

```
#!/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 $sqwa=
```

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

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

Лекция 2. Поиск новых биоактивных молекул и

ХИМОИНФОРМАТИКА

```
if ($qnum >0){  
#system("mkdir $dir");  
my $filename=$ARGV[0];  
$filename=~ s/^.*V//;  
$filename=~ s/\..pbdb//;  
# $filename=$chnum.".Q$num.".$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 m 1 МГУ им М.В. Ломоносова, Факультет Биоинженерии и Биоинформатики
```

```
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;
```

Головин А.В.¹

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 $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 $chnum=keys %qwa;
```

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

Фарминдустрия

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

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

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

```
#$filename=$chnum.". ".$chnum."/". $filename.".dat";
```

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

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

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

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

```
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}{'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;
```

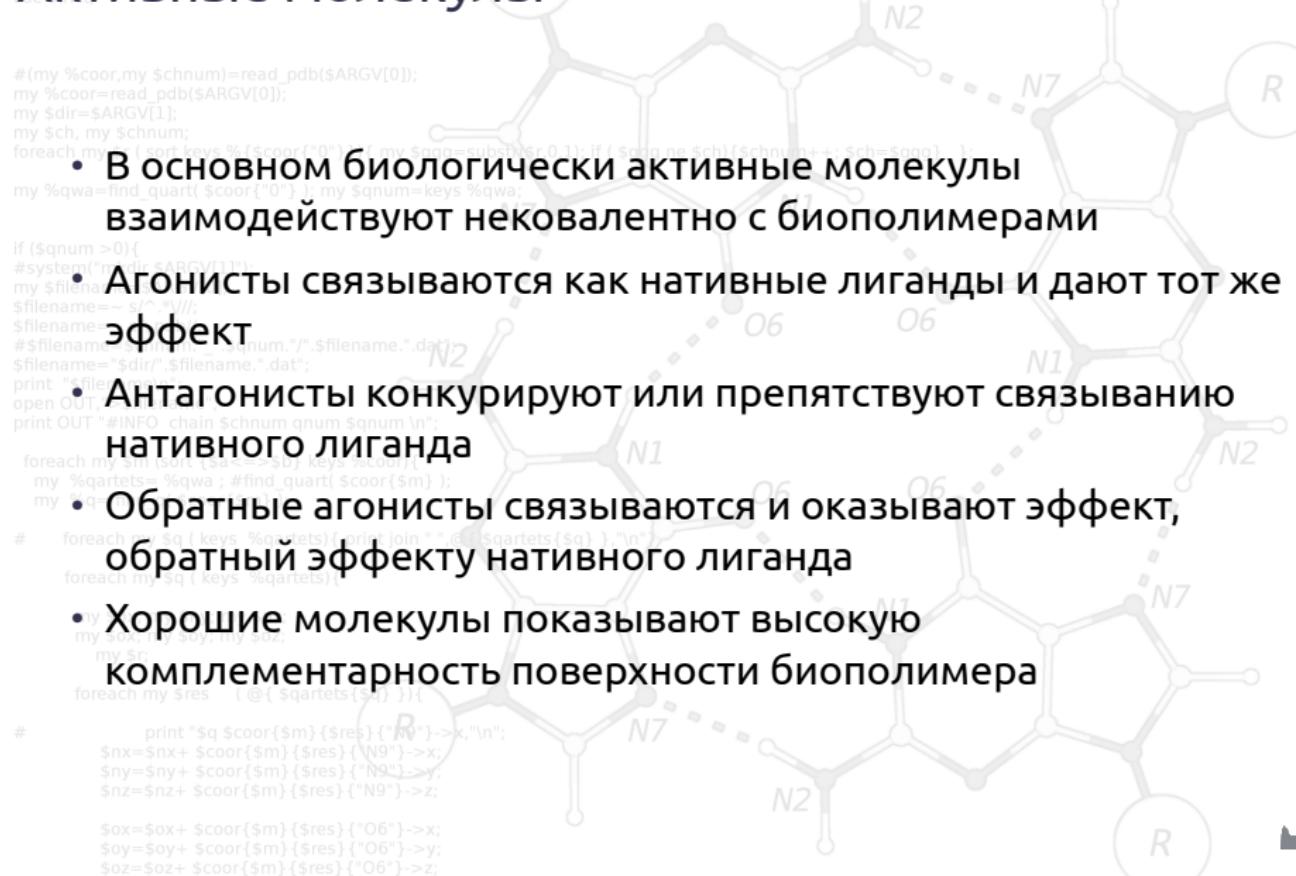


```
#!/usr/bin/perl
```

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

```
use strict;
```

```
use warnings;
```



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

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

Свойства лекарства

```
my %coor,my $chnum;
my %coor=read_pdb($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 ( $ggg ne $ch){$chnum++; $ch=$ggg} }
```

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

- Лекарством обычно являются не только те молекулы, которые хорошо связываются с биополимером.

- Лекарство должно иметь приемлемую растворимость

- Часто бывает, что лекарству надо проникнуть сквозь мембранны.

- Хорошо когда лекарство в итоге метаболизируется, а не накапливается в тканях.

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

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

```
# 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);
```

Как искать активные молекулы?

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
my %coor;
my $chnum;
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 } ;}
```

- Можно пытаться искать вещества в биоматериалах.
- Можно проводить роботизированное сканирование библиотеки соединений на активность в разных тестах.
- Недостаток сканирования: не все тесты можно адаптировать под робота.
- Возможен высокий уровень шума из-за неспецифических взаимодействий
- Можно применить фильтрацию по подобию соединений, для этого нужны ИТ.

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

Особенности деятельности фарм-производителей

```
#!/usr/bin/perl
use Math::Vec;
use Real qw( :all );
my $dir=$ARGV[0];
my $chnum=$ARGV[1];
my %coor=read_pdb($ARGV[0]);
my $sch=$ARGV[1];
my $ch, my $schnum;
foreach my $r ( sort keys %$coor{"0"} ){ my $ggg=substr($r,0,1); if ( $ggg ne $ch){$schnum++; $ch=$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/\..*//;
$filename=$chnum." ".$qnum."/". $filename.".dat";
$filename="$dir/".$filename.".dat";
print "$filename\n";
open OUT ">$filename";
print OUT
```

- Рынок высоко конкурентен.

• Разработка нового лекарства занимает от 10 до 20 лет.

• Новые лекарства приносят основную прибыль

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

```
foreach my $sq ( keys %qartets ){
    foreach my $sox ( keys %{$sq} ){
        my $soy; my $soz;
        my $sr;
        foreach my $res ( @{$qartets{$sq}} ){
            print "$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;
            $sox=$sox+ $coor{$m}{$res}{\"O6\"}->x;
            $soy=$soy+ $coor{$m}{$res}{\"O6\"}->y;
            $soz=$soz+ $coor{$m}{$res}{\"O6\"}->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 keys %coor) { $coor{$r} = $coor{$r}; } if ($chneq=$ch){$chnum+=$chneq}
my %qwa=find_qwa($ch,$chnum);
```

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/^.*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<=>$b} keys %coor){
    my %qartets=%qwa ; find_qartet($coor{$m});
    my %q= find_q( $coor{$m} );
    #foreach my $q ( keys %qartets){ print join(" ",@{ $qartets{$q}}) . "\n";
    foreach my $q ( keys %qartets){
```

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);
```

Новые технологии

```
my %coor,my $chnum;
my %coor=read_pdb($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;
```

- Чипы: экспрессия генов.

Структуры: роботизированный поиск кристаллов.

Высоко-производительный поиск ингибиторов.

- Виртуальный поиск.

Комбинаторная химия.

- Комбинаторная химия.

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

Все это в основном относится к стадии поиска ингибитора

```
foreach my $res ( @{$qwa{$q}} ){\n
    print "$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;\n
    $ox=$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);

Как химоинформатика может помочь?

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my %coor,my $chnum;
my %coor=read_pdb($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;
```

- Разработка методов и управление информацией об лигандах.

Оценка данных *in silico* для минимизации рисков.

- Разработка библиотеки.
- Виртуальный поиск.

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

- Оценка стоимости и выгоды.

• Организация доступа к информации.

- Интеграция процессов.

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

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

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

```
    $nx=$nx+ $coor{$m}{$res}{'\\"9"}->x;
```

```
    $ny=$ny+ $coor{$m}{$res}{'\\"9"}->y;
```

```
    $nz=$nz+ $coor{$m}{$res}{'\\"9"}->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);
```

Пример: HTS, Высоко-производительный поиск ингибиторов

```
my $coor, my $chnum=>read_pub($ARGV[0]);
my %coor=read pdb($ARGV[0]);
my $dir=$ARGV[1];
my $ch, my $chnum;
```

```
foreach my $r ( sort keys %
```

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

```
if ($chnum >0) {
#system("mkdir $ARGV[1]
my $filename=$ARGV[0];
$filename=~ s/^.*V//;
$filename=~ s/\..pb2//;
#$filename=$chnum.".".$
$filename="$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
```

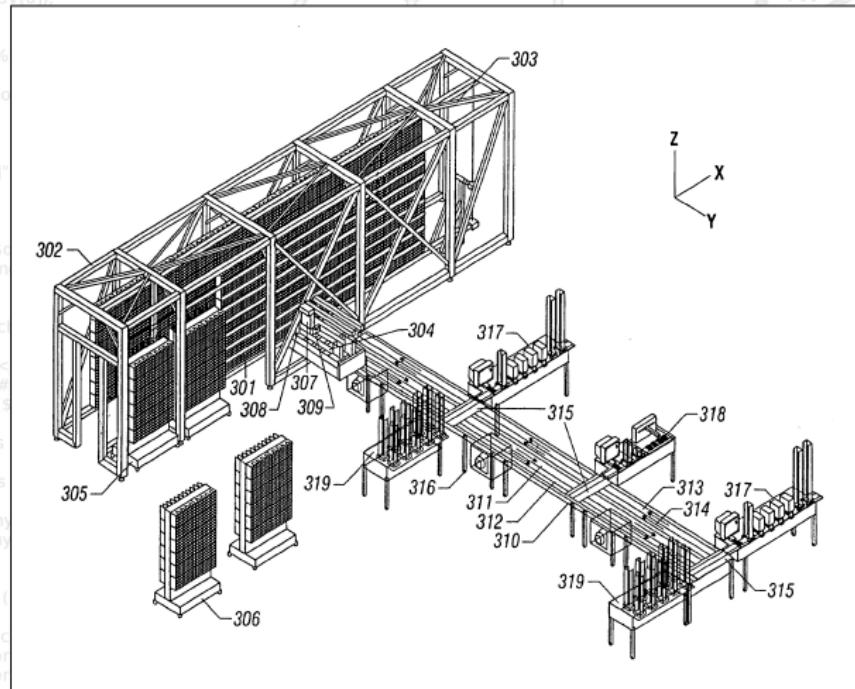
```
foreach my $q ( keys
```

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

```
foreach my $res
```

```
# print "$q $c
$nx=$nx+ $coo
$ny=$ny+ $coo
$nz=$nz+ $coo
```

до 100000 соединений в день



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

HTS и поток данных

```
#(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;
```

- Исполнить HTS.

- Решить какие соединения активны а какие нет.

Кластеризация активных соединений в классы.

Визуализация.

- Идентификация "основы" для каждого класса.

```
foreach my $m (sort {$a<>$b} keys %coor){
    my $sq=substr($m,0,1);
    my $sqz=substr($m,-1,1);
    foreach my $s (keys %qwa){
```

- Поиск причин, элементов структуры, которые приводят к "не активности".

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

- Использование структурной информации для объяснения активности.

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

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

```
die "Usage: $0 <PDB file> <output directory>"
```

Пример, комбинаторная химия

```
#(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;
```

- Исследователи используют "строительные блоки" для быстрого создания большого количества разных соединений.

```
if ($qnum>1){  
#system("mkdir $ARGV[1]");  
my $filename=$ARGV[1]."/";  
$filename.= "01.pdb";  
#$filename="name.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){
```

- Обычно используется некоторая "основа" и "строительные блоки" присоединяются к разным местам основы.

```
my %qarts=%qwa; #find quart $coor{$m};  
my %qarts;  
# foreach my $q (keys %qarts){ print join "\n", @{$qarts{$q}} ."\n"; }  
foreach my $sq (keys %qarts){  
my $nx; my $ny; my $nz;  
my $sx; my $sy; my $sz;  
my $r;  
  
foreach my $res ( @{$qarts{$sq}} ){  
#  
print "$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::Vec; Real qw(:all);

Комбинаторная химия

"Основа" (\$m)=read_pdb(\$ARGV[0]);

my \$dir=\$ARGV[0];

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){

#system "mkdir \$ARGV[1]";

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

\$filename =~ s/^.*\//

\$filename=sA.pdb//;

#\$filename=\$chnum.\$qnum."/". \$filename.

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

print "\$filename\n";

open OUT ">\$filename";

print OUT "#INP chain \$chnum qnum \$qnum\n";

foreach my \$m (sort {\$_<>-\$b} keys %coor){

my \$qartets=%qwa; #find quartet \$coor{\$m};

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

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

R1 ch 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;

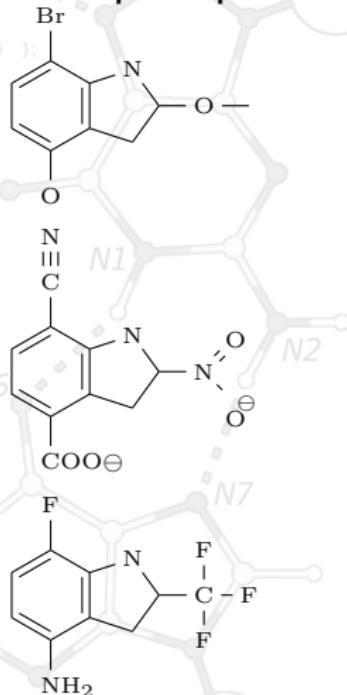
\$oy=\$oy+ \$coor{\$m}{\$res}{'O6'}->y;

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

"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("cd $dir $ARGV[1]");
```

```
my $file="";
```

```
$filename=~ s/\w*\//
```

```
$filename=~ s/\w*pdb/
```

```
#$filename=$chnum.$filename
```

```
$filename="$dir/$filename" date";
```

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

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

```
print OUT "#INFO c
```

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

```
my $q=qw($coor{$m});
```

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

```
foreach my $s (sort { $a <=> $b } keys %q{}
```

```
my $sq=qw($coor{$m}{$s});
```

```
my %sq=qw($coor{$m}{$s});
```

```
foreach my $t (@{ $qartets{$sq} }) {
```

```
my $x=$q{$t[0]}-$q{$t[1]}
```

```
my $y=$q{$t[1]}-$q{$t[2]}
```

```
my $z=$q{$t[2]}-$q{$t[0]}
```

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

```
my $nx=$q{$res[0]}-$q{$res[1]}
```

```
my $ny=$q{$res[1]}-$q{$res[2]}
```

```
my $nz=$q{$res[2]}-$q{$res[0]}
```

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

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

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

```
$ozo=$ozo+$coor{$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

```
print OUT "#INFO chain $qnum num $qnum\n";
foreach my $m (sort { $a<=>$b } keys %coor) {
    my %q = 0.0000 #find q of m();
    my %q = $coor{$m}();
```

```
    0.7145 -0.4125
#    foreach my $m (keys %coor) {
#        my %q = 0.0000 #find q of m();
#        my %q = $coor{$m}();
```

```
    -0.7145 -0.4125
#    foreach my $m (keys %coor) {
#        my %q = 0.0000 #find q of m();
#        my %q = $coor{$m}();
```

```
    my 0.0000 my $nz; my $oy; my $oz;
    my $ox; my $oy; my $oz;
```

```
    1 my $r;
    4 2 0 0 0 0
#    foreach my $es (sort { $a<=>$b } keys %coor) {
#        print "sq $coor{$m} {$res} (" . $v . ")->x,\n";
```

```
    2 1 1 0 0 0 0
#        $ny=$ny+$coor{$m}->{$res}("N9")->y;
#        $ny=$ny+$coor{$m}->{$res}("O9")->y;
```

```
    3 $n1-$nx $coor{$m}->{$res}("O6")->z;
$ny=$ny+$coor{$m}->{$res}("N9")->z;
```

```
M END $nz+$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};
use strict;
use warnings;
```

Линейное представление молекул, SMILES

```
#!my %coor; my $chnum=
```

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

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

```
my %
```

CC

ethane

[OH3+]

hydronium ion

O=C=O

carbon dioxide

[[2H]]O[[2H]]

deuterium oxide

C#N

hydrogen cyanide

[[235U]]

uranium-235

CCN(CC)CC

triethylamine

F/C=C/F

E-difluoroethene

CC(=O)O

acetic acid

F/C=C/F

Z-difluoroethene

C1CCCCC1

cyclohexane

N[[C@@H]](C)C(=O)O

L-alanine

c1ccccc1

benzene

N[[C@H]](C)C(=O)O

D-alanine

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

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

```
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;
```

Реакции в виде SMILES

[I-].[Na+].C=CCBr >> [Na+].[Br-].C=CCI
 $(C(=O)O).(OCC)>>(C(=O)OCC).(O)$

реакция замещения
 образование сложного эфира

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

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

```
$ox=$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: атомы

```
#!/usr/bin/perl
use Math::VectorReal qw(:all);
my %coor;
my $chnum;
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 };
```

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

```
if ($chnum > 1) {
    #system("mkdir $ARGV[1]");
    my $filename = $ARGV[0];
    $filename =~ s/\w+/$chnum/g;
    $filename = "$filename.$chnum";
    $filename = "$filename.dat";
    print "$filename\n";
    open OUT, ">$filename";
    print OUT "
```

• Все остальные атомы записываются в квадратных скобках [Pt]

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

```
# foreach my $q (keys %qartets) {
    my %q=q{nd_q($coor{$q})};
    my %q=join "", @qartets{$q};
    foreach my $sq (keys %qartets) {
        print join "", @qartets{$sq};
```

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

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

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 $ggg=$coor{"0"}{$r} if ($ggg->get_name eq "C" & $ggg->get_ch=$ggg) };
my %qwa=find_quart( $coor{"0"} ); my $qnum=keys %qwa;
```

```
if ($qnum >0){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^.*\//;
$filename=~ s/\.\pdb//;
#filename=$chnum,".",$qnum,$filename.".dat";
$filename="$dir/$filename.dat";
print "$filename\n";
open OUT,>$filename;
print OUT "#INFO chain $chnum qnum $qnum \n";
foreach my $m (sort {$a<=>$b} keys %coor){
    my %qartets=%qwa; #find quart $coor{$m};
    my %q= find_q( $coor{$m} );
    #foreach my $q ( keys %qartets){ print join " ",@{$qartets{$q}} ,"\n" }
    foreach my $q ( keys %qartets ){
        my $nx; my $ny; my $nz;
        my $qz;
```

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

Ветвление цепи отображается в скобках ()

Пример: CCC(CC)COO

```
#      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);
```

Описание SMILES: циклы

```
#{my %coor,my $num}=read_pdb($ARGV[0]);
```

```
my $coor=$coor{$_}{'$num'}
```

```
my $di=$coor{$_}{'$num'}
```

```
my $ch, my $num;
```

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

C1CCCCC1 циклогексан

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

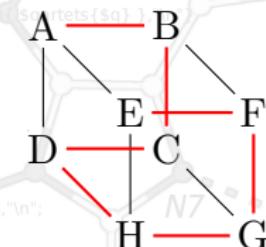
```
if ($qnum >1) {#system("mkdir $ARGV[0]"); my $filename=$ARGV[0]; $filename=~ s/\w*\$//; $filename=~ s/.pdb//; #filename=$num.". ".$qnum.".dat"; $filename="$dir/$filename.dat"; print "$filename\n"; open OUT,>">$filename"; print OUT "#INFO chain $chnum qnum $qnum \n";
```

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

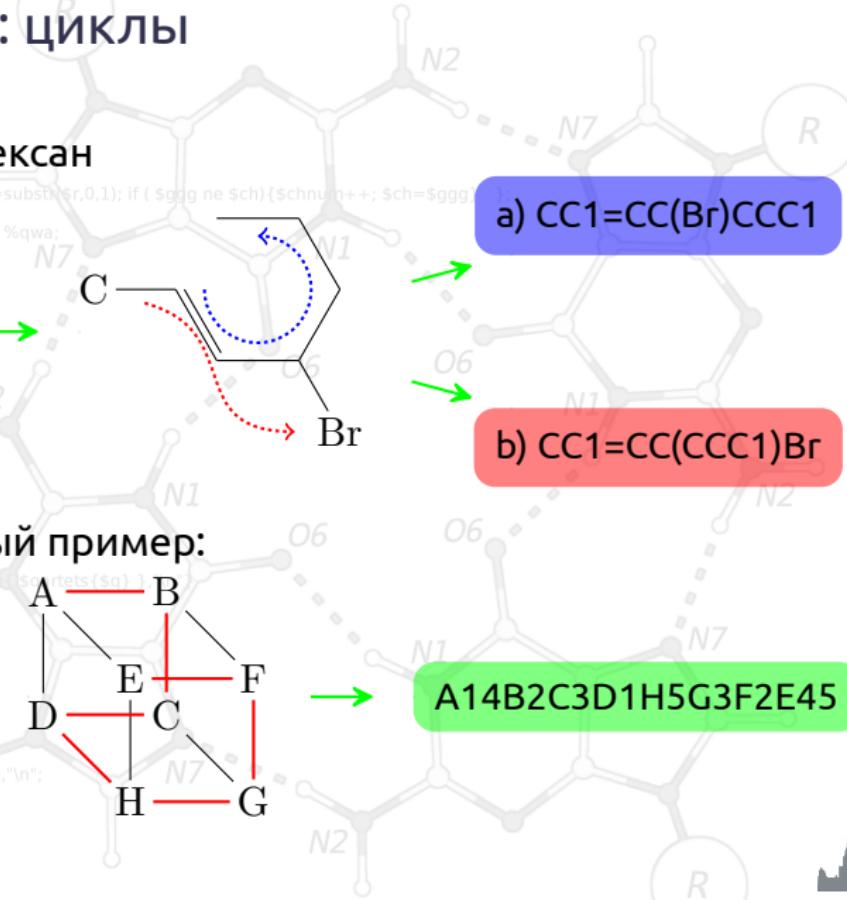
```
my %q1=$m;%q2=find_quart($coor{$m}); my %q1=qw
```

Или более сложный пример:

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



A14B2C3D1H5G3F2E45



a) CC1=CC(Br)CCC1

b) CC1=CC(CCC1)Br

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

Описание SMILES: ароматика

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

```
my %coor;my $chnum=0;
```

```
my $di=$chnum;
```

```
my $ch, my $chnum;
```

```
foreach my $line (@{$coor{$_}}){
```

```
my $ccc=$line->get_atom_name();my $ccc++;my $ccc=$chnum+; $ch=$ccc} );
```

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

- SMILES для определения ароматичности использует расширенный алгоритм Хюкеля.

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

- c1ccccc1 eq C1=CC=CC=C1 тут все атомы находятся в sp₂-гибридизации
- c1ccccc1 eq C1=CC=CC1 , последний атом в гибридизации sp₃.
- Ароматичными могут быть атомы: C, N, O, P, S, As, Se, и *.

• Пример: c1cnc[nH]c(=O)1

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

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

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

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

```
my $r;
```

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

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

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

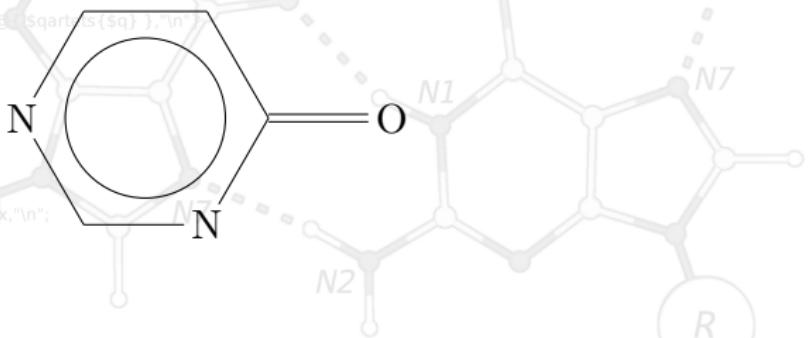
```
$ny=$ny+$coor{$m}{$res}{\"N9"}->y;
```

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

Структуры где есть нековалентные 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 $ggg=$coor{"0"}{$r}; if( $ggg ne $ch){$chnum++; $ch=$ggg} }
my %qwa=find_quart( $coor{"0"} ); my $qnum=keys %$coor{"0"};
```

```
if( $qnum > 0 ){
#system("mkdir $ARGV[1]");
my $filename=$ARGV[0];
$filename=~ s/^.*\//;
$filename=~ s/\.\pdb//;
#filename=$chnum."_".$qnum."/".$filename.".dat";
$filename="$dir/".$filename.".dat";
print "$filename\n";
open OUT ">$filename";
print OUT "#INFO chain $chnum qnum $qnum \n";
foreach my $m (sort { $a<=>$b } keys %$coor){
    my %qartets=%qwa; #find quart $coor{$m};
    my %q= find_q( $coor{$m} );
}
```

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

```
        my $nx; my $ny; my $nz;
        my $sx; my $sy; my $sz;
        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;
            $sx=$sx+ $coor{$m}{$res}{'O6'}->x;
            $sy=$sy+ $coor{$m}{$res}{'O6'}->y;
            $sz=$sz+ $coor{$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." ".$qnum."/"; $filename.=dat";  
$filename=~ s/\.\dat//;
```

```
print "Starting...\n";
```

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

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

```
for my $sq ( keys %qwa){
```

```
my $q= find_q( $coor{$sq} );
```

```
# foreach my $sq ( keys %qwa){ print join " ", $qwa{ $sq } , "\n" }
```

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

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

```
foreach my $res ( @{ $qwa{ $sq } } ){
```

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

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

#!/usr/bin/perl

use Math::VectorReal; use warnings;

SMARTS: паттерны для SMILES

#(my %coor,my \$chnum)=read_pdb(\$ARGV[0]);

В принципе, SMARTS это SMILES + операторы логики и варианты в my %ch, my \$chnum; позициях.

Пример для атомов:

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

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

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

```
[D3] [X3] [v3] foreach my $res (@{$qartets{$q}}) {
    print join " ",@{$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;
```

алифатический углерод
ароматический углерод
любой ароматический атом
любой атом углерода
атом с зарядом +2

атом в кольце
атом с тремя связями (не с водородами)

атом с тремя связями, включая водороды
атом с валентностью 3.

```
#!/usr/bin/perl
use Math::VectorReal;
use warnings;
```

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/\..*/.dat/;
    #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 $ny; my $nz;
        my $sx; my $sy; my $sz;
        my $r;
        foreach my $res (@{
            print "$q $coor{$m} $res\n";
            $nx = $nx + $coor{$m}{$res}{x};
            $ny = $ny + $coor{$m}{$res}{y};
            $nz = $nz + $coor{$m}{$res}{z};
        });
    }
}
```

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

Логика:

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

Пример:

не алифатический С в кольце
 Н в пирроле
 С или Н в пирроле
 Атом с тремя связями не с Н
 Н или С в связи с одним H1

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

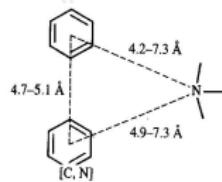
Поиск по 3D-базам данным

• 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



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

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

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

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

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

Проблемы с фармакофорами

```
#(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;
```

- Если молекулы более или менее подвижны, то это накладывает дополнительные требования на учёт конформационных превращений.⁶

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

```
#system('cp $filename $filename')
```

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

```
$filename=$ARGV[1];
```

```
#$filename=$chnum." ".$qnum.". $filename.".dat
```

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

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

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

```
print OUT "#NCS $chnum $qnum $chnum\n";
```

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

```
my $q = $qwa{$m};
```

```
my $q;
```

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

```
# for
```

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

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

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

```
my $r;
```

- 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 $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;
```

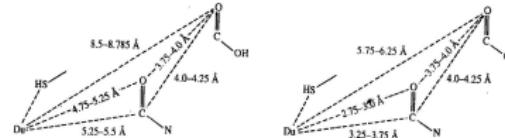
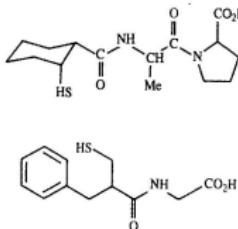
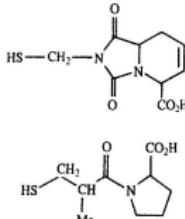
#!/usr/bin/perl

use Math::VectorReal qw(:all);

Систематический поиск

```
#!/usr/bin/perl
use Math::VectorReal qw( :all );
my %coor=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
my $cl=0;
foreach my $key (keys(%coor)) {
    my $sqnum=substr($key,0,1);
    if ($sqnum ne $cl) { $clnum++; $cl=$sqnum }
    $coor{$key} = $coor{$key} - $coor{$key(0,0,0)};
```

my %qwa:



- Выбирают точки, которые по мнению исследователей определяют активность. Делают конформационный поиск для всех молекул. Если находят пересечения по геометрии, то на основе этих точек и геометрии пересечения формулируют фармакофор.

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

Базы данных:

```
#{my %coor,my $chnum}=read_pdb($ARGV[0]);
```

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

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

```
my $ch,my $id;
```

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

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

• Cambridge database

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

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

```
my $filename;
```

```
$filename=~ s/\..*/$chnum."/
```

```
#$filename=$chnum." ".$qnum."/". $filename.".dat";
```

The screenshot shows the PubChem Compound search interface. At the top, there's a search bar with "PubChem Compound" and a dropdown menu. Below the search bar are tabs for "Search", "Help", and "PubChem Structure Search". The main search area has fields for "Search By:" (Name/Text, Identity/Similarity), "Substructure/Substructure", "Molecular Formula", "3D Conformer", and "Saved Search". There are also buttons for "Draw a Structure", "CID, SMILES, InChI", "Structure File", "Edit", "Search", and "Preview". On the right side, the results for "Ibuprofen - Compound Summary (CID 362)" are displayed. It includes sections for "Table of Contents", "Identification", and "2D Structures" (which shows the ibuprofen structure). The identification section lists various names and synonyms for ibuprofen.

```
$n2=$nz+ $coor{$m} {$res} {"N9"}->z;
```

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

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

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

#!/usr/bin/perl

use Math::VectorReal qw(:all);

SOAP доступ к PubChem

```

#(my %coor,my $chnum)=read_pdb($ARGV[0]);
my %coor=read_pdb($ARGV[0]);
my $dir=$ARGV[1];
use SOAP::Lite; # +trace => qw(debug);
import SOAP::Data qw(name);
my $ggg=substr($r,0,1); if ( $ggg ne $ch){$chnum++; $ch=$ggg}  );

my %awa=find_qnum('Squid["0"]'); my $qnum=qkeys %awa;
# Create PUG SOAP service object
my $pug_soap = new SOAP::Lite
If ($qnum){uri => "http://pubchem.ncbi.nlm.nih.gov/",
#system("rm -f $dir/$filename");
my $fil proxy =>"http://pubchem.ncbi.nlm.nih.gov/pug_soap/pug_soap.cgi";
$filename =~ s/\^V//;
$filename =~ s/\^A//;
#*****$qnum." ".$qnum."/". $filename.".dat";
$filename =~ s/\^I//;
$filename =~ s/\^P//;
$p $smile= $ARGV[0];
open OUT,>">$filename.dat";
print OUT "#INFO chain $chnum qnum $qnum\n";
my $StrKey = InputStructure($smile, "eFormat_SMILES");
foreach my $m (sort {$a<=>$b} keys %coor){
    my $sq = $awa{$m}{'find quart $coor($m)'};
    # IdentitySearch($m);
    my $ListKey = StructureSearch($StrKey, 250000);
    #print GetListItemsCount($ListKey) . "structures found\n";
    foreach my $q ( keys %qartets){
        my $url = Download($ListKey, "eFormat_SMILES", "eCompress_GZip");
        my $six, my $sny, my $syz;
        my $sox, my $syy, my $syz;
        print "Downloading... ";
        system ("wget -O $name\_smiles.gz $url");
        print "Done!\n";
    }
}
print "$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;

```