaka kroz koje krivulja treba interpolirati, tada nećemo moći generirati dio krivulje između prve i zadnje dvije točke. Ovo se može riješiti na nekoliko načina: Prvi je se korisnik ručno generira dodatne točke što često nije poželjno jer želimo da cijeli proces bude automatiziran. Drugi način je da jednostavno dupliciramo početnu i završnu točku, no to ponekad uzrokuje čudne završetke, te treći, najčešće korišteni način, je da napravimo refleksiju druge točke preko prve, odnosno

predzadnje točke preko zadnje(ct: <http://algorithmist.net/docs/catmullrom.pdf>). Ovaj način u praksi daje najbolje rezultate.

Kako cjelokupna krivulja, sastavljena od više manjih Catmull-Rom krivulja, ima C^1 kontinuitet, to znači da je možemo u bilo kojoj točki derivirati i time dobiti tangentu koja nam pokazuje trenutni smjer krivulje. To možemo odrediti ako deriviramo ranije napisanu formulu, čime se dobije sljedeći izraz:

$$P'(t) = \begin{bmatrix} -3St^2 + 4St - S \\ 3(2-S)t^2 + 2(s-3)t \\ 3(S-2)t^2 + 2(3-2S)t + S \\ 3St^2 - 2St \end{bmatrix}^T \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

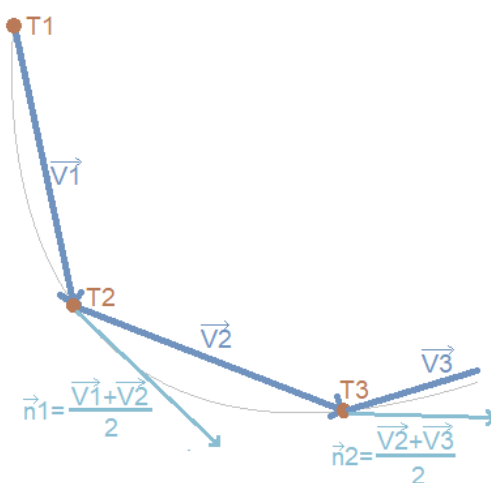
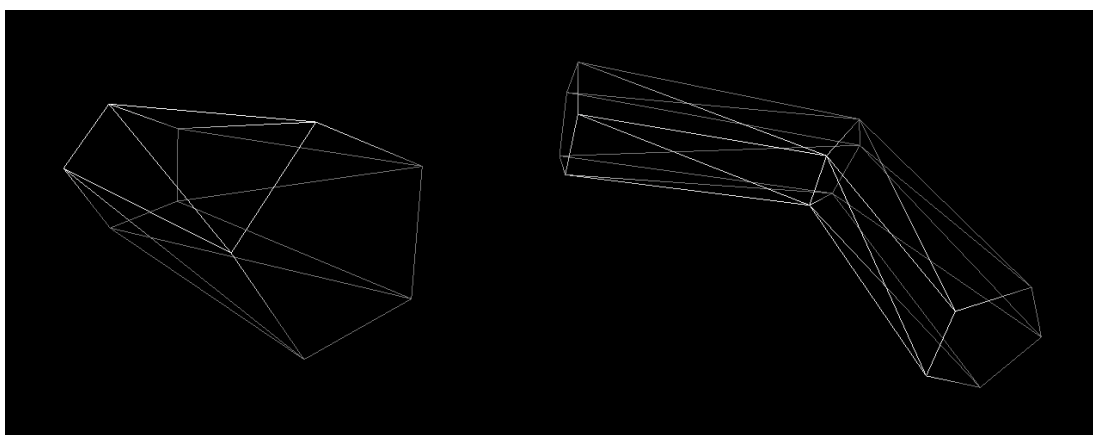
4.2.2. Izrada okosnice proteina

Jednom kada znamo konstruirati krivulju koja prolazi kroz točno određene točke, možemo tu krivulju iskoristiti kao vodilju za konstruiranje 3D struktura koje zajedno daju cjelokupni model proteina. Najvažnija od tih struktura je centralna okosnica (engl. *wireframe*) već sama po sebi može biti dovoljna za analizu proteina. Geometrijski gledano, okosnica se najčešće modelira kao jednostavna, zaobljena cijev koja prati centralnu krivulju. Kako se u grafici sve modelira korištenjem ravnih poligona, nije moguće generirati u potpunosti glatku cijev, no privid glatke površine možemo ostvariti na način da na dovoljno malim razmacima konstruiramo ravne, spojene cijevi, dok ravnu cijev generiramo kao izduženi n-terokut (slika). Generiranje obične ravne cijevi ne predstavlja neki veliki problem. Problem se javlja prilikom spajanja dviju cijevi. Kako bi izbjegli nepotrebno dupliciranje točaka na spojevima, potrebno je malo prostorne geometrije (slika); Na početku, potrebno je generirati bazne točke početnog n-terokuta na ravnini određenoj vektorom normale $V1$ i točkom $T1$. Jednom kada imamo skup početnih točaka ponavljamo sljedeći postupak:

1. Odredimo vektor \vec{n}_i normale ravnine koja dijeli dva susjedna segmenta kao $\frac{\vec{v}_{i-1} + \vec{v}_i}{2}$ (ukoliko se radi o posljednjoj točki na krivulji, za normalu uzmemo posljednji vektor)

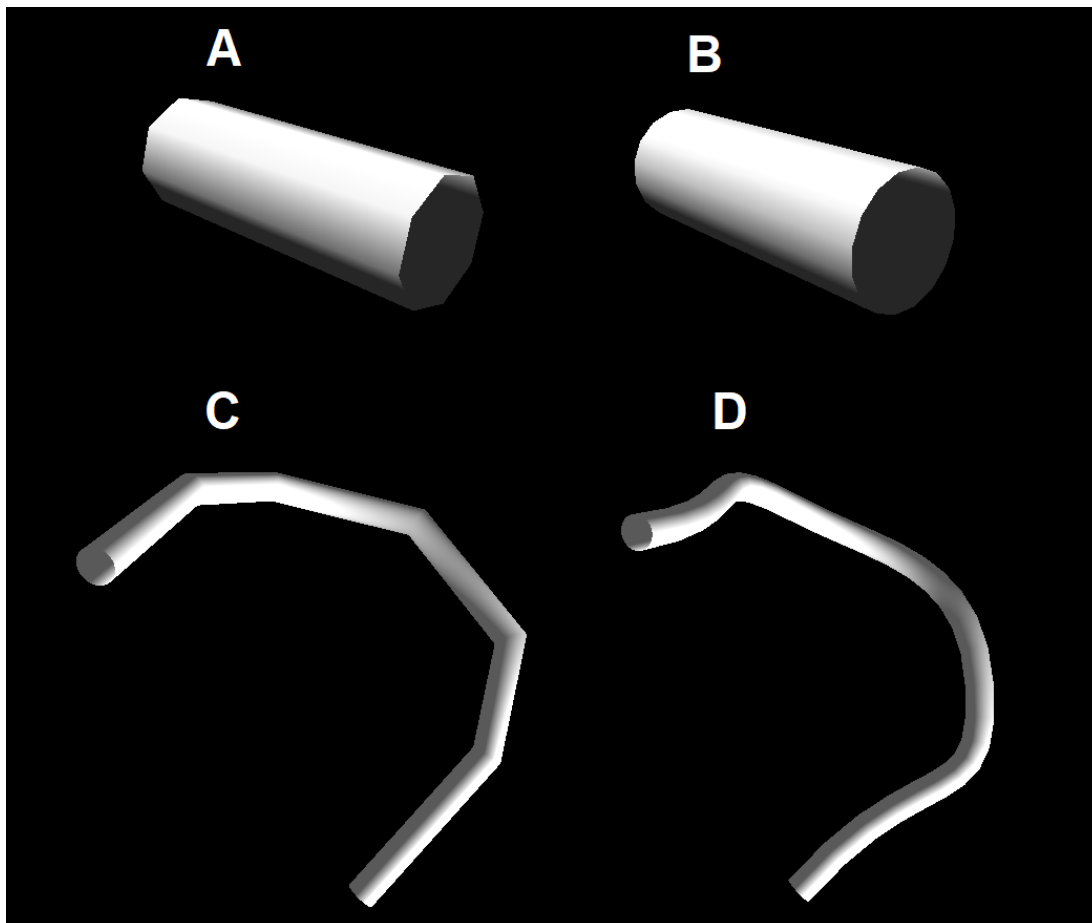
2. Za svaku baznu točku $B_{j,i-1}$ iz prijašnje iteracije, odredimo novu baznu točku $B_{j,i}$ koja nastane kao probodište pravca određenog tom prijašnjom baznom točkom i vektora $\overrightarrow{v_{i-1}}$ sa ravninom koja je definirana ranije određenim vektorom normale \vec{n}_i i točkom T_i

Recimo da se nalazimo u točki T2 na slici (slika). Točka T2 predstavlja spoj cijevi određen točkama T1 i T2, te T2 i T3. U prvom koraku potrebno je odrediti vektor normale \vec{n}_2 u točki T2. Tu normalu dobijemo kao $\vec{n}_2 = \frac{\vec{v}_1 + \vec{v}_2}{2}$. Sada kada imamo ravninu koja čini spoj dviju cijevi, preostaje nam kroz točke prijašnjeg spoja povući pravac, čiji je smjer određen smjerom vektora \vec{v}_1 , te izračunati probodište tog pravca sa ravninom.



Glatkoća zakrivljenost pojedine cijevi određena je veličinom početnog n-terokuta, a glatkoća cjelokupne cijevi određena je brojem centralnih točaka generiranih između dviju kontrolnih točaka krivulje. Na slici (slika), na A dijelu slike nalazi se jedna cijev kreirana pomoću pet baznih točaka, tj. peterokuta.,

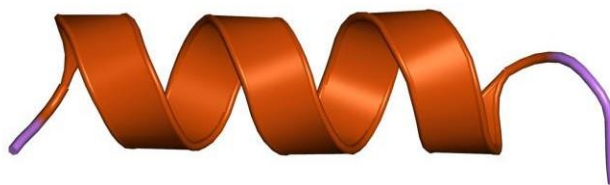
dok je na B dijelu slike cijev generirana pomoću 16 baznih točaka (šesnaesterokut). Na C dijelu slike je generirana cjelokupna cijevi pri čemu je između kontrolnih točaka krivulje generirana samo jedna dodatna točka, dok je na D dijelu slike prikazana ista cijev ali ovaj put sa 10 generiranih točaka između dviju kontrolnih točaka krivulje.



Type equation here.

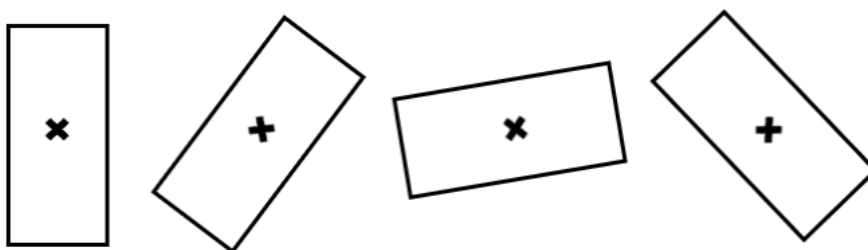
4.2.3. Alfa zavojnice

Ovisno o programu, sekundarne strukture, među kojima je alfa zavojnica, prikazuju se na različite načine, međutim najčešći prikaz ostvaruje se korištenjem vrpce (engl. *ribbons*) (slika)



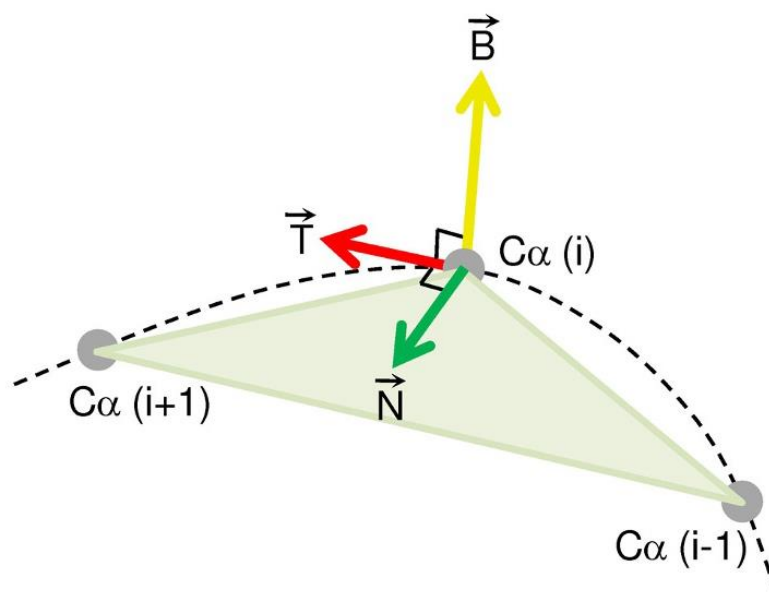
Izrada ovakvih struktura temelji se isto na krivulji koja prolazi kroz centralne (alfa) atome ugljika svake aminokiseline. Ti atomi su dovoljno gusto raspoređeni tako da Catmull-Rom krivulja, uz dovoljno velik koeficijent zakrivljenosti, čini kružnu zavoјnicu, bez potrebe za dodatnim matematičkim popravcima.

Ovakav tip strukture možemo generirati na način da povlačimo (engl. *sweeping*) određen geometrijski oblik, najčešće kvadrat ili uzduženu elipsu po središnjoj krivulji. Problem tog postupka je u tome što u određenoj točki krivulje možemo dobiti samo smjer (tangentu) krivulje te ne možemo odrediti što je „uspravno“. Ako pogledamo sliku (slika), znakom „x“ u sredini svakog kvadrata označen je vektor koji okomito „ulazi“ u svaki pravokutnik. Taj vektor možemo zamisliti da predstavlja tangentu krivulje u proizvoljnoj točki. Iz slike možemo vidjeti da postoji beskonačno mnogo načina rotacije tog pravokutnika, stoga nam je u točki krivulje, osim smjera, potreban još jedan vektor koji pokazuje pravilnu orijentaciju pravokutnika u proizvoljnoj točki –slično kao i „up“ vektor kod kamere. Jednom kada imamo taj drugi vektor koji određuje orijentaciju, vektorskim produktom možemo dobiti i treći vektor i time u konačnici imamo kompletan lokalni koordinatni sustav u svakoj točki krivulje kojim možemo pravilno generirati točke kvadrata kojeg „provlačimo“ po krivulji.



Jedan način kako dobiti je sljedeći: za izračun lokalnog koordinatnog sustava u i -tom alfa atomu ugljika ($C\alpha(i)$) potrebno je poznavati prethodni $C\alpha(i-1)$.

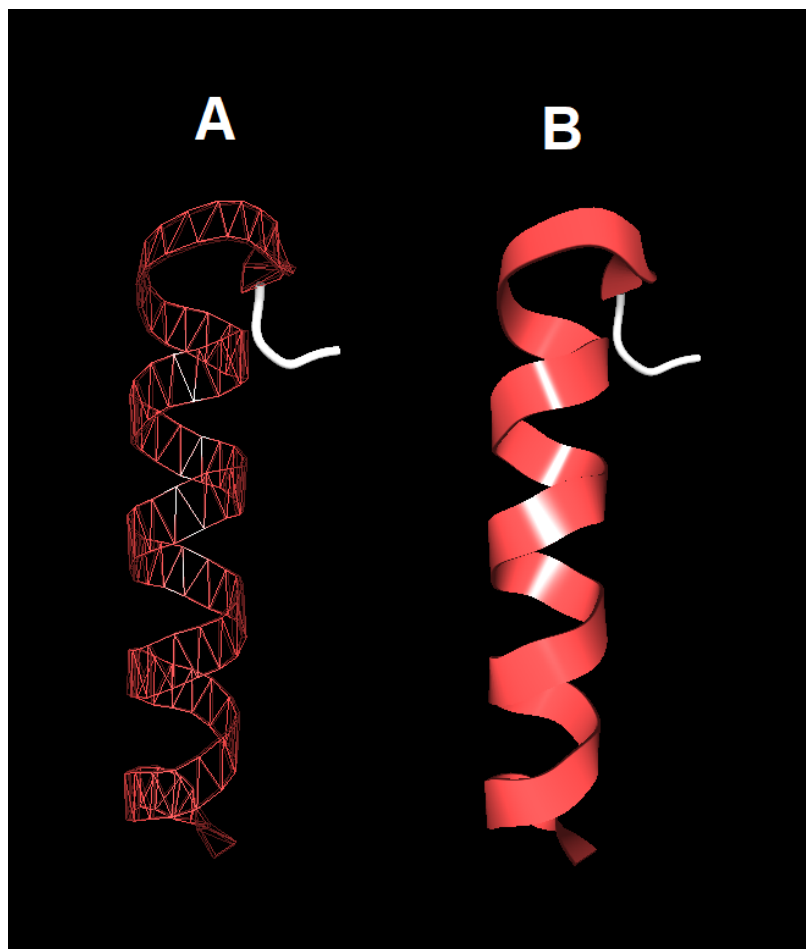
1) te sljedeći $C\alpha(i+1)$ atom. Vektor smjera \vec{T} možemo odrediti korištenjem derivirane formule za Catmull-Rom krivulju ili alternativno, možemo ga definirati kao vektor od $C\alpha(i-1)$ prema $C\alpha(i+1)$ atomu. Kao ključan drugi vektor možemo uzeti vektor normale ravnine određene sa tria susjednim atomima ugljika Taj vektor je na slici označen kao žuti vektor \vec{B} . Konačno treći, bočni vektor, označen oznakom \vec{N} , na slici možemo dobiti kao vektorski produkt vektora \vec{B} i \vec{T} .



Kada izračunamo lokalne koordinatne sustave za dva susjedna atoma alfa ugljika, korištenjem jednostavne linearne interpolacije možemo izračunati pojedinačne lokalne koordinatni osi, i na taj način dobiti cjelokupni sustav za bilo koju točku između ta dva atoma. Linearnu interpolaciju za lokalne koordinatne sustave S_1 i S_2 za parametar $t \in [0,1]$ dobijemo prema sljedećoj formuli:

$$S(t) = (1 - t) \cdot S_1 + t \cdot S_2$$

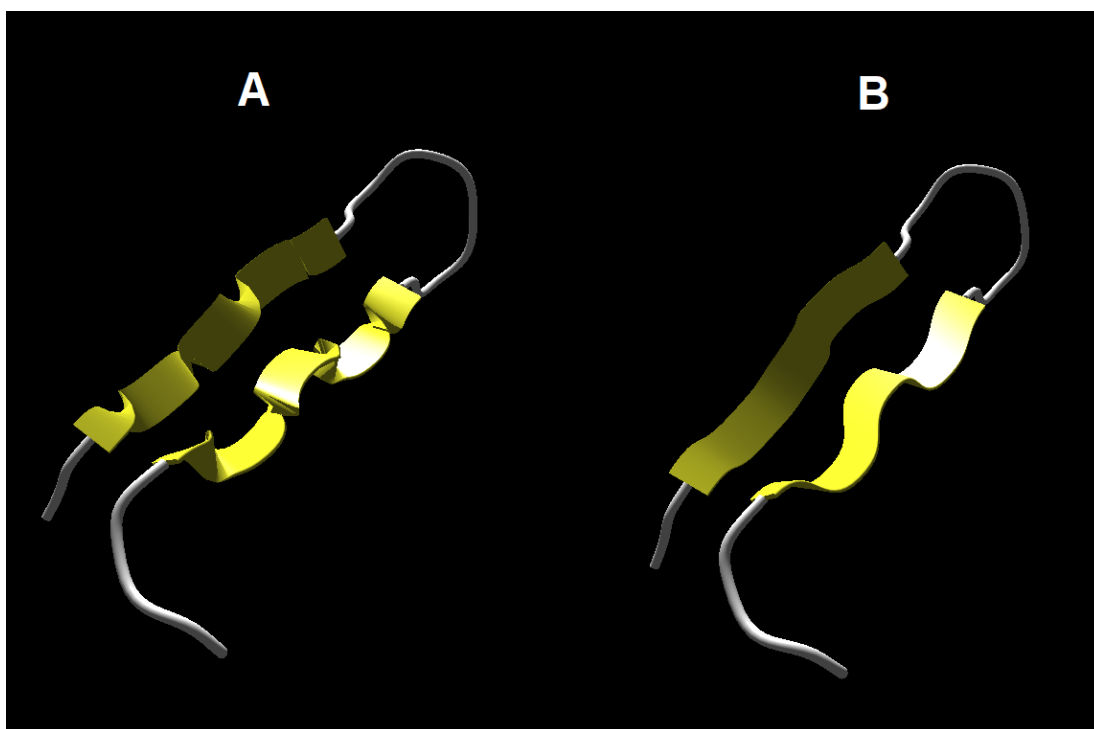
Iz formule je jasno vidljivo za parametar $t = 0$ dobijemo koordinatni sustav koji je jednak sustavu S_1 , dok za vrijednost parametra $t = 1$, interpolirani sustav će biti jednak sustavu S_2 . Ovakav pristup generira dovoljno dobre rezultate koji su vidljivi na slici (slika). Na A dijelu slike je prikazan kostur (engl. *wireframe*) modela, dok je na B dijelu slike prikazan puni model



4.2.4. β -lanci

Beta lanci koji čine beta naboranu ploču, isto se kao i alfa zavojnice, najčešće prikazuju kao vrpce konstruirane pomoću krivulje koja prolazi kroz alfa atome ugljika. stoga se ranije opisana tehnika stvaranja vrpca može primijeniti i za generiranje beta lanaca, no uz jednu manju promjenu. Naime, prilikom generiranja lokalnih koordinatnih sustava za svaki $C\alpha$ koji je dio beta lanca, vrlo često se dogodi da se susjedni sustavi drastično razlikuju, najčešće u tome da su koordinatne osi susjednog sustava zarotirane za 180° oko vektora smjera pružanja (z-os). Prilikom generiranja zavojnice ovo nije bio problem jer se točke zavojnice cijelo vrijeme pružaju u istu stranu. Ovaj problem generira vrpcu koja je vrlo nazubljena (slika A) Kako bi ispravili ovaj problem, potrebno je u algoritam dodati dodatnu provjeru koja provjerava ako je veličina kuta između bočnih osi (x-os) dvaju susjednih lokalnih koordinatnih sustava veća od 90° . U slučaju da je, rotiramo jedan od sustava za 180° oko

njegove z-osi(ct: proteinshader). Konačno, ova promjena rezultira glatkim modelom β -lanaca prikazanim na slici (slika) B



5. Usporedba s postojećim alatima

Popis alata koji mogu prikazati tercijarnu i kvartarnu strukturu proteina poprilično je dugačak – od osnovnih alata čija je namjena općeniti prikaz molekula, pa do kompleksnih alata koji su specijalizirani za proteine. Kako je nemoguće napraviti usporedbu sa svim alatima u nastavku su odabrani i opisani neki od popularnijih alata

6. ZAKLJUČAK

U zaključku se sažimaju rezultati diplomskog rada. Sadrži maksimalno 1 stranicu, a kandidat vlastoručno potpisuje diplomski rad iza zaključka.

7. Literatura