Univerzita Karlova v Praze

Přírodovědecká fakulta

BAKALÁŘSKÁ PRÁCE

Jakub Nierostek

Počítačové studium skládání proteinů na zjednodušených modelech

Katedra fyzikální a makromolekulární chemie

Vedoucí bakalářské práce: doc. RNDr. Filip Uhlík, Ph.D.

Studijní program: Chemie

Studijní obor: Chemie

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ....

Podpis autora

Na tomto místě bych rád srdečně poděkoval doc. RNDr. Filipu Uhlíkovi, Ph.D. za jeho velikou ochotu a pomoc, zahrnující mimo jiné mnoho hodin videohovorů a nespočet vyměněných mailů, bez nichž by sepsání této práce nebylo možné. Děkuji také rodině a svým blízkým, kteří mě celý čas podporovali.

Název práce: Počítačové studium skládání proteinů na zjednodušených modelech

Autor: Jakub Nierostek

Katedra: Katedra fyzikální a makromolekulární chemie

Vedoucí bakalářské práce: doc. RNDr. Filip Uhlík, Ph.D., Katedra fyzikální a makromolekulární chemie

Abstrakt: Cílem práce je navrhnout a popsat vhodný zhrubený model proteinu, na jehož základě bude studován protein-folding. Model bude implementován jako počítačový program, jeho vývoj v čase bude zajišťovat Hamiltonian Monte Carlo. Pomocí simulace na počítači bude zkoumán jak protein-folding samotný, tak veličiny, které jej charakterizují a podobnost nativní konfigurace skutečného a námi simulovaného proteinu.

Klíčová slova: skládání proteinů, počítačová simulace, Hamiltonian Monte Carlo, zhrubené modely

Title: Computer study of protein folding using simplified models

Author: Jakub Nierostek

Department: Department of Physical and Macromolecular Chemistry

Supervisor: doc. RNDr. Filip Uhlík, Ph.D., Department of Physical and Macromolecular Chemistry

Abstract: The aim of this work is to design and describe a suitable coarse-grained protein model, on the basis of which protein-folding will be studied. The model will be implemented as a computer program, development of the model in time will be simulated by Hamiltonian Monte Carlo. Using computer simulations, not only the protein-folding itself will be investigated, but also the quantities that characterize the process and the similarity of the real and simulated protein's native conformation.

Keywords: protein folding, computer simulation, Hamiltonian Monte Carlo, coarse-grained models

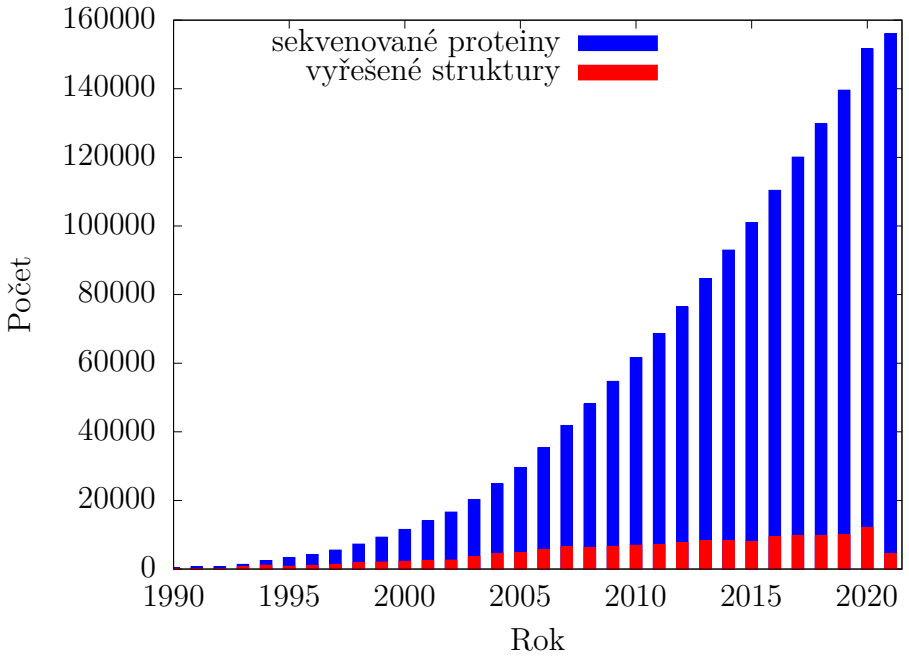
Obsah

1. Úvod

Řetězec aminokyselin, který právě vznikl na ribozomu procesem translace, se velmi rychle sbalí a zaujme dobře definovanou prostorovou strukturu. Ta je zcela klíčová pro jeho biologickou funkci. Právě díky své struktuře je *ferritinový komplex* schopen uchovávat ionty železa, *aktinové* vlákno se umí kontrahovat a *ATP-synthasa* dokáže zásobovat buňku energií.

I více než 50 let poté, co americký chemik Christian B. Anfinsen provedl pokus s reverzibilní denaturací *ribonucleasy*, za nějž obdržel roku 1972 Nobelovu cenu a započal tak éru zkoumání mechanismů sbalování proteinů, je celá řada otázek o průběhu tohoto děje dosud nezodpovězena [1].

Díky výkonnosti a dostupnosti sekvenovacích metod vzrůstá každým rokem rychlost sekvenování a s tím i počet známých proteinových sekvencí [2]. Stanovení struktury, zejména pomocí rentgenové difrakce, je oproti tomu zdlouhavé, finančně nákladné a z důvodu obtížné krystalizace některých proteinů také ne vždy úspěšné [3]. Rychlé a spolehlivé počítačové metody umožnující předpovědět strukturu sbaleného proteinu z jeho sekvence by umožnily překlenout zvětšující se rozdíl[1] mezi počtem známých sekvencí a vyřešených struktur (viz obrázek 1.1). Znalost nativní sekvence proteinů hraje také významnou roli při návrhu léčiv [4], interpretaci biologických dat a v biomedicíně [2].



Obrázek 1.1: Časové porovnání počtu sekvenovaných proteinů a vyřešených struktur

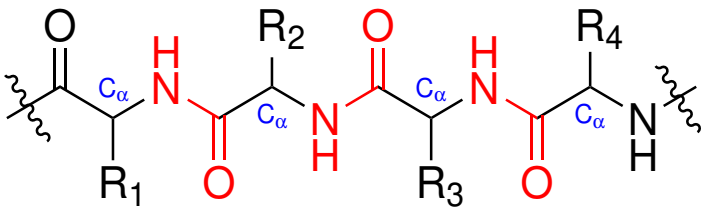
V této práci se pokusíme navrhnout zjednodušený model proteinu, na jeho základě provést počítačovou simulaci sbalování krátkých proteinů s méně než 100 aminokyselinami a výsledky porovnat s aktuálním výzkumem.

[1]Data převzata z https://www.rcsb.org/stats/growth/growth-protein

Než k tomu přistoupíme, seznámíme se v této kapitole s nutným biochemickým popisem skládání proteinů a shrneme současný stav problematiky.

1.1 Biochemie sbalování

Protein je lineární biomakromolekula, jejíž stavebními jednotkami jsou až na výjimky proteinogenní L-aminokyseliny. Ty jsou substitučními deriváty karboxylových kyselin, nesoucí jednu nebo více aminoskupin. Funkční skupiny $-\mathrm{COOH}$ a $-\mathrm{NH_2}$ jsou spojené přes C_α – alfa uhlík, který může nést postranní řetězec (viz obrázek 1.2) a je stereogenním centrem [5].



Obrázek 1.2: Část proteinového řetězce

Aminokyseliny jsou na ribozomu v procesu translace spojeny peptidickou vazbou $-\mathrm{NH}-\mathrm{CO}-$ (červeně). Lze je pojmenovat systematicky, převažují však názvy triviální. Každou proteinogenní aminokyselinu jde označit třípísmennou zkratkou, většinou tvořenou prvními třemi písmeny triviálního názvu. Pro zkrácení zápisu proteinového řetězce pak lze každou aminokyselinu pojmenovat jedním písmenem. Například *kyselinu 3-aminopropanouvou* označujeme triviálním názvem *glycin*, jíž přísluší označení *Gly* nebo *G*.

Aby mohl protein vykonávat svou biologickou funkci, musí zaujmout prostorové uspořádání zvané *nativní konformace*. V roce 1973 americký chemik Christian B. Anfinsen postuloval, že nativní struktura proteinu je určena pouze jeho primární strukturou, tedy sekvencí aminokyselin. Tato konformace je kineticky dosažitelná (při sbalování nedochází k vytváření uzlů, ani jiným složitým změnám) a přísluší ji globální minimum Gibbsovy energie. Toto tvrzení nese označení *Anfinsenovo dogma*, nebo také *termodynamická hypotéza*.

Sekundární struktura zahrnuje opakující se strukturní motivy jako α-helix, β-skládaný list nebo β-otáčka a terciální struktura zahrnuje interakce vzdálených částí řetězce.

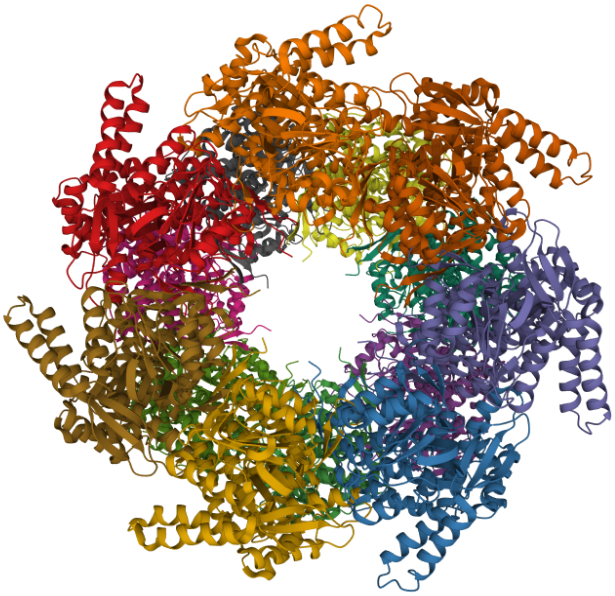
Peptidická vazba má rovinný charakter, v níž převažuje uspořádání *trans*, při němž jsou postranní řetězce sousedících aminokyselin orientovány vždy na opačné straně makromolekuly a dochází tak k menšímu stérickému bránění. Tvar řetězce lze definovat pomocí torzních a dihedrálních úhlů, tradičně označovaných φ a ψ a nabývajících hodnot z intervalu $[-180°,180°]$. Výnos dvojic úhlů (φ_i,ψ_i) označujeme jako Ramchadranův diagram. Z důvodu častého výskytu strukturních motivů jako α-helix a β-skládaný list jsou některé kombinace (φ_i,ψ_i) častější.

Jak poznamenal roku 1969 molekulární biolog Cyrus Levinthal, z důvodu velkého počtu stupňů volnosti proteinu existuje obrovské množství teoreticky možných konformací [6]. Mějme polypeptid složený ze 100 aminokyselin. Konformace proteinu je plně určena jeho dohromady 195 torzními a vazebnými úhly. Pokud

pro každý úhel předpokládáme 3 stabilní konformace, existuje prakticky nevyčíslitelných 3^{195} konformací polypeptidu. Nesoulad mezi vysokým počtem teoretických konformací a skutečností, že většina proteinů se od svého vzniku složí v řádu milisekund, bývá označován jako *Levinthalův paradox*.

Řešením tohoto paradoxu je hypotéza, že protein folding probíhá postupně v rámci několika kroků, při nichž se energie proteinu pozvolna snižuje. Nejprve dojde během rychlé reverzibilní reakce k vytvoření sekundárních struktur jako α-helix a β-skládaný list, tvořících krystalizační zárodky proteinu. Následně dochází ve stavu *roztavené globule* ke vzniku dalších strukturních motivů, včetně možných dysfunkčních struktur. Poté dochází už k jen malým závěrečným změnám, včetně vzniku disulfidových můstků [5].

K protein foldingu dochází v cytoplazmě, která je z více než 30% tvořena proteiny, což má za následek, že právě vytvořený protein nemá ke svému skládání ideální podmínky. Po celou dobu života proteinu může dojít k jeho agregaci s ostatními proteiny v cytoplazmě nebo k nežádoucí konformační změně – *misfoldingu*. K ochraně skládajících se proteinů slouží v buňce bílkoviny *chaperony* GroEL a GroES. Jedná se o enzymy, které za cenu ATP katalyzují proces skládání. GroEL je válcovitý protein, skládající se ze dvou heptamerních prstenců a obsahující vnitřní kavitu o průměru 45 Å (viz obrázek 1.3). V případě lapení proteinu dovnitř kavity se na GroEL za spotřeby ATP naváže chaperon GroES, tvořící víko, které drží protein uvnitř kavity. Uvnitř tohoto komplexu, někdy označovaného jako *Anfinsenova klec*, má pak protein vhodné podmínky ke svému složení do nativní konformace.



Obrázek 1.3: Chaperon GroEL (4V43). Struktura převzata z Protein Data Bank (PDB).

1.1.1 Stabilizující interakce

Při skládání nedochází k náhodnému vzorkování možných konformací. Existuje totiž několik interakcí, umožňující vznik a stabilizaci nativní struktury [5].

Některé postranní řetězce jsou při fyziologickém pH 7 protonované nebo deprotonované a účastní se vzájemných **iontových interakcí**, stabilizující zejména vnitřní části proteinu, kde není přítomna voda. Přítomnost polárního rozpouštědla totiž zmenšuje energetickou výhodnost iontových vazeb. **Vodíkové vazby** jsou dalším typem elektrostatické interakce, které významně přispívají ke stabilitě sekundárních struktur. Vodíková vazba má výrazně směrový charakter, nejstabilnější úhel mezi účastnícími se dipóly je 180°. Závity α-helixu jsou drženy vodíkovými vazbami, které jsou orientovány ve směru osy helixu. β-skládaný list je stabilizován vodíkovými vazbami, které jsou kolmé k proteinovému řetězci. Vedle **Van der Vaalsových interakcí** pak protein stabilizuje ještě **hydrofobní efekt**. Lze jej popsat jako tendenci nepolárních molekul preferovat nepolární okolí, což má za následek zanoření nepolárních postranních řetězců dovnitř proteinu a následný vznik kompaktní globulární struktury. Pro molekuly vody je nevýhodné interagovat s nepolárním řetězcem, raději preferují okolní molekuly vody, s nimiž mohou vytvářet vodíkové vazby. Kolem nepolárního řetězce tvoří molekuly vody entropicky nevýhodné uspořádané struktury, které alespoň částečně kompenzují ztrátu vodíkových vazeb. Shluknutí nepolárních řetězců umožní části molekul vody opustit vysoce uspořádanou strukturu obklopující nepolární molekuly, čímž dojde ke zvýšení entropie systému a tedy i stabilizaci proteinu.

1.2 Termodynamika sbalování

Sbalování proteinu (anglicky *protein-folding*) jde popsat jako rovnovážná reakce mezi složeným (**F**olded) a nesloženým (**U**nfolded) proteinem, U $\rightleftharpoons$ F. Rovnovážnou konstantu této reakce lze vypočítat jako jako podíl aktivity produktu a reaktantu: $K_{\mathrm{eq}} = a(\mathrm{F})/a(\mathrm{U})$. Pro změnu Gibbsovy volné energie platí

$$\Delta G^{\circ} = -RT \ln K_{\mathrm{eq}}. \tag{1.1}$$

Aby reakce probíhala směrem k produktům, musí platit $\Delta G < 0$. Změna Gibbsovy energie závisí podle vztahu

$$\Delta G = \Delta H - T\Delta S \tag{1.2}$$

na změně entalpie a entropie. Změna entalpie souvisí se spotřebovaným nebo uvolněným teplem, entropie se změnou uspořádanosti systému. Aby platilo $\Delta G < 0$ a reakce mohla probíhat, musí být řízena buď entalpicky: $\Delta H < 0$, nebo entropicky: $\Delta S > 0$, případně musí entalpická i entropická složka přispívat k záporné hodnotě Gibbsovy energie [7].

Složené proteiny lze jednoduše denaturovat, ať už zvýšením teploty, nebo změnou pH či iontové síly rozpouštědla. Rovnováha skládání patrně není výrazně nakloněna ve prospěch produktu. Lze říci, zda-li je tento proces řízen entalpicky nebo entropicky?

Na první pohled by se mohlo zdát, že vznik jedné konkrétní složené konformace je entropicky nevýhodný, proto $\Delta S < 0 \Rightarrow \Delta H < 0$ a sbalování je tedy

entalpicky řízená exotermní reakce. Jak bylo zmíněno výše, skládání proteinu je ovšem ve skutečnosti série dílčích reakcí. Vznik sekundárních struktur v první fázi je řízen entalpicky z důvodu velkého množství vznikajících vodíkových vazeb. Jejich následné uspořádání do *roztavené globule* je hnáno hydrofobním efektem, tedy entropicky. Zvýšení entropie okolní vody při tomto kroku částečně kompenzuje entropicky nevýhodný přechod proteinu do jediné nativní konformace.

1.3 Rešerše současného stavu

Jedním z prvních pokusů o simulaci protein-foldingu představuje práce Levitta a Warshela [8] z roku 1975, v níž protein aproximují zhrubeným modelem, jehož jednotky jsou umístěny v polohách C_α jednotlivých aminokyselin. Na ně jsou vyjma glycinu vždy vázány částice, modelující postranní řetězce. Úhel mezi třemi po sobě jdoucími C_α byl považován za konstantní, jakož i délky vazeb. Nevazebné interakce byly modelovány Lennard-Jonesovým potenciálem. Jediným stupněm volnosti tedy byly rotace kolem C_α–C_α vazeb, simulace proteinu o 58 aminokyselinách již tehdy přinesla dobré výsledky.

Mřížkové modely jsou schopny drasticky snížit počet možných konformací proteinu a zrychlit tak výpočet. Průkopníkem v této oblasti je práce Dill, Bromberg [9] z roku 1995. Zkoumají vlastnosti HP modelu (aminokyseliny jsou rozlišeny pouze na **H**ydrofobní a **P**olární) na kubické mřížce. Přes svou jednoduchost jsou schopny mřížkové modely poskytnout netriviální informace o průběhu sbalování [10].

Současný výzkum přistupuje k protein-foldingu z různých stran. *Ab initio* modelování proteinů nevychází ze žádných statistických ani strukturních dat a potenciální energii zkoumaného systému definuje pouze pomocí standardních fyzikálních vztahů (Coulombův zákon, rigidní vazba, případně její aproximace harmonickým oscilátorem, Lennard-Jonesův nevazebný potenciál) [2].

Statistické potenciály se snaží jisté parametry nastavovat podle charakteristiky složených proteinů [2]. Schommers používá přístup zvaný *Boltzmannova iterativní inverze* [11], při němž koriguje výpočet potenciálu členem, který obsahuje podíl aktuální radiální distribuční funkce $\rho(r)$ a experimentálně stanovené radiální distribuční $\rho_{\mathrm{exp}}(r)$.

$$V_{i+1}(r) = V_i(r) - k_{\mathrm{B}}T \ln\left(\frac{\rho(r)}{\rho_{\mathrm{exp}}(r)}\right). \tag{1.3}$$

Aktuální distribuční funkce tak s postupem času konverguje ke správné distribuční funkci. Tanaka a Shega [12] zase formulují interakční energii mezi postranními řetězci dvou kuliček jako

$$E = -k_{\mathrm{B}}T \ln\left(\frac{N}{N_{\mathrm{exp}}}\right), \tag{1.4}$$

kde N je pozorovaný počet kontaktů mezi konkrétním typem postranních řetězců a N_{exp} je experimentálně stanovený počet kontaktů. Statistických potenciálů využívají rozšířené modely *Rosetta* a *CABS*.

Komparativní modelování vychází z pozorování, že se proteiny s podobnou primární strukturou skládají do podobných nativních konformací. Míra schody

proteinových sekvencí nemusí pro dosažení dobrého výsledku v některých případech přesahovat 30% [2]. Komparativním modelováním získáme přibližnou strukturu. Následně je možné porovnávat krátké sekvence (<10 aminokyselin), pro něž je přesnost stanovení správné prostorové konformace vysoká a tuto přibližnou strukturu poupravit.

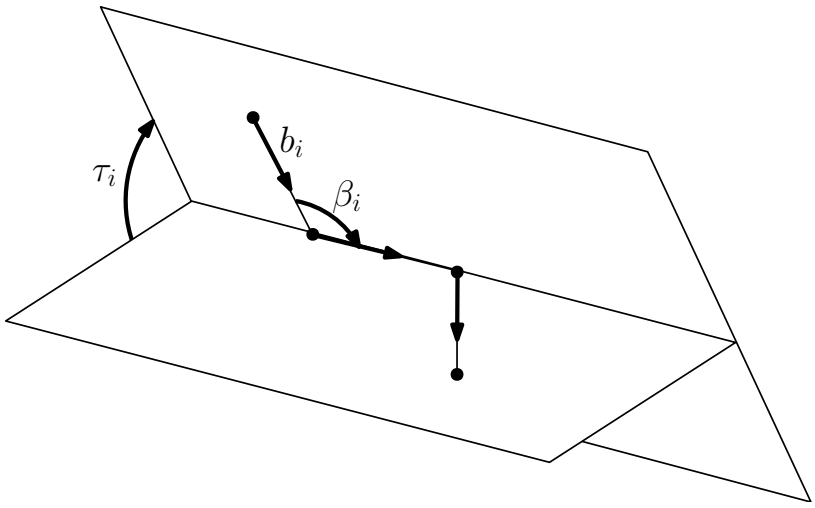
K porovnání přesnosti softwaru řešícího protein-folding se každé dva roky pořádá soutěž CASP (z angl. *critical assessment of methods for protein structure prediction*), jíž se účastní přední skupiny z celého světa. Několik měsíců před samotnou soutěží se od strukturních biologů shromáždí struktury proteinů, které již byly vyřešeny, ale ještě nebyly publikovány. Účastnící se skupiny provedou simulace se sekvencemi těchto proteinů a své výsledky nahrají na CASP server. Výsledky organizátoři vyhodnotí a seřadí podle jejich přesnosti. Přesnost simulací každoročně vzrůstá, prim drží simulace využívající komparativní modelování.

V roce 2018 vyhrál soutěž CASP program AlphaFold od společnosti DeepMind, spadající pod Google [13]. Princip fungování programu AlphaFold je značně složitý, ke zpracování velkého množství dat z DNA databází využívá strojového učení a použitý model zahrnuje 21 milionů parametrů. V roce 2020 představila tatáž společnost program AlphaFold2, dosahující přesnějších výsledků a schopný takřka bezchybně řešit strukturu monomerních proteinů.

2. Zhrubené modely

2.1 Zhrubené modely

Simulovat atomistický model proteinu je výpočetně velmi náročné a omezuje použití této metody pouze na malé systémy. Snížit výpočetní nároky a umožnit simulace větších systémů lze použitím zhrubeného modelu (angl. coarse-grained, CG). Každou aminokyselinu nahradíme kuličkou, jejíž souřadnice budeme uvažovat za polohu C_α. Model proteinu s N aminokyselinami lze definovat pomocí $N-1$ vazebných vektorů b_i, $N-2$ vazebných úhlů β_i a $N-3$ torzních úhlů τ_i.



Obrázek 2.1: Znázornění b_i, β_i, τ_i.

Vazebné interakce modelujeme pomocí harmonického potenciálu. Délka vazby osciluje kolem rovnovážné vzdálenosti r_{eq}.

$$U_{\text{bonding}}(r_{i,j}) = k(r_{i,j} - r_{\text{eq}})^2. \tag{2.1}$$

Lokální interakce hrají důležitou úlohu při stabilizaci nativní struktury [10], proto je nutné tyto interakce do modelu zahrnout. Lze zvolit různé přístupy, jak je modelovat. Analytické vztahy v práci Monasse a Boussinota [14] lze použít pro výpočet derivací potenciálů v cosinové formě

$$\begin{aligned} \kappa(\beta_i) &= \cos(\beta_i) \\ \lambda(\tau_i) &= \cos(\tau_i) \end{aligned}. \tag{2.2}$$

Při jejich odvození je potřeba dbát na to, aby byl součet sil působících na každou uvažovanou trojici (vazebný úhlový potenciál $\kappa(\beta_i)$) či čtveřici (torzní úhlový potenciál $\lambda(\tau_i)$) roven nule. V případě torzního potenciálu je ještě zapotřebí zachovat nulový moment síly všech uvažovaných čtveřic. S ohledem na výpočetní náročnost tohoto přístupu lze za předpokladu $r_{i,j} \approx 1$ aproximovat cosinus uvažovaného úhlu jako skalární součin odpovídajících vazebných vektorů.

$$\begin{aligned} \kappa(\beta_i) = \cos(\beta_i) &\approx (\vec{b} - \vec{a}) \cdot (\vec{c} - \vec{b}), \\ \lambda(\tau_i) = \cos(\tau_i) &\approx (\vec{b} - \vec{a}) \cdot (\vec{d} - \vec{c}). \end{aligned} \tag{2.3}$$

Tento způsob výpočtu potenciální energie je jednodušší naprogramovat a ve výsledku spotřebuje méně procesorového času.

Oba předchozí způsoby výpočtu potenciální energie nebraly v potaz skutečnost, že některé hodnoty β_i, τ_i se v proteinech vyskytují častěji z důvodu přítomnosti α-helixů a β-skládaných listů. Tuto skutečnost lze zohlednit zavedením parametru δ_i, jehož hodnota bude měnit polohu minima. Funkce lze zvolit následovně

$$\begin{aligned} \kappa(\beta_i) &\approx w_{\kappa,i}\cos(\delta_1) - (1 - w_{\kappa,i}^2)\sin(\delta_1), \\ \lambda(\tau_i) &\approx w_{\lambda,i}\cos(\delta_2) - (1 - w_{\lambda,i}^2)\sin(\delta_2), \end{aligned} \tag{2.4}$$

kde $w_{\kappa,i} = (\vec{b} - \vec{a}) \cdot (\vec{c} - \vec{b})$ a $w_{\lambda,i} = (\vec{b} - \vec{a}) \cdot (\vec{d} - \vec{c})$.

Vazebné a lokální interakce dvou následujících modelů budou v simulaci počítány způsobem uvedeným výše, lišit se budou pouze párovými nevazebnými interakcemi. ‘

2.1.1 AB model

Kuličky budeme dělit na hydrofobní (A) a hydrofilní (B). Podle článku [15] rozdělíme 20 základních aminokyselin následujícím způsobem:

$$\begin{aligned} A &= \{I, V, L, P, C, M, A, G\} \\ B &= \{D, E, F, H, K, N, Q, R, S, T, W, Y\} \end{aligned}$$

Párové **nevazebné interakce** kuliček budeme modelovat pomocí Lennardova–Jonesova (LJ) potenciálu upraveného tak, aby zohlednil větší energetickou výhodnost interakce dvou hydrofobních kuliček (AA), oproti interakcím hydrofilní – hydrofobní (AB) nebo hydrofilní – hydrofilní (BB):

$$E_{\mathrm{LJ}}(r_{i,j}, \xi_i, \xi_j) = 4\epsilon(\xi_i, \xi_j)\left[\left(\frac{\sigma}{r_{i,j}}\right)^{12} - \left(\frac{\sigma}{r_{i,j}}\right)^{6}\right], \tag{2.5}$$

$r_{i,j}$ je vzdálenost kuliček i, j; σ je parametr LJ potenciálu, mající význam vzdálenosti, ve které je potenciál energie dvou kuliček nulová. $\epsilon(\xi_i, \xi_j)$ je interakční energie, mající význam hloubky LJ potenciálu. Je definována následovně:

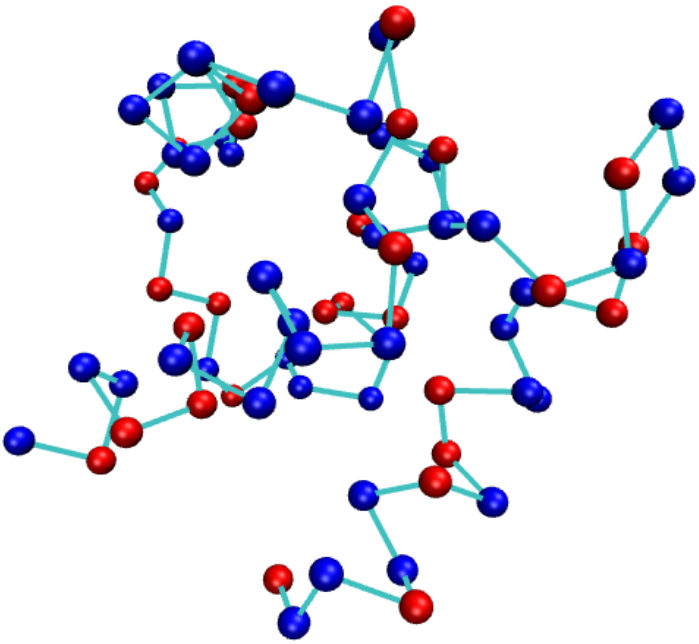
$$\epsilon(\xi_i, \xi_j) = \begin{cases} 1, & \text{pro páry AA,} \\ \frac{1}{2}, & \text{pro páry AB, BB.} \end{cases}$$

2.1.2 AZ model

Pro zvýšení přesnosti modelu přestaneme dělit kuličky na hydrofilní a hydrofobní, každá kulička nyní bude představovat konkretní aminokyselinu simulovaného proteinu. Interakční energie dvou kuliček $\epsilon(\xi_i, \xi_j)$ bude odpovídat energiím, které Miyazava a Jernigen stanovili experimentálně pro všechny dvojice aminokyselin a své výsledky publikovali v článku [16].

Časová složitost výpočtu LJ potenciálu přes všechny dvojice kuliček i,j je $\mathcal{O}(N^2)$. Je možno ji snížit zavedením cutoff vzdálenosti r_{cutoff}. Pak

$$E_{\mathrm{LJ}}(r_{i,j}, \xi_i, \xi_j) = \begin{cases} 0, & \text{pro } r_{i,j} > r_{\mathrm{cutoff}}, \\ 4\epsilon(\xi_i, \xi_j)\left[\left(\frac{\sigma}{r_{i,j}}\right)^{12} - \left(\frac{\sigma}{r_{i,j}}\right)^{6}\right] - E_{\mathrm{cutoff}}, & \text{jinak.} \end{cases}$$



Obrázek 2.2: AB model nativní struktury proteinu 1CLB.

Funkce pro výpočet energie by měla být spojitá a hladká, proto je nutné od takto ořízntého nespojitého LJ potenciálu odečíst $E_{\text{cutoff}} = E_{\text{LJ}}(r_{\text{cutoff}}, \sigma_i, \sigma_j)$, čímž získáme funkci sice opět nehladkou, přesto však mnohem vhodnější pro počítačovou implementaci. Složitost výpočtu takto ořízntého potenciálu lze snížit na $\mathcal{O}(N)$.

Zahrnutím výše zmíněných potenciálů můžeme celkovou potenciální energii modelu definovat následovně:

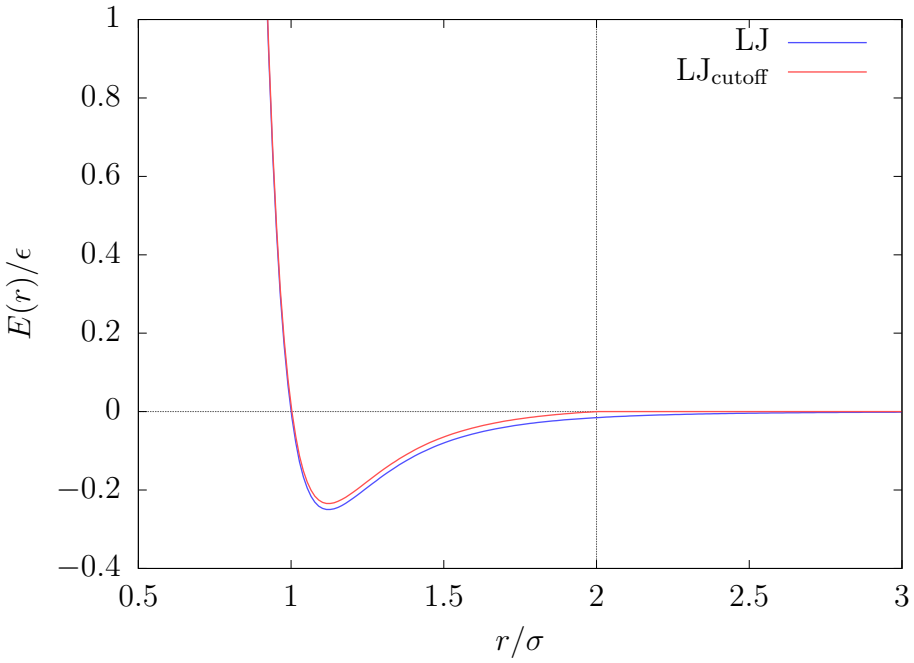
$$\begin{aligned} V(x) = &- k_1 \sum_{i=1}^{N-2} \kappa(\beta_i) - k_2 \sum_{i=1}^{N-3} \lambda(\tau_i) + k_3 \sum_{i=1}^{N-1} (x_{i,i+1} - x_{\text{eq}})^2 \\ &+ \sum_{i=1}^{N-2} \sum_{j=i+2}^{N} E_{\text{LJ}}(r_{i,j}, \xi_i, \xi_j), \end{aligned} \tag{2.6}$$

kinetickou energii definujeme jako

$$K(v) = \sum_{i=1}^{N} \frac{1}{2} m v_i^2. \tag{2.7}$$

2.2 Redukované jednotky

Popis simulovaného systému pomocí jednotek SI je nepraktický. Abychom se vyhnuli práci s malými čísly, můžeme si jednotky v simulaci definovat sami. Z E_{LJ} můžeme jednotku energie definovat jako ϵ a jednotku délky jako σ. Jednotka hmotnosti bude rovna hmotnosti kuličky m. Pomocí ϵ, σ, m jsme schopni vyjádřit ostatní veličiny. Například jednotku času jako $[t] = \sigma\sqrt{m/\epsilon}$, rychlosti jako $[v] = \sqrt{\epsilon/m}$.



Obrázek 2.3: Oříznutý LJ potenciál, $r_{\text{cutoff}} = 2$.

2.3 Studované veličiny

Podobnost dvou konfigurací a, b popisuje veličina RMSD (z angl. root mean square distance), která je definována jako

$$\text{RMSD} = \min \sqrt{\frac{1}{N} \sum_{i=1}^{N} |\vec{R}_i^{(b)} - \vec{R}_i^{(a)}|^2}. \tag{2.8}$$

$\vec{R}_i^{(x)}$ označuje i-tou částici konfigurace x. Vzdálenost $|\vec{R}_i^{(b)} - \vec{R}_i^{(a)}|$ mezi částicemi $\vec{R}_i^{(a)}$ a $\vec{R}_i^{(b)}$ je brána jako minimum přes všechny translace a rotace modelů. Vedle nalezení konfigurace co nejvíce podobné nativnímu proteinu budeme v průběhu skládání studovat následující veličiny:

Gyrační poloměr polymeru je veličina popisující velikost polymerního klubka, která je dostupná experimentálně. Lze ji určit jako

$$R_{\text{gyr}}^2 = \frac{1}{N} \sum_{i=1}^{N} |\vec{R}_i - \vec{R}_{\text{cm}}|^2, \tag{2.9}$$

kde $\vec{R}_{\text{cm}}$ je souřadnice těžiště modelu.

Tepelnou kapacitu lze podle [10] určit jako funkci variance energie následovně:

$$C_{\text{V}}(T) = \frac{1}{T^2}(\langle E^2 \rangle - \langle E \rangle^2). \tag{2.10}$$

Stupeň složenosti Q lze definovat jako [10]

$$Q = \frac{n_0}{n}, \tag{2.11}$$

kde n_0 je počet okupovaných nativních kontaktů a n je celkový počet nativních kontaktů, který lze získat ze strukturních dat konkrétního proteinu. Řekneme, že

kuličky i,j jsou v kontaktu, pokud $r_{ij}^2 < 1.75$. Pro nativní strukturu platí $Q = 1$. Vedle Q definujme ještě lokální stupeň složenosti Q_l a globální stupeň složenosti Q_g:

$$Q_l = \frac{n_{0,l}}{n_l}, \tag{2.12}$$

$$Q_g = \frac{n_{0,g}}{n_g}, \tag{2.13}$$

kde $n_{0,l}$ a $n_{0,g}$ jsou lokální a globální počty okupovaných nativních kontaktů. n_l a n_g jsou celkové počty lokálních a globálních nativních kontaktů. Kuličky i,j jsou v lokálním kontaktu, pokud platí $2 \leq |i - j| \leq 4$ a v globálním kontaktu,pokud platí $|i - j| > 4$. V obou případech ještě musí platit $r_{ij}^2 < 1.75$.

3. Teorie

3.1 Termodynamický popis systému

Jako mikrostav označujeme okamžitý stav systému. Například pro mechanický popis systému N částic může jít o vektor poloh r_i a hybností p_i všech částic $(r_1, \ldots, r_n, p_1, \ldots, p_n)$ v čase t. Vývoj systému ve fázovém prostoru nazýváme trajektorie. Makrostav je pak průměrem velkého množství mikrostavů, tedy to co pozorujeme.

Pokud známe pravděpodobnost, se kterou každý mikrostav nastane, daný soubor označujeme jako statistický.

3.1.1 Mikrokanonický soubor

Mějme izolovaný systém, nevyměňující s okolím teplo, práci ani částice. Energie systému je v čase konstantní, může však být realizována obrovským množstvím mikrostavů. Pokud máme systém o energii E ve stavu ϕ, platí pro pravděpodobnost výskytu stavu $\pi(\phi)$ vztah

$$\pi(\phi) = \frac{1}{W}, \tag{3.1}$$

kde W je počet všech stavů ϕ_i, jejichž energie $E(\phi_i) = E$. Stavy o stejné energii jsou tedy stejně pravděpodobné. Pro konečný počet stavů odpovídá makroskopická (měřená) hodnota veličiny X aritmetickému průměru veličiny přes všechny stavy,

$$\langle X \rangle = \frac{\sum_\phi X(\phi)}{W}, \tag{3.2}$$

kde $\langle X \rangle$ je střední hodnota a $X(\phi)$ je hodnota veličiny X ve stavu ϕ.

3.1.2 Kanonický soubor

Systém, který s okolím může vyměňovat teplo, umístěme do tepelné lázně. Jeho energie přestane být konstantní a po dosažení tepelné rovnováhy bude jeho energie fluktuovat kolem jisté střední hodnoty. Pro takovýto systém již nelze předpokládat, že pravděpodobnost výskytu všech jeho stavů bude stejná. Jelikož všechny stavy se stejnou energií jsou stejně pravděpodobné, je pravděpodobnost výskytu stavu $\pi(\phi)$ pouze funkcí jeho energie $\pi(E(\phi))$ a je úměrná Boltzmannovu faktoru

$$\pi(\phi) \propto e^{-E(\phi)/k_B T}, \tag{3.3}$$

kde k_B je Boltzmannova konstanta a T je termodynamická teplota systému. Na rozdíl od mikrokanonického souboru, kde byly všechny stavy stejně pravděpodobné, je v kanonickém souboru při určovaní střední hodnoty veličiny X zapotřebí zahrnout do výpočtu také pravděpodobnost výskytu každého stavu. Pro spočetný počet stavů:

$$\langle X \rangle = \sum_\phi X(\phi)\pi(\phi) = \frac{\sum_\phi X(\phi) e^{-E(\phi)/k_B T}}{\sum_\phi e^{-E(\phi)/k_B T}}, \tag{3.4}$$

pro spojitý systém přejde rovnice 3.4 do integrovaného tvaru

$$\langle X \rangle = \int_\phi X(\phi)\pi(\phi)\ \mathrm{d}\phi = \frac{\int_\phi X(\phi)e^{-E(\phi)/k_BT}\ \mathrm{d}\phi}{\int_\phi e^{-E(\phi)/k_BT}\ \mathrm{d}\phi}. \tag{3.5}$$

3.2 Molekulová dynamika

Úkolem molekulové dynamiky je pro daný systém určit jeho vývoj v čase (trajektorii). Podle druhého Newtonova zákona je síla $\vec{F}$ působící na částici o hmotnosti m rovna

$$\vec{F} = m\frac{\mathrm{d}^2\vec{r}(t)}{\mathrm{d}t^2}, \tag{3.6}$$

což je diferenciální rovnice druhého řádu. Pokud obecnou diferenciální rovnici nechceme nebo neumíme vyřešit analyticky, můžeme dojít k přibližnému výsledku numericky, použitím integrátoru (algoritmus, implementující konkrétní numerickou metodu). Než bude představen Verletův integrátor, tak pro lepší pochopení ukážu integrátor mnohem primitivnější.

V roce 1768 publikoval švýcarský matematik a fyzik Leonard Euler dílo *Institutionum calculi integralis*, ve kterém je popsán jeden z nejjednodušších způsobů numerické integrace, **Eulerova metoda**. Mějme obecnou diferenciální rovnici prvního řádu,

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x,y), \quad y(x_0) = y_0. \tag{3.7}$$

Pro rozumně malý krok h lze funkci $y(x)$ v bodě $x + h$ aproximovat jako

$$y(x + h) \approx y(x) + hy'(x). \tag{3.8}$$

V bodě $x_1 = x_0 + h$ proto aproximujeme funkci jako $y(x_1) \approx y_0 + hf(x_0, y_0)$, v bodě x_2 jako $y(x_2) \approx y_1 + hf(x_1, y_1)$... Takto lze s postupně rostoucí chybou určit přibližnou hodnotu $y(x)$.

3.2.1 Velocity Verlet

Jedním ze způsobů numerické integrace Newtonových pohybových rovnic, je použití Verletova integrátoru. Ze stávajících informací o systému dokáže spočítat budoucí pozice všech jeho částic. Verletův algoritmus patří mezi symplektické integrátory, což znamená, že je časově reverzibilní a zachovává fázový objem [17]. Pro rozumně malý časový krok Δt jsme schopni nalézt vztah mezi současnou polohou částice $r(t)$ a budoucí $r(t+\Delta t)$. Taylorovým rozvojem $r(t+\Delta t)$ získáme

$$r(t + \Delta t) = r(t) + \Delta t\frac{\mathrm{d}r(t)}{\mathrm{d}t} + \frac{\Delta t^2}{2!}\frac{\mathrm{d}^2r(t)}{\mathrm{d}t^2} + \frac{\Delta t^3}{3!}\frac{\mathrm{d}^3r(t)}{\mathrm{d}t^3} + \mathcal{O}(\Delta t^4), \tag{3.9}$$

kde $\mathcal{O}(\Delta t^4)$ vyjadřuje chybu vzniklou zanedbáním členů obsahujících čtvrtou a vyšší derivaci $r(t)$. Podobně Taylorovým rozvojem $r(t - \Delta t)$ a zanedbáním vyšších derivací získáme:

$$r(t - \Delta t) = r(t) - \Delta t\frac{\mathrm{d}r(t)}{\mathrm{d}t} + \frac{\Delta t^2}{2!}\frac{\mathrm{d}^2r(t)}{\mathrm{d}t^2} - \frac{\Delta t^3}{3!}\frac{\mathrm{d}^3r(t)}{\mathrm{d}t^3} + \mathcal{O}(\Delta t^4). \tag{3.10}$$

Sečtením rovnic 3.9 a 3.10 a následnou úpravou získáme:

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + \Delta t^2 \frac{\mathrm{d}^2 r(t)}{\mathrm{d}t^2} + \mathcal{O}(\Delta t^4). \tag{3.11}$$

Pro výpočet polohy částice v čase $t + \Delta t$ však stále potřebujeme znát polohu částice v čase $t-\Delta t$. Zkusíme se tomu vyhnout tak, že odečteme rovnice $3.9 - 3.10$ a úpravou získáme následující vztah pro rychlost

$$\frac{\mathrm{d}r(t)}{\mathrm{d}t} = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2), \tag{3.12}$$

který ještě dále upravíme

$$r(t-\Delta t) = r(t+\Delta t) - 2\Delta t \frac{\mathrm{d}r(t)}{\mathrm{d}t} + \mathcal{O}(\Delta t^2) \tag{3.13}$$

a dosazením do rovnice 3.11 získáme známý vztah

$$r(t+\Delta t) = r(t) + \frac{\mathrm{d}r(t)}{\mathrm{d}t}\Delta t + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2 r(t)}{\mathrm{d}t^2}, \tag{3.14}$$

který jde pomocí okamžité rychlosti $v(t)$ a okamžitého zrychlení $a(t)$ vyjádřit jako

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2. \tag{3.15}$$

Dalším parametrem algoritmu Velocity-Verlet jsou rychlosti částic po vykonání malého časového kroku, $v(t + \Delta t)$. Pro ně lze z rovnice 3.9 odvodit vztah

$$v(t+\Delta t) = v(t) + \frac{a(t) + a(t+\Delta t)}{2}\Delta t. \tag{3.16}$$

3.3 Monte Carlo metody

Monte Carlo metody (MC) používají generátory náhodných čísel ke generování konfigurací zkoumaného systému. Na rozdíl od *statických MC metod*, které generují posloupnost statisticky nezávislých konfigurací z rozdělení pravděpodobností P, *dynamické MC metody* poskytují posloupnost (Markovův řetězec) korelovaných konfigurací s požadovaným rozdělením P. S pomocí MC metod jsme tedy schopni generovat mnoho konfigurací zkoumaného systému a díky nim pak stanovit různé termodynamické veličiny, popisující tento systém.

Monte Carlo integrace [18] slouží k aproximaci hodnoty určitých integrálů

$$F = \int_a^b f(x)\ \mathrm{d}x. \tag{3.17}$$

Hodnotu tohoto integrálu lze interpretovat jako plochu pod grafem funkce $f(x)$ mezi body a, b. Výraz $F_i = f(x_i)(b-a)$, kde x_i je náhodné číslo z intervalu (a,b), představuje obsah obdélníku o stranách $|ab|$ a $f(x_i)$. Obsah tohoto obdélníku nám nic neříká o hodnotě určitého integrálu a nevíme, jak blízko jí je. Pokud ovšem výraz vyhodnotíme ve velkém množství náhodných bodů a takto získané obsahy obdélníků zprůměrujeme, lze ukázat, že získaná hodnota se blíží k F:

$$\langle F_N \rangle = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i), \tag{3.18}$$

$$\lim_{N\to\infty} \langle F_N \rangle = F, \tag{3.19}$$

kde N je počet použitých náhodných čísel $x_i \in (a,b)$. Tato čísla můžeme získat vyhodnocením výrazu

$$x_i = a + u_{(0,1)} \cdot (b - a), \tag{3.20}$$

kde $u_{(0,1)}$ je náhodné číslo z intervalu (0,1), které lze jednoduše vygenerovat ve kterémkoliv programovacím jazyce. Analogicky lze pomocí Monte Carlo integrace vyhodnocovat N-dimenzionální určité integrály:

$$\int_Q f(x_1, x_2, \cdots, x_N)\, \mathrm{d}x_1 \mathrm{d}x_2 \cdots \mathrm{d}x_N \cong \frac{|Q|}{N} \sum_{i=1}^{N} f(x_1^{(i)}, x_2^{(i)}, \cdots, x_N^{(i)}), \tag{3.21}$$

kde $(x_1^{(i)}, x_2^{(i)}, \cdots, x_N^{(i)})$ je i-tý náhodný bod z oblasti Q.

Zdálo by se, že pomocí náhodných čísel můžeme snadno generovat konfigurace systému N bodů a dosazením do rovnice (3.4) určovat střední hodnoty veličin. Tento přístup bývá někdy označován jako naivní Monte Carlo, je ovšem neefektivní, protože náhodným vzorkováním získáváme spoustu konfigurací s malou pravděpodobností výskytu, které k hodnotě $\langle X \rangle$ téměř nepřispívají.

3.3.1 Importance sampling

Problém generování velkého množství konfigurací s malou pravděpodobností výskytu lze vyřešit tak, že pro výpočet středních hodnot budeme používat pouze ty konfigurace, které významným způsobem přispívají k hodnotě integrálu (3.4). Tímto způsobem pracuje **Metropolisův algoritmus** [19], který generuje pouze ty konfigurace, které podstatně přispívají k hodnotě $\langle X \rangle$. Algoritmus generuje novou konfiguraci systému ze stávající následovně:

1. Vybereme náhodnou částici a náhodně s ní pohneme.
2. Spočítáme změnu potenciální energie ΔU.
3. Pokud $\Delta U \leq 0$, tuto změnu konfigurace přijmeme, jinak ji přijmeme s pravděpodobností $\exp(-\Delta U / k_B T)$. V opačném případě novou konfiguraci zamítneme a vrátíme se zpět k původní konfiguraci.

Trajektorii (časový vývoj) systému lze formálně popsat pomocí náhodné veličiny a Markovových řetězců. Máme náhodnou veličinu $\mathcal{J}$, která nabývá jisté hodnoty z konečné množiny $\mathcal{M}$ (podobně, jako máme jednu konkrétní konfiguraci systému z mnoha možných). Každé této hodnoty nabývá s pravděpodobností $\pi(\mathcal{M}_i) = \pi_i$. Markovův řetězec je pak posloupnost náhodné veličiny $\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_n$. Pokud $\mathcal{J}_k$ nabývá hodnoty $\mathcal{M}_i$ s pravděpodobností $\pi_i^{(k)}$, pak je pravděpodobnost, že $\mathcal{J}_{k+1}$ bude nabývat hodnoty $\mathcal{M}_j$ určena vztahem

$$\pi_j^{(k+1)} = \sum_{i=1}^{n} \pi_i^{(k)} W_{i\to j}. \tag{3.22}$$

$\mathcal{J}_{k+1}$ tedy závisí na $\mathcal{J}_k$. Totéž lze vektorově zapsat jako

$$\pi^{(k+1)} = \pi^{(k)} \cdot \boldsymbol{W}, \tag{3.23}$$

kde $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ je vektor pravděpodobností jednotlivých stavů a $\boldsymbol{W}$ je matice přechodu. Prvky matice $W_{i\to j}$ mají význam pravděpodobnosti přechodu ze stavu i do stavu j. Je-li Markovův řetězec dostatečně dlouhý, pak $\pi_i^{(k)}$ nezávisí na výchozím stavu systému a bude nabývat své limitní hodnoty. Obecně platí, že pokud jsou všechny stavy dosažitelné z libovolného stavu s nenulovou pravděpodobností a žádný stav není periodický, pak je daná množina konfigurací ergodická a pro libovolné počáteční rozdělení pravděpodobností existuje limitní rozdělení pravděpodobností $\boldsymbol{\pi}$:

$$\boldsymbol{\pi} = \lim_{k\to\infty} \pi^{(k)}. \tag{3.24}$$

Střední hodnota náhodné veličiny X proto s rostoucí délkou řetězce konverguje k souborové střední hodnotě dané vztahem 3.4.

Snažíme se tedy sestrojit Markovův řetězec konfigurací, kde pravděpodobnosti výskytu jednotlivých konfigurací odpovídají Boltzmanově váze a představují tak limitní rozložení neznámé matice přechodu $\boldsymbol{W}$. Ta musí splňovat následující podmínky [19]:

$$W_{i\to j} \geq 0, \tag{3.25}$$

$$\sum_{i,j}^{M} W_{i\to j} = 1. \tag{3.26}$$

$$\boldsymbol{\pi} \cdot \boldsymbol{W} = \boldsymbol{\pi}. \tag{3.27}$$

Poslední rovnice je splněna, pokud platí podmínka mikroskopické reverzibility

$$\pi_i W_{i\to j} = \pi_j W_{j\to i}, \tag{3.28}$$

případně pokud jsou oba prvky matice přechodu, $W_{i\to j}$ a $W_{j\to i}$, rovny nule.

V padesátých letech 20. století byla navržena následující matice přechodu [20], pro jejíž prvky platí

$$W_{i\to j} = \alpha_{i\to j} \min\left\{1, \frac{\pi_j}{\pi_i}\right\} \quad \text{pro } i \neq j, \tag{3.29}$$

kde $\alpha_{i\to j}$ je libovolná symetrická stochastická matice, která splňuje podmínku mikroskopické reverzibility (3.28) a odpovídá pravděpodobnosti generování konfigurace A_j z konfigurace A_i. Tato matice je symetrická, proto musí být pravděpodobnost, že ze stavu A_i vygenerujeme stav A_j, stejná, jako pravděpodobnost provedení opačné změny, tedy generování stavu A_i ze stavu A_j. Matici odpovídají kroky Metropolisova algoritmu.

3.3.2 Hamiltonian Monte Carlo

Pro zkoumání velkých systémů, kterými jsou například zhrubené modely proteinů, je použití pouhé MD nebo MC neefektivní, protože dochází ke generovaní příliš korelovaných konfigurací [21]. Hamiltonian Monte Carlo (HMC) je schopné generovat konfigurace, které jsou mnohem více nezávislé, umožňuje volbu delšího časového kroku a snadněji překonává vysoké energetické bariéry. Hamiltonovská dynamika popisuje chování objektů pomocí jeho polohy r a hybnosti p. Celkovou

energii systému popisuje Hamiltonián $H(r,p)$, jež je roven součtu kinetické $K(p)$ a potenciální $V(r)$ energie systému [22].

$$H(r,p) = V(r) + K(p). \tag{3.30}$$

Hamiltonovy rovnice jsou derivacemi Hamiltoniánu podle polohy a hybnosti.

$$\begin{aligned} \frac{\mathrm{d}r_i}{\mathrm{d}t} &= \frac{\partial H}{\partial p_i} = \frac{\partial K(p)}{\partial p_i} \\ \frac{\mathrm{d}p_i}{\mathrm{d}t} &= -\frac{\partial H}{\partial r_i} = -\frac{\partial V(r)}{\partial r_i}. \end{aligned} \tag{3.31}$$

Jde o diferenciální rovnice prvního řádu. Toto soustavu rovnic musíme ještě doplnit počátečními podmínkami – (r_0, p_0) v čase t_0. Vidíme, že druhá z rovnic (3.31) je nápadně podobná druhému Newtonovu pohybovému zákonu, $F = m\frac{\mathrm{d}^2 r(t)}{\mathrm{d}t^2} = -\nabla V(r(t))$, pro $K(p) = \frac{1}{2}mv^2$ je splněna také první rovnice. Skutečně, Newtonova dynamika je speciálním případem Hamiltonovy dynamiky. Pokud dynamiku simulace formulujeme takto, poskytne nám to značné výhody. Hamiltonovská dynamika je symplektická, zachovává objem fázového prostoru a je časově reverzibilní. Zobrazení ze stavu systému v čase t, $(r(t),\ p(t))$ do stavu v čase $t+h$, $(r(t+h),\ p(t+h))$ má proto inverzní zobrazení, které získáme vynásobením rovnic (3.31) faktorem -1. Hodnota Hamiltoniánu se v čase nemění, celková energie systému je proto konstantní:

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \sum_{i=1}^{n} \left(\frac{\mathrm{d}r_i}{\mathrm{d}t}\frac{\partial H}{\partial r_i} + \frac{\mathrm{d}p_i}{\mathrm{d}t}\frac{\partial H}{\partial p_i} \right) = \sum_{i=1}^{n} \left(\frac{\partial H}{\partial p_i}\frac{\partial H}{\partial r_i} - \frac{\partial H}{\partial r_i}\frac{\partial H}{\partial p_i} \right) = 0. \tag{3.32}$$

HMC funguje následovně:

1. Generujeme počáteční konfiguraci (r_0, p_0) a stanovíme číslo aktuálního kroku $n = 0$ a celkový počet kroků n_f.
2. Dokud neplatí $n = n_f$, opakovaně provádíme následující kroky:
3. Generujeme nové hybnosti z Maxwellova-Boltzmannova rozdělení.
4. Systém vyvíjíme v čase pomocí MD, zkušební konfiguraci přijmeme nebo zamítneme na základě Metropolisova kritéria.
5. $n := n + 1$

HMC vzorkuje konfigurace tak, aby jejich rozdělení odpovídalo Boltzmanově váze. MD zachovává $H(r,p)$, v každém HMC cyklu je nutné generovat nové hybnosti z Maxwellova-Boltzmannova rozdělení.

3.3.3 Simulované žíhání

Máme-li funkci, jejíž hodnotu chceme minimalizovat (například potenciální energie coarse grained modelu jako funkce jeho souřadnic r), lze lokálního minima snadno dosáhnout pomocí algoritmu **Steepest descent** (česky algoritmus nejstrmějšího sestupu):

1. Generujeme počáteční konfiguraci systému K_0.
2. Spočteme potenciální energii systému $V(r)$ a její gradient $\nabla V(r)$
3. Nové souřadnice všech bodů získáme jejich posunutím o vzdálenost δ ve směru nejstrmějšího poklesu energie, $-\nabla V(r_i)$, tedy $r_i = r_{i-1} - \delta \nabla V(r_{i-1})$.
4. Kroky 2 – 3 opakujeme, dokud nebude platit $V(K_0) \approx V(K_i)$.

Tato metoda je jednoduchá na implementaci, je však schopná najít pouze nejbližší lokální minimum MIN_0 pro danou počáteční konfiguraci K_0, jelikož se vždy pohybujeme směrem největšího poklesu energie. Má-li funkce velké množství lokálních minim, nejsme schopni najít minimum globální, pokud se s K_0 netrefíme dostatečně blízko. Pro tento typ funkcí je vhodnější metoda **simulovaného žíhaní** [23] (simmulated annealing), která na rozdíl od nejstrmějšího sestupu generuje i konfigurace s vyšší energií a je tak schopna uniknout z dosahu MIN_0 a najít minimum výhodnější. Funguje následovně:

1. Generujeme počáteční konfiguraci systému K_0.
2. Nové souřadnice všech bodů získáme pomocí Monte Carlo metody. Novou konfiguraci přijmeme s pravděpodobností $\exp(-\Delta U/k_B T)$.
3. Snížíme teplotu systému: $T_i = \alpha T_{i-1}$, kde $\alpha \in (0{,}1)$ je chladící konstanta.
4. Kroky 2–3 opakujeme, dokud nedosáhneme předem stanovené finální teploty.

Postupným snižováním teploty se zmenšuje pravděpodobnost přijetí konfigurací s vyšší energií v kroku 2, a tedy i pravděpodobnost nalezení výhodnějších minim. Za dostatečně nízké teploty pak simulované žíhání postupně přechází v nejstrmější sestup.

3.4 Měření veličin v simulaci

Stanovovat hodnoty veličin výpočtem sum a integrálů, uvedených v oddílu 3.1 je velmi náročné, proto střední hodnotu veličiny $\langle X \rangle$ často aproximujeme aritmetickým průměrem naměřených hodnot X_i jednotlivých konfigurací systému, získaných během trajektorie o N krocích:

$$\langle X \rangle \cong \overline{X} = \frac{1}{N} \sum_{i=1}^{N} X_i. \tag{3.33}$$

Ze zákona velkých čísel vyplývá, že aritmetický průměr aproximuje střední hodnotu dobře, je-li počet měření N dostatečně velký a pokud je systém po dobu měření v rovnováze. Toho lze dosáhnout tak, že ze začátku ponecháme systému dostatek času pro relaxaci a teprve poté začneme měřit.

Podle centrální limitní věty se rozdělení naměřených hodnot X_i blíží jistému rozdělení, jehož rozptyl (varianci) můžeme použít k určení chyby měření. Rozptyl je definován jako

$$\sigma^2(X) = \langle X^2 \rangle - \langle X \rangle^2 = \frac{1}{N} \sum_{i=1}^{N} (X_i - \overline{X})^2. \tag{3.34}$$

Každá v simulaci vypočtená veličina je zatížená chybou. Při jejím výpočtu je zapotřebí vzít v úvahu, že data naměřená s použitím dynamických Monte Carlo metod jsou korelovaná a rovnice 3.34 pro ně neplatí. Podle Flyvbjerga a Petersena [24] lze rozptyl korelovaného souboru definovat jako

$$\sigma^2(X) = \frac{1}{N^2}\sum_{i,j=1}^{N}\gamma_{i,j} = \frac{1}{N}\left[\gamma_0 + \sum_{t=1}^{N-1}\left(1-\frac{t}{N}\right)\gamma_t\right], \tag{3.35}$$

kde $\gamma_{i,j}$ je autokorelační funkce, definovaná jako

$$\gamma_{i,j} = \langle X_i X_j\rangle - \langle X_i\rangle\langle X_j\rangle. \tag{3.36}$$

Autokorelační funkce je časově nezávislá, lze proto definovat

$$\gamma_t = \gamma_{i,j}, \text{ kde } t = |i-j|. \tag{3.37}$$

Definujme

$$c_t = \frac{1}{N-t}\sum_{k=1}^{N-t}(X_k - \overline{X})(X_{k+t} - \overline{X}). \tag{3.38}$$

Rozptyl můžeme aproximovat jako

$$\sigma^2(X) \approx \left\langle \frac{c_0 + 2\sum_{t=1}^{T} c_t}{N-2T-1}\right\rangle, \tag{3.39}$$

kde T je délka časového kroku, kterou je potřeba volit tak, aby byla výrazně větší, než exponenciální autokorelační čas τ, definovaný jako

$$\tau = \lim_{t\to\infty}\frac{t}{-\log\gamma_t}, \tag{3.40}$$

mající význam doby, za kterou poklesne míra korelace na hodnotu $1/e$ [25].

3.4.1 Bloková metoda

Jedním ze způsobů, jak se korelace zbavit, bez nutnosti náročného výpočtu korelačního času zmíněného výše, je blokový metoda [24]. Na posloupnost naměřených hodnot $(X_1, X_2, \ldots, X_N)$ aplikujeme blokovací operaci a získáme posloupnost poloviční délky $(X'_1, X'_2, \ldots, X'_{N'})$, kde nová délka posloupnosti $N' = \frac{1}{2}N$ a

$$X'_i = \frac{1}{2}(X_{2i} + X_{2i+1}). \tag{3.41}$$

Definujme $\gamma'_{i,j}$ a γ'_t pro zblokované hodnoty podobně, jako jsme definovali $\gamma_{i,j}$ a γ_t pro původní data. Pak

$$\sigma^2(X') = \frac{1}{N'^2}\sum_{i,j=1}^{N'}\gamma'_{i,j} = \sigma^2(X), \tag{3.42}$$

tato operace tedy nemění rozptyl $\sigma^2(X)$ a lze ukázat, že nemění ani střední hodnotu X. Z rovnice (3.37) vyplývá

$$\sigma^2(X) \geq \frac{\gamma_0}{N}. \tag{3.43}$$

Zároveň platí

$$\sigma^2(X) \geq \left\langle \frac{c_0}{N-1} \right\rangle. \tag{3.44}$$

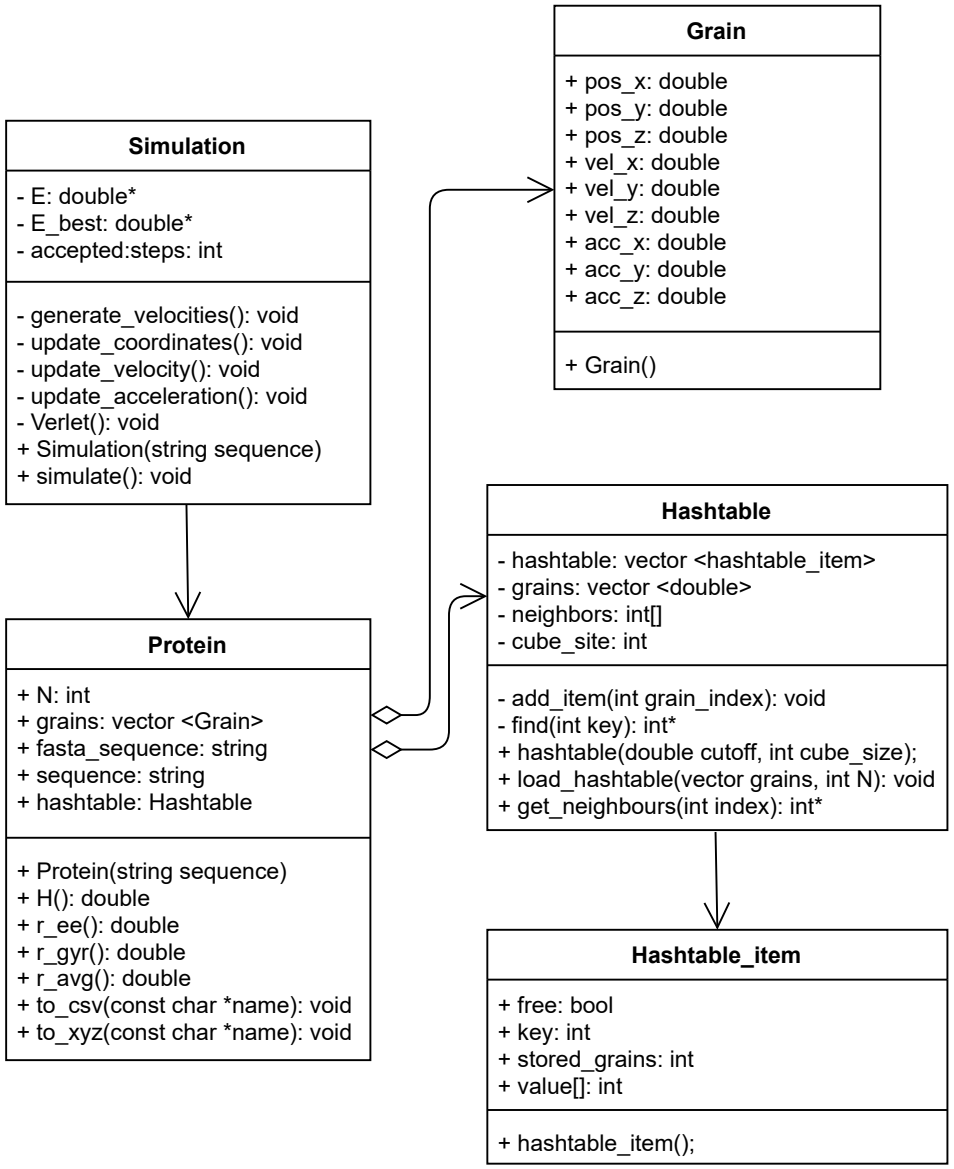
Pro $(X_1, X_2, \ldots, X_N)$ spočteme $c_0/(N-1)$, sloužící jako odhad $\langle c_0/(N-1)\rangle$ a provedeme blokovací operaci, po které vzrůstá hodnota γ_0/N. Poté na datech opakovaně provádíme blokovací operaci a počítáme $c'_0/(N'-1)$, sloužící jako odhad $\langle c'_0/(N'-1)\rangle$. Po čase se hodnota $c'_0/(N'-1)$ přestane měnit. Lze ukázat, že tato stacionární hodnota je dobrou aproximací $\sigma^2(X)$. Hodnoty získané blokováním jsou již nekorelované a jejich rozptyl můžeme vypočítat následovně:

$$\sigma^2(\overline{X}) \approx \frac{c'_0}{N'-1} \pm \sqrt{\frac{2}{N'-1}} \frac{c'_0}{N'-1}. \tag{3.45}$$

4. Počítačová simulace

Simulační program byl napsán v jazyce C++. Pro uložení souřadnic, okamžité rychlosti a zrychlení jednotlivých kuliček byly použity objekty třídy `Grain`. Model proteinu byl realizován pomocí třídy `Protein`, jejímž atributem je kontejner, obsahující objekty typu `Grain`. Metody třídy `Protein` umožňují výpočet celkové energie pomocí rovnic (2.6) a (2.7), výpočet veličin zmíněných v sekci (2.3), převod souřadnic kuliček do celkem $2N-5$ vazebných a torzních úhlů a v neposlední řadě také zápis simulačních dat do souborů.

Třída `Simulation` je potomkem třídy `Protein`. Její metody implementují integrátor Velocity-Verlet, Metropolisovo kritérium a zajišťují derivaci funkcí uvedených v sekci (2.1). V metodě `Simulation::Simulate` lze nastavit typ požadované simulace (MD, HMC nebo SA).



Obrázek 4.1: Zjednodušený UML diagram programu

Vstupním parametrem metody je `string sequence`. Jde o řetězec jednopísmenných zkratek aminokyselin, který udává složení proteinu. Řetězec je možno stáhnout z PDB[1] jako soubor formátu FASTA. Řetězec je zapotřebí ke správnému určení interakčních energií $\epsilon(\xi_i, \xi_j)$.

Strukturní data získaná pomocí NMR nebo rentgenové difrakce lze z PDB stáhnout ve stejnojmenném formátu `PDB`. Jedná se o textový soubor velmi složité struktury. Hlavičková sekce obsahuje proměnlivé spektrum popisných sekcí, téměř vždy obsahuje části `HEADER`, `TITLE`, `AUTHOR`, `EXPDTA`, v nichž uvádí informace o měřeném proteinu, autorovi experimentu a použité metodě. Tělo souboru obsahuje souřadnice všech atomů stanovované struktury, jakož i souřadnice heteroatomů. Nebývá výjimkou, že jeden `PDB` soubor obsahuje více naměřených struktur. Obtížná předvídatelnost obsahu jednotlivých souborů znesnadňuje práci s nimi. Pro převod `PDB` souboru na jiný formát lze použít program Babel[2].

```
HEADER    CALCIUM-BINDING PROTEIN                  08-FEB-95   1CLB
TITLE     DETERMINATION OF THE SOLUTION STRUCTURE OF APO CALBINDIN
TITLE    2 D9K BY NMR SPECTROSCOPY
...
SOURCE   2 ORGANISM_SCIENTIFIC: BOS TAURUS;
SOURCE   3 ORGANISM_COMMON: CATTLE;
AUTHOR    N.J.SKELTON,W.J.CHAZIN
REMARK   1  AUTH   N.J.SKELTON,J.KOERDEL,W.J.CHAZIN
REMARK   1  TITL   SIGNAL TRANSDUCTION VERSUS BUFFERING ACTIVITY
...
ATOM      1  N   LYS A   1       5.142 -12.601   2.549  1.00  2.74
ATOM      2  CA  LYS A   1       6.291 -12.612   3.435  1.00  2.74
ATOM      3  C   LYS A   1       5.854 -12.032   4.786  1.00  2.74
...
```

Obrázek 4.2: Ukázka formátu PDB na výňatku ze souboru 1CLB.pdb

Dalším formátem pro zápis souřadnic je typ souboru `xyz`, který je oproti `PDB` formátu mnohem více strohý. První řádek obsahuje počet částic v souboru, druhý řádek obsahuje volitelný komentář a zbylé řádky obsahují vždy kartézské souřadnice jedné částice.

4.1 Volba počáteční konfigurace

Generujeme-li počáteční konfiguraci modelu proteinu příliš nataženou, trvá její přechod do sbalené konfigurace dlouho. Už sbalenou počáteční konfiguraci proto generujeme tak, že první kuličce přiřadíme souřadnice [0,0,0]. Pak opakujeme následující postup: máme-li vygenerovaný řetězec n kuliček, $n+1$ kuličku zkusíme přidat do několika míst vzdálených r_{eq} od poslední přidané kuličky a vybereme tu konfiguraci, ve níž $n+1$ kulička způsobila nejmenší nárůst energie.

[1] Protein Data Bank; https://www.rcsb.org/

[2] Open Babel – converter for chemistry and molecular modeling data files; https://openbabel.org/wiki/Babel

4.2 Hešování

Výpočet párového potenciálu $E_{\mathrm{LJ}}(r_{i,j}, \xi_i, \xi_j)$ probíhá přes všechny kuličky i, j takové, že $i < j \wedge r_{i,j} < r_{\mathrm{cutoff}}$. LJ potenciál (obrázek 2.3) nabývá nenulové hodnoty i pro $r_{i,j} \geq r_{\mathrm{cutoff}}$, nicméně jde o hodnotu tak malou, že můžeme provést oříznutí potenciálu. V kulovém prostoru o poloměru r_{cutoff}, jehož středem je kulička i může být pouze omezený počet dalších kuliček k, pro něž bychom museli počítat $E_{\mathrm{LJ}}(r_{i,k}, \xi_i, \xi_k)$, časová složitost výpočtu LJ potenciálu je tedy $\mathcal{O}(N)$.

Nicméně, nalezení všech dvojic i, j splňujících podmínku $i < j \wedge r_{i,j} < r_{\mathrm{cutoff}}$ je stále úloha s kvadratickou časovou složitostí. Jedním ze způsobů jak hledání dvojic i, j zrychlit je hešování s otevřenou adresací [26].

Máme N kuliček a k přihrádek (realizované například pomocí pole), takové, že $k > N$ a k je prvočíslo. Do každé přihrádky lze umístit právě jednu kuličku. Prostor ve kterém se kuličky mohou vyskytovat rozřežme na krychličky o hraně délky r_{cutoff} a každou myšlenou krychličku označme unikátním klíčem l. Pořídíme si funkci, která každému klíči přiřadí přihrádku, tedy číslo z intervalu (0,k). Této funkci budeme říkat hešovací funkce $H(l)$. Jednou z možných realizací hešovací funkce je lineární kongruence

$$H(l) = a \cdot l \bmod k, \tag{4.1}$$

kde $a = \phi - 1 \approx 0.618033988$ a ϕ je zlatý řez. Pro každou kuličku tedy vypočteme klíč krychličky, ve které se nachází a pomocí hešovací funkce z tohoto klíče získáme index přihrádky, do které kuličku uložíme.

Během hešování nevyhnutelně nastanou kolize. Máme-li v jedné krychličce více než jednu kuličku, přiřadíme všem kuličkám stejný klíč l a tedy i stejnou přihrádku i. Řešením je pamatovat si pro každou přihrádku, zdali je prázdná. Pokud je přihrádka i prázdná, umístíme do ní kuličku, v opačném případě generujeme nový index přihrádky $i' = (i + 1) \bmod k$, do níž zkusíme kuličku uložit. Je-li plná, generujeme index i'' atd. Protože $k > N$, máme vždy zaručeno, že prázdnou přihrádku najdeme. Vyhledání kuličky v přihrádkách funguje obdobně jako uložení.

Nejprve do hešovací tabulky přidáme všechny kuličky v čase $\mathcal{O}(N)$. Pokud dostaneme dotaz na blízké sousedy kuličky m, spočteme klíč krychličky, v níž se kulička nachází a v konstantním čase z něj vygenerujeme klíče 26 okolních krychliček. Pomocí těchto klíčů pokládáme dotazy hešovací tabulce a každý bude vyřešen v čase $\mathcal{O}(N)$. Snížili jsme tedy složitost problému také na $\mathcal{O}(N)$.

Přihrádku v programu implementujeme třídou `Hashtable_item`, hešovací tabulku pak třídou `Hashtable`, jejíž metody `add_item(int grain_index)`, `find(int key)` a `get_neighbours(int index)` implementují přidání/nalezení kuličky a vrácení seznamu kuliček, sousedících se zadanou kuličkou.

Nutno podotknout, že pro proteiny s méně než 100 aminokyselin, které v práci zkoumáme, nezpůsobuje hešování téměř žádné zvýšení rychlosti výpočtu a zaručuje pouze formální časovou složitost celého programu $\mathcal{O}(N)$. V případě simulování proteinů delších však již bude zrychlení patrné.

5. Vyhodnocení výsledků

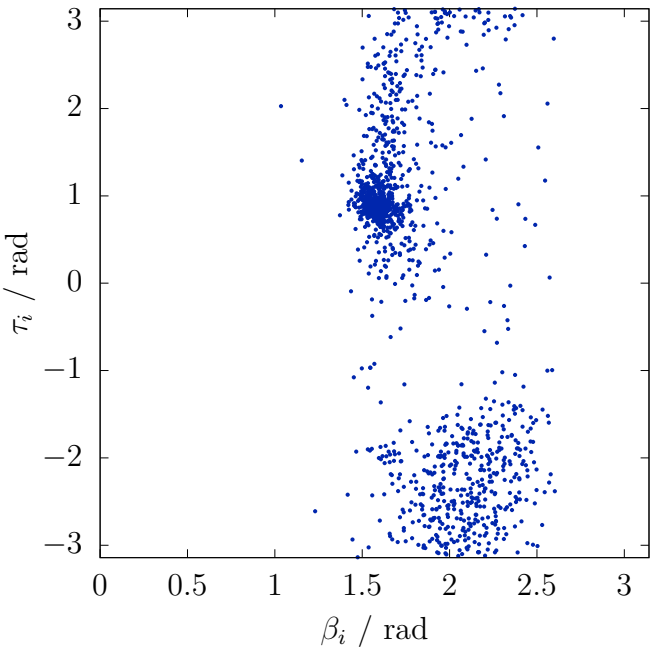
Byla provedena počítačová implementace AB i AZ modelu (viz sekce 2.1). Ač je AZ model pouze minimálním rozšířením AB modelu, i přes opakovanou kontrolu implementace AZ modelu tento model neposkytoval očekávané fyzikální závislosti studovaných veličin. Všechny následující diskutované výsledky proto odpovídají AB modelu. Jedinou výjimkou je určování RMSD, které porovnává oba modely, což je v příslušné části textu dodatečně zdůrazněno. Proces skládání a teplotní závislost termodynamických veličin (viz sekce 2.1) byla studována na šesti proteinových sekvencích se známou nativní konformací (tabulka 5.1). Těchto šest sekvencí bylo vybráno tak, aby zastupovaly nativní konformace obsahující jak pouze α-helixy nebo pouze β-skládané listy, tak i jejich kombinaci.

protein	N	sekvence
1CLB	75	MKSPEELKGIFEKYAAKEGDPNQLSKEELKLLLQTEFPS LLKGGSTLDELFEELDKNGDGEVSFEEFQVLVKKIS
1E0L	37	GATAVSEWTEYKTADGKTYYYNNRTLESTWEKPQELK
2GB1	56	MTYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDN GVDGEWTYDDATKTFTVTE
2YGS	92	MDAKARNCLLQHREALEKDIKTSYIMDHMISDGFLTIS EEEKVRNEPTQQQRAAMLIKMILKKDNDSYVSFYNAL LHEGYKDLAALLHDGIP
3M0R	58	MDETGKELVLALYDYQEKSPDEVTMKKGDILTLLNSTN KDWWKVEVNDRQGFVPAAYV
4RXN	54	MKKYTCTVCGYIYDPEDGDPDDGVNPGTDFKDIPDD WVCPLCGVGKDEFEEVEE

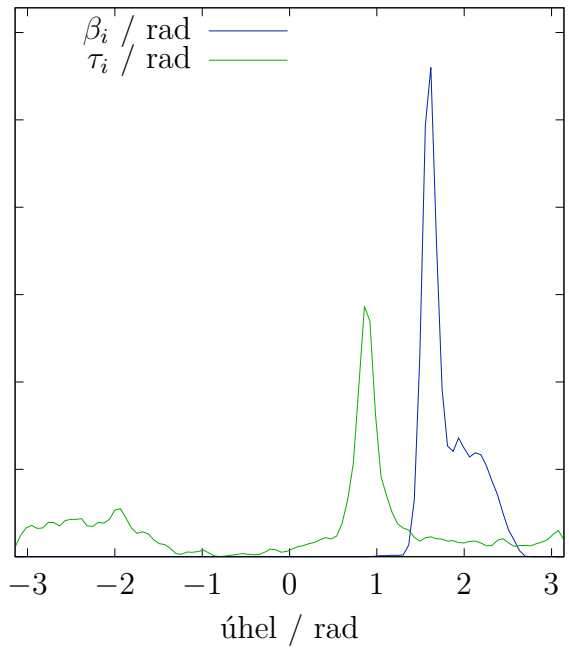
Tabulka 5.1: Proteinové sekvence použité pro stanovení termodynamických veličin.

Pro správné nastavení parametrů lokálních interakcí δ_1, δ_2 byl nejprve proveden výnos úhlových párů (β_i, τ_i) 25[1] složených struktur z PDB (obrázek 5.8). Lze pozorovat dvě oblasti s vysokým zastoupením úhlů (obrázek 5.2). Oblast $\beta_i \in$ [1.5,1.7] a $\tau_i \in [0.5, 1.2]$ odpovídá výskytu α-helixu, oblast $\beta_i \in [1.7, 2.3]$ a $\tau_i \in [-3, -2]$ odpovídá β-skládanému listu [10]. Parametry δ_1, δ_2 byly nastaveny tak, aby se rozdělení úhlu simulované molekuly co nejvíce podobalo experimentálnímu rozdělení.

[1]Šlo o sekvence krátkých proteinů 1BDD, 1CLB, 1E0G, 1E0L, 1FCA, 1GAB, 1IGD, 1LQ7, 1UTG, 1WY3, 2GB1, 2OVO, 2RJV, 2RJY, 2YGS, 3ADG, 3CQT, 3E21, 3I8Z, 3ICB, 3M0R, 3MYE, 4QUC, 4RXN, 5PTI.

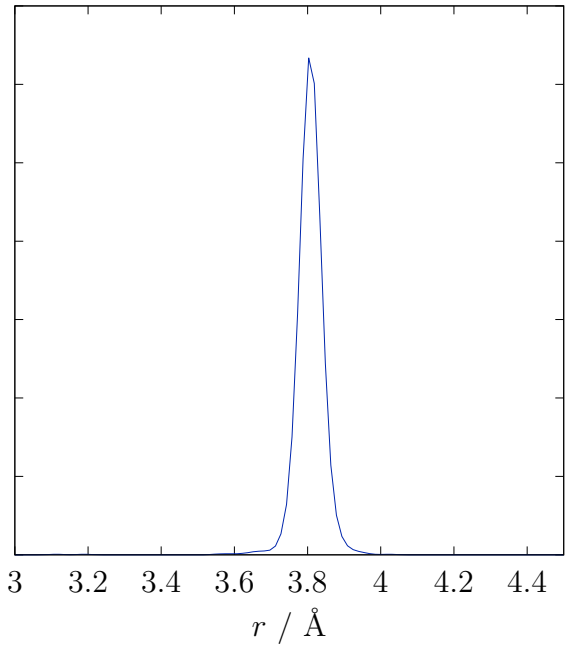


Obrázek 5.1: Výnos dvojic úhlových párů složených struktur z PDB.

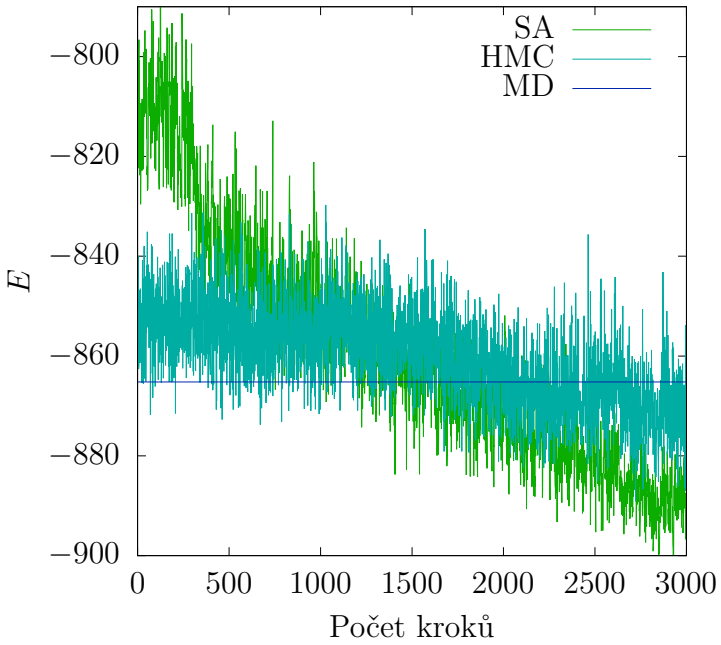


Obrázek 5.2: Hustota pravděpodobnosti výskytu jednotlivých úhlů.

Ze stejných 25 struktur byla vynesena hustota pravděpodobnosti vzdálenosti sousedních C_α atomů. Na jejím základě byla rovnovážná vzdálenost kuliček nastavena přibližně na 3.8 Å.

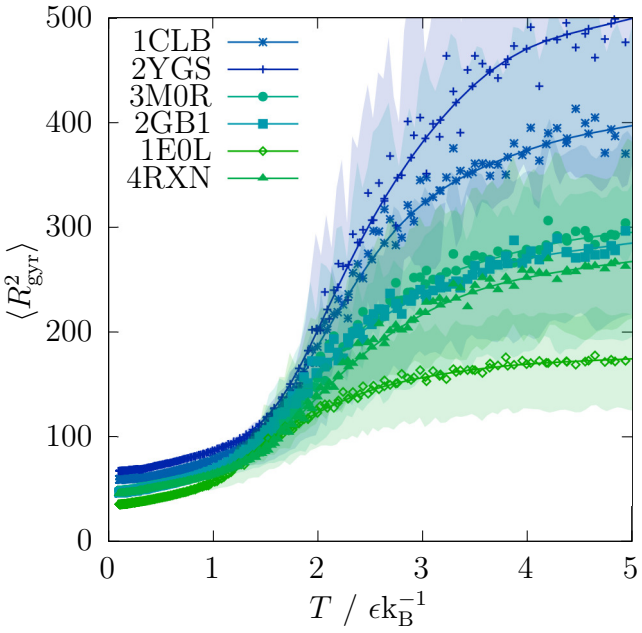


Obrázek 5.3: Hustota pravděpodobnosti výskytu vazebné délky C_α–C_α.

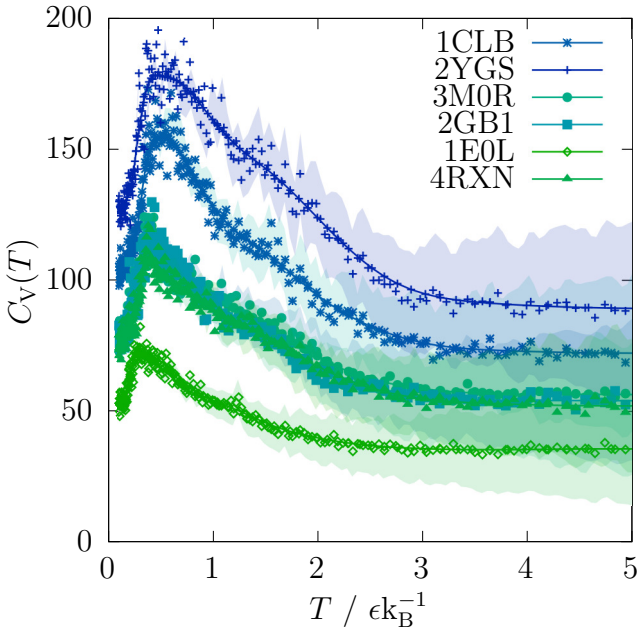


Obrázek 5.4: Ukázka vývoje energie simulovaného proteinu pro různé typy simulace.

Maximální tepelné kapacity, s nimiž souvisí přechod do nativní konfigurace [10], byly pozorovány v intervalu nízkých teplot $[0.3, 1.2]$. Gyrační poloměr se vzrůstající teplotou dle očekávání rostl. Chybu měření, vypočtenou pomocí blokové metody, znázorňuje barevný pás kolem dané křivky.

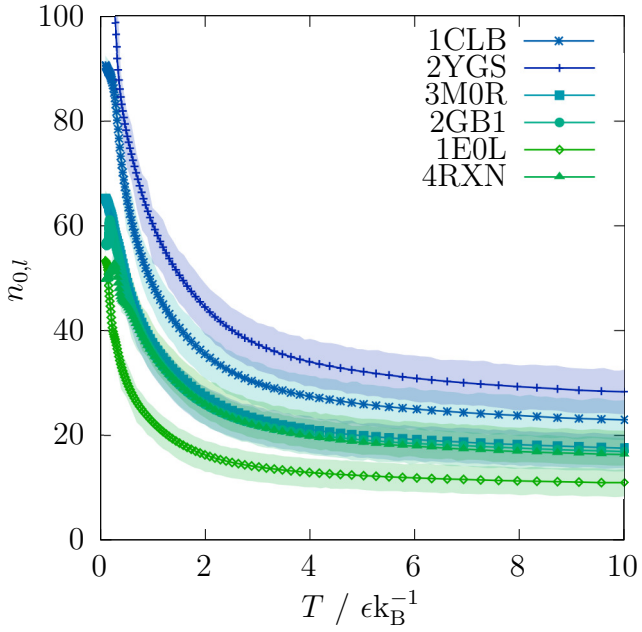


Obrázek 5.5: Gyrační poloměr simulovaného proteinu jako funkce teploty.

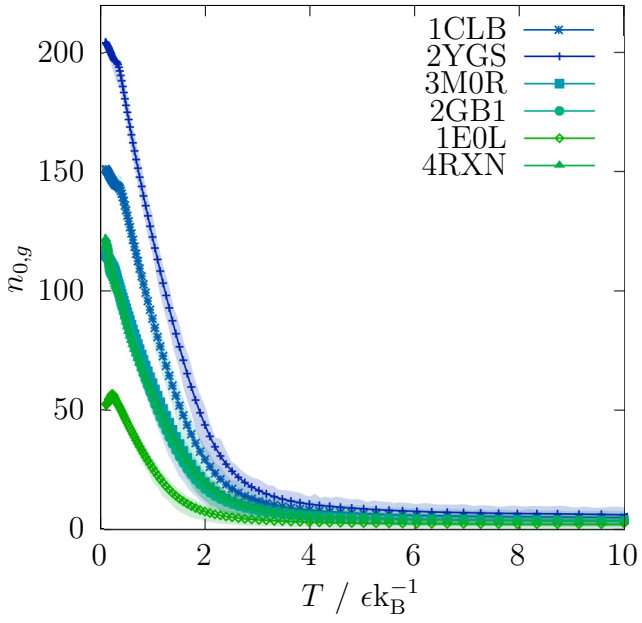


Obrázek 5.6: Tepelná kapacita simulovaného proteinu jako funkce teploty. Barevný pás kolem křivek odpovídá chybě energie.

Z měření $n_{0,l}, n_{0,g}$ vyplývá, že ke vzniku většiny kontaktů dochází v intervalu nízkých teplot $[0.1, 2.0]$. Strmý nárůst počtu kontaktů je pozorován i pro teploty nižší, než je teplota odpovídající píku tepelné kapacity. Výrazně strmější vzrůst $n_{0,g}$ oproti $n_{0,l}$ v tomto intervalu teplot naznačuje výrazné změny globální struktury.

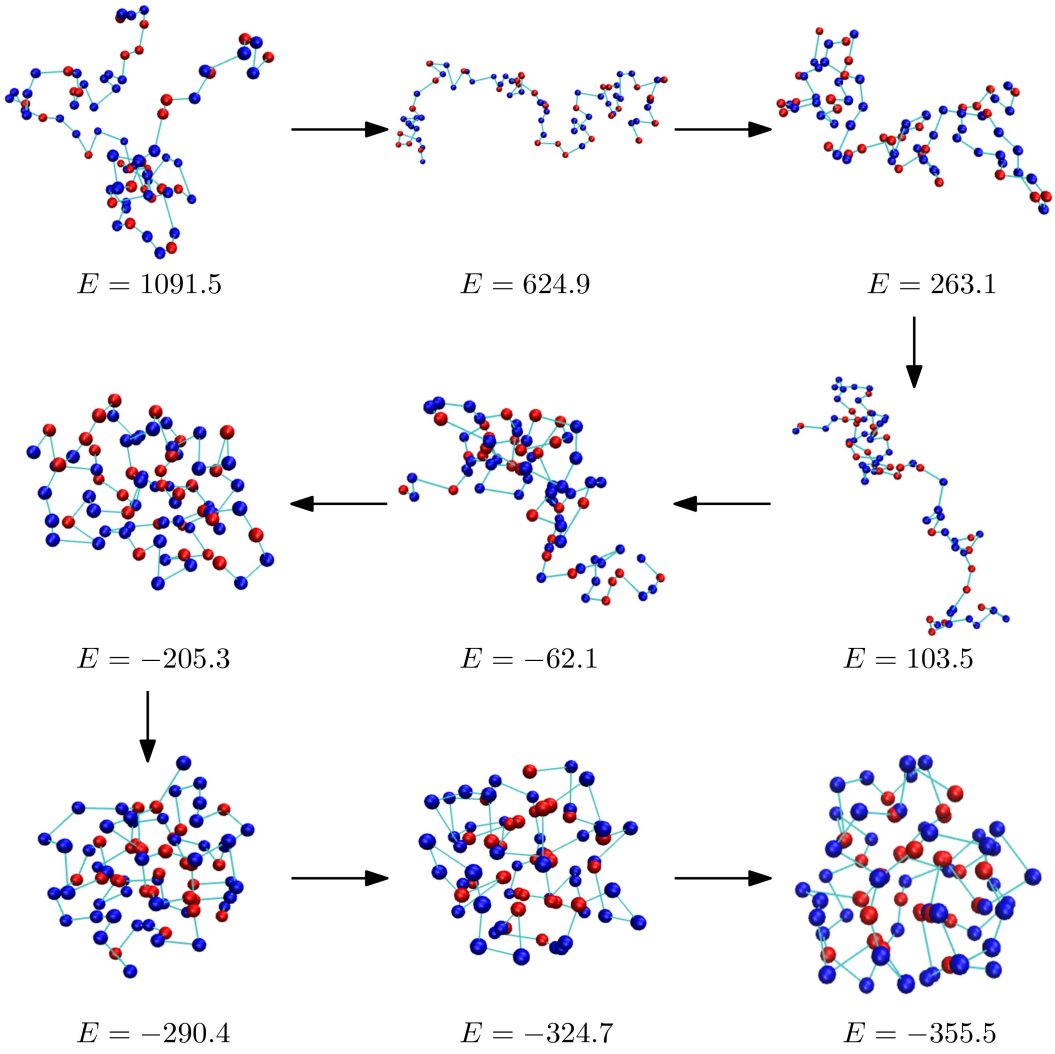


Obrázek 5.7: Počet okupovaných lokálních nativních kontaktů jako funkce teploty.



Obrázek 5.8: Počet okupovaných globálních nativních kontaktů jako funkce teploty.

Při SA simulaci využívající HMC dochází s poklesem teploty simulace k postupnému sbalování (viz obrázek 5.5) a vzniku globulární struktury. Hydrofobní kuličky se přesouvají do středu simulovaného proteinu a jejich nevazebné interakce stabilizují vzniklý útvar, což dokládá klesající energie simulovaného proteinu (viz obrázek 5.9).

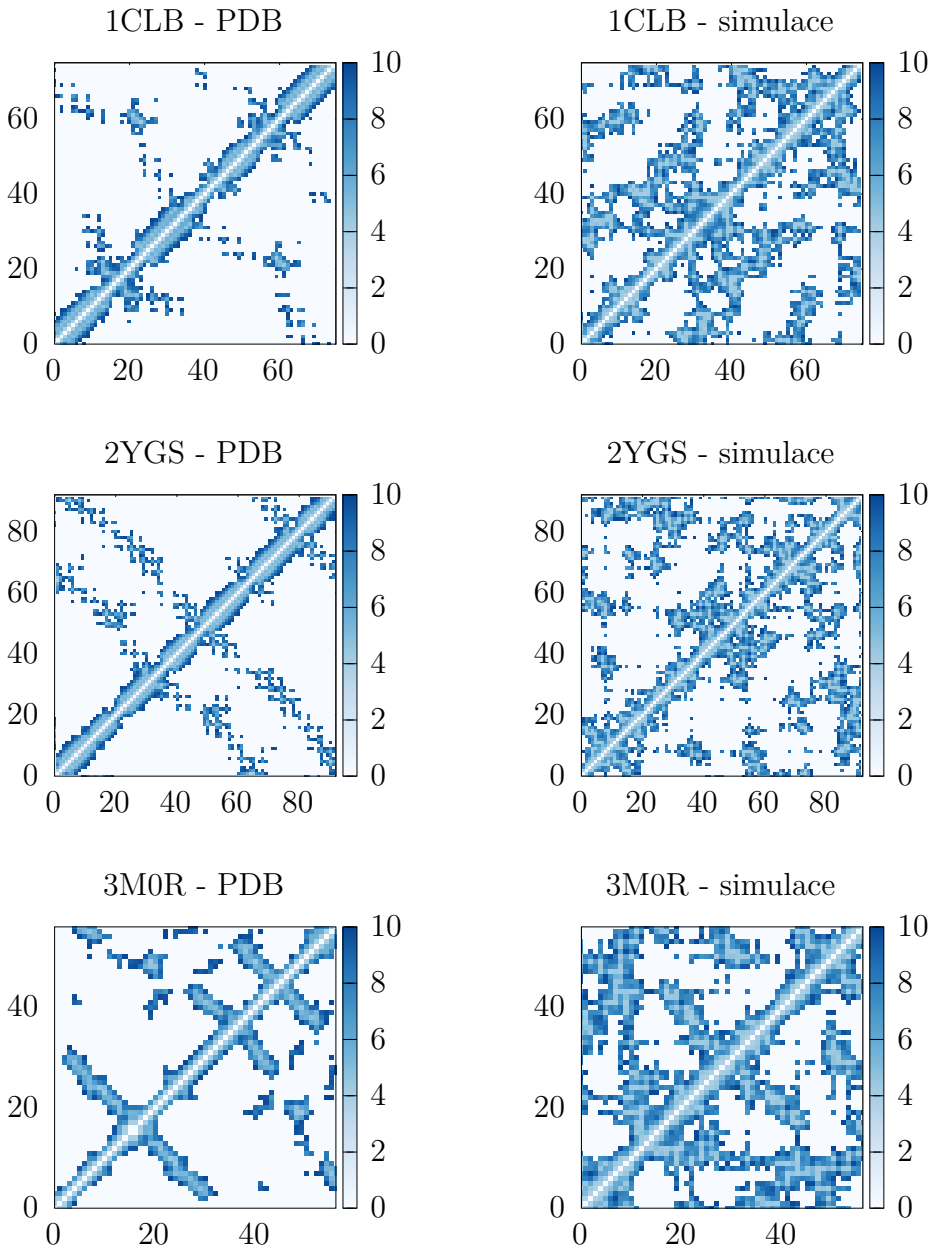


Obrázek 5.9: Postupné sbalování řetězce v průběhu SA simulace.

Porovnáním obrázku skutečné nativní konformace proteinu a struktury ze simulace nelze jednoduše určit, jak moc si jsou struktury podobné. Vedle RMSD lze podobnost struktur vystihnout pomocí map kontaktů [8]. Pro protein obsahující N aminokyselin (nebo model proteinu o N kuličkách) jde o matici M velikosti $N \times N$, jejíž prvky definujeme následovně:

$$M_{i,j} = \begin{cases} r_{i,j}, & \text{pokud } r_{i,j} < d, \\ 0, & \text{jinak.} \end{cases}$$

Pro model proteinu je $r_{i,j}$ vzdálenost kuliček i, j, pro skutečný protein jde o vzdálenost C_α–C_α aminokyselin i, j.

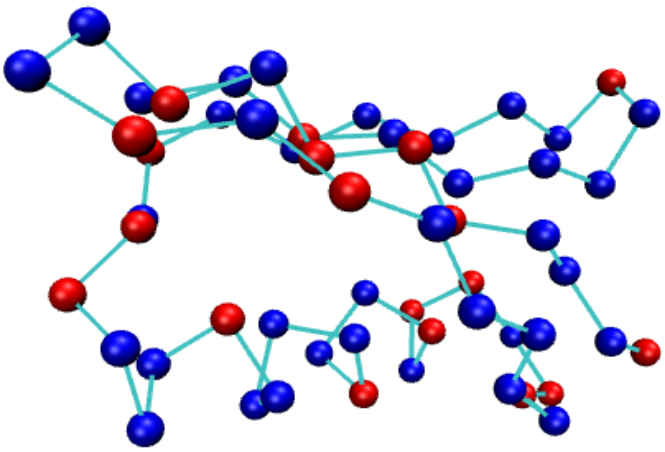


Obrázek 5.10: Porovnání kontaktních map proteinů 1CLB, 2YGS, 3M0R. Vlevo struktura z PDB, vpravo struktura ze simulace s nejnižší energií. Barevná stupnice udává vzdálenost v Å, $d = 10$.

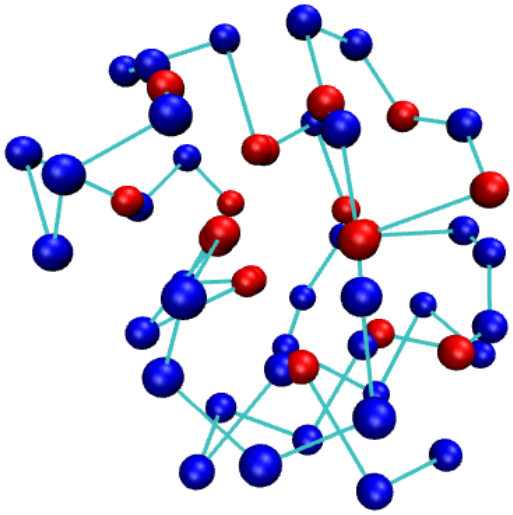
Vedle kontaktních map byla podobnost skutečné a simulované nativní konformace studována pomocí RMSD. Byl zkoumán AB i AZ model. Hodnoty RMSD se pro oba modely pohybovaly u všech sekvencí okolo 8 Å. Nejlepší simulační programy současnosti jsou schopny poskytnout RMSD $\approx$ 1 Å [13]. Nejde tedy o přesné stanovení nativní struktury. Z porovnání kontaktních map (obrázek 5.10) a získaných struktur plyne, že zkoumané modely nevykazovaly přesvědčivý vznik sekundárních struktur (viz obrázky 5.11, 5.12 a 5.13 – porovnání experimentální struktury proteinu 2GB1 a struktur, získaných simulací).

protein	$RMSD_{AB}$ / Å	$RMSD_{AZ}$ / Å
1CLB	8.03	9.43
1E0L	8.22	7.48
2GB1	7.81	7.80
2YGS	7.61	7.80
3M0R	8.37	8.30
4RXN	7.90	7.71

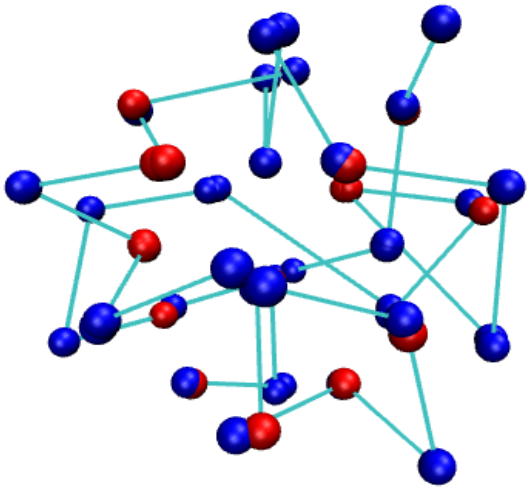
Tabulka 5.2: Nejlepší hodnoty RMSD pro AB a AZ model.



Obrázek 5.11: Experimentálně stanovená struktura proteinu 2GB1. Vykresleno jako AB model.



Obrázek 5.12: Výsledek simulace skládání proteinu 2GB1 s použitím AB modelu. RMSD = 7.81 Å.



Obrázek 5.13: Výsledek simulace skládání proteinu 2GB1 s použitím AZ modelu. RMSD = 7.80 Å. Vykresleno jako AB model.

6. Závěr

V této práci jsme navrhli dva zhrubené modely proteinu, které byly v rámci počítačové simulace použity ke studiu skládání krátkých proteinů s již známou strukturou. Vývoj simulovaného proteinu v čase zajišťovalo Hamiltonian Monte Carlo, k nalezení nativní konformace jakožto globálního energetického minima byla použita optimalizační metoda simulovaného žíhání. Hnací silou sbalování byla hydrofobní interakce, vedoucí k tvorbě hydrofobní domény uvnitř simulovaného proteinu. Modely byly implementovány jako počítačový program v jazyce C++ a pomocí simulací na superpočítači byl studován průběh protein foldingu.

Na šesti reálných proteinových sekvencích byla studována teplotní závislost gyračního poloměru, tepelné kapacity a počtu okupovaných nativních kontaktů. AZ model neposkytoval dobré fyzikální závislosti termodynamických veličin na teplotě, hodnoty RMSD však byly srovnatelné s AB modelem. Tepelná kapacita AB modelu vykazovala pík v oblasti teplot $[0.3, 1.2]$, související s přechodem do nativní konformace. Teplota, při níž byla tepelná kapacita maximální, byla závislá na studované proteinové sekvenci. Gyrační poloměr simulovaného proteinu rostl se zvyšující se teplotou simulace, lokální i globální počet okupovaných nativních kontaktů vykazoval nejstrmější růst v oblasti nízkých teplot $[0.1, 2.0]$. Získané závislosti se plně shodují se současnými poznatky o simulacích sbalování proteinů [10] [15].

Podobnost skutečné a simulací získané nativní konformace byla studována pomocí RMSD a kontaktních map, průměrné RMSD bylo 8 Å. I přes důkladné nastavení simulačních parametrů modely přesvědčivě nevykazovaly vznik sekundárních struktur. Přesnost modelů lze do budoucna zvýšit například nahrazením harmonických vazeb rigidními, prodloužením běhu simulace, nebo nastavením parametrů na základě simulací více než šesti proteinových sekvencí.

Seznam použité literatury

[1] Ken A. Dill and Justin L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):1042–1046, 2012.

[2] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chemical Reviews*, 116(14):7898–7936, 2016.

[3] Johnson A Alberts B. *Molecular Biology of the Cell; Analyzing Protein Structure and Function*. 4th edition. Garland Science, New York, 2002.

[4] T. Schmidt and A. Bergner. Modelling three-dimensional protein structures for applications in drug design. *Drug Discovery Today*, 19(7):890–897, 2014.

[5] Milan Kodíček, Olga Valentová, and Radovan Hynek. *Biochemie*. VŠCHT, 2015.

[6] Dmitry N. Ivankov and Alexei V. Finkelstein. Solution of Levinthal's paradox and a physical theory of protein folding times. *Biomolecules*, 10(2), 2020.

[7] Peter Atkins and Julio Paula. *Atkins' physical chemistry*. Oxford University press, 2008.

[8] Warshel A. Levitt, M. Computer simulation of protein folding. *Nature*, 253(5494):694–698, 1975.

[9] Ken A. Dill and Sarina Bromberg. Principles of protein folding — a perspective from simple exact models. *Protein Science*, 4(4):561–602, 1995.

[10] Anders Irbäck, Carsten Peterson, Frańk Potthast, and Ola Sommelius. Local interactions and protein folding: A three-dimensional off-lattice approach. *The Journal of Chemical Physics*, 107(1):273–282, 1997.

[11] W. Schommers. Pair potentials in disordered many-particle systems: A study for liquid gallium. *Physical Review A*, 28(6):3599–3605, 1983.

[12] Scheraga Tanaka S. Medium- and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules*, 9(6):945–50, 1976.

[13] Ewen Callaway. It will change everything: Deepmind's AI makes gigantic leap in solving protein structures. *Nature*, 588(7837):203–204, 2020.

[14] Boussinot F. Monasse, B. Determination of forces from a potential in molecular dynamics. 2014.

[15] Lizhong Zhang, He Ma, Wei Qian, and Haiyan Li. Protein structure optimization using improved simulated annealing algorithm on a three-dimensional ab off-lattice model. *Computational Biology and Chemistry*, 85:107237, 2020.

[16] Sanzo Miyazawa and Robert L. Jernigan. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3):534–552, 1985.

[17] Prof. Dr. Christian Holm. Simulation methods in physics 1. *Institute for Computational Physics*, (1):12–14, 2012.

[18] Monte carlo integration – url: `https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/monte-carlo-methods-in-practice/monte-carlo-integration`.

[19] Jiří Kolafa. *Molekulové modelování a simulace*. VŠCHT, Praha, 2019.

[20] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[21] A. Irbäck. Hybrid Monte Carlo simulation of polymer chains. *The Journal of Chemical Physics*, 101(2):16611–1667, 1994.

[22] Radford M. Neal. *Handbook of Markov Chain Monte Carlo; Chapter 5*. 1st edition. CRC Press, 2019.

[23] Scott Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science (New York, N.Y.)*, 220:671–80, 06 1983.

[24] H. Flyvbjerg and H. G. Petersen. Error estimates on averages of correlated data. *The Journal of Chemical Physics*, 91(1):461–466, 1989.

[25] Sokal A. D. Madras, N. The pivot algorithm: A highly efficient Monte Carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50(1–2):109–186, 1988.

[26] Martin Mareš and Tomáš Valla. *Průvodce labyrintem algoritmů, kap. 11*. CZ.NIC, z.s.p.o., 2017.