

```
#!/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 $q=0;
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=0;
```

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

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

```
# foreach my $m (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 }{"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;
}
```

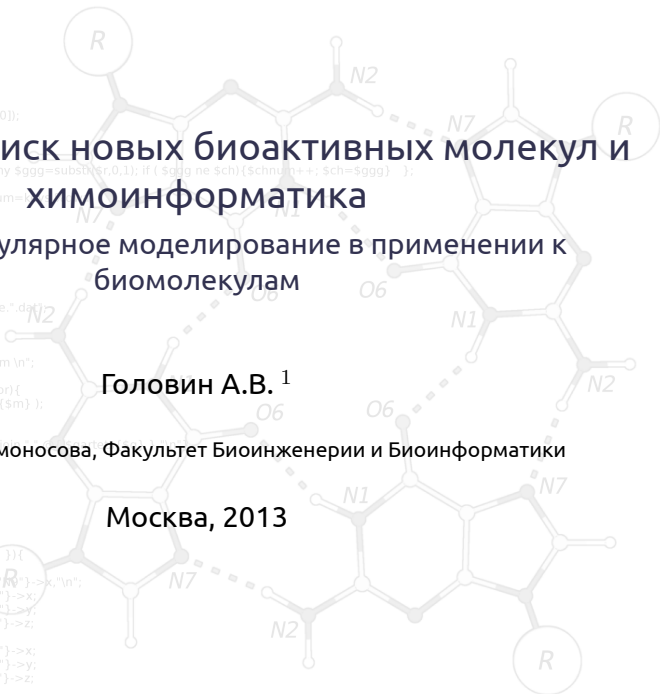
Лекция 2. Поиск новых биоактивных молекул и химинформатика

Курс: Молекулярное моделирование в применении к биомолекулам

Головин А.В. ¹

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

Москва, 2013



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

Содержание

```

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

```

Активные молекулы

Фарминдустрия

```

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

```

HTS

```

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

```

Химоинформатика

```

foreach my $q ( keys %qartets){

```

```

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

```

QSAR

```

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

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

```



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

Особенности деятельности фарм-производителей

```
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;
```

Дженерик - лекарство без патентной защиты (срок вышел)

```
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
```

- Рынок высоко конкурентен.

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

- Разработка нового лекарства занимает от 10 до 20 лет.

```
#
foreach my $q ( keys %quartets) {
foreach my $q ( keys %quartets) {
my $sx; my $sy; my $sz;
my $r;
```

- Новые лекарства приносят основную прибыль

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

- 4 основные фазы: открытие, разработка, испытания, продажи

```
#
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;
```

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

R&D

```

#(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 { $a->$b } keys %coor ){
    my $sggg=substr($ch,$r,$chnum);
    my %qqa=find_qq($sggg,$dir);
    my %qqa=keys %qqa;
}

```

Болезнь



Белок



Ингибитор

Испытания на животных

Лекарство и масштабирование

Клинические испытания



Подтверждение регулятором

```

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 = %qqa; #find_qq($sggg,$dir);
    my %q = find_q($coor{$m});
}

```

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

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

```
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 );
```

Новые технологии

```

#(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 } ;

```

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

```

```

if ($qnum > 0){
#system("ls -l $dir/$q");
my $filename=$dir/$q;
$filename=~ s/^~/\//;
$filename=~ s/ /_/;
# $filename=$dir/$q;
$filename="$dir/$filename.dat";
print "file $filename\n";
open OUT, ">$filename";
print OUT "#INFO chain $schnum qnum $qnum\n";

```

```

foreach my $q ( keys %q ){
my %qartets = %q{$q}; #find quartet $coor{$m} );
my %q = find_q( $coor{$m} );

```

```

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

```

```

my $sx; my $sy; my $sz;
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;

```

```


```

```

my $sx; my $sy; my $sz;
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;

```

```


```

```


```

```


```

- Чипы: экспрессия генов.
- Структуры: роботизированный поиск кристаллов.
- Высоко-производительный поиск ингибиторов.
- Виртуальный поиск.
- Комбинаторная химия.

Все это в основном относится к стадии поиска ингибитора


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

Как химинформатика может помочь?

```

#(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;
```

- Разработка методов и управление информацией об лигандах.

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

- Оценка данных *in silico* для минимизации рисков.

```
#system("cd $dir && cp $filename $schnum");
```

- Разработка библиотеки.

```
$filename=$schnum."/".$schnum."/".$filename;
my $filename="$dir/$schnum/$schnum/$filename";
```

- Виртуальный поиск.

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

- Оценка стоимости и выгоды.

```
print OUT "#INFORMATION";
```

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

- Организация доступа к информации.

```
my %$qartets=%qartets{$m};
```

- Интеграция процессов.

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

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

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

```
my $r;
```

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

```
# print "$q $coor{$m} {$res} {"N7"}->x,"n";
```

```
$nx=$nx+ $coor{$m} {$res} {"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;
```

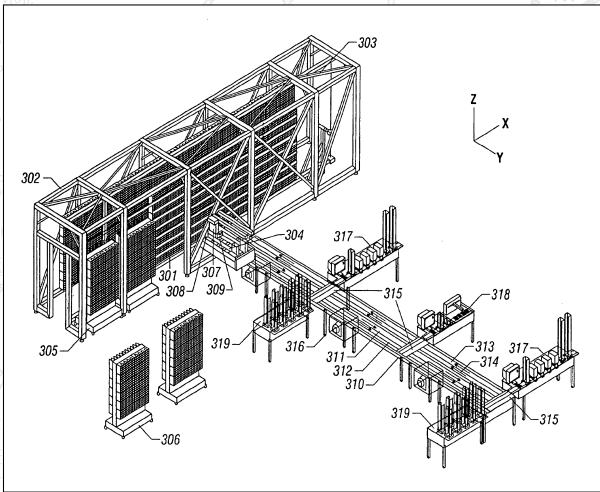
Пример: HTS, Высоко-производительный поиск ингибиторов

```
#!/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 $r ( sort keys %
my %qwa=find_quart($coor
```

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

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

# foreach my $q ( keys
foreach my $q ( keys
my $nx; my $ny; my
my $ox; my $oy; my
my $r;
foreach my $res (
# print "$q $c
$nx=$nx+ $coor
$ny=$ny+ $coor
$nz=$nz+ $coor($my ($res) {
my ($ox,$oy,$oz)=$coor($res){"O"
$ox=$ox+ $coor($m) ($res){"O"
$oy=$oy+ $coor($m) ($res){"O"
$oz=$oz+ $coor($m) ($res){"O"
}
```



до 100000 соединений в день


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

Комбинаторная химия

```
"Основа"
my $m=read_pdb($ARGV[0]);
```

```
my $s=$Sch($m,$ARGV[0]);
```

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

```
my $sch, my $schnum;
```

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

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

```
if ($sqnum){
```

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

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

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

```
  $filename =~ s/\./_/;
```

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

```
  $filename =~ s/\.num/"$filename";
```

```
  $filename =~ s/\.dat";
```

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

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

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

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

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

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

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

```
      $R1, $R2, $R3, $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 $scoor{$m} {$res} {"$R1"}->x,\n";
```

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

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

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

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

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

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

```
        }
```

```
      }
```

```
    }
```

```
  }
```

```
}
```

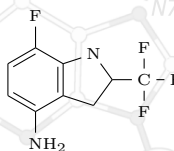
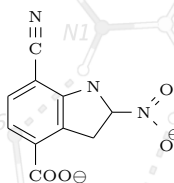
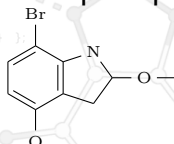
"Блоки"

$R_1 = \text{OH, OCH}_3, \text{NH}_2, \text{Cl, COOH}$

$R_2 = \text{Phe, OH, NH}_2, \text{Br, F, CN}$

$R_3 = \text{CF}_3, \text{NO}_2, \text{OCH}_3, \text{OH, PheO}$

Примеры



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

Химоинформатика и библиотеки

```

#(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;
```

- Какие блоки выбрать?

```
if ($qnum > 0){
#system("cp $dir/$ARGV[0].$ch.$qnum.pdb $dir/$ARGV[0].$ch.$qnum.pdb");
my $filename="-- $r -- $ch -- $qnum --";
$filename="-- $r -- $ch -- $qnum --";
# $filename=$ch.$qnum.$r;
$filename="$dir/$ch.$qnum.$r";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO: $ch $qnum $r\n";

```

- Какие библиотеки строить?

- Дополнение известных наборов

- Модификация под конкретный белок

- Полное "насыщение" библиотеки

```

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

```

- Компьютерное профилирование библиотеки

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

- Виртуальными библиотеками удобно манипулировать на компьютере

```

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

```

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

Компьютерное представление молекул

- Хранение в компьютере молекулы как изображения имеет малую ценность
- Большинство современных баз данных представляет молекулу как граф, с узлами и рёбрами
- Графы представляются как таблицы связей.

```
Marvin 04200617372D
```

```
4 3 0 0 0 0 0 999 V2000
0.0000 0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
0.7145 -0.4125 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0
-0.7145 -0.4125 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
0.0000 0.8250 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0
1 4 2 0 0 0 0
2 1 1 0 0 0 0
3 1 1 0 0 0 0
M END
$ox=$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);

Линейное представление молекул, SMILES

Молекула представляется в виде диаграммы и каждый атом проходится только один раз

<chem>CC</chem>	ethane	<chem>[OH3+]</chem>	hydronium ion
<chem>O=C=O</chem>	carbon dioxide	<chem>[[2H]]O[[2H]]</chem>	deuterium oxide
<chem>C#N</chem>	hydrogen cyanide	<chem>[[235U]]</chem>	uranium-235
<chem>CCN(CC)CC</chem>	triethylamine	<chem>F/C=C/F</chem>	E-difluoroethene
<chem>CC(=O)O</chem>	acetic acid	<chem>F/C=C/F</chem>	Z-difluoroethene
<chem>C1CCCCC1</chem>	cyclohexane	<chem>N[[C@H]](C)C(=O)O</chem>	L-alanine
<chem>c1ccccc1</chem>	benzene	<chem>N[[C@H]](C)C(=O)O</chem>	D-alanine

Реакции в виде SMILES

[I-].[Na+].C=CCBr >> [Na+].[Br-].C=CCI

реакция замещения

(C(=O)O).(OCC)>>(C(=O)OCC).(O)

образование сложного эфира


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

Стандартизация SMILES

- Очевидно, что одну молекулу можно описать разными способами.
- Морган в 1965 году предложил рассматривать каждый атом по свойству его окружения.
- Стандартные SMILES называют Unique.

Input SMILES	Unique SMILES
OCC	CCO
[CH3][CH2][OH]	CCO
C-C-O	CCO
C(O)C	CCO
OC(=O)C(Br)(Cl)N	NC(Cl)(Br)C(=O)O
ClC(Br)(N)C(=O)O	NC(Cl)(Br)C(=O)O
O=C(O)C(N)(Br)Cl	NC(Cl)(Br)C(=O)O

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

Описание SMILES: атомы

```

#(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 } ;

```

- Однобуквенные атомы, а именно : B, C, N, O, P, S, F, Cl, Br, I записываются как есть, как один символ.

- Все остальные атомы записываются в квадратных скобках [Pt]

- Так как атомы водорода обычно не указываются, то “валентность” атомов определяется как наименьшая из ближайших T.e. B (3), C (4), N (3,5), O (2), P (3,5), S (2,4,6).

- “Валентности”, отличные от “нормальных”, указывают в скобках [S], [H+], [Fe+2], [OH-], [Fe++], [OH3+], [NH4+]

```

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 );
```

Описание SMILES: связи

```

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

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

```

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

```

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

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

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

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

Ветвление цепи отображается в скобках ()

Пример: CCC(CC)COO

```

# print "$q $coor{ $m } { $res } { "N7" }->x, "\n";
$nx=$nx+ $coor{ $m } { $res } { "N9" }->x;
$ny=$ny+ $coor{ $m } { $res } { "N9" }->y;
$nz=$nz+ $coor{ $m } { $res } { "N9" }->z;
```

```

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

CC	этан
C=C	этилен
O=C=O	CO2
C#N	HCN
CCO	этанол
[H][H]	водород


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

Структуры где есть нековалентные связи

```
#!/(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{"0"}} ){ my $ggg=sub{<math>cos(0.1)</math>}; if ( $ggg ne $sch ){ $chnum++; $sch=$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=$chnum." ".$qnum." ".$filename." .dat";
$filename="$dir/".$filename." .dat";
print "$filename\n";
open OUT,">$filename";
print OUT "#INFO chain $chnum qnum $qnum \n";
```

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

```
# foreach my $q ( keys %qartets ) { print join " ", $q, $m, "\n"; }
[Na+].[O-]c1ccccc1
```

```
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;
```

В SMILES нотации это:

[Na+].[O-]c1ccccc1

или

c1cc([O-].[Na+])ccc1

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

Изомеры

```
#!/(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;
```

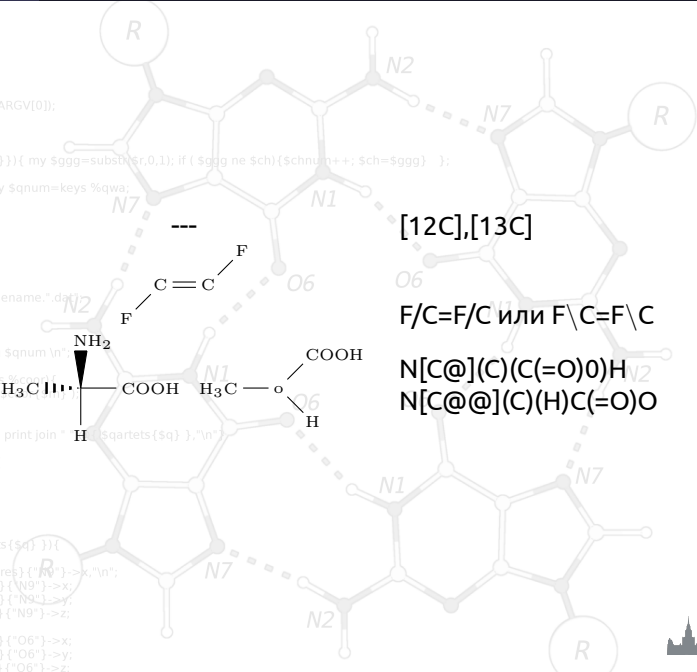
Изотопы

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

Цис-Транс

Хиральность

```
for my $m ( keys %$coor{"0"} ){
my %q = find_q( %coor{$m} );
# foreach my $q ( keys %qartets){ print join " ", $qartets{$q} ,"\n";
foreach my $q ( keys %qartets){
my $nx; my $ny; my $nz;
my $ox; my $oy; my $oz;
my $r;
foreach my $res ( @{$ $qartets{$q} }){
print "$q $coor{$m}{$res} {"$R"}->x,\n";
$nx=$nx+ $coor{$m}{$res} {"$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 );
```

SMARTS: паттерны для SMILES

```
#!/my $coor, my $chnum)=read_pdb($ARGV[0]);
```

В принципе, SMARTS это SMILES + операторы логики и варианты в позициях.

Пример для атомов:

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

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

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

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

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

```
  # $filename="$chnum"."$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->{sb} keys %coor } {
```

```
    my %qartets = %qartets; #find quart ($coor {$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 $sr;
```

```
      [C] алифатический углерод
```

```
      [C] ароматический углерод
```

```
      [a] любой ароматический атом
```

```
      [#6] любой атом углерода
```

```
      [++] атом с зарядом +2
```

```
      [R] атом в кольце
```

```
      [D3] атом с тремя связями ( не с водородами)
```

```
      [X3] атом с тремя связями, включая водороды
```

```
      [v3] атом с валентностью 3.
```

```
      # print join " ", ($m) {$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;
```



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

SMARTS: логические операторы и примеры

```
#!/(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"."$filename.".dat";
print "$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 ){
foreach my $q ( keys %qartets ){
my $nx;
my $ox; my $oy; my $oz;
my $r;
foreach my $res ( @($coor{$q}) ){
print "$q $coor{"$res"} {"$res"} {"O6"}->x;
$nx=$nx+ $coor{$m} {"$res"} {"O6"}->x;
$ny=$ny+ $coor{$m} {"$res"} {"O6"}->y;
$nz=$nz+ $coor{$m} {"$res"} {"O6"}->z;
}
my $qartets = %qwa; #find_quart( %coor{$m} );
my %q= find_q( %coor{$m} );
```

[!C;R]

[n;H1], [n&H1], [nH1]

[c,n&H1]

[X3&H0]

[c,n;H1]

Логика:

!e1	not e1
e1& e2	a1 and e2
e1,e2	e1 or e2
e1;e2	a1 and e2

Пример:

не алифатический C в кольце

H в пирроле

C или H в пирроле

Атом с тремя связями не с N

N или C в связи с одним H1

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

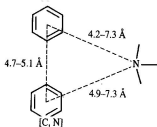
Поиск по 3D-базам данным

- Поиск в 2D-пространстве хорош для поиска подобных молекул, но биологически активные молекулы действуют благодаря специфической 3D-структуре.
- Взаимодействие с биополимером может происходить благодаря нужному расположению в пространстве некоторых групп. При этом различие в 2D-структуре может быть весьма существенным.
- Фармакофор — это набор свойств, которые являются общими для некоторой группы активных молекул.
- Пример: Антигистаминный 3D-фармакофор

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

foreach my $res ( @{$ $qartets{$Sq} } ){
#
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 );
```

Проблемы с фармакофорами

```

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

```

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

- Если молекулы более или менее подвижны, то это накладывает дополнительные требования на учёт конформационных превращений.

- Для определения фармакофора надо определить, какой набор групп располагается в биополимере идентично.

- Надо быть уверенным, что выбранный набор молекул связывается с белком в одном и том же месте. Однозначное указание на это можно получить только экспериментально.

```

# foreach my $m (sort { $a<=>$b } keys %coor){
  my $qnum=substr($m,0,1);
  my $q;
  # 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}{\"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;

```