

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
use Math::Trig ;
use strict;
```

```
##(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor{"O"}}( my $qqa=substr($r,0,1); if ( $qqa ne $ch) { $chnum++; $ch=$qqa; } );
my %qqa=find_quart( $coor{"O"});
```

```
if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^\.//;
$filename=~ s/\.pdb//;
#$filename=$chnum."_"$qnum."_"$filename.".dat";
$filename="$dir/".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";
```

```
foreach my $m (sort { $a<=>$b } keys %coor){
my %qartets;
my %q= find_q( $coor{$m} );
```

```
# foreach my $q { keys %qartets } { print join " ", @{$qartets{$q}}, "\n";
```

```
foreach my $q { keys %qartets } {
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ) {
```

```
# print "$q $coor{$m}{$res}{\"N\"->x,\"n\";
```

```
$nx=$nx+ $coor{$m}{ $res }{"N9"}->x;
$ny=$ny+ $coor{$m}{ $res }{"N9"}->y;
$nz=$nz+ $coor{$m}{ $res }{"N9"}->z;
```

```
$ox=$ox+ $coor{$m}{ $res }{"O6"}->x;
$oy=$oy+ $coor{$m}{ $res }{"O6"}->y;
$oz=$oz+ $coor{$m}{ $res }{"O6"}->z;
$r=$res;
```

```
}
```

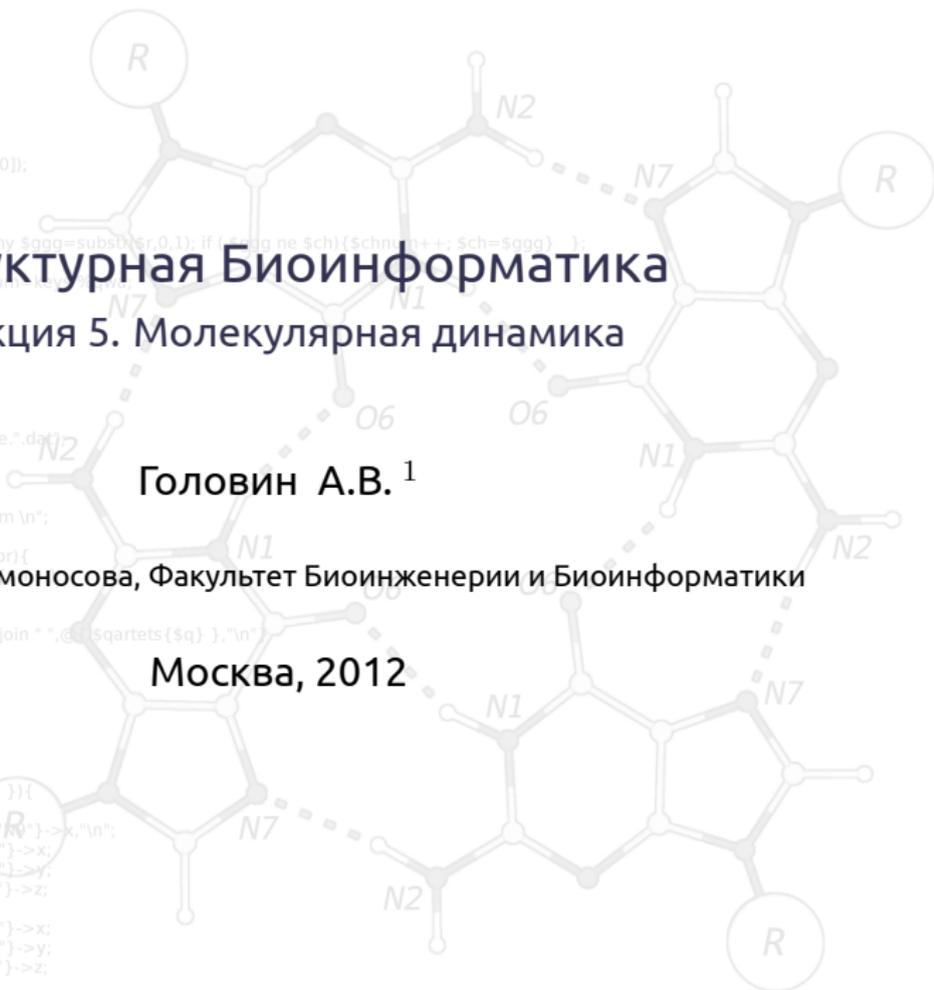
Структурная Биоинформатика

Лекция 5. Молекулярная динамика

Головин А.В. ¹

¹МГУ им М.В. Ломоносова, Факультет Биотехнологии и Биоинформатики

Москва, 2012



Содержание

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r { sort keys %{$coor{"0"}} } { my $ggg=substr($r,0,1); if ( $ggg ne $ch){ $chnum++; $ch=$ggg } };
my %qwa=find_quart( $coor{"0"} ); my $sqnum=keys %qwa;
```

Введение

```
if ($sqnum > 0){
#system("mkdir $ARGV[1]");
$filename="-- s/,.pdb//";
#$filename=$chnum.".".$sqnum.".".$filename.".dat";
$filename="$dir".$filename.".dat";
print "$filename\n";
open OUT ">$filename";
foreach my $m (sort { $a<=>$b } keys %coor){
my %qartets = %qwa; #find_quart( $coor{$m} );
my %q = find_q( $coor{$m} );
```

Молекулярная динамика

Температура и давление

Растворитель в МД

Особенности

```
foreach my $res ( @ { $qartets{$q} } ){
# print "$q $coor{$m} {$res} {"R"}->x,"n";
$nx=$nx+ $coor{$m} {$res} {"N9"}->x;
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;

$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Уравнение Шредингера

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %coor ) {
    my $q=($coor{$r}{x}+$coor{$r}{y}+$coor{$r}{z});
    my %qwa=find_quart($coor{$r}{x},$coor{$r}{y},$coor{$r}{z});
}

```

$$\left(-\frac{\hbar^2}{m} \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right] + V \right) \Psi(r, t) = i\hbar \frac{\partial \Psi(r, t)}{\partial t}$$

Или:

$$H\Psi = E\Psi; \quad H = \frac{-\hbar^2}{m} \nabla^2 - \frac{Ze^2}{4\pi\epsilon_0 r}$$

В молекулярной механике где аппроксимируем электронную плотность уравнениями классической физики.

```

# foreach my $q { keys %qartets } { print join " ", @qartets{$q}, "\n"; }
foreach my $q { keys %qartets } {
    my $nx; my $ny; my $nz;
    my $ox; my $oy; my $oz;
    my $r;
}

```

$$F = m \frac{\partial^2 r}{\partial t^2}$$

Осталось придумать как следить за эволюцией системы во времени.

```

# print "Sq $coor{$m}{x} {$res}{\"N9\"}->x, \"n\";
my $x=$coor{$m}{x} {$res}{\"N9\"}->x;
my $y=$coor{$m}{y} {$res}{\"N9\"}->y;
my $z=$coor{$m}{z} {$res}{\"N9\"}->z;

my $ox=$coor{$m}{x} {$res}{\"O6\"}->x;
my $oy=$coor{$m}{y} {$res}{\"O6\"}->y;
my $oz=$coor{$m}{z} {$res}{\"O6\"}->z;

```

#!/usr/bin/perl

use Math::VectorReal qw(:all);

Простое уравнение силового поля (СП)

```
#!/(my %$coor,$my $schnum)=read_pdb($ARGV[0]);
```

```
my %$coor=read_pdb($ARGV[0]);
```

```
my $dir=$ARGV[1];
```

```
my $sch,$my $schnum;
```

```
foreach my $r ( sort keys %{$coor{"0"}} ) { my $ggg=substr($r,0,1); if ( $ggg ne $sch ) { $schnum++; $sch=$ggg; };
```

$$U = \sum_{\text{bonds}} \frac{k_i}{2} (l_i - l_0)^2 + \sum_{\text{angles}} \frac{k_i}{2} (\phi_i - \phi_0)^2 + \sum_{\text{torsions}} \frac{V_n}{2} (1 + \cos(n\omega - \gamma)) +$$

```
if ($qnum > 0) {
```

```
  #system("mkdir $ARGV[1]");
```

```
  my $filename=$ARGV[0];
```

```
  $filename="-- s/^.*\//";
```

```
  $filename="-- s/\.pdb//";
```

```
  # $filename=$schnum."_$qnum".$filename; # $filename=$schnum."_$qnum".$filename;
```

```
  $filename="$dir".$filename;
```

```
  print "$filename\n";
```

```
  open OUT,">$filename";
```

```
  print OUT "#INFO chain $schnum $qnum\n";
```

```
  foreach my $m ( sort { $a <=> $b } keys %$coor ) {
```

```
    my %$qartets = %qwa; #find_quart( $coor{$m} );
```

```
    my %$q = find_q( $coor{$m} );
```

```
#   foreach my $q ( keys %$qartets ) { print join " ", @{$qartets{$q}}, "\n";
```

```
   foreach my $q ( keys %$qartets ) {
```

```
     my $nx; my $ny; my $nz;
```

```
     my $ox; my $oy; my $oz;
```

```
     my $r;
```

```
     foreach my $res ( @{$qartets{$q}} ) {
```

```
#       print "$q $coor{$m} {$res} {"$r"}->{ "N";
```

```
       $nx=$ny=$nz=$ox=$oy=$oz=$r; # $res {"$r"}->{ "N";
```

```
       $v(t + \frac{\Delta t}{2}) = v(t - \frac{\Delta t}{2}) + \frac{F(t)}{m} \Delta t
```

```
       $r=$r + \frac{\Delta t}{2} $coor{$m} {"$res"} {"$r"}->{ "N";
```

```
       $ox=$ox + $coor{$m} {"$res"} {"$r"}->{ "x";
```

```
       $oy=$oy + $coor{$m} {"$res"} {"$r"}->{ "y";
```

```
       $oz=$oz + $coor{$m} {"$res"} {"$r"}->{ "z";
```

Молекулярная
динамика

Метод Монте Карло

$$\text{acc}(o \rightarrow \eta) = \min \left(1, \exp \left\{ -\beta \left[U(r^N) - U(o^N) \right] \right\} \right)$$

#!/usr/bin/perl

use Math::VectorReal qw(:all);

Алгоритмы интегратора

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my $O6=@{ $coor{"O6"} };
my $N1=@{ $coor{"N1"} };
my $N2=@{ $coor{"N2"} };
my $R=@{ $coor{"R"} };
my $sch, my $schnum;
my %qwa=find_quart( $coor{"O"} );
my $sqnum=keys %qwa;

```

В принципе, угадать будущие координаты не просто.

Ряд Тейлора: $r(t + \delta t) = r(t) + \delta t v(t) + \frac{1}{2} \delta t^2 a(t) + \frac{1}{6} \delta t^3 b(t) \dots$

```

my %qwa=find_quart( $coor{"O"} );
my $sqnum=keys %qwa;

```

```

if ($sqnum > 0){
#system("mkdir $ARGV[0]");
my $filename=$ARGV[0];

```

```

$filename=~ s/\^\.//;
$filename=~ s/\^\.pdb//;

```

Алгоритм Верле:

```

my $name=$ARGV[1];
my $name="O6";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $sqnum\n";

```

```

foreach my $m (sort {$a-<=>$b} keys %qartets){
my %qartets=%qwa; #find_quart($coor{$m});
my %q=find_q($coor{$m});

```

```

# foreach my $q { keys %qartets}{ print join " ",@{$qartets{$q}},"\n";

```

```

foreach my $q { keys %qartets){

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


```

```

my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

```

```

foreach my $res (@{ $qartets{$q} }){

```

```

# print "$q $coor{$m}";

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


```

```

$nx=$nx+ $coor{$m} {$res} {"N9"}->x;
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;

```

```

$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;

```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Интегратор leap-frog

```
#!/(my $coor,my $chnum)=read_pdb($ARGV[0]);
my $coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $chnum;
```

Leap-frog самый быстрый вариант алгоритма Верле:

```
foreach my $cont key %{$coor} { my $contsub=0; if ($cont ne $sch){$sch=$cont; $chnum=$chnum+1;}
```

```
my %qwa=find_quart($coor{"0"}); my $qnum=keys %qwa;
```

```
if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename-- s/^\.*/\//;
$filename-- s/\.pdb//;
#$filename=$chnum." ".$qnum." ".$filename.".dat";
$filename="dir".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";
```

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

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

Тогда скорость в момент t:

```
foreach my $m (sort {$a=<=>$b} keys %coor){
my $sq=find_quart($m);
```

```
# foreach my $q { keys %qartets} { print join " ",@{$qartets{$q}},"n";
```

```
foreach my $q { keys %qartets}{
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res (@{ $qartets{$q} }){
```

```
# print "$q $coor{$m}{$res} {"R"}->x,"n";
```

```
$nx=$nx+ $coor{$m}{$res} {"N9"}->x;
```

```
$ny=$ny+ $coor{$m}{$res} {"N9"}->y;
```

```
$nz=$nz+ $coor{$m}{$res} {"N9"}->z;
```

```
$ox=$ox+ $coor{$m}{$res} {"O6"}->x;
```

```
$oy=$oy+ $coor{$m}{$res} {"O6"}->y;
```

```
$oz=$oz+ $coor{$m}{$res} {"O6"}->z;
```

$$v(t) = \frac{1}{2} \left[v(t + \frac{1}{2} \delta t) + v(t - \frac{1}{2} \delta t) \right]$$

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Периодические граничные условия

```
my ($coor,$sch,$num)=read_pdb($ARGV[0]);
```

```
my %coor=();
my $mdir=
my $sch, $f
foreach my $
```

```
my %qwa
```

```
if ($qnum)
#system(
my $filen
$filenam
$filenam
# $filenar
$filenam
print "$fif
open OUT
print OUT
```

```
foreach
my %c
my %c
```

```
# for
```

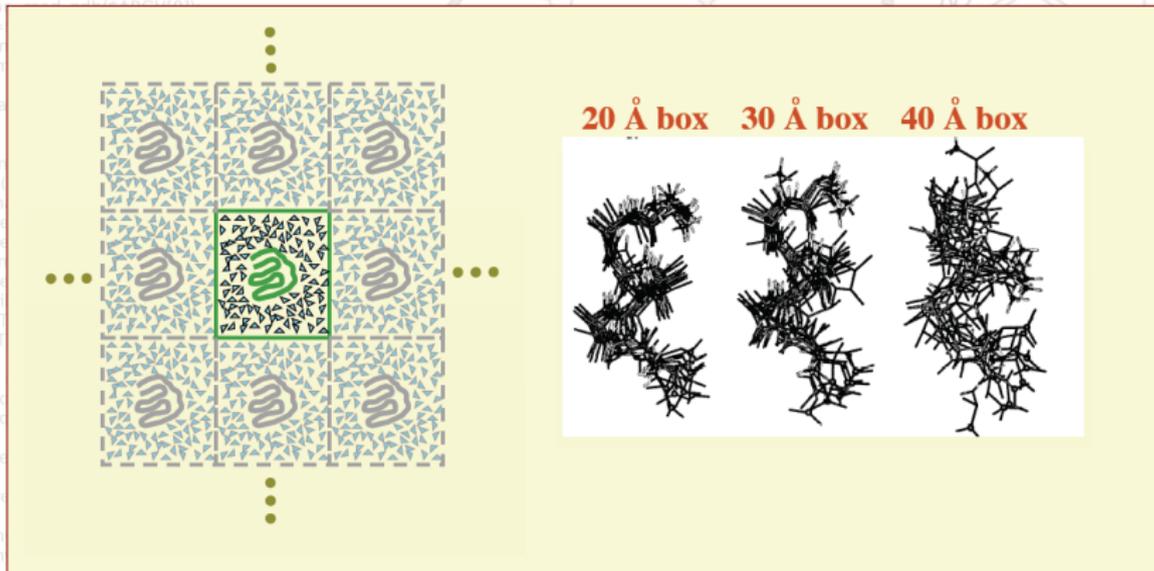
```
for
```

```
m
```

```
m
```

```
# print "x: ", $coor{$m}{$res}{"N9"}->x;
$nx=$nx+
$ny=$ny+ $coor{$m}{$res}{"N9"}->y;
$nz=$nz+ $coor{$m}{$res}{"N9"}->z;
```

```
$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```



МД поли-аланина показала искусственную стабилизацию альфа спирали, при использовании маленькой ячейки.

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Периодические граничные условия

```
my ($coor,$schnum)=read_pdb($ARGV[0]);
```

```
my $coor=
my $dir=$
my $sch, m
foreach my
```

```
my %qwa=
```

```
if ($qnum)
#system("
my $filena
$filename=
$filename)
#$filename
$filename=
$filename=
print "$file
open OUT,
print OUT,
```

```
foreach my
my %qa
my %qc
```

```
# forea
```

```
forea
```

```
my
```

```
my
```

```
for
```

```
#
```

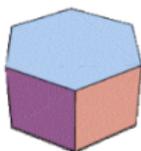
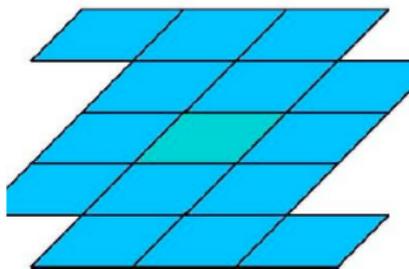
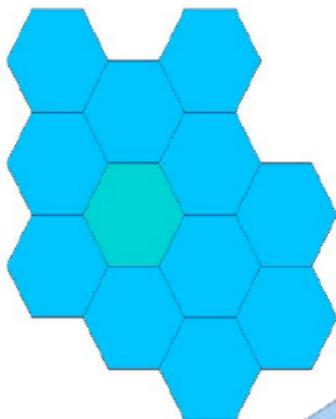
```
my $my
```

```
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;
```

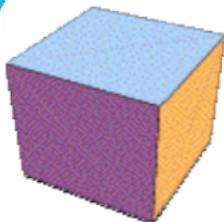
```
$sox=$sox+ $coor{$m} {$res} {"O6"}->x;
```

```
$soy=$soy+ $coor{$m} {$res} {"O6"}->y;
```

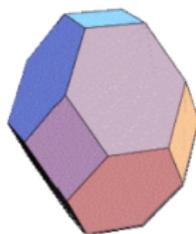
```
$soz=$soz+ $coor{$m} {$res} {"O6"}->z;
```



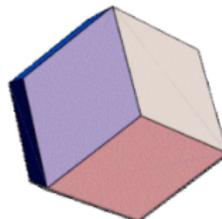
Hexagonal
Prism



Cube



Truncated
Octahedron



Rhombic
Dodecahedron

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Сферические граничные условия

```

#(my %$coor,my $schnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $schnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $sch ){ $schnum++; $sch=$ggg } ;

```

```
my %$qwa=find_quart( $coor{"0"} ); my $qnum=keys %qwa;
```

Бывают системы, для которых применение периодических граничных условий неудобно:

- Капли жидкости
- Ван-дер-Ваальсовы кластеры
- Гетерогенные системы при неравновесии
- Моделирование в вакууме

```

foreach my $q ( sort { $a->{ $qnum } lt $b->{ $qnum } } keys %qnum ){
my %$q=$coor{"$q"};
my %$q=find_q( $coor{"$q"} );

```

```

#
foreach my $q ( keys %qartets ){

```

```

my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

```

```

foreach my $res ( @{$qartets{$q}} ){
print "$q $coor{$m}{$res}{"$r"}->x,"$n";

```

```

$nx=$nx+ $coor{$m}{$res}{"$r"}->x;
$ny=$ny+ $coor{$m}{$res}{"$r"}->y;
$nz=$nz+ $coor{$m}{$res}{"$r"}->z;

```

```

$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;

```



```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Производительность

Система из 80000 атомов, компьютер Core2Quad:

Скорость:

24 шага/сек

Время наблюдения за системой:

1 мкс

Число шагов:

$5 \cdot 10^8$

Длина шага:

2 фс.

Время симуляции:

$5 \cdot 10^8 / 24$ сек

24000 часов

1000

суток

около 3 лет

Кластер (96 процессоров) примерно 25 дней, можно до 2000 процессоров.

```
foreach my $m (sort { $a-<=>$b } keys %coor) {
    my %qartets = %qwa; #find_quart( $coor{$m} );
    my %q = find_q( $coor{$m} );

    #
    foreach my $q ( keys %qartets ) { print join " ", @{$qartets{$q}}, "\n";
        foreach my $q ( keys %qartets ) {
            my $nx; my $ny; my $nz;
            my $ox; my $oy; my $oz;
            my $r;

            foreach my $res ( @{$qartets{$q}} ) {
                print join " ", $coor{$m} {$res} {"N9"}->x,
                    $res {"N9"}->y,
                    $nz=$nz+ $coor{$m} {$res} {"N9"}->z;

                $ox=$ox+ $coor{$m} {$res} {"O6"}->x;
                $oy=$oy+ $coor{$m} {$res} {"O6"}->y;
                $oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Ограничения быстрых колебаний

Частота колебаний C-H, N-H, O-H связей ограничивает временной шаг МД в 1 фс.



Начальные
координаты



Координаты после
одного шага МД

Shake - алгоритм



После применения
Shake.

```
$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Увеличение шага интегратора МД

```
my ($coor, my $schnum) = read_pdb($ARGV[0]);
```

- Можно присвоить атому водорода массу 2 а.е. При этом отняв 1 от тяжелого атома-соседа.

```
my %qwa = find_quart($coor{"O"}); my $qnum = keys %qwa;
```

- Использовать специальные конструкции. Dummies.

```
if ($qnum > 0){
```

```
my $dir = ($dir{"O"});
my $filename = "s/" . $dir . "/";
my $filename = "s/" . $dir . ".pdb/";
my $filename = "s/" . $dir . ".dat";
```

$$V = V(r_d, r_1, \dots, r_n) = V^*(r_1, \dots, r_n)$$

$$\mathbf{F}_i = -\frac{\partial V^*}{\partial \mathbf{r}_i} = -\frac{\partial V}{\partial \mathbf{r}_i} - \frac{\partial V}{\partial r_d} \frac{\partial r_d}{\partial \mathbf{r}_i} = \mathbf{F}_i^{\text{direct}} + \mathbf{F}_i'$$

```
foreach my $m (sort {$a<=>$b} keys %coor) {
  my %qartets = %qwa; #find_quart($coor{$m});
  my %q = find_q($coor{$m});
```

```
# foreach my $q ( keys %qartets ) { print join " ", $q, $qartets{$q}, "\n";
```

```
foreach my $q ( keys %qartets ){
```

```
  my $nx; my $ny; my $nz;
  my $ox; my $oy; my $oz;
  my $r;
```

```
  foreach my $res ( @{$qartets{$q}} ){
```

```
    # print "$q $coor{$m} {$res} {"$R"}->x, "\n";
    $nx = $nx + $coor{$m} {$res} {"$R"}->x;
    $ny = $ny + $coor{$m} {$res} {"$R"}->y;
    $nz = $nz + $coor{$m} {$res} {"$R"}->z;
```

```
    $ox = $ox + $coor{$m} {$res} {"O6"}->x;
```

```
    $oy = $oy + $coor{$m} {$res} {"O6"}->y;
```

```
    $oz = $oz + $coor{$m} {$res} {"O6"}->z;
```

$$\mathbf{F}_i = \begin{bmatrix} \frac{\partial x_s}{\partial x_i} & \frac{\partial y_s}{\partial x_i} & \frac{\partial z_s}{\partial x_i} \\ \frac{\partial x_s}{\partial y_i} & \frac{\partial y_s}{\partial y_i} & \frac{\partial z_s}{\partial y_i} \\ \frac{\partial x_s}{\partial z_i} & \frac{\partial y_s}{\partial z_i} & \frac{\partial z_s}{\partial z_i} \end{bmatrix} \mathbf{F}_d$$

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

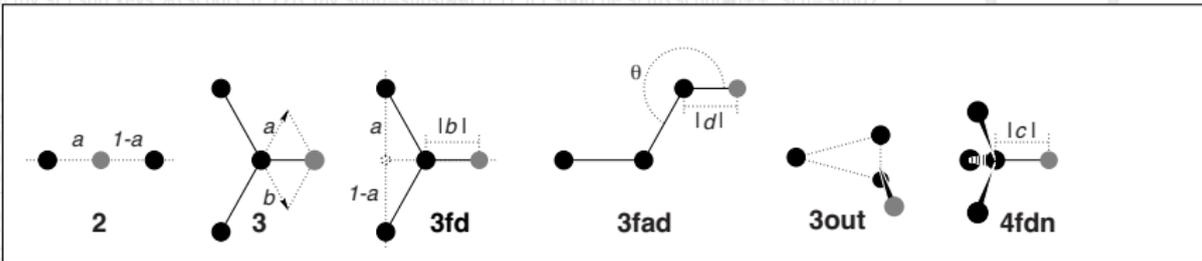
Конструкции атомов-пустышек в GROMACS

```
#!/(my %$coor,my $schnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $schnum;
```

```
foreach my $c ( sort keys %{$coor{"O"}} ){ my $coor=subst("%c",0,1); if ( $coor ne $sch ){ $sch=$coor; } }
```

```
my %c
```

```
if ($q
#system
my $fil
$filena
#filer
$filena
print
open C
print OUT "#INFO chain $schnum qnum $qnum\n";
```



```
foreach my $m ( sort { $a <=> $b } keys %coor ){
my %qartets = %qwa; #find quart( $coor{$m} );
my %q = find_q( $coor{$m} );
```

■ Атомы - пустышки

■ Реальные атомы, входящие в конструкцию

```
# foreach my $q ( keys %qartets ){
foreach my $q ( keys %qartets ){
```

```
my $nx; my $ny; my $nz;
```

Используя атомы-пустышки, можно увеличить шаг до 5-7 фс.

```
foreach my $res ( @({ $qartets{$q} }) ){
```

```
# print "$q $coor{$m} {$res} {"N"}->x,\"n";
$nx=$nx+ $coor{$m} {$res} {"N9"}->x;
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;
```

```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

Температура

При МД часто происходит релаксация структуры и появляется излишек кинетической энергии.

$$E_{NVT} = \frac{3}{2} N k_b T$$

Самый простой способ сохранить температуру — это масштабирование скоростей

$$\frac{\partial T}{\partial t} = \frac{T_0 - T}{\tau} \quad \lambda = \left[1 + \frac{n_{TC} \Delta t}{\tau_T} \left\{ \frac{T_0}{T(t - \frac{1}{2} \Delta t)} - 1 \right\} \right]^{1/2}$$

Кроме масштабирующих термостатов, существуют столкновительные термостаты и термостаты с дополнительной степенью свободы.

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Контроль давления в системе

```
#!/(my %$coor,my $chnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $chnum;
foreach my $r ( sort keys %{$coor{"O"}} ) { my $qng=substr($r,0,1); if ( $qng ne $sch ) { $chnum++; $sch=$qng } };
my %$qwa=find_quart($coor{"O"} ); my $qnum=keys %$qwa;
```

Баростат Берендсена

$$\frac{\partial P}{\partial t} = \frac{P_0 - P}{\tau_p},$$

```
if ($qnum > 0) {
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename="-- s/\ /_//";
```

Баростат Паринелло-Рахмана

$$\frac{\partial b^2}{\partial t^2} = \mathbf{VW}^{-1} \mathbf{b}'^{-1} (P - P_{\text{ref}}).$$

```
$filename="$mdir/$filename.dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";
foreach my $m ( sort {$a<=>$b} keys %$coor ) {
my %$qartets= %$qwa; #find_quart($coor{"O"});
my %$q=find_q($coor{$m});
```

\mathbf{b} - матрица ветров ячейки

V - объём

W - матрица параметров определяющих силу сопряжения

```
foreach my $q ( keys %$qartets ) {
foreach my $g ( keys %$qartets ) {
my $m; my $nz;
my $ox=my $oy; my $oz;
foreach my $res ( @{$ $qartets{$g} } ) {
print "$q $coor{$m} {$res} {"$r"}->x,\n";
$nx=$nx+ $coor{$m} {$res} {"$r"}->x;
$ny=$ny+ $coor{$m} {$res} {"$r"}->y;
$nz=$nz+ $coor{$m} {$res} {"$r"}->z;
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Методология подготовки системы для МД

```
my (%my %coor, my $snum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
```

Построение топологии молекулы на основе координат т.е. перечисление связей углов и тд.

```
if ($snum > 0){
```

Выбор формы и размера ячейки

```
my ($x1, $y1, $z1, $x2, $y2, $z2) = ($coor{"N1"}, $coor{"N2"}, $coor{"N3"}, $coor{"N4"}, $coor{"N5"}, $coor{"N6"});
```

Минимизация энергии структуры в вакууме методы: steep, CG, l-bfgs

```
my ($x1, $y1, $z1, $x2, $y2, $z2) = ($coor{"N1"}, $coor{"N2"}, $coor{"N3"}, $coor{"N4"}, $coor{"N5"}, $coor{"N6"});
```

Добавление растворителя и ионов в ячейку

```
my $nx; my $ny; my $nz;
```

"Утряска" воды и ионов вокруг неподвижной молекулы

```
#
print "$q $coor{$m} {$res} {"$R"}->x, "\n";
$nx=$nx+ $coor{$m} {$res} {"$R"}->x;
$ny=$ny+ $coor{$m} {$res} {"$R"}->y;
$nz=$nz+ $coor{$m} {$res} {"$R"}->z;

$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```

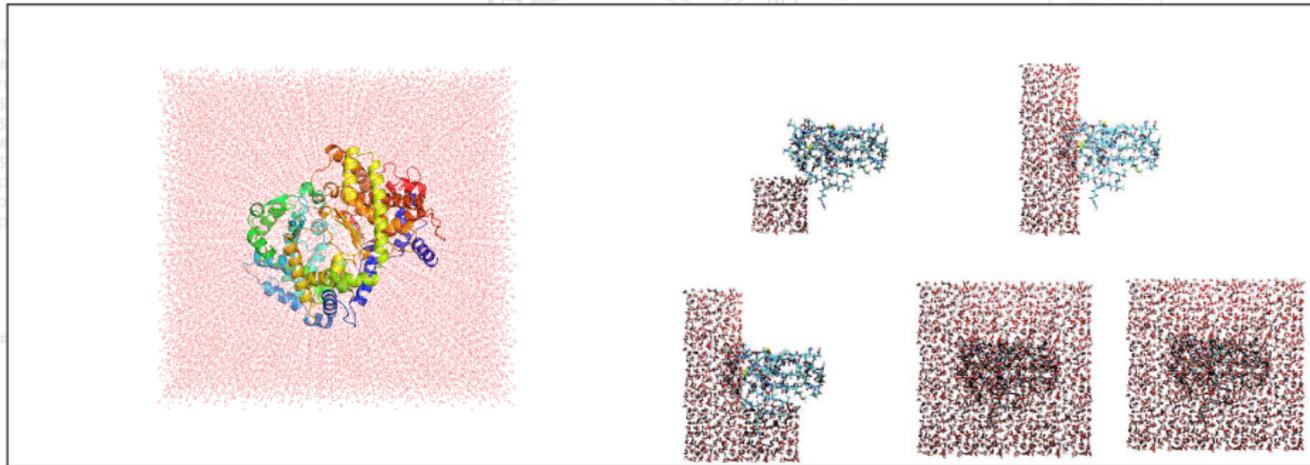
```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Добавление воды в ячейку

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $chnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $sch ){ $chnum++; $sch=$ggg } };
my %qwa=find_quart( $coor{"0"} ); my $qnum=keys %qwa;

```



```

foreach my $res (@{ $qartets{$q} ){
#
    print "$q $coor{$m}{$res} {"N7"}->x,"n";
    $nx=$nx+ $coor{$m}{$res} {"N9"}->x;
    $ny=$ny+ $coor{$m}{$res} {"N9"}->y;
    $nz=$nz+ $coor{$m}{$res} {"N9"}->z;

    $ox=$ox+ $coor{$m}{$res} {"O6"}->x;
    $oy=$oy+ $coor{$m}{$res} {"O6"}->y;
    $oz=$oz+ $coor{$m}{$res} {"O6"}->z;

```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Молекулярная динамика и неявный растворитель

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor{"O"}}){ my $ggg=substr($r,0,1); if ( $ggg ne $ch){$chnum++; $ch=$ggg} };

my %qwa=find_quart( $coor{"O"} ); my $qnum=keys %qwa;

```

- Часто явно заданный растворитель, т. е. молекулы воды заменяют потенциалами.
- Методы на основе поверхности доступной растворителю.

```

if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$dir."/";
$filename=$filename.$ch.".pdb";
#system("cp $ARGV[0] $filename");
$filename=$dir."/";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";

```

```

foreach my $m (sort {$a<=>$b} keys %coor){
my %qartets = %qwa ; #find_quart( $coor{$m} );
my %q = find_q( $coor{$m} );

```

```

# foreach my $q { keys %qartets } { print join " ", @{$qartets{$q}} , "\n";
fore

```

```

my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

```

```

foreach my $res ( @{$qartets{$q}} ){
#
print "$q $coor{$m}{$res} {"$res"}->x,\n";
$nx=$nx+ $coor{$m}{$res}{"N9"}->x;
$ny=$ny+ $coor{$m}{$res}{"N9"}->y;
$nz=$nz+ $coor{$m}{$res}{"N9"}->z;

$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;

```

$$\Delta G_{solv} = \sum_i \sigma_i ASA_i$$

- Метод Пуассона-Больцмана и его производные

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Молекулярная динамика и неявный растворитель

```
my ($coor,$snum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
```

Метод Пуассона-Больцмана точен, но технически очень сложен для счёта.

$$\nabla \cdot \left[\epsilon(\vec{r}) \nabla \Psi(\vec{r}) \right] = -4\pi \rho^f(\vec{r}) - 4\pi \sum_i c_i^\infty z_i q \lambda(\vec{r}) e^{\frac{-z_i q \Psi(\vec{r})}{kT}}$$

Generalized Born и GBSA

$$G_s = \frac{1}{8\pi} \left(\frac{1}{\epsilon_0} - \frac{1}{\epsilon} \right) \sum_{i,j} \frac{q_i q_j}{f_{GB}}$$

$$f_{GB} = \sqrt{r_{ij}^2 + a_{ij}^2} e^{-D}$$

$$D = \left(\frac{r_{ij}}{2a_{ij}} \right)^2, a_{ij} = \sqrt{a_i a_j}$$

```
foreach my $q ( keys %qartets ){
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

foreach my $res (@{ $qartets{$q} ){
print "$q $coor{$sm} {$res} {"$R"}->x,"$n";
$nx=$nx+ $coor{$sm} {$res} {"$R"}->x;
$ny=$ny+ $coor{$sm} {$res} {"$R"}->y;
$nz=$nz+ $coor{$sm} {$res} {"$R"}->z;

$ox=$ox+ $coor{$sm} {$res} {"O6"}->x;
$oy=$oy+ $coor{$sm} {$res} {"O6"}->y;
$oz=$oz+ $coor{$sm} {$res} {"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Молекулярная динамика и неявный растворитель

```

#(my %$coor,$my $schnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch,$my $schnum;
foreach my $f ( sort keys %{$coor{"O"} } ) { if ( $f =~ m/^(O\d+)$/ ) { $sch=$f; } }

```

Основные недостатки неявного растворителя:

- РВ и GB учитывают в основном электростатическую составляющую.
- Гидрофобный эффект не учитывается.
- Вязкость как результат столкновений и скоростей не рассчитывается и не учитывается.
- Водородные связи воды с объектом интереса не могут быть учтены.
- Исчезает возможность учёта водных мостиков.

```

if ($qnum) {
#system("mkdir $ARGV[1]");
my $filename=$ARGV[1];
$filename=~ s/\ .pdb//;
#system("cp $ARGV[0] $filename");
print "$filename\n";
open OUT,">$filename";
print OUT "$qnum\n";
foreach my $m ( keys %$coor ) {
my %$q= find q( $coor{$m} );
# foreach my $s ( keys %$q ) {
foreach my $sres ( @{$ $qartets{$s} } ) {
print "$s $coor{$m} {$sres} {" "N" }->x,"n";
$nx=$nx+ $coor{$m} {$sres} {" "N" }->x;
$ny=$ny+ $coor{$m} {$sres} {" "N" }->y;
$nz=$nz+ $coor{$m} {$sres} {" "N" }->z;
$ox=$ox+ $coor{$m} {$sres} {" "O" }->x;
$oy=$oy+ $coor{$m} {$sres} {" "O" }->y;
$oz=$oz+ $coor{$m} {$sres} {" "O" }->z;
}
}
}

```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Конформации в молекулярной динамике

```
my ($coor,$schnum)=read_pdb($ARGV[0]);
my $coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch_my_schnum;
```

Обычная молекулярная динамика это тепловое движение молекулы.

- Основной компонент такого движения это высокочастотные гармонические колебания атомов.
- Интерес представляют низкочастотные движения больших частей молекулы.
- Экстракция таких движений проводят с помощью Фурье преобразований.
- Низкочастотные колебания молекулы называют основной или существенной динамикой

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Моделирование амфифильных молекул

```

#(my %$coor,my $chnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $chnum;
foreach my $c ( sort keys %{$coor{"O"}} ) { if ( $c =~ /^O/ ) { $chnum++; $sch=$c; }
}

```

Амфифильные молекулы: фосфолипиды, гликолипиды и т.д.

- Ключевая особенность - образование разных фаз
- Возможно образование разных жидко-кристаллических фаз с выраженным порядком вдоль плоскости фазы.
- Параметр порядка
- Нередко используют контроль площади поверхности через поверхностное натяжение.
- К большим системам применяют крупно-зернистое моделирование.

```

if ($qnum) {
#system("mkdir $ARGV[1]");
my $filename=$sch.$chnum;
$filename=~s/\s//g;
print "$filename\n";
open OUT, ">$filename";
print OUT "$c\n";
}

```

```

foreach my $m ( keys %{$coor{"O"}} ) {
my %$q= find_neighbors($m);
# foreach my $n ( keys %{$q} ) {

```

```

my $nx, my $ny, my $nz;
my $ox, my $oy, my $oz;

```

```

foreach my $res ( @{$qartets{$q}} ) {
print "$q $coor{$m}{$res} {"$res"}->x,"n";
$nx=$nx+ $coor{$m}{$res}{"$res"}->x;
$ny=$ny+ $coor{$m}{$res}{"$res"}->y;
$nz=$nz+ $coor{$m}{$res}{"$res"}->z;

$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
}

```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Гибридное QM/MM моделирование

```
my ($coor,$schnum)=read_pdb($ARGV[0]);
```

Основная идея: разделить большую систему на квантовую и молекулярную части.

```
my $qcoor=$schnum;
my $dli=$schnum;
my $sch,$schnum;
foreach my $atom ($atom) { $sch=$sch+$atom; $schnum++; $sch=$sch+$atom; }
```

- Электростатическое окружение из MM части чувствуется QM частью.

- MM часть принимает силы из QM части и соответственно адаптируется.

```
if ($qnum > 0) {
#system("
my $filename="
$filename="--s/ / /";
$filename="--s/ / /";
$filename="--s/ / /";
$filename="--s/ / /";
print "$filename";
open OUT, ">";
print OUT "#INFO chain $schnum qnum $qnum \n";
```

```
foreach my $m (sort { $a-<=>$b } keys %coor) {
my %qartets = %qwa ; #find_quart( $coor{ $m } );
my %q = find_q( $coor{ $m } );
```

```
# foreach my $q { keys %qartets } { print join " ",
```

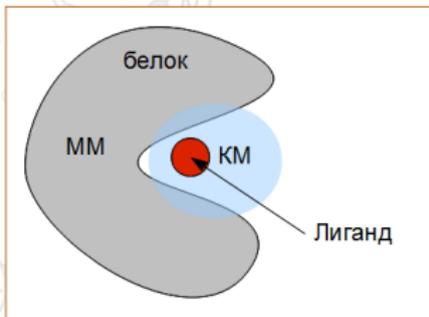
```
foreach my $q { keys %qartets } {
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @ { $qartets{ $q } } ) {
```

```
# print "$q $coor{ $m } { $res } { "N9" }->x;
$nx=$nx+ $coor{ $m } { $res } { "N9" }->x;
$ny=$ny+ $coor{ $m } { $res } { "N9" }->y;
$nz=$nz+ $coor{ $m } { $res } { "N9" }->z;
```

```
$ox=$ox+ $coor{ $m } { $res } { "O6" }->x;
$oy=$oy+ $coor{ $m } { $res } { "O6" }->y;
$oz=$oz+ $coor{ $m } { $res } { "O6" }->z;
```



```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Гибридное QM/MM моделирование

```
#!/(my %coor,my $chnum)=read_pdb($ARGV[0]);
```

```
my %coor=read_pdb($ARGV[0]);
```

Простейший Гамильтониан для QM/MM системы:

```
foreach my $r ( sort keys %{$coor{"O"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $ch ){ $chnum++; $ch=$ggg };
```

```
my %qwa=find_quad_coor{"O"}; my $nucl=0; my %qwa;
```

$$H = -\frac{1}{2} \sum_i \nabla^2 + \sum_i \sum_j \frac{1}{r_{ij}} + \sum_i \sum_j \frac{Z_i Z_j}{R_{ij}} - \sum_i \sum_j \frac{Q_j}{R_{ij}} + \sum_i \sum_j \frac{Z_i Q_j}{R_{ij}}$$

```
my $filename=$ARGV[0];
my $filename=$ARGV[0];
$filename=~ s/\^\.//g;
$filename=~ s/\^\.//g;
```

К QM/MM части можно добавить и VdW составляющую:

```
print "chain $chnum\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";
```

$$H_{QM/MM} = - \sum_i \sum_j \frac{Q_j}{R_{ij}} + \sum_i \sum_j \frac{Z_i Q_j}{R_{ij}} +$$

```
# foreach my $q ( keys %qartets ){ print join " ",@{$qartets{"q"}}, "\n";
```

```
foreach my $q ( keys %qartets ){
```

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

$$\sum_i \sum_j 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$

```
# print "$q $coor{$m}{$res}{"N"}->x,\n";
```

```
$nx=$nx+ $coor{$m}{$res}{"N9"}->x;
$ny=$ny+ $coor{$m}{$res}{"N9"}->y;
$nz=$nz+ $coor{$m}{$res}{"N9"}->z;
```

```
$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Гибридное QM/MM моделирование

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $chnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $sch ){ $chnum++; $sch=$ggg } };

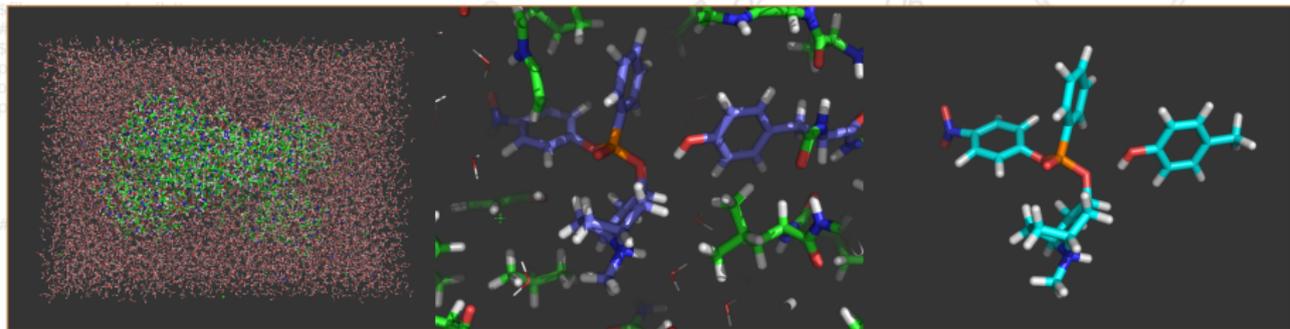
```

```
my %qwa=find_quart( %coor{"0"} ); my $qnum=keys %qwa;
```

- АТОМЫ СВЯЗКИ

```
if ($qnum > 0){
#system("cat $mdir/$qnum.pdb > $mdir/$qnum.pdb");
my $filename=$mdir.$qnum.pdb;
$filename="-- s/ / /";
```

- Специальные орбитали



```
my $r;
```

```
foreach my $res ( @{$ $quartets{$q} } ){
```

```

# print "$q $coor{$m}{$res}{"N"}->x,"n";
$nx=$nx+ $coor{$m}{$res}{"N9"}->x;
$ny=$ny+ $coor{$m}{$res}{"N9"}->y;
$nz=$nz+ $coor{$m}{$res}{"N9"}->z;
```

```

$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Длина траектории МД

```
#!/(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor{"0"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $ch ){ $chnum++; $ch=$ggg } ;
```

```
my %qw
```

```
if ($qu
```

```
#system
```

```
my $file
```

```
$filename
```

```
#$filename
```

```
$filename
```

```
print "$
```

```
open OU
```

```
print OU
```

```
foreach
```

```
my %
```

```
my %
```

```
#
```

```
fo
```

```
fo
```

```
r
```

```
r
```

```
my $r.
```

```
#
```

```
fo
```

```
my $res
```

```
print "$q
```

```
$nx=$nx+
```

```
$ny=$ny+
```

```
$nz=$nz+
```

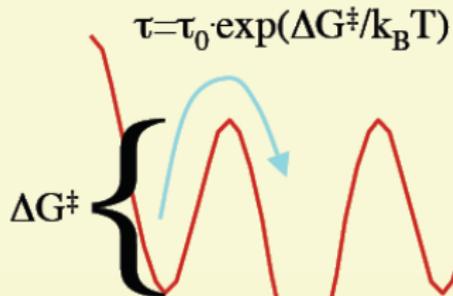
```
$ox=$ox+
```

```
$oy=$oy+
```

```
$oz=$oz+
```

```
my $r.
```

```
#
```



$$\tau_0 \sim 10^{-12}, \Delta G^\ddagger$$

1 kcal/mol: $\sim 1.2 \text{ ps}^{-1}$

5 kcal/mol: $\sim 1.5 \text{ ns}^{-1}$

10 kcal/mol: ms or longer!

```
foreach my $res ( @ { $partets{$q} } ){
```

```
print "$q $coor{$m}{$res}{\"N\"}->x,\"n\";
```

```
$nx=$nx+ $coor{$m}{$res}{\"N9\"}->x;
```

```
$ny=$ny+ $coor{$m}{$res}{\"N9\"}->y;
```

```
$nz=$nz+ $coor{$m}{$res}{\"N9\"}->z;
```

```
$ox=$ox+ $coor{$m}{$res}{\"O6\"}->x;
```

```
$oy=$oy+ $coor{$m}{$res}{\"O6\"}->y;
```

```
$oz=$oz+ $coor{$m}{$res}{\"O6\"}->z;
```

```
my $r.
```

```
#
```

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

Метод обмена репликами (REMD)

```

#(my %$coor,my $chnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $ch_ my %$chnum;
foreach my $m ($mdir) {
```

- Основная идея: запустить параллельно несколько счётов с разными температурами.

```
my %$qwa
```

- Если разница между температурами не большая, то гистограммы распределения энергии должны пересекаться.

```

if ($qnum) {
#system(
my $filename=$ARGV[0];
$filename=$ARGV[1];
$filename=$ARGV[2];
# $filename=$chnum."/".$qnum."/".$filename.".dat";
$filename=$chnum."/".$qnum."/".$filename.".dat";
print "file: ".$filename;
open OUT,">".$filename;
print OUT
```

- Мы можем выбрать правило когда производить обмен конформациями.

```
foreach my $m (sort { $a-<=>$b } keys %$coor) {
```

- Если мы проводим обмен когда потенциальная энергия одной из реплик ниже чем других, то это похоже на моделирование отжига.

```
my %$q
```

```
my %$q
```

```
# foreach my $m ($mdir) {
foreach my $m ($mdir) {
```

```
my $nx; my $ny; my $nz;
```

```
my $m
```

```
my $m
```

```
foreach my $m ($mdir) {
```

```
#
```

```
$nx=$nx+ $coor{$m}{ $res{"N9"}->x;
```

```
$ny=$ny+ $coor{$m}{ $res{"N9"}->y;
```

```
$nz=$nz+ $coor{$m}{ $res{"N9"}->z;
```

```
$ox=$ox+ $coor{$m}{ $res{"O6"}->x;
```

```
$oy=$oy+ $coor{$m}{ $res{"O6"}->y;
```

```
$oz=$oz+ $coor{$m}{ $res{"O6"}->z;
```

Вопросы?

```

#!/usr/bin/perl
use Math::VectorReal qw( :all );

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor{"0"}}){ my $ggg=substr($r,0,1); if ( $ggg ne $ch){$chnum++; $ch=$ggg} };

my %qwa=find_quart( %coor{"0"} ); my $qnum=keys %qwa;

if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/\^.*\///;
$filename=~ s/\.pdb//;
#$filename=$chnum."_"$qnum."_"$filename.".dat";
$filename="$dir"$.filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum \n";

foreach my $m (sort {$a<=>$b} keys %coor){
my %qartets = %qwa; #find_quart( %coor{$m} );
my %q = find_q( %coor{$m} );

# foreach my $q ( keys %qartets){ print join " ",@{$qartets{$q}}," \n";

foreach my $q ( keys %qartets){

my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

foreach my $res (@{ $qartets{$q} }){
# print "$q $coor{$m}{$res}{"$res"}->x," \n";
$nx=$nx+ $coor{$m}{$res}{"$res"}->x;
$ny=$ny+ $coor{$m}{$res}{"$res"}->y;
$nz=$nz+ $coor{$m}{$res}{"$res"}->z;

$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;

```