Facultad de Ciencias
Universidad Zaragoza

Departamento de Bioquímica y Biología Molecular y Celular

Proyecto Fin de Master:

Estudio computacional del mecanismo de unión-disociación de la flavodoxina de *Helicobacter pylori* y su ligando.

ANEXOS

Autor: Lic. Guillermo Gutiérrez Teixeira
Director de Proyecto: Dr. Javier Sancho Sanz
Curso 2012-2013

Estos scripts han sido desarrollados o modificados por mí a lo largo de la investigación. Dado que pertenecen a momentos distintos y el contexto de los scripts ha ido cambiando, lo más probable es que no se ejecuten bien directamente y requieran ciertas modificaciones mínimas para que sean funcionales.

1.Scripts:

Automation:

Config.cfg

```
#########################################################################
##                                                                     ##
##               Config file for automation scripts                    ##
##                  v0.6a by Guillermo Gutierrez                       ##
##               Universidad de Zaragoza, Feb 2013                     ##
##                                                                     ##
#########################################################################

export name="1FLV"
export parm="par_all27_prot_na_fmn.prm"
export cluster="MINOTAURO" # "TERMINUS" or "MEMENTO" or "HOME"
export deleterestarts="YES" # "YES" or anything else.
# the script will delete the restarts files after finishing a whole
stage
# only if $deleterestarts = "YES". any other value will make the
script compress
# the files with the best ratio possible
# id for atoms with a small constrain. this keeps the protein inside
the box
export spring_atoms="744 1739"
export path="$( pwd )"
export PATH=$PATH:$path:$path/scripts
# beware of special chars, dont use spaces in the file or directory
names
# and if you do so, be sure to replace the above command for this one
# path="$( pwd | sed -e 's/ /\\ /g' )
export mail="guillermo.munchkin@gmail.com"
# For debug only
export DEBUGMODE="ON"
export satom="phos"
# With a bit of luck, I won't need to change anything below this line
^^
export version="0.6.a"
if [ "$cluster" = "TERMINUS" ]; then
   export scheduler="PBS"
   export queue="JSS64"
   export parenv="ge.namd.8"
   export cores=16
elif [ "$cluster" = "MEMENTO" ]; then
   export scheduler="PBS"
```

```
   export queue="BIFIZCAM"
   export parenv="openmpi32"  # openmpi 8 or openmpi16, openmpi32,
openmpi64
   export cores=32            # 8 or 16, 32, 64 respectively. it must
match with parenv
elif [ "$cluster" = "MINOTAURO" ]; then
   export scheduler="SLURM"   # most of the configuration is already
set in the job.HEADER.SLURM file in templates/
elif [ "$cluster" = "MARENOSTRUM" ]; then
   export scheduler="LSF"
   export queue="class_a"
fi
```

killall.sh

```
#!/bin/bash
echo marcando para eliminación los siguientes trabajos:
qdel $( cat jid.txt )
kill jid.txt
```

total.sh

```
#!/bin/bash
   ###################################################################
   ##                                                               ##
   ##            Script creado por Guillermo Gutierrez              ##
   ##            Universidad de Zaragoza  Feb 2013                  ##
   ##                                                               ##
   ###################################################################

pid=0
source config.cfg
source scripts/functions.bash
clear
echo "Automatizacion por Guillermo Gutierrez"
echo "Universidad de Zaragoza, Feb 2013"
echo ""
echo "Ejecutando desde $path"
echo "Simulando: $name"
d_echo ""
d_echo "DEBUG MODE:  ON"
d_echo "Version:     $version"
d_echo "Cluster:     $place"
d_echo "Maquina:     $( uname -a )"
if [ ! "$place" = "MINOTAURO" ]; then d_echo "Cola:        $queue";
fi
if [ ! "$place" = "MINOTAURO" ]; then d_echo "Par_env:     $parenv
$cores"; fi
d_echo "Fecha:       $fecha"
d_echo "Path:        $path"
d_echo ""
```

```
if [ $place = "HOME" ]; then #para trabajar en casa...
   #esto no se puede hacer todo en el cluster, porque no tiene VMD
instalado.
   cd preparacion
   ./build.sh
   cd ..
   cp preparacion/output/ready* minimizacion/base/
else

   echo "se generara el archivo $path/jid.txt con los id de los tra-
bajos"
   echo "para matar todos los procesos ejecutar el script $path/kil-
lall.sh"
   echo ""
   if [ 1 = 1 ]; then   #switch to skip
      ###################################################################
      #                                                                 #
      #                  Minimizacion y equilibrado                     #
      #                                                                 #
      ###################################################################
      cd minimizacion
      echo "trabajando en $path/minimizacion"
      ##
      ## FIXED
      ##
      echo "Configurando job FIXED"
      $path/scripts/jobconfig.sh jobs/min.sh min_$name min.inp FIXED
0
      d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/min.sh
min_$name min.inp FIXED 0"
      d_echo "COMMAND: sendjob jobs/min.sh"
      pid=$( sendjob jobs/min.sh )
      d_echo "JID returned: $pid"
      echo $pid > $path/jid.txt
      ##
      ## DYNA
      ##
      echo "Configurando job DYNA"
      $path/scripts/jobconfig.sh jobs/dyna.sh dyna_$name pre-dyna-
cpt-equilibration.inp DYNA 0
      d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/dyna.sh
dyna_$name pre-dyna-cpt-equilibration.inp DYNA 0"
      d_echo "COMMAND: sendjob jobs/dyna.sh $pid"
      pid=$( sendjob jobs/dyna.sh $pid )
      d_echo "JID returned: $pid"
      echo $pid >> $path/jid.txt
      ##
```

```
      ## RESTRAIN
      ##
      for i in {1..10}; do
         echo "Configurando job RESTRAIN.$i"
         $path/scripts/jobconfig.sh jobs/rstr.$i.sh rst$i\_$name pre-
restraining.$i.inp RESTRAIN $i
         d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/rstr.$i.sh
rst$i\_$name pre-restraining.$i.inp RESTRAIN $i"
         d_echo "COMMAND: sendjob jobs/rstr.$i.sh $pid"
         pid=$( sendjob jobs/rstr.$i.sh $pid )
         d_echo "JID returned: $pid"
         echo $pid >> $path/jid.txt
      done
      ##
      ## UNRESTRAINED
      ##
      echo "Configurando job UNRESTRAINED"
      $path/scripts/jobconfig.sh jobs/unrstr.sh unr_$name pre-unre-
straining.inp UNRESTRAINED 11
      d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/unrstr.sh
unr_$name pre-unrestraining.inp UNRESTRAINED 11"
      d_echo "COMMAND: sendjob jobs/unrstr.sh $pid"
      pid=$( sendjob jobs/unrstr.sh $pid )
      d_echo "JID returned: $pid"
      echo $pid >> $path/jid.txt
      ##
      ## SLOW-HEATING
      ##
      echo "Configurando job HEATING"
      $path/scripts/jobconfig.sh jobs/heat.sh heat_$name pre-slow-
heating-cpt.inp HEATING 11
      d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/heat.sh
heat_$name pre-slow-heating-cpt.inp HEATING 11"
      d_echo "COMMAND: sendjob jobs/heat.sh $pid"
      pid=$( sendjob jobs/heat.sh $pid )
      d_echo "JID returned: $pid"
      echo $pid >> $path/jid.txt
      ##
      ## LGV
      ##
      echo "Configurando job LGV"
      $path/scripts/jobconfig.sh jobs/lgv.sh LGV_$name pre-LGV.inp
LGV 11
      d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/lgv.sh
LGV_$name pre-LGV.inp LGV 11"
      d_echo "COMMAND: sendjob jobs/lgv.sh $pid"
      pid=$( sendjob jobs/lgv.sh $pid )
```

```
      d_echo "JID returned: $pid"
      echo $pid >> $path/jid.txt

      ##
      ## Limpieza
      ##

      echo "Configurando la limpieza de los archivos de minimizacion"
      $path/scripts/jobconfig.sh jobs/clean.sh Clean_$name noinput
CLEAN_MINIM 0
      # saliendo de minimizacion...
      cd ..
   fi

   if [ 1 = 0 ]; then   #switch to skip

   ###################################################################
   #                                                                 #
   #                        Dinamica de 100ns                        #
   #                                                                 #
   ###################################################################

      cd produccion
      #Primera simulacion
      echo "Configurando la primera simulacion hasta los 2ns"
      $path/scripts/jobconfig.sh jobs/first.sh P1_$name first.inp
FIRST 1
      d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/first.sh
P1_$name first.inp FIRST 1"
      if [ $pid = 0 ]; then
         d_echo "COMMAND: sendjob jobs/first.sh"
         pid=$( sendjob jobs/first.sh )
      else
         d_echo "COMMAND: jobs/first.sh $pid"
         pid=$( sendjob jobs/first.sh $pid )
      fi
      d_echo "JID returned: $pid"
      if [ -e $path/jid.txt ]; then
         echo $pid >> $path/jid.txt
      else
         echo $pid > $path/jid.txt
      fi
      #Restarts
      for i in {2..50}; do
         echo "Configurando la simulacion nº $i, hasta los $( expr $i
\* 2 )ns"
         $path/scripts/jobconfig.sh jobs/restart.$i.sh R$i\_$name
```

```
restart.$i.inp RESTART $i
         d_echo "COMMAND: $path/scripts/jobconfig.sh jobs/restart.
$i.sh R$i\_$name restart.$i.inp RESTART $i"
         d_echo "COMMAND: sendjob jobs/restart.$i.sh $pid"
         pid=$( sendjob jobs/restart.$i.sh $pid )
         d_echo "JID returned: $pid"
         echo $pid >> $path/jid.txt
      done
      cd ..
      #Concatenacion

      echo "Configurando la concatenación de los archivos dcd de sa-
lida"
   fi
fi
```

scripts/clear.sh

```
#!/bin/bash
function delfiles {
  if [ -e $1 ]; then rm $1; fi
}
delfiles minimizacion/output/*
delfiles minimizacion/tmp/*
delfiles minimizacion/jobs/*
delfiles produccion/output/*
delfiles produccion/tmp/*
delfiles produccion/jobs/*
delfiles minimizacion/*.po*
delfiles minimizacion/*.o*
delfiles produccion/*.po*
delfiles produccion/*.o*
delfiles jid.txt
```

preparacion/build.sh

```
#!/bin/bash
#antes de nada limpiamos los posibles restos
./scripts/clear.sh
#primero preparamos pdb y psf para la proteina y para fmn
vmd -dispdev text -e  ./scripts/build_1FLV_apo.tcl
vmd -dispdev text -e  ./scripts/build_FMN.tcl
#fusionamos los archivos...
vmd -dispdev text -e  ./scripts/merge.tcl -args ./temp/built_1FLV_apo
./temp/built_FMN ./temp/built_1FLV_holo
#solvatamos
vmd -dispdev text -e  ./scripts/waterbox.tcl -args
./temp/built_1FLV_holo
```

```
#neutralizamos
vmd -dispdev text -e  ./scripts/ionization.tcl -args
./temp/built_1FLV_holo_wb
#y se obtiene la base de vectores
vmd -dispdev text -e  ./scripts/getboxparams.tcl -args
./temp/built_1FLV_holo_wb_neut
# ahora copiamos lo importante
cp ./temp/built_1FLV_holo_wb_neut.pdb ./output/ready_1FLV.pdb
cp ./temp/built_1FLV_holo_wb_neut.psf ./output/ready_1FLV.psf
cp ./temp/built_1FLV_holo_wb_neut.boxpar ./output/ready_1FLV.boxpar
```

preparacion/scripts/build_1FLV_apo.tcl

```
package require psfgen
# Read topology file
topology ./base/top_all27_prot_na_fmn.inp
#Aliasing special residues and atoms
pdbalias residue HIS HSE
pdbalias atom ILE CD1 CD
# Build protein segment
segment P1 {pdb ./base/1FLV_apo.pdb}
# Patch protein segment
#patch DISU P1:127 P1:139
#patch DISU P1:134 P1:152
#patch DISU P1:146 P1:163
#patch DISU P1:176 P1:188
#patch DISU P1:183 P1:201
#patch DISU P1:195 P1:210
# Read protein coordinates from PDB file
coordpdb ./base/1FLV_apo.pdb P1

# Build FMN segment
# segment CO {pdb FMN.pdb}
# Read protein coordinates from PDB file
# coordpdb FMN.pdb CO
# Guess missing coordinates
guesscoord
#  Write structure and coordinate files
writepdb ./temp/built_1FLV_apo.pdb
writepsf ./temp/built_1FLV_apo.psf
exit
```

preparacion/scripts/build_FMN.tcl

```
package require psfgen
# Read topology file
```

```
#topology top_all27_prot_na.rtf
topology ./base/top_all27_prot_na_fmn.inp
#Aliasing special residues and atoms
#pdbalias residue HIS HSE
#pdbalias atom ILE CD1 CD
# Build protein segment
#segment P1 {pdb 1FLVp.pdb}
# Patch protein segment
#patch DISU P1:127 P1:139
#patch DISU P1:134 P1:152
#patch DISU P1:146 P1:163
#patch DISU P1:176 P1:188
#patch DISU P1:183 P1:201

#patch DISU P1:195 P1:210
# Read protein coordinates from PDB file
#coordpdb 1FLVp.pdb P1

# Build FMN segment
segment CO {pdb ./base/FMN.pdb}
# Read protein coordinates from PDB file
coordpdb ./base/FMN.pdb CO
# Guess missing coordinates
guesscoord
#  Write structure and coordinate files
writepdb ./temp/built_FMN.pdb
writepsf ./temp/built_FMN.psf
exit
```

preparacion/scripts/clear.sh

```
#!/bin/bash
rm ./outputs/*
rm ./temp/*
```

preparacion/scripts/getboxparams.tcl

```
set fprefix [lindex $argv 0]
set outfile [open $fprefix.boxpar a]
mol load psf $fprefix.psf pdb $fprefix.pdb
set everyone [atomselect top all]
set xtrmc [measure minmax $everyone]
set com [measure center $everyone]
set cbv [vecsub [lindex $xtrmc 1] [lindex $xtrmc 0]]
puts $outfile "cellBasisVector1 [lindex $cbv 0] 0.0 0.0"
puts $outfile "cellBasisVector2 0.0 [lindex $cbv 1] 0.0"
puts $outfile "cellBasisVector3 0.0 0.0 [lindex $cbv 2]"
```

```
puts $outfile "cellOrigin $com"
exit
```

preparacion/scripts/ionization.tcl

```
package require autoionize
set command_line2 {}
set psffile [lindex $argv 0].psf
set pdbfile [lindex $argv 0].pdb
set outputf [lindex $argv 0]_neut
mol load psf $psffile pdb $pdbfile
set charge [eval "vecadd [[atomselect top all] get charge]"]
mol delete top
if {[expr round($charge)] != 0} {
  append command_line2 [concat "-psf" $psffile "-pdb" $pdbfile "-o"
$outputf "-seg ION"]

  if {$charge > 0} {
       set command_line2 [concat $command_line2 "-nna 0 -ncl [expr
round($charge)]"]
  } else {
       set command_line2 [concat $command_line2 "-ncl 0 -nna [expr -
1 * round($charge)]"]
  }
  eval autoionize $command_line2
  mol delete top
} else {
  puts "The system is already neutral"
  exec cp $psffile $outputf.psf
  exec cp $pdbfile $outputf.pdb
}
exit
```

preparacion/scripts/merge.tcl

```
# los argumentos que pilla son:
# la raiz para el par de archivos pdb/psf, es decir, si son foo-
bar.pdb y foobar.psf, foobar
# la raiz para el segundo par de archivos
# la raiz para los archivos generado por la fusion de todos los ante-
riores
package require readcharmmtop
package require psfgen
set m1 [lindex $argv 0]
set m2 [lindex $argv 1]
set o1 [lindex $argv 2]
```

```
set psfcon [psfcontext create]
psfcontext eval $psfcon {
     package require readcharmmtop
     topology ./base/top_all27_prot_na_fmn.inp
     readpsf $m1.psf
     readpsf $m2.psf
     coordpdb $m1.pdb
     coordpdb $m2.pdb
     writepdb $o1.pdb
     writepsf $o1.psf
}
psfcontext delete $psfcon
exit
```

preparacion/scripts/waterbox.tcl

```
package require solvate
set psffile [lindex $argv 0].psf
set pdbfile [lindex $argv 0].pdb
set outputf [lindex $argv 0]_wb
mol load psf $psffile pdb $pdbfile
solvate $psffile $pdbfile -rotate -t 7 -o $outputf
exit
```

scripts/functions.bash

```
function sendjob {
   k=0
   newpid=""
   sleep 1
   if [ "$2" = "" ]; then
      while [ "$newpid" = "" ] && [ $k -le 3 ]; do
         if [ "$scheduler" = "PBS" ]; then
            newpid=$( qsub -terse -R y $1 )
         elif [ "scheduler" = "SLURM" ]; then
            newpid=$( mnsubmit $1 | cut -d" " -f4 )
         fi
         k=$(( $k + 1 ))
         sleep 1
      done
      if [ "$newpid" = "" ]; then
         exit
      fi
      echo $newpid
   else
      while [ "$newpid" = "" ] && [ $k -le 3 ]; do
         newpid=$( qsub -terse -R y -hold_jid $2 $1 )
         k=$(( $k +1 ))
         sleep 1
```

```
      done
      if [ "$newpid" = "" ]; then
         exit
      fi
      echo $newpid
   fi
}

function d_echo {
   if [ "$DEBUGMODE" = "ON" ]; then echo $1; fi
}

function fecha {
   echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )]"
}

function swc {
   if [ "$cluster" = "MINOTAURO" ]; then
      echo "$1:00"
   elif [ "$cluster" = "MARENOSTRUM" ]; then
      echo "$1"
   else
      echo ""
   fi
}
```

scripts/generate_smd_initial_files.pl

```
#!/usr/bin/perl -w -I /opt/local/lib/perl5/site_perl/5.12.4
# script producido por Vladimir Espinosa y modificado por Guillermo
Gutiérrez

use Getopt::Long;# Loading the Getopt module for handling command
line options used by this program
use Pod::Usage;# Loading the POD module to handle HELP and MAN re-
ports about the program
use Math::Random;#For generating random numbers
use constant RANDOM_FRAMES => 10;#Defining the number of random
frames to be chosen from the trajectory
use constant OUTPUT_FREQ => 5000;#Defining the number of random
frames to be chosen from the trajectory.!!!!!PLEASE UPDATE THIS IF
THE FREQUENCY CHANGES!!!!!

my $man = 0;# Setting to (false) the MAN display option
my $help = 0;# Setting to (false) the HELP display option
my $plateau_start = '';
```

```
my $plateau_end = '';
my $smd_atom = '';
my $PSF_PATH = '';
my $XST_PATH = '';
my $VEL_PATH = '';
my $DCD_PATH = '';
my $CATDCD = '/data1/NAMD_2.6_Linux-amd64/catdcd';#This is the PATH
to CATDCD on my system. Please edit according to your
configuration!!!!!
# Necessary step to find out whether some command line options are
actually passed to the program
# because the @ARGV is emptied when parsed by Getopt::Long!!!!
my @command_rescue = @ARGV;
## Parse options and print usage if there is a syntax error or if us-
age was explicitly requested
GetOptions(    "help|?" => \$help,
                     "man" => \$man,
            "initial-frame=i" => \$plateau_start,
            "final-frame=i" => \$plateau_end,
            "smd-atom=s" => \$smd_atom,
                   "psf-file=s" => \$PSF_PATH,
                   "xst-file=s" => \$XST_PATH,
                   "vel-file=s" => \$VEL_PATH,
            "dcd-file=s" => \$DCD_PATH) or pod2usage(2);
pod2usage(1) if $help;# To print the HELP message upon requestion
(i.e condensed format)
pod2usage('-verbose' => 2) if $man;# to print the MAN message upon
requiestion (i.e "verbose" format)
## If no arguments were given, then allow STDIN to be used only
## if it's not connected to a terminal (otherwise print usage)
pod2usage("$0: No files given.")  if ((@command_rescue == 0) && (-t
STDIN));

my $OUTPUT_DIR = '.';# Predefined PATH to the OUTPUT files directory.
Edit according to your preferences

#### Checking for the existence of the PSF and DCD files to be used
as well as for the correctness of the residue-selection and eval-
weight strings

die "Not a valid PATH to the trajectory files!!!!!\n" unless ((-e
"$PSF_PATH") and (-e "$XST_PATH") and (-e "$VEL_PATH") and (-e
"$DCD_PATH"));

#### Checking for the existence of the integers corresponding to the
start and end of the plateau from which we are going to randomly se-
lect the snapshots
```

```
#die "Not a valid PATH to the trajectory files!!!!!\n" unless ((de-
fined $plateau_start) and (defined $plateau_end));

my @seed = random_get_seed ();
random_set_seed (@seed);
my @random_indexes = random_uniform_integer (RANDOM_FRAMES,
$plateau_start, $plateau_end);

#### Loading the (get_vectors) procedure and printing the command
used to make the RMSD calculations
open (TCL_FILE, ">calculate_vectors.tcl");
print TCL_FILE "source get_smd_vectors.tcl\n";
open (XST_FILE, $XST_PATH);
my @xst_lines = <XST_FILE>;
chomp (@xst_lines);
foreach my $index (@random_indexes)
{
    print TCL_FILE "get_vectors top $index $smd_atom\n";
    system ("$CATDCD -o $OUTPUT_DIR/1FLV.$smd_atom-restart.$index.-
coor -otype namdbin -first $index -last $index $DCD_PATH");
    system ("$CATDCD -o $OUTPUT_DIR/1FLV.$smd_atom-restart.$index.vel
-otype namdbin -first $index -last $index $VEL_PATH");
    open (XSC_FILE, ">$OUTPUT_DIR/1FLV.$smd_atom-restart.
$index.xsc");
    #Now printing the header of the XSC file
    print XSC_FILE "# NAMD extended system trajectory file\n#\$LABELS
step a_x a_y a_z b_x b_y b_z c_x c_y c_z o_x o_y o_z s_x s_y s_z s_u
s_v s_w\n";
    my $xsc_frequency = OUTPUT_FREQ * $index;#Setting the correct
frequency for the current frame. !!!!!PLEASE UPDATE THIS IF THE FRE-
QUENCY CHANGES!!!!!
    my @return = grep /^$xsc_frequency/, @xst_lines;
    #In case there are multiple lines for the same frequency (due to
the system hanged up between consecutive restart writing cycles) we
need to return
    #the last ocurrence of the periodic cell description as always
the restart is done with the last printed restart files
    my $xsc_line = pop (@return);
    print XSC_FILE "$xsc_line\n";
    close (XSC_FILE);
}

close (TCL_FILE);
close (XST_FILE);

#### Opening the VMD console interface to process the .tcl temporal
instructions file generated above
```

```
system ("vmd -eofexit -dispdev text -f $PSF_PATH $DCD_PATH < calcu-
late_vectors.tcl");

##########################################################################
#########################################################
#### Section of POD language to be used for printing a MAN page with
general informations about the usage of the program####
##########################################################################
#########################################################

__END__

=head1 NAME

generate_smd_initial_files.pl - An input-handling Perl script to gen-
erate all the files needed to start a SMD trajectory. It reads the
trajectory files
from a long trajectory and ramdomly choses a group of snapshots to be
used as starting points for parallel SMD trajectories.

=head1 SYNOPSIS

generate_smd_initial_files.pl [I<OPTIONS>]
A simple perl script to read a trajectory and randomly chosing a pre-
defined set of snapshots (preferably in the plateau zone).

It also invokes a TCL script that calculates 7 pulling vectors to
pull the FMN out of the binding cage from atom C8M (the extension
for the Phosphate is lacking). It takes as arguments the name of the
initial and final frames withing which the snapshots are going
to be chosen, the name of the SMD atom and the PATHs to the PSF, XST,
VEL and DCD files respectively.
FOR HELP USING THIS PROGRAM PLEASE TRY:
./generate_smd_initial_files.pl --help

FOR A COMPLETE DESCRIPTION OF THE PROGRAM TRY: ./generate_smd_ini-
tial_files.pl --man
=head1 OPTIONS
=over 3
=item B<-help>:
     Print a brief help message and exits (CONDITIONAL).
=item B<-man>:
     Prints the manual page and exits (CONDITIONAL).
=item B<-initial-frame>:
     Integer corresponding to the START frame of the window (plateau
zone) from which snapshots are going to be ramdomly selected (MANDA-
TORY).
```

```
=item B<-final-frame>:
     Integer corresponding to the END frame of the window (plateau
zone) from which snapshots are going to be ramdomly selected (MANDA-
TORY).
=item B<-smd-atom>:
     A character or string to differentiate the files to be generated
for different pulling atoms (MANDATORY).
=item B<-psf-file>:
     PATH to the parameters (i.e. PSF, PARM) file of the simulation
(MANDATORY).
=item B<-xst-file>:
     PATH to the Periodic Conditions file of the simulation (MANDA-
TORY).
=item B<-vel-file>:
     PATH to the Velocities file of the simulation (MANDATORY).
=item B<-dcd-file>:
     PATH to the Trajectory file of the simulation (MANDATORY).
=back
=head1 DESCRIPTION
This program takes seven mandatory arguments from the command line
(i.e. initial and final frames within which the selection is to be
done, a name to be used to differentiate files for different pulling
atoms, and the PATHs to the PSF, XST, VEL and DCD files from a tra-
jectory). So a common way of using this program would be as follows:
=over 3
=item B<Example:>
     ./generate_smd_initial_files.pl --initial-frame=300 --final-
frame=400 --smd-atom=Met --psf-file=PSF_PATH --xst-file=XST_PATH
--vel-file=VEL_PATH --dcd-file=DCD_PATH
=back
The numbers in --initial-frame, and --final-frame should be previ-
ously chosen carefully selecting a zone in the initial simulation
which is structurally estable -i.e. a plateau in a RMSD plot-. For
the --smd-atom
option any string or character that serve to differentiate the atom
which is going to be used for pulling will do -e.g. [Met|M|Metilo]
for C8M or [Pho|P|Fosfato] for the tail Phosphate-. The extension to
calculate the pulling
vector for the Phosphate has not been included yet.
=head1 AUTHOR
Vladimir Espinosa Angarica: March the 21th, 2013. University of
Zaragoza, Spain
=cut
```

scripts/get_smd_data.pl

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

```
open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = split(" ",<FILE>);
#print "($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1)";
$ts0=$ts1;
$px0=$px1;
$py0=$py1;
$pz0=$pz1;
$fx0=$fx1;
$fy0=$fy1;
$fz0=$fz1;
$wacum=0;
$dtot=0;
$mf=0;
print RESULT "TSTEP dTot Wacum ModF\n";
while (<FILE>) {
chomp;
($ts2, $px2, $py2, $pz2, $fx2, $fy2, $fz2) = split(" ");
#$dx= $px2-$px1;
#print "$dx\n";
#$wtemp=(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+($fz2+
$fz1)*($pz2-$pz1))/2;
#print "$wtemp\n";
$wacum=$wacum + (($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+
($fz2+$fz1)*($pz2-$pz1))/2;
$dtot=sqrt(($px2-$px0)**2 + ($py2-$py0)**2 + ($pz2-$pz0)**2);
$mf=sqrt(($fx2-$fx1)**2 + ($fy2-$fy1)**2 + ($fz2-$fz1)**2);
print RESULT "$ts2 $dtot $wacum $mf\n";
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = ($ts2, $px2, $py2, $pz2,
$fx2, $fy2, $fz2);
}
close (FILE);
close (RESULT);
```

scripts/inpconfig.sh

```
#!/bin/bash

   ##################################################################
   ##                                                              ##
   ##            Script creado por Guillermo Gutierrez             ##
   ##            Universidad de Zaragoza  Feb 2013                 ##
   ##                                                              ##
   ##################################################################

type=$1
index=$2
```

```
index=$(( $index ))
if [ $type = "FIXED" ]; then
   if [ -e "$tmp/fixed-atoms.pdb" ]; then
      rm tmp/fixed-atoms.pdb
   fi
   ## Generating the template of the fixed-atoms.pdb file
   cp base/ready_$name.pdb tmp/fixed-atoms.pdb
   ## Setting the B column of the protein atoms to 1
   sed -i -e "s/0\.00\( *\)P1/1\.00\1P1/" tmp/fixed-atoms.pdb
   ## Setting the B column of the cofactor atoms to 1
   sed -i -e "s/0\.00\( *\)CO/1\.00\1CO/" tmp/fixed-atoms.pdb
   ## Editing the minimization.inp file
   if [ -e minimizacion/tmp/min.inp ]; then
      rm minimizacion/tmp/min.inp
   fi
   cat ../templates/min1.template base/ready_$name.boxpar ../tem-
plates/min2.template > tmp/min.inp
   sed -i -e "s/NAME1/$name/" tmp/min.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/min.inp
   sed -i -e "s/PARM1/$parm/" tmp/min.inp
elif [ $type = "DYNA" ]; then
   if [ -e "tmp/fixed-atoms.pdb" ]; then
      rm tmp/fixed-atoms.pdb
   fi
   cp output/$name-pre-min.coor tmp/fixed-atoms.pdb
   ## Setting the B column of the protein atoms to 1
   sed -i -e "s/0\.00\( *\)P1/1\.00\1P1/" tmp/fixed-atoms.pdb
   ## Setting the B column of the cofactor atoms to 1
   sed -i -e "s/0\.00\( *\)CO/1\.00\1CO/" tmp/fixed-atoms.pdb
   ## Editing the pre-dyna-cpt.inp file

   cp ../templates/pre-dyna-cpt.template tmp/pre-dyna-cpt-equilibra-
tion.inp
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/" tmp/pre-
dyna-cpt-equilibration.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/pre-dyna-cpt-equilibration.inp
   sed -i -e "s/PARM1/$parm/" tmp/pre-dyna-cpt-equilibration.inp
elif [ $type = "RESTRAIN" ]; then
   if [ -e "tmp/reference-atoms.pdb" ]; then
      rm tmp/reference-atoms.pdb
   fi
   if [ -e "tmp/spring-atoms.pdb" ]; then
      rm tmp/spring-atoms.pdb
   fi
   if [ $index -eq 1 ]; then
```

```
       ## Generating the reference-atoms.pdb and spring-atoms.pdb
file used in this step
       cp "output/$name-dyna-cpt-equilibrated.coor" tmp/reference-
atoms.pdb
       cp "output/$name-dyna-cpt-equilibrated.coor" tmp/spring-atom-
s.pdb
   else
       ## Using pdb from previous step
       previous=$(( $index - 1 ))
       cp "output/$name-pre-restrained.$previous.coor" tmp/reference-
atoms.pdb
       cp "output/$name-pre-restrained.$previous.coor" tmp/spring-
atoms.pdb
   fi
   ## Setting the B column of the protein atoms to 1
   sed -i -e "s/0\.00\( *\)P1/1\.00\1P1/" tmp/reference-atoms.pdb
spring-atoms.pdb
   ## Setting the B column of the cofactor atoms to 1
   sed -i -e "s/0\.00\( *\)CO/1\.00\1CO/" tmp/reference-atoms.pdb
spring-atoms.pdb
   #Setting the (pre-restraining.inp file) for this specific loop-
step
   cp ../templates/pre-restraining.template tmp/pre-restraining.$in-
dex.inp
   ## Editing the pre-restraining.$index.inp file
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/" tmp/pre-
restraining.$index.inp
   sed -i -e "s/set\( *\)index\( *\)I/set\1index\2$index/" tmp/pre-
restraining.$index.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/pre-restraining.$index.inp
   if [ $index -eq 1 ]; then
       sed -i -e "s/COOR1/$name-dyna-cpt-equilibrated.coor/" tmp/pre-
restraining.$index.inp
       sed -i -e "s/XSC1/$name-dyna-cpt-equilibrated.xsc/" tmp/pre-
restraining.$index.inp
   else
       ## Using pdb from previous step
       previous=$(( $index - 1 ))
       sed -i -e "s/COOR1/$name-pre-restrained.$previous.coor/"
tmp/pre-restraining.$index.inp
       sed -i -e "s/XSC1/$name-pre-restrained.$previous.xsc/"
tmp/pre-restraining.$index.inp
   fi
   #Setting the damping coefficient used to slowly turn down the re-
straint of the protein
   K=$(( 22 - $index*2 ))
```

```
   sed -i -e "s/constraintScaling\
( *\)1\.0/constraintScaling\1$K\.0/" tmp/pre-restraining.$index.inp
   sed -i -e "s/PARM1/$parm/" tmp/pre-restraining.$index.inp
elif [ $type = "UNRESTRAINED" ]; then
   ## Editing the pre-unrestraining.inp file
   cp ../templates/pre-unrestraining.template tmp/pre-unrestrain-
ing.inp
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/" tmp/pre-
unrestraining.inp
   previous=$(( $index - 1 ))
   sed -i -e "s/set\( *\)index\( *\)I/set\1index\2$previous/"
tmp/pre-unrestraining.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/pre-unrestraining.inp
   sed -i -e "s/PARM1/$parm/" tmp/pre-unrestraining.inp
elif [ $type = "HEATING" ]; then
   cp ../templates/pre-slow-heating-cpt.template tmp/pre-slow-heat-
ing-cpt.inp
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/" tmp/pre-
slow-heating-cpt.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/pre-slow-heating-cpt.inp
   sed -i -e "s/PARM1/$parm/" tmp/pre-slow-heating-cpt.inp
elif [ $type = "LGV" ]; then
   if [ -e "tmp/reference-atoms-lgv.pdb" ]; then
       rm tmp/reference-atoms-lgv.pdb
   fi
   if [ -e "tmp/spring-atoms-lgv.pdb" ]; then
       rm tmp/spring-atoms-lgv.pdb
   fi
   ## We need to generate the reference-atoms-lgv.pdb and spring-
atoms-lgv.pdb file used in this step
   ## Final PDB file generated in the minimization
   cp "output/$name-pre-slow-heated-cpt.coor" tmp/reference-atoms-
lgv.pdb

   cp "output/$name-pre-slow-heated-cpt.coor" tmp/spring-atoms-
lgv.pdb
   ## We need to generate now the LGV-coef.pdb file with the Langevin
coefficients used in this step
   if [ -e "tmp/LGV-coef.pdb" ]; then
       rm tmp/LGV-coef.pdb
   fi
   cp "output/$name-pre-slow-heated-cpt.coor" tmp/LGV-coef.pdb
   ## Now Setting the Langevin coefficient of the aminoacid atoms
   sed -i -e "s/0\.00\( *\)P1/0\.50\1P1/" tmp/LGV-coef.pdb
   ## And Now the coefficients of the water
```

```
   sed -i -e "s/0\.00\( *\)WT1/60\.00\1WT1/" tmp/LGV-coef.pdb
   ## And Now the coefficients of the counterions
   sed -i -e "s/0\.00\( *\)ION/60\.00\1ION/" tmp/LGV-coef.pdb
   ## And Now the coefficient of the coordinated chloro
   sed -i -e "s/0\.00\( *\)CO/0\.50\1CO/" tmp/LGV-coef.pdb
   ## Editing the pre-LGV.inp file to set the index name of the muta-
tion
   cp ../templates/pre-LGV.template tmp/pre-LGV.inp
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/" tmp/pre-
LGV.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/pre-LGV.inp
   sed -i -e "s/PARM1/$parm/" tmp/pre-LGV.inp
elif [ $type = "FIRST" ];   then
   ## Copy the files from minimization
   cp ../minimizacion/base/ready_$name.psf base/
   cp ../minimizacion/base/par_all27_prot_na_fmn.prm base/
   cp ../minimizacion/output/$name-pre-lgv.* base/
   #preparando los archivos para dinamica de langevin
   if [ -e "tmp/reference-atoms-lgv.pdb" ]; then
      rm tmp/reference-atoms-lgv.pdb
   fi
   if [ -e "tmp/spring-atoms-lgv.pdb" ]; then
      rm tmp/spring-atoms-lgv.pdb
   fi
   ## Generating the reference-atoms-lgv.pdb and spring-atoms-lgv.pdb
   cp "base/$name-pre-lgv.coor" tmp/reference-atoms-lgv.pdb
   cp "base/$name-pre-lgv.coor" tmp/spring-atoms-lgv.pdb
   ## Generatin the LGV-coef.pdb file
   if [ -e "tmp/LGV-coef.pdb" ]; then
      rm tmp/LGV-coef.pdb
   fi
   cp "base/$name-pre-lgv.coor" tmp/LGV-coef.pdb
   ## Now Setting the Langevin coefficient of the aminoacid atoms
   sed -i -e "s/0\.00\( *\)P1/0\.50\1P1/" tmp/LGV-coef.pdb
   ## And Now the coefficients of the water
   sed -i -e "s/0\.00\( *\)WT1/60\.00\1WT1/" tmp/LGV-coef.pdb
   ## And Now the coefficients of the counterions
   sed -i -e "s/0\.00\( *\)ION/60\.00\1ION/" tmp/LGV-coef.pdb
   ## And Now the coefficient of the coordinated FMN
   sed -i -e "s/0\.00\( *\)CO/0\.50\1CO/" tmp/LGV-coef.pdb
   ## Setting Spring and Reference
   for i in $( echo $spring_atoms ); do
      linea=$( cat base/$name-pre-lgv.coor | grep "ATOM\( \)*$i " )
      sustitucion=$( echo $linea | sed -e "s/0.00/1.00/" )
      sed -i -e "s/$linea/$sustitucion/" tmp/reference-atoms-lgv.pdb
      sustitucion=$( echo $linea | sed -e "s/0.00/0.01/" )
```

```
      sed -i -e "s/$linea/$sustitucion/" tmp/spring-atoms-lgv.pdb
   done

   ## Copy the template
   cp ../templates/first.template tmp/first.inp
   ## Editing the first.inp file to set the index name, path, and
other parameters.
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/"
tmp/first.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/first.inp
   sed -i -e "s/PARM1/$parm/" tmp/first.inp
elif [ $type = "RESTART" ]; then
   ## Setting files for LGV dynamic...
   if [ -e "tmp/reference-atoms-lgv.pdb" ]; then
      rm tmp/reference-atoms-lgv.pdb
   fi
   if [ -e "tmp/spring-atoms-lgv.pdb" ]; then
      rm tmp/spring-atoms-lgv.pdb
   fi
   cp "base/$name-pre-lgv.coor" tmp/reference-atoms-lgv.pdb
   cp "base/$name-pre-lgv.coor" tmp/spring-atoms-lgv.pdb
   if [ -e "tmp/LGV-coef.pdb" ]; then
      rm tmp/LGV-coef.pdb
   fi
   cp "base/$name-pre-lgv.coor" tmp/LGV-coef.pdb
   ## Now Setting the Langevin coefficient of the aminoacid atoms
   sed -i -e "s/0\.00\( *\)P1/0\.50\1P1/" tmp/LGV-coef.pdb
   ## And Now the coefficients of the water
   sed -i -e "s/0\.00\( *\)WT1/60\.00\1WT1/" tmp/LGV-coef.pdb
   ## And Now the coefficients of the counterions
   sed -i -e "s/0\.00\( *\)ION/60\.00\1ION/" tmp/LGV-coef.pdb
   ## And Now the coefficient of the coordinated FMN
   sed -i -e "s/0\.00\( *\)CO/0\.50\1CO/" tmp/LGV-coef.pdb
   ## Copy the template
   cp ../templates/restart.template tmp/restart.$index.inp
   ## Editing the first.inp file to set the index name, path, and
other parameters.
   sed -i -e "s/set\( *\)name\( *\)NAME/set\1name\2$name/"
tmp/restart.$index.inp
   sed -i -e "s/PATH1/$( echo $( pwd ) | sed -e 's/\//\\\//g')/"
tmp/restart.$index.inp
   i=$(( $index - 1 ))
   i=$(( $i * 500 ))
   sed -i -e "s/RESTARTPOINT1/$i/" tmp/restart.$index.inp
   i=$(( $i + 500 ))
   sed -i -e "s/TSTEPS1/$i/" tmp/restart.$index.inp
```

```
   sed -i -e "s/INDEX1/$index/" tmp/restart.$index.inp
   sed -i -e "s/PARM1/$parm/" tmp/restart.$index.inp
else
   echo "Your arguments are invalid : )"

fi
```

scripts/jobconfig.sh

```
#!/bin/bash

   ################################################################
   ##                                                            ##
   ##            Script creado por Guillermo Gutierrez           ##
   ##            Universidad de Zaragoza  Feb 2013               ##
   ##                                                            ##
   ################################################################

#configura la plantilla job.template para un uso concreto
#uso:
#    jobconfig.sh [path/]nombre_del_script.sh Nombre_trabajo inputfi-
le {FIXED|DYNA|RESTRAIN|UNRESTRAINED|HEAT|LGV|CLEAN|CONCAT} INDEX

if [ -e $1 ]; then # si el archivo existe lo borra
     rm $1
fi
#cp $path/templates/job.$place $1 #copia la plantilla...
if [ "$4" = "CLEAN" ]; then
   cat $path/templates/job.HEADER.$scheduler.serial
$path/templates/job.CLEAN $path/templates/job.TAIL > $1
elif [ "$4" = "CONCAT" ]; then
   cat $path/templates/job.HEADER.$scheduler.serial
$path/templates/job.CONCAT $path/templates/job.TAIL > $1
else
   cat $path/templates/job.HEADER.$scheduler $path/templates/job.
$cluster $path/templates/job.TAIL > $1
   sed -i -e "s/CONFIG1/$( echo $path/scripts/inpconfig.sh $4 $5 |
sed -e 's/\//\\\//g' )/" "$1"
   sed -i -e "s/INPFILE1/\"$3\"/" "$1" #donde pone INPFILE pone el
nombre del archivo de input
   sed -i -e "s/PARENV/$parenv/" "$1"
   sed -i -e "s/CORES/$cores/" "$1"
fi
sed -i -e "s/QUEUE/$queue/" "$1"
sed -i -e "s/MAIL/$mail/" "$1"
sed -i -e "s/NAME1/$2/" "$1"
```

scripts/setmininp.sh

```
#!/bin/bash
if [ -e $1/minimizacion/tmp/min.inp ]; then
   rm $1/minimizacion/tmp/min.inp
fi
cat $1/templates/min1.template > $1/minimizacion/tmp/min.inp
cat $1/minimizacion/base/ready_1FLV.boxpar >>
$1/minimizacion/tmp/min.inp
cat $1/templates/min2.template >> $1/minimizacion/tmp/min.inp
```

scripts/smd_analisis.sh

```
#!/bin/bash
#para el analisis de las fuerzas, necesitamos conocer la elongacion y
la fuerza en cada momento.
#el log de salida de NAMD da posicion y fuerza, en lugar de elonga-
cion y fuerza.
#queremos un archivo en el que tengamos lo siguiente
#TS  D0   F0   D1   F1   D2   F2 ..... D9    F9
#para ello vamos a crear un archivo para cada vindex.rindex con TS D
F

for $vindex in {0..6};do
   for $rindex in {0..9};do
      bname=$name.$satom.$vindex.$rindex
      #extraemos coordenadas iniciales
      #initcoor=$( echo $( cat fixed_and_smd_atoms.pdb | grep "ATOM\(
\)*2591" ) | cut -d" " -f7-9 )
      #initx=$( echo $initcoor | cut -d" " -f1 )
      #inity=$( echo $initcoor | cut -d" " -f2 )
      #initz=$( echo $initcoor | cut -d" " -f3 )
      #se puede hacer del propio log:
      cat output/logs/$bname.inp.log | grep "SMD  " > tmp/$bname.smd
      cut -d" " -f3 tmp/$bname.smd > tmp/TSTEPS
      cut -d" " -f4 tmp/$bname.smd > tmp/$bname.px
      cut -d" " -f5 tmp/$bname.smd > tmp/$bname.py
      cut -d" " -f6 tmp/$bname.smd > tmp/$bname.pz
      cut -d" " -f7 tmp/$bname.smd > tmp/$bname.fx
      cut -d" " -f8 tmp/$bname.smd > tmp/$bname.fy
      cut -d" " -f9 tmp/$bname.smd > tmp/$bname.fz
      initx=$( head tmp/$bname.px -n 1 )
      inity=$( head tmp/$bname.py -n 1 )
      initz=$( head tmp/$bname.pz -n 1 )
      echo "calculando el desplazamiento en x..."
      for i in $( cat tmp/$bname.px ); do
         echo "($i - $initx)^2" | bc >> tmp/$bname.dx2
```

```
      done
      echo "calculando el desplazamiento en y..."
      for i in $( cat tmp/$bname.py ); do
         echo "($i - $inity)^2" | bc >> tmp/$bname.dy2
      done
      echo "calculando el desplazamiento en z..."
      for i in $( cat tmp/$bname.pz ); do
         echo "($i - $initz)^2" | bc >> tmp/$bname.dz2
      done
      paste -d" " tmp/$bname.dx2 tmp/$bname.dy2 tmp/$bname.dz2 > tmp/
$bname.dxyz2
      k=0
      h=""
      echo "calculando el modulo del desplazamiento..."
      for i in $( cat tmp/$bname.dxyz2 ); do
         h="$h $i"
         k=$(( $k + 1 ))
         if [ $k = 3 ]; then
            x2=$( echo $h | cut -d" " -f1 )
            y2=$( echo $h | cut -d" " -f2 )
            z2=$( echo $h | cut -d" " -f3 )
            echo "sqrt($x2 + $y2 + $z2)" | bc >> tmp/$bname.d
            h=""
            k=0
         fi
      done
      paste -d" " tmp/$bname.fx tmp/$bname.fy tmp/$bname.fz > tmp/
$bname.fxyz
      echo "calculando el modulo de la fuerza..."
      for i in $( cat tmp/$bname.fxyz ); do
         h="$h $i"
         k=$(( $k + 1 ))
         if [ $k = 3 ]; then
            x=$( echo $h | cut -d" " -f1 )
            y=$( echo $h | cut -d" " -f2 )
            z=$( echo $h | cut -d" " -f3 )
            echo "sqrt($x^2 + $y^2 + $z^2)" | bc >> tmp/$bname.f
            h=""
            k=0
         fi
      done
   done
   ls tmp/$name.$satom.$vindex.*.d > tmp/$name.$satom.$vindex.T.d
   ls tmp/$name.$satom.$vindex.*.f > tmp/$name.$satom.$vindex.T.f
   paste -d" " tmp/$name.$satom.$vindex.T.d tmp/$name.$satom.$vin-
dex.T.f > tmp/$name.$satom.$vindex.T.df
   t=$( cat tmp/$name.$satom.$vindex.T.df )
```

```
   paste -d" " tmp/TSTEPS $t > output/RESULTS/$name.$satom.$vin-
dex.res
done
```

scripts/spread_vectors.sh

```
#!/bin/bash
function spread_vectors {
k=0
j=0
for rindex in {0..9}; do
   for i in $( cat base/$name.$satom.$rindex.vectors ); do
      echo $i >> base/$name.$satom.$rindex.$k.vec
      j=$(( 1+ $j ))
      if [ $j = 3 ]; then
         k=$(( 1+$k ))
         j=0
      fi
   done
done
}
```

SMD/scripts/configjob.sh

```
#!/bin/bash

# generacion de produccion
$steps=16000000
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail -n 1 )
    cp templates/template.job jobs/1FLV.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/1FLV.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/1FLV.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/1FLV.$g.$i.$j.inp/" jobs/1FLV.$g.$i.
$j.sh
    cp templates/smd_general.inp inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/NAME1/1FLV/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/1FLV.$g.$i.$j.inp
```

```
    sed -i -e "s/SVEL1/$svel/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/1FLV.$g.$i.$j.inp
  done
done
g=met
j=0
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail -n 1 )
    cp templates/template.job jobs/1FLV.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/1FLV.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/1FLV.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/1FLV.$g.$i.$j.inp/" jobs/1FLV.$g.$i.
$j.sh
    cp templates/smd_general.inp inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/NAME1/1FLV/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/1FLV.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/1FLV.$g.$i.$j.inp
  done
done
```

SMD/scripts/configjobs.vs.sh

```
#!/bin/bash

# generacion de produccion
vmult=1
for n in $( echo "1FLV slow.1FLV veryslow.1FLV" ); do
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
```

```
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_general.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
  done
done
g=met
j=0
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail -n 1 )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_general.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
  done
done
vmult=$( echo "$vmult * 0.50" | bc )
done
```

SMD/scripts/configjobs.reloaded.sh

```
#!/bin/bash
# generacion de produccion
vmult=10
steps=1550000
jname=""
for i in {0..6}; do
```

```
  jname="$jname dubstep.reloaded.short.$i.1FLV"
done
jtemplate="templates/template_32.job"
inptemplate="templates/reloaded.inp"
for n in $( echo $jname ); do
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp $jtemplate jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp $inptemplate inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME2/1FLV/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
g=met
j=0
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp $jtemplate jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp $inptemplate inps/$n.$g.$i.$j.inp
```

```
    sed -i -e "s/NAME2/1FLV/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
done
```

SMD/scripts/configjobs.fast.sh

```
#!/bin/bash
# generacion de produccion
steps=1500000
vmult=10
for n in $( echo "fast.1FLV" ); do
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_general.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
g=met
j=0
```

```
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail -n 1 )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_general.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
vmult=$( echo "$vmult * 0.50" | bc )
done
```

SMD/scripts/configjobs.continue.sh

```
#!/bin/bash

# generacion de produccion
vmult=10
#para media...
steps=700000
fstep=1250000
jname=""
for i in {0..6}; do
  jname="$jname dubstep.continue.med.$i.1FLV"
done
jtemplate="templates/template_16.job"
inptemplate="templates/smd_continue.inp"
for n in $( echo $jname ); do
g=phos
j=0
echo $n.$g
for j in {0..7}; do
  for i in {0..10}; do
```

```
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cname="$( echo $n | sed -e 's/continue\.med/fast/').$g.$i.$j"
    cp $jtemplate jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp $inptemplate inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME3/$cname/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME2/1FLV/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/FIRSTSTEP1/$fstep/" inps/$n.$g.$i.$j.inp
  done
done
g=met
j=0
echo $n.$g
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cname="$( echo $n | sed -e 's/continue\.med/fast/').$g.$i.$j"
    cp $jtemplate jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp $inptemplate inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME3/$cname/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME2/1FLV/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
```

```
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/FIRSTSTEP1/$fstep/" inps/$n.$g.$i.$j.inp
  done
done
done
#for i in $( ls jobs/dubstep* ); do
#  qsub $i
#done
```

SMD/scripts/configjobs.doublestep.fast.sh

```
#!/bin/bash
# generacion de produccion
vmult=10
steps=1250000
for n in $( echo "dubstep.fast.3.1FLV" ); do
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_doublestep.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME2/1FLV/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
g=met
```

```
j=0
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_doublestep.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME2/1FLV/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
done
```

analisis/base/config.cfg

```
datadir="/media/l.StorageI/PFDM/FINAL"
PATH=$( pwd )/bin:$( pwd )/scripts:$PATH
```

analisis/scripts/analisis_dubstep.sh

```
#!/bin/bash
echo $( date )
source base/config.cfg
./scripts/smd_data_dubstep_kjmol
mkdir $datadir/LOG/WORK -p
rm $datadir/LOG/WORK/dubstep*work
for i in {0..6}; do
  for j in $( ls $datadir/LOG/SMD/dubstep.fast.*.met.$i.* ); do
    tail $j -n 1 | cut -d" " -f3 >> $datadir/LOG/WORK/dubstep.-
fast.met.$i.work
  done
done
```

```
for i in {0..10}; do
  for j in $( ls $datadir/LOG/SMD/dubstep.fast.*.phos.$i.* ); do
    tail $j -n 1 | cut -d" " -f3 >> $datadir/LOG/WORK/dubstep.-
fast.phos.$i.work
  done
done
for i in $( ls $datadir/LOG/WORK/dubstep*work ); do
  ./scripts/block_avg.extrap.pl $i 310 kJ
done
for i in $( ls $datadir/LOG/WORK/dubstep*fn);do
  cat $i | cut -f2 > $( echo $i | sed -e "s/work\.fn/ba/" )
done
h="$datadir/LOG/WORK/lista"
for i in {0..6}; do
  h="$h $datadir/LOG/WORK/dubstep.fast.met.$i.ba"
done

for i in {0..10}; do
  h="$h $datadir/LOG/WORK/dubstep.fast.phos.$i.ba"
done
if [ -e "$datadir/LOG/WORK/lista" ]; then
  rm $datadir/LOG/WORK/lista
fi
for i in `eval echo {0..$( echo $( echo $h | cut -d" " -f2 | wc
-l ))}`;do
  echo $i >> $datadir/LOG/WORK/lista
done
paste $h > $datadir/LOG/WORK/dubstep.fast.summary.ba
for i in $( ls $datadir/LOG/WORK/dubstep.fast*.fn ); do
  echo "Analizando $i"
  ./scripts/df.extrap.pl $i
done
echo $( date)
```

analisis/scripts/block_avg.extrap.pl

```
#!/usr/bin/perl -w
#########################################################################
#                                                                       #
#  This perl script takes a file (e.g., "file.dat")                     #
#  containing work values in single-column format                       #
#  and generates:                                                       #
#    (1) a file containing the block averaged free energy               #
#        estimates "fn.example.dat"                                     #
#    (2) a file containing the Jarzynski running estimate               #
#        "jarz.example.dat"                                             #
#  A plot of (1) and (2) will look like Fig 2 in the                    #
```

```
#  the reference below                                                  #
#                                                                       #
# Usage: perl block_avg.extrap.pl file.dat temp(K) units(kJ or kcal)#
#                                                                       #
#  F.M. Ytreberg and D.M. Zuckerman, J. Comput. Chem.,                  #
#  25:1749 (2004).                                                      #
#                                                                       #
#########################################################################

if(@ARGV!=3){
    die "usage: block_avg.extrap.pl file.dat Temperature(K) [kJ]/
[kcal]\n";
}
$minprecision = 0.001; #precision for block averages; units are % of
fn
$filename=$ARGV[0];
$T=$ARGV[1];
print "    temperature of $T K\n";
$units=$ARGV[2];
if($units eq 'kJ'){
    $kT=0.008314472*$T;
}
elsif($units eq 'kcal'){
    $kT=0.0019872065*$T;
}
else{
    die "\nFATAL: invalid units\n\n";
}
$beta=1.0/$kT;
#read in all work values and generate the Jarzynski estimate
open(IN,"<$filename") or die "\nFATAL: could not open $filename\n\n";
$N=0;
while(<IN>){
    @data=split;
    $N++;
    $work[$N]=$data[0];
}
for($n=1; $n<=$N; $n++){
    $tmpjarz=0.0;
    for($j=1; $j<=$n; $j++){
     $tmpjarz+=exp(-$beta*$work[$j]);
    }
    $jarz[$n]=-$kT*log($tmpjarz/$n);
}
printf("  Jarzynski estimate using all %d work values is %.4f\n",$N,
$jarz[$N]);
```

```
print "  Generating block averaged free energy... \n";
#calculate the sub-sampled block averaged free energy
$fnfilename="$filename.fn";
open(FN,">$fnfilename");
$maxsweep=10000;
$times=0;
for($n=1; $n<=$N; $n++){ #number of work values per block
    $flag=1;
    $sweep=0;
    $fntot=0.0;
    $oldfnave=0.0;
    while($flag>0){
     &shuffle;
     $sweep++;
     $nb=int($N/$n); #number of blocks
     $fntmp=0.0;
     for($bi=1; $bi<=$nb; $bi++){ #block index
         $expave=0.0;
         for($sbi=1; $sbi<=$n; $sbi++){ #sub block index
          $wi=$sbi+($bi-1)*$n;

          $expave+=exp(-$beta*$work[$wi]);
         }
         if($expave==0){
          $expave=1.0e1000;
         }
         else{
          $expave=-$kT*(log($expave)-log($n));
         }
         $fntmp+=$expave;
     }
     $fntmp=$fntmp/$nb;
     $fntot+=$fntmp;
     if($n eq 1 or $n eq $N){
         $flag=-1;
     }
     elsif($sweep%50 eq 0){
         $fnave=$fntot/$sweep;
         $diff=abs($oldfnave-$fnave);
         if($diff<=$fnave*$minprecision*0.01){
          $flag=-1;
         }
         $oldfnave=$fnave;
     }
     elsif($sweep>=$maxsweep){
         $flag=-1;
     }
```

```
    }
    $fn[$n]=$fntot/$sweep;
    if($fn[$n]<1.0e100){
     print "    $n work value(s) per block\n";
     print FN "$n \t $fn[$n] \n";
    }
    else{
     print "    not using $n work value(s) per block --  fn unde-
fined\n";
    }
    if($n>=$N){
     if($times==0){
         $times++;
         $n=$N;
     }
    }
}
close(FN);
print "      block averages output to file $filename.fn \n";
$jarzfilename="$filename.jarz";
open(JARZ,">$jarzfilename");
for($n=1; $n<=$N; $n++){
    print JARZ "$n \t $jarz[$n] \n";
}
close(JARZ);
print "      Jarzynski running average output to file
$filename.jarz \n\n";
#shuffle the work values
sub shuffle{
    for($dest=$N; $dest>1; $dest--){
    $src=int(rand $dest)+1; #random (1,dest)
    $tmp=$work[$src];
    $work[$src]=$work[$dest];
    $work[$dest]=$tmp;
  }
}
```

analisis/scripts/per_resid_rmsd.tcl

```
# este script halla el rmsd de cada residuo y lo escribe en un archi-
vo todo y tal
proc rmsd_per_resid {outfile1 outfile2} {
  set ofile [open $outfile1 w]
  set ofile2 [open $outfile2 w]
  set nf [molinfo top get numframes]
  set sel1  [atomselect top "backbone" ]
  set sel2 [atomselect top "all" ]
```

```
  set f0 [atomselect top "backbone" frame 0]
  set gres [atomselect top "backbone and alpha"]
  set reslist [$gres get resid]
  set header1 [$gres get {resname resid}]
  puts $ofile "FRAME $header1"
  for {set i 0} {$i<=$nf} {incr i 50} {
    set templist $i
    puts "Frame: $i"
    $sel1 frame $i
    $sel2 frame $i
    $sel2 move [measure fit $sel1 $f0]
    foreach r $reslist {
      set sel3 [atomselect top "backbone and resid $r" frame 0]
      set sel4 [atomselect top "backbone and resid $r" frame $i]
      set sel5 [atomselect top "protein and resid $r" frame 0]
      set sel6 [atomselect top "protein and resid $r" frame $i]
      set rmsd1 [measure rmsd $sel3 $sel4]
      set rmsd2 [measure rmsd $sel5 $sel6]
      set templist [lappend templist $rmsd1]
      puts $ofile2 "$i $r $rmsd1 $rmsd2"
      $sel6 delete
      $sel5 delete
      $sel4 delete
      $sel3 delete
    }
    puts $ofile "$templist"
    puts $ofile2 ""
  }
  close $ofile
  close $ofile2
}
```

analisis/scripts/concat_finals.sh

```
#!/bin/bash
#get_finals:
#  si hay un reloaded, pilla ese.
#  si hay un continue, concatena con el base
#  en caso contrario, copia el que ya esta y punto

source base/config.cfg
newdir=/media/l.StorageI/PFDM/FINAL
mkdir $newdir/TRAY -p
mkdir $newdir/VEL -p
mkdir $newdir/LOG -p
mkdir $newdir/XST -p
for t in {0..6}; do
```

```
  g="met"
  for i in {0..6}; do
    for j in {0..7}; do
      f1=$( ls $datadir/TRAY/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-
tray.dcd 2>/dev/null )
      f2=$( ls $datadir/TRAY/dubstep.continue.*.$t.1FLV.$g.$i.$j-
tray.dcd 2>/dev/null )
      if [ -e "$f1" ]; then
        echo "tratando el archivo $( ls $datadir/TRAY/dub-
step.reloaded.*.$t.1FLV.$g.$i.$j-tray.dcd | rev | cut -d'/' -f1 | rev
| sed -e 's/-tray.dcd//')"
        cp $datadir/TRAY/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-tray.dcd
$newdir/TRAY/final.1FLV.$g.$t.$i.$j-tray.dcd
        cp $datadir/VEL/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-vel.dcd
$newdir/VEL/final.1FLV.$g.$t.$i.$j-vel.dcd
        cp $datadir/LOG/dubstep.reloaded.*.$t.1FLV.$g.$i.$j.log
$newdir/LOG/final.1FLV.$g.$t.$i.$j.log
        cp $datadir/XST/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-pcell.xst
$newdir/XST/final.1FLV.$g.$t.$i.$j-pcell.xst
      elif [ -e "$f2" ]; then
        echo "tratando el archivo $( ls $datadir/TRAY/dubstep.con-
tinue.*.$t.1FLV.$g.$i.$j-tray.dcd | rev | cut -d'/' -f1 | rev | sed
-e 's/-tray.dcd//')"
        catdcd -o $newdir/TRAY/final.1FLV.$g.$t.$i.$j-tray.dcd -otype
dcd -stype psf -s $datadir/ready_1FLV.psf -dcd $datadir/TRAY/dub-
step.fast.$t.1FLV.$g.$i.$j-tray.dcd -dcd $datadir/TRAY/dubstep.con-
tinue.*.$t.1FLV.$g.$i.$j-tray.dcd
        catdcd -o $newdir/VEL/final.1FLV.$g.$t.$i.$j-vel.dcd -otype
dcd -stype psf -s $datadir/ready_1FLV.psf -dcd $datadir/VEL/dubstep.-
fast.$t.1FLV.$g.$i.$j-vel.dcd -dcd $datadir/VEL/dubstep.continue.*.
$t.1FLV.$g.$i.$j-vel.dcd
        cat $datadir/LOG/dubstep.fast.$t.1FLV.$g.$i.$j.log
$datadir/LOG/dubstep.continue.*.$t.1FLV.$g.$i.$j.log >
$newdir/LOG/final.1FLV.$g.$t.$i.$j.log
        cp $datadir/XST/dubstep.fast.$t.1FLV.$g.$i.$j-pcell.xst
$newdir/XST/final.1FLV.$t.$i.$j-pcell.xst
        tail $datadir/XST/dubstep.continue.*.$t.1FLV.$g.$i.$j-pcel-
l.xst -n +4 >> $newdir/XST/final.1FLV.$t.$i.$j-pcell.xst
      else
        echo "tratando el archivo $( ls $datadir/TRAY/dubstep.fast.
$t.1FLV.$g.$i.$j-tray.dcd | rev | cut -d'/' -f1 | rev | sed -e 's/-
tray.dcd//')"
        cp $datadir/TRAY/dubstep.fast.$t.1FLV.$g.$i.$j-tray.dcd
$newdir/TRAY/final.1FLV.$g.$t.$i.$j-tray.dcd
        cp $datadir/VEL/dubstep.fast.$t.1FLV.$g.$i.$j-vel.dcd
$newdir/VEL/final.1FLV.$g.$t.$i.$j-vel.dcd
        cp $datadir/LOG/dubstep.fast.$t.1FLV.$g.$i.$j.log
```

```
$newdir/LOG/final.1FLV.$g.$t.$i.$j.log
        cp $datadir/XST/dubstep.fast.$t.1FLV.$g.$i.$j-pcell.xst
$newdir/XST/final.1FLV.$g.$t.$i.$j-pcell.xst
      fi
    done
  done
  g="phos"
  for i in {0..10}; do
    for j in {0..7}; do
      f1=$( ls $datadir/TRAY/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-
tray.dcd 2>/dev/null )
      f2=$( ls $datadir/TRAY/dubstep.continue.*.$t.1FLV.$g.$i.$j-
tray.dcd 2>/dev/null )
      if [ -e "$f1" ]; then
        echo "tratando el archivo $( ls $datadir/TRAY/dub-
step.reloaded.*.$t.1FLV.$g.$i.$j-tray.dcd | rev | cut -d'/' -f1 | rev
| sed -e 's/-tray.dcd//')"
        cp $datadir/TRAY/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-tray.dcd
$newdir/TRAY/final.1FLV.$g.$t.$i.$j-tray.dcd
        cp $datadir/VEL/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-vel.dcd
$newdir/VEL/final.1FLV.$g.$t.$i.$j-vel.dcd
        cp $datadir/LOG/dubstep.reloaded.*.$t.1FLV.$g.$i.$j.log
$newdir/LOG/final.1FLV.$g.$t.$i.$j.log
        cp $datadir/XST/dubstep.reloaded.*.$t.1FLV.$g.$i.$j-pcell.xst
$newdir/XST/final.1FLV.$g.$t.$i.$j-pcell.xst
      elif [ -e "$f2" ]; then
        echo "tratando el archivo $( ls $datadir/TRAY/dubstep.con-
tinue.*.$t.1FLV.$g.$i.$j-tray.dcd | rev | cut -d'/' -f1 | rev | sed
-e 's/-tray.dcd//')"
        catdcd -o $newdir/TRAY/final.1FLV.$g.$t.$i.$j-tray.dcd -otype
dcd -stype psf -s $datadir/ready_1FLV.psf -dcd $datadir/TRAY/dub-
step.fast.$t.1FLV.$g.$i.$j-tray.dcd -dcd $datadir/TRAY/dubstep.con-
tinue.*.$t.1FLV.$g.$i.$j-tray.dcd
        catdcd -o $newdir/VEL/final.1FLV.$g.$t.$i.$j-vel.dcd -otype
dcd -stype psf -s $datadir/ready_1FLV.psf -dcd $datadir/VEL/dubstep.-
fast.$t.1FLV.$g.$i.$j-vel.dcd -dcd $datadir/VEL/dubstep.continue.*.
$t.1FLV.$g.$i.$j-vel.dcd
        cat $datadir/LOG/dubstep.fast.$t.1FLV.$g.$i.$j.log
$datadir/LOG/dubstep.continue.*.$t.1FLV.$g.$i.$j.log >
$newdir/LOG/final.1FLV.$g.$t.$i.$j.log
        cp $datadir/XST/dubstep.fast.$t.1FLV.$g.$i.$j-pcell.xst
$newdir/XST/final.1FLV.$t.$i.$j-pcell.xst
        tail $datadir/XST/dubstep.continue.*.$t.1FLV.$g.$i.$j-pcel-
l.xst -n +4 >> $newdir/XST/final.1FLV.$t.$i.$j-pcell.xst
      else
        echo "tratando el archivo $( ls $datadir/TRAY/dubstep.fast.
$t.1FLV.$g.$i.$j-tray.dcd | rev | cut -d'/' -f1 | rev | sed -e 's/-
```

```
tray.dcd//')"
        cp $datadir/TRAY/dubstep.fast.$t.1FLV.$g.$i.$j-tray.dcd
$newdir/TRAY/final.1FLV.$g.$t.$i.$j-tray.dcd
        cp $datadir/VEL/dubstep.fast.$t.1FLV.$g.$i.$j-vel.dcd
$newdir/VEL/final.1FLV.$g.$t.$i.$j-vel.dcd
        cp $datadir/LOG/dubstep.fast.$t.1FLV.$g.$i.$j.log
$newdir/LOG/final.1FLV.$g.$t.$i.$j.log
        cp $datadir/XST/dubstep.fast.$t.1FLV.$g.$i.$j-pcell.xst
$newdir/XST/final.1FLV.$g.$t.$i.$j-pcell.xst
      fi
    done
  done
done
```

analisis/scripts/configjobs.doublestep.fast.sh

```
#!/bin/bash
# generacion de produccion
vmult=10
steps=625000
for n in $( echo "doublestep.fast.1FLV" ); do
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_doublestep.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
g=met
```

```
j=0
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail -n 1 )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_doublestep.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/NAME1/$n/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
vmult=$( echo "$vmult * 0.50" | bc )
done
```

analisis/scripts/configjobs.fast.sh

```
#!/bin/bash
# generacion de produccion

vmult=10

for n in $( echo "fast.1FLV" ); do
g=phos
j=0
for j in {0..7}; do
  for i in {0..10}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
    svel=$( echo "$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail
-n 1 ) * $vmult" | bc )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
```

```
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_general.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
g=met
j=0
for j in {0..7}; do
  for i in {0..6}; do
    sdir=$( head -n $(( $i + 1)) base/VEC/1FLV.$g.$j.vec | tail -n
1 )
    sk=$( echo "$( head -n $(( $i +1 )) base/1FLV.$g.k | tail -n 1 )
* $vmult" | bc )
    svel=$( head -n $(( $i + 1 )) base/1FLV.$g.speeds | tail -n 1 )
    cp templates/template.job jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[SATOM\]/$g/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[VINDEX\]/$i/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[RINDEX\]/$j/" jobs/$n.$g.$i.$j.sh
    sed -i -e "s/\[INPFILE1\]/$n.$g.$i.$j.inp/" jobs/$n.$g.$i.$j.sh
    cp templates/smd_general.inp inps/$n.$g.$i.$j.inp
    sed -i -e "s/SATOM1/$g/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/VINDEX1/$i/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/RINDEX1/$j/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/K1/$sk/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SVEL1/$svel/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/SDIR1/$sdir/" inps/$n.$g.$i.$j.inp
    sed -i -e "s/STEPS1/$steps/" inps/$n.$g.$i.$j.inp
  done
done
vmult=$( echo "$vmult * 0.50" | bc )
done
```

analisis/scripts/datosclave_v4.tcl

```
proc get_distances { outfile} {
  #mola, porque es mas de dos veces mas rapido que el anterior...
  #selecciones para el alineamiento
  set sel1  [atomselect top "all and backbone" ]
  set sel2 [atomselect top "all" ]
```

```
  set f0 [atomselect top "all and backbone" frame 0]
  #selecciones para las iteraciones
  set nf [molinfo top get numframes]
  set prot0 [atomselect top "protein" frame 0]
  set isoaa0 [atomselect top "index 2572 to 2598" frame 0]
  set phos0 [atomselect top "index 2618 to 2621" frame 0]
  #selecciones para las distancias no iterativas
  set p0 [atomselect top "name P" frame 0 ]
  set p [atomselect top "name P" ]
  set m0 [atomselect top "name C8M" frame 0]
  set m [atomselect top "name C8M" ]

  #inicio de los calculos
  #recorremos todos los frames del archivo...
  set of [open $outfile w]
  puts $of "frame mdistiaa mdistphos dispiaa dispphos"
  for {set i 0} {$i<$nf} {incr i} {
    puts "Frame: $i"
    # alineamos cada frame al inicial comparando los backbones
    $sel1 frame $i
    $sel2 frame $i
    $sel2 move [measure fit $sel1 $f0]
    # y actualizamos el resto de selecciones
    $m frame $i
    $p frame $i
    set red_prot [atomselect top "protein and within 8 of resname
FMN" frame $i];
    #calculo de la distancia minima entre isoaloxazina y proteina
      set aux0 0
      foreach j "[$isoaa0 list] " {
        foreach k "[$red_prot list] " {
          set aux_sel1 [atomselect top "index $j" frame $i]
          set aux_sel2 [atomselect top "index $k" frame $i]
          if { $aux0 == 0} {
            set aux0 1
            set mindistisoaa [veclength [vecsub [lindex [$aux_sel1
get {x y z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
          } else {
            set aux2 [veclength [vecsub [lindex [$aux_sel1 get {x y
z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
            if { $aux2 < $mindistisoaa} {
              set mindistisoaa $aux2
            }
          }
          $aux_sel1 delete
          $aux_sel2 delete
        }
```

```
      }
    #calculo de la distancia minima entre fosfato y proteina
      set aux0 0
      foreach j "[$phos0 list] " {
       foreach k "[$red_prot list] " {
          set aux_sel1 [atomselect top "index $j" frame $i]
          set aux_sel2 [atomselect top "index $k" frame $i]
          if { $aux0 == 0} {
            set aux0 1
            set mindistphos [veclength [vecsub [lindex [$aux_sel1 get
{x y z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
          } else {
            set aux2 [veclength [vecsub [lindex [$aux_sel1 get {x y
z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
            if { $aux2 < $mindistphos} {set mindistphos $aux2}
          }
          $aux_sel1 delete
          $aux_sel2 delete
        }
      }
    #calculo del desplazamiento del metilo
      set despmet [veclength [vecsub [lindex [$m get {x y z}] 0]
[lindex [$m0 get {x y z}] 0 ]]]
    #calculo del desplazamiento del fosforo
      set despphos [veclength [vecsub [lindex [$p get {x y z}] 0]
[lindex [$p0 get {x y z}] 0 ]]]
      $red_prot delete
    puts $of "$i $mindistisoaa $mindistphos $despmet $despphos"
  }
  close $of
}
```

analisis/scripts/df.extrap.pl

```
#!/usr/bin/perl -w
##############################################################
#                                                            #
#  This script takes a file containing block averaged free #
#  energies (e.g., "fn.file.dat") and generates:             #
#    (1) an estimate for the free energy using the           #
#        the standard Jarzynski relation with all            #
#        available work values                               #
#    (2) a cumulative integral extrapolated estimate         #
#    (3) a linearly extrapolated estimate                    #
#                                                            #
#  Usage: perl extrap_est.extrap.pl fn.file.dat              #
#                                                            #
```

```
#  Estimates (2) and (3) are described in                  #
#  F.M. Ytreberg and D.M. Zuckerman, J. Comput. Chem.,     #
#  25:1749 (2004).                                         #
#                                                          #
############################################################

if(@ARGV!=1){
    die "usage: df.extrap.pl fn.file.dat\n";
}
$filename=$ARGV[0];
open(IN,"<$filename") or die "\nFATAL: could not open $filename\n\n";
#read in block averaged data from a file
$N=0;
while(<IN>){
    $N++;
    @data=split;
    $n[$N]=$data[0];
    $fn[$N]=$data[1];
}
$Fjarz=$fn[$N];
printf("  Jarzynski Estimate: %.4f\n",$Fjarz);
#find linear estimate
$tau=0.5;
$nmin=$N-int($N/5);
if($nmin>$N-1){
    $nmin=$N-1;
}
$points=0;
$sumx=0.0;
$sumx2=0.0;
$sumy=0.0;
$sumxy=0.0;
for($i=$nmin; $i<=$N; $i++){
    $points++;
    $tmpx=$n[$i]**(-$tau);
    $tmpy=$fn[$i];
    $sumx+=$tmpx;
    $sumx2+=$tmpx**2;
    $sumy+=$tmpy;
    $sumxy+=$tmpy*$tmpx;
}
$Flin=($sumy*$sumx2-$sumx*$sumxy)/($points*$sumx2-$sumx**2);
printf("  Linearly extrapolated estimate: %.4f\n",$Flin);
#find cumulative integral estimate
if($Fjarz<0){ #if fn values are negative, shift them to be positive
    for($i=1; $i<=$N; $i++){
     $fn[$i]=$fn[$i]-2.0*$Fjarz;
```

```
    }
}
$minslope=1.0e100;
for($tau=0.3; $tau<=3.0; $tau+=0.1){ #find best value of tau
    #use tau to convert n-data
    for($i=1; $i<=$N; $i++){
     $x[$i]=$n[$i]**(-$tau);
    }
    #find cumulative integral (CI)
    for($i=1; $i<=$N-1; $i++){
     $nmin=$i-int($i/5)-1;
     if($nmin<=1){
         $nmin=1;
     }
     $nmax=$i+int($i/5)+1;
     if($nmax>=$N){
         $nmax=$N;
     }
     $points=0;
     $sumx=0.0;
     $sumx2=0.0;
     $sumy=0.0;
     $sumxy=0.0;
     for($j=$nmin; $j<=$nmax; $j++){
         $points++;
         $tmpx=$x[$j];
         $tmpy=$fn[$j];
         $sumx+=$tmpx;
         $sumx2+=$tmpx**2;
         $sumy+=$tmpy;
         $sumxy+=$tmpy*$tmpx;
     }
     $slope=($points*$sumxy-$sumx*$sumy)/($points*$sumx2-$sumx**2);
     $ftmp[$i]=$fn[$i]-(1.0-$x[$i])*$slope;
    }
    $cisum=0.0;
    for($i=1; $i<=$N-2; $i++){
     $cisum+=0.5*($x[$i]-$x[$i+1])*($ftmp[$i]+$ftmp[$i+1]);
     $ci[$i]=$cisum;
    }
    #test the slope of tail of CI
    $nmin=$N-int($N/5);
    if($nmin>$N-3){
     $nmin=$N-3;
    }
    $points=0;
    $sumx=0.0;
```

```
    $sumx2=0.0;
    $sumy=0.0;
    $sumxy=0.0;
    for($i=$nmin; $i<=$N-2; $i++){
     $points++;
     $tmpx=$x[$i];
     $tmpy=$ci[$i];
     $sumx+=$tmpx;
     $sumx2+=$tmpx**2;
     $sumy+=$tmpy;
     $sumxy+=$tmpy*$tmpx;
    }
    $slope=abs(($points*$sumxy-$sumx*$sumy)/($points*$sumx2-
$sumx**2));
    #choose the value of tau that minimizes the slope of the CI
    if($slope<$minslope){
     $minslope=$slope;
     $mintau=$tau;
     if($Fjarz<0){
         $Fci=$ci[$N-2]+2.0*$Fjarz;
     }
     else{
         $Fci=$ci[$N-2];
     }
    }
}
printf("  Cumulative Integral estimate: %.4f (tau=$mintau)\n",$Fci);
```

analisis/scripts/extract_continue

```
#!/bin/bash

# esta saca el ultimo frame del dcd (250) de velocidad y de trayecto-
ria en formato binario, y extrae el xsc del xst...

source base/config.cfg
mkdir $datadir/base/continue/ -p
#dcd a coor
for i in $( ls $datadir/TRAY/*dcd); do
  h=$(echo $i | rev | cut -d'/' -f1 | rev | sed -e "s/-
tray.dcd/.coor/" )
  catdcd -o $datadir/base/continue/$h -otype namdbin -s
$datadir/ready_1FLV.psf -stype psf -first 250 -last 250 -dcd $i
done
#dcd a vel
for i in $( ls $datadir/VEL/*dcd); do
  h=$(echo $i | rev | cut -d'/' -f1 | rev | sed -e "s/-vel.dcd/.vel/"
```

```
)
  catdcd -o $datadir/base/continue/$h -otype namdbin -s
$datadir/ready_1FLV.psf -stype psf -first 250 -last 250 -dcd $i
done
#xst to xsc
for i in $(ls $datadir/XST/*xst ); do
  h=$(echo $i | rev | cut -d'/' -f1 | rev | sed -e "s/-
pcell.xst/.xsc/" )
  head $i -n 2 > $datadir/base/continue/$h
  tail $i -n 1 >> $datadir/base/continue/$h
done
```

analisis/scripts/extract_distances.sh

```
#!/bin/bash

source base/config.cfg
newdir=/media/l.StorageI/PFDM/FINAL
mkdir $newdir/LOG/distancias/ -p
echo "trabajando los archivos $(ls $datadir/TRAY/$1 )"
for i in $(ls $newdir/TRAY/$1 ); do
  vmd -dispdev text -f $datadir/ready_1FLV.psf $i -e scripts/get_dis-
tancias_full.tcl -args $newdir/LOG/distancias/$( echo $i | rev | cut
-d'/' -f1 | rev | sed -e "s/-tray.dcd/.dist/" )
done
```

analisis/scripts/get_distances_full.tcl

```
source ~/PFDM/scripts/unwrap.tcl
source ~/PFDM/scripts/datosclave_v4.tcl
unwrap "resname FMN"
get_distances [lindex $argv 0]
exit
```

analisis/scripts/get_dummy_position.pl

```
#!/usr/bin/perl
open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = split(" ",<FILE>);
#print "($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1)";
$ts0=$ts1;
$px0=$px1;
$py0=$py1;
$pz0=$pz1;
$fx0=$fx1;
$fy0=$fy1;
```

```
$fz0=$fz1;
$wacum=0;
$dtot=0;
$mf=0;
print RESULT "TSTEP dTot Wacum ModF\n";
while (<FILE>) {
chomp;
($ts2, $px2, $py2, $pz2, $fx2, $fy2, $fz2) = split(" ");
#$dx= $px2-$px1;
#print "$dx\n";
#$wtemp=(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+($fz2+
$fz1)*($pz2-$pz1))/2;
#print "$wtemp\n";
$wacum=$wacum + (($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+
($fz2+$fz1)*($pz2-$pz1))/2;
$dtot=sqrt(($px2-$px0)**2 + ($py2-$py0)**2 + ($pz2-$pz0)**2);
$mf=sqrt(($fx2-$fx1)**2 + ($fy2-$fy1)**2 + ($fz2-$fz1)**2);
print RESULT "$ts2 $dtot $wacum $mf\n";
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = ($ts2, $px2, $py2, $pz2,
$fx2, $fy2, $fz2);
}
close (FILE);
close (RESULT);
```

analisis/scripts/get_puntosclave.tcl

```
source ~/PFDM/scripts/unwrap.tcl
source ~/PFDm/scripts/puntosclave.tcl
unwrap "resname FMN"
puntosclave [lindex $argv 0] [lindex $argv 1] [lindex $argv 2] [lin-
dex $argv 3] [lindex $argv 4]
```

analisis/scripts/get_smd_data.pl

```
#!/usr/bin/perl
open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = split(" ",<FILE>);
#print "($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1)";
$ts0=$ts1;
$px0=$px1;
$py0=$py1;
$pz0=$pz1;
$fx0=$fx1;
$fy0=$fy1;
$fz0=$fz1;
$wacum=0;
```

```
$dtot=0;
$mf=0;
print RESULT "TSTEP dTot Wacum ModF\n";
while (<FILE>) {
chomp;
($ts2, $px2, $py2, $pz2, $fx2, $fy2, $fz2) = split(" ");
#$dx= $px2-$px1;
#print "$dx\n";
#$wtemp=(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+($fz2+
$fz1)*($pz2-$pz1))/2;
#print "$wtemp\n";
$wacum=$wacum + (($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+
($fz2+$fz1)*($pz2-$pz1))/2;
$dtot=sqrt(($px2-$px0)**2 + ($py2-$py0)**2 + ($pz2-$pz0)**2);
$mf=sqrt(($fx2-$fx1)**2 + ($fy2-$fy1)**2 + ($fz2-$fz1)**2);
print RESULT "$ts2 $dtot $wacum $mf\n";
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = ($ts2, $px2, $py2, $pz2,
$fx2, $fy2, $fz2);
}

close (FILE);
close (RESULT);
```

analisis/scripts/get_smd_data_dubstep.pl

```
#!/usr/bin/perl
open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = split(" ",<FILE>);
#print "($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1)";
$ts0=$ts1;
$px0=$px1;
$py0=$py1;
$pz0=$pz1;
$fx0=$fx1;
$fy0=$fy1;
$fz0=$fz1;
$wacum=0;
$dtot=0;
$mf=0;
print RESULT "TSTEP dTot Wacum ModF\n";
while (<FILE>) {
chomp;
($ts2, $px2, $py2, $pz2, $fx2, $fy2, $fz2) = split(" ");
#$dx= $px2-$px1;
```

```
#print "$dx\n";
#$wtemp=(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+($fz2+
$fz1)*($pz2-$pz1))/2;
#print "$wtemp\n";
$wacum=$wacum + (($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+
($fz2+$fz1)*($pz2-$pz1))/2;
$dtot=sqrt(($px2-$px0)**2 + ($py2-$py0)**2 + ($pz2-$pz0)**2);
$mf=sqrt(($fx2-$fx1)**2 + ($fy2-$fy1)**2 + ($fz2-$fz1)**2);
print RESULT "$ts2 $dtot $wacum $mf\n";
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = ($ts2, $px2, $py2, $pz2,
$fx2, $fy2, $fz2);
}
close (FILE);
close (RESULT);
```

analisis/scripts/get_smd_data_dubstep_kjmol.pl

```
#!/usr/bin/perl
open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = split(" ",<FILE>);
#print "($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1)";
$ts0=$ts1;
$px0=$px1;
$py0=$py1;
$pz0=$pz1;
$fx0=$fx1;
$fy0=$fy1;
$fz0=$fz1;
$wacum=0;
$dtot=0;
$mf=0;
print RESULT "TSTEP dTot Wacum ModF\n";
while (<FILE>) {
chomp;
($ts2, $px2, $py2, $pz2, $fx2, $fy2, $fz2) = split(" ");
#$dx= $px2-$px1;
#print "$dx\n";
#$wtemp=(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+($fz2+
$fz1)*($pz2-$pz1))/2;
#print "$wtemp\n";
$wacum=$wacum + 0.06022*(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-
$py1)+($fz2+$fz1)*($pz2-$pz1))/2;
$dtot=sqrt(($px2-$px0)**2 + ($py2-$py0)**2 + ($pz2-$pz0)**2);
$mf=sqrt(($fx2-$fx1)**2 + ($fy2-$fy1)**2 + ($fz2-$fz1)**2);
print RESULT "$ts2 $dtot $wacum $mf\n";
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = ($ts2, $px2, $py2, $pz2,
```

```
$fx2, $fy2, $fz2);
}
close (FILE);
close (RESULT);
```

analisis/scripts/get_smd_data_final_kjmol.pl

```
#!/usr/bin/perl



open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = split(" ",<FILE>);
#print "($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1)";
$ts0=$ts1;
$px0=$px1;
$py0=$py1;
$pz0=$pz1;
$fx0=$fx1;
$fy0=$fy1;
$fz0=$fz1;
$wacum=0;
$dtot=0;
$mf=0;
print RESULT "TSTEP dTot Wacum ModF\n";
while (<FILE>) {
chomp;
($ts2, $px2, $py2, $pz2, $fx2, $fy2, $fz2) = split(" ");
#$dx= $px2-$px1;
#print "$dx\n";
#$wtemp=(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-$py1)+($fz2+
$fz1)*($pz2-$pz1))/2;
#print "$wtemp\n";
$wacum=$wacum + 0.06022*(($fx1 + $fx2)*($px2-$px1)+($fy2+$fy1)*($py2-
$py1)+($fz2+$fz1)*($pz2-$pz1))/2;
$dtot=sqrt(($px2-$px0)**2 + ($py2-$py0)**2 + ($pz2-$pz0)**2);
$mf=sqrt(($fx2-$fx1)**2 + ($fy2-$fy1)**2 + ($fz2-$fz1)**2);
print RESULT "$ts2 $dtot $wacum $mf\n";
($ts1, $px1, $py1, $pz1, $fx1, $fy1, $fz1) = ($ts2, $px2, $py2, $pz2,
$fx2, $fy2, $fz2);
}
close (FILE);
close (RESULT);
```

analisis/scripts/get_ts_clave.pl

```
#!/usr/bin/perl
#print "opening file $ARGV[0]\n";
open (FILE, "$ARGV[0]");
open (RESULT, ">$ARGV[1]");
($ts1, $dmm, $dmp, $dm, $dp) = split(" ",<FILE>);
$dmm0=0;
$dmm1=0;
$dmp0=0;
$dmp1=0;
$tsdmm=-1;
$tsdmp=-1;
while (<FILE>) {
chomp;
$lts=$ts1;
$ldp=$dp;
$ldm=$dm;
($ts1, $dmm, $dmp, $dm, $dp) = split(" ");
if (($dmm>=4) && ($dmm0>=4) && ($dmm1>=4) && ($tsdmm==-1))
{ $tsdmm=$ts1; $dpout=$dp; $dmpout=$dmp}
if (($dmp>=4) && ($dmp0>=4) && ($dmp1>=4) && ($tsdmp==-1))
{ $tsdmp=$ts1; $dmout=$dm; $dmmout=$dmm}
$dmm0=$dmm1;
$dmp0=$dmp1;
$dmm1=$dmm;
$dmp1=$dmp;
}
#if ($tsdmm==-1) {print RESULT "WARNING: ISOAA DIDN'T GET FAR
ENOUGH!!\n"; }
#if ($tsdmp==-1) {print RESULT "WARNING: PHOS DIDN'T GET FAR
ENOUGH!!\n"; }
if ($tsdmm==-1) { $tssdmm=$lts; $dpout=$ldp; $dmpout=$dmp1; }
if ($tsdmp==-1) { $tssdmp=$lts; $dmout=$ldm; $dmmout=$dmm1; }
#el resultado lo pone como TS de separacion de ISOAA, Desplazamiento
de phos y separacion de phos
# y sigue en la siguiente linea con TS de separacion de phos, despla-
zamiento de isoaa y separacion de isoaa
print RESULT "$tsdmm $dpout $dmpout\n";
print RESULT "$tsdmp $dmout $dmmout\n";
close (FILE);
close (RESULT);
```

analisis/scripts/get_work

```
#!/bin/bash
source base/config.cfg
mkdir -p $datadir/LOG/SMD/tmp
```

```
if [ ! -e "$datadir/LOG/SMD/tmp/frame" ]; then
  for i in {0..10000}; do
    echo $i >> $datadir/LOG/SMD/tmp/frame
  done
fi
for i in {0..10}; do
  h=""
  for j in {0..19}; do
  cat $datadir/LOG/SMD/1FLV.phos.$i.$j.smd | cut -d" " -f3 >
$datadir/LOG/SMD/tmp/1FLV.phos.$i.$j.work
  h="$h $datadir/LOG/SMD/tmp/1FLV.phos.$i.$j.work"
  done
  paste $datadir/LOG/SMD/tmp/frame $h > $datadir/LOG/SMD/1FLV.phos.
$i.work
  rm $h
done
for i in {0..6}; do
  h=""
  for j in {0..19}; do
  cat $datadir/LOG/SMD/1FLV.met.$i.$j.smd | cut -d" " -f3 >
$datadir/LOG/SMD/tmp/1FLV.met.$i.$j.work
  h="$h $datadir/LOG/SMD/tmp/1FLV.met.$i.$j.work"
  done
  paste $datadir/LOG/SMD/tmp/frame $h > $datadir/LOG/SMD/1FLV.met.
$i.work
  rm $h
done
```

analisis/scripts/mini_figures.sh

```
#/bin/bash
#plots para la minimizacion:
cd /media/HDD500/Documentos/imagenes
i="Ejemplo_convergencia_Jarzynski"
o="ejemplo_jarzynski.jpg"
xrange="[0:2700]"
yrange="[0:800]"
title="Ejemplo de convergencia de la igualdad de Jarzynski"
xlabel="Nº de simulaciones"
ylabel="E(kJ/mol)"
u="1:2"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
```

```
cd /media/l.StorageI/PFDM/MINI/LOG/
#E vs tstep en minim
i="min.inp.out.energy"
o="mini_e_vs_t_min.jpg"
xrange="[0:500]"
yrange="[-100000:500000]"
title="Energía total vs tiempo (minimizacion)"
xlabel="TStep"
ylabel="E(kcal/mol)"
u="2:12"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#V vs tstep en pre-dyna
cd /media/l.StorageI/PFDM/MINI/LOG/
i="pre-dyna-cpt-equilibration.inp.out.energy"
o="mini_v_vs_t_dyna.jpg"
xrange="[0:200000]"
yrange="[110000:135000]"
title="Volumen del sistema vs tiempo (equilibrado del agua)"
xlabel="TStep"
ylabel="Volumen (Å^3)"
u="2:19"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#E vs tstep en restraint
cd /media/l.StorageI/PFDM/MINI/LOG/
i="pre-restraining.total.out.energy.sust"
o="mini_E_vs_t_restrain.jpg"
xrange="[0:5000]"
yrange="[-50000:-35000]"
title="Energía total vs tiempo (equilibrado de la proteína)"
xlabel="TStep"
ylabel="E(kcal/mol)"
u="1:12"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
```

```
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#T vs t en calentamiento
cd /media/l.StorageI/PFDM/MINI/LOG/
i="pre-slow-heating-cpt.inp.out.energy"
o="mini_T_vs_t_heat.jpg"
xrange="[0:3000000]"
yrange="[0:330]"
title="Temperatura del sistema (calentamiento)"
xlabel="TStep"
ylabel="T (K)"
u="2:13"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#Ejemplo convergencia blockaverage
cd ~/PFDM/
i="trabajos_met_con_398.fn"
o="Ejemplo_convergencia_blockaverage.jpg"
xrange="[0:300]"
yrange="[0:850]"
title="Ejemplo de convergencia de la estimación por media de bloques"
xlabel="Tamaño del bloque"
ylabel="E(kJ/mol)"
u="1:2"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#E vs tstep en pre lgv
cd /media/l.StorageI/PFDM/MINI/LOG/
i="pre-LGV.inp.out.energy"
o="mini_E_vs_t_pre_lgv.jpg"
xrange="[0:200000]"
yrange="[-30000:-28000]"
title="Energía total vs tiempo (dinámica previa)"
xlabel="TStep"
ylabel="E(kcal/mol)"
```

```
u="2:12"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#RMSD global durante producción
cd ~/PFDM/
i="rmsd_global"
o="prod_rmsd_global.jpg"
xrange="[0:18400]"
yrange="[0:1.3]"
title="RMSD global en 100ns"
xlabel="TStep"
ylabel="RMSD"
u="1:2"
w="lines"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
$u w $w;"
mv $o /media/HDD500/Documentos/imagenes/
#Proyecciones sobre los dos primeros vectores de PCA
cd /media/l.StorageI/PFDM/Produccion/analisis/proyecciones
o="prod_PCA_proj_1_2.jpg"
xrange="[-8:8]"
yrange="[-8:8]"
title="RMSD global en 100ns"
xlabel="Proyección sobre el 2º vector"
ylabel="Proyección sobre el 1º vector"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l;
plot \"./90.proj\" u 2:3 w dots, \"./snapshots_newsmd\" u 2:3 w
points;"
mv $o /media/HDD500/Documentos/imagenes/
#RMSD por residuo 2D
cd /media/l.StorageI/PFDM/FINAL
i="prod_rmsd_list"
o="prod_rmsd_per_residue_2D.jpg"
xrange="[0:170]"
yrange="[0:6]"
title="RMSD por residuo"
```

```
xlabel="Residuo"
ylabel="RMSD"
l="off"
gnuplot -e "set terminal jpeg size 1600,1200; set output \"$o\"; set
xrange $xrange; set yrange $yrange; set title \"$title\"; set
ylabel \"$ylabel\"; set xlabel \"$xlabel\"; set key $l; plot \"$i\" u
2:4 w dots, \"$i\" u 2:3 w dots;"
mv $o /media/HDD500/Documentos/imagenes/
```

analisis/scripts/pca_all

```
#!/bin/bash
source base/config.cfg
for i in $( ls $datadir/TRAY/*-tray.dcd ); do
  echo "Trabajando archivo $i"
  basename=$( echo $i | rev | cut -d"/" -f1 | rev | sed -e "s/-
tray.dcd//" )
  echo
  echo
  echo "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
  echo "  Eliminando agua y cambiando a formato de AMBER"
  echo "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
  echo
  echo
  catdcd -o $datadir/TRAY/STRIP/$basename.nowat.crd -otype crd -stype
psf -s base/ready_1FLV.psf -i indexfile -dcd $i
  echo
  echo
  echo "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
  echo "  Analizando Componentes Principales"
  echo "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
  echo
  echo
  pcazip -i $datadir/TRAY/STRIP/$basename.nowat.crd -m base/fil-
ter.pdb -n 2622 -o $datadir/PCZ/$basename.pcz
  echo
  echo
  echo "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
  echo "  Extrayendo Información..."
  echo "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
  echo
  echo
  mkdir $datadir/PCZ/$basename
  mkdir $datadir/PCZ/$basename/tmp
  tvec=$( pczdump -i $datadir/PCZ/$basename.pcz --info | grep Vectors
| cut -d":" -f2 | sed -e "s/ //g" )
  for j in {1..2000}; do
```

```
    echo $j >> $datadir/PCZ/$basename/tmp/frames
  done
  h="$datadir/PCZ/$basename/tmp/frames"
  for j in {1..$tvec};do
    pczdump -i $datadir/PCZ/$basename.pcz --proj $j >  $datadir/PCZ/
$basename/tmp/$j
    h="$h $datadir/PCZ/$basename/tmp/$j"
  done
  paste $h > $datadir/PCZ/$basename/$basename.proj
done
```

analisis/scripts/plots_all

```
#!/bin/bash
source base/config.cfg
#hace los dibujos de posicion vs tiempo para cada trayectoria
#en la carpeta $datadir/PLOT/p_vs_t/
mkdir $datadir/PLOT/p_vs_t/tmp -p
mkdir $datadir/PLOT/w_vs_t -p
mkdir $datadir/PLOT/f_vs_p -p
if [ ! -e "$datadir/PLOT/p_vs_t/tmp/frame" ]; then
  for i in {0..10000}; do echo $(( $i*1000 )) >>
$datadir/PLOT/p_vs_t/tmp/frame; done
fi
echo "#######################################"
echo "Gráficos de desplazamiento vs tiempo:"
echo "#######################################"
echo
echo "-Calculando posicion de atomos dummy para:"
for g in $( echo "phos met" ); do
  h=0
  for i in $( cat base/1FLV.$g.speeds ); do
    echo "--1FLV.$g.$h..."
    if [ ! -e "$datadir/PLOT/p_vs_t/tmp/1FLV.$g.$h.dummy" ]; then
      echo "v=$i;t=0;for (t=0;t<10001;t++){print t*v*1000;
print \"\\n\"}" | bc > $datadir/PLOT/p_vs_t/tmp/1FLV.$g.$h.dummy.tmp
      paste $datadir/PLOT/p_vs_t/tmp/frame
$datadir/PLOT/p_vs_t/tmp/1FLV.$g.$h.dummy.tmp >
$datadir/PLOT/p_vs_t/tmp/1FLV.$g.$h.dummy
    fi
    h=$(( $h + 1 ))
  done
done
# grafico desplazamiento (A) vs t (ps) con desplazamiento del dummy
incluido
for g in $( echo "phos met"); do
  for j in {0..10}; do
```

```
    echo "-Creando gráficas de 1FLV.$g.$j..."
    for k in {0..19}; do
      i=$datadir/LOG/SMD/1FLV.$g.$j.$k.smd
      if [ -e "$i" ]; then
        gnuplot -e "set terminal jpeg size 800,600; set
output \"$datadir/PLOT/p_vs_t/$( echo $i | rev | cut -d'/' -f1 | rev
| sed -e 's/.smd/.jpg/' )\"; set yrange [-5:40]; plot \"$i\" using
1:2 with dots, \"$datadir/PLOT/p_vs_t/tmp/1FLV.$g.$j.dummy\" using
1:2 with lines;"
      fi
    done
  done
done

# grafico W (pN·A) vs t (ps)
echo "################################"
echo "Gráficos de trabajo vs tiempo:"
echo "################################"
echo
for g in $( echo "phos met"); do
  for j in {0..10}; do
    echo "-Creando gráficos de 1FLV.$g.$j"
    for k in {0..19}; do
      i=$datadir/LOG/SMD/1FLV.$g.$j.$k.smd
      if [ -e "$i" ]; then
        gnuplot -e "set terminal jpeg size 800,600; set
output \"$datadir/PLOT/w_vs_t/$( echo $i | rev | cut -d'/' -f1 | rev
| sed -e 's/.smd/.jpg/' )\"; plot \"$i\" using 1:3 with dots;"
      fi
    done
  done
done
# grafico F vs D
echo "######################################"
echo "Gráficos de fuerza vs desplazamiento:"
echo "######################################"
echo
for g in $( echo "phos met"); do
  for j in {0..10}; do
    echo "-Creando gráficos de 1FLV.$g.$j"
    for k in {0..19}; do
      i=$datadir/LOG/SMD/1FLV.$g.$j.$k.smd
      if [ -e "$i" ]; then
        r=$(( $( tail $datadir/PLOT/p_vs_t/tmp/1FLV.$g.$j.dummy -n 1
| cut -f2 | cut -d"." -f1 ) + 5))
        gnuplot -e "set terminal jpeg size 800,600; set
output \"$datadir/PLOT/f_vs_p/$( echo $i | rev | cut -d'/' -f1 | rev
```

```
| sed -e 's/.smd/.jpg/' )\"; set yrange [0:1000]; set xrange [0:$r];
plot \"$i\" using 2:4 with dots;"
      fi
    done
  done
done
```

analisis/scripts/plots_w_vs_t_f_vs_t

```
#!/bin/bash
source base/config.cfg
mkdir $datadir/PLOT/w_vs_t -p
mkdir $datadir/PLOT/f_vs_t -p
mkdir $datadir/PLOT/w_vs_d -p
for i in $( ls $datadir/LOG/SMD/*smd ); do
  gnuplot -e "set terminal jpeg size 800,600; set
output \"$datadir/PLOT/w_vs_t/$( echo $i | rev | cut -d'/' -f1 | rev
| sed -e 's/.smd/.jpg/' )\"; plot \"$i\" using 1:3 with dots;"
  gnuplot -e "set terminal jpeg size 800,600; set
output \"$datadir/PLOT/f_vs_t/$( echo $i | rev | cut -d'/' -f1 | rev
| sed -e 's/.smd/.jpg/' )\"; plot \"$i\" using 1:4 with dots;"
  gnuplot -e "set terminal jpeg size 800,600; set
output \"$datadir/PLOT/w_vs_d/$( echo $i | rev | cut -d'/' -f1 | rev
| sed -e 's/.smd/.jpg/' )\"; plot \"$i\" using 2:3 with dots;"
done
```

analisis/scripts/puntosclave.tcl

```
proc get_distances { target_mdiaa target_mdphos target_despm
target_despp outfile} {
  #variables para el control de flujo
  set reach_mdiaa 0
  set reach_mdphos 0
  set reach_despm 0
  set reach_despp 0
  #variables para el almacenamiento de resultados
  set mdiaa_frame 249
  set mdphos_frame 249
  set despm_frame 249
  set despp_frame 249
  #selecciones para el alineamiento
  set sel1  [atomselect top "all and backbone" ]
  set sel2 [atomselect top "all" ]
  set f0 [atomselect top "all and backbone" frame 0]
  #selecciones para las iteraciones
  set nf [molinfo top get numframes]
  set prot0 [atomselect top "protein" frame 0]
```

```
  set isoaa0 [atomselect top "index 2572 to 2598" frame 0]
  set phos0 [atomselect top "index 2618 to 2621" frame 0]
  #selecciones para las distancias no iterativas
  set p0 [atomselect top "name P" frame 0 ]
  set p [atomselect top "name P" ]
  set m0 [atomselect top "name C8M" frame 0]
  set m [atomselect top "name C8M" ]
  #inicio de los calculos
  #recorremos todos los frames del archivo...
  for {set i 0} {$i<$nf} {incr i} {
    # alineamos cada frame al inicial comparando los backbones
    $sel1 frame $i
    $sel2 frame $i
    $sel2 move [measure fit $sel1 $f0]
    # y actualizamos el resto de selecciones
    $m frame $i
    $p frame $i
    #calculo de la distancia minima entre isoaloxazina y proteina
    if { $reach_mdiaa == 0 } {
      set aux0 0
      foreach j "[$isoaa0 list] list" {
        foreach k "[$prot0 list] list" {
          set aux_sel1 [atomselect top "index $j" frame $i]
          set aux_sel2 [atomselect top "index $k" frame $i]
          if { $aux0 == 0}{
            set aux0 1
            set mindistisoaa [veclength [vecsub [lindex [$aux_sel1
get {x y z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
          } else {
            set aux2 [veclength [vecsub [lindex [$aux_sel1 get {x y
z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
            if { $aux2 < $mindistisoaa} {set mindistisoaa $aux2}
          }
        }
      }
    }
    #calculo de la distancia minima entre fosfato y proteina
    if { $reach_mdphos == 0 } {
      set aux0 0
      foreach j "[$phos0 list] list" {
        foreach k "[$prot0 list] list" {
          set aux_sel1 [atomselect top "index $j" frame $i]
          set aux_sel2 [atomselect top "index $k" frame $i]
          if { $aux0 == 0}{
            set mindistphos [veclength [vecsub [lindex [$aux_sel1 get
{x y z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
          } else {
```

```
            set aux2 [veclength [vecsub [lindex [$aux_sel1 get {x y
z}] 0] [lindex [$aux_sel2 get {x y z}] 0]]]
            if { $aux2 < $mindistphos} {set mindistphos $aux2}
          }
        }
      }
    }
    #calculo del desplazamiento del metilo
    if { $reach_despm == 0 } {
      set despmet [veclength [vecsub [lindex [$m get {x y z}] 0]
[lindex [$m0 get {x y z}] ]]]
    }
    #calculo del desplazamiento del fosforo
    if { $reach_despp == 0 } {
      set despphos [veclength [vecsub [lindex [$p get {x y z}] 0]
[lindex [$p0 get {x y z}] ]]]
    }
    #comprobación de los objetivos
    if {$mindistisoaa > $target_mdiaa} {
      set mdiaa_frame $i
      set reach_mdiaa 1
    }
    if {$mindistphos > $target_mdphos} {
      set mdphos_frame $i
      set reach_mdphos 1
    }
    if {$despmet > $target_despm} {
      set despm_frame $i
      set reach_despm 1
    }
    if {$despphos > $target_despp} {
      set despp_frame $i
      set reach_despp 1
    }
  }
  set of [open $outfile w]
  puts $of $mdiaa_frame
  puts $of $mdphos_frame
  puts $of $despm_frame
  puts $of $despp_frame
  close $of
}
```

analisis/scripts/set_references.tcl

```
set i [lindex $argv 0]
set s1 [atomselect top all]
```

```
set s2 [atomselect top "index 743 or index 1738 or index 2341 or in-
dex 2543"]
set met [atomselect top "index 2590"]
set phos [atomselect top "index 2618"]
$s1 set occupancy 0
$s1 set beta 0
$s2 set beta 1
$met set occupancy 1
$s1 writepdb base/REF/1FLV.$i.met.ref
$met set occupancy 0
$phos set occupancy 1
$s1 writepdb base/REF/1FLV.$i.phos.ref
exit
```

analisis/scripts/smd_data_all

```
#!/bin/bash
source  base/config.cfg
mkdir $datadir/LOG/SMD -p
mkdir smdtemp -p
for i in $( find $datadir/LOG/ -type f -iname "*.log" ); do
  cat $i | grep "SMD  " | sed -e "s/SMD  //" > smdtemp/$( echo $i |
rev | cut -d"/" -f1 | rev | sed -e "s/.log//")
done
cd smdtemp
for i in $( find ./ -type f ); do
  echo "archivo: $i"
  get_smd_data.pl $i $datadir/LOG/SMD/$( echo $i | sed -e
"s/\.\///" ).smd
done
```

analisis/scripts/smd_data_dubstep

```
#!/bin/bash
source  base/config.cfg
mkdir $datadir/LOG/SMD -p
mkdir smdtemp -p
for i in $( find $datadir/LOG/ -type f -iname "dubstep*.log" ); do
  cat $i | grep "SMD  " | sed -e "s/SMD  //" > smdtemp/$( echo $i |
rev | cut -d"/" -f1 | rev | sed -e "s/.log//")
done
cd smdtemp
for i in $( find ./ -type f -iname "dubstep*" ); do
  echo "archivo: $i"
  get_smd_data_dubstep.pl $i $datadir/LOG/SMD/$( echo $i | sed -e
"s/\.\///" ).smd
done
```

analisis/scripts/smd_data_dubstep_kjmol

```
#!/bin/bash
source  base/config.cfg
mkdir $datadir/LOG/SMD -p
mkdir smdtemp -p
for i in $( find $datadir/LOG/ -type f -iname "dubstep*.log" ); do
  cat $i | grep "SMD  " | sed -e "s/SMD  //" > smdtemp/$( echo $i |
rev | cut -d"/" -f1 | rev | sed -e "s/.log//")
done
cd smdtemp
for i in $( find ./ -type f -iname "dubstep*" ); do
  echo "archivo: $i"
  get_smd_data_dubstep_kjmol.pl $i $datadir/LOG/SMD/$( echo $i | sed
-e "s/\.\///" ).smd
done
```

analisis/scripts/smd_data_final_kjmol

```
#!/bin/bash
source  base/config.cfg
mkdir $datadir/LOG/SMD -p
mkdir smdtemp -p
for i in $( find $datadir/LOG/ -type f -iname "final*.log" ); do
  cat $i | grep "SMD  " | sed -e "s/SMD  //" > smdtemp/$( echo $i |
rev | cut -d"/" -f1 | rev | sed -e "s/.log//")
done
cd smdtemp
for i in $( find ./ -type f -iname "final*" ); do
  echo "archivo: $i"
  get_smd_data_final_kjmol.pl $i $datadir/LOG/SMD/$( echo $i | sed -e
"s/\.\///" ).smd
done
```

analisis/scripts/trabajosclave_exp.sh

```
#!/bin/bash
source base/config.cfg
mkdir $datadir/LOG/WORK -p
for i in $( ls $datadir/LOG/distancias/* ); do
  echo "procesando $i"
  get_ts_clave.pl $i $datadir/LOG/WORK/$( echo $i | rev | cut -d'/'
-f1 | rev | sed -e 's/.log//' ).limit.work.tmp
done
for i in $( ls $datadir/LOG/WORK/*.limit.work.tmp ); do
  echo "procesando: $i"
  n=$( echo $i | rev | cut -d'/' -f1 | rev | sed -e "s/.dist.limit.-
```

```
work.tmp//" )
  #met work
  l=$( tail $i -n 2 | head -n 1 )
  tsmet=$( echo $l | cut -d" " -f1 )
  dmp=$( echo $l | cut -d" " -f2 )
  dp=$( echo $l | cut -d" " -f3 )
  if [ "$tsmet" = "-1" ]; then
    mstat=0
    wmet=$( tail $datadir/LOG/SMD/$n.smd -n 1 | head -n 1 | cut -d' '
-f3)
    tsmet=$(($( tail $datadir/LOG/SMD/$n.smd -n 1 | head -n 1 | cut
-d' ' -f1) / 5000 ))
  else
    mstat=1
    wmet=$( cat $datadir/LOG/SMD/$n.smd | tail -n +$((($tsmet + 1 ) *
10 + 1)) | head -n 1 | cut -d' ' -f3)
  fi
  #phos work
  l=$( tail $i -n 1 )
  tsphos=$( echo $l | cut -d" " -f1 )
  dmm=$( echo $l | cut -d" " -f2 )
  dm=$( echo $l | cut -d" " -f3 )
  if [ "$tsphos" = "-1" ]; then
    pstat=0
    wphos=$( tail $datadir/LOG/SMD/$n.smd -n 1 | head -n 1 | cut -d'
' -f3)
    tsphos=$(($( tail $datadir/LOG/SMD/$n.smd -n 1 | head -n 1 | cut
-d' ' -f1) / 5000 ))
  else
    pstat=1
    wphos=$( cat $datadir/LOG/SMD/$n.smd | tail -n +$((($tsphos + 1 )
* 10 + 1)) | head -n 1 | cut -d' ' -f3)
  fi
  # tstep  estado  trabajo  desp_contrario
  echo "$tsmet $mstat $wmet $dp" > $datadir/LOG/WORK/$n.partial
  echo "$tsphos $pstat $wphos $dm" >> $datadir/LOG/WORK/$n.partial
done
echo "Recopilando: final.1FLV.met1.summary"
if [ -e "$datadir/LOG/WORK/final.1FLV.met1.summary" ]; then rm
$datadir/LOG/WORK/final.1FLV.met1.summary; fi
for i in $( ls  $datadir/LOG/WORK/final.1FLV.met.*.partial  ); do
  j=$(head $i -n 1 )
  if [ ! "$( echo $j | cut -d' ' -f2)" = "0" ]; then
    echo $j >> $datadir/LOG/WORK/final.1FLV.met1.summary
  fi
done
echo "Recopilando: final.1FLV.phos1.summary"
```

```
if [ -e "$datadir/LOG/WORK/final.1FLV.phos1.summary" ]; then rm
$datadir/LOG/WORK/final.1FLV.phos1.summary; fi
for i in $( ls  $datadir/LOG/WORK/final.1FLV.phos.*.partial  ); do
  j=$(tail $i -n 1 )
  if [ ! "$( echo $j | cut -d' ' -f2)" = "0" ]; then
    echo $j >> $datadir/LOG/WORK/final.1FLV.phos1.summary
  fi
done
echo "Recopilando: final.1FLV.met2.summary"
if [ -e "$datadir/LOG/WORK/final.1FLV.met2.summary" ]; then rm
$datadir/LOG/WORK/final.1FLV.met2.summary; fi
for i in $( ls  $datadir/LOG/WORK/final.1FLV.phos.*.partial  ); do
  j=$(head $i -n 1 )
  if [ ! "$( echo $j | cut -d' ' -f2)" = "0" ]; then
    echo $j >> $datadir/LOG/WORK/final.1FLV.met2.summary
  fi
done
echo "Recopilando: final.1FLV.phos2.summary"
if [ -e "$datadir/LOG/WORK/final.1FLV.phos2.summary" ]; then rm
$datadir/LOG/WORK/final.1FLV.phos2.summary; fi
for i in $( ls  $datadir/LOG/WORK/final.1FLV.met.*.partial  ); do
  j=$(tail $i -n 1 )
  if [ ! "$( echo $j | cut -d' ' -f2)" = "0" ]; then
    echo $j >> $datadir/LOG/WORK/final.1FLV.phos2.summary
  fi
done
echo "Recopilando: final.1FLV.combo.total"
if [ -e "$datadir/LOG/WORK/final.1FLV.combo.total" ]; then rm
$datadir/LOG/WORK/final.1FLV.combo.total; fi
for i in $( ls $datadir/LOG/WORK/final.1FLV.*.partial ); do
  j=$(head $i -n 1)
  k=$(tail $i -n 1)
  h=$( echo $i | rev | cut -d'/' -f1 | rev )
  fullid=$( echo $h  | sed -e "s/final\.1FLV\.//" | sed -e "s/\.par-
tial//" )
  f1=$( echo $fullid | cut -d'.' -f1 )
  f2=$( echo $fullid | cut -d'.' -f2 )
  f3=$( echo $fullid | cut -d'.' -f3 )
  f3=$( echo $fullid | cut -d'.' -f3 )
  if [ ! "$(echo $j | cut -d' ' -f2)" = "0" ] && [ ! "$(echo $k | cut
-d' ' -f2)" = "0" ]; then
    echo "$f1 $f2 $f3 $j $k" >> $datadir/LOG/WORK/final.1FLV.combo.-
total
  fi
done
rm $datadir/LOG/WORK/met.*.tmp
rm $datadir/LOG/WORK/phos.*.tmp
```

```
while read line; do
  echo $line >> $datadir/LOG/WORK/$(echo $line | cut -d' ' -f1 ).$
(echo $line | cut -d' ' -f3 ).tmp
done < $datadir/LOG/WORK/final.1FLV.combo.total
cat $( ls $datadir/LOG/WORK/met.*.tmp $datadir/LOG/WORK/phos.*.tmp )
> final.1FLV.ordered.combo.total
for i in $(ls $datadir/LOG/WORK/met.*.tmp
$datadir/LOG/WORK/phos.*.tmp ); do
  if [ "$( echo $i | rev | cut -d'/' -f1 | rev | cut -d'.' -f1 )" =
"met" ]; then
    cut -d' ' -f10 $i > $( echo $i | sed -e "s/tmp/work/" )
  else
    cut -d' ' -f6 $i > $( echo $i | sed -e "s/tmp/work/" )
  fi
  ./scripts/block_avg.extrap.pl $( echo $i | sed -e "s/tmp/work/" )
310 kJ
done
cut -d' ' -f3 $datadir/LOG/WORK/final.1FLV.phos1.summary >
$datadir/LOG/WORK/final.1FLV.phos1.work.summary
cut -d' ' -f3 $datadir/LOG/WORK/final.1FLV.phos2.summary >
$datadir/LOG/WORK/final.1FLV.phos2.work.summary
cut -d' ' -f3 $datadir/LOG/WORK/final.1FLV.met1.summary >
$datadir/LOG/WORK/final.1FLV.met1.work.summary
cut -d' ' -f3 $datadir/LOG/WORK/final.1FLV.met2.summary >
$datadir/LOG/WORK/final.1FLV.met2.work.summary
./scripts/block_avg.extrap.pl
$datadir/LOG/WORK/final.1FLV.phos1.work.summary 310 kJ
./scripts/block_avg.extrap.pl
$datadir/LOG/WORK/final.1FLV.phos2.work.summary 310 kJ
./scripts/block_avg.extrap.pl
$datadir/LOG/WORK/final.1FLV.met1.work.summary 310 kJ
./scripts/block_avg.extrap.pl
$datadir/LOG/WORK/final.1FLV.met2.work.summary 310 kJ
```

analisis/scripts/unwrap.tcl

```
# Unwrap 1.1
# Unwraps a selection in a trajectory
# Procedures:
#    unwrap      is residue-based
#    unwrap_atom is atom-based
#
# Jerome Henin <jhenin@cmm.upenn.edu>
proc unwrap { wrapSelText } {
    molinfo top set frame 0
    foreach {cutoffX cutoffY cutoffZ} [vecscale .5 [molinfo top get
{a b c}]] {}
```

```
    foreach {mcutoffX mcutoffY mcutoffZ} [vecinvert "$cutoffX $cut-
offY $cutoffZ"] {}
    if { ($cutoffX < 1) && ($cutoffY < 1) && ($cutoffZ < 1) } {
     error "Box abnormally small or no box size info available"
    }
    puts "Using cutoffs $cutoffX $cutoffY $cutoffZ"
    set wrapSel [atomselect top "($wrapSelText)"]
    set residueList [lsort -integer -unique [$wrapSel get residue]]
    set resList {}
    set refIndex {}
    foreach r $residueList {
     set res [atomselect top "($wrapSelText) and residue $r"]
     lappend resList $res
     # First atom of each residue
     lappend refIndex [lindex [$res list] 0]
     lappend offsetX 0
     lappend offsetY 0
     lappend offsetZ 0
    }
    set refAtoms [atomselect top "index $refIndex" frame 0]
    set previousX [$refAtoms get x]
    set previousY [$refAtoms get y]
    set previousZ [$refAtoms get z]
    set nf [molinfo top get numframes]
    for {set f 1} {$f < $nf} {incr f} {
     if {[expr {$f % ($nf/10)}] == 0} {
         puts "Frame $f"
     }
     molinfo top set frame $f
     foreach {a b c} [molinfo top get {a b c}] {}

     set i 0
     $refAtoms frame $f
     set allX [$refAtoms get x]
     set allY [$refAtoms get y]
     set allZ [$refAtoms get z]
     foreach res $resList X $allX Y $allY Z $allZ prevX $previousX
prevY $previousY prevZ $previousZ \
                              offX $offsetX offY $offsetY offZ $off-
setZ {
            foreach {dX dY dZ} [vecsub "$X $Y $Z" "$prevX $prevY
$prevZ"] {}
            if { $dX > $cutoffX } {
                incr offX -1; lset offsetX $i $offX
            } elseif { $dX < $mcutoffX } {
                incr offX; lset offsetX $i $offX
            }
```

```
            if { $dY > $cutoffY } {
                incr offY -1; lset offsetY $i $offY
            } elseif { $dY < $mcutoffY } {
                incr offY; lset offsetY $i $offY
            }
            if { $dZ > $cutoffZ } {
                incr offZ -1; lset offsetZ $i $offZ
            } elseif { $dZ < $mcutoffZ } {
                incr offZ; lset offsetZ $i $offZ
            }
          # move necessary residues
          if { $offX || $offY || $offZ } {
           $res frame $f
           $res moveby "[expr {$a*$offX}] [expr {$b*$offY}] [expr
{$c*$offZ}]"
          }
            incr i
      }
      # update saved positions
      set previousX $allX
      set previousY $allY
      set previousZ $allZ
    }
    # cleanup
    foreach res $resList {
     $res delete
    }
    $wrapSel delete
    $refAtoms delete
}

##########################################
# Same thing, atom-based
##########################################
proc unwrap_atom { wrapSelText } {
    molinfo top set frame 0
    foreach {cutoffX cutoffY cutoffZ} [vecscale .5 [molinfo top get
{a b c}]] {}
    foreach {mcutoffX mcutoffY mcutoffZ} [vecinvert "$cutoffX $cut-
offY $cutoffZ"] {}
    if { ($cutoffX < 1) && ($cutoffY < 1) && ($cutoffZ < 1) } {

     error "Box abnormally small or no box size info available"
    }
    puts "Using cutoffs $cutoffX $cutoffY $cutoffZ"
    set wrapSel [atomselect top "$wrapSelText" frame 0]
    set previousX [$wrapSel get x]
```

```
    set previousY [$wrapSel get y]
    set previousZ [$wrapSel get z]
    foreach i [$wrapSel list] {
     lappend offsetX 0
     lappend offsetY 0
     lappend offsetZ 0
    }
    set nf [molinfo top get numframes]
    for {set f 1} {$f < $nf} {incr f} {
     if {[expr {$f % ($nf/10)}] == 0} {
         puts "Frame $f"
     }
     molinfo top set frame $f
     foreach {a b c} [molinfo top get {a b c}] {}
     $wrapSel frame $f
     set i 0
     set allX [$wrapSel get x]
     set allY [$wrapSel get y]
     set allZ [$wrapSel get z]
     set newX {}
     set newY {}
     set newZ {}
     foreach X $allX Y $allY Z $allZ prevX $previousX prevY $previ-
ousY prevZ $previousZ \
                    offX $offsetX offY $offsetY offZ $offsetZ {
         foreach {dX dY dZ} [vecsub "$X $Y $Z" "$prevX $prevY
$prevZ"] {}
            if { $dX > $cutoffX } {
                incr offX -1; lset offsetX $i $offX
            } elseif { $dX < $mcutoffX } {
                incr offX; lset offsetX $i $offX
            }
            if { $dY > $cutoffY } {
                incr offY -1; lset offsetY $i $offY
            } elseif { $dY < $mcutoffY } {
                incr offY; lset offsetY $i $offY
            }
            if { $dZ > $cutoffZ } {
                incr offZ -1; lset offsetZ $i $offZ
            } elseif { $dZ < $mcutoffZ } {
                incr offZ; lset offsetZ $i $offZ
            }

         lappend newX [expr {$X+$a*$offX}]
         lappend newY [expr {$Y+$b*$offY}]
         lappend newZ [expr {$Z+$c*$offZ}]
         incr i
```

```
      }
      # move all atoms
      $wrapSel set x $newX
      $wrapSel set y $newY
      $wrapSel set z $newZ
      # update saved positions
      set previousX $allX
      set previousY $allY
      set previousZ $allZ
     }
     $wrapSel delete
}
```

2.Plantillas:

automation/templates/job.HEADER.LSF

```
#!/bin/bash
#BSUB -W WCL1
#BSUB -J NAME1
#BSUB -cwd
#BSUB -eo output/error.log
#BSUB -oo output/
#BSUB -n 2
#BSUB -R"span[ptile=16]"
source ../scripts/functions.bash
inpfile=INPFILE1
```

automation/templates/job.HEADER.LSF.serial

```
#!/bin/bash
#BSUB -W WCL1
#BSUB -J NAME1
#BSUB -cwd
#BSUB -eo output/error.log
#BSUB -oo output/
source ../scripts/functions.bash
```

automation/templates/job.HEADER.PBS

```
#!/bin/bash
#$ -N NAME1
#$ -e output/error.log
#$ -pe PARENV CORES
#$ -q QUEUE
#$ -cwd
#$ -m bea
#$ -M MAIL
source ../scripts/functions.bash
inpfile=INPFILE1
```

automation/templates/job.HEADER.PBS.serial

```
#!/bin/bash
#$ -N NAME1
#$ -e output/error.log
#$ -q QUEUE
#$ -cwd
#$ -m bea
```

```
#$ -M MAIL
source ../scripts/functions.bash
```

automation/templates/job.HEADER.SLURM

```
#!/bin/bash
# @ wall_clock_limit = WCL1
# @ job_name = NAME1
# @ error = output/error.log
# @ initialdir = .
# @ gpus_per_node = 2
# @ cpus_per_task = 1
# @ total_tasks = 12
source ../scripts/functions.bash
inpfile=INPFILE1
```

automation/templates/job.HEADER.SLURM.serial

```
#!/bin/bash
# @ wall_clock_limit = WCL1
# @ job_name = NAME1
# @ error = output/error.log
# @ initialdir = .
# @ gpus_per_node = 2
# @ cpus_per_task = 1
# @ total_tasks = 12
source ../scripts/functions.bash
inpfile=INPFILE1
```

automation/templates/job.MEMENTO

```
#Setting the environment to run with the 2.9 version
module load openmpi/gcc/64/1.4.4
module load namd/2.9
COMMAND="mpirun -np $NSLOTS namd2"
```

automation/templates/job.MINOTAURO

```
#Setting the environment to run with the 2.9 version
module load NAMD/2.9
COMMAND="srun namd2-cuda4.1 +idlepoll"
```

automation/templates/job.SMD.TAIL

```
source ../config.cfg
echo
```

```
f=$fecha
echo "$f - Starting [$inpfile]" >> output/logs/status.log
echo "Running from $( pwd )" >> output/logs/status.log
$COMMAND inps/$inpfile > output/logs/$inpfile.log
f=$fecha
echo "$f - Finishing [$inpfile]" >> output/logs/status.log
echo >> output/logs/status.log
```

automation/templates/job.TAIL

```
source ../config.cfg
CONFIG1
echo
f=$fecha
echo "$f - Starting [$inpfile]" >> output/status.log
echo "Running from $( pwd )" >> output/status.log
$COMMAND tmp/$inpfile > output/$inpfile.log
f=$fecha
echo "$f - Finishing [$inpfile]" >> output/status.log
echo >> output/status.log
```

automation/templates/job.TERMINUS

```
#Setting the environment to run with the 2.9 version
source setup.intelComp 64 12.0
source setup.mpi 64 openmpi 1.4.2-ifort-12.0
source ../config.cfg
COMMAND="mpirun -np $NSLOTS -machinefile $TMPDIR/machines
/apps/apps64/namd-2.9/namd2"
```

automation/templates/min1.template

```
#####################################################################
## Minimization using CONJ-GRAD under NAMD                         ##
## FIRST PREPARATION STEP                                          ##
## MINIMIZE AND ADECUATE PROTEIN FIXED                             ##
## Author: S.Cuesta Lopez - Oct. 2006                              ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#####################################################################



#####################################################################
## Updated and modified by Vladimir Espinosa                       ##
## University of Zaragoza - Nov. 2007                              ##
#####################################################################
```

```
#####################################################################
## PREPARATION PHASE - NAMD                                        ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007                       ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## by Vladimir Espinosa                                            ##
#####################################################################

#####################################################################
## ADJUSTABLE PARAMETERS                                           ##
#####################################################################

set name                NAME1
set path         PATH1
coordinates             $path/base/ready_$name.pdb
structure               $path/base/ready_$name.psf

# Force-Field Parameters
paratypecharmm          on
parameters              $path/base/PARM1
exclude                 scaled1-4
1-4scaling              1.0
COMmotion               no
margin                  0.0
stepspercycle           20

# Integrator Parameters - Varios ... shake, Hs
rigidBonds              water
rigidDieOnError         on
##useSettle              on

# NOTE: Do not set the periodic cell basis if you have also
# specified an .xsc restart file!
# Periodic Boundary Conditions
```

automation/templates/min2.template

```
wrapAll                 on
# PME (for full-system periodic electrostatics)
PME yes
PMEGridSizeX            64 # grid-pts along cellBasisVector1.
PMEGridSizeY            64 # along cellBasisVector2.
PMEGridSizeZ            64 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.
# (...cont) approximations
switching               on
```

```
switchdist              12.0
cutoff                  14.0
pairlistdist            22.0
pairlistsPerCycle       2
outputPairlists         5
## Center-of-mass motion is automatically removed.
## Fixed Atoms Constraint (set PDB beta-column to 1)
## Minimization fixing the protein
fixedAtoms              on
fixedAtomsFile          $path/tmp/fixed-atoms.pdb ## Only the protein
and the coordinating calcium are fixed
fixedAtomsCol           B  ## indicators for backbone atoms are 1.0 in
this col.
fixedAtomsForces        off
## Outputs
outputname              $path/output/$name-pre-min
binaryoutput            no
## Standard output INFO
outputEnergies          500
outputTiming            500

#############################################################

## EXECUTION SCRIPT                                         ##

#############################################################

# Minimization
minimization            on
minimize                500
```

automation/templates/pre-dyna-cpt.template

```
#############################################################
## Adecuate using CPT under NAMD                            ##
## SECOND PREPARATION STEP                                  ##
## ADEQUATE SOLVENT CAGE AND ENVIRONMENT - PROTEIN FIXED    ##
## Author: S.Cuesta Lopez - Oct. 2006                       ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#############################################################

#############################################################
## Updated and modified by Vladimir Espinosa                ##
## University of Zaragoza - Nov. 2007                       ##
#############################################################

#############################################################
```

```
## PREPARATION PHASE - NAMD                                          ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007                        ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## by Vladimir Espinosa                                             ##
#####################################################################

#####################################################################
## ADJUSTABLE PARAMETERS                                            ##
#####################################################################

set name                NAME
set path                        PATH1
set temperature         300
coordinates             $path/output/$name-pre-min.coor
structure               $path/base/ready_$name.psf
firsttimestep           0


#####################################################################
## SIMULATION PARAMETERS                                            ##
#####################################################################

# Force-Field Parameters
paratypecharmm          on
parameters              $path/base/PARM1
exclude                 scaled1-4
1-4scaling              1.0
COMmotion               no
margin                  0.0
# Integrator Parameters - Varios ... shake, Hs
rigidBonds              water
rigidDieOnError         on
##useSettle               on

## Getting the periodic cell parameters from the previous run
extendedSystem          $path/output/$name-pre-min.xsc

# NOTE: Do not set the periodic cell basis if you have also
# specified an .xsc restart file!
# Periodic Boundary Conditions
wrapAll                 on

# PME (for full-system periodic electrostatics)
PME yes
PMEGridSizeX            48 # grid-pts along cellBasisVector1.
PMEGridSizeY            54 # along cellBasisVector2.
PMEGridSizeZ            48 # along cellBasisVector3
```

```
# use numbers with small integer factors: 2,3,5.

# (...cont) approximations
switching              on
switchdist             12.0
cutoff                 14.0
pairlistdist           22.0
pairlistsPerCycle      2
outputPairlists        5

## Center-of-mass motion is automatically removed.

## Fixed Atoms Constraint (set PDB beta-column to 1)
## Minimization fixing the protein
fixedAtoms             on
fixedAtomsFile         $path/tmp/fixed-atoms.pdb ## Only the protein
and the coordinating calcium are fixed
fixedAtomsCol          B  ## indicators for backbone atoms are 1.0 in
this col.
fixedAtomsForces       off
# Integrator Parameters - Varios ... shake, Hs (...cont)
timestep               1.0  # in fs
stepspercycle          20
nonbondedFreq          2
temperature            $temperature
# Constant Temperature Control
langevin               on
langevinTemp           $temperature
langevinDamping        10.0 # Applied to all atoms
langevinHydrogen       on

## Constant Pressure Control (variable volume)
## Nose-Hoover Langevin Piston for pressure coupling
useGroupPressure       yes # needed for rigidBonds
useFlexibleCell        no
useConstantArea        no
langevinPiston         on
langevinPistonTarget   1.01325 # in bar -> 1 atm
langevinPistonPeriod   200.
langevinPistonDecay    100.
langevinPistonTemp     $temperature

## Outputs
outputname             $path/output/$name-dyna-cpt-equilibrated
binaryoutput           no
DCDfile                $path/output/$name-dyna-cpt-equilibrated-
tray.dcd
```

```
DCDUnitCell             yes
DCDfreq                 500
velDCDfile              $path/output/$name-dyna-cpt-equilibrated-
vel.dcd
velDCDfreq              500
XSTfile                 $path/output/$name-dyna-cpt-equilibrated-pcel-
l.xst
XSTfreq                 500

## Standard output INFO
outputEnergies          500
outputTiming            500

## Restart output files
restartname             $path/output/$name-dyna-cpt-equilibrated-
restart
restartfreq             500
restartsave             yes
binaryrestart           yes # Preserves more accuracy

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

## When restarting the number of efective steps MUST be (numsteps -
firsttimestep)
numsteps                500  # (1 000 000 - 0) -> 1 ns
```

automation/templates/pre-lgv.template

```
#############################################################
## INITIAL GLOBAL EQUILIBRATION AND HEATING under NAMD     ##
## FIFTH PREPARATION STEP                                  ##
## ADEQUATE PROTEIN - LGV EQUILIBRATION [NO CONSTRAINS]    ##
## Author: S.Cuesta Lopez - Oct. 2006                      ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#############################################################

#############################################################
## Updated and modified by Vladimir Espinosa               ##
## University of Zaragoza - Nov. 2007                      ##
#############################################################

#############################################################
## PREPARATION PHASE - NAMD                                ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007               ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
```

```
## by Vladimir Espinosa                                              ##
#####################################################################

#####################################################################
## ADJUSTABLE PARAMETERS                                            ##
#####################################################################

set name                NAME
set temperature         310
set path                        PATH1
coordinates             $path/output/$name-pre-slow-heated-cpt.coor
structure               $path/base/ready_$name.psf

## Getting the periodic cell parameters from the previous run
extendedSystem         $path/output/$name-pre-slow-heated-cpt.xsc
firsttimestep          0

#####################################################################
## SIMULATION PARAMETERS                                            ##
#####################################################################

## Input
paratypecharmm          on
parameters              $path/base/PARM1
## NOTE: Do not set the initial velocity temperature if you
## have also specified a .vel restart file!
temperature             $temperature
# Force-Field Parameters
exclude                 scaled1-4
1-4scaling              1.0
COMmotion               no
margin                  0.0
# (...cont) approximations
switching               on
switchdist              12.0
cutoff                  14.0
pairlistdist            22.0
pairlistsPerCycle       2
outputPairlists         5
# Integrator Parameters - Varios ... shake, Hs
timestep                1.0  # in fs
stepspercycle           20
nonbondedFreq           2
rigidBonds              water
rigidDieOnError         on
##useSettle on
```

```
# Constant Temperature Control
langevin                on
langevinTemp            $temperature
langevinFile            $path/tmp/LGV-coef.pdb
langevinCol             B
##langevinDamping       10.0 # Applied to all atoms
##langevinHydrogen        on
wrapAll                 on
# PME (for full-system periodic electrostatics)
PME yes
PMEGridSizeX            48 # grid-pts along cellBasisVector1.
PMEGridSizeY            54 # along cellBasisVector2.
PMEGridSizeZ            48 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.

## Center-of-mass motion is automatically removed.
## Constant Pressure Control (variable volume)
## Nose-Hoover Langevin Piston for pressure coupling
useGroupPressure        yes # needed for rigidBonds
useFlexibleCell         no
useConstantArea         no
langevinPiston          on
langevinPistonTarget    1.01325 # in bar -> 1 atm
langevinPistonPeriod    200.
langevinPistonDecay     100.
langevinPistonTemp      $temperature

## Outputs
outputname              $path/output/$name-pre-lgv
binaryoutput            no
DCDfile                 $path/output/$name-pre-lgv-tray.dcd
DCDUnitCell             yes
DCDfreq                 500
velDCDfile              $path/output/$name-pre-lgv-vel.dcd
velDCDfreq              500
XSTfile                 $path/output/$name-pre-lgv-pcell.xst
XSTfreq                 500

## Standard output INFO
outputEnergies          500
outputPressure          500
outputMomenta           500
outputTiming            500

## Restart output files
restartname             $path/output/$name-pre-lgv-restart
restartfreq             500
```

```
restartsave             yes
binaryrestart           yes # Preserves more accuracy

#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################

## Harmonic constraints
if {1} {
constraints             on
consexp                 2
consref                 $path/tmp/reference-atoms-lgv.pdb
conskfile               $path/tmp/spring-atoms-lgv.pdb
conskcol                B
constraintScaling       1.0
}
#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

## When restarting the number of efective steps MUST be (numsteps -
firsttimestep)
numsteps                500  # (2 000 000 - 0) -> 2 ns

#############################################################
## END OF CONFIG FILE                                      ##
## Updated by Vladimir Espinosa - Nov 2007                 ##
## to run in MNOSTRUM - ZGZ                                ##
#############################################################
```

automation/templates/pre-restraining.template

```
#############################################################
## Adecuate using RESTRAINED MINIMIZATION under NAMD       ##
## THIRD PREPARATION STEP                                  ##
## ADEQUATE PROTEIN - EVOLVING HARMONIC CONSTRAINS         ##
## Author: S.Cuesta Lopez - Oct. 2006                      ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#############################################################


#############################################################
## Updated and modified by Vladimir Espinosa               ##
## University of Zaragoza - Nov. 2007                      ##
#############################################################

#############################################################
```

```
## PREPARATION PHASE - NAMD                                    ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007                   ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## by Vladimir Espinosa                                        ##
################################################################

################################################################
## ADJUSTABLE PARAMETERS                                       ##
################################################################

set name                NAME
set index               I
set path                        PATH1

coordinates             $path/output/COOR1
structure               $path/base/ready_$name.psf

################################################################
## SIMULATION PARAMETERS                                       ##
################################################################

# Force-Field Parameters
paratypecharmm          on
parameters              $path/base/PARM1
exclude                 scaled1-4
1-4scaling              1.0
COMmotion               no
margin                  0.0
stepspercycle           20

# Integrator Parameters - Varios ... shake, Hs
rigidBonds              water
rigidDieOnError         on
nonbondedFreq           2
##useSettle               on

## Getting the periodic cell parameters from the previous run
extendedSystem          $path/output/XSC1

# NOTE: Do not set the periodic cell basis if you have also
# specified an .xsc restart file!
# Periodic Boundary Conditions

wrapAll                 on

# PME (for full-system periodic electrostatics)
PME yes
```

```
PMEGridSizeX           48 # grid-pts along cellBasisVector1.
PMEGridSizeY           54 # along cellBasisVector2.
PMEGridSizeZ           48 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.


# (...cont) approximations
switching              on
switchdist             12.0
cutoff                 14.0
pairlistdist           22.0
pairlistsPerCycle      2
outputPairlists        5

## Outputs
outputname             $path/output/$name-pre-restrained.$index

binaryoutput           no

## Standard output INFO
outputEnergies         100
outputTiming           100



#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################

## Harmonic constraints
constraints            on
consexp                2
consref                $path/tmp/reference-atoms.pdb
conskfile              $path/tmp/spring-atoms.pdb
conskcol               B
constraintScaling      1.0

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Minimization
minimization            on
numsteps                500
```

automation/templates/pre-slow-heating-cpt.template

```
#############################################################
## INITIAL GLOBAL EQUILIBRATION AND HEATING under NAMD     ##
## FOURTH PREPARATION STEP                                 ##
## ADEQUATE PROTEIN - CPT EQUILIBRATION [NO CONSTRAINS]    ##
## Author: S.Cuesta Lopez - Oct. 2006                      ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#############################################################

#############################################################
## Updated and modified by Vladimir Espinosa               ##
## University of Zaragoza - Nov. 2007                      ##
#############################################################

#############################################################
## PREPARATION PHASE - NAMD                                ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007               ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## by Vladimir Espinosa                                    ##
#############################################################

#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

set name               NAME
set path                       PATH1
coordinates            $path/output/$name-pre-unrestrained.coor
structure              $path/base/ready_$name.psf

set temperature        10.0
reassignFreq           50
reassignTemp           $temperature
reassignIncr           30
reassignHold           310

firsttimestep          0

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################
# Force-Field Parameters
paratypecharmm         on
parameters             $path/base/PARM1
exclude                scaled1-4
1-4scaling             1.0
COMmotion              no
margin                 0.0
```

```
# Integrator Parameters - Varios ... shake, Hs
rigidBonds              water
rigidDieOnError         on
##useSettle              on

## Getting the periodic cell parameters from the previous run
extendedSystem          $path/output/$name-pre-unrestrained.xsc

wrapAll                 on

# PME (for full-system periodic electrostatics)
PME yes
PMEGridSizeX            48 # grid-pts along cellBasisVector1.
PMEGridSizeY            54 # along cellBasisVector2.
PMEGridSizeZ            48 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.

# (...cont) approximations
switching               on
switchdist              12.0
cutoff                  14.0
pairlistdist            22.0
pairlistsPerCycle       2
outputPairlists         5

## Center-of-mass motion is automatically removed.


# Integrator Parameters - Varios ... shake, Hs (...cont)
timestep                1.0  # in fs
stepspercycle           20
nonbondedFreq           2
temperature             $temperature

# Constant Temperature Control
langevin                on
langevinTemp            $temperature
langevinDamping         10.0 # Applied to all atoms
langevinHydrogen        on

## Constant Pressure Control (variable volume)
## Nose-Hoover Langevin Piston for pressure coupling
useGroupPressure        yes # needed for rigidBonds
useFlexibleCell         no
useConstantArea         no
```

```
langevinPiston         on
langevinPistonTarget   1.01325 # in bar -> 1 atm
langevinPistonPeriod   200.
langevinPistonDecay    100.
langevinPistonTemp     $temperature

## Outputs
outputname             $path/output/$name-pre-slow-heated-cpt

binaryoutput           no
DCDfile                $path/output/$name-pre-slow-heated-cpt-ray.dcd
DCDUnitCell            yes
DCDfreq                500
velDCDfile             $path/output/$name-pre-slow-heated-cpt-vel.dcd
velDCDfreq             500
XSTfile                $path/output/$name-pre-slow-heated-cpt-pcel-
l.xst
XSTfreq                500

## Standard output INFO
outputEnergies         500
outputTiming           500

## Restart output files
restartname            $path/output/$name-pre-slow-heated-cpt-restart

restartfreq            500
restartsave            yes
binaryrestart          yes # Preserves more accuracy


#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

## Here we run a 30-step loop to set up a temperature ramp with a
dT=10
## each step is 100ps-long
for { set temperature 10 } { $temperature < 310 } { incr temperature
30 } {
run 50
reinitvels $temperature
langevinTemp $temperature
}
```

automation/templates/pre-unrestraining.template

```
#############################################################
## Adecuate using RESTRAINED MINIMIZATION under NAMD       ##
## THIRD PREPARATION STEP                                  ##
## ADEQUATE PROTEIN - EVOLVING HARMONIC CONSTRAINS         ##
## Author: S.Cuesta Lopez - Oct. 2006                      ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#############################################################

#############################################################
## Updated and modified by Vladimir Espinosa               ##
## University of Zaragoza - Nov. 2007                      ##
#############################################################

#############################################################
## PREPARATION PHASE - NAMD                                ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007               ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## by Vladimir Espinosa                                    ##
#############################################################

#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

set name               NAME
set index              I
set path                       PATH1

coordinates            $path/output/$name-pre-restrained.$index.coor
structure              $path/base/ready_$name.psf



#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################

# Force-Field Parameters
paratypecharmm         on
parameters             $path/base/PARM1
exclude                scaled1-4
1-4scaling             1.0
COMmotion              no
margin                 0.0
stepspercycle          20

# Integrator Parameters - Varios ... shake, Hs
```

```
rigidBonds              water
rigidDieOnError         on
nonbondedFreq           2
##useSettle              on

## Getting the periodic cell parameters from the previous run
extendedSystem         $path/output/$name-pre-restrained.$index.xsc

wrapAll                 on

# PME (for full-system periodic electrostatics)

PME yes
PMEGridSizeX            48 # grid-pts along cellBasisVector1.
PMEGridSizeY            54 # along cellBasisVector2.
PMEGridSizeZ            48 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.


# (...cont) approximations
switching               on
switchdist              12.0
cutoff                  14.0
pairlistdist            22.0
pairlistsPerCycle       2
outputPairlists         5

## Outputs
outputname              $path/output/$name-pre-unrestrained

binaryoutput            no

## Standard output INFO
outputEnergies          100
outputTiming            100
#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Minimization
minimization             on
numsteps                 500
```

automation/templates/first.template

```
#############################################################
## Concatenate & Continue previous runs NAMD               ##
```

```
## Running LANGEVIN DYNAMICS INH. DAMPING                    ##
## Author: S.Cuesta Lopez - Oct. 2006                        ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE ##
#############################################################

#############################################################
## Updated and modified by Vladimir Espinosa                ##
## University of Zaragoza - Nov. 2007                       ##
#############################################################

#############################################################
## Updated and modified by Guillermo Gutierrez              ##
## University of Zaragoza - Feb. 2013                       ##
#############################################################

#############################################################
## PRODUCTION PHASE - NAMD                                  ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007                ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## TO MAKE CONTINOUS RUNS FROM PREVIOUS RESTARTING          ##
## FILES GENERATED IN MNOSTRUM - BCNA (Vladimir Espinosa)   ##
#############################################################

#############################################################
## ADJUSTABLE PARAMETERS                                    ##
#############################################################

set name               NAME
set path                       PATH1
set temperature        310

coordinates            $path/base/$name-pre-lgv.coor
structure              $path/base/ready_$name.psf

## Getting the periodic cell parameters from the previous run
extendedSystem         $path/base/$name-pre-lgv.xsc

firsttimestep          0

#############################################################
## SIMULATION PARAMETERS                                    ##
#############################################################

## Input
paratypecharmm         on
parameters             $path/base/PARM1
```

```
## NOTE: Do not set the initial velocity temperature if you
## have also specified a .vel restart file!
temperature            $temperature

# Force-Field Parameters
exclude                scaled1-4
1-4scaling             1.0
COMmotion              no
margin                 0.0
# (...cont) approximations
switching              on
switchdist             12.0
cutoff                 14.0
pairlistdist           22.0
pairlistsPerCycle      2
outputPairlists        5

# Integrator Parameters - Varios ... shake, Hs
timestep               1.0  # in fs
stepspercycle          20
nonbondedFreq          2
rigidBonds             water
rigidDieOnError        on
##useSettle on

# Constant Temperature Control
langevin               on
langevinTemp           $temperature
langevinFile           LGV-coef.pdb
langevinCol            B
##langevinDamping      10.0 # Applied to all atoms
##langevinHydrogen       on

wrapAll                on

# PME (for full-system periodic electrostatics)
PME yes
PMEGridSizeX           48 # grid-pts along cellBasisVector1.
PMEGridSizeY           54 # along cellBasisVector2.
PMEGridSizeZ           48 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.


## Center-of-mass motion is automatically removed.

## Constant Pressure Control (variable volume)
## Nose-Hoover Langevin Piston for pressure coupling
```

```
useGroupPressure       yes # needed for rigidBonds
useFlexibleCell        no
useConstantArea        no

langevinPiston         on
langevinPistonTarget   1.01325 # in bar -> 1 atm
langevinPistonPeriod   200.
langevinPistonDecay    100.
langevinPistonTemp     $temperature


## Outputs
outputname             $path/output/$name-prod-lgv
binaryoutput           no
CDfile                $path/output/$name-prod-lgv-tray.dcd
DCDUnitCell            yes
DCDfreq                500
velDCDfile             $path/output/$name-prod-lgv-vel.dcd
velDCDfreq             500
XSTfile                $path/output/$name-prod-lgv-pcell.xst
XSTfreq                5000

## Standard output INFO
outputEnergies         500
outputPressure         500
outputMomenta          500
outputTiming           500

## Restart output files
restartname            $path/output/$name-prod-lgv-restart
restartfreq            500
restartsave            yes
binaryrestart          yes # Preserves more accuracy


#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################

## Harmonic constraints
constraints            on
consexp                2
consref                reference-atoms-lgv.pdb
conskfile              spring-atoms-lgv.pdb
conskcol               B
constraintScaling      1.0
```

```
#####################################################################
## EXECUTION SCRIPT                                                 ##
#####################################################################

## When restarting the number of efective steps MUST be (numsteps -
firsttimestep)
numsteps               500 #2E6->2ns

## !!!!THE .dcd FILE GENERATED MUST BE CONCATENATED WITH THE PREVIOUS
ONE WITH catdcd

#####################################################################
## END OF CONFIG FILE                                               ##
## Updated by Vladimir Espinosa - Nov 2007                          ##
## to run in MNOSTRUM - ZGZ                                         ##
#####################################################################
```

automation/templates/restart.template

```
#####################################################################
## Concatenate & Continue previous runs NAMD                        ##
## Running LANGEVIN DYNAMICS INH. DAMPING                           ##
## Author: S.Cuesta Lopez - Oct. 2006                               ##
## Copyright S.Cuesta Lopez . ENS-Lyon 2006. LAB. PHYSIQUE          ##
#####################################################################

#####################################################################
## Updated and modified by Vladimir Espinosa                        ##
## University of Zaragoza - Nov. 2007                               ##
#####################################################################

#####################################################################
## Updated and modified by Guillermo Gutierrez                      ##
## University of Zaragoza - Feb. 2013                               ##
#####################################################################

#####################################################################
## PRODUCTION PHASE - NAMD                                          ##
## ADECUATED TO MNOSTRUM - BCNA - March 2007                        ##
## (...cont) AND NOW ADECUATED TO MNOSTRUM - ZGZ - NOV 2007##
## TO MAKE CONTINOUS RUNS FROM PREVIOUS RESTARTING                  ##
## FILES GENERATED IN MNOSTRUM - BCNA (Vladimir Espinosa)           ##
#####################################################################

#####################################################################
## ADJUSTABLE PARAMETERS                                            ##
```

```
#############################################################


set name                NAME
set path          PATH1
set restartpoint        RESTARTPOINT1
set temperature         310
set tsteps             TSTEPS1
set index         INDEX1

## Continuing a job from the restart files
## COMMENT THE TEMPERATURE DEFINITION LINE IF YOU USE THIS BLOCK!!!!
structure         $path/base/ready_$name.psf
coordinates             $path/base/$name-pre-lgv.coor
set inputname           $path/output/$name-prod-lgv
binCoordinates          $inputname-restart.$restartpoint.coor
binVelocities           $inputname-restart.$restartpoint.vel
extendedSystem         $inputname-restart.$restartpoint.xsc

firsttimestep           $restartpoint

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################

## Input
paratypecharmm          on
parameters              $path/base/PARM1

# Force-Field Parameters
exclude                 scaled1-4
1-4scaling              1.0
COMmotion               no
margin                  0.0
# (...cont) approximations
switching               on
switchdist              12.0
cutoff                  14.0
pairlistdist            22.0
pairlistsPerCycle       2
outputPairlists         5

# Integrator Parameters - Varios ... shake, Hs
timestep                1.0  # in fs
stepspercycle           20
nonbondedFreq           2
rigidBonds              water
```

```
rigidDieOnError        on
##useSettle on

# Constant Temperature Control
langevin               on
langevinTemp           $temperature
langevinFile           LGV-coef.pdb
langevinCol            B
##langevinDamping      10.0 # Applied to all atoms
##langevinHydrogen       on

wrapAll                on

# PME (for full-system periodic electrostatics)
PME yes
PMEGridSizeX           48 # grid-pts along cellBasisVector1.
PMEGridSizeY           54 # along cellBasisVector2.
PMEGridSizeZ           48 # along cellBasisVector3
# use numbers with small integer factors: 2,3,5.


## Center-of-mass motion is automatically removed.

## Constant Pressure Control (variable volume)
## Nose-Hoover Langevin Piston for pressure coupling

useGroupPressure       yes # needed for rigidBonds
useFlexibleCell        no
useConstantArea        no

langevinPiston         on
langevinPistonTarget   1.01325 # in bar -> 1 atm
langevinPistonPeriod   200.
langevinPistonDecay    100.
langevinPistonTemp     $temperature


## Outputs
outputname             $path/output/$name-prod-lgv-rst-$index

binaryoutput           no
DCDfile                $path/output/$name-prod-lgv-rst-$index-
tray.dcd
DCDUnitCell            yes
DCDfreq                500
velDCDfile             $path/output/$name-prod-lgv-rst-$index-vel.dcd
velDCDfreq             500
```

```
XSTfile                 $path/output/$name-prod-lgv-rst-$index-pcel-
l.xst
XSTfreq                 500

## Standard output INFO
outputEnergies          500
outputPressure          500
outputMomenta           500
outputTiming            500

## Restart output files
restartname             $path/output/$name-prod-lgv-restart
restartfreq             500
restartsave             yes
binaryrestart           yes # Preserves more accuracy


#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################

## Harmonic constraints
constraints             on
consexp                 2
consref                 $path/tmp/reference-atoms-lgv.pdb
conskfile               $path/tmp/spring-atoms-lgv.pdb
conskcol                B
constraintScaling       1.0

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

## When restarting the number of efective steps MUST be (numsteps -
firsttimestep)
numsteps                $tsteps #20000000  # (20 000 000) -> 20 ns

## !!!!THE .dcd FILE GENERATED MUST BE CONCATENATED WITH THE PREVIOUS
$index-lgv-prod-tray.dcd
## FILE WITH /gpfs/apps/VMD/vmd-1.8.5/32/plugins/catdcd TO GET THE
FINAL 10ns TRAJECTORY!!!!!!

#############################################################
## END OF CONFIG FILE                                      ##
## Updated by Vladimir Espinosa - Nov 2007                 ##
## to run in MNOSTRUM - ZGZ                                ##
#############################################################
```

SMD/templates/template.job

```
#!/bin/bash
#$ -N [SATOM].[VINDEX].[RINDEX]
#$ -e output/error.log
#$ -pe openmpi16 16
#$ -q BIFIZCAM
#$ -cwd
#$ -m bea
#$ -M guillermo.munchkin@gmail.com
# @ initialdir = .
# @ gpus_per_node = 2
# @ cpus_per_task = 1
# @ total_tasks = 12

#source ../config.cfg
#source ../scripts/functions.bash
inpfile=[INPFILE1]
#Setting the environment to run with the 2.9 version
module load openmpi/gcc/64/1.4.4
module load namd/2.9
COMMAND="mpirun -np $NSLOTS namd2"
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Starting
[$inpfile]" >> output/status.log
echo "Running from $( pwd )" >> output/status.log
$COMMAND inps/$inpfile > output/LOG/$( echo $inpfile | sed -e
"s/inp/log/" )
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Finishing
[$inpfile]" >> output/status.log
echo >> output/status.log
```

SMD/templates/template_16.job

```
#!/bin/bash
#$ -N [SATOM].[VINDEX].[RINDEX]
#$ -e output/error.log
#$ -pe openmpi16 16
#$ -q BIFIZCAM
#$ -cwd
#$ -m bea
#$ -M guillermo.munchkin@gmail.com
# @ initialdir = .
# @ gpus_per_node = 2
# @ cpus_per_task = 1
# @ total_tasks = 12
```

```
#source ../config.cfg
#source ../scripts/functions.bash
inpfile=[INPFILE1]
#Setting the environment to run with the 2.9 version
module load openmpi/gcc/64/1.4.4
module load namd/2.9
COMMAND="mpirun -np $NSLOTS namd2"
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Starting
[$inpfile]" >> output/status.log
echo "Running from $( pwd )" >> output/status.log
if [ -e   "$logfile" ]; then
  echo "Archivo    de recuperación    de $logfile" >>    $logfile.-
bak
  echo "fecha: $( date )" >> $logfile.bak
  echo >> $logfile.bak
  cat $logfile >> $logfile.bak
  echo >> $logfile.bak
fi
$COMMAND inps/$inpfile > output/LOG/$( echo $inpfile | sed -e
"s/inp/log/" )
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Finishing
[$inpfile]" >> output/status.log
echo >> output/status.log
```

SMD/templates/template_32.job

```
#!/bin/bash
#$ -N [SATOM].[VINDEX].[RINDEX]
#$ -e output/error.log
#$ -pe openmpi32 32
#$ -q BIFIZCAM
#$ -cwd
#$ -m bea
#$ -M guillermo.munchkin@gmail.com
# @ initialdir = .
# @ gpus_per_node = 2
# @ cpus_per_task = 1
# @ total_tasks = 12

#source ../config.cfg
#source ../scripts/functions.bash
inpfile=[INPFILE1]
#Setting the environment to run with the 2.9 version
module load openmpi/gcc/64/1.4.4
module load namd/2.9
COMMAND="mpirun -np $NSLOTS namd2"
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Starting
```

```
[$inpfile]" >> output/status.log
echo "Running from $( pwd )" >> output/status.log
if [ -e   "$logfile" ]; then
  echo "Archivo    de recuperación    de $logfile" >>    $logfile.-
bak
  echo "fecha: $( date )" >> $logfile.bak
  echo >> $logfile.bak
  cat $logfile >> $logfile.bak
  echo >> $logfile.bak
fi
$COMMAND inps/$inpfile > output/LOG/$( echo $inpfile | sed -e
"s/inp/log/" )
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Finishing
[$inpfile]" >> output/status.log
echo >> output/status.log
```

SMD/templates/reloaded.job

```
#!/bin/bash
#$ -N [SATOM].[VINDEX].[RINDEX]
#$ -e output/error.log
#$ -pe openmpi32 32
#$ -q BIFIZCAM
#$ -cwd
#$ -m bea
#$ -M guillermo.munchkin@gmail.com
# @ initialdir = .
# @ gpus_per_node = 2
# @ cpus_per_task = 1
# @ total_tasks = 12

#source ../config.cfg
#source ../scripts/functions.bash
inpfile=[INPFILE1]
#Setting the environment to run with the 2.9 version
module load openmpi/gcc/64/1.4.4
module load namd/2.9
COMMAND="mpirun -np $NSLOTS namd2"
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Starting
[$inpfile]" >> output/status.log
echo "Running from $( pwd )" >> output/status.log
if [ -e   "$logfile" ]; then
  echo "Archivo    de recuperación    de $logfile" >>    $logfile.-
bak
  echo "fecha: $( date )" >> $logfile.bak
  echo >> $logfile.bak
  cat $logfile >> $logfile.bak
```

```
  echo >> $logfile.bak
fi
$COMMAND inps/$inpfile > output/LOG/$( echo $inpfile | sed -e
"s/inp/log/" )
echo "[$( date +%d/%m/%y ) $( date +%H:%M:%S )] - Finishing
[$inpfile]" >> output/status.log
echo >> output/status.log
```

SMD/templates/reloaded.inp

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Steered Molecular Dynamic of
# Holo-Flavodoxin (1FLV) in a Water Box


#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

set iname           NAME2
set name            NAME1
set satom           SATOM1
set vindex          VINDEX1
set rindex          RINDEX1
set sk              K1
set stps            STEPS1
structure           ../base/ready_$iname.psf
coordinates         ../base/PDB/$iname.$rindex.pdb
bincoordinates      ../base/COOR/$iname.$rindex.coor

set temperature     310
set svel            SVEL1
set output          ../output/

## Getting the periodic cell parameters from the previous run
extendedSystem      ../base/XSC/$iname.$rindex.xsc

# Continuing a job from the restart files

binVelocities          ../base/VEL/$iname.$rindex.vel  ;# COMMENT THE
TEMPERATURE DEFINITION LINE IF YOU USE THIS BLOCK!!!!


firsttimestep       0
```

```
#####################################################################
## SIMULATION PARAMETERS                                           ##
#####################################################################

# Input
paraTypeCharmm      on
parameters          ../base/par_all27_prot_na_fmn.prm


# NOTE: Do not set the initial velocity temperature if you
# have also specified a .vel restart file!
# temperature         $temperature

# Force-Field Parameters
exclude              scaled1-4
1-4scaling           1.0
cutoff               14.0
switching            on
switchdist           12.0
pairlistdist         22.0
pairlistsPerCycle    2
outputPairlists      5
margin               0.0
# Integrator Parameters
timestep             1.25       ; #
rigidBonds           water
rigidDieOnError      on
#rigidTolerance       0.0005
nonbondedFreq        2
fullElectFrequency   2
stepspercycle        20

# Constant Temperature Control
langevin              on
langevinTemp          $temperature
langevinFile          ../base/LGV-coef.pdb
langevinCol           B
##langevinDamping       5      ; # damping coefficient (gamma) of /ps
##langevinHydrogen      off    ; # don't couple langevin bath to hy-
drogens
wrapAll              on
wrapWater            on

# PME (for full-system periodic electrostatics)
PME                  yes
```

```
PMEGridSizeX           48 # grid-pts along cellBasisVector1.
PMEGridSizeY           54 # along cellBasisVector2.
PMEGridSizeZ           48 # along cellBasisVector3

# Constant Pressure Control (variable volume)
useGroupPressure      yes ; # needed for rigidBonds
useFlexibleCell       no
useConstantArea       no
langevinPiston        on
langevinPistonTarget  1.01325 ; #  in bar -> 1 atm
langevinPistonPeriod  200.0
langevinPistonDecay   100.0
langevinPistonTemp    $temperature
# Output
outputName            ../output/$name.$satom.$vindex.$rindex
binaryoutput          no
DCDfile               ../output/TRAY/$name.$satom.$vindex.$rindex-
tray.dcd
DCDUnitCell           yes
DCDfreq               5000
velDCDfile            ../output/VEL/$name.$satom.$vindex.$rindex-
vel.dcd
velDCDfreq            5000
XSTfile               ../output/XST/$name.$satom.$vindex.$rindex-
pcell.xst
XSTFreq               5000

## Restart output files
restartname            ../output/RESTARTS/$name.$satom.$vindex.
$rindex-restart

restartfreq            1250000     ;# 1000steps = every 1ps
restartsave            yes
binaryrestart          yes  # Preserves more accuracy

## Standard output INFO
outputEnergies        5000
outputPressure        5000
outputMomenta         5000
outputTiming          5000

# Fixed Atoms Constraints (set PDB beta-column to 1)
fixedAtoms           on
fixedAtomsForces     on
fixedAtomsFile       ../base/REF/$iname.$rindex.$satom.ref
fixedAtomsCol        B
```

```
#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################
SMD                 on
SMDFile             ../base/REF/$iname.$rindex.$satom.ref
SMDk                $sk
SMDVel              $svel
SMDDir              SDIR1
SMDOutputFreq       500

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Execution
run $stps ;   #
```

SMD/templates/smd_continue.inp

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Steered Molecular Dynamic of
# Holo-Flavodoxin (1FLV) in a Water Box

#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

set iname           NAME2
set cname           NAME3
set name            NAME1
set satom           SATOM1
set vindex          VINDEX1
set rindex          RINDEX1
set sk              K1
set stps            STEPS1
set fststep         FIRSTSTEP1
structure           ../base/ready_$iname.psf
coordinates         ../base/PDB/$iname.$rindex.pdb
bincoordinates      ../base/continue/$cname.coor
set temperature     310
set svel            SVEL1
set output          ../output/
## Getting the periodic cell parameters from the previous run
extendedSystem      ../base/continue/$cname.xsc
```

```
# Continuing a job from the restart files
binVelocities          ../base/continue/$cname.vel  ;# COMMENT THE
TEMPERATURE DEFINITION LINE IF YOU USE THIS BLOCK!!!!
firsttimestep       $fststep

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################

# Input
paraTypeCharmm      on
parameters          ../base/par_all27_prot_na_fmn.prm
# NOTE: Do not set the initial velocity temperature if you
# have also specified a .vel restart file!
# temperature         $temperature
# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              14.0
switching           on
switchdist          12.0
pairlistdist        22.0
pairlistsPerCycle   2
outputPairlists     5
margin              0.0

# Integrator Parameters
timestep            1.25       ; #
rigidBonds          water
rigidDieOnError     on
#rigidTolerance       0.0005
nonbondedFreq       2
fullElectFrequency  2
stepspercycle       20

# Constant Temperature Control
langevin             on
langevinTemp         $temperature
langevinFile         ../base/LGV-coef.pdb
langevinCol          B
##langevinDamping      5      ; # damping coefficient (gamma) of
5/ps
##langevinHydrogen     off    ; # don't couple langevin bath to hy-
drogens
wrapAll             on
wrapWater           on
```

```
# PME (for full-system periodic electrostatics)
PME                    yes
PMEGridSizeX            48 # grid-pts along cellBasisVector1.
PMEGridSizeY            54 # along cellBasisVector2.
PMEGridSizeZ            48 # along cellBasisVector3

# Constant Pressure Control (variable volume)
useGroupPressure      yes ; # needed for rigidBonds
useFlexibleCell       no
useConstantArea       no
langevinPiston        on
langevinPistonTarget  1.01325 ; #  in bar -> 1 atm
langevinPistonPeriod  200.0
langevinPistonDecay   100.0
langevinPistonTemp    $temperature
# Output
outputName            ../output/$name.$satom.$vindex.$rindex
binaryoutput          no
DCDfile               ../output/TRAY/$name.$satom.$vindex.$rindex-
tray.dcd
DCDUnitCell           yes
DCDfreq               5000
velDCDfile            ../output/VEL/$name.$satom.$vindex.$rindex-
vel.dcd
velDCDfreq            5000
XSTfile               ../output/XST/$name.$satom.$vindex.$rindex-
pcell.xst
XSTFreq               5000

## Restart output files
restartname            ../output/RESTARTS/$name.$satom.$vindex.
$rindex-restart

restartfreq            1000000     ;# 1000steps = every 1ps
restartsave            yes
binaryrestart          yes # Preserves more accuracy

## Standard output INFO
outputEnergies        5000
outputPressure        5000
outputMomenta         5000
outputTiming          5000

# Fixed Atoms Constraints (set PDB beta-column to 1)
fixedAtoms           on
fixedAtomsForces     on
fixedAtomsFile       ../base/REF/$iname.$rindex.$satom.ref
```

```
fixedAtomsCol        B

#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################
SMD                  on
SMDFile              ../base/REF/$iname.$rindex.$satom.ref
SMDk                 $sk
SMDVel               $svel
SMDDir               SDIR1
SMDOutputFreq        500

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Execution
run $stps ;   #
```

SMD/templates/smd_doublestep.inp

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Steered Molecular Dynamic of
# Holo-Flavodoxin (1FLV) in a Water Box


#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

set iname            NAME2
set name             NAME1
set satom            SATOM1
set vindex           VINDEX1
set rindex           RINDEX1
set sk               K1
set stps             STEPS1
structure            ../base/ready_$iname.psf
coordinates          ../base/PDB/$iname.$rindex.pdb
bincoordinates       ../base/COOR/$iname.$rindex.coor

set temperature      310
set svel             SVEL1
set output           ../output/
```

```
## Getting the periodic cell parameters from the previous run
extendedSystem      ../base/XSC/$iname.$rindex.xsc

# Continuing a job from the restart files

binVelocities          ../base/VEL/$iname.$rindex.vel  ;# COMMENT THE
TEMPERATURE DEFINITION LINE IF YOU USE THIS BLOCK!!!!

firsttimestep       0

#####################################################################
## SIMULATION PARAMETERS                                           ##
#####################################################################

# Input
paraTypeCharmm      on
parameters          ../base/par_all27_prot_na_fmn.prm
# NOTE: Do not set the initial velocity temperature if you
# have also specified a .vel restart file!
# temperature         $temperature
# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              14.0
switching           on
switchdist          12.0
pairlistdist        22.0
pairlistsPerCycle   2
outputPairlists     5
margin              0.0

# Integrator Parameters
timestep            1.25       ; #
rigidBonds          water
rigidDieOnError     on
#rigidTolerance       0.0005
nonbondedFreq       2
fullElectFrequency  2
stepspercycle       20
# Constant Temperature Control
langevin              on
langevinTemp          $temperature
langevinFile          ../base/LGV-coef.pdb
langevinCol           B
##langevinDamping       5      ; # damping coefficient (gamma) of
5/ps
##langevinHydrogen      off    ; # don't couple langevin bath to hy-
```

```
drogens
wrapAll               on
wrapWater             on
# PME (for full-system periodic electrostatics)
PME                   yes
PMEGridSizeX           48 # grid-pts along cellBasisVector1.
PMEGridSizeY           54 # along cellBasisVector2.
PMEGridSizeZ           48 # along cellBasisVector3
# Constant Pressure Control (variable volume)
useGroupPressure      yes ; # needed for rigidBonds
useFlexibleCell       no
useConstantArea       no
langevinPiston        on
langevinPistonTarget  1.01325 ; #  in bar -> 1 atm
langevinPistonPeriod  200.0
langevinPistonDecay   100.0
langevinPistonTemp    $temperature
# Output
outputName            ../output/$name.$satom.$vindex.$rindex
binaryoutput          no
DCDfile               ../output/TRAY/$name.$satom.$vindex.$rindex-
tray.dcd
DCDUnitCell           yes
DCDfreq               5000
velDCDfile            ../output/VEL/$name.$satom.$vindex.$rindex-
vel.dcd
velDCDfreq            5000
XSTfile               ../output/XST/$name.$satom.$vindex.$rindex-
pcell.xst
XSTFreq               5000
## Restart output files
restartname            ../output/RESTARTS/$name.$satom.$vindex.
$rindex-restart
restartfreq            1000000     ;# 1000steps = every 1ps
restartsave            yes
binaryrestart          yes # Preserves more accuracy
## Standard output INFO
outputEnergies        5000
outputPressure        5000
outputMomenta         5000
outputTiming          5000
# Fixed Atoms Constraints (set PDB beta-column to 1)
fixedAtoms           on
fixedAtomsForces     on
fixedAtomsFile       ../base/REF/$iname.$rindex.$satom.ref
fixedAtomsCol        B
```

```
#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################

SMD                 on
SMDFile             ../base/REF/$iname.$rindex.$satom.ref
SMDk                $sk
SMDVel              $svel
SMDDir              SDIR1
SMDOutputFreq       500

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Execution
run $stps ;   #
```

SMD/templates/smd_general.inp

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Steered Molecular Dynamic of
# Holo-Flavodoxin (1FLV) in a Water Box

#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

set name            NAME1
set satom           SATOM1
set vindex          VINDEX1
set rindex          RINDEX1
set sk              K1
set stps            STEPS1
structure           ../base/ready_$name.psf
coordinates         ../base/PDB/$name.$rindex.pdb
bincoordinates      ../base/COOR/$name.$rindex.coor

set temperature     310
set svel            SVEL1
set output          ../output/

## Getting the periodic cell parameters from the previous run
extendedSystem      ../base/XSC/$name.$rindex.xsc
```

```
# Continuing a job from the restart files

binVelocities          ../base/VEL/$name.$rindex.vel  ;# COMMENT THE
TEMPERATURE DEFINITION LINE IF YOU USE THIS BLOCK!!!!

firsttimestep      0

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################

# Input
paraTypeCharmm     on
parameters          ../base/par_all27_prot_na_fmn.prm

# NOTE: Do not set the initial velocity temperature if you
# have also specified a .vel restart file!
# temperature         $temperature

# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              14.0
switching           on
switchdist          12.0
pairlistdist        22.0
pairlistsPerCycle   2
outputPairlists     5
margin              0.0

# Integrator Parameters
timestep            1.0       ; # 1fs/step
rigidBonds          water
rigidDieOnError     on
#rigidTolerance       0.0005
nonbondedFreq       2
fullElectFrequency  2
stepspercycle       20

# Constant Temperature Control
langevin              on
langevinTemp          $temperature
langevinFile          ../base/LGV-coef.pdb
langevinCol           B
##langevinDamping       5      ; # damping coefficient (gamma) of
5/ps
```

```
##langevinHydrogen      off    ; # don't couple langevin bath to hydrogens
wrapAll               on
wrapWater             on

# PME (for full-system periodic electrostatics)
PME                   yes
PMEGridSizeX           48 # grid-pts along cellBasisVector1.
PMEGridSizeY           54 # along cellBasisVector2.
PMEGridSizeZ           48 # along cellBasisVector3

# Constant Pressure Control (variable volume)
useGroupPressure      yes ; # needed for rigidBonds
useFlexibleCell       no
useConstantArea       no
langevinPiston        on
langevinPistonTarget  1.01325 ; #  in bar -> 1 atm
langevinPistonPeriod  200.0
langevinPistonDecay   100.0
langevinPistonTemp    $temperature

# Output
outputName            ../output/$name.$satom.$vindex.$rindex
binaryoutput          no
DCDfile               ../output/TRAY/$name.$satom.$vindex.$rindex-tray.dcd
DCDUnitCell           yes
DCDfreq               5000
velDCDfile            ../output/VEL/$name.$satom.$vindex.$rindex-vel.dcd
velDCDfreq            5000
XSTfile               ../output/XST/$name.$satom.$vindex.$rindex-pcell.xst
XSTFreq               5000

## Restart output files
restartname            ../output/RESTARTS/$name.$satom.$vindex.$rindex-restart
restartfreq            1000000     ;# 1000steps = every 1ps
restartsave            yes
binaryrestart          yes # Preserves more accuracy

## Standard output INFO
outputEnergies        5000
outputPressure        5000
outputMomenta         5000
outputTiming          5000
```

```
# Fixed Atoms Constraints (set PDB beta-column to 1)
fixedAtoms           on
fixedAtomsForces     on
fixedAtomsFile       ../base/REF/$name.$rindex.$satom.ref
fixedAtomsCol        B

#####################################################################
## EXTRA PARAMETERS                                                ##
#####################################################################
SMD                 on
SMDFile             ../base/REF/$name.$rindex.$satom.ref
SMDk                $sk
SMDVel              $svel
SMDDir              SDIR1
SMDOutputFreq       1000

#####################################################################
## EXECUTION SCRIPT                                                ##
#####################################################################

# Execution
run $stps ;   #
```