NMR studies of protein-small molecule and protein-peptide interactions

Guan, J.

Appendix A: 1D-^{1}H NMR spectra of FKBP12 ligand titrations

General description: The proton assignments correspond to the numbers on the ligand structures shown in each figure. Signal intensities were scaled non-linearly for ease of viewing the shapes of the resonances. Asterisks indicate the protein/buffer/solvent resonances. Table A1 summarizes the details of the titration experiments shown in Figures A1-A8.

Table A1: Details of the titration experiments in Figures A1-A8.

Figure	ZB code	Initial [P] (μM)	Final [L] (mM)	Bound [L] at end point (%)	T_2 filter (ms)	K_D
A1	1	50	3.0	1.6	20	83 μM
A2	3	50	1.8	2.7	20	41 μM
A3	390	50	2.5	0.8	60	3.6 mM
A4	429	100	1.0	2.9	60	2.4 mM
A5	1051	100	1.0	n.d.[a)]	0.5	n.d.
A6	1104	100	1.0	n.d.	60	n.d.
A7	1406	100	1.0	n.d.	0.5	n.d.
A8	1489	100	1.0	n.d.	20	n.d.

.[a)] n.d.: not determined.

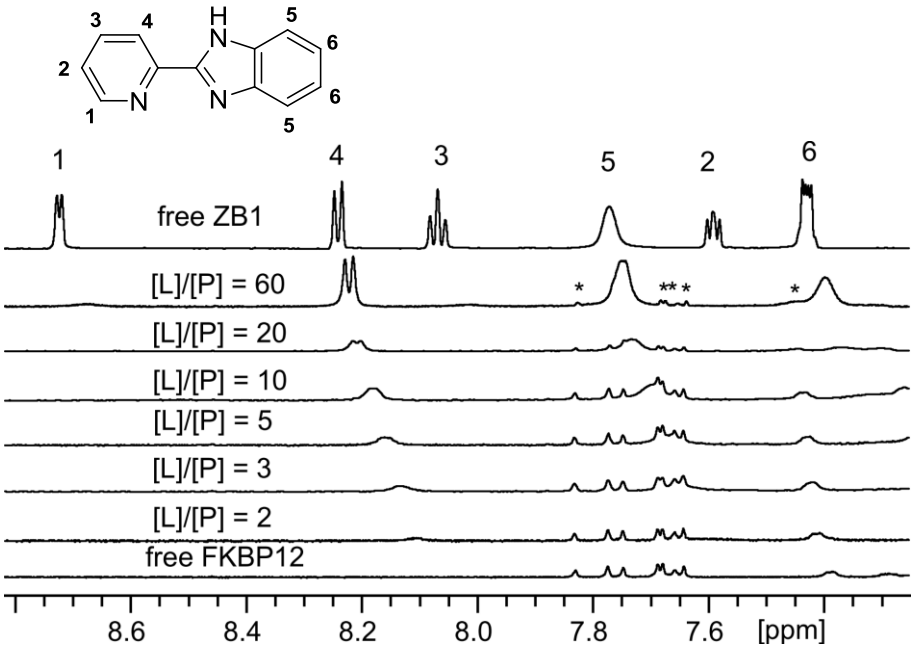

Figure A1: 1D-^{1}H NMR spectra of ZB1 titrated into wt FKBP12.

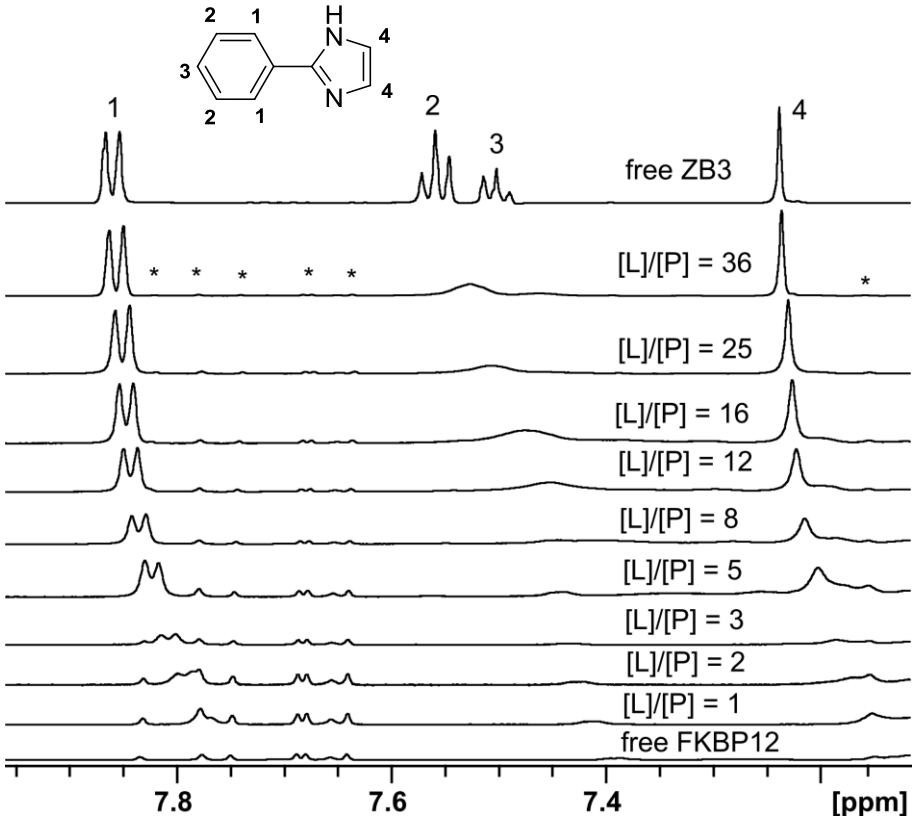

Figure A2: 1D-^{1}H NMR spectra of ZB3 titrated into wt FKBP12.

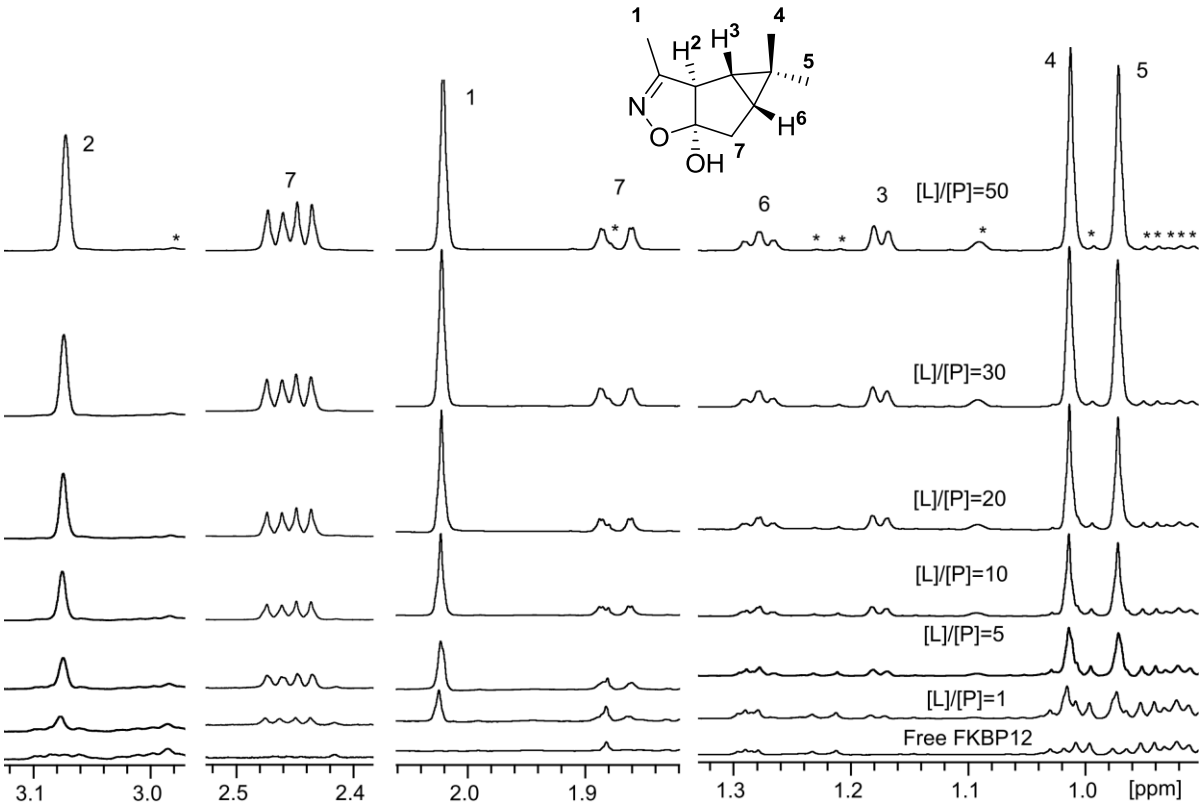

Figure A3: 1D-^{1}H NMR spectra of ZB390 titrated into wt FKBP12.

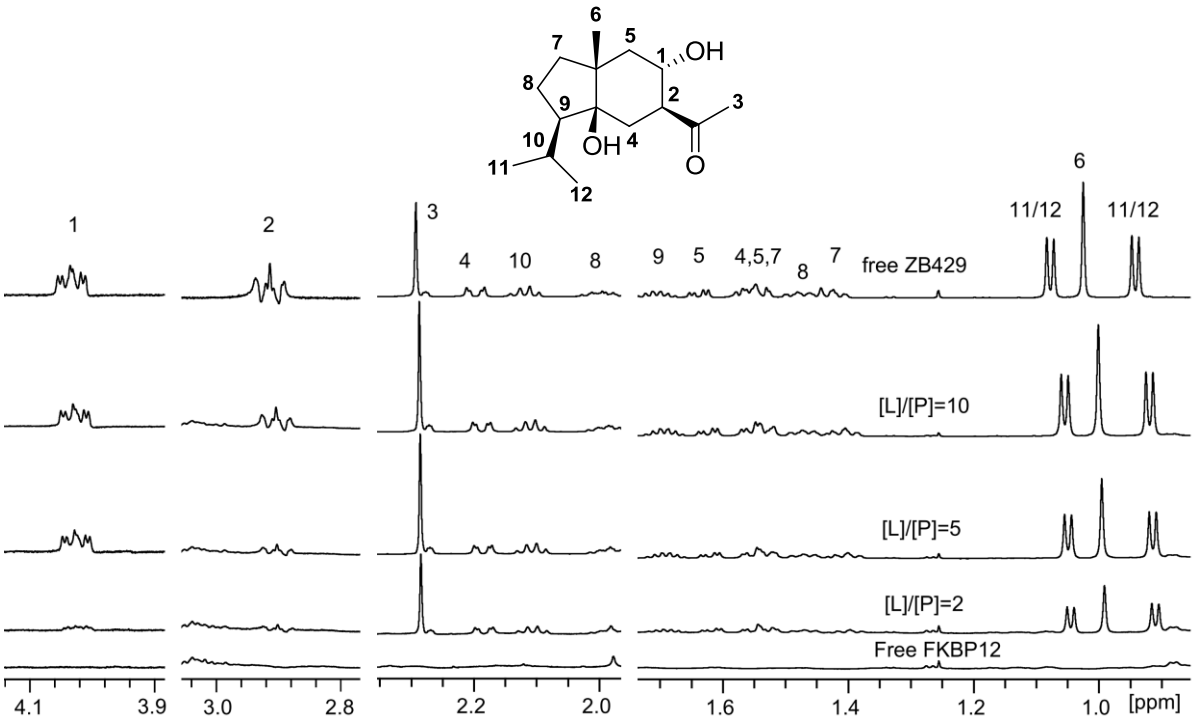

Figure A4: 1D-^{1}H NMR spectra of ZB429 titrated into wt FKBP12.

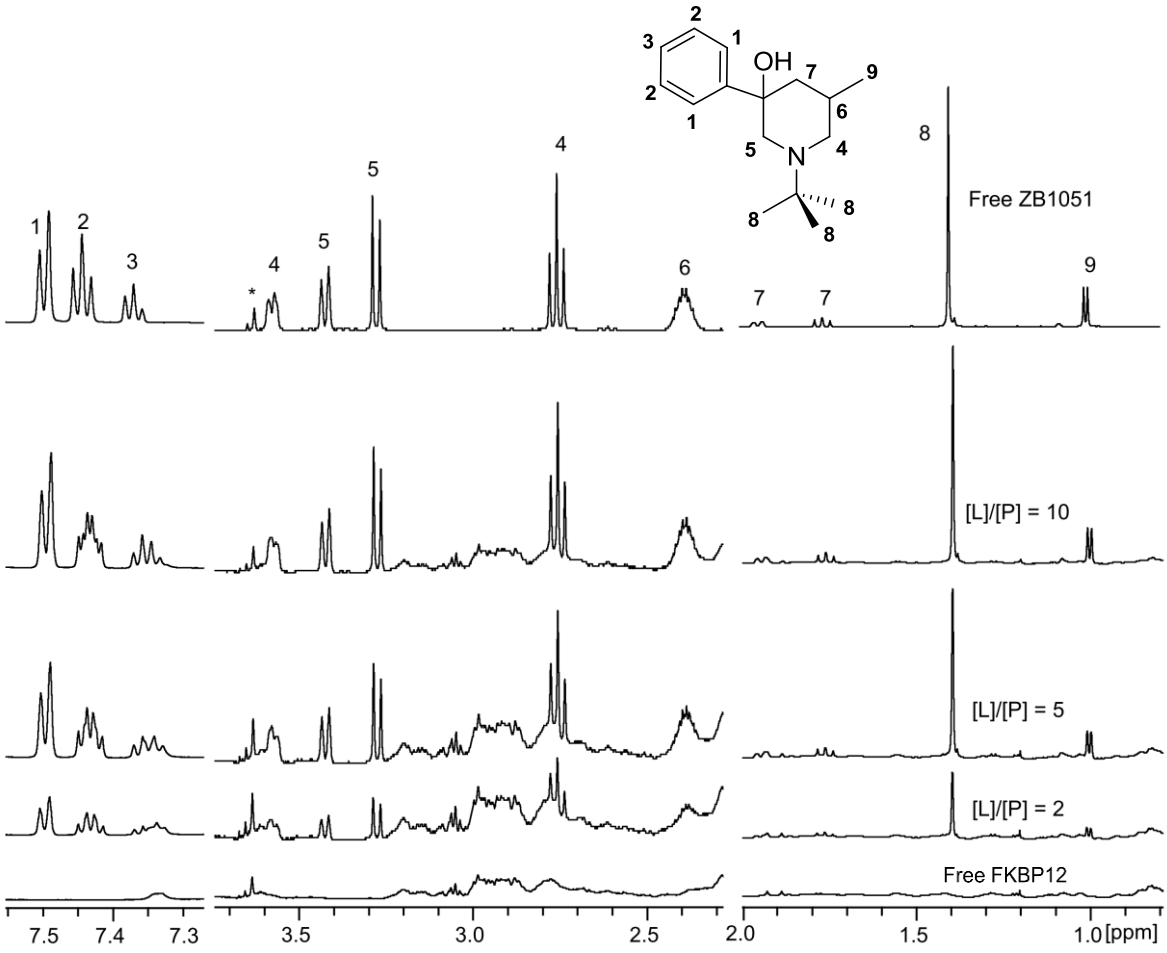

Figure A5: 1D-^{1}H NMR spectra of ZB1051 titrated into wt FKBP12.

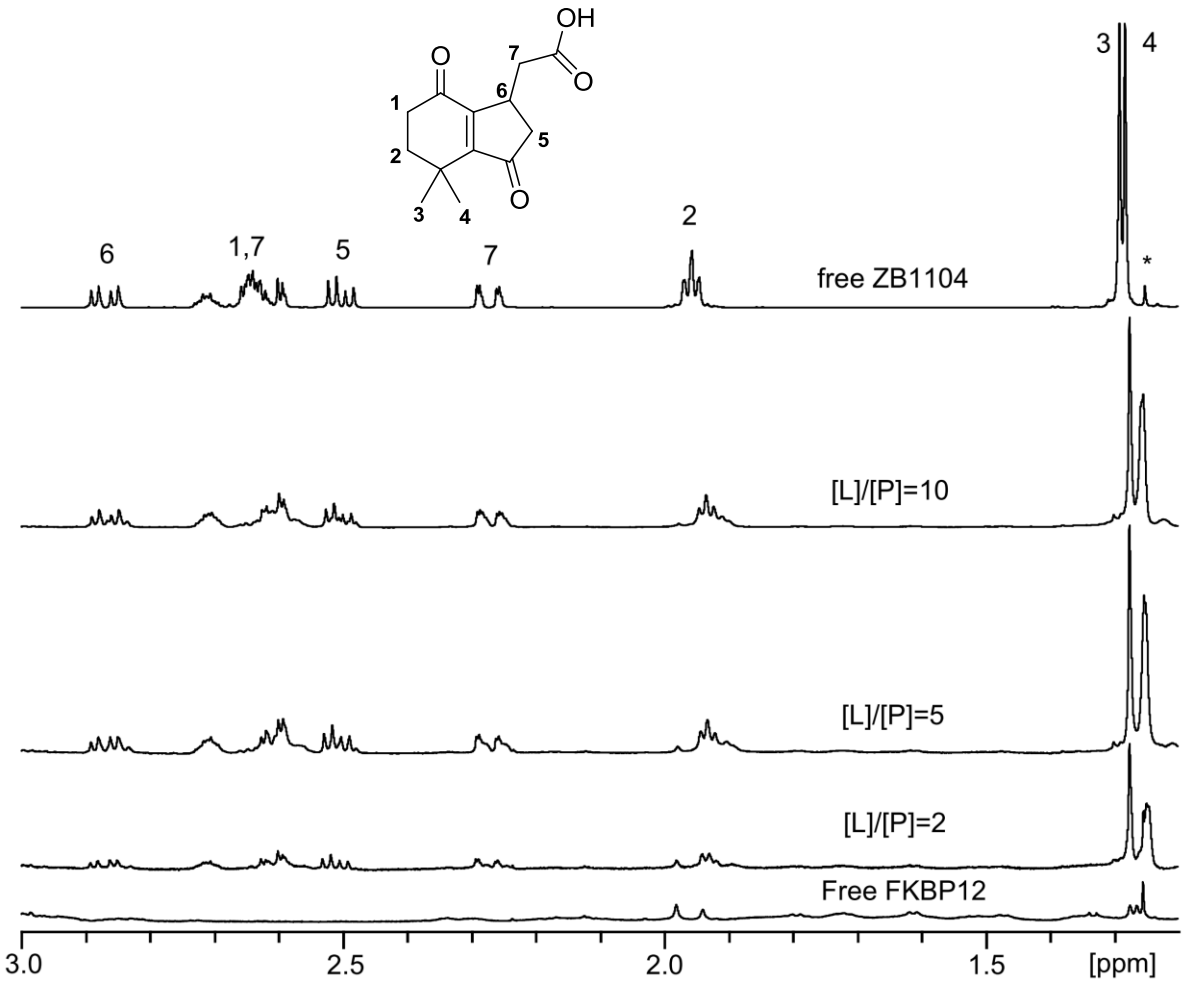

Figure A6: 1D-^{1}H NMR spectra of ZB1104 titrated into wt FKBP12. The proton assignments were based on prediction.

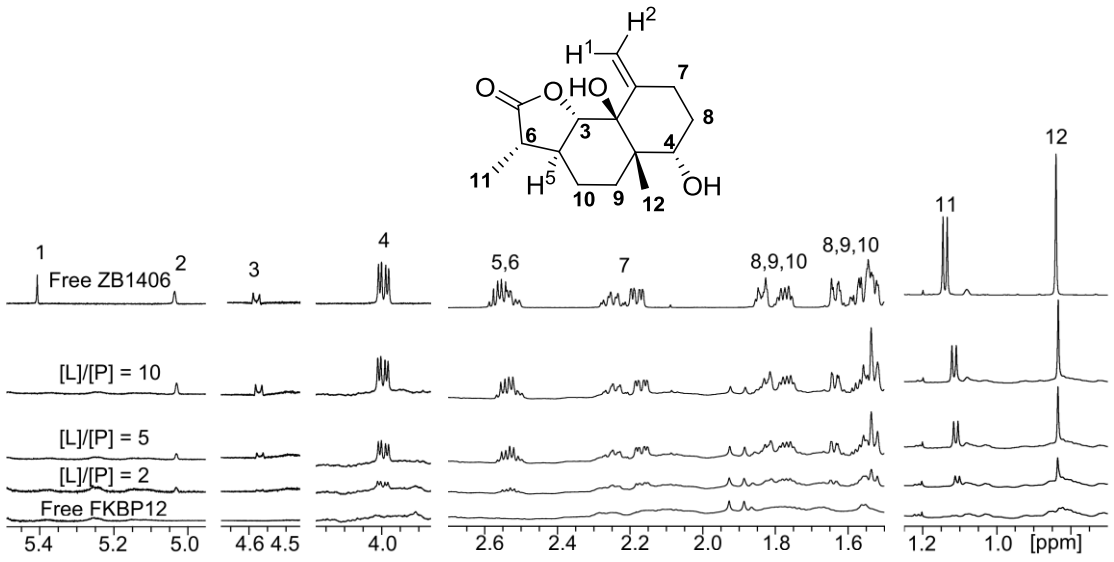

Figure A7: 1D-^{1}H NMR spectra of ZB1406 titrated into wt FKBP12. The proton assignments were based on prediction.

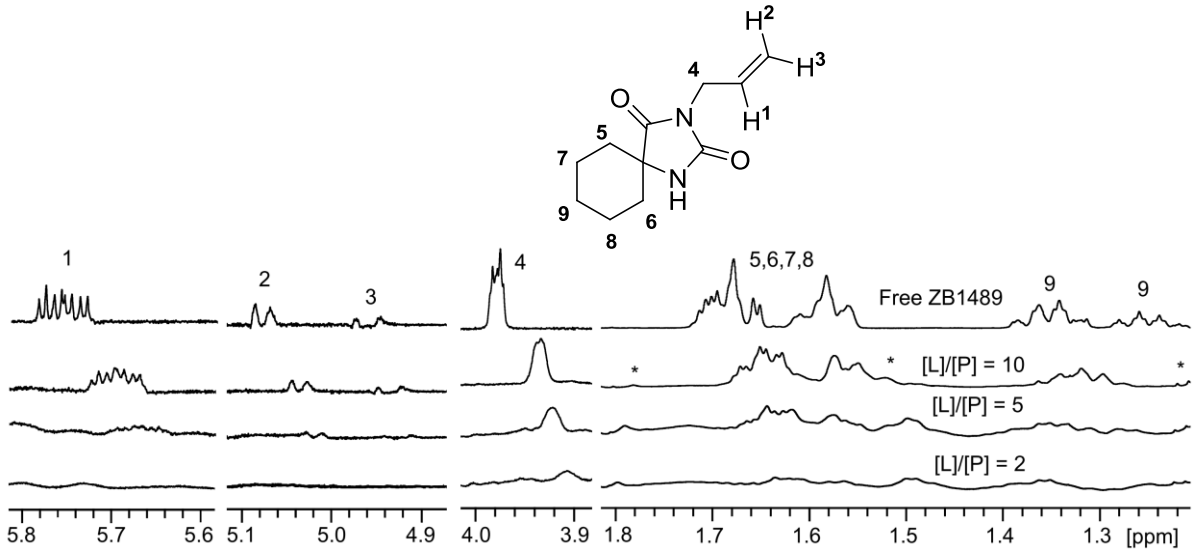

Figure A8: 1D-^{1}H NMR spectra of ZB1489 titrated into wt FKBP12. This compound showed complicated scalar coupling due to the aliphatic ring. The proton assignments were based on prediction.

Appendix B: XPLOR-NIH script for NOE- and PCS-based structure calculations

```
display $DATE $TIME
! dock n ligand copies in FKBP12 using NOE restraints (pdb:1FKR14)
! metal positions and tensor directions placed on PDB of protein
! mutant 2 = 34-35 Cys
! mutant 3 = 44-47 Cys
! mutant 5 = 61-65 Cys, 22 Ala

!-----------------DEFINE USER VARIABLES-----------------
 eval ($frbound2 = 0.015)            ! fraction ligand bound to mutant 2
 eval ($frbound3 = 0.009)            ! fraction ligand bound to mutant 3
 eval ($frbound5 = 0.019)            ! fraction ligand bound to mutant 5
 eval ($field = 600.13)              ! magnetic field in MHz
 eval ($MUT2="Y")                ! data for mutant 2 Y/N
 eval ($MUT3="Y")                ! data for mutant 3 Y/N
 eval ($MUT5="Y")                ! data for mutant 5 Y/N
 eval ($metal2="Yb")             ! Ln in mutant 2: Yb or Tm
 eval ($metal3="Yb")             ! Ln in mutant 3: Yb or Tm
 eval ($metal5="Yb")             ! Ln in mutant 5: Yb or Tm
eval ($chaxYbm2=8.7e-32)             ! chi axial in m^3 CLaNP-5 Yb mutant 2
 eval ($chrhoYbm2=3.4e-32)       ! chi rhombic in m^3 CLaNP-5 Yb mutant 2
 eval ($chaxYbm3=8.9e-32)             ! chi axial in m^3 CLaNP-5 Yb mutant 3
 eval ($chrhoYbm3=3.3e-32)       ! chi rhombic in m^3 CLaNP-5 Yb mutant 3
 eval ($chaxYbm5=7.8e-32)             ! chi axial in m^3 CLaNP-5 Yb mutant 5
 eval ($chrhoYbm5=2.7e-32)       ! chi rhombic in m^3 CLaNP-5 Yb mutant 5
 eval ($fres= 2)                 ! first residue to be used
 eval ($lres= 107)               ! last residue to be used
 eval ($PRF= (1e30*1e6)/(12*3.14159)) ! factor in PCS equation
 eval ($f01=6.0)                 ! force constant xpcs
 eval ($f02=6.0)                 ! force constant xpcs
 eval ($f03=6.0)                 ! force constant xpcs
 eval ($stdev1=0.04)             ! stdev for PCS
 eval ($stdev2=0.04)             ! stdev for PCS
 eval ($stdev3=0.04)             ! stdev for PCS
 eval ($a20=50)                  ! random distance max + 0 A
 eval ($a20a=360)                ! random rotation max +/-0.5*
 eval ($a05= 40000)              ! nr cycle
 eval ($a53= 300)                ! nr of steps
 eval ($a59= 10)                 ! .pdb writing threshold
 eval ($a55= 300.0)              ! TBATH (temperatuur)
 eval ($rot_5ring="Y")           ! rorate ligand oxazole ring
 eval ($intac="N")               ! interactions to BB (Y) or BB + side chain (N)
 eval ($rigid="N")               ! dynamics mode = rigid (Y) or rigid+langevin" (N)
 eval ($pcs_r="Y")               ! docking with ligand PCS restraints
 eval ($noe_r="N")               ! docking with NOE restraints
 eval ($vdw_r="Y")               ! switch on/off vdw
 eval ($flex_noe="Y")          ! also docking with NOE from flexible loop
 eval ($lig_noe="Y")           ! include ligand intramolecular (transferred) NOE
 eval ($attraction_noe = "Y") ! switch on/off attraction noe
 eval ($pcs_calc="Y")            ! back-calc PCS at end
 eval ($noe_calc="Y")            ! back-calc NOE distance at end
 eval ($opttens="N")             ! optimize tensor orienations iteratively
 eval ($kick=100)                ! # cycles per approach !100
 eval ($vdwmax=0.2)              ! starting rcon value, scaling of vdW energy
 eval ($lignr = 1)               ! number of ligand copies to be used
eval ($genscale = 0.1)           ! general scale for noe
 eval ($noescale = 0.5)          ! scale real noes
 eval ($g01 = 20)                ! nr of cycles to get to max vdw in rigid docking
(should be 2<$g01< $kick) !20
!-------------------------------------------------------------------------
!------------OTHER VARIABLES; DO NOT CHANGE-----------------------------
!-------------------------------------------------------------------------
 eval ($vleckaxYbm2=$chaxYbm2/3.77E-35)    !chi axial in van Vleck units
 eval ($vleckrhYbm2=$chrhoYbm2/3.77E-35)   !chi rhombic in van Vleck units
 eval ($vleckaxYbm3=$chaxYbm3/3.77E-35)    !chi axial in van Vleck units
 eval ($vleckrhYbm3=$chrhoYbm3/3.77E-35)   !chi rhombic in van Vleck units
 eval ($vleckaxYbm5=$chaxYbm5/3.77E-35)    !chi axial in van Vleck units
 eval ($vleckrhYbm5=$chrhoYbm5/3.77E-35)   !chi rhombic in van Vleck units
 eval ($a01=1)             !cycle counter
 eval ($a02=$kick)         !counts cycles after last kick
 eval ($a02a=1) !counts approach step in which min ener was reached
 eval ($a02b=1)            !counts cycle number in current approach
 eval ($c01=0)             !counts number of saved structures
```

```
 eval ($a58=30.0)          !fbeta, frictinal constant 30.0
 eval ($a56=1.5)           !velocity factor! 1.5
 eval ($a48=$cpu*1e4)      !($cpu*1e4)
 set seed=$a48 end         !generate random number
 eval ($a54 = 0.01)        !timestep in ps
 eval ($f1 = 0)            ! van der waals scaling factor
 eval ($kk2 = $kick/2)
!-------------------------------------------------------------------------
!--------------------- READ STRUCTURE ------------------------------------
!-------------------------------------------------------------------------
set echo=off end
set mess=off end
structure  @1FKR14.opt_lig.h.psf end
param  @1FKR14.opt_lig.h.par end
coor  @1FKR14.opt_lig.h.pdb
set echo=on end
set mess=on end

 parameter
        nbonds
                inhi=0.75
                ctofnb=5.0
                ctonnb=4.0
                repel=0.8
                rexp=2
                irex=2
                rcon=$f1
        end
 end

 parameters
    BOND   (segid "drg") (segid "drg") 400. TOKEN
    ANGLE  (segid "drg") (segid "drg") (segid "drg") 60. TOKEN
    IMPR  (segid "drg") (segid "drg") (segid "drg") (segid "drg") 50. TOKEN TOKEN
 end

 para
   nonb (name OO) 0.12  9.00     0.12  9.00
   nonb (name X)       0.0498   2.2272      0.0498   2.2272
   nonb (name Y)       0.0498   2.2272      0.0498   2.2272
   nonb (name Z)       0.0498   2.2272      0.0498   2.2272
  IMPR     OO  X  Z  Y 500.0 0 -54.7356
 end
!-----------------------------rotate ligand oxazole ring--------------------------
if ($rot_5ring="Y") then
 vector show (x) (resn DRG and name CAC)
 eval ($ix=$RESULT)
 vector show (y) (resn DRG and name CAC)
 eval ($iy=$RESULT)
 vector show (z) (resn DRG and name CAC)
 eval ($iz=$RESULT)
 vector show (x) (resn DRG and name CAD)
 eval ($jx=$RESULT)
 vector show (y) (resn DRG and name CAD)
 eval ($jy=$RESULT)
 vector show (z) (resn DRG and name CAD)
 eval ($jz=$RESULT)

 eval ($kx=$jx-$ix)
 eval ($ky=$jy-$iy)
 eval ($kz=$jz-$iz)

 coor rota SELE=(resn DRG and (name CAD or name OAE or name CAF or name HAG
   or name HAF or name CAK or name NAJ or name CAO or name CAL or name
   HAL or name HAM or name HAN or name HAR or name HAS or name HAQ))
      CENT=($jx $jy $jz) axis=($kx $ky $kz) -90.0 end
end if
!-------------------------------------------------------------------------
!--------------------- DEFINE RESTRAINTS ---------------------------------
!-------------------------------------------------------------------------
!-----------------DEFINE PCS-------------------------
    ! ---- mutant 2, pcs in Hz, as observed ---
 vector do (store1=-2.5) (segid="drg" and name CAL)
 vector do (store1=-3.5) (segid="drg" and name CAO)
 vector do (store1=-2.5) (segid="drg" and name HAF)
 vector do (store1=-2.5) (segid="drg" and name HAG)
 vector do (store1=-3.1) (segid="drg" and name HAB)
```

```
 vector do (store1=-6.9) (segid="drg" and name HAA)
 vector do (store1=-9.4) (segid="drg" and name HAH)
 vector do (store1=-5.7) (segid="drg" and name HAI)
     ! ---- mutant 3, pcs in Hz, as observed ---
 vector do (store2=2.2) (segid="drg" and name CAL)
 vector do (store2=2.2) (segid="drg" and name CAO)
 vector do (store2=1.3) (segid="drg" and name HAF)
 vector do (store2=1.3) (segid="drg" and name HAG)
 vector do (store2=0.9) (segid="drg" and name HAB)
 vector do (store2=0.4) (segid="drg" and name HAA)
 vector do (store2=0.9) (segid="drg" and name HAH)
 vector do (store2=2.3) (segid="drg" and name HAI)
    ! ---- mutant 5, pcs in Hz, as observed ---
 vector do (store3=3.5) (segid="drg" and name CAL)
 vector do (store3=2.9) (segid="drg" and name CAO)
 vector do (store3=3.7) (segid="drg" and name HAF)
 vector do (store3=3.7) (segid="drg" and name HAG)
 vector do (store3=1.6) (segid="drg" and name HAB)
 vector do (store3=1.5) (segid="drg" and name HAA)
 vector do (store3=1.5) (segid="drg" and name HAH)
 vector do (store3=1.3) (segid="drg" and name HAI)

if ($pcs_r="Y") then
!-----------------DEFINE CONDITIONS MUTANT 2-------------------------
 xpcs reset end
 xpcs nres 500 end

 if ($MUT2="Y") then
  xpcs
   class 2
   force $f01
   if ($metal2="Yb") then
     coeff $vleckaxYbm2 $vleckrhYbm2
   elseif ($metal2="Tm") then
     coeff $vleckaxTm $vleckrhTm
   end if
  end

  for $id in ID (store1)
  loop E
   vector show elem (store1) (ID $id)
   eval ($obs=$RESULT/($frbound2*$field))
   xpcs
     assign     (atom "tens" 1 OO)
           (atom "tens" 1 Z)
           (atom "tens" 1 X)
           (atom "tens" 1 Y)
           (ID $id) $obs $stdev1
   end
  end loop E
 end if
!-----------------DEFINE CONDITIONS MUTANT 3-------------------------

 if ($MUT3="Y") then
  xpcs
   class 3 !B
   force $f02
   if ($metal3="Yb") then
     coeff $vleckaxYbm3 $vleckrhYbm3
   elseif ($metal3="Tm") then
     coeff $vleckaxTm $vleckrhTm
   end if
  end

  for $id in ID (store2)
  loop A
   vector show elem (store2) (ID $id)
   eval ($obs=$RESULT/($frbound3*$field))
   xpcs
     assign     (atom "tens" 2 OO)
           (atom "tens" 2 Z)
           (atom "tens" 2 X)
           (atom "tens" 2 Y)
           (ID $id) $obs $stdev2
   end
  end loop A
 end if
```

```
!-----------------DEFINE CONDITIONS MUTANT 5-----------------------

 if ($MUT5="Y") then
  xpcs
   class 5 !B
   force $f03
   if ($metal5="Yb") then
     coeff $vleckaxYbm5 $vleckrhYbm5
   elseif ($metal5="Tm") then
     coeff $vleckaxTm $vleckrhTm
   end if
  end

 for $id in ID (store3)
 loop G
   vector show elem (store3) (ID $id)
   eval ($obs=$RESULT/($frbound5*$field))
    xpcs
     assign     (atom "tens" 3 OO)
           (atom "tens" 3 Z)
           (atom "tens" 3 X)
           (atom "tens" 3 Y)
           (ID $id) $obs $stdev3
   end
  end loop G
 end if
end if
!-----------------DEFINE REAL NOEs-----------------------
set echo=off end
set mess=off end
if ($noe_r="Y") then
@noe.xpl
end if
set echo=on end
set mess=on end
!-----------------DEFINE ATTRACTION NOE-----------------------
if ($attraction_noe = "Y") then
noe
  class      attraction
  scale      attraction 0.5     !0.3 0.1
  sqconstant  attraction $genscale
  sqexponent  attraction 2
end

eval ($l1 = 1)
while ($l1 LE $lignr) loop G1
  noe
    assign (segid = "drg" and resi $l1 and name CAD) (name C2 and resn CM2) 13.0
13.0 5.0
    assign (segid = "drg" and resi $l1 and name HAT) (segid = "drg" and resi $l1
and name NAJ) 2.0  1.0  0.1  end
eval ($l1 = $l1 +1)
end loop G1

end if

!-------------------------------------------------------------------------
!--------------------DYNAMICS RUN & OUTPUT--------------------------------
!-------------------------------------------------------------------------

 vector show (x) (name C2)
 eval ($xc2 = $RESULT)
 vector show (y) (name C2)
 eval ($yc2 = $RESULT)
 vector show (z) (name C2)
 eval ($zc2 = $RESULT)

 while ($a01 LE $a05) loop calc

  eval ($a02 = $a02 + 1)
  if ($a02 > $kick) then   ! -----------put ligand randomly away--------------
   coor init end           ! always use the same starting position of ligands
   coor @@1FKR14.opt_lig.h.pdb
  if ($rot_5ring="Y") then
   coor rota SELE=(resn DRG and (name CAD or name OAE or name CAF or name HAG
   or name HAF or name CAK or name NAJ or name CAO or name CAL or name
```

```
   HAL or name HAM or name HAN or name HAR or name HAS or name HAQ))
      CENT=($jx $jy $jz) axis=($kx $ky $kz) -90.0 end
  end if

   eval ($l1 = 1)
   while ($l1 LE $lignr) loop F2

     vector show ave (x)(segid="drg" and resi $l1)
     eval ($x1 = $RESULT)
     vector show ave (y)(segid="drg" and resi $l1)
     eval ($y1 = $RESULT)
     vector show ave (z)(segid="drg" and resi $l1)
     eval ($z1 = $RESULT)
     eval ($xr1 = $xc2 - $x1)
     eval ($yr1 = $yc2 - $y1)         ! move ligands to centre of protein
     eval ($zr1 = $zc2 - $z1)
     coor trans SELE=(segid="drg" and resi $l1) VECT=($xr1 $yr1 $zr1) end

     eval ($ang1=(0.5-RANDOM())*$a20a)
     eval ($vec1=(0.5-RANDOM())*3)
     eval ($vec2=(0.5-RANDOM())*3)
     eval ($vec3=(0.5-RANDOM())*3)  ! move and rotate ligands randomly away
     coor trans SELE=(segid="drg" and resi $l1) VECT=($vec1 $vec2 $vec3) DIST=50.0
end
     vector show ave (x)(segid="drg" and resi $l1)
     eval ($x1 = $RESULT)
     vector show ave (y)(segid="drg" and resi $l1)
     eval ($y1 = $RESULT)
     vector show ave (z)(segid="drg" and resi $l1)
     eval ($z1 = $RESULT)
     coor rota SELE=(segid="drg" and resi $l1)
              cent=($x1 $y1 $z1) axis ($vec3 $vec1 $vec2) $ang1 end
     eval ($l1 = $l1 + 1)
  end loop F2

  vector do (fbeta=$a58) (segid="drg")
  vector do (vx=$a56) (segid="drg")
  vector do (vy=$a56) (segid="drg")
  vector do (vz=$a56) (segid="drg")

  eval ($a02=1)
  eval ($miner=999999)
  !write coor FROM=MAIN SELE=(all) OUTPUT= temp2.pdb end
 end if

! ---------------------dynamics run-----------------------------
  if ($intac="Y") then
   constraints interactions
    (segid="drg")
    (segid= "tens" or (segid "fkbp" and (name CA or name N or name C or name CB or
name HN or name O)))  end
  else
   constraints interactions (segid="drg")(segid "tens" or segid "fkbp") end
  end if

  constraints FIX=(not segid "drg") end

  if ($pcs_r="Y") then
    if ($vdw_r="Y") then
           if ($attraction_noe="Y") then
               flag exclude * include noe vdw xpcs end
           else
               flag exclude * include vdw xpcs end
           end if
    else
         flag exclude * include noe xpcs end
    end if
  else
    if ($vdw_r="Y") then
        flag exclude * include noe vdw end
    else
        flag exclude * include noe end
    end if
  end if

  para nbond rcon = $f1 end end
  if ($f1 < $vdwmax) then
```

```
    eval ($f1 = $f1 + $vdwmax/($g01-1))  !valid for next round
  end if
  display QQQQQ $f1

  display cycle $a01
  dynamics rigid
     dt=$a54
     group=(segid="drg" and resi 1)
     dynmode=TCOU !TCOU FREE LANG
     tbath=$a55
     nprint=50
     nstep=$a53
     NTRFRQ=0 !new for XPLOR vs 3.8
  end

if ($rigid="N") then
 if ($a02b > $g01) then
 !------------local langevin dynamics--------------
 constraints
   interactions ((segid "fkbp" and not (name Ca or name C or name O or name N or
name
   HN or name C2)  and segid "drg" around 8.0) ) (segid "fkbp")
   interactions (segid "drg") (all)
 end

  constraints  !fix protein Bb + sc outside 8A from ligand + tens; ligand free
   FIX (not ((segid "fkbp" and not (name Ca or name C or name O or name N or name
   HN or name C2) and segid "drg" around 8.0) or (segid="drg") ))
  end

 display cycle $a01

 if ($pcs_r="Y") then
     if ($vdw_r="Y") then
          if ($attraction_noe="Y") then
           flag exclude * include noe vdw  bond angle dihedral impro xpcs end
          else
           flag exclude * include vdw  bond angle dihedral impro xpcs end

          end if
    else
           flag exclude * include noe bond angle dihedral impro xpcs end
     end if
 else
     if ($vdw_r="Y") then
           flag exclude * include noe vdw  bond angle dihedral impro end
    else
      flag exclude * include noe bond angle dihedral impro end
    end if
 end if
 energy end

 vector do ( fbeta = 6.657235 ) ((segid "fkbp" and not (name Ca or name C or name
O or name N or name HN or name C2)  and segid "drg" around 8.0) or segid "drg" )

 dynamics langevin
    timestep = 0.001
    nstep = 2000
    ilbfrq = 2000
    nprint = 200
    tbath = 300.
    iasvel = maxwell
    rbuf = 0.0
    origin = ( 0. 0. 0. )
 end

 minimize powell
  drop=1
  nprint=10
  nstep=50
 end

 end if
end if      !------------local langevin dynamics finished--------------

    !------------CALCULATE ENERGIES AFTER EACH CYCLE--------------
   constraints
```

```
 interactions (segid "drg") (segid "fkbp")
end
if ($pcs_r="Y") then!!MAKE SURE THIS MATCHES THE FLAG DURING THE RUN
   if ($vdw_r="Y") then
    if ($attraction_noe="Y") then
      flag exclude * include noe vdw xpcs end
    else
      flag exclude * include vdw xpcs end
    end if
   else
      flag exclude * include noe xpcs end
   end if
else
   if ($vdw_r="Y") then
      flag exclude * include noe vdw end  !xpcs
   else
      flag exclude * include noe end  !xpcs
   end if
end if
energy end

set disp=ener.dat end         !write energy values
if ($a01=1) then
   display file: ener.dat     $DATE $TIME
  if ($pcs_r="Y") then
        if ($vdw_r="Y") then
             if ($attraction_noe="Y") then
                  display cycle   appr.step  Etot  Evdw  Enoe  Epcs
             else
                  display cycle   appr.step  Etot  Evdw  Epcs
             end if
        else
             display cycle   appr.step  Etot  Enoe  Epcs
        end if
  else
        if ($vdw_r="Y") then
             display cycle   appr.step  Etot  Evdw  Enoe
        else
             display cycle   appr.step  Etot  Enoe
        end if
  end if
end if

if ($pcs_r="Y") then
   if ($vdw_r="Y") then
        if($attraction_noe="Y") then
             display $a01    $a02a $a02  $ENER $VDW  $NOE  $XPCS
        else
             display $a01    $a02a $a02  $ENER $VDW  $XPCS
        end if
   else
        display $a01    $a02a $a02  $ENER $NOE  $XPCS
   end if
else
   if ($vdw_r="Y") then
     display $a01  $a02a $a02  $ENER $VDW  $NOE
   else
     display $a01  $a02a $a02  $ENER $NOE
   end if
end if
set disp=OUTPUT end

if ($miner>$ENER) then
if ($a02b > $g01) then
coor copy end
 eval ($miner = $ENER)
 eval ($mincyc = $a01)
 eval ($minstep = $a02)
 if ($vdw_r="Y") then
        eval ($minvdw = $VDW)
 end if
 if ($noe_r="Y") then
     eval ($minnoe = $NOE)
 end if
 if ($pcs_r="Y") then
      eval ($minpcs = $XPCS)
     if ($attraction_noe="Y") then
```

```
            eval ($minnoe = $NOE)
       end if
   end if
  end if
  end if
   !------------END OF APPROACH REACHED --------------
  if ($a02 = $kick) then
    if ($miner LE $a59) then
     !---write structure---
        eval ($c01 = $c01 + 1)
        eval ($a52="cycle_"+encode($c01)+".pdb")
     coor swap end
        write coord output=$a52 end
     !---write information about this cycle----
     set disp=coor.dat end!write details of step
     if ($c01=1) then
           display file: coor.dat   $DATE $TIME
         if ($pcs_r="Y") then
          if ($vdw_r="Y") then
           if($attraction_noe="Y") then
               display file    cycle approach   app. step  Etot  Evwd  Enoe  Epcs
           else
               display file    cycle approach   app. step  Etot  Evwd  Epcs
           end if
          else
               display file    cycle approach   app. step  Etot  Enoe  Epcs
          end if
         else
           if ($vdw_r="Y") then
             display file cycle approach   app. step  Etot  Evwd  Enoe
           else
             display file cycle approach   app. step  Etot  Enoe
           end if
         end if
       end if
      if ($pcs_r="Y") then
       if ($vdw_r="Y") then
        if($attraction_noe="Y") then
          display $c01 $mincyc $a02a $minstep $miner $minvdw $minnoe $minpcs
        else
          display $c01 $mincyc  $a02a $minstep $miner $minvdw $minpcs
        end if
          else
            display $c01 $mincyc $a02a $minstep $miner $minnoe $minpcs
          end if
      else
           if ($vdw_r="Y") then
             display $c01  $mincyc    $a02a $minstep   $miner     $minvdw    $minnoe
                else
             display $c01  $mincyc    $a02a $minstep   $miner     $minnoe
           end if
      end if
        set disp=OUTPUT end
     !---write file with PCS violations for this cycle---
        if ($pcs_calc="Y") then
         @pcs_calc.xpl
        end if
     !---write file with distances of atoms with NOEs---
           if ($noe_calc="Y") then
       @noe_calc.xpl
     end if
     set echo=on end
     set mess=on end
    end if
    eval ($a02a = $a02a + 1 )    !approach counter
    eval ($f1 = 0)  !set vdw back to zero
    eval ($a02b = 0) !cycle per approach
  end if

  eval ($a02b = $a02b + 1) !cycle per approach
  eval ($a01 = $a01 + 1)   !cycle counter
 end loop calc
display $DATE $TIME
!---------------------------------------------------------------
!----------------------FINISH-----------------------------------
!---------------------------------------------------------------
Stop
```

Appendix C: XPLOR-NIH script for PRE-based ensemble docking

```
! Energy minimization of TOAC pseudoatom position using PRE restraints
! Using Plastocyanin for docking TOAC on tetra-Lys peptide
! M. Ubbink 15.2.2013
! file: red_run.inp

 set mess=off end
 set echo=off end
!------------------------------------------------------------------------
!---------------VARIABLES--------------------------------------------------
!------------------------------------------------------------------------
 eval ($idi="plas")        ! SEGID of plastocyanin
 eval ($d1= 0.1)           ! lower bound CL3, real PRE
 eval ($d2= 0.1)           ! upper bound CL3, real PRE
! eval ($d3= 10.0)         ! target distance CL1
 eval ($d4= 7)             ! lower bound CL1, disappeared
 eval ($d5= 0)             ! upper bound CL1, disappeared
 eval ($d6= 15)       ! target distance CL2, unaffected
 eval ($d7= 0)             ! lower bound CL2, unaffected
 eval ($d8= 100.0)         ! upper bound CL2, unaffected
 eval ($d9= 0.5)           ! general scaling
 eval ($d10= 0.1)          ! scaling CL1 (disappeared) 1
 eval ($d11= 0.1)          ! scaling CL2 (not affected) 0.1
 eval ($d12= 0.1)          ! scaling CL3 (real PREs) 1
 eval ($d13= 100)          ! nr steps Powell energy minimization
 eval ($d15= 0.01)         ! scaling NOE to pull all the Toac to the protein
 eval ($tau_c= 5.54E-9)! tau-c in s
 eval ($fbeta=0.06)   !TC friction coefficient in psec-1
 eval ($run_nr= 1)    !cycle counter, $a01 in fkbp script
 eval ($kick=20)           ! # cycles per approach
 eval ($total_cycles= 11000)    ! nr runs EM (nr cycles),  $a05 in fkbp script
 eval ($a02=1)             !counts cycles after last kick
 eval ($min_app=1)    !counts approach step in which min ener
                           ! was reached, $a02a in fkbp script
 eval ($min_cycle=1)  !counts cycle number in current approach, $a01 in fkbp script
 eval ($a20=40)            ! random distance max + 0 A
 eval ($a20a=30)           ! random rotation max +/-0.5*
 eval ($c01=0)             !counts number of saved structures, $c01 in fkbp script
 eval ($threshold=  4.00944*1.25)     ! .pdb writing threshold, $a59 in fkbp script

 eval ($frb=1)        ! fraction in which the peptide is bound to PoPc
 eval ($frl=1)             ! fraction of spin labelled TOAC peptide
 eval ($ens=6)             ! nr of TOAC molecules in ensemble
 eval ($t3=1)              ! first residue number of TOAC
 eval ($t4=$t3+$ens-1)
 eval ($a48=$cpu*1e4)   !($cpu*1e4)
 set seed=$a48 end         !generate random number
 eval ($Crd="Y")      ! Control loop Crd (Y/N)to kick TOAC after each approach
 eval ($ener_check="Y") ! switch on energy barrier (Y/N)
 eval ($checkener=2.1*$threshold)  ! energy barrier to prevent coordinate
explosion
 eval ($attraction_noe="Y") ! switch on attraction noe (Y/N)
 eval ($fix_CL2="Y")    ! fix experimental distance for CL2 (unaffected)
!------------------------------------------------------------------------
!---------------PARAMETERS & STRUCTURE ----------------------------------
!------------------------------------------------------------------------

 parameter
     @PoPc.cu.h.par
 end

 parameter
  NONBonded  TC      0.1200   3.3      0.1200   3.3
 end

 structure
     @PoPc.cu.h.psf
 end

 coordinates
     @PoPc.cu.h.pdb


!---------------define TOAC ----------------------------
```

```
 topology
    residue TOAC
     atom TC type=C charge=0.000 mass 12.00 end
    end
 end


 segment
    molecule
     name=TOAC
     number=$ens
   end
 end

 para
  nbon
    repe = 0.65
    rcon = 60

  end
 end

 vector do (X=10.0) (resn TOAC and name TC)!
 vector do (Y=10.0) (resn TOAC and name TC)!define initial TC coordinates
 vector do (Z=10.0) (resn TOAC and name TC)!

!------------------------------------------------------------------------
!----------------RESTRAINTS ---------------------------------------------
!------------------------------------------------------------------------

 !DEFINE NMR DERIVED DISTANCES FOR TC-NH
 @restraints_PoPc_KKKKX_Kd93uM.xpl

 if ($attraction_noe="Y") then !
 NOE
   class CL4
   averaging CL4 cent
   potential CL4 square
   sqconstant CL4 0.01
   sqoffset CL4 0.0
   scale CL4 $d15
   sqexponent CL4 2
   ceil=10.0 !20
  end

 eval ($t1 = $t3)
 while ($t1 LE $t4) loop D ! set random coordinates for TOAC atoms
  noe
    assign (name TC and resi $t1) (name C2 and resn CM2) 20.0 10.0 10.0
  end
 eval ($t1 = $t1 + 1)
 end loop D
 end if ! if ($attraction_noe="Y") then

!------------------------------------------------------------------------
!----------------RUN -----------------------------------------------------
!------------------------------------------------------------------------

 set echo=off end
 set mess=off end

 constraints fix (not name TC) end
 constraints interactions (name TC) (not name TC and not name C2) end
 flag exclude * include noe vdw end

 while ($run_nr LE $total_cycles) loop RUNS
   eval ($t1 = $t3)
   !coor orient sele=(segid "plas") end
   if ($Crd="Y") then
    if ($a02 = 1) then
     while ($t1 LE $t4) loop Crd     ! set random coordinates for TOAC atoms
     !------------put TC at origin-----------------
     vector do (fbeta=0.025) (name TC and resi $t1)
     vector do (vx=0.1)(name TC and resi $t1)
     vector do (vy=0.1)(name TC and resi $t1)
     vector do (vz=0.1)(name TC and resi $t1)
     vector show ave (x) (name TC and resi $t1)
```

```
     eval ($x2= $RESULT)
     vector show ave (y) (name TC and resi $t1)
     eval ($y2= $RESULT)
     vector show ave (z) (name TC and resi $t1)
     eval ($z2= $RESULT)
     eval ($dist=sqrt($x2**2+$y2**2+$z2**2))
     coor trans SELE=(name TC and resi $t1) VECT=(-$x2 -$y2 -$z2) DIST=$dist end

     !------------- move TOAC randomly away----------------------

     eval ($vec1=0.5-RANDOM())
     eval ($vec2=0.5-RANDOM())
     eval ($vec3=0.5-RANDOM())
     coor trans SELE=(name TC and resi $t1) VECT=($vec1 $vec2 $vec3) DIST=$a20 end
     eval ($t1 = $t1 +1)
    end loop Crd

    minimize powell
      drop=10
      nprint=50
       nstep=$d13
       tolgradient=0.0001
    end

   end if ! if ($a02 = 1) then
   end if ! if($Crd="Y") then

   dynamics internal
     nstep = 500
     depred = 1
     etol = 1
     maxenergy = 100
     timestep = 0.002
     friction = $fbeta
     nsavc = 20
     fix = (segid $idi)
     group = (name TC and resi 1)
     group = (name TC and resi 2)
     group = (name TC and resi 3)
     group = (name TC and resi 4)
     group = (name TC and resi 5)
     group = (name TC and resi 6)
     tbath = 300
     nprint = 50
      NTRFRQ=0 !new for XPLOR vs 3.8
   end

   energy end

!-------------write all energy info-------------------------

  set disp=ener.dat end         !write energy values
  if ($run_nr=1) then
   display ener.dat   $DATE $TIME
   display cycle      appr step Etot  Enoe Evdw
  end if
  display $run_nr     $min_app $a02   $ENER $NOE   $VDW
  set disp=OUTPUT end

!-------------check energy-----------------------------------
  if ($ener_check="Y") then
    if ( $a02 = 1 ) then
     if ($ENER>$checkener) then
      display ----------------------ENERGY TOO HIGH, GO TO NEXT CYCLE -----------
---------
      eval ($a02=0)
      eval ($min_app = $min_app + 1 ) !approach counter
     end if ! if ($ENER>$checkener) then
    end if !if ( $a02 = 1 ) then

    if ( $a02 = 2 ) then
     if ($ENER>$miner) then
      display ----------------------ENERGY TOO HIGH, GO TO NEXT CYCLE -----------
---------
      eval ($a02=0)
      eval ($min_app = $min_app + 1 ) !approach counter
     end if ! if ($ENER>$miner) then
```

```
    end if ! if ( $a02 = 2 ) then
  end if !if ($ener_check="Y") then
!------------save energy info for currently lowest energy structure--------------

  if ($a02 = 1) then
    eval ($miner=999999)
  end if

  if ($miner>$ENER) then
    coor copy end
    eval ($miner = $ENER)
    eval ($mincyc = $run_nr)
    eval ($minstep = $a02)
    eval ($minvdw = $VDW)
    eval ($minnoe = $NOE)
  end if


!------------END OF APPROACH REACHED --------------

!-------------------------------------------------------------------------
!----------------ANALYSIS-------------------------------------------------
!-------------------------------------------------------------------------

!------------write the coordinates for lowest energy structure--------------

  if ($a02 = $kick) then

    if ($miner LE $threshold) then
     eval ($c01 = $c01 + 1)
     eval ($a52="cycle_"+encode($c01)+".pdb")
     coor swap end
     write coord sele=(name TC) OUTPUT=$a52 end

     !------------save the PRE info for lowest energy structure--------------
     @pre_calc.xpl

     !------------write the energy info for lowest energy structure--------------

     set disp=coor.dat end !write details of step
     if ($c01=1) then
            display file: coor.dat   $DATE $TIME
          display file     cycle approach   app.step   Etot  Enoe  Evwd violation
        end if
        display $c01  $mincyc    $min_app   $minstep   $miner     $minnoe    $minvdw
     $viol
     set disp=OUTPUT end


    end if ! if ($miner LE $threshold) then

    eval ($min_app = $min_app + 1 )   !approach counter
    eval ($min_cycle = 0) !cycle per approach
    eval ($a02 = 0)
  end if !  if ($a02 = $kick) then
  display ------------------------END OF RUN $run_nr-----------------------

!----------------go to the next run-----------------------------
  eval ($run_nr = $run_nr +1) ! run counter
  eval ($a02 = $a02 + 1)
  eval ($min_cycle = $min_cycle + 1) !cycle per approach



end loop RUNS
display $DATE $TIME
set echo=off end
set mess=on end
!---------------------------------------------------------------
!----------------------FINISH-----------------------------------
!---------------------------------------------------------------
Stop
```

Appendix D: Backbone and side chain assignments of Zn-substituted PoPc

Residue	Cα	Cβ	H	N
I1	59.73	40.89	-	-
I1	59.76	38.60	-	-
D2	52.86	42.05	7.793	126.2
D2	53.10	41.58	8.310	111.2
V3	60.62	34.86	8.810	124.1
V3	60.64	35.08	8.773	124.0
L4	54.68	42.58	8.963	126.0
L5	53.18	41.42	8.626	119.4
G6	44.37	0.00	7.816	116.2
A7	50.76	20.87	7.995	125.1
D8	57.51	39.98	8.935	119.9
D9	53.42	39.59	7.627	115.3
G10	44.71	0.00	8.189	108.7
S11	60.35	63.09	8.046	115.8
L12	53.16	39.11	8.682	125.9
A13	50.06	22.39	7.351	124.6
F14	58.35	40.45	9.077	121.2
V15	59.31	34.20	8.823	121.5
P16	63.63	35.03	-	-
S17	59.21	64.62	8.160	107.7
E18	55.26	32.27	7.500	119.9
F19	56.00	39.07	7.632	119.0
S20	56.74	66.30	8.592	115.3
S20	56.67	66.27	8.626	115.3
I21	59.62	42.71	8.716	116.5
I21	59.70	43.03	8.716	116.5
S22	56.78	63.29	8.228	116.8
S22	56.99	63.17	8.406	117.4
P23	64.51	31.46	-	-
P23	64.39	31.48	-	-
G24	44.97	0.00	9.216	113.3
G24	44.95	0.00	9.095	113.9
E25	56.99	31.06	7.655	121.7
E25	56.60	30.65	7.801	123.1
K26	56.16	33.67	8.040	122.3
K26	56.19	33.98	7.981	120.4
I27	60.34	39.64	8.867	125.6
I27	60.65	39.11	8.858	125.0
V28	61.64	31.39	8.468	126.3
V28	61.63	31.14	8.437	126.8
F29	57.18	39.29	9.042	126.5
K30	54.83	35.19	9.173	122.8
K30	54.88	35.11	9.138	122.6
N31	54.46	38.45	9.229	126.8
N32	56.50	42.50	9.016	129.2
A33	52.28	22.05	8.647	119.8
G34	47.51	0.00	7.750	116.5
F35	52.80	36.98	5.720	114.5
P36	62.47	36.82	-	-

Residue	Cα	Cβ	H	N
H37	53.92	36.59	7.852	110.2
N38	52.66	39.93	9.196	120.3
I39	61.92	40.27	6.082	112.6
V40	61.15	34.73	8.605	126.4
F41	57.89	39.58	8.529	123.0
D42	53.75	42.31	8.385	121.3
E43	59.06	29.61	8.748	125.7
D44	55.37	41.47	8.314	117.6
S45	56.75	64.33	8.125	116.5
I46	58.83	39.45	7.060	114.9
P47	59.77	31.97	-	-
S48	59.76	63.32	8.235	115.5
G49	45.11	0.00	8.729	112.8
V50	62.26	32.42	7.372	121.1
D51	52.25	41.03	8.251	126.7
A52	54.95	18.47	9.348	130.0
S53	61.47	63.13	8.617	114.2
K54	56.76	32.96	7.314	118.8
I55	61.15	38.30	6.984	109.6
S56	58.17	67.13	6.649	110.2
M57	56.75	33.54	8.213	122.5
M57	56.53	33.12	8.247	122.6
S58	58.50	63.50	8.548	115.8
E59	59.72	29.48	8.597	122.9
E60	56.38	29.80	8.047	113.4
D61	54.05	42.14	7.601	122.5
L62	53.52	46.62	7.723	118.3
L63	54.07	40.34	8.882	119.7
N64	54.79	42.10	8.467	124.7
A65	50.11	21.55	8.349	121.2
K66	58.67	32.49	8.538	125.0
G67	45.13	0.00	8.806	113.6
E68	58.18	32.33	7.183	119.9
T69	59.38	73.68	8.373	110.6
T69	59.48	73.63	8.382	110.8
F70	57.87	43.78	8.661	120.7
F70	57.88	43.72	8.632	120.8
E71	54.00	32.74	7.789	126.2
E71	54.01	32.75	7.826	126.2
V72	60.21	35.52	8.568	120.1
V72	60.24	35.50	8.509	119.9
A73	50.39	20.61	8.235	128.9
L74	53.46	43.46	7.595	120.3
L74	53.35	43.48	7.556	120.3
S75	59.70	64.97	8.685	115.2
N76	53.95	37.54	7.901	120.0
K77	58.09	33.51	8.726	124.5
G78	43.70	0.00	8.683	108.2
G78	43.69	0.00	8.745	108.2

Residue	Cα	Cβ	H	N
E79	56.04	31.99	8.224	119.2
Y80	56.54	39.77	9.513	122.6
S81	58.36	64.76	8.268	118.7
F82	55.10	41.15	7.978	119.8
Y83	56.12	41.21	9.183	115.4
C84	56.22	29.46	7.804	121.2
S85	64.80	61.34	10.230	126.5
P86	65.14	30.36	-	-
H87	54.11	34.88	7.189	112.5
Q88	60.88	28.55	7.982	115.1
G89	46.30	0.00	8.964	108.5
A90	51.44	19.11	7.296	121.3
G91	45.35	0.00	7.829	110.4
M92	56.89	30.37	7.758	121.5
V93	58.74	35.98	8.016	121.5
G94	45.22	0.00	8.258	112.2
K95	55.78	36.67	8.394	119.7
V96	56.90	33.50	9.022	122.2
T97	61.77	70.28	8.244	124.4
V98	60.94	31.57	9.236	128.7
N99	55.42	42.04	8.648	132.1

Residue	H(sc1)	H(sc2)	N(sc)
N38	7.043	6.242	107.2
N64	7.631	6.865	113.5
Q88	6.731	6.189	110.8
N99	7.418	6.683	112.2

Appendix E: Backbone assignments of Zn-substituted DPc

Residue	Cα	Cβ	H	N
A1	51.87	20.70	-	-
K2	55.70	35.28	8.579	123.2
V3	61.23	35.98	8.568	125.4
E4	55.79	30.83	9.265	126.7
V5	61.75	31.83	9.060	125.4
G6	43.85	-	8.015	112.6
D7	52.58	43.17	6.938	109.9
E8	59.14	29.44	8.915	116.9
V9	62.03	31.73	7.415	113.5
G10	45.51	-	7.841	108.9
N11	54.45	39.01	9.108	119.1
F12	51.90	34.75	8.662	125.6
K13	54.33	38.05	6.459	117.4
F14	57.37	41.32	8.332	119.6
Y15	54.80	41.22	9.328	118.4
P16	63.69	35.15	-	-
D17	53.10	40.29	7.859	111.0
S18	56.93	64.35	7.184	114.5
I19	59.99	42.59	8.649	125.1
T20	61.43	70.56	8.469	124.6
V21	58.40	35.73	9.124	119.6
S22	58.03	64.49	8.138	116.4
A23	54.41	18.25	8.672	125.7
G24	45.16	-	8.738	113.1
E25	55.98	30.80	8.118	123.4
A26	51.72	19.64	8.497	127.6
V27	61.44	32.63	8.924	122.4
E28	54.95	30.76	8.439	126.7
F29	57.07	40.30	9.443	127.5
T30	60.83	71.85	8.771	118.2
L31	56.23	43.71	9.455	130.6
V32	62.51	33.33	8.743	131.5
G33	43.85	-	8.279	112.5
E34	57.38	31.12	8.006	114.3
T35	63.05	69.04	7.865	117.7
G36	46.38	-	8.556	112.7
H37	55.24	36.53	6.585	116.3
N38	51.16	40.36	9.503	123.8
I39	61.51	40.08	5.990	113.0
V40	61.34	35.83	8.706	125.3
F41	58.75	40.72	8.264	124.0
D42	54.35	42.60	8.822	122.9
I43	58.10	37.75	8.469	125.2
P44	62.77	32.41	-	-
A45	53.68	18.05	8.606	126.7
G46	45.31	-	8.774	110.3
A47	50.79	17.41	7.482	123.5
P48	62.40	32.67	-	-
G49	47.82	-	8.887	110.9
T50	64.51	67.95	8.040	112.2
V51	65.49	32.34	6.966	124.0

Residue	Cα	Cβ	H	N
A52	56.14	18.11	8.234	122.0
S53	61.86	63.17	8.185	111.9
E54	59.61	30.32	7.695	124.9
L55	57.47	41.49	8.406	119.8
K56	59.66	32.20	8.160	122.0
A57	54.16	17.97	7.772	120.9
A58	51.60	19.40	7.223	121.2
S59	58.12	66.67	7.159	108.3
M60	56.24	34.29	8.489	123.1
D61	55.08	42.98	8.567	120.2
E62	59.42	29.74	8.460	123.5
N63	53.28	39.44	8.655	115.4
D64	53.26	43.33	7.593	121.7
L65	54.30	47.17	8.001	117.4
L66	55.37	44.15	8.308	121.3
S67	56.81	67.73	8.003	115.1
E68	59.52	29.62	8.688	118.2
D69	55.86	41.60	7.757	115.1
E70	53.79	30.18	7.251	120.2
P71	65.22	33.45	-	-
S72	56.90	66.60	7.904	114.3
F73	57.84	44.44	9.029	125.5
K74	54.81	34.76	7.569	127.5
A75	50.40	23.60	8.753	125.6
K76	55.85	34.50	8.355	122.0
V77	62.23	33.58	8.862	128.0
S78	59.58	64.54	9.329	123.0
T79	61.79	68.79	8.145	123.5
P80	63.84	32.46	-	-
G81	44.78	-	8.537	110.4
T82	62.25	71.00	7.628	115.2
Y83	56.60	42.36	9.569	128.0
T84	61.67	71.15	9.610	119.2
F85	54.32	40.36	8.089	121.4
Y86	56.42	41.47	8.616	115.1
C87	57.18	29.85	7.105	120.1
T88	69.99	67.19	9.995	126.6
P89	65.03	31.09	-	-
H90	57.96	32.67	7.459	114.7
K91	61.58	29.21	7.903	128.0
S92	60.28	62.36	-	-
A93	51.33	18.54	7.202	123.7
N94	54.62	36.96	8.195	114.0
M95	56.66	31.08	7.492	119.7
K96	54.22	37.51	7.369	125.5
G97	44.21	-	8.212	112.7
T98	61.09	71.56	8.074	112.9
L99	53.01	46.81	9.640	128.8
T100	62.59	69.82	9.384	126.3
V101	60.99	33.30	9.301	128.9
K102	57.42	34.75	8.780	133.5

Appendix F: Supplementary figures for Chapter 5

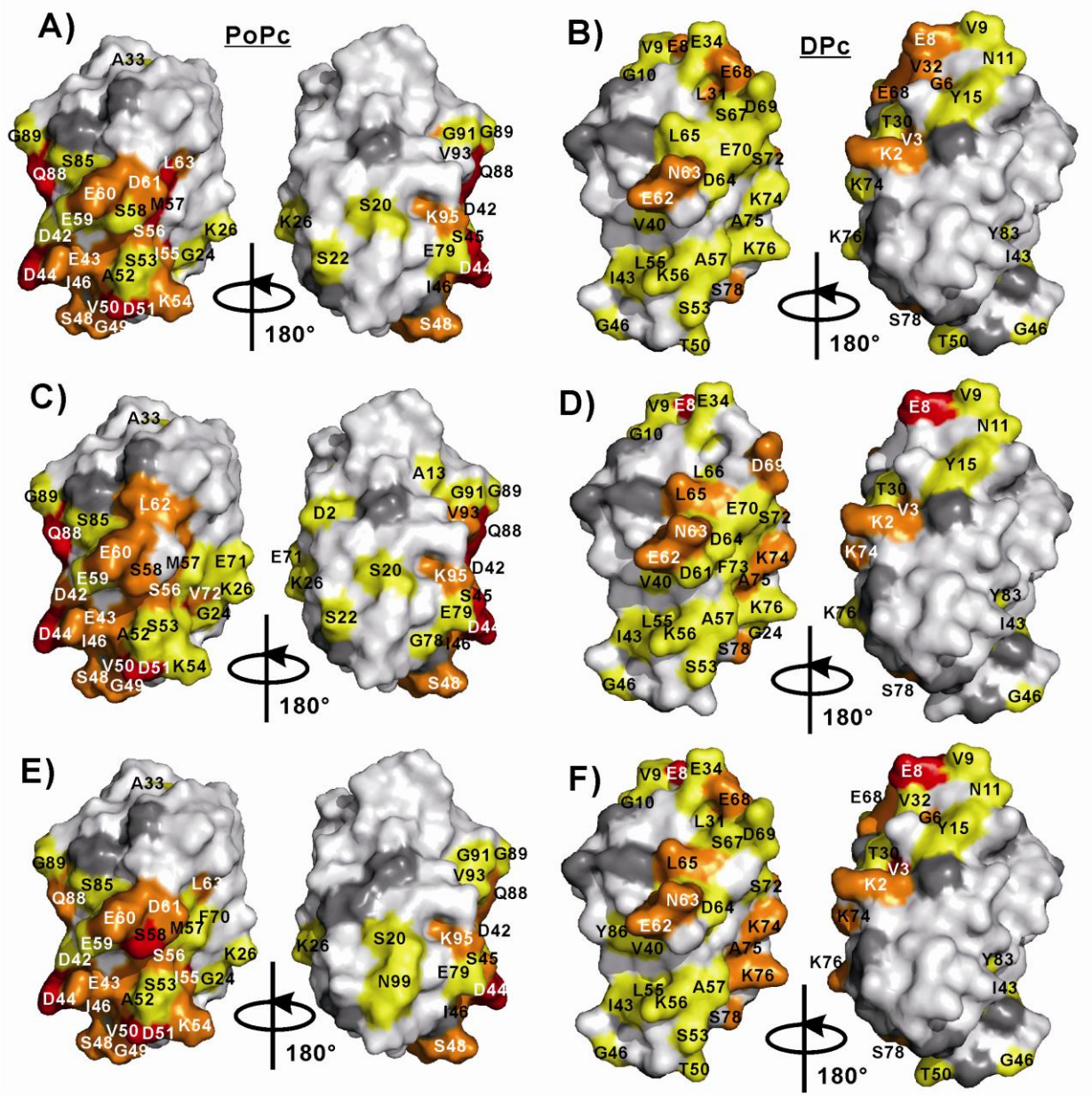

Figure F1: CSPs (extrapolated to 100% bound) mapped onto the protein surfaces from the binding of KKKKX (panels A and B), KKKKA (panels C and D) and AKKKK (panels E and F) to PoPc (left, PDB entry 1TKW[47]) and DPc (right, PDB entry 1KDI[43]). Color representations: red, $\Delta\delta_{ave} \geq 0.04$ ppm; orange, $0.04 > \Delta\delta_{ave} \geq 0.02$ ppm; yellow, $0.02 > \Delta\delta_{ave} \geq 0.01$ ppm; white, $\Delta\delta_{ave} < 0.01$ ppm; grey, no data or overlapping resonances.

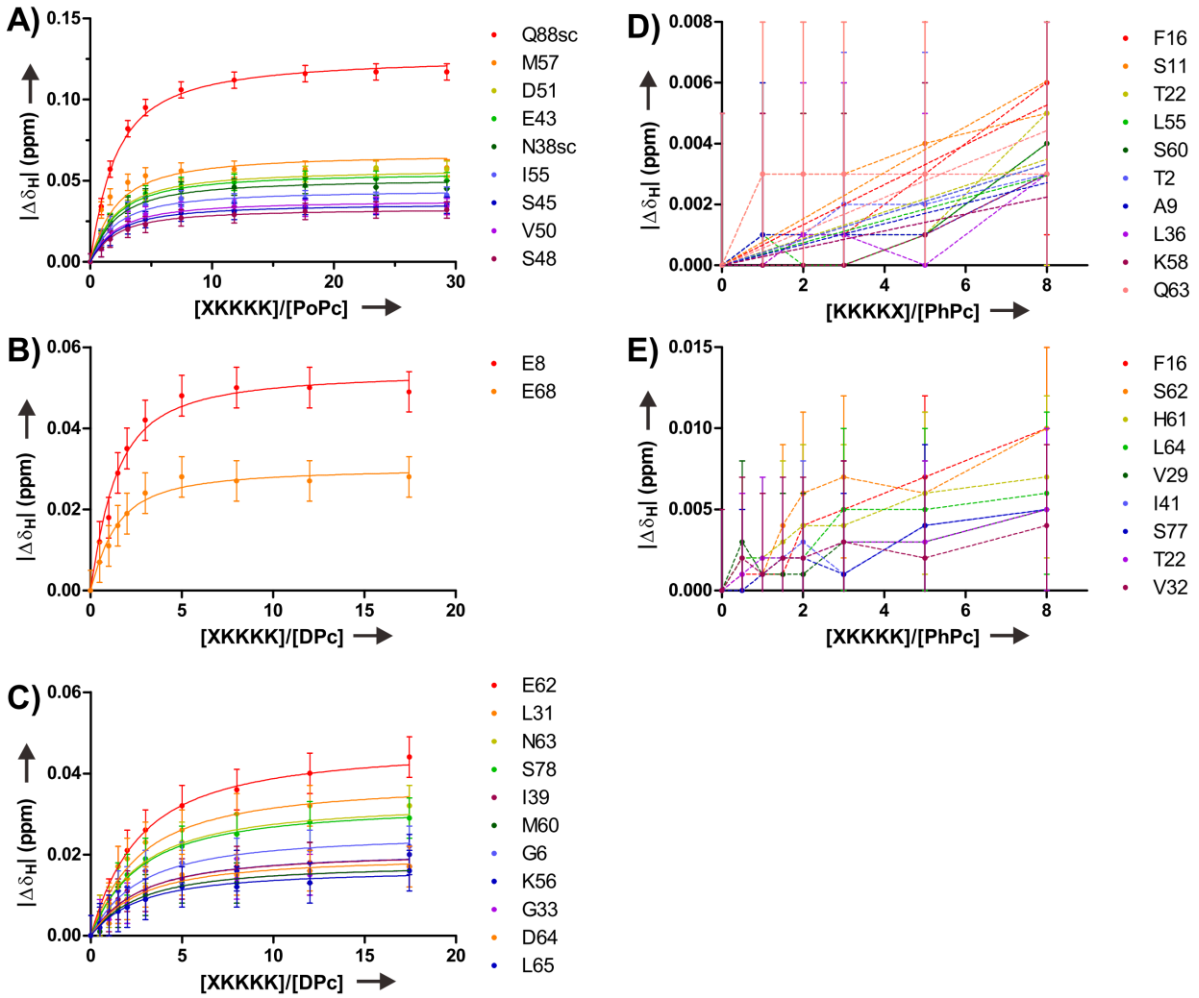

Figure F2: Chemical shift changes of Pcs resonances as a function of increasing [peptide]/[Pc] for peptides XKKKK and KKKKX. The dissociation constants of the corresponding peptides (Table 5.1) were obtained by simultaneous fitting to a 1:1 binding model for PoPc (solid lines) and by simulation for 2-site binding for DPc. (A) XKKKK with PoPc; (B) XKKKK with DPc, strong-binding residues; (C) XKKKK with DPc, weak-binding residues; (D) KKKKX with PhPc; (E) XKKKK with PhPc. The titration points for each residue in (D) and (E) are connected with dashed lines. Error bars represent ± 0.005 ppm.

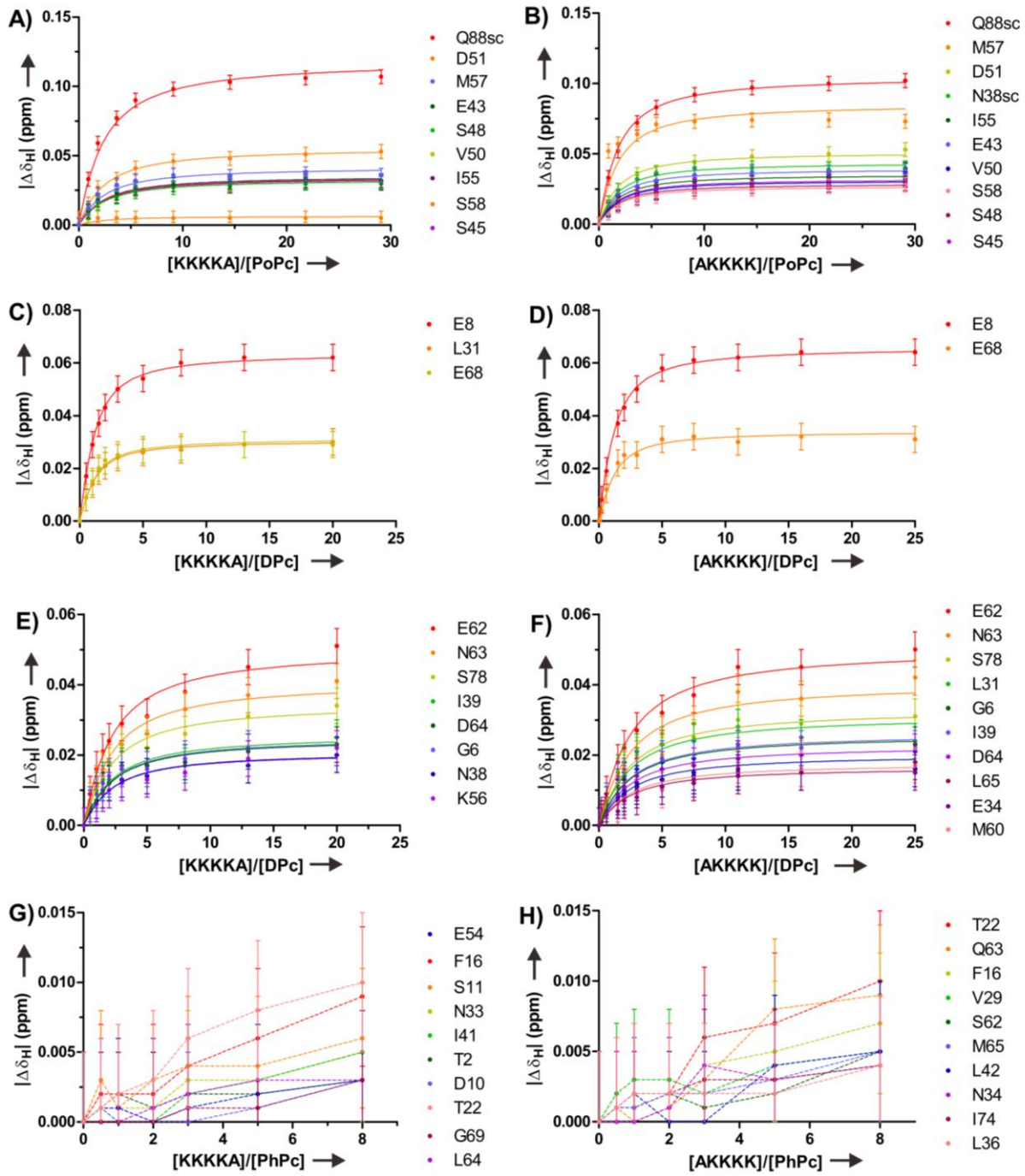

Figure F3: Chemical shift changes of Pcs resonances as a function of increasing [peptide]/[Pc] for peptides KKKKA and AKKKK. The residues which showed largest perturbations are shown. The dissociation constants of the corresponding peptides (Table 5.1) were obtained by simultaneous fitting to a 1:1 binding model for PoPc (solid lines) and by simulation for 2-site binding for DPc. (A) KKKKA with PoPc; (B) AKKKK with PoPc; (C) KKKKA with DPc, strong-binding residues; (D) AKKKK with DPc, strong-binding residues; (E) KKKKA with DPc, weak-binding residues; (F) AKKKK with DPc, weak-binding residues; (G) KKKKA with PhPc; (B) AKKKK with PhPc. The titration points for each residue in (G) and (H) are connected with dashed lines. Error bars represent ± 0.005 ppm.

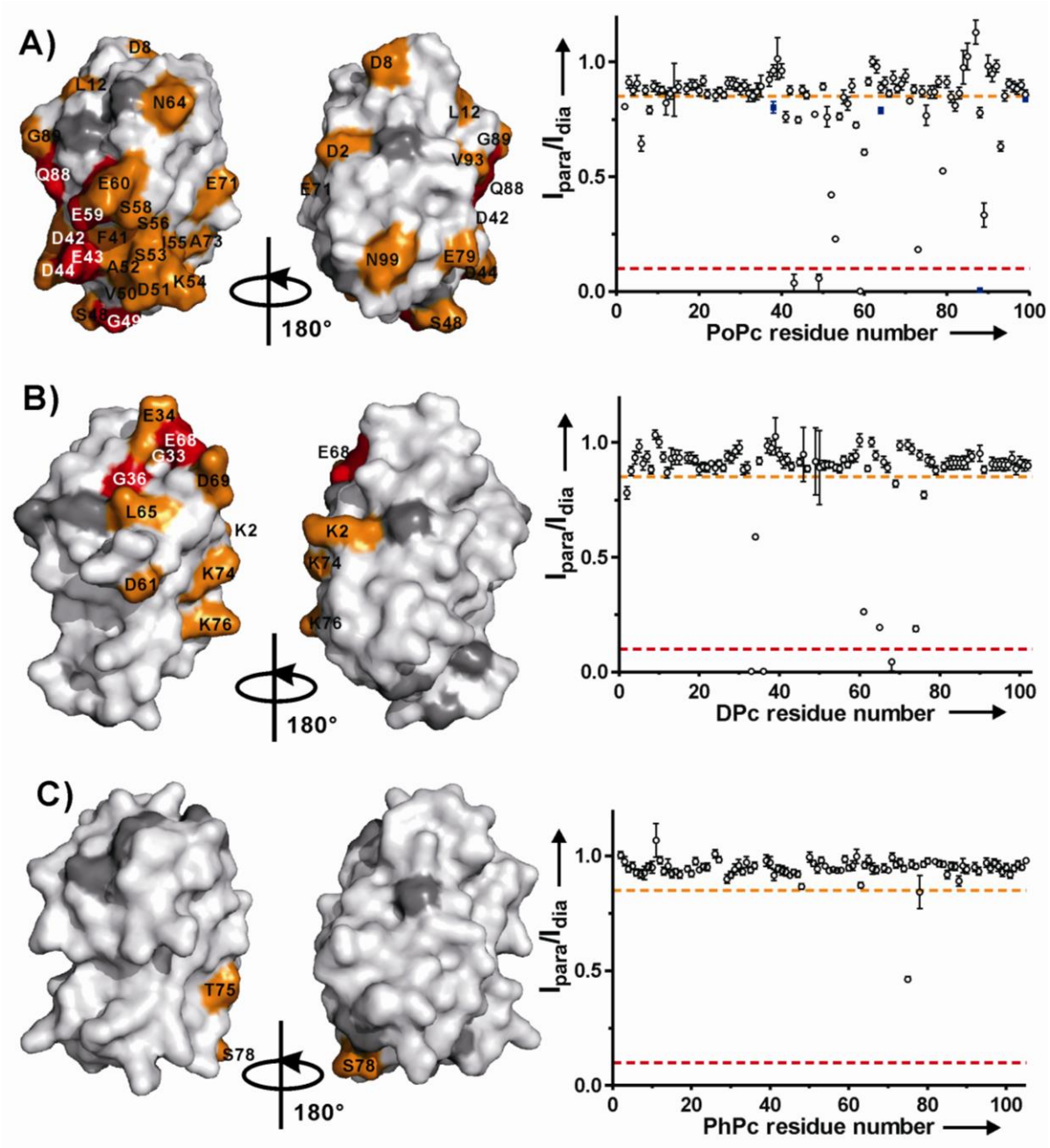

Figure F4: PRE effects in Pc-XKKKK complexes. Left: PRE maps of PoPc (A, PDB entry 1TKW[47]), DPc (B, PDB entry 1KDI[43]) and PhPc (C, PDB entry 2Q5B) bound to XKKKK peptide, color-coded on surface models of Pc: red, $I_{para}/I_{dia} < 0.1$; orange, $0.1 \leq I_{para}/I_{dia} < 0.85$; white, $I_{para}/I_{dia} \geq 0.85$; grey, prolines, unassigned, and overlapping resonances. Right: relative [^{1}H,^{15}N]-HSQC intensities of amides PoPc (A), DPc(B) and PhPc(C) in the complex with TOAC-containing peptides. For PoPc, the side chains are also included (blue squares). The dashed horizontal lines indicate $I_{para}/I_{dia} = 0.85$ (orange lines) and 0.1 (red lines). The error bars denote 2× standard deviations, derived from spectral noise levels using standard error propagation procedures.

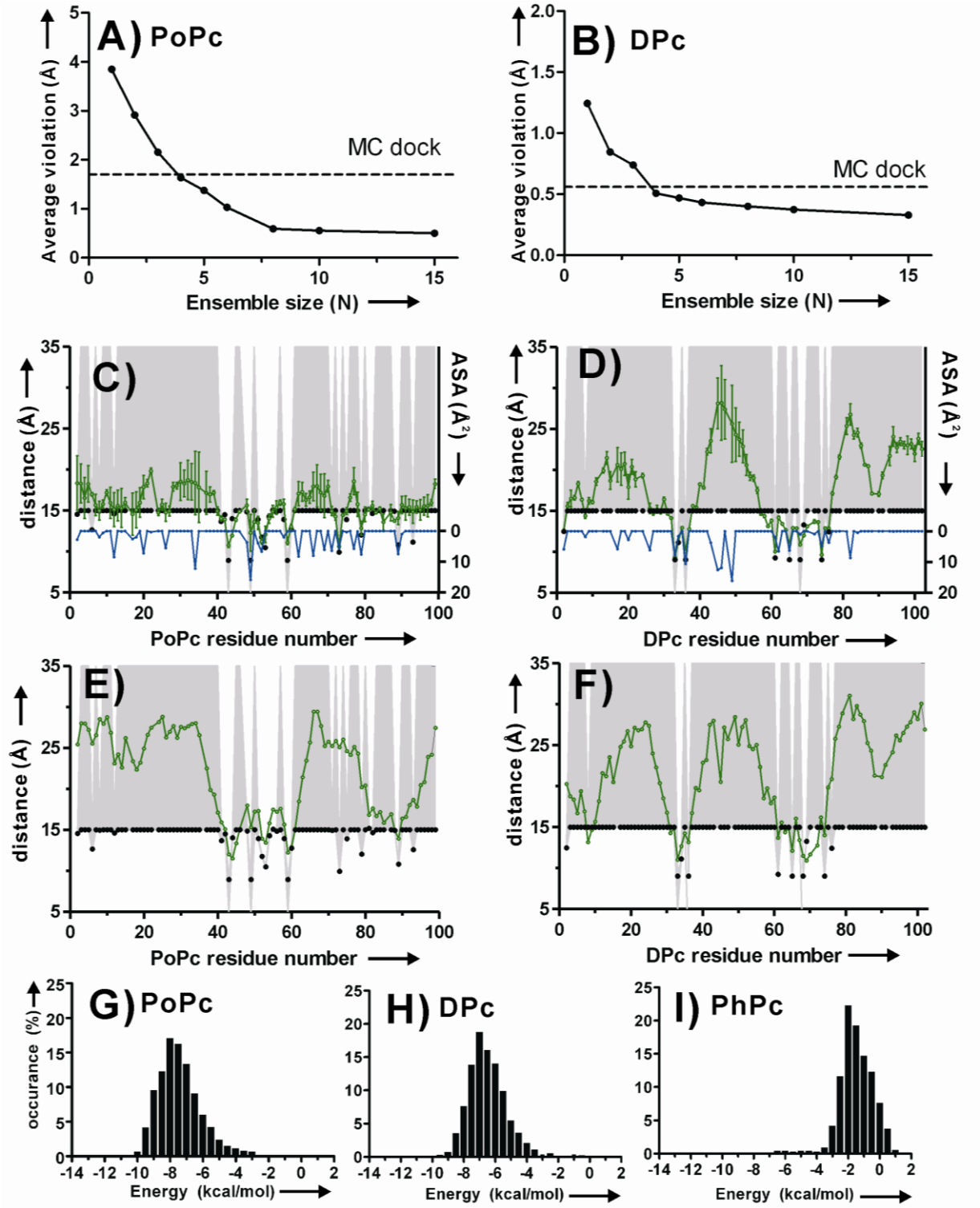

Figure F5: (A-B) Averaged distance violations against number of XKKKK peptides (N=1-6,8,10,15) in the ensemble docking for PoPc (A) and DPc (B). (C-D) Correlation of experimental distances (black dots) and back-calculated average distances (green circles with connecting lines) from the ensemble docking (N=6) of XKKKK bound to PoPc (C) and DPc (D). The average distances from the 20 lowest-energy solutions of the PRE driven ensemble docking are shown as black circles connected by black lines with error bars representing the standard deviation. Right y axes show the accessible surface area (ASA) of each amide. Grey areas indicate the error margins of the

experimental distances. (E-F) Comparison of experimental distances (black dots) and back-calculated average distances (green dots with connecting lines) between Pc amides and the 2000 ensembles of peptide paramagnetic oxygen atoms from MC simulations for PoPc (E) and DPc (F). Grey areas indicate the error margins of the experimental distances. (G-I) Histograms showing the energy distribution of 2000 ensembles from MC simulations: (G) PoPc-XKKKK, (H) DPc-XKKKK, (I) PhPc-XKKKK.