

```
#!/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 $mdir=$ARGV[1];
my $sch, my $chnum;
foreach my $r ( sort keys %{$coor} ) {
```

```
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=$chnum.".".$snum."/".$filename.".dat";
$filename="$mdir".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $snum\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 } { "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;
```

```
}
```

Структурная Биоинформатика

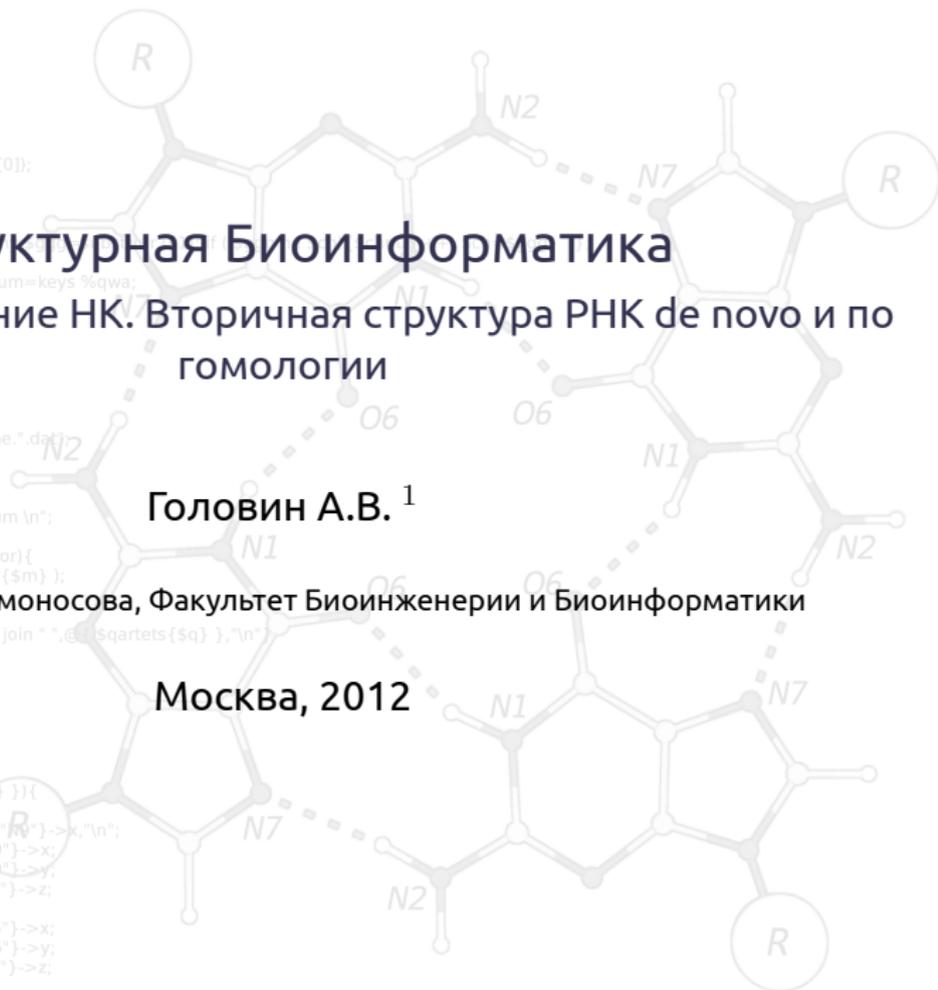
Лекция 9. Строение НК. Вторичная структура РНК de novo и по

ГОМОЛОГИИ

Головин А.В. ¹

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

Москва, 2012



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

Содержание

```

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

```

Введение

```

if ($sqnum > 0){
#system("mkdir $ARGV[1]");

```

Химическое строение НК

```

$filename="-- s/\.pdb//;
#$filename=$chnum."."$sqnum."."$filename." dat";
$filename="$dir".$filename." dat";
print "$filename\n";
open(FILE,">$filename");
print FILE "$chnum\n";

```

3D Структура ДНК

```

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

```

Структура РНК

```

foreach my $q ( keys %qartets ){
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";

```

Вторичная структура РНК

```

#
$nx=$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 );

```

ДНК

```

#(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 } );

```

Расшифровка аббревиатуры ДНК:

```

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

```

- 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

```

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 );
```

Дезоксирибоза и ортофосфорная кислота

```
#!/(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{"O"}} ){ my $ggg=substr($r,0,3); if ($ggg ne $sch){ $chnum++; $sch=$ggg; } }
my $qwa=find_quart($coor{"O"}); my $qna=keys %$qwa;
```

Дезоксирибоза

```
if ($qnum > 0){
#system("mkdir -p $dir/$qnum");
my $filename="-- s/~/V//";
$filename="-- s/\./pdb//";
#$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});
```

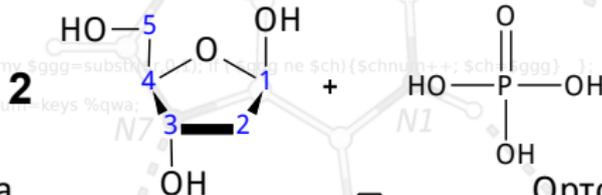
```
# foreach my $q ( keys %qartets ){ print join " ",@{$qartets{$q}},"\n";
foreach my $q ( keys %qartets ){
```

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

```
foreach my $res (@{ $qartets{$q} ){
# print "$q $coor{$m}{$res} {"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;
```



Ортофосфорная кислота

Основания

```

#!/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) ( $ggg ne $ch){$chnum++; $ch=$ggg } };

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

if ($qnum > 7) {
#system("mkdir $dir/$ARGV[1]");
my $filename=$dir."/N($qnum)";
my $filename=$dir."/N($qnum)";
my $filename=$dir."/N($qnum).pdb//";
my $filename=$dir."/N($qnum).dat";
my $filename=$dir."/N($qnum).dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum\n";

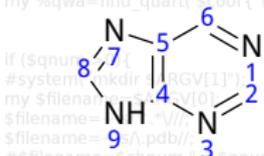
foreach my $m (sort { $a-<=>$b } keys %coor){
my %qartets= %qwa; #find_quart( %coor{$m} );
my %q= find_quart( %coor{$m} );

# 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 $qx; my $qy; my $qz;

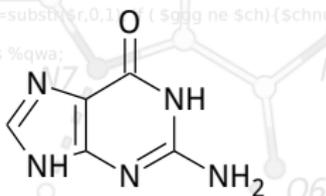
foreach my $res ( @{$qartets{$q}} ){
print "$q $coor{$m}{$res} {"R"}->x,"n";
$nx=$nx+ $coor{$m}{$res} {"N9"}->x;
$ny=$ny+ $coor{$m}{$res} {"N9"}->y;
$nz=$nz+ $coor{$m}{$res} {"N9"}->z;

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

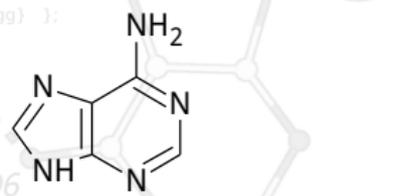
```



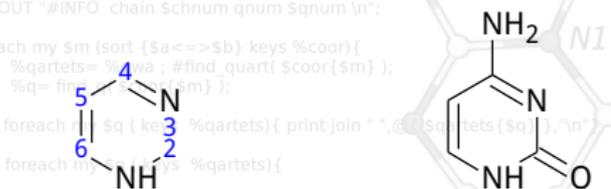
Purine



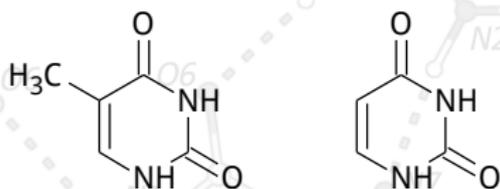
Guanine



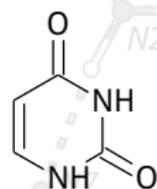
Adenine



Cytosine



Thymine



Uracil



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

ОСНОВАНИЯ И ОСТОВ

```
#!/(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 ( keys %coor{"O"} ){ my $OH=substr($r,0,1); if ( $sgg ne "" ){ $chnum++; $ch=$sgg }
my %qwa=find_quart( $coor{"O"} ); $snum=keys %qwa;
```

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

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

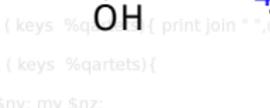
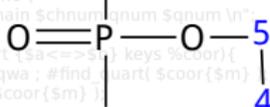
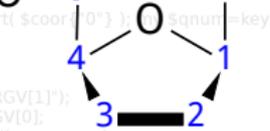
```
# foreach my $q { keys %qartets { print join " ", $qartets{$q}, "\n" }
foreach my $q { keys %qartets {
```

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

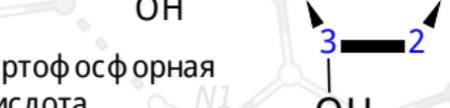
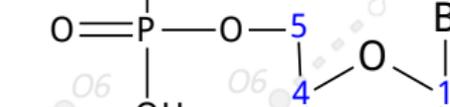
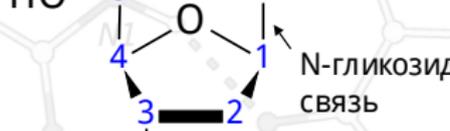
```
foreach my $res ( @ { $qartets{$q} } ){
```

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



+ 2B



основание

N-гликозидная
связьОртофосфорная
кислота

Дезоксирибоза



Нуклеотид РНК

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

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

my $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;
# $filename=$schnum;
$filename="$mdir".$filename;
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $schnum qnum $sqnum\n";
```

```
foreach my $m ( sort { $a-<=>$b } keys %$qartets ) {
my %qartets = %qwa ; #find_quart($coor{"0"});
my %q = find_q( $coor{$m} );
```

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

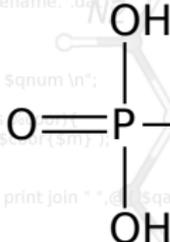
```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
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;
```

```
$ox=$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 %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;
```

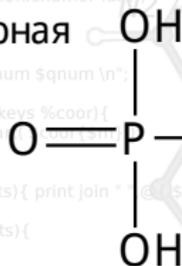
```
if ($qnum > 0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename="-- s/^.*\./";
$filename="-- s/\.pdb//";
#$filename=$chnum.".".$qnum.".".$filename.".dat";
$filename="-- s/\.dat//";
print "S: $filename";
open OUT, ">$filename";
print OUT "P: $chnum qnum $qnum \n";
```

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

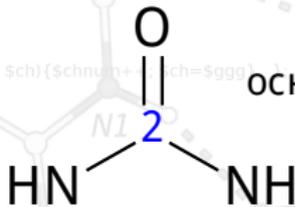
```
# foreach my $q ( keys %qartets ){ print join " ", $qartets{$q} , "\n";
foreach my $q ( keys %qartets ){
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ){
# print "$q $coor{$m} {$res} {"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;
```

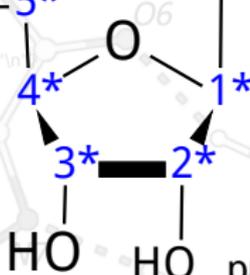
Ортофосфорная
кислота



основание



гликозидная связь



рибоза

Цепь ДНК

```

#!/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 $ggg=subst( "0,1,1", "$ggg ne Sch" )($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 ){

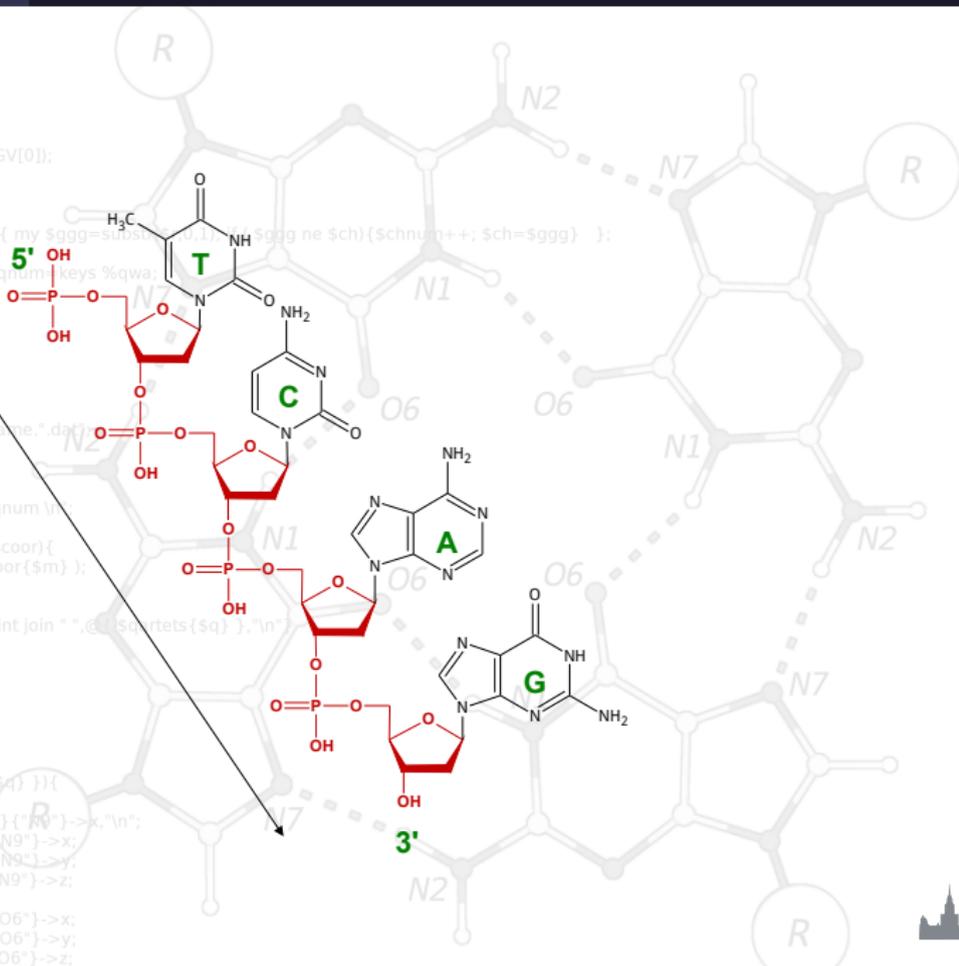
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;

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

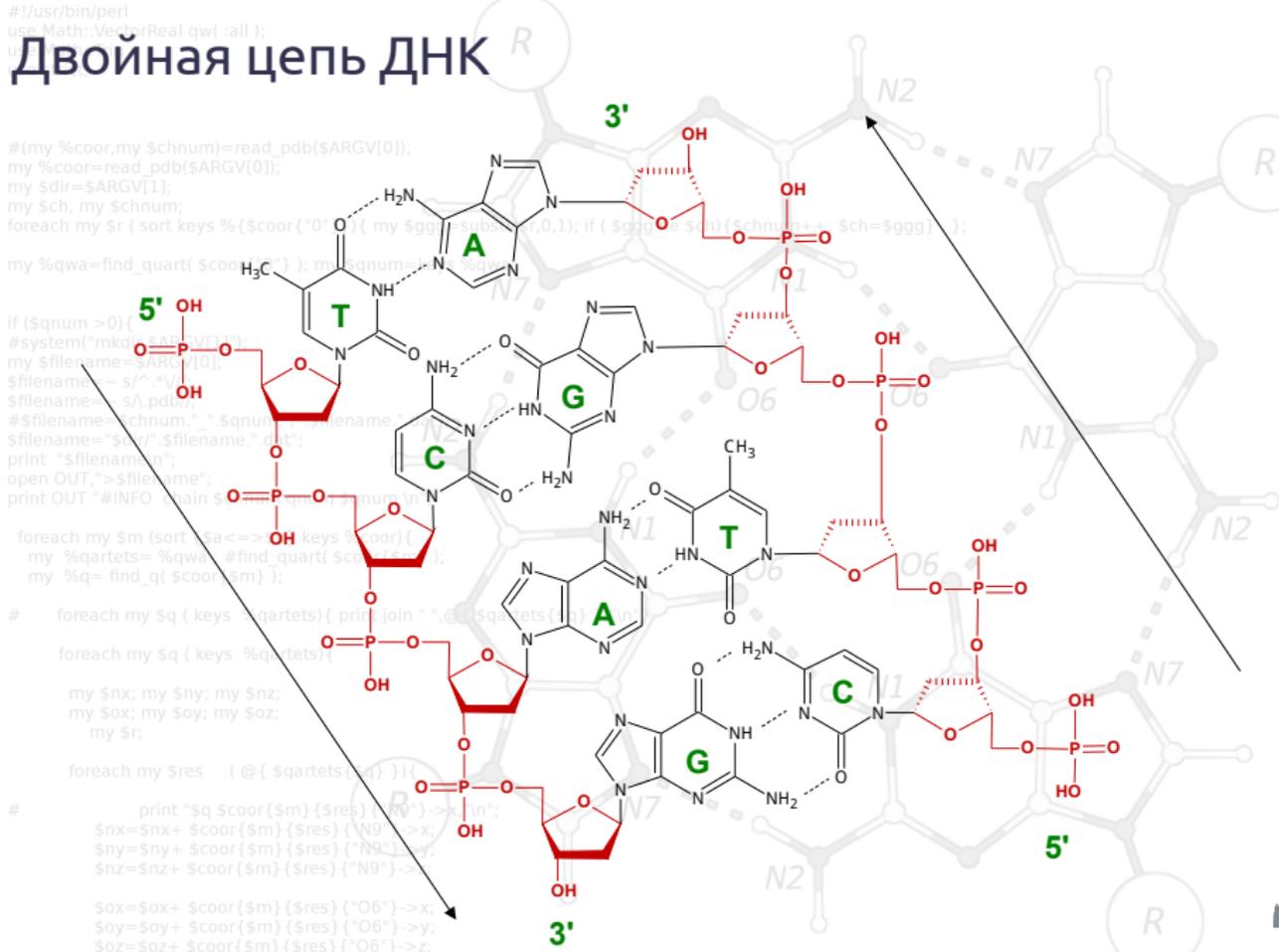
print "$q %coor{$m}{ $res } {"N9"}->x, "\n";
$nx=$nx+ %coor{$m}{ $res }{"N9"}->x;
$ny=$ny+ %coor{$m}{ $res }{"N9"}->y;
$nz=$nz+ %coor{$m}{ $res }{"N9"}->z;

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

```



Двойная цепь ДНК



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

Канонические 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{"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/^\./\.\./;
$filename=~ s/\./_./;
#$filename=$chnum.".".$qnum.".".$filename.".dat";
$filename=$dir.$filename.".dat";
print "$filename\n";
open OUT ">$filename";
print OUT "chain $ch $chnum $qnum\n";
```

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

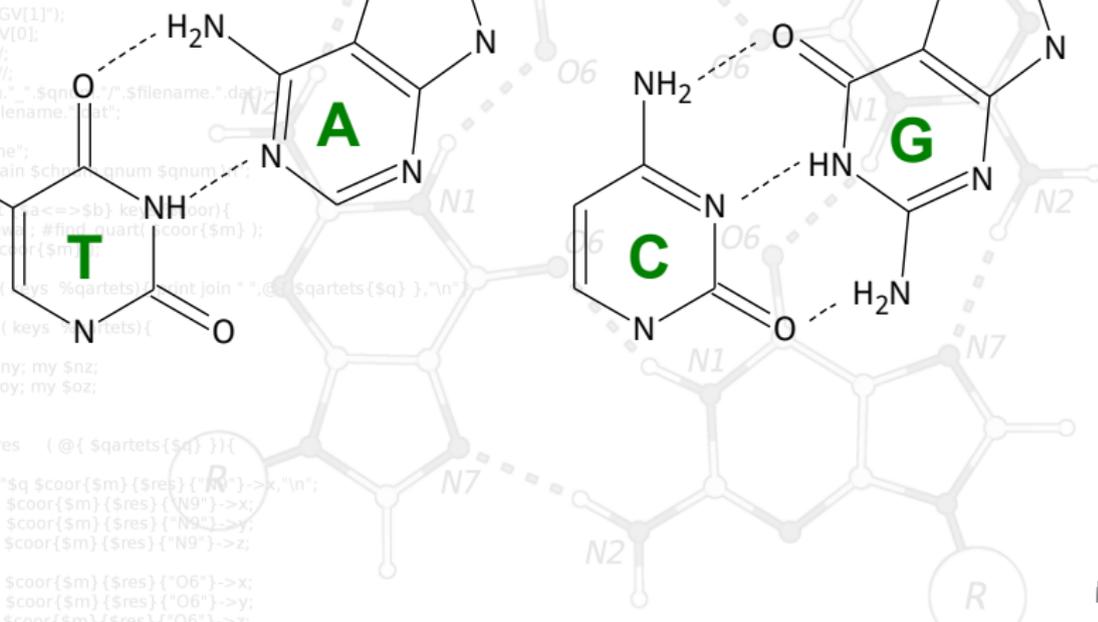
```
$ny=$ny+ $coor{$m}{$res}{\"N9\"}->y;
```

```
$nz=$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;
```



Структура ДНК

```

#!/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 ) {
    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."_".$r;
        $filename="$dir"/.$filename;
        print "$filename\n";
        open OUT,">$filename";
        print OUT "#INFO chain $ch\n";

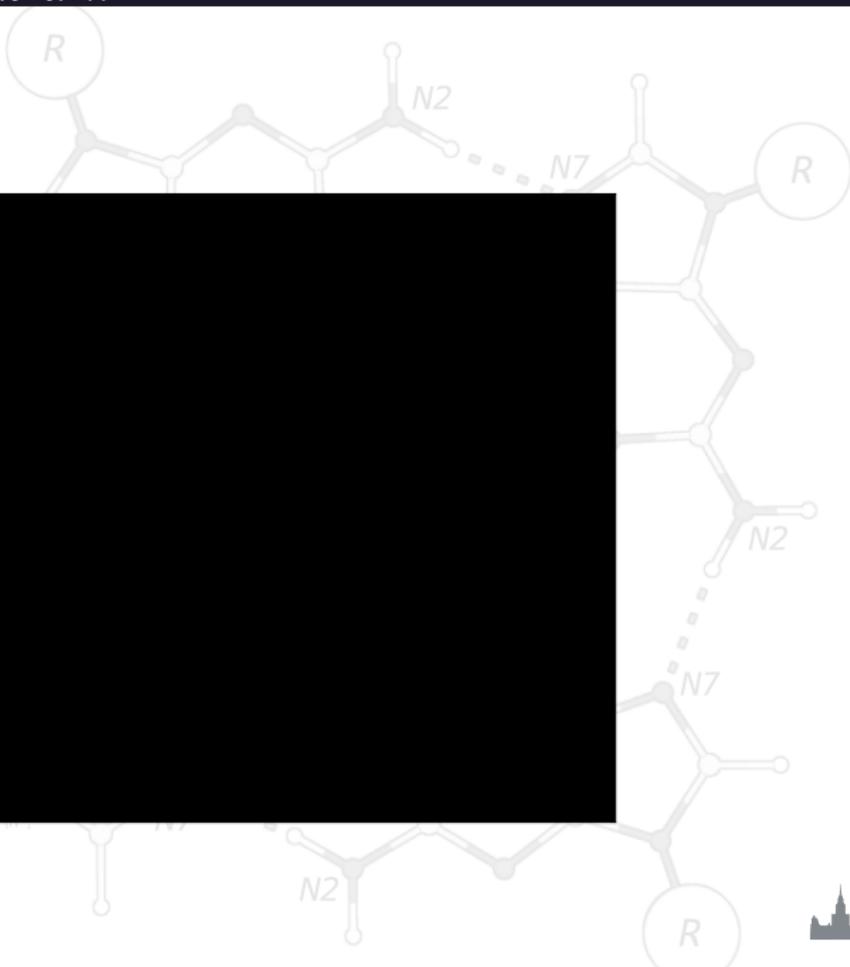
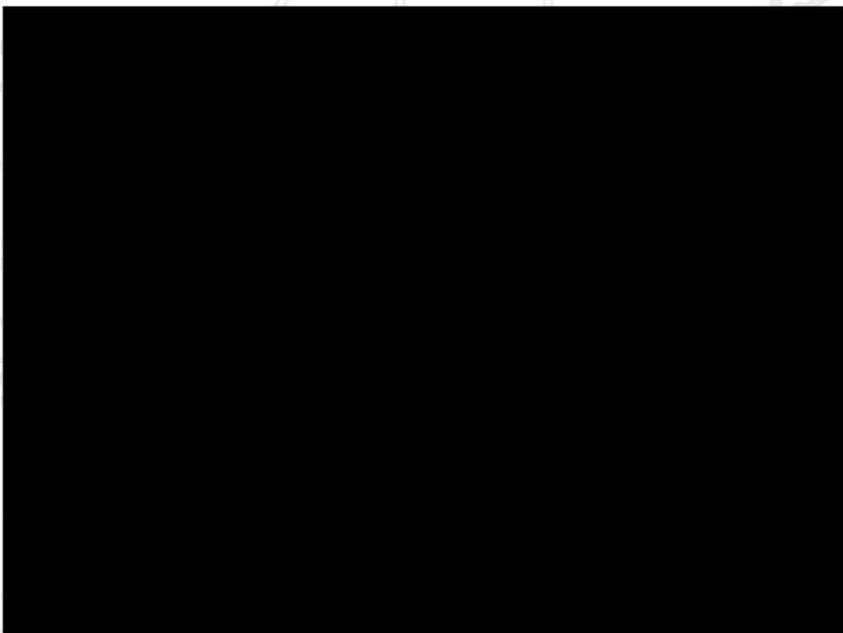
        foreach my $m (sort { $a<
            my %qartets = %qwa ; #
            my %q = find_q( $coor{$m} );

            #   foreach my $q { keys
            foreach my $q { keys
                my $nx; my $ny; my
                my $ox; my $oy; my
                my $r;

                foreach my $res {
                    print "$q So
                    $nx=$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;

```



Структура ДНК

```
#!/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{"O"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $ch){ $chnum++; $ch=$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 $q ( keys %$coor ){
my $q=substr($q,0,1);
my %q= find_q( $coor{$q} );
# foreach my $q ( keys %$q ){
foreach my $q ( keys %$q ){
my $x1, my $y1, my $z1;
my $x2, my $y2, my $z2;
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 );
```

Взаимодействия в ДНК

```

#(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;
```

Два типа взаимодействий гетероциклических оснований в ДНК.

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

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

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

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

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

```
    my $q;
```

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

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

```
      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;
```

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

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

Атомы принадлежащие большой и малой бороздкам

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my $pdb=$ARGV[0];
my $scor=$ARGV[1];
my $sch, my $schnum;
foreach my $r ( sort keys %{$scor{"0"}} ){ my $ggg=subst($r,0,1); if ( $ggg ne $sch ){ $schnum++; $sch=$ggg } ;
```

	Большая бороздка	Малая бороздка
Adenine	C6, N6, C5, N7, C5	C2, N3, C4, N9
Guanine	C6, O6, C5, N7, C8	C2, N2, N3, C4, N9
Cytosine	C6, C5, C4, N4	O2, N1, C2
Thymine	C6, C5, C4, O4, C5M	O2, C2, N1, C6

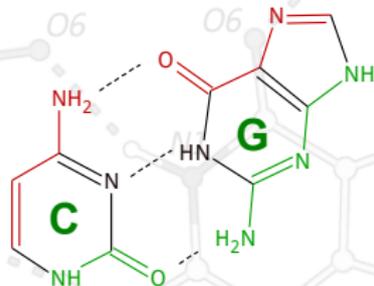
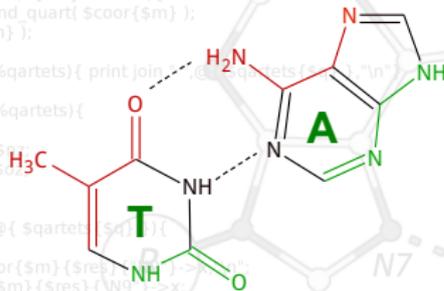
```
my $qwa=find_quart( $scor{"0"} ); my $sqnum=k
if ($sqnum >0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/~/~/;
$filename=~ s/\.pdb//;
#$filename=$schnum.".".$sqnum.".$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $schnum\n";
```

```
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} {$res} {"$qartets{$q}";
$nx=$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 );
```

Регулярные формы спирали ДНК

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

```
my %
```

```
if ($
```

```
my $
```

```
print
```

```
for
```

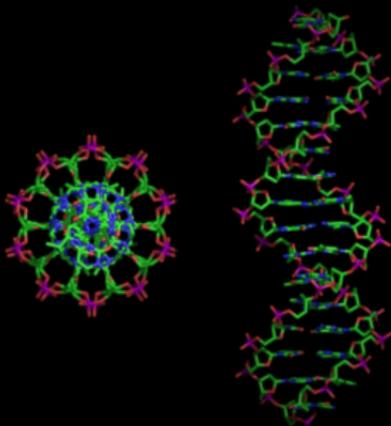
```
#
```

```
#
```

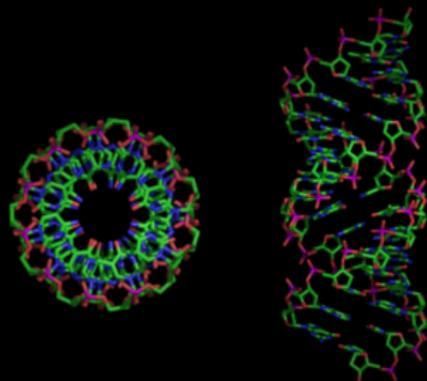
```
$nx=$nx+
```

```
$sox=$sox+
```

В-форма



А-форма



```
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 );
```

Регулярные формы спирали ДНК

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

```
my %
```

```
if ($
```

```
#sys
```

```
my $
```

```
$file
```

```
$file
```

```
#fil
```

```
$file
```

```
print
```

```
open
```

```
print
```

```
fore
```

```
m
```

```
m
```

```
#
```

```
#
```

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

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

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

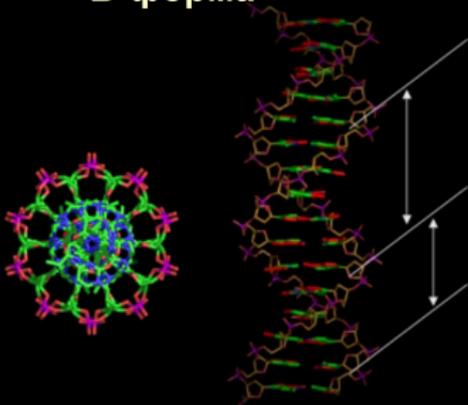
```
$nz=$nz+ $coor{$m}{$res}{\"N9\"}->z;
```

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

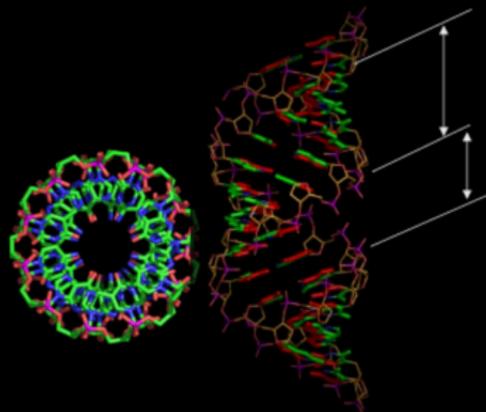
```
$oy=$oy+ $coor{$m}{$res}{\"O6\"}->y;
```

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

В-форма



А-форма



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

```

Торсионные углы НК

```

#(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 != $r-$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 $res ( sort keys %{$coor{"$q"}} ){
```

Форма	α	β	γ	δ	ϵ	ζ	χ
А-ДНК	62	173	52	88/3	178	-50	-160
В-ДНК	63	171	54	123/131	155	-90	-117

```

$ox=$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 %snum;
for($i=0;$i<10;$i++){
my %snum;
```

```
my %snum;
```

```
if ($i
```

```
#sys
```

```
my $
```

```
$file
```

```
$file
```

```
#fil
```

```
$file
```

```
print
```

```
open
```

```
print
```

```
for
```

```
m)
```

```
m)
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

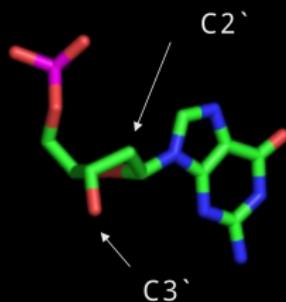
```
#
```

```
#
```

```
#
```

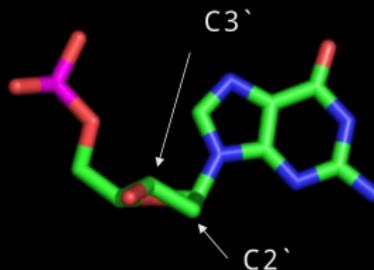
```
#
```

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 ($qnum,$qname)=$ARGV[0],$ARGV[1];
my $qname=$qname~/\./?($qname.".dat"):$qname;
my $qname=$qname~/\./?($qname.".dat"):$qname;
my $qname=$qname~/\./?($qname.".dat"):$qname;
```

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

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

- В-форма стабильна при нормальных физиологических условиях

- Дегидратация, понижение относительной влажности до 0.75 инициирует переход В=>А

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

```
my $r;
foreach my $s ( keys %$qartets ){
print "$s $qartets{$s}";
my $r;
my $nx=$nx+ $coor{$s} {$sres} {"N9"}->x;
my $ny=$ny+ $coor{$s} {$sres} {"N9"}->y;
my $nz=$nz+ $coor{$s} {$sres} {"N9"}->z;

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

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

Переход ДНК из А в В форму в солевом растворе

```

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{ $r } );
}

```

```

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

```

```

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

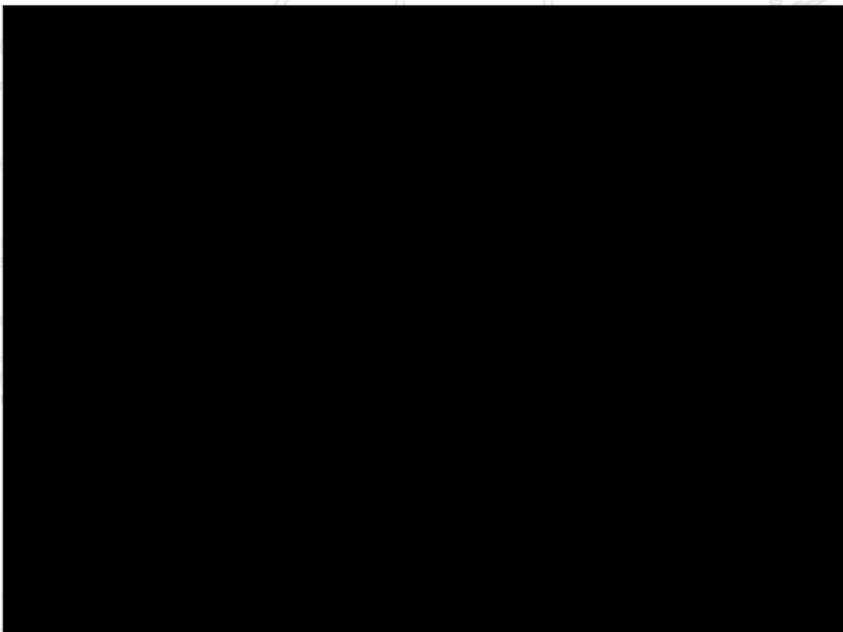
```

```

# print "$q So";
$nx=$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 (%coor,%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 ) {
    my %qwa=find_quart( $coor{ $r } );
}

```

```

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

```

```

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

```

```

            # print "$q $sch $res\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 );
```

Структура РНК

```

#(my %coor,my $s
my %coor=read_p
my $dir=$ARGV[1]
my $ch, my $chn
foreach my $r ( so
my %qwa=find_q
```

```

if ($qnum >0){
#system("mkdir $
my $filename=$A
$filename=~ s/\^
$filename=~ s/\.p
#$filename=$chn
$filename="$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
```

```

# pr
$nx=$r
$ny=$r
$nz=$r
```

```

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

Структура РНК

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

my ($f,$d,$s,$c,$n)=read_pdb($ARGV[0]);
my $qwa=find_quart($c{"O"}); my $qnum=keys %$qwa;
if ($qnum > 0){
  #system("mkdir $ARGV[1]");
  my $filename=$ARGV[0];
  $filename=~s/~/~/;
  $filename=~s/\.pdb//;
  #filename=$filename.".dat";
  print "$filename\n";
  open OUT ">$filename";
  print OUT "chain $chain $chain $chain $chain\n";
  foreach my $m (keys %$qartets){
    my $qartets=$qwa; #find_quart($c{$m});
    my $q=find_q($c{$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 $c{$m} {$res} {"R"}->x,\n";
      $nx=$nx+$c{$m} {$res} {"N9"}->x;
      $ny=$ny+$c{$m} {$res} {"N9"}->y;
      $nz=$nz+$c{$m} {$res} {"N9"}->z;
      $ox=$ox+$c{$m} {$res} {"O6"}->x;
      $oy=$oy+$c{$m} {$res} {"O6"}->y;
      $oz=$oz+$c{$m} {$res} {"O6"}->z;
    }
  }
}
```

Основные свойства.

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

Основные типы: рРНК, мРНК, тРНК.

```
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
foreach my $res (@{$qartets{$q}}){
  print "$q $c{$m} {$res} {"R"}->x,\n";
  $nx=$nx+$c{$m} {$res} {"N9"}->x;
  $ny=$ny+$c{$m} {$res} {"N9"}->y;
  $nz=$nz+$c{$m} {$res} {"N9"}->z;
  $ox=$ox+$c{$m} {$res} {"O6"}->x;
  $oy=$oy+$c{$m} {$res} {"O6"}->y;
  $oz=$oz+$c{$m} {$res} {"O6"}->z;
}
```

РНК и ДНК

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

my %coor;
my $coor=ref;
my $dir=$AR;
my $sch, my $
foreach my $

my %qwa=fin

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

foreach my $
my %qarte
my %q= fi

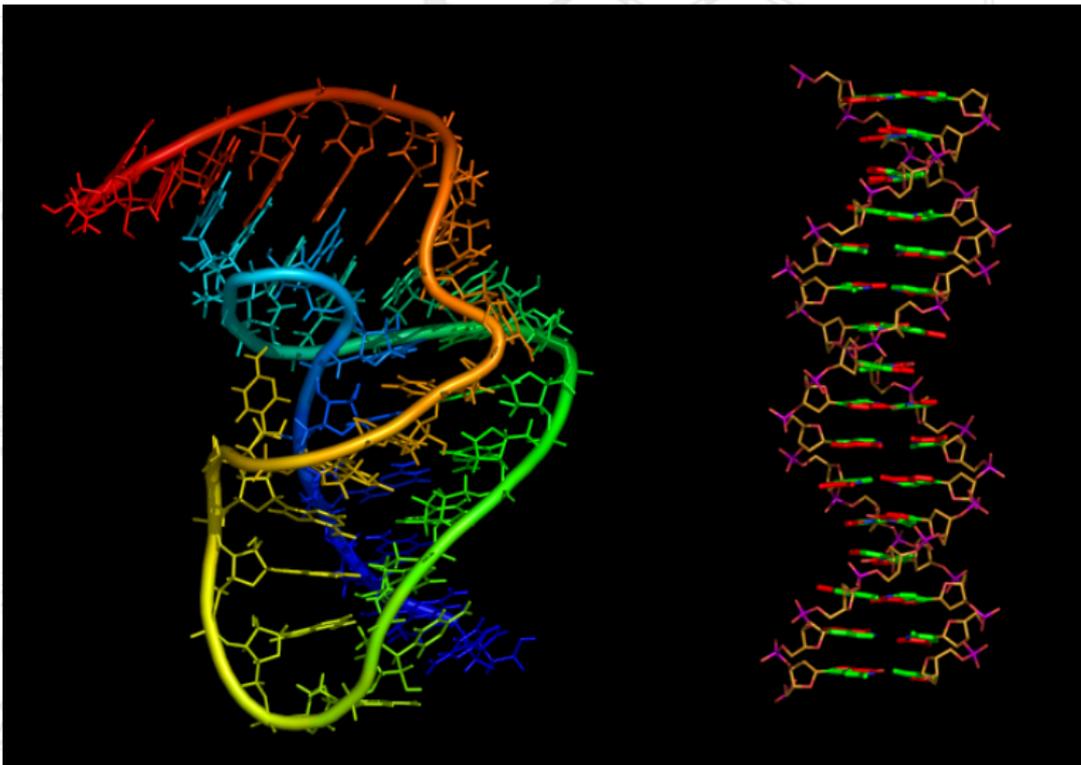
# foreach

foreach

my $n
my $o
my

foreac

#
$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 );
```

Почему важна структура РНК?

```
#!/(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 } ;
```

Структура РНК определяет её функцию:

- Регуляторная
- Структурная
- Каталитическая (рибозимы)

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

Некоторые вирусы имеют РНК геном (HIV,грипп).

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

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 );
```

Вторичная структура РНК

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

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

```
my %snum=
```

```
my %
```

```
if ($
```

```
#sys
```

```
my $
```

```
$file
```

```
#file
```

```
$file
```

```
print
```

```
open
```

```
print
```

```
for
```

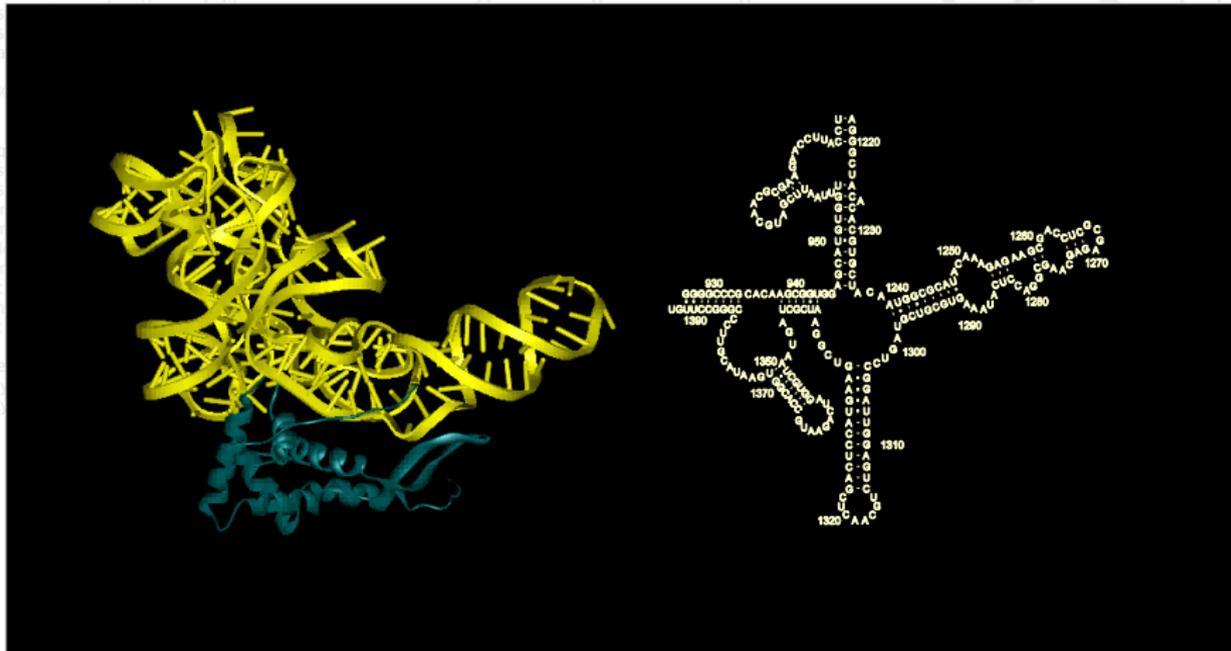
```
m
```

```
#
```

```
#
```

```
$nx=$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 );
```

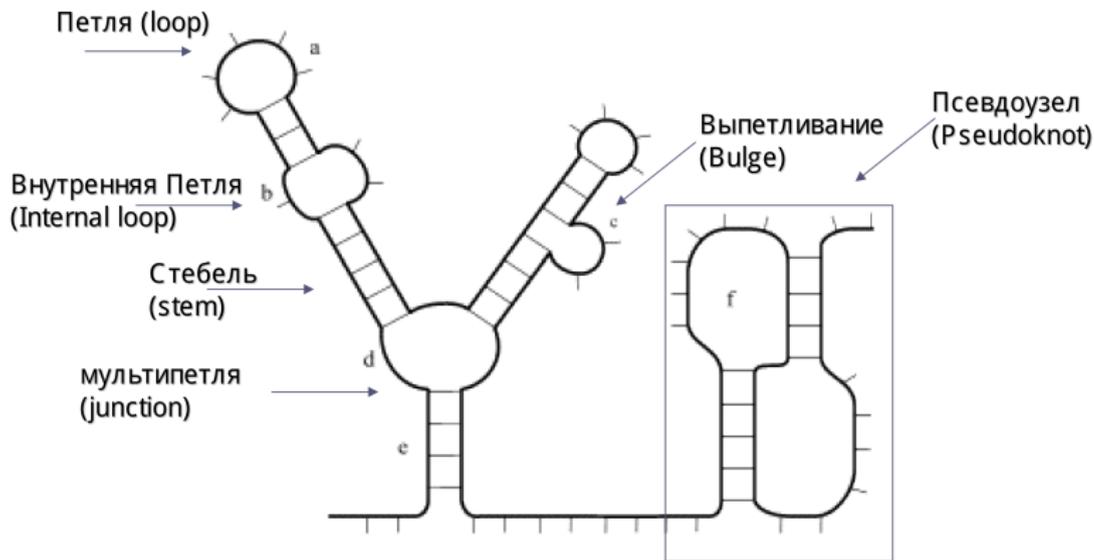
Вторичная структура РНК

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

```
if ($cnum
#system
my $filer
$filenam
$filenam
# $filena
$filenam
print "$f
open OU
print OU"
```

```
foreach
my %
my %
# for
for
π
π
fc
```

```
#
$ny=$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 );
```

Методы предсказания вторичной структуры РНК

```

#(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){
```

- Поиск максимума количества пар оснований (алгоритм Nussinov)

- Минимизация энергии (алгоритм Зукера)

- Сравнительное предсказание на основе выравнивания последовательностей РНК (Vienna RNA Package).

```
# 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} {"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 );
```

Расчёт энергии структуры по алгоритму Зукера

```
my ($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 $qwa=subst($r,0,1); if ( $qwa ne $sch ){ $schnum=$r+$sch-$qwa; }
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/\./\.\./;
# $filename=$schnum.".".$qnum.".".$filename;
$filename="$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 "m\t", $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 ( @coor{ $q } ){
my $x; my $y; my $z;
```

```
# 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' Неструктурированный 5' конец 0 Ккал/моль
A

5'

Δ G = -3.2 Ккал/моль

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my $i;
my $R;
my $r;
```

РНК, которые трудно предсказать алгоритмом

Зукера

```
my ($m,%coor,%schnum)=read_pdb($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 $ARGV[1]");
my $filename=$ARGV[0];
$filename="-- s/^\.*/";
$filename="-- s/\.pdb//";
# $filename="-- s/.*\.? $qnum ?/ $filename $dir";
$filename="-- s/.*\.? $qnum ?/ $filename $dir";
```

- РНК связанная с белками.

- Длинные РНК.

- Псевдоузлы.

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

```
# 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;
```

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

Моделирование структуры

```

#(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 %
my %qwa=find_quart( $coo
```

```

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

```

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

```

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

```

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

```

foreach my $res
```

```

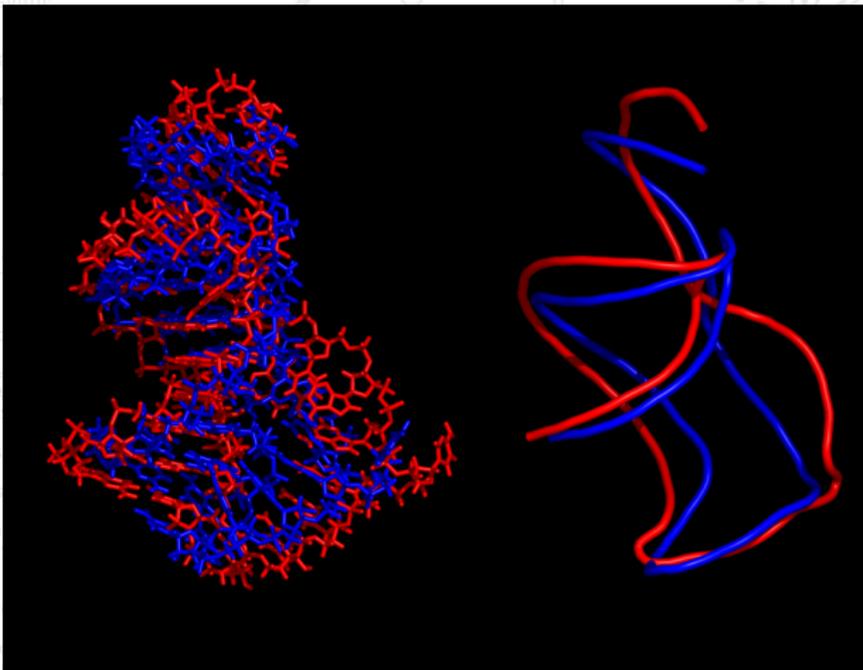
# print "$q $
```

```

$nx=$nx+ $coo
$ny=$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 $schnum)=read_pdb($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 $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^\.*/;
$filename=~ s/\.pdb//;
#$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}{$res}{"$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;
}
}
}
}

```