



Carlos Enrique Olivares Rodríguez

**Dinâmica estocástica efetiva de sequências
proteicas simplificadas**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Física do Departamento de Física da PUC-Rio

Orientador: Prof. Celia Beatriz Anteneodo de Porto

Rio de Janeiro
Julho de 2013



Carlos Enrique Olivares Rodriguez

**Dinâmica estocástica efetiva
de sequências proteicas simplificadas**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Física do Departamento de Física do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Profa. Celia Beatriz Anteneodo de Porto
Orientadora
Departamento de Física – PUC-Rio

Prof. Welles Antonio Martinez Morgado
Departamento de Física – PUC-Rio

Prof. Jürgen Fritz Stilck
UFF

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 11 de julho de 2013.

Todos os direitos reservados. Proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Carlos Enrique Olivares Rodríguez

Graduou-se em Física na Universidade Nacional de Ingeniería (Lima, Perú) em 2010.

Ficha Catalográfica

Olivares Rodríguez, Carlos Enrique

Dinâmica estocástica efetiva de sequências proteicas simplificadas / Carlos Enrique Olivares Rodríguez; orientador: Celia Beatriz Anteneodo de Porto. — Rio de Janeiro : PUC-Rio, Departamento de Física, 2013.

v., 95 f: il. ; 29,7 cm

1. Dissertação (Mestrado em Física) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Física.

Inclui referências bibliográficas.

1. Física – Tese. 2. Dinâmica estocástica. 3. enovelamento proteico. 4. dinâmica molecular. I. Anteneodo de Porto, Celia Beatriz. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Física. III. Título.

CDD: 510

Para os meus pais, Rosa e Rolando,
Pelo apoio e confiança.

Agradecimentos

As primeiras pessoas a agradecer são meus pais Rolando Olivares, que descansa em paz, e Rosa Rodriguez pelo apoio constante e nas etapas mais importantes de minha vida e por me dar muito amor.

A minha irmã Ligia Eduarda pelo amor, apoio e inspiração constante que ela representa na minha vida.

A minha esposa Cynthia por ser a mulher da minha vida e me acompanhar nos momentos bons e sobretudo nos ruins com um sorriso encantador.

A minha orientadora Celia Anteneodo, pelo apoio no trabalho e pelo cansativo sofrimento de trabalhar comigo e ainda assim manter a paciência e estimular a criatividade.

A todos meus amigos e colegas, pelas conversas, a companhia, o afeto e alento para o trabalho Eduardo, Marlon, Nuno, Victor e Sabrina do grupo de sistemas complexos da PUC-rio.

Aos meus amigos Ayrton, Melissa, Jimmy, Rafael que o destino cruzou no meu caminho.

A meus amigos peruanos pelos quais tenho um carinho especial, Enrique, Giovana, Juanpablo, Max, Lucho, Edward, Diego, Carlos L., Raquel, Cristabel, Richard e Catalina por la companhia e vivências na Cidade Maravilhosa.

Aos professores do departamento de Física da PUC-Rio pelas aulas de mestrado.

Por fim, agradeço ao CNPq por ter financiado minha bolsa de mestrado.

Resumo

Olivares Rodríguez, Carlos Enrique; Anteneodo de Porto, Celia Beatriz. **Dinâmica estocástica efetiva de sequências proteicas simplificadas**. Rio de Janeiro, 2013. 95p. Dissertação de Mestrado — Departamento de Física, Pontifícia Universidade Católica do Rio de Janeiro.

As proteínas e outros peptídeos são cadeias de aminoácidos que desempenham funções biológicas específicas dentro de um organismo. A funcionalidade dessas estruturas depende da sua organização tridimensional, portanto é importante determinar quais são os fatores que controlam o bom enovelamento. Se a sequência é conhecida em princípio poder-se-ia prever sua estrutura 3D mediante uma dinâmica molecular de todos os átomos da sequência e das moléculas de água circundantes, mas é claro que esse tipo de simulação é inviável com os recursos computacionais atuais. Alternativamente, consideramos modelos simplificados que levem em conta somente as características principais de cada monômero e das partículas do meio. Efetuamos simulações de dinâmica molecular, considerando interações do tipo Lennard-Jones entre monômeros (distinguindo entre monômeros polares e hidrofóbicos) e adicionalmente incorporando uma força estocástica (Langevin) para complementar a influência do meio aquoso. Consideramos diversas sequências lineares, simétricas e de comprimento fixo, evoluindo no espaço bi ou tridimensional. Como resultado destas simulações, podemos descrever a evolução temporal no espaço de conformações mediante variáveis efetivas ou coordenadas de reação, tais como o raio de giro, a distância entre as extremidades ou o número de contatos entre monômeros não ligados. Da análise das séries temporais dessas variáveis efetivas, extraímos os coeficientes que permitem construir seja a equação diferencial estocástica do movimento das variáveis efetivas ou a equação de Fokker-Planck associada. Estas equações para um número reduzido de graus de liberdade permitem, em princípio, obter informações sobre mudanças conformacionais, difíceis de acessar na descrição completa no espaço de fases original, de alta dimensionalidade. Discutimos as vantagens e limitações desta abordagem.

Palavras-chave

Dinâmica estocástica ; enovelamento proteico ; dinâmica molecular.

Abstract

Olivares Rodríguez, Carlos Enrique; Anteneodo de Porto, Celia Beatriz (advisor). **Effective stochastic dynamics of simplified protein sequences**. Rio de Janeiro, 2013. 95p. MSc. Dissertation — Departamento de Física, Pontifícia Universidade Católica do Rio de Janeiro.

Proteins and other peptides are aminoacid chains that perform specific biological functions within an organism. The functionality of these structures depends on their three-dimensional organization, so it is important to determine what are the factors that control the proper folding. If the sequence is known, in principle it would be possible to predict its 3D structure by means of molecular dynamics of all atoms of the sequence and the surrounding water molecules, but it is clear that this type of simulation is not feasible with the current computational resources. Alternatively, we consider simplified models that take into account only the main characteristics of each monomer and the particles of the medium. We have performed molecular dynamics simulations, considering the Lennard-Jones-like interactions between monomers (distinguishing between polar and hydrophobic monomers) and additionally incorporating a stochastic (Langevin) force to complement the influence of the aqueous medium. We considered several linear sequences, symmetric, with fixed-length, evolving in the tri or bi-dimensional space. As a result of these simulations, we can describe the temporal evolution in the space of conformations through effective variables or reaction coordinates, such as gyration radius, distance between ends or number of contacts between unbound monomers. From the analysis of the time series of the effective variables, we extract the coefficients that allow to build the stochastic differential equation of motion of the effective variables or its associated Fokker-Planck equation. These equations for a limited number of degrees of freedom provide, in principle, information on conformational changes, which are difficult to access in the description of the original, high dimensional, phase. We discuss the advantages and limitations of this approach.

Keywords

Stochastic dynamics ; protein folding ; molecular dynamics.

Sumário

1	Introdução	13
1.1	Motivação e contexto biológico	13
1.2	Modelos matemáticos simplificados	19
2	Métodos	27
2.1	Dinâmica estocástica	27
2.2	Dinâmica molecular	32
3	Sequências estudadas	40
3.1	Sequências binárias	40
3.2	Variáveis efetivas	42
4	Resultados	45
4.1	Séries temporais	46
4.2	Momentos e correlações	49
4.3	Perfis de energia	52
4.4	Coeficientes de arraste e difusão	53
5	Conclusões e perspectivas	59
	Referências Bibliográficas	62
A	Perfis de energia	65
B	Coeficientes de Kramers-Moyal	69
C	Código fonte da dinâmica molecular	71

Lista de figuras

Figura 1.1	(a) Estruturas de uma proteína. (b) Tabela dos 20 aminoácidos que compõem as proteínas, organizadas em grupos com diferentes propriedades físicas e químicas (1).	14
Figura 1.2	Estrutura química do ADN, uma molécula helicoidal formada por quatro bases nitrogenadas diferentes G, A, T, C (guanina, adenina, timina e citosina).(2)	14
Figura 1.3	Transcrição do ADN para o ARNm (3).	15
Figura 1.4	Dicionário de códons para aminoácidos (1).	16
Figura 1.5	Experimentos de Anfinsen (4).	17
Figura 1.6	Cadeia peptídica. Representação das ligações peptídicas entre o carbono α de um resíduo e o nitrogênio do grupo amino do resíduo consecutivo (em vermelho), interação que mantém unida a cadeia. Representação dos ângulos diedros ϕ e ψ .	18
Figura 1.7	Classificação das sequências peptídicas.	21
Figura 1.8	Complexidade do perfil de energia com funil, extraído de (5).	24
Figura 1.9	Sequência na rede quadrada onde os quadradinhos pretos representam os monômeros hidrofóbicos e os brancos os polares. As linhas tracejadas mostram as interações entre os HHs .	24
Figura 2.1	Gráfico do potencial (1-2) como função da distância para diferentes valores de q .	35
Figura 2.2	Gráfico do potencial da Eq. (1-3) entre monômeros hidrofóbicos para diferentes valores de energia de interação E_c .	36
Figura 2.3	Evolução temporal da variável Z (distância entre extremidades) para a sequência 1000100010001, onde 1(0) corresponde a monômero hidrofóbico (hidrofílico), para diferentes valores da temperatura indicados na figura.	38
Figura 2.4	Evolução temporal de X (distância entre dois monômeros consecutivos, representativos, 1 e 2 no caso da figura) como função do tempo, para diferentes valores do passo de integração dt e da temperatura T . De esquerda a direita $dt = 0,001; 0,01$ e $0,1$ e de cima para baixo $T = 0,01; 0,1$ e $1,0$. Grandezas adimensionais segundo será explicado no texto.	39
Figura 3.1	(a) Conformação da cadeia na rede de contatos, as linhas verdes representam contatos tipo 11, as azuis 10 e as vermelhas 00. (b) Conformação espacial.	43

Figura 4.1	Estados conformacionais de uma sequência com $N = 13$ e $H = 4$: (a) desnaturado, (b) mal-enovelado e (c) nativo. As bolinhas amarelas representam os monômeros hidrofóbicos(1s) e as de cor cinza representam os polares (0s).	45
Figura 4.2	Evolução temporal das variáveis efetivas (a) Z , R_g , R_g^P e R_g^H , que representam distâncias características da estrutura, e (b) n_{11} , n_{10} , n_{00} e n_t que representam os números de contatos entre os diversos monômeros para uma distância de corte $d_c = 1, 2\sigma$ para a sequência 1000100010001.	47
Figura 4.3	Diagrama de contatos para diferentes valores de d_c (corte), com $\sigma = 1$.	48
Figura 4.4	Evolução temporal da variável Z para as 15 sequências simétricas com $N = 13$, a $T = 0,026$.	49
Figura 4.5	Média (esquerda) e desvio padrão (direita) das variáveis efetivas R_g , Z , R_g^H e R_g^P .	50
Figura 4.6	Média (esquerda) e desvio padrão (direita) das variáveis efetivas n_{11} , n_{10} , n_{00} , n_t .	50
Figura 4.7	Autocorrelação de Z para as 15 sequências de interesse.	51
Figura 4.8	Perfil de energia para a sequência 1000100010001.	53
Figura 4.9	Perfil de energia para a sequência 0001010101000.	53
Figura 4.10	Coefficientes de Kramers-Moyal D_1 (arraste) e D_2 (difusão), para temperaturas de $T = 0,021$ até $T = 0,026$.	54
Figura 4.11	Coefficientes de Kramers-Moyal D_1 e D_2 , para temperaturas de $T = 0,021$ até $T = 0,026$.	55
Figura 4.12	(Acima) Potencial efetivo para a variável Z , calculado por integração numérica de $-D_1(Z)$. (Abaixo) Coeficiente de difusão D_2 para comparação. Para todas as sequências com extremos hidrofóbicos.	56
Figura 4.13	(Acima) Potencial efetivo para a variável Z , calculado por integração numérica de $-D_1(Z)$. (Abaixo) Coeficiente de difusão D_2 para comparação. Para todas as sequências com extremos polares.	57
Figura 4.14	Mínimos do potencial efetivo para as sequências.	57
Figura 4.15	Comparação entre os coeficientes D_2 e D_4 da sequência 1000100010001.	57
Figura A.1	Perfil de energia para a sequência 0000110110000.	65
Figura A.2	Perfil de energia para a sequência 0001010101000.	65
Figura A.3	Perfil de energia para a sequência 0001100011000.	66
Figura A.4	Perfil de energia para a sequência 0010010100100.	66
Figura A.5	Perfil de energia para a sequência 0010100010100.	66
Figura A.6	Perfil de energia para a sequência 0011000001100.	66
Figura A.7	Perfil de energia para a sequência 0100010100010.	67
Figura A.8	Perfil de energia para a sequência 0100100010010.	67
Figura A.9	Perfil de energia para a sequência 0101000001010.	67
Figura A.10	Perfil de energia para a sequência 0110000000110.	67
Figura A.11	Perfil de energia para a sequência 1000010100001.	68

Figura A.12	Perfil de energia para a sequência 1001000001001.	68
Figura A.13	Perfil de energia para a sequência 1010000000101.	68
Figura A.14	Perfil de energia para a sequência 1100000000011.	68
Figura B.1	Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0010100010100.	69
Figura B.2	Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0010010100100.	69
Figura B.3	Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0001100011000.	70
Figura B.4	Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0001010101000.	70
Figura B.5	Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0000110110000.	70

Lista de tabelas

- Tabela 3.1 Nesta tabela mostramos, os valores do índice S dado pela equação Eq. (3-1), para as sequências simétricas com $N = 13$ e $H = 4$. 41
- Tabela 4.1 Valores dos parâmetros da função de ajuste da autocorrelação de Z , $C(t) = A \exp(-t/\tau_1) + (1 - A) \exp(-t/\tau_2)$, para as sequências simétricas com $N = 13$ e $H = 4$, ordenadas segundo o índice S . 52
- Tabela 4.2 Valores do índice S dado pela equação Eq. (3-1), para as sequências simétricas com $N = 13$ e $H = 4$. Os valores min_1 e min_2 representam as posições dos mínimos de potencial das Figs. 4.12 e 4.13. 58

1

Introdução

1.1

Motivação e contexto biológico

Introduzimos nesta seção as questões que motivam o presente trabalho de pesquisa e o contexto biológico em que elas se inserem.

1.1.1

Enovelamento: questões principais

As proteínas são macromoléculas complexas de importância fundamental para o desenvolvimento de sistemas vivos e a manutenção dos mesmos (6). São cadeias de aminoácidos (Fig. 1.1) e agem como robôs da natureza em diferentes tarefas e processos da vida do organismo (7). Além das proteínas, existem outras sequências de aminoácidos, ou cadeias peptídicas, de tamanho menor, a exemplo dos hormônios, também de importância biológica.

Mas como é que a função destas moléculas está associada com a estrutura? Como atingem a forma tridimensional que precisam ter para cumprir sua função? Quais são os mecanismos de maior importância? São essas as questões de interesse na investigação do enovelamento das proteínas (*protein folding*).

A formação das proteínas inicia-se no núcleo celular a partir da informação armazenada no ADN (ácido desoxirribonucleico, Fig. 1.2). O ADN é constituído por polímeros de unidades chamados nucleotídeos, formados por grupos fosfato e um açúcar à que está ligada uma de quatro bases nitrogenadas: adenina, timina, guanina e citosina (A, T, G, C). O ADN encontra-se no núcleo celular¹, normalmente na forma de uma dupla hélice de fitas com bases complementares.

Quando a ARN-polimerase, uma enzima presente no núcleo celular, encontra-se com um *promotor*², a dupla hélice de ADN é desenrolada e lida para criar uma cadeia linear de mARN (ácido ribonucleico mensageiro)

¹Em células eucariotas.

²Promotor é a sequência inicial de ADN que marca onde o gene começa a ser sintetizado.

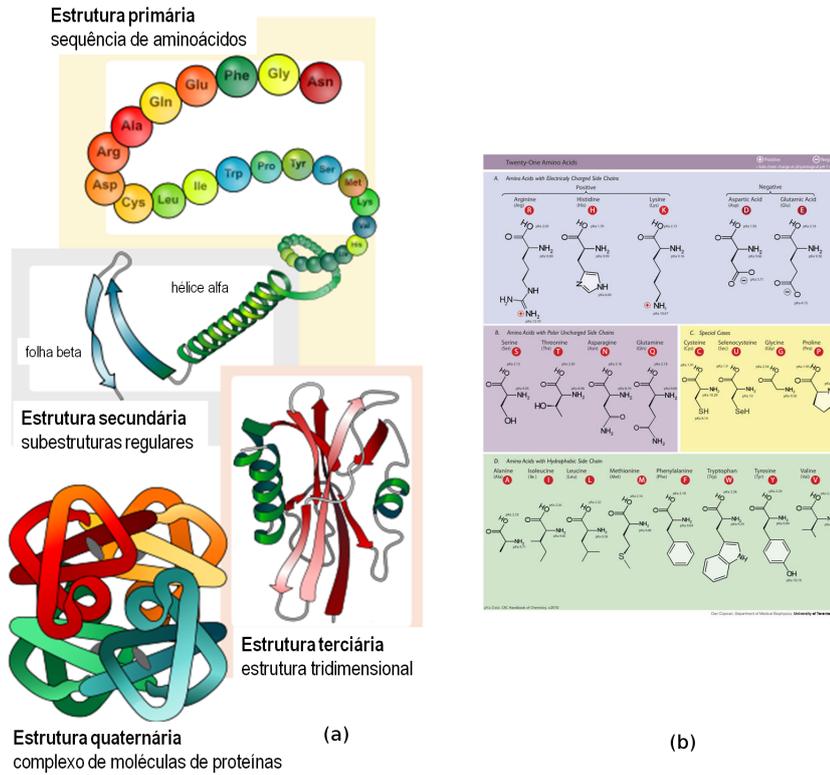


Figura 1.1: (a) Estruturas de uma proteína. (b) Tabela dos 20 aminoácidos que compõem as proteínas, organizadas em grupos com diferentes propriedades físicas e químicas (1).

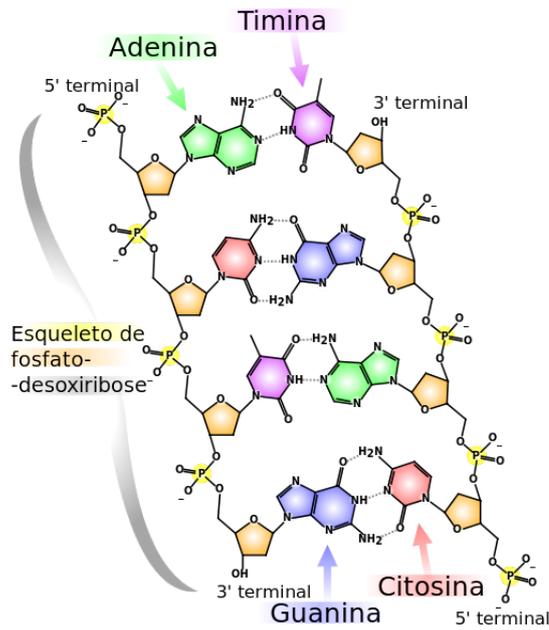


Figura 1.2: Estrutura química do ADN, uma molécula helicoidal formada por quatro bases nitrogenadas diferentes G, A, T, C (guanina, adenina, timina e citosina).(2)

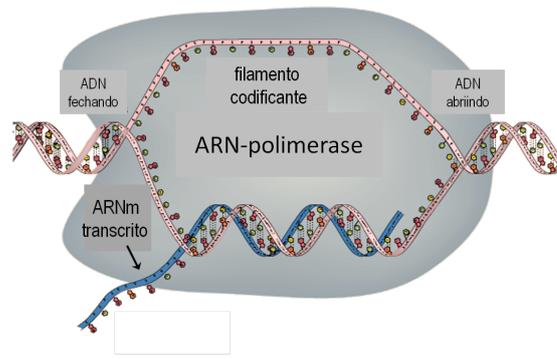


Figura 1.3: Transcrição do ADN para o ARNm (3).

que por sua vez armazena a informação do *gene* necessária para sintetizar a proteína desejada. Este processo é conhecido como *transcrição* (Fig. 1.3). Depois, o ARNm é transportado do núcleo para o citoplasma e é capturado pelo ribossomo para iniciar a tradução do ARNm à proteína.

O ribossomo faz a construção da proteína ligando os aminoácidos do ARNt (ARN de transferência) com a informação do ARNm. Esta tradução é feita baseada na existência de um dicionário de códons. No qual, para cada terna de bases do ARN (códon), existe um aminoácido correspondente, como mostrado na Fig. 1.4 (1).

Os aminoácidos tem diferentes propriedades físicas e químicas, como carga elétrica e hidrofobicidade. A estrutura destas moléculas (ver Fig. 1.1) é formada pelos grupos químicos amino ($-NH_2$), ácido carboxílico ($-COOH$) e uma cadeia lateral (chamada resíduo) que é diferente para cada aminoácido e confere suas físico-químicas particulares.

1.1.2

Experimento de Anfinsen

É natural que exista uma relação entre a sequência e a função (que depende da organização 3D), mas não é claro que a sequência linear precise de algum agente externo que faça o enovelamento por ela. Se a proteína for desnaturada (desenovelada) como fazê-la voltar a seu *estado nativo*? Quais são as interações importantes no enovelamento das proteínas? Estas foram as perguntas que Anfinsen e colaboradores (8) tentavam responder nos anos 70.

O chamado experimento de Anfinsen consiste em desnaturar (*denaturate*, *unfold*)³ uma proteína (a ribonuclease), previamente purificada, com ureia e 2-mercaptoetanol o que faz a proteína perder a sua estrutura 3D e adotar uma

³Desnaturar é mudar as condições normais da proteína para que perca a sua forma e portanto sua função.

1ª letra do códon (terminal 5')

2ª letra do códon

	U	C	A	G
U	UUU } Phe UUC } UUA } Leu UUG }	UCU } Ser UCC } UCA } UCG }	UAU } Tyr UAC } UAA } Stop UAG }	UGU } Cys UGC } UGA } Stop UGG } Trp
C	CUU } Leu CUC } CUA } CUG }	CCU } Pro CCC } CCA } CCG }	CAU } His CAC } CAA } Gln CAG }	CGU } Arg CGC } CGA } CGG }
A	AUU } Ile AUC } AUA } AUG } Met	ACU } Thr ACC } ACA } ACG }	AAU } Asn AAC } AAA } Lys AAG }	AGU } Ser AGC } AGA } Arg AGG }
G	GUU } Val GUC } GUA } GUG }	GCU } Ala GCC } GCA } GCG }	GAU } Asp GAC } GAA } Glu GAG }	GGU } Gly GGC } GGA } GGG }

Figura 1.4: Dicionário de códons para aminoácidos (1).

forma desorganizada, que podemos denominar serpentina aleatória (em inglês, *random coil*), no qual a proteína não apresenta atividade (Fig. 1.5). Mas ao remover a ureia e o 2-mercaptoetanol a proteína voltava a apresentar atividade, recuperando o estado nativo. Isto levou à conclusão de que a informação necessária para que a proteína adquira a sua forma independia de algum agente externo. Além disso, notemos que aparentemente o processo só dependia da sequência e não da história particular, como se a proteína não tivesse memória de ter sido desnaturada. Entretanto sabemos que, dependendo da história particular, uma proteína pode ficar presa em estados metaestáveis e não chegar ao estado nativo.

No segundo experimento (Fig. 1.5), remove-se o 2-mercaptoetanol sem remover a ureia e encontra-se que a atividade da ribonuclease cai para 1%, o qual é atribuído à formação de ligações dissulfeto⁴ diferentes das do estado nativo. A ribonuclease tem 8 (oito) cisteínas, portanto $7 \times 5 \times 3 = 105$ combinações de possíveis ligações dissulfeto, mas só uma existe no estado nativo (1/105, $\log \approx 1\%$). Ao remover a ureia e adicionar uma pequena quantidade de 2-mercaptoetanol para fomentar a troca de ligações dissulfeto, a proteína recupera novamente sua atividade, o que sugere que ela *procura*

⁴Ligações dissulfeto ou pontes dissulfeto são ligações covalentes entre os átomos de enxofre de duas cisteínas e são responsáveis pela estabilidade da proteína.

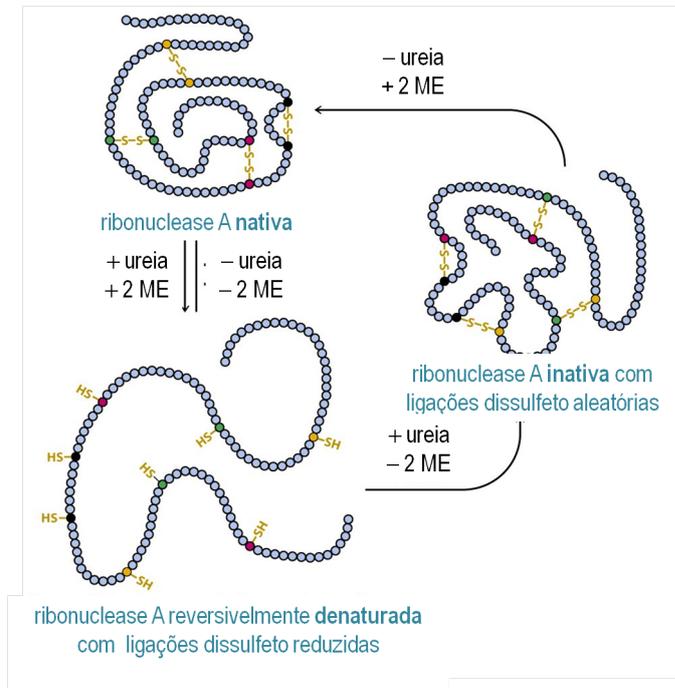


Figura 1.5: Experimentos de Anfinsen (4).

espontaneamente seu estado nativo⁵, estado que corresponde a uma das 105 possíveis conformações.

1.1.3 Paradoxo de Levinthal

Depois dos resultados de Anfinsen e colaboradores, surgiu a pergunta sobre o tempo de busca do estado nativo porque, se a sequência precisasse tentar todas as possíveis conformações, então encontrar o estado nativo demoraria um tempo tão grande quanto a idade do universo. Este é o chamado paradoxo de Levinthal (9).

O raciocínio de Levinthal foi o seguinte. Para uma sequência de 101 aminoácidos e considerando que cada um pode ter três estados diferentes (correspondendo a 3 valores diferentes dos ângulos diedrais), há $3^{100} \approx 5 \times 10^{47}$ conformações. Então, se a procura acontecesse com uma taxa de 10^{13} conformações por segundo, o tempo para testar todas elas seria de 10^{27} anos.

Mas isso não é coerente, porque se assim for nunca poderíamos ter proteínas funcionais dentro da nossa escala de tempo. Isso a não ser que exista uma rota especial para que ela encontre a sua conformação nativa nas escalas de tempo, relativamente curtas, observadas experimentalmente (10^2 s in vitro e 10^{-6} s in vivo (10)).

⁵Não chega-se a obter uma atividade completa da ribonuclease, o qual levou à descoberta de uma enzima que ajuda a corrigir as ligações dissulfeto incorretas.

Para tentar resolver este paradoxo, Dawkins (9) propôs um problema semelhante em que um macaco deve escrever a sequência de letras da frase *Methinks it is like a weasel?* com a condição de que, caso ele consiga colocar uma letra na posição certa, esta não será mais tentada. Como resultado deste modelo simples é possível corroborar que o tempo de Levinthal pode ser reduzido enormemente⁶.

Isto quer dizer que deve existir uma regra para a busca rápida no envelhecimento? E, se assim for, qual é essa regra?

1.1.4

Interações nas proteínas

Para poder entender qual é o mecanismo do envelhecimento, precisamos conhecer as interações que acontecem na proteína.

A primeira interação a mencionar é a que mantém a cadeia unida. Isto ocorre mediante ligações peptídicas (ver Fig. 1.6), que são ligações covalentes entre o carbono alfa de um aminoácido e o nitrogênio do grupo amino do aminoácido adjacente. Esta é a ligação de maior estabilidade, com energia da ordem de 50 – 250 kcal/mole, a qual confere-lhe estabilidade à temperatura ambiente $K_B T = 0.617$ kcal/mole.

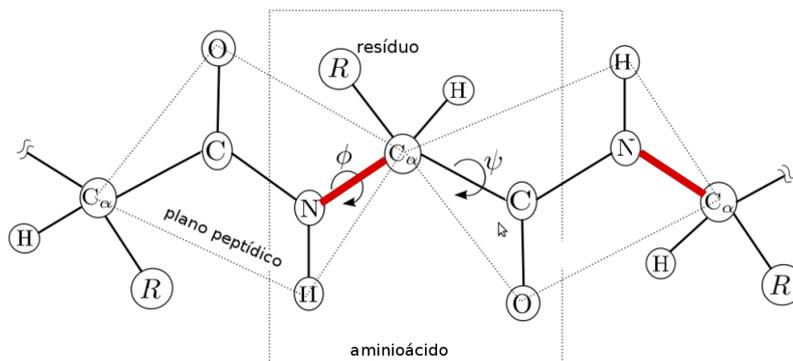


Figura 1.6: Cadeia peptídica. Representação das ligações peptídicas entre o carbono α de um resíduo e o nitrogênio do grupo amino do resíduo consecutivo (em vermelho), interação que mantém unida a cadeia. Representação dos ângulos diedros ϕ e ψ .

A parte atrativa da interação de van der Waals, devida à interação entre os momentos dipolares instantâneos induzidos dos átomos interagentes, é fraca e decai com a sexta potência da distância. A parte repulsiva é forte a curta distância, entretanto é importante para nosso problema somente no caso de existir superfícies complementares envolvendo muitos átomos (efeitos estéricos).

⁶Para mais detalhes, ver referência (9).

Estes efeitos reduzem o número de conformações possíveis (limitando os valores dos ângulos diedros ϕ e ψ (Fig. 1.6).

Com relação aos efeitos eletrostáticos, o principal é devido à carga elétrica dos aminoácidos (em condições fisiológicas de pH). Entretanto, as interações coulombianas não contribuem substancialmente para a estrutura e estabilidade das proteínas (11).

Existem outras interações importantes como as pontes de hidrogênio (essencialmente interações entre moléculas polares em que o hidrogênio de uma é atraído por um átomo altamente eletronegativo da outra, no caso da água o oxigênio). Hélices alfa e folhas beta são produto de pontes de hidrogênio locais e não locais que são responsáveis da estrutura tridimensional de pequenos peptídeos⁷ ou regiões de uma proteína, mas não de peptídeos maiores (11). Por outro lado, Kauzmann argumentou que a interação entre a proteína desnaturada e a água do meio é aproximadamente da mesma ordem de grandeza que as pontes de hidrogênio no estado nativo, o qual levou-lhe à conclusão de que as pontes de hidrogênio não podem ser relevantes para o enovelamento (11).

As forças consideradas mais relevantes para o enovelamento são as chamadas hidrofóbicas⁸. Interações entre moléculas de água tendem a ser mais favoráveis que entre a água e certas partes da proteína, o que força a proteína a adotar formatos particulares para que essas partes ditas hidrofóbicas (que repelem a água) não fiquem expostas ao solvente aquoso. A hidrofobicidade de um resíduo (Fig. 1.1b) é determinada por sua distribuição de cargas e tamanho. Quanto menos polar e quanto mais interrompe a estrutura da água, mais hidrofóbico. A energia associada ao efeito hidrofóbico é da ordem de 2 kcal/mol (10).

1.2

Modelos matemáticos simplificados

As proteínas são sistemas de alta complexidade, formados por monômeros com diferentes estruturas e tipos de interações, com diferentes sequências e comprimentos. Por outro lado, em princípio deveria ser usado um formalismo quântico. Entretanto, a aproximação de Born-Oppenheimer permite a separação da função de onda molecular em uma do núcleo e outra eletrônica devido à grande diferença de massas. Então o sistema pode ser tratado classicamente, em função das posições do núcleo (12). Além do mais, a molécula

⁷Peptídeos são sequências de aminoácidos, usualmente curtas.

⁸Mas cabe comentar que não existe uma força hidrofóbica propriamente dita o que se tem é a influência da água sobre a cadeia.

não está isolada, mas interage com o meio ao redor que é o que induz o enovelamento e mantém a estrutura nativa (13). Uma descrição matemática precisa deverá levar em conta todos estes fatores. Mas segundo o apresentado na seção anterior, algumas interações são de maior importância do que outras, o qual permite reduzir a descrição do problema.

É claro que em um sistema biológico, encontramos um número muito grande de unidades de uma dada proteína, constituindo um verdadeiro sistema termodinâmico. Uma única proteína, mesmo sendo uma macromolécula, é um sistema pequeno para ser considerado um sistema termodinâmico no mesmo sentido que o anterior. Entretanto, podemos considerar as suas diferentes conformações como um *ensemble*, como os devidos pesos estatísticos, e a partir dele calcular propriedades termodinâmicas.

A seguir descreverei alguns modelos matemáticos de polímeros aptos, em particular, para cadeias peptídicas.

1.2.1

Descrição estatística de um polímero

Um polímero linear é constituído por monômeros unidos por ligações covalentes de forma sequencial. Um polímero com $n + 1$ monômeros idênticos (e sem estrutura interna) pode ser representado matematicamente mediante o conjunto de n vetores $\{\Delta\mathbf{r}_j = \mathbf{r}_j - \mathbf{r}_{j-1}, \text{ com } j = 0, \dots, n\}$, que correspondem às n ligações e contém a mesma informação que o conjunto de coordenadas $\{\mathbf{r}_j\}$ quando a posição do monômero $j = 0$ é fixa (10).

Podemos calcular a densidade de probabilidade associada a uma ligação $\Delta\mathbf{r}$, mediante o peso de Boltzmann, como sendo

$$p(\Delta\mathbf{r}) = \exp(-\beta u(|\Delta\mathbf{r}|)),$$

onde u é a energia potencial da interação covante entre monômeros ligados e $\beta \equiv 1/(K_B T)$. A energia potencial total do polímero pode ser expressa como

$$U(\{\Delta\mathbf{r}_j\}) = \sum_{j=1}^n u(|\Delta\mathbf{r}_j|) + W(\{\mathbf{r}_j\}),$$

onde W contém as interações entre monômeros não ligados e as interações com o meio. Portanto, a densidade de probabilidade de uma dada conformação será

$$P_{conf}(\{\Delta\mathbf{r}_j\}) = \frac{1}{Z} \left[\prod_{j=1}^n p(\Delta\mathbf{r}_j) \right] \exp(-\beta W(\{\Delta\mathbf{r}_j\})), \quad (1-1)$$

sendo Z o fator de normalização que fornece a função de partição e que é obtido integrando sobre as coordenadas, ou seja,

$$Z = \int \left[\prod_{j=1}^n p(\{\Delta \mathbf{r}_j\}) \right] \exp(-\beta W(\{\Delta \mathbf{r}_j\})) \prod_{j=1}^n d^3 \Delta \mathbf{r}_j.$$

No caso mais simples em que $W = 0$, a distribuição de probabilidade da variável \mathbf{R} , correspondente ao vetor do início ao fim da cadeia ($\mathbf{R} \equiv \mathbf{r}_n - \mathbf{r}_0$), no limite de n grande, teria a forma (10)

$$P_n(\mathbf{R}) = \left(\frac{3}{2\pi n l^2} \right)^{3/2} \exp\left(-\frac{3|\mathbf{R}|^2}{2n l^2}\right).$$

Isso porque, no caso $W = 0$, \mathbf{R} é a soma de n vetores aleatórios independentes ($\Delta \mathbf{r}_j$) e pelo teorema central do limite sua distribuição tende a uma distribuição normal.

1.2.2 Modelo de Go

O conjunto de todas as sequências peptídicas possíveis é incrivelmente grande⁹, mas nem todas elas são enoveláveis. Dentre as que podem ser enoveladas, algumas tem um enovelamento rápido mas nem todas apresentam uma função biológica evolutivamente favorável para formar parte da natureza (ver esquema de conjuntos na Fig. 1.7).

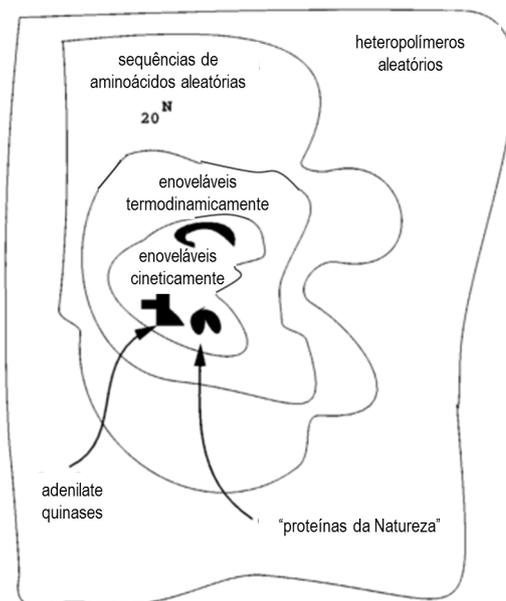


Figura 1.7: Classificação das sequências peptídicas.

Para a construção de modelos, devem ser feitas algumas suposições que simplifiquem o problema permitindo uma melhor compreensão do enovela-

⁹Para uma cadeia de 100 aminoácidos existem $20^{100} \sim 10^{130}$ combinações.

mento. Como foi descrito na seção anterior o fator de maior importância no envelhecimento é a interação hidrofóbica.

O modelo de Go (14) é o primeiro modelo que visa entender o problema do envelhecimento do ponto de vista físico e se baseia em considerar o estado nativo como um estado energeticamente ótimo. A energia é medida em termos da presença ou não de contatos nativos. Entre dois monômeros não ligados existe um contato se a distância entre eles é menor que certo valor d_0 . Se esse contato também ocorre na estrutura nativa, se diz que é um contato nativo. Estes contatos são considerados energeticamente favoráveis e a energia total de uma sequência é calculada como

$$E = Q\epsilon \equiv qE_*,$$

onde $\epsilon < 0$ é a energia média associada a um contato nativo e Q o número de contatos nativos. Esta energia também pode ser expressa em função da fração de contatos nativos $q = Q/Q_{max}$ (que tem valor unitário para a proteína ativa), sendo $E_* = \epsilon Q_{max}$ a energia do estado nativo.

Pode-se notar que o modelo assume que todo contato nativo contribui igualmente, além do mais não apresenta contribuições aleatórias independentes. Este modelo é conveniente para prever a grande escala se a estrutura nativa foi atingida mas não para analisar sua evolução. A maior crítica do modelo de Go é que usa a resposta para resolver o problema.

1.2.3

Modelo do energia aleatória

O modelo REM (sigla em inglês de *Random Energy Model*) é o modelo mais simples para as proteínas onde as conformações são estabelecidas pelo conjunto de contatos $\{\Delta(i, j)\}$, os quais são definidos como 1 se o monômero i está em contato com o monômero j e 0 se não. Assim a energia da sequência é

$$E(\Delta(i, j)) = \sum_{i < j} B_{i,j} \Delta(i, j),$$

onde $B_{i,j}$ são variáveis aleatórias com densidade de probabilidade

$$p_{contact}(B_{i,j}) = \frac{1}{\sqrt{2\pi B^2}} \exp(-B_{i,j}^2/(2B^2)).$$

As propriedades deste modelo foram discutidas por Bryngelson e Wolynes (15) mediante uma aproximação de energias aleatórias (modelo de energias aleatórias, em inglês REM) na qual é suposto que as energias de duas con-

formações diferentes devem ser consideradas variáveis independentes, de modo que se chamamos $P(E)$ à distribuição de probabilidade de uma conformação, então a probabilidade conjunta é

$$P(E_1, E_2) = P(E_1)P(E_2),$$

é dizer, supõe que as probabilidades conjuntas são fatorizáveis. Adicionalmente, a energia é considerada distribuída gaussianamente.

$$P(E) = (2\pi N\varepsilon^2)^{-1/2} \exp\left(-\frac{E^2}{2N\varepsilon}\right).$$

Estas suposições permitem realizar cálculos analíticos para obter o diagrama de fases de sequências lineares, concluindo-se sobre a existência de quatro estados ou fases para as cadeias lineares: não enovelado, enovelado, congelado e mal-enovelado congelado.

1.2.4

Modelo do perfil de energia com funil

Baseia-se numa descrição estatística em analogia com os fenômenos críticos em vidros de spin, a partir dos trabalhos de modelo de energia aleatória de Derrida (16). Modela o enovelamento como ocorrendo num complexo espaço de fases com muitos mínimos locais, que deve ter uma forma de funil (ou canalizado) que permita chegar ao estado nativo do sistema, como mostrado na Fig. 1.8.

Estas conclusões foram tiradas também a partir dos modelos simplificados de Brygelson e Wolynes (15) referidos antes, dos quais foi derivada a forma do funil que orienta a evolução para o estado nativo.

1.2.5

Modelo HP

Dentro deste modelo, uma proteína é vista como uma sequência binária de aminoácidos hidrofóbicos (H) e hidrofílicos (P), daí o nome do mesmo. Lau e Dill usaram basicamente um modelo de polímero na rede, considerando uma rede quadrada (bidimensional). Caminhadas autoevitantes (*self avoiding walk*) na rede quadrada modelam realizações da proteína no espaço bidimensional.

Na rede quadrada, o número de coordenação (número de vizinhos) é $z = 4$, o número de orientações possíveis para cada ligação interna será $z - 1 = 3$ e o número de possíveis conformações $(z - 1)^3$. Porém, cada monômero tem um volume que deve ser excluído. Portanto, o número de conformações deve diminuir a $(a - 1)^3$ onde $(a < z)$. Cálculos numéricos estimam um valor de

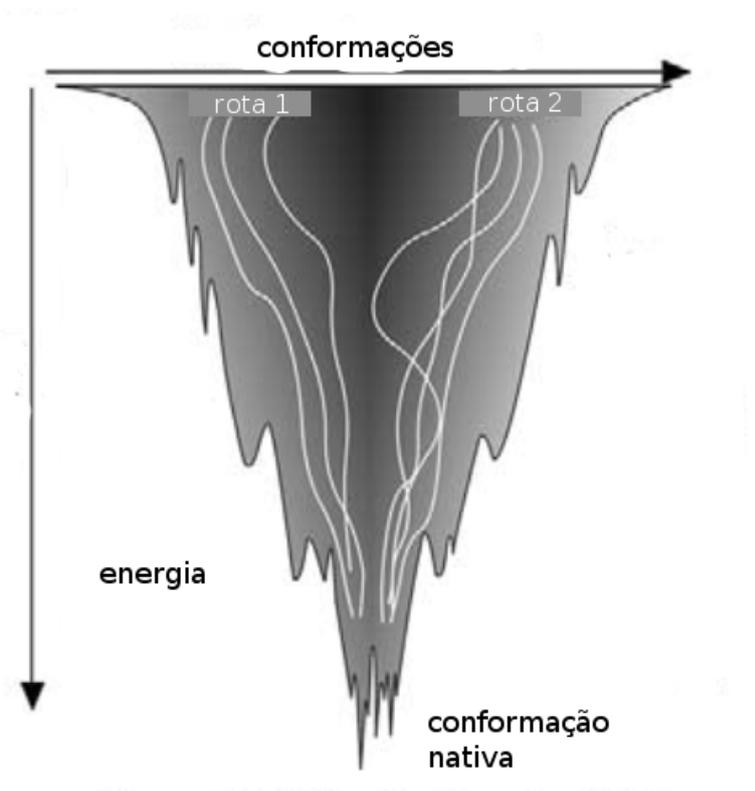


Figura 1.8: Complexidade do perfil de energia com funil, extraído de (5).

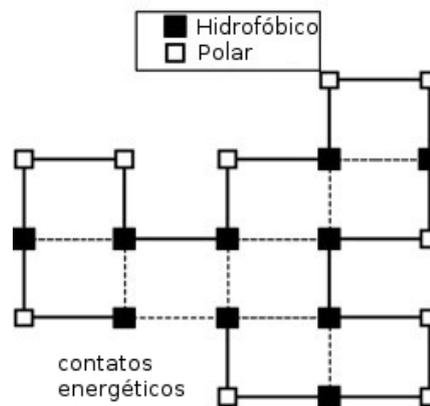


Figura 1.9: Seqüência na rede quadrada onde as quadradinhos pretos representam os monômeros hidrofóbicos e os brancos os polares. As linhas tracejadas mostram as interações entre os HH s.

$a \sim 2,71$ e para $N = 10$, onde N é o comprimento da cadeia, são obtidas 2034 conformações possíveis. Por outro lado no conjunto das seqüências, o número de elementos será 2^N .

Eles consideram todas essas seqüências (1024) e simulam as diferentes realizações conformacionais de cada uma de elas, calculando a energia. Supondo que somente os contatos HH contribuem energeticamente, eles concluem que

existe um único estado nativo (mínimo absoluto da energia). Assim, o modelo HP proposto por Lau e Dill (17) relaciona a sequência de aminoácidos da proteína e sua estrutura nativa.

1.2.6 Modelo de Loís

Lois (18) propôs um campo de forças simplificado, considerando dois tipos de monômeros: hidrofóbicos e hidrofílicos, para levar em conta a interação com o meio.

O potencial Φ_{cc} , dado pela Eq. (1-2), é encarregado de manter unida a cadeia, governando a interação entre monômeros ligados. Pode-se observar que o valor de q determina a flexibilidade da cadeia, é dizer este valor define quanto uma ligação entre monômeros consecutivos pode ser esticada. No nosso caso foi escolhido o valor $q = 0,13$. Aliás deve-se notar que a função limita as distâncias r_{ij} a valores menores que $1 + q$, ou seja $r_{ij} < 1,13$.

$$\Phi_{cc}(\bar{r}_{ij}) = \begin{cases} \epsilon(\bar{r}_{ij}^{-12} - 2\bar{r}_{ij}^{-6} + 1), & \bar{r}_{ij} \leq 1, \\ -\epsilon \log(1 - q^{-2}(\bar{r}_{ij} - 1)^2), & \bar{r}_{ij} > 1, \end{cases} \quad (1-2)$$

onde as variáveis foram adimensionalizadas e $\epsilon > 0$.

O potencial Φ_{att} que define a interação entre monômeros hidrofóbicos não ligados é dado por

$$\Phi_{att}(\bar{r}_{ij}) = -\epsilon E_c(\bar{r}_{ij}^{-12} - 2\bar{r}_{ij}^{-6}). \quad (1-3)$$

Finalmente, Φ_{rep} regula a interação entre monômeros não ligados em que participa pelo menos um monômero hidrofílico:

$$\Phi_{rep}(\bar{r}_{ij}) = \begin{cases} \epsilon(\bar{r}_{ij}^{-12} - 2\bar{r}_{ij}^{-6} + 1), & \bar{r}_{ij} \leq 1, \\ 0, & \bar{r}_{ij} > 1. \end{cases} \quad (1-4)$$

A partir deste modelo, Lois (18) mostra que a pesar de que algumas proteínas não possuem um perfil de energia canalizado (*funneled landscape*), elas podem enovelar-se se a taxa com a qual a temperatura (ou algum outro fator como a concentração) varia é menor do que um valor crítico. Se não for, pode ocorrer um mal-enovelamento (*misfolding*)¹⁰

1.2.7 Dinâmica estocástica efetiva

O enovelamento de proteínas é claramente um problema multidimensional mas como no caso dos sistemas termodinâmicos em que tem-se um

¹⁰Enovelamento indesejado, ou seja, que não cumpre sua função biológica.

número muito grande de partículas e mesmo assim (e por isso) podem-se obter descrições termodinâmicas a partir de umas poucas variáveis como pressão, volume, temperatura, aqui também podemos em princípio obter uma descrição em termos de um número reduzido de variáveis que caracterizem o enovelamento.

É por esta razão que pretendemos encontrar uma (ou algumas poucas) variáveis efetivas capazes de caracterizar a conformação do sistema, a exemplo do raio de giro, a distância entre extremidades ou o número de contatos nativos. Nesse caso, pode-se obter uma descrição simplificada da dinâmica do enovelamento (19), reduzindo a dimensionalidade do problema original.

No capítulo seguinte revisaremos os principais conceitos de processos estocásticos.

2 Métodos

Neste capítulo revisaremos os principais aspectos técnicos envolvidos na nossa análise, sem pretender realizar uma descrição exaustiva.

2.1 Dinâmica estocástica

A dinâmica estocástica estuda processos de comportamento não determinista. Esse comportamento, aparentemente errático, surge devido ao grande número de fatores externos e internos que foram desprezados como sendo secundários mas que interferem na evolução do sistema. Estes fatores podem ser considerados ruídos sobre o sistema. Portanto é necessário efetuar uma descrição probabilística. No enovelamento, os efeitos de agitação térmica do meio sobre a cadeia podem ser tratados como um ruído e assim simplificar a descrição matemática das moléculas de água no meio mediante um processo estocástico apropriado.

2.1.1 Processos estocásticos

Um processo estocástico pode ser pensado como uma função aleatória do tempo, em que a cada instante (possível valor de um índice t , seja ele discreto ou contínuo) é atribuída uma variável aleatória X_t . O processo é definido completamente conhecendo as densidades de probabilidade conjuntas das variáveis aleatórias para os diferentes valores de t .

As propriedades que caracterizam diferentes processos estocásticos, dependem da natureza do espaço de estados de cada X_t , da natureza do conjunto de índices t e especialmente das relações de dependências entre as variáveis X_t , que derivam das densidades de probabilidade conjuntas e que também podem ser expressas em termos de probabilidades condicionadas.

A *densidade condicional* $f_{X_{t_2}|X_{t_1}}(x_2, x_1) \equiv f_{1|1}(x_2, t_2|x_1, t_1)$ é dada por:

$$f_{1|1}(x_2, t_2|x_1, t_1) = \frac{f_2(x_2, t_2; x_1, t_1)}{f_1(x_1, t_1)}, \quad (2-1)$$

que verifica a condição de normalização

$$\int dx_2 f_{1|1}(x_2, t_2 | x_1, t_1) = 1. \quad (2-2)$$

No caso geral:

$$\begin{aligned} f_{l|k}(x_1, t_1; \dots; x_l, t_l | x_{l+1}, t_{l+1}; \dots; x_{l+k}, t_{l+k}) &= \\ &= \frac{f_{l+k}(x_1, t_1; \dots; x_l, t_l; x_{l+1}, t_{l+1}; \dots; x_{l+k}, t_{l+k})}{f_k(x_{l+1}, t_{l+1}; \dots; x_{l+k}, t_{l+k})}. \end{aligned} \quad (2-3)$$

Um processo estocástico será estacionário quando as distribuições conjuntas não mudam ao substituir $t \rightarrow t + \tau$. Em particular, os momentos não são afetados por um deslocamento no tempo. Para que isso ocorra é necessário e suficiente que

$$f_n(x_1, t_1 + \Delta t; \dots; x_n, t_n + \Delta t) = f_n(x_1, t_1; \dots; x_n, t_n). \quad (2-4)$$

para todo n, τ e $\{t_1, \dots, t_n\}$. Da Eq. (2-4) vemos que $f_1(x_1, t_1) = f_1(x_1) \equiv$ independente do tempo, para um processo estacionário.

2.1.2

Processos markovianos

Um processo de Markov é um processo estocástico que somente tem memória de um tempo anterior. Em outras palavras, se o valor presente é conhecido, o conhecimento futuro não é alterado por informação adicional sobre o passado. Assim, a probabilidade condicional deste processo perde a memória das situações anteriores à última medida. Logo

$$f_{1|n-1}(x_n, t_n | x_1, t_1; x_2, t_2; \dots; x_{n-1}, t_{n-1}) = f_{1|1}(x_n, t_n | x_{n-1}, t_{n-1}). \quad (2-5)$$

Apenas o conhecimento sobre o ponto (x_{n-1}, t_{n-1}) é necessário para sabermos o que acontece probabilisticamente em (x_n, t_n) .

Estes tipos de processos são totalmente caracterizados (ou seja, sua hierarquia de densidades de probabilidade f_n) pelas densidades $f_{1|1}$ e f_1 . Por exemplo, para $t_1 < t_2 < t_3$:

$$\begin{aligned} f_3(x_1, t_1; x_2, t_2; x_3, t_3) &= f_2(x_1, t_1; x_2, t_2) f_{1|1}(x_3, t_3 | x_2, t_2; x_1, t_1) \\ &= f_1(x_1, t_1) f_{1|1}(x_2, t_2 | x_1, t_1) f_{1|1}(x_3, t_3 | x_2, t_2). \end{aligned} \quad (2-6)$$

Se soubermos a forma de $f_{1|1}(x_2, t_2 | x_1, t_1)$ e $f_1(x_1, t_1)$ podemos calcular todas as quantidades de interesse para o processo. Devido a essas relações,

as probabilidades condicionais também são conhecidas como *probabilidades de transição*. Podemos proceder da mesma maneira para todas as densidades f_n dessa hierarquia. Assim podemos expressar f_n por um produto de probabilidades condicionais e f_1 . Isto torna a hierarquia muito mais manejável que no caso geral.

Num processo markoviano só há memória do valor da variável aleatória para o instante da última medida. O intervalo de tempo $t_2 - t_1$ da probabilidade condicional $f(x_2, t_2 | x_1, t_1)$ de um processo de Markov é arbitrário. Se a diferença é grande, a dependência de f em x_1 será pequena (ou seja, a memória do valor da variável aleatória é quase inexistente). Se a diferença de tempo é infinitesimal, a densidade de probabilidade condicional terá valor exato em x_1 , ou seja, $\lim_{t_2 \rightarrow t_1} f(x_2, t_2 | x_1, t_1) = \delta(x_1 - x_2)$.

2.1.3

Coeficientes de Kramers-Moyal

Da Eq. (2-3), a densidade de probabilidade $P(x, t + \Delta t)$ ¹ em um instante $t + \Delta t$ e a densidade de probabilidade $P(x, t)$ em um instante t estão conectadas (para $\Delta t \geq 0$), segundo

$$P(x, t + \Delta t) = \int P(x, t + \Delta t | x', t) P(x', t) dx'. \quad (2-7)$$

Para obter $\partial P(x, t) / \partial t$, precisamos conhecer as probabilidades de transição $P(x, t + \Delta t | x', t)$ para Δt pequeno. Supondo que conhecemos todos os momentos condicionados M_k , com $k \geq 1$, segundo

$$M_k(x', t, \Delta t) = \langle [X(t + \Delta t) - X(t)]^k \rangle_{|_{X(t)=x'}} = \int (x - x')^k P(x, t + \Delta t | x', t) dx, \quad (2-8)$$

existem diversas formas de obter uma expansão geral para a probabilidade de transição. Um dos métodos mais diretos é utilizar a expansão de Taylor para a função de densidade de probabilidade e para a probabilidade de transição.

Definindo $\Delta x = x - x'$, podemos expandir o integrando da Eq. (2-7) em série de Taylor:

$$\begin{aligned} P(x, t + \Delta t | x', t) P(x', t) &= P(x + \Delta x - \Delta x, t + \Delta t | x - \Delta x, t) P(x - \Delta x, t) \\ &= \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \Delta x^k \left(\frac{\partial}{\partial x} \right)^k P(x + \Delta x, t + \Delta t | x, t) P(x, t). \end{aligned} \quad (2-9)$$

¹No que segue, simplificaremos a notação por $f_1(x, t) \equiv P(x, t)$.

Inserindo a Eq. (2-9) na Eq. (2-7) e integrando com relação a Δx , obtemos diretamente:

$$P(x, t + \Delta t) - P(x, t) = \sum_{k=1}^{\infty} \left(-\frac{\partial}{\partial x} \right)^k \frac{M_k(x, t, \Delta t)}{k!} P(x, t). \quad (2-10)$$

Supondo que os momentos M_k podem ser expandidos em série de Taylor em relação a Δt ($k \geq 1$), ou seja,

$$\frac{M_k(x, t, \Delta t)}{k!} = D^{(k)}(x, t)\Delta t + O(\Delta t^2), \quad (2-11)$$

e levando em conta apenas os termos lineares em Δt , temos:

$$\frac{\partial P(x, t)}{\partial t} = \sum_{k=1}^{\infty} \left(-\frac{\partial}{\partial x} \right)^k D^{(k)}(x, t)P(x, t). \quad (2-12)$$

A Eq. (2-12) é conhecida como *expansão de Kramers-Moyal* (KM).

A probabilidade de transição $P(x, t|x', t')$ é a distribuição $P(x, t)$ para a condição inicial especial $P(x, t') = \delta(x - x')$. Assim, a probabilidade condicional também deve seguir Eq. (2-12).

2.1.4

Equação de Fokker-Planck

Se a expansão de KM é truncada após o segundo termo, a equação resultante é chamada de *equação de Fokker-Planck* (EFP). Ela se escreve de forma geral como:

$$\frac{\partial}{\partial t} P(x, t) = -\frac{\partial}{\partial y} (D^{(1)} P(x, t)) + \frac{1}{2} \frac{\partial^2}{\partial x^2} (D^{(2)} P(x, t)). \quad (2-13)$$

Neste formalismo $D^{(2)} > 0$ é chamado de coeficiente de difusão local e $D^{(1)}$ é um campo de força externo, conhecido como coeficiente de arraste ou *drift*. Esses coeficientes podem ou não depender do tempo. Matematicamente a Eq. (2-13) é um equação diferencial parcial de segunda ordem linear parabólica. Na literatura matemática, a Eq. (2-13) também é conhecida como *equação de Kolmogorov*.

Um importante teorema relacionado à EFP é o *teorema de Pawula* (20). De acordo com este teorema, se qualquer coeficiente $D^{(2s)} = 0$ para $s \geq 2$, todos os coeficientes $D^{(n)}$ com $n \geq 3$ devem ser zero, reduzindo naturalmente uma expansão de KM a uma EFP.

Os coeficientes de KM, $D^{(k)}(r, \tau)$, são definidos, a partir da Eq. (2-11) como:

$$D^{(k)}(r, \tau) = \lim_{\Delta \tau \rightarrow 0} \tilde{D}^{(k)}(r, \tau, \Delta \tau), \quad (2-14)$$

com

$$\tilde{D}^{(k)} = M^{(k)}(r, \tau, \Delta\tau) / \Delta\tau / k!, \quad (2-15)$$

sendo $M^{(k)}$ os momentos obtidos pela Eq. (2-8). Estes momentos pode ser computados numericamente a partir de uma série temporal, medida a intervalos Δt apropriados.

A equação de Fokker-Planck (EFP) (20) proporciona uma poderosa ferramenta para lidar com os problemas levantados até aqui e tem sido utilizada em muitos campos diferentes em ciências naturais, incluindo aplicações em física do estado sólido e de plasma, ótica quântica, cinética química e nuclear, biologia molecular e dinâmica populacional.

Entretanto, existem várias maneiras de se estudar a dinâmica de um sistema que evolui efetuando um processo estocástico. Podemos estudar a evolução das distribuições de probabilidade, via uma equação mestra ou algum tipo de equação, como a equação de Boltzmann ou a EFP. Podemos ainda estudar a evolução das variáveis mesoscópicas do sistema com o auxílio de funções aleatórias que descrevam o comportamento correto das variáveis microscópicas do sistema até, no máximo, algum momento determinado (em geral a variância), como veremos a seguir.

2.1.5 Equação de Langevin

Do ponto de vista intuitivo é mais confortável trabalhar com variáveis de estado do que com distribuições de probabilidade. Uma maneira direta de se levar em conta a estocasticidade de um sistema é introduzir as incertezas via termos aleatórios nas variáveis microscópicas ou mesoscópicas do sistema.

A equação de Langevin (EL) é uma equação diferencial estocástica, ou seja, uma equação diferencial alguns de cujos coeficientes são funções aleatórias do tempo t , com propriedades estocásticas bem definidas.

No tratamento de Langevin, tomamos como força macroscópica um termo resistivo $-\gamma x$ que suplementamos com um termo de força aleatório $\eta(t)$, obtendo-se

$$\dot{x} = -\gamma x + \sigma\eta(t). \quad (2-16)$$

Um bom modelo para a situação descrita pela Eq. (2-16) será:

$$- \langle \eta(t) \rangle = 0,$$

onde a média é tomada sobre um *ensemble* de sistemas.

$$- \langle \eta(t)\eta(t') \rangle = \Gamma\delta(t - t')$$

que significa que os choques que modificam a variável são descorrelacionados uns dos outros. Γ é uma constante.

Diversas generalizações da equação de Langevin, com outras formas particulares da força determinística e da intensidade do ruído, têm sido utilizadas desde faz muitos anos para modelar sistemas relacionados a movimentos aleatórios (movimentos brownianos e suas generalizações).

2.2

Dinâmica molecular

Nós usamos uma linguagem matemática para compreender, modelar e resolver problemas de interesse. Em geral quando queremos ter uma solução de um problema matemático temos que resolver algum tipo de equação. As equações diferenciais são as mais importantes porque descrevem muitos fenômenos físicos, mas nem toda equação diferencial admite uma solução analítica. Portanto, para aquelas equações que não conseguimos resolver analiticamente precisamos da ajuda de técnicas numéricas.

A dinâmica molecular é uma técnica que permite realizar simulações numéricas de um sistema físico dadas as condições iniciais (posições e momentos) e interações entre as partículas para determinar a sua evolução temporal.

Faremos primeiro uma rápida revisão dos métodos numéricos para equações diferenciais deterministas.

Uma equação diferencial de ordem m

$$\frac{d^m y}{dt^m} = f\left(t, y, \frac{dy}{dt}, \dots, \frac{d^{m-1}y}{dt^{m-1}}\right)$$

pode ser reduzida a m equações diferenciais ordinárias de primeira ordem definindo novas variáveis $y_1 = y$, $y_{j+1} = d^j y / dt^j$ para $j = 1, \dots, m-1$ e assim

$$\frac{dy_j}{dt} = y_{j+1},$$

$$\frac{dy_m}{dt} = f(t, y_1, y_2, \dots, y_m),$$

onde $j = 1, 2, \dots, m-1$. Portanto é suficiente revisar os métodos para a solução de equações diferenciais ordinárias de primeira ordem.

Os métodos de diferenças finitas se baseiam em discretizar os passos de tempo e calcular os novos valores iterativamente usando a aproximação em diferenças finitas da equação.

A ideia da maioria destes métodos é determinar a posição e momento num instante posterior $t + \delta t$, com uma boa convergência para a solução e estabilidade, iterativamente (21).

Cabe mencionar o método corretor-preditor e do de Runge-Kutta, basea-

dos em diferentes expansões em série de Taylor das equações de movimento discretizadas. Existem diversos esquemas, de diferentes ordens de precisão, tanto implícitos quanto explícitos.

2.2.1

Algoritmo leap-frog

O algoritmo denominado *leap-frog* (salto de sapo) tem suas origens no algoritmo proposto por Verlet, o qual é baseado em dois truncamentos de segunda ordem na expansão em série de Taylor da posição, na forma

$$r(t + \delta t) = r(t) + \delta t v(t) + \frac{\delta t^2}{2} a(t),$$

$$r(t - \delta t) = r(t) - \delta t v(t) + \frac{\delta t^2}{2} a(t).$$

E portanto temos

$$r(t + \delta t) = 2r(t) - r(t - \delta t) + \delta t^2 a(t),$$

$$v(t) = \frac{r(t + \delta t) - r(t - \delta t)}{2\delta t}.$$

Observa-se que a posição do próximo instante é calculada independentemente da velocidade, mas depende da posição, ao tempo t e num instante anterior ($r(t)$ e $r(t - \delta t)$), assim como também da aceleração. As vantagens de usar este método são: (i) as posições são calculadas como erros da ordem de δt^2 , (ii) apresenta reversibilidade temporal para forças conservativas, (iii) a nova posição é calculada numa iteração e (iv) a energia é conservada. Mas o fato de que um termo tão pequeno é adicionado a cada passo pode levar a tempos longos a imprecisões numéricas (problemas de convergência) (21). Porém, existe uma modificação do algoritmo de Verlet conhecida como do passo médio ou *leap-frog*, que consiste de uma desfasagem no tempo do cálculo das velocidades e posições de modo que as equações têm a forma

$$r(t + \delta t) = r(t) + v(t + \frac{1}{2}\delta t)\delta t,$$

$$v(t + \frac{1}{2}\delta t) = v(t - \frac{1}{2}\delta t) + a(t)\delta t.$$

Entretanto, no caso em que existem forças estocásticas de Langevin $-\gamma v + F(t)$, não existe uma adaptação simples deste algoritmo, sendo usado por exemplo o algoritmo de Ermak (21)

$$r(t + \delta t) = r(t) + c_1 v(t)\delta t + c_2 a(t)\delta t^2 + \delta r^G,$$

$$v(t + \delta t) = c_0 v(t) + c_1 a(t) \delta t + \delta v^G,$$

onde $c_0 = \exp(-\alpha \delta t)$, $c_1 = (1 - c_0)/(\alpha \delta t)$ e $c_2 = (1 - c_1)/(\alpha \delta t)$, sendo $\alpha = \gamma/m$ e onde δr^G e δv^G são variáveis aleatórias gaussianas com correlações apropriadas (21).

2.2.2

Integração numérica de equações diferenciais estocásticas

Para uma equação diferencial estocástica da forma

$$\frac{dx}{dt} = a(x, t) + b(x, t)\xi(t),$$

não existe uma adaptação simples dos métodos numéricos de equações diferenciais ordinárias (22) e isto é porque a idealização do ruído é não diferenciável. Portanto a solução

$$x(t) = x_0 + \int_{t_0}^t a(x(s))ds + \int_{t_0}^t b(x(s))dW(s),$$

onde $dW(t) \equiv \xi(t)dt$, tem dois termos integrais, o primeiro do tipo riemanniano e o segundo termo uma integral que pode ser calculada desde dois pontos de vista: Itô ou Stratonovich, do qual depende a escolha da forma de cálculo da integral estocástica segundo o valor atribuído a $b(x, t)$ (22).

Porém no nosso caso de ruído aditivo, o termo $b(x, t)$ é constante portanto o segundo termo de integração não depende do estado do sistema, sendo

$$x(t) = x_0 + \int_{t_0}^t a(x(s))ds + b \int_{t_0}^t dW(s),$$

o qual permite usar os métodos de integração numérica usuais e acrescentar o termo do ruído a cada passo de tempo.

Então a integração numérica de um processo estocástico que obedece a equação semelhante à de Langevin pode-se discretizar fazendo $t = n\tau$, de tal modo que a equação discretizada será

$$x_{n+1} = x_n + a(x)\tau + \sqrt{\tau\Gamma}\xi_n, \quad (2-17)$$

onde ξ_n , $i = 0, 1, \dots$ são variáveis aleatórias independentes de forma tal que $\langle \xi_n \rangle = 0$, $\langle (\xi_n - \langle \xi_n \rangle)^2 \rangle = 1$ e Γ é proporcional ao coeficiente de difusão (22).

2.2.3

Campo de forças

Para as simulações de dinâmica molecular consideramos o campo de forças do modelo de Lois (18), já definido na seção 1.2.6.

A forma do potencial Φ_{cc} , dado pela Eq. (1-2), encarregado de manter a cadeia unida, é mostrado na Fig. 2.1 para diferentes valores de q . Lembremos que este parâmetro determina a flexibilidade da cadeia.

Neste modelo a interação com o meio é simplificada mediante interações diferentes para monômeros hidrofóbicos e polares. O potencial Φ_{att} dado pela Eq. (1-3), tipo Lennard-Jones, regula a interação entre monômeros hidrofóbicos não ligados, sendo atrativo a distâncias moderadas e repulsivo a curtas distâncias. Enquanto Φ_{rep} , que regula a interação entre monômeros não ligados em que participa pelo menos um monômero hidrofílico, contém somente um termo repulsivo (volume excluído). A Fig. 2.2 mostra o perfil do potencial Φ_{att} para diferentes valores de E_c , que controla a profundidade do poço de potencial.

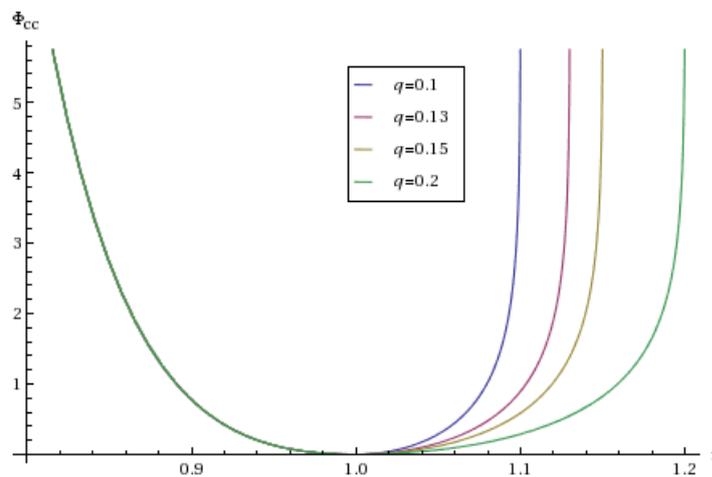


Figura 2.1: Gráfico do potencial (1-2) como função da distância para diferentes valores de q .

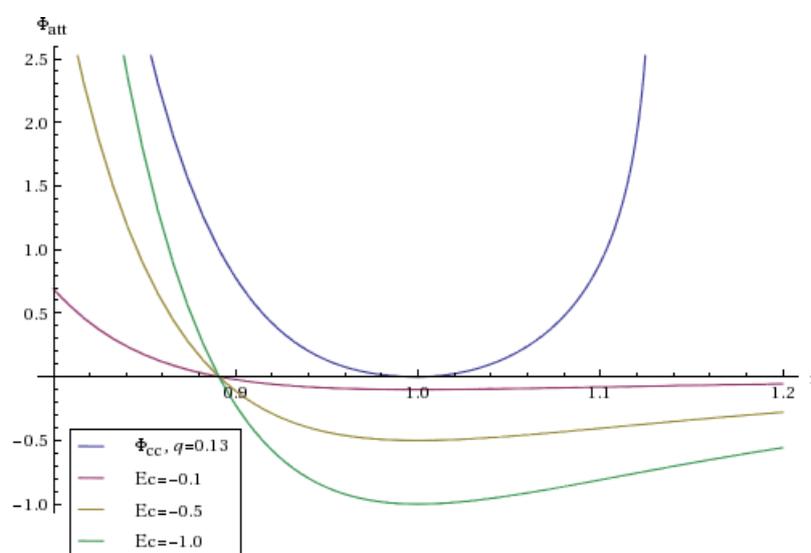


Figura 2.2: Gráfico do potencial da Eq. (1-3) entre monômeros hidrofóbicos para diferentes valores de energia de interação E_c .

2.2.4 Simplificação do termo inercial

Seguindo o modelo de Lois (18), realizamos simulações de dinâmica molecular integrando a equação de movimento

$$m_i \frac{d^2 \bar{r}_i}{dt^2} = \bar{F}_i(t) - \gamma \bar{v}_i - \frac{\partial}{\partial \bar{r}_i} \sum_{i \neq j} \Phi_{tot}(\bar{r}_{ij}), \quad (2-18)$$

onde

$$\Phi_{tot}(\bar{r}_{ij}) \equiv \Phi_{cc}(\bar{r}_{ij}) + \Phi_{att}(\bar{r}_{ij}) + \Phi_{rep}(\bar{r}_{ij})$$

e $\bar{F}_i(t)$ é a força aleatória (ruído branco gaussiano de média nula), em que as componentes são variáveis independentes.

O primeiro passo é obter uma equação independente das unidades do problema. Para isso escalamos a Eq. (2-18). Considerando as variáveis adimensionais $\tilde{r} = r/\sigma$ (onde r representa qualquer coordenada espacial) e $\tau = t/t_c$ (sendo t_c um tempo característico), então a Eq. (2-18) pode ser reescrita como

$$\zeta \frac{d^2 \tilde{r}_i}{d\tau^2} = f_i(\tau) - \alpha \frac{d\tilde{r}_i}{d\tau} - \frac{\partial}{\partial \tilde{r}_i} [\Phi_{tot}(\tilde{r}_i)],$$

onde $f_i = \frac{\sigma}{\zeta} F_i$ e t_c é escolhido de modo que $\alpha = 1$ e $\zeta = \frac{m_i \sigma^2}{\epsilon t_c^2}$.

Podemos considerar que a massa molar média dos aminoácidos é da ordem de ~ 100 g/mole, a energia das interações covalentes da ordem ~ 100 kcal/mole e σ da ordem de $\sim 10^{-9}$ m. Logo, para que o valor de ζ seja desprezível, $t_c \gg 1$ ps. Isso significa que, dentro de escalas de tempo maiores que 10^{-12} s, que é uma condição razoável, é válido desprezar o termo inercial. Então, renomeando as novas variáveis de espaço e tempo adimensionais com os nomes iniciais, a Eq. (2-18) pode ser escrita da forma:

$$\frac{dr_i}{dt} = f_i(t) - \frac{\partial}{\partial r_i} \sum_{i \neq j} \Phi_{tot}(\bar{r}_{ij}). \quad (2-19)$$

Esta equação tem a forma de uma equação de Langevin multi-dimensional e pode ser integrada numericamente para obter as trajetórias da cadeia.

Devemos notar que a exclusão do termo inercial leva a equação de segundo ordem nas coordenadas espaciais a uma de primeira ordem. Porém, a descrição do problema continua sendo válida porque a influência do termo inercial afeta só os primeiros instantes ou *transiente* (dentro da escala do tempo característico)². Neste caso, a equação é do tipo da equação de Langevin. Portanto, pode ser integrada numericamente, segundo a Eq. (2-17).

²Para mais detalhes recomendo ao leitor o capítulo 3 da referência (23).

Com isso podemos obter a dinâmica molecular para nossos cálculos.

2.2.5

Parâmetros das simulações

Na Fig. 2.3 são apresentados os efeitos da temperatura sobre a coordenada de reação Z (distância entre extremidades) para a sequência 1000100010001, onde 1(0) corresponde a monômero hidrofóbico (hidrofílico). Pode-se observar que, para temperaturas maiores a $T > T_r = 0,035$, a cadeia consegue esticar-se quase-completamente, mas não estamos interessados em configurações nas quais a cadeia fica desenovelada, por tanto usamos temperaturas menores do que esse valor.

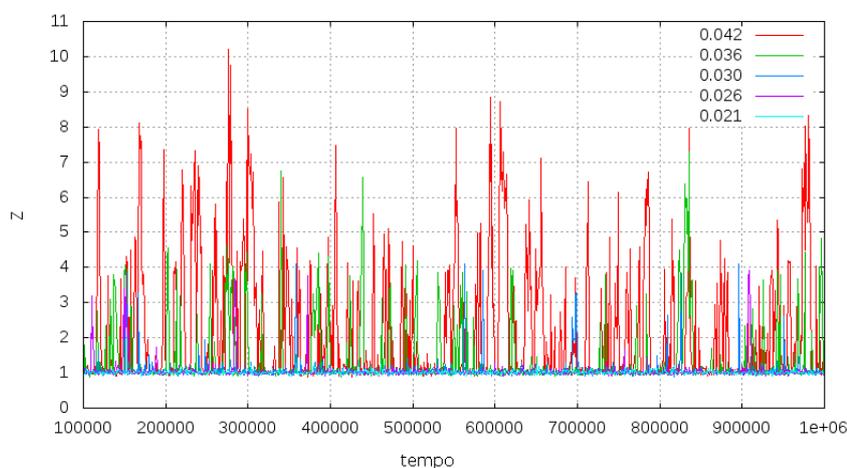


Figura 2.3: Evolução temporal da variável Z (distância entre extremidades) para a sequência 1000100010001, onde 1(0) corresponde a monômero hidrofóbico (hidrofílico), para diferentes valores da temperatura indicados na figura.

Além do mais, para altas temperaturas a cadeia pode se romper e é por essa razão que realizamos uma análise do efeito do ruído sobre o potencial (1-2) que mantém unida a cadeia para evitar o rompimento da mesma, assim como para escolher um valor ótimo do dt (ver Fig. 2.4). Esta análise determinou a escolha dos parâmetros da simulação $dt = 10^{-3}$ e $0,01 < T < 0,1$.

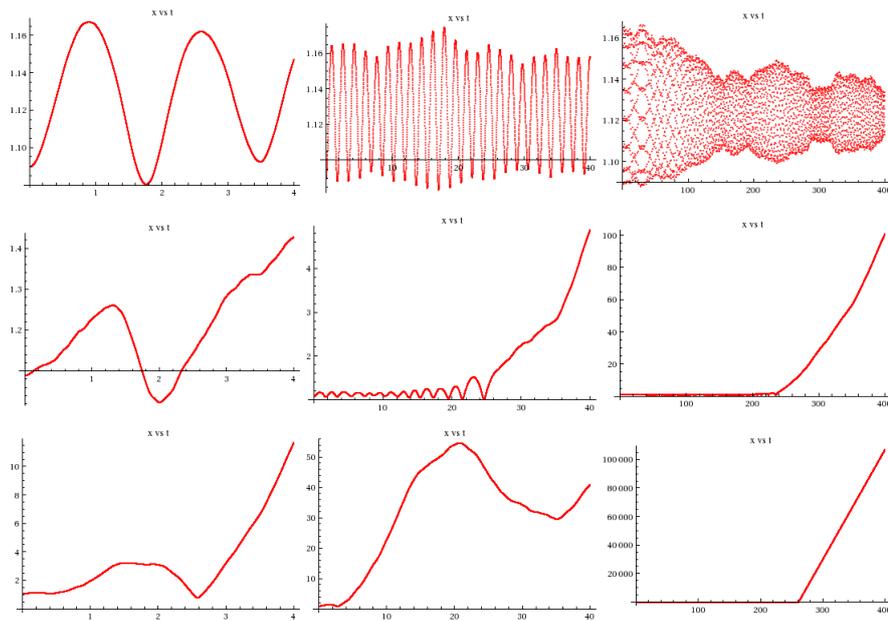


Figura 2.4: Evolução temporal de X (distância entre dois monômeros consecutivos, representativos, 1 e 2 no caso da figura) como função do tempo, para diferentes valores do passo de integração dt e da temperatura T . De esquerda a direita $dt = 0,001$; $0,01$ e $0,1$ e de cima para baixo $T = 0,01$; $0,1$ e $1,0$. Grandezas adimensionais segundo será explicado no texto.

3

Sequências estudadas

3.1

Sequências binárias

Discutimos na Seção 1.2 como a grande complexidade das sequências peptídicas pode ser parcialmente simplificada levando-se em consideração as principais propriedades físico-químicas dos aminoácidos e de suas interações.

Considerando, como é de consenso, que a influência mais importante que governa o enovelamento é o efeito hidrofóbico, devemos pelo menos caracterizar os monômeros por sua afinidade pelo meio aquoso. Assim, estudaremos a dinâmica de sequências binárias de N monômeros (que denotaremos $h_1 \dots h_N$, onde $h_i = 1$ se o monômero i for hidrofóbico ou $i = 0$ se for polar. Este tipo de sequências é apropriado para o modelo que utilizaremos, que é o modelo de Lois descrito na Seção 1.2.6.

Mesmo com essa simplificação, o número de cadeias possíveis é enormemente grande. Portanto, devemos selecionar algumas delas para o nosso estudo. Primeiramente, fixamos o comprimento N e variamos a proporção h de monômeros hidrofóbicos. Escolhemos $N = 13$, porque representa uma sequência curta (que permite simulações em tempos viáveis) mas que contudo permite certa variabilidade nos possíveis motivos ou padrões da sequência. Com relação ao número total de unidades hidrofóbicas, tipicamente consideramos $H = hN = 4$. Esta escolha é justificada porque essa proporção corresponde aproximadamente à observada em proteínas solúveis reais (para as que $0.3 \leq h \leq 0.5$). Com efeito, se a fração for muito pequena, teríamos uma serpentina aleatória (*random coil*), se for muito grande, os resíduos polares seriam insuficientes para criar um envoltório que evite a exposição dos hidrofóbicos ao meio aquoso (24).

Além do mais, esses foram os valores considerados por Lois para o estudo em 2D (18), permitindo assim comparações com resultados anteriores.

Com relação à sequência, dado que pelo menos 60% das proteínas reais têm uma distribuição aleatória dos resíduos hidrofóbicos sobre a cadeia (25),

em princípio, poderíamos considerar sequências aleatórias na nossa análise. Para $N = 13$ e $H = 4$, teríamos um total de 365 possíveis combinações diferentes¹

Somente 15 dessas sequências apresentam simetria central. Todas elas foram usadas para nosso estudo. Estas sequências simétricas foram identificadas com o índice S que mede a proximidade dos monômeros hidrofóbicos aos extremos da cadeia, de modo análogo a um momento de inércia,

$$S = \sum_{i=1}^6 i^2 h_i, \quad (3-1)$$

onde i vale zero para o resíduo central e aumenta em uma unidade a cada posição em direção à extremidade, sendo computado sobre a metade da sequência. Portanto, as sequências com extremos hidrofóbicos terão o índice maior, para o caso considerado, em que não existe degenerescência (ver tabela 3.1).

sequência	S
1100000000011	61
1010000000101	52
1001000001001	45
1000100010001	40
1000010100001	37
0110000000110	41
0101000001010	34
0100100010010	29
0100010100010	26
0011000001100	25
0010100010100	20
0010010100100	17
0001100011000	13
0001010101000	10
0000110110000	5

Tabela 3.1: Nesta tabela mostramos, os valores do índice S dado pela equação Eq. (3-1), para as sequências simétricas com $N = 13$ e $H = 4$.

¹Resultado de somar as combinações simétricas ($6!/(2! \times 4!) = 15$) mais metade das assimétricas, sendo estas últimas obtidas considerando o número total de combinações ($13!/(4! \times 9!) = 715$) menos as simétricas (15).

3.2

Variáveis efetivas

O espaço de fases é multidimensional, porém o envelamento pode em princípio ser descrito em termos de um pequeno número de variáveis macroscópicas, que definem o espaço de conformações. Estamos interessados em achar variáveis efetivas que permitam essa descrição. Este não é um problema trivial, nem sua solução geral, porque algumas variáveis podem ser apropriadas para determinados tipos de sequências e para as estruturas em que elas podem se organizar mas não para outras (26). Aqui consideraremos as variáveis usuais, que definimos a continuação.

A coordenada de reação Z é a distância de um extremo da cadeia ao outro

$$Z = |\vec{r}_N - \vec{r}_1|.$$

Nóte-se que Z corresponde a $|\mathbf{R}|$, onde \mathbf{R} foi definido na seção 1.2.1.

O raio de giro R_g mede o desvio quadrático médio com relação ao centro de massa da cadeia (aqui consideramos um valor único, unitário, para a massa).

$$R_g = \frac{\sqrt{\sum_{i=1}^N (\vec{r}_i - \vec{r}_{cm})^2}}{N},$$

sendo

$$\vec{r}_{cm} = \frac{\sum_{i=1}^N \vec{r}_i}{N},$$

onde \vec{r}_i é o vetor posição do monômero i , \vec{r}_{cm} a posição do centro de massa e N o número total de monômeros.

Adicionalmente definimos R_g^H e R_g^P como:

$$R_g^H = \frac{\sqrt{\sum_{i=1}^N (\vec{r}_i - \vec{r}_{cm})^2 h_i}}{N},$$

$$R_g^P = \frac{\sqrt{\sum_{i=1}^N (\vec{r}_i - \vec{r}_{cm})^2 (\bar{h}_i)}}{N},$$

que representam o raio de giro considerando só os monômeros hidrofóbicos (H) e só os polares (P), respectivamente, sendo \bar{h}_i operador de *negação* sobre o índice de hidrofobicidade h_i (ou seja $\bar{h}_i = 1 - h_i$, por tanto $\bar{h}_i = 0$ se o resíduo for hidrofóbico e 1 se for polar).

Também foram definidos os números de contatos n_{11} , n_{10} e n_{00} como o número de contatos entre os monômeros dos tipos correspondentes, assim

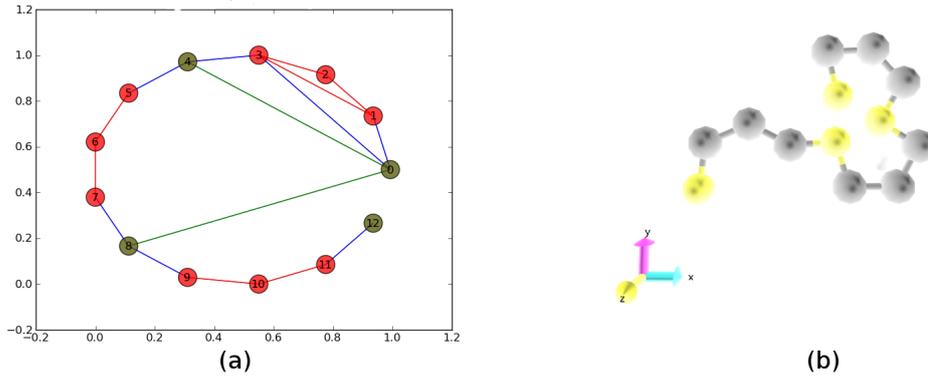


Figura 3.1: (a) Conformação da cadeia na rede de contatos, as linhas verdes representam contatos tipo 11, as azuis 10 e as vermelhas 00. (b) Conformação espacial.

como também o número total de contatos n_t . Para isso consideramos a matriz de contatos C_{ij} , onde os contatos são definidos como

$$C_{ij} = \begin{cases} 1, & \text{se } r_{ij} \geq d_c \sigma, \\ 0, & \text{se } r_{ij} > d_c \sigma, \end{cases}$$

onde d_c é um valor de corte (em nosso caso $d_c = 1, 2$) e r_{ij} é a distância entre os monômeros i e j . Então a matriz \mathbf{C} será uma matriz quadrada e simétrica de ordem N . Usando esta matriz calculamos para cada instante de tempo os valores

$$n_{11} = \sum_{i=1}^N \sum_{j>i}^N C_{ij} h_i h_j,$$

$$n_{10} = \sum_{i=1}^N \sum_{j>i}^N C_{ij} h_i \bar{h}_j,$$

$$n_{00} = \sum_{i=1}^N \sum_{j>i}^N C_{ij} \bar{h}_i \bar{h}_j,$$

$$n_t = \sum_{i=1}^N \sum_{j>i}^N C_{ij}.$$

Como ilustração mostramos na Fig. 3.1(a) a representação dos contatos entre monômeros não ligados (linhas de cor). As bolinhas verdes representam os 1s e as vermelhas os 0s, as linhas de cor verde representam os contatos entre os 1s, as vermelhas entre os 0s e as azuis contatos 10. Portanto para esta configuração teríamos $n_{11}=2$, $n_{10}=1$ e $n_{00}=1$ (estas variáveis efetivas têm valores inteiros).

Estas variáveis efetivas foram usadas em nossas análises da dinâmica das sequências.

4 Resultados

Neste capítulo apresentamos os resultados da dinâmica molecular das sequências simplificadas apresentadas no capítulo anterior.

Para as simulações de dinâmica molecular usamos o campo de forças de Lois (18). Consideramos diferentes temperaturas dentro de uma faixa apropriada para obter um mapeamento do perfil de energia visitado pelo sistema.

Abordamos inicialmente o problema em duas dimensões, como primeira tentativa mais simples.

Na Fig. 4.1 apresentamos três possíveis conformações da sequência

1000100010001

no espaço 2D. À esquerda (a) observamos um estado desenovelado ou desnaturado, à direita (c) um estado compacto que poderia ser considerado o estado nativo da sequência, que como veremos corresponde à mínima energia e no centro (b) um estado com grau de organização intermediário (mal-enovelado), correspondendo a um mínimo local de energia.

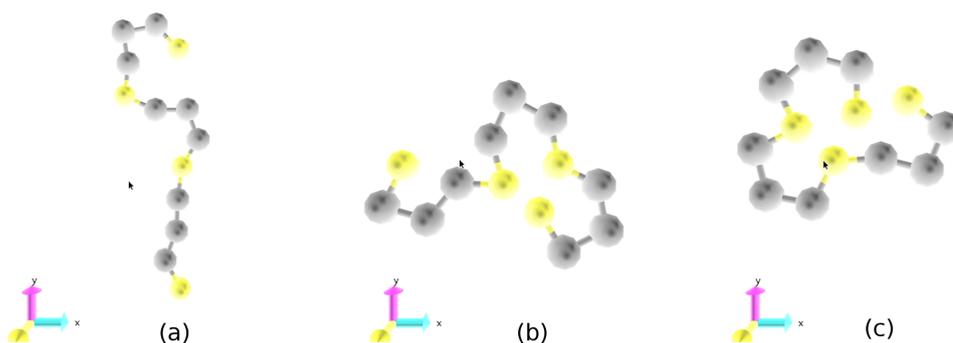


Figura 4.1: Estados conformacionais de uma sequência com $N = 13$ e $H = 4$: (a) desnaturado, (b) mal-enovelado e (c) nativo. As bolinhas amarelas representam os monômeros hidrofóbicos(1s) e as de cor cinza representam os polares (0s).

A implementação do algoritmo para a dinâmica molecular foi feita em C++ (usando o código apresentado no apêndice B¹

Como condição inicial da dinâmica, cada sequência foi preparada no estado linear da cadeia, com distâncias unitárias² e velocidades nulas. As coordenadas de posição durante as simulações foram armazenadas em formato de arquivo PDB³, tanto para visualizar a dinâmica quanto para analisar posteriormente os dados. Também foram salvos os valores da energia potencial (E_p) e as variáveis efetivas (ver seção 3.2).

4.1

Séries temporais

Nesta seção vamos apresentar os resultados das séries temporais das variáveis da seção 3.2, para as sequências consideradas na Tabela 3.1.

Na Fig. 4.2, observamos a evolução temporal das variáveis efetivas Z , R_g , R_g^P e R_g^H (a); e n_{11} , n_{10} , n_{00} e n_t (b), para a sequência 1000100010001. Podemos apreciar que as variáveis efetivas em (a), relativas a distâncias características da estrutura, têm evoluções temporais similares. Para os números de contatos pode-se observar um comportamento com amplas flutuações, isto pode ser causado pela escolha do valor de corte ($d_c = 1, 2\sigma$). A Fig. 4.3, mostra a rede de contatos para diversas escolhas de d_c , da ordem do alcance da interação atrativa entre 1s (Fig. 2.2). A partir da observação destas redes, escolhemos um valor intermediário de d_c em que a rede não seja nem muito rarefeita nem muito densa.

Os quadros da Fig. 4.4 mostram uma realização da série temporal da variável Z para cada uma das 15 sequências analisadas (ver Tabela 3.1), à temperatura $T = 0.026$. Destes gráficos pode-se notar que à medida que o S aumenta as flutuações de Z diminuem. Isto é coerente devido a que resíduos hidrofóbicos mais perto dos extremos, estabilizam a distância Z entre extremidades. Entretanto, a sequência 1000010100001 ($S = 37$) apresenta saltos entre os estados $Z = 1$ e $Z = 4$, devido à centralidade de dois dos monômeros hidrofóbicos.

Observou-se um comportamento bem estável para sequências com extremos hidrofóbicos com poucas mudanças de estado, enquanto as sequências com extremos polares visitam mais conformações (segundo a variável efetiva Z).

¹A versão mais atual pode ser descarregada da página <http://ceor.googlecode.com/svn/trunk/MD>.

²Lembremos que a distância de equilíbrio entre monômeros foi usada para adimensionar as distâncias.

³Das siglas em inglês de *Protein Data Bank*.

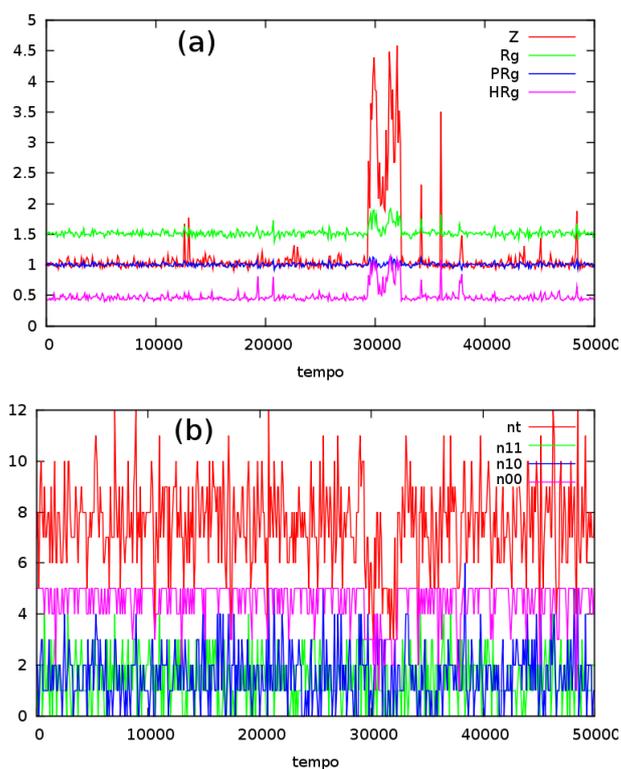


Figura 4.2: Evolução temporal das variáveis efetivas (a) Z , R_g , R_g^P e R_g^H , que representam distâncias características da estrutura, e (b) n_{11} , n_{10} , n_{00} e n_t que representam os números de contatos entre os diversos monômeros para uma distância de corte $d_c = 1, 2\sigma$ para a sequência 1000100010001.

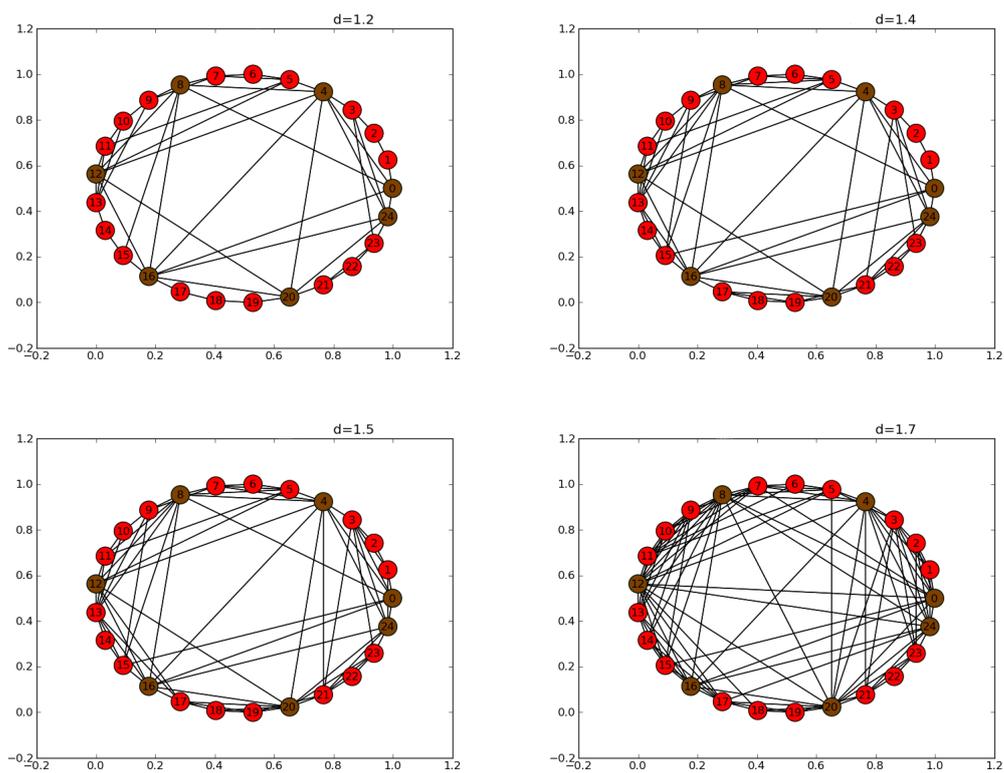


Figura 4.3: Diagrama de contatos para diferentes valores de d_c (corte), com $\sigma = 1$.

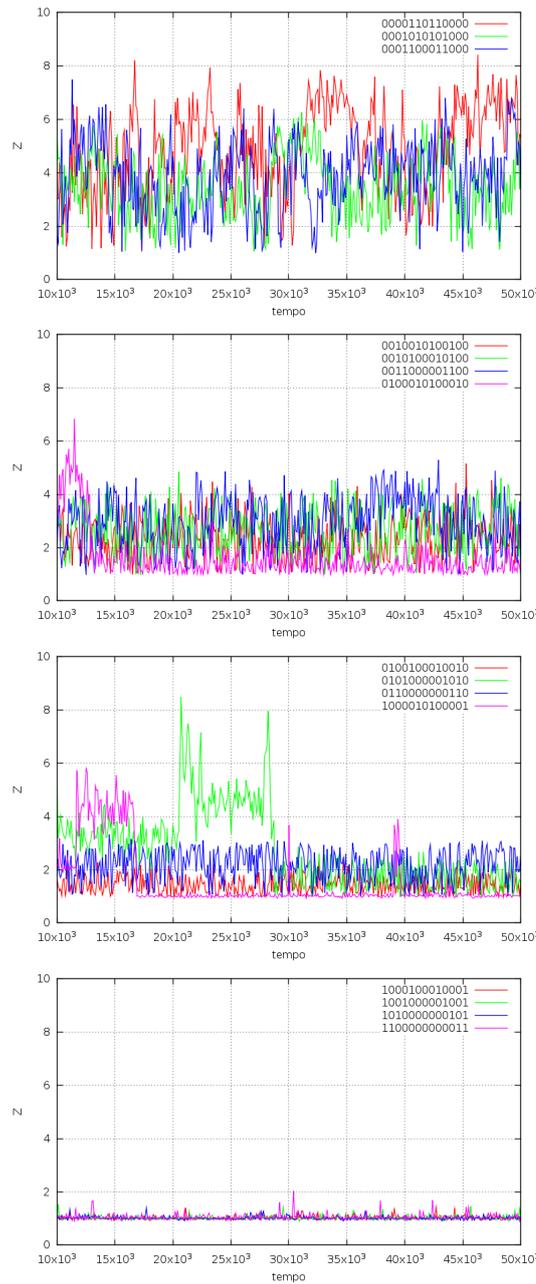


Figura 4.4: Evolução temporal da variável Z para as 15 seqüências simétricas com $N = 13$, a $T = 0,026$.

4.2 Momentos e correlações

Na Fig. 4.5, mostramos a média e o desvio padrão de cada variável efetiva R_g , Z , R_g^H e R_g^P para as seqüências caracterizadas por S na Tabela 4.1. Deve-se notar que a variável que mais muda com o índice da seqüência é a Z e a que tem um maior desvio padrão é R_g^H . Este decresce à medida que os monômeros hidrofóbicos se aproximam dos extremos (ou seja, S aumenta).

Por outro lado, na Fig. 4.6 se apresentam a média e desvio padrão das variáveis n_{11} , n_{10} , n_{00} , n_t para as seqüências analisadas. Notamos que não há dependência notável da média nem do desvio padrão com o índice S .

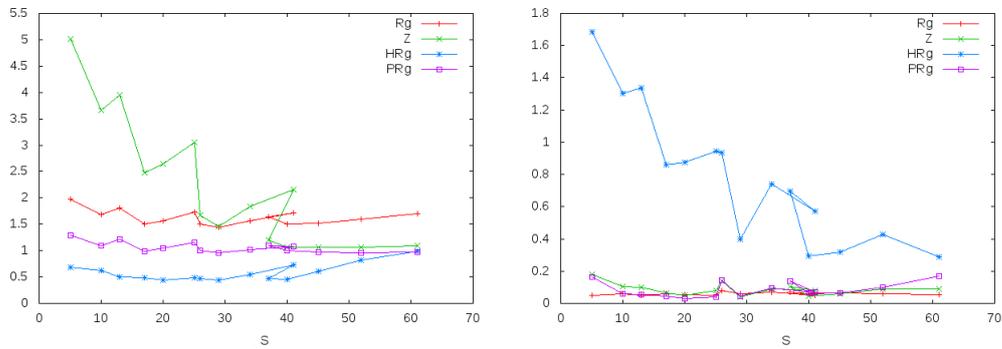


Figura 4.5: Média (esquerda) e desvio padrão (direita) das variáveis efetivas R_g , Z , R_g^H e R_g^P .

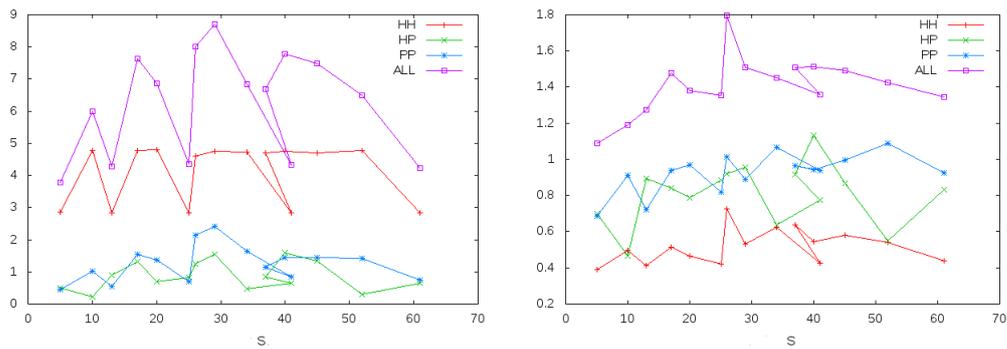


Figura 4.6: Média (esquerda) e desvio padrão (direita) das variáveis efetivas n_{11} , n_{10} , n_{00} , n_t .

Apresentaremos agora os cálculos da função de autocorrelação linear da variável Z .

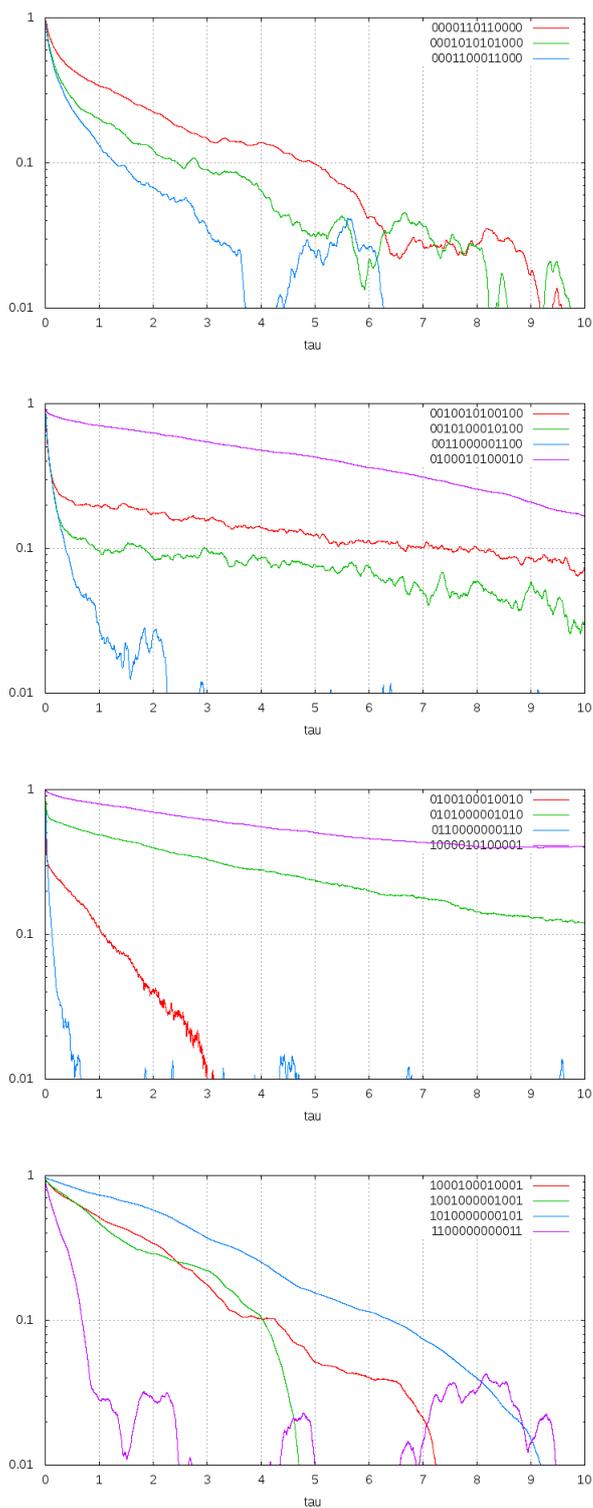


Figura 4.7: Autocorrelação de Z para as 15 seqüências de interesse.

sequências	S	A	$\tau_1(Z)$	$\tau_2(Z)$
1100000000011	61	0,00	-	0,31
1010000000101	52	0,00	-	3,00
1001000001001	45	0,24	0,28	2,09
1000100010001	40	0,33	1,59	16,4
1000010100001	37	0,17	0,030	7,00
0110000000110	41	0,72	0,019	0,13
0101000001010	34	0,42	0,001	5,64
0100100010010	29	0,69	0,006	0,97
0100010100010	26	0,17	0,030	7,00
0011000001100	25	0,72	0,051	0,41
0010100010100	20	0,88	0,066	8,86
0010010100100	17	0,78	0,047	8,80
0001100011000	13	0,70	0,122	1,38
0001010101000	10	0,70	0,132	2,50
0000110110000	5	0,51	0,141	2,70

Tabela 4.1: Valores dos parâmetros da função de ajuste da autocorrelação de Z , $C(t) = A \exp(-t/\tau_1) + (1 - A) \exp(-t/\tau_2)$, para as sequências simétricas com $N = 13$ e $H = 4$, ordenadas segundo o índice S .

Nos painéis da Fig. 4.7, observamos os gráficos da autocorrelação para cada sequência em escala log-linear. Em cada caso foi ajustada uma função exponencial da forma $A \exp(-t/\tau_1) + (1 - A) \exp(-t/\tau_2)$, sendo assim obtidos os tempos de correlação mostrados na Tabela 4.1.

Notamos que nos casos de extremidades polares (0s) existem duas escalas de tempo características bem separadas sendo uma delas muito curta ($\tau_1 \ll 1$) e a correlação cai a um valor pequeno dentro nesse tempo, indicando uma dinâmica descorrelacionada. Entretanto, nos casos de extremidades hidrofóbicas (1s) predomina uma única exponencial, com um tempo característico longo comparado com o τ_1 do caso anterior.

4.3 Perfis de energia

A Fig. 4.8 ilustra o perfil de energia no plano $Z - R_g$ para a sequência 1000100010001, construído a partir de 50 realizações. Deste gráfico podemos observar que existem três poços de energia. Gostaria de chamar a atenção sobre o fato de que é possível identificar esses poços com diferentes valores da variável Z mas não com a variável R_g que apresenta valores muito próximos para todos eles. Isto sugere que seja Z uma variável efetiva mais apropriada.

Diferentemente, para sequências em que os resíduos hidrofóbicos são mais centrais, observamos uma perfil sem múltiplos mínimos, como no exemplo

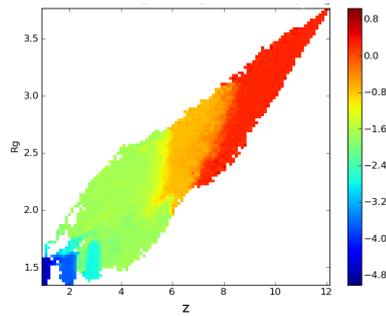


Figura 4.8: Perfil de energia para a sequência 1000100010001.

ilustrativo da Fig. 4.9 para a cadeia 0001010101000. Neste caso a estrutura muda de modo aleatório, não sendo possível identificar estados estruturais em que a cadeia reside preferencialmente. Isto é consistente com os menores tempos de correlação observados para este tipo de sequências.

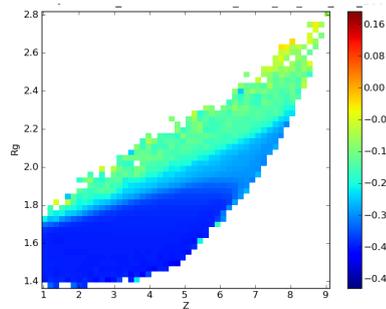


Figura 4.9: Perfil de energia para a sequência 0001010101000.

Os perfis de energia para as demais sequências são apresentados no apêndice A.

4.4 Coeficientes de arraste e difusão

Calculamos os dois primeiros coeficientes de Kramers-Moyal para a variável Z , a partir das séries temporais dessa variável, usando a técnica descrita na seção 2.1.3. Os resultados são apresentados nos gráficos das Figs. 4.10 e 4.11, agrupados, respectivamente, em sequências com extremos hidrofóbicos e hidrofílicos.

Nas Figs. 4.12 e 4.13 apresentamos a forma do potencial efetivo para dois grupos de painéis (o primeiro para sequências com extremos hidrofóbicos), obtido por integração numérica do oposto do coeficiente de arraste D_1 , ou seja $-\int^Z D_1(Z')dZ'$.

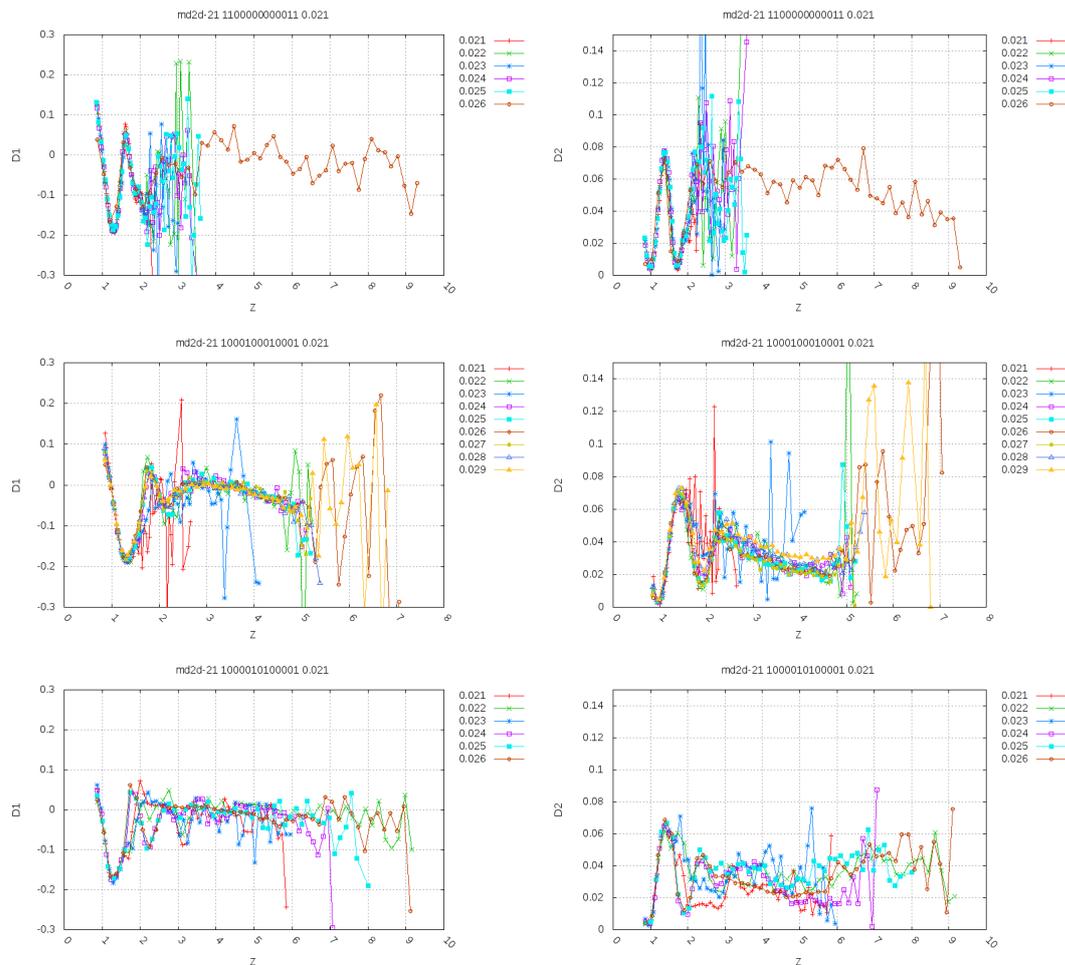


Figura 4.10: Coeficientes de Kramers-Moyal D_1 (arraste) e D_2 (difusão), para temperaturas de $T = 0,021$ até $T = 0,026$.

Na Fig. 4.14 resumimos a posição dos mínimos como função do S . Podemos notar que para as sequências com extremos hidrofóbicos existem poços de potencial bem definidos (Fig. 4.10), mas quando os extremos são polares, os poços desaparecem (Fig. 4.11). Isto mostra que D_1 é fortemente influenciado pelo índice S , o qual é claro intuitivamente porque estamos usando como variável efetiva a distância entre extremidades.

No caso dos coeficientes de difusão, notamos uma diferença para os extremos 1 ou 0. Além do mais, parece ter também uma influência do segundo monômero hidrofóbico, à medida que ele se desloca para o centro, o poço de potencial fica melhor definido.

Observamos que para todas as sequências, o coeficiente de difusão D_2 não é constante mas depende da variável de estado Z .

Para as sequências com terminais 1, o coeficiente D_2 varia mais abruptamente, numa faixa de 0 a 0,08 (Fig. 4.10). D_2 tem um comportamento oscilante.

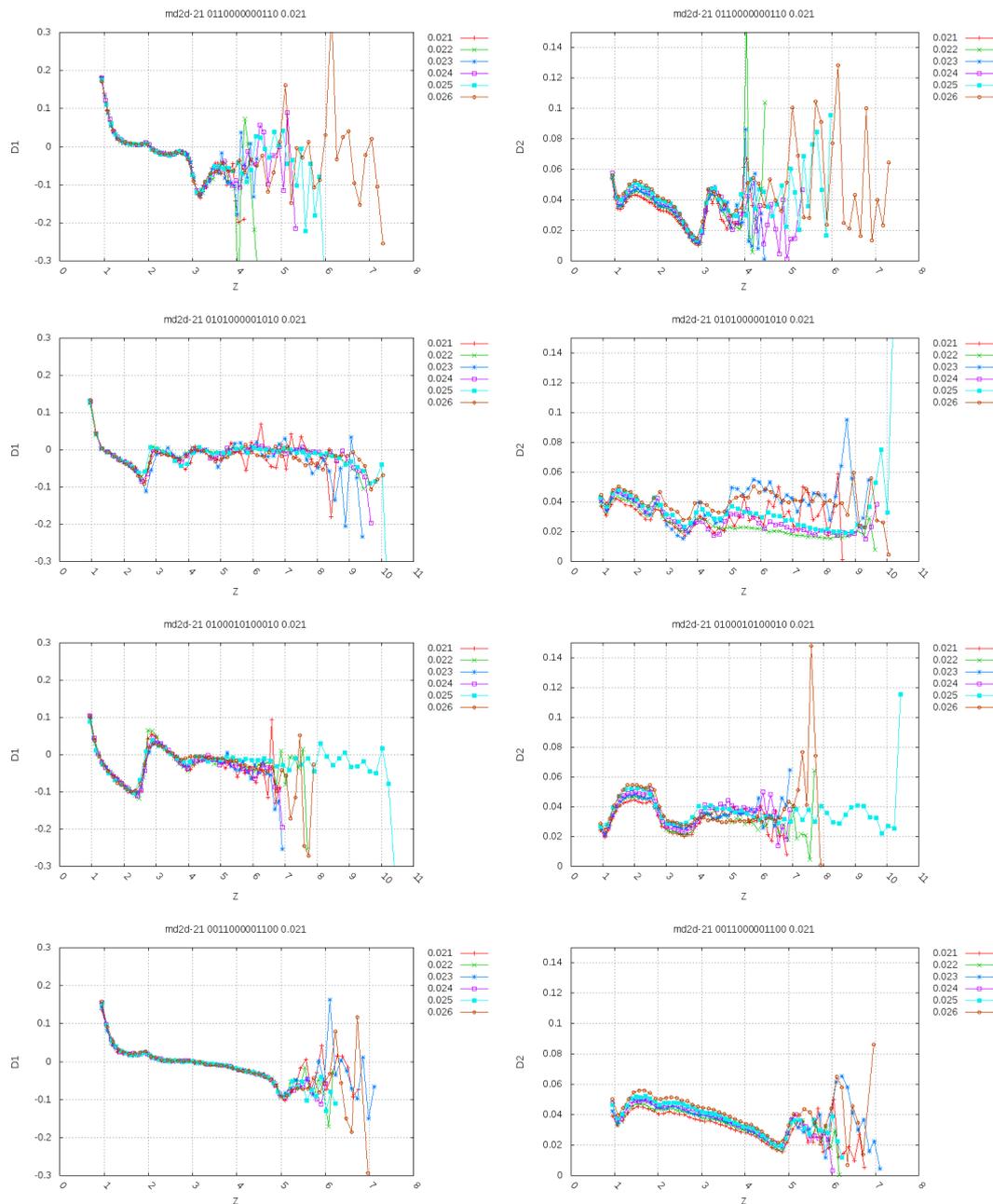


Figura 4.11: Coeficientes de Kramers-Moyal D_1 e D_2 , para temperaturas de $T = 0,021$ até $T = 0,026$.

Os vários mínimos que ocorrem estão em correspondência com os mínimos do potencial de D_1 (Fig. 4.12).

Perto do estado compacto, pequenas mudanças de Z provocam grandes mudanças de D_2 . Para o estado desenovelado as variações de D_2 são mais suaves.

Para as seqüências mais desorganizados (com terminais 0) coeficiente varia tipicamente entre 0,04 e 0,06, de forma suave (Fig. 4.11). Um mínimo ocorre para $Z \simeq 1,1$ associado a um mínimo do potencial de D_1 (estado mais

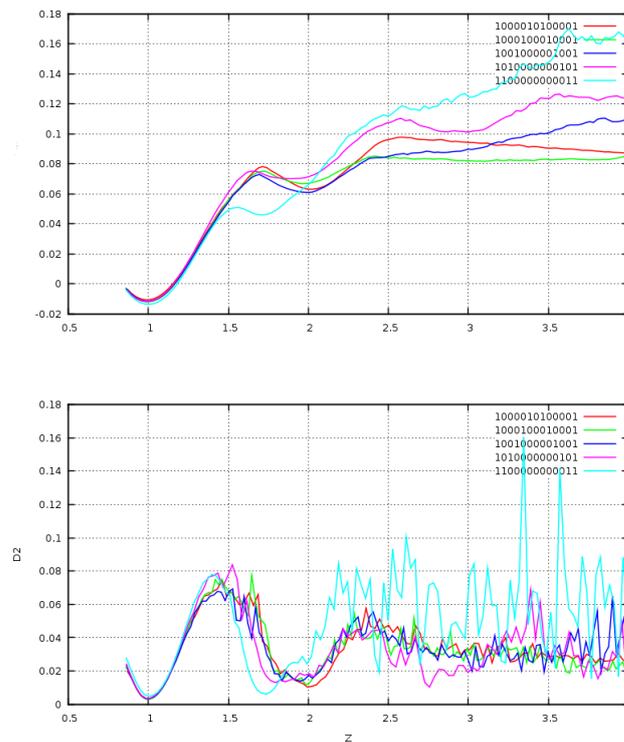


Figura 4.12: (Acima) Potencial efetivo para a variável Z , calculado por integração numérica de $-D_1(Z)$. (Abaixo) Coeficiente de difusão D_2 para comparação. Para todas as seqüências com extremos hidrofóbicos.

compacto), ver Fig. 4.13 .

Para valores grandes de Z , D_2 é aproximadamente 0,03 para ambos os tipos de seqüências.

Em soma, observamos que o coeficiente D_2 é mais uniforme para as seqüências menos enoveláveis. Para as de terminação 1, as variações de D_2 tem maior amplitude. Estas seqüências visitam mais estruturas, e as variações de D_2 refletem que este coeficiente é diferente para estados compactos que para estados desordenados. Além do mais parece existir uma relação entre D_2 e D_1 . A análise dessa relação pode permitir separar os efeitos que são puramente difusivos dos associados à curvatura do potencial.

O conhecimento dos coeficientes D_1 e D_2 permitiria, em princípio, escrever uma equação de Fokker-Planck ou uma equação diferencial estocástica (tipo de Langevin) para a variável efetiva Z . Entretanto, para isso, deveria ser desprezível o coeficiente D_4 legitimando (pelo teorema de Pawula) o truncamento da expansão Kramers-Moyal a segunda ordem. Na Fig. 4.15 ilustramos o comportamento do quarto coeficiente, mostrando que, efetivamente, é pequeno comparado com D_2 .

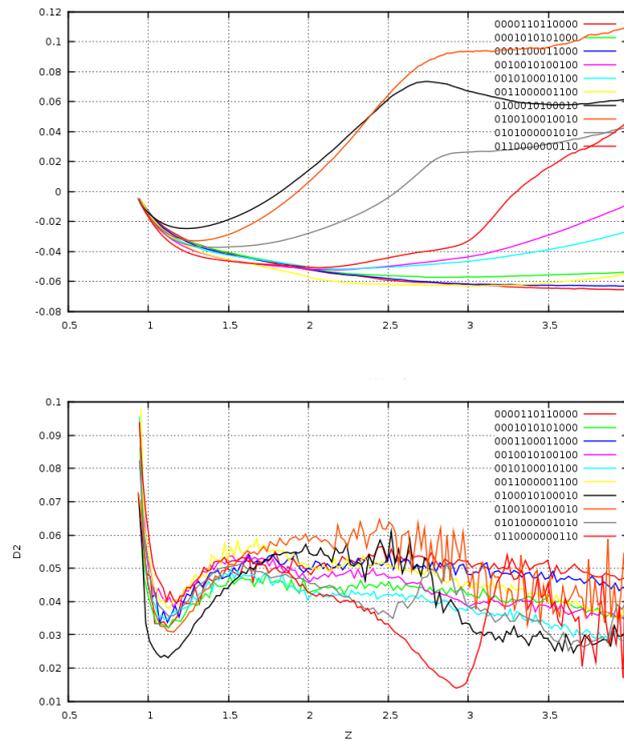


Figura 4.13: (Acima) Potencial efetivo para a variável Z , calculado por integração numérica de $-D_1(Z)$. (Abaixo) Coeficiente de difusão D_2 para comparação. Para todas as seqüências com extremos polares.

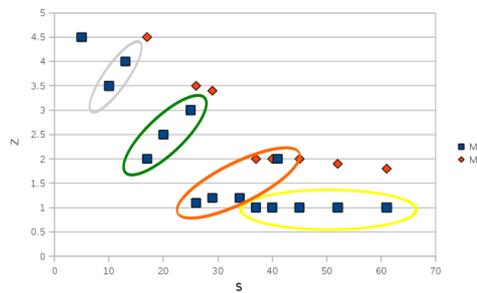


Figura 4.14: Mínimos do potencial efetivo para as seqüências.

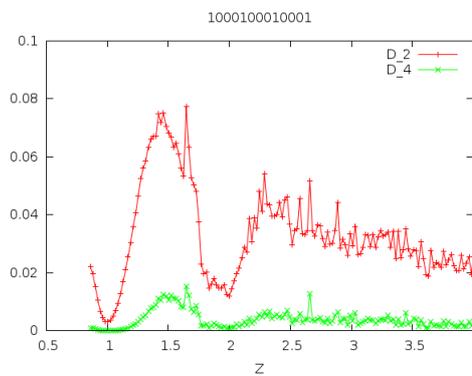


Figura 4.15: Comparação entre os coeficientes D_2 e D_4 da seqüência 1000100010001.

sequências	S	min_1	min_2
1100000000011	61	1	1,8
1010000000101	52	1	1,9
1001000001001	45	1	2
1000100010001	40	1	2
1000010100001	37	1	2
0110000000110	41	2	-
0101000001010	34	1,2	-
0100100010010	29	1,2	3,4
0100010100010	26	1,1	3,5
0011000001100	25	3	-
0010100010100	20	2,5	-
0010010100100	17	2	4,5
0001100011000	13	4	-
0001010101000	10	3,5	-
0000110110000	5	4,5	-

Tabela 4.2: Valores do índice S dado pela equação Eq. (3-1), para as sequências simétricas com $N = 13$ e $H = 4$. Os valores min_1 e min_2 representam as posições dos mínimos de potencial das Figs. 4.12 e 4.13.

5

Conclusões e perspectivas

O processo de enovelamento pode ser modelado mediante a difusão numa paisagem de energia de baixa dimensão (27). Neste trabalho, como nos trabalhos (28), (19) e (26), procuramos obter uma dinâmica efetiva para o problema de enovelamento, apesar de que em cada caso diferentes modelos para a dinâmica microscópica foram considerados. No nosso caso, escolhemos o modelo proposto por Lois como ponto de partida para obter, da dinâmica microscópica, as séries temporais das variáveis efetivas. Em particular, investigamos como essa dinâmica efetiva é influenciada pela distribuição de monômeros hidrofóbicos dentro da cadeia.

Uma questão importante que não foi discutida ainda é a da dimensionalidade do problema. A maior parte dos resultados foram obtidos para dinâmicas confinadas no plano. Esta redução do problema, permite encontrar mudanças conformacionais para sequências de comprimentos menores que os que seriam necessários em 3D. Notemos que estruturas estáveis em 2D, perderiam estabilidade ao agregar uma dimensão espacial. Por exemplo, a estrutura de menor energia para a sequência 1000100010001, mostrada na Fig. 4.1(c) é muito estável em 2D, entretanto no espaço 3D poderia sofrer deformações locais sem mudança energética significativa (por exemplo “folhas” “do trevo” flutuando, enquanto os monômeros hidrofóbicos permaneçam nas mesmas posições relativas). Também é importante salientar que algumas estruturas, por exemplo envolvendo nós, dependem da dimensionalidade crucialmente.

A simplificação 2D pode constituir uma limitação pouco realista na maioria dos casos, dado que a natureza das proteínas é tridimensional, exceto para algumas subestruturas como as folhas beta. Contudo, fica justificada desde a perspectiva do presente trabalho de, em primeiro lugar, aprender as técnicas envolvidas (dinâmica molecular, análise das séries temporais, etc.). A vantagem do ponto de vista operacional é o menor tempo computacional, devido a (i) termos menos variáveis de estado microscópicas ($6N \rightarrow 4N$), (ii) precisar de sequências de comprimento menor para a aparição de estruturas conformacionais bem diferenciadas e (iii) gerar menor diversidade nas possíveis

sequências da cadeia. Itens todos que contribuem para um menor tempo computacional.

Como primeiro passo devem ser escolhidas as variáveis efetivas apropriadas. A variável Z é uma coordenada de reação melhor do que R_g , como mostrado na Fig. 4.8. Isso porque, enquanto R_g não permite discernir conformações energeticamente diferentes, Z sim. Para cadeias como a do perfil da Fig. 4.9, se Z não permite discriminar poços de potencial, isso reflete o fato de que realmente não existe um número reduzido de conformações energeticamente favoráveis, sendo observadas serpentinas aleatórias. Entretanto, a variável Z tem a limitação de que várias estruturas claramente diferentes (por exemplo, anel e trevo) podem ter Z semelhante, independentemente da sua estabilidade.

Vários estudos que levam em consideração os contatos foram muito bem sucedidos, tanto para estudar as transições de fase quanto para a dinâmica efetiva do enovelamento. Entretanto, no nosso trabalho, nos restringimos à variável Z , após ter testado diferentes variáveis efetivas, por achar que é a que melhor reflete a estrutura espacial de nosso interesse. Ainda, a escolha da variável efetiva depende da dimensão. Com efeito, o problema de determinar uma variável efetiva não é trivial (26) e ainda outras variáveis podem ser pesquisadas.

Mediante a variável efetiva Z , observamos como a sequência da cadeia influencia sobre os potenciais, sendo crucial a hidropaticidade dos monômeros das extremidades. Observamos uma maior sensibilidade do perfil do potencial à sequência quando as extremidades não são hidrofóbicas (Fig. 4.13).

Na faixa de temperatura entre $T = 0,021$ a $T = 0,042$, a cadeia muda de uma conformação compacta, à temperatura inferior, a uma esticada completamente, à temperatura superior (Fig. 2.3). Mas nós estamos interessados em encontrar uma dinâmica efetiva que descreva a evolução na vizinhança do estado nativo, portanto usamos temperaturas na faixa entre 0,021 a 0,026.

No cálculo dos coeficientes de Kramers-Moyal para a variável Z , observamos $D_2 \gg D_4$ o que permite truncar o desenvolvimento à segunda ordem e considerar somente os coeficientes D_1 (de arraste) e D_2 (de difusão) para descrever a dinâmica efetiva.

Em geral, o coeficiente de difusão depende da coordenada de reação. Diferentes estudos mostraram diferentes comportamentos da difusão. Alguns encontraram um máximo no coeficiente para os valores intermédios da coordenada de reação perto do topo da barreira, outros encontraram um decréscimo monotônico em direção ao estado enovelado (27). Esse comportamento

depende da variável considerada.

No nosso caso, o coeficiente de difusão D_2 não é constante mas depende da variável de estado Z , especialmente para as sequências de terminação 1, segundo descrito na seção de Resultados. Portanto, D_2 permite separar os diferentes estados e transições.

É interessante notar que os comportamentos observados para um modelo tão simples como o aqui considerado são compatíveis com os observados previamente para modelos mais sofisticados e levando em conta a variável q (fração de contatos nativos) como coordenada de reação (27, 19).

Referências Bibliográficas

- [1] NELSON, D.; COX, M. **Lehninger Principles of Biochemistry**. W. H. Freeman & Company, 2008.
- [2] WIKIPEDIA. Esta imagen foi adaptada de <https://en.wikipedia.org/wiki/dna>, 2012.
- [3] HYPERPHYSICS. Esta imagen foi adaptada de <http://hyperphysics.phy-astr.gsu.edu/hbase/organic/transcription.html>, 2012.
- [4] MORAN, L. A. Esta imagen foi adaptada de <http://sandwalk.blogspot.com.br/2007/02/anfinsen-experiment-in-protein-folding.html>, 2007.
- [5] ONUCHIC, J. N.; WOLYNES, P. G. **Current opinion in structural biology**. Theory of protein folding., journal, v.14, n.1, p. 70–5, 2004.
- [6] BAKER, D. **Nature**. A surprising simplicity to protein folding., journal, v.405, n.6782, p. 39–42, 2000.
- [7] TANFORD, C.; REYNOLDS, J. **Nature's robots**. Oxford University Press, 2003.
- [8] ANFINSSEN, C. B. **Science**. Principles that Govern the Folding of Protein Chains, journal, v.181, n.4096, p. 223–230, 1973.
- [9] ZWANZIG, R.; SZABO, A. ; BAGCHI, B. Levinthal ' s paradox, journal, v.89, p. 20–22, 1992.
- [10] FLYVBJERG, H. **Physics of biological systems : from molecules to species**. Springer, 1997.
- [11] DILL, K. A.; BROMBERG, S.; YUE, K.; FIEBIG, K. M.; YEE, D. P.; THOMAS, P. D. ; CHAN, H. S. **Protein science : a publication of the Protein Society**. Principles of protein folding—a perspective from simple exact models., journal, v.4, n.4, p. 561–602, 1995.

- [12] SAKURAI, J. **Modern Quantum Mechanics**. Pearson Education, 1994.
- [13] HUANG, K. **Lectures on statistical physics and protein folding**. World Scientific Pub Co Inc, 2005.
- [14] PANDE, V. S.; GROSBERG, A. Y. **Phase Transitions**. Heteropolymer freezing and design : Towards physical models of protein folding, journal, v.72, n.1, p. 259–314, 2000.
- [15] BRYNGELSON, J. D.; WOLYNES, P. G. **Biophysics**. Spin glasses and the statistical mechanics of protein folding, journal, v.84, p. 7524–7528, 1987.
- [16] DERRIDA, B. Random-energy model: An exactly solvable model of disordered systems, journal, v.24, n.5, p. 2613–2626, 1981.
- [17] LAU, K. F.; DILL, K. A. **Macromolecules**. Sequence Spaces of Proteins, journal, v.22, n.10, p. 3986–3997, 1989.
- [18] LOIS, G.; BLAWZDZIEWICZ, J. ; O'HERN, C. S. **Biophysical journal**. Reliable protein folding on complex energy landscapes: the free energy reaction path., journal, v.95, n.6, p. 2692–701, 2008.
- [19] YANG, S.; ONUCHIC, J. N. ; LEVINE, H. **The Journal of chemical physics**. Effective stochastic dynamics on a protein folding energy landscape., journal, v.125, n.5, p. 054910, 2006.
- [20] RISKEN, H. **The Fokker-Planck Equation: Methods of Solution and Applications**. Lecture Notes in Mathematics. Springer-Verlag, 1996.
- [21] ALLEN, M. P.; TILDESLEY, D. J. **Computer Simulation of Liquids**. Oxford University Press, 1989.
- [22] GARDINER, C. W. **Handbook of stochastic methods**. Springer, 2003.
- [23] STROGATZ, S. **Nonlinear Dynamics And Chaos**. Sarat Book House, 2000.
- [24] MIAO, J.; KLEIN-SEETHARAMAN, J. ; MEIROVITCH, H. **Journal of molecular biology**. The optimal fraction of hydrophobic residues required to ensure protein collapse., journal, v.344, n.3, p. 797–811, 2004.
- [25] WHITE, S. H.; JACOBS, R. E. **Biophysical journal**. Statistical distribution of hydrophobic residues along the length of protein chains. Implications for protein folding and evolution., journal, v.57, n.4, p. 911–21, 1990.

- [26] ROHRDANZ, M.; ZHENG, W.; MAGGIONI, M. ; CLEMENTI, C. **The Journal of chemical physics**. Determination of reaction coordinates via locally scaled diffusion map., journal, v.134, n.12, p. 124116, 2011.
- [27] BESTA, R.; HUMMERB, G. **PNAS**. Coordinate-dependent diffusion in protein folding., journal, v.107, n.3, p. 1088–1093, 2010.
- [28] SOCCI, N. D.; ONUCHIC, J. N. ; WOLYNES, P. G. **The Journal of Chemical Physics**. Diffusive dynamics of the reaction coordinate for protein folding funnels, journal, v.104, n.15, p. 5860, 1996.

A

Perfis de energia

Perfis de energia para as sequências analisadas.

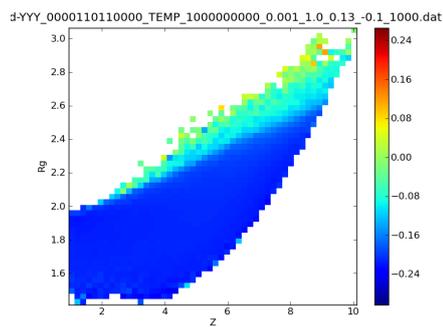


Figura A.1: Perfil de energia para a sequência 0000110110000.

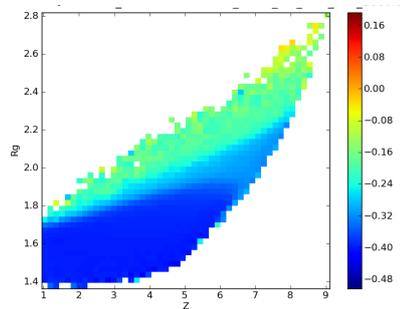


Figura A.2: Perfil de energia para a sequência 0001010101000.

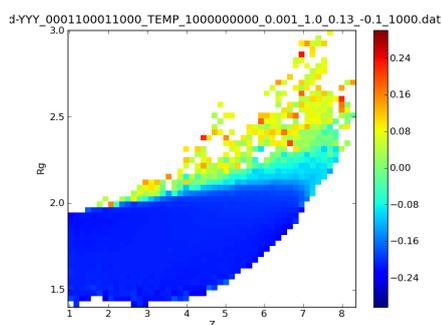


Figura A.3: Perfil de energia para a seqüência 0001100011000.

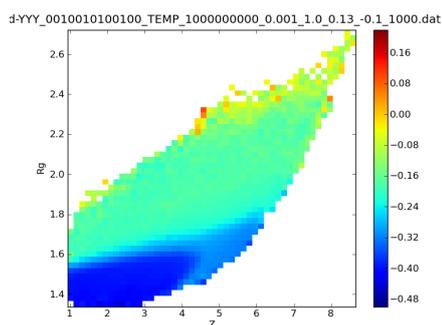


Figura A.4: Perfil de energia para a seqüência 0010010100100.

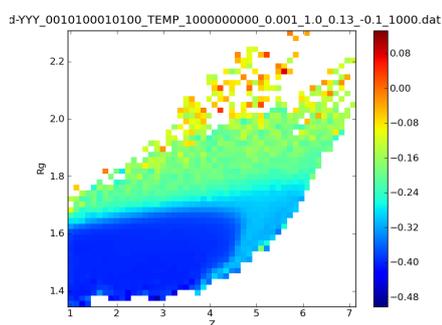


Figura A.5: Perfil de energia para a seqüência 0010100010100.

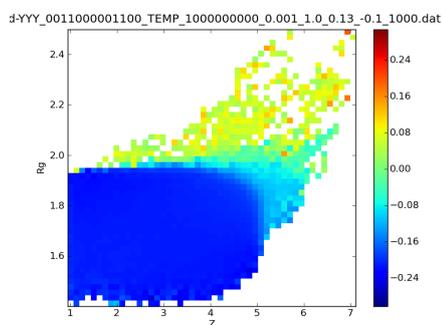


Figura A.6: Perfil de energia para a seqüência 0011000001100.

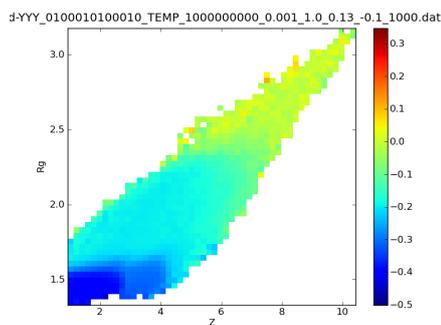


Figura A.7: Perfil de energia para a seqüência 0100010100010.

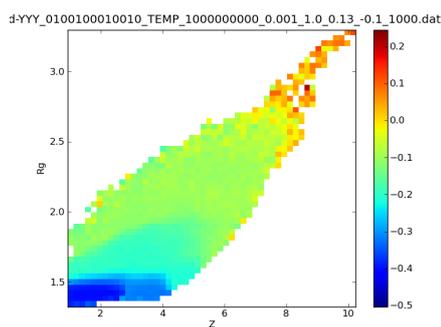


Figura A.8: Perfil de energia para a seqüência 0100100010010.

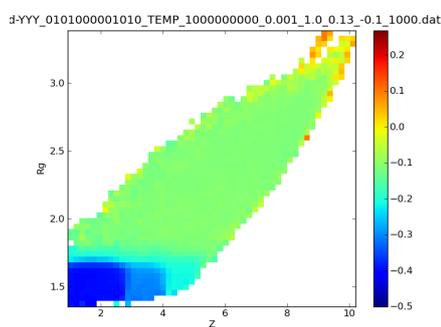


Figura A.9: Perfil de energia para a seqüência 0101000001010.

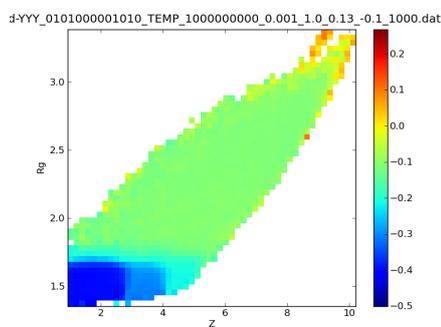


Figura A.10: Perfil de energia para a seqüência 0110000000110.

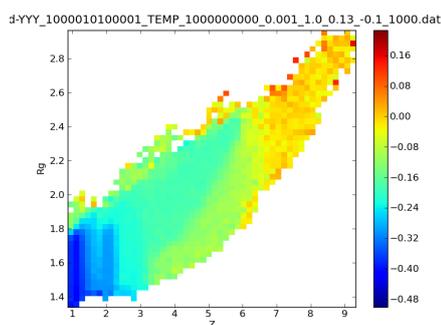


Figura A.11: Perfil de energia para a seqüência 1000010100001.

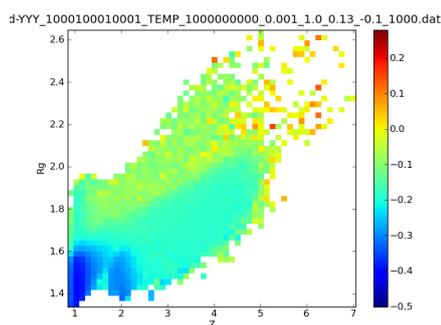


Figura A.12: Perfil de energia para a seqüência 1001000001001.

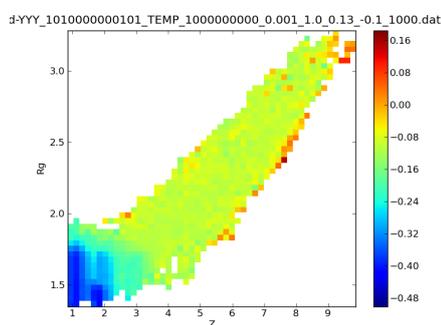


Figura A.13: Perfil de energia para a seqüência 1010000000101.

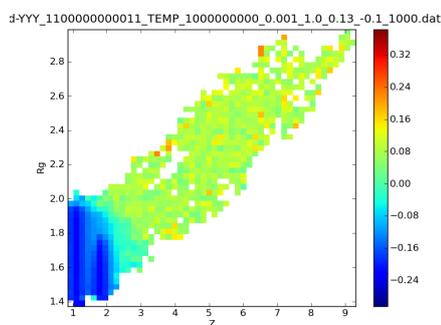


Figura A.14: Perfil de energia para a seqüência 1100000000011.

B Coeficientes de Kramers-Moyal

Cálculo dos coeficientes de Kramers-Moyal para as sequências analisadas na seção 3.

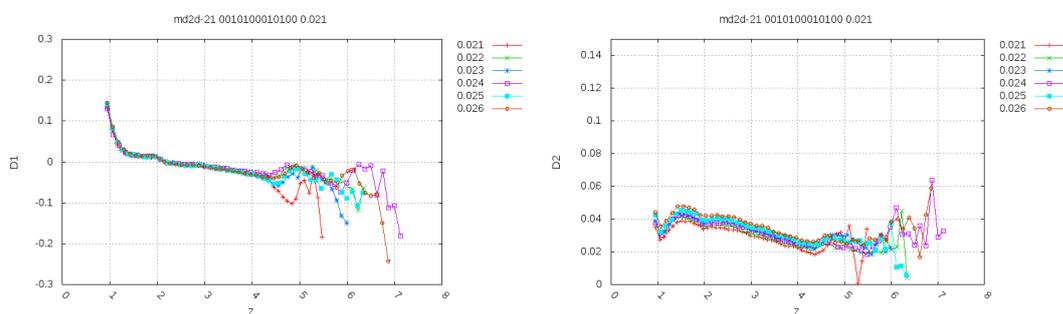


Figura B.1: Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0010100010100.

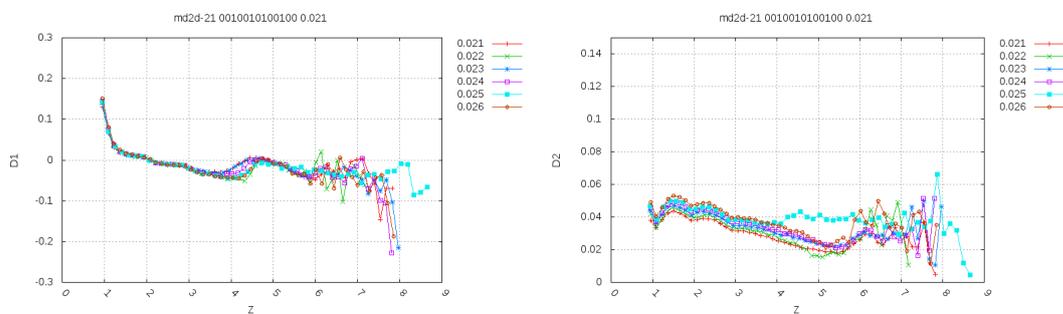


Figura B.2: Coeficientes de Kramers-Moyal D_1 e D_2 da sequência 0010010100100.

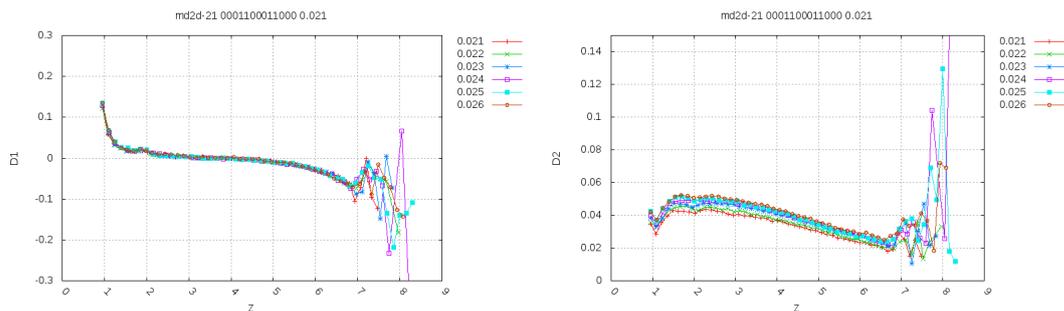


Figura B.3: Coeficientes de Kramers-Moyal D_1 e D_2 da seqüência 0001100011000.

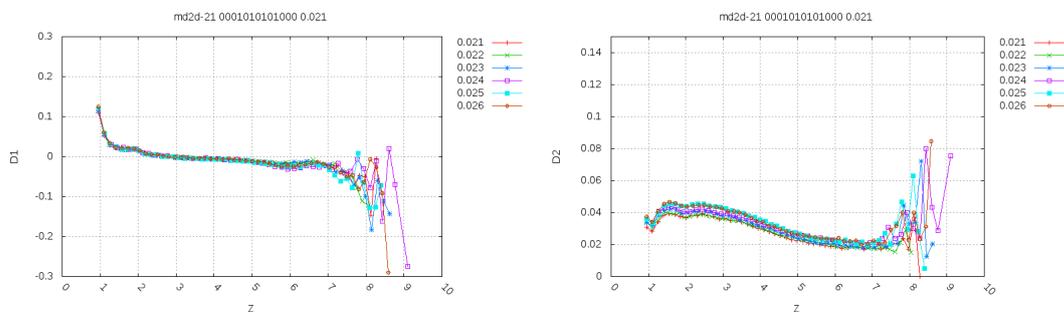


Figura B.4: Coeficientes de Kramers-Moyal D_1 e D_2 da seqüência 0001010101000.

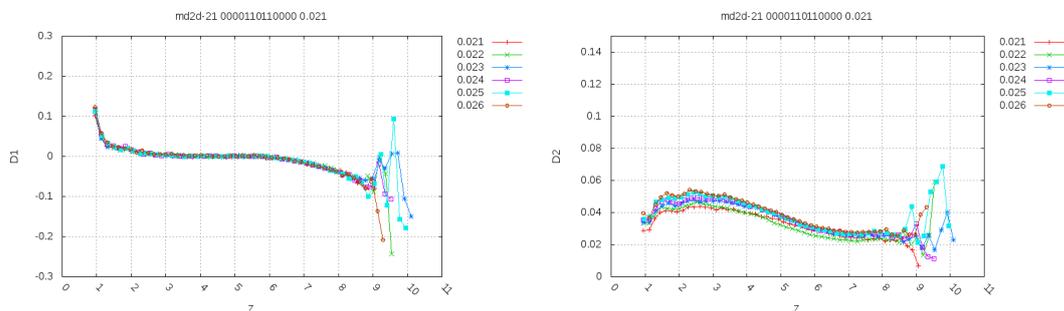


Figura B.5: Coeficientes de Kramers-Moyal D_1 e D_2 da seqüência 0000110110000.

C

Código fonte da dinâmica molecular

O programa está estruturado da seguinte forma.

```
//=====
// Name      : MD.cpp
// Author    : Carlos Olivares
// Version   : 0.5
// Copyright  : GPL
// Description : Molecular dynamics simulation
//=====

// TODO: Add function to calculate the matrix contacts
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <gsl/gsl_math.h>
#include <gsl/gsl_randist.h>
#include <gsl/gsl_rng.h>
#include <time.h>

#include "Conformation.h"

int main(int argc, char* argv[]) {
// input parameters
if(argc != 10){
printf("Usage: MD2d [rev_prefix] [chain] [temperature]
[total_time] [dt] [epsi] [q] [Ec] [print_each]\n");
return EXIT_FAILURE;
}

char *prefix_file, *hydroChain;
```

```
prefix_file = argv[1];
hydroChain = argv[2];
const int M = strlen(argv[2]); //13;
double temp = atof(argv[3]); //0.04;
int total_time = atoi(argv[4]); //200000;
double dt = atof(argv[5]); //0.001;
double epsi = atof(argv[6]); //1.0;
double q = atof(argv[7]); //0.1;
double Ec = atof(argv[8]); //-1.0;
int print_each = atoi(argv[9]);

// FIXME: find a better way to put the cutoff
char strdcutoff[]="1.2";
double dcutoff = atof(strdcutoff); // cutoff for the Contact matrix

// TODO: Add fin file
char filename_pattern[90];
char filename_pdb[100];
char filename_dat[100];
char filename_ini[100];
char filename_con[100];
char filename_seed[100];
char ext_pdb[]=".pdb";
char ext_dat[]=".dat";
char ext_ini[]=".ini";
char ext_con[]=".con";
char ext_seed[]=".seed";

strcpy(filename_pattern, prefix_file);
for (int i=2; i<argc; i++){
    strcat(filename_pattern, "_");
    strcat(filename_pattern, argv[i]);
}

strcpy(filename_pdb, filename_pattern);
strcpy(filename_dat, filename_pattern);
strcpy(filename_ini, filename_pattern);
strcpy(filename_con, filename_pattern);
```

```

strcpy(filename_seed,filename_pattern);

strcat(filename_pdb,ext_pdb);
strcat(filename_dat,ext_dat);
strcat(filename_ini,ext_ini);
strcat(filename_seed,ext_seed);

strcat(filename_con,"_d-");
strcat(filename_con,strdcutoff); //FIXME: dcutoff as string
strcat(filename_con,ext_con);

// Seed for Random numbers
const gsl_rng_type *T; T = gsl_rng_default;
gsl_rng *r; r = gsl_rng_alloc(T);
long unsigned int seed;
seed = (long unsigned int) time(NULL);
gsl_rng_set(r,seed);

//protein.calculateTotalForces(epsil,q,Ec);
FILE *fp_pdb, *fp_dat, *fp_con, *fp_seed;
fp_pdb = fopen(filename_pdb,"w");
fp_dat = fopen(filename_dat,"w");
fp_con = fopen(filename_con,"w");
fp_seed = fopen(filename_seed,"w");

fprintf(fp_dat,"%s\t%s\t%s\t%s\t%s\t%s\n",
"time", "PotentialEnergy", "Rg",
"Z", "HRg", "PRg");
fprintf(fp_con,"%s\t%s\t%s\t%s\t%s\n",
"time", "HHcontacts", "HPcontacts",
"PPcontacts", "ALLcontacts");
fprintf(fp_seed,"seed %lu\n",seed);

// =====> BEGIN Simulation -----
Conformation protein(M, hydroChain, temp, filename_ini);
int ttime = 0;

```

```

int ttaux = total_time/100;
protein.print_pdb_unfoldedchain(fp_pdb);

while(ttime<total_time){
protein.calculateTotalForces(epsil,q,Ec);
protein.actualizePositions(dt);
protein.addPosition2DNoise(dt,temp,r);
protein.actualizeVelocities(dt);
protein.calculateTotalEnergy(epsil,q,Ec);

if ( ttime % print_each == 0 ){
protein.calculateRg();
protein.calculateD();
fprintf(fp_dat,"%d\t%f\t%f\t%f\t%f\t%f\n",
ttime, protein.PotentialEnergy, protein.Rg,
protein.D, protein.HRg, protein.PRg);
//FIXME: Review the number of contacts, something is WRONG!
protein.binarizeDeltaR2(dcutoff);
protein.calculateContacts();
fprintf(fp_con,"%d\t%f\t%f\t%f\t%f\n",
ttime, protein.HHcontacts, protein.HPcontacts,
protein.PPcontacts, protein.ALLcontacts);

}
if (ttime%ttaux==0) protein.print_pdb_conformation(fp_pdb,ttime);
ttime++;
}
protein.print_pdb_conformation(fp_pdb,ttime); // last positions

//=====> END Simulation -----

long unsigned int finaltime;
finaltime = (long unsigned int) time(NULL);
fprintf(fp_seed,"End %lu",finaltime);

fclose(fp_pdb);
fclose(fp_dat);
fclose(fp_con);

```

```
fclose(fp_seed);
// first is printed the existence of an ini file
printf("%s # END_SIMULATION %lu %lu\n",filename_pattern,seed,finaltime);

return EXIT_SUCCESS;
}

/*
 * Conformation.cpp
 *
 * Created on: Feb 7, 2012
 * Author: alfaceor
 */

#include "Conformation.h"

Conformation::Conformation(int N, char *hydroChain, double temp,
char *basename)
{
// Initialize chain
this->N = N;
this->chain = new Monomer[N] ();
this->deltaR2 = new double[N*N] ();
this->bin_dR2 = new double[N*N] ();
this->basename = basename;
this->Energy = 0.0;
this->KinecticEnergy =0.0;
this->PotentialEnergy=0.0;

FILE *fp;
fp=fopen(basename,"r");
//FIXME: temporal flag remove later
if(fp==NULL){
printf("#inifile:False\t");
fp=fopen(basename,"w");

// randomPositions();

// initial velocities equal to zero.
```

```

for(int i=0; i< N; i++){
for (int d=0; d<DIM; d++){
this->chain[i].vec_v[d] = 0.0;
}
chain[i].vec_r[0] = chain[i].sigma*i;
}
//gaussianRandomVelocities(temp); // temp

for (int i=0;i<N;i++){
//-----
// la misma cadena que usa Lois 2008
if(hydroChain[i]=='1'){
    this->chain[i].hydro = -1.0;// Polar or hydrophilic
}else{
    this->chain[i].hydro = 1.0;// Hydrophobic
}

// save data in a file.
fprintf(fp,"%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t%lf\t\n",
    this->chain[i].vec_r[0], this->chain[i].vec_r[1],
    this->chain[i].vec_r[2], this->chain[i].vec_v[0],
    this->chain[i].vec_v[1], this->chain[i].vec_v[2],
    this->chain[i].hydro);
}

}else{
// read a file input data
printf("#infile:True\t");
for (int i=0; i<N; i++){
for (int d=0; d<DIM; d++){
fscanf(fp,"%lf",&this->chain[i].vec_r[d]);
}
for (int d=0; d<DIM; d++){
fscanf(fp,"%lf",&this->chain[i].vec_v[d]);
}
fscanf(fp,"%lf",&this->chain[i].hydro);
}
}
}

```

```

fclose(fp);
}

Conformation::~Conformation() {
// TODO Auto-generated destructor stub
delete deltaR2;
delete bin_dR2;
}

void Conformation::calculateDeltaR2(){
double auxdelta;
for (int k=0; k<N; k++){
for (int l=0; l<N; l++){
deltaR2[k*N+l] = 0.0;
for (int d=0;d<DIM;d++){
auxdelta = (chain[k].vec_r[d] - chain[l].vec_r[d]);
deltaR2[k*N+l] += auxdelta*auxdelta;
}
}
}
deltaR2[l*N+k] = deltaR2[k*N+l];
}
}
}

void Conformation::binarizeDeltaR2(double dcutoff){
double d2cutoff=dcutoff*dcutoff;
for (int k=0; k<N; k++){
for (int l=k; l<N; l++){
if (l == k || l == k+1){ // avoid chain contacts
bin_dR2[k*N+l]=0;
}else{
if (d2cutoff < deltaR2[k*N+l]){
// No contact
bin_dR2[k*N+l]=0;
}else{
// there are in contact
bin_dR2[k*N+l]=1;
}
}
}
}
}

```

```

}
// contact matrix is symmetric
bin_dR2[l*N+k] = bin_dR2[k*N+1];
}
}
}

void Conformation::calculateContacts(){

HHcontacts = 0.0;
HPcontacts = 0.0;
PPcontacts = 0.0;
ALLcontacts = 0.0;

// Count matrix contacts by type HH,HP,PP,ALL
for (int k=0; k<N; k++){
for (int l=k; l<N; l++){
if( chain[k].hydro == -1 ){ // H_k
if ( chain[l].hydro == -1 ){ // H_l
HHcontacts += bin_dR2[k*N+1];
}else{ // P_l
HPcontacts += bin_dR2[k*N+1];
}
}else{ // P_k
if ( chain[l].hydro == -1 ){ // H_l
HPcontacts += bin_dR2[k*N+1]; // HP = PH
}else{ // P_l
PPcontacts += bin_dR2[k*N+1]; // PP
}
}
}
}

ALLcontacts = HHcontacts+HPcontacts+PPcontacts;
}

void Conformation::calculateCenterMass(){
CenterMass[0]=0.0; CenterMass[1]=0.0; CenterMass[2]=0.0;

```

```
for(int i=0; i<N; i++){
for(int d=0; d<DIM; d++){
CenterMass[d]+=chain[i].mass*chain[i].vec_r[d];
}
}

for(int d=0; d<DIM; d++){
CenterMass[d]=CenterMass[d]/N;
}
}

void Conformation::calculateRg(){
// Radius of Gyration: the rms distance of each atom to the
centroid.
double rms = 0.0;
double hrms =0.0;
double prms =0.0;
double aux = 0.0;
int Nh = 0, Np = 0;
calculateCenterMass();
for (int i=0; i<N; i++){
for (int d=0; d<DIM; d++){
aux = (chain[i].vec_r[d]-CenterMass[d])
*(chain[i].vec_r[d]-CenterMass[d]);
rms += aux;
// calculate HRg and PRg
if (chain[i].hydro == -1.0){
// Hydrophobic
hrms += aux;
Nh++;
}else{
// Polar
prms += aux;
Np++;
}
}
}
}
```

```

    Rg = sqrt(rms/N);
    HRg= sqrt(hrms/Nh);
    PRg= sqrt(prms/Np);
}

void Conformation::calculatedD(){
double aux = 0.0;
double vecdiff=0.0;
for (int d=0; d<DIM; d++){
vecdiff=(chain[0].vec_r[d] - chain[N-1].vec_r[d]);
aux += vecdiff*vecdiff;
}
D = sqrt(aux);
}

void Conformation::set_D_to(double Dnew){
// TODO: Change the new value of Ends
chain[N-1].vec_r[0] = Dnew;
D=Dnew;
}

void Conformation::calculateBondForces(double epsi, double q){
double auxvar;
double auxforce[DIM];
for (int m=0;m<N-1;m++){
auxvar = force_cc_r(epsi,q,deltaR2[m*N+(m+1)]);
for (int d=0;d<DIM;d++){
auxforce[d] = auxvar*(chain[m].vec_r[d] - chain[m+1].vec_r[d]);
chain[m].total_force[d] += auxforce[d];
chain[m+1].total_force[d] += -auxforce[d];
}
}
}

void Conformation::calculateBondPotential(double epsi,double q){
double auxvar=0.0;

```

```

for (int i=0;i<N-1;i++){
auxvar += phi_cc(epsil,q,deltaR2[i*N+(i+1)]);
}
PotentialEnergy+=auxvar;
}

void Conformation::calculateHydroForces(double epsil,double Ec){
double auxvar;
double auxforce[DIM];
for (int m=0;m<N-2;m++){
// FIXME: las fuerzas hidrophobicas o hidrofílicas
for (int i=m+2;i<N;i++){
auxvar =
force_hydro(epsil,Ec,deltaR2[m*N+i],chain[m].hydro+chain[i].hydro);
for (int d=0;d<3;d++){
auxforce[d] = auxvar*(chain[m].vec_r[d]-chain[i].vec_r[d]);
chain[m].total_force[d] += auxforce[d];
chain[i].total_force[d] += -auxforce[d];
}
}
}
}

void Conformation::calculateDampingForces(){
for (int i=0; i<N; i++){
for (int d=0; d<DIM; d++){
chain[i].total_force[d] += -chain[i].gamma*(chain[i].vec_v[d]);
}
}
}

void Conformation::calculateRandomForces(){
for (int i=0; i<N; i++){
//chain[i].total_force
}
}

void Conformation::calculateHydroPotential(double epsil,double

```

```

Ec){
    double auxvar=0.0;
    for(int m=0; m<N-2; m++){
        for (int i=m+2;i<N;i++){
            auxvar +=
            potential_hydro(epsil,Ec,deltaR2[m*N+i],chain[m].hydro+chain[i].hydro);
        }
    }
    PotentialEnergy+=auxvar;
}

```

```

void Conformation::calculateTotalForces(double epsil, double q,
double Ec){
    cleanForces();
    calculateDeltaR2();
    calculateBondForces(epsil,q);
    // FIXME:
    // Para realizar algunas pruebas de la resistencia de
    // la cadena se comentan las hydroforces momentaneamente
    calculateHydroForces(epsil,Ec);
}

```

```

void Conformation::calculateTotalEnergy(double epsil,double q,
double Ec){
    cleanEnergyValues();
    calculateBondPotential(epsil,q);
    calculateHydroPotential(epsil,Ec);
    calculateKineticEnergy();
    Energy = KineticEnergy + PotentialEnergy;
}

```

```

void Conformation::setTemperature(double temp){
    // FIXME: First calculateTemperature and then scaled
    calculateTemperature();
}

```

```

double scale_temp=1/sqrt(Temperature/temp);

```

```

for (int i=0;i<N;i++){

```

```
for (int d=0; d<DIM; d++){
chain[i].vec_v[d] = scale_temp*chain[i].vec_v[d];
}
}
}

void Conformation::calculateTemperature(){
// FIXME:
calculateKineticEnergy();

Temperature = 2.0*KinecticEnergy/3.0; // XXX: Boltzmann constant
}

void Conformation::actualizePositions(double dt){
for (int i=0;i<N;i++){
chain[i].actualizeVec_r(dt);
}
}

void Conformation::actualizePositionsFixedEnds(double dt){
for (int i=1;i<N-1;i++){
chain[i].actualizeVec_r(dt);
}
}

void Conformation::addPosition2DNoise(double dt, double KT,
gsl_rng *r){

for (int i=0;i<N;i++){
chain[i].addPosition2DNoise(dt,KT,r);
}

}

void Conformation::addPosition3DNoise(double dt, double KT,
gsl_rng *r){
```

```
for (int i=0;i<N;i++){
chain[i].addPosition3DNoise(dt,KT,r);
}

}

void Conformation::addPosition3DNoiseFixedEnds(double dt,double
KT, gsl_rng *r){

for (int i=1;i<N-1;i++){
chain[i].addPosition3DNoise(dt,KT,r);
}

}

void Conformation::actualizeVelocities(double dt){
for (int i=0;i<N;i++){
chain[i].actualizeVec_v(dt);
}
}

void Conformation::actualizeVelocitiesFixedEnds(double dt){
for (int i=1;i<N-1;i++){
chain[i].actualizeVec_v(dt);
}
}

void Conformation::calculateKineticEnergy(){
double kinectic_energy=0.0;
for (int i=0; i<N; i++){
for (int d=0; d<DIM; d++){
kinectic_energy+=chain[i].mass*(chain[i].vec_v[d]*chain[i].vec_v[d]);
}
}
KinecticEnergy=0.5*kinectic_energy;
}
```

```
void Conformation::displace(double d_x, double d_y, double d_z){
// TODO: make a displacement
// double vec_displace[3];
// vec_displace[0]= - chain[0].vec_r[0];
// vec_displace[1]= - chain[0].vec_r[1];
// vec_displace[2]= - chain[0].vec_r[2];
for (int i=0; i<N; i++){
chain[i].vec_r[0] = chain[i].vec_r[0] + d_x;
chain[i].vec_r[1] = chain[i].vec_r[1] + d_y;
chain[i].vec_r[2] = chain[i].vec_r[2] + d_z;
}
}

void Conformation::alingWithDaxis(){

printf("Before DISPLACE\n%f\t%f\t%f\n",
chain[N-1].vec_r[0],chain[N-1].vec_r[1],chain[N-1].vec_r[2]);

displace(-chain[0].vec_r[0],-chain[0].vec_r[1],-chain[0].vec_r[2]);
printf("Before rotate\n%f\t%f\t%f\n",chain[N-1].vec_r[0],
chain[N-1].vec_r[1],chain[N-1].vec_r[2]);

double norma_xyz = sqrt(
(chain[N-1].vec_r[0])*(chain[N-1].vec_r[0])+
(chain[N-1].vec_r[1])*(chain[N-1].vec_r[1])+
(chain[N-1].vec_r[2])*(chain[N-1].vec_r[2])
);

double norma_xy = sqrt(
(chain[N-1].vec_r[0])*(chain[N-1].vec_r[0])+
(chain[N-1].vec_r[1])*(chain[N-1].vec_r[1])
);

printf("norma_xy=%f\tnorma_xyz=%f\n",norma_xy,norma_xyz);
double cos_phi = chain[N-1].vec_r[0]/norma_xy;
double sin_phi = chain[N-1].vec_r[1]/norma_xy;
double cos_tetha = chain[N-1].vec_r[2]/norma_xyz;
double sin_tetha = norma_xy/norma_xyz;
```

```

//Rotation in polar angle
for (int i=0; i<N; i++){
    double aux0 = chain[i].vec_r[0]*cos_phi + chain[i].vec_r[1]*sin_phi;
    double aux1 = -chain[i].vec_r[0]*sin_phi + chain[i].vec_r[1]*cos_phi;
    chain[i].vec_r[0] = aux0;
    chain[i].vec_r[1] = aux1;
}

printf("After rotate Polar\n%f\t%f\t%f\n",
    chain[N-1].vec_r[0],chain[N-1].vec_r[1],chain[N-1].vec_r[2]);

//Rotation in azimuthal angle
for (int i=0; i< N; i++){
    double aux0 =
        chain[i].vec_r[0]*sin_tetha + chain[i].vec_r[2]*cos_tetha;
    double aux2 =
        chain[i].vec_r[0]*cos_tetha - chain[i].vec_r[2]*sin_tetha;
    chain[i].vec_r[0] = aux0;
    chain[i].vec_r[2] = aux2;
}

printf("After rotate azimuthal\n%f\t%f\t%f\n",chain[N-1].vec_r[0],
    chain[N-1].vec_r[1],chain[N-1].vec_r[2]);
}

void Conformation::cleanForces(){
for(int i=0; i<N; i++){
for (int d=0; d<DIM; d++){
chain[i].total_force_old[d] = chain[i].total_force[d];
chain[i].total_force[d]=0.0;
}
}
}

void Conformation::cleanEnergyValues(){
this->Energy = 0.0;

```

```
this->KineticEnergy = 0.0;
this->PotentialEnergy= 0.0;
}

void Conformation::randomPositions(){
gsl_rng *r1;
const gsl_rng_type * T;
T = gsl_rng_default;
r1 = gsl_rng_alloc(T);
double randomaux;

for (int i=0; i<N; i++){
for (int d=0;d<DIM;d++){
// vector positions and velocities
chain[i].vec_r[d] = 0.0;
if (d==0){
randomaux = 0.001*gsl_rng_uniform_pos(r1);
chain[i].vec_r[d] = randomaux;
}
}
chain[i].vec_r[0] += chain[i].zigma*i;
}

}

void Conformation::gaussianRandomVelocities(double temp){
gsl_rng *r2;
gsl_rng *r1;
const gsl_rng_type * T;
T = gsl_rng_default;
r2 = gsl_rng_alloc(T);
r1 = gsl_rng_alloc(T);
double randomaux;
double randomang;
// double temp = 0.04;
double avg_vel[DIM];

for (int i=0;i<N;i++){
```

```
randomaux = gsl_ran_gaussian(r2,1);
randomang = M_PI*gsl_rng_uniform(r1);
for (int d=0;d<DIM;d++){
// vector positions and velocities
chain[i].vec_v[d] = 0.0;
// vector velocities in any direction.

if (d==0){
chain[i].vec_v[d] = randomaux*cos(randomang);
}else if (d==1){
chain[i].vec_v[d] = randomaux*sin(randomang);
}else{
chain[i].vec_v[d] = 0.0;
}
avg_vel[d] += chain[i].vec_v[d];
}
}
// TOTAL MOMENTUM = 0 (ZERO)
for (int i=0;i<N;i++){
for (int d=0; d<DIM; d++){
if(i == 0){ avg_vel[d] = avg_vel[d]/N; }
chain[i].vec_v[d] -= avg_vel[d];
}
}

// Set temperature for the system
setTemperature(temp);
}

void Conformation::print_r(){
for(int i=0;i<N;i++){
this->chain[i].print_r(); printf("\n");
}
}

void Conformation::print_pdb_conformation(FILE *fp,int time_model){
char name[]=" ";
fprintf(fp,"MODEL\t%d\n",time_model);
```

```

for (int i=0;i<N;i++){
    if(chain[i].hydro == 1){ name[0]='Z';name[1]='n';}
    else{ name[0]='N';name[1]=' ';}
    print_pdb_line(fp,i+1,chain[i].vec_r[0],chain[i].vec_r[1],
        chain[i].vec_r[2],name,chain[i].vec_v[0]);
}
fprintf(fp,"ENDMDL\n");
printf("ttime=%d\n",time_model);
}

void Conformation::print_pdb_unfoldedchain(FILE *fp){
    char name[]=" ";
    int time_model = -1;
    fprintf(fp,"MODEL\t%d\n",time_model);
    for (int i=0;i<N;i++){
        if(chain[i].hydro == 1){ name[0]='Z';name[1]='n';}
        else{ name[0]='N';name[1]=' ';}
        // Chain in x axis and in corresponding positions
        // The propose of this is just make the first pdb
        // model to a good view in vmd (visualization program)
        print_pdb_line(fp,i+1,
            chain[i].zigma*i, 0.0, 0.0,name,chain[i].vec_v[0]);
    }
    fprintf(fp,"ENDMDL\n");
    printf("ttime=%d\n",time_model);
}

void Conformation::print_pdb_line(FILE *fp,int serial, double x,
double y,
    double z,char *name,double tempFactor){
    char recordname[]="HETATM"; // 1 - 6 Record name "HETATM"
// int serial =1; // 7 - 11 Integer Atom serial number.
// char name []=" "; //13 - 16 Atom Atom name.
    char altLoc []=" "; //17 character Alternate location
indicator.
    char resName []=" "; //18 - 20 Residue name Residue name.
    char chainID []="A"; //22 character Chain identifier.
    char resSeq []="1";

```

```

    char iCode    []=" ";
// double x      = 13.872;
// double y      = -2.555;
// double z      = -29.045;
    double occupancy = 1.00;
// double tempFactor = 27.36;
    char element []=" ";
    char charge  []=" ";

// char pdb_line[80];
// const char atom_line_iformat[]=
//"%6s%5d %4s%1s%3s %1s%4s%1s  %8.3f%8.3f%8.3f%6.2f%6.2f
//  %2s%2s";
//
// sprintf(pdb_line,atom_line_iformat, recordname, serial, name,
altLoc,
resName, chainID, resSeq, iCode, x, y, z, occupancy, tempFactor,
element,charge );
    fprintf(fp,
"%6s%5d %4s%1s%3s %1s%4s%1s  %8.3f%8.3f%8.3f%6.2f%6.2f
%2s%2s",
        recordname, serial, name, altLoc, resName, chainID, resSeq,
iCode, x, y, z,
        occupancy, tempFactor, element,charge );
    fprintf(fp, "\n");
}

/*
 * Monomer.cpp
 *
 * Created on: Feb 7, 2012
 * Author: alfaceor
 */

#include "Monomer.h"

Monomer::Monomer() {
    zigma=1;

```

```

mass=1;
hydro=1;
gamma=0.001;
for (int d=0; d<DIM; d++){
total_force[d]=0.0;
total_force_old[d]=0.0;
vec_r[d]=0.0;
vec_v[d]=0.0;
}
}

Monomer::~Monomer() {
// TODO Auto-generated destructor stub
}

void Monomer::actualizeVec_v(double dt){
for (int d=0; d<DIM; d++){
//vec_v[d] += 0.5*dt*(total_force[d]+total_force_old[d])/mass;
vec_v[d] = (vec_r[d] - vec_r_old[d])/dt;
}
}

void Monomer::actualizeVec_r(double dt){
for (int d=0; d<DIM; d++){
//vec_r[d] += dt*vec_v[d]+0.5*(dt*dt)*total_force[d]/mass;
vec_r_old[d] = vec_r[d];
vec_r[d] += dt*total_force[d];
}
}

void Monomer::addPosition2DNoise(double dt, double KT, gsl_rng *r){
// Position noise for 2D dimensions.

double etha=sqrt(2*KT*dt)*gsl_ran_gaussian(r,1);

double phi =gsl_rng_uniform_pos(r);
vec_r[0] += etha*cos(phi);
vec_r[1] += etha*sin(phi);

```

```
vec_r[2] = 0;
}

void Monomer::addPosition3DNoise(double dt, double KT, gsl_rng *r){
// Position noise for 3D dimensions.
double etha=sqrt(2*KT*dt)*gsl_ran_gaussian(r,1);
double teta=gsl_rng_uniform_pos(r);
double phi =gsl_rng_uniform_pos(r);
//printf("%lf\t%lf\t%lf\n",etha,teta,phi);
vec_r[0] += etha*sin(teta)*cos(phi);
vec_r[1] += etha*sin(teta)*sin(phi);
vec_r[2] += etha*cos(teta);
}

void Monomer::print_r(){
for (int i=0;i< DIM;i++){
printf("%f\t",vec_r[i]);
}
}

void Monomer::print_v(){
for (int i=0;i< DIM;i++){
printf("%f\t",vec_v[i]);
}
}

/*
 * util_functions.cpp
 *
 * Created on: Mar 8, 2012
 * Author: alfaceor
 */
#include "util_functions.h"

// r2 = r*r
double phi_cc(double epsilon, double q,double r2){
if (r2<=1){
double r6 = 1/(r2*r2*r2);
double r12 = r6*r6;
}
```

```

return (epsilon*(r12 -2*r6 +1));
}else{
double r = sqrt(r2);
return (-epsilon*log(1-((r-1)*(r-1))/(q*q)));
}
}

double phi_att(double epsilon,double E_c,double r2){
double r6 = 1/(r2*r2*r2);
double r12 = r6*r6;
return (-epsilon*E_c*(r12-2*r6));
}

double phi_rep(double epsilon,double r2){
double r6 = 1/(r2*r2*r2);
double r12 = r6*r6;
if (r2<=1){
return ( epsilon*(r12 - 2*r6 + 1) );
}else{
return 0;
}
}

double potential_hydro(double epsilon,double E_c,
double r2, int hydroValue){
if(hydroValue===-2){
// fuerza atractiva
return(phi_att(epsilon,E_c,r2));
}else{
// fuerza repulsiva
return(phi_rep(epsilon,r2));
}
}

// Calculate quadratic norm
double norm2(double *vector1, double *vector2, int d){
double norm2=0;
for (int i=0;i<d;i++){

```

```

norm2 += (vector1[i]-vector2[i])*(vector1[i]-vector2[i]);
}
return norm2;
}

```

```

// FORCES
// FIXME: Complete function
double force_cc_r(double epsilon, double q, double r2){
double force;
double r6 = 1.0/(r2*r2*r2);
double r8 = r6*(1.0/(r2));

if(r2 <=1){
force = ( 12*epsilon*r8*(r6-1) );
}else{
double r = sqrt(r2);
if(r < 1+q){
double aux = (r-1)/q;
double denominador = r*q*(1.0/aux - aux);
force = -2*epsilon/denominador;
}else{
force = 0;
}

}
return force;
}

```

```

double force_att_r(double epsilon, double Ec, double r2){
double r6 = 1.0/(r2*r2*r2);
double r8 = r6*(1.0/(r2));

return (-12*epsilon*Ec*r8*(r6-1));
}

```

```

double force_rep_r(double epsilon, double r2){
double force;
double r6 = 1.0/(r2*r2*r2);

```

```
double r8 = r6*(1.0/(r2));  
if (r2<1){  
force = 12*epsilon*r8*(r6-1);  
}else{  
force = 0;  
}  
return force;  
}
```

```
double force_hydro(double epsilon, double Ec,  
    double r2, int hydroValue){  
if(hydroValue==-2){  
// fuerza atractiva  
return(force_att_r(epsilon,Ec,r2));  
}else{  
// fuerza repulsiva  
return(force_rep_r(epsilon,r2));  
}  
}
```