



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

André Luiz Martins

Uso de modelo de Rede no Estudo do Enovelamento de Proteínas

São José do Rio Preto
2012

André Luiz Martins

Uso de modelo de Rede no Estudo do Enovelamento de Proteínas

Tese apresentada como parte dos requisitos para obtenção do título de Doutor em Biofísica Molecular, junto ao Programa de Pós-Graduação em Biofísica Molecular, Área de Concentração – Biofísica Molecular, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientador: Prof. Dr. Jorge Chahine

São José do Rio Preto
2012

Martins, André Luiz

Uso de modelos de rede no estudo do enovelamento de proteínas / André Luiz Martins. - São José do Rio Preto: [s.n.], 2009.

117 f. : il.35; 30 cm.

Orientador: Jorge Chahine

Dissertação (doutorado) - Universidade Estadual Paulista, Instituto de Biociências, Letras e Ciências Exatas.

1. Crescimento de cadeia. 2. Predição de estrutura de proteína. 3. Estado de transição. 4. Φ -values. I. Chahine, Jorge. II. Universidade Estadual Paulista, Instituto de Biociências, Letras e Ciências Exatas. III. Título.

CDU – 577.112

André Luiz Martins

Uso de modelo de Redes no Estudo do Enovelamento de Proteínas

Tese apresentada como parte dos requisitos para obtenção do título de Doutor em Biofísica Molecular, junto ao Programa de Pós-Graduação em Biofísica Molecular, Área de Concentração – Biofísica Molecular, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Banca Examinadora

Prof. Dr. Jorge Chahine
UNESP – São José do Rio Preto
Orientador

Prof. Dr. Leandro Cristante de Oliveira
CTBE – Laboratório Nacional de Ciência e Tecnologia do Bioetanol

Prof. Dr. Ronaldo Junio de Oliveira
CTBE – Laboratório Nacional de Ciência e Tecnologia do Bioetanol

Prof. Dr. Marcelo Andrés Fossey
UNESP – São José do Rio Preto

Prof. Dr. Sidney Jurado de Carvalho
UNESP – São José do Rio Preto

São José do Rio Preto
18 de Dezembro de 2012

Feliz o homem que encontrou a sabedoria e alcançou o entendimento, porque a sabedoria vale mais do que a prata, e dá mais lucro que o ouro.

Pro 3, 13-14

Agradecimentos

Em primeiro lugar, agradeço a Deus, por me conceder o dom da vida.

Foram muitas pessoas que contribuíram, direta ou indiretamente, para a realização desse trabalho de doutorado.

Gostaria de agradecer ao Prof. Dr. Jorge Chahine pela sábia orientação desse trabalho, pela sua paciência e pelos seus valiosos ensinamentos para o meu amadurecimento profissional e pessoal.

Quero agradecer também à minha esposa, Marcela R. L. Martins, que me trouxe o bem mais valioso que um homem pode ter: um filho, o ser que nos enche de alegria, e preocupação, o meu bem amado João Francisco L. Martins. Aos meus pais Gilberto C. Martins e Marlene P. Martins; irmãos Célio R. M. e Gilda M. A., cunhado(a) Simonia C. M. e Fabrício O. A.; e aos meus queridos sobrinhos: Alex C. M., Arthur C. M., Fabrício Junior M. A., Gabriel M. A. e Murilo M. A.

Sumário

Índice de Figuras	5
Resumo	3
Abstract	4
Capítulo 1	12
Introdução	12
Introdução e Motivação	12
Capítulo 2	15
2.1 Aminoácidos: blocos de construção das Proteínas	15
2.2 Estruturas de Proteínas	22
Capítulo 3	26
Metodologia	26
3.1 – Descrição da Rede Cubo-Octaédrica	26
3.2 Algoritmo de Crescimento de Estruturas Proteicas em Rede: Descrição do Método Estatístico	31
3.3 Algoritmo de Predição de Estruturas Proteicas com Potencial Estatístico de Interação entre os Resíduos de Aminoácidos	36
Capítulo 4	42
4.1 Estudo do Estado de Transição com o Potencial de Interação Dependente da Distância entre os Resíduos de Aminoácidos da Estrutura Nativa	42
Capítulo 5	48
Resultados	48
5.1 Método de Crescimento de Estruturas sem Interação	48
5.2 Algoritmo de Crescimento de Proteína com Potencial de Interação entre Resíduos de Aminoácidos	54
5.3 Estudo do Estado de Transição da Proteína <i>I coa</i>	59

Capítulo 6	68
6.1 Conclusão e Perspectivas Futuras	68
Referências Bibliográficas	70
Apêndice A	74
Figuras do Algoritmo de Predição de Estruturas Proteicas	74
Apêndice B	86
Algoritmo de Dinâmica para o Estudo do Estado de Transição da Proteína 1 <i>c o a</i>	86

Índice de Figuras

Figura 1: Fórmula geral ionizada de um aminoácido α . R representa a cadeia lateral.	15
Figura 2: Aminoácidos com cadeias laterais alifáticas.	16
Figura 3: Representação estrutural do aminoácido Prolina.	16
Figura 4: Aminoácidos com cadeias laterais aromáticas.	17
Figura 5: Aminoácidos com enxofre em suas cadeias laterais.	18
Figura 6: Aminoácidos que têm cadeias laterais alifáticas hidroxiladas.	18
Figura 7: Aminoácidos com cadeias laterais básicas.	19
Figura 8: Aminoácidos com cadeias laterais ácidas, aspartato e glutamato, e suas amidas derivadas, asparagina e glutamina.	20
Figura 9: Representação esquemática da formação de uma ligação peptídica – destacada em amarelo na figura.	21
Figura 10: Ligação dissulfeto entre resíduos de cisteína.	21
Figura 11: Estrutura 3D de uma proteína de colágeno, código PDB 2f6a.	24
Figura 12: Estrutura quaternária da hemoglobina, código PDB 3qjb.	25
Figura 13: Estrutura de uma proteína de membrana, código do PDB 2wmz.	25
Figura 14 - Ilustração do modelo de uma cela unitária da rede adotada nas simulações. O ponto central, em vermelho, tem 12 vizinhos mais próximos, pontos em azul: (a) 6 no mesmo plano, (b) 3 acima e 3 abaixo deste plano . A distância entre 2 sítios vizinhos é de 3,8 Å.	28
Figura 15: Ilustração dos ângulos permitidos na rede. O ponto em vermelho é o sítio central e faz um ângulo de (a) 60°, (b) 90°, (c) 120° e (d) 180° com outros dois sítios.	29
Figura 16: Exemplo de uma estrutura proteica de 20 resíduos: (a) permitida e duas estruturas não permitidas com (b) ângulo obtuso, 180° e (c) com superposição de dois resíduos.	32
Figura 17: Ilustração de um passo no processo de crescimento. O ponto vermelho representa o resíduo i e o verde o $i+1$	35
Figura 18: Valores do raio de giro médio versus N em um gráfico di -log para as estruturas geradas pelo algoritmo de crescimento com peso estatístico W_N . A inclinação da curva ajustada é de aproximadamente 0,58.	49
Figura 19: Valores do raio de giro médio versus N em um gráfico di -log para as estruturas geradas pelo algoritmo de crescimento sem o peso estatístico W_N . A inclinação da curva ajustada é de 0,52. .	50
Figura 20: Gráfico di -log da sobreposição das duas curvas do raio de giro médio em função do número de monômeros; a curva em vermelho é com o peso estatístico e a azul é sem o peso.	51

Figura 21: Valores do raio de giro médio versus N em um gráfico <i>di-log</i> para as estruturas geradas pelo algoritmo de crescimento na rede cúbica com o peso estatístico. A inclinação da curva ajustada é de 0,59.....	53
Figura 22: Distribuição da Energia em função do RMSD (Å) do ensemble de estruturas geradas para a proteína 2 mhu em $\beta = 0.5$. O menor RMSD é de 4.0 Å.....	56
Figura 23: Distribuição do Raio de giro em função do RMSD (Å) do ensemble de estruturas geradas para a proteína 2mhu em $\beta = 0.5$. O menor RMSD é de 4.0 Å.....	56
Figura 24: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 0,5$ e RMSD = 4.0 Å. Código PDB da proteína: 2 mhu.	57
Figura 25: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 0,5$. Código PDB da proteína: 1crn. Entre os resíduos 7 e 22 as estruturas apresentam um RMSD de 3,4 Å.....	58
Figura 26: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 0,5$ e RMSD = 5.5 Å. Código PDB da proteína: 1r69. .	58
Figura 27: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 1,02$ e RMSD = 5.8 Å. Código PDB da proteína: 2 cro.	58
Figura 28: Ilustração das estruturas secundárias da proteína 1 coa.....	59
Figura 29: Ao transcorrer da simulação a estrutura visita estados conformacionais desenovelados caracterizados por baixos valores de Q , ilustrado em (a) $\beta = 0,4$, ou seja, $T = 2.5$; em (b) temos flutuação entre os estados enovelado e desenovelado, temperaturas próximas do estado de transição, em torno de $\beta = 0,8$, ou seja, $T = 1.25$, e (c) para temperaturas acima do estado de transição, $\beta = 1,4$, ou seja, $T = 0,71$, a estrutura permanece a maior parte do tempo no estado enovelado, com cerca de 75 % dos contatos nativos sendo realizados.....	61
Figura 30: Gráfico do calor específico para a proteína 1 coa. Sabemos que o calor específico mede a flutuação da energia, $\frac{C}{k} = \beta^2[\langle E^2 \rangle - \langle E \rangle^2]$, sendo assim o pico da curva nos mostra que a temperatura de transição de fase da estrutura está em torno da temperatura 1,3.	62
Figura 31: Gráfico da energia livre em função do número de contatos em diferentes valores de temperatura.....	62
Figura 32: Ilustração representando a barreira de energia livre dos estados de transição de proteínas (XIU, 2012).	64
Figura 33: Gráficos da sobreposição do RMSD – curva em azul – e Raio de Giro – curva em vermelho – da estrutura ao longo da simulação, ambos estão medidos em Å. Em (a) $T = 2.5$, a temperatura está acima; em (b) $T = 1.25$, está próximo e, (c) $T = 0.7$, está abaixo da temperatura de transição.....	65
Figura 34: Sobreposição dos ϕ values obtidos pelo modelo em rede realizado neste trabalho – barra em preto – com os valores obtidos teoricamente com o modelo C_α no espaço contínuo.	66

Figura 35: Valores de $\phi_{i,j}$ values da estrutura no estado de transição para a proteína *Icoa* segundo o modelo C_α (a) na rede cubo-octaédrica e (b) no espaço contínuo. Diferentes cores indicam diferentes valores de 0 a 1, como quantificado por uma escala de cores. De uma forma geral, o ensemble do estado de transição apresenta-se como uma versão expandida do estado nativo onde a maioria dos contatos tem uma mesma probabilidade de participação, mas algumas interações são menos prováveis de ocorrerem (CLEMENTI, 2000)..... 67

Resumo

O enovelamento de proteínas é um problema fundamental em Biofísica Molecular. Propõe-se neste trabalho um método de crescimento de estruturas proteicas utilizando uma rede denominada cubo-octaédrica. Neste processo de crescimento a partir da estrutura primária utiliza-se um potencial estatístico de interação entre pares de resíduos a fim de determinarmos a estrutura terciária da sequência. Para o *ensemble* estrutural analisam-se propriedades estruturais como o raio de giro das cadeias e o desvio médio quadrático das distâncias. Além deste estudo de previsão de estruturas, analisamos parâmetros energéticos termodinâmicos no estado de transição da proteína CI2 (cujo código PDB é *1 coa*) bem como o cálculo dos valores de ϕ -values para os resíduos que fazem contatos no estado de transição.

Palavras-chave: crescimento de cadeia proteica em rede, algoritmo de previsão de estruturas, enovelamento de proteínas, estado de transição.

Abstract

Protein folding is a fundamental problem in Molecular Biophysics. We present a method of chain growth algorithm to lattice generation of protein structure. The interaction between pairs of residues in this process is a statistical potential proposed by THOMAS and DILL. We examine properties as radius of gyration and root mean square deviation of ensemble protein structure. Finally, we verify the thermodynamics energetic parameter in the transition state of protein folding 1 coa.

Keywords: *chain growth in lattice, protein prediction structure algorithm, protein folding, transition state.*

Capítulo 1

Introdução

Introdução e Motivação

As proteínas são moléculas que desempenham os mais diversos papéis no processo da vida. Existem proteínas responsáveis pela catálise, proteínas de transporte e armazenamento, proteínas do sistema imunológico e proteínas regulatórias (CREIGHTON, 2003). A função de uma proteína está fortemente ligada a sua estrutura nativa, que é o produto final do processo de enovelamento. O estudo desse processo, através de um modelo de rede simplificado, é a motivação desse trabalho.

Inicialmente, estudamos o método de crescimento de cadeia proteica em rede, (ROSENBLUTH, 1945). Uma rede consiste num conjunto de pontos do espaço tridimensional, denominados de sítios, que serão utilizados para introduzir um resíduo. Cada

sítio da rede poderá ou não receber um resíduo de aminoácido durante o processo de crescimento de cadeia ou de dinâmica, conforme veremos a seguir. Dessa forma, nossa simulação tem um espaço discreto, ao invés de espaço contínuo, o que na maioria dos algoritmos de dinâmica molecular contribui para o aumento do tempo de simulação. As simulações são divididas em três partes:

- *na primeira parte* não existe interação entre os aminoácidos que não a de volume excluído. Assim, o crescimento da cadeia segue as regras de um passeio aleatório. O objetivo, nessa fase, é o de verificar se o algoritmo reproduz bem o expoente que governa a dependência do raio de giro da cadeia em função no número N de aminoácidos idênticos. Esse passo é importante para se certificar que os sítios da rede cubo-octaédrica – descrita na seção 3.1 – estão sendo visitados de forma aleatória e regular. Neste algoritmo, nosso objetivo é analisar a relação entre o número de monômeros de uma estrutura com o seu raio de giro; sabendo que o raio de giro está relacionado com as distâncias médias quadráticas de um conjunto de coordenadas dos resíduos;

- *na segunda parte*, um potencial estatístico, extraído de banco de dados de estruturas proteicas nativas, é usado para introduzir interações entre os aminoácidos, presentes numa dada sequência proteica. Nesse caso, o objetivo é o de tentar obter estruturas de baixa energia que sejam parecidas com a estrutura nativa correspondente daquela sequência primária. Dessa forma, nessa segunda parte, o objetivo consiste em, usando o algoritmo de crescimento de cadeias e um potencial estatístico, predizer a estrutura nativa de uma dada sequência de aminoácidos;

- finalmente, *na terceira parte*, as simulações são feitas numa cadeia completa, na qual, já de partida, todos os aminoácidos estão presentes ao se iniciar as simulações. O potencial é construído a partir da estrutura nativa e o objetivo aqui é o de verificar se as simulações na rede cubo-octaédrica fornecem estruturas “nativas”, em rede, que sejam bem pouco

distorcidas quando comparadas com a estrutura nativa presente no *protein data bank (PDB)*. Faremos também uma análise das propriedades termodinâmicas e cinéticas do estado de transição (CLEMENTI, 2000) da proteína CI2, código do PDB *I coa*, fazendo a contagem dos contatos nativos – Q – que são formados ao longo da simulação de minimização da energia potencial e, através deste, verificar quais são os contatos que são formados no estado de transição e que contribuem para o aumento da barreira da energia do estado de transição.

Realizamos simulações nesta última etapa a fim de determinar a temperatura, ou o parâmetro de temperatura β , na qual podíamos verificar que a estrutura realizava a transição entre o estado enovelado (alto valor de Q) e desenovelado (baixo valor de Q), (OLIVEIRA, *et. al.*, 2010).

Capítulo 2

2.1 Aminoácidos: blocos de construção das Proteínas

Os aminoácidos são as unidades estruturais fundamentais das proteínas. Eles são constituídos por um grupamento amina (H_3N^+), uma carboxila (COO^-), um átomo de hidrogênio e um radical R, cadeia lateral que caracteriza cada um dos 20 tipos de aminoácidos naturais e lhes confere especificidades, ligados a um átomo de carbono, que é chamado de carbono α por ser o adjacente ao grupamento carboxila (ácido), conforme mostramos na Figura 1:

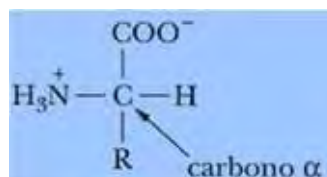


Figura 1: Fórmula geral ionizada de um aminoácido α . R representa a cadeia lateral.

As 20 diferentes cadeias laterais dos aminoácidos variam em tamanho, forma, carga, capacidade de formação de pontes de hidrogênio e reatividade química (STRYER, 1992). Estas características são cruciais no processo de formação de estrutura tridimensional da proteína. O aminoácido com cadeia lateral mais simples é a glicina, que tem somente um átomo de hidrogênio. A alanina vem em seguida, com um radical metil, (*ver Figura 2*). As cadeias laterais alifáticas maiores (com três e quatro carbonos) são hidrofóbicas. Nas representações abaixo, *figuras 2 a 8*, para as estruturas dos aminoácidos mostramos suas formas não ionizadas.

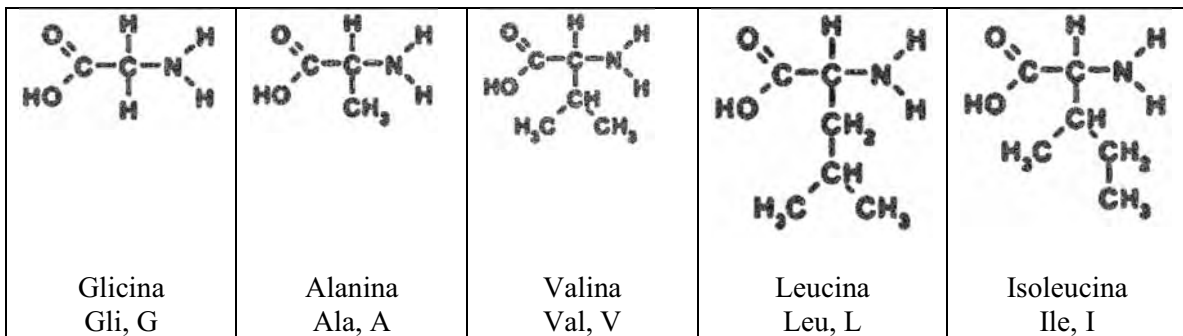


Figura 2: Aminoácidos com cadeias laterais alifáticas.

A prolina tem uma cadeia lateral alifática, porém ela se difere dos demais aminoácidos, pois sua cadeia lateral está ligada tanto ao carbono α quanto ao nitrogênio, (*Figura 3*). Esta estrutura cíclica influencia profundamente na arquitetura das proteínas.

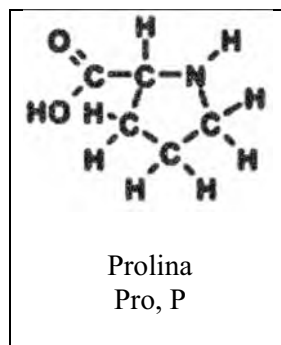


Figura 3: Representação estrutural do aminoácido Prolina.

Dentre os aminoácidos com cadeias laterais aromáticas, tem-se: a fenilalanina, que contém um radical fenil ligado a um radical metileno; o triptofano, que tem um anel indólico ligado ao radical metileno e a tirosina, que tem uma hidroxila ligada no anel aromático, conforme ilustra a *Figura 4*. A fenilalanina e o triptofano são altamente hidrofóbicos.

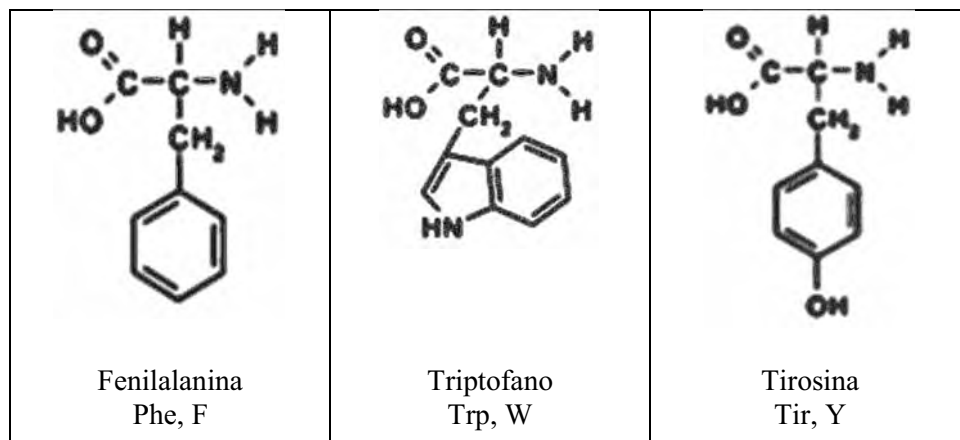


Figura 4: Aminoácidos com cadeias laterais aromáticas.

Um átomo de enxofre aparece na cadeia lateral da cisteína e da metionina, (*ver Figura 5*). Essas cadeias laterais sulfuradas são hidrofóbicas. O radical sulfidrila ($-SH$) da cisteína é altamente reativo, o que lhe confere um papel especial na determinação da estrutura de algumas proteínas por formar ligação dissulfeto.

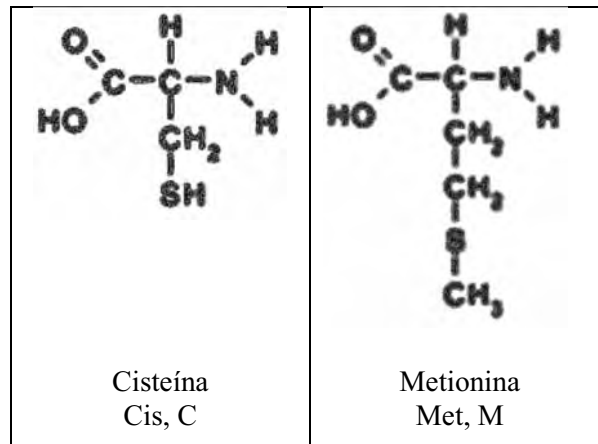


Figura 5: Aminoácidos com enxofre em suas cadeias laterais.

A serina e a treonina possuem um radical hidroxila ($-\text{OH}$) em suas cadeias laterais, (*Figura 6*). Pode-se considerar a serina como uma versão hidroxilada da alanina, e a treonina, uma versão hidroxilada da valina. Esses radicais hidroxilados tornam a serina e a treonina aminoácidos hidrofílicos, isto é, favorece para ficarem expostos na estrutura da proteína, fazendo ligações com as moléculas de água do meio em que se encontra a proteína.

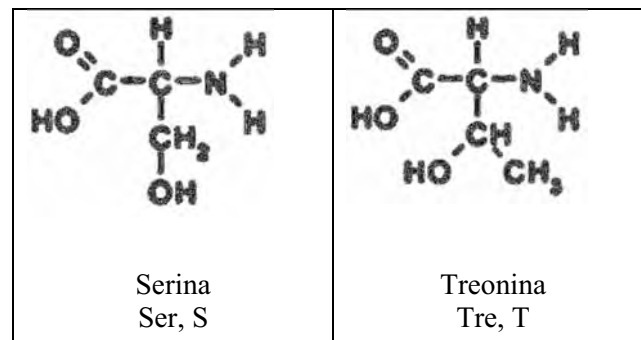


Figura 6: Aminoácidos que têm cadeias laterais alifáticas hidroxiladas.

Aminoácidos que possuem as cadeias laterais polares são altamente hidrofílicos, como a lisina e a arginina que têm carga positiva em pH neutro, (ver *Figura 7*). A histidina pode estar sem carga ou com carga positiva, dependendo do seu ambiente local. A histidina é

frequentemente encontrada nos centros ativos de enzimas, onde o anel imidazólico pode variar entre esses dois estados, para catalisar a formação e a ruptura de ligações.

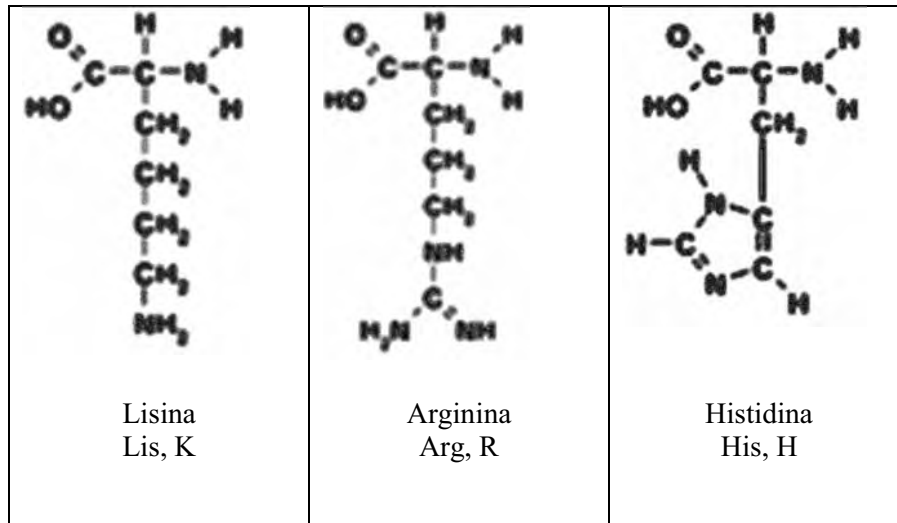


Figura 7: Aminoácidos com cadeias laterais básicas.

Finalizamos esta caracterização com as cadeias laterais ácidas, composta pelos aminoácidos: ácido aspártico e ácido glutâmico também conhecido como aspartato e glutamato, respectivamente, pois suas cadeias laterais estão quase sempre com carga negativa em pH fisiológico; a glutamina e asparagina, que são os derivados não carregados do glutamato e aspartato, e que contêm um grupo amida terminal em vez da carboxila.

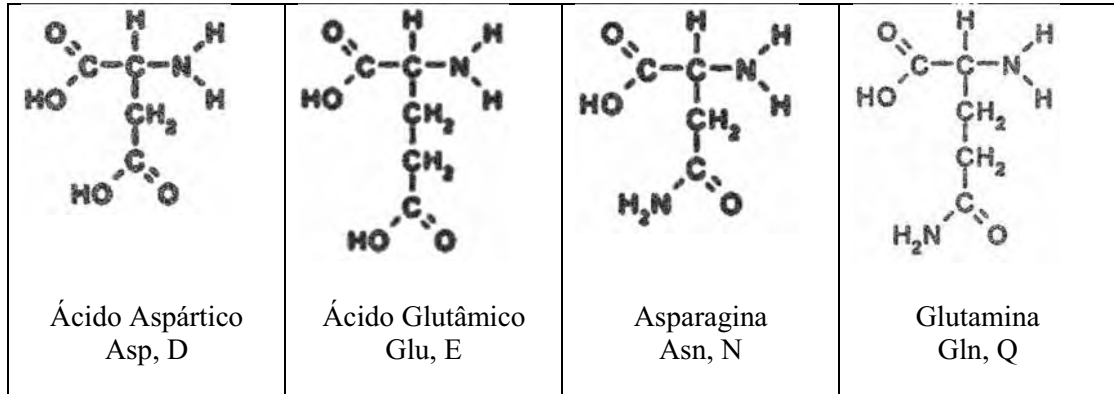


Figura 8: Aminoácidos com cadeias laterais ácidas, aspartato e glutamato, e suas amidas derivadas, asparagina e glutamina.

Os aminoácidos são frequentemente designados por uma abreviação de três letras ou por um símbolo com uma letra, que aparece sob cada um dos seus nomes nas figuras anteriores.

Sabemos que nas proteínas os aminoácidos são unidos por ligações peptídicas, formando cadeias polipeptídicas. A ligação peptídica é feita entre o grupamento carboxila de um aminoácido com o radical amina do outro, ocorrendo sempre a perda de uma molécula de água, (*ver Figura 9*). Os átomos que formam a ligação peptídica são chamados de grupo funcional peptídico. Uma unidade de aminoácido em um polipeptídio pode ser chamada de radical ou “resíduo de aminoácido”. A cadeia polipeptídica possui extremidades diferentes, por isso convencionou-se que a extremidade que possui o grupo amina é o início da cadeia, sendo escrita começando com o radical aminoterminal. Por exemplo, no tripeptídio, Ala-Lis-Asp (A-K-D), a alanina é o aminoterminal, enquanto que o aspartato é o carboxiterminal ou carbonoterminal. Sendo assim o tripeptídio D-K-A é diferente do anterior.

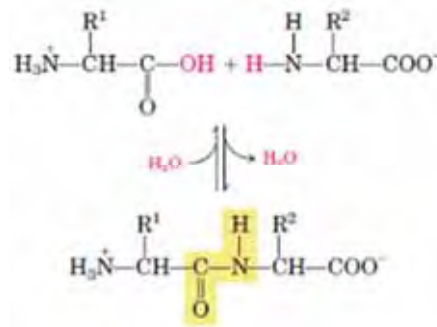


Figura 9: Representação esquemática da formação de uma ligação peptídica – destacada em amarelo na figura.

Além das interações eletrostáticas ou *coulombianas* entre os resíduos de aminoácidos outra forma de interação entre eles são as pontes dissulfeto. Uma ponte dissulfeto (–S–S–) é formada a partir de grupamentos sulfidrila (–SH) de duas cisteínas. O dissulfeto resultante é chamado de cistina.

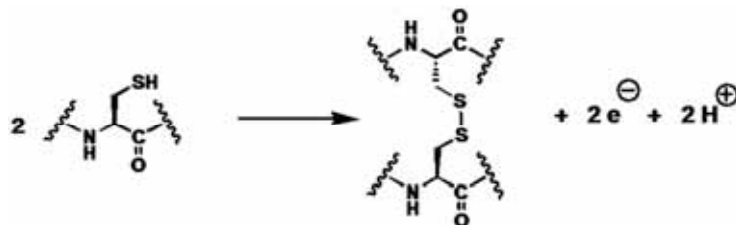


Figura 10: Ligação dissulfeto entre resíduos de cisteína.

2.2 Estruturas de Proteínas

***Proteína:** uma palavra estabelecida por Jöns J. Berzelius, em 1838, para salientar a importância dessa classe de moléculas. Derivada da palavra grega **proteios**, que significa “da primeira classe”.*

As proteínas exercem papéis cruciais nos processos biológicos. Sua importância e atividade são exemplificadas em algumas funções, tais como: catálise enzimática, transporte e armazenamento, movimento coordenado, sustentação mecânica, proteção imunitária, geração e transmissão de impulsos nervosos, controle do crescimento e da diferenciação (STRYER, 1992). Essa notável gama de funções exercidas pelas proteínas resulta da diversidade e da versatilidade dos seus 20 tipos de blocos de construção. Para desempenhar sua função biológica, as proteínas devem estar em seus estados nativos (conformação na qual a proteína existe em seu meio natural), com suas estruturas primária, secundária, terciária e, quando for o caso, quaternária, íntegras. Muitos esforços nas últimas décadas têm sido empregados na tentativa de compreender como as proteínas se enovelam e descrever quais são os mecanismos naturais que regem sua função. Apesar de tantos estudos científicos o problema do enovelamento proteico permanece como um dos mais fundamentais no ramo da biofísica molecular computacional.

O processo de perda da forma tridimensional de uma proteína é conhecido como desnaturação, que ocorre por ação de qualquer fator capaz de destruir as estruturas secundária,

terciária e/ou quaternária. Para várias proteínas a desnaturação é reversível. Sendo assim, na desnaturação proteica não há perda da estrutura primária, ou seja, os aminoácidos continuam unidos na mesma sequência. Porém, em determinadas situações a desnaturação é irreversível, como, por exemplo, a mudança de cor e de textura que observamos ao fritarmos um ovo. Os principais agentes físicos desnaturantes são o calor e a pressão. Numa reação química os átomos ou moléculas se repelem ou se atraem entre si sem que ocorra a quebra ou a formação de novas ligações químicas. Essas interações são frequentemente chamadas de interações não covalentes ou interações intermoleculares (ROCHA, 2001). Sendo assim, podemos desnaturar quimicamente uma proteína, dentre os desnaturantes químicos podemos destacar os ácidos e bases fortes, ureia e detergentes, como o *dodecil-sulfato de sódio* (SDS). Se as condições experimentais forem escolhidas de modo adequado, a conformação nativa da proteína poderá ser recuperada (renaturação). Os experimentos desse tipo fornecem evidências fortes de que a sequência de aminoácidos da proteína contém toda a informação necessária para a sua estrutura tridimensional completa (CAMPBELL, 2000). Baseados nessa ideia é que propomos este trabalho, no qual analisaremos o processo de enovelamento da proteína a partir de sua sequência de resíduos, isto é, a estrutura primária.

Basicamente podemos classificar as proteínas quanto à conformação em duas classes: proteínas fibrosas e proteínas globulares. As proteínas fibrosas são estruturalmente simples, a grande maioria das proteínas que pertencem a essa classe possui um único tipo de estrutura secundária (*folha β* ou *α -hélice*), e são geralmente insolúveis em água. Elas possuem forma de longos filamentos, entrelaçados como uma corda, gerando fibras altamente resistentes. Por esse motivo estão adaptadas para exercer a função estrutural. Exemplos: o colágeno (veja *Figura 11*), a queratina e a elastina.



Figura 11: Estrutura 3D da proteína colágeno, código PDB *2f6a*.

As proteínas globulares são estruturalmente mais complexas que as fibrosas porque contêm frequentemente, vários tipos de estrutura secundária, além de apresentar estruturas terciárias e quaternárias tais como hemoglobina e as enzimas, veja *Figura 12*. As proteínas globulares representam a maior classe das proteínas e encontram-se estruturalmente dobradas sobre si mesmas, apresentando formato esférico, ou seja, conformações compactas (CHAN & DILL, 1993). A estrutura secundária ocorre devido à possibilidade de rotação entre o carbono alfa (C_{α}) dos aminoácidos e seus grupamentos amina e carboxila, sendo assim a estrutura secundária é definida pelo arranjo espacial de aminoácidos próximos entre si na sequência primária. Já a estrutura terciária resulta do dobramento de α -hélices e folhas β , mantido pelas ligações de hidrogênio e pontes dissulfeto. Esta estrutura está intrinsecamente ligada à atividade biológica da proteína. Resumindo, a estrutura secundária é formada por interações de curto alcance, enquanto que a terciária é formada por interações de longo alcance entre os aminoácidos, como, por exemplo, as interações eletrostáticas entre resíduos que estão carregados eletricamente.



Figura 12: Estrutura quaternária da hemoglobina, código PDB 3qjb.

As proteínas desempenham também a maioria das funções específicas nas membranas. São elas que conferem as propriedades funcionais características de cada tipo de membrana. As proteínas de membrana podem ser: proteínas transmembrana (atravessam a bicamada lipídica e são anfipáticas) e proteínas periféricas (se prende à superfície interna ou externa da membrana plasmática). As proteínas transmembrana podem ser em α -hélice ou em arranjos de barril de folhas β , veja *Figura 13*.

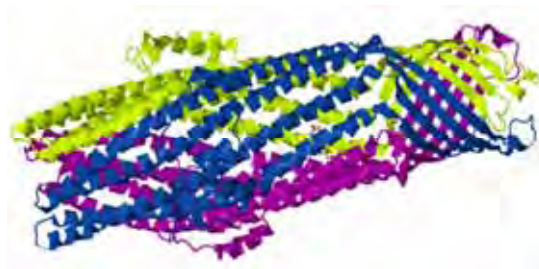


Figura 13: Estrutura de uma proteína de membrana, código do PDB 2wmz.

Capítulo 3

Metodologia

3.1 – Descrição da Rede Cubo-Octaédrica

O enovelamento de proteína é estabilizado por um balanço delicado de forças originadas através de diferentes tipos de interações. Podemos fazer uma classificação dessas forças em: local e não-local. Força de interação local envolve aquelas interações entre átomos ou resíduos que são sequencialmente adjacentes, e uma manifestação destas forças nas estruturas de proteínas são, por exemplo, a presença de α -hélice, onde as conformações dos aminoácidos são estabilizadas por ligação de hidrogênio dentro da cadeia. Enquanto que interações não-locais são mais difíceis de serem tratadas porque elas originam-se substancialmente em interações com o solvente. De uma forma geral, estas forças resultam na segregação dos aminoácidos carregados para o exterior da proteína e os aminoácidos apolares, frequentemente, estão no interior da proteína, devido a forças hidrofóbicas. Estudando estas

interações, RAGHUNATHAN e JERNIGAN (1997) verificaram que o empacotamento dos resíduos no interior da proteína é essencial para compreender e caracterizar a estabilidade da estrutura de proteínas no estado nativo. Nesta análise foi utilizado o modelo do átomo C_{α} para representar o aminoácido, assim todos os resíduos têm a mesma forma e tamanho. Essa simplificação nos permite ver a proteína como um homopolímero que ocupará os pontos de uma rede regular. Os autores observaram certa ordem e simetria, para todos os resíduos, na distribuição espacial dos vizinhos não-ligados (não-adjacentes na estrutura primária) em torno de um resíduo central, e isso os levaram a olharem para os princípios fundamentais da estrutura e organização das proteínas e a possibilidade de explorar e usar um conjunto de pontos para estudos de enovelamento. Sendo assim eles desenharam uma rede de pontos discretos no espaço tridimensional que melhor ajustasse a distribuição dos resíduos de aminoácidos de uma proteína, que foi a rede denominada cubo-octaédrica.

Um modelo de cela unitária desta rede é mostrado na Figura 14. Cada ponto da rede, por exemplo, o ponto em vermelho nesta figura, tem 12 vizinhos mais próximos, que são os pontos representados em azul. Esta unidade sólida semirregular é referida como cubo-octaédrica (RAGHUNATHAN *and* JERNIGAN, 1997), e tem 14 faces e 12 vértices. Ela é um único poliedro convexo e contém somente dois tipos de polígonos regulares: triângulos e quadrados. As 14 faces são compostas de oito triângulos e seis quadrados. Cada triângulo está cercado por três quadrados, e cada quadrado por quatro triângulos. Existem seis pontos no plano horizontal, três acima e três abaixo deste plano.

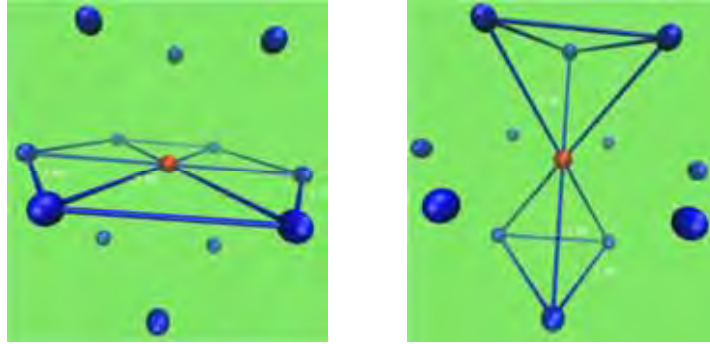


Figura 14 - Ilustração do modelo de uma cela unitária da rede adotada nas simulações. O ponto central, em vermelho, tem 12 vizinhos mais próximos, pontos em azul: (a) 6 no mesmo plano, (b) 3 acima e 3 abaixo deste plano. A distância entre 2 sítios vizinhos é de 3,8 Å.

Esta rede é uma versão da rede cúbica de face-centrada, que tem conexão permitida somente entre pontos de face-centrada e 12 extremidades mais próximas. Todas as conexões têm o mesmo comprimento, a saber, $\sqrt{2}$.

Os 12 vetores são,

no plano xy : $(1, 1, 0)$, $(1, -1, 0)$, $(-1, 1, 0)$, $(-1, -1, 0)$;

no plano yz : $(0, 1, 1)$, $(0, -1, 1)$, $(0, -1, -1)$ e

no plano zx : $(1, 0, 1)$, $(1, 0, -1)$, $(-1, 0, 1)$, $(-1, 0, -1)$.

Expressados desta forma, os vetores unitários são mais simples de serem entendidos. O conjunto de pontos na *Figura 14*, quando repetidos, formam um arranjo periódico de pontos da rede no espaço. Um número infinito de camadas pode ser crescido a partir do centro, na direção radial. O número N de pontos sobre a superfície de qualquer camada é dado pela relação:

$$N = 10.L^2 + 2 \quad (1)$$

sendo L é o número da camada.

Portanto, a primeira camada tem 12 pontos, a segunda 42, a terceira 92, e assim por diante. Este modelo oferece umas das melhores formas para inspecionar visualmente traços na cadeia da proteína, bem como estereoquímica da cadeia lateral em relação à cadeia principal, a estrutura secundária e outros detalhes geométricos de estruturas de proteínas. Quando conectados aos pares, estes pontos formam ângulos de ligação 0° , 60° , 90° , 120° e 180° , veja *Figura 15*.

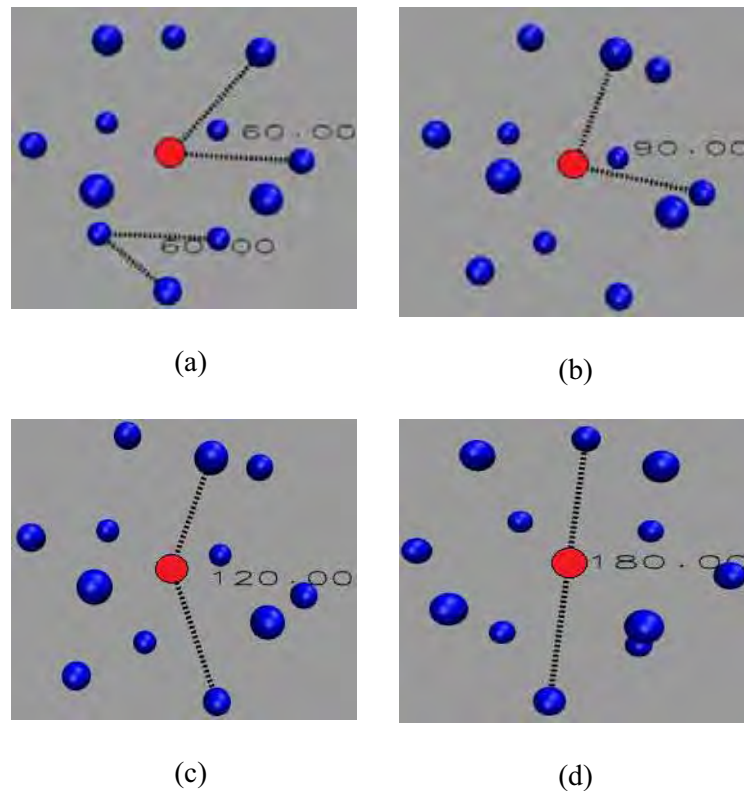


Figura 15: Ilustração dos ângulos permitidos na rede. O ponto em vermelho é o sítio central e faz um ângulo de (a) 60° , (b) 90° , (c) 120° e (d) 180° com outros dois sítios.

A escolha de um modelo de rede/proteína representa um balanço entre a precisão dos resultados atingidos e o custo computacional. As geometrias das redes utilizadas para simular as estruturas de proteínas variam de modelos de baixa e alta resolução. Exemplos de modelos de redes de baixa resolução são a rede cúbica simples (COVELL and JERNIGAN, 1990) e

diamante (HINDS, 1992). Redes de baixa resolução são mais eficientes em amostragem do espaço conformacional, é possível até mesmo a enumeração exata de todas as conformações compactas de pequenas cadeias. Mas redes de alta resolução são necessárias para reproduzir mais precisamente elementos da estrutura secundária e terciária. Nessas simulações cada resíduo é representado por um sítio de interação e os resíduos interagem via um potencial de dois corpos proposto por THOMAS and DILL (1996). Na próxima seção descreveremos como realizamos o método de crescimento de estruturas proteicas utilizando a rede cubo-octaédrica.

3.2 Algoritmo de Crescimento de Estruturas Proteicas em Rede: Descrição do Método Estatístico

Inicialmente procuramos realizar o método de crescimento estatístico de estruturas proteicas, proposto por ROSENBLUTH e ROSENBLUTH (1954) na rede cubo-octaédrica e posteriormente estudamos a cinética de formação dessas estruturas realizando movimentos de *Monte Carlo* segundo o critério de minimização de energia. Nas simulações computacionais descrevemos a estrutura primária da proteína usando o modelo simplificado de proteínas conhecido como modelo C_{α} , no qual a cadeia lateral do aminoácido é representada por um único sítio da rede. A distância entre dois sítios vizinhos da rede é suficiente para englobar todos os comprimentos das cadeias laterais dos aminoácidos (RAGHUNATHAN e JERNIGAN, 1997). Para um dado número N de resíduos da estrutura proteica, teremos $N-1$ ligações, que no algoritmo de crescimento, consistem de um conjunto de sítios da rede ligados em sucessão. Se considerarmos o crescimento de estruturas proteicas como um problema de *random walk*, em um dado estágio de m ligações finalizando na posição $(x,y,z)_N$, todas as 12 posições da rede seriam, *a priori*, igualmente provável para a próxima ligação, $m+1$. Sendo

assim, esse método consiste em resolver um problema de *random walk* na rede cubo-octaédrica sujeito às restrições:

- Efeito de volume excluído: é representado no algoritmo pelo critério que a cadeia não pode cruzar sobre si mesma ou dobrar sobre si mesma, evitando assim a sobreposição de dois ou mais resíduos em um mesmo sítio da rede;
- As ligações com ângulo de 180° entre os resíduos i e $i-2$ são proibidas, uma vez que esta conformação gera um estresse na estrutura, dificultando o movimento de rotação dos resíduos.

Representamos na *Figura 16* três estruturas com 20 resíduos de aminoácidos. Em (a) a estrutura foi gerada obedecendo aos critérios apresentados acima; destacamos na *Figura 16* (b) um trecho da estrutura que forma um ângulo de 180° entre os resíduos 7, 8 e 9; por fim, a estrutura apresentada em (c) possui uma superposição dos resíduos 10 e 12, caso que é comum acontecer no processo de crescimento de estruturas quando o algoritmo não possui o critério de volume excluído.

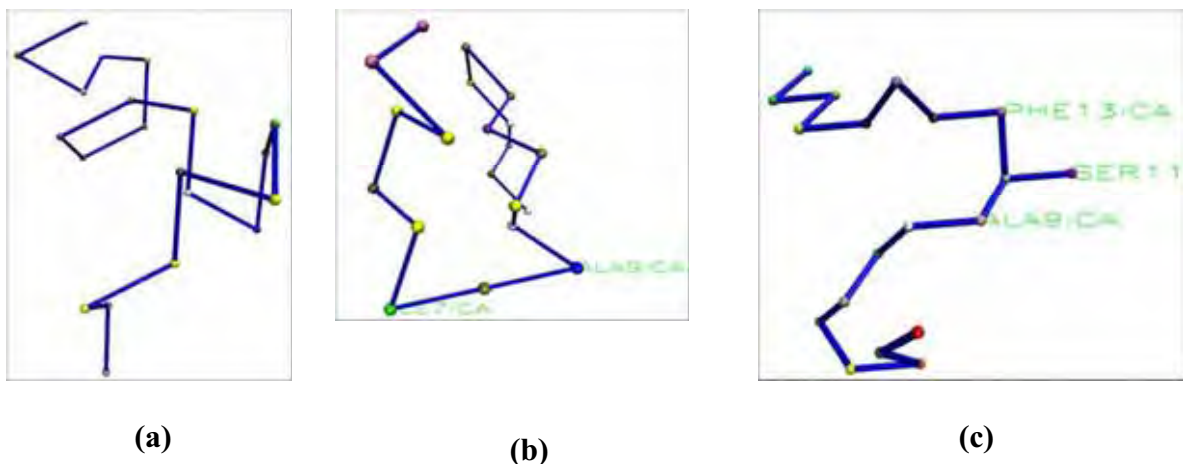


Figura 16: Exemplo de uma estrutura proteica de 20 resíduos: (a) permitida e duas estruturas não permitidas com (b) ângulo obtuso, 180° e (c) com superposição de dois resíduos.

Consequentemente, em um dado estágio do crescimento existirá no máximo 11 sítios disponíveis para a posição subsequente. O raio de giro descreve a dispersão espacial da molécula e é definido como a raiz quadrática média da distância do conjunto de átomos em relação a um centro de gravidade comum. Segundo JAMAYARAM (2010) fazendo-se uma análise para um conjunto de 1.000 proteínas globulares selecionadas do *Protein Data Bank* (PDB) o raio de giro dessas proteínas varia com uma lei de escala, de aproximadamente:

$$R_{\text{giro}} = 0,395 \cdot N^{3/5} + 7,257 \quad (2)$$

onde N é o número de aminoácidos.

Neste trabalho realizamos um passeio aleatório (*random walk*) através de um procedimento de Monte Carlo (MC) proposto por ROSENBLUTH and ROSENBLUTH (1954). Um conjunto de milhares de estruturas foi gerado aleatoriamente de acordo com os critérios abaixo:

- (i) O primeiro resíduo é fixado no sítio (0,0,0), em seguida escolhe-se aleatoriamente um dos 12 vizinhos mais próximos da rede para ser a posição do segundo resíduo, formando-se assim a primeira ligação, $m = 1$.
- (ii) Para um dado conjunto de m ligações, partindo da origem e alcançando a posição $(x, y, z)_m$ está associado uma função peso (W_m) calculada em cada passo de acordo com o critério (iii). A função peso é necessária uma vez que algumas configurações são geradas mais frequentemente do que outras, assim a função peso é introduzida a fim de que todas as configurações sejam contadas igualmente. Após $N-1$ ligações serem obtidas pelo critério (iii), veja logo abaixo, com a posição final $(x, y, z)_N$ e peso W_{N-1} , o valor do raio de giro é pesado com W_{N-1} . Matematicamente temos,

$$R_{giro}^2 = \frac{1}{N^2} \sum_{i>j}^N R_{i,j}^2 \quad (3)$$

sendo, $R_{i,j}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2$. Para um *ensemble* de n estruturas com o mesmo número N de resíduos de aminoácidos, calcula-se o raio de giro médio destas estruturas e também o peso estatístico médio através das equações abaixo:

$$\langle R_{giro_N} \rangle = \frac{\sum_{i=1}^n R_{giro_{N,i}} \cdot W_{N,i}}{\sum_{i=1}^n W_{N,i}} \quad (4)$$

$$\overline{W} = \frac{\sum_{i=1}^n W_{N,i}}{n} \quad (5)$$

- (iii) Como mencionado anteriormente, em um dado estágio do crescimento com m ligações, atingindo a posição $(x, y, z)_m$, os 12 sítios da rede devem ser levados em consideração. No entanto, um desses 12 sítios está ocupado formando a ligação $(x, y, z)_{m-1}$, assim este sítio deve ser descartado imediatamente. Desta forma existem três possibilidades para a obtenção do peso estatístico W_{m+1} :
- Todos os 11 novos sítios estão ocupados, neste caso, o processo de crescimento é encerrado com um peso $W = 0$ e uma nova estrutura deve ser crescida, iniciando novamente em $(0, 0, 0)$.
 - Todos os 11 novos sítios estão desocupados, então o peso estatístico será:

$$W_{m+1} = W_m (W_1 = 1) \quad (6)$$

E assim, um desses 11 sítios é selecionado aleatoriamente fornecendo a ligação $(x, y, z)_{m+1}$.

- Somente n dos novos sítios estão desocupados, com $0 < n < 11$, então o peso estatístico será:

$$W_{m+1} = \frac{n}{11} \cdot W_m (W_m = 1) \quad (7)$$

Novamente, um número aleatório é gerado para selecionar um desses n sítios e assim completar a ligação $(x, y, z)_{m+1}$.

Ilustramos na Figura 17 um passo no processo de crescimento. O ponto em vermelho representa o resíduo i . Para adicionarmos o próximo resíduo na estrutura, $i+1$, do conjunto de doze sítios mais próximos somente oito estão desocupados, pois os quatro pontos roxos já estão ocupados; assim o algoritmo escolhe aleatoriamente um dos oitos desocupados, por exemplo, o ponto verde. No próximo capítulo apresentaremos os resultados deste método de crescimento de estruturas proteicas.

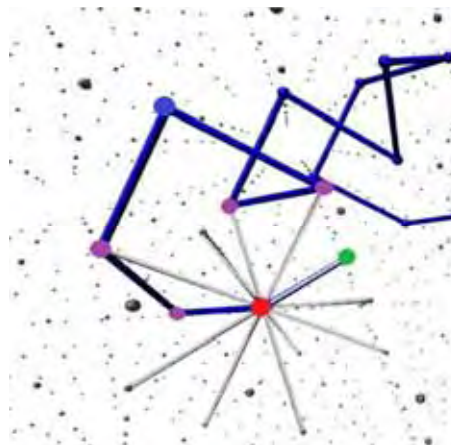


Figura 17: Ilustração de um passo no processo de crescimento. O ponto vermelho representa o resíduo i e o verde o $i+1$. Em um dado estágio i do crescimento são excluídos os sítios que já estão ocupados e aquele que faz um ângulo de 180° com o resíduo $i-2$. Nesta figura seriam excluídos 5 sítios, logo este estágio do crescimento teria apenas 7 sítios disponíveis dos 12 sítios que a rede contém.

3.3 Algoritmo de Predição de Estruturas Proteicas com Potencial Estatístico de Interação entre os Resíduos de Aminoácidos

Ao método de crescimento de estruturas proteicas, descrito na seção anterior, fizemos uma modificação acrescentando a ele uma probabilidade de transição dependente da temperatura para determinar qual será a nova posição para ser o sítio de interação da estrutura (GAN, 2000). Uma contribuição importante desse esquema de crescimento é que a probabilidade de transição para o crescimento das ligações é influenciada por um peso estatístico de Boltzmann, dependente da interação energética entre os aminoácidos, descrita a partir de um potencial estatístico. Desta forma, poderemos gerar configurações que contribuem significativamente para a média termodinâmica numa dada temperatura. Por exemplo, em baixas temperaturas observa-se a formação de configurações compactas mais frequentemente do que configurações abertas; o inverso ocorre em altas temperaturas. Vale lembrar que neste modelo simplificado de proteína, os sítios de interações são localizados nas posições dos C_α e a cadeia lateral não é levada em consideração. A reprodução das características geométricas da cadeia polipeptídica foi inserida no algoritmo pela restrição do ângulo de 180° que é permitido pela rede; bem como o critério de volume excluído, para

evitarmos a sobreposição de dois ou mais resíduos no mesmo sítio da rede. Para cada par de resíduo não consecutivo i e j separados por distância R_{ij} , o potencial de interação entre eles tem uma extensão finita, de forma que:

$$u_{ij}(R_{ij}) = \begin{cases} \varepsilon_{ij} & \text{se } R_{ij} \leq 7,0 \text{ \AA} \\ 0 & \text{se } R_{ij} > 7,0 \text{ \AA} \end{cases} \quad (8)$$

sendo ε_{ij} o potencial estatístico de interação entre pares de resíduos de aminoácidos, (ver *Tabela 1*). Potenciais derivados de banco de dados de proteínas são quantidades que representam a energia de interação entre os aminoácidos, sendo amplamente utilizados em algoritmos computacionais que têm como propósito, de alguma forma, predizer a estrutura da proteína a partir de sua sequência de aminoácidos (THOMAS *and* DILL, 1990). Esses potenciais de interação foram extraídos na observação da frequência de pareamento ρ_{ij} do aminoácido do tipo i com o aminoácido tipo j e tomando o logaritmo para obter as quantidades energéticas adimensionais (VENDRUSCOLO, 2000), ε_{ij} :

$$\varepsilon_{ij} = \frac{\text{energia}}{kT} = -\ln\left(\frac{\rho_{ij}}{\rho_{ij}^*}\right) \quad (9)$$

sendo ρ_{ij}^* representa a frequência de pareamento no estado de referência, k é a constante de Boltzmann e T é a temperatura absoluta.

Tabela 1: Potencial baseado na probabilidade de contato para os 20 aminoácidos.

	C	M	F	I	L	V	W	Y	A	G	T	S	Q	N	E	D	H	R	K	P	
C	-1.79	-1.23	-0.98	-0.48	-0.69	-0.94	-0.3	-0.96	-0.3	-0.42	-0.38	-0.2	-0.49	-0.32	0.04	0.55	-0.82	-0.4	0.00	0.07	
M		0.36	-1.03	-0.41	-0.31	-0.94	-0.07	-1.1	0.05	0.00	0.06	-0.47	-0.54	0.31	0.02	1.07	-0.35	-0.43	0.55	-0.25	
F			-0.61	-0.66	-1.02	-0.78	-0.89	-0.82	-0.05	0.21	-0.19	0.14	0.1	-0.02	0.19	0.2	-1.75	-0.22	-0.17	-0.43	
I				-0.71	-1.04	-0.98	-0.89	-0.87	-0.64	0.4	-0.29	-0.13	-0.39	0.39	-0.2	0.04	-0.52	-0.08	-0.26	0.25	
L					-1.14	-1.03	-0.97	-0.6	-0.57	-0.08	-0.39	-0.07	-0.13	-0.1	-0.05	0.5	-0.36	-0.1	0.1	0.09	
V						-1.15	-0.6	-0.7	-0.6	-0.2	0.06	-0.31	-0.09	-0.24	-0.02	0.25	-0.35	-0.48	-0.08	-0.08	
W							0.02	-0.99	-0.08	-0.14	0.07	-0.2	0.4	-0.68	0.32	0.24	-0.41	-0.78	-0.3	-0.44	
Y								0.35	-0.37	-0.32	-0.23	0.25	-0.39	-0.74	0.22	0.11	-0.67	0.21	-0.2	-0.45	
A									-0.08	-0.09	-0.22	-0.01	-0.11	-0.14	0.03	0.1	-0.15	0.07	0.00	0.41	
G										0.04	0.13	-0.04	0.12	-0.18	0.4	-0.06	0.00	0.15	0.1	0.4	
T											0.26	0.05	-0.17	-0.27	0.15	-0.03	-0.27	-0.17	0.09	0.36	
S												-0.13	0.4	0.37	0.3	-0.09	-0.59	0.61	0.18	0.44	
Q													-0.08	-0.05	0.62	0.46	0.05	0.62	0.04	-0.21	
N														-0.86	-0.25	-0.12	0.06	0.04	0.18	0.11	
E															0.21	0.68	-0.53	-0.26	-0.09	0.33	
D																0.6	-0.06	-0.15	-0.09	0.84	
H																	0.14	-0.01	0.14	-0.22	
R																		0.23	0.3	-0.02	
K																			1.45	0.51	
P																					0.28

Essencialmente, o algoritmo de crescimento de cadeia gera conformações pela adição sequencial de ligações até que o comprimento total da cadeia seja alcançado. Desde que cada configuração seja gerada *de novo*, as configurações são independente estatisticamente. Para cadeias em que as propriedades são independentes da temperatura, todas as possíveis direções, sujeito à condição de volume excluído, têm a mesma probabilidade de transição. Para descrever um polímero dependente da temperatura e do volume excluído, MEIROVITCH, (1983), utilizou a probabilidade de transição com peso de Boltzmann. Essa probabilidade de transição além de favorecer a direção de crescimento com contatos de baixa energia permite gerar configurações de cadeias abertas e fechadas, dependendo da temperatura a ser observada.

Considere uma cadeia molecular com $N+1$ sítios de interação, num determinado passo i , o processo de crescimento será governado pela probabilidade de transição P_i , que especifica a probabilidade de selecionar uma dada direção para a próxima ligação, supondo que já foram selecionadas $i-1$ ligações. A nova direção não deve sobrepor-se com qualquer outro sítio de interação, uma vez que a cadeia deve obedecer à condição de volume exclusivo. Portanto, a probabilidade de transição dependente da temperatura no passo i será dada por:

$$P_i(\vec{R}_i + \vec{v}_{n_i} | \vec{R}_1, \dots, \vec{R}_i; \beta) = e^{-\beta u_i(\vec{R}_i + \vec{v}_{n_i})} / \sum_{n_i=1}^{C_i} e^{-\beta u_i(\vec{R}_i + \vec{v}_{n_i})} \quad (10)$$

sendo $u_i(\vec{R}_i + \vec{v}_{n_i})$ a energia potencial de interação entre pares de resíduos não ligados covalentemente dada por:

$$u_i(\vec{R}_i + \vec{v}_{n_i}) = \sum_{j=1}^{i-1} u_{ij}(R_{ij}) \quad (11)$$

β é um parâmetro de temperatura $\beta = \frac{1}{k_B T}$;

$\vec{R}_1, \vec{R}_2, \dots, \vec{R}_i$ são os vetores dos sítios de interação $1, 2, \dots, i$;

\vec{v}_{n_i} é o vetor da rede na direção selecionada;

C_i é o número de sítios disponíveis para o passo i , e

$u_{ij}(R_{ij})$ é o potencial de interação dependente da distância R_{ij} entre os sítios i e j .

A probabilidade deve ser normalizada, logo:

$$\sum_{n_i=1}^{C_i} P_i(\vec{R}_i + \vec{v}_{n_i}; \beta) = 1 \quad (12)$$

Vimos anteriormente que um dado sítio da rede cubo-octaédrica possui 12 vizinhos mais próximos, sendo assim podemos escrever que o número de coordenação da rede é $n_c = 12$; para este algoritmo de crescimento de cadeias com volume excluído teremos $C_i = n_c - 1$, e para cadeias ideais cuja sobreposição seja permitida $C_i = n_c$. Verifica-se também que a probabilidade de transição dependente da temperatura, equação

$$10P_i(\vec{R}_i + \vec{v}_{n_i} | \vec{R}_1, \dots, \vec{R}_i; \beta) = e^{-\beta u_i(\vec{R}_i + \vec{v}_{n_i})} / \sum_{n_i=1}^{C_i} e^{-\beta u_i(\vec{R}_i + \vec{v}_{n_i})} \quad ($$

10, favorece os estados configuracionais com altos pesos de Boltzmann e/ou a direção de crescimento com contatos energéticos favoráveis. As configurações geradas dessa forma serão compactas em baixa temperatura e mais abertas em altas temperaturas.

No processo de crescimento de estruturas na rede cubo-octaédrica, a primeira ligação pode ser colocada em qualquer direção, uma vez que todos os 12 sítios terão o mesmo peso na probabilidade de transição. As direções dos movimentos subsequentes, determinando as próximas ligações da estrutura, para um determinado sítio calculam-se todas as possibilidades de crescimento de acordo com o número de sítios disponíveis; ordenam-se, em ordem decrescente, as probabilidades de transição desses sítios e seleciona-se através de procedimento de Monte Carlo, um dos mais prováveis por comparação com uma variável aleatória $x \in [0;1]$. Este procedimento de crescimento será repetido até atingir o comprimento total da cadeia de N ligações ($N+1$ resíduos). Cada configuração Λ é gerada com uma probabilidade,

$$P_\Lambda(\beta) = \prod_{i=1}^N P_i(\vec{R}_i + \vec{v}_{n_i} | \vec{R}_1, \dots, \vec{R}_i; \beta)$$

$$P_{\Lambda}(\beta) = e^{-\beta E_{\Lambda}} / W_{\Lambda}(\beta) \quad (13)$$

sendo E_{Λ} a energia da configuração Λ e o peso estatístico

$$W_{\Lambda}(\beta) = \prod_{i=1}^N \left\{ \sum_{n_i=1}^{C_i} e^{-\beta u_i(\vec{R}_i + \vec{v}_{n_i})} \right\} \quad (14)$$

Caso o algoritmo chegue num estágio em que $C_i = 0$, ou seja, não há sítios disponíveis para continuar o crescimento, o processo é interrompido, descartando-se essa configuração, e uma nova configuração será gerada. Dessa forma o ensemble de estruturas geradas para um mesmo número N , não são correlacionadas.

Uma vez que as configurações foram geradas através da probabilidade de transição com pesos estatísticos a média do raio de giro do ensemble estrutural será dada por:

$$\langle R_{giro} \rangle = \frac{\sum_{\Lambda} R_{\Lambda} \cdot W_{\Lambda}}{\sum_{\Lambda} W_{\Lambda}} \quad (15)$$

sendo R_{Λ} o raio de giro para a configuração Λ .

Capítulo 4

4.1 Estudo do Estado de Transição com o Potencial de Interação Dependente da Distância entre os Resíduos de Aminoácidos da Estrutura Nativa

Neste capítulo descreveremos uma terceira etapa realizada neste trabalho de doutorado; trata-se de um algoritmo computacional que escrevemos para tentar encontrar uma

estrutura proteica, na rede, que tivesse energia de contato entre pares de resíduos, não ligados covalentemente, próximo ou igual à estrutura nativa.

Este tipo de algoritmo é conhecido na literatura como algoritmo de minimização de energia, pois um dos critérios para ele movimentar um dado resíduo é a verificação se a energia da nova posição será menor do que a da posição atual (VENDRUSCOLO, 2000).

Neste algoritmo utilizamos como dados de entrada:

- uma estrutura de 64 resíduos (código PDB: *Icoa*) crescida pelo algoritmo de crescimento descrito na seção anterior;
- uma rede crescida até a trigésima camada, totalizando 94.611 sítios. Inicialmente o algoritmo lê a estrutura de entrada e marca os sítios da rede como ocupados, $rede(x,y,z) = 1$;
- mapa de contatos nativos (veja a descrição de “contatos nativos” abaixo);
- parâmetro de temperatura β , que pode ser fixo ou variável;
- uma variável para contar o número de passos – movimentos realizados na estrutura.

Utilizamos um potencial baseado no mapa de contatos nativos, extraído a partir da estrutura nativa do PDB. Dizemos que dois aminoácidos estão em contato quando eles estão distantes na estrutura primária, mas estão relativamente próximos um do outro na estrutura terciária. Utilizamos para este teste o mapa de contatos nativos da proteína *Icoa*, sendo este mapa uma matriz de distâncias para cada resíduo com todos os seus vizinhos mais próximos na estrutura 3D, ver *Tabela 2*. A partir dessa matriz de contatos define-se a energia dependente das distâncias entre os pares de resíduos i e j como sendo -1 para os resíduos que estão em contato e 0 para aqueles que não fazem contato. Ao todo, esta proteína possui 136 contatos, desta forma podemos dizer que a estrutura nativa tem uma energia mínima de $-136,0$

unidades. Devemos tomar um cuidado quando iremos verificar essas distâncias de contatos no cálculo da energia. Por exemplo, se o resíduo 1 estiver em contato com o 24 (sendo $0,8D_{nat(1,24)} < D_{1,24} < 1,2D_{nat(1,24)}$) durante a dinâmica o algoritmo também deve contar o contato do 24 com o 1, logo, $E_{1,24} = E_{24,1} = -1$.

Tabela 2: Mapa de contatos nativos da proteína lcoa (64 aminoácidos).

Aminoácido	Aminoácido	Distância (Å)
1	24	7,63479
1	26	11,87266
1	44	11,02642
2	7	12,27184
2	23	7,23028
2	24	7,76383
2	63	8,58569
3	42	12,71629
3	62	5,37401
3	63	5,21929
4	60	6,8141
4	61	5,64225
4	62	4,3917
5	19	8,70402
5	20	7,84187
5	23	9,84246
5	24	11,09883
5	47	10,0093
5	61	5,37792
5	63	9,38691
6	60	5,04579
7	11	8,44222
7	19	9,7699
8	15	8,55815
8	16	6,01872
8	19	8,39351
8	20	9,2816
8	57	6,62541
8	61	7,39067
9	57	5,09186
9	58	5,45866
9	59	5,58802
9	60	5,89983
10	56	5,33151
10	57	5,14509
11	15	6,57655
11	16	6,29825
11	19	10,43609
11	56	5,28451
11	57	5,59911
12	16	6,10475
12	55	6,14394
12	56	4,26708
12	57	6,05979
13	17	5,96758
13	31	7,94663
13	49	7,74603
13	51	7,92231
13	55	6,33396
13	57	6,46893
14	18	6,51575
14	55	9,37353
15	19	6,59674
16	20	6,22776
16	49	8,49859
16	57	7,54453
17	21	6,01997
17	29	7,34037

17	31	10,1581
17	49	9,2371
18	22	6,13629
19	23	6,3664
20	24	5,5468
20	27	7,26079
20	29	8,47384
20	47	8,23499
20	49	10,48675
20	61	9,1766
21	25	4,78299
21	27	6,31015
21	29	8,87451
24	44	10,46026
24	45	8,24106
24	47	9,31553
24	63	9,7699
26	45	6,26291
27	45	5,04777
27	47	6,28912
28	43	9,37251
28	45	6,02046
28	46	4,69649
28	47	6,002
29	47	4,84149
29	49	7,06994
30	46	7,33253
30	47	6,19581
30	48	4,59587
30	49	5,94718
31	49	4,72388
31	51	6,97273
31	55	8,74643
32	36	6,75796
32	38	9,54521
32	48	7,13835
32	49	6,04955
32	50	4,93184
32	51	6,18587
33	50	5,23288
33	51	4,90326
34	50	4,77954
34	51	5,48899
34	52	6,20645
34	58	7,15339
34	59	8,10407
35	50	7,95908
35	59	10,70831
36	50	8,28662
38	48	10,09351
38	50	11,08057
39	48	12,48579
41	46	10,5119
41	48	12,2614
41	64	7,69025
42	63	8,51657
42	64	4,86528
43	64	4,6983
44	63	5,99258
46	63	5,57934
46	64	5,45197
47	61	6,7161
47	62	5,97947
47	63	4,6653
48	61	6,26817
48	62	6,8325
48	64	7,39013
49	57	6,30714
49	61	5,65349
50	57	5,88073
50	58	5,86259
50	59	6,10827
51	55	4,95792
51	56	5,21795
51	57	4,54368
52	56	5,54905

52	58	5,76689
57	61	7,88835

Para uma cadeia com N resíduos de aminoácidos, inicialmente seleciona-se aleatoriamente um resíduo, i , e calcula a energia conformacional da estrutura nesse estado:

$$E_{i(ola)} = \sum_{j=1}^N u_{i,j}(R_{i,j}) \quad (16)$$

sendo,

$$u_{i,j}(R_{i,j}) = \begin{cases} -1, & \text{se } 0,8D_{nat(i,j)} < D_{i,j} < 1,2D_{nat(i,j)} \\ 0, & \text{para valores fora deste intervalo} \end{cases} \quad (17)$$

Se i for um dos resíduos da extremidade ($i=1$ ou $i=N$) as condições a serem verificadas para realizar os movimentos são:

- verificar se existe um dos 12 vizinhos da rede que estão desocupados, $rede(x,y,z) = 0$, se não, seleciona-se outro monômero;
- eliminar os sítios que fazem ângulo de 180° , para isso verifica se o produto escalar triplo $\vec{A} \cdot (\vec{B} \times \vec{C}) = 0$ e a $Dist_{i,i+2} \geq 2\sqrt{2}$;

$$E_{i(new)} = \sum_{j=1}^N u_{i,j}(R_{i,j})$$

(18.

Para os monômeros do meio da cadeia ($1 < i < N$), além das condições citadas acima, deve-se calcular o ângulo que o monômero está fazendo com os seus vizinhos ligados covalentemente, pois para mantermos as ligações covalentes em $3,8 \text{ \AA}$ os vizinhos desocupados devem fazer o mesmo ângulo que o monômero faz na conformação original; caso essa distância não seja mantida teremos distorções na estrutura, divergindo do fato que observamos experimentalmente – onde as distâncias médias entre os C_α são de $3,8 \text{ \AA}$. A

estrutura foi crescida podendo ter somente ângulos de 60°, 90° e 120°, logo estes ângulos devem ser mantidos constantes durante a dinâmica. Calculamos estes ângulos através do produto escalar entre os vetores que descrevem a posição dos vizinhos adjacentes em relação ao monômero i :

$$\vec{A}_{i,i-1} \cdot \vec{B}_{i,i+1} = A_{i,i-1} B_{i,i+1} \cos\alpha$$

Para cada um dos sítios disponíveis, e que satisfazem as restrições acima, calcula-se novamente a energia conformacional:

$$E_{i(new)} = \sum_{j=1}^N u_{i,j}(R_{i,j}) \quad (18)$$

sendo a energia $u_{i,j}(R_{i,j})$ dada pela (17).

Ressaltamos que na rede cubo-octaédrica um sítio da rede tem 12 vizinhos mais próximos; destes, o monômero i já tem 2 sítios ocupados pelos monômeros $i+1$ e $i-1$, logo, ele poderá ter no máximo 10 sítios desocupados para se deslocar. Após ter sido satisfeitas as condições acima, o movimento será aceito desde que obedeça ao seguinte critério:

- Se $E_{i(new)} < E_{i(old)}$: aceita o movimento.
- Se $E_{i(new)} > E_{i(old)}$, calcula-se a probabilidade de aceitação dada por:

$$P = e^{-\beta \cdot \Delta E} \begin{cases} P > n, & \text{aceita o movimento.} \\ P < n, & \text{descarta o movimento.} \end{cases} \quad (19)$$

sendo que $\Delta E = E_{i(new)} - E_{i(old)}$, β é o parâmetro de temperatura e n é um número aleatório calculado pelo algoritmo.

Estes movimentos se repetem várias vezes, ao todo 1.10^7 passos de Monte Carlo, e para cada nova estrutura calcula-se a energia, o raio de giro e o desvio médio quadrático (*RMSD*, sigla em *inglês*).

Capítulo 5

Resultados

5.1 Método de Crescimento de Estruturas sem Interação

Como descrito na seção 3.2, apresentaremos agora os resultados do método de crescimento de estruturas sem interação que utiliza pesos estatísticos em cada passo do processo de crescimento (ROSENBLUTH and ROSENBLUTH, 1954). Para o cálculo do raio de giro médio do ensemble de estruturas com um valor fixo de resíduos calculamos um conjunto de 10.000 estruturas. Já mencionamos que a função peso é necessária porque algumas configurações são geradas mais frequentemente do que outras, e assim a função peso é introduzida para que todas as configurações sejam contadas igualmente. A fim de verificarmos essa importância do peso estatístico na contagem das estruturas para calcular grandezas físicas médias, fizemos o crescimento de estruturas com e sem o peso estatístico.

Para o ensemble de estruturas geradas com o peso estatístico, *Figura 18*, verificamos através de um gráfico *di-log* do raio de giro versus o número de resíduos que a reta ajustada apresentou um coeficiente angular de $0,580 \pm 0,009$.

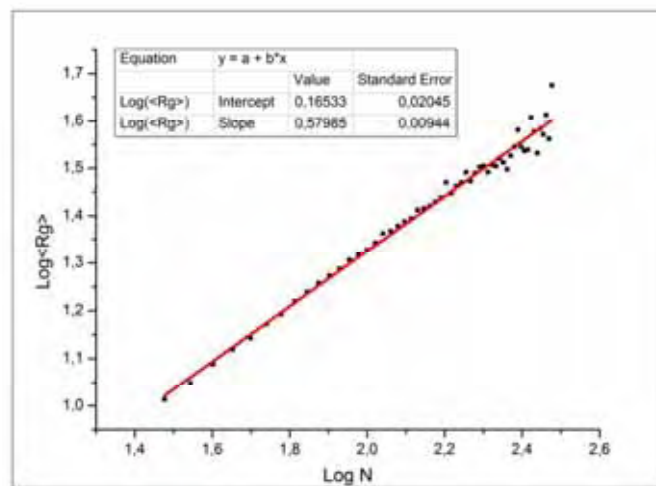


Figura 18: Valores do raio de giro médio versus N em um gráfico *di-log* para as estruturas geradas pelo algoritmo de crescimento com peso estatístico W_N . A inclinação da curva ajustada é de $0,58 \pm 0,01$.

Para o ensemble de estruturas geradas como um simples passeio aleatório, *Figura 19*, verificamos através do gráfico *di-log* do raio de giro versus o número de resíduos que a lei de escala obtida empiricamente em que o $R_{giro} \approx N^{3/5}$ (FLORY, 1965) é obedecida com boa precisão. Neste gráfico cada ponto representa o raio de giro médio para um dado valor de N . A reta ajustada apresentou um coeficiente angular de $0,522 \pm 0,001$.

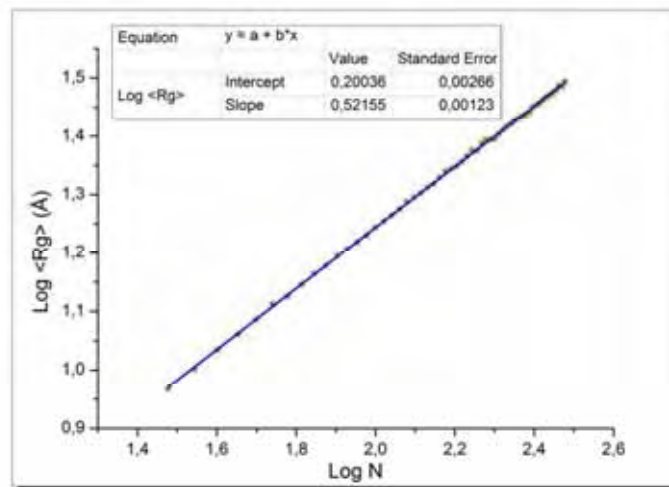


Figura 19: Valores do raio de giro médio versus N em um gráfico *di-log* para as estruturas geradas pelo algoritmo de crescimento sem o peso estatístico W_N . A inclinação da curva ajustada é de $0,522 \pm 0,001$.

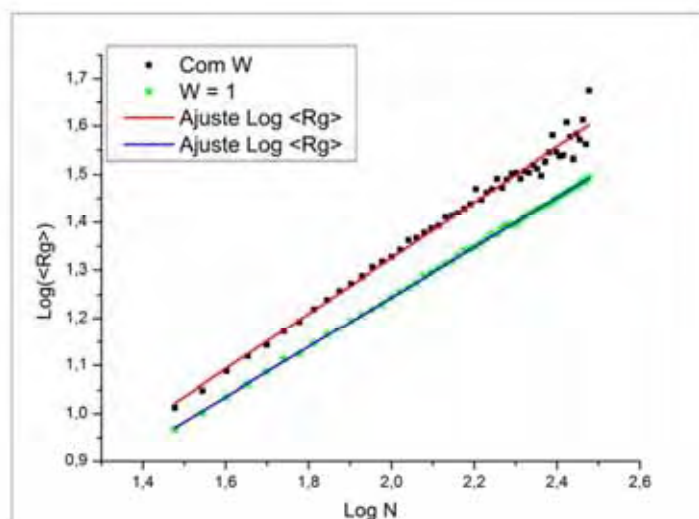


Figura 20: Gráfico *di-log* da sobreposição das duas curvas do raio de giro médio em função do número de monômeros; a curva em vermelho é com o peso estatístico e a azul é sem o peso.

Na Figura 20 sobreposamos as duas curvas do ajuste linear do raio de giro em função de N em um gráfico *di-log*. A diferença que observamos na análise dos gráficos acima para os coeficientes angulares das retas deve-se ao fato de que para o mesmo conjunto de estruturas os valores do raio de giro médio calculado com peso estatístico são maiores do que aqueles calculados sem o peso estatístico, conforme podemos observar nas *Tabela 3* e *Tabela 4*.

N	<Rgiro>	<W>	N	<Rgiro>	<W>
30	10,31	9,22E-02	170	28,98	5,79E-08
35	11,17	5,70E-02	175	29,47	2,84E-08
40	12,24	3,50E-02	180	31	1,71E-08
45	13,16	2,60E-02	185	29,69	7,72E-09
50	13,89	1,20E-02	190	30,87	4,77E-09
55	14,85	8,10E-03	195	31,78	2,99E-09
60	15,59	4,50E-03	200	31,93	2,29E-09
65	16,55	2,90E-03	205	31,03	1,30E-09
70	17,32	1,60E-03	210	32,15	9,47E-10
75	18,08	9,90E-04	215	31,93	5,36E-10
80	18,73	6,50E-04	220	33,14	1,93E-10
85	19,37	3,90E-04	225	32,51	1,11E-10
90	20,24	2,10E-04	230	31,44	1,05E-10
95	20,78	1,10E-04	235	33,59	8,87E-11
100	21,25	7,40E-05	240	35,04	2,66E-11
105	21,99	4,50E-05	245	38,09	1,65E-11
110	23,02	2,90E-05	250	35,11	1,09E-11
115	23,31	1,70E-05	255	34,39	4,35E-12
120	23,88	9,20E-06	260	34,58	5,18E-12
125	24,41	5,30E-06	265	40,5	2,62E-12
130	24,73	3,46E-06	270	37,78	1,09E-12
135	25,81	1,90E-06	275	34,04	2,17E-12
140	26,01	1,15E-06	280	38,33	5,29E-13
145	26,28	6,28E-07	285	37,26	1,77E-13
150	26,84	4,80E-07	290	41,04	1,51E-13
155	27,4	2,29E-07	295	36,47	2,54E-13
160	29,51	1,72E-07	300	47,36	1,45E-13
165	28	1,09E-07			

Tabela 3: Valores do raio de giro médio e do peso estatístico médio em função do número de resíduos N das estruturas proteicas gerado pelo algoritmo de crescimento.

N	<R _{giro} >	N	<R _{giro} >
30	9,28	170	23,23
35	10,05	175	23,77
40	10,83	180	23,72
45	11,51	185	24,42
50	12,2	190	24,79
55	12,96	195	24,77
60	13,35	200	24,87
65	14	205	25,35
70	14,6	210	25,8
75	15,07	215	26,15
80	15,66	220	26,4
85	16,12	225	26,78
90	16,55	230	26,96
95	17	235	27,27
100	17,54	240	27,38
105	18	245	27,69
110	18,4	250	28,28
115	18,85	255	28,46
120	19,41	260	28,71
125	19,76	265	29,07
130	20,1	270	29,37
135	20,53	275	29,61
140	20,83	280	29,81
145	21,34	285	30,18
150	21,87	290	30,72
155	22,09	295	30,58
160	22,32	300	31,2
165	22,73		

Tabela 4: Valores do raio de giro médio em função do número de resíduos N para as estruturas geradas sem o peso estatístico, $W=1$.

A fim de verificarmos que a rede cubo-octaédrica possui uma tendência em gerar estruturas mais compactas, fizemos o crescimento de cadeias na rede cúbica simples, cujos vetores que descrevem a rede são: $(1,0,0)$, $(-1,0,0)$, $(0,1,0)$, $(0,-1,0)$, $(0,0,1)$ e $(0,0,-1)$. Note que para um sítio no centro da rede, $(0,0,0)$, ele terá no máximo 6 sítios vizinhos mais próximos, que é a metade da rede cubo-octaédrica. No caso de crescimento de cadeias com

restrição de volume excluído (cada sítio da rede poderá ser ocupado por somente um resíduo de aminoácido), num dado estágio do crescimento, ao acrescentarmos o resíduo i , um destes 6 sítios já estará sendo ocupado pelo resíduo $i-1$, logo teremos no máximo 5 sítios disponíveis para colocarmos o resíduo i . Na *Figura 21* apresentamos o gráfico *di-log* do raio de giro, calculado com o peso estatístico, em função do número de resíduos (N) das cadeias geradas na rede cúbica, para N variando de 30 a 300. O coeficiente angular da reta ajustada foi de $0,586 \pm 0,007$.

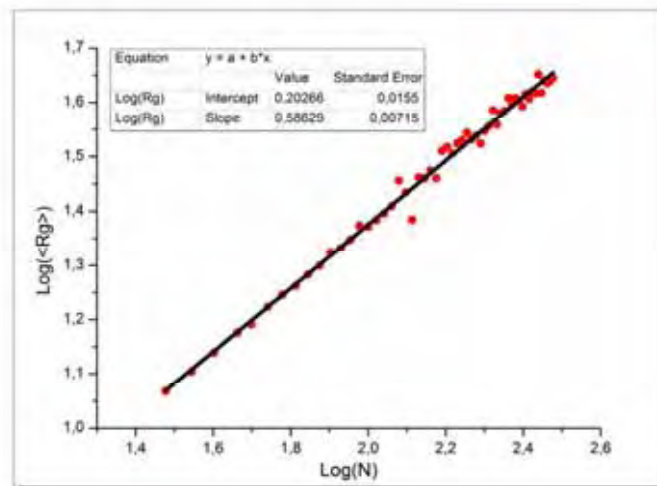


Figura 21: Valores do raio de giro médio versus N em um gráfico *di-log* para as estruturas geradas pelo algoritmo de crescimento na rede cúbica com o peso estatístico. A inclinação da curva ajustada é de $0,586 \pm 0,007$.

0,10	4,3	6,0	5,0	10,2	5,3	15,0	6,0	13,1
0,30	4,1	6,7	5,0	7,6	5,7	12,8	5,9	9,7
0,50	4,0	6,0	4,9	7,4	5,5	8,9	5,9	8,0
0,67	4,1	5,8	5,0	6,7	6,1	7,4	5,9	7,6
0,80	4,2	5,7	5,2	6,5	5,8	7,3	5,8	7,4
0,96	4,1	5,8	5,2	6,4	5,8	7,5	6,2	7,1
1,00	4,0	5,7	5,3	6,4	5,9	7,5	6,2	7,2
1,02	4,2	5,8	5,2	6,4	5,5	7,7	5,8	7,4
1,50	4,1	5,6	5,2	6,4	6,0	7,4	6,0	7,2
2,0	4,3	5,6	5,2	6,2	6,0	7,3	6,0	7,1
4,0	4,4	5,5	5,2	6,1	5,6	7,1	6,3	-

* *Root Mean Square Deviation*: é a medida da distância média entre os C_{α} dos resíduos de aminoácidos da estrutura gerada pelo algoritmo de crescimento com a estrutura nativa do PDB.

** Logo abaixo os códigos de cada proteína destacam-se o seu número de resíduos e o raio de giro da estrutura nativa.

*** A lacuna não preenchida para o $\langle R_{giro} \rangle$ da proteína *2cro*, em $\beta = 4,0$, deve-se ao fato de que o termo do denominador da equação 15 tende a zero, e desta forma, extrapola os limites para o cálculo do valor médio.

Testamos este algoritmo de crescimento para quatro proteínas, cujos códigos do *Protein Data Bank (PDB)* são: *2mhu*, *1crn*, *1r69* e *2cro*. Cada uma dessas com 30, 46, 63 e 65 aminoácidos, respectivamente. Conforme mostramos na *Tabela 5*, para o *ensemble* de estruturas crescidas (10^5 para cada β) pelo algoritmo calcularam-se o raio de giro da estrutura e também o desvio médio quadrático em relação à estrutura no estado nativo. O raio de giro da estrutura crescida está intrinsecamente relacionado com o parâmetro de temperatura β , por isso foi necessário fazer uma varredura para vários valores deste parâmetro. Nesta análise do crescimento ressaltamos que a busca pela estrutura que apresenta o menor RMSD nem sempre

é aquela com a menor energia de interação. Podemos verificar isso nos gráficos da *Energia x RMSD*, Figura 25, a distribuição de energia do *ensemble* de estruturas geradas em $\beta = 0,5$ para a proteína *2 mhu*.

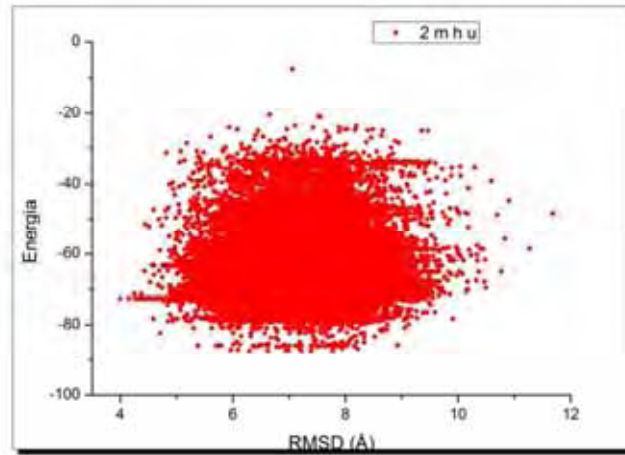


Figura 22: Distribuição da Energia em função do RMSD (Å) do ensemble de estruturas geradas para a proteína *2 mhu* em $\beta = 0.5$. O menor RMSD é de 4.0 Å.

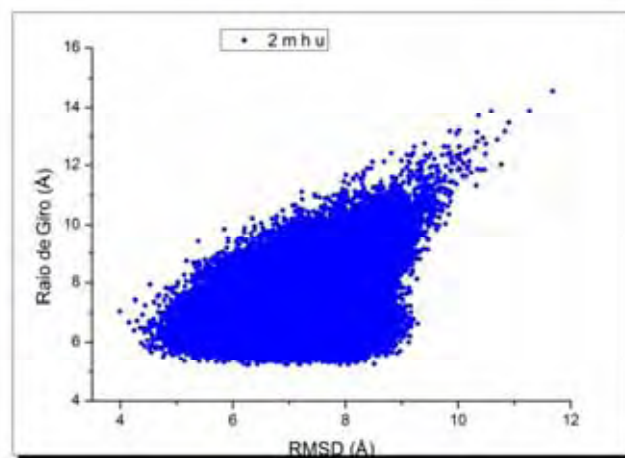


Figura 23: Distribuição do Raio de giro em função do RMSD (Å) do ensemble de estruturas geradas para a proteína *2mhu* em $\beta = 0.5$. O menor RMSD é de 4.0 Å.

De uma forma geral, o raio de giro da estrutura que apresentava o menor RMSD era sempre próximo da estrutura nativa, e quase sempre um pouco maior do que o raio de giro

$$R_{giro} = \text{ARA.WAAWA} \quad ($$

15. Ilustramos na Figura 23 que em $\beta = 0,5$ o algoritmo encontra uma estrutura de 4,0 Å de RMSD e 7,1 Å de raio de giro. Isso nos mostra que este método de crescimento privilegia as estruturas mais compactas.

Para o conjunto de β apresentados na Tabela 5, destacamos aqueles que, do ensemble de estruturas geradas, apresentava uma conformação com o menor valor de RMSD. Fizemos a sobreposição dessas estruturas utilizando o software VMD (*Visual Molecular Dynamic*) que é fornecido pelo Grupo de Física Computacional e Teórica da Universidade de Illinois, USA. Disponível para baixar através do site: (<http://www.ks.uiuc.edu/vmd>). Nas Figura 24, Figura 25, Figura 26 e Figura 27 representamos a estrutura nativa em azul escuro e a estrutura gerada pelo algoritmo de predição em azul claro.



Figura 24: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 0,5$ e RMSD = 4.0 Å. Código PDB da proteína: *2 mhu*.

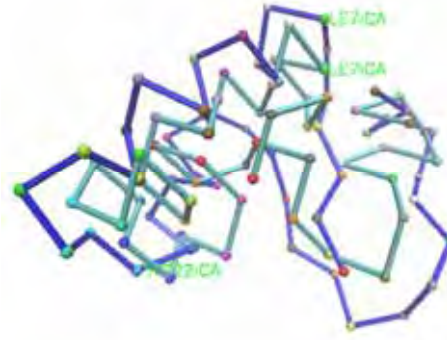


Figura 25: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 0,5$. Código PDB da proteína: *1crn*. Entre os resíduos 7 e 22 as estruturas apresentam um RMSD de 3,4 Å.



Figura 26: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 0,5$ e RMSD = 5.5 Å. Código PDB da proteína: *1r69*.



Figura 27: Sobreposição da estrutura nativa (azul escuro) com a correspondente estrutura gerada pelo algoritmo de crescimento (azul claro) em $\beta = 1,02$ e RMSD = 5.8 Å. Código PDB da proteína: *2cro*.

5.3 Estudo do Estado de Transição da Proteína CI2 – código PDB *1coa*

A proteína *CI2* é muito estudada na literatura, por se tratar de uma enzima serino protease inibidora de quimotripsina (OLIVEIRA, 2010; JACKSON e FERSHT, 1991). Estudos sobre o enovelamento de proteína revelam que ela apresenta um enovelamento de dois estados, isto significa que em determinadas condições físicas, como, por exemplo, pH, temperatura ou concentração salina, ela apresenta estrutura bem definida – enovelada ou desenovelada. Ela é formada por 64 resíduos de aminoácidos e possui como estruturas secundárias: uma folha beta, um beta *hairpin*, três *strands*, duas hélices, cinco betas *tuns* e um gama *turn*; veja na Figura 28 uma ilustração destacando essas estruturas secundárias.

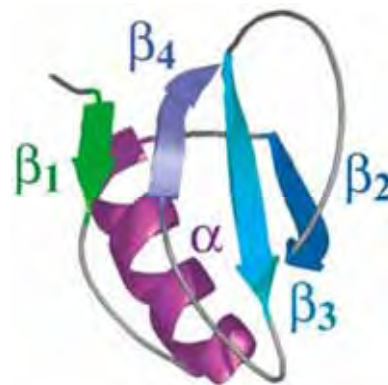


Figura 28: Ilustração das estruturas secundárias da proteína CI2, código PDB *1coa*; em roxo destacamos a hélice α e as setas, em verde e azul, as folhas betas.

Neste trabalho, analisaremos um modelo teórico computacional conhecido como modelo C_α na rede cubo-octaédrica, como descrito anteriormente. Realizamos simulações com um potencial baseado nos contatos retirados da estrutura nativa e durante a simulação cada monômero era movimentado segundo uma probabilidade dependente da temperatura,

$$\text{veja } P = e^{-\beta \cdot \Delta E} \begin{cases} P > n, & \text{aceita o movimento.} \\ P < n, & \text{descarta o movimento.} \end{cases} \quad ($$

19. Segue no anexo **B** o programa que escrevemos, em linguagem *Fortran*, para realizar esta simulação. Este programa realiza os movimentos de *end-move* e *corner-move*, ainda não conseguimos implementá-lo para realizar o *crankshaft*.

Ao longo da simulação verificam-se periodicamente o raio de giro, o RMSD e os contatos nativos que são formados, Figura 29. Neste algoritmo observamos uma excelente consistência entre as estruturas de baixo RMSD e os valores de baixa energia da cadeia, ou seja, alta porcentagem do número de contatos nativos sendo formados; o que nem sempre acontecia no algoritmo de predição de estruturas. Quando a estrutura forma mais de 100 contatos pode-se verificar valores baixos de RMSD, em torno de 3,5 Å, como mostra a Figura 29 (c).

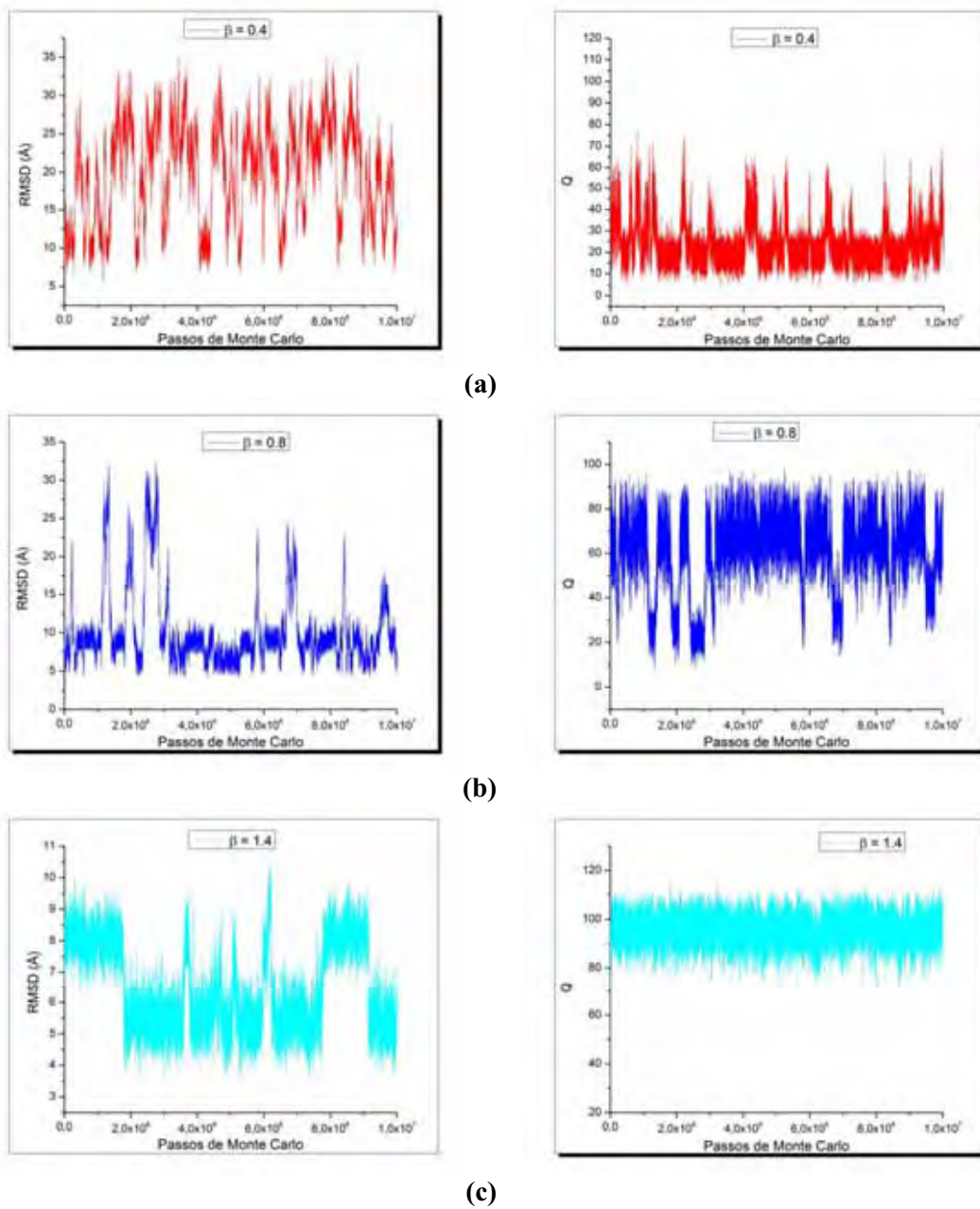


Figura 29: Ao transcorrer da simulação a estrutura visita estados conformacionais desenovelados caracterizados por baixos valores de Q , ilustrado em **(a)** $\beta = 0,4$, ou seja, $T = 2,5$; em **(b)** temos flutuação entre os estados enovelado e desenovelado, temperaturas próximas do estado de transição, em torno de $\beta = 0,8$, ou seja, $T = 1,25$, e **(c)** para temperaturas abaixo do estado de transição, $\beta = 1,4$, ou seja, $T = 0,71$, a estrutura permanece a maior parte do tempo no estado enovelado, com cerca de 75 % dos contatos nativos sendo realizados.

Realizamos várias simulações entre o intervalo de β igual a 0,2 até 1,6 a fim de calcularmos a curva de calor específico, Figura 30, e energia livre para o estado de transição, Figura 31.

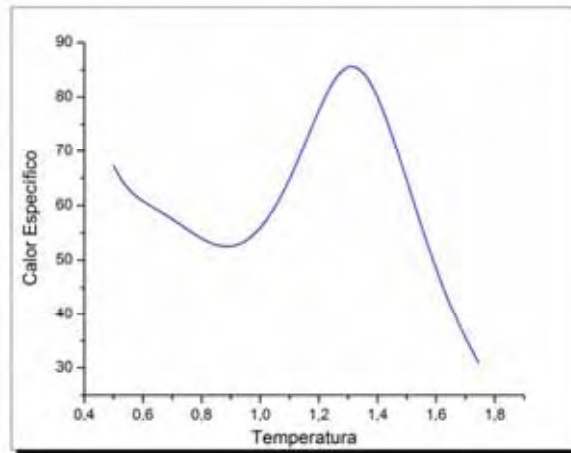


Figura 30: Gráfico do calor específico para a proteína *I coa*. Sabemos que o calor específico mede a flutuação da energia, $\frac{C}{k} = \beta^2 [\langle E^2 \rangle - \langle E \rangle^2]$, sendo assim o pico da curva nos mostra que a temperatura de transição de fase da estrutura está em torno da temperatura 1,3.

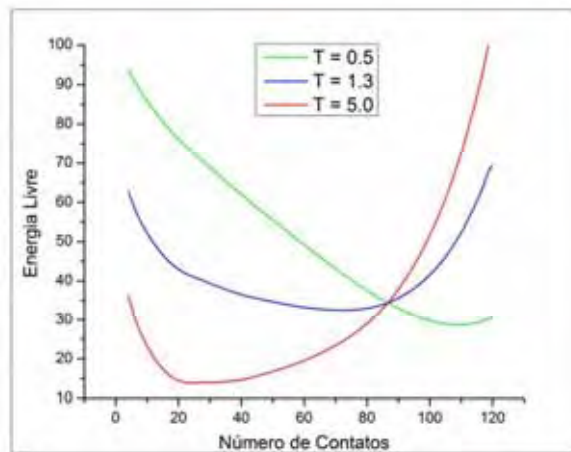


Figura 31: Gráfico da energia livre ($F(Q) = -K_B T \ln Z$) em função do número de contatos em diferentes valores de temperatura.

Vários métodos têm sido usados para calcular as mudanças na energia livre entre moléculas que interagem entre si e investigar as estabilidades relativas de diferentes estados conformacionais. Utilizamos o método dos histogramas proposto por KUMAR, 1992 e 1995; tal método consiste em determinar a densidade de estados a partir de vários histogramas coletados em diferentes temperaturas. O programa que implementa o método de análise dos

histogramas ponderados (KUMAR, *et. al.* 1995) em uma e duas dimensões é o WHAM – *Weighted Histogram Analysis Method*.

A probabilidade $P(E)$ de gerarmos um estado com energia E é dada por:

$$P(E) = \frac{\Omega(E)e^{-\beta E}}{Z} \quad , \quad Z = \sum_E \Omega(E)e^{-\beta E} \quad (20)$$

onde $\Omega(E)$ é a densidade de estados com energia E , que é a grandeza a ser determinada.

Considere um número R de simulações feitas em diferentes temperaturas. A simulação i é realizada na temperatura $T_i = 1/\beta_i$; as simulações são realizadas em um intervalo de temperatura suficiente para cobrir o domínio de energias relevante para a descrição do sistema. Segundo o método dos histogramas a densidade de estados é calculada por:

$$\Omega(E) = \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} e^{-\beta E_j}} \quad (21)$$

onde n é o número de medidas da energia $N(E)$ numa dada conformação.

Para determinar a função de partição Z_j parte-se de um valor arbitrário para a função de partição:

$$Z_k = \sum_E \Omega(E) e^{-\beta_k E} \quad (22)$$

$$= \sum_j n_j Z_j^{-1} e^{-\beta E_j} \quad Z_j^{-1} \quad e^{-\beta E_j}$$

(21 que permite obter $\Omega(E)$. Este, por sua vez, inserido na $Z_k = \sum_E \Omega(E) e^{-\beta_k E}$

(22 leva a novos valores para Z_k , e assim sucessivamente até que os valores das funções de partição Z_k , não variem significativamente. E, por fim, a energia livre de Gibbs é definida como:

$$F(Q) = -K_B T \ln Z \quad (23)$$

onde K_B é a constante de Boltzmann.

Na Figura 31, apresentamos o gráfico da energia livre – em unidades de $K_B T$ – em função do número de contatos. Esta curva é caracterizada na literatura como uma transição *down-hill* (ladeira abaixo), XIU *et. al.*, 2012. Porém, sabe-se que esta proteína realiza uma transição de dois estados (JACKSON, 1991; CHAHINE, 2007), onde um pico na barreira de energia representa o estado de transição, veja Figura 32. Provavelmente, essa divergência deve-se ao fato da proteína não ter conseguido se relaxar suficientemente durante a simulação, ou por causa do tipo do movimento realizado (*corner move* e *end move*) ou pelo tempo de simulação.

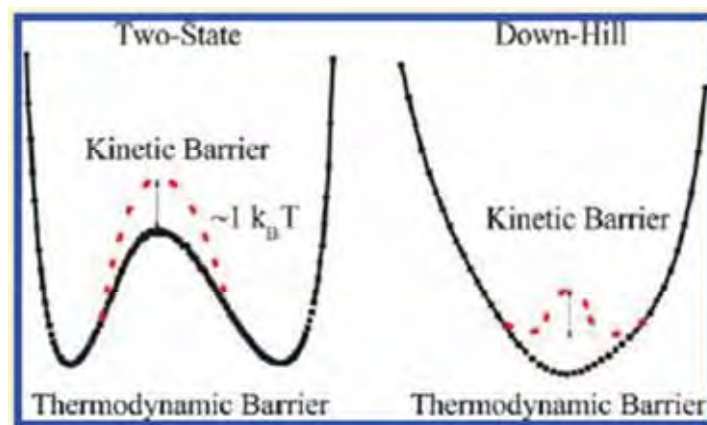


Figura 32: Ilustração representando a barreira de energia livre dos estados de transição de proteínas (XIU, 2012).

Apesar dessa inconsistência quanto à barreira de energia do estado de transição, o modelo utilizado conseguiu alguns resultados interessantes como, por exemplo, a concordância entre o raio de giro e o RMSD, veja Figura 33. No estado nativo PDB a proteína *I coa* apresenta um raio de giro de 10,8 Å; durante a simulação nos intervalos de tempo que a

proteína visita estados de baixos valores de RMSD o raio de giro da conformação fica próximo do valor nativo Figura 33 (c). Para temperaturas abaixo do estado de transição a estrutura fica congelada numa conformação com um raio de giro abaixo do valor nativo.

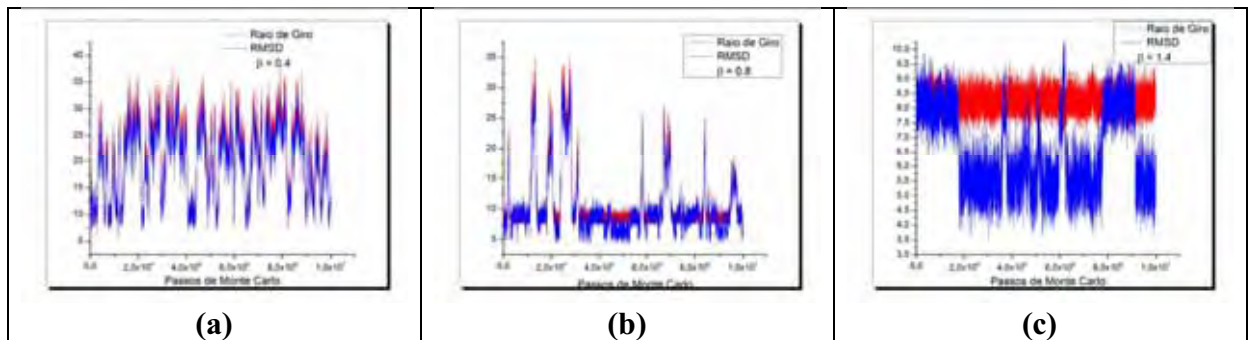


Figura 33: Gráficos da sobreposição do RMSD – curva em azul – e Raio de Giro – curva em vermelho – da estrutura ao longo da simulação, ambos estão medidos em Å. Em (a) $T = 2.5$, a temperatura está acima; em (b) $T = 1.25$, está próximo e, (c) $T = 0.7$, está abaixo da temperatura de transição.

Outro resultado relevante deste modelo foi concordância com relação aos valores de ϕ values (OLIVEIRA, 2010). Experimentalmente, os valores de ϕ estão relacionados com a importância de um par de resíduos fazer um contato durante o estado de transição, sendo que este contato contribui significativamente para a estabilidade da proteína nativa. Computacionalmente, os valores de ϕ values em simulações conformacionais, são calculados aproximadamente (CLEMENTI, 2000) para cada par de resíduo i e j como:

$$\phi_{i,j} \approx \frac{P_{i,j}^{TS} - P_{i,j}^U}{P_{i,j}^F - P_{i,j}^U} \quad (24)$$

onde $P_{i,j}^X$ é a probabilidade do contato entre i e j ser formado no estado X (com X sendo F , TS , ou U ; siglas em inglês para Enovelado, Estado de Transição e Desenovelado,

$$\phi_{i,j} \approx P_{i,j}^{TS} - P_{i,j}^U P_{i,j}^F - P_{i,j}^U$$

(24 podemos calcular os valores de ϕ_i pesados sobre todos os contatos nativos com o resíduo i .

Representamos na Figura 34 os valores de ϕ_i values obtidos neste modelo em rede com os valores calculados no espaço contínuo (OLIVEIRA, 2010). Mesmo com um modelo relativamente simples conseguimos determinar, com boa concordância, os resíduos que têm um papel crucial nos contatos realizados no estado de transição.

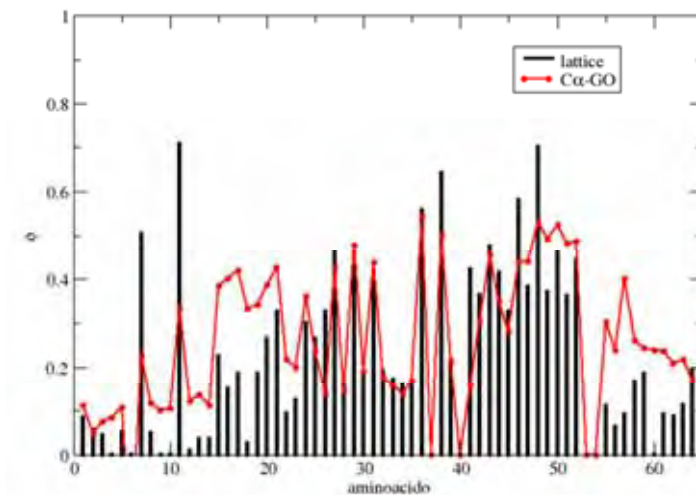


Figura 34: Sobreposição dos ϕ values obtidos pelo modelo em rede realizado neste trabalho – barra em preto – com os valores obtidos teoricamente com o modelo C_α no espaço contínuo.

O $\phi_{i,j}$ value é uma medida conveniente da fração estrutura nativa formada no estado de transição (CLEMENTI, 2000). Um $\phi_{i,j}$ value próximo de 1 significa que a probabilidade de um contato ser formado no estado de transição é maior do que a probabilidade dele ser feito

no estado desenovelado, portanto o contato foi formado no estado de transição; de outra forma, um $\phi_{i,j}$ value próximo de 0 significa que a probabilidade de um contato existir no estado de transição é a mesma dele ser feito no estado desenovelado, portanto este contato não é importante no estado de transição, uma vez que ele já havia sido formado no estado desenovelado.

Na Figura 35 comparamos os $\phi_{i,j}$ values obtidos pelo nosso modelo em rede com o modelo C_α no espaço contínuo (OLIVEIRA, 2010). Verifica-se novamente uma boa correlação entre esses modelos, os contatos entre os resíduos do meio da cadeia são realizados quase com o mesmo peso. Porém, no modelo em rede observamos que não ocorreu a formação de alguns contatos de longo alcance na cadeia como, por exemplo, entre os resíduos das pontas (2 e 63 ou entre 4 e 63). Estes contatos são mais difíceis de serem formados devido às grandes distâncias entre estes resíduos; o que pode estar dificultando a formação destes contatos pode ser uma falta de cooperatividade do sistema.

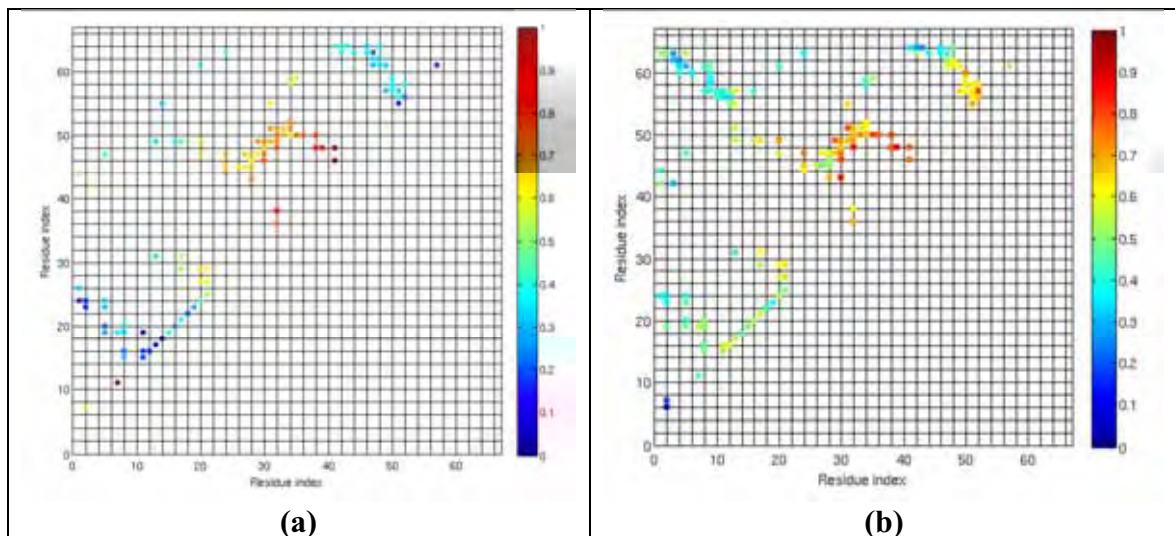


Figura 35: Valores de $\phi_{i,j}$ values da estrutura no estado de transição para a proteína CI2 - *Icoa* segundo o modelo C_α (a) na rede cubo-octaédrica e (b) no espaço contínuo. Diferentes cores indicam diferentes valores de 0 a 1, como quantificado por uma escala de cores. De uma forma geral, o ensemble do estado de transição apresenta-se como uma versão expandida do estado nativo onde a maioria dos contatos tem uma mesma probabilidade de participação, mas algumas interações são menos prováveis de ocorrerem (CLEMENTI, 2000).

Capítulo 6

6.1 Conclusão e Perspectivas Futuras

Realizamos neste trabalho três linhas de estudo. Na primeira, estudamos um método de crescimento de estruturas proteicas, sem potencial de interação entre os resíduos não ligados. Analisando o ensemble de estruturas crescidas na rede calculamos o coeficiente angular da reta que relaciona o raio de giro com o número de monômeros da cadeia. Fizemos estes cálculos com e sem a função de pesos estatísticos proposta por ROSENBLUTH and ROSENBLUTH, 1954. O uso destes pesos permite que um maior número de sítios da rede seja visitado durante o processo de crescimento, contribuindo desta forma com um melhor ajuste para a curva do raio de giro em função do número de monômeros.

Uma segunda linha de estudo que realizamos foi a predição de estrutura proteica. Sabendo que a rede cubo-octaédrica podia representar com boa precisão a estrutura de proteína, implementamos o algoritmo de crescimento de cadeia com um potencial de interação estatístico entre os resíduos de aminoácidos. No processo de crescimento, os sítios da rede são selecionados para receberem um monômero da cadeia de acordo com uma probabilidade dependente da temperatura. Mesmo não apresentando uma boa concordância

entre as estruturas de menor RMSD com aquelas de menor energia de interação entre os resíduos, o algoritmo conseguia obter valores médios do raio de giro do ensemble que estavam bem próximo ao da estrutura nativa; verificamos também que a probabilidade dependente da temperatura contribui para gerar estruturas mais abertas ou fechadas dependendo da temperatura utilizada.

Uma última abordagem que fizemos foi o estudo do estado de transição da proteína *I coa*. Comparamos nossos resultados de ϕ values na rede com os da literatura, calculados no espaço contínuo, a fim de comprovarmos que nosso método estava reproduzindo confiavelmente os valores esperados. Verificamos que, dada as limitações do método, existe uma boa concordância entre os resultados encontrados com os da literatura.

Temos como perspectivas para esse trabalho a implementação do movimento de crankshaft no algoritmo de dinâmica, verificar se utilizando outro potencial de interação seria possível observar os valores de ϕ value para os contatos de longo alcance e, também, fazer os estudos de predição de estrutura utilizando o potencial estatístico associado com o algoritmo de dinâmica implementado com todos os movimentos.

Referências Bibliográficas

CAMPBELL M. K. *Bioquímica*, 3ª Ed., Porto Alegre: Artmed, 2000.

CHAHINE J. *et. al.* Configuration-dependent Diffusion can shift the kinetic transition state and barrier height of protein folding. *Proceedings of the National Academy of Sciences*, v. 104, nº 37, 14646-14651, 2007.

CHAN H S.; DILL K. A. The protein folding problem. *Physics Today*, v. 24, 1993.

CLEMENTI C., *et. al.* Topological and Energetic Factors: What Determines the Structural Details of the Transition State Ensemble an “En-route” Intermediates for Protein Folding? An Investigation for Small Globular Proteins. *Journal Molecular Biology*, v. 298, p. 937-953, 2000.

COVELL D. G.; JERNIGAN R. L. Conformations of Folded Proteins in Restricted Spaces. *Biochemistry*, v.29, p. 3287-3284, 1990.

CREIGHTON T. E. *Proteins: Structures and Molecular Properties*. 2nd ed. New York: W. H. Freeman, 2003.

FLORY P. J. *Principles of Polymer Chemistry*. Cornell University Press, Ithaca, NY, 1965.

GAN H. H.; TROPSHA A.; SCHLICK T. Generating Folded Protein Structures with a Lattice Chain Growth Algorithm. *Journal of Chemical Physics*, v. 113, n. 13, p. 5511-5524, 2000.

HINDS D.; LEVITT M. A Lattice Model for Protein Structure Prediction at Low Resolution. *Proc. Natl. Acad. Sci. USA*, v. 89, p. 2536-2540, 1992.

HWANG S.; HILTY C. Folding Determinants of disulfide Bond Forming Protein B Explored by Solution Nuclear Magnetic Resonance Spectroscopy. *Proteins: Structure, Function, and Bioinformatics*, 2011.

JACKSON S. E.; FERSHT A. R. Folding of Chymotrypsin Inhibitor 2. Evidence for a Two-State Transition. *Biochemistry*, v. 30, p. 10428-10435, 1991.

JAYMARAM B. *Supercomputing facility for bioinformatics & computational biology*. Indian Institute of Technology, Índia. Disponível em: <<http://www.scfbio-iitd.res.in/software/proteomics/rg.jsp>>. Acesso em: 21 Julho, 2010.

KUMAR S. *et. al.* The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *Journal of Computational Chemistry*, v.13, p. 1011-1021, 1992.

KUMAR S. *et. al.* Multidimensional Free-Energy Calculations Using the Weighted Histogram Analysis Method. *Journal of Computational Chemistry*, v. 16, p. 1339-1350, 1995.

MEIROVITCH H. Method for Estimating the Entropy of Macromolecules with Computer Simulation. Chains with Exclude Volume. *Macromolecules*, v. 16, p. 249-252, 1983.

OLIVEIRA R. J., *et. al.* The Origin of Nonmonotonic Complex Behavior and the Effects of Nonnative Interactions on the Diffusive Properties of Protein Folding. *Biophysical Journal*, v. 99, p. 600-608, 2010.

RAGHUNATHAN G.; JERNIGAN R. L. Ideal architecture of residue packing and its observation in protein structures. *Protein Science*, v.6, p. 2072-2083, 1997.

ROCHA W. R. Interações Intermoleculares. *Cadernos Temáticos de Química, Nova Escola*, Nº 4, Maio, 2001.

ROSENBLUTH M. N.; ROSENBLUTH A. W. Monte Carlo Calculation of the Average Extension of Molecular Chains. *The Journal of Chemical Physics*, v. 23, n. 2, p. 356-359, 1954.

STRYER L. *Bioquímica*, 3ª ed., Rio de Janeiro: Guanabara Koogan S.A, 1992.

THOMAS P. D.; DILL K. A. An Iterative Method for Extracting Energy-Like Quantities from Protein Structures. *Proc. Natl. Acad. Sci. USA*, v.93, p.11628-11633, 1996.

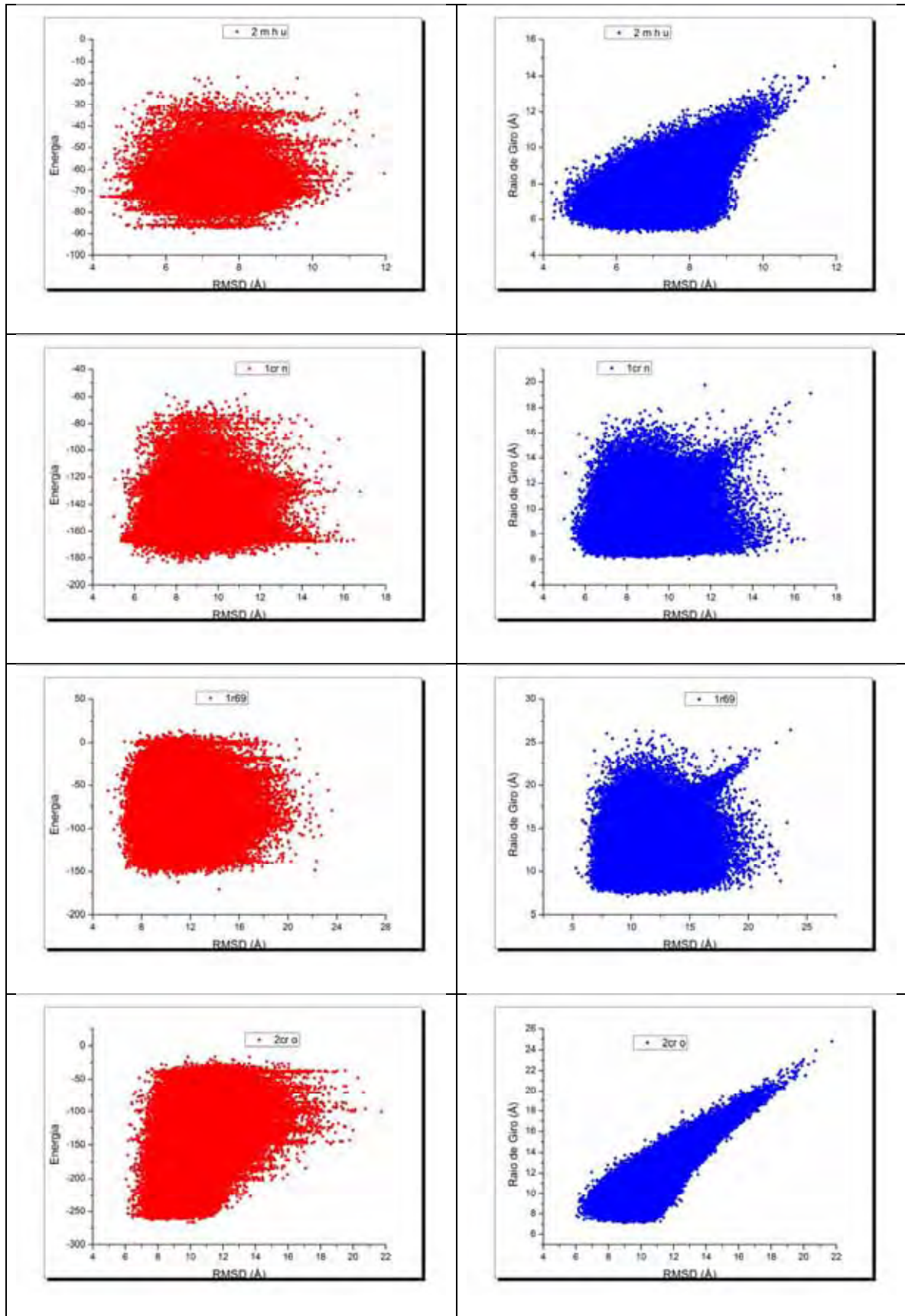
VENDRUSCOLO M.; DOMANY E. Protein Folding Using Contact Maps. *Vitamins and Hormones*, v. 58, p.171-212, 2000.

Apêndice A

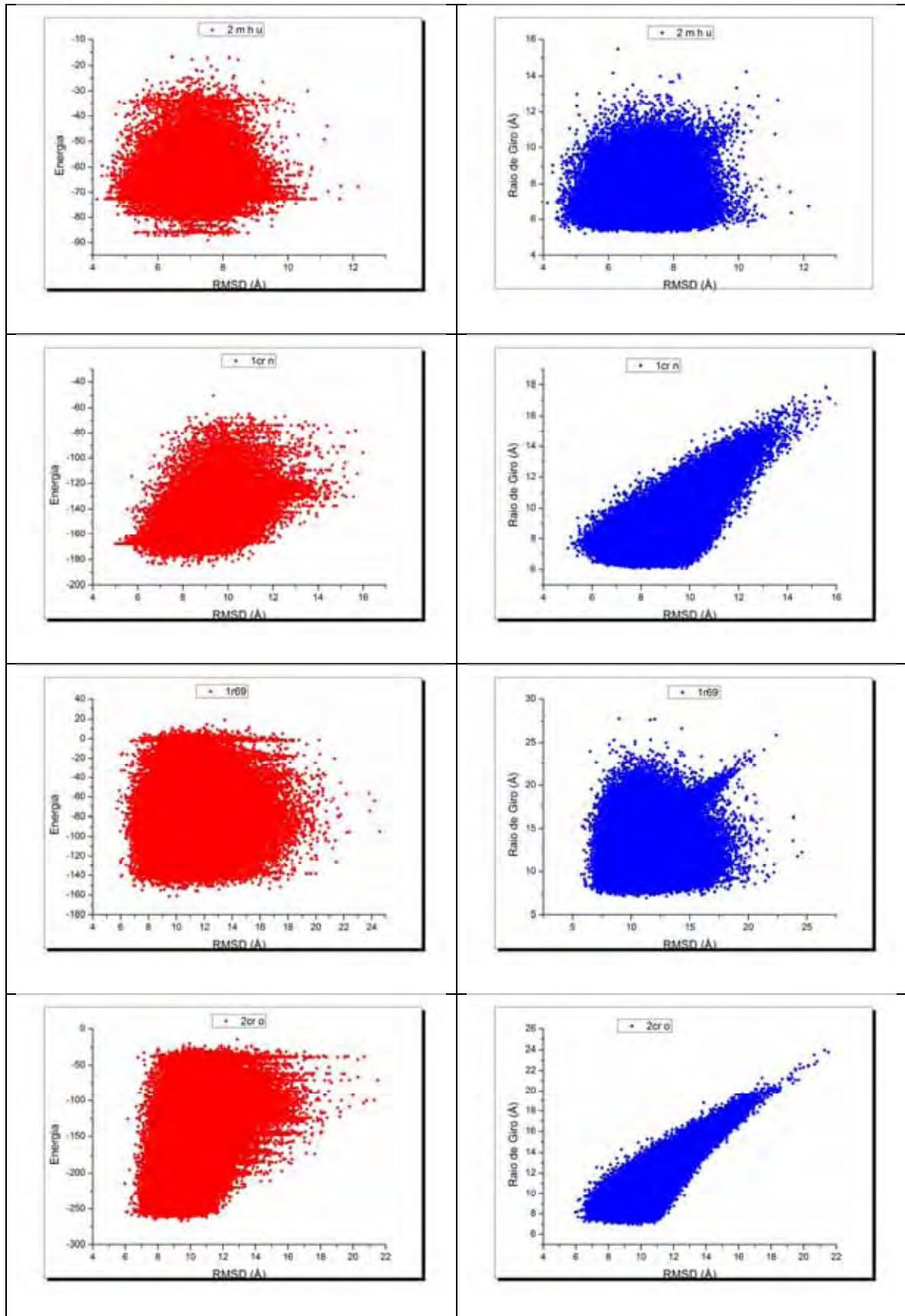
Figuras do Algoritmo de Predição de Estruturas Proteicas

Segue abaixo uma lista de todos os gráficos das energias e raios de giro em função do RMSD para as temperaturas que foram realizadas o crescimento de cadeias proteicas com o potencial de interação entre os resíduos de aminoácidos, conforme descrito na seção 2.3. Nas figuras abaixo informamos o valor do parâmetro de temperatura e, logo em seguida, os gráficos de Energia *versus* RMSD (figuras em vermelho) e Raio de Giro *versus* RMSD (figuras em azul) para cada uma das 4 proteínas utilizadas neste algoritmo de crescimento. Na legenda de cada figura está o código PDB da proteína à qual se refere.

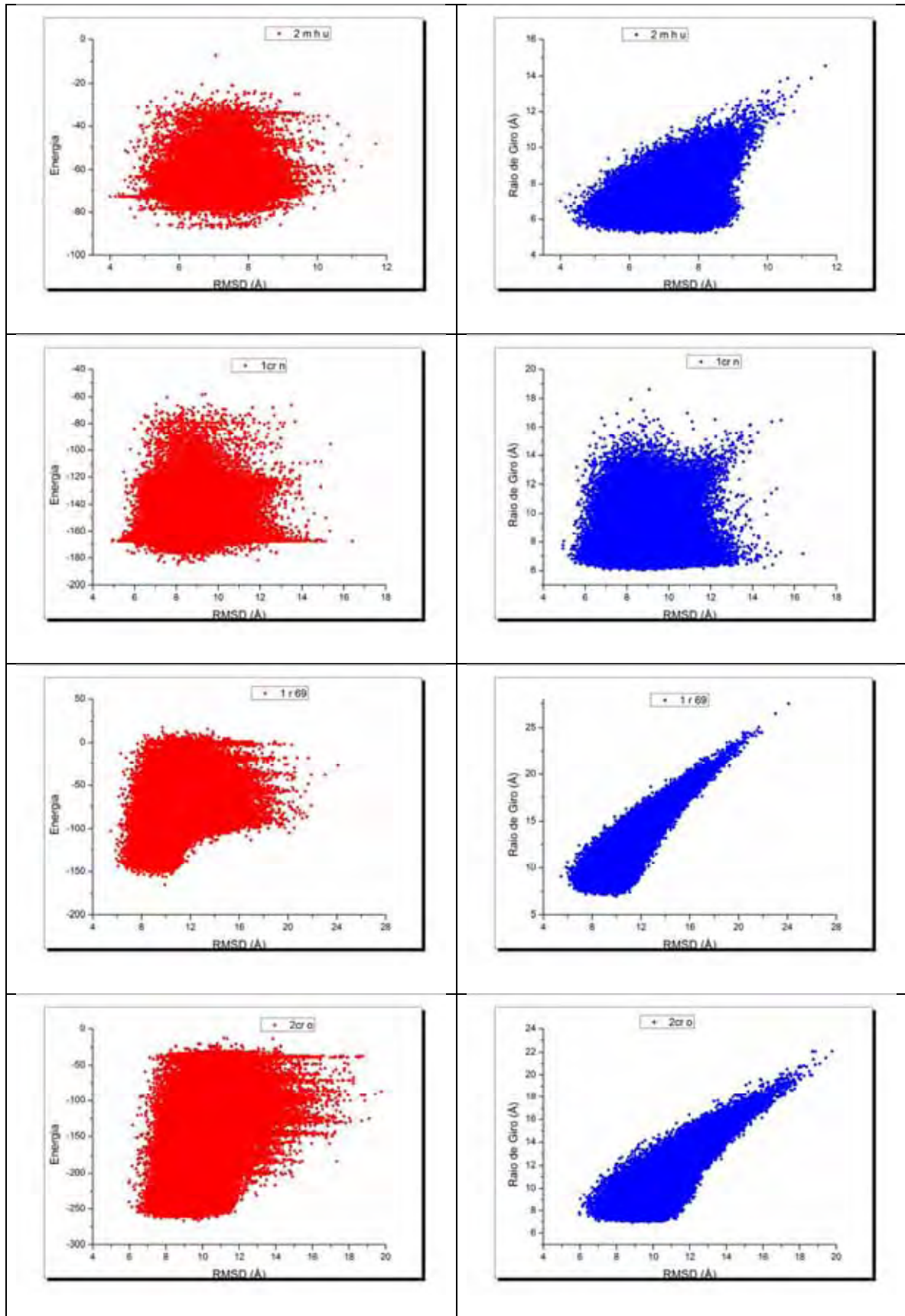
$$\beta = 0,10$$



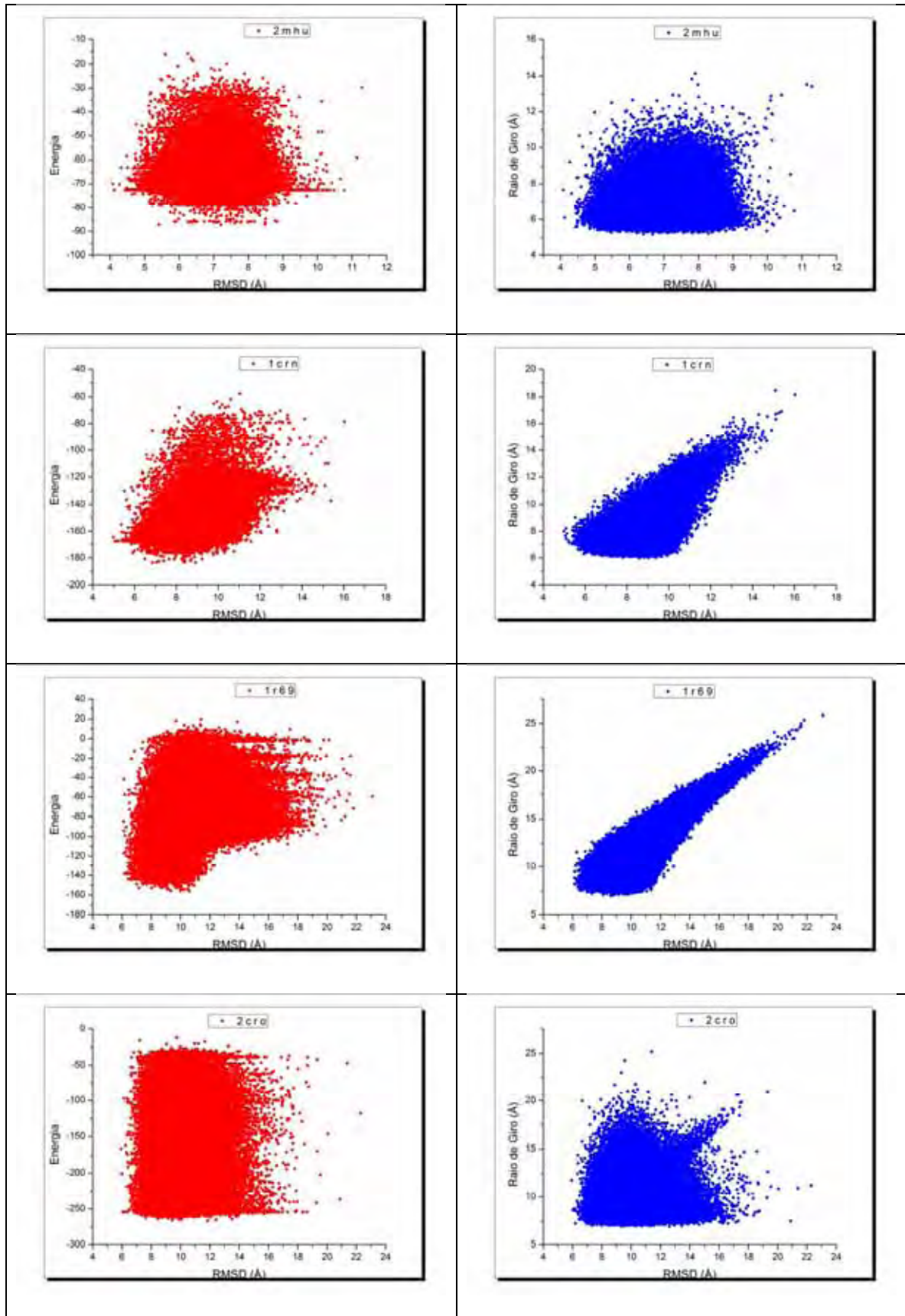
$$\beta = 0,30$$



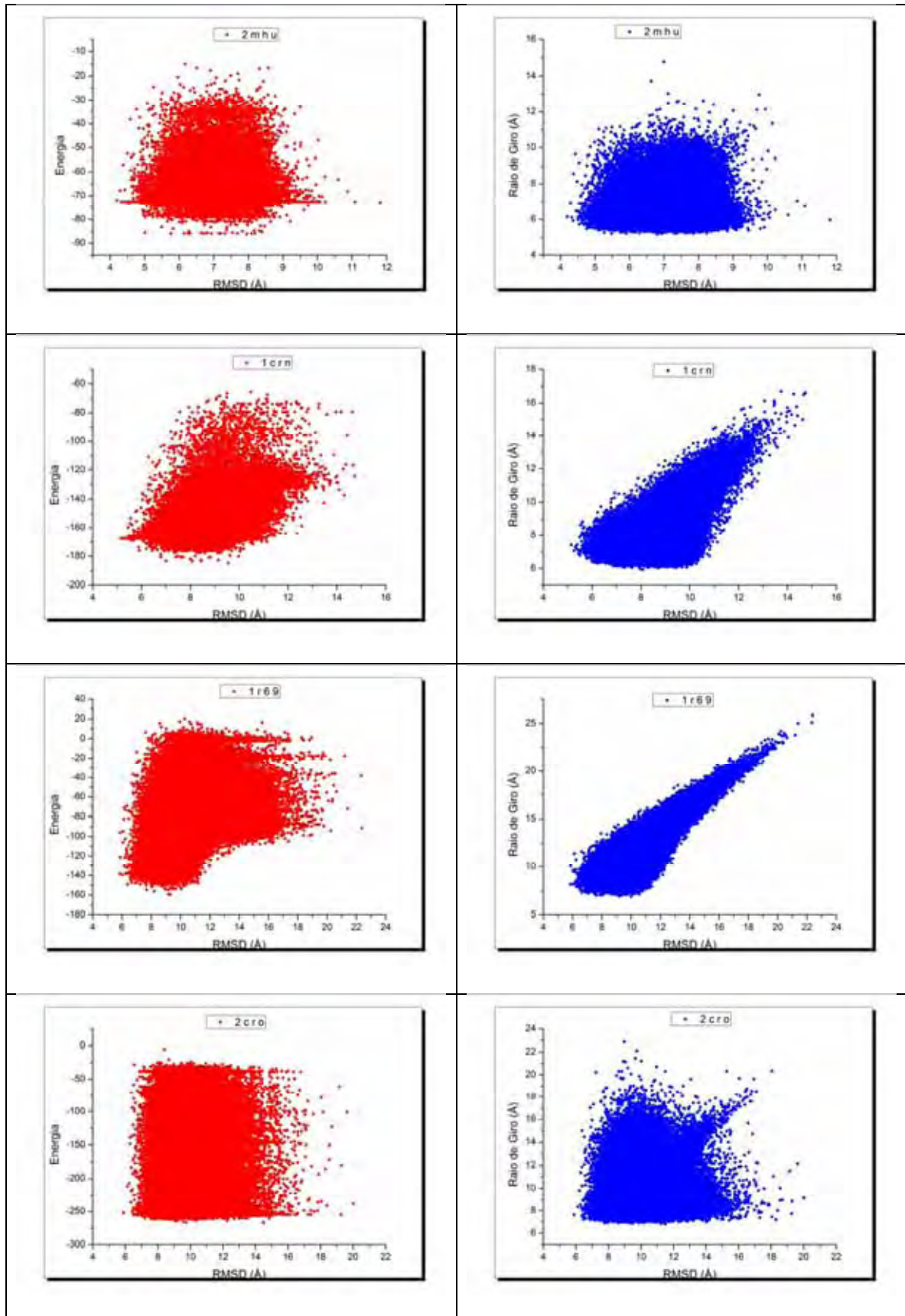
$$\beta = 0,50$$



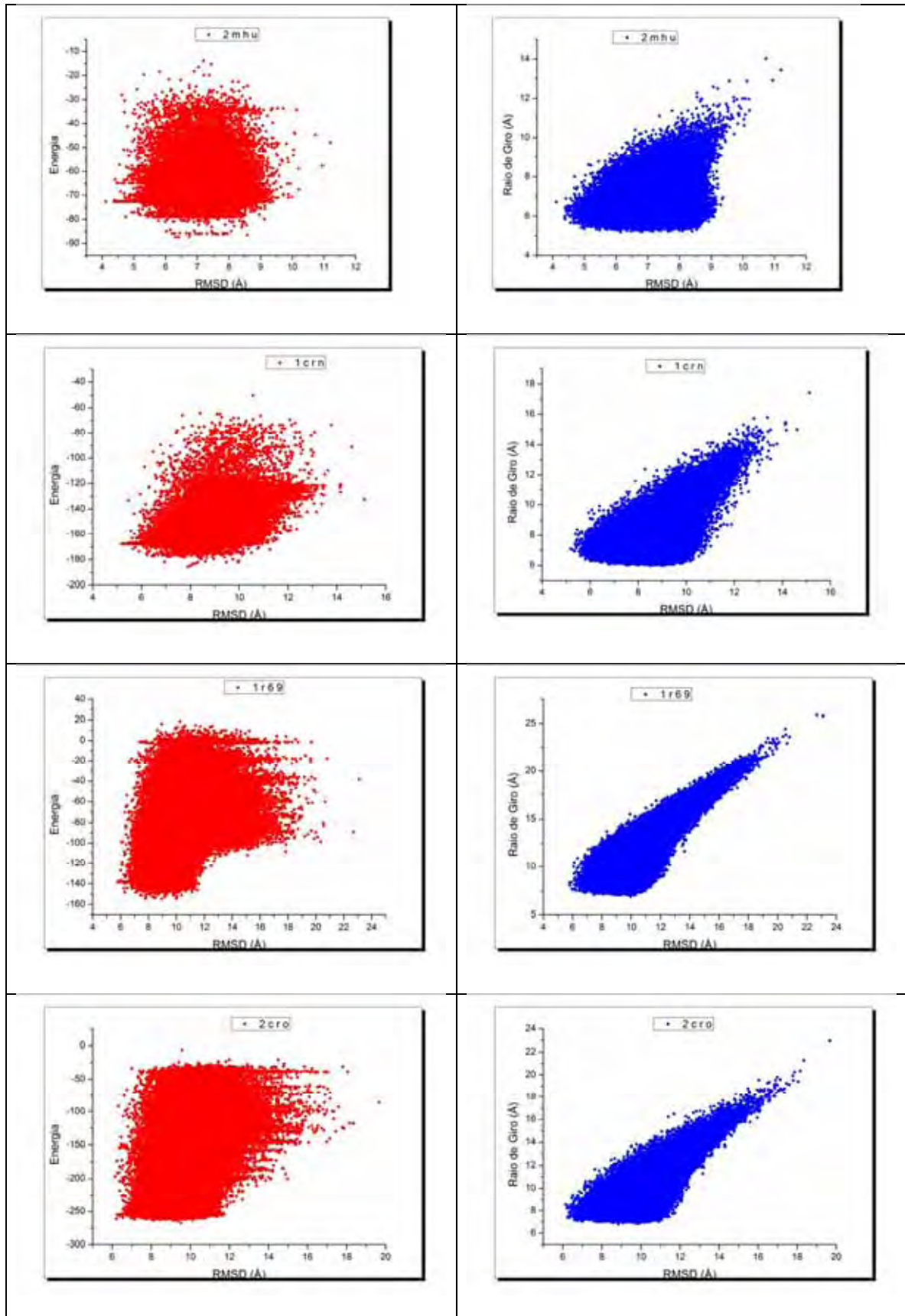
$$\beta = 0,67$$



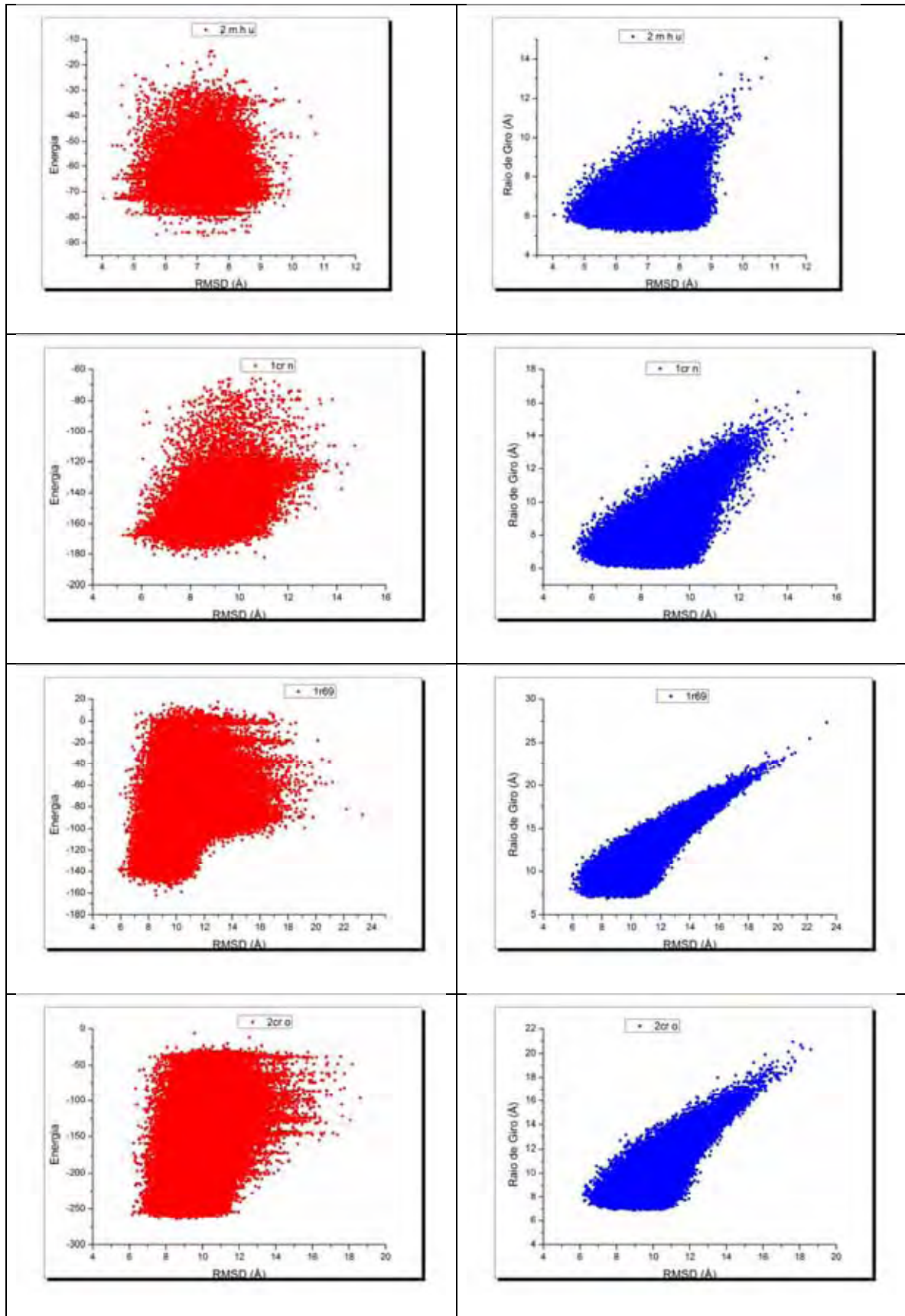
$$\beta = 0,80$$



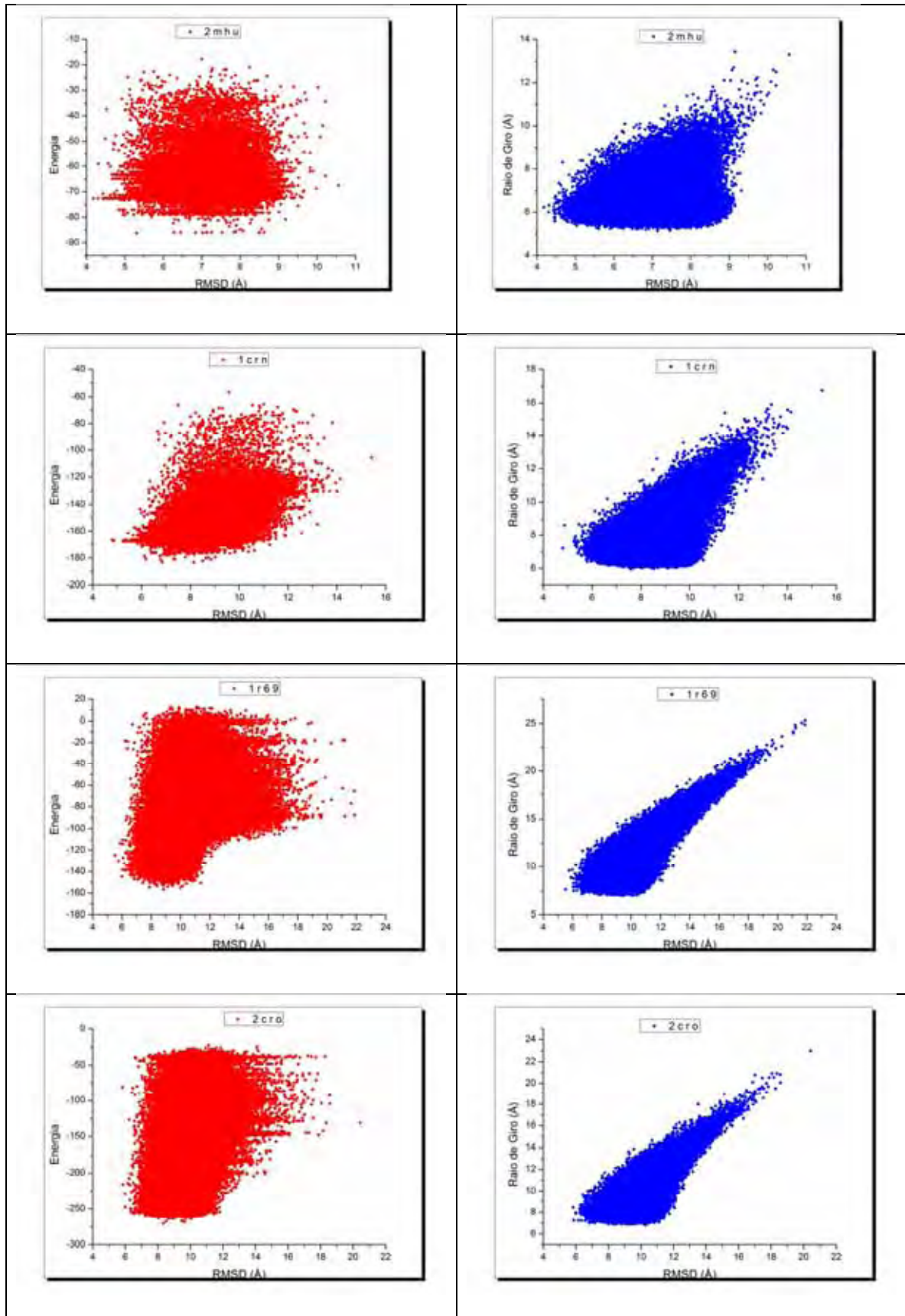
$$\beta = 0,96$$



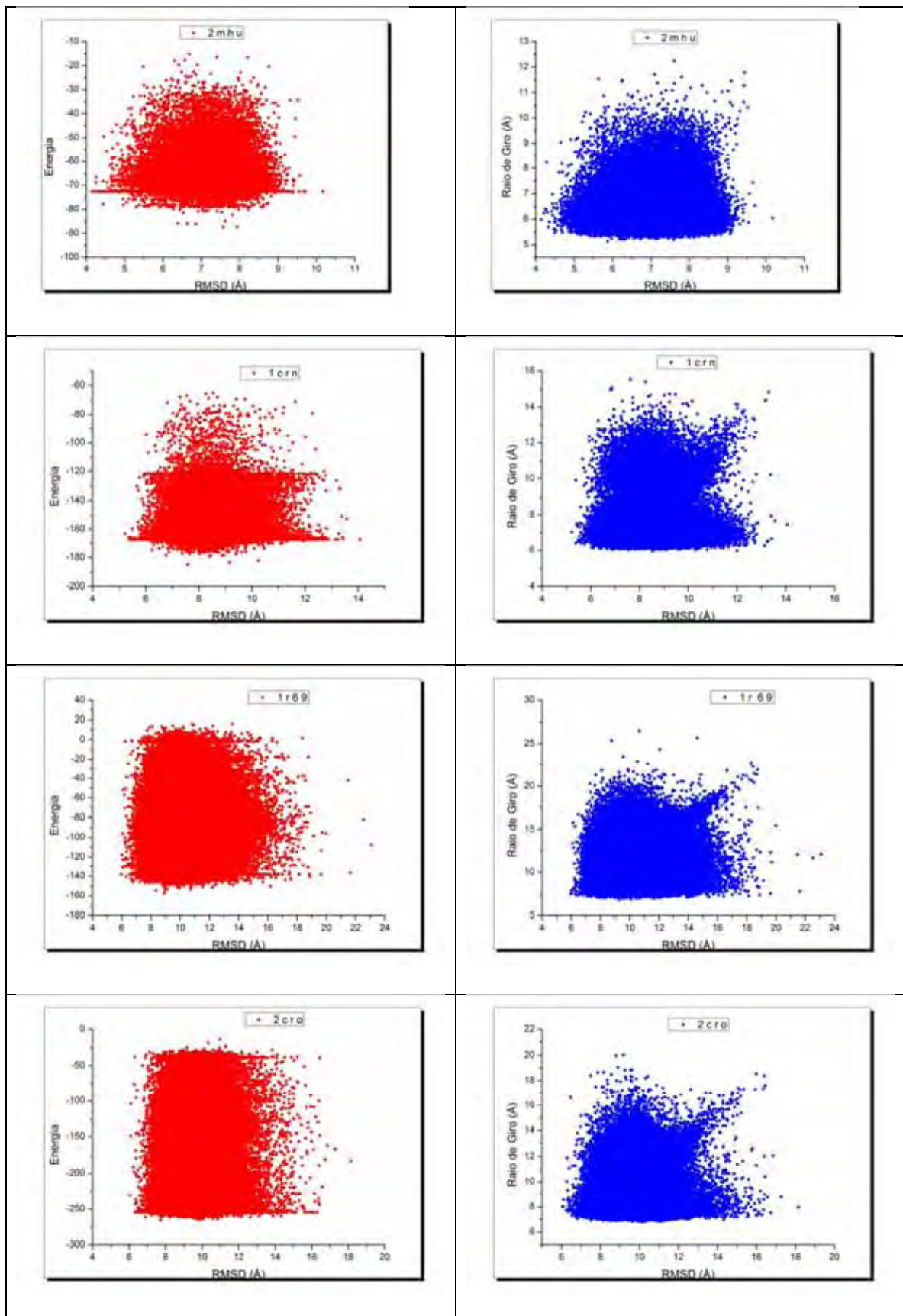
$$\beta = 1,00$$



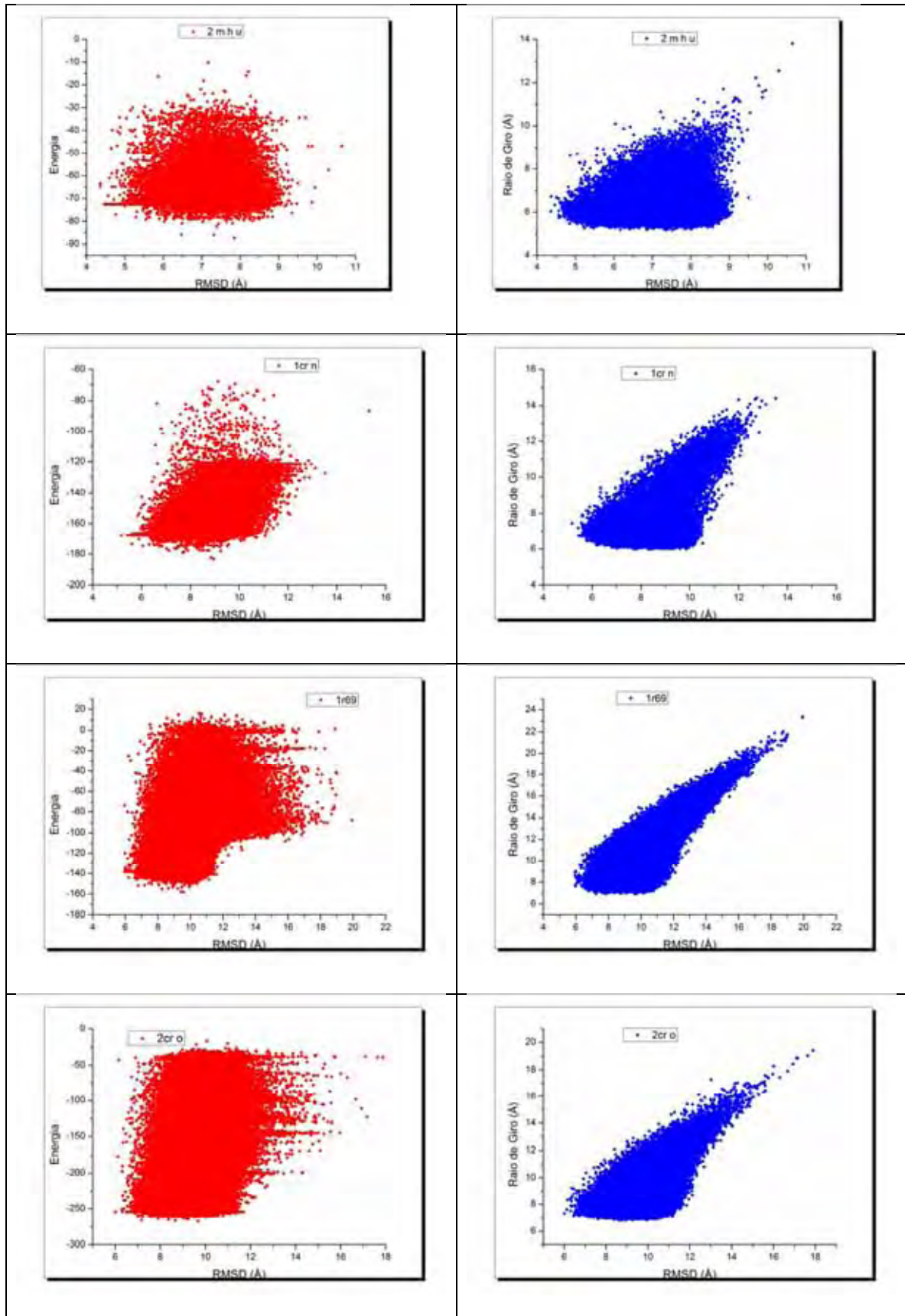
$$\beta = 1,02$$



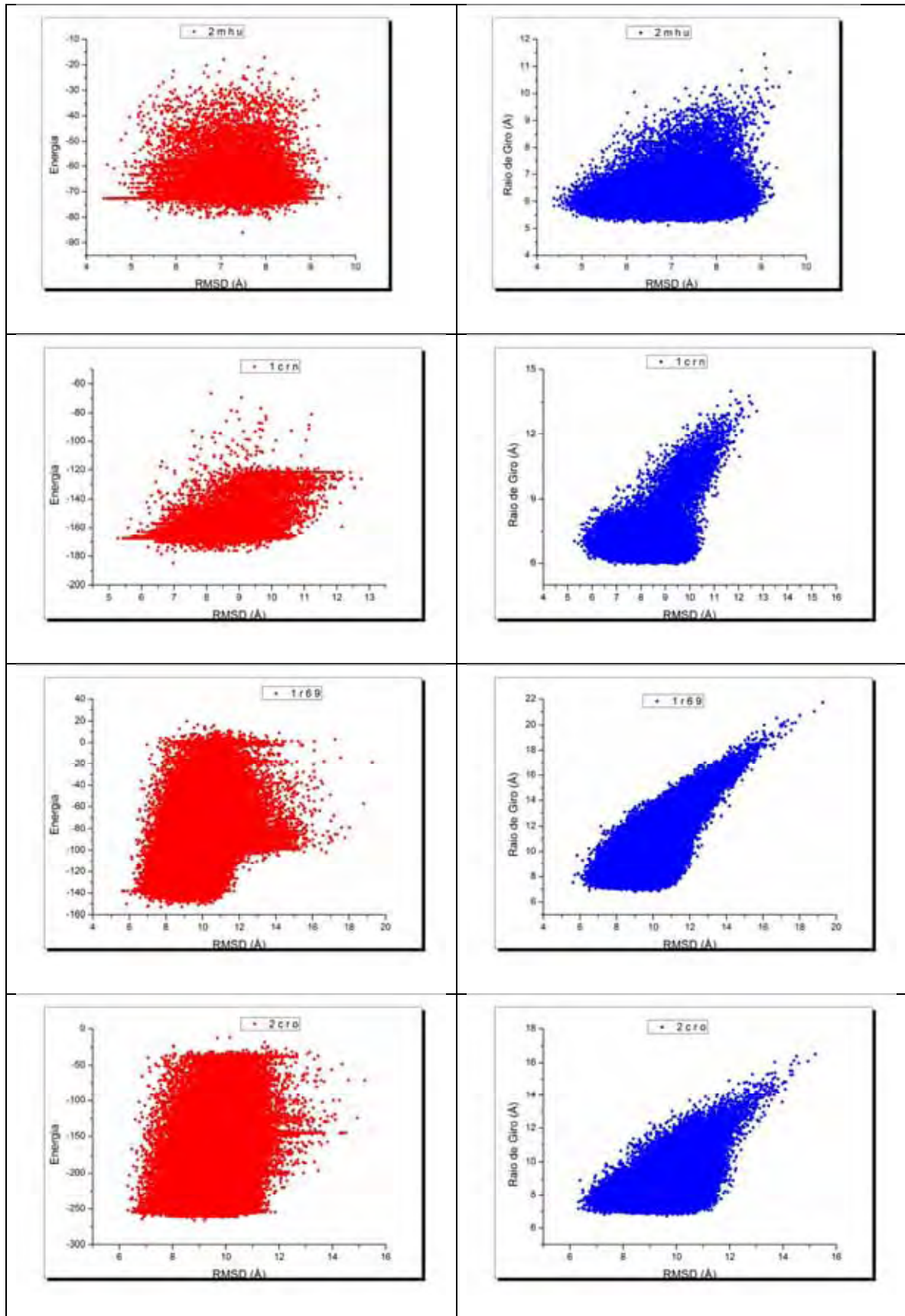
$$\beta = 1,50$$



$$\beta = 2,00$$



$$\beta = 4,00$$



Apêndice B

Algoritmo de Dinâmica para o Estudo do Estado de Transição da Proteína CI2

```

*****
*   Programa para realizar movimentos de Monte Carlo numa cadeia           *
*   proteica. A estrutura inicial foi gerada pelo algoritmo                 *
*   ChainGrowth.f; Nresiduos = 64, codigo PDB: 1coa                         *
*           RMSD = 8,0 A                                                    *
*           Rgiro = 9,7 A   Rnat= 10,8 A                                    *
*****

Program Dynamic

Real*8 rede(100000),rx(100000),ry(100000),rz(100000), vx(12), r
Real*8 x(64), y(64),z(64), Dnat(64,64), vy(12), vz(12), Eord(12)
Real*8 Tx(12), Ty(12), Tz(12), Dist(64,64), Eold(64), Enew(64)
Real*8 deltaE, dt,Txo(12), Tyo(12), Tzo(12), Dot_product
Real*8 xnorm, ynorm, znorm, somadist, Deter, beta, percent
Real*8 Energy(100000000), Rg(100000000)
Real*8 xnew(64), ynew(64), znew(64)
Real cos_alfa, alfa, cos_teta, teta
Integer i, j, k, mono, mono1, mono2, step, vaga, imark, contador
Integer flag, m, ix, iy, iz, count, point, Q

```



```

Character*3 aa(64)

Character*9 conta

Parameter (pi= 3.141592654)

*
-----
*

! Iniciamos com os arquivos de entrada para o Algoritmo:

!

! Arquivo 1: Estrutura inicial da proteina.

!     3: Codigo de 3 letras dos aminoacidos.

!     5: Rede cubo-octaedrica, ate a 15ª camada (12431 pontos)

!     7: Distancia entre pares de residuos no estado nativo.

!     9: Vetores da rede cubo-octahedrica

!*****

iseed= 1795      ! Semente para o gerador de nº aleatorio

n= 64           ! Numero de residuos da estrutura prot.

point= 55301    ! Numero de sitios da rede.

Open(3,File='aminoacid')

Read(3,*)(aa(mono), mono= 1,n)

Open(5,File='net25')

read(5,*)(rx(i),ry(i),rz(i),i= 1,point)

Open(7,File='map-Normalised')

Do 10 i= 1,n

    do 12 j= 1,n

        Dnat(i,j)= 0.0  ! Matriz de distancia entre

12  continue          ! residuos no estado nativo.

10  continue

    do 14 contador= 1,136  ! Numero de linhas do arq map-Normalised.

```

```

    read(7,*)mono1,mono2,Dnat(mono1,mono2)
! Os residuos 57 e 61 fazem contanto, Dnat(57,61)= 2.94 U = 7,9 A.
! Temos que informar para o algoritmo que a Dnat(61,57) tambem
! vale 2.94 U.
    do 16 i=1,n
        do 18 j=1,n
            if(i.eq.mono2.and.j.eq.mono1)Dnat(i,j)=Dnat(mono1,mono2)
18    continue
16    continue
14    continue
    Open(9,File='net1')
    read(9,*)(vx(i),vy(i),vz(i), i=1,12)
    Open(17,File='Contador',Status='new')
    Open(19,File='Energies',Status='new')
    Open(21,File='Rgiro',Status='new')
    Open(25,File='Central',Status='new')
    Open(27,File='Contato-Nativo',Status='new')
*


---


* Modificacoes a serem realizadas para cada ciclo:
    Open(1,File='structure-begin')
    Read(1,*)(x(mono),y(mono),z(mono), mono= 1,n)
    beta= 1.45
*


---


*
! Vamos batizar a rede com a estrutura de entrada, ou seja,
! informar para o algoritmo quais sao os sitios que serao
! ocupados pelos aminoacidos.

```

```

*
Do 20 i= 1,point
  rede(i)= 0.0      ! Sitio desocupado.
  do 22 mono= 1,n
    if(rx(i).eq.x(mono)
&      .and.ry(i).eq.y(mono)
&      .and.rz(i).eq.z(mono))then
      rede(i)= 1.0  ! A rede recebeu o monomero. Rede ocupada.
    endif
22  continue
20  continue
  contador= 0      ! Contador de tentativas de movimento.
  step= 0          ! Contador de passos de Monte Carlo.
50  r = rnd(iseed)
  mono= r*64 + 1
  Do 32 i= 1,n
    Eold(i)= 0.0
    Enew(i)= 0.0
    xnew(i)= 0.0
    ynew(i)= 0.0
    znew(i)= 0.0
    do 34 j= 1,n
      Dist(i,j)= 0.0
34  continue
32  Continue
*
! >>>> Iniciamos calculando a energia inicial do monomero

```

```

! na estrutura de origem.
Do 72 k= 1,n
  IF(k.eq.mono)goto 72 ! O monomero nao interage consigo mesmo.
  Dist(mono,k)= sqrt((x(mono)-x(k))**2 +(y(mono)-y(k))**2 +
&          (z(mono)-z(k))**2)
  If(Dist(mono,k).gt.0.8*Dnat(mono,k).and.
&   Dist(mono,k).lt.1.2*Dnat(mono,k))then
    Eold(mono)= Eold(mono) - 1.0
  Endif
72 Continue
  vaga = 0
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
*>>>> End Move: Movimento para os residuos da ponta:1, 2, 63 e 64.
!
!>>>>>> Inicio dos testes para movimentar o mono1:
  IF(mono.eq.1)then
    Do 24 j= 1,12
      Tx(j)= x(mono+1) + vx(j) ! mono+1 = mono2
      Ty(j)= y(mono+1) + vy(j)
      Tz(j)= z(mono+1) + vz(j)
    do 26 i= 1,point
      If(Tx(j).eq.rx(i)
&      .and.Ty(j).eq.ry(i)
&      .and.Tz(j).eq.rz(i))then
        if(rede(i).eq.0.0)then

```

- ! Os sitios da rede que estao alojando os monomeros foram marcados como
- ! ocupados (rede(i)= 1.0). Agora vamos eliminar o sitio que forma um
- ! angulo obtuso, 180°, com os monomeros 2 e 3.
- ! O produto escalar triplo entre os 3 vetores coplanares eh nulo.
- ! A distancia entre 2 sitios que formam angulo de 180° eh 2*sqrt(2.0)

```

Deter= ((x(mono+1)*y(mono+2)*Tz(j) +
&      y(mono+1)*z(mono+2)*Tx(j) +
&      z(mono+1)*x(mono+2)*Ty(j)) -
&      (z(mono+1)*y(mono+2)*Tx(j) +
&      y(mono+1)*x(mono+2)*Tz(j) +
&      x(mono+1)*z(mono+2)*Ty(j)))
Dist(mono+2,j)= sqrt((x(mono+2)-Tx(j))**2 +
&      (y(mono+2)-Ty(j))**2 +
&      (z(mono+2)-Tz(j))**2)
if(Deter.eq.0.0.and.Dist(mono+2,j).gt.2.8)goto 24

```

*>>>>>>> Calculando a energia do possivel candidato a receber o mono1:

```

vaga= vaga +1
do 28 k= mono+2,n
  Dist(j,k)= sqrt((Tx(j)-x(k))**2 +
&      (Ty(j)-y(k))**2 +(Tz(j)-z(k))**2)
  if(Dist(j,k).gt.0.8*Dnat(mono,k).and.
&      Dist(j,k).lt.1.2*Dnat(mono,k))then
    Enew(vaga)= Enew(vaga) - 1.0
  endif
28  continue
Tx(vaga)= Tx(j)
Ty(vaga)= Ty(j)

```

```

        Tz(vaga)= Tz(j)
    endif
Endif
26  continue
24  Continue
    If(vaga.eq.0)then
        goto 40      ! Finaliza este passo.
    else
        goto 30      ! Ordena as energias de cada sitio desocupado
    Endif
ENDIF
!      Final do movimento para o mono1.
*****
*
!      Inicio dos testes para movimentar o mono2:
    IF(mono.eq.2)Then
!      Podemos determinar o angulo entre o monomero 2 e seus vizinhos
!      adjacentes, mono1 e mono3, atraves do produto escalar entre os
!      vetores que descrevem esses monomeros em relacao ao primeiro.
!
        Dist(mono,mono-1)= sqrt((x(mono)-x(mono-1))**2 +
&          (y(mono)-y(mono-1))**2 + (z(mono)-z(mono-1))**2)
        Dist(mono,mono+1)= sqrt((x(mono)-x(mono+1))**2 +
&          (y(mono)-y(mono+1))**2 + (z(mono)-z(mono+1))**2)
        Dot_product= ((x(mono)-x(mono-1))*(x(mono)-x(mono+1))) +
&          ((y(mono)-y(mono-1))*(y(mono)-y(mono+1))) +
&          ((z(mono)-z(mono-1))*(z(mono)-z(mono+1)))

```

```
cos_alfa= (Dot_product/(Dist(mono,mono-1)*Dist(mono,mono+1)))
```

```
alfa= (180/pi)*acos(cos_alfa)
```

```
Do 116 j= 1,12
```

```
  Tx(j)= x(mono+1) + vx(j) ! mono-1 = mono1
```

```
  Ty(j)= y(mono+1) + vy(j)
```

```
  Tz(j)= z(mono+1) + vz(j)
```

```
do 118 i= 1,point
```

```
  If(Tx(j).eq.rx(i)
```

```
  &      .and.Ty(j).eq.ry(i)
```

```
  &      .and.Tz(j).eq.rz(i))then
```

```
    if(rede(i).eq.0.0)then
```

```
! Os sitios da rede que estao alojando os monomeros foram marcados como
```

```
! ocupados (rede(i)= 1.0). Agora vamos eliminar o sitio que forma um
```

```
! angulo obtuso, 180°, com os monomeros 3 e 4.
```

```
! O produto escalar triplo entre os 3 vetores coplanares eh nulo.
```

```
! A distancia entre 2 sitios que formam angulo de 180° eh 2*sqrt(2.0)
```

```
*
```

```
  Deter= ((x(mono+1)*y(mono+2)*Tz(j) +
```

```
  &      y(mono+1)*z(mono+2)*Tx(j) +
```

```
  &      z(mono+1)*x(mono+2)*Ty(j)) -
```

```
  &      (z(mono+1)*y(mono+2)*Tx(j) +
```

```
  &      y(mono+1)*x(mono+2)*Tz(j) +
```

```
  &      x(mono+1)*z(mono+2)*Ty(j)))
```

```
  Dist(mono+2,j)= sqrt((x(mono+2)-Tx(j))**2 +
```

```
  &      (y(mono+2)-Ty(j))**2 +
```

```
  &      (z(mono+2)-Tz(j))**2)
```

```
  if(Deter.eq.0.0.and.Dist(mono+2,j).gt.2.8)goto 116
```

*

!>>>>> Para um dado ponto central os seus vizinhos adjacentes

! podem formar um angulo de 60°, 90° ou 120°.

$$\text{Dist}(j, \text{mono}-1) = \sqrt{(\text{Tx}(j) - x(\text{mono}-1))^2 +$$

$$\& \quad (\text{Ty}(j) - y(\text{mono}-1))^2 + (\text{Tz}(j) - z(\text{mono}-1))^2}$$

$$\text{Dist}(j, \text{mono}+1) = \sqrt{(\text{Tx}(j) - x(\text{mono}+1))^2 +$$

$$\& \quad (\text{Ty}(j) - y(\text{mono}+1))^2 + (\text{Tz}(j) - z(\text{mono}+1))^2}$$

$$\text{Dot_product} = ((\text{Tx}(j) - x(\text{mono}-1)) * (\text{Tx}(j) - x(\text{mono}+1))) +$$

$$\& \quad ((\text{Ty}(j) - y(\text{mono}-1)) * (\text{Ty}(j) - y(\text{mono}+1))) +$$

$$\& \quad ((\text{Tz}(j) - z(\text{mono}-1)) * (\text{Tz}(j) - z(\text{mono}+1)))$$

$$\text{cos_teta} = (\text{Dot_product} / (\text{Dist}(j, \text{mono}-1) * \text{Dist}(j, \text{mono}+1)))$$

$$\text{teta} = (180/\pi) * \text{acos}(\text{cos_teta})$$

*

! A fim de manter as ligacoes, entre residuos adjacentes, em 3,8A

! o novo sitio deve manter o mesmo angulo com os monomeros 1 e 3.

! Logo, somente calculamos a energia dos sitios desocupados que

! satisfazem essa condicao.

*

if(ifix(alfa).eq.ifix(teta))then

vaga= vaga +1

do 120 k= mono+3,n

$$\text{Dist}(j, k) = \sqrt{(\text{Tx}(j) - x(k))^2 +$$

$$\& \quad (\text{Ty}(j) - y(k))^2 + (\text{Tz}(j) - z(k))^2}$$

if(Dist(j,k).gt.0.8*Dnat(mono,k).and.

& \quad Dist(j,k).lt.1.2*Dnat(mono,k))then

Enew(vaga)= Enew(vaga) - 1.0

endif


```

120         continue

           Tx(vaga)= Tx(j)

           Ty(vaga)= Ty(j)

           Tz(vaga)= Tz(j)

           endif

           endif

           Endif

118         continue

116         Continue

           If(vaga.eq.0)then

             goto 40      ! Finaliza este passo.

           else

             goto 30      ! Ordena as energias de cada sitio desocupado

           Endif

           ENDIF

!         Final do movimento para o mono2.

*****

*

!         Inicio dos testes para movimentar o mono63:

           IF(mono.eq.63)Then

!         Podemos determinar o angulo entre o monomero 63 e seus vizinhos

!         adjacentes, mono62 e mono64, atraves do produto escalar entre os

!         vetores que descrevem esses monomeros em relacao ao primeiro.

!

           Dist(mono,mono-1)= sqrt((x(mono)-x(mono-1))**2 +

&           (y(mono)-y(mono-1))**2 + (z(mono)-z(mono-1))**2)

           Dist(mono,mono+1)= sqrt((x(mono)-x(mono+1))**2 +

```

```

&      (y(mono)-y(mono+1))**2 + (z(mono)-z(mono+1))**2)
Dot_product= ((x(mono)-x(mono-1))*(x(mono)-x(mono+1))) +
&      ((y(mono)-y(mono-1))*(y(mono)-y(mono+1))) +
&      ((z(mono)-z(mono-1))*(z(mono)-z(mono+1)))
cos_alfa= (Dot_product/(Dist(mono,mono-1)*Dist(mono,mono+1)))
alfa= (180/pi)*acos(cos_alfa)
Do 122 j= 1,12
Tx(j)= x(mono+1) + vx(j) ! mono-1 = mono1
Ty(j)= y(mono+1) + vy(j)
Tz(j)= z(mono+1) + vz(j)
do 124 i= 1,point
  If(Tx(j).eq.rx(i)
&      .and.Ty(j).eq.ry(i)
&      .and.Tz(j).eq.rz(i))then
    if(rede(i).eq.0.0)then
! Os sitios da rede que estao alojando os monomeros foram marcados como
! ocupados (rede(i)= 1.0). Agora vamos eliminar o sitio que forma um
! angulo obtuso, 180°, com os monomeros 3 e 4.
! O produto escalar triplo entre os 3 vetores coplanares eh nulo.
! A distancia entre 2 sitios que formam angulo de 180° eh 2*sqrt(2.0)
*
Deter= ((x(mono+1)*y(mono+2)*Tz(j) +
&      y(mono+1)*z(mono+2)*Tx(j) +
&      z(mono+1)*x(mono+2)*Ty(j)) -
&      (z(mono+1)*y(mono+2)*Tx(j) +
&      y(mono+1)*x(mono+2)*Tz(j) +
&      x(mono+1)*z(mono+2)*Ty(j)))

```

```

Dist(mono+2,j)= sqrt((x(mono+2)-Tx(j))**2 +
&          (y(mono+2)-Ty(j))**2 +
&          (z(mono+2)-Tz(j))**2)
if(Deter.eq.0.0.and.Dist(mono+2,j).gt.2.8)goto 122
*
!>>>>> Para um dado ponto central os seus vizinhos adjacentes
!        podem formar um angulo de 60°, 90° ou 120°.
Dist(j,mono-1)= sqrt((Tx(j)-x(mono-1))**2 +
&          (Ty(j)-y(mono-1))**2 + (Tz(j)-z(mono-1))**2)
Dist(j,mono+1)= sqrt((Tx(j)-x(mono+1))**2 +
&          (Ty(j)-y(mono+1))**2 + (Tz(j)-z(mono+1))**2)
Dot_product= ((Tx(j)-x(mono-1))*(Tx(j)-x(mono+1))) +
&          ((Ty(j)-y(mono-1))*(Ty(j)-y(mono+1))) +
&          ((Tz(j)-z(mono-1))*(Tz(j)-z(mono+1))))
cos_teta= (Dot_product/(Dist(j,mono-1)*Dist(j,mono+1)))
teta =(180/pi)*acos(cos_teta)
*
! A fim de manter as ligacoes, entre residuos adjacentes, em 3,8A
! o novo sitio deve manter o mesmo angulo com os monomeros 1 e 3.
! Logo, somente calculamos a energia dos sitios desocupados que
! satisfazem essa condicao.
*
if(ifix(alfa).eq.ifix(teta))then
vaga= vaga +1
do 126 k= mono-3,1,-1
Dist(j,k)= sqrt((Tx(j)-x(k))**2 +
&          (Ty(j)-y(k))**2 + (Tz(j)-z(k))**2)

```

```

        if(Dist(j,k).gt.0.8*Dnat(mono,k).and.
&          Dist(j,k).lt.1.2*Dnat(mono,k))then
            Enew(vaga)= Enew(vaga) - 1.0
        endif
126      continue
        Tx(vaga)= Tx(j)
        Ty(vaga)= Ty(j)
        Tz(vaga)= Tz(j)
    endif
  endif
Endif
124  continue
122  Continue
    If(vaga.eq.0)then
        goto 40      ! Finaliza este passo.
    else
        goto 30      ! Ordena as energias de cada sitio desocupado
    Endif
ENDIF
!      Final do movimento para o mono63.
*****
*
!      Inicio dos testes para movimentar o mono64:
IF(mono.eq.64)then
    Do 60 j= 1,12
        Tx(j)= x(mono-1) + vx(j) ! mono-1 = mono63
        Ty(j)= y(mono-1) + vy(j)

```

Tz(j)= z(mono-1) + vz(j)

do 62 i= 1,point

 If(Tx(j).eq.rx(i)

& .and.Ty(j).eq.ry(i)

& .and.Tz(j).eq.rz(i))then

 if(rede(i).eq.0.0)then

- ! Os sitios da rede que estao alojando os monomeros foram marcados como
- ! ocupados (rede(i)= 1.0). Agora vamos elimiar o sitio que forma um
- ! angulo obtuso, 180°, com os monomeros 63 e 62.
- ! O produto escalar triplo entre os 3 vetores coplanares eh nulo.
- ! A distancia entre 2 sitios que formam angulo de 180° eh 2*sqrt(2.0)

 Deter= ((x(mono-1)*y(mono-2)*Tz(j) +

& y(mono-1)*z(mono-2)*Tx(j) +

& z(mono-1)*x(mono-2)*Ty(j)) -

& (z(mono-1)*y(mono-2)*Tx(j) +

& y(mono-1)*x(mono-2)*Tz(j) +

& x(mono-1)*z(mono-2)*Ty(j)))

 Dist(mono-2,j)= sqrt((x(mono-2)-Tx(j))**2 +

& (y(mono-2)-Ty(j))**2 +

& (z(mono-2)-Tz(j))**2)

 if(Deter.eq.0.0.and.Dist(mono-2,j).gt.2.8)goto 60

*>>>>>>>> Calculando a energia do possivel candidato a receber o mono64:

 vaga= vaga +1

 do 64 k= mono-3,1,-1

 Dist(j,k)= sqrt((Tx(j)-x(k))**2 +

& (Ty(j)-y(k))**2 +(Tz(j)-z(k))**2)

 if(Dist(j,k).gt.0.8*Dnat(mono,k).and.

! podem formar um angulo de 60°, 90° ou 120°.

*

! Podemos determinar o angulo entre o monomero selecionado e

! seus vizinhos adjacentes, mono-1 e mono+1, atraves do produto escalar

! entre os vetores que descrevem esses monomeros em relacao ao primeiro.

!

Dist(mono,mono-1)= sqrt((x(mono)-x(mono-1))**2 +

& (y(mono)-y(mono-1))**2 + (z(mono)-z(mono-1))**2)

Dist(mono,mono+1)= sqrt((x(mono)-x(mono+1))**2 +

& (y(mono)-y(mono+1))**2 + (z(mono)-z(mono+1))**2)

Dot_product= ((x(mono)-x(mono-1))*(x(mono)-x(mono+1))) +

& ((y(mono)-y(mono-1))*(y(mono)-y(mono+1))) +

& ((z(mono)-z(mono-1))*(z(mono)-z(mono+1)))

cos_alfa= (Dot_product/(Dist(mono,mono-1)*Dist(mono,mono+1)))

alfa= (180/pi)*acos(cos_alfa)

*

!>>>> Para cada angulo temos um movimento diferente. Iniciamos

! com o angulo de 60°.

*

IF(ifix(alfa).eq.60)then

Do 74 j= 1,12

Tx(j)= x(mono-1) + vx(j)

Ty(j)= y(mono-1) + vy(j)

Tz(j)= z(mono-1) + vz(j)

do 76 i= 1,point

If(Tx(j).eq.rx(i)

& .and.Ty(j).eq.ry(i)

```

& .and.Tz(j).eq.rz(i))then
    if(rede(i).eq.0.0)then
*
! Os sitios da rede que estao alojando os monomeros foram marcados como
! ocupados (rede(i)= 1.0). Agora vamos eliminar o sitio que forma um
! angulo obtuso, 180°, com os monomeros adjacentes ao selecionado.
! O produto escalar triplo entre os 3 vetores coplanares eh nulo.
! A distancia entre 2 sitios que formam angulo de 180° eh 2*sqrt(2.0)
! Primeiramente, olhamos para os 2 monomeros anteriores ao selecionado:
*
    Deter= ((x(mono-1)*y(mono-2)*Tz(j) +
&          y(mono-1)*z(mono-2)*Tx(j) +
&          z(mono-1)*x(mono-2)*Ty(j)) -
&          (z(mono-1)*y(mono-2)*Tx(j) +
&          y(mono-1)*x(mono-2)*Tz(j) +
&          x(mono-1)*z(mono-2)*Ty(j)))
    Dist(mono-2,j)= sqrt((x(mono-2)-Tx(j))**2 +
&          (y(mono-2)-Ty(j))**2 +
&          (z(mono-2)-Tz(j))**2)
    if(Deter.eq.0.0.and.Dist(mono-2,j).gt.2.8)goto 74
*
! Agora, olhamos para os 2 monomeros posteriores:
*
    Deter= ((x(mono+1)*y(mono+2)*Tz(j) +
&          y(mono+1)*z(mono+2)*Tx(j) +
&          z(mono+1)*x(mono+2)*Ty(j)) -
&          (z(mono+1)*y(mono+2)*Tx(j) +

```



```

&          y(mono+1)*x(mono+2)*Tz(j) +
&          x(mono+1)*z(mono+2)*Ty(j))
          Dist(mono+2,j)= sqrt((x(mono+2)-Tx(j))**2 +
&          (y(mono+2)-Ty(j))**2 +
&          (z(mono+2)-Tz(j))**2)
          if(Deter.eq.0.0.and.Dist(mono+2,j).gt.2.8)goto 74
*
!  O novo sitio para o monomero selecionado deve respeitar a restricao
!  de que as ligacoes com os monomeros adjacentes seja mantida em
!  3,8 A ou sqrt(2.0). Sendo assim devemos manter o angulo entre os
!  monomeros, a saber, alfa = 60°.
!>>>>>>> Calculando o angulo entre o sitio vazio e os mono-1 e mono+1:
*

```

```

          Dist(j,mono-1)= sqrt((Tx(j)-x(mono-1))**2 +
&          (Ty(j)-y(mono-1))**2 + (Tz(j)-z(mono-1))**2)
          Dist(j,mono+1)= sqrt((Tx(j)-x(mono+1))**2 +
&          (Ty(j)-y(mono+1))**2 + (Tz(j)-z(mono+1))**2)
          Dot_product= ((Tx(j)-x(mono-1))*(Tx(j)-x(mono+1))) +
&          ((Ty(j)-y(mono-1))*(Ty(j)-y(mono+1))) +
&          ((Tz(j)-z(mono-1))*(Tz(j)-z(mono+1)))
          cos_alfa= (Dot_product/(Dist(j,mono-1)*Dist(j,mono+1)))
          alfa =(180/pi)*acos(cos_alfa)
          if(ifix(alfa).ne.60)goto 74
*

```

```

*>>>>>>>> Calculando a energia do possivel candidato a receber o monomero:

```

```

          vaga= vaga +1

```

```

          do 78 k= 1,n ! O monomero nao interage

```

```

IF(k.eq.mono)goto 78 ! consigo mesmo.
Dist(j,k)= sqrt((Tx(j)-x(k))**2 +
&          (Ty(j)-y(k))**2 +(Tz(j)-z(k))**2)
if(Dist(j,k).gt.0.8*Dnat(mono,k).and.
&      Dist(j,k).lt.1.2*Dnat(mono,k))then
    Enew(vaga)= Enew(vaga) - 1.0
endif
78    continue
    Tx(vaga)= Tx(j)
    Ty(vaga)= Ty(j)
    Tz(vaga)= Tz(j)
endif
Endif    ! Final da verificacao do sitio vazio.
76    continue
74    Continue
ENDIF    ! Final da anlise para o angulo 60°.
*
! Inicio do movimento para os monomeros que fazem angulo de 90°
! com seus vizinhos adjacentes:
*
IF(ifix(alfa).eq.90)then
Do 86 j= 1,12
    Tx(j)= x(mono-1) + vx(j)
    Ty(j)= y(mono-1) + vy(j)
    Tz(j)= z(mono-1) + vz(j)
do 88 i= 1,point
    If(Tx(j).eq.rx(i)

```

```

&      .and.Ty(j).eq.ry(i)
&      .and.Tz(j).eq.rz(i))then
      if(rede(i).eq.0.0)then
*
!      Eliminando os angulos de 180° entre:
!      mono-2 >> mono-1 >> sitio desocupado e
!      mono+2 >> mono+1 >> sitio desocupado.
!      Primeiramente, olhamos para os 2 monomeros anteriores ao selecionado:
*
      Deter= ((x(mono-1)*y(mono-2)*Tz(j) +
&          y(mono-1)*z(mono-2)*Tx(j) +
&          z(mono-1)*x(mono-2)*Ty(j)) -
&          (z(mono-1)*y(mono-2)*Tx(j) +
&          y(mono-1)*x(mono-2)*Tz(j) +
&          x(mono-1)*z(mono-2)*Ty(j)))
      Dist(mono-2,j)= sqrt((x(mono-2)-Tx(j))**2 +
&          (y(mono-2)-Ty(j))**2 +
&          (z(mono-2)-Tz(j))**2)
      if(Deter.eq.0.0.and.Dist(mono-2,j).gt.2.8)goto 86
*
!      Agora, olhamos para os 2 monomeros posteriores:
*
      Deter= ((x(mono+1)*y(mono+2)*Tz(j) +
&          y(mono+1)*z(mono+2)*Tx(j) +
&          z(mono+1)*x(mono+2)*Ty(j)) -
&          (z(mono+1)*y(mono+2)*Tx(j) +
&          y(mono+1)*x(mono+2)*Tz(j) +

```

```

&          x(mono+1)*z(mono+2)*Ty(j))
          Dist(mono+2,j)= sqrt((x(mono+2)-Tx(j))**2 +
&          (y(mono+2)-Ty(j))**2 +
&          (z(mono+2)-Tz(j))**2)
          if(Deter.eq.0.0.and.Dist(mono+2,j).gt.2.8)goto 86
*
!  O novo sitio para o monomero selecionado deve respeitar a restricao
!  de que as ligacoes com os monomeros adjacentes seja mantida em
!  3,8 A ou sqrt(2.0). Sendo assim devemos manter o angulo entre os
!  monomeros, a saber, alfa = 90°.
!>>>>>>> Calculando o angulo entre o sitio vazio e os mono-1 e mono+1:
*
          Dist(j,mono-1)= sqrt((Tx(j)-x(mono-1))**2 +
&          (Ty(j)-y(mono-1))**2 + (Tz(j)-z(mono-1))**2)
          Dist(j,mono+1)= sqrt((Tx(j)-x(mono+1))**2 +
&          (Ty(j)-y(mono+1))**2 + (Tz(j)-z(mono+1))**2)
          Dot_product= ((Tx(j)-x(mono-1))*(Tx(j)-x(mono+1))) +
&          ((Ty(j)-y(mono-1))*(Ty(j)-y(mono+1))) +
&          ((Tz(j)-z(mono-1))*(Tz(j)-z(mono+1)))
          cos_alfa= (Dot_product/(Dist(j,mono-1)*Dist(j,mono+1)))
          alfa =(180/pi)*acos(cos_alfa)
          if(ifix(alfa).ne.90)goto 86
*
!>>>>>>>> Calculando a energia do possivel candidato a receber o monomero:
          vaga= vaga +1
          do 90 k= 1,n ! O monomero nao interage
          IF(k.eq.mono)goto 90 ! consigo mesmo.

```

```

        Dist(j,k)= sqrt((Tx(j)-x(k))**2 +
&           (Ty(j)-y(k))**2 +(Tz(j)-z(k))**2)
        if(Dist(j,k).gt.0.8*Dnat(mono,k).and.
&           Dist(j,k).lt.1.2*Dnat(mono,k))then
            Enew(vaga)= Enew(vaga) - 1.0
        endif
90    continue

        Tx(vaga)= Tx(j)
        Ty(vaga)= Ty(j)
        Tz(vaga)= Tz(j)
    endif
Endif
88    continue
86    Continue
ENDIF      ! Final da anlise para o angulo 90°.
*
!  Inicio do movimento para os monomeros que fazem angulo de 120°
!      com seus vizinhos adjacentes:
*

IF(ifix(alfa).eq.120)then
    Do 110 j= 1,12
        Tx(j)= x(mono-1) + vx(j)
        Ty(j)= y(mono-1) + vy(j)
        Tz(j)= z(mono-1) + vz(j)
    do 112 i= 1,point
        If(Tx(j).eq.rx(i)
&           .and.Ty(j).eq.ry(i)

```

```

&      .and.Tz(j).eq.rz(i))then
      if(rede(i).eq.0.0)then
*
!      Eliminando os angulos de 180° entre:
!      mono-2 >> mono-1 >> sitio desocupado e
!      mono+2 >> mono+1 >> sitio desocupado.
!      Primeiramente, olhamos para os 2 monomeros anteriores ao selecionado:
*
      Deter= ((x(mono-1)*y(mono-2)*Tz(j) +
&          y(mono-1)*z(mono-2)*Tx(j) +
&          z(mono-1)*x(mono-2)*Ty(j)) -
&          (z(mono-1)*y(mono-2)*Tx(j) +
&          y(mono-1)*x(mono-2)*Tz(j) +
&          x(mono-1)*z(mono-2)*Ty(j)))
      Dist(mono-2,j)= sqrt((x(mono-2)-Tx(j))**2 +
&          (y(mono-2)-Ty(j))**2 +
&          (z(mono-2)-Tz(j))**2)
      if(Deter.eq.0.0.and.Dist(mono-2,j).gt.2.8)goto 110
*
!      Agora, olhamos para os 2 monomeros posteriores:
*
      Deter= ((x(mono+1)*y(mono+2)*Tz(j) +
&          y(mono+1)*z(mono+2)*Tx(j) +
&          z(mono+1)*x(mono+2)*Ty(j)) -
&          (z(mono+1)*y(mono+2)*Tx(j) +
&          y(mono+1)*x(mono+2)*Tz(j) +
&          x(mono+1)*z(mono+2)*Ty(j)))

```

```

      Dist(mono+2,j)= sqrt((x(mono+2)-Tx(j))**2 +
&
      (y(mono+2)-Ty(j))**2 +
&
      (z(mono+2)-Tz(j))**2)
      if(Deter.eq.0.0.and.Dist(mono+2,j).gt.2.8)goto 110
*
!  O novo sitio para o monomero selecionado deve respeitar a restricao
!  de que as ligacoes com os monomeros adjacentes seja mantida em
!  3,8 A ou sqrt(2.0). Sendo assim devemos manter o angulo entre os
!  monomeros, a saber, alfa = 120°.
!>>>>>>> Calculando o angulo entre o sitio vazio e os mono-1 e mono+1:

```

```

*
      Dist(j,mono-1)= sqrt((Tx(j)-x(mono-1))**2 +
&
      (Ty(j)-y(mono-1))**2 + (Tz(j)-z(mono-1))**2)
      Dist(j,mono+1)= sqrt((Tx(j)-x(mono+1))**2 +
&
      (Ty(j)-y(mono+1))**2 + (Tz(j)-z(mono+1))**2)
      Dot_product= ((Tx(j)-x(mono-1))*(Tx(j)-x(mono+1))) +
&
      ((Ty(j)-y(mono-1))*(Ty(j)-y(mono+1))) +
&
      ((Tz(j)-z(mono-1))*(Tz(j)-z(mono+1))))
      cos_alfa= (Dot_product/(Dist(j,mono-1)*Dist(j,mono+1)))
      alfa =(180/pi)*acos(cos_alfa)
      if(ifix(alfa).ne.120)goto 110

```

```

*
!>>>>>>>> Calculando a energia do possivel candidato a receber o monomero:

```

```

      vaga= vaga +1
      do 114 k= 1,n ! O monomero nao interage
      IF(k.eq.mono)goto 114 ! consigo mesmo.
      Dist(j,k)= sqrt((Tx(j)-x(k))**2 +

```

```

&          (Ty(j)-y(k))**2 +(Tz(j)-z(k))**2)
          if(Dist(j,k).gt.0.8*Dnat(mono,k).and.
&          Dist(j,k).lt.1.2*Dnat(mono,k))then
          Enew(vaga)= Enew(vaga) - 1.0
          endif
114      continue
          Tx(vaga)= Tx(j)
          Ty(vaga)= Ty(j)
          Tz(vaga)= Tz(j)
          endif
      Endif
112      continue
110      Continue
      ENDIF          ! Final da analise para o angulo 120°.
      If(vaga.eq.0)then
          goto 40          ! Finaliza este passo.
      else
          goto 30          ! Ordena as energias de cada sitio desocupado
      Endif
      ENDIF          ! Final da condicao de movimento para os
!          monomeros 2 < mono < 63.
!////////////////////////////////////
!////////////////////////////////////
*
!  O movimento sera realizado atraves da comparacao entre as
!  energias Eold(mono) e Enew(vaga).
*

```



```

30 Do i= 1,vaga
    Eord(i)= 0.0      ! Vetor para ordenar, em ordem decrescente,
Enddo              ! as energias dos sitios disponiveis.

Do 80 i= 1,vaga
    imark= i        ! Marcador de energias.

    Eord(i)= Enew(i)

    do 82 j= 1,vaga

! Vamos ordenar as energias de cada sitio, em ordem decrescente,
! a fim de que o sitio de menor energia seja selecionado.

        If(i.ne.j)then    ! Nao compara um sitio com ele mesmo.

            if(Eord(i).le.Enew(j))then

                Eord(i)= Enew(j)

                imark= j

            endif

        Endif

82 continue

Enew(imark)= -1000.0 * float(imark)

Txo(i)= Tx(imark)    ! As coordenadas de cada vaga sao
Tyo(i)= Ty(imark)    ! ordenadas de acordo com a Energia
Tzo(i)= Tz(imark)    ! do sitio, Eord(vaga).

80 Continue

Do 84 i= vaga,1,-1

If(Eord(i).lt.Eold(mono))then

    x(mono)= Txo(i)    ! O sitio de menor energia recebe o
    y(mono)= Tyo(i)    ! mono selecionado.
    z(mono)= Tzo(i)

    step= step + 1

```

```

    flag= 0      ! Codigo para marcar a estrutura escrita.
    goto 48
else
    deltaE= Eord(i)-Eold(mono)
    prob= exp(-beta*deltaE)
Endif
r= rnd(iseed)
If(prob.gt.r)then
    x(mono)= Txo(i)  ! O sitio de maior enrgia recebe o
    y(mono)= Tyo(i)  ! mono selecionado.
    z(mono)= Tzo(i)
    step= step + 1
    flag= 0      ! Codigo para marcar a estrutura escrita.
    goto 48
Endif
84 Continue
*
!   Rebatizando a rede com a nova estrutura.
*
48 Do 36 i= 1,point
    rede(i)= 0.0    ! Sitio desocupado.
    do 38 mono= 1,n
        If(rx(i).eq.x(mono)
&      .and.ry(i).eq.y(mono)
&      .and.rz(i).eq.z(mono))then
            rede(i)= 1.0  ! Sitio ocupado por um monomero.
        Endif

```



```

endif
*
! Calculando e gravando a Energia da nova estrutura.
*

write(17,*)step ! Contador

Q= 0 ! Contador de contatos nativos.

Energy(step)= 0.0

Do 100 mono= 1,n-4

do 102 j= mono+3,n

Dist(mono,j)= sqrt((x(mono)-x(j))**2 +
& (y(mono)-y(j))**2 +(z(mono)-z(j))**2)

if(Dist(mono,j).gt.0.8*Dnat(mono,j).and.
& Dist(mono,j).lt.1.2*Dnat(mono,j))then

Energy(step)= Energy(step) - 1.0

Q= Q + 1

endif

102 continue

100 Continue

write(19,*)Energy(step) ! Energies

write(27,*)Q,Energy(step) ! Contato-Nativo

*

! Calculando e gravando o Raio de Giro da nova estrutura.

*

somaDist= 0.0

Do 104 mono= 1,n-1

do 106 j= mono+1,n

somaDist= somaDist + ((x(mono)-x(j))**2 +

```

```

&      (y(mono)-y(j))**2 + (z(mono)-z(j))**2)
106    continue
104    Continue

      Rg(step)= (3.8/sqrt(2.0))*(sqrt(somaDist)/n)
      write(21,108)step,Rg(step) ! Rgiro
108    format(i9,f7.2)

      ENDIF

      endif

!\\////////////////////////////////////
!////////////////////////////////////////
!

*****

*    Apos fazer varios movimentos na estrutura, esta vai se afastando
*
*    do centro da rede. A fim de evitar que a estrutura chegue nas
*
*    extremidades da rede (camada 25) faremos uma centralizacao da
*
*    estrutura a cada 10 passos.
*****

      If(x(32).eq.0.0.and.y(32).eq.0.0.and.z(32).eq.0.0)goto 40

      m= mod(step,10)

      if(m.eq.9)then

          xnorm= x(32)

          ynorm= y(32)

          znorm= z(32)

          do 138 mono= 1,n

              x(mono)= x(mono) - xnorm

              y(mono)= y(mono) - ynorm

              z(mono)= z(mono) - znorm

```

```
138 continue
*
!>>>>>>>  Estrutura Centralizada.  <<<<<<<<<<<<<<<<<<<<<<<<<<<<
*
! Rebatizando a rede com a nova estrutura.
*

count= 0           ! Verifica se os monomeros estao nos

Do 140 i= 1,point   ! sitios da rede.

rede(i)= 0.0      ! Sitio desocupado.

do 142 mono= 1,n

  If(rx(i).eq.x(mono)
&      .and.ry(i).eq.y(mono)
&      .and.rz(i).eq.z(mono))then

    rede(i)= 1.0 ! Sitio ocupado por um monomero.

    count= count + 1

  Endif

142 continue

140 Continue

  Write(25,*)step,count ! Central

endif

40 contador= contador + 1

*

! Condicao para realizar uma certa quantidade de movimentos para um
! dado valor de Temperatura.

*

If(step.lt.10000000)goto 50

percent= 100.0*step/contador
```

