

Supplementary Material to  
A Unified Framework for Multimodal Structure–function  
Mapping Based on Eigenmodes

Samuel Deslauriers-Gauthier, Mauro Zucchelli, Matteo Frigo, and Rachid Deriche

*Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur, France*

## 1 Parameter estimation

We discuss here the optimization strategy that was used to estimate the model parameters from the training data. The problem to solve can be formulated as

$$\underset{f}{\text{minimize}} \sum_{k=1}^K \|f(\mathbf{S}_k) - \mathbf{F}_k\|_{\mathcal{F}}^2 \quad (1)$$

where  $K$  is the number of training subjects,  $\|\cdot\|_{\mathcal{F}}$  is the Frobenius norm, and where  $f$  is the mapping. Because our proposed general mapping has the form

$$f(\mathbf{A}) = \sum_{n=0}^{N-1} g(\lambda_n)h(\mathbf{u}_n) + \mathbf{C} \quad (2)$$

the optimization is over the parameters of  $g$ ,  $h$ , and the entries of  $\mathbf{C}$ . In general, the optimization uses an iterative procedure, where the only parameters of either  $g$ ,  $h$ , or  $\mathbf{C}$  are allowed to vary in turn. The full optimization procedure therefore iterates over three subproblems as described by the following pseudo-code.

**Data:**  $K, m, (\mathbf{A}_k, \mathbf{F}_k)$  for  $k = 1 \dots K$   
**Result:**  $g, h, \mathbf{C}$   
 $g, h, \mathbf{C} = \text{initialize}();$   
 $i = 0;$   
**while**  $i < m$  **do**  
     $\mathbf{C} = \text{optimizeConstant}(\mathbf{C}, g, h, (\mathbf{A}_k, \mathbf{F}_k)$  for  $k = 1 \dots K);$   
     $g = \text{optimizeEigenvalues}(\mathbf{C}, g, h, (\mathbf{A}_k, \mathbf{F}_k)$  for  $k = 1 \dots K);$   
     $h = \text{optimizeEigenvectors}(\mathbf{C}, g, h, (\mathbf{A}_k, \mathbf{F}_k)$  for  $k = 1 \dots K);$   
     $i = i + 1;$   
**end**

**Algorithm 1:** General optimization strategy

Note that the input is  $A_k$  and not  $S_k$ , which implies the Laplacian (or any other transform) has already been applied, if it is part of the model, before starting the optimization routine. In the three functions `optimizeConstant`, `optimizeEigenvalues`, and `optimizeEigenvectors`, a single aspect of the mappings is optimized, while keeping the others constant. For the initialization and for each subproblem embodied by these functions, the implementation depend on the choice of the form of  $g$ ,  $h$ , and  $\mathbf{C}$ . As an example, for the diffusion model of Abdelnour et al. (2014), the maximum number of iterations  $m$  is set to 1, the optimization over  $\mathbf{C}$  always returns  $\mathbf{0}$ , the optimization over  $h$  always returns a function that computes the outer product of the input vectors, and the optimization over  $g$  is a polynomial regression. Table 1 provides the optimization details for each of the possible choices of the constant, eigenvalue map, and eigenvector map. For the minimization over the parameters  $h$ , Pymanopt (Townsend et al. 2016) was used to ensure that  $\mathbf{R}$  is a rotation matrix and that  $\mathbf{Q}$  is orthonormal, respectively. For the minimization over the parameters of  $g$ , the BFGS solver provided by Scipy (Jones et al. 2001) was used, and the polynomial regression is a custom implementation.

As stated previously, the value of  $m$  sets the maximum number of iterations of the general optimization problem. For models involving multiple objectives (e.g. over eigenvalues and eigenvectors), it was empirically set to 20 by observing the converge plots. An example of a convergence plot is illustrated in Figure 1 for subject 105923 (and therefore trained on the other 49 subjects) of the Human Connectome Project (HCP), and the convergence of the algorithm is clearly visible.

Program	Form	Initialization	Solver
optimizeConstant	$\bar{\mathbf{F}}$	$\mathbf{C} = \frac{1}{K} \sum_{k=1}^K \mathbf{F}_k$	–
optimizeConstant	$\mathbf{0}$	$\mathbf{C} = \mathbf{0}$	–
optimizeConstant	$b\mathbf{I}$	$\mathbf{C} = \mathbf{I}$	BFGS (Scipy)
optimizeConstant	$b\mathbf{C}$	$\mathbf{C} = \mathbf{I}$	Steepest descent (Pymanopt)
optimizeEigenvalues	$\lambda$	–	–
optimizeEigenvalues	$e^{-\beta t \lambda}$	$\beta t = 0$	BFGS (Scipy)
optimizeEigenvalues	$a e^{-\alpha \lambda}$	$\beta t = 0$	BFGS (Scipy)
optimizeEigenvalues	$\sum_{m=0}^M a_m \lambda^m$	$a_m = 0, m = 0, \dots, M$	Polynomial regression
optimizeEigenvectors	$\mathbf{u}_n \mathbf{u}_n^T$	–	–
optimizeEigenvectors	$\mathbf{R} \mathbf{u}_n \mathbf{u}_n^T \mathbf{R}^T$	$\mathbf{R} = \mathbf{I}$	Steepest descent (Pymanopt)
optimizeEigenvectors	$\mathbf{q}_n \mathbf{q}_n^T$	$\mathbf{q}_n = \mathbf{u}_n$	Steepest descent (Pymanopt)

Table 1: The initialization and optimization strategy used for each form of the constant, eigenvalue map, and eigenvector map. The BFGS algorithm is provided by Scipy, the steepest descent over rotation matrix by Pymanopt, and the polynomial regression is a custom implementation.

## 2 Results on session 2

The main text of the manuscript presents the performance of the mappings on the HCP data of the first resting state functional MRI session. For completeness, the results computed on the data from the second session are illustrated here in Figure 2. We observed no significant differences between the two sessions.

## 3 Density connectomes

Figure 3 illustrated the performance of the mapping when using a density connectome instead of a count, length, or SIFT2 connectome. The density connectome is created by normalizing each entry of the count connectome by the sum of the area of the two involved regions.

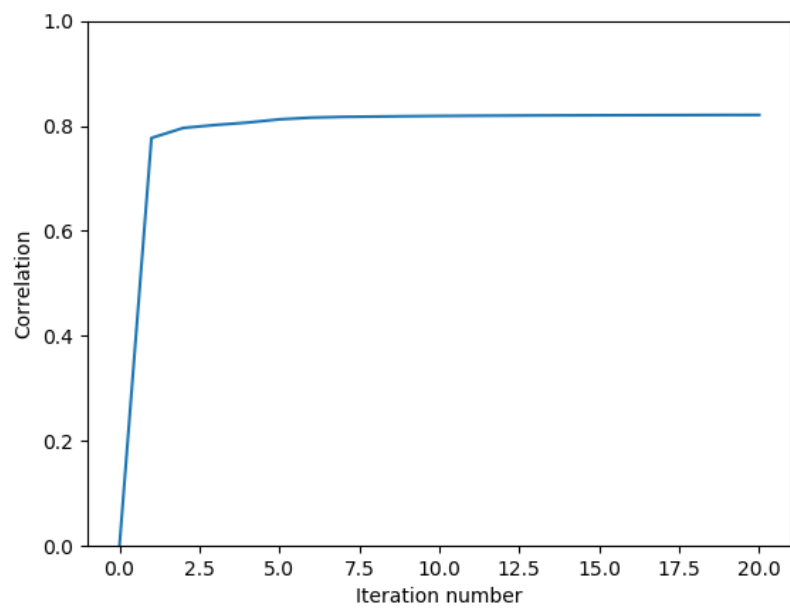


Figure 1: Illustration of the convergence plot for model 7 applied to subject 105923 of the HCP.

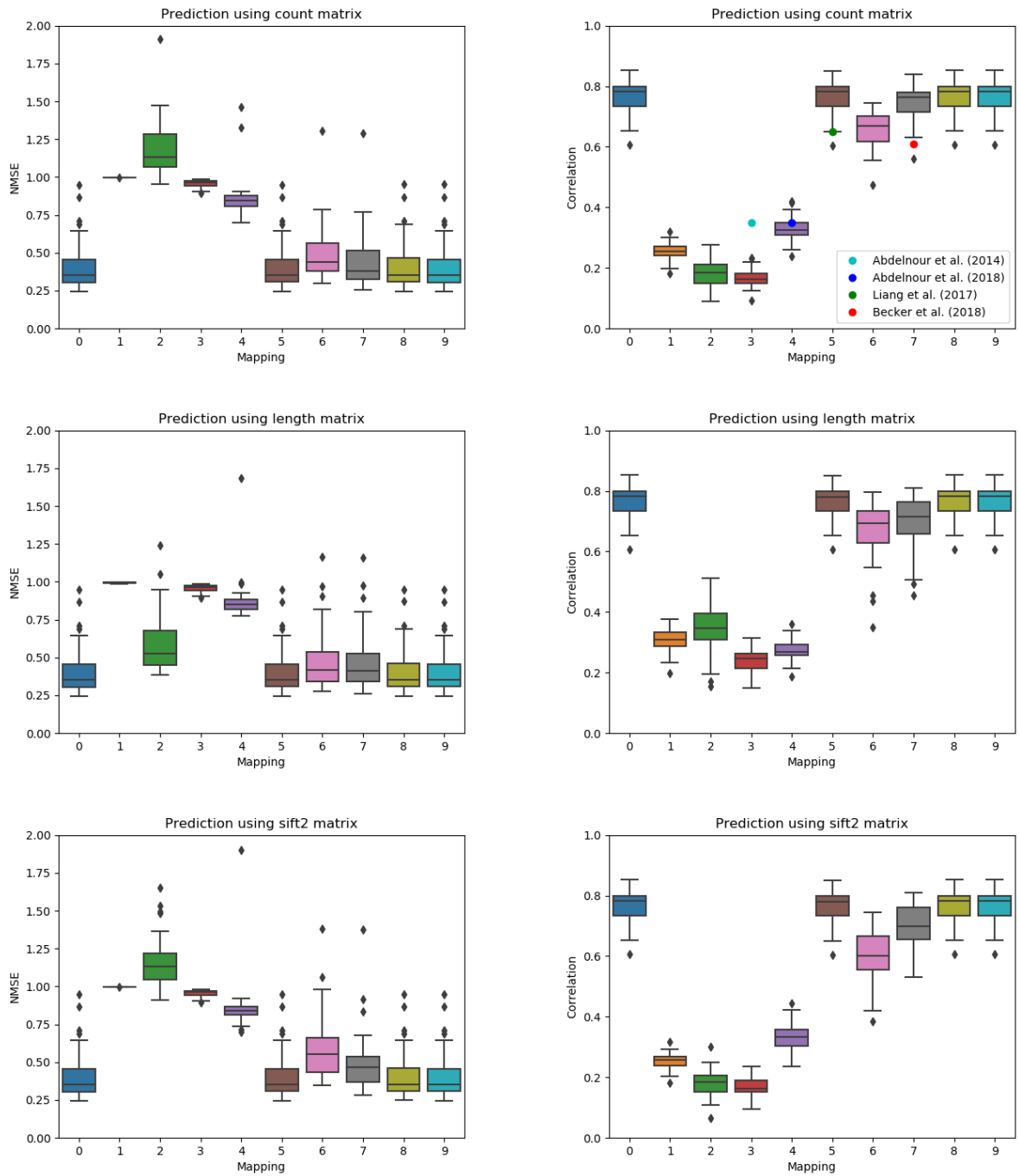


Figure 2: Normalized mean squared error (left panel, lower is better) and correlation (right panel, higher is better) between the predicted and true functional matrices for each mapping of Table 1 (of the main text) for multi-subject testing. The mean correlation reported by previous publications are illustrated using circles in the column corresponding to their model. These results use the second resting state session of the HCP, unlike the results of Figure 4 in the main text which use the first session.

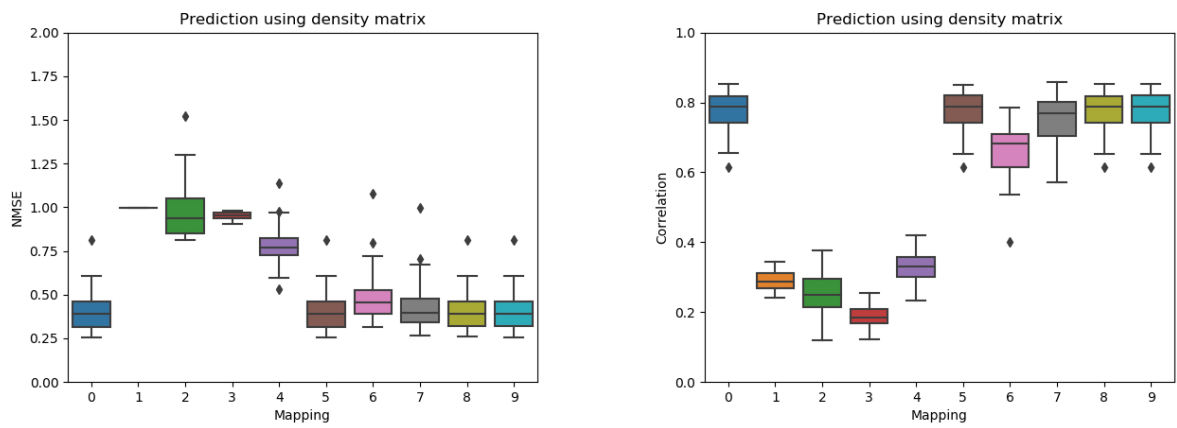


Figure 3: Normalized mean squared error (left panel, lower is better) and correlation (right panel, higher is better) between the predicted and true functional matrices for each mapping of Table 1 (of the main text) for multi-subject testing. Unlike the count, length, and SIFT2 structural matrices used in the manuscript, here the density connectome was used.