

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

```
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{"0"}}){ my $ggg=substr($r,0,1); if ( $ggg ne $sch/$schnum+; ;$sch=$ggg ) ;
```

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

```
}
```

# Строение белков

## Структура и организация

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

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

## Введение

```

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

## Уровни организации структуры белка

## Типы взаимодействий в белках

```

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

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

## PDB

```

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

## Визуализация с PyMol

```

# 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;
```

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

# Структура каталитического антитела

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

```
my %coor=read_pdb($coor);
my $dir=$ARGV[1];
my $sch,$my $schnum;
foreach my $f ( sort keys %coor ) {
    my %qwa=find_quartets($coor{$f});
```

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

```
foreach my $m (sort keys %qwa) {
    my %qartets = %qwa{$m};
    my %q = find_q($qartets);
```

```
# 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 $m (keys %q) {
```

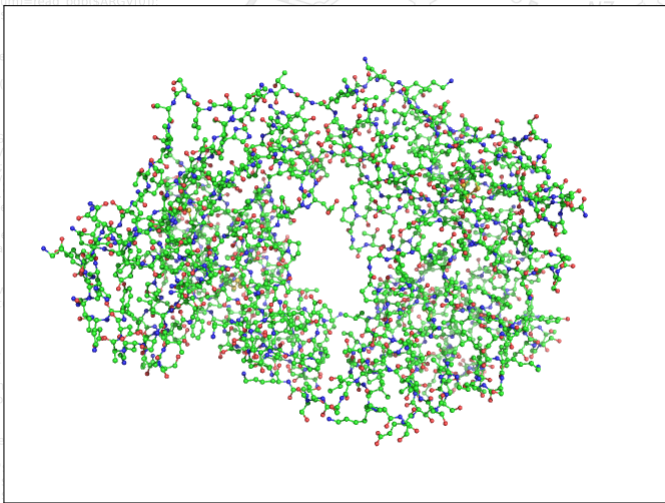
```
# print "Q: $q\n";
```

```
$nx=$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;
```

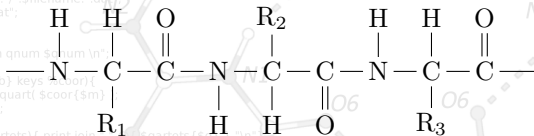
```
$oy=$oy+$coor{$m}{$res}{"O6"}->y;
```

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



# Что такое белок?

**Белки** — высокомолекулярные органические вещества, состоящие из соединённых в цепочку пептидной связью альфа-аминокислот. (wikipedia)



**Или:** белок это линейный полярный полимер, где мономерами является выборка из примерно 20 L-альфа-аминокислот.



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

# Что такое L альфа-аминокислота?

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

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

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

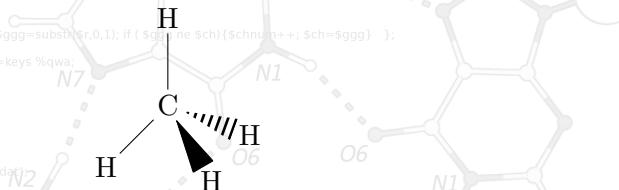
```
foreach my $q ( keys %qartets ){
```

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

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

```
# print "$q %coor{$m}";
$nx=$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;
```



атом углерода в  $sp^3$  гибридизации имеет тетраэдрическое окружение

L-аминокислота

D-аминокислота

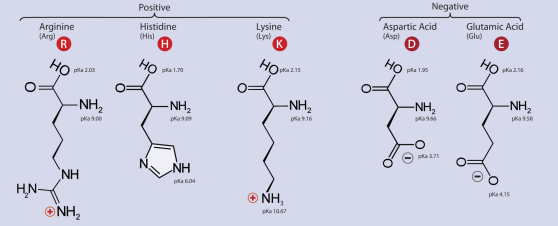
# АМИНОКИСЛОТЫ



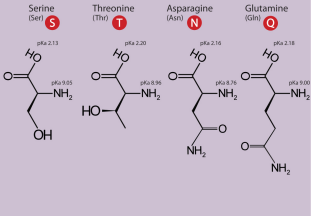
## Twenty-One Amino Acids

⊕ Positive      ⊖ Negative  
 • Side chain charge at physiological pH 7.4

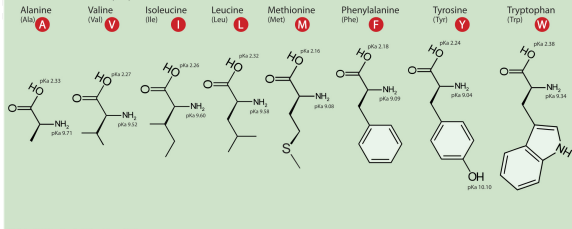
### A. Amino Acids with Electrically Charged Side Chains



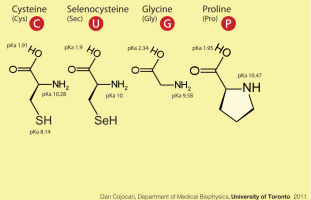
### B. Amino Acids with Polar Uncharged Side Chains



### D. Amino Acids with Hydrophobic Side Chain



### C. Special Cases



pKa Data: CRC Handbook of Chemistry, v. 2010

Dan Coopers, Department of Medical Biophysics, University of Toronto 2011

$\$ox=\$ox+ \$coor\{\$m\}\{\$res\}\{^{\circ}O6^{\circ}\}\rightarrow x;$   
 $\$oy=\$oy+ \$coor\{\$m\}\{\$res\}\{^{\circ}O6^{\circ}\}\rightarrow y;$   
 $\$oz=\$oz+ \$coor\{\$m\}\{\$res\}\{^{\circ}O6^{\circ}\}\rightarrow z;$



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

# Пептидная связь

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

```
my %qwa=find
```

```
if ($qnum > 0) {
#system("mkdir");
my $filename=$ARGV[0];
$filename="-- $";
$filename="-- $";
# $filename=$";
$filename="-- $";
print "$filename";
open OUT,">$filename";
print OUT "#IN";
```

```
foreach my $fr
my %qartet;
my %q= find
```

```
# foreach r
```

```
foreach r
```

```
my $nx;
```

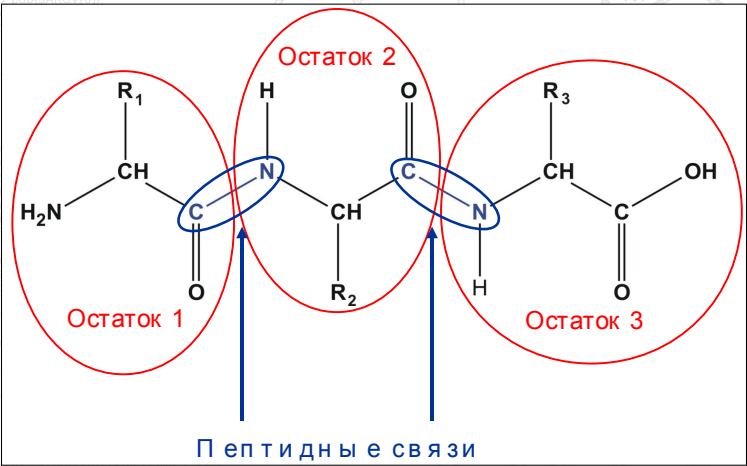
```
my $ox;
```

```
my $y;
```

```
foreach
```

```
#
$nx=$nx+ $coor{$m}{x};
$ny=$ny+ $coor{$m}{y};
$nz=$nz+ $coor{$m}{z};
```

```
$ox=$ox+ $coor{$m}{x};
$oy=$oy+ $coor{$m}{y};
$oz=$oz+ $coor{$m}{z};
```



#!/usr/bin/perl

use Math::VectorReal qw( :all );

# Пептидная связь, таутомерия

```
#!/(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 $dir/$qnum");
```

```
  my $filename=$ARG
```

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

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

```
  # $filename=$schnum."
```

```
  $filename="$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;
```

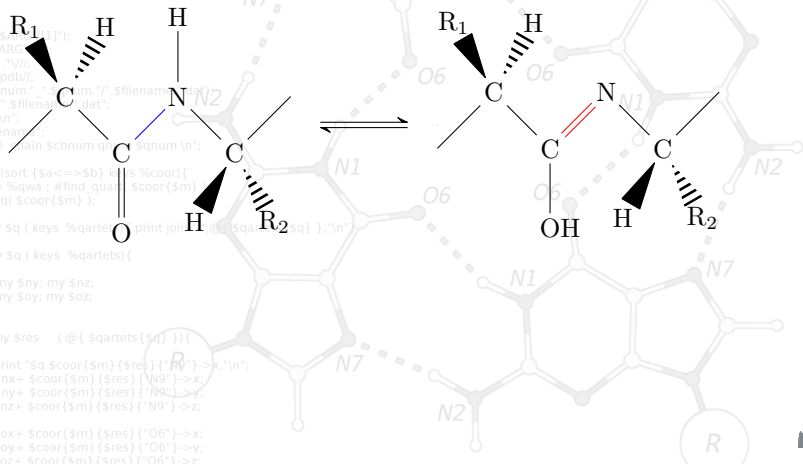
```
        $ny=$ny+ $coor{$m} {$res} {"N9"}->y;
```

```
        $nz=$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;
```



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

# Пептидная связь, свойства

```
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 $sggg=substr($r,0,1); if ( $sggg ne $sch ) { $schnum++; $sch=$sggg } ;
```

```
my %qwa
```

- Пептидная связь прочнее, чем другие амиды

- Атомы пептидного звена (  $C_{\alpha}$ -C-N-  $C_{\alpha}$  ) лежат в одной плоскости

- Валентные углы у атомов C и N примерно равны  $120^{\circ}$

- Вращение вокруг связи C-N затруднено

- Возможны cis- и trans-конфигурации; в белках преобладают trans

- Карбонильный кислород – хороший акцептор водорода

- Амидный азот – хороший донор водорода

```
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 );
```

# Вращения вокруг связей в остове белка

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

```
my %co
```

```
if ($qu
```

```
#system
```

```
my $file
```

```
$file纳
```

```
#$file纳
```

```
$file纳
```

```
print "s
```

```
open O
```

```
print O
```

```
foreach
```

```
my %
```

```
my %
```

```
# fo
```

```
fo
```

```
#
```

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

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

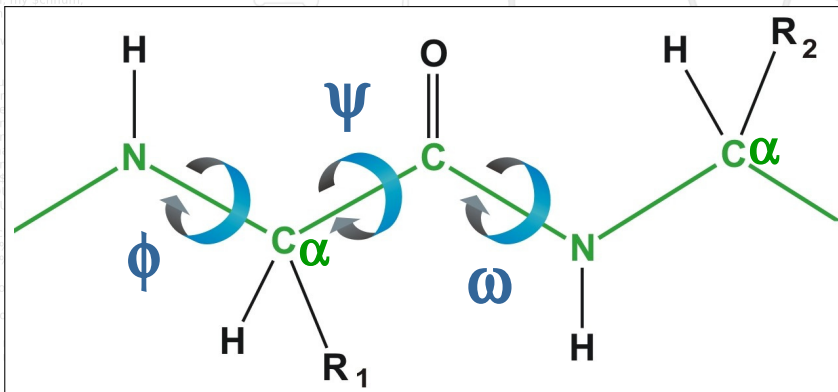
```
$ny=$ny+ $coor{$m}{$res}{\"N9\"}->y;
```

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

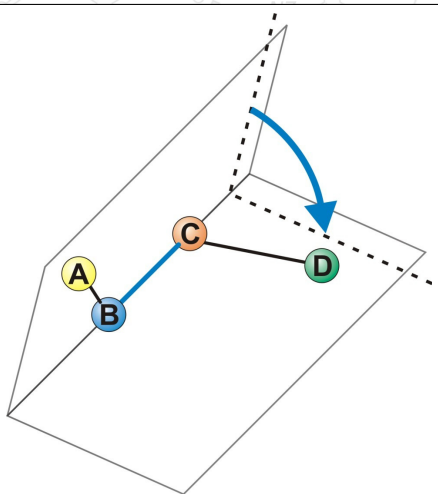
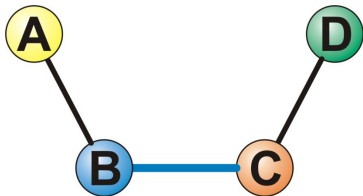
```
$ox=$ox+ $coor{$m}{$res}{\"O6\"}->x;
```

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

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



# Двугранные (торсионные) углы



```
$nz=$nz+ $coor{$m}{$res}{"N"}->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 );
```

# Двугранные (торсионные) углы

```

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

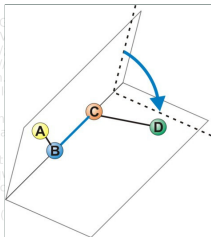
```

Могут принимать значения от  $-180$  до  $+180$

```

if ($sqnum > 0) {
    #system("mkdir $ARGV[0]");
    my $filename=$ARGV[0];
    $filename="-- s/\./\./";
    $filename="-- s/\./\./";
    $filename="-- s/\./\./";
    $filename=$chnum;
    $filename="$dir/".$filename;
    print "$filename\n";
    open OUT,">$filename";
    print OUT "#INFO ch";
}

```



```

foreach my $m ( sort keys %qartets ) {
    my %qartets= %q;
    my %q= find_q( $coor{"0"} );
}

```

```

# foreach my $m ( sort keys %qartets ) {
    foreach my $q ( keys %qartets ) {

```

```

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

```

Построим проекции всех связей на плоскость, перпендикулярную связи B-C

```

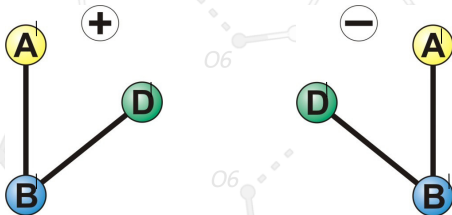
my $res = ( @{ $qartets{$q} } ) {
    print "$q $coor{$m}{$res} {"N9"}->x,"n";
    $nx=$nx+ $coor{$m}{$res} {"N9"}->x;
    $ny=$ny+ $coor{$m}{$res} {"N9"}->y;
    $nz=$nz+ $coor{$m}{$res} {"N9"}->z;
}

```

```

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

```



В D совмещается с B A поворотом против часовой стрелки

В D совмещается с B A поворотом по часовой стрелке





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

# Карта Рамачандрана

даже в полиглициновой цепи существуют стерические ограничения

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

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

```
my %qwa=fin
```

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

```
#system("mk
```

```
my $filename
```

```
$filename=-
```

```
#$filename=-
```

```
$filename=$
```

```
$filename="";
```

```
print "$filen
```

```
open OUT,">$
```

```
print OUT "#R
```

```
foreach my $
```

```
my %qarte
```

```
my %q= fin
```

```
# foreach
```

```
foreach
```

```
my $n
```

```
my $o
```

```
my $
```

```
foreach
```

```
$nx
```

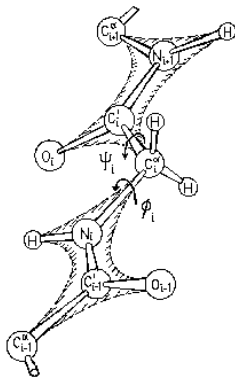
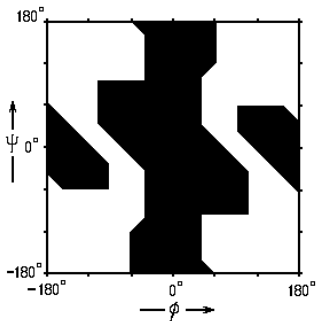
```
$ny
```

```
$nz
```

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

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

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



Карта запрещенных (■) конформаций глицина

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

# Уровни организации структуры белка

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

- Первичная структура
- Вторичная структура
- Укладка (fold)
- Третичная структура
- Четвертичная структура

```
# if ($qnum > 0){
#system("cat my $filename | perl $filename -- s / C / V / G /");
# $filename = " $idir / $filename . dat ";
print "file $filename\n";
open OUT, ">$filename";
print OUT "#INFO chain $chain qnum $qnum\n";
foreach my $m ( keys %$qnum ){
my %$qartets = %$qwa; #find_quart( %$coor{"$m"} );
my %$q = find_of( %$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;
```

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

# Первичная структура

```
#!/(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=schnum . $qnum . "/" . $filename . ".dat";
$filename="$dir"."$filename"."dat";
print "$filename\n";
open OUT,">$filename";
```

Первичная структура – это аминокислотная последовательность:

Met-Ala-Gly-Trp-Ala-Val-Asp ...

```
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;
```

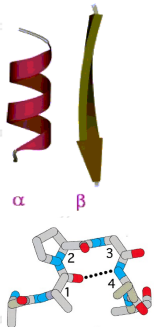
# Вторичная структура

## Вторичная структура

**белка** - это упорядоченные расположения атомов основной цепи полипептида, безотносительно к типам боковых цепей (групп) и их конформациям.

Если упорядоченность такова, что двугранные углы одинаковы у всех остатков, то говорят о регулярной вторичной структуре. Регулярными вторичными структурами являются спирали и  $\beta$ -структуры.

Пример нерегулярной вторичной структуры  $\beta$ -поворот ( $\beta$ -изгиб, реверсивный поворот).



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

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

foreach my $r ( sort keys %{$coor} ) {
    my ($sggg=substr($r,0,1); if ( $sggg ne $ch) { $chnum++; $ch=$sggg } );
    if ($chnum > 0) {
        my ($s1=$coor{$r}; my ($s2=$coor{$r});
        my $filename="-- s/" . $r . ".v";
        $filename="-- s/" . $r . ".pdb";
        # $filename=$chnum . " . $qnum . "/" . $filename . ".dat";
        $filename=$dir . $filename . ".dat";
        print OUT "#INFO chain $chnum group $qnum\n";
        foreach my $m ( sort (keys %>sb) keys %coor) {
            foreach my $q ( keys %qartets) {
                my $nx; my $ny; my $nz;
                my $ox; my $oy; my $oz;
                $nx=$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

Любую регулярную структуру можно представить в виде линейной группы, т.е. спирали. Спираль можно описать с помощью следующих параметров:

$d$  – смещение вдоль оси, в расчете на 1 элемент (атом  $C_{\alpha}$ ),

$r$  – расстояние от  $C_{\alpha}$ -атома до оси,

$n$  – число элементов на один виток спирали, хиральность (направление закрутки) определяется знаком, «+» - право закрученные структуры, «-» – левозакрученные.

/Г.Шульц, Р.Ширмер «Принципы структурной организации белков»/

foreach my \$m (sort { \$a &lt;=&gt; \$b } keys %qartets);

my %qartets = %qwa; #find quartets

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 ( @{ \$qart

# print "\$q \$coor{\$m} (

\$nx=\$nx+ \$coor{\$m} {\$res

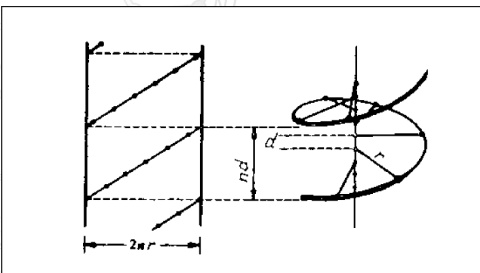
\$ny=\$ny+ \$coor{\$m} {\$res

\$nz=\$nz+ \$coor{\$m} {\$res

\$ox=\$ox+ \$coor{\$m} {\$res

\$oy=\$oy+ \$coor{\$m} {\$res} {"O6"}-&gt;y;

\$oz=\$oz+ \$coor{\$m} {\$res} {"O6"}-&gt;z;



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

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

```
my %qwa=fin
```

```
if ($qnum > 0)
#system("mkd
my $filename=
$filename="--
$filename="--
# $filename=$
$filename="$
print "$filen
open OUT,">$
print OUT "#I
```

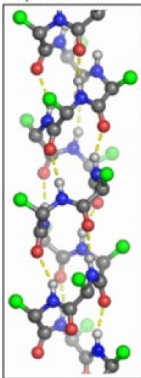
```
foreach my $
my %qarte=
my %q= fin
# foreach
foreach
my $n;
my $o;
my $;
foreach
```

```
# 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;

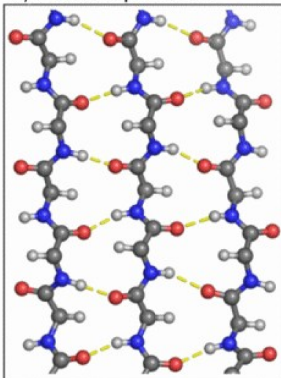
$ox=$ox+ $coor{$m}{ $res }{"O6"}->x;
$oy=$oy+ $coor{$m}{ $res }{"O6"}->y;
$oz=$oz+ $coor{$m}{ $res }{"O6"}->z;
```

# Регулярные вторичные структуры

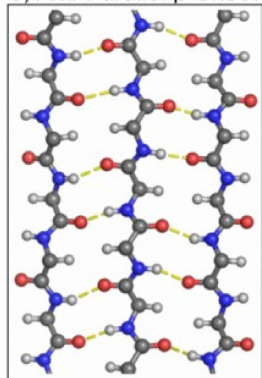
a)  $\alpha$ -Helix



b) Parallel  $\beta$ -Sheet



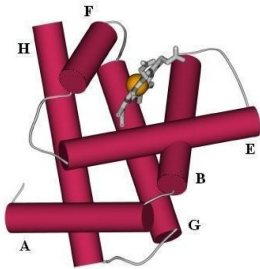
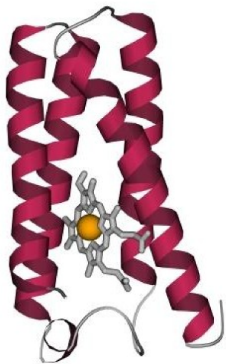
c) Anti-Parallel  $\beta$ -Sheet



# Укладка (fold)

Укладкой называют организацию в пространстве элементов регулярной вторичной структуры.

Пример:  $\alpha$ -спиральные белки



```

if ($qn
#syste
my $fil
$sfilena
$sfilena
#sfilen
$sfilena
print
open C
print C

forea
my
my

#
#

```

```

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

```



# β структурные белки

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
use Math::PlanePath;
my $stb = "beta";

my ($m, $c) = read_pdb($ARGV[0]);
my $coor = read_pdb($ARGV[0]);
my $dir = $ARGV[1];
my $sch, my $sc;
foreach my $r
```

```
my %qwa = fin;
```

```
if ($sqnum > 0)
#system("mkc
my $filename=
$filename=-:
$filename=-:
#$filename=$
$filename="$
print "$filena
open OUT,">$
print OUT "#|
```

```
foreach my $
my %qarte
my %q = fi

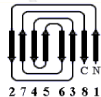
# foreach
foreach
my $n
my $o
my :
foreach
```

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

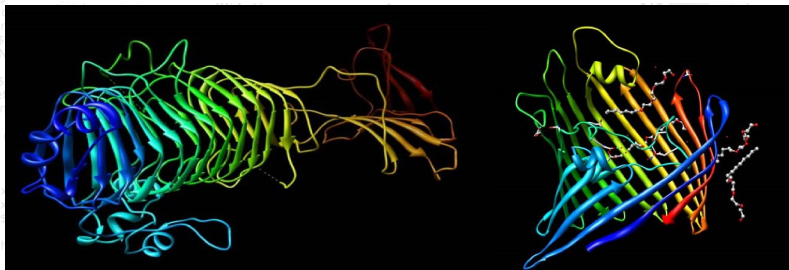
$ox=$ox+ $coor($m) ($res) {"O6"}->x;
$oy=$oy+ $coor($m) ($res) {"O6"}->y;
$oz=$oz+ $coor($m) ($res) {"O6"}->z;
```



ГРЕЧЕСКИЙ КЛЮЧ



РУЛЕТ



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

# Распределение в природе

```

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

```

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

```

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 ) {
```

```

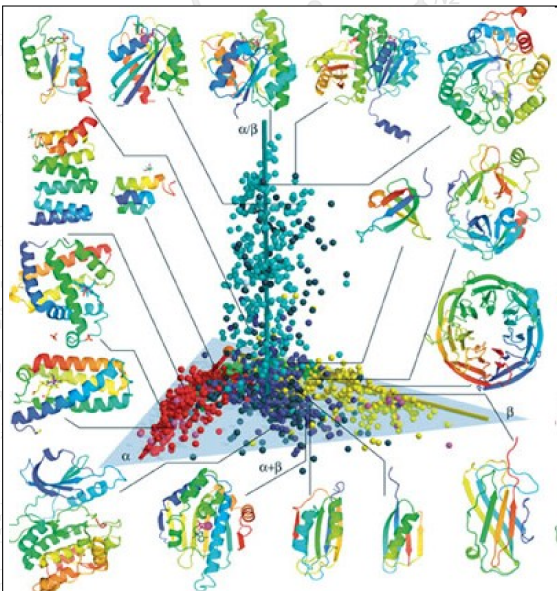
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```

foreach my $res ( @($coor{$m}{ $q } ) ) {
# print "$q $coor{ $m }{ $q }{ $res }";
$nx=$nx+ $coor{ $m }{ $q }{ $res }{ $x };
$ny=$ny+ $coor{ $m }{ $q }{ $res }{ $y };
$nz=$nz+ $coor{ $m }{ $q }{ $res }{ $z };
```

```

$ox=$ox+ $coor{ $m }{ $q }{ $res }{ $x };
$oy=$oy+ $coor{ $m }{ $q }{ $res }{ $y };
$oz=$oz+ $coor{ $m }{ $q }{ $res }{ $z };
```



```
#!/usr/bin/perl
```

```
use Math::VectorReal qw( :all );
```

# Третичная структура

```
#!/(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{"O"}} ){ my $ggg=substr($r,0,1); if ( $ggg ne $sch ){ $schnum++; $sch=$ggg };
```

Третичной структурой называют расположение в пространстве всех атомов одной полипептидной цепи.

Т.е. описание третичной структуры включает в себя:

- описание элементов вторичной структуры,
- описание типа укладки,
- описание структуры петель,
- описание конформаций боковых групп всех аминокислотных остатков.

```
# foreach my $a ( keys %$cartets ){ my $a1=substr($a,0,1);
```

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

```
my $q1=substr($q,0,1);
```

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

```
my $r;
```

```
foreach my $res ( @{$cartets{$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 );
```

# Вспомогательные взаимодействия: водородные СВЯЗИ

```
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 %my $coor ) {
    my %qwa=find_quartet($r, %my $coor);
```

```
if ($qnum > 0) {
    #system("mkdir $ARGV
    my $filename=$ARGV
    $filename=~ s/^.*\//;
    $filename=~ s/\..*/;
    # $filename=$schnum
    $filename="$mdir".$fil
    print "$filename\n";
    open OUT,">$filename
    print OUT "#INFO cha
```

```
foreach my $m (sort
    my %qartets = %qwa
    my %q = find_q($m, %my $coor);
```

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

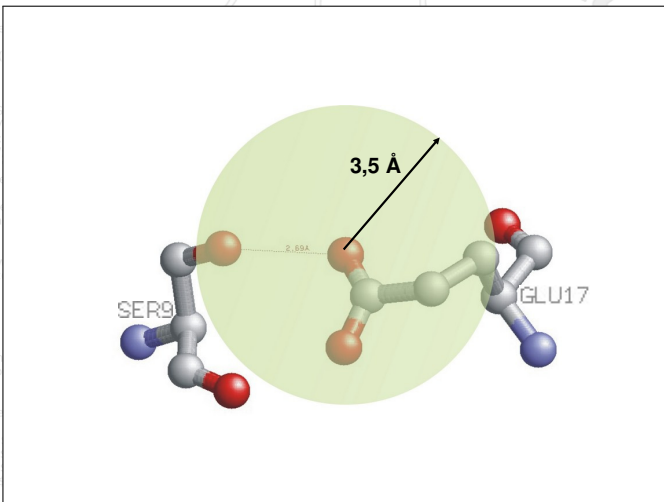
```
my $nx; my $ny;
my $ox; my $oy;
my $r;
```

```
foreach my $r (
    print "
```

```
$nx=$nx+
$ny=$ny+
$nz=$nz+

```

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



# Гидрофобные взаимодействия – главный фактор, заставляющий глобулу свертываться

```
#!/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 $r ( sort keys %coor ) {
    my %qwa=find_quartets($r,%coor);
}
```

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

```
foreach my $m ( sort keys %qwa ) {
    my %qartets = %qwa{$m};
    my %q = find_quartets($m,%coor);
}
```

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

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

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

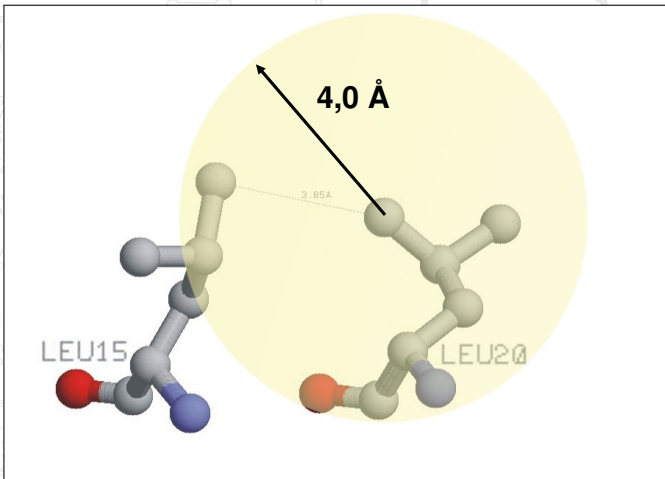
```
foreach my $r ( keys %coor ) {
```

```
# print "

```

```
$nx=$nx+
$ny=$ny+
$nz=$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 );
```

# Ионные пары

```
#!/my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sch, my $chnum;
foreach my $r ( sort keys %coor ) {
    my %qwa=find_quartet($r,%coor);
```

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

```
foreach my $m (sort keys %qwa) {
    my %qartets = %qwa{$m};
    my %q = find_q($m,%coor);
```

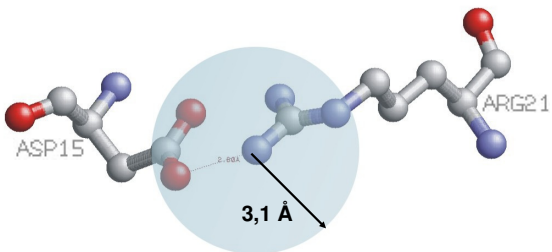
```
# foreach my $q (keys %qartets) {
    foreach my $q (keys %qartets) {
```

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

```
foreach my $res (keys %coor) {
    my $x, my $y, my $z;
    my $ox, my $oy, my $oz;
```

```
# print "
    $nx=$nx+
    $ny=$ny+
    $nz=$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;
```



# Дисульфидные мостики характерны для секреторируемых белков

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
use IO::File;
my $dir = "data";
my $scn = "1p2a.pdb";
my $coor = read_pdb($SARGV[0]);
my $dir = $SARGV[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 } };
my %qwa = find_quart( $coor{"0"} ); my $sqnum = keys %qwa;
```

```
if ($sqnum > 0) {
#system("mkdir $SARGV[1]");
my $filename = $SARGV[0];
my $filename = "s/~/";
my $filename = "s/~/";
my $filename = $schnum;
my $filename = "$dir/";
print "$filename\n";
open OUT, ">$filename";
print OUT "#INFO cha
```

```
foreach my $m ( sort keys %qwa ) {
my %qartets = %qwa{$m};
my %q = find_q($m);
```

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

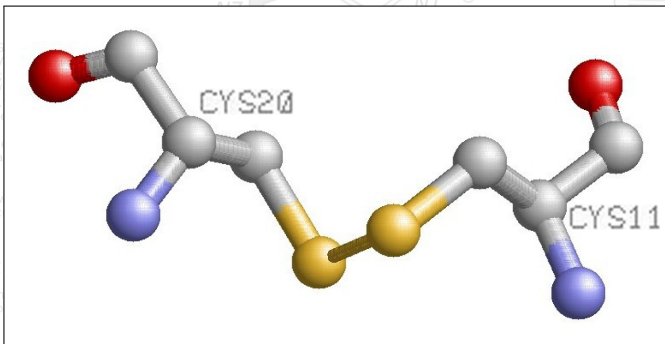
```
foreach my $q ( keys %q ) {
```

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

```
foreach my $res ( keys %{$coor{"0"}} ) {
```

```
# print "$q $coor{$m}{$res} {"N7"}->x,"n";
my $nx = $nx + $coor{$m}{$res}{"N7"}->x;
my $ny = $ny + $coor{$m}{$res}{"N7"}->y;
my $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;
```



```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my $f1 = read_pdb($ARGV[0]);
my $f2 = read_pdb($ARGV[1]);
```

# От четвертичной структуры к молекулярным

## машинам

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

```
my %qwa
```

```
if ($qnum)
#system
my $f1len
my $f2len
my $f1name
my $f2name
my $f1name
my $f2name
my $f1name
my $f2name
print "$f1name"
open OUT
print OUT
```

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

```
# for
```

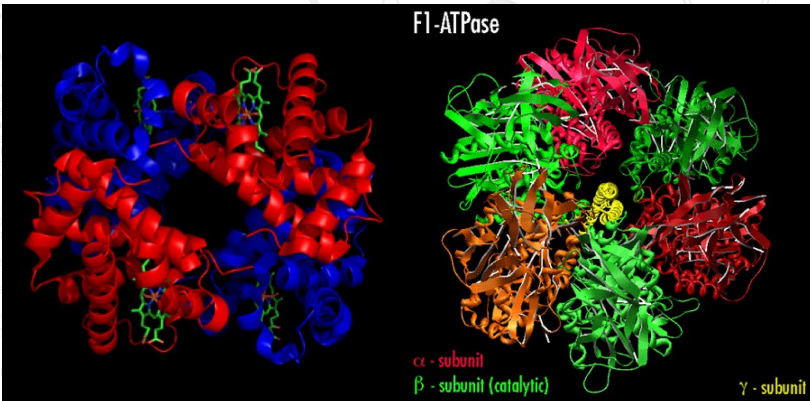
```
for
```

```
my
```

```
my
```

```
for
```

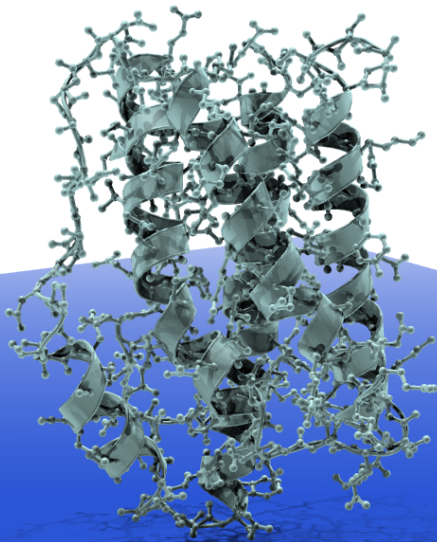
```
#
```



```
$nx=$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;
```





Визуализация третичных структур, PyMol

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

# Основные этапы расшифровки 3D-структуры

## Эксперимент

```
my $pdb=$ARGV[0];
my $mdir=$ARGV[1];
my $ch_ my $chnum;
foreach
```

```
my %c
```

```
if ($qn
```

```
#system
```

```
my $fil
```

```
$filena
```

```
$filena
```

```
#filer
```

```
$filena
```

```
print
```

```
open C
```

```
print C
```

```
foreach
```

```
my
```

```
my
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

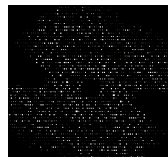
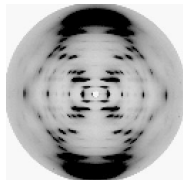
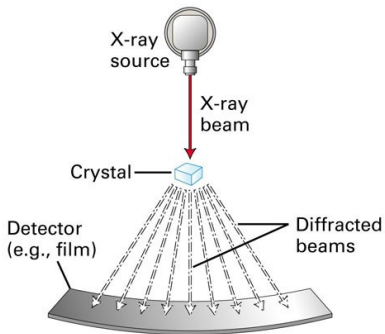
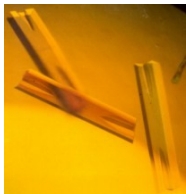
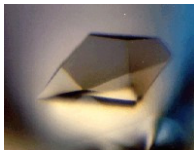
```
#
```

```
#
```

```
#
```

```
#
```

```
#
```



```
$nz=$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 );
```

# Основные этапы расшифровки 3D-структуры

## Вычисления и моделирование

```
my %ccoor=read_pdb($ARGV(0));
my $mdir=$ARGV[1];
my $cmr=$ARGV[2];
foreach my $atom ($ccoor{ATOM}){
```

```
my %c
```

```
if ($qm
```

```
#system
```

```
my $f
```

```
$flena
```

```
$flena
```

```
#fller
```

```
$flena
```

```
print
```

```
open C
```

```
print C
```

```
forea
```

```
my
```

```
my
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#
```

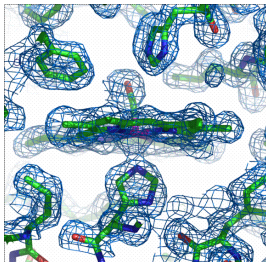
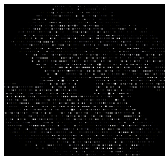
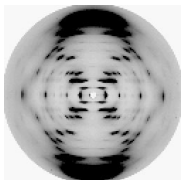
```
#
```

```
#
```

```
#
```

```
#
```

```
#
```



ATOM	188	N	LYS	A	27	-35.123	77.795	11.208
ATOM	189	CA	LYS	A	27	-35.949	78.814	11.849
ATOM	190	C	LYS	A	27	-36.907	78.113	12.808
ATOM	191	O	LYS	A	27	-37.292	76.978	12.588
ATOM	192	CB	LYS	A	27	-36.706	79.638	10.758
ATOM	193	CG	LYS	A	27	-37.635	80.765	11.277
ATOM	194	CD	LYS	A	27	-37.718	82.122	10.499
ATOM	195	CE	LYS	A	27	-38.234	82.152	9.034
ATOM	196	NZ	LYS	A	27	-38.418	83.510	8.535
ATOM	197	N	GLY	A	28	-37.102	78.672	13.993
ATOM	198	CA	GLY	A	28	-38.068	78.114	14.937
ATOM	199	C	GLY	A	28	-37.531	77.128	15.942
ATOM	200	O	GLY	A	28	-36.321	76.902	16.023
ATOM	201	N	LEU	A	29	-38.399	76.581	16.790
ATOM	202	CA	LEU	A	29	-38.000	75.559	17.744
ATOM	203	C	LEU	A	29	-37.661	74.380	16.879
ATOM	204	O	LEU	A	29	-38.320	74.195	15.864
ATOM	205	CB	LEU	A	29	-39.135	75.187	18.665
ATOM	206	CG	LEU	A	29	-38.918	75.630	20.108
ATOM	207	CD1	LEU	A	29	-40.273	75.849	20.760
ATOM	208	CD2	LEU	A	29	-38.051	74.608	20.847

```
$nz=$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;
```

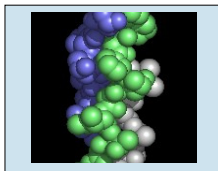
```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
```

# Основные этапы расшифровки 3D-структуры

## Анализ

```
my $hnum=read_pdb($ARGV[0]);
my $model=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $c;
foreach
```

```
ATOM 188 N LYS A 27 -35.123 77.795 11.208
ATOM 189 CA LYS A 27 -35.949 78.814 11.849
ATOM 190 C LYS A 27 -36.907 78.113 12.808
ATOM 191 O LYS A 27 -37.292 76.978 12.588
ATOM 192 CB LYS A 27 -36.706 79.638 10.758
ATOM 193 CG LYS A 27 -37.635 80.765 11.277
ATOM 194 CD LYS A 27 -37.718 82.122 10.499
ATOM 195 CE LYS A 27 -38.234 82.152 9.034
ATOM 196 NZ LYS A 27 -38.418 83.510 8.535
ATOM 197 N GLY A 28 -37.102 78.672 13.993
ATOM 198 CA GLY A 28 -38.068 78.114 14.937
ATOM 199 C GLY A 28 -37.531 77.128 15.942
ATOM 200 O GLY A 28 -36.321 76.902 16.023
ATOM 201 N LEU A 29 -38.399 76.581 16.790
ATOM 202 CA LEU A 29 -38.000 75.559 17.744
ATOM 203 C LEU A 29 -37.661 74.380 16.879
ATOM 204 O LEU A 29 -38.320 74.195 15.864
ATOM 205 CB LEU A 29 -39.135 75.187 18.665
ATOM 206 CG LEU A 29 -38.918 75.630 20.108
ATOM 207 CD1 LEU A 29 -40.273 75.849 20.760
ATOM 208 CD2 LEU A 29 -38.051 74.608 20.847
```



**Программы для визуализации и анализа 3D —  
RasMol, PyMol, SPDBViewer, WebMol, .....**

```
$nz=$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 );
```

# Брукхейвенский банк пространственных структур

## (PDB)

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

```
my %qv
```

```
if ($qu
```

```
#system
```

```
my $file
```

```
$filenam
```

```
$filenam
```

```
$filenam
```

```
print "s
```

```
open OU
```

```
print OU
```

```
foreach
```

```
my %
```

```
my %
```

```
# fo
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

```
fo
```

```
#
```

The screenshot displays the RCSB Protein Data Bank (PDB) website. At the top, the PDB logo is prominent, along with the text 'A MEMBER OF THE PDB' and 'An Information Portal to Biological Macromolecular Structures'. The page is dated 'Tuesday Sep 04, 2012 at 5 PM PDT' and shows '84381 Structures'. Navigation options include 'All Categories', 'Author', 'Macromolecule', 'Sequence', and 'Ligand'. A search bar is present with the text 'e.g., PDB ID, molecule name, author'. The main content area is titled 'Biological Macromolecular Resource' and includes a 'Full Description' section. Under 'Featured Molecules', there is a 'Structural View of Biology' section with a 'List View of Archive By: Title | Date | Category'. The 'Molecule of the Month' is the 'Pyruvate Dehydrogenase Complex', described as a combination of crystallography, NMR spectroscopy, and electron microscopy revealing the secrets of the pyruvate dehydrogenase complex. The 'Full Article' section discusses the Protein Structure Initiative Featured System 'Solute Channels'. The right sidebar contains sections for 'New Structures', 'New Features', 'RCSB PDB News', and 'Explore Archive'. The URL at the bottom is 'http://www.pdb.org/pdb/home/home.do < 128/281 All'.

```
$ox=$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 Math::Tetrahedron;
use Math::Tetrahedron;
use Math::Tetrahedron;
```

# Что хранится в PDB?

```
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 $fr ( sort keys %{$coor{"0"}} ){ my $sggg=substr($fr,0,1); if ( $sggg ne $sch ){ $schnum++; $sch=$sggg } };
my %qv;
```

```
if ($sqr
#system
my $file
$file
$file
$file
$file
print "s
open OU
print OU
foreach
my %
my %
```

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

```
foreach my $res ( @{$ $partets{$sq} } ){
#
print "$q $coor{$m} {$res} {" "N" -> x, "n";
$nx=$nx+ $coor{$m} {$res} {" "N" -> x;
$ny=$ny+ $coor{$m} {$res} {" "N" -> y;
$nz=$nz+ $coor{$m} {$res} {" "N" -> z;
$ox=$ox+ $coor{$m} {$res} {" "O" -> x;
$oy=$oy+ $coor{$m} {$res} {" "O" -> y;
$oz=$oz+ $coor{$m} {$res} {" "O" -> z;
```

## An Information Portal to Biological Macromolecular Structures

As of Tuesday Nov 10, 2009 at 4 PM PST there are 61418 Structures [?](#) | [PDB Statistics](#) [?](#)

**PDB**  
PROTEIN DATA BANK

[WHAT'S NEW](#) | [HELP](#) | [PRINT](#)




### PDB Current Holdings Breakdown

	Proteins	Nucleic Acids	Protein/NA Complexes	Other	Total
X-ray	49409	1176	2289	17	52891
NMR	7082	874	150	6	8112
<b>Exp. Method</b>	Electron Microscopy	175	16	66	257
	Hybrid	18	1	1	21
	Other	116	4	4	13
<b>Total</b>	<b>56800</b>	<b>2071</b>	<b>2510</b>	<b>37</b>	<b>61418</b>







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

# Координаты атомов в PDB-файле

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

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

```
ATOM 1 N THR A 3 16.514 -2.279 12.062 1.00 43.86 N
```

```
ATOM 2 CA THR A 3 17.180 -2.102 13.371 1.00 45.43 C
```

```
ATOM 3 C THR A 3 16.995 -0.675 13.903 1.00 48.26 C
```

```
ATOM 4 O THR A 3 16.888 -0.476 15.109 1.00 56.27 O
```

```
ATOM 5 CB THR A 3 18.658 -2.818 13.510 1.00 61.61 C
```

```
ATOM 6 OG1 THR A 3 18.953 -3.305 14.848 1.00 40.35 O
```

```
ATOM 7 CG2 THR A 3 19.796 -1.970 12.934 1.00 66.69 C
```

```
ATOM 8 N ILE A 4 16.880 0.318 13.028 1.00 35.25 N
```

```
ATOM 9 CA ILE A 4 16.614 1.653 13.545 1.00 31.81 C
```

```
ATOM 10 C ILE A 4 15.180 1.537 14.149 1.00 36.74 C
```

```
ATOM 11 O ILE A 4 14.824 2.204 15.125 1.00 23.77 O
```

```
ATOM 12 CB ILE A 4 16.557 2.686 12.441 1.00 32.25 C
```

```
ATOM 13 CG1 ILE A 4 16.613 4.069 13.040 1.00 32.26 C
```

```
ATOM 14 CG2 ILE A 4 15.242 2.611 11.664 1.00 22.31 C
```

```
ATOM 15 CD1 ILE A 4 16.468 5.127 11.966 1.00 56.11 C
```

```
ATOM 16 N LYS A 5 14.363 0.675 13.544 1.00 41.48 N
```

```
ATOM 17 CA LYS A 5 13.005 0.429 14.018 1.00 40.63 C
```

```
ATOM 18 C LYS A 5 13.126 -0.115 15.426 1.00 43.60 C
```

```
ATOM 19 O LYS A 5 12.360 0.211 16.357 1.00 45.74 O
```

```
ATOM 20 CB LYS A 5 12.399 -0.681 13.198 1.00 41.30 C
```

```
ATOM 21 CG LYS A 5 11.236 -0.268 12.361 1.00 61.61 C
```

```
ATOM 22 CD LYS A 5 11.427 -0.757 10.930 1.00 66.72 C
```

```
ATOM 23 CE LYS A 5 10.112 -0.760 10.137 1.00 90.19 C
```

```
ATOM 24 NZ LYS A 5 10.059 -1.825 9.080 1.00 69.42 N
```

```
ATOM 25 N ASP A 6 14.102 -0.973 15.597 1.00 38.66 N
```

```
my $nx=$coor{$m}{$sres}{"N9"}->x;
```

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

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

```
my $sox=$coor{$m}{$sres}{"O6"}->x;
```

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

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

```
my $sox=$coor{$m}{$sres}{"O6"}->x;
```

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

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

```
my $sox=$coor{$m}{$sres}{"O6"}->x;
```

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

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

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

# Визуализация с PyMol

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

```
my $my $coor=read_pdb($
my $mdir=$ARGV[1];
my $sch, my $schnum;
foreach my $f ( sort key
```

```
my %qwa=find_quart( :
```

```
if ($sqnum >0){
#system("mkdir $ARGV
my $filename=$ARGV[
$filename="-- s/^~/V//;
$filename="-- s/\.pdb//;
# $filename=$schnum."
$filename="-- $mdir". $filei
print "$filename\n";
open OUT,">$filename
print OUT "#INFO chait
```

```
foreach my $m (sort {
my %qartets= %qwa
my %q= find_qf $co
```

```
# foreach my $q ( k
```

```
foreach my $q ( k
```

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

```
foreach my $res
```

```
# print "$
```

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

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

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

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

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

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



# PyMol

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

```
my $cmd = "open read_rdtbi@ARGV[0]";
```

```
my $dir = $cmd;
```

```
my $sch = $cmd;
```

```
foreach
```

```
my %qw
```

```
if ($qu
```

```
#system
```

```
my $file
```

```
$filename
```

```
$filename
```

```
#$filename
```

```
$filename
```

```
print "s
```

```
open OU
```

```
print OU
```

```
foreach
```

```
my %
```

```
my %
```

```
# fo
```

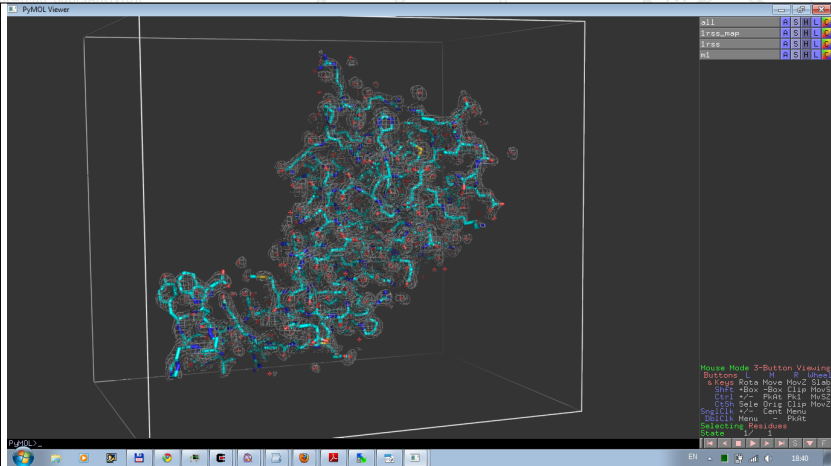
```
fo
```

```
#
```

```
#
```

```
#
```

```
#
```



```
$ny=$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;
```

# Для чего нужен PyMol

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

```
  $filename=$schnum . $filename;
```

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

```
  #system("cp $filename $dir");
```

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

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

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

```
  foreach my $m (sort {$a<=>$b} keys %{$coor{"0"}}){
```

```
    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;
```

- Визуализация pdb и прочих файлов с координатами атомов
- Изготовление высококачественных изображений
- Начальное редактирование структур

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

# Системные требования

```

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

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

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

**Компьютер:** чем мощнее процессор и чем больше памяти, тем лучше

**3D монитор** не обязателен, но поддерживается

**Операционная система:** любая, под Linux проще установить, и он лучше работает с памятью.

```
# 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;
```

# Как установить?

```
#!/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;
```

- Компиляция из исходников: <http://pymol.svn.sourceforge.net/>

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

- Установка бинарных пакетов в Ubuntu Linux: `sudo apt-get install pymol`

- Установка бинарных пакетов в Windows:

- Ресурс для установки с python:

```
foreach my $m (sort { $coor{$m}{x} < $coor{$m}{y} } keys %$coor){
my %$qartets= %$qwa{find_quart($coor{$m})};
my %$q= find_qt($qartets);
```

- Компиляция под Windows:

```
# foreach my $s ( @{$qartets{$q}}){
foreach my $s ( @{$qartets{$q}}){
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 );
```

# PyMol - это GPL программа?

```

#(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;
```

Да, PyMol это GPL-программа;

- исходный код доступен на [sourceforge.net](http://sourceforge.net)
- Бинарные пакеты для windows стоят денег и продаются: <http://pymol.org/academic.html>
- Бинарные пакеты для Linux собираются майтенерами

```

# 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;
```