Classifying brain connectivity data using graph embeddings

Jonas Richiardi
Medical Image Processing Laboratory

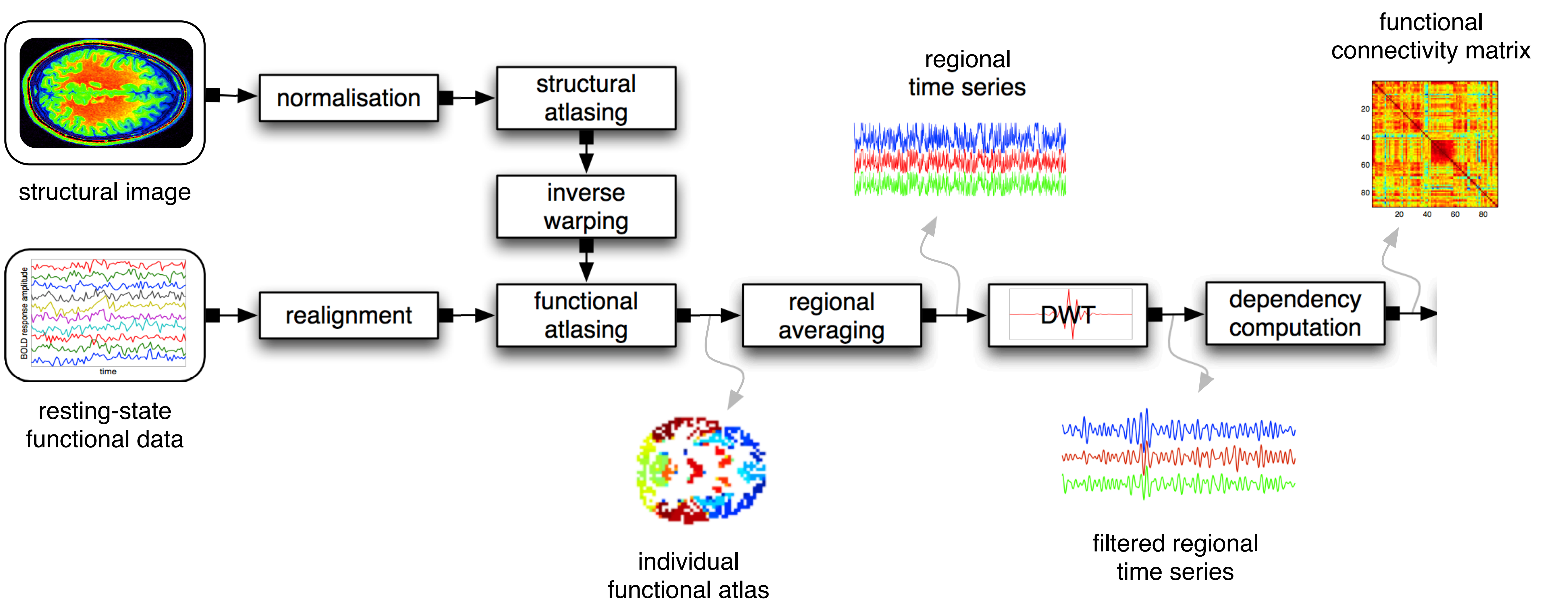
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
Institute of Bioengineering

UNIVERSITÉ DE GENÈVE
Dept. of Radiology and Medical Informatics

MLNI Workshop / Nov 2011

From imaging data to functional connectivity

- Functional connectivity: "*statistical dependence between time series in distinct brain locations*"
- "Classical" wavelet correlation pipeline*:



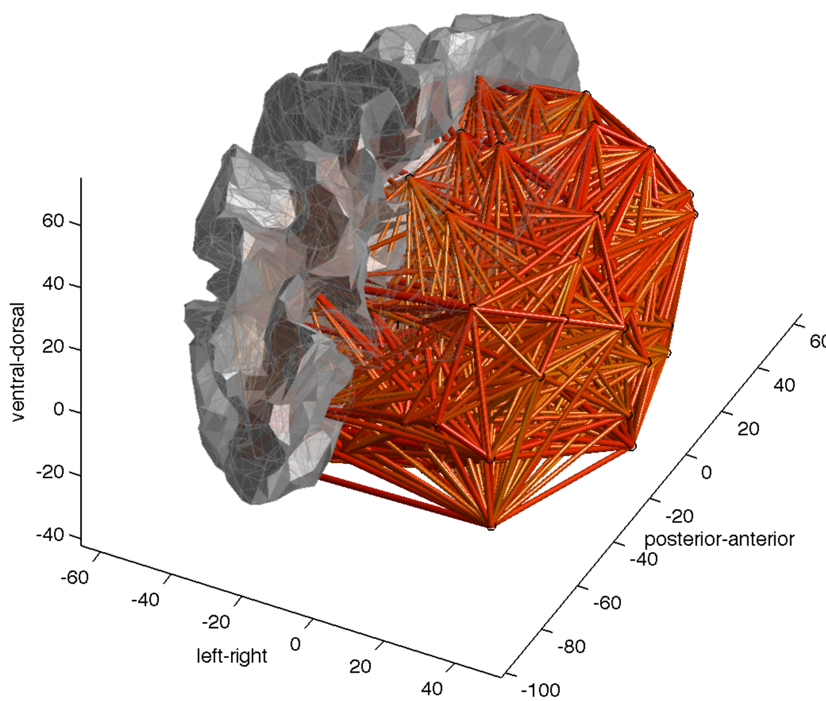
* [Achard et al., J. Neurosci 2006]

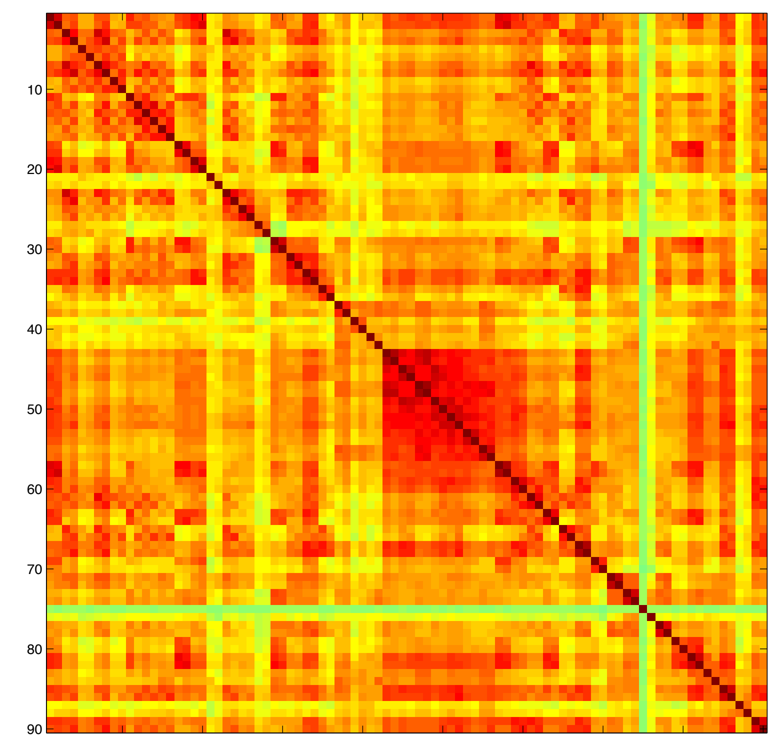
Matlab code: http://miplab.unige.ch/richiardi/software.php

Sophie Achard's R code: http://cran.r-project.org/web/packages/brainwaver/

Functional connectivity as a graph

- The correlation matrix (minus the diagonal) can be seen as the adjacency matrix **A** of a "functional connectivity graph":
  - **Vertices** correspond to voxels or regions
  - **Edge labels** encode pairwise strength of temporal dependence





Agenda for this talk

- Preliminaries
- Graph family of interest
- Three different graph embeddings

Functional connectivity as a labelled graph

- “Functional connectivity graphs” can be written formally as labelled graphs.
- Labelled graphs are written: $g = (V, E, \alpha, \beta)$
  - V: the *set of vertices* (nodes, brain regions, ICA components)
  - E: the *set of edges* (connections between nodes)
  - α: *vertex labelling function* (returns a name or number for each node, for example the anatomical label of the region)
  - β: *edge labelling function* (returns a name or number for each edge, for example the temporal correlation strength)
  - A square *adjacency matrix* (“connectivity matrix”) can encode the presence/absence of connections, and their strengths. It is generally denoted **A**.

A restricted class for atlased connectivity graphs

- Functional brain networks obtained by atlasing can adequately be modelled by a restricted class of labelled graphs we call **graphs with fixed-cardinality vertex sequences**, a subclass of Dickinson et al.'s *graphs with unique node labels*:
  - Fixed number of vertices for all graph instances: $\forall i \ |V_i| = M$
  - Fixed ordering of the set (sequence) V: $V = (v_1, v_2, \ldots, v_M)$
  - Scalar edge labelling functions: $\beta : (v_i, v_j) \mapsto \mathbb{R}$
  - Undirected: $\mathbf{A}^T = \mathbf{A}$
- This is a very restricted (but still expressive) class of graphs
- This limits the effectiveness of many "classical" methods for classifying general graphs (based on *graph matching*).

[Richiardi et al., NeuroImage, 2011] [Richiardi et al., ICPR, 2010] [Dickinson et al., IJPRAI, 2004]

Graph matching techniques

- Goal: recover an optimal permutation matrix $\hat{\mathrm{P}}$ to transform one graph into the other (map nodes).
  - But in our case, $\hat{\mathrm{P}} = \mathrm{I}$ by def.
- *Discrete optimisation*: search algorithm (A*, branch-and-bound...) + cost function
  - Cost function is typically Graph Edit Distance (GED), but in our case, reduces to

$$d(g_1, g_2) = |C_{\oplus}| + |C_{\beta_i \neq \beta_j}|$$

set of edges belonging exclusively to one or the other graph

set of edges with unequal labels

Graph matching techniques (2)

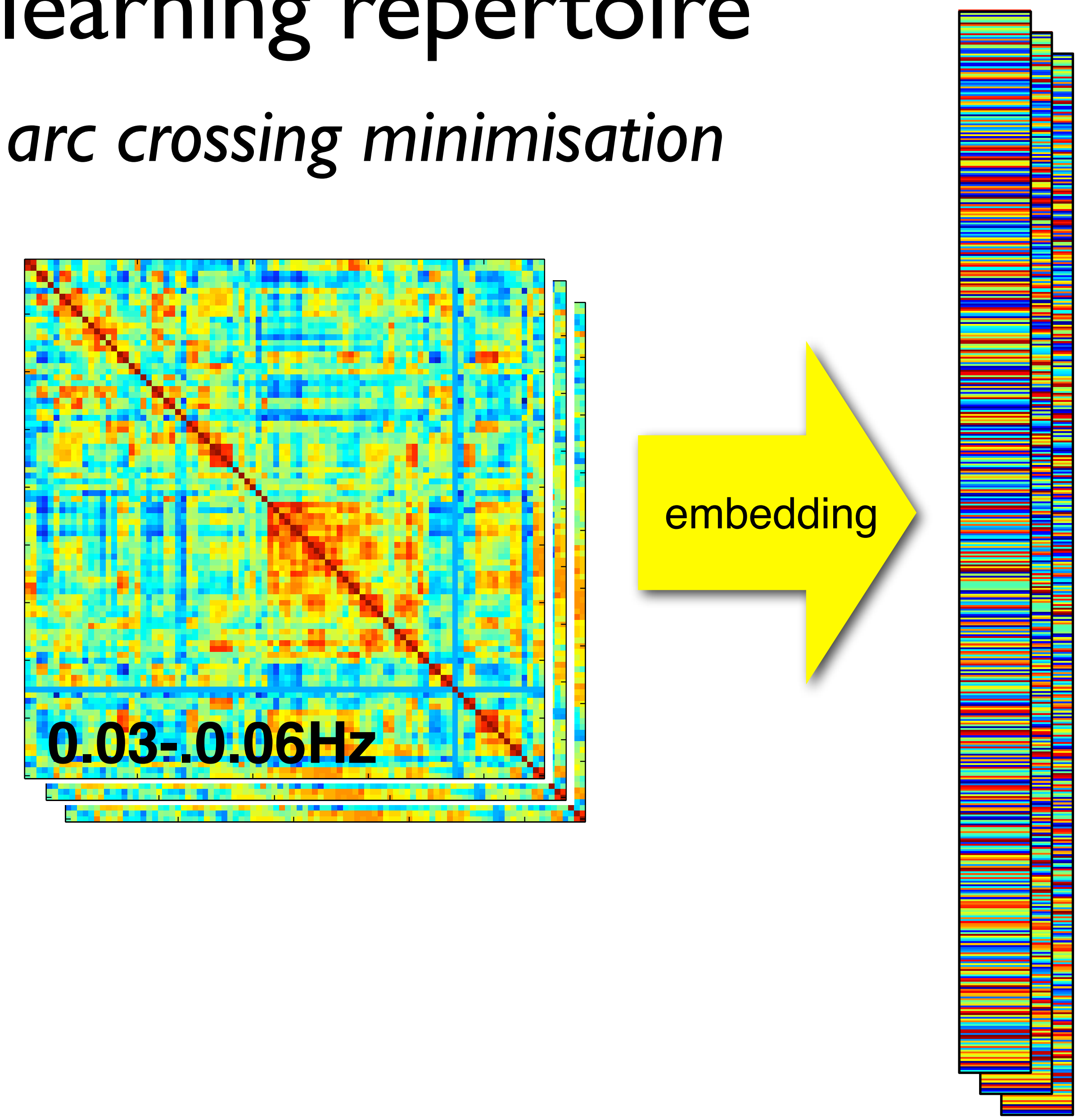
- *Continuous optimisation*: find $\hat{\mathbf{P}}$ to minimise the cost $||\mathbf{A}_1 - \mathbf{P}\mathbf{A}_2\mathbf{P}^T||_F$
  - In our case, reduces to $\sqrt{tr((\mathbf{A}_1 - \mathbf{A}_2)^T(\mathbf{A}_1 - \mathbf{A}_2))}$
- *Spectral methods*: eigendecomposition of adjacency matrix or Laplacian
  - Look more promising for our type of graph
  - But many methods don't make use of eigenvectors
  - ... and not all decompositions are desirable

$$\begin{pmatrix} (1,1) & \dots & (1,|V_i|) \\ & \ddots & \\ & & (|V_i|,|V_i|) \end{pmatrix} \rightarrow \begin{pmatrix} (1,1) \\ \vdots \\ (|V_i|,|V_i|) \end{pmatrix} \rightarrow (\mathbf{B}_1|\mathbf{B}_2|\dots|\mathbf{B}_N) \rightarrow \mathbf{S} = \mathbf{B}\mathbf{B}^T \rightarrow \mathbf{S}\mathbf{e}_i = \lambda_i\mathbf{e}_i$$

$\mathbf{A}_i \in \mathbb{R}^{|V_i|\times|V_i|}$ $\mathbf{B}_i \in \mathbb{R}^{|V_i|^2\times 1}$ $\mathbf{B} \in \mathbb{R}^{|V_i|^2\times N}$ $\mathbf{S} \in \mathbb{R}^{|V_i|^2\times|V_i|^2}$

Embedding connectivity graphs

- Representing the connectivity graph in a **vector space** via **graph embedding** allows the use of a vast statistical machine learning repertoire
  - Here we're not interested in the *arc crossing minimisation problem* or *planar graphs*
- We proposed several ways of doing this, including
  1. Direct embedding
  2. Dissimilarity embedding
  3. Graph and vertex attribute embedding



Agenda for this talk

- Preliminaries
- Graph family of interest
- Three different graph embeddings
  - Direct embedding
  - Dissimilarity embedding
  - Graph/vertex attribute embedding

1: Direct graph embedding

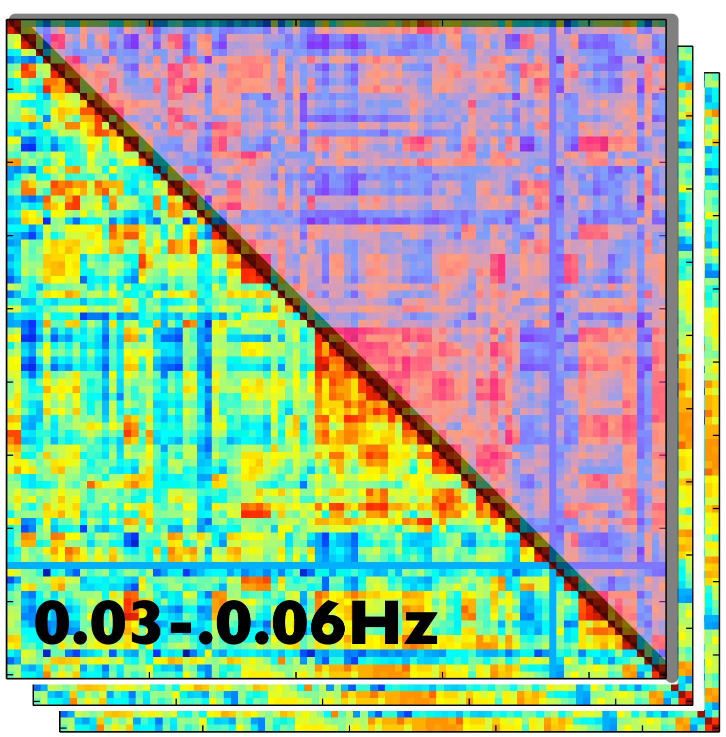
- Direct embedding provides a suitable vector-space representation for the class of graphs of interest

$$\begin{pmatrix} (1,1) & \dots & (1,|V_i|) \\ & \ddots & \\ & & (|V_i|,|V_i|) \end{pmatrix}$$

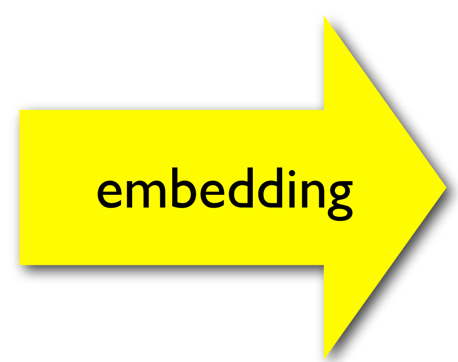
$$\mathbf{A}_i \in \mathbb{R}^{|V_i|\times|V_i|}$$

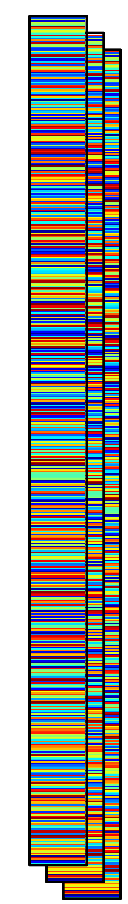
$$\begin{pmatrix} (1,2) \\ \vdots \\ (|V_i|-1,|V_i|) \end{pmatrix}$$

$$\mathbf{B}_i \in \mathbb{R}^{\binom{|V_i|}{2}\times 1}$$



90 regions,
4005 connections



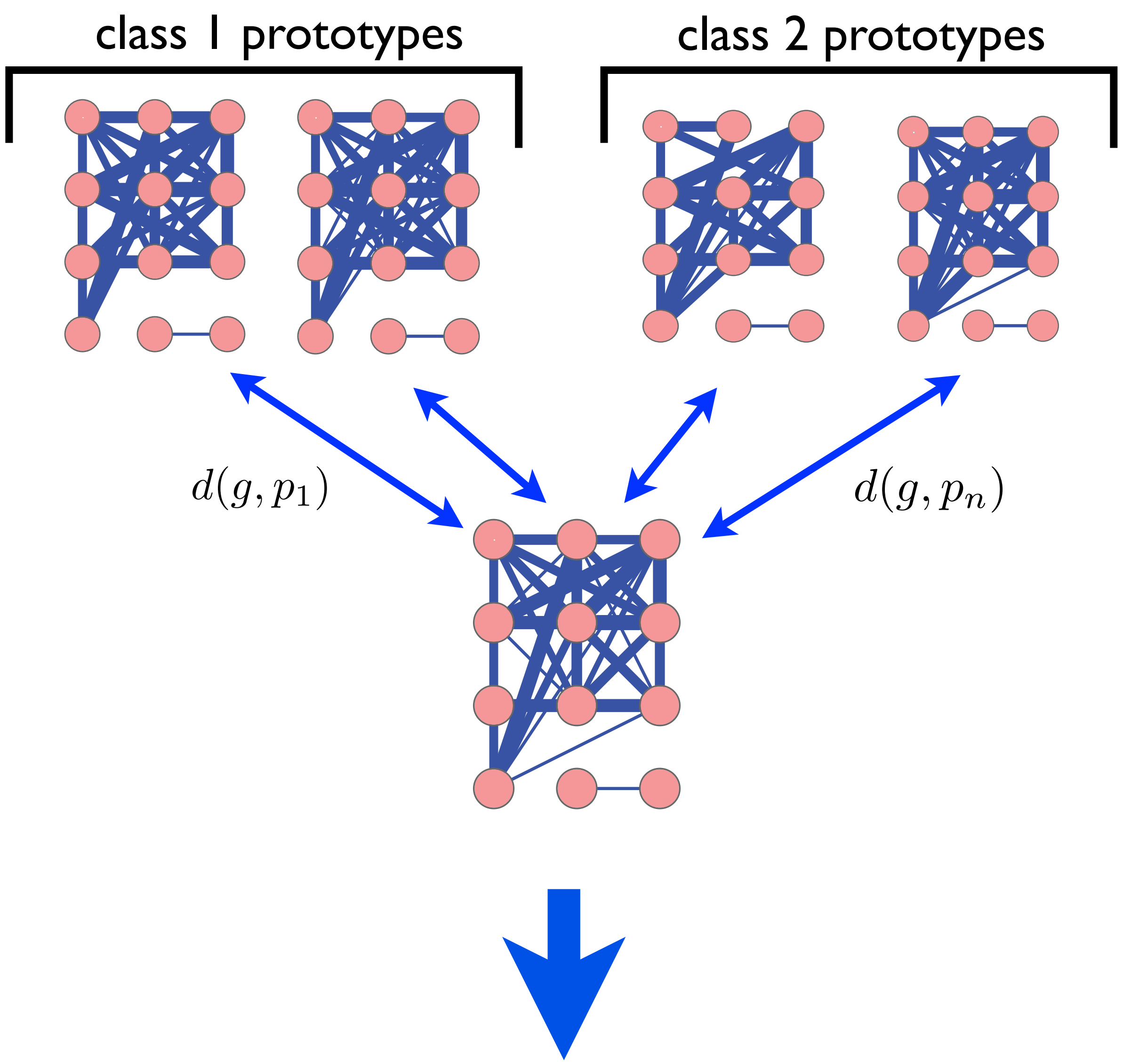


Agenda for this talk

- Preliminaries
- Graph family of interest
- Three different graph embeddings
  - Direct embedding
  - Dissimilarity embedding
  - Graph/vertex attribute embedding

2: Dissimilarity embedding

Principle



Embedding vector

$$\varphi_n^{\mathcal{P}}(g) = (d(g, p_1), \ldots, d(g, p_n)) \in \mathbb{R}^n$$

Fixed dissimilarity

Edge label disssimilarity

$$d(c_{ij}, c'_{ij}) = \begin{cases} |\beta(i,j) - \beta'(i,j)| & c_{ij} \in C, c'_{ij} \in C' \\ K & otherwise \end{cases}$$

Graph dissimilarity

$$d(g,p) = \sum_{i=1}^{|E|} \sum_{j=i+1}^{|E|} d(c_{ij}, c'_{ij})$$

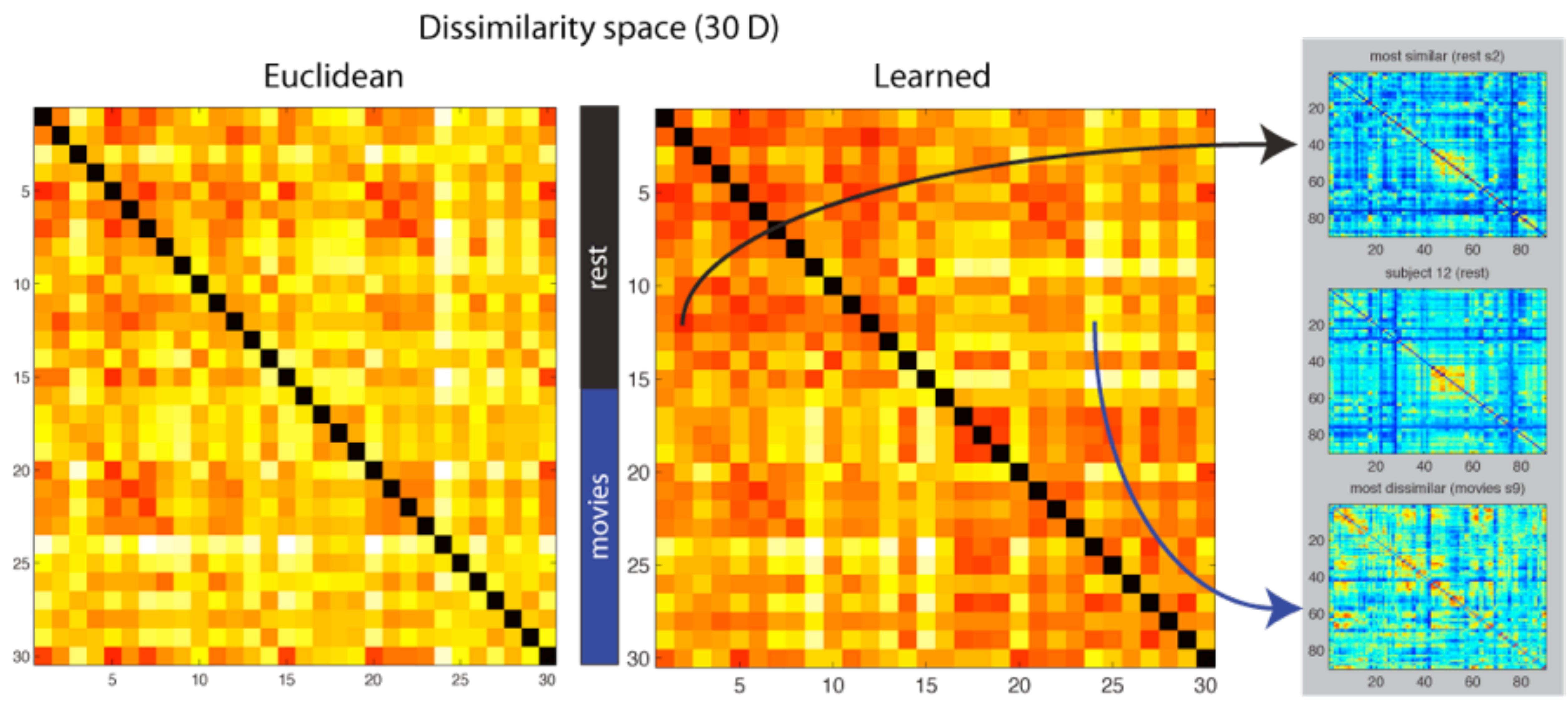
$d(g,p) = \frac{1}{2}||\mathbf{a}_g - \mathbf{a}_p||_1$ (if no missing edges)

Dissimilarity metric learning

$$d(g,p) = ||\mathbf{a}_g - \mathbf{a}_p||_{\mathbf{D}} = \sqrt{(\mathbf{a}_g - \mathbf{a}_p)^T \mathbf{D} (\mathbf{a}_g - \mathbf{a}_p)}$$

[Richiardi et al., ICPR 2010]
based on [Riesen & Bunke, Int. J. Pat. Rec. Artif. Int. 2009]
and [Xing et al. NIPS 2002]

Dissimilarity space



Agenda for this talk

- Preliminaries
- Graph family of interest
- Three different graph embeddings
  - Direct embedding
  - Dissimilarity embedding
  - Graph/vertex attribute embedding

3: "Attributes" of connectivity graphs

- Graphs G, H are isomorphic iff there exists a permutation matrix **P** s.t. $\mathbf{P}\mathbf{A}_g\mathbf{P}^T = \mathbf{A}_h$
  - In our case (atlased connectivity graph): $\mathbf{P} \triangleq \mathbf{I}$
  - Hence connectivity graphs are isomorphic iff

$$\begin{array}{l} \mathcal{E}_g = \mathcal{E}_h \quad and \\ \forall i, j \;\; \beta_g(v_i, v_j) = \beta_h(v_i, v_j) \end{array}$$

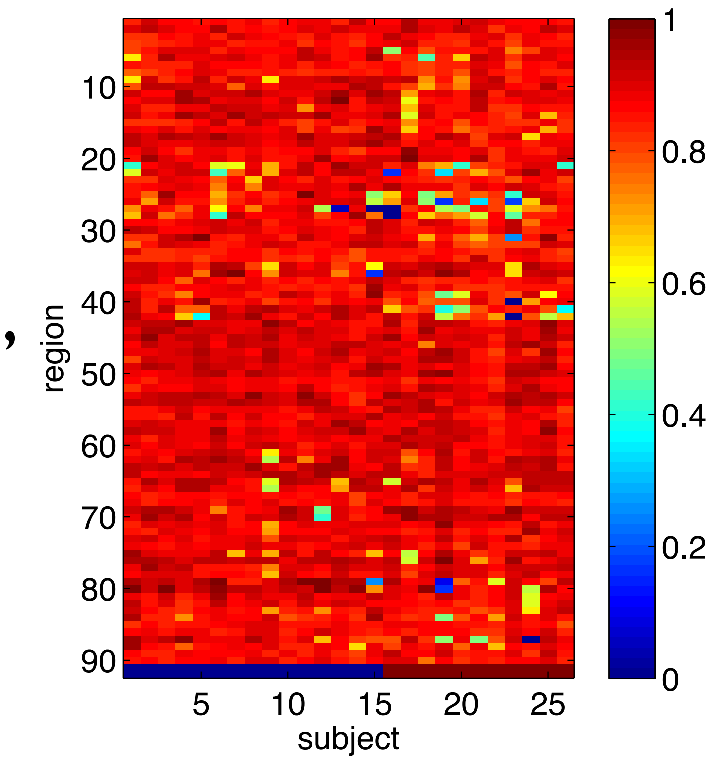
- Graph invariant: *(set of) parameter(s) yielding the same value for isomorphic graphs*
  - To compare noisy connectivity graphs we are more interested in ε-isomorphism, and ε-invariants*
  - Some invariants may degenerate depending on $|\mathcal{V}|$: non-isomorphic graphs may have the same value. Use several invariants**.

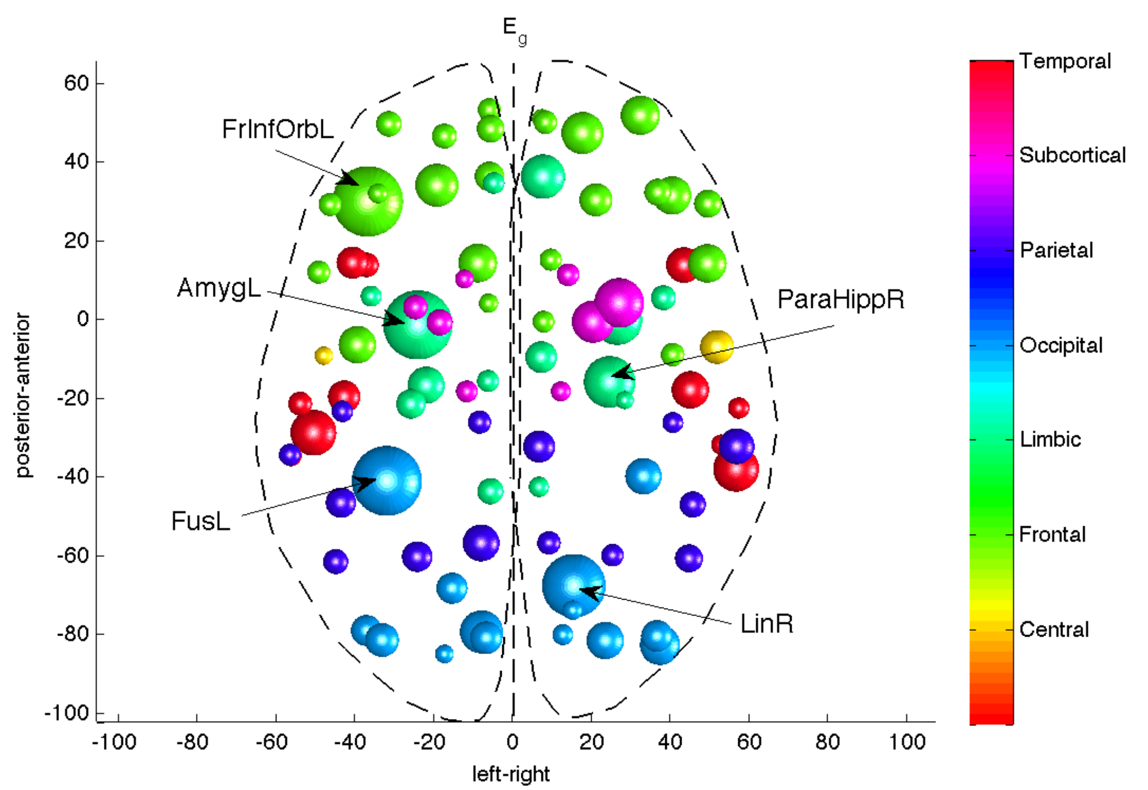
*[Jain & Wysotzki, Neurocomputing, 2005]

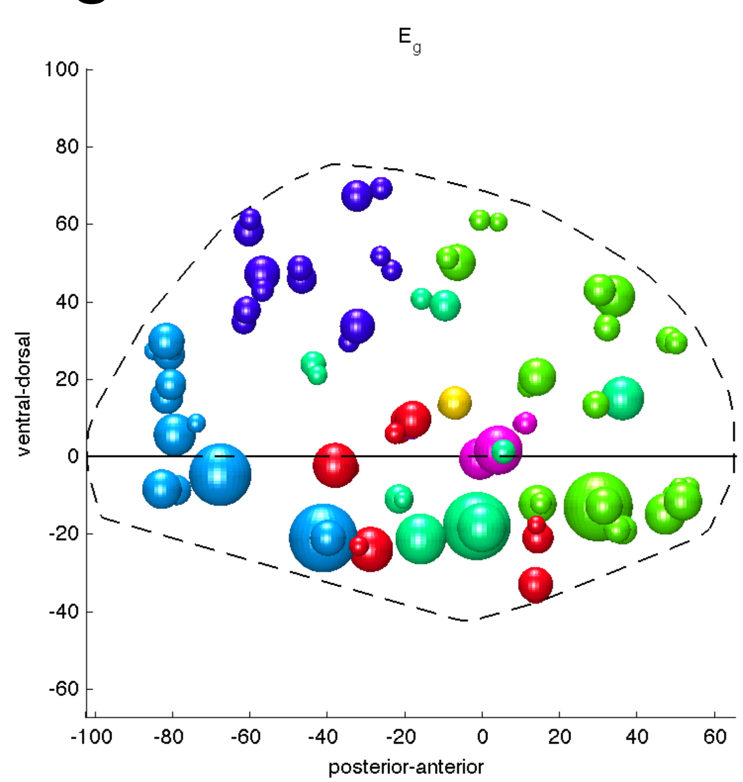
** as in chemometrics: [Bonchev et al, J Comput Chemistry 1981]

Experiments

- Task: inter-subjet age group prediction (15 x 24 y.o. avg vs 11x 67 y.o. avg) from graph/vertex properties of resting-state connectivity graphs.
- Threshold graphs using a fixed and 'range' number of edges, and use {strength, diversity, degree, global efficiency, and local efficiency}
- Results: only global and local efficiency are convincing (up to 89% accuracy (CI 71-96%)). But on this dataset this works better than direct embedding.







- Orbito-frontal cortex, amygdala, and parahippocampal formation are relatively the most predictive regions (broadly agrees with previous studies*)
- In addition, the lingual gyrus shows age-related activation changes during memory tasks

[Richiardi et al., PRNI, 2011]

*[Achard & Bullmore, PLoS CompBiol, 2007]

Summary: pros and cons

- Direct embedding:

  \+ satisfactory prediction on several datasets

  \+ easy mapping of discriminative pattern

  \- curse of dimensionality!

- Dissimilarity embedding:

  \+ low-dimensional representation (O(N))

  \+ custom dissimilarity metrics promising, on the way to graph kernels

  \- performs worse than direct embedding on most datasets

- Graph/vertex attribute embedding:

  \+ low-dimensional representation (O(|V|))

  \+ interpretable in terms of network properties

  \+ yields "deep"(ish) features

  \- many attributes are weakly discriminative

Final Thoughts

- Learning with connectivity graphs is useful for a range of cognitive and clinical neuroscience problems
  - Complementarity with BOLD activation modelling is clear (focuses on functional integration)
  - We can visualise and interpret results both in terms of connections and in terms of regions
  - Atlasing imposes some restrictions but there is plenty of room
  - We can trivially restrict analysis to small subnetworks (e.g. speech processing areas)
- Much work to do: physiological noise, modelling, and LF oscillations interpretation

Thanks

- Medical Image Processing Lab, EPFL/ U. of Geneva
  - N. Leonardi, D. Van De Ville
- LabNIC, U. of Geneva
  - P. Vuilleumier, S. Schwartz, H. Eriylmaz, M. Van Der Meulen
- Neurology, University hospital of Lausanne
  - M.Gschwind, S. Simioni, J-M. Annoni, M. Schluep
- CIBM, Geneva University Hospitals
  - F. Lazeyras
- Merck-Serono Research Alliance
  - B. Greco, P. Hagmann
- GIPSA-Lab, Institut National Polytechnique de Grenoble
  - Sophie Achard
- Brain Mapping Unit, University of Cambridge
  - Ed Bullmore
- Inst. of Computer Science and Applied Mathematics, U. of Bern
  - H. Bunke, K. Riesen