Special Issue Paper

International Journal of HIGH PERFORMANCE COMPUTING APPLICATIONS

The International Journal of High Performance Computing Applications
2018, Vol. 32(3) 380–392

DOI: 10.1177/1094342016662508
journals.sagepub.com/home/hpc
SAGE

# NES²RA: Network expansion by stratified variable subsetting and ranking aggregation

**Francesco Asnicar[1], Luca Masera[1], Emanuela Coller[2], Caterina Gallo[1], Nadir Sella[1], Thomas Tolio[1], Paolo Morettin[1], Luca Erculiani[1], Francesca Galante[1], Stanislau Semeniuta[1], Giulia Malacarne[2], Kristof Engelen[2], Andrea Argentini[3], Valter Cavecchia[4], Claudio Moser[2] and Enrico Blanzieri[1]**

**Abstract**

Gene network expansion is a task of the foremost importance in computational biology. Gene network expansion aims at finding new genes to expand a given known gene network. To this end, we developed gene@home, a BOINC-based project that finds candidate genes that expand known local gene networks using NESRA. In this paper, we present NES²RA, a novel approach that extends and improves NESRA by modeling, using a probability vector, the confidence of the presence of the genes belonging to the local gene network. NES²RA adopts intensive variable-subsetting strategies, enabled by the computational power provided by gene@home volunteers. In particular, we use the skeleton procedure of the PC-algorithm to discover candidate causal relationships within each subset of variables. Finally, we use state-of-the-art aggregators to combine the results into a single ranked candidate genes list. The resulting ranking guides the discovery of unknown relations between genes and a priori known local gene networks. Our experimental results show that NES²RA outperforms the PC-algorithm and its order-independent PC-stable version, ARACNE, and our previous approach, NESRA. In this paper we extensively discuss the computational aspects of the NES²RA approach and we also present and validate expansions performed on the model plant *Arabidopsis thaliana* and the model bacteria *Escherichia coli*.

**Keywords**

Volunteer computing, distributed computing, BOINC, bioinformatics, gene network expansion

## 1 Introduction

Biological processes are often regulated at the transcriptional level, via gene regulatory networks (GRNs) (Hasty et al., 2001) comprising regulatory genes, known as transcription factors, and regulated genes. So far, in most cases, only a small fraction of the genes involved in a GRN are known, and usually collected in local gene networks (LGNs) that are subsets of genes known to be causally connected. These genes are discovered through ad hoc experiments testing in vivo the hypothesis that a given gene participates in a specific biological process. Thus, there is an urgent need to fill this knowledge gap in order to have a better picture of most biological processes and translate biology into medical, biotechnological, and agricultural applications. A major contribution in this field has come from the new sequencing technologies, which dramatically increased the sequence output capacity and equally decreased the cost per sequenced base.[1]

Nowadays, we are witnessing an exponential increase of sequencing data and gene expression data in public databases. The collection and integration of these data sets has offered new opportunities and challenges to the field of computational biology. In particular, the analysis of the huge amount of available gene

[1]DISI, University of Trento, Italy
[2]CRI, Fondazione Edmund Mach, Italy
[3]Department of Biochemistry, Ghent University and Medical Biotechnology Center VIB, Belgium
[4]CNR-IMEM, Trento, Italy

**Corresponding author:**
Francesco Asnicar, DISI, University of Trento, Via Sommarive, 9 Povo Trento, TN 38123, Italy.
Email: f.asnicar@unitn.it

expression data can lead to the discovery of causal relationships between the genes of an organism and link them to a specific biological process. However, to date these causal relationships are not yet well known, even when considering the most studied model organisms. It is very common in biological research, when studying a particular process, to start taking into account the prior available knowledge such as the genes participating in that process. In this scenario, methods that can suggest new candidate genes, which are potentially playing a role within a given gene network, are of essential importance for biologists. In particular, the gene network expansion (GNE) task starts with a LGN of an organism and can be defined as: given a LGN, find other candidate genes that regulate or are regulated by genes belonging to the LGN.

The PC-algorithm (Spirtes and Glymour, 1991) discovers causal relationships among variables by systematically testing the conditional independence of two nodes given subsets of their adjacent nodes. The computational cost of the PC-algorithm is exponential in the number of nodes, but it behaves reasonably in the case of sparse, scale-free networks (Maathuis et al., 2010), like biological networks (Barabási, 2003). The PC-algorithm has been comprehensively presented and evaluated by Kalisch and Bühlmann (2007) and applied to gene network reconstruction (Maathuis et al., 2010). The PC-algorithm has also been successfully employed in other network inference approaches (Tan et al., 2008, 2011; Wang et al., 2010; Zhang et al., 2012). The results of the PC-algorithm depend on the order of the nodes in the input file; the order-independent version is called PC-stable (Colombo and Maathuis, 2012).

At the time of writing, other popular methods for network inference (NI) are the Bayesian Network Inference with Java Objects (BANJO (Hartemink, 2005)), network inference by reverse-engineering (NIR (Gardner et al., 2003)), and the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE (Margolin et al., 2006a,b)). The last one has been empirically proved to be the state of the art NI method (Allen et al., 2012). The available reconstruction methods applied to genome wide data are computationally demanding due to the huge size of the solution space (Kalisch and Bühlmann, 2007). Moreover, as we will see here, these methods are not accurate enough to use the results to perform a network expansion (Marbach et al., 2012).

In this paper, we explicitly define the task of finding candidates for gene network expansion. Then we propose a method called Network Expansion by Stratified Subsetting and Ranking Aggregation (NES$^2$RA ) to solve it. NES$^2$RA generalizes our previous proposal NESRA (Asnicar et al., 2015a) with the main difference being that it is now possible to model with a probability the presence of the genes of the network to be expanded in the subsets, namely the sampling is stratified. Both NESRA and NES$^2$RA are based on the PC-algorithm that we run on our gene@home project (Asnicar et al., 2015b), developed on the Berkeley Open Infrastructure for Network Computing (BOINC) platform (Anderson, 2004). We evaluate NES$^2$RA on real data on model organisms (*Arabidopsis thaliana* and *Escherichia coli*), and compare it against NESRA and ARACNE.

The paper is organized as follows. Section 2 introduces and defines the task accomplished by NES$^2$RA . Section 3 presents the NES$^2$RA algorithm. Section 4 details the development of NES$^2$RA exploiting the gene@home BOINC project, based on volunteer distributed computing. Section 5 presents an extensive evaluation of NES$^2$RA performed on two different data sets. Finally, Section 6 draws some conclusions providing future insights for the gene@home project and the proposed methods.

## 2 Gene network expansion

In this paper, we consider the task of discovering candidate genes for the gene network expansion. We report here the definition of the task, as already introduced by Asnicar et al. (2015a). Given a set $\mathcal{S}$ of gene transcripts whose level of expression has been measured $p$ times in different conditions, such that for each $s_i \in \mathcal{S}$ there is a vector $x_i \in \mathbb{R}^p$ of expression levels. Let us assume that there exists a generally unknown ground-truth direct graph $\mathcal{G} = (\mathcal{S}, \mathcal{B})$ with $\mathcal{B} \subseteq \mathcal{S} \times \mathcal{S}$ which represents the real causal relationships between the gene transcripts. The discovery of candidate genes for GNE is thus defined as follows.

**Definition 1.** *Given a graph $G = (N, B)$ where $N \subseteq \mathcal{S}$ and $B \subseteq N \times N$, find a ranked list of elements of $\mathcal{S} \backslash N$ such that the elements of the list are connected or very near to the elements of $N$ in $\mathcal{G}$.*

The choice of facing the gene network expansion task is motivated by the fact that the biological research is often guided by incomplete prior knowledge about the relevance of some genes, in particular biological processes. Considering the current validation methods that involve a complex mix of analytical and wet-lab techniques, the ability to provide a high-quality list of candidate genes is of essential importance for a biologist. The NES$^2$RA approach is particularly suitable in this context, since it introduces the possibility of modeling the confidence of biologists about each gene belonging to the LGN.

NI methods can be used to solve the task of finding candidates for GNE. Indeed, a perfect solution for the NI task would also perfectly solve the GNE task and consequently the task of finding candidate genes. However, the considered NI methods are far from

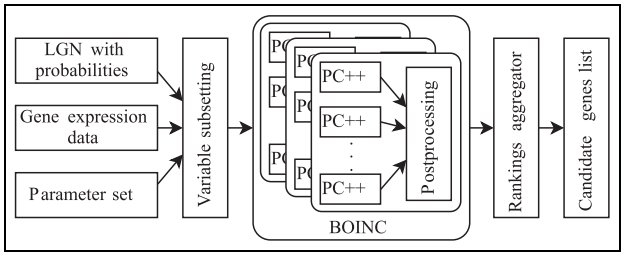


**Figure 1.** NES$^2$RA workflow.

perfect and computationally very demanding due to the enormous size of the solution space. For instance, in the PC-algorithm the solution space is super-exponential in the number of nodes (Kalisch and Buhlmann, 2007). Even the task of finding candidates for gene network expansion is computationally demanding. Indeed, there exists $|\mathcal{S}\backslash N|!$ possible ranked expansion lists, where $\mathcal{S}$ usually contains thousands of genes.

## 3 NES$^2$RA

NES$^2$RA is an improved and generalized version of NESRA (Asnicar et al., 2015a), which considers as input data the LGN and the probability of each gene of the LGN to be included in the subsets, the set of parameters to be used, and the gene expression levels for the considered organisms. The inclusion of the LGN in the subsetting step improves the quality of the results (as we will see in Section 5), because the composition of each subset is influenced by the LGN nodes added. The vector of probabilities is a representation of the knowledge of the user (e.g. a biologist) about the presence of specific genes in the network. Probability 1 means that the gene is definitely in the network, whereas probability 0 means that there is no knowledge about the presence of the gene in the network. Depending on the probabilities, the genes will be included in the data for the run of the PC-algorithm. If all the probabilities are zero NES$^2$RA coincides with its previous version NESRA (Asnicar et al., 2015a). The high-level structure of NES$^2$RA is described in Figure 1 and Algorithm 1.

The ranking procedure (RP) presented in Algorithm 2 is composed of three main steps, which respectively: create the subsets; execute several calls of the skeleton procedure of the PC-algorithm (Algorithm 3); and compute the transcripts frequency that defines the order of each ranking.

The RP takes as parameters the number of iterations $i$, the dimension of the subset $d$, the significance level $\alpha$ for the skeleton, and the probability vector $\Pi$ for the genes of the LGN. The output of the skeleton depends on the order of the inputs. Hence iterating $i$ times its application mitigates this effect, reaching a more stable solution. The RP returns a ranked list of $k$ elements

**Algorithm 1.** Pseudo-code of NES$^2$RA.

**Data:** $\mathcal{S}$ set of candidate transcripts, $\mathcal{S}_{\text{LGN}}$ set of LGN transcripts, $E$ expression data, a vector $\Pi = (\pi_1, \ldots, \pi_l, \ldots, \pi_{|\mathcal{S}_{\text{LGN}}|})$ of the probabilities of each $g_l \in S_{\text{LGN}}$ to be in the LGN.
**Input:** $I$ set of number of iterations, $D$ set of the subset dimensions, $A$ set of the significance levels $\alpha$, $k$ maximum length of the candidate gene lists
**Result:** ordered list of candidate transcripts
$L \leftarrow \emptyset$ //$L$ set of ordered lists
**foreach** $\alpha \in A$ **do**
  **foreach** $d \in D$ **do**
    **foreach** $i \in I$ **do**
      $L \leftarrow L \cup$ RP($\mathcal{S}$, $\mathcal{S}_{\text{LGN}}$, $E$, $\Pi$, $i$, $d$, $\alpha$) //call Algorithm 2
$L \leftarrow$ top($L$, $k$) //cut each list in $L$ to the first $k$ elements
**return** Ranking_aggregation(L)

**Algorithm 2.** NES$^2$RA ranking procedure (RP).

**Data:** $\mathcal{S}$ set of candidate transcripts, $S_{\text{LGN}}$ set of LGN transcripts, $E$ expression data, $\Pi = (\pi_1, \ldots, \pi_l, \ldots, \pi_{|\mathcal{S}_{\text{LGN}}|})$ probability vector for each $g_l \in \mathcal{S}_{\text{LGN}}$
**Input:** $i \geq 1$ number of iterations, $d$ subset dimension, $\alpha$ significance level
**Result:** $l$, ordered list of candidate transcripts
**foreach** $g \in S$ **do**
  $p_g \leftarrow 0$, $f_g \leftarrow 0$
/*Step 1: Subsets creation */
**foreach** $j$, $1 \leqslant j \leqslant i$ **do**
  $h \leftarrow 1$, $\mathcal{S}_{\text{temp}} \leftarrow \mathcal{S}\backslash\mathcal{S}_{\text{LGN}}$
  **while** $\mathcal{S}_{\text{temp}} \neq \emptyset$ **do**
    **foreach** $g_l \in \mathcal{S}_{\text{LGN}}$ **do**
      with probability $\pi_l$: $T_{h,j} \leftarrow T_{h,j} \cup \{g_l\}$
    $S_{\text{temp}} \leftarrow S_{\text{temp}} \cup (S_{\text{LGN}}\backslash T_{h,j})$
    **while** $|T_{h,j}| < d$ **do**
      uniformly random select $g \in \mathcal{S}_{\text{temp}}$
      $T_{h,j} \leftarrow T_{h,j} \cup \{g\}$
      $S_{\text{temp}} \leftarrow S_{\text{temp}}\backslash\{g\}$
      $p_g \leftarrow p_g + 1$
    **if** $\mathcal{S}_{\text{temp}} = \emptyset$ *and* $|T_{h,j}| < d$ **then**
      **while** $|T_{h,j}| < t$ **do**
        uniformly random select $g \in S\backslash T_{h,j}$
        $T_{h,j} \leftarrow T_{h,j} \cup \{g\}$
        $p_g \leftarrow p_g + 1$
    $h \leftarrow h + 1$
  $N_j \leftarrow h$
/*Step 2: skeleton */
**foreach** $j$, $1 \leqslant j \leqslant i$ **do**
  **foreach** $h$, $1 \leqslant h \leqslant N_j$ **do**
    $R_{h,j} \leftarrow$ *skeleton*($T_{h,j}$, $E$, $\alpha$) //Algorithm 3 Asnicar et al. (2015a)
/*Step 3: Frequency computations */
**foreach** $g \in \mathcal{S}$ **do**
  **foreach** $q \in \mathcal{S}_{\text{LGN}}$ **do**
    **foreach** $j$, $1 \leqslant j \leqslant i$ **do**
      **foreach** $h$, $1 \leqslant h \leqslant N_j$ **do**
        **if** $g \in Adj_{R_{h,j}}(q)$ **then**
          $l \leftarrow l \cup \{g\}$ //adjacent nodes of $q$ in $R_{h,j}$
          $f_g \leftarrow f_g + 1$
  $f'_g \leftarrow f_g/(p_g * |\mathcal{S}_{\text{LGN}}|)$ //Normalized frequency
**return** $l$ ordered w.r.t. $f'_g$

that is partially computed on the gene@home BOINC project, while the frequencies calculation and the rankings aggregation are executed off-line. The novelty of NES$^2$RA is in step 1 of the ranking procedure (Algorithm 2), where we take into consideration the knowledge of the LGN with its associated probabilities.

NES$^2$RA systematically and iteratively applies subsetting on the whole data set in order to randomly select genes that will be processed with the skeleton procedure. The subsetting is controlled by the iterations $i$ and subset size $d$ parameters. In NES$^2$RA the subsetting is stratified, and the genes of the LGN can have an increased probability of being in the subsets. In fact, for a given pair of subset size $d$ and iteration $i$, a first selection, controlled by the probability vector $\Pi$, specifies which genes of the LGN are present in the subsets. The genes of the LGN that are not selected in the first selection are considered, together with the other candidate genes, for a second selection with uniform probability. Finally, a third selection restricted to the genes not already present in the current subset, permits completion of the last subset whenever it is of the desired dimension $d$. The overall effect of the vector $\Pi$ (analyzed in Appendix A) in the algorithm is such that its $l$th component $\pi_l$ modulates the probability of the presence of the $l$th gene ($g_l$) in the subsets. When $\pi_l = 1$ then the gene $g_l$ is present in all the subsets. In the case where $\pi_l = 0$, the probability is the same of the other genes. For each pair $d, i$, NES$^2$RA executes a number of skeleton procedures, given by equation (2) (Appendix A). The results of these executions are combined in a single list of genes, ranked by their number of appearances. The skeleton procedure produces a graph, providing the causal relationships between nodes, but not their directions. The PC-algorithm estimates the orientation of the edges after the skeleton procedure, and the orientation steps do not remove or add any edge. The execution of the skeleton procedure indeed produces the most important information that we want to exploit in NES$^2$RA : the existence of an edge between two nodes. Such an edge, in fact, represents the existence of a causal relationship between the two nodes, even though we do not know its direction.

Finally, NES$^2$RA produces the list of candidate genes by applying different ranking aggregation methods on the ranked lists. These methods comprise a base technique, i.e. the *number of appearances*, and more sophisticated methods, namely Borda Count (Borda, 1781) and MC4 heuristic (Lin, 2010). The method we considered as baseline is the *number of appearances*, which counts how many rankings a certain gene has, i.e. the more a gene is present, the higher its position in the aggregated rank is. The Borda Count method consists of constructing a matrix $A(m, n)$ with $m$ rows and $n$ columns, corresponding to the genes and the rankings, respectively. The element $a_{ij}$ is the rank of gene $i$ on ranking $j$, and a statistic for each gene is computed on the rows of the matrix. The two statistics that we used are the mean (BC-mean) and the minimum (BC-min) of the elements. The MC4 heuristic is an aggregator based on Markov chains. It consists of computing the transition matrix of the pairwise comparison of all the rankings for each gene. A step in the Markov chain assigns a higher probability to a gene $q$ if $\text{rank}(q) < \text{rank}(p)$ for a majority of the lists that ranked both $p$ and $q$ (Dwork et al., 2001). The steady state of the chain assigns higher probability to the genes with higher ranks. To avoid a non-unique stationary distribution, MC4 has as a parameter the significance level $\alpha_{\text{MC4}}$, for which we considered two values: 0.05 and 0.01.

Both NESRA and NES$^2$RA exploit the gene@home project for computing the first two steps of ranking procedure.

## 4 NES$^2$RA on the gene@home BOINC project

Nowadays, the literature reports several successful research projects that exploit the power of volunteer grid computing in order to achieve their goals (Anderson et al., 2002; Das et al., 2007). BOINC, for instance, is an open-source framework particularly convenient for projects that require a large amount of computation, but do not have access to suitable resources. NES$^2$RA requires $O_{\text{PC}}$ executions (see equation (2) in Appendix A) that can be easily parallelized. Therefore, we decided to exploit the gene@home (Asnicar et al., 2015b) BOINC project, hosted by the TN-Grid platform,[2] to distribute the computation of the skeleton procedure (Step 2 of Algorithm 2).

Every BOINC project is composed of several components, such as the work generator and the validator. The aim of the former is to create the workunits that will be then distributed to the volunteers, while the latter validates the results of the finished workunits. The validator performs a bitwise comparison of two workunits that have been computed by two different machines. This step is required to ensure the consistency of the results. We designed and developed our custom work generator using the Python language and two C++ wrappers to interface the work generator and the Python scripts with the BOINC framework. The subsets creation (step 1 of Algorithm 2) is implemented in the work generator. Each workunit corresponds to many runs of the skeleton procedure (step 2 of Algorithm 2), and the duration is estimated in order to inform the volunteers about execution times, a fundamental aspect in any BOINC project.

The core of gene@home is the client application. To date, it is available for both 32 and 64 bit architectures, for three operating systems: Linux, Windows, and Mac

**Algorithm 3.** Skeleton procedure of the PC-algorithm (Kalisch and Bühlmann, 2007).

**Data:** T, Set of transcripts, *E* expression data
**Input:** Significance level $\alpha$
**Result:** An undirected graph with causal relationship between transcripts
Graph $G \leftarrow$ complete undirected graph with nodes in T
$l \leftarrow -1$
**while** $l < |G|$ **do**
  $l \leftarrow l + 1$
  **foreach** $\exists u, v \in G$ s.t. $|Adj_G(u) \backslash \{v\}| \geq l$ **do** //$Adj_G(u)$ adjacent nodes of $u$ in $G$
    **if** $v \in Adj_G(u)$ **then**
      **foreach** $\mathbf{k} \subseteq Adj_G(u) \backslash \{v\}$ s.t. $|\mathbf{k}| = l$ **do**
        **if** *u, v are conditionally independent given* **k** *w.r.t. E with significance level* $\alpha$ **then**
          remove edge $\{u, v\}$ from $G$
**return** G

**Algorithm 4.** Partial correlations function.

**Input:** *i, j* analyzed variables, **k** separation set
**Result:** $\rho_{i,j|k}$
$l \leftarrow |\mathbf{k}|$; $M \leftarrow [i, j, k_1, \ldots, k_l]$; $d \leftarrow |M|$
Initialize the $d \times d$ matrix $\rho$ s.t. $\rho[u][v] \leftarrow correlation(M[u], M[v])$
**for** $n = 1$ **to** $l$ **do**
  **for** $u = 0$ **to** $l - n$ **do**
    **for** $v = u + 1$ **to** $d - n$ **do**
      $\rho[u][v] \leftarrow \frac{\rho[u][v] - \rho[u][d-n] \cdot \rho[v][d-n]}{\sqrt{(1-\rho^2[u][d-n])(1-\rho^2[v][d-n])}}$
**return** $\rho[0][1]$

OS X. The original implementation of the skeleton procedure, present in the pcalg R package (Hauser and Bühlmann, 2012; Kalisch et al., 2012), is not suitable for high-performance volunteer-computing projects due to both its software requirements, i.e. the R interpreter and numerous R packages, and its low speed and high memory consumption. To the best of our knowledge, no alternative open-source implementation of the skeleton procedure is available, thus we implemented our C++ version, namely PC++ .[3] The PC++ implementation makes use of efficient data structures and avoids the storing of the separation sets, which are not needed in NES$^2$RA, to reduce the memory usage. Moreover, the original recursive computation of the partial correlation (Proposition 2 (Kalisch and Bühlmann, 2007)) between the $i$th and the $j$th node given the separation set $k$ has been replaced with an iterative version based on a dynamic programming technique (Cormen et al., 2001), shown in Algorithm 4. This solution reduces the complexity of the computation from $O(3^l)$ to $O(l^3)$, where $l$ corresponds to the size of the separation sets. These optimizations and the possibility to natively integrate PC++ into the BOINC client application drastically decreased the computational time and memory usage.

In Table 1 we report the detailed comparison between the PC++ and the skeleton procedure of the pcalg package, conducted on the *E. coli* data set. From the results in Table 1 we can appreciate that the PC++ implementation gained a speed-up of more than 200 and decreased the memory usage by an order of magnitude.[4] For a fair comparison, we modified the original skeleton procedure to avoid storing the separation sets. No data is available for the skeleton on subset size $d = 4065$, because it reached the two weeks time limit we imposed. We estimated the execution time of skeleton for $d = 4065$ via a regression analysis on the other subset sizes, in more than 200 days.

The post-processing phase in NES$^2$RA consists of a two-step pipeline. The workunits results are firstly combined on the volunteers' local machines by the client applications, in order to reduce the size of the data to be uploaded. Lastly, the partially aggregated results are aggregated on the server.

## 5 Evaluation

In this section we report the results of three different experiments that assess the performance of NES$^2$RA . The aim of the first experiment is to biologically evaluate the results of NES$^2$RA in comparison with NESRA, ARACNE, and the PC-algorithm. The second experiment has the goal of analyzing the impact of the probability vector $\Pi$ on the final expansion list. Finally, we analyze the computational aspects of NES$^2$RA executed by the gene@home project.

The data set considered in the first experiment is composed of gene expression hybridizations for the *A. thaliana* plant model organism, namely microarray expression values publicly available in the Plex database (Dash et al., 2012). The data set comprises 393 hybridization experiments of the GeneChip Arabidopsis ATH1 Genome Array that encompass 22,810 probe sets. The LGN that we used for *A. thaliana* is the Flower Organ Specification Gene Regulatory Network (FOS). The FOS gene network has been characterized and validated in vivo by the use of specific mutants (Espinosa-Soto et al., 2004), and is composed of 15 genes connected by 54 causal relationships

**Table 1.** Comparison between skeleton and PC++ in terms of running time and memory usage on the *E. coli* data set using different subset sizes.

| | | $d = 50$ | $d = 100$ | $d = 200$ | $d = 500$ | $d = 4065$ |
|---|---|---|---|---|---|---|
| skeleton | time (s) | 86.13 | 924.96 | 15470.62 | 169869.69 | timed out |
| | RAM (MB) | 95.23 | 104.83 | 145.88 | 200.11 | |
| PC ++ | time (s) | 0.36 | 3.82 | 69.59 | 716.88 | 94525.63 |
| | RAM (MB) | 5.85 | 7.85 | 13.73 | 35.51 | 506.75 |

**Table 2.** Candidate genes list of the FOS LGN of *A. thaliana* produced by NES$^2$RA .

| Rank | AffyID | Gene | Annotation | Class |
|---|---|---|---|---|
| 1 | 259089_at | AT3G04960 | Similar to unknown protein | Class 1 (Lee et al., 2005) |
| 2 | 248496_at | AT5G50790 | ATSWEET10 | Class 3 (Chen et al., 2012) |
| 3 | 265441_at | AT2G20870 | Cell wall protein precursor | Class 1 (Cai et al., 2007) |
| 4 | 255644_at | AT4G00870 | Basic helix-loop-helix (bHLH) family protein | Class 2 (Hu et al., 2003) |
| 5 | 261375_at | AT1G53160 | SPL4 (SQUAMOSA PROMOTER BINDING PROTEIN-LIKE 4) | Class 1 (Lal et al., 2011) |
| 6 | 249939_at | AT5G22430 | Similar to unknown protein | Class 1 (Zik and Irish, 2003) |
| 7 | 255448_at | AT4G02810 | FAF1 (FANTASTIC FOUR 1) | Class 1 (Wahl et al., 2010) |
| 8 | 245842_at | AT1G58430 | RXF26 | Class 1 (Shi et al., 2011) |
| 9 | 256259_at | AT3G12460 | DEDDy 3'-5' exonuclease domain-containing protein | Class 4 |
| 10 | 260355_at | AT1G69180 | CRC (CRABS CLAW) | Class 1 (Lee et al., 2005) |

**Table 3.** NES$^2$RA precisions using different aggregation methods.

| Method | k=5 | k=10 | k=20 | k=55 |
|---|---|---|---|---|
| NES$^2$RA *N* of appearances | 0.57 | 0.57 | 0.57 | 0.51 |
| NES$^2$RA BC-mean | 0.80 | 0.90 | 0.75 | 0.51 |
| NES$^2$RA BC-min | 0.80 | 0.88 | 0.80 | 0.51 |
| NES$^2$RA MC4 ($\alpha_{MC4} = 0.05$) | 0.80 | 0.90 | 0.75 | 0.51 |
| NES$^2$RA MC4 ($\alpha_{MC4} = 0.01$) | 0.80 | 0.90 | 0.75 | 0.51 |
| NESRA BC-mean (Asnicar et al., 2015a) | 0.90±0.098 | 0.65±0.049 | 0.63±0.038 | 0.43±0.016 |
| ARACNE (Asnicar et al., 2015a) | 0.20 | 0.30 | 0.35 | 0.45 |

(Sanchez-Corrales et al., 2010). In this case the presence of the genes in the network is certain and so the vector $\Pi$ of NES$^2$RA has all its components set to 1.

The second experiment has been conducted on the bacterial model organism *E. coli*. The data set contains 4065 genes for 2470 hybridizations and it is publicly available in the COLOMBOS (Meysman et al., 2014) database. The LGN considered is a transcription factor network called gadW collected from the EcoCyc (Keseler et al., 2013) database. The gadW LGN is composed of 13 nodes connected by 12 edges, and it is involved in the acid resistance system of *E. coli*. In this experiment we compared the results of NES$^2$RA using two probability vectors: $\Pi_H$ and $\Pi_L$. The former has just the probability of the hub node, the gadW gene is set to 1 while the probabilities of all other genes are set to 0. The latter has all the entries set to 1. The same experimental setup has been used to assess the computational aspects of NES$^2$RA .

We assessed the biological validity of the results by performing a bibliographic research, classifying genes in four different classes, as follows.

***Class 1*** collects genes reported to be biologically or functionally related to the genes in the LGN.
***Class 2*** contains genes not reported to be directly related with the input network, but reported to be related to genes of Class 1.
***Class 3*** comprises all the genes described in the literature that were reported not to be related with the input network or with the genes of Class 1.
***Class 4*** are genes for which no description was found in the available literature.

When we found a gene belonging to Class 1 or Class 2 we considered it to be a true positive, while a gene falling in Class 3 or Class 4 was considered a false positive. The precision of the genes in the candidate output list is

the ratio between the number of true positives and the sum of true positives and false positives. Other measures, like F1 and Recall, can not be computed on real organisms' data sets because no complete ground truth is available. We can only exploit the manually curated classification that we have performed for the resulting genes provided by the methods considered.

Table 2 reports an example of the candidate genes list for the FOS LGN of *A. thaliana*, produced by NES$^2$RA . The list has a precision value of 80% and it has been obtained by aggregating 60 different ranked lists using the MC4 method with the parameter $\alpha_{\text{MC4}} = 0.01$. The values considered for this run are: $I = \{100, 250, 500, 1000, 1500, 2000\}$, $D = \{50, 100, 250, 500, 750, 1000, 1250, 1500, 1750, 2000\}$, and $A = \{0.05\}$. The gene AT3G12460, ranked in position 9, is considered as a false positive. However, only a biological wet-lab validation could rule out if it is actually involved in the FOS LGN.

Table 3 shows the precision values of NES$^2$RA using the same set of experiments presented in Table 4 by Asnicar et al. (2015a). Using the very same set of parameters for NES$^2$RA we aggregated only the six rankings that have the same values as in NESRA (Asnicar et al., 2015a): iterations $I = \{100, 500, 2000\}$, subset dimensions $D = \{1000, 2000\}$ and $A = \{0.05\}$. It is possible to see that the best performances are obtained with list lengths of $k = 5$, 10, and 20. In particular, if we compare these results of NES$^2$RA with the results of NESRA (Table 3) we can see that NES$^2$RA has better precision when considering longer lists ($k = 55$). Moreover, NES$^2$RA proves to have better precision when compared with ARACNE. NES$^2$RA can be also compared with the PC-algorithm and the PC-stable using their precision values reported by Asnicar et al. (2015a), which are $0.39 \pm 0.03$ and 0.43, respectively. It can be seen that NES$^2$RA outperforms both the PC-algorithm and the PC-stable in the task of finding candidates for GNE.

Although the PC-algorithm, PC-stable, and ARACNE are used for a NI task, here we compared them to a GNE approach. In order to do such a comparison, we obtained a pseudo-expansion list by running each method on the whole data set of *A. thaliana*. Their results were then filtered with respect to the FOS LGN, selecting only the edges connected with at least a node in the LGN. Then, when possible, the results were sorted according to the *p*-value provided by the methods.

Table 4 reports the results obtained with different probability vectors to expand the gadW LGN of *E. coli* with parameters $A = \{0.01, 0.05\}$, $D = \{100, 200\}$, and $I = \{80, 100, 500, 1000, 1500, 2000\}$. Here NESRA can also be interpreted as a special case of NES$^2$RA where the probability vector $\Pi$ is set to 0 for each gene in the LGN. Additionally, we analyzed the results of

**Table 4.** Ranked lists of the gadW LGN of *E. coli* showing true positives in bold, false positives in italics, and their references.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| NESRA | **hdeD** Masuda and Church (2003) | **ybaS** Lu et al. (2013) | **yhiM** Tucker et al. (2002) | **ybaT** Tucker et al. (2002) | **cfa** Tucker et al. (2002) | **cbpM** Tucker et al. (2002) | *aidB* Volkert and Nguyen (1984) | *kch* Milkman (1994) | cbpA Tucker et al. (2002) | *yjbQ* Kim et al. (2010) |
| NES$^2$RA $\Pi_H$ | **hdeD** Masuda and Church (2003) | **yhiM** Tucker et al. (2002) | **ybaS** Lu et al. (2013) | **ybaT** Tucker et al. (2002) | **cfa** Tucker et al. (2002) | **cbpA** Tucker et al. (2002) | *kch* Milkman 1994) | **ycaC** Tucker et al. (2002) | **cbpM**, *cueR* Tucker et al. (2002), Stoyanov et al. (2001) | – |
| NES$^2$RA $\Pi_L$ | **hdeD** Masuda and Church (2003) | **yhiM** Tucker et al. (2002) | **cbpA** Tucker et al. (2002) | **ybaT** Tucker et al. (2002) | **appC** Hayes et al. (2006) | **ybaS** Lu et al. (2013) | *aidB* Volkert and Nguyen (1984) | **hyaF** Hayes et al. (2006) | **hyaA** Hayes et al. (2006) | **hyaC** Hayes et al. (2006) |
| ARACNE | **hdeD** Masuda and Church (2003) | **ybaS** Lu et al. (2013) | *dps* Dukan and Touati (1996), Gundlach and Winter (2014) | *aidB* Volkert and Nguyen (1984) | *sra* Akira (1986), Izutsu et al. (2001) | **cbpA** Tucker et al. (2002) | **ybaT** Tucker et al. (2002) | *elaB* Yoshida et al. (2012) | *yegP* Kumar et al. (2016) | *talA* Weber et al. (2006) |

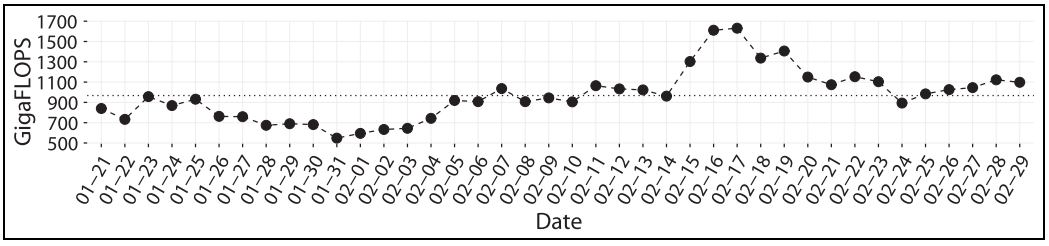


**Figure 2.** gene@home GigaFLOPS in the timespan of 40 days. The dotted line shows the average.

**Table 5.** Cumulative BOINC statistics for the *E. coli* experiments.

| | #RP runs | Workunits | Hosts | Hosts GigaFLOPS | Tot. PetaFLOP |
|---|---|---|---|---|---|
| NESRA | 24 | 6424 | 141 | 2.76±0.65 | 177.59 |
| NES$^2$RA $\Pi_H$ | 24 | 6528 | 138 | 2.98±0.88 | 173.57 |
| NES$^2$RA $\Pi_L$ | 24 | 7150 | 151 | 2.85±0.68 | 156.80 |

**Table 6.** Statistics of the workunits computational costs on the *E. coli* data set.

| | GigaFLOP per workunit | | | |
|---|---|---|---|---|
| | Min | Max | Avg ± SD | RSD |
| NESRA | 0.04 | 186.65 | 13.67 ± 19.03 | 1.39 |
| NES$^2$RA $\Pi_H$ | 1.00 | 134.57 | 13.21 ± 16.90 | 1.28 |
| NES$^2$RA $\Pi_L$ | 0.04 | 228.68 | 10.86 ± 15.17 | 1.40 |

ARACNE on the same data set for the same LGN, using the same analysis applied by Asnicar et al. (2015a). Interestingly, both NESRA and NES$^2$RA produce a higher quality expansion list, in particular, when considering only the first five genes in the output lists. We manually curated the classification of the genes found, and report in bold the genes are classified as either Class 1 or Class 2, and in italics the ones belonging to either Class 3 or Class 4. It can be noticed that the injection of prior knowledge in the form of presence probability in the subsets, positively impacts the final quality of the expansion list. Indeed, as we can see from Table 4, the more prior knowledge is used, the more precise the expansion lists are, reaching up to 90% precision.

Figure 2 shows the trend of the computational power expressed by the gene@home project in a time span of 40 days. BOINC is a volunteer-based distributed computing system, and the statistics are computed on the basis of the daily credits generated by the system and assigned to the volunteers.[5] Despite no guarantee of a continuous influx of computational power, we can notice that the overall throughput of the system is always over the 500 GigaFLOPS with an average of 967.41 GigaFLOPS, shown with a dotted line in Figure 2. This is the result of at least 80 active users providing more than 400 active hosts.[6]

Tables 5 to 7 report the BOINC statistics regarding the experiments conducted on the *E. coli* data set. The FLOPs for the workunits have been computed on the basis of the computation time and the FLOPS of the hosts machines, determined by BOINC running a Whetstone benchmark (Curnow and Wichmann, 1976). In the gene@home project, each workunit is computed twice in order to be able to validate the results, as explained in Section 4. Thus, the actual FLOPs required for a NES$^2$RA experiment would be half of the ones reported in Table 5. By comparing the average throughput of gene@home and the FLOPs required for executing NES$^2$RA we see that, by exploiting the volunteer computational power, we could execute a NES$^2$RA experiment in about 2.5 days. The real execution time, however, may vary depending on several factors, such as the number of different experiments running at the same time on gene@home. Moreover, the double validation required by the gene@home project can increase the completion time of an experiment. Table 6 reports a summary of the workunit computational costs for the experiments conducted on *E. coli*. Table 7 presents the details of the computational effort requested by a run

**Table 7.** Detailed BOINC statistics for NES$^2$RA $\Pi_L$ on the *E. coli* data set.

| $\alpha$ | $d$ | $i$ | Workunits | GigaFLOP per workunit | | | | | GigaFLOP per PC++ |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Min | Max | Avg ± SD | RSD | Sum | |
| 0.01 | 100 | 80 | 38 | 0.66 | 3.39 | 2.52 ± 0.54 | 0.21 | 191.75 | 0.05 |
| | | 100 | 47 | 0.68 | 5.28 | 2.65 ± 0.63 | 0.24 | 248.78 | 0.05 |
| | | 500 | 235 | 0.68 | 6.11 | 2.52 ± 0.60 | 0.24 | 1186.04 | 0.05 |
| | | 1000 | 470 | 0.66 | 8.41 | 2.64 ± 1.00 | 0.38 | 2493.82 | 0.05 |
| | | 1500 | 705 | 0.63 | 8.67 | 2.45 ± 0.95 | 0.39 | 3461.11 | 0.05 |
| | | 2000 | 940 | 0.04 | 28.07 | 2.32 ± 1.18 | 0.51 | 4492.96 | 0.05 |
| | 200 | 80 | 18 | 3.17 | 14.93 | 11.51 ± 2.54 | 0.22 | 414.29 | 0.24 |
| | | 100 | 22 | 6.13 | 18.32 | 12.05 ± 3.00 | 0.25 | 530.25 | 0.24 |
| | | 500 | 110 | 3.17 | 18.13 | 9.92 ± 2.82 | 0.28 | 2182.22 | 0.20 |
| | | 1000 | 220 | 3.11 | 24.72 | 10.38 ± 3.10 | 0.30 | 4567.99 | 0.21 |
| | | 1500 | 330 | 2.91 | 40.92 | 10.64 ± 6.14 | 0.58 | 7024.33 | 0.21 |
| | | 2000 | 440 | 1.18 | 40.47 | 9.96 ± 4.44 | 0.45 | 8771.42 | 0.20 |
| 0.05 | 100 | 80 | 38 | 1.50 | 11.06 | 5.65 ± 1.39 | 0.25 | 429.41 | 0.11 |
| | | 100 | 47 | 1.48 | 11.16 | 5.82 ± 1.46 | 0.25 | 547.30 | 0.12 |
| | | 500 | 235 | 1.49 | 10.94 | 4.64 ± 1.30 | 0.28 | 2180.19 | 0.09 |
| | | 1000 | 470 | 2.01 | 19.22 | 5.16 ± 1.96 | 0.38 | 4863.63 | 0.10 |
| | | 1500 | 705 | 1.40 | 13.00 | 4.77 ± 1.88 | 0.39 | 6719.52 | 0.10 |
| | | 2000 | 940 | 1.44 | 228.68 | 4.99 ± 5.37 | 1.08 | 9403.84 | 0.10 |
| | 200 | 80 | 18 | 14.49 | 115.36 | 46.83 ± 17.49 | 0.37 | 1686.03 | 0.96 |
| | | 100 | 22 | 23.18 | 56.14 | 40.58 ± 9.21 | 0.23 | 1785.45 | 0.81 |
| | | 500 | 110 | 12.21 | 88.43 | 38.44 ± 10.90 | 0.28 | 8571.81 | 0.78 |
| | | 1000 | 220 | 12.19 | 98.01 | 38.79 ± 13.43 | 0.35 | 17184.80 | 0.78 |
| | | 1500 | 330 | 12.02 | 93.27 | 41.20 ± 15.54 | 0.38 | 27358.36 | 0.83 |
| | | 2000 | 440 | 5.45 | 134.84 | 43.28 ± 14.84 | 0.34 | 40513.54 | 0.92 |

of NES$^2$RA, for each combination of the parameters $A$, $D$, and $I$. The number of the workunits is $\lceil O_{PC} \times \frac{i}{100} \rceil$ were $O_{PC}$ is computed with equation (2), where the second term is zero in this case. The computational effort requested for each run of the RP function varies within the same run, as it is apparent to consider the minimum and the maximum GigaFLOP values. It is also worth noting that the pair $(\alpha, d)$ determines the computational cost required by a single PC++ execution.

## 6 Conclusions

We presented NES$^2$RA, our novel approach for generating ranked candidate genes lists, which expands known LGNs starting from gene expression data. It exploits iterated variable subsetting and ranking aggregation, as our previous proposal NESRA (Asnicar et al., 2015a), allowing the user to integrate the available prior knowledge on the network that has to be expanded. This makes it possible to model the biologists' knowledge about the presence of certain genes in the LGN that is translated into a higher probability of presence of these genes in the variable subsets generated. The injection of such prior knowledge shows encouraging results. NES$^2$RA relies on the computational power provided by the gene@home BOINC project, hosted by the TN-Grid platform (Asnicar et al., 2015b). We exploit the gene@home project for extensive executions of the PC++ algorithm, while all the post-processing, ranking, and aggregation analyses are performed off-line. The parallel nature of our approach together with the efficient implementation of the PC-algorithm (namely, PC++ ), allow us to easily distribute the computational work using the gene@home project. We evaluated the performances of NES$^2$RA on the FOS LGN of the model plant *A. thaliana*. NES$^2$RA outperforms both ARACNE, which has been proven to be a state of the art NI method (Allen et al., 2012), and our previous proposal NESRA (Asnicar et al., 2015a). The runs on the gadW network of *E. coli* confirmed the good results and permitted the assessment of the computational load of our application. Considering the performances of NES$^2$RA, its ability to scale with respect to the size of the input data, and the quality of the results, we plan to perform an extensive evaluation using different types of data that encompass several organisms, which include the bacterial model organism *Escherichia coli* and the eukaryote organism *Vitis vinifera*.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

## Notes

1. Wetterstrand KA. DNA sequencing costs: Data from the NHGRI genome sequencing program (GSP) Available at: www.genome.gov/sequencingcosts (accessed 28 October 2015).
2. http://gene.disi.unitn.it/test/.
3. Publicly available at https://bitbucket.org/francesco-asnicar/pc-boinc.
4. The experiments were executed on an Intel® Core™ i5-4590 processor at 3.30 GHz, with 8 GB of RAM running a 64 bit Linux with the 3.19.0-32 kernel.
5. Data available at: http://boincstats.com/it/stats/150/project/detail/.
6. Data available at: http://boincstats.com/it/stats/150/project/detail/user and http://boincstats.com/it/stats/150/project/detail/host.

## References

Akira W (1986) Analysis of *Escherichia coli* ribosomal proteins by an improved two dimensional gel electrophoresis. I. Detection of four new proteins. *Journal of Biochemistry* 100(6): 1583–1594.

Allen JD, Xie Y, Chen M, et al. (2012) Comparing statistical methods for constructing large scale gene networks. *PLoS ONE* 7(1): 1–9.

Anderson DP (2004) BOINC: A system for public-resource computing and storage. In: *Proceedings of the 5th IEEE/ACM international workshop on grid computing*, GRID '04, Washington, DC, USA, 8 November 2004, pp.4–10. IEEE Computer Society.

Anderson DP, Cobb J, Korpela E, et al. (2002) SETI@home: An experiment in public-resource computing. *Communications of the ACM* 45(11): 56–61.

Asnicar F, Erculiani L, Galante F, et al. (2015a) Discovering candidates for gene network expansion by distributed volunteer computing. In: *ISPA-15, Pbio*, Helsinki, Finland, 20–22 August 2015. IEEE.

Asnicar F, Sella N, Masera L, et al. (2015b) TN-Grid and gene@home project: Volunteer computing for bioinformatics. In: Ivashko E (ed) *Second international conference BOINC-based high performance computing: Fundamental research and development (BOINC:FAST 2015)*, number 1502 in CEUR Workshop Proceedings, Petrozavodsk, Russia, pp.1–15. CEUR-WS. Available at: http://ceur-ws.org/Vol-1502/paper1.pdf.

Barabási AL (2003) *Linked: How everything is connected to everything and what it means for business, science and everyday life*. New York: Penguin.

Borda J. C. (1781) *Memoire sur les elections au Scrutin*. Historie de l'Academie Royale des Sciences, Paris.

Cai X, Ballif J, Endo S, et al. (2007) A putative CCAAT-binding transcription factor is a regulator of flowering timing in arabidopsis. *Plant Physiology* 145(1): 98–105.

Chen LQ, Qu XQ, Hou BH, et al. (2012) Sucrose efflux mediated by SWEET proteins as a key step for phloem transport. *Science* 335(6065): 207–211.

Colombo D and Maathuis MH (2012) Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research* 15(1): 3741–3782.

Cormen TH, Leiserson CE, Rivest RL, et al. (2001) *Introduction to Algorithms*, Volume 6. Cambridge: MIT Press.

Curnow HJ and Wichmann BA (1976) A synthetic benchmark. *The Computer Journal* 19(1): 43–49.

Das R, Qian B, Raman S, et al. (2007) Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home. *Proteins: Structure, Functions, Bioinformatics* 69(S8): 118–128.

Dash S, Van Hemert J, Hong L, et al. (2012) PLEXdb: Gene expression resources for plants and plant pathogens. *Nucleic Acids Research* 40(Database issue): D1194–1201.

Dukan S and Touati D (1996) Hypochlorous acid stress in *Escherichia coli*: Resistance, DNA damage, and comparison with hydrogen peroxide stress. *Journal of Bacteriology* 178(21): 6145–6150.

Dwork C, Kmar R, Naor M, et al. (2001) Rank aggregation methods for the web. In: *Proceedings of the 10th WWW conference*, Hong Kong, May 1–5 2001 publisher: ACM New York, NY pp.613–622.

Espinosa-Soto C, Padilla-Longoria P, Alvarez-Buylla ER, et al. (2004) A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *Plant Cell* 16(11): 2923–2939.

Gardner TS, Di Bernardo D, Lorenz D, et al. (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301(5629): 102–105.

Gundlach J and Winter J (2014) Evolution of *Escherichia coli* for maximum HOCl resistance through constitutive expression of the OxyR regulon. *Microbiology* 160(8): 1690–1704.

Hartemink AJ (2005) Reverse engineering gene regulatory networks. *Nature Biotechnology* 23(5): 554–555.

Hasty J, McMillen D, Isaacs F, et al. (2001) Computational studies of gene regulatory networks: in numero molecular biology. *Nature Reviews Genetics* 2(4): 268–279.

Hauser A and Bühlmann P (2012) Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research* 13: 2409–2464.

Hayes ET, Wilks JC, Sanfilippo P, et al. (2006) Oxygen limitation modulates pH regulation of catabolism and hydrogenases, multidrug transporters, and envelope composition in *Escherichia coli* K-12. *BMC Microbiology* 6(1): 1.

Hu W, Wang Y, Bowers C, et al. (2003) Isolation, sequence analysis, and expression studies of florally expressed cDNAs in *Arabidopsis*. *Plant Molecular Biology* 53(4): 545–563.

Izutsu K, Wada C, Komine Y, et al. (2001) *Escherichia coli* ribosome-associated protein SRA, whose copy number increases during stationary phase. *Journal of Bacteriology* 183(9): 2765–2773.

Kalisch M and Bühlmann P (2007) Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research* 8: 613–636.

Kalisch M, Mächler M, Colombo D, et al. (2012) Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software* 47(11): 1–26.

Keseler IM, Mackie A, Peralta-Gil M, et al. (2013) EcoCyc: Fusing model organism databases with systems biology. *Nucleic Acids Research* 41(Database issue): D605–612.

Kim J, Kershner JP, Novikov Y, et al. (2010) Three serendipitous pathways in *E. coli* can bypass a block in pyridoxal-5'-phosphate synthesis. *Molecular Systems Biology* 6(1): 436.

Kumar A, Belglazova N, Bundalovic-Torma C, et al. (2016) Conditional epistatic interaction maps reveal global functional rewiring of genome integrity pathways in. *Escherichia coli*. *Cell Reports*. 14(3): 648–661.

Lal S, Pacis LB and Smith HM (2011) Regulation of the *SQUAMOSA PROMOTER-BINDING PROTEIN-LIKE genes/microRNA156* module by the homeodomain proteins PENNYWISE and POUND-FOOLISH in *Arabidopsis*. *Molecular Plant* 4(6): 1123–1132.

Lee J, Baum SF, Alvarez J, et al. (2005) Activation of *CRABS CLAW* in the nectaries and carpels of arabidopsis. *The Plant Cell* 17(1): 25–36.

Lin S (2010) Rank aggregation methods. *Wiley Interdisciplinary Reviews on Computational Statistics* 2(5): 555–570.

Lu P, Ma D, Chen Y, et al. (2013) L-glutamine provides acid resistance for *Escherichia coli* through enzymatic release of ammonia. *Cell Research* 23(5): 635–644.

Maathuis MH, Colombo D, Kalisch M, et al. (2010) Predicting causal effects in large-scale systems from observational data. *Nature Methods* 7(4): 247–248.

Marbach D, Costello JC, Küffner R, et al. (2012) Wisdom of crowds for robust gene network inference. *Nature Methods* 9(8): 796–804.

Margolin AA, Nemenman I, Basso K, et al. (2006a) ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 7(Suppl 1): S7.

Margolin AA, Wang K, Lim WK, et al. (2006b) Reverse engineering cellular networks. *Nature Protocols* 1(2): 662–671.

Masuda N and Church GM (2003) Regulatory network of acid resistance genes in *Escherichia coli*. *Molecular Microbiology* 48(3): 699–712.

Meysman P, Sonego P, Bianco L, et al. (2014) COLOMBOS v2.0: An ever expanding collection of bacterial expression compendia. *Nucleic Acids Research* 42(Database issue): D649–653.

Milkman R (1994) An *Escherichia coli* homologue of eukaryotic potassium channel proteins. *Proceedings of the National Academy of Sciences of the United States of America* 91(9): 3510–3514.

Sanchez-Corrales YE, Alvarez-Buylla ER, Luis M, et al. (2010) The *Arabidopsis thaliana* flower organ specification gene regulatory network determines a robust differentiation process. *Journal of Theoretical Biology* 264(3): 971–983.

Shi JX, Malitsky S, De Oliveira S, et al. (2011) SHINE Transcription factors act redundantly to pattern the archetypal surface of Arabidopsis flower organs. *PLOS Genetics* 7(5): 1–16.

Spirtes P and Glymour C (1991) An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review* 9: 62–72.

Stoyanov JV, Hobman JL and Brown NL (2001) CueR (YbbI) of *Escherichia coli* is a MerR family regulator controlling expression of the copper exporter CopA. *Molecular Microbiology* 39(2): 502–512.

Tan M, AlShalalfa M, Alhajj R, et al. (2008) Combining multiple types of biological data in constraint-based learning of gene regulatory networks. In: *IEEE Symposium on computational intelligence in bioinformatics and computational biology, CIBCB '08*, 2008, Sun Valley, ID, pp.90–97. IEEE.

Tan M, et al. (2011) Influence of prior knowledge in constraint-based learning of gene regulatory networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8(1): 130–142.

Tucker DL, Tucker N and Conway T (2002) Gene expression profiling of the pH response in *Escherichia coli*. *Journal of Bacteriology* 184(23): 6551–6558.

Volkert MR and Nguyen DC (1984) Induction of specific *Escherichia coli* genes by sublethal treatments with alkylating agents. *Proceedings of the National Academy of Sciences of the United States of America* 81(13): 4110–4114.

Wahl V, Brand LH, Guo YL, et al. (2010) The FANTASTIC FOUR proteins influence shoot meristem size in *Arabidopsis thaliana*. *BMC Plant Biology* 10(1): 285.

Wang M, Benedito VA, Zhao PX, et al. (2010) Inferring large-scale gene regulatory networks using a low-order constraint-based algorithm. *Molecular BioSystems* 6(6): 988–998.

Weber A, Kögl SA and Jung K (2006) Time-dependent proteome alterations under osmotic stress during aerobic and anaerobic growth in *Escherichia coli*. *Journal of Bacteriology* 188(20): 7165–7175.

Yoshida H, Maki Y, Furuike S, et al. (2012) YqjD is an inner membrane protein associated with stationary-phase ribosomes in *Escherichia coli*. *Journal of Bacteriology* 194(16): 4178–4183.

Zhang X, Zhao XM, He K, et al. (2012) Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics* 28(1): 98–104.

Zik M and Irish VF (2003) Global identification of target genes regulated by APETALA3 and PISTILLATA floral homeotic gene action. *Plant Cell* 15(1): 207–222.

## Appendix

The overall effect of the probability vector $\Pi$ in Algorithm 2 is such that the probability of a gene $g$ to be present in the $h$th subset of genes $T_{h,i}$ at the $i$th iteration is given by

$$P(g \in T_{h,i}) = \begin{cases} \pi_l + (1-\pi_l)\dfrac{d-\sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|}\pi_m}{|\mathcal{S}|-\sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|}\pi_m}, & \text{if } g = g_l \in \mathcal{S}_{\mathrm{LGN}} \\ \dfrac{d-\sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|}\pi_m}{|\mathcal{S}|-\sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|}\pi_m}, & \text{if } g \in \mathcal{S}\backslash\mathcal{S}_{\mathrm{LGN}} \end{cases} \quad (1)$$

where $\mathcal{S}$ is the set of candidate genes, $\mathcal{S}_{\mathrm{LGN}}$ is the set of genes of the LGN, $d$ with $|S_{\mathrm{LGN}}| < d \leq |S|$ is the subset dimension, $\pi_l$ is the $l$th component of $\Pi$ corresponding

to the probability of the $g_l$ gene of the LGN to be selected in the first selection, and $\sum_{m=1}^{|\mathrm{LGN}|} \pi_m$ is the expected number of LGN genes selected after the first selection. The last subset of each iteration is a special case: the third selection can intervene for its completion and the formula above does not hold anymore in a rigorous way. The exact correction of the fractional term of equation (1) requires a more detailed analysis that is beyond the current aim to illustrate the effect of $\Pi$.

The probability of a gene $g_l \in \mathcal{S}_{\mathrm{LGN}}$ of being in a subset is the convex combination of the probability of being in the subset of a gene that is not in the LGN, controlled by the parameter $\pi_l$. For a gene $g_l$ of the LGN, if $\pi_l = 1$ then $P(g_l \in T_{h,i})|_{\pi_l = 1} = 1$ and the $l$th gene is present in all the subsets. Alternatively, if $\pi_l = 0$ then

$$P(g_l \in T_{h,i})|_{\pi_l = 0} = \frac{d - \sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|} \pi_m}{|\mathcal{S}| - \sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|} \pi_m}$$

namely the same probability of a gene that is not in the LGN. The probability of equation (1) can be written as

$$\begin{aligned} P(g_l \in T_{h,i}) &= \pi_l + (1 - \pi_l) P(g_l \in T_{h,i})|_{\pi_l = 0}) = \\ &= \pi_l (1 - P(g_l \in T_{h,i})|_{\pi_l = 0}) + P(g_l \in T_{h,i})|_{\pi_l = 0} \end{aligned}$$

Setting $\pi_l$ permits modulation of the probability of the presence of each gene $g_l$ in the subsets. In the case $\pi_l = 0$ for all the genes of the LGN the probability becomes $P(g \in T_{h,i}) = d/|\mathcal{S}|$ for each gene and NES$^2$RA corresponds to NESRA where the probability of presence of the genes of the LGN in the subsets is the same of the other genes.

The number of executions of the skeleton procedures (Algorithm 3) that are generated in NES$^2$RA by the parameter $d$ are

$$\begin{aligned} O_{\mathrm{PC}} = \mathbb{E}(\#\text{ runs at iteration } j) = \\ \left\lceil \frac{|\mathcal{S}| - \sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|} \pi_m}{d - \sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|} \pi_m} \right\rceil \times P\left( \bigcap_h T_{h,j} \backslash S_{\mathrm{LGN}} \neq \emptyset \right) + \\ + \frac{|\mathcal{S}| - \sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|} \pi_m}{d - \sum_{m=1}^{|\mathcal{S}_{\mathrm{LGN}}|} \pi_m} \times P\left( \bigcap_h T_{h,j} \backslash S_{\mathrm{LGN}} = \emptyset \right) \end{aligned} \quad (2)$$

skeleton executions. Note that this formula is independent from the specific iteration $j$.

## Author biographies

*Francesco Asnicar* is a PhD student participating in the interdisciplinary PhD program on Computational Biology at the Department of Engineering and Computer Science (DISI) at the University of Trento, Italy. He received his MSc in computer science at the University of Trento.

*Luca Erculiani* is a master student of Computer Science at the University of Trento, Italy. His field of research is data mining.

*Francesca Galante* is a master student of Computer Science at the University of Trento, Italy.

*Caterina Gallo* is a master student at the Department of Mathematics at the University of Trento, Italy. She is currently working as a data analyst.

*Luca Masera* is a PhD student at the Department of Engineering and Computer Science (DISI) at the University of Trento, Italy. His research interests include Distributed Systems, Machine Learning and Bioinformatics.

*Paolo Morettin* is an MSc student in Computer Science at the University of Trento, Italy. His interests include Machine Learning, Data Mining and Logics.

*Nadir Sella* is a PhD student at the RNA dynamics and Biomolecular Systems group at Institut Curie in Paris. He received his MSc in computer science at the University of Trento. His research topic includes information-theoretic methods for the study of mutations in cancer.

*Stanislau Semeniuta* received the MSc degree in 2015 in Computer Science from the University of Trento, Italy. Currently he is a PhD student in the University of Luebeck, Germany.

*Thomas Tolio* received the MSc degree in Computer Science from the University of Trento, Italy, in 2015, working on a Thesis in Computational Metagenomics. At the moment he works as a Business Intelligence consultant.

*Giulia Malacarne* received an MSc degree and a PhD in Agro-Industrial Biotechnology from the University of Verona in 2003 and in 2007. Since 2007 she works at the Research & Innovation Centre of the Fondazione E. Mach (San Michele all'Adige, Italy) where she is a Researcher in the Gene Function Group. Her main research interests are grape berry quality and defence against pathogens by integrating genetic, metabolic and bioinformatics approaches.

*Kristof Engelen* holds a PhD in ''Engineering - Bioinformatics'' from the KU Leuven, Belgium. Since February 2013, he has been working at the Fondazione Edmund Mach as leader of the Integrative Genomics group in the

Department of Computational Biology. His main research interest is in the field of (top-down) systems biology and centered around transcriptomics: studying genome-wide transcriptional mechanisms and the role of gene regulatory networks in driving cellular behavior.

*Andrea Argentini* received an Msc and a PhD in Computer Science from the University of Trento respectively in 2008 and 2012. He is working as a post-doc at the CompOmics lab at Medical Biotechnology Center VIB (Belgium). His research interests include Machine Learning, mass-spectrometry based Proteomics and Bioinformatics.

*Valter Cavecchia* holds an MSc degree in Physics from the University of Trento, Italy. He is working as technologist, research assistant at the Institute of Materials for Electronics and Magnetism of the National Research Council of Italy in Trento. His research interests include computer graphics algorithms/techniques and volunteer-based distributed computing.

*Claudio Moser* has got an MSc degree in Biology from the University of Pavia, Italy and a PhD in Natural Sciences from the University of Heidelberg, Germany. Since 2001 he works at the Research & Innovation Centre of the Fondazione E. Mach (San Michele all'Adige, Italy) where he is Group Leader of the Gene Function group. His major research interests are the grapevine berry development and defence from pathogens studied by molecular, genetic and bioinformatic approaches. He is author of more than 40 peer reviewed publications.

*Enrico Blanzieri* received the Laurea degree (cum laude) in electronic engineering from the University of Bologna, Italy, and the PhD in cognitive science from the University of Turin, Italy, in 1992 and 1998, respectively. Since 2012, he is Associate Professor at the Dipartimento di Scienze e Ingegneria dell'Informazione, University of Trento, Italy where he works on machine learning and bioinformatics.