Gaussian process regression and recurrent neural networks for fMRI image classification

Emanuele Olivetti[1,2], Diego Sona[1], Sriharsha Veeramachaneni[1]

{olivetti|sona|sriharsha}@itc.it

(1)ITC/Irst
(2)ICT International Doctorate School, University of Trento

Via Sommarive, Povo – Trento (Italy)

ITC irst
CENTRO PER LA RICERCA SCIENTIFICA E TECNOLOGICA

UNIVERSITY OF TRENTO - Italy

Or better...

Dimensionality reduction and recurrent neural networks for feature rating prediction from fMRI data

Emanuele Olivetti[1,2], Diego Sona[1], Sriharsha Veeramachaneni[1]

{olivetti|sona|sriharsha}@itc.it

(1)ITC/Irst
(2)ICT International Doctorate School, University of Trento

Via Sommarive, Povo – Trento (Italy)

ITC irst
CENTRO PER LA RICERCA SCIENTIFICA E TECNOLOGICA

UNIVERSITY OF TRENTO - Italy

Outline

- Foreword
- Pre-Processing:
  - Smoothing
  - Mutual Information
  - Clustering
- Prediction:
  - Recurrent Neural Networks (NO GPR!)
- Results & Conclusion
- Future Work

Prologue: February 2006

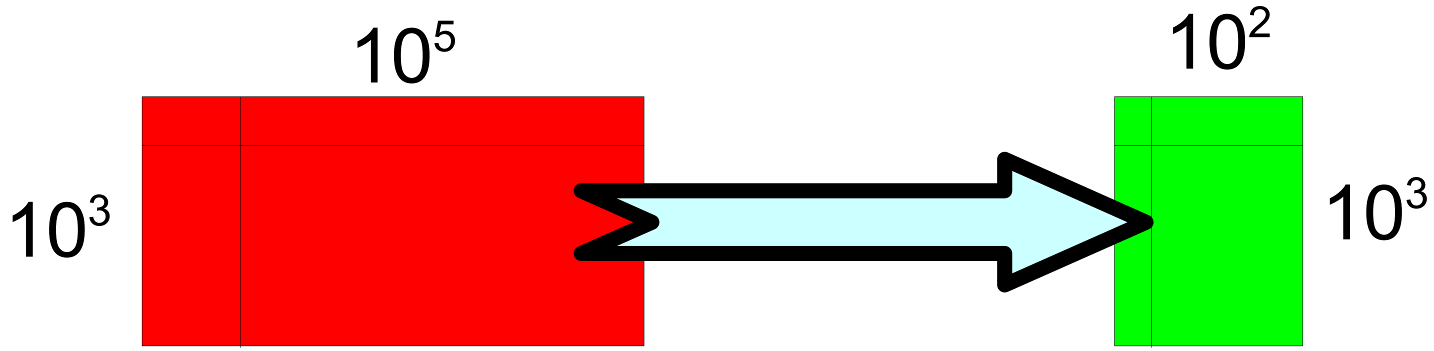
- no prior domain knowledge (wikipedia)
- no specific software
- just for fun, starting from scratch

Initial Dataset:

- **3**x**3** Pre-processed and Spatially Normalized Brain Image Data (Analyze format): INPUT
- **2**x**3**x27(**19**) Feature ratings data convolved with a hemodynamic filter : TARGET

Schema of our Approach

- Create a generic model to map fMRI data of a single subejct to a single Target Feature Rating
  - **Pre-processing**: reduce noise and dimensionality of the input dataset (*curse of dimesionality*)

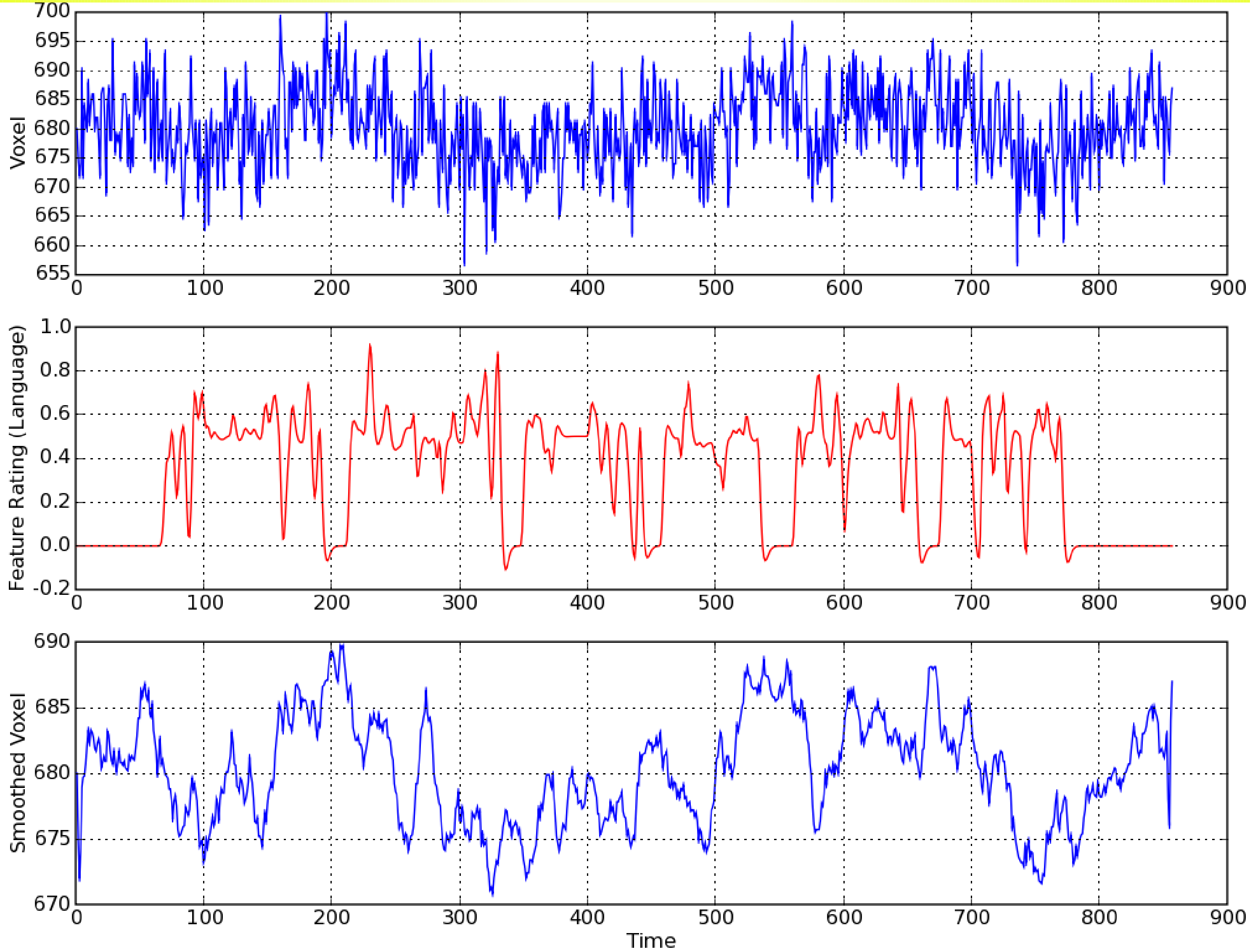


  - **Predict** Feature Ratings using non-linear time-dependent regression

From now on: Subject 1 and Language

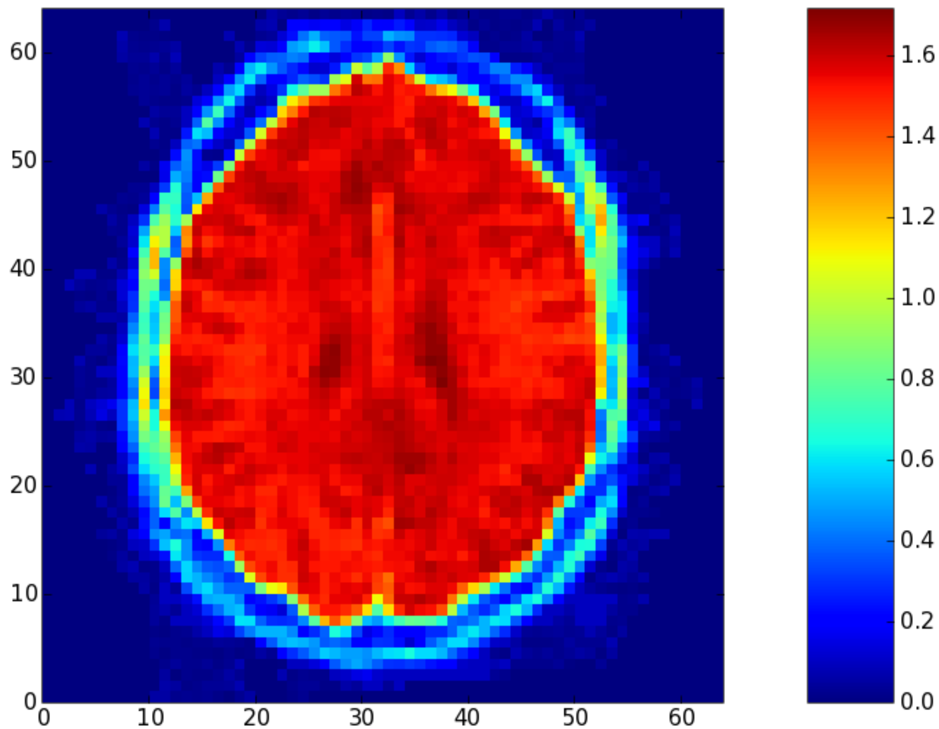
Pre-processing

Pre-Processing (1): Smoothing



Pre-Processing (2): Mutual Information

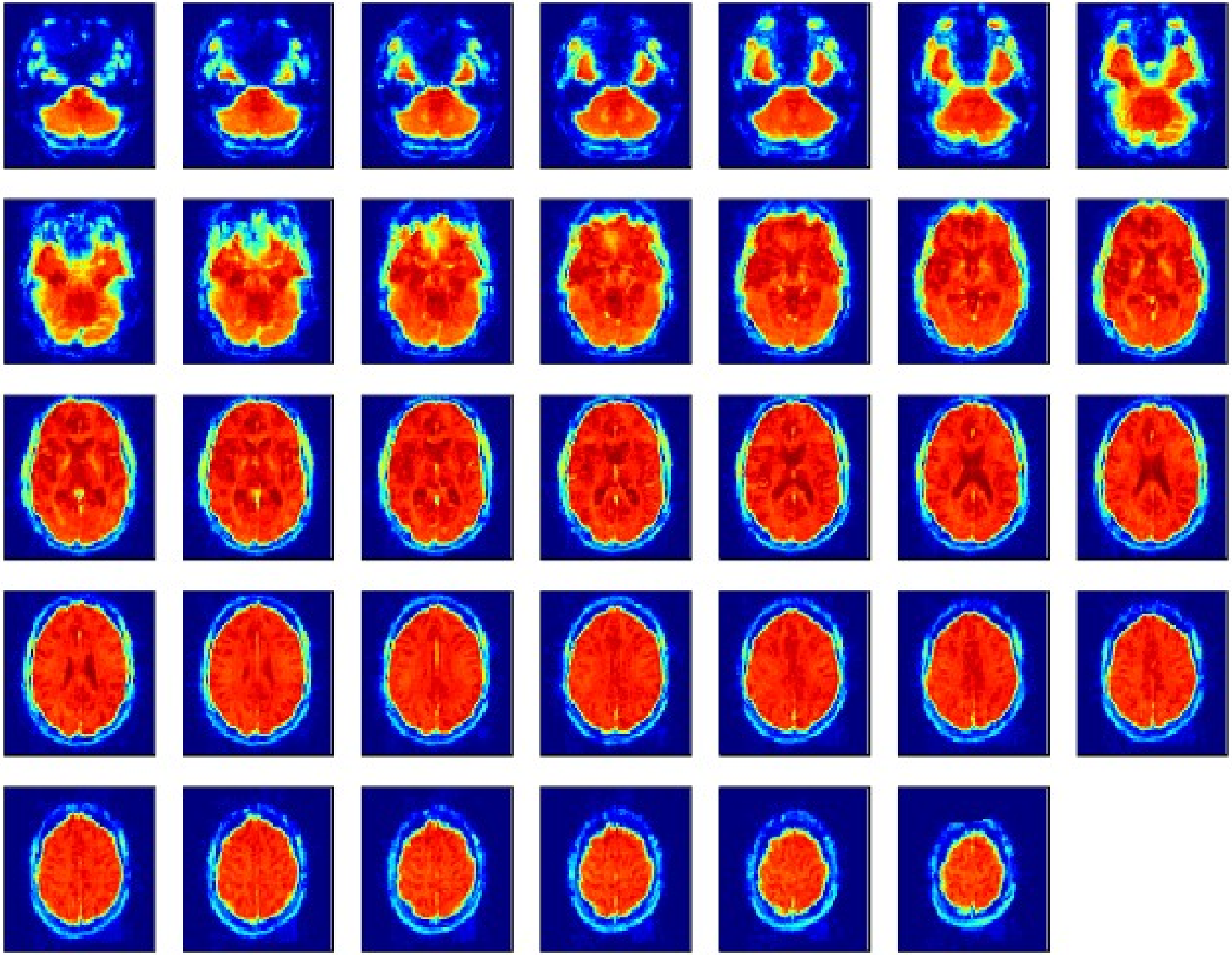
$$I(X;Y)=\sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)\,p(y)}$$



Subject 1 – Movie 1 – Language: slice 22

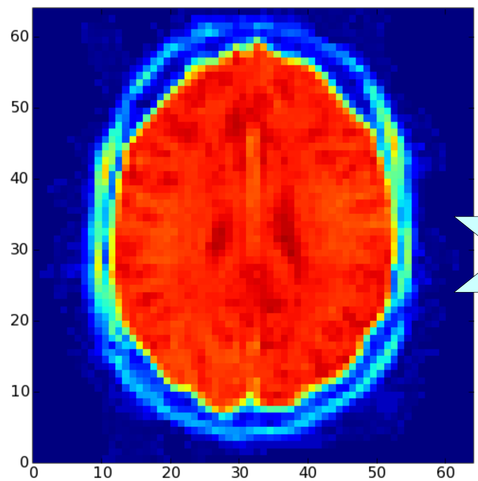
- **Mutual Information** measures "informativeness" of a voxel with respect to a given Target Feature
- For each voxel: compute Mutual Information between its values in time and each feature rating

Pre-processing (2): Mutual Information



Language – Subject 1, Movie 1

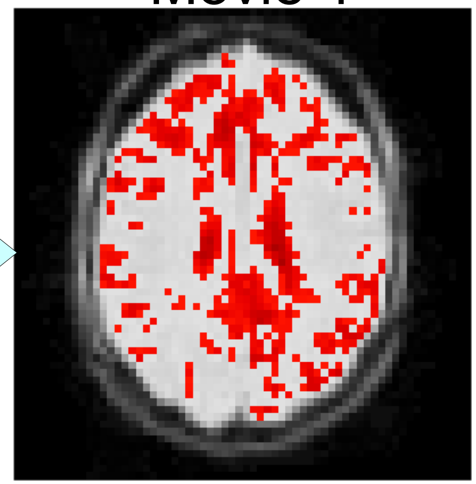
Pre-processing (2): Mutual Information
Thresholding and Merging



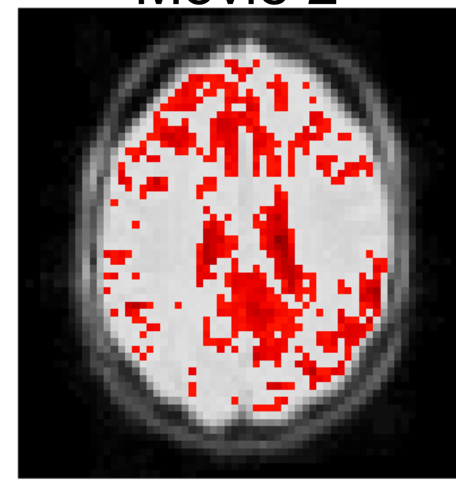
Subject 1 – Movie 1 – Language
slice 22

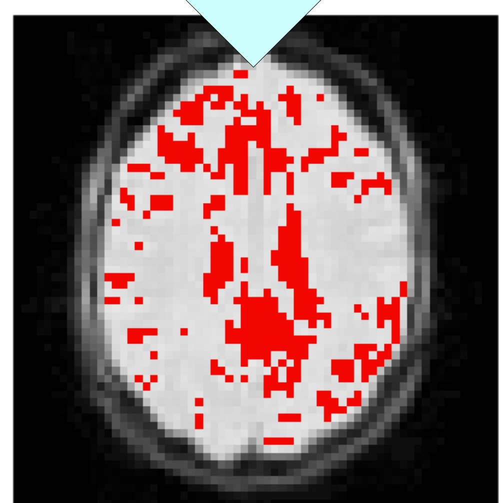
top
1.5×10^4
voxels

Movie 1



Movie 2

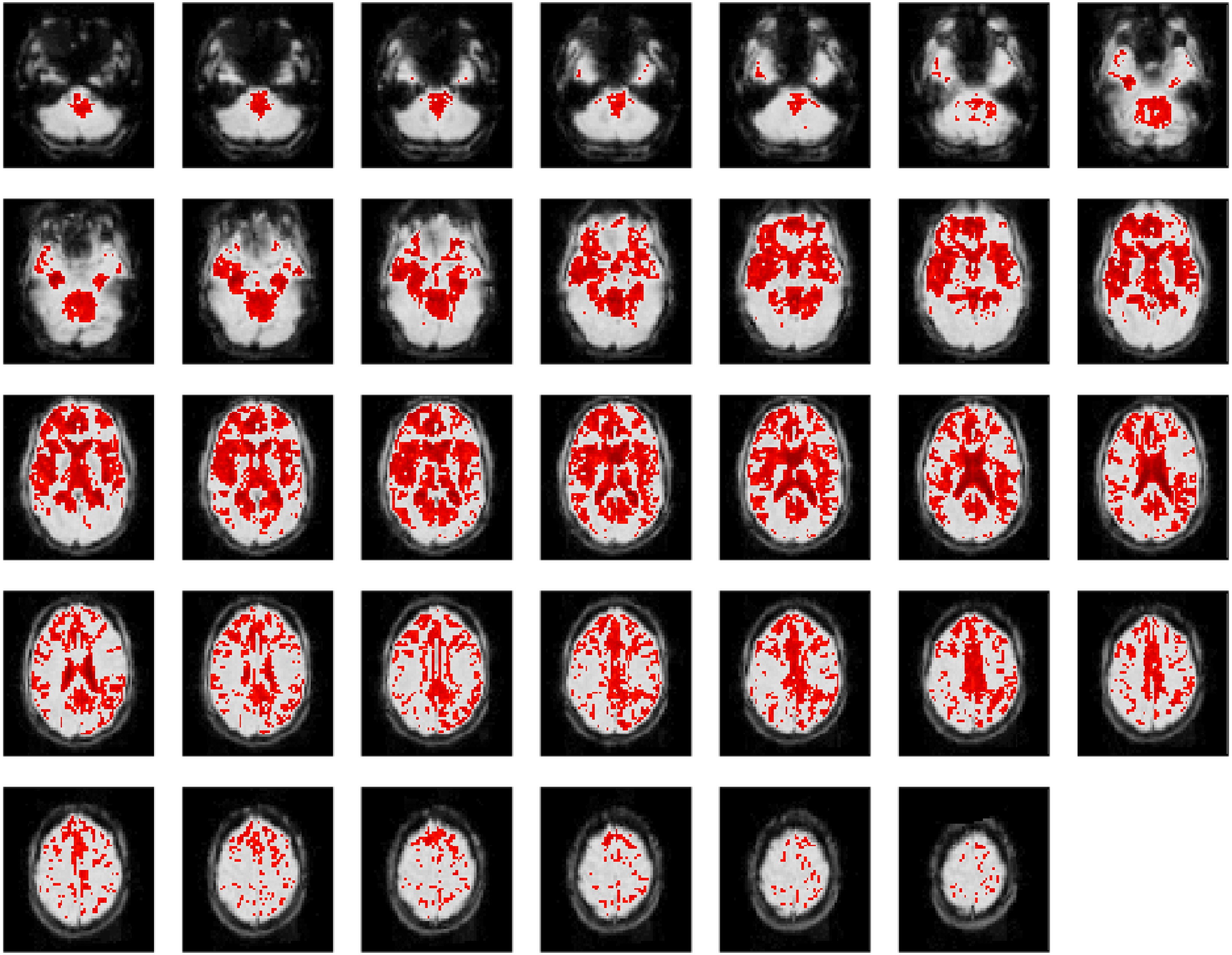




~10^4 voxels

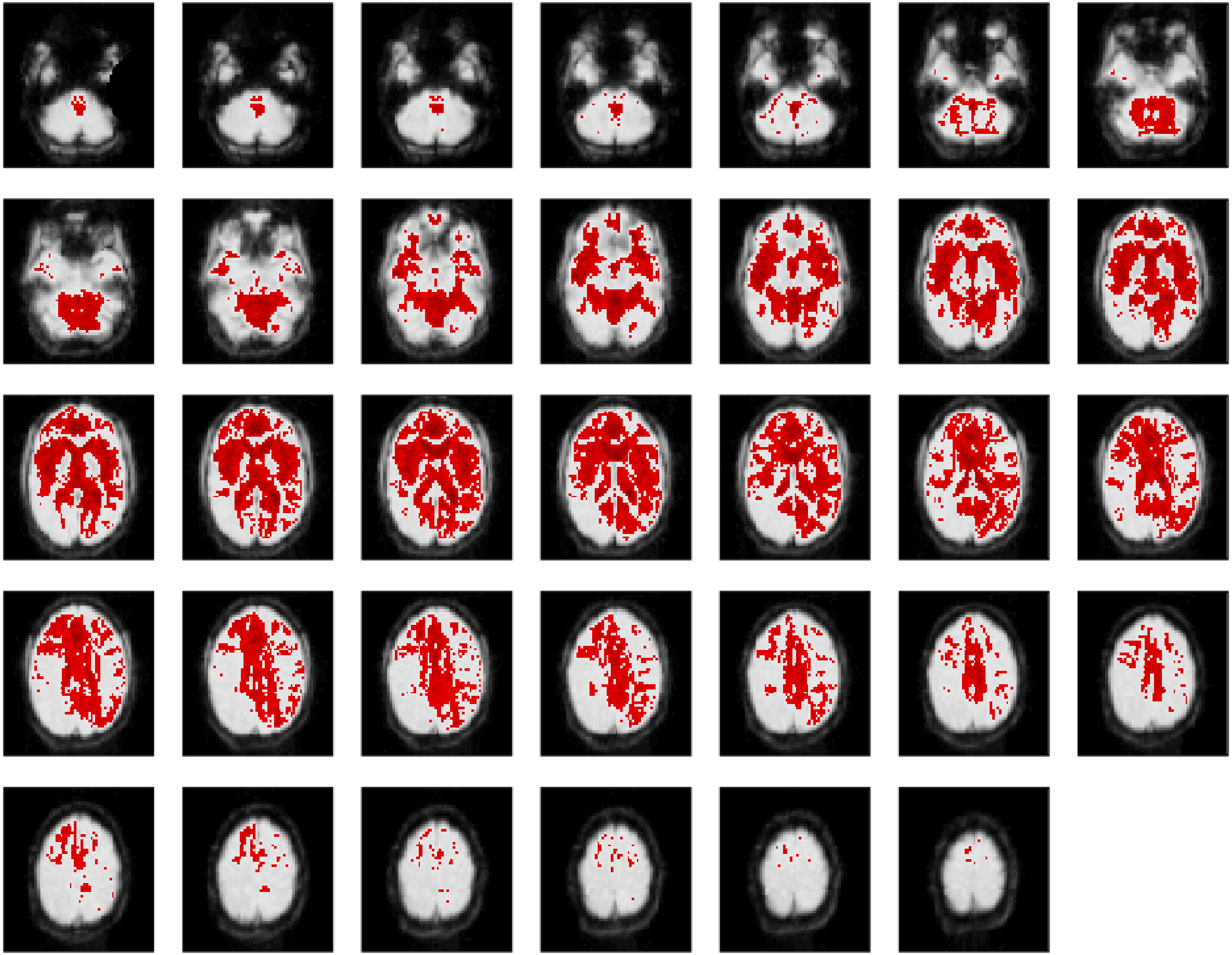
- Merging = Intersection
- ~80% overlap between movie 1 and movie 2

Pre-processing (2): Mutual Information - Thresholding and Merging



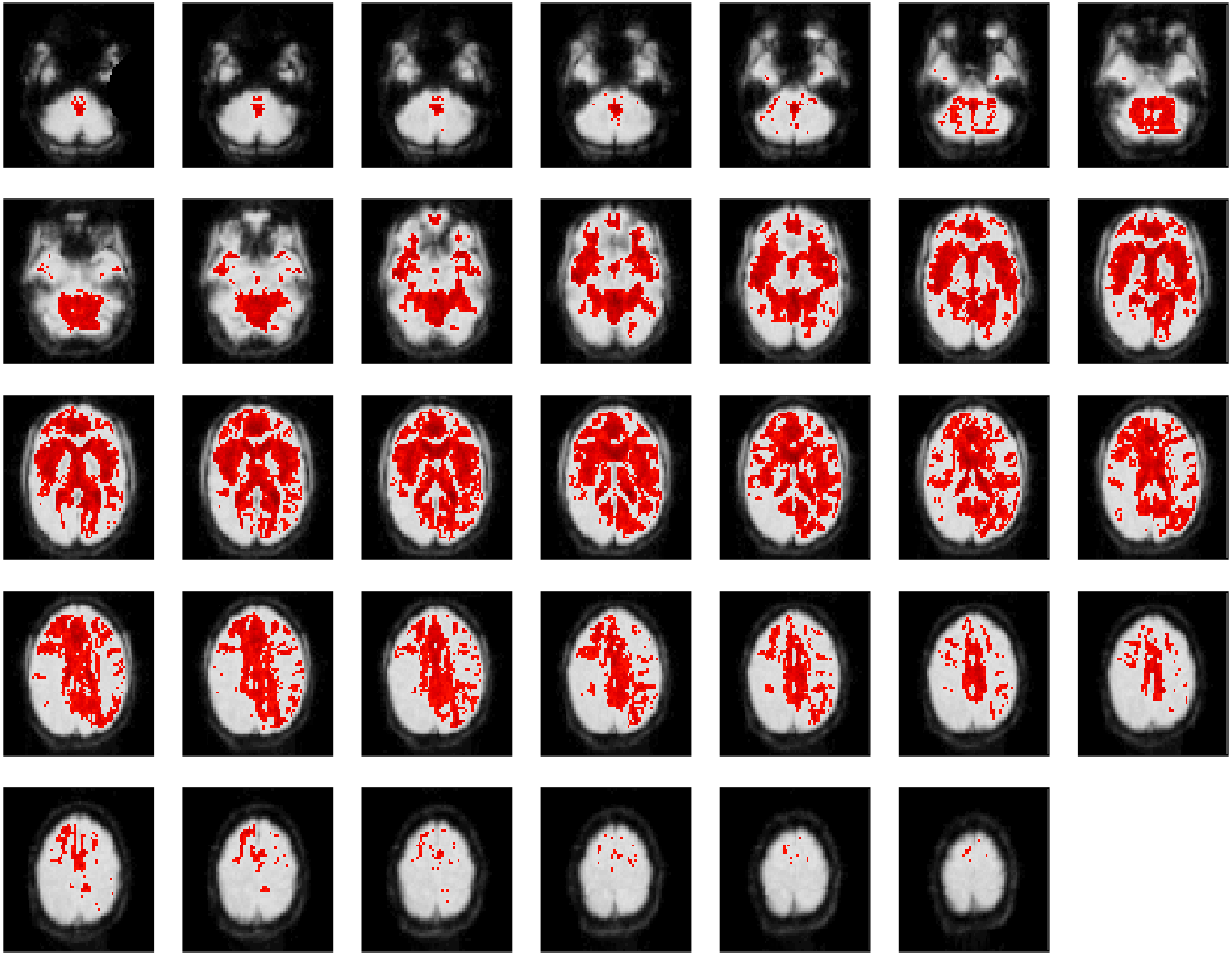
Language – Subject 1

Pre-processing (2): Mutual Information - Thresholding and Merging



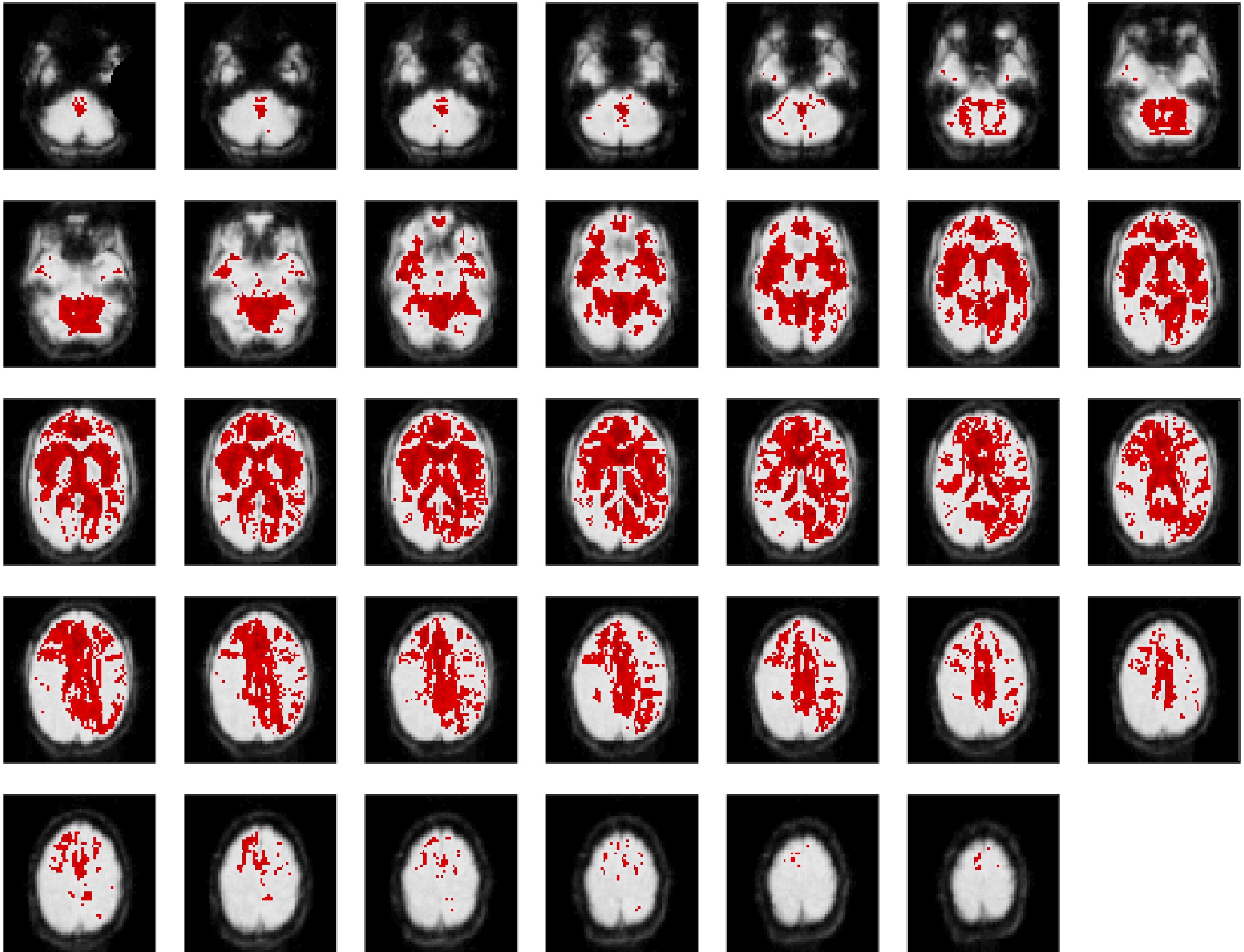
Language – Subject 2

Pre-processing (2): Mutual Information - Thresholding and Merging



Music – Subject 2

Pre-processing (2): Mutual Information - Thresholding and Merging



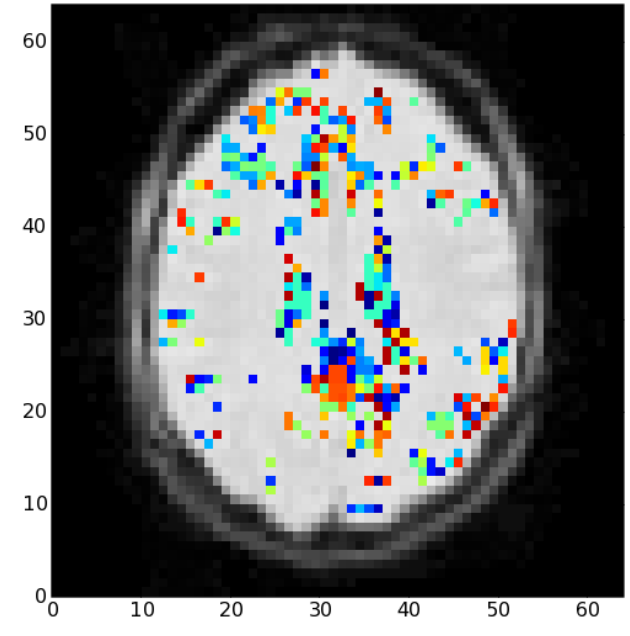
Motion – Subject 2

Pre-Processing (3): Clustering

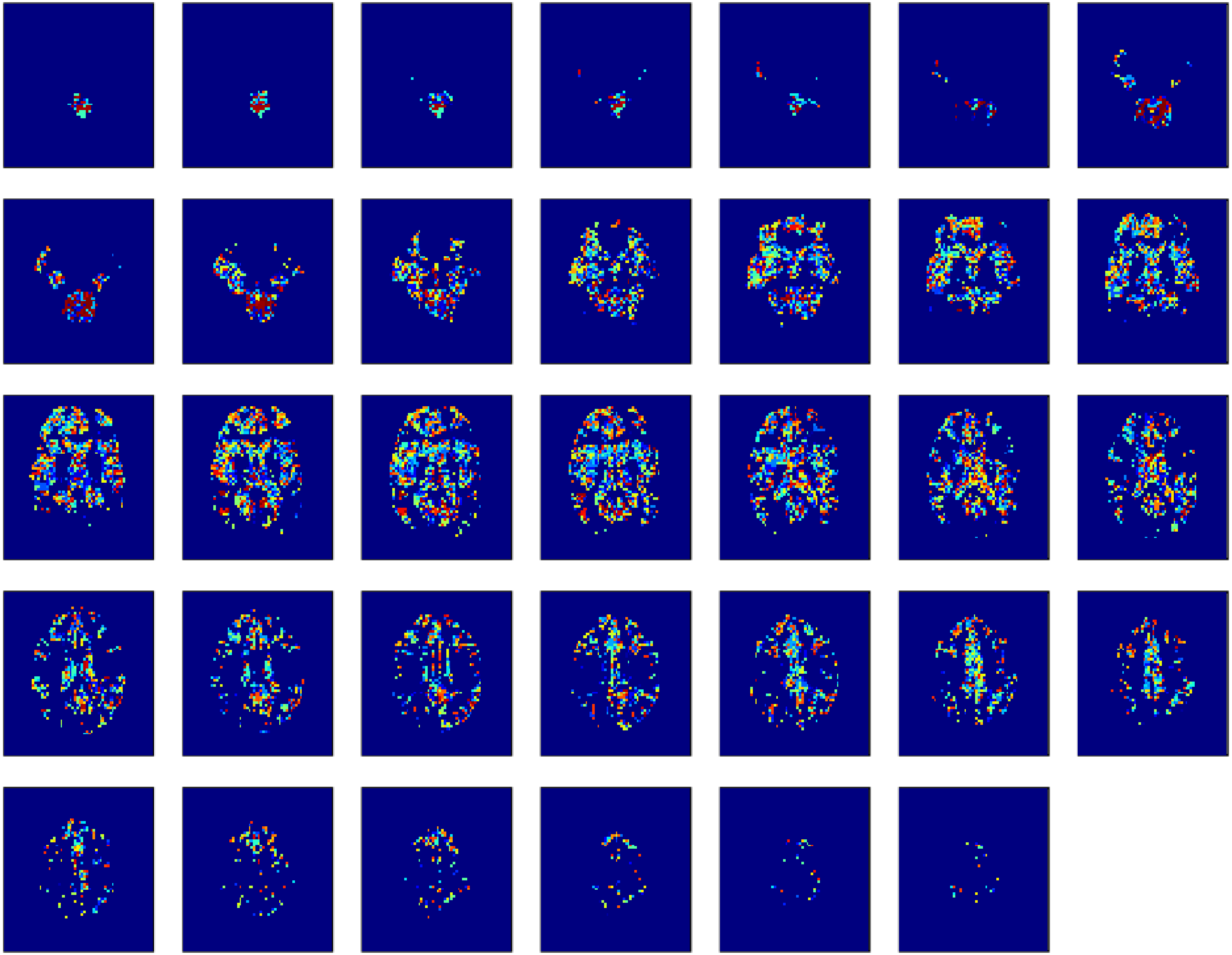
- Idea: reduce 10^4 voxels to ~10^2 representatives
- K-Means clustering, K=200
- d(voxel$_A$,voxel$_B$) = f(spatial distance, temporal correlation)

$$d = d_{spatial}^{\alpha}\left(1 - r_{temporal}\right)^{1-\alpha}$$

- α = 0.5
- correlation: movie 1*+2*+3

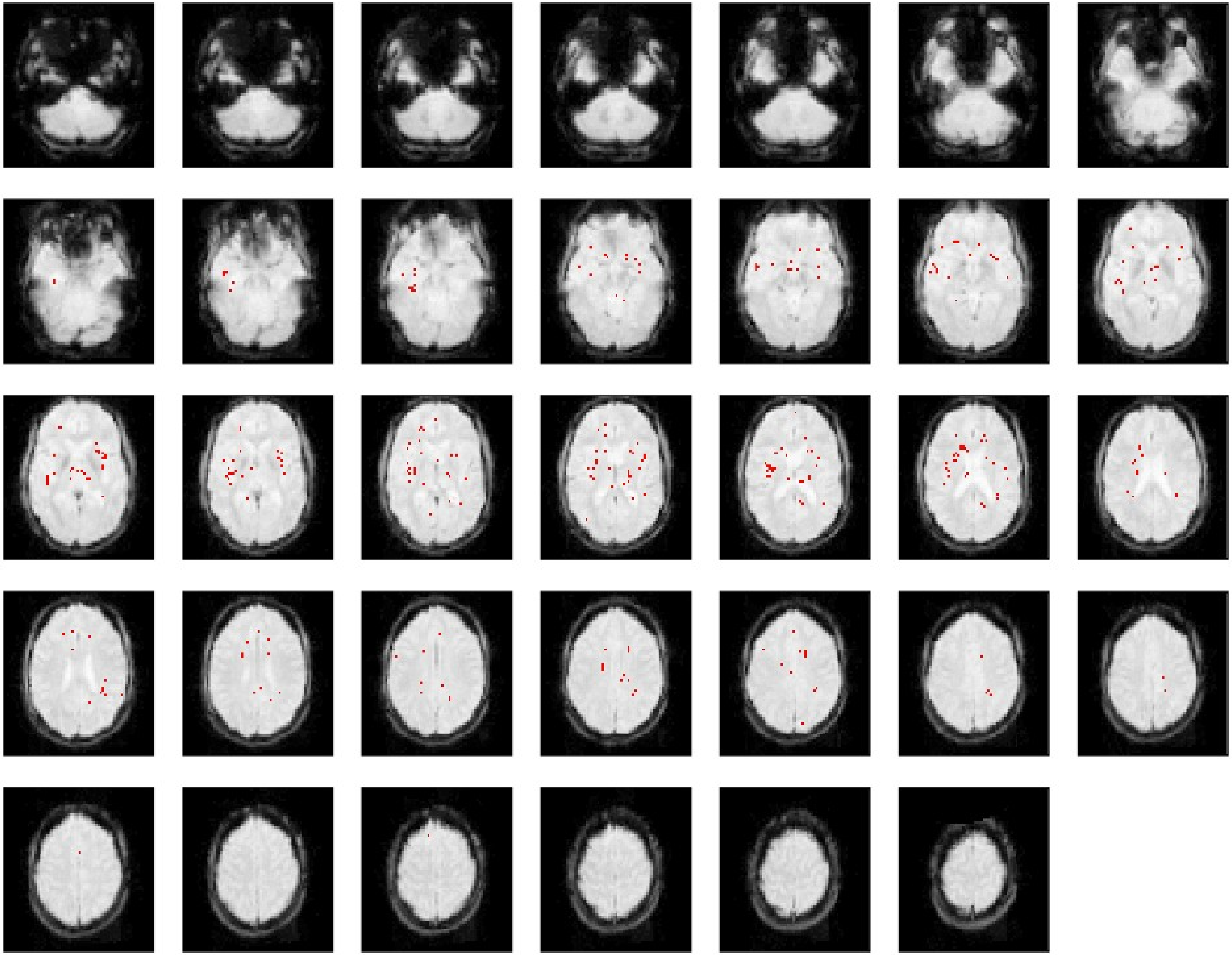


Pre-processing (3): Clustering



Language – Subject 1

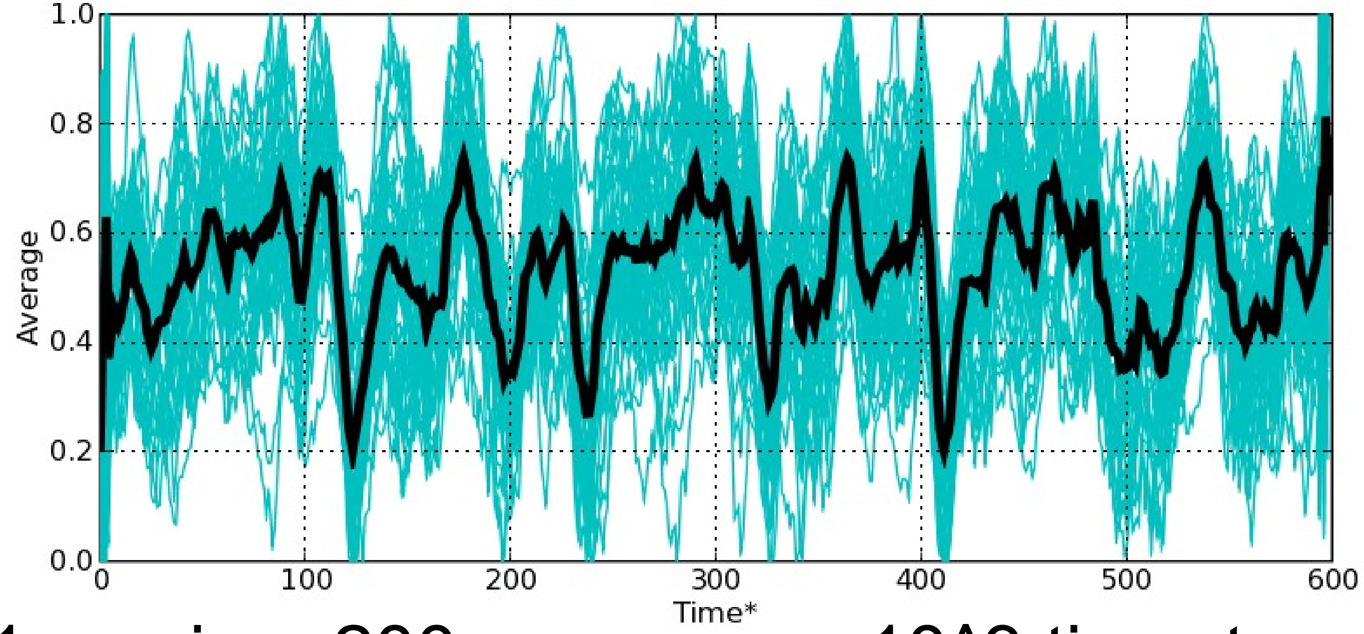
Pre-processing (3): Clustering



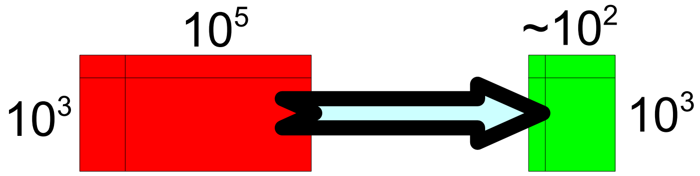
Language - Subject 1 - cluster 21 (~300

Pre-processing (3): Clustering
Averaging

- For each cluster compute **average** over its voxels:
  - 1 cluster (= ~10^2 voxels) → 1 average



  - 1 movie = 200 averages x 10^3 timesteps

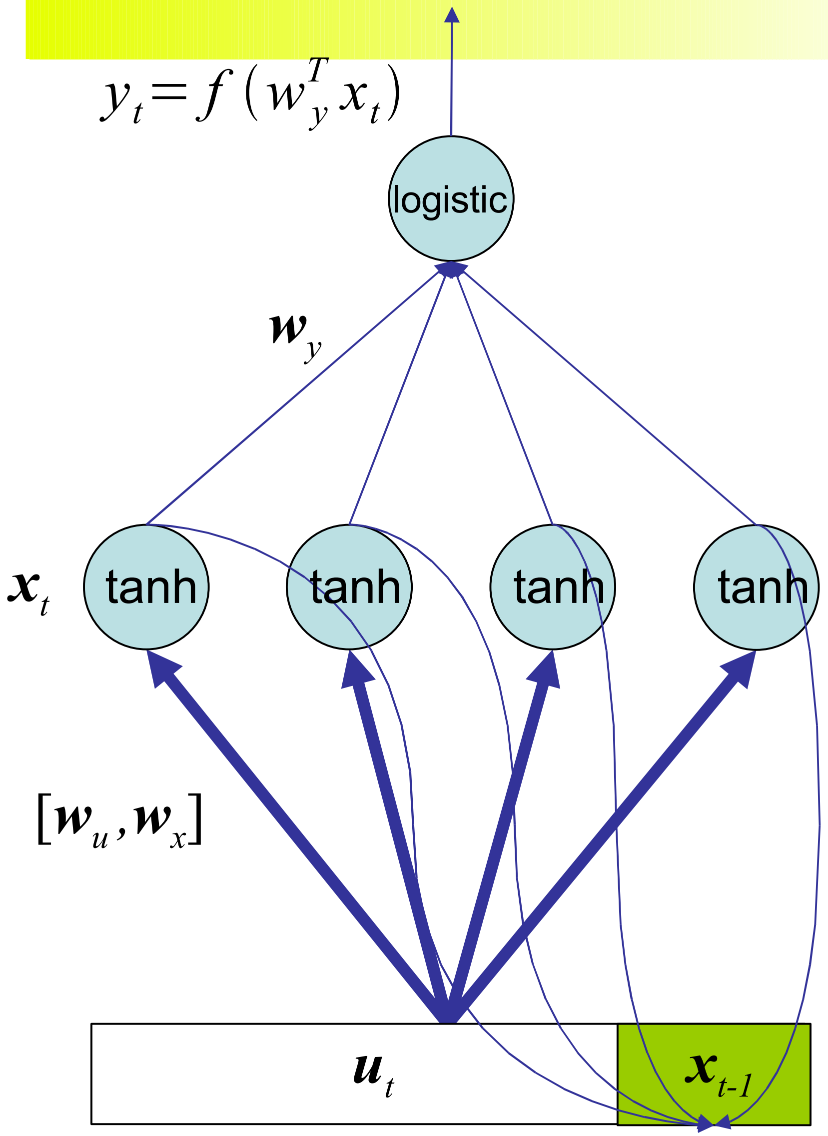


Prediction

Prediction: mixture of **R**ecurrent **N**eural **N**etworks

- RNN pros:
  - simple/standard
  - non-linear
  - exploit time-dependence inputs (brain and movie inertia)
  - no need for priors on data distribution
- RNN cons:
  - RNN could overfit

Prediction: RNN - Model



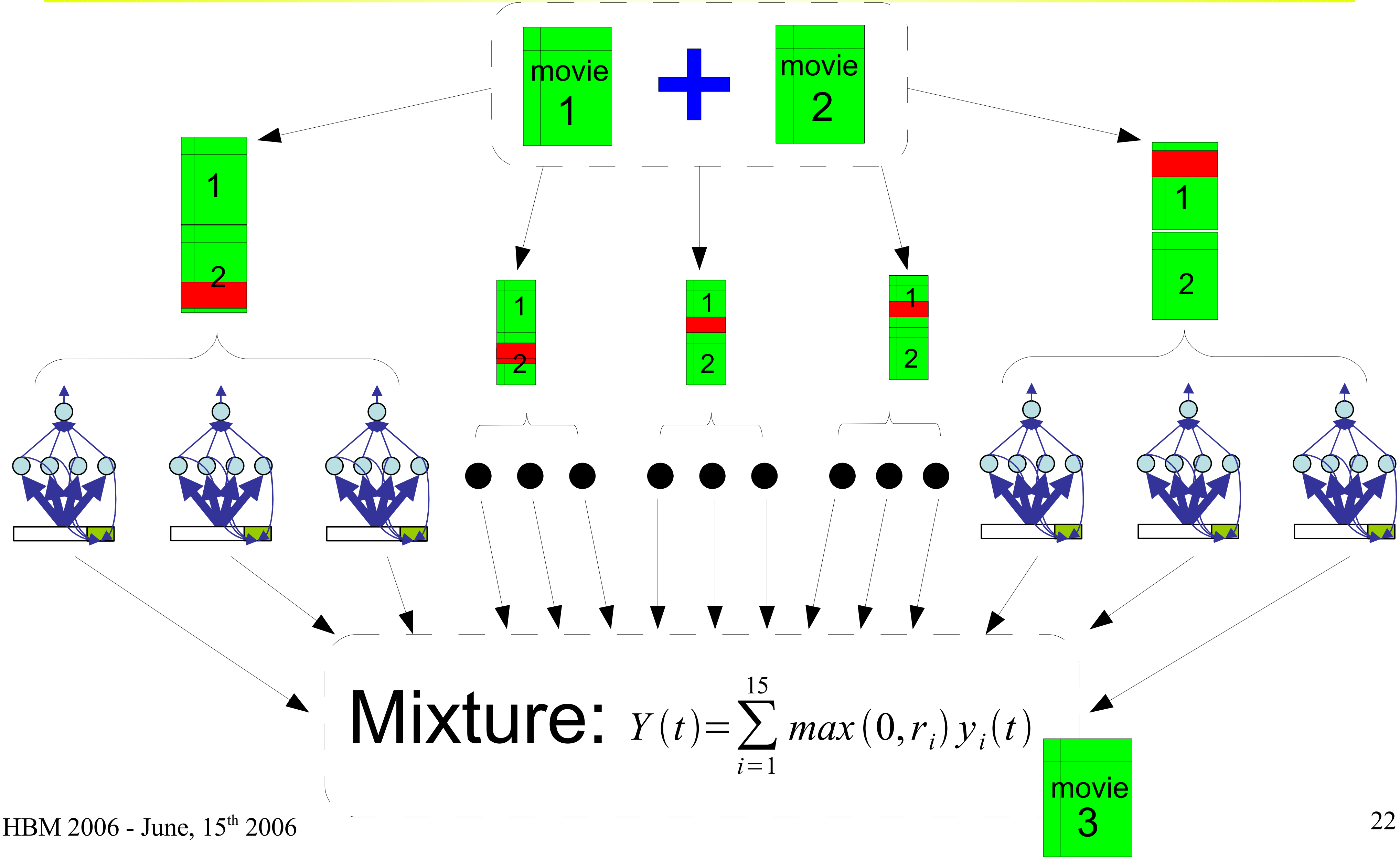
u_t = input vector x_{t-1} = internal state at (t-1)

- 200 input units
- 1 output unit
- 4 hidden units

$$x_t = g(w_u^T u_t + w_x^T x_{t-1}),\ x_0 = 0$$

- trained with Back-Propagation through time
- each input normalized separately

Prediction: RNN mixture



Mixture: $Y(t) = \sum_{i=1}^{15} max(0, r_i)\, y_i(t)$

Prediction: Observations / Insights

- *easy* features (Faces, Language, etc.): 1 or 2 hidden units are good
- *difficult* features (Food, Attention, etc.): accuracy (low) independent of number of hidden units:
  - Pre-processing removed important information?
  - Information not present in fMRI data?
- NOTE: we don't exploit correlation between features

Conclusions

- Proposed method very general:
  - No hand tuning for features
  - No hand tuning for subjects
- score>0.482 on base features using movie 1 and 2 (train set) to predict movie 3
- Pre-processing more important than prediction
- Averaging predictions over subjects is more accurate (objectivity vs. subjectivity ?)
- Our 3rd (best) submission was only on base features because lack of CPU time

Future Work

- Exploit:
  - volumes' autocorrelation during pre-processing
  - objectivity/subjectivity *per feature*
  - features inter-dependence
  - linguistic context (features' smooth evolution)
  - morphological context (volumes' smooth evolution)
  - past and future (*contextual RNN*)
- Explore different spatial/temporal trade-off during clustering

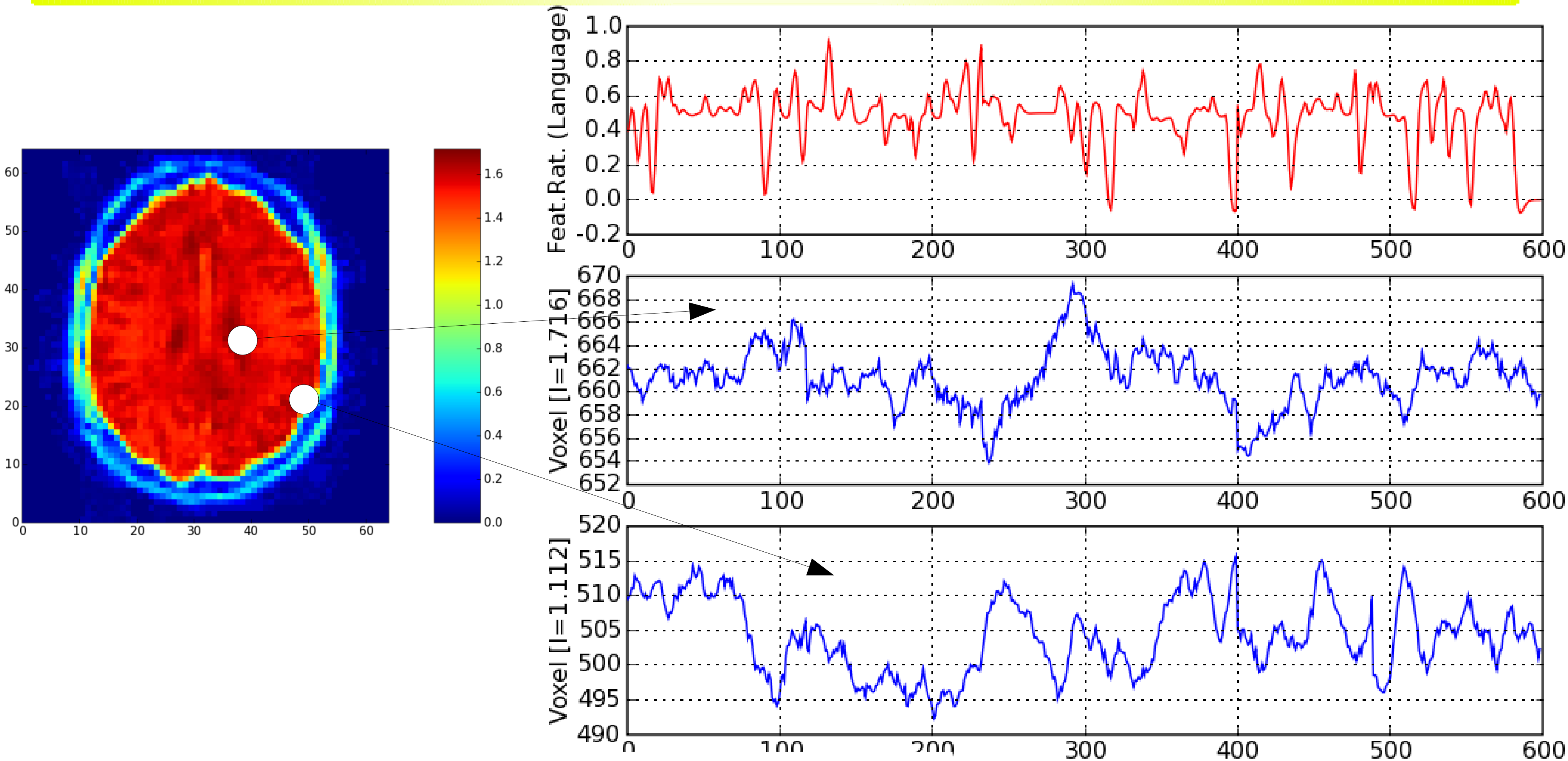
intereseted in details? Come to poster 677

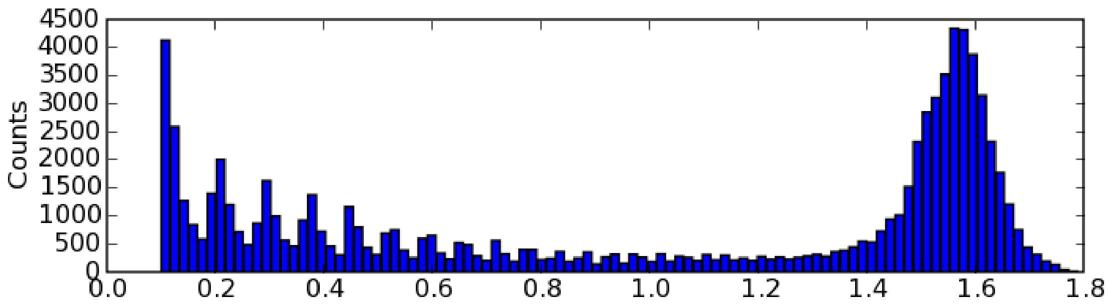
We'd like to thank...

- Scientific/Numerical Python project http://www.scipy.org, http://www.numpy.org for their amazing numerical libraries
- M.de Hoon, S.Imoto, S.Miyano, Laboratory of DNA Information Analysis, Human Genome Center – University of Tokyo, for **pycluster** library
- Francisco Pereira, for his post on “Machine Learning (Theory)” blog: http://hunch.net/?p=166

Some extra slides

Mutual Information (continued)





Training Process: RNN mixture

Given a subject and a feature:

- join movie 1 and movie 2, remove blanks
- 5-fold cross validation
- train 3 times each of the five models with different random initialization of parameters,
- among the 15 resulting models remove those having negative correlation on the validation set
- evaluate remaining models on movie 3
- average all outputs according to the correlations on validation set (weighted average)