

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

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

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

```
foreach my $m (sort { $a<=>$b } keys %coor){
my %qartets;
my %q= find_q( $coor{$m} );
# foreach my $q { keys %qartets } { print join " ", @{$qartets{$q}}, "\n";
foreach my $q { keys %qartets } {
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
foreach my $res ( @{$qartets{$q}} ){
# print "$q $coor{$m}{$res}{\"N\"->x,\"n\";
$nx=$nx+ $coor{$m}{ $res }{"N9"}->x;
$ny=$ny+ $coor{$m}{ $res }{"N9"}->y;
$nz=$nz+ $coor{$m}{ $res }{"N9"}->z;
$ox=$ox+ $coor{$m}{ $res }{"O6"}->x;
$oy=$oy+ $coor{$m}{ $res }{"O6"}->y;
$oz=$oz+ $coor{$m}{ $res }{"O6"}->z;
$r=$res;
}
}
```

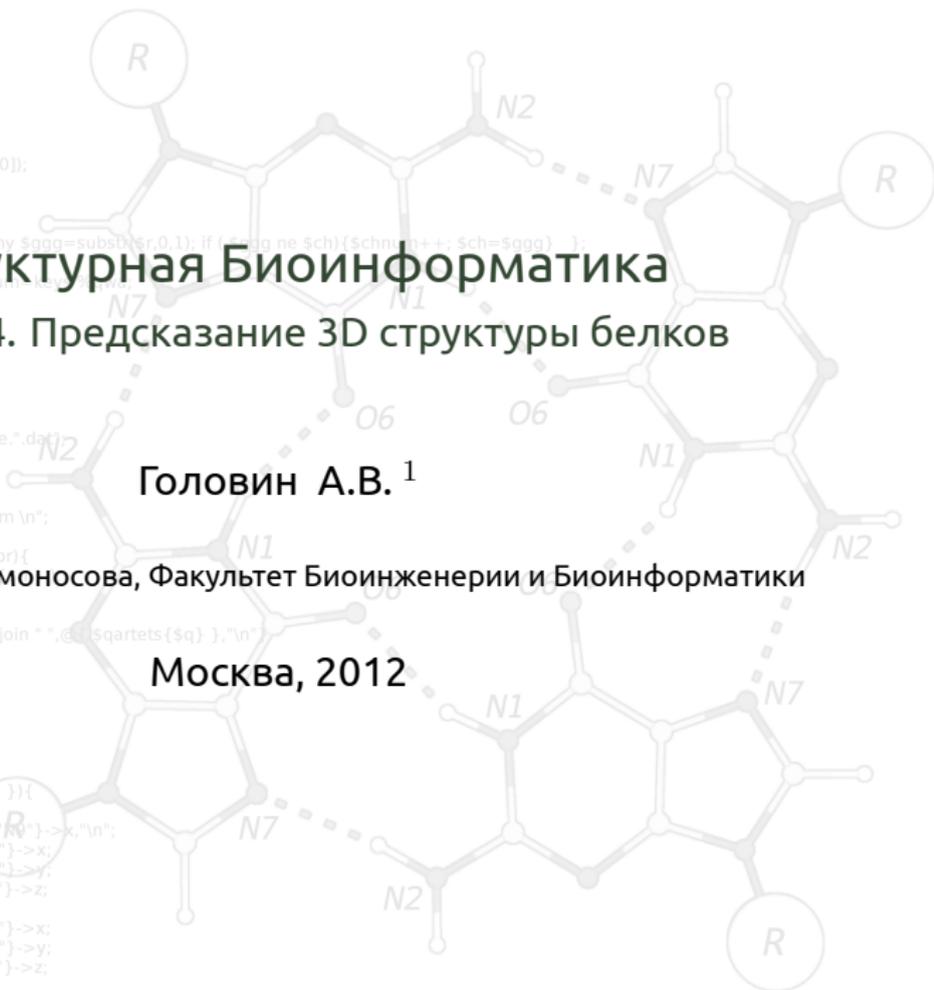
# Структурная Биоинформатика

## Лекция 4. Предсказание 3D структуры белков

Головин А.В.<sup>1</sup>

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

Москва, 2012



# Содержание

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

## Введение

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

## Сравнительное моделирование

```
my %qwa=find_quart( $coor{"0"} ); my $qnum=keys %qwa;
if ( $qnum > 0 ){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/\/\//;
$filename=schrnd( " " . $qnum / $filename . " " );
$filename="dir".$filename.".dat";
print "$filename\n";
open OUT.">$filename";
```

## Моделирование *Ab initio*

```
foreach my $m ( sort { $a<=>$b } keys %coor ){
my %qartets= %qwa ; #find_quart( $coor{ $m } );
my %q= find_q( $coor{ $m } );
print "qartets{ $q } ,\n";
foreach my $q ( keys %qartets ){
my $nx= my $ny= my $nz=
```

## Threading — протягивание нити

```
foreach my $s ( @ { $qartets{ $q } } ){
my $nx= my $ny= my $nz=
my $r;
```

## Распознавание укладки

```
foreach my $res ( @ { $qartets{ $q } } ){
my $nx= my $ny= my $nz=
my $r;
```

## Мета серверы

```
foreach my $res ( @ { $qartets{ $q } } ){
my $nx= my $ny= my $nz=
my $r;
```

## Заключение

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

# Основные проблемы:

```

#(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 } ;

```

- Монте-Карло: 100 а.к.  $3^N$  степеней свободы, получаем  $10^{48}$  конфигураций.

- **Парадокс Левинталя:** "Промежуток времени, за который полипептид приходит к своему скрученному состоянию, на много порядков меньше, чем если бы полипептид просто перебирал все возможные конфигурации".

- Для решения разумно использовать накопленные знания для моделирования.

```

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

# Последовательность-структура

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

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

## Причины парадокса Левинталя:

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

- Теоретические модели, не соответствуют тому, что природа старается оптимизировать;

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

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

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

- В ходе эволюции были отобраны только те белки, которые легко сворачиваются;

```
  $filename="Schnum_".$schnum.".c"; $filename="$filename".$qnum; $filename="Schnum_".$schnum.".c"; $filename="$filename".$qnum;
```

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

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

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

```
  foreach my $q ( keys %{$coor{"O"}} ) {
```

```
    my %qartets= %qwa - #find quartets $coor{$q} ;
```

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

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

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

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

- Считается, что структура определяется последовательностью, но иногда нужны другие факторы.

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

# Сравнительное моделирование

```

#(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;
```

- Зачем искать конформации если можно представить, что при подобии последовательностей подобны и структуры.

```

if ($qnum > 0){
#system("cat $dir/$ch/$qnum/$ch.qwa");
my $filename="$dir/$ch/$qnum/$ch.qwa";
$filename=~ s/~/~/;
$filename=~ s/\././;
#system("cat $dir/$ch/$qnum/$ch.qwa");
$filename="$dir/$ch/$qnum/$ch.qwa";
print "$filename\n";
open OUT ">OUT.$ch.$qnum.$ch.qwa";
print OUT "chain $chnum $qnum $qnum\n";

```

- Надо оценить насколько вероятно, что отличие в последовательности может привести изменению способа укладки цепи.

```

foreach my $m ( keys %coor ){
my %quarts = %qwa; #find_quart( $coor{$m} );
my %q = find_q( $coor{$m} );

```

- Надо отфильтровать ошибки полученные при определении структуры.

```

#
foreach my $s ( keys %quarts ){
my $nx, my $ny, my $nz;
my $ox, my $oy, my $oz;
my $r;

foreach my $res ( @{$quarts{$s}} ){
print "$s $coor{$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 );
```

# Известные структуры и последовательности

```

#(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 %coor{"O"}{ $r } = %coor{"O"}{ $ggg } ;
}

```

- Сейчас известно порядка  $10^5$  структур. Примерно 10% это уникальные белки.

```

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

```

- Только 30% из первого пункта имеют разрешение лучше 3.0 ангстрем.

```

foreach my $r ( sort keys %{$coor{"O"}} ) {
my %qartets = %qwa; #find quart ($coor{$r});
my %qartets = %qwa; #find quart ($coor{$r});
}
# foreach my $q ( keys %qartets ) { print join " ", @qartets{$q} , "\n";
}

```

- Примерно 25% известных последовательностей можно использовать для сравнительного моделирования.
- Для 50% последовательностей можно предсказать способ укладки.

```

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

```

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

# Степень идентичности и сравнительное моделирование

```
my %scoor=read_pdb($ARGV[0]);
my $mdir=$ARGV[1];
my $sch, my $schnum;
foreach my $f ( sort keys %{$scoor{"0"}} ){ my $s
my %qwa=find_quart( $scoor{"0"} ); my $qnum=
```

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

```
my $q1=$schnum.".".$qnum.".".$filename."."
my %q1=find_quart($scoor{"$q1"});
# foreach my $q { keys %qartets } { print join
#
foreach my $q { keys %qartets } {
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ){
#
print "$q $coor{$m}{$res} {"$r"}
$nx=$nx+ $coor{$m}{$res} {"N9"}->
$ny=$ny+ $coor{$m}{$res} {"N9"}->
$nz=$nz+ $coor{$m}{$res} {"N9"}->
```

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

```
my $sx; my $sy; my $sz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ){
#
print "$q $coor{$m}{$res} {"$r"}
$nx=$nx+ $coor{$m}{$res} {"N9"}->
$ny=$ny+ $coor{$m}{$res} {"N9"}->
$nz=$nz+ $coor{$m}{$res} {"N9"}->
```

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

```
my $sx; my $sy; my $sz;
my $ox; my $oy; my $oz;
my $r;
```

```
foreach my $res ( @{$qartets{$q}} ){
#
print "$q $coor{$m}{$res} {"$r"}
$nx=$nx+ $coor{$m}{$res} {"N9"}->
$ny=$ny+ $coor{$m}{$res} {"N9"}->
$nz=$nz+ $coor{$m}{$res} {"N9"}->
```

99 %

- Поиск ингибитора



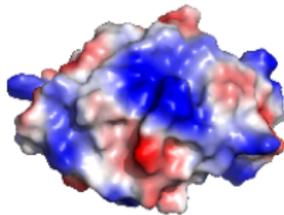
60 %

- Молекулярное замещение в PCA
- Планирование мутагениза



30 %

- Поиск сайтов связывания
- Определение упаковки



0 %

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

# Как это реализовать?

```

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

```

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

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

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  my $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

```
  #system("cat $ARGV[0].pdb > $ARGV[1].pdb");
```

```
  $filename=$ARGV[0].".pdb";
```

```
  $filename="--s/~/V/";
```

```
  $filename="--s/~/V/";
```

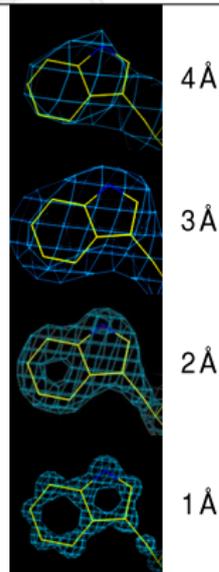
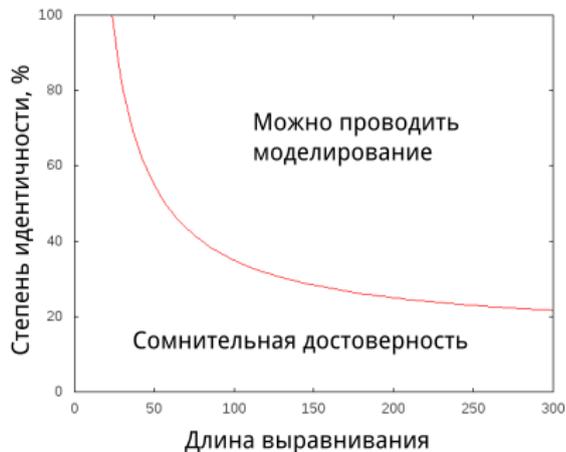
- Надо найти белок заготовку с известной структурой.
- Построить первичное выравнивание.
- Улучшить выравнивание.
- Построить ход основной цепи.
- Моделирование петель
- Достроить/моделировать положение боковых радикалов
- Проверка модели





# Качество белка заготовки

- Выбор качественного белка заготовки очень важен.
- Лучший вариант не обязательно обладает лучшей степенью идентичности.
  - Белок 1: ID 93%, 3.5 ангстрема разрешение. Хуже.
  - Белок 2: ID 90%, 1.5 ангстрема разрешение. Лучше!



# Если структура белка заготовки получена ЯМР

- Определимся какие области определены лучше.
- Соотнесём с выравниванием.
- Если низкая гомология выпадает на “подвижные” области, то структура подходит.

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

#(my %coor=
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $sdir=$dir~/^([a-z]+)/;
foreach my $r ( sort keys %{$coor{0}} ) { (my $ggg=substr($r,0,1); if ($ggg ne $sch){ $schnum++; $sch=$ggg } );
my %a

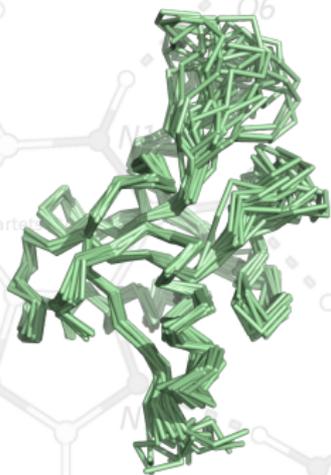
if ($qnum
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename-- s/^\.//;
$filename-- s/\.pdb//;
#$filename=$schnum.".".$qnum.".".$filename.".dat";
$filename="$sdir"."$filename.".dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $schnum qnum $qnum\n";

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

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

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

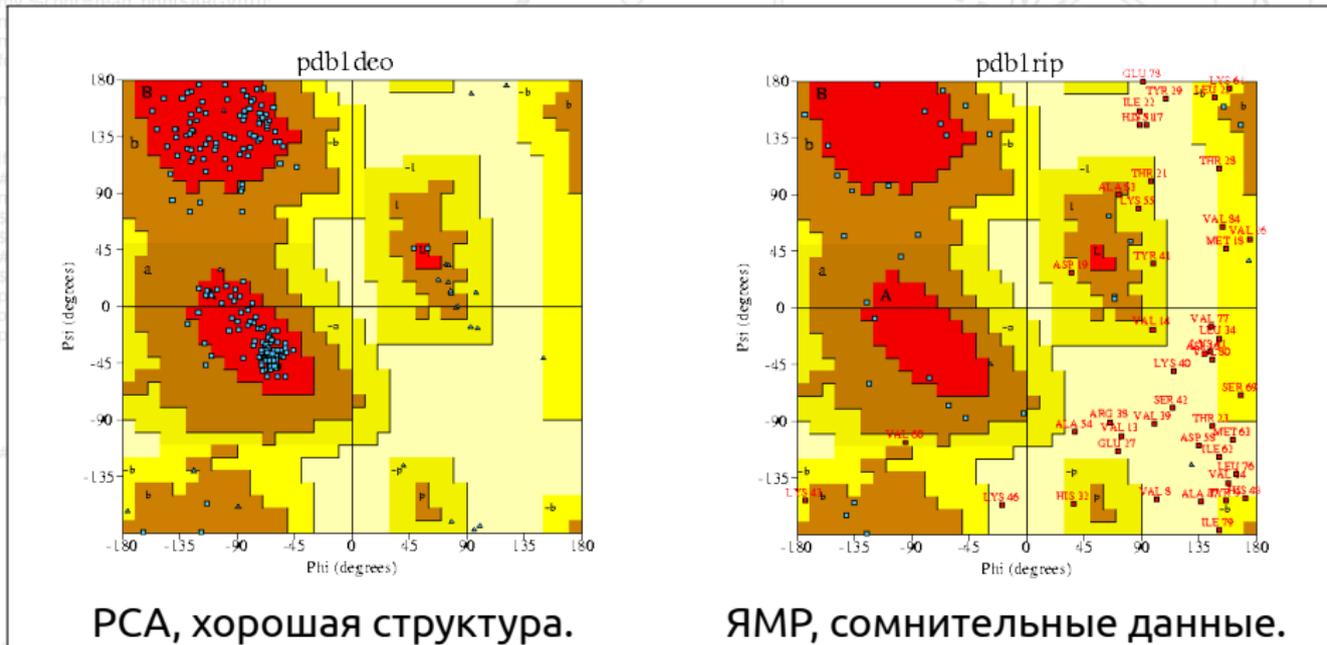
$ox=$ox+ $coor{ $m } { $res } { "O6" }->x;
$oy=$oy+ $coor{ $m } { $res } { "O6" }->y;
$oz=$oz+ $coor{ $m } { $res } { "O6" }->z;
```



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

# Качество заготовки, Рамачандран

```
#[my %$coor,$my $chnum]=read_pdb($ARGV[0]);
my $R=cosor-read_rdb($ARGV[0]);
```



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

# Построение остова

```

#(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;

```

- Генерируем координаты остова моделируемого белка для остатков из выравненных областей.

- Не обязательно использовать координаты, могут подойти дистанционные ограничения.

- Большинство исследователей предпочитают Modeller. Modeller использует дистанционные ограничения.

```

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

```

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

# Моделирование петель

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

• **Эмпирическое моделирование:**

- Поиск подходящего фрагмента по PDB
- Использовать базы данных (LIP, etc..)

```
my %qwa=find_qwa($my $coor, $my $schnum);
```

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

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

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

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

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

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

```
  print "$filename";
```

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

```
  print OUT "m qnum $qnum\n";
```

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

```
    my %qartets;
```

```
    my %q= find_q($my $coor, $my $schnum);
```

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

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

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

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

```
        my $r;
```

• **Комбинации выше перечисленных.**

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

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

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

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

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

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

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

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

# Моделирование боковых радикалов

```

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

```

- Если идентичность последовательностей высока то можно ожидать высокую консервативность третичных контактов.
- Если анализ показывает, что важные контакты консервативны то:

**Лучше оставить конформацию боковых радикалов из заготовки чем моделировать.**

```

foreach my $m (sort { $a<=>$b } keys %coor ){
  my %qartets = %qwa ; #find quart( $coor{$m} );
  #
  foreach my $i ( keys %qartets ){
    print "Sq $coor{$m}{ $res } {" $i }->x,"n";
    $nx=$nx+ $coor{$m}{ $res } {" $i }->x;
    $ny=$ny+ $coor{$m}{ $res } {" $i }->y;
    $nz=$nz+ $coor{$m}{ $res } {" $i }->z;
  }
  $ox=$ox+ $coor{$m}{ $res } {" $i }->x;
  $oy=$oy+ $coor{$m}{ $res } {" $i }->y;
  $oz=$oz+ $coor{$m}{ $res } {" $i }->z;
}

```

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

# Моделирование боковых радикалов

```

#(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;
```

- Конформация боковых радикалов зависит от конформации основной цепи.
- Существуют базы данных ротамеров.
- Некоторые исследователи считают, что SCWRL метод самый удачный.

Это эмпирический метод на основе теории графов.

<http://dunbrack.fccc.edu/SCWRL3.php>

```

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

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

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

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

# Точность моделирования боковых радикалов

```

#(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;
```

- Высокая точность моделирования достигается для боковых радикалов внутри глобулы.

- Причина: в экспериментах остатки на поверхности более подвижны.

- Вычислительное проще упаковать гидрофобные остатки, чем учесть полярные контакты и водородные связи с водой или с участием воды.

```

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

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

# Улучшение модели

```

#(my %$coor,my $schnum)=read_pdb($ARGV[0]);
my %$coor=read_pdb($ARGV[0]);
my $sdir=$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 $sqnum=keys %$qwa;
```

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

```

#system("if [ -e $sdir/$sfilename ]; then mv $sfilename $sdir/$sfilename; else mv $sfilename $sdir/$sfilename; fi");

```

```
$sfilename=$sdir/$sfilename;
```

```
#$sfilename=$sdir/$sfilename;
```

```
$sfilename=$sdir/$sfilename;
```

```
print "$sfilename\n";
```

```
open OUT ">$sfilename";
```

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

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

```

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

- Методы минимизации энергии.
- Моделирование молекулярной динамики (оптимизация гидрофобики)
- Моделирование Монте-Карло.
- Любой известный подход для оптимизации структуры.

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

# Ошибки

```

#(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;
```

• Обычно ошибки не исправляются на последующих этапах моделирования.

• Хорошее выравнивание не исправит плохой выбор белка заготовки.

• Хорошее моделирование петель не исправит плохое выравнивание.

• При обнаружении ошибки необходимо повторять некоторые этапы.

```

foreach my $m (sort keys %qwa){
  my %qartets= %qwa{$m};
  my %q= find_q( %coor{$m} );
  #
  foreach my $s ( @{$qartets{$q}}){
    print "$q $coor{$m}{$s} {"$R"}->x,\"n";
    $nx=$nx+ $coor{$m}{$s}{"N9"}->x;
    $ny=$ny+ $coor{$m}{$s}{"N9"}->y;
    $nz=$nz+ $coor{$m}{$s}{"N9"}->z;

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

```



# Проверка

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

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

- Большинство программ для моделирования по гомологии выдают правильные значения для связей и валентных углов.

- Карта Рамачандрана в большинстве случаев для модели выглядит также, как для белка заготовки

- Проверка на ориентацию или положение заряженных остатков может быть полезна.

- Использование любых экспериментальных данных:
  - Остатки активного центра.
  - Места модификаций.
  - Места контактов.

ProQ сервер оптимизирован на поиск правильной модели а не нативной структуры.

```
$nx=$nx+ $coor($m)($res){ "O6" }->x;
$ny=$ny+ $coor($m)($res){ "O6" }->y;
$nz=$nz+ $coor($m)($res){ "O6" }->z;
```

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

# Ресурсы для гомологического моделирования

```
#!/(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;
```

- Modeller

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

- SwissModel

```
foreach my $q (keys %qartets){
my %qartets=@qartets;
my %q=find_quart($coor{$m});
my %q=find_q($coor{$m});
```

- Eva-CM

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

- Nest И т.д.

```
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 );
```

# Предсказание структуры белка *Ab initio*

```
#!/(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];
```

- Теоретически можно использовать молекулярную динамику.

- Моделирование отжига, как в МД так и в Монте-Карло.

- На основе фрагментов, Rosseta

```
  $filename="$dir/$ch.$qnum";
  $filename="$ch.$qnum";
  $filename="$chnum.$qnum";
  $filename="$ch.$qnum.$chnum";
  print "file: $filename\n";
  open OUT, ">$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}{\"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;
```



# Ab initio, Rosseta

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

my ($R,$N1,$N2,$N7,$O6);

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

- Для определения хорошей конформации использую специальные потенциалы, которые делают модель похожей на нативную

- Что можно использовать:

- Потенциалы для третичных контактов

- Гидрофобные потенциалы

- Потенциал для уменьшения радиуса вращения молекулы

- Водородные связи и т.д.

Можно добавить знание об дисульфидных мостиках, местах связывания катионов металлов и т.д.

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

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

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

```

# Threading — протягивание нити

```

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

```

```
my %$o;
• Сравниваем последовательность со всеми известными способами укладки.
```

```

if ($schnum > 0) {
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=$ARGV[1];
# $filename=$ARGV[2];
# $filename=$ARGV[3];
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chislo schinnum=$schnum\n";

```

```

foreach my $m (sort keys %{$coor{"O"}}){
my %$qartets=();
my %$q= find_q($coor{"O"}{$m});
• Каждую аминокислоту из модели помещаем в позиции белков разных укладок

```

```

# foreach my $q (keys %$qartets){
# foreach my $s (keys %$qartets{"$q"}){
• Определяем как хорошо эта аминокислота подходит белку заготовке на основе парных взаимодействий

```

```

my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
• Но основе суммарного результата определяем белок заготовку.

```

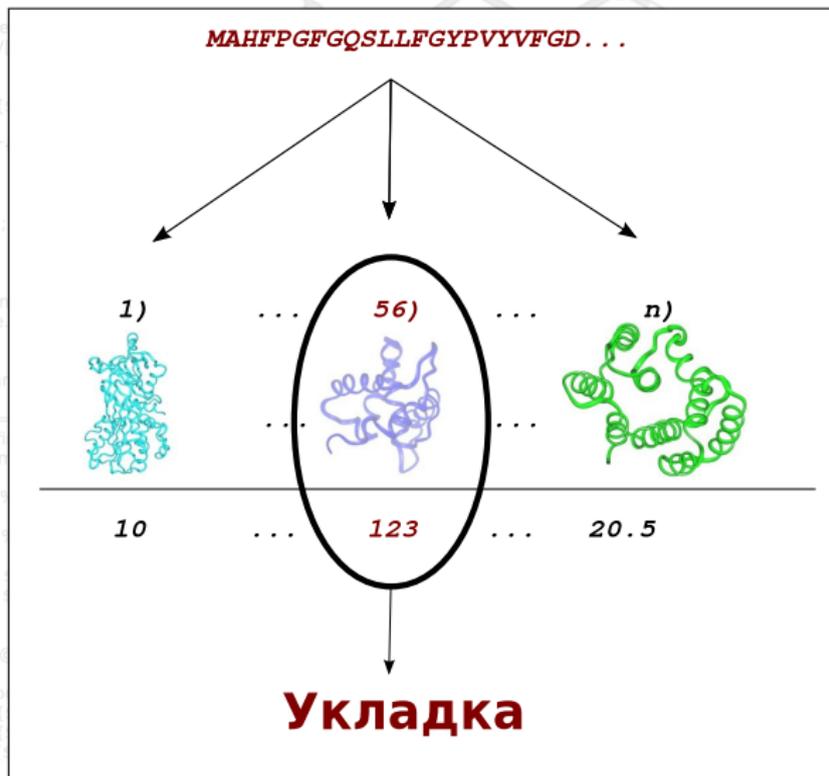
```

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;

```

## Threading — протягивание нити



```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
use List::Util qw( min );
use List::MoreUtils qw( uniq );
```

# Threading — недостатки

```

#(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;
```

- Взаимодействия в белке не всегда описываются парными контактами.
- Потенциалы часто основываются на профилях последовательностей.

```

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";
```

## Есть гибридные методы Rosseta/Threading: I-Tasser

```

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

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

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

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

# Распознавание укладки, Phyge2

```
#!/(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//";
ARLDLVIPMIYCGHGY
```

**PSI-Blast**

**HMM**

```
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";
```

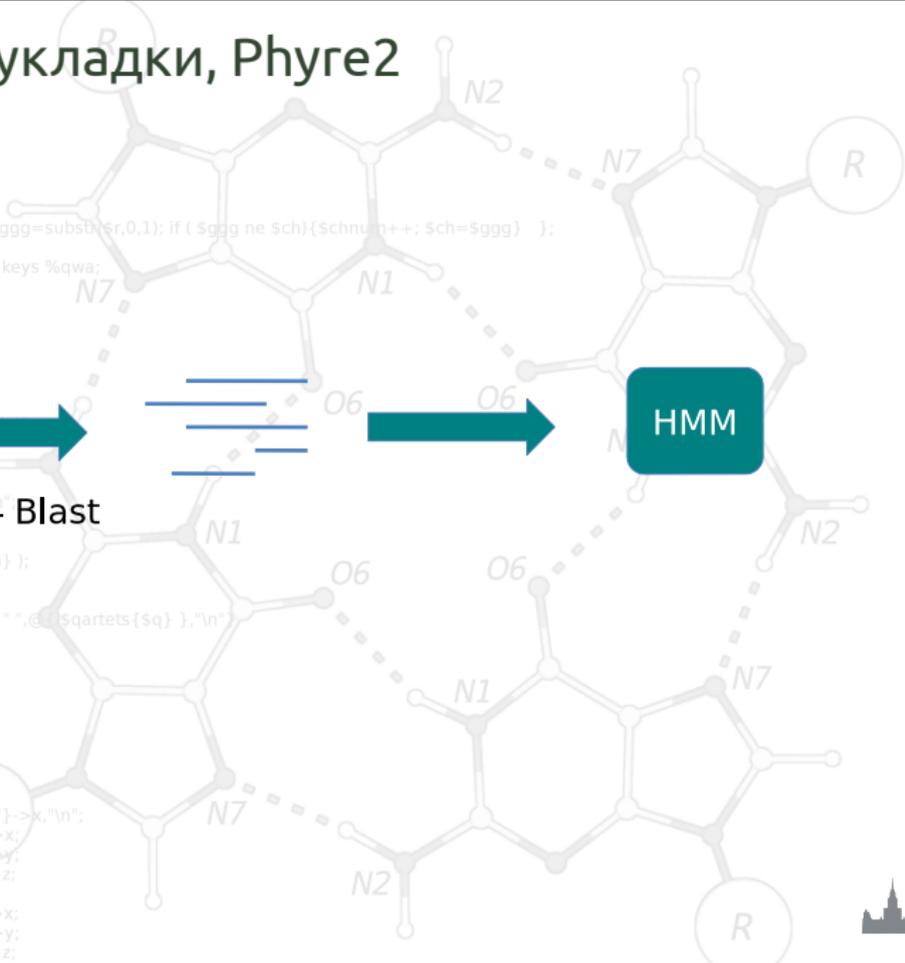
```
foreach my $q ( keys %qartets){
```

```
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;
```



# Phyre2

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

my $seq = "ARDLVIPMIYCGHGY";
my $target = "AFDLCDLIPV";

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

substr($seq, 0, 1), if ( $seq !~ /C/ ) {
    $seq = "R";
}
```

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

**PSI- Blast**

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

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

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

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

            foreach my $res ( @{$qartets{$q}} ){
                print "$q $coor{$m} {$res} {"$R"}->x,\n";
                $nx = $nx + $coor{$m} {$res} {"$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;
            }
        }
    }
}
```

**HMM**  
профили  
известных  
структур

**HMM**

**HMM-HMM**  
сравнение

**ARDL - -VIPMIYCGHGY**  
**AFDLCDLIPV - -CGMAY**



# Phyre2

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

ARDLVIPMIYCGHGY → PSI- Blast → HMM
```

**HMM - Позволяет найти далёкие гомологи**  
**Часто предсказываются хорошие модели при идентичности менее 15%**

```
3D-Model es
```

```
# print "$q :
  $nx=$nx+ $co
  $ny=$ny+ $co
  $nz=$nz+ $co

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



**HMM профили известных структур**

**HMM-HMM сравнение**

**ARLD - -VIPMIYCGHGY**  
**AFDLCDLIPV - -CGMAY**



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

# Мета серверы

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

- Сравнение разных методов.
- Большинство методов предсказывают правильную укладку в первых 10-20 результатах.
- Удаление структур с высоким значением параметров модели, но с единственной укладкой.
- Суперпозиция результатов, взвешивание.
- Часто выдают только позиции атомов остова.

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

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

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

```
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;
```

# Заключение

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

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

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

- Суть современного моделирования белков - эмпирическая
- Чем больше известной информации используется при моделировании тем точнее модель.
- Каждый метод имеет недостатки.
- Критический анализ модели позволяет выявить ошибки и улучшить модель.

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

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