

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

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

```
my %qwa=find_quart( %coor{"0"}, my $qqq=substr($r,0,1); my $qqq=substr($r,0,1); my $qqq=substr($r,0,1);
```

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

Структура нуклеиновых кислот

Строение и организация

Головин А.В.¹

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

Москва, 2012

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

Содержание:

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

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

Введение

```
#!/usr/bin/perl
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^\s*//;
$filename=~ s/\s*\s*//;
$filename="$snum"."$filename".dat;
open OUT,">$filename";
print OUT "#INFO chain $snum qnum $qnum\n";
```

Химическое строение НК

```
#!/usr/bin/perl
my %qwa=find_quart( $coor{"0"} );
my %q= find_q( $coor{"0"} );
```

3D Структура ДНК

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

Структура РНК

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

Вторичная структура РНК

```
# 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 );
```

Нуклеиновые кислоты.

```

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

```

Нуклеиновые кислоты (НК) – это высокомолекулярные линейные полярные биополимеры, **полинуклеотиды**, которые построены из **нуклеотидных остатков**.

2 типа нуклеиновых кислот:

- дезоксирибонуклеиновая кислота – ДНК
- рибонуклеиновая кислота – РНК

```

#
foreach my $res ( @{$ $qartets{$s} } ){
    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::Vector;
use Math::Fig;
use Graphics;

```

Расшифровка аббревиатуры ДНК:

```
# { $qnum >0 }  
#system("mkdir $ARGV[1]");  
my $filename=$ARGV[0];  
$filename-- s/^.*\///;  
$filename=~ s/\_pdb//;  
#$filename =~ s/" *"/"  
$filename = my $filename . ".dat";  
print "$filename  
open OUT,">$filename;  
print OUT "SQUID LEUCOCYTE COUNT FROM SQUID M
```

• ДНК – открытие и лейкоцитов Ф. М.

- ДНК – открытие и выделение **"нуклеина"** из ядер (нуклеус) лейкоцитов Ф. Мишером 1869 г.
- ДНК – линейный сополимер на основе ортофосфорной **кислоты**.
- ДНК – линейный сополимер ортофосфорной кислоты и **дезоксирибозы**

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

Дезоксирибоза и ортофосфорная кислота

```
##(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;
```

```
if ($qnum > 0) {
#system("mkdir -p $dir/$schnum");
my $filename=~ s/^\./\./;
$filename=~ s/\./\./;
$filename=~ s/\./\./;
# $filename=$schnum.".".$qnum."/".$filename.".dat";
$filename="$dir/".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
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 " ",@{$qartets{$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;
```

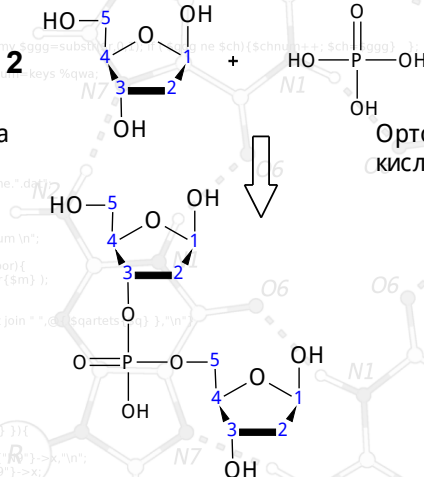
```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
```

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

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

Дезоксирибоза

Ортофосфорная кислота



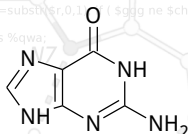
Purine

```
foreach my $m (sort {$a<=>$b} keys %coor){
    my %qartets= %4wa ; #find quart( $coor{$m} );
    my %q= find_qartets($m);
```

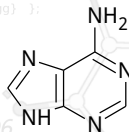
```
# foreach my $q ( keys %qartets){
6      foreach my $k ( keys %qartets){
2          ...
}
```

my \$nx; my **1**ny; my \$nz;
my \$ox; my \$oy; my \$oz;

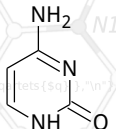
Pyrimidin



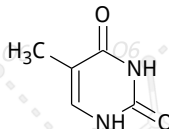
Guanine



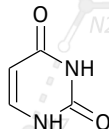
Adenine



Cytosine



Thymine



Uracil

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

ОСНОВАНИЯ И ОСТОВ

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

```
foreach my $r ( keys %coor{"0"} ) { my $snum=0; if ( $snum ne $snum ) { $snum++;
```

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

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

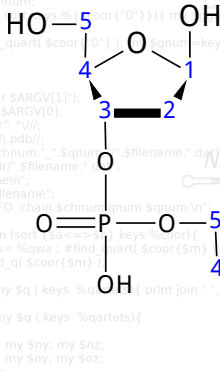
```
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
```

```
$nz=$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;
```



+ 2B



Ортофосфорная
кислота

Дезоксирибоза

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

Нуклеотид ДНК

```

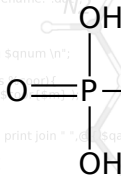
my ($my $coor, my $snum)=read_pdb($ARGV[0]);
my $my $coor=read_pdb($ARGV[0]);
my $my $dir=$ARGV[1];
my $my $sch, my $snum;
foreach my $r ( sort keys %{$coor{"0"}} ) { my $sggg=substr($r,0,1); if ( $sggg ne $sch ) { $snum=$sggg } };

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

```

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

Ортофосфорная
кислота



```

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

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

  foreach my $q ( keys %qartets ) {

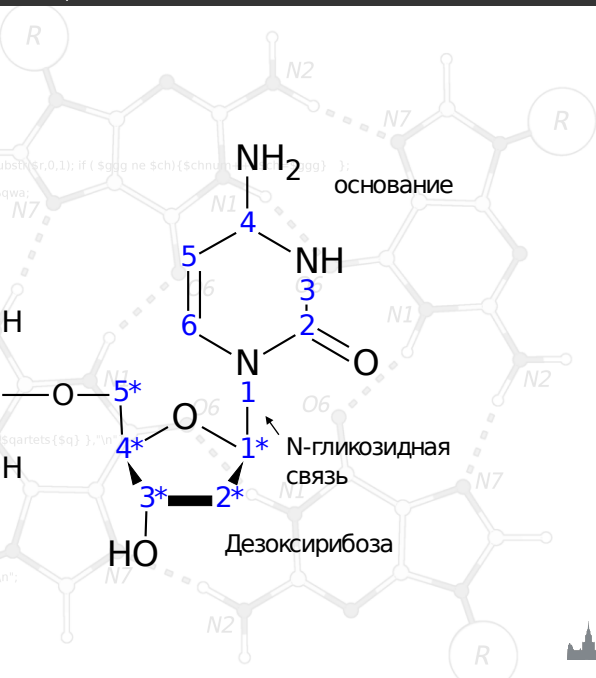
    my $nx; my $ny; my $nz;
    my $ox; my $oy; my $oz;
    my $r;

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

```

      print "$q $coor{$m} {$res} {"R"}->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 );
```

Нуклеотид РНК

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

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

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

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

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

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

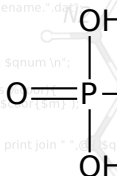
```
  my $nx; my $ny; my $nz;
  my $ox; my $oy; my $oz;
  my $r;
```

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

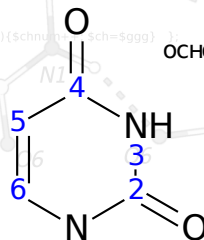
```
    # print "$q $coor{$m} {$res} {"R"}->x, "\n";
    $nx=$nx+ $coor{$m} {$res} {"N9"}->x;
    $ny=$ny+ $coor{$m} {$res} {"N9"}->y;
    $nz=$nz+ $coor{$m} {$res} {"N9"}->z;
```

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

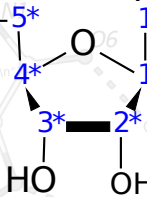
Ортофосфорная
кислота



основание



Н-гликозидная
связь



рибоза

Псевдоуридин

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

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

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

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

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

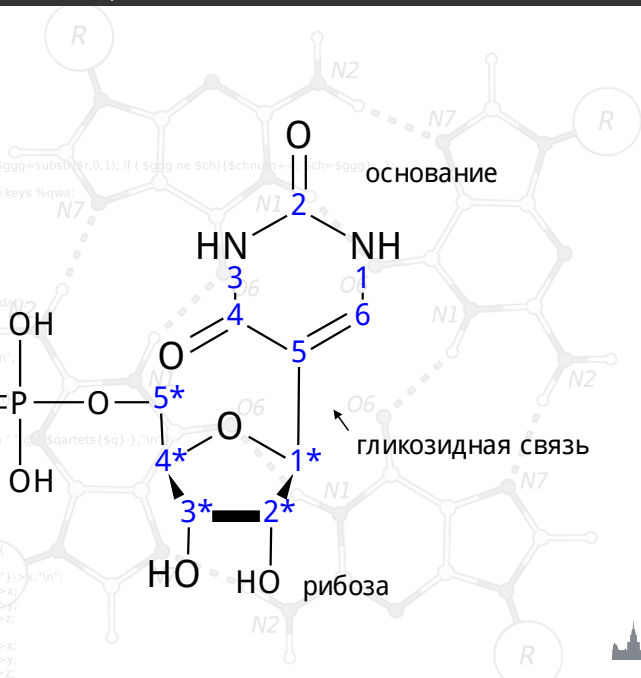
```
foreach my $q ( keys %qartets ){
```

```
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"}->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;
```



Цепь ДНК

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

my ($mycoor,$mysch,$my$ch,$my$chnum)=read_pdb($ARGV[0]);
my $mycoor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch,$my$chnum;
foreach my $f ( sort keys %{$coor{"0"}} ) { my $sgg=substr($f,0,1); if ($sgg ne $sch) { $schnum++; $sch=$sgg } };
```

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

```
if ($qnum > 0) {
    #system("mkdir $ARGV[1]");
    my $filename=$ARGV[0];
    $filename=~ s/^\.//;
    $filename=~ s/\.pdb//;
    # $filename=$schnum.".".$qnum."/".$filename.".dat";
    $filename=$mdir/".$filename.".dat";
    print "$filename\n";
    open OUT,">$filename";
    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 $q ( keys %qartets ) { print join " ",@{$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;
```

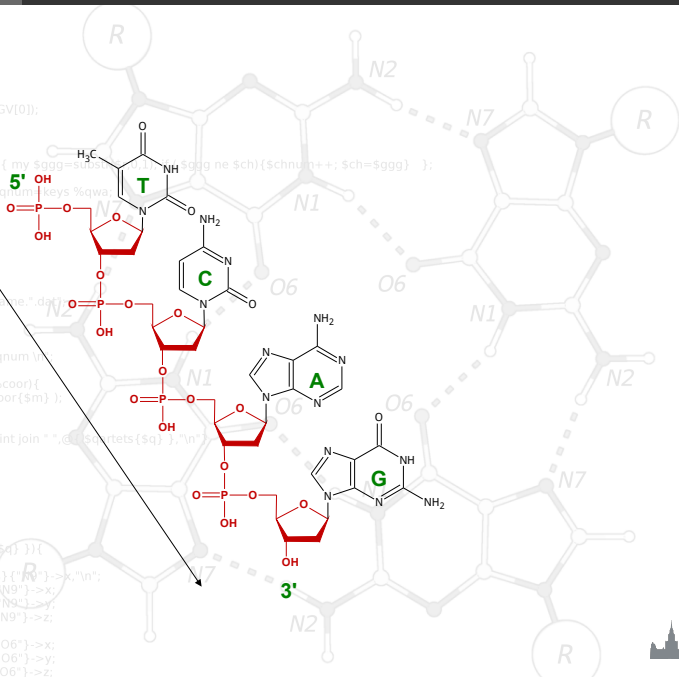
```
        $ny=$ny+ $coor{$m} {$res} {"N9"}->y;
```

```
        $nz=$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;
```



ДВОИ

```
foreach my $r ( sort keys %{$coor{"0"0}}{ my $ggg=subst($r,0,1); if ( $ggg ne $ch){ $chnu1++; $sch=$ggg } \;
```

```
my %qwa=find quart( $coo{"O"} ); my $qnum=length $qwa
```

print OUT "#INFO chain \$O=P—

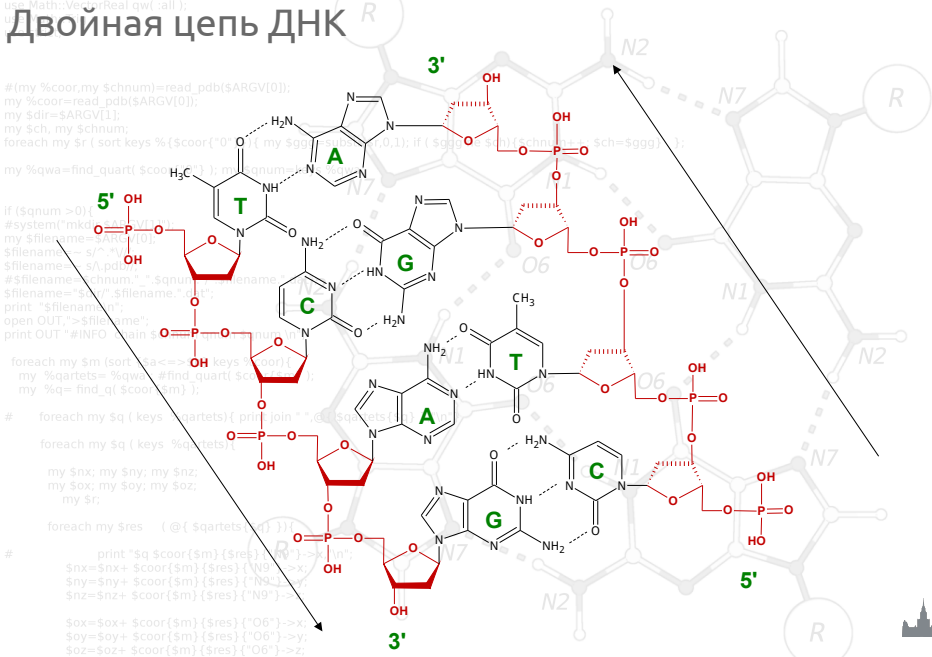
```
my %q= find_q( $coord,$m );
```

```
#    foreach my $q ( keys %gartets)
```

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

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

```
$ny=$ny+ $coor{$m}{ $res }{"N9"}=
```



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

Канонические взаимодействия

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

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

```
if ($snum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^.*\./;
$filename=~ s/\./_/;
#$filename=$snum."_"$filename;
$filename=$dir.$filename;
print "$filename\n";
open OUT ">$filename";
print OUT "QUO_chain sch $snum $snum\n";
```

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

```
# 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;
```

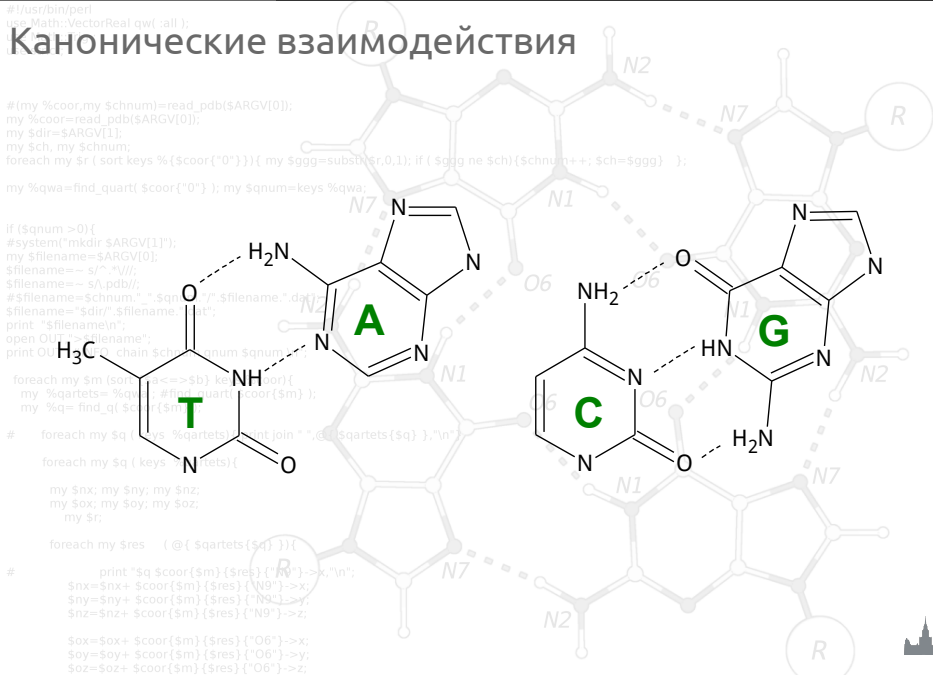
```
$ny=$ny+ $coor{$m} {$res} {"N9"}->y;
```

```
$nz=$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;
```



Неканони

Неканонические взаимодействия

```

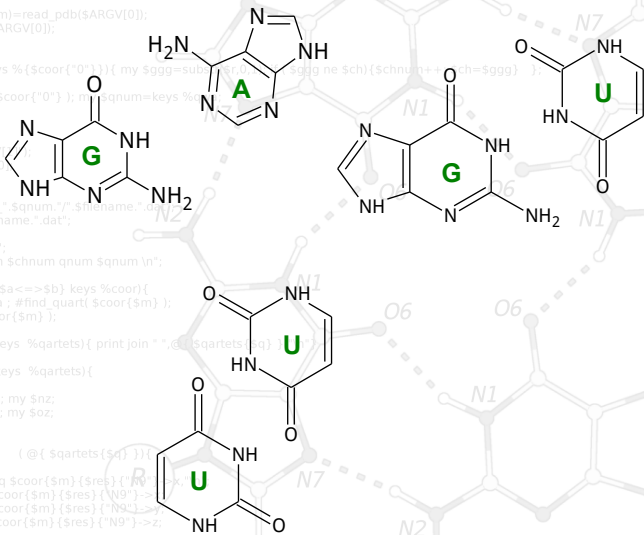
#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch,my $chnum;
foreach my $r ( sort keys %{$coor{"O"}} ){ my $sqq=substr($r,0,$sqq ne $ch){ $chnum++;$h=$sqq;

```

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

```
if ($?num > 0){
    $system\mkdir $ARGV[0];
    my $filename=$ARGV[0];
    $filename== s/~/ /;
    $filename== s/./_/;
    $filename=$cnum.".".$?num."/".$filename;
    $filename="$dir/".$filename;
    print "$filename\n";
    open OUT,">$filename";
    print OUT "INFO chain $cnum anum $?num\n";
}
```

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



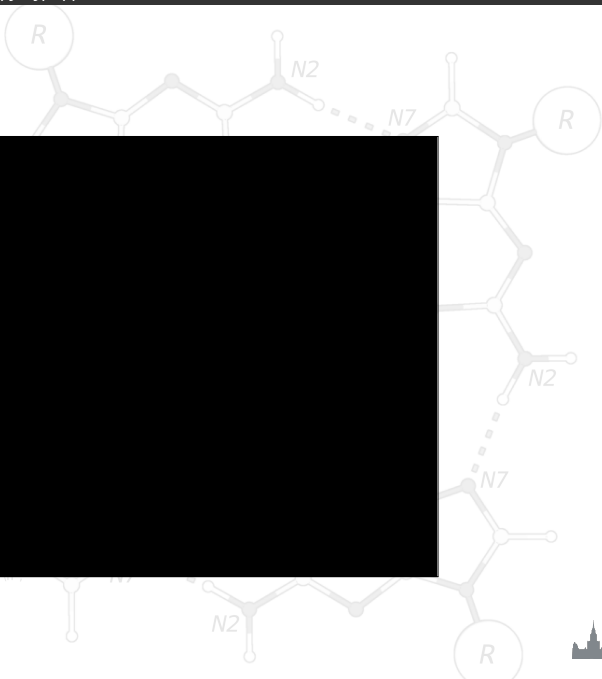
Структура

```

#(my %coo;my $cnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $cnum;
foreach my $r ( sort keys %
my %qwa=find quart( $coo

```

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



Структура ДНК

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

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

Структура ДНК впервые была предложена Watson и Crick в 1953 на основе результатов РСА низкого разрешения

Основные свойства:

- Две антипараллельные цепи
- ДНК это двойная спираль
- Имеет две оси симметрии

```
foreach my $s ( keys %$snum ) {
    my $q="find_q( $coor{$s} );
    my $q="find_q( $coor{$s} );

    #
    foreach my $q ( keys %$q ) {
        my $x1,$y1,$z1,$x2,$y2,$z2;
        my $x;
        my $y;
        my $z;

        foreach my $res ( @{$q{$q}} ) {
            print "$q $coor{$s} {$res} {"R"}->x,"n";
            $x=$x+$coor{$s} {$res} {"N9"}->x;
            $y=$y+$coor{$s} {$res} {"N9"}->y;
            $z=$z+$coor{$s} {$res} {"N9"}->z;

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



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

Взаимодействия в ДНК

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

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

Два типа взаимодействий гетероциклических оснований в ДНК.

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

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

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

```
$filename=~ s/^\s*//;
```

```
$filename=~ s/\s*.pdb//;
```

```
#$filename=$snum." ".$snum." ".$filename." .dat";
```

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

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

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

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

```
foreach my $m { sort keys %{$coor{"0"}} {
```

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

```
my %$q= %$qartets;
```

```
# 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;
```

```
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;
```

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

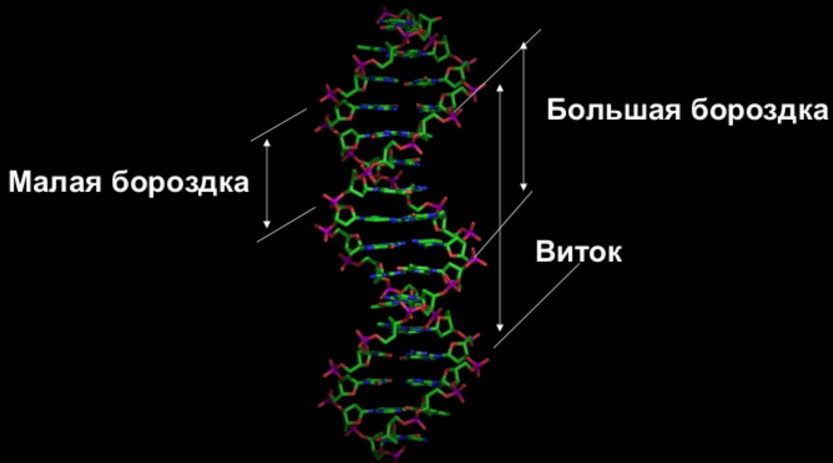
```
print "$q $coor{$m} {$res} {"R"}->y,"n";
```

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

• Копланарные взаимодействия (в одной плоскости). В основном реализуются как водородные связи.

• Стопочные взаимодействия основаны на Ван-дер-Ваальсовых взаимодействиях.

Спираль ДНК



Атомы принадлежащие большой и малой бороздкам

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my ($ARGV[0], $ARGV[1]) = @ARGV;
my $coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $schnum;
foreach my $f ( sort keys %{$coor{"0"}} ) { my $ggg=substr($f,0,1); if ( $ggg ne $sch ) { $schnum++; $sch=$ggg } }
```

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

```
if ($qnum > 0) {
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/~/V//;
$filename=~ s/\.pdb//;
#$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 " ", $q, "\n";
foreach my $q ( keys %qartets ) {
my $nx; my $ny; my $z;
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;
```

Большая бороздка

Adenine C6, N6, C5, N7, C5

Guanine C6, O6, C5, N7, C8

Cytosine C6, C5, C4, N4

Thymine C6, C5, C4, O4, C5M

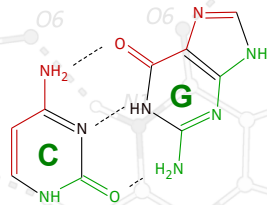
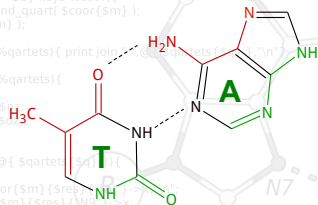
Малая бороздка

C2, N3, C4, N9

C2, N2, N3, C4, N9

O2, N1, C2

O2, C2, N1, C6



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

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

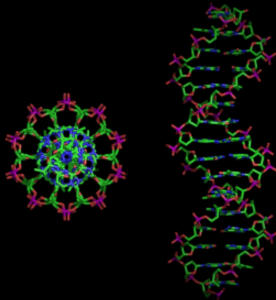
```
if ($
#sys
my $
$file
$file
#fil
$file
print
open
print
fore
my
my
#
```

```
#
print "sq $coor{$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;

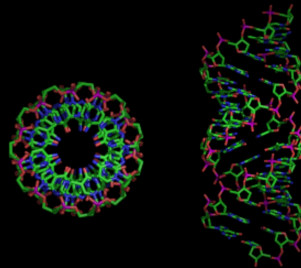
$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```

Регулярные формы спирали ДНК

В-форма



А-форма



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

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

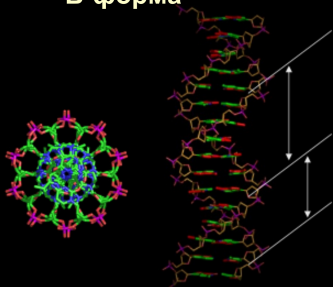
```
if ($%
#sys
my $
$file
$file
#$fil
$file
print
open
print
for
m
m
#
```

```
#
print "sq $coor{$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;

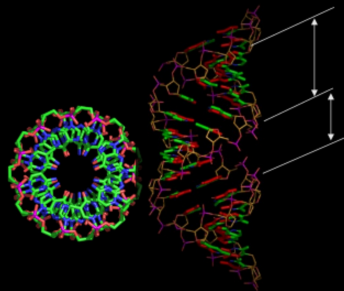
$ox=$ox+ $coor{$m}{$res}{"O6"}->x;
$oy=$oy+ $coor{$m}{$res}{"O6"}->y;
$oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```

Регулярные формы спирали ДНК

В-форма



А-форма



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

Торсионные углы НК

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

```
my %mcoor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $snum;
foreach my $r ( sort keys %
```

```
my %qwa=find_quart( $coor
```

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

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

```
# foreach my $q ( keys
```

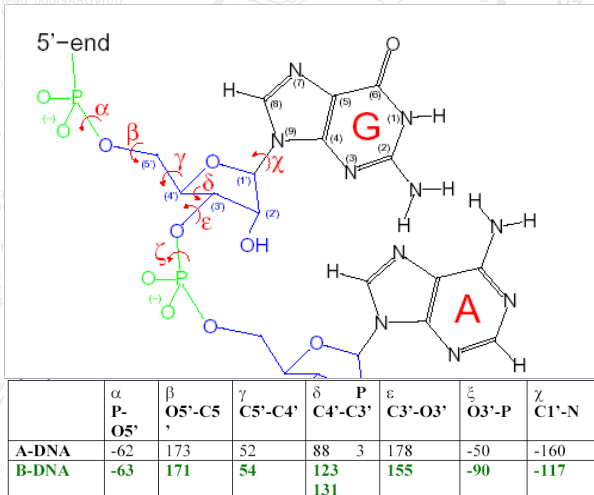
```
foreach my $q ( keys
```

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

```
foreach my $res (
```

```
# print "$q $snum";
$nx=$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 );
```

Конформация дезоксирибозы

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

```
my %$dir=
```

```
if ($s
```

```
#sys
```

```
my $
```

```
$file
```

```
$file
```

```
#fil
```

```
$file
```

```
print
```

```
open
```

```
print
```

```
for
```

```
my
```

```
my
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

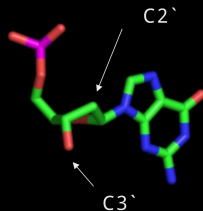
```
#
```

```
#
```

```
#
```

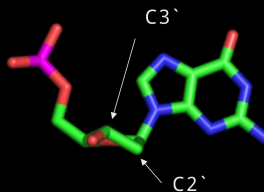
```
#
```

В-форма ДНК



C2'-endo

А-форма ДНК



C3'-endo

```
print "sq $coor{$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 );
```

Условия существования различных форм

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

Разные формы ДНК переходят друг в друга при изменении условий внешней среды:

```
if ($snum > 0) {
    #system("mkdir $ARGV[1]");
    my $filename=$ARGV[0];
    $filename=~ s/^\s*//;
    $filename=$sdir.$filename;
    #system("cp $filename $sdir/$filename.dat");
    print "File $filename created\n";
    open OUT, ">$filename";
    print OUT "#INFO: chain $snum qnum $snum\n";
```

- В-форма стабильна при нормальных физиологических условиях

- Дегидратация, понижение относительной влажности до 0.75 инициирует переход B=>A

Пример: смеси вода-этанол(метанол) при росте доли спирта > 0.75 , переход B=>A

```
# 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 );
```

Переход ДНК из А в В форму в солевом растворе

```
my ($my $coor, my $schnum)=read_pdb($ARGV[0]);
my $coor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $schnum;
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."_".$sch;
    $filename="$mdir/".$filename;
    print "$filename\n";
    open OUT, ">$filename";
    print OUT "#INFO chain $coor{$r}";
```

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

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

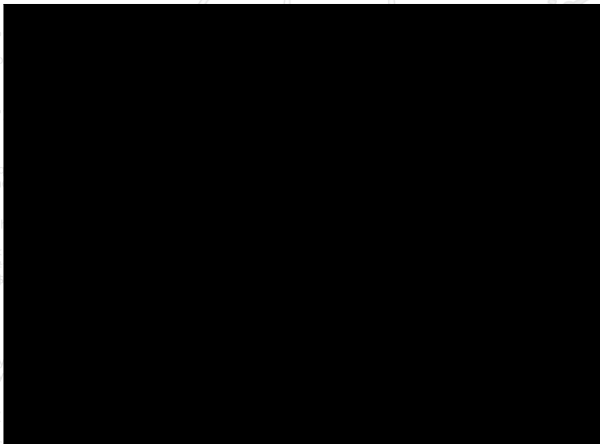
```
    foreach my $q ( keys %$qartets ) {
```

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

```
        foreach my $res ( keys %$coor{$m} ) {
```

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

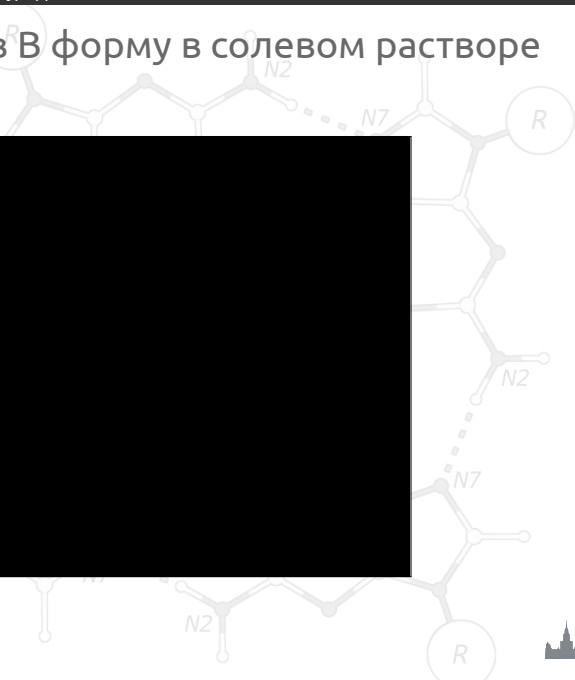
```
            $ox=$ox+ $coor{$m}{$res}{"O6"}->x;
            $oy=$oy+ $coor{$m}{$res}{"O6"}->y;
            $oz=$oz+ $coor{$m}{$res}{"O6"}->z;
```



Переход Д

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

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



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

Структура РНК

```
#{my %coor,my $  
my %coor=read_f  
my $dir=$ARGV[0]  
my $sch, my $chn  
foreach my $r ( s  
my %qwa=find_q
```

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

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

```
# foreach my
```

```
foreach my
```

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

```
foreach m
```

```
#  
$nx=$i  
$ny=$i  
$nz=$i
```

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

Структура

Структура РНК

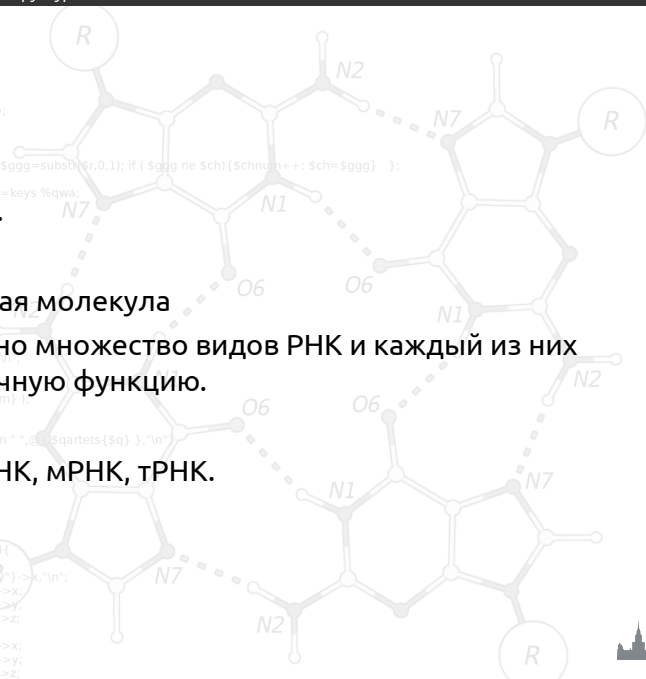
Основные свойства.

```
if ($qnum >0){  
#system("mkdir $ARGV[1]");  
my $filename=$ARGV[0];  
$filename=~ s/^.*//;  
$filename=~ s/\A pdb//;  
• Одно-цепо  
$filename="$qnum.$filename.dat";  
print "$filename\n";  
open COT, ">$filename";  
print OUT "В клетке н
```

- Одно-цепочечная молекула
- В клетке найдено множество видов РНК и каждый из них имеет специфичную функцию.

имеет специфичн

Основные типы: рРНК, мРНК, тРНК.



РНК и ДНК

```

#(my %coor,r
my %coor=re
my $dir=$AR
my $sch, my $
foreach my $i

```

```

my %qwa=fin

```

```

if ($qnum > 0)
#system("mk
my $filename
$filename==
$filename==
#$filename=
$filename="$
print "$filena
open OUT,">$
print OUT "##I

```

```

foreach my $i
my %qarte
my %q= fin

```

```

# foreach

```

```

foreach

```

```

my $n
my $o
my

```

```

foreach

```

```

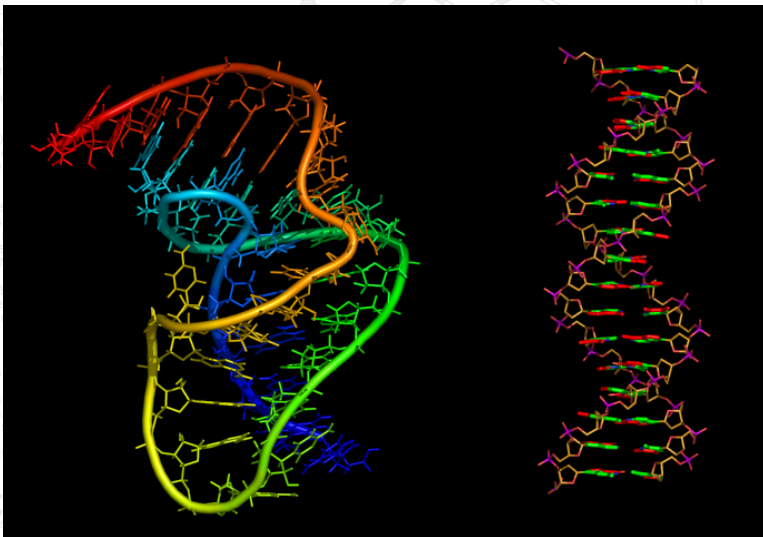
#

```

```

$nx
$ny
$nz

```



```

$ox=$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 );
```

```
use strict;
```

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

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

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

```
my $sch,$my $snum;
```

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

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

Структура РНК определяет её функцию:

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

```
  #system("ls -l $dir/$ARGV[0].");
```

```
  my $filename="-- $dir/$ARGV[0].";
```

```
  $filename="-- $dir/$ARGV[0].";
```

```
  $filename="-- $dir/$ARGV[0].";
```

```
  #system("ls -l $dir/$ARGV[0].");
```

```
  $filename="-- $dir/$ARGV[0].";
```

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

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

```
  print OUT "RIBONUCLEIC ACID\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";
```

```
    }
```

```
    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;
```

```
    }
```

```
  }
```

```
  }
```

```
  }
```

```
  }
```

```
  }
```

```
  }
```

```
  }
```

```
  }
```

Некоторые вирусы имеют РНК геном (HIV,грипп).

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

Вторичная структура РНК

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

```
if ($
#sys
my %$
$file
$file
#fil
$file
print
open
print
```

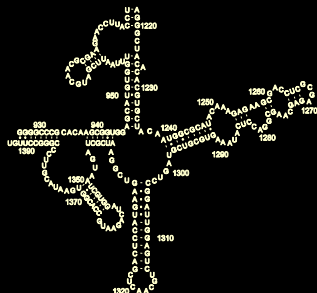
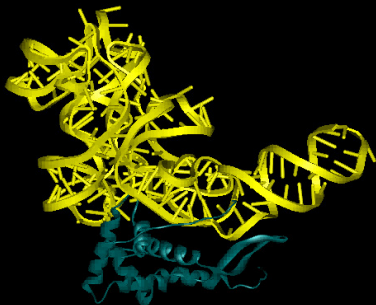
```
fore
my
my
```

#

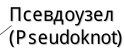
#

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



BIOP

$$\$02 = \$02 + \$C001\{\$m\}\{\$ies\}\{00\} \rightarrow Z;$$


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

Возможность предсказания вторичной структуры

```
my ($mycoor,$my $snum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch,$pov,$schew;
foreach r
```

```
my %qwa
```

```
if ($snum
```

```
#system
```

```
my $fiter
```

```
$filenam
```

```
$filenam
```

```
#$filenam
```

```
$filenam
```

```
print "$f
```

```
open OU
```

```
print OU
```

```
foreach
```

```
my %
```

```
my %
```

```
#
```

```
for
```

```
for
```

```
m
```

```
m
```

```
f
```

```
#
```

```
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;
```

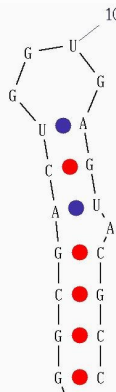
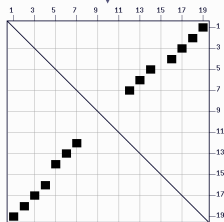
```
$ox=$ox+ $coor{$m} {$res} {"O6"}->x;
```

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

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

GGCGACUGGUGAGUACGCC

GGCGACUGGUGAGUACGCC



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

Расчёт энергии структуры по алгоритму Зукера

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

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

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

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

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

```
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} { "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;
```

UU Петля +5.9 Ккал/моль

A A
G C Стэкинг+Пара -2*2.9 Ккал/моль
G C

A Выпетливание + 3.3 Ккал/моль

G C Стекинг+Пара -1.8 Ккал/моль

U A -0.9 Ккал/моль

A U -1.8 Ккал/моль

C G -2.1 Ккал/моль

A U 3'

A Неструктурированный 5' конец 0 Ккал/моль

A

5'

Δ G = -3.2 Ккал/моль



PHK. KOTO

Зукера

- РНК связанная с белками.

- **Длинные РНК.**

• Псевдоузлы.

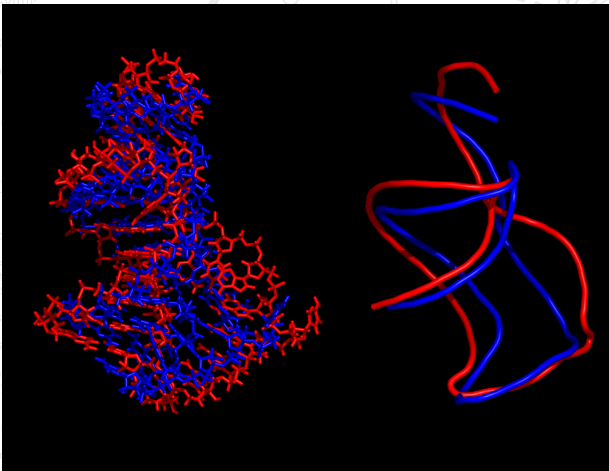
Осень, 2012

35 / 37



Моделирование структуры

ГОЛОВИН



Вопросы?

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

my ($mycoor,$my $schnum)=read_pdb($ARGV[0]);
my $mycoor=read_pdb($ARGV[0]);
my $mdir=$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 $myqwa=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=$schnum."_".$qnum."_".$filename.".dat";
$filename="$mdir/".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
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 $q ( keys %qartets ){ print join " ",@{$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;
```

