# Intertwinning wavelets on graphs: a tool for the inverse problem in E/MEG ?

Joint work with
L. Avena (Univ. di Firenze), F. Castell (I2M, Marseille), A. Gaudillière (I2M, Marseille), C. Melot (I2M, Marseille) Spécials guests: Dominique Benielli and Sasha Duverger (Project Fosfor, CNRS initiative Open)

Workshop BMWs, June 2025

# Data on graphs

## Non oriented graph

$G = (\mathcal{X}, w)$,

# Data on graphs

## Non oriented graph

$G = (\mathcal{X}, w)$,
$\mathcal{X}$ set of vertices, $n = |\mathcal{X}|$ ;

# Data on graphs

## Non oriented graph

$G = (\mathcal{X}, w)$,
$\mathcal{X}$ set of vertices, $n = |\mathcal{X}|$ ;
$w : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$ weight function ;
For $(x, y) \in \mathcal{X} \times \mathcal{X}$, $(x, y)$ is an edge if and only if $w(x, y) > 0$.

# Data on graphs

## Non oriented graph

$G = (\mathcal{X}, w)$,
$\mathcal{X}$ set of vertices, $n = |\mathcal{X}|$ ;
$w : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$ weight function ;
For $(x, y) \in \mathcal{X} \times \mathcal{X}$, $(x, y)$ is an edge if and only if $w(x, y) > 0$.

## Assumption :

$$w(x, y) = w(y, x).$$

# Data on graphs

## Non oriented graph

$G = (\mathcal{X}, w)$,
$\mathcal{X}$ set of vertices, $n = |\mathcal{X}|$ ;
$w : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$ weight function ;
For $(x, y) \in \mathcal{X} \times \mathcal{X}$, $(x, y)$ is an edge if and only if $w(x, y) > 0$.

## Assumption :

$$w(x, y) = w(y, x).$$

## Laplacian of the graph $\mathcal{L}$

$$\mathcal{L}(x, y) = w(x, y) \text{ if } x \neq y$$

$w(x) := \sum_{y \neq x} w(x, y)$ .

$$\mathcal{L}(x, x) = -w(x).$$

# Data on graphs

## Non oriented graph

$G = (\mathcal{X}, w)$,
$\mathcal{X}$ set of vertices, $n = |\mathcal{X}|$ ;
$w : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$ weight function ;
For $(x, y) \in \mathcal{X} \times \mathcal{X}$, $(x, y)$ is an edge if and only if $w(x, y) > 0$.

## Assumption :

$$w(x, y) = w(y, x).$$

## Laplacian of the graph $\mathcal{L}$

$$\mathcal{L}(x, y) = w(x, y) \text{ if } x \neq y$$

$w(x) := \sum_{y \neq x} w(x, y)$ .

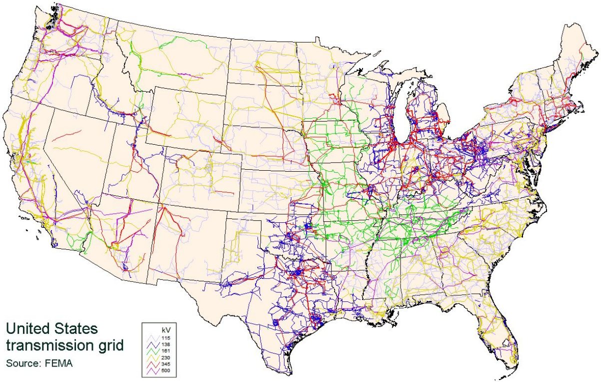
$$\mathcal{L}(x, x) = -w(x).$$

$(-\mathcal{L})$ is a positive symetric matrix with eigenvalues :

$$\lambda_0 = 0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}.$$

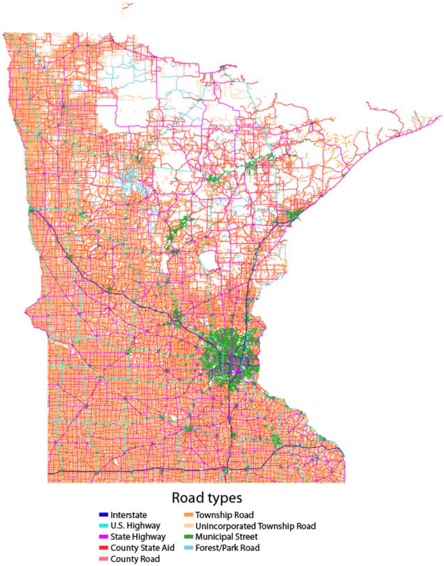
# Examples

## Electrical grid



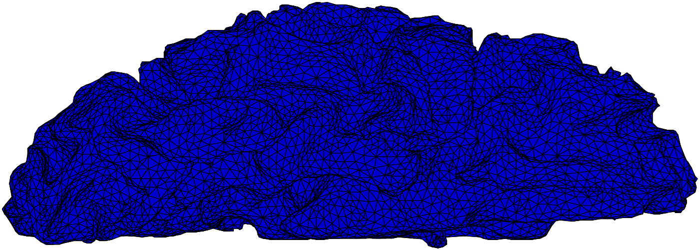
# Examples

## Street network



# Examples

Discretized surfaces (data J. Lefèvre, LIS Marseille)



# Sparsity of a signal

## Signal on graph

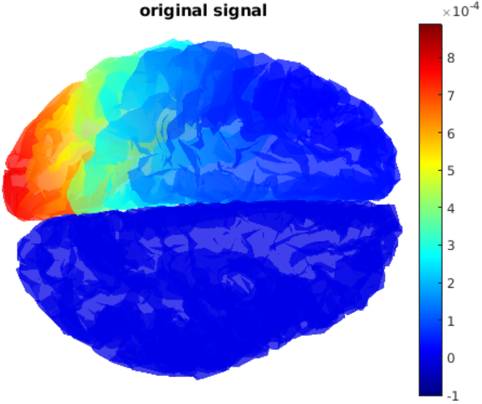
A signal on $G$ is a function $f : \mathcal{X} \to \mathbb{R}$ identified with a vector of $\mathbb{R}^n$.

# Sparsity of a signal

## Signal on graph

A signal on $G$ is a function $f : \mathcal{X} \to \mathbb{R}$ identified with a vector of $\mathbb{R}^n$.

A smooth signal (data J.M Lina, Université de Montréal)



# Sparsity of a signal

## Signal on graph

A signal on $G$ is a function $f : \mathcal{X} \to \mathbb{R}$ identified with a vector of $\mathbb{R}^n$.

## Aim.

- We want to build a multiresolution analysis of signals defined on a generic graph, i.e. to encode $f \in \mathbb{R}^n$ as a sum of a general trend, the approximation, and oscillations at different scales, the details :
  our signal $f$ is encoded through $n$ coefficients structured as
  $$\Big[ \underbrace{f_k}_{\text{approximation}}, \quad \underbrace{g_1, \cdots, g_k}_{\text{details}} \Big].$$

# Sparsity of a signal

## Signal on graph

A signal on $G$ is a function $f : \mathcal{X} \to \mathbb{R}$ identified with a vector of $\mathbb{R}^n$.

## Aim.

- We want to build a multiresolution analysis of signals defined on a generic graph, i.e. to encode $f \in \mathbb{R}^n$ as a sum of a general trend, the approximation, and oscillations at different scales, the details :
  our signal $f$ is encoded through $n$ coefficients structured as
  $$[\underbrace{f_k}_{\text{approximation}}, \quad \underbrace{g_1, \cdots, g_k}_{\text{details}}].$$
- we would like $[g_1, \cdots, g_k]$ to be a sparse vector whenever $f$ has some "regularity".

## Classical scheme on $\mathbb{R}^n$.

An iterative algorithm made of filtering and subsampling.

## Classical scheme on $\mathbb{R}^n$.

An iterative algorithm made of filtering and subsampling.

- Start with $f_0 = f$.

## Classical scheme on $\mathbb{R}^n$.

An iterative algorithm made of filtering and subsampling.

- Start with $f_0 = f$.
- Step 1 : Approximation and detail at scale 1.

## Classical scheme on $\mathbb{R}^n$.

An iterative algorithm made of filtering and subsampling.

- Start with $f_0 = f$.
- Step 1 : Approximation and detail at scale 1.
  - Filtering to separate low and high frequency parts of the signal.
    local "mean" : $\tilde{f}_1(m) = (h \star f_0)(m)$ for the approximation.
    local "gradient" : $\tilde{g}_1(m) = (g \star f_0)(m)$ for the detail.

$f_0 \in \mathbb{R}^n$ is coded by $(\tilde{f}_1, \tilde{g}_1) \in \mathbb{R}^n \times \mathbb{R}^n$.

# Classical scheme on $\mathbb{R}^n$.

An iterative algorithm made of filtering and subsampling.

- Start with $f_0 = f$.
- Step 1 : Approximation and detail at scale 1.
  - Filtering to separate low and high frequency parts of the signal.
    local "mean" : $\tilde{f}_1(m) = (h \star f_0)(m)$ for the approximation.
    local "gradient" : $\tilde{g}_1(m) = (g \star f_0)(m)$ for the detail.

    $$f_0 \in \mathbb{R}^n \text{ is coded by } (\tilde{f}_1, \tilde{g}_1) \in \mathbb{R}^n \times \mathbb{R}^n.$$
  - Subsampling by keeping one out of two points : $f_1(m) = \tilde{f}_1(2m)$, $g_1(m) = \tilde{g}_1(2m)$.

    $$f_0 \in \mathbb{R}^n \text{ is coded by } (f_1, g_1) \in \mathbb{R}^{\frac{n}{2}} \times \mathbb{R}^{\frac{n}{2}}.$$

## Classical scheme on $\mathbb{R}^n$.

An iterative algorithm made of filtering and subsampling.

- Start with $f_0 = f$.
- Step 1 : Approximation and detail at scale 1.
  - Filtering to separate low and high frequency parts of the signal.
    local "mean" : $\tilde{f}_1(m) = (h \star f_0)(m)$ for the approximation.
    local "gradient" : $\tilde{g}_1(m) = (g \star f_0)(m)$ for the detail.

    $$f_0 \in \mathbb{R}^n \text{ is coded by } (\tilde{f}_1, \tilde{g}_1) \in \mathbb{R}^n \times \mathbb{R}^n.$$
  - Subsampling by keeping one out of two points : $f_1(m) = \tilde{f}_1(2m)$, $g_1(m) = \tilde{g}_1(2m)$.

    $$f_0 \in \mathbb{R}^n \text{ is coded by } (f_1, g_1) \in \mathbb{R}^{\frac{n}{2}} \times \mathbb{R}^{\frac{n}{2}}.$$
  - $h$ and $g$ are well chosen to have perfect reconstruction of $f$ from $(f_1, g_1)$

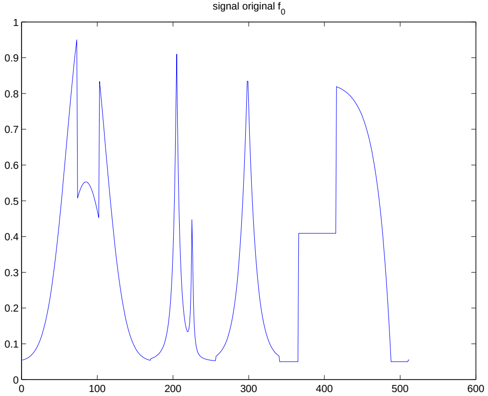
# Classical scheme on $\mathbb{R}^n$.

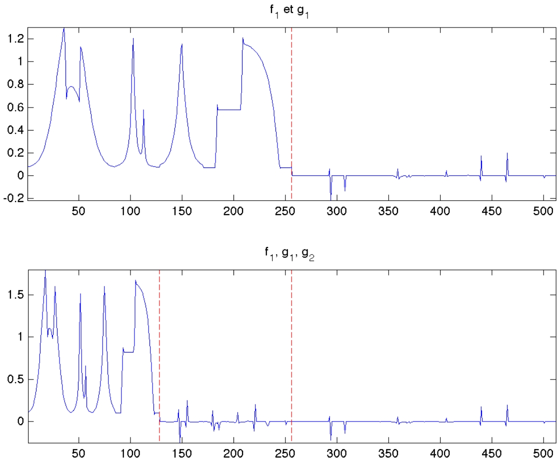
An iterative algorithm made of filtering and subsampling.

- Start with $f_0 = f$.
- Step 1 : Approximation and detail at scale 1.
  - Filtering to separate low and high frequency parts of the signal.
    local "mean" : $\tilde{f}_1(m) = (h \star f_0)(m)$ for the approximation.
    local "gradient" : $\tilde{g}_1(m) = (g \star f_0)(m)$ for the detail.
    $$f_0 \in \mathbb{R}^n \text{ is coded by } (\tilde{f}_1, \tilde{g}_1) \in \mathbb{R}^n \times \mathbb{R}^n.$$
  - Subsampling by keeping one out of two points : $f_1(m) = \tilde{f}_1(2m)$, $g_1(m) = \tilde{g}_1(2m)$.
    $$f_0 \in \mathbb{R}^n \text{ is coded by } (f_1, g_1) \in \mathbb{R}^{\frac{n}{2}} \times \mathbb{R}^{\frac{n}{2}}.$$
  - $h$ and $g$ are well chosen to have perfect reconstruction of $f$ from $(f_1, g_1)$
- Iterate.

$$\begin{array}{ccccccccc}
f_0(\text{size } n) & \to & f_1(\text{size } n/2) & \to & f_2(\text{size } n/2^2) & \dots & \to & f_k(\text{size } n/2^k) \\
 & \searrow & & \searrow & & \dots & \searrow & \\
 & & g_1(\text{size } n/2) & & g_2(\text{size } n/2^2) & \dots & & g_k(\text{size } n/2^k)
\end{array}$$

# Example


signal original $f_0$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
100
200
300
400
500
600




one step or two steps of the scheme

## Basis functions

One step $\Longleftrightarrow$ definition of a basis of $\mathbb{R}^n$ :

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

## Basis functions

One step $\Longleftrightarrow$ definition of a basis of $\mathbb{R}^n$ :

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

- Approximation coefficients :

$$f_1(m) =< \phi_m, f_0 >;$$

# Basis functions

One step $\Longleftrightarrow$ definition of a basis of $\mathbb{R}^n$ :

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

- Approximation coefficients :

  $f_1(m) =< \phi_m, f_0 >$;

- Detail coefficients :

  $g_1(m) =< \psi_m, f_0 >$;

# Basis functions

One step $\Longleftrightarrow$ definition of a basis of $\mathbb{R}^n$ :

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

- Approximation coefficients :
  $$f_1(m) =< \phi_m, f_0 > ;$$
- Detail coefficients :
  $$g_1(m) =< \psi_m, f_0 > ;$$

Needed properties to generalize the scheme :

we want to find basis functions with good

## Basis functions

One step $\Longleftrightarrow$ definition of a basis of $\mathbb{R}^n$ :

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

- Approximation coefficients :
$$f_1(m) = < \phi_m, f_0 > ;$$
- Detail coefficients :
$$g_1(m) = < \psi_m, f_0 > ;$$

Needed properties to generalize the scheme :

we want to find basis functions with good

- Localization in space ;

## Basis functions

**One step $\Longleftrightarrow$ definition of a basis of $\mathbb{R}^n$ :**

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

- Approximation coefficients :
$$f_1(m) = < \phi_m, f_0 >;$$
- Detail coefficients :
$$g_1(m) = < \psi_m, f_0 >;$$

**Needed properties to generalize the scheme :**

we want to find basis functions with good

- Localization in space ;
- Localization in frequency ;

## Basis functions

**One step $\iff$ definition of a basis of $\mathbb{R}^n$ :**

$\{\phi_m, 0 \leq m \leq \frac{n}{2}\} \bigcup \{\psi_m, 0 \leq m \leq \frac{n}{2}\}$

- Approximation coefficients :
  $$f_1(m) =< \phi_m, f_0 >;$$
- Detail coefficients :
  $$g_1(m) =< \psi_m, f_0 >;$$

**Needed properties to generalize the scheme :**

we want to find basis functions with good

- Localization in space ;
- Localization in frequency ;
- Conditioning index, to be able to reconstruct signal from the coefficients.

# Multiresolution scheme

## Iterating using the same scheme

$$\begin{array}{ccccccccc}
f_0(\text{size } n) & \rightarrow & f_1(\text{size } n/2) & \rightarrow & f_2(\text{size } n/2^2) & \dots & \rightarrow & f_k(\text{size } n/2^k) \\
& \searrow & & \searrow & & \dots & \searrow & \\
& & g_1(\text{size } n/2) & & g_2(\text{size } n/2^2) & \dots & & g_k(\text{size } n/2^k)
\end{array}$$

# Multiresolution scheme

## Iterating using the same scheme

$$\begin{array}{ccccccccc}
f_0(\text{size } n) & \rightarrow & f_1(\text{size } n/2) & \rightarrow & f_2(\text{size } n/2^2) & \ldots & \rightarrow & f_k(\text{size } n/2^k) \\
& \searrow & & \searrow & & \ldots & \searrow & \\
& & g_1(\text{size } n/2) & & g_2(\text{size } n/2^2) & \ldots & & g_k(\text{size } n/2^k)
\end{array}$$

## Needed properties to generalize on a graph :

# Multiresolution scheme

## Iterating using the same scheme

$$
\begin{array}{ccccccccc}
f_0(\text{size } n) & \rightarrow & f_1(\text{size } n/2) & \rightarrow & f_2(\text{size } n/2^2) & \dots & \rightarrow & f_k(\text{size } n/2^k) \\
 & \searrow & & \searrow & & \dots & \searrow & \\
 & & g_1(\text{size } n/2) & & g_2(\text{size } n/2^2) & \dots & & g_k(\text{size } n/2^k)
\end{array}
$$

## Needed properties to generalize on a graph :

- A sequence of subgraphs on which the approximation will be decomposed

$$
\begin{array}{ccccccccc}
f_0 \text{ (on } G_0 = G) & \rightarrow & f_1 \text{ (on } G_1) & \rightarrow & f_2 \text{ (on } G_2) & \dots & \rightarrow & f_k \text{ (on } G_k) \\
 & \searrow & & \searrow & & \dots & \searrow & \\
 & & g_1 & & g_2 & \dots & & g_k
\end{array}
$$

# Multiresolution scheme

## Iterating using the same scheme

$$\begin{array}{ccccccccc} f_0(\text{size } n) & \rightarrow & f_1(\text{size } n/2) & \rightarrow & f_2(\text{size } n/2^2) & \dots & \rightarrow & f_k(\text{size } n/2^k) \\ & \searrow & & \searrow & & \dots & \searrow & \\ & & g_1(\text{size } n/2) & & g_2(\text{size } n/2^2) & \dots & & g_k(\text{size } n/2^k) \end{array}$$

## Needed properties to generalize on a graph :

- A sequence of subgraphs on which the approximation will be decomposed

$$\begin{array}{ccccccccc} f_0 \text{ (on } G_0 = G) & \rightarrow & f_1 \text{ (on } G_1) & \rightarrow & f_2 \text{ (on } G_2) & \dots & \rightarrow & f_k \text{ (on } G_k) \\ & \searrow & & \searrow & & \dots & \searrow & \\ & & g_1 & & g_2 & \dots & & g_k \end{array}$$

- We want $G_1$, $G_2$, $\dots G_k$ to be a sequence of subgraphs which keep as much as possible the important features of the original graph.

## Generalization on graphs

**Some questions :**

## Generalization on graphs

**Some questions :**

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?

## Generalization on graphs

**Some questions :**

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?
- Graph structure problem : how do we choose the new graph on which we iterate the scheme ?

## Generalization on graphs

**Some questions :**

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?
- Graph structure problem : how do we choose the new graph on which we iterate the scheme ?
- Filtering problem.

## Generalization on graphs

**Some questions :**

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?
- Graph structure problem : how do we choose the new graph on which we iterate the scheme ?
- Filtering problem.

**Proposed answers to these questions :**

## Generalization on graphs

**Some questions :**

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?
- Graph structure problem : how do we choose the new graph on which we iterate the scheme ?
- Filtering problem.

**Proposed answers to these questions :**

$\rightarrow$ Subsampling a graph : find a random set with garanties that in some sense it is well spread on the graph

# Generalization on graphs

**Some questions :**

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?
- Graph structure problem : how do we choose the new graph on which we iterate the scheme ?
- Filtering problem.

**Proposed answers to these questions :**

$\rightarrow$ Subsampling a graph : find a random set with garanties that in some sense it is well spread on the graph

$\rightarrow$ Filtering the data : compute local means.

# Generalization on graphs

## Some questions :

- Sampling problem : How can we sample "one out of two points" ? How do we choose a "well spread" subset $\bar{\mathcal{X}}$ of $\mathcal{X}$ ?
- Graph structure problem : how do we choose the new graph on which we iterate the scheme ?
- Filtering problem.

## Proposed answers to these questions :

$\rightarrow$ Subsampling a graph : find a random set with garanties that in some sense it is well spread on the graph

$\rightarrow$ Filtering the data : compute local means.

$\rightarrow$ Compute the weights between the points of the subsampling set, which means compute the coefficients of a Laplacian matrix based on the subsampling set.

## A non exhaustive bibliography

## A non exhaustive bibliography

- Coifman et Maggioni : diffusion wavelets (2006).

## A non exhaustive bibliography

- Coifman et Maggioni : diffusion wavelets (2006).
- Hammond, Vandergheynst et Gribonval (2010) : a wavelet frame as a discretization of a continuous wavelet transform.

## A non exhaustive bibliography

- Coifman et Maggioni : diffusion wavelets (2006).
- Hammond, Vandergheynst et Gribonval (2010) : a wavelet frame as a discretization of a continuous wavelet transform.
- Ortega : multiresolution analysis on bipartite graphs (2012)

## A non exhaustive bibliography

- Coifman et Maggioni : diffusion wavelets (2006).
- Hammond, Vandergheynst et Gribonval (2010) : a wavelet frame as a discretization of a continuous wavelet transform.
- Ortega : multiresolution analysis on bipartite graphs (2012)
- Schuman, Faraji, Vandergheynst (2016) : Fielder vector and Kron reduction.

## A non exhaustive bibliography

- Coifman et Maggioni : diffusion wavelets (2006).
- Hammond, Vandergheynst et Gribonval (2010) : a wavelet frame as a discretization of a continuous wavelet transform.
- Ortega : multiresolution analysis on bipartite graphs (2012)
- Schuman, Faraji, Vandergheynst (2016) : Fielder vector and Kron reduction.
- Tremblay, Borgnat (2016) : Haar filters generalized to graphs

## A non exhaustive bibliography

- Coifman et Maggioni : diffusion wavelets (2006).
- Hammond, Vandergheynst et Gribonval (2010) : a wavelet frame as a discretization of a continuous wavelet transform.
- Ortega : multiresolution analysis on bipartite graphs (2012)
- Schuman, Faraji, Vandergheynst (2016) : Fielder vector and Kron reduction.
- Tremblay, Borgnat (2016) : Haar filters generalized to graphs
- ...

# Setting

## Markov process in continuous time

We have

# Setting

## Markov process in continuous time

We have

- a Laplacian matrix. $\mathcal{L}f(x) = \sum_{y \in V} w(x,y)\,(f(y) - f(x))$ for any vector $(f(x))_{x \in \mathcal{X}}$.

# Setting

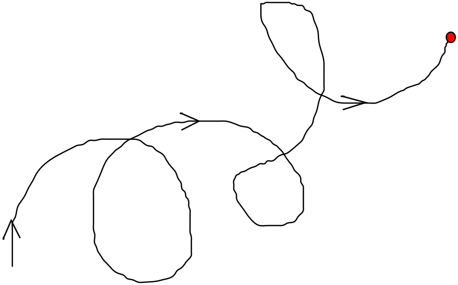
## Markov process in continuous time

We have

- a Laplacian matrix. $\mathcal{L}f(x) = \sum_{y \in V} w(x,y)\,(f(y) - f(x))$ for any vector $(f(x))_{x\in\mathcal{X}}$.
- We denote $X = (X_t, t \geq 0)$ a Markov process with generator $\mathcal{L}$ : $X$ jumps from $x$ to $y$ with probability $w(x,y)/w(x)$ after a random time of law $\mathcal{E}(w(x))$ .

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

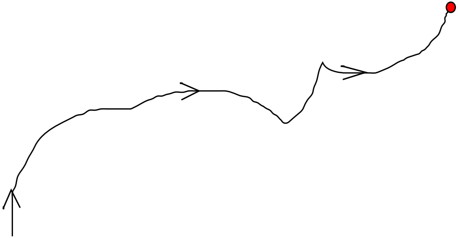
Wilson's algorithm :



- Choose a point $x \in \mathcal{X}$. From $x$, run a Markov process $X$ with generator $\mathcal{L}$ until a random time $T_q$, an exponential time with parameter $q$.

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

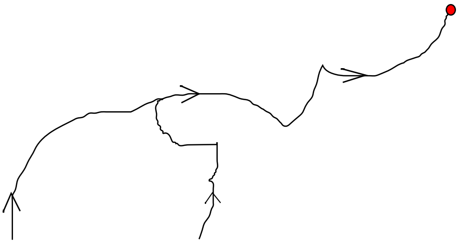
Wilson's algorithm :



- Choose a point $x \in \mathcal{X}$. From $x$, run a Markov process $X$ with generator $\mathcal{L}$ until a random time $T_q$, an exponential time with parameter $q$.
- Erase the loops.

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

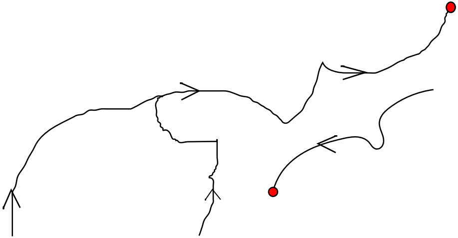
Wilson's algorithm :



- Choose a point $x \in \mathcal{X}$. From $x$, run a Markov process $X$ with generator $\mathcal{L}$ until a random time $T_q$, an exponential time with parameter $q$.
- Erase the loops.
- Choose a point outside the drawn trajectory. Run $X$ and stop it at an exponential time with parameter $q$, or when it reaches the drawn trajectory. Erase the loops.

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

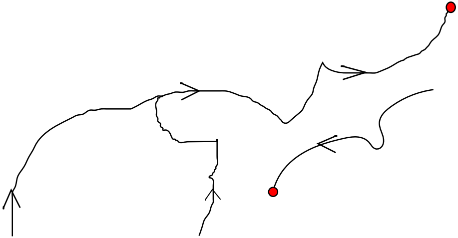
Wilson's algorithm :



- Choose a point $x \in \mathcal{X}$. From $x$, run a Markov process $X$ with generator $\mathcal{L}$ until a random time $T_q$, an exponential time with parameter $q$.
- Erase the loops.
- Choose a point outside the drawn trajectory. Run $X$ and stop it at an exponential time with parameter $q$, or when it reaches the drawn trajectory. Erase the loops.
- Go on until exhaustion of $\mathcal{X}$.

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

Wilson's algorithm :

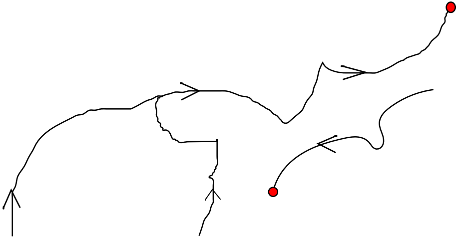


- Choose a point $x \in \mathcal{X}$. From $x$, run a Markov process $X$ with generator $\mathcal{L}$ until a random time $T_q$, an exponential time with parameter $q$.
- Erase the loops.
- Choose a point outside the drawn trajectory. Run $X$ and stop it at an exponential time with parameter $q$, or when it reaches the drawn trajectory. Erase the loops.
- Go on until exhaustion of $\mathcal{X}$.

We end up with an oriented spanning forest $\Phi_q$.

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

Wilson's algorithm :



- Choose a point $x \in \mathcal{X}$. From $x$, run a Markov process $X$ with generator $\mathcal{L}$ until a random time $T_q$, an exponential time with parameter $q$.
- Erase the loops.
- Choose a point outside the drawn trajectory. Run $X$ and stop it at an exponential time with parameter $q$, or when it reaches the drawn trajectory. Erase the loops.
- Go on until exhaustion of $\mathcal{X}$.

We end up with an oriented spanning forest $\Phi_q$.

**Our proposal to choose "one out of two points".**

$$\bar{\mathcal{X}} = \text{set of the trees roots} = \rho(\Phi_q).$$

## Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

### Properties of this set (Wilson (96)).

Let $(X(t), t \geq 0)$ be a Markov process with generator $\mathcal{L}$ and $T_q \sim \mathcal{E}(q)$. Set

$$K_q(x, y) = q(q\mathrm{Id} - \mathcal{L})^{-1}(x, y) = P_x\left[X(T_q) = y\right].$$

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

## Properties of this set (Wilson (96)).

Let $(X(t), t \geq 0)$ be a Markov process with generator $\mathcal{L}$ and $T_q \sim \mathcal{E}(q)$. Set

$$K_q(x,y) = q(q\mathrm{Id} - \mathcal{L})^{-1}(x,y) = P_x\left[X(T_q) = y\right].$$

- $\rho(\Phi_q)$ is a determinantal process with kernel $K_q$ :
$$\forall A \subset \mathcal{X},\ \mathbb{P}\left[A \subset \rho(\Phi_q)\right] = \det{}_A(K_q).$$

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

## Properties of this set (Wilson (96)).

Let $(X(t), t \geq 0)$ be a Markov process with generator $\mathcal{L}$ and $T_q \sim \mathcal{E}(q)$. Set

$$K_q(x,y) = q(q\mathrm{Id} - \mathcal{L})^{-1}(x,y) = P_x\left[X(T_q) = y\right].$$

- $\rho(\Phi_q)$ is a determinantal process with kernel $K_q$ :
$$\forall A \subset \mathcal{X},\ \mathbb{P}\left[A \subset \rho(\Phi_q)\right] = \det_A(K_q).$$
- $|\rho(\Phi_q)|$ is distributed as the sum of independent Bernoulli with parameters $\frac{q}{q+\lambda_i}$. Hence,
$$m = \mathbb{E}\left[|\rho(\Phi_q)|\right] = \sum_{i=0}^{n-1} \frac{q}{q+\lambda_i}.$$

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

## Properties of this set (Wilson (96)).

Let $(X(t), t \geq 0)$ be a Markov process with generator $\mathcal{L}$ and $T_q \sim \mathcal{E}(q)$. Set

$$K_q(x,y) = q(q\mathrm{Id} - \mathcal{L})^{-1}(x,y) = P_x\left[X(T_q) = y\right].$$

- $\rho(\Phi_q)$ is a determinantal process with kernel $K_q$ :
$$\forall A \subset \mathcal{X},\ \mathbb{P}\left[A \subset \rho(\Phi_q)\right] = \det_A(K_q).$$
- $|\rho(\Phi_q)|$ is distributed as the sum of independent Bernoulli with parameters $\frac{q}{q+\lambda_i}$. Hence,
$$m = \mathbb{E}\left[|\rho(\Phi_q)|\right] = \sum_{i=0}^{n-1} \frac{q}{q+\lambda_i}.$$

As a determinantal process, the points in $\rho(\Phi_q)$ repulse one each other :

$$\text{For } x \neq y,\ \mathbb{P}\left[y \in \rho(\Phi_q) | x \in \rho(\Phi_q)\right] \leq \mathbb{P}\left[y \in \rho(\Phi_q)\right]$$

# Subsampling. Choosing a well-spread subset $\bar{\mathcal{X}}$ of $\mathcal{X}$

## Properties of this set (Wilson (96)).

Let $(X(t), t \geq 0)$ be a Markov process with generator $\mathcal{L}$ and $T_q \sim \mathcal{E}(q)$. Set

$$K_q(x,y) = q(q\mathrm{Id} - \mathcal{L})^{-1}(x,y) = P_x\left[X(T_q) = y\right].$$

- $\rho(\Phi_q)$ is a determinantal process with kernel $K_q$ :
$$\forall A \subset \mathcal{X}, \mathbb{P}\left[A \subset \rho(\Phi_q)\right] = \det_A(K_q).$$
- $|\rho(\Phi_q)|$ is distributed as the sum of independent Bernoulli with parameters $\frac{q}{q+\lambda_i}$. Hence,
$$m = \mathbb{E}\left[|\rho(\Phi_q)|\right] = \sum_{i=0}^{n-1} \frac{q}{q+\lambda_i}.$$

As a determinantal process, the points in $\rho(\Phi_q)$ repulse one each other :

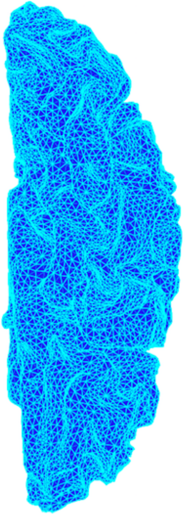
$$\text{For } x \neq y,\ \mathbb{P}\left[y \in \rho(\Phi_q) | x \in \rho(\Phi_q)\right] \leq \mathbb{P}\left[y \in \rho(\Phi_q)\right]$$

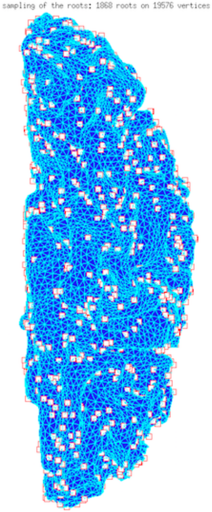
Moreover (Avena & Gaudillière (17)), let $H_{\rho(\Phi_q)}$ the hitting time of $\rho(\Phi_q)$ :
$$\mathbb{E}\left(E_x\left[H_{\rho(\Phi_q)}\right]\right) \text{ does not depend on } x.$$

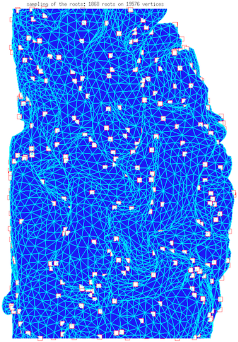
In some sense, the points of $\rho(\Phi_q)$ are well spread in $\mathcal{X}$.

# Examples

Set of roots for small $q$. About 10% of the points are kept.

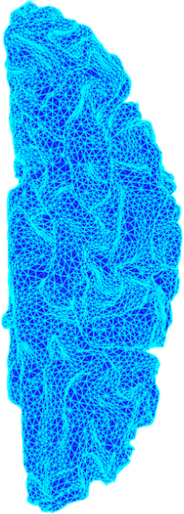


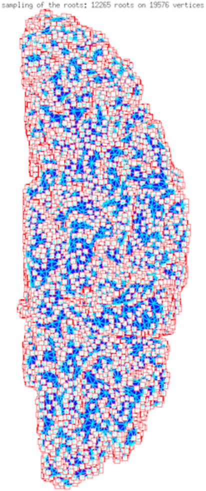


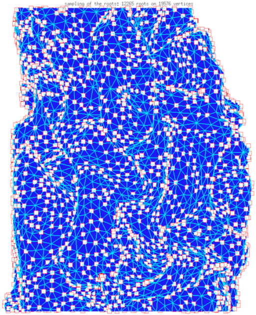



# Examples.

Set of roots for large $q$. About 2/3 of the points are kept.





## Subgraph and filtering.

**Issues :**

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

# Subgraph and filtering.

Issues :

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

- $\mathcal{X}_1 = \rho(\Phi_q)$ is well spread on the graph

# Subgraph and filtering.

**Issues :**

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

- $\mathcal{X}_1 = \rho(\Phi_q)$ is well spread on the graph
- $f_1$ is smooth if $f$ is smooth so that $\mathcal{L}_1 f_1(\bar{x}) = \mathcal{L}_1 \Lambda_1 f(\bar{x})$ is small.

# Subgraph and filtering.

**Issues :**

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

- $\mathcal{X}_1 = \rho(\Phi_q)$ is well spread on the graph
- $f_1$ is smooth if $f$ is smooth so that $\mathcal{L}_1 f_1(\bar{x}) = \mathcal{L}_1 \Lambda_1 f(\bar{x})$ is small.

$\rightarrow$ This is satisfied if $\mathcal{L}_1 \Lambda_1 f(x) = \Lambda_1 \mathcal{L} f(x)$ and more generally if

$$\mathcal{L}_1 \Lambda_1 = \Lambda_1 \mathcal{L}$$

## Subgraph and filtering.

**Issues :**

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

- $\mathcal{X}_1 = \rho(\Phi_q)$ is well spread on the graph
- $f_1$ is smooth if $f$ is smooth so that $\mathcal{L}_1 f_1(\bar{x}) = \mathcal{L}_1 \Lambda_1 f(\bar{x})$ is small.

$\rightarrow$ This is satisfied if $\mathcal{L}_1 \Lambda_1 f(x) = \Lambda_1 \mathcal{L} f(x)$ and more generally if

$$\mathcal{L}_1 \Lambda_1 = \Lambda_1 \mathcal{L}$$

our goal :

# Subgraph and filtering.

**Issues :**

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

- $\mathcal{X}_1 = \rho(\Phi_q)$ is well spread on the graph
- $f_1$ is smooth if $f$ is smooth so that $\mathcal{L}_1 f_1(\bar{x}) = \mathcal{L}_1 \Lambda_1 f(\bar{x})$ is small.

$\rightarrow$ This is satisfied if $\mathcal{L}_1 \Lambda_1 f(x) = \Lambda_1 \mathcal{L} f(x)$ and more generally if

$$\mathcal{L}_1 \Lambda_1 = \Lambda_1 \mathcal{L}$$

our goal :

- find exact or approximate solutions of $\mathcal{L}_1 \Lambda_1 = \Lambda_1 \mathcal{L}$ (Intertwining equation, Rogers & Pitman (81), Diaconis & Fill (90)).

# Subgraph and filtering.

**Issues :**

We want to compute a subgraph $G_1 = \{\mathcal{X}_1, \mathcal{L}_1\}$ and $f_1 = \Lambda_1 f$ defined on $\mathcal{X}_1$ such that

- $\mathcal{X}_1 = \rho(\Phi_q)$ is well spread on the graph
- $f_1$ is smooth if $f$ is smooth so that $\mathcal{L}_1 f_1(\bar{x}) = \mathcal{L}_1 \Lambda_1 f(\bar{x})$ is small.

$\rightarrow$ This is satisfied if $\mathcal{L}_1 \Lambda_1 f(x) = \Lambda_1 \mathcal{L} f(x)$ and more generally if

$$\mathcal{L}_1 \Lambda_1 = \Lambda_1 \mathcal{L}$$

our goal :

- find exact or approximate solutions of $\mathcal{L}_1 \Lambda_1 = \Lambda_1 \mathcal{L}$ (Intertwining equation, Rogers & Pitman (81), Diaconis & Fill (90)).
- Avoid diagonalization of the Laplacian.

# Weighting and Filtering : Markov intertwining

**Markov intertwining (Rogers & Pitman (81), Diaconis & Fill (90)) :**

A way to link two Markov processes on different state spaces :

$$\Lambda \mathcal{L} = \bar{\mathcal{L}} \Lambda \,,$$

# Weighting and Filtering : Markov intertwining

**Markov intertwining (Rogers & Pitman (81), Diaconis & Fill (90)) :**

A way to link two Markov processes on different state spaces :

$$\Lambda\mathcal{L} = \bar{\mathcal{L}}\Lambda\,,$$

where

- $\mathcal{L}$ is a Markov generator on $\mathcal{X}$ ;

# Weighting and Filtering : Markov intertwining

### Markov intertwining (Rogers & Pitman (81), Diaconis & Fill (90)) :

A way to link two Markov processes on different state spaces :

$$\Lambda\mathcal{L} = \bar{\mathcal{L}}\Lambda,$$

where

- $\mathcal{L}$ is a Markov generator on $\mathcal{X}$ ;
- $\bar{\mathcal{L}}$ is a Markov generator on $\bar{\mathcal{X}}$ ;

# Weighting and Filtering : Markov intertwining

## Markov intertwining (Rogers & Pitman (81), Diaconis & Fill (90)) :

A way to link two Markov processes on different state spaces :

$$\Lambda\mathcal{L} = \bar{\mathcal{L}}\Lambda\,,$$

where

- $\mathcal{L}$ is a Markov generator on $\mathcal{X}$ ;
- $\bar{\mathcal{L}}$ is a Markov generator on $\bar{\mathcal{X}}$ ;
- $\Lambda$ is a rectangular matrix indexed by $\bar{\mathcal{X}} \times \mathcal{X}$ with positive entries : $\nu_{\bar{x}} = \Lambda(\bar{x}, \cdot)$ is a probability measure on $\mathcal{X}$.

# Weighting and Filtering : Markov intertwining

## Markov intertwining (Rogers & Pitman (81), Diaconis & Fill (90)) :

A way to link two Markov processes on different state spaces :

$$\Lambda\mathcal{L} = \bar{\mathcal{L}}\Lambda\,,$$

where

- $\mathcal{L}$ is a Markov generator on $\mathcal{X}$ ;
- $\bar{\mathcal{L}}$ is a Markov generator on $\bar{\mathcal{X}}$ ;
- $\Lambda$ is a rectangular matrix indexed by $\bar{\mathcal{X}} \times \mathcal{X}$ with positive entries : $\nu_{\bar{x}} = \Lambda(\bar{x}, \cdot)$ is a probability measure on $\mathcal{X}$.

## Why is it useful for us ?

- It provides a natural choice of the weights on $\bar{\mathcal{X}}$ : $\bar{w}(\bar{x}, \bar{y}) = \bar{\mathcal{L}}(\bar{x}, \bar{y})$.
- It provides a natural choice of the approximation coefficients : $\bar{f}(\bar{x}) = \nu_{\bar{x}}(f) = \Lambda f(\bar{x})$.

# Weighting and Filtering : Markov intertwining

Our goal : Given $\bar{\mathcal{X}} \subset \mathcal{X}$, find an approximate solution $(\Lambda, \bar{\mathcal{L}})$ to $\Lambda\mathcal{L} = \bar{\mathcal{L}}\Lambda$ such that

- $\bar{\mathcal{L}}$ is symmetric.
- The $(\nu_{\bar{x}}; \bar{x} \in \bar{\mathcal{X}})$ are "well-localized" in space (non overlapping), to get good reconstruction.
- The $(\nu_{\bar{x}}; \bar{x} \in \bar{\mathcal{X}})$ are "well-localized" in frequency, to separate high and low frequency parts of the signal.
- $\bar{\mathcal{L}}$ and $\Lambda$ are easy to compute (we do not want to compute the spectral decomposition of $\mathcal{L}$).

# Weighting and Filtering : Markov intertwining

**Our proposal.** Assume $\bar{\mathcal{X}} \subset \mathcal{X}$ and $q' > 0$ are given.

- For $\bar{x} \in \bar{\mathcal{X}}$ and $y \in \mathcal{X}$,
  $$\nu_{\bar{x}}(y) = \Lambda(\bar{x}, y) := K_{q'}(\bar{x}, y) = q'(q'\mathrm{Id} - \mathcal{L})^{-1}(\bar{x}, y) = P_{\bar{x}}\left[X(T_{q'}) = y\right].$$

# Weighting and Filtering : Markov intertwining

**Our proposal. Assume $\bar{\mathcal{X}} \subset \mathcal{X}$ and $q' > 0$ are given.**

- For $\bar{x} \in \bar{\mathcal{X}}$ and $y \in \mathcal{X}$,
  $$\nu_{\bar{x}}(y) = \Lambda(\bar{x}, y) := K_{q'}(\bar{x}, y) = q'(q'\mathrm{Id} - \mathcal{L})^{-1}(\bar{x}, y) = P_{\bar{x}}\left[X(T_{q'}) = y\right].$$
- For $\bar{x} \in \bar{\mathcal{X}}$ and $\bar{y} \in \bar{\mathcal{X}}$,
  $$\bar{P}(\bar{x}, \bar{y}) := P_{\bar{x}}\left[X(H^+_{\bar{\mathcal{X}}}) = \bar{y}\right], \ \bar{\mathcal{L}} = \alpha(\bar{P} - \mathrm{Id}),$$
  where $H^+_{\bar{\mathcal{X}}}$ is the return time of the process $X$ in $\bar{\mathcal{X}}$.
  $\bar{\mathcal{L}}$ is computed as a Schur complement.

# Weighting and Filtering : Markov intertwining

**Our proposal. Assume $\bar{\mathcal{X}} \subset \mathcal{X}$ and $q' > 0$ are given.**

- For $\bar{x} \in \bar{\mathcal{X}}$ and $y \in \mathcal{X}$,
  $$\nu_{\bar{x}}(y) = \Lambda(\bar{x}, y) := K_{q'}(\bar{x}, y) = q'(q'\mathrm{Id} - \mathcal{L})^{-1}(\bar{x}, y) = P_{\bar{x}}\left[X(T_{q'}) = y\right].$$
- For $\bar{x} \in \bar{\mathcal{X}}$ and $\bar{y} \in \bar{\mathcal{X}}$,
  $$\bar{P}(\bar{x}, \bar{y}) := P_{\bar{x}}\left[X(H^{+}_{\bar{\mathcal{X}}}) = \bar{y}\right], \ \bar{\mathcal{L}} = \alpha(\bar{P} - \mathrm{Id}),$$
  where $H^{+}_{\bar{\mathcal{X}}}$ is the return time of the process $X$ in $\bar{\mathcal{X}}$.
  $\bar{\mathcal{L}}$ is computed as a Schur complement.

**Definition of the approximation and detail coefficients.**

- For $\bar{x} \in \bar{\mathcal{X}}$, $\bar{f}(\bar{x}) = \nu_{\bar{x}}(f) = K_{q'}f(\bar{x})$.
- For $\breve{x} \in \breve{\mathcal{X}}$, $\breve{f}(\breve{x}) = (K_{q'} - \mathrm{Id})f(\breve{x})$.

# Weighting and Filtering : Markov intertwining

**Our proposal. Assume $\bar{\mathcal{X}} \subset \mathcal{X}$ and $q' > 0$ are given.**

- For $\bar{x} \in \bar{\mathcal{X}}$ and $y \in \mathcal{X}$,
  $$\nu_{\bar{x}}(y) = \Lambda(\bar{x}, y) := K_{q'}(\bar{x}, y) = q'(q'\mathrm{Id} - \mathcal{L})^{-1}(\bar{x}, y) = P_{\bar{x}}\left[X(T_{q'}) = y\right].$$
- For $\bar{x} \in \bar{\mathcal{X}}$ and $\bar{y} \in \bar{\mathcal{X}}$,
  $$\bar{P}(\bar{x}, \bar{y}) := P_{\bar{x}}\left[X(H^+_{\bar{\mathcal{X}}}) = \bar{y}\right], \ \bar{\mathcal{L}} = \alpha(\bar{P} - \mathrm{Id}),$$
  where $H^+_{\bar{\mathcal{X}}}$ is the return time of the process $X$ in $\bar{\mathcal{X}}$.
  $\bar{\mathcal{L}}$ is computed as a Schur complement.

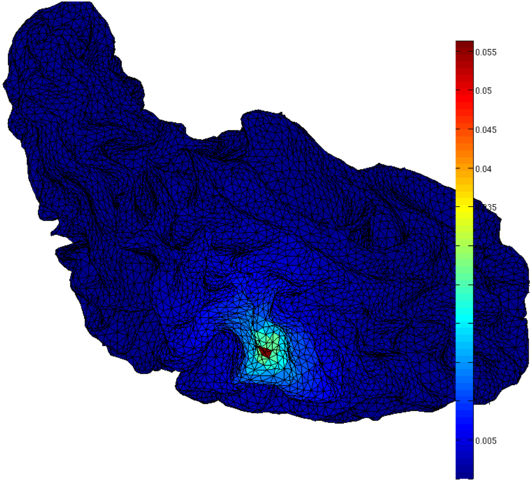
**Definition of the approximation and detail coefficients.**

- For $\bar{x} \in \bar{\mathcal{X}}$, $\bar{f}(\bar{x}) = \nu_{\bar{x}}(f) = K_{q'}f(\bar{x})$.
- For $\breve{x} \in \breve{\mathcal{X}}$, $\breve{f}(\breve{x}) = (K_{q'} - \mathrm{Id})f(\breve{x})$.

**Some comments.**

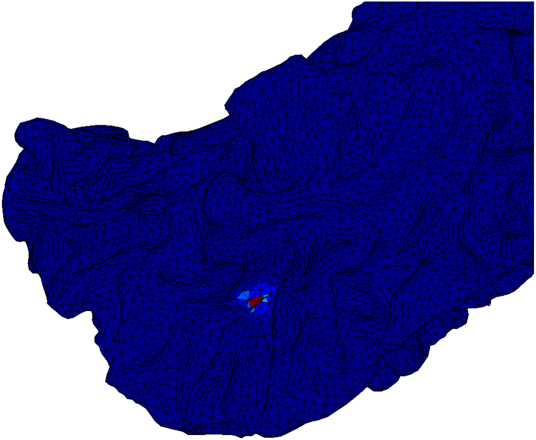
- When $q' \ll 1$, $K_{q'}(\bar{x}, \cdot) \simeq \mu$ is well frequency-localized, is a solution to the intertwining relation, is poorly space-localized.
- When $q' \gg 1$, $K_{q'}(\bar{x}, \cdot) \simeq \delta_{\bar{x}}$ is well space-localized. The frequency localization is lost, and depends on the choice of the subset $\bar{\mathcal{X}}$.

# Example of one $\nu_{\bar{x}}$




small $q'$

# Example of one $\nu_{\bar{x}}$



large $q'$

# Some additional results

## An explicit reconstruction formula.

$$f = \begin{pmatrix} \mathrm{Id}_{\bar{\mathcal{X}}} - \frac{1}{q'}\bar{\mathcal{L}} & \mathcal{L}_{\bar{\mathcal{X}}\breve{\mathcal{X}}}(-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1} \\ (-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1}\mathcal{L}_{\breve{\mathcal{X}}\bar{\mathcal{X}}} & q'\mathcal{L}^{-1}_{\breve{\mathcal{X}}\breve{\mathcal{X}}} - \mathrm{Id}_{\breve{\mathcal{X}}} \end{pmatrix} \begin{pmatrix} \bar{f} \\ \breve{f} \end{pmatrix} = \bar{R}\bar{f} + \breve{R}\breve{f}\,,$$

# Some additional results

**An explicit reconstruction formula.**

$$f = \begin{pmatrix} \mathrm{Id}_{\bar{\mathcal{X}}} - \frac{1}{q'}\bar{\mathcal{L}} & \mathcal{L}_{\bar{\mathcal{X}}\breve{\mathcal{X}}}(-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1} \\ (-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1}\mathcal{L}_{\breve{\mathcal{X}}\bar{\mathcal{X}}} & q'\mathcal{L}^{-1}_{\breve{\mathcal{X}}\breve{\mathcal{X}}} - \mathrm{Id}_{\breve{\mathcal{X}}} \end{pmatrix} \begin{pmatrix} \bar{f} \\ \breve{f} \end{pmatrix} = \bar{R}\bar{f} + \breve{R}\breve{f}\,,$$

**Conditioning of the reconstruction operator. Space localization.**

$$\left\|\bar{R}\right\|_{\infty,\infty} \leq 1 + 2\frac{\bar{\alpha}}{q'}\ ,\ \left\|\breve{R}\right\|_{\infty,\infty} \leq \max\left(\frac{\alpha}{\beta}; 1 + \frac{q'}{\gamma}\right)\ .$$

# Some additional results

**An explicit reconstruction formula.**

$$f = \begin{pmatrix} \mathrm{Id}_{\bar{\mathcal{X}}} - \frac{1}{q'}\bar{\mathcal{L}} & \mathcal{L}_{\bar{\mathcal{X}}\breve{\mathcal{X}}}(-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1} \\ (-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1}\mathcal{L}_{\breve{\mathcal{X}}\bar{\mathcal{X}}} & q'\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}}^{-1} - \mathrm{Id}_{\breve{\mathcal{X}}} \end{pmatrix} \begin{pmatrix} \bar{f} \\ \breve{f} \end{pmatrix} = \bar{R}\bar{f} + \breve{R}\breve{f}\,,$$

**Conditioning of the reconstruction operator. Space localization.**

$$\left\|\bar{R}\right\|_{\infty,\infty} \leq 1 + 2\frac{\bar{\alpha}}{q'}\ ,\ \left\|\breve{R}\right\|_{\infty,\infty} \leq \max\left(\frac{\alpha}{\beta}; 1 + \frac{q'}{\gamma}\right)\ .$$

**Error in the intertwining relation. Frequency localization.**

$$\left\|\bar{\mathcal{L}}\Lambda - \Lambda\mathcal{L}\right\|_{\infty,\infty} \leq 2q'\frac{\alpha}{\beta}\ .$$

# Some additional results

**An explicit reconstruction formula.**

$$f = \begin{pmatrix} \mathrm{Id}_{\bar{\mathcal{X}}} - \frac{1}{q'}\bar{\mathcal{L}} & \mathcal{L}_{\bar{\mathcal{X}}\breve{\mathcal{X}}}(-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1} \\ (-\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}})^{-1}\mathcal{L}_{\breve{\mathcal{X}}\bar{\mathcal{X}}} & q'\mathcal{L}_{\breve{\mathcal{X}}\breve{\mathcal{X}}}^{-1} - \mathrm{Id}_{\breve{\mathcal{X}}} \end{pmatrix} \begin{pmatrix} \bar{f} \\ \breve{f} \end{pmatrix} = \bar{R}\bar{f} + \breve{R}\breve{f}\,,$$

**Conditioning of the reconstruction operator. Space localization.**

$$\left\|\bar{R}\right\|_{\infty,\infty} \leq 1 + 2\frac{\bar{\alpha}}{q'}\ , \ \left\|\breve{R}\right\|_{\infty,\infty} \leq \max\left(\frac{\alpha}{\beta}; 1 + \frac{q'}{\gamma}\right)\ .$$

**Error in the intertwining relation. Frequency localization.**

$$\left\|\bar{\mathcal{L}}\Lambda - \Lambda\mathcal{L}\right\|_{\infty,\infty} \leq 2q'\frac{\alpha}{\beta}\ .$$

**Regularity implies small details. Jackson type inequality**

$$\left\|f - \bar{R}_0\bar{R}_1 ... \bar{R}_{K-1}f_K\right\|_{\infty} \leq C_K \left\|\mathcal{L}f\right\|_{\infty} + D_K \left\|f\right\|_{\infty}\ .$$

with

$$\frac{1}{\beta} := \max_{\bar{x}\in\bar{\mathcal{X}}} E_{\bar{x}}\left[H^+_{\breve{\mathcal{X}}} - \tau_1\right],\ \frac{1}{\gamma} := \max_{\breve{x}\in\breve{\mathcal{X}}} E_{\breve{x}}\left[H_{\bar{\mathcal{X}}}\right]\ .$$

# Choice of $q$ and $q'$.

## Wished Properties.

We are looking for $(q, q')$ such that

- $|\rho(\Phi_q)|$ is approximately a given proportion of $|\mathcal{X}|$ (say between [1/2,2/3]) ;
- The reconstruction error is small : $\frac{\bar{\alpha}}{q'}$ and $\frac{q'}{\gamma}$ small.

# Choice of $q$ and $q'$.

## Wished Properties.

We are looking for $(q, q')$ such that

- $|\rho(\Phi_q)|$ is approximately a given proportion of $|\mathcal{X}|$ (say between [1/2,2/3]) ;
- The reconstruction error is small : $\frac{\bar{\alpha}}{q'}$ and $\frac{q'}{\gamma}$ small.
- The intertwining error is small : $\frac{q'}{\beta}$ small.

# Choice of $q$ and $q'$.

## Wished Properties.

We are looking for $(q, q')$ such that

- $|\rho(\Phi_q)|$ is approximately a given proportion of $|\mathcal{X}|$ (say between [1/2,2/3]) ;
- The reconstruction error is small : $\frac{\bar{\alpha}}{q'}$ and $\frac{q'}{\gamma}$ small.
- The intertwining error is small : $\frac{q'}{\beta}$ small.

## A systematic procedure

Let us skip the details...

# Choice of $q$ and $q'$.

## Wished Properties.

We are looking for $(q, q')$ such that

- $|\rho(\Phi_q)|$ is approximately a given proportion of $|\mathcal{X}|$ (say between [1/2,2/3]) ;
- The reconstruction error is small : $\frac{\bar{\alpha}}{q'}$ and $\frac{q'}{\gamma}$ small.
- The intertwining error is small : $\frac{q'}{\beta}$ small.

## A systematic procedure

Let us skip the details...

- we know how to choose automatically appropriate $(q, q')$ from one scale to the other.

## Choice of $q$ and $q'$.

### Wished Properties.

We are looking for $(q, q')$ such that

- $|\rho(\Phi_q)|$ is approximately a given proportion of $|\mathcal{X}|$ (say between [1/2,2/3]) ;
- The reconstruction error is small : $\frac{\bar{\alpha}}{q'}$ and $\frac{q'}{\gamma}$ small.
- The intertwining error is small : $\frac{q'}{\beta}$ small.

### A systematic procedure

Let us skip the details...

- we know how to choose automatically appropriate $(q, q')$ from one scale to the other.
- and also... we have a procedure to keep the subgraph "sparse"

# The bases and its by by-products

**You give to the algorithm**

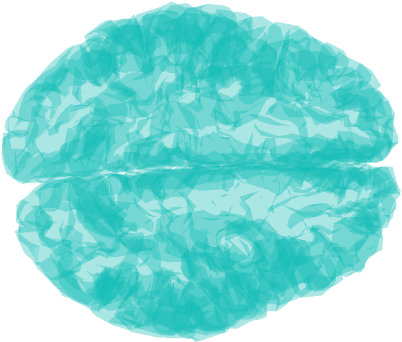
- A graph
- and if you wish a maximum number of levels

**you end up with**

- a sequence of subgraphs
- multiscale bases on your graph until the coarsest approximation level you decided.

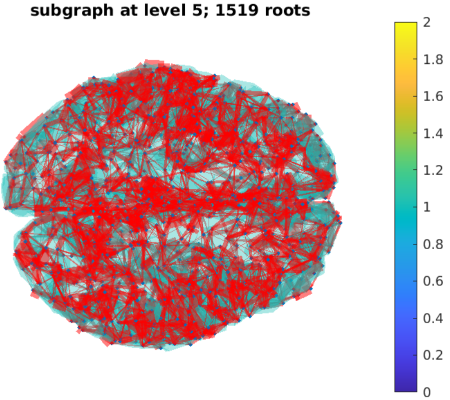
# Subgraphs

(data J.M. Lina, Université de Montréal)



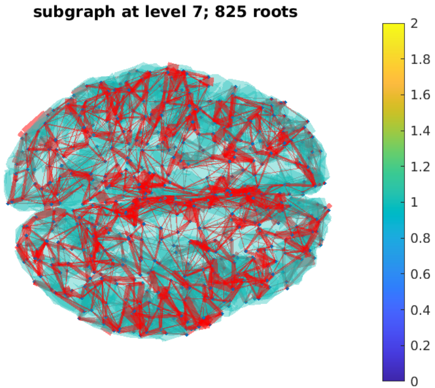
A transparent brain

# Subgraphs



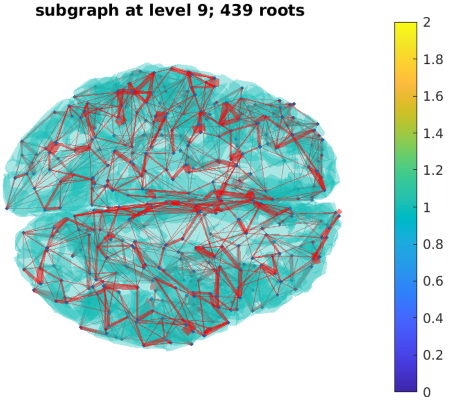

Subgraph at level 5

# Subgraphs



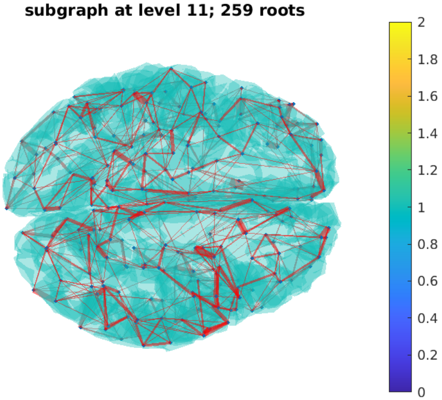

Subgraph at level 7

# Subgraphs



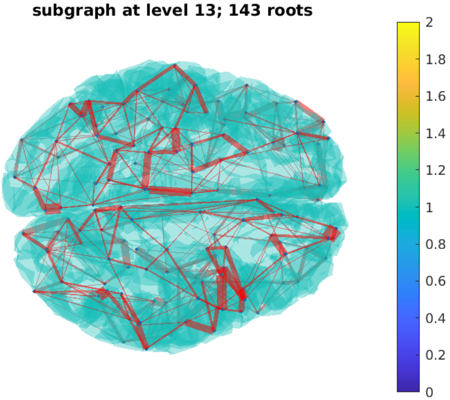

Subgraph at level 9

# Subgraphs



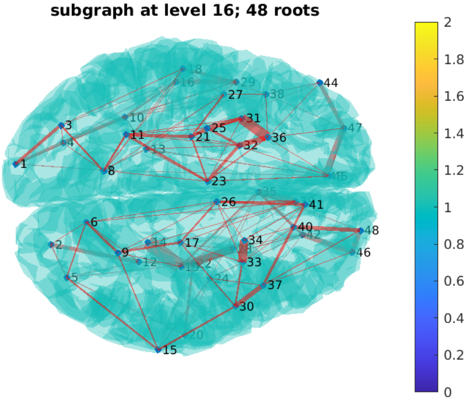

Subgraph at level 11

# Subgraphs




Subgraph at level 13

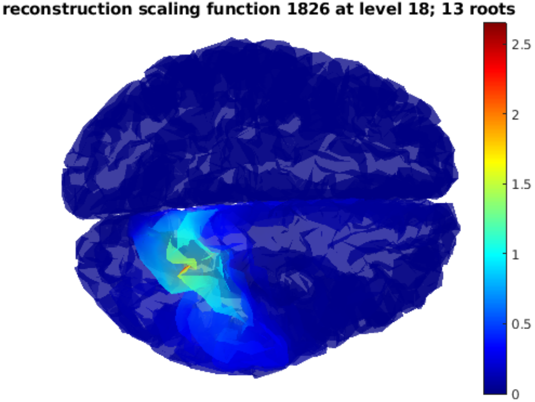
# Subgraphs




Subgraph at level 16

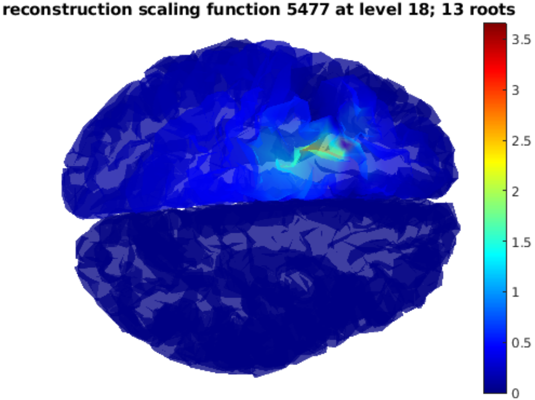
# Choose your level of approximation

Coarse level : approximation function



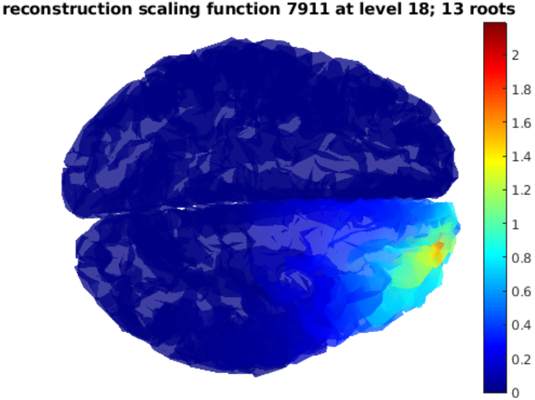
# Choose your level of approximation

Coarse level : approximation function



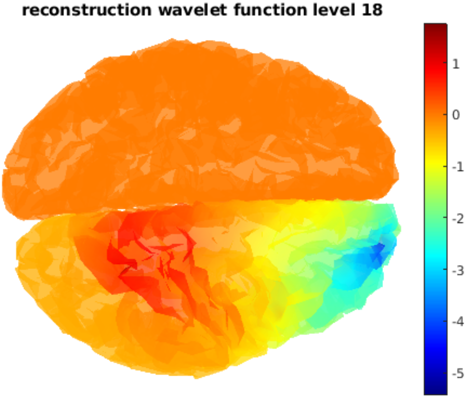
# Choose your level of approximation

Coarse level : approximation function



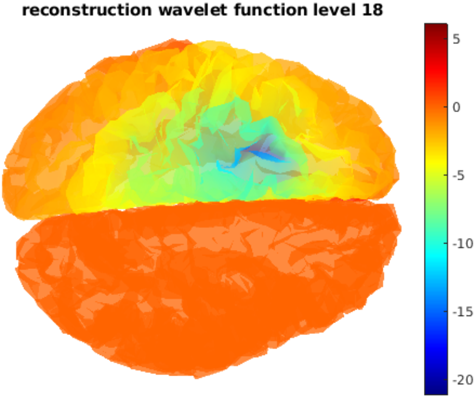
# Choose your level of approximation

Coarse level : wavelet function



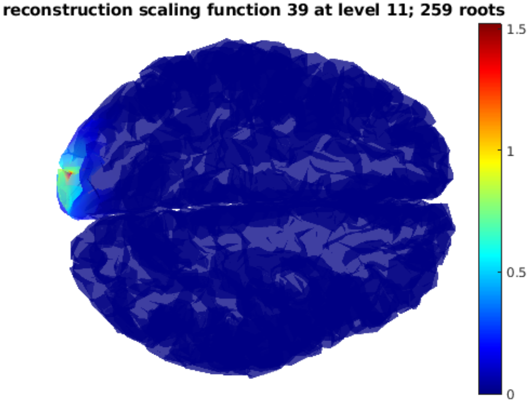
# Choose your level of approximation

Coarse level : wavelet function



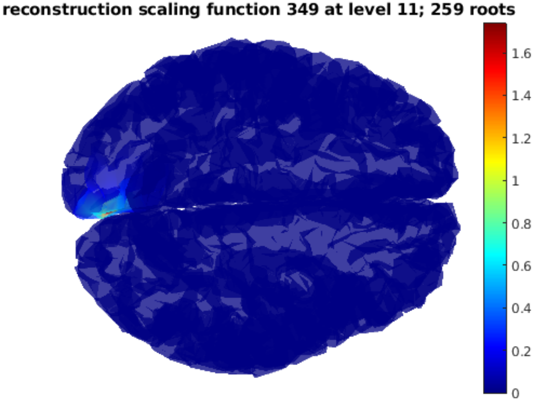
# Choose your level of approximation

Intermediate level : approximation function



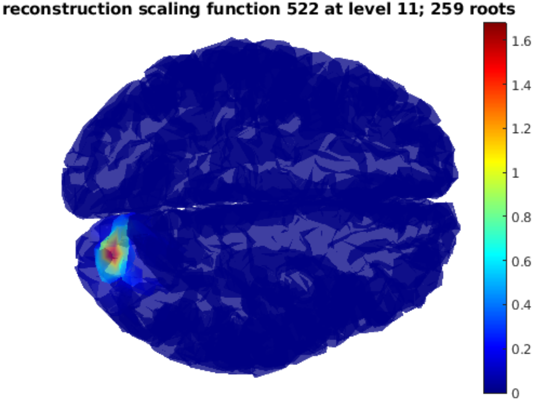
# Choose your level of approximation

Intermediate level : approximation function



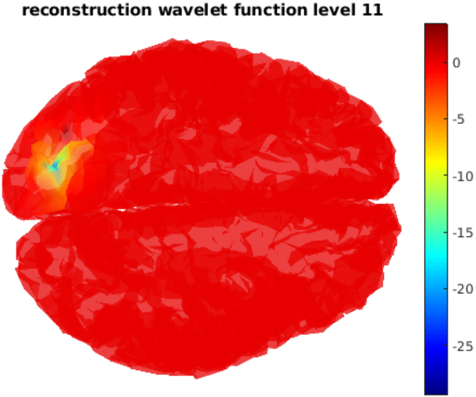
# Choose your level of approximation

Intermdiate level : approximation function



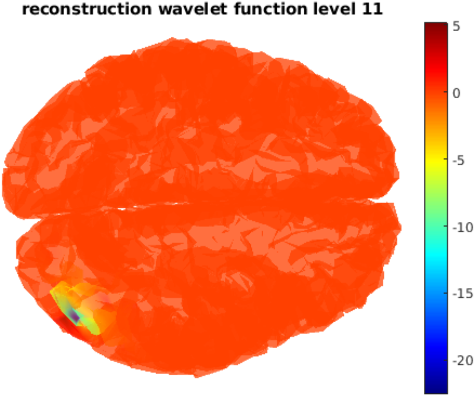
# Choose your level of approximation

Intermediate level : wavelet function



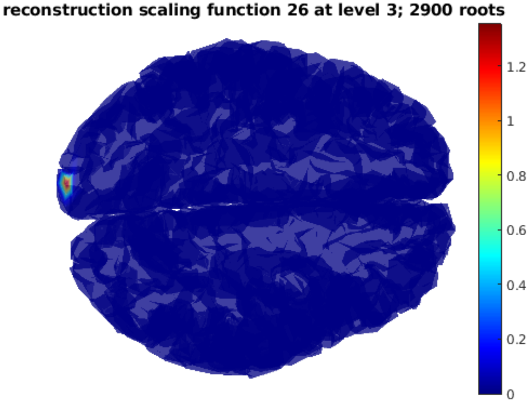
# Choose your level of approximation

Intermediate level : wavelet function



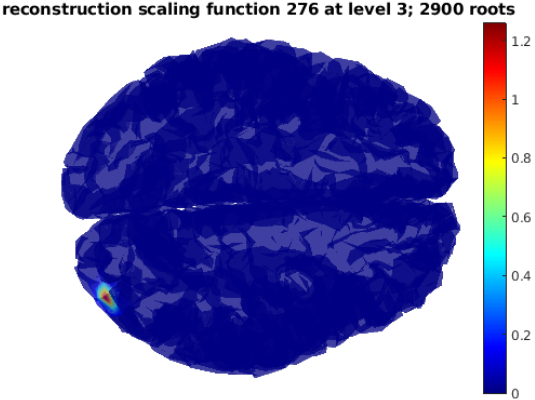
# Choose your level of approximation

Fine level : approximation function



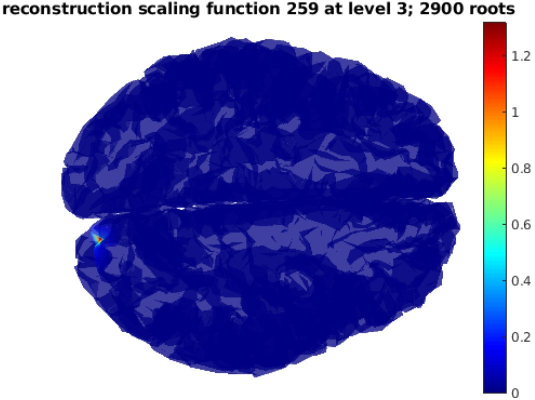
# Choose your level of approximation

Fine level : approximation function



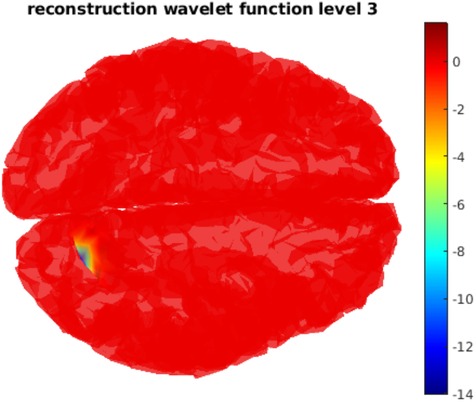
# Choose your level of approximation

Fine level : approximation function



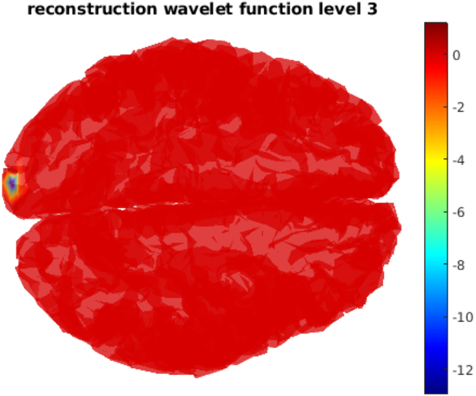
# Choose your level of approximation

Fine level : wavelet function



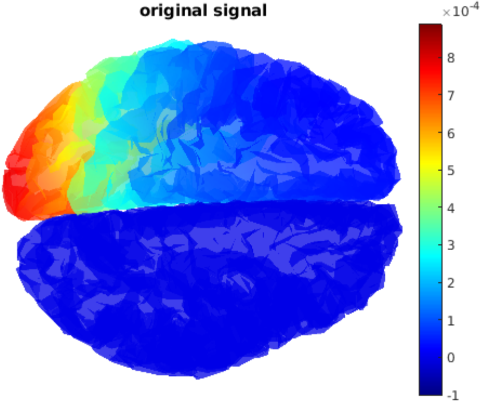
# Choose your level of approximation

Fine level : wavelet function



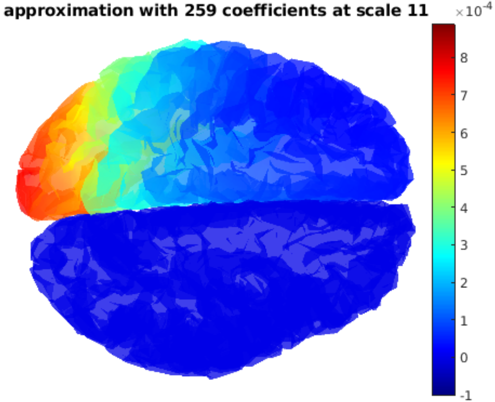
# A sparse representation of smooths signals

A smooth signal



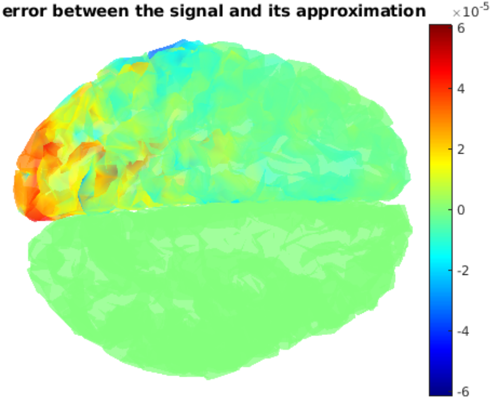
# A sparse representation of smooths signals

Approximation with 3% of coefficients



# A sparse representation of smooths signals

Error of approximation



# Good news !

A Python toolbox should be soon available !