**Supplementary Materials**

**Spatial Disorder Analysis Code for Python**

This is a short python module that provides a single function that computes the sample entropy of normalized-time x-coordinate vector x, with tolerance r, and window size m (to be compared with m + 1). Researchers need only import their data into Python, where each trajectory is either a list or NumPy array x. This function may then be used to obtain Sample Entropy per trajectory, to be used in statistical analyses. Recommended tolerance r is the standard deviation of x-shifts ($\Delta x$) across conditions, which may be calculated using the NumPy standard deviation function std() and the x-shifts function in this module xshifts(). This module is for use with Python 2.7 or greater, and both NumPy and Math modules in Python.

Available for download at: [https://freemanlab.net/GPIR/SampEn.py](https://freemanlab.net/GPIR/SampEn.py)
Or: [https://github.com/rystoli/psychtools/blob/master/MT/SampEn.py](https://github.com/rystoli/psychtools/blob/master/MT/SampEn.py)

Sample call to calculate Sample Entropy for a single trajectory x, with window size m of 3, and a tolerance r of .2 multiplied by the standard deviation of $\Delta x$:

```python
SampEn( x , 3 , .2 * std( xshifts( x ) ) )
```

```python
from math import *
from numpy import *

def xshifts( x ):
    '''
    Returns vector of N-1 step-wise changes in x value ( xt+1 – xt )
    x: vector of normalized-times decision axis location (x-values)
    '''
    xs = [ x[ i + 1 ] - x[ i ] for i in range( len( x ) - 1 ) ]
    return xs

def windowmaker( xs, m ):
    '''
    Returns all possible summed windows of sequential x-shift values
    xs: N-1 x-shifts vector
    m: window size
    '''
    w = [ xs[ i : i + m ] for i in range( len( xs ) - m - 1 ) ]
    return w

def pairindices( w, r ):
    '''
    Returns list of indices of similar pairs of windows in windows list
```
w: list/array of window lists e.g., via windowmaker()
r: tolerance

```
pi = []
for i, s in enumerate( w[ : -1 ]):
    for j, s2 in enumerate( w[ i + 1 : ]):
        if max( abs( array( s ) - array( s2 ) ) ) <= r: pi.append( [ i , j ] )
return pi
```

def Mcounter( pim , pim1 ):
    
    Returns count of similar windows retained between two window sizes (e.g., m, m+1)

    pim: list of indices of similar windows in the smaller window size
    pim1: list of indices of similar windows in the larger window size
    
    Mm1 = 0
    for i in pim1:
    if i in pim and i in pim1: Mm1 += 1
    return Mm1

def SampEn( x, m, r ):
    
    Returns Sample Entropy (pairwise) of vector x

    x: vector (list or NumPy array) of normalized times decision axis location (x-values)
    m: window size
    r: tolerance
    
    dx = xshifts(x)
    wm, wm1 = windowmaker( dx , m )[ : -1 ], windowmaker( dx , m + 1 )
    pim, pim1 = pairindices( wm , r ), pairindices( wm1 , r )
    Mm , Mm1 = len( pim ), Mcounter( pim , pim1 )
    e = -log( float( Mm1 ) / float( Mm ) )
    return e

**Principle Components Analysis Syntax for SPSS**

SPSS syntax for principle components analysis using Varimax rotation. Here, we use raw x-coordinates with 75 ms time bins and a 1500 ms cut off for analysis (red font). However, researchers might replace these variables with any factors of interest (e.g., velocity, proximity, y-coordinates, etc.)

FACTOR
/VARIABLES X_1_75ms X_76_150ms X_151_225ms X_226_300ms X_301_375ms X_376_450ms X_451_525ms X_526_600ms X_601_675ms X_676_750ms X_751_825ms X_826_900ms X_901_975ms X_976_1050ms X_1051_1125ms X_1126_1200ms X_1201_1275ms X_1276_1350ms X_1351_1425ms X_1426_1500ms
/MISSING LISTWISE
/ANALYSIS X_1_75ms X_76_150ms X_151_225ms X_226_300ms X_301_375ms X_376_450ms X_451_525ms X_526_600ms X_601_675ms X_676_750ms X_751_825ms X_826_900ms X_901_975ms X_976_1050ms X_1051_1125ms X_1126_1200ms X_1201_1275ms X_1276_1350ms X_1351_1425ms X_1426_1500ms
/PRINT INITIAL EXTRACTION ROTATION
/FORMAT SORT
/PLOT EIGEN
/CRITERIA MINEIGEN(1) ITERATE(25)
/EXTRACTION PC
/CRITERIA ITERATE(25)
/ROTATION VARIMAX
/METHOD=CORRELATION.