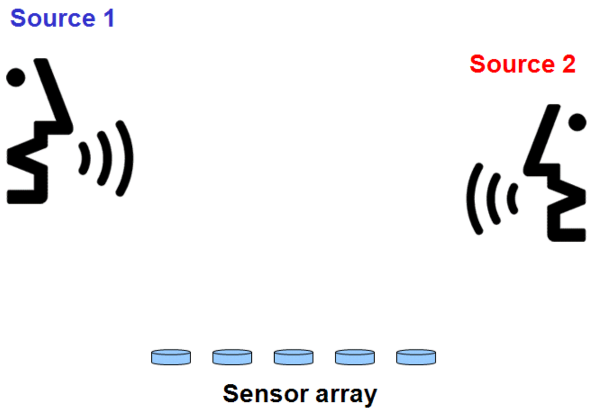
Adapting to Non-Stationarity in EEG using a Mixture of Multiple ICA Models

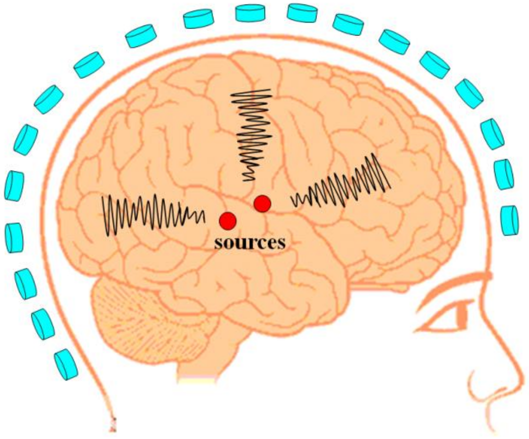
Jason A. Palmer[1] Scott Makeig[1] Julie Onton[1]
Zeynep Akalin-Acar[1] Ken Kreutz-Delgado[2]
Bhaskar D. Rao[2]

[1] Swartz Center for Computational Neuroscience
[2] Dept of Electrical and Computer Engineering
University of California San Diego, La Jolla, CA

Introduction

- Want to model sensor array data with multiple independent sources — ICA





- Non-stationary source activity — mixture model
- Want the adaptation to be computationally efficient — Newton method

Outline

- Introduction
  - Non-stationarity in EEG
  - What is a mixture model?
- ICA Mixture Model
  - Model definition
  - Computational feasibility and Newton Method
- Examples
  - Application to epileptic seizure ECoG data
  - Application to typical EEG task data
- Implementation
  - Parallel computation

Non-stationarity

- What kinds of non-stationarity exist in EEG?
  - Environmental transients—lights, train, A/C
  - Different brain sources for different tasks
  - Muscle activity
  - Arousal level change
  - Seizure
- Are EEG components stable over recording? Which are and which are not?
- We approach this problem by using a mixture model of component bases with separate component maps and source statistics

What is a Mixture Model?

- A mixture model is a probabilistic combination of several models:

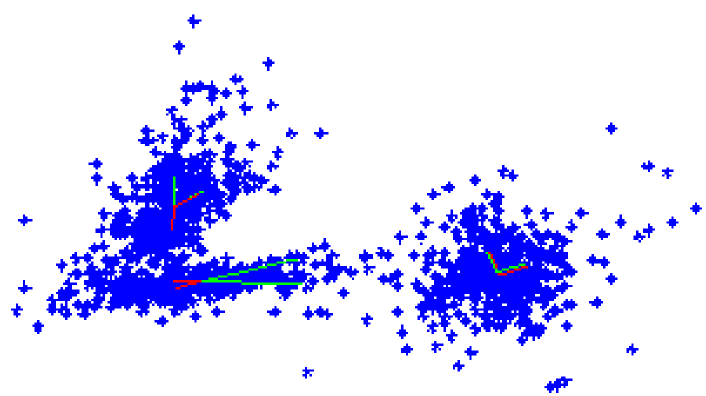
mixture proportions

means

$$p(x) = \sum_{j=1}^{M} \gamma_j \, p_j\left(\frac{x - \mu_j}{\sigma_j}\right)$$

scales

- Each data point modeled as being generated by one of the models in the mixture

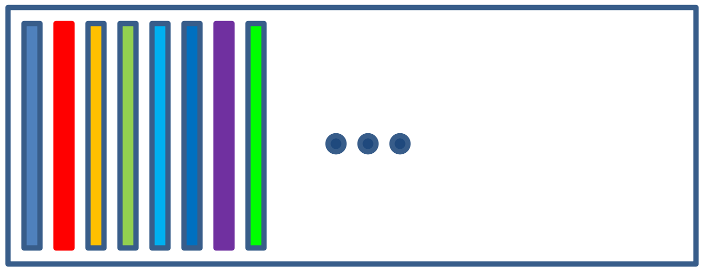


UCSD

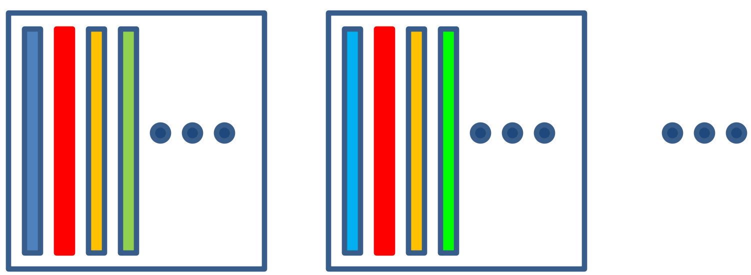
Swartz Center for Computational Neuroscience

Mixture vs. Overcomplete

- Approach 1 – Overcomplete dictionary



- Approach 2 – Mixture of bases (like best basis selection)



- Assumptions:
  - At a given time at most num channels basis vectors present
  - Basis vectors do not combine arbitrarily but form subsets or groups of commonly occurring or mutually exclusive features

Computational Feasibility

- We will use an iterative algorithm, in which the basic steps are:
  - Estimate the independent source activations given models
  - Update models given estimated sources

- For large dimensional problems estimation of sources by iterative or even one-step methods takes non-trivial time, requiring inversion of a matrix for each sample
  - Example: data = 100 x 1,000,000, time to get sources = 1 ms per sample, one complete iteration takes at least 1000 seconds = 15 minutes, 500 iterations takes 6 days
  - Need iterations to be order seconds, so need source estimation to be very fast (less than 1ms) – use simple matrix multiplication, can't afford inversion

ICA Mixture Model

- Want to model observations $\boldsymbol{x}(t)$, $t = 1,\ldots,N$, different models "active" at different times
- Bayesian linear mixture model, $h = 1, \ldots, M$:

$$\mathbf{x}(t) = \mathbf{A}_h \mathbf{s}(t) + \mathbf{c}_h$$

- Conditionally linear given the model, $\mathbf{W}_h \triangleq \mathbf{A}_h^{-1}$:

$$p(\mathbf{x}(t) \,|\, h) = |\det \mathbf{W}_h| \, q_h\big(\mathbf{W}_h(\mathbf{x}(t) - \mathbf{c}_h)\big)$$

- Samples are modeled as independent in time:

$$p(\mathbf{X}; \Theta) = \prod_{t=1}^{N} \sum_{h=1}^{M} \gamma_h \, p(\mathbf{x}(t) \,|\, h)$$

UCSD

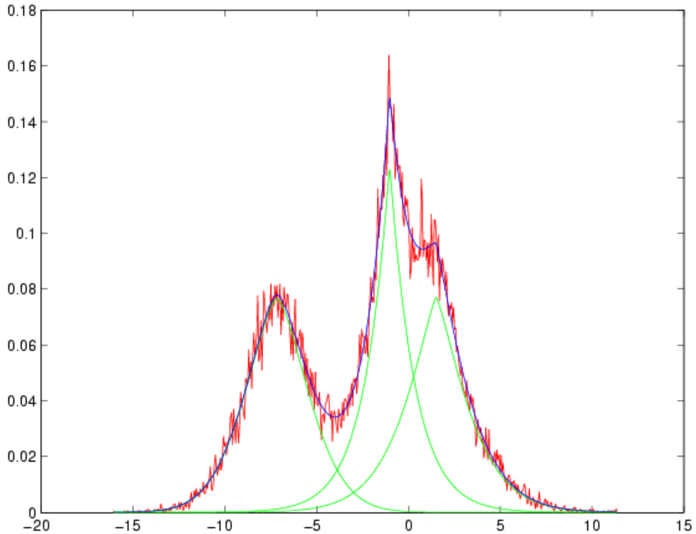
Swartz Center for Computational Neuroscience

Source Density Mixture Model

- Each source density mixture component has unknown location, scale, and shape:

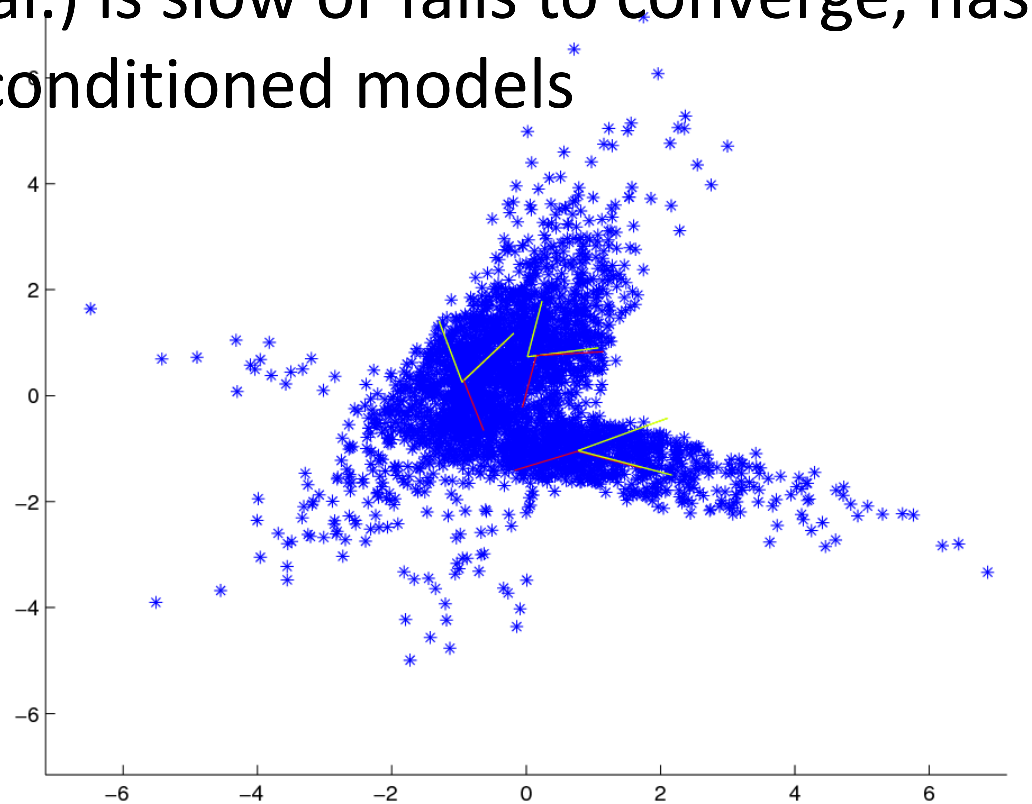
$$q_{hi}\big(s_i(t)\big) = \sum_{j=1}^{m} \alpha_{hij}\sqrt{\beta_{hij}}\, q_{hij}\Big(\sqrt{\beta_{hij}}\big(s_i(t)-\mu_{hij}\big)\,;\rho_{hij}\Big)$$

- Generalized Gaussian mixture model is convenient and flexible



Sub- and Super-Gaussian sources

- With mixture source model, model sources can be sub- or super-Gaussian, no need to check
- Newton method converges very fast, natural gradient (Lee et al.) is slow or fails to converge, has difficulty on poorly conditioned models



ICA Mixture Model—Invariances

- The complete set of parameters to be estimated is:

$$\Theta = \{\mathbf{W}_h, \mathbf{c}_h, \gamma_h, \alpha_{hij}, \mu_{hij}, \beta_{hij}, \rho_{hij}\}$$

$h = 1, \ldots, M, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m$

- Invariances: **W** row norm/source density scale and model centers/source density locations:

$$[\mathbf{W}'_h]_{i:} = [\mathbf{W}_h]_{i:}/\tau_{hi},$$
$$\mu'_{hij} = \mu_{hij}/\tau_{hi}, \quad \beta'_{hij} = \beta_{hij}\tau_{hi}^2, \quad j = 1, \ldots, m$$

$$\mathbf{c}'_h = \mathbf{c}_h + \Delta\mathbf{c}_h, \quad \mu'_{hij} = \mu_{hij} - [\mathbf{W}_h \Delta\mathbf{c}_h]_i, \quad j = 1, \ldots, m$$

Basic ICA Newton Method

- Transform gradient (1st derivative) of cost function using inverse Hessian (2nd derivative)
- Cost function is data log likelihood:

$$p(\mathbf{X}) = \prod_{t=1}^{N} |\det \mathbf{W}|\, p_{\mathbf{s}}(\mathbf{W}\mathbf{x}_t)$$

$$L(\mathbf{W}) = \sum_{t=1}^{N} -\log|\det \mathbf{W}| + f(\mathbf{y}_t)$$

- Gradient:

$$\nabla L(\mathbf{W}) \propto -\mathbf{W}^{-T} + \frac{1}{N}\sum_{t=1}^{N} \nabla f(\mathbf{y}_t)\mathbf{x}_t^T$$

- Natural gradient (positive definite transform):

$$\Delta \mathbf{W} = \left(\mathbf{I} - \frac{1}{N}\sum_{t=1}^{N} \mathbf{g}_t \mathbf{y}_t^T\right)\mathbf{W}$$

UCSD

Swartz Center for Computational Neuroscience

Newton Method – Hessian

- Take derivative of (*i,j*)th element of gradient with respect to (*k,l*)th element of $\mathbf{W}$:

$$\frac{\partial g_{ij}}{\partial w_{kl}} = [\mathbf{W}^{-1}]_{li}[\mathbf{W}^{-1}]_{jk} + \Big\langle f_i''\big([\mathbf{W}\mathbf{x}_t]_k\big)x_j x_l \delta_{ik}\Big\rangle_N$$

- This defines a linear transform $\mathbf{C} = \mathcal{H}(\mathbf{B})$:

$$c_{ij} = \sum_k \sum_l [\mathbf{W}^{-1}]_{li}[\mathbf{W}^{-1}]_{jk} b_{kl} + \Big\langle f_i''(y_i)\, x_j \sum_l b_{il} x_l \Big\rangle_N$$

- In matrix form, this is:

$$\mathbf{C} = \mathbf{W}^{-T}\mathbf{B}^T\mathbf{W}^{-T} + \frac{1}{N}\sum_{t=1}^{N} \mathrm{diag}(f''(\mathbf{y}_t))\mathbf{B}\mathbf{x}_t\mathbf{x}_t^T$$

UCSD

Swartz Center for Computational Neuroscience

Newton Method – Hessian

- To invert: rewrite the Hessian transformation $\mathbf{C} = \mathcal{H}(\mathbf{B})$ in terms of the source estimates:

$$\mathbf{C} = (\mathbf{B}\mathbf{W}^{-1})^T\mathbf{W}^{-T} + \left\langle \mathrm{diag}\big(f''(\mathbf{y})\big)\mathbf{B}\mathbf{W}^{-1}\mathbf{W}\mathbf{x}\mathbf{y}^T\mathbf{W}^{-T} \right\rangle_N$$

- Define $\tilde{\mathbf{C}} \triangleq \mathbf{C}\mathbf{W}^T$, $\tilde{\mathbf{B}} \triangleq \mathbf{B}\mathbf{W}^{-1}$, $\tilde{\mathbf{C}} = \tilde{\mathcal{H}}(\tilde{\mathbf{B}})$:

$$\tilde{\mathbf{C}} = \tilde{\mathbf{B}}^T + \left\langle \mathrm{diag}\big(f''(\mathbf{y})\big)\tilde{\mathbf{B}}\mathbf{y}\mathbf{y}^T \right\rangle_N$$

- Want to solve linear equation $\mathbf{C} = \mathcal{H}(\mathbf{B})$:

$$\mathbf{B} = \mathcal{H}^{-1}(\mathbf{C}) = \tilde{\mathcal{H}}^{-1}(\mathbf{C}\mathbf{W}^T)\mathbf{W}$$

Newton Method – Hessian

- The Hessian transformation can be simplified using source independence and zero mean:

$$\tilde{c}_{ii} \;\rightarrow\; \tilde{b}_{ii} + E\Big\{ f_i''(y_i) \sum_k \tilde{b}_{ik} y_k y_i \Big\} = \tilde{b}_{ii}(1+\eta_i)$$

$$\tilde{c}_{ij} \;\rightarrow\; \tilde{b}_{ji} + E\Big\{ f_i''(y_i) \sum_k \tilde{b}_{ik} y_k y_j \Big\} = \tilde{b}_{ji} + \kappa_i \sigma_j^2 \tilde{b}_{ij}$$

$$\tilde{c}_{ji} \;\rightarrow\; \tilde{b}_{ij} + E\Big\{ f_j''(y_j) \sum_k \tilde{b}_{jk} y_k y_i \Big\} = \tilde{b}_{ij} + \kappa_j \sigma_i^2 \tilde{b}_{ji}$$

$$\eta_i \triangleq E\{y_i^2 f_i''(y_i)\}, \quad \kappa_i \triangleq E\{f_i''(y_i)\}, \quad \sigma_i^2 \triangleq E\{y_i^2\}$$

- This leads to 2x2 block diagonal form:

$$\begin{bmatrix} \tilde{c}_{ij} \\ \tilde{c}_{ji} \end{bmatrix} = \begin{bmatrix} \kappa_i \sigma_j^2 & 1 \\ 1 & \kappa_j \sigma_i^2 \end{bmatrix} \begin{bmatrix} \tilde{b}_{ij} \\ \tilde{b}_{ji} \end{bmatrix}$$

Newton Direction

- Invert Hessian transformation, evaluate at gradient:

$$\Delta \mathbf{W} = \tilde{\mathcal{H}}^{-1}(-\mathbf{G}\mathbf{W}^T)\mathbf{W}$$

- Leads to the following equations:

$$\tilde{\mathbf{B}} = \tilde{\mathcal{H}}^{-1}(-\mathbf{G}\mathbf{W}^T)$$

$$\mathbf{\Phi} \triangleq \frac{1}{N}\sum_{t=1}^{N} \mathbf{g}_t \mathbf{y}_t^T$$

$$-\mathbf{G}\mathbf{W}^T = \mathbf{I} - \mathbf{\Phi}$$

$$\begin{aligned} \tilde{b}_{ii} &= \frac{1-\phi_{ii}}{1+\eta_i}, \quad i = 1, \ldots, n \\ \tilde{b}_{ij} &= \frac{\phi_{ji} - \kappa_j \sigma_i^2 \phi_{ij}}{\kappa_i \kappa_j \sigma_i^2 \sigma_j^2 - 1}, \quad \forall i \neq j \end{aligned}$$

- Calculate the Newton direction:

$$\Delta \mathbf{W} = \tilde{\mathbf{B}}\mathbf{W}$$

UCSD

Swartz Center for Computational Neuroscience

Positive Definiteness of Hessian

- Conditions for positive definiteness:

$$
\begin{aligned}
&1)\quad 1+\eta_i > 0, \quad \forall i \\
&2)\quad \kappa_i > 0, \quad \forall i, \text{ and,} \\
&3)\quad \kappa_i \kappa_j \sigma_i^2 \sigma_j^2 - 1 > 0, \quad \forall i \neq j
\end{aligned}
$$

- Always true for true when model source densities match true densities:

1)

$$
\begin{aligned}
1 + E\{y^2 f''(y)\} &= \int_{-\infty}^{\infty} \left(y^2 f'(y)^2 - 2yf'(y) + 1\right) p(y)\,dy \\
&= E\left\{\left(yf'(y) - 1\right)^2\right\} \geq 0
\end{aligned}
$$

2)

$$
E\{f''(y)\} = \int_{-\infty}^{\infty} f'(y)^2 p(y)\,dy = E\left\{f'(y)^2\right\} > 0
$$

3)

$$
E\{y^2\}E\{f''(y)\} = E\left\{y^2\right\}E\left\{f'(y)^2\right\} \geq \left(E\{yf'(y)\}\right)^2 = 1
$$

UCSD

Swartz Center for Computational Neuroscience

Newton for ICA Mixture Model

- Similar derivation applies to ICA mixture model:

$$p(\mathbf{X};\Theta) = \sum_{\mathbf{V},\mathbf{Z}} \prod_{t=1}^{N} \prod_{h=1}^{M} \gamma_h^{v_{ht}} |\det \mathbf{W}_h|^{v_{ht}} \prod_{i=1}^{n} \prod_{j=1}^{m} {Q_{hijt}^{l}}^{v_{ht} z_{hijt}}$$

$$F^l(\Theta) = \sum_{t=1}^{N} \sum_{h=1}^{M} \Big[\hat{v}_{ht}^l \big(-\log \gamma_h - \log|\det \mathbf{W}_h| \big) + \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{r}_{hijt}^l \big(-\log \alpha_{hij} - \tfrac{1}{2}\log \beta_{hij} + f_{hij}(y_{hijt}) \big) \Big]$$

$$\mathbf{C} = \mathbf{W}_h^{-T} \mathbf{B}^T \mathbf{W}_h^{-T} + \frac{1}{\sum_t \hat{v}_{ht}^l} \sum_{t=1}^{N} \mathbf{D}_{ht}^l \mathbf{B} (\mathbf{x}_t - \mathbf{c}_h)(\mathbf{x}_t - \mathbf{c}_h)^T$$

$$\tilde{\mathbf{C}} = \tilde{\mathbf{B}}^T + \frac{1}{\sum_t \hat{v}_{ht}^l} \sum_{t=1}^{N} \mathbf{D}_{ht}^l \tilde{\mathbf{B}} \mathbf{y}_{ht} \mathbf{y}_{ht}^T$$

$$\mathbf{y}_{ht} \triangleq \mathbf{W}_h (\mathbf{x}_t - \mathbf{c}_h)$$

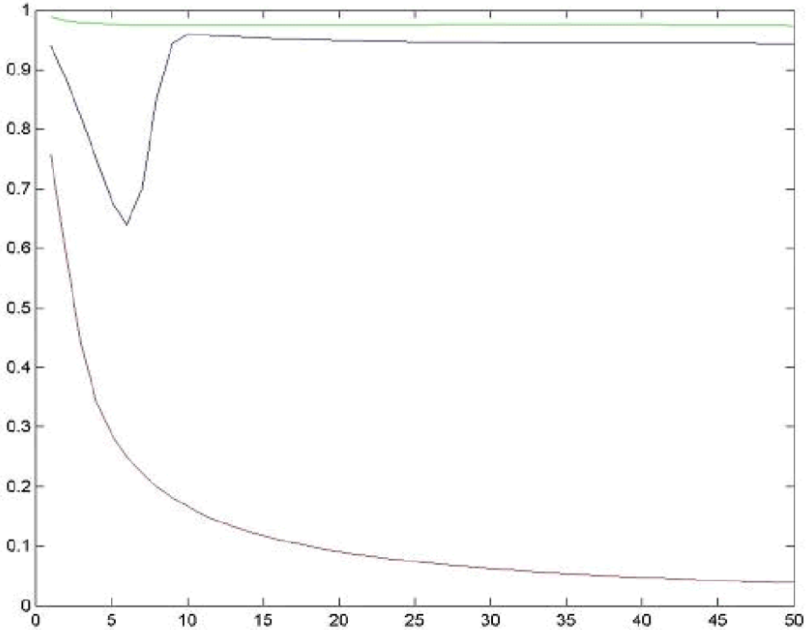
UCSD

Swartz Center for Computational Neuroscience

Convergence Rates

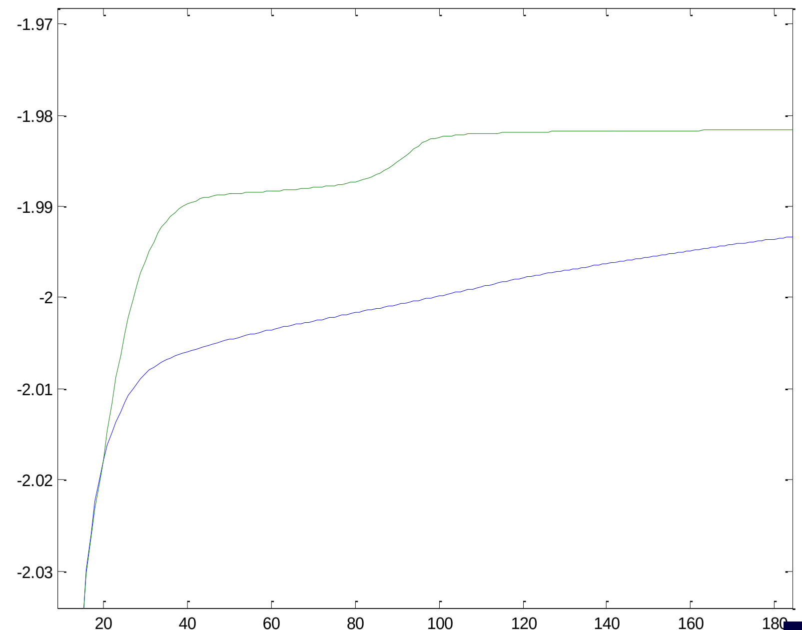
- Convergence is really much faster than natural gradient. Works with step size 1.0!
- Need correct source density model

$$\|\mathbf{W}^{l+1} - \mathbf{W}^*\| \,/\, \|\mathbf{W}^l - \mathbf{W}^*\|$$



iteration

log likelihood



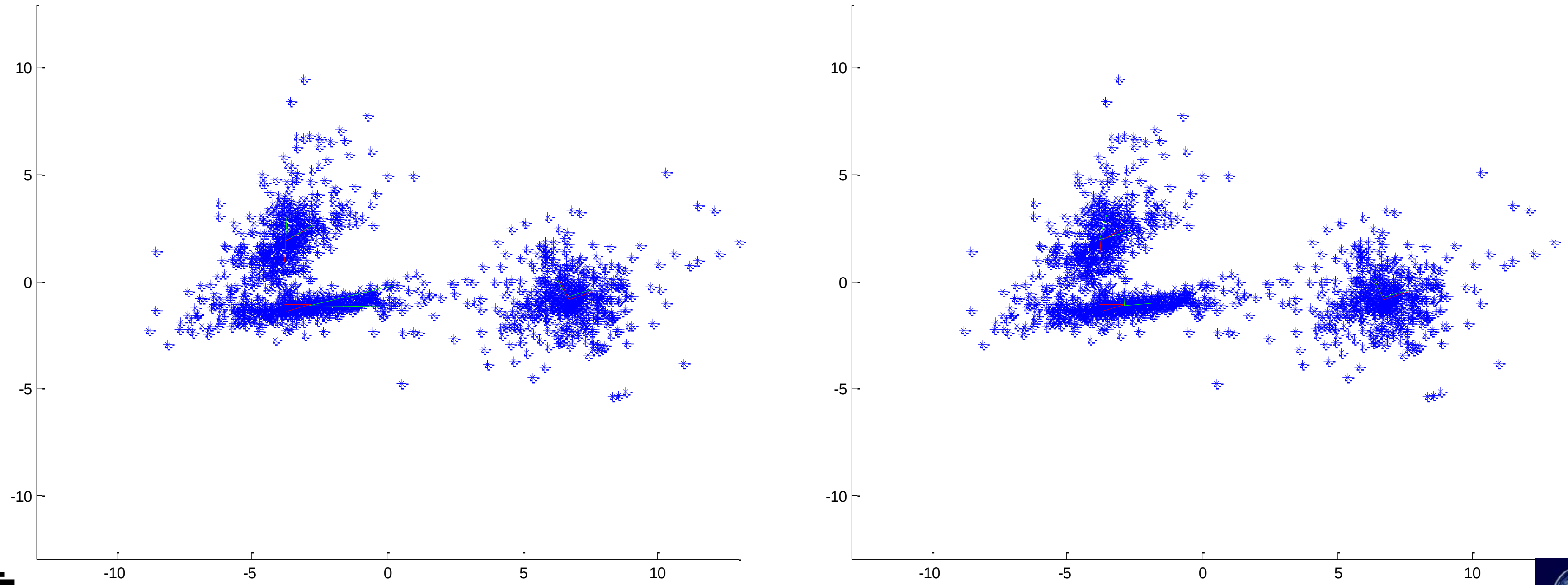
iteration

UCSD

Swartz Center for Computational Neuroscience

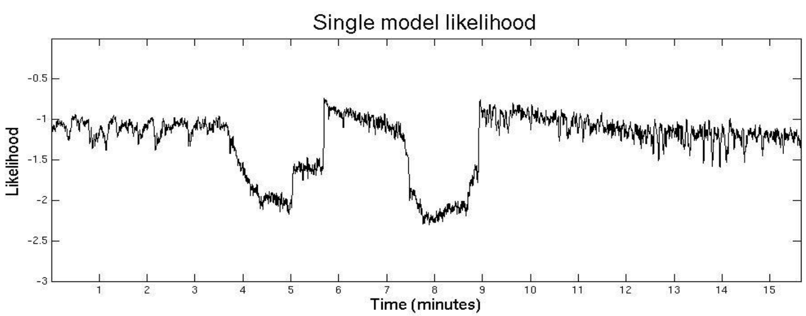
Natural Gradient Vs. Newton

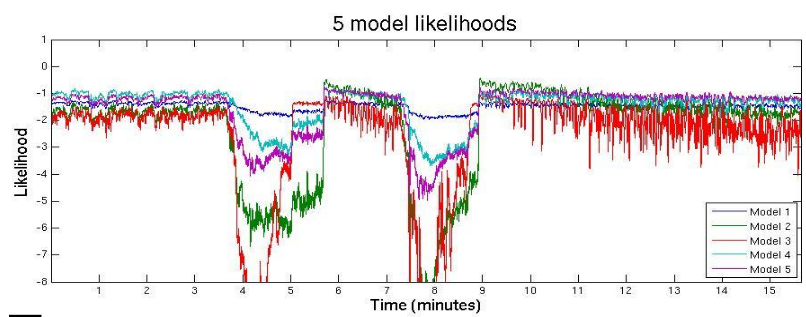
- 3 models in two dimensions, 500 pts per model
- Newton method converges, natural gradient (Lee et al.) is slow or fails to converge, has difficulty on poorly conditioned models

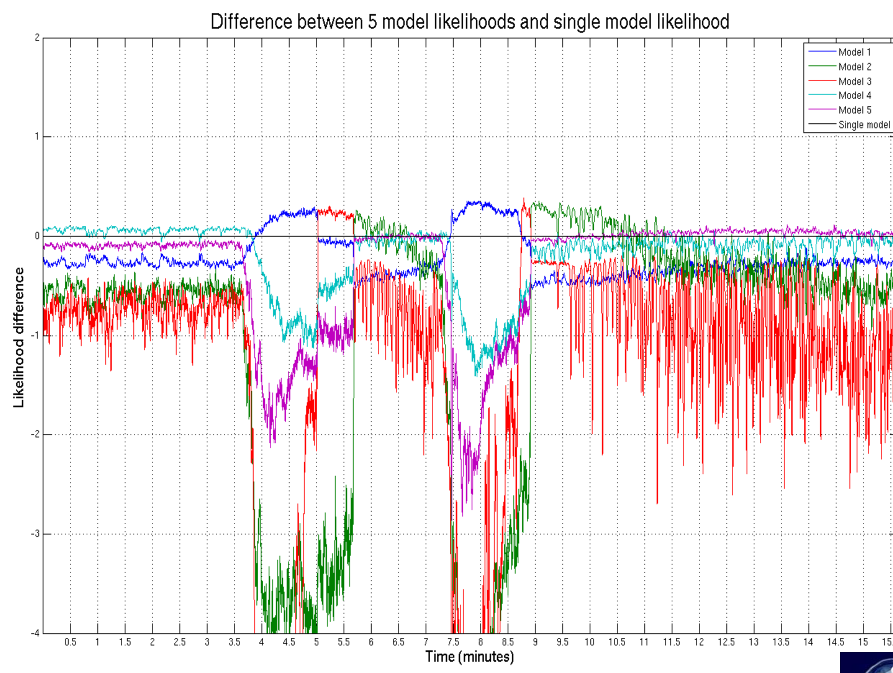


Epilepsy

- Data: 15 minutes from 1 subject containing 2 seizures
- Single model does not represent seizure well
- We learned 5 models – new models consistently adapt to portions of seizure

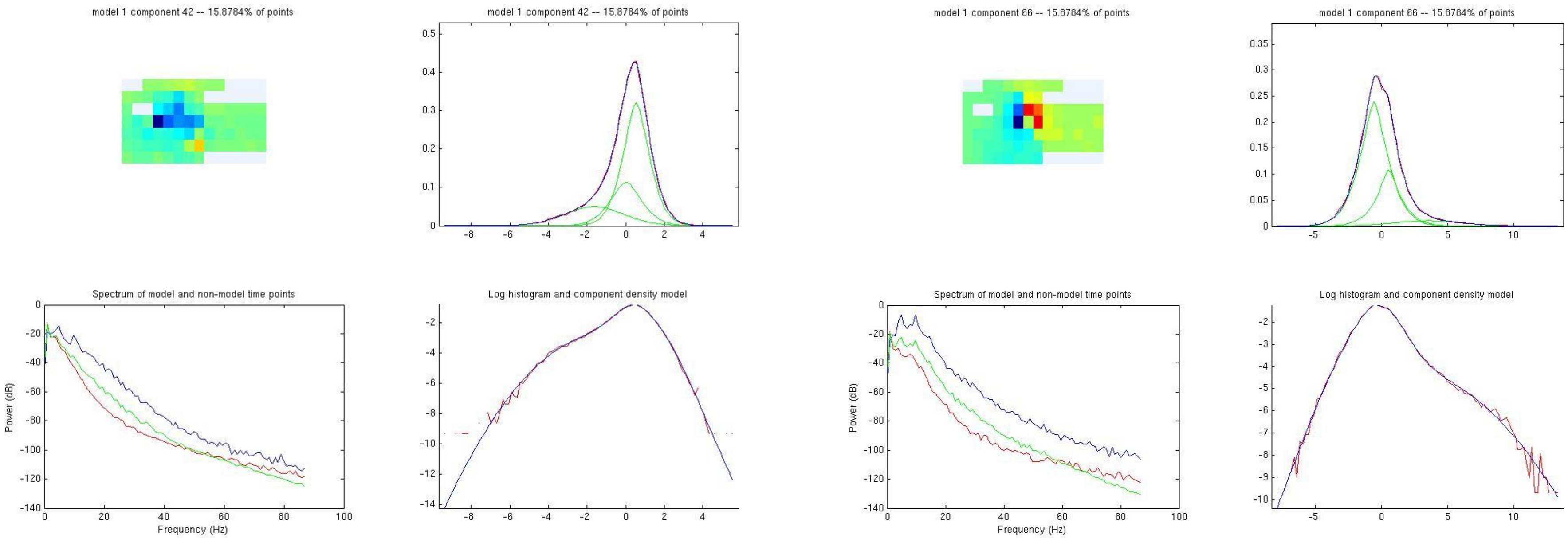




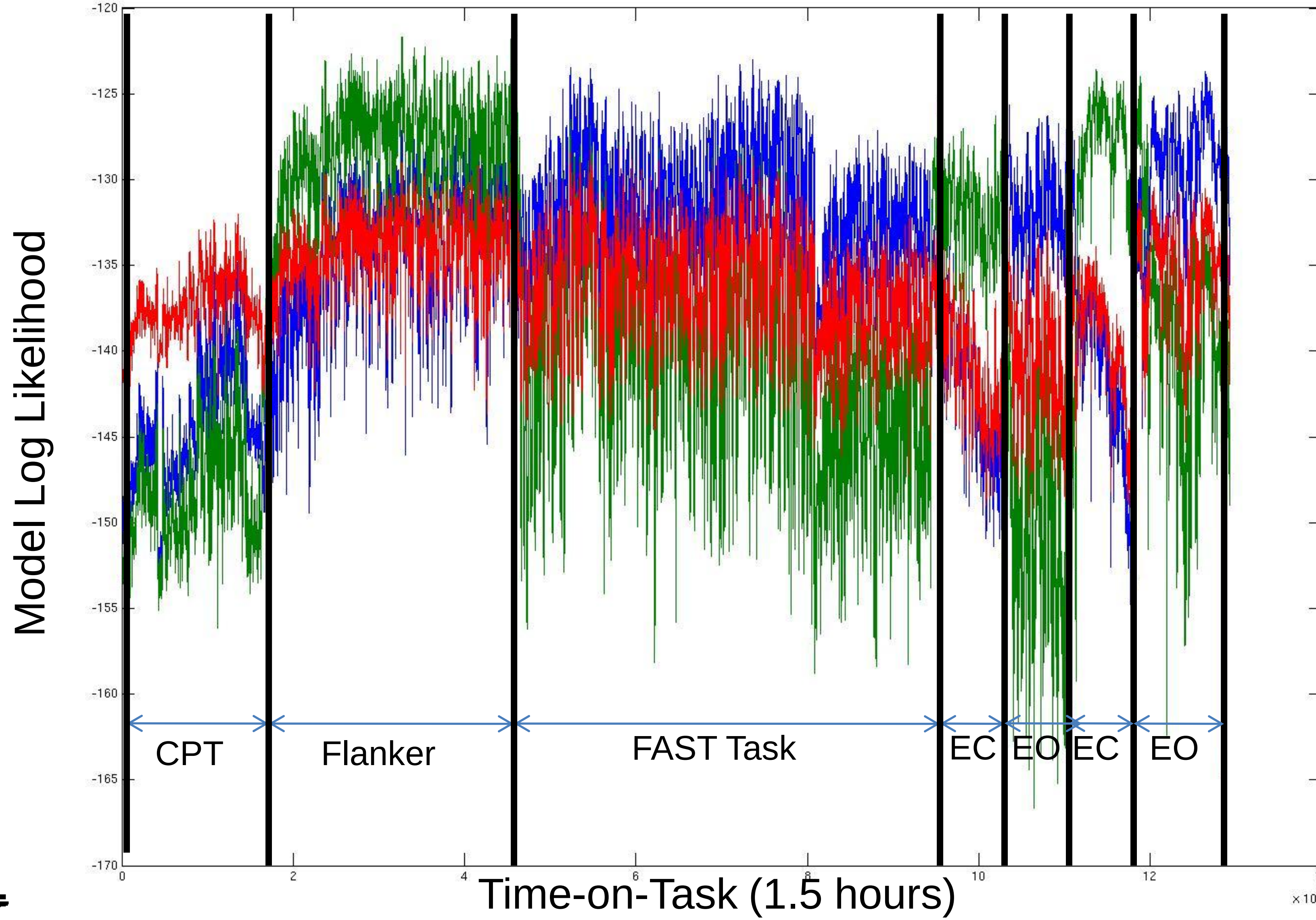


Epilepsy Grid Maps

- Maps from grid of electrodes placed intercranially over seizure area
- Source probability densities are fit by mixture model



Segmentation of Tasks

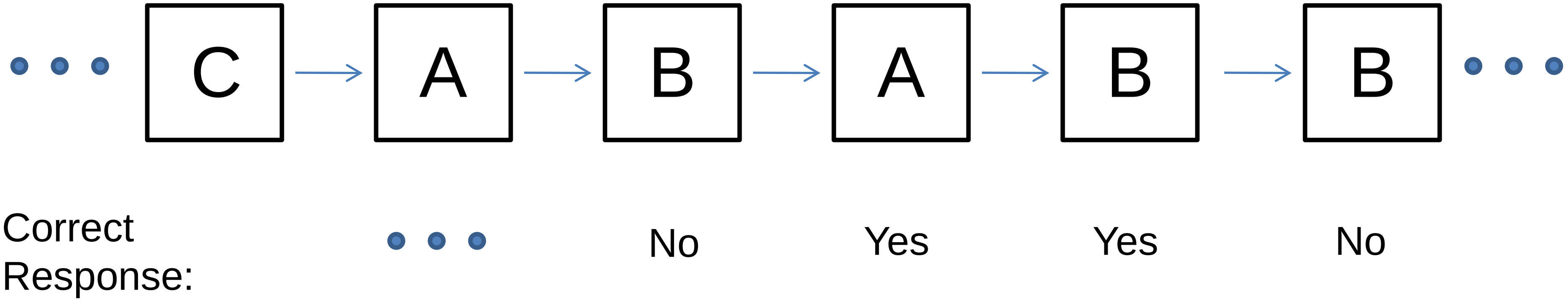


UCSD

Swartz Center for Computational Neuroscience

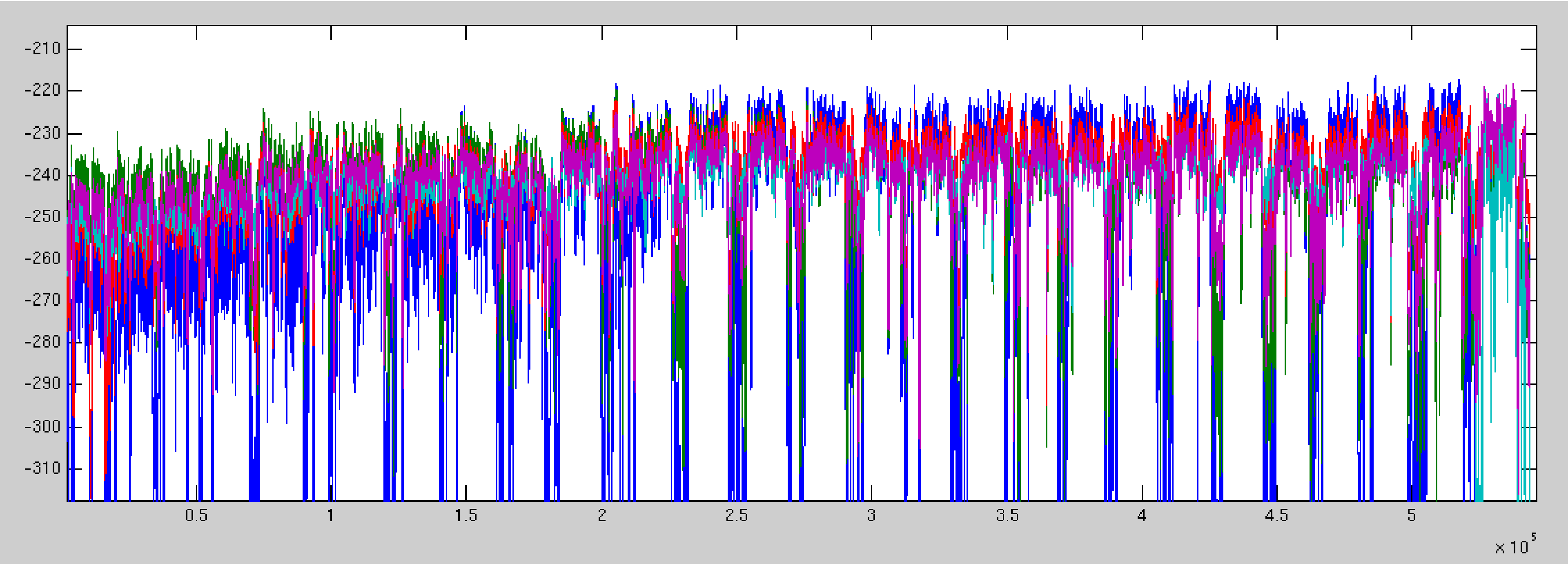
Twoback Task

- Data recording supervised by Julie Onton
- Subject presented with sequence of letters and must respond whether current letter is the same as the one two letters back



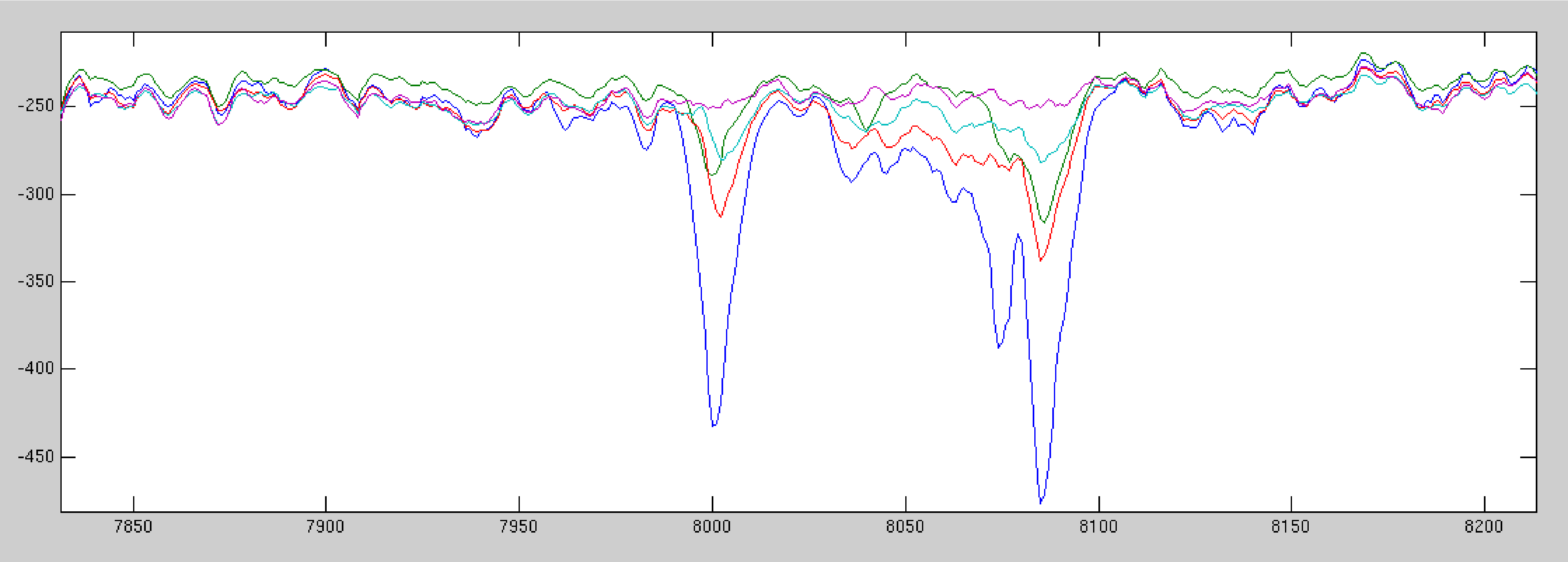
bt73 segmentation

- Task trials are represented by green and blue models
- Inter-task intervals represented by red and cyan model
- Eye blinks represented by magenta model



bt73 segmentation zoom (green)

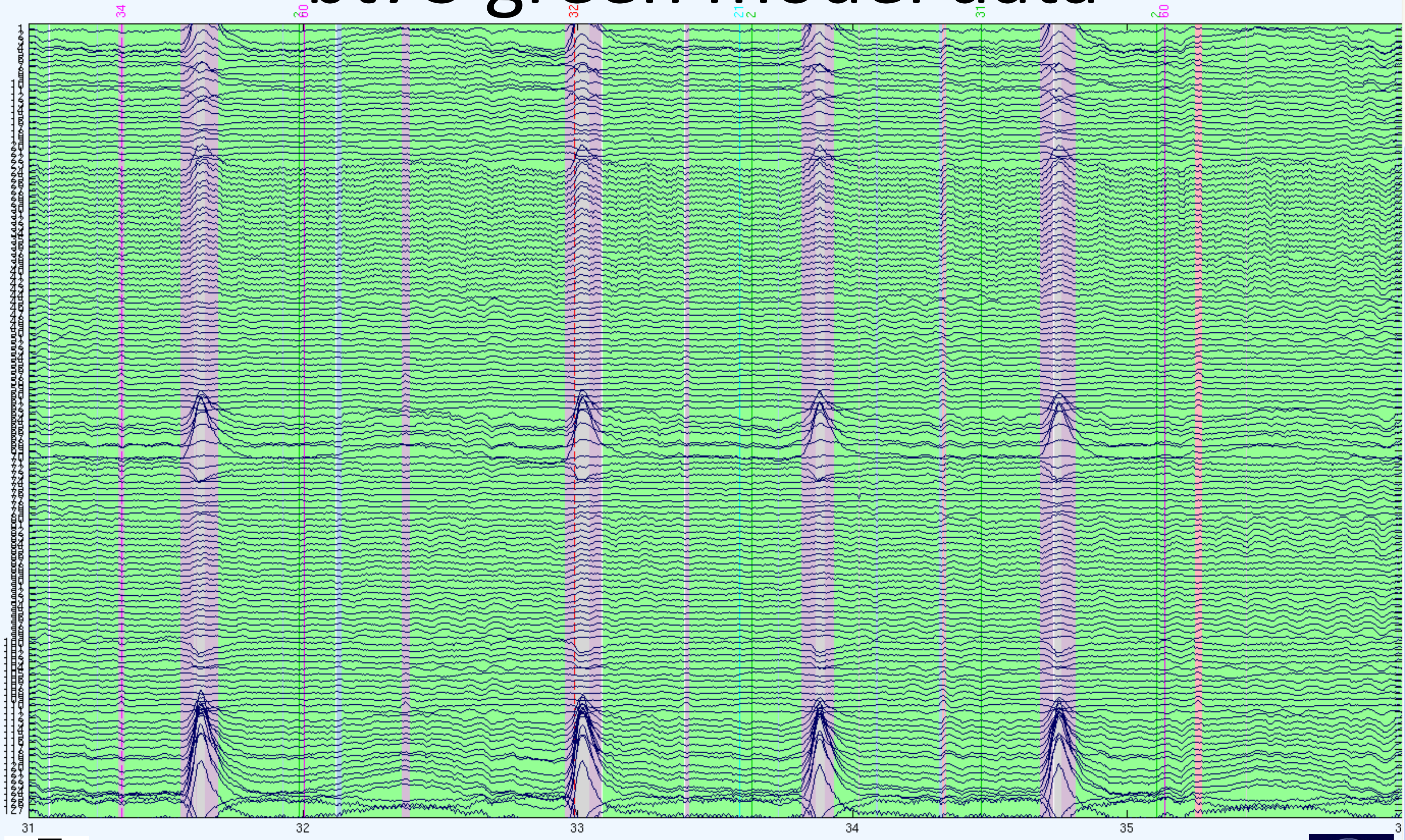
- Task trials are represented by green and blue models
- Inter-task intervals represented by red and cyan model
- Eye blinks represented by magenta model



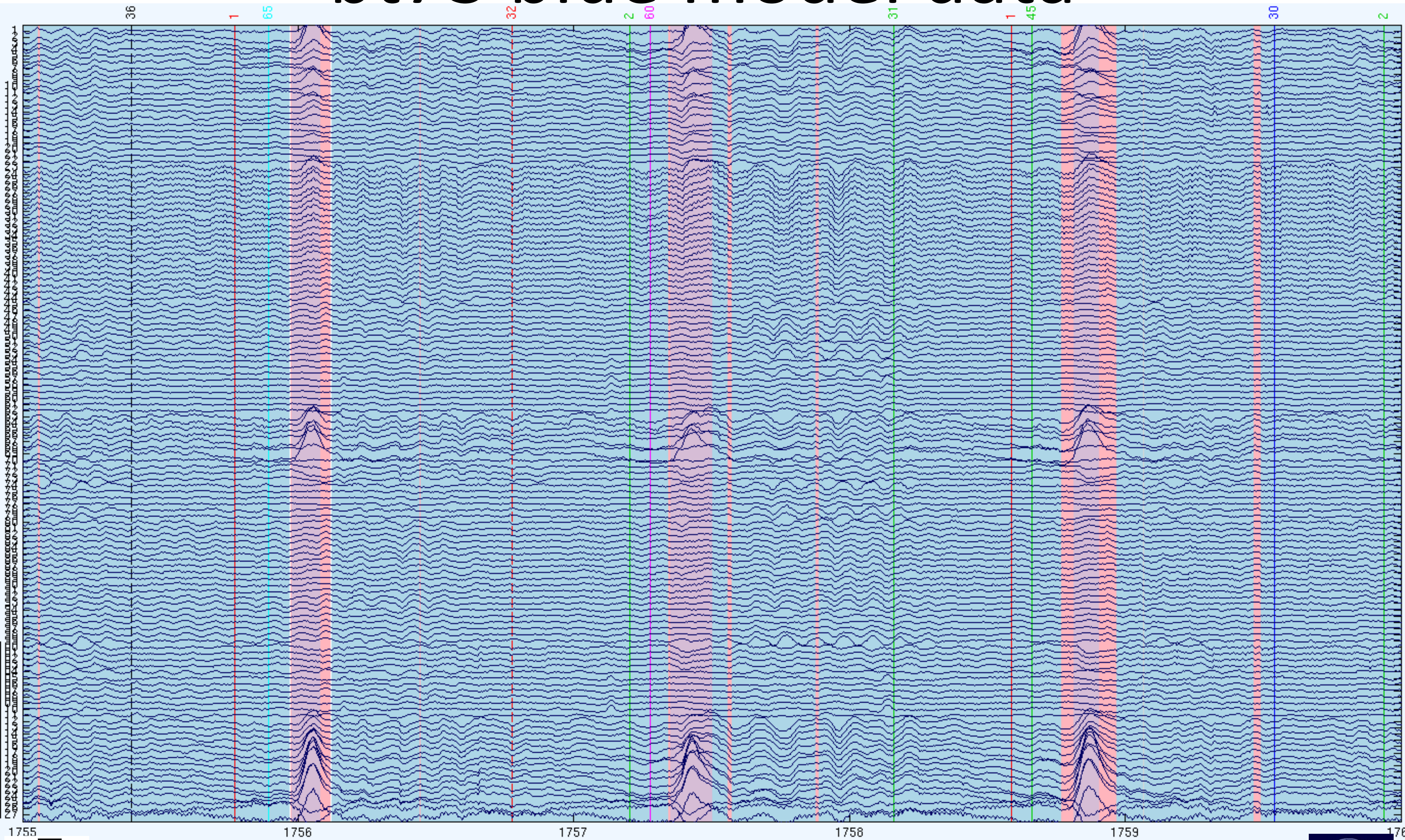
UCSD

Swartz Center for Computational Neuroscience

bt73 green model data



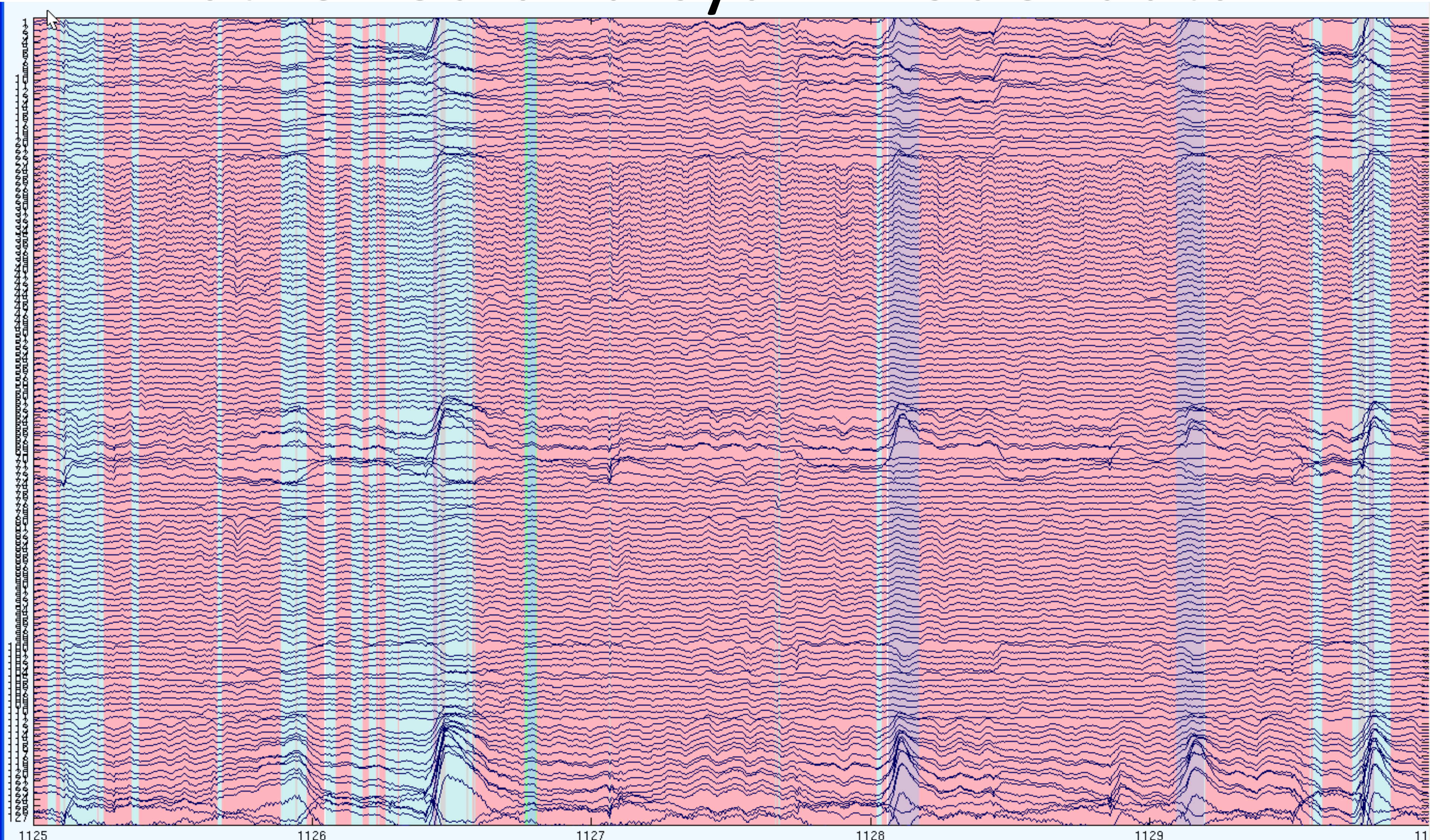
bt73 blue model data



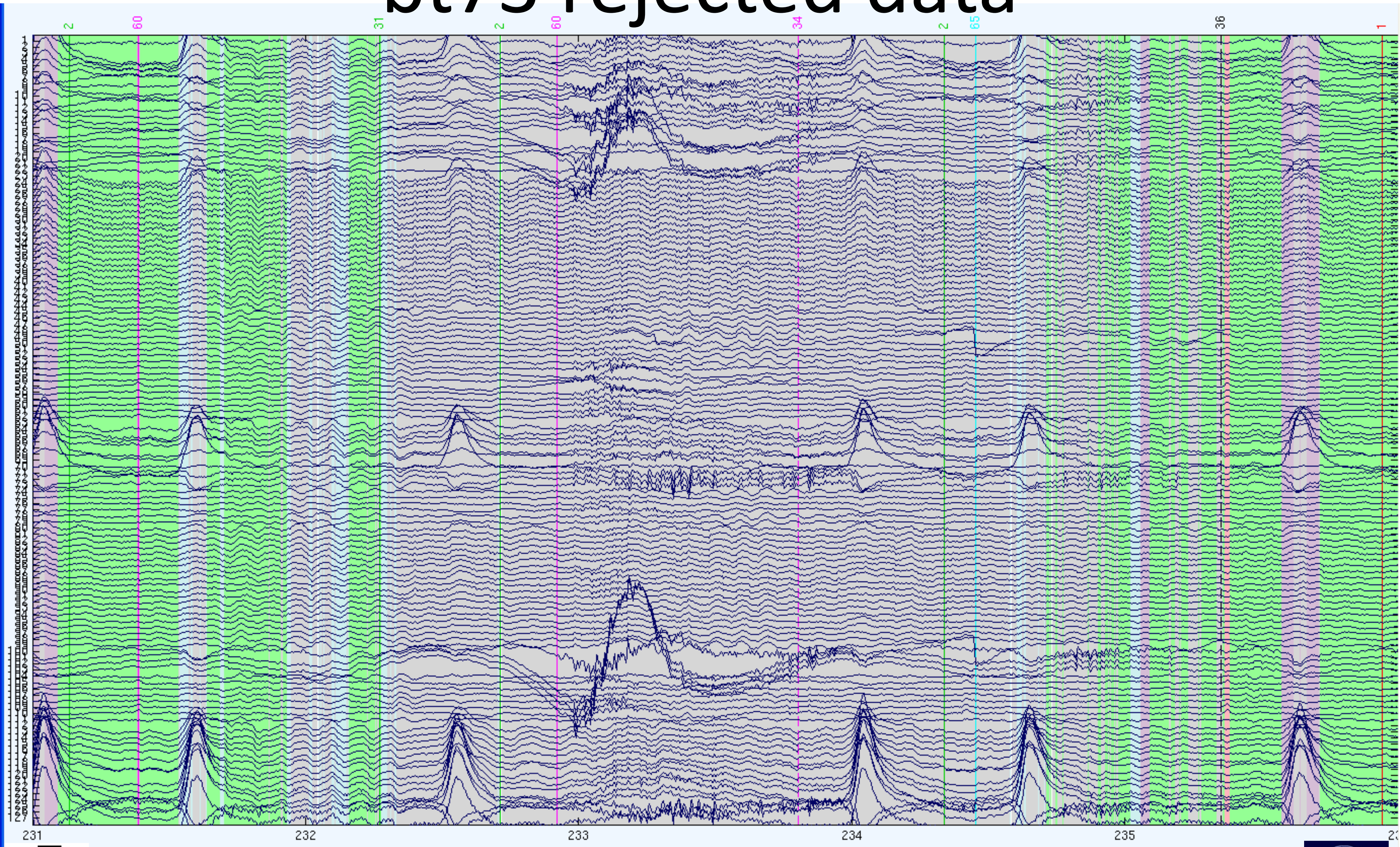
UCSD

Swartz Center for Computational Neuroscience

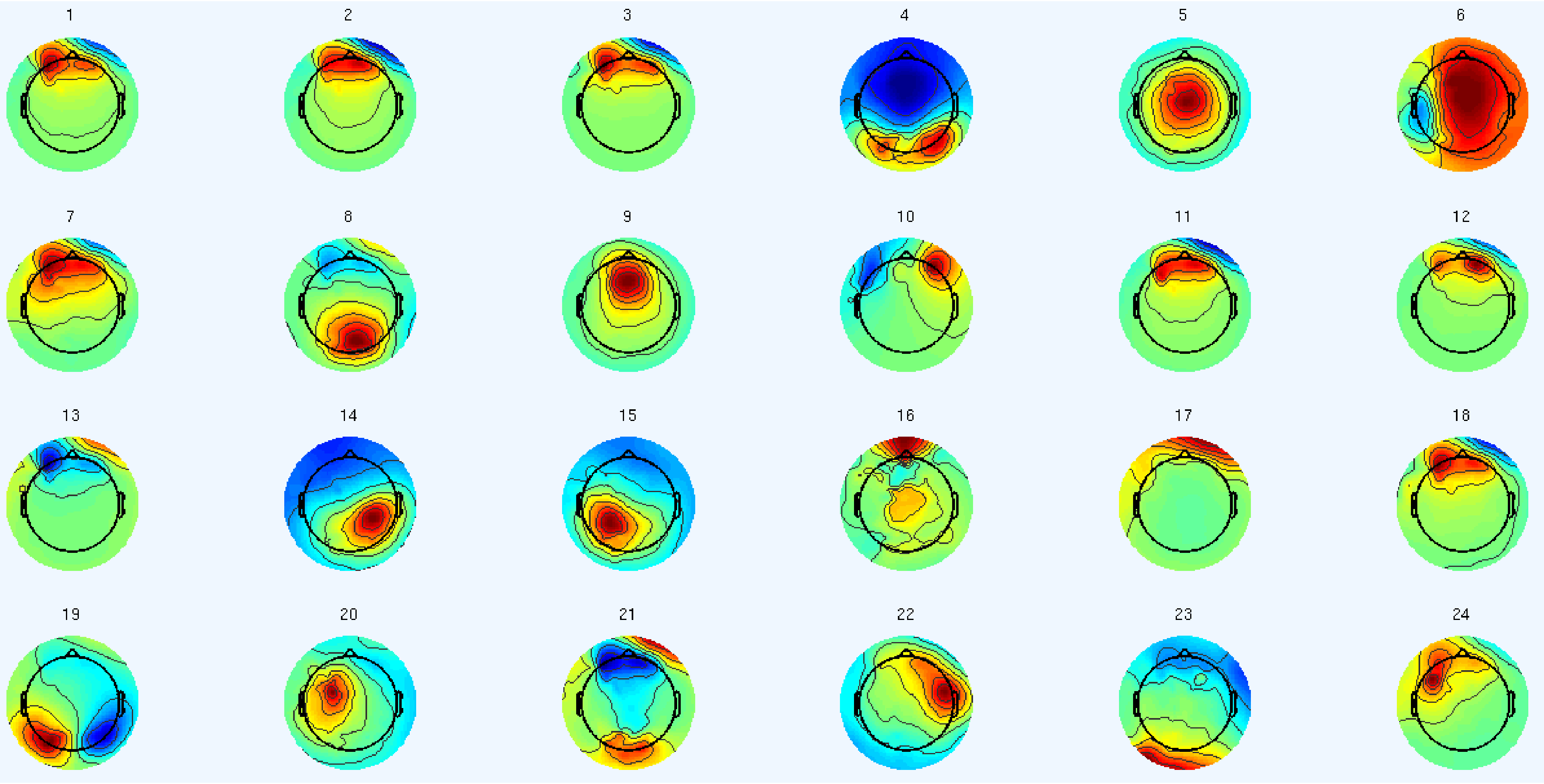
bt73 red and cyan model data



bt73 rejected data

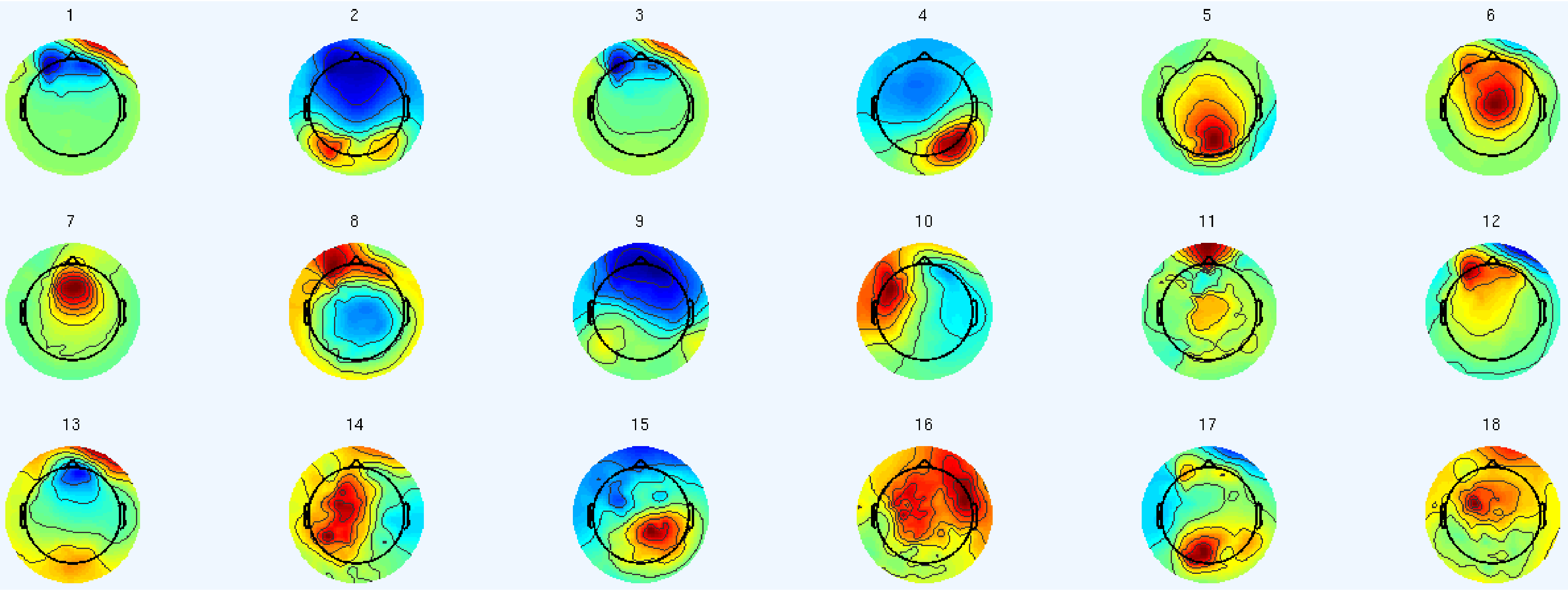


bt73 single model components



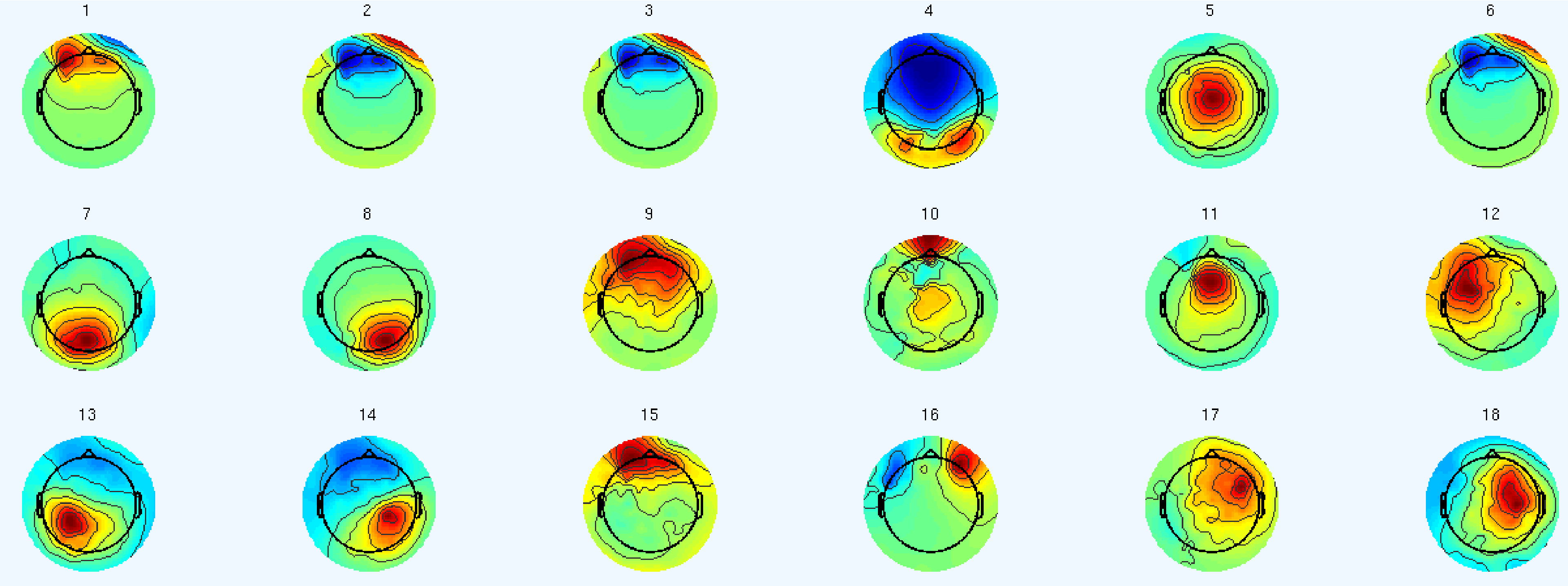
bt73 green model components

- Prominent alpha and frontal midline components
- Weak mu components



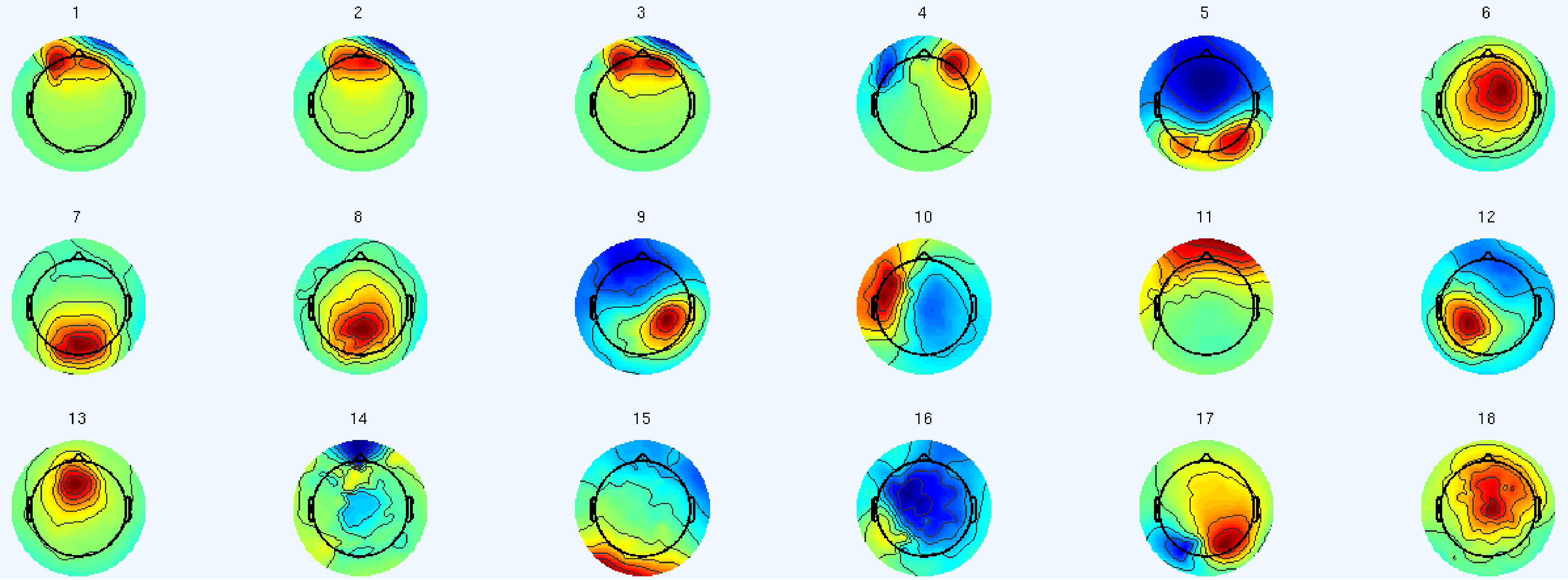
bt73 blue model components

- Prominent alpha and central midline components
- Weak mu components
- Different occipital alpha components (7, 8)



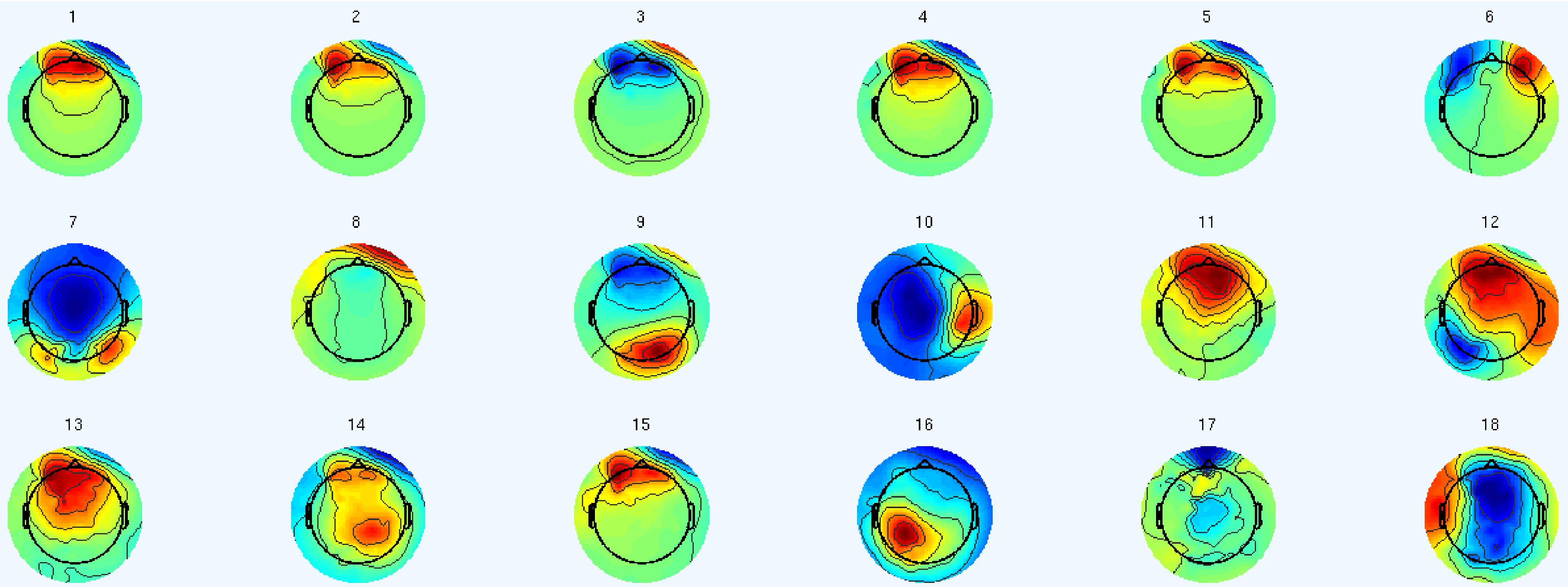
bt73 red model components

- Prominent alpha and central midline components
- Lateral eye movement component (4)
- Tangential occipital component (17)



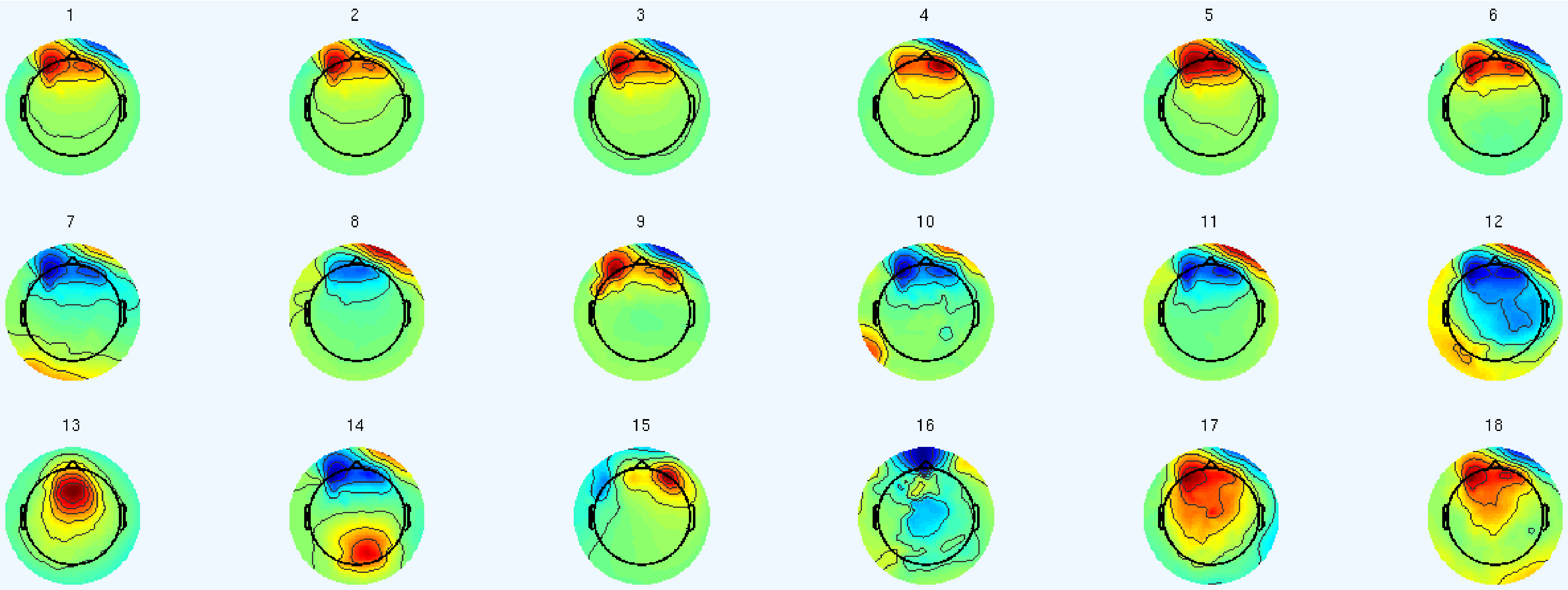
bt73 cyan model components

- Prominent eye-blink components (1-5)
- Lateral eye-movement (6)



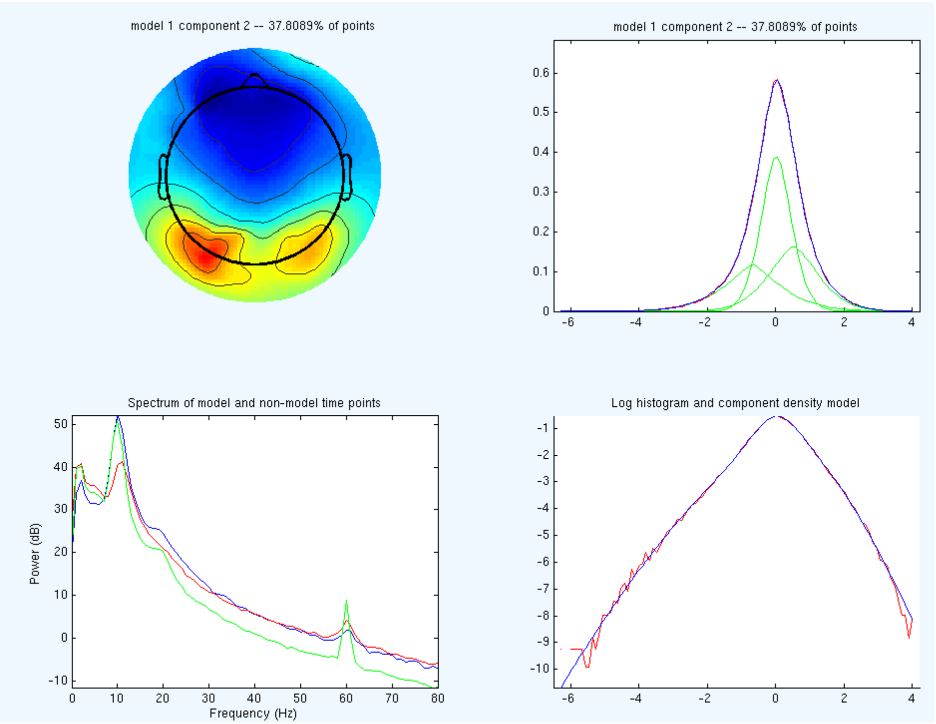
bt73 magenta model components

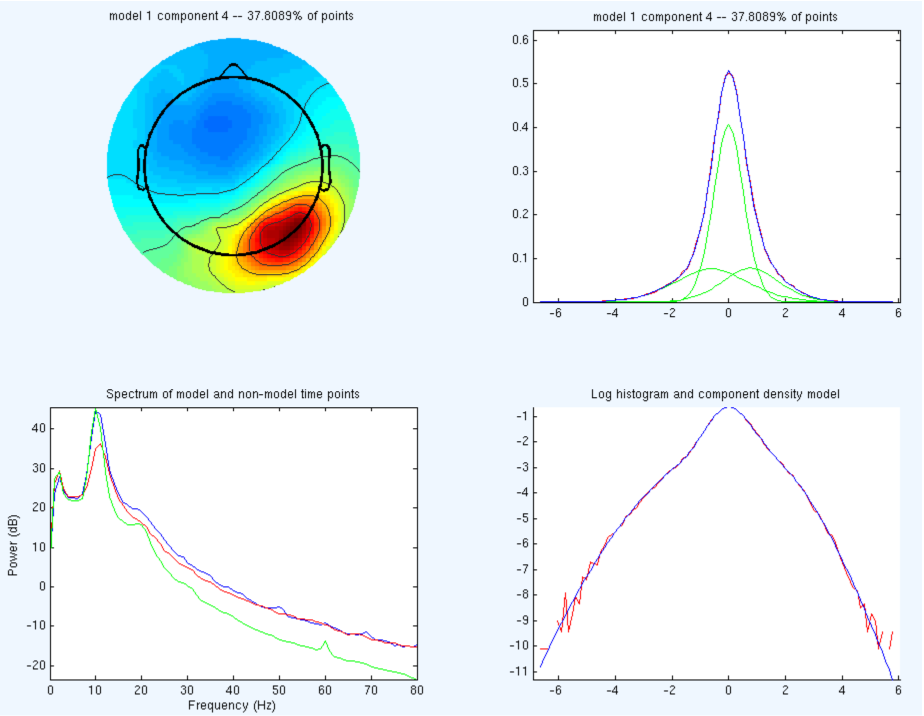
- Mostly eye-blink components (1-12)
- Frontal midline component (13)



bt73 green model alpha

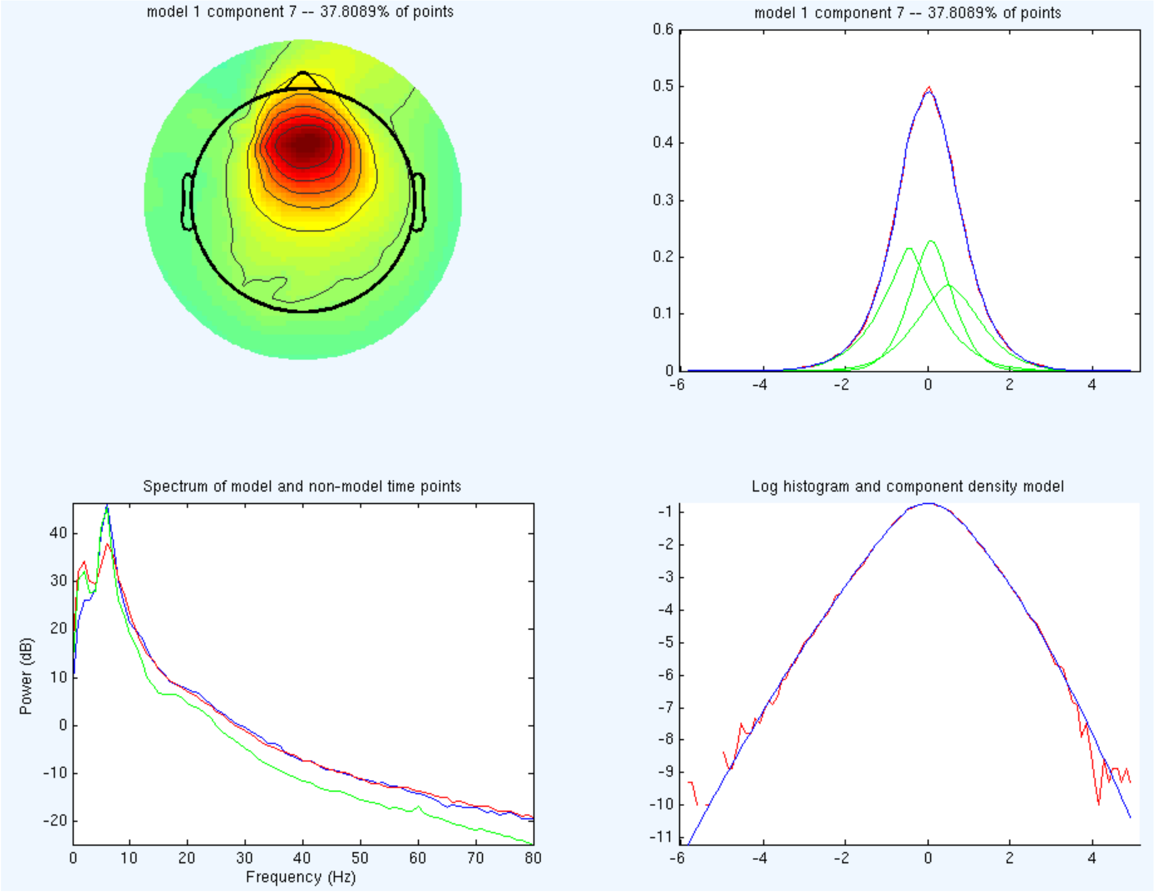
- Components have more power in segments represented by model than in non-model segments





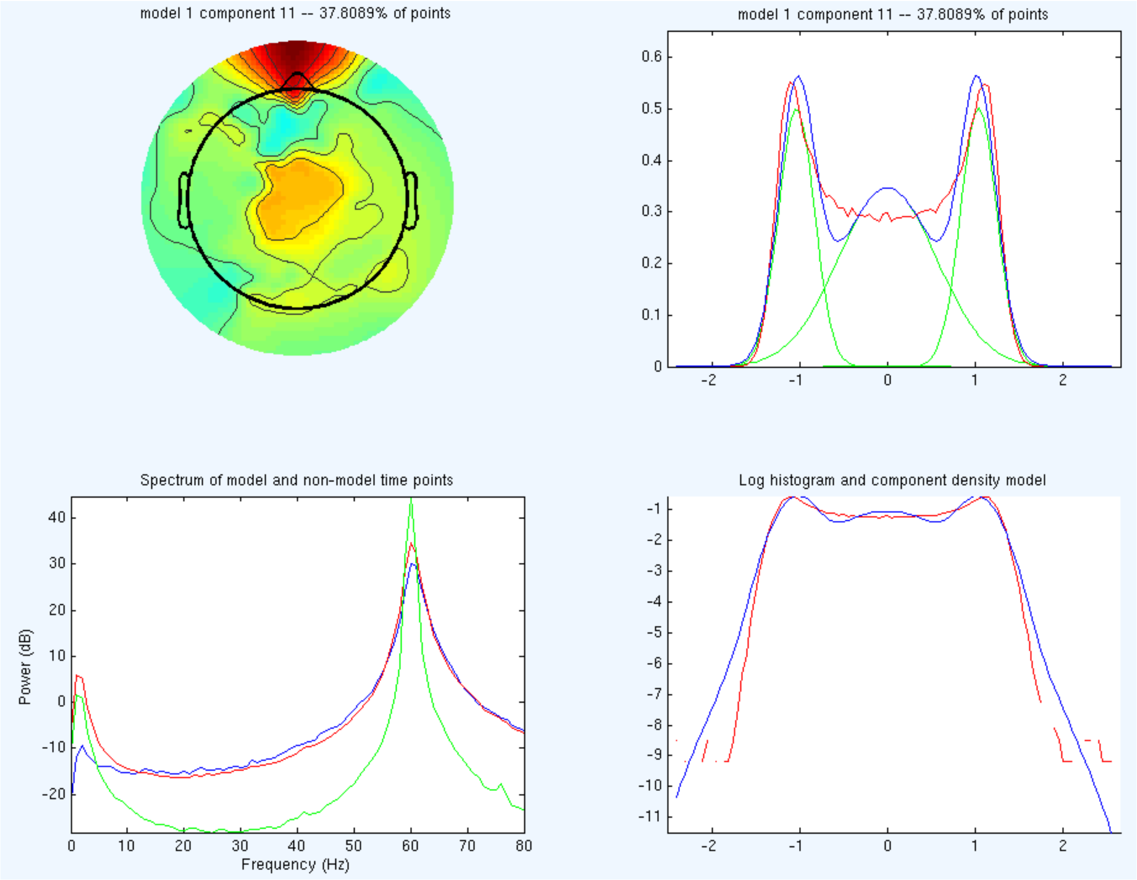
bt73 green model frontal midline θ

- Component again has more power in segments represented by model than in non-model segments



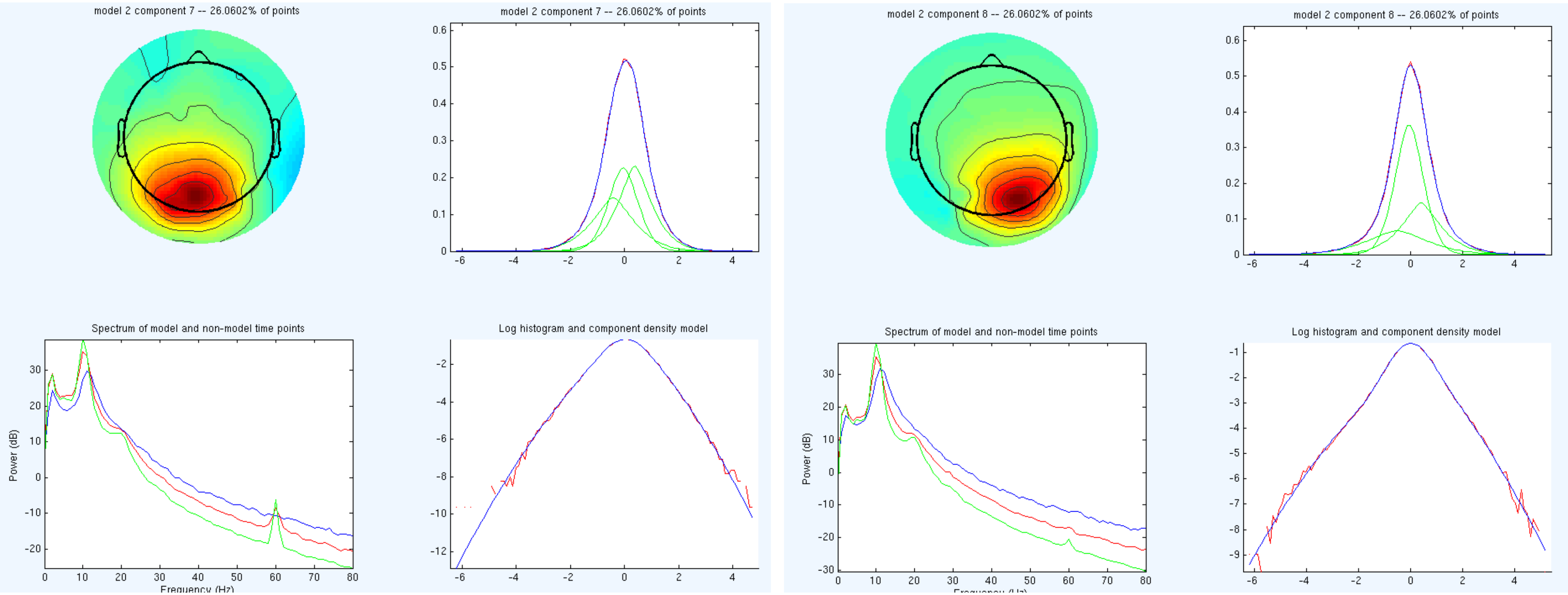
bt73 green model power line comp

- Sub-Gaussian component represented by mixture model of Generalized Gaussian densities



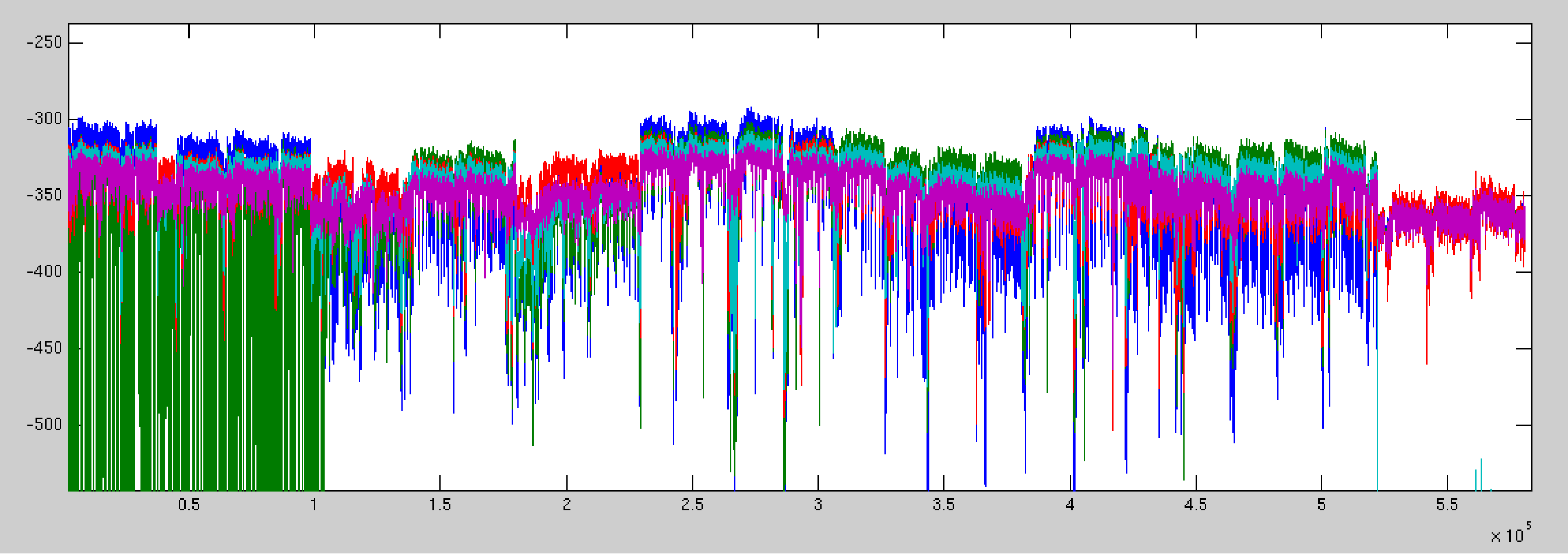
bt73 blue model alpha

- Alpha peak shifted in model segments shifted slightly higher than in non-model time segments

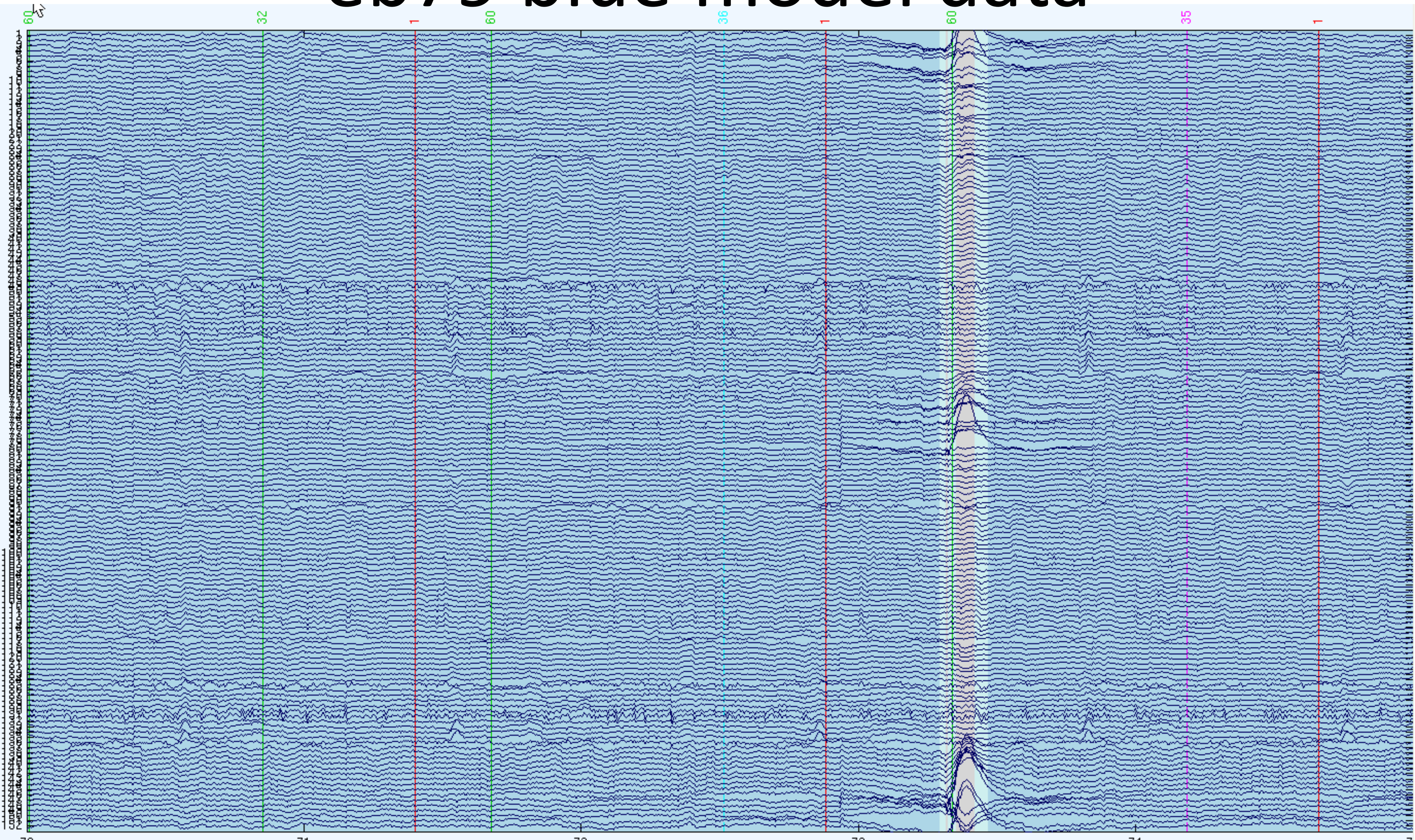


eb79 segmentation

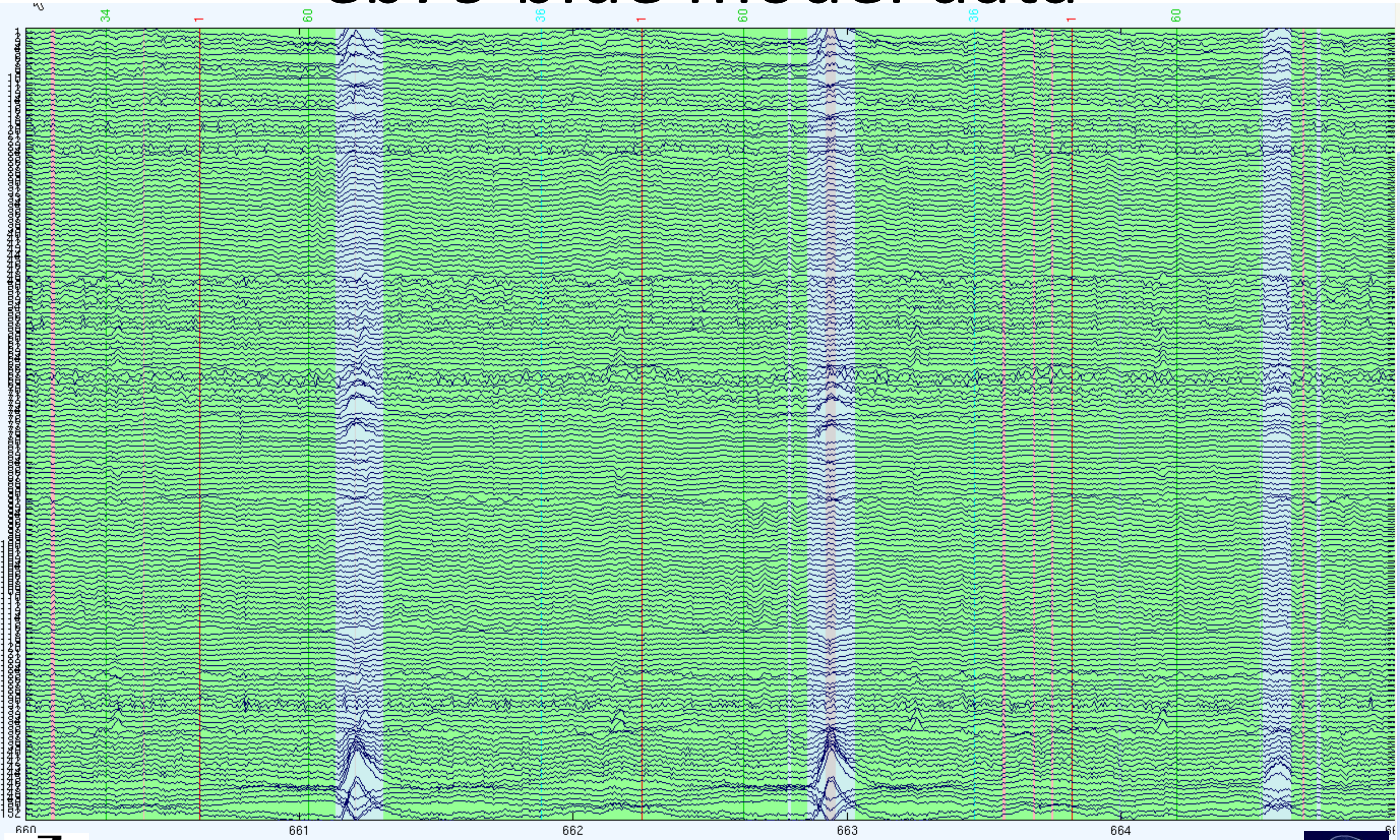
- Task trials are represented by blue, green, and red models
- Red model contains muscle activity not present in blue and green
- Eye blinks represented by cyan and magenta models



eb79 blue model data



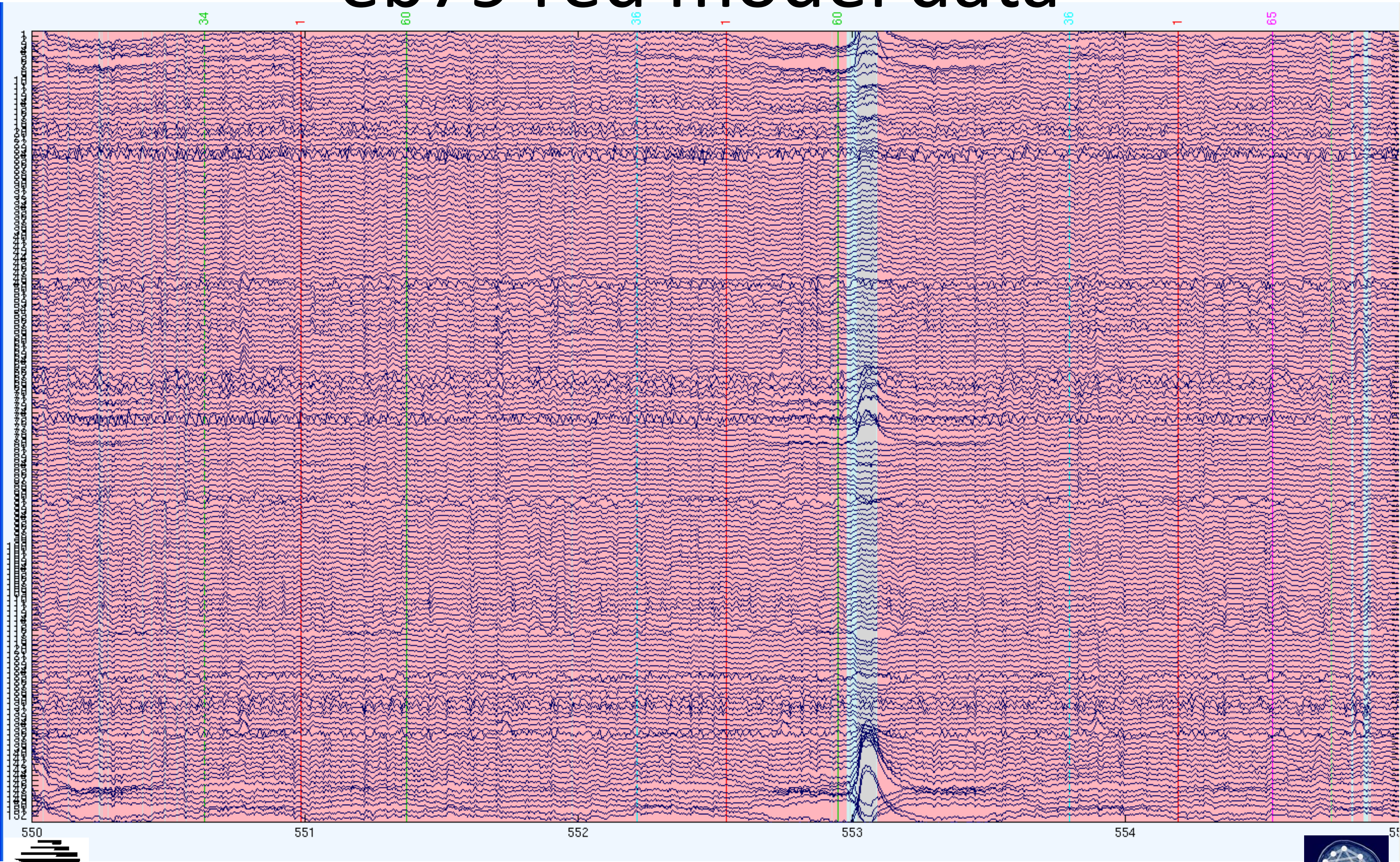
eb79 blue model data



UCSD

Swartz Center for Computational Neuroscience

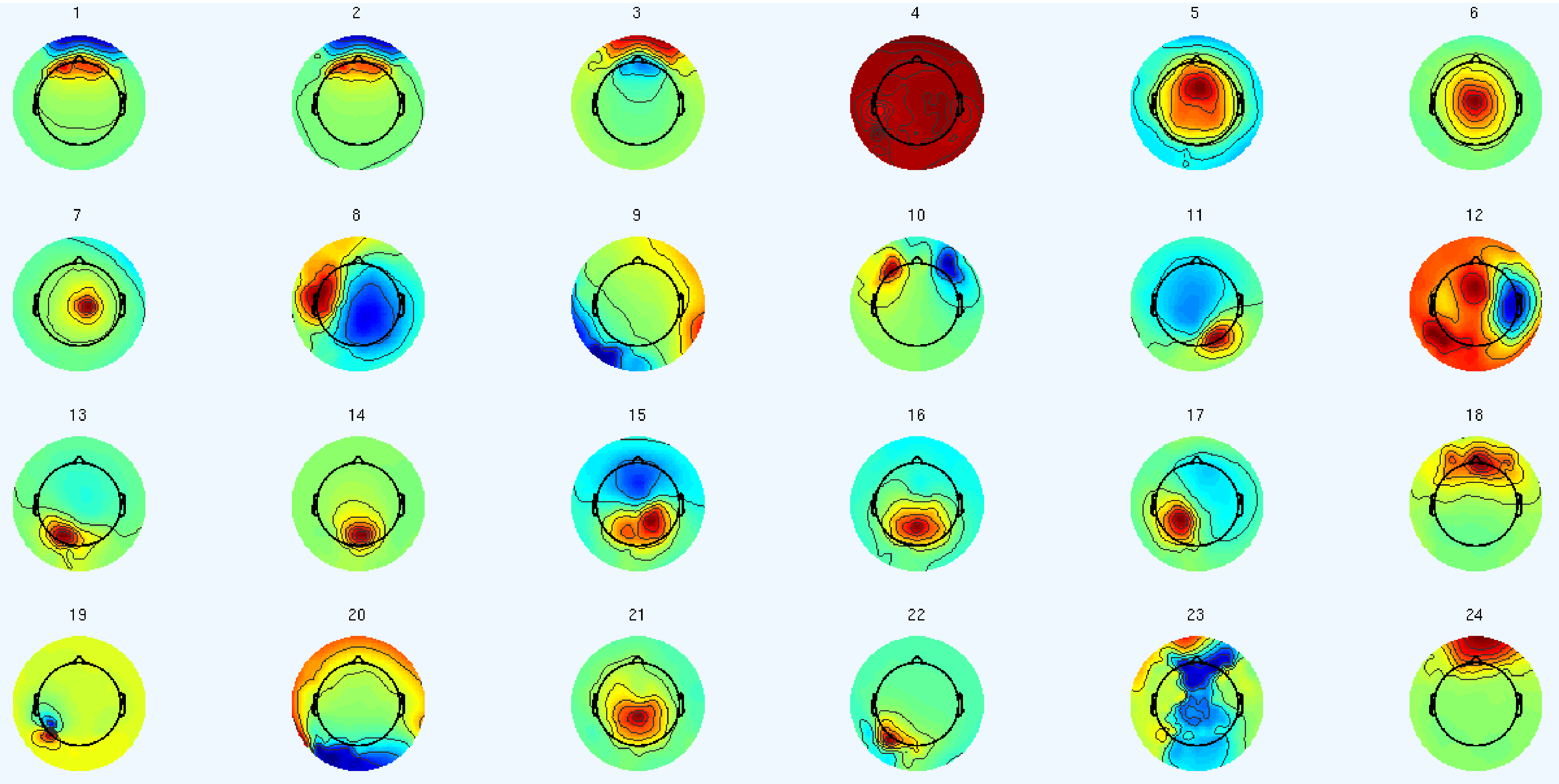
eb79 red model data



UCSD

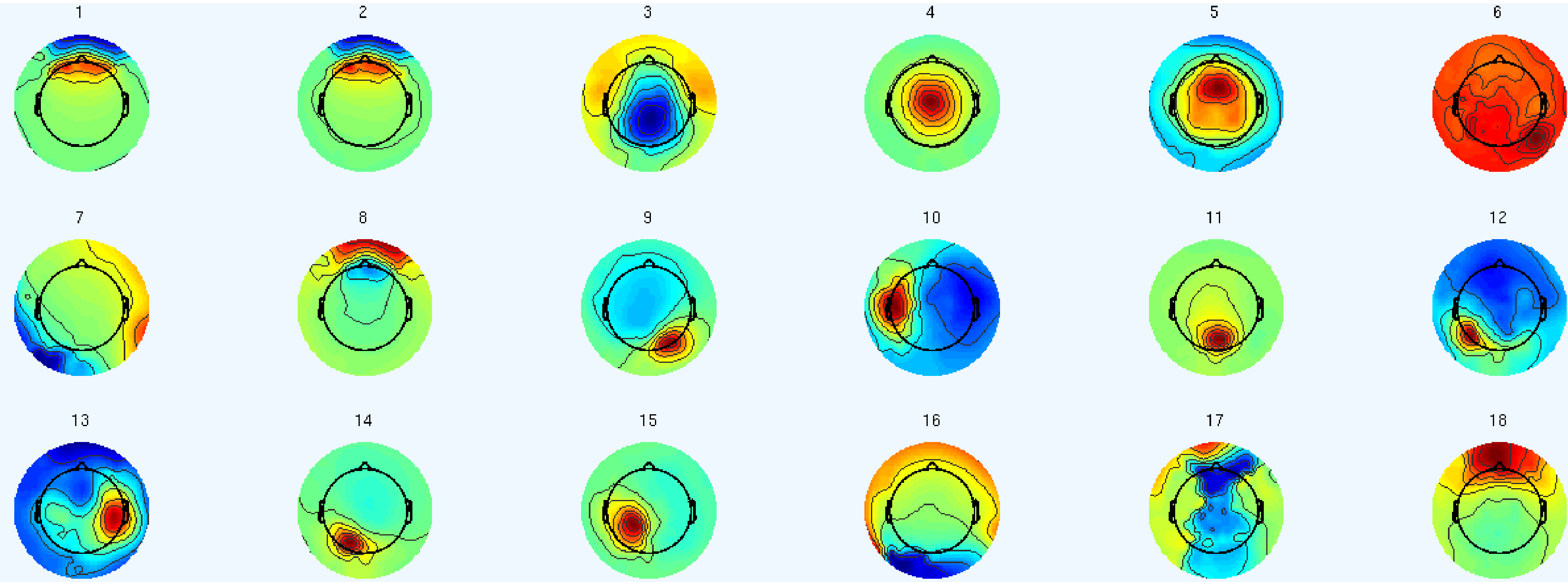
Swartz Center for Computational Neuroscience

bt73 single model components



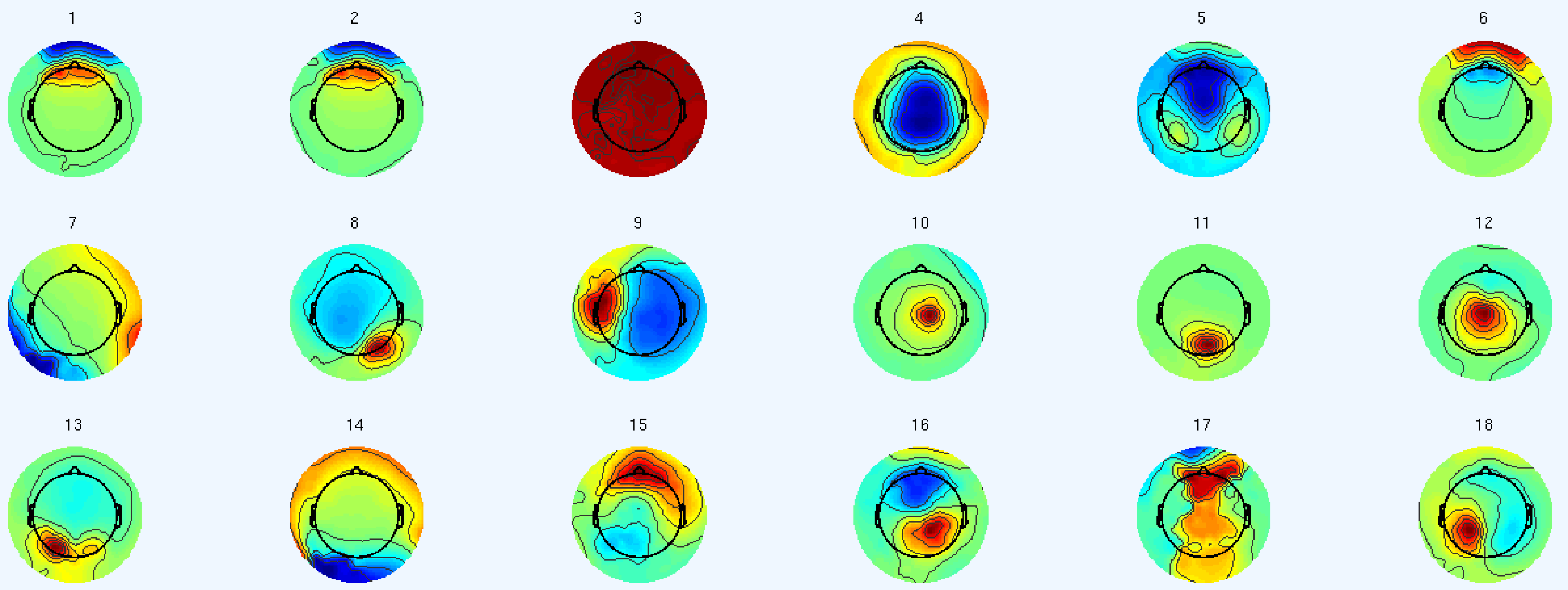
eb79 blue model components

- Prominent midline and occipital alpha components
- Weak mu components



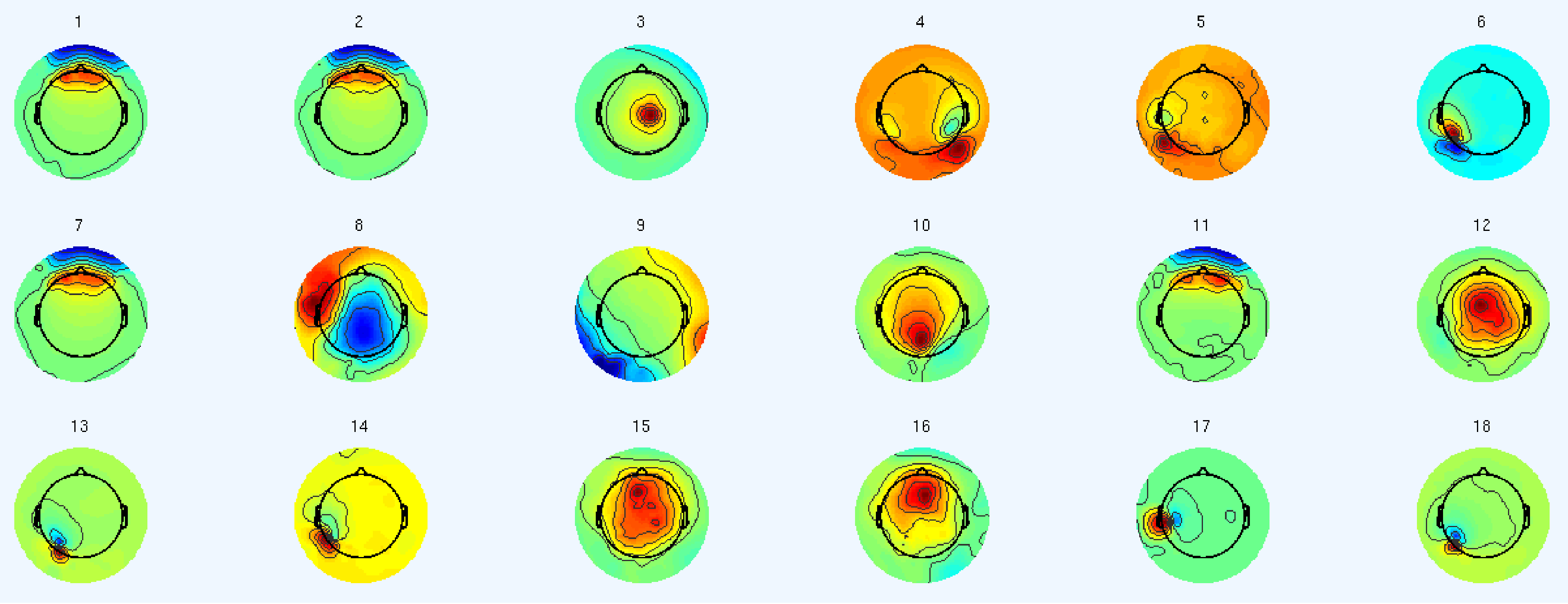
eb79 green model components

- Prominent occipital alpha components
- Weaker frontal midline

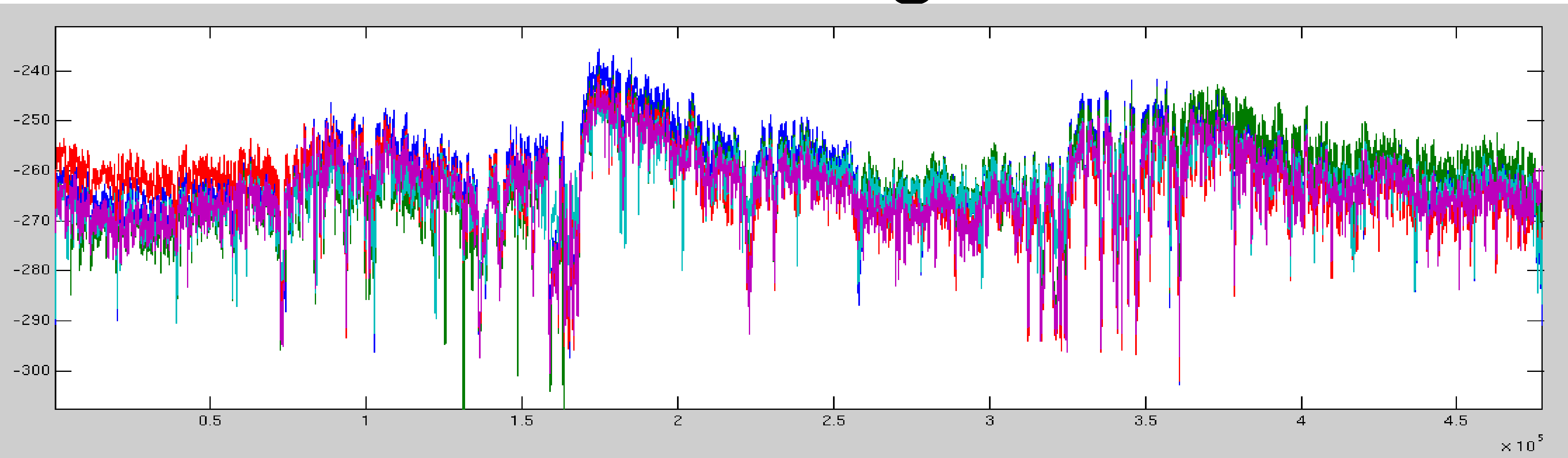


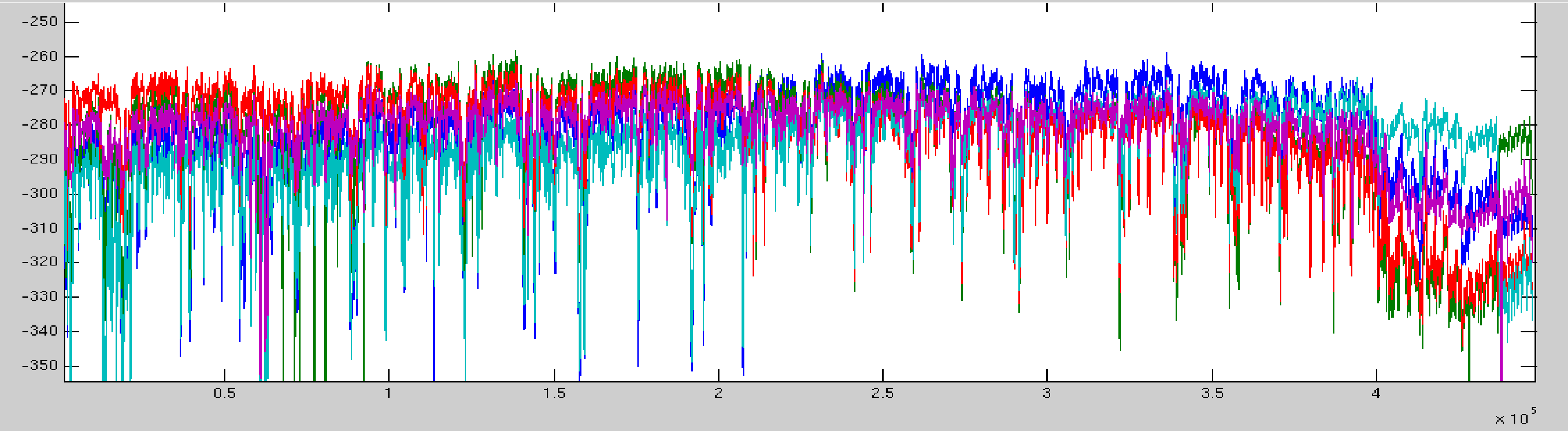
eb79 red model components

- Prominent muscle components (4, 5, 6, 13, 14, 17, 18)



ld81 and dh84 segmentation





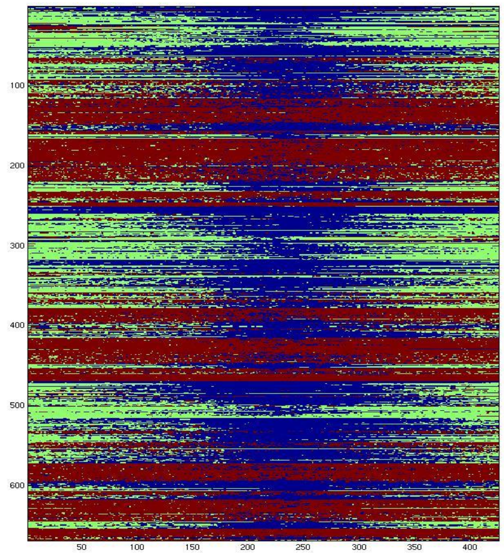
UCSD

Swartz Center for Computational Neuroscience

Consistency over Number of Models

3 models

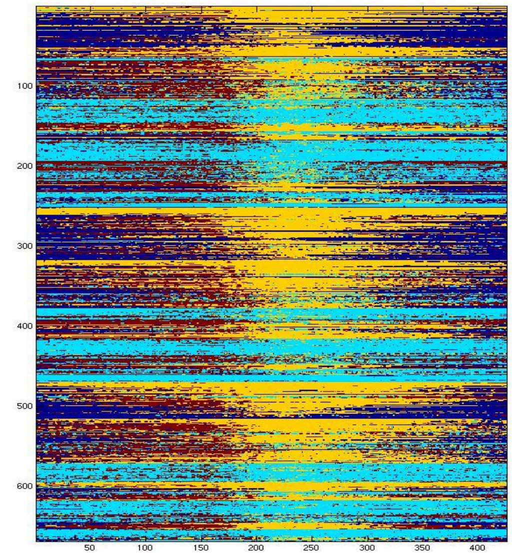
trial



time

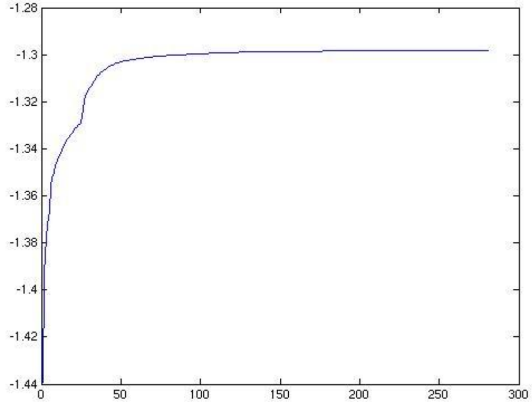
4 models

trial



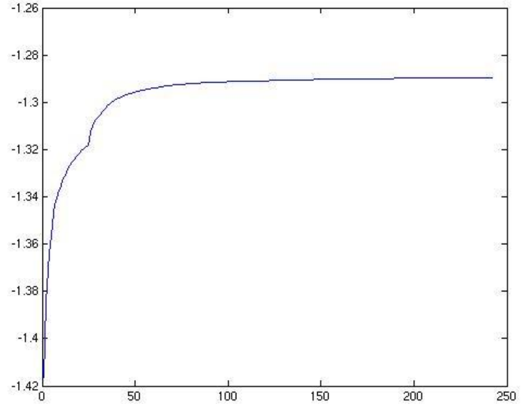
time

log likelihood



iteration

log likelihood

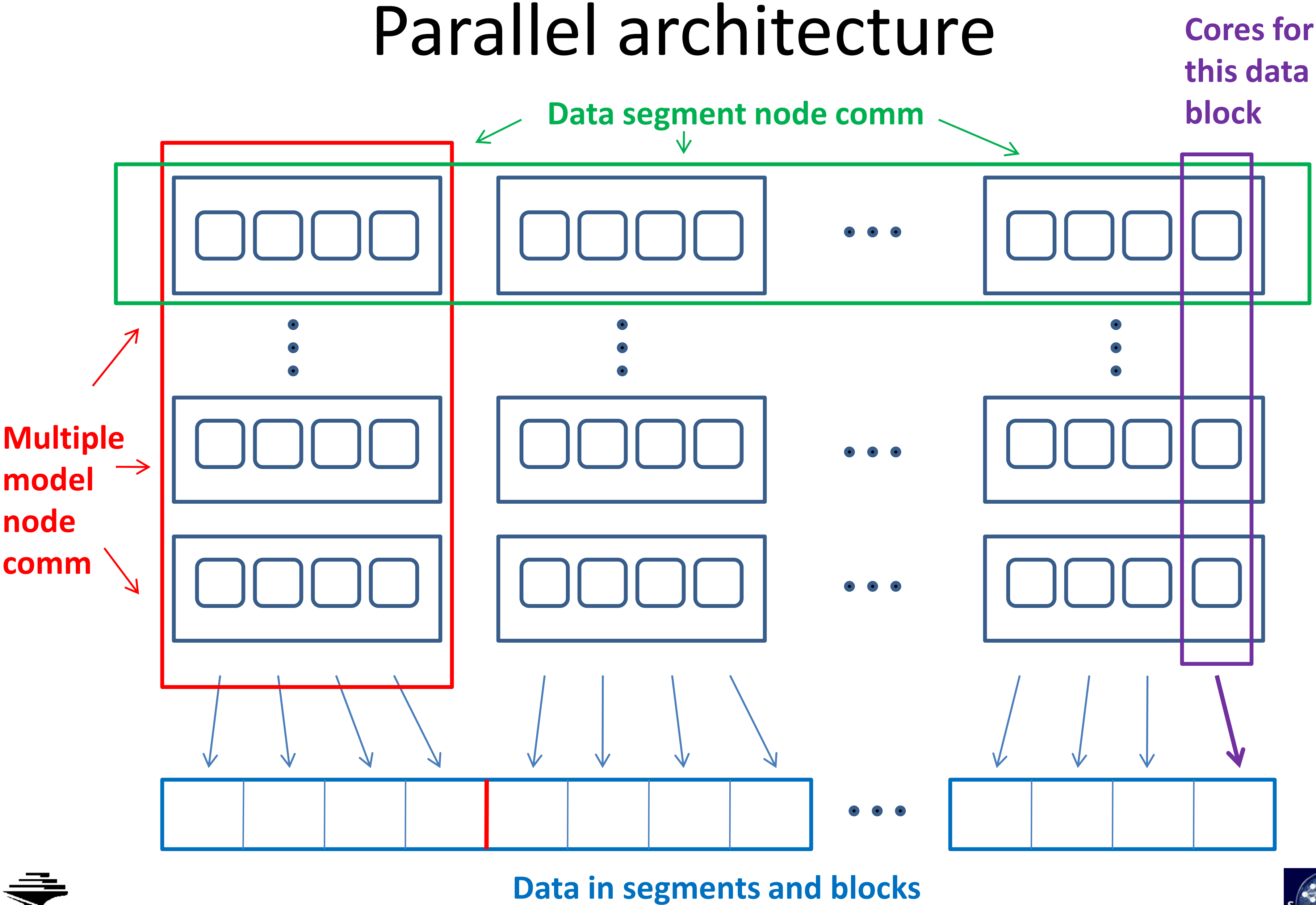


iteration

UCSD

Swartz Center for Computational Neuroscience

Parallel architecture



Parallel architecture (cont.)

- Parallelization is implemented using MPI to parallelize over nodes, and OpenMP to parallelize over cores within a node (using shared memory)
- Data is divided into segments (assigned to nodes) and blocks (assigned to cores)
- Multiple nodes are devoted to the same segment, one for each model
- An "update" is computed for each segment. Two directions of data communication flow:
  – Model nodes communicate to normalize update by likelihood of segment over all models
  – Segment nodes communicate to average the segment updates into one global update of parameters
- Global update computed at root node and sent back to model and segment nodes
- Also implemented with unstructured collection of cores for random assignment on large cluster
- Portable implementation allows execution on many platforms, including Teragrid, an NSF project with NCSA, SDSC, and others

Take Home Messages

- With sufficient amount of data, **multiple ICA models can be estimated simultaneously** and used overcome non-stationarity and segment data.
- **Newton method** significantly improves convergence rate, and conditioning in multiple model case.
- **Arbitrary source densities** modeled with non-Gaussian source mixture model.
- Likelihood can be conveniently used to **reject data**.
- **Some EEG sources really are stationary** (eyes, heartbeat, power line, frontal midline, mu, etc.) These should be identified across models to improve efficiency of estimation (in progress). Alpha components seem to be variable.

Code and Papers

- There is Matlab code available!
  - Generate toy mixture model data for testing
  - Full method implemented: mixture sources, mixture ICA, Newton
- Paper draft available, with derivation of mixture model Newton updates
- Download from:

http://sccn.ucsd.edu/~jason

Acknowledgements

- Thanks to Scott Makeig, Julie Onton, Gráinne McLoughlin, Ruey-Song Hwang, Rey Ramirez, Diane Whitmer, and Allen Gruber for collection and consultion on EEG data
- Thanks to Jerry Swartz for founding and providing ongoing support the Swartz Center for Computational Neuroscience

- Thanks for your attention!