



information-dynamics-toolkit

JIDT: Java Information Dynamics Toolkit for studying information-theoretic measures of computation in complex systems

 [Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#) [Administer](#) for [Home](#)[Getting started](#)[Implemented Measures](#)[Demos](#)[Tutorial](#)[Non-Java environments](#)[Matlab/Octave](#)[Python](#)[R](#)[Julia](#)[Clojure](#)[FAQs](#)[Miscellaneous](#)[For serious developers!](#)[Publications resulting](#)[Edit](#) <<

UseInClojure

How to use the toolkit in Clojure

clojure

Updated Sep 8, 2014 by [joseph.liz](#)

Introduction

The Java code from this toolkit can easily be used in [Clojure](#) -- the project jar has been deployed to maven as [me.lizier/jidt](#) for automated inclusion in clojure projects (see further details below).

Here we give only a brief overview of calling Java code from [Clojure](#); several longer examples of using the JIDT toolkit in Clojure can be viewed at [Clojure Examples](#).

Using Java objects in Clojure

No special installations are required to begin using Java objects in Clojure, these are supported natively as described [here](#).

You can run your Java code in Clojure as follows:

1. Add the `[me.lizier/jidt "LATEST"]` for the latest version (or name a specific version) to the `:dependencies` vector of your `project.clj` file (this automatically references the JIDT jar from the leiningen repository and pulls it in). See our sample [project.clj](#) file in our [Clojure Examples](#).
2. Import the classes you wish to use, e.g. `(import infodynamics.measures.discrete.TransferEntropyCalculatorDiscrete)`.
3. Create an instance of the calculator you wish to use, e.g. `(def teCalc (TransferEntropyCalculatorDiscrete. 2 1))`
4. Call methods on the object, e.g. `(.addObservations teCalc sourceArray destArray)`.

Array conversion -- is generally straightforward -- see some details at the [Clojure-Java interoperability documentation](#), and the use of `int-array`, `into-array` and `map` in our [Clojure Examples](#).

Acknowledgements

A big thank you to Matthew Chadwick for showing me how to do this and getting things up and running with clojure and leiningen.

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)