

information-dynamics-toolkit



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

 Search projects[Project Home](#)[Downloads](#)[Wiki](#)[Issues](#)[Source](#)[Administer](#)[Export to GitHub](#)

for

>>

**UseInPython***How to use the toolkit in Python*[Phase-Deploy](#), [python](#)Updated Aug 20, 2014 by [joseph.lizier](#)

Introduction

The java code from this toolkit can easily be used in Python. First we describe calling java code from Python, then specifically describe the use of this toolkit.

Several longer examples of using the toolkit in Python can be viewed at [PythonExamples](#).

Using Java objects in Python

There are several alternatives for using Java objects in Python, e.g. [JPyPe](#) and [Jython](#).

Here, we focus on using JPyPe for several reasons, including:

- It is run as an import to ordinary python code in the standard interpreter (jython runs as an alternative python interpreter, and so is not always up to date with the main python interpreter)
- It is packaged for ubuntu (which is my platform of choice) as `python-jpype`

As such, here we will only describe use via JPyPe, though of course you could use an alternative method quite easily.

Using JPyPe

The first step is to install JPyPe. On ubuntu, one simply installs `python-jpype` via the ubuntu package manager. On other platforms, install as you would other python extensions.

You can then run your java code fairly simply in python with JPyPe:

1. Import the relevant packages from jPyPe, e.g.: `from jpype import *`
2. Start the JVM and tell it where our jar file is, e.g.: `startJVM(getDefaultJVMPath(), "-Djava.class.path=" + jarLocation)`
3. Create a reference to the package and class that you wish to create an instance of, e.g.: `teCalcClass = JPackage("infodynamics.measures.discrete").TransferEntropyCalculatorDiscrete`
4. Create an instance of the class, e.g.: `teCalc = teCalcClass(2,1)`
5. Use the object, e.g.: `teCalc.initialise()`
6. Shutdown the JVM when it's no longer required, e.g.: `shutdownJVM()`

Array conversion between python and java requires some extra code, but is perfectly do-able - see sections [6](#) and [7](#) of the [JPyPe user guide](#). For example, to convert a one-dimensional python array `myArray` to a java one dimensional array of doubles, use `JArray(JDouble, 1)(myArray)` -- see e.g. Example 3 in [PythonExamples](#); alternatively, if you use [numpy](#).array objects these seem to be directly accepted by the java methods -- see e.g. Example 6 in [PythonExamples](#).

For static methods, you can call them on the reference to the class itself.

Several longer examples of using the toolkit in Python can be viewed at [PythonExamples](#).

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)Powered by [Google Project Hosting](#)