Predicting individual traits from models of brain dynamics accurately and reliably using the Fisher kernel

Ahrends C[1], Woolrich M[2], & Vidaurre D[1,2]

[1] Center of Functionally Integrative Neuroscience, Department of Clinical Medicine, Aarhus University, Aarhus, Denmark

[2] Oxford Centre for Human Brain Activity, Department of Psychiatry, University of Oxford, Oxford, United Kingdom

Abstract

Predicting an individual's cognitive traits or clinical condition using brain signals is a central goal in modern neuroscience. This is commonly done using either structural aspects, or aggregated measures of brain activity that average over time. But these approaches are missing what can be the most representative aspect of these complex human features: the uniquely individual ways in which brain activity unfolds over time, that is, the dynamic nature of the brain. The reason why these dynamic patterns are not usually taken into account is that they have to be described by complex, high-dimensional models; and it is unclear how best to use information from these models for a prediction. We here propose an approach that describes dynamic functional connectivity and amplitude patterns using a Hidden Markov model (HMM) and combines it with the Fisher kernel, which can be used to predict individual traits. The Fisher kernel is constructed from the HMM in a mathematically principled manner, thereby preserving the structure of the underlying HMM. In this way, the unique, individual signatures of brain dynamics can be explicitly leveraged for prediction. We here show in fMRI data that the HMM-Fisher kernel approach is not only more accurate, but also more reliable than other methods, including ones based on time-averaged functional connectivity. This is important because reliability is critical for many practical applications, especially if we want to be able to meaningfully interpret model errors, like for the concept of brain age. In summary, our approach makes it possible to leverage information about an individual's brain dynamics for prediction in cognitive neuroscience and personalised medicine.

1 Introduction

Observing a person's behaviour over time is how we understand the individual's personality, cognitive traits, or psychiatric condition such as schizophrenia or depression. The same applies at the brain level, by observing the patterns by which its activity unfolds over time. However, although research into brain dynamics has gained traction over the last years, there is currently no established methodology to use this level of description to characterise subject differences or predict individual traits. Since brain dynamics are not an explicit feature of the data, intermediate representations of the dynamics are used as features for prediction. These representations can be estimated from modalities such as fMRI or MEG using models of varying complexity (Lurie et al., 2019), and are usually parametrised by a high number of parameters with complex mathematical relationships between them that reflect the model assumptions. How to leverage these parameters for prediction is an open question.

We here propose the Fisher kernel for predicting individual traits from brain dynamics, using information of how brain activity fluctuates over time during unconstrained cognition —that is, during wakeful rest. We focus on models of brain dynamics that capture within-session changes in functional connectivity and amplitude from fMRI scans, and how the parameters from these models can be used to predict behavioural variables or traits. In particular, we use the Hidden Markov Model (HMM) as a model of time-varying amplitude and functional connectivity (FC) dynamics (Vidaurre et al., 2017), which was previously shown to be able to predict certain subject traits, such as fluid intelligence, more accurately than structural or static (time-averaged) FC representations (Vidaurre et al., 2021). Specifically, the Fisher kernel allows using the entire set of parameters of a generative probabilistic model of brain dynamics in a computationally efficient way. It takes the complex relationships between the parameters into account, preserving the structure of the underlying model of brain dynamics (here, the HMM) (Jaakkola et al., 1999; Jaakkola & Haussler, 1998).

For empirical evaluation, we consider two criteria that are important in both scientific and practical applications. First, predictions should be as accurate as possible; i.e., the correlation between predicted and actual values should be high. Second, they should be reliable, in the sense that a predictive model should never produce excessively large errors, and the outcome should be robust to the choice of which subjects are included in the training set. The latter criterion is especially important if we want to be able to meaningfully interpret prediction errors, e.g., in assessing brain age (Cole & Franke, 2017; Denissen et al., 2022; Smith et al., 2019). Despite this crucial role in interpreting model errors, reliability is not often taken into account in models predicting individual traits from neuroimaging features.

In summary, we show that using the Fisher kernel approach, which preserves the complex structure of the underlying HMM, we can predict individual traits from brain dynamics accurately and reliably. We show that our approach significantly outperforms methods that do not take the mathematical structure of the model into account, as well as methods based on time-averaged FC that do not consider brain dynamics. For interpretation, we also investigate which aspects of the brain dynamics drive the prediction accuracy. Bringing accuracy, reliability and interpretation together, this work therefore opens possibilities for practical applications such as the development of biomarkers and the investigation of individual differences in cognitive traits.

2 Methods

We here aim to predict behavioural and demographic variables from a model of brain dynamics using different kernel functions. The general workflow is illustrated in **Figure 1**. We start with the concatenated fMRI time-series of a group of subjects (**Figure 1**, step 1). We estimate a state-space model of brain dynamics at the group level, where the state descriptions, initial state probabilities, and the state transition probability matrix are shared across subjects (**Figure 1**, step 2). Next, we estimate subject-specific versions of this group-level brain dynamics model by dual estimation, where the group-level HMM parameters are re-estimated to fit the individual-level timeseries (Vidaurre et al., 2021) (**Figure 1**, step 3). Next, we use this description of the individuals' brain dynamics to predict their individual traits. For this, we first map the parameters from the model of brain dynamics into a feature space (**Figure 1**, step 4). This works in different ways for the different kernels: In the naïve kernel (step 4a), the features are simply the parameters in the Euclidean space; while in the Fisher kernel (step 4b), the features are mapped into the gradient space. We then estimate the similarity between each pair of subjects in this feature space using kernel functions. Finally, we use these kernels to predict the behavioural variables using kernel ridge regression (**Figure 1**, step 5). The first three steps are identical for all kernels and therefore carried out only once. The fourth step (mapping the examples and constructing the kernels) is carried out once for each of the different kernels. The last step is repeated 3,500 times for each kernel to predict a set of 35 different behavioural variables using 100 randomised iterations of 10-fold nested cross validation. We evaluate 24,500 predictive models using different kernels constructed from the same model of brain dynamics in terms of their ability to predict phenotypes, as well as 14,000 predictive models based on time-averaged features.

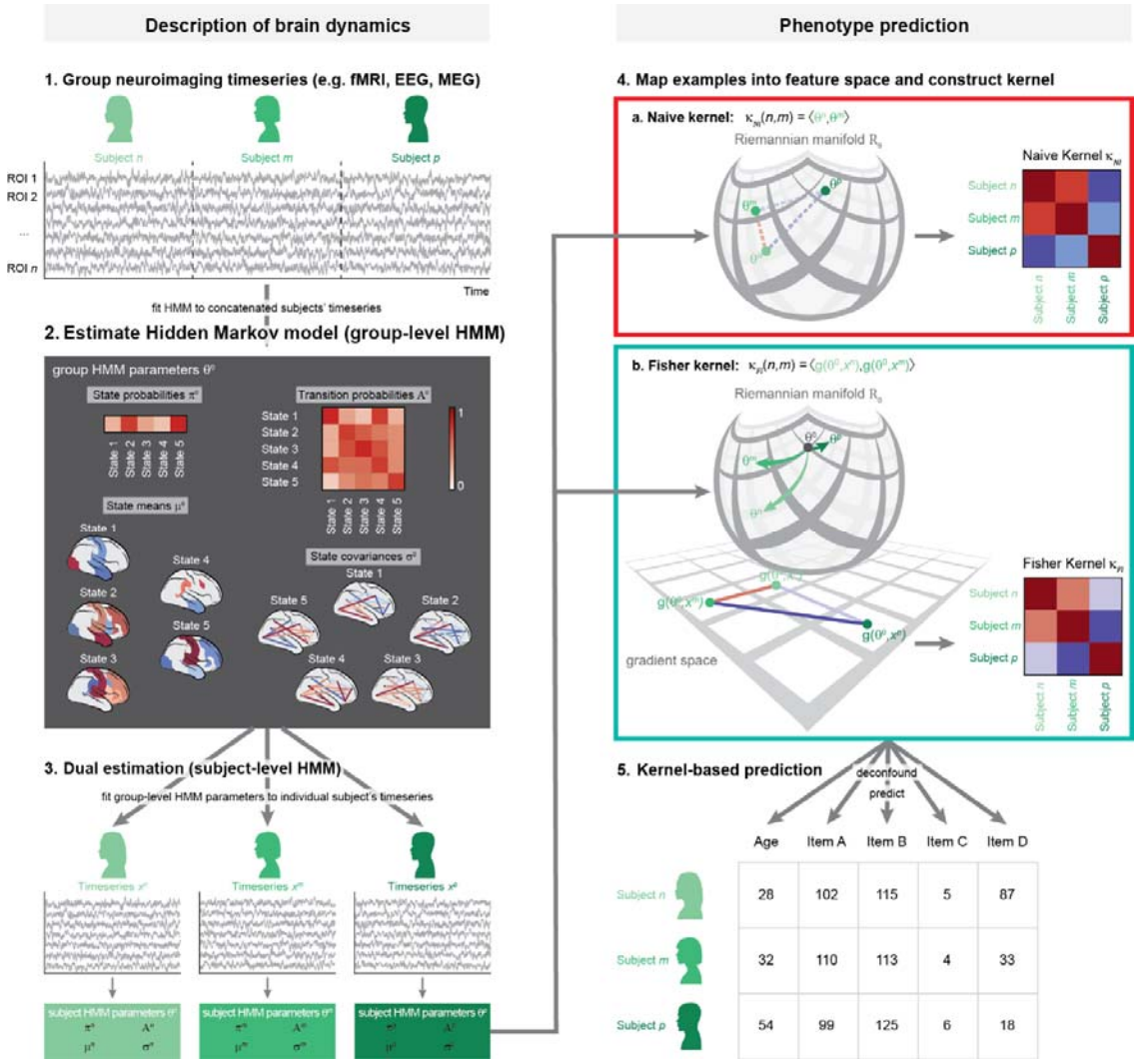

Figure 1 *Workflow of the Fisher kernel prediction approach. To generate a description of brain dynamics, we (1) concatenate all subjects' individual timeseries; then (2) estimate a Hidden Markov Model (HMM) on these timeseries to generate a group-level description of brain dynamics; then (3) dual-estimate into subject-level HMM models. Steps 1-3 are the same for all kernels. In order to then use this description of all subjects' individual brain dynamics, we map each subject into a feature space (4). This mapping can be done in different ways: In the naïve kernels (a), the manifold on which the parameters lie is ignored and examples are treated as if they were in Euclidean space. The Fisher kernel (b), on the other hand, reflects the structure of the parameters in their original Riemannian manifold by working in the gradient space. We then construct kernel matrices κ, where each pair of subjects has a similarity value given their parameters in the respective embedding space. Finally, we feed κ to kernel ridge regression to predict a variety of demographic and behavioural traits in a cross-validated fashion (5).*

2.1 HCP imaging and behavioural data

We used data from the open-access Human Connectome Project (HCP) S1200 release (Smith, Beckmann, et al., 2013; Van Essen et al., 2013), which contains MR imaging data and various demographic and behavioural data from 1,200 healthy, young adults (age 22-35). All data described below, i.e., timecourses of the resting-state fMRI data and demographic and behavioural variables, are publicly available at https://db.humanconnectome.org.

Specifically, we used resting state fMRI data of 1,001 subjects, for whom any of the behavioural variables of interest were available. Each participant completed four resting state scanning sessions of 14 mins and 33 seconds duration each. This resulted in 4,800 timepoints per subject. For the main results, we used all four resting state scanning sessions of each participant to fit the model of brain dynamics (but see **Supplementary Figure 1** for results with just one session). The acquisition parameters are described in the HCP acquisition protocols and in Van Essen et al. (2013); Van Essen et al. (2012). Briefly, structural and functional MRI data were acquired on a 3T MRI scanner. The resting state fMRI data was acquired using multiband echo planar imaging sequences with an acceleration factor of 8 at 2 mm isotropic spatial resolution and a repetition time (TR) of 0.72 seconds. The preprocessing and timecourse extraction pipeline is described in detail in Smith, Beckmann, et al. (2013). For the resting state fMRI scans, preprocessing consisted of minimal spatial preprocessing and surface projection (Glasser et al., 2013), followed by temporal preprocessing. Temporal preprocessing consisted of single-session Independent Component Analysis (ICA) (Beckmann, 2012) and removal of noise components (Griffanti et al., 2014; Salimi-Khorshidi et al., 2014). Data were high-pass-filtered with a cut-off at 2,000 seconds to remove linear trends.

The parcellation was estimated from the data using multi-session spatial ICA on the temporally concatenated data from all subjects. Using this approach, a data-driven functional parcellation with 50 parcels was estimated, where all voxels are weighted according to their activity in each parcel, resulting in a weighted, overlapping parcellation. While other parcellations are available for the resting-state fMRI HCP dataset, we chose this parcellation because dynamic changes in FC can be better detected in this parcellation compared to other functional or anatomical parcellations or more fine-grained parcellations (Ahrends et al., 2022). Timecourses were extracted using dual regression (Beckmann et al., 2009), where group-level components are regressed onto each subject's fMRI data to obtain subject-specific versions of the parcels and their timecourses. We normalised the timecourses of each subject to ensure that the model of brain dynamics and, crucially, the kernels were not driven by (averaged) amplitude and variance differences between subjects.

Subjects in the HCP study completed a range of demographic and behavioural questionnaires. Following Vidaurre et al. (2021), we here focus on a subset of those items, including age and various cognitive variables. The cognitive variables span items assessing memory, executive function, fluid intelligence, language, processing speed, spatial orientation, and attention. The full list of the 35 behavioural variables used here, as well as their categorisation within the HCP dataset can be found in **Supplementary Table 1**.

2.2 The Hidden Markov Model as a model of brain dynamics

To estimate brain dynamics, we here use the Hidden Markov Model (Vidaurre et al., 2016; Vidaurre et al., 2017). However, the kernels, which are explained in detail in the following section, can be constructed from any generative probabilistic model.

The Hidden Markov model (HMM) is a generative probabilistic model, which assumes that an observed time-series, such as BOLD signal in a given parcellation, was generated by a sequence of "hidden states" (Baum & Eagon, 1967; Baum & Petrie, 1966). We here model the states as Gaussian distributions, defined both in terms of mean and covariance —which can be interpreted as distinct patterns of amplitude and FC (Vidaurre et al., 2017). Since we are often interested in modelling time-varying FC specifically, we repeated all computations for a variety of the HMM where state means were pinned to zero (given that the data was demeaned and the global average is zero) and only the covariance was allowed to vary across states (Vidaurre et al., 2017), which is shown in **Supplementary Figure 2**.

The HMM is described by a set of parameters θ, containing the state probabilities π, the transition probabilities A, the mean vectors μ of all states, and the covariance matrices σ of all states:

$$\theta = [\pi, A, \mu, \sigma] \qquad \pi \in \mathbb{R}^{1\mathrm{x}K}, A \in \mathbb{R}^{K\mathrm{x}K}, \mu \in \mathbb{R}^{K\mathrm{x}M}, \sigma \in \mathbb{R}^{K\mathrm{x}MxM}$$

where K is the number of states and M is the number of parcels in the parcellation. The entire set of parameters θ is estimated from the data. Here, we chose a small number of states, $K = 6$, to ensure that the group-level HMM states are general enough to be found in all subjects, since a larger number of states increases the chances of certain states being present only in a subset of subjects. Note that the same HMM estimation is used for all kernels.

The HMM is a generative model. The generative process in this case works by sampling from a Gaussian distribution with mean μ_k and covariance σ_k when state k is active:

$$X_t | q_t = k \sim N(\mu_k, \sigma_k)$$

where q_t is the currently active state. Which state is active depends on the previous state q_{t-1} and is determined by the transition probabilities A, so that the generated state sequence is sampled from a categorical distribution with parameters:

$$q_t | q_{t-1} = k \sim Cat(A_k)$$

where A_k indicates the k-th row of the transition probability matrix.

The space of parameters θ forms a Riemannian manifold R_θ, where the relationships between the different parameters of the HMM are acknowledged by construction. The Fisher kernel, as described below, is built upon a projection on this manifold, so predictions based on this kernel account for the mathematical structure of the HMM.

Here, we fit the HMM to the concatenated timeseries of all N subjects (see **Figure 1a**, step 1). We refer to the group-level estimate as HMM^0, which is defined by the parameters θ^0 (see **Figure 1a**, step 2):

$$\theta^0 = [\pi^0, A^0, \mu^0, \sigma^0]$$

In order to use the information from the HMM to predict subjects' phenotypes, we estimate subject-specific versions of the group-level HMM (see **Figure 1a**, step 3) through dual estimation (Vidaurre et al., 2017). Dual estimation refers to the process of fitting the previously estimated group-level model again to a single subject's timeseries, so that the parameters from the group-level model HMM^0 are adapted to fit the individual. We will refer to the subject-specific estimate for subject n as HMM^n, with parameters θ^n.

These subject-specific HMM parameters are the features that we construct kernels from. To understand which features are most important for the predictions, we also construct versions of the kernels that include only subsets of the features. Specifically, we can group the features into two subsets: 1. the state features, describing *what* states look like, containing the mean vectors μ, and the covariance matrices σ of all states, and 2. the transition features, describing *how* individuals transition between these states, containing the initial state probabilities π and the transition probabilities A. By removing one or the other set of features and evaluating how model performance changes compared to the full kernels, we can draw conclusions about the importance of these two different types of changes for the predictions. Since the state features are considerably more numerous than the transition features (15,300 state features compared to 42 transition features in this case), we also construct a version of the kernels where state features have been reduced to the same number as the transition features using PCA, i.e., we use all 42 transition features and the first 42 PCs of the state features.

2.3 Kernels from Hidden Markov Models

Kernels (Shawe-Taylor & Cristianini, 2004) are a convenient approach to work with high-dimensional, complex features, such as parameters from a model of brain dynamics. In general, kernels are similarity functions between subjects, and they can be used straightforwardly in a prediction algorithm. While feature matrices can be very high

dimensional, a kernel is represented by a (no. of subjects by no. of subjects) matrix. Kernel methods can readily be adapted to deal with nonlinear decision boundaries in prediction, by projecting the data into a high-dimensional (possibly infinite-dimensional) space through an embedding $x \rightarrow \phi_x$; then, by estimating a linear separating hyperplane on this space, we can effectively have a nonlinear estimator on the original space (Shawe-Taylor & Cristianini, 2004). In practice, instead of working explicitly in a higher-dimensional embedding space, the so-called kernel trick uses a kernel function $\kappa(n, m)$ containing the similarity between data points n and m (here, subjects) in the higher-dimensional embedding space (Schölkopf et al., 2002; Shawe-Taylor & Cristianini, 2004), which can be simpler to calculate. Once $\kappa(\cdot,\cdot)$ is computed for each pair of subjects, this is all that is needed for the prediction. This makes kernels computationally very efficient, since in most cases the number of subjects will be smaller than the number of features —which, in the case of HMMs, can be very large (potentially, in the order of millions). However, finding the right kernel can be a challenge because there are many available alternatives for the embedding.

Here, in combination with a linear predictive model, we apply a kernel that is specifically conceived to be used to compare instances of a generative model like the HMM. We expected that this will result in better predictions than existing methods. Using the same HMM estimate, we compare three different kernels, which map the HMM parameters into three distinct spaces, corresponding to different embeddings (see **Figure 1b**, step 4): the naïve kernel, the naïve normalised kernel, and the Fisher kernel. While the first two kernels (naïve and naïve normalised kernel) do not take the relationships between the HMM's parameters into account, the Fisher kernel preserves the structure of the HMM and weights the parameters according to how much they contribute to generating a particular subject's timeseries. We then construct linear and Gaussian versions of the different kernels (see **Figure 1b**, step 5), which take the general form $\kappa_l(n, m) = {\phi_x n}^T \phi_x m$ for the linear kernel and $\kappa_g(n, m) = exp\ (-\frac{\|\phi_x n - \phi_x m\|^2}{2\tau^2})$ for the Gaussian kernel. We compare these to a kernel constructed using Kullback-Leibler divergence, previously used for predicting behavioural phenotypes (Vidaurre et al., 2021).

2.3.1 Naïve kernel

The naïve kernel is based on a simple vectorisation of the subject-specific version of the HMM's parameters, each on their own scale. This means that the kernel does not take relationships between the parameters into account and the parameters are here on different scales. This procedure can be thought of as computing Euclidean distances between two sets of HMM parameters, ignoring the actual geometry of the space of parameters. For each

subject n, we vectorise parameters θ^n obtained through dual estimation of the group-level parameters θ^0 to map the example $x^n \rightarrow \phi_{x^n}$ to:

$$\theta^n = (\pi^n, A^n, \mu^n, \sigma^n) \qquad \theta^n \in \mathbb{R}^{1x(K+K*K+K*M+K*M*M)}$$

We will refer to this vectorised version of the subject-specific HMM parameters as "naïve θ". The naïve θ are the features used in the naïve kernel κ_N. We first construct a linear kernel from the naïve θ features using the inner product of the feature vectors. The linear naïve kernel κ_{Nl} between subjects n and m is thus defined as:

$$\kappa_{Nl}(n,m) = \langle \theta^n, \theta^m \rangle \qquad \kappa_{Nl} \in \mathbb{R}^{NxN}$$

where $\langle \theta^n, \theta^m \rangle$ denotes the inner product between θ^n and θ^m. Using the same feature vectors, we can also construct a Gaussian kernel from the naïve θ. The Gaussian naïve kernel κ_{Ng} for subjects n and m is defined as:

$$\kappa_{Ng}(n,m) = \exp\left(-\frac{\|\theta^n - \theta^m\|^2}{2\tau^2}\right) \qquad \kappa_{Ng} \in \mathbb{R}^{NxN}$$

where τ is the radius of the radial basis function, and $\|\theta^n - \theta^m\|$ is the L_2-norm of the difference of the feature vectors (naïve θ for subjects n and m). Compared to a linear kernel, a Gaussian kernel embeds the features into a more complex space, which can potentially improve the accuracy. However, this kernel has an additional parameter τ that needs to be chosen, typically through cross-validation. This makes a Gaussian kernel computationally more expensive and, if the additional parameter τ is poorly estimated, more error prone. The effect of the hyperparameters on errors is shown in **Supplementary Figure 3**.

While the naïve kernel takes all the information from the HMM into account by using all parameters from a subject-specific version of the model, it uses these parameters in a way that ignores the structure of the model that these parameters come from. In this way, the different parameters in the feature vector are difficult to compare, since e.g., a change of 0.1 in the transition probabilities between two states is not of the same magnitude as a change of 0.1 in one entry of the covariance matrix of a specific state. In the naïve kernel, these two very different types of changes would be treated indistinctly.

2.3.2 Naïve normalised kernel

To address the problem of parameters being on different scales, the naïve normalised kernel makes the scale of the subject-specific vectorised parameters (i.e., the naïve θ) comparable across parameters. Here, the mapping $x \rightarrow \phi_x$ consists of a vectorisation and normalisation across subjects of the subject-specific HMM parameters, by subtracting the mean over

subjects from each parameter and dividing by the standard deviation. This kernel does not respect the geometry of the space of parameters either.

As for the naïve kernel, we can then construct a linear kernel from these vectorised, normalised parameters $\tilde{\theta}$ by computing the inner product for all pairs of subjects n and m to obtain the linear naïve normalised kernel κ_{NNl}:

$$\kappa_{NNl}(n,m) = \langle \tilde{\theta}^n, \tilde{\theta}^m \rangle \qquad \kappa_{NNl} \in \mathbb{R}^{NxN}$$

We can also compute a Gaussian kernel from the naïve normalised feature vectors to obtain the Gaussian version of the naïve normalised kernel κ_{NNg}:

$$\kappa_{NNg}(n,m) = \exp\left(-\frac{\|\tilde{\theta}^n - \tilde{\theta}^m\|^2}{2\tau^2}\right) \qquad \kappa_{NNg} \in \mathbb{R}^{NxN}$$

In this way, we have constructed a kernel in which parameters are all on the same scale, but which still ignores the complex relationships between parameters originally encoded by the underlying model of brain dynamics.

2.3.3 Fisher kernel

The Fisher kernel (Jaakkola et al., 1999; Jaakkola & Haussler, 1998) is specifically designed to preserve the structure of a generative probabilistic model (here, the HMM). This can be thought of as a "proper" projection on the manifold, as illustrated in **Figure 1b**, step 4b. Similarity between subjects is here defined in reference to a group-level model of brain dynamics. The mapping $x \rightarrow \phi_x$ is given by the "Fisher score", which indicates how (i.e., in which direction in the Riemannian parameter space) we would have to change the group-level model to better explain a particular subject's timeseries. The similarity between subjects can then be described based on this score, so that two subjects are defined as similar if the group-level model would have to be changed in a similar direction for both, and dissimilar otherwise.

More precisely, the Fisher score is given by the gradient of the log-likelihood with respect to each model parameter:

$$g(\theta^0, x^n) = \left(\frac{\partial \log \mathcal{L}_{\theta^0}(x^n)}{\partial \theta^0}\right) \qquad g \in \mathbb{R}^{1x(K+K*K+K*M+K*M*M)}$$

where x^n is the timeseries of subject n, and $\mathcal{L}_\theta(x) = P(x|\theta)$ represents the likelihood of the timeseries x given the model parameters θ. This way, the Fisher score maps an example (i.e., a subject's timeseries) x^n into a point in the gradient space of the Riemannian manifold R_θ defined by the HMM parameters.

The invariant Fisher kernel κ_{F-} is the inner product of the Fisher score g, scaled by the Fisher information matrix F, which gives a local metric on the Riemannian manifold R_θ:

$$\kappa_{F-}(n,m) = g(\theta^0, x^n)^{\mathrm{T}} F_{R_\theta}^{-1} g(\theta^0, x^m) \qquad \kappa_{F-} \in \mathbb{R}^{NxN}$$

for subjects n and m. F_{R_θ} is the Fisher information matrix, defined as

$$F_{R_\theta} = \mathbb{E}_x[g(\theta^0, x) g(\theta^0, x)^{\mathrm{T}}]$$

where the expectation is with respect to x under the distribution $P(x|\theta)$. The Fisher information matrix F_{R_θ} can be approximated empirically:

$$\hat{F}_{R_\theta} = \frac{1}{N}\sum_{i=1}^{N} g\left(\theta^0, x^i\right) g\left(\theta^0, x^i\right)^{\mathrm{T}}$$

which is simply the covariance matrix of the gradients g. Using $\hat{F}_{R_\theta}$ essentially serves to whiten the gradients; therefore, given the large computational cost associated with $\hat{F}_{R_\theta}$, we here disregard the Fisher information matrix and reduce the invariant Fisher kernel to the so-called practical Fisher kernel (Jaakkola et al., 1999; Jaakkola & Haussler, 1998; van der Maaten, 2011), for which the linear version κ_{Fl} takes the form:

$$\kappa_{Fl}(n,m) = \langle g(\theta^0, x^n), g(\theta^0, x^m)\rangle \qquad \kappa_{Fl} \in \mathbb{R}^{NxN}$$

In this study, we will use the practical Fisher kernel for all computations.

One issue when working with the linear Fisher kernel is that the gradients of typical examples (i.e., subjects whose timeseries can be described by similar parameters as the group-level model) are close to zero, while gradients of atypical examples (i.e., subjects who are very different from the group-level model) can be very large. This may lead to an underestimation of the similarity between two typical examples because their inner product is very small, although they are very similar. To mitigate this, we can plug the gradient features (i.e., the Fisher scores g) into a Gaussian kernel, which essentially normalises the kernel. For subjects n and m, where x^n is the timeseries of subject n and x^m is the timeseries of subject m, the Gaussian Fisher kernel κ_{Fg} is defined as

$$\kappa_{Fg}(n,m) = \exp\left(-\frac{\|g(\theta^0, x^n) - g(\theta^0, x^m)\|^2}{2\tau^2}\right) \quad \kappa_{Fg} \in \mathbb{R}^{NxN}$$

where $\|g(\theta^0, x^n) - g(\theta^0, x^m)\|$ is the distance between examples n and m in the gradient space, and τ is the width of the Gaussian kernel.

2.3.4 Kullback-Leibler divergence

Kullback-Leibler (KL) divergence is an information-theoretic distance measure which estimates divergence between probability distributions —in this case between subject-specific versions of the HMM. Here, KL divergence of subject n from subject m, $\mathrm{KL}(\mathrm{HMM}^n||\mathrm{HMM}^m)$ can be interpreted as how much new information the HMM of subject n contains if the true distribution was the HMM of subject m. KL divergence is not symmetric, i.e., $\mathrm{KL}(\mathrm{HMM}^n||\mathrm{HMM}^m)$ is different than $\mathrm{KL}(\mathrm{HMM}^m||\mathrm{HMM}^n)$. We use an approximation of KL divergence which is described in detail in Do (2003), as in Vidaurre et al. (2021). That is, given two models HMM^n from HMM^m for subject n and subject m, we have

$$\mathrm{KL}(\mathrm{HMM}^n||\mathrm{HMM}^m) = \sum_k v_k KL(A_k^n, A_k^m) + v_k KL(G_k^n, G_k^m)$$

Where A_k^n are the transition probabilities from state k into any other state according to HMM^n and G_k^n are the state Gaussian distributions for state k and HMM^n (respectively A_k^m and G_k^m for HMM^m) (see MacKay et al. (2003)). Since the transition probabilities are Dirichlet-distributed and the state distributions are Gaussian distributed, KL divergence for those has a closed-form solution. Variables v_k can be computed numerically such that

$$vA^n = v,$$

$$\lim_{x\to\infty} \pi^n A^{n^x} = v$$

To be able to use KL divergence as a kernel, we symmetrise the KL divergence matrix as

$$\mathcal{D}_{KL}(n,m) = 0.5\ \mathrm{KL}(\mathrm{HMM}^n||\mathrm{HMM}^m) + 0.5\ \mathrm{KL}(\mathrm{HMM}^m||\mathrm{HMM}^n) \qquad \mathcal{D}_{KL} \in \mathbb{R}^{NxN}$$

This symmetrised KL divergence can be plugged into a radial basis function, analogous to the Gaussian kernels to obtain a similarity matrix κ_{KL}

$$\kappa_{KL}(n,m) = \exp\left(-\frac{(\mathcal{D}_{KL}(n,m))^2}{2\tau^2}\right) \qquad \kappa_{KL} \in \mathbb{R}^{NxN}$$

The resulting KL similarity matrix can be used in the predictive model in a similar way as the kernels described above.

2.4 Predictive model: Kernel ridge regression

Similarly to Vidaurre et al. (2021), we use kernel ridge regression (KRR) to predict demographic and behavioural variables from the different kernels, although these could be used in any kernel-based prediction model or classifier, such as a support vector machine,

which we demonstrate in simulations. KRR is the kernelised version of ridge regression (Saunders et al., 1998):

$$\hat{y} = h\alpha \qquad \alpha \in \mathbb{R}^{S_{train}x1}, h \in \mathbb{R}^{S_{test}xS_{train}}$$

where α are the regression weights; h is the (number of subjects in test set by number of subjects in training set) kernel matrix between the subjects in the training set and the subjects in the test set; $\hat{y}$ are the predictions in the (out-of-sample) test set; S_{train} are the number of subjects in the training set; and S_{test} are the number of subjects in the test set. The regression weights α can be estimated using the kernels specified above as

$$\alpha = (\kappa + \lambda I)^{-1} * y \qquad \kappa \in \mathbb{R}^{S_{train}xS_{train}}$$

where λ is a regularisation parameter that we can choose through cross-validation; I is the identity matrix; κ is the (S_{train} by S_{train}) kernel matrix of the subjects in the training set; and y are the training examples.

We use KRR to separately predict each of the 35 demographic and behavioural variables from each of the different methods, removing subjects with missing entries from the prediction. The hyperparameters λ (and τ in the case of Gaussian kernels) were selected in a nested cross-validated (CV) fashion using 10 folds for both inner and outer loops assigning 90% of the data to the training set and 10% to the test set for each fold. We used a CV scheme to account for family structure in the HCP dataset so that subjects from the same family were never split across folds (Winkler et al., 2015). We repeated the nested 10-fold CV 100 times, so that different combinations of subjects were randomly assigned to the folds at each new CV iteration to obtain a distribution of model performance values for each variable. This is to explicitly show how susceptible each model was to changes in the training folds, which we can take as a measure of the robustness of the estimators, as described below.

2.4.1 Evaluation criteria

We evaluate the models in terms of two outcome criteria: prediction accuracy and reliability. The first criterion, prediction accuracy, concerns how close the model's predictions were to the true values on average. We here quantify prediction accuracy as Pearson's correlation coefficient $r(\hat{y}, y)$ between the model-predicted values $\hat{y}$ and the actual values y of each variable.

The second criterion, reliability, concerns two aspects: i) that the model will never show excessively large errors for single subjects that could harm interpretation; and ii) that the model's accuracy will be consistent across random variations of the training set —in this

case by using different (random) partitions for the CV folds. As mentioned, this is important if we want to interpret prediction errors for example in clinical contexts, which assumes that the error size of a model in a specific subject reflects something biologically meaningful, e.g., whether a certain disease causes the brain to “look” older to a model than the actual age of the subject (Denissen et al., 2022). Maximum errors inform us about single cases where a model (that may typically perform well) fails. The maximum absolute error (MAXAE) is the single largest error made by the kernel ridge regression model in each iteration, i.e.,

$$MAXAE = \max_{i \in N}(|y_i - \hat{y}_i|)$$

Since the traits we predict are on different scales, the MAXAE is difficult to interpret, e.g., a MAXAE of 10 would be considered small if the true range of the variable we are predicting was 1,000, while it would be considered large if the true range of the variable was 1. To make the results comparable across the different traits, we therefore normalise the MAXAE by dividing it by the range of the respective variable. In this way, we obtain the NMAXAE:

$$NMAXAE = \frac{MAXAE}{y_{max} - y_{min}}$$

Since the NMAXAEs follow extreme value distributions, it is more meaningful to consider the proportion of the values exceeding relevant thresholds than testing for differences in the means of these distributions (Gumbel, 1958). We here consider the risk of large errors (NMAXAE > 10), very large errors (NMAXAE > 100), and extreme errors (NMAXAE > 1,000) as the percentage of runs (across variables and CV iterations) where the model’s NMAXAE exceeds the given threshold. Since NMAXAE is normalised by the range of the actual variable, these thresholds correspond to one, two, and three orders of magnitude of the actual variable’s range. If we are predicting age, for instance, and the true ages of the subjects range from 25 years to 35 years, an NMAXAE of 1 would mean that the model’s least accurate prediction is off by 10 years, an NMAXAE of 10 would mean that the least accurate prediction is off by 100 years, an NMAXAE of 100 would be off by 1,000 years, and an NMAXAE of 1,000 would be off by 10,000 years. A model that makes such large errors, even in single cases, would be unusable for interpretation. Our reliability criterion in terms of maximum errors is therefore that the risk of large errors (NMAXAE > 10) should be 0%.

For a model to be reliable, it should also be robust in the sense of susceptibility to changes in the training examples. Robustness is an important consideration in prediction studies, as it determines how reproducible a study or method is —which is often a shortcoming in neuroimaging-based prediction studies (Varoquaux et al., 2017). We evaluated the robustness by iterating nested 10-fold CV 100 times for each variable, randomising the

subjects in the folds at each iteration, so that the models would encounter a different combination of subjects in the training and test sets each time they are run. Looking at this range of 100 predictions for each variable, we can assess whether the model's performance changes drastically depending on which combinations of subjects it encountered in the training phase, or whether the performance is the same regardless of the subjects encountered in the training phase. The former would be an example of a model that is susceptible to changes in training examples and the latter an example of a model that is robust. We here quantify robustness as the standard deviation (S.D.) of the prediction accuracy *r* across the 100 CV iterations of each variable, where a small mean S.D. over variables indicates higher robustness.

We test for significant differences in mean prediction accuracy and robustness between the methods using permutation t-tests with 20,000 permutations across variables and iterations. For maximum errors, we test whether NMAXAE distributions are significantly different between the methods using permutation-based Kolmogorov-Smirnov tests. P-Values are corrected across multiple comparisons using Bonferroni-correction to control the family-wise error rate (FWER). To compare the Fisher kernel to the naïve and the naïve normalised kernels, we first evaluate the linear and Gaussian versions of the kernels together. We then compare the linear and Gaussian versions of each kernel with each other. We also compare each kernel to the KL divergence model. Finally, we compare the linear Fisher kernel with the methods based on time-averaged FC features described in the following section.

2.5 Regression models based on time-averaged FC features

To compare our approach's performance to simpler methods that do not take dynamics into account, we compared them to four different regression models based on time-averaged FC features. For each subject, time-averaged FC was computed as z-transformed Pearson's correlation between each pair of regions. The first time-averaged model is, analogous to the HMM-derived KL divergence model, a time-averaged KL divergence model. The model is described in detail in Vidaurre et al. (2021). Briefly, we construct symmetrised KL divergence matrices of each subject's time-averaged FC and predict from these matrices using the KRR pipeline described above. We refer to this model as time-averaged KL divergence. The other three time-averaged FC benchmark models do not involve kernels but predict directly from the features instead. Namely, we use two variants of an Elastic Net model (Zou & Hastie, 2005), one using the unwrapped time-averaged FC matrices as input, and one using the time-averaged FC matrices in Riemannian space (Barachant et al., 2013; Smith, Vidaurre, et al., 2013), which we refer to as Elastic Net and Elastic Net (Riem.), respectively. Finally, we compare our models to the approach taken in Rosenberg et al. (2016), where relevant edges

of the time-averaged FC matrices are first selected, and then used as predictors in a regression model. We refer to this model as Selected Edges. All time-averaged FC models are fitted using the (nested) cross-validation strategy, accounting for family structure in the dataset, and repeated 100 times with randomised folds, as described above.

2.6 Simulations

To further understand the behaviour of the different kernels, we simulate data and compare the kernels' ability to recover the ground truth. Specifically, we aim to understand which type of parameter change the kernels are most sensitive to. We generate timeseries for two groups of subjects, timeseries X^1 for group 1 and timeseries X^2 for group 2, from two separate HMMs with respective sets of parameters θ^1 and θ^2:

$$X^1 \sim HMM(\theta^1)$$

$$X^2 \sim HMM(\theta^2)$$

We simulate timeseries from HMMs with these parameters through the generative process described in 2.2.

For the simulations, we use the group-level HMM of the real dataset with $K = 6$ states used in the main text as basis for group 1, i.e., $HMM(\theta^1) = HMM(\theta^0)$. We then manipulate two different types of parameters, the state means μ and the transition probabilities A, while keeping all remaining parameters the same between the groups. In the first case, we manipulate one state's mean between the groups, i.e.:

$$\theta^1 = [\pi^1, A^1, \mu^1, \sigma^1]$$

$$\theta^2 = [\pi^1, A^1, \mu^2, \sigma^1]$$

where μ^2 is obtained by simply adding a Gaussian noise vector to the state mean vector of one state:

$$\mu_1^2 = \mu_1^1 + \varphi$$

Here, φ is the Gaussian noise vector of size 1 x M, M is the number of parcels, here 50, and μ_1^1 and μ_1^2 are the first rows (corresponding to the first state) of the state mean matrices for groups 1 and 2, respectively. We control the amplitude of φ so that the difference between μ_1^2 and μ_1^1 is smaller than the minimum distance between any pair of states within one HMM. This is to ensure that the HMM recovers the difference between groups as difference in one state's mean vector, rather than detecting a new state for group 2 that does not occur in group 1 and consequently collapsing two other states. Since the state means μ and the state

covariances σ are the first- and second-order parameters of the same respective distributions, it is to be expected that, although we only directly manipulate μ, σ changes as well.

In the second case, we manipulate the transition probabilities for one state between the groups, while keeping all other parameters the same, i.e.:

$$\theta^1 = [\pi^1, A^1, \mu^1, \sigma^1]$$

$$\theta^2 = [\pi^1, A^2, \mu^1, \sigma^1]$$

where A^2 is obtained by randomly permuting the probabilities of one state to transition into any of the other states, excluding self-transition probability:

$$A^2_{1,1} = A^1_{1,1}$$

$$A^2_{1,2:K} = pA^1_{1,2:K}$$

Here, p is a random permutation vector of size (K-1) x 1, $A^1_{1,1}$ and $A^2_{1,1}$ are the self-transition probabilities of state 1, and $A^1_{1,2:K}$ and $A^2_{1,2:K}$ are the probabilities of state 1 to transition into each of the other states.

We then concatenate the generated timeseries

$$X = [X^1, X^2]$$

containing 100 subjects for group 1 and 100 subjects for group 2, with 1,200 timepoints and 50 parcels per subject. Note that we do not introduce any differences between subjects within a group, so that the between-group difference in HMM parameters should be the most dominant distinction and easily recoverable by the classifiers. We then apply the pipelines described above, running a new group-level HMM and constructing the linear versions of the naïve kernel, the naïve normalised kernel, and the Fisher kernel on these synthetic timeseries. The second case of simulations, manipulating the transition probabilities, only introduces a difference in few (K - 1) features and keeps the majority of the features the same between the groups, while the first case introduces a difference in a large number of features (M features directly, by changing one state's mean vector, and an additional M x M features indirectly, as this state's covariance matrix will also be affected). To account for this difference, we additionally construct a version of the kernels for the second case of simulations that includes only π and A, removing the state parameters μ and σ. Finally, we use a support vector machine (SVM) in combination with the different kernels to recover the group labels and measure the error produced by each kernel. We repeat the whole process (generating timeseries, constructing kernels, and running the SVM) 10 times, in each

iteration randomising on 3 levels: generating new random noise/permutation vectors to simulate the timeseries, randomly initialising the HMM parameters when fitting the group-level HMM to the simulated timeseries, and randomly assigning subjects to 10 CV folds in the SVM.

2.7 Implementation

The code used in this study was implemented in Matlab and is publicly available within the repository of the HMM-MAR toolbox at https://github.com/OHBA-analysis/HMM-MAR. A Python-version of the Fisher kernel is also available at https://github.com/vidaurre/glhmm. Code for Elastic Net prediction from time-averaged FC features is available at https://github.com/vidaurre/NetsPredict/blob/master/nets_predict5.m. The procedure for Selected Edges time-averaged FC prediction is described in detail in Shen et al. (2017) and code is provided at https://www.nitrc.org/projects/bioimagesuite/behavioralprediction.m. All code used in this paper, including scripts to reproduce the figures and additional application examples of the Fisher Kernel can be found in the repository https://github.com/ahrends/FisherKernel.

3 Results

We compared different prediction methods derived from a model of brain dynamics with each other, and with models that do not account for dynamics. We assess the performance of the predictive models in terms of two outcome criteria: prediction accuracy and reliability. We show the superiority in both accuracy and reliability of the Fisher kernel, a principled approach for prediction using brain dynamics, because of its capacity to preserve the underlying structure of the brain dynamics model.

3.1 The Fisher kernel predicts more accurately than other methods

We found that the Fisher kernel has the highest prediction accuracy on average across the range of variables and CV iterations, as shown in **Figure 2a**. Specifically, the Fisher kernel has a significantly higher correlation coefficient between model-predicted and actual values than the naïve kernel (mean r κ_F: 0.196 vs. mean r κ_N: 0.127, p=0.0008) and the naïve normalised kernel (mean r κ_{NN}: 0.159, p=0.0008). Comparing the two naïve kernels, the normalised version significantly improved upon the non-normalised naïve kernel (κ_N vs. κ_{NN}: p=0.0008).

We next compared the linear with the Gaussian versions of each kernel. The linear versions performed significantly better than the Gaussian versions for the Fisher kernel (mean *r* κ_{Fl}: 0.204 vs. mean *r* κ_{Fg}: 0.187, *p*=0.0008) and the naïve normalised kernel (mean *r* κ_{NNl} 0.187 vs. mean *r* κ_{NNg}: 0.131, *p*=0.0008). The opposite was the case for the naïve kernel, where the Gaussian version significantly outperformed the linear version (mean *r* κ_{Nl}: 0.103 vs. mean *r* κ_{Ng}: 0.150, *p*=0.0008).

We next compared all models to the KL divergence model. The linear and the Gaussian Fisher kernels and the linear version of the naïve normalised kernel significantly outperformed the KL divergence model (mean *r* κ_{KL}:0.154 vs. κ_{Fl}: *p*=0.0008; vs. κ_{Fg}: *p*=0.0008; vs. κ_{NNl}: *p*=0.0008). The Gaussian version of the naïve normalised kernel performed significantly worse than the KL divergence model (*p*=0.0008). Both versions of the naïve kernel were less accurate than the KL divergence model in terms of correlation between model-predicted and actual values: This comparison was significant for the linear version of the naïve kernel (*p*=0.0008), but not for the Gaussian version (*p*=0.856).

The models based on time-averaged features had significantly lower accuracies than the linear Fisher kernel (κ_{Fl} vs. time-averaged KL div.: *p*=0.0008; vs. Elastic Net: *p*=0.0008; vs. Elastic Net Riem.: *p*=0.0008; vs. Selected Edges: *p*=0.0008), as shown in **Figure 2a**. We observed that the non-kernel-based time-averaged FC models, i.e., the two Elastic Net models and the Selected Edges model, make predictions at a smaller range than the actual variables, close to the variables' means. This leads to weak relationships between predicted and actual variables but smaller errors, as shown in **Supplementary Figure 3**. The distributions of correlation coefficients for the different methods are shown in **Figure 2a**. The performance of all methods is also summarised in **Supplementary Table 2**.

We found that the Fisher kernel also predicted more accurately than other kernels when fitting the HMM to only the first resting-state session of each subject, as shown in **Supplementary Figure 1**. However, the overall accuracies of all kernels were lower in this case, indicating that predicting traits benefits from a large amount of available data per subject. Similarly, the Fisher kernel outperformed the other kernels when HMM states were defined only in terms of covariance (not mean) at comparable accuracy, as shown in **Supplementary Figure 2**.

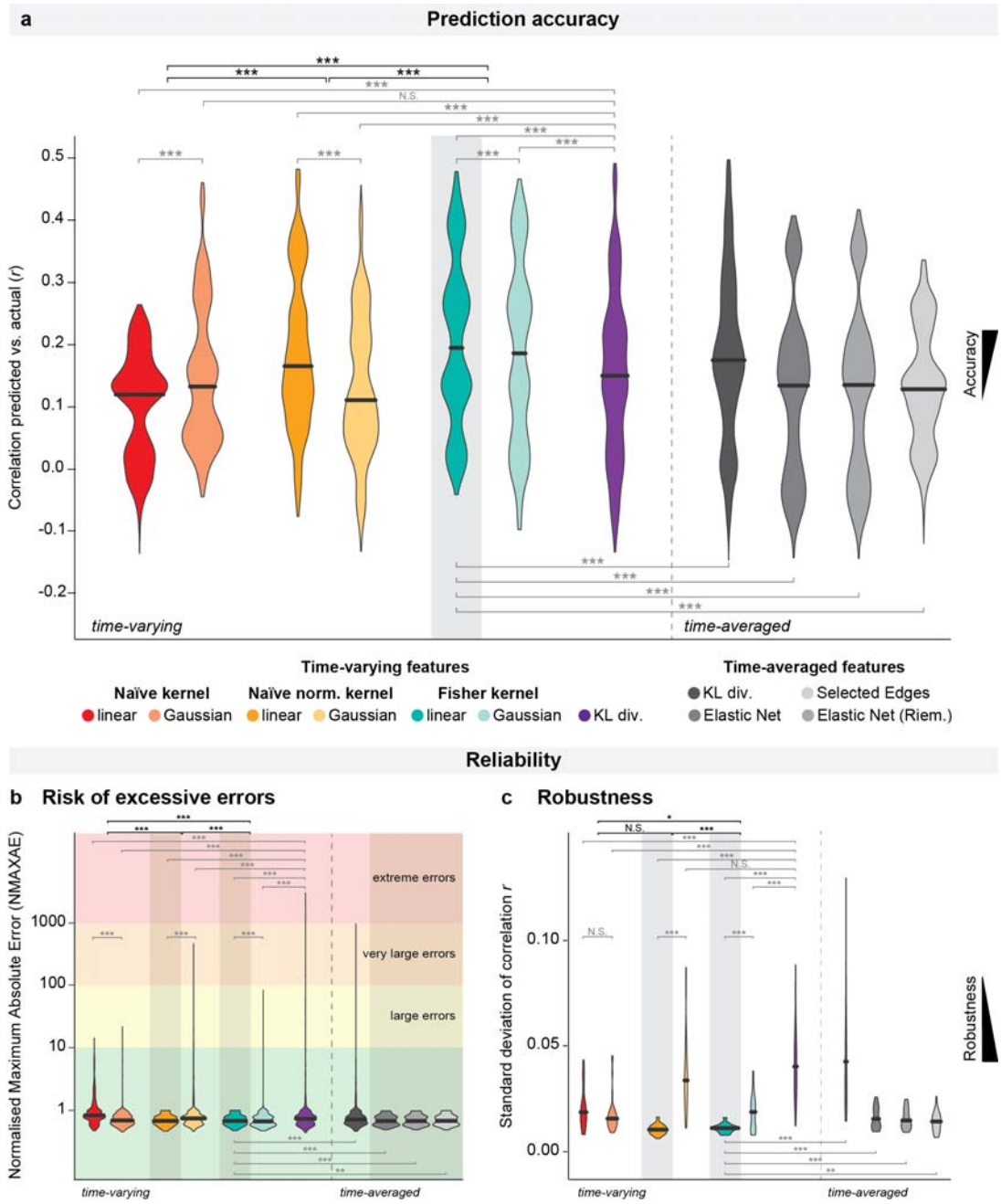

Figure 2 *Distributions of model performance by kernel. The best-performing methods are highlighted by grey columns in each plot.* ***a)*** *Pearson's correlation coefficients (r) between predicted and actual variable values as a measure of prediction accuracy (y-axis) of each method (x-axis). Larger values indicate that the model predicts more accurately. The linear Fisher kernel has the highest average accuracy.* ***b)*** *Normalised maximum errors (NMAXAE) as a measure of excessive errors (y-axis) by kernel (x-axis). Large maximum errors indicate that the model predicts very poorly in single cases. Differences between the kernels mainly lie in the tails of the distributions, where the KL divergence model produces extreme maximum errors in some runs (NMAXAE > 1,000), while the linear naïve normalised kernel and the linear Fisher kernel have the smallest risk of excessive errors (NMAXAE < 10). The y-axis is plotted on the log-scale. Asterisks here indicate significant results of Kolmogorov-Smirnov permutation tests.* ***c)*** *Robustness of correlation between model-predicted and actual values. The plot shows the distribution across variables of the standard deviation of correlation coefficients over CV iterations on the y-axis for each kernel (on the x-axis). Smaller values indicate greater robustness. The linear naïve normalised kernel and the linear Fisher kernel are most robust. a),b) Each violin plot shows the distribution over 3,500 runs (100 iterations of 10-fold CV for all 35 variables) that were predicted from each kernel. c) Each*

*violin plot shows the distribution over 35 variables that were predicted from each kernel. Asterisks indicate significant Bonferroni-corrected p-values using 20,000 permutations: *: p < 0.05, **: p < 0.01, ***: p < 0.001.*

As shown in **Figure 3**, there are differences in how well these demographic or behavioural variables *can* be predicted from a model of brain dynamics. While there are some variables which may show a strong link to brain dynamics, other variables may be more related to static or structural measures (Liégeois et al., 2019; Vidaurre et al., 2021), or just be difficult to predict in general. **Table 1** shows the variables that can be predicted at relatively high accuracy by at least one of the time-varying methods. We here consider a mean Pearson's *r* across CV iterations greater than 0.3 as "relatively high" accuracy. Among these, five variables (variables ReadEng_AgeAdj, PicVocab_Unadj, PicVocab_AgeAdj, VSPLOT_OFF, and WM_Task_Acc) are best predicted by the Fisher kernel, one variable is best predicted by the naïve normalised kernel (variable VSPLOT_TC), and two variables are best predicted by the time-averaged KL divergence model (variables Age and ReadEng_Unadj). The linear naïve kernel and the time-averaged Elastic Net models produce the least accurate predictions in these variables. Examples of best-predicted variables for all HMM-derived kernels can be found in **Supplementary Figure 6**.

Table 1 *Variables predicted at relatively high accuracy by at least one method. The table shows mean correlation coefficients for all methods in those variables that could be predicted at relatively high accuracy (correlation coefficient > 0.3) by at least one method. The best performing model is highlighted in bold.*

Method		**Mean correlation coefficient *r***							
		Age	**ReadEng _Unadj**	**ReadEng _AgeAdj**	**PicVocab _AgeAdj**	**PicVocab _Unadj**	**WM_Task _Acc**	**VSPLOT _TC**	**VSPLOT _OFF**
Naïve	Linear	0.06	0.193	0.2	0.189	0.131	0.234	0.227	0.223
	Gaussian	0.43	0.329	0.318	0.33	0.288	0.254	0.282	0.277
Naïve norm.	Linear	0.47	0.383	0.38	0.372	0.346	0.341	**0.32**	0.342
	Gaussian	0.37	0.267	0.252	0.245	0.242	0.269	0.228	0.296
Fisher	Linear	0.46	0.399	**0.401**	**0.42**	**0.392**	**0.375**	0.305	**0.362**
	Gaussian	0.44	0.396	0.4	0.419	0.388	0.37	0.295	0.352
KL div.		0.47	0.357	0.351	0.338	0.314	0.283	0.248	0.235
ta KL div.		**0.47**	**0.418**	0.378	0.265	0.297	0.298	0.246	0.245
Elastic Net		0.39	0.359	0.348	0.347	0.355	0.241	0.17	0.189
Elastic Net (Riem.)		0.39	0.358	0.348	0.347	0.353	0.241	0.173	0.188
Selected Edges		0.27	0.264	0.271	0.32	0.268	0.223	0.24	0.248

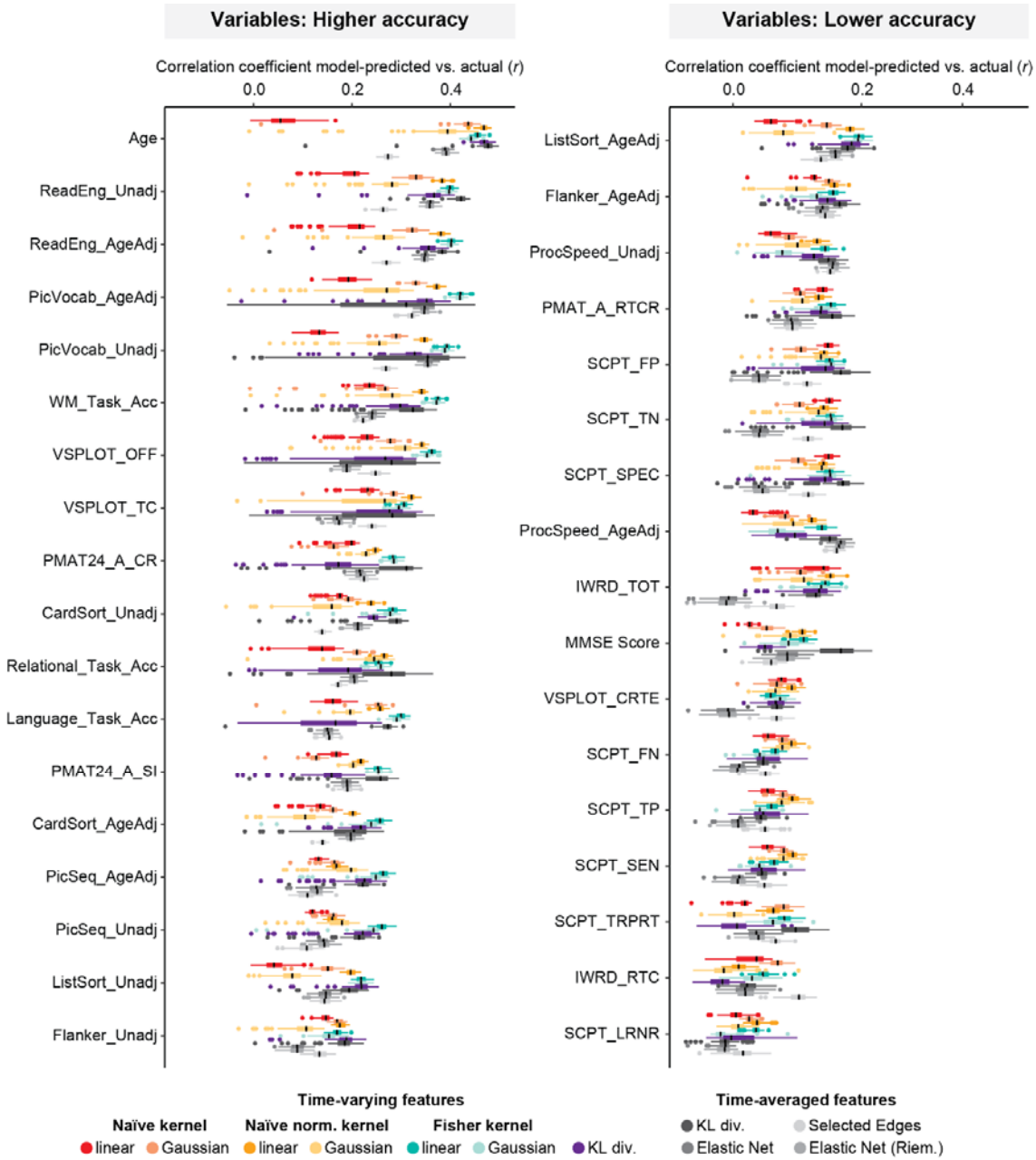

Figure 3 *Model performance estimates over CV iterations by behavioural variable and method. Boxplots show the distribution over 100 iterations of 10-fold CV of correlation coefficient values (x-axis) of each method, separately for each of the 35 predicted variables (y-axes). The Fisher kernel (green) not only predicts at higher accuracy for many variables, but also shows the narrowest range, indicating high robustness. Black lines within each boxplot represent the median.*

In summary, the linear Fisher kernel has the highest prediction accuracy, significantly outperforming the naïve kernels, Gaussian version of the Fisher kernel, and the KL divergence model. The linear Fisher kernel also has a higher prediction accuracy than benchmark methods using time-averaged FC features. The linear Fisher kernel, closely

followed by its Gaussian version, is the kernel that produces the best predictions in the traits that can be predicted well by any one kernel.

3.2 The Fisher kernel has a lower risk of excessive errors and is more robust than other methods

We now show empirically that the Fisher kernel is more reliable than other kernels, both in terms of risk of large errors and in terms of robustness over CV iterations.

The linear versions of the Fisher kernel and the naïve normalised kernel have the overall lowest risk of large errors, as shown in **Figure 2b**. We assessed the risk of large errors (NMAXAE > 10), very large errors (NMAXAE > 100), and extreme errors (NMAXAE > 1,000), corresponding to one, two, and three orders of magnitude of the range of the actual variable. For the Fisher kernel, the risk of large errors is low: 0.057% in the Gaussian version κ_{Fg} and 0% in the linear version κ_{Fl}. That means that the linear Fisher kernel never makes large errors exceeding the range of the actual variable by orders of magnitude. In the naïve kernel, both the linear κ_{Nl} and the Gaussian version κ_{Ng} have a low risk of large errors at 0.029% for the linear version and 0.057% for the Gaussian version. The Gaussian naïve normalised kernel κ_{NNg} on the other hand has a slightly higher risk of large errors at 1.140% and a risk of very large errors at 0.057%, while its linear version κ_{NNl} has a 0% risk of large errors. The risk of large errors was highest in the KL divergence model κ_{KL} at 0.890% risk of large errors, 0.110% risk of very large errors, and even 0.057% risk of extreme errors. While the time-averaged KL divergence model had similarly large maximum errors as the time-varying KL divergence model, the three other time-averaged models had no risk of excessive errors, similar to the linear Fisher kernel. The maximum error distributions are shown in **Figure 2b**. The summary of maximum errors is also reported in **Supplementary Table 2** and **Supplementary Table 3**. **Supplementary Figure 4** also shows the quantile-quantile plots of the normalised maximum errors.

A reason for the higher risk of large errors in the Gaussian versions of the kernels is likely that the radius τ of the radial basis function needs to be selected (using cross-validation), introducing an additional factor of variability and leaving more room for error. **Supplementary Figure 5** shows the relation between the estimated hyperparameters (the regularisation parameter λ and the radius τ of the radial basis function) and how large errors in the predictions may be related to poor estimation of these parameters.

With respect to robustness, we found that the linear Fisher kernel κ_{Fl} and the linear naïve normalised kernel κ_{NNl} had the most robust performance on average across the range of variables tested, as shown in **Figure 2c**. Robustness was quantified as the standard deviation of the correlation between model-predicted and actual values over 100 iterations of 10-fold CV. A low standard deviation indicates high robustness, since the method's performance does not differ greatly depending on the specific subjects it was trained and tested on. The Fisher kernel was significantly more robust than both the naïve kernel and the naïve normalised kernel (κ_F mean S.D. *r*: 0.015 vs. naïve kernel κ_N mean S.D. *r*: 0.019, *p*=0.02; vs. naïve normalised kernel κ_{NN} mean S.D. *r*: 0.024, *p*=0.0008). The naïve kernel and the naïve normalised kernel did not significantly differ from each other (*p*=0.49). We found that the linear versions of the Fisher kernel and the naïve normalised kernel were significantly more robust than their respective Gaussian versions (κ_{Fl} mean S.D. *r*: 0.011 vs. κ_{Fg} mean S.D. *r*: 0.018, *p*=0.0008; κ_{NNl} mean S.D. *r*: 0.011 vs. κ_{NNg} mean S.D. *r*: 0.036, *p*=0.0008), while there was no significant difference between the linear and the Gaussian naïve kernel (κ_{Nl} S.D. *r*: 0.020 vs. κ_{Ng} S.D. *r*: 0.018, *p*>1). All but the Gaussian naïve normalised kernel κ_{NNg} significantly outperformed the KL divergence model κ_{KL} in terms of robustness (κ_{KL} mean S.D. *r*: 0.042 vs. κ_{Nl} *p*=0.0008; vs. κ_{Ng} *p*=0.0008; vs. κ_{NNl} *p*=0.0008; vs. κ_{NNg} *p*>1; vs. κ_{Fl} *p*=0.0008; vs. κ_{Fg} *p*=0.0008). This large variation in model performance depending on the CV fold structure in the KL divergence model is problematic. The same problem occurs, even more drastically, when using time-averaged FC features in a KL divergence model, indicating robustness issues of the KL divergence measure in this type of predictive model independent of the underlying features. Like the time-varying KL divergence model, the time-averaged KL divergence model was significantly less robust than the linear Fisher kernel (*p*=0.0008). Also the other time-averaged models were significantly less robust than the linear Fisher kernel (κ_{Fl} vs. Elastic Net: *p*=0.0008; vs. Elastic Net Riem.: *p*=0.0008; vs. Selected Edges: *p*=0.007). The ranges in model performance across CV iterations for each variable of the different kernels are shown in **Figure 3**. Robustness of all kernels is also summarised in **Supplementary Table 3**.

Overall, the linear Fisher kernel and the linear naïve normalised kernel were the most reliable: For both kernels, the risk of large errors was zero and the variability over CV iterations was the smallest. The Gaussian versions of all kernels had higher risks of large errors and a larger range in model performance over CV iterations, indicating that their performance was less reliable. The KL divergence model was the most problematic in terms of reliability with a risk of extreme errors ranging up to three orders of magnitude of the actual variable's range and considerable susceptibility to changes in CV folds.

3.3 Fisher kernel predictions are driven by individual differences in state features

The Fisher kernel is most sensitive to individual differences in state parameters, both in simulated timeseries and in the real data. That means that it is more relevant for the predictions what states look like in the individuals than how the individuals transition between states. However, the Fisher kernel can be modified to recover differences in transition probabilities if these are relevant for specific traits.

As shown in **Figure 4a**, panel 1, when we simulated two groups of subjects that are different in terms of the mean amplitude of one state, the Fisher kernel was able to recover this difference in all runs with 0% error, meaning that it identified all subjects correctly in all runs. The Fisher kernel significantly outperformed the other two kernels (Fisher kernel κ_{Fl} vs. naïve kernel κ_{Nl}: p=0.0003, vs. naïve normalised kernel κ_{NNl}: p=0.0001), while there was no significant difference between the naïve and the naïve normalised kernel (p>1). Both the naïve kernel and the naïve normalised kernel (**Figure 4a**, panels 2, 3) do not show a group-difference (i.e., there is no clear dissimilarity between the first and the second half of subjects) and are prone to producing outliers. In the Fisher kernel, there are no outliers in the features or kernels, and the group difference is obvious with a strong checkerboard pattern in the kernels (**Figure 4a**, panel 4).

When we simulated differences in transition probabilities between two groups, neither of the kernels were able to reliably recover this difference and the Fisher kernel performed significantly worse than the other two kernels on average (compared to naïve kernel: p=0.006, compared to naïve normalised kernel: p=0.004), as shown in **Figure 4b**, panel 1. Like the previous case, where we simulated differences in state means, the naïve kernel and the naïve normalised kernel did not significantly differ from each other (p>1), and they produced outliers in features and kernels (**Figure 4b**, panels 2-3).

This indicates that all kernels, but particularly the Fisher kernel, are most sensitive to differences in state parameters rather than differences in transition probabilities. To understand whether the difference in transition probabilities can be recovered when it is not overshadowed by the more dominant state parameters, we run the second case of simulations again, where we introduce a group difference in terms of transition probabilities, but this time we exclude the state parameters when we construct the kernels. As shown in **Figure 4c**, panel 1, the Fisher kernel is now able to recover the group difference with minimal errors, while the naïve normalised kernel improves but does not perform as well as the Fisher kernel, and the naïve kernel performs below chance. Here, the Fisher kernel

significantly outperformed both the naïve kernel (p=0.0001) and the naïve normalised kernel (p=0.02), and the naïve normalised kernel was significantly more accurate than the naïve kernel (p=0.0004). This shows that the Fisher kernel is able to recover the group difference in transition probabilities, if this difference is not overshadowed by the state parameters.

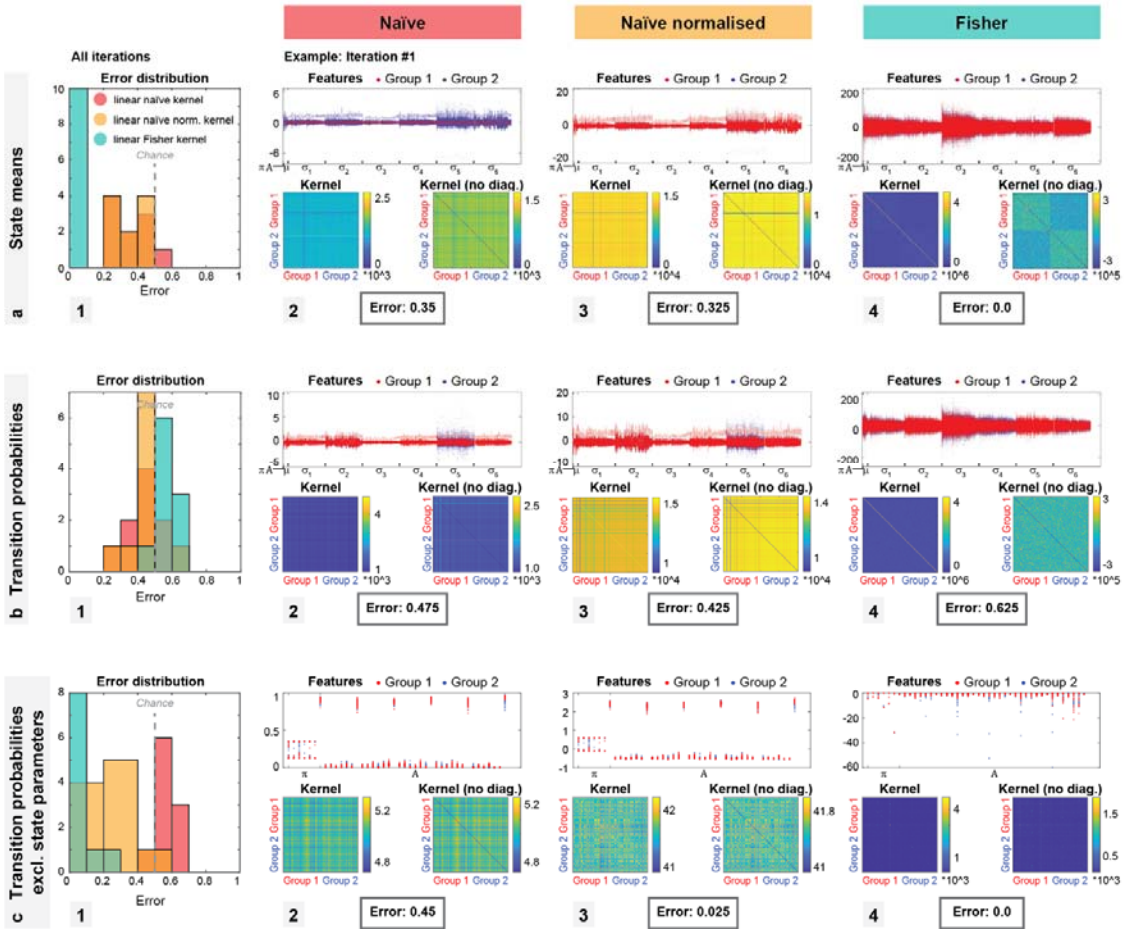

Figure 4 *Simulations. If the group difference was apparent in the features, there would be a clear difference between red and blue dots in the feature plot. If the group difference was apparent in the kernels, there would be a checkerboard-pattern with four squares in the kernel matrices: high similarity within the first half and within the second half of subjects, and low similarity between the first and the second half of subjects. Each kernel should also have the strongest similarity on the diagonal because each subject should be more similar to themselves than to any other subject. In the second kernel plots, we remove the diagonal for visualisation purposes to show the group difference more clearly.* ***a)*** *Simulating two groups of subjects that are different in their state means. The error distributions of all 10 iterations show that the Fisher kernel recovers the simulated group difference in all runs with 0% error (1). Features, kernel matrices, and kernel matrices with the diagonal removed for the first iteration for the linear naïve kernel (2), the linear naïve normalised kernel (3), and the linear Fisher kernel (4). The Fisher kernel matrices show an obvious checkerboard pattern corresponding to the within-group similarity and the between-group dissimilarity of the first and the second half of subjects.* ***b)*** *Simulating two groups of subjects that are different in their transition probabilities. Neither kernel is able to reliably recover the group difference, as shown in the error distribution of all 10 iterations (1), and the features and kernel matrices of one example iteration (2-4).* ***c)*** *Simulating two groups of subjects that are different in their transition probabilities but excluding state parameters when constructing the kernels. The Fisher kernel performs best in recovering the group difference as shown by the error distributions of all 10 iterations (1).*

In real data, the features driving the prediction may differ from trait to trait: For some traits, state parameters may be more relevant, while for other traits, transitions may be more

relevant. We therefore compare the effects of removing state features and the effects of removing transition features in all traits in the real data.

We found that state parameters were the most relevant features for the Fisher kernel predictions in all traits: As shown in **Figure 5a**, the prediction accuracy of the Fisher kernel significantly suffered when state features were removed (p=0.0009), while removing transition features had no significant effect (p>1). We observed the same effect in the naïve normalised kernel (no state features: p=0.0009; no transition features: p>1), while the naïve kernel was on average improved by removing state features (no state features: p=0.0009; no transition features: p>1). One reason for the dominance of state parameters may simply be that the state parameters outnumber the other parameters: In the full kernels, we have 15,300 state features (300 features associated with the state means and 15,000 features associated with the state covariances), but only 42 transition features (6 features associated with the state probabilities and 36 features associated with the transition probabilities). To compensate for this imbalance, we also constructed a version of the kernels where state parameters were reduced to the same amount as transition features using PCA, so that we have 84 features in total (42 transition features and the first 42 PCs of the 15,300 state features). These PCA-kernels performed significantly better than the ones where state features were removed in the Fisher kernel (p=0.0009), but worse than the kernels including all features at the original dimensionality (p=0.0009). This indicates that the fact that state parameters are more numerous than transition parameters does not explain why kernels including state features perform better. Instead, the content of the state features is more relevant for the prediction than the content of the transition features.

When looking at the performance separately for each variable (**Figure 5b and c**), we found that all but one variable were better predicted by the version of the kernel which included state features than the ones where state features were removed, while it did not seem to matter whether transition features were included. This indicates that the simulation case described above, where the relevant changes are in the transition probabilities, did not occur in the real data. In certain variables, reducing state features using PCA improved the accuracy compared to the full kernels, which is not unexpected since feature dimensionality reduction is known to be able to improve prediction accuracy by removing redundant features (Mwangi et al., 2014).

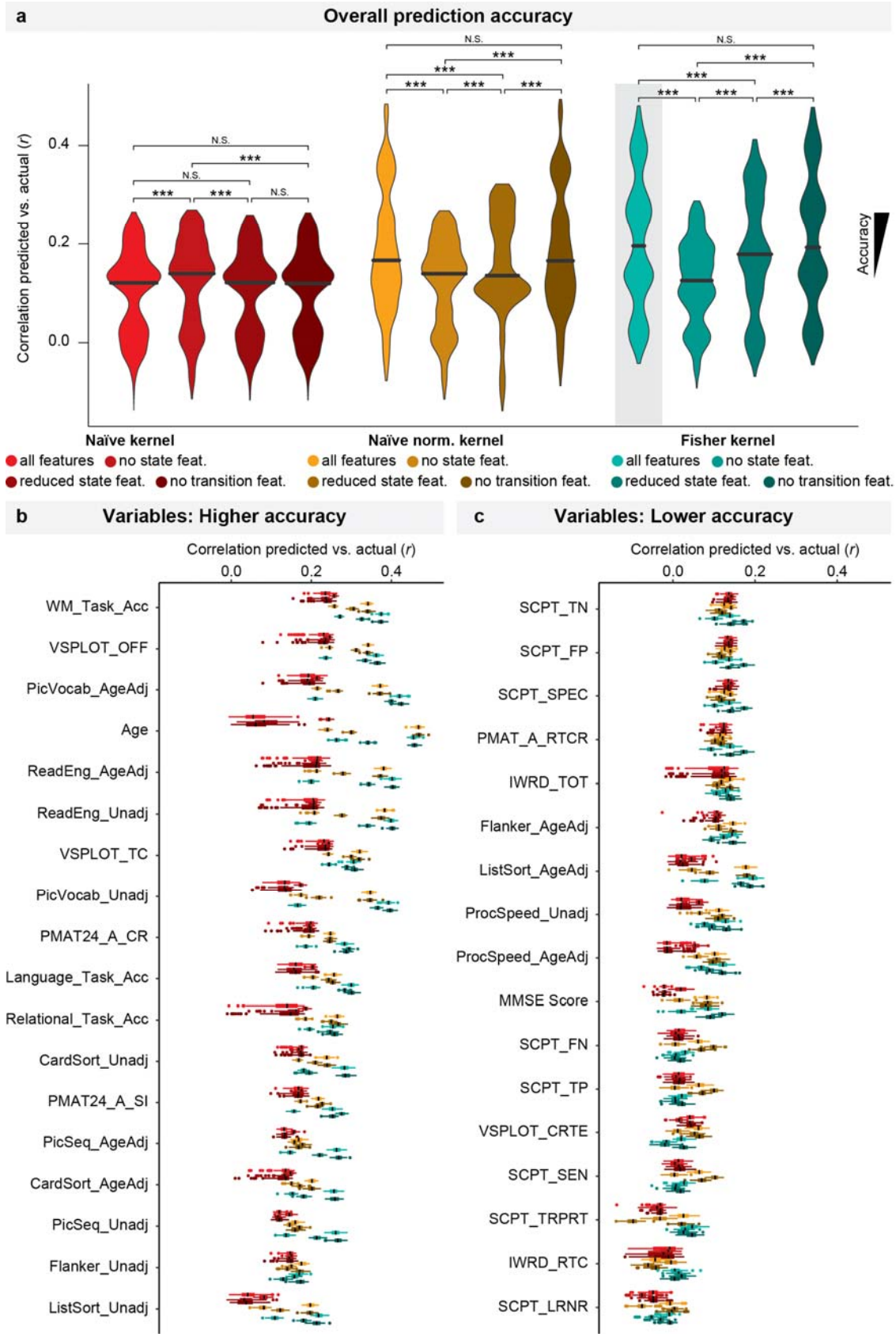
a
Overall prediction accuracy
Correlation predicted vs. actual (r)
0.4
0.2
0.0
N.S.
***
Accuracy
Naïve kernel
Naïve norm. kernel
Fisher kernel
all features
no state feat.
reduced state feat.
no transition feat.
b
Variables: Higher accuracy
WM_Task_Acc
VSPLOT_OFF
PicVocab_AgeAdj
Age
ReadEng_AgeAdj
ReadEng_Unadj
VSPLOT_TC
PicVocab_Unadj
PMAT24_A_CR
Language_Task_Acc
Relational_Task_Acc
CardSort_Unadj
PMAT24_A_SI
PicSeq_AgeAdj
CardSort_AgeAdj
PicSeq_Unadj
Flanker_Unadj
ListSort_Unadj
c
Variables: Lower accuracy
SCPT_TN
SCPT_FP
SCPT_SPEC
PMAT_A_RTCR
IWRD_TOT
Flanker_AgeAdj
ListSort_AgeAdj
ProcSpeed_Unadj
ProcSpeed_AgeAdj
MMSE Score
SCPT_FN
SCPT_TP
VSPLOT_CRTE
SCPT_SEN
SCPT_TRPRT
IWRD_RTC
SCPT_LRNR

__Figure 5__. Effects of removing sets of features on prediction accuracies. __a)__ In the overall prediction accuracies, removing state features significantly decreased performance in the Fisher kernel and the naïve normalised kernel, while removing transition features had no significant effect. __b)__ and __c)__ Removing features has similar effects on all variables, both better predicted (b) and worse predicted ones (c).

4 Discussion

We showed that the Fisher kernel accurately and reliably predicts traits from brain dynamics models trained on neuroimaging data, resulting in better generalisability and interpretability compared to other methods. It preserves the structure of the underlying brain dynamics model, making it ideal for combining generative and predictive models. We have compared the Fisher kernel to kernels which ignore the structure of the brain dynamics model ("naïve" kernels), to a previously used model based on Kullback-Leibler divergence (Vidaurre et al., 2021), and to models based on time-averaged FC. The linear Fisher kernel had a higher prediction accuracy than all other methods and was among the most reliable methods: It never produced excessive errors and was robust to changes in training sets.

We here focussed on fMRI, but the method can also be applied to other modalities like MEG or EEG and it can straightforwardly be implemented in any kernel-based prediction model or classifier, including kernel ridge regression, support vector machines (SVM), kernel fisher discriminant analysis (k-FDA), kernel logistic regression (KLR), or nonlinear PCA.

Our results were consistent in two alternative settings: when using less data per subject and when modelling brain dynamics only in terms of functional connectivity. This supports the generalisability of the results. It should be noted though that we observed overall lower accuracies when using only one scanning session consisting of 1,200 timepoints per subject, indicating that predicting phenotypes from models of brain dynamics requires a sufficiently large amount of data for each individual. Using a set of 35 different demographic and cognitive traits, we found that the traits that were best predicted by kernels from models of brain dynamics were language and spatial orientation skills, and here the Fisher kernel was generally the most accurate. We also found that the Gaussian version of the kernels are generally more error-prone and susceptible to changes in the training set, although they may predict more accurately in certain runs. Implementing Gaussian kernels in a predictive model is also computationally more expensive, making them less practical. While we here only tested robustness in terms of susceptibility to changes in CV folds, it remains to be shown to what extent model performance is sensitive to the random initialisation of the HMM, which affects the parameter estimation (Alonso & Vidaurre, 2023). Finally, we showed that the Fisher kernel is most sensitive to changes in state descriptions, i.e., individual differences in

the amplitude or functional connectivity of certain brain states. While this could be a disadvantage if a trait was more closely related to how an individual transitions between brain states, we found that this was not the case in any of the traits we tested here. This is surprising, given that we predicted a variety of cognitive traits, expecting that some traits would, for instance, be more closely related to an individual's FC in a certain brain state, while other traits would be better explained by their temporal pattern of switching between brain states. Other constructs than the ones we tested here may of course be more related to individual transition patterns. For this case, we showed in simulations that the Fisher kernel can be modified to recognise changes in transitions if they are of interest for the specific research question.

Models of brain dynamics in unconstrained cognition do not directly lend themselves to being used as linear predictors, because of the high number of parameters and their complex relationships. Kernels are a computationally efficient approach that allow working with nonlinear, high-dimensional features, such as parameters from a model of brain dynamics, in a simple linear predictive model. How to construct a kernel is however not always straightforward (Azim & Ahmed, 2018; Shawe-Taylor & Cristianini, 2004). We here demonstrated that constructing a kernel that preserves the structure of the underlying model, like the Fisher kernel, can achieve better, more reliable performance than other methods when using models of brain dynamics to predict subject traits.

Such a kernel can be useful in various applications. For instance, there is growing interest in combining different data types or modalities, such as structural, static, and dynamic measures to predict phenotypes (Engemann et al., 2020; Schouten et al., 2016). While directly combining the features from each modality can be problematic, kernels that are properly derived for each modality can be easily combined using approaches such as Multi Kernel Learning (MKL) (Gönen & Alpaydın, 2011), which can improve prediction accuracy of multimodal studies (Vaghari et al., 2022). In a clinical context, describing individual variability for diagnosis and individual patients' outcome prediction is the ultimate goal (Marquand et al., 2016; Stephan et al., 2017; Wen et al., 2022; Wolfers et al., 2015), but the focus so far has mostly been on static or structural information, leaving the potentially crucial information from brain dynamics untapped. In order to be able to use predictive models from brain dynamics in a clinical context, predictions must be reliable, particularly if we want to interpret model errors, as in models of "brain age". As we demonstrated, there can be extreme errors and large variation in some predictive models, and these issues are not resolved by estimating model performance in a standard cross-validated fashion. We here showed that taking the structure of the underlying model into account, and thoroughly assessing not only accuracy but also errors and robustness, we can reliably use information from brain

dynamics to predict individual traits. This will allow gaining crucial insights into cognition and behaviour from how brain function changes over time, beyond structural and static information.

Acknowledgements

DV is supported by a Novo Nordisk Foundation Emerging Investigator Fellowship (NNF19OC-0054895) and an ERC Starting Grant (ERC-StG-2019-850404). We thank Ben Griffin and Steve Smith for useful discussions and technical collaboration.

References

Ahrends, C., Stevner, A., Pervaiz, U., Kringelbach, M. L., Vuust, P., Woolrich, M. W., & Vidaurre, D. (2022). Data and model considerations for estimating time-varying functional connectivity in fMRI. *Neuroimage*, *252*, 119026. https://doi.org/10.1016/j.neuroimage.2022.119026

Alonso, S., & Vidaurre, D. (2023). Towards stability of dynamic FC estimates in neuroimaging and electrophysiology: solutions and limits. *bioRxiv*, 2023.2001.2018.524539. https://doi.org/10.1101/2023.01.18.524539

Azim, T., & Ahmed, S. (2018). *Composing Fisher Kernels from Deep Neural Models*. Springer.

Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2013). Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, *112*, 172-178. https://doi.org/https://doi.org/10.1016/j.neucom.2012.12.039

Baum, L. E., & Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, *73*(3), 360-363, 364. https://doi.org/

Baum, L. E., & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, *37*(6), 1554-1563, 1510. https://doi.org/10.1214/aoms/1177699147

Beckmann, C. F. (2012). Modelling with independent components. *Neuroimage*, *62*(2), 891-901. https://doi.org/10.1016/j.neuroimage.2012.02.020

Beckmann, C. F., Mackay, C. E., Filippini, N., & Smith, S. M. (2009). Group comparison of resting-state FMRI data using multi-subject ICA and dual regression. *Neuroimage*, *47*, S148. https://doi.org/10.1016/S1053-8119(09)71511-3

Cole, J. H., & Franke, K. (2017). Predicting Age Using Neuroimaging: Innovative Brain Ageing Biomarkers. *Trends Neurosci*, *40*(12), 681-690. https://doi.org/10.1016/j.tins.2017.10.001

Denissen, S., Engemann, D. A., De Cock, A., Costers, L., Baijot, J., Laton, J., Penner, I. K., Grothe, M., Kirsch, M., D'Hooghe M, B., D'Haeseleer, M., Dive, D., De Mey, J., Van Schependom, J., Sima, D. M., & Nagels, G. (2022). Brain age as a surrogate marker for cognitive performance in multiple sclerosis. *Eur J Neurol*, *29*(10), 3039-3049. https://doi.org/10.1111/ene.15473

Do, M. N. (2003). Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models [Article]. *IEEE Signal Processing Letters*, *10*(4), 115-118. https://doi.org/10.1109/LSP.2003.809034

Engemann, D. A., Kozynets, O., Sabbagh, D., Lemaître, G., Varoquaux, G., Liem, F., & Gramfort, A. (2020). Combining magnetoencephalography with magnetic resonance imaging enhances learning of surrogate-biomarkers. *eLife*, *9*, e54055. https://doi.org/10.7554/eLife.54055

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., Van Essen, D. C., Jenkinson, M., & Consortium, W. U.-M. H. (2013). The minimal preprocessing pipelines for the Human Connectome Project. *Neuroimage*, *80*, 105-124. https://doi.org/10.1016/j.neuroimage.2013.04.127

Griffanti, L., Salimi-Khorshidi, G., Beckmann, C. F., Auerbach, E. J., Douaud, G., Sexton, C. E., Zsoldos, E., Ebmeier, K. P., Filippini, N., Mackay, C. E., Moeller, S., Xu, J., Yacoub, E., Baselli, G., Ugurbil, K., Miller, K. L., & Smith, S. M. (2014). ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging. *Neuroimage*, *95*, 232-247. https://doi.org/10.1016/j.neuroimage.2014.03.034

Gumbel, E. J. (1958). *Statistics of Extremes*. Columbia University Press. https://doi.org/doi:10.7312/gumb92958

Gönen, M., & Alpaydın, E. (2011). Multiple kernel learning algorithms. *J Mach Learn Res*, *12*, 2211-2268. https://doi.org/10.5555/1953048.2021071

Jaakkola, T., Diekhans, M., & Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. *Proc Int Conf Intell Syst Mol Biol*, 149-158.

Jaakkola, T., & Haussler, D. (1998). Exploiting Generative Models in Discriminative Classifiers. NIPS,

Liégeois, R., Li, J., Kong, R., Orban, C., Van De Ville, D., Ge, T., Sabuncu, M. R., & Yeo, B. T. T. (2019). Resting brain dynamics at different timescales capture distinct aspects of human behavior. *Nat Commun*, *10*(1), 2317. https://doi.org/10.1038/s41467-019-10317-7

Lurie, D. J., Kessler, D., Bassett, D. S., Betzel, R. F., Breakspear, M., Kheilholz, S., Kucyi, A., Liégeois, R., Lindquist, M. A., McIntosh, A. R., Poldrack, R. A., Shine, J. M., Thompson, W. H., Bielczyk, N. Z., Douw, L., Kraft, D., Miller, R. L., Muthuraman, M., Pasquini, L., . . . Calhoun, V. D. (2019). Questions and controversies in the study of time-varying functional connectivity in resting fMRI. *Netw Neurosci*, *4*(1), 30-69. https://doi.org/10.1162/netn_a_00116

MacKay, D. J. C., Kay, D. J. C. M., & Press, C. U. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press. https://books.google.dk/books?id=AKuMj4PN_EMC

Marquand, A. F., Rezek, I., Buitelaar, J., & Beckmann, C. F. (2016). Understanding Heterogeneity in Clinical Cohorts Using Normative Models: Beyond Case-Control Studies. *Biological psychiatry*, *80*(7), 552-561. https://doi.org/10.1016/j.biopsych.2015.12.023

Mwangi, B., Tian, T. S., & Soares, J. C. (2014). A Review of Feature Reduction Techniques in Neuroimaging. *Neuroinformatics*, *12*(2), 229-244. https://doi.org/10.1007/s12021-013-9204-3

Rosenberg, M. D., Finn, E. S., Scheinost, D., Papademetris, X., Shen, X., Constable, R. T., & Chun, M. M. (2016). A neuromarker of sustained attention from whole-brain functional connectivity. *Nat Neurosci*, *19*(1), 165-171. https://doi.org/10.1038/nn.4179

Salimi-Khorshidi, G., Douaud, G., Beckmann, C. F., Glasser, M. F., Griffanti, L., & Smith, S. M. (2014). Automatic denoising of functional MRI data: combining independent component analysis and hierarchical fusion of classifiers. *Neuroimage*, *90*, 449-468. https://doi.org/10.1016/j.neuroimage.2013.11.046

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables.

Schouten, T. M., Koini, M., de Vos, F., Seiler, S., van der Grond, J., Lechner, A., Hafkemeijer, A., Möller, C., Schmidt, R., de Rooij, M., & Rombouts, S. A. R. B. (2016). Combining anatomical, diffusion, and resting state functional magnetic resonance imaging for individual classification of mild and moderate Alzheimer's disease. *NeuroImage: Clinical*, *11*, 46-51. https://doi.org/10.1016/j.nicl.2016.01.002

Schölkopf, B., Smola, A. J., & Bach, F. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511809682

Shen, X., Finn, E. S., Scheinost, D., Rosenberg, M. D., Chun, M. M., Papademetris, X., & Constable, R. T. (2017). Using connectome-based predictive modeling to predict individual behavior from brain connectivity. *Nat Protoc*, *12*(3), 506-518. https://doi.org/10.1038/nprot.2016.178

Smith, S. M., Beckmann, C. F., Andersson, J., Auerbach, E. J., Bijsterbosch, J., Douaud, G., Duff, E., Feinberg, D. A., Griffanti, L., Harms, M. P., Kelly, M., Laumann, T., Miller, K. L., Moeller, S., Petersen, S., Power, J., Salimi-Khorshidi, G., Snyder, A. Z., Vu, A. T., . . . Consortium, W. U.-M. H. (2013). Resting-state fMRI in the Human Connectome Project. *Neuroimage*, *80*, 144-168. https://doi.org/10.1016/j.neuroimage.2013.05.039

Smith, S. M., Vidaurre, D., Alfaro-Almagro, F., Nichols, T. E., & Miller, K. L. (2019). Estimation of brain age delta from brain imaging. *Neuroimage*, *200*, 528-539. https://doi.org/https://doi.org/10.1016/j.neuroimage.2019.06.017

Smith, S. M., Vidaurre, D., Beckmann, C. F., Glasser, M. F., Jenkinson, M., Miller, K. L., Nichols, T. E., Robinson, E. C., Salimi-Khorshidi, G., Woolrich, M. W., Barch, D. M., Uğurbil, K., & Van Essen, D. C. (2013). Functional connectomics from resting-state fMRI. *Trends Cogn Sci*, *17*(12), 666-682. https://doi.org/10.1016/j.tics.2013.09.016

Stephan, K. E., Schlagenhauf, F., Huys, Q. J. M., Raman, S., Aponte, E. A., Brodersen, K. H., Rigoux, L., Moran, R. J., Daunizeau, J., Dolan, R. J., Friston, K. J., & Heinz, A. (2017). Computational neuroimaging strategies for single patient predictions. *Neuroimage*, *145*, 180-199. https://doi.org/https://doi.org/10.1016/j.neuroimage.2016.06.038

Vaghari, D., Kabir, E., & Henson, R. N. (2022). Late combination shows that MEG adds to MRI in classifying MCI versus controls. *Neuroimage*, *252*, 119054. https://doi.org/10.1016/j.neuroimage.2022.119054

van der Maaten, L. (2011). Learning Discriminative Fisher Kernels. Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., & Ugurbil, K. (2013). The WU-Minn Human Connectome Project: an overview. *Neuroimage*, *80*, 62-79. https://doi.org/10.1016/j.neuroimage.2013.05.041

Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T. E. J., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S. W., Della Penna, S., Feinberg, D., Glasser, M. F., Harel, N., Heath, A. C., Larson-Prior, L., Marcus, D., Michalareas, G., Moeller, S., . . . Consortium, W. U.-M. H. (2012). The Human Connectome Project: a data acquisition perspective. *Neuroimage*, *62*(4), 2222-2231. https://doi.org/10.1016/j.neuroimage.2012.02.018

Varoquaux, G., Raamana, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., & Thirion, B. (2017). Assessing and tuning brain decoders: Cross-validation, caveats, and guidelines. *Neuroimage*, *145*, 166-179. https://doi.org/https://doi.org/10.1016/j.neuroimage.2016.10.038

Vidaurre, D., Llera, A., Smith, S. M., & Woolrich, M. W. (2021). Behavioural relevance of spontaneous, transient brain network interactions in fMRI. *Neuroimage*, *229*, 117713. https://doi.org/10.1016/j.neuroimage.2020.117713

Vidaurre, D., Quinn, A. J., Baker, A. P., Dupret, D., Tejero-Cantero, A., & Woolrich, M. W. (2016). Spectrally resolved fast transient brain states in electrophysiological data. *Neuroimage*, *126*, 81-95. https://doi.org/10.1016/j.neuroimage.2015.11.047

Vidaurre, D., Smith, S. M., & Woolrich, M. W. (2017). Brain network dynamics are hierarchically organized in time. *Proc Natl Acad Sci U S A*, *114*(48), 12827-12832. https://doi.org/10.1073/pnas.1705120114

Wen, J., Varol, E., Sotiras, A., Yang, Z., Chand, G. B., Erus, G., Shou, H., Abdulkadir, A., Hwang, G., Dwyer, D. B., Pigoni, A., Dazzan, P., Kahn, R. S., Schnack, H. G., Zanetti, M. V., Meisenzahl, E., Busatto, G. F., Crespo-Facorro, B., Rafael, R. G., . . . Alzheimer's Disease Neuroimaging, I. (2022). Multi-scale semi-supervised clustering

of brain images: Deriving disease subtypes. *Med Image Anal*, *75*, 102304. https://doi.org/10.1016/j.media.2021.102304

Winkler, A. M., Webster, M. A., Vidaurre, D., Nichols, T. E., & Smith, S. M. (2015). Multi-level block permutation. *Neuroimage*, *123*, 253-268. https://doi.org/https://doi.org/10.1016/j.neuroimage.2015.05.092

Wolfers, T., Buitelaar, J. K., Beckmann, C. F., Franke, B., & Marquand, A. F. (2015). From estimating activation locality to predicting disorder: A review of pattern recognition for neuroimaging-based psychiatric diagnostics. *Neurosci Biobehav Rev*, *57*, 328-349. https://doi.org/10.1016/j.neubiorev.2015.08.001

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x