Eulerian tour algorithms for data visualization and the PairViz package

Catherine Hurley
NUI Maynooth

R.W. Oldford
U. Waterloo

NUI MAYNOOTH
Ollscoil na hÉireann Má Nuad

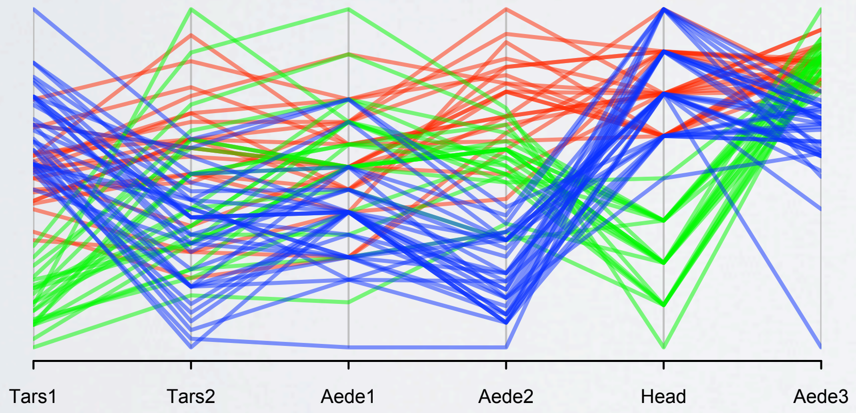
UNIVERSITY OF
Waterloo

July 8 2009 UseR!

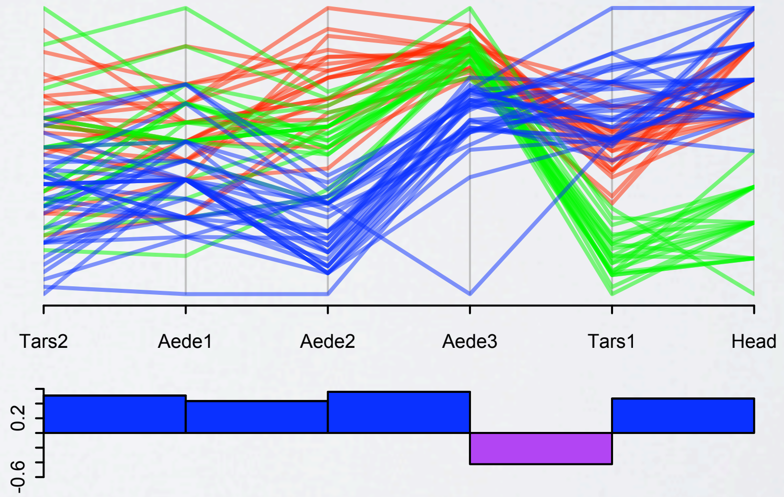
Graphics: Effect Ordering

- Packages: **seriation, gclus, corrgram**
- Example: PCP Flea data

Standard order



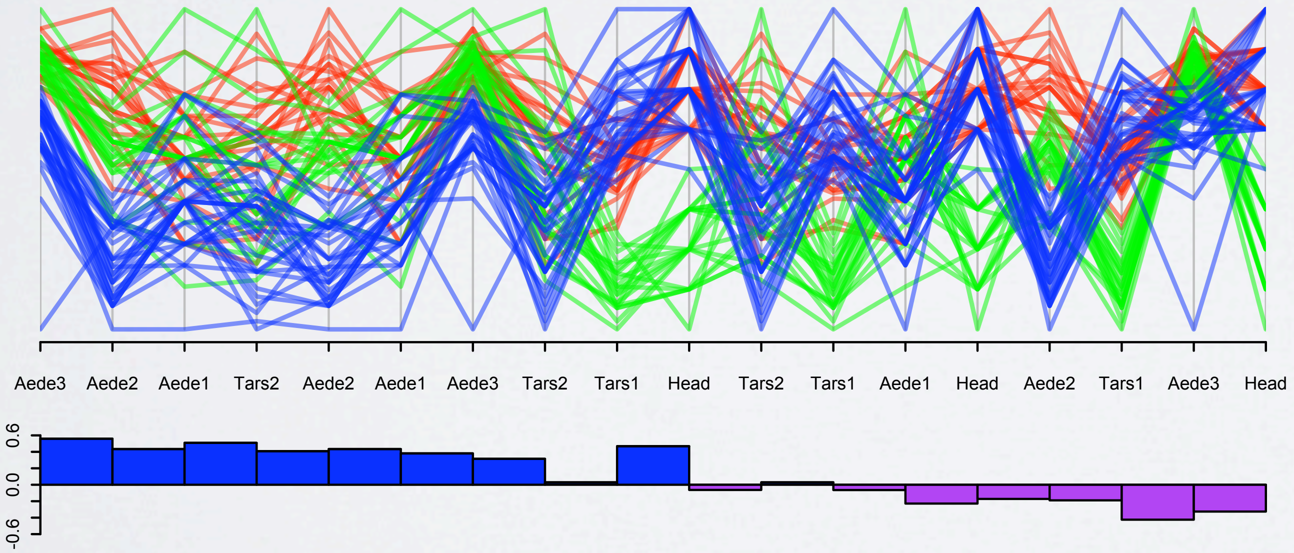
Correlation order



Pairviz: relationship ordering

- Statistical graphics are about comparisons

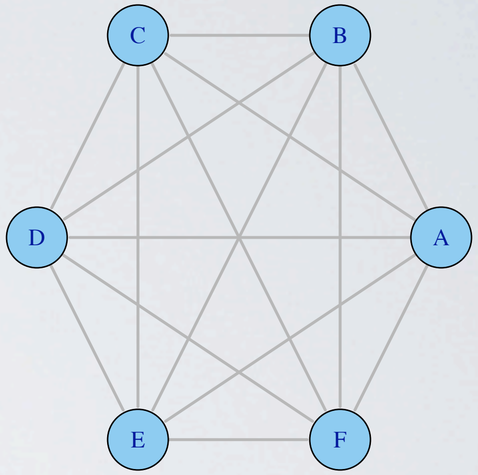
  between variables, cases, groups, models



Flea data: correlation order

A graph model

- Build a graph where nodes are statistical objects



- Edges are relationships

- Example:

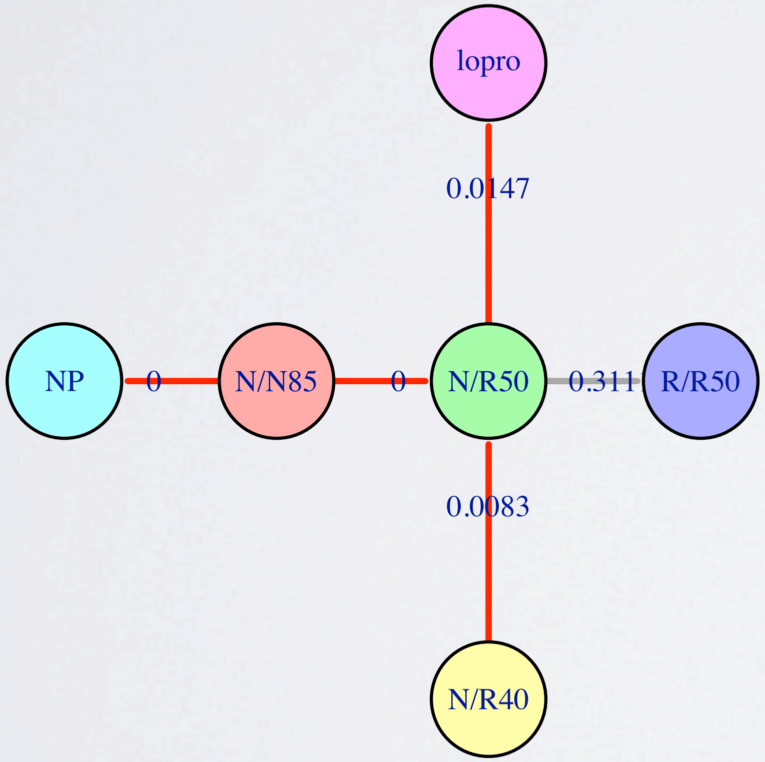
Node	Vis	Edge	Vis
Group	Boxplot	Two groups	CI for mean diff
Var	Hist	Two vars	Scatterplot
2 vars	Scat	4-d space	Dynamic scat
Model	Resid	2 Models	PCP

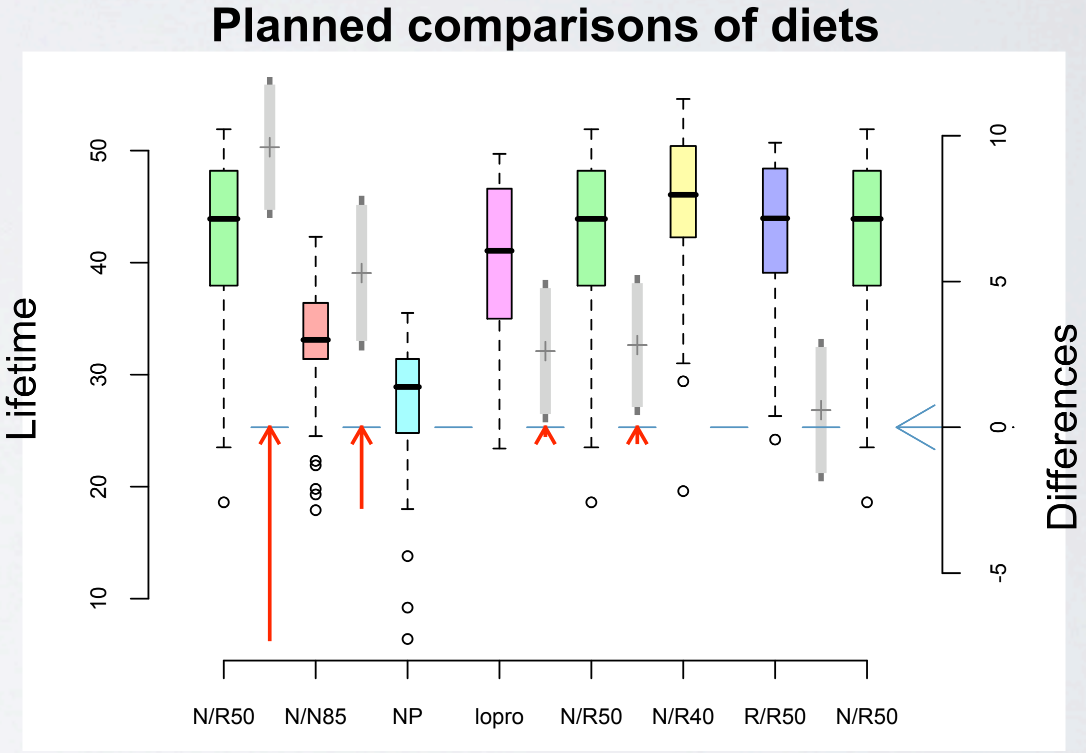
Example: planned comparisons

Mice in 5 diet groups, response is lifetime

Nodes are treatments, edges are planned comparisons

Weights are p-values

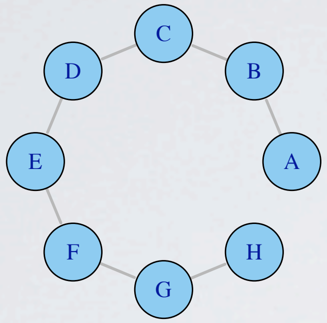




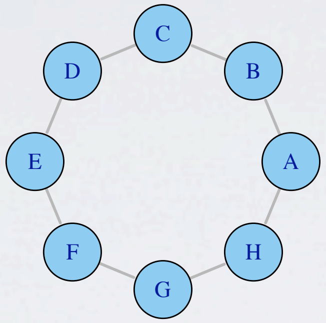
Reducing calories and protein increases lifetime

Graph Traversal

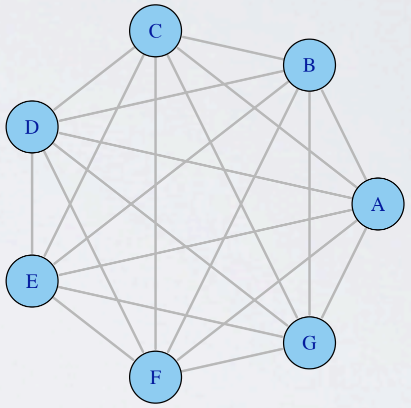
- Traverse all nodes: hamiltonian path



Open hamiltonian path



Closed hamiltonian path



Closed eulerian path on K_7

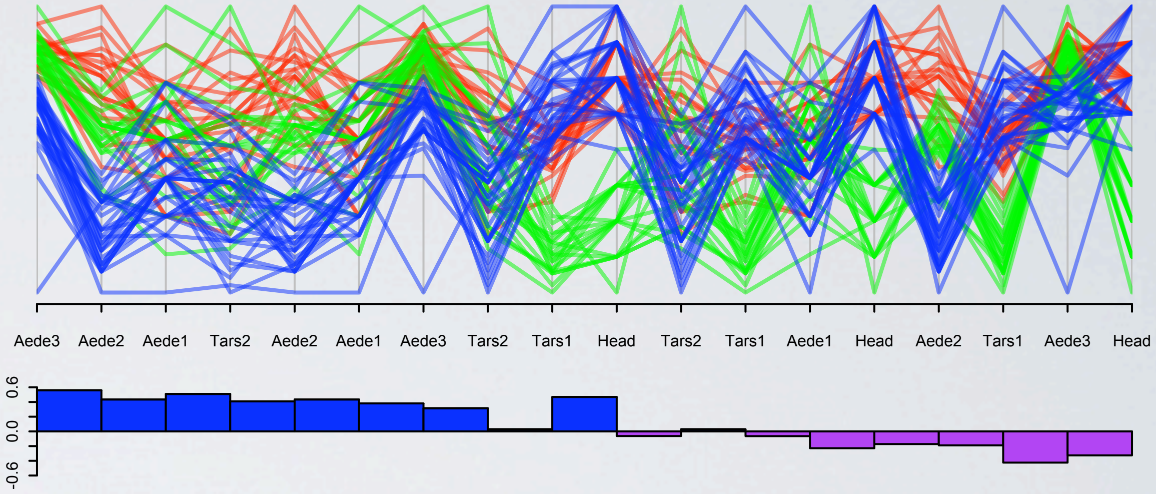
- Traverse all edges: eulerian path

- Use **gclus, seriation**: hamiltonian paths on complete graphs

- **PairViz:** eulerian paths

Graph Structures

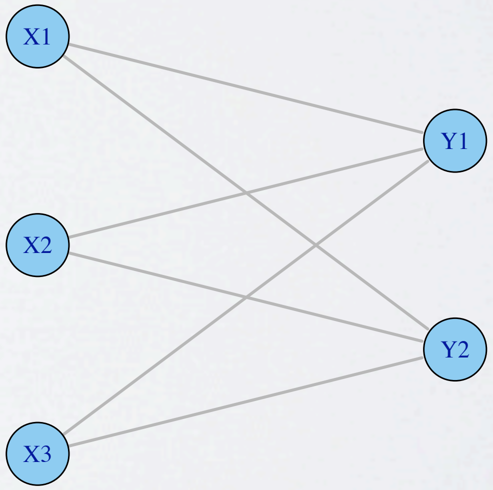
- Complete graph: all comparisons are interesting



- Edge-weighted graphs: low weight edges are more interesting

Weight edges by 1-corr, eulerian follows low weight edges

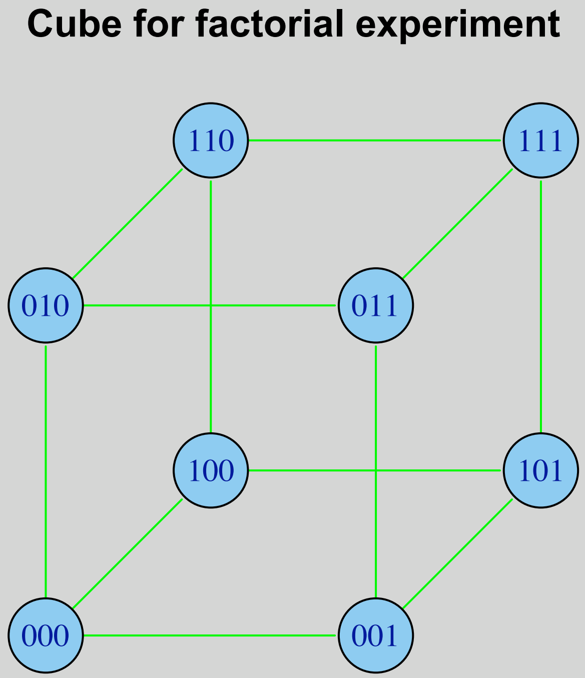
- Bipartite graph



eg only treatment-control comparisons are of interest

Graph Structures- cont'd

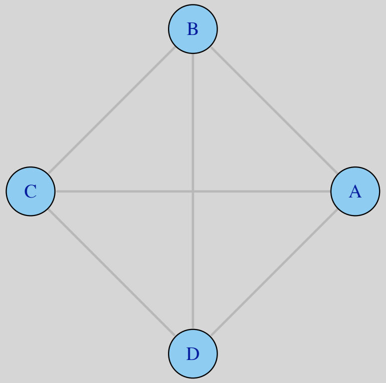
- Hypercube graph



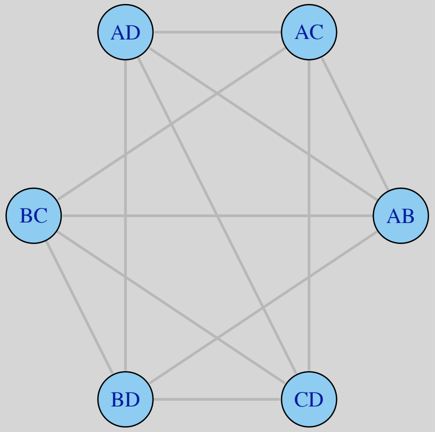
or model selection:
Each node in G is a predictor subset
edge: add/drop predictor

- Line graph

transform G



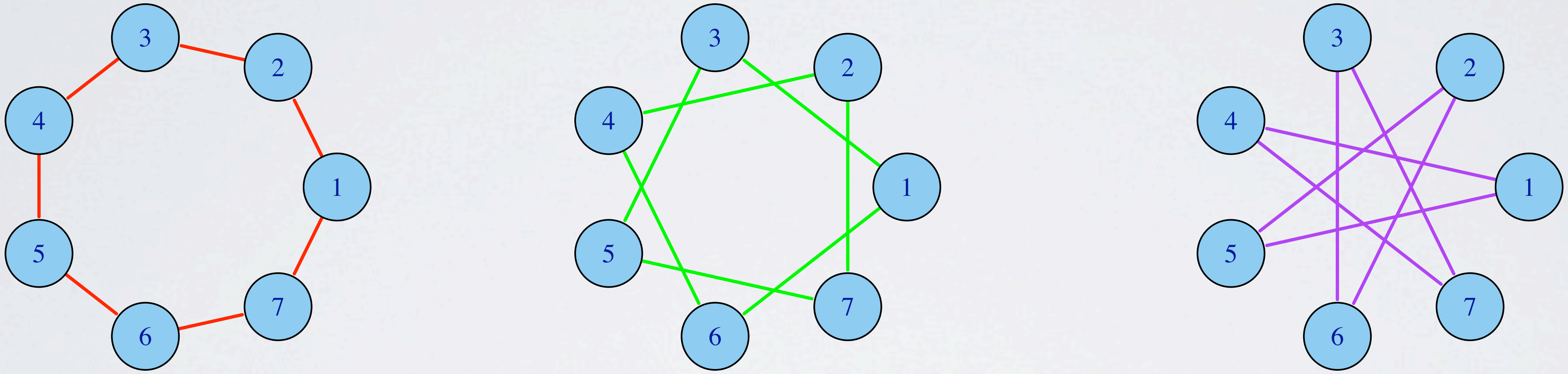
to L(G)



eg Each node in G is a var,
each node in L(G) is var pair,
edge is 3-d transition

Algorithms- Complete graph

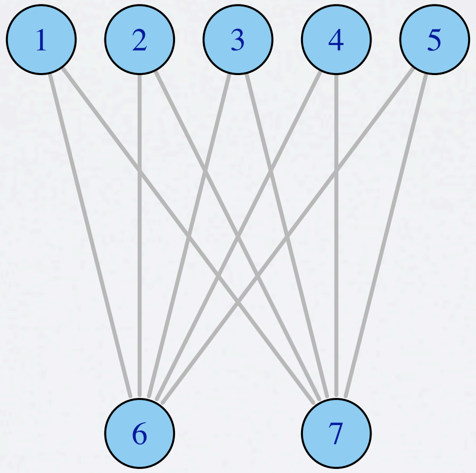
- Closed eulerian path exists when each node has odd number of vertices: ie for K_{2n+1}
- Hamiltonian decomposition of graph
    - into hamiltonian cycles: eulerian for K_{2n+1}



    - into hamiltonian paths: approx eulerian for K_{2n}
- classical algorithm: **hpaths**
- WHam: **weighted_hpaths:** pick best for H_1, best orientaton and order for others.

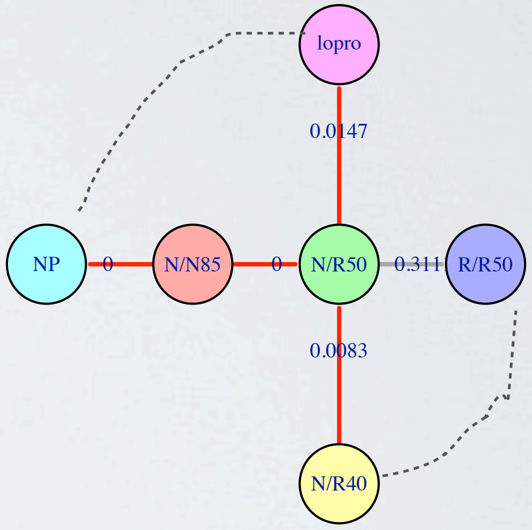
Algorithms-Complete graph cont'd

- Recursive algorithm: **eseq**:
- Start with eulerian on K_n, append edges to get eulerian on K_{n+2}



Algorithms- general

- Eulerian graph: connected, all nodes have even number of edges



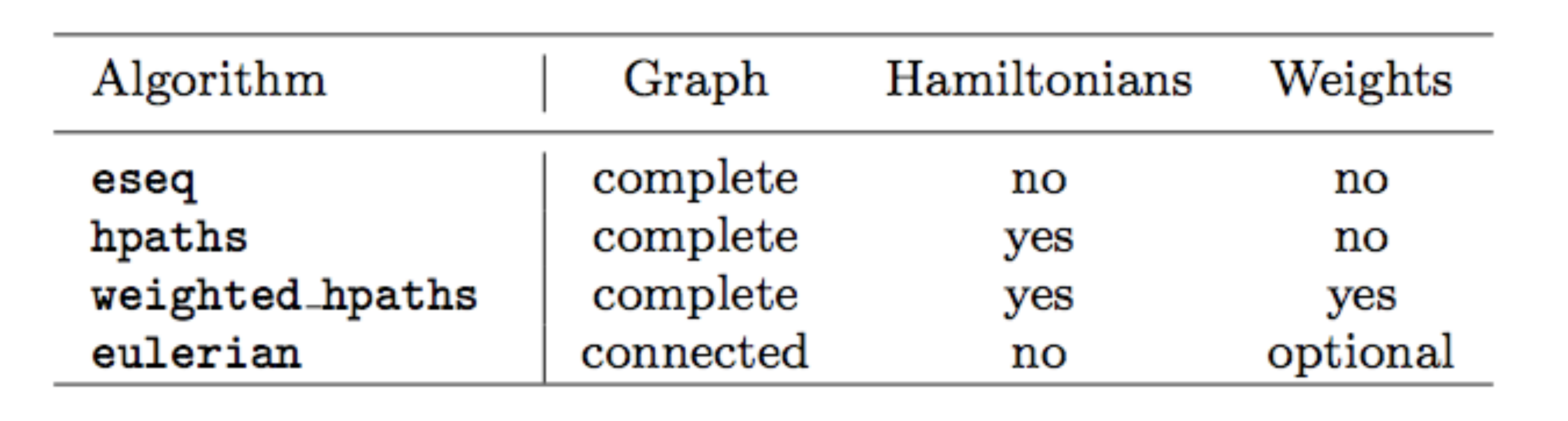
- Otherwise, add edges, pairing up odd nodes

  Chinese postman does this in optimal way

- Classical algorithm (Hierholzer, Fleury)

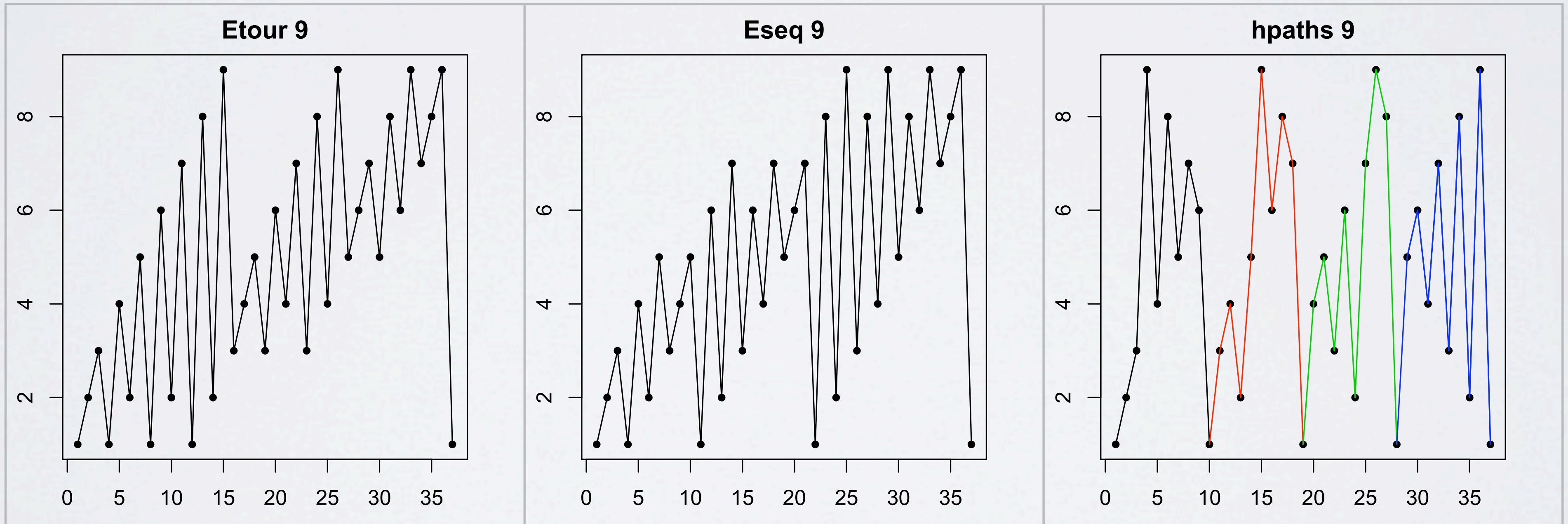
- Our version GrEul, (**etour**) follows weight increasing edges

Algorithms comparison



Algorithm	Graph	Hamiltonians	Weights
eseq	complete	no	no
hpaths	complete	yes	no
weighted_hpaths	complete	yes	yes
eulerian	connected	no	optional

Complete-no weights



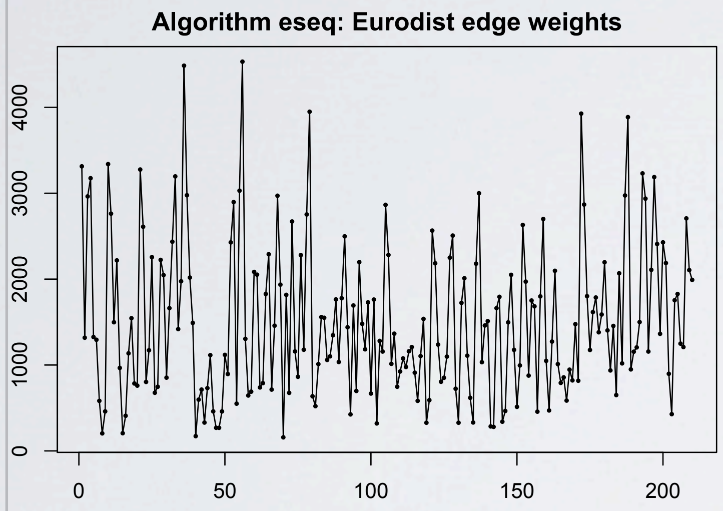
prefers low vertices

prefers low edges

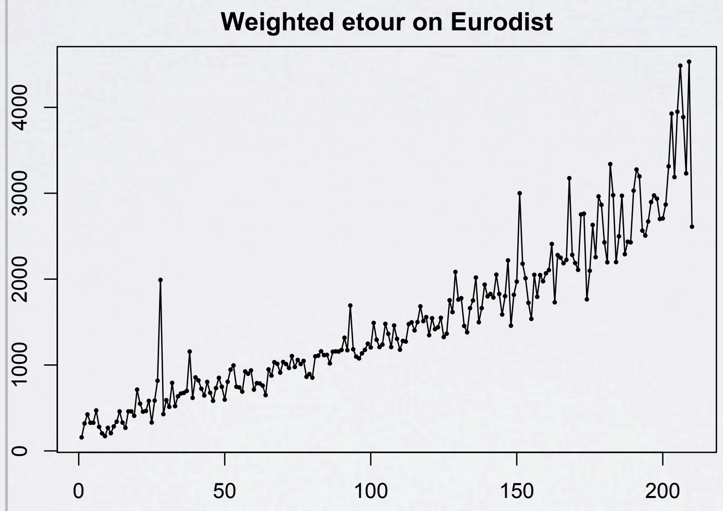
4 hamiltonians

Algorithms: complete, weighted

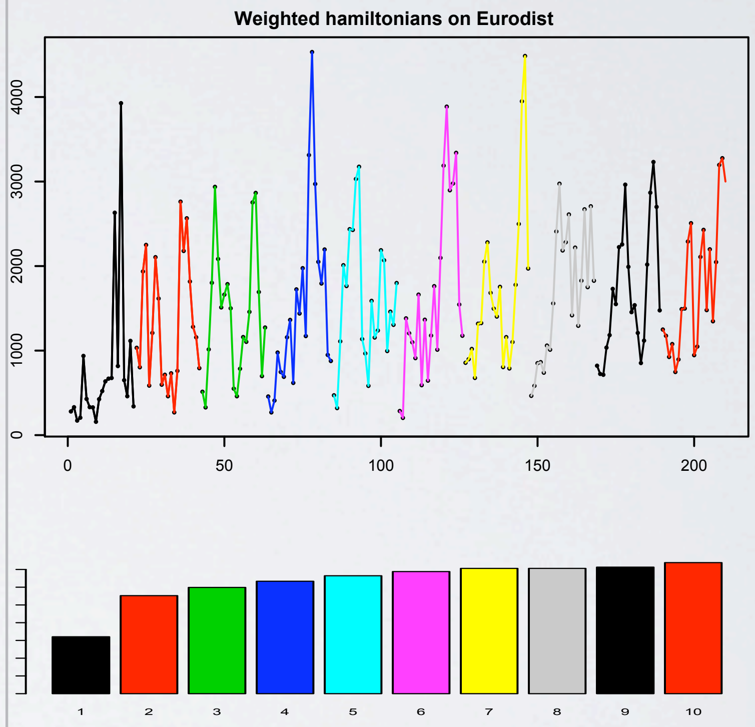
Eurodist: 21 European cities



ignores weights



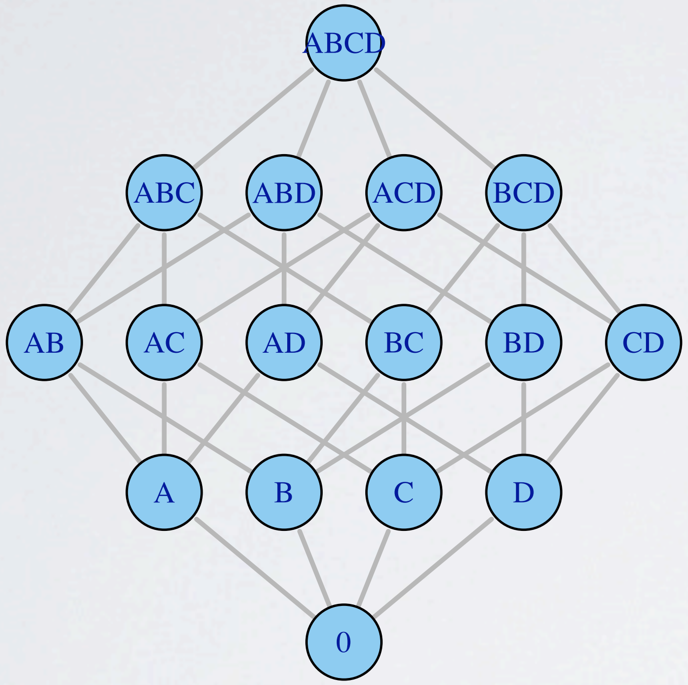
Starts in Geneva



hamiltonian decomp, with increasing path lengths

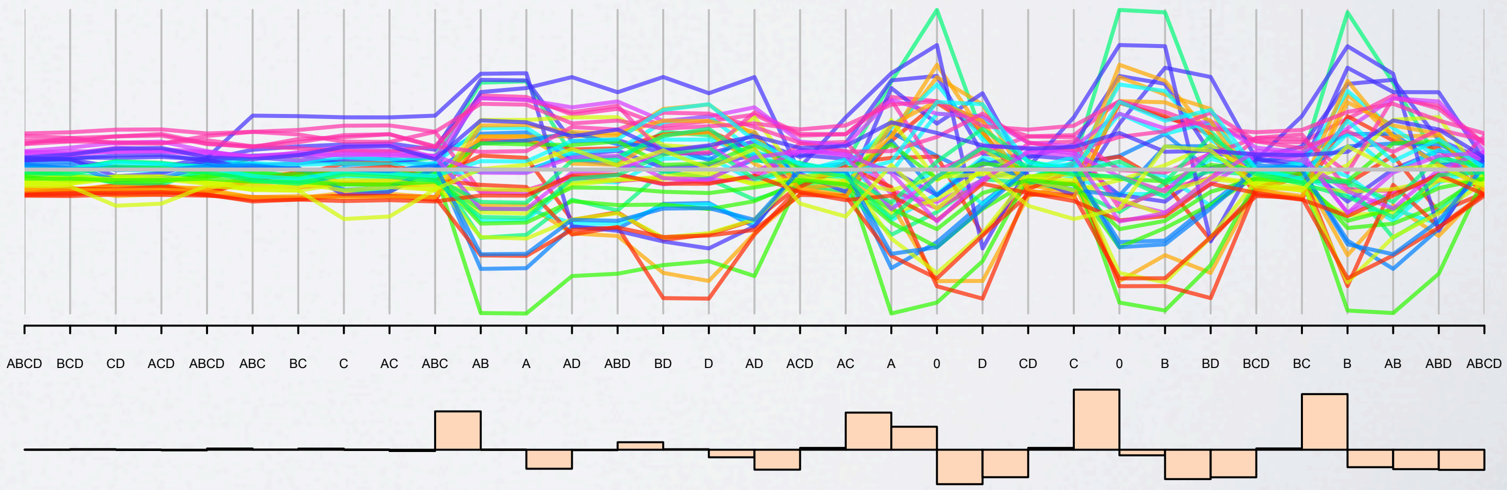
Example: model selection

Mammal sleep data
Y= log brain wt.
Predictors A= non dreaming sleep, B=dreaming sleep,
C=log body wt, D=life span



- Hypercube graph represents possible moves in a stepwise regression algorithm
- Graph Q_n is hamiltonian, and eulerian for even n
- Edge weights: change in SSE

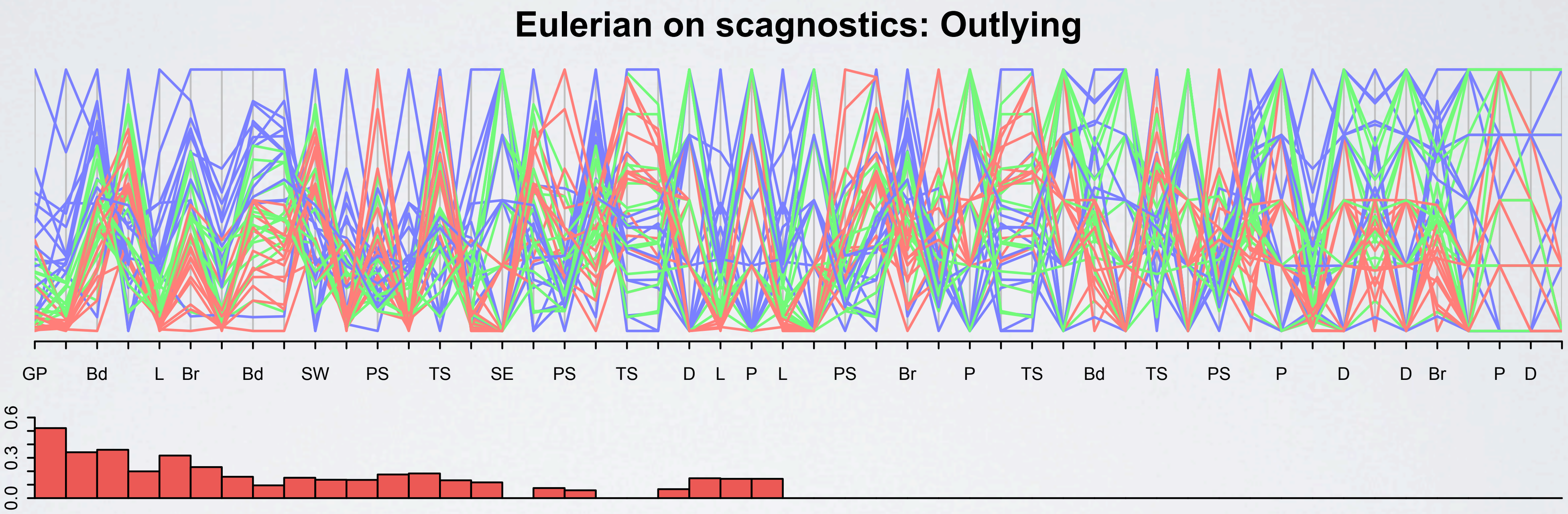
Sleep data: Model residuals.



- Eulerian starting with full model
- All models with C are good
- Bar chart: change in SSE

More variables

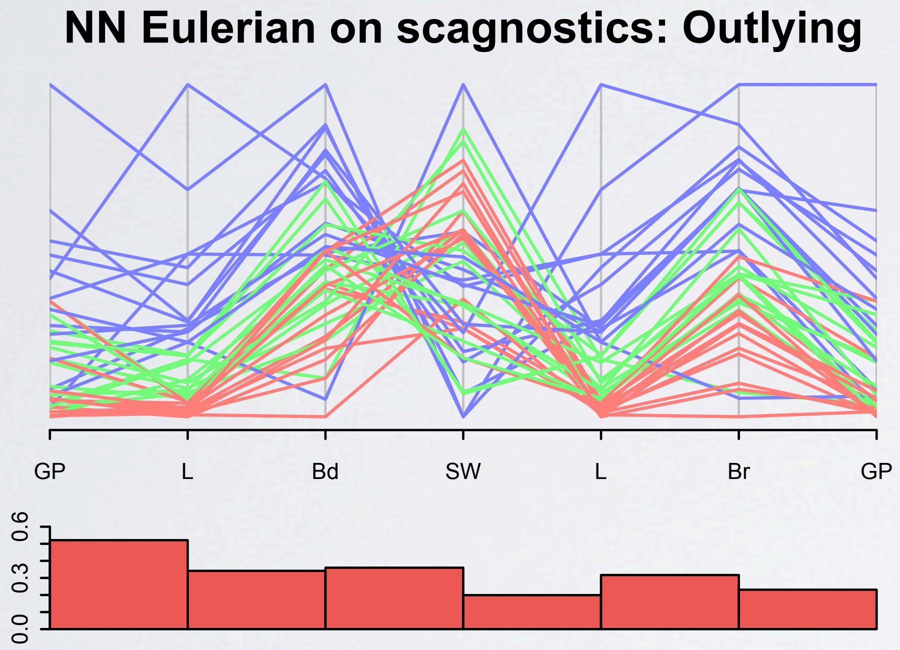
Sleep data: 10 vars (nodes)
45 edges
Eulerian has length 50

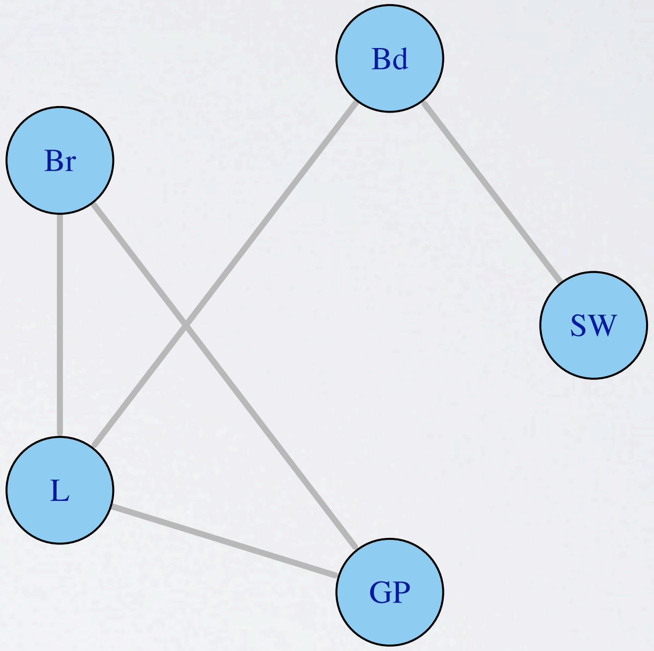


Using outlying index from scagnostics package for eulerian traversal
zoom on first half of display

More variables-cont'd

Reduce the graph
NN graph: eliminate edges with outlier index < .2





Reduces graph from 10 to 5 nodes, and 45 to 5 edges
Other nodes have no edges

IN CONCLUSION..

- **Pairviz** package: relationship ordering for data visualisation
- Current version: algorithms presented here
- Thanks to **graph, igraph**
- Work in progress: ordering dynamic visualisations via **ggobi**.

with Adrian Waddell, UW