



## **Simulação de Dinâmica Molecular – Fundamentos e Aplicações em Proteínas.**

Minicurso – VII SEMAQ

*Prof. Dr. Davi S Vieira*

*Universidade Federal do Rio Grande do Norte*

*Centro de Ciências Exatas e da Terra*

*Instituto de Química*

*Laboratório de Modelagem Molecular e Simulação Aplicados*



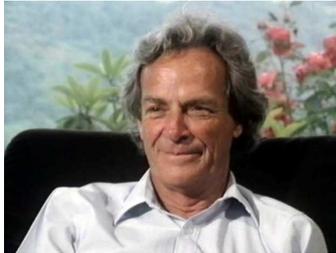






## Tópicos – Minicurso: Simulação Molecular

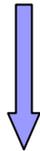
1. Fundamentos – Proteínas/Enzimas
2. Simulação Molecular – Introdução e Breve histórico
3. Mecânica Estatística e aproximações
4. Fundamentos de Dinâmica Molecular
5. Banco de Dados de Proteínas e Programas de Visualização de Proteínas.
6. Aplicações em proteínas.
7. Tutorial GROMACS



*Richard Feynman: Nobel Física em 1965 – conceito de Nanotecnologia.*

*O desenvolvimento nanotecnológico está vinculado ao surgimento de ferramentas necessárias para trabalhar com a matéria na escala de nanômetros (nm); átomos, moléculas, macromoléculas (1-100nm).*

**Nanotecnologia:** Matemática, Física, Química, Biologia e Engenharia



Problemas Biológicos



**Nanobiotecnologia ou Biotecnologia**



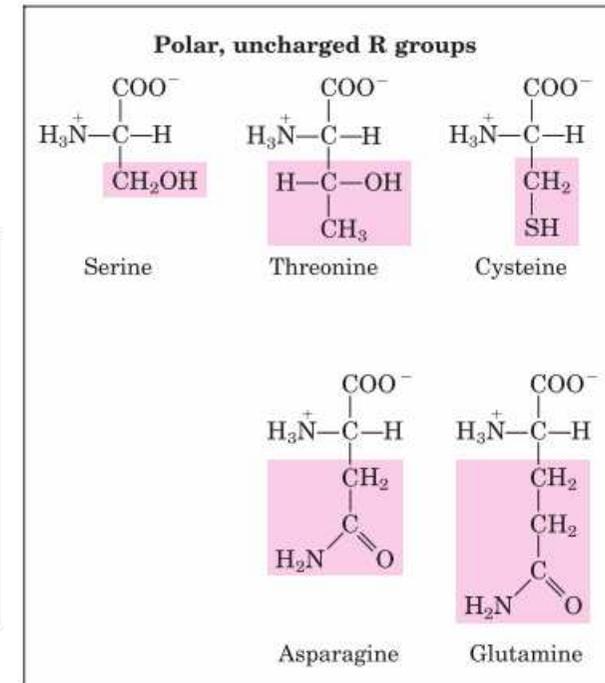
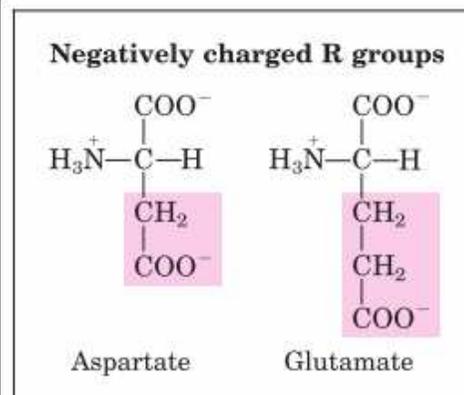
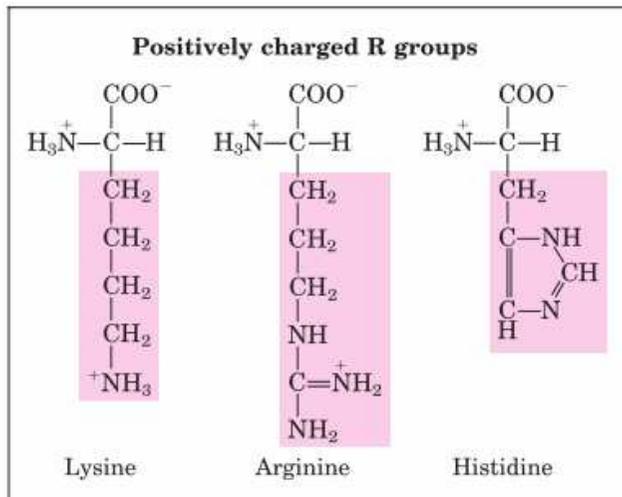
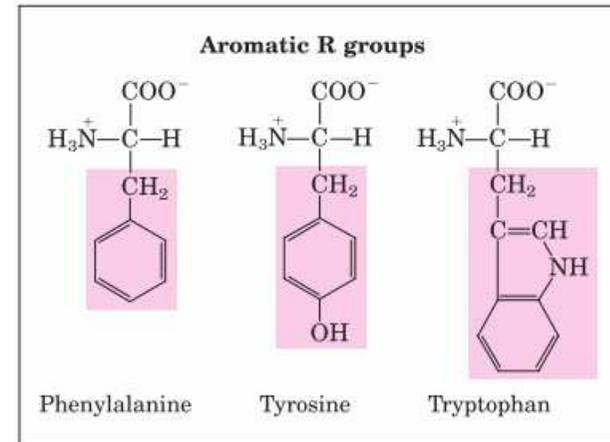
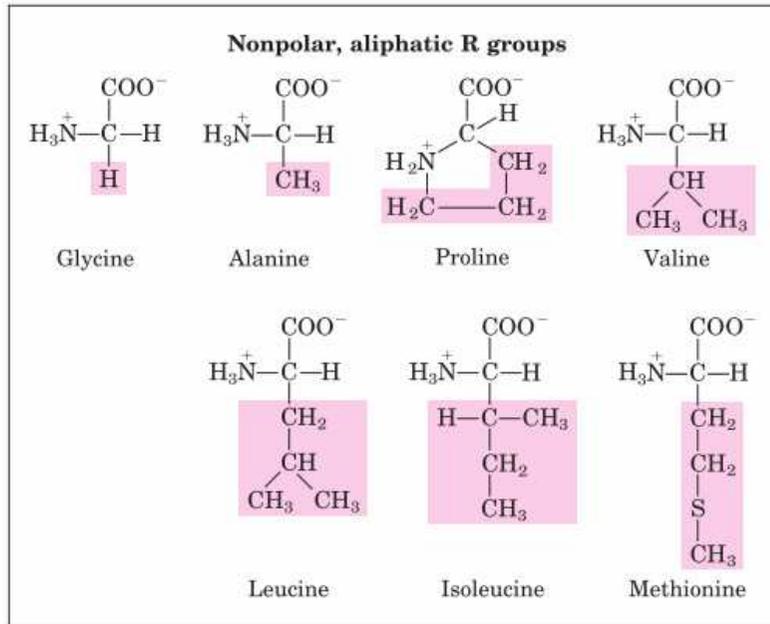
- Agricultura
- Medicina
- Farmácia
- Química
- Engenharia genética



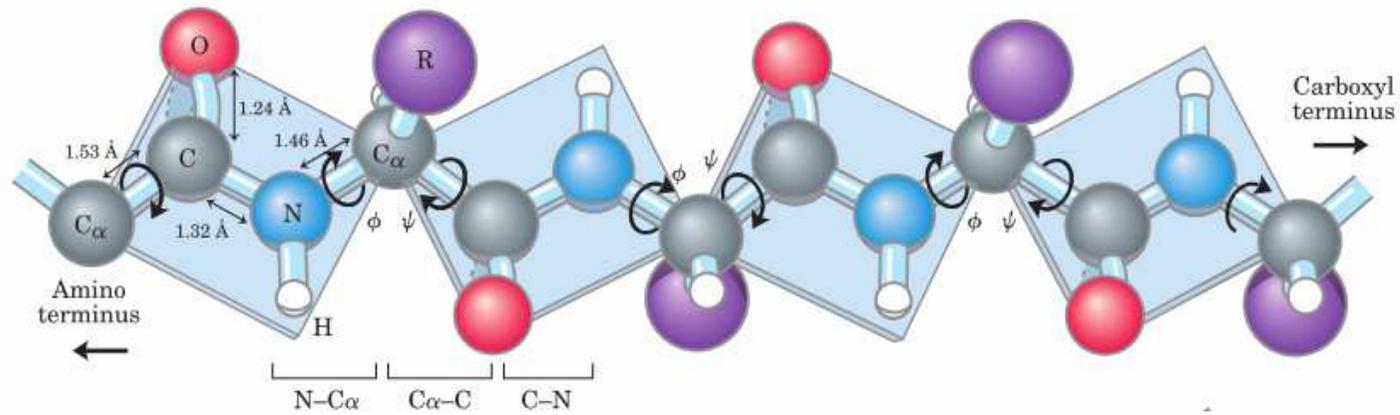
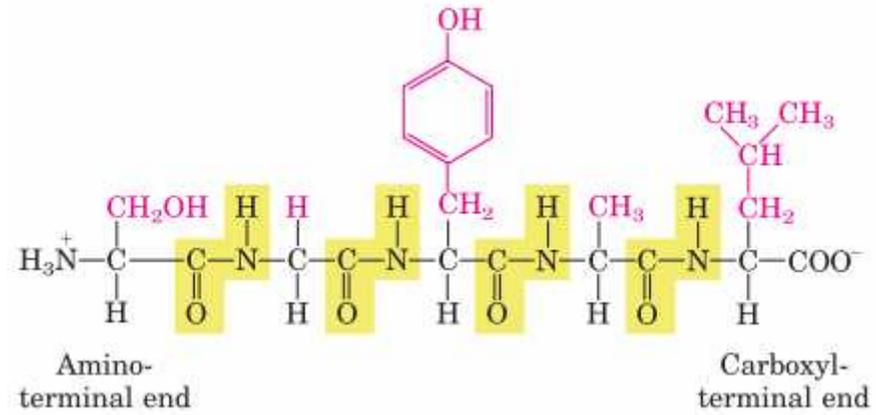
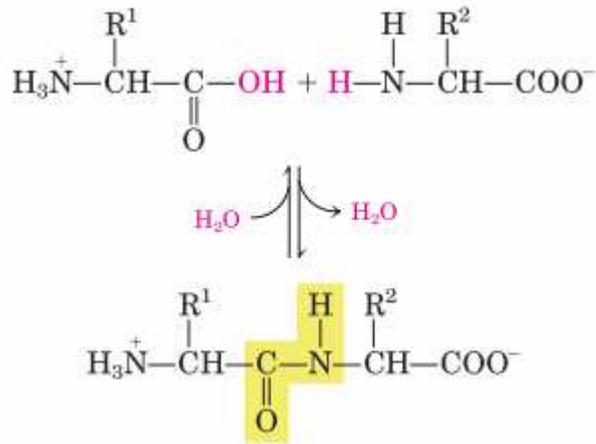
**Metodologias Computacionais**

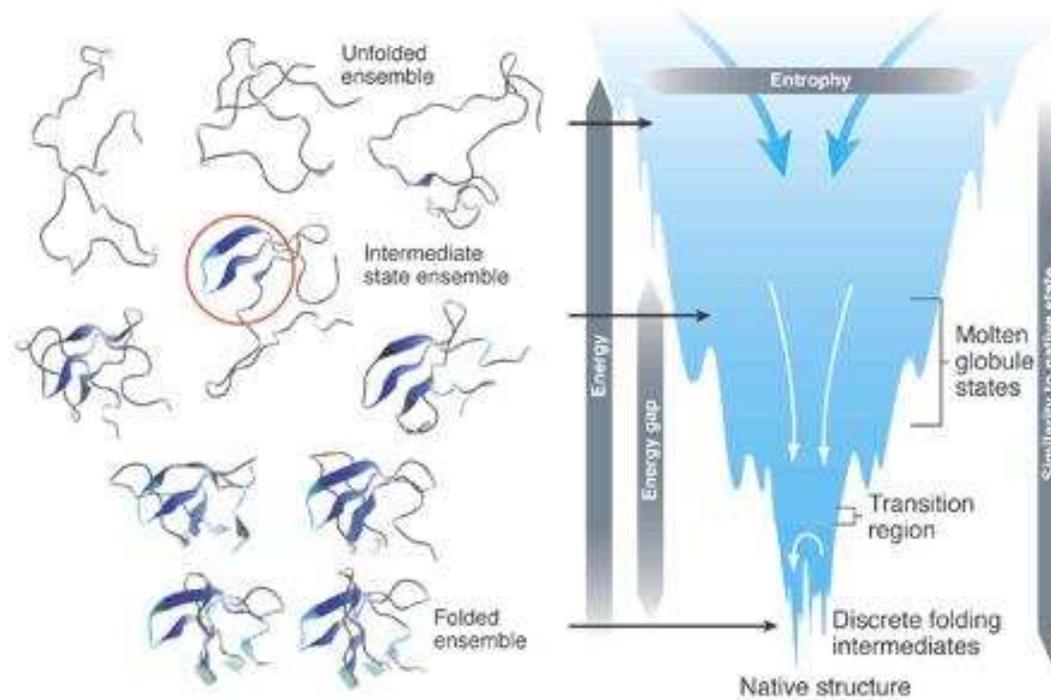
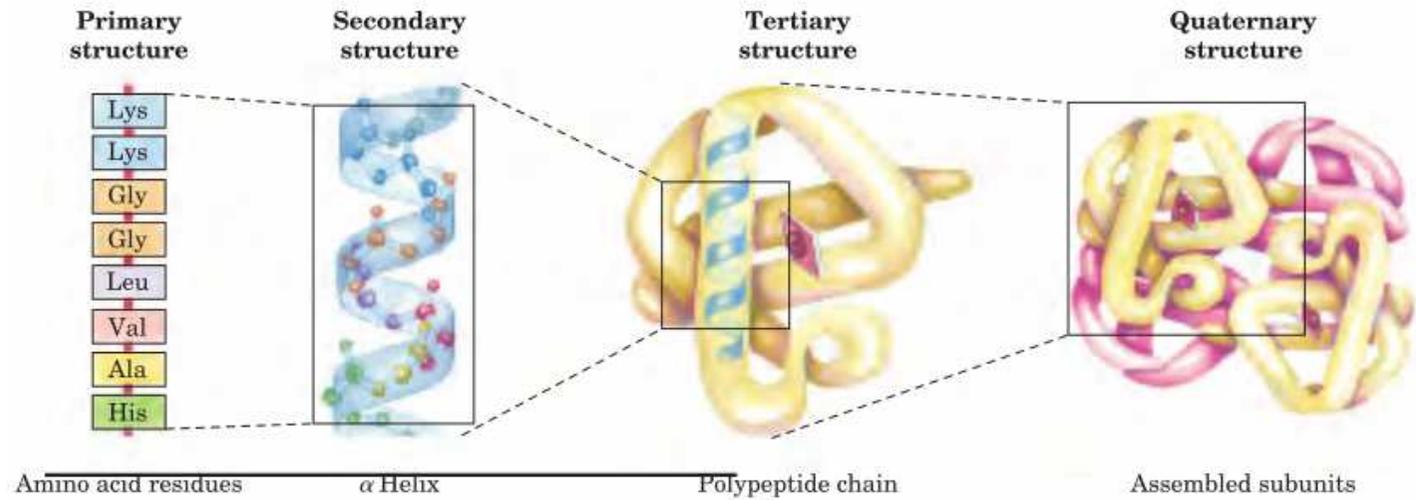


# Aminoácidos e Proteínas



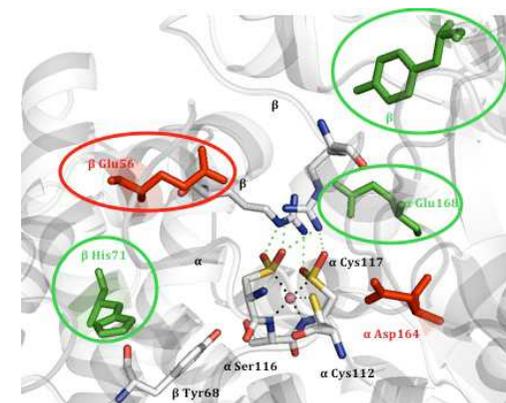
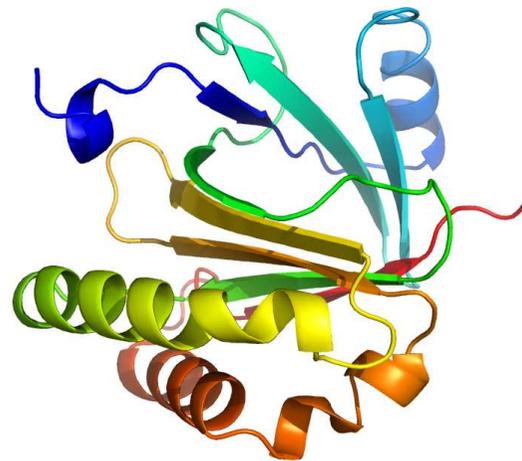
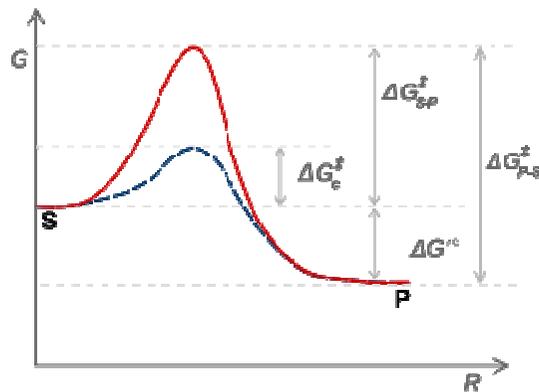
# Aminoácidos e Proteínas





# Enzimas

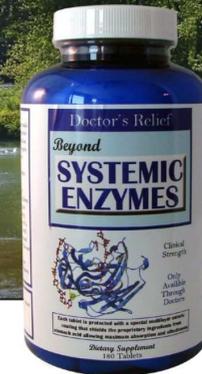
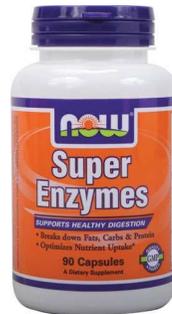
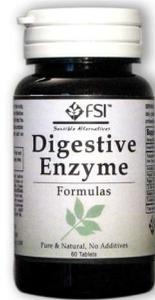
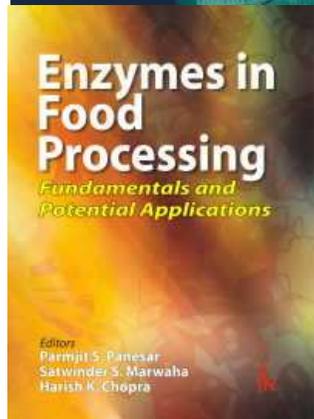
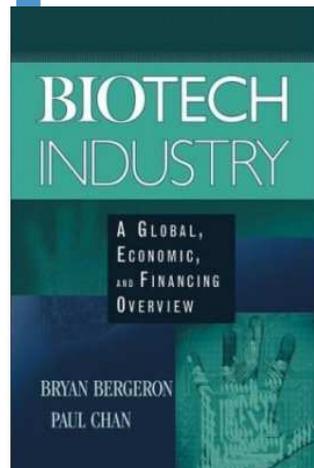
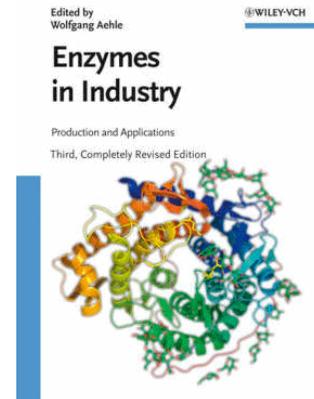
*Do ponto de vista nano(bio)tecnológico, as enzimas são consideradas nanomáquinas que produzem um produto ou uma ação desejada sobre um substrato.*

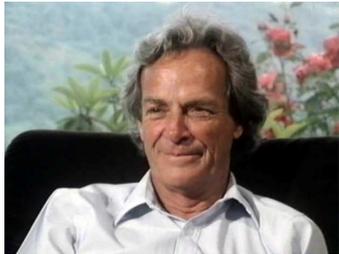


*Relação Estrutura/Atividade*



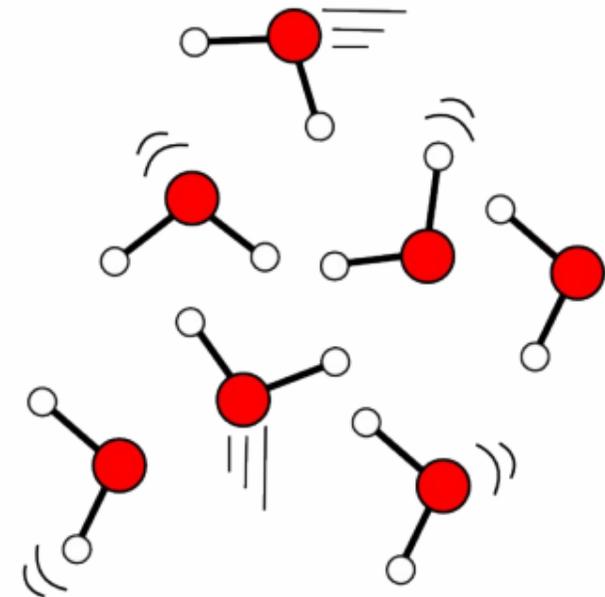
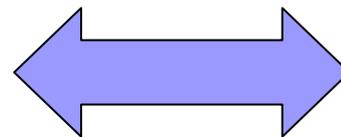
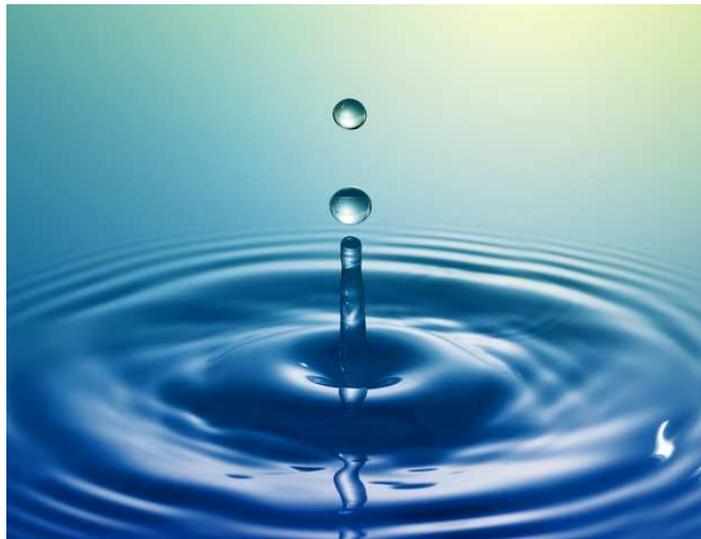
*Melhorias - Aplicação*





*Richard Feynman*: Nobel Física em 1965 – conceito de Nanotecnologia.

*“ Temos que levar adiante a ideia que TUDO pode ser compreendido a partir da movimentação dos átomos.” – Lectures on Physics (1963).*



## Agitando as moléculas!!



## Metodologia - Simulação molecular

Um sistema molecular está longe de ser estático – átomos e moléculas estão em constante movimentação, e principalmente no caso das biomoléculas, estas movimentações estão correlacionadas e determinam a função da molécula.

- ➡ *Ferramentas para prever a movimentação dos componentes do sistema.*
- ➡ *Experimento computacional (in silico)*
- ➡ *Visão atômico-molecular dos processos*
- ➡ **Metodologias: Estocástica e Determinística**

## Métodos de simulação mais usados

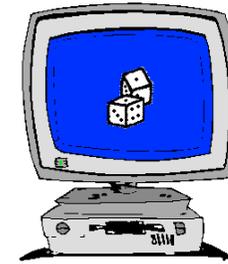
**Monte Carlo** (Estocástico): gera um número grande de configurações (microestados) de sistemas equilibrados. As configurações são obtidas aleatoriamente.

**Dinâmica Browniana:** é um eficiente método para simulação de grandes moléculas poliméricas ou partículas coloidais em solvente (moléculas pequenas). O solvente é tratado como um meio viscoso contínuo que dissipa energia quando o soluto se move através dele. A movimentação das partículas surge da colisão as mesmas com o solvente.

**Dinâmica Molecular (DM):** é o mais detalhado método de simulação que computa a movimentação das moléculas individuais. As equações de movimento de Newton descrevem as posições e momentos das partículas.

- As equações de movimento para estas partículas que interagem via potenciais intra e intermoleculares podem ser resolvidas usando várias métodos de integração número, por exemplo; Verlet, Leap-Frog, predição-correção, etc.
- A DM avalia eficientemente diferentes propriedades configuracionais e quantidades dinâmicas que não podem ser obtidas por Monte Carlo.

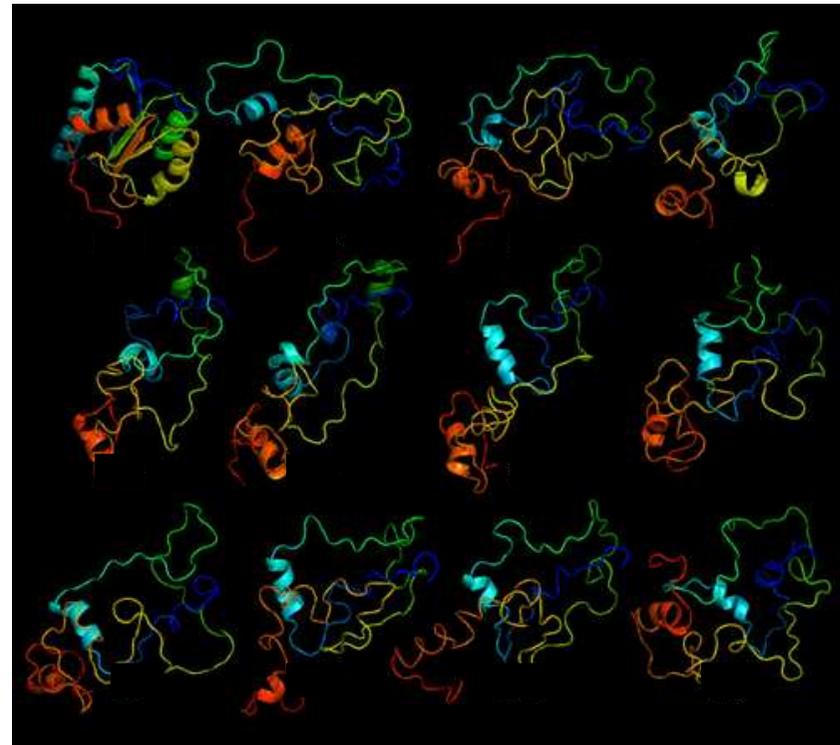
## Monte Carlo



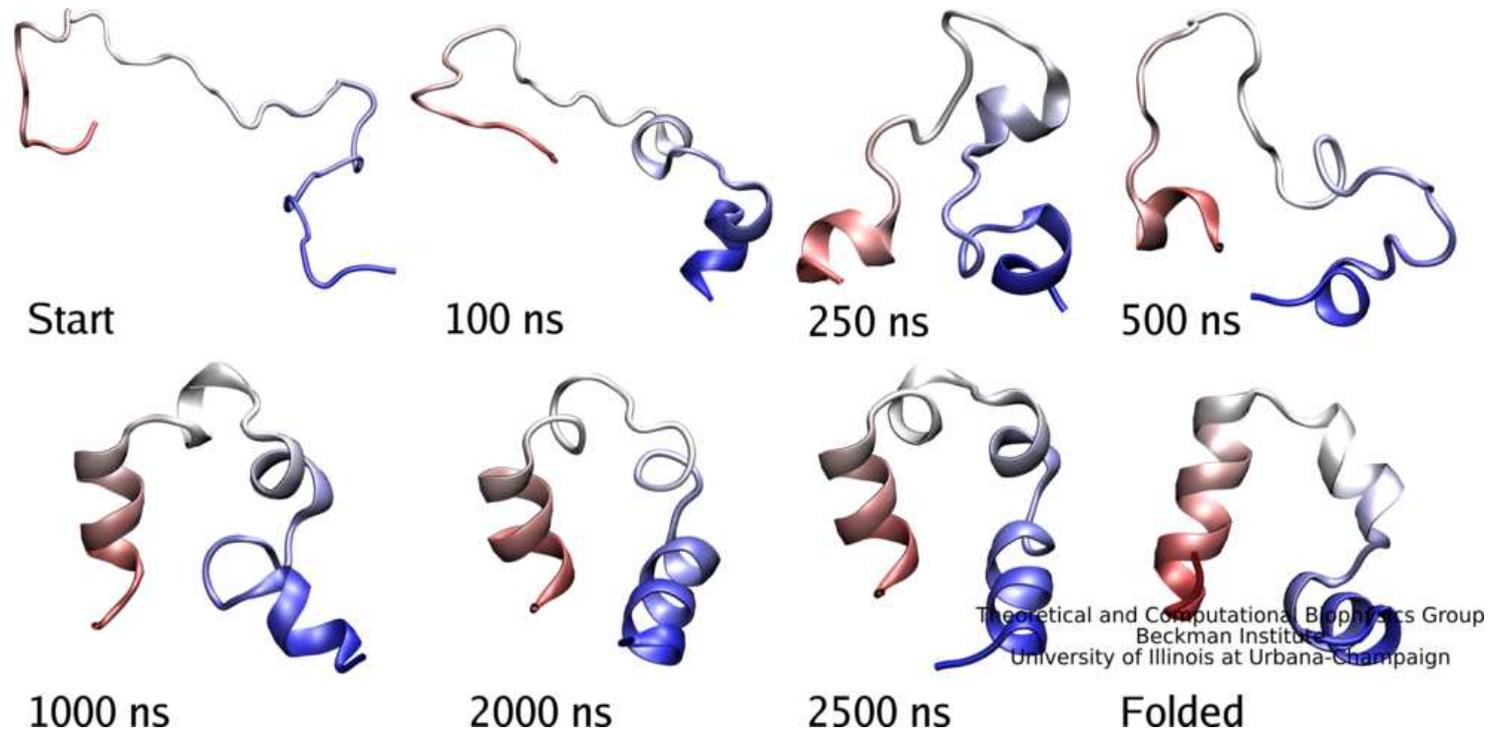
→ *Movimentação é aleatória. A energia do estado 2 deve ser favorável em relação ao estado 1.*

→ *Fator de Boltzmann:*  $e^{-\frac{\Delta V}{k_B T}}$

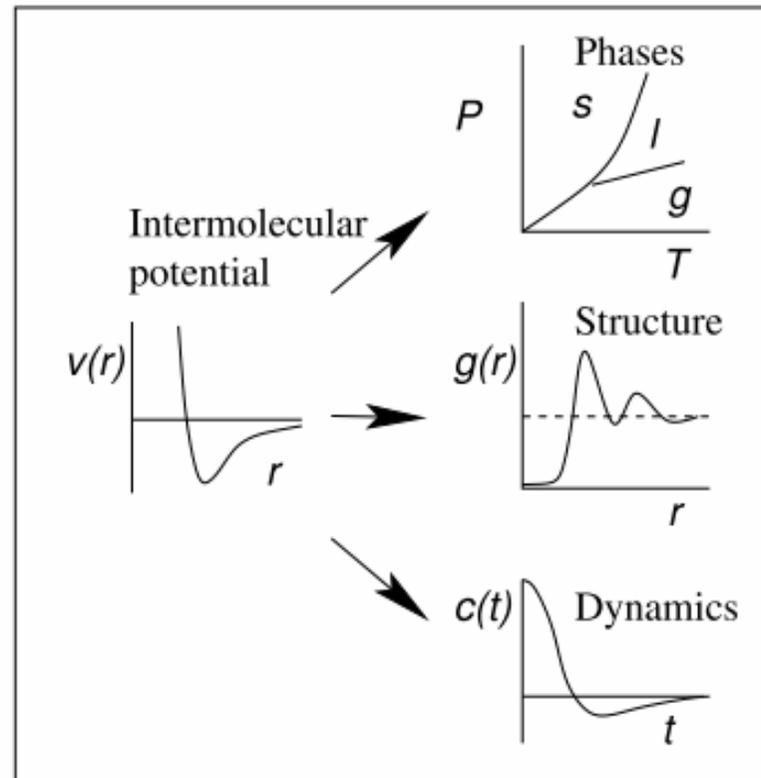
→ *Sem correlação temporal.*



## Dinâmica molecular (DM)



*Técnica computacional para determinar o movimento das partículas de qualquer sistema, do qual se conhece o **potencial de interação**,  $V(r)$ , entre as partículas e as **equações que regem seu movimento**.*



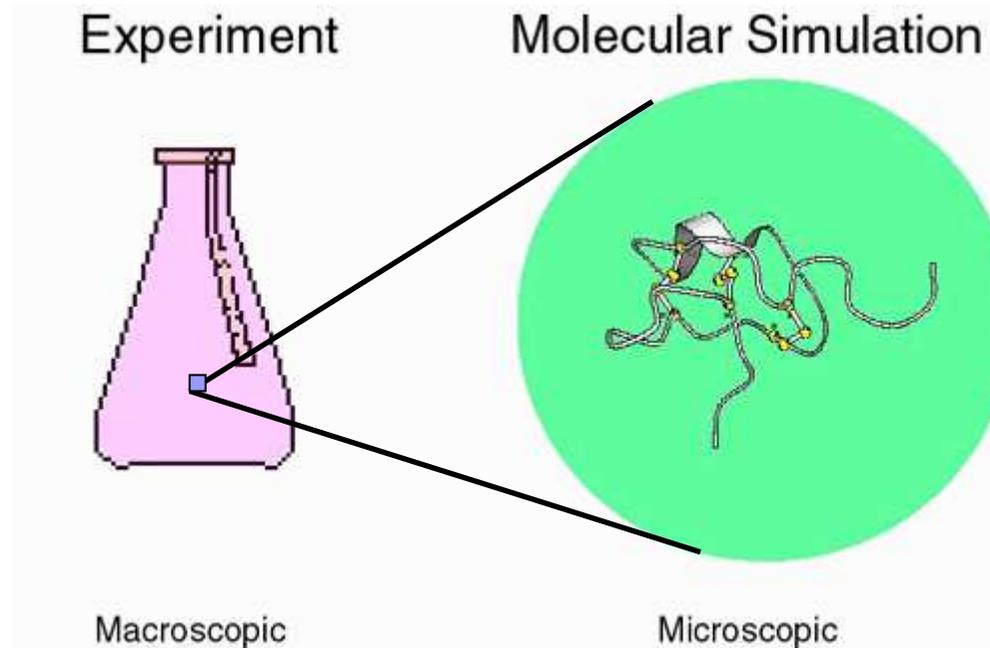
## O que é dinâmica molecular??

1. Um método computacional que descreve propriedades dinâmicas e de equilíbrio de um sistema molecular.
2. Gera configurações do sistema pela integração da Lei de Movimento de Newton – calcula as propriedades temporais do sistema.
3. Gera informações a nível microscópico – posições e velocidades atômicas.
4. Promove a conexão entre Estrutura e Função de moléculas fornecendo informações adicionais aos experimentos de cristalografia de Raios-X e RMN.
5. A mecânica estatística se faz necessária para conectar a movimentação atômico/molecular com as propriedades macroscópicas do sistema.

## Simulação Molecular – Mecânica Estatística

➔ *Simulação: Coleção de configurações (ensemble) do sistema que está sob condições específicas.*

*Por exemplo:  $N, V, T$ , ou  $N, p, T$ .*



## Médias observáveis – Mecânica Estatística

$$\Rightarrow \text{Experimental: } \langle A \rangle_{ensemble} = \int \int d\vec{p}^N d\vec{r}^N A(\vec{p}^N, \vec{r}^N) \rho(\vec{p}^N, \vec{r}^N)$$

$$\rho(\vec{p}^N, \vec{r}^N) = \frac{1}{Q} e^{[-H(\vec{p}^N, \vec{r}^N)/k_B T]} \quad \text{(Densidade de probabilidade)}$$

$$Q = \int \int d\vec{p}^N d\vec{r}^N e^{[-H(\vec{p}^N, \vec{r}^N)/k_B T]} \quad \text{(Função de partição)}$$

$$\Rightarrow \text{Simulação: } \langle A \rangle_{tempo} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{t=0}^{\tau} A[\vec{p}^N(t), \vec{r}^N(t)] dt \approx \frac{1}{M} \sum_{t=1}^M A(\vec{p}^N, \vec{r}^N)$$

$N$ : número de partículas do sistema

$p$  e  $r$ : momento linear e posições das partículas

$A(\vec{p}^N, \vec{r}^N)$ : propriedade observável qualquer – ex. entalpia

$$\langle A \rangle_{ensemble} = \langle A \rangle_{tempo}$$

## Breve histórico

A **1ª simulação computacional de líquidos** foi realizada em 1953 (Metrópolis et al., 1953). O modelo idealizado foi um disco bidimensional para representar as moléculas.

**Apenas 4 anos depois** já havia sido realizada a simulação considerando o modelo tridimensional - Fluido de Lennard-Jones (Wood and Parker, 1957), tornando possível comparar dados obtidos dos experimentos com dados gerados computacionalmente.

**No final dos anos 50** a DM foi introduzida aos estudos de interações entre esferas rígidas - átomos que interagem através de colisões perfeitas (Alder and Wainwright, 1957)

**1ª simulação de Ar líquido** foi realizada por Rahman em 1964 .

**1ª simulação de água líquida** foi realizada por Barker and Watts, 1969 .

**1ª simulação de proteínas** foi realizada por McCammon em 1977.

**Hoje:** Proteínas solvatadas, Complexos Proteína-DNA, Proteína-Carboidrato, Proteína-fármaco, Membranas, etc.

## Equações de movimento

*Equação de Schrödinger dependente do tempo:*

$$H_{nuc} \Psi_{nuc} = i\hbar \frac{\partial}{\partial t} \Psi_{nuc}$$

Aproximação: Quando o comprimento de onda térmico de *de Broglie* ( $\Lambda$ ) for muito menor que a distância média entre as partículas, podemos tratar o sistema classicamente.

$$\Lambda = \left( \frac{2\pi\hbar^2}{mk_B T} \right)^{\frac{1}{2}}$$

*Por exemplo:* Para um núcleo de carbono a 298K temos  $\Lambda=0.3\text{Å}$ , enquanto que a distância mínima entre os átomos numa simulação é de  $1\text{Å}$ . Então os núcleos podem ser considerados partículas pontuais e suas trajetórias podem ser propagadas classicamente.

Obs: Átomos leves ( H ) apresentam  $\Lambda=1\text{Å}$  à temperatura ambiente.

**Classicamente:** Em DM, as sucessivas configurações do sistema são geradas pela integração das equações de movimento de Newton. O resultado é uma trajetória que especifica como as posições e velocidades das partículas dos sistema variam com o tempo. As leis de movimento de Newton podem ser descritas como:

1. Um corpo continua se movendo em uma linha reta a velocidade constante a menos que uma força atue sobre ele.
2. Força é igual a taxa de variação do momento (massa x aceleração).
3. Para toda ação há igual e oposta reação

A trajetória é obtida resolvendo as equações diferenciais incorporadas na 2ª Lei de Newton:

$$\vec{F}_i = m_i \vec{a}_i$$

$$\vec{F}_i(t) = m_i \vec{a}_i(t) = m_i \frac{d^2 \vec{r}_i(t)}{dt^2} = m_i \ddot{\vec{r}}_i$$

$$\vec{F}_i(t) = -\nabla V(\vec{r}_1, \dots, \vec{r}_N) = -\frac{\partial V(\vec{r}_1, \dots, \vec{r}_N)}{\partial \vec{r}_i}$$

$$i = 1, \dots, N$$

$$\vec{F}_i(t) = m_i \vec{a}_i(t) = m_i \frac{d^2 \vec{r}_i(t)}{dt^2} = - \frac{\partial V(\vec{r}_1, \dots, \vec{r}_N)}{\partial \vec{r}_i}$$

Energia Potencial = Potencial de Interação

$$V(\vec{r}) = \sum_i v_1(\vec{r}_i) + \sum_i \sum_{j>i} v_2(\vec{r}_i, \vec{r}_j) + \sum_i \sum_{j>i} \sum_{k>j>i} v_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) + \dots$$

$$\vec{a}_i = \frac{d^2 \vec{r}}{dt^2} = \ddot{\vec{r}}_i(t) = \frac{\vec{F}_i(t)}{m_i} \quad i = 1, \dots, N$$

*Equação que descreve a movimentação da partícula de massa  $m$  a partir de uma força  $F$  atuando sobre ela.*

## DM – Potenciais de Interação

\* Os potenciais são aproximações clássicas dos potenciais quânticos.

Em princípio:

$$-\frac{\hbar^2}{2m} \nabla^2 \Psi(\vec{x}, t) + V(\vec{x}, t) \Psi(\vec{x}, t) = i\hbar \frac{\partial}{\partial t} \Psi(\vec{x}, t)$$

Quando o potencial não depende explicitamente do tempo:

$$\Psi(\vec{x}, t) = \varphi(\vec{x}) f(t)$$

$$-\frac{\hbar^2}{2m} \nabla^2 \varphi(\vec{x}) + V(\vec{x}) \varphi(\vec{x}) = E \varphi(\vec{x}) \longrightarrow H\Phi = E\Phi$$

$$H = -\sum_{i=1}^N \frac{1}{2} \nabla^2_i - \sum_{A=1}^M \frac{1}{2M} \nabla^2_A - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^N \sum_{j>1}^N \frac{1}{r_{ij}} + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{R_{AB}}$$

\* Aproximação de Born-Oppenheimer.

$$H_{\text{núcleo}} = -\sum_{A=1}^M \frac{1}{2M_A} \nabla_A^2 + \mathcal{E}_{\text{total}}$$

$\mathcal{E}_{\text{total}}$  : superfície de energia potencial gerada pela presença dos elétrons para cada posição dos núcleos mais a energia de repulsão entre os núcleos.

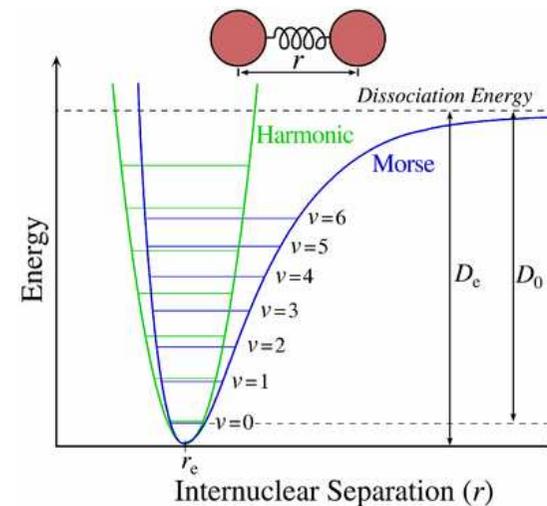
➡ Potencial para o movimento nuclear

➡ Obtida por cálculo quântico *ab-initio* ou semi-empírico.

**MUITO CARO COMPUTACIONALMENTE!**

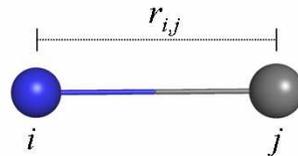
\* Aproximação de  $\mathcal{E}_{\text{total}}$  para potenciais clássico analíticos.

$$V(r) = D_e [1 - e^{-a(r-r_e)}]^2$$



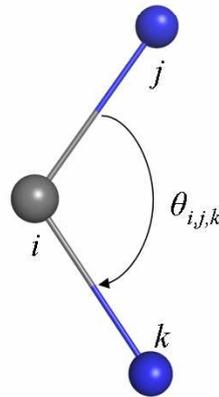
Estiramento da ligação

$$V_{lig}(r) = \frac{1}{2} K_r (r - r_0)^2$$

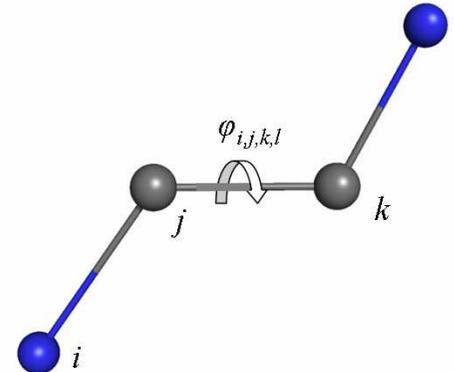


Vibração angular

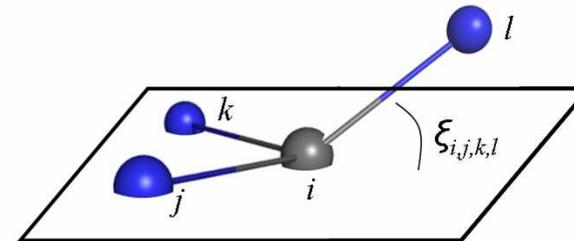
$$V_{ang}(\theta) = \frac{1}{2} K_\theta (\theta - \theta_0)^2$$



Vibração diedral própria

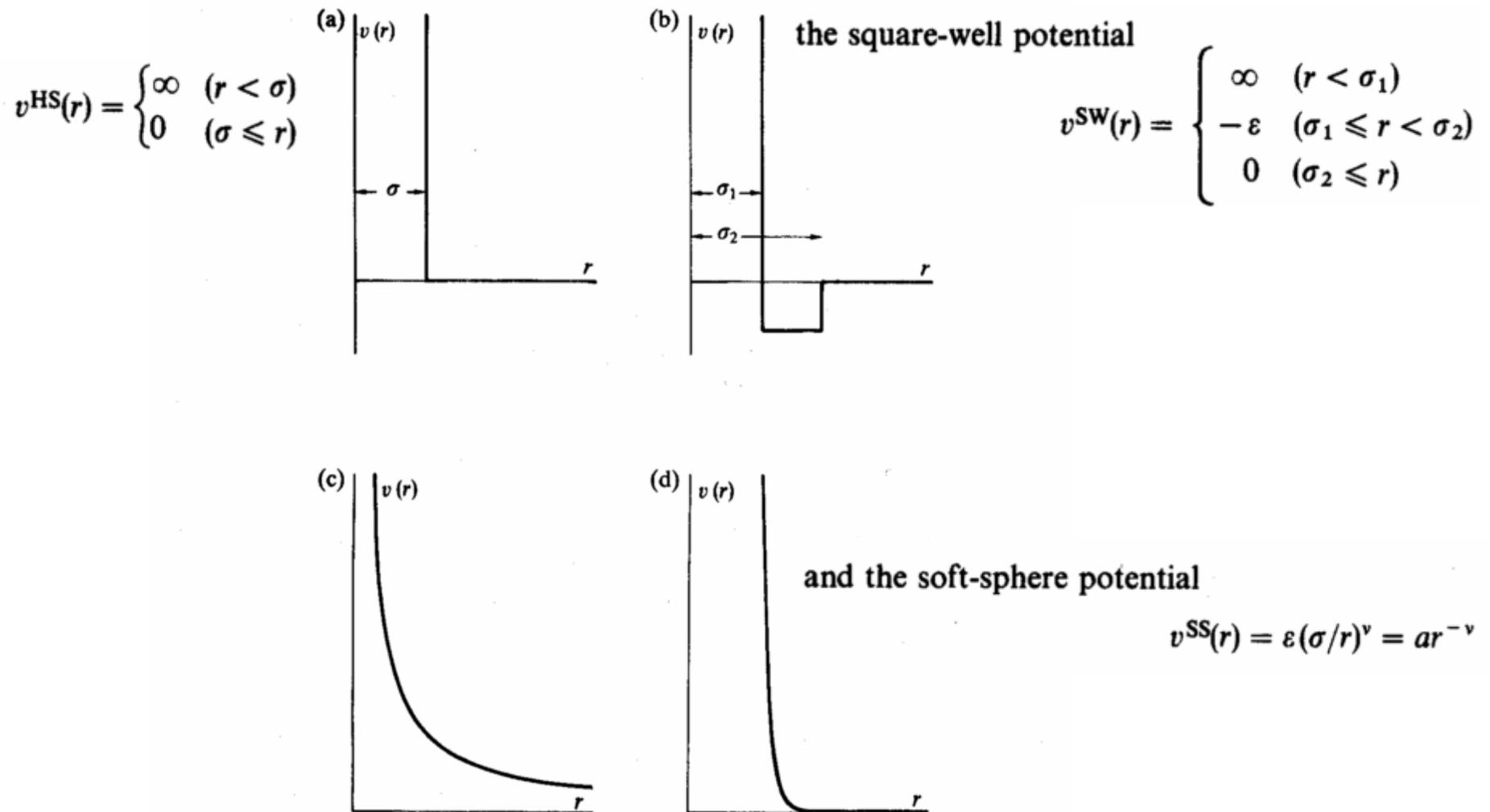


Vibração diedral imprópria



$$V_{died}(\varphi) = V_0 + \frac{V_1}{2} [1 + \cos(\varphi)] + \frac{V_2}{2} [1 + \cos(2\varphi)] + \frac{V_3}{2} [1 + \cos(3\varphi)]$$

## Potenciais Simples



**Fig. 1.4** Idealized pair potentials. (a) The hard-sphere potential; (b) The square-well potential; (c) The soft-sphere potential with repulsion parameter  $\nu = 1$ ; (d) The soft-sphere potential with repulsion parameter  $\nu = 12$ .

## Potenciais Simples: Lennard-Jones (Potencial 6-12)

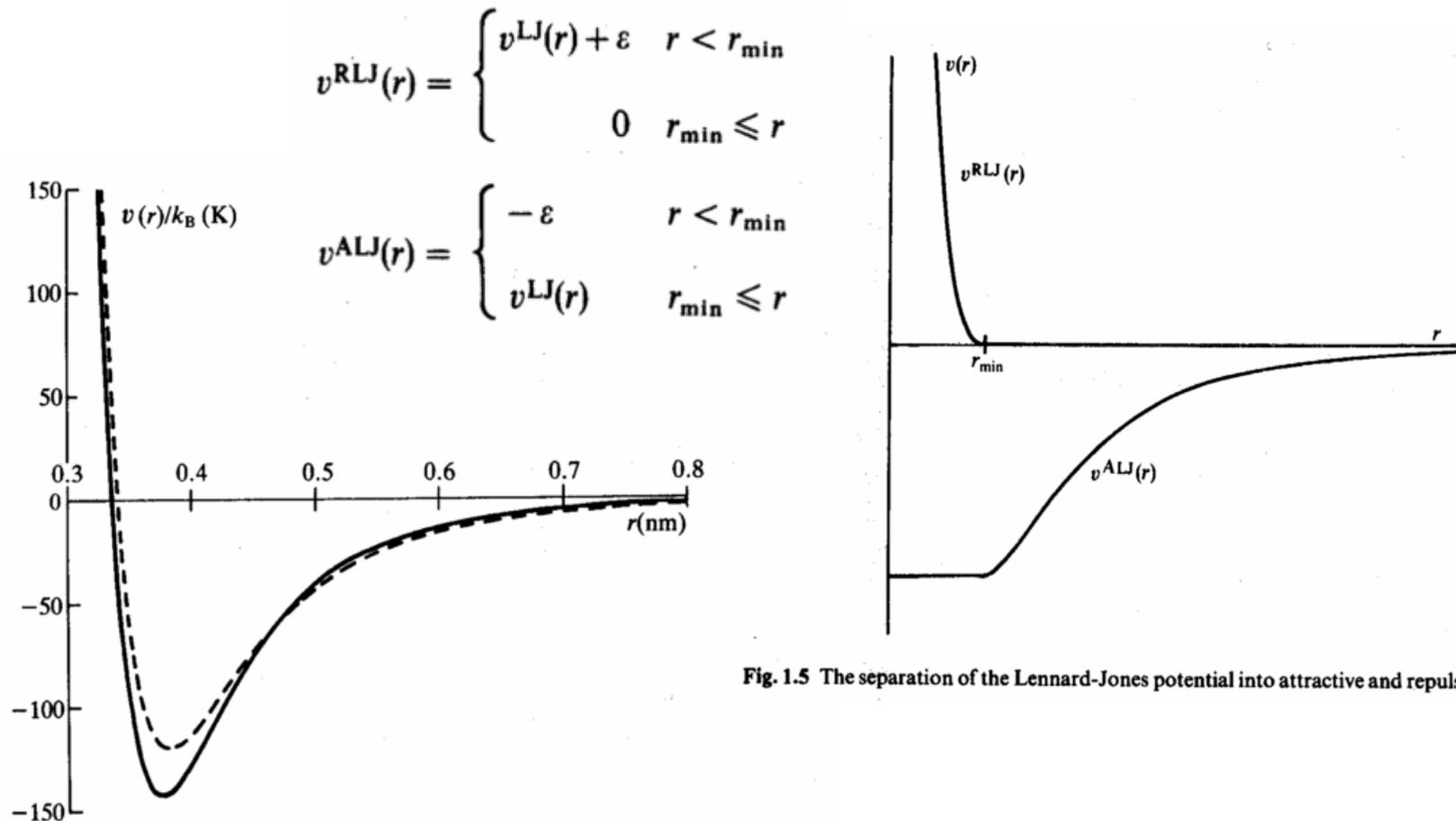
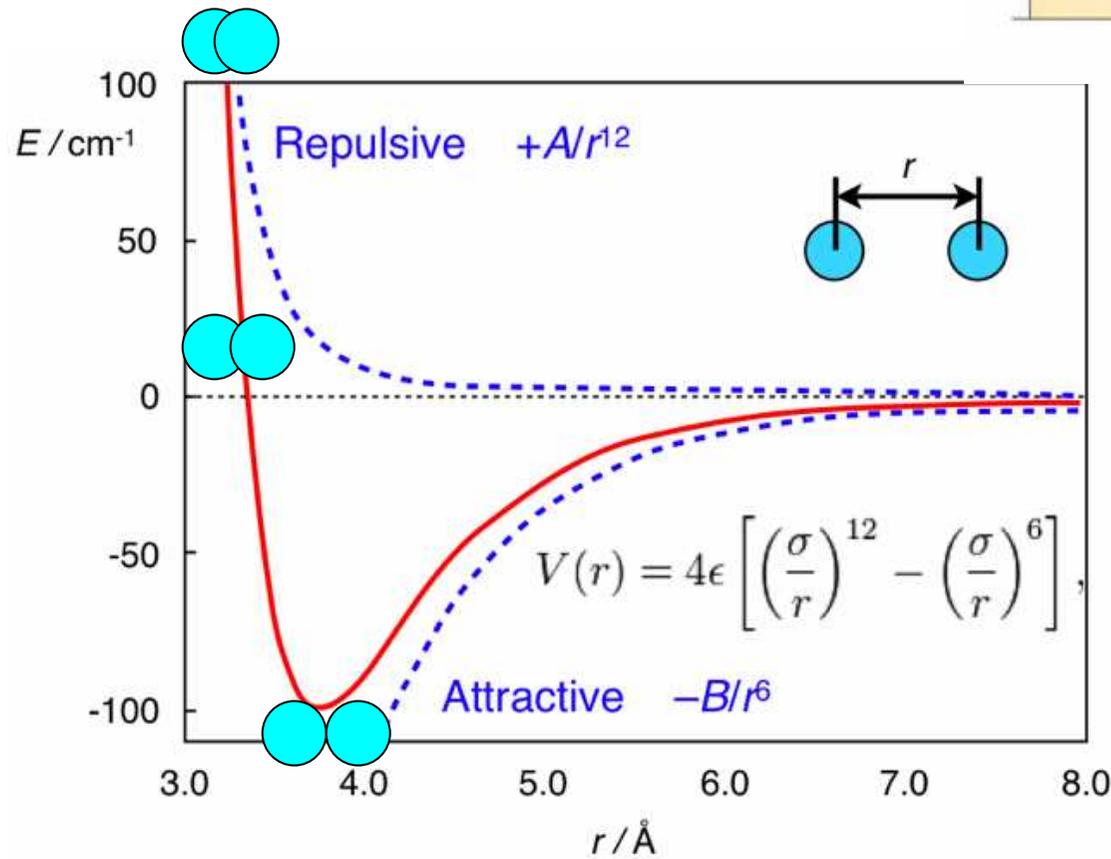
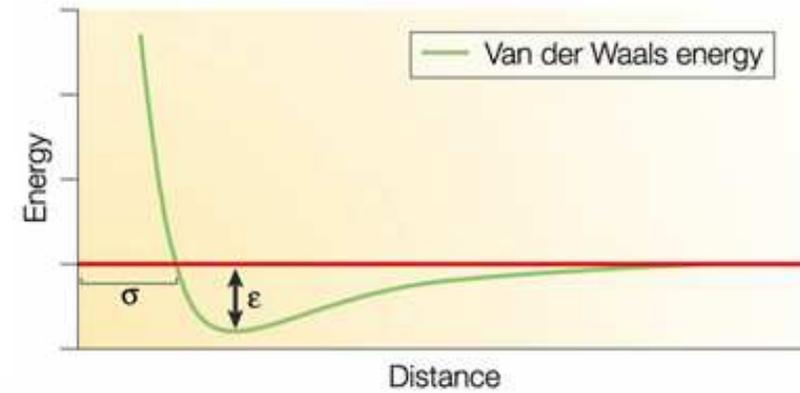


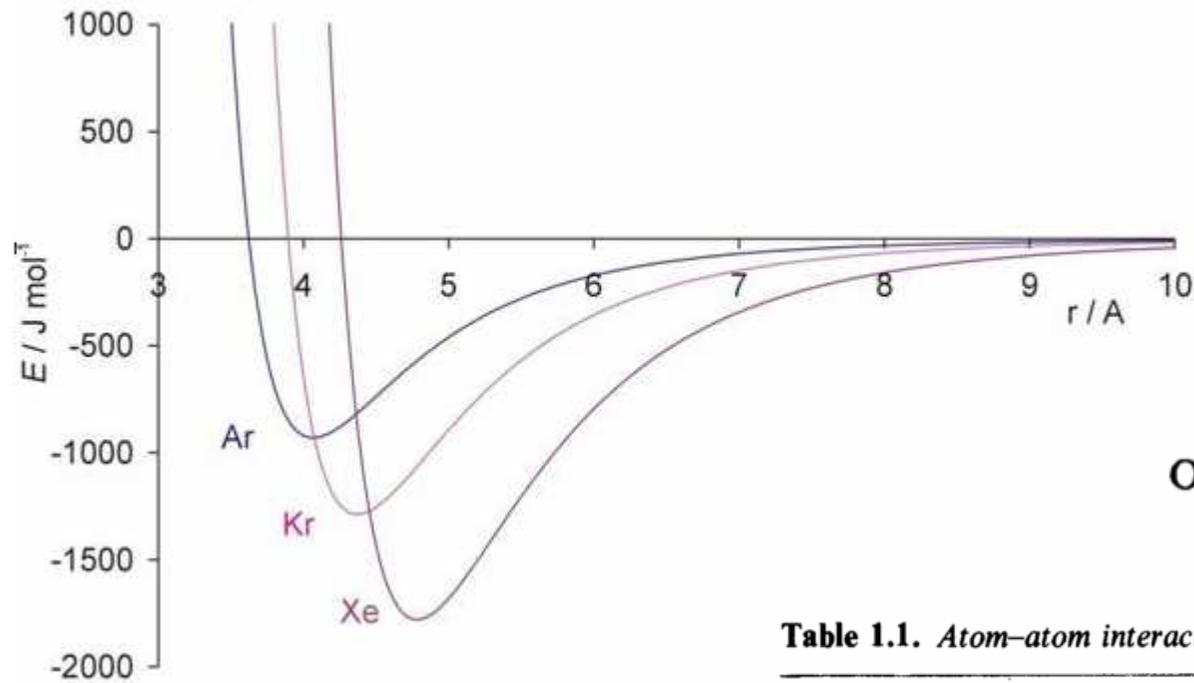
Fig. 1.5 The separation of the Lennard-Jones potential into attractive and repulsive components.

Fig. 1.3 Argon pair potentials. We illustrate the BBMS pair potential for argon (solid line) [Maitland and Smith 1971]. The BFW potential [Barker *et al.* 1971] is numerically very similar. Also shown is the Lennard-Jones 12-6 effective pair potential (dashed line) used in computer simulations of liquid argon.

## Potenciais Simples: Lennard-Jones (Potencial 6-12)

$$V(r) = \frac{A}{r^{12}} - \frac{B}{r^6}$$





$$\sigma_{CS} = \frac{1}{2}[\sigma_{CC} + \sigma_{SS}]$$

$$\epsilon_{CS} = [\epsilon_{CC} \epsilon_{SS}]^{1/2}$$

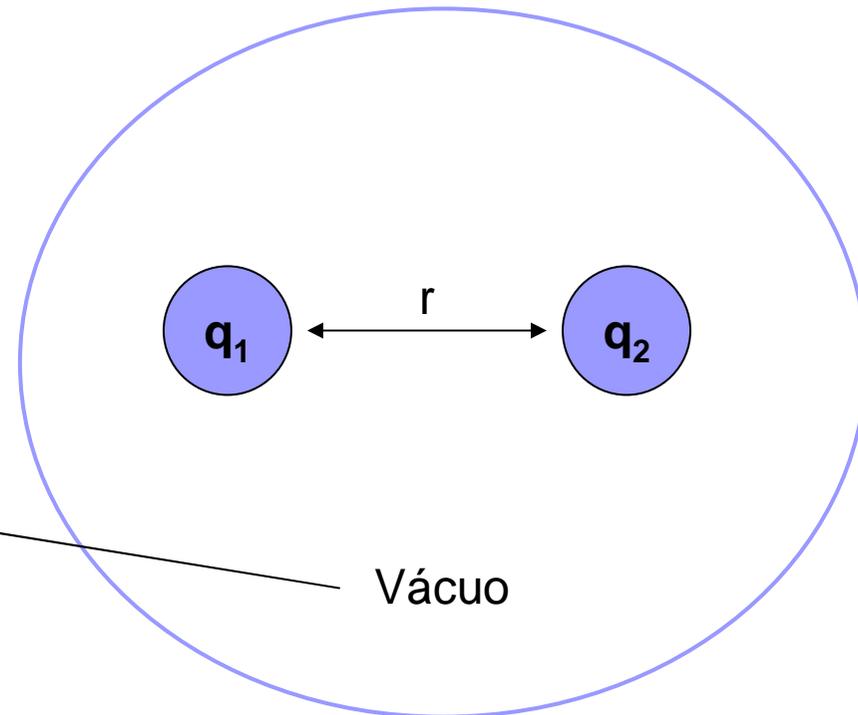
**Table 1.1.** Atom–atom interaction parameters

Atom	Source	$\epsilon/k_B(\text{K})$	$\sigma(\text{nm})$
H	[Murad and Gubbins 1978]	8.6	0.281
He	[Maitland <i>et al.</i> 1981]	10.2	0.228
C	[Tildesley and Madden 1981]	51.2	0.335
N	[Cheung and Powles 1975]	37.3	0.331
O	[English and Venables 1974]	61.6	0.295
F	[Singer <i>et al.</i> 1977]	52.8	0.283
Ne	[Maitland <i>et al.</i> 1981]	47.0	0.272
S	[Tildesley and Madden, 1981]	183.0	0.352
Cl	[Singer <i>et al.</i> 1977]	173.5	0.335
Ar	[Maitland <i>et al.</i> 1981]	119.8	0.341
Br	[Singer <i>et al.</i> 1977]	257.2	0.354
Kr	[Maitland <i>et al.</i> 1981]	164.0	0.383

## Potencial de Coulomb

$$V(r)^{coul} = \frac{q_1 q_2}{4\pi\epsilon_0 r}$$

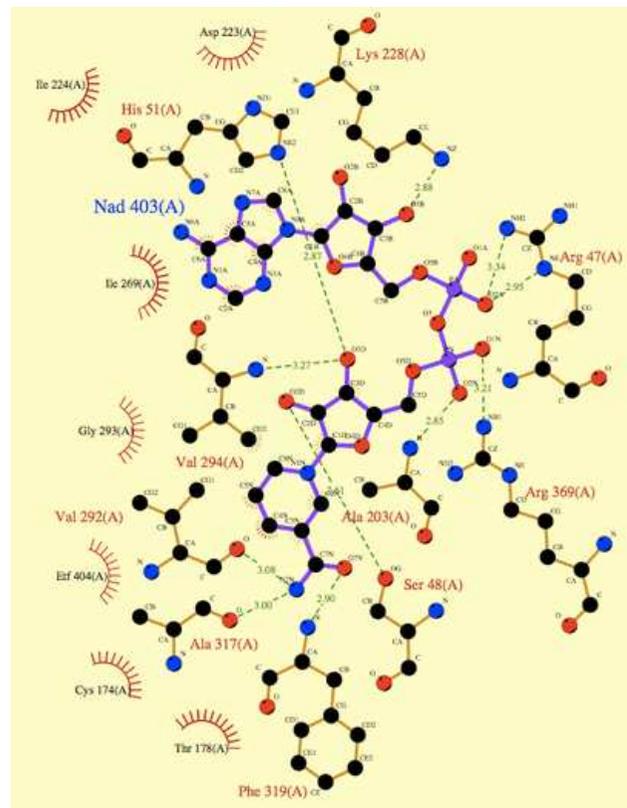
$$\epsilon_0 = 8,854 \times 10^{-12} \text{ J}^{-1} \cdot \text{C}^2 \cdot \text{m}^{-1}$$



$$V(r) = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right] + \frac{q_1 q_2}{4\pi\epsilon_0 r}$$

$$V_{Total} = \sum V_{Intra} + \sum V_{inter}$$

$$V(\vec{r}_1, \dots, \vec{r}_N) \equiv \sum V_{ligação} + \sum V_{angular} + \sum V_{diedral}^{próprio} + \sum V_{diedral}^{impróprio} + \sum V_{vdW} + \sum V_{Coul}$$



## Campo de Força

- Ao conjunto de parâmetros necessários para descrever todas as interações dá-se o nome de campo de força.

$$V_{Total} = \sum V_{Inter} + \sum V_{int ra}$$

- **OPLS**
- **AMBER**
- **GLYCAM**
- **GROMOS**
- **CHARMM**
- **COMPASS**
- **GROMACS**

Campos de forças bem estabelecidos e usados para biomoléculas em geral (sacarídeos, peptídeos, proteínas, lipídeos e ácidos nucléicos), diferentes tipos de solventes, íons, moléculas orgânicas.

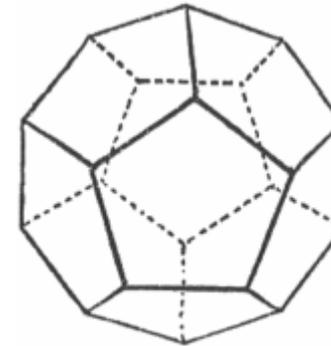
$$H_{núcleo} = -\sum_{A=1}^M \frac{1}{2M_A} \nabla_A^2 + \mathcal{E}_{total}$$

## Etapas de uma simulação de DM

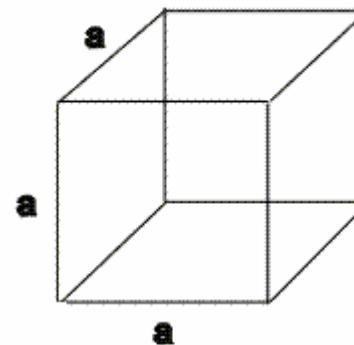
1. *Gerar configuração inicial das moléculas.*
2. *Cálculo das forças exercidas sobre cada partícula devido às interações intermoleculares.*
3. *Movimentação das partículas.*
4. *Condições (Ensembles) e controle da simulação*
5. *Armazenamento das configurações obtidas (trajetória)*
6. *Análise da trajetória atômica.*

## Configuração Inicial do sistema

1. Gerar a caixa de simulação onde se encerram as  $N$  moléculas do sistema.
2. As dimensões são escolhidas de modo que a densidade numérica,  $N/V$ , corresponda àquela do sistema real nas condições termodinâmicas desejadas.



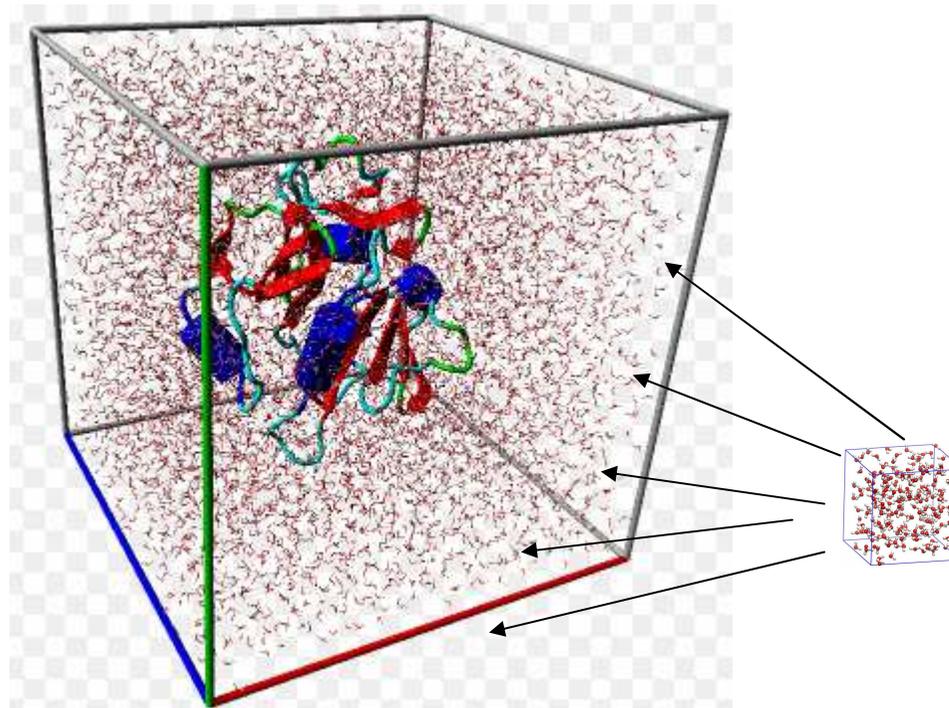
Dodecahedron.

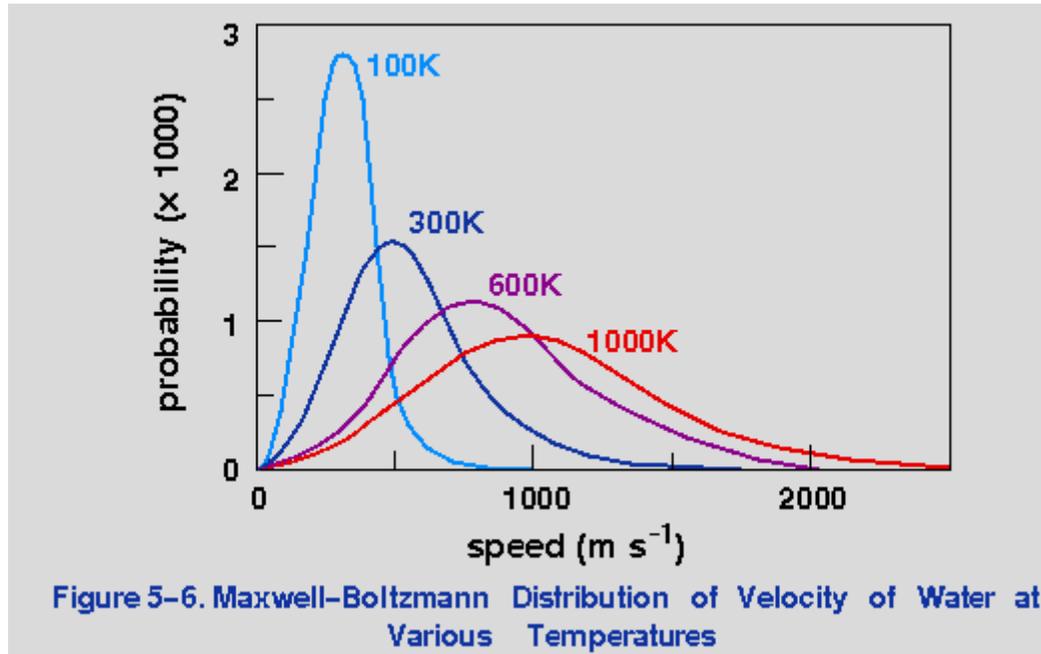


## COORDENADAS INICIAIS (X, Y, Z)

1. **Soluto:** molécula orgânica, proteína, DNA, complexo enzima/receptor. Geralmente estruturas de macromoléculas são obtidas dos banco de dados apropriados, por exemplo, o banco de dados de proteínas (PDB).
2. **Solvente e íons:** São adicionados posteriormente e posicionadas em uma rede cúbica de face centrada.

- Banco de dados de proteínas: PDB ([http:// ww.rcsb.org/pdb](http://ww.rcsb.org/pdb)).
- Banco de dados de carboidratos: CARBBANK (<http://www.boc.chem.uu.nl/sugabase/carbbank.html>).



VELOCIDADES INICIAIS: Distribuição de Maxwell-Boltzmann na temperatura desejada.

$$\frac{3}{2} N k_B T = \frac{1}{2} \sum_{i=1}^N m_i v_i^2$$

**OU:** Velocidade inicial nula para todos os átomos e deixar o sistema evoluir a partir do repouso. De qualquer forma, é importante que a soma das velocidades de todas as partículas seja um vetor nulo, evitando assim o deslocamento da caixa.

## Ensembles

*Durante a simulação, alguns parâmetros macroscópicos podem ser mantidos constantes em conjunto:*

*$(N, V, T)$  – Canônico*

*$(N, p, T)$  – Isotérmico-Isobárico*

*$(N, V, E)$  – Microcanônico*

*$(\mu, V, T)$  – grand canônico*

Esses conjuntos de parâmetros caracterizam ensembles diferentes e definem uma equação de estado para o sistema, permitindo que diferentes funções termodinâmicas sejam calculadas em diferentes ensembles.

**NVT:** O número de partículas é fixo. O volume é fixo mantendo fixas as dimensões da caixa e a temperatura pode ser fixada de várias maneiras.

## NVT: Termostato – Fixar a Temperatura!

- \* Berendsen
- \* V-Rescale
- \* Nosé-Hoover

Todos correspondem à banhos térmicos capazes de fornecer ou retirar energia térmica (energia cinética) do sistema, ou seja, a energia total não é conservada nesse ensemble.

### Calculando a temperatura!!

$$\frac{3}{2} N k_B T = \frac{1}{2} \sum_{i=1}^N m_i v_i^2$$

### Escalonando as velocidades!!

$$\lambda = \sqrt{\frac{T_0}{T(t)}}$$

Multiplicar as velocidades atômicas pelo coeficiente  $\lambda$  em cada passo ou intervalo de modo que a temperatura a cada passo da simulação  $T(t)$  convirja para  $T_0$

### Termostato de Berendsen

$$\lambda = \sqrt{1 + \frac{n_{TC} dt}{\tau_T} \left( \frac{T_0}{T\left(t - \frac{1}{2} dt\right)} - 1 \right)}$$

$\tau_T$  Constante temporal de acoplamento de temperatura aplicada a cada passo ou  $n_{TC}$  passos. Geralmente 0.1ps para simulações em fase condensada.

$\tau_T \rightarrow \infty$  Acoplamento inativo: ensemble NVE.

## Movimentação das partículas

Estabelecidas as condições iniciais (posições e velocidades atômicas) prosseguimos com a movimentação das partículas, ou seja, determinar as posições e velocidades futuras.

$$\vec{F}_i(t) = -\nabla V(\vec{r}_1, \dots, \vec{r}_N) = -\frac{\partial V(\vec{r}_1, \dots, \vec{r}_N)}{\partial \vec{r}_i}$$

$$\vec{a}_i = \frac{d^2 \vec{r}}{dt^2} = \ddot{\vec{r}}_i(t) = \frac{\vec{F}_i(t)}{m_i} \quad i = 1, \dots, N$$

**Métodos de diferenças finitas: Algoritmo de Verlet e suas variantes.**

## Como integrar as equações de movimento?

A energia potencial é uma função das posições de todas as partículas do sistema.

Devido a esta complexidade, não há solução analítica para o problema.

Usamos algoritmos que aproximam as posições, velocidades e acelerações das partículas -

**Expansão de Taylor:** Expansão de uma série de funções ao redor de um ponto. A expansão de Taylor em 1 dimensão de uma função real  $f(x)$  ao redor do ponto em que  $x=a$ ;

$$f(x) = f(a)(x-a)^0 + \frac{1}{1!} \frac{df(a)}{dt} (x-a)^1 + \frac{1}{2!} \frac{d^2 f(a)}{dt^2} (x-a)^2 + \frac{1}{3!} \frac{d^3 f(a)}{dt^3} (x-a)^3 + \dots + \frac{1}{n!} \frac{d^n f(a)}{dt^n} (x-a)^n$$

$$x = t + \Delta t$$

Posição!

$$a = t$$

$$x(t + \Delta t) = x(t) + \frac{dx}{dt} \Delta t + \frac{d^2 x(t)}{dt^2} \frac{\Delta t^2}{2} + \frac{d^3 x(t)}{dt^3} \frac{\Delta t^3}{6} + \dots$$

## Método das diferenças finitas

Método empregado para gerar trajetórias atômicas com modelo de potencial contínuo. A ideia essencial é que a integração é decomposta em muitos estágios separados no tempo por um intervalo, ou tempo de integração,  $dt$ .

A força total em cada partícula na configuração no tempo  $t$  é calculada como um vetor soma de suas interações com as outras partículas. Da força calcula-se as acelerações das partículas, que são então combinadas com as posições e velocidades no tempo  $t$  para obtermos as posições e velocidades no tempo  $t+dt$ ,  $t+2dt$ ,  $t+3dt$ , e assim por diante.

$$\vec{r}_i(t + dt) = \vec{r}_i(t) + \vec{v}_i(t)dt + \frac{1}{2}\vec{a}_i(t)dt^2 + \frac{1}{6}\vec{b}_i(t)dt^3 + \dots$$

$$\vec{v}_i(t + dt) = \vec{v}_i(t) + \vec{a}_i(t)dt + \frac{1}{2}\vec{b}_i(t)dt^2 + \dots$$

$$\vec{a}_i(t + dt) = \vec{a}_i(t) + \vec{b}_i(t)dt + \dots$$

## Algoritmo de Verlet (Verlet, 1967)

É o método de integração das equações de movimento mais amplamente empregado. O algoritmo de Verlet usa as posições e acelerações no tempo  $t$  e as posições e acelerações no tempo  $t-dt$  para calcular as novas posições em  $t+dt$ .

$$\vec{r}_i(t + dt) = \vec{r}_i(t) + \vec{v}_i(t)dt + \frac{1}{2}\vec{a}_i(t)dt^2 + \frac{1}{6}\vec{b}_i(t)dt^3 + \dots$$

+

$$\vec{r}_i(t - dt) = \vec{r}_i(t) - \vec{v}_i(t)dt + \frac{1}{2}\vec{a}_i(t)dt^2 - \frac{1}{6}\vec{b}_i(t)dt^3 + \dots$$

---


$$\vec{r}_i(t + dt) = 2\vec{r}_i(t) - \vec{r}_i(t - dt) + \vec{a}_i(t)dt^2$$

$$\vec{r}_i(t + dt) = 2\vec{r}_i(t) - \vec{r}_i(t - dt) + \frac{\vec{F}_i(t)}{m_i}dt^2$$

Uma vez que as velocidades não aparecem na equação de determina a trajetória, temos que calculá-las, e uma das maneiras é;

$$\vec{r}_i(t + dt) = \vec{r}_i(t) + \vec{v}_i(t)dt + \frac{1}{2}\vec{a}_i(t)dt^2 + \dots$$

-

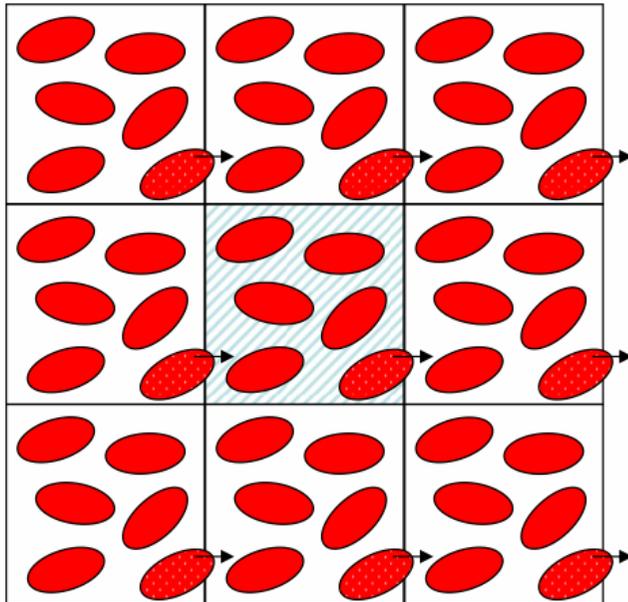
$$\vec{r}_i(t - dt) = \vec{r}_i(t) - \vec{v}_i(t)dt + \frac{1}{2}\vec{a}_i(t)dt^2 + \dots$$

---

$$\vec{v}_i(t) = \frac{\vec{r}_i(t + dt) - \vec{r}_i(t - dt)}{2dt}$$

## SISTEMAS PEQUENOS!! – EFEITOS DE SUPERFÍCIE SÃO MUITO FORTES

1. **Condições periódicas de contorno:** minimizar efeitos de fronteira. Réplicas idênticas da caixa de simulação são consideradas como estando dispostas ao redor da caixa principal formando um grande sistema que tende ao limite termodinâmico ( $N$  e  $V$  tendendo ao infinito porém  $N/V$  constante).
2. A movimentação das partículas nas réplicas são idênticas à movimentação na caixa principal, ou seja, as partículas não são limitadas pelas paredes das caixas.
3. As dimensões são escolhidas de modo que a densidade numérica,  $N/V$ , corresponda àquela do sistema real nas condições termodinâmicas desejadas.



*Exemplo.: 1000 átomos arranjados em uma caixa cúbica 10x10x10. Quase metade dos átomos estão na superfície!!!!*

**Convenção da imagem mínima:** Como as interações entre as imagens das partículas pertencentes a diferentes réplicas periódicas devem ser manipuladas. (Raio de corte)

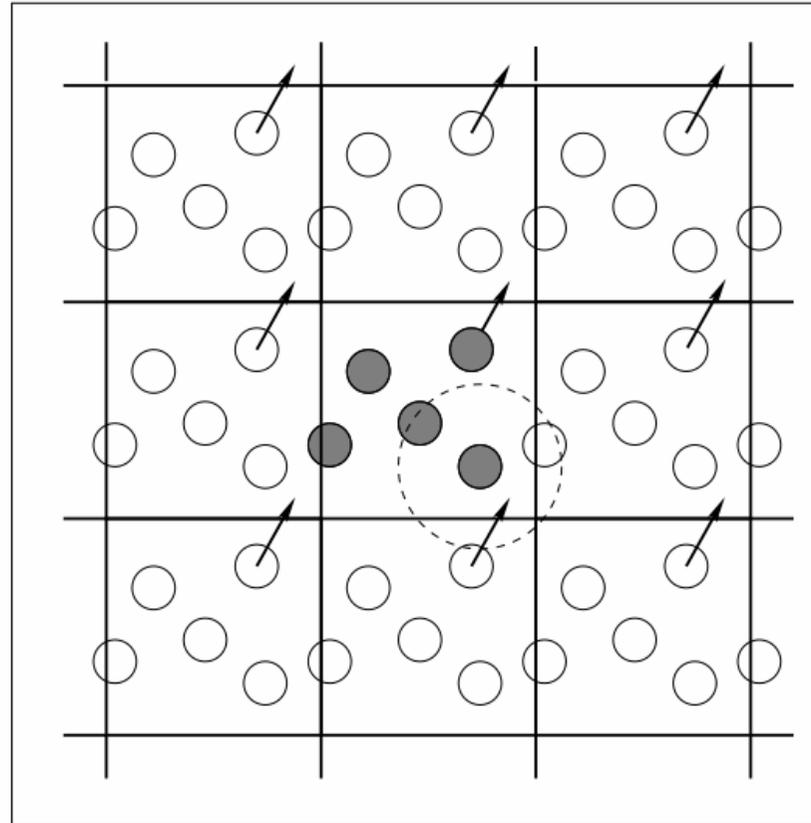


Figure 5. Periodic boundary conditions. As a particle moves out of the simulation box, an image particle moves in to replace it. In calculating particle interactions within the cutoff range, both real and image neighbours are included.

## THE GLOBAL MD ALGORITHM

### 1. Input initial conditions

Potential interaction  $V$  as a function of atom positions  
 Positions  $\mathbf{r}$  of all atoms in the system  
 Velocities  $\mathbf{v}$  of all atoms in the system

↓

repeat 2,3,4 for the required number of steps:

### 2. Compute forces

The force on any atom

$$\mathbf{F}_i = -\frac{\partial V}{\partial \mathbf{r}_i}$$

is computed by calculating the force between non-bonded atom pairs:

$$\mathbf{F}_i = \sum_j \mathbf{F}_{ij}$$

plus the forces due to bonded interactions (which may depend on 1, 2, 3, or 4 atoms), plus restraining and/or external forces.

The potential and kinetic energies and the pressure tensor are computed.

↓

### 3. Update configuration

The movement of the atoms is simulated by numerically solving Newton's equations of motion

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i}$$

or

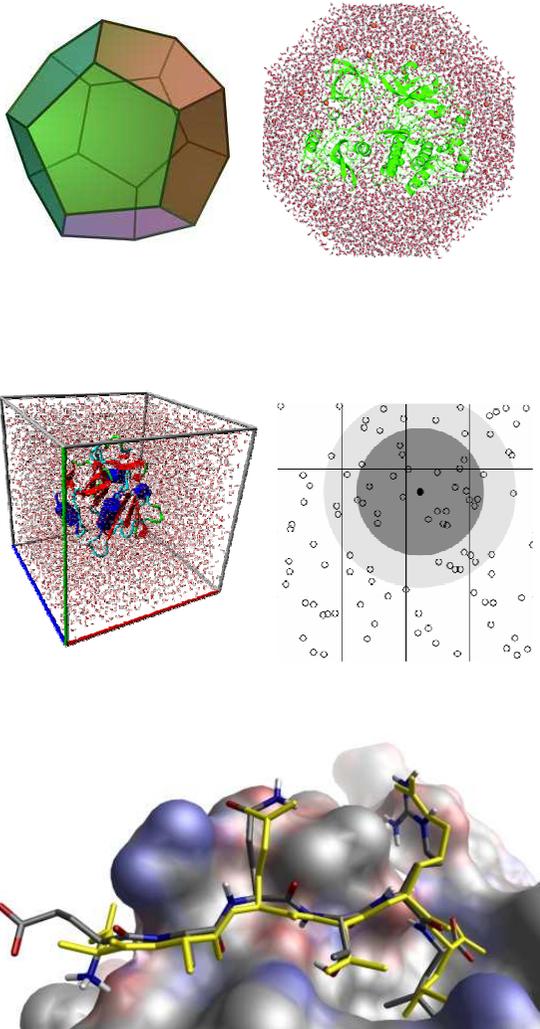
$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i; \quad \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{F}_i}{m_i}$$

↓

### 4. if required: Output step

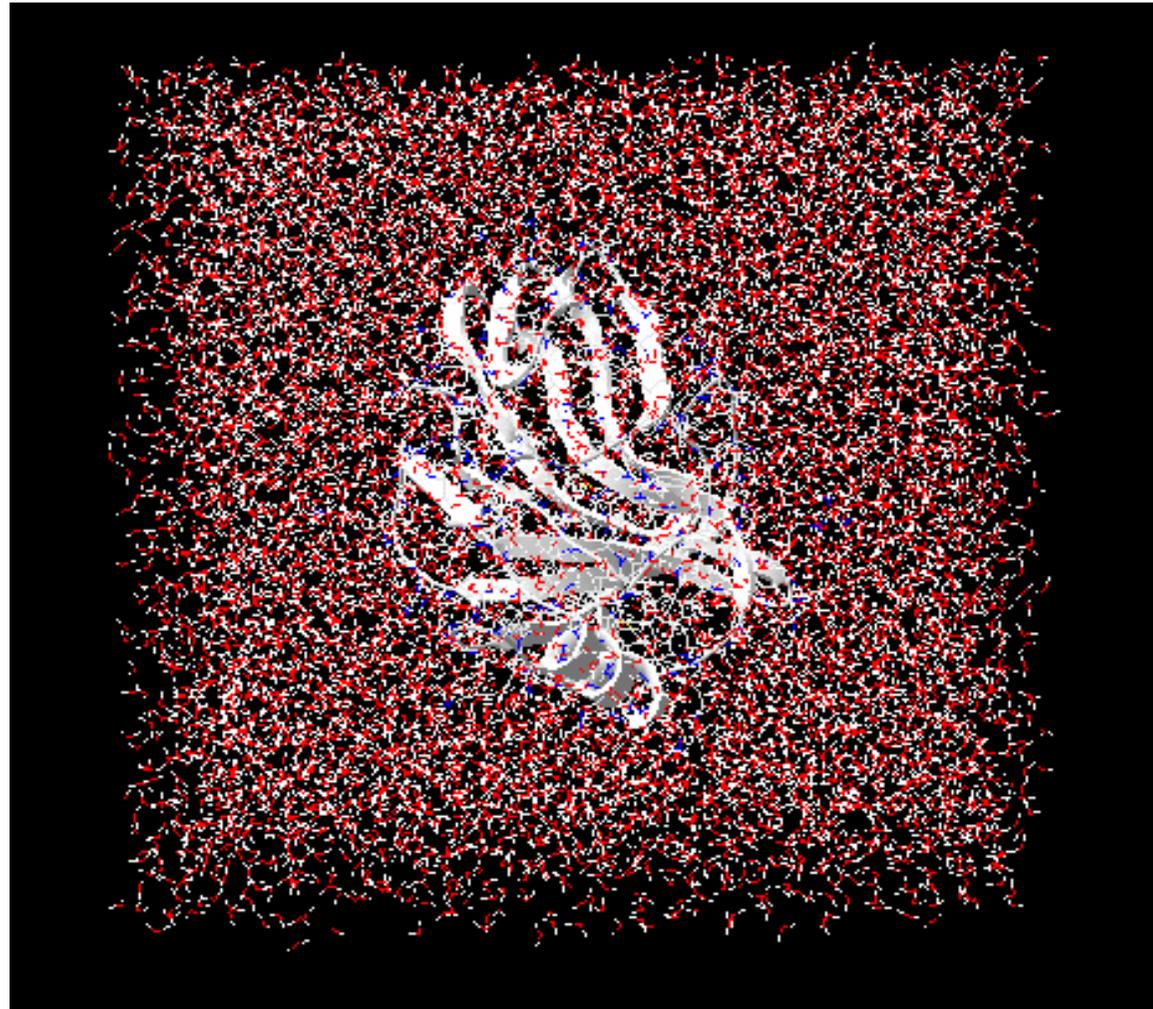
write positions, velocities, energies, temperature, pressure, etc.

## Protocolo básico de simulação

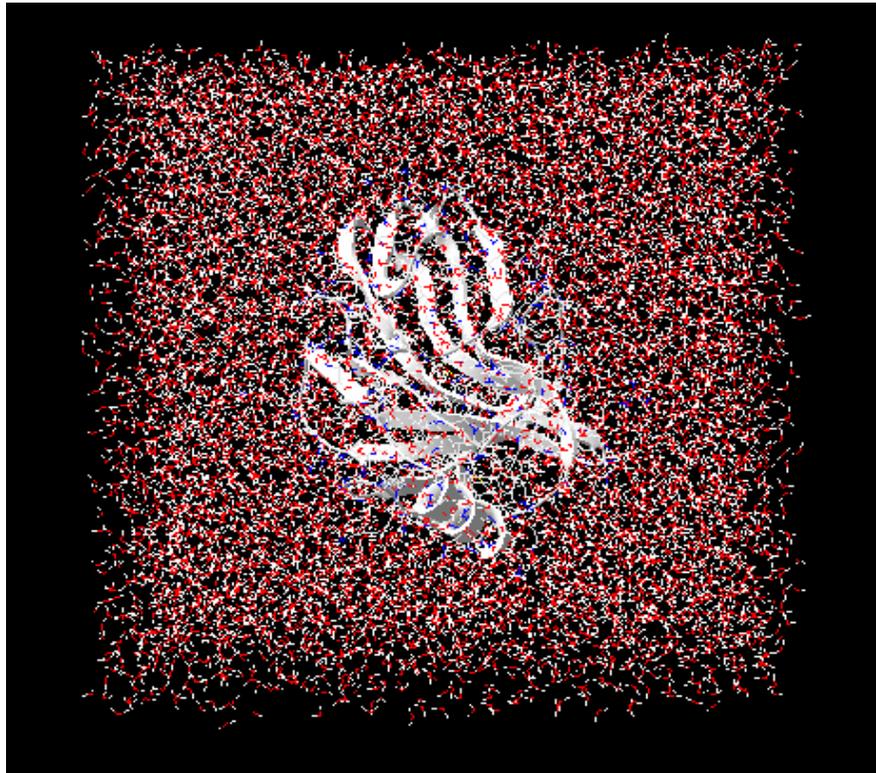


- Estrutura inicial ( Protein Data Bank/PDB\*).
- Campo de força: ex GROMOS96 (43a1).
- ensemble NpT, NVT, etc.
- tempo de integração e algoritmo ( dt=2.0fs e leap-frog )
- pH e temperatura.
- moléculas de água SPC, SPC/E, TIP4P, TIP5P.
- adição de íons – eletroneutralidade do sistema.
- raio de corte = 15Å.
- interações de longo alcance : Soma de Ewald (PME).
- sistema cúbico, dodecaédrico, octaédrico, triclínico, etc.

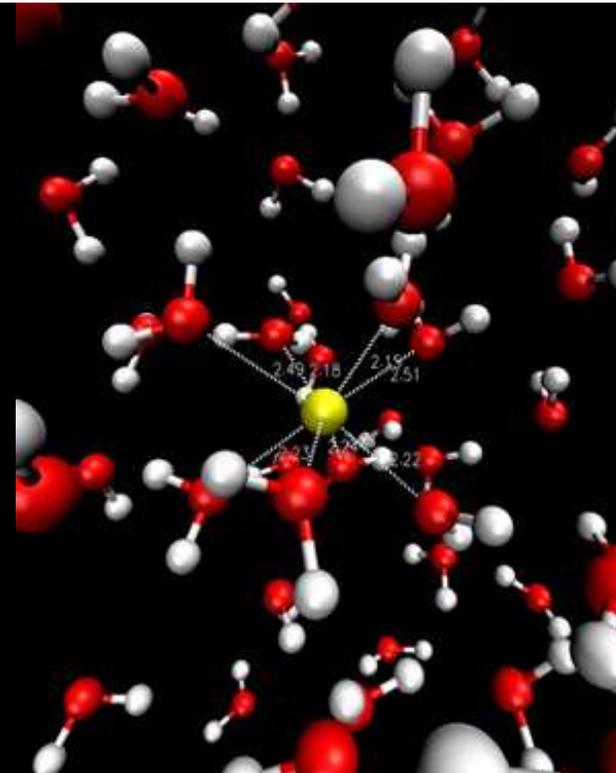
\* [www.rcsb.org/pdb](http://www.rcsb.org/pdb)



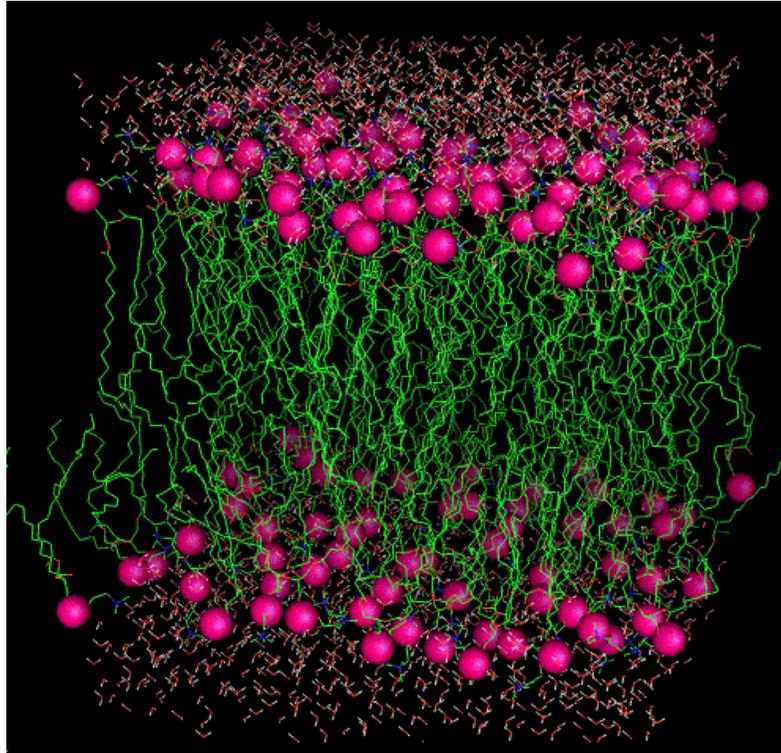
Proteína+água+íons



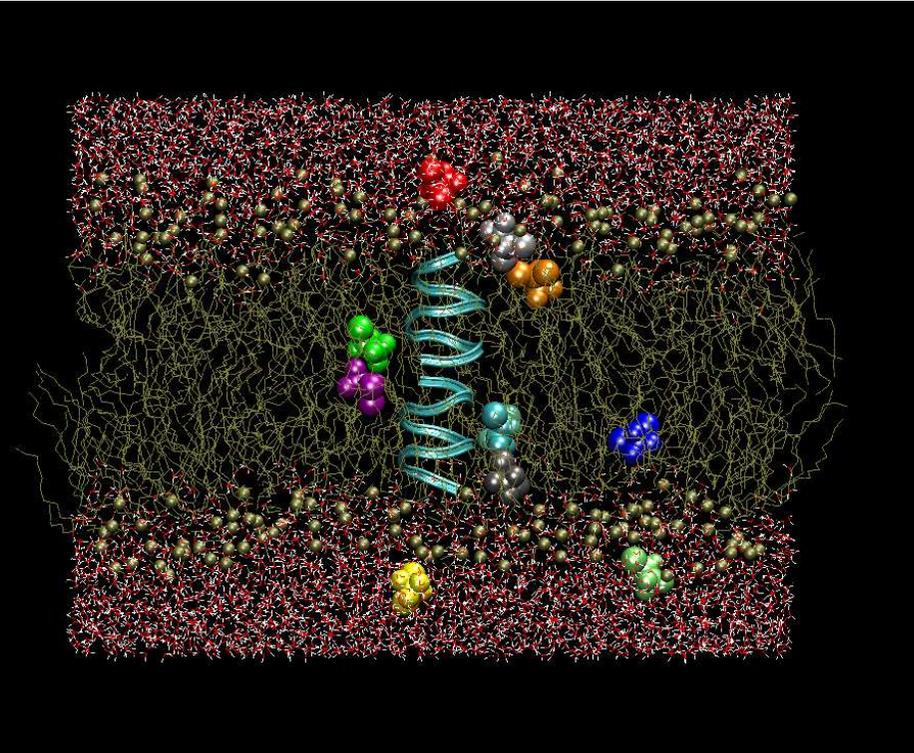
Água+íons



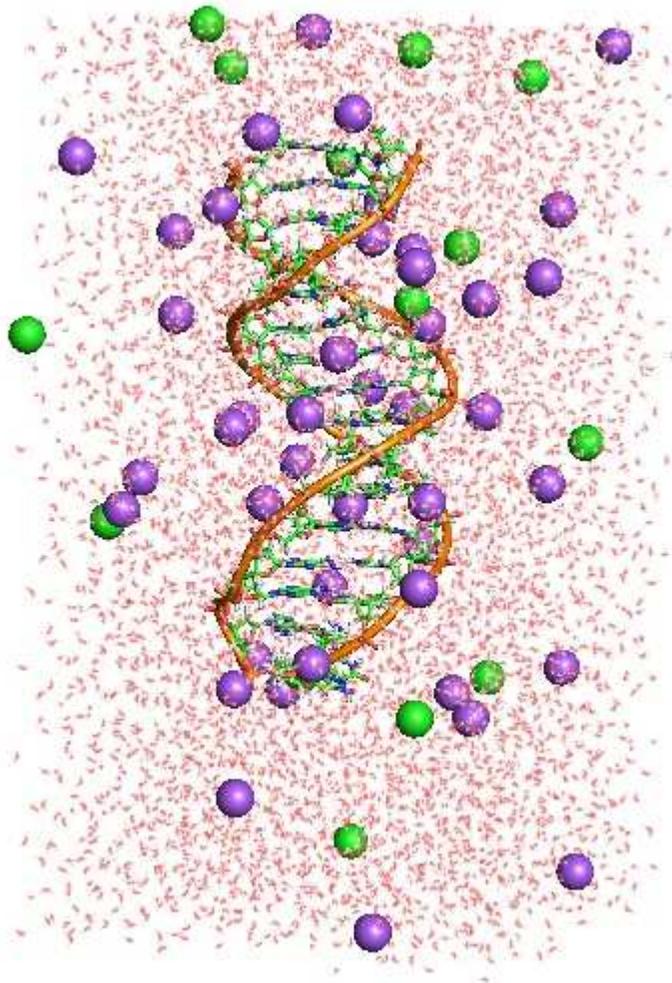
Membrana+água+íons



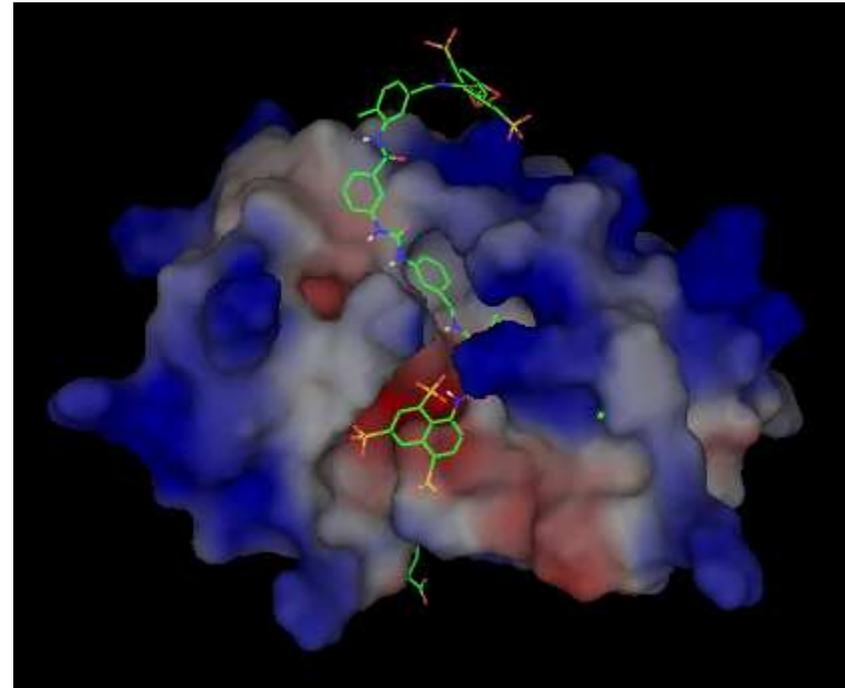
Membrana+peptídeo+molécula X+água+íons



DNA+água+íons



Enzima+inibidor



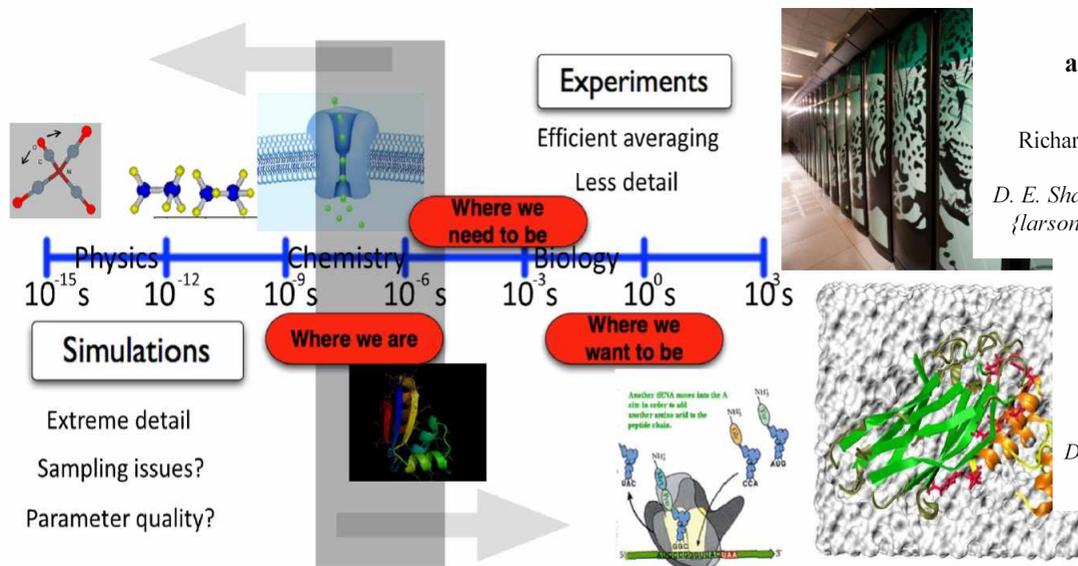
## Sinergia Experimento – Simulação (DM)



www.rcsb.org/pdb



- Experimentos motivam e guiam os estudos por DM (mais de 77000 estruturas depositadas PDB).
- Os resultados obtidos por DM fornecem explicações detalhadas a nível microscópico dos fenômenos observados experimentalmente – Propriedades energéticas, dinâmicas e estruturais.
- Diversidade de sistemas tratáveis: gases, fluídos supercríticos, líquidos, soluções e misturas, interfaces, filmes de Langmuir-Blodgett, membranas, polímeros, proteínas, sacarídeos, lipídeos, zeólitos, sólidos cristalinos, nanomateriais e etc.



### High-Throughput Pairwise Point Interactions in Anton, a Specialized Machine for Molecular Dynamics Simulation

Richard H. Larson, John K. Salmon, Ron O. Dror, Martin M. Deneroff, Cliff Young, J.P. Grossman, Yibing Shan, John L. Klepeis, and David E. Shaw\*  
 D. E. Shaw Research, 39th Floor, Tower 45, 120 West 45th Street, New York, NY 10036  
 {larsonr,salmonj,dror,deneroff,youngc,grossmaj,shan,klepeisj,shaw}@deshaw.com\*

### Incorporating Flexibility in Anton, a Specialized Machine for Molecular Dynamics Simulation

Jeffrey S. Kuskin, Cliff Young, J.P. Grossman, Brannon Batson, Martin M. Deneroff, Ron O. Dror, and David E. Shaw\*  
 D. E. Shaw Research, 39th Floor, Tower 45, 120 West 45th Street, New York, NY 10036  
 {kuskinj,youngc,grossmaj,batsonb,deneroff,dror,shaw}@deshaw.com\*

## Atomic-Level Characterization of the Structural Dynamics of Proteins

David E. Shaw,<sup>1,2\*</sup> Paul Maragakis,<sup>1†</sup> Kresten Lindorff-Larsen,<sup>1†</sup> Stefano Piana,<sup>1†</sup> Ron O. Dror,<sup>1</sup> Michael P. Eastwood,<sup>1</sup> Joseph A. Bank,<sup>1</sup> John M. Jumper,<sup>1</sup> John K. Salmon,<sup>1</sup> Yibing Shan,<sup>1</sup> Willy Griggers<sup>1</sup>

Molecular dynamics (MD) simulations are widely used to study protein motions at an atomic level of detail, but they have been limited to time scales shorter than those of many biologically critical conformational changes. We examined two fundamental processes in protein dynamics—protein folding and conformational change within the folded state—by means of extremely long all-atom MD simulations conducted on a special-purpose machine. Equilibrium simulations of a WW protein domain captured multiple folding and unfolding events that consistently follow a well-defined folding pathway; separate simulations of the protein's constituent substructures shed light on possible determinants of this pathway. A 1-millisecond simulation of the folded protein BPTI reveals a small number of structurally distinct conformational states whose reversible interconversion is slower than local relaxations within those states by a factor of more than 1000.

Many biological processes involve functionally important changes in the three-dimensional structures of proteins. Conformational changes associated with protein

folding (1), signal transduction (2), the catalytic cycles of enzymes (3), and the operation of molecular machines and motor proteins (4) often involve transitions among two or more structur-

ally distinct states. These states are often characterized as “basins” separated by barriers on a “energy landscape” (5).

Substantial progress has been made, using both experimental (1, 6) and computational (7, 8) techniques, in characterizing conformational basins and the ways that proteins move within and among them. It has proven difficult, however, to structurally characterize sparsely populated or disordered states and to elucidate the “basin-hopping” mechanisms involved in the interconversion of various states.

All-atom molecular dynamics (MD) simulations are designed to provide a high-resolution view of the motions of biological macromolecules (9), producing continuous trajectories with the potential to connect static structural snapshots generated from experimental data. Computational constraints, however, have limited such

<sup>1</sup>D. E. Shaw Research, 120 West 45th Street, New York, NY 10036, USA. <sup>2</sup>Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10032, USA.

\*To whom correspondence should be addressed. E-mail: David.Shaw@DEShawResearch.com

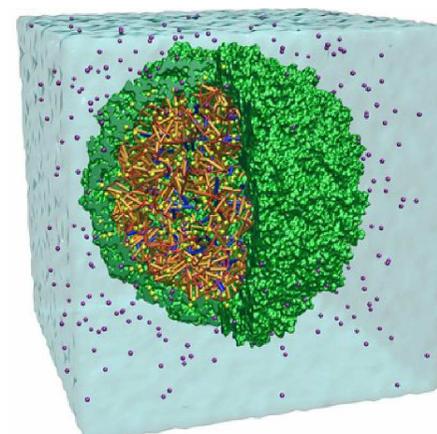
†These authors contributed equally to this work.

VOLUME 35 NUMBER 6  
**ACCOUNTS**  
*of*  
**CHEMICAL**  
**RESEARCH**<sup>®</sup>  
 JUNE 2012

Registered in U.S. Patent and Trademark Office. Copyright 2012 by the American Chemical Society

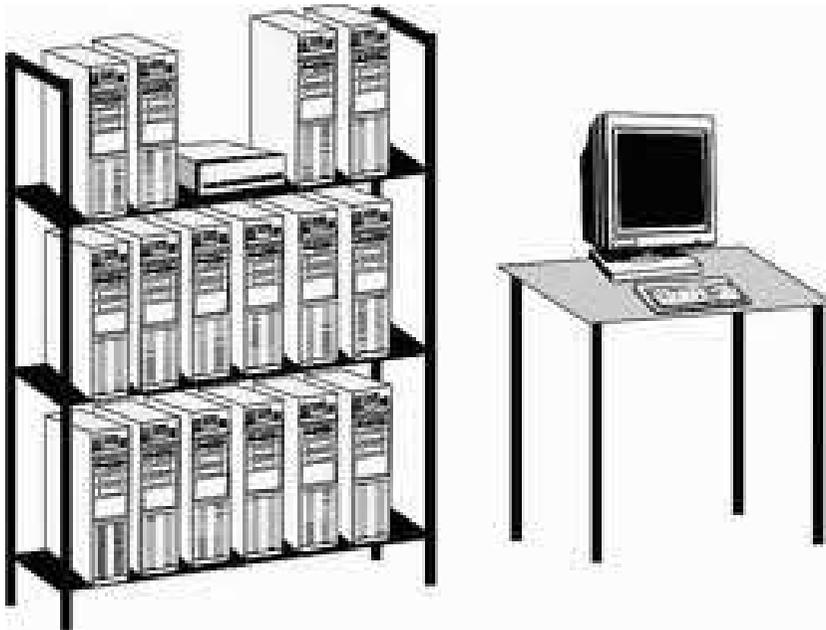
GUEST EDITORIAL

Molecular Dynamics Simulations of Biomolecules



## Equipamentos – Cluster de computadores

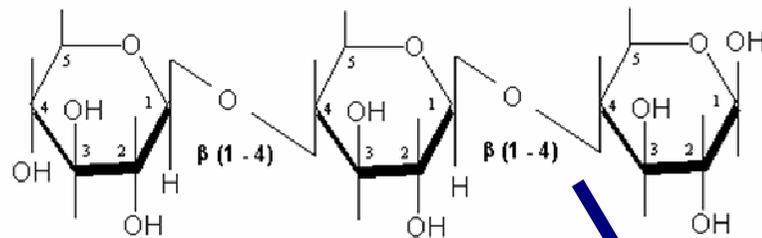
Aprox. 240 CPU's +GPU  
\$190.000



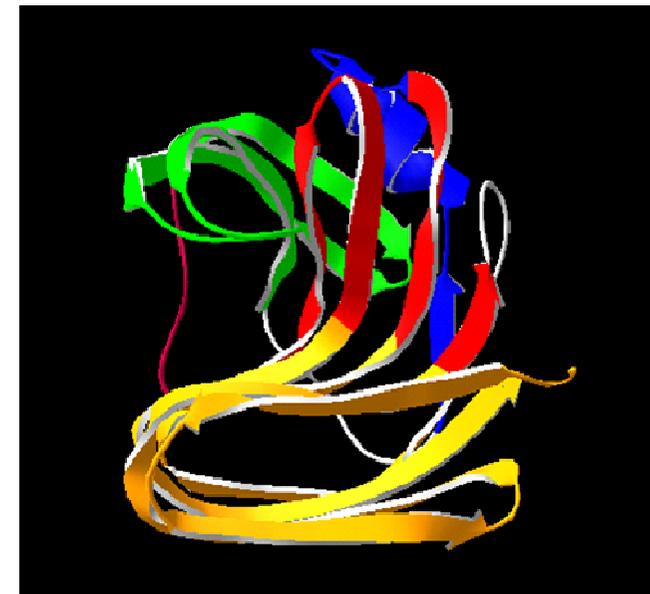
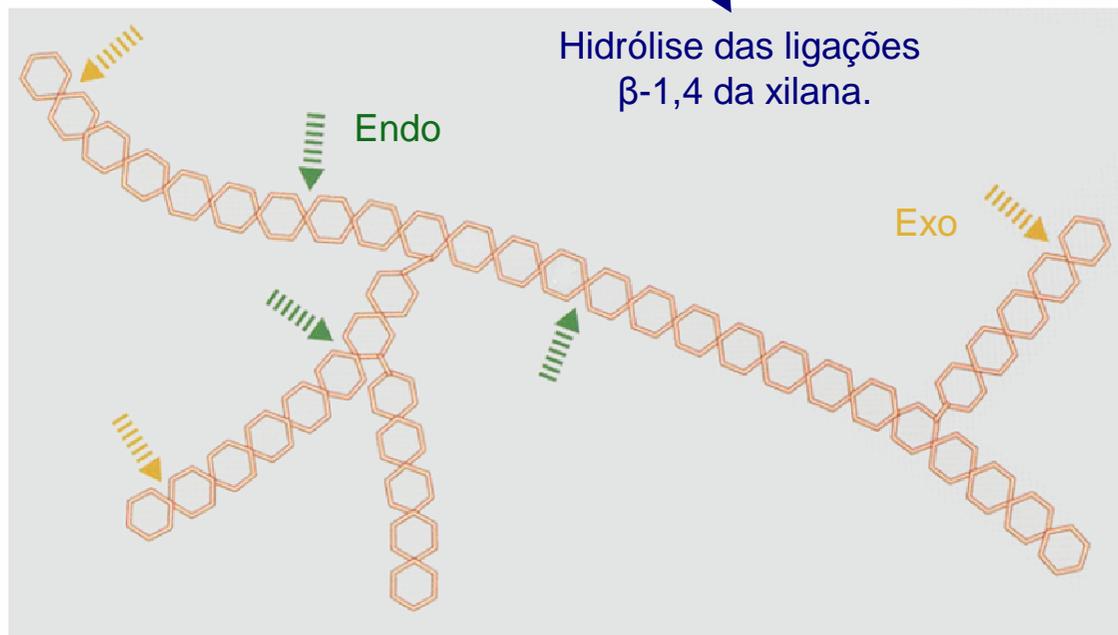


# Xilanases

Grupo das glicosil-hidrolases que catalisam a degradação da xilana.



*Um dos principais componentes do material hemicelulósico das plantas, formando interface entre a lignina e a celulose.*



→ Doutorado: Termoestabilidade de um par mesofílico-termofílico de xilanases

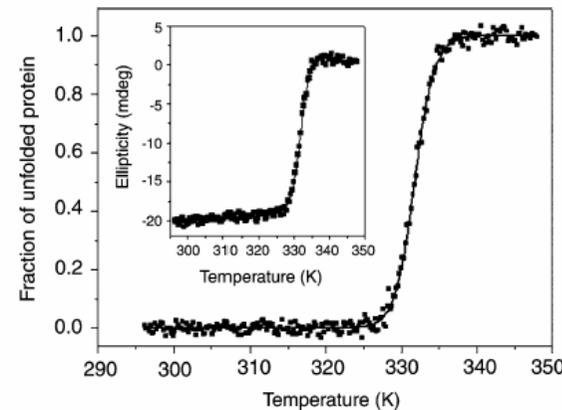
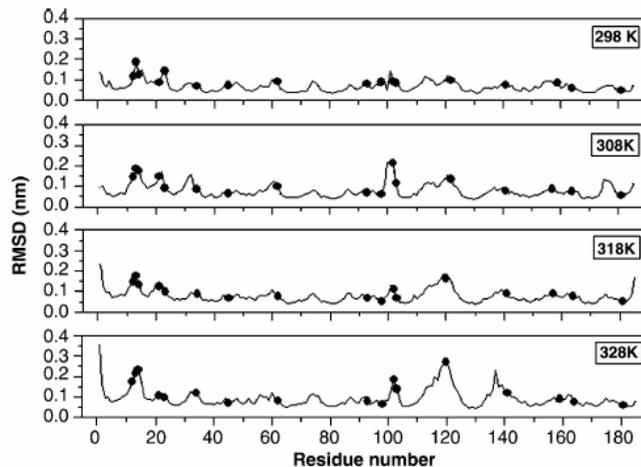
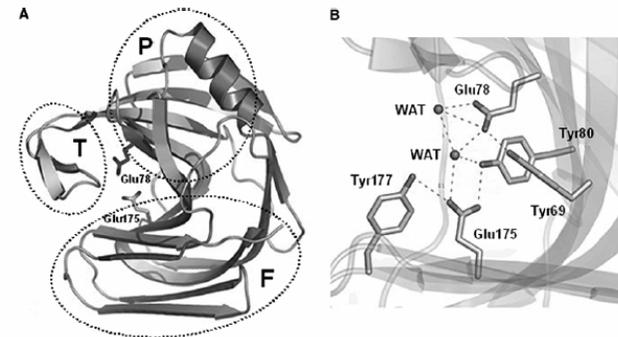
*Bacillus circulans* (BCX: 55°C) e *Termomyces lanuginosus* (TLX: 70°C)

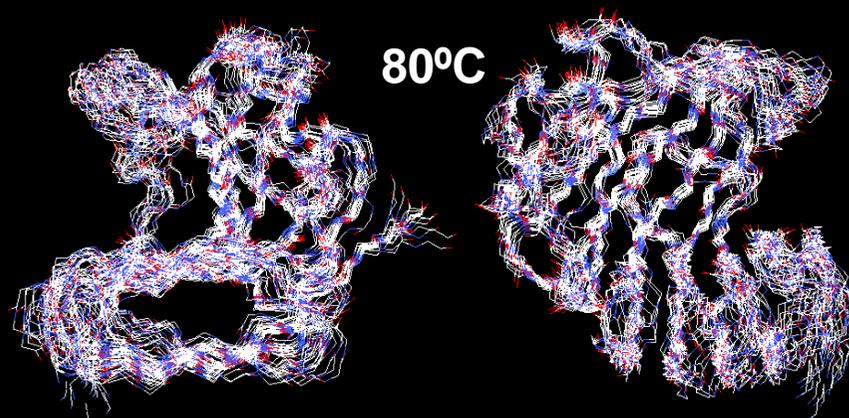
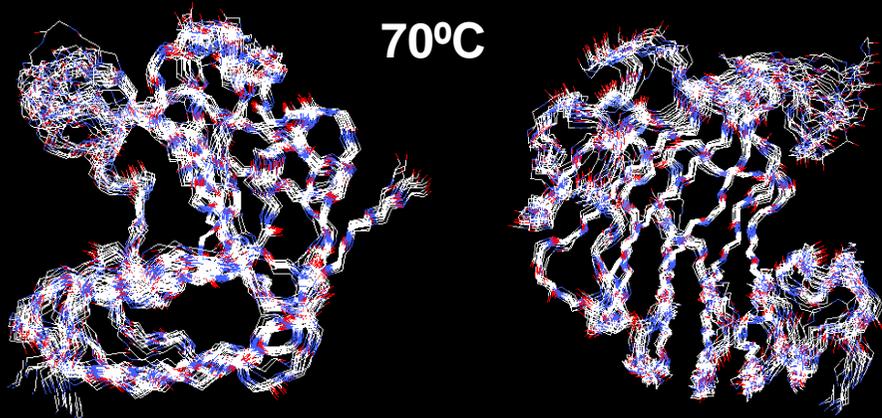
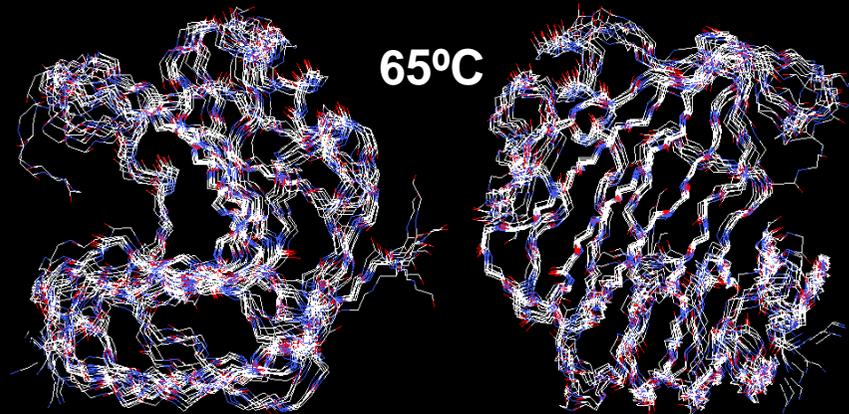
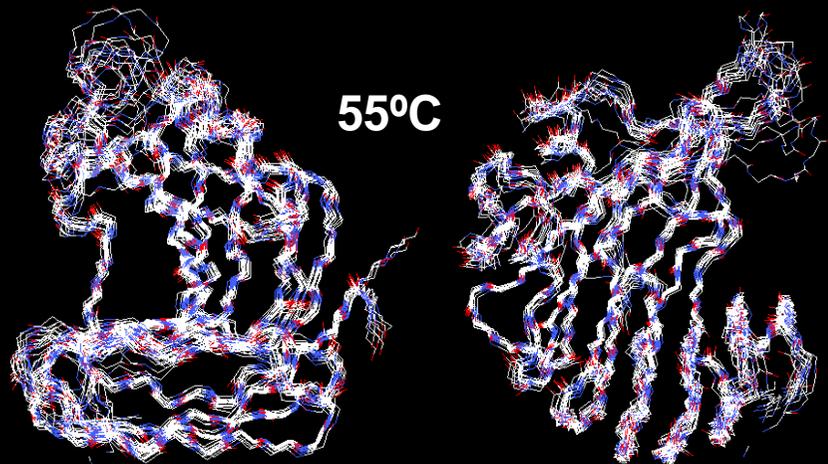
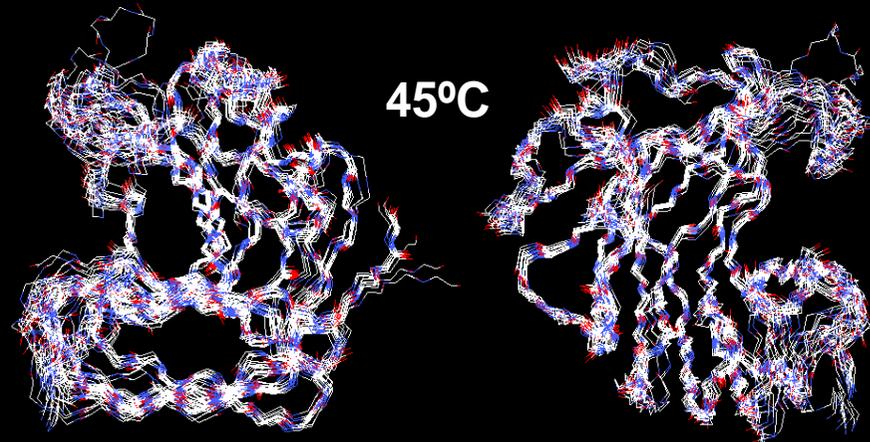
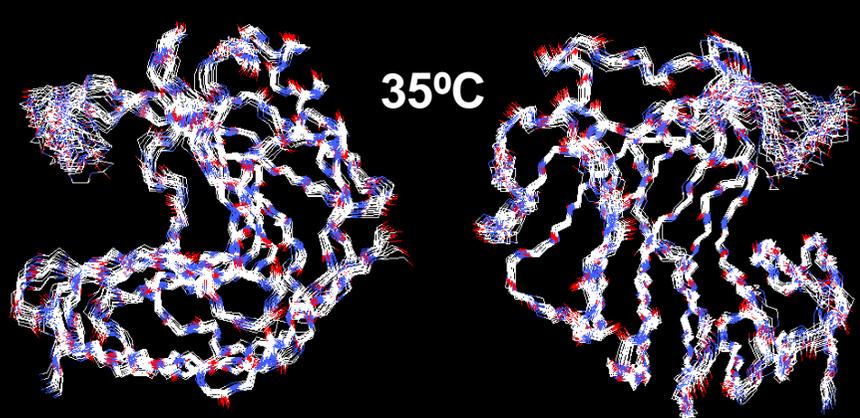


## Correlation of temperature induced conformation change with optimum catalytic activity in the recombinant G/11 xylanase A from *Bacillus subtilis* strain 168 (1A1)

Mário T. Murakami<sup>a</sup>, Raghuvir K. Arni <sup>\*,a</sup>, Davi S. Vieira<sup>c</sup>, Léo Degreève <sup>\*,c</sup>, Roberto Ruller<sup>b</sup>, Richard J. Ward<sup>c,\*</sup>

- 2.1. Expression, purification, crystallization and data collection
- 2.2. Structure solution and refinement
- 2.3. Thermal denaturation monitored by circular dichroism
- 2.4. Sequence alignment and structural analysis
- 2.5. Molecular dynamics simulations



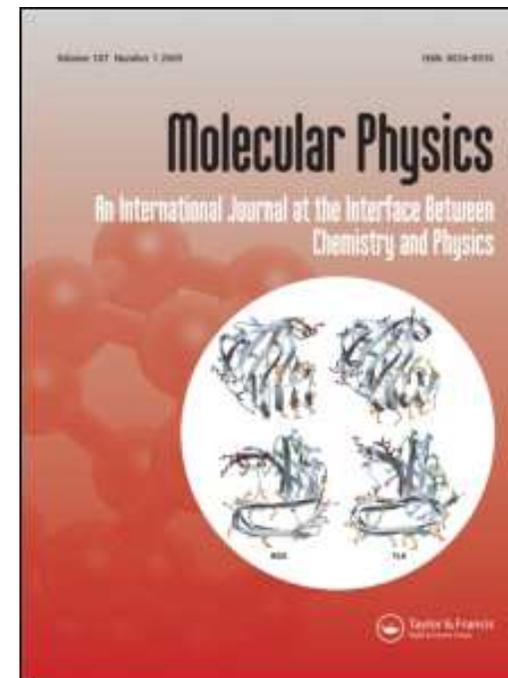
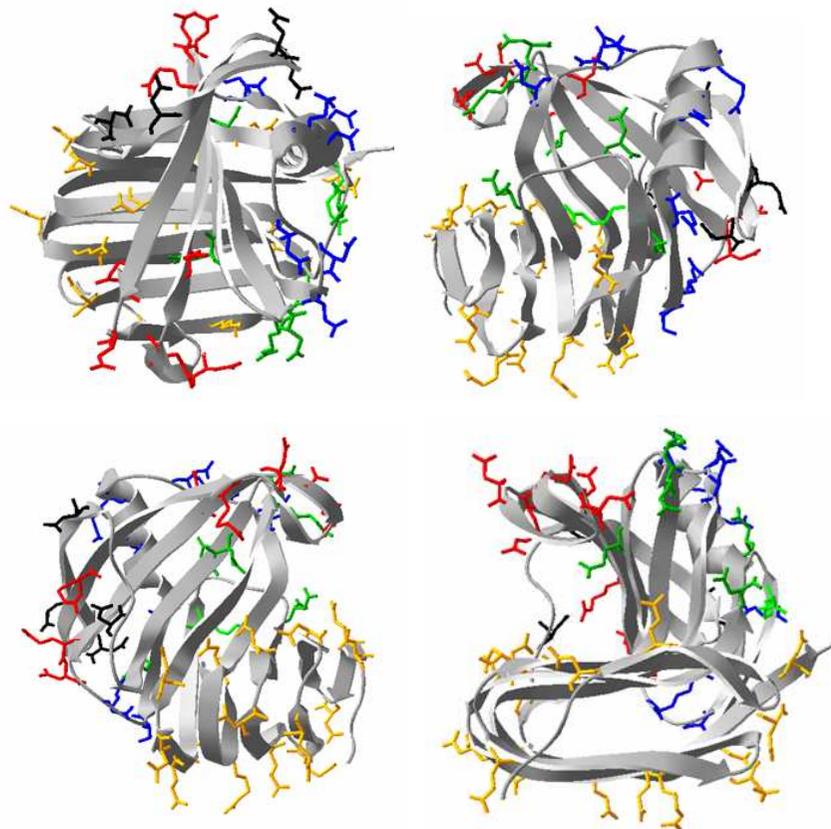


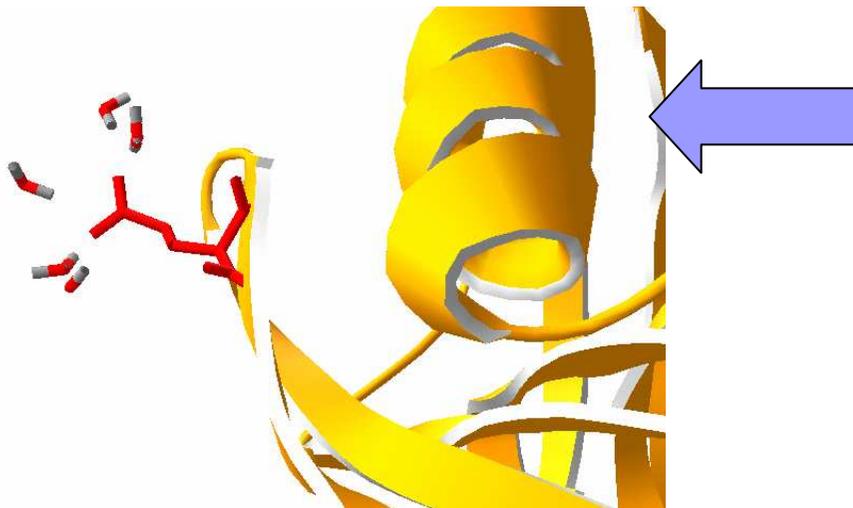
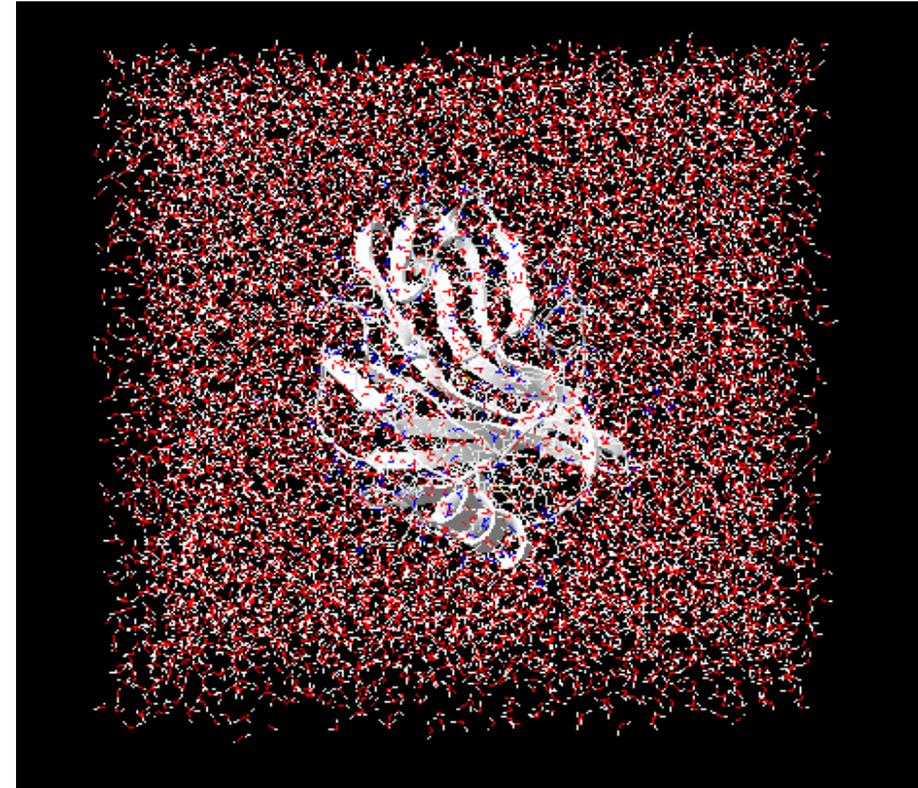
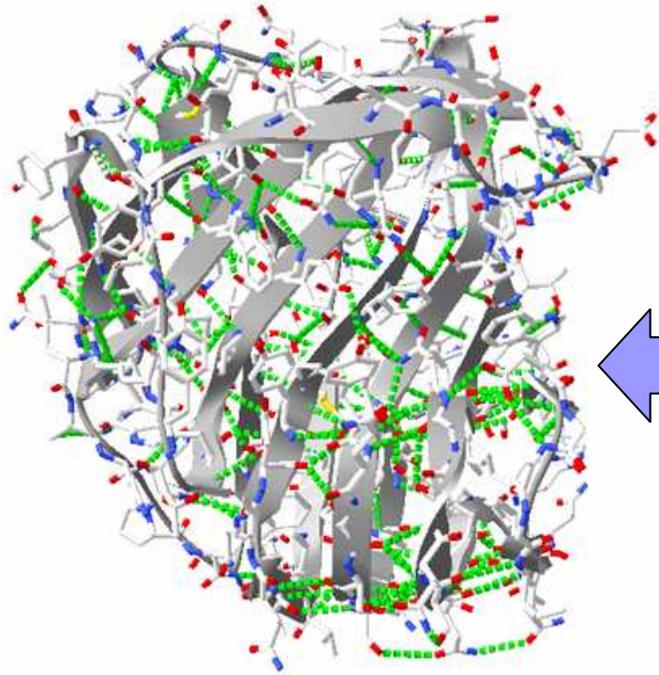
## RESEARCH ARTICLE

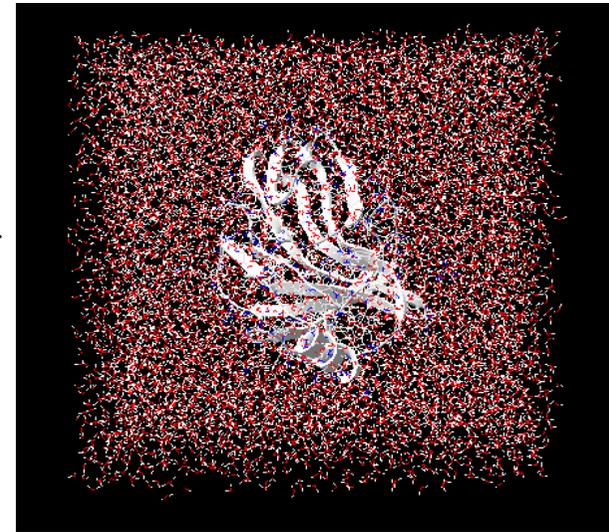
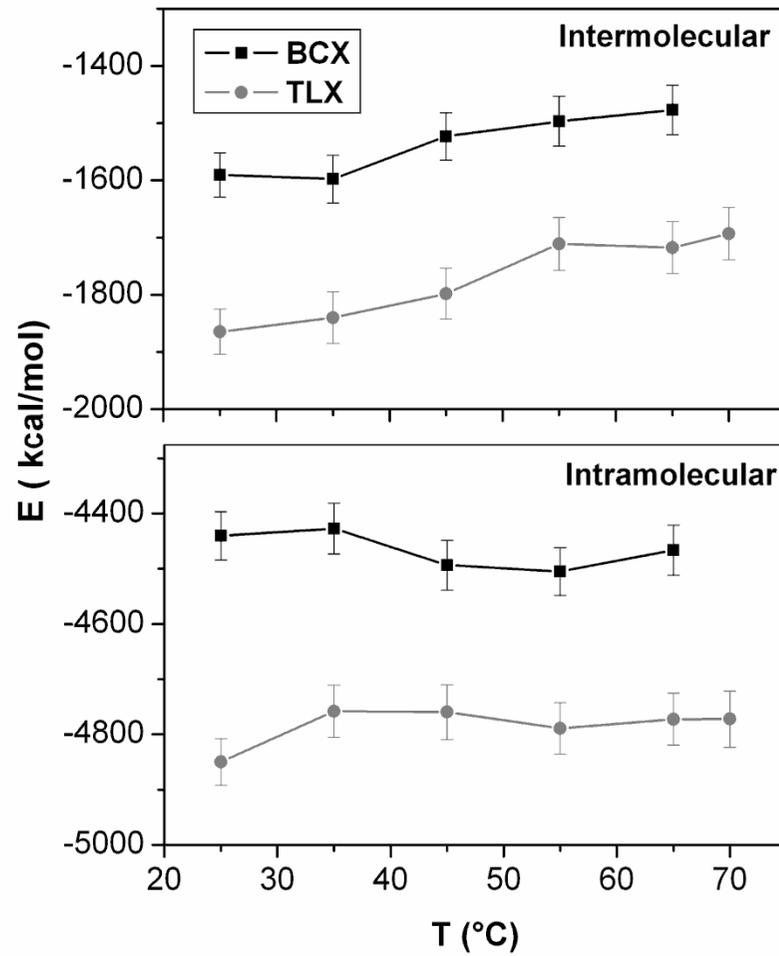
### An insight into the thermostability of a pair of xylanases: the role of hydrogen bonds

Davi Serradella Vieira\* and Léo Degrève

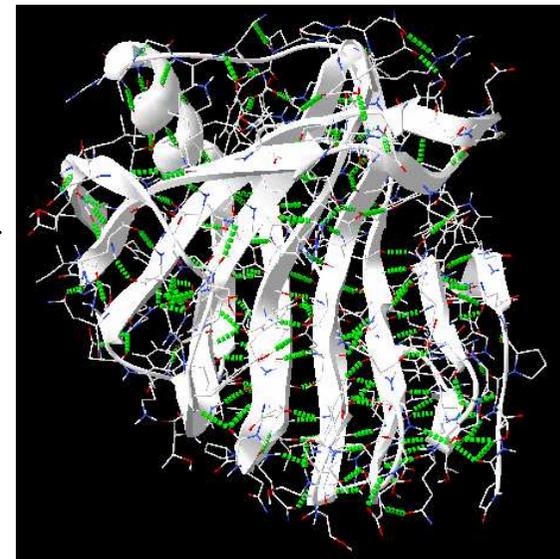
*Departamento de Química, Faculdade de Filosofia Ciências e Letras de Ribeirão Preto, Universidade de São Paulo,  
Avenida dos Bandeirantes, 3900 – 14040-901, Ribeirão Preto, SP, Brazil*







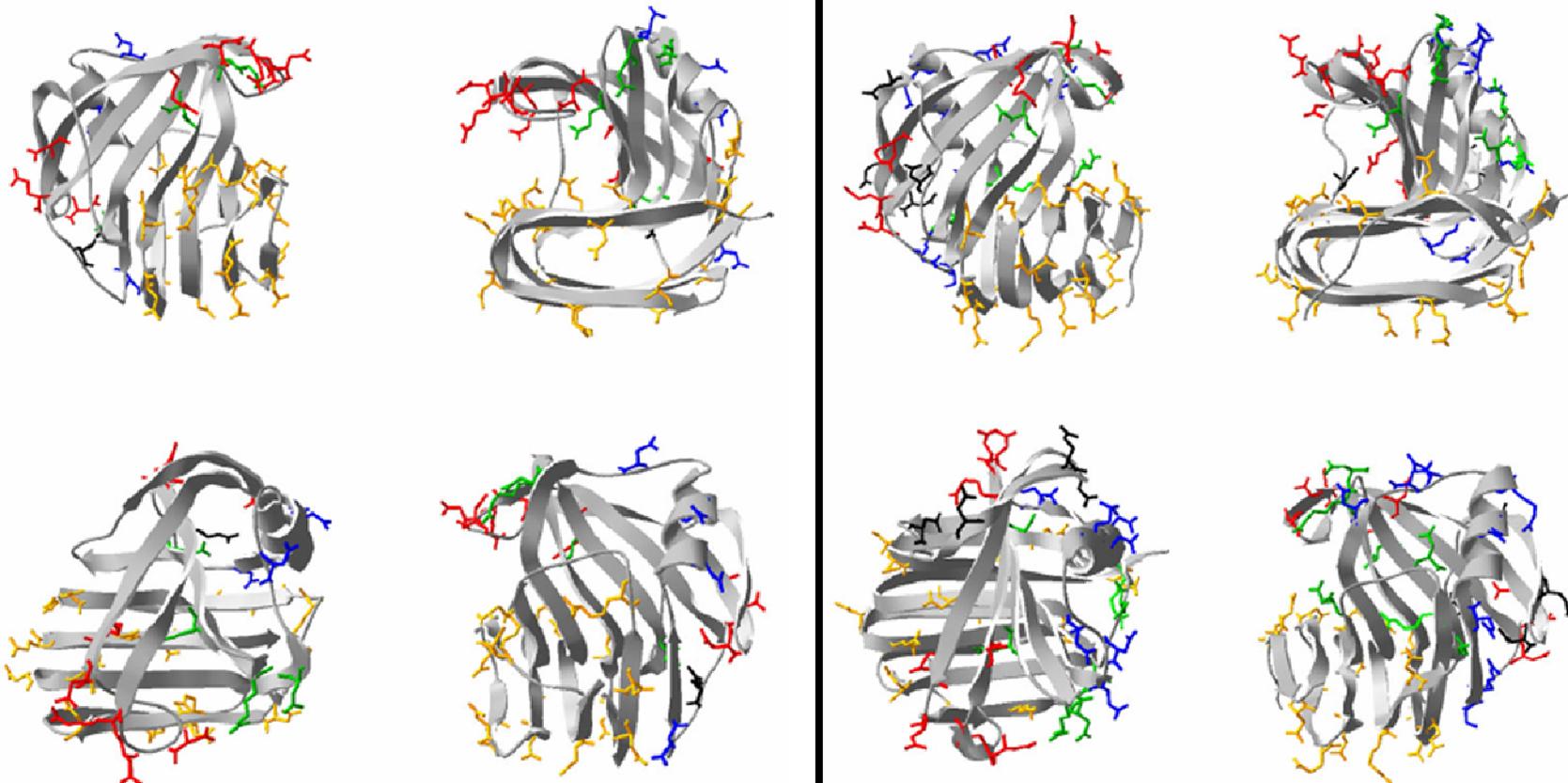
Ligações de Hidrogênio (LH)

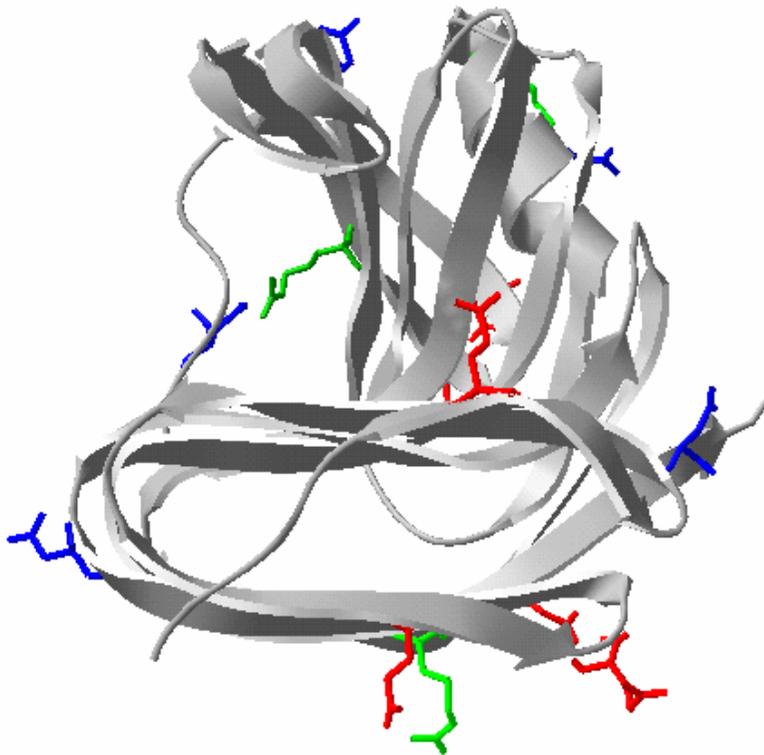


**LH Intermoleculares com energias mais atrativas que  $-5.0\text{kcal.mol}^{-1}$  !**

BCX

TLX





- ➔ *Identificar e quantificar os efeitos da termoestabilização.*
- ➔ *Identificar os resíduos e quantificar suas propriedades de estabilização.*
- ➔ *Propor mutações.*

**Projeto de mestrado CNPq**

Aluna: Juliana Sanchez Alponi

Orientador: Richard J Ward

*Estudo da termoestabilidade de uma endoxilanase G11 recombinante através de mutagênese sítio-dirigida.*

Resíduos Asp41, Asp97, Asp157, Asp135, Asp192, Glu7, Glu31, Glu36, Glu63, Glu91, Glu109, Arg116, Arg187 e Arg161. Os resíduos de Asp estão na cor azul, os de Glu na cor vermelha e os de Arg em verde.



Contents lists available at ScienceDirect

Biochimica et Biophysica Acta

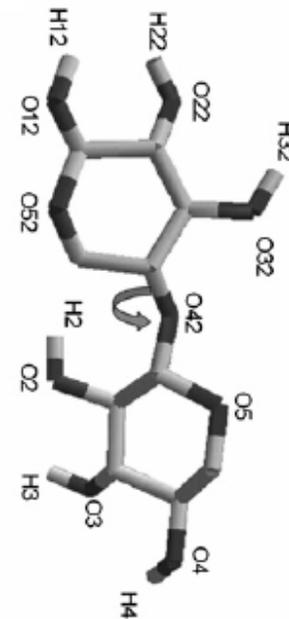
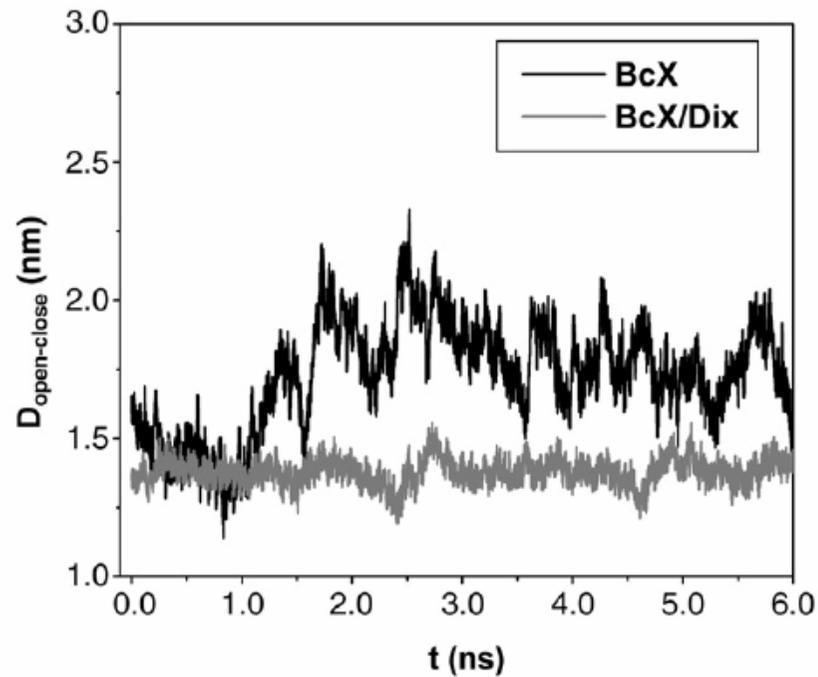
journal homepage: [www.elsevier.com/locate/bbagen](http://www.elsevier.com/locate/bbagen)



## Characterization of temperature dependent and substrate-binding cleft movements in *Bacillus circulans* family 11 xylanase: A molecular dynamics investigation

Davi Serradella Vieira\*, Léo Degrève, Richard John Ward

Departamento de Química, Faculdade de Filosofia Ciências e Letras de Ribeirão Preto – Universidade de São Paulo Ribeirão Preto, SP, Brazil



## MOLECULAR MODELING PRACTICAL

This tutorial introduces the student to the practice of Molecular Dynamics (MD) simulations of proteins. The protocol used is a suitable starting point for investigation of proteins, provided that the system does not contain non-standard groups. At the end of the tutorial, the student should know the steps involved in setting up and running a simulation, including some reflection on the choices made at different stages. Besides, the student should know how to perform quality assurance checks on the simulation results and have a feel for methods of analysis to retrieve information.

### QUICK LINKS

- [Introduction](#)
- [Starting structures](#)
- [Preparation](#)
- [Simulation setup](#)
- [Production run](#)
- [Analysis Part 1](#)

### MOLECULAR DYNAMICS HANDS-ON SESSION

Molecular dynamics (MD) simulations consist of three stages: First, the input data (the simulation system) have to be prepared. Second, the production simulation can be run and finally the results have to be analysed and be put in context. Although the second stage is the most computationally demanding step, with simulations commonly running for several months, the most laborious steps are the preparation and the analysis. This part of the tutorial provides an example session of setting up a protein for molecular dynamics simulation, running it and analysing the results.

This tutorial uses Gromacs (<http://www.gromacs.org>) version 4.5.3 and most of the commands given are specific to this package. The workflow(s) will be more general though and will also apply to other MD packages. Each step will be presented schematically, as a node with input and output. A step can be the execution of a single program or a combination of programs, with output from one being directed to the other. In the case of a program, the input/output types and flags will be specified. For most of the programs/steps only the input/output options used are listed. Most of the programs offer options for more advanced control, which will not be mentioned here. For those interested, the help page for each program gives a full listing and description of the functionality. The help page can be shown by running a program with the option "-h". In the workflow, programs are indicated by the name on a yellow background. Input/output data is given as a descriptive name on a green background. The blue blocks indicate input data which is essentially not dependent on the structure under study. The orange blocks are processes comprising of several steps in terms of the programs involved. In the present scheme, orange blocks are solely used for the simulation steps, which are explained in more detail below. The lines connect the data to the processes and for each program the command line option for the input or the output is indicated in white on black.



## PREPARING THE SYSTEM

The preparation of the simulation system is the most important stage of MD. MD simulations are performed to gain insight into processes on an atomistic scale. They can be used to understand experimental observations, to test hypotheses or as a basis for new hypotheses to be assessed experimentally. However, for each of these cases, the simulations have to be designed such that they are suited for the purpose. This means that the setup should be performed with care.

### **Missing residues, missing atoms and non-standard groups**

In this tutorial we consider the simulation of a protein. The first step in the sequence then is the selection of a starting structure, which has been covered on the previous page. The second step should be checking whether the structure has missing residues and atoms, which should be modeled in some way. In our case, the structures are complete and this step is skipped.

Another problem which one should be aware of is that structure files may contain non-standard residues, modified residues or ligands, for which force field parameters are not yet available. Such groups should either be removed or parameters should be determined for them, which is an advanced topic in MD. The structures in this tutorial only consist of natural amino acids.

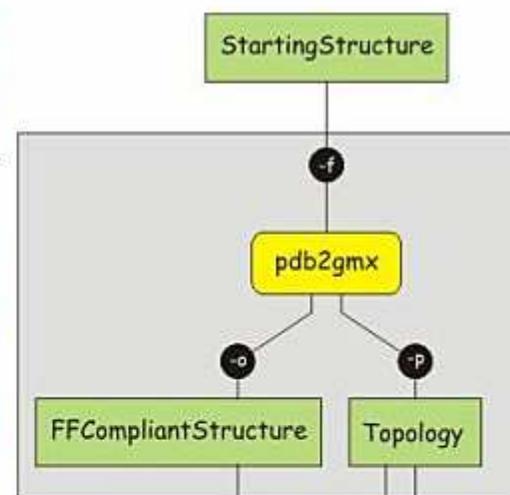
### **Structure quality**

It is good practice to run further checks on the structure to assess the quality. For instance, the refinement process in crystal structure determination may not always yield proper orientations of glutamine and asparagine amide groups. Also, the protonation state and side chain orientation of histidine residues may be problematic. A listing of the most relevant chemical properties of the proteinogenic amino acids is available on [Wikipedia](#). To perform a proper validation of the structure, several programs and servers exist (e.g. WHATIF). For this tutorial we will assume that the structure is good enough for our purpose and continue with preparing the simulation system.

## STRUCTURE CONVERSION AND TOPOLOGY

A molecule is defined by the coordinates of the atoms as well as by a description of the bonded and nonbonded interactions. Since the structure obtained from the PDB only contains coordinates, we first have to construct the topology, which describes the system in terms of atom types, charges, bonds, etc. This topology is specific to a certain force field and the force field to be used is one of the issues requiring careful consideration. Here we use the GROMOS96 53a6 united atom force field, which is parameterized to give good predictions of free energies of solvation of amino acid side chains and which generally gives good agreement with NMR experiments.

It is important that the topology matches with the structure, which means that the structure needs to be converted too, to adhere to the force field used. To convert the structure and construct the topology, the program `pdb2gmx` can be used. This program is designed to build topologies for molecules consisting of distinct building blocks, such as amino acids. It uses a library of building blocks for the conversion and will fail to recognize molecules or residues not present in the library. Issue the following command to convert the structure; choose the GROMOS 53a6 force field and the SPC water model when prompted. Note the flag `-ignh`, which causes hydrogen atoms present in the file to be removed, and to be rebuilt according to the description in the force field.



```
pdb2gmx -f protein.pdb -o protein.gro -p protein.top -ignh
```

**You can always generate a `pdb` file from a `gro` file using the following command:**

```
editconf -f XXXXX.gro -o XXXXX.pdb
```

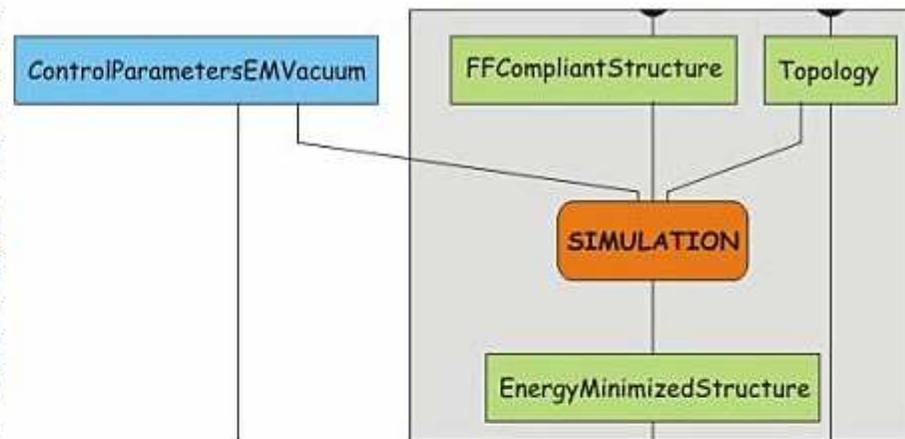
Read through the output on the screen and check the choices made for the histidine protonation and the resulting total charge of the protein. Also browse through the input structure file (`protein.pdb`, `pdb` format) and output structure file (`protein.gro`, `gromacs` format). Note the differences between the two formats. Browse then through the topology file and look at the structure.

Write down the number of atoms before and after the conversion and explain the difference

List the atoms, atom types and charges from a tyrosine residue as given in the topology file

## ENERGY MINIMIZATION OF THE STRUCTURE (VACUUM)

Now a structure is generated in the correct format for the force field chosen, as well as the corresponding topology. However, the conversion of the structure involves the deletion and or addition of hydrogen atoms and may cause strain to be introduced, e.g. due to atoms positioned too close together. Therefore, we have to perform an energy minimization on the structure, and let it relax a bit. This is in fact a **simulation step**, and involves two processes. First, the structure and the topology are combined into a single description of the system, together with a number of control parameters. This yields a run input file, which can be used as the single input for the simulation program mdrun. The advantage of splitting the two sub-processes is that the run-input file can be transferred to a dedicated (high-performance) computer network or a supercomputer and be processed there. However, this is only relevant for the production simulation.



To perform these steps first save the parameter file `minim.mdp` to your directory. This file contains the control parameters for the energy minimization. Have a look at the contents of this file. Notice the line starting with "integrator", which specifies the algorithm to be used. In this case it is set to perform a steepest descent energy minimization of 5000 steps. Now, combine the structure, topology and control parameters using grompp:

```
grompp -v -f minim.mdp -c protein.gro -p protein.top -o protein-EM-vacuum.tpr
```

grompp will note that the system has a non-zero charge and will print some other information regarding the system and the control parameters. The program will also generate an additional output file, which contains the settings for all control parameters (`mdout.mdp`). Now, continue with mdrun:

```
mdrun -v -deffnm protein-EM-vacuum -c protein-EM-vacuum.gro
```

Since there are only about 1500 atoms, the energy minimization finishes quickly. The `-v` flag causes the potential energy and the maximum force to be printed at each step, allowing following the minimization. The `-deffnm` flag sets the default filename for all file options, it avoids multiple flags usage. Now that the structure is relaxed, it is time to set up the periodic boundary conditions and add the solvent.

## minim.mdp

```
; Lines starting with ';' ARE COMMENTS
; Everything following ';' is also comment

title          = Energy Minimization ; Title of run

; The following line tell the program the standard locations where to find certain files
cpp            = /lib/cpp           ; Preprocessor

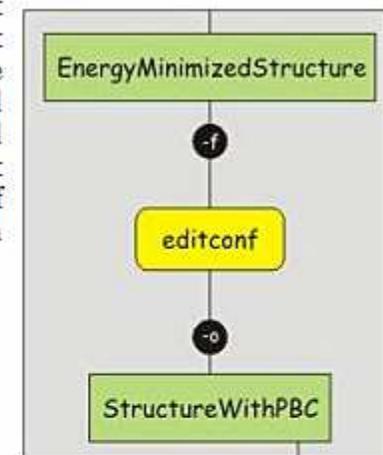
; Define can be used to control processes
define         = -DFLEXIBLE

; Parameters describing what to do, when to stop and what to save
integrator     = steep              ; Algorithm (steep = steepest descent minimization)
emtol         = 10.0                ; Stop minimization when the maximum force < 1.0 kJ/mol
nsteps        = 5000                ; Maximum number of (minimization) steps to perform
nstenergy     = 1                   ; Write energies to disk every nstenergy steps
energygrps    = System              ; Which energy group(s) to write to disk

; Parameters describing how to find the neighbors of each atom and how to calculate the interactions
nstlist       = 1                   ; Frequency to update the neighbor list
ns_type       = grid                ; Method to determine neighbor list (simple, grid)
coulombtype   = Reaction-Field      ; Treatment of long range electrostatic interactions
epsilon_rf    = 78
rcoulomb      = 1.4                  ; long range electrostatic cut-off
rvdw          = 1.4                  ; long range Van der Waals cut-off
constraints   = none                 ; Bond types to replace by constraints
pbc           = xyz                 ; Periodic Boundary Conditions (yes/no)
```

## PERIODIC BOUNDARY CONDITIONS

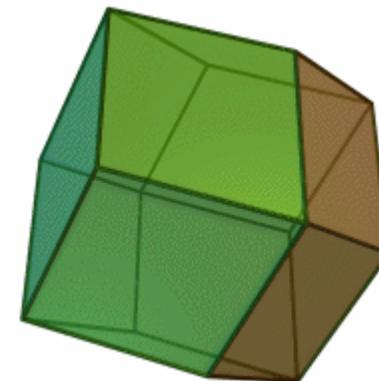
Before adding the solvent, a general layout (the space) of the simulation setup has to be chosen. Most commonly simulations are performed under periodic boundary conditions (PBC), meaning that a single unit cell is defined, which can be stacked in a space filling way. In this way, an infinite, periodic system can be simulated, avoiding edge effects due to the walls of the simulation volume. There are only a few general shapes available to set up PBC. We will use a **rhombic dodecahedron** because it corresponds to the optimal packing of a sphere, and is therefore the best choice for freely rotating molecules. To disallow direct interactions between periodic images we set a minimal distance of 1.0 between the protein and the wall of the cell, such that two neighbours will not be closer than 2.0 nm. Periodic boundary conditions are set with editconf:



```
editconf -f protein-EM-vacuum.gro -o protein-PBC.gro -bt dodecahedron -d 1.2
```

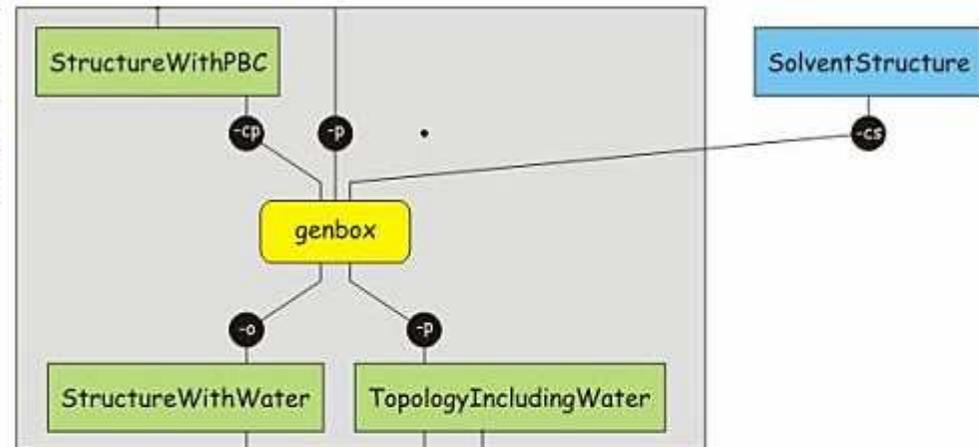
Have a look at the output of editconf, and note the changes in the volume. Also, have a look at the last line of the file protein-PBC.gro . In the gromacs format (.gro), the last line specifies the unit cell shape. It always uses the triclinic matrix representation, with the first three numbers specifying the diagonal elements (xx, yy, zz) and the last six giving the off-diagonal elements (xy, xz, yx, yz, zx, zy).

What is the new unit cell volume?



## SOLVENT ADDITION

Now that the unit cell is set up, solvent can be added. There are several solvent models, each of which is more or less intimately linked to a force field. The GROMOS96 force fields are generally used with the simple point charge (SPC) water model, which you should have chosen *before*. The topology is not required for solvent generation, but may be updated to include the solvent added. To fill the unit cell with SPC issue the following command:



```
genbox -cp protein-PBC.gro -cs spc216.gro -p protein.top -o protein-water.gro
```

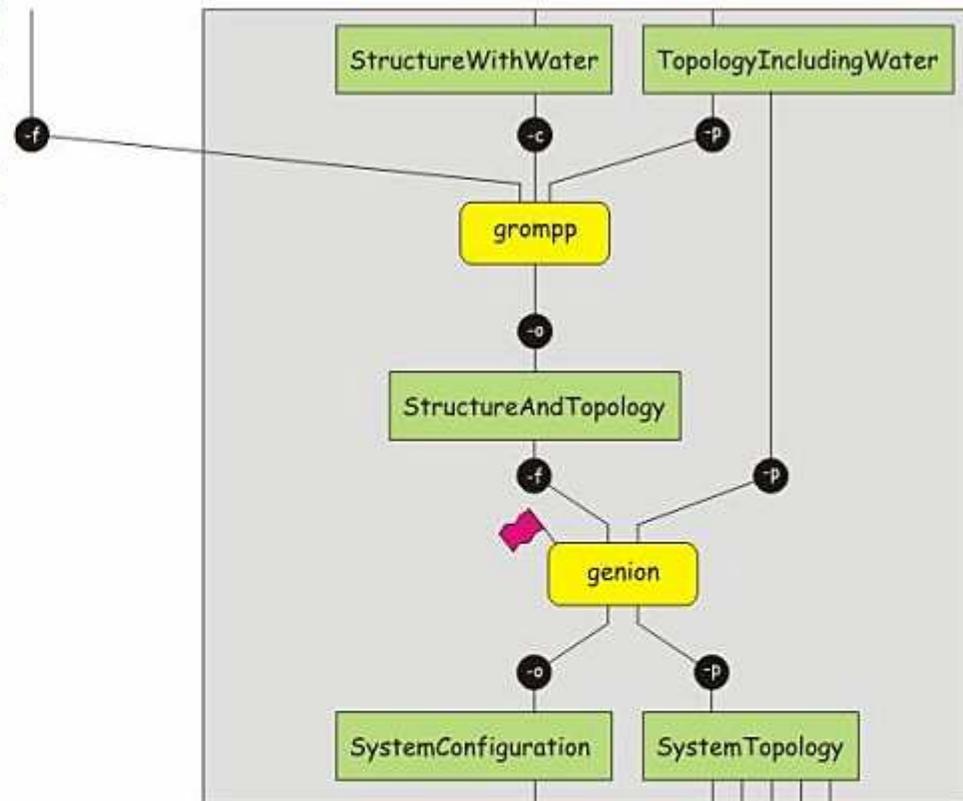
Have a look at the end of the file protein.top to see the addition, and check the numbers with the amount of solvent added according to genbox.

How many water molecules are added to the system and what volume does that correspond to?

How many atoms are in the system now?

## ADDITION OF IONS: COUNTER CHARGE AND CONCENTRATION

At this point, we have a solvated protein, but the net charge of the system still remains. To make the system neutral we have to add a number of counterions. In addition, it may be considered good practice to add ions up to a certain concentration. The program genion can take care of both tasks, but requires a run input file as input, i.e. a file containing both the structure and the topology. As we saw before, such a file can be generated with grompp:



```
grompp -v -f minim.mdp -c protein-water.gro -p protein.top -o protein-water.tpr
```

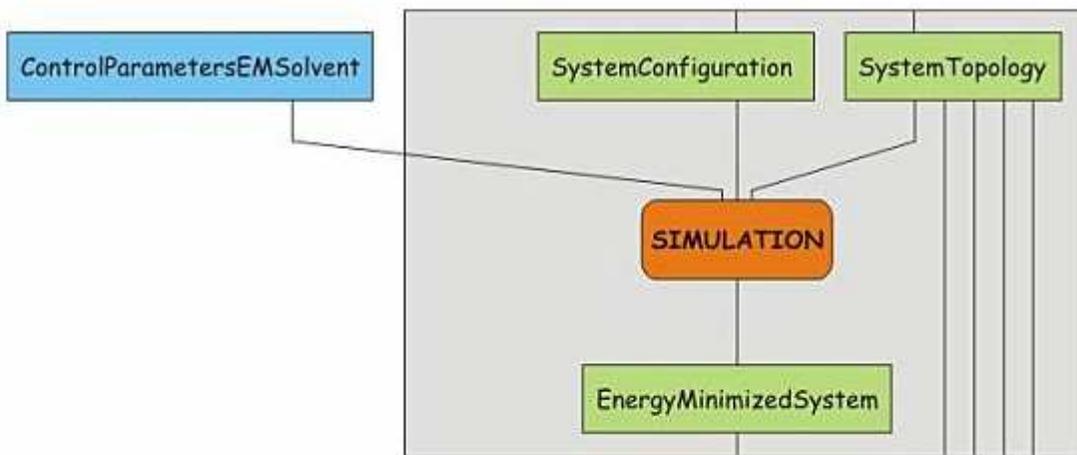
Subsequently, the file `protein-water.tpr` can be used as input for `genion`. We specify a concentration of 0.15 M (`-conc 0.15`) of `NA+/CL-` (`-pname NA+ -nname CL-`) to be added and indicate that an excess of one species of ion has to be added to neutralize the system (`-neutral`). `genion` will ask for a group of molecules to be used to partly replace with ions. The group 'SOL' should be chosen.

```
genion -s protein-water.tpr -o protein-solvated.gro -conc 0.15 -neutral -pname NA+ -nname CL-
```

Note the numbers of ions added, and verify that an excess of chloride ions is added for neutralization. Having replaced a number of water molecules with ions, the system topology in `protein.top` is not correct anymore.

## ENERGY MINIMIZATION OF THE SOLVATED SYSTEM

Now the whole simulation system is defined. The only steps before starting a production run involve the controlled relaxation of the system. The generation of solvent as well as the placement of ions, may have caused unfavourable interactions, e.g. overlapping atoms, equal charges too close together. Therefore, the system is energy minimized again, following the same steps as before Run grompp and mdrun:



```
grompp -v -f minim.mdp -c protein-solvated.gro -p protein.top -o protein-EM-solvated.tpr
```

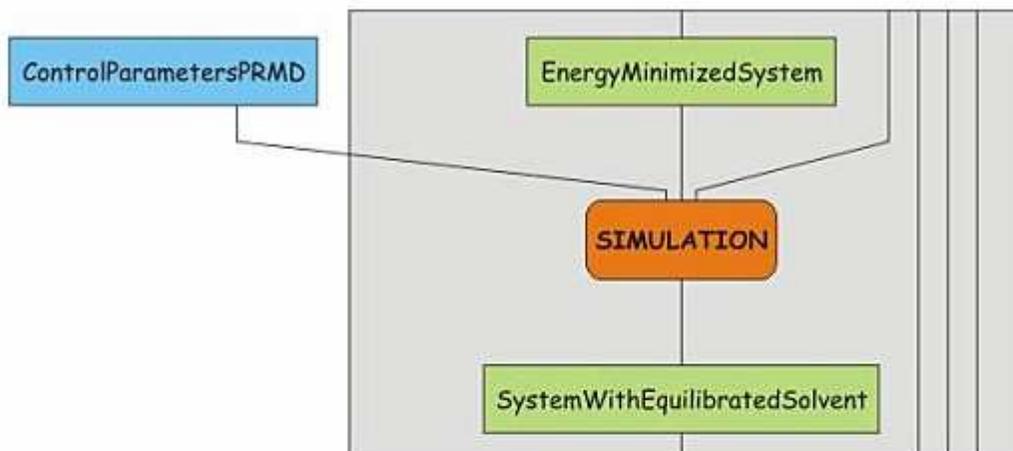
```
mdrun -v -deffnm protein-EM-solvated -c protein-EM-solvated.gro
```

How many steps were specified in the parameter file and how many steps did the minimization take?

What is the final potential energy of the system?

## RELAXATION OF SOLVENT AND HYDROGEN ATOM POSITIONS: POSITION RESTRAINED MD

With the greatest strain dissipated from the system, we now let the solvent adapt to the protein, i.e. we allow the solvent to move freely, while keeping the non-hydrogen atoms of the proteins more or less fixed to the reference positions. This is done to assure that the solvent configuration 'matches' the protein. This step is the first actual MD step. The control parameters for this step can be found in `pr.mdp`, which should be downloaded. Have a look at this file and note the integrator used as well as the define statement. The latter is used to allow flow control in the topology file. `define = -DPOSRES` will cause the keyword "POSRES" to be globally defined. Have a look at the end of the topology file to see how this is used to turn on position restraints. Use `grompp` and `mdrun` to run the simulation:



**Remark: edit the file `pr.mdp` and change the value of `gen_seed`**

```
grompp -v -f pr.mdp -c protein-EM-solvated.gro -p protein.top -o protein-PR.tpr
```

```
mdrun -v -deffnm protein-PR
```

## pr.mdp

```
; VARIOUS PREPROCESSING OPTIONS
title           = Position Restrained Molecular Dynamics
cpp             = /lib/cpp
define          = -DPOSRES

; RUN CONTROL PARAMETERS
integrator      = md
dt              = 0.001 ; time step (ps)
nsteps         = 2500 ; number of steps

; OUTPUT CONTROL OPTIONS
nstenergy      = 10
energygrps     = Protein Non-Protein

; NEIGHBORSEARCHING PARAMETERS
nstlist        = 5
ns_type        = grid
pbc            = xyz
rlist          = 0.9

; OPTIONS FOR ELECTROSTATICS AND VDW
coulombtype    = Reaction-Field
rcoulomb       = 1.4
epsilon_rf     = 78
vdw-type       = Cut-off
rvdw           = 1.4

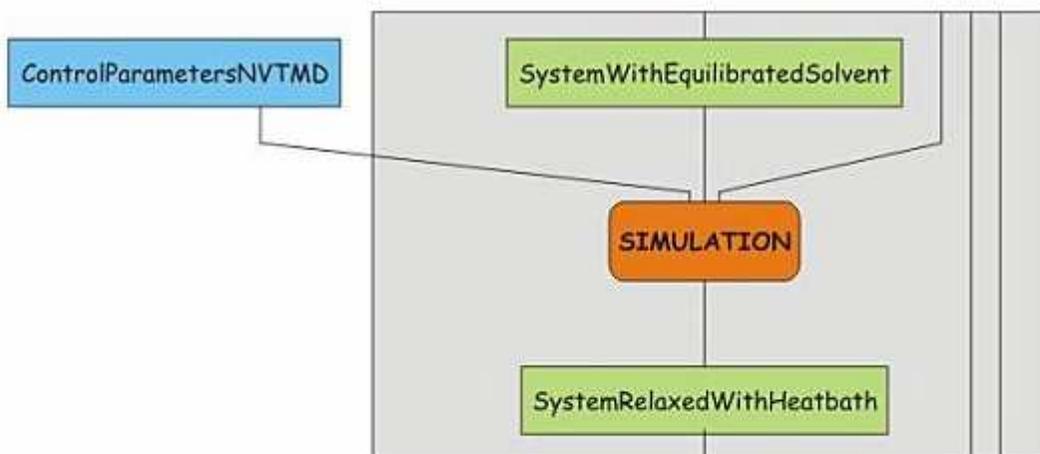
; Temperature coupling
tcoupl         = Berendsen           ; Couple temperature to external heat bath according to Berendsen method
tc-grps        = Protein Non-Protein ; Use separate heat baths for Protein and Non-Protein groups
tau_t          = 0.1 0.1             ; Coupling time constant, controlling strength of coupling
ref_t          = 200 200             ; Temperature of heat bath

; GENERATE VELOCITIES FOR STARTUP RUN
gen_vel        = yes                ; Assign velocities to particles by taking them randomly from a Maxwell distribution
gen_temp       = 200.0              ; Temperature to generate corresponding Maxwell distribution
gen_seed       = 9999               ; Seed for (semi) random number generation. Different numbers give different sets of velocities

; OPTIONS FOR BONDS
constraints    = all-bonds          ; All bonds will be treated as constraints (fixed length)
```

## UNRESTRAINED MD SIMULATION, TURNING ON TEMPERATURE COUPLING

The system should now be relatively free of strain. Time to turn up the heat, and start to couple the system to the heat bath. In other words, a short run will be performed in which the temperature coupling is turned on and the system is allowed to relax to the new conditions. Download the control parameter file [nvt.mdp](#) and have a look at the temperature coupling parameters in it. Then run the simulation, using `grompp` and `mdrun`:



```
grompp -v -f nvt.mdp -c protein-PR.gro -p protein.top -o protein-NVT.tpr
```

```
mdrun -v -deffnm protein-NVT
```

There are two groups which are separately coupled to a heat bath. Which groups are that and what do you think is in each of these groups?

After the run finishes have a look at the energies and temperature. Extract them in the same way as before. Note again that you may want to start the next simulation before looking at the energies from this one.

```
g_energy -f protein-NVT.edr -o thermodynamics-NVT.xvg
```

Have a look at the energy graphs.

```

; VARIOUS PREPROCESSING OPTIONS
title                = NVT simulation (constant number, volume and temperature)
cpp                  = /lib/cpp

; RUN CONTROL PARAMETERS
integrator           = md
dt                   = 0.002
nsteps               = 1250

; OUTPUT CONTROL OPTIONS
nstxout              = 0                ; No output, except for last frame (coordinates)
nstvout              = 0                ; No output, except for last frame (velocities)
nstfout              = 0                ; No output, except for last frame (forces)
nstlog               = 1                ; Write every step to the log
nstenergy            = 10               ; Write energies at every step
nstxtcout            = 0                ; Do not write a compressed trajectory
energygrps           = Protein Non-Protein ; Write energy information separately for these groups

; NEIGHBORSEARCHING PARAMETERS
nstlist              = 5
ns-type              = Grid
pbc                  = xyz
rlist                = 0.9

; OPTIONS FOR ELECTROSTATICS AND VDW
coulombtype          = Reaction-Field
rcoulomb             = 1.4
epsilon_rf           = 78
vdw-type             = Cut-off
rvdw                 = 1.4

; Temperature coupling
tcoupl               = Berendsen
tc-grps              = Protein Non-Protein
tau_t                = 0.1 0.1
ref_t                = 300 300

; Pressure coupling
pcoupl               = no

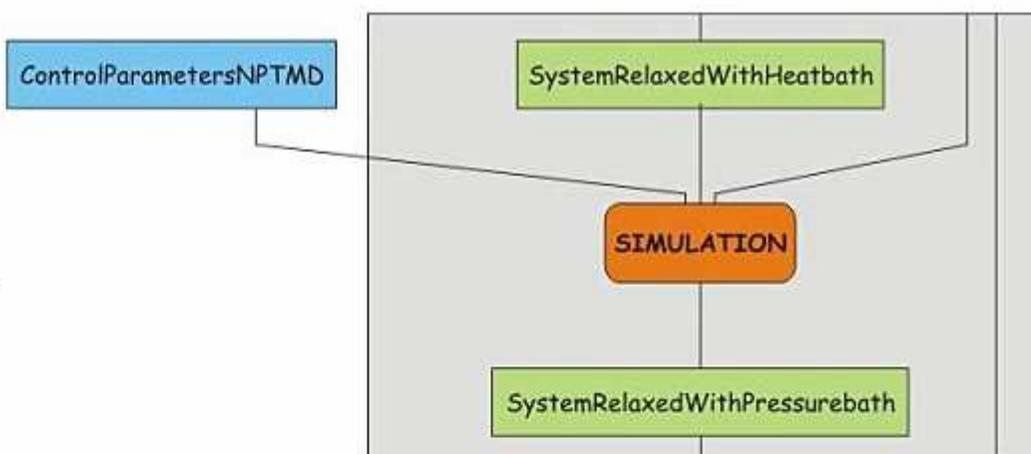
; OPTIONS FOR BONDS
constraints           = all-bonds

```

## UNRESTRAINED MD SIMULATION, TURNING ON PRESSURE COUPLING

After relaxation of the temperature and the temperature coupling, it is time to start putting things under pressure. Perform a short simulation with pressure coupling turned on, using the control parameter file `npt.mdp`. Have a look at this file and identify the parameters controlling the pressure coupling.

Also note that we reassign velocities to the particles, which is controlled with the parameter `gen_vel`. Below that line, the temperature for the distribution of velocities (the Maxwell distribution) is set and the seed for the random number generator is specified (`gen_seed`). `gen_seed` is currently set to 9999, but should be changed to a new value. 🤖



MD simulations are deterministic, so starting with the same set of coordinates and velocities and the same parameters would cause the outcomes of the simulations to be identical, which is not what we want.

After you've edited the parameter file prepare and run the simulation:

**Remark: edit the file `npt.mdp` and change the value of `gen_seed`**

```
grompp -v -f npt.mdp -c protein-NVT.gro -p protein.top -o protein-NPT.tpr
```

```
mdrun -v -deffnm protein-NPT
```

After the run finishes, again have a look at the energies and the temperature. Also have a look at the pressure. Extract all of them in the same way as before.

```
g_energy -f protein-NPT.edr -o thermodynamics-NPT.xvg
```

Have a look at the energy graphs, the temperature and the pressure.

```

; VARIOUS PREPROCESSING OPTIONS
title           = NVT simulation (constant number, volume and temperature)
cpp            = /lib/cpp

; RUN CONTROL PARAMETERS
integrator      = md
dt             = 0.002
nsteps        = 1250

; OUTPUT CONTROL OPTIONS
nstxout        = 0                ; No output, except for last frame (coordinates)
nstvout        = 0                ; No output, except for last frame (velocities)
nstfout        = 0                ; No output, except for last frame (forces)
nstlog         = 1                ; Write every step to the log
nstenergy      = 10               ; Write energies at every step
nstxtcout      = 0                ; Do not write a compressed trajectory
energygrps     = Protein Non-Protein ; Write energy information separately for these groups

; NEIGHBORSEARCHING PARAMETERS
nstlist        = 5
ns-type        = Grid
pbc            = xyz
rlist          = 0.9

; OPTIONS FOR ELECTROSTATICS AND VDW
coulombtype    = Reaction-Field
rcoulomb       = 1.4
epsilon_rf     = 78
vdw-type       = Cut-off
rvdw           = 1.4

; Temperature coupling
tcoupl         = Berendsen
tc-grps        = Protein Non-Protein
tau_t          = 0.1 0.1
ref_t          = 300 300

; Pressure coupling
pcoupl         = no

```

```

; Pressure coupling
Pcoupl         = Berendsen
Pcoupltype     = Isotropic
tau_p          = 1.0
compressibility = 4.5e-5
ref_p          = 1.0

```

```

; OPTIONS FOR BONDS
constraints    = all-bonds

```

## PRODUCTION SIMULATION

Now finally, we have a more or less equilibrated solvated system containing our protein of interest and it is time to start a production simulation. Mind that production simulation does not imply that the whole simulation can be used for analysis of the properties of interest. Although some of the memory of the initial configuration has been erased, it is unlikely that the system has already reached equilibrium. In the analysis section we will investigate which part of the simulation can be considered in equilibrium and is thus suited for further processing. But first it is time to set up the simulation. This merely involves running another simulation step, similar to the last step of the preparation. However, this is another of these times where it is necessary to think about the purpose of the simulation. The control parameters have to be chosen such that the simulation will allow analysing the properties one has in mind. Questions to be addressed are:

- At what time scale do the processes of interest manifest themselves or how long should the simulation be run?
- How many frames are necessary, or what time resolution is required?
- Is there need to store velocities?
- Should all atoms be in the output, or is it sufficient to save only protein coordinates?
- What frequency should be used for writing energies and log entries?
- What frequency should be used for writing check point coordinates and velocities?
- ...

We're going to run ten (10) nanosecond simulations. The simulations will be run on the NMR group cluster, so you only need to prepare the run input file.

Download or copy the control parameter file [md.mdp](#) and have a look at the contents of it.

How many steps are needed to get a total simulation time of 10 ns?

You'll have to edit the file and specify how many steps should be taken to get a total simulation time of 10 ns. Then take the final structure file and topology resulting from the preparation and combine them into a run input file using `grompp`.

Give the run input file a clear name, combining the protein identifier (E2Awt, E2Amut, E2Bwt or E2Bmut), your group number and your name. Also note that that the simulation will be run on four processors.

```
grompp -v -f md.mdp -c protein-NPT.gro -p protein.top -o topol.tpr
```

```

; VARIOUS PREPROCESSING OPTIONS
title           = Production Simulation
cpp            = /lib/cpp

; RUN CONTROL PARAMETERS
integrator      = md
tinit          = 0           ; Starting time
dt             = 0.002      ; 2 femtosecond time step for integration
nsteps         =           ; Make it 10 ns

; OUTPUT CONTROL OPTIONS
nstxout        = 50000     ; Writing full precision coordinates every nanosecond
nstvout        = 50000     ; Writing velocities every nanosecond
nstfout        = 0         ; Not writing forces
nstlog         = 2500      ; Writing to the log file every step
nstenergy      = 2500      ; Writing out energy information every step
nstxtcout      = 2500      ; Writing coordinates every 5 ps
energygrps     = Protein Non-Protein

; NEIGHBORSEARCHING PARAMETERS
nstlist        = 5
ns-type        = Grid
pbc            = xyz
rlist          = 0.9

; OPTIONS FOR ELECTROSTATICS AND VDW
coulombtype    = Reaction-Field
rcoulomb       = 1.4
epsilon_rf     = 78
epsilon_r      = 1
vdw-type       = Cut-off
rvdw           = 1.4

```

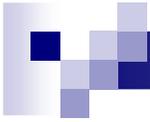


```
; Temperature coupling
Tcoupl          = Berendsen
tc-grps         = Protein  Non-Protein
tau_t           = 0.1      0.1
ref_t           = 300      300
; Pressure coupling
Pcoupl          = Berendsen
Pcoupltype      = Isotropic
tau_p           = 1.0
compressibility = 4.5e-5
ref_p           = 1.0

; GENERATE VELOCITIES FOR STARTUP RUN
gen_vel         = no

; OPTIONS FOR BONDS
constraints      = all-bonds
constraint-algorithm = Lincs
unconstrained-start = yes
lincs-order     = 4
lincs-iter      = 1
lincs-warnangle = 30
```

**mdrun -nt 64 -deffnm nome -v >& saida.log &**



**pdb2gmx** -f struct.pdb -o struct.gro -p topol.top -ignh -ter (-ss, -arg, -lys, -glu, -asp, -gln, -his)

**editconf** -f struct.gro -o box.gro -d (distância entre proteína-caixa) 1.50 -bt DODECAHEDRON

**genbox** -cp box.gro -cs spc216.gro -p topol.top -o solv.gro

**grompp** -f em.mdp -c solv.gro -p topol.tpr -o genion.tpr

**genion** -s genion.tpr -o full.gro -p topol.top -g -pot -rmin 0.6 -neutral -conc 0.15 -pname NA -pq 1 -nname CL -nq -1  
Corrigir número de águas e íons no arquivo de topologia - topol.top

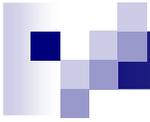
**grompp** -f em.mdp -c full.gro -p topol.tpr -o em.tpr

**mpirun** cX-Y **mdrun** -deffnm em -nice 4 -v > & saida.log &

ou

**mdrun** -np 4 deffnm em -v > & saida.log &

**mdrun** -nt 64 -deffnm md\_c04 -v >& saida.log &



davisv@ufrnet.br

FIM