# Project idea

* It started with a paper showing GPT models straighten language trajectories.
* We wanted to see if the same effects are present in humans.
* Then the question is how to define curvature in human fMRI.
* First attempt is using coherence analysis as an approximate. Coherence is like correlation in the Fourier domain. Computing coherence between BOLD signals gives us an idea of the integration timescale. If higher coherence in lower frequency, that means signals exhibit consistency over larger timescales, which implies that brain regions are synchronized over longer periods of time.
* Our prediction is that lower areas like AC would have coherence centered on higher frequencies, compared to higher areas like PFC.
* Plan is to do this with both language data and vision data to observe the connections.

# Data/code

Surface data: download the AA directory into your pycortex database after you set that up, and then plot summary statistics for the data by using the following function and providing:

*def flatmap(subject, xfm, data, colormap = 'hot', vmin = 0, vmax = 0.5):*

*mask = cortex.db.get\_mask(subject, xfm)*

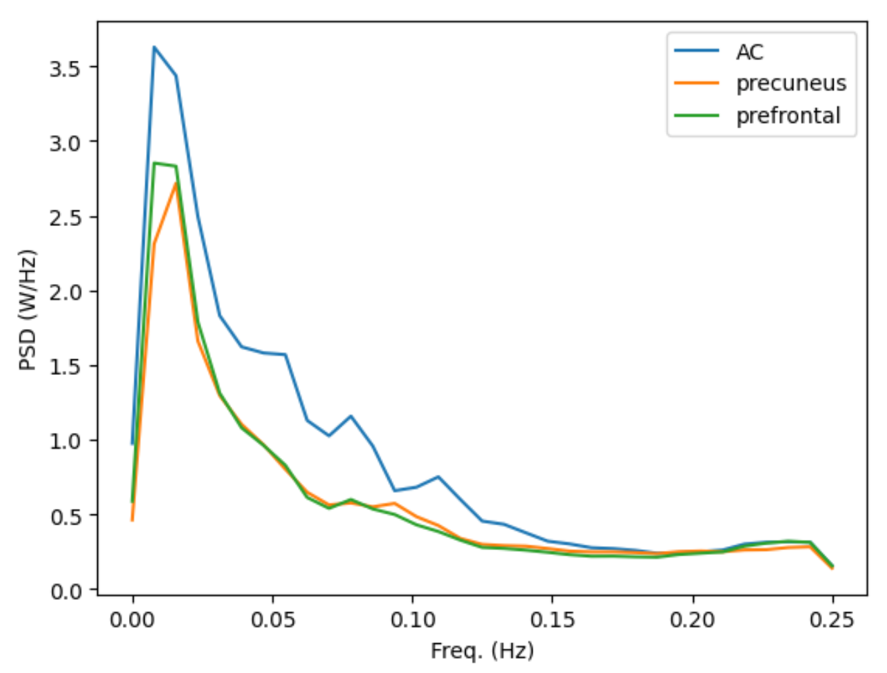
*data\_3d = cortex.unmask(mask, data)*

*data\_vol = cortex.Volume(data\_3d, sname, xfm, cmap='hot', vmin = vmin, vmax = vmax)*

*cortex.quickshow(data\_vol, with\_labels = True, with\_colorbar = True, with\_rois = True, thick = 1);*

# 2/22/24

* **Average PSD of each frequency bin for whole brain**
  + The peak should reflect higher SNR



* **PSD for selected SOIs**
  + AC has bigger stimulus related signals, so this pattern is reasonable??
  + Comparing different areas: look at the ratio
  + Look at overall SNR and by frequency
  + Normalize and compare – the center of mass is different for each area
  + Right now, the data I have was z-scored for the signal in each voxel for each repeat (standard preprocessing). Mean = 0, SD = 1 for each voxel across one stimulus for each repeat.
  + When average across the different repetitions, mean is still 0, but the SD is not 1 anymore (it will be less than 1); the variance corresponds to the SNR, if we don’t re-normalize. As a result, when computing the PSD, we will get more power for some areas than others because the SNR is higher for those areas
  + So if we re-normalize after trial averaging, we can find in which frequencies these areas are different from each other in power. Multiply by the inverse of the SD
  + Potential problems with re-normalizing: in higher frequencies, all areas are basically the same (just noise). After re-normalizing, area under the curve would be the same, then AC will look like it has lower power in higher frequencies, but that’s not a real effect.
  + How are these plots supposed to look like after re-normalizing?
  + Re-normalize, plot on log axes, compare the slops. First guess is that the time scale information will be reflected in these slops.



* **Center of mass plots (PSD)**
  + Re-normalizing shouldn’t matter for this plot
  + This looks like an SNR map. Darker areas are areas with good signal, higher SNR
  + We expect to see higher SNR in areas related to the stimulus
  + The bright yellow areas are areas we typically don’t get any brain data at all
  + Noise is typically observed in higher frequencies
    - 1/f noise (physiological)
    - White noise (measurement)
  + We should make a filter that filters out high-frequency noise

*What we need to do is disentangling the PSD measure and the center of mass measure from SNR*

* + Make a scatterplot of overall SNR and center of mass for each voxel, there should be a relationship, but the deviation away from this relationship is what’s interesting. We should get rid of anything that can be explained purely by SNR, look at the residuals



* + Coherence is an SNR metric
* **Coherence by frequency**
  + When computing coherence in the classic way, using the coherence function, it doesn’t assume the signals are phase-aligned. Instead, for each frequency it tells you the magnitude of the alignment and the relative phase. But here we DO want to assume phases are aligned, because it is a systematic measure of responses to the same stimulus
  + Phase alignment: the synchronization of signals in time. When two signals are phase-aligned, it means their peaks, troughs, and oscillations occur at the same points in time. Since we are dealing with BOLD responses to the same stimulus, the responses are expected to occur at similar times relative to stimulus presentation, which means the phase of the signals should be consistent across trials. In coherence analysis, phase alignment matters because coherence measures the consistency of phase and amplitude relationships between signals across trials. Without phase alignment, coherence may still detect some relationships like shared frequencies, but it won’t necessarily capture the fact that the signals are consistently aligned in time. We should account for the timing of these responses relative to the stimulus and align them before computing coherence. Otherwise, we lose information about the consistent phase relationship between the signals.
  + Use Alex’s code to compute coherence with phase-alignment



* **Center of mass plot (coherence)**
  + There’s still SNR bias, meaning the highest values in this plot (bright yellow) do not reflect high SNR in high frequencies, but reflect noise.
  + I’m computing coherence between one pair of repeats, but this is double counting the noise because both signals are noisy.
  + Average all 10 repeats and compute coherence with one of them (under counting noise) – use this as an upper bound of the true coherence



* Another thing to consider: Alex recently realized there’s a strange onset effect happening in AC, where over the first minute to 2 minutes of each recording, they observe ramping responses in AC that ramp down, which is repeatable across presentations and also across different stimuli. This could show up as a low frequency effect predominately in AC.
  + Solutions: throw out the first 50-60 TRs, see what happens



10 repeats of a 90TR stimulus, shorter than the vision data we have, but subjects were fixated in this. The vision data we have now are free viewing and 2 repeats

# 10/25/24

New analysis: fit AR(1) directly to data to estimate timescale of signals

* I fitted AR(1) to raw language data (1 repeat x 251 TR x ROIs) directly. By doing this we already observed a trend that we are expecting: earlier areas had smaller AR(1) coefficients compared to higher areas
* I fitted both AR(1) and AR(2), AR(2) had better fit, but that could be due to overfitting the noise.

Alex’s suggestion:

* We want to fit signals, not noise. To get rid of noise, we should do a cross-AR fit.

## 

* I fitted AR(1) to raw language data (1 repeat x 251 TR x ROIs) directly. By doing this we already observed a trend that we are expecting: earlier areas had smaller AR(1) coefficients compared to higher areas
* I fitted both AR(1) and AR(2), AR(2) had better fit, but that could be due to overfitting the noise.
* My question: to get rid of noise, why can't we just average the 10 repeats and fit to AR(1) directly? A: the reason is that averaging only gets rid of so much noise. averaging across 10 repeats will drop the strength of (gaussian-like) noise by a factor of sqrt(10)=~3.1. it’s not noiseless, merely less noisy. So the AR model on averaged responses will be biased to capture the shape of the noise (divided by ~3) and the signal together.
* Cross-fitting should make it so that AR model is only made noisy and not biased – in the case of infinite length data, the model weights should converge to the true weights of for the signal.