calc sans fort variable D2O fraction (not %)

dc 11/19, 12/22, 7/23 8/23 modify for existing out's

```
datetime
```

```
ans = datetime
   06-Sep-2023 11:15:15
```

```
clearvars
close all
cd ('/gs/gsfs0/shared-lab/cowburnlab'); % working areas
Lbase=pwd;
addpath(genpath('./lib1')) % DC library
addpath('./TraDES-2')      % local
cd ('ATSAS');
P=pwd;
addpath(P);

pwd
```

```
ans =
'/gs/gsfs0/shared-lab/cowburnlab/ATSAS'
```

```
delete(gcp('nocreate'));
```

```
Parallel pool using the 'Threads' profile is shutting down.
```

```
 parpool('processes');
```

```
Starting parallel pool (parpool) using the 'processes' profile ...
Connected to parallel pool with 32 workers.
```

```


datetime
```

```
ans = datetime
   06-Sep-2023 11:17:27
```

```
narginX=0;
switch narginX   % Use defaults, all changeable
    case 0      % just for test cases,   Relic of function call
       N0=1; % case A , 0 was 9000001
       N=2e6;
        fD2O=0.43;
        fDeut=1;
        ROOT='FSFG12';
        new=0;
    case 1      % make a lot more with default values
        fD2O=0.42; fDeut=1;    ROOT='FSFG12';  new=0;
    case 2
```

```
        fDeut=1;    ROOT='FSFG12';     new=0;
    case 3
        ROOT='FSFG12'; new=0;
    case 4
        new=0;
    otherwise   % exit
end
% make strings of values
fD2O=iif(ischar(fD2O),fD2O,strip(num2str(fD2O)));
fDeut=iif(ischar(fDeut),fDeut,strip(num2str(fDeut)));
```

```
% set up TRADES scripts
BIN='/public/apps/xplor/xplor-nih-3.5/bin/calcSAXS';

BIN_T=([Lbase filesep 'TraDES-2']);
TRADES=[BIN_T filesep 'trades -p T -a F ']; % executable command
TEMP='./tmp12';                             % working tmp dir, must survice
to analysis
TAG='_0000001';                             % relic of trades
P=pwd;      %push current
cd (BIN_T);
xStatus=0;
if new || ~exist ([ROOT '.trj'],'file')     % do the default setup for Trades
    % adaped from SaSp DaCo 11/19, uses default values
    system ([BIN_T filesep 'benchtraj']);
    [xStatus, cmdout]=system([BIN_T filesep 'seq2trj -f ' ROOT '.faa -o '
ROOT ' -cT -v ' ]);
    if xStatus, error(cmdout); end
    cd (P);

end         % pop current
if xStatus , error(cmdout); end
checkPDB=1000;      % check files of reasonable size
checkPHE=100;
outM=struct([]);
```

from the past

```
try
    %load Aug2
    N0= 1;
catch
%    R=readmatrix('out1400000R.txt');
%Q=readmatrix('out1400000Q.txt');
%R=single(R);    Q=single(Q);
%N0=numel(R)+1;
%save Aug2
end
```

```
istep=10000;
N=2000006
```

```
N = 2000006
```

```
quick=false
```

```
quick = logical
   0
```

```
 for iout=N0:istep:N
    nz=min(iout+istep-1, N);
     parfor jj=iout:nz


        cmdout=''; Status=1;
        s_N=strip(num2str(jj));         % simple number  as char
        tmpS=[TEMP filesep ROOT s_N ]   % root for current file
if ~quick
        if ~exist ([tmpS TAG '.phe'],'file')

            if exist ([tmpS TAG '.pdb.gz'],'file') && ~exist([tmpS TAG
'.pdb'],'file')
                gunzip([tmpS TAG '.pdb.gz']);
            end         % probably may need to run in  for loop once
            if exist([tmpS '.pdb'],'file')
                movefile([tmpS '.pdb'],[tmpS TAG '.pdb'])
            end
            if ~exist([tmpS TAG '.pdb'],'file')     % execute Trades case
                cmd=[TRADES ' -f ' ROOT ' -o ' tmpS ' -b 1 '];
                [Status, cmdout]= system (cmd); %#ok<*ASGLU>
            end
            if Status, error(cmdout); end

            ff=fileread([tmpS TAG '.pdb']);

            if numel(ff) < checkPDB
                warning(' PDB short, deleted');
                delete ([tmpS TAG '.pdb']);
                continue;
            end

            infi=[tmpS TAG  '.phe'];
            cmd = ['grep ''^ATOM.*PHE''  ' tmpS TAG  '.pdb  > ' infi  ];
            [a,b]=system(cmd);
            if a, error(b); end
            ff=fileread([tmpS TAG '.phe']);
            if numel(ff) < checkPHE
                warning(' PHE short, deleted');
                delete ([tmpS TAG '.phe']);
```

```
                continue;
            end
        end
end % quick
        outf=[tmpS TAG s_N  '.out'];    % file for SANS profile
        tmpQ=[BIN '  -sans -fractionD2O ' fD2O ...
            ' -fractionDeuterated ' fDeut ' ' tmpS TAG  '.phe > ' outf ...
            ' ; sync ' ] ; % set up command

        if new || ~exist(outf,'file')       % look for file
            if new || ~exist([outf '.gz'],'file')

                [Status, cmdout]=system(tmpQ);
                if Status, error(cmdout);   end
            else
                gunzip([outf '.gz']);
            end
        end

        Status=-10;
        while Status
            try
                out(jj)=getoutf(outf);
                Status=0;
            catch
                disp([ 'error outf, retry  ' tmpS] );
                system('sync');
                [Status2, cmdout]=system(tmpQ);  % retry
                if Status2 , warning (cmdout); end
                Status=Status+1;    % seems not needed, but was at one time,
possibly because of file sync issues
                if ~Status, warning(['file not processed ' tmpS]); end

            end
        end

        if ~rem(jj,1000), disp(jj); system('sync'); end   % systen monitor


      %    end
    end
%save Aug2
end
```

```
        3000

        4000

        5000

        1000
```

```
     1989000

     1990000

     1995000

     1991000

     1992000

     1994000

     1993000

     1997000

     1996000

     1998000

     1999000

     2000000
```

```
close all;

save("out.mat","-v7.3");        % -v needed for size
```

do preliminary plots for survey

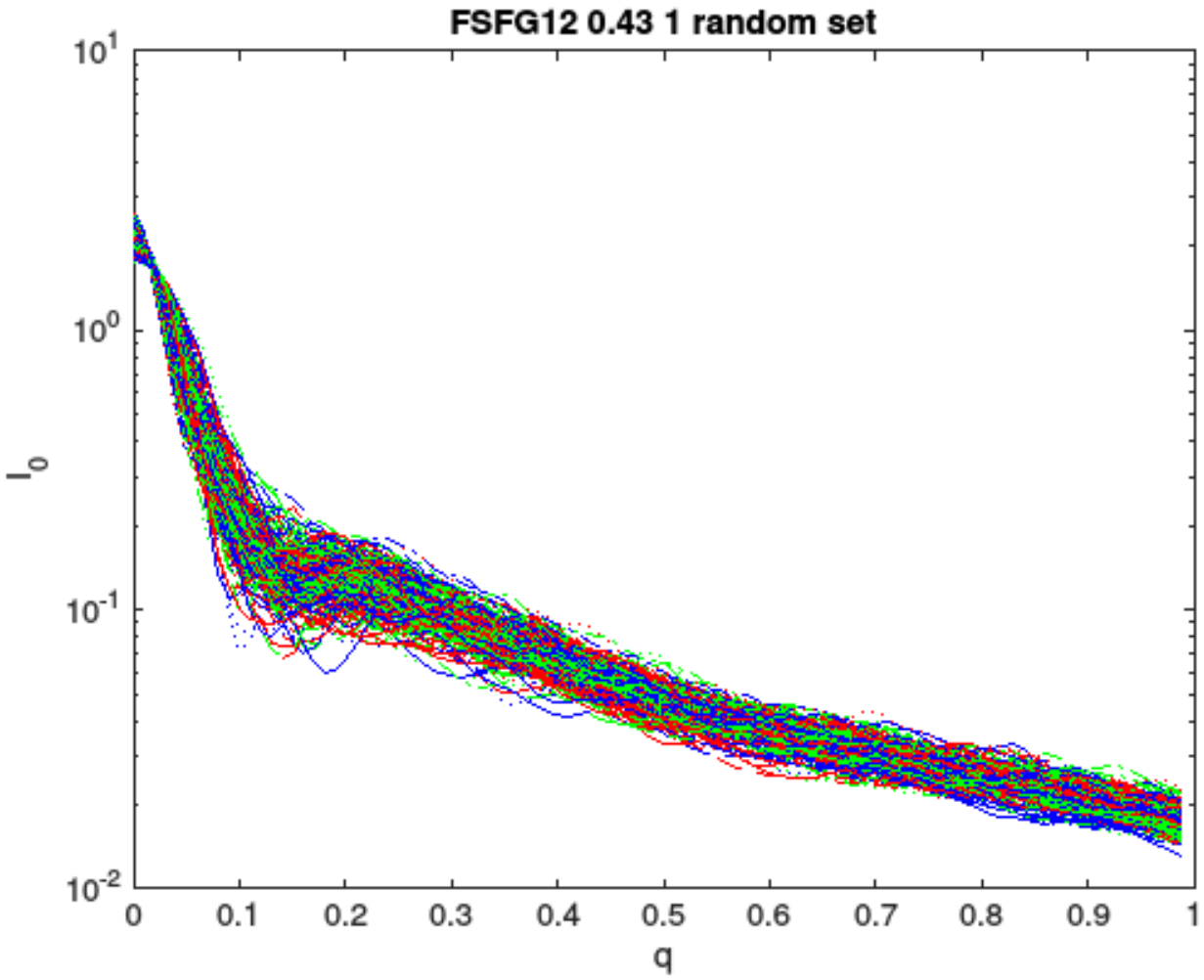
```
datetime
```

```
ans = datetime
   06-Sep-2023 14:04:41
```

```
% fix  R I2
%
%I2=Q(:,2:2:end);
%for ii=numel(R)+1:numel(out)
parfor ii=1:numel(out)
      R(ii)=out(ii).R;
    I2(:,ii)=out(ii).Q(:,2);
end


rq=out(end).Q(:,1);  % range of q
```

```
set(groot,'defaultAxesColorOrder',[1 0 0;0 1 0;0 0 1], ...
    'defaultAxesLineStyleOrder','-|--|:|-.');       % plot formatF
```

```
h0=figure;

iran=300;
dN=randperm(numel(out),iran);
dN=max(dN,1);

for jji=1:iran
    jj=dN(jji);
    semilogy(rq,I2(:,jj));
    if jji==1, hold on; end % slow, but not easly done otherwise
end
xlabel('q'); ylabel('I_0'); title([ROOT ' ' fD2O ' ' fDeut ' random set']);
axis([0 1 0.01 10]);
```




```
savefig(h0,'fig0', 'compact');
```

plot extreme cases

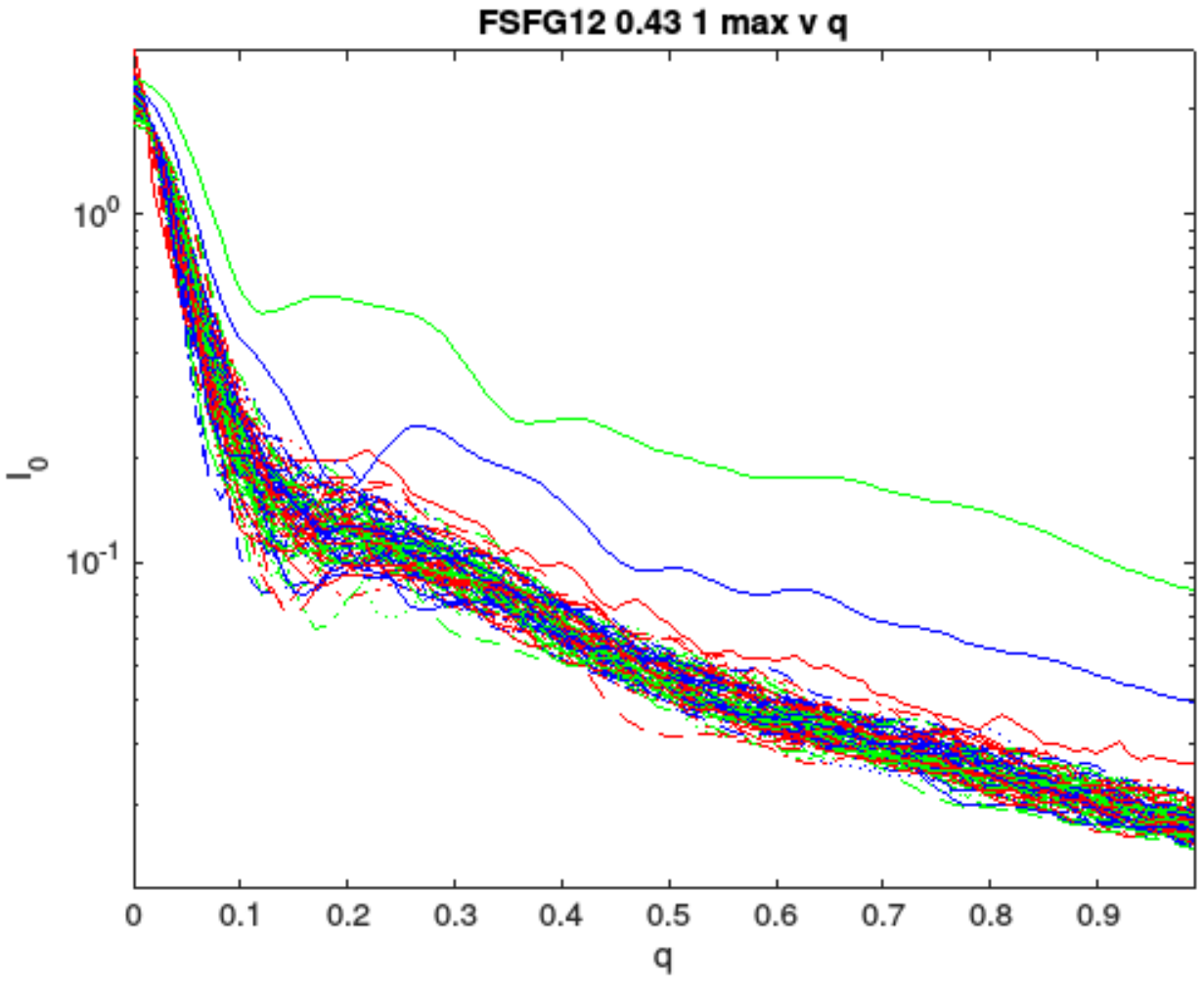
```
qset=rq;  % assumes q's same in all
iset=zeros(numel(qset),N,'single');
iset=I2;
```

```
%for ii=1:N
 %   iset(:,ii)=out(ii).Q(:,2);      %matrix of I values
%end
mxis=max(iset(:));
mnis=min(iset(:));
```

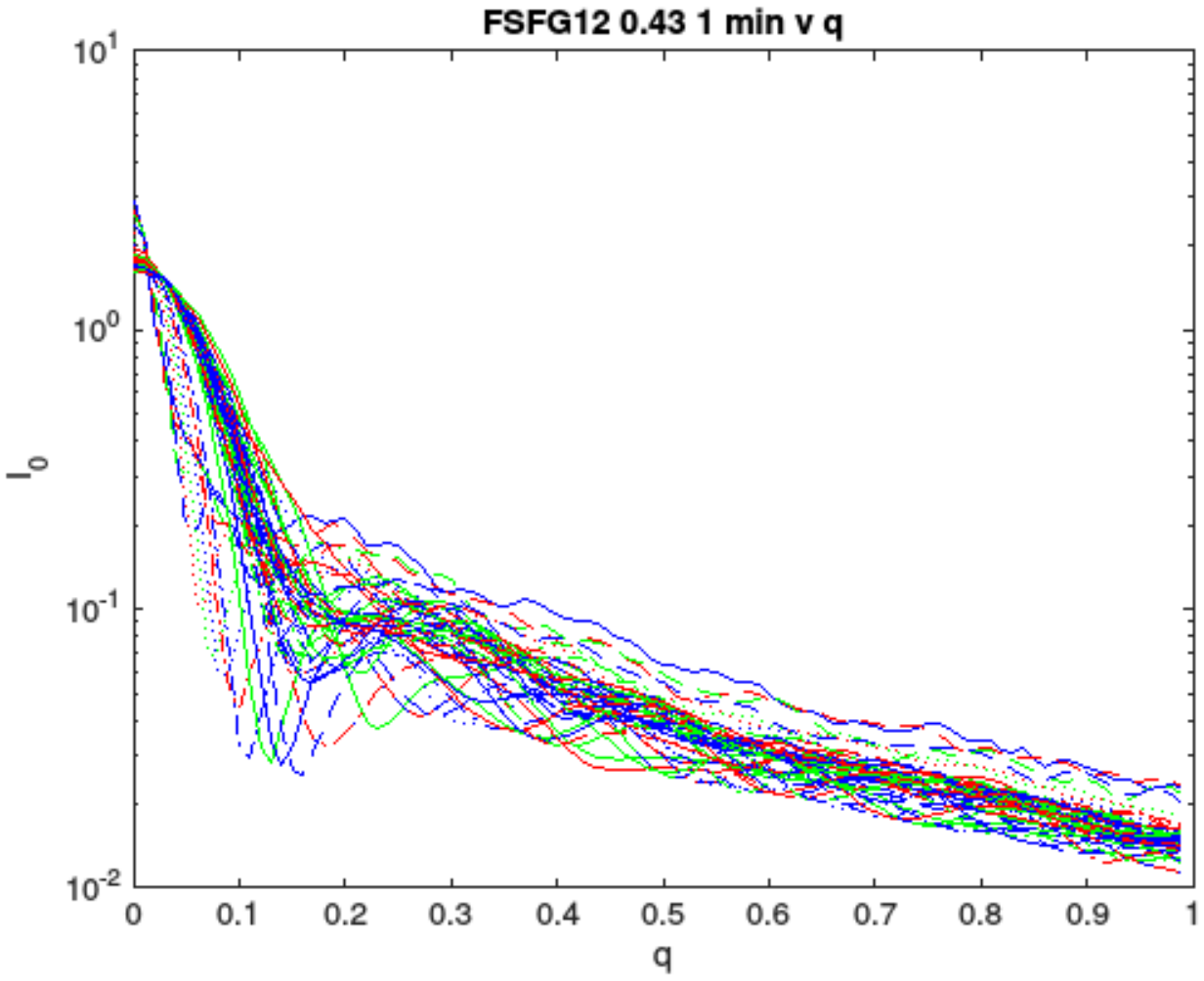
```
datetime
```

```
ans = datetime
   06-Sep-2023 14:06:09
```

```
h1=figure;
mask=true(size(I2,2),1);
for jj=1:numel(qset)
    [imax,ind]=max(I2(jj,mask));
    semilogy(qset,I2(:,ind));
    Pind(1)=ind;
    mask(ind)=false;
    if jj==1, hold on; end
end
xlabel('q'); ylabel('I_0'); title([ROOT ' ' fD2O ' ' fDeut ' max v q']);
axis([0 qset(end) mnis mxis]);
```

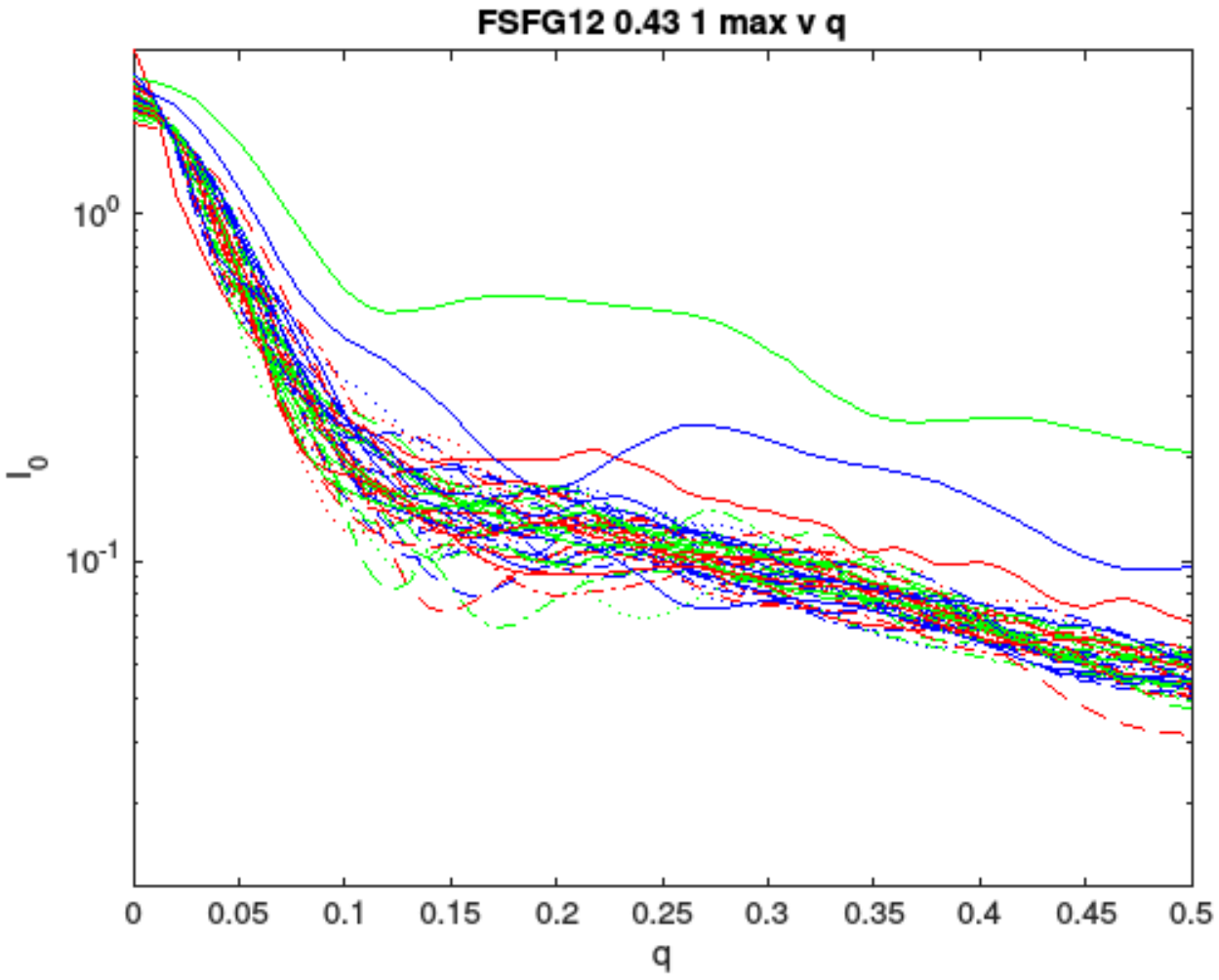



```
savefig(h1,'fig1','compact');
h2=figure;
for jj=1:numel(qset)
    [imax,ind]=min(I2(jj,:));
    semilogy(qset,I2(:,ind));
    Pind(2)=ind;
    if jj==1, hold on; end
end
%axis([0 qset(end) mnis mxis]);
xlabel('q'); ylabel('I_0'); title([ROOT ' ' fD2O ' ' fDeut ' min v q' ]);
```

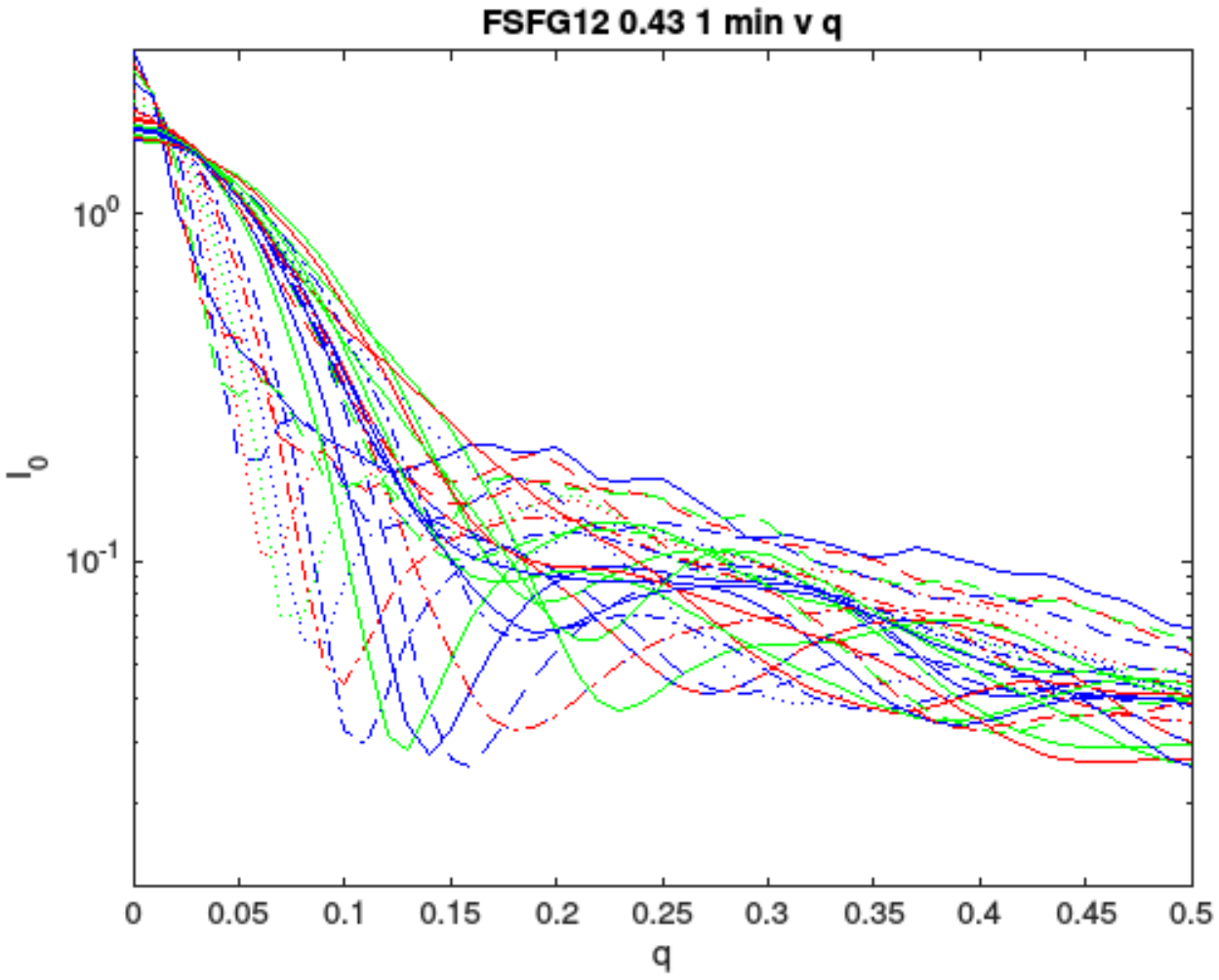


reduced range

```
savefig(h2,'fig2','compact');
qset2=qset(qset <= 0.5);
n2= numel(qset2);


h3=figure;
mask=true(size(iset,2),1) ;
for jj=1:numel(qset2)
    [imax,ind]=max(I2(jj,mask));
    semilogy(qset2,I2(1:n2,ind));
    Pind(jj,1)=ind;
    mask(ind)=false;
    if jj==1, hold on; end
end
xlabel('q'); ylabel('I_0'); title([ROOT ' ' fD2O ' ' fDeut ' max v q']);
axis([0 qset2(end) mnis mxis]);
```




```
savefig(h3,'fig3','compact');
h4=figure;
for jj=1:numel(qset2)
    [imax,ind]=min(I2(jj,:));
    semilogy(qset2,I2(1:n2,ind));
    Pind(jj,2)=ind; %#ok<*SAGROW>
    if jj==1, hold on; end
end
axis([0 qset2(end) mnis mxis]);
xlabel('q'); ylabel('I_0'); title([ROOT ' ' fD2O ' ' fDeut ' min v q' ]);
```




```
savefig(h4,'fig4','compact');

%save out
```

get structures

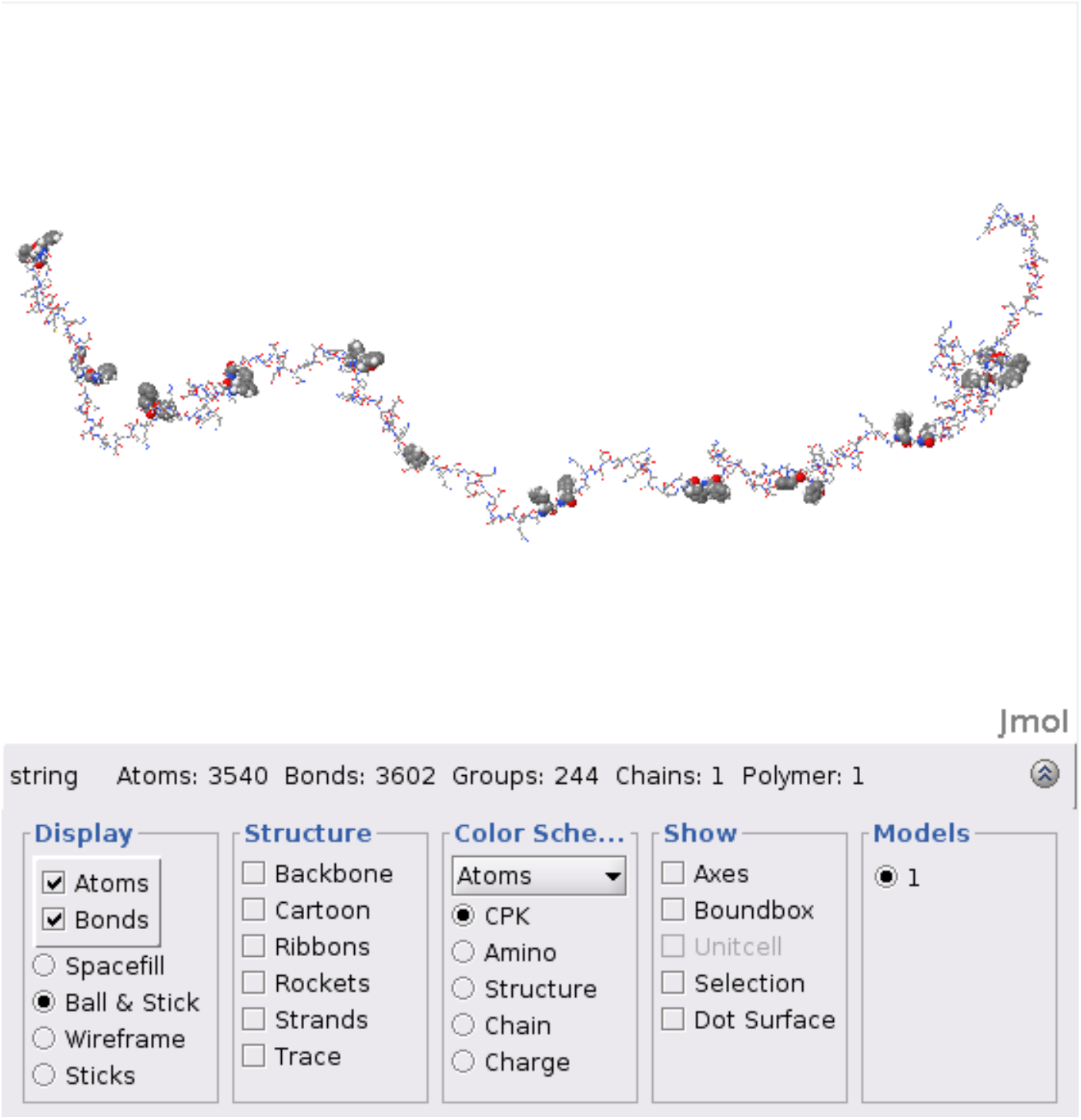
```
datetime
```

```
ans = datetime
   06-Sep-2023 14:06:21
```

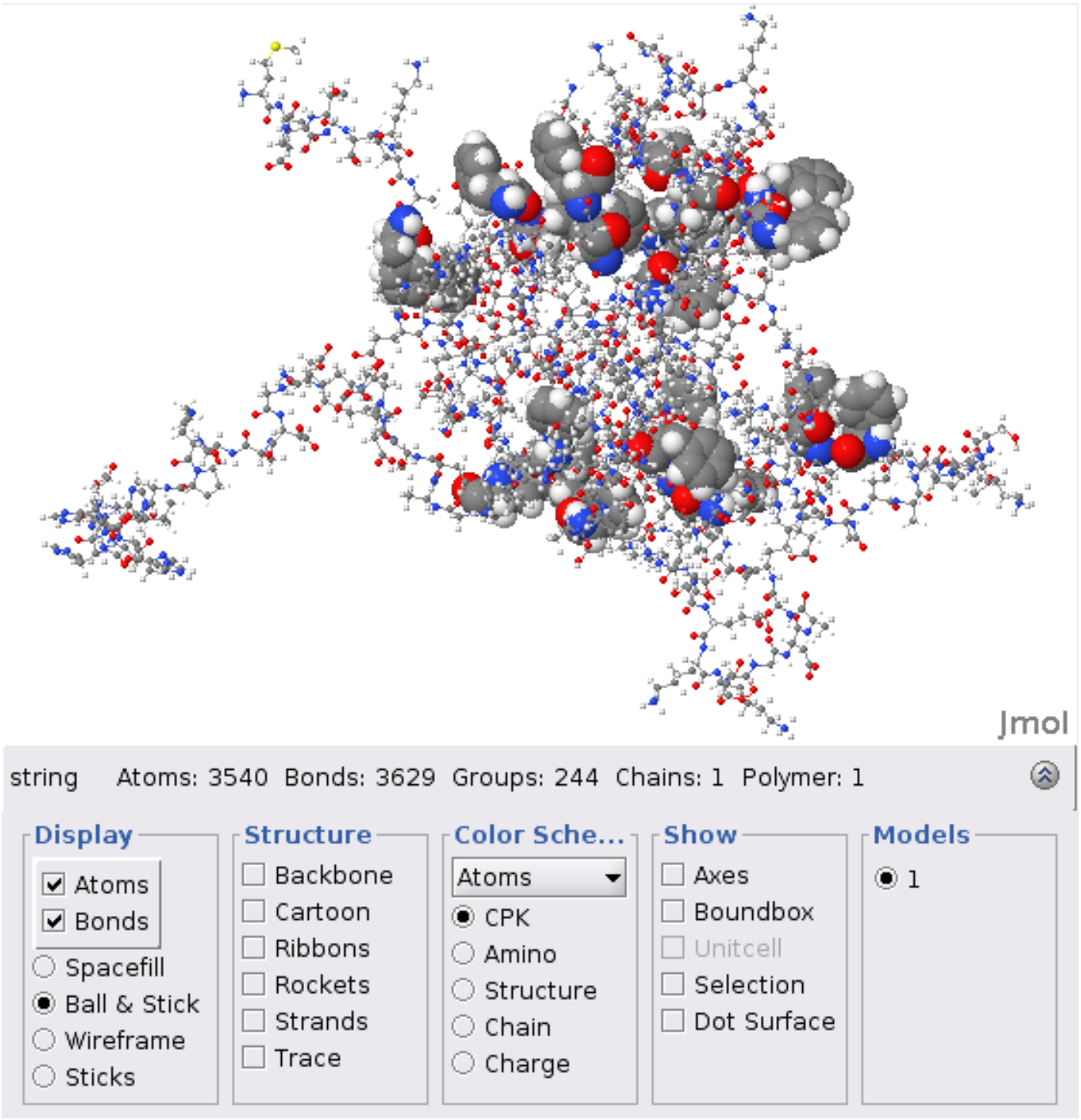
```
fS=@(x) strip(num2str(x));
for ii=1:5
    inds=Pind(ii,:);
    sii=fS(ii);
    u=gettmppdb(inds(1));
     disp(num2str([inds(1)  R(inds(1))]));
    movefile ('tmpget.pdb', ['tmp' sii 'X.pdb']);
    u=gettmppdb(inds(2 ));
      disp(num2str([inds(2)  R(inds(2   ))]));
```

```
    movefile('tmpget.pdb',['tmp' sii 'm.pdb']);
end
```

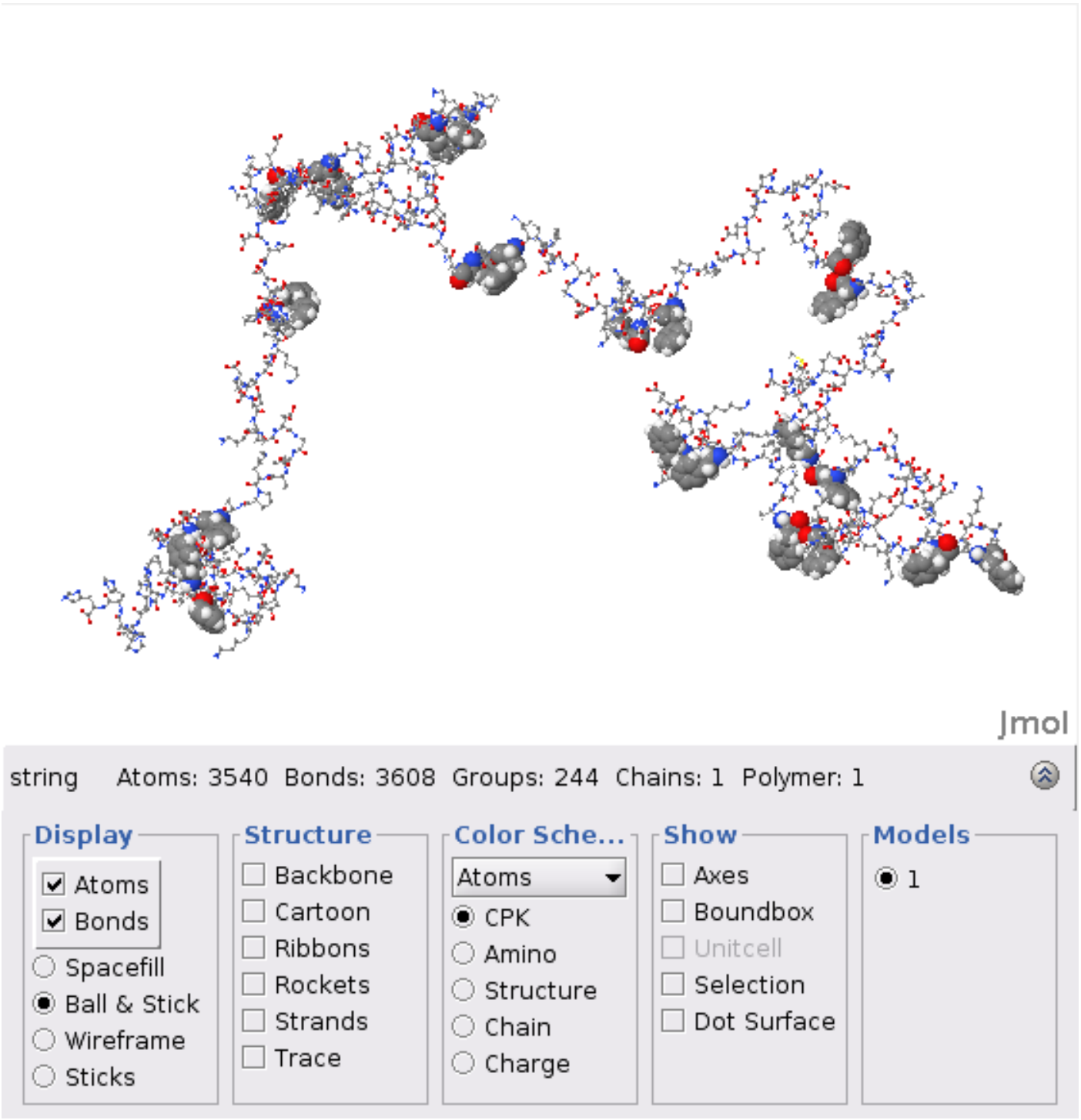
```
Warning: MOLVIEWER will be removed in a future release.
```



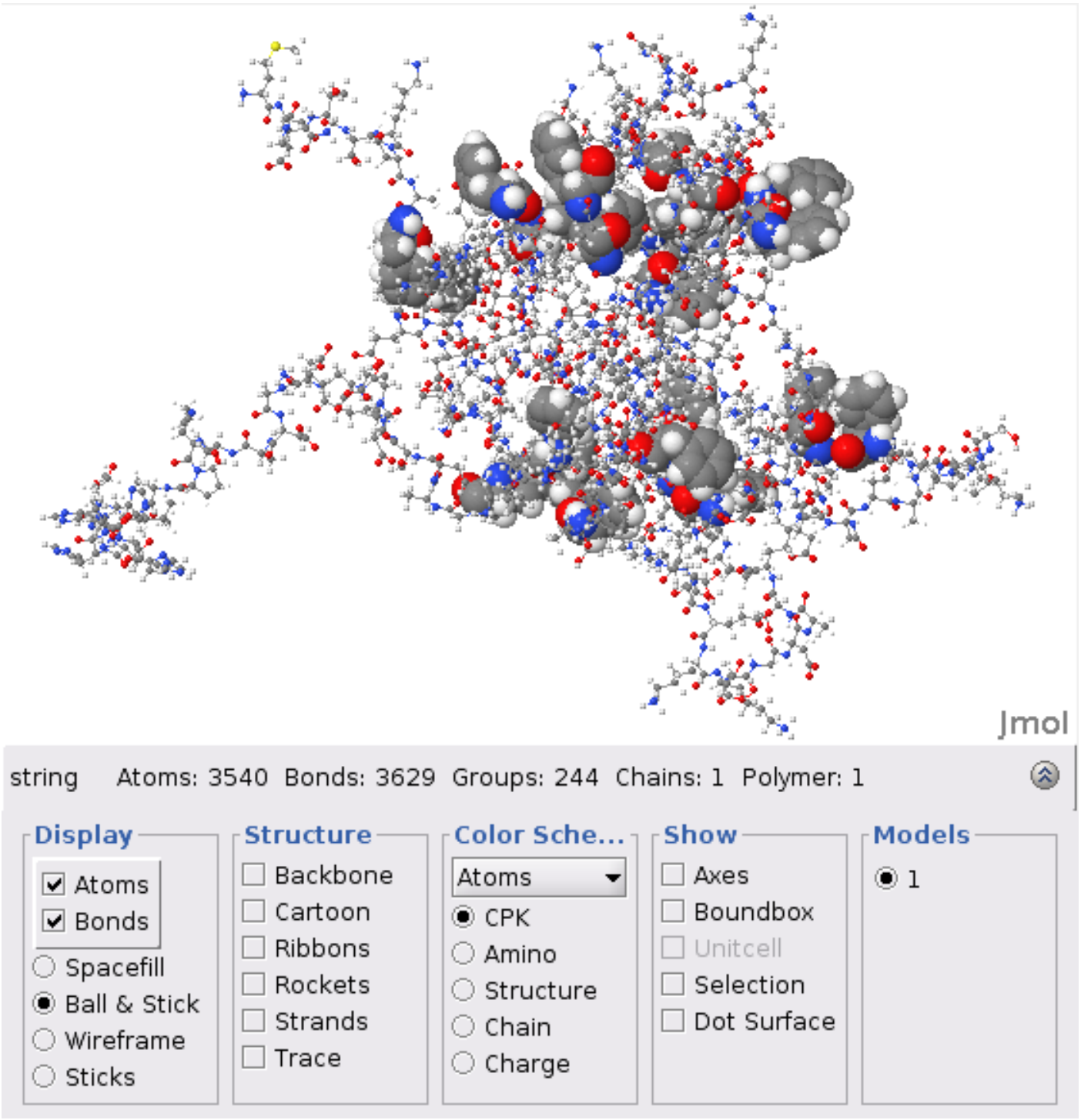

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1669209      99.369239807
Warning: MOLVIEWER will be removed in a future release.
```



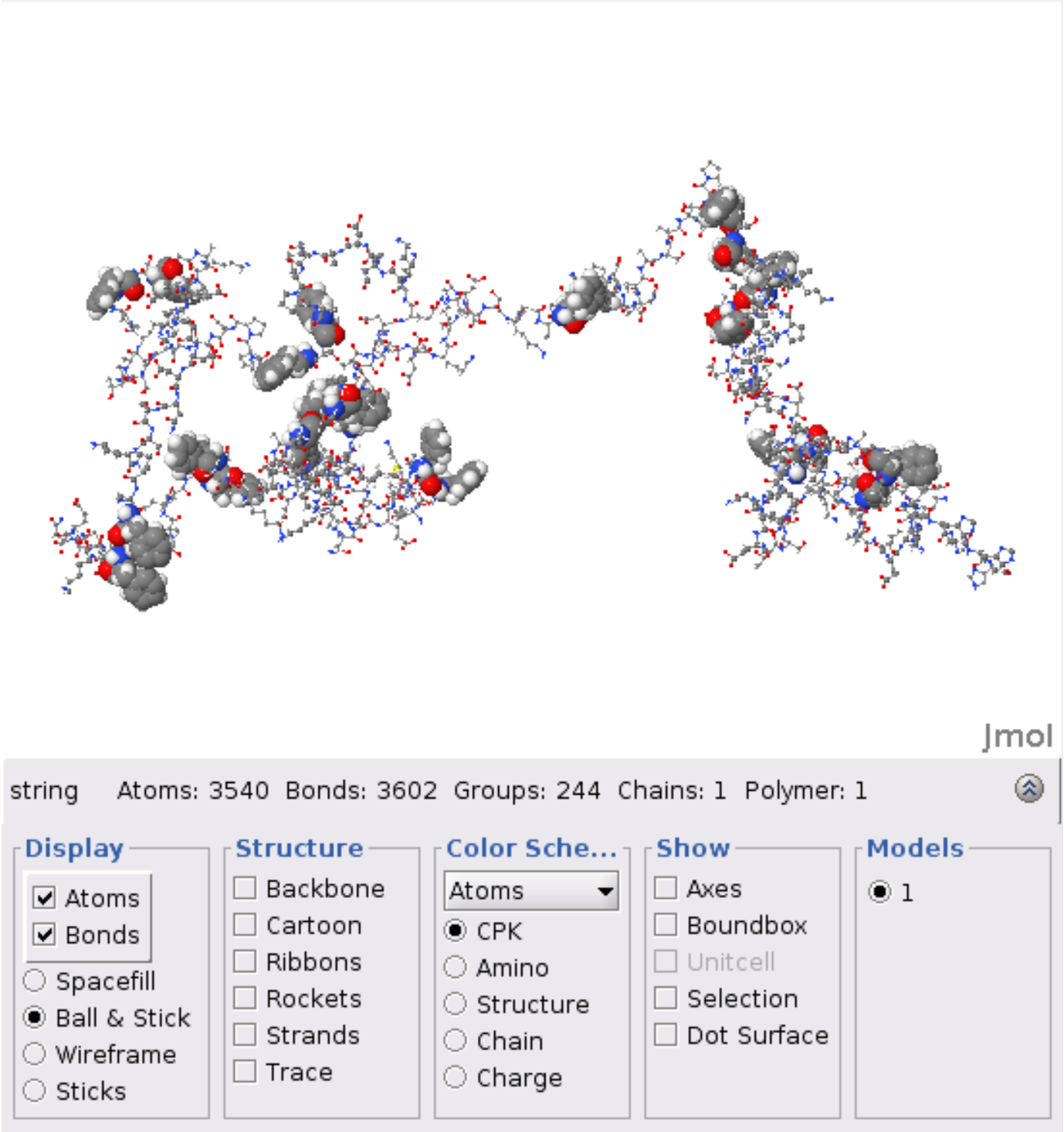

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1142651      16.972011566
Warning: MOLVIEWER will be removed in a future release.
```



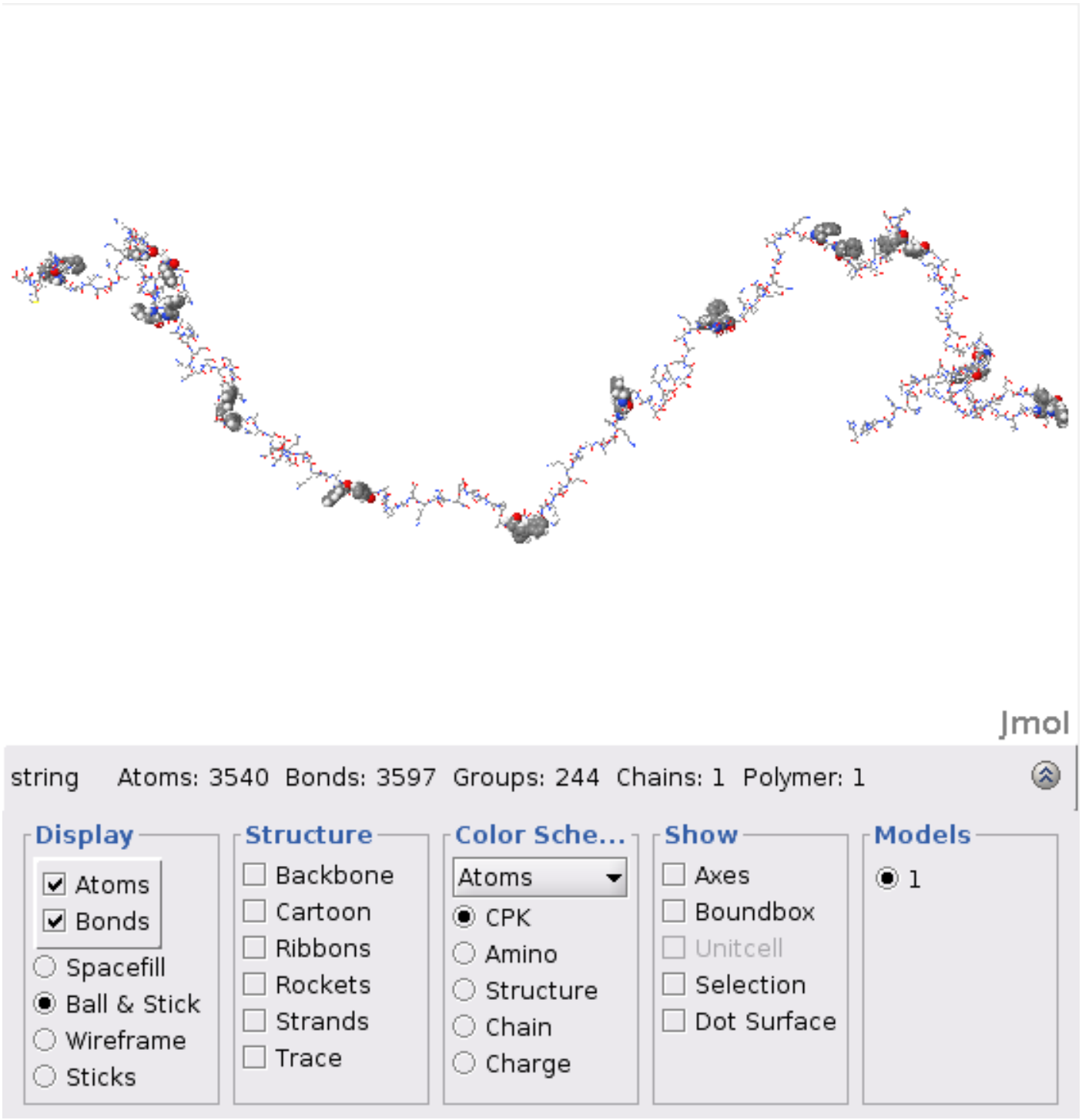

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
44685       23.1513748
Warning: MOLVIEWER will be removed in a future release.
```



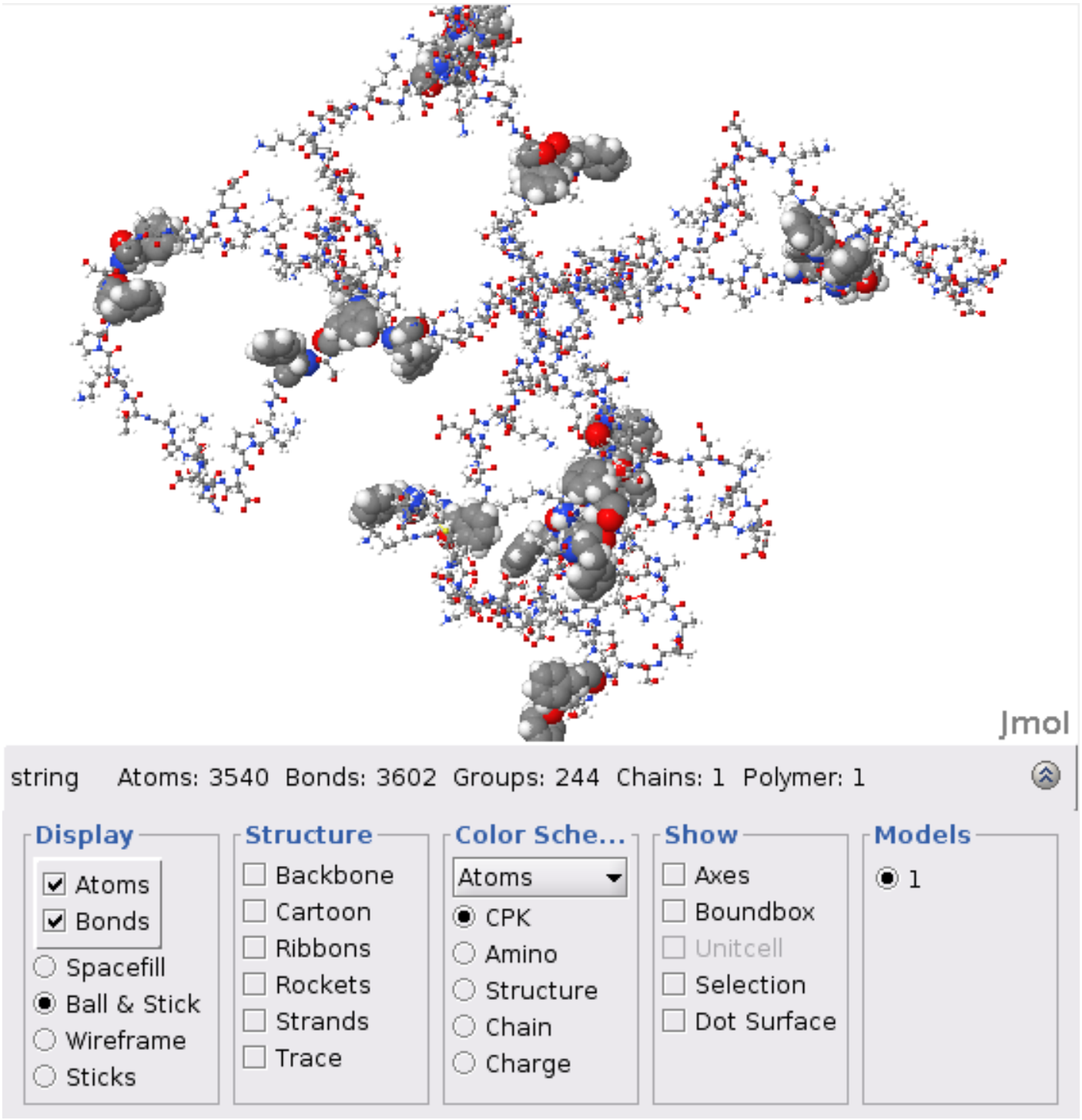

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1142651      16.972011566
Warning: MOLVIEWER will be removed in a future release.
```



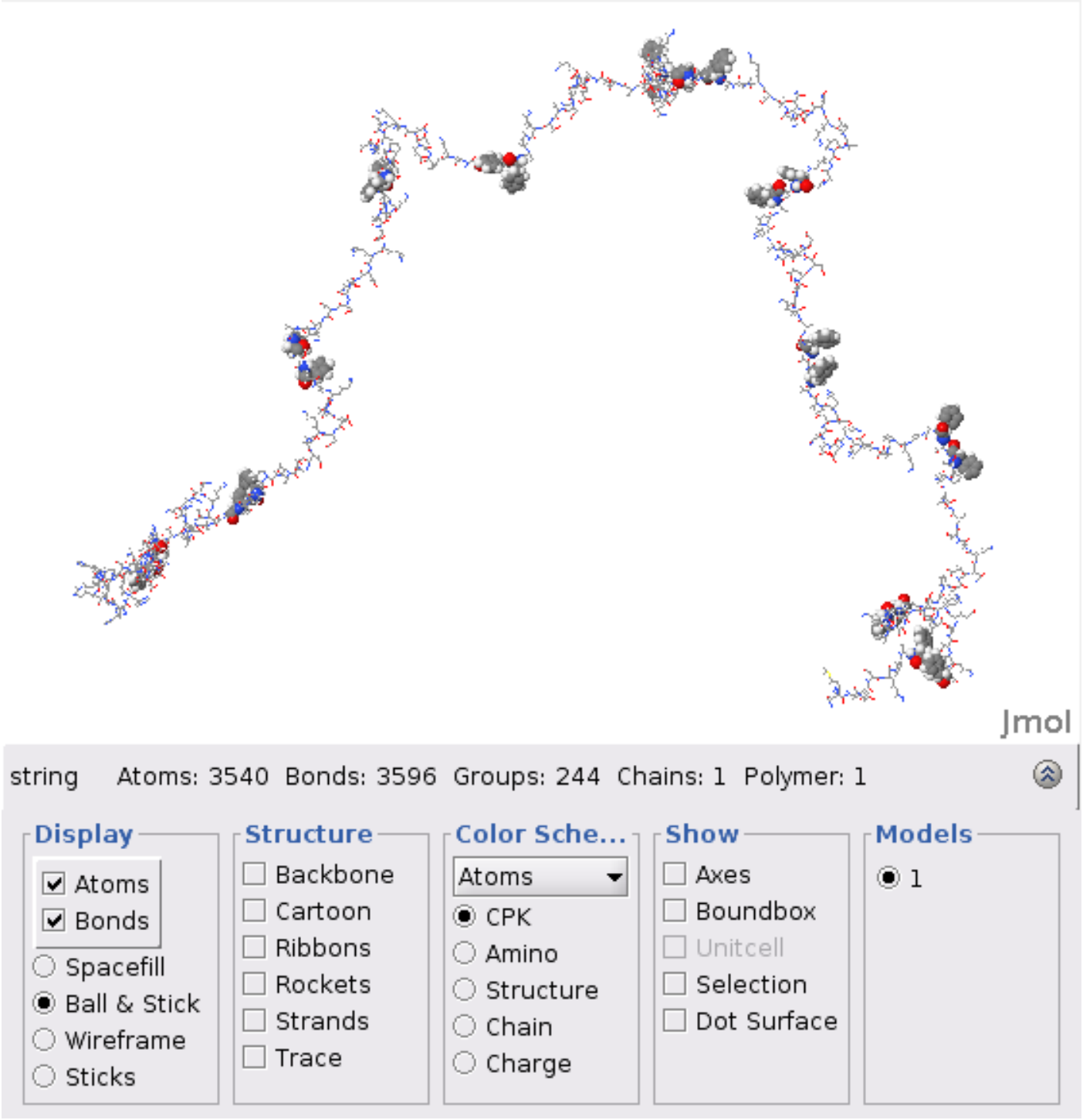
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
50 28.8684
Warning: MOLVIEWER will be removed in a future release.



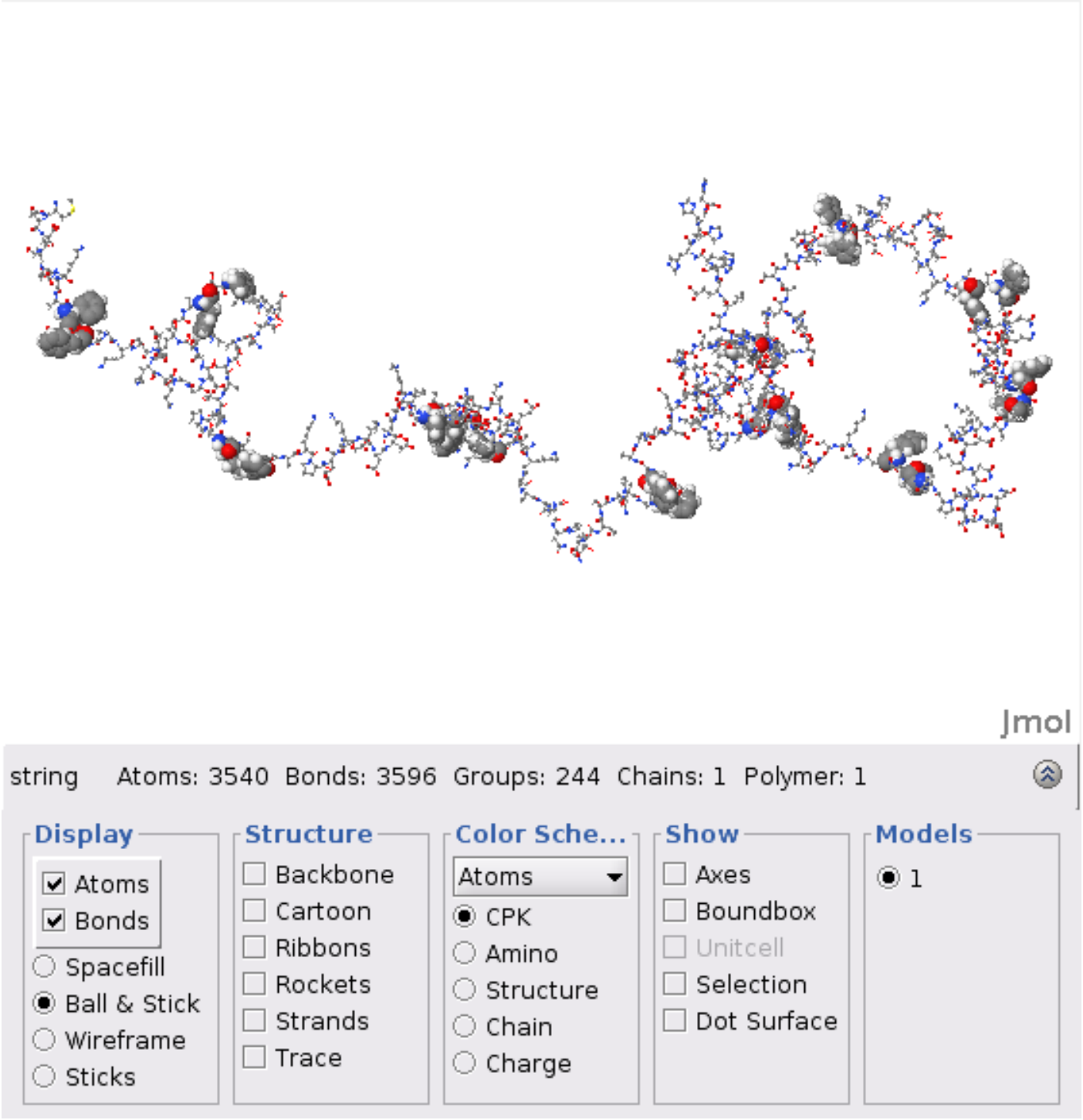

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1239893       99.42199707
Warning: MOLVIEWER will be removed in a future release.
```



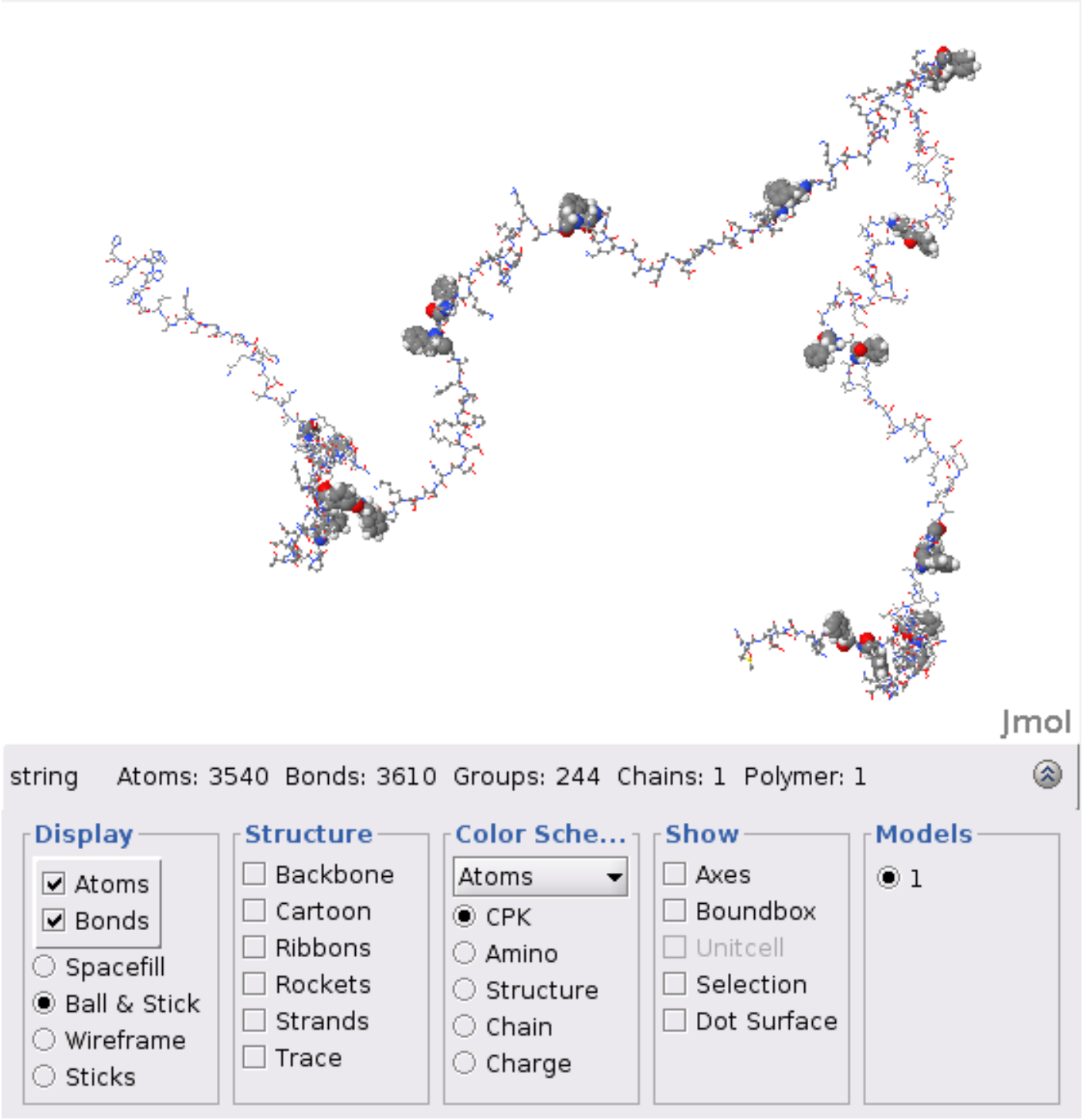

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1009867      31.733154297
Warning: MOLVIEWER will be removed in a future release.
```




```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
477494      77.09577942
Warning: MOLVIEWER will be removed in a future release.
```




```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1370820      57.321182251
Warning: MOLVIEWER will be removed in a future release.
```



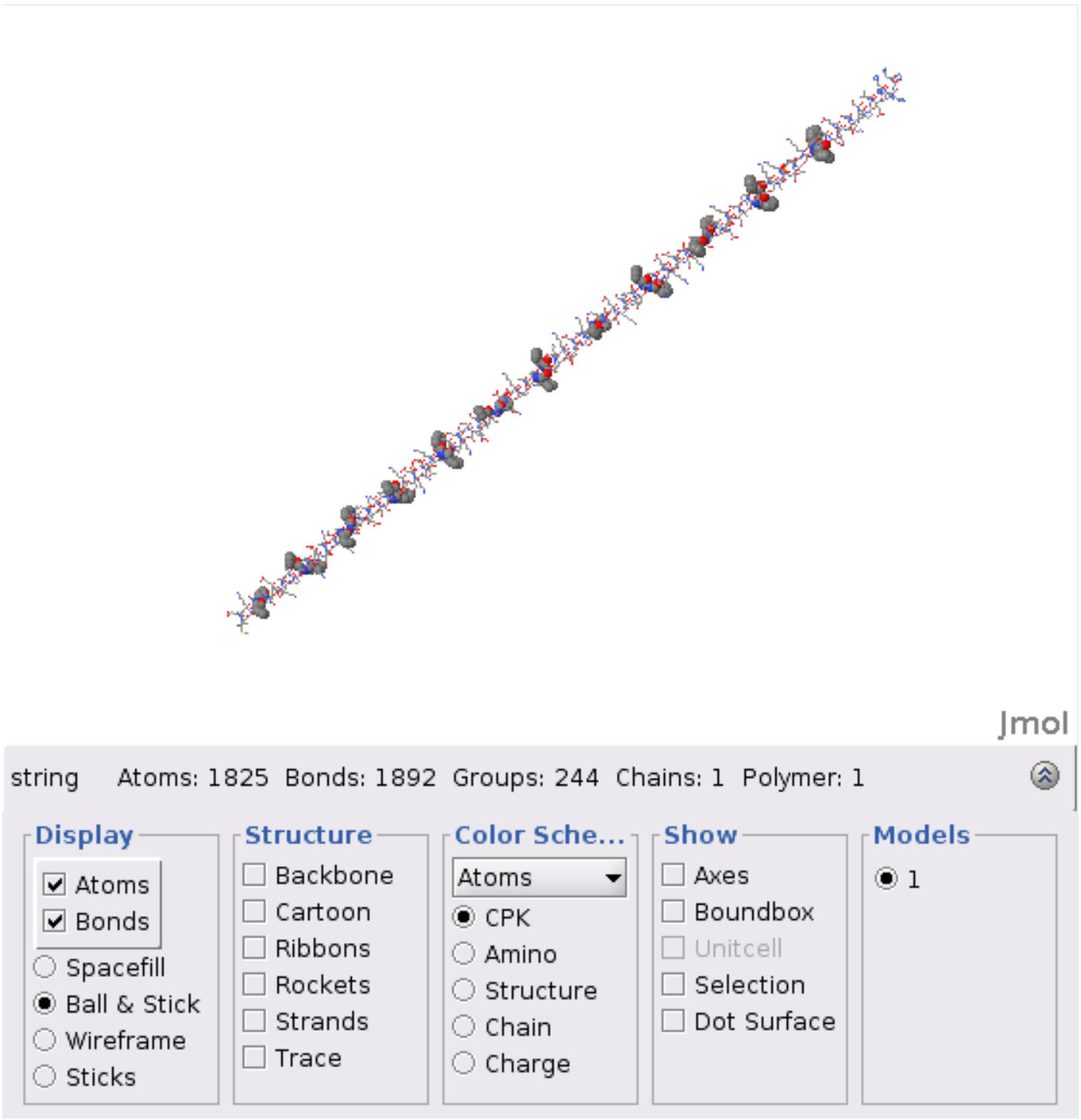

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
1214997      66.142753601
```

new members

```
fS=@(x) strip(num2str(x));
for ii=2e6+1:2e6+6
```

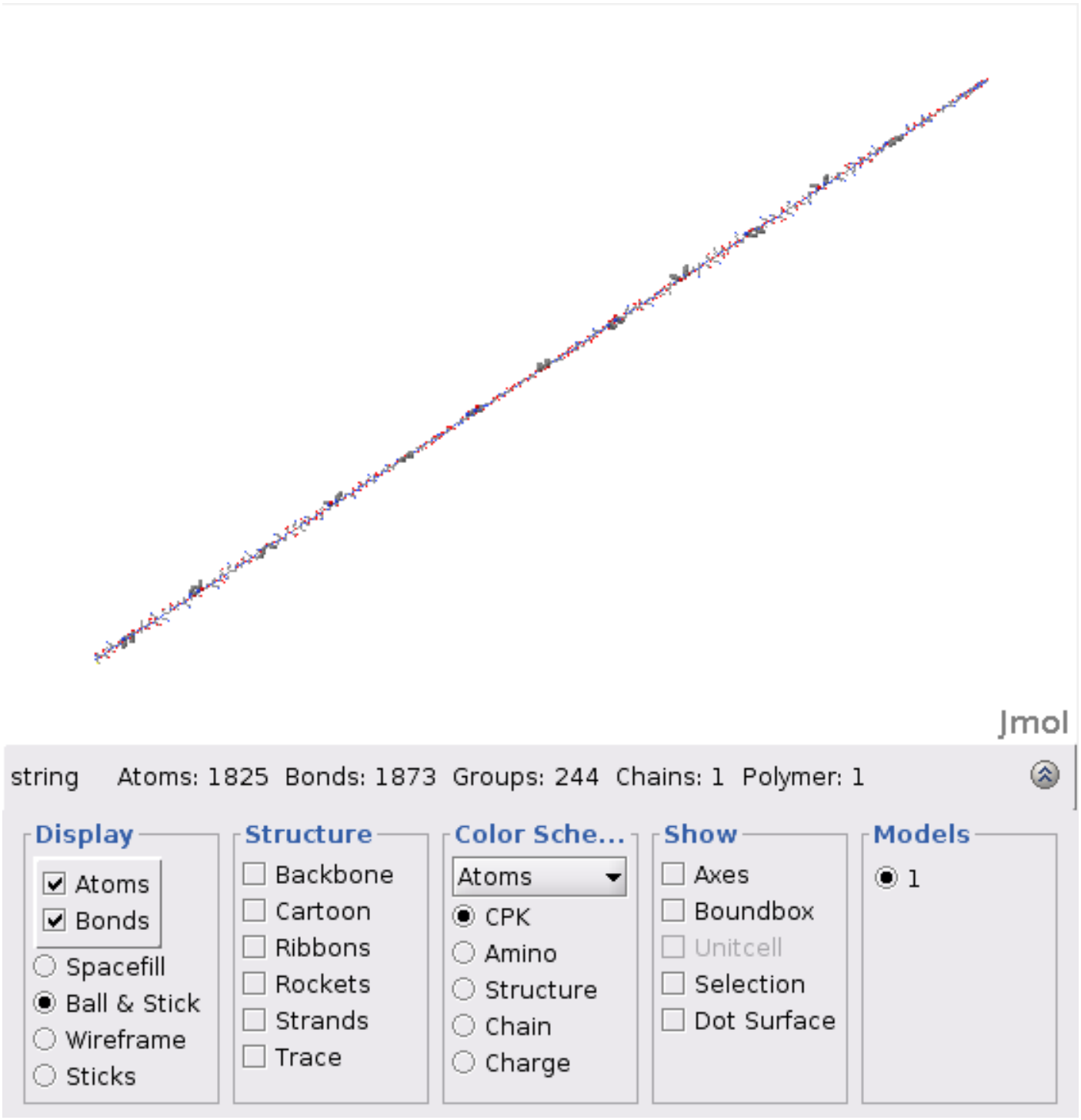
```
    sii=fS(ii);
    u=gettmppdb(ii);
     disp(num2str([ii    R(ii)]));
 %   movefile ('tmpget.pdb', ['tmp' sii 'X.pdb']);
 %   u=gettmppdb(inds(2 ));
%      disp(num2str([inds(2)  R(inds(2   ))]));
 %   movefile('tmpget.pdb',['tmp' sii 'm.pdb']);
end
```

```
Warning: MOLVIEWER will be removed in a future release.
```

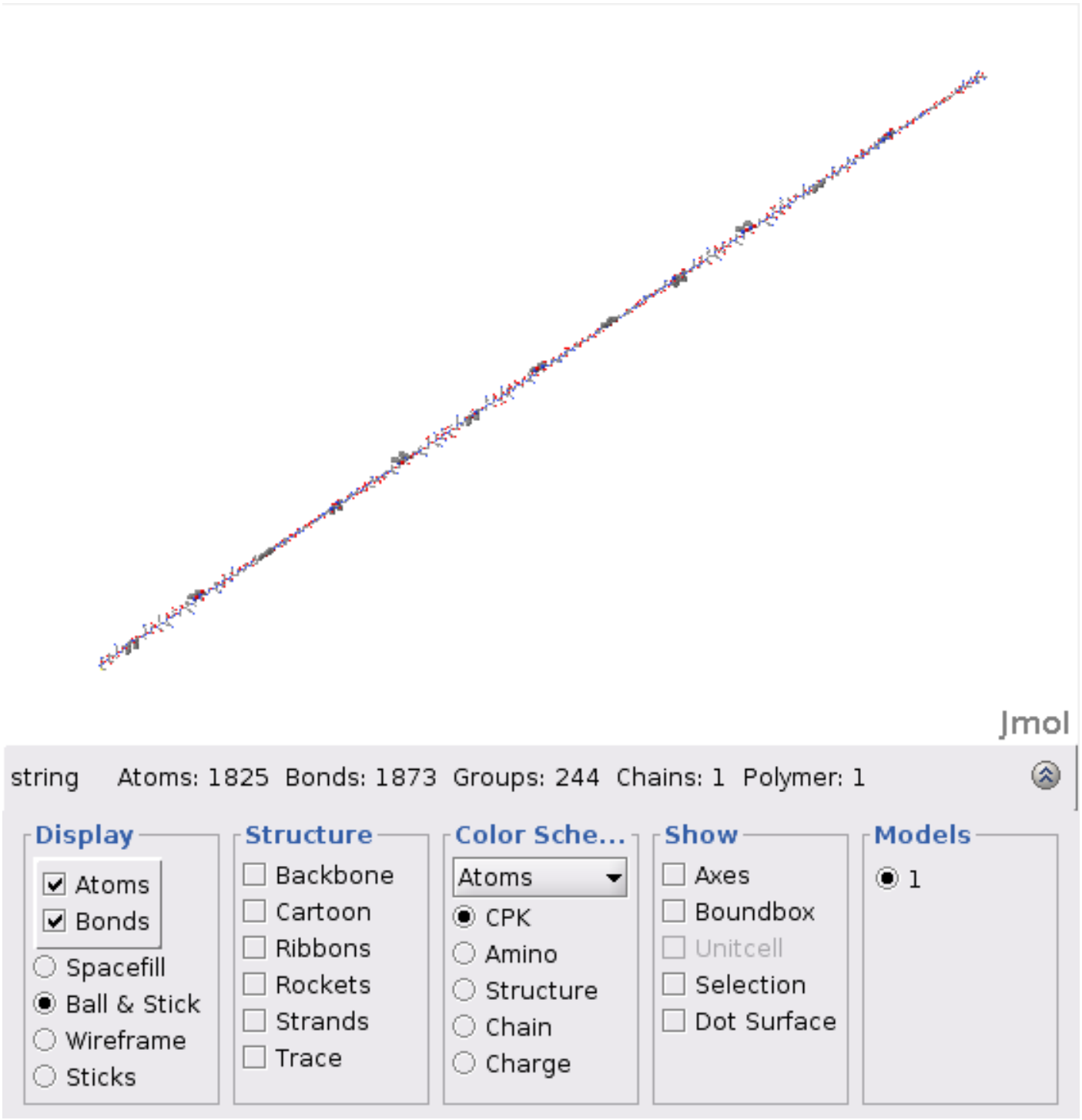



```
Warning: All the mandatory fields are missing from the input PDB structure.
```

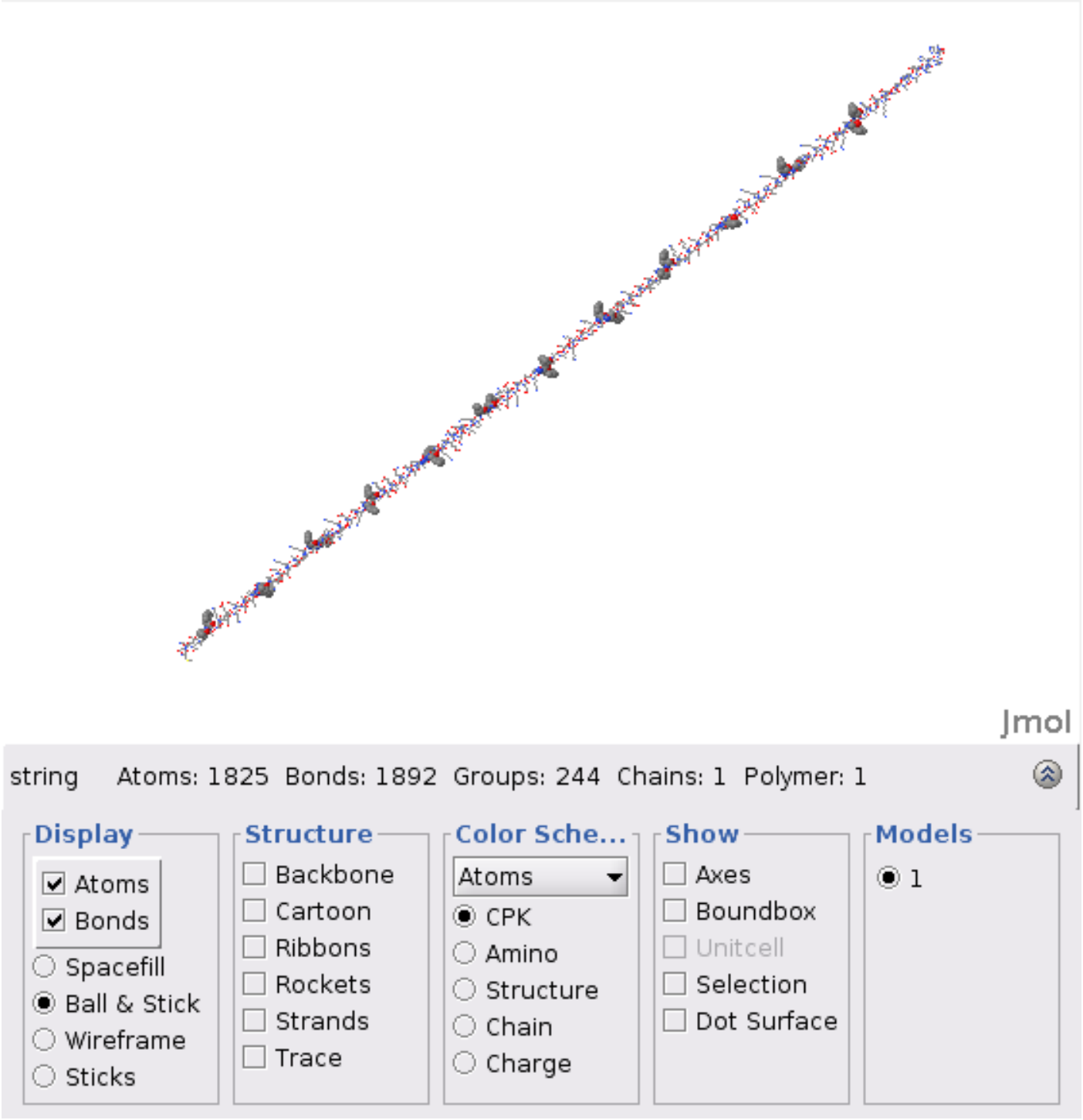
```
Warning: EVALRASMOLSCRIPT will be removed in a future release.
2000001       51.19789505
Warning: MOLVIEWER will be removed in a future release.
```



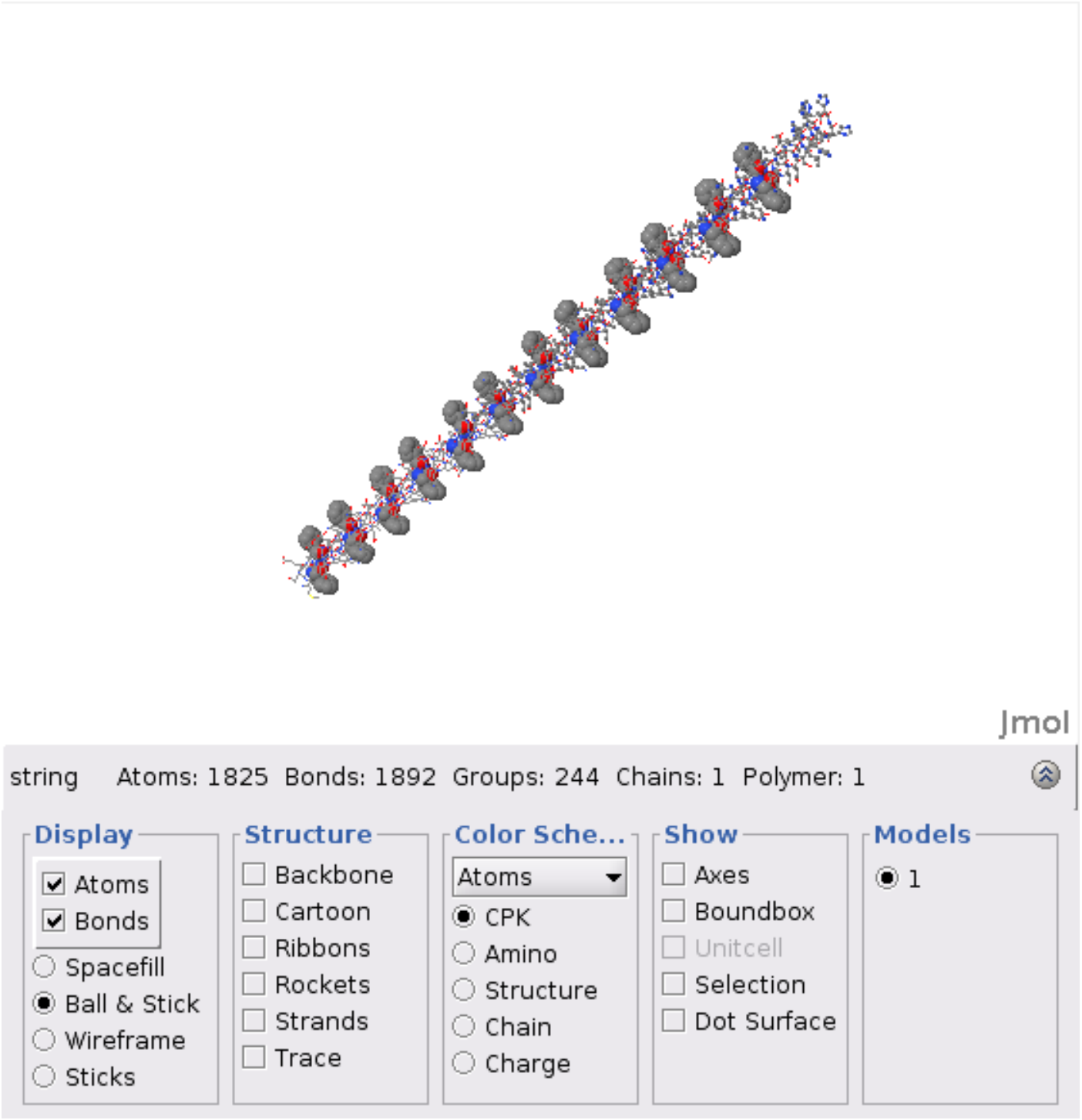

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
2000002       36.006622314
Warning: MOLVIEWER will be removed in a future release.
```



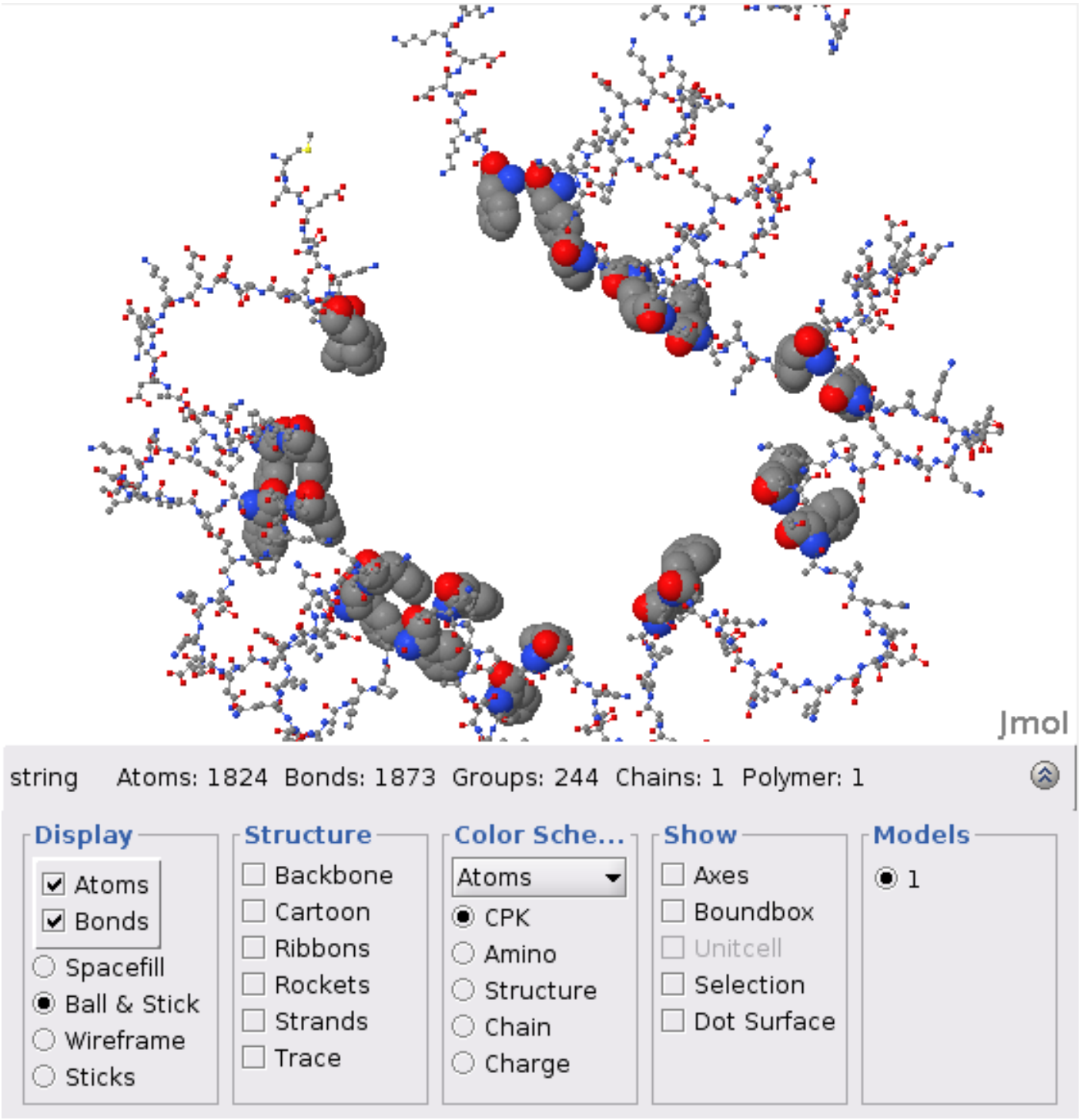

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
2000003      39.223724365
Warning: MOLVIEWER will be removed in a future release.
```




```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
2000004      37.658313751
Warning: MOLVIEWER will be removed in a future release.
```



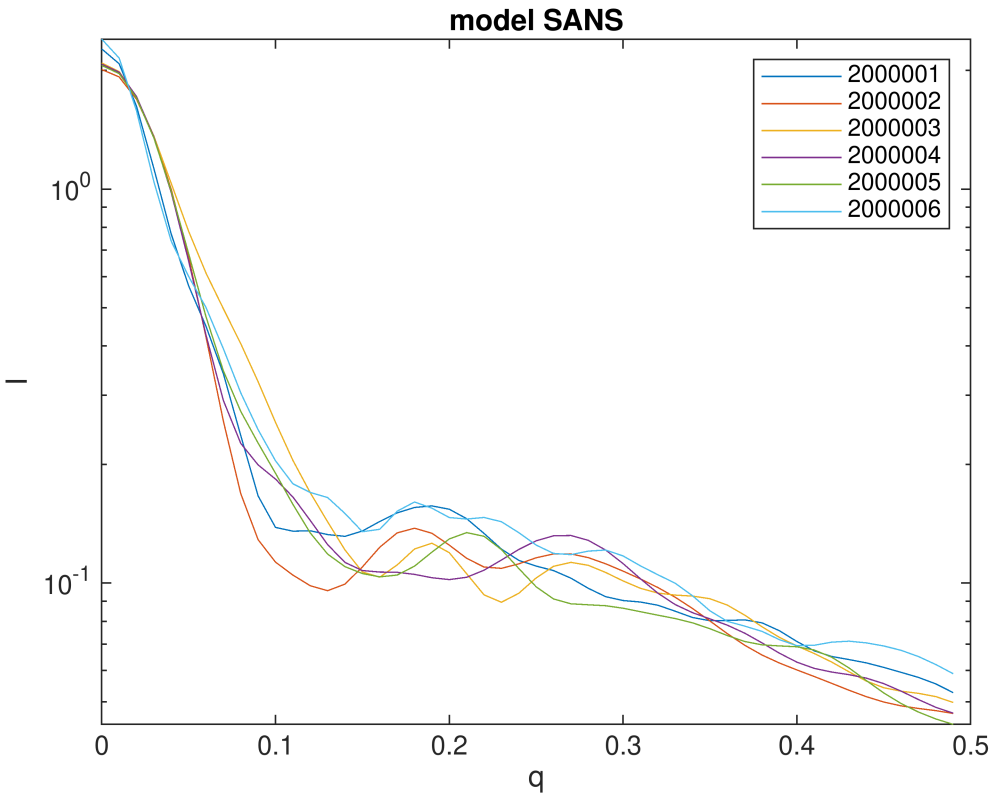

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
2000005      37.945552826
Warning: MOLVIEWER will be removed in a future release.
```



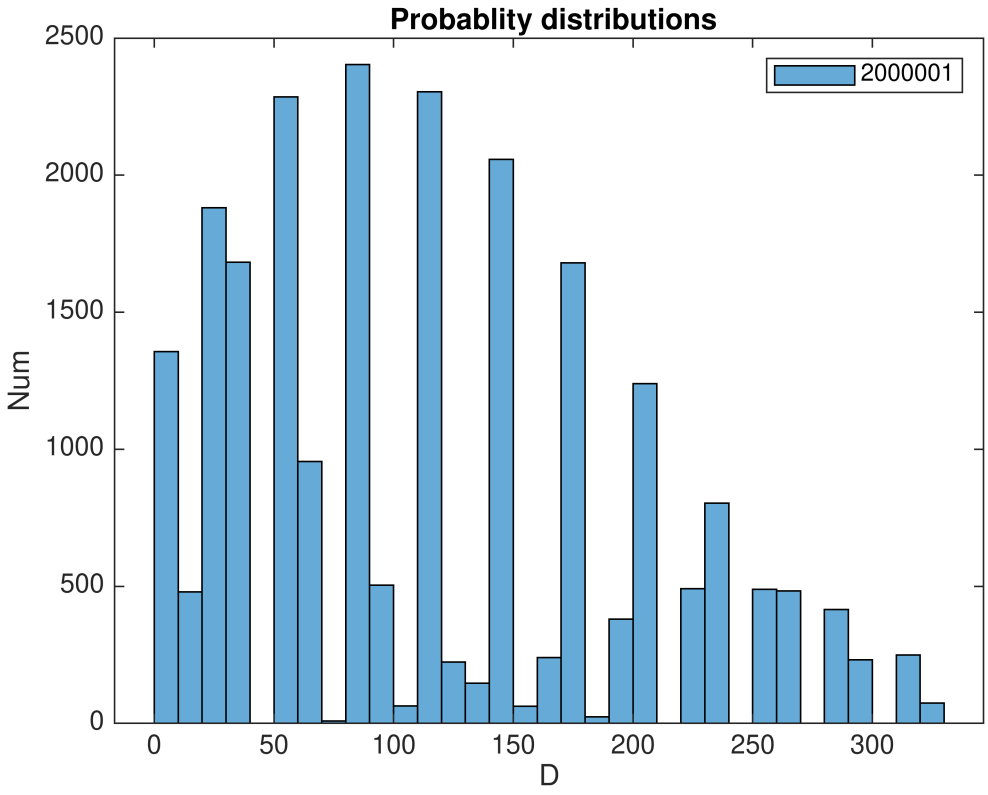

```
Warning: All the mandatory fields are missing from the input PDB structure.
Warning: EVALRASMOLSCRIPT will be removed in a future release.
2000006      57.939476013
```

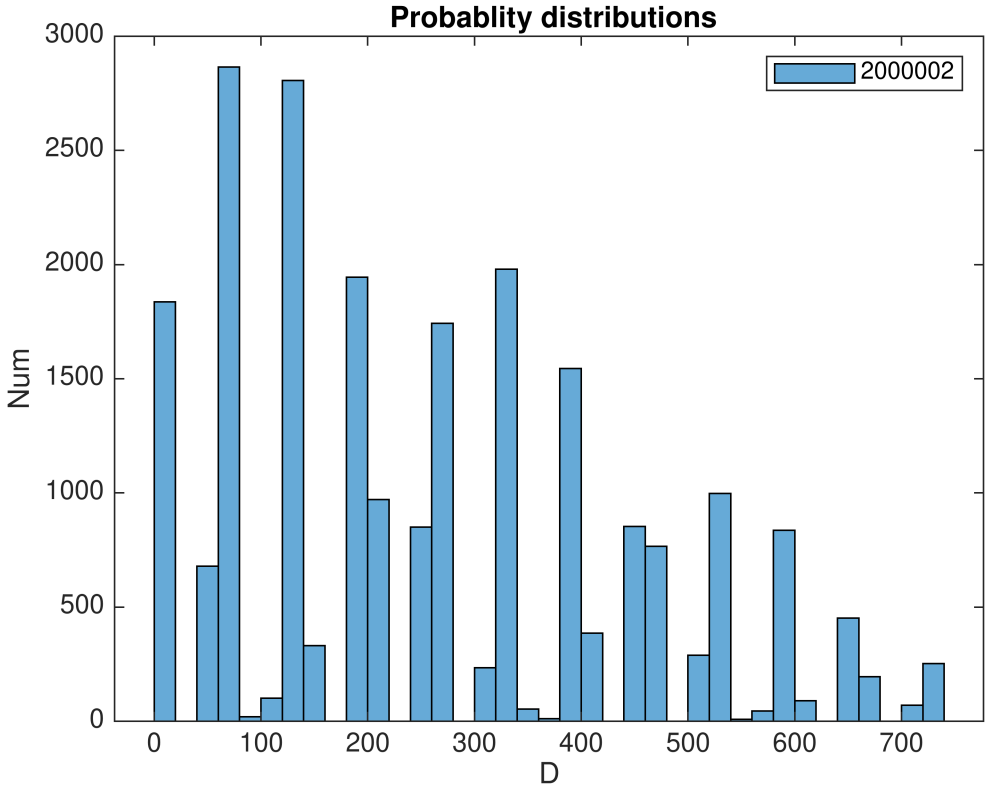
```
figure; legs=[];
ran=1:50;
for ii=1:6
    jj=ii+2e6;
    qq=out(jj).Q;
    semilogy(qq(ran,1),qq(ran,2));
```

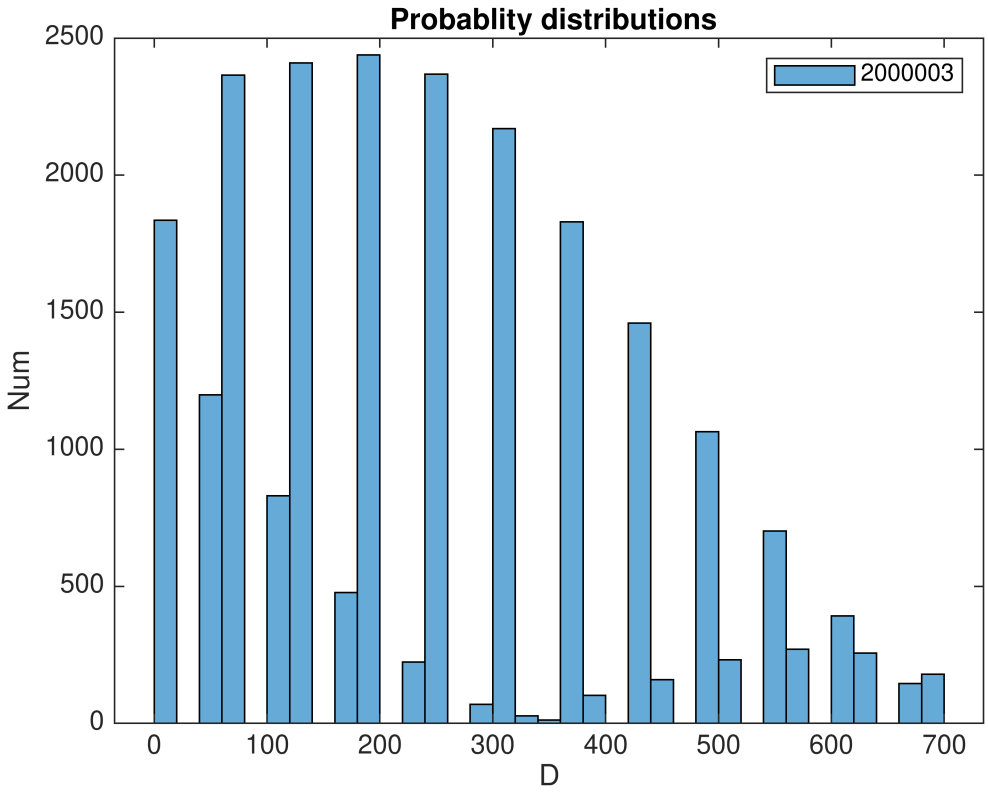
```
    legs{ii}=num2str(jj);
    hold on;
end
xlabel('q'); ylabel('I'); title ('model SANS'); legend(legs);
```

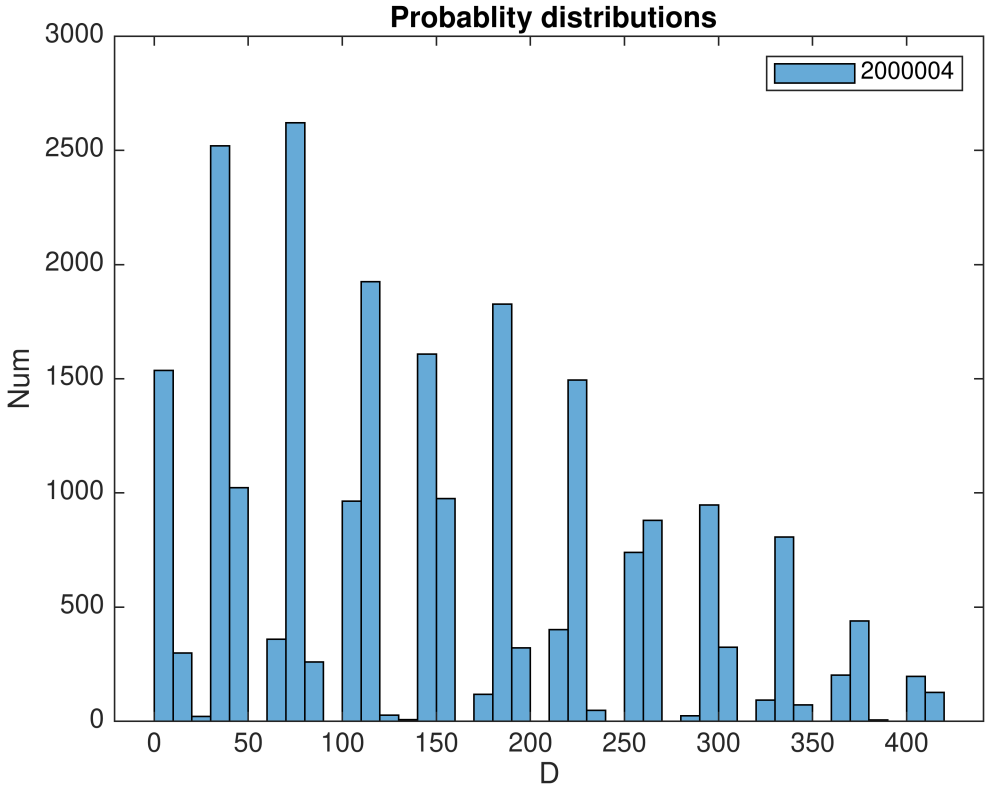


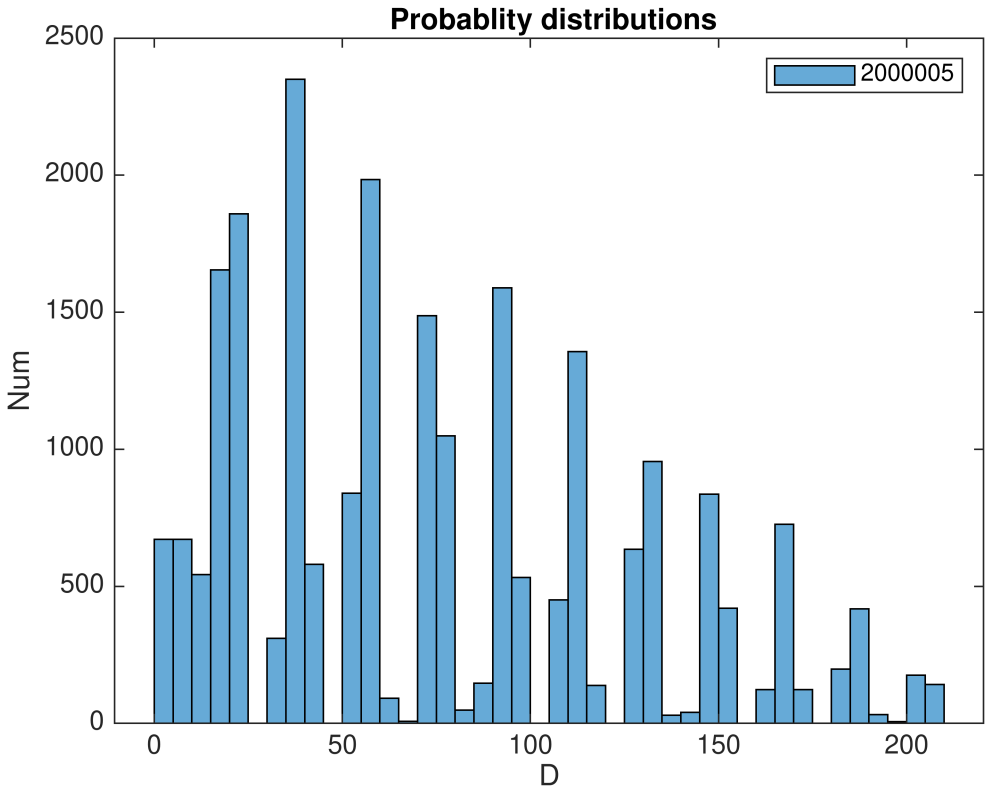

```
bdis=FFdis(2000001:2000006);

for ii=2000001:2000006
    tmp=bdis{ii};
    mx=max([mx ; tmp(:)]);
end
for ii=2000001:2000006
    tmp=bdis{ii}; figure; legs=[]; mx=0; ilegs=1;
    tmp=tmp(tmp(:)>0);
    histogram(tmp);% hold on;
    legs{ilegs}=int2str(ii);
    ilegs=ilegs+1;
    xlabel('D'); ylabel('Num'); title ( ' Probablity distributions');
legend(legs);
end
```

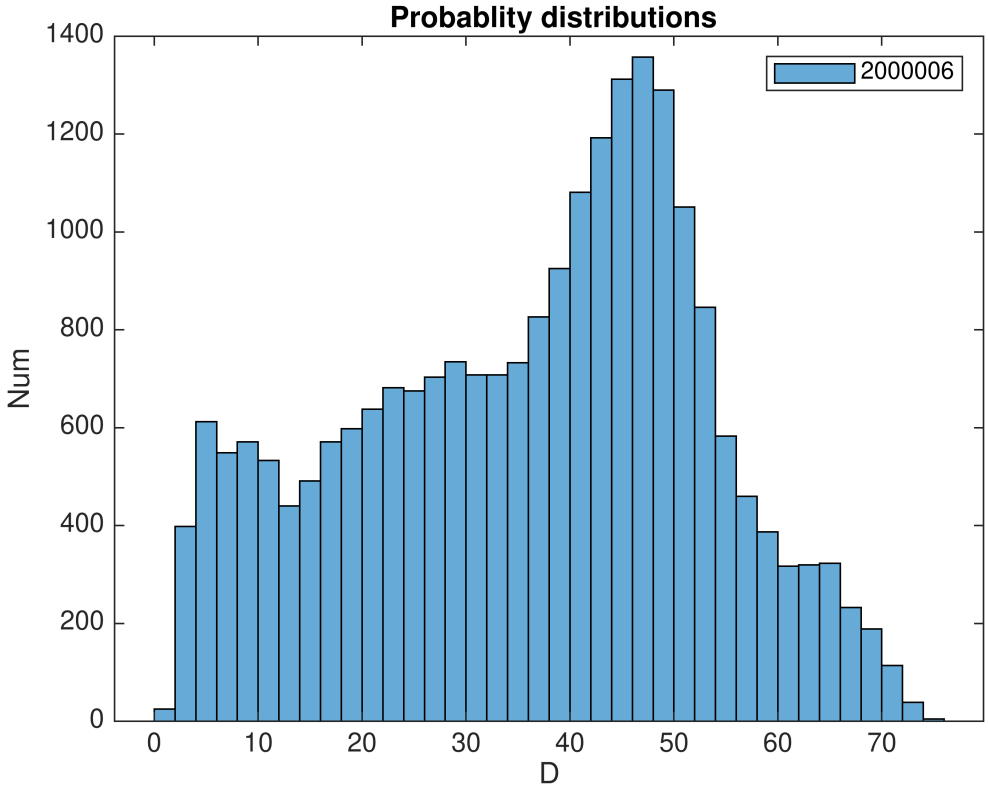

Probablity distributions
2000001
Num
D
0
500
1000
1500
2000
2500
0
50
100
150
200
250
300


Probablity distributions
2000002
Num
D
0
500
1000
1500
2000
2500
3000
0
100
200
300
400
500
600
700


Probablity distributions
2000003
Num
D
0
500
1000
1500
2000
2500
0
100
200
300
400
500
600
700


Probablity distributions
2000004
Num
D
0
500
1000
1500
2000
2500
3000
0
50
100
150
200
250
300
350
400


Probablity distributions
2000005
Num
D
0
500
1000
1500
2000
2500
0
50
100
150
200


Probablity distributions
2000006
Num
D
0
200
400
600
800
1000
1200
1400
0
10
20
30
40
50
60
70

```
return
```