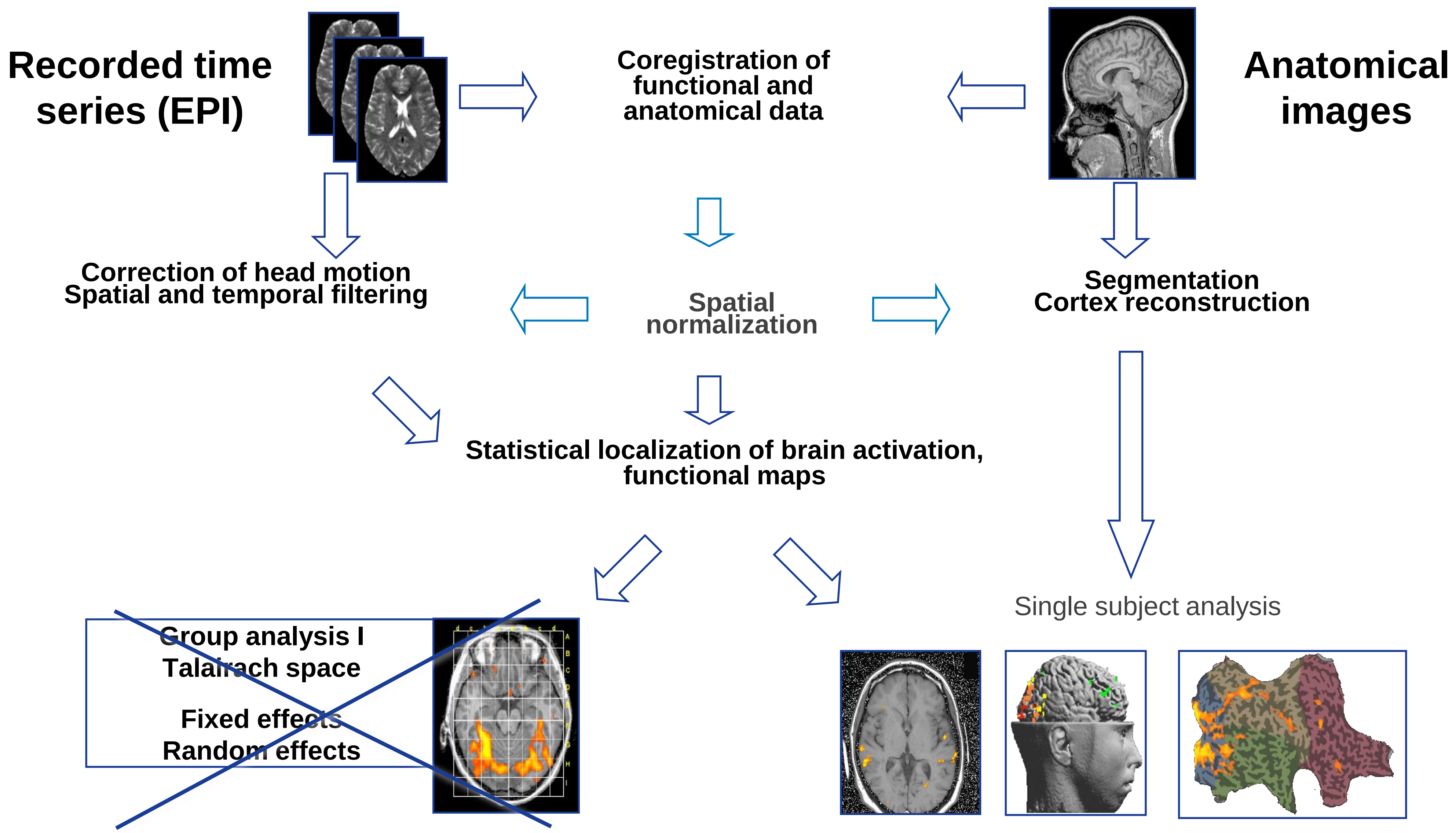
Data preprocessing & Analysis

Michael Lührs

Overview

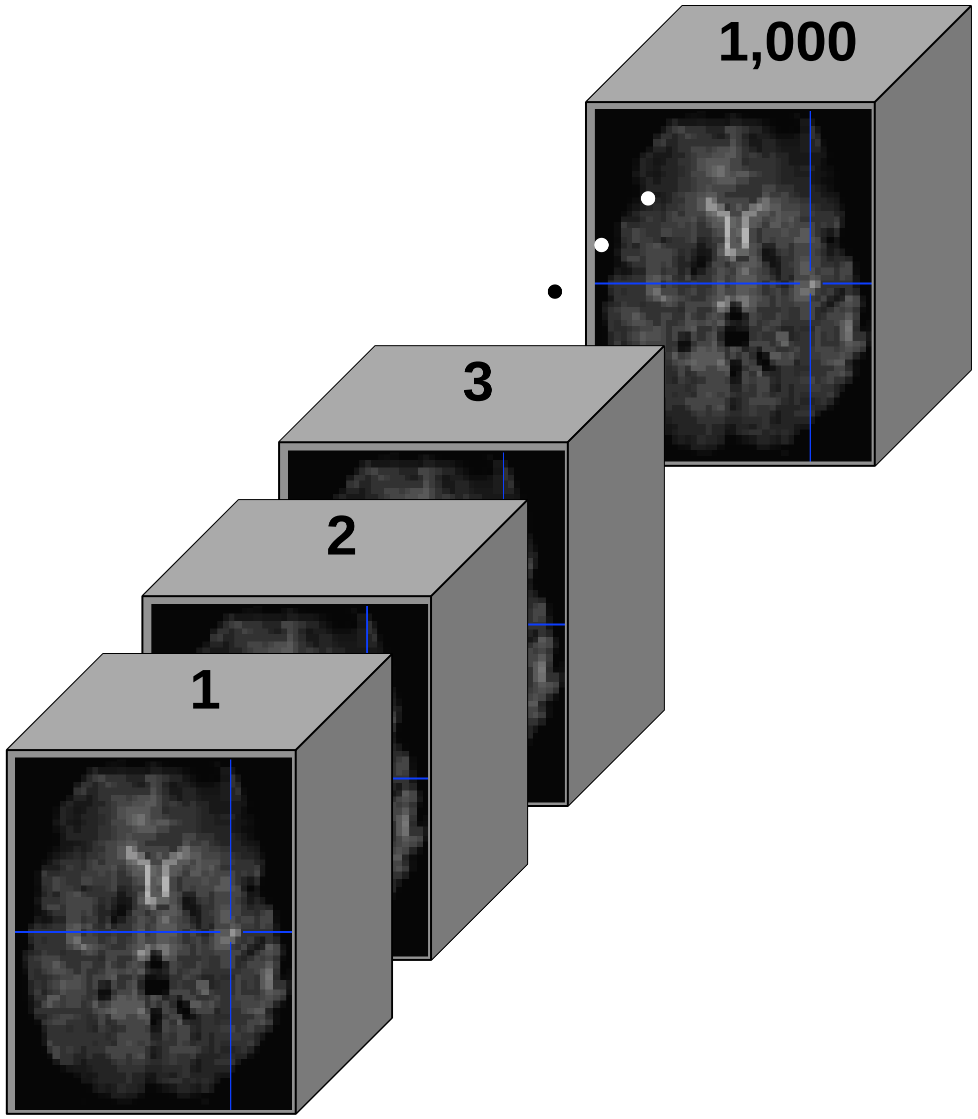
1. Data preprocessing & Analysis
2. Hands-On
3. Exporting information

Flow Chart of Basic fMRI Data Analysis Steps



Functional MRI Data Analysis

Time series of 3D data volumes



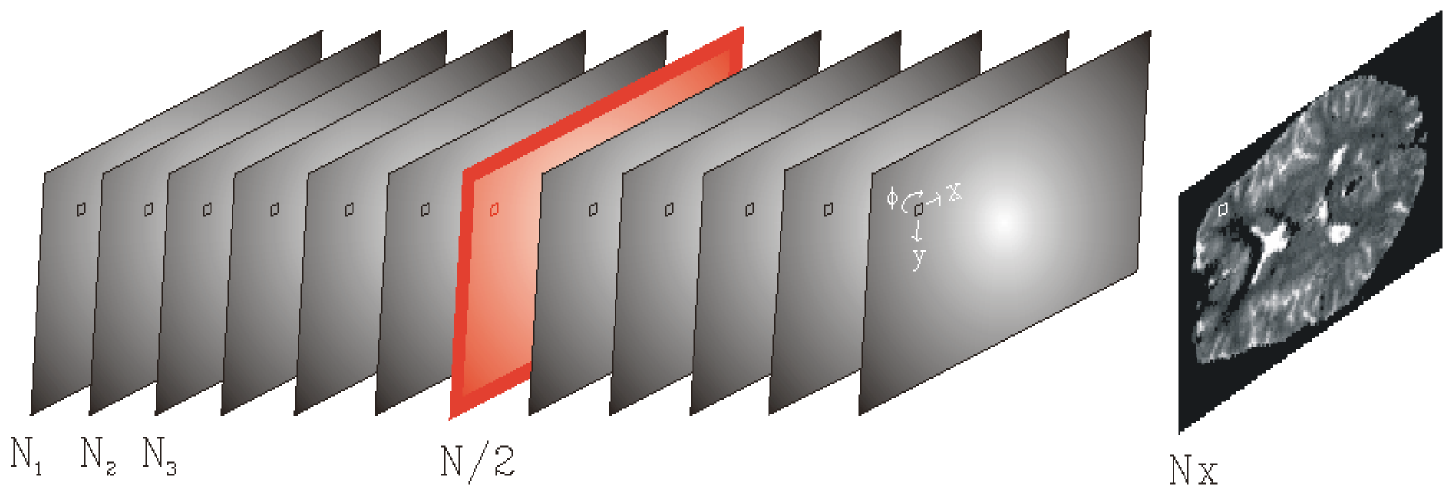
How to Ensure Real-Time Processing?

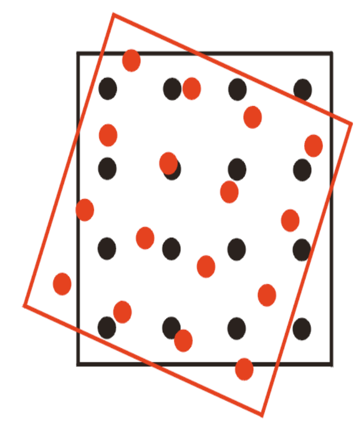
- **Standard routines should be reformulated to allow efficient incremental processing with a constant time per time point (TR). One can, for example, use recursive least squares instead of standard GLM implementation.**
- **To gain maximum speed, computational routines should be implemented in a language producing most efficient machine code such as C/C++ or Python/Matlab using a C/C++ backend.**
- **For standard routines such as vector algebra calls (e.g. matrix multiplication and matrix inversion), efficient parallel CPU implementations can be used such as the Intel 'Math Kernel Library' (MKL).**
- **For efficient implementation of routines that can be parallelized, GP-GPU (e.g. OpenCL) implementations can be used, e.g. to implement 3D motion correction with sinc interpolation sampling.**
- **Graphical User Interface (GUI) and display of e.g. volume / surface maps and time courses should use efficient visualization libraries, preferentially implemented using C++ and OpenGL / Vulkan.**

Example: 3D Motion Correction Using GPU Sinc Interpolation

Motion Detection

Intensity-based algorithm searches for subvoxel *translation* (t-x, t-y and t-z in mm) and *rotation* (r-x, r-y and r-z in degrees) w.r.t. to x, y and z axes



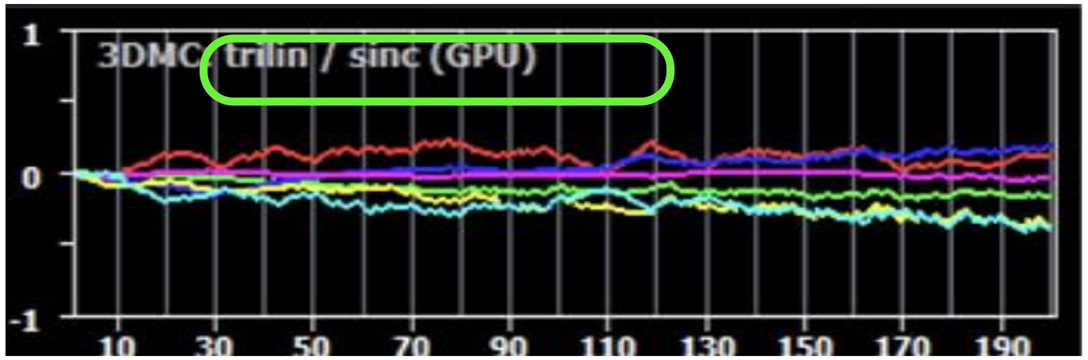


Motion Correction

Spatial interpolation is necessary and should use high quality approach such as cubic spline or sinc. Different interpolation options can be used during iterative estimation and final interpolation step:

- ***full trilinear* (very fast)**
- ***full sinc* (very slow)**
- **trilinear / sinc (good speed)**

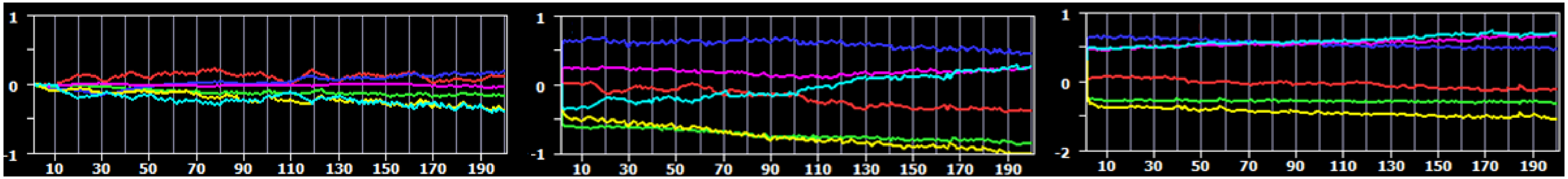
Sinc - sin(x)/x - interpolation can be optimised using GPU!



Plot from real-time software Turbo-BrainVoyager

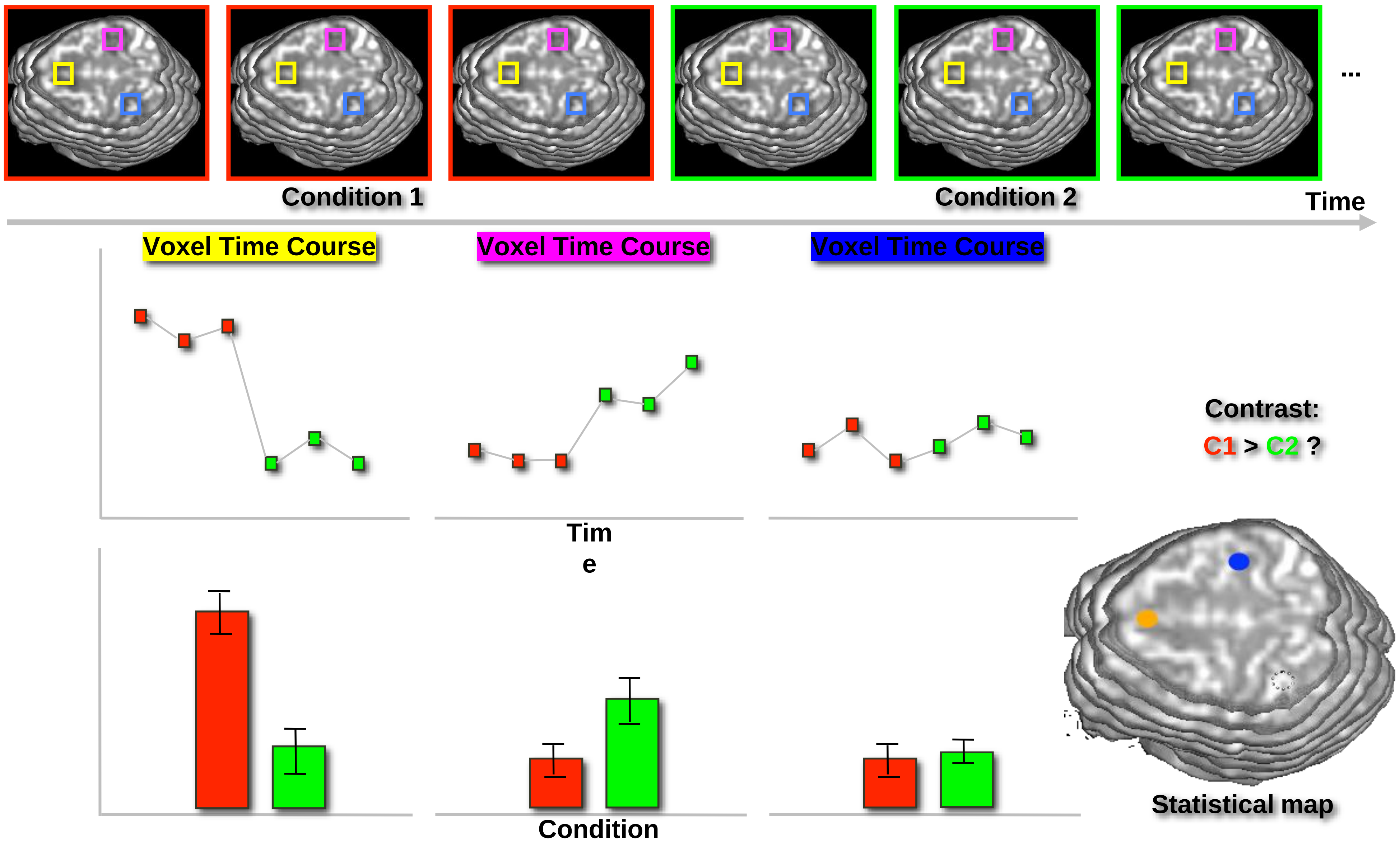
Intra-Session Motion Correction

- Correct motion to the first run of your real-time session
  - Ensures that same region is used for each run

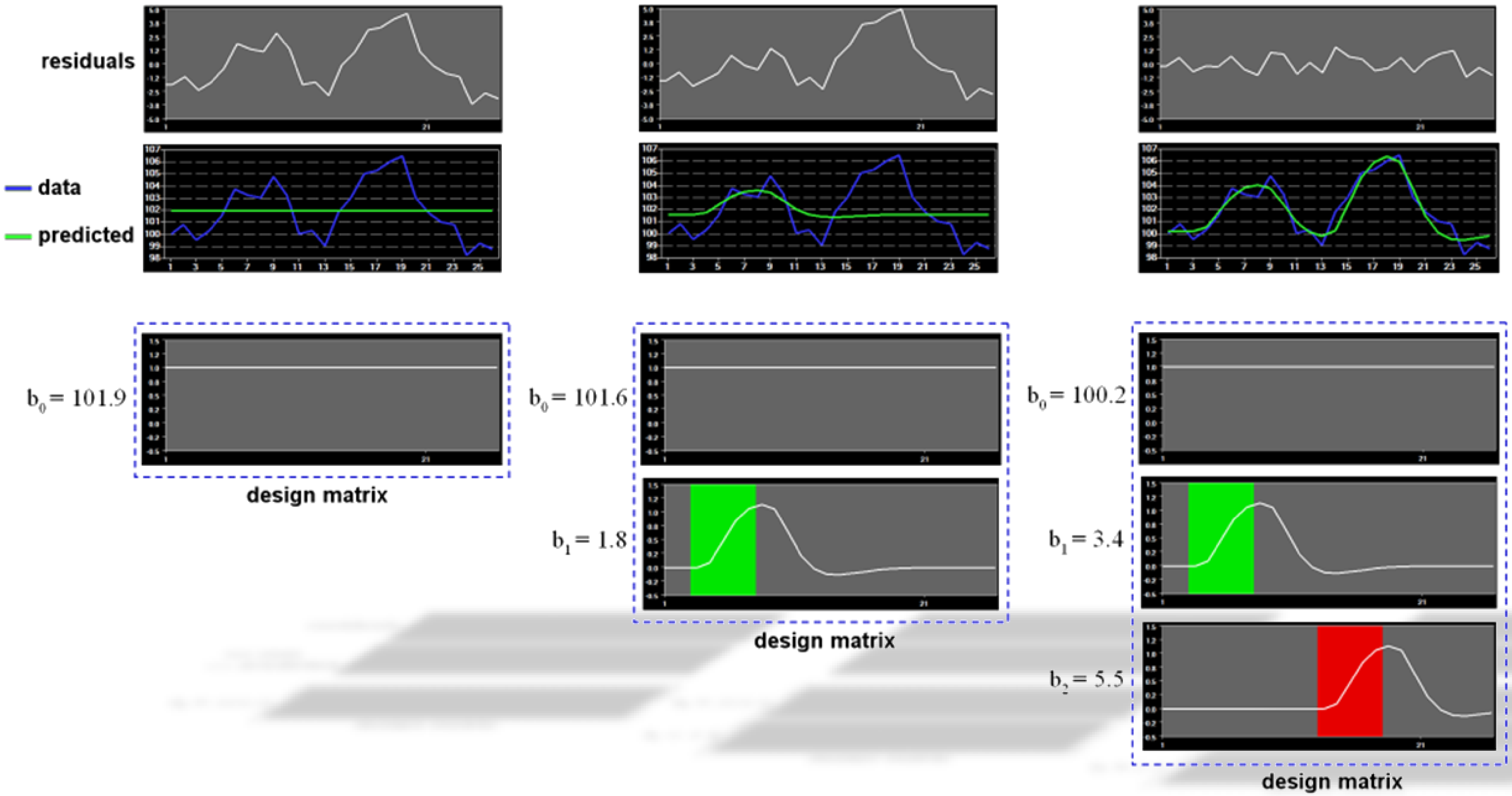


Run-1 Run-2 Run-3

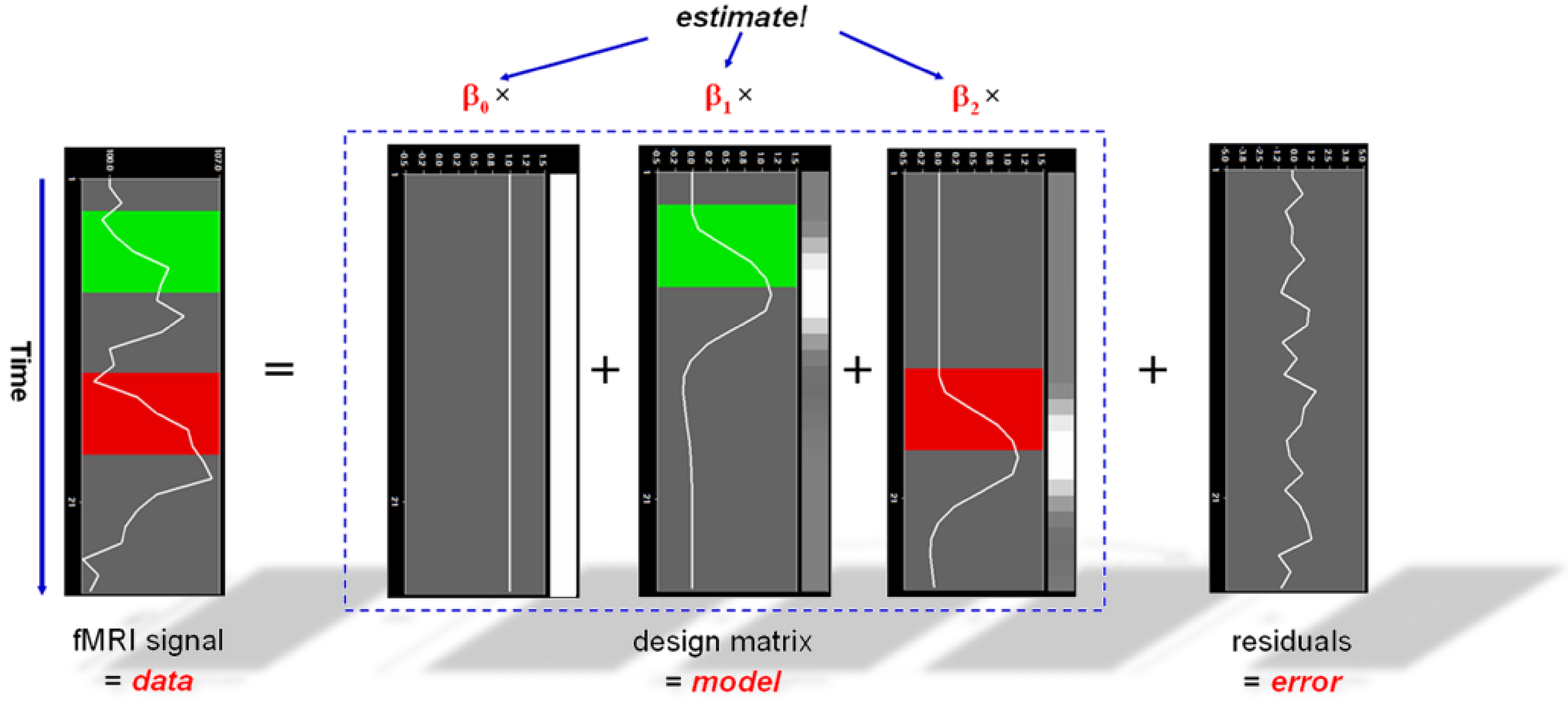
Univariate Statistical Data Analysis with GLM



Statistical Analysis – Background: Standard GLM



Standard GLM Analysis



Standard GLM Analysis

$$\begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & \cdots & \cdots & \cdots & X_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} & \cdots & \cdots & \cdots & X_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \vdots \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} e_1 \\ \vdots \\ \vdots \\ \vdots \\ e_n \end{bmatrix}$$

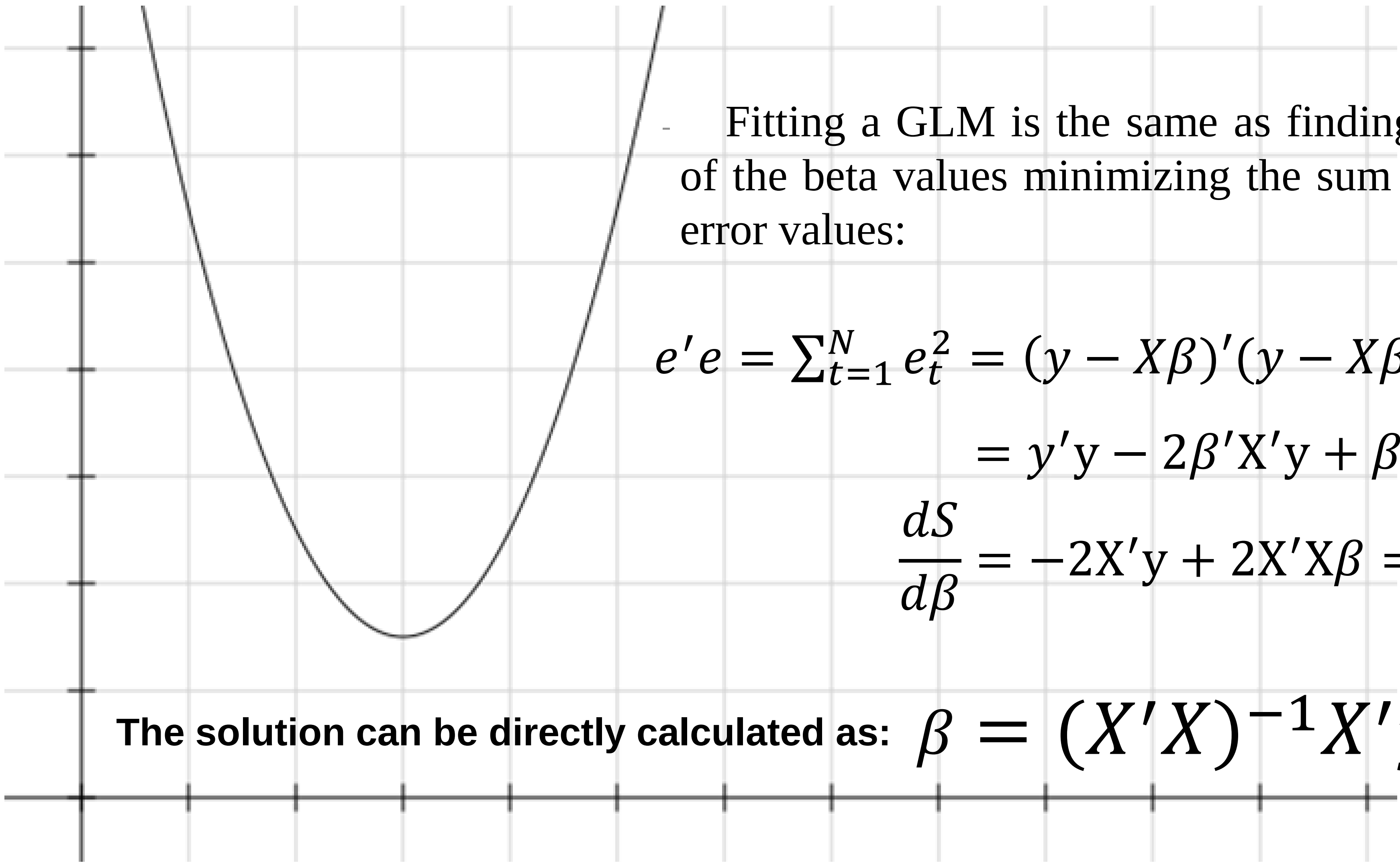
Observed fMRI signal: $y = X\beta + e$

- Fitting a GLM is the same as finding estimates of the beta values minimizing the sum of squared error values:

$$e'e = \sum_{t=1}^{N} e_t^2 = (y - X\beta)'(y - X\beta) \rightarrow \textbf{min}$$

The solution can be directly calculated as: $\beta = (X'X)^{-1}X'y$

Fitting a GLM – Standard Least Squares



- Fitting a GLM is the same as finding estimates of the beta values minimizing the sum of squared error values:

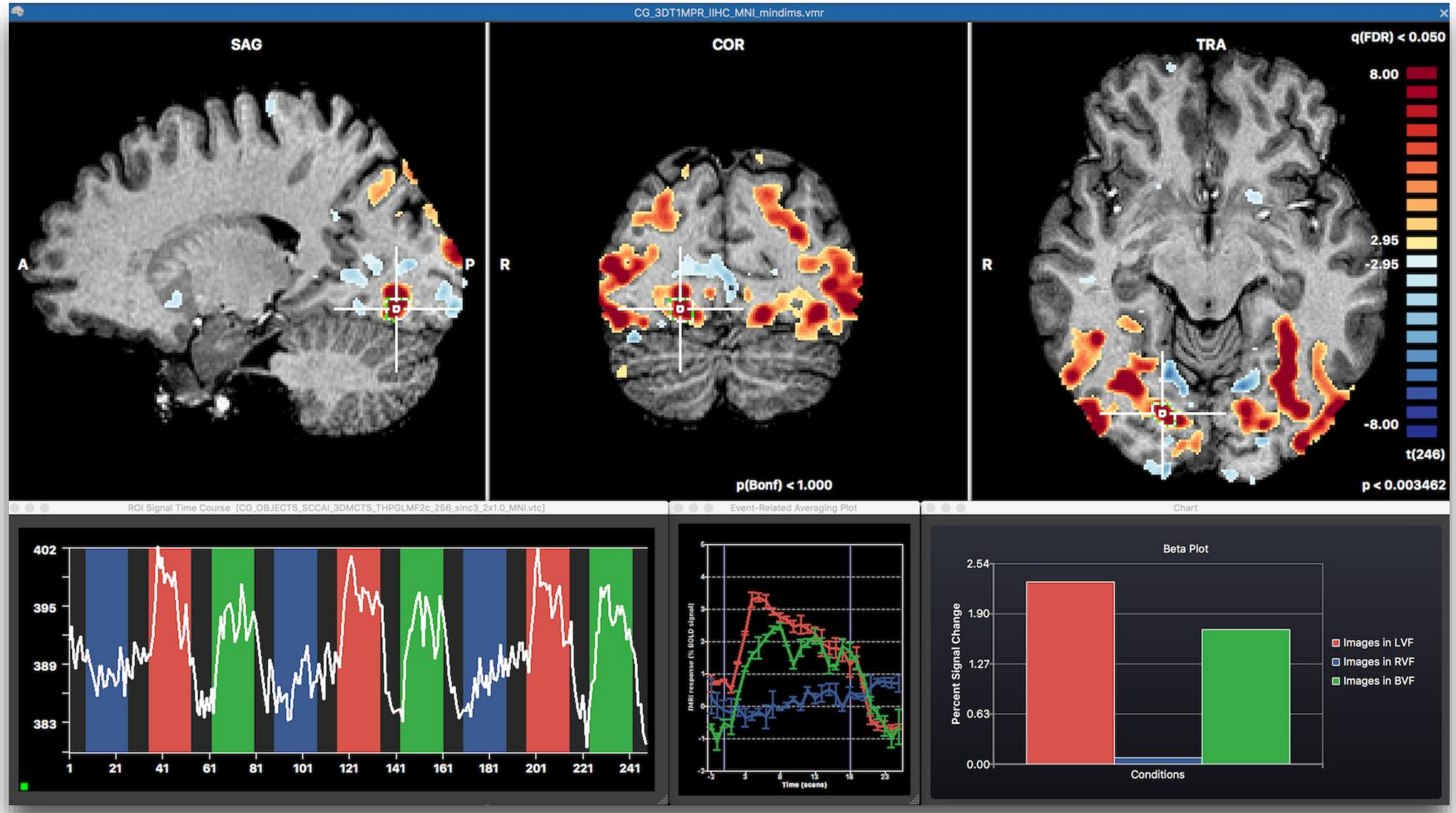
$$e'e = \sum_{t=1}^{N} e_t^2 = (y - X\beta)'(y - X\beta) \rightarrow \textbf{min}$$

$$= y'\mathrm{y} - 2\beta'\mathrm{X}'\mathrm{y} + \beta'\mathrm{X}'\mathrm{X}\beta$$

$$\frac{dS}{d\beta} = -2\mathrm{X}'\mathrm{y} + 2\mathrm{X}'\mathrm{X}\beta = 0$$

The solution can be directly calculated as: $\beta = (X'X)^{-1}X'y$

A GLM Contrast in Normalized (MNI) Space



Offline GLM analysis using BrainVoyager software

GLM Significance Test

- The variance of the fMRI signal time course is the sum of the covariance of the predicted values (model-related variance) and the variance of the residuals:

$$Var(y) = Var(\hat{y}) + Var(\mathrm{e})$$

- The square of the multiple correlation coefficient quantifies the amount of variance explained by the model:

$$R^2 = \frac{Var(\hat{y})}{Var(y)} = \frac{Var(\hat{y})}{Var(\hat{y}) + Var(e)}$$

- The amount of explained variance is statistically tested with an F test. Individual contrast – comparisons between beta values – are done using *t* tests:

$$F_{n-1,n-p} = \frac{R^2(n-p)}{(1-R^2)(p-1)} \qquad t = \frac{c'\beta}{\sqrt{Var(\mathrm{e})c'}(X'X)^{-1}c}$$

Incremental GLM: Recursive Least Squares

- The beta values and inverted $X'X$ matrix can be updated incrementally using only information of the new time point with the following recursive equations:

$$\beta_{t+1} = \beta_t + (X_t'X_t)^{-1} \frac{(y_{t+1} - x_{t+1}\beta_t)}{1 + x_{t+1}'(X_t'X_t)^{-1}x_{t+1}}$$

$$(X_{t+1}'X_{t+1})^{-1} = (X_t'X_t)^{-1} - \frac{(X_t'X_t)^{-1}x_{t+1}x_{t+1}'(X_t'X_t)^{-1}}{1 + x_{t+1}'(X_t'X_t)^{-1}x_{t+1}}$$

- Note: Since $(X'X)^{-1}$ term is the same for all voxels, it can be pre-computed before solving the β for individual voxels

Incremental GLM: Recursive Least Squares

- In its standard formulation, RLS GLMs result in the same beta estimates as a standard GLM over the whole time course up to the current point in time.

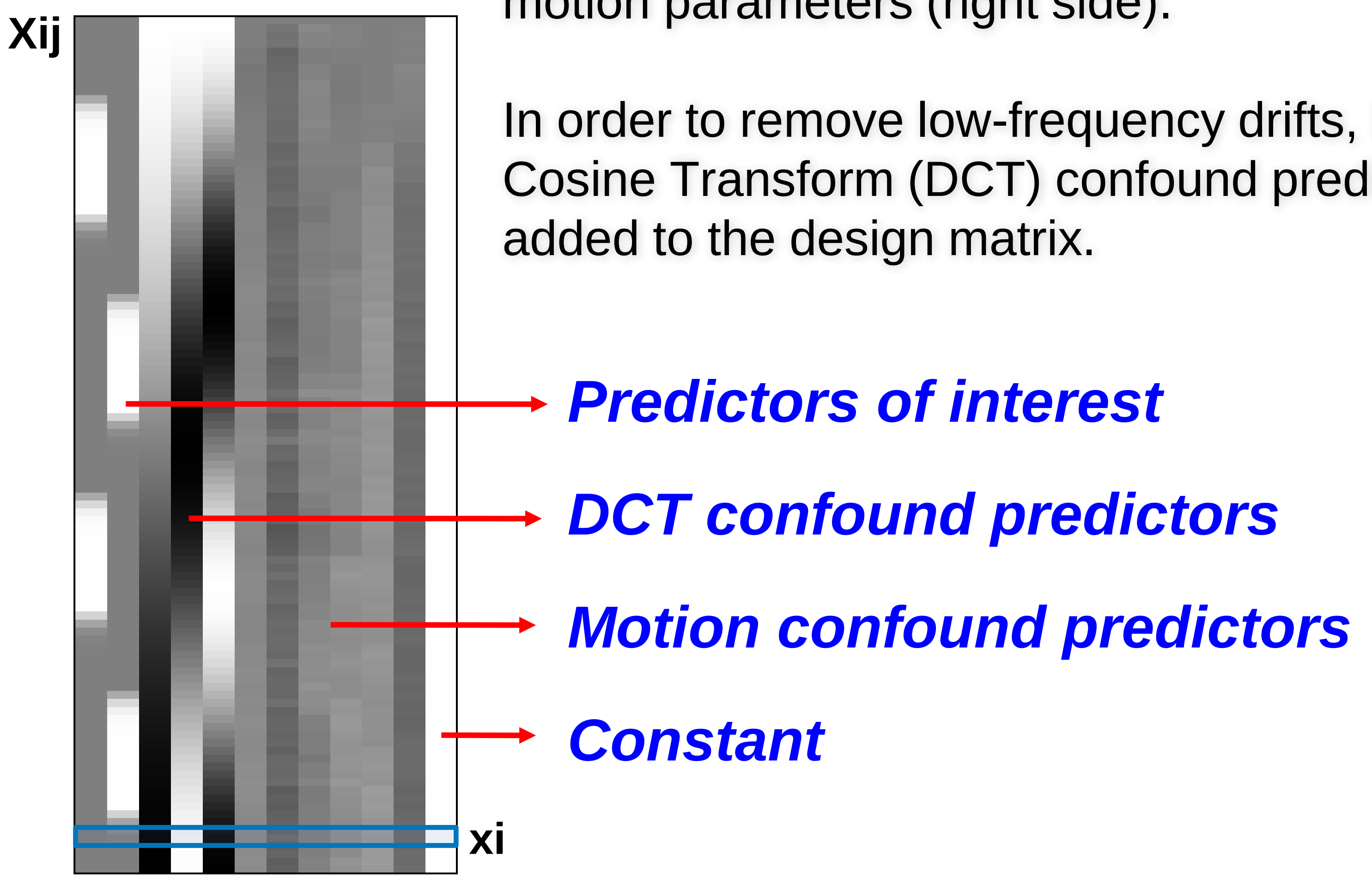
- With a slight modification, RLS can be used to weight past values exponentially or to run windowed GLMs (Pollock, 1999).

```
void RLS_GLM_Didactic()
{
    int y; // new value of current time point
    int k; // number of predictors
    double *beta = dvector(0,k-1); // betas to be estimated
    double **p = dmatrix(0,k-1, 0,k-1); // p -> invXX
    double *x = dvector(0,k-1); // row of design matrix of current time point
    int sign = 1; // standard or windowed RLS-GLM
    double lambda = 1.0; // 0 < lambda < 1 -> exponential weighting
    double *kappa = dvector(0,k-1);
    double f, *g = dvector(0,k-1);
    f = sign * lambda;

    for (i=0; i<k; i++)
    {
        g[i] = 0.0;
        for (j=0; j<k; j++)
            g[i] += p[i][j] * x[j];

        f += g[i] * x[i];
        y -= x[i] * beta[i];
    }

    for (i=0; i<k; i++)
    {
        kappa[i] = g[i] / f;
        beta[i] += kappa[i] * y;

        for (j=i; j<k; j++)
        {
            p[i][j] = (p[i][j] - kappa[i] * g[j]) / lambda;
            p[j][i] = p[i][j];  // invXX is symmetric
        }
    }
}
```

Dynamic design matrix

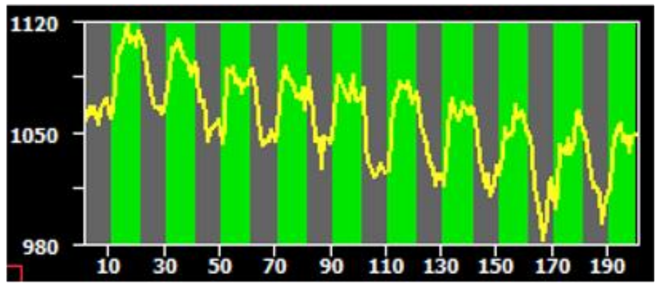
Design matrix is incrementally built and can incorporate real-time data, e.g. error trials and just computed 3D motion parameters (right side).

In order to remove low-frequency drifts, Discrete Cosine Transform (DCT) confound predictors can be added to the design matrix.

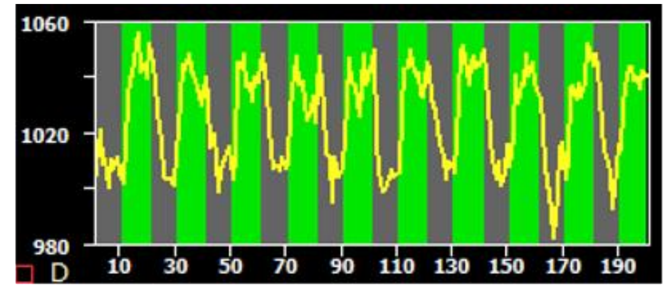


Obtaining Stationary Time Courses Using Counfound Predictors

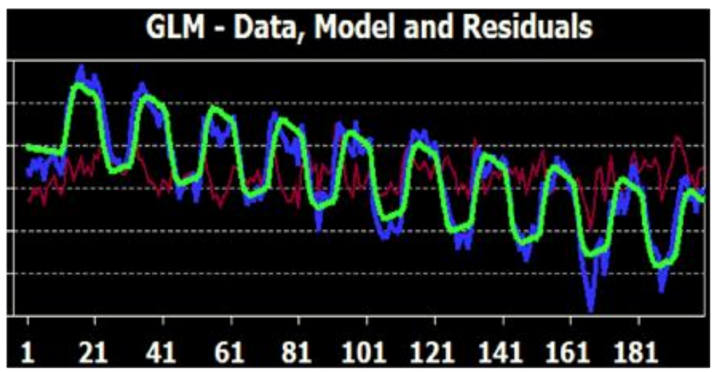
Original data time course



Detrended

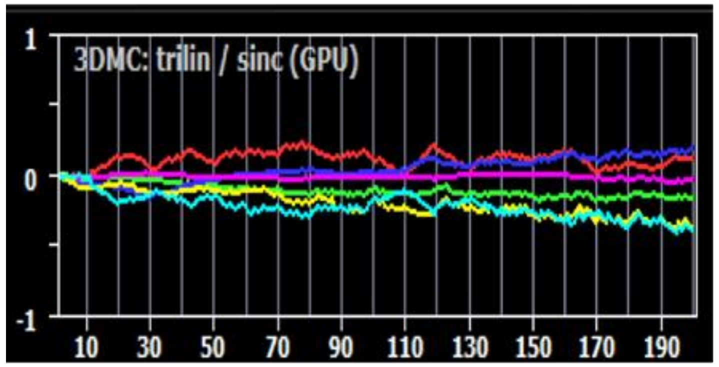


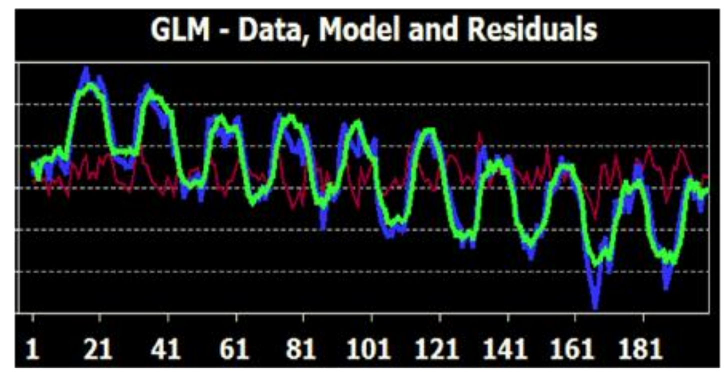
“Cleaned” data without confound effects



Confounds: Constant + Linear Trend

Predictor	beta	se	*t*	*p*
Hand Movement	34.950	1.347	25.943	0.000000
Constant	1041.105	0.960	1084.869	0.000000
Linear Trend	-33.303	1.210	-27.519	0.000000

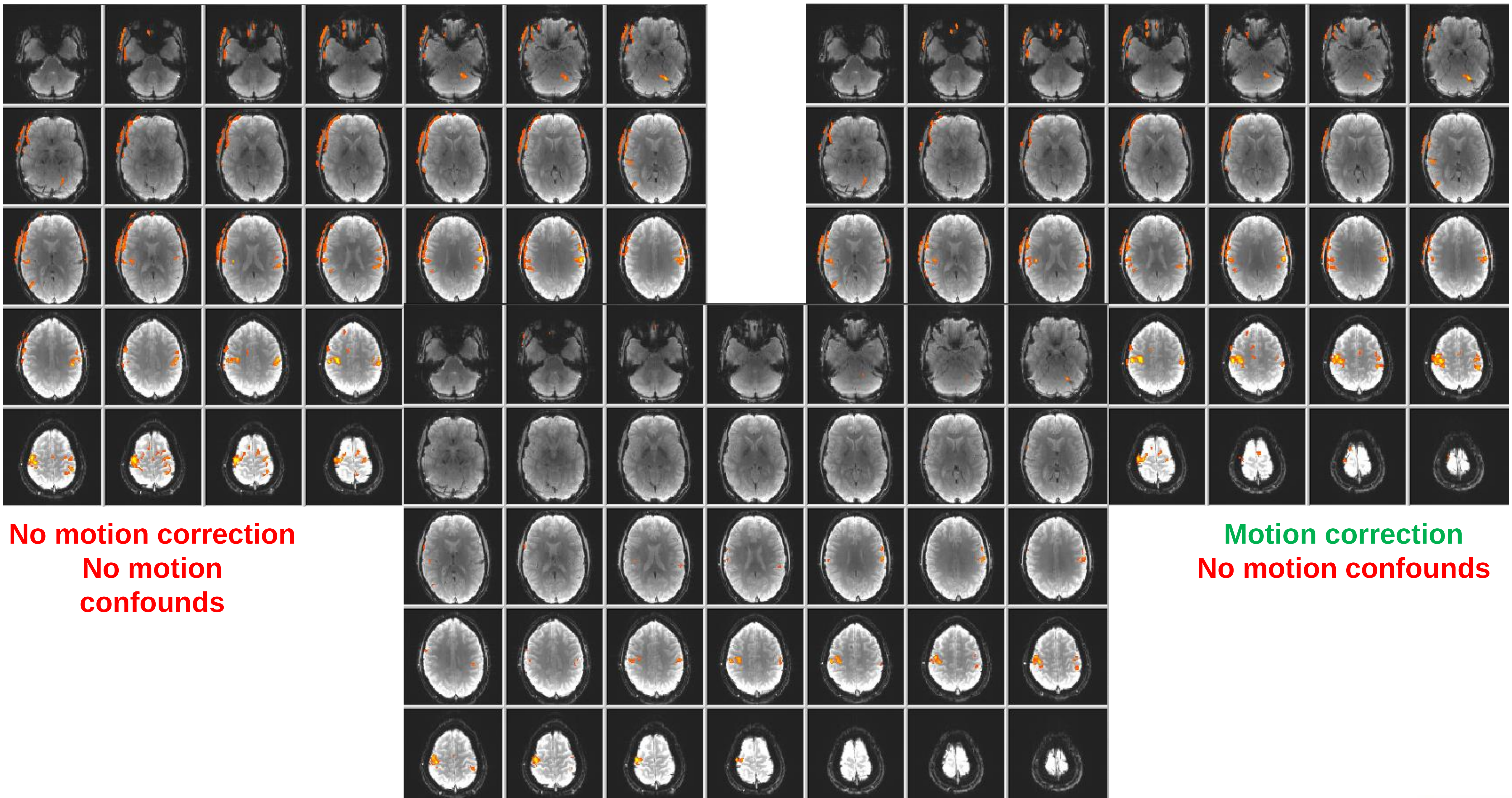




Confounds: Constant + Linear Trend + MC

Predictor	beta	se	*t*	*p*
Hand Movement	28.424	1.584	17.949	0.000000
Constant	1012.332	5.814	174.130	0.000000
Linear Trend	-52.254	6.265	-8.341	0.000000
3DMC Transl-X	68.125	22.349	3.048	0.002628
3DMC Transl-Y	196.934	43.251	4.553	0.000009
3DMC Transl-Z	14.828	26.664	0.556	0.578777
3DMC Rot-X	-192.846	30.052	-6.417	0.000000
3DMC Rot-Y	20.813	61.641	0.338	0.736000
3DMC Rot-Z	-27.250	20.941	-1.301	0.194726

Obtaining Stationary Time Courses Using Counfound Predictors



No motion correction
No motion
confounds

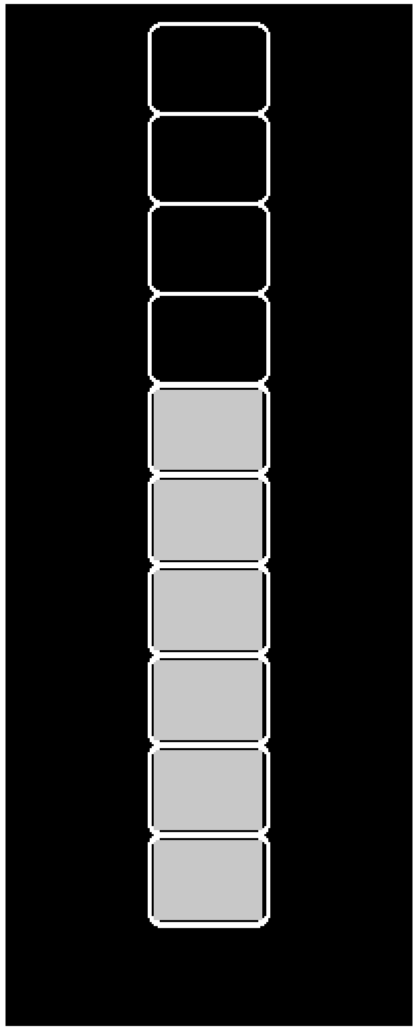
Motion correction
No motion confounds

Motion correction
Motion confounds

Feedback calculation

- Direct fMRI data:
  - feedback = (value - baseline) / baseline * 100

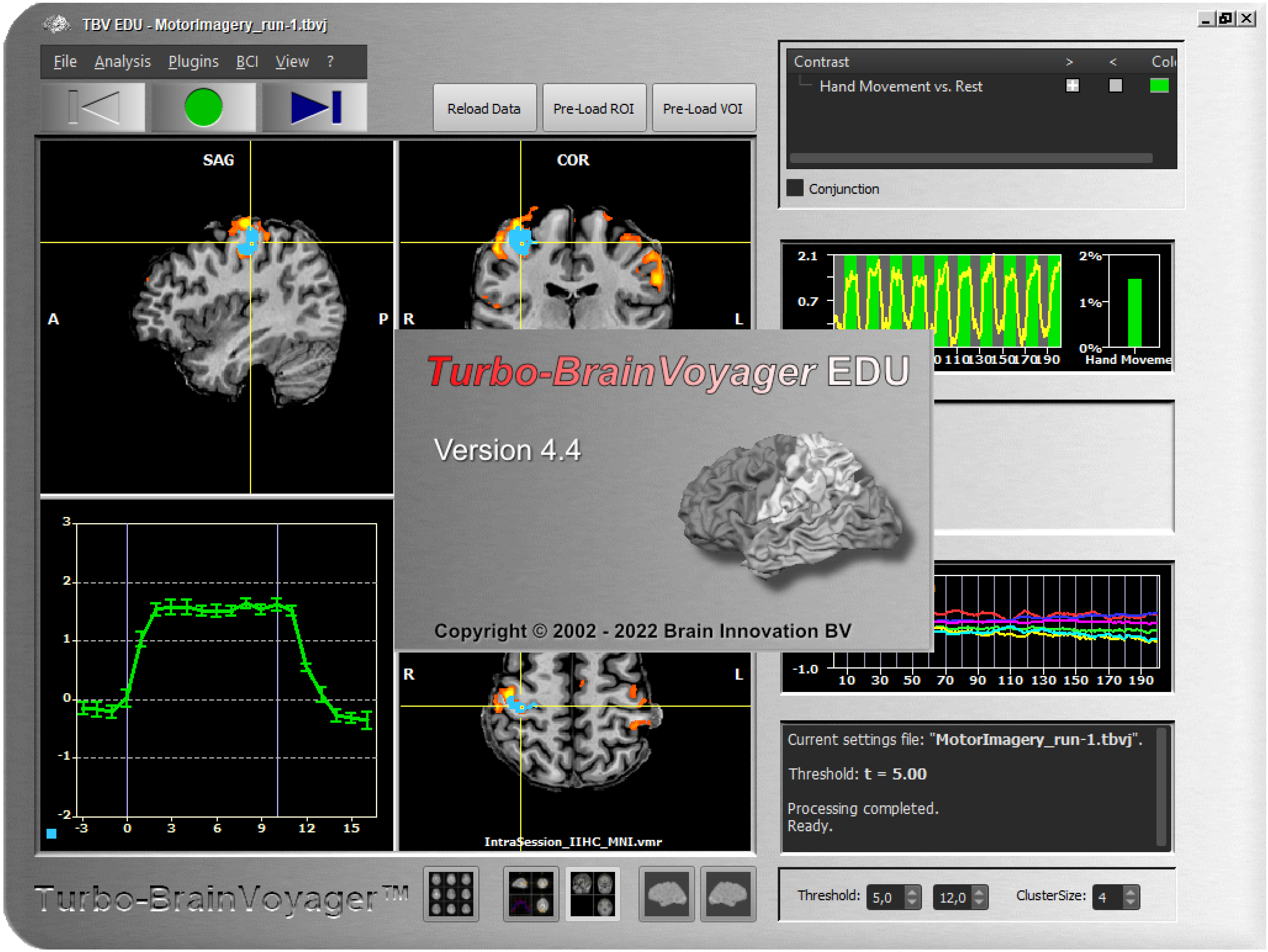
- Percent signal change (PSC) transformed:
  - feedback = (value - baseline)



- Transformation into activation value:
  - activity = (feedback / maximum_PSC)

2.
Hands-On

Hands-On using Turbo-BrainVoyager EDU



2.
Exporting information

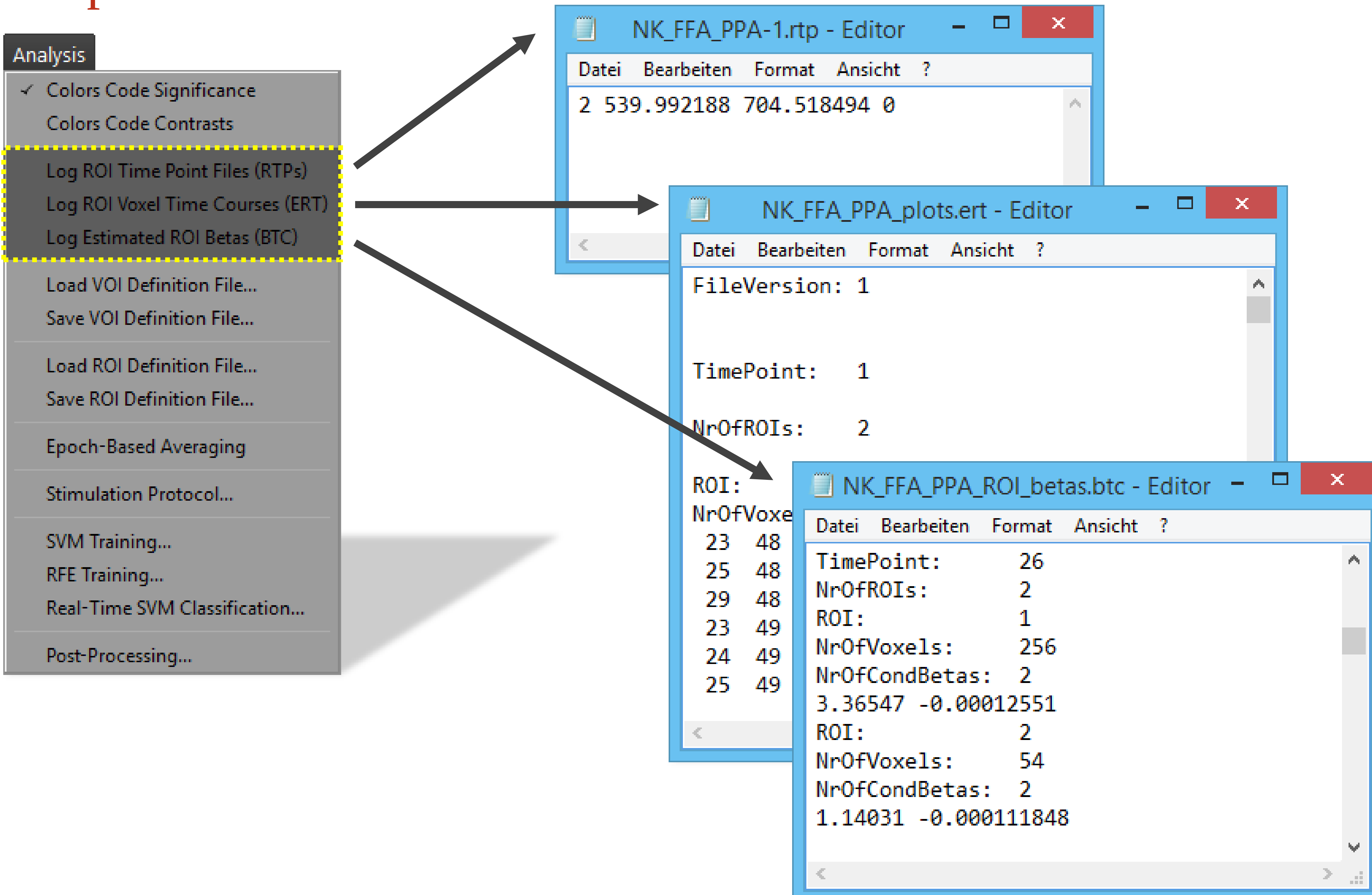
Exporting information

- Information that is not exported in real-time might not be possible to recover after the real-time session!

  – Examples:
    – Timing information
    – Responses from participant
    – Feedback signal
    – …

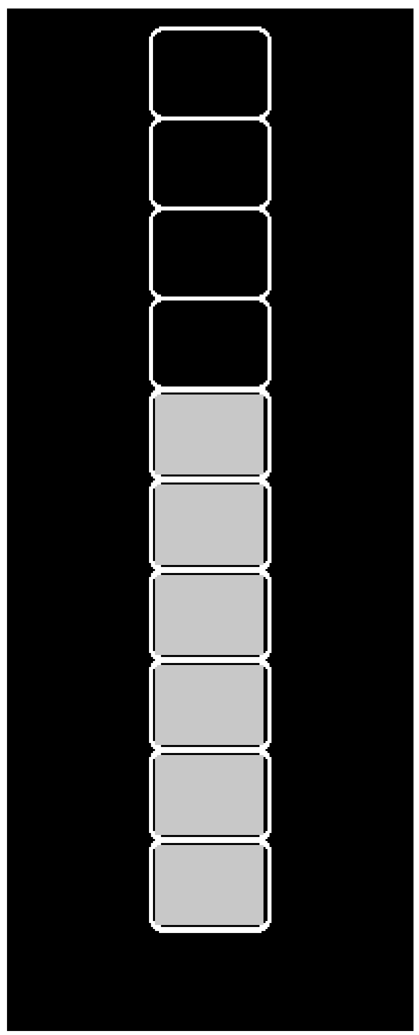
- Think about the analysis you want to perform before starting the study to ensure all information is available post-hoc!

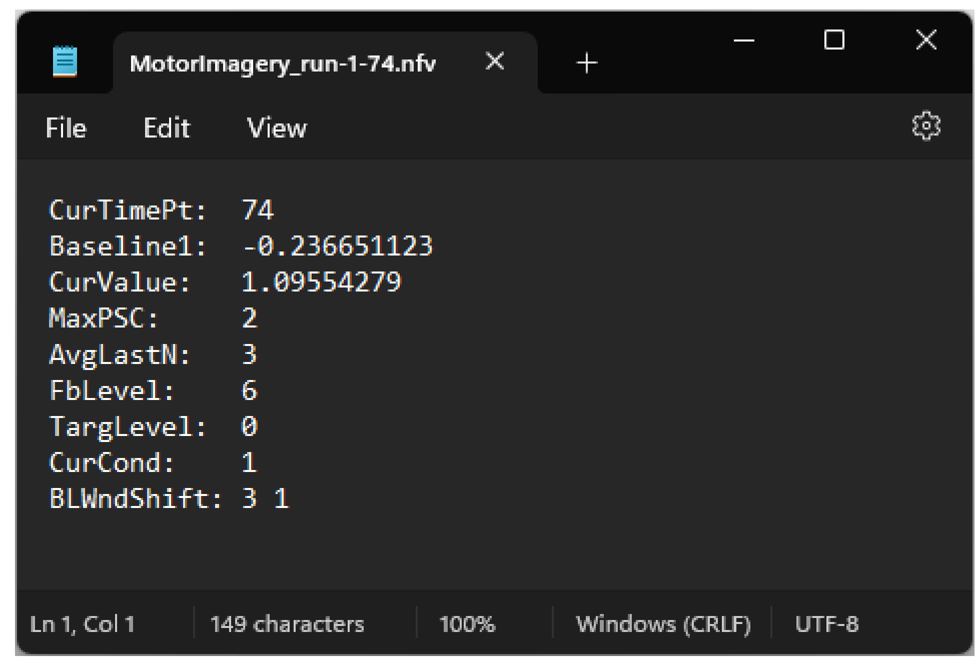
Export to files



Export neurofeedback parameters

- Think about all potential information you would need!



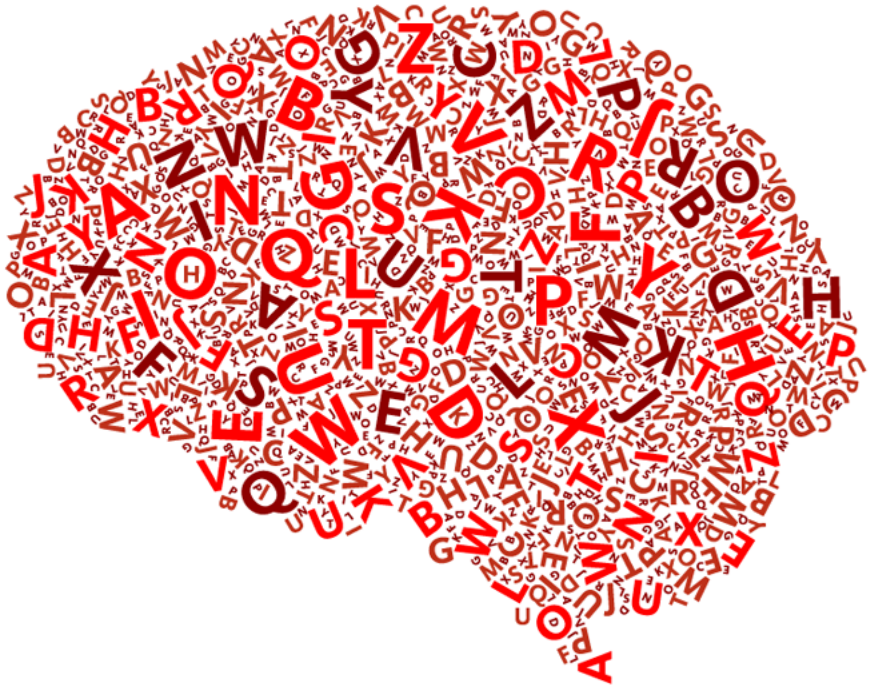


Ros T, Enriquez-Geppert S, Zotev V, et al. Consensus on the reporting and experimental design of clinical and cognitive-behavioural neurofeedback studies (CRED-nf checklist). *Brain*. 2020;143(6):1674-1685. doi:10.1093/brain/awaa009

Use the CRED-nf checklist for help

CRED-nf best practices checklist 2020			
Domain	**Item #**	**Checklist item**	**Reported on page #**
Pre-experiment			
	1a	Pre-register experimental protocol and planned analyses	
	1b	Justify sample size	
Control groups			
	2a	Employ control group(s) or control condition(s)	
	2b	When leveraging experimental designs where a double-blind is possible, use a double-blind	
	2c	Blind those who rate the outcomes, and when possible, the statisticians involved	
	2d	Examine to what extent participants and experimenters remain blinded	
	2e	In clinical efficacy studies, employ a standard-of-care intervention group as a benchmark for improvement	
Control measures			
	3a	Collect data on psychosocial factors	
	3b	Report whether participants were provided with a strategy	
	3c	Report the strategies participants used	
	3d	Report methods used for online-data processing and artefact correction	
	3e	Report condition and group effects for artefacts	
Feedback specifications			
	4a	Report how the online-feature extraction was defined	
	4b	Report and justify the reinforcement schedule	
	4c	Report the feedback modality and content	
	4d	Collect and report all brain activity variable(s) and/or contrasts used for feedback, as displayed to experimental participants	
	4e	Report the hardware and software used	
Outcome measures			
Brain	5a	Report neurofeedback regulation success based on the feedback signal	
	5b	Plot within-session and between-session regulation blocks of feedback variable(s), as well as pre-to-post resting baselines or contrasts	
	5c	Statistically compare the experimental condition/group to the control condition(s)/group(s) (not only each group to baseline measures)	
Behaviour	6a	Include measures of clinical or behavioural significance, defined *a priori*, and describe whether they were reached	
	6b	Run correlational analyses between regulation success and behavioural outcomes	
Data storage			
	7a	Upload all materials, analysis scripts, code, and raw data used for analyses, as well as final values, to an open access data repository, when feasible	

Ros T, Enriquez-Geppert S, Zotev V, et al. Consensus on the reporting and experimental design of clinical and cognitive-behavioural neurofeedback studies (CRED-nf checklist). ***Brain*****. 2020;143(6):1674-1685. doi:10.1093/brain/awaa009**

Questions?