Metabolic pathways synthesis based on ant colony optimization

Matias F. Gerard[1,*], **Georgina Stegmayer**[1], **and Diego H. Milone**[1]

[1]Research Institute for Signals, Systems and Computational Intelligence (sinc(*i*)), FICH–UNL/CONICET, Ciudad Universitaria UNL, (S3000) Santa Fe, Argentina.
[*]mgerard@sinc.unl.edu.ar

ABSTRACT

One of the current challenges in bioinformatics is to discover new ways to transform a set of compounds into specific products. The usual approach is finding the reactions to synthesize a particular product, from a given substrate, by means of classical searching algorithms. However, they have three main limitations: difficulty in handling large amounts of reactions and compounds; absence of a step that verifies the availability of substrates; and inability to find branched pathways. We present here a novel bio-inspired algorithm for synthesizing linear and branched metabolic pathways. It allows relating several compounds simultaneously, ensuring the availability of substrates for every reaction in the solution. Comparisons with classical searching algorithms and other recent metaheuristic approaches show clear advantages of this proposal, fully recovering well-known pathways. Furthermore, solutions found can be analyzed in a simple way through graphical representations on the web.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

Introduction

Nowadays, information of metabolic pathways for a large number of living beings is available in databases such as KEGG[1], MetaCyc[2] and Brenda[3]. This allows the online exploration of the enzymes, biochemical reactions catalyzed, and the involved substrates and products. Although individual rules for producing compounds are well-known, it is still a challenge to identify the adequate sequence of reactions required for the synthesis of several compounds as part of a (novel) complex metabolic network with several branches[4].

Traditionally, metabolic pathway synthesis of a target compound from a given source has been addressed by methods based on graphs. The main reason is to avoid shortcomings of stoichiometric approaches when applied to networks of large size[5]. The first step is to model compounds and reactions as an appropriate graph[6]. In a general approach for modeling, nodes indicate compounds and edges link substrates and products of the same reaction. The next step is searching for a path over the graph, that connects the source with the target compound using some search method. These methods were based mostly on classical Breadth-First Search (BFS) and Depth-First Search (DFS) algorithms[7]. The main problem faced by these methods is avoiding the commonly called *pool compounds*, such as ATP, NAD and water, which are involved in many different reactions carrying out several tasks. Since they have a high connectivity degree, pool compounds are frequently included as intermediate in the solutions found, producing biologically unfeasible pathways.

A systematic approach to deal with pool compounds consists in describing their structures in terms of features vectors[8] or fingerprints[9]. These representations can be used, in combination with a similarity measure, to select the next more similar compound to the current one, thus avoiding pool compounds. Another option is assigning a cost to nodes or links of the graph according to the number of reactions in which each compound participates, and then search for pathways with the lower costs[10,11]. Kotera and co-workers have manually characterized each substrate-product pair on every known reaction according to the fulfilled function[12,13]. Using this characterization, several methods first build graphs without biologically irrelevant connections, and then search for metabolic pathways taking into account only those pairs describing main functions[14,15]. The number of atoms shared between substrates and products of reactions has also been used to avoid pool compounds. Based on this information, some methods search for metabolic pathways that maximize the number of atoms transferred from the source to the target compound[16], or at least preserve a given number of them in the path[17]. An improved version of this approach assigns a cost to the connections in the graph based on structural similarity of the compounds and the thermodynamics of the reaction that involves them[18]. Khosraviani *et al.* have proposed an AND/OR boolean representation of the graph using matrix notation[19]. It allowed search for pathways between source and target compounds over a reduced search space by applying boolean operations. However, these strategies have the limitation of finding solutions only as linear sequences of reactions (or a combination of them), and they do not take into account the availability of compounds already synthesized. In many cases, this

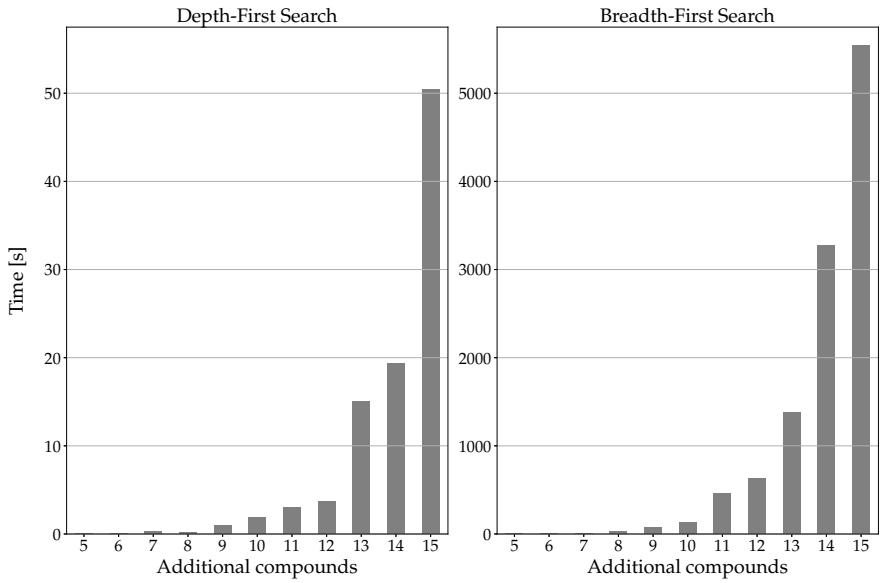

Figure 1. Average time required (100 runs) for DFS (left) and BFS (right) to search for a pathway between 2 compounds. The *x*-axis indicates the number of compounds added to the minimum required set.

leads to the synthesis of metabolic pathways in an uncoordinated fashion, providing solutions without biological sense.

In a previous work, we proposed an algorithm called Evolutionary Metabolic Seeker (EvoMS), which synthesized metabolic pathways using information on the availability of compounds[20,21]. EvoMS models a metabolic pathway as a sequence of feasible reactions that start from a set of initial substrates. In this tool, we proposed the *set-of-compounds* (SoC) model, where a set containing substrates for all reactions in a given sequence is iteratively updated with the products of each new feasible reaction added to the sequence. Thereby, searching for a pathway consist in finding a sequence of feasible reactions that relates a given group of compounds, subject to the availability of initial substrates. However, EvoMS is unable to preserve a set of feasible solutions along the search, since the availability of substrates is not guaranteed when intermediate solutions are combined to produce new ones. As a consequence, the search is very slow and the synthesis of pathways that relate multiple compounds in large search spaces is not always possible.

Synthesizing metabolic pathways has the challenge of exploring a large solution space, which grows when more reactions and compounds are involved. When the SoC model is considered, this problem becomes more difficult and clearly imposes a limitation on the use of classical DFS- and BFS-based algorithms. As frequently happens in real problems, the minimum set of available compounds required for synthesizing a pathway may be not completely known in advance. In consequence, more compounds than necessary are generally added to the initial set, trying to prevent losing potential solutions because some substrates are not available. In order to evaluate how the performance of the algorithms behaves in this context, we designed an experiment where we knew in advance the solution to be found and the minimum set of compounds needed to synthesize it. For this purpose, a list of 79 reactions belonging to the glycolysis was extracted from KEGG, and used to synthesize a pathway to produce 2-phospho-D-glycerate from D-glucose-1-phosphate. Furthermore, the minimum set of compounds required to find the solution was identified. Then, we systematically added a larger number of compounds to increase the size of the set of initially available ones, and run BFS and DFS to explore the search space generated by the SoC model for each initial set of compounds. Figure 1 shows the growth of searching time (average over 100 runs) for BFS and DFS algorithms according to the number of compounds added to the minimum set of initially available ones. As it can be expected, average time grows exponentially with the increase in the size of the available compounds set.

A bio-inspired metaheuristic that has proved to efficiently solve such large graph-based problems is the ant colony optimization algorithm (ACO). The ACO is an important technique in the field of Swarm Intelligence, and it is inspired on the behavior of real ant colonies searching for food[22]. The ants deposit pheromone on the ground in order to mark the routes, from the nest to food, which should be followed by other members of the colony. Accumulation of pheromones over paths along the iterations favor solutions that minimize a cost function[23]. Those algorithms have been successfully applied to a wide range of problems in many different areas[24]. Particularly, they have proved to be a powerful tool solving biological problems related to protein folding[25], genetic interactions detection[26], RNA sequence design[27], protein-protein interaction inhibitors design[28], protein structure optimization[29] and protein-ligand docking[30]. Moreover, they have outperformed genetic algorithms in a wide range of combinatorial optimization problems[31–35].

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

In this work we propose a novel ant-based algorithm to synthesize metabolic pathways, to efficiently explore large search spaces of reactions. Our proposal takes advantage of the way on which ants perform the exploration to incorporate information of the compounds availability, in order to build feasible solutions. Furthermore, since this algorithm uses the SoC model to search, it is possible to find solutions with both linear and branched topology. This algorithm can be suitable for applications such as synthetic biology, interpretation of metabolomics experiments and gap filling in metabolic reconstructions.

Proposed computational method

State space model and metabolic pathways.

Metabolic pathways are networks built by compounds and the biochemical reactions that relate them. These reactions allow the synthesis of new compounds from other ones. Formally, the reactions are described by typical chemical equations as $S(r) \leftrightarrow P(r)$, where $S(r)$ and $P(r)$ correspond to the substrates and the products, respectively[36]. Then, metabolic pathways can be described as sequences of sets of compounds (substrates for next reactions), with composition and size defined by the order on which the reactions of the pathway are performed[44]. Following this reasoning, the state space for the problem of synthesizing metabolic pathways can be build considering each state as a set of compounds and the relations among them. Then, transitions between states are given by those reactions that can be carried out with the available substrates in the current state. It must be noted that while available connections among compounds are known and fixed for a given set of reactions (typical compounds-and-reactions graph), the graph describing the state space changes according to the initial set of available compounds specified. Furthermore, the number of nodes is even larger than for compounds-and-reactions graphs, since each node represents a unique state, which in turn corresponds to a metabolic network in itself (set of compounds and the relations among them). As a result, every path in the search tree built to find a solution on this graph will be feasible, because reactions for which substrates are available in the current node can only be performed.

Figure 2 shows an example of a typical tree to explore the search space. The root node, composed by a set of four compounds without links, is the initial state of the search. When reaction r_1 is applied, for example, a new state with one link and five compounds (triangle added) is reached. Now, applying reaction r_2 over this node, we can reach a new one with a new link and an additional compound. Following this strategy we can reach to a final state describing a metabolic pathway that relates 13 compounds (bottom of the figure).

It is important to highlight here that this approach is clearly different from graph search methods[44], because our proposal does not build first the complete graph and then performs the search within it. Instead, metabolic pathways are grown step by step, by choosing one feasible reaction at a time from a list of available reactions. Then, the chosen reaction is added to a sequence that starts from a set of available compounds. Moreover, choosing reactions is done with a given probability which is learned while solutions are synthesized. Finally, each state explored during the process corresponds to a complete metabolic pathway and not just to a single compound, as typically occurs in with classical searching methods.

Ant-based algorithm for searching metabolic pathways.

Table 1 presents the general steps of the ant-based method which we called PhDSeeker (Pheromone-Directed Seeker). The algorithm receives four inputs: a set $\mathscr{D}$ of compounds to relate; a subset $\mathscr{I} \subseteq \mathscr{D}$ of compounds which can be used as initial substrate for the metabolic pathway; a list $\mathscr{R}$ of reactions that can be used to build the solution; a set $\mathscr{C}$ of freely available compounds (such as water, ATP, etc). As output, it returns a list of reactions $\pi^* = [\pi_0^*, \cdots, \pi_i^*, \cdots, \pi_N^*]$ describing the best feasible metabolic pathway found from the initial conditions specified.

PhDSeeker starts building the list of reactions $\mathbf{r}_0$ that use any compound in $\mathscr{I}$ as substrate. From those reactions, each ant will choose one as the initial reaction of its path. Next, the pheromone matrix Ω is initialized setting $\Omega_{i,j} = 1$. This is used by ants along the search to store the frequency of reactions $i \to j$ in the solutions. For the particular case of $i = 0$, Ω stores information of the usefulness for the reactions in $\mathbf{r}_0$. This value indicates how frequently a given initial reaction leads to a solution.

The searching process begins after $\mathbf{r}_0$ and Ω have been initialized. On each iteration of the algorithm, every ant performs an independent search, following the same steps (lines 7 to 19). Initially, the ant k chooses one reaction r from $\mathbf{r}_0$ according to its probability p_r, which depends on the values stored in the first row of Ω (line 8). Then, it is used to set the first reaction π_0^k of the pathway π^k. In addition, substrates $S(\pi_0)$ and products $P(\pi_0)$ of this reaction are combined with the available compounds $\mathscr{C}$ to build the initial set of compounds C^k that the ant uses to synthesize a pathway linking compounds in $\mathscr{D}$ (line 9).

After the first reaction is inserted into the pathway, the ant repeat five operations (lines 10 to 17). Initially, the ant identifies all the reactions $\mathbf{r}$ for which the substrates are in C^k, filtering those that are already in the pathway π^k. Then, it chooses one reaction $r \in \mathbf{r}$ according to its probability $p_{\pi_i,r}$, and adds the selected reaction to the pathway (lines 14-15). Finally, the ant updates its set of available compounds C^k with products of the selected reaction (line 16). These operations are repeated by every ant until there are no more feasible reactions to synthesize the pathway ($\mathbf{r} = \emptyset$), or a metabolic pathway synthesizing all the final products ($\mathscr{D} \subseteq C$), is found.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

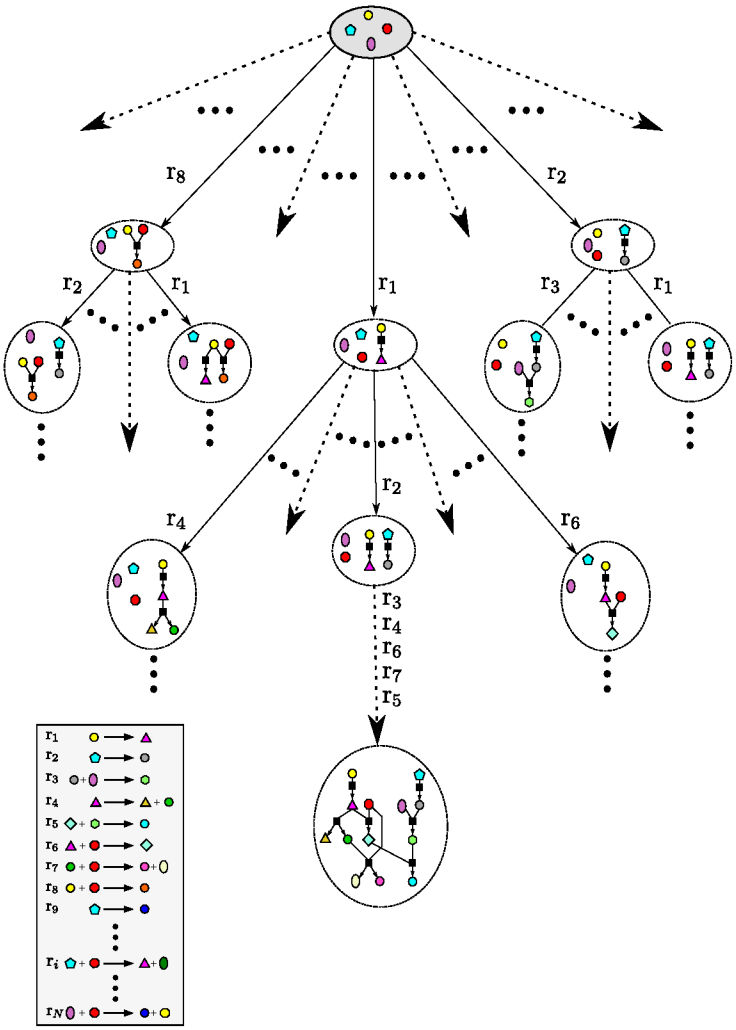

Figure 2. Example of a tree for the exploration of the state space.

Table 1: Ant-based synthesis of metabolic pathways.

Input: a list of compounds to relate $\mathscr{D}$, a list of initial substrates $\mathscr{I} \subseteq \mathscr{D}$, available reactions $\mathscr{R}$, available compounds $\mathscr{C}$.
Output: best pathway found π^*.

1 **begin**
2 Identify initial reactions: $\mathbf{r_0} \leftarrow \{r/r \in \mathscr{R} \wedge S(r) \cap \mathscr{I} \neq \emptyset\}$
3 Initialize usefulness of initial reactions: $\Omega_{0,j} \leftarrow 1 \; \forall \; j \in \mathbf{r_0}$
4 Initialize pheromone matrix: $\Omega_{i,j} \leftarrow 1 \; \forall \; i,j \in \mathscr{R} \wedge i \neq j$
5 **while** *best solution changes* ***and*** *not all ants follow the same path* **do**
6 **foreach** *ant k* **do**
7 Choose $r \in \mathbf{r_0}$ according to: $p_r = \Omega_{0,r} / \sum_{\forall j \in \mathbf{r_0}} \Omega_{0,j}$
8 Initialize the pathway: $\pi_0^k \leftarrow r$
9 Initialize the set of available compounds for the ant: $C \leftarrow \mathscr{C} \cup S(r) \cup P(r)$
10 **do**
11 Identify feasible reactions from C: $\mathbf{r} \leftarrow \{r/r \in \mathscr{R} \wedge S(r) \subseteq C\}$
12 Filter from $\mathbf{r}$ reactions already in π
13 **if** $\mathbf{r} \neq \emptyset$ **then**
14 Choose $r \in \mathbf{r}$ from the last reaction π_i^k inserted, according to: $p_{\pi_i,r} = \Omega_{\pi_i^k,r} / \sum_{\forall j \in \mathbf{r}} \Omega_{\pi_i^k,j}$
15 Extend the pathway: $\pi_{i+1}^k \leftarrow r$
16 Update the set of available compounds: $C \leftarrow C \cup P(r)$
17 **while** *new reactions have been added to the pathway* ***and*** *there are still compounds to relate*
18 Filter unnecessary reactions from π^k:
19 $$\widehat{\pi}^k \leftarrow \{\pi_j / P(\pi_j^k) \cap \mathscr{D} \neq \emptyset \vee \exists n > 0 / \left(P(\pi_j^k) - \mathscr{C}\right) \cap S(\pi_{j+n}^k) \neq \emptyset\}$$
20 Evaluate the pathway cost according to connectivity $\kappa(\cdot)$, unique reactions $\varphi(\cdot)$, and reactions synthesizing new compounds $\phi(\cdot)$: $f(\widehat{\pi}^k) = \kappa(\widehat{\pi}^k)\varphi(\widehat{\pi}^k)|\widehat{\pi}^k| / \phi(\widehat{\pi}^k)$
21 Evaporate pheromones: $\Omega \leftarrow (1-\rho)\Omega$
22 Update pheromones: $\Omega_{i,j} \leftarrow \Omega_{i,j} + \sum_k \sum_l \delta(i, \widehat{\pi}_l^k) \delta(j, \widehat{\pi}_{l+1}^k) / f(\widehat{\pi}^k)$
23 Update best pathway found π^*

Once the ant completed the search, unnecessary reactions are removed of π^k, and the cost of the resulting pathway $\widehat{\pi}^k$ is calculated (line 18). The pathway cleaning step consists in discarding reactions that do not synthesize any of the compounds in $\mathscr{D}$, that is, the ones that only produce compounds belonging to $\mathscr{C}$, or those which synthesize compounds that are not substrate for any reaction. The cost $f(\widehat{\pi})$ is calculated based on the evaluation of four characteristics of the pathway (line 19): number of reactions in the pathway $|\widehat{\pi}|$; number of unique reactions $\varphi(\widehat{\pi})$; number of reactions synthesizing new compounds $\phi(\widehat{\pi})$; and connectivity of the pathway $\kappa(\widehat{\pi})$. The number of unique reactions is calculated as $\varphi(\widehat{\pi}) = |\{\widehat{\pi}_i / \widehat{\pi}_i \neq \widehat{\pi}_j, \forall j < i\}|$, and penalize solutions including reactions used with both directions. The number of reactions synthesizing new compounds is determined as $\phi(\widehat{\pi}) = |\{\widehat{\pi}_i / \{P(\widehat{\pi}_i) - \{\cup_{\forall j<i} P(\widehat{\pi}_j)\} - \mathscr{C}\} \neq \emptyset\}|$, where $P(\widehat{\pi}_i)$ is the set of products for reaction $\widehat{\pi}_i$. This measure reaches its maximum value when all reactions in the pathway produce at least one new compound, not previously synthesized. Connectivity evaluates the number of final products (compounds in $\mathscr{D}$ without the initial substrate) synthesized from the initial substrate in the pathway. Let $X_0 = \{S(\widehat{\pi}_0) \cap \mathscr{D}\}$ be an initial set of compounds containing only the initial substrate used by the first reaction of the pathway. The update of this set is performed according to

$$X_{i+1} = \begin{cases} X_i \cup (P(\widehat{\pi}_{i+1}) - \mathscr{C}) & \text{if } X_i \cap S(\widehat{\pi}_{i+1}) \neq \emptyset, \\ X_i & \text{in other case.} \end{cases} \quad (1)$$

The latest updated set X_N contains all the compounds synthesized by any reaction related to the initial compound. Based on this set, connectivity $\kappa(\widehat{\pi})$ can take value

$$\kappa(\widehat{\pi}) = \begin{cases} 1 & \text{if } |X_N \cap \mathscr{D}| / |\mathscr{D}| = 1, \\ \alpha & \text{in other case,} \end{cases} \quad (2)$$

being α a constant that establishes the cost difference between partial solutions (only some final products are synthesized from the initial substrate) and complete ones. Therefore, when $\alpha \ll 1.0$ the solutions that relate only some of the compounds will

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

cost less than those that relate them all. In contrast, when $\alpha \gg 1.0$, the solutions that link all the compounds will have lower cost, and will be the ones that the ants will try to build. A recommended value is $\alpha = 10N_k$, being N_k the number of ants used in the search (see Supplementary Figure S2 for a detail on the effect of α for more details).

After ants have removed unnecessary reactions from the pathways and the cost of each solution was evaluated, the pheromone matrix is updated following two mechanisms (lines 20 and 21). First, the pheromone evaporation is done by removing a proportion ρ of the pheromones, in order to emulate the natural process of loss of information associated to evaporation. Next, the elements of Ω are updated according to the reactions used in the pathways found, and the cost of the solutions. Thus, given a pathway $\widehat{\pi}^k$, the usefulness of the first reaction $\widehat{\pi}^k_0$ is updated by adding the quantity $1/f(\widehat{\pi}^k)$ to $\Omega_{0,\widehat{\pi}^k_0}$. Then, the reactions sequence of the pathway is traversed, and the pheromone value $\Omega_{\widehat{\pi}^k_i,\widehat{\pi}^k_{i+1}}$ corresponding to every couple $\widehat{\pi}^k_i, \widehat{\pi}^k_{i+1}$ is updated by adding the quantity $1/f(\widehat{\pi}^k)$. Once the three steps of collective knowledge update are finished, the best solution found is saved in π^*.

The algorithm searches until the best solution does not change for a given number of iterations, and all the ants follow different paths according to their costs.

Datasets and measures

Reactions used in the experiments were extracted from the KEGG database[1] (other repositories such as MetaCyc[37] could be used as well). The direction for each reaction was assigned using the information contained in the KGML files associated to the reference maps[38,39]. Each reversible reaction was modelled as a pair of independent reactions with opposite direction. A total of 5 datasets of reactions (*glycolysis*, *proline*, *xproline*, *multipaths*, *ecoli*) were used in the experiments. Details on the datasets of reactions and the list of freely available compounds, are provided in the Supplementary Material, Tables S1 and S2.

Algorithms were evaluated on searching time t, the number of reactions N_R in the solution and the branching factor β. Even though searching time depends on many elements, it was used as a rough indicator of computational cost. The branching factor evaluates the relation among reactions in the pathway, measuring the average number of reactions that use every non-abundant substrate. It is calculated according to

$$\beta(\pi) = \frac{1}{|S^*_f|} \sum_{i=1}^{|S^*_f|} \sum_{j=1}^{|\pi|} \mathbf{1}_{s_i \subseteq S(r_j)}, \tag{3}$$

where S^*_f are the substrates of all reactions in π after filtering the abundant compounds, $|\pi|$ is the pathway size, $\mathbf{1}$ is the indicator function, s_i is the $i-$th compound of S^*_f, and $S(r_j)$ are the substrates of reaction r_j.

For comparisons with other state-of-the-art methods, a benchmark dataset of 42 reference pathways derived from the aMAZE database[40] and provided by Huang *et al.*[18] have been used. It consists of real pathways up to 10 reactions belonging to *E. coli*, *S. cerevisiae*, and *H. sapiens* that are commonly used for evaluation of pathfinding methods in the literature[18]. Performance on the available synthesized pathways was evaluated according to measures defined in literature[17], being: true positives (TP) those elements (compounds or reactions) found in both the reference and the synthesized pathway; false positives (FP) those elements in the synthesized pathway but not in the reference one; and false negatives (FN) correspond to elements in the reference pathway but not in the synthesized one. Precision is calculated as PR=TP/(TP+FP) and, in this context, it provides information about the proportion of compounds/reactions in the synthesized pathway which effectively are in the reference one. The higher this value, the fewer compounds/reactions outside the reference pathway will be part of the synthesized pathway. Recall is calculated as RC=TP/(TP+FN), and indicates the proportion of compounds/reactions of the reference pathway that are in the synthesized one (proportion of the reference pathway effectively recovered). Accuracy is calculated as Acc = (PR+RC)/2, and gives a balance between both previous measures.

Results and discussion

Comparison of searching times.

Searching time required for DFS, BFS, EvoMS and PhDSeeker for finding metabolic pathways was evaluated in a simple problem. This consisted on the search of a metabolic pathway between compounds C00103 (D-glucose-1-phosphate) and C00631 (2-phospho-D-glycerate), using the *glycolysis* dataset of reactions and C00103 as initial substrate. For this experiment, the initial set of available compounds containing only those required for finding at least one solution was built combining freely available compounds with substrates of reactions using C00103. However, it is important to remark that the initial set of available compounds can be built with as many compounds as it is desired. A reasonable starting point could include compounds such as water, NADH, H^+ and many others that are actually freely available in living organisms (see Supplementary

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

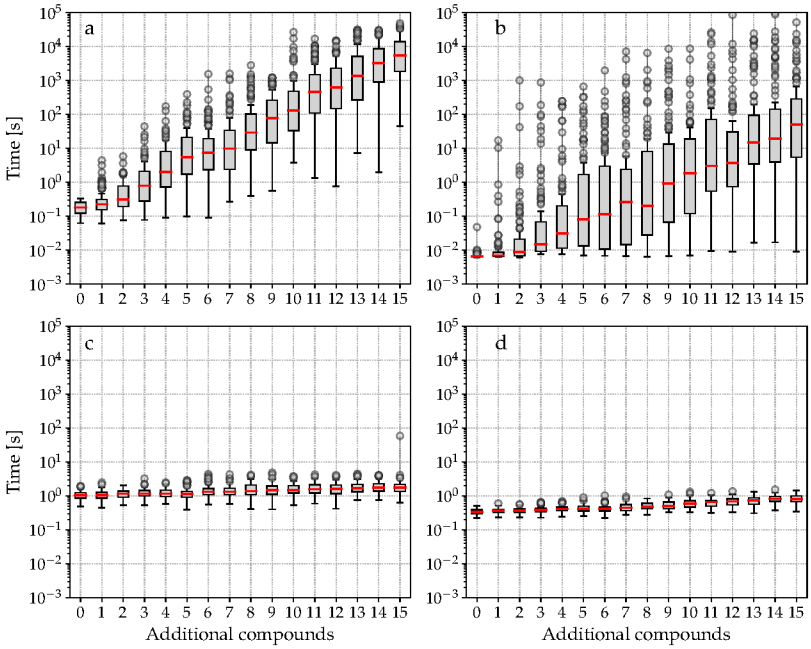

Figure 3. Searching times required to find a metabolic pathway considering a growing number of compounds added to the minimum set of available ones. a) BFS; b) DFS; c) EvoMS; d) PhDSeeker. Red line denotes the median, and circles indicate outliers.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

Table S2 for an example). Then, this set could be extended with other compounds based on the knowledge of the organism. For example, metabolomic information could be used to identify some extra compounds that could be included in the set of available ones.

In order to generate different initial conditions for the searching problem, an increasing number of extra compounds, randomly selected, was added to the initial set of available compounds. It should be noted that the minimum set of initially available compounds can be specified for this problem, since the solution is well-known. We performed 16 experiments, each one with 100 runs. Search operators (reactions) were randomly sorted on each run of the BFS and DFS algorithms. The maximum search depth for DFS was 10, corresponding to twice the number of reactions of the shortest metabolic pathway. Preliminary experiments indicated that EvoMS required up to 100 individuals to find a solution to this problem. In the case of PhDSeeker, it was observed that 5 ants were enough to build a metabolic pathway linking both compounds.

Figure 3 shows a boxplot of the searching time for all methods. As it can be seen, classical methods show an exponential increase in the searching time, from tenths of seconds to minutes, when a large number of extra compounds is taken into account. Since the root node corresponds to the set of initially available compounds, feasible reactions from this set are the possible branches for the first level of the search tree. In consequence, the increase in the size of this initial set is quickly reflected as an increase in the number of branches for the first level of the search tree, and this effect is then translated to the following levels. Clearly, it leads to a growth in the number of states to be explored to find a solution. Moreover, it can also be appreciated a high variability in the searching time and large number of outliers, because there is not *a-priori* knowledge about how the search operators should be applied to find solutions in the minimum number of steps. In contrast, searching time for metaheuristic algorithms are practically not modified when increasing the number of extra compounds, because they perform a smartest exploration of the search space. Furthermore, this makes searching times variability very small, staying always around the second.

Clearly, this result shows that performance of classic search methods is strongly influenced by the initial conditions of the problem. Even in this simple problem with a relatively small search space, the searching time easily becomes unmanageable. Instead, the effect on metaheuristics is minimal, making them a suitable tool to address real problems of higher complexity.

Increasing the search space.

Performance of EvoMS and PhDSeeker was compared by searching pathways among compounds C00025 (glutamate), C00122 (fumarate) and C00763 (proline) in *proline* and *xproline* datasets of reactions. Solution is well-known for *proline* dataset of reactions, and corresponds to a branched pathway starting from C00025. We expected that both algorithms be able to find the solution, regardless the size of the search space.

	proline		*xproline*	
	EvoMS	PhDSeeker	EvoMS	PhDSeeker
N_R	13.99 (3.85)	9.80 (0.81)	8.20 (3.34)	6.99 (2.69)
β	1.47 (0.18)	1.27 (0.08)	1.45 (0.28)	1.24 (0.08)
t	13.31 (7.44)	1.90 (0.38)	5.01 (3.02)	6.18 (1.67)

Table 2. Average performance for 3-compounds (100 runs of each method). N_R: Number of reactions comprising the solution. β: branching factor (average number of reactions that use every non-abundant substrate). t: time required to find a solution. Standard deviation in brackets.

Experiments were performed with the following configuration. EvoMS was run with $N_k = 100$ individuals, crossover probability $p_x = 0.8$, mutation probability $p_m = 0.08$, erasure probability $p_e = 0.8$ and valid insertion probability $p_v = 0.5$. Those parameters were determined in previous experiments[21], specifying the maximum number of generations to $G_M = 1000$. In all cases, the best individual was preserved on each generation (elitism) and a generational gap of 30 individuals was used. PhDSeeker was run up to a maximum of 100 iterations, using $N_k = 10$ ants and an evaporation rate of $\rho = 0.1$. In a preliminary experiment with a completely independent dataset and reference pathway it was observed that those parameters provide a good performance (see Supplementary Figure S2 for more details). For both algorithms, the number N_k of individuals was selected to be the minimum number of individuals required for finding a metabolic pathway linking the specified compounds.

Table 2 shows performance measures evaluated in both datasets of reactions. While both algorithms generate solutions in a wide range of sizes, metabolic pathways found by EvoMS have significantly more reactions than the pathways found by PhDSeeker ($p < 0.001$, with the Wilcoxon signed-rank test), when considering the *proline* dataset of reactions. This is due to the presence of a greater number of redundant reactions that are not filtered in the solutions. Regarding the branching factor, it must be noted that both algorithms have $\beta > 1.0$, indicating that, in fact, solutions are branched. Difference in the average value are given by the way in which each method initializes the pathway search. In PhDSeeker, only one reaction using the initial substrate is allowed; since only substrates, for this reaction, are provided together with the available compounds. It makes that the branching factor depends exclusively on the branches in the pathway found. Instead, EvoMS builds the set of available compounds taking into account substrates for all the reactions using the initial substrate. It makes feasible the incorporation of several reactions that depend on the initial substrate. Thus, the branching factor will be increased by the presence of these additional initial reactions. Concerning the searching time, results show that the practical performance of both algorithms is comparable in general terms. Summarizing, we can say that results obtained with both algorithms become more similar when increasing the size of the search space.

Linking more compounds.

Results of the previous section shown that both metaheuristics find pathways that relate the specified compounds in search spaces of different sizes. In this section, we compared the capability for searching pathways in situations where several compounds can be used as starting substrate. Experiments were performed using the *multipaths* dataset of reactions, searching for pathways that relate compounds C00036 (oxaloacetate), C00118 (glyceraldehyde-3P), C00181 (D-xylose) and C00267 (α-D-glucose). Using these compounds as initial substrate, four experiments (100 runs each) were done. Additionally, we also performed experiments using the automatic initialization strategy to select the best initial substrate. In all cases, we analyzed the proportion of runs where a metabolic pathway linking the four compounds was found. EvoMS and PhDSeeker were run with similar experimental configuration used in the previous section.

Figure 4 shows the number of runs where a metabolic pathway was found. Every couple of bars presents results for EvoMS (striped bars) and PhDSeeker using a given initial substrate. As it can be seen, EvoMS is only able to find solutions only when C00267 is used as initial substrate. Furthermore, this algorithm is unable to find any solution using C00118. Instead, PhDSeeker founds solutions in every run, regardless of the initial substrate considered. Moreover, the automatic selection of initial substrate used three of the four compounds for finding pathways to relate them. These results indicate that PhDSeeker outperformed EvoMS for searching pathways linking several compounds. The ant-based algorithm found a solution on each run, regardless of the mechanism used to select the initial substrate.

Comparison with state-of-the-art algorithms.

Performance of PhDSeeker was compared with several state-of-the-art metabolic pathfinding methods included in a very recent review[45]. Based on the availability of the algorithms, the comparison was made using AGPathFinder (search based on group-of-atoms-tracking and thermodinamics)[18], LPAT (search based on maximization of atoms transferred from source to target)[17], FMM (search based on the minimization of the number of known pathways to be combined in the solution)[41] and RouteSearch (search based on atom-tracking and thermodinamics)[42]. Furthermore, in order to extend the comparison to other

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

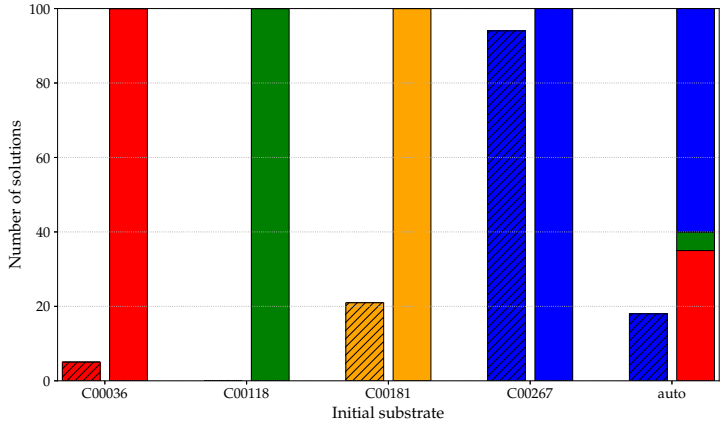

Figure 4. Number of runs in which a solution was found.

	COMPOUNDS						
	AGPathFinder	LPAT	FMM	RouteSearch	Graphtools	SubNet	PhDSeeker
Precision	0.866	0.873	0.887	0.822	0.927	0.457	**0.958**
Recall	0.826	0.872	**0.926**	0.818	0.836	0.678	0.914
Accuracy	0.846	0.873	0.907	0.820	0.881	0.568	**0.936**
	REACTIONS						
	AGPathFinder	LPAT	FMM	RouteSearch	Graphtools	SubNet	PhDSeeker
Precision	0.648	0.777	0.875	0.662	0.712	0.160	**0.883**
Recall	0.629	0.841	0.840	0.690	0.681	0.584	**0.861**
Accuracy	0.638	0.809	0.857	0.676	0.697	0.372	**0.872**

Table 3. Comparison of performance between PhDSeeker and several state-of-the-art methods. Best results in bold.

approaches, we also include two methods for subgraph extraction (see Table 3): Graphtools[11] and SubNet[43] (using *kWalks* strategy). For each pathfinding method, the first 10 solutions for each real reference pathway were evaluated, and the solution with higher accuracy was chosen to calculate performance measures. In case of subgraph extraction methods, each network found was taken as solution. Clearly, results shown in Table 3 are similar for measures calculated on compounds or reactions. Precision results indicate that pathways recovered by PhDSeeker are composed mainly by elements of the reference pathways, incorporating only very few foreign components. Recall values show that a high proportion of the reference pathways is recovered by PhDSeeker, being only FMM slightly better in terms of compounds. However, it is important to note that although compounds for FMM are mostly the same than in the reference pathways, this is not the case for reactions, since it does not use the same reactions as in the reference pathways. Regarding accuracy, PhDSeeker has the highest values for both compounds and reactions, indicating that it can achieve the best balance between Precision and Recall: pathways mainly contain elements of the reference pathway, and only a few external elements are included in some of the solutions.

Furthermore, it must be highlighted that our algorithm is designed for finding the shortest feasible pathways. In consequence, PhDSeeker was capable of finding solutions that share reactions with the reference pathways and that are also shorter, because it replaced several reactions by a unique step in order to minimize the pathway cost. While this may reduce precision and recall, pathways found are still fully feasible. Finally, it is important to note that our proposal has achieved these good results by using a simple model, which does not need to use information of the structure of compounds nor the thermodynamics of reactions.

Metabolic pathways in a model organism.

Validation using a standard pathway

Due to the fact that the most important point is the biological significance of results, we have used here the real well-known pathway for the synthesis of L-lysine (C00047), L-methionine (C00073) and L-threonine (C00188) from oxaloacetate (C00036), performed in *E. coli*, to evaluate the feasibility of solutions found with PhDSeeker. The algorithm was run using 10 ants, being the number of reactions in the *ecoli* dataset the boundary specified for the search.

Experimental results show that both pathways were similar in most reactions, having only a small difference in the

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

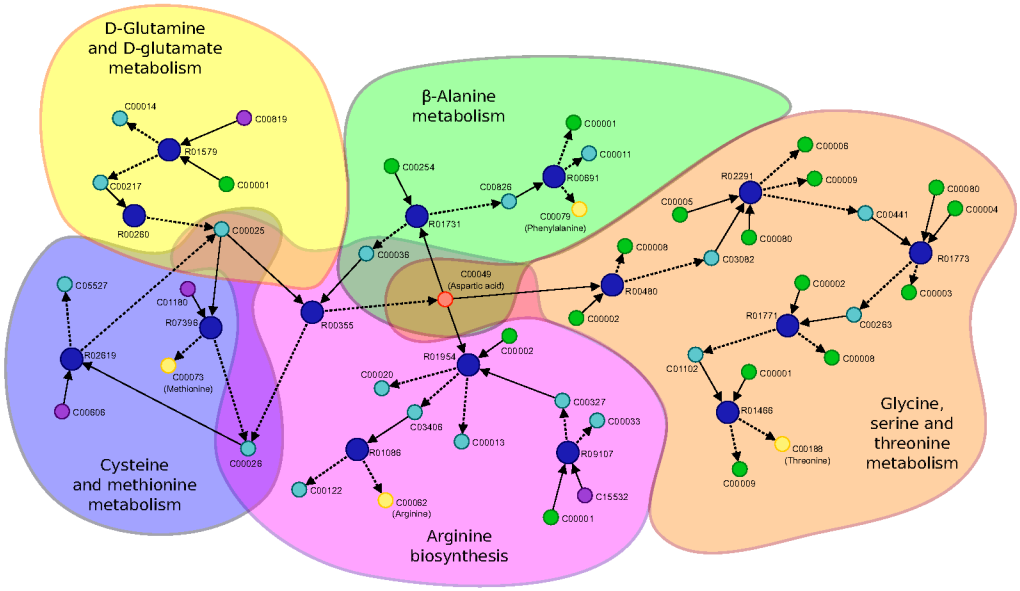

Figure 5. Example of a metabolic pathway linking threonine (C00188, neutral and polar), methionine (C00073, neutral and nonpolar), phenylalanine (C00079, aromatic), arginine (C00062, basic) and aspartic acid (C00049, acid). Well-known pathways involved in the solution are indicated with different colors.

mechanism for synthesizing C00073. While the standard pathway uses reactions R03260 and R01286 to transform C01118 into C00155, the solution found with PhDSeeker only requires reaction R01288 to perform this transformation. It is important to highlight that reactions R03260 and R01288 are catalyzed by the same enzyme (EC 2.5.1.48), but differ in the substrates used. This indicates that C00155 can be produced by means of the two-step way when substrates for the one-step transformation are not available. In conclusion, from a biological point of view, both pathways are similar. A simple representation of both pathways, the standard and the solution found by PhDSeeker, is provided in Supplementary Material, Figure S1.

Discovering a metabolic pathway linking several amino acids.

In this section we analyze the capability of PhDSeeker to build a pathway that relate five amino acids with different properties. For this purpose, we selected threonine (C00188, neutral and polar), methionine (C00073, neutral and nonpolar), phenylalanine (C00079, aromatic), arginine (C00062, basic) and aspartic acid (C00049, acid), and we use the latter as initial substrate. The search was performed using 10 ants, and reactions in the *ecoli* dataset also were specified as the boundary for this search.

Figure 5 shows an example of a metabolic pathway found and the known pathways to which compounds and reactions belong. Clearly, some compounds participate in several pathways, such as the aspartic acid (C00049) and oxaloacetate (C00036). As it can be appreciated, arginine, threonine and phenylalanine are synthesized by their own pathways, and only share the aspartic acid as initial substrate. This situation is different for methionine, since it can be produced without using aspartic acid, through an independent pathway (reactions in the yellow region) that produce L-glutamate (C00025). This compound together with 4-methylthio-2-oxobutanoate (C01180) are then used to produce methionine. Although aspartic acid does not contribute to produce methionine, it is still related to its synthesis. It is evident that C00025 is a key compound in the synthesis of C00049 and C00073, and must be consumed by R07396 and R00355 to produce their corresponding products. In case of a heavy consumption of C00049, production of this amino acid probably will be preferred, decreasing production of C00073. Fortunately, both reactions produce 2-oxoglutarate (C00026), which is used for reaction R02916 to synthesize more C00025 and continue with the production of methionine. In this context, if R02916 was not present, synthesis of methionine probably would be stopped.

It should be noted here, that this solution comprises several known metabolic pathways, and that the algorithm is clearly able to overcome these limitations and find a feasible pathway that relates all the amino acids. Moreover, this search was performed automatically, saving time and avoiding the need to explore, by hand or text mining, all potential connections among the compounds.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

Conclusion

Synthesizing metabolic pathways is still an open challenge that requires the development of novel and more powerful computational methods. Here, we presented PhDSeeker, a novel ant-based algorithm for synthesizing feasible linear and branched metabolic pathways. Starting from a set of freely available compounds without connections, this algorithm searches for a sequence of feasible reactions that relate a given set of compounds. While explores the solutions space, it expands the original set with new compounds and connections, in order to carry out more reactions. Therefore, each state corresponds to a set of compounds and the relations among them, while transitions between them are performed by applying feasible reactions. This definition leads to a more extensive search space than the one associated to a typical compounds-and-reactions graph. However, our algorithm avoids this problem building solutions while searching and never working on the whole graph. Results show that this algorithm is able to find metabolic pathways linking several compounds, even when considering many compounds and a large number of available reactions. Validation tests demonstrate that this proposal can reproduce well-known pathways and can also synthesize novel solutions. This new algorithm can be a valuable tool for the study of the metabolism, and also for designing novel pathways in metabolic engineering and synthetic biology.

References

1. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* **45**, D353–D361 (2017).
2. Caspi, R. *et al.* The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.* **44**, D471–80 (2016).
3. Placzek, S. *et al.* BRENDA in 2017: new perspectives and new tools in BRENDA. *Nucleic Acids Res.* **45**, D380–D388 (2017).
4. Xu, Z., Sun, J., Wu, Q. & Zhu, D. Find_tfSBP: find thermodynamics–feasible and smallest balanced pathways with high yield from large–scale metabolic networks. *Sci. Rep.* **7** (2017).
5. Planes, F.J. & Beasley, J.E. A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Brief Bioinform* **9**, 422–436 (2008).
6. Arita, M. From metabolic reactions to networks and pathways. *Methods Mol. Biol.* **804**, 93–106 (2012).
7. Russell, S. J. & Norvig, P. *Artificial Intelligence: A Modern Approach* (Prentice Hall, 2010).
8. McShan, D. C., Rao, S. & Shah, I. PathMiner: predicting metabolic pathways by heuristic search. *Bioinforma.* **19**, 1692–1698 (2003).
9. Rahman, S. A., Advani, P., Schunk, R., Schrader, R. & Schomburg, D. Metabolic pathway analysis web service (pathway hunter tool at CUBIC). *Bioinforma.* **21**, 1189–1193 (2005).
10. Faust, K., Croes, D. & van Helden, J. Metabolic pathfinding using RPAIR annotation. *J. Mol. Biol.* **388**, 390–414 (2009).
11. Faust, K., Dupont, P., Callut, J. & van Helden, J. Pathway discovery in metabolic networks by subgraph extraction. *Bioinforma.* **26**, 1211–1218 (2010).
12. Kotera, M., Okuno, Y., Hattori, M., Goto, S. & Kanehisa, M. Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. *J. Am. Chem. Soc.* **126**, 16487–16498 (2004).
13. Kotera, M. *et al.* RPAIR: a reactant-pair database representing chemical changes in enzymatic reactions. *Genome Inf.* **15**, P062 (2004).
14. Easton, J. M., Harris, L. M., Viant, M. R., Peet, A. C. & Arvanitis, T. N. Linked Metabolites: A tool for the construction of directed metabolic graphs. *Comput. Biol. Medicine* **40**, 340–349 (2010).
15. Gerard, M. F., Stegmayer, G. & Milone, D. H. An evolutionary approach for searching metabolic pathways. *Comput. Biol. Med.* **43**, 1704–1712 (2013).
16. Pitkänen, E., Jouhten, P. & Rousu, J. Inferring branching pathways in genome-scale metabolic networks. *BMC Syst. Biol.* **3**, 103 (2009).
17. Heath, A. P., Bennett, G. N. & Kavraki, L. E. Finding metabolic pathways using atom tracking. *Bioinforma.* **26**, 1548–1555 (2010).
18. Huang, Y., Zhong, C., Lin, H. X. & Wang, J. A method for finding metabolic pathways using atomic group tracking. *PLoS One* **12**, e0168725 (2017).
19. Khosraviani, M., Saheb Zamani, M. & Bidkhori, G. FogLight: an efficient matrix-based approach to construct metabolic pathways by search space reduction. *Bioinforma.* **32**, 398–408 (2016).
20. Gerard, M. F., Stegmayer, G. & Milone, D. H. EvoMS: An evolutionary tool to find de novo metabolic pathways. *Biosyst.* **134**, 43–47 (2015).
21. Gerard, M. F., Stegmayer, G. & Milone, D. H. Evolutionary algorithm for metabolic pathways synthesis. *Biosyst.* **144**, 55–67 (2016).
22. Dorigo, M., Birattari, M. & Stützle, T. Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006).
23. Bonabeau, E., Dorigo, M. & Theraulaz, G. Inspiration for optimization from social insect behaviour. *Nat.* **406**, 39–42 (2000).
24. Chandra Mohan, B. & Baskaran, R. A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert. Syst. Appl.* **39**, 4618–4627 (2012).
25. Shmygelska, A. & Hoos, H. H. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC Bioinforma.* **6**, 30 (2005).
26. Jing, P.-J. & Shen, H.-B. MACOED: a multi-objective ant colony optimization algorithm for SNP epistasis detection in genome-wide association studies. *Bioinforma.* **31**, 634–641 (2015).
27. Kleinkauf, R., Mann, M. & Backofen, R. antaRNA: ant colony-based RNA sequence design. *Bioinforma.* **31**, 3114–3121 (2015).

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.

28. Zaidman, D. & Wolfson, H. J. PinaColada: peptide-inhibitor ant colony ad-hoc design algorithm. *Bioinforma.* **32**, 2289–2296 (2016).

29. Oakley, M. T., Richardson, E. G., Carr, H. & Johnston, R. L. Protein structure optimization with a "lamarckian" ant colony algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **10**, 1548–1552 (2013).

30. Korb, O., Stützle, T. & Exner, T. E. An ant colony optimization approach to flexible protein–ligand docking. *Swarm Intell.* **1**, 115–134 (2007).

31. Gao, Y., Guan, H., Qi, Z., Hou, Y. & Liu, L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *J. Comput. Syst. Sci.* **79**, 1230–1242 (2013).

32. Robbins, K. R., Zhang, W., Bertrand, J. K. & Rekaya, R. The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. *Math. Med. Biol.* **24**, 413–426 (2007).

33. Jabbarpour, M. R., Malakooti, H., Noor, R., Anuar, N. B. & Khamis, N. Ant colony optimisation for vehicle traffic systems: applications and challenges. *Int. J. Bio-Inspired Comput.* **6**, 32–56 (2014).

34. Dorigo, M. & Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**, 53–66 (1997).

35. Jiang, Z. *et al.* Comparing an ant colony algorithm with a genetic algorithm for replugging tour planning of seedling transplanter. *Comput. Electron. Agric.* **113**, 225–233 (2015).

36. Lacroix, V., Fernandes, C. G. & Sagot, M.-F. Motif search in graphs: application to metabolic networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **3**, 360–368 (2006).

37. Altman, T., Travers, M., Kothari, A., Caspi, R. & Karp, P. A systematic comparison of the MetaCyc and KEGG pathway databases. *BMC Bioinforma.* **14**, 1–15 (2013).

38. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).

39. Kanehisa, M. *et al.* KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.* **44**, D457–D462 (2016).

40. Lemer, C. *et al.* The aMAZE LightBench: a web interface to a relational database of cellular processes. *Nucleic Acids Res.* **32**, D443–8 (2004).

41. Chou, C.-H., Chang, W.-C., Chiu, C.-M., Huang, C.-C. & Huang, H.-D. FMM: a web server for metabolic pathway reconstruction and comparative analysis. *Nucleic Acids Res.* **37**, W129–34 (2009).

42. Latendresse, M., Krummenacker, M. & Karp, P. Optimal metabolic route search based on atom mappings. *Bioinforma.* **30**, 2043–2050 (2014).

43. Lemetre, C., Zhang, Q. & Zhang, Z.D. SubNet: a Java application for subnetwork extraction. *Bioinforma.* **29**, 2509–2511 (2013).

44. Handorf, T.,Ebenhöh, O. & Heinrich, R. Expanding Metabolic Networks: Scopes of Compounds, Robustness, and Evolution. *J. Mol. Evol.* **61**, 498–512 (2005).

45. Sarah M. Kim,Matthew I. Peña,Mark Moll,George N. Bennett &Lydia E. Kavraki A review of parameters and heuristics for guiding metabolic pathfinding. *J. Chemoinformatics* **9**:51, 1–13 (2017).

Acknowledgements

This work was supported by CONICET (PIP 2013-2015 #117), ANPCyT (PICT 2014-2627, PICT 2015-2472) and UNL (CAI+D 2016 #42, CAI+D 2016 #59, CAI+D 2016 #82). Present work used computational resources from the Pirayu cluster, acquired with funds from the Santa Fe Science, Technology and Innovation Agency (ASACTEI), Government of the Province of Santa Fe, through Project AC-00010-18, Resolution No. 117/14. This equipment is part of the National System of High Performance Computing of the Ministry of Science and Technology of Argentina.

Author contributions statement

Conceived and designed the experiments: MFG, GS, DHM. Performed the experiments: MFG. Analyzed the results: MFG. Wrote the paper: MFG, GS, DHM.

Additional information

Supplementary information accompanies this paper at https://www.nature.com/srep.

Source code of this algorithm is available at https://sourceforge.net/projects/sourcesinc/files/phdseeker/. Examples for searching metabolic pathways among several compounds are provided. The software is also available as a web demo at http://sinc.unl.edu.ar/web-demo/phdseeker2/.

Competing Interests: The authors declare no competing interests.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
M. Gerard, G. Stegmayer & D. H. Milone; "Metabolic pathways synthesis based on ant colony optimization"
Scientific Reports, Vol. 8, No. 1, pp. 16398-, 2018.