

Running eJTK on Mac
OS: MacOS High Sierra Version 10.13.3 (17D102)
Laptop: MacBook Air

##Version Info (important)

Xcode Version 9.3.1 (9E501) ## this might be the biggest factor in your code running or not.

Note: I tried to run the following code on my MacBook Pro with Xcode version 11, and every time it gave me errors. Running it on my MacBook Air with Xcode Version 9.x.x worked fine.

```
python setup.py build_ext --inplace
```

Once you have that, you will need to create a new environment with python 2.7 & to be safe, install the following packages with versions that the python script was originally written in (original version info can be found on the GitHub page):

python version 2.7

```
conda create --name py2 python=2.7
```

Note: This will ask you to install a bunch of packages. Do it.

VERY IMPORTANT:

Make sure you activate this environment before you perform any of the following steps.

To activate the environment py2 you created above, use the following code:

```
conda activate py2
```

Now, install the following packages:

1. cython version 0.24

```
pip install cython==0.24
```

2. scipy version 0.15.1

numpy version 1.11.0

```
pip install scipy==0.15.1 numpy==1.11.0
```

3. statsmodels version 0.6.1

```
pip install statsmodels==0.6.1
```

My GCC was 4.2.1 Apple LLVM version 9.1.0 (clang-902.0.39.1) and it worked fine.

Now, change your directory to the eJTK-master folder and follow the instructions as below:

Execution

Use

```
chmod 755 eJTK-CalcP.py
```

to make eJTK-CalcP.py executable.

Type

```
cd bin/
python setup.py build_ext --inplace
cd ../bin/
to compile the Cython code.
```

Note: I got the following warnings. If you see this, don't sweat.

```
$ python setup.py build_ext --inplace
running build_ext
building 'accessories' extension
gcc -fno-strict-aliasing -I/Users/biplabendudas/miniconda3/envs/py2/include -arch x86_64 -DDEBUG -g -fwrapv -O3 -Wall -Wstrict-prototypes -I/Users/biplabendu
das/miniconda3/envs/py2/include/python2.7 -c accessories.c -o build/temp.macosx-10.6-x86_64-2.7/accessories.o
accessories.c:2073:7: warning: variable '__pyx_v_y' is used uninitialized whenever 'if' condition is false [-Wsometimes-uninitialized]
    if (__pyx_t_5) {
    ~~~~~^~~~~~
accessories.c:2172:34: uninitialized use occurs here
    __pyx_t_1 = PyFloat_FromDouble(__pyx_v_y); if (unlikely(!__pyx_t_1)) __PYX_ERR(0, 46, __pyx_l1_error)
                                   ~~~~~^~~~~~
accessories.c:2073:3: remove the 'if' if its condition is always true
    if (__pyx_t_5) {
    ~~~~~^~~~~~
accessories.c:1933:19: initialize the variable '__pyx_v_y' to silence this warning
    double __pyx_v_y;
           ^
           = 0.0
accessories.c:2376:7: warning: variable '__pyx_v_y' is used uninitialized whenever 'if' condition is false [-Wsometimes-uninitialized]
    if (__pyx_t_5) {
    ~~~~~^~~~~~
accessories.c:2427:34: uninitialized use occurs here
    __pyx_t_3 = PyFloat_FromDouble(__pyx_v_y); if (unlikely(!__pyx_t_3)) __PYX_ERR(0, 57, __pyx_l1_error)
                                   ~~~~~^~~~~~
accessories.c:2376:3: remove the 'if' if its condition is always true
    if (__pyx_t_5) {
    ~~~~~^~~~~~
accessories.c:2273:19: initialize the variable '__pyx_v_y' to silence this warning
    double __pyx_v_y;
           ^
           = 0.0
2 warnings generated.
gcc -bundle -undefined dynamic_lookup -L/Users/biplabendudas/miniconda3/envs/py2/lib -arch x86_64 -arch x86_64 build/temp.macosx-10.6-x86_64-2.7/accessories.o
-L/Users/biplabendudas/miniconda3/envs/py2/lib -o /Users/biplabendudas/eJTK-master-copy/bin/accessories.so
(py2)
```

Proceed to the next step:

Running this command will produce three files

```
./eJTK-CalcP.py -f example/TestInput4.txt -w ref_files/waveform_cosine.txt -p
ref_files/period24.txt -s ref_files/phases_00-22_by2.txt -a
ref_files/asymmetries_02-22_by2.txt -x cos24_ph00-22_by2_a02-22_by2_OTHertext
```

This code has a small typo. It needs to be “./eJTK-CalcP.py – f” for it to work. So, try this code:

```
./eJTK-CalcP.py -f example/TestInput4.txt -w ref_files/waveform_cosine.txt -p
ref_files/period24.txt -s ref_files/phases_00-22_by2.txt -a
ref_files/asymmetries_02-22_by2.txt -x cos24_ph00-22_by2_a02-22_by2_OTHertext
```

If it works, great!

Chances are, you will get the following error:

```
ImportError: No module named matplotlib.pyplot
```

It means you need to install another package matplotlib before you can move on. Do it.

4. matplotlib

```
pip install matplotlib
```

Note: It will install other packages/dependencies. That's okay.

##Version info:

```
backports.functools-lru-cache-1.5
```

```
cycler-0.10.0
```

kiwisolver-1.1.0
matplotlib-2.2.4
pyparsing-2.4.2
python-dateutil-2.8.0
pytz-2019.2
six-1.12.0
subprocess32-3.5.4

Now, run the following again:

```
../eJTK-CalCP.py -f example/TestInput4.txt -w ref_files/waveform_cosine.txt -p  
ref_files/period24.txt -s ref_files/phases_00-22_by2.txt -a  
ref_files/asymmetries_02-22_by2.txt -x cos24_ph00-22_by2_a02-22_by2_OTHERTEXT
```

Most likely you will get the following error:

RuntimeError: Python is not installed as a framework. The Mac OS X backend will not be able to function correctly if Python is not installed as a framework. See the Python documentation for more information on installing Python as a framework on Mac OS X. Please either reinstall Python as a framework, or try one of the other backends. If you are using (Ana)Conda please install python.app and replace the use of 'python' with 'pythonw'. See 'Working with Matplotlib on OSX' in the Matplotlib FAQ for more information.

I found a solution to solve this.

In the command line, type the following:

```
cat > ~/.matplotlib/matplotlibrc  
then write  
backend: TkAgg  
and then press Ctrl+D
```

Double check to see if it worked.

Type in the following in the command line:

```
cat ~/.matplotlib/matplotlibrc
```

You should see the following output:

```
backend: TkAgg
```

Here is the original post for the solution:

<https://markhneedham.com/blog/2018/05/04/python-runtime-error-osx-matplotlib-not-installed-as-framework-mac/>

Alright, you are almost ready to roll!

Change directory to the 'bin' folder inside your eJTK-master folder and run the code again.

```
../eJTK-CalCP.py -f example/TestInput4.txt -w ref_files/waveform_cosine.txt -p  
ref_files/period24.txt -s ref_files/phases_00-22_by2.txt -a  
ref_files/asymmetries_02-22_by2.txt -x cos24_ph00-22_by2_a02-22_by2_OTHERTEXT
```

If you get the following error

ImportError: No module named pandas

You will need to install the package pandas (you know how!).

5. pandas

```
pip install pandas
```

Note: Installing pandas will update the numpy package to the latest one.

##Version info:

numpy-1.16.4

pandas-0.24.2

[Important]

Before you run the file again, you need to copy the 'ref_files' and 'example' folder into the 'bin' folder for the code to work. Let's do that.

You are all set. Let's run the code one last time.

```
../eJTK-CalcP.py -f example/TestInput4.txt -w ref_files/waveform_cosine.txt -p
ref_files/period24.txt -s ref_files/phases_00-22_by2.txt -a
ref_files/asymmetries_02-22_by2.txt -x cos24_ph00-22_by2_a02-22_by2_OTHERTEXT
```

This worked for me, and gave me the following output:

```
example/TestInput4_cos24_ph00-22_by2_a02-22_by2_OTHERTEXT_jtkout.txt already
exists, appending counter to filename!!!
../bin/CalcP.py:45: FutureWarning: read_table is deprecated, use read_csv instead,
passing sep='\t'.
  jtk = pd.read_table(fn_jtk,index_col='ID')
../bin/CalcP.py:53: FutureWarning: read_table is deprecated, use read_csv instead,
passing sep='\t'.
  taus = pd.read_table(fn_pk1)['Tau']
```

Let's break it down:

```
example/TestInput4_cos24_ph00-22_by2_a02-22_by2_OTHERTEXT_jtkout.txt already
exists, appending counter to filename!!!
```

This tells you that the files already exist in the example folder, so it is going to rename the new text files by appending a counter ('1' in this case) at the end. The other two are warnings about using read_table, no biggie.

Interpreting output files

Running this command will produce three files

1. **example/TestInput4_cos24_ph00-22_by2_a02-22_by2_OTHERTEXT_jtkout.txt** This is the output of eJTK.py, it contains the best reference waveform matching each time series. Best is defined as having the highest Tau value. This becomes input for CalcP.py.
2. **example/TestInput4_cos24_ph00-22_by2_a02-22_by2_OTHERTEXT_jtknull1000.txt** This is the output of eJTK.py unless otherwise specified by the -n flag (see eJTK-CalcP.py -h for more information). It is similar to *jtkout.txt only it contains the results of 1000 runs of Gaussian noise. It is also an input for CalcP.py
3. **example/TestInput4_cos24_ph00-22_by2_a02-22_by2_OTHERTEXT_jtkout_GammaP.txt** This is the output of CalcP.py. It is the equivalent of *jtkout.txt, only now with correct p-values as estimated by fitting the time series to a Gamma distribution. It also contains a column of these p-values adjusted with the Benjamini-Hochberg correction.

If you run the above command as is will produce files with a '_1' appended, as these files already exist in the examples folder.

Output information:

The output *jtkout.txt and *GammaP.txt files have columns as described below

ID: Name of time series analyzed

Waveform: Waveform used in analysis

Period: Period of best matching reference waveform

Phase: Phase of best matching reference waveform

Nadir: Trough of best matching reference waveform

Mean: Mean of time series

Std_Dev: Standard Deviation of time series

MaxLoc: Location of maximum of time series

MinLoc: Location of minimum of time series

Max: Maximum value of time series

Min: Minimum value of time series

Max_Amp: Max-Min of time series

FC: Fold Change (Max/Min)

IQR_FC: Fold Change of 25% and 75% percentiles of time series

Tau: Highest Kendall Tau Correlation between time series and reference waveforms

P: P-value corresponding to Tau, uncorrected for multiple hypothesis testing

BF: Bonferroni adjusted P-value from P

empP: min(P-value calculated from empirical null distribution, BF)

GammaP: min(P-value calculated from Gamma Fit of empirical null distribution, empP)

GammaBH: Benjamini-Hochberg adjusted p-value of GammaP