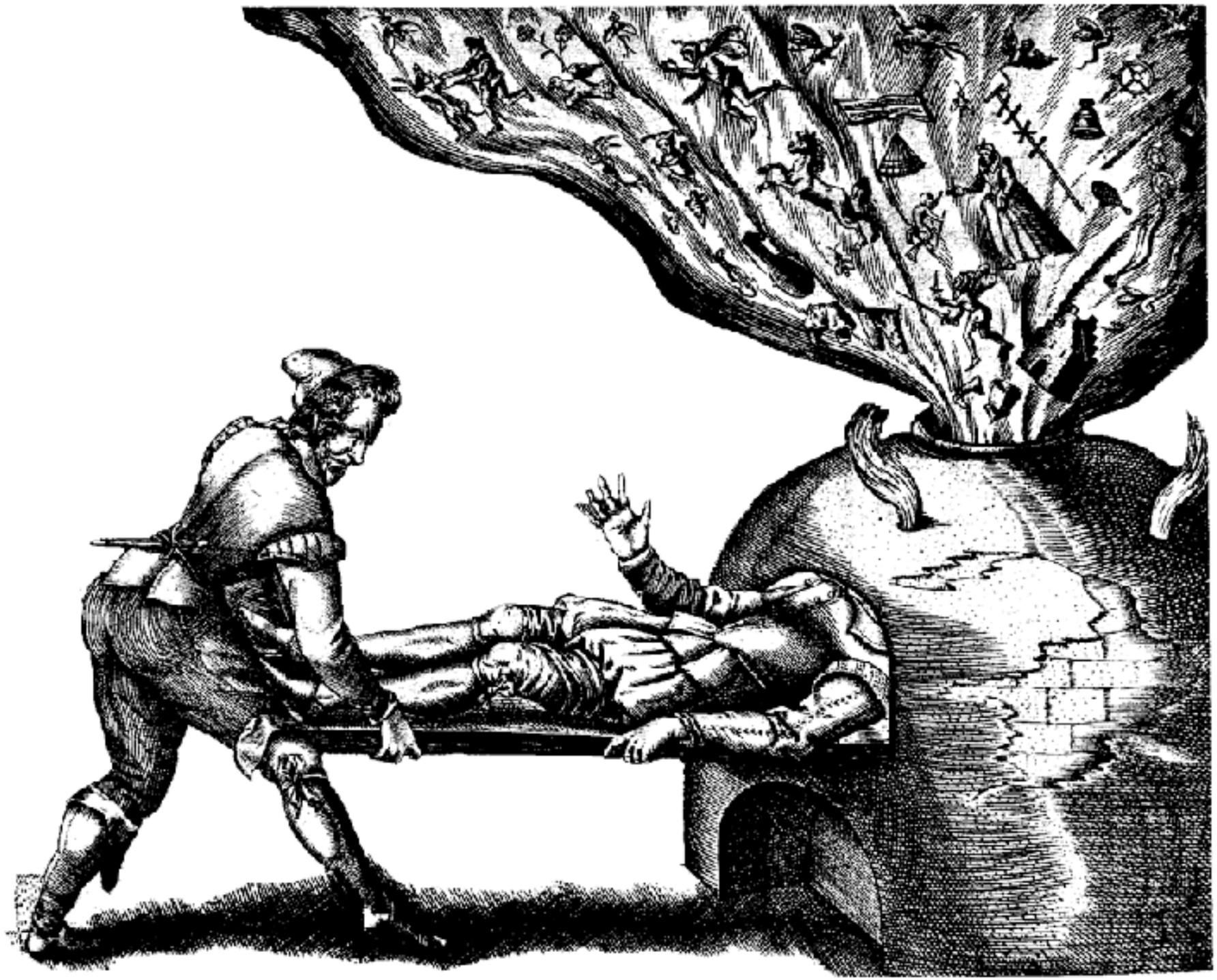
# Statistical Machine Learning and Optimisation Challenges for Brain Imaging at a Millisecond Timescale

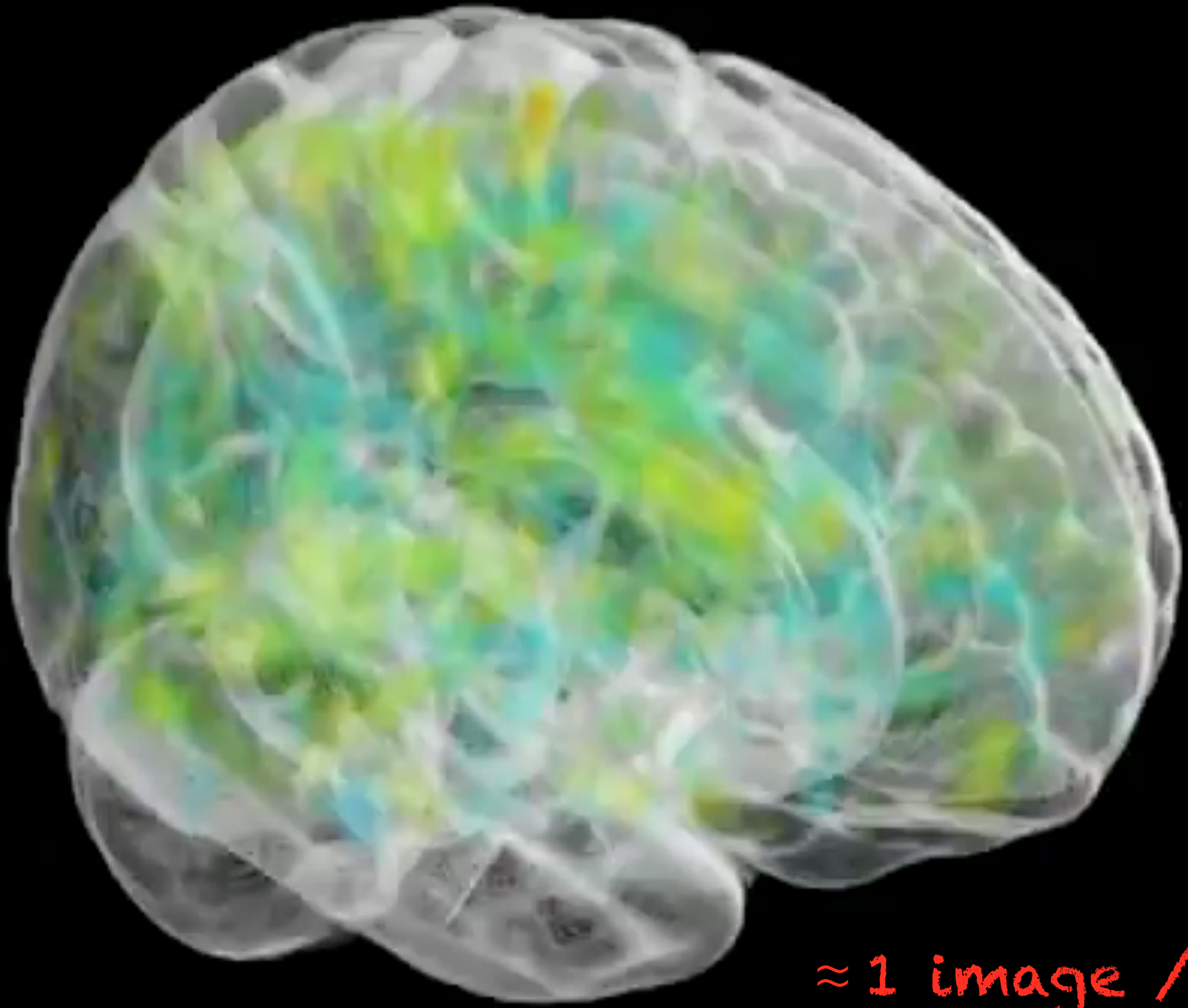
Alexandre Gramfort
http://alexandre.gramfort.net

Inria, Parietal Team

*Cambridge - Sept. 2017*

# What is functional brain imaging?

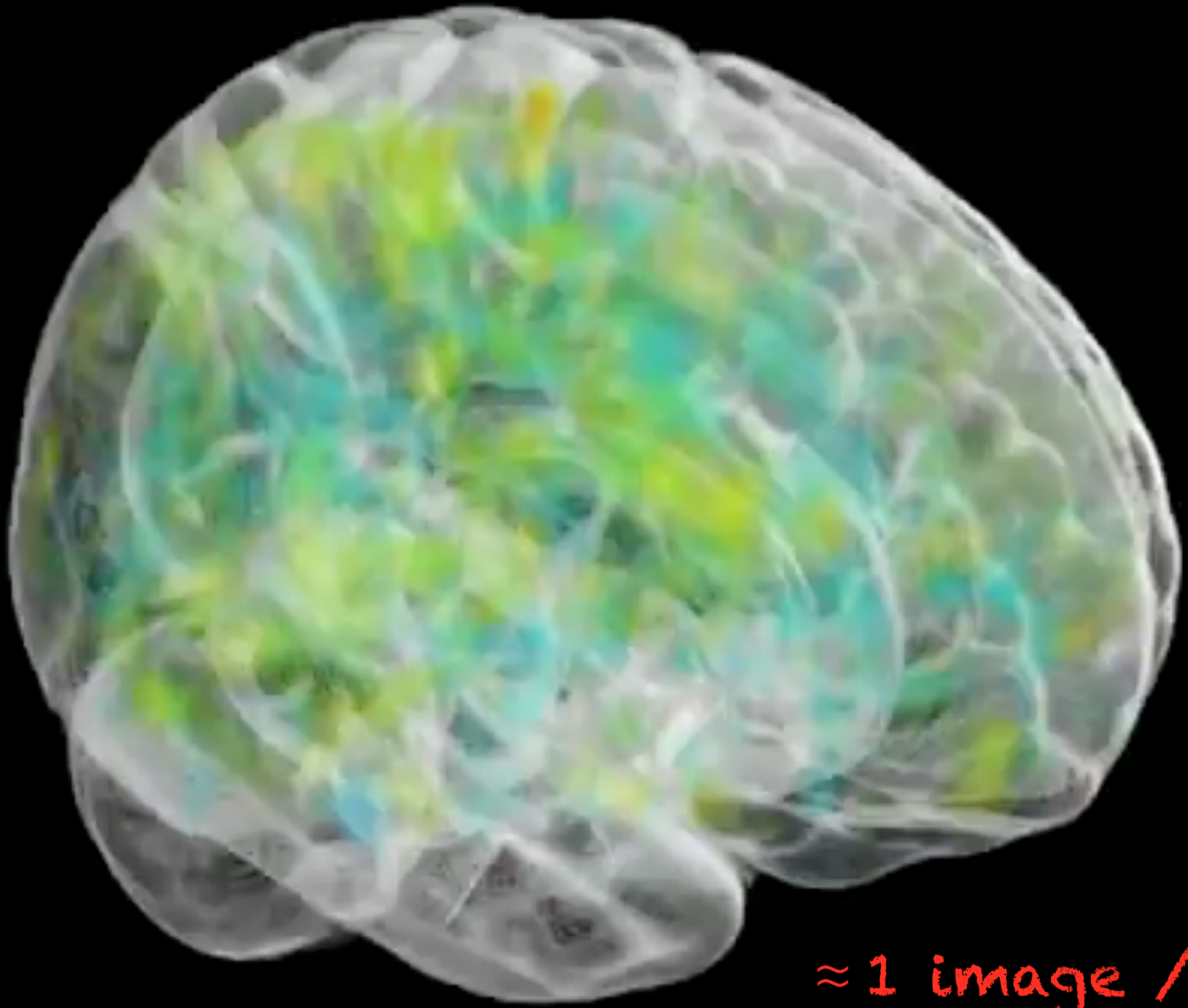




≈ 1 image / 2s

http://www.youtube.com/watch?v=uhCF-zlk0jY

courtesy of Gael Varoquaux

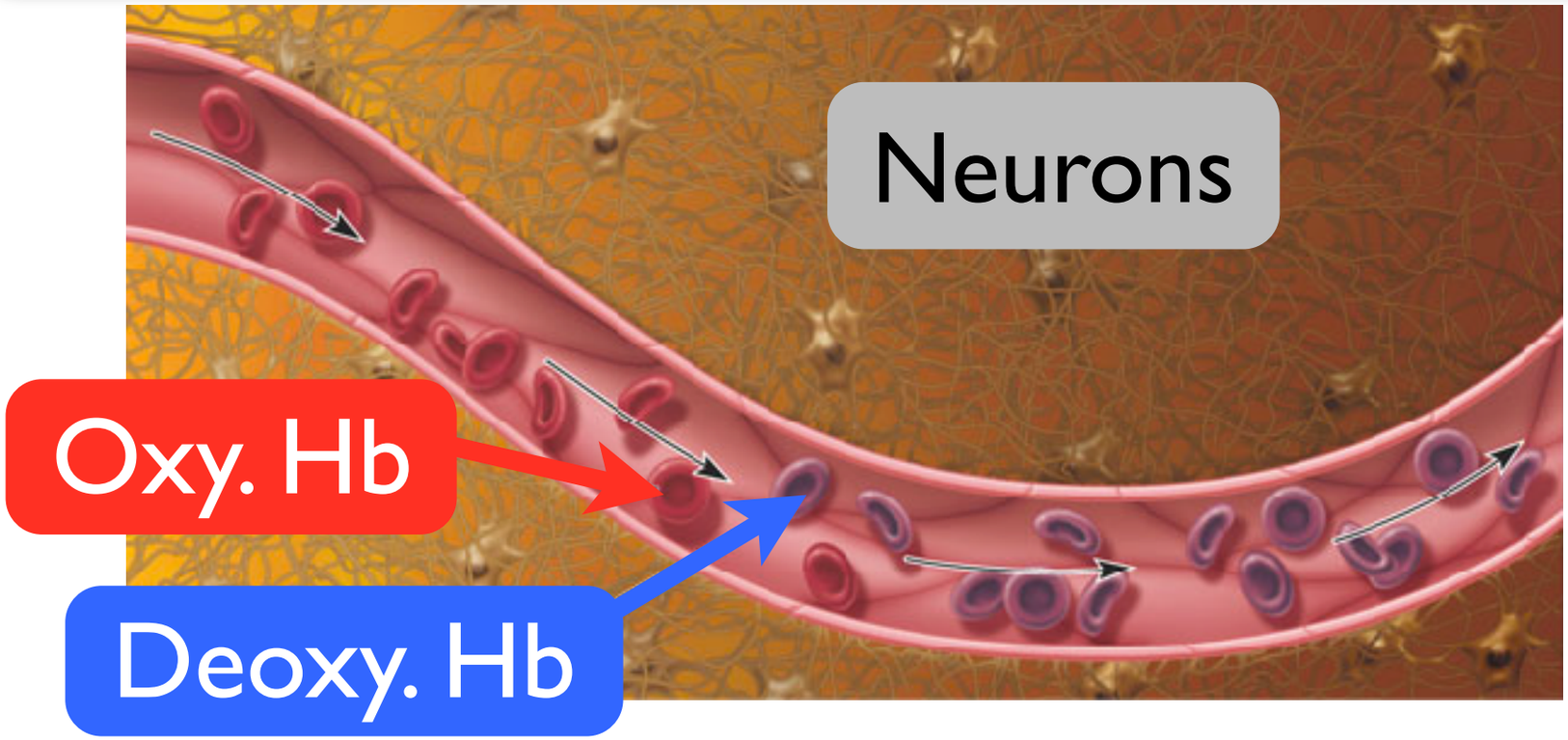


≈ 1 image / 2s

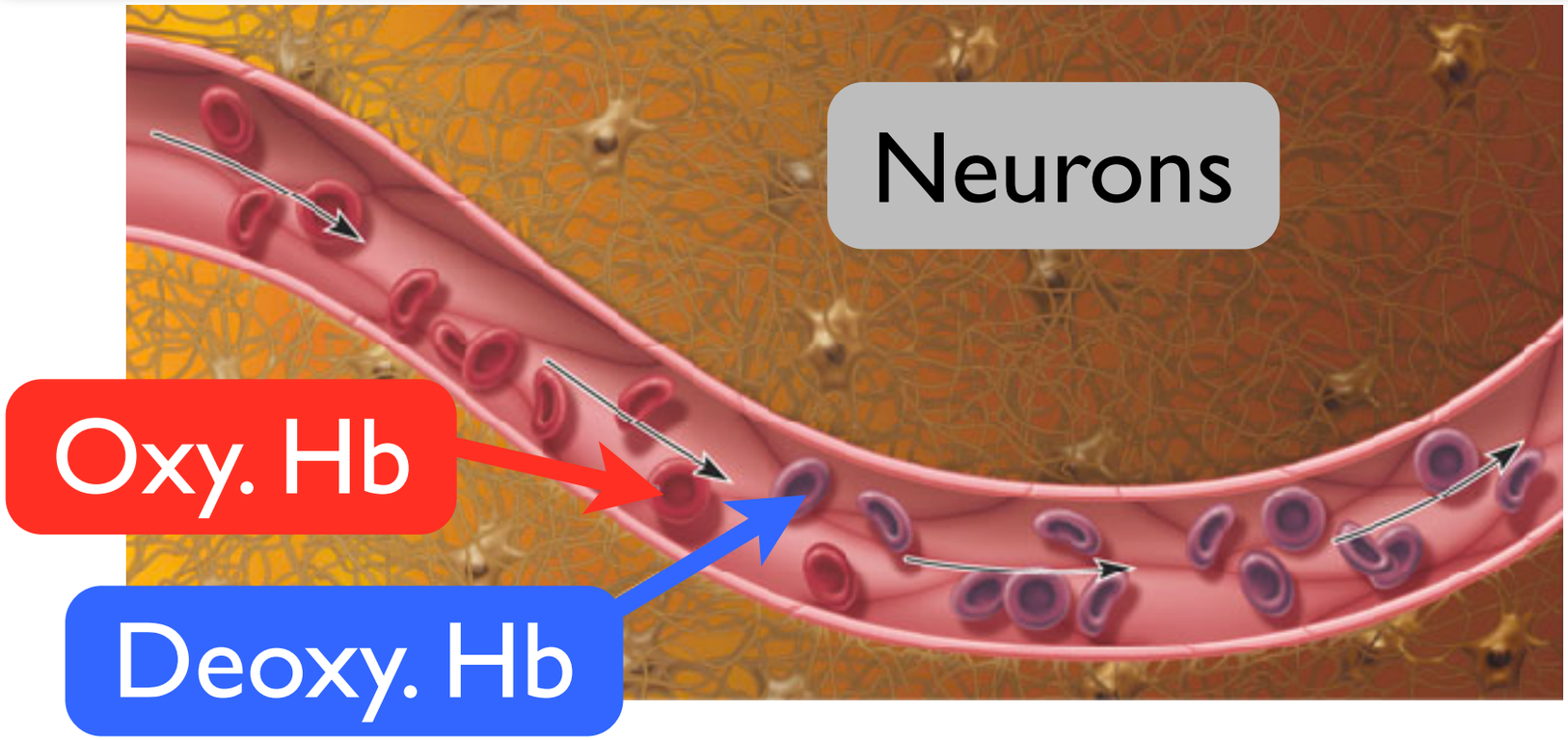
http://www.youtube.com/watch?v=uhCF-zlk0jY

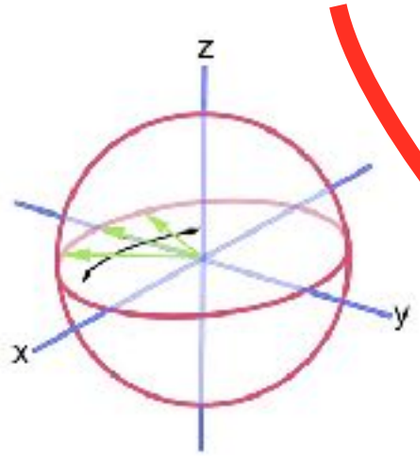
courtesy of Gael Varoquaux

# Functional MRI (fMRI)



# Functional MRI (fMRI)

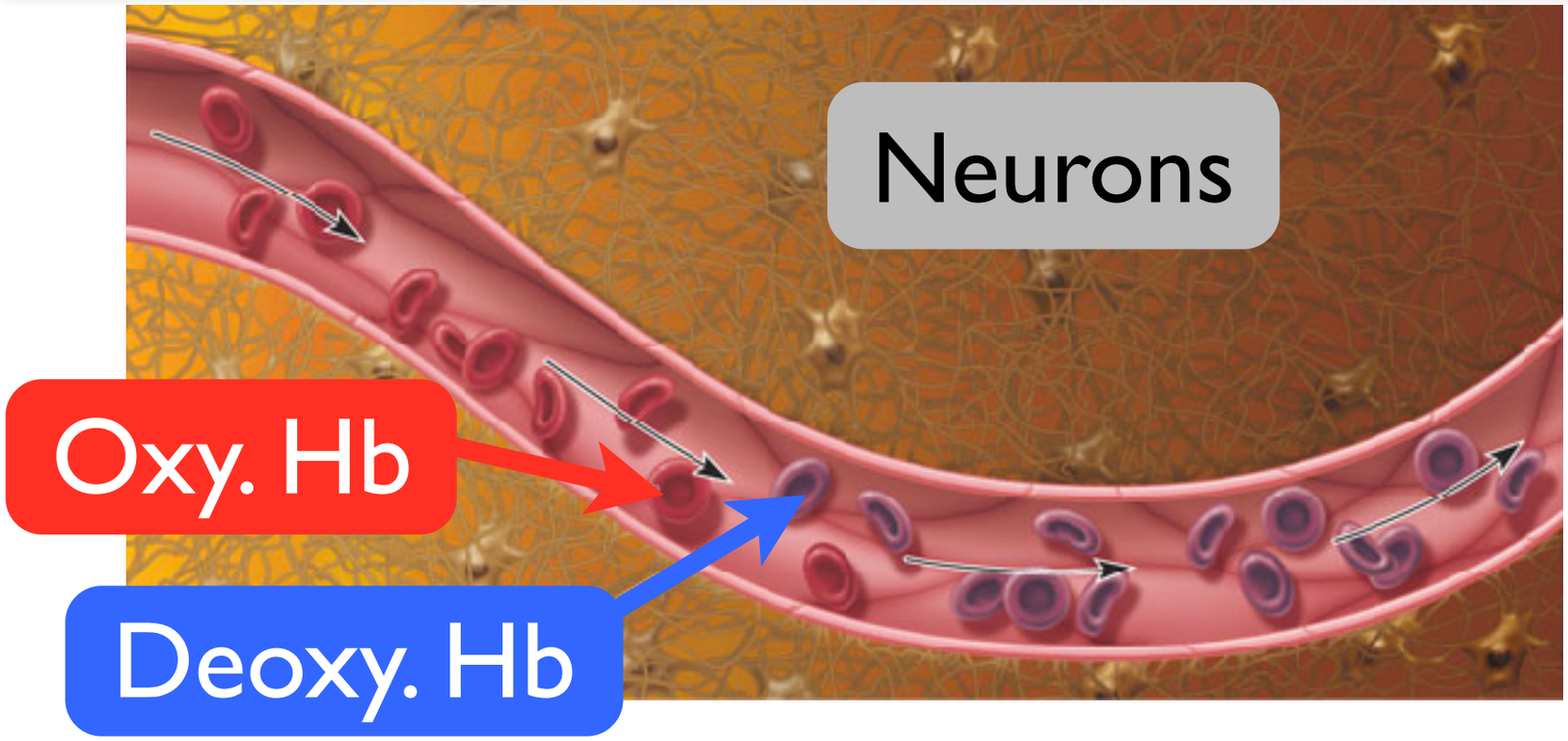


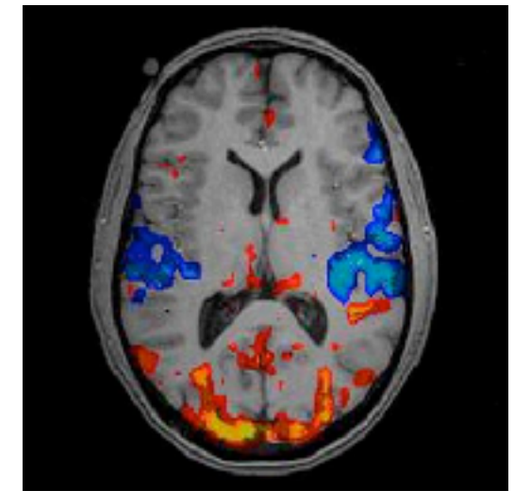



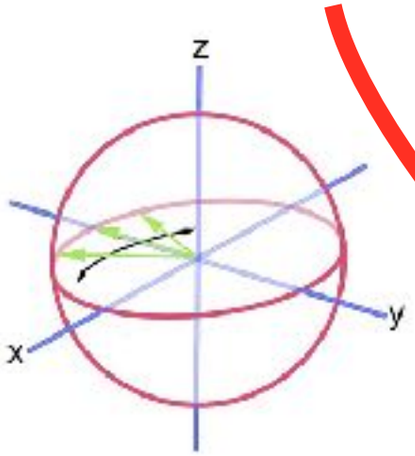

Magnetic resonance imaging

Scanner

# Functional MRI (fMRI)

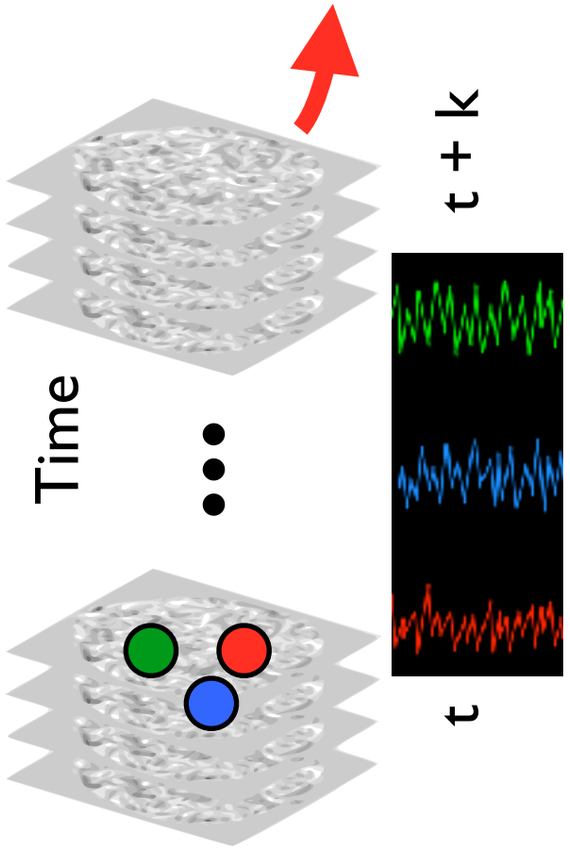


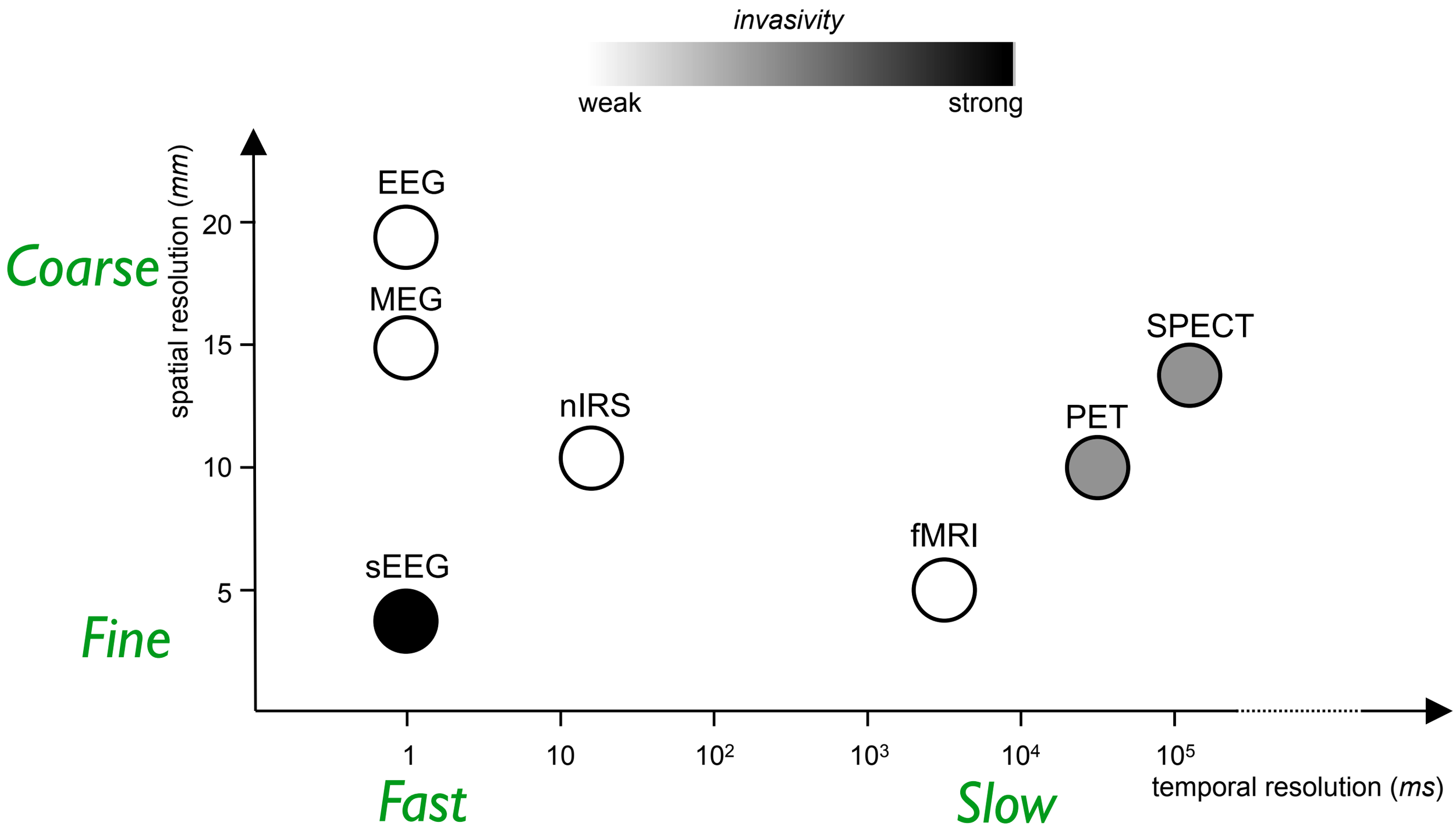



Magnetic resonance imaging

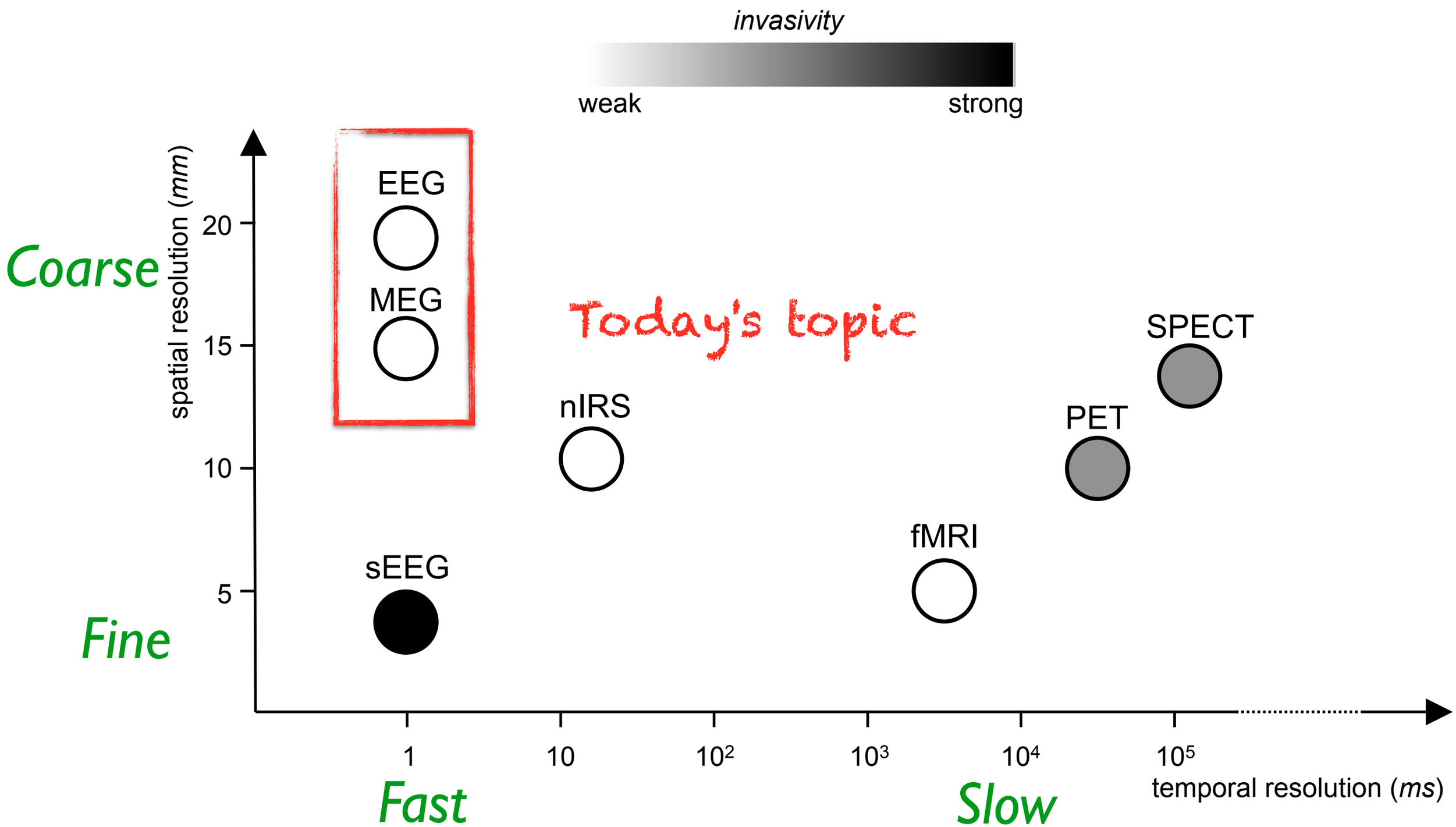
Scanner



# What is functional brain imaging?



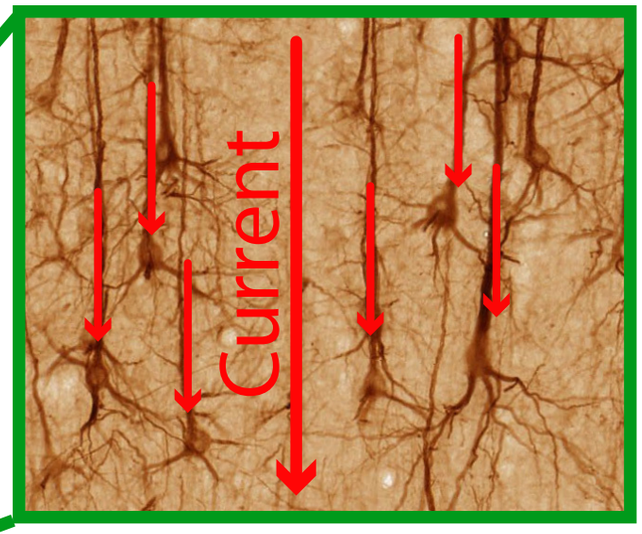
# What is functional brain imaging?

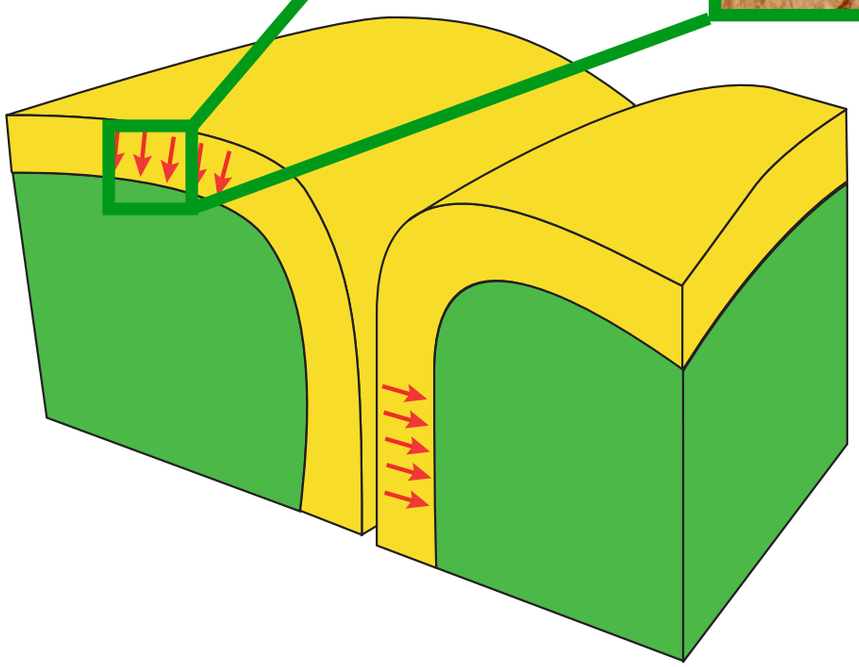


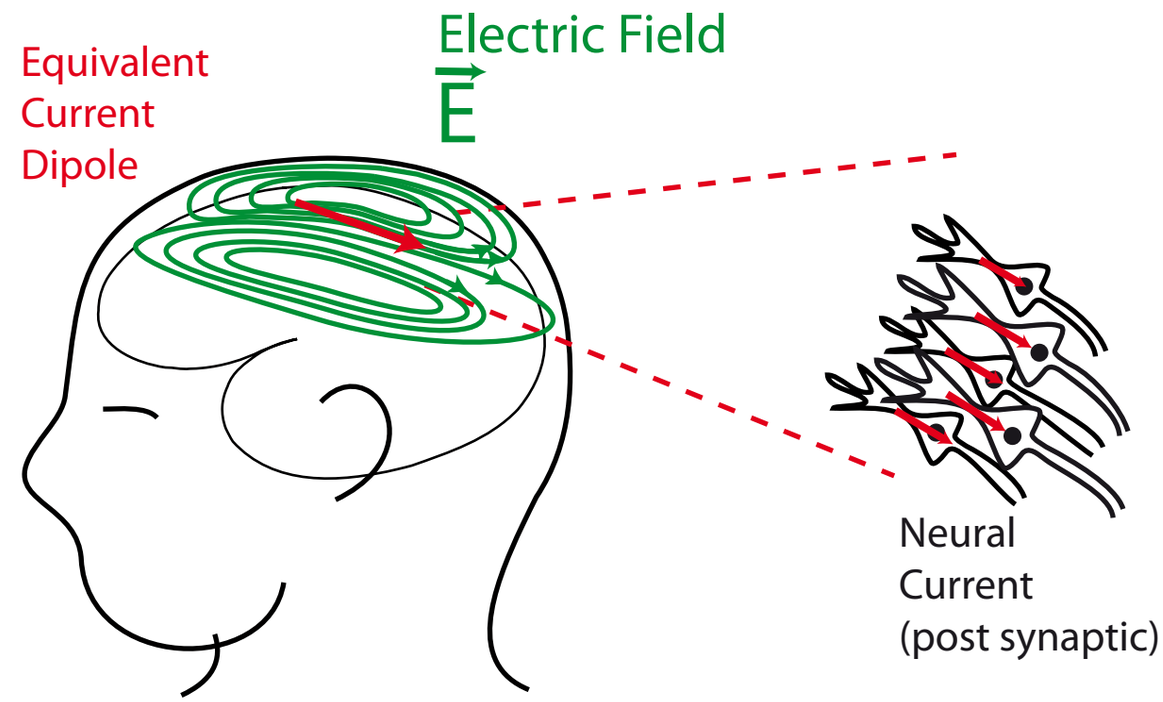
# Neurons as current generators

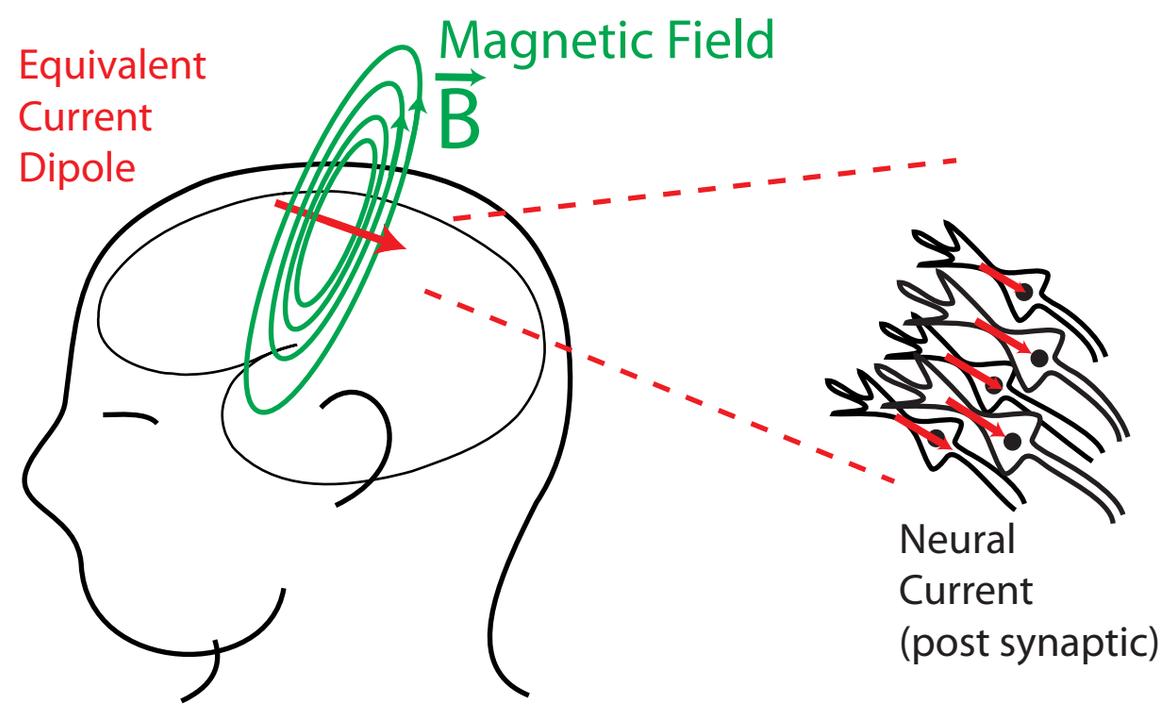
Large cortical pyramidal cells organized in macro-assemblies with their **dendrites normally oriented to the local cortical surface**

$Q = I \times d$
(10 to 100 nAm) with the equivalent current dipole (ECD) model

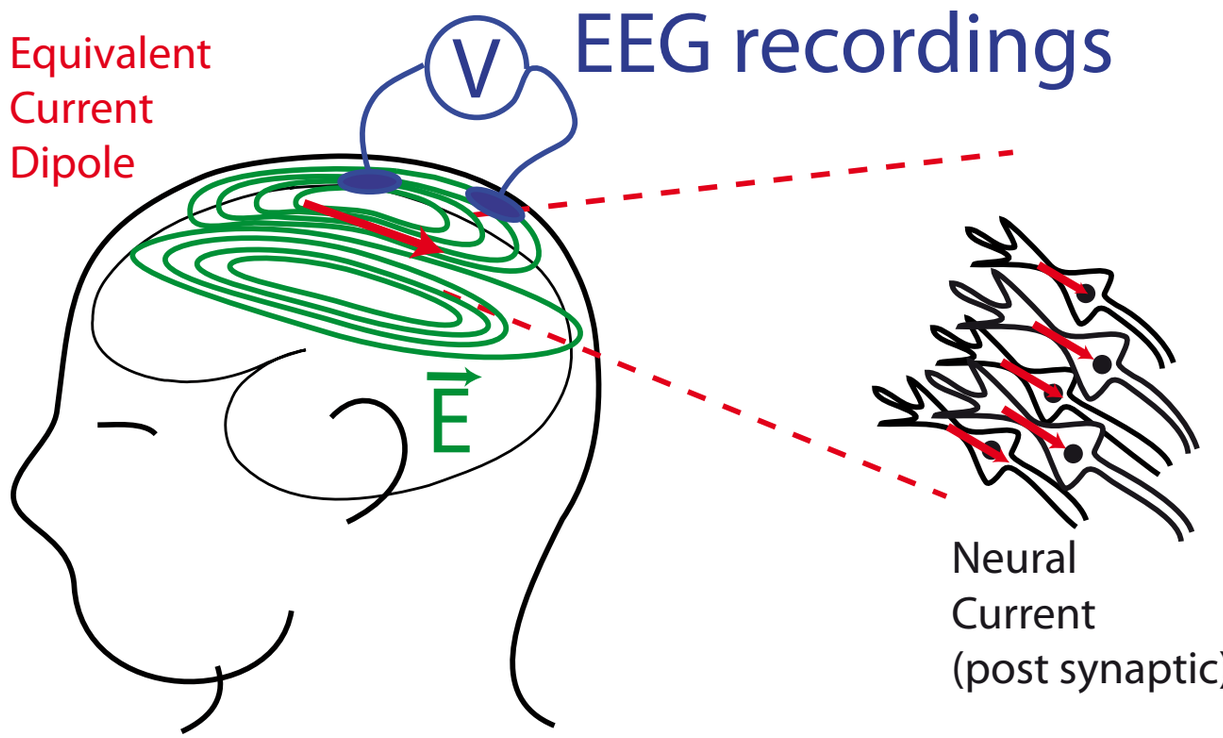


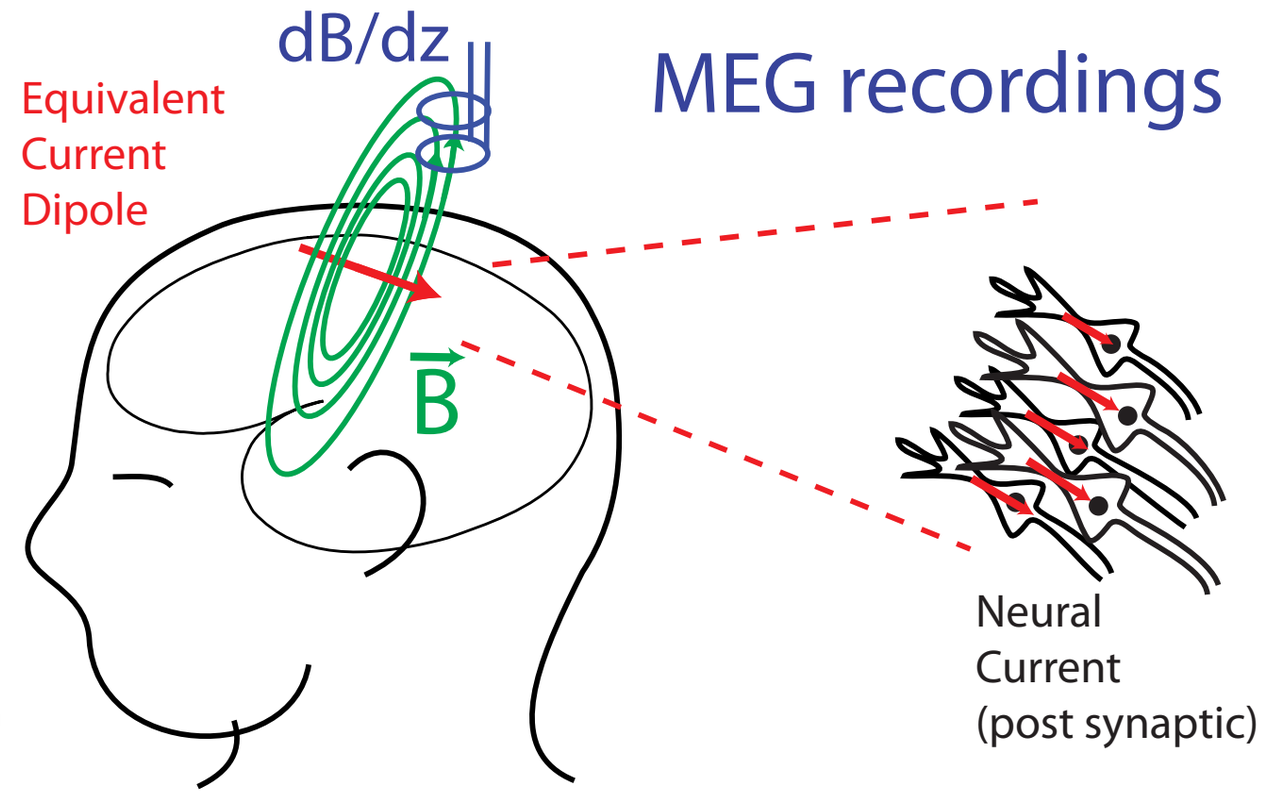



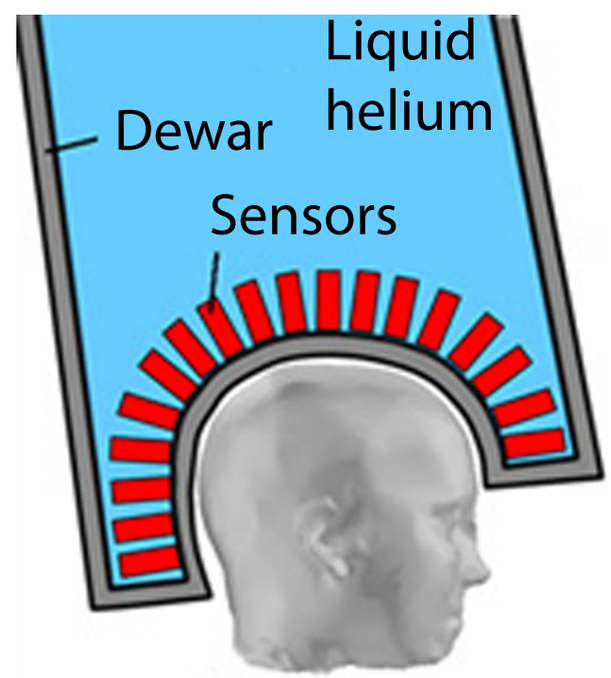



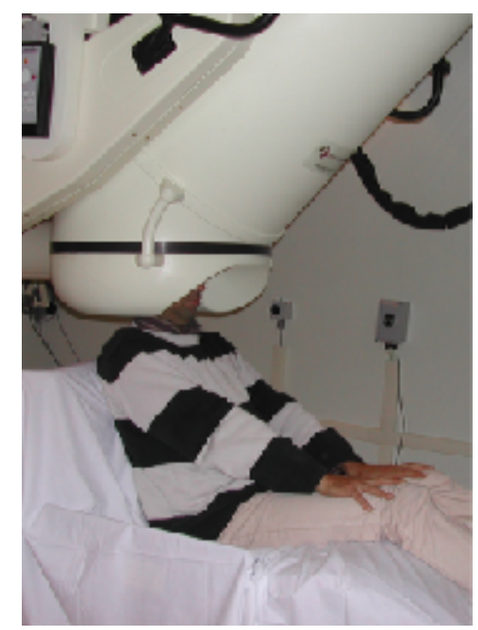
# Electro- & Magneto-encephalography



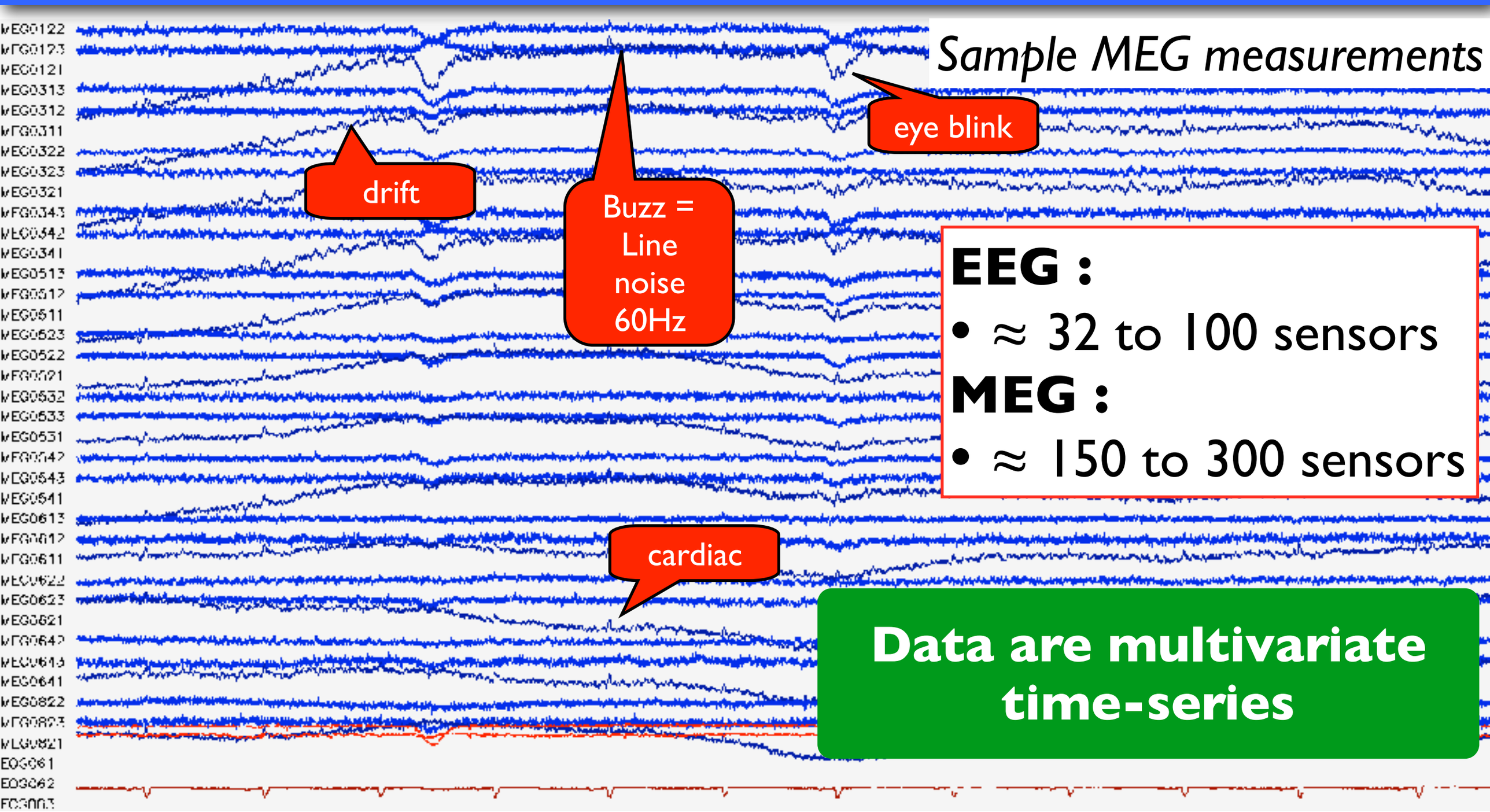

First EEG recordings in 1929 by H. Berger



Hôpital La Timone
Marseille, France

# M/EEG Measurements




**EEG :**

- ≈ 32 to 100 sensors

**MEG :**

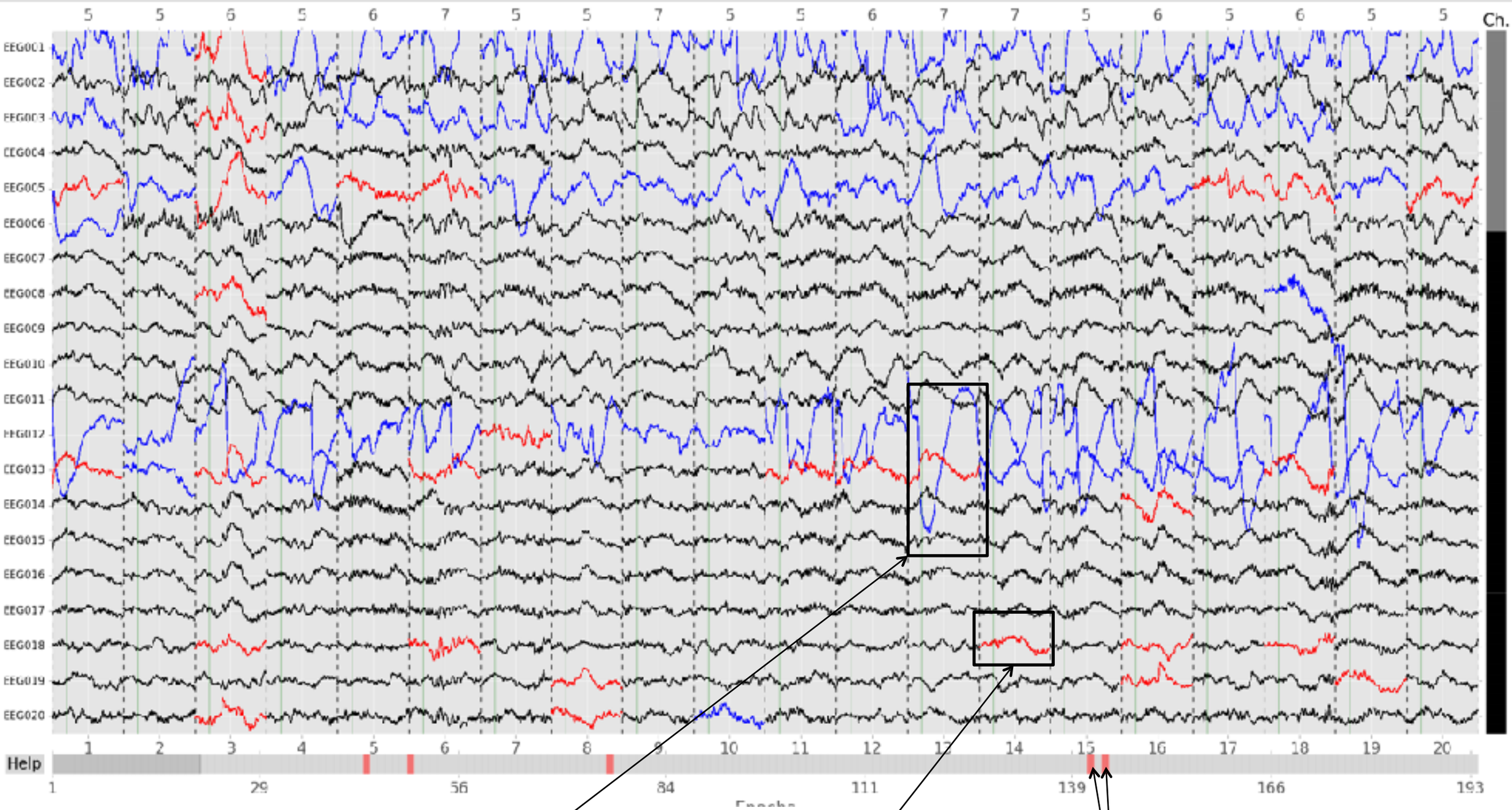
- ≈ 150 to 300 sensors

**Data are multivariate time-series**

Time frame: 10 seconds

≈ 1000 samples / s

# Artifacts everywhere & tedious to manually fix



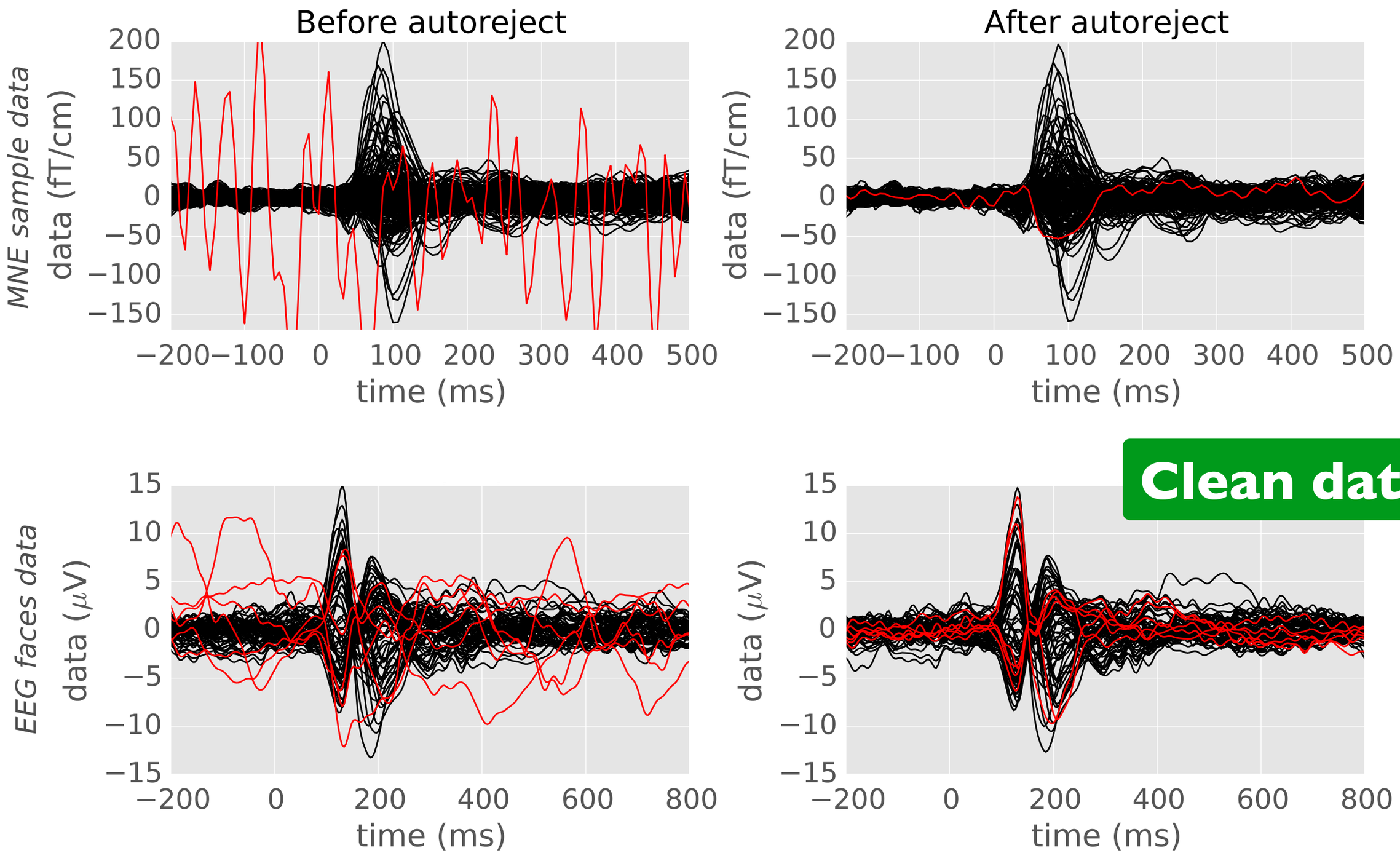

Sensor to be interpolated

Bad sensor but not going to be interpolated

Bad trials

http://autoreject.github.io/auto_examples/plot_visualize_bad_epochs.html

# Automatic artifact cleaning tool

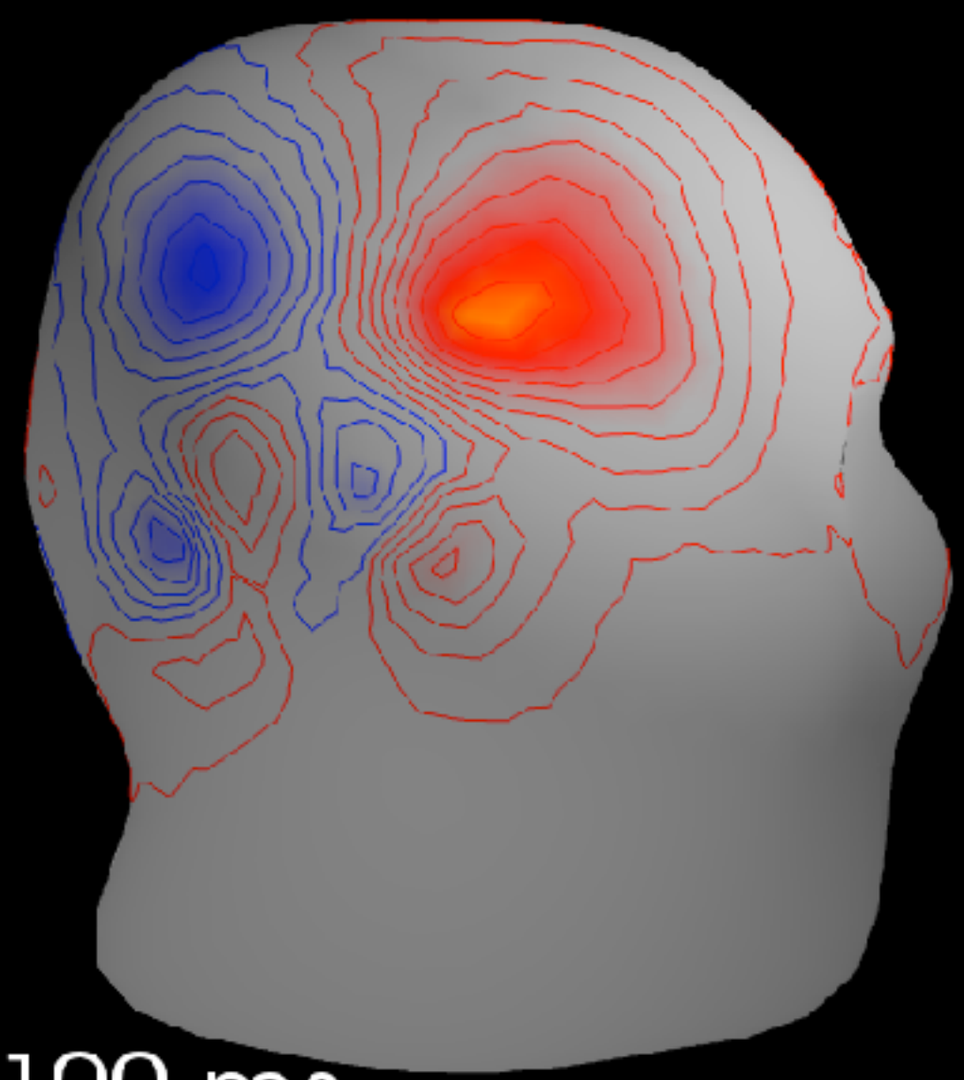



[M Jas, D Engemann, Y Bekhti, F Raimondo, and A Gramfort, "Autoreject: Automated artifact rejection for MEG and EEG.", NeuroImage 2017]

# Imaging the brain at a millisecond time scale with MEG and EEG

## and stats and optimization

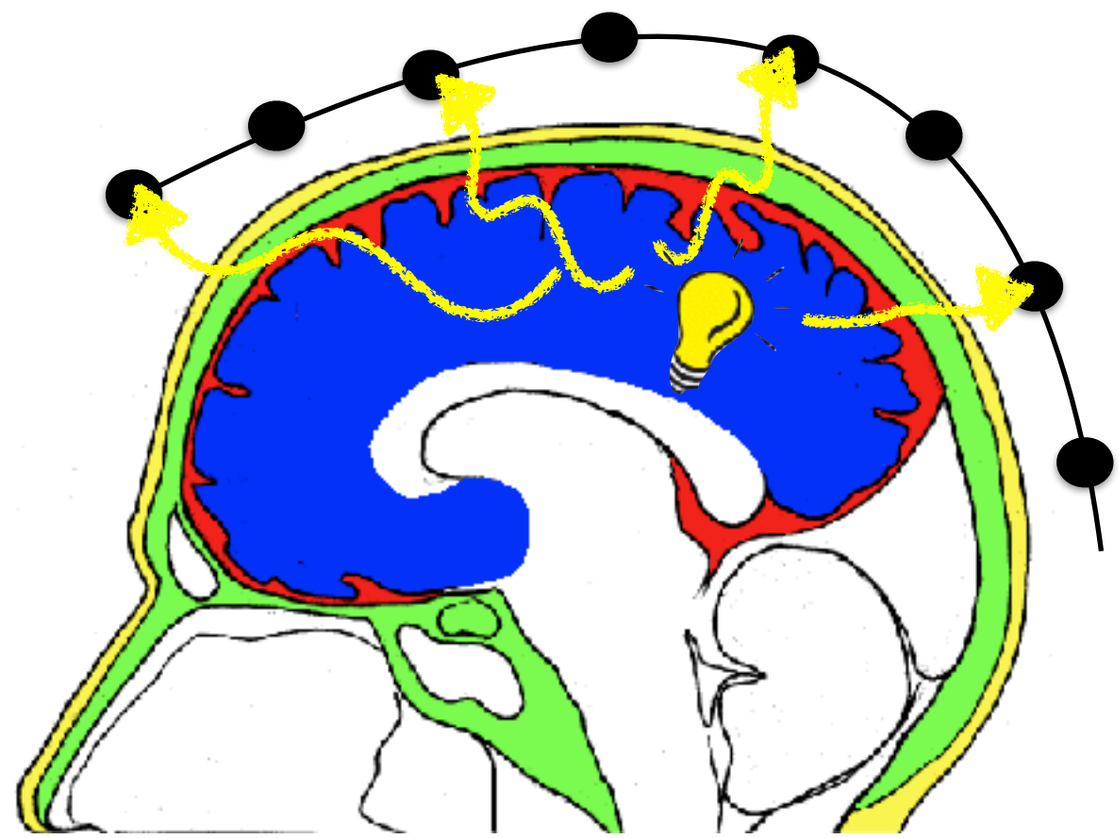
**Find the current generators** that produced the MEG measurements



t = 100 ms

# What do we measure?

$$\nabla \times \vec{B} = \mu_0 \vec{J}$$

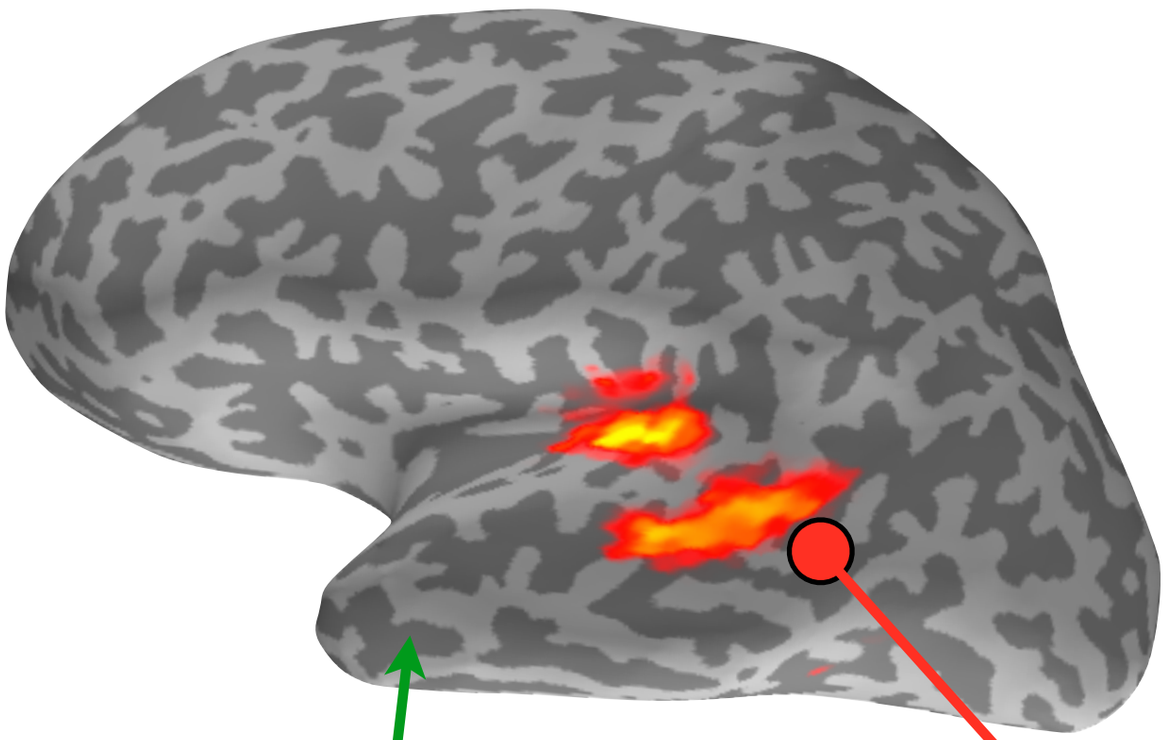


Boundary element method (BEM),
i.e., numerical solver with
approximate solution.

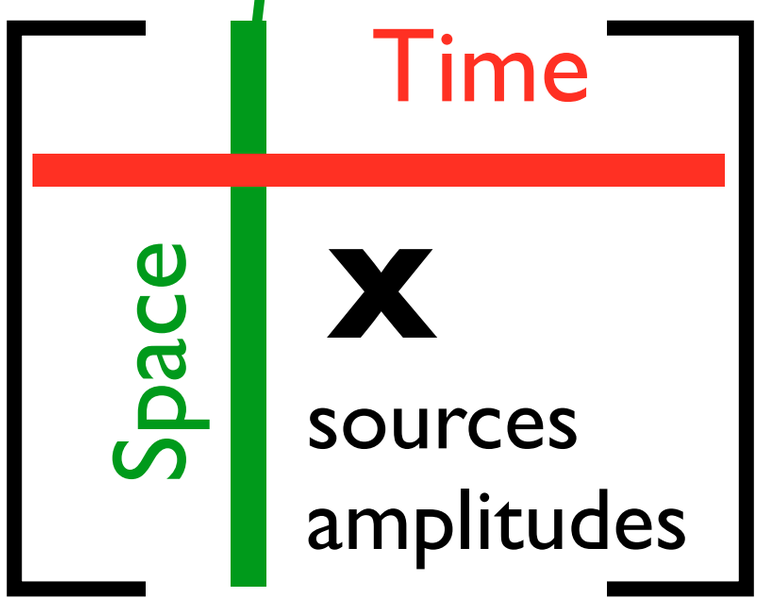
Linear PDE -> Linear forward problem / Fixed design

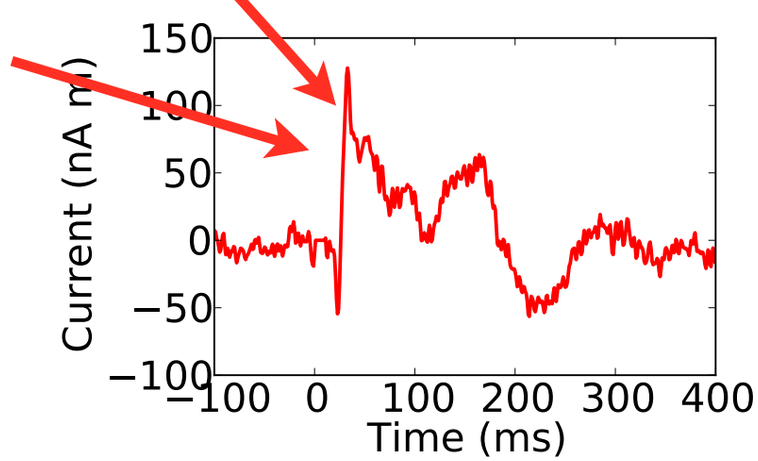
[Geselowitz 67, De Munck 92, Kybic et al. 2005, Gramfort et al. 2010 ]

# The source model



Position 5000 candidate sources over each hemisphere (e.g. every 5mm)



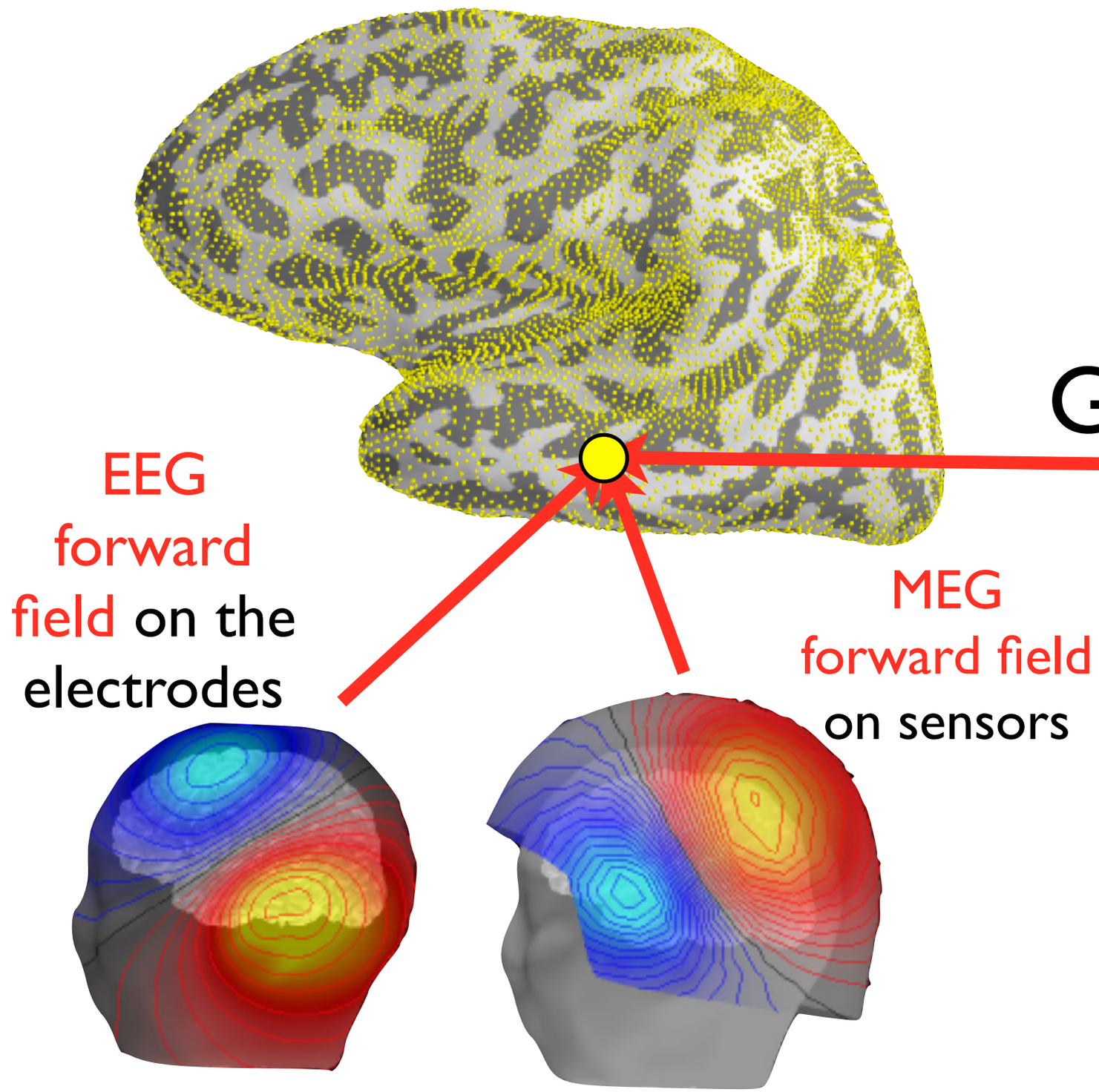




$$\mathbf{X} \in \mathbb{R}^{\mathbf{P} \times \mathbf{T}}$$

*Scalar field defined over time*

[Dale and Sereno 93]

# The source model

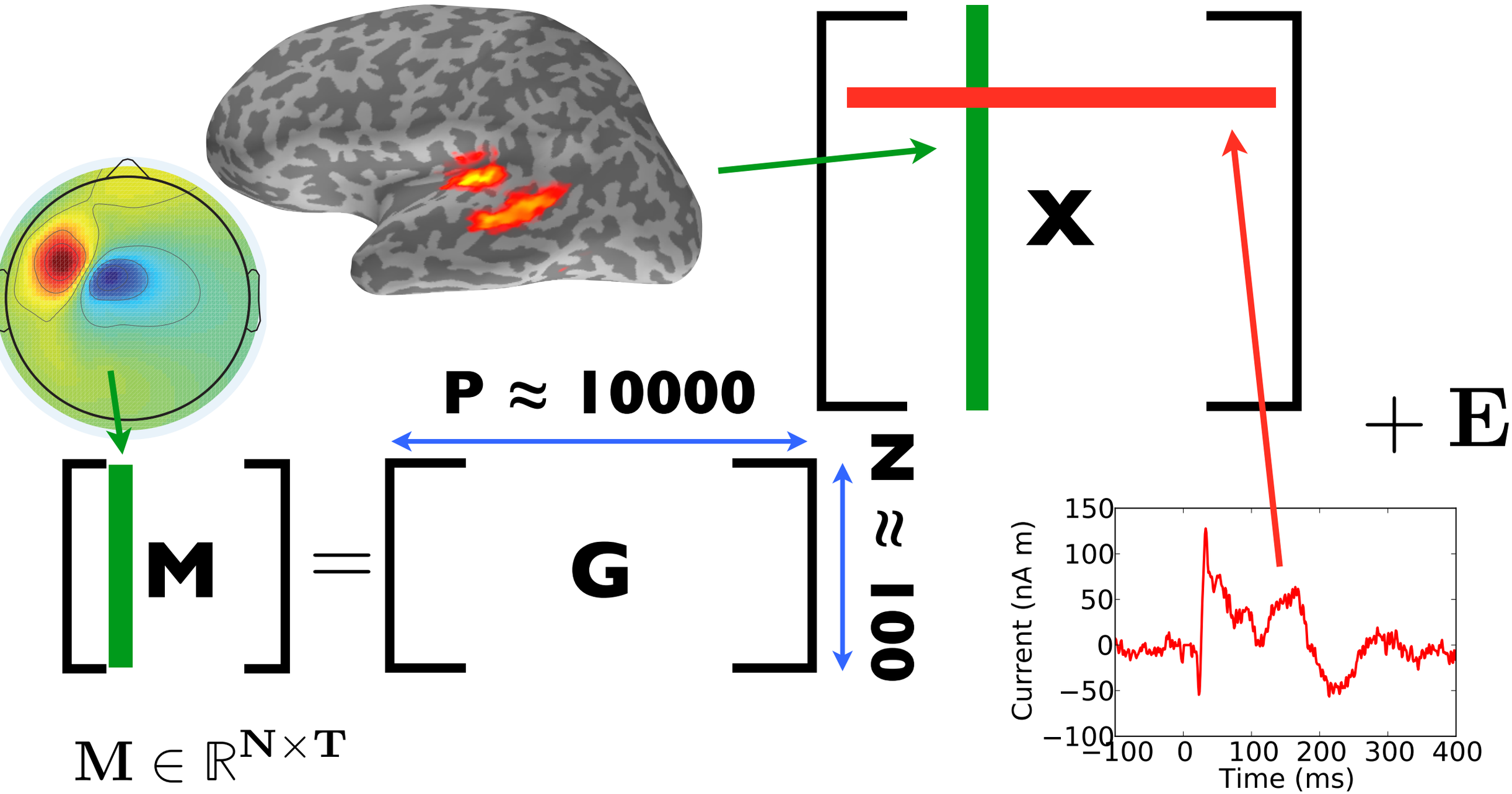



one column = Forward field of one dipole

$$G = \left[\begin{array}{c} G_{EEG} \\ \hline G_{MEG} \end{array}\right] \in \mathbb{R}^{\mathbf{N} \times \mathbf{P}}$$

G is the gain matrix / forward operator obtained by concatenation of the forward fields

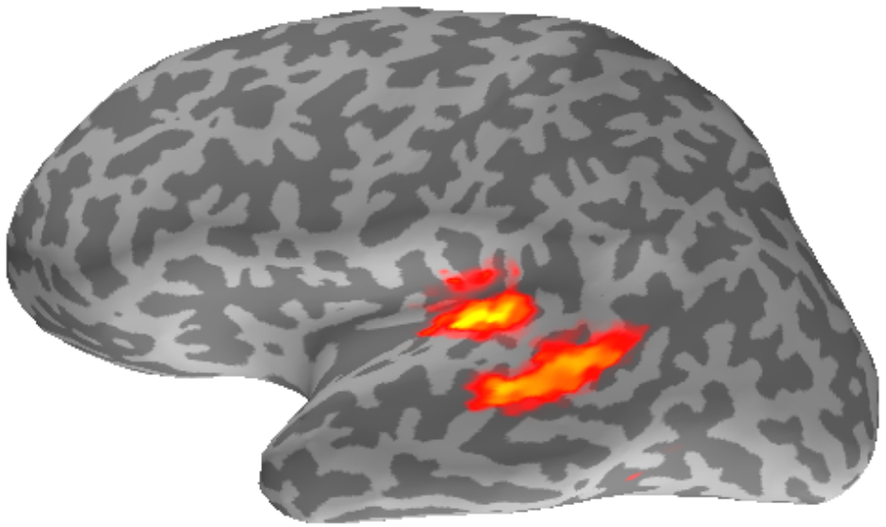
# M = GX+E : An ill-posed problem




$\mathbf{M} \in \mathbb{R}^{\mathbf{N} \times \mathbf{T}}$

**Small "N" large "P" problem**

# $y_t = X\beta_t + e_t$ : An ill-posed problem



Standard statistics notations

$\boldsymbol{\beta_t} \in \mathbb{R}^{\mathbf{P}}$

regression coefficients

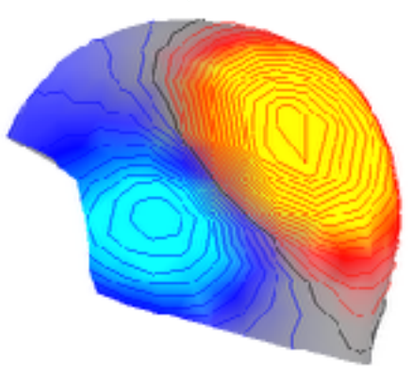
**P ≈ 10000**

$$\mathbf{y_t} = \mathbf{X} \quad \mathbf{N \approx 100}$$

$\in \mathbb{R}^{\mathbf{N}}$ (design matrix)

$+ \mathbf{e_t}$

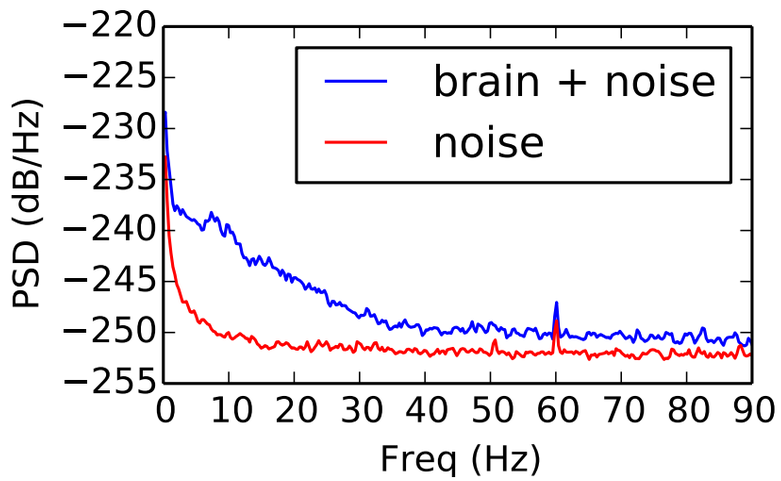
$\in \mathbb{R}^{\mathbf{N}}$



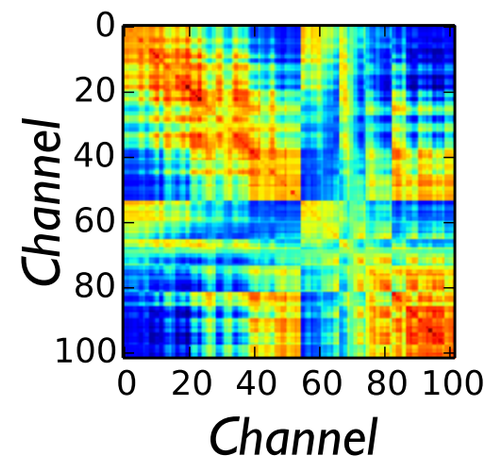
At **each time instant** the M/EEG inverse problem **IS** a **regression** with more variables than observations

# Why is it challenging?

- Complex signal dynamics: oscillations and transients
- Complex noise structure: colored and heteroscedastic



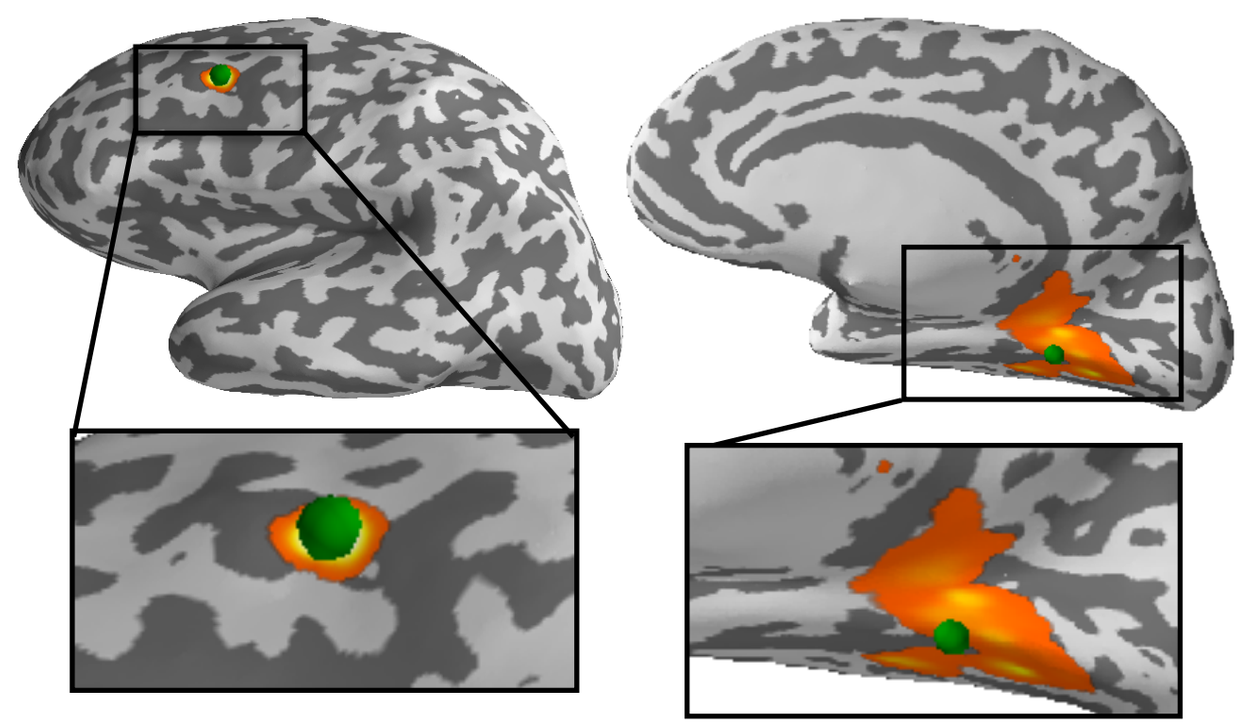

*Spectrum*




*Spatial correlation*

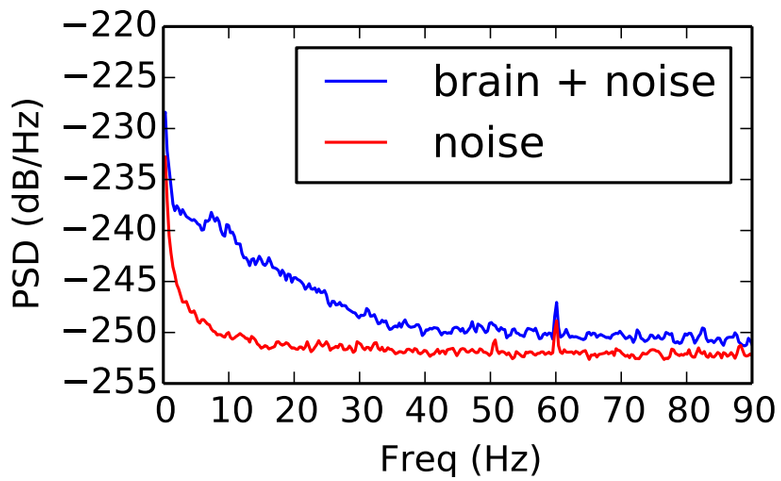
*[Engemann & Gramfort, Neuroimage 2015]*

- Device sensitivity varies with sensor types & source locations

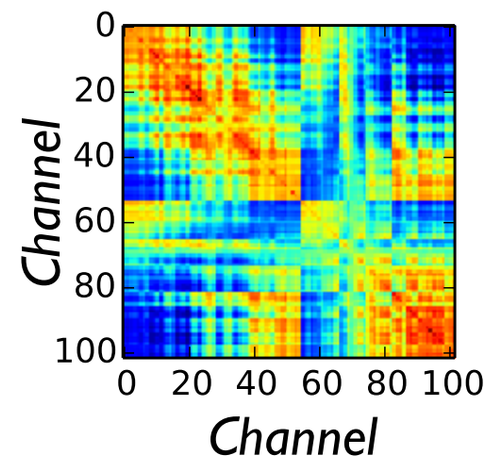


# Why is it challenging?

- Complex signal dynamics: oscillations and transients
- Complex noise structure: colored and heteroscedastic




*Spectrum*




*Spatial correlation*

*[Engemann & Gramfort, Neuroimage 2015]*

- Device sensitivity varies with sensor types & source locations

*What not giving up?*
*Clinical use (sleep, epilepsy, stroke, autism) & Cognitive Neurosience, Neuroengineering*

# Variational formulation

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \|\mathbf{M} - \mathbf{G}\mathbf{X}\|_F^2 + \lambda\phi(\mathbf{X}), \lambda > 0$$

Data fit Regularization

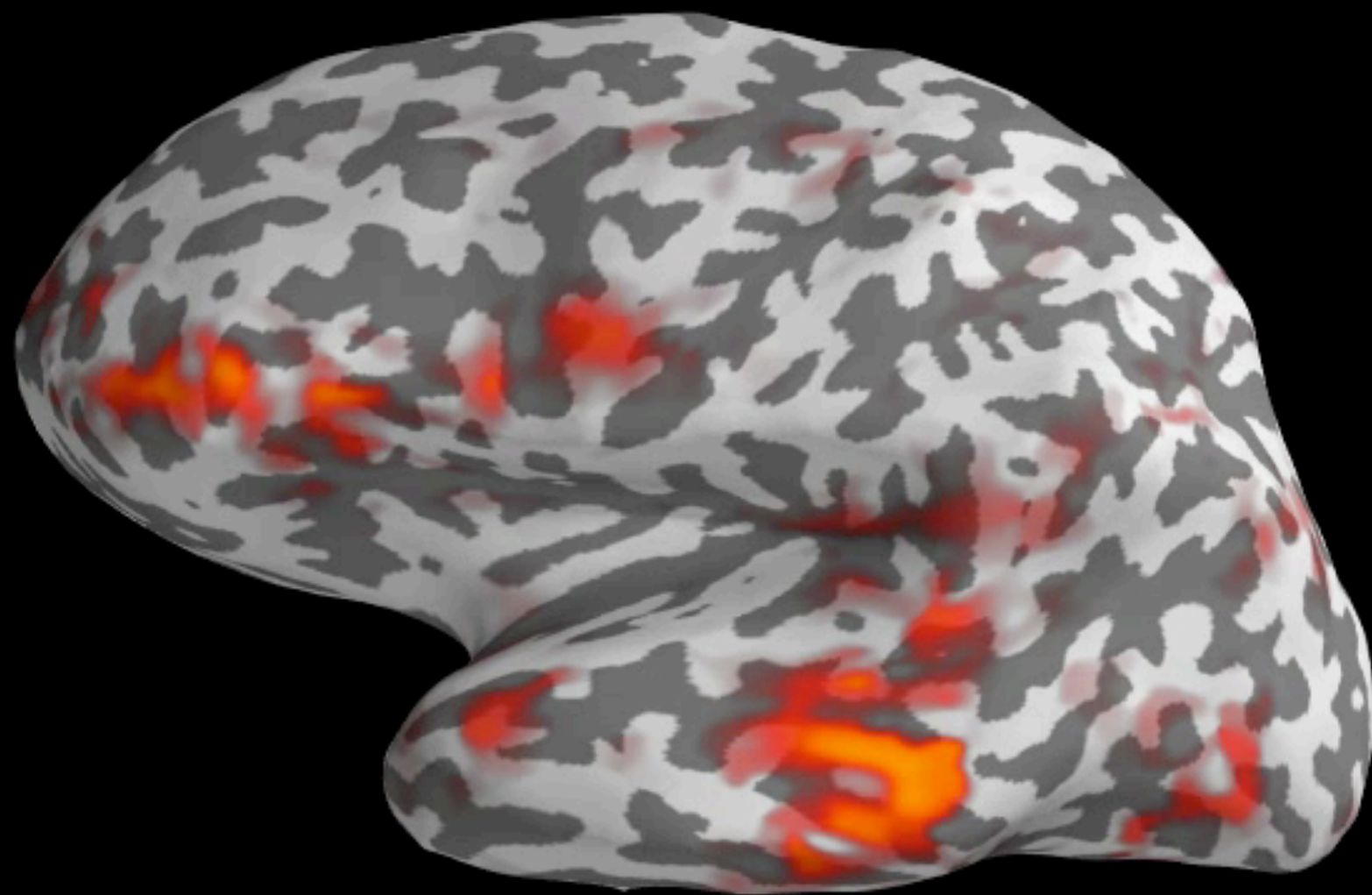
$\lambda$ : Trade-off between the data fit and the regularization

where $\|\mathbf{A}\|_F^2 = \mathbf{tr}(\mathbf{A}^T\mathbf{A})$

**Remark:** We assume here Gaussian i.i.d. homoscedastic noise. For heteroscedastic regression see e.g.

[Massias et al. Heteroscedastic Concomitant Lasso for sparse multimodal electromagnetic brain imaging, Arxiv]

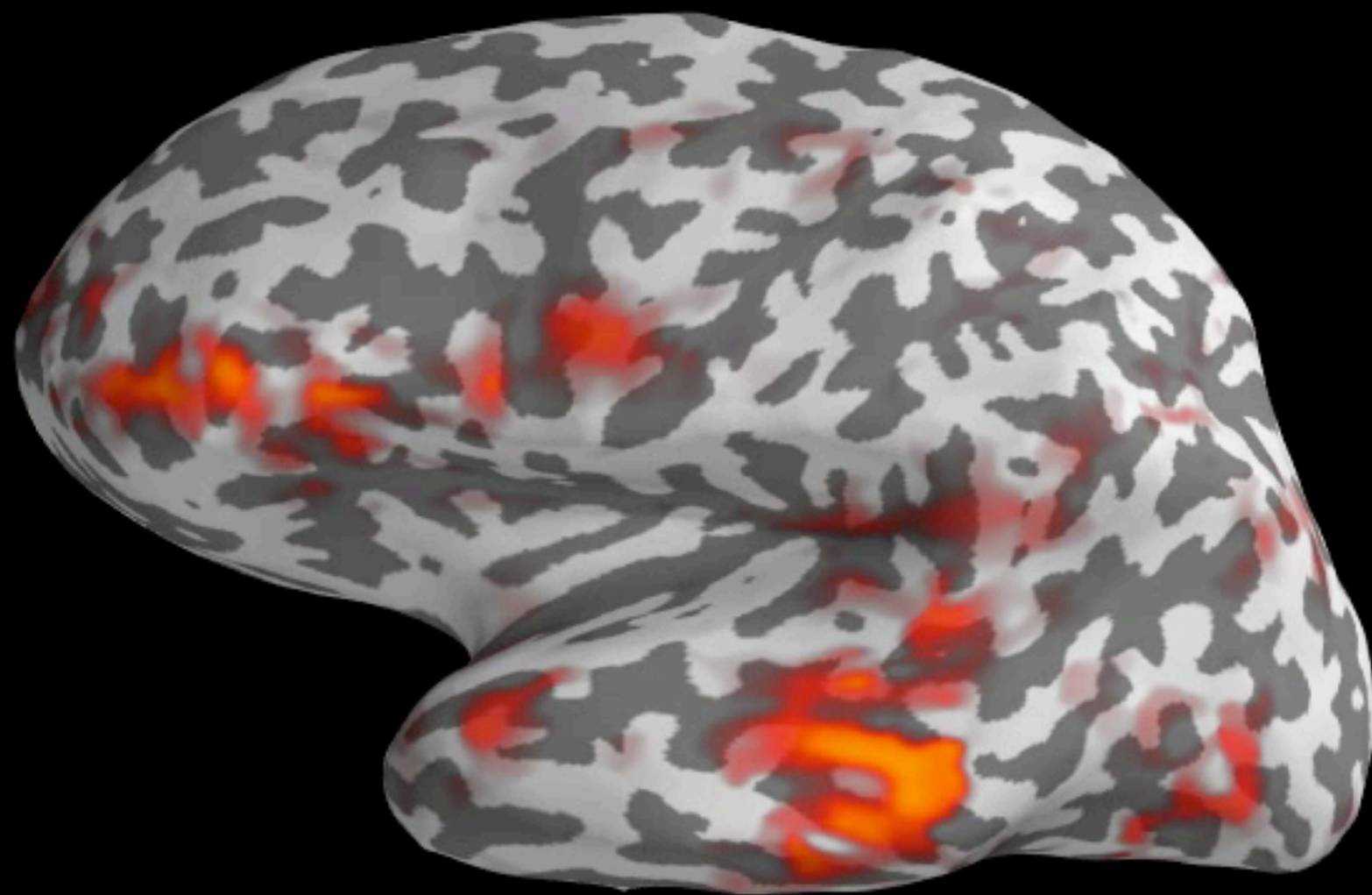
Result obtained with L2 regularization: $\phi(\mathbf{X}) = \|\mathbf{X}\|_F^2$



time=0.00 ms

http://youtu.be/Uxr5Pz7JPrs

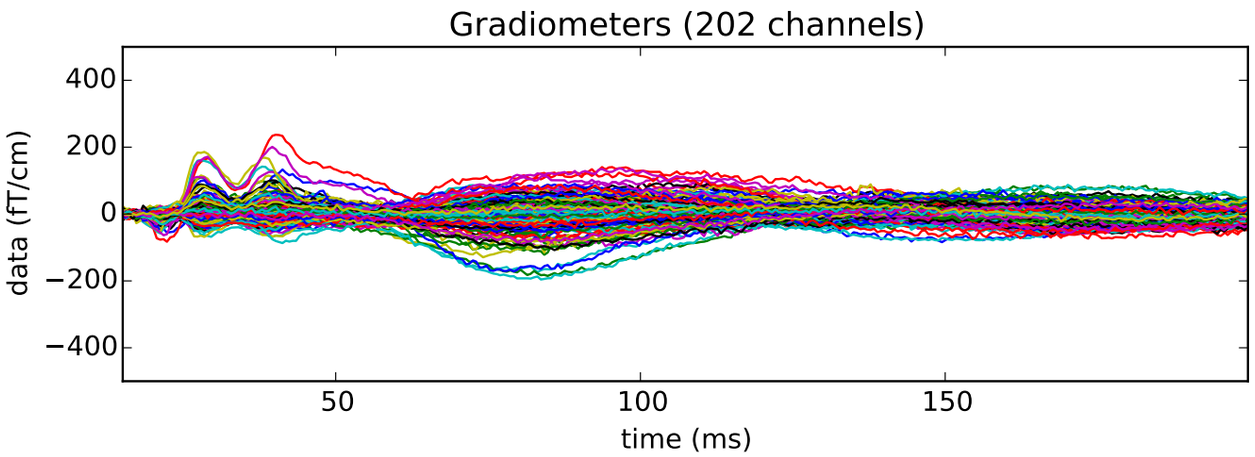
Result obtained with L2 regularization: $\phi(\mathbf{X}) = \|\mathbf{X}\|_F^2$

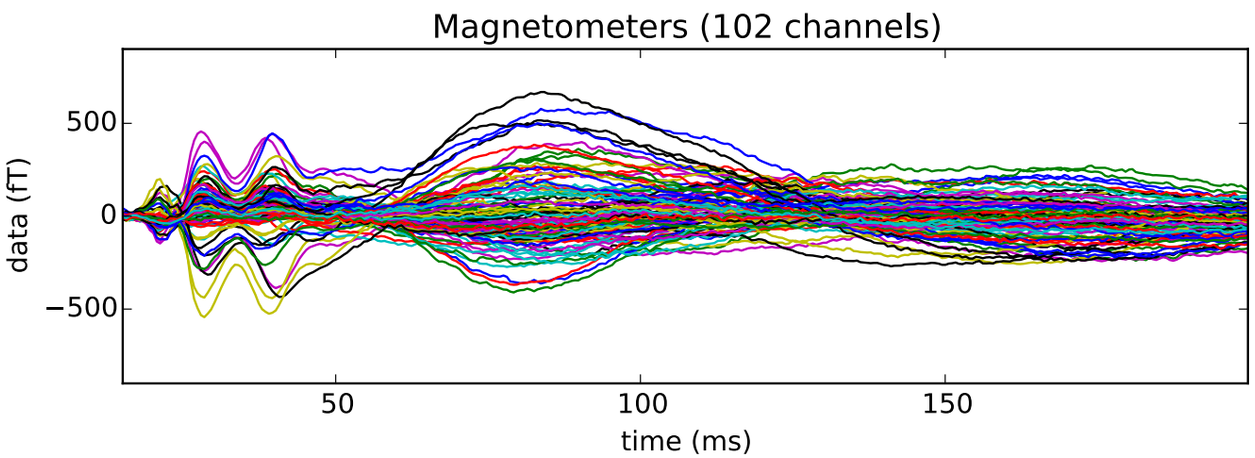


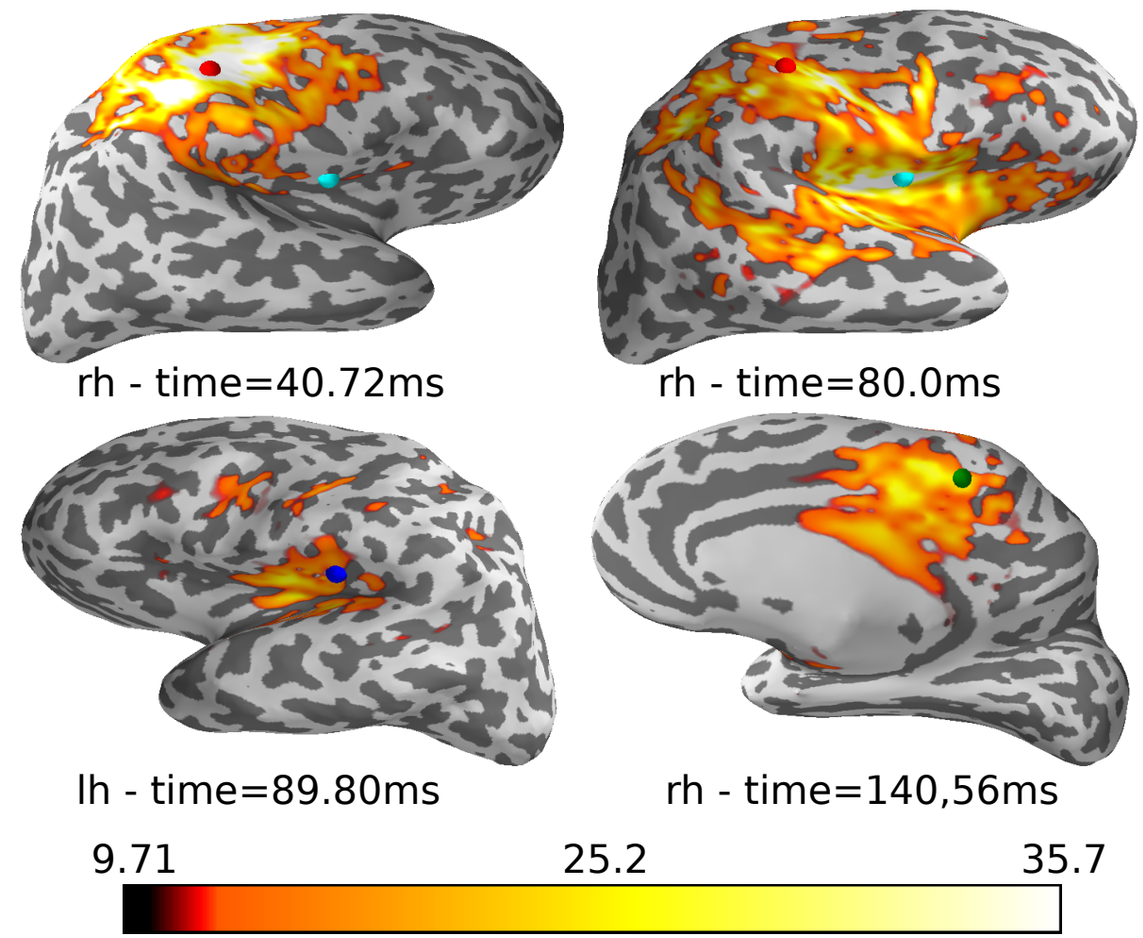
time=0.00 ms

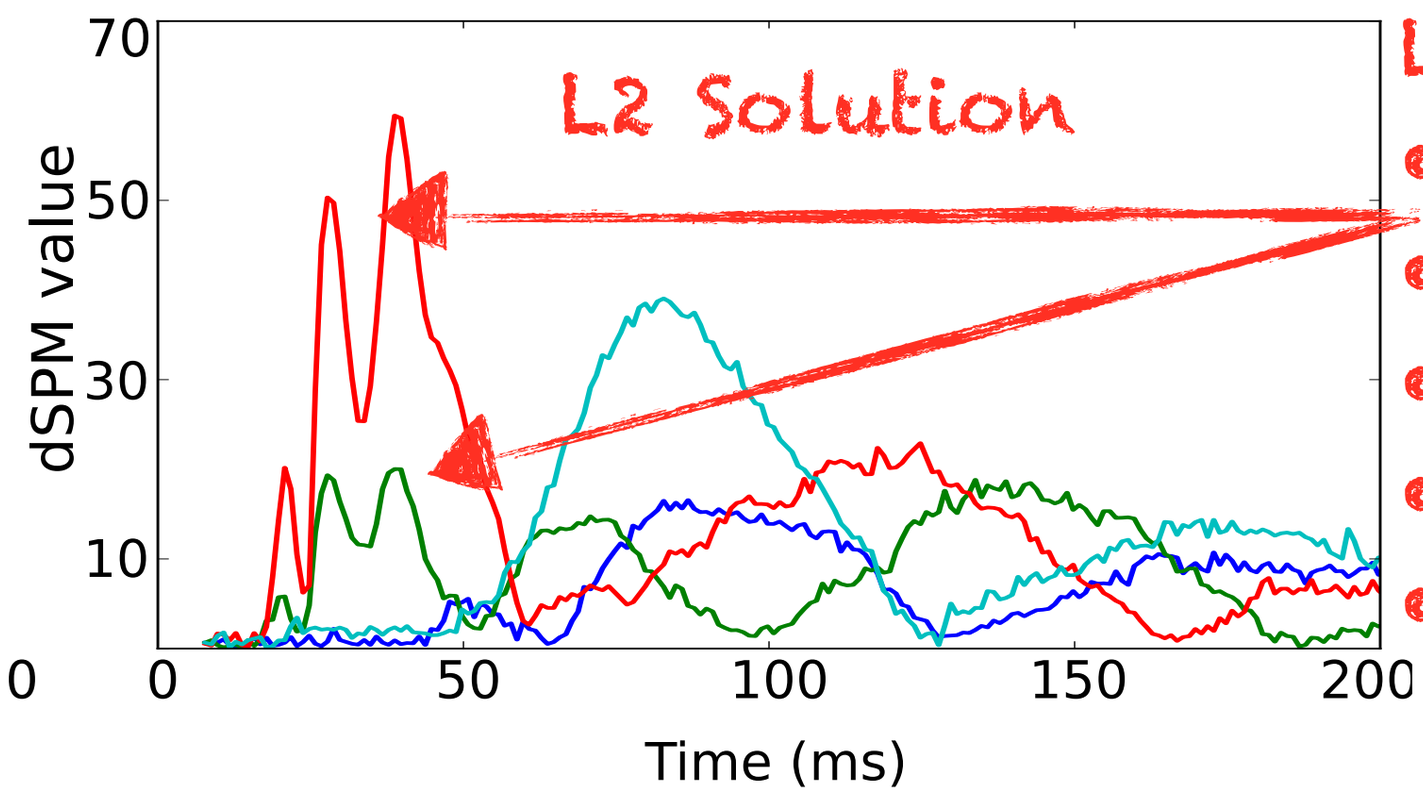
http://youtu.be/Uxr5Pz7JPrs

$$\phi(\mathbf{X}) = \|\mathbf{X}\|_F^2$$



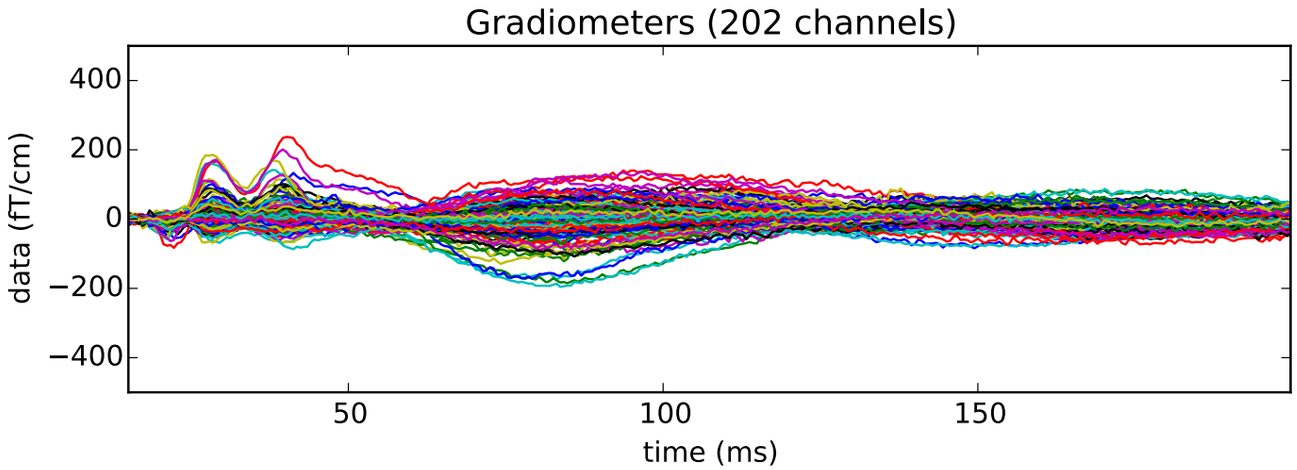



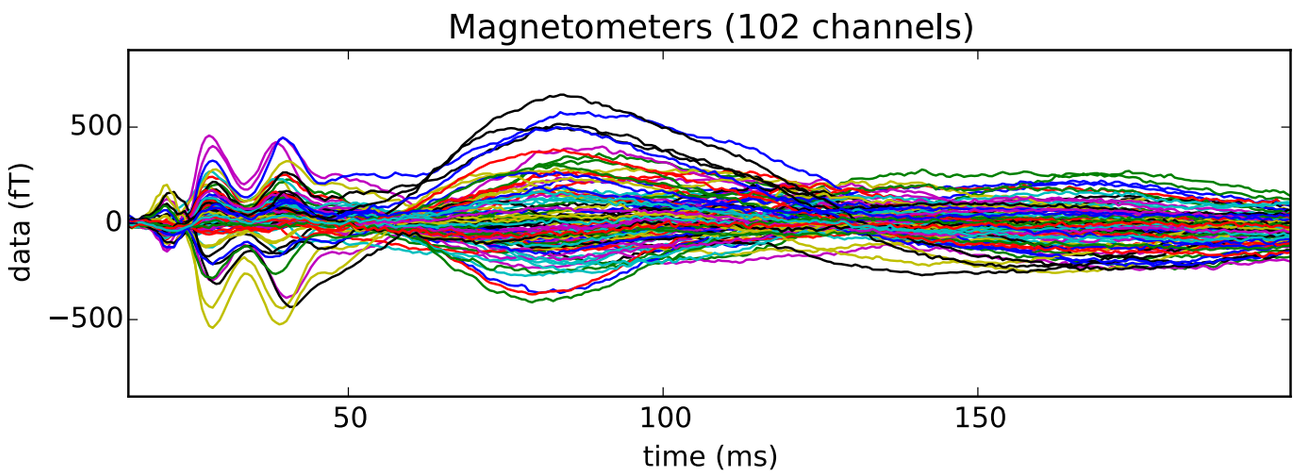



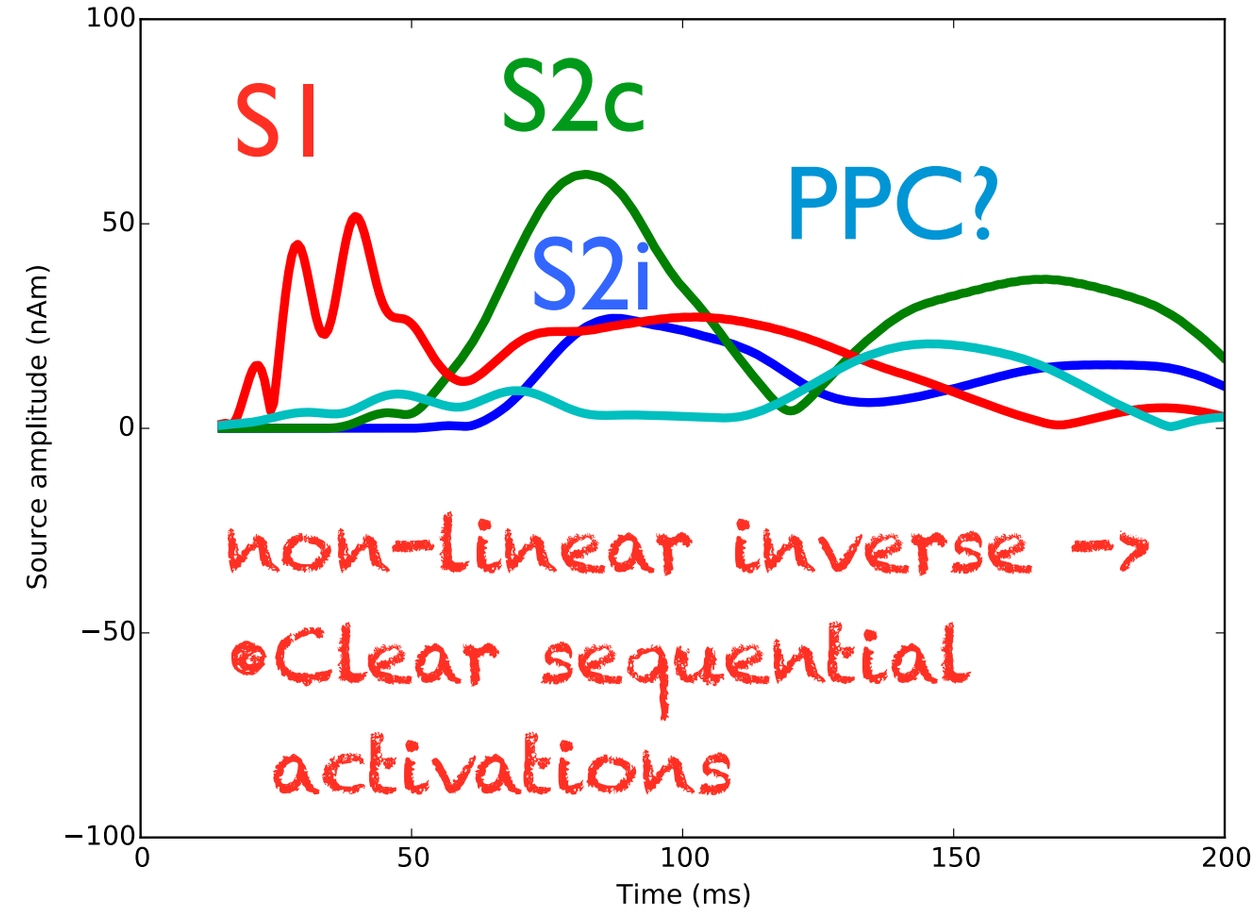

Linear inverse ->

- Imperfect deconvolution
- spatial leakage
- smeared activations
- no temporal smoothing
- but really fast...

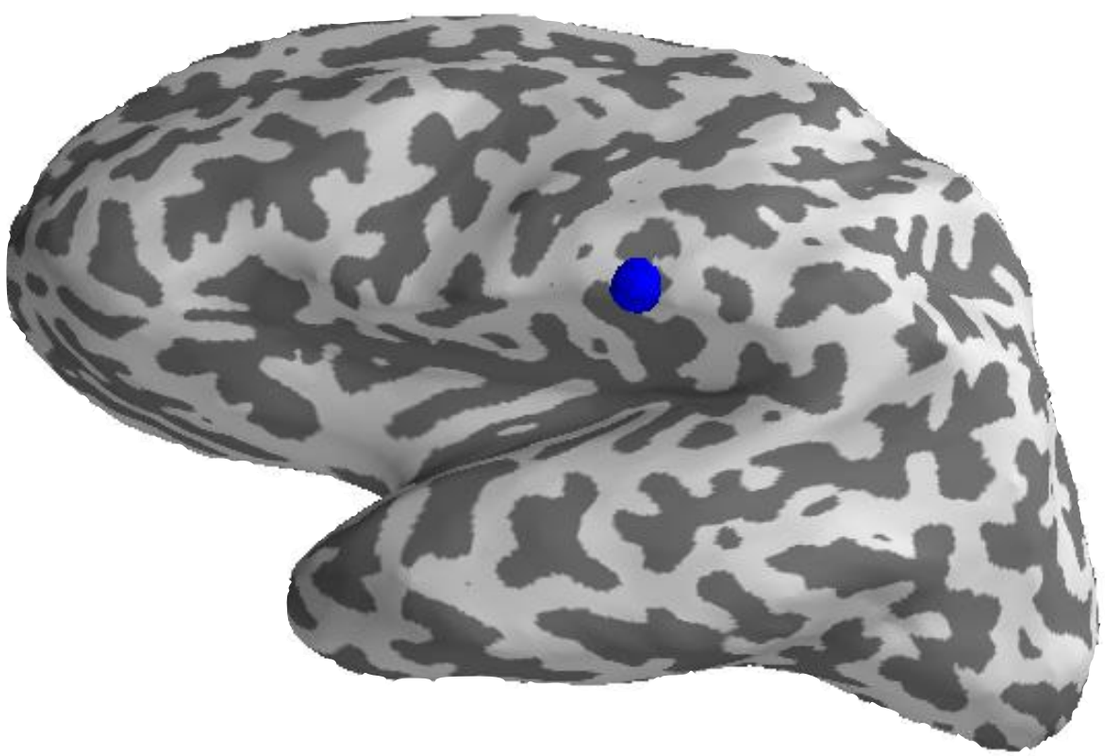
# $\phi(\mathbf{X})$ sparse / non-smooth

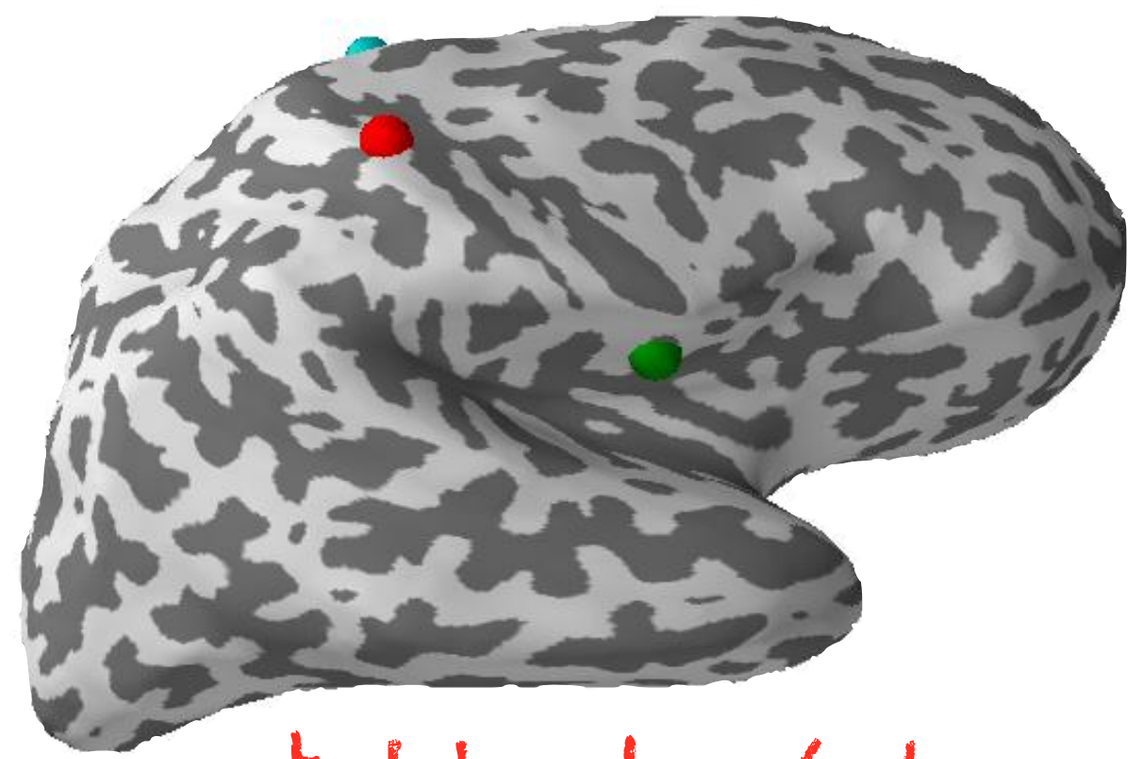


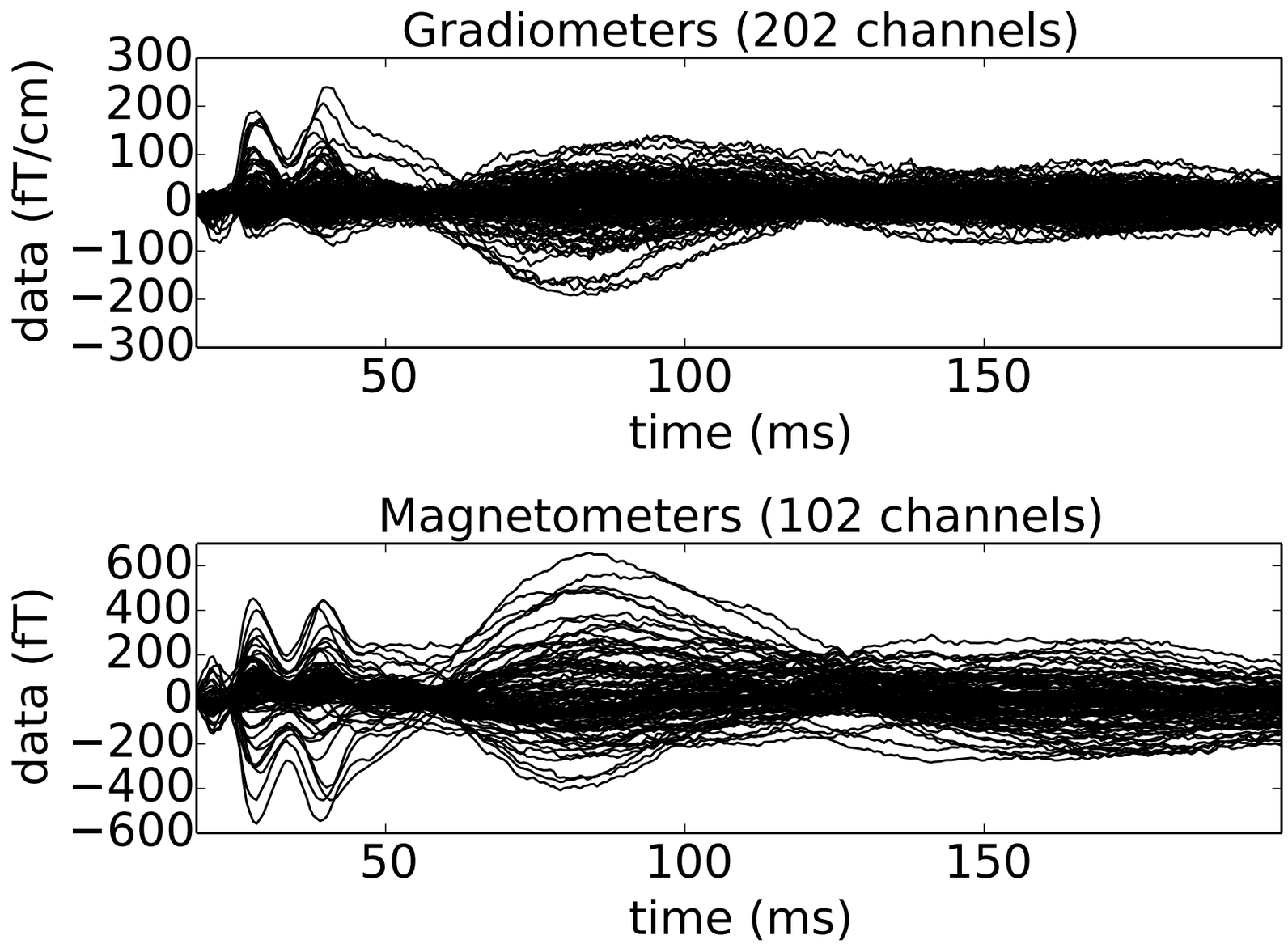

non-linear inverse ->
•Clear sequential activations





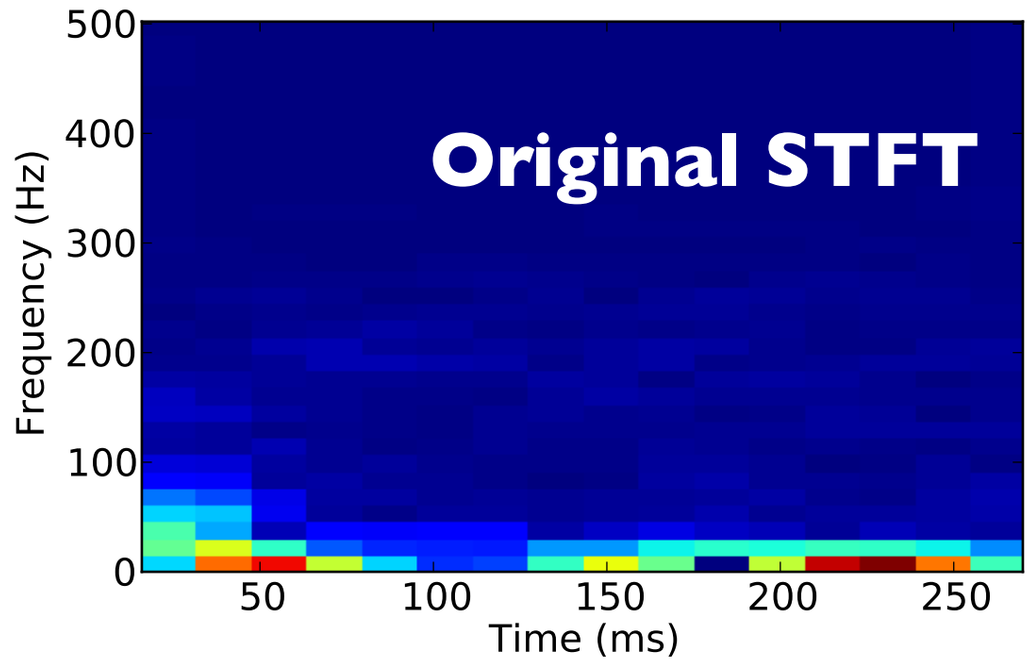
but harder / slower ....

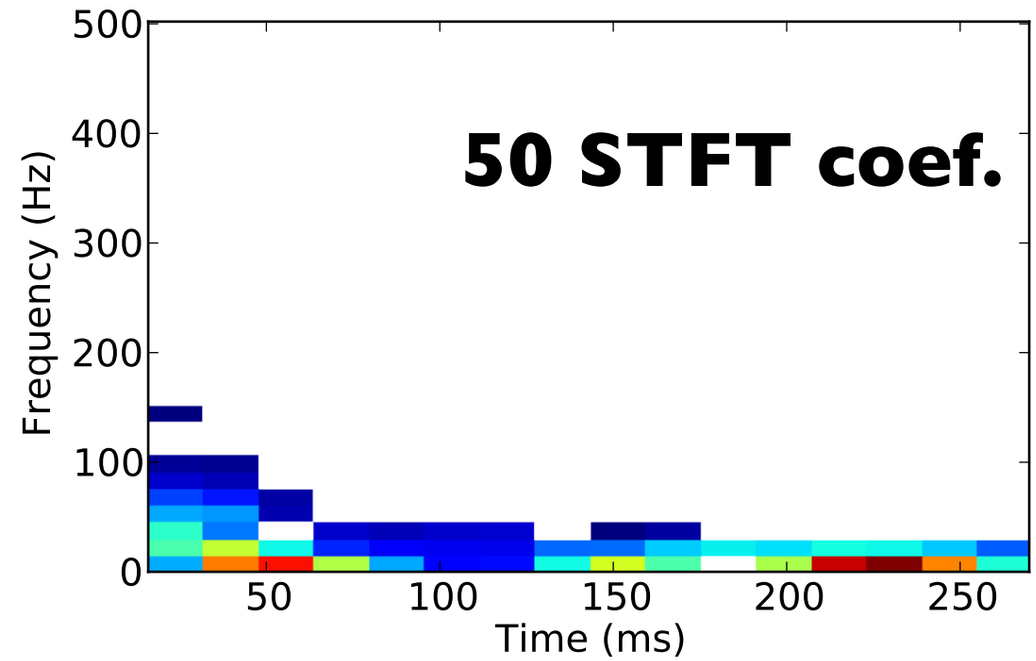
# Data are spatio-temporal

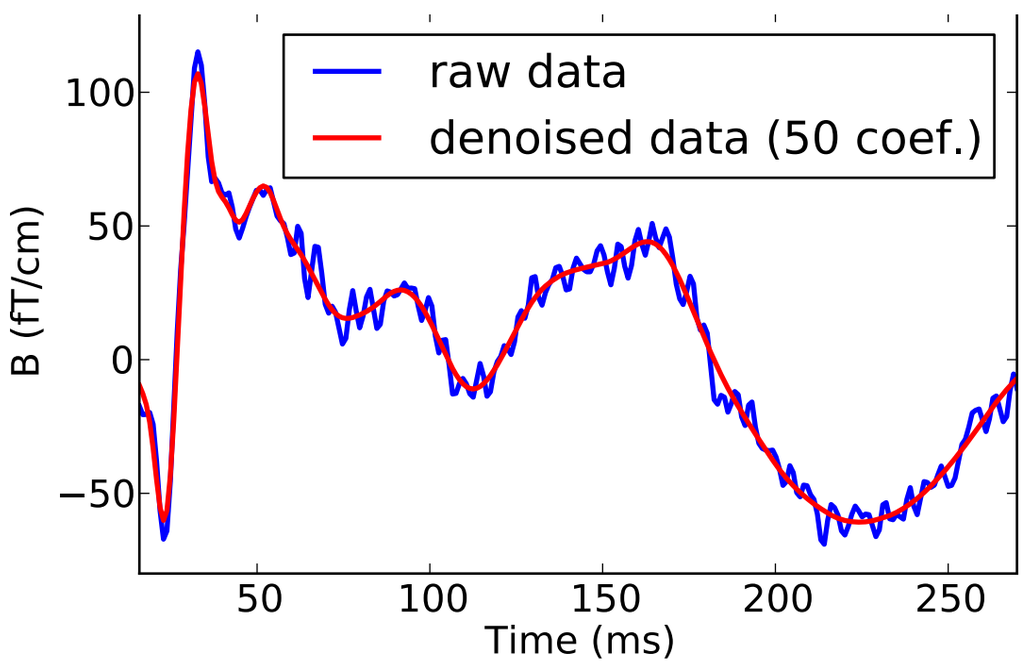



**Challenge: How do you promote sparse solutions with non-stationary sources?**

# Change the represensation



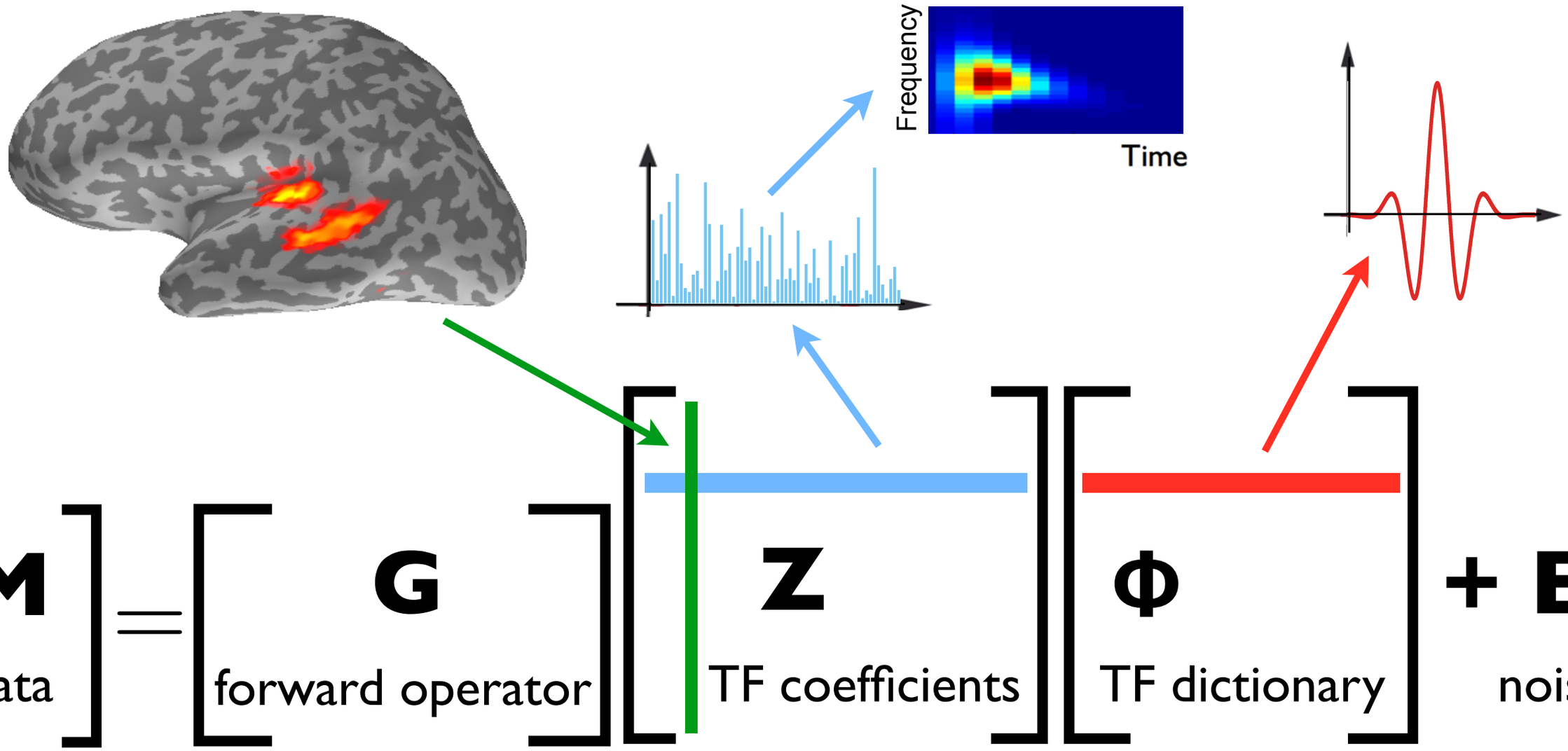



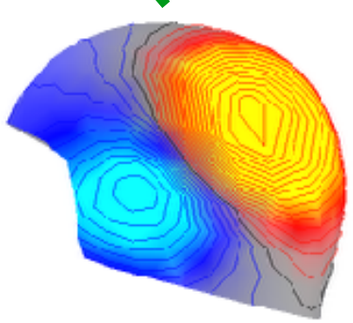




[“Wavelet shrinkage” Donoho & Johnstone 94]
[“Soft thresholding” Donoho 95]
[Application to evoked EEG, O. Bertrand et al. 94]
[Application to ST EEG, Quiroga et al. 03]
etc.

[Moussallam, Gramfort, Richard, Daudet, Signal Processing Letters 2014 ]

# $M = GZ\Phi + E$



**Objective: estimate Z given M**

[Gramfort et al., *Time-Frequency Mixed-Norm Estimates: Sparse M/EEG imaging with non-stationary source activations*, Neuroimage 2013]

# Time-frequency (TF) regularization

**The classical approach** [MNE, dSPM, sLORETA]:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \underbrace{\|\mathbf{M} - \mathbf{G}\mathbf{X}\|_F^2}_{\text{data fit}} + \underbrace{\lambda\phi(\mathbf{X})}_{\text{regularization}}, \ \lambda > 0$$

**we propose:**

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{Z}} \|\mathbf{M} - \mathbf{G}\mathbf{Z}\mathbf{\Phi}^{\mathcal{H}}\|_F^2 + \lambda\phi(\mathbf{Z}), \ \text{then } \hat{\mathbf{X}} = \hat{\mathbf{Z}}\mathbf{\Phi}^{\mathcal{H}}$$

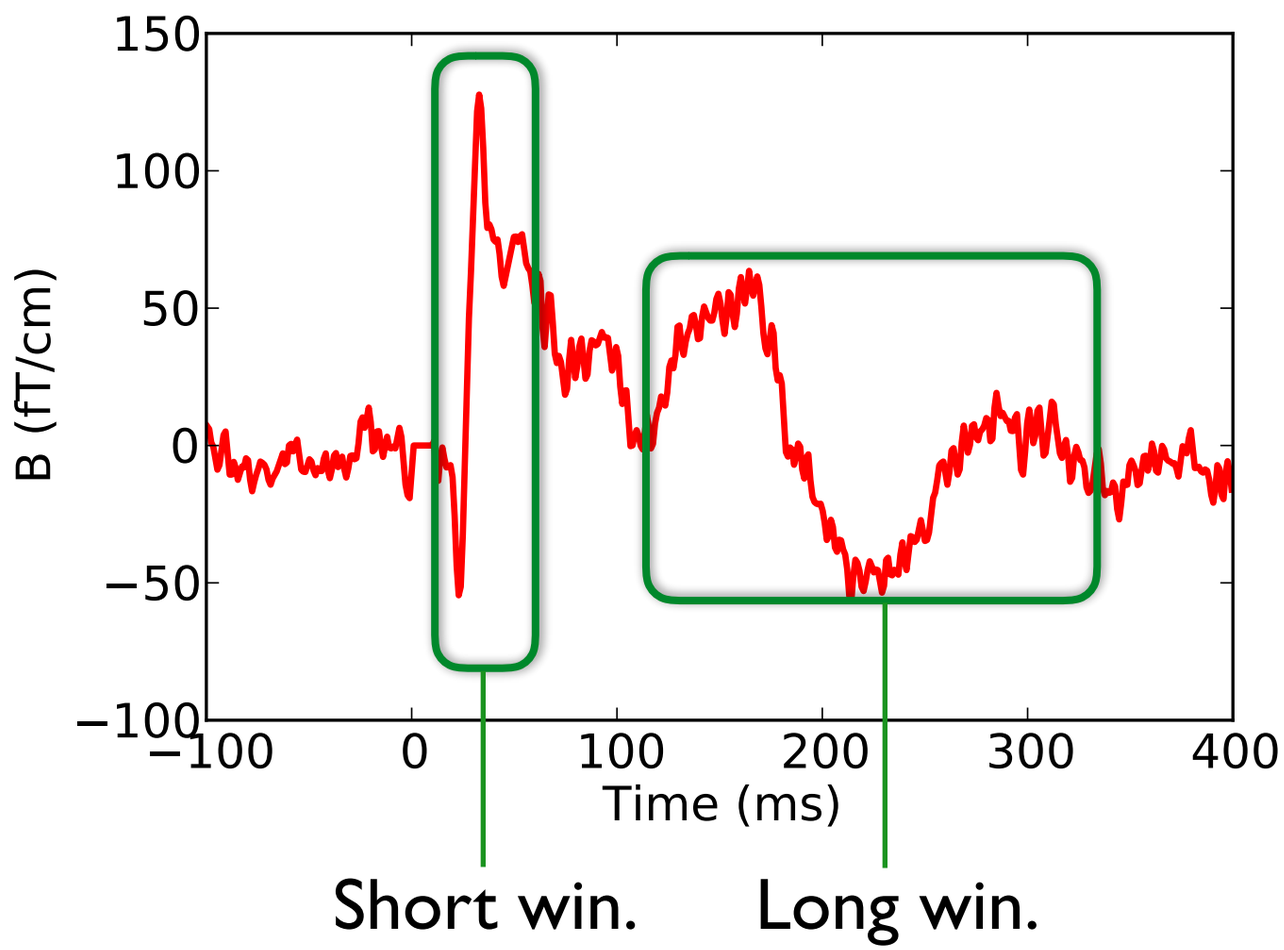
- $\Phi$ : is a **TF dictionary**
- $Z$ : **coefficients** of the **TF transform** of the sources

localization in space, time and frequency in one step

# Multi-scale dictionary

$$\hat{\mathbf{Z}} = \arg\min_{\mathbf{Z}} \|\mathbf{M} - \mathbf{G}\mathbf{Z}\mathbf{\Phi}^{\mathcal{H}}\|_F^2 + \lambda\phi(\mathbf{Z}), \text{ then } \hat{\mathbf{X}} = \hat{\mathbf{Z}}\mathbf{\Phi}^{\mathcal{H}}$$

- $\mathbf{\Phi}$ : union of n **STFT dict. with diff. window lengths**
- $\mathbf{Z}$ : is the combination of **coefficients** of the diff. **TF transforms** of the sources

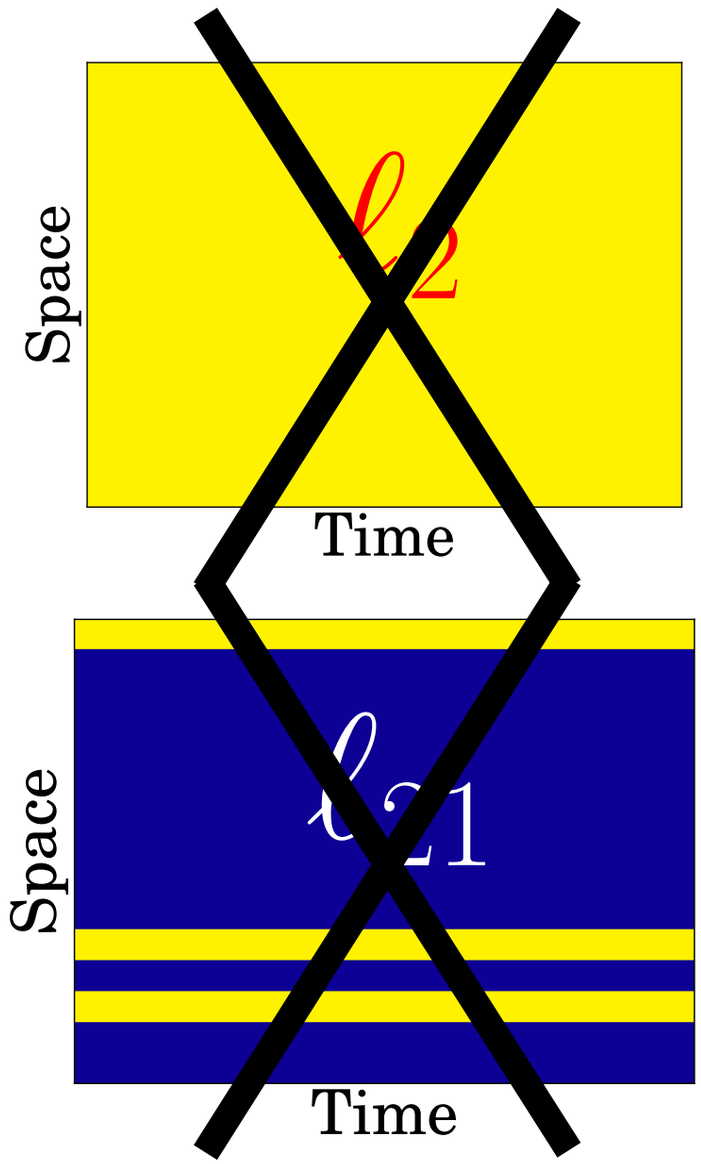


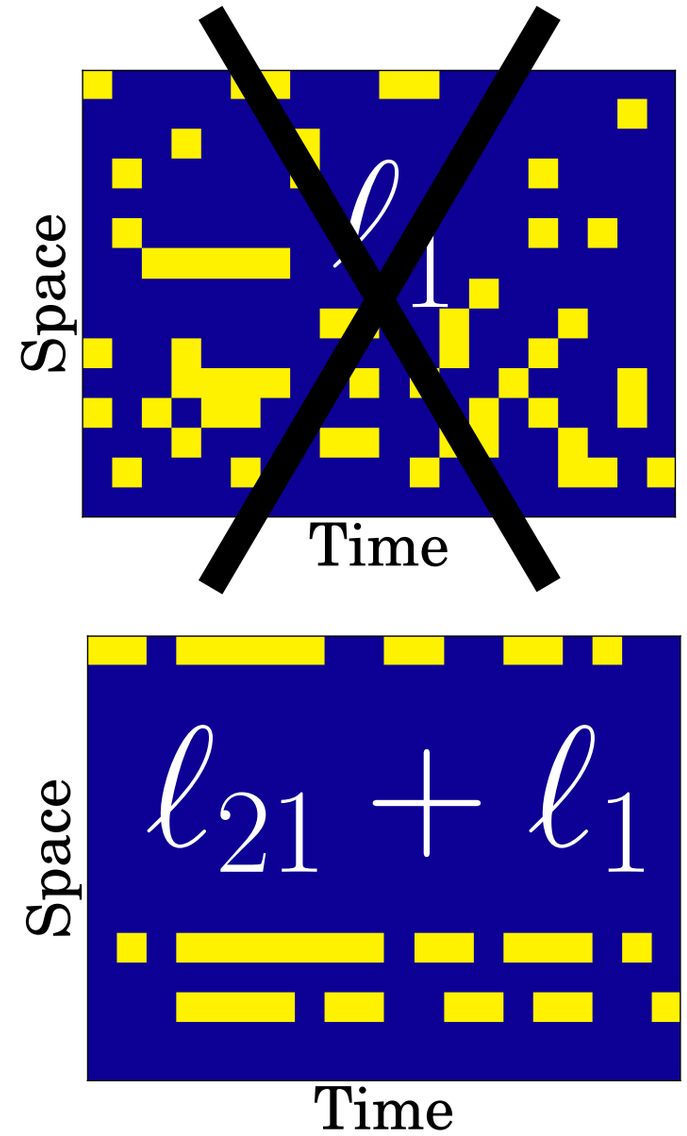

[Bekhti et al. 2016]

cf. [Kowalski et al. 2008]

cf. [Starck et al. 2005]

# What regularization?




Sparse Group Lasso

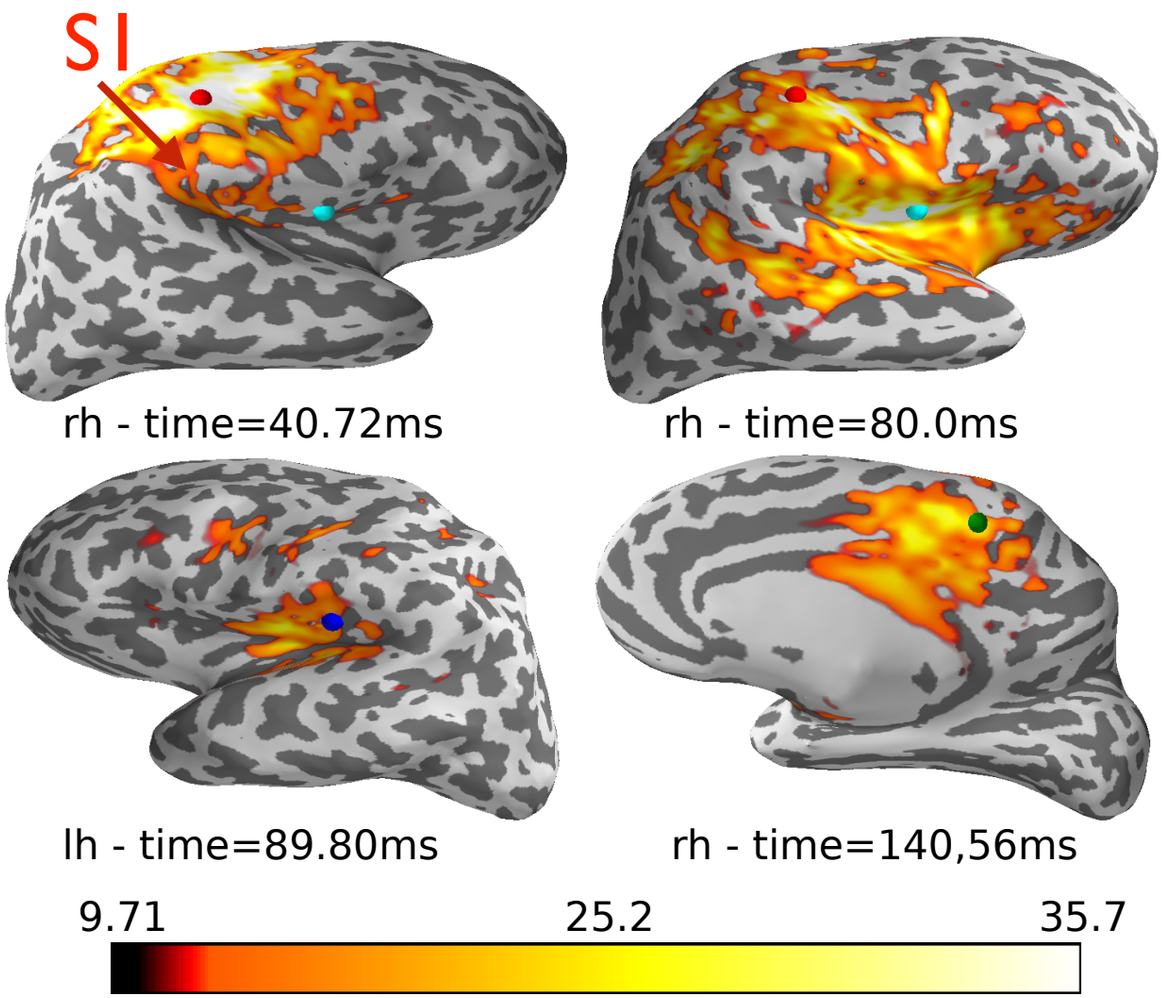
$$\phi(Z) = \lambda(\rho\|Z\|_1 + (1-\rho)\|Z\|_{21})$$

$$\|\mathbf{X}\|_{21} = \sum_i \sqrt{\sum_t |x_{i,t}|^2}$$

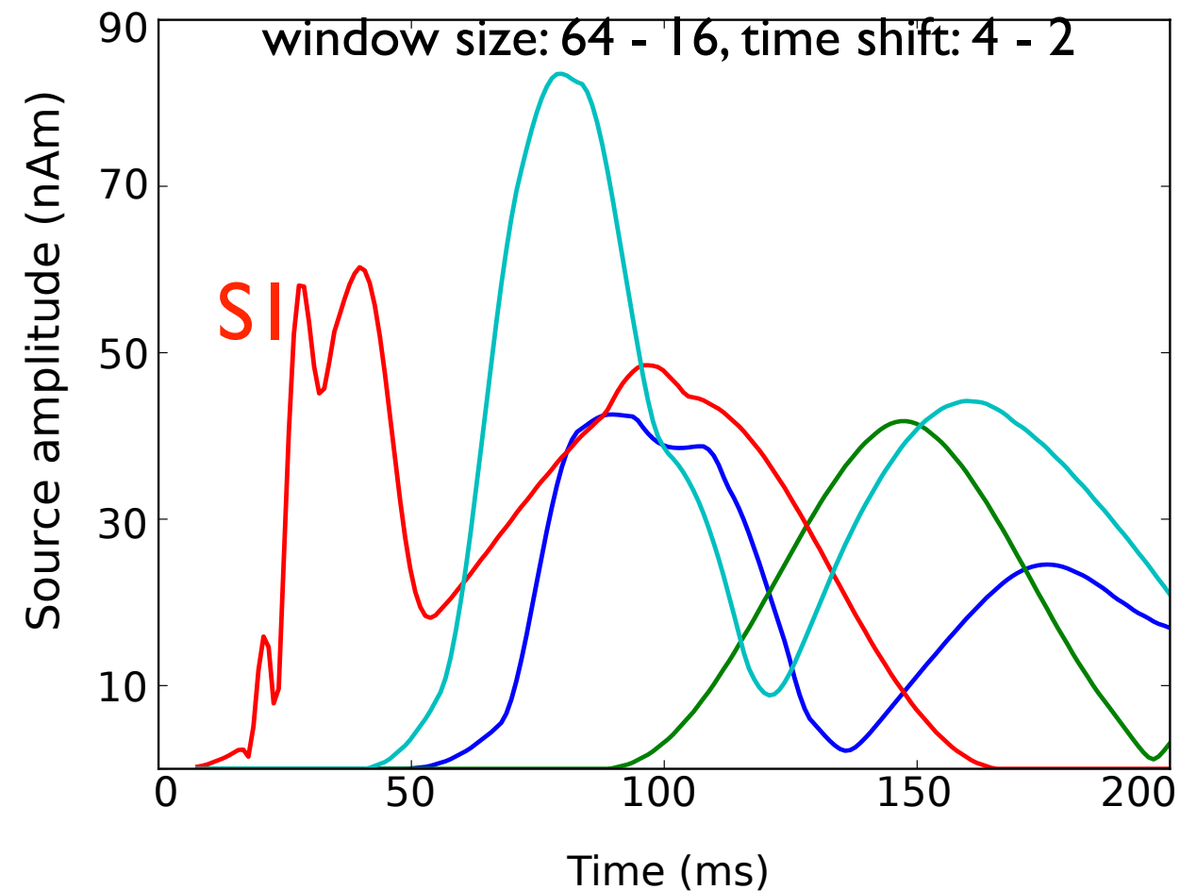
Safe screening for this model in [Ndiaye Fercoq Gramfort Salmon NIPS 2016 ]

# Results

Somatosensory - MIND dataset




dSPM




- Clear sequential activations
- No spatial leakage

[Bekhti et al. PRNI 2016]

# Optimization procedure

- Keys to fast solvers:
  - Block coordinate descent it the way !

    eg. [Tseng 2001, Friedman et al. 2007]
  - Gap Safe Screening rules

    [Fercoq et al. ICML 2015, N'Diaye et al. NIPS. 2016]
  - Working set strategies (cf. Blitz method)

    eg. [Johnson et al. ICML 2015]

**Algorithm 2** Coordinate descent (Lasso) with GAP Safe screening

**Input:** $X, y, \epsilon, K, f, (\lambda_t)_{t \in [T-1]}$

Initialization: $\lambda_0 = \lambda_{\max}, \quad \beta^{\lambda_0} = 0$
**for** $t \in [T-1]$ **do** ▷ Loop over $\lambda$'s
  $\beta \leftarrow \beta^{\lambda_{t-1}}$ ▷ previous $\epsilon$-solution
  **for** $k \in [K]$ **do**
    **if** $k \mod f = 1$ **then**
      Construct $\theta \in \Delta_X$, $A^{\lambda_t}(\mathcal{C}) = \{j \in [p] : \mu_{\mathcal{C}}(\mathbf{x}_j) \geqslant 1\}$
      **if** $G_{\lambda_t}(\beta, \theta) \leqslant \epsilon$ **then** ▷ Stop if duality gap small
        $\beta^{\lambda_t} \leftarrow \beta$
        **break**
      **end if**
    **end if**
    **for** $j \in A^{\lambda_t}(\mathcal{C})$ **do** ▷ Soft-Threshold coordinates
      $\beta_j \leftarrow \mathrm{ST}\Big(\frac{\lambda_t}{\|\mathbf{x}_j\|^2}, \beta_j - \frac{\mathbf{x}_j^\top (X\beta - y)}{\|\mathbf{x}_j\|^2}\Big)$
    **end for**
  **end for**
**end for**

**Algorithm 2** Coordinate descent (Lasso) with GAP Safe screening

**Input:** $X, y, \epsilon, K, f, (\lambda_t)_{t \in [T-1]}$

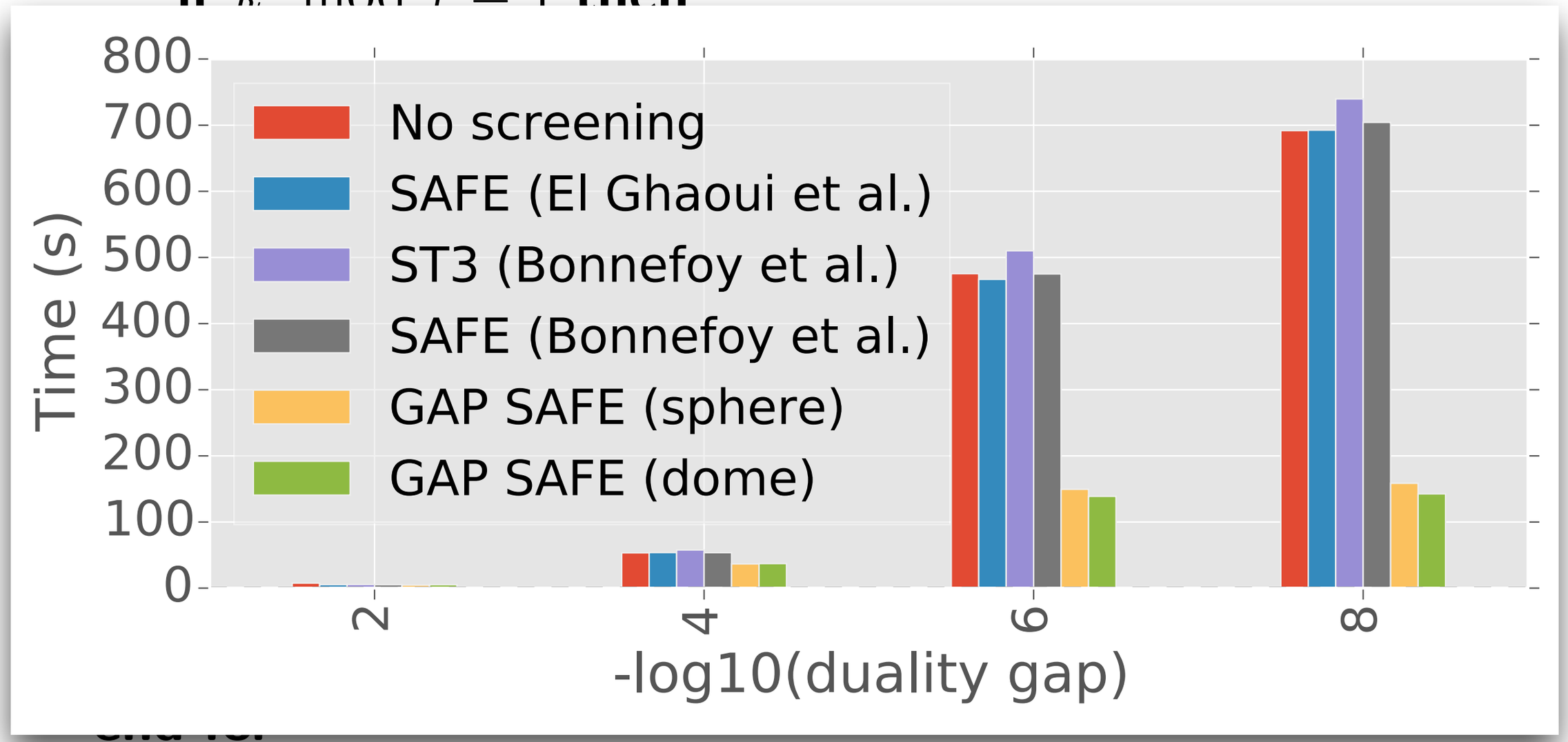
Initialization: $\lambda_0 = \lambda_{\max}, \quad \beta^{\lambda_0} = 0$

**for** $t \in [T-1]$ **do** ▷ Loop over $\lambda$'s

  $\beta \leftarrow \beta^{\lambda_{t-1}}$ ▷ previous $\epsilon$-solution

  **for** $k \in [K]$ **do**

    **if** $k \mod f = 1$ **then**




**end for**

[Fercoq et al. ICML 2015]

# Conclusion / Challenges

- Signals are very weak and can be terribly noisy
- Noise is often physiological (e.g. brain itself) so its statistics are very complex (non-stationarity, heteroscedastic, etc.)
- Data are not so small and studies move towards population for which computation time and automatic processing are the key.
- Brain imaging people need easy access to the tools / software to process their data.

# Conclusion / Challenges

- Signals are very weak and can be terribly noisy
- Noise is often physiological (e.g. brain itself) so its statistics are very complex (non-stationarity, heteroscedastic, etc.)
- Data are not so small and studies move towards population for which computation time and automatic processing are the key.
- Brain imaging people need easy access to the tools / software to process their data.

scikit-learn: machine learning in Python — scikit-learn 0.14 documentation

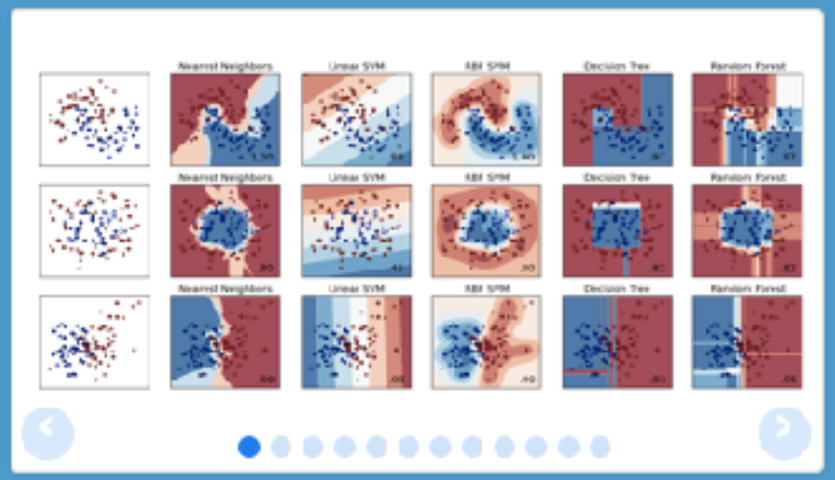
scikit-learn.org/stable/

scikit learn

Home | Installation | Documentation | Examples

Search

Fork me on GitHub



# scikit-learn

*Machine Learning in Python*

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## Classification

Identifying to which set of categories a new observation belong to.

**Applications**: Spam detection, Image recognition.
**Algorithms**: *SVM*, *nearest neighbors*, *random forest*, ... — *Examples*

## Regression

Predicting a continuous value for a new example.

**Applications**: Drug response, Stock prices.
**Algorithms**: *SVR*, *ridge regression*, *Lasso*, ... — *Examples*

## Clustering

Automatic grouping of similar objects into sets.

**Applications**: Customer segmentation, Grouping experiment outcomes
**Algorithms**: *k-Means*, *spectral clustering*, *mean-shift*, ... — *Examples*

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications**: Visualization, Increased efficiency
**Algorithms**: *PCA*, *Isomap*, *non-negative matrix factorization*. — *Examples*

## Model selection

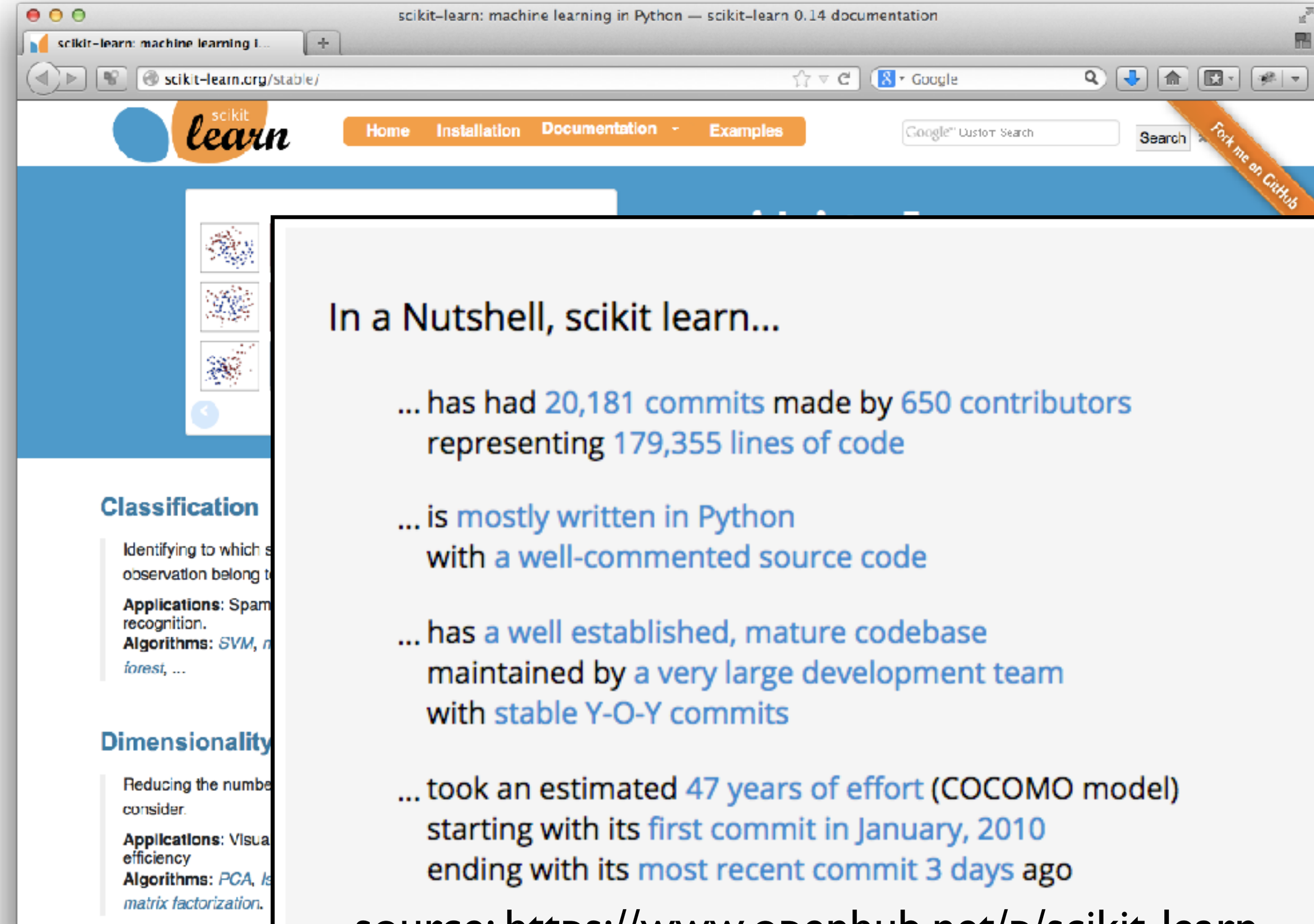
Comparing, validating and choosing parameters and models.

**Goal**: Improved accuracy via parameter tuning
**Modules**: *grid search*, *cross validation*, *metrics*. — *Examples*

## Preprocessing

Feature extraction and normalization.

**Application**: Transforming input data such as text for use with machine learning algorithms.
**Modules**: *preprocessing*, *feature extraction*. — *Examples*




# In a Nutshell, scikit learn...

... has had 20,181 commits made by 650 contributors
representing 179,355 lines of code

... is mostly written in Python
with a well-commented source code

... has a well established, mature codebase
maintained by a very large development team
with stable Y-O-Y commits

... took an estimated 47 years of effort (COCOMO model)
starting with its first commit in January, 2010
ending with its most recent commit 3 days ago

source: https://www.openhub.net/p/scikit-learn

scikit-learn: machine learning in Python — scikit-learn 0.14 documentation

scikit-learn.org/stable/

Home | Installation | Documentation | Examples

Fork me on GitHub

## Classification

Identifying to which s
observation belong t

**Applications**: Spam
recognition.
**Algorithms**: *SVM*, *n*
*forest*, ...

## Dimensionality

Reducing the numbe
consider.

**Applications**: Visua
efficiency
**Algorithms**: *PCA*, *Is*
*matrix factorization*.

# In a Nutshell, scikit learn...

... has had 20,181 co
representing 17

... is mostly wri
with a well-

... has a well
maintained
with stable

... took an estim
starting with its
ending with its m

Funding:

source: https://www. scikit-learn

MNE Tutorials Gallery Manual API FAQ Cite Site Page Search

# MNE

MEG + EEG ANALYSIS & VISUALIZATION

http://www.martinos.org/mne

MNE is a community-driven software package designed for for **processing electroencephalography (EEG) and magnetoencephalography (MEG) data** providing comprehensive tools and workflows for:

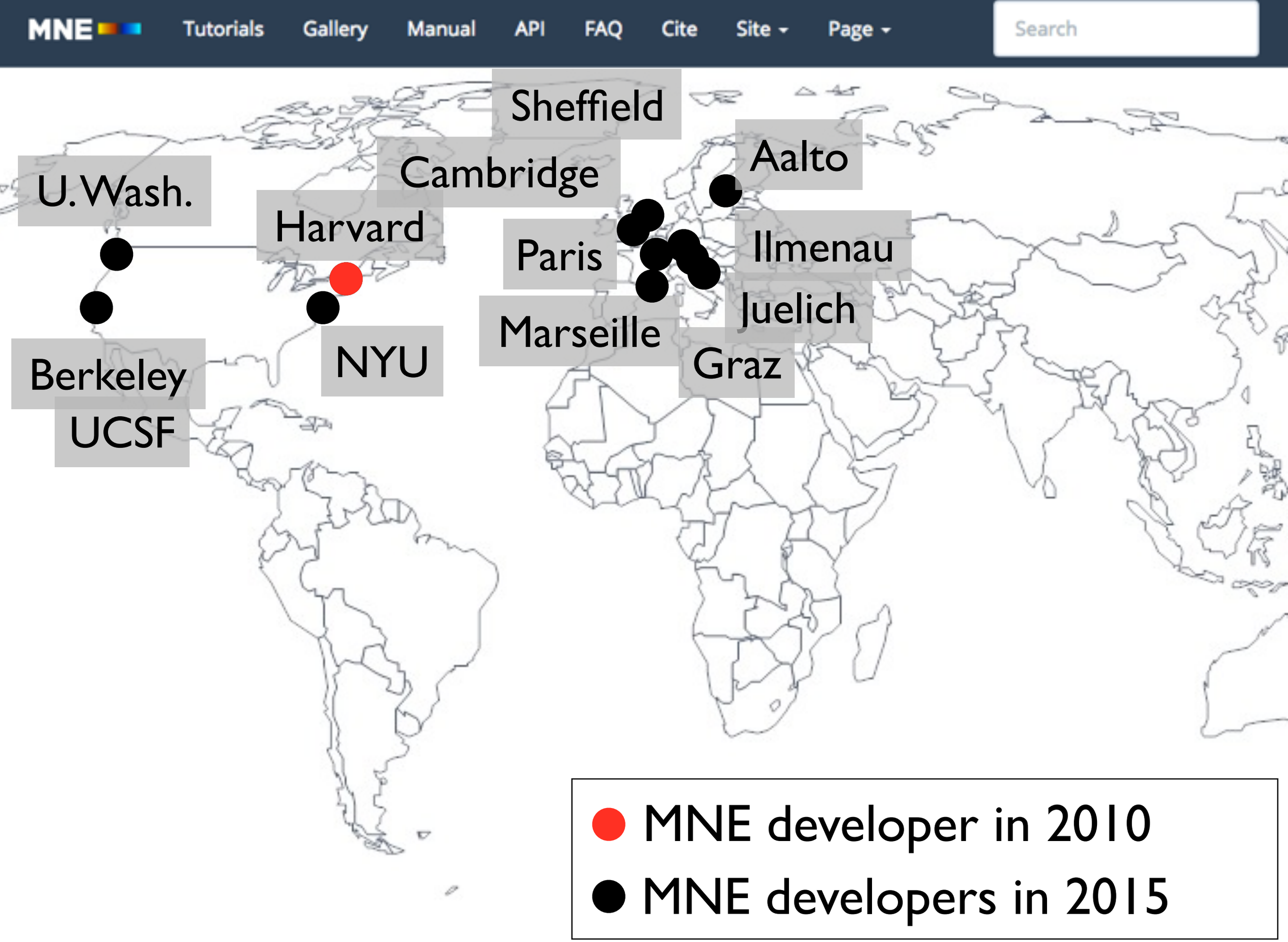
1. Preprocessing
2. Source estimation
3. Time–frequency analysis
4. Statistical testing
5. Estimation of functional connectivity
6. Applying machine learning algorithms
7. Visualization of sensor- and source-space data

MNE includes a comprehensive Python package (provided under the simplified BSD license), supplemented by tools compiled from C code for the LINUX and Mac OSX operating systems, as well as a MATLAB toolbox.

## Documentation

- Getting Started
- What's new
- Cite MNE
- Related publications
- Tutorials
- Examples Gallery
- Manual
- API Reference
- Frequently Asked Questions
- Advanced installation and setup
- MNE with CPP

*MNE software for processing MEG and EEG data*, A. Gramfort, M. Luessi, E. Larson, D. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, M. Hämäläinen, Neuroimage 2013


MNE
Tutorials
Gallery
Manual
API
FAQ
Cite
Site
Page
Search
Sheffield
Cambridge
Aalto
U. Wash.
Harvard
Paris
Ilmenau
Juelich
Marseille
Graz
NYU
Berkeley
UCSF
MNE developer in 2010
MNE developers in 2015

# Thanks !

- M. Hämäläinen
- M. Kowalski
- D. Strohmeier
- J. Haueisen
- Y. Bekhti
- M. Jas
- G. Varoquaux
- B. Thirion
- M. Eickenberg
- F. Pedregosa
- J. Salmon
- O. Fercoq
- E. Ndiaye
- … the scikit-learn contributors
- … the MNE contributors

## Post-docs positions available !

**Contact** http://alexandre.gramfort.net

*GitHub : @agramfort* *Twitter : @agramfort*

**Support** ERC SLAB, ANR THALAMEEG ANR-14-NEUC-0002-01
NIH R01 MH106174, DFG HA 2899/21-1.

# References


Gramfort et al., *Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods*, Physics in Medicine and Biology, 2012

Gramfort et al. *Time-frequency mixed-norm estimates: Sparse M/EEG imaging with non-stationary source activations*, NeuroImage, 2013

Strohmeier et al., *Improved MEG/EEG source localization with reweighted mixed-norms*, International Workshop on Pattern Recognition in Neuroimaging (PRNI), 2014

Strohmeier et al., *MEG/EEG source imaging with a non-convex penalty in the time-frequency domain*, Int. Workshop on Pattern Recognition in Neuroimaging (PRNI), 2015

Fercoq et al., *Mind the duality gap: Safer rules for the Lasso,* ICML, 2015

Ndiaye et al., *GAP Safe screening rules for sparse multi-task and multi-class models,* NIPS, 2015

Strohmeier et al. *The iterative reweighted Mixed-Norm Estimate for spatio-temporal MEG/EEG source localization*. IEEE Transactions on Medical Imaging (TMI), 2016.

Bekhti et al. *M/EEG source localization with multi-scale time-frequency dictionaries,* Int. Workshop on Pattern Recognition in Neuroimaging (PRNI), 2015

Jas et al. *Autoreject: Automated artifact rejection for MEG and EEG*, NeuroImage 2017