Important concepts and considerations in predictive modeling

Oscar Miranda-Domínguez, PhD, MSc.

Research Assistant Professor

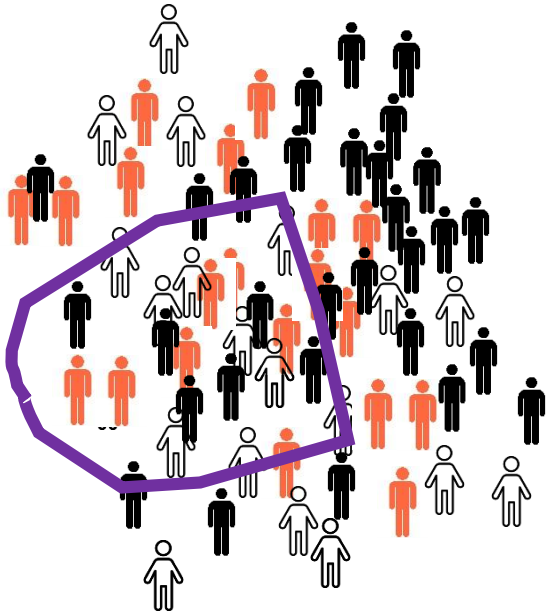
Developmental Cognition and Neuroimaging Lab, OHSU

Models try to identify associations between variables:

X, predictor variables
y, outcome variables

Models in clinical research have specific problems:

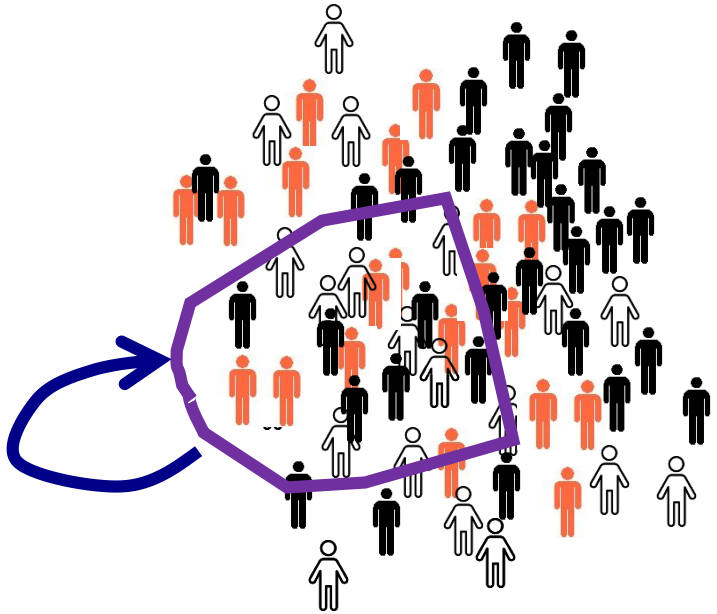
- Limited samples
- Multiple variables
  - Thousands!
- Unknown model structure



Entire population

While it is easy to obtain models that can describe within-sample data…

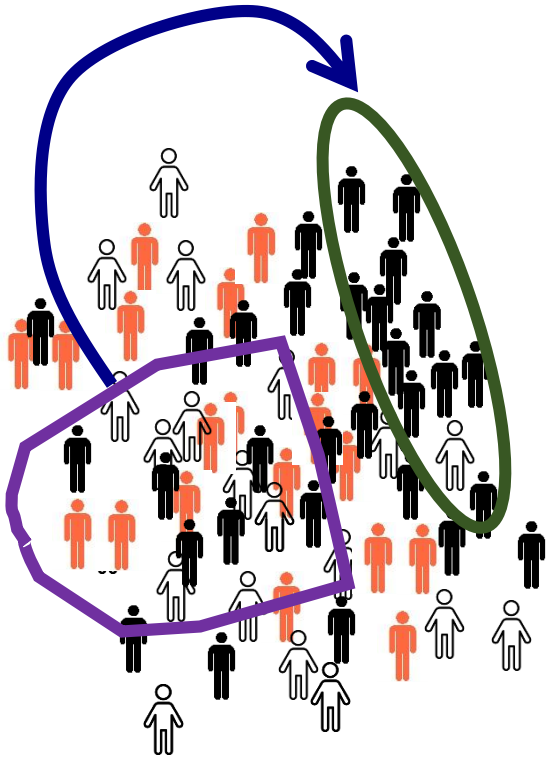
- Limited samples
- Multiple variables
  - Thousands!
- Unknown model structure



Entire population

it is hard to obtain models that can predict outcome in out-of-sample data

- Limited samples
- Multiple variables
  - Thousands!
- Unknown model structure



Entire population

The question is why?

More importantly, what can be done to improve predictions across datasets?

Topics

- Partial-least squares Regression
  - Feature Selection
  - Cross-Validation
  - Null Distribution/Permutations
  - An Example
- Regularization
  - Truncated singular value decomposition
  - Connectotyping: model based functional connectivity
  - Example: models that generalize across datasets!

Feature Selection

How relevant is the balance between the number of variables and observations?

 # Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \\ 3.9 &= 2.1A \end{aligned}$$

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

<u>Select the solution with the lowest mean square error!</u>

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

<u>Select the solution with the lowest mean square error!</u>

$$\begin{bmatrix} 4.0 \\ 3.9 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.1 \end{bmatrix} A$$

$$y = xA$$

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

<u>Select the solution with the lowest mean square error!</u>

$$\begin{bmatrix} 4.0 \\ 3.9 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.1 \end{bmatrix} A$$

$$y = xA$$

Using linear algebra (***x*** **pseudo-inverse**)

$$A = (x'x)^{-1}x'y$$

$$A \approx 1.9286$$

This A minimizes $\sum \mathbf{residuals}^2$

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

<u>Select the solution with the lowest mean square error!</u>

$$\begin{bmatrix} 4.0 \\ 3.9 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.1 \end{bmatrix} A$$

$$y = xA$$

Using linear algebra (***x*** **pseudo-inverse**)

$$A = (x'x)^{-1}x'y$$

$$A \approx 1.9286$$

This A minimizes $\sum \mathbf{residuals}^2$

Measurements < # Variables

What about (real) limited data:

$$8 = 4\alpha + \beta$$

There are 2 variables (α and β) and 1 measurements.

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

<u>Select the solution with the lowest mean square error!</u>

$$\begin{bmatrix} 4.0 \\ 3.9 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.1 \end{bmatrix} A$$

$$y = xA$$

Using linear algebra (***x*** **pseudo-inverse**)

$$A = (x'x)^{-1}x'y$$

$$A \approx 1.9286$$

This A minimizes $\sum \text{residuals}^2$

Measurements < # Variables

What about (real) limited data:

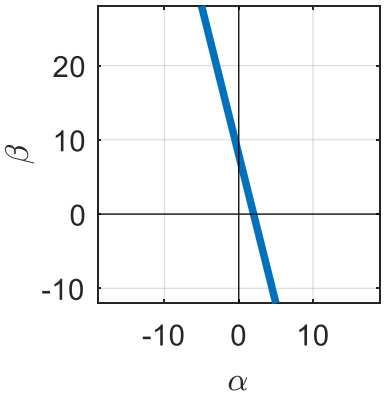
$$8 = 4\alpha + \beta$$

There are 2 variables (α and β) and 1 measurements.

Solving the system:

$$8 - 4\alpha = \beta$$

All the points on $\beta = 8 - 4\alpha$ solve the system.



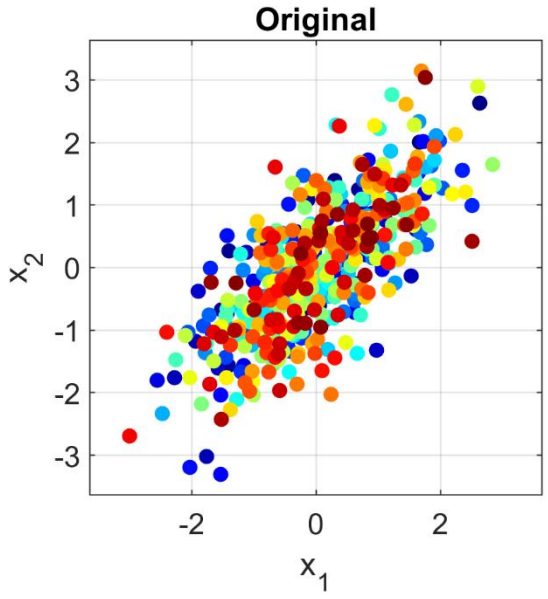
In other words, there is an infinite number of solutions!

For predictive models it’s important to limit the number of features relative to your sample size

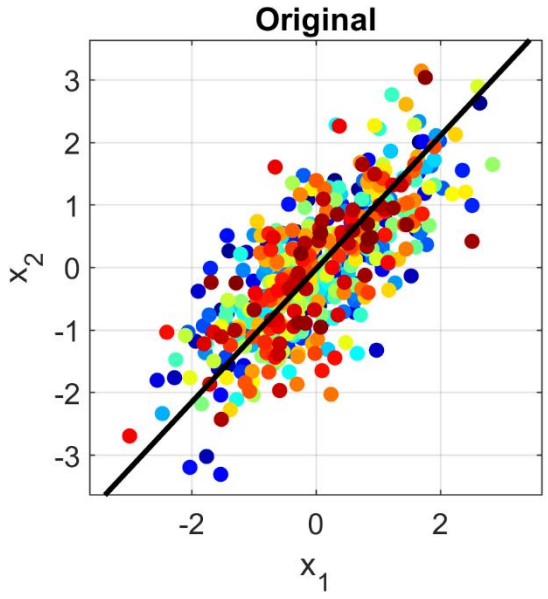
- This ‘feature reduction’ can be done in a number of ways.
- For partial least squares regression you reduce features based on how well models predict outcome.
  - What do I mean by that?

Let's revisit Principal Components Analysis

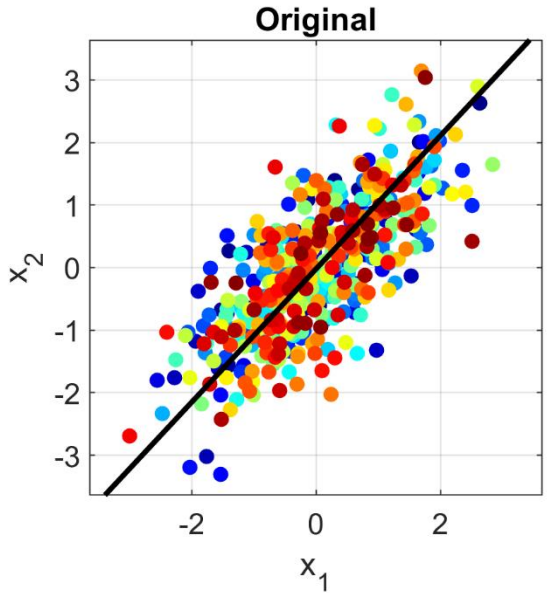
Let say you have a set of **predictor** variables with some correlation

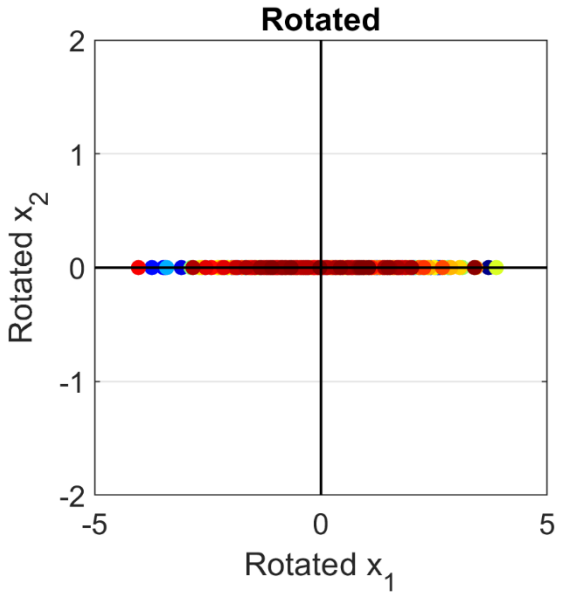


If you define a new set of axis, you might have a better description of the system



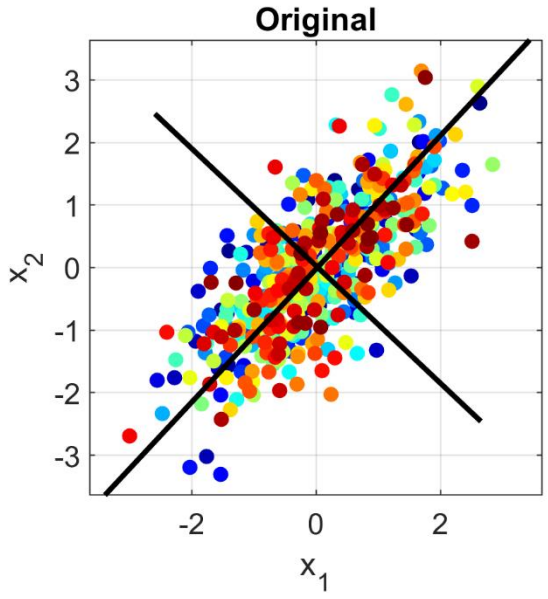
As most of the variance is observed across the black line, we can use it as a new base or axis

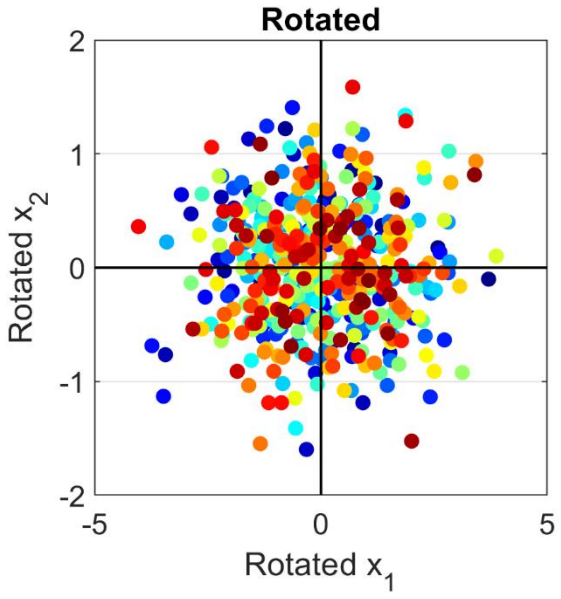




You can add more axis to explain more variance

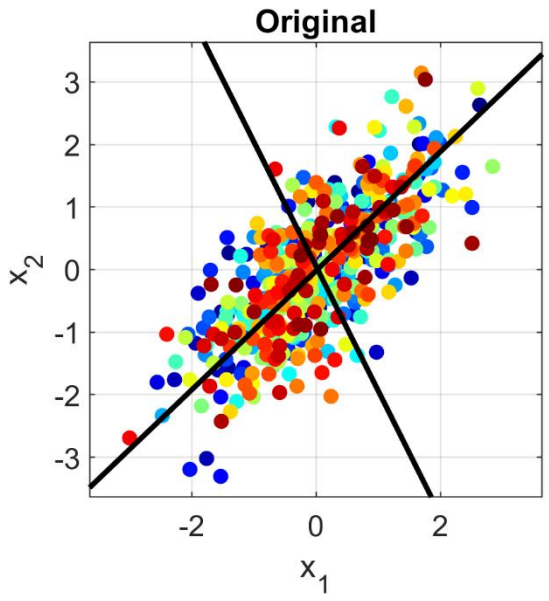
Additional axis are selected to be perpendicular to each other (orthogonal)

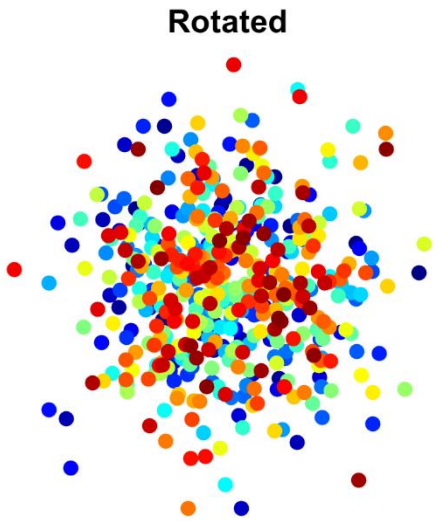




While useful, PCA does not take into account the outcome variable

In partial least squares regression (PLSR) you add an extra constrain selecting a rotation that maximizes outcome prediction

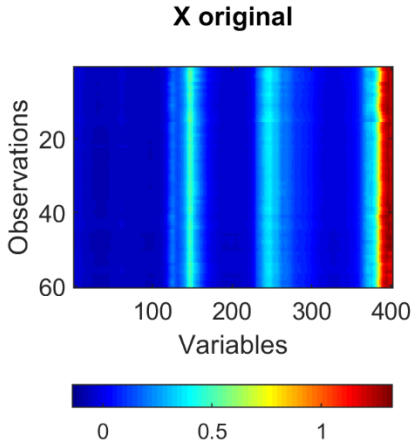


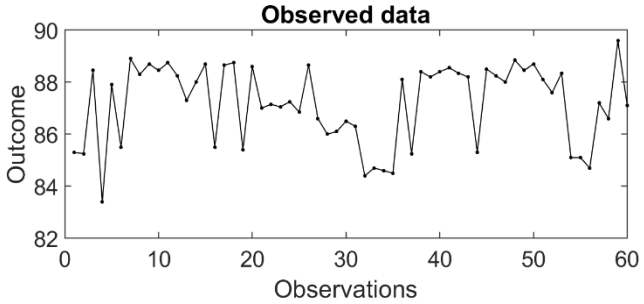


You can reduce the number of features by selecting different number of components (axis) and make predictions with those components

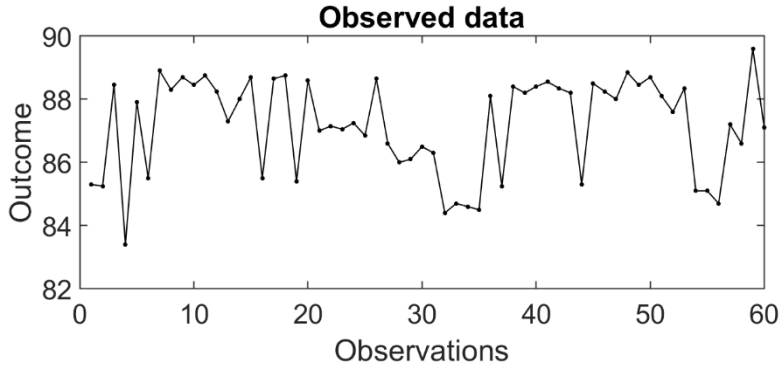
Example

Let’s suppose we like to predict an outcome given 401 variables and 60 observations

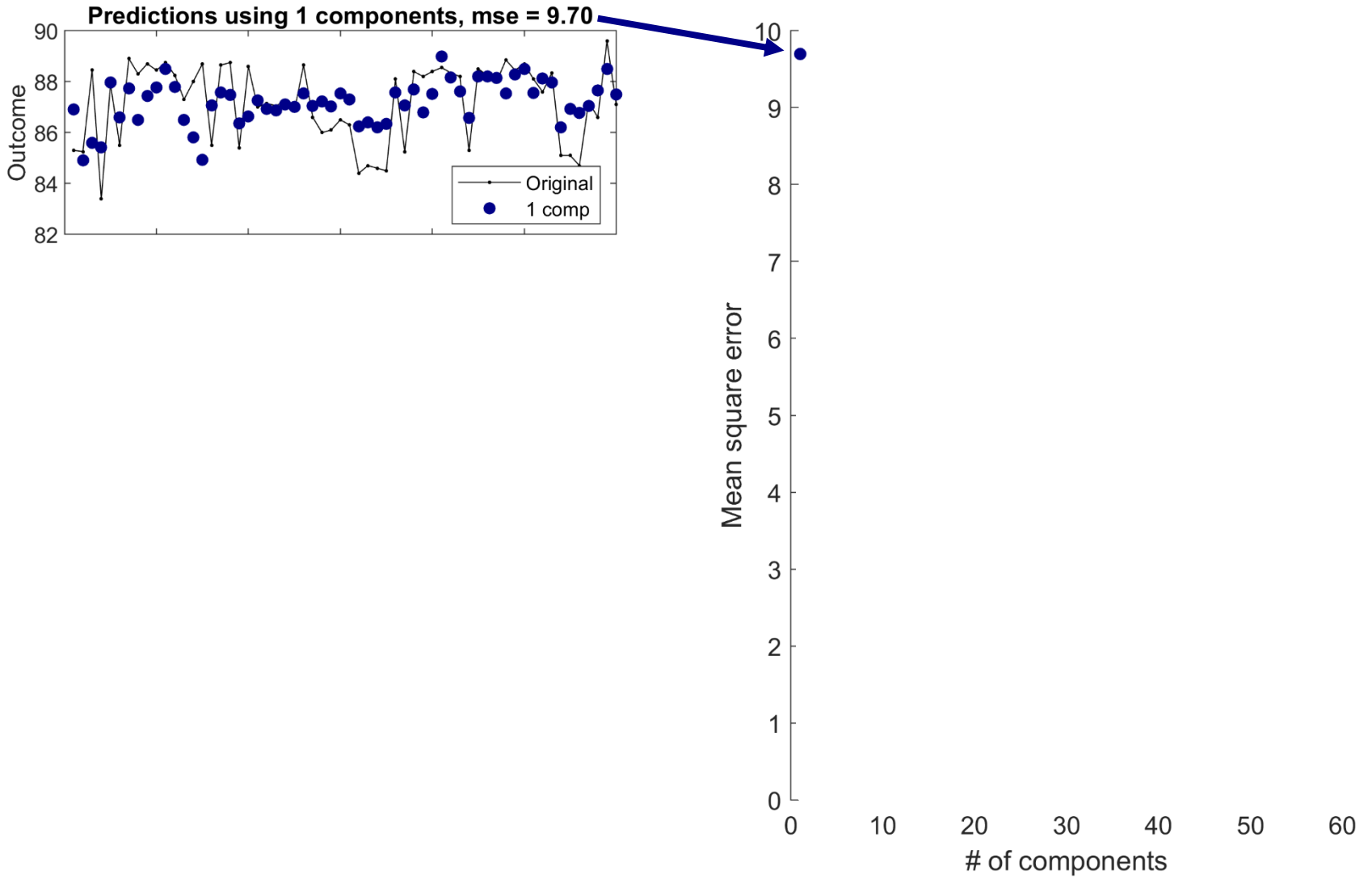




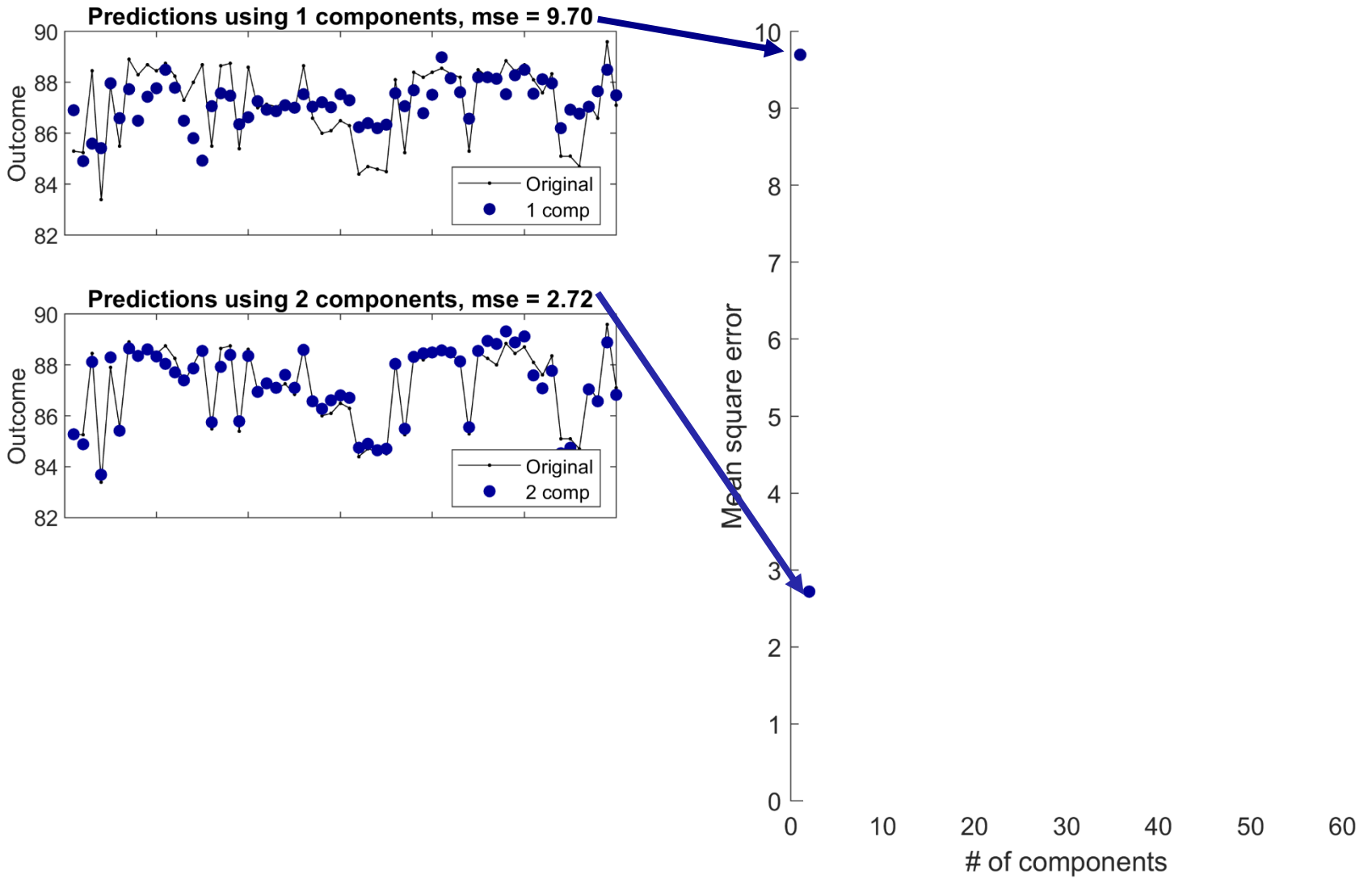
Observations



Predictions using only one component

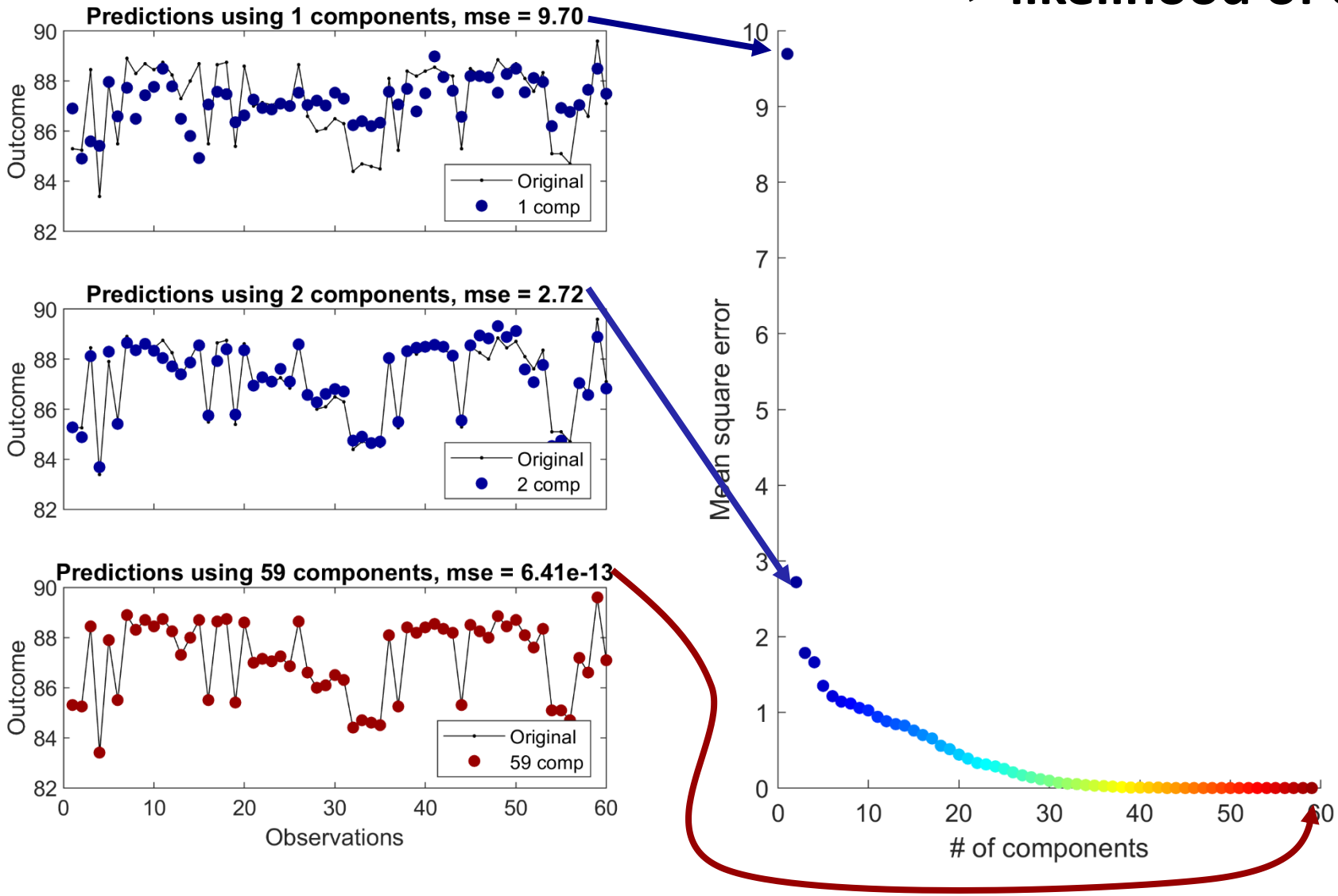


Two components

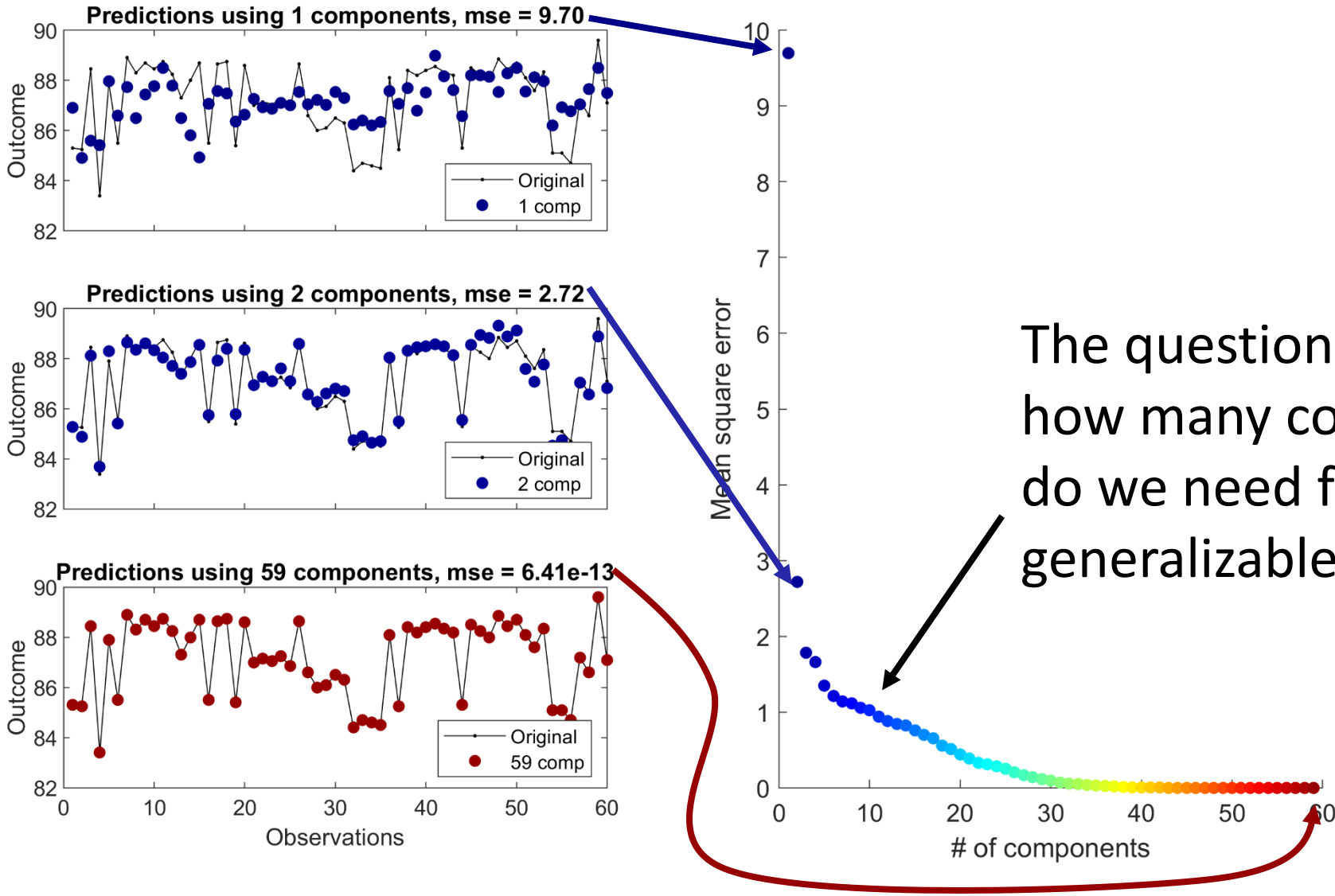


More components:

- Low error
- > likelihood of overfitting



For partial least squares regression, within sample tests can lead to over fitting



The question is, how many components do we need for a generalizable model?

How do we avoid over fitting with cross validation?

Cross-Validation

Definition: Using different samples to model and predict

- *hold-out*: you use the unique dataset you have to make random partitions, one to model and the other to predict

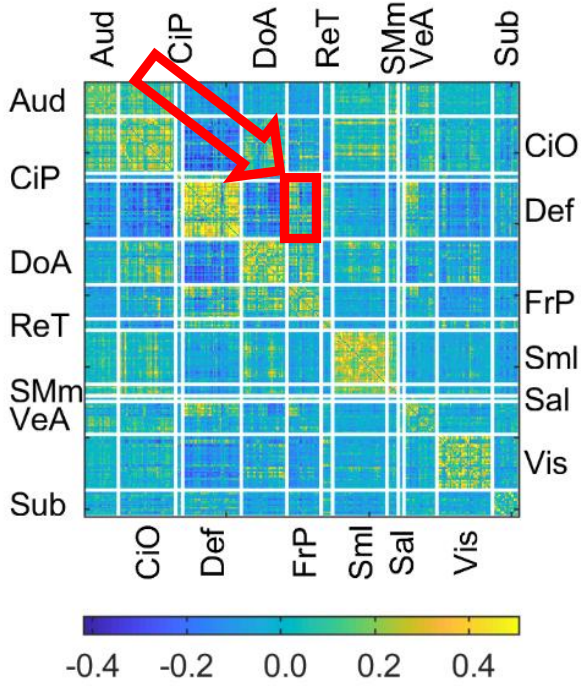
Other forms of out of sample sampling

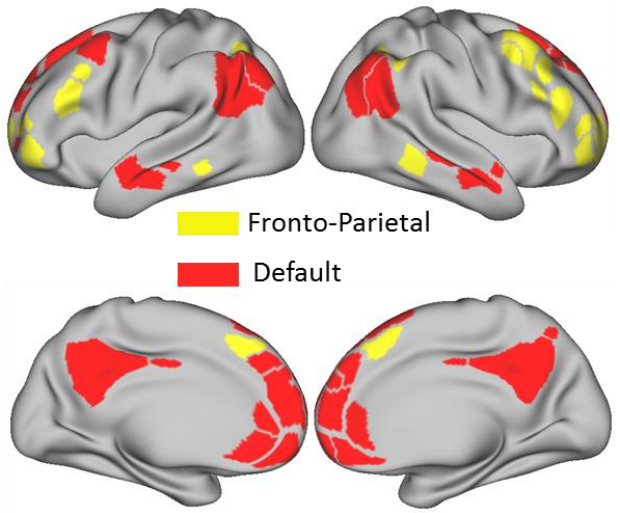
- **Bootstrapping** : random sampling with replacement

Let’s use an example to illustrate the problem of **overfitting** and how **hold-out cross validation can minimize it**

Imagine an “executive functioning” score is related to mean functional connectivity

The modeler does not know the model structure but it is given by a third order polynomial:

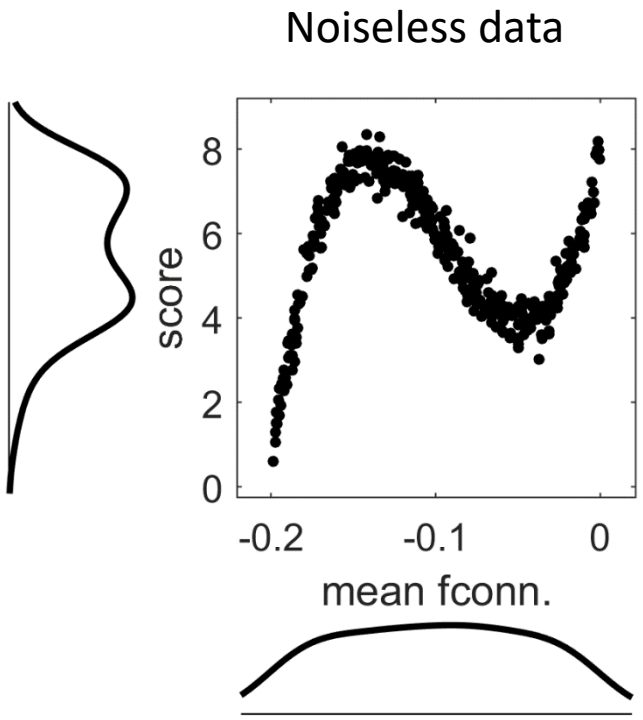




x = mean fconn between the Fronto-parietal and default networks

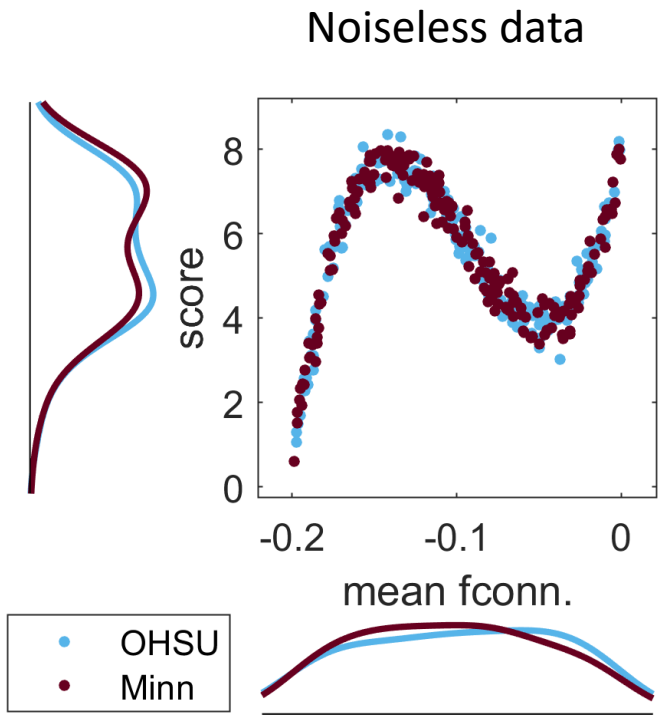
score$= p_0 + p_1 x + p_2 x^2 + p_3 x^3$

Data was measured on multiple participants

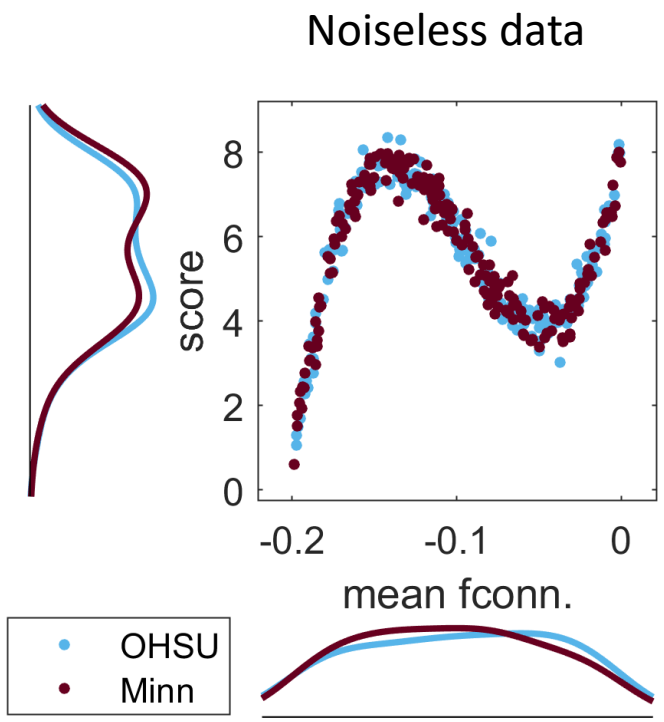


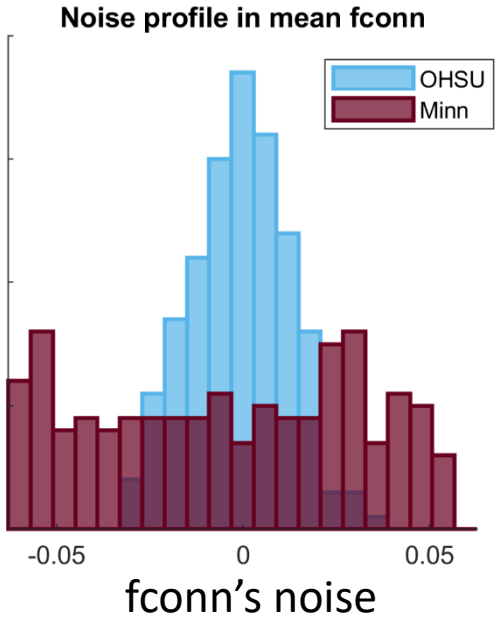
- Unique participant

However, data was collected on two sites

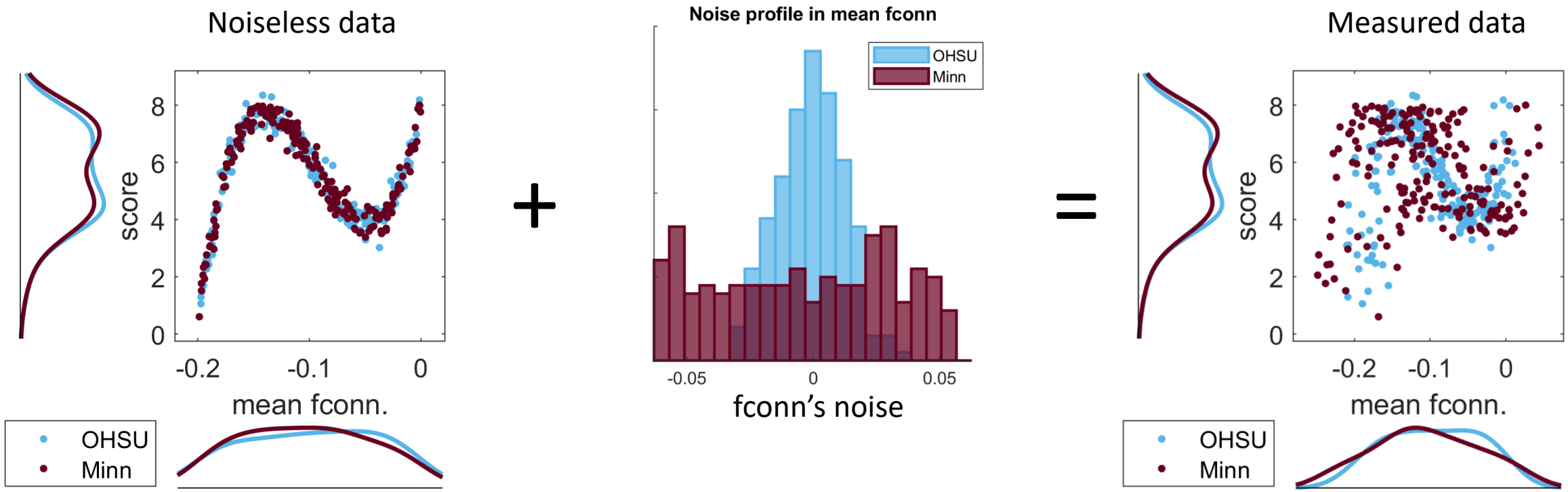


and each site has a different scanner's noise profile,

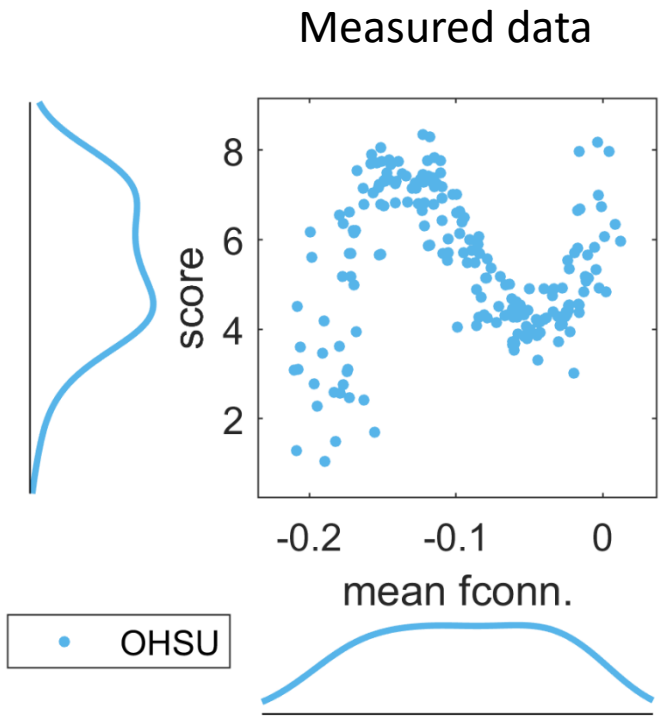




which leads to significant batch effects.



We, however, only have access to OHSU data.



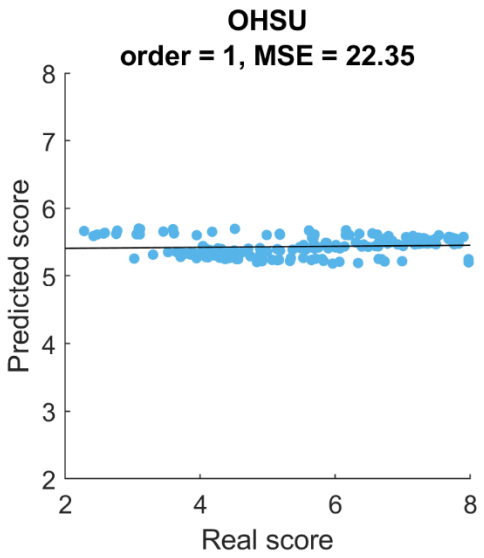
Modeling approach

- Predict executive functioning score based on mean fconn using polynomials of different order
  - Starting from simplest to more complex models

- Estimate “goodness of the fit” (mean square errors in predictions)

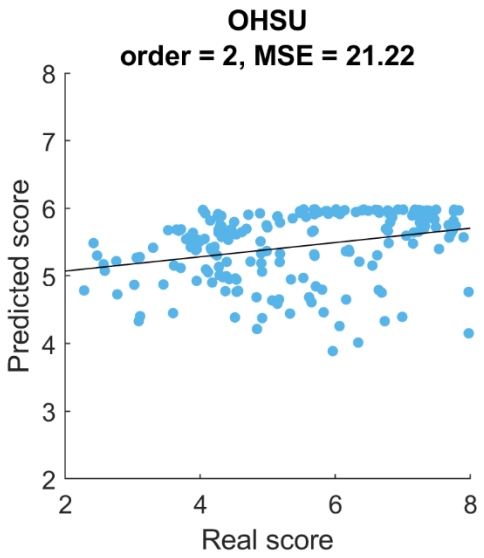
- Select the model with the “best fit” *i.e.*, lowest error

First order



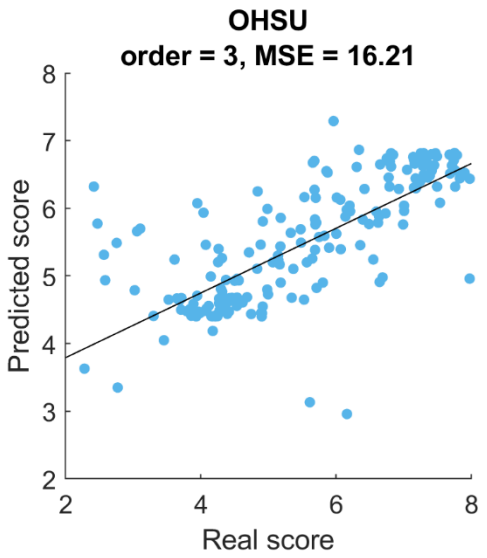
Polynomial order	Mean Square Error
	OHSU
1	22.35

Second order



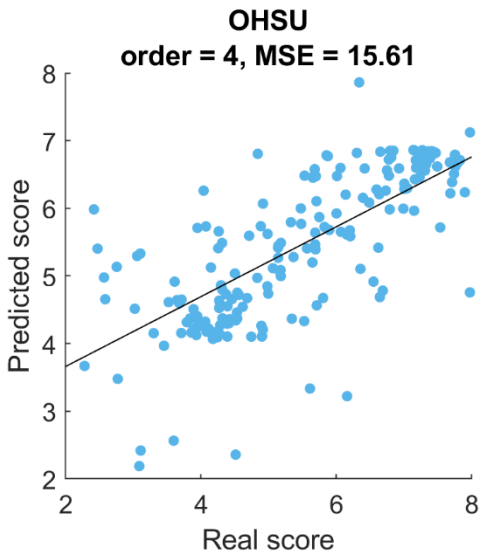
Polynomial order	Mean Square Error
	OHSU
1	22.35
2	21.22

Third order



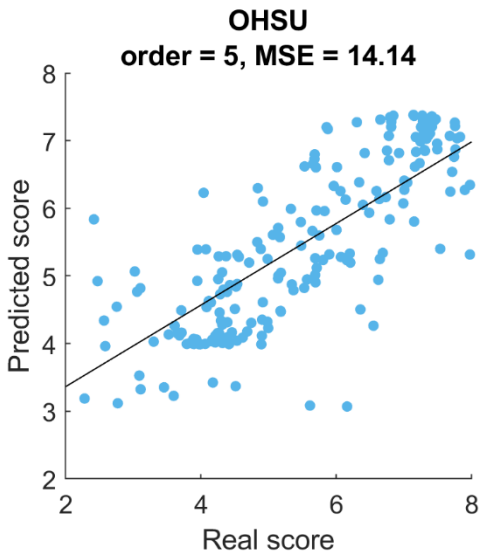
Polynomial order	Mean Square Error
	OHSU
1	22.35
2	21.22
3	16.21

Fourth order



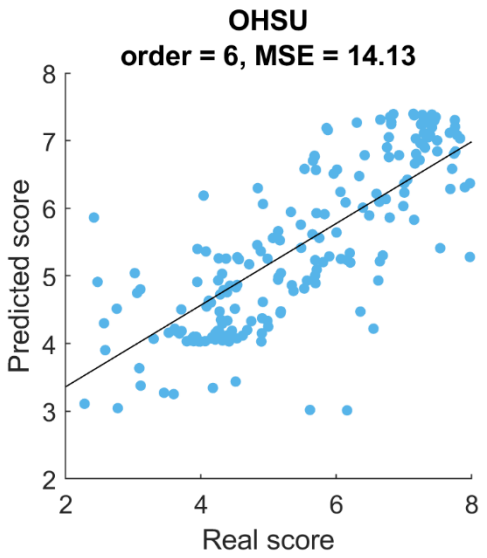
Polynomial order	Mean Square Error
	OHSU
1	22.35
2	21.22
3	16.21
4	15.61

Fifth order



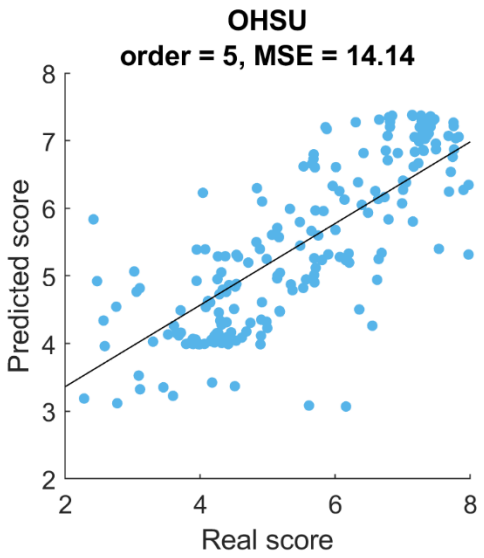
Polynomial order	Mean Square Error
	OHSU
1	22.35
2	21.22
3	16.21
4	15.61
5	14.14

Sixth order



Polynomial order	Mean Square Error
	OHSU
1	22.35
2	21.22
3	16.21
4	15.61
5	14.14
6	14.13

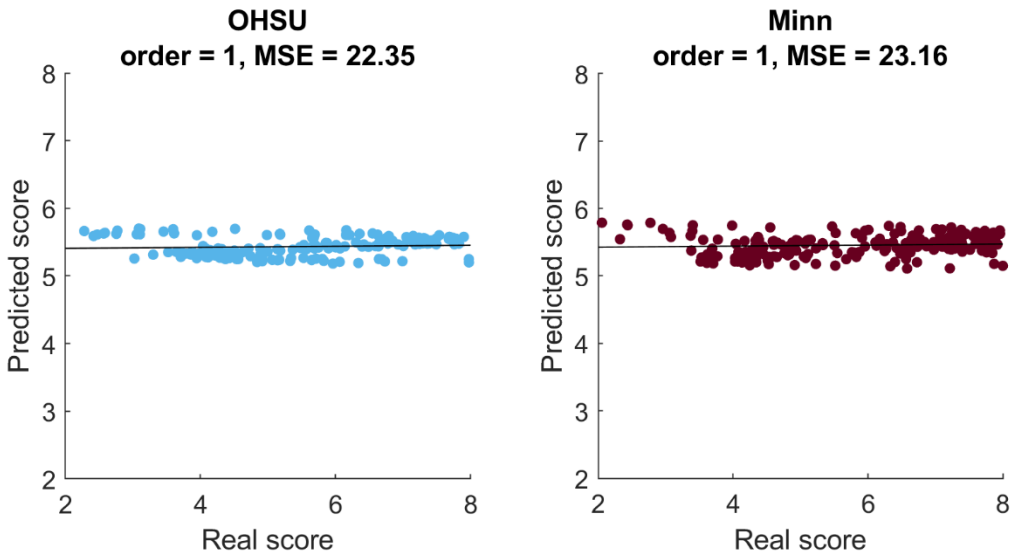
Fifth order seems to be the best fit



Polynomial order	Mean Square Error
	OHSU
1	22.35
2	21.22
3	16.21
4	15.61
5	14.14
6	14.13

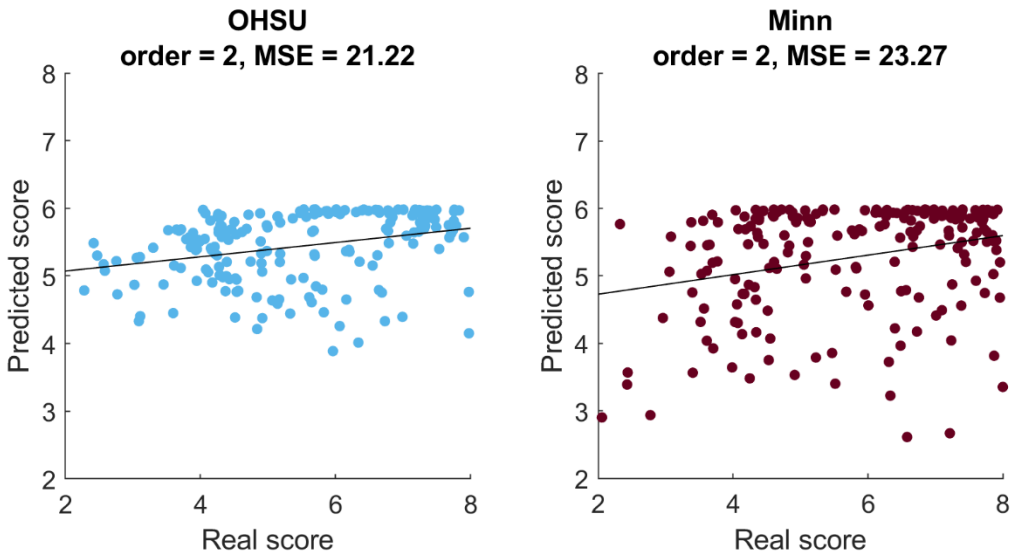
Let’s use OHSU’s models on Minn’s data

First order



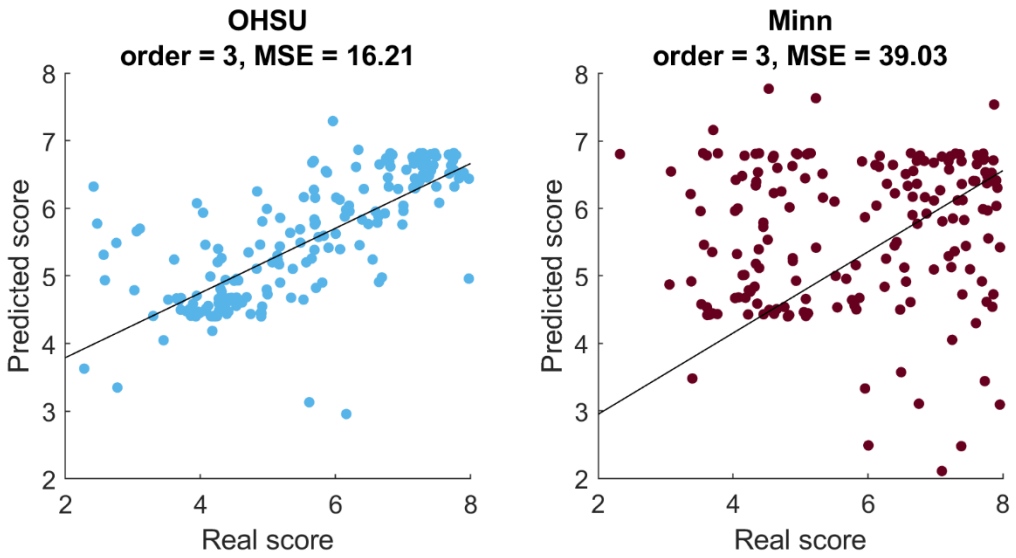
Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16

Second order



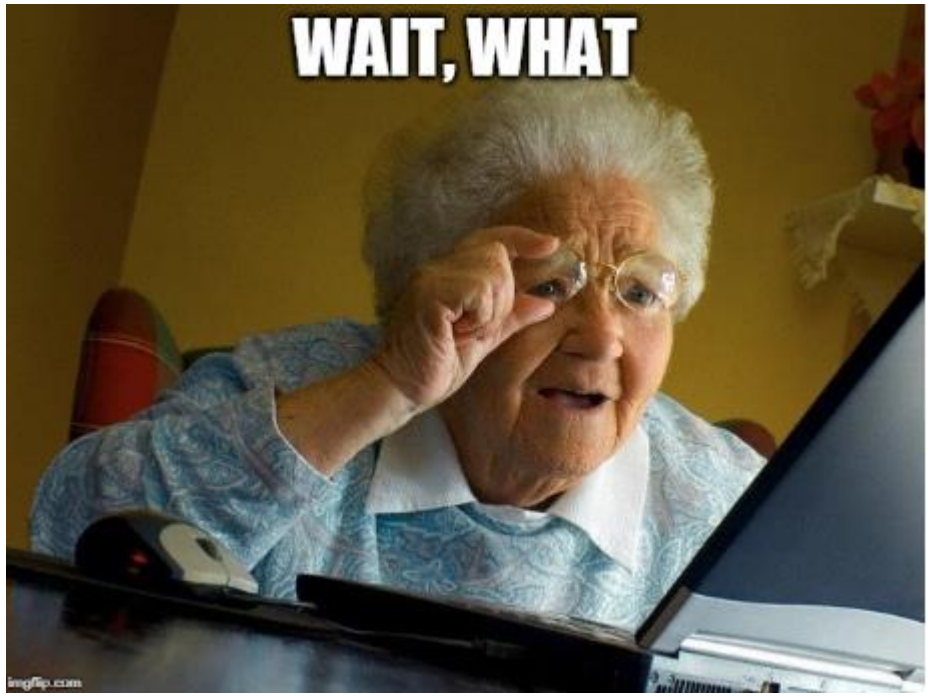
Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27

Third order



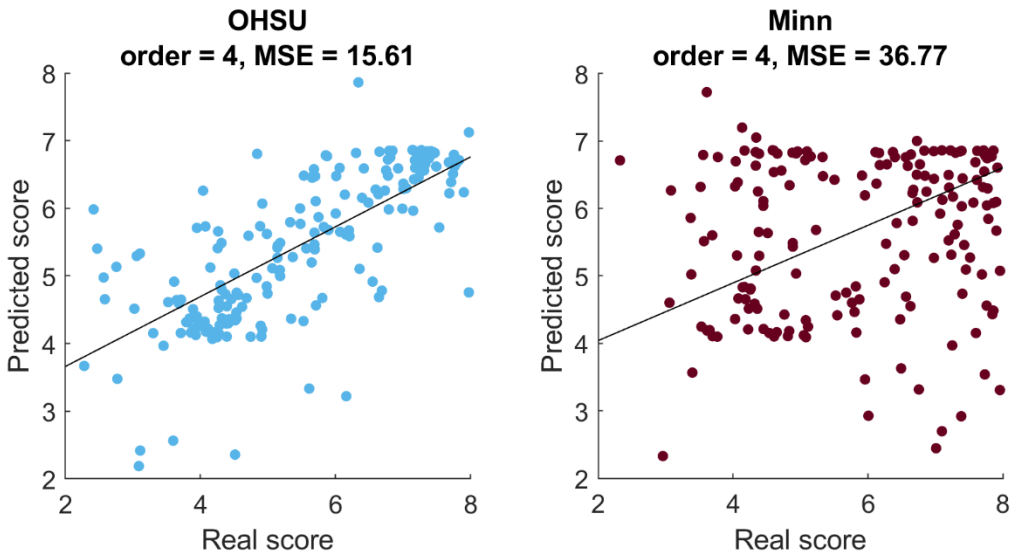
Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03

Third order



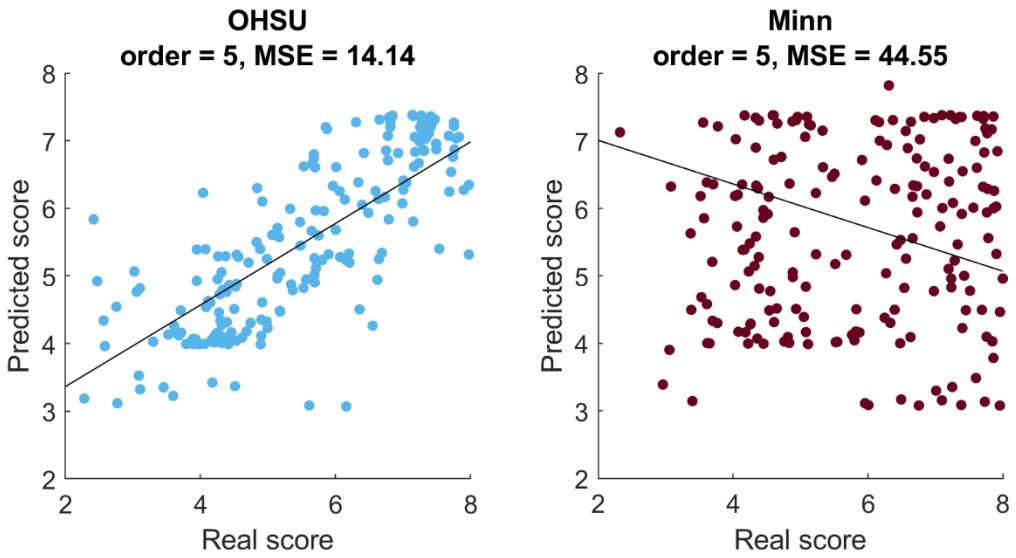
Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03

Fourth order



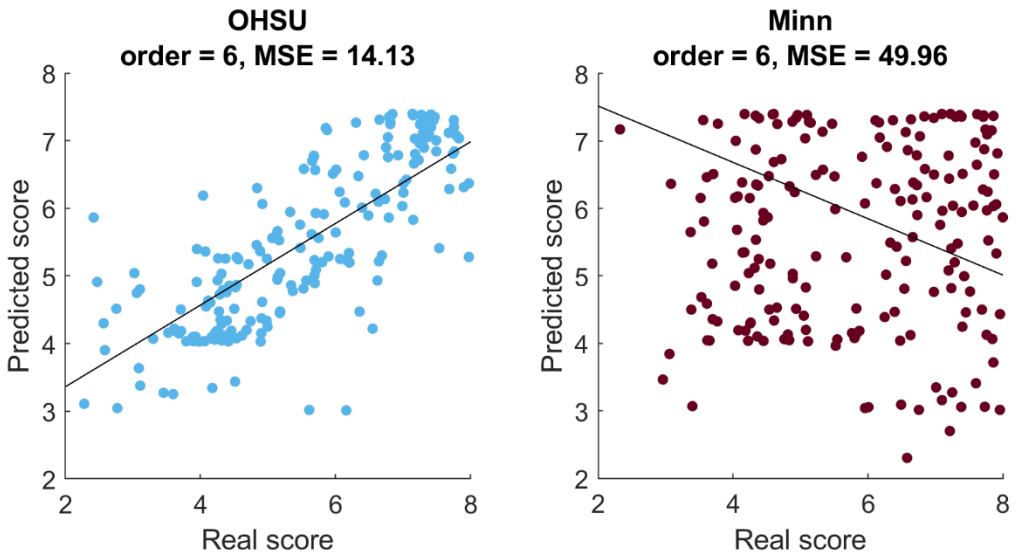
Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03
4	15.61	36.77

Fifth order



Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03
4	15.61	36.77
5	14.14	44.55

Sixth order

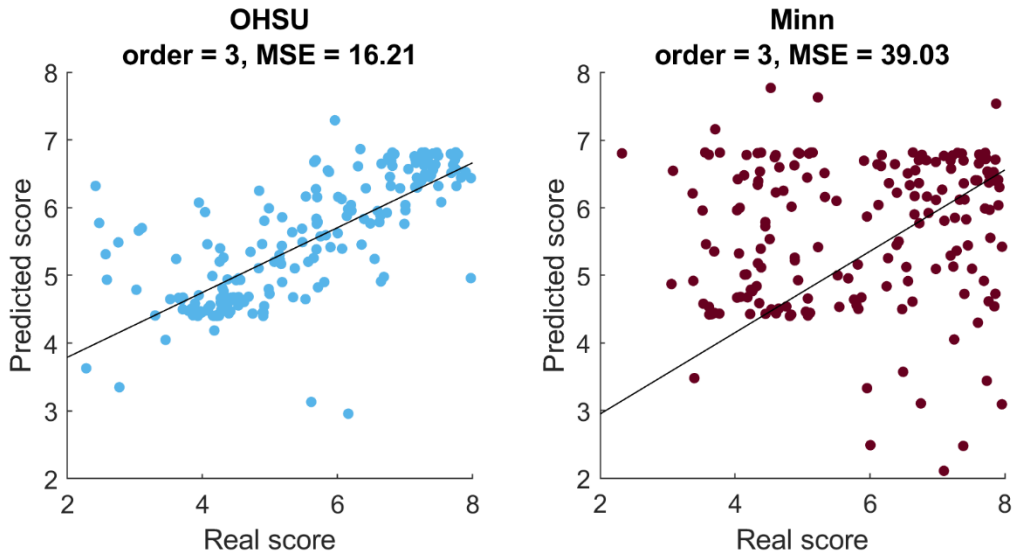


Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03
4	15.61	36.77
5	14.14	44.55
6	14.13	49.96

Take-home message

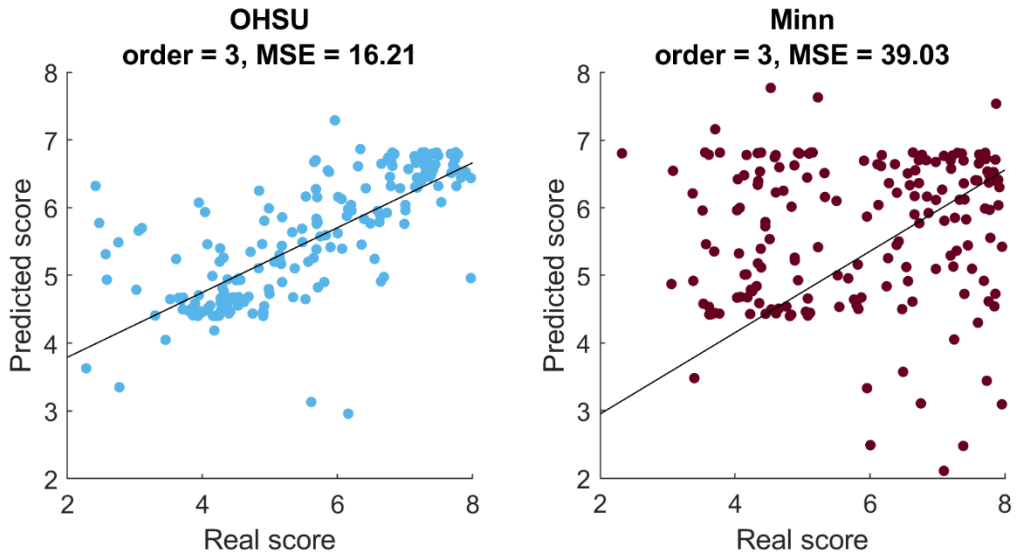
Testing performance on the same data used to obtain a model leads to overfitting. **Do not do it.**

How to know that the best model is a third order polynomial?



Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03
4	15.61	36.77
5	14.14	44.55
6	14.13	49.96

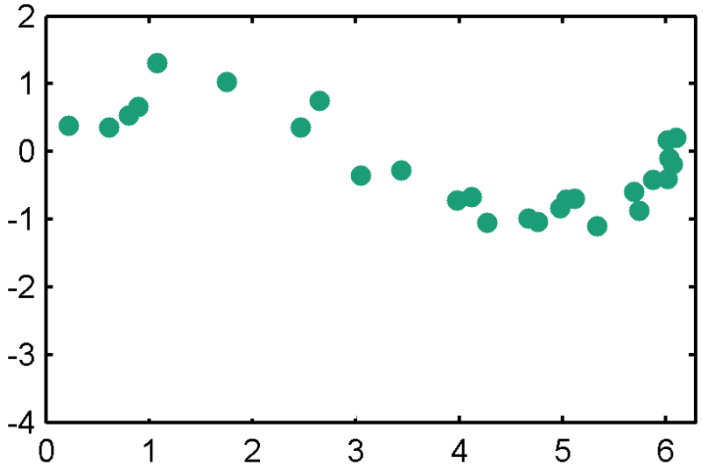
How to know that the best model is a third order polynomial?



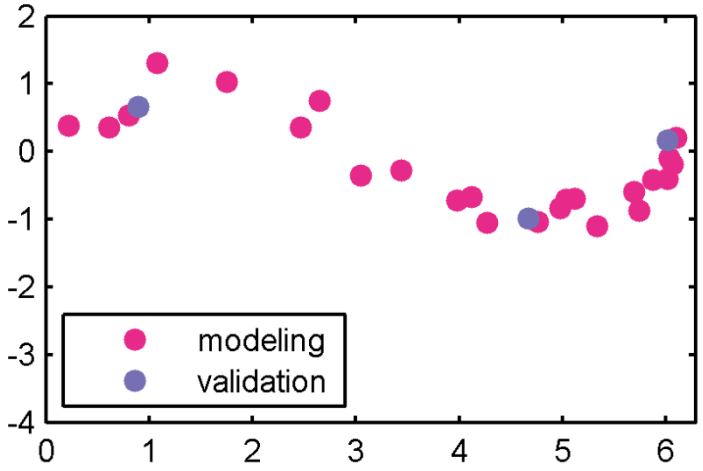
Polynomial order	Mean Square Error	
	OHSU	Minn
1	22.35	23.16
2	21.22	23.27
3	16.21	39.03
4	15.61	36.77
5	14.14	44.55
6	14.13	49.96

Use hold-out cross-validation!

Let’s use hold-out cross-validation to fit the most generalizable model for this data set



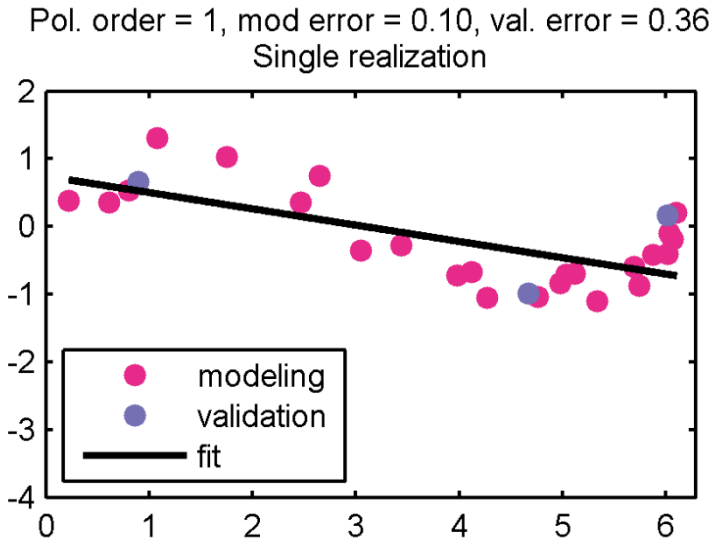
Make two partitions: Let's use 90% of the sample for modeling and **hold** 10% **out** for testing



Use the partition **modeling** to fit the simplest model.

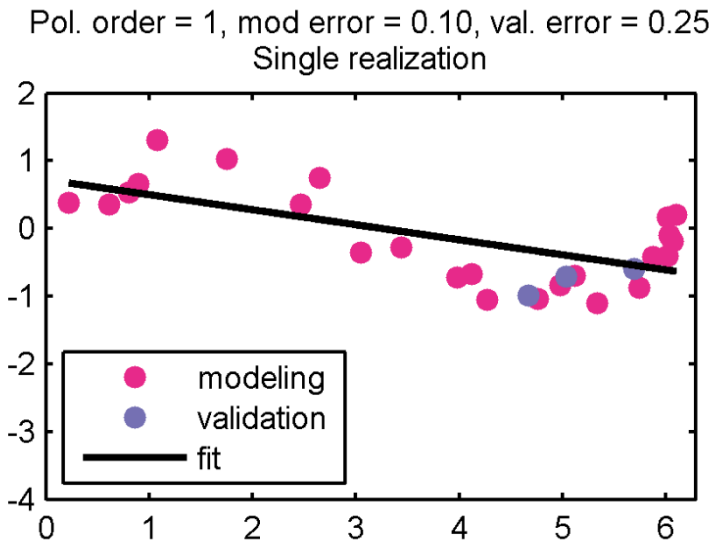
Then predict in-sample and out-sample data

A reasonable cost function is the mean of the sum of squares's residuals

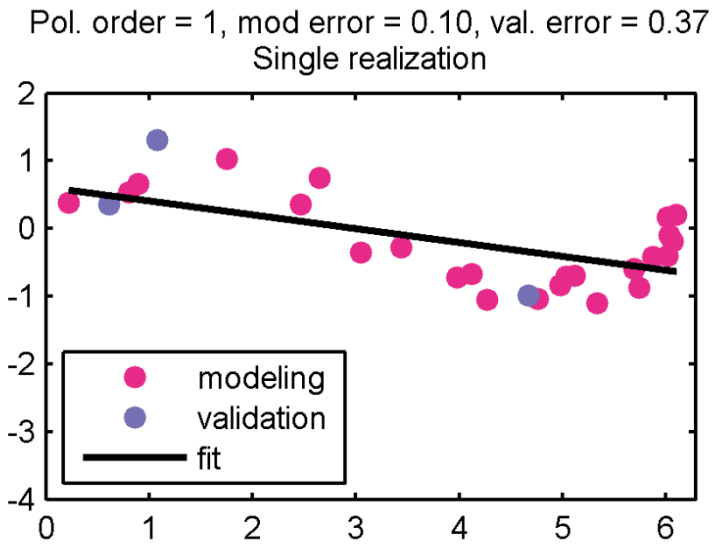


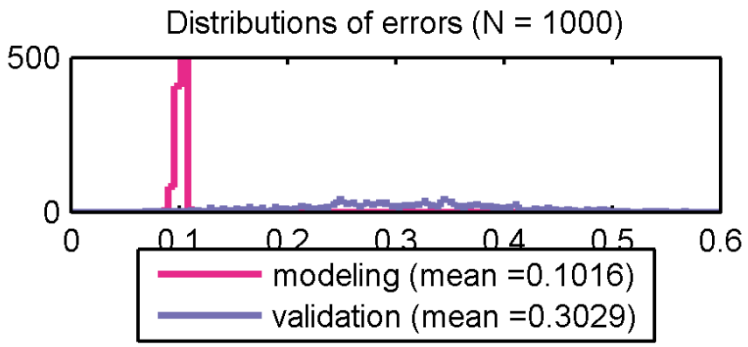
Resample and repeat

Keep track of the errors.



Repeat N times

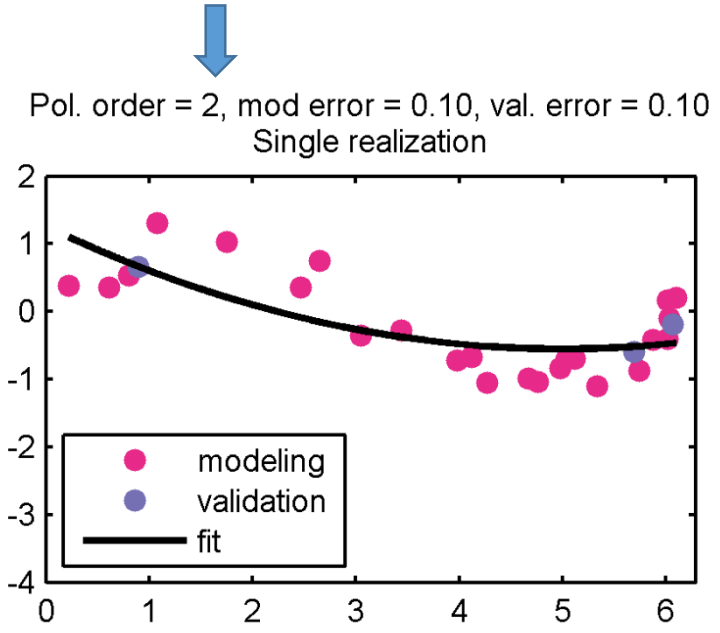


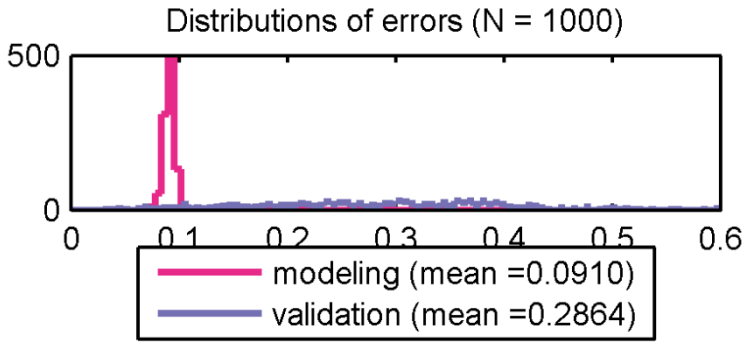


Increase model complexity,

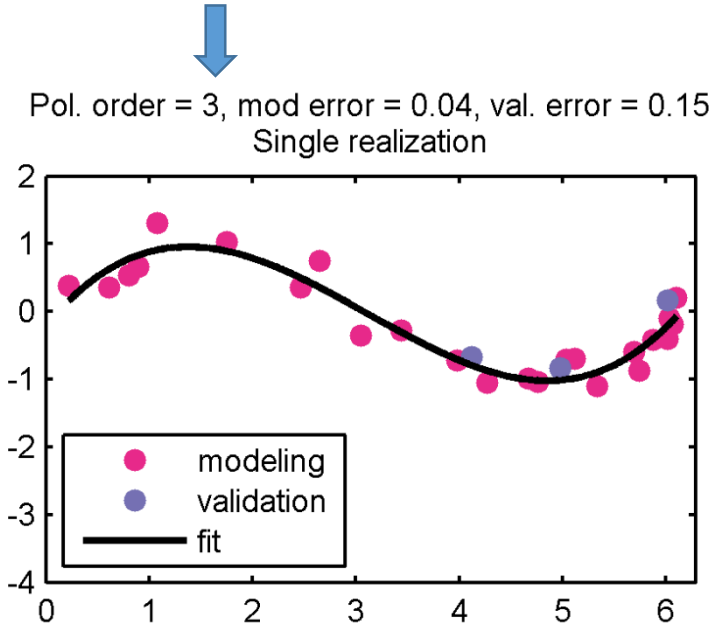
Increase order complexity

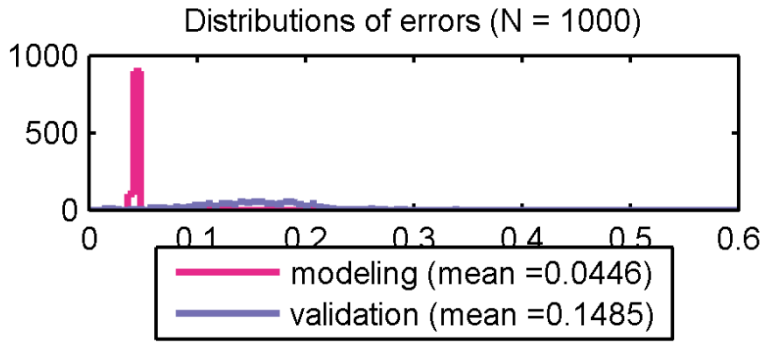
Keep track of the errors.



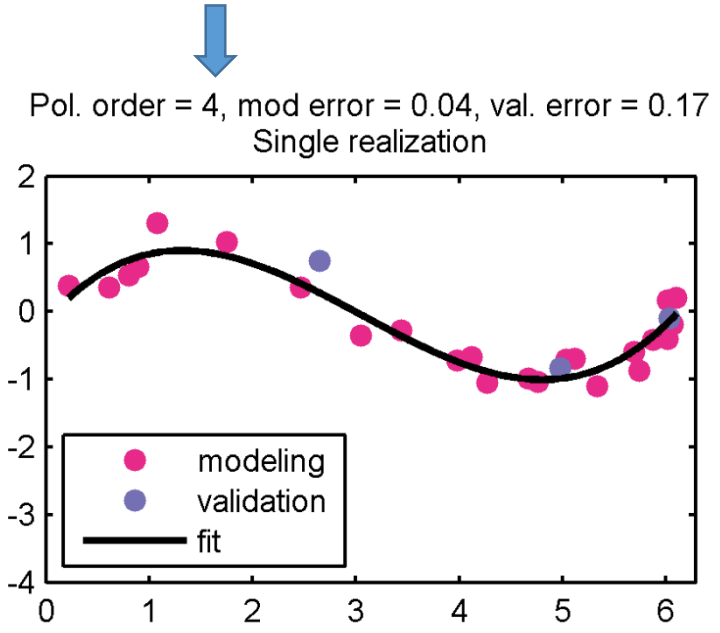


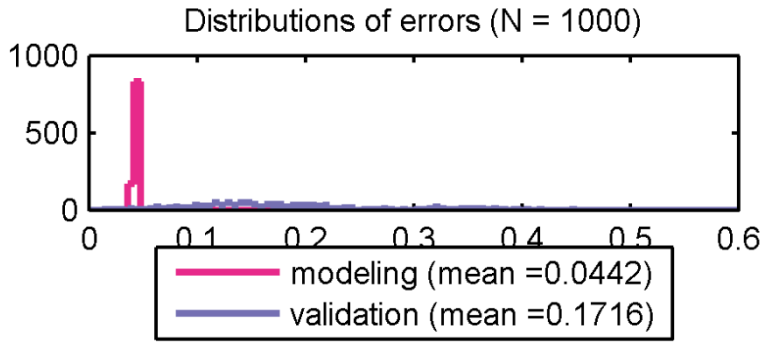
Third order


Pol. order = 3, mod error = 0.04, val. error = 0.15
Single realization
2
1
0
-1
-2
-3
-4
0
1
2
3
4
5
6
modeling
validation
fit


Distributions of errors (N = 1000)
1000
500
0
0
0.1
0.2
0.3
0.4
0.5
0.6
modeling (mean =0.0446)
validation (mean =0.1485)

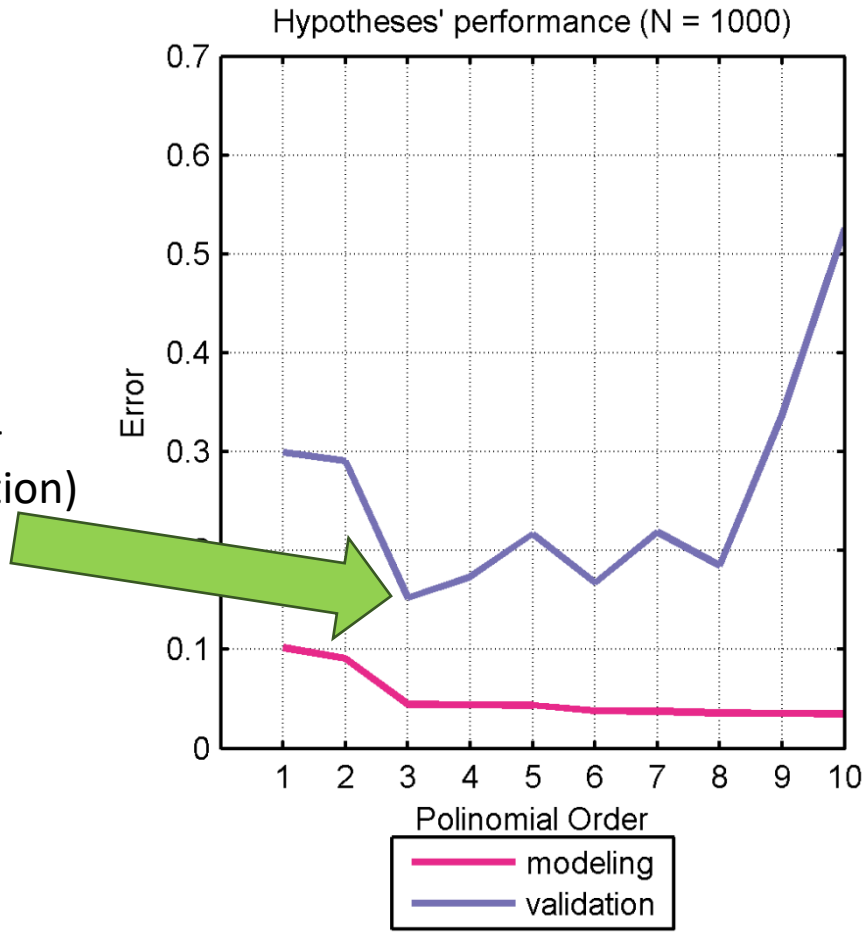
Fourth order





Visualize results

Pick the best (lowest out-of-sample prediction)



Notice how the in-sample (modeling) error decreases as order increases: **OVERFITTING**

Take-home message

Cross-validation is a useful tool towards predictive modeling.

Partial-least squares regression requires cross-validation for predictive modeling to avoid overfitting

Generating Null hypothesis data

Why is it important to generate a null distribution?

How do you know that your model behaves better than chance?

- What is chance in the context of modeling and hold-out cross-validation?

Let’s suppose this is your data

Original data

$$9x_1 - 7x_2 + \cdots - 4x_n = 21$$
$$-x_1 + 9x_2 + \cdots + 2x_n = 19$$
$$2x_1 + 7x_2 + \cdots + 2x_n = 77$$
$$1x_1 - 6x_2 + \cdots + 1x_n = 20$$
$$7x_1 - 2x_2 + \cdots - 9x_n = 62$$

Make two random partitions: modeling and validation

Original data	Modeling	Validation
$9x_1 - 7x_2 + \cdots - 4x_n = 21$	$9x_1 - 7x_2 + \cdots - 4x_n = 21$	
$-x_1 + 9x_2 + \cdots + 2x_n = 19$		$-x_1 + 9x_2 + \cdots + 2x_n = 19$
$2x_1 + 7x_2 + \cdots + 2x_n = 77$	$2x_1 + 7x_2 + \cdots + 2x_n = 77$	
$1x_1 - 6x_2 + \cdots + 1x_n = 20$	$1x_1 - 6x_2 + \cdots + 1x_n = 20$	
$7x_1 - 2x_2 + \cdots - 9x_n = 62$		$7x_1 - 2x_2 + \cdots - 9x_n = 62$

Randomize predictor and outcomes in the partition used for modeling

Original data

$9x_1 - 7x_2 + \cdots - 4x_n = 21$

$-x_1 + 9x_2 + \cdots + 2x_n = 19$

$2x_1 + 7x_2 + \cdots + 2x_n = 77$

$1x_1 - 6x_2 + \cdots + 1x_n = 20$

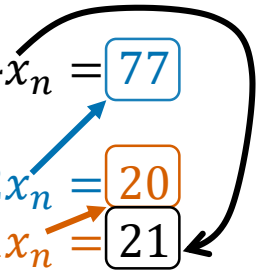
$7x_1 - 2x_2 + \cdots - 9x_n = 62$

Modeling

$9x_1 - 7x_2 + \cdots - 4x_n = 77$

$2x_1 + 7x_2 + \cdots + 2x_n = 20$

$1x_1 - 6x_2 + \cdots + 1x_n = 21$



Validation

$-x_1 + 9x_2 + \cdots + 2x_n = 19$

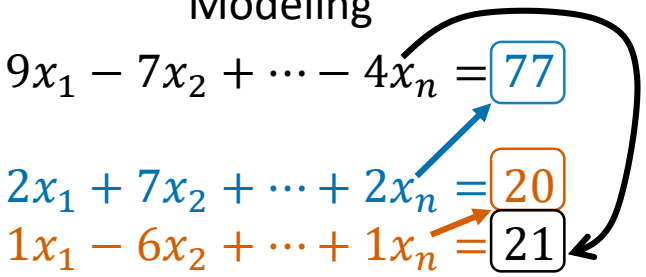
$7x_1 - 2x_2 + \cdots - 9x_n = 62$

Estimate out-of-sample performance:

Original data

$9x_1 - 7x_2 + \cdots - 4x_n = 21$
$-x_1 + 9x_2 + \cdots + 2x_n = 19$
$2x_1 + 7x_2 + \cdots + 2x_n = 77$
$1x_1 - 6x_2 + \cdots + 1x_n = 20$
$7x_1 - 2x_2 + \cdots - 9x_n = 62$

Modeling



Validation

$-x_1 + 9x_2 + \cdots + 2x_n = 19$

$7x_1 - 2x_2 + \cdots - 9x_n = 62$

- Calculate the model in the partition “Modeling”
- Predict outcome on the partition “Validation”
- Estimate “goodness of the fit”: mean square error

Repeat and keep track of the errors

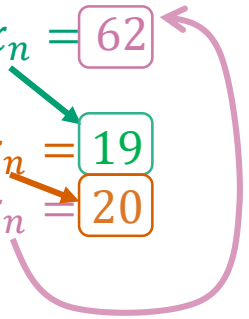
Original data

$9x_1 - 7x_2 + \cdots - 4x_n = 21$
$-x_1 + 9x_2 + \cdots + 2x_n = 19$
$2x_1 + 7x_2 + \cdots + 2x_n = 77$
$1x_1 - 6x_2 + \cdots + 1x_n = 20$
$7x_1 - 2x_2 + \cdots - 9x_n = 62$

Modeling

$-x_1 + 9x_2 + \cdots + 2x_n = 62$

$1x_1 - 6x_2 + \cdots + 1x_n = 19$
$7x_1 - 2x_2 + \cdots - 9x_n = 20$



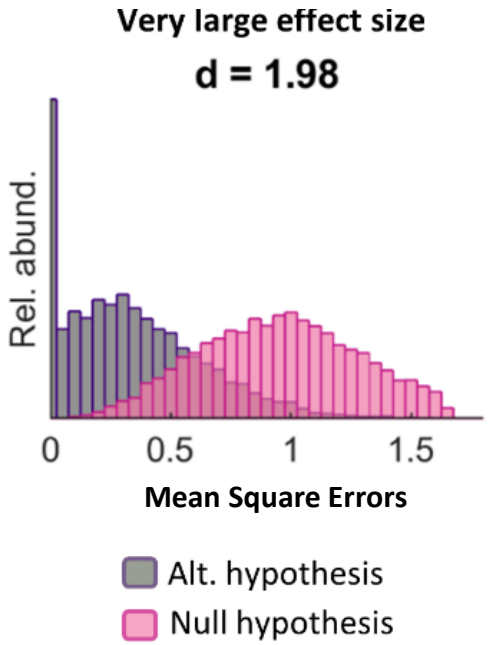
Validation

$9x_1 - 7x_2 + \cdots - 4x_n = 21$

$2x_1 + 7x_2 + \cdots + 2x_n = 77$

- Calculate the model in the partition “Modeling”
- Predict outcome on the partition “Validation”
- Estimate “goodness of the fit”: mean square error

Compare performance (mean squares error in out-of-sample data) to determine if your model predicts better than chance!



Example using Neuroimaging data
cross-validation, regularization and PLSR

fconn_regression tool

I’ll use as a case the study of cueing in freezing of gait in Parkinson’s disease

Freezing of gait, a pretty descriptive name, is an additional symptom present on some patients

Freezing can lead to falls, which adds an extra burden in Parkinson’s disease

https://en.wikipedia.org/wiki/Parkinson's_disease

http://parkinsonteam.blogspot.com/2011/10/prevencion-de-caidas-en-personas-con.html

Auditory cues, like beats at a constant rate, are an effective intervention to reduce freezing episodes in some patients

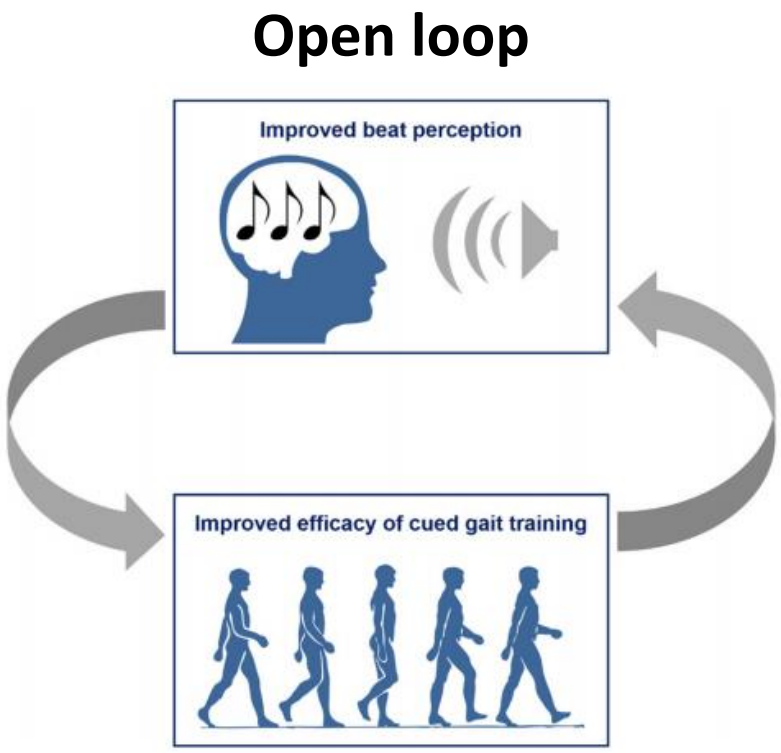


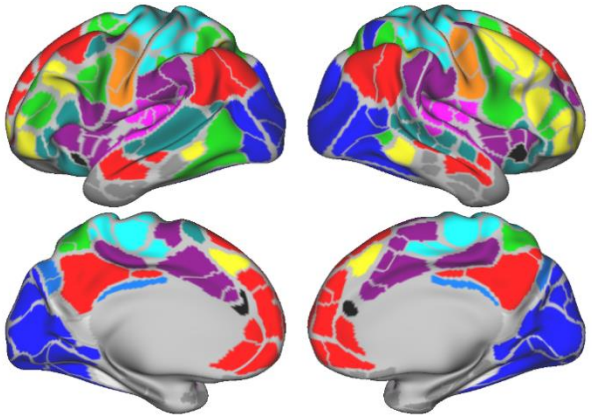
FIGURE 2 | Self-improving relationship between beat perception and gait training efficacy.

Ashoori A, Eagleman DM, Jankovic J. Effects of Auditory Rhythm and Music on Gait Disturbances in Parkinson's Disease [Internet]. Front Neurol 2015;

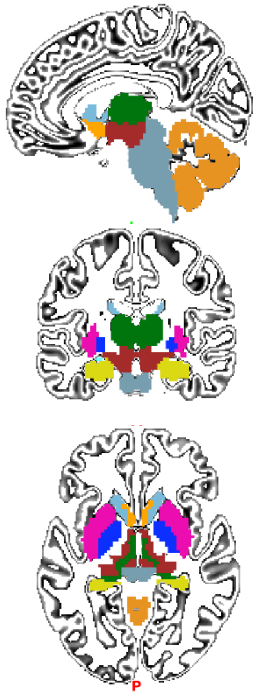
The goal of the study is to determine whether improvement after cueing can be predicted by resting state functional connectivity

Available data

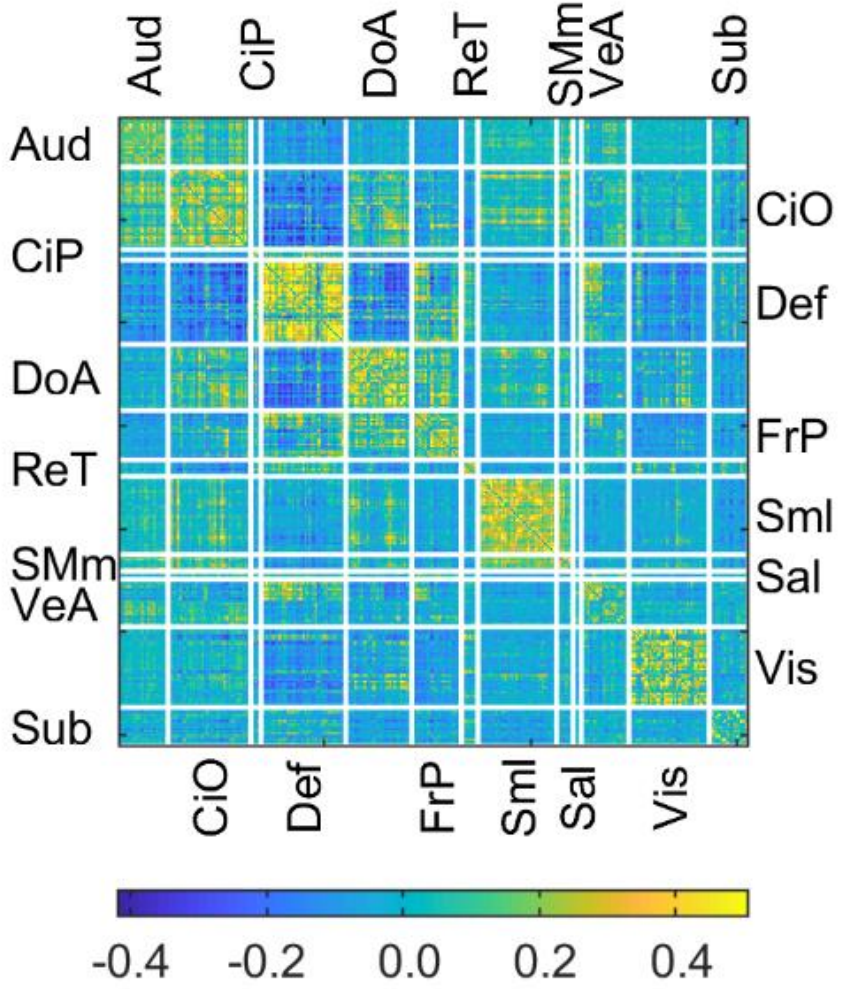
Resting state functional MRI



- Auditory (Aud, n=24)
- Cingulo-Opercular (CiO, n=40)
- Cingulo-Parietal (CiP, n=5)
- Default (Def, n=41)
- Dorsal Attention (DoA, n=32)
- Fronto-Parietal (FrP, n=24)
- Retrosplenial-Temporal (ReT, n=8)
- Somato-sensory lateral (Sml, n=38)
- Somato-sensory medial (SMm, n=8)
- Salience (Sal, n=4)
- Ventral-Attention (VeA, n=23)
- Visual (Vis, n=39)



- Cerebellum
- Thalamus
- Caudate
- Putamen
- Pallidum
- Brain Stem
- Hippocampus
- Amygdala
- Accumbens
- Diencephalon Ventral



Approach

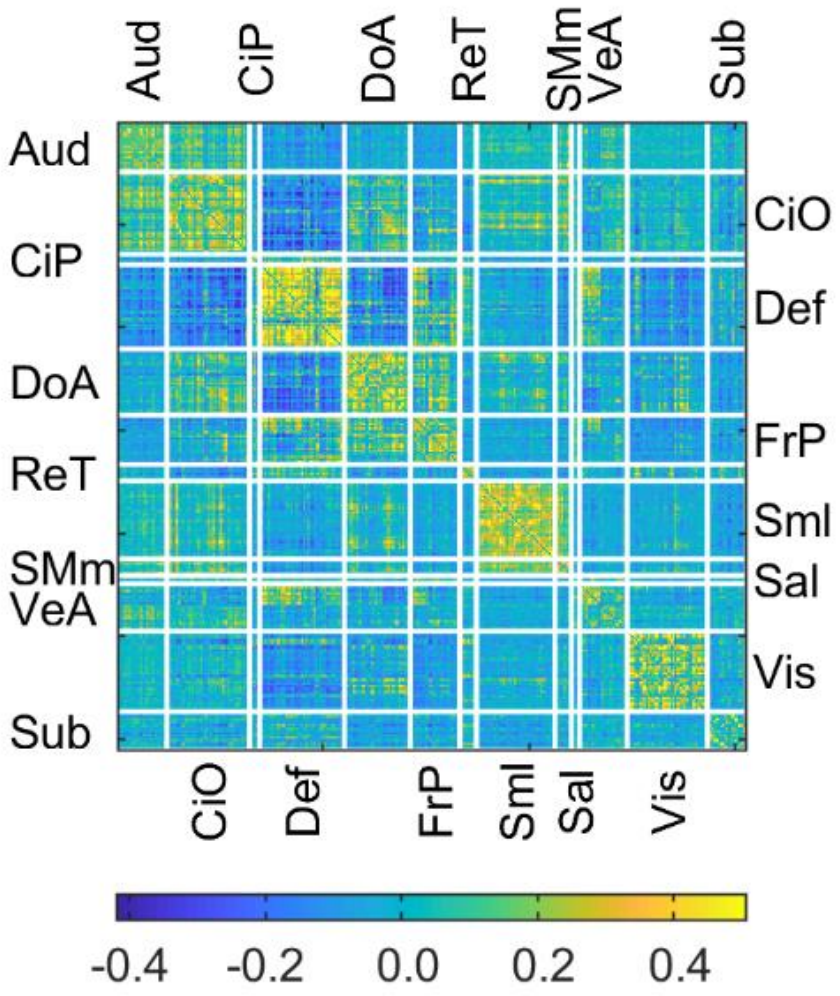
1. Calculate rs-fconn
   - Group data *per* functional network pairs: Default-Default, Default-Visual, …

2. Use PLSR and cross-validation to determine whether improvement can be predicted using connectivity from specific brain networks

3. Explore outputs

4. Report findings

First step is to calculate resting state functional connectivity and group data per functional system pairs



PLSR and cross-validation

Parameters

- Partition size
  - Hold-one out
  - Hold-three out
- How many components:
  - 2, 3, 4,…
- Number of repetitions
  - 100?, 500?,…
- Calculate null-hypothesis data
  - Number of repetitions: 10,000?

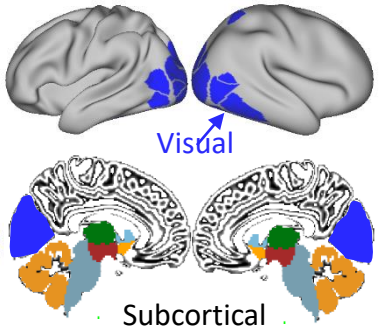
This can be done using the tool fconn_regression

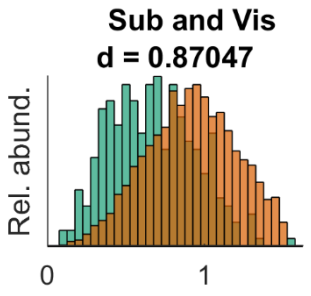
Comparing distribution of prediction errors for real *versus* null-hypotheses data

Sorted by Cohen effect size

Visual and subcortical

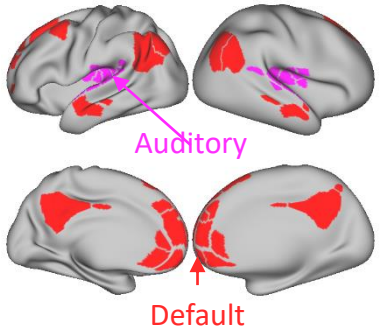
Effect size = 0.87

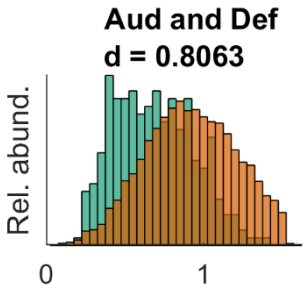




Auditory and default

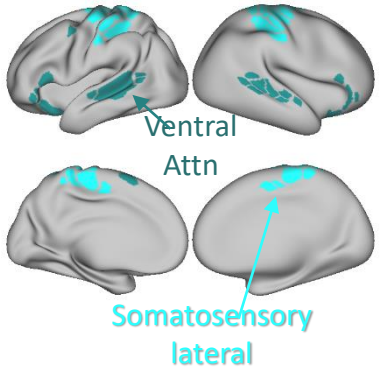
Effect size = 0.81

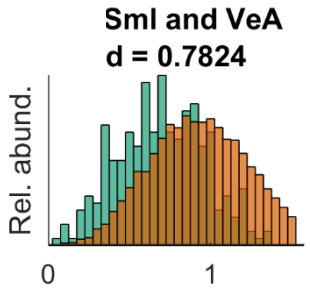




Somatosensory lateral and Ventral attention

Effect size = 0.78





We have a virtual machine and a working example

Let us know if you are interested in a break-out session

Topics

- Partial-least squares Regression
  - Feature Selection
  - Cross-Validation
  - Null Distribution/Permutations
  - An Example
- Regularization
  - Truncated singular value decomposition
  - Connectotyping: model based functional connectivity
  - Example: models that generalize across datasets!

Regularization

Truncated singular value decomposition

Measurements = # Variables

The system

$$4 = 2A$$

has a unique solution

$$A = 2$$

Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

<u>Select the solution with the lowest mean square error!</u>

$$\begin{bmatrix} 4.0 \\ 3.9 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.1 \end{bmatrix} A$$

$$y = xA$$

Using linear algebra (***x*** **pseudo-inverse**)

$$A = (x'x)^{-1}x'y$$

$$A \approx 1.9286$$

This A minimizes $\sum \text{residuals}^2$

Measurements < # Variables

What about (real) limited data:

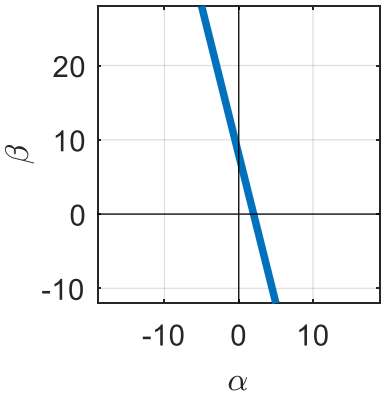
$$8 = 4\alpha + \beta$$

There are 2 variables (α and β) and 1 measurements.

Solving the system:

$$8 - 4\alpha = \beta$$

All the points on $\beta = 8 - 4\alpha$ solve the system.



In other words, there is an infinite number of solutions!

What if you can’t reduce the number of features?

Regularization is a powerful approach to handle this kind of problems (ill-posed systems)

We know that the pseudo-inverse offers the optimal solution (lowest least squares) for systems with more measurements than observations

 # Measurements > # Variables

What about repeated measurements (real data with noise)

$$\begin{aligned} 4.0 &= 2.0A \rightarrow A = 2.00 \\ 3.9 &= 2.1A \rightarrow A \approx 1.86 \end{aligned}$$

Select the solution with the lowest mean square error!

$$\begin{bmatrix} 4.0 \\ 3.9 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.1 \end{bmatrix} A$$

$$y = xA$$

Using linear algebra (**x pseudo-inverse**)

$$A = (x'x)^{-1}x'y$$

$$A \approx 1.9286$$

This A minimizes $\sum \text{residuals}^2$

We can use the pseudo-inverse to calculate a solution in systems with more measurements than observations

Example: Imagine a given outcome can be predicted by 379 variables,…

1) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

And that you have 163 observations:

1) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

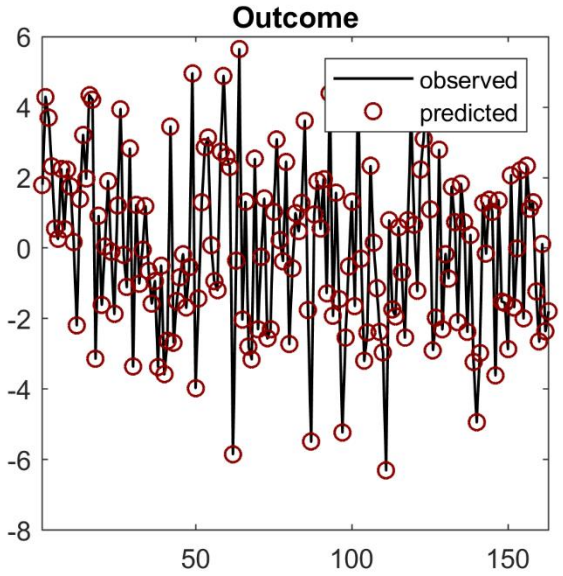
2) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

3) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

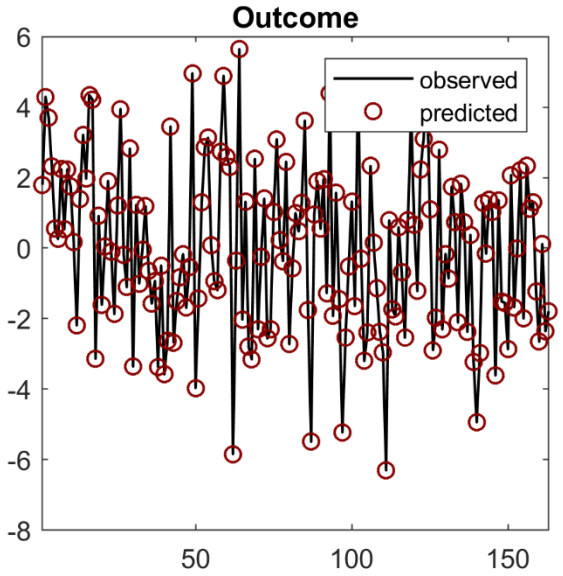
...

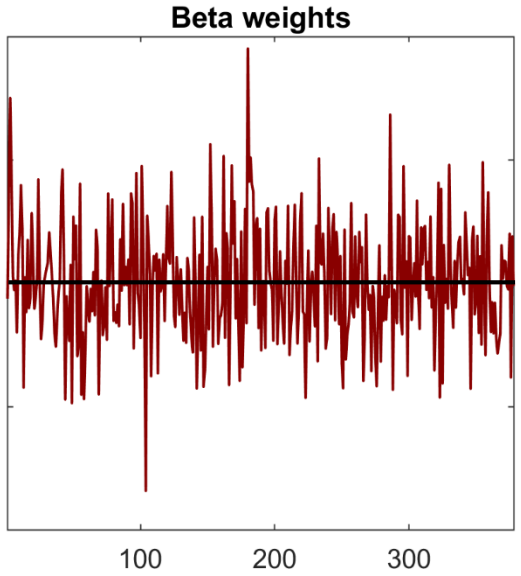
163) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

Using the pseudo-inverse you can obtain a solution with high predictability



Using the pseudo-inverse you can obtain a solution with high predictability





This solution, however, is problematic:

*unstable beta weights
*over fitting
*not applicable to outside dataset

What does “unstable beta weights” mean?

Let’s suppose age and weight are two variables used in your model

For one participant you used

- Age: 10.0 years
- Weight: 70 pounds
- Corresponding outcome: “score” of **3.7**

There was, however, an error in data collection and the real values are:

- Age: 10.5 years
- Weight: 71 pounds

Updating predictions in the same model

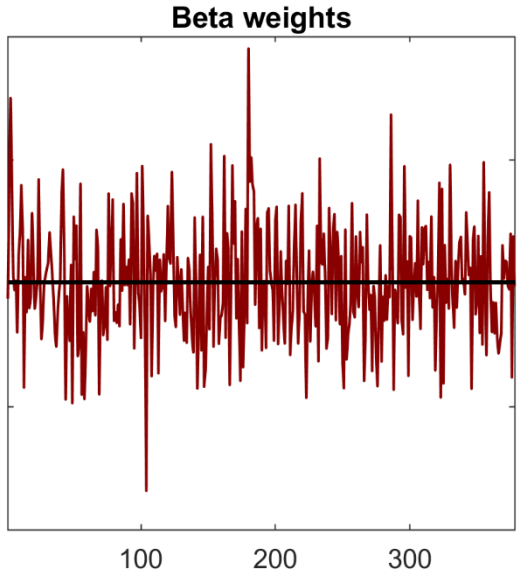
Let's suppose age and weight are two variables used in your model

For one participant you used

- Age: 10.0 years
- Weight: 70 pounds
- Corresponding outcome: "score" of **3.7**

There was, however, an error in data collection and the real values are:

- Age: 10.5 years
- Weight: 71 pounds



Stable beta-weights:

score ~ 3.9

Unstable beta weights:

score ~ -344,587.42

What is the best solutions for the system?

1) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

2) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

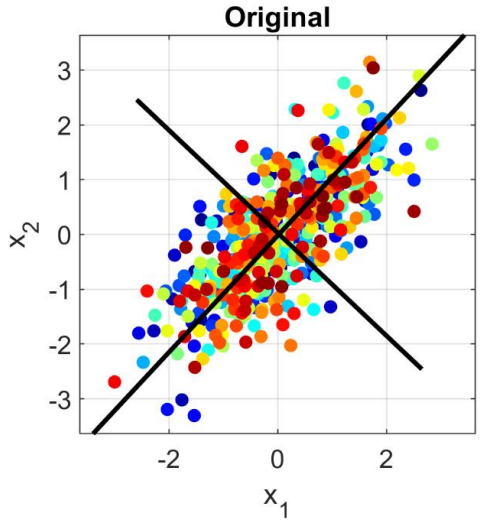
3) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

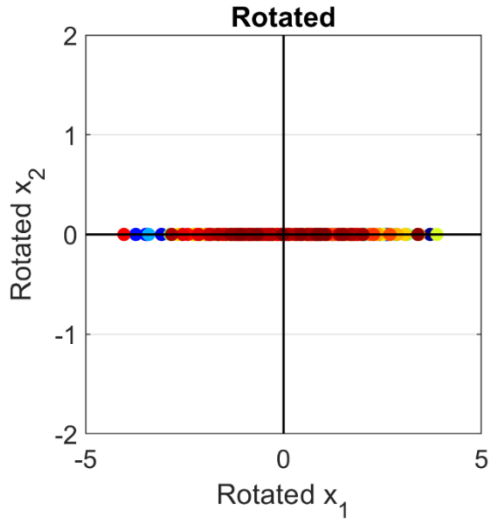
...

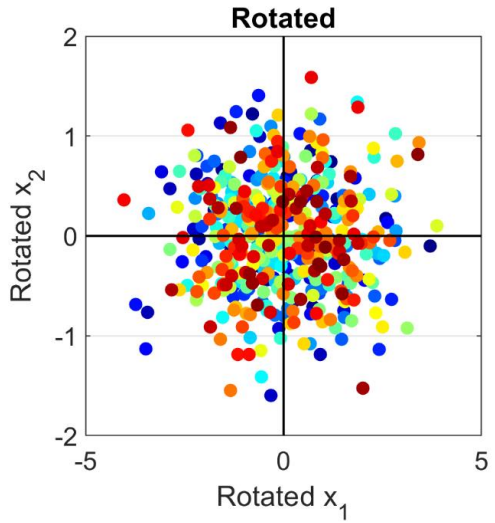
163) $y = \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_{379} x_{379}$

$$y = X\beta$$

Remember the PCA section?







We said that we can rotate X (the data) to find optimal projections

We can use different number of axis

Adding more axis leads to:

- More explained variance
- More over-fitting

In truncated singular value decomposition, we follow a similar approach

- Decompose X in such a way that we can explore effect of inclusion/exclusion of components (singular value decomposition)

$$X = U\Sigma V^T$$

$$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & \sigma_M \end{bmatrix},$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0.$$

The smaller singular values of X are more unstable (susceptible to noise)

In truncated singular value decomposition, we follow a similar approach

- Decompose X in such a way that we can explore effect of inclusion/exclusion of components (singular value decomposition)

- Make a new X **truncating** some components

$$X = U\Sigma V^T$$

$$\Sigma_{truncated} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & \cdots & 0 \end{bmatrix},$$

$$X_{truncated} = U\Sigma_{truncated}V^T$$

In truncated singular value decomposition, we follow a similar approach

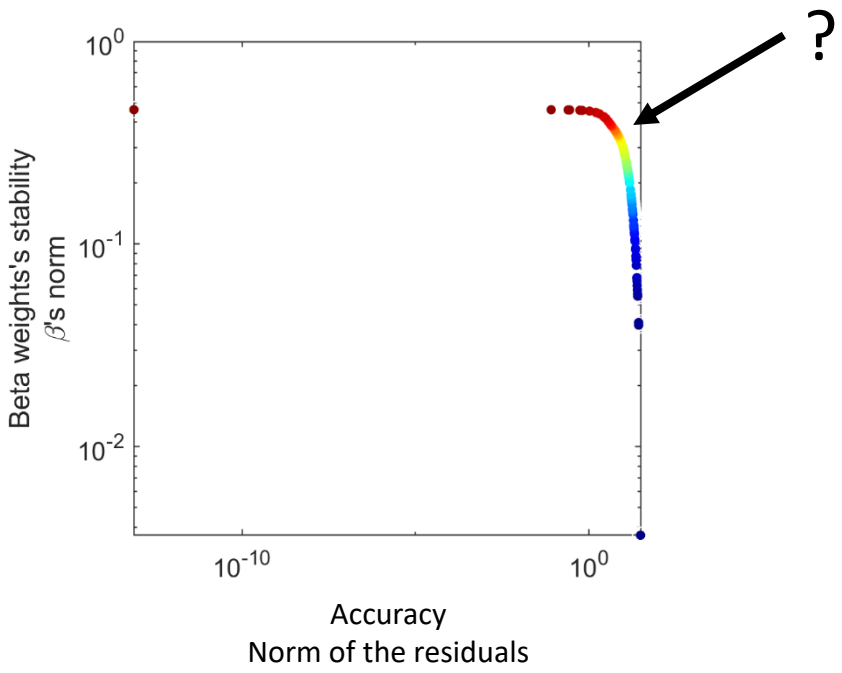
- Decompose X in such a way that we can explore effect of inclusion/exclusion of components (singular value decomposition)
- Make a new X **truncating** some components
- Solve the system plugging $X_{truncated}$ into the pseudo-inverse

$$\beta = \underbrace{(X'X)^{-1}X'}_{\text{Pseudo-inverse}}y$$

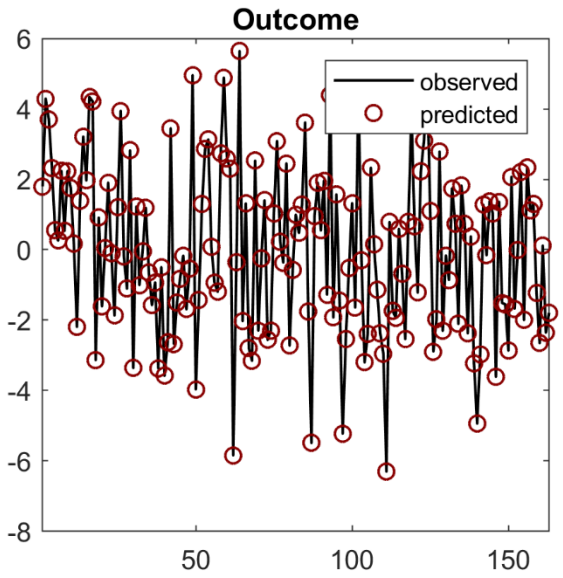
$$\beta_{truncated} = (X_{truncated}{}'X_{truncated})^{-1}X_{truncated}{}'y$$

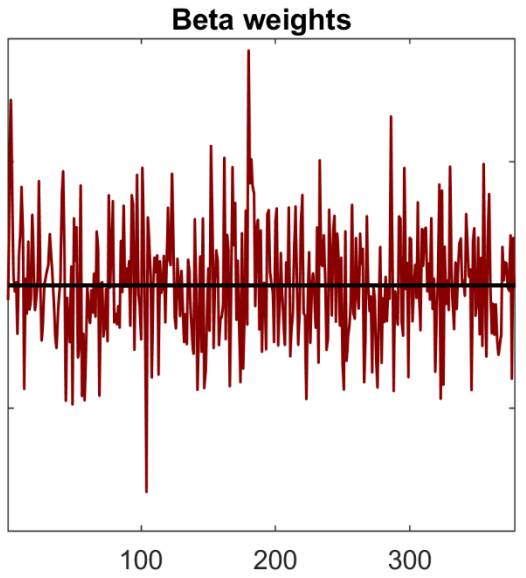
In truncated singular value decomposition, we follow a similar approach

- Decompose X in such a way that we can explore effect of inclusion/exclusion of components (singular value decomposition)
- Make a new X **truncating** some components
- Solve the system plugging $X_{truncated}$ into the pseudo-inverse
- Select the optimal number of components



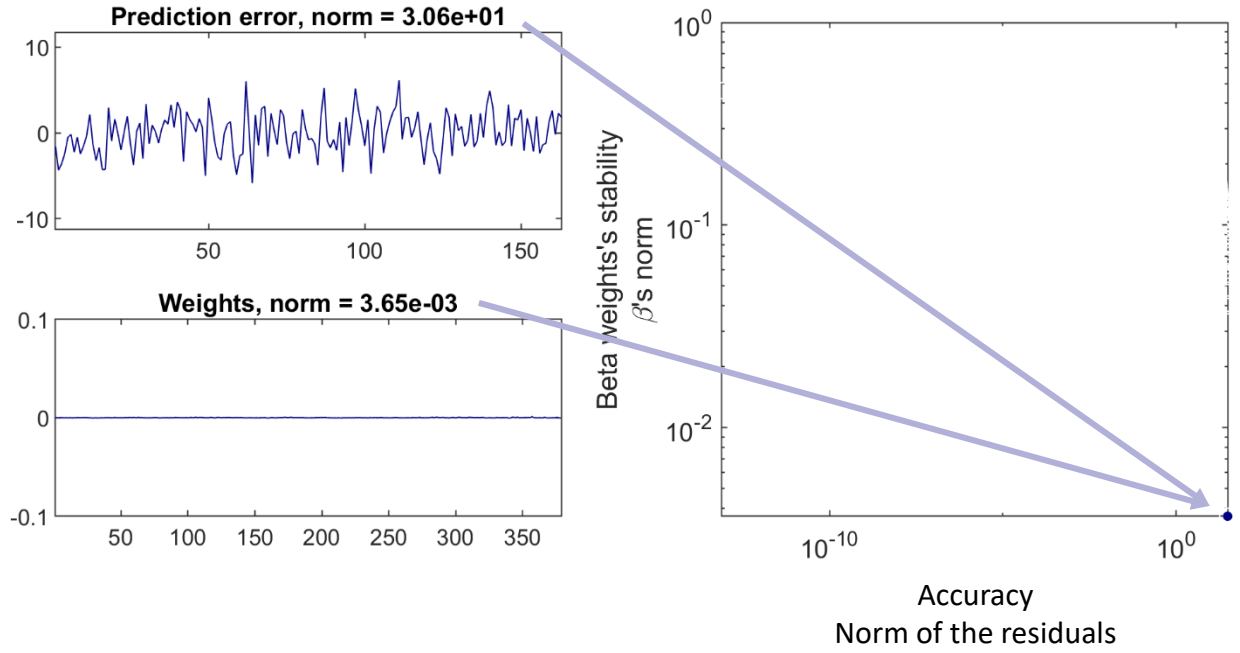
Let’s get back to our example:
379 variables and 163 observations



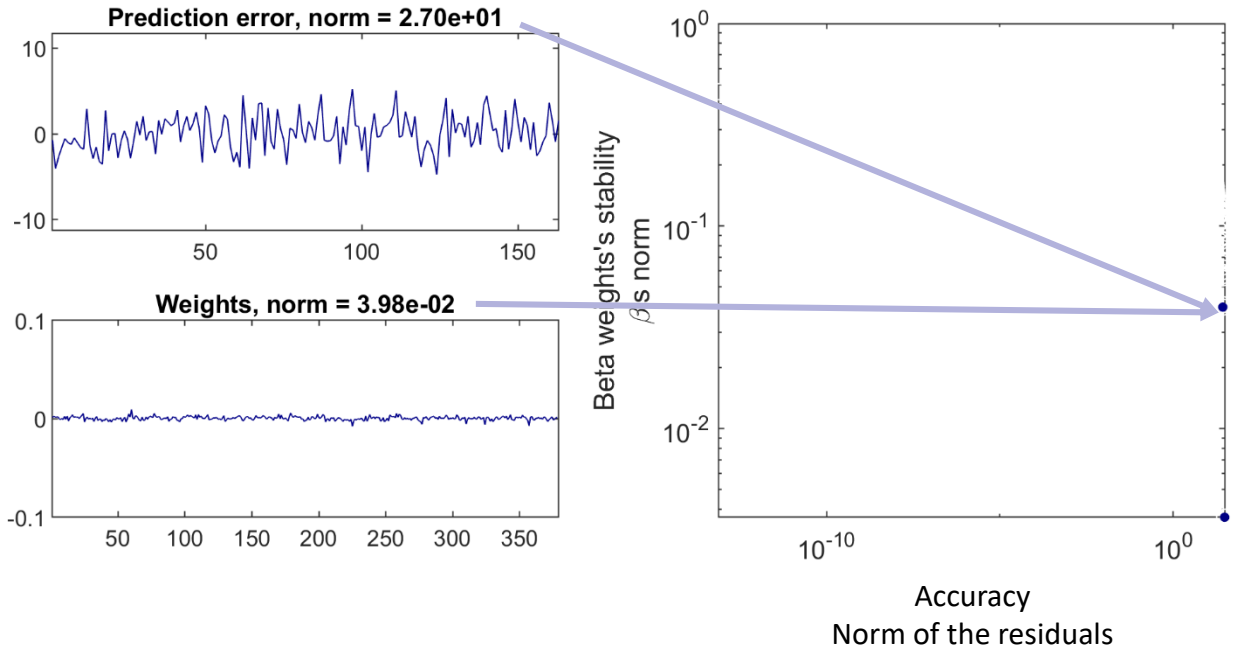


Unstable
Pseudo-inverse
solution

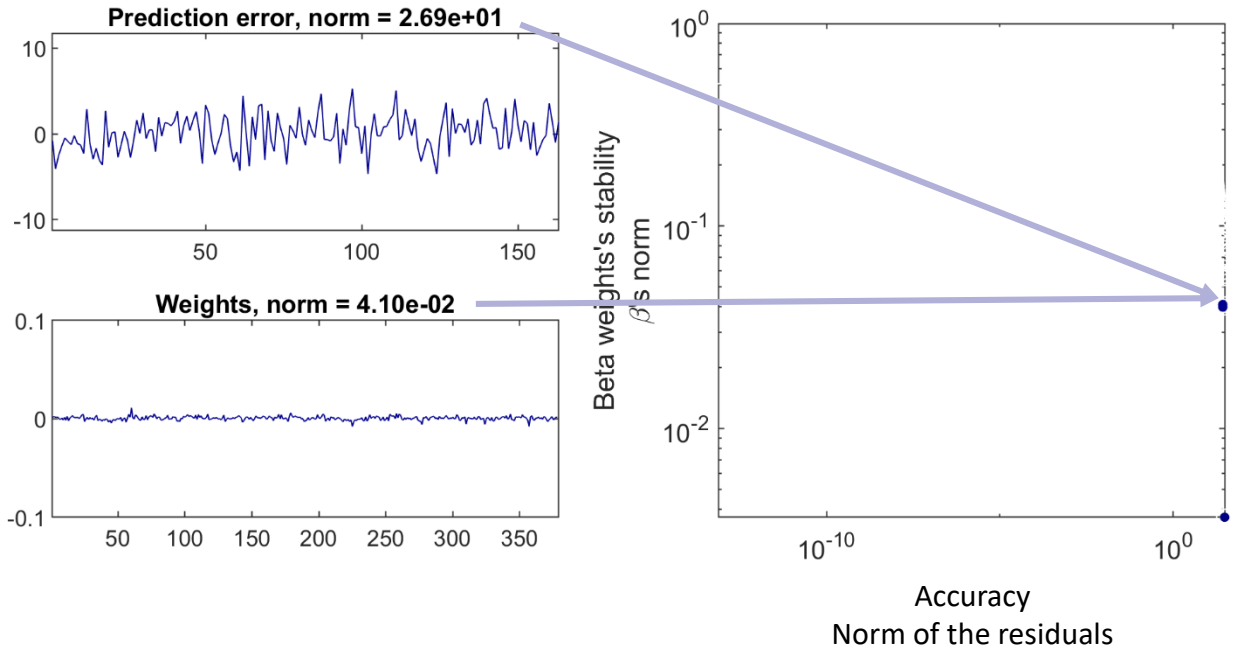
Solving the system preserving only the largest singular value



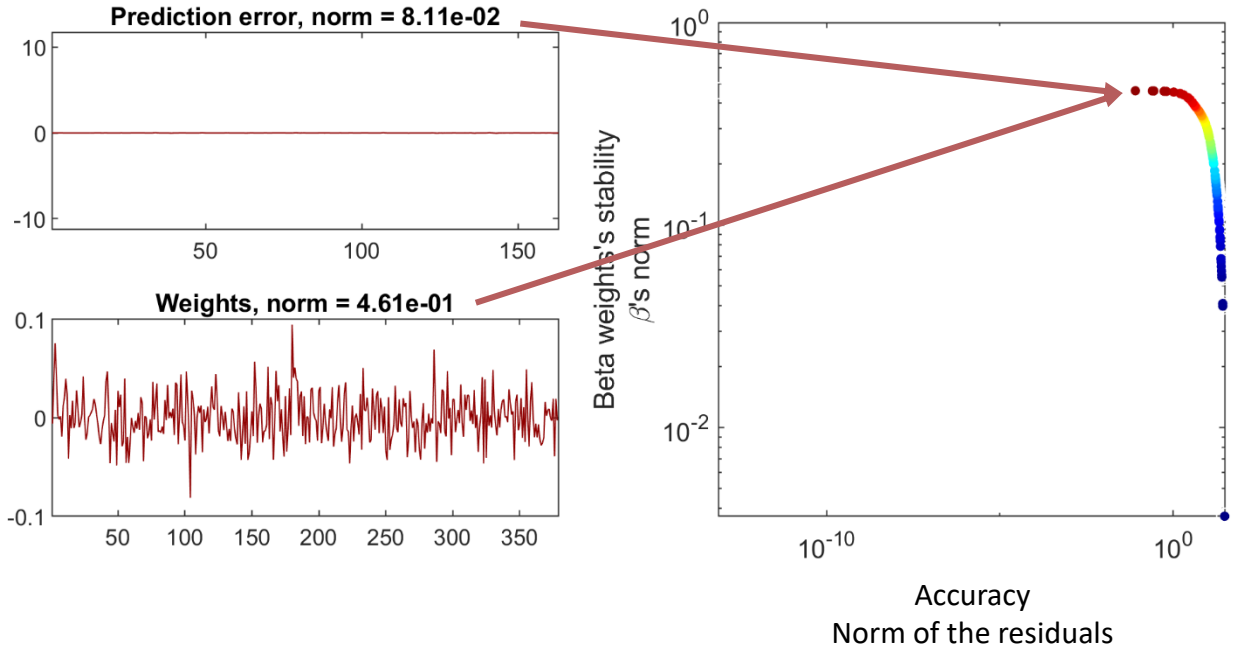
Preserving two singular values



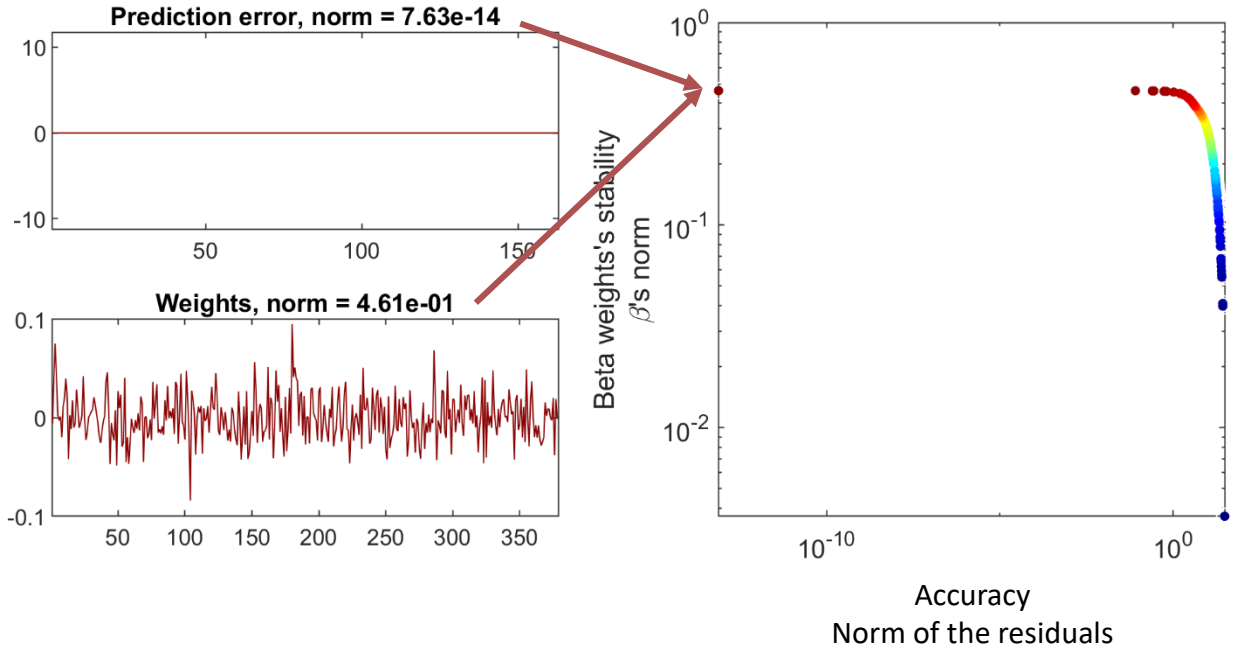
Keeping 3


Prediction error, norm = 2.69e+01
10
0
-10
50
100
150
Weights, norm = 4.10e-02
0.1
0
-0.1
50
100
150
200
250
300
350
Beta weights's stability
β's norm
10^0
10^-1
10^-2
10^-10
10^0
Accuracy
Norm of the residuals

All minus one



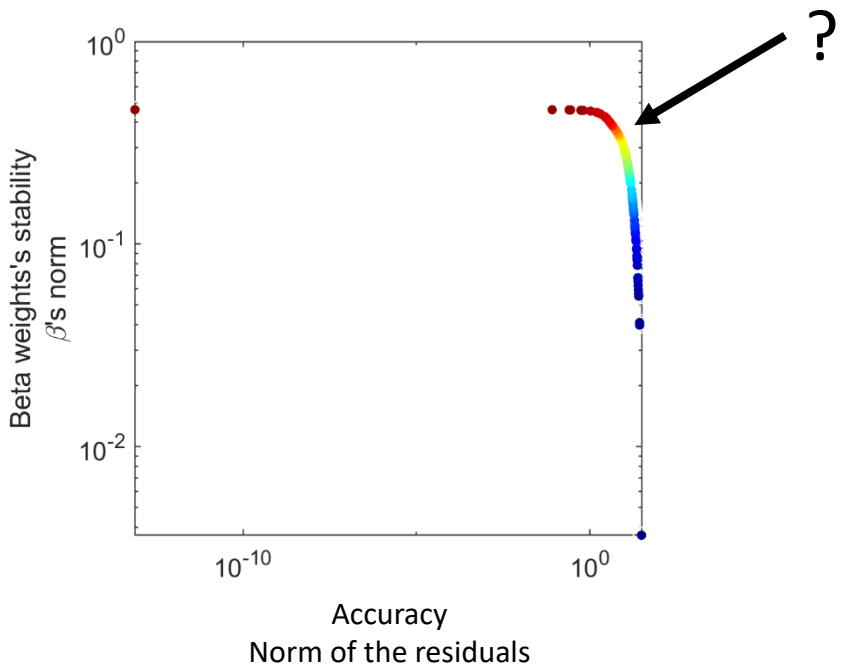
Keeping all


Prediction error, norm = 7.63e-14
10
0
-10
50
100
150
Weights, norm = 4.61e-01
0.1
0
-0.1
50
100
150
200
250
300
350
Beta weights's stability
β's norm
10^0
10^-1
10^-2
10^-10
10^0
Accuracy
Norm of the residuals

You can select the “optimal” number of components using cross-validation and maximizing predictions of out-of-sample data

Use tsvd and cross-validation

*more stable beta weights
*less over fitting
*applicable to outside dataset



Section's summary

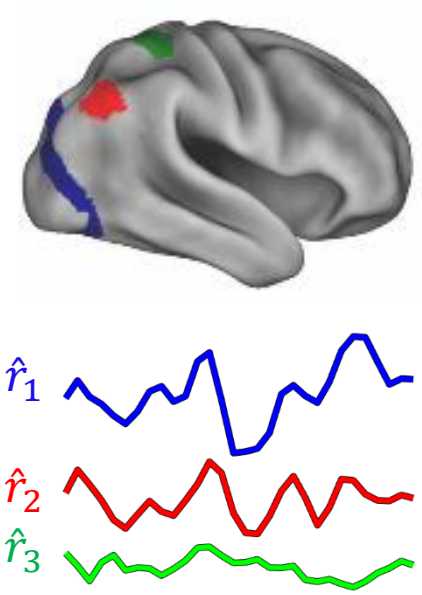
- Testing performance on the same data used to obtain a model leads to overfitting. Do not do it. Use cross-validation instead.
- Modeling is hard, especially when the number of "unknowns" exceeds the number of measurements: "ill-posed" systems
- These types of problems are common on neuroimaging projects
- Regularization and cross-validation can minimize the risk of overfitting and lead to better out-of-sample performance

Towards estimates of functional connectivity that generalize across datasets

Correlations might not be enough with limited data (~5 mins)

Connectotyping

The activity of each brain region can be predicted by the weighted contribution of all the other brain regions

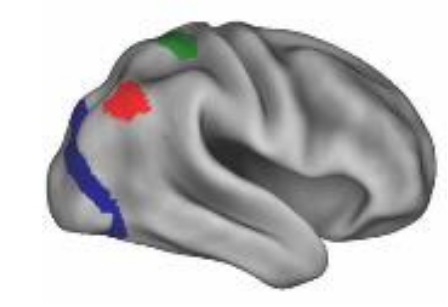


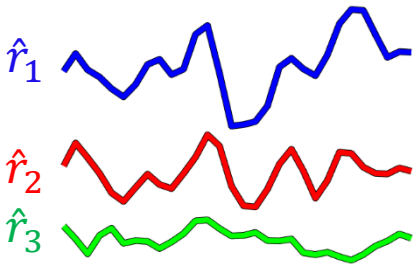
Connectotyping: Model Based Fingerprinting of the Functional Connectome

Oscar Miranda-Dominguez, Brian D. Mills, Samuel D. Carpenter, Kathleen A. Grant, Christopher D. Kroenke, Joel T. Nigg, Damien A. Fair

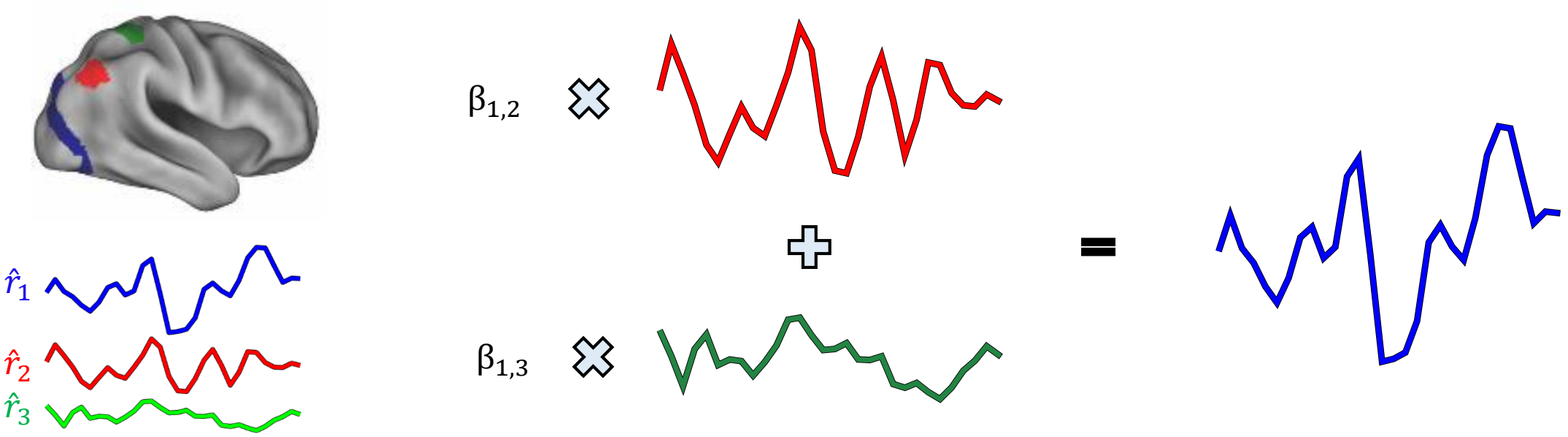
Published: November 11, 2014 • https://doi.org/10.1371/journal.pone.0111048

How can we make an educated guess of “blue” given “red” and “green”

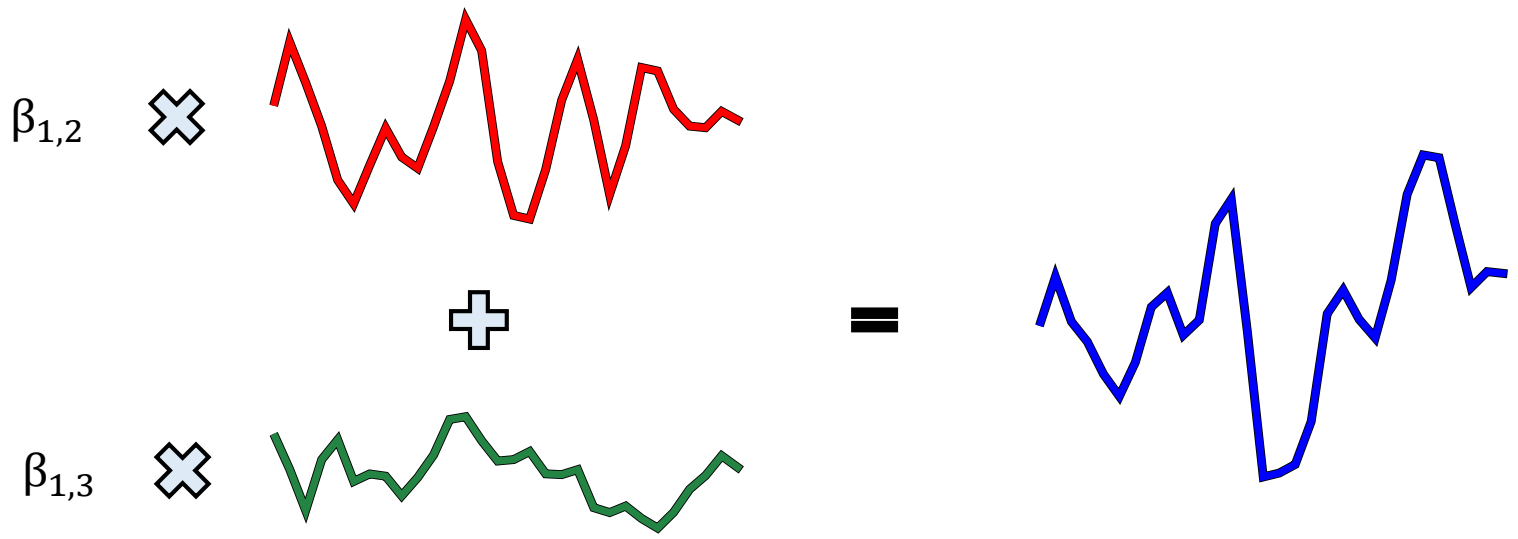




We can combine them linearly and estimate the beta weights

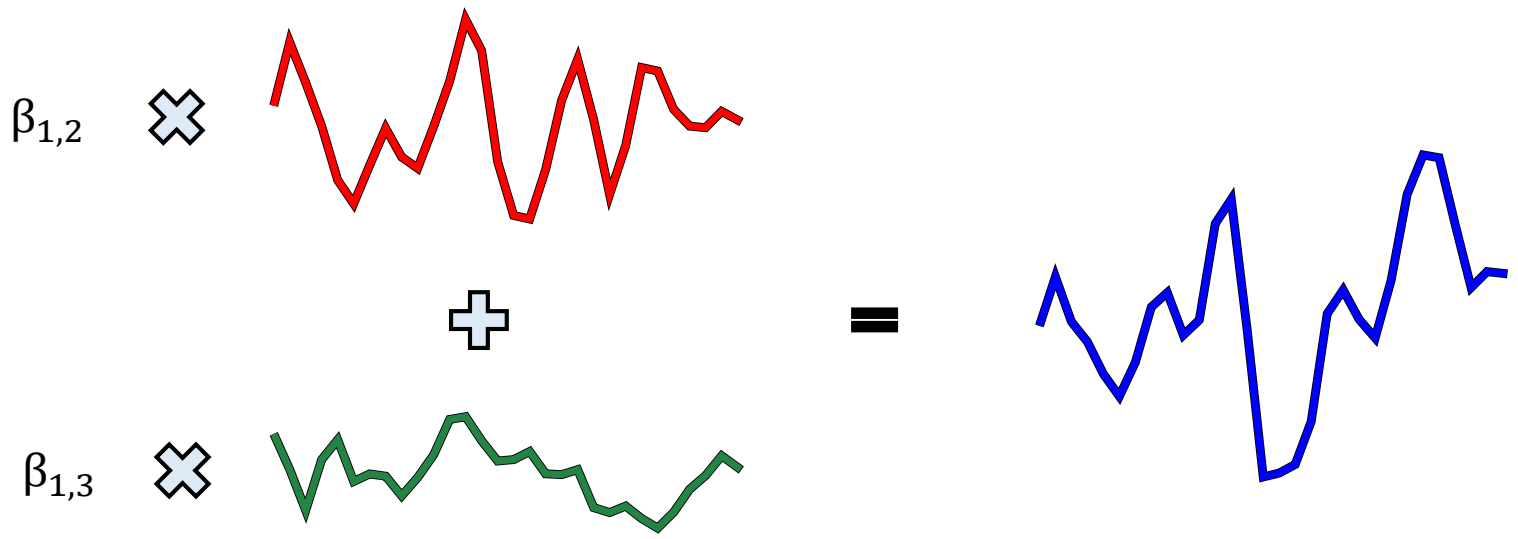


And formulate this mathematically



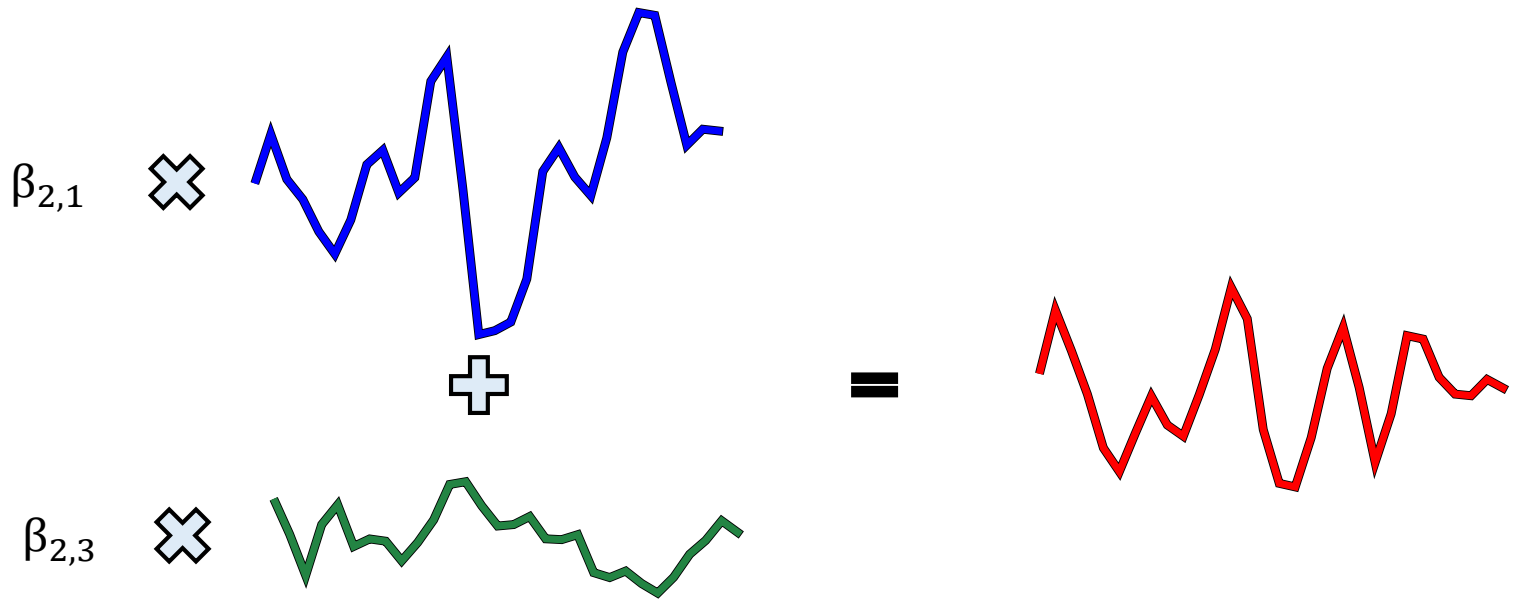
$$\hat{r}_1 = \mathbf{0}\, r_1 + \beta_{1,2}\, r_2 + \beta_{1,3}\, r_3$$

Notice that blue does not depend on blue



$$\hat{r}_1 = \mathbf{0}\, r_1 + \beta_{1,2}\, r_2 + \beta_{1,3}\, r_3$$

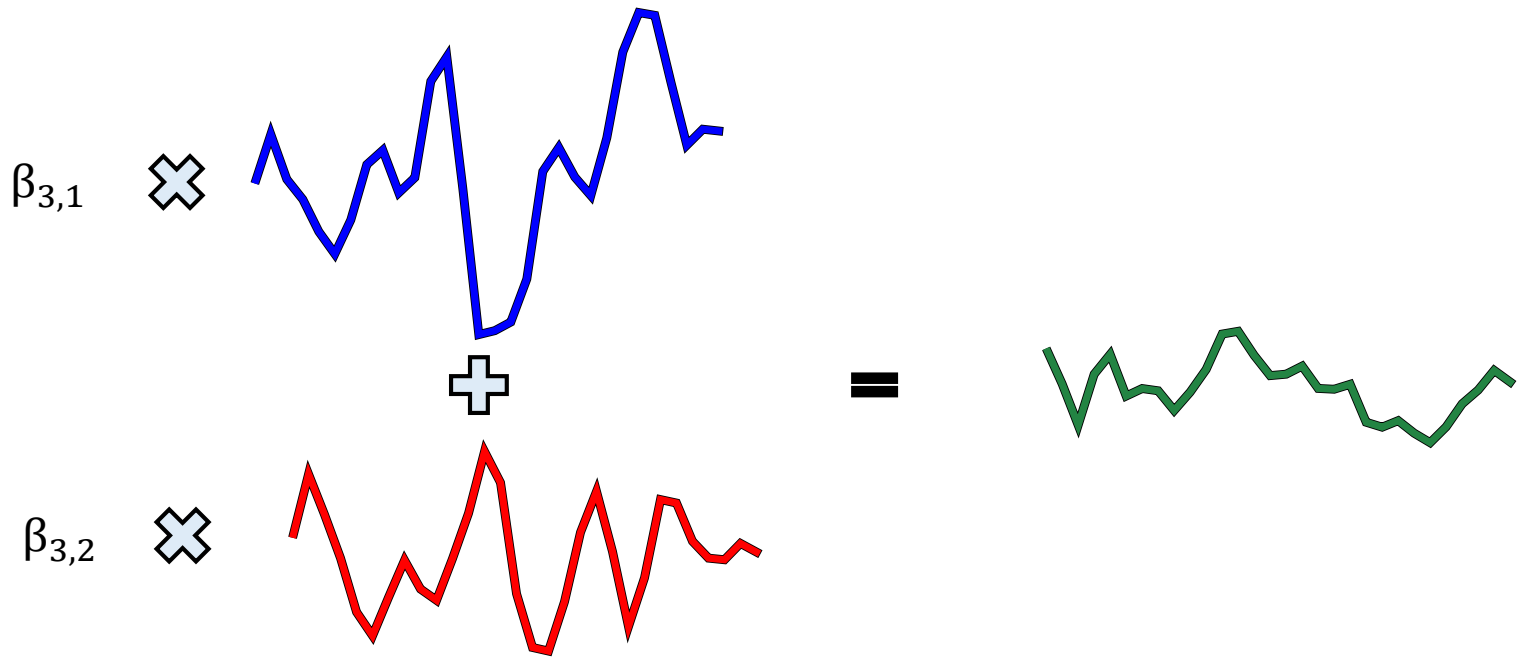
Repeat approach for red



$\hat{r}_2 = \beta_{2,1}\, r_1 + \mathbf{0}\, r_2 + \beta_{2,3}\, r_3$

Red does not depend on red

And green



$\hat{r}_3 = \beta_{3,1}\, r_1 + \beta_{3,2}\, r_2 + \mathbf{0}\, r_3$

Green does not depend on green

Which can be represented as a 3x3 matrix

$$\hat{r}_1 = 0\, r_1 + \beta_{1,2}\, r_2 + \beta_{1,3}\, r_3$$

$$\hat{r}_2 = \beta_{2,1}\, r_1 + 0\, r_2 + \beta_{2,3}\, r_3$$

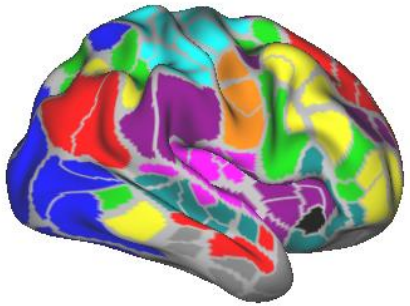
$$\hat{r}_3 = \beta_{3,1}\, r_1 + \beta_{3,2}\, r_2 + 0\, r_3$$

Matricial form

$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \beta_{1,2} & \beta_{1,3} \\ \beta_{2,1} & \mathbf{0} & \beta_{2,3} \\ \beta_{3,1} & \beta_{3,2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

General case (“M” instead of 3 ROIs):

A bigger matrix



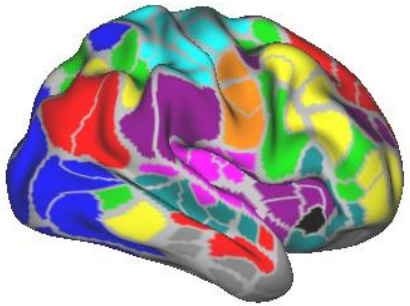
General case

$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_M \end{bmatrix} = \begin{bmatrix} 0 & \beta_{1,2} & \dots & \beta_{1,M} \\ \beta_{2,1} & 0 & \dots & \beta_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M,1} & \beta_{M,2} & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix}$$

Ill-posed system (more unknowns that data)

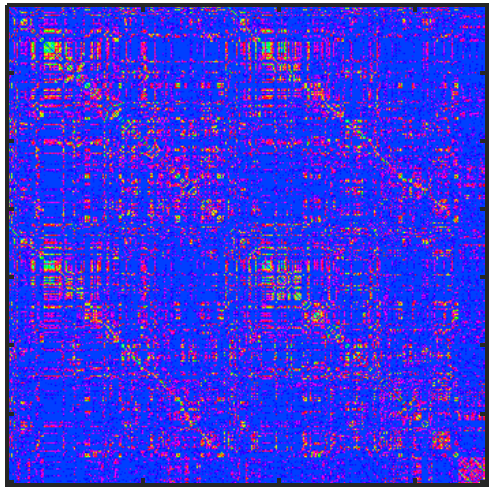
Solved by regularization and cross validation

And the solution is an individualized connectivity matrix



General case

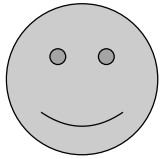
$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_M \end{bmatrix} = \begin{bmatrix} 0 & \beta_{1,2} & \dots & \beta_{1,M} \\ \beta_{2,1} & 0 & \dots & \beta_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M,1} & \beta_{M,2} & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix}$$

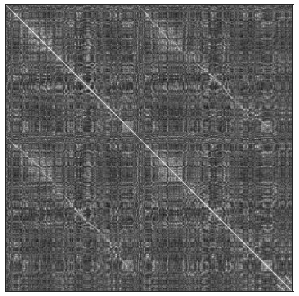


Solution!

Connectivity matrices (models) can be compared

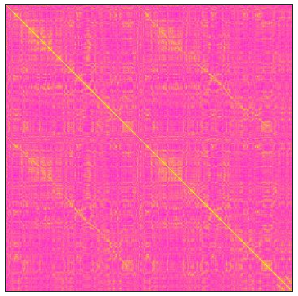
Subject 1





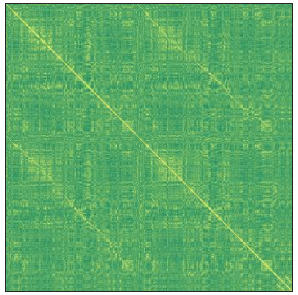
$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_M \end{bmatrix} = \begin{bmatrix} 0 & \beta_{1,2} & \dots & \beta_{1,M} \\ \beta_{2,1} & 0 & \dots & \beta_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M,1} & \beta_{M,2} & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix}$$

Subject 2



$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_M \end{bmatrix} = \begin{bmatrix} 0 & \beta_{1,2} & \dots & \beta_{1,M} \\ \beta_{2,1} & 0 & \dots & \beta_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M,1} & \beta_{M,2} & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix}$$

Subject 3

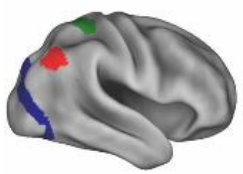


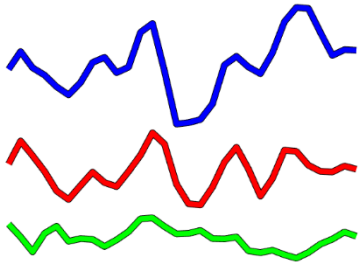
$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_M \end{bmatrix} = \begin{bmatrix} 0 & \beta_{1,2} & \dots & \beta_{1,M} \\ \beta_{2,1} & 0 & \dots & \beta_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{M,1} & \beta_{M,2} & \dots & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix}$$

- models can also predict brain activity

To predict brain activity

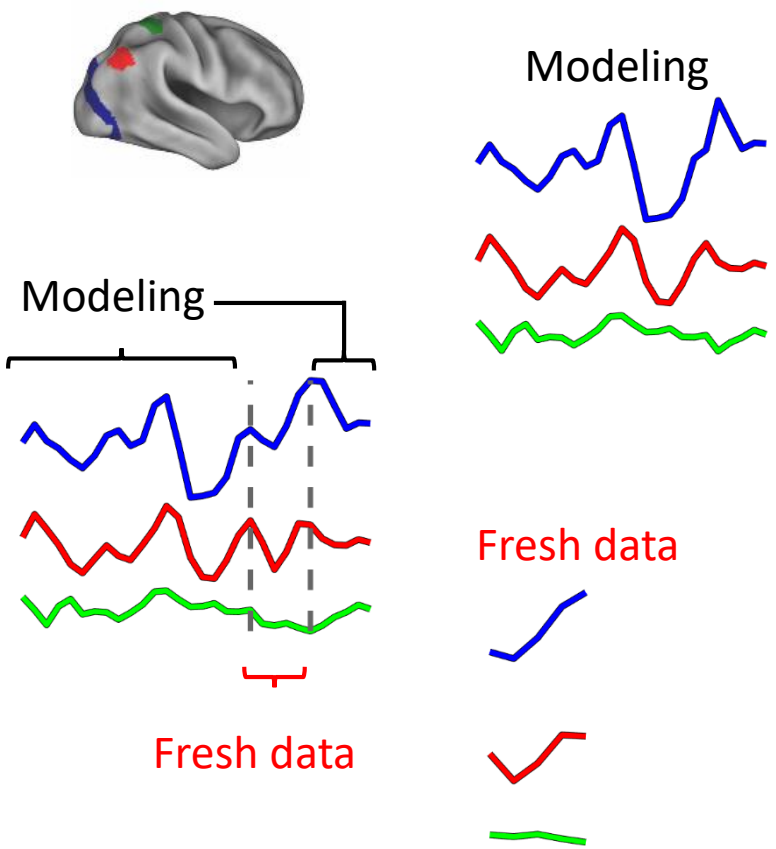
- Start with the original fMRI data (after cleaning)





Next, split the data randomly in 2 sections:

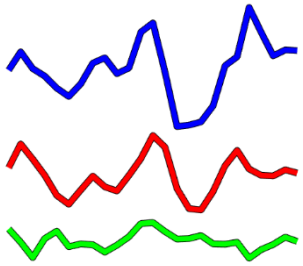
One for modeling, the other for prediction



Use the section modeling for connectotyping

Calculate the beta weights (connectivity matrix)!

Modeling



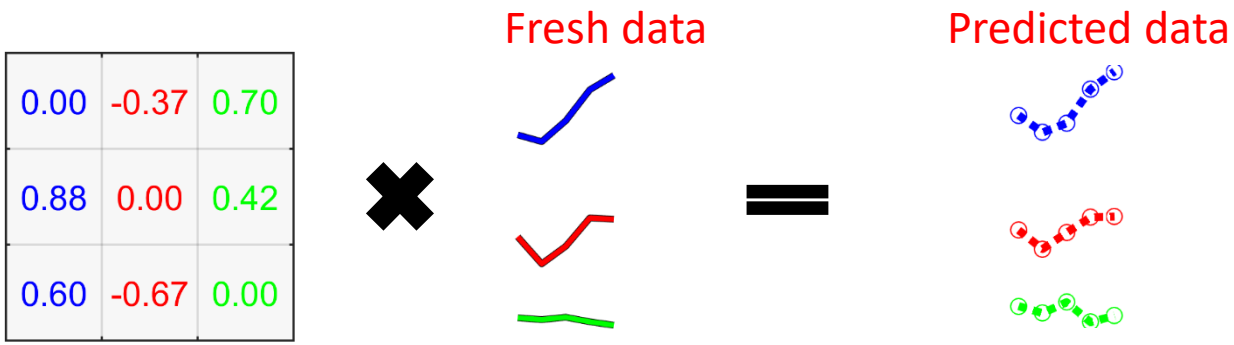
Connectotype

0.00	-0.37	0.70
0.88	0.00	0.42
0.60	-0.67	0.00

$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \beta_{1,2} & \beta_{1,3} \\ \beta_{2,1} & \mathbf{0} & \beta_{2,3} \\ \beta_{3,1} & \beta_{3,2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

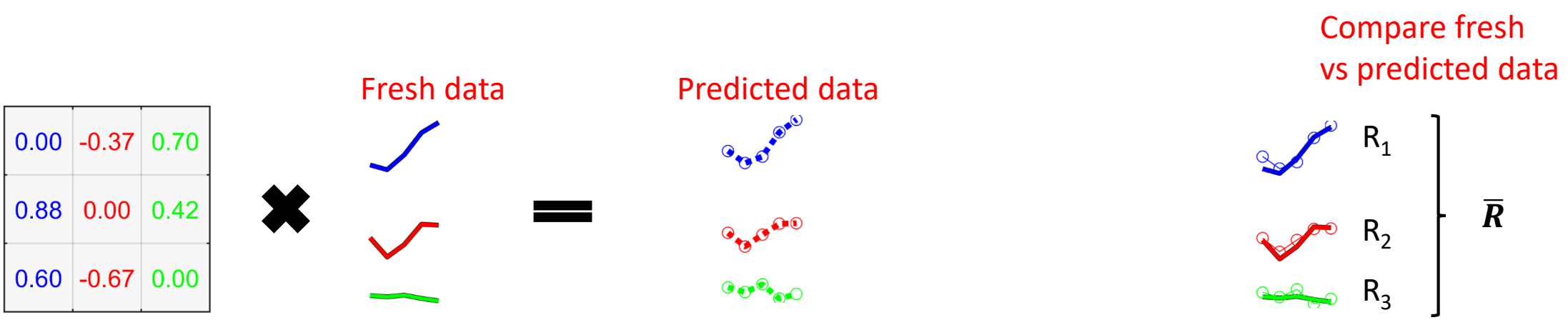
Fresh data

Use the matrix to predict brain activity in fresh data



Compare fresh data with predicted data

You may use correlation coefficients!



Validation

Data sets

HUMANS:

- 27 healthy adult humans (16 females)
  age 19 to 35 years

- Subset scanned a second time
  two weeks later

(Validated in data from 11 macaques too)

Validation
Step 1

Approach:

1. A model was calculated for each participant using partial data

Validation
Step 2

Approach:

1. A model was calculated for each participant using partial data

2. Each model was used to predict fresh data for each scan

Validation

Step 3

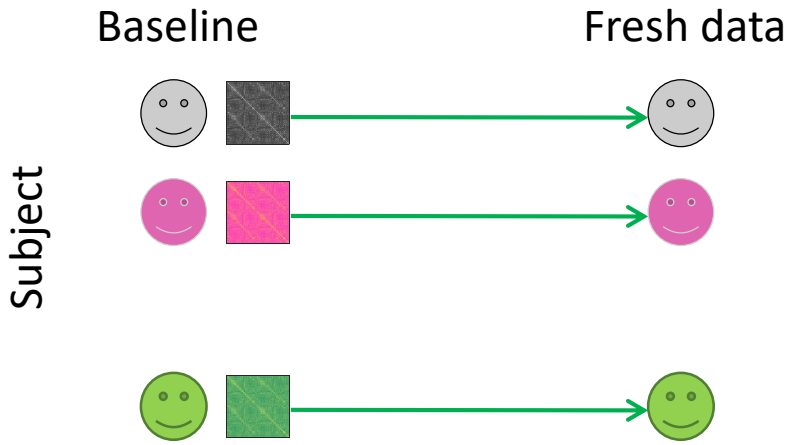
Approach:

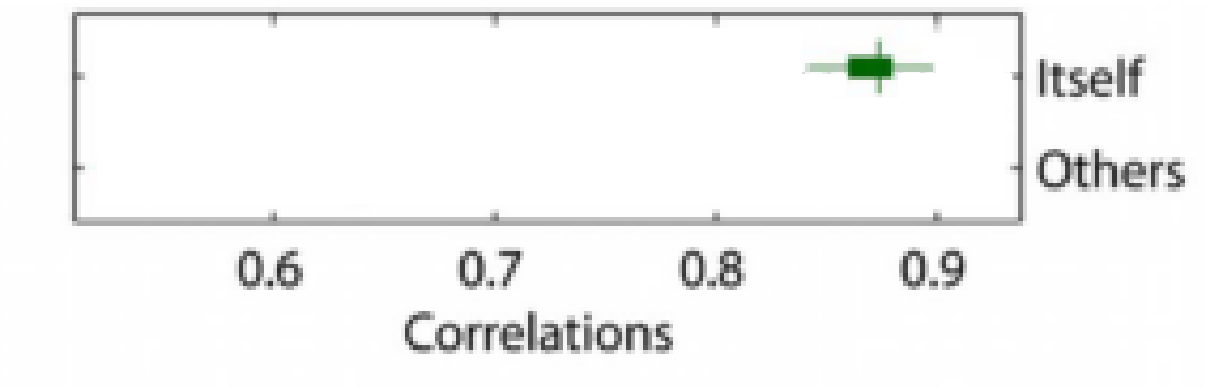
1. A model was calculated for each participant using partial data

2. Each model was used to predict fresh data for each scan

3. Average correlation between predicted and observed timecourses was calculated

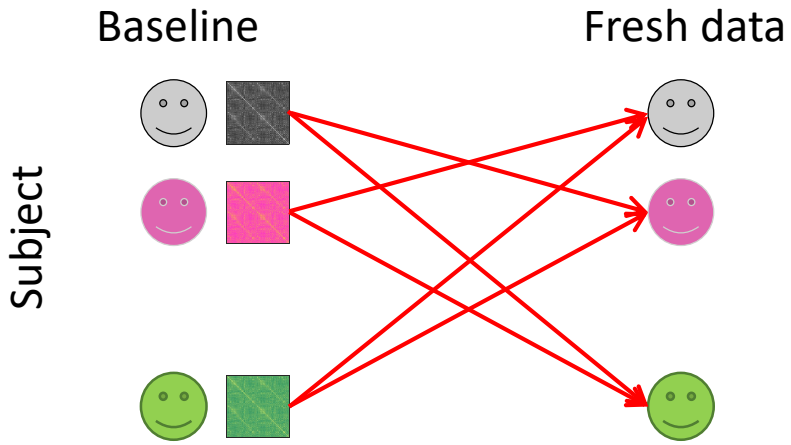
When the model and fresh data came from the same participants, $\bar{R} \approx 0.87$

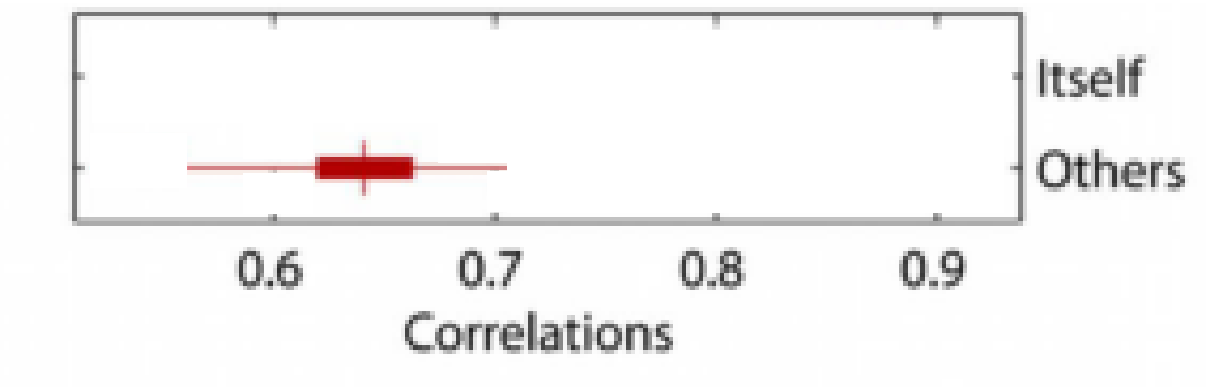




Miranda-Dominguez O, et al.. PLoS One. 2014

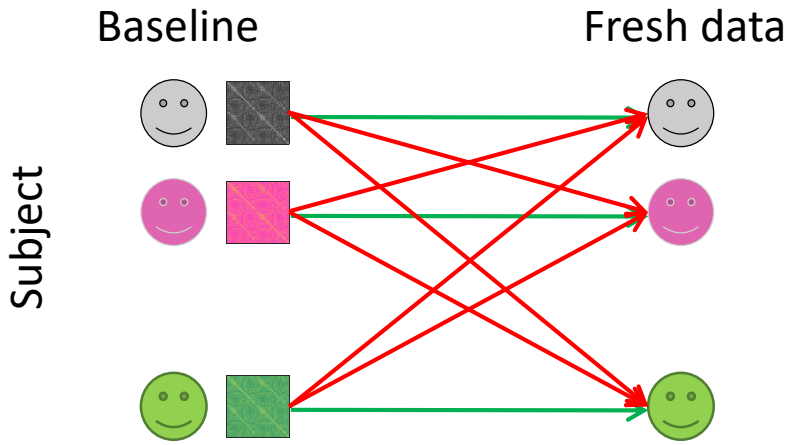
When the model and fresh data came from different participants, $\overline{R} \approx 0.64$

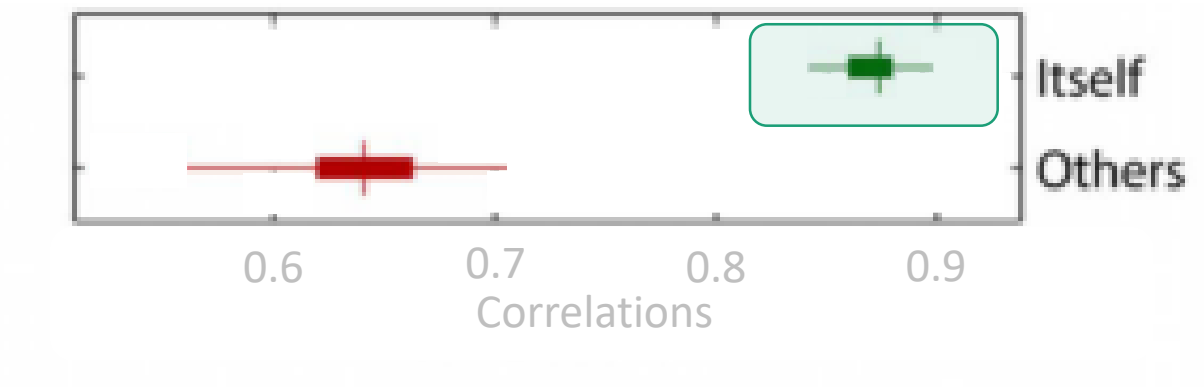




Miranda-Dominguez O, et al.. PLoS One. 2014

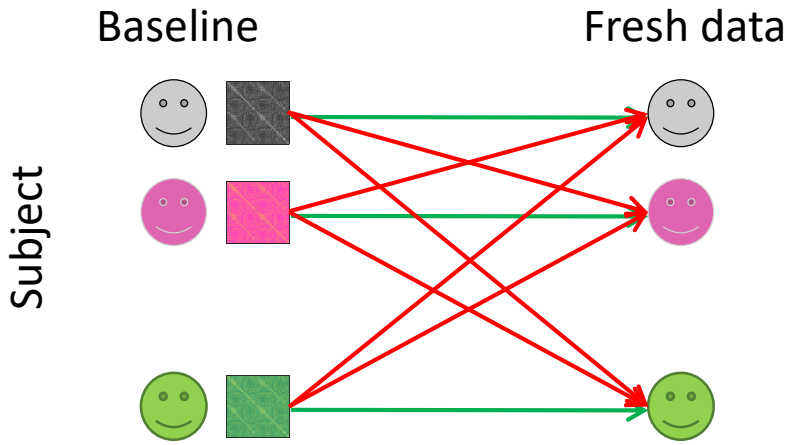
Notice that by looking at a single number ($\overline{\boldsymbol{R}}$) we can characterize individuals, since there was no overlap in predicting *self* versus *others*

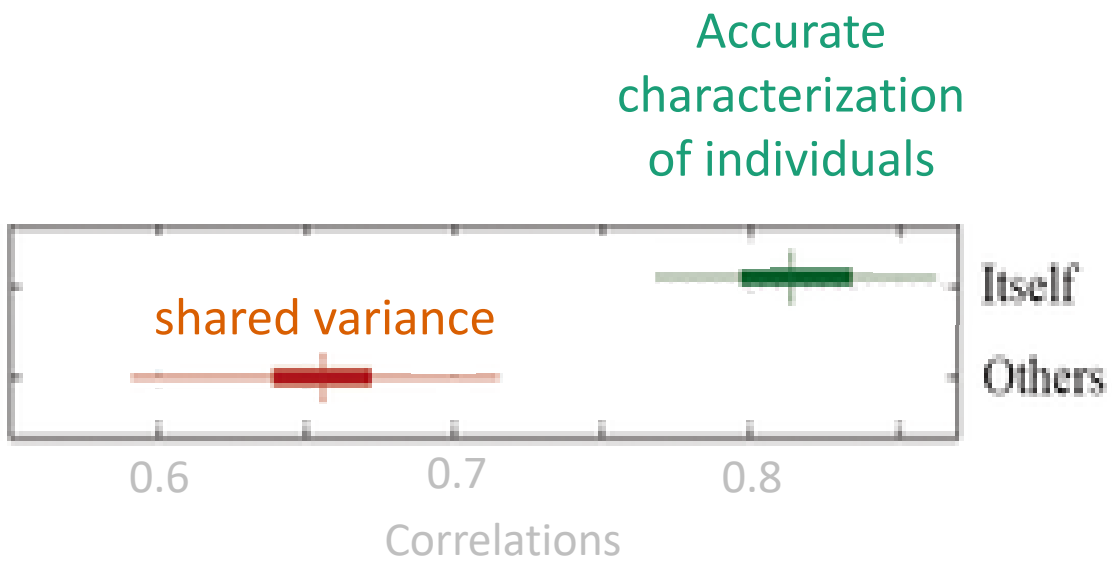




Miranda-Dominguez O, et al.. PLoS One. 2014

As further validation, we predicted fresh data acquired 2 weeks later, finding the same trend:

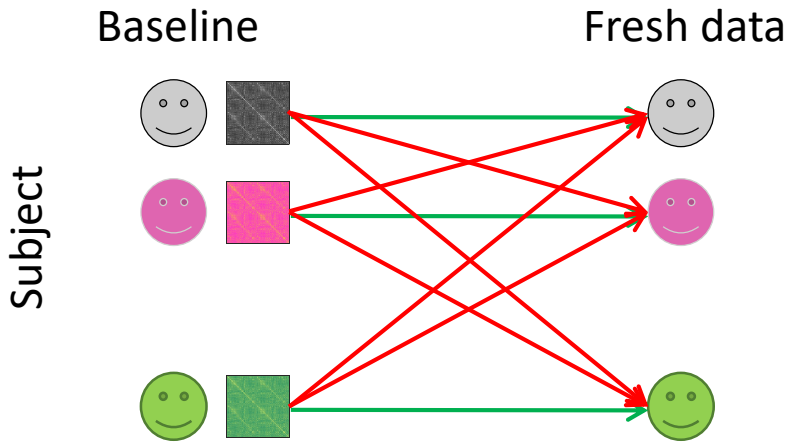


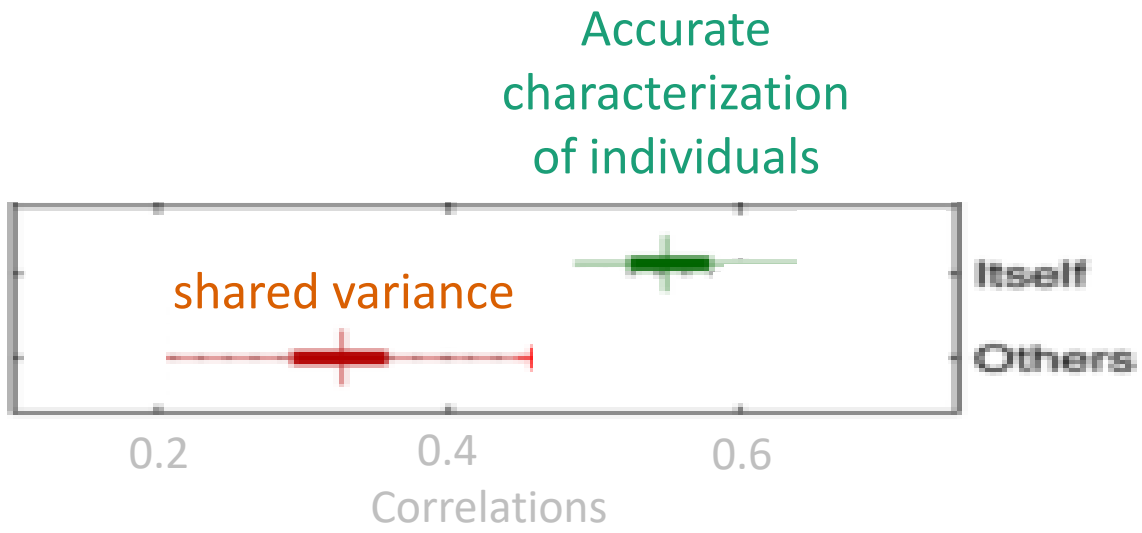


Miranda-Dominguez O, et al.. PLoS One. 2014

Same trend is also observed in macaques

$\overline{R}$ are reduced

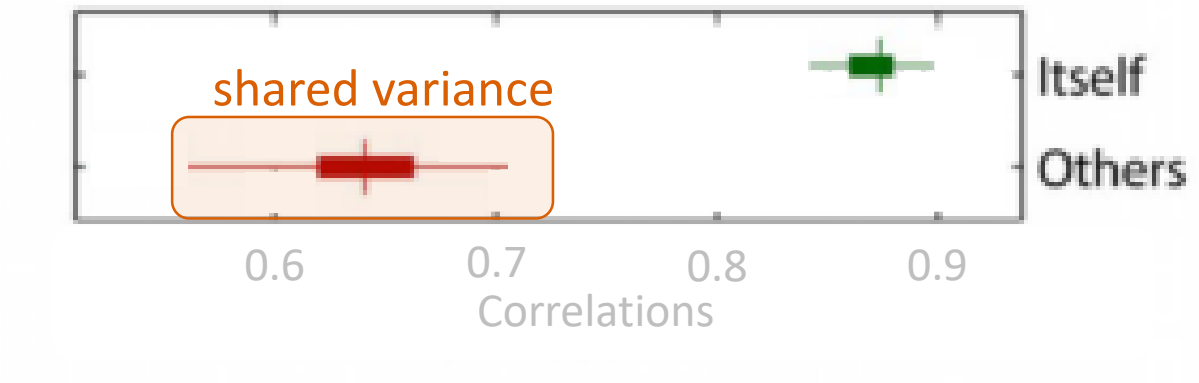




Miranda-Dominguez O, et al.. PLoS One. 2014

These findings suggest that

We are all equipped with functional networks that process certain stimuli in the same way



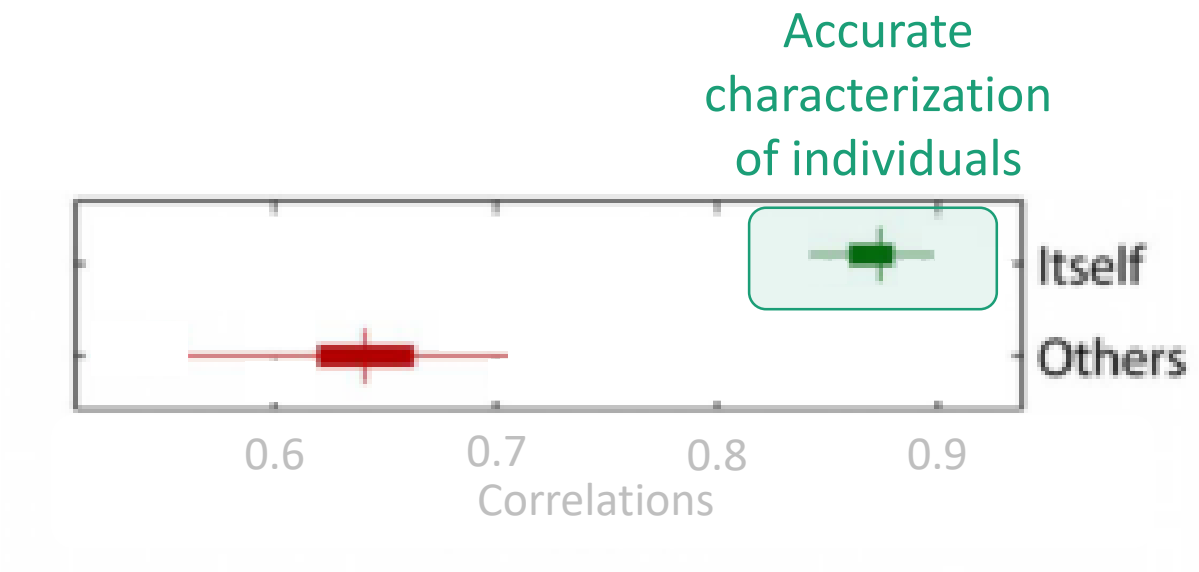
Miranda-Dominguez O, et al.. PLoS One. 2014

These findings suggest that

We are all equipped with functional networks that process certain stimuli in the same way

… on top of this…

we all each have unique salient functional networks that make us unique

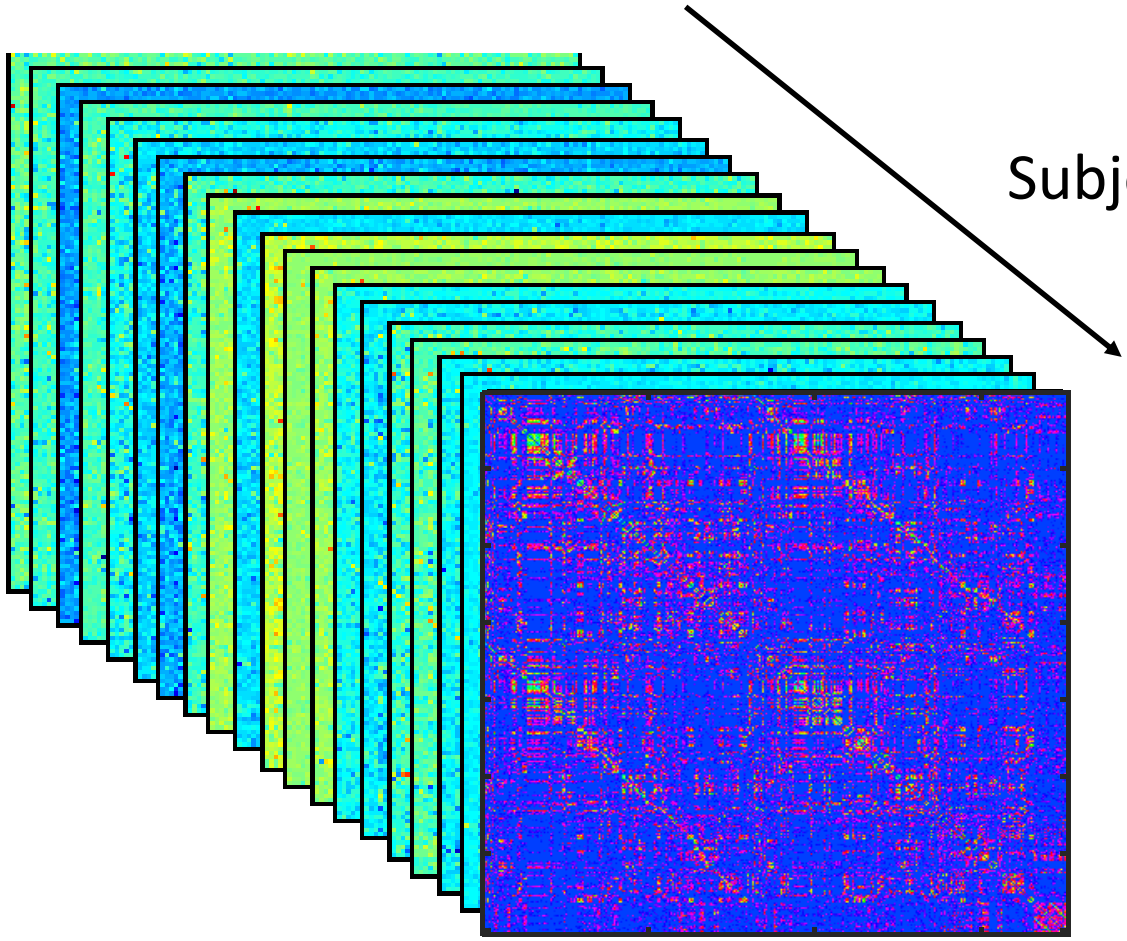


Miranda-Dominguez O, et al.. PLoS One. 2014

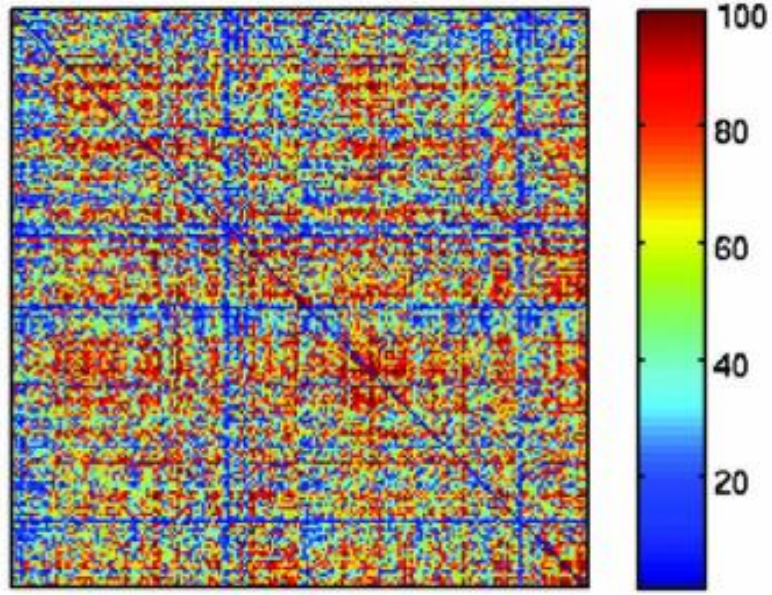
So, the next question is

“What brain systems make a connectome unique”

To do this, we look at how similar or different the models were across participants

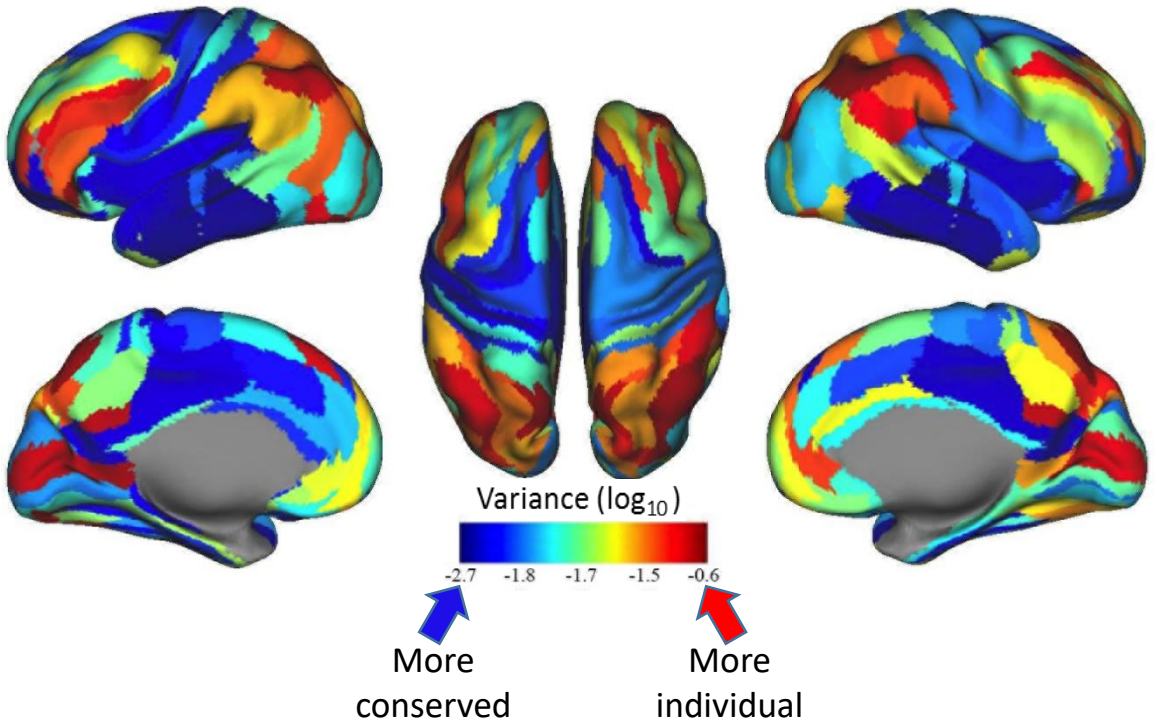


Variance Across Subjects



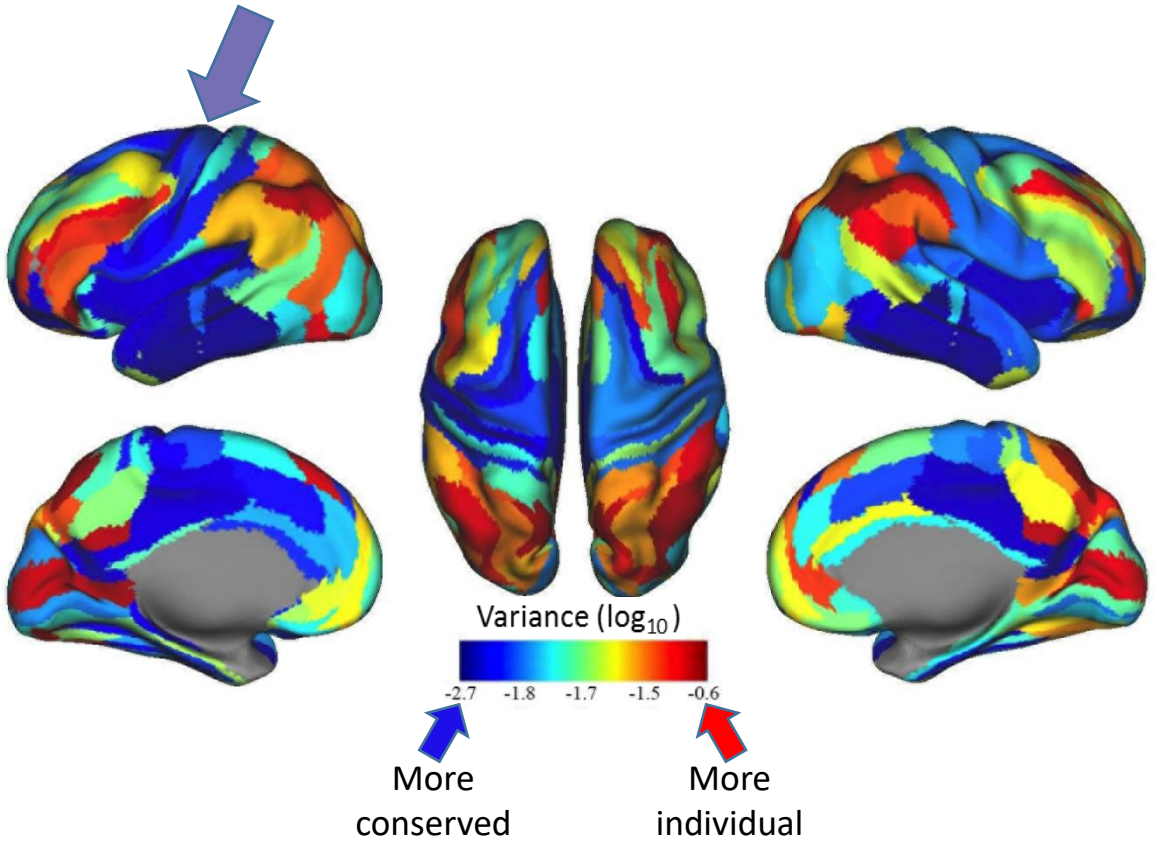
Miranda-Dominguez O, et al.. PLoS One. 2014

Fronto-parietal cortex
makes a connectome unique



Miranda-Dominguez O, et al.. PLoS One. 2014

In contrast, notice how similar motor systems are across individuals



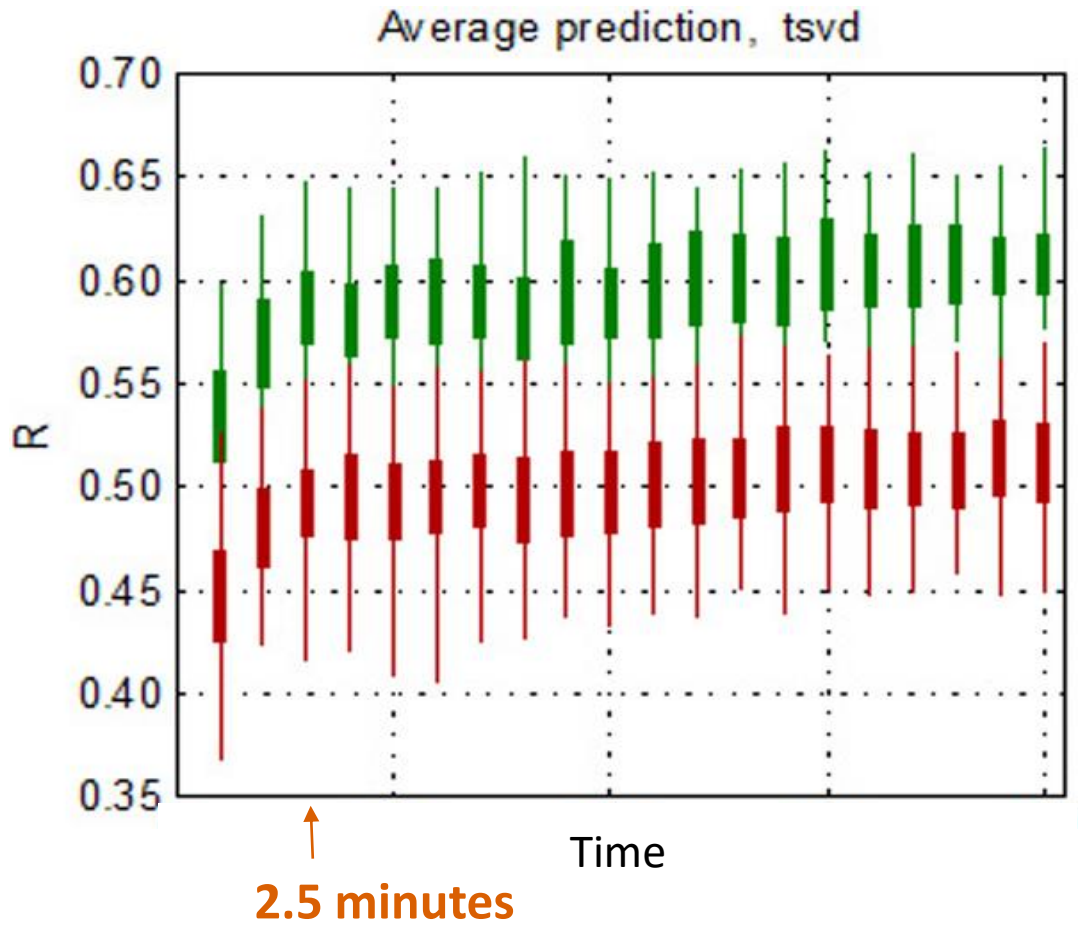
Miranda-Dominguez O, et al.. PLoS One. 2014

How much data is needed to connectotype?

2.5 minutes of data is enough to connectotype!

- Self vs others experiment was repeated using different amounts of data

- 2.5 minutes of data is enough to connectotype!



Miranda-Dominguez O, et al.. PLoS One. 2014

In summary, connectotyping

Identifies connectivity patterns unique to individuals

The connectotype is robust in adults and can be obtained with limited amounts of data

fronto-parietal systems are highly variable amongst individuals.

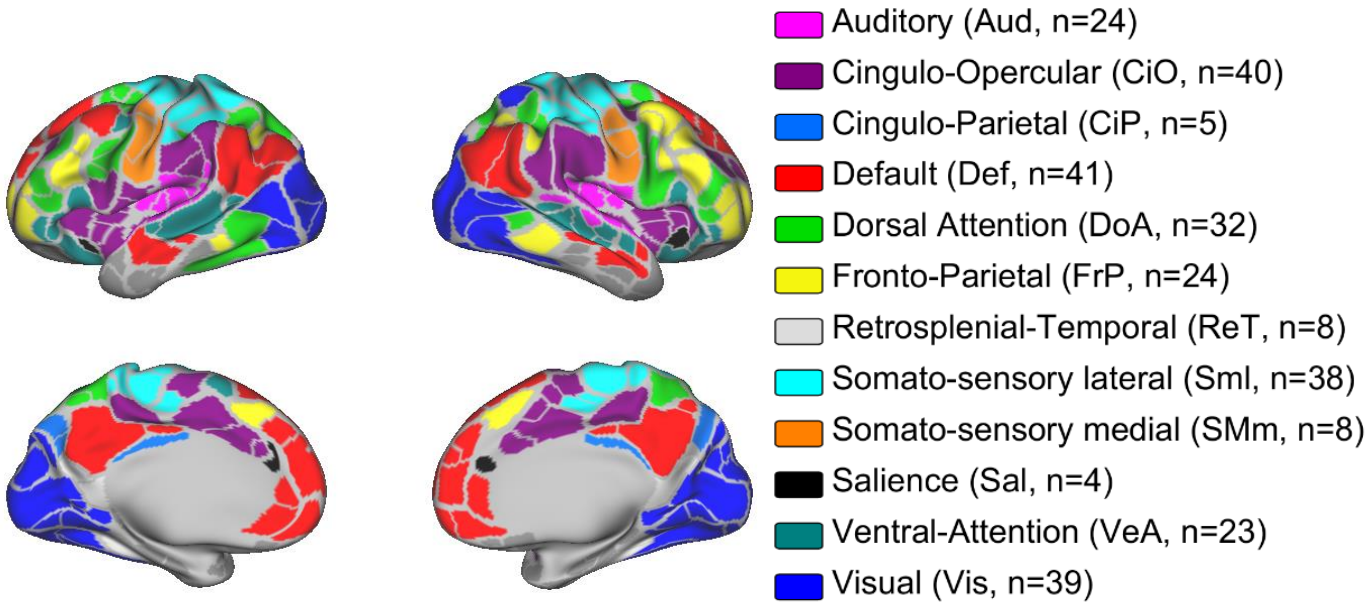
Can we use connectotyping in youth?

Participants

Controls passing QC:

- N=188 scans (159 subjects)
  - 131 subjects with 1 scan
  - 27 subjects with 2 scans
  - 1 subjects with 3 scans
- Age: 7-15
- 60% males
- Siblings (16 pairs)
  - 16 families with 2 siblings each

“Gordon” parcellation schema



Gordon et al, Cerebral Cortex, 2014

Connectotyping in youth

Step 1

Approach:

1. A model was calculated for each scan (N=188)

Connectotyping in youth

Step 2

Approach:

1. A model was calculated for each scan (N=188)

2. Each model was used to predict fresh data for each scan (N= 188 x 188 x ROIs)

Connectotyping in youth

Step 3

Approach:

1. A model was calculated for each scan (N=188)

2. Each model was used to predict fresh data for each scan (N= 188 x 188 x ROIs)

3. Average correlation between predicted and observed timecourses were calculated (N = 188 x 188)

Connectotyping in youth

Step 4

Approach:

1. A model was calculated for each scan (N=188)
2. Each model was used to predict fresh data for each scan (N=188 x 188 x ROIs)
3. Average correlation between predicted and observed timecourses were calculated (N = 188 x 188)
4. Average correlations were grouped based on the datasets used for modeling and prediction

I. Same scan
II. Same participant
III. Sibling
IV. Unrelated

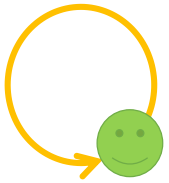
Connectotyping in youth

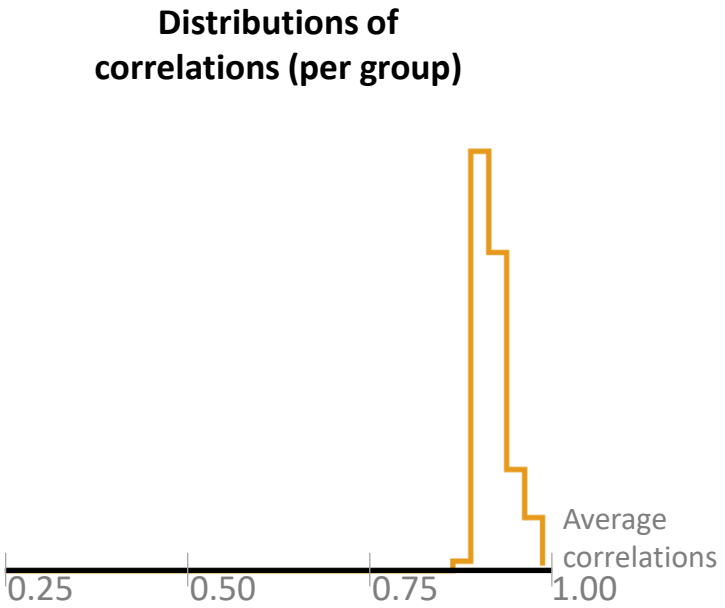
Predicting time courses

Same
scan
(N=188)

Predicting fresh data from the same scan

Same scan (N=188)

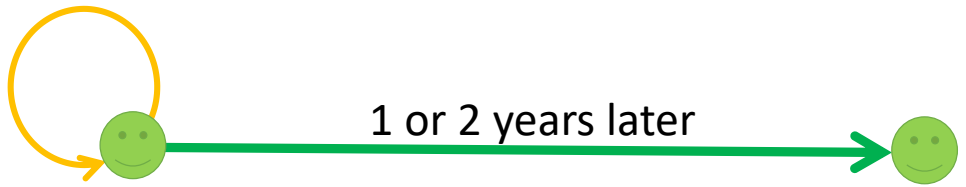




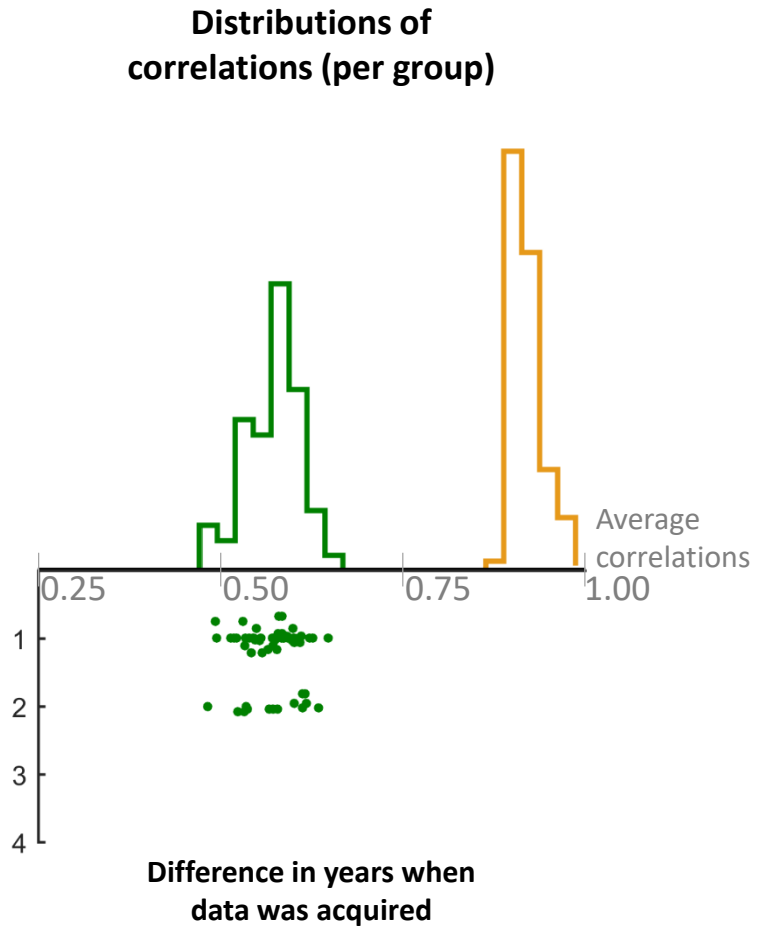
Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

Predicting data from the same participant acquired 1 or 2 years later

Same scan (N=188)

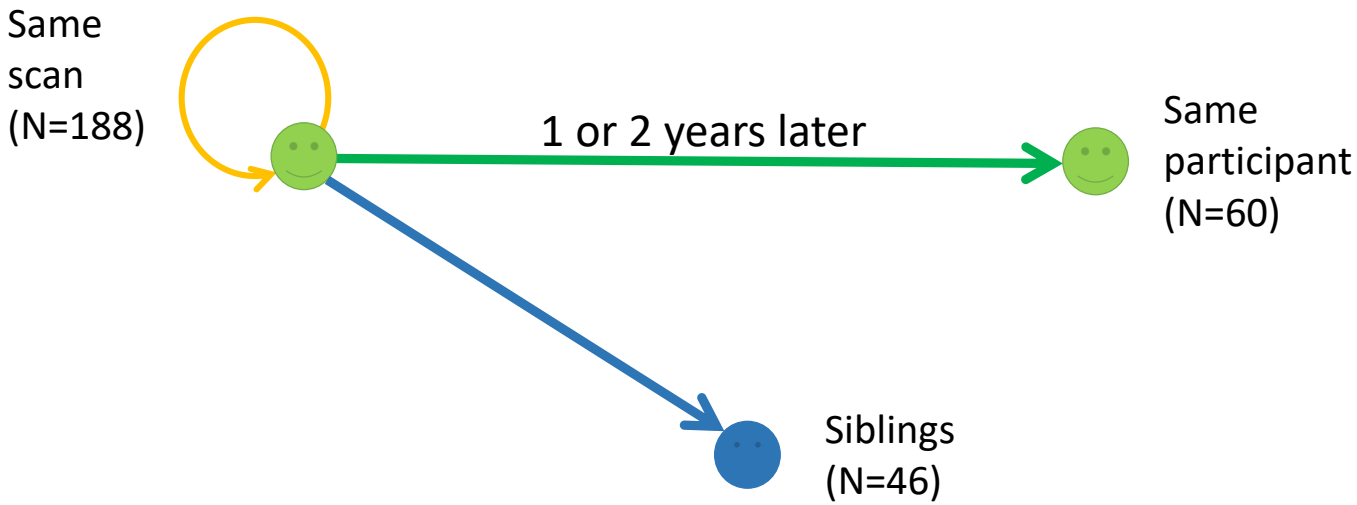


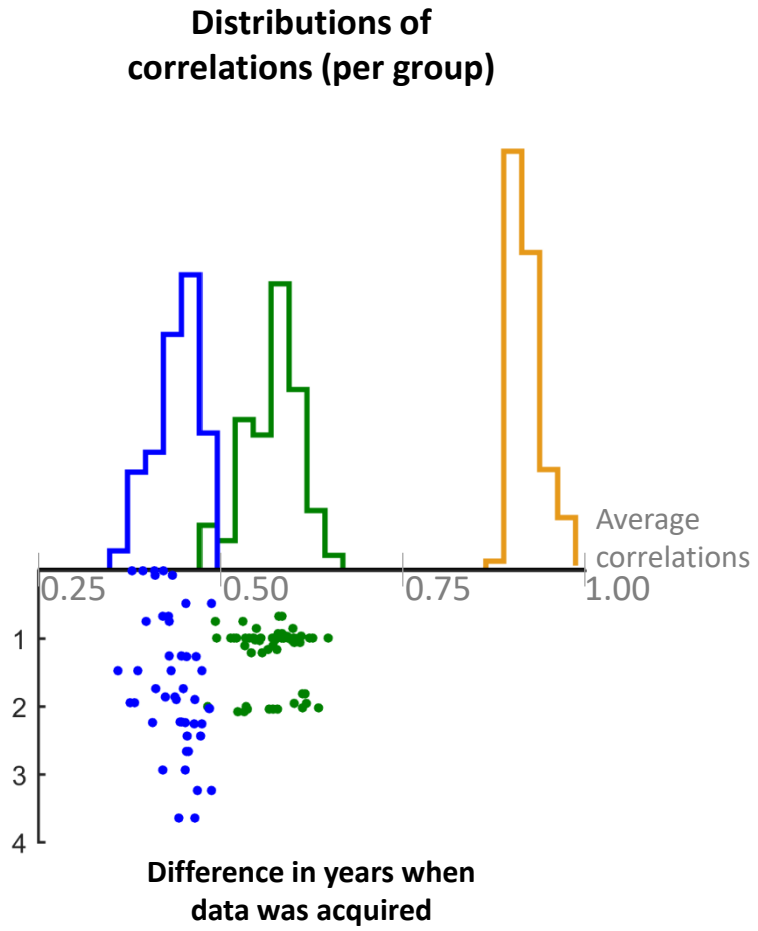
Same participant (N=60)



Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

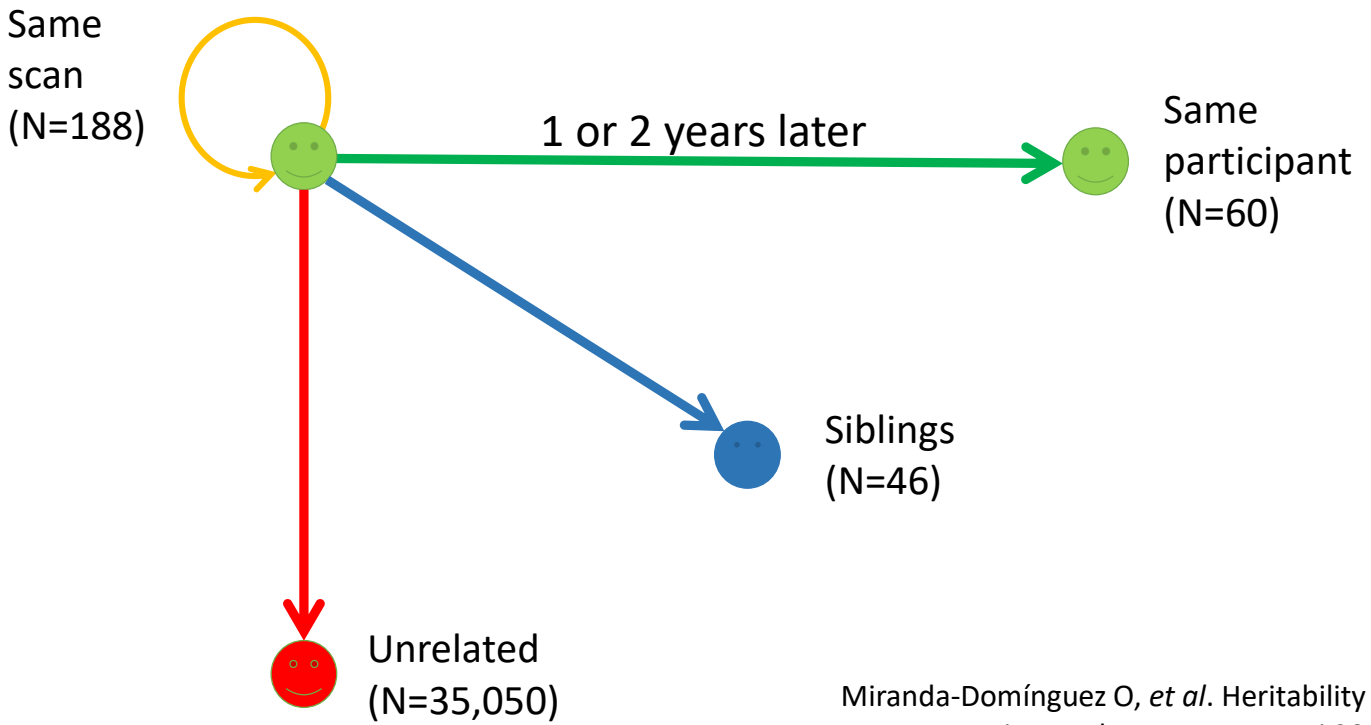
Predicting timecourses amongst siblings

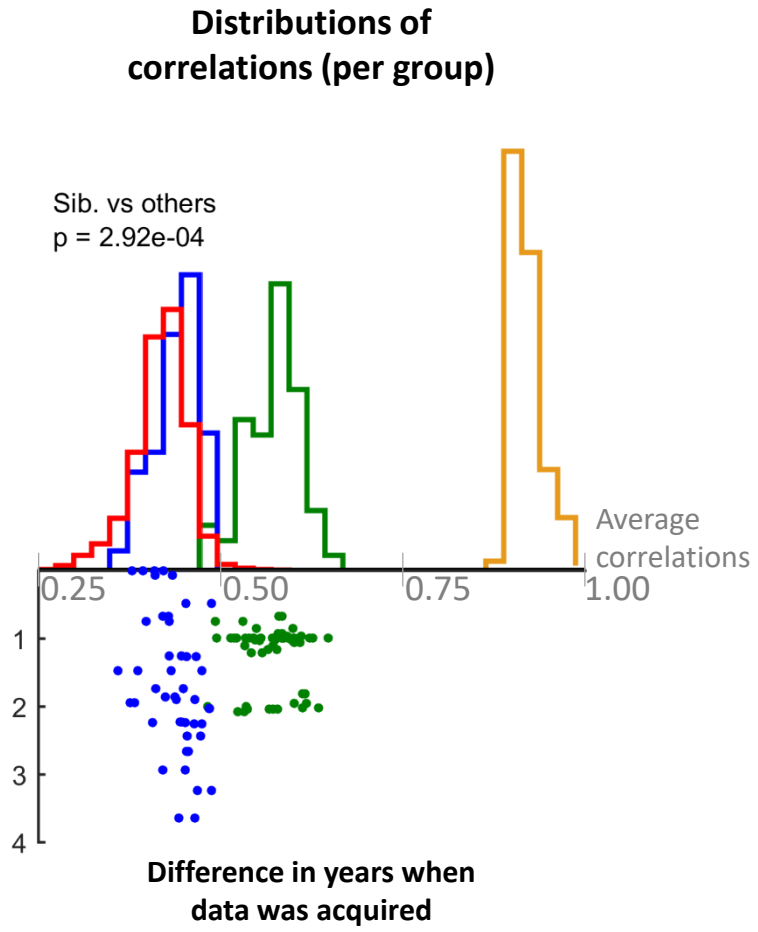




Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

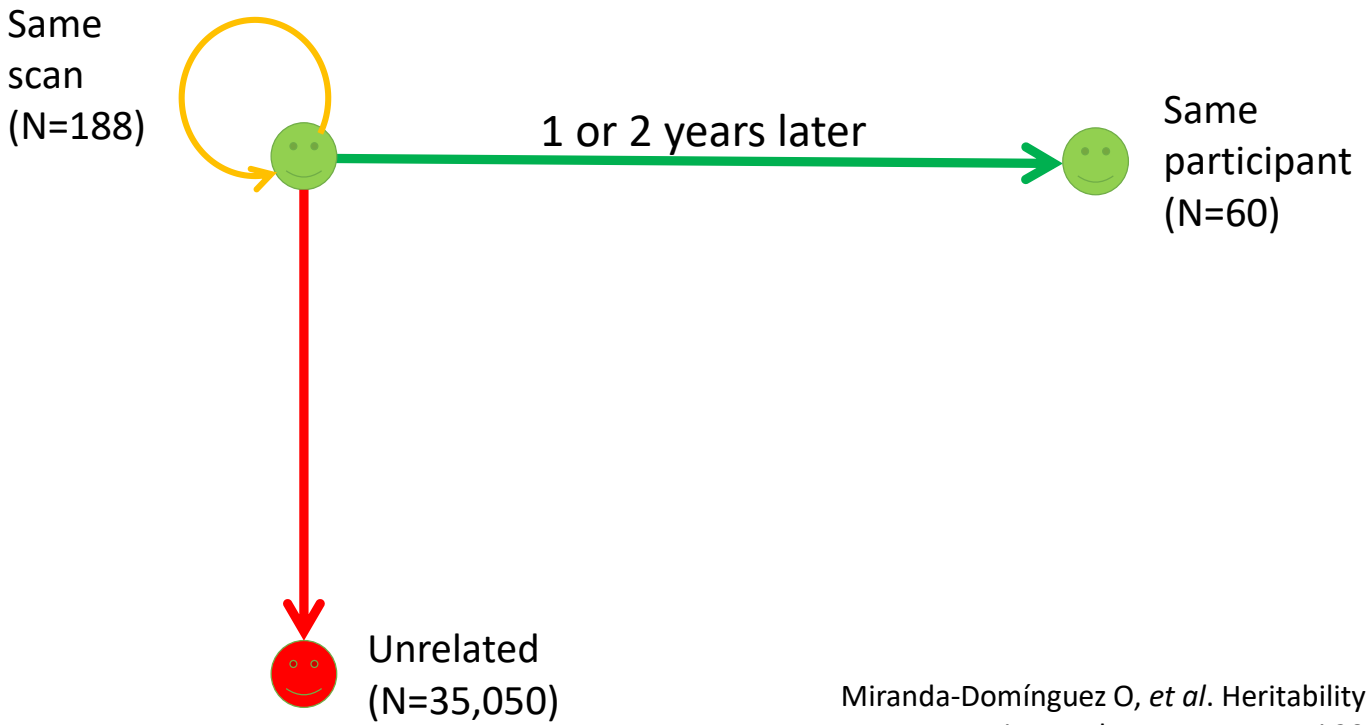
Predicting timecourses amongst unrelated

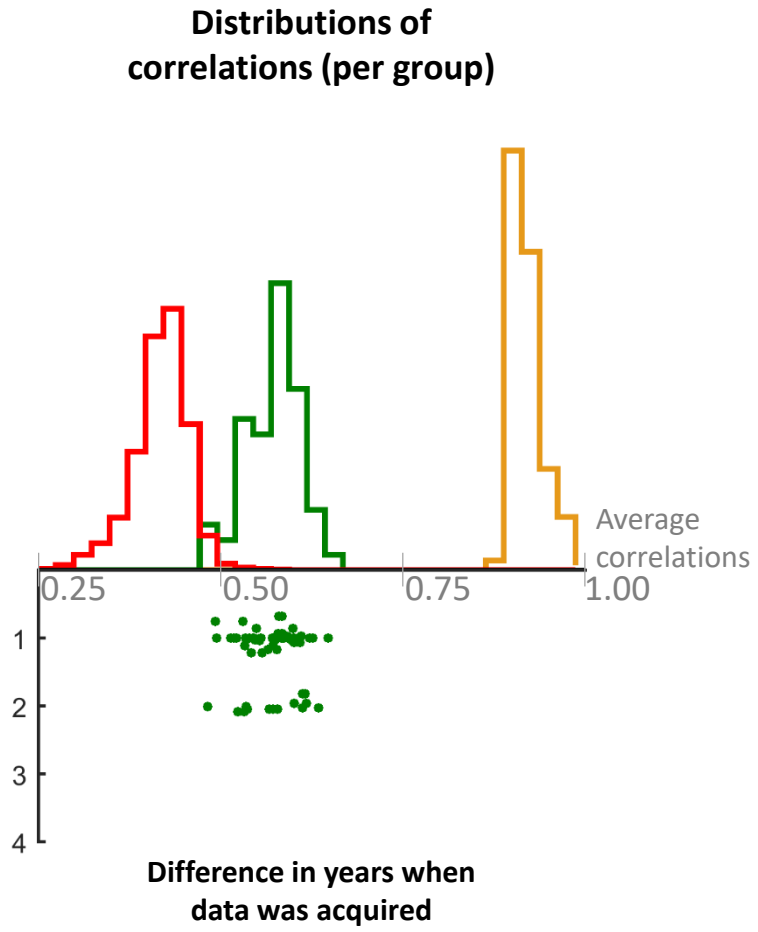




Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

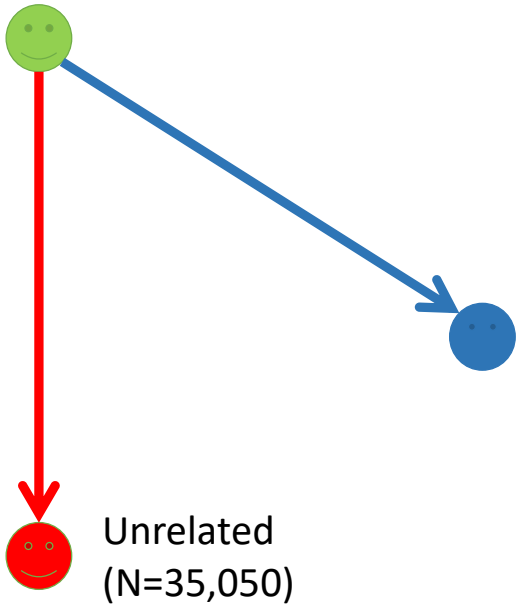
Characterization of individuals are stable (at least over a period of 2 years)





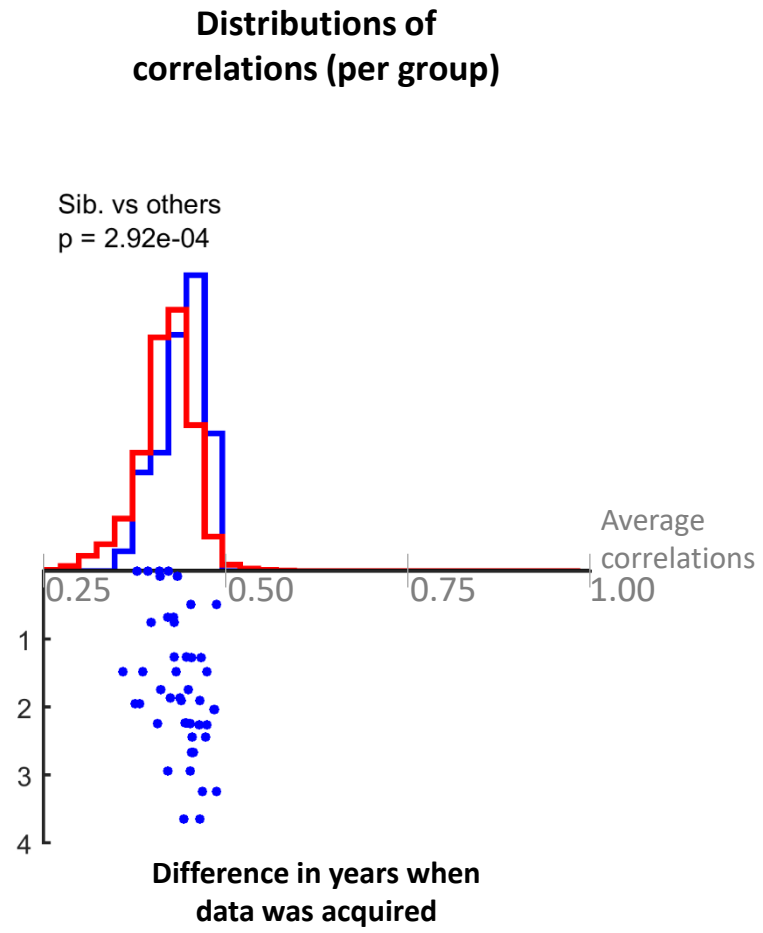
Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

Siblings cluster together higher than unrelated



Siblings
(N=46)

Unrelated
(N=35,050)



Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

These findings suggest that

The connectotype is similarly predictive in children as shown in adults,
across a wider timespan,
and some features appear to be familial

What if we now use multivariate statistics (instead of using the average correlation) to compare connectomes?

Can we identify heritable patterns of functional connectivity?

- Some mental disorders run strongly among families

- It might be useful to identify what is the “baseline” shared connectome across siblings?

There is evidence of similar thoughts among siblings

http://edition.cnn.com/2015/09/06/tennis/tennis-venus-serena-bouchard/

http://www.tampabay.com/news/politics/national/bush-dynasty-continues-to-impact-republican-politics/1248057

Datasets

OHSU

Data from 32 unique participants
5 mins of low-head movement of RS
7-15 yo, 60% males

Siblings (16 pairs)
16 families with 2 siblings each

Human Connectome Project

Data from 198 unique participants
1 hour of data each
22-36 yo, 45% males

79 pairs of siblings:
- 10 identical twins
- 11 non-identical twins
- 58 sibling non-twins

Approach

Within dataset

- Calculate functional connectivity
  - Connectotyping
  - Correlations

- Compared each participant pair
  - Connectotyping: predicting timecourses
  - Correlations: spatial correlations

- Train classifiers (SVM) to identify each pair of participants as siblings or unrelated

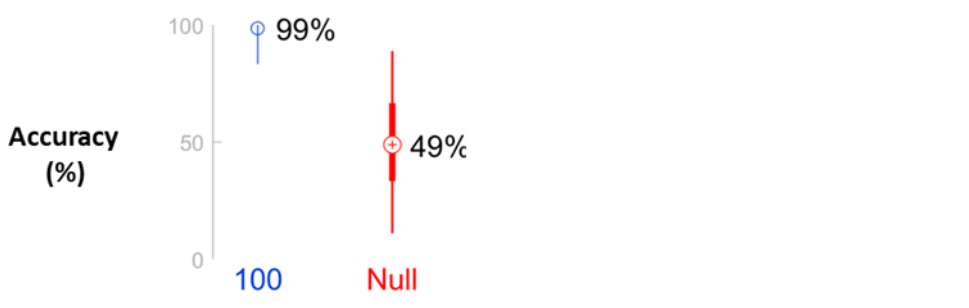
Between datasets

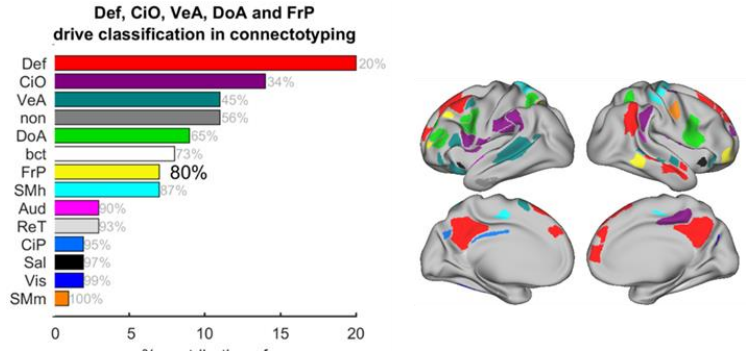
- Test classifiers' performance across datasets

Within OHSU results

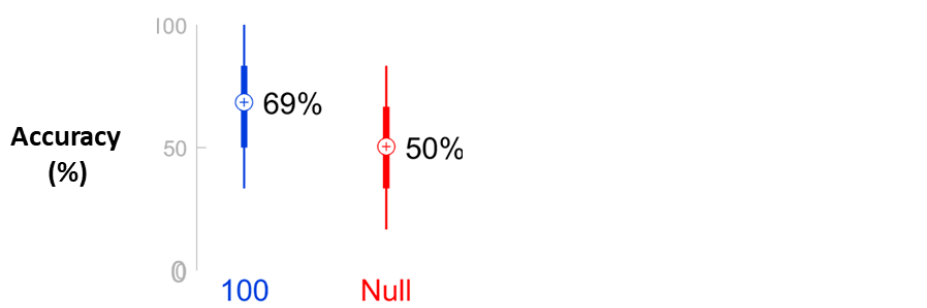
Out-of-sample performance

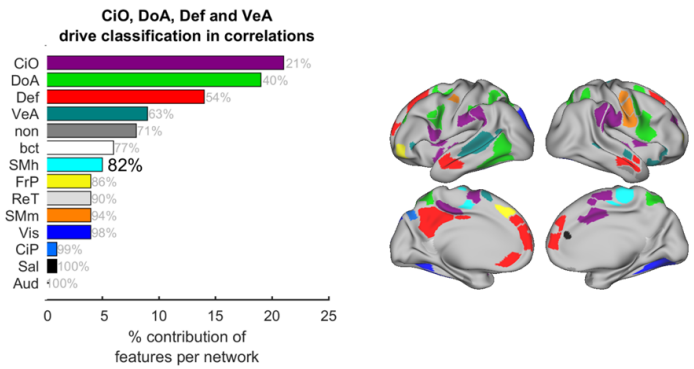
Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.





Connectotyping



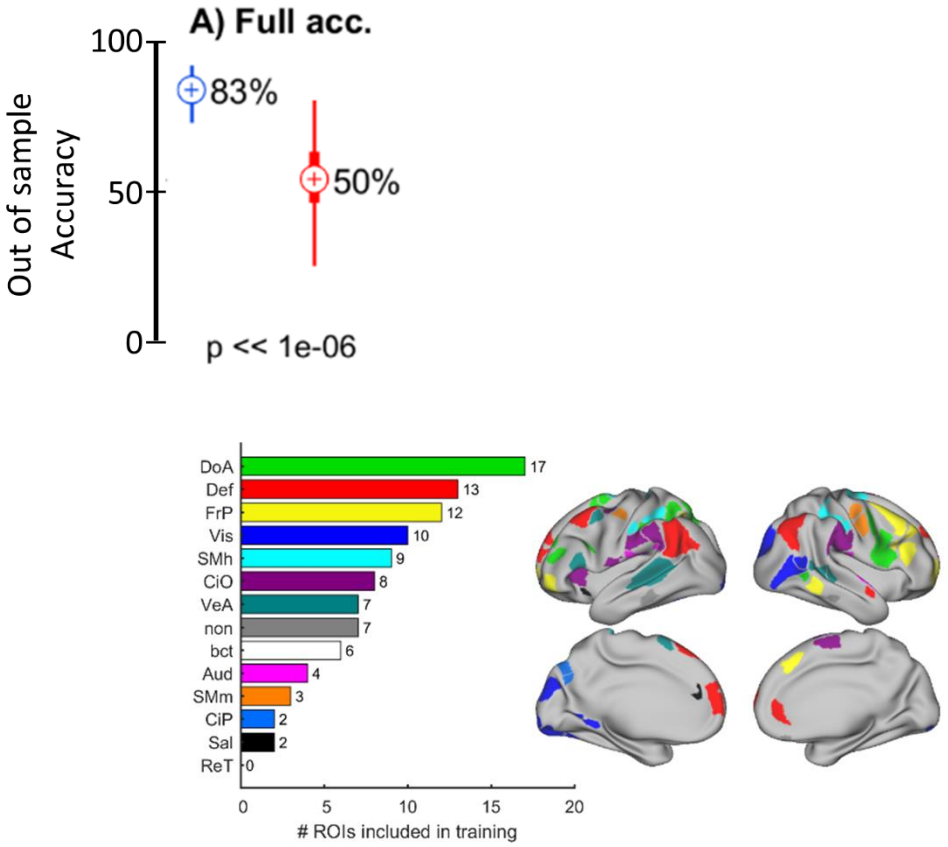


Correlations

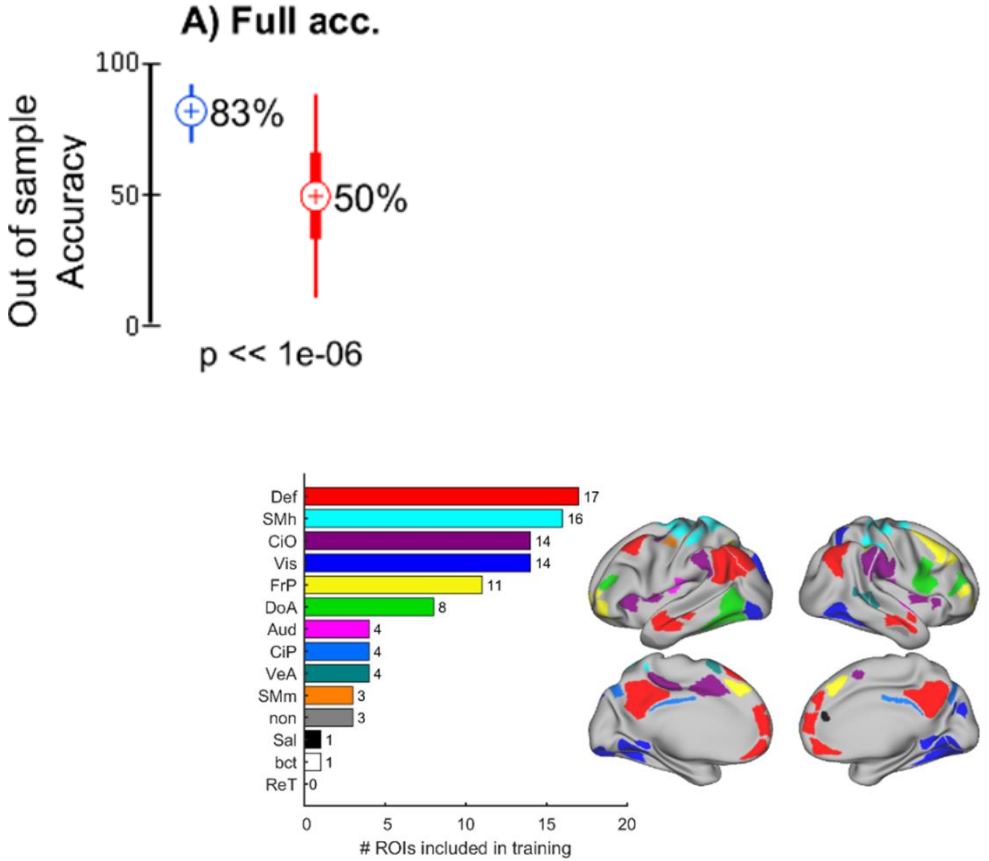
Within HCP results

Out-of-sample performance

Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.



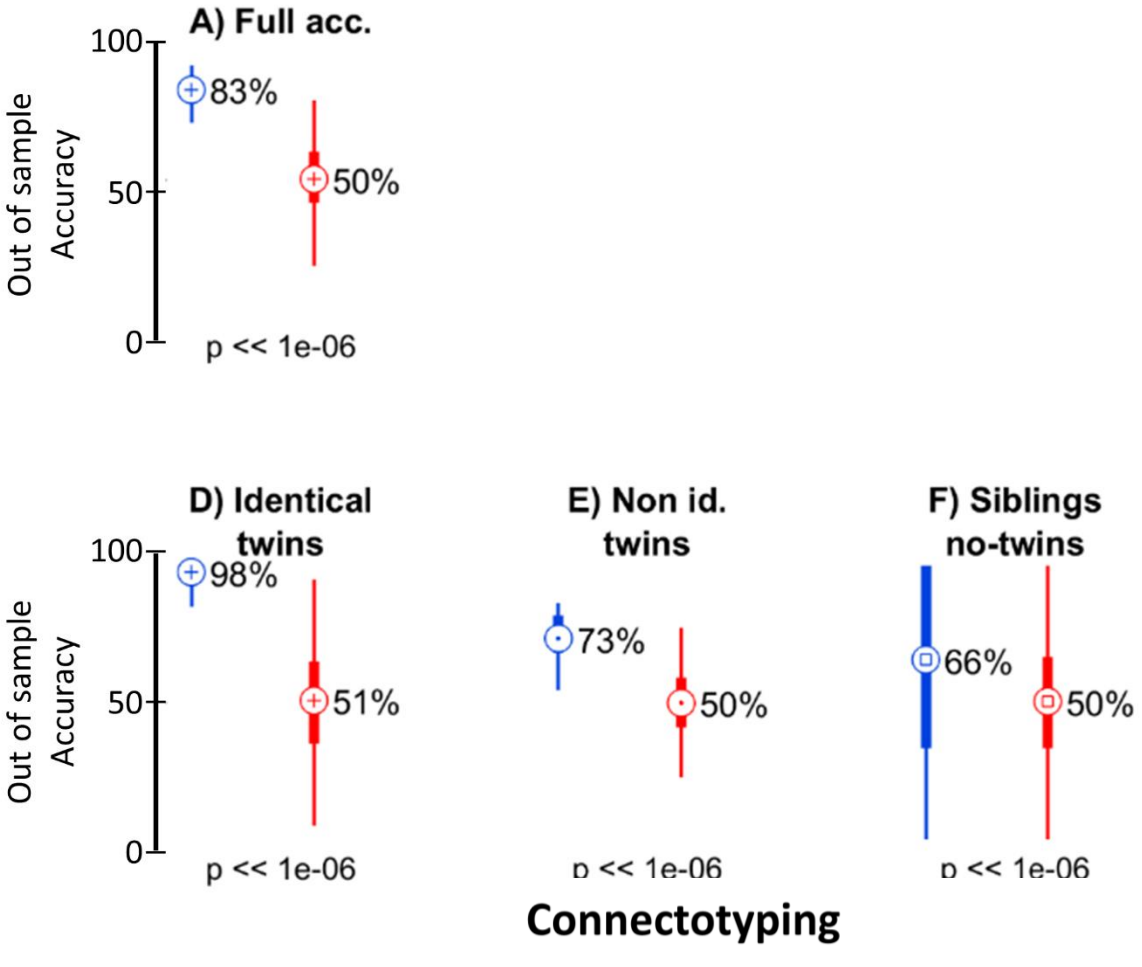
Connectotyping

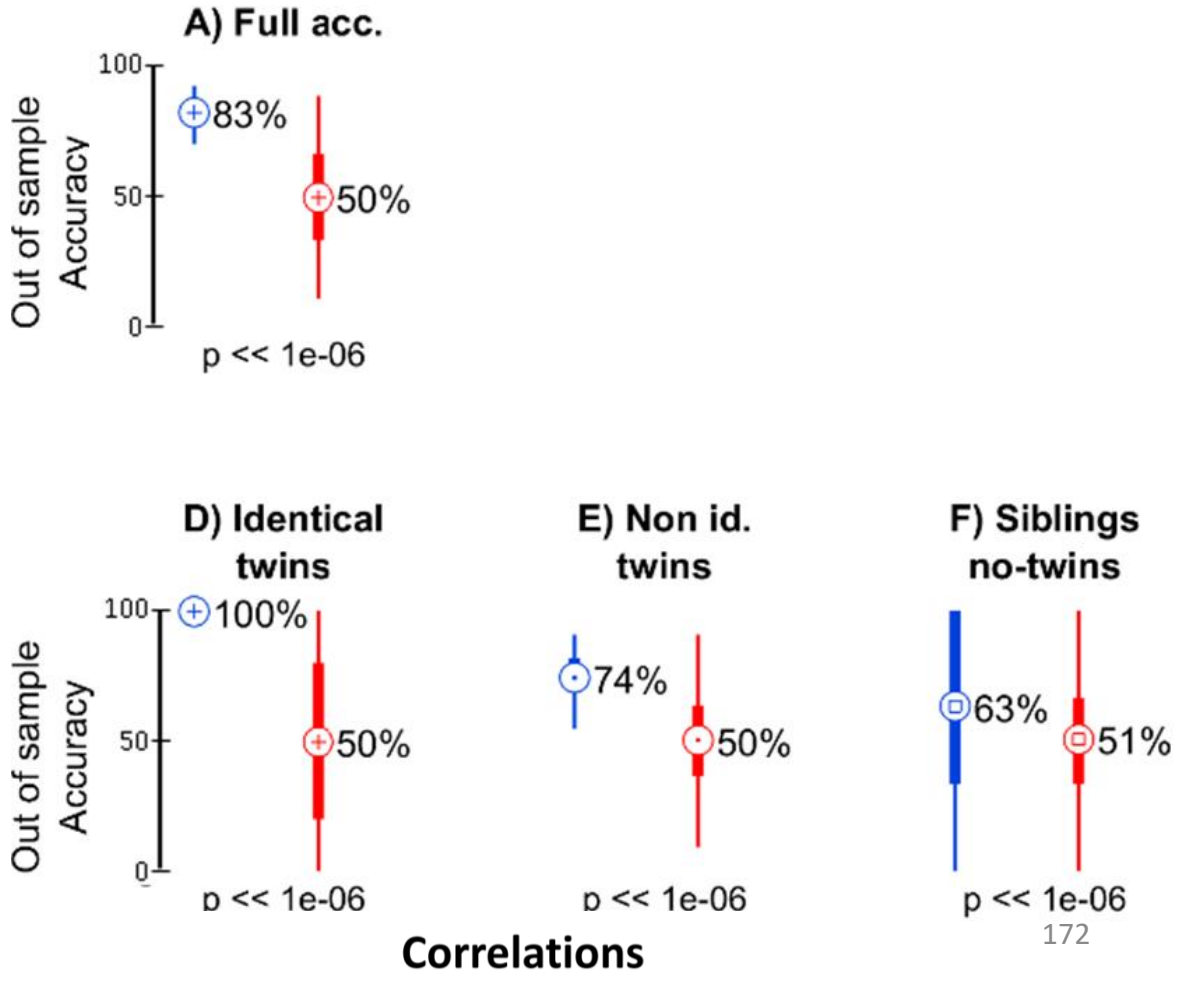


Correlations

Within HCP results
Out-of-sample performance

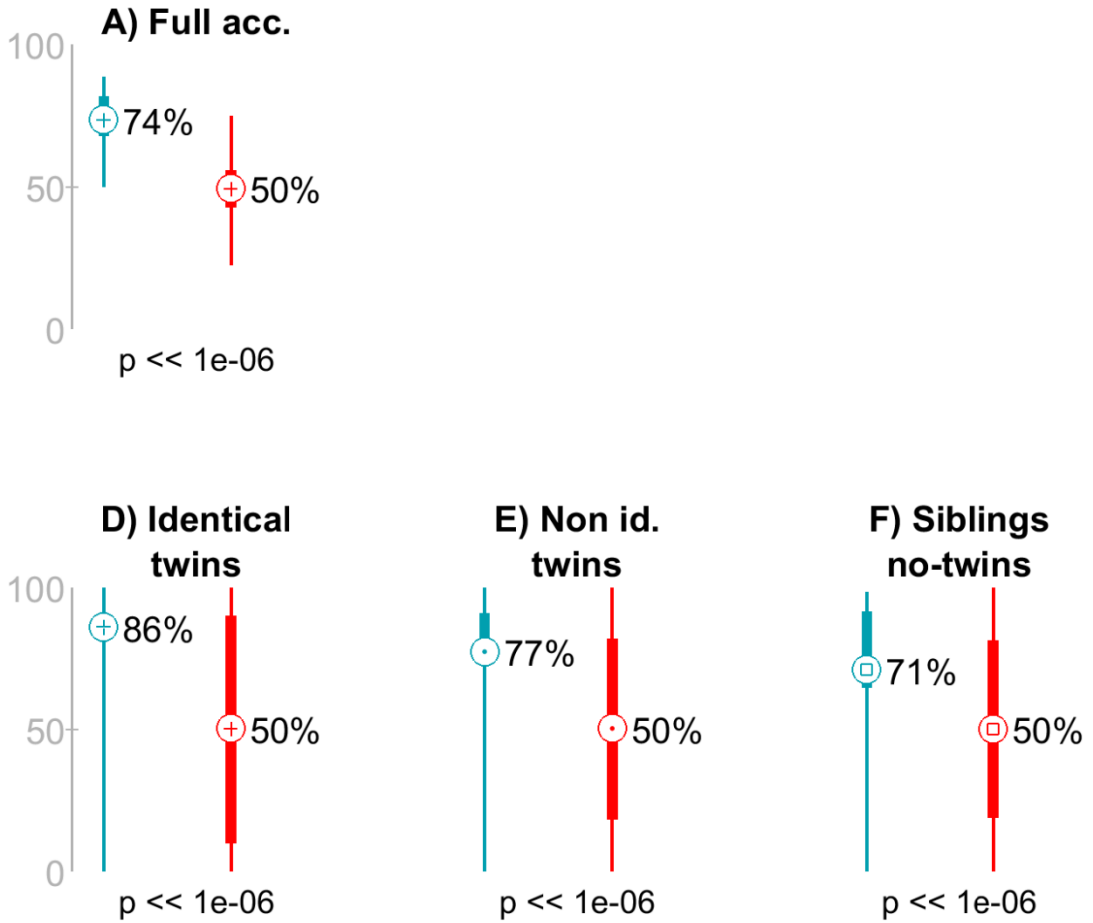
Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.





Predictions across datasets

Only connectotyping was able to predict kinship



Miranda-Domínguez O, *et al*. Heritability of the human connectome: A connectotyping study. Netw Neurosci 2018.

FOCUS FEATURE:
New Trends in Connectomics

Heritability of the human connectome: A connectotyping study

Oscar Miranda-Dominguez[1], Eric Feczko[1], David S. Grayson[1,4], Hasse Walum[5], Joel T. Nigg[1,2], and Damien A. Fair[1,2,3]

[1]Department of Behavioral Neuroscience, Oregon Health and Science University, Portland, OR, USA
[2]Department of Psychiatry, Oregon Health and Science University, Portland, OR, USA
[3]Advanced Imaging Research Center, Oregon Health and Science University, Portland, OR, USA
[4]Center for Neuroscience, University of California, Davis, Davis, CA, USA
[5]Silvio O. Conte Center for Oxytocin and Social Cognition, Center for Translational Social Neuroscience, Yerkes National Primate Research Center, Department of Psychiatry and Behavioral Sciences, Emory University, Atlanta, GA, USA

Keywords: Development, Heritability, Effective connectivity, MRI, Functional connectivity, Resting-state MRI

Rules of thumb

- In selecting predictor variables
  - Make sure predictor variables are related to outcome
  - Try to select variables with the lowest redundancy
  - It is better to have more observations than variables

- Regardless of modeling framework, you should use
  - Cross-validation to have an estimate of out-of-sample performance
  - Regularization to obtain more stable beta weights
  - Test performance on null data, to determine whether your models predict better than chance

Acknowledgements

DCAN Lab

AJ Mitchell
Alice Graham
Alina Goncharova
Anders Perrone
Anita Randolph
Anjanibhargavi Ragothaman
Anthony Galassi
Bene Ramirez
Binyam Nardos
Damien Fair
Elina Thomas
Eric Earl
Eric Feczko
Greg Conan
Johnny Uriarte-Lopez
Kathy Snider
Lisa Karstens
Lucille Moore
Michaela Cordova
Mollie Marr
Olivia Doyle
Robert Hermosillo
Samantha Papadakis
Thomas Madison

Members of the DCAN Lab

Funding: Parkinson's Center of Oregon Pilot Grant,
OHSU Fellowship for Diversity, Tartar Family grant, NIMH