

Using connectome-based predictive modeling to predict individual behavior from brain connectivity

Xilin Shen¹, Emily S Finn², Dustin Scheinost¹, Monica D Rosenberg³, Marvin M Chun^{2–4}, Xenophon Papademetris^{1,5} & R Todd Constable^{1,2,6}

¹Department of Radiology and Biomedical Imaging, Yale School of Medicine, New Haven, Connecticut, USA. ²Interdepartmental Neuroscience Program, Yale School of Medicine, New Haven, Connecticut, USA. ³Department of Psychology, Yale University, New Haven, Connecticut, USA. ⁴Department of Neuroscience, Yale School of Medicine, New Haven, Connecticut, USA. ⁵Department of Biomedical Engineering, Yale University, New Haven, Connecticut, USA. ⁶Department of Neurosurgery, Yale School of Medicine, New Haven, Connecticut, USA. Correspondence should be addressed to R.T.C. (todd.constable@yale.edu).

Published online 9 February 2017; doi:10.1038/nprot.2016.178

Neuroimaging is a fast-developing research area in which anatomical and functional images of human brains are collected using techniques such as functional magnetic resonance imaging (fMRI), diffusion tensor imaging (DTI), and electroencephalography (EEG). Technical advances and large-scale data sets have allowed for the development of models capable of predicting individual differences in traits and behavior using brain connectivity measures derived from neuroimaging data. Here, we present connectome-based predictive modeling (CPM), a data-driven protocol for developing predictive models of brain–behavior relationships from connectivity data using cross-validation. This protocol includes the following steps: (i) feature selection, (ii) feature summarization, (iii) model building, and (iv) assessment of prediction significance. We also include suggestions for visualizing the most predictive features (i.e., brain connections). The final result should be a generalizable model that takes brain connectivity data as input and generates predictions of behavioral measures in novel subjects, accounting for a considerable amount of the variance in these measures. It has been demonstrated that the CPM protocol performs as well as or better than many of the existing approaches in brain–behavior prediction. As CPM focuses on linear modeling and a purely data-driven approach, neuroscientists with limited or no experience in machine learning or optimization will find it easy to implement these protocols. Depending on the volume of data to be processed, the protocol can take 10–100 min for model building, 1–48 h for permutation testing, and 10–20 min for visualization of results.

INTRODUCTION

Establishing the relationship between individual differences in brain structure/function and individual differences in behavior is a major goal of modern neuroscience. Historically, many neuroimaging studies of individual differences have focused on establishing correlational relationships between brain measurements and cognitive traits such as intelligence, memory, and attention, or disease symptoms.

Note, however, that the term ‘predicts’ is often used loosely as a synonym for ‘correlates with’—for example, it is common to say that brain property x ‘predicts’ behavioral variable y , where x may be an fMRI-derived measure of univariate activity or functional connectivity and y may be a measure of task performance, symptom severity, or another continuous variable. Yet, in the strict sense of the word, this is not prediction but rather correlation. Correlation or similar regression models tend to overfit the data and, as a result, often fail to generalize to novel data. The vast majority of brain–behavior studies do not perform cross-validation, which makes it difficult to evaluate the generalizability of the results. In the worst case, Kriegeskorte *et al.*¹ demonstrated that circularity in selection and selective analyses leads to completely erroneous results. Proper cross-validation is key to ensuring independence between feature selection and prediction/classification, thus eliminating spurious effects and incorrect population-level inferences². There are at least two important reasons to test the predictive power of brain–behavior correlations discovered in the course of basic neuroimaging research: (i) From the standpoint of scientific rigor, cross-validation is a more conservative way to infer the presence of a brain–behavior relationship than is correlation. Cross-validation is designed to protect against overfitting by testing the strength of the relationship in a novel sample, increasing

the likelihood of replication in future studies. (ii) From a practical standpoint, establishing predictive power is necessary to translate neuroimaging findings into tools with practical utility³. In part, fMRI has struggled as a diagnostic tool because of low generalizability of results to novel subjects. Testing and reporting performance in independent samples will facilitate evaluation of the generalizability of a result and eventual development of useful neuroimaging-based biomarkers with real-world applicability.

Nevertheless, the design and construction of predictive models remains a challenge.

Recently, we have developed CPM with built-in cross-validation, a method for extracting and summarizing the most relevant features from brain connectivity data in order to construct predictive models⁴. Using both resting-state fMRI and task-based fMRI, we have shown that cognitive traits, such as fluid intelligence and sustained attention, can be successfully predicted in novel subjects using this method^{4,5}. Although CPM was developed with fMRI-derived functional connectivity as the input, we propose that it could be adapted to work with structural connectivity data measured with DTI or related methods, or functional connectivity data derived from other modalities such as EEG.

Here, we present a protocol for developing predictive models of brain–behavior relationships from connectivity data using CPM, which includes the following steps: (i) feature selection, (ii) feature summarization, (iii) model building and application, and (iv) assessment of prediction significance. We also include suggestions for visualization of results. This protocol is designed to serve as a framework illustrating how to construct and test predictive models, and to encourage investigators to perform these types of analyses.

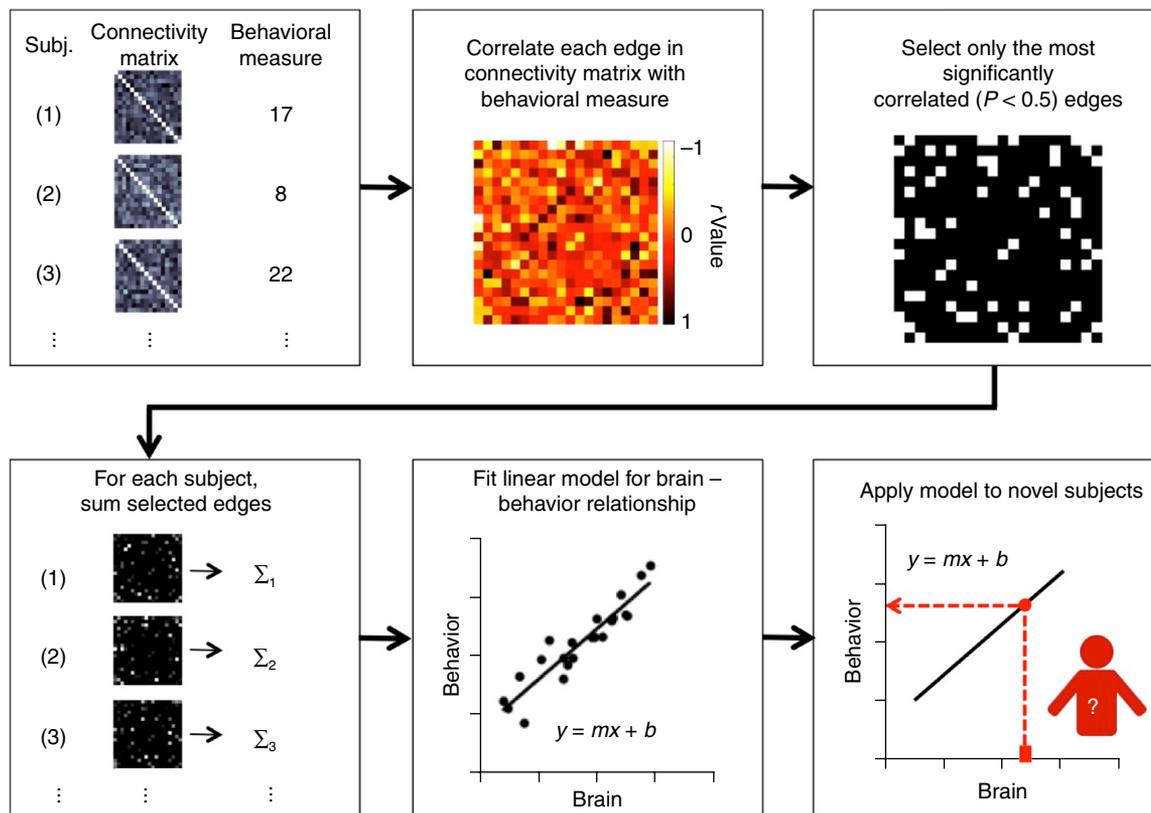


Figure 1 | Schematic of CPM. For each subject, inputs to CPM are a connectivity matrix and behavioral measures. Connectivity matrices can be from several different modalities, and behavioral measures should have a sufficient dynamic range or spread across subjects to support prediction in novel data. The input data need to be divided into a training set and a testing set (Steps 1 and 2). Next, across all subjects in the training set, each edge in the connectivity matrices is related to the behavioral measures using a form of linear regression, including Pearson’s correlation, Spearman’s correlation, or robust regression (Step 3). Next, after linear regression, the most important edges are selected for further analysis. Typically, important edges are selected using significance testing, although other strategies exist—e.g., selecting edges whose correlation value is above a predefined threshold (Step 4). Next, for each subject, the most important edges are then summarized into a single subject value. Usually, the edge strengths are simply summed (Step 5). Next, a predictive model is built that assumes a linear relationship between the single-subject summary value of connectivity data (independent variable) and the behavioral variable (the dependent variable) (Step 6). Next, summary values are calculated for each subject in the testing set. This value is then input into the predictive model. The resulting value is the predicted behavioral measure for the current test subject (Step 7).

Development of the protocol

In this protocol, we describe an algorithm for building predictive models based on a set of single-subject connectivity matrices, and test these models using cross-validation of novel data (shown as a schematic in Fig. 1). We also discuss a number of options in model building, including selection of features from predefined networks rather than from the whole brain. We address the issue of how to assess the significance of the predictive power using permutation tests. Finally, we provide examples of how to visualize the features—in this case, brain connections—that contribute the most predictive power. This protocol has been designed for users familiar with connectivity analysis and neuroimaging data processing. Data preprocessing and related issues are out of the scope of this protocol, as the methods presented in Finn *et al.*⁴ and Rosenberg *et al.*⁵ generalize to any set of connectivity matrices. Therefore, we assume that individual data have been fully preprocessed and the input to this protocol is a set of M by M connectivity matrices, where M represents the number of distinct brain regions, or nodes, under consideration, and each element of the matrix is a continuous value representing the strength of the connection between two nodes.

Applications of the method

Human neuroimaging studies routinely collect behavioral variables along with structural and functional imaging. In addition, open-source data sets including the Human Connectome Project (HCP)⁶, the NKI-Rockland sample⁷, the ADHD-200 (ref. 8), and the Philadelphia Neurodevelopmental Cohort⁹ include large samples of subjects ($N > 500$) with both imaging data and many behavioral variables. Therefore, vast amounts of data exist to explore which brain connections predict individual differences in behavior. Further, as demonstrated in Rosenberg *et al.*⁵, these open-source data sets can be pooled or combined with local data sets to test whether a predictive model generalizes across different scanners, different subject populations, and even different measures of the underlying phenotype of interest.

We have applied the CPM protocol in our research and demonstrated robust relationships between brain connectivity and fluid intelligence in Finn *et al.*⁴ and between brain connectivity and sustained attention in Rosenberg *et al.*⁵. Here, we aim to provide a user-friendly protocol for performing prediction of a behavioral variable in novel subjects using connectivity data. The models described in this protocol offer a rigorous way to establish a brain–behavior relationship using cross-validation.

Comparison with other methods

The strengths of CPM include its use of linear operations and its purely data-driven approach. Linear operations allow for fast computation (for example, roughly 60 s to run leave-one-subject-out cross-validation on 100 subjects), easy software implementation (<100 lines of MATLAB code), and straightforward interpretation of feature weights. Although state-of-the-art brain parcellation methods typically divide the brain into ~300 regions, resulting in ~45,000 unique connections or edges^{10–13}, many hypothesis-driven approaches focus on a single edge, region, or network of interest. These approaches ignore a large number of connections and may limit predictive power. By contrast, CPM searches for the most relevant features (edges) across the whole brain and summarizes these selected features for prediction.

The simplest and most popular method for establishing brain–behavior relationships using neuroimaging data is a correlation or regression model¹⁴. As mentioned in the introduction, these methods often overfit the data and limit generalizability to novel data. Often these correlational relationships are tested on *a priori* regions of interest, but may also be tested in a whole-brain, data-driven manner. Importantly, using a cross-validated approach helps guard against the potential for false positives inherent in a whole-brain, data-driven analysis, and obviates traditional correction for multiple comparisons.

The method that is most directly comparable to CPM may be the multivariate-prediction and univariate-regression method used by the HCP Netmats MegaTrawl release¹⁵ (<https://db.humanconnectome.org/megatrawl/index.html>). This set of algorithms uses independent component analysis and partial correlation to generate connectivity matrices from resting-state fMRI data. These matrices are then related to behavior using elastic-net feature selection and prediction and tenfold cross-validation (inner loop for parameter optimization and outer loop for prediction evaluation). The main differences between this approach and the proposed CPM approach are as follows: (i) use of group-wise independent component analysis (ICA) to derive subject-specific functional brain subunits (and associated time courses) versus use of an existing functional brain atlas registered to each subject; (ii) use of partial correlation versus Pearson’s correlation to measure connectivity; (iii) use of the elastic net algorithm versus Pearson’s correlation with the behavioral measure to select meaningful edges; and (iv) use of the elastic net algorithm versus use of a linear model based on mean connectivity strength for prediction. This approach is computationally more complex and requires substantial expertise in optimization. We focus on the use of a purely linear model that can be easily implemented with basic programming skills. Although no direct comparison has been made, the two methods perform similarly for predicting fluid intelligence^{4,16}.

Another alternative method for developing predictive models from brain connectivity data is support vector regression (SVR)¹⁷, an extension of the support vector machine classification framework to continuous data. In this approach, rather than performing mass univariate calculations to select relevant features (edges) and combining these into a single statistic for each subject, a supervised learning algorithm considers all features simultaneously and generates a model that assigns different weights to different features in order to best approximate each observation (distinct behavioral measurement) in the training set. Features from

the test subject(s) are then combined using the same weights and the trained model outputs a predicted behavioral score. See Dosenbach *et al.*¹⁸ for an example of SVR applied to functional connectivity data to predict subject age. A comparison between CPM and SVR in terms of performance and running time is provided in the **Supplementary Note**.

Finally, many studies have used similar machine-learning techniques with connectivity data in a classification framework to distinguish healthy control participants from patients. Reliable classification of patients has been shown in several disorders, including attention deficit hyperactivity disorder (ADHD)¹⁹, autism^{20,21}, schizophrenia²², Alzheimer’s disease²³, and depression²⁴. A fundamental difference between most classification methods and CPM (or the multivariate-prediction and univariate-regression method or the SVR-based methods described above) is that in classification the outcome variable is discrete (often binary) instead of continuous. Prediction of individual differences in a continuous measure across a healthy sample is considerably more challenging than binary classification of disease state. Variations in behavior among healthy participants generally have substantially lower effect sizes than differences due to pathology. In addition, accurate prediction of continuous variables requires accurate modeling over the whole range of the variable, whereas accurate binary classification largely requires accurate grouping of participants near the margin. In the case in which subsets of participants are distributed far from the margin, the correct classification of these subsets is often guaranteed.

Although SVR and related multivariate methods can provide good predictive power, in our experience, predictions generated using CPM are often as good or better than those generated using SVR, and CPM has at least two advantages over multivariate methods. First, from a practical standpoint, CPM is simpler to implement and requires less expertise in machine learning. This makes it more accessible to the general neuroimaging community. It is our hope that in providing this protocol we can encourage researchers to perform cross-validated analyses of the brain–behavior relationships that they discover, which will set more rigorous statistical standards for the field and improve replicability across studies.

The second major advantage of the CPM approach, as compared with multivariate methods, is that the predictive networks obtained by CPM can be clearly interpreted. It is a frequently overlooked problem in the literature that interpreting weights generated by multivariate regression models—even linear ones—is not straightforward²⁵. For example, researchers often erroneously equate large weights with greater importance, and it is even harder to interpret nonlinear models. CPM allows researchers to rigorously test the predictive value of a brain–behavior relationship while still providing a one-to-one mapping back to the original feature space so that researchers can visualize and investigate the underlying brain connections contributing to the model. This is critical for comparing results with existing literature, generating new hypotheses about network structure and function, and advancing our understanding of functional brain organization in general.

Limitations

CPM is based on linear relationships typically with a slope and an intercept (i.e., $y = mx + b$). These models may not be optimal

for capturing complex, nonlinear relationships between connectivity and behavior. Higher-order polynomial terms could be added to the model (i.e., $y = \sum_{i=1}^K \alpha_i x^i + \beta$) to capture additional variance. In addition, the predictive models tend to produce predicted values with a range that is smaller than the range of true values. That is, models overestimate the behavior of the individuals with the lowest measurements and underestimate the behavior of the individuals with the highest.

It is possible that other methods—particularly multivariate approaches such as the one used in the HCP Netmats MegaTrawl release or the SVR-based methods described in the previous section—may outperform CPM in terms of prediction accuracy for certain data sets. In deciding whether CPM is suitable for their purposes, researchers should carefully consider their priorities. If the goal is to maximize the accuracy of the prediction at all costs, researchers may consider using one of the multivariate methods described above, comparing the results with CPM, and selecting the method that gives the best prediction. Moreover, recent literature on machine learning has suggested that combined results from different prediction models (classification models) usually outperform results obtained from a single ‘best’ approach.

Overview of the procedure

In the following section, we describe step by step how to implement the CPM protocol with example MATLAB code. We discuss a number of confounding issues that could affect the model, and suggest ways to avoid or eliminate these issues. Alternative options to each step in the implementation are provided with detail. Several examples are given to illustrate how to use the online visualization tool to plot the most predictive connections in the context of brain anatomy. Codes and documents that are needed for implementing the CPM protocol are all available online, and the visualization tool is freely accessible.

Experimental design

The goal of the analysis is to establish a relationship between brain connectivity data and behavioral measure(s) of interest. Although CPM was originally developed using resting-state fMRI⁴, the connectivity data can be from a wide range of modalities, including fMRI, DTI, EEG, and magnetoencephalography. These methods have the most utility in studies with moderate to large sample sizes ($N > 100$). In most cases, the neuroimaging data and behavioral measures should be collected in a short temporal window in order to minimize differences in ‘state’ behaviors between the time of scanning and of behavioral testing. However, CPM may also be applied when ‘trait’ behavioral measures are collected a considerable time after scanning, such as using connectivity data to predict long-term symptom changes after an intervention.

Although this protocol allows researchers to rigorously test the presence of a brain–behavior relationship using within-data set cross-validation (e.g., leave-one-out or K -fold), a particularly powerful experimental design is to use two (or more) data sets, with one data set serving as the discovery cohort on which the model is built, and the second serving as the test cohort. For an example of this experimental design, see Rosenberg *et al.*⁵ In the first data set⁵, a group of healthy adults were scanned at Yale while they were performing an attention-taxing continuous performance task, and the CPM method was used to create a model to predict accuracy on the task. This same model successfully predicted

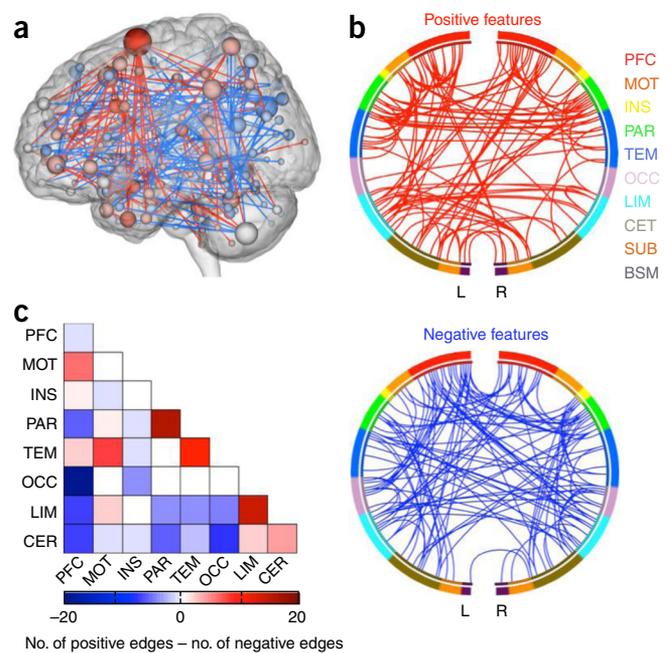


Figure 2 | Visualizing selected connectivity features. These illustrations were created using BioImage Suite (<https://www.nitrc.org/projects/bioimagesuite/>). (a) Glass brain plots: each node is represented as a sphere, where the size of the sphere indicates the number of edges emanating from that node. The set of positive features (edges) is coded red, and the set of negative features (edges) is coded blue. (b) Circle plots: nodes are arranged in two half circles approximately reflecting brain anatomy from anterior (top of the circle, 12 o'clock position) to posterior (bottom of the circle, 6 o'clock position), and the nodes are color coded according to the cortical lobes. Positive and negative features (edges) are drawn between the nodes on separate plots. The lobes are prefrontal (PFC), motor (MOT), insula (INS), parietal (PAR), temporal (TEM), occipital (OCC), limbic (LIM), cerebellum (CER), subcortical (SUB), and brain stem (BSM). (c) Matrix plots: rows and columns represent predefined networks (usually including multiple nodes). The cells represent the difference between the total number of positive edges and the total number of negative edges connecting the nodes in the two networks.

ADHD symptom scores in a group of children scanned at rest in Beijing⁸ The combination of these two independent data sets demonstrates the impressive generalizability of models built with CPM.

Evaluating the predictive power of CPM in the setting of within-data set cross-validation is an important step in this protocol. The correlation between predicted values and true values is calculated as a test statistic. However, it is unclear as to what null distribution is associated with this test statistic, and thus a permutation test should be used to generate an empirical null distribution to assess the significance. For the case in which two independent data sets are used, linear regression can provide metrics for evaluating the performance of the predictive model, the degrees of freedom can be directly calculated, and standard conversions of correlations or t values to P values can be used.

The subject’s head movement has been known to introduce confounding effects into both functional²⁶ and structural²⁷ data during data acquisition. In particular, large amounts of head motion create robust, but spurious, patterns of connectivity. These motion patterns could artifactually increase prediction performance if motion and the behavioral data are correlated.

In addition to performing state-of-the-art preprocessing methods to minimize these confounds (for discussion of various ways to estimate and correct for head motion, see reviews^{28,29}), researchers need to ensure that motion is not correlated with the behavioral data and that connectome-based models cannot predict motion⁵. If there is an association between motion and the behavioral or connectivity data, additional controls for motion, such as removing high-motion subjects from analysis, need to be performed. Other potential variables that could be correlated with the behavior of interest include age, gender, and IQ. The effects of these confounding factors could be removed using partial correlation (see feature selection and model building, Step 3 in the PROCEDURE).

Commonly, investigators will want to know not only whether a behavior can be predicted from brain connectivity but also which specific edges contribute to the predictive model. Projecting model features back into brain space facilitates interpretation based on known relationships between brain structure and function, and comparison with existing literature. However, because of the dimensionality of connectivity data, and the potential for large numbers of edges to be selected as features, visualization can be challenging.

First, one must choose a set of features (edges) to visualize and interpret. Because of the nature of cross-validation, it is likely that a slightly different set of edges will be selected as features in each iteration of the cross-validation. However, good features (edges) should overlap across different iterations. The most conservative approach is to visualize only edges that were selected in all iterations of the analysis (i.e., the overlap of all the models). Alternatively, a looser threshold may be set such that edges are included if they appear in at least 90% of the iterations, for example. We offer three suggestions for feature visualization: glass brain plots, circle plots, and matrix plots (Fig. 2).

The 3D glass brain plot (Fig. 2a) is used to visualize selected edges. In a glass brain plot, the center of mass of each node is represented by a sphere; if an edge is present, a line is drawn to connect the two spheres. (If desired, lines can be color-coded to indicate whether the edge strength is positively or negatively

correlated with the behavior of interest.) The size of the sphere can be scaled to reflect the total number of edges connecting to that node (degree), and may also be color-coded to indicate the relative number of edges that are positively versus negatively correlated with behavior. The glass brain plot is most useful to visualize a small number of edges or to demonstrate the degree distribution across the whole brain.

The circle plot (Fig. 2b) is also used to visualize edges; however, all edges are now drawn in a 2D plane. Nodes are grouped into macroscale regions, such as cortical lobes, and these regions are arranged in two half circles approximately reflecting brain anatomy from anterior (top of the circle) to posterior (bottom of the circle). The circle plots are more useful to show the general trend of connections—e.g., if there is a big bundle of edges connecting the prefrontal regions to the temporal regions, or if there are a great number of connections between right and left homologs.

The matrix plot (Fig. 2c) is useful to visualize summary statistics within and between predefined regions or networks. For example, each cell could represent the total number of edges between two regions. If two tails are present (positively versus negatively correlated features), cells may represent the difference—i.e., the total number of positive features minus the total number of negative features linking those two regions or networks (as in Fig. 2c).

Level of expertise needed to implement the protocol. Users should be familiar with the MATLAB programming environment and be able to modify software written in MATLAB. We assume that the input to this protocol is a set of M by M connectivity matrices and a set of behavioral variables. Generally, this assumption requires the user of this protocol to understand connectivity analysis and to have experience with processing of neuroimaging data. However, there are online resources, such as the HCP and the Addiction Connectome Preprocessed Initiative (http://fcon_1000.projects.nitrc.org/indi/ACPI/html/index.html), that provide independent researchers with both preprocessed data and individual connectivity matrices. These resources may allow users without expertise in connectivity analysis to use this guide.

MATERIALS

EQUIPMENT

- A computer (PC, Mac or Linux) with MATLAB 6.5 or above and Google Chrome or another modern web browser (e.g., Firefox) installed
- Individual connectivity matrices ▲ **CRITICAL** These are the first inputs to the protocol and can come from several modalities and processing methods. We assume that these matrices are symmetric. ▲ **CRITICAL** If the study collects both imaging and behavioral data, it must be approved by the appropriate ethical review board, all subjects must give informed consent (if using local data sets), and institutional data-sharing agreements must be approved (if using certain open-source data sets). With regard to the data used as example in this protocol, all participants provided written informed consent in accordance with a protocol approved by the Human Research Protection Program of Yale University. The HCP scanning protocol was approved by the local institutional review board at Washington University in St. Louis.
- Behavioral measures ▲ **CRITICAL** These are the second inputs to the protocol and are discrete or continuous variables with a sufficient dynamic range to predict individual differences. Binary variables are not appropriate for this protocol.
- Head-motion estimates ▲ **CRITICAL** Eliminating any significant ($P < 0.05$) correlation between subjects' behavioral measures and their motions is required before modeling.

- (Optional) Anatomical labels associated with each element in the connectivity matrices (The labels are used for visualization and can aid in the interpretation of results.)
- (Optional) Network labels associated with each element in the connectivity matrices (These labels further organize the elements in the matrices into networks, or collections of elements with similar functions. These labels can be used in the feature selection step to build models using a subset of connections.)
- Example code A (This is an example MATLAB script called 'behavioralprediction.m' demonstrating how to select the features, generate the summary statistics, and build the predictive model with cross-validation using CPM. This script is available from <https://www.nitrc.org/projects/bioimagesuite/>, and it can be modified by the user as needed.)
- Example code B (This is an example MATLAB script called 'permutation_test_example.m' demonstrating how to perform permutation testing to assess the significance of the prediction results. This script is available from <https://www.nitrc.org/projects/bioimagesuite/>, and it can be modified by the user as needed.)
- Online visualization tool (The tool is used for making the circle plots and glass brain plot shown in Figure 2. The URL is <http://bisweb.yale.edu/conviewer/>. The brain parcellation is preloaded with both lobe definition and network definition. Network definition is based on the Power atlas¹²,

http://www.nil.wustl.edu/labs/petersen/Resources_files/Consensus264.xls

▲ **CRITICAL** The use of this tool is optional; however, this tool was created specifically for brain connectivity analysis. It combines viewers for the brain anatomy and viewers for the connections, and thus it allows users to relate

function to structure immediately as they explore the data. This tool has many features that are desirable for understanding the nodal property of the connectivity matrices generated by the CPM protocol, which would not be accessible via other software options.

PROCEDURE

Feature selection, summarization, and model building using CPM ● **TIMING 10–100 min**

1| *Load connectivity matrices and behavioral data into MATLAB (Fig. 3a, item i).* The example MATLAB code (example code A) assumes that connectivity matrices of all subjects are stored in a 3D matrix of size $M \times M \times N$, where M is the number of nodes/regions used in the connectivity analysis and N is the number of subjects, and the behavioral variable is stored in an $N \times 1$ array (Fig. 3a, item i).

▲ **CRITICAL STEP** Subjects with high motion (mean framewise head displacement >0.15 mm) are usually not good candidates to be included in predictive modeling, because nonlinear effects of movement could persist even after extensive motion correction. Moreover, it is critical to make sure that the behavioral measure of interest is not significantly ($P < 0.05$) correlated with motion. If such an association exists, removing additional high-motion subjects is probably necessary.

? TROUBLESHOOTING

2| *Divide data into training and testing sets for cross-validation (Fig. 3a, item ii).* For cross-validation, reserve data from some subjects as novel observations and use these novel subjects to test the prediction performance. Strategies include leave-one-subject-out and K -fold cross-validation. For example, in twofold cross-validation, use half of the data to build the model (training) and the other half to evaluate the predictive power (testing), and then exchange the roles of training and testing to complete the validation.

Example code A uses leave-one-subject-out cross-validation. As shown in **Figure 3a**, item ii, one subject is removed from the training data and $N-1$ subjects are used to build the predictive model. Repeat this step in an iterative manner with a different subject left out in each iteration. To perform a K -fold cross-validation, the total number of subjects are binned into K equal size bins, and for each iteration subjects from $K-1$ bins are used for training and subjects from the one left-out bin are used for testing.

! **CAUTION** Although leave-one-subject-out cross-validation is the most popular choice, this method gives estimates of the prediction error that are more variable than those from K -fold cross-validation³⁰. However, the choice of the number of folds (K) in K -fold cross-validation can be critical and depends on many factors, including sample size and effect size (see Kohavi³⁰ for a discussion). Even when using cross-validation, incorporating a second independent data set for establishing generalizability is desirable^{31,32} (see Rosenberg *et al.*⁵ for an example).

▲ **CRITICAL STEP** It is important that no information is shared between the training set and the testing data. One common mistake occurs when some normalization is applied to all data before the cross-validation loop. Such overall normalization could contaminate the testing set with information from the training set, erroneously increasing prediction performance. Normalization should always be performed within the training set, and any parameters used (e.g., mean and standard deviation for z-scoring) should be saved to apply to variables in the test set. In addition, in studies in which related subjects are recruited into the same data sets, family structure needs to be taken into account. For example, instead of leaving one subject out, whole families may need to be left out of the training set. Here in the code, we assume that the subjects are all unrelated.

▲ **CRITICAL STEP** All free parameters or hyper-parameters that were used in the training set should be strictly followed in the test set. One should never change the parameter depending on the outcome of the test set. To optimize the free parameters, perform a two-layered cross-validation. The first layer is called the inner loop for parameter estimation, and the second layer is called the outer loop for prediction evaluation. Data are kept independent between the inner loop and the outer loop. Different models and parameters are tested within the inner loop, and the best model and parameters are selected. In the outer loop, the selected model and parameters are applied to the independent data and prediction is performed.

3| *Relate connectivity to behavior (Fig. 3a, item iii).* Across all subjects in the training set, use a form of linear regression, such as Pearson's correlation, Spearman's correlation, or robust regression, to relate each edge in the connectivity matrices to the behavioral measure. For simplicity, Pearson's correlation is used in the example script (Fig. 3a, item iii). Use Spearman's rank correlation as an alternative to Pearson's correlation if the behavioral data do not follow a normal distribution, and/or if a nonlinear (but monotonic) relationship between brain and behavior is expected³³. Use MATLAB to compute Spearman's correlation using a function call to 'corr()' and specifying the argument 'type' as 'spearman'. When using linear regression, additional covariates may be added to the regression analysis to account for possible confounding effects, such as age or sex. The MATLAB function 'partialcorr()' can be used for this purpose; 'partialcorr()' also has the option to calculate rank correlation. In addition, robust regression may be used to reduce

```

a
32
33 - clear;
34 - clc;
35
36 % ----- INPUTS -----
37
38 all_mats = rest_1_mats; (i)
39 all_behav = PMAT_CR;
40
41 % threshold for feature selection
42 thresh = 0.01;
43
44 % -----
45
46 no_sub = size(all_mats,3);
47 no_node = size(all_mats,1);
48
49 behav_pred_pos = zeros(no_sub,1);
50 behav_pred_neg = zeros(no_sub,1);
51
52
53 for leftout = 1:no_sub;
54     fprintf('\n Leaving out subj # %3f',leftout);
55
56 % leave out subject from matrices and behavior (ii)
57
58 train_mats = all_mats;
59 train_mats(:,leftout) = [];
60 train_vcts = reshape(train_mats,[],size(train_mats,3));
61
62 train_behav = all_behav;
63 train_behav(leftout) = [];
64
65 % correlate all edges with behavior
66
67 [r_mat,p_mat] = corr(train_vcts',train_behav); (iii)
68
69 r_mat = reshape(r_mat,no_node,no_node);
70 p_mat = reshape(p_mat,no_node,no_node);
71
b
Alternative 1
65 % correlate all edges with behavior using rank correlation
66
67 [r_mat, p_mat] = corr(train_vcts', train_behav, 'type', 'Spearman');
68
69 r_mat = reshape(r_mat,no_node,no_node);
70 p_mat = reshape(p_mat,no_node,no_node);
71
c
Alternative 2
65 % correlate all edges with behavior using partial correlation
66
67 train_age = all_age;
68 train_age(leftout) = [];
69
70 [r_mat, p_mat] = partialcorr(train_vcts', train_behav, train_age);
71
72 r_mat = reshape(r_mat,no_node,no_node);
73 p_mat = reshape(p_mat,no_node,no_node);
74
d
Alternative 3
65 % correlate all edges with behavior using robust regression
66
67 edge_no = size(train_vcts,1);
68 r_mat = zeros(1, edge_no);
69 p_mat = zeros(1, edge_no);
70
71 for edge_i = 1: edge_no;
72     [~, stats] = robustfit(train_vcts(edge_i,:), train_behav);
73     cur_t = stats.t(2);
74     r_mat(edge_i) = sign(cur_t)*sqrt(cur_t^2/(no_sub-1+cur_t^2));
75     p_mat(edge_i) = 2*tcdf(cur_t, no_sub-1-2); %two tailed
76 end
77
78 r_mat = reshape(r_mat,no_node,no_node);
79 p_mat = reshape(p_mat,no_node,no_node);
80

```

Figure 3 | Example CPM code for Steps 1–3. (a) (i) Load connectivity matrices and behavioral data into memory. (ii) Divide data into training and testing sets for cross-validation. In this example, leave-one-out cross-validation is used. (iii) Relate connectivity to behavior. In this example, Pearson's correlation is used. Note that the code outlined with a red box in **a** may be replaced with any of three alternatives in the right-hand panels: (b) rank (Spearman) correlation; (c) partial correlation; and (d) robust regression.

the influence of outliers^{34,35}. MATLAB has an implementation of robust regression in the function 'robustfit()'. Example scripts are shown in **Figure 3b–d**. To give a sense of run time, a single execution of the function 'robustfit' with 125 data samples and fewer than five outliers takes ~0.0023 s.

? TROUBLESHOOTING

4 | *Edge selection* (**Fig. 4a**, item iv). After each edge in the connectivity matrices is related to the behavior of interest, select the most relevant edges for use in the predictive model. Typically, this selection is based on the significance of the association between the edge and behavior. For example, a significance threshold of $P = 0.01$ may be applied (see **Fig. 4a**, line 42: 'thresh=0.01'), as in the example script (**Fig. 4a**, item iv). Investigators may wish to explore the effect of this threshold on model accuracy (although it is important to perform this exploration only within the training set; see CAUTION note in Step 2). In the **Supplementary Note**, we compare accuracy between the CPM and SVR models using a set of edges obtained at different thresholds.

(Optional) Instead of a whole-brain approach, only connections within or between predefined networks can be considered for experiments with *a priori* hypotheses. For example, we can restrict the feature selection to be only within the frontal–parietal network such that only edges belonging to a node in this network are kept for later analysis. Several large-scale network definitions exist (see, for example, Finn *et al.*⁴, Yeo *et al.*³⁶, and Power *et al.*¹²). Alternatively, networks can be estimated directly from the connectivity matrices using methods described in Finn *et al.*⁴ and Shen *et al.*¹¹.

! CAUTION When the number of *a priori* selected regions is very small, the risk of no edges or very few edges being selected within some iterations of cross-validation grows remarkably higher. This could lead to unstable models with poor predictive ability. Thus, we suggest using an atlas at the resolution of 200–300 regions covering the whole brain (this is consistent with observations made from the HCP Netmats MegaTrawl release).

▲ CRITICAL STEP Edges showing a significant ($P < 0.05$) brain–behavior relationship include those with both positive and negative (inverse) associations. In this protocol, these edges should be separated into two distinct sets in preparation for

```

a
68 r_mat = reshape(r_mat,no_node,no_node);
69 p_mat = reshape(p_mat,no_node,no_node);
70
71 % set threshold and define masks
72
73 pos_mask = zeros(no_node,no_node);
74 neg_mask = zeros(no_node,no_node);
75
76 pos_edges = find(r_mat > 0 & p_mat < thresh);
77 neg_edges = find(r_mat < 0 & p_mat < thresh);
78
79 pos_mask(pos_edges) = 1;
80 neg_mask(neg_edges) = 1;
81
82 % get sum of all edges in TRAIN subs (divide by 2 to control for the
83 % fact that matrices are symmetric)
84
85 train_sumpos = zeros(no_sub-1,1);
86 train_sumneg = zeros(no_sub-1,1);
87
88 for ss = 1:size(train_sumpos);
89 train_sumpos(ss) = sum(sum(train_mats(:,:,ss).*pos_mask))/2;
90 train_sumneg(ss) = sum(sum(train_mats(:,:,ss).*neg_mask))/2;
91 end
92
93 % build model on TRAIN subs
94
95 fit_pos = polyfit(train_sumpos, train_behav,1);
96 fit_neg = polyfit(train_sumneg, train_behav,1);
97
98 % run model on TEST sub
99
100 test_mat = all_mats(:,:,leftout);
101 test_sumpos = sum(sum(test_mat.*pos_mask))/2;
102 test_sumneg = sum(sum(test_mat.*neg_mask))/2;
103
104 behav_pred_pos(leftout) = fit_pos(1)*test_sumpos + fit_pos(2);
105 behav_pred_neg(leftout) = fit_neg(1)*test_sumneg + fit_neg(2);
106
107 end
108
109 % compare predicted and observed scores
110
111 [R_pos, P_pos] = corr(behav_pred_pos,all_behav);
112 [R_neg, P_neg] = corr(behav_pred_neg,all_behav);
113
114 figure(1); plot(behav_pred_pos,all_behav,'r'); %line
115 figure(2); plot(behav_pred_neg,all_behav,'b'); %line
116
117
118
119
120
121
122
b
Alternative
80 %-----sigmoidal weighting-----%
81 pos_edges = find(r_mat > 0);
82 neg_edges = find(r_mat < 0);
83
84 % covert p threshold to r threshold
85 T = tinv(thresh/2, no_sub-1-2);
86 R = sqrt(T^2/(no_sub-1-2+T^2));
87
88 % create a weighted mask using sigmoidal function
89 % weight = 0.5, when correlation = R/3;
90 % weight = 0.88, when correlation = R;
91 pos_mask(pos_edges) = sigmf( r_mat(pos_edges), [3/R, R/3]);
92 neg_mask(neg_edges) = sigmf( r_mat(neg_edges), [-3/R, R/3]);
93 %-----sigmoidal weighting-----%
c
Alternative
94
95 % build model on TRAIN subs
96 % combining both positive and negative features
97 b = regress(train_behav, [train_sumpos, train_sumneg, ones(no_sub-1,1)]);
98
99 % run model on TEST sub
100
101 test_mat = all_mats(:,:,leftout);
102 test_sumpos = sum(sum(test_mat.*pos_mask))/2;
103 test_sumneg = sum(sum(test_mat.*neg_mask))/2;
104
105 behav_pred(leftout) = b(1)*test_sumpos + b(2)*test_sumneg + b(3);
106
107 end
d
Alternative
133 % compare predicted and observed scores using mean squared error
134
135 MSE_pos = sum((behav_pred_pos-all_behav).^2)/(no_sub-length(fit_pos)-1)
136 MSE_neg = sum((behav_pred_neg-all_behav).^2)/(no_sub-length(fit_neg)-1)
137

```

Figure 4 | Example CPM code for Steps 4–8 (a). (Step 4) Edge selection. In this example, a significance threshold of $P = 0.01$ is used (see line 42 in Fig. 3: 'thresh = 0.01'). Alternatively, a sigmoidal weighting function may be used by replacing the code in the green box with the code provided in b. (Step 5) Forming single-subject summary values. For each subject in the training set, the selected edges are then summarized to a single value per subject for the positive edge set and the negative edge set separately. (Step 6) Model fitting. In this example, a linear model ($y = mx + b$) is fitted for the positive edge set and the negative edge set, separately. Alternatively, a model combining the two terms may be used by replacing the code in the red box with the code provided in c. (Step 7) Prediction in novel subjects. Single-subject summary values are calculated for each subject in the testing set and are used as an input to the predictive model (equation) estimated in Step 6. The resulting value is the predicted behavioral measure for the current subject. (Step 8) Evaluation of predictive model. Correlation and linear regression between the predicted values and true values provide measures to evaluate prediction performance. Alternatively, the predictive model may also be evaluated using mean-squared error by replacing the code in the blue box with the code provided in d.

the summary statistics calculation step. In the example code A (Fig. 4a, item iv), one variable ('pos_edge') contains only the significant positive correlations and one variable ('neg_edge') contains only the significant negative correlations. The set of positive and the set of negative edges may be interpreted differently in terms of their functional roles.

? TROUBLESHOOTING

5 | Calculate single-subject summary values (Fig. 4a, item v). For each subject in the training set, summarize the selected edges to a single value per subject for the positive edge set and the negative edge set separately, as shown in Figure 4a, item v. To follow the example, mask the individual connectivity matrices with the positive and negative edges selected in Step 4 and sum the edge-strength values for each individual (akin to computing the dot product between an individual connectivity matrix and the binary feature masks generated above). Alternatively, a sigmoid function centered at the P value threshold can be used as a weighting function, and a weighted sum can be used as the single-subject summary value; this obviates a binary threshold. Example code for using a sigmoid weight function is shown in Figure 4b.

6 | Model fitting (Fig. 4a, item vi). The CPM model assumes a linear relationship between the single-subject summary value (independent variable) and the behavioral variable (the dependent variable). This step can be performed for the positive edge set and the negative edge set separately, as shown in Figure 4a, item vi. Alternatively, a single linear model combining both positive edge and negative edge sets can be used (Fig. 4c). The number of variables used in the predictive models can be adjusted based on the specific application. For example, most models would include an intercept term to account for an offset. However, if Spearman's correlation and ranks are used, models excluding this intercept

term may perform better. In addition, higher-order polynomial terms or variance-stabilizing transformation of the summary values can be included to account for possible nonlinear effects.

7| Prediction in novel subjects (Fig. 4a, item vii). After the predictive model is estimated, calculate single-subject summary values for each subject in the testing set using the same methodology as in Step 5. For each subject in the testing set, this value should be input into the predictive model estimated in Step 6. The resulting value is the predicted behavioral measure for the current subject. In **Figure 4**, item vii, the behavioral measures are predicted for each testing subject separately for the positive edge set and the negative edge set.

▲ CRITICAL STEP The exact methodological choices made in Step 5 for the training subjects must be applied to the testing subjects.

8| Evaluation of the predictive model (Fig. 4a, item viii). After predicting the behavioral values for all subjects in the testing set, compare the predicted values with the true observed values to evaluate the predictive model. When a line is fitted to the predicted values (ordinate) versus the true values (abscissa), a slope greater than zero should be observed (i.e., positive correlation between predicted and observed values). If the slope is near one, the range of the predicted values closely matches the range of the observed values. A slope greater than one indicates that the predictive model overestimates the range of the true values, and a slope less than one indicates that the model underestimates the range of the true values. If the cross-validation is performed within a single data set (leave-one-out), statistical significance for the correlation between predicted and observed values should be assessed using permutation testing (see ‘Assessment of prediction significance’ below). If it is performed across data sets—i.e., an existing model trained on one data set is applied in a single calculation to an independent data set containing a nonoverlapping set of subjects—parametric statistics may be used to assign a *P* value to the resulting correlation coefficient.

Prediction performance can also be evaluated using the mean squared error (MSE) between the predicted values and the observed values (**Fig. 4d**). Correlation and MSE are usually dependent—i.e., higher correlation implies lower MSE and vice versa. Although correlation gives an estimate of relative predictive power (i.e., the model’s ability to predict where a novel subject will fall within a previously known distribution), MSE gives an estimate of the absolute predictive power (i.e., how much predicted values deviate from true values). We note that in the fMRI literature correlation seems to be the most common way to evaluate predictions of a continuous variable generated by cross-validation^{37–40}. Because correlation is always scaled between –1 and 1, it is easy to get an initial sense of how well the model is performing from the correlation coefficient; MSE, by contrast, does not scale to any particular values, and thus it is not particularly meaningful at first glance. However, in situations in which absolute accuracy of predictions is important, researchers may wish to use MSE as their primary measure of prediction assessment.

Calculate the true prediction correlation (**Fig. 5, item i**). In **Figure 5**, item i, these steps are assumed to be implemented in the function ‘predict_behavior’, which returns the correlation coefficient between the predicted and true behavioral measures.

Assessment of prediction significance ● TIMING 10 min–100 h

▲ CRITICAL Permutation testing is used to generate an empirical null distribution of the test statistic. A large number of possible values of the test statistic are calculated under random rearrangements of the labels for the data. Specifically, permutation is done by preserving the structure of the connectivity matrices but randomly reassigning behavioral scores (e.g., subject 1’s connectivity matrix is paired with subject 2’s behavioral score).

9| Shuffle data labels, calculate correlation coefficient, and repeat for 100–10,000 iterations (Fig. 5, item ii). After the true prediction correlation is calculated, randomly assign behavioral measurements to different subjects, breaking the true brain–behavior relationship. Use these new label assignments as inputs for the ‘predict_behavior’ function and store the resulting correlation coefficient for later use. Repeat this process 100–10,000 times in order to produce enough samples to estimate the distribution of the test statistic.

▲ CRITICAL STEP In the ideal case, all possible rearrangements of the data would be performed. However, for computational reasons, this is often not practical. Nevertheless, it is important that the number of iterations is large enough to properly estimate the distribution of the test statistic. Larger data sets may need a greater number of iterations to properly estimate *P* values.

10| Calculate *P* values (Fig. 5, item iii). Calculate the *P* value of the permutation test as the proportion of sampled permutations that are greater or equal to the true prediction correlation. The number of iterations determines the range of possible *P* values.

Visualization of data and generation of circle plots and 3D glass brain plots ● TIMING 10–20 min per plot

11| Open the tool in a web browser and switch between viewers. Open the following link: <http://bisweb.yale.edu/connviewer/>. (We have tested this primarily in Google Chrome, although other modern browsers should work—e.g., Firefox). Once the program loads, a pop-up window will appear that says ‘Connectivity Viewer initialized. The node definition loaded is from Shen

et al. Neuroimage 2013 with 268 nodes'. The user interface includes four panels: the circle plot panel (Fig. 6a, upper left), the 3D glass brain viewer panel (Fig. 6a, center), the orthogonal brain viewer panel (Fig. 6a, lower left), and the control panel (Fig. 6a, right). The user can use the 'Toggle 3D mode' button on the control panel to switch among the default settings for all three panels, with setting I focusing on the circle plots panel and setting II focusing on the 3D glass brain viewer. All three settings have the orthogonal brain viewer and the control panel available. Within the 3D glass brain viewer, the user can use the mouse to rotate the brain to an arbitrary viewing angle.

```

37 all_mats = rest_1_mats;
38 all_behav = PMAT_CR;
39
40 no_sub = size(all_mats,3);
41
42 % calculate the true prediction correlation
43 [true_prediction_r_pos, true_prediction_r_neg] = predict_behavior(all_mats, all_behav);
44
45 % number of iterations for permutation testing
46 no_iterations = 1000;
47 prediction_r = zeros(no_iterations,2);
48 prediction_r(1,1) = true_prediction_r_pos;
49 prediction_r(1,2) = true_prediction_r_neg;
50
51 % create estimate distribution of the test statistic
52 % via random shuffles of data labels
53
54 for it=2:no_iterations
55     fprintf('\n Performing iteration %d out of %d', it, no_iterations);
56     new_behav = all_behav(randperm(no_sub));
57     [prediction_r(it,1), prediction_r(it,2)] = predict_behavior(all_mats, new_behav);
58 end
59
60 sorted_prediction_r_pos = sort(prediction_r(:,1),'descend');
61 position_pos = find(sorted_prediction_r_pos==true_prediction_r_pos);
62 pval_pos = position_pos(1)/no_iterations;
63
64 sorted_prediction_r_neg = sort(prediction_r(:,2),'descend');
65 position_neg = find(sorted_prediction_r_neg==true_prediction_r_neg);
66 pval_neg = position_neg(1)/no_iterations;
67

```

Figure 5 | Example permutation test code for Steps 8–10. (i) Calculate the true prediction correlation. (ii) Shuffle data labels, calculate correlation coefficient, and repeat for 100–10,000 iterations. (iii) Calculate *P* values.

12 | *Load node definition.* When the visualization tool initializes for the first time, the program loads the 268-node brain parcellation that can be downloaded from https://www.nitrc.org/frs/?group_id=51. This is the parcellation image that was used in both the fluid intelligence study⁴ and the sustained attention study⁵.

In the circle plot panel (Fig. 6a, upper left), the inner dotted circles represent all nodes in the brain parcellation. The user can click on any one dot along the inner circle, and a curved line will be drawn from the center of the plot to the selected dot. Relevant information of the selected node will appear at the bottom of the panel, including the node index (defined by the input parcellation), lobe label, network label (derived from the Power atlas), Brodmann area label, the Montreal Neurological Institute (MNI) coordinates of the center of mass, and the internal node index (for plotting purposes). Simultaneously, the selected nodes will also be marked on the orthogonal brain viewer (Fig. 6a, lower left) and the 3D glass brain viewer (Fig. 6a, center) by a crosshair. Vice versa, one can select a node from the orthogonal viewer, and the corresponding dot in the circle plot will be marked. Nodes cannot be selected directly from the 3D glass brain viewer. In the circle plot, the nodes are organized into lobes and each lobe is color-coded accordingly. The color legends of the lobes can be turned on and off using the 'Toggle Legends' button in the Connectivity control tab in the control panel (Fig. 6a, right). The nodes in the orthogonal brain viewer are randomly colored.

Instead of using the default parcellation image, users can also load their own node definition into the visualization tool. There are two ways to provide such node definitions to the software. If the NIFTI (Neuroimaging Informatics Technology Initiative) image of the brain parcellation is available, one can load it via the 'Advanced' menu by selecting 'Import Node Definition (Parcellation) image'. If the image file is valid, a pop-up window will appear asking for a description of the newly loaded parcellation image. After the description is provided, the tool will need to save a .parc file to your local disk. This file contains a series of nodal data that are essentially the information being displayed under the circle plot. If a corresponding .parc file has been generated from a previous operation, one can directly load the .parc file via the 'File' menu by selecting 'Load Node Definition File'.

In many applications, nodes may be defined by the center of mass; therefore, we also provide a way to import node definition using the MNI coordinates only. By selecting 'Advanced' > 'Import Node Positions Text File', one can load a simple text file (with no header information) of *M* rows (number of nodes) and three columns (*x*, *y*, *z* coordinates). After the text file is determined to be valid, a .parc file will be generated and saved, and this file can be used for future purposes.

13 | *Load the positive and negative matrices.* Go to the 'File' menu and click 'Load Sample Matrices'. A pop-up window will appear when the matrices are loaded properly. The sample matrices are two binary matrices representing the set of positive features (edges) and the set of negative features (edges)—e.g., the matrices obtained in Step 4. Two schematic square images representing the positive and negative matrices will appear on the left side of the circle plot. The curved line is initialized to point to the node with the largest sum of degrees of both the positive and negative matrices. The information under the circle plot will be updated with additional information: nodal degree (number of edges) in the positive matrix, degree in the negative matrix, and sum of degrees. Go to the 'View' menu and select 'Show High Degree Nodes'; a list of top-degree nodes will be displayed and one can navigate through these nodes by clicking the 'GO' button.

To create one's own file of positive or negative matrices, use MATLAB 'save' function with the '-ascii' option. These should be binary matrices of size *M* by *M*, with 1 in elements for which an edge is to be visualized (i.e., element *i,j* should contain a 1 if an edge is to be drawn between node *i* and node *j*) and 0s elsewhere. Elements may be separated by comma, tab, or space delimiter. Users can load their own matrices by using the 'Load Positive Matrix' and (optionally) 'Load Negative Matrix' options from the 'File' menu. The display of the square matrix images may be turned on/off using the 'Toggle Legends' button in the Connectivity control tab, in the control panel.

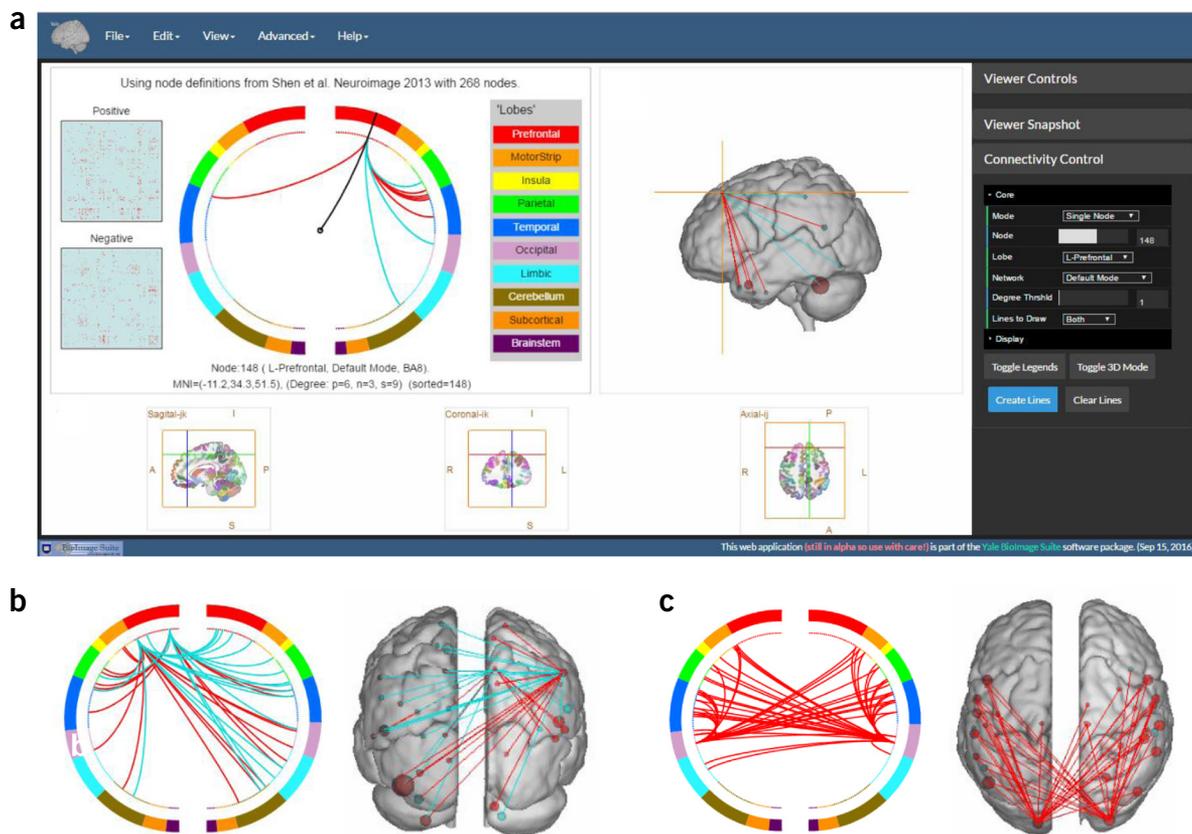


Figure 6 | Online visualization tool for making circle plots and glass brain plots described in Steps 11–14. The tool can be accessed via <http://bisweb.yale.edu/connviewer/>. (a) The user interface includes four panels. (Left) The circle plot and relevant information of a selected node. (Center) The 3D view of a glass MNI brain that can be rotated using a mouse. (Bottom) Three orthogonal views of the brain parcellation from Shen *et al.*¹¹ with color-coded nodes overlaid on top of an MNI brain. (Right) The set of control modules, including 'Viewer Controls', 'Viewer Snapshot', and 'Connectivity Control'. The 'Connectivity Control' module is expanded, and the six filters under 'Core' are shown with specified parameters. All three circle plots and the glass brain plots were generated using the sample matrices. Edges in the sample matrices are generated randomly, and the matrices are preloaded when the tool initializes. Circle plot and glass brain plot in **a** were created by setting 'Mode = Single Node', 'Node = 148', and 'Lines to Draw = Both'. (b) Circle plot and glass brain plot created by setting 'Mode = Single Lobe', 'Lobe = R-Prefrontal', 'Degree Threshld = 15', 'Lines to Draw = Both'. (c) Circle plot and glass brain plot created by setting 'Mode = Single Network', 'Network = Visual', 'Degree Threshld = 35', 'Lines to Draw = Positive'. These figures were created to instruct the users in how to set the different parameters in the visualization tool.

14 | *Make a circle plot and a 3D glass brain plot.* Expand the 'Connectivity Control' tab in the control panel and notice that under the header 'Core' there are six filters for the users to specify: (i) mode, (ii) node, (iii) lobe, (iv) network, (v) threshold, and (vi) lines to draw.

There are four options under the filter 'Mode' that control the proportion of connections to be drawn in the circle plot. By choosing 'All', one plots all connections in the matrix. 'Single Node' plots connections from a chosen node specified by the second filter 'Node'. 'Single Lobe' plots connections from all nodes within a lobe specified by the third filter 'Lobe' (default lobe definition is preloaded to the tool). 'Single Network' plots connections from all nodes within a network specified by the fourth filter 'Network'. The Power atlas was used to generate the network labels for the brain parcellation and is preloaded to the tool. When a mode is chosen, the filters that are irrelevant to the chosen mode are ignored—e.g., if one chooses the 'Single Node' mode, specifying the filter 'Lobe' or 'Network' will not have an effect on the plot drawing. The fifth filter 'Degree Threshld' sets a threshold value for nodal degree when plotting under the 'All', 'Single Lobe', or 'Single Network' option. A connection will be plotted only when at least one of the emanating nodes has a degree value (in this case the sum of positive and negative degrees) greater than the threshold. The last filter, 'Lines to Draw', specifies whether the plot includes connections from positive matrix only, from negative matrix only, or from both matrices. When all filters have been set, click the 'Create Lines' button and the circle plot and the glass brain plot will be generated. In both plots, the red lines are connections defined in the positive matrix and the cyan lines are connections in the negative matrix. The size of the node in the 3D glass brain plot is proportional to the sum of positive and negative degrees of the nodes. The node is colored red if the positive degree is greater, and the node is colored cyan if the opposite is true.

In **Figure 6**, we show several examples of circle plots and glass brain plots. **Figure 6a** (upper left) shows a circle plot from one node in the prefrontal region; parameters are 'Mode = Single Node', 'Node = 148', and 'Lines to Draw = Both'.

Figure 6a (center) is the glass brain plot of the same node viewed from left. Six standard viewing angles of the glass brain can be selected via the 'View' menu by selecting 'Select 3D View to Front/Back/Top/Bottom/Left/Right'. **Figure 6b** shows a circle plot and a glass brain plot from nodes in the right prefrontal region; parameters are 'Mode = Single Lobe', 'Lobe = R-Prefrontal', 'Degree Thrshld = 15', 'Lines to Draw = Both'. **Figure 6c** shows a circle plot and a glass brain plot from nodes within the visual network; parameters are 'Mode = Single Network', 'Network = Visual', 'Degree Thrshld = 35', 'Lines to Draw = Positive'.

! CAUTION Do not use 'All' mode option unless the connectivity matrix is thresholded to be very sparse.

! CAUTION The lines in the circle plot and glass brain plot will not be erased until the 'Clear Lines' button is clicked. In other words, connections generated from different set of parameters can be drawn on top of each other to create composite plots.

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 1**.

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason(s)	Solution
1	Non-normal distribution of behavioral data	Relatively small sample size, or behavior does not follow normal distribution in the population	Use rank order statistics instead of parametric ones (i.e., Spearman's (rank) correlation instead of Pearson's correlation)
3	Insignificant ($P \geq 0.05$) relationship between connectivity and behavior	(i) Outliers in the data; (ii) nonlinear relationship; (iii) data do not contain the required information for predicting the variable (i.e., there is no evidence for a link between neuroimaging and the behavior of interest; null result)	(i) Use robust regression to minimize the impact of outliers; (ii) use rank correlation to capture nonlinear effects; (iii) we are unable to provide a suggestion for what neuroimaging data should be collected to guarantee a successful prediction. It is a possibility that for some behavior the link to neuroimaging data does not exist
4	Low overlap of edges in each leave-one-out iteration	(i) Threshold introduces sharp nonlinear effects; (ii) data do not contain the required information for predicting the variable (see (iii) in the Step 3 entry), thus the model is overfitting to noise specific to each iteration	(i) Use a looser significance threshold (ii) Use a weighting sigmoidal scheme to reduce threshold nonlinearity
	In some cross-validation iterations, single-subject summary statistics equal zero	No edges pass the significance threshold for feature selection	Use a looser significance threshold, or use a weighting sigmoidal scheme

● TIMING

Steps 1–8, feature selection, summarization, and model building using CPM: 10–100 min, depending on the number of subjects, parameter choices, and computer. Using 100 subjects and a 2.7-GHz Core i7 computer, the example script takes just under a minute to run.

Steps 9 and 10, permutation testing: this can be computationally expensive and can take between a few minutes to several days, depending on choices in parameters.

Steps 11–14, visualization: these steps are mainly exploratory and it may take 10–20 min for users to generate a circle plot and a 3D glass brain plot.

ANTICIPATED RESULTS

The final result should be a correlation between predicted behavior and true behavior in the testing set. Typically, this correlation will be between 0.2 and 0.5, although we have reported correlations as high as $r = 0.87$ in Rosenberg *et al.*⁵. As outlined in Whelan *et al.*¹⁴, prediction results will often have lower within-sample effect sizes than results generated with simple correlation analysis. This is expected, as simple correlation analysis can dramatically overfit the data and increase within-sample effect size. Thus, the cross-validated prediction result provides a more conservative estimate of the strength of the brain–behavior relationship. Such a result is more likely to generalize to independent data and, eventually, to contribute to the development of neuroimaging-based assessments with practical utility.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS M.D.R. and E.S.F. are supported by US National Science Foundation Graduate Research Fellowships. This work was also supported by US National Institutes of Health grant EB009666 to R.T.C.

AUTHOR CONTRIBUTIONS X.S., E.S.F., D.S., X.P., and R.T.C. conceptualized the study. X.S. developed this protocol with help from E.S.F. and D.S. E.S.F. developed the prediction framework with help from X.S. and M.D.R. E.S.F., X.P., and X.S. contributed previously unpublished tools. X.P. developed the online visualization tools with help from X.S. and D.S. X.P., M.M.C., and R.T.C. provided support and guidance with data interpretation. All authors made valuable comments on the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare competing financial interests: details are available in the [online version of the paper](#).

Reprints and permissions information is available online at <http://www.nature.com/reprints/index.html>.

- Kriegeskorte, N., Simmons, W.K., Bellgowan, P.S. & Baker, C.I. Circular analysis in systems neuroscience: the dangers of double dipping. *Nat. Neurosci.* **12**, 535–540 (2009).
- Vul, E., Harris, C., Winkielman, P. & Pashler, H. Puzzlingly high correlations in fMRI studies of emotion, personality, and social cognition. *Perspect. Psychol. Sci.* **4**, 274–290 (2009).
- Gabrieli, J.D., Ghosh, S.S. & Whitfield-Gabrieli, S. Prediction as a humanitarian and pragmatic contribution from human cognitive neuroscience. *Neuron* **85**, 11–26 (2015).
- Finn, E.S. *et al.* Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nat. Neurosci.* **18**, 1664–1671 (2015).
- Rosenberg, M.D. *et al.* A neuromarker of sustained attention from whole-brain functional connectivity. *Nat. Neurosci.* **19**, 165–171 (2016).
- Van Essen, D.C. *et al.* The WU-Minn human connectome project: an overview. *Neuroimage* **80**, 62–79 (2013).
- Nooner, K.B. *et al.* The NKI-Rockland Sample: a model for accelerating the pace of discovery science in psychiatry. *Front. Neurosci.* **6**, 152 (2012).
- Milham, M.P., Fair, D., Mennes, M. & Mostofsky, S.H. The ADHD-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Front. Syst. Neurosci.* **6**, 62 (2012).
- Satterthwaite, T.D. *et al.* The Philadelphia Neurodevelopmental Cohort: a publicly available resource for the study of normal and abnormal brain development in youth. *Neuroimage* **124**, 1115–1119 (2016).
- Van Essen, D.C., Glasser, M.F., Dierker, D.L., Harwell, J. & Coalson, T. Parcellations and hemispheric asymmetries of human cerebral cortex analyzed on surface-based atlases. *Cereb. Cortex* **22**, 2241–2262 (2012).
- Shen, X., Tokoglu, F., Papademetris, X. & Constable, R.T. Groupwise whole-brain parcellation from resting-state fMRI data for network node identification. *Neuroimage* **82**, 403–415 (2013).
- Power, J.D. *et al.* Functional network organization of the human brain. *Neuron* **72**, 665–678 (2011).
- Craddock, R.C., James, G.A., Holtzheimer, P.E., Hu, X.P. & Mayberg, H.S. A whole brain fMRI atlas generated via spatially constrained spectral clustering. *Hum. Brain Mapp.* **33**, 1914–1928 (2012).
- Whelan, R. & Garavan, H. When optimism hurts: inflated predictions in psychiatric neuroimaging. *Biol. Psychiatry* **75**, 746–748 (2014).
- Smith, S. *et al.* HCP beta-release of the Functional Connectivity MegaTrawl. https://www.humanconnectome.org/documentation/S500/HCP500_MegaTrawl_April2015.pdf (2015).
- Smith, S.M. *et al.* Functional connectomics from resting-state fMRI. *Trends Cogn. Sci.* **17**, 666–682 (2013).
- Smola, A. & Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **9**, 155–161 (1997).
- Dosenbach, N.U. *et al.* Prediction of individual brain maturity using fMRI. *Science* **329**, 1358–1361 (2010).
- Brown, M.R. *et al.* ADHD-200 global competition: diagnosing ADHD using personal characteristic data can outperform resting state fMRI measurements. *Front. Syst. Neurosci.* **6**, 69 (2012).
- Plitt, M., Barnes, K.A. & Martin, A. Functional connectivity classification of autism identifies highly predictive brain features but falls short of biomarker standards. *NeuroImage Clin.* **7**, 359–366 (2015).
- Anderson, J.S. *et al.* Functional connectivity magnetic resonance imaging classification of autism. *Brain* **134**, 3742–3754 (2011).
- Arbabshirani, M.R., Kiehl, K.A., Pearlson, G.D. & Calhoun, V.D. Classification of schizophrenia patients based on resting-state functional network connectivity. *Front. Neurosci.* **7**, 133 (2013).
- Khazaee, A., Ebrahimzadeh, A. & Babajani-Feremi, A. Identifying patients with Alzheimer’s disease using resting-state fMRI and graph theory. *Clin. Neurophysiol.* **126**, 2132–2141 (2015).
- Zeng, L.L. *et al.* Identifying major depression using whole-brain functional connectivity: a multivariate pattern analysis. *Brain* **135**, 1498–1507 (2012).
- Haufe, S. *et al.* On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage* **87**, 96–110 (2014).
- Van Dijk, K.R., Sabuncu, M.R. & Buckner, R.L. The influence of head motion on intrinsic functional connectivity MRI. *Neuroimage* **59**, 431–438 (2012).
- Yendiki, A., Koldewyn, K., Kakunoori, S., Kanwisher, N. & Fischl, B. Spurious group differences due to head motion in a diffusion MRI study. *Neuroimage* **88C**, 79–90 (2013).
- Power, J.D., Schlaggar, B.L. & Petersen, S.E. Recent progress and outstanding issues in motion correction in resting state fMRI. *Neuroimage* **105**, 536–551 (2015).
- Yan, C.-G. *et al.* A comprehensive assessment of regional variation in the impact of head micromovements on functional connectomics. *Neuroimage* **76**, 183–201 (2013).
- Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. in *Proceedings of the 14th International Joint Conference on Artificial Intelligence* **2**, 1137–1143 (1995).
- Justice, A.C., Covinsky, K.E. & Berlin, J.A. Assessing the generalizability of prognostic information. *Ann. Intern. Med.* **130**, 515–524 (1999).
- Steyerberg, E.W. *et al.* Assessing the performance of prediction models: a framework for traditional and novel measures. *Epidemiology* **21**, 128–138 (2010).
- Gibbons, J.D. & Chakraborti, S. *Nonparametric Statistical Inference* (Springer, 2011).
- Holland, P.W. & Welsch, R.E. Robust regression using iteratively reweighted least-squares. *Commun. Stat.-Theory Methods* **6**, 813–827 (1977).
- Street, J.O., Carroll, R.J. & Ruppert, D. A note on computing robust regression estimates via iteratively reweighted least squares. *Am. Stat.* **42**, 152–154 (1988).
- Yeo, B.T. *et al.* The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *J. Neurophysiol.* **106**, 1125–1165 (2011).
- Li, N. *et al.* Resting-state functional connectivity predicts impulsivity in economic decision-making. *J. Neurosci.* **33**, 4886–4895 (2013).
- Supekar, K. *et al.* Neural predictors of individual differences in response to math tutoring in primary-grade school children. *Proc. Natl. Acad. Sci. USA* **110**, 8230–8235 (2013).
- Doehrmann, O. *et al.* Predicting treatment response in social anxiety disorder from functional magnetic resonance imaging. *JAMA Psychiatry* **70**, 87–97 (2013).
- Ullman, H., Almeida, R. & Klingberg, T. Structural maturation and brain activity predict future working memory capacity during childhood development. *J. Neurosci.* **34**, 1592–1598 (2014).