Supplementary Information for *Protein complex prediction via cost-based clustering*: The Restricted Neighbourhood Search Clustering Algorithm

King, A. D.,[*] Department of Computer Science, University of Toronto, Toronto, M5S 3G4, Canada
Pržulj, N., Department of Computer Science, University of Toronto, Toronto, M5S 3G4, Canada
Jurisica, I., Ontario Cancer Institute, Division of Cancer Informatics, Toronto, M5G 2M9, Canada.

April 5, 2004

1 Introduction

A *clustering* of a network is a partitioning of the network's nodes into sets called *clusters*. In a good clustering, we want the clusters of nodes to be highly intra-connected, or *dense*. We also want few connections between nodes in separate clusters, i.e. the clusters are to be sparsely interconnected. In the case of protein-protein interaction networks, the nodes are proteins and two nodes are connected if there is an interaction between them.

The Restricted Neighbourhood Search Clustering Algorithm (RNSC) is a *local search algorithm* for network clustering (King, 2004). This means that it searches the *solution space* of all possible clusterings of a network for a clustering with low cost; every possible clustering has an associated cost that reflects the goodness of the clustering. RNSC uses two separate cost functions.

2 The Cost Functions

RNSC uses two cost functions to judge the goodness of clusterings. One is the *naive cost function*, which is simple to compute and has integer values. The other is the *scaled cost function*, which is more complicated and can have non-integer values, but considers more information about the clustering being assessed. The naive cost function is computationally undemanding, and is therefore used as a fast preprocessing tool, whereas we really want to minimize the scaled cost function.

Consider a node v in a network G, and a clustering $\mathcal{C}$ of the network. Let α_v be the number of *bad connections* incident with v. A bad connection incident with v is one that exists between v and a node in a different cluster from v, or one that does not exist between v and a node u in the same cluster as v. The naive cost function of $\mathcal{C}$ is then defined as

$$C_n(G, \mathcal{C}) = \frac{1}{2} \sum_{v \in V} \alpha_v, \tag{1}$$

where V is the set of nodes in G.

For a vertex v in G with a clustering $\mathcal{C}$, let β_v be the size of the following set: v itself, any node connected to v, and any node in the same cluster of v. This measure reflects the size of the area that v effects in the clustering. We define the scaled cost function of $\mathcal{C}$ as

$$C_s(G, \mathcal{C}) = \frac{|V| - 1}{3} \sum_{v \in V} \frac{\alpha_v}{\beta_v}. \tag{2}$$

We can see that in both cost functions, what we want, ideally, is a clustering in which the nodes in a cluster are all connected to one another and there are no other connections.

[*] To whom correspondence regarding RNSC should be addressed

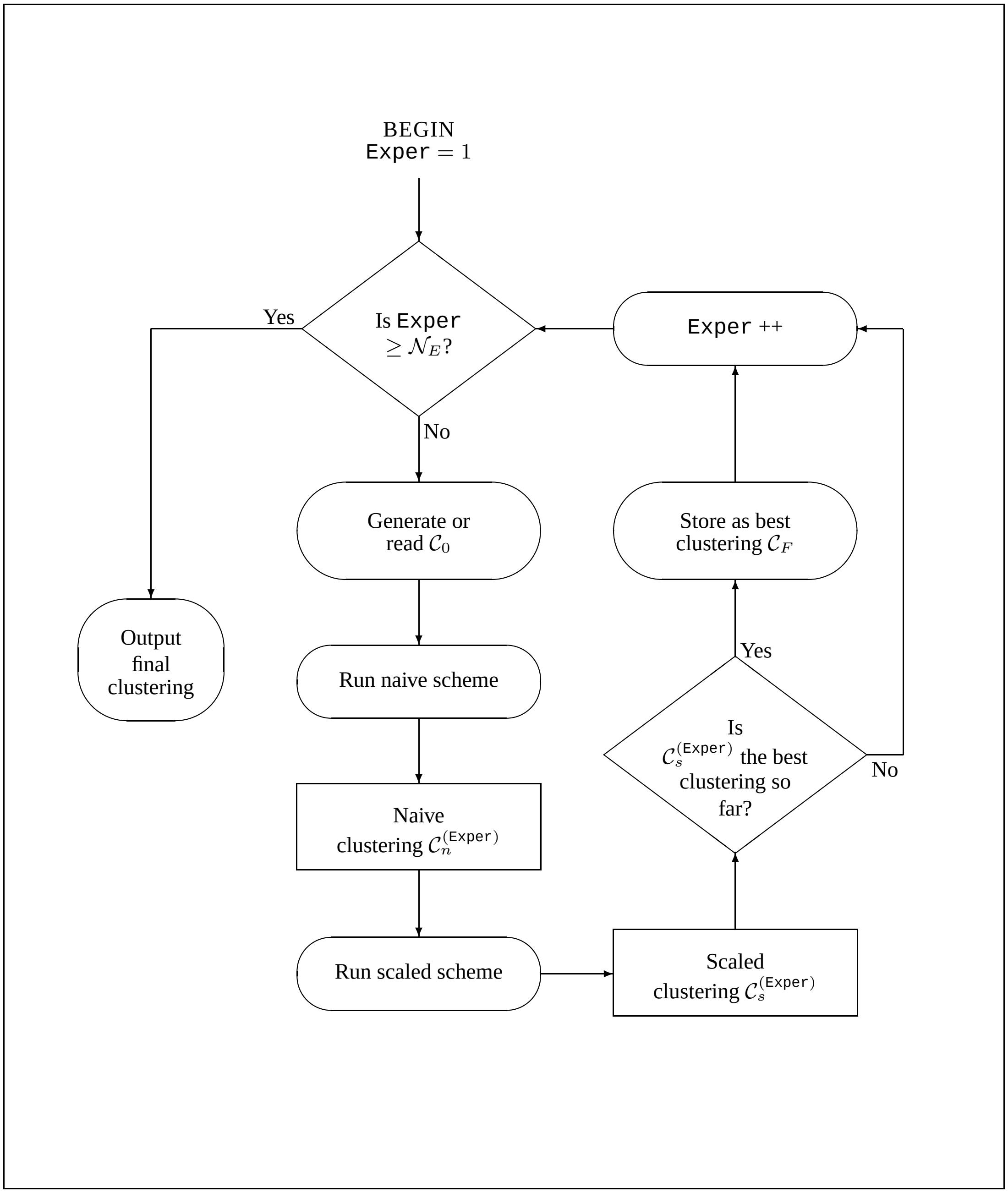
BEGIN
$\texttt{Exper} = 1$
Is $\texttt{Exper}$ $\geq \mathcal{N}_E$?
Yes
No
$\texttt{Exper}$ ++
Generate or read $\mathcal{C}_0$
Store as best clustering $\mathcal{C}_F$
Output final clustering
Run naive scheme
Yes
Is $\mathcal{C}_s^{(\texttt{Exper})}$ the best clustering so far?
No
Naive clustering $\mathcal{C}_n^{(\texttt{Exper})}$
Run scaled scheme
Scaled clustering $\mathcal{C}_s^{(\texttt{Exper})}$

Figure 1: The RNSC algorithm

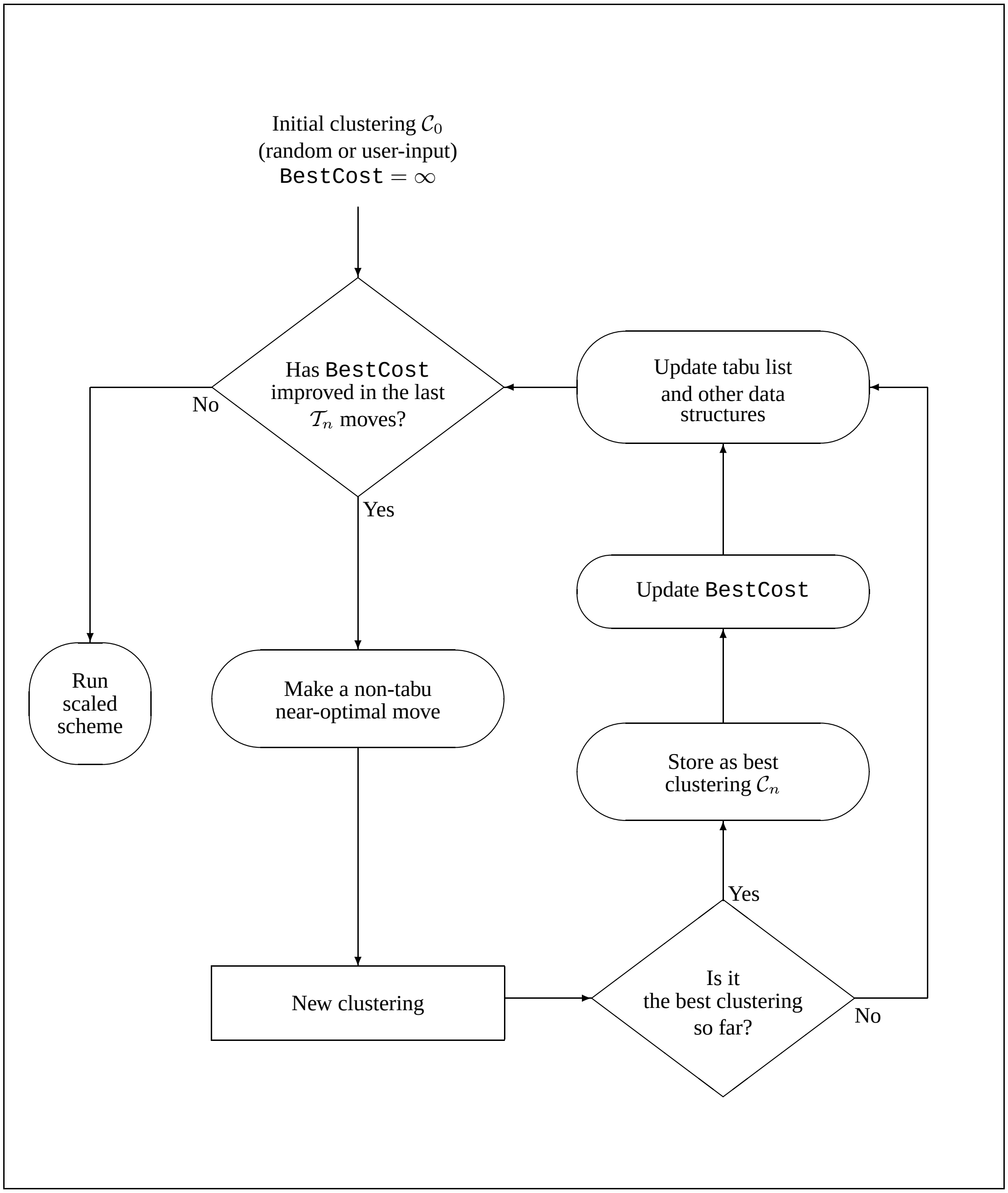
Initial clustering $\mathcal{C}_0$
(random or user-input)
$\texttt{BestCost} = \infty$
Has $\texttt{BestCost}$ improved in the last $\mathcal{T}_n$ moves?
No
Yes
Run scaled scheme
Make a non-tabu near-optimal move
New clustering
Is it the best clustering so far?
Yes
No
Store as best clustering $\mathcal{C}_n$
Update $\texttt{BestCost}$
Update tabu list and other data structures

Figure 2: The RNSC naive cost scheme

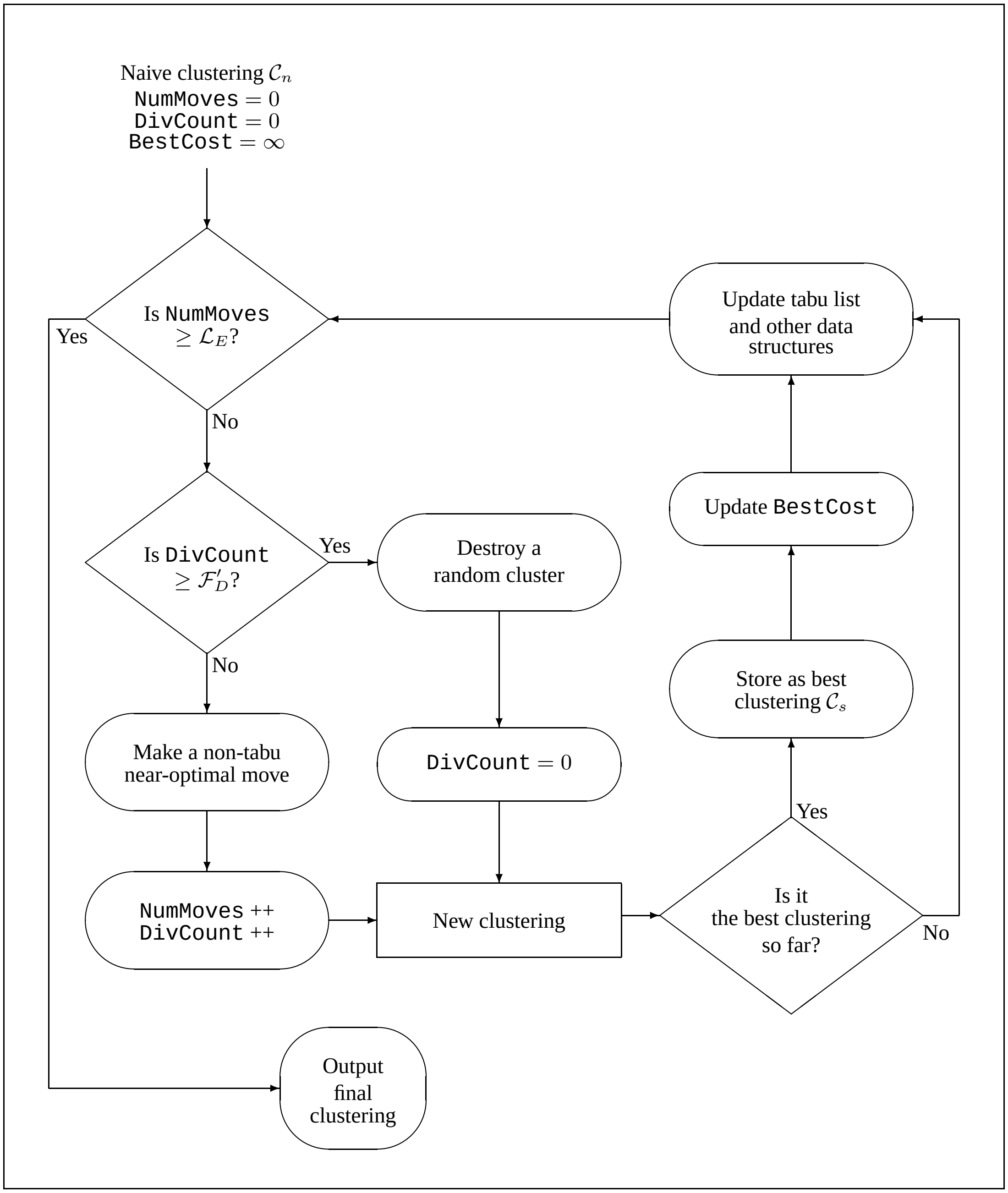
Naive clustering $\mathcal{C}_n$
$\texttt{NumMoves} = 0$
$\texttt{DivCount} = 0$
$\texttt{BestCost} = \infty$
Is NumMoves $\geq \mathcal{L}_E$?
Yes
No
Update tabu list and other data structures
Update BestCost
Is DivCount $\geq \mathcal{F}'_D$?
Yes
Destroy a random cluster
No
Store as best clustering $\mathcal{C}_s$
Make a non-tabu near-optimal move
$\texttt{DivCount} = 0$
Yes
NumMoves ++
DivCount ++
New clustering
Is it the best clustering so far?
No
Output final clustering

Figure 3: The RNSC scaled cost scheme

3 The RNSC Algorithm

We want to find a good clustering of a network G. Since RNSC is a randomized algorithm, we do this by running the algorithm a certain number of times. Each run generates a clustering $\mathcal{C}_s$. We take the $\mathcal{C}_s$ with lowest scaled cost as our final output clustering $\mathcal{C}_F$.

A single run, or *experiment*, of the algorithm begins with a random clustering $\mathcal{C}_0$ and attempts to find a clustering $\mathcal{C}_s$ with low scaled cost. It does this by first finding a clustering $\mathcal{C}_n$ with low naive cost, which can be done quickly. This stage is called the *naive cost scheme*. Such a clustering $\mathcal{C}_n$ is generally a good approximation to $\mathcal{C}_s$. RNSC then improves $\mathcal{C}_n$ by running the *scaled cost scheme*, which searches for a clustering with low scaled cost, outputting the lowest-cost clustering it finds, $\mathcal{C}_s$.

Both the naive cost scheme and the scaled cost scheme involve gradually improving the clustering by successively making *moves*. A move involves moving a vertex from its present cluster to another cluster. Figure 1 shows a flowchart of the entire RNSC process. Figures 2 and 3 show flowcharts of the naive cost scheme and the scaled cost scheme, respectively. $\mathcal{N}_E$ is the number of experiments that we wish to perform. $\mathcal{T}_n$ is the *naive stopping tolerance*: In the naive cost scheme, we stop when the best naive cost has not been updated in $\mathcal{T}_n$ moves. $\mathcal{L}_E$ is the *scaled experiment length*: In the scaled cost scheme, we stop when a total of $\mathcal{L}_E$ moves have been made.

In the scaled cost scheme, RNSC performs *diversification*. Diversification is a common strategy in local search algorithms. It involves periodically making a set of random moves to avoid settling into a clustering that is locally optimal but globally poor. We have a *diversification period* of $\mathcal{F}'_D$ moves: Every $\mathcal{F}'_D$ moves, RNSC destroys a randomly selected cluster by moving each node in the cluster to a random cluster.

Another strategy that RNSC uses to avoid choosing a globally poor clustering is the use of a *tabu list*. A tabu list acts as memory, forbidding a set of moves based on the moves that were recently made in order to prevent cycling (Glover, 1989). In this case, the tabu list is a list of vertices that cannot be moved. The use of diversification and a tabu list greatly improve the performance of RNSC (King, 2004).

4 Computational Performance

RNSC uses a number of data structures in order to search the set of clusterings for a network quickly. However, making moves is still costly: making a move in the naive scheme carries a computational cost of $\mathcal{O}(|V|)$, and making a move in the scaled scheme carries a computational cost of $\mathcal{O}(|V|^2)$. It is certainly not the fastest existing clustering algorithm, but it is very effective in finding a clustering of low cost according to our cost functions.

On a Pentium 4 2.8GHz processor, RNSC took as little as 10 seconds per experiment for Y_{2k}, the yeast network containing 988 proteins and 2455 interactions. The most computation-intensive network was F_{20k}, the *D. melanogaster* network containing 6985 proteins and 20,007 interactions. For this network, RNSC took roughly 150 minutes per experiment. Results for all of the PPI networks analyzed are given in Table 1.

RNSC is discussed in great detail from a graph-theoretic and operational research perspective in King's Master's thesis (King, 2004).

References

Giot, L., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., Vijayadamodar, G., Pochart, P., Machineni, H., Welsh, M., Kong, Y., Zerhusen, B., Malcolm, R., Varrone, Z., Collis, A., Minto, M., Burgess, S., McDaniel, L., Stimpson, E., Spriggs, F., Williams, J., Neurath, K., Ioime, N., Agee, M., Voss, E., Furtak, K., Renzulli, R., Aanensen, N., Carrolla, S., Bickelhaupt, E., Lazovatsky, Y., DaSilva, A., Zhong, J., Stanyon, C. A., Finley, R. L., Jr., White, K. P., Braverman, M., Jarvie, T., Gold, S., Leach, M., Knight, J., Shimkets, R. A., McKenna, M. P., Chant, J. & Rothberg, J. M. (2003) A protein interaction map of Drosophila melanogaster. *Science,* **302** (5651), 1727–1736.

Glover, F. (1989) Tabu search, part I. *ORSA Journal on Computing,* **1** (3), 190–206. "ORSA" is called Informs today.

Description	Name	# of proteins	# of interactions	Source	Time per experiment
Yeast 1	Y_{2k}	988	2455	(von Mering *et al.*, 2002)	10 sec.
Yeast 2	Y_{11k}	2401	11,000	(von Mering *et al.*, 2002)	3 min.
Yeast 3	Y_{45k}	4687	45,000	(von Mering *et al.*, 2002)	48 min.
Yeast 4	Y_{78k}	5321	78,390	(von Mering *et al.*, 2002)	65 min.
Fly 1	F_{5k}	4602	4637	(Giot *et al.*, 2003)	54 min.
Fly 2	F_{20k}	6985	20,007	(Giot *et al.*, 2003)	180 min.
Worm 1	W_{5k}	3115	5222	(Li *et al.*, 2004)	10 min.

Table 1: Protein-protein interaction networks clustered by RNSC

King, A. D. (2004). Graph clustering with restricted neighbourhood search. Master's thesis University of Toronto.

Li, S., Armstrong, C. M., Bertin, N., Ge, H., Milstein, S., Boxem, M., Vidalain, P.-O., Han, J.-D. J., Chesneau, A., Hao, T., Goldberg, D. S., Li, N., Martinez, M., Rual, J.-F., Lamesch, P., Xu, L., Tewari, M., Wong, S. L., Zhang, L. V., Berriz, G. F., Jacotot, L., Vaglio, P., Reboul, J., Hirozane-Kishikawa, T., Li, Q., Gabel, H. W., Elewa, A., Baumgartner, B., Rose, D. J., Yu, H., Bosak, S., Sequerra, R., Fraser, A., Mango, S. E., Saxton, W. M., Strome, S., van den Heuvel, S., Piano, F., Vandenhaute, J., Sardet, C., Gerstein, M., Doucette-Stamm, L., Gunsalus, K. C., Harper, J. W., Cusick, M. E., Roth, F. P., Hill, D. E. & Vidal, M. (2004) A map of the interactome network of the metazoan C. elegans. *Science*, **303** (5657), 540–543.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S. & Bork, P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417** (6887), 399–403.