Analysis of bio-molecular networks through semi-supervised graph-based learning methods

Matteo Re, Marco Mesiti, Marco Frasca,
Jianyi Lin, Giorgio Valentini

Computer Science
Department

UNIVERSITÀ
DEGLI STUDI
DI MILANO

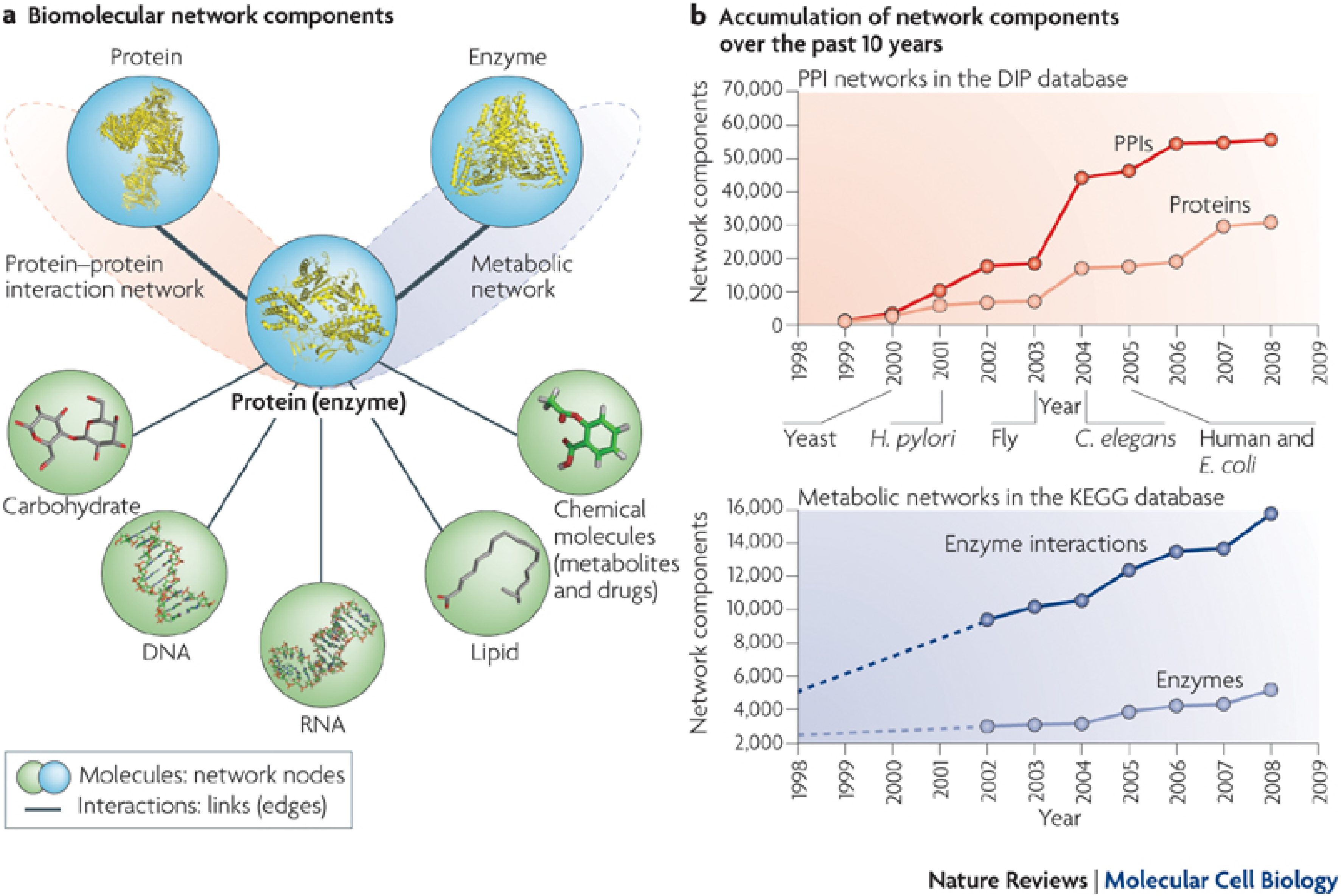
Anacleto
Lab

Computational Biology and Bioinformatics

Outline

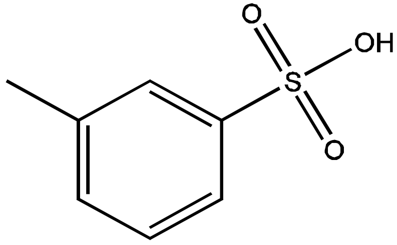
- Relevant problems in molecular biology and medicine can be modeled through graphs
- The node labeling and ranking problem in complex biological networks
- Merging local and global learning strategies: the kernelized score functions algorithmic scheme
- Analysis of huge biological networks with off-the-shelf machines: results and perspectives

Biomolecular networks

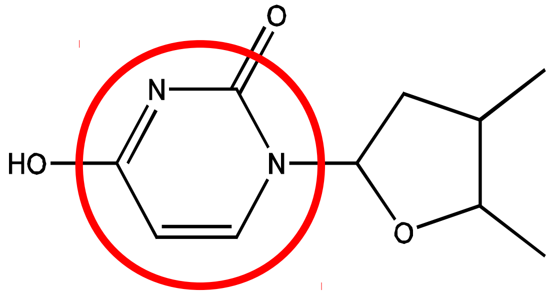


Drug repositioning

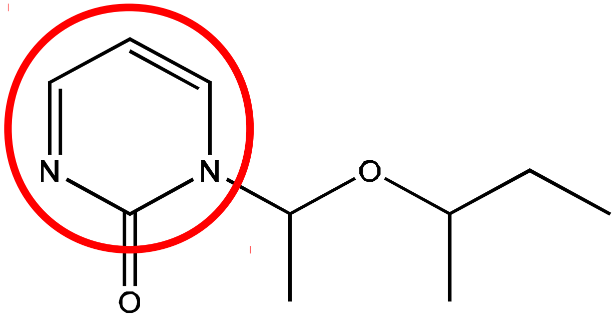
Given a collection of molecules



(A)



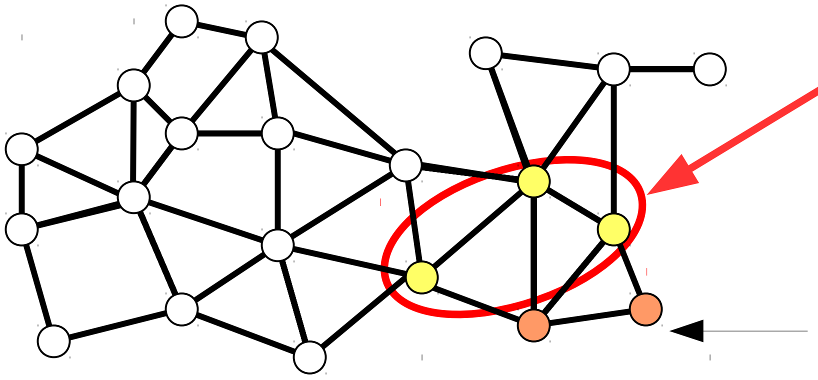
(B)



(C)

Find a meaningful way to express a similarity between them (i.e. binary profiles indicating the presence/absence of substructures used as proxy for the computation of a global similarity score between each pair of molecules).

Nodes: drugs
Edges: similarity between drugs



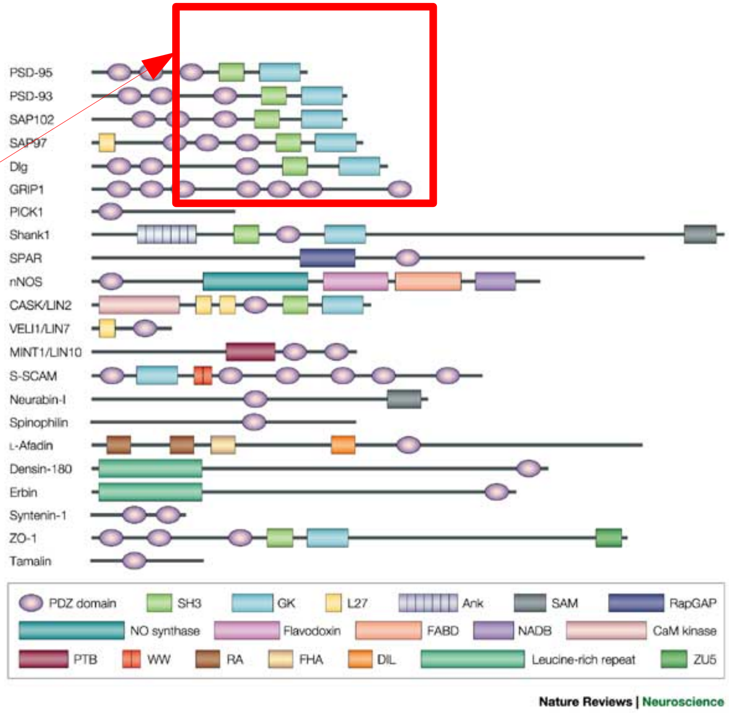
The **most similar** nodes (drugs) are candidates for the development of novel anticonvulsant drugs

Seed node, a **marketed** drug (i.e. anticonvulsant)

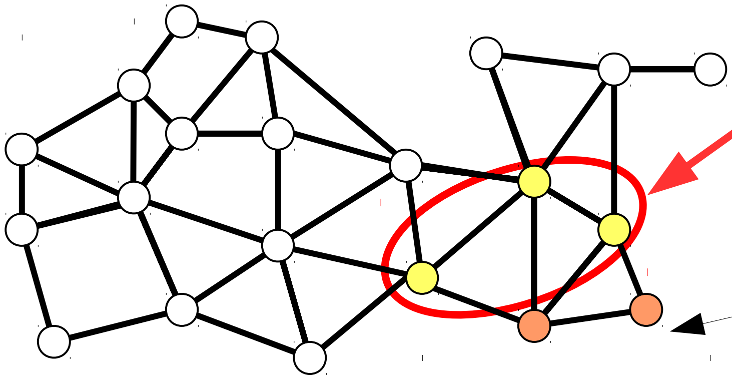
Automated Function Prediction (AFP)

Given a collection of proteins.

Find a meaningful way to express a similarity between them (i.e. binary profiles indicating the presence/absence of protein domains, 3D structure signatures, presence/absence of catalytic groups used as proxy for the computation of a global similarity score between each pair of ptoreins).



Nature Reviews | Neuroscience

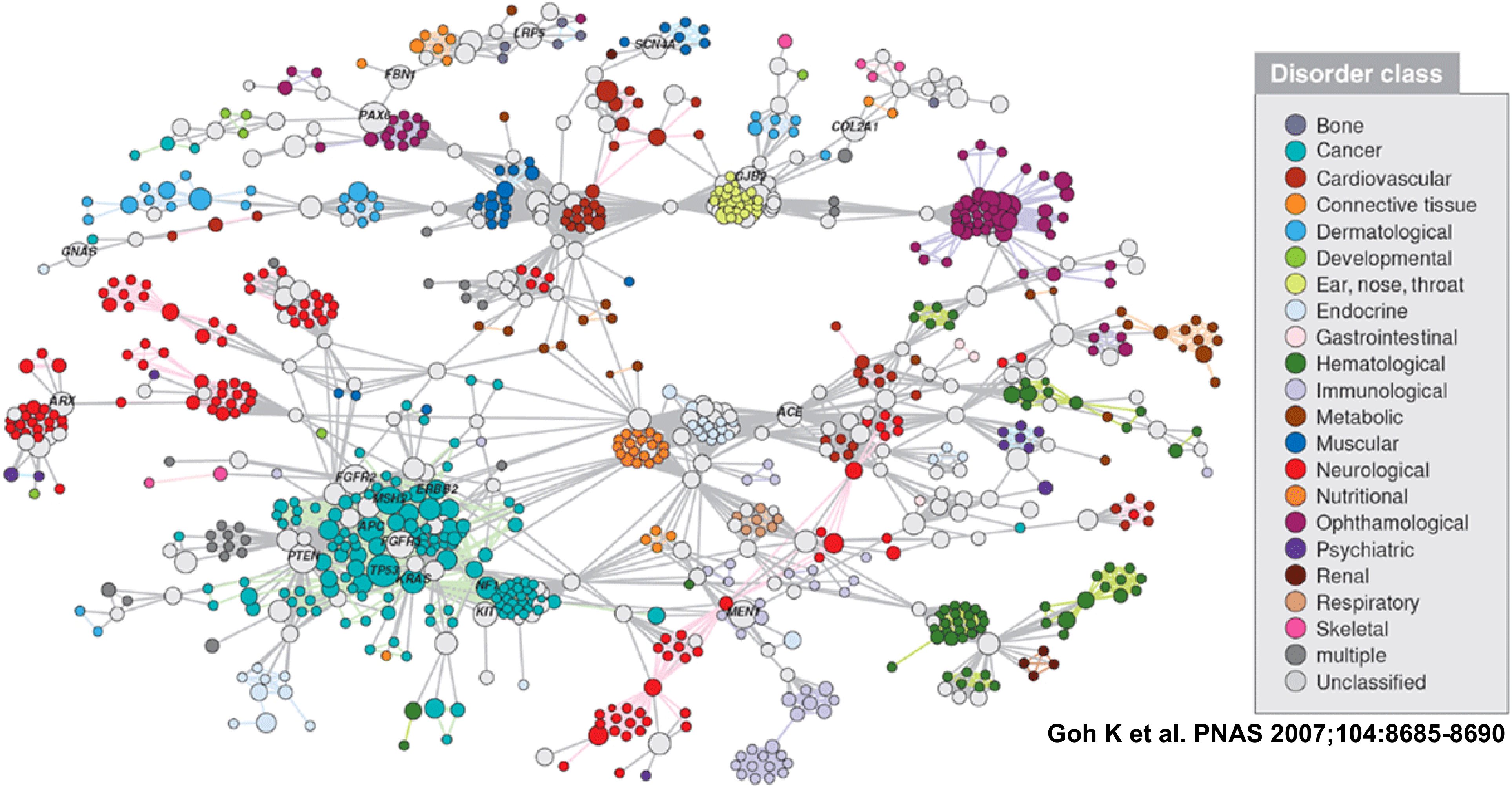


The **most similar** nodes (proteins) are candidates for the association to the functional term associated to the seeds

Seed node, associated to a **functional vocabulary term** (i.e. Gene Ontology)

Disease gene networks

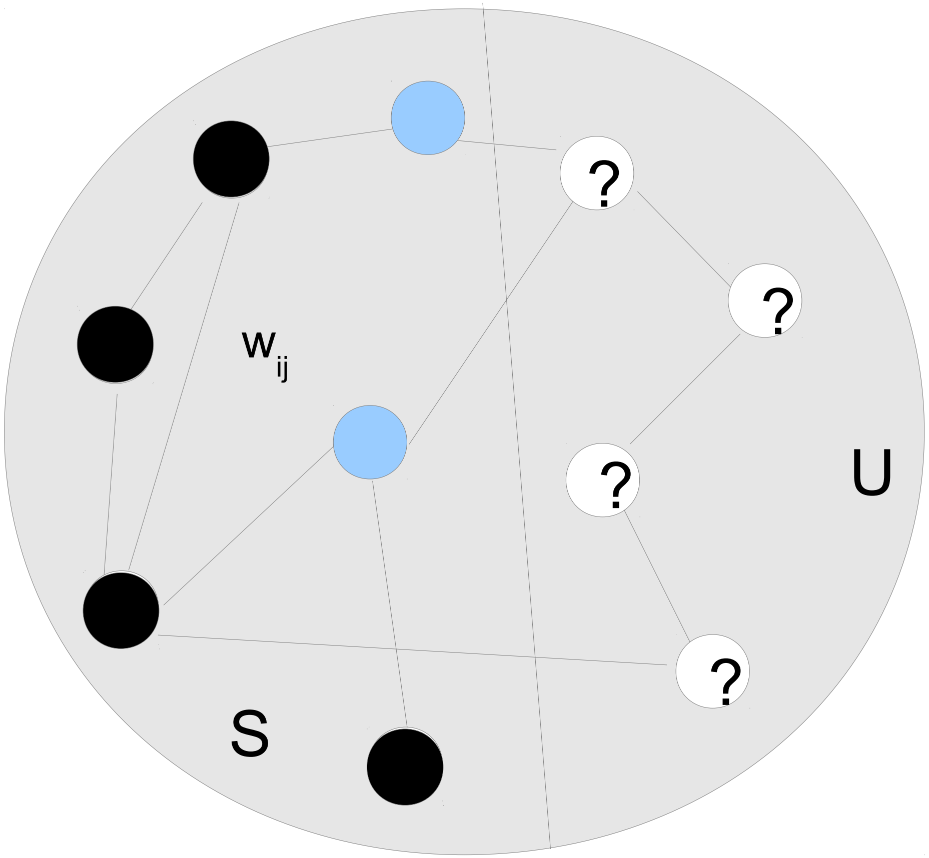
Given a collection of genes. Build a network whose nodes (genes) are connected only if they are involved into disorders of the same class.



Goh K et al. PNAS 2007;104:8685-8690

Graph Semi-Supervised Learning (GSSL) problem

$G = <V,E>$



V : proteins,genes,drugs,...

E : functional similarities/relationships

W : similarity matrix

S : labeled nodes

U : unlabeled nodes

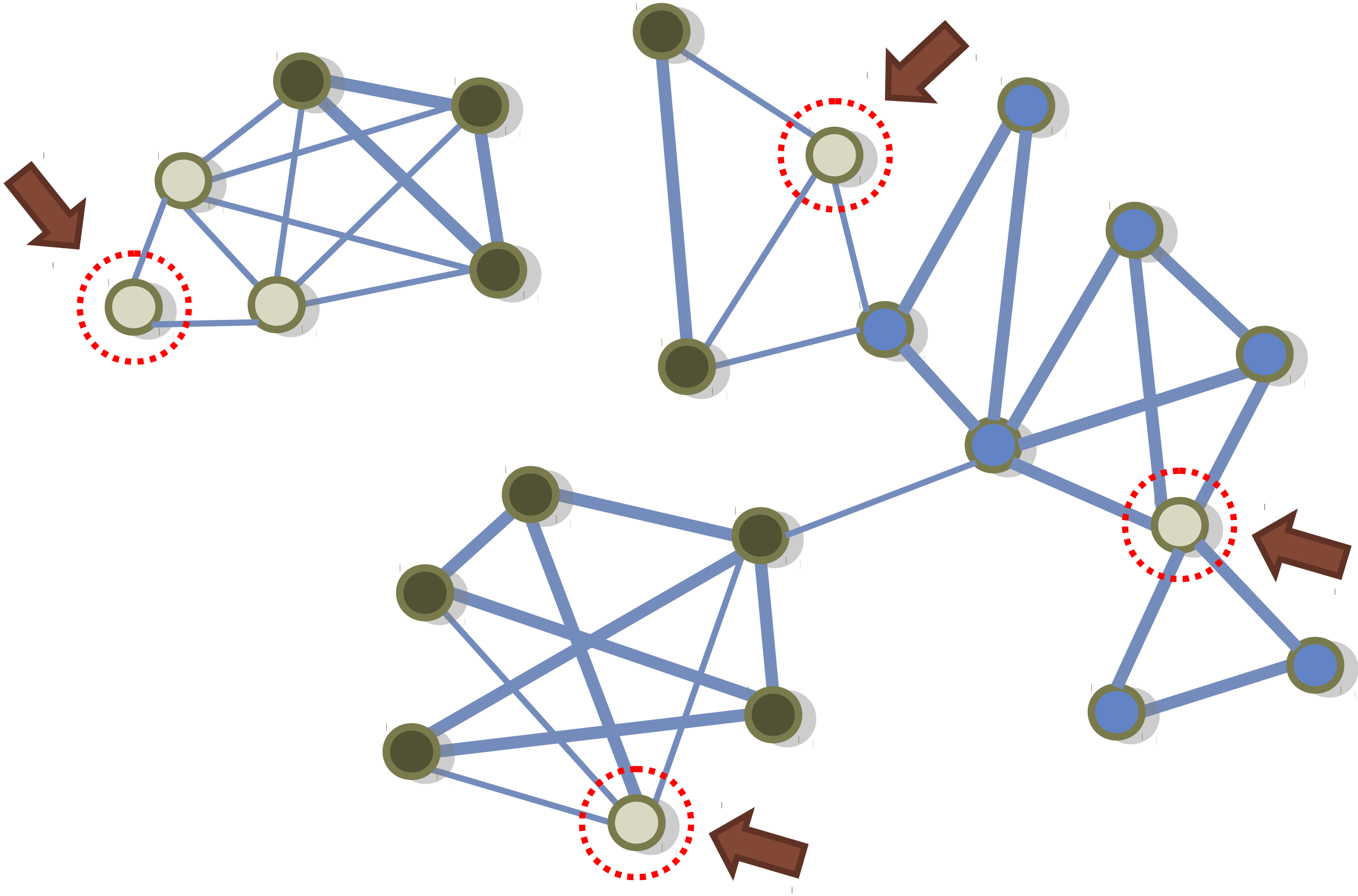
GOAL: predict labels for unlabeled nodes (*labeling problem*) or rank nodes with respect to the class to be predicted (*ranking problem*)

State-of-the-art node labeling/ranking methods in computational biology

- Guilt by association (*Marcotte* et al., 1999, *Oliver* et al. 2000)
- Evaluation of functional flow in graphs (*Vazquez* et al. 2003)
- Hopfield network-based methods (*Karaoz* et al. 2004, *Bertoni et al*. 2011)
- Local learning and weighed integration (*Chua* et al 2007)
- Label propagation based on Markov fields (*Deng* et al. 2004)
- Kernel methods for semi-supervised learning and integration of networks (*Tsuda* et al. 2005, *Borgwardt et al.* 2011)
- Label propagation based on Gaussian random fields and ridge regression (*Mostafavi* et al. 2008)
- Random walk-based algorithms (*Kohler et al*., 2008, *Bogdanov* and *Singh*, 2010)
- ...

Local learning strategy:

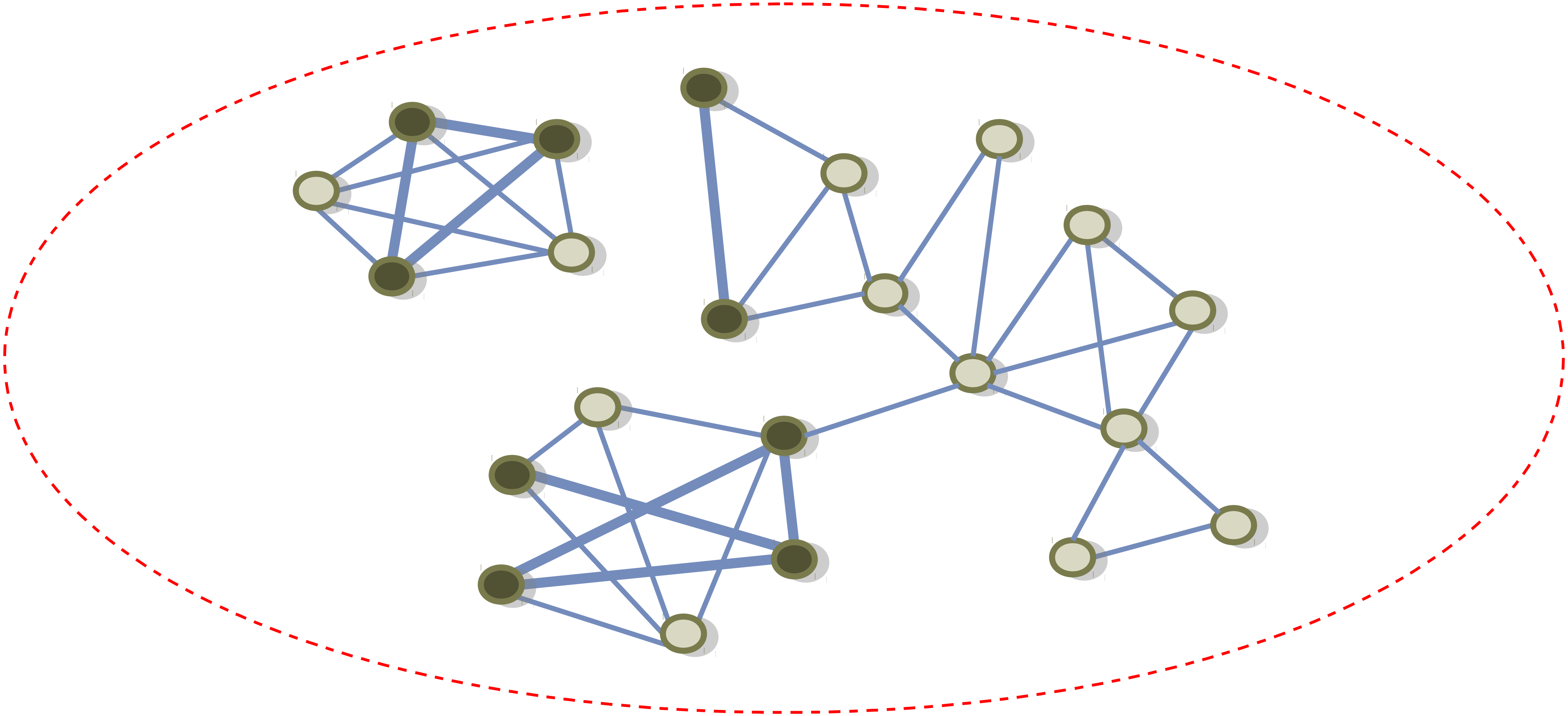
Guilt-by-association (*Marcotte* et al., 1999, *Oliver* et al. 2000)



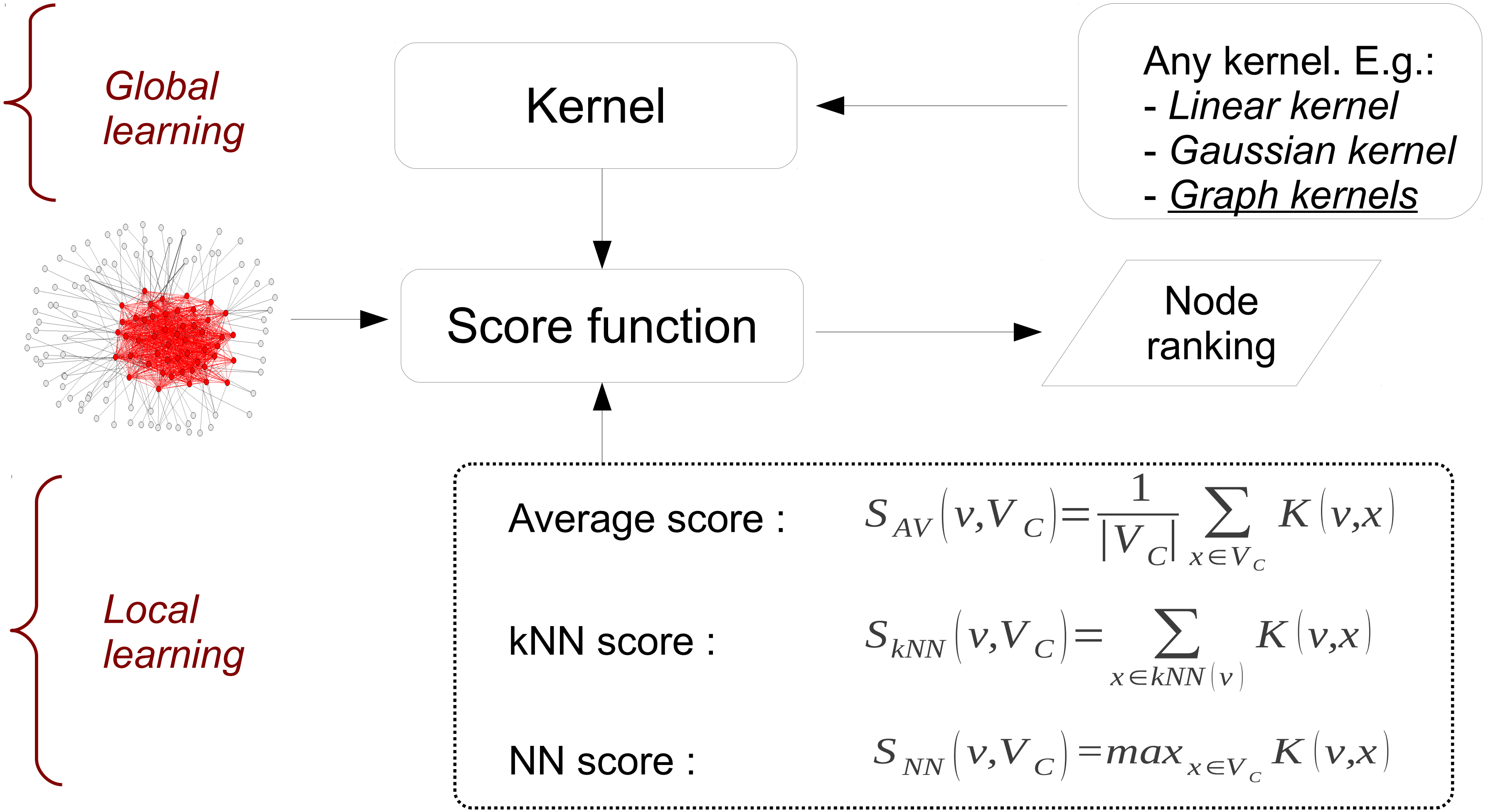
Global learning strategy:

Exploitation of the overall network topology

(*Karaoz et al. 2004, Bengio et al. 2008, Borgwardt et al. 2011*)



Kernelized score functions: putting together local and global learning strategies (*Re et al. 2012*)



Global learning

Kernel

Any kernel. E.g.:
- *Linear kernel*
- *Gaussian kernel*
- *Graph kernels*

Score function

Node ranking

Local learning

Average score : $S_{AV}(v,V_C)=\frac{1}{|V_C|}\sum_{x\in V_c} K(v,x)$

kNN score : $S_{kNN}(v,V_C)=\sum_{x\in kNN(v)} K(v,x)$

NN score : $S_{NN}(v,V_C)=max_{x\in V_c} K(v,x)$

Example of a kernel well-suited to capture the topology of the graph: the Random Walk Kernel (Smola and Kondor, 2003)

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \quad d_{ii} = \sum_j w_{ij}$$

Normalized graph Laplacian

$$\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{D}^{-\frac{1}{2}} (\boldsymbol{D} - \boldsymbol{W}) \boldsymbol{D}^{-\frac{1}{2}} =$$

$$\boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{D} \boldsymbol{D}^{-\frac{1}{2}} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}$$

$$\boldsymbol{K}_{rw} = a\boldsymbol{I} - \tilde{\boldsymbol{L}} = a\boldsymbol{I} - \boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}} =$$

$$(a-1)\boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}$$

1 - step RW kernel

$$\boldsymbol{K}_{rw}^{q} = (a\boldsymbol{I} - \tilde{\boldsymbol{L}})^{q}$$

q - step RW kernel

Derivation of kernelized score functions

$\phi \; : \; X \;\rightarrow\; \mathcal{H}$

$$D_{AV}(i, V_C) = \left|\left| \phi(\boldsymbol{x}_i) - \frac{1}{|V_C|} \sum_{j \in V_C} \phi(\boldsymbol{x}_j) \right|\right|^{\,2}$$

$$D_{AV}(i, V_C) = < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_i) > - \frac{2}{|V_C|} \sum_{j \in V_C} < \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) > \; + \frac{1}{|V_C|^2} \sum_{k \in V_C} \sum_{j \in V_C} < \phi(\boldsymbol{x}_k), \phi(\boldsymbol{x}_j) >$$

$$Sim_{AV}(i, V_C) = -K(\boldsymbol{x}_i, \boldsymbol{x}_i) + \frac{2}{|V_C|} \sum_{j \in V_C} K(\boldsymbol{x}_i, \boldsymbol{x}_j \; - \frac{1}{|V_C|^2} \sum_{k \in V_C} \sum_{j \in V_C} K(\boldsymbol{x}_k, \boldsymbol{x}_j)$$

$$S_{AV}(i, V_C) = -K(\boldsymbol{x}_i, \boldsymbol{x}_i) + \frac{2}{|V_C|} \sum_{j \in V_C} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Score functions are used to rank nodes in a undirected graph

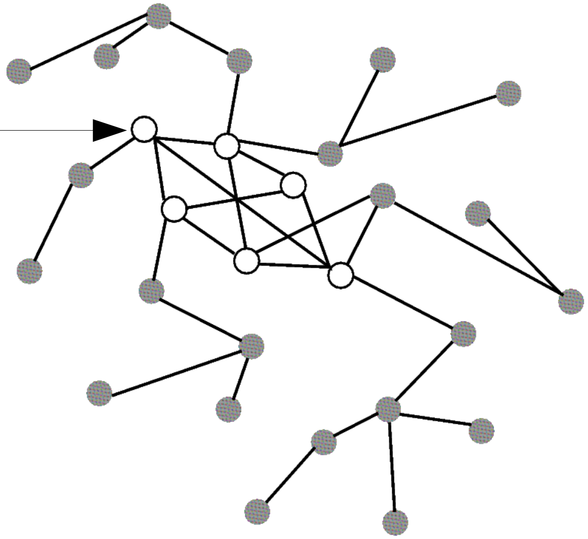
A modular approach:

1. Select a distance - score function

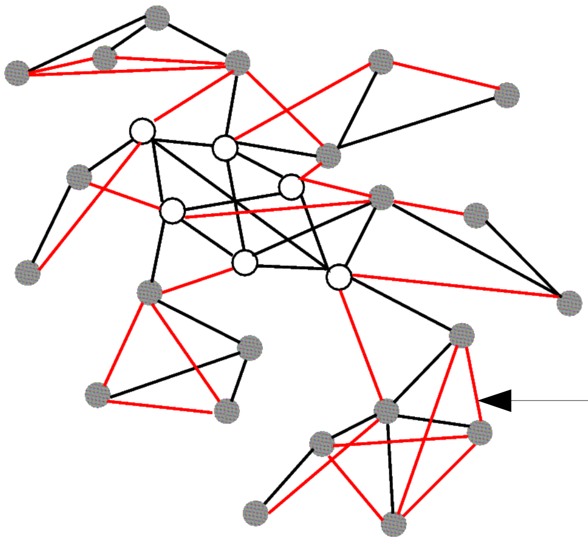
2. Select a suitable kernel

Kernelized score functions: a picture of the ranking method

A positive node

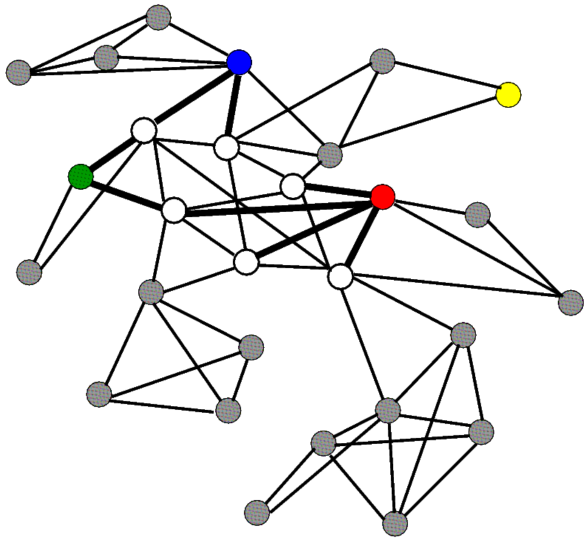


Original network



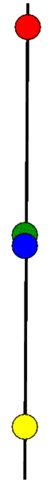
Augmented connectivity

Random walk kernel



Scoring of unlabeled nodes

high rank



connected with 4 positives

connected with 2 positives

. . .

low rank

Kernelized score functions : a drug repositioning case study

M. Re, and G. Valentini, Network-based Drug Ranking and Repositioning with respect to DrugBank Therapeutic Categories, IEEE ACM Transactions on Computational Biology and Bioinformatics 10(6), pp. 1359-1371, Nov-Dec 2013

- A network G=(V,E) connecting a large set of drugs: $\begin{cases} \text{Nodes} \rightarrow \text{drugs} \\ \text{Edges} \rightarrow \text{similarities} \end{cases}$
- A subset $V_C \subset V$ of drugs belonging to a given therapeutic category *C*

Rank drugs $v \in V$ w.r.t. to a given therapeutic category *C*

Many strategies for drugs networks construction: pairwise chemical similarity, bipartite network projection (projection in drug space of drug-target networks : drugs connected if they target the same protein/s).

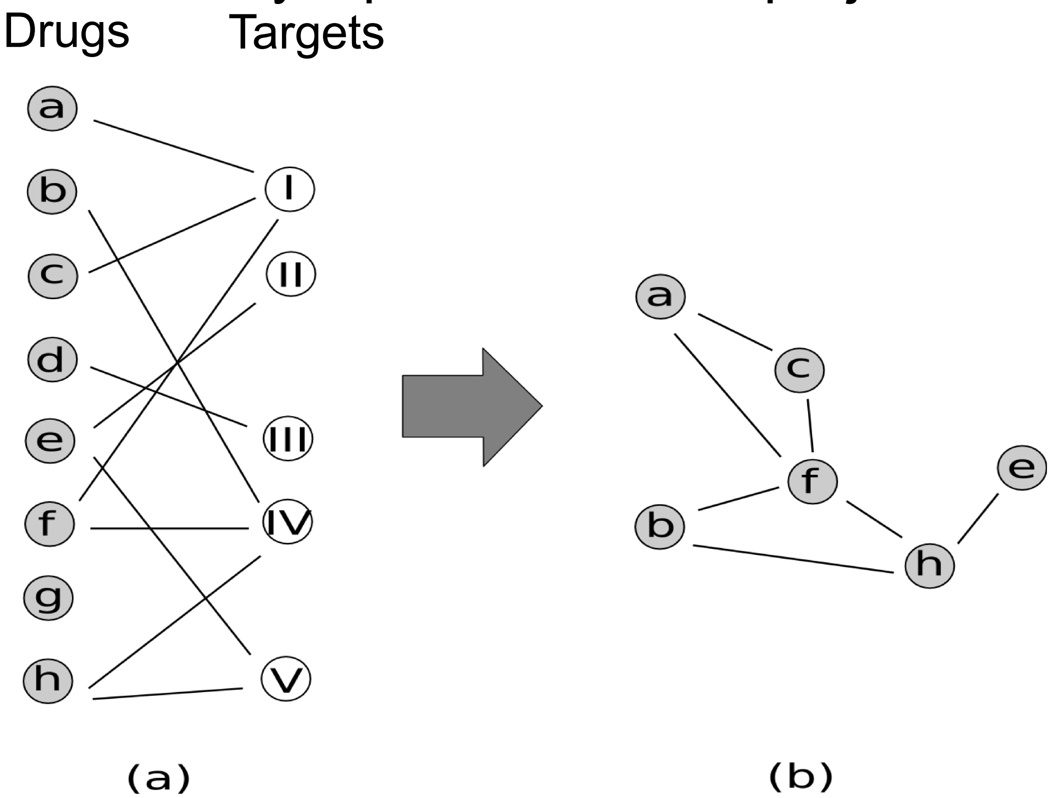
Kernelized score functions: experiments

- 1253 FDA approved drugs
- 51 DrugBank therapeutic classes
- 3 pharmacological networks:
  - $N_{structSim}$: pairwise chemical similarity (*Tanimoto* coefficients)
  - $N_{drugTarget}$: projection from drug-target interactions (from *DrugBank 3.0*)
  - $N_{drugChem}$: projection from chemical interactions (from *STITCH 2.0*)

Problem: inhomogeneous coverage in the 3 networks. Solution: networks integration.

Kernelized score functions

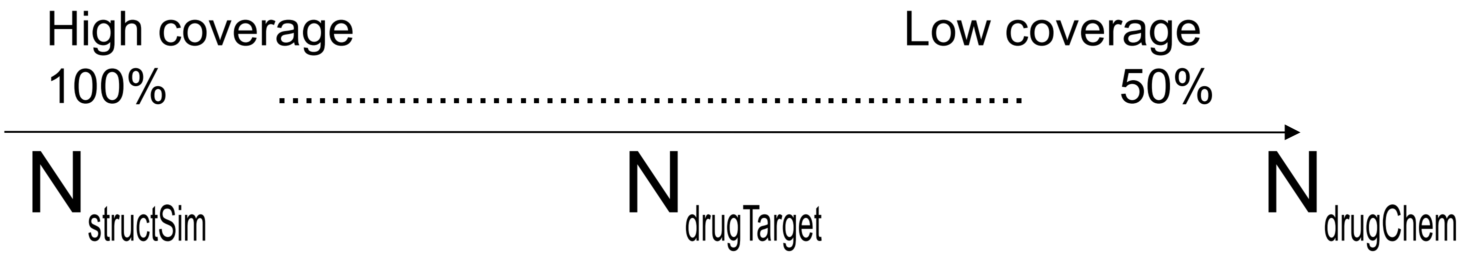
Network construction by bipartite network projection



Bipartite network (e.g. drug-target, drug-drug interaction)

One-mode drug network

Kernelized score functions: experiments



High coverage

100% 50%

Low coverage

$N_{\text{structSim}}$ $N_{\text{drugTarget}}$ N_{drugChem}

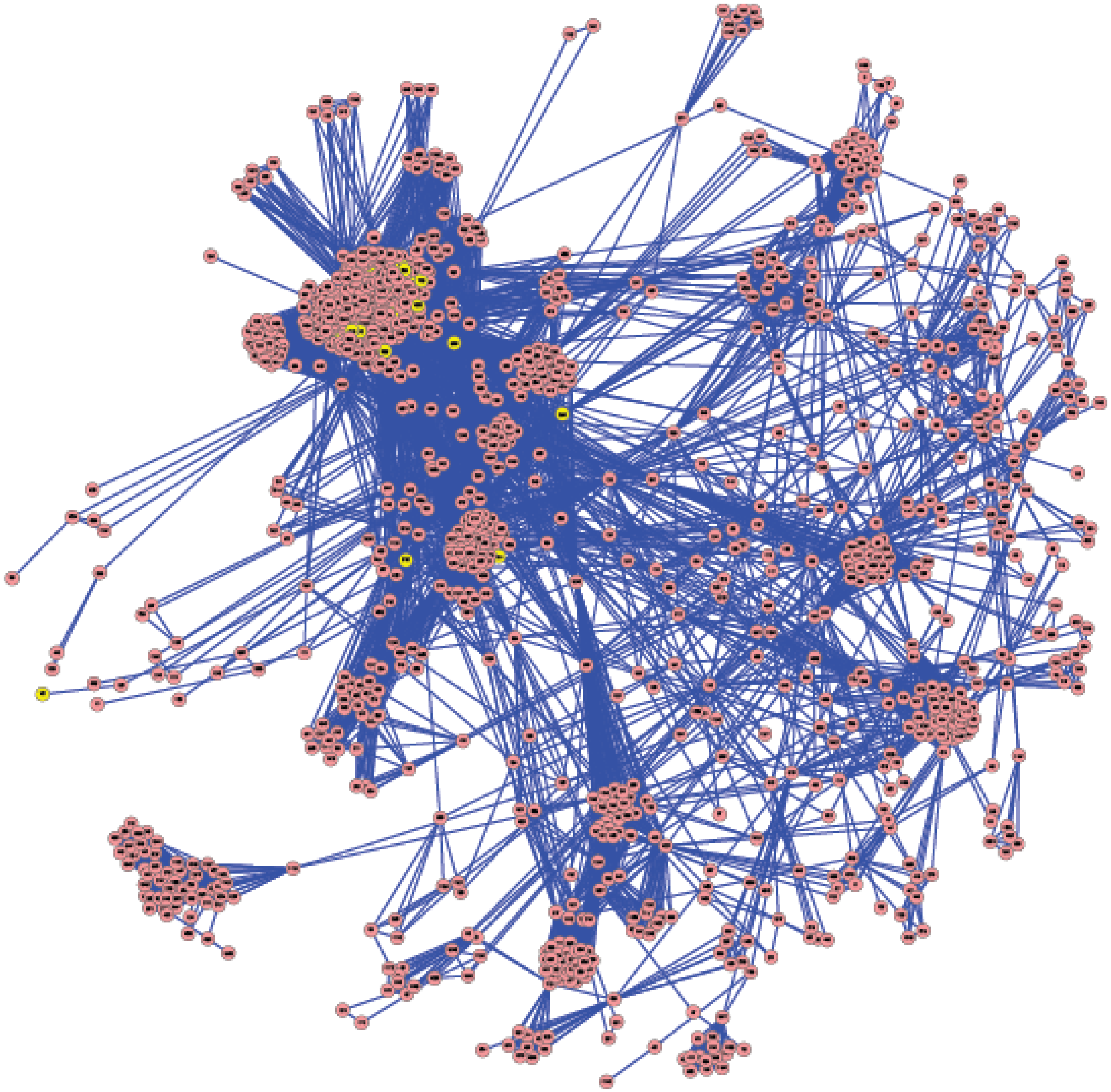
$$N_{\text{structSim}} \longrightarrow W_1$$ (1253 nodes, **13010** edges)

$$N_{\text{structSim}} + N_{\text{drugTarget}} \longrightarrow W_2$$ (1253, **43827**)

$$N_{\text{structSim}} + N_{\text{drugTarget}} + N_{\text{drugChem}} \longrightarrow W_3$$ (1253, **96711**)

NB: networks integration **increase the connectivity**!

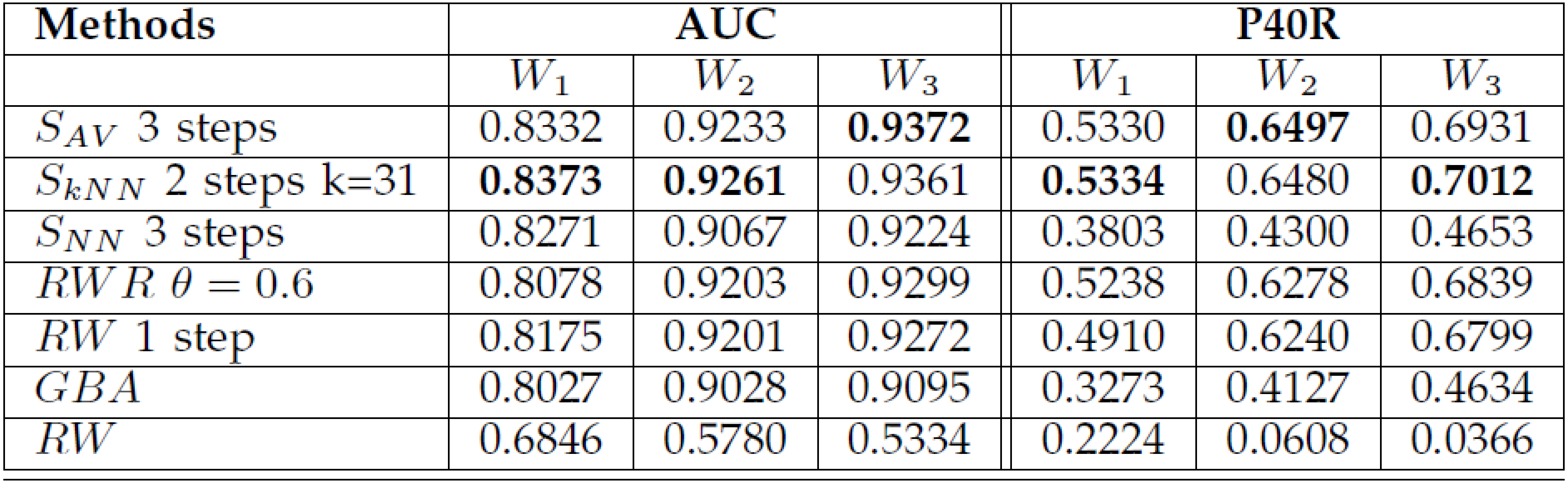
A view of the integrated pharmacological network



Kernelized score functions: results (AUC)

Kernelized score functions with random walk kernels **compared with *Random Walk* (RW) and *Random Walk with Restart* (RWR) algorithms**:

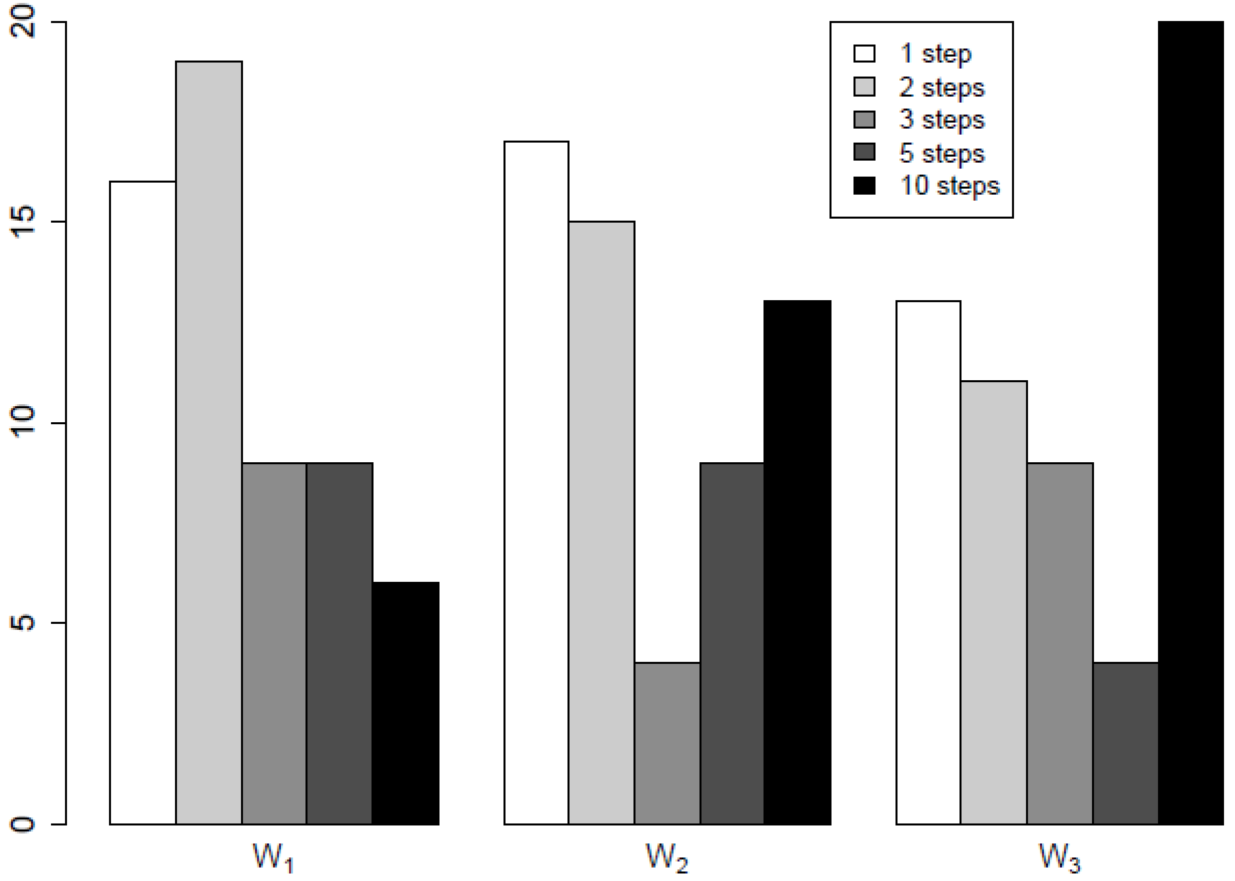
- 5-fold CV
- Results averaged across 51 DrugBank therapeutic classes having more than 15 drugs:



Methods	AUC			P40R		
	W_1	W_2	W_3	W_1	W_2	W_3
S_{AV} 3 steps	0.8332	0.9233	**0.9372**	0.5330	**0.6497**	0.6931
S_{kNN} 2 steps k=31	**0.8373**	**0.9261**	0.9361	**0.5334**	0.6480	**0.7012**
S_{NN} 3 steps	0.8271	0.9067	0.9224	0.3803	0.4300	0.4653
$RWR\ \theta = 0.6$	0.8078	0.9203	0.9299	0.5238	0.6278	0.6839
RW 1 step	0.8175	0.9201	0.9272	0.4910	0.6240	0.6799
GBA	0.8027	0.9028	0.9095	0.3273	0.4127	0.4634
RW	0.6846	0.5780	0.5334	0.2224	0.0608	0.0366

- $W_1 \rightarrow W_2 \rightarrow W_3$: AUC increments are statistically significant (Wilcoxon rank sum test, α=0.01)
- S_{AV} and S_{kNN} significantly better than the other methods (Wilcoxon rank sum test, α=0.01)

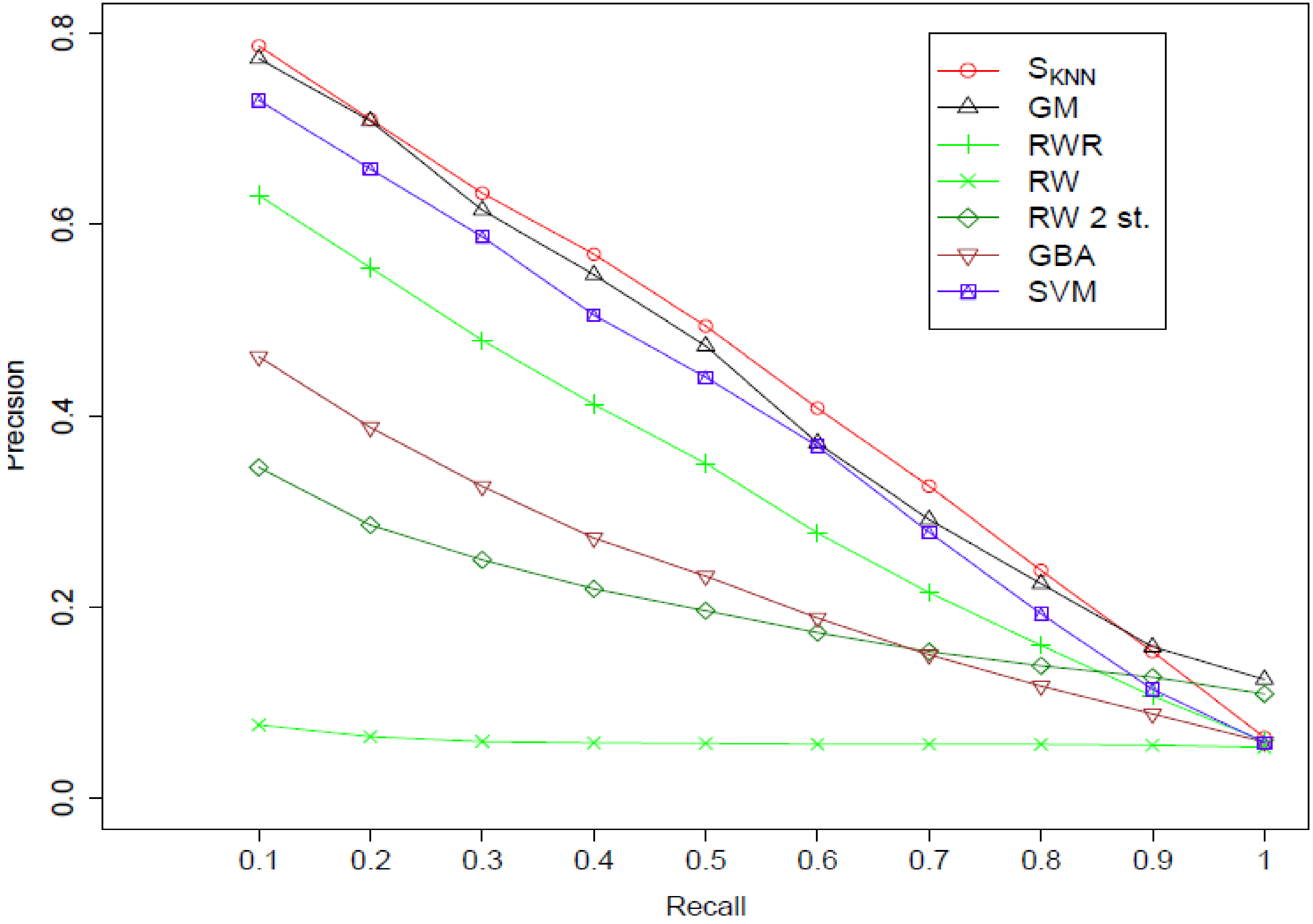
Kernelized score functions: Exploring deeply the integrated pharmacological space yields better results



Counts of the ”wins” across the 1254 therapeutic classes for the average score with 1, 2, 3, 5 and 10 steps random walk kernels

Kern. score functions : a gene function prediction case study

M. Re, M. Mesiti, and G. Valentini, “A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks,” IEEE ACM Transactions on Computational Biology and Bioinformatics, vol. 9, no. 6, pp. 1812–1818, 2012.



Kern. score functions : a gene disease prioritization case study

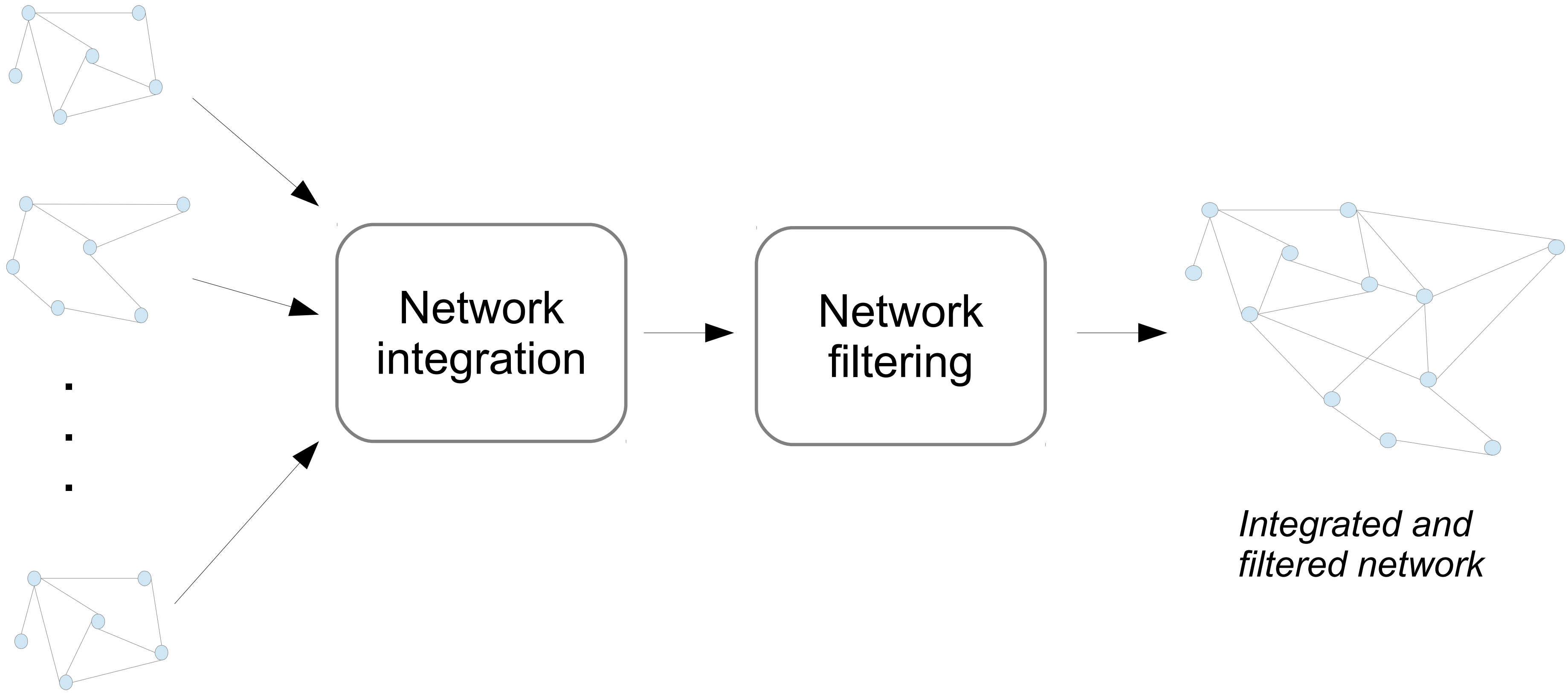
G. Valentini, A. Paccanaro, H. Caniza, A. Romero, M. Re, An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods, Artificial Intelligence in Medicine 61 (2) (2014)

Goals:

- An extensive analysis of gene-disease associations, considering a large set of diseases (708 MeSH diseases)
- Finding novel gene-disease associations for unannotated genes
- Analysis of the impact of network integration on gene prioritization

Our contribution

Analysis of the impact of network integration on gene prioritization



Network integration

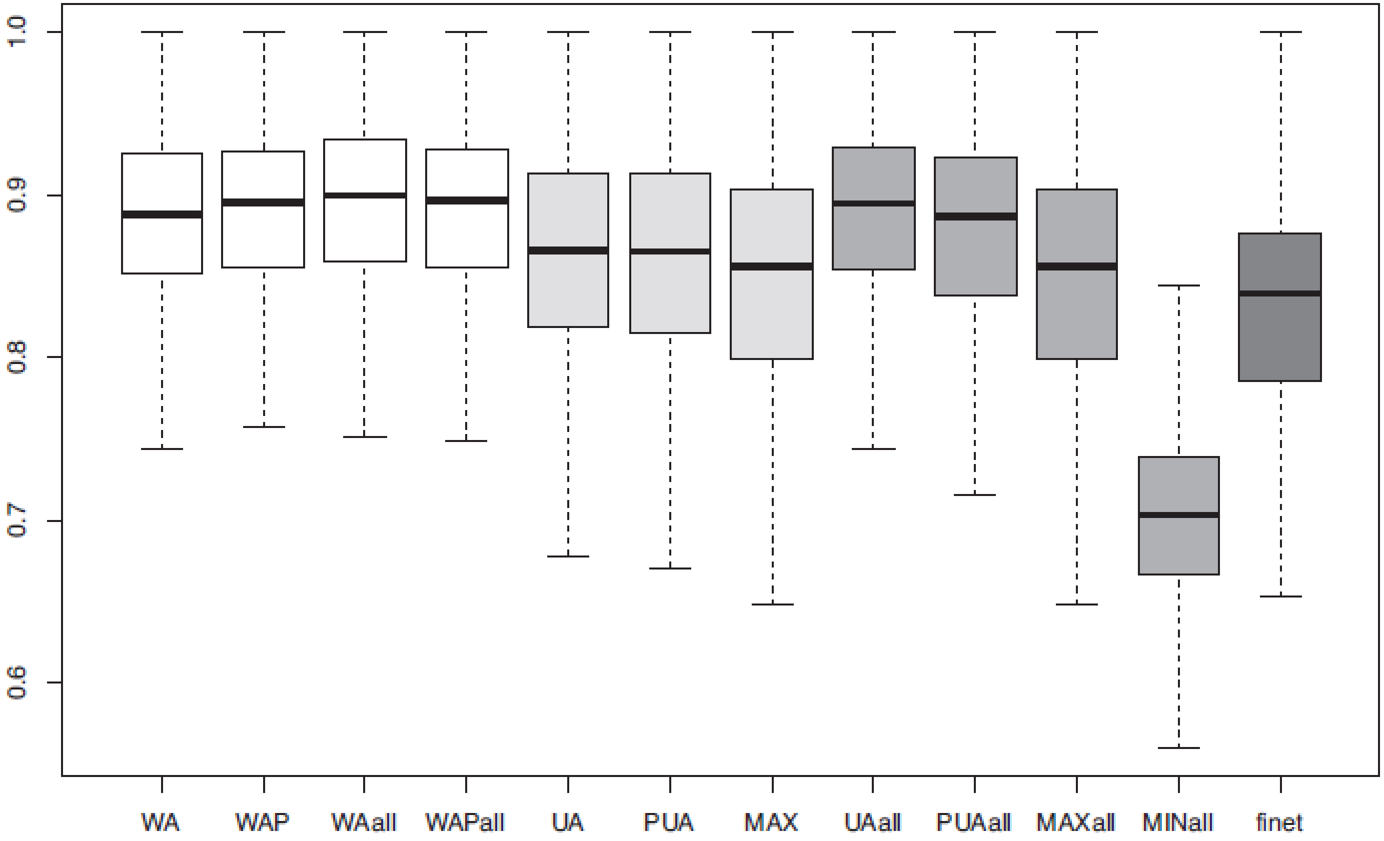
Network filtering

Integrated and filtered network

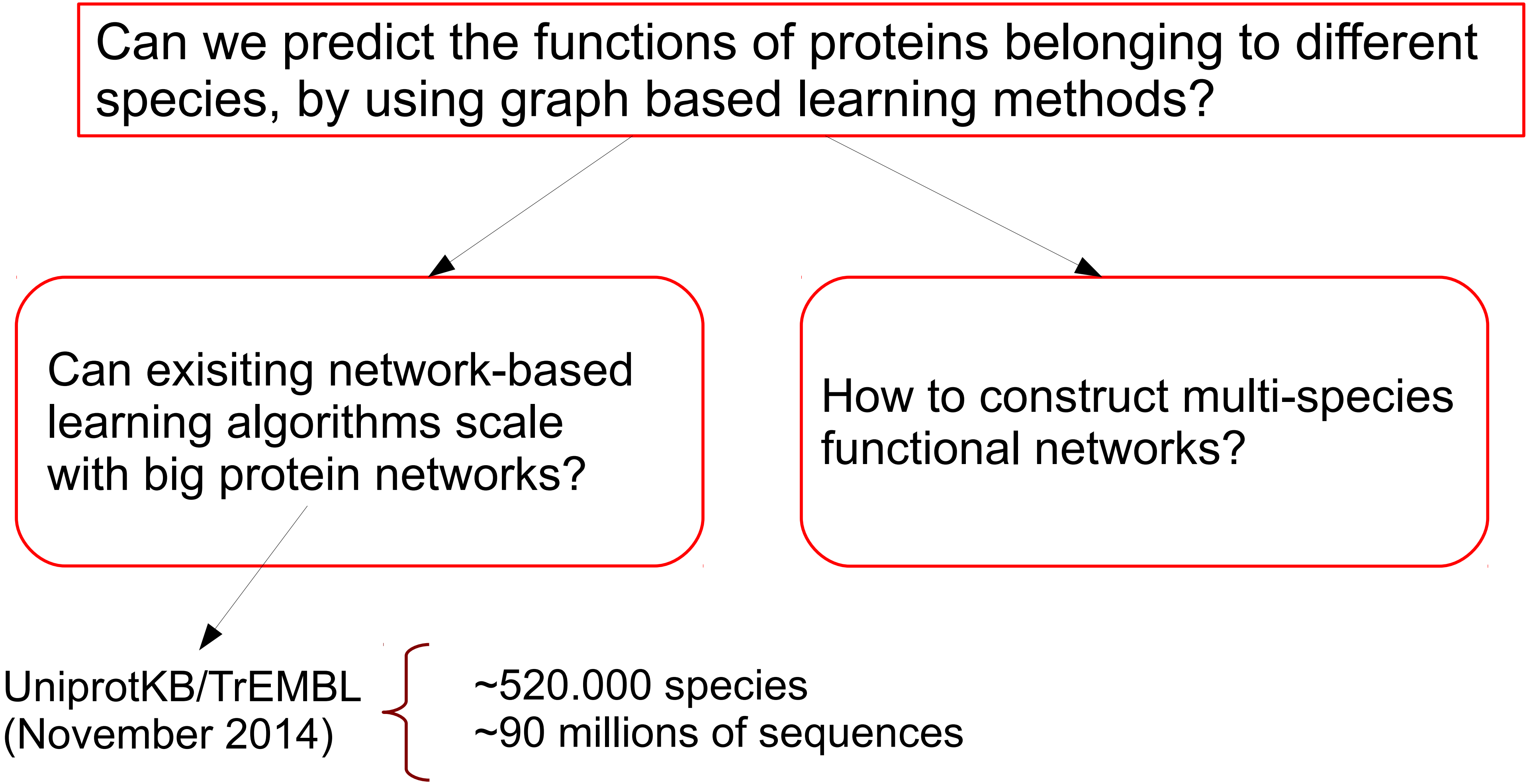
But also proper pre-processing and normalization of the networks is fundamental ...

Analysis of the impact of network integration on gene prioritization

Network	Description	Type	Nodes	Edges	Density
finet	Obtained from multiple sources of evidence	Binary	8449	271466	0.0038
hnnet	Obtained from multiple sources of evidence	Binary	8449	502222	0.0070
cmnet	Network projections from cancer modules	Binary	8449	3414722	0.0478
gcnet	Network projections from CTD	Binary	7649	1421298	0.0242
bgnet	Network projections from BioGRID	Binary	8449	120169	0.0016
dbnet	Direct relationships obtained from BioGRID	Binary	8449	3023084	0.0423
bpnet	Semantic similarity network from GO BP	Real valued	6923	44506147	0.9286
mfnet	Semantic similarity network from GO MF	Real valued	6145	26611887	0.7047
ccnet	Semantic similarity network from GO CC	Real valued	6693	39652637	0.8851



A relevant computational biology problem: Multi-species protein function prediction



Can we predict the functions of proteins belonging to different species, by using graph based learning methods?

Can exisiting network-based learning algorithms scale with big protein networks?

How to construct multi-species functional networks?

UniprotKB/TrEMBL (November 2014)

~520.000 species
~90 millions of sequences

Possible approaches to the scalability problem

1) Parallel distributed computation

- MapReduce framework (*Dean* and *Ghemawat*, 2004)
- Distributed graph parallel learning (*Gonzalez* et al. 2012)

Problems:

- Partitioning graphs across cluster nodes is hard (*Leskovec* et al 2009)
- Debugging and optimization is difficult
- Requires cluster / cloud systems

2) Secondary memory-based computation

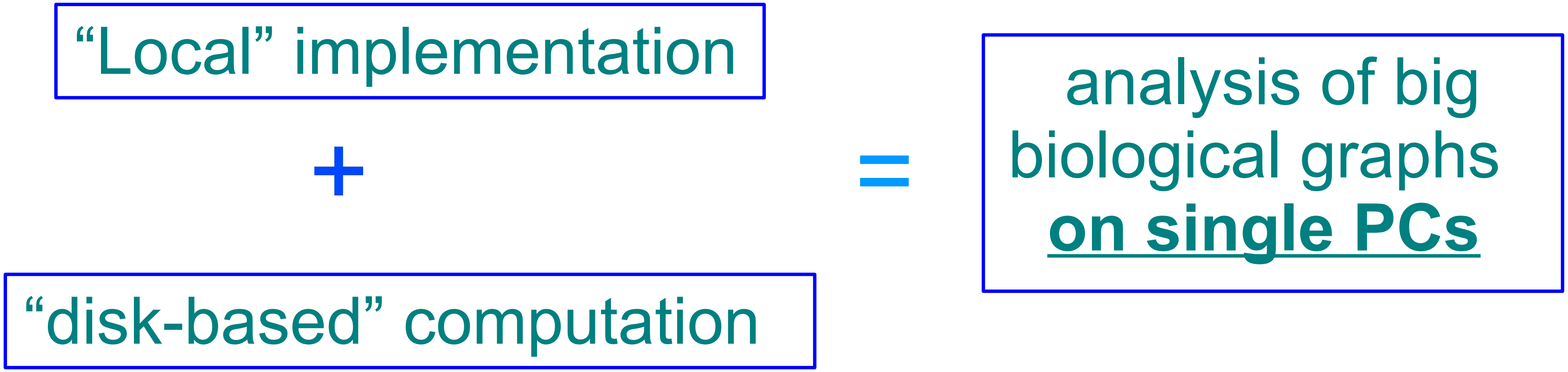
- Graph Database technologies (*Webber* et al. 2012)
- Secondary memory-based systems for the analysis of big graphs (*Kyrola* et al. 2014)

Problems:

- Design of **novel data structures** to store graphs on disks
- **Efficient I/O operations** and **graph processing** on disk

Our approach to big biological network analysis

M. Mesiti, M. Re, G. Valentini Think globally and solve locally: secondary memory-based network learning for automated multi-species function prediction, GigaScience, 3:5, 2014



"Local" implementation

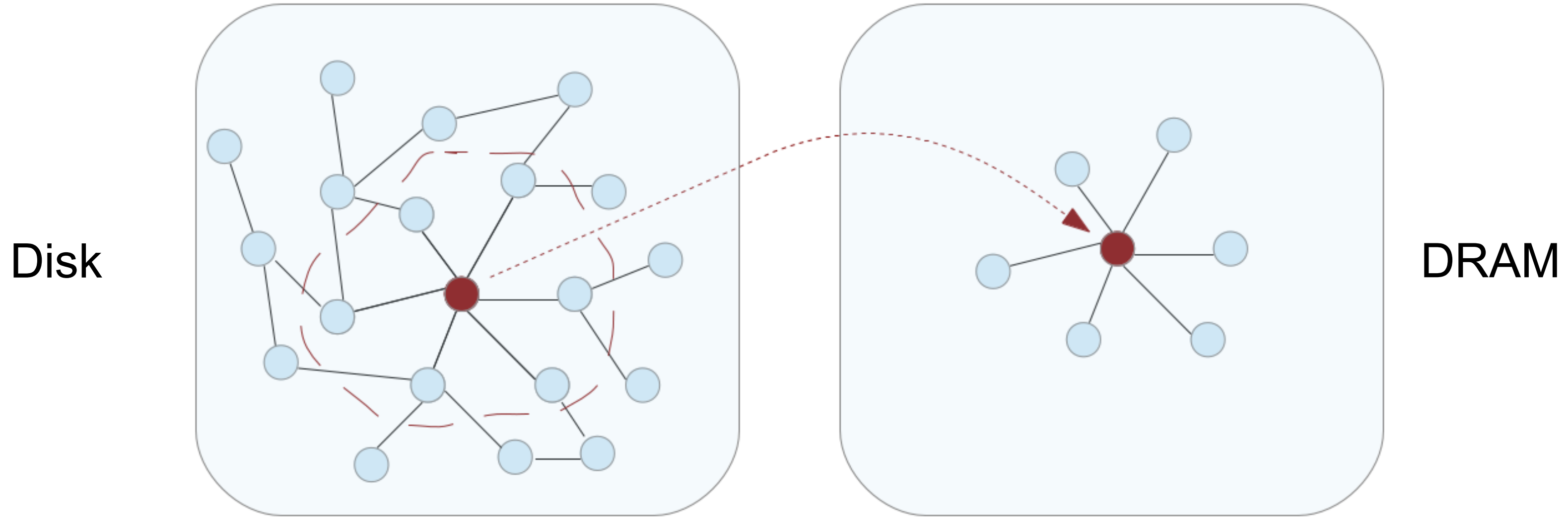
+

"disk-based" computation

=

analysis of big biological graphs **on single PCs**

"local" implementation of network-based algorithms



- We need DRAM to *store only the neighborhood of a single node*
- ***Vertex centric** computational model*:
translate "global" network-based methods to "local" implementation

The problem is: can we express a global GSSL algorithm as an iterative computation involving each time **only a single vertex and its neighborhood**?

An example: the classical random walk algorithm

Random walk: the classical algorithm in "global" version:

$\boldsymbol{W}$: weighted adjacency matrix of the graph

$\boldsymbol{D}$: diagonal matrix with $d_{ii} = \sum_j w_{ij}$ $\quad \boldsymbol{Q} = \boldsymbol{D}^{-1} \boldsymbol{W}$: the stochastic matrix

Probability update : $\boldsymbol{p}^{t+1} = \boldsymbol{Q}^T \boldsymbol{p}^t$

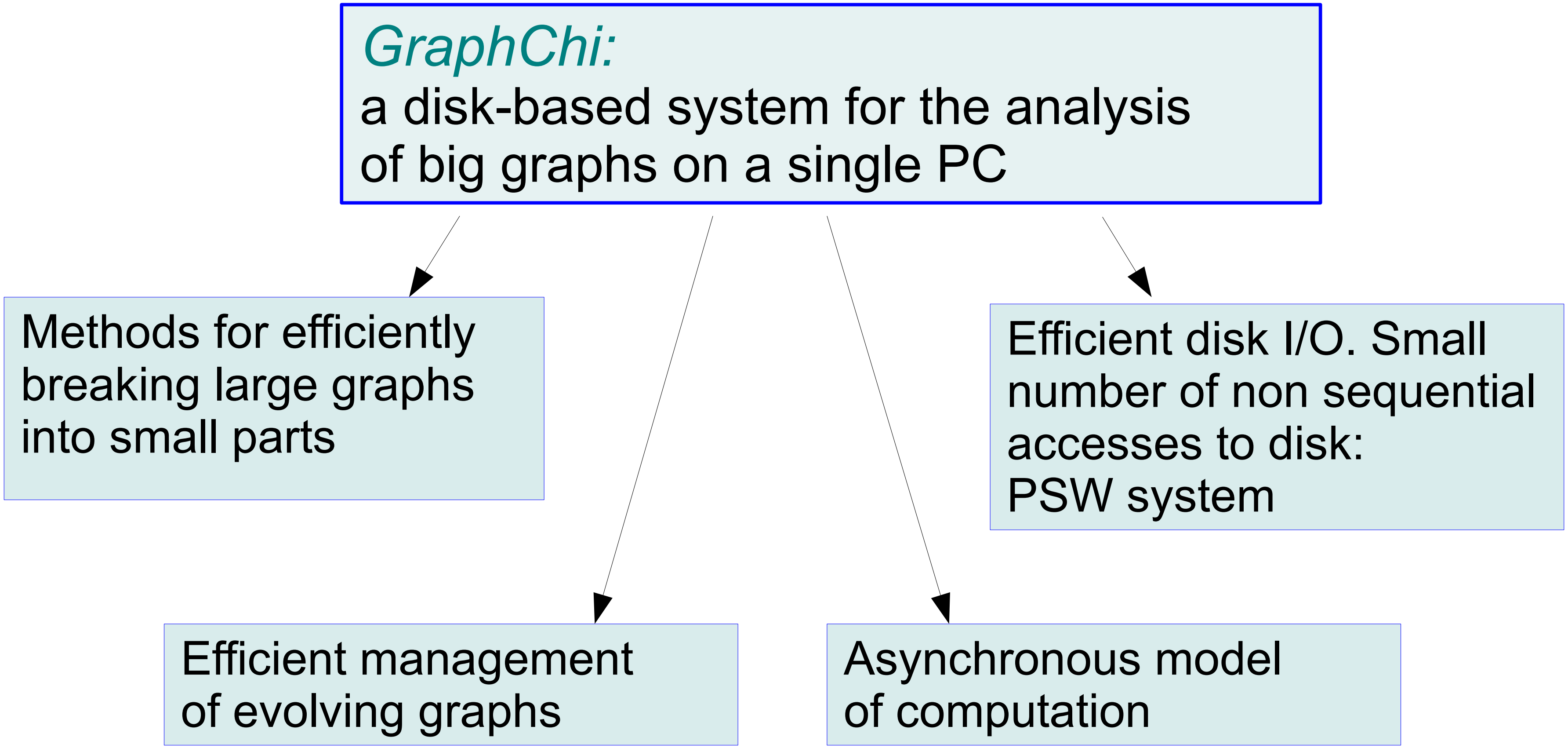
Random walk: the "local" vertex-centric implementation:

$$p_i^{t+1} = Q_i \, p^t = D^{-1} W_i \, p^t = \sum_j d_{jj} w_{ji} p_j^t$$

For each vertex i we need only its neighbours (at worst the i^th column of $\boldsymbol{W}$, the diagonal of $\boldsymbol{D}^{-1}$ and the probabilities computed at the previous iteration)

But we need fast disk access ...

GraphChi (Kyrola et al. 2012)



GraphChi:
a disk-based system for the analysis of big graphs on a single PC

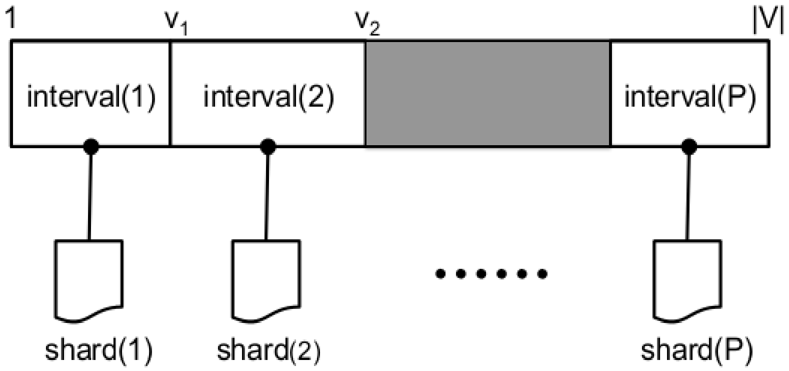
Methods for efficiently breaking large graphs into small parts

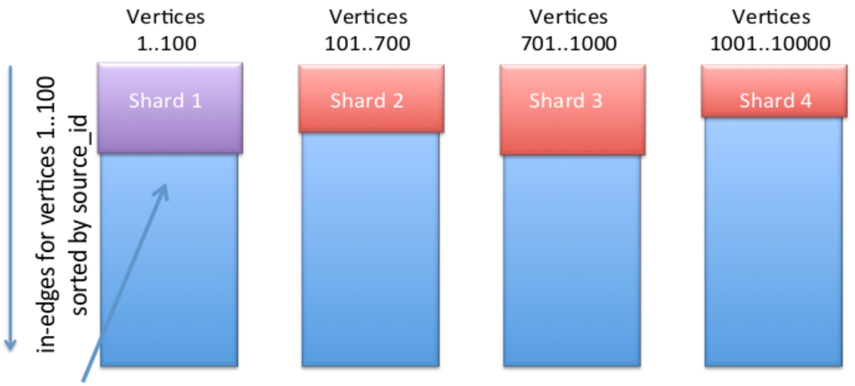
Efficient management of evolving graphs

Asynchronous model of computation

Efficient disk I/O. Small number of non sequential accesses to disk: PSW system

GraphChi: Parallel Sliding Windows (PSW)





Vertices split in P intervals. For each interval: in-edges stored in a shard, sorted by out-edges → To read each interval at most P non sequential reads (PSW method)

READ

Multi-thread asynchronous computation in main mem. → Parallel update of vertices and edges in the memory shards

EXEC

Blocks written back to disk → At most P^2 non sequential reads/writes on disk/full pass on the graph

WRITE

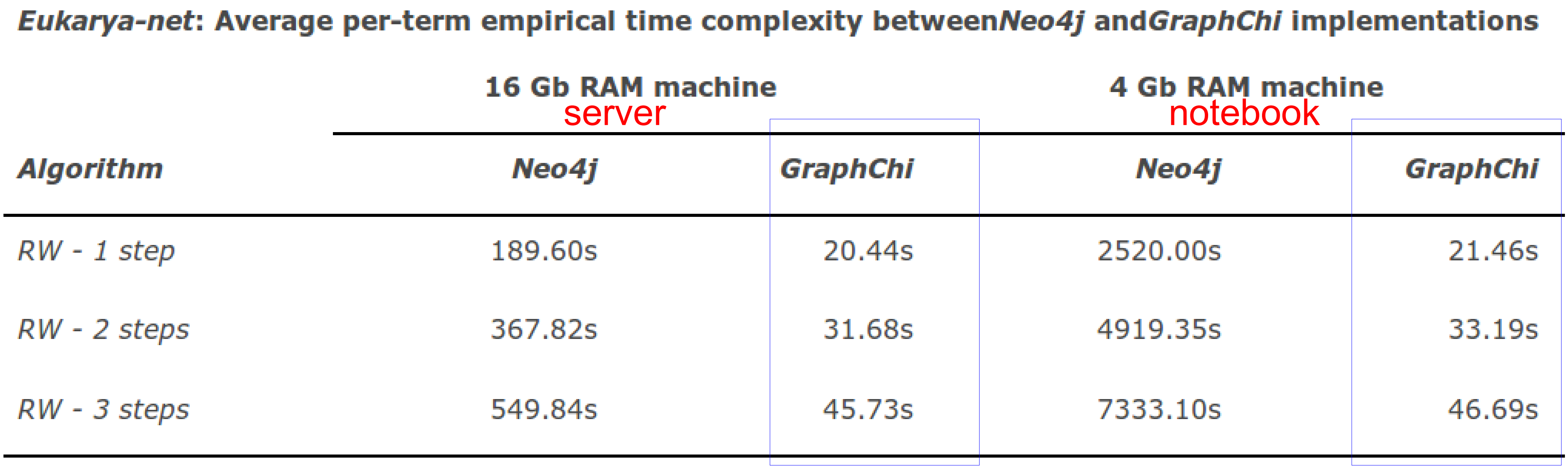
Experiments:

- **13** organisms
- **202,442** proteins
- **25,132,538** edges
- **50** classes

M. Mesiti, M. Re, G. Valentini Think globally and solve locally: secondary memory-based network learning for automated multi-species function prediction, GigaScience, 3:5, 2014

5 folds CV. Learning method: classical random walk. Implementations: GraphChi, Neo4j (a graph database)

Empirical time complexity :



***Eukarya-net*: Average per-term empirical time complexity between*Neo4j* and*GraphChi* implementations**

	16 Gb RAM machine server		4 Gb RAM machine notebook	
Algorithm	***Neo4j***	***GraphChi***	***Neo4j***	***GraphChi***
RW - 1 step	189.60s	20.44s	2520.00s	21.46s
RW - 2 steps	367.82s	31.68s	4919.35s	33.19s
RW - 3 steps	549.84s	45.73s	7333.10s	46.69s

Experiments: Comparison of multi-species and single species approaches

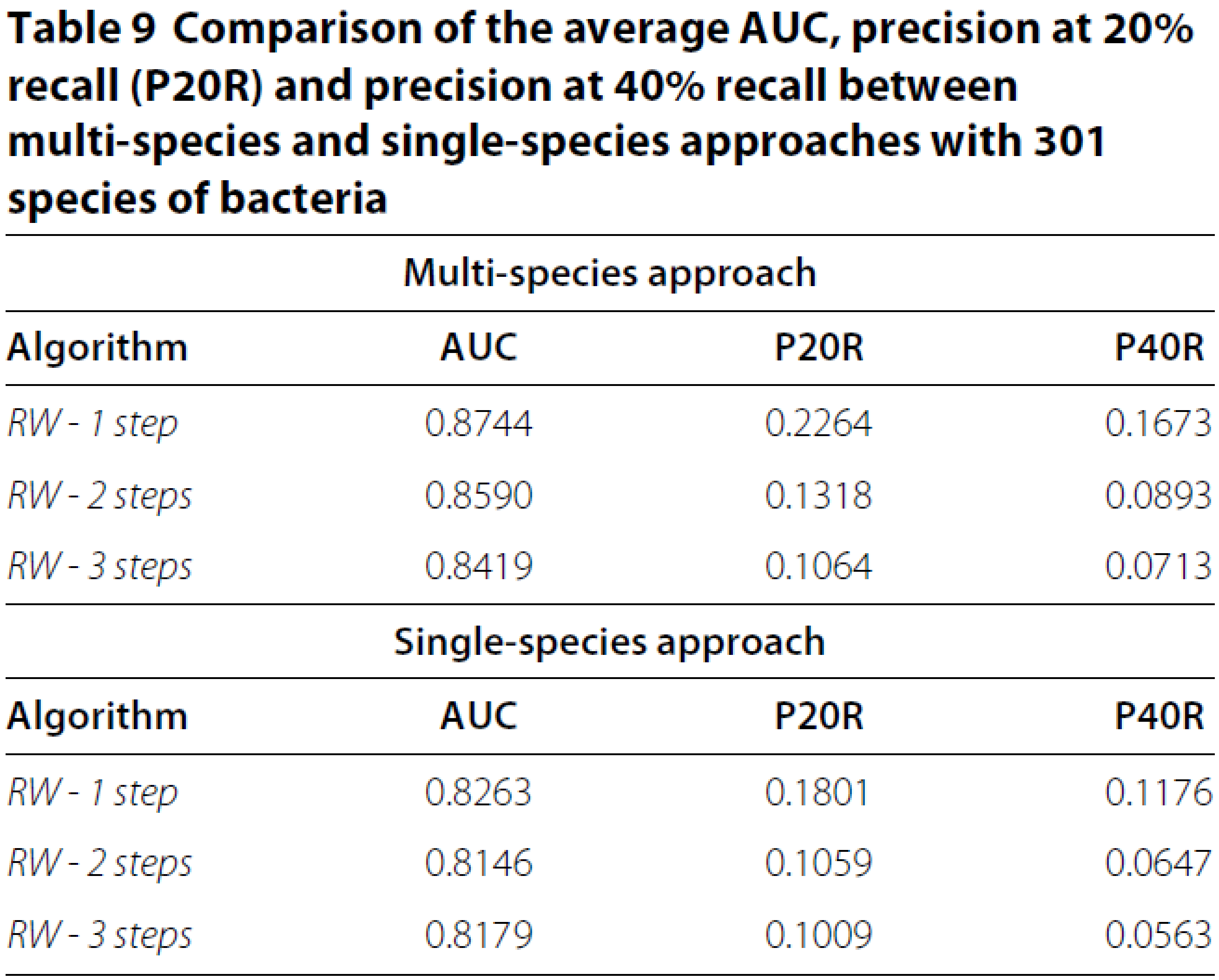


Table 9 Comparison of the average AUC, precision at 20% recall (P20R) and precision at 40% recall between multi-species and single-species approaches with 301 species of bacteria

Multi-species approach			
Algorithm	**AUC**	**P20R**	**P40R**
RW - 1 step	0.8744	0.2264	0.1673
RW - 2 steps	0.8590	0.1318	0.0893
RW - 3 steps	0.8419	0.1064	0.0713

Single-species approach			
Algorithm	**AUC**	**P20R**	**P40R**
RW - 1 step	0.8263	0.1801	0.1176
RW - 2 steps	0.8146	0.1059	0.0647
RW - 3 steps	0.8179	0.1009	0.0563

On going work on multi-species protein function prediction (MAFP) with kernelized score function

1. GraphChi vertex-centric implementation of the kernelized score functions

2. Construction of a big network including all the core proteins of the STRING database:
- more than 400 organisms
- 1.5 millions of proteins
- hundreds of millions of edges
- thousands of GO functional classes to be predicted

Main goals:

- Showing that MAFP can be exploited on off-the-shelf computers
- Showing that multi-species functional prediction significantly improves on single species functional prediction.

Conclusions:

- Semi-supervised graph-based methods are widely applied in several relevant problems in computational biology and medicine
- Kernelized score functions is a flexible algorithmic framework that can be applied in a broad range of interesting bioinformatics problems
- Kernelized score functions and the others state-of-the-art semi-supervised learning methods for biological network analysis are affected by serious scalability problems on big networks
- Local implementation of GSSL methods coupled with the usage of recent secondary memory technologies can make feasible GSSL tasks on very large (and dense) graphs, allowing novel biological insights from the analysis of bio-medical networks.

References:

- M. Mesiti, M. Re, G. Valentini Think globally and solve locally: secondary memory-based network learning for automated multi-species function prediction, *GigaScience*, 3:5, 2014
- G. Valentini, A. Paccanaro, H. Caniza, A. Romero, M. Re, An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods, *Artificial Intelligence in Medicine*, Volume 61, Issue 2, pages 63-78, June 2014
- M. Re, and G. Valentini, Network-based Drug Ranking and Repositioning with respect to DrugBank Therapeutic Categories, *IEEE ACM Transactions on Computational Biology and Bioinformatics* 10(6), pp. 1359-1371, Nov-Dec 2013
- M. Frasca, A. Bertoni, M. Re, and G. Valentini, A neural network algorithm for semi-supervised node label learning from unbalanced data, *Neural Networks* 43, pp.84-98, July 2013
- M. Re, M. Mesiti and G. Valentini, A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks, *IEEE ACM Transactions on Computational Biology and Bioinformatics* 9(6) pp. 1812-1818, 2012

Thank you!

Thank you for your attention!

And thanks also from Anacleto !

http://anacletolab.di.unimi.it