A New Physarum Learner for Network Structure Learning from Biomedical Data

T. Schön[1,2], M. Stetter[2], A. M. Tomé[3,1] and E. W. Lang[1]

[1]*CIML Group, Biophysics, University of Regensburg, Regensburg, Germany*

[2]*University of Applied Science Weihenstephan-Triesdorf, Freising, Germany*

[3]*IEETA, DEETI, University of Aveiro, Aveiro, Portugal*

elmar.lang@biologie.uni-regensburg.de

Abstract: A novel structure learning algorithm for Bayesian Networks based on a *Physarum Learner* is presented. The length of the connections within an initially fully connected Physarum-Maze is taken as the inverse Pearson correlation coefficient between the connected nodes. The Physarum Learner then estimates the shortest indirect paths between each pair of nodes. In each iteration, a score of the surviving edges is incremented. Finally, the highest scored connections are combined to form a *Bayesian Network*. The novel *Physarum Learner* method is evaluated with different configurations and compared to the LAGD Hill Climber showing comparable performance with respect to quality of training results and increased time efficiency for large data sets.

1 INTRODUCTION

In 2000, Nakagaki et al. (Nakagaki et al., 2000) showed that the slime mould *Physarum polycephalum* can find the shortest path between two points in a maze. In the following years, research efforts focused on examining the detailed strategy used by Physarum to understand the adaptive dynamics of its transport network (Nakagaki et al., 2001; Nakagaki, 2001; Nakagaki et al., 2004). Tero et al. (Tero et al., 2006; Tero et al., 2007) proposed a physical model for the underlying transport network based on hydrodynamics and showed that their model (*Physarum Solver*) can solve the shortest path problem of the maze introduced by Nakagaki (Nakagaki et al., 2000) in a manner similar to the biological slime mould. A detailed mathematical analysis and studies of the convergence properties of the *Physarum Solver* are discussed in (Tomoyuki and Isamu, 2008; Ito et al., 2011; Brummitt et al., 2010) Apart form path finding problems, new applications for the *Physarum Solver* are now being investigated. In this paper, we present a novel approach where the strategy used by *Physarum Solver* is adapted to a well known NP-hard problem, namely learning *Bayesian Network* structure from data.

2 DATA SETS

Three real binary data sets are used for evaluations, Asia (8 nodes, 8 arcs, average in-degree of 2)(Lauritzen and Spiegelhalter, 1988), Cancer (Korb and Nicholson, 2010) and Earthquake (5 nodes, 4 arcs, average in-degree of 1.6, each) (Korb and Nicholson, 2010). Furthermore, a set of 15 binary networks has been created randomly with different characteristics using the Bayesian Network sampler implemented in WEKA. The networks are denoted as n*X*p*Y*a*Z* where *X* is replaced by the number of nodes, *Y* by the number of allowed parents and *Z* by the number of arcs. The network configurations can be seen in Table 1. For each network, WEKA was used to sample data sets randomly with 1000 instances each.

3 METHODS

Tero et al. (Tero et al., 2007) introduced a graphical model for the maze created by Nakagaki (Nakagaki et al., 2000).

To a graph where at each branch in the maze, a node N is inserted. If the maze contains a direct way from node N_i to N_j, a connection between the nodes is added. We refer to this kind of graph as *Physarum-*

Maze.

For a wide-sense stationary system, the mass flux Q_{ij} through the tube segment M_{ij} between two nodes N_i and N_j follows Fick's first law

$$Q_{ij} = -D_{ij}\nabla p_{ij} = \frac{D_{ij}}{L_{ij}}(p_j - p_i) = P_{ij}(p_j - p_i) \quad (1)$$

where D_{ij} is the conductivity, L_{ij} is the length of the edge M_{ij} and P_{ij} its permeability. Finally, the pressures at node N_i and N_j, representing free energy densities, are denoted by p_i and p_j. By considering mass conservation following Kirchhoff's first law, the Poisson equation for the pressures result. The flux through each edge is controlled by a time-dependent conductivity $D_{ij}(t)$ following a Master equation which models a balance between a positive feedback term, causing conductivity to increase with increasing flux, and a relaxation term which occasionally causes edges to vanish from the graph.

3.1 Building a *Physarum-Maze* from Sampling Data

A *Physarum-Maze* is generated by adding, for each attribute in the data set, a node to the network, and adding an undirected connection between each pair of nodes, so that the maze is fully connected. The conductivity D_{ij} of each connection M_{ij} is initialized randomly in the range of $D_{min} \leq D_{ij} \leq D_{max}$. Estimating the length L_{ij} of the connection between two nodes is the key task of learning a *Bayesian Network* structure with the *Physarum Solver*. Here we propose that the length of the connection represents the level of independence of two adjacent nodes in the network. The latter are taken to be nominal or ordinal, and the connections in the *Physarum Solver* are considered undirected. Node independence is approximated by a correlation coefficient ρ_{ij} between the connected nodes. Therefore, a symmetric correlation coefficient for nominal data is needed. As the main goal of this study is to examine whether it is generally possible to learn *Bayesian Network* structure with the *Physarum Solver*, for the sake of simplicity we restricted the benchmark data sets to be binary. This restriction allows to use the simple normalized correlation coefficient ϕ given by (Davenport and El-Sanhurry, 1991)

$$\phi = \frac{(BC - AD)}{\sqrt{(A+B)(C+D)(A+C)(B+D)}} \quad (2)$$

where the normalized value $-1 \leq \phi_{norm} = \frac{\phi}{\phi_{max}} \leq 1$ is calculated with

$$\phi_{max} = \begin{cases} \frac{\sqrt{(A+C)(C+D)}}{\sqrt{(B+D)(A+B)}}, & \text{if } B > C \\ \frac{\sqrt{(B+D)(A+B)}}{\sqrt{(A+C)(C+D)}}, & \text{else} \end{cases} \quad . \quad (3)$$

The coefficient ϕ uses a two-dimensional *Contingency* table of the binary variables X and Y, see Table 1.

Table 1: Contingency table.

X \ Y	y1	y0	Total
x1	A	B	A+B
x0	C	D	C+D
Total	A+C	B+D	N

The normalized correlation coefficient is interpreted in a similar way as the Pearson correlation coefficient. For each connection, the length L_{ij} between node N_i and N_j is given by

$$L_{ij} = (10(1 - |\phi_{norm,(i,j)}| + l))^{\gamma} \quad (4)$$

The control parameter γ assures growth of the length L_{ij} of the segment if $\gamma > 1.0$. The correlation coefficient ϕ is subtracted from 1 to yield short segment lengths between highly correlated nodes. A constant $l \leq 0.1$ is added to avoid connections with vanishing length $L_{ij} \to 0$ if $\phi \to 1.0$.

3.2 The *Physarum Learner*

Two additional nodes, denoted as source and sink node, are added to the *Physarum-Maze* which, initially, are not connected to any node in the maze. Furthermore, the Physarum connections M_{ij} are completed by a score s that is initialized with $s_{ij} = 0 \ \forall M_{ij} \in \mathcal{M}$. The *Physarum Learner* iterates over all possible connections between two nodes in the maze. For each pair of nodes $N_i \leftrightarrow N_j$, the direct connection M_{ij} between N_i and N_j is removed. The source node N_{+1} is connected to N_i while N_j is connected to the sink node N_{-1} with length $L_{+1,i} = L_{j,-1} = L_0$. The value of the constant L_0 can be chosen arbitrarily, because there is only one connection to the source and one connection to the sink node which, therefore, survive in any case, independent of their length. Then the *Physarum Solver* (Tero et al., 2008) is applied to the maze. The score of the connections which remained in the network after one epoch of the *Physarum Solver* is increased by 1. Next the previously removed connection between N_i and N_j is reinserted. The source and sink nodes are disconnected from N_i and N_j and re-assigned to other, randomly

chosen nodes. Finally, the conductivities D_{ij} are reset before the next epoch is started. After all iterations have finished, each node has been connected to the source and to the sink node exactly $N-1$ times. The result of the *Physarum Learner* is an undirected and fully connected graph, where each connection has a score $0 \le s_{ij} \le \frac{n(n-1)}{2}$. A score $s_{ij} = 0$ indicates that the connection M_{ij} has never survived in any of the *Physarum Learner* epochs. As the length of connections are indirectly proportional to the correlation between the connected nodes, a connection between highly correlated nodes is expected to survive more often than a connection between uncorrelated nodes. In other words, the *Physarum Learner* forces the *Physarum Solver* to find indirect paths that explain the correlation between two nodes, by blocking the direct path between these two nodes.

3.3 Generate *Bayesian Network* from *Physarum-Maze*

The basic problem on transforming the scored graph into a *Bayesian Network* is that the connections established by the *Physarum Learner* are undirected whereas a *Bayesian Network* requires directed edges. Hence, learning a *Bayesian Network* from data by the *Physarum Learner* is only possible if a correct ordering of the nodes is known. Otherwise, the *Physarum Learner* generates an undirected graphical model of the data set, also called Markov network. Transforming a Markov network into a *Bayesian Network* can be done by *triangulation* which, however, often results in a loss of independence information (Koller and Friedman, 2009).

The graphical network is built from the *Physarum Learner* output in the following way: A list of connections is created and sorted by their score so that the connection with the highest score is the first entry. An empty graphical model is initialized. The connections are added to the graph in the sequence of the list. The highest scored connection is added first, followed by the second highest scored connection and so on until the score falls below a given threshold θ. In this work, θ is set to the arithmetic average of the scores of all connections whenever *score* $\ne 0$ for each simulated network. Unconnected nodes, i. e. nodes whose score falls below the threshold, are added afterwards according to their highest scored connection.

In case where the ordering is known and a *Bayesian Network* is to be learned, the connections are already directed from N_i to N_j. Therefore, the *Physarum Learner* only contains correctly directed connections. In case of unknown ordering, however, a connection is only added to the *Bayesian Network* if

it does not cause a directed cycle in the graph. After the structure of the network has been established from the data, the WEKA software package is employed to learn the *Bayesian Network* parameters.

4 EXPERIMENTS

4.1 Configurations of the *Physarum Learner*

Structure learning results for 18 different test networks are quantified with the Bayesian score *BDeu* and the information theoretic score *MDL* which is equivalent to the Bayesian information criterion (BIC) score. Further, we compare the learnt structures with the known original structures (the ones we sampled the test data from) by measuring the number of learnt (A), extra (E), missing (M) and reversed (R) arcs. The conductivities D_{ij} of the connections have been initialized randomly in the range $D_{min} = 0.5 \le D_{ij} \le D_{max} = 1.0$. The constant l in eqn.(4) was fixed at $l = 0.1$.

The *Physarum Learner* has been applied to all 18 networks employing different values for γ. The following simulations have been performed: Each of the 18 networks has been learnt three times setting $\gamma = 1.0, 2.0, 3.0$. In these simulations, the best scored network has been learnt 3 times with $\gamma = 1.0$, 8 times with $\gamma = 2.0$ and 8 times with $\gamma = 3.0$. The network structure closest to the original one has been learnt 3 times with $\gamma = 1.0$, 9 times with $\gamma = 2.0$ and 8 times with $\gamma = 3.0$. The results corroborated that the value of γ clearly influences the quality of the results. It turned out that a larger γ exponent increases the deviations of length values. This results in the selection of paths over a larger number of nodes in the *Physarum Solver*. Additional results, not presented in this paper, show that for small networks the performance is better for $\gamma = 2.0$, but for larger networks (in terms of the number of arcs) $\gamma = 3.0$ leads to better results. Preliminary tests also showed a clear advantage in learning performance if the positive feedback term of the master equation was modelled as $f(Q) = |Q|^{\mu}$.

Next, the parameter μ was set to $\mu = 0.8, 1.0, 1.2, 1.4, 1.8$ while again learning each of the 18 networks. As the results for $\mu \ne 1.0$ depend on the randomly initialized values of the conductivities D_{ij}, the results presented have been averaged over 10 runs. A comparison indicating which value for μ performed best is given in Table 2. This table displays the number of cases out of 5×18 trials where a learned network was best in terms of

structure similarity to the true network (col.2), the BDeu score (col. 3) and the MDL score (col. 4). The most similar structure is identified as the one with the lowest value of the sums of extra and missing arcs.

Table 2: Counts where values of μ performed best.

μ	Structure	BDeu	MDL
0.8	4	5	6
1.0	**13**	**11**	**10**
1.2	9	7	7
1.4	9	5	5
1.8	6	9	9

The results clearly favour $\mu = 1.0$ in all three metrics. Note that the learnt structures for $\mu \neq 1.0$ differ in their performance, and that only average results have been compared.

4.2 Comparison of *Physarum Learner* and LAGD

Referring to the results presented, the *Physarum Learner* is compared to the LAGD Hill Climbing learning algorithm setting $l = 0.1, \gamma = 2.0, f(Q) = |Q|$. Both learning algorithms are evaluated on 18 benchmark data sets. Again the number of learnt, extra, missing and reversed arcs is measured compared to the original reference network from where the data is sampled. Note that the *Physarum Learner* cannot learn reversed arcs as attributes in the data sets are already given in the correct order. Furthermore, the BDeu and the MDL scores are computed also. The results indicate that the LAGD Hill Climbing outperforms the *Physarum Learner* in all test cases with respect to the BDeu and MDL score. This is not surprising, as LAGD optimizes BDeu and MDL during learning, while the *Physarum Learner* does not consider any score during learning at all. But especially in the three real networks, the *Physarum Learner* was able to learn network structures with a quality comparable to LAGD. In particular, *Physarum Learner* learned the exact true network for the Cancer data set, where LAGD has indeed the better scores, but two of the four arcs in the network are reversed.

Results also indicate that the *Physarum Learner* tends to learn less arcs for networks with a higher number of arcs. This is due to the fact, that the threshold θ is simply the average of all non-vanishing connection scores. The algorithm has no ability to fit the threshold to the requirements of the dataset. A detailed comparison of thresholds is needed but is subject to future work.

Removing the correct ordering from the datasets and learning the networks under same configurations again, the results for the three real datasets learned with the *Physarum Learner* are shown in Table 3. The results show a slight decrease of the scores com-

Table 3: *Physarum Learner* results for reordered real datasets.

Dataset	A	E	M	R	BDeu	MDL
Asia	9	3	2	2	-2475	-2498
Cancer	4	0	0	3	-2244	-2251
Earthquake	5	1	0	2	-542	-558

pared to the ordered measurements due to some reversed edges. As expected, the *Physarum Learner* selected exactly the same arcs as in the ordered version, because the direction of the arcs is not considered within the learner. Therefore, the number of extra and missing arcs is the same as for the ordered datasets, but some arcs are reversed. When neglecting the directions and no ordering is provided, the result of the *Physarum Learner* can be seen as a *Markov Network* instead of a *Bayesian Network*. To validate the learning performance of *Physarum Learner* for *Markov Networks*, we transformed the three real *Bayesian Networks* to *Markov Networks* using moralization (Koller and Friedman, 2009). The results are given in table 4.

Table 4: *Physarum Learner* networks compared to moralized true networks.

Dataset	Mor. Arcs	A	E	M
Asia	10	9	4	3
Cancer	5	4	0	1
Earthquake	5	5	1	1

The column *Mor. Arcs* displays the number of arcs in the moralized network. For *Asia*, the learned network is closer to the *Bayesian Network* version (2 extra and 2 missing arcs) than to the *Markov Network* (4 extra and 3 missing arcs). The same behaviour can be seen for *Cancer*, where *Physarum Learner* reconstructed the true network skeleton in table 3 but misses the extra arc created while moralization in table 4. For *Earthquake*, one additional arc is learned, but not the correct one.

Finally, the computational load for LAGD and *Physarum Learner* for sample sizes of 1000, 10.000, 100.000, 1.000.000 and 10.000.000 has been tested. It turned out that, although the complexity of the *Physarum Solver* grows with the number of nodes as n^2, it is three times more efficient as the *Physarum Learner*. This is because the latter needs to deal with the data set only once when calculating $\frac{n(n-1)}{2}$ Phi correlation coefficients.
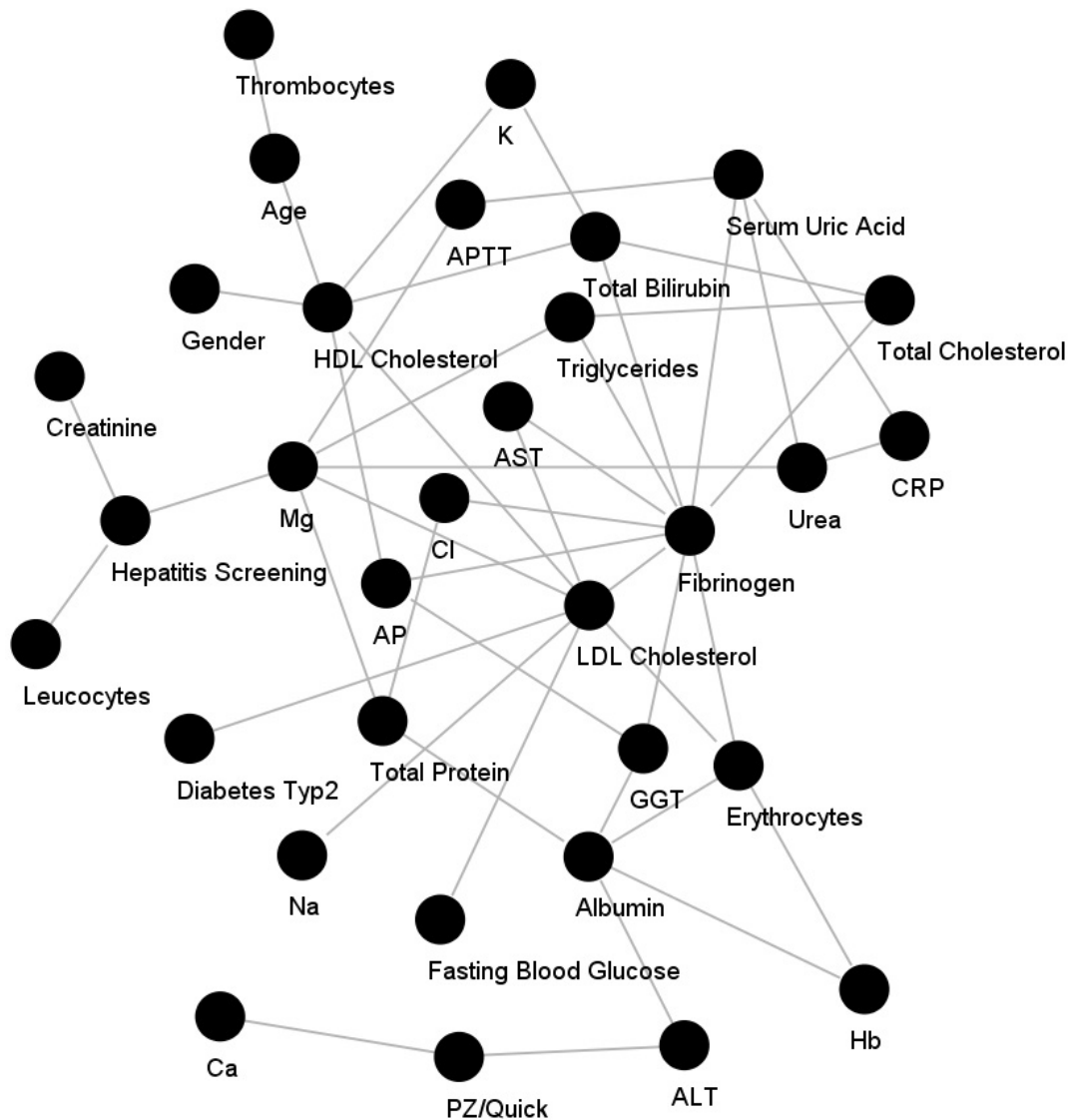
Figure 1: NAFLD Net learned by the Physarum Learner.

4.3 Applying the Physarum Learner to an NAFLD Dataset

Finally, the Physarum Learner performance is evaluated on a medical dataset retrieved at the Medical University of Graz (MUG) from 29 patients with liver-biopsy confirmed non-alcoholic fatty liver disease (NALFD). Patient data safety is followed throughout the entire process according to the standard operating procedures at the MUG. From the dataset, the parameters present in less than 50% of the patients and parameters with a constant value over all patients have been removed. The missing values of the remaining 32 attributes have been inserted by modes and means using the built in ReplaceMissingValues filter

of WEKA (http://www.cs.waikato.ac.nz/ml/weka/). Further, the dataset has been extended to 2900 instances by bootstrapping. As the dataset contains a mixture of real, nominal and ordinal attributes, and the Physarum Learner requires binary attributes, all attributes have been transformed to binaries.

The Physarum Learner is applied with configurations identical to the ones described in 4.2. The learned network shown in Fig. 1 is evaluated by medical experts of the MUG. As the Physarum Learner can only learn undirected graphs, hence is insensitive to the directions of edges, all interactions are considered undirected. All 42 connections displayed in Fig. 1 have been classified as comprehensible in a medical context. However, 35 possible connections, indi-

cated by general medical knowledge, have been identified as missing. For example, a connection between *Magnesium* (Mg) and *HDL Cholesterol*. From a medical point of view, the NAFLD graph generated by the Physarum Learner correctly reflects current medical knowledge and shows great potential for future applications and further refinement. The behaviour of the Physarum Learner is in line with previous experience showing that it tends to learn the stronger, hence more important, connections of the network.

5 CONCLUSIONS

A novel structure learning algorithm for *Bayesian Networks* has been introduced that initializes a fully connected network and uses the *Physarum Solver* to delete edges between nodes which are below a given correlation threshold. The *Physarum Learner* was compared to the LAGD Hill Climber where the *Physarum Learner* could learn competitive network structures for the three real networks, but performed worse than LAGD with respect to the learned BDeu and MDL score. It was shown, that it is generally possible to learn the structure of a *Bayesian Network* with the *Physarum Solver*. The *Physarum Learner* shows strongly growing execution time for networks with many nodes. This is in part because the algorithm has not yet been optimized for speed and memory usage. However, for networks with a small number of nodes and a large number of data points, the *Physarum Learner* is clearly more time efficient than LAGD and probably other score based heuristic methods.

ACKNOWLEDGEMENTS

REFERENCES

Brummitt, C., Laureyns, I., Lin, T., Martin, D., Parry, D., Timmers, D., Volfson, A., Yang, T., Yaple, H.,
and Rossi, M. L. (2010). A Mathematical Study of Physarum polycephalum. The Tero Model.

Davenport, E. C. and El-Sanhurry, N. A. (1991). Phi/Phimax: Review and Synthesis. *Educational and Psychological Measurement*, 51(4):821–828.

Ito, K., Johansson, A., Nakagaki, T., and Tero, A. (2011). Convergence Properties for the Physarum Solver. *arXiv:1101.5249v1*.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. The MIT Press.

Korb, K. and Nicholson, A. (2010). *Bayesian Artificial Intelligence*. Chapman and Hall, 2nd edition.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local Computation with Probabilities on Graphical Structures and their Application to Expert Systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 50(2):157–224.

Nakagaki, T. (2001). Smart behavior of true slime mold in a labyrinth. *Research in Microbiology*, 152(9):767–770.

Nakagaki, T., Yamada, H., and Hara, M. (2004). Smart network solutions in an amoeboid organism. *Biophysical Chemistry*, 107(1):1–5.

Nakagaki, T., Yamada, H., and Toth, A. (2000). Intelligence: Maze-solving by an amoeboid organism. *Nature*, 407(6803):470.

Nakagaki, T., Yamada, H., and Tóth, Á. (2001). Path finding by tube morphogenesis in an amoeboid organism. *Biophysical Chemistry*, 92(1–2):47–52.

Tero, A., Kobayashi, R., and Nakagaki, T. (2006). Physarum solver: A biologically inspired method of road-network navigation. *Physica A*, 363(1):115–119.

Tero, A., Kobayashi, R., and Nakagaki, T. (2007). A mathematical model for adaptive transport network in path finding by true slime mold. *J. Theo. Biol.*, 244(4):553–564.

Tero, A., Yumiki, K., Kobayashi, R., Saigusa, T., and Nakagaki, T. (2008). Flow-network adaptation in Physarum amoebae. *Theory in Biosciences*, 127(2):89–94.

Tomoyuki, M. and Isamu, O. (2008). Physarum can solve the shortest path problem on riemannian surface mathematically rigorously. *Int. J. Pure and Appl. Math.*, (47 (3)):353–369.