sifm = Symbolic Information Flow Measurement

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

#

# Id: sifm.doc,v 1.0 10-02-2020, IBU

#

# This source code is part of

#

# Symbolic Information Flow Measure Code for studying the

# information flow in dynamical systems

#

# VERSION 1.0

#

# Written by Hiqmet Kamberaj.

# Copyright (C) 2019 Hiqmet Kamberaj.

# Check out h.kamberaj@gmail.com for more information.

#

# This program is free software; you can redistribute it and/or modify it under

# the terms of the GNU General Public License as published by the Free Software Foundation;

# GPL-3.0

#

# This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;

# without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

# See the GNU General Public License for more details.

#

# You should have received a copy of the GNU General Public License along with this program;

# if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330,

# Boston, MA 02111-1307 USA

#

#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

The toolkit is composed of the following MODULES:

# Embedded Parameters computations:

A description of this module can be found following the link below:

[Embd.doc](embd.docx)

# Symbolic analysis Module

A description of the symbolic analysis of the time series module can be found following the link below:

[Symb.doc](symba.docx)

# Symbolic Transfer Entropy

A description of the symbolic transfer entropy module can be found following the link below:

[Ste.doc](ste.docx)

# Local Symbolic Transfer Entropy

A description of the symbolic local transfer entropy module can be found following the link below:

[Slte.doc](slte.docx)

# Symbolic Mutual Information

A description of the symbolic mutual information module can be found following the link below:

[Smi.doc](smi.docx)

# System configuration setups

A description of the settting up different benchmark test systems can be found following the link below:

[Setup.doc](setup.docx)

# Auxiliary Program functions/subroutines

A description of sime auxiliary functions and subroutines for different modules can be found following the link below:

[Auxil.doc](auxil.docx)

# Analyzing Collective Variables Using Machine Learning

BSANNPy folder contains python codes for performing Machine Learning to encode the information from the real dynamical variables.

[Bsann.doc](bsann.docx)