Detecting Multivariate Cross-Correlation Between Brain Regions

Jordan Rodu[1*], Natalie Klein[2,3], Scott L. Brincat[4,5], Earl K. Miller[4,5], Robert E. Kass[2,3,6]

1 Department of Statistics, University of Virginia, Charlottesville, VA, USA
2 Department of Statistics, Carnegie Mellon University, Pittsburgh, PA, USA
3 Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA
4 The Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, MA, USA
5 Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA
6 Center for the Neural Basis of Cognition, Carnegie Mellon University/University of Pittsburgh, Pittsburgh, PA, USA

* jsr6q@virginia.edu

Abstract

The problem of identifying functional connectivity from multiple time series data recorded in each of two or more brain areas arises in many neuroscientific investigations. For a single stationary time series in each of two brain areas statistical tools such as cross-correlation and Granger causality may be applied. On the other hand, to examine multivariate interactions at a single time point, canonical correlation, which finds the linear combinations of signals that maximize the correlation, may be used. We report here a new method that produces interpretations much like these standard techniques and, in addition, (a) extends the idea of canonical correlation to 3-way arrays (with dimensionality number of signals by number of time points by number of trials), (b) allows for non-stationarity, (c) also allows for nonlinearity, (d) scales well as the number of signals increases, and (e) captures predictive relationships, as is done with Granger causality. We demonstrate the effectiveness of the method through simulation studies, and illustrate by analyzing local field potentials recorded from a behaving primate.

New and Noteworthy

Multiple signals recorded from each of multiple brain regions may contain information about cross-region interactions. This article provides a method for visualizing the complicated interdependencies contained in these signals, and assessing them statistically. The method combines signals optimally, but allows the resulting measure of dependence to change, both within and between regions, as the responses evolve dynamically across time. We demonstrate the effectiveness of the method through numerical simulations and by uncovering a novel connectivity pattern between hippocampus and prefrontal cortex during a declarative memory task.

Introduction

When recordings from multiple electrode arrays are used to establish functional connectivity across two or more brain regions, multiple signals within each brain region must be considered. If, for example, local field potential (LFP) signals in each of two regions are examined, the problem is to describe the multivariate relationship between all the signals from the first region and all the signals from the second region, as it evolves across time, during a task. One possibility is to take averages across signals in each region and then apply cross-correlation or Granger causality (Brovelli et al., 2004; Granger, 1969). Alternatively, one might apply these techniques across all pairs involving one signal from each of the two regions, and then average the results. Such averaging, however, may lose important information, as when only a subset of the time series from one region correlates well with a subset of time series from the other region. Furthermore, Granger causality is delicate in the sense that it can be misleading in some common situations (Wang et al., 2008; Ding and Wang, 2014; Barnett and Seth, 2011). We have developed and investigated a new method, which is descriptive (as opposed to involving generative models such as auto-regressive processes used in Granger causality), and is capable of finding subtle multivariate interactions among signals that are highly non-stationary due to stimulus or behavioral effects.

Our approach begins with the familiar cross-correlogram, which is used to understand the correlation of two univariate signals, including their lead-lag relationship, and generalizes this in two ways: first, we extend it to a pair of multivariate signals using the standard multivariate technique known as canonical correlation analysis (CCA); second, we allow the correlation structure to evolve dynamically across time. In addition, we found that a comparatively recent variation on CCA, known as kernel-CCA (KCCA), provides a more flexible and computationally efficient framework. We call the initial CCA-based method dynamic CCA (DCCA) and the kernel-based version dynamic KCCA (DKCCA).

We assume the signals of interest are recorded across multiple experimental trials, and the correlations we examine measure the tendency of signals to vary together across trials: a positive correlation between two of the signals would indicate that trials on which the first signal is larger than average, tend also to be trials on which the second signal is larger than average. At a single point in time we could measure the correlation (across trials) between any two signals. At a single point in time we could also take any linear combination of signals in one region and correlate it with a linear combination of signals in the other region; the canonical correlation is the maximum such correlation among all possible linear combinations. A technical challenge is to find a way to compute canonical correlation while taking into account multiple time points at which the signals are collected.

In a different context, Lu (2013) proposed time-invariant weights over both multivariate signals and time points, such that the resulting vectors are maximally correlated. This does not solve satisfactorily the problem we face because it ignores the natural ordering of time and can therefore produce non-physiological combinations. Another proposal (Bießmann et al., 2010) applies to individual trials, and therefore examines correlation across time (as opposed to correlation across trials), which requires signals that are stationary (time-invariant), whereas we wish to describe the dynamic evolution of their correlation across time. Our DKCCA solution applies KCCA in sliding windows across time, similarly to the way a spectrogram computes frequency decompositions in sliding windows across time. After describing the method, we evaluate DKCCA on simulated data, where there is ground truth, and show that DKCCA can recover correlation structure where simpler averaging methods fail. We then apply DKCCA to data collected simultaneously from the prefrontal cortex and hippocampus of a rhesus macaque, and we uncover a novel connectivity pattern that is not detected by traditional averaging methods.

Materials and Methods

In this section, we first review CCA and kernel CCA, and then describe our algorithms, DCCA and DKCCA, our artificial-data simulations, and our experimental methods.

CCA

CCA (Hotelling, 1936) provides a natural way to examine correlation between two sets of N multivariate observations. The algorithm finds maximally correlated linear combinations of the variables in each set, reducing a multivariate correlation analysis into several orthogonal univariate analyses. Specifically, given two multivariate, zero mean data sets $X \in \mathbb{R}^{N\times q_x}$ and $Y \in \mathbb{R}^{N\times q_y}$, where q_x and q_y are the number of variables in X and Y respectively, CCA seeks to find canonical weights w_X and w_Y, the coefficients for linear combinations of the columns of X and Y, respectively, such that

$$\rho_1 = \max_{w_X^1, w_Y^1} \frac{w_X^{1\top} X^\top Y w_Y^1}{\sqrt{w_X^{1\top} X^\top X w_X^1 \cdot w_Y^{1\top} Y^\top Y w_Y^1}} \tag{1}$$

where ρ_1 indicates the first canonical correlation and w_X^1, w_Y^1 the first canonical weight pairs. Successive canonical correlations and weight pairs are similarly defined such that

$$\rho_p = \max_{w_X^p, w_Y^p} \frac{w_X^{p\top} X^\top Y w_Y^p}{\sqrt{w_X^{p\top} X^\top X w_X^p \cdot w_Y^{p\top} Y^\top Y w_Y^p}} \tag{2}$$

additionally satisfying

$$\begin{aligned} \mathrm{corr}\left(Xw_X^i, Xw_X^j\right) &= 0 \\ \mathrm{corr}\left(Yw_Y^i, Yw_Y^j\right) &= 0, \quad i \neq j \\ \mathrm{corr}\left(Xw_X^i, Yw_Y^j\right) &= 0. \end{aligned} \tag{3}$$

Further, $p \leq \min(q_x, q_y)$ but often in practice $p << \min(q_x, q_y)$. Essentially, each canonical weight pair from CCA projects the two multivariate sets of N observations to two univariate sets of N observations, called canonical components, from which correlation can be obtained in the usual way.

Kernel CCA

Kernel CCA (Hardoon et al., 2004) extends CCA to allow for nonlinear combinations of the variables, and it also remains numerically stable while CCA can become unstable with large numbers of variables. Importantly, even though kernel CCA applies CCA to a transformation of the data determined by the nonlinearity of interest, all calculations can be performed using inner products of the transformed data matrices. These inner products are defined by a kernel function and are collected in a matrix often called the kernel matrix. Because the resulting computations are efficient, and avoid explicit calculation of the nonlinear transformations, this is usually called the "kernel trick." It has been studied extensively (Christopher, 2016). We next make this explicit in our context.

Following the example of Hardoon et al. (2004), kernel CCA maps observations into a "feature space"

$$\phi : x \longrightarrow (\phi_1(x), \phi_2(x), \ldots, \phi_M(x)) \tag{4}$$

on which CCA is performed. Using the fact that the weights $w_{\phi(X)}$ and $w_{\phi(Y)}$ lie in the row space of $\phi(X)$ and $\phi(Y)$ respectively,

$$\begin{aligned} w_{\phi(X)} &= \phi(X)^\top \alpha \\ w_{\phi(Y)} &= \phi(Y)^\top \beta \end{aligned} \tag{5}$$

and substituting equation 5 into equation 1, we have

$$\rho_1 = \max_{\alpha_1, \beta_1} \frac{\alpha_1^\top \phi(X)\phi(X)^\top \phi(Y)\phi(Y)^\top \beta_1}{\sqrt{\begin{array}{c} \alpha_1 \phi(X)\phi(X)^\top \phi(X)\phi(X)^\top \alpha_1 \\ \cdot \beta_1^\top \phi(Y)\phi(Y)^\top \phi(Y)\phi(Y)^\top \beta_1 \end{array}}}. \tag{6}$$

Rewriting $\phi(X)\phi(X)^\top$ as K_X and $\phi(Y)\phi(Y)^\top$ as K_Y, we can express equation 6 as

$$\rho_1 = \max_{\alpha_1, \beta_1} \frac{\alpha_1^\top K_X K_Y \beta_1}{\sqrt{\alpha_1 K_X^2 \alpha_1 \cdot \beta_1^\top K_Y^2 \beta_1}} \tag{7}$$

with successive canonical correlations found as in plain CCA. The Gram matrices K_X and K_Y require regularization, and the regularization parameter is typically set using cross-validation.

Instead of calculating the canonical weights, canonical components can be calculated directly using

$$\begin{aligned} \phi(X)w_{\phi(X)}^1 &= K_X \alpha_1 \\ \phi(Y)w_{\phi(Y)}^1 &= K_Y \beta_1. \end{aligned} \tag{8}$$

In this paper we use the linear kernel, which returns weights that can be interpreted like they are in plain CCA. Even with the linear kernel, kernel CCA provides benefits when the number of variables in the data matrices is larger — in our case, much larger — than the number of observations. Kernel CCA requires the estimation and use of the kernel matrices of size $N \times N$, rather than the much larger matrices of size $q_x \times q_x$, $q_y \times q_y$, and $q_x \times q_x$. Further, as we describe below, the Gram matrices used in kernel CCA permit a decomposition specific to our method that is not possible with plain CCA, and that allows for substantial improvement in computational speed.

Dynamic Canonical Correlation Analysis

We first describe DCCA and DKCCA. Then, we show how to integrate multiple components of CCA analysis into the DCCA framework, and determine significance of observed canonical correlations based on a permutation test.

To motivate the problem and our solution, we first describe the procedure for assessing correlation structure in the case of two univariate signals. Suppose we have two simultaneously recorded univariate time series of length T over N repeated trials collected into matrices $X \in \mathbb{R}^{N\times T}$ and $Y \in \mathbb{R}^{N\times T}$. Let $X(i,s)$ and $Y(i,t)$ denote the i^{th} trial at times s and t in X and Y, then the the cross-correlation between times s and t is

$$\text{cc}(s,t) = \frac{\sum_{i=1}^{N}\big(X(i,s)-\bar{X}(s)\big)\big(Y(i,t)-\bar{Y}(t)\big)}{\sqrt{\sum_{i=1}^{N}\big(X(i,s)-\bar{X}(s)\big)^2\sum_{i=1}^{N}\big(Y(i,t)-\bar{Y}(t)\big)^2}} \tag{9}$$

where $\bar{X}(s)=\frac{1}{N}\sum_{i=1}^{N}X(i,s)$ and $\bar{Y}(t)=\frac{1}{N}\sum_{i=1}^{N}Y(i,t)$. Our goal is to extend this to a multivariate version

$$\text{ccc}(s,t) = \max_{w_{X(s,t)},w_{Y(s,t)}} \frac{w_{X(s,t)}^{\top}\mathbf{X}(s)^{\top}\mathbf{Y}(t)w_{Y(s,t)}}{\sqrt{\chi(s,t)\cdot\gamma(s,t)}} \tag{10}$$

$$\begin{aligned}\text{with }\ \chi(s,t) &= w_{X(s,t)}^{\top}\mathbf{X}(s)^{\top}\mathbf{X}(s)w_{X(s,t)}\\ \gamma(s,t) &= w_{Y(s,t)}^{\top}\mathbf{Y}(t)^{\top}\mathbf{Y}(t)w_{Y(s,t)}.\end{aligned}$$

where arrays $\mathbf{X}\in\mathbb{R}^{q_x\times T\times N}$ and $\mathbf{Y}\in\mathbb{R}^{q_Y\times T\times N}$ collect the N multivariate time series of length T of dimensions q_x and q_Y respectively, and $\mathbf{X}(s)\in\mathbb{R}^{N\times q_x}$ and $\mathbf{Y}(t)\in\mathbb{R}^{N\times q_y}$ denote slices of arrays $\mathbf{X}$ and $\mathbf{Y}$ corresponding to times s and t. The challenge is to identify linear combinations $w_{X(s,t)}$ and $w_{Y(s,t)}$ for all s,t. As mentioned in the introduction, a natural first thought is to solve for $w_{X(s,t)}$ and $w_{Y(s,t)}$ by using CCA for every pair of times s and t, but this leads to correlations and weights that are difficult to interpret. Alternatively we could find fixed $w_{X(s,t)}$ and $w_{Y(s,t)}$ for all s,t, but this does not take into account the nonstationarity of signals in the brain. DCCA, which we now describe, solves these problems by inferring a single set of weights that modulate over time without knowing a priori the lagged correlation structure of the system.

We start by creating extended observations at each time point that are a concatenation of observations in a local window of time. Fix local window g, and let $\mathbf{X}(s')\in\mathbb{R}^{N\times q_x\cdot(2g+1)}$ and $\mathbf{Y}(s')\in\mathbb{R}^{N\times q_y\cdot(2g+1)}$ be matrices defined as

$$\begin{aligned}\mathbf{X}(s') &= \big[X(s-g),X(s-g+1),\ldots,X(s), \\ &\qquad \ldots,X(s+g-1),X(s+g)\big] \\ \mathbf{Y}(s') &= \big[Y(s-g),Y(s-g+1),\ldots,Y(s), \\ &\qquad \ldots,Y(s+g-1),Y(s+g)\big]\end{aligned} \tag{11}$$

where $\big[A,\ldots Z\big]$ indicates concatenation of matrices $A,\ldots,Z$ along columns. For each time point s, we run a CCA between $\mathbf{X}(s')$ and $\mathbf{Y}(s')$, which yields linear combination weights $w_{X(s')}$ and $w_{Y(s')}$ of lengths $q_x\cdot(2g+1)$ and $q_y\cdot(2g+1)$ respectively (see equation 1). Since the CCA is run using concatenated observations matrices $\mathbf{X}(s')$ and $\mathbf{Y}(s')$, we can express the concatenated weights as

$$\begin{aligned} w_{X(s')} &= \begin{bmatrix} w'_{X(s-g)} \\ \vdots \\ w'_{X(s)} \\ \vdots \\ w'_{X(s+g)} \end{bmatrix} \in \mathbb{R}^{q_x\cdot(2g+1)\times 1} \\ w_{Y(s')} &= \begin{bmatrix} w'_{Y(s-g)} \\ \vdots \\ w'_{Y(s)} \\ \vdots \\ w'_{Y(s+g)} \end{bmatrix} \in \mathbb{R}^{q_y\cdot(2g+1)\times 1} \end{aligned} \tag{12}$$

where the w' emphasizes that these weights are not the same weights that would be obtained from a CCA with observations that are not concatenated. We then set the canonical weights for $X(s)$ and $Y(s)$ equal to $w'_{X(s)}$ and $w'_{Y(s)}$ in equation 12 and calculate matrices $X^{\text{proj}}\in\mathbb{R}^{N\times T}$ and $Y^{\text{proj}}\in\mathbb{R}^{N\times T}$ where

$$\begin{aligned} X^{\text{proj}}(s) &= X(s)w'_{X(s)} \\ Y^{\text{proj}}(s) &= Y(s)w'_{Y(s)} \end{aligned} \tag{13}$$

and the superscript label is used to denote projection. Finally, we calculate the matrix *ccc* as in equation 9 using matrices X^{proj} and X^{proj}.

In this approach, exact lagged relationships need not be established apriori. Instead, if there is a strong lagged correlation between X and Y at times s and t, then as long as $|s-t|<g$, this lagged relationship will be a driving factor in setting the linear combination weights for both X at time s and Y at time t (though these will not be set simultaneously). Further, the strong correlation between X and Y at times s and t will drive setting the linear combination weights at time s' where $0<|s'-s|<g$ and $0<|s'-t|<g$, instead of $w_X(s')$ and $w_Y(s')$ being set to greedily maximize correlation at time s'.

Extended observations are the key to discovering a priori unknown lagged cross-correlation, as they make observations within the specified local window visible to each other. To see this, if array $\mathbf{X}$ has observations that are independent of each other across time, and $\mathbf{Y}$ is a *shifted* copy of $\mathbf{X}$, so $\mathbf{Y}(t)=\mathbf{X}(t-k)$, then without creating extended observations, this procedure would only return false correlations since in the shifted case the instantaneous observations are independent (and hence uncorrelated) by design. Recovery of the actual lagged correlation without extended observations would require knowledge of the true lag, which in the above example would be k.

While DCCA with CCA might work well in some scenarios, as described above it has some disadvantages. First, in many cases, $q_x\cdot(2g+1) >> N$ and $q_y\cdot(2g+1) >> N$, which causes numerical instability. Second, as a result of forming concatenated matrices $\mathbf{X}(s')$ and $\mathbf{Y}(s')$, we must estimate a covariance matrix for all observation pairs $X(s)$ and $X(t)$, $Y(s)$ and $Y(t)$, and $X(s)$ and $Y(t)$ for $|s-t|<2g+1$.

Finally, for inference involving trial permutations or bootstrapping, these covariance matrices must be recalculated for each simulated dataset.

The first and third problems are well understood from the literature, and the solution is to use kernel CCA (KCCA) instead of CCA. In the next section, we describe DCCA with KCCA and show that not only is KCCA a solution to the first and third problems with respect to the DCCA procedure, but also the second. Further, KCCA allows for nonlinear combinations over signals.

Dynamic Kernel Canonical Correlation Analysis

In theory, DKCCA proceeds as described above, but with KCCA instead of CCA. Observations are concatenated over a local window of time, $s-g$ to $s+g$, and kernel matrices are computed from these concatenated observations. Weights for the concatenated observations (see equation 12) are derived according to equation 5, from which the weights at time s are extracted and matrices X^{proj} and Y^{proj} are created as in equation 13.

As indicated in equation 7, the KCCA algorithm finds optimal $\alpha(s)$ and $\beta(s)$, which from equation 5 we can interpret as coefficients for the linear combinations over the N replications in $X(s)$ and $Y(s)$ respectively (see, for instance, Hardoon et al. (2004) for details). Because the concatenated observation matrix $\mathbf{X}((s+1)')$ contains all the observations in the matrix $\mathbf{X}(s')$ with the exception of one, and likewise for matrices $\mathbf{Y}((s+1)')$ and $\mathbf{Y}(s')$, $\alpha(s)$ and $\beta(s)$ modulate smoothly over time in the DKCCA procedure. As a consequence, using equation 5, we have that $w_{X(s)}$ and $w_{Y(s)}$ evolve smoothly over time as well. Note that the degree of smoothness depends on the smoothness of the original time series.

Calculating the canonical weights in equation 5, while informative in many scientific settings, is not strictly necessary. In cases where the focus lies solely on the temporal lagged correlations, $X^{\text{proj}}(s)$ and $Y^{\text{proj}}(s)$ in equation 13 can be calculated directly using equation 8. This is especially useful when ϕ maps observations into high-dimensional or infinite-dimensional space, where the weights become difficult to interpret. Calculating $X^{\text{proj}}(s)$ and $Y^{\text{proj}}(s)$ directly allows for a wide range of nonlinear combinations of the time series to be considered with low computational cost.

In addition to extending CCA to nonlinear combinations, KCCA allows for numerical stability when the number of signals multiplied by concatenated time points is larger than the number of trials. Also, bootstrap and permutation testing procedures are much faster with KCCA, since they amount to selection or permutation of the rows and columns of the kernel matrices. Further, KCCA solves the second problem mentioned above, that of needing to estimate a covariance matrix for all observation pairs $X(s)$ and $X(t)$, $Y(s)$ and $Y(t)$, and $X(s)$ and $Y(t)$ for $|s-t| < 2g+1$ under DCCA with CCA. Focusing for now on the case where ϕ is the identity map, we calculate the kernel matrix from equation 7, $K_X(s') \in \mathbb{R}^{N\times N}$, as $K_X(s') = \mathbf{X}(s')\mathbf{X}^\top(s')$ (likewise with the kernel matrix $K_Y(s')$). This calculation can be decomposed as

$$K_X(s') = K_X(s-g) + \ldots + K_X(s) + \ldots + K_X(s+g) \tag{14}$$

and since in equation 7 only K_X and K_Y need to be computed (whereas in CCA the cross-covariance terms between X and Y must be computed), then for each time point s, $K_X(s)$ and $K_Y(s)$ need be computed only once, with $K_X(s')$ and $K_Y(s')$ computed as the sum of the relevant kernel matrices. For arbitrary ϕ, the decomposition in equation 14 does not necessarily hold, as interactions between time points are potentially allowed to occur. To take advantage of the decomposition, we can impose a constraint on ϕ that allows for nonlinearities across time series dimensions, but not across time.

Incorporating multiple components

Typically in CCA-based methods, it suffices to describe the procedure using only the first canonical correlation, as canonical correlations associated with components $p \geq 2$ can be studied independently, or can be added to get the total canonical correlation over all components. This is because the canonical components satisfy the constraints in equation 3. However, in DKCCA, we don't have these guarantees. While the constraints in equation 3 do apply to the canonical weights calculated from the concatenated observations (equation 12), they do not apply to the canonical componets $X^{\text{proj}}(s)$ and $Y^{\text{proj}}(s)$ in equation 13 as these are derived from *extracted* weights. As an analogy, a set of orthogonal bases over the interval $(0,1)$ are not necessarily orthogonal when restricted to an arbitrary interval $(a,b) \subset (0,1)$. In order to incorporate canonical correlations across multiple components in DKCCA, we therefore cannot add the *ccc* matrices calculated from each component.

We now describe a way of combining the top k correlations between components at times s and t in X and Y, respectively. Let $w^i_{X(s)}$ and $w^i_{Y(s)}$ be the i^{th} canonical weights for X and Y at time s, and similarly to equation 13, for $i = 1\ldots k$, let $X_i^{\text{proj}}(s) = X(s)w^i_{X(s)}$ and $Y_i^{\text{proj}}(t) = Y(t)w^i_{Y(t)}$ be the k components for $X(s)$ and $Y(t)$. Then,

1. Set $\gamma_1 = \text{corr}(X_1^{\text{proj}}(s), Y_1^{\text{proj}}(t))$.

2. Decompose $X_2^{\text{proj}}(s)$ and $Y_2^{\text{proj}}(t)$ into a projection onto $X_1^{\text{proj}}(s)$ and $Y_1^{\text{proj}}(t)$, respectively, and the associated residual:

$$X_2^{\perp}(s) = \frac{X_2^{\text{proj}}(s)^\top X_1^{\text{proj}}(s)}{||X_1^{\text{proj}}(s)||^2} X_1^{\text{proj}}(s) \tag{15}$$

$$X_2^{\text{resid}}(s) = X_2^{\text{proj}}(s) - X_2^{\perp}(s) \tag{16}$$

$$Y_2^{\perp}(t) = \frac{Y_2^{\text{proj}}(t)^\top Y_1^{\text{proj}}(t)}{||Y_1^{\text{proj}}(t)||^2} Y_1^{\text{proj}}(t) \tag{17}$$

$$Y_2^{\text{resid}}(t) = Y_2^{\text{proj}}(t) - Y_2^{\perp}(t). \tag{18}$$

3. Set

$$\gamma_2 = \frac{\begin{array}{c}\operatorname{cov}\big(X_2^{\text{resid}}(s), Y_2^{\text{resid}}(t)\big)\\ + \operatorname{cov}\big(X_2^{\text{resid}}(s), Y_2^{\perp}(t)\big) + \operatorname{cov}\big(X_2^{\perp}(s), Y_2^{\text{resid}}(t)\big)\end{array}}{\sqrt{\operatorname{var}\big(X_2^{\text{proj}}(s)\big)}\sqrt{\operatorname{var}\big(Y_2^{\text{proj}}(t)\big)}}. \tag{19}$$

   This is almost $\operatorname{cov}\big(X_2^{\text{proj}}(s), Y_2^{\text{proj}}(t)\big)$, however we have removed the covariance component $\operatorname{cov}\big(X_2^{\perp}(s), Y_2^{\perp}(t)\big)$, as this is redundant. Importantly, we still allow for correlations across components, where strict orthogonalization would not.

4. For the i^{th} canonical components, decompose $X_i^{\text{proj}}(s)$ into a projection onto the subspace spanned by components $\{X_1^{\text{proj}}(s), \dots, X_{i-1}^{\text{proj}}(s)\}$ and its orthogonal residual, and $Y_i^{\text{proj}}(t)$ into a projection onto the subspace spanned by components $\{Y_1^{\text{proj}}(t), \dots, Y_{i-1}^{\text{proj}}(s)\}$ and the orthogonal residual. Calculate γ_i as in step 3.

5. Finally, set $TC(s,t) = \sum_{i=1}^{k} \gamma_i$.

This procedure has the nice property that if it were performed on output from a vanilla CCA, for canonical correlations ρ_i, we have that $\gamma_i = \rho_i$.

Identifying significant regions of cross-correlation

We generate a null distribution for cross-correlations between regions with a permutation test, combined with an excursion test. We create B new data sets by randomly permuting the order of the trials of Y, rerunning DKCCA on each of the permuted data sets. This results in matrices ccc_i^* for $i \in 1 : B$. For each entry (s,t), we have B sampled correlations $\text{ccc}_i^*(s,t)$ from the null distribution, and say that point (s,t) is $\alpha_{\text{pw}}(s,t) -$ significant if its correlation value is greater than $1 - \alpha_{\text{pw}}$ percent of the B sampled correlations. As mentioned above, the relevant kernel matrices need not be recalculated for each permuted data set. Only the order of the rows and columns of kernel matrices $K_Y(s)$ are affected.

Because we are interested in broad temporal regions of correlated activity, and not isolated time point pairs, we use the B data sets from the permutation test to perform an excursion test (see Xu et al. (2011)). We say that two α_{pw} significant points (s,t) and (s',t'), with either $s \neq s'$, $t \neq t'$, or both, belong to the same *contiguous region* if there exists a connected path between them such that each point along the path is also α_{pw} significant. To perform the excursion test, we identify contiguous regions in the ccc_i^* matrices, and for each contiguous region $C_i^{m_i}$ with $m_i \in [1 \dots M_i]$ where M_i is the number of contiguous regions in the i^{th} bootstrapped dataset, we record the sum of the excess correlation above the α_{pw} cutoff values,

$$k(i, m_i) = \sum_{(s,t) \in C_i^{m_i}} \text{ccc}_i^*(s,t) - \alpha_{\text{pw}}(s,t). \tag{20}$$

The collection $\{k(i, m_i) : i \in [1 \dots B], m_i \in [1 \dots M_i]\}$ defines a null distribution over the total excess correlation for contiguous regions, from which an α_{region}-level cutoff value can be calculated. We consider as significant any contiguous regions C^m whose total excess correlation exceeds the α_{region}-level cutoff value.

Simulations

We ran several simulation scenarios to evaluate the effectiveness of the DKCCA procedure[1]. Each scenario simulated an experiment with two multivariate time series of dimensions 96 and 16, with 100 repeated time-locked trials. Each time series for each trial was generated from a two-dimensional latent variable model

$$X(t) = A(t)H(t) + \epsilon(t) \tag{21}$$

where $H(t) \in \mathbb{R}^2$ is the latent trajectory with the first and second components uncorrelated, and $A(t) \in \mathbb{R}^{q \times 2}$ (where $q = 96$ for time series 1 and $q = 16$ for time series 2) is a mapping from the latent variable to the signal space, and $\epsilon(t)$ is a noise term. $H(t)$ and $\epsilon(t)$ were both resampled for each trial, while $A(t)$ was fixed across trials, though as is clear by the notation, was allowed to vary across time. We refer to the collection of $A(t)$ for all t as the *A-operator.*

We introduced correlation between time series 1 and 2 through the first latent dimension for each time series. Let H_1^1 and H_2^1 be the sample paths of the first latent dimension for time series 1 and 2, respectively. Further, let $\rho(k)$ be a "ramp function" that is piecewise linear with $\rho(0) = 0$, ramps up to a plateau where it stays for a length of time, then returns to 0. Then starting at time s and for desired lag l,

$$H_2^1(s+k) = \rho(k) H_1^1(s+k-l) + (1 - \rho(k)) H_2^1(s+k). \tag{22}$$

While we kept lag l fixed across trials, we allowed the induced correlation starting time s to vary uniformly within 10 time steps in order to simulate small differences in the time-course of brain signals with respect to the time-locked stimulus.

We sampled each latent trajectory of H for each time series, and the trajectories for each entry in the time-dynamic A-operators for each time series, as zero mean Gaussian processes with covariance kernels of the form

$$\Sigma(i,j) = \sigma^2 \exp\left(-.5 * \left(\frac{i-j}{\lambda}\right)^2\right). \tag{23}$$

Let A_1 be the A-operator for time series 1 and A_2 the A-operator time series 2. For A_1, A_2, H_1, and H_2, we let $\sigma = 1$ in equation 23. For H_1 and H_2 we let λ be 40 and 20, respectively, and for A_1 and A_2 we set $\lambda = 100$.

Finally, $\epsilon(t)$ for each time series is simulated as a two-dimensional latent state model as in equation 21 in order to control spatial-dependence and long-range temporal-dependence in the noise. For each A_ϵ-operator (where the

[1] Code for the DKCCA algorithm and this simulation is provided at https://github.com/jrodu/DKCCA.git

subscript emphasizes that it is used to simulate the noise terms $\epsilon(t)$), we let $\sigma = 1$ and $\lambda = 30$ in equation 23. For H_ϵ we let $\lambda = 80$ and, across various simulation conditions on the signal to noise ratio, fix $\sigma_{\text{noise}} = \sigma$ in equation 23 to be between 0.2 and 2.

Experimental Methods

The experiment consisted of the presentation of one of four cue objects, each mapped to one of two associated objects. A rhesus macaque was required to fixate on a white center dot, after which one of the four object cues was shown, followed a correct or incorrect associated object. A correct object presentation required an immediate saccade to a target, while an incorrect object presentation required a delay until a correct object was presented. The presentations of fixation, object cue, and associated objects were separated by blank intervals. Multi-electrode recordings were made from both Hippocampus (HPC) and lateral prefrontal cortex (PFC) in order to examine the roles of HPC and PFC in non-spatial declarative memory.

Each day over 1000 trials were conducted. Electrodes were daily re-implanted into the macaque brain, hence the exact location, and number, of electrodes in the HPC and PFC varied from day to day. For the analysis of the DKCCA algorithm we only considered days in which there were at least 8 electrodes in each section, with typical numbers being between 8 and 20 per brain region. The sampling rate for each trial was 1000Hz. For further details on the experimental setup and data collection, see Brincat and Miller (2016). All procedures followed the guidelines of the MIT Animal Care and Use Committee and the US National Institutes of Health. We expand upon details relevant to analysis of the DKCCA algorithm here.

Past work has shown that for multielectrode recordings such as LFP and EEG, the use of a common pickup and the presence of volume conduction can adversely affect functional connectivity analysis. See, for example, the work of Bastos and Schoffelen (2016) and Trongnetrpunya et al. (2016) for a description of the problem and steps that can be taken to mitigate the effect. These same issues impact DKCCA, especially when the instantaneous correlation of the contaminating signal is almost as strong as or stronger than the lagged correlation of interest, and we recommend following the guidance proposed by past work. For the current data, to minimize pickup of any specific signals through the reference, collection was made using a common reference via a low-impedance connection to animal ground.

For the analyses reported here, we filtered each trial using a Morlet wavelet centered around 16 Hz, where this frequency was determined from previous studies. It is critical in correlation based analyses to isolate the frequency band of interest, and 16 Hz was chosen for this analysis because it is in this region (the beta band) that we see the strongest LFP power and coherence effects (see Brincat and Miller (2016) for details). Finally, we downsampled the signal to 200Hz.

Results

The goal of DKCCA was to extend dynamic cross-correlation analysis to two multivariate time series with repeated trials. The algorithm infers a linear or nonlinear combinations over the signal dimensions that (a) are interpretable, (b) are allowed to change over time, and (c) do not require prior knowledge of the lagged correlation structure. Because it uses sliding windows, the combinations can adapt to the underlying nonstationarity of the brain regions and their interactions. DKCCA can thereby recover correlation structure where averaging methods fail. In this section, we evaluate DKCCA on both simulated and real data. In simulations, we show that DKCCA is able to recover the underlying correlation structure between two highly dynamic multivariate time series under varying noise conditions, and simulate a realistic example where DKCCA recovers the correlation structure but averaging methods do not. In the real data, DKCCA detected a novel cross-correlation while the averaging methods did not.

DKCCA analysis of simulated data

To illustrate DKCCA's ability to recover dynamic cross correlation from two multivariate signals, we ran several simulations (see equations 21, 22, and 23) under varying noise conditions, with lag $l = 20$ and simulation start time s sampled for each trial uniformly between min $= 310$ and max $= 320$. We designed our simulations to verify three important properties of the algorithm: (a) that it can recover cross-correlation between highly dynamic time series, (b) that it is robust against spurious correlation, and (c) that it can recover cross-correlation in adversarial conditions, where averaging methods fail. We subsequently illustrate these properties in our analysis of the real data.

Despite the highly dynamic nature of the simulated time series, DKCCA correctly identifies the temporal region in which lagged correlation exists, and indicates no correlation outside of that region. Fig. 1 shows the raw cross-correlograms for various noise levels generated from one such set of simulations. Fig. 2 shows the cross-correlograms after accounting for the inference step. The magnitude displayed is excess pairwise canonical correlation after inference. As noise increases, cross-area correlated multivariate signals are more difficult to find, but DKCCA is successful even in high noise regimes. Further, our simulations suggest that the method is robust to finding spurious correlations, since in Figs. 1 and 2 almost no cross-correlation is indicated outside of the regions of induced correlation.

To see this more directly, we ran simulations with no cross-correlation structure, again under varying noise conditions. Results are in Fig 3. For all analyses, we set our window size parameter, g, to be 20.

To asses DKCCA's recovery of the correct lag, we computed a standard cross correlogram from the ccc matrix generated by the DKCCA algorithm by averaging the lagged correlations of one region with respect to the other, within the time period of induced lagged correlation. The results in Fig. 4 show that our algorithm correctly recovers the location of the maximal correlation at a lag of 20. As is expected the accuracy of the method decreases as the signal-to-noise ratio decreases.

Finally we compared the performance of DKCCA to two common methods, averaging the pairwise correlations between signals (APC) and taking the correlation between the average of signals in each region (CAS). Figure 5 shows the results of the comparison with the same simulation setup as before, but with signal present in only a portion of the simulated electrodes. We achieved this by permuting the remaining electrodes across trials, thereby leaving the temporal characteristics of each electrode intact, but breaking their dependence with other electrodes in a given trial. While APC and CAS do not capture the simulated correlation structure, DKCCA does capture the correct correlation dynamics.

DKCCA analysis of LFPs from PFC and HPC

We applied DKCCA to analyze local field potential (LFP) data from the paired-association task described in the Experimental Methods section. In the experiment, a cue was presented, followed by a series of objects, one of which had a learned association with the cue. A rhesus macaque was required to make a saccade to a target following the display of the correct associated object, and a reward was provided as feedback following a correct saccade. In the original study, Brincat and Miller (2015) concentrated their analysis on the feedback period of each trial. We focused our analysis on the period between the initial presentation of the cue object and the appearance of the first potential associated object. In this portion of the experiment, Brincat and Miller (2015) showed there was high power in a band around 16Hz in both HPC and PFC, and that, broadly, signal in HPC led the signal in PFC (HPC → PFC). In our analysis we (a) verify that DKCCA also identified that, broadly, HPC → PFC, and (b) compare DKCCA to standard averaging methods to show that DKCCA is better able to detect nuanced cross-correlation. In addition to (a) and (b), our analysis uncovered an interesting reversal in the lead-lag relationship between HPC and PFC that was not found by traditional averaging methods.

In our analysis we used a block of 200 trials, and a total sliding window size of 21 (g=10) time steps, or 105ms. We used kernels generated by setting ϕ to be the identity map. The first column of Fig. 6 shows the result of DKCCA generated from a representative set of trials. The figure shows both the ccc matrix for the full trial length (top panel) as well as the ccc matrix zoomed in to the time period corresponding to the portion of the trial between the presentation of the cue and the presentation of the associated pair (bottom panel). In the full trial, DKCCA shows strong correlation before the presentation of the cue (time -.5s to 0s), and during the presentation of the cue (time 0s to .5s). Lag time varies throughout the interval before and during presentation of the cue, but generally suggests that HPC leads PFC. Correlation is weaker and more intermittent between the presentation of the cue and the presentation of the associated pair, with a noticeable temporary increase in correlation just before the associated pair presentation. While not as distinct as during and just before cue presentation, correlation during the associated pair presentation (timing 1.25s to 1.75s) is persistent. Although neither APC nor CAS show significant lagged activity in the time period between the presentation of the cue and the presentation of the associated pair, the zoomed cross-correlogram for DKCCA shows a burst of correlation just after time=1s, and in particular suggests that PFC leads HPC in that time period. The effect is significant in the magnitude of the excursion with $p << .001$.

The highly significant correlation between PFC and HPC found by DKCCA in the bottom left panel of Fig. 6 appears to be asymmetric about the diagonal. We found the point of maximal cross correlation to occur when PFC leads HPC by approximately 65ms, as marked with a red cross in that figure. The within-task timing of a switch in lead-lag relationship from HPC → PFC to PFC → HPC would be roughly consistent with the timing of the lead-lag switch found in Place et al. (2016), who studied the theta band in rats. However, with our descriptive method we are unable to provide a statistical test of the lead-lag relationship. We also do not contribute to the as of yet inconclusive discussion of why the theta band is predominant in hippocampal-cortical interaction in rodents, while beta is predominant in primates. But despite the difference in frequencies, in both rats and primates, interactions with HPC → PFC directionality seem to be prominent early in trials, during the preparatory and cue presentation periods, but switch to PFC → HPC directionality near the time period when the cue is eliciting memory retrieval. This suggests (in both situations) that PFC may be involved in guiding memory retrieval in the HPC. This figure confirms on real data that DKCCA can effectively recover cross-correlation, and that it has the power to detect subtle changes in the lead-lag relationship between time series, such as the reversal seen between object presentations.

We next compare the results of DKCCA to results from APC. In Fig. 6, DKCCA reports a much richer correlation structure throughout the trial than does APC. There are two places in particular where APC differs from DKCCA: first in the period before the presentation of the cue (time -.5s to 0s), and second between the presentation of the cue and the associated pair (time .5s to 1.25s), where APC indicates that correlation disappears between HPC and PFC. The reason is that with APC there is a single set of weights used when combining the pairwise cross-correlations, regardless of the temporal location of those cross correlations in the trial. However, because of the dynamic nature of the brain, signal pairs do not contribute to population-level correlation activity in a consistent manner over time. In Fig. 6, even where APC does indicate correlation, it does not capture the optimal correlation structure. DKCCA, on the other hand, accommodates these changing dynamics by modulating the weights of the signals over time.

CAS has comparable performance to APC. In particular, the CAS method is unable to fully capture the dynamic cross-correlation structure both before the presentation of the cue and between the presentation of the cue and the presentation of the associated pair. APC and CAS both treat all signals with the same weight, which does not optimize for, and therefore potentially misses, dynamic cross-correlation between the two regions.

Discussion

In this paper we derived and studied a pair of descriptive methods for assessing dynamic cross-correlation between two multivariate time series. DCCA extends canonical correlation analysis to three-way arrays indexed by signals, time, and trials, where the canonical weights over signals are allowed to evolve slowly over time. Because it is based on sliding windows, DCCA accommodates nonstationary time series while avoiding strong assumptions on the dynamics of the lagged correlation over time. We prefer to use the kernelized variant, DKCCA because it scales well with a growing number of signals, incorporates nonlinear combinations, and provides computational efficiencies not available in DCCA.

Even though nonlinear kernels might provide powerful results, they are difficult to interpret. In our analyses, we have used the linear kernel, which yields weights that are interpreted identically to those in CCA while providing more stable and efficient computation.

DKCCA requires a few parameters to be set, or inferred. The first is a regularization parameter used in implementing KCCA. In one run of the DKCCA algorithm, KCCA is called multiple times, and in order to allow for comparisons between different entries in the *ccc* matrix, it is important to use a single regularization value for all KCCA calls. In our experiments, we set this global regularization value through cross-validation. The second parameter is the half-window size g. This value should be set according to scientific context, though there are a few considerations to take into account. A large g heavily smooths the kernel matrices over time. This leads to a slow evolution of $\alpha(s)$ and $\beta(s)$ in equation 7 and could possibly obscure the non-stationary characteristic of the time series. On the other hand, small g might miss important lagged-correlation structure. In this paper, we did not explore optimizing g in the absence of intuition about the maximal lag of interest. We leave this for future work.

We also suggest a possible method for computational savings when the time series under consideration are long and prohibit DKCCA (or DCCA) from running in the desired amount of time. As presented in this paper, the algorithms set the weights for a single time point for each window of the sliding window. Instead, weights for j time points can be set per window. We advise that j should be much smaller than the length of the window, $2g + 1$, in order to ensure the smoothness of weights over time. For the analyses in this paper, we set $j = 1$.

We validated DKCCA on both simulated and real data, comparing the results to methods commonly used to compute cross-correlation. On simulated data, DKCCA successfully captured the dynamic cross-correlation structure, even under adversarial conditions where traditional methods failed. On real data, in addition to showing more intricate dynamics where traditional methods also captured some correlation, DKCCA found cross-correlation where those methods did not, suggesting a switch in the lead-lag relationship between the hippocampus (HPC) and prefrontal cortex (PFC) in the rhesus macaque.

Signals captured from the brain are highly dynamic, demanding new statistical tools to characterize them. DKCCA provides one such tool to describe dynamic cross-correlation.

Grants

This work was supported by NIMH grants MH064537 (to R.E.K.), R37MH087027 (to E.K.M), and F32MH081507 (to S.L.B) and The MIT Picower Institute Innovation Fund (to E.K.M. and S.L.B).

Disclosures

No conflicts of interest, financial or otherwise, are declared by the authors.

References

Barnett, L. and Seth, A. K. (2011). Behaviour of granger causality under filtering: theoretical invariance and practical application. *J. Neurosci. Methods*, 201(2):404–419.

Bastos, A. M. and Schoffelen, J.-M. (2016). A tutorial review of functional connectivity analysis methods and their interpretational pitfalls. *Frontiers in systems neuroscience*, 9:175.

Bießmann, F., Meinecke, F. C., Gretton, A., Rauch, A., Rainer, G., Logothetis, N. K., and Müller, K.-R. (2010). Temporal kernel cca and its application in multimodal neuronal data analysis. *Machine Learning*, 79(1-2):5–27.

Brincat, S. L. and Miller, E. K. (2015). Frequency-specific hippocampal-prefrontal interactions during associative learning. *Nature neuroscience*, 18(4):576–581.

Brincat, S. L. and Miller, E. K. (2016). Prefrontal cortex networks shift from external to internal modes during learning. *Journal of Neuroscience*, 36(37):9739–9754.

Brovelli, A., Ding, M., Ledberg, A., Chen, Y., Nakamura, R., and Bressler, S. L. (2004). Beta oscillations in a large-scale sensorimotor cortical network: directional influences revealed by granger causality. *Proc. Natl. Acad. Sci. U. S. A.*, 101(26):9849–9854.

Christopher, M. B. (2016). *Pattern Recognition and Machine Learning.* Springer-Verlag New York.

Ding, M. and Wang, C. (2014). Analyzing MEG data with granger causality: Promises and pitfalls. In Supek, S. and Aine, C. J., editors, *Magnetoencephalography*, pages 309–318. Springer Berlin Heidelberg.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438.

Hardoon, D. R., Szedmak, S. R., and Shawe-taylor, J. R. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, pages 321–377.

Lu, H. (2013). Learning canonical correlations of paired tensor sets via tensor-to-vector projection. In Rossi, F., editor, *IJCAI*. International Joint Conference on Artificial Intelligence/Association for the Advancement of Aritificial Intelligence.

Place, R., Farovik, A., Brockmann, M., and Eichenbaum, H. (2016). Bidirectional prefrontal-hippocampal interactions support context-guided memory. *Nature Neuroscience*.

Trongnetrpunya, A., Nandi, B., Kang, D., Kocsis, B., Schroeder, C. E., and Ding, M. (2016). Assessing granger causality in electrophysiological data: removing the adverse effects of common signals via bipolar derivations. *Frontiers in systems neuroscience*, 9:189.

Wang, X., Chen, Y., and Ding, M. (2008). Estimating granger causality after stimulus onset: a cautionary note. *Neuroimage*, 41(3):767–776.

Xu, Y., Sudre, G. P., Wang, W., Weber, D. J., and Kass, R. E. (2011). Characterizing global statistical significance of spatiotemporal hot spots in magnetoencephalography/electroencephalography source space via excursion algorithms. *Statistics in medicine*, 30(23):2854–2866.

Figure 1: DKCCA captures lagged correlation across a variety of noise levels. Representative results for running DKCCA on simulated data with $\sigma_{\text{noise}} = .2, .6, 1, 1.2, 1.4$ and 2 (left to right, top to bottom). y-axis is *time (ms)* in section X, x-axis is *time (ms)* in section Y. Trial length: 500ms. True lag of 20ms begins in each trial between 300 and 310 in section Y (280 and 290 in section X) and lasts about 80ms. DKCCA recovers the true correlation structure well even for relatively small signal to noise ratio.

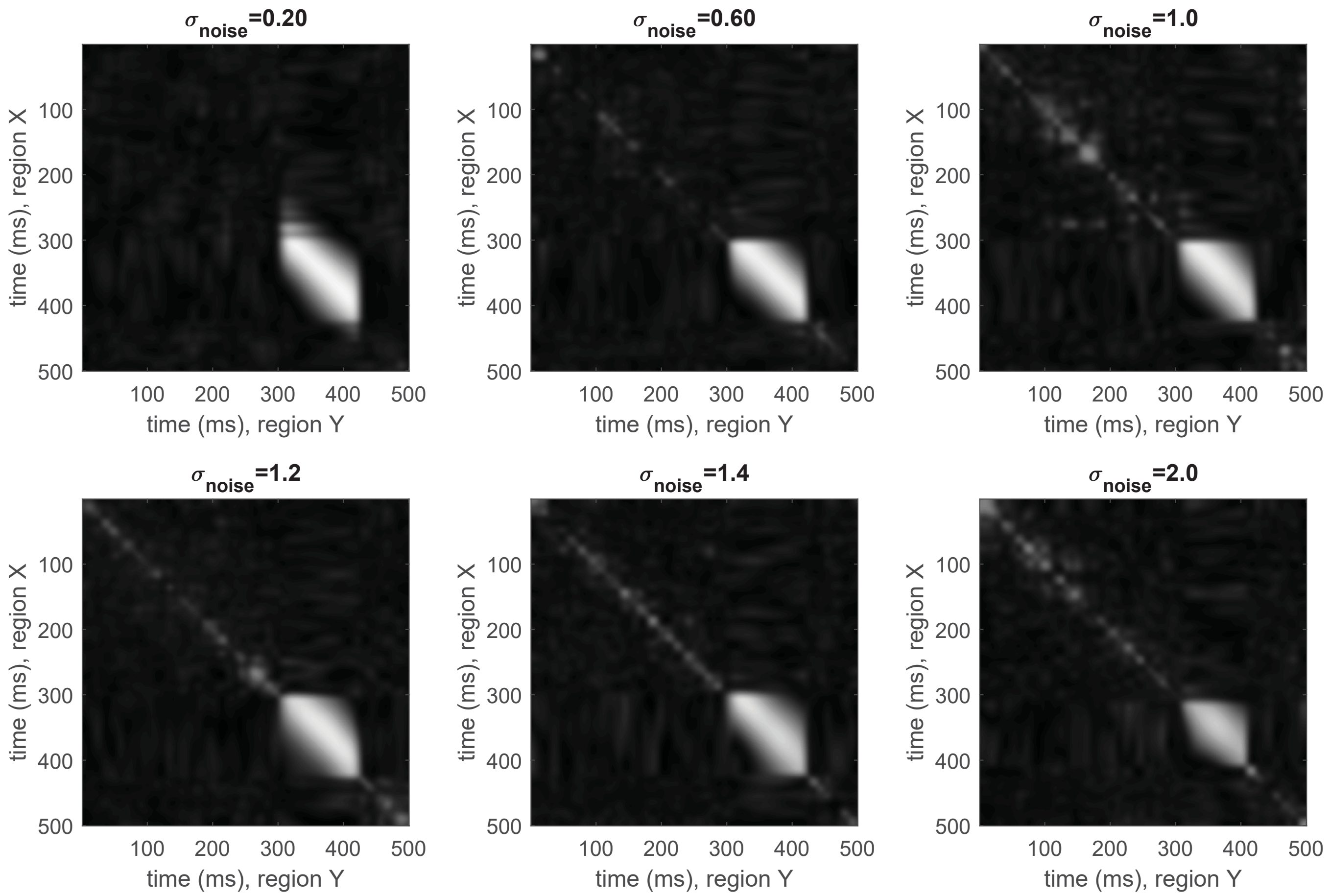
Figure 2: Excursion test for DKCCA correctly identifies significant regions of cross-correlation. Representative results after bootstrap and excursion test for trials from simulations with $\sigma_{\text{noise}} = .2, .6, 1, 1.2, 1.4$ and 2 (left to right, top to bottom). y-axis is *time (ms)* in section X, x-axis is *time (ms)* in section Y. Trial length: 500ms. True lag of 20ms begins in each trial between 300 and 310 in section Y (280 and 290 in section X) and lasts about 80ms. Nonzero values are excess above cutoff. Zero values indicate time-pair correlation that was at cutoff or below.

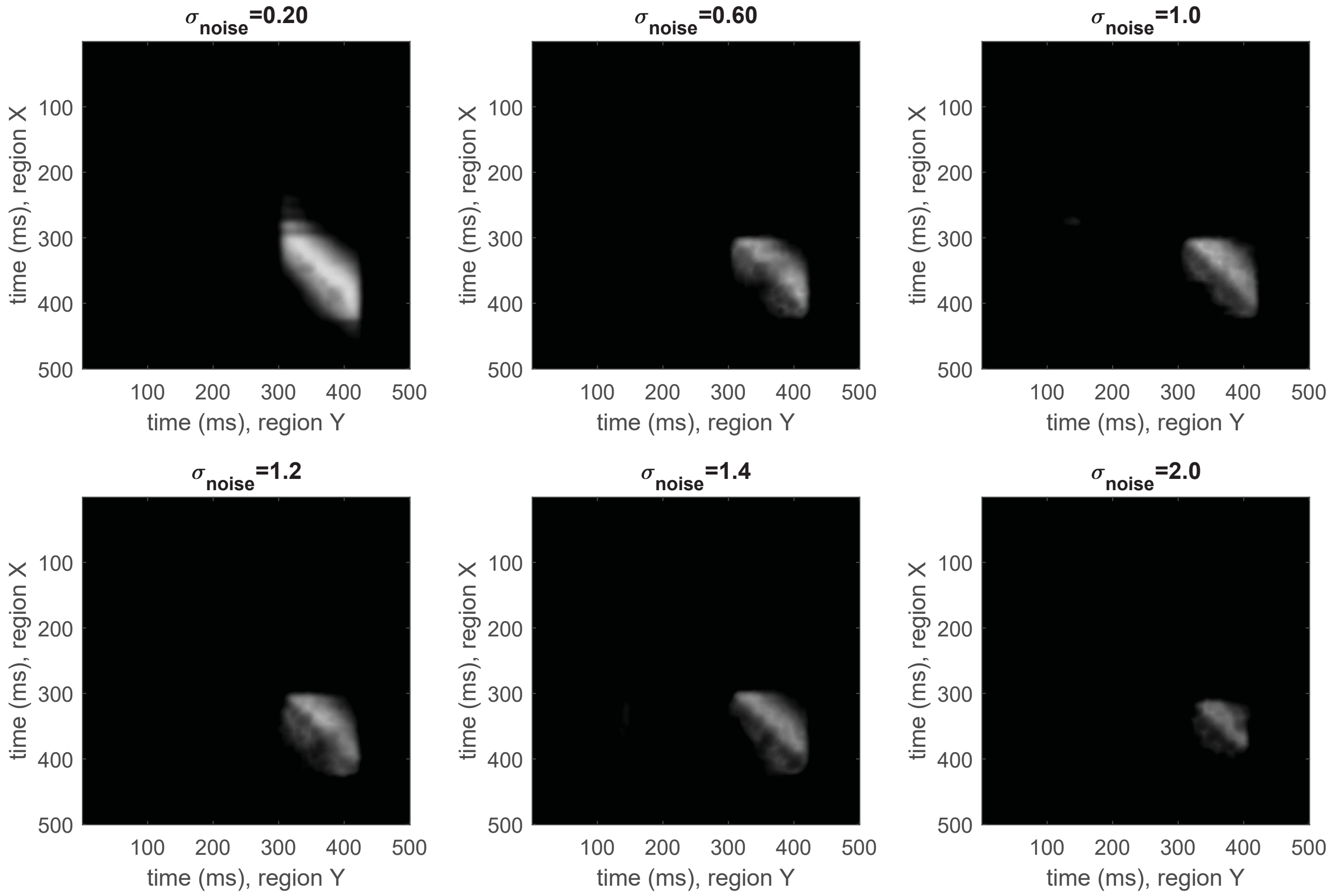
Figure 3: Excursion test for DKCCA avoids identifying spurious cross-correlation. Representative results for running DKCCA on simulated data with $\sigma_{\text{noise}} = .2, .6, 1, 1.2, 1.4$ and 2 (left to right, top to bottom). y-axis is *time (ms)* in section X, x-axis is *time (ms)* in section Y. Trial length: 500ms. No true lead-lag relationships present.

Figure 4: DKCCA accurately identifies the correct lag length. Each grey line is the averaged lagged correlation of area of interest for trials from simulations (10 simulations for each noise level) with $\sigma_{\text{noise}} = .2, .6, 1, 1.2, 1.4$ and 2. Lag $\tau = 20$ is indicated with a vertical line.

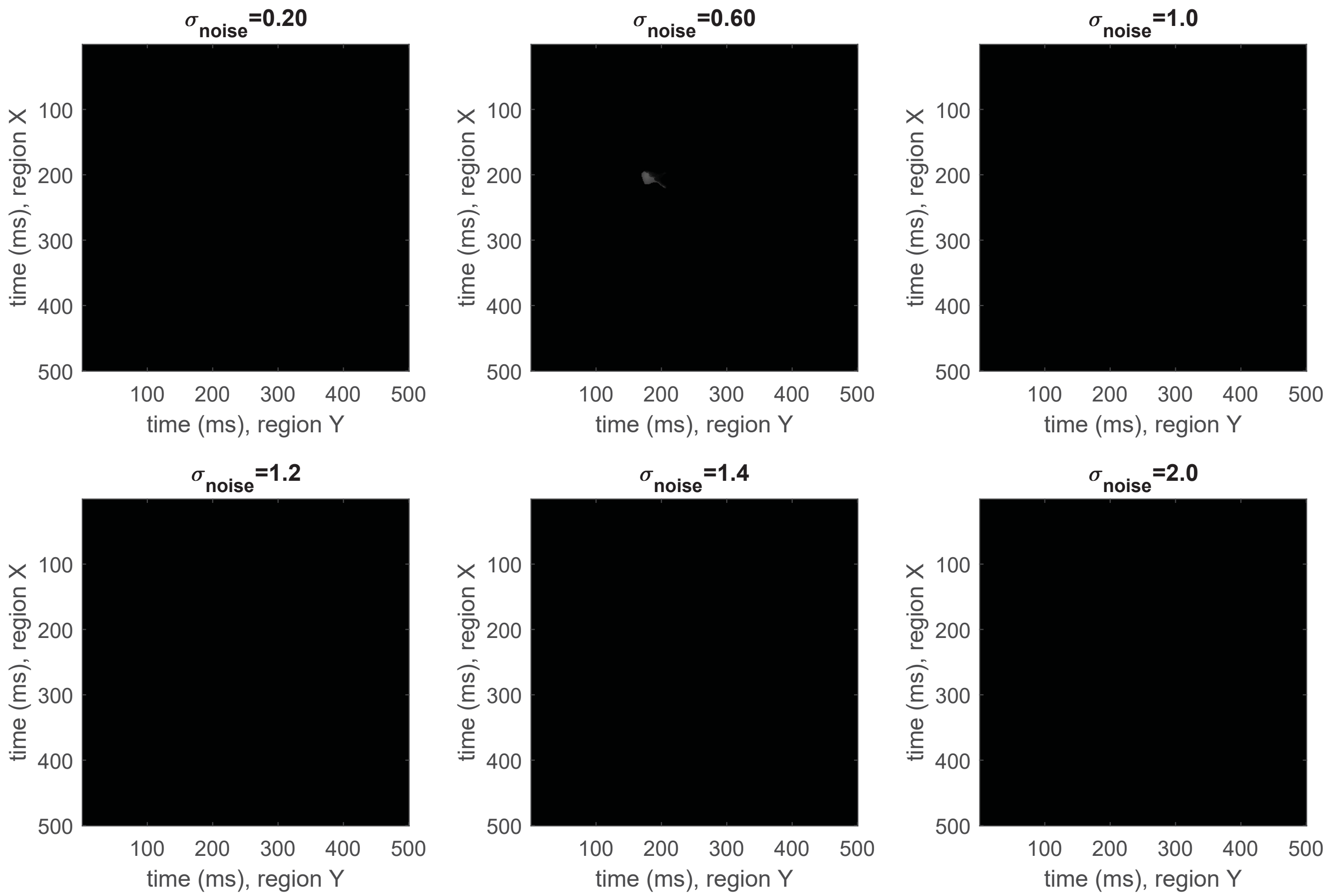
Figure 5: DKCCA identifies the correct region of interest where other methods fail. Comparison of DKCCA, APC, and CAS on simulated data with signal present in only a portion of the electrodes. DKCCA correctly identifies signal, as above. APC and CAS are unable to identify signal. Note that intensity is not comparable across methods as all correlations in APC and CAS are comparatively small, so correlations have been scaled for visual presentation.

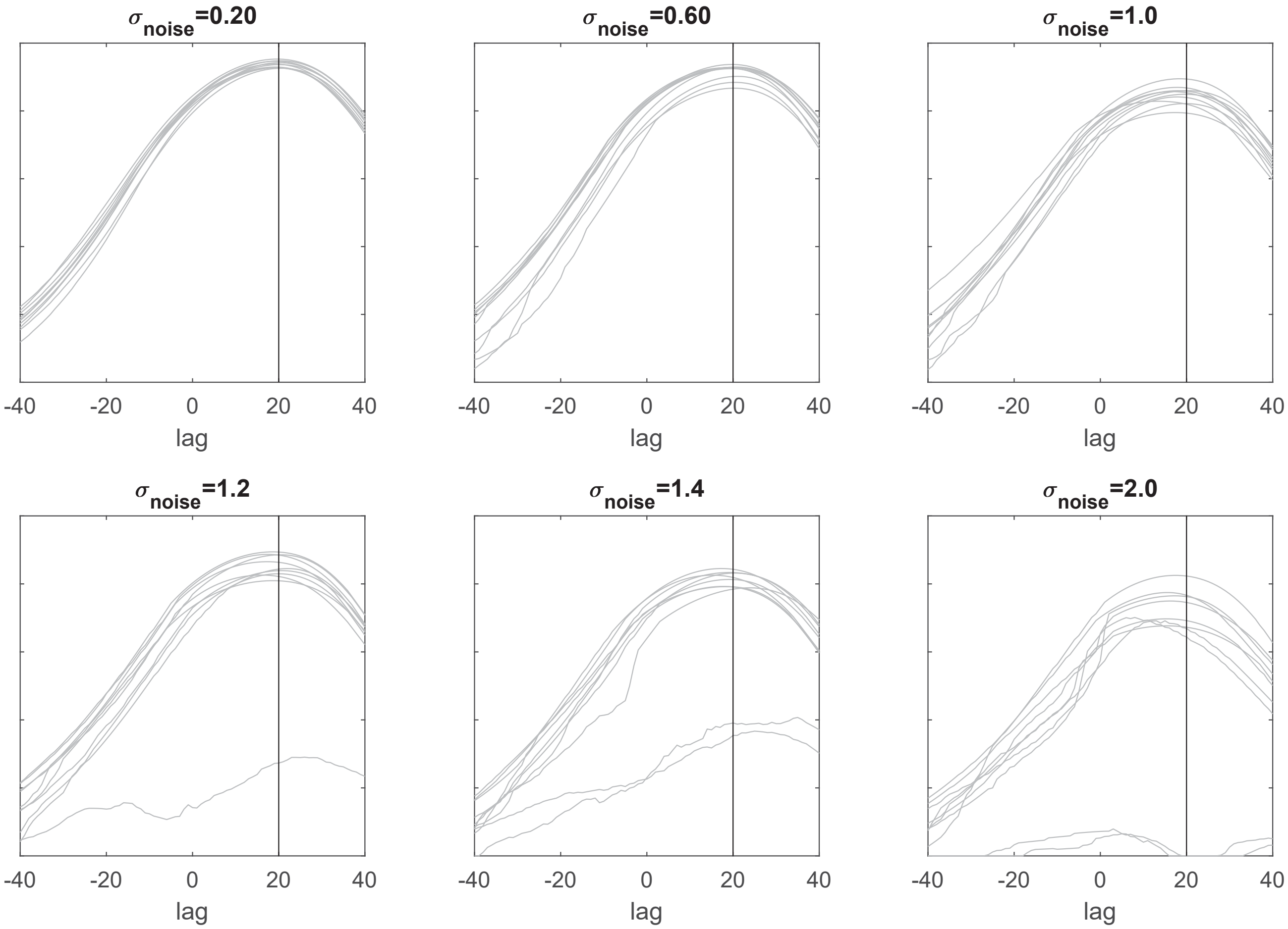
Figure 6: DKCCA finds lagged cross-correlation where traditional methods fail. Top left: cross-correlogram created by the DKCCA algorithm on full trial from -.75 seconds to about 1.75 seconds. The small red and green boxes show timing of display of cue and associated pairs, respectively. Bottom left: cross-correlogram from trial zoomed in to the period between the end of the cue presentation (time = .5 seconds) and the beginning of the associated pair presentation (time = 1.25 seconds), as indicated by the large red box on the full cross-correlogram. The red diagonal line shows time in HPC = time in PFC, and the red cross in the bottom left panel indicates the location of the maximum cross correlation in the excursion of interest. Results based on DKCCA are contrasted to the cross-correlogram and zoomed cross-correlogram created by averaging the absolute value of the pairwise cross-correlograms (APC, middle column) and those created by averaging the signals prior to calculating the cross-correlogram (CAS, right column). In both APC and CAS, we do not see any activity in the zoomed cross-correlograms. For display purposes, correlation magnitudes are not comparable between DKCCA, APC, and CAS as APC and CAS cross-correlations tend to be small compared to the DKCCA, so they have been rescaled.

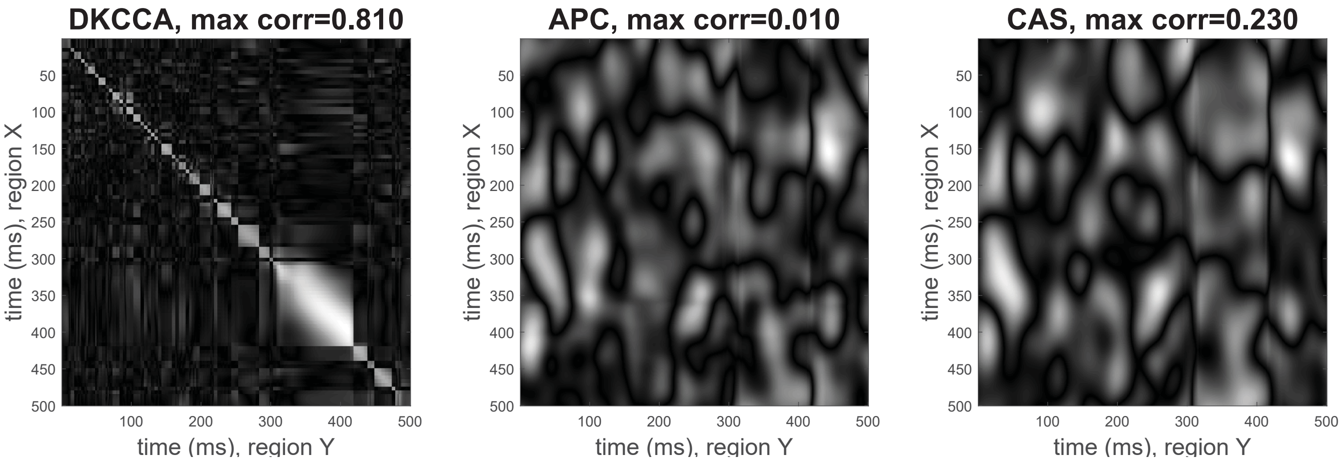

σ_{noise}=0.20
σ_{noise}=0.60
σ_{noise}=1.0
σ_{noise}=1.2
σ_{noise}=1.4
σ_{noise}=2.0
time (ms), region X
time (ms), region Y
100
200
300
400
500

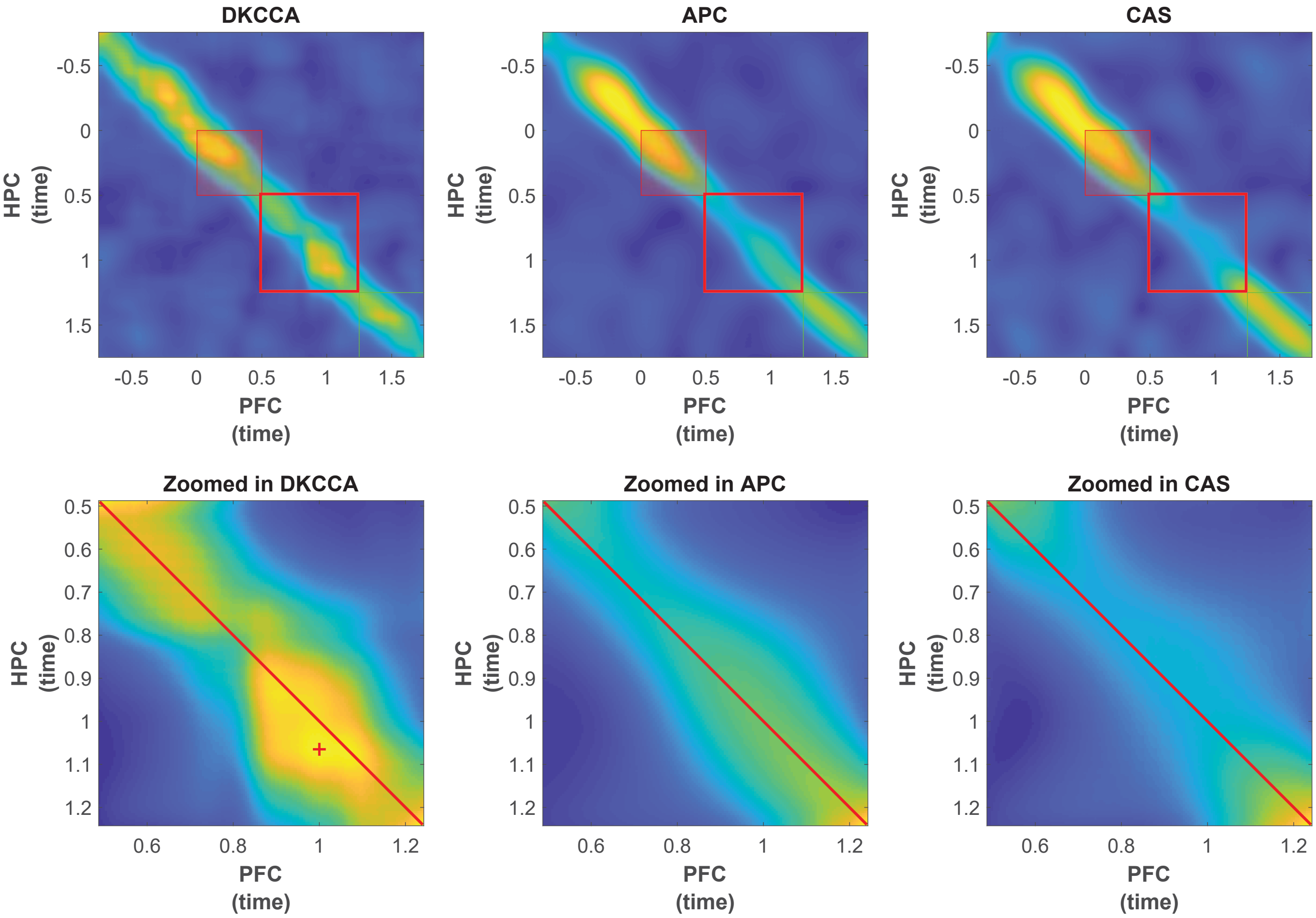


σ_{noise}=0.20
σ_{noise}=0.60
σ_{noise}=1.0
σ_{noise}=1.2
σ_{noise}=1.4
σ_{noise}=2.0
time (ms), region X
time (ms), region Y
100
200
300
400
500


σ_{noise}=0.20
σ_{noise}=0.60
σ_{noise}=1.0
σ_{noise}=1.2
σ_{noise}=1.4
σ_{noise}=2.0
time (ms), region X
time (ms), region Y
100
200
300
400
500


σ_{noise}=0.20
σ_{noise}=0.60
σ_{noise}=1.0
σ_{noise}=1.2
σ_{noise}=1.4
σ_{noise}=2.0
-40
-20
0
20
40
lag


DKCCA, max corr=0.810
APC, max corr=0.010
CAS, max corr=0.230
time (ms), region X
time (ms), region Y
50
100
150
200
250
300
350
400
450
500


DKCCA
APC
CAS
HPC
(time)
PFC
(time)
-0.5
0
0.5
1
1.5
Zoomed in DKCCA
Zoomed in APC
Zoomed in CAS
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
0.6
0.8
1
1.2