cppEDM   pyEDM   rEDM

Package Testing and Installation

JosephPark@IEEE.org

Table of Contents

Overview

The EDM software suite consists of three components: cppEDM, pyEDM, rEDM. The core computation engine is cppEDM. pyEDM and rEDM are language-specific interfaces to cppEDM. The pyEDM interface is based on the `pybind11` module, rEDM uses `Rcpp`.

Required Testing

The following tests are required prior to a new release of the EDM software:

1. cppEDM API and application build test
2. cppEDM numerical validation tests
3. cppEDM graphical tests

4. pyEDM examples
5. pyEDM unit tests

6. rEDM Examples
7. rEDM unit tests

Building cppEDM

To build cppEDM

```
cd cppEDM/src
make
```

Testing cppEDM

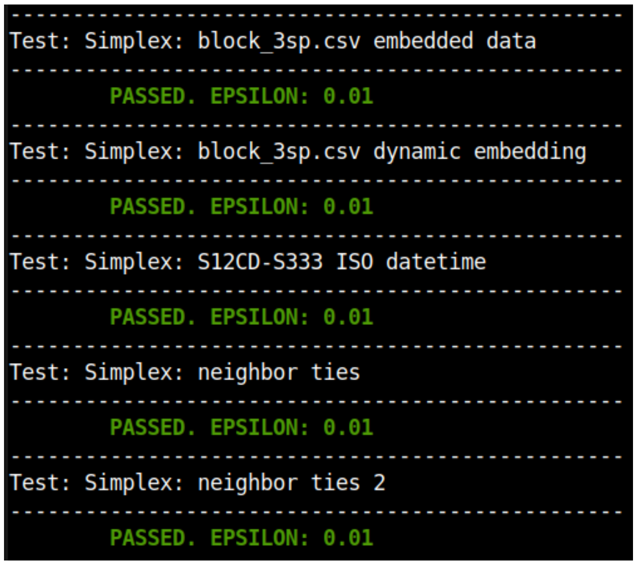
1) Step 1 verifies applications using the cppEDM API can be built and executed. This is done with the `cppEDM/etc/check` shell script. CCM results will vary.

```
etc> ./check

------- Building ------------------------------------------
------- Embed ---------------------------------------------
normal termination
------- Simplex -------------------------------------------
simplex on ../data/block_3sp.csv:
rho   0.934374  RMSE   0.291486  MAE   0.228646
normal termination
------- CCM -----------------------------------------------
normal termination
LibSize,anchovy:np_sst,np_sst:anchovy
10.0000,0.0998,-0.0244
70.0000,0.2214,-0.0606
75.0000,0.2097,-0.0558
------- Multiview -----------------------------------------
Multiview() Set view sample size to 9
Multiview()../data/block_3sp.csv
rho 0.943597  MAE 0.228478  RMSE 0.276099
normal termination
------- EmbedDimension ------------------------------------
EmbedDimension:
normal termination
PredictInterval:
normal termination
PredictNonlinear:
normal termination
EmbedDimension:
normal termination
```

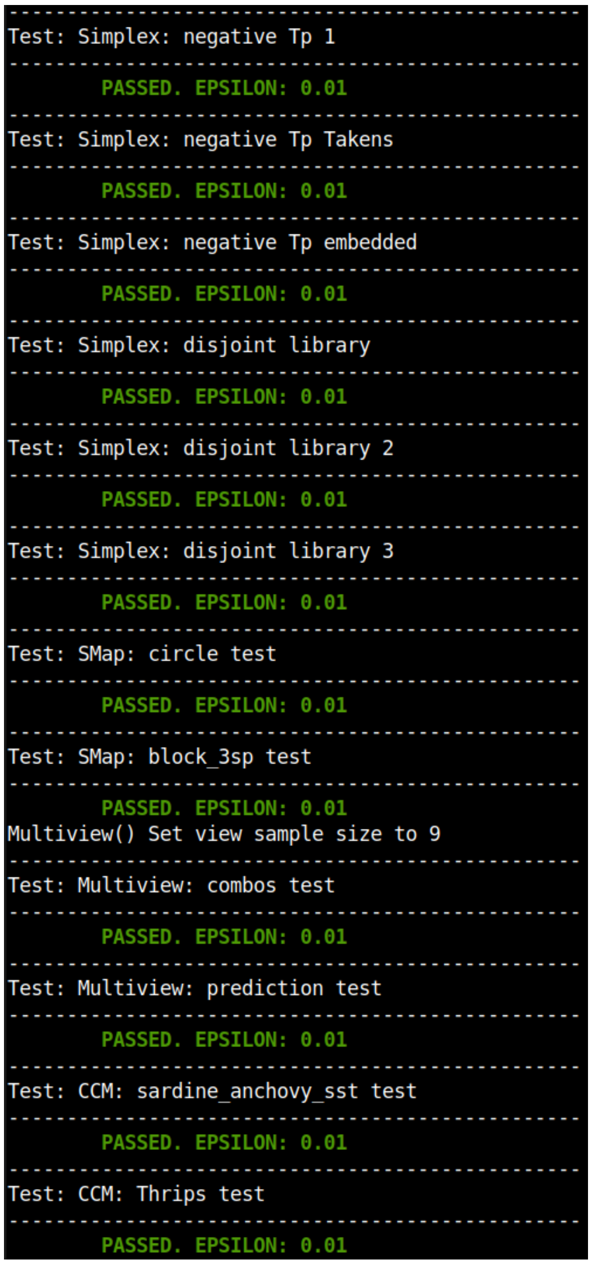
2) Essential numerical checks are done in `cppEDM/tests` by the programs: `CCMTest.cc DateTimeTest.cc MultiviewTest.cc SimplexTest.cc TestCommon.cc SMapTest.cc`

Tests are built and run with the `cppEDM/tests/run` shell script. PASS/FAIL is reported on the console:

```
tests> ./run

g++ TestCommon.cc -c -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack
g++ SimplexTest.cc -o SimplexTest -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack TestCommon.o
g++ TestCommonTest.cc -o TestCommonTest -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack TestCommon.o
g++ SMapTest.cc -o SMapTest -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack TestCommon.o
g++ CCMTest.cc -o CCMTest -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack TestCommon.o
g++ MultiviewTest.cc -o MultiviewTest -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack TestCommon.o
g++ DateTimeTest.cc -c -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack
g++ DateTimeTest.cc -o DateTimeTest -std=c++11 -D
PRINT_DIFFERENCE_IN_RESULTS -lstdc++ -L../lib/
-I../src/ -lEDM -lpthread -llapack TestCommon.o
```



```
-----------------------------------------------
Test: Simplex: block_3sp.csv embedded data
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: block_3sp.csv dynamic embedding
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: S12CD-S333 ISO datetime
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: neighbor ties
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: neighbor ties 2
-----------------------------------------------
       PASSED. EPSILON: 0.01
```

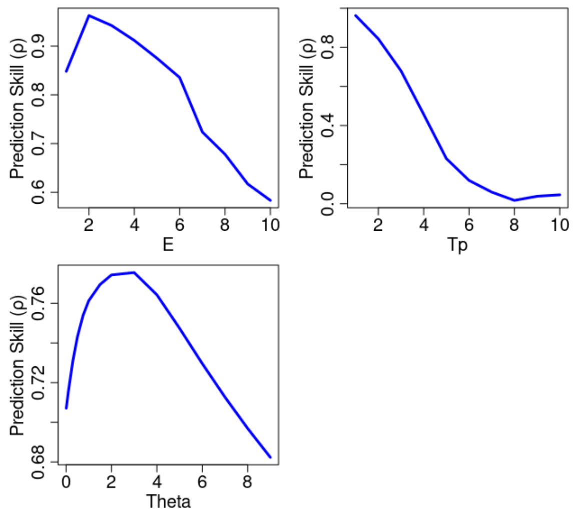


```
-----------------------------------------------
Test: Simplex: negative Tp 1
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: negative Tp Takens
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: negative Tp embedded
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: disjoint library
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: disjoint library 2
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Simplex: disjoint library 3
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: SMap: circle test
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: SMap: block_3sp test
-----------------------------------------------
       PASSED. EPSILON: 0.01
Multiview() Set view sample size to 9
-----------------------------------------------
Test: Multiview: combos test
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: Multiview: prediction test
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: CCM: sardine_anchovy_sst test
-----------------------------------------------
       PASSED. EPSILON: 0.01
-----------------------------------------------
Test: CCM: Thrips test
-----------------------------------------------
       PASSED. EPSILON: 0.01
```

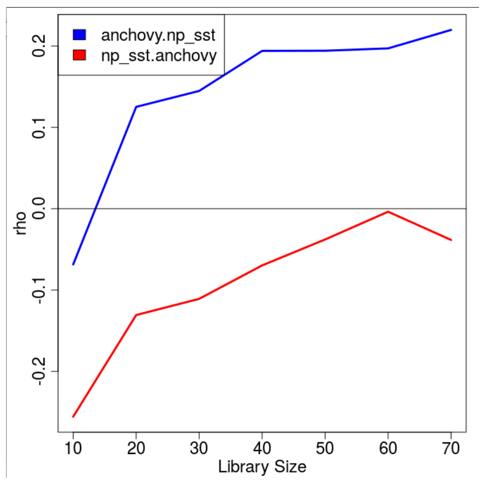
3) Graphical tests are run by the cppEDM/etc/Test.cc program and rendered with the R application PlotTest.R. Carefully check that graphical output matches the images shown below. The CCM test will not match exactly, but the relative behavior should be the same as shown.

```
cd cppEDM/etc
./test
```

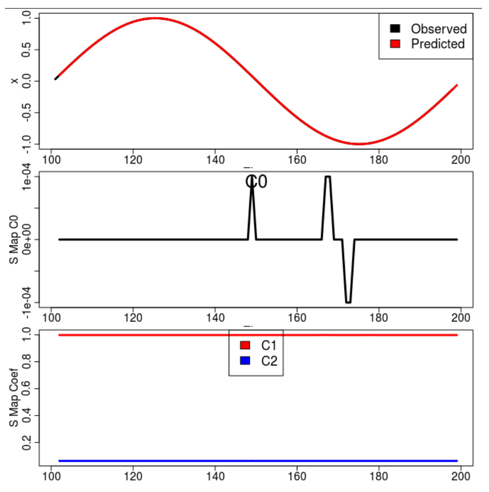
Embed Dimension Predict Interval Predict Nonlinear



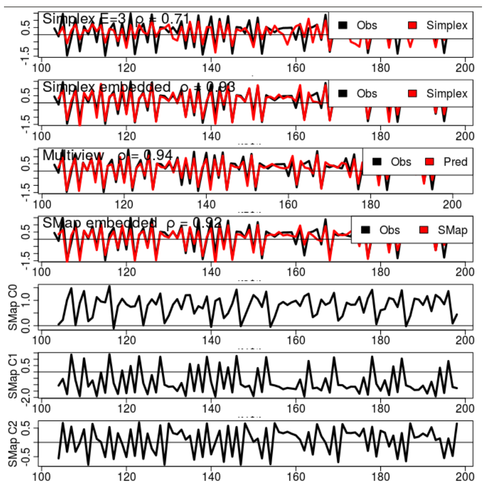
CCM



SMap



Simplex SMap Multiview



Building cppEDM on Windows

This has been found to work on Windows 10 with MSVC 2019 build tools and mingw.

Build cppEDM/src:

```
nmake /f makefile.windows
```

Compile cppEDM/etc/Test.cc into Test.obj:

```
 cl /c Test.cc /EHsc /MD /I../src
```

Download .lib and .dll from Windows for LAPACKE:
https://icl.cs.utk.edu/lapack-for-windows/lapack/#lapacke
Copy .dll and .lib from LAPACKE_examples.zip into ../../lapacke

Link Test.obj into Test.exe:

```
  link /OUT:Test.exe /LIBPATH:../lib /LIBPATH:../../lapacke EDM.lib
  liblapack.lib Test.obj
```

Get missing libraries for LAPACK legacy:
Downloaded libgfortran-3.dll into ../../lapacke
https://www.opendll.com/index.php?file-download=libgfortran-3.dll&arch=32bit

Downloaded libwinpthread-1.dll into ../../lapacke
https://wikidll.com/mingw-w64/libwinpthread-1-dll

Set PATH to find the lapacke and mingw dll's:

```
PATH=../../lapacke;C:\MINGW\BIN;%PATH%
```

Run Test.exe

Building pyEDM

pyEDM can be installed from the PyPI respository using pip: `pip install pyEDM`

pyEDM can be built locally from the github repository:

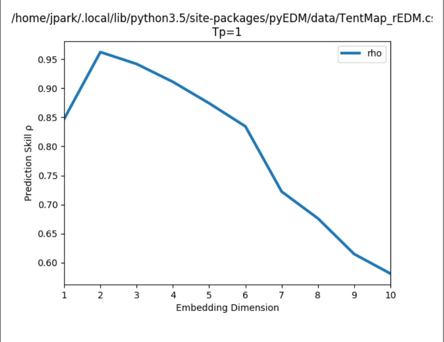
```
     git clone https://github.com/SugiharaLab/pyEDM.git
     cd pyEDM
     python -m pip install . --user
```

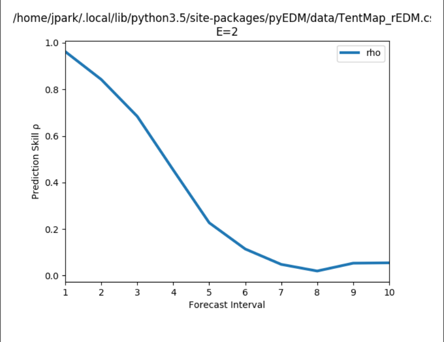
```
Processing /home/temp/pyEDM
Requirement already satisfied: pybind11>=2.3 in /home/jpark/.local/lib/python3.8/site-packages (from
pyEDM==1.8.1.0) (2.5.0)
Requirement already satisfied: pandas>=0.20.3 in /home/jpark/.local/lib/python3.8/site-packages (from
pyEDM==1.8.1.0) (1.1.3)
Requirement already satisfied: matplotlib>=2.2 in /home/jpark/.local/lib/python3.8/site-packages (from
pyEDM==1.8.1.0) (3.3.2)
Requirement already satisfied: numpy>=1.15.4 in /home/jpark/.local/lib/python3.8/site-packages (from
pandas>=0.20.3->pyEDM==1.8.1.0) (1.19.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/lib/python3/dist-packages (from
pandas>=0.20.3->pyEDM==1.8.1.0) (2.7.3)
Requirement already satisfied: pytz>=2017.2 in /usr/lib/python3/dist-packages (from pandas>=0.20.3-
>pyEDM==1.8.1.0) (2019.3)
Requirement already satisfied: cycler>=0.10 in /home/jpark/.local/lib/python3.8/site-packages (from
matplotlib>=2.2->pyEDM==1.8.1.0) (0.10.0)
Requirement already satisfied: certifi>=2020.06.20 in /home/jpark/.local/lib/python3.8/site-packages
(from matplotlib>=2.2->pyEDM==1.8.1.0) (2020.6.20)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/home/jpark/.local/lib/python3.8/site-packages (from matplotlib>=2.2->pyEDM==1.8.1.0) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in /usr/lib/python3/dist-packages (from matplotlib>=2.2-
>pyEDM==1.8.1.0) (7.0.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/jpark/.local/lib/python3.8/site-packages
(from matplotlib>=2.2->pyEDM==1.8.1.0) (1.2.0)
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from cycler>=0.10-
>matplotlib>=2.2->pyEDM==1.8.1.0) (1.14.0)
Building wheels for collected packages: pyEDM
  Building wheel for pyEDM (setup.py) ... done
  Created wheel for pyEDM: filename=pyEDM-1.8.1.0-cp38-cp38-linux_x86_64.whl size=2458609
sha256=a5e3ae184c02269558e4cc8b7afaf4f40977da682c8a6a85dbad2d1ba9fdaa31
  Stored in directory:
/tmp/pip-ephem-wheel-cache-7n0f5pmu/wheels/06/6a/48/4e758cb2564bab33ec7291fc308c90e3c7cdb52e58fba2c06b
Successfully built pyEDM
Installing collected packages: pyEDM
  Attempting uninstall: pyEDM
    Found existing installation: pyEDM 1.8.1.0
    Uninstalling pyEDM-1.8.1.0:
      Successfully uninstalled pyEDM-1.8.1.0
Successfully installed pyEDM-1.8.1.0
```

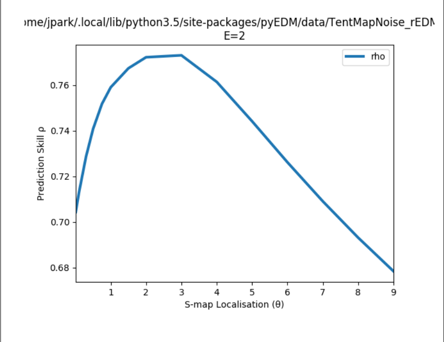
Testing pyEDM

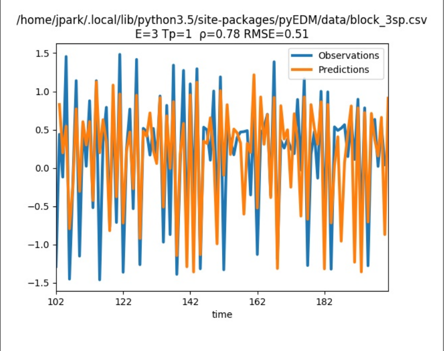
The pyEDM/pyEDM/tests/examples.py program runs a series of tests for the python wrapper and interface. The CCM test will not be numerically equivalent, but must have the same behavior.

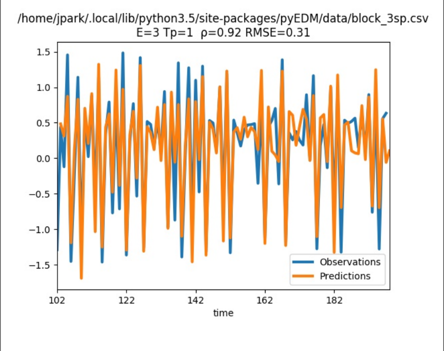
```
cd pyEDM/tests/
./examples.py
```

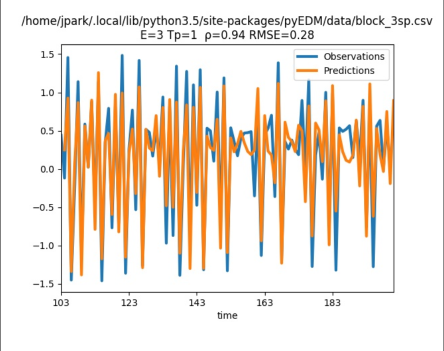


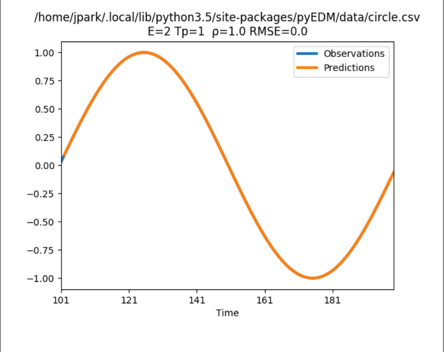


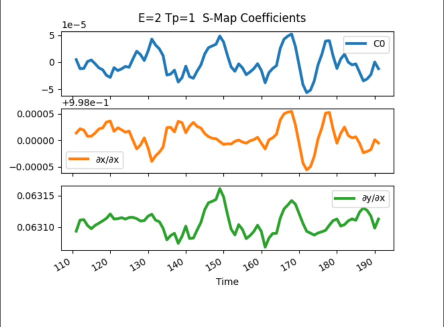


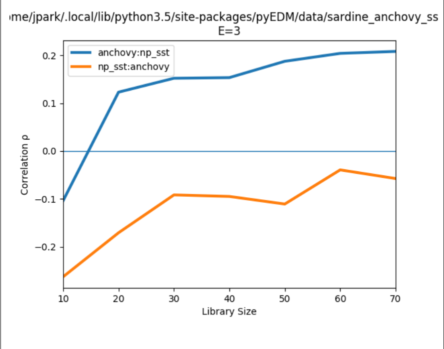












PyEDM python unittests are run in `pyEDM/tests/` with:

```
python -m unittest discover
```

```
        --- CCM ---
        Parameters::Validate(): Set knn = 4 (E+1) for Simplex.
        cppEDM Version 1.2.1 2020-02-05
        CrossMap(): Simplex cross mapping from anchovy to np_sst  E=3  knn=4  Library range: [10 75 5]
        10 15 20 25 30 35 40 45 50 55 60 65 70 75

        cppEDM Version 1.2.1 2020-02-05
        CrossMap(): Simplex cross mapping from np_sst to anchovy  E=3  knn=4  Library range: [10 75 5]
        10 15 20 25 30 35 40 45 50 55 60 65 70 75

        cppEDM Version 1.2.1 2020-02-05
        .--- Multiview ---
        Multiview() Set view sample size to 9
        .--- Simplex embedded = False ---
        .--- Simplex embedded = True ---
        .--- S-map circle embedded = True ---
        .--- S-map block_3sp embedded = True ---
        .
        ----------------------------------------------------------------------
        Ran 6 tests in 0.170s

        OK
```

```
tests> rm -rf __pycache__/
```

Building rEDM

rEDM can be locally built with R CMD in rEDM/.

First, you may wish to cleanup a previous build:

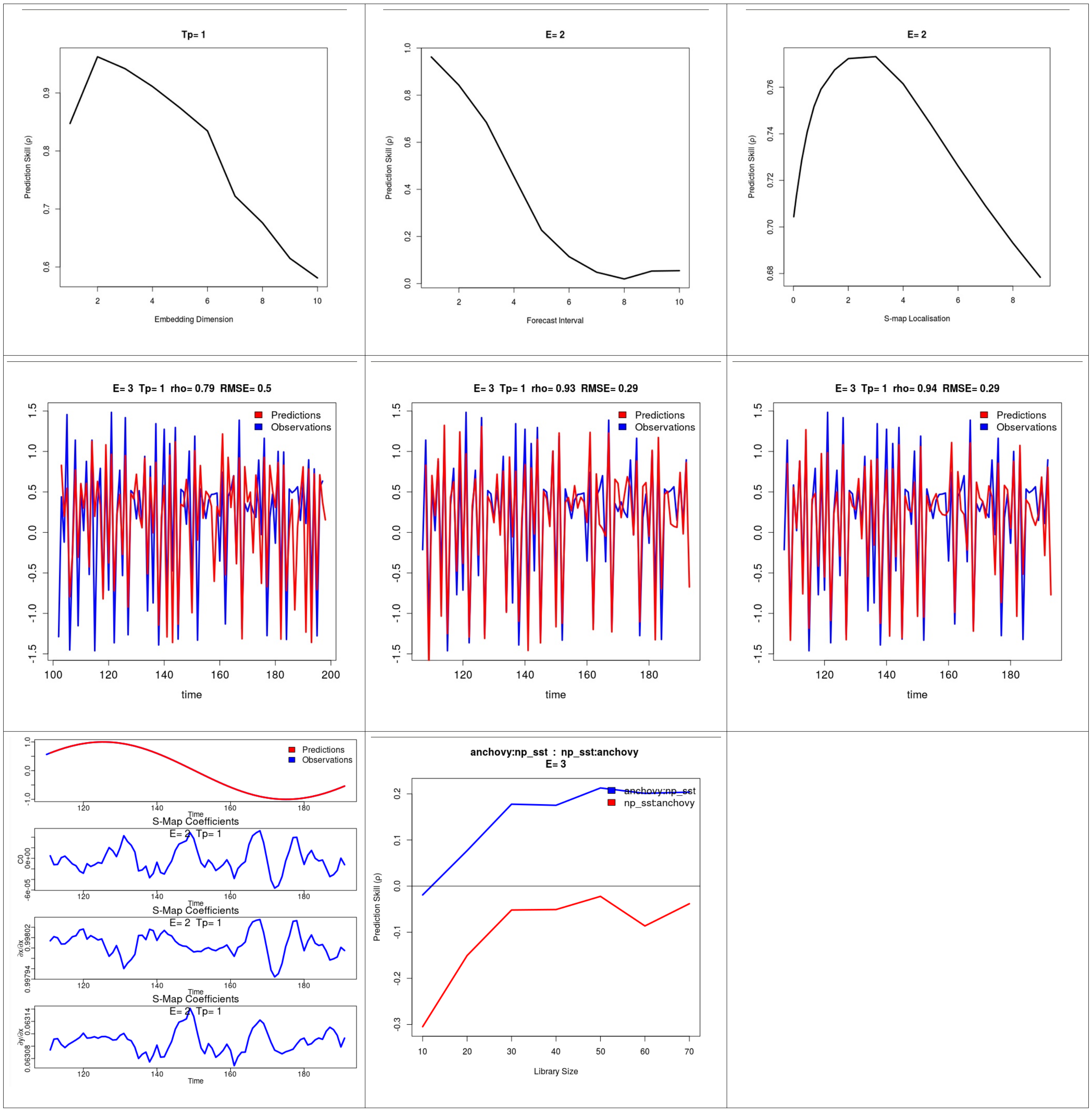
```
cd rEDM
rm -rf src/*.o src/rEDM.so src/cppEDM/lib/libEDM.a

R CMD INSTALL .
```

```
* installing to library ‘/usr/local/lib/R/site-library’
* installing *source* package ‘rEDM’ ...
** libs
g++ -std=gnu++11 -I/usr/share/R/include -DNDEBUG -I ./cppEDM/src/ -I"/usr/local/lib/R/site-
library/Rcpp/include" -I"/usr/local/lib/R/site-library/RcppThread/include"    -fpic  -g -O2 -
fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g -c
CCM.cpp -o CCM.o
...
(cd ./cppEDM/src/; make; make clean)
make[1]: Entering directory 'rEDM/src/cppEDM/src'
g++ -c Common.cc -std=c++11 -DCCM_THREADED -DMULTIVIEW_VALUES_OVERLOAD -O3 -fPIC
g++ -c AuxFunc.cc -std=c++11 -DCCM_THREADED -DMULTIVIEW_VALUES_OVERLOAD -O3 -fPIC
...
ar -rcs libEDM.a Common.o AuxFunc.o DateTimeUtil.o Parameter.o Embed.o Interface.o Neighbors.o
Simplex.o Eval.o CCM.o Multiview.o SMap.o
cp libEDM.a ../lib/
make[1]: Leaving directory 'rEDM/src/cppEDM/src'
make[1]: Entering directory 'rEDM/src/cppEDM/src'
rm -f Common.o AuxFunc.o DateTimeUtil.o Parameter.o Embed.o Interface.o Neighbors.o Simplex.o
Eval.o CCM.o Multiview.o SMap.o  libEDM.a
make[1]: Leaving directory 'rEDM/src/cppEDM/src'
g++ -std=gnu++11 -shared -L/usr/lib/R/lib -Wl,-Bsymbolic-functions -Wl,-z,relro -o rEDM.so CCM.o
ComputeError.o DataFrame.o Embed.o EmbedDim.o Multiview.o PredictInterval.o PredictNL.o
RcppEDMCommon.o RcppExports.o SMap.o Simplex.o -L ./cppEDM/lib/ -lEDM -llapack -L/usr/lib/R/lib
-lR
installing to /usr/local/lib/R/site-library/rEDM/libs
** R
** data
*** moving datasets to lazyload DB
** inst
** preparing package for lazy loading
** help
*** installing help indices
*** copying figures
** building package indices
** installing vignettes
   ‘rEDM-tutorial.Rmd’ using ‘UTF-8’
** testing if installed package can be loaded
* DONE (rEDM)
```

Testing rEDM

The `rEDM/R/Examples.R` program executes `Rcpp` wrapper graphical tests. The CCM test will not be numerically equivalent, but must have the same behavior.

```
cd R/
R
> source("Examples.R")
> Examples()
```



rEDM unit tests are run from `rEDM/tests`

```
cd tests
R
> source('testthat.R')
```

```
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
[1] "Error: ColumnsInDataFrame(): Target  not found."
[1] "Error: ColumnsInDataFrame(): Target None not found."
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
[1] "Error: ColumnsInDataFrame(): Target None not found."
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
[1] "Error: ColumnsInDataFrame(): Target  not found."
Multiview() Set view sample size to 9
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
[1] "Error: ColumnsInDataFrame(): Target None not found."
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
[1] "Error: ColumnsInDataFrame(): Target None not found."
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
[1] "Error: ColumnsInDataFrame(): dataFrame is not valid."
══ testthat results  ═══════════════════════════════════════════════════════════
[ OK: 64 | SKIPPED: 0 | WARNINGS: 0 | FAILED: 0 ]
```

rEDM CRAN

To test and prepare rEDM for CRAN, use the `devtools` package.

1) CRAN build check on local system

```
R
> library(devtools)
> devtools::check()
```

```
── Building ──────────────────────────────────────────────────────────── rEDM ──
Setting env vars:
● CFLAGS    : -Wall -pedantic -fdiagnostics-color=always
● CXXFLAGS  : -Wall -pedantic -fdiagnostics-color=always
● CXX11FLAGS: -Wall -pedantic -fdiagnostics-color=always
────────────────────────────────────────────────────────────────────────────────
✔  checking for file ‘rEDM.build/DESCRIPTION’ ...
─  preparing ‘rEDM’:
✔  checking DESCRIPTION meta-information ...
─  cleaning src
─  installing the package to build vignettes
✔  creating vignettes (1m 16.9s)
─  building ‘rEDM_1.2.2.tar.gz’

── Checking ──────────────────────────────────────────────────────────── rEDM ──
Setting env vars:
● _R_CHECK_CRAN_INCOMING_USE_ASPELL_: TRUE
● _R_CHECK_CRAN_INCOMING_REMOTE_    : FALSE
● _R_CHECK_CRAN_INCOMING_           : FALSE
● _R_CHECK_FORCE_SUGGESTS_          : FALSE
── R CMD check ─────────────────────────────────────────────────────────────────
─  using R version 3.4.4 (2018-03-15)
─  using platform: x86_64-pc-linux-gnu (64-bit)
─  using session charset: UTF-8
─  using options ‘--no-manual --as-cran’
✔  checking for file ‘rEDM/DESCRIPTION’
─  checking extension type ... Package
─  this is package ‘rEDM’ version ‘1.2.2’
✔  checking package namespace information
✔  checking package dependencies (2.3s)
✔  checking dependencies in R code ...
✔  checking compilation flags in Makevars ...
✔  checking compiled code ...
✔  checking sizes of PDF files under ‘inst/doc’ (849ms)
✔  checking installed files from ‘inst/doc’ ...
✔  checking files in ‘vignettes’
✔  checking examples (1.1s)
✔  checking for unstated dependencies in vignettes ...
✔  checking package vignettes in ‘inst/doc’ ...
✔  checking re-building of vignette outputs (13.6s)
   See
     ‘/tmp/RtmpgTq4pn/rEDM.Rcheck/00check.log’
   for details.

── R CMD check results ───────────────────────────────────────── rEDM 1.2.2 ────
Duration: 1m 32.4s

❯ checking installed package size ... NOTE
    installed size is  8.7Mb
    sub-directories of 1Mb or more:
      libs   7.7Mb

0 errors ✔ | 0 warnings ✔ | 1 note ✖
```

2) CRAN build check Using cloud servers. This will email build results to the package maintainer address.

```
R
> library( rhub )
> cranCheck = check_for_cran()
> sanCheck  = check_with_sanitizers()
```

3) Build CRAN release file to upload to CRAN

```
> devtools::build()
```

rEDM Documentation Utilities

Useful commands and rmarkdown package commands to build and convert documentation.

```
rmarkdown::render("rEDM-tutorial.Rmd","pdf_document")
rmarkdown::render("rEDM-tutorial.Rmd","html_document")

R CMD Rd2pdf rEDM
R CMD Rdconv -t html ./rEDM/man/rEDM.Rd > rEDM.html
```

pyEDM PyPI

Microsoft Azure pipeline builds are automatically run when new versions are pushed to the pyEDM github respository as defined in pyEDM/azure-pipelines.yml. The dashboard is here: https://dev.azure.com/cos0080412/pyEDM

The pyEDM package is distributed on the PyPI archives: https://pypi.org/project/pyEDM/

To upload to PyPI, the version string must be different from the previously published one.
Increment the 4th element of `__version__ = "1.2.1.1" in pyEDM/pyEDM/__init__.py`

Optional: Build a single-platform wheels on local machine:

```
python setup.py bdist_wheel
```

Download the Azure wheels (Artificats) from the Azure pipeline.

Upload to PyPI using twine:

```
twine upload [wheel output location]
```

Note: `manylinux` is the only Linux wheel that can be uploaded to PyPI.