TUTORIAL FOR "TORUS GRAPHS FOR MULTIVARIATE PHASE COUPLING ANALYSIS"

By Natalie Klein[‡,§,*], Josue Orellana[¶,§,*], Scott Brincat[‖], Earl K. Miller[‖], and Robert E. Kass[‡,¶,§,†]

Department of Statistics and Data Science, Carnegie Mellon University[‡], Machine Learning Department, Carnegie Mellon University[§], Center for the Neural Basis of Cognition[¶], and Department of Brain and Cognitive Science, Massachusetts Institute of Technology[‖]

This tutorial describes the implementation of torus graphs and gives detail on how to use the code. The code is implemented in Matlab.

You can use `start_here_results_fig8_S8.m` to follow Sections 1-7 of this tutorial and obtain the results for Fig. 8 in the main text and Fig. S8 in the supplement.

The rest of the code is organized as follows:

- `results_fig4_figS10.m`
  Realistic simulated data sets based on positive rotational dependence, dimensions 3 and 5.
- `results_fig5_figS4.m`
  Investigation of the ability of torus graphs to recover the true structure as a function of true edge density, sample size, and data dimension.
- `results_fig7_figS7.m`
  Torus graphs and PLV graphs to infer low-dimensional networks of interest in real LFP data.
- `results_fig9.m`
  Comparison between phase angles from three LFPs located in PFC and the theoretical torus graph distribution.
- `results_figS11.m`
  Investigation of False Positive Rate (FPR) and False Negative Rate (FNR) for graphs of varying dimensions as sample size increases.

[*]N.K. and J.O. contributed equally to this work.
[†]To whom correspondence should be addressed. E-mail: kass@stat.cmu.edu
Keywords and phrases: graphical models, circular statistics, network analysis

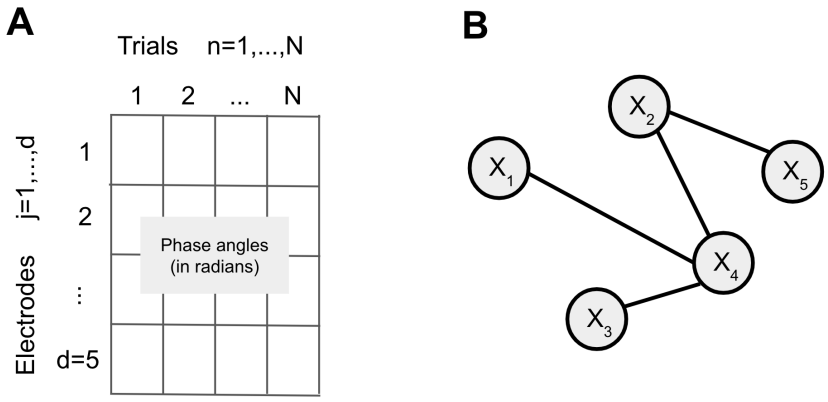

FIG T1. *(A) Input to obtain a TG: data matrix of angle observations over repeated trials. (B) Example TG; this is a conditional independence graph, meaning there are no edges between pairs of nodes that are conditionally independent given the other nodes in the network.*

T1. Torus Graph (TG) model, basic usage. Let $\mathbf{X} \in [-\pi, \pi)^{d\times 1}$ be a d-dimensional circular random vector with $d \geq 2$[1]. The word "circular" in this context refers to angles, thus we consider $j = 1, .., d$ simultaneous angle measurements. We are interested in phase coupling at a fixed frequency and time point over measurements of the same experimental condition (repeated trials). The input to obtain a TG is the data matrix of angle observations: $\mathbf{x} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$ (Figure T1).

The function `TG=torus_graphs(x)` fits a TG model using the observed data x in the same format as in Figure T1. The output TG is a binary "graph with undirected edges" object, see https://www.mathworks.com/help/matlab/ref/graph.html.

T2. Edgewise statistical testing. To obtain the binary undirected graph, we perform edgewise statistical tests. Briefly, for a pair of nodes, the null hypothesis is that of conditional independence given the other nodes in the network (no edge). For each edge we obtain a p-value using the results of Lemma 5.1 in the main text. We reject the null hypothesis if the p-value is less than the predefined alpha level.

[1]The TG function works for any choice of circular domain in radians, such as $[0, 2\pi)$.

You can define an alpha level in the second input of the torus_graphs function as: `[TG, edges]=torus_graphs(x, alpha_level)`. The default value is `alpha_level=0.05`. The second output, edges, is a structure that contains three fields: all possible edges between node pairs (`egdes.all_possible`), corresponding p-values (`egdes.p_vals`), and edges for which we reject the null hypothesis, `p_val<alpha_level` (`edges.active`). For the phase difference model with uniform margins, the edges structure also contains `egdes.cond_coupling_coeff`, see Section T5 below and Section 4.4. in the main text for details.

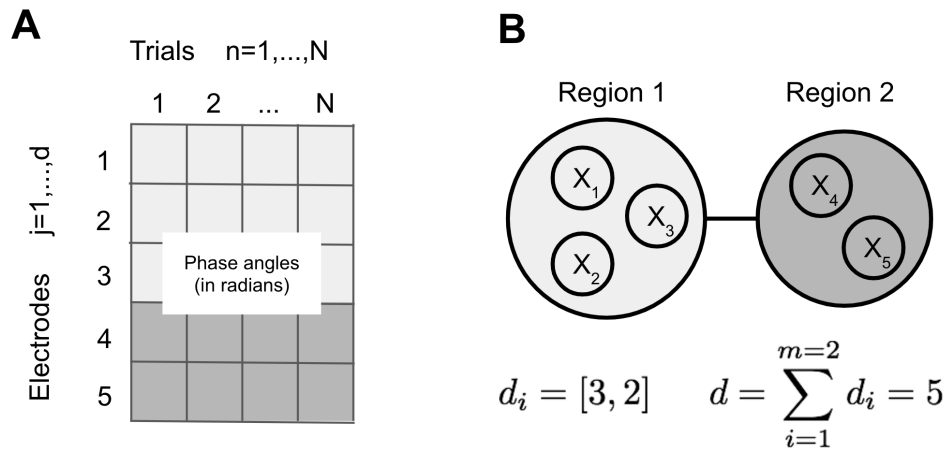

FIG T2. *(A) Example of data matrix input to TG with electrodes grouped using labels d_i. (B) Example output graph. Only the cross-region edge decisions are given when you provide grouping information.*

T3. Grouping electrodes by brain region. If the nodes can be grouped according to brain regions, and those group labels are available, the `torus_graphs` function can take that information to output a network with as many nodes as brain regions (see Figure T2).

You can use `[TG, edges]=torus_graphs(X,alpha_level,di)` where the third input is $d_i = [d_1, d_2, ..., d_m]$, and has the number of electrodes in each of m regions (where the dimensions are ordered by regions, as shown in Figure T2). Note that $d = \sum_{i=1}^{m} d_i$.

T4. TG probability density function. Using the natural parameters $\boldsymbol{\phi}$, the TG density is:

$$p(\mathbf{x}|\boldsymbol{\phi}) \propto \exp\left(\sum_{j=1}^{d} \boldsymbol{\phi}_j^T \begin{bmatrix}\cos x_j \\ \sin x_j\end{bmatrix} + \sum_{j<k} \boldsymbol{\phi}_{jk}^T \begin{bmatrix}\cos(x_j - x_k) \\ \sin(x_j - x_k) \\ \cos(x_j + x_k) \\ \sin(x_j + x_k)\end{bmatrix}\right) \tag{T4.1}$$

We define a *d-dimensional torus graph* to be any member of the family of distributions specified by Equation T4.1. The sufficient statistics involving only a single angle are

$$\mathbf{S}_j^1(\mathbf{x}) = [\cos(x_j), \sin(x_j)]^T,$$

and the sufficient statistics involving pairs of angles are

$$\mathbf{S}_{jk}^2(\mathbf{x}) = [\cos(x_j - x_k), \sin(x_j - x_k), \cos(x_j + x_k), \sin(x_j + x_k)]^T.$$

We will use $\boldsymbol{\phi}$ and $\mathbf{S} \equiv [\mathbf{S}^1, \mathbf{S}^2]$ to refer to the full vectors of parameters and sufficient statistics for all angles. Because each angle has two marginal parameters, and each unique pair of angles has four coupling parameters, we have $2d + 4[d(d-1)/2] = 2d^2$ parameters. The vector of natural parameters is real-valued, i.e. $\boldsymbol{\phi} \in \mathbb{R}^{2d^2}$.

In `[TG,edges,phi_hat,inference]=torus_graphs(X,alpha_level,di)`, the third output (`phi_hat`) is the vector of estimated ϕ parameters. The fourth output (`inference`) is a structure containing three fields: the asymptotic covariance estimate for ϕ (`inference.Sigma_hat`), the test statistic for each edge group (`inference.t_hat`), and the number of degrees of freedom for each edge group (`inference.dofs`).

T5. TG submodels. If you have reason to believe that your data could be described by a submodel, you can fit a submodel by setting some of the entries of ϕ, corresponding to specific types of sufficient statistics, to zero. For instance, a phase difference submodel sets the parameters corresponding to phase sums to zero, while a uniform marginal model sets the parameters corresponding to marginal concentrations to zero.

In `[TG,edges,phi_hat,inference]=torus_graphs(X,alpha_level,di,sel_mode)`, the fourth input is a 1 by 3 Boolean array to select a submodel. The first entry corresponds to marginal concentrations, the second entry to phase difference concentrations, and the third entry to phase sum concentrations. The default is to fit `[true, true, true]`, which corresponds to the full TG model; `[false, true, true]` fits a uniform marginal model, `[true, true, false]` fits a phase difference model.

T6. Exploratory data analysis. To guide selection of a submodel, we can use the Rayleigh test for uniformity applied on the concentrations of the angles (marginally), on the pairwise phase differences, and on the pairwise phase sums. The null hypothesis of this test is that the input vector of angles is uniformly distributed on the circle, i.e. zero marginal concentration. We will have d p-values for the the tests of marginal concentrations and $d(d-1)/2$ p-values for phase sums or phase differences. Instead of interpreting the p-values individually, we combine them using Fisher's method for combining p-values. Then for example, if the marginal concentration p-value is < 0.05, we might choose to fit a model with marginal concentrations, but not if $p > 0.05$.

You can use the following function to get combined p-values (Fisher's method) from Rayleigh tests: `[marginal, phase_diffs, phase_sums]= submodel_eda(x)`. The input is the data matrix `x` and, for each case, the output is a structure with two fields: individual p-values and aggregated p-values.

T7. Sampling from a TG. As detailed in the main text, we can use a Gibbs sampler to sample from a TG distribution.

You can use the function `Xsamp=sampleGibbs_tutorial(d,phi_hat,opt)` to draw samples from a torus graph. The output is a data matrix of samples. The inputs are the dimension (`d`), a vector of parameters (`phi_hat`), and a Gibbs sampler options structure (`opt`) with fields for number of desired samples, burn-in, and thinning. Note that higher dimensional models may require more samples to accurately represent the target distribution.

N. Klein
J. Orellana
R. E. Kass
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
E-mail: neklein@stat.cmu.edu
josue@cmu.edu
kass@stat.cmu.edu

S. Brincat
E. K. Miller
The Picower Institute for Learning and Memory
Department of Brain and Cognitive Science
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
E-mail: sbrincat@mit.edu
ekmiller@mit.edu