LECTURE 14
CHARMM Course
24-Jan-2006

CHARMM

Chemistry at **HAR**vard **M**olecular **M**echanics

Fourteenth Lecture:

Replica Exchange Molecular Dynamics

Review dynamc.doc

Topics

Overview of Concepts and Applications
MMTSB interface to CHARMM
A general *ad hoc* interface

Richard M Venable

with grateful acknowledgement to

Paul Maragakis and Martin Spichty

Replica Exchange Molecular Dynamics (REMD)

Replica exchange, also known as parallel tempering, is an emerging method which is being increasingly applied as a conformational sampling tool; the results of a literature search for 'replica exchange' over the past 2 years are included as a bibliography at the end of this document. REMD commonly utilizes multiple simultaneous simulations at different temperatures, although other system properties may be (and have been) used. Initially used in the context of Monte Carlo sampling, it was extended to molecular dynamics by Okamoto and Sugita via an expression to scale the momenta

$$p' = \sqrt{\frac{T_{new}}{T_{old}} p}$$

after a swap of configurations between two simulations at 2 different temperatures. The decision to swap or not is essentially the standard Metropolis criterion used in Monte Carlo sampling

$$p = 1 \;;\; \Delta \leq 0$$
$$p = \exp(-\Delta) \;;\; \Delta > 0$$

$$\Delta = \left(\frac{1}{kTi} - \frac{1}{kTj}\right)(Ej - Ei)$$

where typically *j* is step *i+1* for an MC move; for REMD, *j* is normally T bath *i+1* at the same step.

The general concept is that as configurations move from bath to bath, conformational changes that are easily possible at higher T will migrate into the lower T baths, improving the sampling of states found in the low T baths. A number of published studies confirm the sampling efficiency of the REMD method; it can sample systems that would normally be completely outside the timescale of MD simulations at the target (low) T. It offers some advantages over MC sampling, as the configurations evolve more naturalisticly via MD, with little chance of creating impossible conformations. It also generates a Boltzmann ensemble of conformations with the same potential energy distribution as a straight MD simulation at the same temperature. The simulations are, of course, discontinuous at the swap points, so that one cannot evaluate time dependent properties.

Usually the lowest T bath is the one of interest; the question remains of how to choose the T values for the remaining baths, and how many T baths to use. One criterion is that the potential energy ranges at each T must overlap with the bath above and below it; the degree of overlap determines the swap acceptance probability. There is not complete agreement over the optimum accepance probability, but values in the range of 0.2 to 0.4 are generally regarded as the most useful. Another factor is resources—REMD requires a fair number of processors (although it is embarassingly parallel), and generous amounts of disk storage; this can have some impact on the choice of the number of baths. One must at times strike a balance between the highest T desired and the number of processors available.

The REMD method has been applied via most of the common macromolecular simulation packages (including CHARMM) at this point, often via an external interface (*e.g.* the MMTSB). Conformational sampling of proteins and peptides, *esp.* with an implicit solvent model, has become quite popular. The strength of the method, however, is that it can be applied with condensed phase systems just as easily, *i.e.* a protein in explicit solvent, or a peptide in a membrane bilayer. The sampling efficiency of REMD suggests it will be an effective tool for evaluating force fields, as a couple of published studies have already done.

The MMTSB in Brief

The CL Brooks group at Scripps has developed a set of **perl** utilities for a number of modeling tasks, mostly protein related, including a tool for performing replica exchange simulations with either CHARMM or AMBER. Information on the MMTSB can be found at http://mmtsb.scripps.edu/ along with the tool set itself. The tool set is also described in the paper "MMTSB Tool Set: enhanced sampling and multiscale modeling methods for applications in structural biology", Michael Feig, John Karanicolas, Charles L. Brooks III,*Journal of Molecular Graphics and Modelling* 22 (2004) 377–395. There is some support through a mailing list (mmtsb@scripps.edu), and a dedicated forum at http://www.charmm.org/ is also available. The tool for replica exchange is named `aarex.pl;` a syntax listing and usage examples follow.

aarex.pl

```
    aarex.pl [options] [files]
    options:     [-n runs]
        [-par initruns=value,equilruns=value,
          [no]save,savebestfreq=value,archive
          ensmode=add|replace,natpdb=file,psf=file]
        [-temp nwin:min:max]
        [-condfile file]
        [-f listfile]
        [-mdpar CHARMMparams]
        [-mdopt [no]trajout,[no]restout,[no]conslim,
          limforce=value,limsel=ca|cb|cab|heavy]
        [-l refPDB min:max[=min:max …]]
        [-cons [ca|cb|cab|heavy] ref|self min:max[_force][=…]]
        [-opt optionsfile]
        [-custom setup|pre|post[:init|equi|prod] file]
        [-dir workdir]
        [-ens tag] [-ensdir dir]
        [PARALLELoptions]
        [-log file] [-elog file] [-charmmlog file]
```

This script is used to run replica exchange simulations. In most parallel environments it will start the replica exchange server automatically. The options –n, -par, -temp, -condfile, -ens, -ensdir are available as in rexserver.pl.

Alternatively, it can connect to an external replica exchange server if its address, port, and ID are given with –rserv. In that case –jobs may be used to run only some of the temperature windows instead of all windows if multiple clients are launched on different machines.

For shared memory environments the option –cpus can be used to specify a smaller number of CPUs than temperature windows if necessary due to computational restraints. A host file can be given with –hosts for automatic remote submission in a distributed environment. (see calcprop.pl for a more detailed explanation).

In order to contact the server from an external program (e.g. for monitoring purposes) the server ID is required. The option –saveid is available to write this server information to a file.

The intial input PDB files for each MD simulation are expected either as the last command line arguments or from an external file that is given with –f.

The remaining parameters are used to control the MD simulations. Parameters that can be given with –mdpar are shown here.

The option –l is available to provide a list of residues and a template PDB structure for loop modeling. Please note that with this option RMSD values that are automatically calculated if a reference PDB structure is given are also limited to only these residues.

Further options given with –mdopt control whether a trajectory or restart file is written out ([no]trajout and [no]restout), how many MD steps are used (default: 1) for averaging the energy score used in the replica exchange Metropolis criteria (avgener), whether the rest of protein is restrained outside the loop ([no]conslim), and the maximum restraint force (limforce) and type of restrained atoms (limsel) in this case. Other restraints may be specified with –cons.

The simulation protocol can be further customized by providing CHARMM commands through external files. This can be done with –custom which expects a keyword and a file name as arguments. The keyword is used to specify when the custom command sequence should be inserted in the standard protocol. If setup is used the commands will be sent to CHARMM only once during the initial setup phase. If the keywords pre or post the commands will be executed before or after the dynamics command, respectively. These keywords can be further qualified with :init, :equi, and :prod corresponding to the replica exchange cycle modes to allow custom equilibration protocols.

The option –log is available to request a server log file. A CHARMM log file is generated for each client if –charmmlog is given, an energy log file is generated with –elog.

EXAMPLES

```
aarex.pl –n 8 –mdpar dynsteps=100,param=22,gb,nocut –par
initruns=2,equilruns=2,natpdb=1vii.exp.pdb –temp 4:298:400
1vii.sample.{1,2,3,4}.pdb
```

runs 8 replica exchange MD simulation cycles with four exponentially spaced temperature windows from 298 to 400K. The first 2 runs are considered initialization runs, the next 2 runs are equilibration runs. The native PDB structure is given as reference for calculation RMSD values. MD parameters are set to run 100 steps for each cycle, use CHARMM22 parameters with GB implicit solvent and no electrostatic cutoffs. Initial conformations are taken from the files 1vii.sample.?.pdb.

```
aarex.pl –n 4
```

runs 4 additional cycles continuing a previous replica exchange simulation run

```
aarex.pl –n 5 –par initruns=2,equilruns=0,nosave –temp 4:298:400 –ensdir data –
ens rex –mdpar dynsteps=200,gb,nocut 1vii.sample.{1,2,3,4}.pdb
```

runs 5 replica exchange simulation runs with 2 initialization and no equilibration runs. The conformation from the lowest temperature at each run is saved under the rex tag in an ensemble in the directory data. No other conformations during the simulation are saved.

```
aarex.pl –n 6 –par initruns=2,equilruns=2,natpdb=1vii.exp.pdb –temp 4:298:400 –
cpus 2 –mdpar dynsteps=200,gb –l 1vii.exp.pdb 10:21 –mdopt
conslim,limforce=5.0,limsel=ca –log server.log –charmmlog charmm.log –f
init.files
```

runs a replica exchange simulation runs for loop modeling. The loop residues are located at 10 through 21, the rest of the protein is restrained at C-alpha atoms with a force constant of up to

5 kcal/mol. This run has 4 temperature windows but uses only 2 CPUs at a time. A server log file and for each client a CHARMM log file are written out. The intial input files are taken from init.files.

```
aarex.pl –n 8 –mdpar dynsteps=100,gb,nocut –par initruns=2,equilruns=2 –temp
4:298:400 –hosts sgi.workstations 1vii.sample.{1,2,3,4}.pdb
```

runs a replica exchange simulation across distributed workstations

```
aarex.pl –n 8 –mdpar dynsteps=100,gb,nocut –par initruns=2,equilruns=2 –temp
4:298:400 –mp –hosts sgi.local 1vii.sample.1.pdb
```

runs a replica exchange simulation across distributed workstations with local directories. All temperature windows are started from the same initial file.

```
aarex.pl –n 10 –mdpar dynsteps=100,gb,nocut –par
initruns=2,equilruns=0,natpdb=1vii.exp.pdb –condfile conditions
1vii.sample.{1,2,3,4}.pdb
```

replica exchange simulation with all replicas at the same temperature but with different radius of gyration umbrella potentials

```
aarex.pl –n 10 –mdpar dynsteps=100,gb –par
initruns=2,equilruns=0,natpdb=1vii.exp.pdb –condfile conditions –log server.log
–charmmlog charmm.log 1vii.sample.{1,2,3,4}.pdb
```

2D replica exchange with two temperatures and two different radius of gyration umbrellas

```
aarex.pl –charmmlog clog –n 8 –mdpar dynsteps=100,param=19,nogb –custom setup
acesetup.inp –custom pre ace.inp –par
initruns=2,equilruns=0,natpdb=1vii.exp.pdb –temp 4:298:350 1vii.sample.2.pdb
```

runs a replica exchange simulation with a customized potential function. Through the files acesetup.inp and ace.inp the ACE solvation model is used instead of the default Generalized Born model.

REQUREMENTS

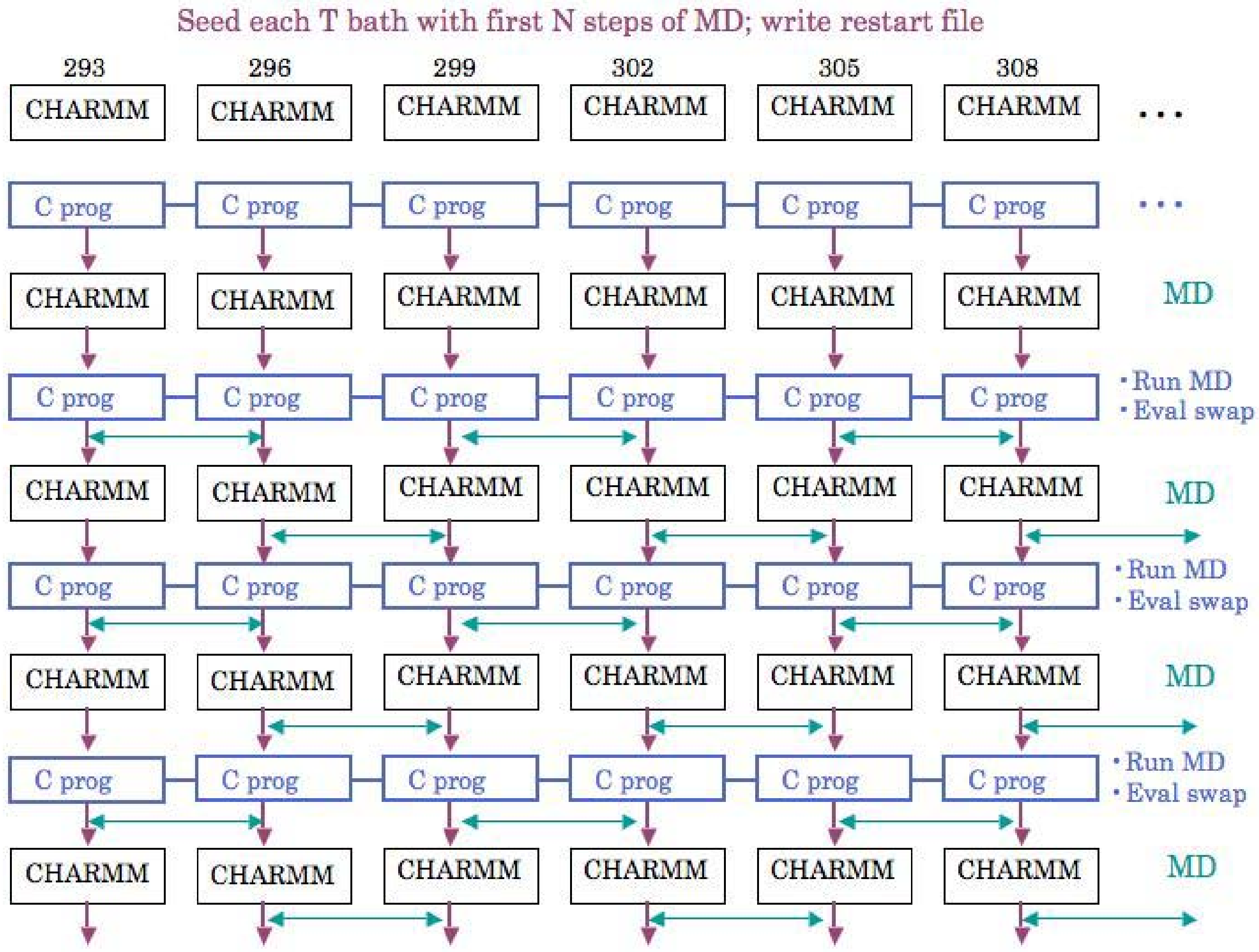
Server.pm ReXServer.pm Client.pm ReXClient.pm GenUtil.pm Molecule.pm Ensemble.pm CHARMM.pm

AUTHORS

Michael Feig, Brooks group, TSRI
John Karanicolas, Brooks group, TSRI

An *ad hoc* interface to CHARMM for REMD

This section describes a more general interface to CHARMM for replica exchange, using C and Python programs as key components, and taking advantage of the dynamics restart capabilities. Unlike the MMTSB, some editing of the programs and scripts is needed as part of the setup, and the C program must be compiled; however, the interface is fairly flexible for these same reasons. The programs and scripts will be presented in the form used to perform some test REMD simulations on gel phase bilayers of the lipid DMPC. The C and Python code are from the authors of "Optimal estimates of free energies from multi-state nonequilibrium work data", Paul Maragakis, Martin Spichty, and Martin Karplus, submitted to *Phys Rev Lett*, (2006).


Seed each T bath with first N steps of MD; write restart file
293
296
299
302
305
308
CHARMM
C prog
MD
• Run MD
• Eval swap
. . .

As indicated in the above flow chart, the C program controls the synchronization of the MD simulations and swaps, and calls various other programs (including CHARMM). After each MD run, the final potential energy is used by a Python program (`rex.py`) to evaluate the Metropolis criterion; for a swap, the CHARMM restart files are exchanged, and the velocities are scaled based on the T change. From dynamc.doc, the feature used to scale the velocities is:

```
ISCALE      0     This option is to allow the user to scale the velocities
                  by a factor SCALE at the beginning of a restart run.
                  This may be useful in changing the desired temperature.
                  .eq. 0  no scaling done (usual input value)
                  .ne. 0  scale velocities by SCALE.
                  WARNING:
                  Please use this option only when you are changing the
                  temperature of the run.

SCALE      1.     Scale factor for the previous option.
```

The value of SCALE is 1.0 unless a swap occurs; for a swap, the value computed in rex.py from the T ratio is used. The values are communicated to CHARMM by a formatted file, which is read via the STREAM command.

As noted earlier, REMD is embarassingly parallel; it is ideally suited to Linux clusters, as there is no communication between processes except via files between the short MD runs, and each process is self contained. Parallel MD, on the other hand, must exchange a fair amount of data between all processes on every integration step, *esp.* if particle-mesh Ewald (PME) summation is being used. It's simplest to use one processor for each T bath, although it is certainly possible and reasonable (but more complicated) to use both processors in a dual-processor compute node. However, for a fixed number of nodes, it may often be better to have twice as many T baths than to double the speed.

The following listing outlines the steps required to use this collection of scripts and programs to perform REMD with CHARMM:

- Choose the number of T baths and the spacing between baths; some iterative tests may be needed to get the optimum spacing and to maximize the T range for a fixed number of baths.
- Decide how many MD steps to perform between swaps, and how many swap steps for a particular run; again, some initial tests may be needed.
- Based on these choices, modify the `Tbaths.csh` script to produce a list of the T bath values chosen; this will help ensure consistency across the various components (programs and scripts).
- Edit the files `replica_exchange.c` and `rex.py` so that their respective arrays of T bath values are equivalent to the list produced by the `Tbaths.csh` script; this is the only change needed for the `rex.py` program.

- Edit the file `replica_exchange.c` to reflect the number of T baths chosen, the number of MD steps, and the number of swap steps; only the C program has (and needs) all of this information. Compile the C program using **mpicc**, as indicated directly above the source listing later in this document.
- Create a link named "charmm" in the working directory that points to a single threaded (non-parallel) version of CHARMM; on Biowulf use something like `ln –s /usr/local/charmm/c31b2/pgi-med.one charmm`
- Edit `rex_start.inp, rex.inp`, and `init.str` based on the number of MD steps and the actual files (RTF, PARAM, PSF, COOR, etc.) that will be used; note that at least one coord set must be written to the .trj file.
- Modify `rex_start.csh` to run CHARMM as appropriate for the cluster in use; the example is Biowulf specific (the 'qcharmm' command), and submits the initial CHARMM runs needed to "seed" the replica exchange process.
- Edit the `run_rex.csh` script to reflect the number of T baths chosen; it may also need to be modified for some other queuing system besides PBS, or for the location of the MPICH installation.
- Edit the `rexdone.csh` script based on the number of swap steps; the last 2 output and restart files are processed by this after the replica exchange run.
- Review the `cleantemp.csh` script; it is currently designed to be called every 50 swap steps from the replica_exchange program, performing: [1] deletion of unneeded restart files, [2] gzip compression and relocation of output files, and [3] merge of the trajectories into larger files and deletion of the originals (via `merge.inp`). *Without these steps, one can easily become overwhelmed by the number of files produced and the disk space used.*
- Run the `rex_start.csh` script; besides the initial MD run, it also sets up the subdirs for each T bath, and the first restart stream files for each bath.
- Run the `post_start.csh` script; this does some cleanup of files and links created by the `rex_start.csh` script, and completes the setup phase.
- Launch the `run_rex.csh` script to start the replica exchange process.
- Run the `rexdone.csh` script to cleanup the last 2 files for each T bath.

Program and Script Listings

The C program that follows manages the REMD simulation, coordinating the execution of the various components via MPI. When all baths have completed a run, it calls rex.py to evaluate swaps, and calls cleantemp.csh every 50 swap steps to help manage file proliferation.

```
% set path = ( /usr/local/mpich/bin $path )
% mpicc replica_exchange.c -o replica_exchange
```

replica_exchange.c

```
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* C code by Paul Maragakis; changes for subdir per T bath by Rick Venable */
/* MPI management program that runs everything; edit array T, compile with mpicc */
/* counter for swaps may require change; also, value of final timestep */

int main (int argc, char *argv[])
{
      int rank, size, i, j, k;
      char cmd[1000];
        int temp;
        int T[] =
{293,296,299,302,305,308,311,314,317,320,323,326,329,332,335,338,341,344,347,350,353,35
6,359,362,365,368,371,374,377,380};

      MPI_Init( &argc, &argv );
      MPI_Comm_rank( MPI_COMM_WORLD, &rank);
      MPI_Comm_size( MPI_COMM_WORLD, &size);
/* adjust for the number of processors; may be 2x or 4x no. of T baths */
        if (size!=30) {
            printf("You have to use 30 procs\n");
            exit(1);
        }
      printf( "Running process %d of %d\n", rank, size );

/* counter i is the number of swap steps (rex_start.csh does step 1) */
/* the MPI process no. (rank) determines the bath T for this process */
        for (i=2; i<=1000; i++) {
           temp=T[rank];
/* build CHARMM command line, pluck final ENER from dynamics in rex.inp */
/* note the 'sync' command; assures files will be up to date (NFS issue) */
           sprintf (cmd, "./charmm tact:%d i:%d < rex.inp > %d/rex%d.out ; grep \"DYNA>
500\" %d/rex%d.out | awk '{print $6;}' > %d.ene; sync\n", temp, i, temp, i, temp, i,
temp);
           system(cmd);
           MPI_Barrier( MPI_COMM_WORLD );
           if (rank==0) {
               sprintf (cmd, "sync; ./rex.py %d >> swap.log", i);
               system(cmd);
           }
           MPI_Barrier( MPI_COMM_WORLD );
/* every 50 steps merge all the previous files, other cleanup */
           if ((i)%50==0) {
               sprintf(cmd,"./cleantemp.csh %d %d > /dev/null\n",temp,i);
               system(cmd);
           }
        }

        MPI_Finalize();
        return 0;
}
```

rex.py

```
#!/usr/bin/python

# this python prog evaluates and manages the actual swaps
# original code by Martin Spichty
# modified by Rick Venable to use a subdir for each T bath
# only the temp= line needs to be changed

from random import random
import math
import sys

# the temperature baths
temp=[293,296,299,302,305,308,311,314,317,320,323,326,329,332,335,338,341,344,347,350,353,356,359,362,365,368,371,374,377,380]
data={}

for tact in temp:
  name=str(tact) + '.ene'
  f=open(name)
  data[(tact,'ene')]=float(f.readline())
  f.close()

# write restart stream file assuming no swaps
for i in range(len(temp)):
   tact=temp[i]
   scale=1.0
   name=str(tact) + '.str'
   g=open(name,'w')
   g.write('* stream\n')
   g.write('*\n')
   towrite = 'set oldrest ' + str(tact) + '/rex.res.@{J}'
   g.write(towrite + '\n')
   towrite = 'set sfactor ' + str(scale)
   g.write(towrite + '\n')
   g.write('return\n')
   g.close()

# Randomly pick a starting point at
# either bath 0, or bath 1
r=random()
if r > 0.5:
  next=0
else:
  next=1

# For all the consecutive pairs of baths starting
# from the starting point we picked before
for i in range(next,len(temp)-1,2):
    tact=temp[i]
    tnext=temp[i+1]
    b1=1/(8.314*float(tact))
    b2=1/(8.314*float(tnext))
    delta=4184*(b1-b2)*(float(data[(tnext,'ene')])-float(data[(tact,'ene')]))
    # This is the odds ratio for the swap move
    prob=math.exp(-delta)
    r=random()
    if prob > r:
      # If we accept the swap move, replace restart stream files
      # create the stream file for the forward swap
      scale=math.sqrt(float(tact)/float(tnext))
      name=str(tact) + '.str'
      g=open(name,'w')
      g.write('* stream\n')
      g.write('*\n')
      towrite = 'set oldrest ' + str(tnext) + '/rex.res.@{J}'
      g.write(towrite + '\n')
      towrite = 'set sfactor = ' + str(scale)
      g.write(towrite + '\n')
      g.write('return\n')
      g.close()

      # create the stream file for the reverse swap
      scale=math.sqrt(float(tnext)/float(tact))
      name=str(tnext) + '.str'
      g=open(name,'w')
      g.write('* stream\n')
      g.write('*\n')
      towrite = 'set oldrest ' + str(tact) + '/rex.res.@{J}'
      g.write(towrite + '\n')
      towrite = 'set sfactor = ' + str(scale)
      g.write(towrite + '\n')
      g.write('return\n')
      g.close()

      # create a simple log file with information about
      # accepted switches
      towrite= 'the following baths were switched:'
      print towrite,tact,tnext,sys.argv[1]
      towrite= 'e1,e2,probability: '
      print towrite,data[(tact,'ene')],data[(tnext,'ene')],prob
```

The above Python program computes the swap odds ratio based on the potential energy for the final MD step, and makes the accept or reject decision. The T baths are considered as non-overlapping adjacent pairs, starting with either the lowest T bath or the next lowest; a uniform random number is compared to 0.5 in order to choose the starting point. For the 30 baths in the example, there are either 15 swaps (start from lowest) or 14 swaps (next lowest) considered at each swap evaluation step. The program also prepares a special restart stream file, which instructs CHARMM (when it is read by rex.inp) to either continue with the restart file at the same T, or to swap restart files and scale the velocities. Finally, it prints a record of the swaps, which are appended to a file named swap.log (see the C program).

Shell (csh) scripts

Tbaths.csh

```
#!/bin/csh

# provide a list of the T bath values for other scripts
# for non-uniform spacing use a “foreach” loop instead of “while”
# space separated by default; comma (,) as arg 1 uses , separator
@ t = 293
if ( $1 == ',' ) then
 while ( $t <= 380 )
  echo -n "$t,"
  @ t += 3
 end
 echo ''
else
 while ( $t <= 380 )
  echo -n "$t "
  @ t += 3
 end
endif
```

rex_start.csh

```
#!/bin/csh

# N.B. 'qcharmm' is NIH Biowulf specific; it combines a CHARMM cover script
# with an interface to PBS qsub to submit and run a CHARMM input script
# see URL http://biowulf.nih.gov/apps/charmm/index.html

# T bath values must match those in replica_exchange.c, rex.py

foreach tact ( `./Tbaths.csh` )

  if ( ! -d $tact ) mkdir $tact
  if ( ! -e $tact/rex.res.1 ) then
   if ( ! -e start$tact.inp ) ln -s rex_start.inp start$tact.inp
   @ s = $tact * `date '+%S%M%H'`
   qcharmm proc=2 prfx=start$tact tact:${tact} seed:$s
   sleep 1
  endif
  echo "* temp $tact" > ${tact}.str
  echo "*" >> ${tact}.str
  echo "set oldrest "${tact}"/rex.res.@{J}" >> ${tact}.str
  echo "set sfactor 1.0" >> ${tact}.str
  echo "return" >> ${tact}.str

end
```

post_start.csh

```
#!/bin/csh
# cleanup after rex_start
foreach tact ( `./Tbaths.csh` )
 echo -n "$tact "
 mv start$tact.out $tact/rex1.out
 rm start$tact.inp
end
echo ''
```

run_rex.csh

```
#!/bin/csh
#PBS -N ReplicaEx
#PBS -j oe
#PBS -m ae

# same path for mpirun as for mpicc; -np N must match no. of baths
set path = ( /usr/local/mpich/bin $path )

cd $PBS_O_WORKDIR
mpirun -machinefile $PBS_NODEFILE -np 30 ./replica_exchange
```

cleantemp.csh

```
#!/bin/csh
# utility script, called from replica_exchange prog; cleanup files
# arg 1 is T value, arg 2 is swap step counter (multiple of 50)
set t = $1
@ j = $2 - 49

pushd $t >& /dev/null
sync
if ( ! -e Out ) mkdir Out
# list of unprocessed files
ls -1tr rex.res.* > r.t
set ff = `head -1 r.t | cut -d. -f3`
set lf = `tail -1 r.t | cut -d. -f3`
@ lf = $lf - 2
@ i = $ff
# compress output, move to subdir; wipe restart
while ( $i <= $lf )
 gzip rex$i.out
 mv rex$i.out.gz Out
 rm rex.res.$i
 @ i += 1
end
popd >& /dev/null

# merge .trj files; orig removed via CLOSE w. DISP DELETE
./charmm < merge.inp TACT:$t I:$j >> $t/merge.out
```

rexdone.csh

```
#!/bin/csh

# final .out, .res cleanup
# N.B. the script Tbaths.csh makes a space separated list of T values

foreach t ( `./Tbaths.csh` )
 pushd $t >& /dev/null
 echo -n "$t "
 @ i = 999
# compress output, move to subdir; wipe restart
 while ( $i <= 1000 )
  gzip rex$i.out
  mv rex$i.out.gz Out
  @ i += 1
 end
 rm rex.res.999
 gzip rex.res.1000
 popd >& /dev/null
end
echo ''
```

CHARMM scripts

rex_start.inp

```
* setup for the parallel tempering simulation; 0.5 ps runs
*

!BOMBLEV -4

stream init.str

open write formatted unit 3 name @{TACT}/rex.res.1
open write unformatted unit 2 name @{TACT}/rex.trj.1

shake bonh param fast

dyna cpt strt nstep 500 timestep 0.001 echeck 9999. iseed @SEED -
    pcons pint pref 1.0 pmass 0. pgamma 10.0 -
    hoover reft @TACT tmass 10000. -
    inbfrq -1 atom vatom cutnb 15.0 ctofnb 10. cdie eps 1. -
    ctonnb 8. vswitch cutim 15.0 imgfrq -1 wmin 1.0 -
    ewald pmew fftx 48 ffty 48 fftz 64  kappa .33 spline order 6 -
    iprfrq 500 ihtfrq 0 ieqfrq 0 ntrfrq 100 -
    iuncrd 2 iunrea -1 iunwri 3 kunit -1 -
    nprint 100 nsavc 500 nsavv 0 ihbfrq 0 -
    firstt @TACT finalt @TACT tstruct @TACT teminc 0.0 -
    iasors 1 iasvel 1 iscvel 0 ichecw 0

stop
```

rex.inp

```
* parallel tempering simulation single trajectory; 0.5 ps
*

stream init.str
calc k = @i - 2
calc j = @i - 1
stream @{tact}.str

open read formatted unit 13 name @oldrest
open write unformatted unit 2 name @{TACT}/rex.trj.@{I}
open write formatted unit 3 name @{TACT}/rex.res.@{I}

shake bonh param fast
dyna cpt rest nstep 500 timestep 0.001 echeck 999. scale @SFACTOR  -
    pcons pint pref 1.0 pmass 0. pgamma 10.0 -
    hoover reft @TACT tmass 10000. -
    inbfrq -1 atom vatom cutnb 15.0 ctofnb 10. cdie eps 1. -
    ctonnb 8. vswitch cutim 15.0 imgfrq -1 wmin 1.0 -
    ewald pmew fftx 48 ffty 48 fftz 64  kappa .33 spline order 6 -
    iprfrq 500 ihtfrq 0 ieqfrq 0 ntrfrq 100 -
    iuncrd 2 iunrea 13 iunwri 3 kunit -1 -
    nprint 100 nsavc 500 nsavv 0 ihbfrq 0 -
    firstt @TACT finalt @TACT tstruct @TACT teminc 0.0 -
    iasors 1 iasvel 1 iscvel 0 ichecw 0 iscale 1

stop
```

init.str

```
* C27r protein-lipid parameters; initial psf and coor; crystal setup
*

open unit 1 read card name ../protlpd27.rtf
read rtf unit 1 card
close unit 1
open unit 2 read card name ../protlpd27r.prm
read param unit 2 card
close unit 2

open unit 2 read card name ../gelaw1.psf
read psf  card unit 2
close unit 2
open unit 2 read card name ../gel963.crd
read coor card unit 2
close unit 2

stream ../crys963.str
crystal define tetr @boxx @boxy @boxz 90. 90. 90.
open unit 23 read card name ../box.cry
crystal read unit 23 card
close unit 23
image byres
return
```

merge.inp

```
* merge of parallel tempering simulation bath @TACT ; 50 files, first @I
*

stream init.str

set nunit 50
set offs  20
calc end = @i + @nunit - 1
calc first = @offs + 1

set j @i
set cc 1
label looptraj
calc unit = @offs + @cc
open read unform unit @unit name @tact/rex.trj.@j
incr cc by 1
incr j by 1
if cc .le. @nunit goto looptraj

open write unform unit @offs name @tact/mrg.trj.@end
merge first @first nunit @nunit outp @offs

set cc 1
label clostraj
calc unit = @offs + @cc
close unit @unit disp delete
incr cc by 1
if cc .le. @nunit goto clostraj

stop
```

293.str; continuation

```
* stream
*
set oldrest 293/rex.res.@{J}
set sfactor 1.0
return
```

302.str; swap with 299

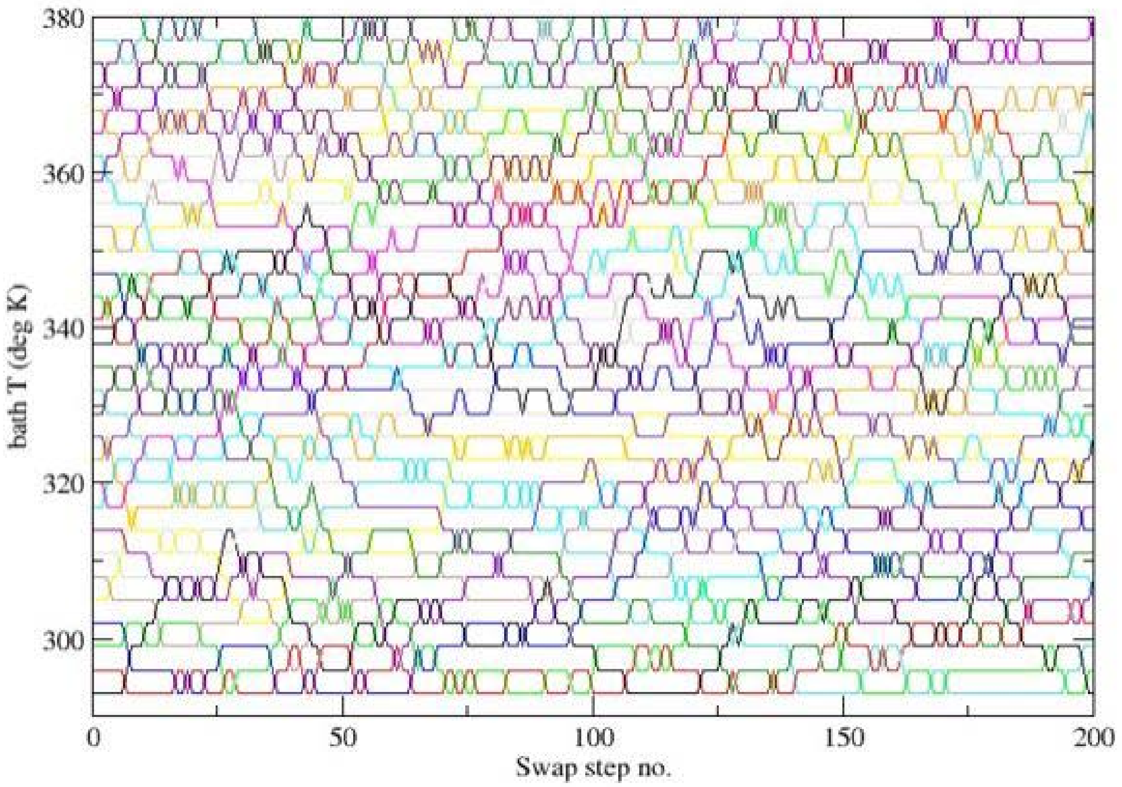
```
* stream
*
set oldrest 299/rex.res.@{J}
set sfactor = 1.00500420139
return
```

DMPC gel replica echange test run

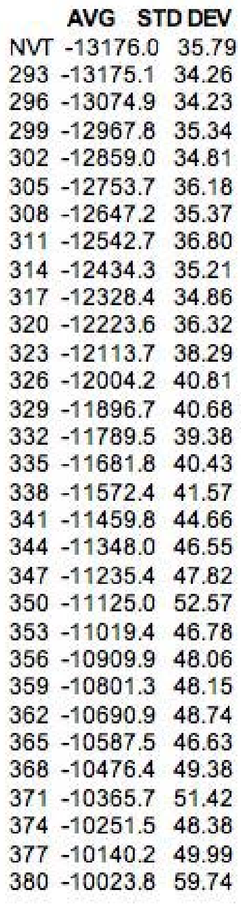
As noted earlier, the program and script examples are for a DMPC gel with 30 T baths, which start at 293 K and extend to 380 K, with a 3 deg spacing. Each NVT ensemble MD run was 500 steps at a 1 fs integration step size, or 0.5 ps; 1000 swap steps were performed, for a nominal run time of 500 ps. Using a single processor per T bath on Biowulf, this took a few days to complete. The swap efficiency is indicated below:

All a(swap) = .368

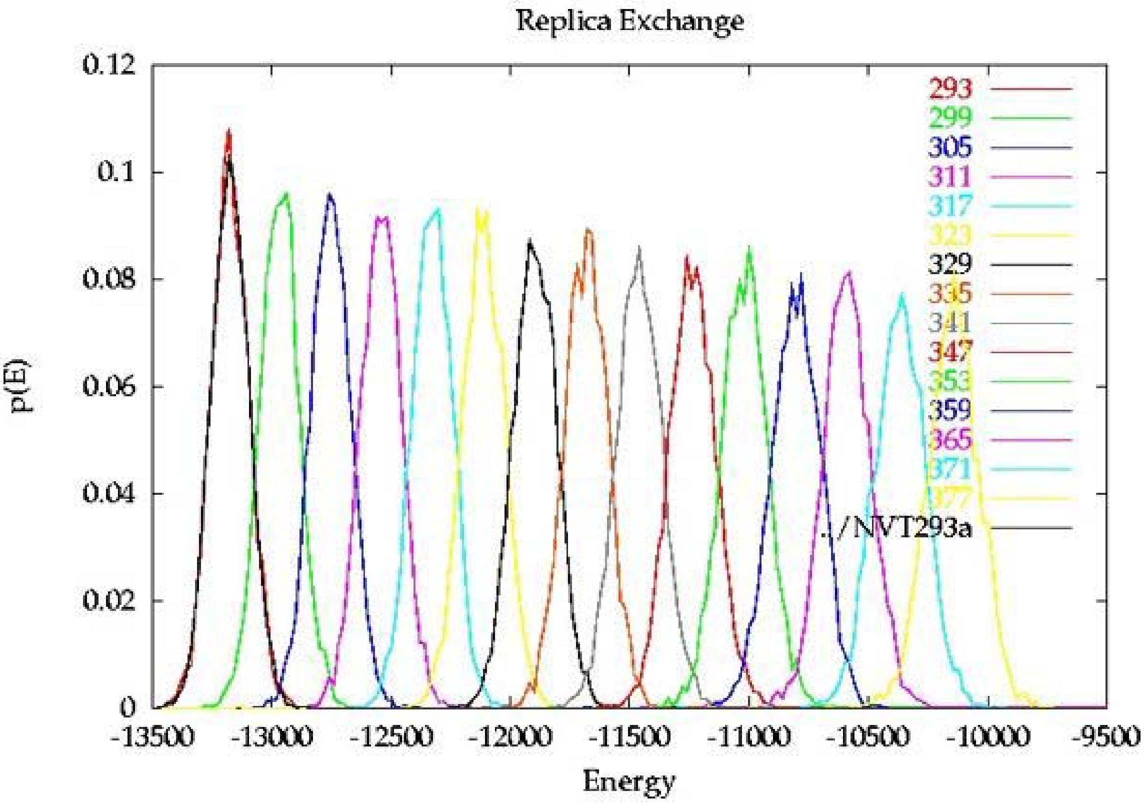
T	a(swap)
293	0.330
296	0.315
299	0.308
302	0.317
305	0.302
308	0.305
311	0.323
314	0.311
317	0.334
320	0.341
323	0.319
326	0.347
329	0.375
332	0.385
335	0.388
338	0.370
341	0.382
344	0.398
347	0.389
350	0.424
353	0.409
356	0.385
359	0.397
362	0.399
365	0.420
368	0.424
371	0.421
374	0.424
377	0.418
380	0.410



The above plot shows the bath swaps for the first 200 steps; the acceptance rates indicate good efficency for the 3 deg spacing used, and suggest the spacing between baths could larger. The steady increase in the acceptance rate at higher T values also suggests that the spacing be increased a bit for the higher T ranges, *i.e.* it need not be uniform, consistent with published studies. To further emphasize this, the potential energy statistics and plots of the distributions at a 6 deg spacing are shown in the following listing and plot.



	AVG	STD DEV
NVT	-13176.0	35.79
293	-13175.1	34.26
296	-13074.9	34.23
299	-12967.8	35.34
302	-12859.0	34.81
305	-12753.7	36.18
308	-12647.2	35.37
311	-12542.7	36.80
314	-12434.3	35.21
317	-12328.4	34.86
320	-12223.6	36.32
323	-12113.7	38.29
326	-12004.2	40.81
329	-11896.7	40.68
332	-11789.5	39.38
335	-11681.8	40.43
338	-11572.4	41.57
341	-11459.8	44.66
344	-11348.0	46.55
347	-11235.4	47.82
350	-11125.0	52.57
353	-11019.4	46.78
356	-10909.9	48.06
359	-10801.3	48.15
362	-10690.9	48.74
365	-10587.5	46.63
368	-10476.4	49.38
371	-10365.7	51.42
374	-10251.5	48.38
377	-10140.2	49.99
380	-10023.8	59.74



The overlap of these distributions appears to be fairly good at the 6 deg spacing, further suggesting that the 3 deg spacing was too conservative for this system. Note that, consistent with the increase in acceptance with increasing T values, the RMS energy fluctuations (STD DEV) show a similar trend with increasing T values. (*Author's note*: I had planned to show some convergence tests and comparisons to longer NPAT MD simulations of the DMPC gel, but uncovered a technical flaw during the preparation of this lecture.) Based on the above evaluation (and technical problem), the REMD run will be repeated using a modified spacing between T baths, with 5 deg for baths 1-10, 6 deg for baths 11-20, and 7 deg for baths 21-30, with a maximum of 468 K. The new Tbaths.csh script looks like

Tbaths.csh

```
#!/bin/csh
# provide a list of the T bath values for other scripts
foreach t ( 293 298 303 308 313 318 323 328 333 338 \
            344 350 356 362 368 374 380 386 392 398 \
            405 412 419 426 433 440 447 454 461 468 )
 if ( $1 == ',' ) then
  echo -n "$t,"
 else
  echo -n "$t "
 endif
end
if ( $1 == ',' ) echo ‘’
```

Note that a comma separated list suitable for cut-and-paste into the C and Python programs can be obtained via

```
% Tbaths.csh ,
293,298,303,308,313,318,323,328,333,338,344,350,356,362,368,374,380,386,
392,398,405,412,419,426,433,440,447,454,461,468,
% _
```

A final note—I’ve covered the essential concepts and machinery for running REMD with CHARMM, but I obviously have some additional tools for collating and analyzing the data, both statistically and graphically. However, presentation of this material is beyond the scope of this lecture; I will prepare a .tar.gz archive file which will include the programs and scripts in this document, as well as the additional analysis scripts. The package will be available via an email request to Rick_Venable@nih.gov

Replica Exchange Bibliography 2004-2005

Nguyen PH, Stock G, Mittag E, et al.
Free energy landscape and folding mechanism of a beta-hairpin in explicit water: A replica exchange molecular dynamics study
PROTEINS 61 (4): 795-808 DEC 1 2005

Bernard D, Coop A, Mackerell AD
Conformationally sampled pharmacophore for peptidic delta opioid ligands
J MED CHEM 48 (24): 7773-7780 DEC 1 2005

Baumketner A, Shea JE
The influence of different treatments of electrostatic interactions on the thermodynamics of folding of peptides
J PHYS CHEM B 109 (45): 21322-21328 NOV 17 2005

Seibert MM, Patriksson A, Hess B, et al.
Reproducible polypeptide folding and structure prediction using molecular dynamics simulations
J MOL BIOL 354 (1): 173-183 NOV 18 2005

Lwin TZ, Luo R
Overcoming entropic barrier with coupled sampling at dual resolutions
J CHEM PHYS 123 (19): Art. No. 194904 NOV 15 2005

Borrelli KW, Vitalis A, Alcantara R, et al.
PELE: Protein energy landscape exploration. A novel Monte Carlo based technique
J CHEM THEORY COMPUT 1 (6): 1304-1311 NOV-DEC 2005

Griffin MA, Friedel M, Shea JE
Effects of frustration, confinement, and surface interactions on the dimerization of an off-lattice beta-barrel protein
J CHEM PHYS 123 (17): Art. No. 174707 NOV 1 2005

Adisa B, Bruce DA
Solvophobic and steric effects of side groups on polymer folding: Molecular modeling studies of amine-functionalized m-poly(phenyleneethynylene) foldamers in aqueous solution
J PHYS CHEM B 109 (42): 19952-19959 OCT 27 2005

Zhang W, Wu C, Duan Y
Convergence of replica exchange molecular dynamics
J CHEM PHYS 123 (15): Art. No. 154105 OCT 15 2005

Ravindranathan KP, Gallicchio E, Levy RM
Conformational equilibria and free energy profiles for the allosteric transition of the ribose-binding protein
J MOL BIOL 353 (1): 196-210 OCT 14 2005

Chen JH, Im W, Brooks CL
Application of torsion angle molecular dynamics for efficient sampling of protein conformations
J COMPUT CHEM 26 (15): 1565-1578 NOV 30 2005

Oakley MT, Garibaldi JM, Hirst JD
Lattice models of peptide aggregation: Evaluation of conformational search algorithms
J COMPUT CHEM 26 (15): 1638-1646 NOV 30 2005

Nishino M, Sugita Y, Yoda T, et al.
Structures of a peptide fragment Of beta(2)-microglobulin studied by replica-exchange molecular dynamics simulations - Towards the understanding of the mechanism of amyloid formation
FEBS LETT 579 (24): 5425-5429 OCT 10 2005

Kawashima Y, Sugita Y, Yoda T, et al.
Effects of the fixed end in single-molecule imaging techniques: A replica-exchange molecular dynamics simulation study
CHEM PHYS LETT 414 (4-6): 449-455 OCT 14 2005

Liu P, Kim B, Friesner RA, et al.
Replica exchange with solute tempering: A method for sampling biological systems in explicit water
P NATL ACAD SCI USA 102 (39): 13749-13754 SEP 27 2005

Roe DR, Hornak V, Simmerling C
Folding cooperativity in a three-stranded beta-sheet model
J MOL BIOL 352 (2): 370-381 SEP 16 2005

Nanias M, Chinchio M, Oldziej S, et al.
Protein structure prediction with the UNRES force-field using Replica-Exchange Monte Carlo-with-Minimization; Comparison with MCM, CSA, and CFMC
J COMPUT CHEM 26 (14): 1472-1486 NOV 15 2005

Woods CJ, Ng MH, Johnston S, et al.
Grid computing and biomolecular simulation
PHILOS T ROY SOC A 363 (1833): 2017-2035 AUG 15 2005

Plewczynska D, Kolinski A
Protein folding with a reduced model and inaccurate short-range restraints
MACROMOL THEOR SIMUL 14 (7): 444-451 AUG 25 2005

Yoshida K, Yamaguchi T, Okamoto Y
Replica-exchange molecular dynamics simulation of small peptide in water and in ethanol
CHEM PHYS LETT 412 (4-6): 280-284 SEP 5 2005

Baumketner A, Shea JE
Free energy landscapes for amyloidogenic tetrapeptides dimerization
BIOPHYS J 89 (3): 1493-1503 SEP 2005

Murata K, Sugita Y, Okamoto Y
Molecular dynamics simulations of DNA dimers based on replica-exchange umbrella sampling. I. Test of sampling efficiency
J THEOR COMPUT CHEM 4 (2): 411-432 JUN 2005

Murata K, Sugita Y, Okamoto Y
Molecular dynamics simulations of DNA dimers based on replica-exchange umbrella sampling. II. Free energy analysis
J THEOR COMPUT CHEM 4 (2): 433-448 JUN 2005

Escobedo FA
A unified methodological framework for the simulation of nonisothermal ensembles
J CHEM PHYS 123 (4): Art. No. 044110 JUL 22 2005

Nguyen PH, Mu YG, Stock G
Structure and energy landscape of a photoswitchable peptide: A replica exchange molecular dynamics study
PROTEINS 60 (3): 485-494 AUG 15 2005

Thomas GL, Sessions RB, Parker MJ
Density guided importance sampling: application to a reduced model of protein folding
BIOINFORMATICS 21 (12): 2839-2843 JUN 15 2005

Pokarowski P, Droste K, Kolinski A
A minimal proteinlike lattice model: An alpha-helix motif
J CHEM PHYS 122 (21): Art. No. 214915 JUN 1 2005

Malolepsza E, Boniecki M, Kolinski A, et al.
Theoretical model of prion propagation: A misfolded protein induces misfolding
P NATL ACAD SCI USA 102 (22): 7835-7840 MAY 31 2005

Best RB, Clarke J, Karplus M
What contributions to protein side-chain dynamics are probed by NMR experiments? A molecular dynamics simulation analysis
J MOL BIOL 349 (1): 185-203 MAY 27 2005

Gnanakaran S, Garcia AE
Helix-coil transition of alanine peptides in water: Force field dependence on the folded and unfolded structures
PROTEINS 59 (4): 773-782 JUN 1 2005

Nymeyer H, Woolf TB, Garcia AE
Folding is not required for bilayer insertion: Replica exchange simulations of an alpha-helical peptide with an explicit lipid bilayer
PROTEINS 59 (4): 783-790 JUN 1 2005

Wang JS, Swendsen RH
Replica Monte Carlo simulation (revisited)
PROG THEOR PHYS SUPP (157): 317-323 2005

Paschek D, Gnanakaran S, Garcia AE
Simulations of the pressure and temperature unfolding of an alpha-helical peptide
P NATL ACAD SCI USA 102 (19): 6765-6770 MAY 10 2005

Im W, Brooks CL
Interfacial folding and membrane insertion of designed peptides studied by molecular dynamics simulations
P NATL ACAD SCI USA 102 (19): 6771-6776 MAY 10 2005

Andrec M, Felts AK, Gallicchio E, et al.
Protein folding pathways from replica exchange simulations and a kinetic network model
P NATL ACAD SCI USA 102 (19): 6801-6806 MAY 10 2005

Sugita Y, Okamoto Y
Molecular mechanism for stabilizing a short helical peptide studied by generalized-ensemble simulations with explicit solvent
BIOPHYS J 88 (5): 3180-3190 MAY 2005

Vreede J, Crielaard W, Hellingwerf KJ, et al.
Predicting the signaling state of photoactive yellow protein
BIOPHYS J 88 (5): 3525-3535 MAY 2005

Cheng XL, Cui GL, Hornak V, et al.
Modified replica exchange simulation methods for local structure refinement
J PHYS CHEM B 109 (16): 8220-8230 APR 28 2005

Gallicchio E, Andrec M, Felts AK, et al.
Temperature weighted histogram analysis method, replica exchange, and transition paths
J PHYS CHEM B 109 (14): 6722-6731 APR 14 2005

Predescu C, Predescu M, Ciobanu CV
On the efficiency of exchange in parallel tempering Monte Carlo simulations
J PHYS CHEM B 109 (9): 4189-4196 MAR 10 2005

Chen JH, Won HS, Im WP, et al.
Generation of native-like protein structures from limited NMR data, modern force fields and advanced conformational sampling
J BIOMOL NMR 31 (1): 59-64 JAN 2005

Habeck M, Nilges M, Rieping W
Replica-exchange Monte Carlo scheme for Bayesian data analysis
PHYS REV LETT 94 (1): Art. No. 018105 JAN 14 2005

Poulain P, Antoine R, Broyer M, et al.
Monte Carlo simulations of flexible molecules in a static electric field: electric dipole and conformation
CHEM PHYS LETT 401 (1-3): 1-6 JAN 1 2005

Yang X, Saven JG
Computational methods for protein design and protein sequence variability: biased Monte Carlo and replica exchange
CHEM PHYS LETT 401 (1-3): 205-210 JAN 1 2005

Chen JH, Im W, Brooks CL
Refinement of NMR structures using implicit solvent and advanced sampling techniques
J AM CHEM SOC 126 (49): 16038-16047 DEC 15 2004

Mitsutake A, Kinoshita M, Okamoto Y, et al.
Combination of the replica-exchange Monte Carlo method and the reference interaction site model theory for simulating a peptide molecule in aqueous solution
J PHYS CHEM B 108 (49): 19002-19012 DEC 9 2004

Paschek D, Garcia AE
Reversible temperature and pressure denaturation of a protein fragment: A replica exchange molecular dynamics simulation study
PHYS REV LETT 93 (23): Art. No. 238105 DEC 3 2004

Jas GS, Kuczera K
Equilibrium structure and folding of a helix-forming peptide: Circular dichroism measurements and replica-exchange molecular dynamics simulations
BIOPHYS J 87 (6): 3786-3798 DEC 2004

Nguyen HD, Hall CK
Phase diagrams describing fibrillization by polyalanine peptides
BIOPHYS J 87 (6): 4122-4134 DEC 2004

Cecchini M, Rao F, Seeber M, et al.
Replica exchange molecular dynamics simulations of amyloid peptide aggregation
J CHEM PHYS 121 (21): 10748-10756 DEC 1 2004

Nguyen HD, Marchut AJ, Hall CK
Solvent effects on the conformational transition of a model polyalanine peptide
PROTEIN SCI 13 (11): 2909-2924 NOV 2004

Arkin H
A combination of replica exchange Monte Carlo and energy landscape paving algorithms to increase the effectiveness of conformational sampling
INT J MOD PHYS C 15 (7): 933-937 SEP 2004

Kokubo H, Okamoto Y
Classification and prediction of low-energy membrane protein helix configurations by replica-exchange Monte Carlo method
J PHYS SOC JPN 73 (9): 2571-2585 SEP 2004

Furlan S, La Penna G, Perico A, et al.
Conformational dynamics of hyaluronan oligomers in solution. 3. Molecular dynamics from Monte Carlo replica-exchange simulations and mode-coupling diffusion theory
MACROMOLECULES 37 (16): 6197-6209 AUG 10 2004

Mitsutake A, Okamoto Y
Replica-exchange extensions of simulated tempering method
J CHEM PHYS 121 (6): 2491-2504 AUG 8 2004

Felts AK, Harano Y, Gallicchio E, et al.
Free energy surfaces of beta-hairpin and alpha-helical peptides generated by replica exchange molecular dynamics with the AGBNP implicit solvent model
PROTEINS 56 (2): 310-321 AUG 1 2004

Kokubo H, Okamoto Y
Self-assembly of transmembrane helices of bacteriorhodopsin by a replica-exchange Monte Carlo simulation
CHEM PHYS LETT 392 (1-3): 168-175 JUL 1 2004

Gnanakaran S, Hochstrasser RM, Garcia AE
Nature of structural inhomogeneities on folding a helix and their influence on spectral measurements
P NATL ACAD SCI USA 101 (25): 9229-9234 JUN 22 2004

Kolinski A
Protein modeling and structure prediction with a reduced representation
ACTA BIOCHIM POL 51 (2): 349-371 2004

Swope WC, Pitera JW, Suits F, et al.
Describing protein folding kinetics by molecular dynamics simulations. 2. Example applications to alanine dipeptide and beta-hairpin peptide
J PHYS CHEM B 108 (21): 6582-6594 MAY 27 2004

Kokubo H, Okamoto Y
Prediction of membrane protein structures by replica-exchange Monte Carlo simulations: Case of two helices
J CHEM PHYS 120 (22): 10837-10847 JUN 8 2004

Feig M, Karanicolas J, Brooks CL
MMTSB Tool Set: enhanced sampling and multiscale modeling methods for applications in structural biology
J MOL GRAPH MODEL 22 (5): 377-395 Sp. Iss. SI MAY 2004

Okamoto Y
Generalized-ensemble algorithms: enhanced sampling techniques for Monte Carlo and molecular dynamics simulations
J MOL GRAPH MODEL 22 (5): 425-439 Sp. Iss. SI MAY 2004

Zhou RH
Exploring the protein folding free energy landscape: coupling replica exchange method with P3ME/RESPA algorithm
J MOL GRAPH MODEL 22 (5): 451-463 Sp. Iss. SI MAY 2004

Im W, Brooks CL
De novo folding of membrane proteins: An exploration of the structure and NMR properties of the fd coat protein
J MOL BIOL 337 (3): 513-519 MAR 26 2004

Romiszowski P, Sikorski A
Thermodynamics of polypeptide model chains
PHYSICA A 336 (1-2): 187-195 MAY 1 2004

Shin S, Son WJ, Jang S
Quantum phase transition of water clusters: molecular dynamics simulations with a model potential
J MOL STRUC-THEOCHEM 673 (1-3): 109-113 MAR 19 2004

Yoda T, Sugita Y, Okamoto Y
Comparisons of force fields for proteins by generalized-ensemble simulations
CHEM PHYS LETT 386 (4-6): 460-467 MAR 11 2004

Yang WY, Pitera JW, Swope WC, et al.
Heterogeneous folding of the trpzip hairpin: Full atom simulation and experiment
J MOL BIOL 336 (1): 241-251 FEB 6 2004

Murata K, Sugita Y, Okamoto Y
Free energy calculations for DNA base stacking by replica-exchange umbrella sampling
CHEM PHYS LETT 385 (1-2): 1-7 FEB 2 2004

Garcia AE
Characterization of non-alpha helical conformations in Ala peptides
POLYMER 45 (2): 669-676 JAN 15 2004

Beauvais C, Guerrault X, Coudert FX, et al.
Distribution of sodium cations in faujasite-type zeolite: A canonical parallel tempering simulation study
J PHYS CHEM B 108 (1): 399-404 JAN 8 2004