Symbolic Mutual Information

# Makefile\_mi\_mpi --- make file for compiling and creating the executable mirun.exe (MPI version); To be used as: make –f Makefile\_mi\_mpi

# Makefile\_mi\_ser --- make file for compiling and creating the executable mirun\_ser.exe (serial version); To be used as: make –f Makefile\_mi\_ser

# This class contains the following files:

# demo\_mi.f90

This file is initializing the MI\_CLASS, which is the module used for calculation of symbolic mutual information between time series. It contains the subroutines/functions used to read the input parameters for computation of the symbolic mutual information.

# This subroutine calls one of the following functions (which are part of the MI\_CLASS in minfo.f90 (MPI version) and minfo\_ser.f90 (serial version):

# IF (qMIMethod == 1) THEN

# CALL SMI\_DRIVER(frm, Nframes, Ndim, Natoms, qMIShuffle, debug, Nshuffles, Rcut, StatP) -- It computes symbolic MI using method by Kamberaj & Van der Vaart

# ELSE

# CALL MI\_DRIVER(frm, Nframes, Ndim, Natoms, qMIShuffle, debug, Nshuffles, Rcut, StatP) -- It computes symbolic MI using method by Schreiber

# ENDIF

# subroutine write\_MI(qMIShuffle) -- It writes the symbolic MI in an output file

# subroutine Allocate\_MI(qmimethod, qMIShuffle) -- It allocates memory for global dynamical variables of the MI\_CLASS

# subroutine deAllocate\_MI() -- It frees memory allocated for global dynamical variables of the MI\_CLASS

# subroutine MPI\_BroadCast(qMIShuffle, debug, Nshuffles, r0, statP, model) -- It broad casts the global variables of the MI\_CLASS to all processors (MPI version)

# subroutine getSMINDIM\_MPIDOF(debug) -- It computes symbolic MI using the method by Kamberaj & van der Vaart (MPI version) without shuffling

# subroutine getSMINDIM\_MPISFL(qMIShuffle, debug, Nshuffles, r0, statP) -- It computes symbolic MI using the method by Kamberaj & van der Vaart (MPI version) with shuffling

# subroutine getMINDIM\_MPIDOF(debug) -- It computes symbolic MI using the method by Schreiber (MPI version) without shuffling

# subroutine getMINDIM\_MPISFL(qMIShuffle, debug, Nshuffles, r0, statP) -- It computes symbolic MI using the method by Schreiber (MPI version) with shuffling

#

# Auxiliary functions / subroutines:

# subroutine symbolic\_MI\_ND(ndata,ndim,xs,ys,m1,m2,tau1,tau2,mi) -- It computes symbolic MI between two symbolic time series xs, ys using method by Kamberaj & van der Vaart

# subroutine symbolic\_MI2\_ND(ndata,ndim,xs,ys,m1,m2,tau1,tau2,MI) -- It computes symbolic MI between two symbolic time series xs, ys using method by Schreiber

# The Input Parameters of this Module are:

Length of time series - **Nframes**

Number of time series - **Natoms**

Dimensionality of the problem - **Ndim**

Flag for Method of MI calculation (1: Discrete; 2: Schreiber) - **qMIMethod**

Flag for Method of Shuffling of MI (1: Permutation; 2: Block shuffling) - **qMIShuffle**

Number of Shuffling of time series - **Nshuffles**

Cutoff for Mutual Information minimum value - **Rcut**

Confidence level for averages - **statP**

Set debugging flag value - **Debug**

Chose the format of Input/Output data – **frmt**

A bash shell script is given below:

!#/bin/bash

root=../sifm-master

nframes=10000

natoms=2

ndim=1

qMIMethod=1

qMIShuffle=1

Nshuffles=10

Rcut=0.02

statP=0.95

Debug=1

parallel=0

NP=2

frmt=’csv’

if [ parallel == 1 ]; then

mpirun -np $NP ${root}/mirun.exe ${nframes} ${natoms} ${ndim} ${qMIMethod} ${qMIShuffle} ${Nshuffles} ${Rcut} ${statP} ${Debug} ${frmt}

else

${root}/mirun.exe ${nframes} ${natoms} ${ndim} ${qMIMethod} ${qMIShuffle} ${Nshuffles} ${Rcut} ${statP} ${Debug} ${frmt}

fi