



Universiteit
Leiden
The Netherlands

Dynamics in electron transfer protein complexes

Bashir, Q.

Citation

Bashir, Q. (2010, October 27). *Dynamics in electron transfer protein complexes*. Retrieved from <https://hdl.handle.net/1887/16077>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/16077>

Note: To cite this publication please use the final published version (if applicable).

Appendices

Appendix A

Expression and purification of Cytochrome *c*

Cc is produced in *E. coli* BL21 competent cells. The competent cells are transformed with the relevant pUC-Cc plasmid (containing yeast haem lyase gene and yCc or hCc gene), spread on LB/Amp plates and incubated overnight at 37°C. The next morning a 3 ml culture (LB with 0.1 g/L Amp) is inoculated with the single colony and grown at 37°C and 250 rpm shaking until turbidity is evident (3 - 4 hours). One milliliter of the pre-culture is used to inoculate larger cultures. For expression of the ¹⁵N labelled protein, 1 ml of the pre-culture is used to inoculate 1 L of M9 minimal medium in a 2 L flask and grown at 37°C and 200 rpm shaking for 36 hours. For the expression of unlabelled protein, 1 ml of the pre-culture is used to inoculate 1.7 L of rich medium (LB with 0.1 g/L Amp and 1 mM KNO₃) in a 2 L flask. The culture is grown semi-anaerobically at 37°C with 160 rpm shaking, for 24 hours.

The cells are harvested by centrifugation at 7000 rpm for 15 minutes. The pellet is re-suspended in 50 mM Tris-HCl pH 6.8 containing 1 mM EDTA. A few milligrams of lysozyme, DNase and RNase are added and the mixture is stirred at 4°C until smooth (about 1 hour). The cells are broken by passing through a pre-cooled French press at 1200 lb/psi. The solution is mixed with 326 g/L of (NH₄)₂SO₄ and stirred at 4°C for about 30 minutes. The solution is centrifuged at 8000 rpm for 30 minutes and the red coloured supernatant is collected. The red coloured protein solution is put in dialysis bags (3000 M_r cut-off) leaving three-fourth of the bag empty to allow for expansion.

For yeast Cc and its mutants, the protein solution is dialyzed against 2 L of 46 mM NaPi pH 6.8 for 3 hours followed by a change of buffer and overnight dialysis. The dialysate is centrifuged at 6000 rpm for 15 minutes to get rid of any precipitate. The supernatant is loaded to CM column equilibrated with 46 mM NaPi pH 6.8. The protein is eluted with a linear gradient of the equilibration buffer containing 0-0.5 M NaCl.

For horse Cc and its mutants, the protein solution is dialyzed against 2 L of 46 mM NaPi pH 6.0 for 3 hours followed by a change of buffer and overnight dialysis. The dialysate is centrifuged at 6000 rpm for 15 minutes to get rid of any precipitate. The

supernatant is filtered and loaded to SP column equilibrated with 46 mM NaPi pH 6.0. The protein is eluted with a linear gradient of equilibration buffer containing 0-0.5 M NaCl.

The elution profile is monitored at 410 nm. The fractions containing Cc are combined, mixed with a few crystals of $K_3[Fe(CN)_6]$ to oxidize the protein, exchanged in 20 mM NaPi pH 6.0 containing 100 mM NaCl and concentrated in an Amicon fitted with a YM3 membrane. The concentrated protein solution is then applied to G-75 gel-filtration column equilibrated with 20 mM NaPi pH 6.0 containing 100 mM NaCl and eluted with the equilibration buffer. A single peak corresponding to the pure Cc is recorded at 280 nm. The protein concentration is estimated from the UV-vis absorbance at 410 nm ($\epsilon = 106.1 \text{ mM}^{-1}\text{cm}^{-1}$, $M_r = 12,100$) for both Cc.

Expression and purification of cytochrome *c* peroxidase

E. coli BL21 competent cells are transformed with the pT7-CcP plasmid, spread on LB/Amp plates and incubated overnight at 37°C. The next morning a 3 ml culture (SB with 0.1 g/L Amp) is inoculated with single colony and grown at 37°C and 250 rpm shaking until turbidity is evident (3 - 4 hours). Larger culture (50 ml SB/Amp in a 250 ml flask) is inoculated with 0.5 ml of the pre-culture and further grown for 3 hours at 37°C and 250 rpm shaking. The final culture (750 ml SB/Amp in a 2 L flask) is inoculated with 0.75 ml of the above culture and grown for 10-11 hours at 37°C and 250 rpm shaking followed by addition of 1.6 ml of IPTG (0.125 g/ml stock, sterile-filtered) and followed by 3 hours of incubation.

The cells are harvested by centrifugation at 7000 rpm for 15 minutes. The pellet is re-suspended in the minimal volume of the lysis buffer (0.1 M KPi pH 6.0 containing 1 mM EDTA and 1 mM PMSF). A few milligrams of lysozyme are added and the mixture is stirred at 4°C until smooth (about 1 hour). The smooth cell paste is frozen in liquid nitrogen and stored at -80°C. The frozen cell pellet is thawed, a few milligrams of DNase, RNase and 10 ml of $MgCl_2$ are added and the mixture is stirred at 4°C for about 1 hour. The mixture is then centrifuged at 8500 rpm for 30 minutes and the supernatant is collected.

A 10-50 fold molar excess of hemin is dissolved in 1 ml of 0.1 M NaOH and diluted with 10 ml of the lysis buffer. The hemin solution is added to the protein solution and stirred at 4°C for 1 hour during which the haem is incorporated in the apoenzyme. The protein solution is then acidified to pH 5.0 with 1.0 N acetic acid and frozen at -20°C. The protein solution is thawed and incubated at 37°C for 2-3 hours. The protein solution is then centrifuged at 12000 rpm for 25 minutes to get rid of the precipitated proteins. The supernatant is filled in dialysis bags (6000 M_r cut-off) and dialyzed overnight against 3 L of 50 mM sodium acetate pH 5.0. The dialyzed protein solution is centrifuged at 8000 rpm for 15 minutes and the supernatant is loaded on a DEAE column equilibrated with 50 mM sodium acetate pH 5.0. The protein is eluted with a linear gradient of the equilibration buffer containing 0-0.4 M NaCl. The protein elution is monitored at 410 nm. The fractions containing CcP are pooled and incubated with 10 mM DTT for 2-3 hours, in case of cysteine mutants. The protein is exchanged into 20 mM NaPi pH 6.0 containing 100 mM NaCl and 1 mM DTT, and concentrated in an Amicon fitted with a YM10 membrane. The concentrated protein is applied to G-75 gel-filtration column equilibrated with the 20 mM NaPi pH 6.0 containing 100 mM NaCl and 1 mM DTT and eluted with the equilibration buffer. A single peak corresponding to the pure CcP is recorded at 280 nm. The protein concentration is estimated from the UV-vis absorbance at 408 nm ($\epsilon = 98 \text{ mM}^{-1}\text{cm}^{-1}$, $M_r = 34,100$).

Site-specific spin labeling of cytochrome *c* peroxidase

The required CcP mutant is exchanged in 100 mM Tris-HCl buffer pH 8.0 containing 0.1 M NaCl and incubated with 10 mM DTT at room temperature for about 2 hours. Then DTT is removed by passing the protein solution through a PD-10 column. The protein concentration is measured and the protein solution is divided into two equal portions. A 7-fold excess of MTSL or MTS is added to each portion and the mixture is incubated at room temperature for overnight.

Next morning the unreacted MTSL or MTS is removed through a PD-10 column. The protein solution containing CcP-MTS or CcP-MTSL is exchanged in 20 mM NaPi pH 6.0 containing 0.1 M NaCl and concentrated to 0.6-0.8 mM.

Appendix B

Input file (for XPLOR-NIH) to determine distances from CcP-SL to Cc amides

```

! NB SL1,2,3,4 = N38,N200,T288,S263 (this order)
! N38 = store1, store2 (calc., obs.)
! N200 = store3, store4 (calc., obs.)
! T288 = store5, store6 (calc., obs.)
! S263 = store7, store8 (calc., obs.)

set mess=off end
set echo=off end

!-----VARIABLES -----

eval ($rescc = 103)           ! last residue in Cc
eval ($first = -5)           ! 1st residue in Cc
eval ($nrrescc = 107)        ! number of residues in Cc minus 1
eval ($nrprocc = 4)          ! number of PRO in Cc
eval ($nrrealcc = $nrrescc - $nrprocc)
eval ($nrstruc = 1)          ! number of structures in the ensemble
eval ($ensm = "wtcc_new_1701") ! name of the ensemble
eval ($frmajor = 0.000001)    ! fraction of well-defined form
eval ($frminor = 1 - $frmajor) ! fraction of the ensemble
eval ($startfile = "cc_ccp_4sl.pdb")
eval ($ccpfile = "CCP.pdb.con") ! name of the CcP file in the same
                                ! orientation as the one used to create ensemble
eval ($SL_FILE="cc_ccp_rand_SL_inclCc_01.pdb.SL")
                                ! name of file with new spin label oxygen positions

!-----PARAMETERS & STRUCTURE -----

structure @cc_ccp/cc_ccp_4sl.psf end

parameter @cc_ccp_4sl.par end

coor @$startfile

coor init SELE=(segid=SL) end
coor @$SL_FILE

@learn_ccp.xpl
@learn_cc.xpl

!write coor OUTPUT=test1.pdb end

!-----

set mess=off end
set echo=off end

eval ($count=$first+1)

while ($count < $rescc) loop A1
  vector do (store1=0.00) (atom "cytc" $count HN)
  vector do (store3=0.00) (atom "cytc" $count HN)

```

```

        vector do (store5=0.00) (atom "cytc" $count HN)
        vector do (store7=0.00) (atom "cytc" $count HN)
eval ($count=$count+1)
end loop A1

vector idend ( store9 ) ((name ca) and (segid="cytc"))

eval ($struc1 = 0)
eval ($struc2 = 0)
eval ($struc3 = 0)
eval ($struc4 = 0)

!-----READ STRUCTURES-----

!NOTE in the structure, only have cytc coordinates present, otherwise
!loop !becomes too long
! (loop-too-long error)

coor DISP=COMP @$ccpfile end                !read CcP file from
                                              !MC in COMP
coor fit SELE=(segid="ccp" and name = CA) end !fit CcP in complex
                                              !to CcP from !MC
coor copy SELE=(not segid="cytc") end        !copy CcP to COMP

!write coor OUTPUT=test1c.pdb end

eval ($w = 1)

while ($w LE $nrstruc) loop A2

coor swap SELE=(segid="cytc") end            !move MC CA atoms to main and
                                              !initialize
coor init SELE=(segid="cytc") end
coor swap SELE=(segid="cytc") end            !move cytc back to main for fitting
eval ($file = "cytc." + encode($w) + ".pqr.ed")
coor DISP=COMP @$file end                    !read new MC CA atoms
coor fit SELE=(segid="cytc" and name CA) end  !fit all main atoms,
                                              !using Cc CA atoms
coor swap SELE=(not segid="cytc") end        !replace fitted CcP+SL by
                                              !original
coor copy SELE=(not segid="cytc") end        !copy original CcP+SL for
                                              !later use CcP is now in same orientation as ccpfile and Cc is
                                              !fitted to new MC Cc orientation

!write coor OUTPUT=test2.pdb end

!-----CALCULATING DISTANCES FOR THE ENSEMBLE -----

eval ($a = 1)
while ($a < 5) loop main                    !number of SL + 1

eval ($cont1="no")
eval ($cont2="no")
eval ($cont3="no")
eval ($cont4="no")

for $b02 in ID (store9) loop contact        !CA atoms of Cytc; check for
                                              !contacts
    pick bond (ID $b02) (atom SL $a 01) geom
    if ($RESULT < 5.0) then eval ($cont1 = "yes") end if

```

```

    pick bond (ID $b02) (atom SL $a O2) geom
    if ($RESULT < 5.0) then eval ($cont2 = "yes") end if
    pick bond (ID $b02) (atom SL $a O3) geom
    if ($RESULT < 5.0) then eval ($cont3 = "yes") end if
    pick bond (ID $b02) (atom SL $a O4) geom
    if ($RESULT < 5.0) then eval ($cont4 = "yes") end if
end loop contact

eval ($count=$frst+1)
while ($count LE $rescc) loop A1

    if ($count # 0 ) then

        vector show elem (resn) (atom "cytc" $count N)
        if ($RESULT # "PRO") then

            eval ($b01 = 0)
            eval ($avgA = 0)

            if ($cont1="no") then
                pick bond (atom "cytc" $count HN) (atom SL $a O1) geom
                eval ($a1=$RESULT^(-6))
                eval ($b01 = $b01 +1)
            else eval ($a1 = 0)
            end if

            if ($cont2="no") then
                pick bond (atom "cytc" $count HN) (atom SL $a O2) geom
                eval ($a2=$RESULT^(-6))
                eval ($b01 = $b01 +1)
            else eval ($a2 = 0)
            end if

            if ($cont3="no") then
                pick bond (atom "cytc" $count HN) (atom SL $a O3) geom
                eval ($a3=$RESULT^(-6))
                eval ($b01 = $b01 +1)
            else eval ($a3 = 0)
            end if

            if ($cont4="no") then
                pick bond (atom "cytc" $count HN) (atom SL $a O4) geom
                eval ($a4=$RESULT^(-6))
                eval ($b01 = $b01 +1)
            else eval ($a4 = 0)
            end if

            if ($b01 # 0) then
                eval ($avgA=($a1+$a2+$a3+$a4)/$b01)*1E10)
            else eval ($avgA = 0)
            end if

            if ($a = 1) then
                vector show element (store1) (atom "cytc" $count HN)
                eval ($v3 = $RESULT + $avgA)
                vector do (store1=$v3) (atom "cytc" $count HN)
                if ($avgA # 0) then eval ($struc1 = $struc1 + 1) end if
                display QQQQQQ $avgA      $count      $struc1      $b01      $a1      $a2      $a3
            $a4
            elseif ($a = 2) then

```

```

        vector show element (store3) (atom "cytc" $count HN)
        eval ($v3 = $RESULT + $avgA)
        vector do (store3=$v3) (atom "cytc" $count HN)
        if ($avgA # 0) then eval ($struc2 = $struc2 + 1) end if
    elseif ($a = 3) then
        vector show element (store5) (atom "cytc" $count HN)
        eval ($v3 = $RESULT + $avgA)
        vector do (store5=$v3) (atom "cytc" $count HN)
        if ($avgA # 0) then eval ($struc3 = $struc3 + 1) end if
    elseif ($a = 4) then
        vector show element (store7) (atom "cytc" $count HN)
        eval ($v3 = $RESULT + $avgA)
        vector do (store7=$v3) (atom "cytc" $count HN)
        if ($avgA # 0) then eval ($struc4 = $struc4 + 1) end if
    end if
end if
end if
eval ($count=$count+1)

end loop A1

eval ($a = $a + 1)

end loop main

eval ($w = $w + 1)
end loop A2

! calculate number of structures used for each spin label
display SSSSSS $struc1 $struc2 $struc3 $struc4
eval ($struc1 = $struc1/$nrrealcc)
eval ($struc2 = $struc2/$nrrealcc)
eval ($struc3 = $struc3/$nrrealcc)
eval ($struc4 = $struc4/$nrrealcc)
display SSSSSS $struc1 $struc2 $struc3 $struc4

!-----CALCULATING DISTANCES FOR THE WELL-DEFINED ONE-----

coor init SELE=(all) end
coor @$startfile
coor init SELE=(segid=SL) end
coor @$SL_FILE

write coor OUTPUT=test4.pdb end

!calculate correction factors for fraction well-defind for each spin
!label
eval ($corr1 = $struc1*$frrmajor/$frrminor)
eval ($corr2 = $struc2*$frrmajor/$frrminor)
eval ($corr3 = $struc3*$frrmajor/$frrminor)
eval ($corr4 = $struc4*$frrmajor/$frrminor)

eval ($a = 1)
while ($a < 5) loop main2          !number of SL + 1

eval ($cont1="no")
eval ($cont2="no")
eval ($cont3="no")
eval ($cont4="no")

```

```

for $b02 in ID (store9) loop contact !CA atoms of Cc; check for contacts
pick bond (ID $b02) (atom SL $a O1) geom
if ($RESULT < 5.0) then eval ($cont1 = "yes") end if
pick bond (ID $b02) (atom SL $a O2) geom
if ($RESULT < 5.0) then eval ($cont2 = "yes") end if
pick bond (ID $b02) (atom SL $a O3) geom
if ($RESULT < 5.0) then eval ($cont3 = "yes") end if
pick bond (ID $b02) (atom SL $a O4) geom
if ($RESULT < 5.0) then eval ($cont4 = "yes") end if
end loop contact

eval ($count=$first+1)
while ($count LE $rescc) loop A1

  if ($count # 0 ) then

    vector show elem (resn) (atom "cytc" $count N)
    if ($RESULT # "PRO") then

      eval ($b01 = 0)
      eval ($avgA = 0)

      if ($cont1="no") then
        pick bond (atom "cytc" $count HN) (atom SL $a O1) geom
        eval ($a1=$RESULT^(-6))
        eval ($b01 = $b01 +1)
      else eval ($a1 = 0)
      end if

      if ($cont2="no") then
        pick bond (atom "cytc" $count HN) (atom SL $a O2) geom
        eval ($a2=$RESULT^(-6))
        eval ($b01 = $b01 +1)
      else eval ($a2 = 0)
      end if

      if ($cont3="no") then
        pick bond (atom "cytc" $count HN) (atom SL $a O3) geom
        eval ($a3=$RESULT^(-6))
        eval ($b01 = $b01 +1)
      else eval ($a3 = 0)
      end if

      if ($cont4="no") then
        pick bond (atom "cytc" $count HN) (atom SL $a O4) geom
        eval ($a4=$RESULT^(-6))
        eval ($b01 = $b01 +1)
      else eval ($a4 = 0)
      end if

      if ($b01 # 0) then
        eval ($avgA=(( $a1+$a2+$a3+$a4)/$b01)*1E10)
      else eval ($avgA = 0)
      end if

      if ($a = 1) then
        vector show element (store1) (atom "cytc" $count HN)
        eval ($v3 = $RESULT + $avgA*$corrl)
        vector do (store1=$v3) (atom "cytc" $count HN)
      end if
    end if
  end if
end loop A1

```

```

! display QQQQQQ $avgA    $count    $corr1    $b01    $a1    $a2    $a3
$a4
elseif ($a = 2) then
  vector show element (store3) (atom "cytc" $count HN)
  eval ($v3 = $RESULT + $avgA*$corr2)
  vector do (store3=$v3) (atom "cytc" $count HN)
  !display QQQQQQ $avgA    $count    $corr2    $b01    $v3    $RESULT
!$a1    $a2    $a3    $a4
elseif ($a = 3) then
  vector show element (store5) (atom "cytc" $count HN)
  eval ($v3 = $RESULT + $avgA*$corr3)
  vector do (store5=$v3) (atom "cytc" $count HN)
  !display QQQQQQ $avgA    $count    $corr3    $b01    $v3    $RESULT
!$a1    $a2    $a3    $a4
elseif ($a = 4) then
  vector show element (store7) (atom "cytc" $count HN)
  eval ($v3 = $RESULT + $avgA*$corr4)
  vector do (store7=$v3) (atom "cytc" $count HN)
end if
end if
end if
eval ($count=$count+1)

end loop A1

eval ($a = $a + 1)

end loop main2

!-----WRITE OUTPUT -----

! read distances from observed PREs for comparison

@n38c.xpl
set display="ens_viol_n38c.dat" end
display Experimental and ensemble-averaged (r-6) distances for yeast Cc
and N38C-SL CcP
display name of the ensemble:          $ensm
display name of ccp/cc startfile:      $startfile
display name of ccp orientation file:   $ccpfile
display name of spin label coordinate file: $SL_FILE
display # structures used:              $struc1
display fraction well-defined: $frmajor    virtual # structure: $corr1
display Date: $DATE    Time: $TIME
display
display RES    ATOM    d(obs) d(calc)
set display=OUTPUT end

for $ide in ID (store2) loop OUT
  vector show elem (store1) (ID $ide)
  eval ($w1 = (($RESULT/(1E10*($struc1 + $corr1)))^(-1/6)))
  vector show elem (resi) (ID $ide)
  eval ($w2 = $RESULT)
  vector show elem (name) (ID $ide)
  eval ($w3 = $RESULT)
  vector show elem (store2) (ID $ide)
  eval ($w4 = $RESULT)
  set display="ens_viol_n38c.dat" end
  display $w2    $w3    $w4    $w1
  set display=OUTPUT end

```

```

end loop OUT

@n200c.xpl
set display="ens_viol_n200c.dat" end
display Experimental and ensemble-averaged (r-6) distances for yeast Cc
and N200C-SL CcP
display name of the ensemble:          $ensm
display name of ccp/cc startfile:      $startfile
display name of ccp orientation file:  $ccpfile
display name of spin label coordinate file: $SSL_FILE
display # structures used:             $struc2
display fraction well-defined: $frmajor   virtual # structure: $corr2
display Date: $DATE    Time: $TIME
display
display RES    ATOM    d(obs) d(calc)
set display=OUTPUT end

for $side in ID (store4) loop OUT
  vector show elem (store3) (ID $side)
  eval ($w1 = (($RESULT/(1E10*($struc2 + $corr2)))^(-1/6)))
  vector show elem (resi) (ID $side)
  eval ($w2 = $RESULT)
  vector show elem (name) (ID $side)
  eval ($w3 = $RESULT)
  vector show elem (store4) (ID $side)
  eval ($w4 = $RESULT)
  set display="ens_viol_n200c.dat" end
  display $w2      $w3      $w4      $w1
  set display=OUTPUT end
end loop OUT

@t288c.xpl
set display="ens_viol_t288c.dat" end
display Experimental and ensemble-averaged (r-6) distances for yeast Cc
and T288C-SL CcP
display name of the ensemble:          $ensm
display name of ccp/cc startfile:      $startfile
display name of ccp orientation file:  $ccpfile
display name of spin label coordinate file: $SSL_FILE
display # structures used:             $struc3
display fraction well-defined: $frmajor   virtual # structure: $corr3
display Date: $DATE    Time: $TIME
display
display RES    ATOM    d(obs) d(calc)
set display=OUTPUT end

for $side in ID (store6) loop OUT
  vector show elem (store5) (ID $side)
  eval ($w1 = (($RESULT/(1E10*($struc3 + $corr3)))^(-1/6)))
  vector show elem (resi) (ID $side)
  eval ($w2 = $RESULT)
  vector show elem (name) (ID $side)
  eval ($w3 = $RESULT)
  vector show elem (store6) (ID $side)
  eval ($w4 = $RESULT)
  set display="ens_viol_t288c.dat" end
  display $w2      $w3      $w4      $w1
  set display=OUTPUT end
end loop OUT

```

```
@s263c.xpl
set display="ens_viol_s263c.dat" end
display Experimental and ensemble-averaged (r-6) distances for yeast Cc
and S263C-SL CcP
display name of the ensemble: $ensm
display name of ccp/cc startfile:      $startfile
display name of ccp orientation file:   $ccpfile
display name of spin label coordinate file: $SL_FILE
display # structures used:              $struc4
display fraction well-defined: $frmajor   virtual # structure: $corr4
display Date: $DATE      Time: $TIME
display
display RES      ATOM      d(obs) d(calc)
set display=OUTPUT end

for $ide in ID (store8) loop OUT
  vector show elem (store7) (ID $ide)
  eval ($w1 = (($RESULT/(1E10*($struc4 + $corr4)))^(-1/6)))
  vector show elem (resi) (ID $ide)
  eval ($w2 = $RESULT)
  vector show elem (name) (ID $ide)
  eval ($w3 = $RESULT)
  vector show elem (store8) (ID $ide)
  eval ($w4 = $RESULT)
  set display="ens_viol_s263c.dat" end
  display $w2      $w3      $w4      $w1
  set display=OUTPUT end
end loop OUT

stop
```

Appendix C

Input file (for XPLOR-NIH) to determine DRMS between ensemble structure and XRD structure

```

set mess=off end
set echo=off end

!-----VARIABLES -----

eval ($rescc = 103)           ! last residue in Cc
eval ($first = -4)           ! 1st residue in Cc (NB -4 in ensemble)
eval ($nrrescc = 108)        ! number of residues in Cc
eval ($resccp = 294)
eval ($firstccp = 2)
eval ($nrstepsc = 3)
eval ($nrstepscp = 10)
eval ($nrstruc = 1)           ! number of structures in the ensemble
eval ($ensm = "wtcc_new_1701") ! name of the ensemble (whatever you like)
eval ($startfile = "cc_ccp_4sl.pdb")
eval ($ccpfile = "CCP.pdb.con") ! name of the CcP file in the same
                                ! orientation as the one used to create ensemble
eval ($SL_FILE="cc_ccp_rand_SL_inclCc_01.pdb.SL")
                                ! name of file with new spin label oxygen

!-----PARAMETERS & STRUCTURE -----

structure @cc_ccp_4sl.psf end

parameter @cc_ccp_4sl.par end

coor @$startfile

coor init SELE=(segid=SL) end
coor @$SL_FILE

@learn_ccp.xpl
@learn_cc.xpl

!write coor OUTPUT=test1.pdb end

!-----

set mess=off end
set echo=off end

!-----READ STRUCTURES-----

!NOTE in the structure, only have cytc coordinates present, otherwise
!loop becomes too long
! (loop-too-long error)

coor DISP=COMP @$ccpfile end           !read CcP file from MC in COMP
coor fit SELE=(segid="ccp" and name = CA) end !fit CcP in complex
                                           !to CcP from MC
coor copy SELE=(not segid="cytc") end    !copy CcP to COMP

```

```
!write coor OUTPUT=test1b.pdb end

eval ($w = 1)

set display="rmsd.dat" end
display struc rmsd drms
set display=OUTPUT end
while ($w LE $nrstruc) loop A2

  coor swap SELE=(segid="cytc") end          !move MC CA atoms to main
                                              !and initialize

  coor init SELE=(segid="cytc") end
  coor swap SELE=(segid="cytc") end          !move Cc back to main for fitting
  eval ($file = "cytc." + encode($w) + ".pqr.ed")
  coor DISP=COMP @@$file end                !read new MC CA atoms
  coor rms SELE=(segid="cytc" and name CA) end !rms all main atoms, using
                                              !Cc CA atoms

  eval ($rmsd = $RESULT)

  eval ($drms = 0)
  eval ($g01 = $first)
  eval ($g03 = 0)
  while ($g01 LE $rescc) loop G1

    if ($g01 = 0) then eval ($g01 = $g01 + 1) end if
    if ($g01 = 102) then eval ($g01 = $g01 + 1) end if

    eval ($g02 = $firstccp)
    while ($g02 LE $resccp) loop G2

      pick bond (atom "cytc" $g01 CA) (atom "ccp" $g02 CA) geom
      eval ($d1=$RESULT)
      coor swap SELE=(all) end
      pick bond (atom "cytc" $g01 CA) (atom "ccp" $g02 CA) geom
      eval ($d2=$RESULT)
      coor swap SELE=(all) end
      eval ($d3 = $d2 - $d1)
      eval ($drms = $drms + ABS($d3))

      eval ($g03 = $g03 + 1)                  !total number of pairs
      eval ($g02 = $g02 + $nrstepsccp)
    end loop G2

  eval ($g01 = $g01 + $nrstepscc)
end loop G1

eval ($drms = $drms/$g03)

write coor OUTPUT=test2.pdb end

set display="rmsd.dat" end
display $w      $rmsd  $drms
set display=OUTPUT end

eval ($w = $w + 1)
end loop A2

stop
```

Appendix D

Input file (for XPLOR-NIH) to determine theta and phi

```

set mess=off end
set echo=off end

!-----VARIABLES -----
eval ($nrstruc = 1701)           ! number of structures in
                                ! the ensemble
eval ($startfile = "cc_ccp_4sl.pdb")
eval ($ccpfile = "ccp_from_ensemble.pdb") ! name of the CcP ensemble
                                ! file
eval ($dum="cytc-cm.wt.1701.pdb.ed2")    ! dummy atoms of CoM of Cc
                                ! from ensemble
eval ($outfile = "wt_new_1701_reori.pdb") ! output file

!-----PARAMETERS & STRUCTURE -----

structure @cc_ccp_4sl.psf end

parameter @cc_ccp_4sl.par end

coor
  @$startfile

@learn_ccp.xpl
@learn_cc.xpl

topology mass G 1.0 residue GB atom G type=G charge=0 end end end

set mess=off end
set echo=off end

segment
  molecule name="GB" number=2 end
end

vector do (SEGid="C") (atom * * G)

topology mass D 1.0 residue DUM atom CM type=D charge=0 end end end

segment
  molecule name="DUM" number=$nrstruc end
end

!-----READ STRUCTURES-----

vector do (segid="  ") (segid="ccp")

coor DISP=COMP @$ccpfile      !read CCP file from MC in COMP
coor DISP=COMP @$dum end      !read dummies file from MC in COMP

vector do (segid="ccp") (segid="  " and not resn DUM)

```

```

coor fit SELE=(segid="ccp" and name = CA) end          !fit CCP in complex
                                                    !to CCP from MC
coor swap SELE=(all) end
coor copy SELE=(resn DUM) end                        !copy DUM to COMP
coor swap SELE=(all) end

!write coor OUTPUT=test1.pdb end

coor orient SELE=(segid="cytc" and not resn HEM) MASS=true LSQ=false end
vector do (X=0) (atom "C" 2 G)
vector do (Y=0) (atom "C" 2 G)
vector do (Z=0) (atom "C" 2 G)

coor orient SELE=(segid="ccp" and not resn HEM) MASS=true LSQ=false end
vector do (X=0) (atom "C" 1 G)
vector do (Y=0) (atom "C" 1 G)
vector do (Z=0) (atom "C" 1 G)

vector show (x) (atom "C" 2 G)
eval ($xb1=$RESULT)
vector show (y) (atom "C" 2 G)
eval ($yb1=$RESULT)
vector show (z) (atom "C" 2 G)
eval ($zb1=$RESULT)

eval ($r=SQRT($xb1^2+$yb1^2+$zb1^2))
eval ($th=ACOS($zb1/$r))

eval ($xr=-$yb1/$xb1)
coor rota sele=(all) cent=(0 0 0) axis=($xr, 1, 0) $th end

vector show (x) (atom "C" 1 G)
vector show (y) (atom "C" 1 G)
vector show (z) (atom "C" 1 G)

vector show (x) (atom "C" 2 G)
eval ($x00=$RESULT)
vector show (y) (atom "C" 2 G)
eval ($y00=$RESULT)
vector show (z) (atom "C" 2 G)
eval ($z00=$RESULT)

write coor OUTPUT=$outfile end

stop

```

Appendix E

Input file (for XPLOR-NIH) for rigid body docking of Cc-CcP complex using PRE derived distance restraints

```

set mess=off end
set echo=off end

!-----PARAMETERS & STRUCTURE -----

parameter @cc_ccp_4sl.par end

structure @cc_ccp_4sl.psf end

eval ($inifile="cc_ccp_4sl.pdb")
coordinates @$inifile

@learn.cc_ccp.par

constraints fix (segid="ccp" or segid="SLA" or segid="SLB" or
segid="SLC" or segid="SLD") end

!-----DEFINE VARIABLES-----

eval ($a01 = 1)           !cycle counter
eval ($a05 = 10)          !total nr cycles / $a53
eval ($a53 = 1000)        !nr of steps
eval ($a54 = 0.01)        !timestep in ps
eval ($a61 = $a54*$a53)    !time per cycle (ps)
eval ($a80 = $a61*$a05)    !total time (ps)
eval ($a58=30.0)          !velocity factor
eval ($a56=1.5)           !fbeta
eval ($sqconst=0.2)        !general scaling rest. !0.002
eval ($a55=300.0)         !TBATH
eval ($a57=0.0)           !signpenalty factor
eval ($a59=500)           !.pdb writing threshold
eval ($van=100.0)         ! vdw writing threshold
eval ($a59a=0.01*$a59)    !vx-reset minimum
eval ($a60=0)             !.pdb lag factor           !5
eval ($a301=1.0)          !VDW repel scale factor
eval ($a93=1)             !time-spent factor: 0<$a93<1
eval ($sqexp=2)           !sqexponent in the NOE statement
eval ($viol_cutoff=200.0) !violation writing threshold
eval ($scale=1.0)         !scale factor for CL2a (d<13A)
eval ($low1=4.0)          !lower(d_minus) limit (restraints with lower limit
!only)
eval ($up1=100.0)         !upper(d_plus) limit (restraints with lower limit
!only)
eval ($low2a=12.5)        !lower(d_minus) limit (SLA restraints with
!upper limit only)
eval ($low2b=13.0)        !lower(d_minus) limit (SLB restraints with upper
!limit only)
eval ($low2c=12.3)        !lower(d_minus) limit (SLC restraints with upper
!limit only)
eval ($up2=4.0)           !upper(d_plus) limit (restraints with upper limit only)
eval ($low3=4.0)          !lower(d_minus) limit (restraints with both limits)

```

```

eval ($up3=4.0)      !upper(d_plus) limit (restraints with both limits)
eval ($a86=306)      !nr of restraints

!-----DO NOT CHANGE ANYTHING HERE-----

eval ($a48=$cpu*1e4)
set seed=$a48 end
eval ($a14=0)
eval ($a15=0)
eval ($a16=0)
eval ($a17=0)
eval ($a18=0)
eval ($a19=1)
eval ($a210=0)
eval ($a211=1)
eval ($nout1=0)      ! Number of structures output per run
eval ($nout=0)       ! Number of structures below violation cut-off per run
eval ($min1=9999.0)

!-----

set display=coord.dat end          !write parameters
display xx -----PARAMETERS-----
display xx startdate:      $DATE
display xx starttime:     $TIME
display xx ini.file:      $inifile
display xx pc_rota:       cent=(0 0 0) axis(0 1 0) -20.0
display xx pc_tran:       x+120.0 y+50.0 z-50.0 (update 30.12.2003)
display xx timestep (ps):  $a54
display xx nr of steps:    $a53   time /cycle (ps):  $a61
display xx nr of cycles:   $a05   total time (ps):   $a80
display xx time-spent fact: $a93   signpenalty fact:  $a57
display xx general scale:  $sqconst
display xx TBATH:         $a55   velocity factor:    $a56   fbeta:
$a58
display xx .pdb threshold  $a59
display xx .pdb lag factor  $a60
display xx Local mimim. impulse after 10 cycles with 'constant' Etot >
$a59a
display xx -----
display
set display=OUTPUT end

!-----SET CONDITIONS-----

flag exclude elec bond angl dihe impr include vdw noe end

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

set disp=ener.dat end
display Energies for the output structures
display file Etot VDW NOE
display -----
set disp=OUTPUT end

set disp=toten.dat end
display Energies for all cycles

```

```

display Cycle          Etot
display -----
set disp=OUTPUT end

!-----DEFINE RESTRAINTS-----

set echo=off end
set mess=off end

noe
  NRES=2000
end

@noeA.16ns.xpl          ! Defines restraints
@noeB.16ns.xpl
@noeC.16ns.xpl
@noeD.16ns.xpl

set echo=off end
set mess=off end

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

parameter

  @nbfix.4sl_expl.xpl

  nbonds
    cutnb=8.5
    inhi=0.25
    ctofnb=7.5
    ctonnb=6.5
    repel=0.6          !0.6
    NBXMod=-2
    rexp=2
    irex=2
    rcon=$a301
    wmin=1.5
  end

end

constraints
  interactions (segid="ccp") (segid="B")
end

energy end

while ($a01 LE $a05) loop calc

  display cycle $a01
  dynamics rigid
    dt=$a54
    group=(segid="B")
    dynmode=TCOU
    tbath=$a55
    nprint=50
    nstep=$a53

```

```

NTRFRQ=0 !new for XPLOR vs 3.8
end

eval ($a17=$a16)           !write .pdb structures
eval ($a16=$a15)           !write energy data of each structure
eval ($a15=$a14)
eval ($a14=$ENER)
eval ($abc=$VDW)

if ($a14<$a15) then
  if ($a14<$a16) then
    if ($a14<$a17) then
      if ($a14<$a59) then
        if ($abc<$van) then
          if ($a18>$a60) then
            eval ($nout1=$nout1+1)
            eval ($pdb="structure_"+encode($nout1)+".pdb")
            set disp=ener.dat end
            display $nout1 $a14    $VDW    $NOE
            set disp=OUTPUT end
            write coord output=$pdb end
            if ($a14<$min1) then
              eval ($ref=$nout1)
            end if
            eval ($min1=min($a14,$min1))
            pick bond (segid="ccp" and name FE) (segid="B" and name FE) geom
            eval ($a50=$RESULT)
            set disp=coord.dat end
            display Cycle: $a01    File#: $a19    Fe-Fe: $a50    temp: $TEMP
            display Etot: $a14      Evdw:    $VDW    Enoe:    $NOE
            display -----
            set disp=OUTPUT end

            eval ($a18=0)
            eval ($a19=$a19+1)
          end if
        end if
      end if
    end if
  end if
end if
eval ($a18=$a18+1)

set disp=toten.dat end
display $a01    $a14
set disp=OUTPUT end

if ($TEMP>200000.0) then           !correct excessive temperature

  eval ($a45=RAND()+0.01)
  vector do (vx=$a56*$a45*rand()) (segid="B")
  vector do (vy=$a56*$a45*rand()) (segid="B")
  vector do (vz=$a56*$a45*rand()) (segid="B")
end if

if ($a210=10) then                !impulse to escape local minimum
  if ($ENER>$a59a) then
    eval ($a212=$ENER/$a211)
    if ($a212>0.8) then

```

```

        if ($a212<2.0) then
            eval ($a213=100*(rand()-0.5) )
            eval ($a214=100*(rand()-0.5) )
            eval ($a215=100*(rand()-0.5) )
            vector do (x=x+$a213) (segid="B")
            vector do (y=y+$a214) (segid="B")
            vector do (z=z+$a215) (segid="B")
            vector do (vx=50) (segid="B")
            vector do (vy=50) (segid="B")
            vector do (vz=50) (segid="B")
            set disp=coord.dat end
            display cycle $a01: impulse: $a213, $a214, $a215 to x,y,z; v=50
            set disp=OUTPUT end
            display cycle $a01: impulse: $a213, $a214, $a215 to x,y,z; v=50
        end if
    end if
    eval ($a211=$ENER)
else
    eval ($a211=1)
end if
eval ($a210=0)
end if
eval ($a210=$a210+1)
eval ($a01 = $a01 + 1)

end loop calc

!-----FINISH-----
!----- Calculating rmsds -----

vector idend ( store9 ) ( name ca or name n or name c )      !backbone
                                                                !selection
eval ($ref_file="structure_"+encode($ref)+".pdb")

set display=rms.dat end
display Backbone pairwise RMSD from the lowest energy strucutre
($ref_file)
display file                      rmsd
display -----
set disp=OUTPUT end

    coord disp=comp @@$ref_file

eval ($count1=0)
while ($count1 < $nout1) loop fill

    evaluate ($count1=$count1+1)
        evaluate ($file="structure_"+encode($count1)+".pdb")
        coord init end
        coord @@$file
        coord sele=(recall 9) fit end
        coord sele=(recall 9) rms end
        eval ($b1 = $result)

set display=rms.dat end
display $file                      $b1

end loop fill

set display=OUTPUT end

```

```
!@viol_final.xpl          ! Writes viol.dat file  
  
set echo=true end  
  
stop
```