fMRI Analysis

Sackler Institute 2011
How do we get from this to this?
How do we get from this to this?

And what are those colored blobs we’re all trying to see, anyway?
Raw fMRI data straight from the scanner
Raw fMRI data straight from the scanner
fMRI Signal

• What does real data look like?
fMRI Signal

- What does real data look like?
So what’s in our signal?

Task-Related Variability

Non-task-related Variability
Task-Related Variability
“SIGNAL”

Non-task-related Variability
“NOISE”
Signal to Noise Ratio (SNR)

Task-Related Variability

Non-task-related Variability
• Task-related variability (SIGNAL)
  - Difference in response related to experimental manipulation
• **Task-related variability (SIGNAL)**
  - Difference in response related to experimental manipulation

• **Non-task related variability (NOISE)**
  - Thermal noise, e.g., scanner heating
  - Variability in scanner function
  - Head motion effects
  - Temporal artifacts from acquisition
  - Physiological changes (heart rate, breathing)
    - Artifact-induced problems
    - Subject-induced noise - unexpected changes in attention, alertness, performance
• Task-related variability (SIGNAL)
  - Difference in response related to experimental manipulation

• Non-task related variability (NOISE)
  - Thermal noise, e.g., scanner heating
  - Variability in scanner function
  - Head motion effects
  - Temporal artifacts from acquisition
  - Physiological changes (heart rate, breathing)
    - Artifact-induced problems
    - Subject-induced noise - unexpected changes in attention, alertness, performance

GOAL OF PREPROCESSING  
↑ SNR
Preprocessing

• Separate steps to deal with separate, known sources of noise

1. Deal with temporal artifacts of acquisition - **Slice Timing Correction**
2. Dealing with movement-related noise – **Realignment**
3. Decreasing influence of noise in data generally - **Smoothing**
4. Dealing with the fact that each person’s brain is a different shape and size – **Alignment and Normalization**
Slice Timing Correction
The slice timing problem

TR = 4s
Collecting 4 slices

Collected at:

0s   1s   2s   3s
The slice timing problem

- EPI images collected at varying points in time after TR onset
- Correction shifts each voxel's time series so that they "appear" to have been captured at exactly the same time
The slice timing solution

The slices of one functional volume are shifted in time.

This is corrected by sinc (or linear) interpolation in time.
Realignment/ Motion correction
• Translations
  – X
  – Y
  – Z
• Rotations
  – Pitch
  – Roll
  – Yaw
Head movement: The problems

• 1. All movements
  – Images are not lined up

When data is combined across un-aligned scans, it can create artificial activations on the edges of the brain
Head movement: The problems

2. Sudden movements/Movement during scans:
   - Blurry images
Head movement: The solution

• Realignment
  – Algorithm that detects and corrects movement across scans
  – Realigns images to a reference image
  – Uses rigid-body transformation in the 6 planes
  – Writes out motion parameters
In theory ...
Reference Image

In practice ...
Movement: The Good
Movement: The Bad
Movement: The Ugly
Smoothing
Smoothing

• Reasons to smooth
  – Increases signal-to-noise ratio
  – Improves ability to make comparisons across subjects

Smoothing tool of choice:

Gaussian kernel
How does smoothing work?

- Replaces voxel value with a weighted average of itself and its neighbors

Before smoothing

After smoothing

Gaussian kernel
How does smoothing work?

• Smoothing decreases signal value in high-intensity voxels, while increasing the spatial extent of this signal.
Benefit #1 of smoothing

- Increases probability that across subjects, clusters will line up

Subject A activation
Intensity: 50

Subject B activation
Intensity: 50

Unsmoothed Version

Composite
Max intensity: 50
Benefit #1 of smoothing

- Increases probability that across subjects, clusters will line up

Subject A activation

Subject B activation

Smoothed Version

Composite

Intensity : 40

Intensity : 40

Max intensity : >50
Benefit #2 of smoothing

• ‘Real’ signal should consist of voxels with a high intensity value, *clustered together*
Benefit #2 of smoothing

- Noise should consist of high intensity values located *randomly* across space
Benefit #2 of smoothing

- Smoothing increases SNR by enhancing impact of clustered signal and suppressing impact of random noise

Probable signal overlaps with others

Probable noise drops out
Alignment and normalization
One subject’s dataset
2 problems

1. Within a single subject’s scan session, there is no guarantee that each of their scans will be lined up in the same space.

2. Everyone’s brain is a different shape and size compared to everyone else’s ... but we eventually want to combine data from several individuals to perform group statistics.

Goal: We want everything to line up!
2 solutions

1. Within a single subject’s scan session, there is no guarantee that each of their scans will be lined up in the same space.
   SOLUTION: ALIGNMENT / COREGISTRATION

2. Everyone’s brain is a different shape and size compared to everyone else’s ... but we eventually want to combine data from several individuals to perform group statistics.
   SOLUTION: NORMALIZATION
Alignment
Spatially align scans within each subject
Normalization – “Talairaching”
Recap of what we’ve done so far

1. Corrected for acquisition delays
2. Corrected as much as possible for motion
3. Increased SNR by smoothing
4. Aligned scans within each person
5. Warped data into a common space
Are we any closer to this?

• Yes, but we have to do a lot of math.

Tool of choice:
GENERAL LINEAR MODEL
General linear model in layman terms

- Signal in a voxel is a conglomeration of TASK-INDUCED CHANGE and noise.
General linear model in layman terms

• Signal in a voxel is a conglomeration of TASK-INDUCED CHANGE and noise.

• The GLM allows us to estimate what portion of the signal is best explained by our task (as well as by noise).
General linear model in layman terms

• The GLM itself is “the math” used to accomplish this.
General linear model in layman terms

• The GLM itself is “the math” used to accomplish this.
• During the GLM we are creating maps of **PARAMETER ESTIMATES** which represent how well our task explains signal.
How it works in layman terms

We tell the GLM everything we can about sources of variability in our signal.

“THE MODEL”
How it works in layman terms

We tell the GLM everything we can about sources of variability in our signal.

“THE MODEL”

The GLM tells us how much variability each of these sources explains.

“THE SOLUTION”
A very simple experiment

- 2 conditions – block design
- 1st Condition: Watching FACES
A very simple experiment

- 2 conditions – block design
- 1st Condition: Watching FACES
- 2nd Condition: Watching TAXIs
A very simple experiment

- 2 conditions – block design
- 1st Condition: Watching FACES
- 2nd Condition: Watching TAXIs
- And, there’s rest
Hypothetical Responses

Time

Faces Taxis Rest

Time

[Graph showing hypothetical responses over time with brain images and colored bars representing different conditions: Faces, Taxis, Rest. The graph includes a line chart depicting a pattern of responses.]
Hypothetical Responses

Faces  Taxis  Rest

Time

Face Responsive
Hypothetical Responses

Faces

Taxis

Rest

Time

Face Responsive

Face + House Responsive

Time
Hypothetical Responses

- Face Responsive
- Face + House Responsive
- Task-insensitive

Time

Faces  Taxis  Rest
It’d be good to tell the GLM about our task

- A model consists of a set of assumptions of the type:

  "I think a voxel that is into faces might have a time-series looking like this"

- and

  "I think a voxel that is into taxis might have a time-series looking like this"
The Model:
A Set of Hypothetical Time-series

- For a given voxel (time-series) we try to figure out just what type that is by "modelling" it as a linear combination of the hypothetical time-series.
The Estimation:
Finding the "best" parameter values

- The estimation entails finding the parameter values such that the linear combination "best" fits the data.
The Estimation: Finding the ”best” parameter values

- The estimation entails finding the parameter values such that the linear combination ”best” fits the data.
The Estimation: Finding the "best" parameter values

- The estimation entails finding the parameter values such that the linear combination "best" fits the data.
The Estimation:
Finding the "best" parameter values

- The estimation entails finding the parameter values such that the linear combination "best" fits the data.

Cool!

0.83
0.16
2.98
The Estimation:
Finding the ”best” parameter values

• And the nice thing is that the same model fits all the time-series, only with different parameters.

$\beta_1 \cdot + \beta_2 \cdot + \beta_3 \cdot$

0.68  0.82  2.17
The Estimation:
Finding the ”best” parameter values

- And the nice thing is that the same model fits all the time-series, only with different parameters.
The Estimation:
The format of data, model and parameters

- Same model for all voxels.
- Different parameter estimates for each voxel.
GLM converts the units of your data

One timeseries per voxel

One beta estimate for each input of the GLM per voxel
errorTS \approx \beta_1 \cdot + \beta_2 \cdot + \beta_3 \cdot

0.68 0.82 2.17

Whatever we do not explain in our model becomes part of the error term
Group analysis

• Runs statistical tests on the betas

• Each participant contributes one beta value to the analysis per voxel per condition (much easier to handle than timeseries)
  – Random effects group analysis (GOOD!)
Contrasts/GLTs are simple math

• Is the houses-taxis difference score consistently different from zero?
• This is the output of a one-sample t-test
• If significant in an area, then “region X is significantly more active to Houses than Taxis”