INFERRING EFFECTIVE CONNECTIVITY FROM HIGH-DIMENSIONAL ECOG RECORDINGS

CHRIS ENDEMANN
RESEARCH INTERN,
BANKS LAB
DEPARTMENT OF
ANESTHESIOLOGY
UW – MADISON, SMPH

THE BRAIN AS A NETWORK OF SPECIALIZED COMPUTING COMPARTMENTS

Neuroscience has come a long way in terms of revealing how individual cortical regions respond to various stimuli/tasks/etc.

However, we've barely scratched the surface in terms of understanding how these regions function together in concert i.e. how the brain functions as an integrated computational system.

We can begin to reveal the brain's systems-level algorithms by measuring the strength and direction of information flow between specialized functional regions

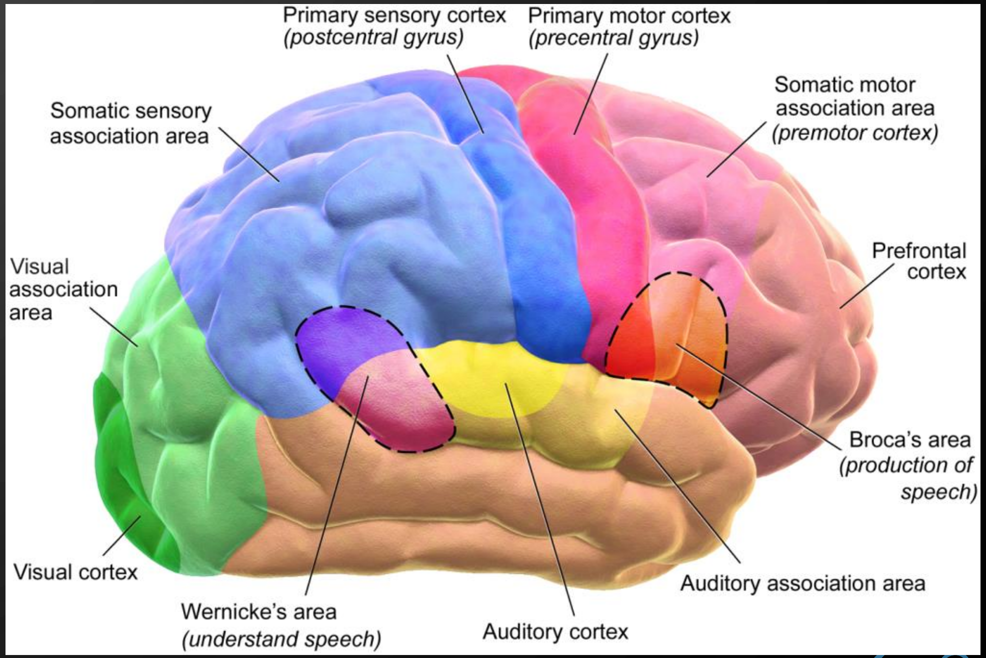
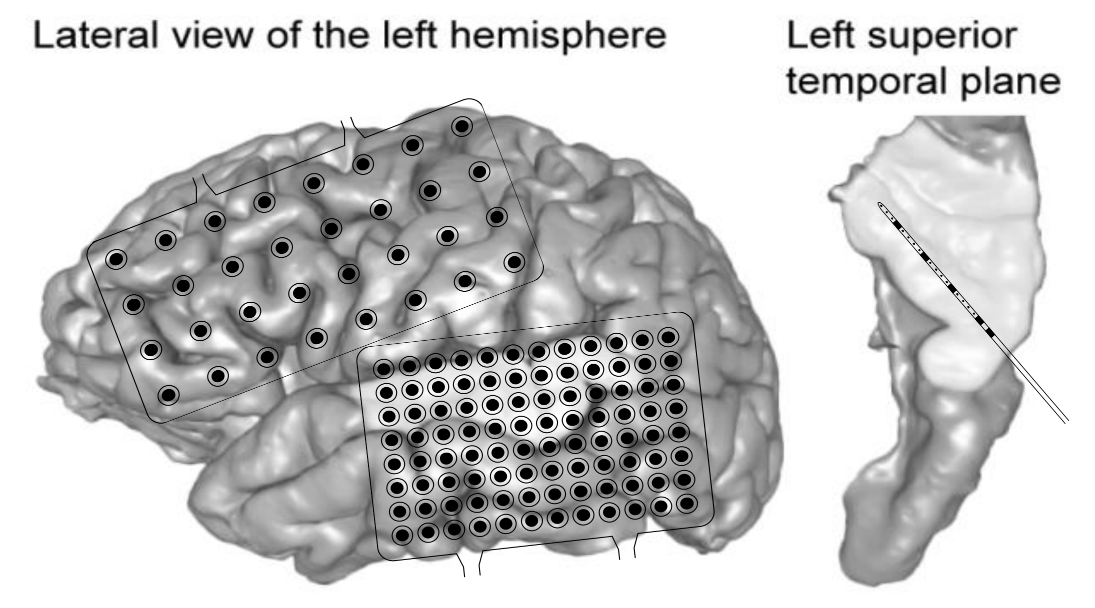
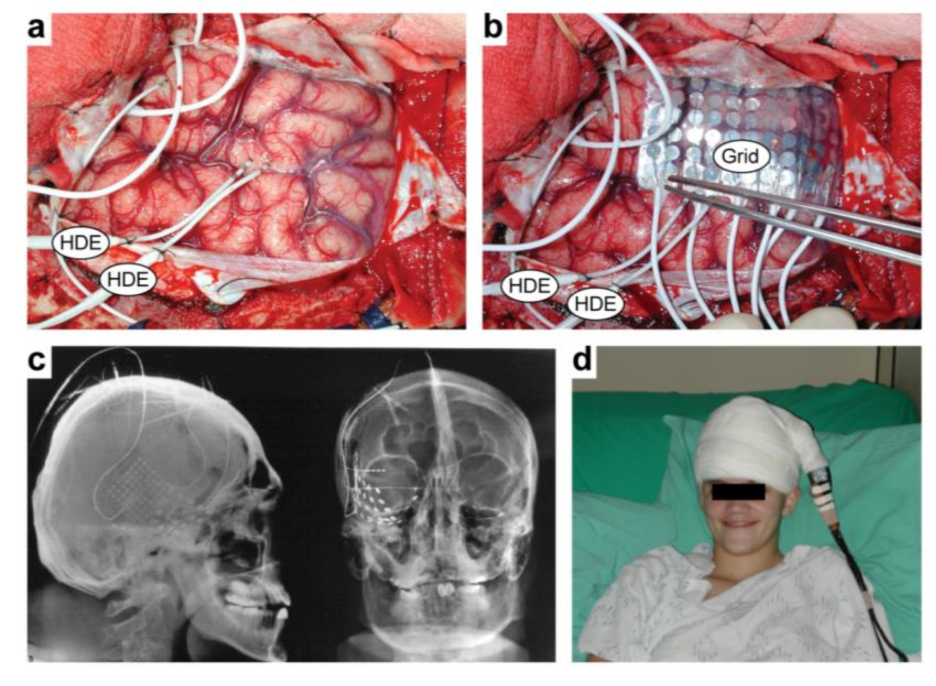


Image source: https://en.wikipedia.org/wiki/Human_brain

Electrocorticography (ECoG):
Direct intracranial recording in neurosurgical patients



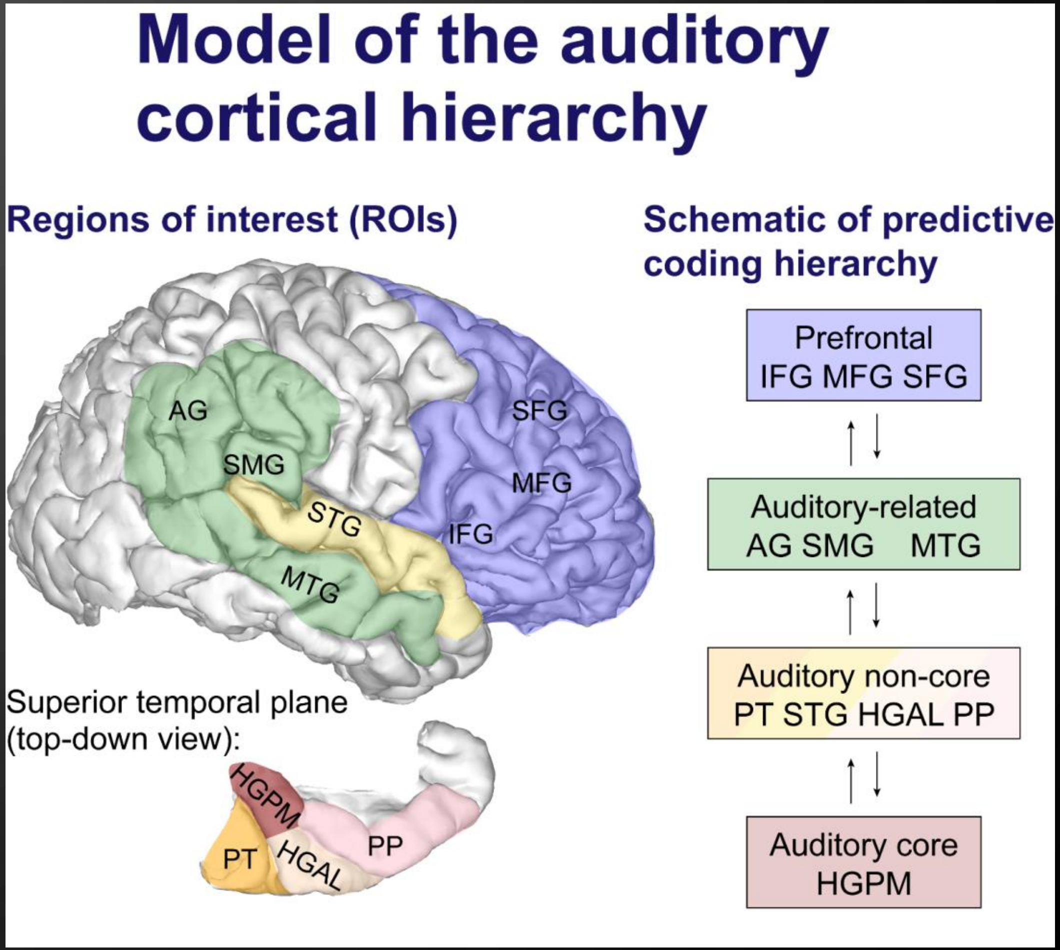


Howard, Nourski & Brugge (2012). In: The Human Auditory Cortex, pp. 39-67.

Human Brain Research Laboratory (Matthew A. Howard, MD, Director)

The University of Iowa

- 100-200 channels per patient
- 30-40 ROIs
- Electrode coverage allows us to study how auditory sensory information is computed and transmitted across various functional regions



Model of the auditory cortical hierarchy

Regions of interest (ROIs)

Schematic of predictive coding hierarchy

TRACKING THE FLOW OF INFORMATION BETWEEN SPECIALIZED FUNCTIONAL REGIONS

- Preferred approach is to assess ***Granger Causality*** (GC) between nodes (recording channels) of the brain
- Crux of GC: Do past values of one/more variables predict the present of another variable?
- Strength of causal influence between variables is referred to as ***effective connectivity*** in neuroscience

X Granger Causes Y

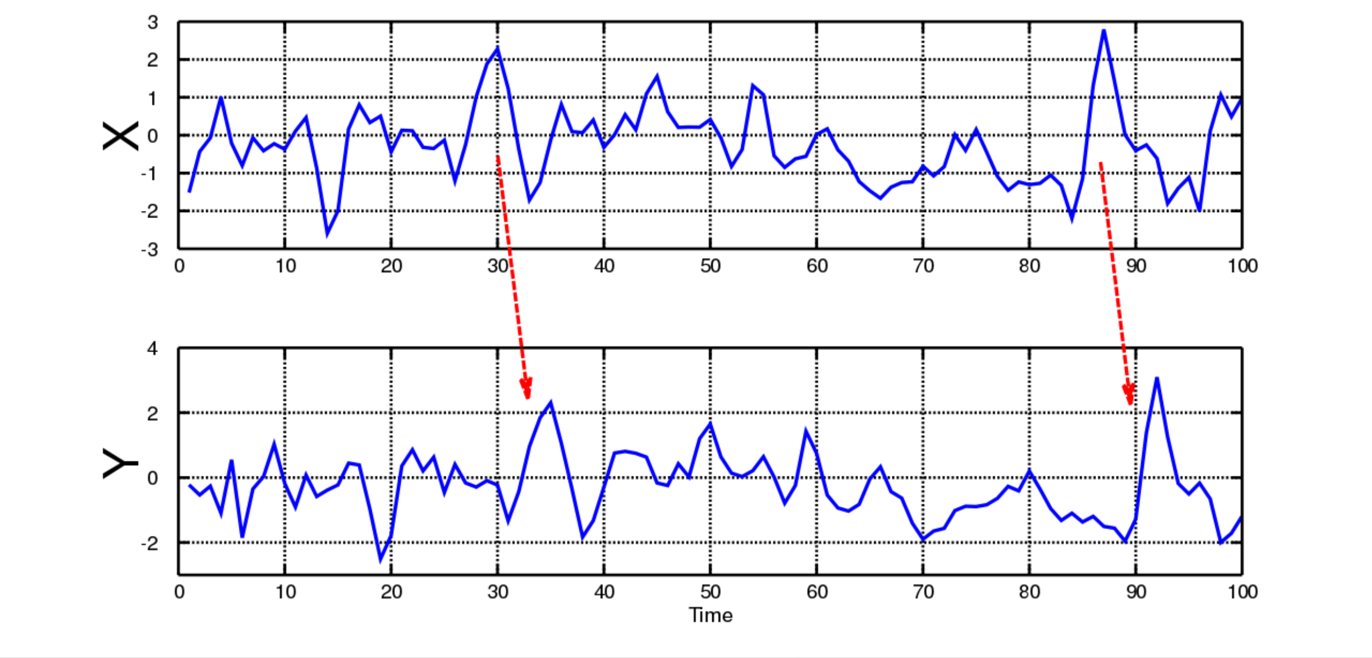


Image source: https://commons.wikimedia.org/wiki/File:GrangerCausalityIllustration.svg

CAN MEASURE GC USING VECTOR AUTOREGRESSIVE (VAR) MODELS

$$Y(t) = \sum_{k=1}^{p} \mathbf{A}(k) Y(t-k) + U(t)$$

$Y(t)$	Vector of observed values for all Q variables at time *t*
p	Model-order (i.e. how many past time samples or lags to use to predict the present sample)
$\mathbf{A}(k)$	Q-by-Q autoregressive parameter matrix at lag=*k*. Estimated via model fitting.
$U(t)$	Innovation noise (i.e. the difference between the model's predictions and observed data at time *t*)

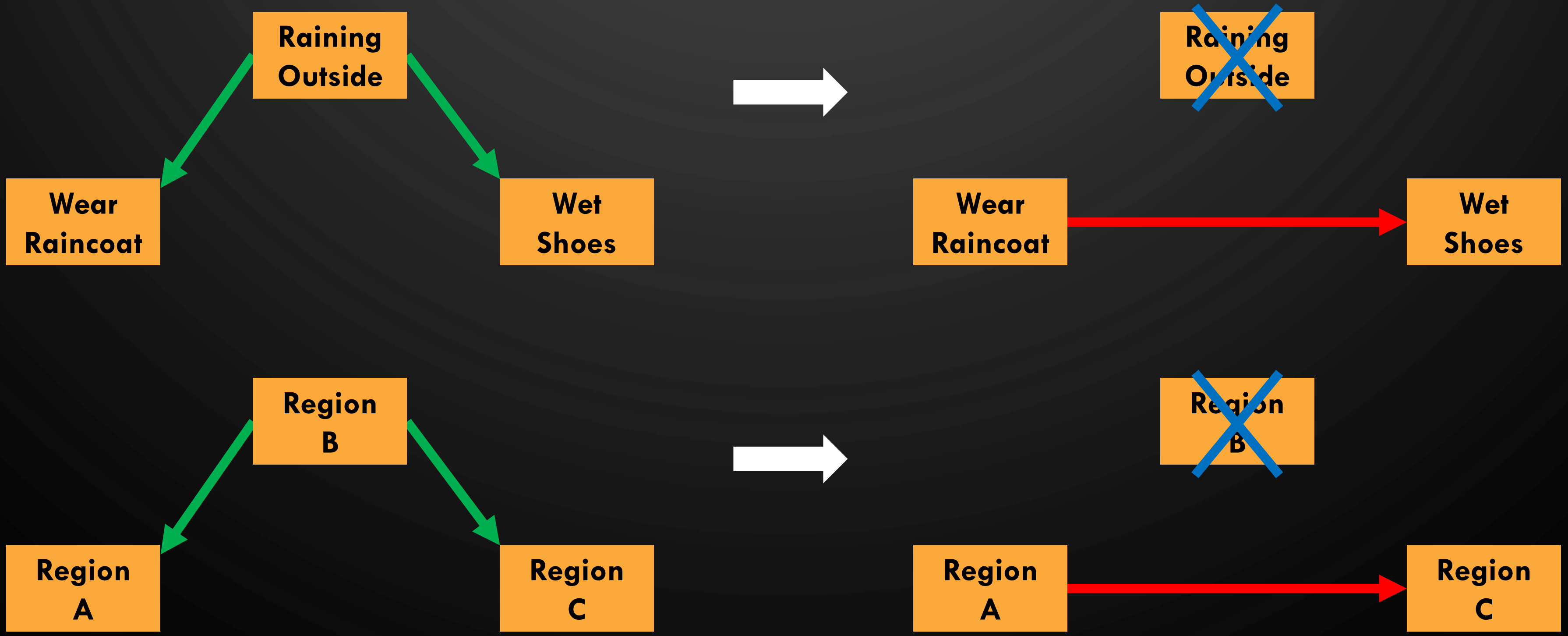
$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{Q,t} \end{bmatrix} = \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,Q} \\ a^1_{2,1} & a^1_{1,2} & \cdots & a^1_{2,Q} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{Q,1} & a^1_{Q,2} & \cdots & a^1_{Q,Q} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ \vdots \\ y_{Q,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,Q} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,Q} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{Q,1} & a^p_{Q,2} & \cdots & a^p_{Q,Q} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ y_{2,t-p} \\ \vdots \\ y_{Q,t-p} \end{bmatrix} + \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{Q,t} \end{bmatrix}$$

$$N = \sum_{k=1}^{p} Q^2$$

Model parameter count, *N*, grows quadratically with channel count

EVERYTHING IS CONNECTED, MAN…. ESPECIALLY IN THE BRAIN

- Most connectivity analyses focus on small sub-networks (< 10 channels) due to computational challenges and model-overfitting concerns
- Manually excluding variables risks the detection of spurious causal connections



Raining Outside

Wear Raincoat

Wet Shoes

~~Raining Outside~~

Wear Raincoat

Wet Shoes

Region B

Region A

Region C

~~Region B~~

Region A

Region C

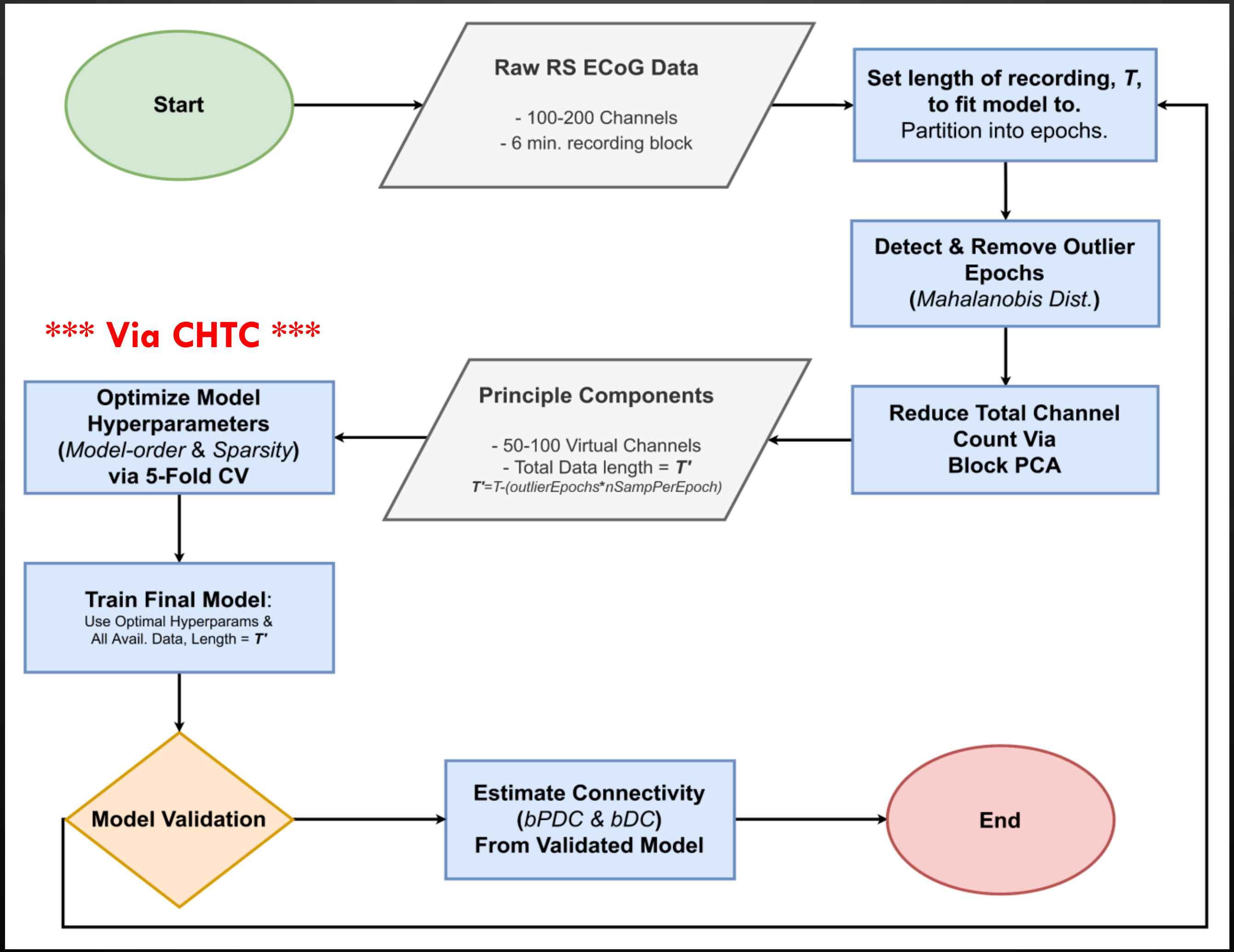
OUR LAB'S RESEARCH GOALS

1. Construct analysis pipeline capable of modeling effective (i.e. causal) connectivity from high-dimensional (100-200 channels) recordings

2. Assess strength and direction of information flow between specialized functional regions across the cortical hierarchy
    1. Which nodes drive the activity of others?

3. Assess how connectivity changes across awareness states during sleep and anesthesia.

METHODOLOGICAL CHALLENGE – DEVELOP PIPELINE TO EFFICIENTLY MODEL LARGE-SCALE (**100-200** CHANNELS, **DOZENS** OF ROI’S) EFFECTIVE CONNECTIVITY NETWORKS



Start

Raw RS ECoG Data

- 100-200 Channels
- 6 min. recording block

Set length of recording, *T*, to fit model to.
Partition into epochs.

Detect & Remove Outlier Epochs
(*Mahalanobis Dist.***)**

Reduce Total Channel Count Via Block PCA

Principle Components

- 50-100 Virtual Channels
- Total Data length = ***T’***

T’*=T-(outlierEpochs*nSampPerEpoch)*

***** Via CHTC *****

Optimize Model Hyperparameters
(*Model-order* & *Sparsity***)**
via 5-Fold CV

Train Final Model:
Use Optimal Hyperparams & All Avail. Data, Length = ***T’***

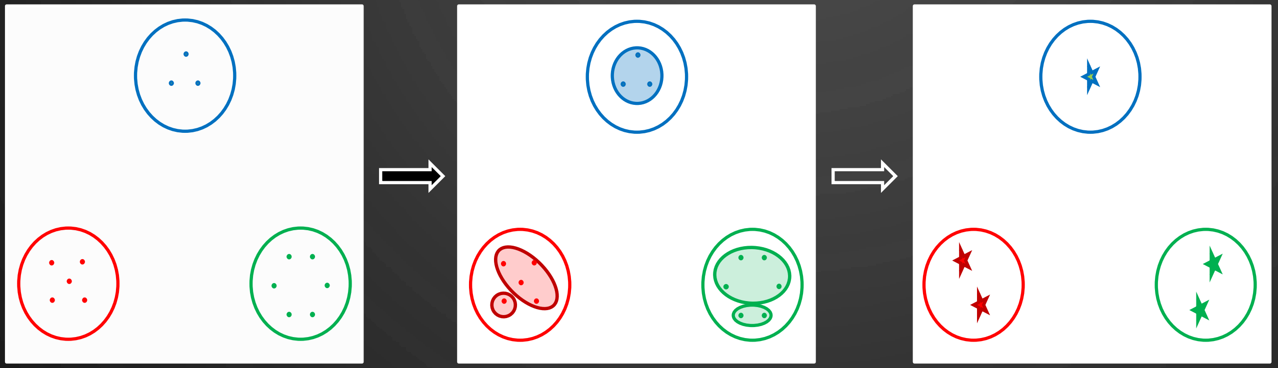
Model Validation

Estimate Connectivity
(*bPDC & bDC***)**
From Validated Model

End

HIGH-DIM. MODEL FITTING: APPLY DIM-REDUCTION TECHNIQUES TO PREVENT OVERFITTING

Pre-Process Data: Block PCA Run on 3 ROIs



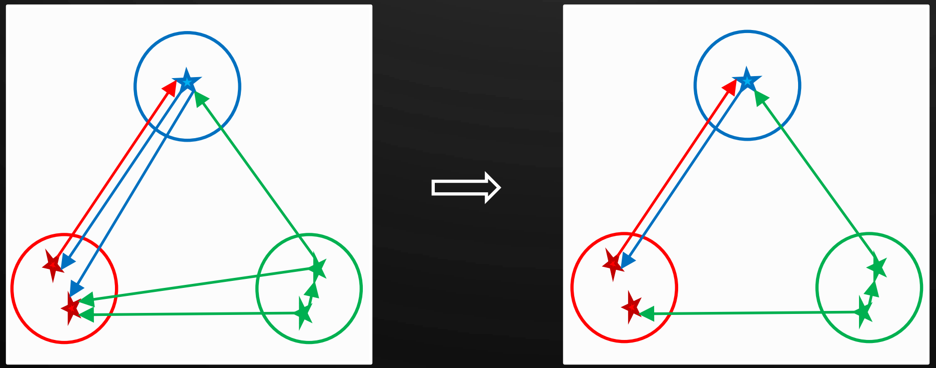
○ Single ROI

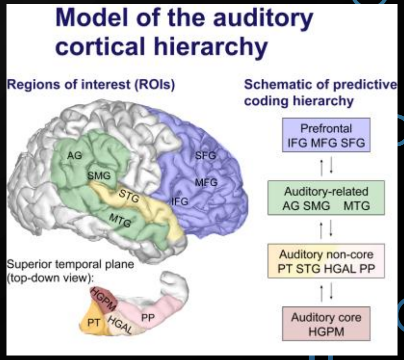
• Channel

★ Principle Component (Virtual Channel)

Apply Regularization Technique, **Group Lasso**, To Eliminate Weak/Redundant Connections (i.e. VAR model coeficients)

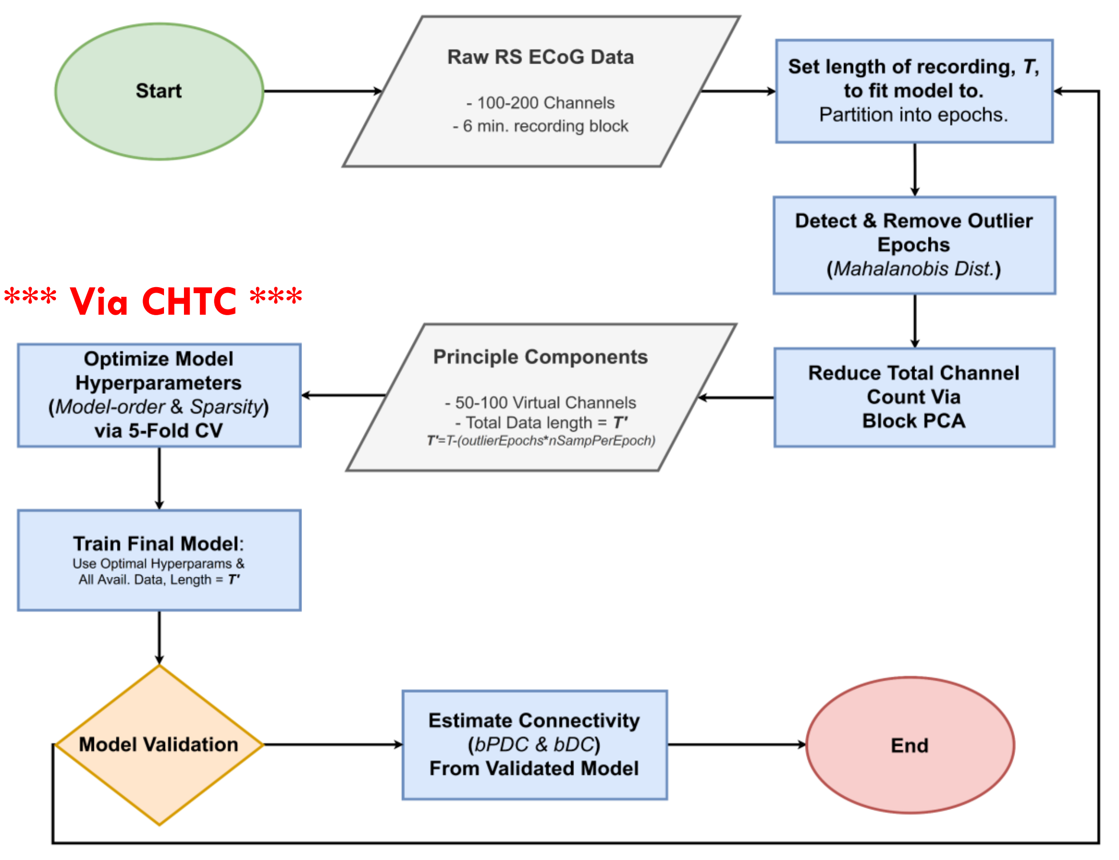
→ Adds Additional Hyperparameter To Model, ***Sparsity Weight***





METHODOLOGICAL CHALLENGE – DEVELOP PIPELINE TO EFFICIENTLY MODEL LARGE-SCALE (**100-200** CHANNELS, **DOZENS** OF ROI'S) EFFECTIVE CONNECTIVITY NETWORKS

- Primary computational burden arises from optimizing model hyperparameters
  - ***Model-order:*** How many lags to use to predict the present value of each channel
  - ***Sparsity Weight:*** How many model-coefs/connections to remove during model-fitting
- Optimize hyperparameters via 5-fold Cross-validation



Start

Raw RS ECoG Data
- 100-200 Channels
- 6 min. recording block

Set length of recording, *T*, to fit model to. Partition into epochs.

Detect & Remove Outlier Epochs (*Mahalanobis Dist.*)

Reduce Total Channel Count Via Block PCA

Principle Components
- 50-100 Virtual Channels
- Total Data length = ***T'***

***T'**=T-(outlierEpochs*nSampPerEpoch)*

***** Via CHTC *****

Optimize Model Hyperparameters (*Model-order* & *Sparsity*) via 5-Fold CV

Train Final Model: Use Optimal Hyperparams & All Avail. Data, Length = ***T'***

Model Validation

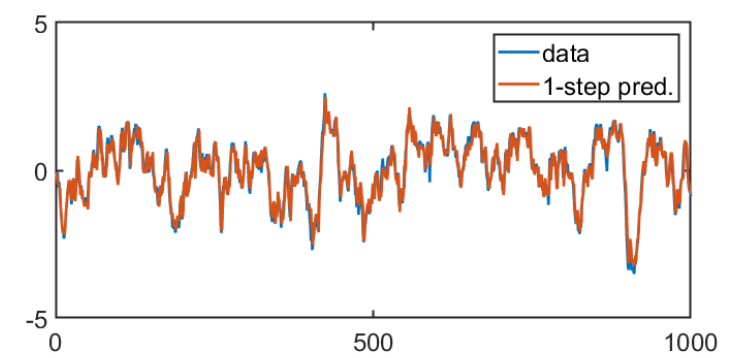
Estimate Connectivity (*bPDC* & *bDC*) From Validated Model

End

CROSS-VALIDATION PROCEDURE: “GRID-SEARCH”

Optimizing single model…

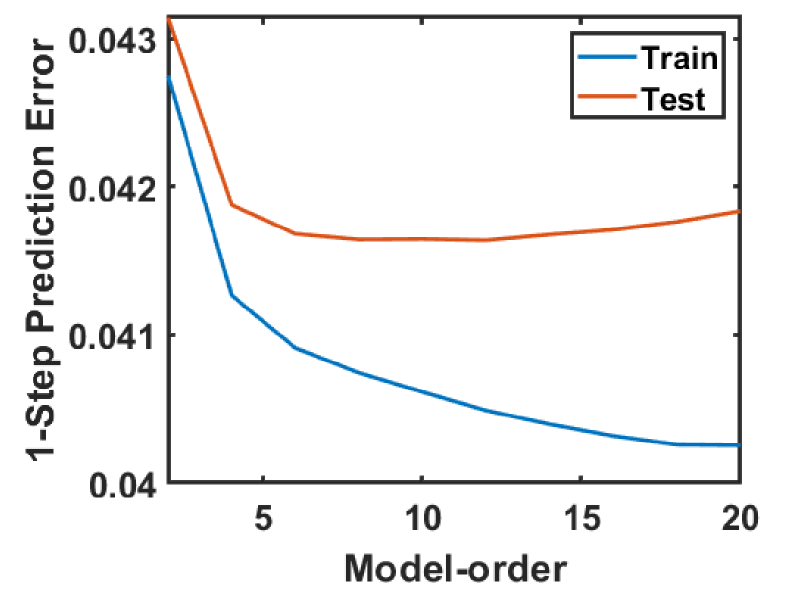
- 1-minute of recording data
- 50-100 virtual channels → Fit each channel individually (using history of all channels) and stitch together model coefficients at the end



- *K* = 5-Fold Cross-validation (train/test splits)
    - 3-5 ***model-orders*** to evaluate
        - 5-10 ***sparsity weights*** to evaluate

$100_{\text{Ch}} * 5_{\text{Folds}} * 5_{\text{Model-orders}} * 10_{\text{SparsityLvls}} = \mathbf{25{,}000}$

single-channel models!



GROUPING (SMALL) JOBS CAN REDUCE TOTAL RUNTIME

$100_{\text{Ch}} * 5_{\text{Folds}} * 5_{\text{Model-orders}} * 10_{\text{SparsityLvls}} =$ **25,000 single-channel models**

1. For a given model-order and training fold, can run models at all sparsity levels in ~1-2 hours

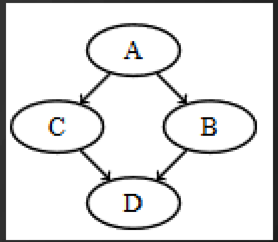
2. Rather than running many individual jobs (~6-12 min. each), group into one job submission

$25{,}000 \,/\, 10_{\text{SparsityLvls}} \rightarrow$ **2500 total jobs**

Avoids queuing more jobs than needed → Reduces total runtime by avoiding unnecessary job queues, file transfers, etc.

DIRECTED ACYCLIC GRAPH (DAG) UTILIZATION

Use DAG to specify order of jobs, e.g. stitching channel coefs back together after all single-channel models are fit

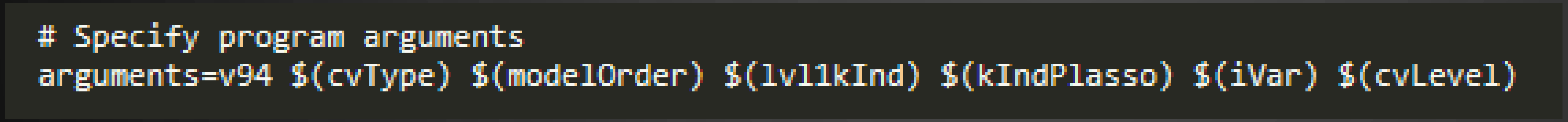


```
1. For iFold=1:K
    1. For modelOrder=modelOrderRange
        1. For iCh=1:nCh
            1. fitSingleChCoefs(iFold,modelOrder,iCh,sparsityRange)
        2. stitchTogetherChCoefs()
        3. measureFoldErr(iFold,modelOrder,sparsityRange)
2. setOptimalHyperparams_trainFInalModelAllData()
```

One additional CHTC feature that might be helpful is some sort of DAG visualization tool to help debug large DAGs that are incorrectly specified.

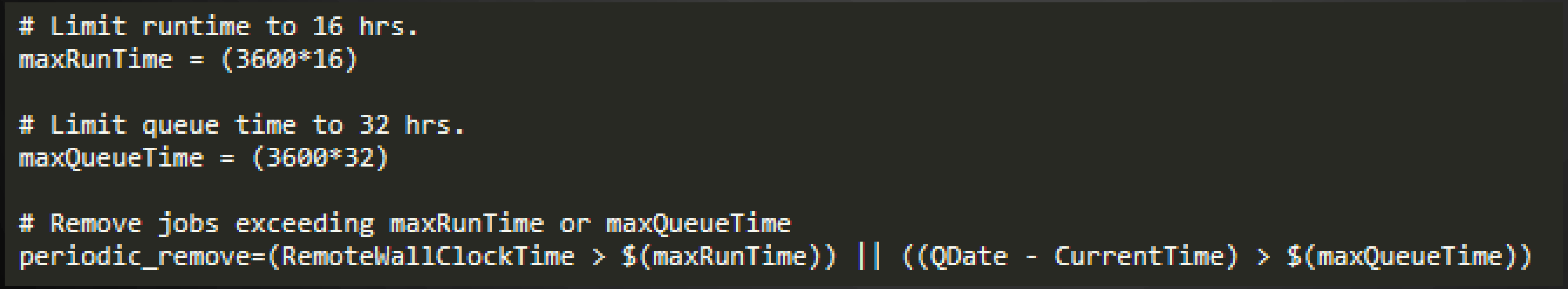
SUBMIT FILE FEATURES

- Specify vars within DAG file, queue 1



```
# Specify program arguments
arguments=v94 $(cvType) $(modelOrder) $(lvl1kInd) $(kIndPlasso) $(iVar) $(cvLevel)
```

- Limit runtime and queue time for stalled jobs or one-off errors

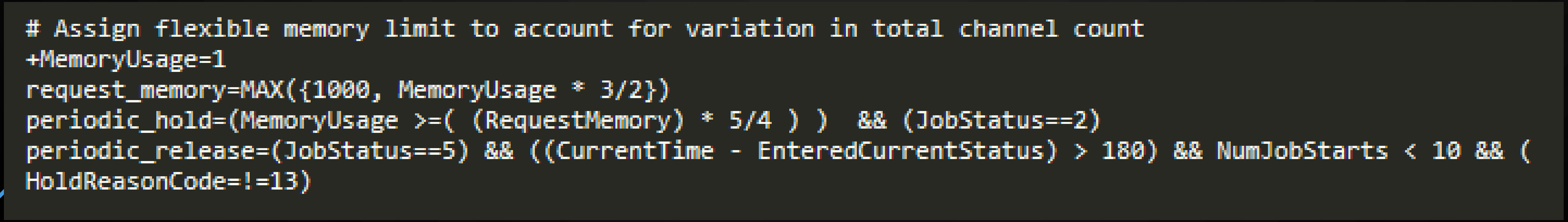


```
# Limit runtime to 16 hrs.
maxRunTime = (3600*16)

# Limit queue time to 32 hrs.
maxQueueTime = (3600*32)

# Remove jobs exceeding maxRunTime or maxQueueTime
periodic_remove=(RemoteWallClockTime > $(maxRunTime)) || ((QDate - CurrentTime) > $(maxQueueTime))
```

- Request dynamic memory limit (at average of job requirements) to account for variation in input size (total channel count) across expt. conditions



```
# Assign flexible memory limit to account for variation in total channel count
+MemoryUsage=1
request_memory=MAX({1000, MemoryUsage * 3/2})
periodic_hold=(MemoryUsage >=( (RequestMemory) * 5/4 ) )  && (JobStatus==2)
periodic_release=(JobStatus==5) && ((CurrentTime - EnteredCurrentStatus) > 180) && NumJobStarts < 10 && (
HoldReasonCode=!=13)
```

DAG SPLICING

- CV Procedure outlined optimizes single model fit to → 1-minute segment/single patient/single experimental condition

- **Total data (currently) that requires hyperparameter optimization**
  - 5 patients * 3-5 recording conditions * 2-10 single minute segments

“A weakness in scalability exists when submitting a DAG within a DAG. Each executing independent DAG requires its own invocation of condor_dagman to be running.”

- Loop over additional experimental variables (patients/conditions/segments) using **SPLICES** rather than **subdags**

- I originally utilized subdags for this (suboptimal), and it took forever ☹. Splices are key in most cases.

- Can run all models in approximately a week or two ☺

CONCLUDING REMARKS

CHTC UTILITY

- Total job count is the primary hurdle for this analysis pipeline. Such computations are not tractable on a single local machine.
- With the help of CHTC, we can understand the computations of the brain by ***efficiently*** modeling how dozens of different cortical regions (hundreds of recording channels) causally influence one another

OTHER

- Will be making this pipeline's code publicly available in ~1 month
- Includes MATLAB code to construct DAGs and submit files for GRID-SEARCH CV
- Feel free to contact me, endemann@wisc.edu, or follow my GitHub activity, https://github.com/qualiaMachine, to be notified when the code is released

Personnel, collaborators, funding

Banks Lab

- Matthew Banks, P.I.
- Declan Campbell
- Sean Grady
- Bryan Krause
- Caitlin Murphy
- Ziyad Sultan

Collaborators

- Kirill Nourski, U Iowa
- Matt Howard, U Iowa
- Robert Sanders, UW SMPH
- Barry Van Veen, UW SoE

Funding

- NIGMS
- Dept. of Anesthesiology

Compute Resources

- UW-Madison's Center For High Throughput Computing (CHTC)