

SOFTWARE

Open Access

MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics

Jose A Egea¹, David Henriques^{2,3}, Thomas Cokelaer³, Alejandro F Villaverde², Aidan MacNamara³, Diana-Patricia Danciu³, Julio R Banga^{2*} and Julio Saez-Rodriguez^{3*}

Abstract

Background: Optimization is the key to solving many problems in computational biology. Global optimization methods, which provide a robust methodology, and metaheuristics in particular have proven to be the most efficient methods for many applications. Despite their utility, there is a limited availability of metaheuristic tools.

Results: We present MEIGO, an R and Matlab optimization toolbox (also available in Python via a wrapper of the R version), that implements metaheuristics capable of solving diverse problems arising in systems biology and bioinformatics. The toolbox includes the enhanced scatter search method (eSS) for continuous nonlinear programming (cNLP) and mixed-integer programming (MINLP) problems, and variable neighborhood search (VNS) for Integer Programming (IP) problems. Additionally, the R version includes BayesFit for parameter estimation by Bayesian inference. The eSS and VNS methods can be run on a single-thread or in parallel using a cooperative strategy. The code is supplied under GPLv3 and is available at <http://www.iim.csic.es/~gingproc/meigo.html>. Documentation and examples are included. The R package has been submitted to BioConductor. We evaluate MEIGO against optimization benchmarks, and illustrate its applicability to a series of case studies in bioinformatics and systems biology where it outperforms other state-of-the-art methods.

Conclusions: MEIGO provides a free, open-source platform for optimization that can be applied to multiple domains of systems biology and bioinformatics. It includes efficient state of the art metaheuristics, and its open and modular structure allows the addition of further methods.

Background

Mathematical optimization plays a key role in systematic decision making processes, and is used virtually in all areas of science and technology where problems can be stated as finding the best among a set of feasible solutions. In bioinformatics and systems biology, there has been a plethora of successful applications of optimization during the last two decades (see reviews in [1-5]). Many problems in computational biology can be formulated as IP problems, such as sequence alignment, genome rearrangement and protein structure prediction problems [1,3], or the

design of synthetic biological networks [6]. Deterministic and stochastic/heuristic methods have been extensively applied to optimization problems in the area of machine learning [2]. In addition to combinatorial optimization, other important classes of optimization problems that have been extensively considered, especially in systems biology, are cNLP and mixed-integer dynamic optimization. Such problems arise in parameter estimation and optimal experimental design [5,7].

A number of authors have stressed the need to use suitable global optimization methods due to the non-convex (multimodal) nature of many of these problems [4,8,9]. Roughly speaking, global optimization methods can be classified into exact and stochastic approaches. Exact methods can guarantee convergence to global optimality, but the associated computational effort is usually prohibitive for realistic applications. In contrast,

*Correspondence: julio@iim.csic.es; saezrodriguez@ebi.ac.uk

²(Bio)Process Engineering Group, Spanish National Research Council, IIM-CSIC, 36208 Vigo, Spain

³European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Cambridge CB10 1SD, UK

Full list of author information is available at the end of the article

stochastic methods are often able to locate the vicinity of the global solution in reasonable computation times, but without guarantees of convergence. Metaheuristics (i.e. guided heuristics) are a particular class of stochastic methods that have been shown to perform very well in a broad range of applications [5].

Motivated by this, we developed the software suite MEIGO (MEtaheuristics for systems biology and bioinformatics Global Optimization) which provides state of the art metaheuristics (eSS and VNS) in open-source R (here with the addition of the Bayesian inference method BayesFit) and Matlab versions (it is also available in Python via a wrapper for the R version). MEIGO covers the most important classes of problems, namely (i) problems with real-valued (cNLPs) and mixed-integer decision variables (MINLPs), and (ii) problems with integer and binary decision variables (IPs). Furthermore, MEIGO allows the user to apply parallel computation using cooperative strategies [10]. MEIGO can optimize arbitrary objective functions that are handled as black-boxes. Thus, it is applicable to optimize complex systems that may involve solving inner problems (e.g. simulations or even other optimization problems) to obtain explicit values for the objective function and/or the possible constraints. For example, CellNOpt [11], SBToolbox [12], AMIGO [13] and Potterswheel [14] use eSS for dynamic model calibration. Some recent successful applications of eSS in the field of systems biology can be found in [15-26]. It has also been shown that eSS outperformed the various optimization methods available in the Systems Biology Toolbox [27].

Methods

Enhanced Scatter Search (eSS)

Scatter search [28] is a population-based metaheuristic which can be classified as an evolutionary optimization method. In contrast with other popular population-based metaheuristics like, for example, genetic algorithms, the population size, N , in scatter search is small, and the combinations among its members are performed systematically, rather than randomly. The current population is commonly named the “Reference Set” (RefSet). The *improvement method*, which consists of a local search to increase the convergence to optimal solutions, can be applied with more or less frequency to the members of this RefSet. A set of improvements has been implemented in the enhanced scatter search method. Among the most remarkable changes, we can mention the replacement method. Unlike in the original scatter search scheme, which uses a $\mu + \lambda$ replacement (i.e. the new population or RefSet will consist in the best N solutions selected from the previous RefSet members and the new *offspring* solutions), the enhanced scatter search uses a $1 + 1$ replacement, similar to the strategy used in a very

efficient evolutionary method, Differential Evolution [29]. This means that a RefSet member can only be replaced by a solution that has been generated combining by the former and another RefSet member. In other words, an *offspring* solution can only replace the RefSet member that generated it, and not any other. This strategy enhances diversity and prevents the search from premature stagnation by not allowing too similar solutions to be present in the RefSet at the same time. The “go-beyond” strategy to exploit combinations which explore promising directions has also been implemented. This strategy analyzes the search directions defined by a RefSet member and their *offspring*. If an *offspring* solution outperforms its corresponding RefSet member (i.e. the RefSet member that generates it), then the method considers that the explored direction is promising and a new solution is generated within such direction, exploring an area beyond the segment defined by the RefSet member and its *offspring* solution. The process is repeated until the new generated solutions do not outperform the previous ones and it favours intensification in the current iteration. Additionally, the use of memory is also exploited to select the most efficient initial points to perform local searches, to avoid premature convergence and to perturb solution vectors which are stuck in stationary points. More details about the enhanced scatter search scheme can be found in [30].

Variable Neighbourhood Search (VNS)

Variable Neighbourhood Search is a trajectory-based metaheuristic for global optimization. It was introduced by Mladenović and Hansen [31] and has gained popularity in recent years in the field of global optimization. VNS performs a local search by evaluating the objective function around an incumbent solution and repeats the procedure visiting different neighbourhoods to locate different local optima, among which the global optimum is expected to be found. One of the key points of the algorithm is the strategy followed to change the current neighbourhood. VNS usually seeks a new neighbourhood by perturbing a set of decision variables using a distance criterion. Once a new solution has been created in the new neighbourhood, a new local search is performed. The typical scheme consists of visiting neighbourhoods close to the current one (i.e. perturbing a small set of solutions), until no further improvement is achieved. Then, more distant neighbourhoods are explored. Apart from this basic scheme, we have implemented advanced strategies to avoid cycles in the search (e.g. not repeating the perturbed decision variables in consecutive neighbourhood searches) in order to increase the efficiency when dealing with large-scale problems (e.g. by allowing a maximum number of perturbed decision variables, like in the Variable Neighbourhood Decomposition Search strategy [32]). We have also modified the search aggressiveness

to locate high quality solutions (even if they are not the global optimum) in short computational times if required. Other heuristics, like the “go-beyond” strategy (explained above), that is used to exploit promising directions during the local search, have been adapted from other meta-heuristics for continuous optimization [30].

BayesFit

BayesFit is a Bayesian inference method for parameter estimation that uses Markov Chain Monte Carlo (MCMC) to sample the complete probability distributions of parameters. This accounts for both experimental error and model non-identifiability. It is available in the R version of MEIGO and has been adapted from the Python package BayesSB [33]. The sampling of the probability distributions uses a multi-start MCMC algorithm where the number of visits to a position in the parameter space is proportional to the posterior probability. The MCMC walk is punctuated by a Metropolis Hastings (M-H) criterion that allows more distant neighbourhoods to be explored, based on a probabilistic calculation.

Cooperation

The cooperation scheme implemented in MEIGO is based on the following idea: to run, in parallel, several implementations or threads of an optimization algorithm, which may have different settings and/or random initializations, and exchange information between them. Since the nature of the optimization algorithms implemented in MEIGO is essentially different, we distinguish between eSS (the population based method) and VNS (the trajectory based method), following the classification proposed in [34] (currently there is no cooperation scheme for BayesFit):

1. Information available for sharing: the best solution found and, optionally for eSS, the *RefSet*, which contains information about the diversity of solutions.
2. Threads that share information: all of them.
3. Frequency of information sharing: the threads exchange information at a fixed interval τ .
4. Number of concurrent programs: η .

Each of the η threads has a fixed degree of aggressiveness. “Conservative” threads have an emphasis on diversification (global search) and are used to increase the probability of finding a feasible solution, even if the parameter space is rugged or weakly structured. “Aggressive” threads have an emphasis on intensification (local search) and they speed up the calculations in smoother areas. Communication, which takes place at fixed time intervals, enables each thread to benefit from the knowledge gathered by the others. Thus this strategy has several degrees of freedom that have to be fixed: the time between

communication (τ), the number of threads (η), and the strategy adopted by each thread. These adjustments should be chosen carefully depending on the particular problem we want to solve. Some guidelines for doing this can be found in [10] and in the Additional files 1, 2, 3, 4 and 5 accompanying this paper.

Implementation

MEIGO runs on Windows, Mac, and Linux, and provides implementations in both Matlab and R. So far, MEIGO includes: (i) eSS (Enhanced Scatter Search, [30]), for solving cNLP and MINLP problems, and (ii) VNS (Variable Neighbourhood Search), following the implementation described in [35], to solve IP problems (see Figure 1). The R version of MEIGO also includes the Bayesian parameter inference method BayesFit. Cooperative parallel versions (CeSS, CVNS), which can run on multicore PCs or clusters, are also included. Cooperation enhances the efficiency of the methods, not only in terms of speed, but also in terms of range: the threads running in parallel are completely independent so they can be customized to cover a wide range of search options, from aggressive to robust. In a sense the cooperation, as it has been designed, acts as a combination of different metaheuristics since each of the threads may present a different search profile. Four different kernel functions per method are included depending on the programming language chosen and the parallelization capabilities. Parallel computation in Matlab is

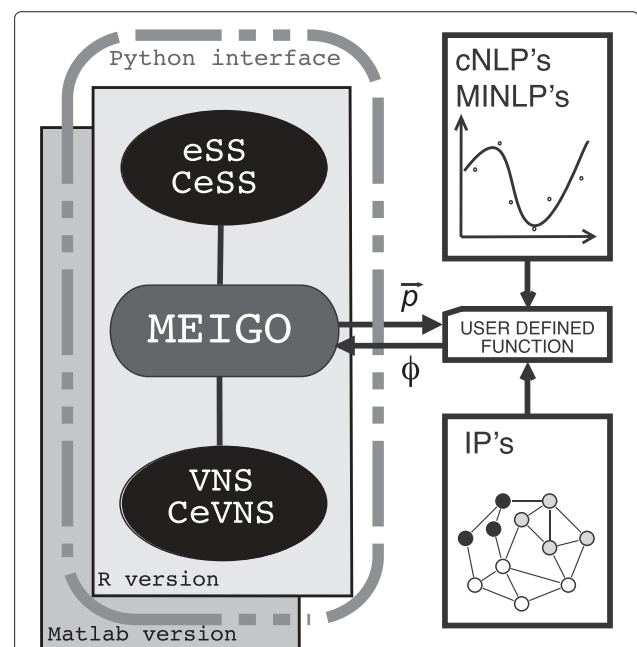


Figure 1 MEIGO workflow. Figure depicting the global structure of MEIGO.

carried out making use of the jpar tool [36]. Parallel computation in R can be performed using the package snowfall [37].

The methods implemented in MEIGO consider the objective functions to be optimized as black-boxes, with no requirements with respect to their structure. The user must provide a function that can be externally called for evaluation, accepting as input the variables to be estimated, and providing as output the objective value, ϕ , as a function of the input parameters. For constrained problems, the values of the constraints are also provided as output so that penalization functions can be calculated. For eSS and VNS, the user must define a set of compulsory fields (e.g. the name of the objective function, the bounds in the parameters, the maximum number of function evaluations). Further options take default values or can be changed. After each optimization, all the necessary results are stored in data files for further analysis with the tools provided by the host platforms. BayesFit is similarly robust to the form of the problem; in this case the likelihood function is provided by the user and this is incorporated into the calculation for the posterior probability for the parameter set, given the data.

Importantly, MEIGO is an open optimization platform in which other optimization methods can be implemented regardless of their nature (e.g. exact, heuristic, probabilistic, single-trajectory, population-based, etc.).

Illustrative examples

To illustrate the capabilities of the methods presented here, a set of optimization problems, including cases from systems biology and bioinformatics, have been solved and are presented as case studies. The examples include (i) a set of state of the art benchmark cases for global optimization (from the Competition on Large Scale Global Optimization, 2012 IEEE World Congress on Computational Intelligence), (ii) a metabolic engineering problem based on a constraint-based model of *E. coli*, (iii) training of logic models of signaling networks to phosphoproteomic data [38], and (iv) an additional toy logic model [22] to compare BayesFit to eSS. The corresponding code for these examples is included in the distribution of the MEIGO software.

Large-scale continuous global optimization benchmark

These are benchmark functions used in the Special Session on Evolutionary Computation for Large Scale Global Optimization, which was part of the 2012 IEEE World Congress on Computational Intelligence (CEC@WCCI-2012). These objective functions can be regarded as state-of-the-art benchmark functions to test numerical methods for large-scale (continuous) optimization. Information about the functions as well as computer codes can be downloaded from <http://staff.ustc.edu.cn/~ketang/>

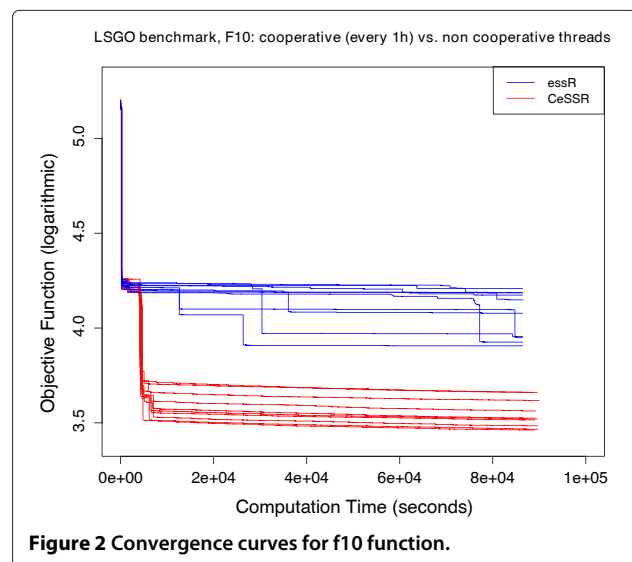
http://staff.ustc.edu.cn/~ketang/cec2012/lsgo_competition.htm. Some of these functions were previously solved in [10] using CeSS, a cooperative version of the Enhanced Scatter Search metaheuristic implemented in Matlab and available within MEIGO. Large-scale calibration of systems biology models were also presented and solved in that paper. Here we present the solution of 3 of these functions (i.e. **f10**, **f17** and **f20**) using the R version of CeSS used by MEIGO. The convergence curves for the solution of these benchmark functions in R are coherent with those presented in [10], which were solved with Matlab, and the results are also competitive with the reference results for these functions presented in http://staff.ustc.edu.cn/~ketang/cec2012/lsgo_competition.htm. The convergence curves corresponding to these results are presented in Figures 2, 3 and 4.

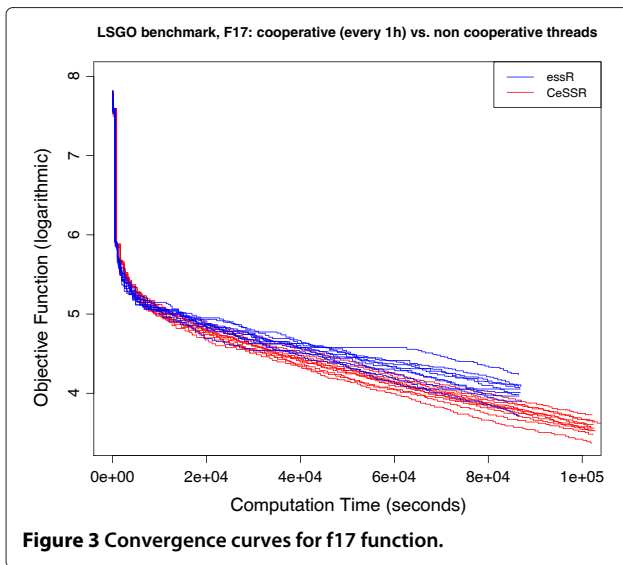
Integer optimization benchmark problems

A set of integer optimization problems arising in process engineering and coded in AMPL (*A Modeling Language for Mathematical Programming*) were solved using the Matlab version of VNS and making use of the AMPL-Matlab interface files provided by Dr. Sven Leyffer, available at <http://www.mcs.anl.gov/~leyffer/macminlp/>. VNS solved all the problems and, in some cases, achieved a better solution than the best reported one. A summary of the tested problems is presented in Table 1. These benchmarks have been solved using the Matlab version of MEIGO under Windows only, since the dynamic library to access AMPL files runs on Windows.

Metabolic engineering example

In this section we illustrate the application of the VNS algorithm to a metabolic engineering problem. Here VNS





was used to find a set of potential gene knock-outs that will maximize the production of a given metabolite of interest. The objective function is given by flux-balance analysis (FBA) where a steady-state model is simulated by means of linear programming (LP). The mathematical formulation is similar to that presented in [41]. FBA assumes that cells have a biological objective that is often considered as growth rate maximization, minimization of ATP consumption or both.

In this example we considered a small steady-state model from *E. coli* central carbon metabolism, available at <http://gcrucg.ucsd.edu/Downloads/EcoliCore>. Here the metabolite of interest is succinate and we considered the biological objective as biomass maximization. To solve the inner FBA problem we used openCOBRA ([LSGO benchmark, F20: cooperative \(every 1h\) vs. non cooperative threads

Figure 4 Convergence curves for f20 function.](http://</p>
</div>
<div data-bbox=)

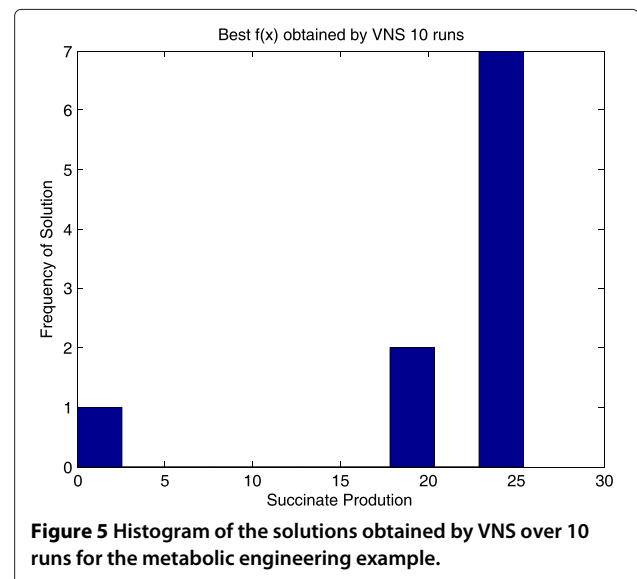
Table 1 Summary of solutions for integer programming problems

Name	nvar	Ref.	Best reported solution	Best VNS solution
geartrain	4	[39]	7.78e-7	2.70e-12
mittelman	16	-	13.0	13.0
trimlon2	8		5.3	5.3
trimlon4	24	[40]	11.3	8.3
trimlon5	35		12.1	10.6

Data in boldface: solutions outperforming the best reported solution for that problem.

opencobra.sourceforge.net/) with Gurobi as an LP solver (<http://www.gurobi.com/>). For the problem encoding, 5 integer variables were chosen as decision variables, one for each possible gene knock-out. Each of these variables was allowed to vary from 0 (no knock-out) to 52, the total number of possible genes to be knocked-out. Repeated KOs were filtered by the objective function.

Additionally we also implemented and solved the problem with a genetic algorithm from the Matlab Global Optimization Toolbox. The point here was to cross-check the VNS results, not to perform an extensive comparison between the performances of GA and VNS. However we found that for our particular problem and encoding, VNS achieved the optimal solution more often (see Figures 5 and 6). The Wilcoxon rank sum test with continuity correction for comparing means provides a p-value of 0.068 (or 0.021 if we remove the outlier VNS solution) showing that the solutions provided by VNS are significantly better. Please note that the GA was used out of the box (with default settings). Results can vary when using other



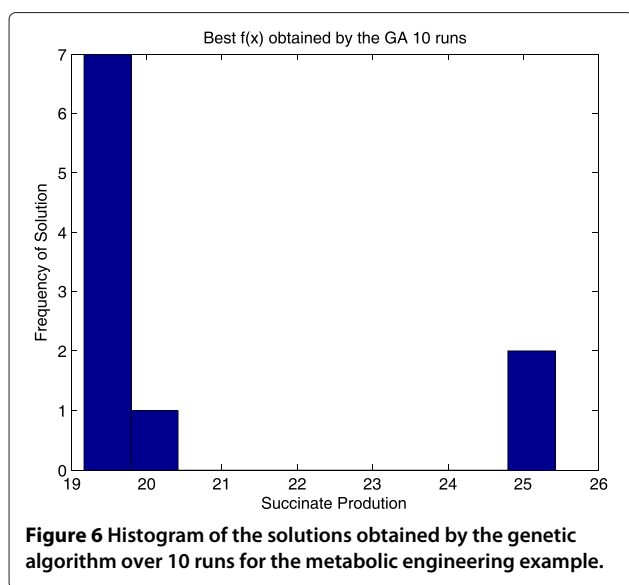


Figure 6 Histogram of the solutions obtained by the genetic algorithm over 10 runs for the metabolic engineering example.

encodings and further tuning of the search parameters. In any case, the purpose was to illustrate how this class of problems can be easily solved using VNS.

Training of logic models of signalling networks to phospho-proteomic data

In this section we compare the performance of variable neighborhood search (VNS) and a discrete genetic algorithm (GA) implementation in training a logic model of a signalling network to phospho-proteomic data [38].

The problem is formulated as follows: one starts from a signed directed graph, containing the prior knowledge about a signaling network of interest. This graph contains directed edges among nodes (typically proteins) and their sign (activating or inhibitory). From this graph, one generates all possible AND and OR gates compatible with the graph. That means, if there are more than one edge arriving at a node, these are combined as OR and AND gates. Mathematically, this is encoded as an hyper graph, where edges with two or more inputs (hyperedges) represent a logical disjunction (AND gate). OR gates are encoded implicitly, by means of edges with only one input arriving at a node. See [38] for details.

To calibrate such models, the authors formulated the inference problem as a binary multi-objective problem, where the first objective corresponded to how well the model described the experimental data and the second consisted of a complexity penalty to avoid over-fitting:

$$\theta(P) = \theta_f(P) + \alpha \cdot \theta_s(P) \quad (1)$$

where $\theta_f(P) = \frac{1}{n_E} \sum_{k=1}^s \sum_{l=1}^m \sum_{t=1}^n (B_{k,l,t}^M(P) - B_{k,l,t}^E)^2$ and $\theta_s(P) = \frac{1}{v_e} \sum_{e=1}^r v_e P_e$ such that $B_{k,l,t}(P) \in \{0, 1\}$ is the value (0 or 1) as predicted by computation of the model's

logical steady state [42] and $B_{k,l,t}^E \in [0, 1]$ is the data value for readout l at time t under the k th experimental condition. $\theta_f(P)$ is the mean squared error and $\alpha \cdot \theta_s(P)$ is the product between a tunable parameter α and a function denoting the model complexity (each hyper edge receives a penalty proportional to the number of inputs. E.g. an AND gates with 3 inputs is penalised 3 times as a single edge. OR gates arise implicitly from the combination of single input edges.).

Noticeably, the binary implementation of this problem contains redundant solutions in the search space. This can be addressed by compressing the search space into a reduced set containing only the smallest non-redundant combinations of hyperedges [38] (equivalent to the Sperner hypergraph). By doing this, the problem is transformed from a binary to an integer programming problem that was solved in [38] using a genetic algorithm.

Here, we implemented this benchmark by using the Matlab version of CellNetOptimizer (CNO or CellNOpt, available at <http://www.cellnopt.org/downloads.html>). The prior-knowledge network and data-set are also publicly available and thoroughly described at <http://www.ebi.ac.uk/~cokelaer/cellnopt/data/ExtLiverPCB.html>.

In order to assess the performance of both algorithms we solved each problem 100 times using VNS and the GA implementation from CNO. In the allowed time budget, VNS returned solutions that were on average better than those found by the GA (see Figure 7). The Welch Two Sample t-test for comparing means provides a p-value of $3.5 \cdot 10^{-14}$, which clearly shows that VNS outperforms GA for this problem. Since both methods are sensitive to the tuning parameters, we tried to tune both algorithms fairly. Also, we note that the solution of this problem in

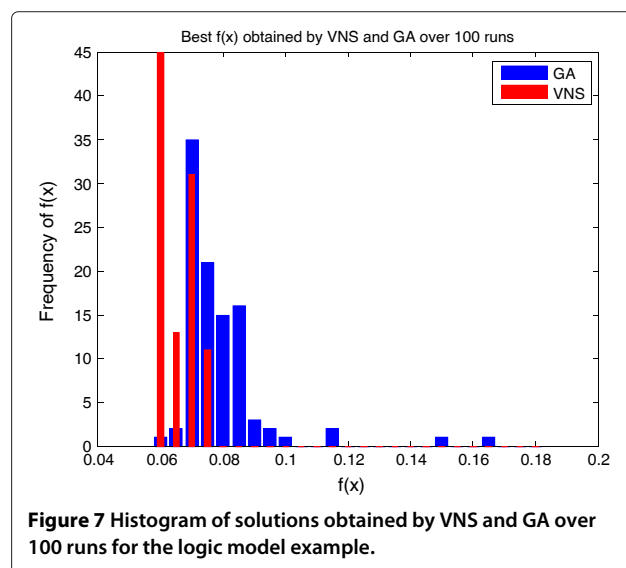


Figure 7 Histogram of solutions obtained by VNS and GA over 100 runs for the logic model example.

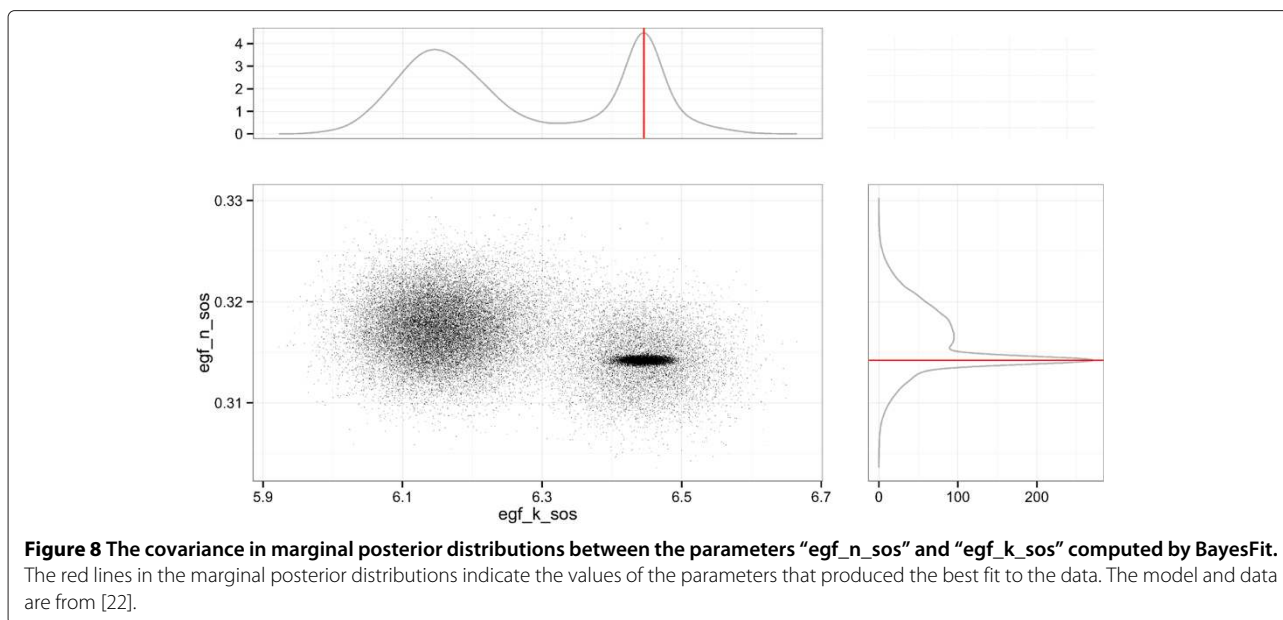


Figure 8 The covariance in marginal posterior distributions between the parameters “egf_n_sos” and “egf_k_sos” computed by BayesFit. The red lines in the marginal posterior distributions indicate the values of the parameters that produced the best fit to the data. The model and data are from [22].

its original, binary implementation can be solved using deterministic methods based either on Integer Linear Programming [43,44] or Answer Set Programming [45].

Training of logic ODEs to in-silico generated data

Here, in order to demonstrate the additional information that can be derived from the probability distributions of optimized parameters, we compared the R implementations of BayesFit and eSS. The problem is again based on a logic model where, this time, the topology of the model is known and the goal is to optimize the parameters of the transfer functions used to generate a continuous simulation of the model. The parameters were optimized to reduce the distance between the model simulation and in silico generated data. This example is as described in section 6 from [22]; the only difference is that the model used here is the compressed model used to generate the in silico data in [22]. BayesFit produced a good fit to the data, comparable to that of eSS (Mean Squared Error: BayesFit, 0.007; eSS, 0.005). One of the advantages of estimating parameters by Bayesian inference is that parameter identifiability can be deduced from the marginal distributions for each parameter. For example Figure 8 shows two parameters of a single interaction between the species “egf” and “sos” in the model; these parameters n and k , control the shape of the transfer function between the two species [22]. From this figure, the covariation between the 2 parameters is evident. The best fit parameters (red line) lie in one region of high probability. However, there are additional correlated peaks in the marginal distributions of the two parameters, which suggests different parameter values could also produce a strong fit to the data.

Conclusions

Here, we present MEIGO, a free, open-source and flexible package to perform global optimization in R, Matlab, and Python. It includes advanced metaheuristic methods. Furthermore, its modular nature (Figure 1), enables the connection to existing optimization methods.

Availability and requirements

Project name: Metaheuristics for global optimization in systems biology and bioinformatics (MEIGO)

Project home page: <http://www.iim.csic.es/~gingproc/meigo.html>

Operating system(s): Windows, Linux, Mac OS X

Programming language: Matlab 7.5 or higher and R 2.15 or higher

Licence: GPLv3

Additional files

Additional file 1: MEIGO - Matlab’s users manual. The file includes the user’s manual of the Matlab version of MEIGO.

Additional file 2: MEIGO - R’s user’s manual. The file includes the users manual of the R version of MEIGO.

Additional file 3: Test of options for the integer programming benchmark problems. The file includes the test carried out with the different search options to solve the integer programming benchmark problems as well as the extracted conclusions.

Additional file 4: MEIGO Matlab version source code and examples. The file includes the source code of the Matlab version of MEIGO and the examples included in the users manual.

Additional file 5: MEIGO R version source code and examples. The file includes the source code of the R version of MEIGO and the examples included in the users manual.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

JSR and JRB conceived, designed and coordinated the project. JAE implemented the metaheuristic methods included in MEIGO. AFV and DH developed and implemented the methods for parallel computation. JAE, AFV and DH performed the numerical computations for the metaheuristic methods. AMN and DPD implemented and performed the computational test of the bayesian method. TC designed and developed the Python wrapper and helped with many technical aspects. All authors contributed to the writing of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The authors thank Alexandra Vatsiou for testing MEIGO. We acknowledge the funding received from the EU through projects "BioPreDyn" (FP7-KBBE grant 289434) and "NICHE" (FP7-ITN grant 289384), from Ministerio de Economía y Competitividad (and the FEDER) through the projects "Multi-Scales" (DPI2011-28112-C04-03 and DPI2011-28112-C04-04), and from the CSIC intramural project "BioREDES" (PIE-201170E018).

Author details

¹Department of Applied Mathematics and Statistics, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain. ²(Bio)Process Engineering Group, Spanish National Research Council, IIM-CSIC, 36208 Vigo, Spain. ³European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Cambridge CB10 1SD, UK.

Received: 23 October 2013 Accepted: 24 April 2014

Published: 10 May 2014

References

- Greenberg HJ, Hart WE, Lancia G: **Opportunities for combinatorial optimization in computational biology.** *Inform J Computing* 2004, **16**(3):211–231.
- Larrañaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, Lozano JA, Armañanzas R, Santafé G, Pérez A, Robles V: **Machine learning in bioinformatics.** *Brief Bioinform* 2006, **7**:86–112.
- Festa P: **On some optimization problems in molecular biology.** *Math Biosci* 2007, **207**(2):219–234.
- Banga JR: **Optimization in computational systems biology.** *BMC Syst Biol* 2008, **2**:47.
- Sun J, Garibaldi JM, Hodgman C: **Parameter estimation using Metaheuristics in systems biology: a comprehensive review.** *IEEE/ACM Trans Comput Biol Bioinform* 2012, **9**:185–202.
- Marchisio M: **Stelling J: Computational design tools for synthetic biology.** *Curr Opin Biotechnol* 2009, **20**(4):479–85.
- Banga JR, Balsa-Canto E: **Parameter estimation and optimal experimental design.** *Essays Biochem* 2008, **45**:195.
- Moles CG, Mendes P, Banga JR: **Parameter estimation in biochemical pathways: a comparison of global optimization methods.** *Genome Res* 2003, **13**(11):2467–2474.
- Ashyraliyev M, Fomekong-Nanfack Y, Kaandorp JA, Blom JG: **Systems biology: parameter estimation for biochemical models.** *FEBS J* 2008, **276**(4):886–902.
- Villaverde AF, Egea JA, Banga JR: **A cooperative strategy for parameter estimation in large scale systems biology models.** *BMC Syst Biol* 2012, **6**:75.
- Terfve C, Cokelaer T, MacNamara A, Henriques D, Goncalves E, Morris MK, van Iersel M, Lauffenburger DA, Saez-Rodriguez J: **CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms.** *BMC Syst Biol* 2012, **6**:133+.
- Schmidt H, Jirstrand M: **Systems biology toolbox for MATLAB: a computational platform for research in systems biology.** *Bioinformatics* 2006, **22**(4):514–515.
- Balsa-Canto E, Banga JR: **AMIGO, a toolbox for advanced model identification in systems biology using global optimization.** *Bioinformatics* 2011, **27**(16):2311–2313.
- Maiwald T, Eberhardt O, Blumberg J: **Mathematical modeling of biochemical systems with PottersWheel.** In *Computational Modeling of Signaling Networks, Series: Methods in Molecular Biology, Volume 880.* Edited by Liu X. Berrerton MD. New York: Humana Press; 2012:119–138.
- Balsa-Canto E, Alonso A, Banga JR: **An iterative identification procedure for dynamic modeling of biochemical networks.** *BMC Syst Biol* 2010, **4**:11+.
- Skanda D, Lebedez D: **An optimal experimental design approach to model discrimination in dynamic biochemical systems.** *Bioinformatics* 2010, **26**(7):939–945.
- Yuraszcek TM, Neveu P, Rodriguez-Fernandez M, Robinson A, Kosik KS, Doyle FJ III: **Vulnerabilities in the Tau network and the role of ultrasensitive points in Tau pathophysiology.** *PLoS Comput Biol* 2010, **6**(11):e1000997.
- Jia G, Stephanopoulos G, Gunawan R: **Parameter estimation of kinetic models from metabolic profiles: two-phase dynamic decoupling method.** *Bioinformatics* 2011, **27**(14):1964–1970.
- Heldt F, Frensing T, Reichl U: **Modeling the intracellular dynamics of influenza virus replication to understand the control of viral RNA synthesis.** *J Virol* 2012, **86**(15):7806–7817.
- Higuera C, Villaverde AF, Banga JR, Ross J, Morán F: **Multi-criteria optimization of regulation in metabolic networks.** *PLoS ONE* 2012, **7**(7):e41122.
- Jia G, Stephanopoulos G, Gunawan R: **Incremental parameter estimation of kinetic metabolic network models.** *BMC Syst Biol* 2012, **6**:142+.
- MacNamara A, Terfve C, Henriques D, Peñalver-Bernabé B, Saez-Rodriguez J: **State-time spectrum of signal transduction logic models.** *Phys Biol* 2012, **9**(4):045003.
- Sriram K, Rodriguez-Fernandez M, Doyle FJ III: **Modeling cortisol dynamics in the neuro-endocrine axis distinguishes normal, depression, and post-traumatic stress disorder (PTSD) in humans.** *PLoS Comput Biol* 2012, **8**(2):e1002379.
- Sriram K, Rodriguez-Fernandez M, Doyle FJ: **A detailed modular analysis of heat-shock protein dynamics under acute and chronic stress and its implication in anxiety disorders.** *PLoS ONE* 2012, **7**(8):e42958+.
- Freund S, Rath A, Barradas OP, Skerhutt E, Scholz S, Niklas J, Sandig V, Rose T, Heinzele E, Noll T, Pörtner R, Zeng AP, Reichl U: **Batch-to-batch variability of two human designer cell lines – AGE1.HN and AGE1.HN.AAT – carried out by different laboratories under defined culture conditions using a mathematical model.** *Eng Life Sci* 2013, **00**:1–13.
- Glover F, García MR, Middleton RH: **A single compartment model of pacemaking in dissociated Substantia nigra neurons.** *J Comput Neurosci* 2013, **35**(3):295–316.
- Egea JA, Schmidt H, Banga JR: **A new tool for parameter estimation in nonlinear dynamic biological systems using global optimization.** *Poster. 9th International Conference on Systems Biology, ICSB 2008.* Goteborg (Sweden), 22-28 August 2008. Available at http://www.iim.csic.es/gingproc/Poster_ICSB2008.pdf.
- Glover F, Laguna M, Martí R: **Fundamentals of scatter search and path relinking.** *Control Cybernet* 2000, **39**(3):653–684.
- Storn R, Price K: **Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces.** *J Global Optimization* 1997, **11**(4):341–359.
- Egea JA, Martí R, Banga JR: **An evolutionary method for complex-process optimization.** *Comput Oper Res* 2010, **37**(2):315–324.
- Mladenović N, Hansen P: **Variable neighborhood search.** *Comput Oper Res* 1997, **24**:1097–1100.
- Hansen P, Mladenović N, Perez-Brito D: **Variable neighborhood decomposition search.** *J Heuristics* 2001, **7**:335–350.
- Eydgahi H, Chen WW, Muhlich JL, Vitkup D, Tsitsiklis JN, Sorger PK: **Properties of cell death models calibrated and compared using Bayesian approaches.** *Mol Syst Biol* 2013, **9**:644.
- Toulouse M, Crainik TG, Sanso B: **Systemic behavior of cooperative search algorithms.** *Parallel Comput* 2004, **30**:57–79.
- Hansen P, Mladenović N, Moreno-Pérez JA: **Variable neighbourhood search: methods and applications.** *Ann Oper Res* 2010, **175**:367–407.
- Karbowski A, Majchrowski M, Trojaneck P: **jPar—a simple, free and lightweight tool for parallelizing Matlab calculations on multicores**

- and in clusters.** *9th International Workshop on State-of-the-Art in Scientific and Parallel Computing (PARA 2008)*, May 13–16 2008, Trondheim (Norway).
37. Knaus J: **Developing parallel programs using snowfall.** 2010. [<http://cran.r-project.org/web/packages/snowfall/vignettes/snowfall.pdf>]
 38. Saez-Rodriguez J, Alexopoulos LG, Epperlein J, Samaga R, Lauffenburger DA, Klamt S, Sorger PK: **Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction.** *Mol Syst Biol* 2009, **5**:331.
 39. Sandgren E: **Nonlinear integer and discrete programming in mechanical design optimization.** *J Mech Des* 1990, **112**(2):223–229.
 40. Harjunkski I, Westerlund T, Pörn R, Skrifvars H: **Different transformations for solving non-convex trim loss problems by MINLP.** *Eur J Oper Res* 1998, **105**:594–603.
 41. Burgard AP, Pharkya P, Maranas CD: **Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization.** *Biotechnol Bioeng* 2003, **84**(6):647–657.
 42. Klamt S, Saez-Rodriguez J, Lindquist J, Simeoni L, Gilles ED: **A methodology for the structural and functional analysis of signaling and regulatory networks.** *BMC Bioinformatics* 2006, **7**:56.
 43. Mitsos A, Melas IN, Siminelakis P, Chairakaki AD, Saez-Rodriguez J, Alexopoulos LG: **Identifying drug effects via pathway alterations using an integer linear programming optimization formulation on phosphoproteomic data.** *PLoS Comput Biol* 2009, **5**(12):e1000591.
 44. Sharan R, Karp RM: **Reconstructing boolean models of signaling.** In *Proceedings of the 16th Annual International Conference on Research in Computational Molecular Biology, RECOMB'12*. Berlin, Heidelberg: Springer-Verlag; 2012:261–271.
 45. Guziolowski C, Videla S, Eduati F, Thiele S, Cokelaer T, Siegel A, Saez-Rodriguez J: **Exhaustively characterizing feasible logic models of a signaling network using Answer Set Programming.** *Bioinformatics* 2013, **29**(18):2320–2326.

doi:10.1186/1471-2105-15-136

Cite this article as: Egea et al.: MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinformatics* 2014 **15**:136.

**Submit your next manuscript to BioMed Central
and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

