OXFORD

Briefings in Bioinformatics, 2025, **26(2)**, bbaf098
https://doi.org/10.1093/bib/bbaf098
Problem Solving Protocol

Testing and overcoming the limitations of modular response analysis

Jean-Pierre Borg[1,2,3], Jacques Colinge[1,2,3,*], Patrice Ravel[1,2,3,*]

[1]Université de Montpellier, 5 Bd Henri IV, 34000 Montpellier, France
[2]Institut régional du Cancer Montpellier (ICM), 208 Av. des Apothicaires, 34090 Montpellier, France
[3]Institut de Recherche en Cancérologie de Montpellier (IRCM), Inserm U1194, 208 Av. des Apothicaires, 34090 Montpellier, France

[*]Corresponding authors. Jacques Colinge, Université de Montpellier, Montpellier, France. E-mail: jacques.colinge@umontpellier.fr; Patrice Ravel, Université de Montpellier, Montpellier, France. E-mail: patrice.ravel@umontpellier.fr

Abstract

Modular response analysis (MRA) is an effective method to infer biological networks from perturbation data. However, it has several limitations such as strong sensitivity to noise, need of performing independent perturbations that hit a single node at a time, and linear approximation of dependencies within the network. Previously, we addressed the sensitivity of MRA to noise by reinterpreting MRA as a multilinear regression problem. We demonstrated the advantages of this approach over the conventional MRA and other known inference methods, particularly in handling noise measurements and nonlinear networks. Here, we provide new contributions to complement this theory. First, we overcome the need of perturbations to be independent, thereby augmenting MRA applicability. Second, using analysis of variance and lack-of-fit tests, we can now assess MRA compatibility with the data and identify the primary source of errors. In cases where nonlinearity prevails, we propose extending the model to a second-order polynomial. Third, we demonstrate how to effectively use prior knowledge about a network. We validated these results using 4 networks with known dynamics (3, 4, and 6 nodes) and 40 simulated networks, ranging from 10 to 200 nodes. Finally, we incorporated these innovations into our R software package MRARegress to offer a comprehensive, extended theory for MRA and to facilitate its use by the community. Mathematical aspects, tests details, and scripts are provided as Supplementary Information (see 'Data Availability Statement').

Keywords: MRA; regression; network inference; lack of fit; convex optimization; MRARegress

Introduction

The study of biological systems is inherently complex because they are governed by intricate molecular interactions that orchestrate the cell and tissue vital functions and responses to external stimuli. Therefore, the inference of biological networks is critical to understand homeostasis and diseases. Depending on the study granularity level, biological networks may involve genes, proteins, metabolites, or larger structures, such as protein complexes or even a subnetwork that would be regarded as a single object. Combinations are possible with some network participants, such as individual molecules, whereas other participants could be larger structures that do not need to be modeled in details. For concision, we name these unresolved larger structures modules.

In addition, to determine the activity of each network participant, a quantitative measure must be obtained in the various tested experimental conditions. In the case of a gene, transcriptional abundance is a natural measure, but it could also be the methylation level of its promoter. In the case of a protein, its direct abundance or activation status (e.g. phosphorylation level) is an obvious choice. In the case of a module, the chosen measure must represent its overall activity.

Several methods can be used to infer biological networks from the experimental measurements of their component activity [1]. These methods can be broadly categorized as follows:

- Data-based methods: such as statistical correlations (e.g. Pearson, Spearman, or Kendall [2]), Mutual Independence (e.g. ARACNE [3], CLR [4], or MRNET [5]), probabilistic graphical models (e.g. GeneNet [6]), Bayesian [7, 8], and regression methods (e.g. TIGRESS [9]).
- Machine learning methods (random forest [10], support vector machine [11], neural networks [12]). These methods are efficient, but require extensive datasets for training and validation.
- Perturbation-based methods, including the classical modular response analysis (MRA) [13], built using ordinary differential equations that describe the network dynamics. MRA is an effective method to infer networks, including when granularity may vary (i.e. the network includes modules).

This article focuses on the extension of the MRA approach to embed it in linear and polynomial regressions, and to exploit the available prior knowledge on some interactions in the network to be inferred. The first MRA developments were proposed by Boris Kholodenko in his seminal paper [13]. Subsequent enhancements incorporated noisy data integration and statistical estimation of

Received: September 23, 2024. **Revised:** January 4, 2025. **Accepted:** February 25, 2025

connectivity coefficients, using the maximum likelihood principle (MLMSMRA [14, 15]) or by estimating the confidence interval of the parameters by Monte Carlo analysis [16, 17].

Other authors adopted a Bayesian approach [18, 19], leveraging the distribution of prior knowledge on connectivity coefficients and partial network topology, including their sparsity, extracted from the literature or reference databases, such as STRING [20]. In our recent work, we extended MRA by expressing it as a multilinear regression problem [21] where the connectivity coefficients, derived from the regression residual variance and the model parameter, were estimated in parallel. This methodology, which we named MRARegress hereafter, allowed us to define a whole family of MRA methods, one method for each regression algorithm essentially (e.g. LASSO, STEP, least squares, random forest).

One of the important issues, raised by most of the previously quoted authors, is MRA sensitivity to measurement noise and perturbation levels, commonly known as signal-to-noise ratio [16, 21, 22]. This highlights the need to consider nonlinearity.

In the comparative analysis of the classical MRA method [13], some methods based on Mutual Independence [3] and MRARegress demonstrated MRARegress' superior performance in the presence of noisy data and medium to large networks (10 to 1000 nodes) [21]. In this article, the significance of the connectivity coefficients is naturally linked to the estimation of the regression parameters in the form of 95% confidence intervals. In addition, dimension-reduction methods, such as Step Forward, have demonstrated their effectiveness in inferring sparse networks compared to standard methods.

Here, we enriched MRARegress functionalities by proposing holistic solutions to its constraints that we then validated across networks with known dynamics and extensive simulations. Specifically, following a review of the principles and prerequisites for applying MRA:

- We optimized the use of the available measurement data by allowing the use of perturbations that are not necessarily independent, and by accounting for both technical and biological replicates. This strategy improves the prediction of connectivity coefficients, an improvement verified by systematically and automatically estimating the parameter confidence intervals via the MRARegress package.
- We introduced a Lack Of Fit (LOF) test to assess nonlinearity versus measurement noise and determine the linear model relevance.
- For cases when the model is not relevant, we detailed an extension of MRARegress that accommodates nonlinearity by incorporating second-order terms, thereby harnessing potential synergistic effects between variables.
- For cases when prior partial knowledge of the studied network is available, we showed that MRARegress, coupled with convex optimization, effectively incorporates this knowledge.

These methodological developments have led to the creation of an R package, that we also named MRARegress, and that provides user-friendly and free access to our new MRA algorithms for the scientific community.

Methods

In MRA, every node within a network is sequentially perturbed and the network response to these perturbations is analyzed. Each node must be associated with a measurable quantity related to its activity: abundance or state (e.g. phosphorylation level of a protein or expression level of a gene). Perturbations may include administering a drug, small interfering RNA (siRNA), or small hairpin RNA (shRNA) that completely inhibits, reduces, or increases the expression of a target gene. These perturbations are referred to as Knock Out (KO), Knock Down (KD), or Knock Up, respectively. Figure 1 illustrates the MRA method.

Mathematical aspects

Expression of ΔX_i *based on* $\Delta X_k, k \neq i$

All the measures associated with the N modules at a given time will be represented by a vector.

$X \in \mathbb{R}^N$. Thus, $X_i(t)$ corresponds to the measurement associated with node i at time t. Because of the interconnection of the modules, X_i depends on the other X_j $(j \neq i)$.

All $X_i, i \in [\![1, N]\!]$ also depend on a set of M parameters p_k, $k \in [\![1, M]\!]$ (identified by a vector P) and the time t.

In a very general way, this vector X is the solution of the system of differential equations:

$$\frac{dX}{dt} = f(t, X(t, P), P), \text{ with } t \in \mathbb{R}^+, X \in \mathbb{R}^N \text{ and } P \in \mathbb{R}^M \quad (1)$$

A fundamental assumption of MRA is that there is a set of parameters, referred to as P_0, a neighborhood of P_0 (noted $V(P_0)$) and a time t_0, beyond which the system, starting from the initial conditions $X(0, P)$, reaches a stable and stationary state, which will be noted $X(P_0)$. For networks that comply with this hypothesis, all the measurements $X_i(t, P)$ become constant. That is,

$\forall i \in [\![1, N]\!]$, $\frac{dX_i}{dt} = 0$, which is written $\frac{dX}{dt} = 0$ and (1) becomes:

$$\forall t > t_0, \forall P \in V(P_0), \quad f(X(P), P) = 0. \quad (2)$$

Therefore, MRA cannot be used when the network dynamics are periodic or pseudo-periodic.

MRA estimates the gradient of the function f, and the gradient components represent the interactions of the module network. They are called connectivity coefficients and are the unknowns of the problem.

Function f, of $\mathbb{R}^{N+M}$ in $\mathbb{R}^N$, is usually not known. Its gradient can be approached with the following development.

Assuming the function f of class C^1, it can be demonstrated, using the implicit function theorem [23] and Taylor's first order development, that for each component X_i of vector X, it exists a neighborhood Ω of $(X(P_0), P_0)$ and a function φ_i which depends on other X_j components and parameters, such as:

$$\Delta X_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} r^e_{i,k} * \Delta X_k + \sum_{j=1}^{M} \frac{\partial \varphi_i}{\partial p_j} * \Delta {P_0}^j + o(\|\Delta P_0\|) \quad (3)$$

The functions φ_i are generally unknown. We only know that they exist and that they exhibit the same degree of smoothness as f. Consequently, they are differentiable if f is of class C^1.

$\Delta X_j = X_j(P_0 + \Delta P_0) - X_j(P_0), \forall j \in [\![1, N]\!]$ represents the modification of module j expression when applying a ΔP_0 perturbation from the stationary state P_0.

$r^e_{i,k}$, called exact connectivity coefficient, represents the action of node k on node i.

The implicit function theorem allows, for $k \neq i$, to calculate the derivatives.

$r^e_{i,k} = \frac{\partial \varphi_i}{\partial X_k}(P_0) = -\frac{\partial f_i / \partial X_k}{\partial f_i / \partial X_i}(P_0)$. For $k = i$, this definition is extended by setting $r^e_{i,i} = -1$ to simplify the subsequent calculations.

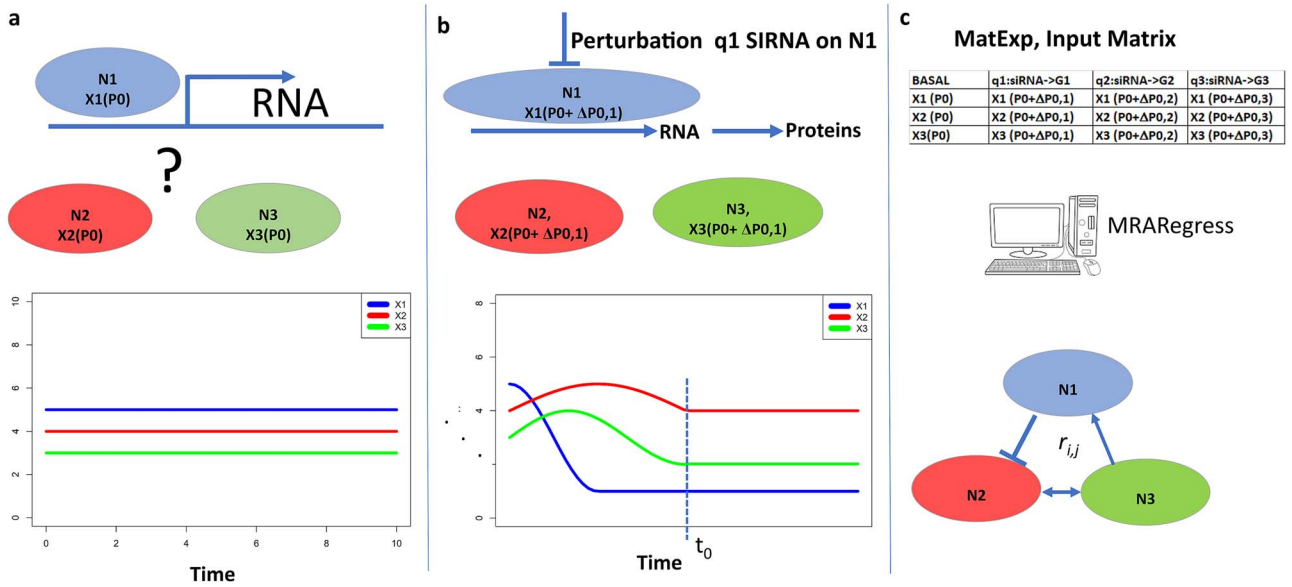

Figure 1. Schematic illustration of the methodology associated with the MRA method for a network of three genes N1, N2, N3. (a) Unperturbed network (basal state). (b) A perturbation of type siRNA (q1), specific to gene N1, reduces RNA production and consequently the protein associated with this gene. The measurement X associated with the network's activity (N1, N2, N3) can be assessed by RNAseq or by the enzymatic activity of the expressed proteins, when the steady state is reached at t_0. Similarly, starting from the basal state (a), this experiment is successively reproduced for two other siRNA perturbations, (q2) then (q3) on genes N2 and N3. (c) All measurements are stored in an input matrix MatEXP. The "MRARegress" package can then infer the network from this matrix and estimate the connectivity coefficients r_(i,j). The arc connecting node j (origin) to node i (target) represents the action of gene j on gene i, and r_(i,j) measures this action. See SI Fig. S1 for details of MatEXP processing in a similar example.

The other terms of this equation (3) are explained in the Supplementary Information (SI § 1.1).

Assumption of the independence of perturbations

Now, let us assume that:

$M = N$, p_i influences ONLY X_i (in other words: $\frac{\partial \varphi_i}{\partial p_j} = 0$ if $j \neq i$ and $\frac{\partial \varphi_i}{\partial p_i} \neq 0$), AND perturbations ΔP_0 can be achieved that affect the parameter $p_i, \forall i \in [\![1, N]\!]$.

This strong hypothesis, both from a mathematical and biological point of view, was made implicitly or explicitly in various articles on MRA [1, 13, 24]. Throughout this article, this hypothesis will be designated as the "Assumption of Independence Of Perturbations" (AIOP). This assumption means that each X_i depends on only one parameter, denoted by P_i (from which it follows that $N = M$). It further states that there is a way to perturb *only* that parameter, and we denote such a perturbation by q_i.

AIOP is verified in certain gene or protein networks. From a practical point of view, chemical compounds, antibodies, siRNAs, or shRNAs can perturb the activity of specific nodes.

In these conditions, equation (3) becomes:

$$\Delta X_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} r^e_{i,k} * \Delta X_k + \frac{\partial \varphi_i}{\partial p_i} * \Delta {P_0}^i + o(\|\Delta P_0\|) \qquad (4)$$

See Fig. 2 for a geometric interpretation of this equation.

By neglecting the remainder $o(\|\Delta P_0\|)$, which is acceptable for small perturbations, and by remembering that $r^e_{i,i} = -1$, the system of approximate equations is obtained:

$$\sum_{k=1}^{N} r_{i,k} * \Delta X_k + \frac{\partial \varphi_i}{\partial p_i} * \Delta {P_0}^i = 0 \qquad (5)$$

(note that $r^e_{i,k}$ have become $r_{i,k}$).

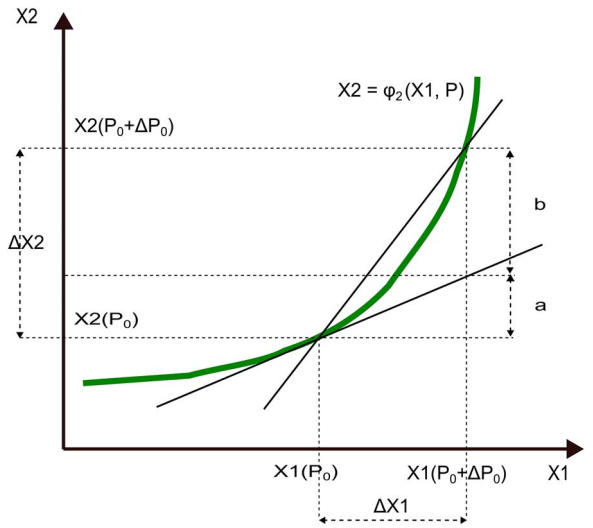

Figure 2. a and b are, respectively, the linear and nonlinear parts of the model following a variation in the parameters (X_1 increases by ΔX_1 and X_2 by ΔX_2). The first-order approximation leads to an increase of a. We can observe the error b caused by this approximation.

If one chooses a ΔP_0 perturbation that acts ONLY on the parameter $p_j, (j \neq i)$, therefore on the node j, which is possible according to AIOP (perturbation designated by q_j), the coordinates of q_j on the other parameters p_k, with $k \neq j$, are null.

This leads to a system of $N-1$ equations to $N-1$ unknowns $r_{i,k}, k \neq i$, if $N-1$ perturbations of this type can be performed.

$$\sum_{k=1}^{N} r_{i,k} * \Delta X_{k,j} = 0, \text{with } j \neq i \qquad (6)$$

In this system of equations, $\Delta X_{k,j} = X_k(P_0 + q_j) - X_k(P_0)$ and $r_{i,i} = -1$. The solution of this system allows computing the coefficients $r_{i,k}$ if the rank of the matrix $(\Delta X_{k,j}), k$ and$j \neq i$, is $N-1$.

By giving i the successive values 1 to N, all coefficients $r_{i,j}$ of the connectivity matrix, called "r," can be obtained. The coefficient

$r_{i,j}$ is an approximate value of $r^e_{i,j}$ and will be called connectivity coefficient. It also represents the action of node j on node i. See Supplementary Informations (SI) (SI Fig. S1) for an illustration of the method, with an example that can be calculated by hand.

In practice, other parameters may affect the quantities X_i, or some parameters may act on multiple nodes, in which case AIOP is not satisfied. Because our approach is intended to be general, one of the contributions of this article is to demonstrate that, under certain conditions, this type of experimental design can be nonetheless handled to infer networks.

If AIOP is not true, equation (3) cannot be simplified. By neglecting the remainder $o(\|\Delta P_0\|)$, equation 7 is obtained:

$$\sum_{k=1}^{N} r_{i,k} * \Delta X_k = -\left(\sum_{j=1}^{M} \frac{\partial \varphi_i}{\partial p_j} * \Delta {P_0}^j\right) \tag{7}$$

This equation can be applied to Q perturbations $q_m, m \in [\![1, Q]\!]$. For each value of $i \in [\![1, N]\!]$, equation (7) becomes a system of Q linear equations:

$$\sum_{k=1}^{N} r_{i,k} * \Delta X_{k,m} = -\left(\sum_{j=1}^{M} \frac{\partial \varphi_i}{\partial p_j} * q^j_m\right) \tag{8}$$

q^j_m is the jth coordinate of perturbation q_m in $\mathbb{R}^M$ and $\Delta X_{k,m} = X_k(P_0 + q_m) - X_k(P_0)$.

This system of Q equations (8) has $N - 1 + Q * M$ unknowns in general:

- $r_{i,k}$, $\quad k \in [\![1, N]\!], k \neq i$,
- $\frac{\partial \varphi_i}{\partial p_j} * q^j_m$, $\quad j \in [\![1, M]\!], m \in [\![1, Q]\!]$.

Without additional hypothesis on perturbations, this system cannot be solved because there are more unknowns than equations. Consequently, we have combined $\frac{\partial \varphi_i}{\partial p_j}$ and q^j_m into a single unknown.

in order to reduce the total number of unknowns, given that we do not seek their individual values.

We say that, for a given network, AIOP is *partially* satisfied if, for each node i:

- On the one hand, there exists a set of Q perturbations q_m, such that, for each parameter p_j, either p_j does not influence node i or q_m does not act on parameter p_j.
- On the other hand, the rank of the square matrix (derived from these perturbations) $[\Delta X_{k,m}], k \in [1, N], k \neq i$ and $m \in [\![1, Q]\!]$ is equal to $N - 1$.

For these networks, however, the system (8) will have a solution.

The MRARegress package automatically checks this condition and solves the system when possible (see SI § 1.2).

Principle of the LOF test

For a node i of the network, equation (6), in which the remainder $o(\|\Delta P_0\|)$ is not neglected, can be written as follows, because $r_{i,i} = -1$:

$$\Delta X_{i,j} = \sum_{\substack{k=1 \\ k \neq i}}^{N} r_{i,k} * \Delta X_{k,j} + \epsilon, j \epsilon [\![1, Q]\!]$$

Errors due to nonlinearity $o(\|\Delta P_0\|)$ and measurement noise have been grouped in the term ϵ.

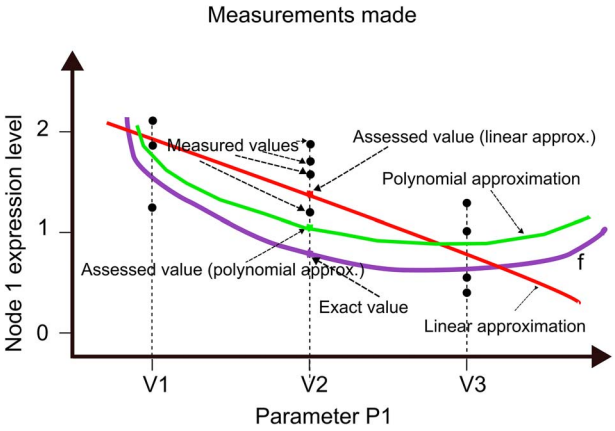

Figure 3. Illustration of the estimation of connectivity coefficients using linear or polynomial regression. The lower curve represents X1 as a function of parameter P1. To approach this curve, P1 is perturbed by successively giving it the values V1, V2, V3. For each value of P1, several measurements (technical replicates) corresponding to the black dots on the figure are performed. From these measurements, the upper line is obtained by linear regression and the middle curve by polynomial approximation.

In the presence of replicates, under certain regularity assumptions [25], the LOF test can be applied to a multilinear regression [26]. This technique allows us, using replicates, to estimate for each node the pure error variance (due to measurement noise only), as shown in Fig. 3.

For each node i, Q measurements were performed (perturbations including replicates, $Q = 11$ for P_1 in Fig. 3).

By performing two analysis of variance (ANOVA) tests on the regression itself and the model adequacy (see SI § 3), one can estimate the sum of squares error due to measurement noise (SSE_P) and due to model inadequacy (SSE_{LOF}), together with the corresponding degrees of freedom (df_P and df_{LOF}).

The LOF test is a Fisher's test of the estimated variances $CME_p = \frac{SSE_P}{df_P}$ and $CME_{LOF} = \frac{SSE_{LOF}}{df_{LOF}}$ where the degrees of freedom are $df_P = \sum_{t=1}^{m}(n_t - 1) = Q - m$ and $df_{LOF} = (Q - N + 1) - (Q - m) =$.

$m - N + 1$, respectively. Here, m is the number of values given to the parameters and n_t the number of measurements made for each of these values.

If the CME_{LOF}/CME_p ratio is large (P-value <.05), this test rejects the null hypothesis "residual errors stem from measurement noise" and concludes that there is a significant effect of the model nonlinearity, relative to the measurements noise. Otherwise, the linear model is considered acceptable if the noise level, estimated by the replicates, is also acceptable.

Extension to polynomial regression (order 2)

If the linear model is not acceptable, let us assume that:

f is now a class C^2 function of $\mathbb{R}^{N+M}$ in $\mathbb{R}^N$, functions φ_i have continuous first- and second-order partial derivatives in the vicinity of $(X(P_0), P_0)$.

Using again the implicit functions theorem [23] and Taylor's development to order 2 this time, it can be demonstrated that for each X_i component of vector X in the vicinity of P_0:

$$\Delta X_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} r^e_{i,k} * \Delta X_k + \frac{1}{2} * \sum_{\substack{k=1 \\ k \neq i}}^{N} \sum_{\substack{m=1 \\ m \neq i}}^{N} s^e_{i,m,k} * \Delta X_m * \Delta X_k + err \tag{9}$$

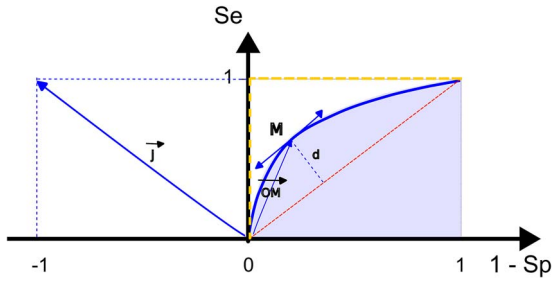

Figure 4. The ROC curve (Sensitivity vs. 1 – Specificity) is shown. One commonly used quality measure is the AUROC (Area Under the ROC Curve), shaded in light blue in the diagram. We introduce an equivalent quality indicator, but one that is faster to compute: the Distance to the Diagonal (see SI §5).

$s^e_{i,m,k}$ represents the influence of quadratic terms ($s^e_{i,k,k} * (\Delta X_k)^2$) and quadratic cross terms

($s^e_{i,m,k} * \Delta X_m * \Delta X_k, m \neq k$). See SI § 4 for an explanation of the other terms.

For each i, equation (9) reveals $N-1$ numbers $r^e_{i,k}$ and $\frac{N*(N-1)}{2}$ numbers $s^e_{i,m,k}$ because $s^e_{i,m,k} = s^e_{i,k,m}$. It is a total of $\frac{N*(N-1)*(N+2)}{2}$ unknowns.

As in the linear case, by applying Q perturbationsq_p, with $p \in [\![1, Q]\!]$ and neglecting an error of order 1 or 2, the approximate coefficients $r_{i,k}$ and $s_{i,m,k}$ can be found by solving the system of linear equations:

$$\Delta X_{i,p} = \sum_{\substack{k=1 \\ k \neq i}}^{N} r_{i,k} * \Delta X_{k,p} + \frac{1}{2} * \sum_{\substack{k=1 \\ k \neq i}}^{N} \sum_{\substack{m=1 \\ m \neq i}}^{N} s_{i,m,k} * \Delta X_{m,p} * \Delta X_{k,p} \quad (10)$$

if the system rank is sufficient.

In this scenario, the Q perturbations must satisfy the rank condition for system (10). In practice, it is possible to use double perturbations to infer cross terms.

Methods used to evaluate results

The purpose of the methods to measure module network inference is to calculate, as accurately as possible, the real connectivity coefficients $r^e_{i,j}$, because they correspond to the interactions between modules in the network. To evaluate the new developments described in this article, the exact (r^e) and the computed (r) connectivity matrices were compared by means of scores that depend on what is known about $\left(r^e_{i,j}\right)$:

- If $r^e_{i,j}$ exact values are known, we measure the distance (according to the L2-norm): $d\ (r^e, r)$. Smaller distances indicate better results.
- If only the existence or absence of interactions between nodes is known (Boolean networks), the Euclidean distance does not make sense. We then use the "Distance to Diagonal" (DtoD). This indicator, we defined (see SI § 5 and [21]), is equivalent to AuROC but is much less time consuming. $DtoD = 0$ if detection is purely random and $DtoD = 1$ if detection is perfect (Fig. 4), see also SI Figure S2. Larger values indicate better results.

The results given in this article (tables, figures) are derived from simulations (see SI § 6 and 7 for details):

- Four networks with known dynamics (3, 4, and 6 nodes).
- Five 10-node networks and five 100-node networks from DREAM Challenge (DC4): (info@sagebase.org—https://www.synapse.org/#!Synapse:syn3049712/wiki/74630).
- Thirty networks generated by FRANK [27], corresponding to networks of 30, 60, and 100 nodes where all genes interact (TF = 30, 60, or 100 and TA = 0) and networks of 60, 100, and 200 nodes where half of the genes do not regulate any other gene (TF = TA = 30, 50, or 100). Noise, simulated by independent Gaussian random variables ($N(0, \sigma), \sigma = k *$ the mean concentration of genes), was added to the measurements of these 30 networks, corresponding to the gene expression levels, non-perturbed or perturbed by a KO (−100%) and a KD (−50%). Two noise levels were simulated ($k = 0.1$: medium noise and $k = 0.5$: strong noise).

Noise was simulated by adding independent Gaussian random variables $N(0, \sigma)$ to the measures. Noise level σ was proportional to the signal level because, in many cases, the signal is electronically amplified to facilitate its detection and measurement. Accordingly, the important parameter is the signal-to-noise ratio (represented by k). The script AdvancedMRA.R (available from GitHub) enables reproducing the simulations, and SI explains how to proceed with real data and how to check results relevance (§ 6.4 and 6.5).

MRARegress package

The theoretical study allowed developing an easy-to-use software package, also called MRARegress, that implements these advancements and checks the necessary conditions. This package, written in R, is freely available to the Scientific Community on GitHub, including the source code and the many unit tests data, covering 92% of the code.

Installation guidelines and simple examples for verification are provided in SI § 9 as well as an overview of its main features (see SI Figure S4 for an illustration of its various outcomes). The list of R packages used and their version are indicated. A thumbnail presentation, an online manual, and email support are available.

All results presented here were obtained using this software (Data Availability).

Results and discussion

Statement of results

In a previous article [21], we demonstrated that MRA based on linear regression (MRARegress) substantially mitigates result estimation errors in the presence of noisy measurements or significant nonlinearity of the function (referred to as "f") that describes the system dynamics. Here, we extended MRARegress:

- By permitting the use of not necessarily independent perturbations (given sufficient system rank),
- By enabling the detection of the estimation error primary source by ANOVA that considers the availability of replicates,
- By facilitating the use of polynomial regression (order 2) when error is primarily attributed to the nonlinearity of "f" and noise levels are not excessive,
- By incorporating the user's prior knowledge of the studied networks,
- By developing a software package named MRARegress that integrates previous work [21] and all the improvements brought by this study.

Assumption of independence of perturbations

From both mathematical and biological perspectives, AIOP is a strong assumption. It limits the types of perturbations that can be used to infer networks using classical MRA [13] because each perturbation must act on exactly one system parameter, and that

parameter must affect exactly one node. To illustrate the importance of this assumption, we provide in SI (§ 6.3.3) a very simple analytical example (N = 2 nodes, M = 2 parameters) in which $r^e_{1,2}$ and $r^e_{2,1}$ can be computed analytically:

$$\begin{cases} X1 = \rho * \theta * \cos\theta \\ X2 = \rho * \theta * \sin\theta \end{cases}$$

in the vicinity of $P0\left(\rho = 1, \theta = \frac{\pi}{4}\right)$, and $r^e_{1,2} = r^e_{2,1} = -1$ (see SI). In this example, AIOP is not satisfied. If we ignore this restriction and apply MRA perturbing ρ and θ, we obtain $r_{1,2} = 0.25$ and $r_{2,1} = 1$, which are largely incorrect.

By contrast, MRARegress can handle this problem by using non-independent perturbations, yielding $r_{1,2} = -1.46$ and $r_{2,1} = -0.68$. These values are much closer to the expected result (−1, −1). The discrepancy arises from the nonlinearity in functions φ_1 and φ_2. The ability to use non-independent perturbations also makes it possible to carry out more practically feasible perturbations (SI § 7.1.1) and to employ experimental designs more suitable for second-order analyses.

Errors due to measurement noise

The error in estimating the coefficients $r^e_{i,j}$ by the MRA method originates from two sources:

- Measurement noise (e.g. experimental conditions, measuring devices accuracy, operators' skills),
- Nonlinearity of function "f" in equation (2).

To decrease the measurement noise-induced error, it is advisable to increase the perturbation levels to boost the signal/noise ratio [21]. Unfortunately, the perturbation level cannot always be chosen experimentally. In addition, increasing this level increases the nonlinearity-induced error. Hence, a balance must be found.

Alternatively, conventional methods involve multiplying measurements by providing multiple values to the p_j parameters, or performing successive measurements with the same parameter set (technical replicates), or on similar objects (e.g. several samples of the same cell line: biological replicates). While the classical MRA method only accommodates data averaging [22], linear regression leverages over-determined systems (number of measurements exceeding N) far more effectively. In the presence of noisy measurements, this technique significantly reduces the result errors, as demonstrated in [19, 21].

Errors due to nonlinearity

Given the cost and difficulty of measurements, it is crucial to optimize the number of measurements and parameter values.

Analyzing the error variance (using ANOVA) allows identifying its primary cause and assessing whether the used linear model is appropriate (with the LOF test). If the error predominantly stems from the nonlinearity of function "f", a polynomial regression (of order 2) should be used instead of linear regression.

We applied these methodologies to various networks (see Methods) in which three perturbations (−80%, −10%, and −1%) for the 3- and 4-node networks and five perturbations (−80%, −50%, −10%, −1%, and +50%) for the 6-node network were simulated. Each measurement had two replicates, corresponding to very low noise (independent Gaussian random variables $N(0, \sigma)$, $\sigma = 0.01 *$ mean of non-perturbed expressions):

- 3-Kinase network,
- Linear 3-gene network,
- 4-Node network,
- MAPK cascade (6 nodes).

We simulated low noise levels to determine whether the observed error arises from the regression model itself. Indeed, if the main source of error is model nonlinearity, increasing the number of measurements does not help. For the ANOVA results to be significant, the measurement noise must be small relative to the signal.

Table 1 shows the results for the 3-kinase network.

The obtained results indicated that the P-value for SSR was $<10^{-6}$. Consequently, the null hypothesis stating that "the value of ΔX_i is not correlated with the values of $\Delta X_k, k \neq i$" could be rejected for all nodes.

Conversely, the P-value for SSE_{LOF} ranged between 0 and 17%. For some nodes, the null hypothesis (H_0) "residual errors stem from measurement noise" could be rejected with 95% certainty, indicating that for these nodes, error originated mostly from the chosen linear model, rendering linear regression inadequate. This concerned two of the three nodes according to the MRARegress package.

In the linear 3-gene network (see Table 2), the null hypothesis (H_0) could not be rejected with 95% certainty for any node, consistent with the linear system of equations that described the dynamics (SSE_{LOF} P-value ≥7% for all nodes).

For the 4-node and MAPK cascade networks, results were similar to those obtained for the 3-kinase network.

The detailed results for each network (F-value and P-value for each node) are in SI § 7.1.

Figure 5 describes the discovered networks using the specified perturbations with two replicates.

Polynomial regression (order 2)

ANOVA indicated the presence of significant error due to nonlinearity for several nodes within these networks. To evaluate the utility of processing this nonlinearity, we computed the L2-norm distance between the exact and approximate connectivity matrices, for the three nonlinear networks (3-kinase, 4-node, and MAPK cascade) using the linear MRARegress and polynomial MRARegress (second degree) methods.

Results are summarized in Table 3 and the detailed matrices are provided in SI § 7.1.

The superiority of employing second-order polynomial regression is evident. However, this method introduces more unknowns than linear regression, rendering results more susceptible to measurement noise.

We performed the sensitivity analysis by simulating measurement noise at different levels (see Methods). This demonstrated the advantage of using ANOVA for gene expression measurements. If ANOVA reveals that the primary source of errors stems from the function f nonlinearity and measurement noise is low, employing second-order polynomial regression is highly advantageous; otherwise, linear regression remains preferable.

The obtained results are listed in Table 4 (average over 20 simulations for each value of the variation coefficient k). Networks, perturbations, and replicates were the same as in Table 3.

Contribution of prior knowledge

Linear regression offers many advantages to the MRA method due to extensive mathematical developments that facilitate module network inference. For example, various cost functions can be used to consider the network characteristics (e.g. size, sparsity) and the importance attributed to specific parameters. Different methods corresponding to these cost functions (e.g. LSE, LASSO,

Table 1. ANOVA results for the 3-kinase network

Variation	*df*	Sum of squares	*F*-value	*P*-value
Regression (SSR)	2	10.08; 87.59	6312.53; 26 831.38	<1E-6; <1E-6
Error (SSE)	10	0.01; 0.05		
Lack-of-fit error (SSE_{LOF})	4	0.01; 0.03	2.33; 25.87	<1E-6; .17
Pure error (SSE_P)	6	<1E-6; <1E-6		
Total	12			

df: degrees of freedom; the value pairs, shown in the table and separated by semicolons, represent the minimum and maximum values for all nodes.

Table 2. ANOVA results for the linear 3-gene network

Variation	*df*	Sum of Squares	*F*-value	*P*-value
Regression (SSR)	2	1.46; 5.51	485.54; 1629.26	<1E-6; <1E-6
Error (SSE)	10	0.01; 0.02		
Lack-of-fit error (SSE_{LOF})	4	<1E-6; 0.01	0.04;3.94	.07; 1
Pure error (SSE_P)	6	<1E-6; 0.02		
Total	12			

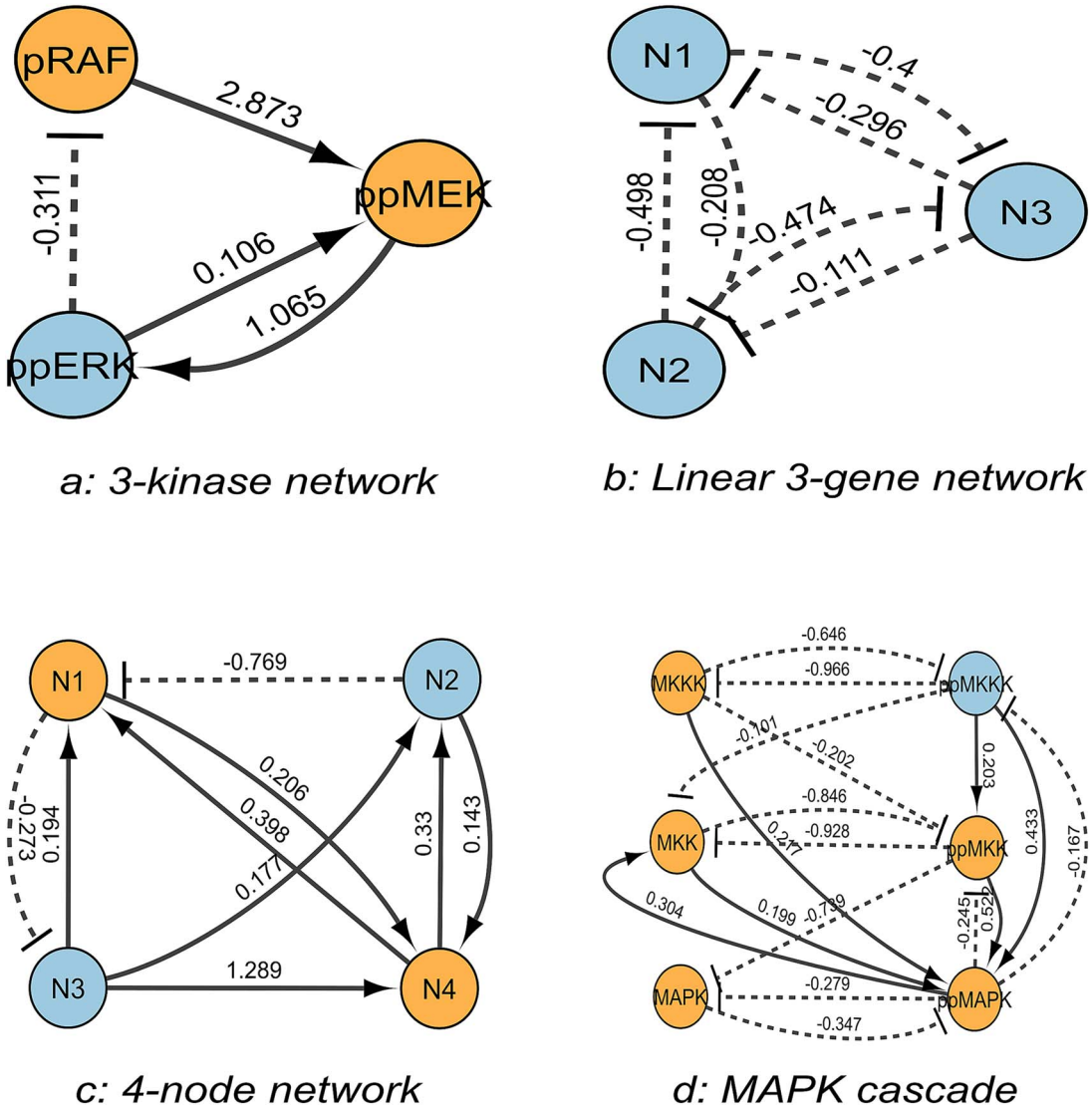

Figure 5. Representation by MRARegress of the four test networks whose dynamics are known and used in the article. Nonlinearity detected by MRARegress is highlighted in orange (blue means nodes for which linear regression is appropriate). Solid line arcs indicate an amplification of the expression of node i (target) by node j (origin). Dotted arcs indicate inhibition. The values associated with each arc corresponds to ri,j.

Table 3. Euclidean distance between exact and approximate connectivity matrices (noise-free measurements)

Network	3 Kinases	4 Genes (4 nodes)	MAPK cascade (6 nodes)
Perturbations	−80%, −10%, −1%	−80%, −10%, −1%	−80%, −50%, −10%, −1%, +50%
Distance (linear MRA)	0.254	0.617	0.871
Distance (2nd degree MRA)	0.009	0.002	0.036

Table 4. Euclidean distances between exact and approximate connectivity matrices, depending on the noise level

Network		3 Kinases	4 Genes (4 nodes)	MAPK cascade (6 nodes)
Perturbations		−80%, −10%, −1%	−80%, −10%, −1%	−80%, −50%, −10%, −1%, +50%
$k = 0.001$	Linear MRA	0.253	0.615	0.873
	2nd degree	0.106	0.199	0.190
$k = 0.003$	Linear MRA	0.253	0.611	0.877
	2nd degree	0.227	0.411	0.455
$k = 0.005$	Linear MRA	0.253	0.608	0.880
	2nd degree	0.361	0.568	0.722
$k = 0.007$	Linear MRA	0.254	0.604	0.888
	2nd degree	0.508	0.716	0.974

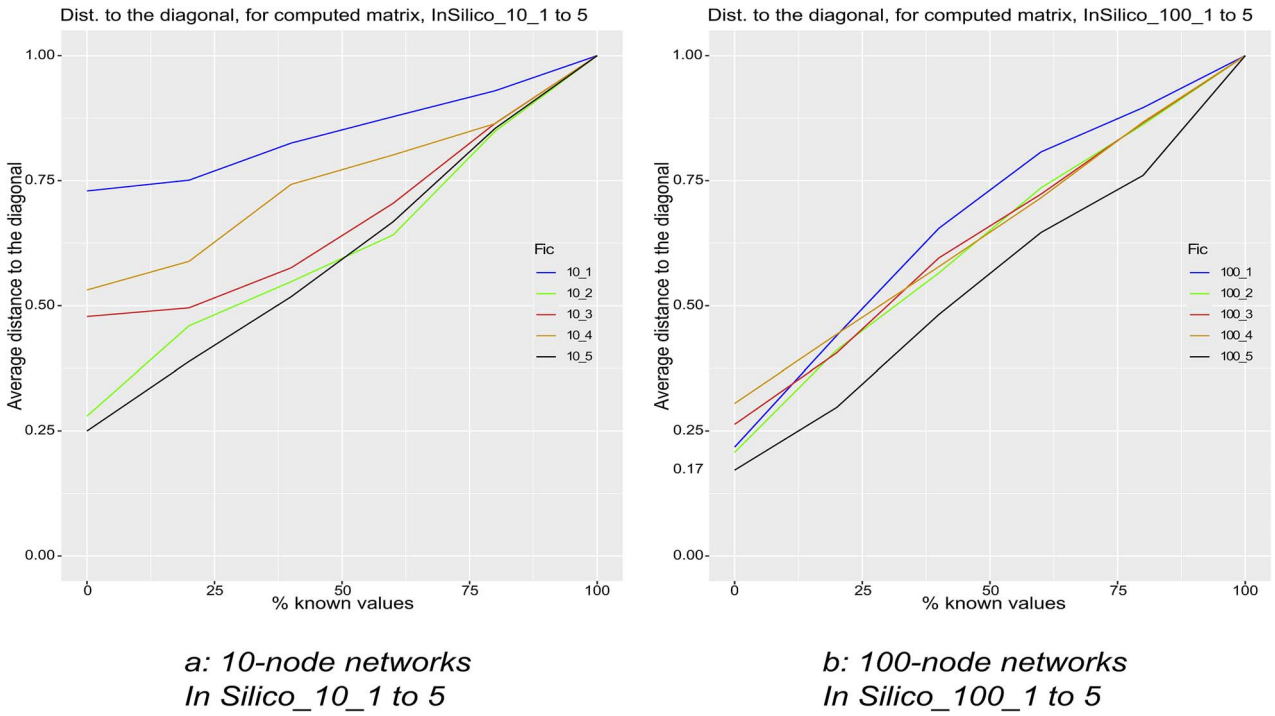

Figure 6. Mean values of the distance to diagonal of the DC4 networks based on the percentage of known data (10 random draws for each percentage). To maintain readability, the standard deviations are not displayed in the figures, but are provided in SI (Supplementary Table 1). The colors correspond to the different networks of the DC4 challenge as indicated in the legend. Distance is 0 in case of purely random detection and 1 for a perfect detection.

RIDGE, Elastic Net, STEP) have been compared [21] for problem dimension reduction and for eliminating irrelevant network arcs.

Whatever the cost function, the MRARegress method typically involves solving N optimization problems with values in $\mathbb{R}^{N-1}$.

However, in some cases, prior information on these values may be available (e.g. from biological knowledge databases, e.g. STRING or Reactome) such as:

- The j module has no effect on the i module (therefore, $r_{i,j} = 0$), or
- The j module amplifies or inhibits the action of the i module (therefore, $r_{i,j} \geq \alpha$, or $r_{i,j} \leq -\alpha$ where $\alpha \geq 0$) or
- There is a linear relationship between specific coefficients $r_{i,j}$ etc ...

The incorporation of this information implies that the solutions of the optimization problems are not in $\mathbb{R}^{N-1}$, but lie within a convex subspace of it.

Furthermore, leveraging regression-based methods grants access to a comprehensive mathematical toolkit. For instance, the CVXR library of R functions, developed by João Neto, based on the work by [28], addresses convex optimization problems. This library is applicable when the cost function used and

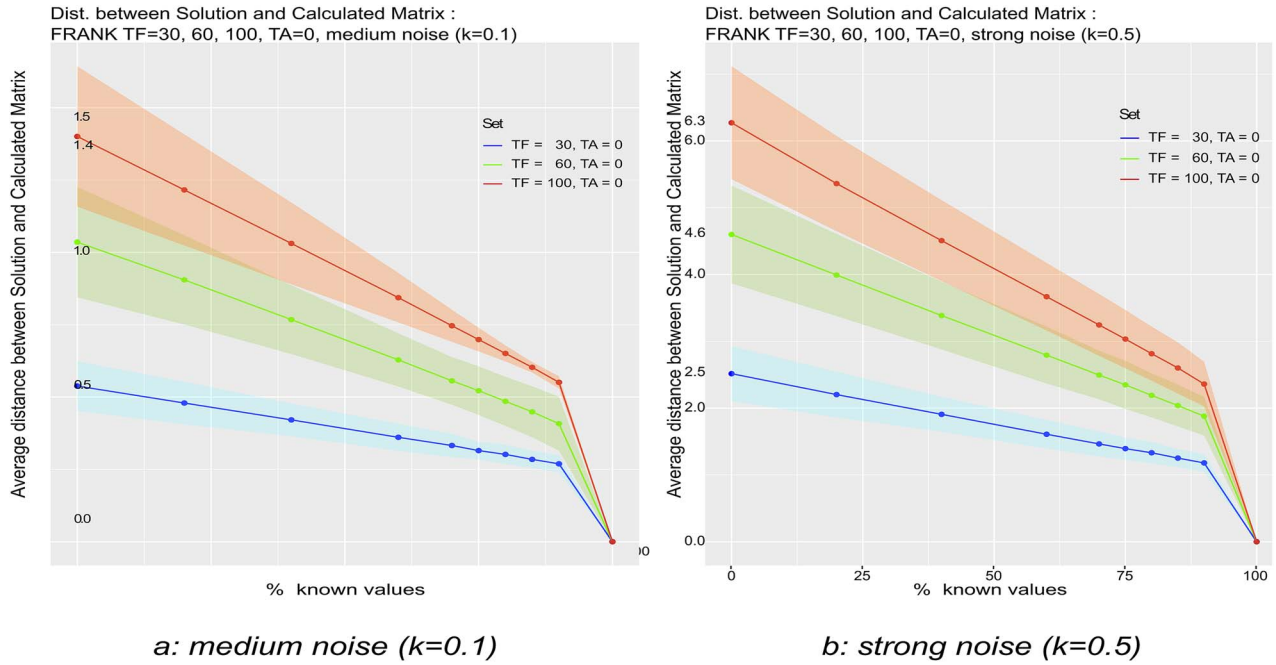

Figure 7. Average distance between solution and calculated matrix for FRANK-generated networks (30: lower curve, 60: middle curve, 100 nodes: upper curve) in function of the percentage of known data and noise level. Each curve shows the mean value ± standard deviation (smoothing line) – 5 networks for each curve. For these networks, the exact solution is known, so the score is obtained by measuring the Euclidean distance between the exact and computed matrices re and r. the lower this value, the better the detection (0 if perfect detection). It can be seen that the gain is practically linear as a function of the percentage of known values.

all inequality type constraints on the unknown $r_{i,j}$ are convex functions, and all equality type constraints are affine. This suitability can be met by using the least squares method as a cost function (also the Threshold Linear Regression method, called TLR, of the MRARegress package, see SI § 5). The MRARegress package facilitates incorporating most of these constraints, as described in SI § 8.2.

The efficacy of this knowledge in terms of performance was assessed using 25 networks in total (see SI § 8.3). We compared the simulation results with the known solutions.

Figure 6 shows these scores, in function of the percentage of known data ("p," see SI § 8) for the five 10-node networks (Fig. 6a) and five 100-node networks (Fig. 6b), from DC4. See also SI (Supplementary Table 1) for standard deviation results.

Figure 7 shows the mean of the scores (Euclidean distance) in function of "p" for the three sets of five networks of 30, 60, and 100 nodes generated by FRANK (TF = 30, 60, or 100 and TA = 0) at two noise levels (medium: $k = 0.1$ and strong: $k = 0.5$, as described in Methods).

For another set of 15 networks generated by FRANK (60, 100, and 200 nodes, TF = TA), results were similar (see SI Fig. S3).

These results underscore the importance of incorporating prior knowledge and the consequent substantial performance enhancement. Additionally, the very small standard deviation values indicated that this improvement predominantly relies on the percentage of known values rather than on their specific location.

Conclusion

Understanding biological systems often involves the meticulous inference of interaction networks. This prompted the development of various methods, including MRARegress. MRARegress was introduced to efficiently address inference in the presence of noise, which is inevitable, and for networks larger than 10 nodes. In this study, we described and validated strategies to overcome some limitations of MRA, such as the prerequisite of perturbation independence or the near-linearity of the function that describes the system dynamics. We also highlighted the usefulness of polynomial regression in conditions of low noise and nonlinear behavior and the substantial benefit provided by prior knowledge about the studied network, when available.

All these advancements have been integrated into our MRARegress software package. This package allows:

- The application of different methods (e.g. ARACNE, LSE, TLR, LASSO, STEP, random forest) to analyze measurements,
- The optimal consideration of over-determined systems and replicates,
- The accommodation of non-independent perturbations (subject to system rank) to incorporate different experiment designs,
- The calculation of the 95% confidence interval for the computed connectivity coefficients,
- ANOVA to identify nodes that are unsuitable for linear regression and to what extent,
- The use of polynomial regression (order 2) to compute synergy-related coefficients alongside connectivity coefficients,
- The incorporation of any prior knowledge on the network under study.

The research landscape surrounding MRA continues to evolve. Future endeavors may include automating program hyperparameters, integrating additional artificial intelligence–based processing functionalities, preprocessing measurements based on their noise characteristics, and analyzing networks with periodic or pseudo-periodic behavior while considering the measurement changes over time. MRARegress is an open-source software and this should allow integrating these potential advancements.

Key Points

- MRA, integrated with linear regression ("MRARegress"), is an effective method for inferring biological networks, including noisy and large networks.
- Regression helps to overcome some MRA limitations, such as perturbation independence. The ANOVA and LOF tests identify the primary source of errors (measurement noise or nonlinearity). If nonlinearity prevails, a second-order polynomial regression significantly improves the results.
- MRARegress facilitates the use of prior knowledge on the studied network. It allows reducing almost linearly the estimation error of network connectivity coefficients.
- These accomplishments are implemented in a software package, also named "MRARegress," developed in R and freely available to the scientific community.

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Conflict of interest: None declared.

Funding

None declared.

Data availability

The source code and the unit tests data are available online (GNU General Public License v3.0 license; https://github.com/J-P-BORG/MRARegress). The files containing all calls to MRARegress and its modules and the corresponding parameters (AdvancedMRA.R) and all the data files used to obtain the results in this article (tables, figures) are also available online at https://github.com/J-P-BORG/MRA or https://github.com/J-P-BORG/MRARegress, within the data folder.

References

1. Mekedem M, Ravel P, Colinge J. Application of modular response analysis to medium-to large-size biological systems. *PLoS Comput Biol* 2022;**18**:e1009312. https://doi.org/10.1371/journal.pcbi.1009312
2. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter, P. Molecular Biology of the Cell (4th ed.). Garland Science. 2002.
3. Margolin AA, Nemenman I, Basso K. *et al.* ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 2006;**7**:1–15. https://doi.org/10.1186/1471-2105-7-S1-S7
4. Faith JJ, Hayete B, Thaden JT. *et al.* Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol* 2007;**5**:e8. https://doi.org/10.1371/journal.pbio.0050008
5. Meyer PE, Kontos K, Lafitte F. *et al.* Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J Bioinform Syst Biol* 2007;**2007**:1–9. https://doi.org/10.1155/2007/79879
6. Schaffter T, Marbach D, Floreano D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* 2011;**27**:2263–70. https://doi.org/10.1093/bioinformatics/btr373
7. Friedman N, Koller D. Being Bayesian about network structure. *Mach Learn* 2003;**50**:95–125. https://doi.org/10.1023/A:1020249912095
8. Hill SM, Lu Y, Molina J. *et al.* Bayesian inference of Signaling network topology in a cancer cell line. *Bioinformatics* 2012;**28**:2804–10. https://doi.org/10.1093/bioinformatics/bts514
9. Haury A-C, Mordelet F, Vera-Licona P. *et al.* TIGRESS: trustful inference of gene REgulation using stability selection. *BMC Syst Biol* 2012;**6**:145. https://doi.org/10.1186/1752-0509-6-145
10. Huynh-Thu VA, Irrthum A, Wehenkel L. *et al.* Inferring regulatory networks from expression data using tree-based methods. *PloS One* 2010;**5**:e12776. https://doi.org/10.1371/journal.pone.0012776
11. Brouard C. Inference of protein-protein interaction networks through statistical learning. PhD thesis, University of Évry-Val d'Essonne, France. 2013.
12. Cherif A, Cardot H, Boné R. Hierarchical Clustering for Local Time Series Forecasting. In Lee M, Hirose A, Hou Z-G, Kil RM. (eds). *Neural Information Processing*, Springer: Berlin, Heidelberg, 2013; pp 59–66. https://doi.org/10.1007/978-3-642-42042-9_8
13. Kholodenko BN, Kiyatkin A, Bruggeman FJ. *et al.* Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proc Natl Acad Sci* 2002;**99**:12841–6. https://doi.org/10.1073/pnas.192442699
14. Klinger B, Blüthgen N. Reverse engineering gene regulatory networks by modular response analysis–a benchmark. *Essays Biochem* 2018;**62**:535–47. https://doi.org/10.1042/EBC20180012
15. Bosdriesz E, Prahallad A, Klinger B. *et al.* Comparative network reconstruction using mixed integer programming. *Bioinformatics* 2018;**34**:i997–1004. https://doi.org/10.1093/bioinformatics/bty616
16. Andrec M, Kholodenko BN, Levy RM. *et al.* Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy. *J Theor Biol* 2005;**232**:427–41. https://doi.org/10.1016/j.jtbi.2004.08.022
17. Santos SD, Verveer PJ, Bastiaens PI. Growth factor-induced MAPK network topology shapes Erk response determining PC-12 cell fate. *Nat Cell Biol* 2007;**9**:324–30. https://doi.org/10.1038/ncb1543
18. Santra T, Kolch W, Kholodenko BN. Integrating Bayesian variable selection with modular response analysis to infer biochemical network topology. *BMC Syst Biol* 2013;**7**:57–19. https://doi.org/10.1186/1752-0509-7-57
19. Santra T, Rukhlenko O, Zhernovkov V. *et al.* Reconstructing static and dynamic models of signaling pathways using modular response analysis. *Curr Opin Syst Biol* 2018;**9**:11–21. https://doi.org/10.1016/j.coisb.2018.02.003
20. Szklarczyk D, Franceschini A, Kuhn M. *et al.* The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res* 2011;**39**:D561–8. https://doi.org/10.1093/nar/gkq973
21. Borg J-P, Colinge J, Ravel P. Modular response analysis reformulated as a multilinear regression problem. *Bioinformatics* 2023;**39**:btad166. https://doi.org/10.1093/bioinformatics/btad166
22. Thomaseth C, Fey D, Santra T. *et al.* Impact of measurement noise, experimental design, and estimation methods on modular response analysis based network reconstruction. *Sci Rep* 2018;**8**:16217. https://doi.org/10.1038/s41598-018-34353-3
23. Couty R, Ezra J, Analyse MP. *Deuxième Année et Spéciales AA', Ed. Armand Colin, Collection U* T.2, France, 1970.

24. Jimenez-Dominguez G, Ravel P, Jalaguier S. *et al*. An R package for generic modular response analysis and its application to estrogen and retinoic acid receptor crosstalk. *Sci Rep* 2021;**11**: 1–14. https://doi.org/10.1038/s41598-021-86544-0
25. Saporta G. Probabilités, analyse des données et statistique; Editions TECHNIP, 2011.
26. Mason RL, Gunst RF, Hess JL. Statistical Design and Analysis of Experiments: With Applications to Engineering and Science (2e éd.). Hoboken, NJ: Wiley-Interscience. 2003. https://doi.org/10.1002/0471458503
27. Carré C, Mas A, Krouk G. Reverse engineering highlights potential principles of large gene regulatory network design and learning. *NPJ Syst Biol Appl* 2017;**3**:17. https://doi.org/10.1038/s41540-017-0019-y
28. Boyd S, Vandenberghe L. Convex Optimization. Cambridge University Press. 2004. https://doi.org/10.1017/CBO9780511804441

Briefings in Bioinformatics, 2025, **26(2)**, bbaf098
https://doi.org/10.1093/bib/bbaf098
Problem Solving Protocol