Michael Hughes

Michael.evan.hughes@gmail.com

Hogenesch / Nitabach Labs

Updated 9-21-15

**JTK User’s Guide**

Overview:

JTK is a nonparametic test procedure that detects cycling transcripts from microarray data. In addition to providing optimal phase (LAG), amplitude (AMP) and period (PER) estimates for each transcript, JTK also outputs permutation-based p-values (ADJ.P) and Benjamini–Hochberg q-values (BH.Q). Compared to commonly used cycling tests (for examples, see Hughes et al. Plos Genetics 2009), JTK has advantages in the statistical power of its p-value assignments, improved resistance to outliers, as well as relative computational efficiency.

In our tests (Windows Vista, 2.66 GHz dual processors, R version 2.10.0), most standard analyses (48 time points, ~45k transcripts, 3 hour (23-25h) period range) finish within 15-20 minutes. Increasing the period range can significantly add to the length of the analysis, but even very large ranges (e.g. 4-40h) completed within two hours.

Additionally, JTK\_Cycle can output confidence intervals for amplitude (AMP) estimates. The scripts necessary to perform these calculation are explained by Example 3.

Since version 3, JTK\_Cycle can analyze time-series datasets with missing values, which is explained by Example 4 and Example 5.

Protocol:

**Formatting your data.** JTK assumes your data is stored as a tab-delimited .txt file with data for each probeset on a different row. The first column of data includes the probeset IDs and the first row contains the ZT/CT times your samples were collected (see “Example1\_data.txt”). If you collected samples in replicate, make sure the replicate samples are in adjacent columns (see “Example2\_data.txt”). In addition, prepare an annotation file which includes probeset IDs in the first column and the annotation of your choice (typically gene names or symbols) in the adjacent columns, taking care to ensure that these annotations are in the same order as your data file (see “ExampleX\_annot.txt”).

**Setting up R.** Open ‘R’ and change the active directory (File>Change Dir) to a convenient folder. Make sure this folder includes your data file (described above), your annotation file (described above), and the source code for JTK (“JTK\_CYCLEv3.1.R”).

**Modifying the JTK script.**

*Reading your data and annotation files:* Using a text editor (I prefer SciTE, but most any editor works), open the “Run\_JTK\_Cycle (Example1).R” file. Change the project name to some convenient label (line 4, by default: ‘Example1’). Change the filenames on lines 7 and 8 so that they correspond with your data and annotation files (by default, ‘Example1\_annot.txt’ and ‘Example1\_data.txt’).

*Handling replicates and sampling intervals:* By default, JTK assumes one sample per time point, sampled at a 1 hour resolution. This is illustrated in Example 1 which shows 48 time points with no replicates, collected once every hour. If your experiment uses replicates or a different sampling density, you’ll need to make the following changes: (1) Change Line 12 to reflect the total number of time points, and the number of replicates per time point (see “Run\_JTK\_CYCLE (Example2).R”). For example, if you have 13 time points, and 2 replicates per time point, change Line 12 to read: “jtkdist(13,2). (2) Change Line 15 to reflect the spacing of your time points. For example, if your data points were collected every 2 hours, Line 15 will read: “jtk.init(periods, 2)”. (3) Change Line 14 to reflect the period lengths you’re interested in. These number are NOT measured in hours, but instead based on the number of time points per cycle. For example, if your time points are spaced 2 hours apart and you’re looking for circadian genes (20-28h) Lines 14 and 15 will read: “periods <- 10:14” and “jtk.init(periods,2)”.

*Uneven replicates:* In rare circumstances, the number of replicates per time point might not be equal for every time point (e.g. if one replicate from a large experiment is excluded as an obvious outlier). In this case, add a line of code defining an integer vector that specifies the number of samples in each replicate group (named “group.sizes”), and modify line 12 so that the function ‘jtkdist’ uses your replicate layout rather than the default, e.g. jtkdist(length(group.sizes),group.sizes).

**Running JTK.** After modifying the ‘Run\_JTK\_Cycle’ script, select all and paste into R’s graphical interface. The first half of the script (through line 19) will run quickly; the rest will take some time depending on your machine and the size of the analysis.

**JTK\_Cycle Output.** JTK\_Cycle outputs two files: (1) a .txt with all the important cycling statistics, and (2) a .rda file. The .rda file contains R objects and will not typically be needed by most users. The .txt includes BH.Q (Benjamani-Hochberg q-value), ADJ.P (p-value), PER (period), LAG (phase), and AMP (amplitude), along with the raw data to visually double-check the best hits.

**AMP Confidence Intervals.** Use ‘Run\_JTK\_Cycle (Example3).R’ within this folder. The key difference are (1) “jtkx(z, ampci=TRUE, conf=0.95)” and (2) “c(JTK.ADJP, JTK.PERIOD, JTK.LAG, JTK.AMP, JTK.AMP.CI, JTK.AMP.PVAL)”. The first change sets ‘ampci’ (calculation of amplitude confidence) as ‘TRUE’ and the desired confidence interval (in this case, 95%). The second receives calculated amplitude confidence. Importantly, JTK\_Cycle should work exactly as described above, except the amplitude is estimated through ‘wilcox.test’ and confidence intervals for AMP (AMP.Hi and AMP.Lo) are added to the output. In addition, calculation of “AMP confidence intervals” will decrease JTK\_Cycle’s computational efficiency a lot.

**Handling Missing Values.** Use ‘Run\_JTK\_Cycle (Example4&5).R’ in this folder. If there are missing samples at some time points, several columns of missing values will exist in the data (like “Example4\_data.txt”). If studied molecules (eg. transcripts or proteins) are randomly un-detectable in some samples, missing values will randomly exist in the data (like “Example5\_data.txt”). Users could follow the instruction mentioned above (“Exampe 1” and “Example 2”) to analyzed datasets with missing values. The minor modification in ‘Run\_JTK\_Cycle (Example4&5).R’ help users who do not have annotation file. JTK\_Cylce runs ‘jtkdist()’ several times in analyzing datasets with randomly missing values, so it takes more time in analyzing this kind of dataset.

**Citations.**

When using JTK\_Cycle, please cite: Hughes, Hogenesch, Kornacker (2010) JBR.

When using the AMP confidence intervals, please additionally cite: Miyazaki, Schroder, Edelmann, Hughes, Kornacker, Balke, Esser (2011). PLoS ONE.

**Troubleshooting.**

1. *“Error in data.frame(..., check.names = FALSE) : arguments imply differing number of rows:”*

Verify that your data and annotation files have the same number of rows.

1. *“Error: all(is.finite(z)) is not TRUE”*

Verify that there are numerical values for every entry in your data file.

1. *“Error in cf[[u]] :index out of bound”*

Don’t look for period lengths shorter than your sampling resolution.

Good luck!