

```
#!/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} ) {
```

```
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
```

```
# 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 }{ "N9" }->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;
```

```
}
```

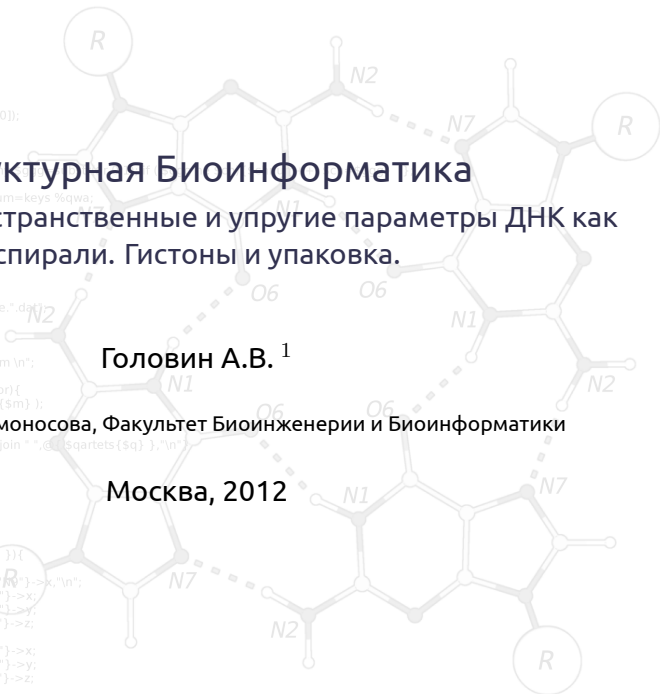
# Структурная Биоинформатика

## Лекция 11. Пространственные и упругие параметры ДНК как спирали. Гистоны и упаковка.

Головин А.В. <sup>1</sup>

<sup>1</sup>МГУ им М.В. Ломоносова, Факультет Биоинженерии и Биоинформатики

Москва, 2012



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

# Содержание

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

## Введение

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

## FJC DNA model

```
my $filename=$ARGV[0];
my $filename=$ARGV[0];
my $filename=~ s/^\.//;
my $filename=~ s/\.pdb//;
my $filename="$mdir"/"$qnum"."$filename".dat;
```

## Rod DNA model

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

## WLC DNA model

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

## Мезомоделирование ДНК

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

## Хроматин

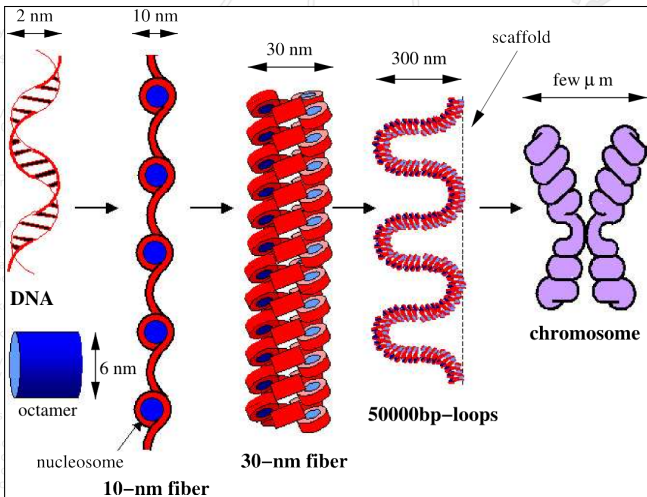
```
( @ { $qartets { $q } } ) {
# print "$q $coor { $m } { $res } { "N7" }->x, "\n";
$nx=$nx+ $coor { $m } { $res } { "N7" }->x;
$ny=$ny+ $coor { $m } { $res } { "N7" }->y;
$nz=$nz+ $coor { $m } { $res } { "N7" }->z;

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



# ДНК не только...

## Путь организации ДНК от молекулы до хромосомы:



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

# Основные модели для описания гибкости ДНК

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

Есть несколько основных подходов для описания гибкости ДНК, не все из них учитывают разницу между dsDNA и ssDNA.

```
my %qwa=find_quart( $coor{"0"} ); my $schnum=keys %qwa;
if ($schnum > 0){
my $filename=$ARGV[0];
$filename="-- s/^.*\//";
$filename="-- s/\.pdb//";
#$filename=$schnum." ".$schnum." ".$filename." .dat";
$filename="$dir/".$filename." .dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO: chain $schnum group $schnum\n";
foreach my $m ( sort keys %{$coor{"0"}} ){
my %qartets = %qwa ; #find quart( $coor{$m} );
my $q = $qartets{"0"} ;
# foreach my $q ( keys %qartets ){ print join " ", @{$qartets{$q}} , "\n";
#
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} {$res} {"$R"}->x;
$oy=$oy+ $coor{$m} {$res} {"$R"}->y;
$oz=$oz+ $coor{$m} {$res} {"$R"}->z;
```

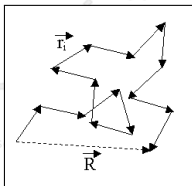
- Freely Jointed Chain, сегменты соединены шарнирами
- Rod model, представление ДНК как стержня.
- Worm Like Chain, изотропный стержень гибкий по всей длине.

# Freely Jointed Chain

Имеется полимер из  $N$  мономеров, тогда не упакованная длина

$$L = Nl$$

- сегменты между собой не взаимодействуют
- полимер флуктуирует и его форма определяется простым распределением
- Если полимер образует глобулу:



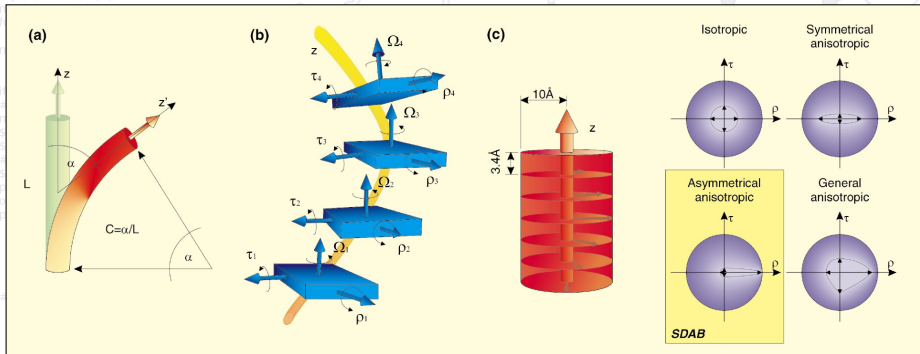
$$\langle R^2 \rangle = Nl^2 = Ll; \quad P(\vec{R}) = \left( \frac{3}{2\pi Nl^2} \right)^{3/2} e^{-\frac{3\vec{R}^2}{2Nl^2}}$$

$$\sqrt{\langle R^2 \rangle} = \sqrt{N} l = \sqrt{Ll}$$

# Rod model

ДНК представляется как стержень:

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my ($R) = @ARGV;
my ($L) = @ARGV;
my ($alpha) = @ARGV;
my ($coor) = @ARGV;
my ($sdir) = @ARGV;
my ($snum) = @ARGV;
my ($sname) = @ARGV;
my ($sfile) = @ARGV;
my ($sdir) = @ARGV;
my ($snum) = @ARGV;
my ($sname) = @ARGV;
my ($sfile) = @ARGV;
```



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

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

$$\Delta G = \frac{1}{2} EIL\alpha^2$$

$E$  - коэффициент Юнга

$I$  - момент инерции, для цилиндра радиусом  $r$ ,  $I = \frac{\pi r^4}{4}$

## Rod model, последовательность контекста

Table I. Sequence-dependent Young's modulus values calculated for various bending models<sup>26,34,40</sup>

Trinucleotide	Relative bendability (arbitrary units)		Young's modulus (10 <sup>9</sup> N/m <sup>2</sup> )	
	DNaseI scale	Consensus scale	DNaseI scale	Consensus scale
AAA/ TTT	0.1	0.05	2.307	2.23
AAC/ GTT	1.6	2.65	2.016	1.71
AAG/ CTT	4.2	4.70	1.523	1.30
AAT/ ATT	0.0	0.35	2.327	2.17
ACA/ TGT	5.8	5.50	1.225	1.15
ACC/ GGT	5.2	5.30	1.336	1.18
ACG/ CGT	5.2	5.30	1.336	1.18
ACT/ AGT	2.0	7.80	1.94	1.46
AGA/ TCT	6.5	4.90	1.096	1.26
AGC/ GCT	6.3	6.90	1.133	0.87
AGG/ CCT	4.7	5.05	1.429	1.23
ATA/ TAT	9.7	6.25	0.519	0.99
ATC/ GAT	3.6	4.45	1.636	1.35
ATG/ CAT	8.7	7.70	0.697	0.71
CAA/ TTG	6.2	4.75	1.151	1.29
CAC/ GTG	6.8	6.65	1.041	0.92
CAG/ CTG	9.6	6.90	0.536	0.87
CCA/ TGG	0.7	3.05	2.19	1.63
CCC/ GGG	5.7	5.85	1.244	1.08
CCG/ CGG	3.0	3.85	1.749	1.47
CGA/ TCG	5.8	7.05	1.225	0.84
CGC/ GCG	4.3	5.90	1.504	1.07
CTA/ TAG	7.8	5.00	0.859	1.24
CTC/ GAG	6.6	6.00	1.078	1.05
GAA/ TTC	5.1	4.05	1.355	1.43
GAC/ GTC	5.6	5.50	1.262	1.14
GCA/ TGC	7.5	6.75	0.914	0.90
GCC/ GGC	8.2	9.10	0.787	0.45
GGA/ TCC	6.2	5.00	1.151	1.24
GTA/ TAC	6.4	5.05	1.115	1.23
TAA/ TTA	7.3	4.65	0.95	1.31
TCA/ TGA	10.0	7.70	0.465	0.71

# Rod model, гибкость

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

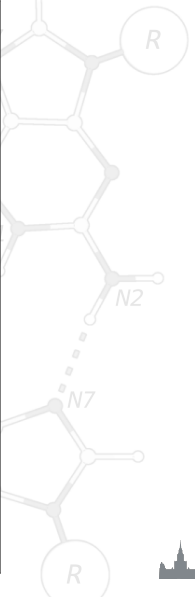
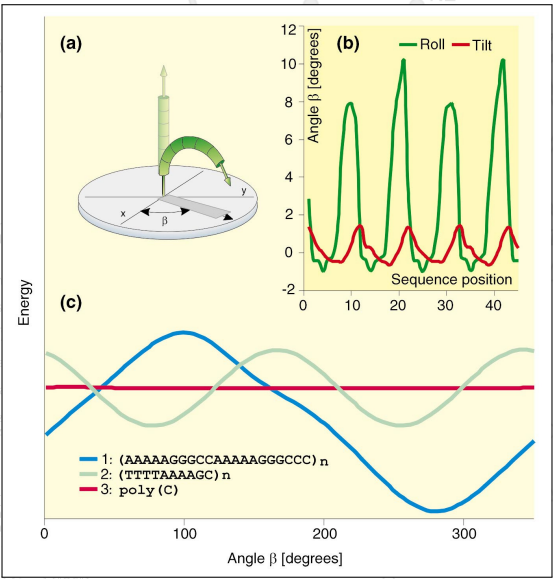
#(my %coor,my $chnum)=read
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
foreach my $r ( sort keys %{$coor} )
my %qwa=find_quart( %{$coor{$r}} );

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

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

# foreach my $q { keys %qartets }
foreach my $q { keys %qartets }
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

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





## Rod DNA model, биология

```

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

my $q = $ARGV[0];

#(my %$coor,my %$nnum)=read
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my %$nnum;
foreach my $r ( sort keys %{$coor} )
my %qwa=find_quart( %$coor{ $r }

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

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

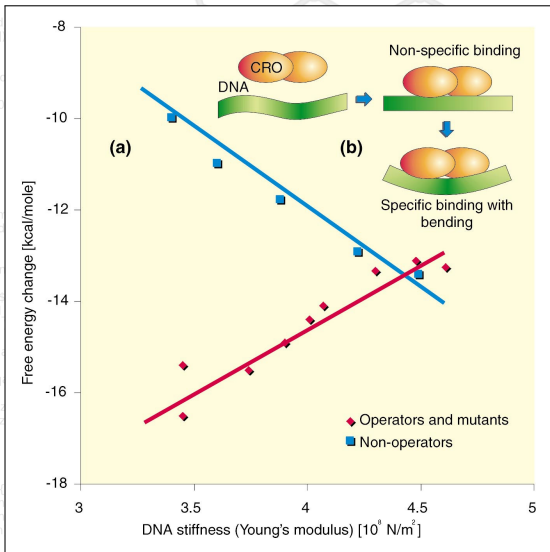
# foreach my $q { keys %q
foreach my $q { keys %q

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

foreach my $res ( @{
# print "$q $coor{
$nx=$nx+ %coor{$m}
$ny=$ny+ %coor{$m}
$nz=$nz+ %coor{$m}

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

```



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

# Worm Like Chain model

```

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

```

Суть модели это непрерывное описание цепи для решения ряда ограничений:

- Энтропийная эластичность ДНК цепи состоит из малых девиаций по оси молекулы из-за температуры
  - Направление цепи коррелирует с длиной цепи, называемой "the persistence length".
- Для ДНК в 10 mM растворе NaCl,  $P_{DNA} = 150$  п.о. или 550 nm.
- Силы порядка  $\frac{k_B T}{P}$  нужны для выравнивания и направления единиц эластичности вдоль оси полимера.

```

#
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 );
```

# Worm Like Chain model

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

Фиксированная величина длины контура,  $L$  и некая стандартная величина "сгибаемости",  $C$ .

```
foreach my $c ($cort keys %{$coor{"00"}}) { my $sgg=sub{0,1}; if ( $sgg ne $c) { $chnum+=$c, $c=$sgg } ;
```

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

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

```
# foreach my $q { keys %q
```

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

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

```
foreach my $res ( @{
```

```
# print "$q $coor{$m} {$res} {"N9"}->x,"n";
```

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

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

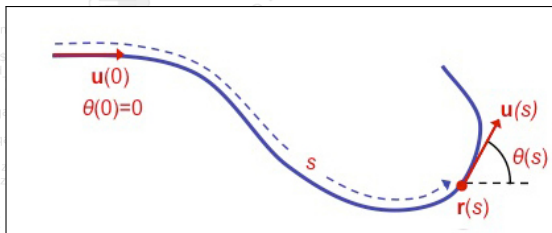
```
$nz=$nz+ $coor{$m} {$res} {"N9"}->z;
```

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

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

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

$$U = C \int_{s=0}^{s=L} \left| \frac{\partial t}{\partial s} \right|^2$$



# Подытожим:

```
#!/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 } ;
```

- Для сил действующих в диапазоне  $<100$  fN и  $>5$  pN модель FJC работает хорошо.

$$f = \frac{k_B T}{b} \frac{1}{1 - z/L}$$

- WLC модель работает во всех диапазонах.

```
my %qwa=find_quart( %coor{"0"} ); my $qnum=keys %qwa;
if ( $qnum > 0 ){
#system("ls -l $dir/$ch/$qnum/*");
my $filename=$ch.$qnum;
$filename-- s/^.*//;
$filename-- s/\.pdb//;
#$filename=$ch.$qnum.".".$qnum.".".$filename.".dat";
$filename=$dir.$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "chain $chnum group $qnum\n";
foreach my $m (sort { $a<=>$b } keys %coor){
my %qartets = %qwa; #find_quart( %coor{$m} );
my %q = find_q( %coor{$m} );
```

$$f = \frac{k_B T}{P} \left[ \frac{1}{4(1 - z/L)^2} - \frac{1}{4} + \frac{z}{L} \right]$$

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

# Моделирование ДНК на уровне пар оснований

Рассмотрим типы движений пар оснований друг относительно друга.

```
my %scoor=read_pdb($ARGV[0]);
my $snum=$scoor{1};
my $snum;
foreach my $r ( sort keys %{$scoor{"0"}} ){ my $sggg=substr($r,0,1); if ( $sggg ne $ch ){ $snum++; $ch=$sggg } };
my %qwa=find_quart( $scoor{"0"} ); my
```

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

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

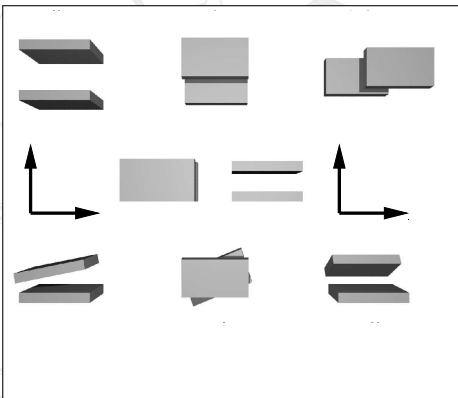
```
# foreach my $q { keys %qartets }
foreach my $q { keys %qartets }{
```

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

```
foreach my $res ( @{$qartets
```

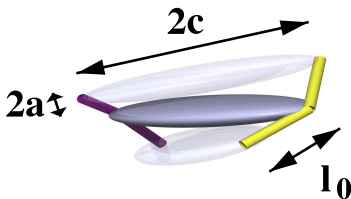
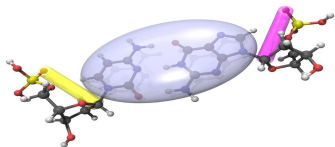
```
# print "$q $scoor{$m} {$
```

```
$nx=$nx+ $scoor{$m} {$res} {"N9"}->x;
$ny=$ny+ $scoor{$m} {$res} {"N9"}->y;
$nz=$nz+ $scoor{$m} {$res} {"N9"}->z;
$ox=$ox+ $scoor{$m} {$res} {"O6"}->x;
$oy=$oy+ $scoor{$m} {$res} {"O6"}->y;
$oz=$oz+ $scoor{$m} {$res} {"O6"}->z;
```



Rise (Ri), Slide (Sl), Shift (Sh), Twist (Tw), Roll (Ro), and Tilt (Ti)

# Потенциал



```

#(my %
my %coo
my $dir
my $sch
foreach
my %qw

if ($qnu
#system
my $file
$filenam
$filenam
#$filea
$filenam
$filenam
print "$
open OU
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

```

```

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";

```

```

$nz=$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;

```

$$U(A_1, A_2, r_{12}) = U_r \eta_{12} \chi_{12}$$

$$U_r = 4\epsilon \left( \left( \frac{\sigma}{h + \gamma\sigma} \right)^{12} - \left( \frac{\sigma}{h + \gamma\sigma} \right)^6 \right)$$

$\eta_{12}$  и  $\chi_{12}$  отвечают за силу взаимодействия в зависимости от ориентации эллипсов.

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

# Вид зависимостей

```
my (%coor,%my $coor=read_p
my $dir=$ARGV[1];
my $sch, my $chnu
foreach my $r ( sort
```

```
my %qwa=find_q
```

```
if ($qnum > 0){
#system("mkdir $A
my $filename=$A
$filename="-- s/\^
$filename="-- s/\.p
#$filename=$chnu
$filename="$dir"/
print "$filename\
open OUT,">$filen
print OUT "#INFO
```

```
foreach my $m (s
my %qartets=
my %q= find_q
```

```
# foreach my $
```

```
foreach my $
```

```
my $nx; my
my $ox; my
my $r;
```

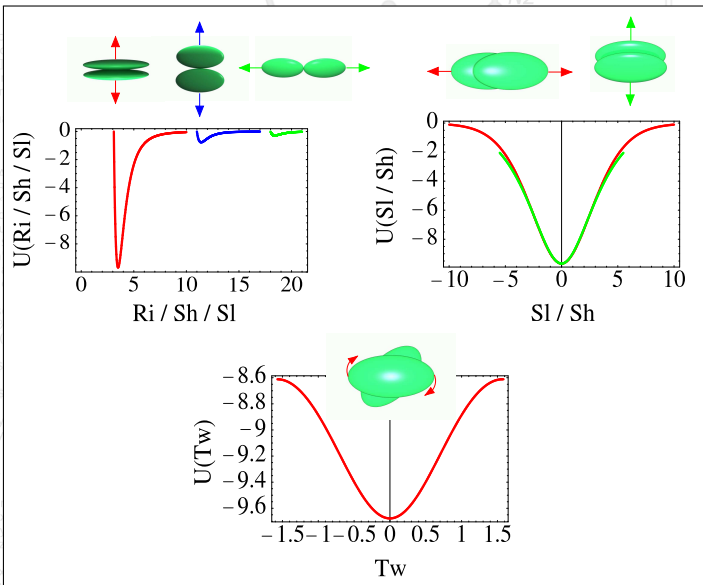
```
foreach my
```

```
# pri
```

```
$nx=$n
$ny=$n
$nz=$n
```

```
$ox=$o
$oy=$o
```

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



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

# Растягивание

```
my %coor;
my %coor=rea
my $dir=$ARG
my $sch, my $c
foreach my $r
my %qwa=find
```

```
if ($qnum > 0) {
#system("mko
my $filename=
$filename="-- $
$filename="-- $
#$filename=$
$filename="$d
print "$filenar
open OUT,">";$
print OUT "#IN
```

```
foreach my $
my %qartel
my %q=fin
```

```
# foreach r
```

```
foreach r
```

```
my $nx
my $ox
my $
```

```
foreach
```

```
#
```

```
$nx=$nx+ $coor($m) ($res) ("N9")->>x;
```

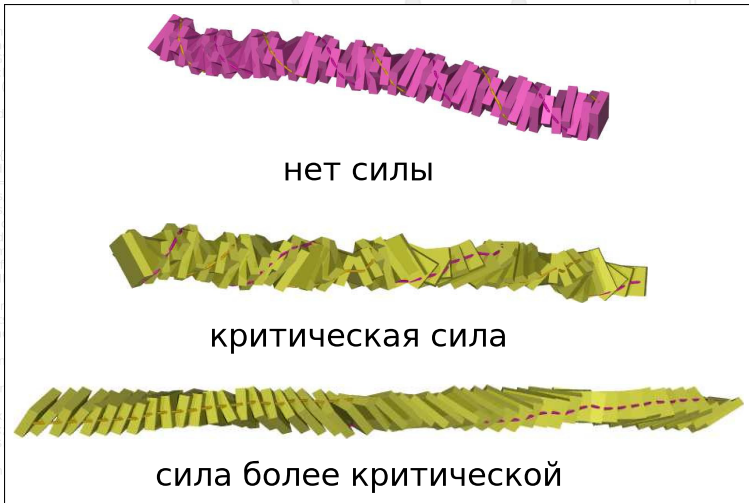
```
$ny=$ny+ $coor($m) ($res) ("N9")->>y;
```

```
$ox=$ox+ $coor($m) ($res) ("O1")->>x;
```

```
$oy=$oy+ $coor($m) ($res) ("O1")->>y;
```

```
$oz=$oz+ $coor($m) ($res) ("O1")->>z;
```

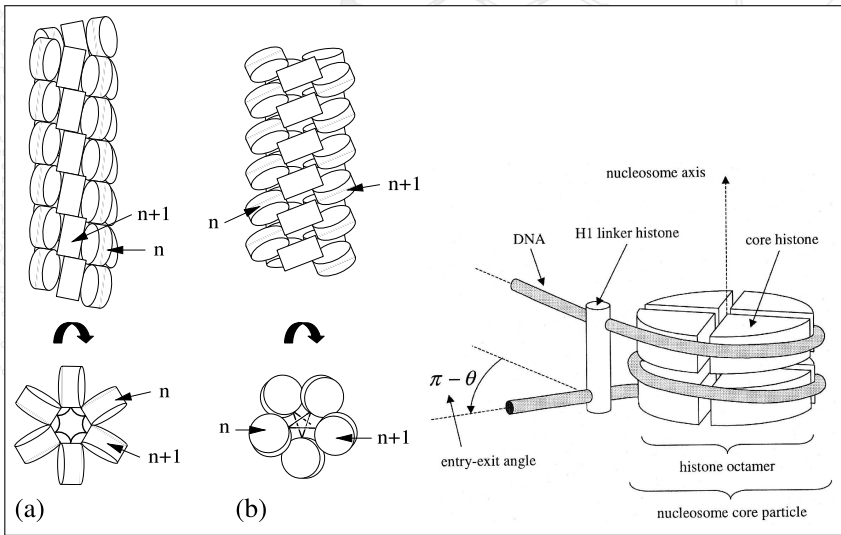
```
$oz=$oz+ $coor($m) ($res) ("O1")->>z;
```



Значения сил совпадают с экспериментальными. Локальные деформации реалистичны.



# Хроматин



```

$ox=$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 );
```

# Хроматин, модель блока

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

```
my %qw
```

```
if ($qur
#system
my $file
$filenam
$filenam
$filena
$filenam
print "$i
open OU
print OU
```

```
foreach
my %
my %
```

```
# for
```

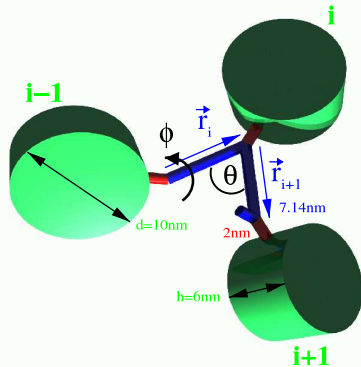
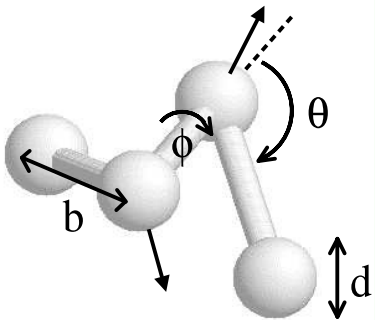
```
for
```

```
n
n
```

```
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;
```



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

# Хроматин, фазовое пространство

```
#!/my %coor,my $s
my %coor=read_p
my $dir=$ARGV[1]
my $sch, my $chnu
foreach my $r ( sor
my %qwa=find_qu
```

```
if ($qnum >0){
#system("mkdir $A
my $filename=$A
$filename-- s/\^
$filename-- s/\.p
#$filename=$chnu
$filename="$dir"/
print "$filename\
open OUT,">$filen
print OUT "#INFO
```

```
foreach my $m ($
my %qartets=
my %q= find_q
```

```
# foreach my $
```

```
foreach my $
```

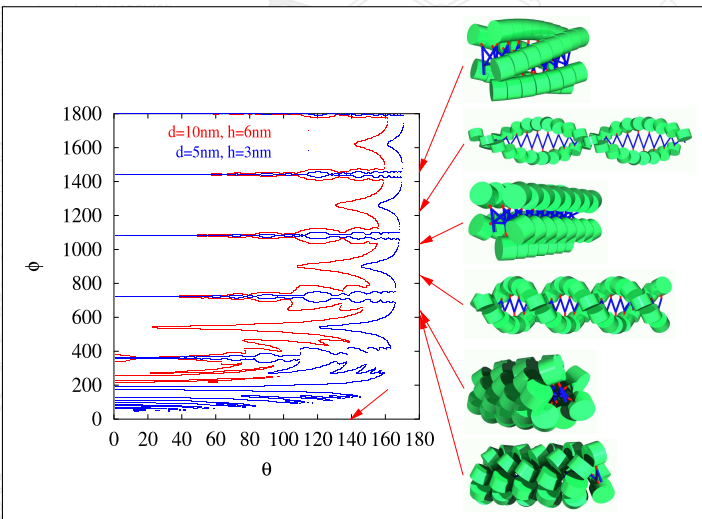
```
my $nx; my
my $ox; my
my $r;
```

```
foreach my
```

```
# pri
```

```
$nx=$n
$ny=$n
$nz=$n
```

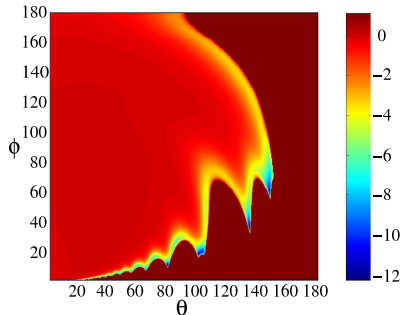
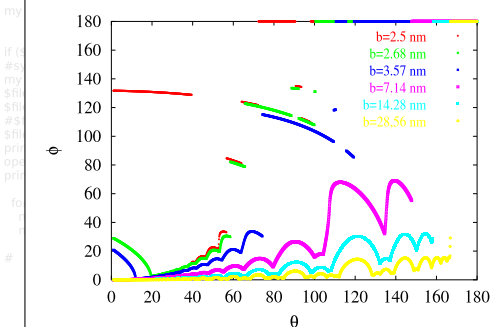
```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
$oy=$oy+ $coor{$m} {$res} {"O6"}->y;
$oz=$oz+ $coor{$m} {$res} {"O6"}->z;
```



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

# Хроматин, запрещённые зоны

```
#!/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 $qgq=subst( $r,0,1); if ( $qgq ne $ch ){ $chnum++; $ch=$qgq } }
```



```
foreach my $res ( @{$ $partets{$q} } ){
#
print "$q $coor{$m}{$res}{\"N9\"}->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 );
```

# Хроматин, растягивание

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

```
my %coor;
my $dd;
my $sch;
foreach
```

```
my %coor;
```

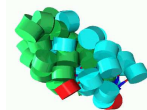
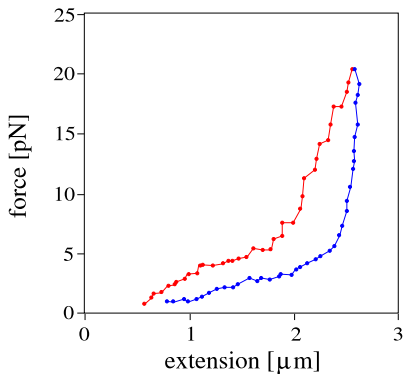
```
if ($qnu)
#system
my $file;
my $file;
my $file;
my $file;
print
open O
print O
```

```
foreach
my
my
```

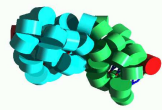
```
#
```

```
#
```

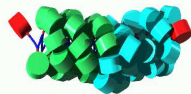
```
#
```



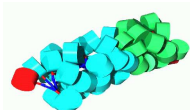
f=0pN



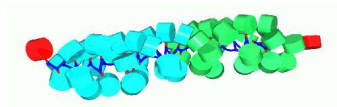
f=0.5pN



f=2.0pN



f=3.5pN



f=15pN

```
$nz=$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 $schnum)=read_pdb($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){ $schnum++; $sch=$ggg } };

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

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

```

