Bioinformatics, 36(5), 2020, 1553–1561
doi: 10.1093/bioinformatics/btz781
Advance Access Publication Date: 14 October 2019
Original Paper

Data and text mining

Exploiting transfer learning for the reconstruction of the human gene regulatory network

Paolo Mignone [1,2], Gianvito Pio [1,2,*], Domenica D'Elia [3] and Michelangelo Ceci [1,2,4]

[1]Department of Computer Science, University of Bari Aldo Moro, Bari 70125, Italy, [2]National Interuniversity Consortium for Informatics (CINI), Roma 00185, Italy, [3]Institute for Biomedical Technologies, CNR, Institute for Biomedical Technologies, Bari 70126, Italy and [4]Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana 1000, Slovenia

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on June 12, 2019; revised on September 13, 2019; editorial decision on October 7, 2019; accepted on October 9, 2019

Abstract

Motivation: The reconstruction of gene regulatory networks (GRNs) from gene expression data has received increasing attention in recent years, due to its usefulness in the understanding of regulatory mechanisms involved in human diseases. Most of the existing methods reconstruct the network through machine learning approaches, by analyzing known examples of interactions. However, (i) they often produce poor results when the amount of labeled examples is limited, or when no negative example is available and (ii) they are not able to exploit information extracted from GRNs of other (better studied) related organisms, when this information is available.

Results: In this paper, we propose a novel machine learning method that overcomes these limitations, by exploiting the knowledge about the GRN of a source organism for the reconstruction of the GRN of the target organism, by means of a novel transfer learning technique. Moreover, the proposed method is natively able to work in the positive-unlabeled setting, where no negative example is available, by fruitfully exploiting a (possibly large) set of unlabeled examples. In our experiments, we reconstructed the human GRN, by exploiting the knowledge of the GRN of *Mus musculus*. Results showed that the proposed method outperforms state-of-the-art approaches and identifies previously unknown functional relationships among the analyzed genes.

Availability and implementation: http://www.di.uniba.it/~mignone/systems/biosfer/index.html.

Contact: gianvito.pio@uniba.it

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

High throughput technologies and computational methods, mainly based on machine learning approaches, have significantly contributed to achieving so many goals in biology that nowadays can be considered fundamental for the understanding of biological processes in several organisms. Such processes are usually modeled through biological networks, which can describe the entities under study, as well as their relationships. The most prominent example is that of networks modeling the functional relationships among genes, called gene regulatory networks (GRNs) (Hartemink *et al.*, 2002). Indeed, the reconstruction and analysis of GRNs is considered fundamental for a better understanding of the mechanisms behind the expression of genes and the impact of their perturbation in the context of specific biological processes or pathways. These studies provide unprecedented opportunities for improvements in the diagnostics and treatment of different types of cancers and other human diseases (Ament *et al.*, 2018; Singh *et al.*, 2018).

The direct validation of GRNs can be technically and financially demanding (Penfold and Wild, 2011), because it is based on classical and complex in-lab experimental methods, such as transcriptome analysis, knock-down and overexpression strategies, medium throughput transcript quantification and chromatin immunoprecipitation (Park, 2009; Streit *et al.*, 2013), or protein-binding microarrays (Berger and Bulyk, 2009). For this reason, several machine learning methods have been proposed to infer the structure of such networks, based on different approaches, such as relevance networks, Bayesian Networks, clustering, differential equations, probabilistic models, random walk processes, Markov chains and maximum likelihood (Lu and Zhou, 2011). Independently of the specific approach, the prediction of the regulatory structure is usually performed on the basis of expression data of single genes or of pairs of genes (which represent possible interactions among them), measured under different conditions (e.g. in the presence of specific diseases or after treatment with drugs). Moreover, we can also find some attempts to improve the reconstruction accuracy

through the combination of the output of several prediction methods (Ceci *et al.*, 2015; Marbach *et al.*, 2012).

The most common limitation of existing reconstruction methods stems from their inability to produce a good model of the network when a poor amount of known interactions of the organism is available. This issue is even more exacerbated by the fact that available datasets about gene regulatory activities consist of only positive examples, i.e. they lack (or are very poor of) examples of proven non-existing regulatory activities (negative examples). In this context, classical machine learning methods fail to learn a prediction model if they do not make any assumptions about the existence of negative examples (Liu *et al.*, 2003; Zhang and Zuo, 2009). However, this solution may introduce some wrong biases (Hou *et al.*, 2018). On the contrary, machine learning tools that natively work in the positive-unlabeled (PU) setting are able to deal with this problem.

Additionally, existing reconstruction methods typically work on a single organism (Chang *et al.*, 2008) or tissue (Liu *et al.*, 2014), because of the considered representation requirements and the lack of organism-independent information in the available data sources.

Motivated by such limitations, in this paper we propose a novel transfer learning method, specifically designed to obtain a high-quality reconstruction of GRNs, even when the amount of labeled examples is poor. In particular, the proposed method is able to 'transfer' the knowledge about one domain (called source) to another (called target). In this way, we use the knowledge about the GRN of a model organism for the reconstruction of the GRN of another organism, aiming at exploiting possible analogies, in terms of regulatory activities, among the organisms (Denas *et al.*, 2015). Moreover, the proposed method is natively able to work in the PU setting, where we have (i) few positively labeled examples (i.e. known existing interactions) for both the source and the target networks; (ii) a huge amount of unlabeled examples for both the source and the target networks; (iii) no negative example.

The rest of the paper is organized as follows: in Section 2, we introduce some background notions about transfer learning and briefly review some existing methods; in Section 3, we describe the proposed method; in Section 4, we discuss the results obtained in the reconstruction of the human GRN, by exploiting the mouse GRN as source network; in Section 5, we show the result of a gene-functional analysis. Finally, in Section 6, we draw some conclusions and outline possible future work.

2 Background

In the classical supervised setting, machine learning methods analyze a training set of examples (i.e. a sample of the population), described by a set of features and associated with a known category (label), and learn a prediction model which is able to assign a category to new, unseen examples. This setting usually works effectively when the set of examples is large enough and if each category is properly represented within the considered sample. Classical supervised algorithms can be naturally applied to solve the task of network reconstruction, where: (i) each example corresponds to a (possible) regulatory activity between two genes; (ii) features correspond to expression data regarding the two genes; (iii) labels can be {*Yes*, *No*}, representing the fact that the interaction exists or not. In the classical supervised learning setting, however, the quality of the reconstruction may be affected by the poor availability of labeled examples, which, in this specific case, are usually only positive, i.e. they are only examples of existing interactions or functional relationships.

In order to alleviate the problem of poor availability of labeled examples, one possibility is to consider a multi-organism scenario, where missing information about the functional relationships between two genes in one organism can be compensated by the presence of known relationships between similar (e.g. homologous or, more specifically, orthologous) genes in other organisms (Warwick Vesztrocy *et al.*, 2018).

When this is possible, transfer learning techniques (Weiss *et al.*, 2016) can provide substantial help. Indeed, such techniques aim at exploiting the knowledge about another related domain, called *source domain*, to improve the quality of the results on the main domain, called *target domain*. Formally, in a classical supervised learning setting, given:

- the feature space X of training examples;
- the output space Y, i.e. the set of possible labels;
- n training examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, s.t. $x_i \in X$ and $y_i \in Y$;

the goal is to learn a function f: $X \rightarrow Y$, which is able to predict the label of unseen, unlabeled examples.

Transfer learning differs from this formulation since it works on two different domains simultaneously. Formally, given:

- the source and the target feature spaces X_s and X_t;
- the output spaces Y_s and Y_t;
- n_s training examples for the source domain in the form $\{(x_1^s, y_1^s), (x_2^s, y_2^s), \ldots, (x_{n_s}^s, y_{n_s}^s)\}$, s.t. $x_i^s \in X_s$ and $y_i^s \in Y_s$;
- n_t training examples for the target domain in the form $\{(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_{n_t}^t, y_{n_t}^t)\}$, s.t. $x_i^t \in X_t$ and $y_i^t \in Y_t$;

the goal is to learn a function $f_t : X_t \rightarrow Y_t$, also exploiting the knowledge acquired by learning a function $f_s : X_s \rightarrow Y_s$ on the source domain. A graphical view of the transfer learning setting is shown in Figure 1.

In the literature, we can find many applications of transfer learning techniques in several domains, such as the classification of Web documents (Dai *et al.*, 2007), WiFi localization (Pan *et al.*, 2008), sentiment analysis (Pan *et al.*, 2010) and software defect prediction (Jinyin *et al.*, 2019). For a comprehensive list of possible applications of transfer learning we refer the reader to the survey by Weiss *et al.* (2016). However, in bioinformatics, transfer learning approaches have been only marginally exploited. One of the notable approaches is proposed by Achanta *et al.* (2016), where the idea is to identify breast cancer tumors by exploiting two different data sources. A different approach (Sevakula *et al.*, 2018), based on deep learning, has been employed for molecular cancer classification, where the feature representation learned while classifying two tumor types also exploits information conveyed, in an unsupervised manner, by other tumor types. In another paper, Breckels *et al.* (2016) propose to extend a state-of-the-art transfer learning framework to solve the predictive task of mouse protein sub-cellular localization. Finally, Mignone and Pio (2018) propose a first attempt (hereafter PU-TL-LP) to perform gene network reconstruction by exploiting the knowledge of another organism via transfer learning. In this case, source and target examples are used altogether to learn one classifier for the target organism. This leads to overestimate the importance of source examples and to not properly model the regulatory activities of the source organism. On the contrary, we learn two separate models for the two organisms and properly combine their contributions. Additionally, the experimental evaluation in Mignone and Pio (2018) is very preliminary, without any *in silico* functional analysis from a biological point of view.

From a methodological viewpoint, Weiss *et al.* (2016) classify transfer learning approaches according to the homogeneity or the heterogeneity of the feature spaces (X_s and X_t) and of the solved tasks. The method we propose works with homogeneous feature spaces (i.e. $X_s = X_t$), whereas the solved tasks are, according to Weiss *et al.* (2016), 'different but related' (i.e. $f_t \neq f_s$). Our solution is based on parameter-transfer (the goal is to discover and exploit shared parameters or priors between the source domain and target domain models), where the model f_t is learned by considering the model f_s, learned from a training set that possibly follows a different data distribution.

Except for the preliminary work in Mignone and Pio (2018), all the existing transfer learning methods are able to work when both

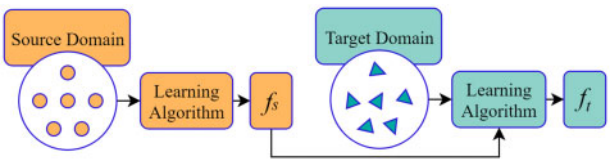

Fig. 1. Transfer learning setting, which exploits an additional (source) domain

positive and negative examples are available. On the contrary, in this paper we work in the PU learning setting (Elkan and Noto, 2008), where negative examples are not available. Although it is, in principle, possible to address the transfer learning problem in the PU setting by discarding unlabeled examples and by learning a One-Class classifier (Xiao *et al.*, 2015), this would lead to ignoring possibly useful information conveyed by unlabeled examples (He *et al.*, 2010). The approach we propose in this paper overcomes this limitation by directly working in the PU learning setting, by exploiting a large amount of unlabeled examples usually available in GRN reconstruction. To the best of our knowledge, this is the first method with such characteristics.

3 System and methods

Before describing the proposed transfer learning method, we introduce some useful notions and formally define the network reconstruction task, in the PU setting, for a single domain. Let:

- V be the set of nodes of the network;
- $x = \langle v', v'' \rangle \in (V \times V)$ be a (possible) link between two nodes v' and v'', where $v' \neq v''$;
- $z(v) = [z_1(v), z_2(v), \ldots, z_m(v)]$ be the vector of features associated with the node v, where $z_i(v) \in \mathbb{R}$, $\forall i \in \{1, 2, \ldots, m\}$;
- $q(x) = [q_1(x), q_2(x), \ldots, q_{2m}(x)] = [z_1(v'), \ldots, z_m(v'), z_1(v''), \ldots, z_m(v'')]$ be the feature vector of the link $x = \langle v', v'' \rangle$;
- *sim*: $\mathbb{R}^{2m} \times \mathbb{R}^{2m} \rightarrow [0, 1]$ be a similarity function between the $2m$-dimensional feature vectors associated with two links;
- $L \subset (V \times V)$ be the set of (only positive) labeled links;
- $U = (V \times V) \setminus L$ be the set of unlabeled links.

The network reconstruction task aims to find a function $f : \mathbb{R}^{2m} \rightarrow [0, 1]$ that takes a vector of features $q(x)$ associated with the unlabeled link x and returns a score representing the degree of certainty of its existence.

As introduced in Section 2, our transfer learning method works with two different domains, i.e. the source domain and the target domain. Following the previous formalization, henceforth we will use the subscripts s and t to refer to the source and the target domains, respectively (e.g. L_s will refer to the set of labeled links of the source domain).

Our method consists of four stages. The first two stages aim to solve the problem of the poor availability of labeled examples. In particular, in the first stage we apply a clustering algorithm on the positive links, in order to identify prototype links, i.e. centroids, that summarize the different kinds of positive links in the feature space. In the second stage we assign a weight in [0,1] to each unlabeled link, according to its similarity with the centroids identified in the first stage. This weight provides us with an indication of the position, in the feature space, of unlabeled links with respect to known existing prototypes of links. In the third stage, we exploit both the labeled links and the set of weighted unlabeled links to learn an instance-weighted classification model, based on weighted support vector machines (WSVM). Thanks to the result of the first two stages, the classifier will be able to exploit unlabeled examples. This means that our approach works in the semi-supervised learning setting and is able to exploit the 'smoothness assumption' (Levatic *et al.*, 2017), namely if two links are close (in the feature space $\mathbb{R}^{2m}$) than their degree of certainty should be similar.

These three stages are performed for both the source and the target domains, independently. Finally, in the fourth stage, according to the parameter-transfer framework, we combine the models obtained for the source and for the target domains, in order to build a new model that better describes the target network, also according to the knowledge acquired from the source network model. It is noteworthy that this stage is not based on ensemble learning, but on a geometrical data-driven model combination, which guarantees a better model stability (i.e. the model does not depend on any preliminary sampling step). The general workflow of the proposed method is depicted in Figure 2, while each stage is detailed in the following subsections.

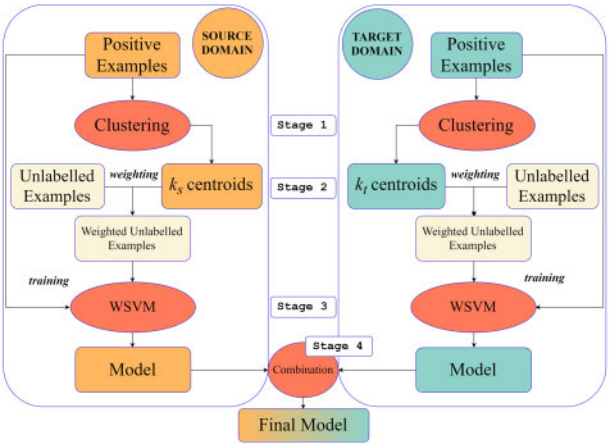

Fig. 2. Overview of the proposed transfer learning approach

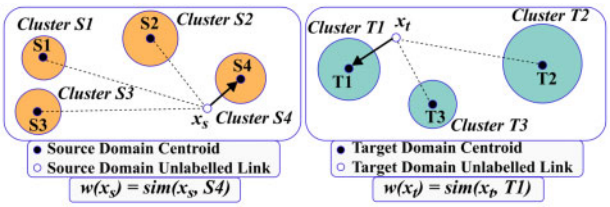

Fig. 3. Stage II–weighting of unlabeled links

3.1 Stage I: clustering

The first stage aims at identifying groups of similar interactions, each with its own specificity. Indeed, as emphasized by Stark *et al.* (2006), there are different kinds of possible interactions among genes, such as genetic interactions, protein–protein interaction, chemical associations or post-translational modifications. Therefore, we apply a clustering algorithm aiming to identify, in an unsupervised fashion, different kinds of interactions. In particular, we exploit a prototype-based clustering algorithm, in order to identify k_s and k_t prototypes of positive interactions, from the positive examples of each domain, i.e. from L_s and L_t. In this paper, we adopt the classical *k-means* algorithm, since it is well established in the literature. However, any other prototype-based clustering algorithm, possibly able to catch specific peculiarities of the data at hand, could be plugged into our method.

3.2 Stage II: instance weighting

Each unlabeled link could be either a positive or negative example. In this stage, for each unlabeled link, we compute a weight representing the degree of certainty in $[0, 1]$: a value close to 0 means that the example is likely to be a negative example, whereas a value close to 1 means that the example is likely to be a positive example. This weight is computed according to the similarity between the feature vector associated with the unlabeled link and the feature vectors of the cluster prototypes. In particular, the weight of an unlabeled link x_s (respectively, x_t) is computed according to its similarity with respect to its closest centroid identified from L_s (respectively, L_t) in the first stage. Formally:

$$\begin{aligned} w(x_s) &= \max_{c \in C_s} (\mathrm{sim}(q(x_s), c)) \\ w(x_t) &= \max_{c \in C_t} (\mathrm{sim}(q(x_t), c)) \end{aligned} \quad (1)$$

where C_s (respectively, C_t) is the set of cluster prototypes identified from L_s (respectively, L_t), and sim is a similarity function (Fig. 3). In this paper, we exploit the Euclidean distance, after applying a min–max normalization (in the range $[0, 1]$) to all the features. Formally: $\mathrm{sim}(q(x'), q(x'')) = 1 - \sqrt{\sum_{i=1}^{2n} (q_i(x') - q_i(x''))^2}$.

An overview of the weighting strategy can be observed in Figure 3.

It is noteworthy that working with clusters, and representing them as feature vectors of their prototypes, also provides a clear advantage in terms of computational complexity. Indeed, the computation of similarities necessary to estimate the weights [Equation (1)] can be performed with respect to centroids, rather than single examples. This consideration further motivates the first stage of our approach.

3.3 Stage III: learning instance-weighted probabilistic classifiers

In the third stage, we train a probabilistic classifier for each domain, based on linear WSVM (Yang *et al.*, 2007) with Platt scaling (Platt, 1999). We selected an SVM-based classifier mainly because (i) it has a relatively good computational efficiency, especially in the prediction phase, and (ii) it has already proved to be effective (with Platt scaling) in the semi-supervised setting (Elkan and Noto, 2008).

The weighted implementation of SVMs has the advantage of catching the different weights associated with the training examples. In particular, we consider the positive labeled examples as certainly positive (i.e. $w(x) = 1.0, \forall x \in L$), while we consider the unlabeled examples as positive examples with a lower degree of certainty, according to the weight computed in Stage II. The output of the training phase is a hyperplane function h, which, contrary to classical SVM classification, is not necessarily intended as a discrimination function between positive and negative examples. Rather, the hyperplane function h is intended as a way to understand how much an example is likely to be negative. Moreover, coherently with the classical formulation of SVMs, the area close to the hyperplane has the highest degree of uncertainty.

Therefore, the hyperplane is learned by optimizing the following factors: (i) training examples should appear on the same side of the hyperplane; (ii) high-weighted (i.e. possibly positive) examples should appear far from the hyperplane; (iii) low-weighted (i.e. possibly negative) examples should appear close to the hyperplane (Fig. 4). Finally, by exploiting the Platt scaling, for each unlabeled link x, we can compute the probability estimation of being a positive example as $f(q(x)) = 1/(1 + e^{-h(q(x))})$, where $h(q(x))$ represents the signed Euclidean distance between x and the hyperplane h.

The output of this stage consists of two hyperplanes, i.e. the hyperplane for the source domain h_s, learned from L_s and U_s, and the hyperplane for the target domain h_t, learned from L_t and U_t.

3.4 Stage IV: model combination for knowledge transfer

In the fourth stage, we apply our transfer learning approach. In particular, we exploit the hyperplanes h_s and h_t in an equation discovery algorithm, whose main goal is to identify the best hyperplane for the target domain. The problem of equation discovery (Todorovski, 2017) consists in the identification of the most appropriate coefficients of a set of equations, guided by data and based on heuristics. The heuristics are used to reduce the inherent complexity of the problem (i.e. of class NP-hard; Cubitt *et al.*, 2012), which makes an exhaustive search in the space of the possible hypothesis unfeasible.

In order to guide the search into the space of the hypothesis (equations), we use the parameters (coefficients) of the hyperplanes h_s, h_t. Both the hyperplanes are defined in the feature space defined by $q(\cdot)$, indicating the feature variables with $q_1, q_2, \ldots, q_{2m}$.

Additionally, we use a further hyperplane, h_b, which is the bisector of the acute angle between the source and the target hyperplanes. h_b intuitively represents the 'average' hyperplane [Fig. 5b for a geometric interpretation]. Let h_s and h_t be the following hyperplanes:

$$\begin{aligned} h_s &: \alpha_1 q_1 + \alpha_2 q_2 + \cdots + \alpha_{2m} q_{2m} + \alpha_{2m+1} = 0 \\ h_t &: \beta_1 q_1 + \beta_2 q_2 + \cdots + \beta_{2m} q_{2m} + \beta_{2m+1} = 0 \end{aligned} \tag{2}$$

the bisector hyperplane of their acute angle is:

$$h_b : \gamma_1 q_1 + \gamma_2 q_2 + \cdots + \gamma_{2m} q_{2m} + \gamma_{2m+1} = 0 \tag{3}$$

where $\forall i \in [1, 2, \ldots, 2m+1]$:

$$\gamma_i = \frac{\alpha_i}{\sqrt{\alpha_1^2 + \alpha_2^2 + \cdots + \alpha_{2m}^2}} + s \cdot \frac{\beta_i}{\sqrt{\beta_1^2 + \beta_2^2 + \cdots + \beta_{2m}^2}}$$

where $s = +1$ if $\alpha_1\beta_1 + \alpha_2\beta_2 + \cdots + \alpha_{2m}\beta_{2m} > 0$; $s = -1$ otherwise.

Given the three hyperplanes h_s, h_t and h_b, we apply a greedy iterative strategy for equation discovery to evaluate if some knowledge from the source domain can be injected into the target hyperplane. In particular, the search space is defined in terms of all the possible equations of the form:

$$h_{st} : \delta_1 q_1 + \delta_2 q_2 + \cdots + \delta_{2m} q_{2m} + \delta_{2m+1} = 0 \tag{4}$$

where

$$\delta_i \in \{\alpha_i, \beta_i, \gamma_i\} \quad \forall i \in 1, 2, \ldots, 2m+1. \tag{5}$$

This means that the search space of possible hyperplanes has size 3^{2m+1}. Actually, all these hyperplanes can be considered as the result of a weighted combination of the source and the target hyperplanes, within the sheaf of hyperplanes defined by the source and the target hyperplanes. This search space is explored according to a greedy search that, at each step, maximizes the recall@1% measure on the

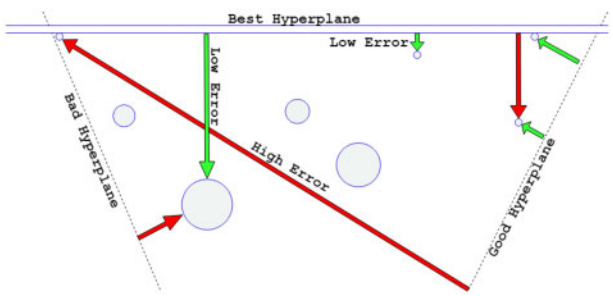

Fig. 4. Training of a weighted SVM in the positive-unlabeled setting. Big circles represent examples with a high weight (possibly positive examples), while small circles represent examples with a low weight (possibly negative examples). A good hyperplane should be close to small circles and distant from big circles

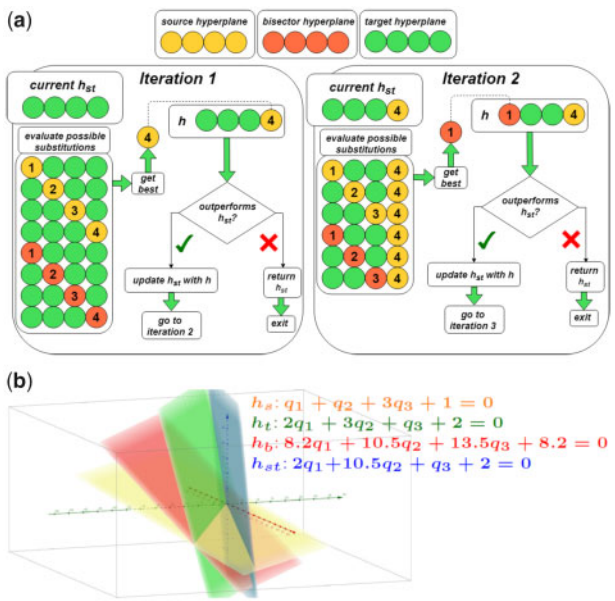

Fig. 5. (**a**) An example of two consecutive iterations of the identification of the best substitution on the current hyperplane (function *get_best_sub* in Algorithm 1–Line 19). (**b**) A graphical representation of the combination of source (yellow), target (green) and bisector (red) hyperplanes. The method returns the new combined hyperplane (blue). (Color version of this figure is available at *Bioinformatics* online.)

training set: the percentage of validated interactions among the top-1% of those returned by the algorithm. This choice is the result of a preliminary experiment, where we evaluated the quality of the reconstruction with different percentages for the recall@k measure (Results in Supplementary File S4).

Algorithm 1. Pseudocode of our transfer learning approach

Data:
- $\cdot L_s, L_t$: positive links of the source and the target domains
- $\cdot U_s, U_t$: unlabeled links of the source and the target domains
- $\cdot k_s, k_t$: number of clusters for the source and the target domains
- $\cdot eval$: a function that evaluates the quality of a hyperplane with respect to a set of examples

Result:
A prediction function $f: \mathbb{R}^{2m} \rightarrow [0, 1]$ which takes a vector of features $q(x)$ associated with an unlabeled link x and returns a score representing the degree of certainty of its existence.

begin
/* Stage I - Clustering */
$C_s \leftarrow kmeans(L_s, k_s)$; $C_t \leftarrow kmeans(L_t, k_t)$;
/* Stage II - Instance weighing */
$TR_s \leftarrow \emptyset$; $TR_t \leftarrow \emptyset$;
foreach $x_s \in L_s$ **do**
  $TR_s \leftarrow TR_s \cup \{\langle q(x_s), 1.0\rangle\}$
foreach $x_s \in U_s$ **do**
  $w(x_s) \leftarrow max_{(c_s \in C_s)}(sim(q(x_s), c_s))$;
  $TR_s \leftarrow TR_s \cup \{\langle q(x_s), w(x_s)\rangle\}$;
foreach $x_t \in L_t$ **do**
  $TR_t \leftarrow TR_t \cup \{\langle q(x_t), 1.0\rangle\}$
foreach $x_t \in U_t$ **do**
  $w(x_t) \leftarrow max_{(c_t \in C_t)}(sim(q(x_t), c_t))$;
  $TR_t \leftarrow TR_t \cup \{\langle q(x_t), w(x_t)\rangle\}$;
/* Stage III - Learning of weighted SVM */
$h_s \leftarrow WSVM(TR_s)$; $h_t \leftarrow WSVM(TR_t)$;
/* Stage IV - Combination of models */
$h_b \leftarrow bisector(h_s, h_t)$; /* Equation (3) */
$h_{st} \leftarrow h_t$; $best\_score \leftarrow eval(h_{st}, TR_t)$;
$subs \leftarrow \emptyset$;
repeat
  $\langle h, subs\rangle \leftarrow get\_best\_sub(h_{st}, h_s, h_b, subs, TR_t)$;
  $score \leftarrow eval(h, TR_t)$;
  if $score > best\_score$ **then**
    $h_{st} \leftarrow h$;
    $best\_score \leftarrow score$;
until $score \leq best\_score$;
$f(\cdot) = \frac{1}{1+e^{-h_{st}(\cdot)}}$; /* Platt Scaling */
return $f(\cdot)$;

Algorithmically, starting from the target hyperplane h_t, we evaluate possible substitutions of its coefficients as follows. Let $2m$ be the number of features. At the first iteration, we evaluate $2m+1$ possible substitutions from the source hyperplane h_s and $2m+1$ possible substitutions from the bisector hyperplane h_b. If one of the substitutions provides a higher recall@1% in the reconstruction with respect to the initial hyperplane, we keep such a coefficient and repeat the process, evaluating further possible substitutions for the other coefficients (Fig. 5a). When further substitutions do not provide any improvement on the result, the iterative process stops and returns the final hyperplane h_{st}. In Figure 5b, we show an example where the final model h_{st} (the blue hyperplane) is the result of the substitution of the coefficient of the variable q_2 of the target hyperplane with the respective coefficient of the bisector hyperplane. The hyperplane h_{st} can be finally adopted to predict the degree of certainty of each unlabeled link in U_t.

A formal description of the whole transfer learning approach proposed in this paper is reported in Algorithm 1. According to the greedy search approach, the algorithm works in a top-down fashion and at each step identifies the locally optimal choice, with the final intent of finding an approximation of global optimum in a reasonable amount of time.

4 Evaluation

We implemented our approach in the system **BioSfer** (**Bio**logical tran**Sfer** learning) and focused our evaluation on the reconstruction of the human gene network (target domain), by exploiting and transferring knowledge extracted from the mouse gene network (source domain), since the mouse is a model organism for the human. For the sake of completeness, we also performed the experiments for the opposite task, i.e. the reconstruction of the mouse GRN considering the human as source. All the experiments were performed on a server with 8 CPUs and 64 GB of RAM.

4.1 Datasets

The dataset adopted in the experiments was built by considering gene expression data related to 6 different organs (liver, lung, brain, skin, bone marrow, heart), obtained by control samples available at Gene Expression Omnibus (GEO) (www.ncbi.nlm.nih.gov/geo/). Overall, 161 and 174 raw samples were considered for mouse and human organisms, respectively. All the samples were processed according to the workflow adopted for the DREAM5 challenge (Marbach *et al.*, 2012), that led to a dataset of 5404 mouse genes and to a dataset of 15 345 human genes. Each gene was associated with six features (one for each organ), by averaging the expression levels measured within the same organ. Finally, the dataset of the interactions among genes was built by considering all the possible pairs of genes (excluding the self-links), each associated with the concatenation of the feature vectors of the genes involved in the interaction. Therefore, each link was represented through a 12D feature vector.

We extracted 235 706 validated human gene interactions (i.e. L_t) and 14 613 validated mouse gene interactions (i.e. L_s) from BioGRID (available at https://thebiogrid.org). As regards the unlabeled examples of interactions (i.e. U_s and U_t), we randomly selected, without replacement, a balanced number (i.e. equals to the number of labeled examples) of interactions involving at least one gene that appears in the set of labeled interactions.

4.2 Experimental setting

Since the first stage of our method exploits the k-means clustering algorithm, we estimated the optimal value for k_s and k_t (Section 3) on the basis of the silhouette coefficient (Rousseeuw, 1987). According to the results, shown in Figure 6, we selected the best configurations, i.e. $k_s \in \{2, 3, 4\}$ and $k_t \in \{2, 3, 4\}$, for the subsequent experiments.

Since we work in the PU learning setting (i.e. the dataset does not contain any negative examples), similarly to the evaluation

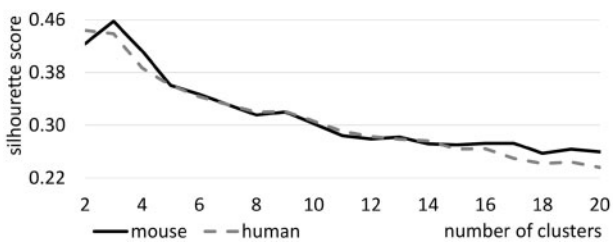

Fig. 6. Analysis of the silhouette coefficient for mouse (k_s) and human (k_t)

performed by Pio *et al.* (2015), we evaluated the results in terms of recall@k (r@k) and area under the r@k curve (AUr@k). Moreover, we also report the results in terms of area under the ROC curve (AUROC) and area under the precision-recall curve (AUPR). Note that the r@k and the AUr@k measures do not introduce any (possibly wrong) bias in the ground truth, while for the computation of the AUROC and AUPR measures, it is necessary to consider the unlabeled examples as negative examples.

All the experiments have been performed through a 10-fold cross-validation, where each fold consists of 9/10 of positive examples for training and 1/10 of positive examples for testing, while all the unlabeled examples are considered for both training and testing. Thus, the experimental setting we consider is the classical one used in *semi-supervised learning* and typically adopted in network data analysis.

We initially compared BioSfer with three methods. The first is the system **PU-TL-LP** proposed in Mignone and Pio (2018), whereas the other two are variants of BioSfer, called **no_transfer**, which discards the information about the mouse gene network, and **union**, which treats examples of mouse gene interactions like human gene interactions. Moreover, we compared **BioSfer** with a state-of-the-art method for Gene Network Reconstruction, i.e. **GENIE3** (Aibar *et al.*, 2017; Huynh-Thu *et al.*, 2010), and with three state-of-the-art transfer learning methods, namely **JGSA** (Zhang *et al.*, 2017), **BDA** (Wang *et al.*, 2017) and **TJM** (Long *et al.*, 2014), that we introduced in Section 2. It is noteworthy that JGSA, BDA and TJM are not able to natively work in the PU learning setting. Therefore, we exploited the ranking of interactions produced by the first stage of BioSfer, and considered the top-33.3%, the top-50% or the top-66.6% as positive examples for them. The remaining examples were considered as negative examples. For **JGSA**, **BDA** and **TJM**, we optimized the input parameter dim $\in \{1, 6, 12\}$, that is the number of features of the common feature space (between the source and the target domains), identified by such methods. Since they do not return a ranking, but only a set of predicted interactions, we measured the average r@k computed on 100 random shuffles of the result. This solution guarantees a good estimation of the number of true positives in the top-k returned interactions. For GENIE3, we adopted the variant based on Random Forests, with 50 trees and the number of features evaluated at each split of the trees equals to the square root of the number of features, as suggested in Breiman (2001).

Table 1. AUROC, AUPR and AUr@k results obtained by no_transfer, union, PU-TL-LP and BioSfer on the full dataset

	no_transfer	union	PU-TL-LP	BioSfer
AUROC	0.680	0.688	0.689	**0.701**
AUPR	0.190	0.201	0.202	**0.214**
AUr@k	0.679	0.687	0.688	**0.700**

The best results are highlighted in boldface.

4.3 Results and discussion

In Table 1, we show the obtained AUROC, AUPR and AUr@k, while in Figure 7 we show the improvement, in terms of r@k (for different values of k), of **BioSfer**, **union** and **PU-TL-LP** over the approach which does not exploit transfer learning (i.e. **no_transfer**), measured with different values of k_s and k_t. The improvement of **BioSfer**, **union** and **PU-TL-LP** is computed as $(r@k_{\text{BioSfer}} - r@k_{\text{no\_transfer}})/r@k_{\text{no\_transfer}}$, $(r@k_{\text{union}} - r@k_{\text{no\_transfer}})/r@k_{\text{no\_transfer}}$ and $(r@k_{\text{PU-TL-LP}} - r@k_{\text{no\_transfer}})/r@k_{\text{no\_transfer}}$, respectively. We can observe that **union**, **BioSfer** and **PU-TL-LP** are able to obtain better results with respect to **no_transfer**. This result and the amount of the improvement (almost 20% in some cases) confirms that the transfer learning approach conveys strong improvement in terms of recall@k. This confirms our initial intuition that the information conveyed by mouse gene interactions can be helpful to improve the quality of the reconstruction of the human gene network, and that transfer learning techniques can be profitably exploited for this purpose. Moreover, by comparing **union** and **PU-TL-LP** with **BioSfer**, also in terms of AUROC, AUPR and AUr@k (Table 1), we can observe that our method obtains much better results, which proves that the adopted strategies for weighting the examples and combining the models actually leads to a better reconstruction. This is even more evident in the first part of the curve, where our method is able to retrieve up to 189 *additional* true interactions over 2592 possible interactions at the top-1% of the ranking (Table 2 for the detailed number of correctly returned interactions). Finally, by comparing the results obtained with different values of k_s and k_t, the best result is obtained with $k_s = 3$ and $k_t = 2$, which is coherent with the result obtained by the analysis of the silhouette coefficient (Fig. 6). This means that the proposed clustering-based approach can positively contribute to the final quality of the reconstruction, but the superiority of **BioSfer** is notable independently of the considered number of clusters.

Focusing on the best configuration ($k_s = 3$ and $k_t = 2$), we compare the results obtained by **BioSfer** with those obtained by state-of-the-art approaches, on a subset consisting of 2% of the initial dataset. We focus this comparison on such a small portion because it is the only setting for which all the considered approaches are able to provide a result with the available computational resources. In particular, only **BioSfer** is able to analyze the whole dataset, while **GENIE3** is able to analyze up to 50% of the dataset, **BDA** and **TJM** are able to analyze only up to 5% of the dataset, and **JGSA** is not able to analyze more than 2%. By observing the results in Figure 8, we can conclude that **BioSfer** is able to obtain better results with respect to all the considered competitors, and that the measured r@k is consistently higher than that achieved by competitors, regardless of the value of k. This clear dominance is confirmed by

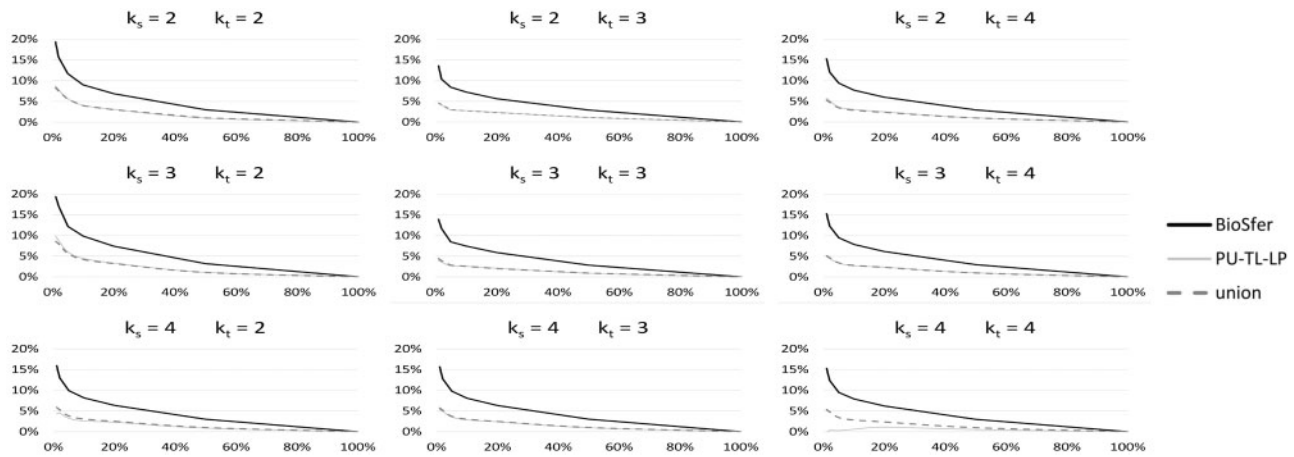

Fig. 7. Improvement over **no_transfer** in terms of recall@k, with different values of k_s and k_t, and over different values of k (x-axis)

Table 2. Number of true positives, at different percentages of the ranking

		1%	2%	5%	10%	20%	50%	100%
$k_s = 2$,	no_transfer	979	1670	3382	5708	9424	17 194	23 571
$k_t = 2$	union	1060	1795	3568	5934	9709	17 369	23 571
	PU-TL-LP	1064	1801	3575	5941	9715	17 367	23 571
	BioSfer	**1168**	**1931**	**3778**	**6222**	**10 072**	**17 711**	23 571
$k_s = 2$,	no_transfer	1023	1737	3476	5793	9495	17 186	23 571
$k_t = 3$	union	1071	1811	3581	5949	9711	17 378	23 571
	PU-TL-LP	1069	1808	3579	5952	9718	17 374	23 571
	BioSfer	**1163**	**1917**	**3769**	**6213**	**10 031**	**17 685**	23 571
$k_s = 2$,	no_transfer	1011	1720	3455	5779	9490	17 202	23 571
$k_t = 4$	union	1065	1803	3574	5943	9714	17 372	23 571
	PU-TL-LP	1070	1809	3580	5955	9724	17 377	23 571
	BioSfer	**1166**	**1928**	**3779**	**6224**	**10 060**	**17 705**	23 571
$k_s = 3$,	no_transfer	977	1663	3377	5698	9406	17 185	23 571
$k_t = 2$	union	1060	1795	3568	5938	9709	17 370	23 571
	PU-TL-LP	1072	1808	3579	5953	9718	17 376	23 571
	BioSfer	**1166**	**1946**	**3787**	**6258**	**10 103**	**17 734**	23 571
$k_s = 3$,	no_transfer	1027	1744	3484	5806	9524	17 220	23 571
$k_t = 3$	union	1072	1812	3582	5955	9715	17 378	23 571
	PU-TL-LP	1068	1807	3578	5954	9726	17 375	23 571
	BioSfer	**1170**	**1948**	**3779**	**6239**	**10 085**	**17 710**	23 571
$k_s = 3$,	no_transfer	1012	1721	3455	5780	9492	17 204	23 571
$k_t = 4$	union	1064	1801	3572	5941	9713	17 371	23 571
	PU-TL-LP	1062	1798	3571	5936	9717	17 376	23 571
	BioSfer	**1167**	**1933**	**3783**	**6234**	**10 076**	**17 715**	23 571
$k_s = 4$,	no_transfer	1006	1712	3442	5766	9477	17 199	23 571
$k_t = 2$	union	1065	1803	3573	5943	9715	17 372	23 571
	PU-TL-LP	1050	1789	3554	5916	9695	17 352	23 571
	BioSfer	**1167**	**1935**	**3784**	**6238**	**10 080**	**17 718**	23 571
$k_s = 4$,	no_transfer	1008	1715	3447	5770	9481	17 198	23 571
$k_t = 3$	union	1066	1804	3575	5945	9715	17 373	23 571
	PU-TL-LP	1062	1800	3569	5936	9709	17 364	23 571
	BioSfer	**1166**	**1934**	**3784**	**6238**	**10 080**	**17 718**	23 571
$k_s = 4$,	no_transfer	1012	1721	3456	5781	9493	17 204	23 571
$k_t = 4$	union	1066	1804	3575	5945	9716	17 373	23 571
	PU-TL-LP	1009	1727	3466	5815	9598	17 296	23 571
	BioSfer	**1167**	**1935**	**3784**	**6238**	**10 082**	**17 719**	23 571

Note: The reported average number of true positives refers to a single testing fold, including 1/10 of positive examples and all the unlabeled examples.

The best results are highlighted in boldface.

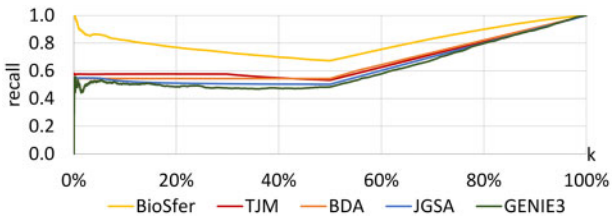

Fig. 8. Recall@k results, obtained by BioSfer ($k_s = 3$ and $k_t = 2$) and by all the considered competitors (in their best configuration). A recall@k close to 1 means that almost all the k returned interactions are positive (correctly returned)

the AUROC, AUPR and AUr@k values achieved by BioSfer, compared with those obtained by JGSA, BDA, TJM and GENIE3 (Table 3). A possible explanation of the significant superiority of BioSfer stems from its ability to natively work in the PU setting, without artificially identifying potential negative examples. The effect is that users who adopt BioSfer are guaranteed to identify the most promising interactions, regardless of the considered value of k (i.e. the size of the list).

Finally, we performed an analysis of the required computational resources, in terms of running time and RAM usage. In Figure 9, it is possible to observe that, although **BioSfer** took a slightly higher time to analyze the smallest sample of the dataset, it required a running time which is up to two orders of magnitude lower than that required by **JGSA**, **BDA**, **TJM** and **GENIE3** for larger datasets.

Table 3. AUROC, AUPR and AUr@k results obtained by GENIE3 TJM, BDA, JGSA and BioSfer on the reduced dataset

	GENIE3	TJM	BDA	JGSA	BioSfer
AUROC	0.480	0.546	0.549	0.505	**0.721**
AUPR	0.490	0.460	0.574	0.303	**0.719**
AUr@k	0.613	0.666	0.660	0.633	**0.809**

The best results are highlighted in boldface.

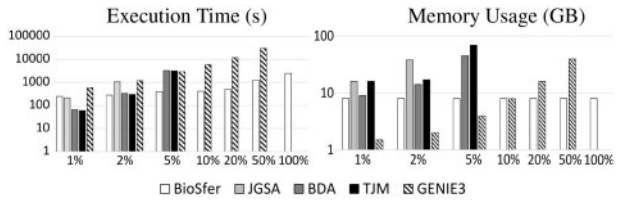

Fig. 9. Average running time (in seconds) and memory usage (in GB) required by the considered methods, on different samples of the dataset of increasing size

Moreover, **BioSfer** required a reasonable amount of RAM (~8GB) also to process the whole dataset, while the competitors led to saturating the 64GB available on the server, even to process a small sample of the dataset (see, e.g. JGSA on the sample containing 5% of the dataset). This result allows us to state that **BioSfer** not only provides the best results in terms of accuracy of the reconstruction, showing the positive effect of the proposed transfer learning approach, but it is also the most efficient method among those considered in the experiments, in terms of computational resources.

As anticipated in Section 4, we also performed the experiments for the opposite task, i.e. the reconstruction of the mouse GRN by exploiting the human as source. From the results of such an analysis (Supplementary File S5), it is possible to draw similar conclusions to those related to the reconstruction of the human GRN, confirming the advantage provided by the transfer learning approach as well as the superiority of BioSfer.

5 Biological analysis

In this section, we report the result of a qualitative analysis on 1375 top-ranked interactions (i.e. with a score $\geq$0.85) returned by BioSfer (Supplementary File S1, sheet 'Top-ranked gene-pairs'). The aim is 2-fold: to check the biological consistency of results and to prove the effectiveness of our tool in discovering new meaningful gene-functional interactions.

5.1 g:Profiler functional analysis

Top-ranked gene-pairs were analyzed using g:GOSt, an application of g:Profiler (Version e96_eg43_p13_563554d) (Reimand *et al.*, 2016) which can also perform a comparative functional analysis of multiple gene lists. The analysis was performed by defining two lists of genes, containing, respectively, the genes in the first position and the genes in the second position, of the selected gene pairs returned by BioSfer.

By observing the result of the enrichment analysis produced by g:GOSt (Manhattan-like plots in Supplementary File S2), it is evident that genes in the two lists show a very similar plotting structure with significant P-values. These results confirm the ability of BioSfer to associate genes that are functionally related, because they are in the same or related biological processes or pathways. Some examples are listed in the bottom of Supplementary File S2, while complete results are in Supplementary File S3.

5.2 Functional relationship of gene-pairs not associated by g:Profiler

The previous analysis showed that 1160 out of 1375 gene pairs returned by BioSfer were confirmed by g:GOSt. Subsequently,

we analyzed the genes belonging to the remaining 215 gene-pairs, that were associated with a score ranging from 0.85 to 0.95 by BioSfer (Supplementary File S1, sheet 'Gene-pairs not ass. by gGOSt'). These genes appeared to be enriched in different pathways or processes, but not together with the gene they were associated by BioSfer. We compared such pathways and processes with those in which the genes of the 1160 pairs confirmed by g:GOSt are enriched (Manhattan-like-plots in Supplementary File S6 and the detailed results in Supplementary File S7). The result shows that processes and pathways are the same for both the lists of genes and that the only difference is in terms of *P*-values, which is mainly due to the different amounts of genes analyzed. This means that the absence of confirmation by g:GOSt for those 215 gene pairs does not depend on the specific pathways and processes in which the genes are enriched, but rather on poor annotations for their statistical enrichment in the same category or process in g:GOSt. This is an additional evidence that BioSfer can associate genes that show meaningful relationships, even when gene functional annotations are missing or poor.

Among the gene pairs not identified by g:Profiler, the pair with the highest score (0.948) involves NBPF8P and ND4. According to an extensive search on the web (including PubMed), these two genes do not appear to be functionally associated. NBPF8P is a pseudogene belonging to the neuroblastoma breakpoint family (NBPF), which consists of dozens of recently duplicated genes, whose gene copy-number variations have been implicated in some developmental and neurogenetic diseases (GeneCards, GCID:GC01P120417). ND4 is a mitochondrial gene that encodes for the NADH dehydrogenase (Complex I), the core subunit of the mitochondrial respiratory chain, which is associated with several human diseases including, among others, encephalopathy. Since the correlation between the impaired mitochondrial respiratory chain function and the pathogenesis of several neurological diseases is well-known (Chaturvedi and Beal, 2013), the prediction provided by BioSfer is biologically reasonable.

Another interesting relationship (BioSfer score: 0.945) involves LINC00657 (aliases for NORAD Gene) and RPL39. LINC00657 is a long non-coding RNA (lncRNA) gene. A recent work provided evidence of a tissue-specific expression of this gene in vascular endothelial cells and of its involvement in the angiogenesis process as a miR-590-3p sponge (Bao *et al.*, 2018). RPL39 is a ribosomal protein that is necessary for the biogenesis of the ribosomal subunit 60S. The relationship between these two genes is not reported anywhere, but it is plausible because of the recent discovery of the existence of ribosome-associated ncRNAs (rancRNAs), that act as regulators of the translation of some RNA transcripts. Indeed, ribosomes can be the target for numerous small and long non-coding RNAs (ncRNAs) in various organisms (Pircher *et al.*, 2014). By targeting the ribosome, RNA molecules would allow a fast and direct regulation of protein production. Such a rapid response is essential, e.g. under sudden environmental changes or other types of stress conditions, since it allows the required massive reprograming of the gene expression pattern in a very short time (Lintner and Cate, 2014). Nevertheless, to the best of our knowledge, only a few reports have been published about the characterization of rancRNAs, which are actually related to studies about other organisms than *Homo sapiens* (Mleczko *et al.*, 2018).

Finally, a comparison with no_transfer supports the conclusion that the knowledge about the mouse network was successfully exploited by BioSfer to push relevant gene functional relationships up in the ranking. In particular, in the ranking provided by no_transfer, the NBPF8P-ND4 and LINC00657-RPL39 gene pairs appear, respectively, 827 and 943 positions below, compared with the ranking provided by BioSfer.

The prediction of these associations, like many others not discussed in this paper due to space limitations, reveals the ability of BioSfer to suggest potential functional relationships among genes, even when they are still not fully characterized. This is significantly important for the advance of knowledge in the biological and clinical research fields, especially in those cases, like that of ncRNAs, for which the functional and regulatory mechanisms are still largely unknown.

6 Conclusion

In this paper, we proposed the transfer learning method BioSfer, which is able to exploit the knowledge about a (reconstructed) source GRN to improve the reconstruction of a target regulatory network. BioSfer is natively able to work in the PU setting and to exploit a large amount of unlabeled examples. Experimental evaluation, performed by considering the human gene network as target and the mouse gene network as source, showed that BioSfer is able to fruitfully exploit the knowledge acquired from the source network. Moreover, we showed that it is able to outperform existing state-of-the-art transfer learning approaches and that the suggested interactions, which were not detected by other tools, appear biologically plausible.

As future work we will develop a distributed version of BioSfer, able to analyze very large networks in the Big Data context. Moreover, we plan to evaluate the possible benefits that could be obtained by injecting multiple source networks for the reconstruction of the target network. Finally, we will evaluate the possible adoption of BioSfer for the reconstruction of heterogeneous networks, where nodes, also of different types, interact each other according to different types of relationships.

Funding

We acknowledge the support of the EU Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944) and of the National Research Council (CNR) Flagship Project InterOmics.

Acknowledgements

We thank Lynn Rudd for her help in reading the manuscript.

Conflict of Interest: none declared.

References

Achanta,H.K. *et al.* (2016) A transfer learning approach for integrating biological data across platforms. In: *2016 American Control Conference (ACC)*, pp. 6695–6697.

Aibar,S. *et al.* (2017) Scenic: single-cell regulatory network inference and clustering. *Nat. Methods*, **14**, 1083–1086.

Ament,S.A. *et al.* (2018) Transcriptional regulatory networks underlying gene expression changes in huntington's disease. *Mol. Syst. Biol.*, **14**, e7435.

Bao,M.-H. *et al.* (2018) Long noncoding RNA LINC00657 acting as a miR-590-3p sponge to facilitate low concentration oxidized low-density lipoprotein-induced angiogenesis. *Mol. Pharmacol.*, **93**, 368–375.

Berger,M.F. and Bulyk,M.L. (2009) Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. *Nat. Protoc.*, **4**, 393–411.

Breckels,L.M. *et al.* (2016) Learning from heterogeneous data sources: an application in spatial proteomics. *PLoS Comput. Biol.*, **12**, e1004920–26.

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Ceci,M. *et al.* (2015) Semi-supervised multi-view learning for gene network reconstruction. *PLoS One*, **10**, e0144031.

Chang,C. *et al.* (2008) Fast network component analysis (fastnca) for gene regulatory network reconstruction from microarray data. *Bioinformatics*, **24**, 1349–1358.

Chaturvedi,R.K. and Beal,M.F. (2013) Mitochondrial diseases of the brain. *Free Radic. Biol. Med.*, **63**, 1–29.

Dai,W. *et al.* (2007) Boosting for transfer learning. In: *ICML 2007*, pp. 193–200.

Denas,O. *et al.* (2015) Genome-wide comparative analysis reveals human-mouse regulatory landscape and evolution. *BMC Genomics*, **16**, 87.

Elkan,C. and Noto,K. (2008) Learning classifiers from only positive and unlabeled data. In: *SIGKDD 2008*, pp. 213–220.

Hartemink,A.J. *et al.* (2002) Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics*, **18(Suppl. 1)**, S216–24.

He,J. *et al.* (2010) Naive Bayes classifier for positive unlabeled learning with uncertainty. In: *Proceedings of the SIAM International Conference on Data*

Mining, SDM 2010, April 29–May 1, 2010, Columbus, OH, USA, pp. 361–372.

Hou,M. *et al.* (2018) Generative adversarial positive-unlabelled learning. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, Stockholm, Sweden*. International Joint Conferences on Artificial Intelligence Organization, pp. 2255–2261.

Huynh-Thu,V.A. *et al.* (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS One*, **5**, e12776.

Singh,A.J. *et al.* (2018) Differential gene regulatory networks in development and disease. *Cell. Mol. Life Sci.*, **75**, 1013–1025.

Jinyin,C. *et al.* (2019) Multiview transfer learning for software defect prediction. *IEEE Access*, **7**, 8901–8916.

Levatic,J. *et al.* (2017) Self-training for multi-target regression with tree ensembles. *Knowl. Based Syst.*, **123**, 41–60.

Lintner,N.G. and Cate,J.H. (2014) Regulating the ribosome: a spotlight on RNA dark matter. *Mol. Cell*, **54**, 1–2.

Liu,B. *et al.* (2003) Building text classifiers using positive and unlabeled examples. In: Icdm 2003, pp. 179–186.

Liu,J. *et al.* (2014) Reconstruction of the gene regulatory network involved in the sonic hedgehog pathway with a potential role in early development of the mouse brain. *PLoS Comput. Biol.*, **10**, e1003884.

Long,M. *et al.* (2014) Transfer joint matching for unsupervised domain adaptation. In: *CVPR 2014*, pp. 1410–1417.

Lu,L. and Zhou,T. (2011) Link prediction in complex networks: a survey. *Phys. A: Stat. Mech. Appl.*, **390**,

Marbach,D. *et al.* (2012) Wisdom of crowds for robust gene network inference. *Nat. Methods*, **9**, 796–804.

Mignone,P. and Pio,G. (2018) Positive unlabeled link prediction via transfer learning for gene network reconstruction. In: *ISMIS 2018*, pp. 13–23.

Mleczko,A.M. *et al.* (2018) Transfer RNA-derived fragments target and regulate ribosome-associated aminoacyl-transfer RNA synthetases. *Biochim. Biophys. Acta Gene Regul. Mech.*, **1861**, 647–656.

Pan,S.J. *et al.* (2008) Transfer learning for wifi-based indoor localization. In: Workshop on Transfer Learning for Complex Task of AAAI, 2008.

Pan,S.J. *et al.* (2010) Cross-domain sentiment classification via spectral feature alignment. In: *Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC*, WWW'10, pp. 751–760. ACM.

Park,P.J. (2009) Chip-seq: advantages and challenges of a maturing technology. *Nat. Rev. Genet.*, **10**, 669–680.

Penfold,C. and Wild,D. (2011) How to infer gene networks from expression profiles, revisited. *Interface Focus*, **1**, 857–870.

Pio,G. *et al.* (2015) ComiRNet: a web-based system for the analysis of miRNA-gene regulatory networks. *BMC Bioinformatics*, **16**, S7.

Pircher,A. *et al.* (2014) Ribosome-associated ncRNAs: an emerging class of translation regulators. *RNA Biol.*, **11**, 1335–1339.

Platt,J.C. (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola,A.J. *et al.* (eds.) *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA, pp. 61–74.

Reimand,J. *et al.* (2016) g:Profiler—a web server for functional interpretation of gene lists (2016 update). *Nucleic Acids Res.*, **44**, W83–W89.

Rousseeuw,P.J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**, 53–65.

Cubitt,T.S. *et al.* (2012) Extracting dynamical equations from experimental data is np hard. *Phys. Rev. Lett.*, **108**, 120503.

Sevakula,R.K. *et al.* (2018) Transfer learning for molecular cancer classification using deep neural networks. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. doi: 10.1109/TCBB.2018.2822803.

Stark,C. *et al.* (2006) Biogrid: a general repository for interaction datasets. *Nucleic Acids Res.*, **34**, D535–539.

Streit,A. *et al.* (2013) Experimental approaches for gene regulatory network construction: the chick as a model system. *Genesis (New York, N.Y.: 2000)*, **51**, 296.

Todorovski,L. (2017) Equation discovery. In: Sammut,C. and Webb,G.I. (eds) *Encyclopedia of Machine Learning and Data Mining*. Springer, Berlin, pp. 410–414.

Wang,J. *et al.* (2017) Balanced distribution adaptation for transfer learning. In: *ICDM 2017*, pp. 1129–1134.

Warwick Vesztrocy,A. *et al.* (2018) Prioritising candidate genes causing QTL using hierarchical orthologous groups. *Bioinformatics*, **34**, i612–i619.

Weiss,K.R. *et al.* (2016) A survey of transfer learning. *J. Big Data*, **3**, 9.

Xiao,Y. *et al.* (2015) A robust one-class transfer learning method with uncertain data. *Knowl. Inf. Syst.*, **44**, 407–438.

Yang,X. *et al.* (2007) A weighted support vector machine for data classification. *IJPRAI*, **21**, 961–976.

Zhang,B. and Zuo,W. (2009) Reliable negative extracting based on knn for learning from positive and unlabeled examples. *J Comp.*, **4**, 94–101.

Zhang,J. *et al.* (2017) Joint geometrical and statistical alignment for visual domain adaptation. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, January 2017, pp. 5150–5158.