Computers in Biology and Medicine 43 (2013) 1704–1712

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

An evolutionary approach for searching metabolic pathways

Matias F. Gerard [a,b,*], Georgina Stegmayer [a,b], Diego H. Milone [b]

[a] Center for Research and Development of Information Systems (CIDISI), National Scientific and Technical Research Council (CONICET), Argentina
[b] Research Center for Signals, Systems and Computational Intelligence (Sinc(i)), FICH-UNL, National Scientific and Technical Research Council (CONICET), Argentina

ARTICLE INFO

Article history:
Received 10 September 2012
Accepted 21 August 2013

Keywords:
Search strategies
Evolutionary algorithms
Metabolic pathways

ABSTRACT

Searching metabolic pathways that relate two compounds is a common task in bioinformatics. This is of particular interest when trying, for example, to discover metabolic relations among compounds clustered with a data mining technique. Search strategies find sequences to relate two or more states (compounds) using an appropriate set of transitions (reactions). Evolutionary algorithms carry out the search guided by a fitness function and explore multiple candidate solutions using stochastic operators. In this work we propose an evolutionary algorithm for searching metabolic pathways between two compounds. The operators and fitness function employed are described and the effect of mutation rate is studied. Performance of this algorithm is compared with two classical search strategies. Source code and dataset are available at http://sourceforge.net/projects/sourcesinc/files/eamp/

1. Introduction

Using search strategies to solve different problems is common in many areas of knowledge. In many cases, employing classical strategies for sequential state space exploration allows to find solutions rapidly. When all possible solutions are exhaustively explored the strategies are called uninformed search strategies, and this is the case of breadth first search (BFS) and depth first search (DFS) algorithms [1]. It is a well-known fact that there are problems in which a very high number of solutions must be explored, making classical methods practically inapplicable. For example, in KEGG database [2] there are around 17,000 compounds with approximately 14,000 connections among them, and a high branching factor. There are different approaches to address these problems, among which evolutionary algorithms (EAs) have an important place. The main difference with DFS and BFS is that EAs do not explore the state space exhaustively, but rather use several heuristics to select the most promising regions to explore. These methods are grouped in four families: Genetic Algorithms [3], Evolutionary Strategies [4], Genetic Programming [5] and Evolutionary Programming [6]. Each one was originated by different motivations, and they differ mainly by their representation schemes, and operators of selection and reproduction [7]. Although the convergence for genetic algorithms is guaranteed by the schema theorem [8], real values codification is limited by the number of bits used. Instead, the evolutionary strategies directly use real values to encode the problem variables, but their convergence depends on the operators used. Evolutionary algorithms use stochastic search based on the evolution of a population of candidate solutions, applying a set of operators and a fitness function that evaluates the quality of the solutions generated. Some interesting aspects about these techniques are the simplicity of the operators used, the possibility of using fitness functions with very few formal requirements, and the ability to explore multiple points of the search space in each iteration [9]. These characteristics make them an attractive alternative to deal with several problems in biology [10–12].

Different search strategies to find metabolic pathways that relate compounds have been recently proposed. The algorithm described by Ogata et al. [13] is based on BFS and builds pathways between pairs of compounds by the combination of allowed relations (metabolic reactions). The method of Linked Metabolites [14] first builds an integrated graph and then performs the pathway search specifying a maximum number of reactions between source and target compounds. Metabolic PathFinding Tool [15] assigns to each operator a cost equal to the number of reactions where the compound participates. McShan et al. [16] use the A* search algorithm to explore the solutions space guided by a cost function based on the Manhattan distance and a heuristic function that uses structural information of compounds to generate characteristic descriptors. A more recent algorithm based on BFS is proposed by Heath et al. [17], where a metabolic pathway linking two compounds is found preserving a specified number of atoms

* Corresponding author at: Research Center for Signals, Systems and Computational Intelligence (Sinc(i)), FICH-UNL, CONICET, Argentina. Tel: +54 342 457 5234x119.
E-mail addresses: mgerard@santafe-conicet.gov.ar (M.F. Gerard), gstegmayer@santafe-conicet.gov.ar (G. Stegmayer), d.milone@ieee.org (D.H. Milone).

0010-4825/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compbiomed.2013.08.017

(atom tracking) between the beginning and ending compounds. However, these last two algorithms require information about the molecular structure of the compounds to be used, and BFS-based methods require a significant amount of memory to store all the search tree. Furthermore, given that methods based on BFS look for a specified number of pathways, the order in which the nodes of the tree are visited can bias the search to particular solutions (unless the successors are selected using randomized traversals). Another alternative is based on elementary modes. These methods use stoichiometry of reactions and several restrictions to identify minimal sets of enzymes that can operate at steady state, with all irreversible reactions used in the appropriate direction [18,19]. Computing all elementary modes is expensive, even for small networks [20]. There also exist retrosynthesis-based methods that build new metabolic pathways to produce a compound of interest in an organism [21]. These methods begin with the desired compound and use reverse enzymatic reactions to synthesize a metabolic pathway from simpler compounds. Although the problem being addressed in those works is close to our own, their objective differs from the one proposed in this paper.

Finding relationships between two given compounds is not an easy task, in particular when data come from different sources such as metabolic and transcriptional profiles.[1] In this case, one possible approach is to create clusters from the combined sources using a data mining tool [22]. This way, applying the "guilt-by-association" principle [23,24] genes and metabolites that vary coordinately can be found. Since the relationship among metabolites and transcripts is mainly given by metabolic pathways,[2] the following step would be searching such pathways with the available data. Traditionally, metabolic pathways search was manually performed, but the current increase in the volume of data demands for computational tools to perform the search automatically [17]. Many efforts have been made to automate the process, but obtained results are not biologically feasible. For example, in [25] a search for metabolic pathways with up to 9 reactions among glucose and pyruvate was performed and approximately 5×10^5 metabolic pathways were found, many of them biologically not possible.

The main contribution of this work is the proposal of an evolutionary algorithm to find metabolic pathways, capable of relating two compounds in a common and valid reaction chain. To achieve this, a data mining tool was used to generate clusters from a real biological dataset, and pairs of compounds within the clusters were used for the genetic search of metabolic pathways. Afterwards, objective measures were defined to quantify the performance of the algorithms, and the effect of the mutation rate on the evolution was studied. Finally, the proposed algorithm was compared with two methods based on classical search algorithms.

The paper is organized as follows: Section 2 describes the proposed algorithm for the evolutionary search of metabolic pathways between two compounds. The data used, the objective measures, and the results obtained are briefly described in Section 3. Finally, Section 4 presents the conclusions of this work.

2. Proposed algorithm

This section presents the proposed algorithm, that we will call evolutionary algorithm for the search of metabolic pathways (EAMPs)[3]. First, the state space and search operators employed are defined. Then, the structure of the chromosomes and the way the information is coded is presented. Afterwards, the genetic operators used and their functioning are described. Finally, the fitness function employed is presented, the terms that compose it are analyzed and the effect that each of them produces on the search is described.

There are different approaches to explore the space of all the possible metabolic pathways linking two specific compounds. One proposal consists of generating a list of compounds that must be excluded from the search [26]. However, incorrect definitions can exclude compounds necessary to produce results of biological interest. A different approach was proposed in [27] where sets of "substrate-product" binary relations were used to represent the reactions and each relation was labeled according to its function inside the reaction. The main stream of the pathways was built using only the relations containing information about the transformation of the substrates.

Following that idea, the state space is defined as the set C of all metabolic compounds in the KEGG database. This database contains information of genes, proteins and metabolic compounds of hundreds of different organisms and the allowed binary relations between compounds are describe by transformations r. The compound on which the transformation is applied will be called substrate s, and p will be the product or new resulting state. Transformations will be represented as ordered pairs $r_i = (s_i, p_i)$, with $s_i, p_i \in C$ and $s_i \neq p_i$. In addition, the substrate and product of r_i will be identified using the notation s_i and p_i respectively, being $\hat{s}$ the initial compound and $\hat{p}$ the final compound of the metabolic pathway. In this way, a metabolic pathway is built as a sequence of transformations that produce $\hat{p}$ starting from $\hat{s}$. Finally, the sequence of possible states $\mathbf{q} = [\hat{s}, p_1, p_2, \ldots, \hat{p}]$ is defined as the sequence of compounds that take part in the transformation.

2.1. Structure of the chromosomes

The sequence of transformations r leading to the production of $\hat{p}$ from $\hat{s}$ is coded in the chromosome as $\mathbf{c} = [r_1, r_2, \ldots, r_i, \ldots, r_N]$, where N indicates the number of genes and the sequence is read from left to right. In this context, the term *chromosome* indicates a data structure such as a vector, and should not be interpreted in a biological way. This value can vary in the range $[1, N_M]$, where N_M is the maximum number of reactions the metabolic pathway can contain. When the number of reactions exceeds this level, the chromosome truncates to contain only the first N_M reactions.

2.2. Genetic operators

This section describes the genetic operators[4] designed for the EAMP. Due to the requirements of this application in particular, it has been necessary to make various changes to classical genetic operators, which, if directly applied, would limit the convergence of the algorithm. In order to facilitate their explanation, four sets of transformations are defined. R^* contains the complete set of allowed transformations, $R^1 = \{r_i \backslash r_i = (\hat{s}, p_i)\} \wedge R^1 \subset R^*$ contains only those transformations that use $\hat{s}$, $R^N = \{r_i \backslash r_i = (s_i, \hat{p})\} \wedge R^N \subset R^*$ contains all transformations that produce $\hat{p}$, and $R^+ = R^1 \cup R^N$ contains the union of the two previous sets. The algorithm finds a solution when it reaches a predefined maximum number of generations or when the fitness of an individual takes the value 1, indicating that it encodes a metabolic pathway that relates the indicated compounds.

[1] Metabolic profile: measurement of concentration levels of small molecules. Transcriptional profile: measurement of activity levels of a set of genes.

[2] A metabolic pathway is a sequence of chemical reactions that transform a substrate into one or various products through a series of intermediary compounds.

[3] Source code and dataset are available at http://sourcesinc.sourceforge.net/eamp/

[4] This general term indicates operations applied over chromosomes. For example, the crossover operator combines genetic information of two chromosomes to produce a new one.

Moreover, if maximum number of generations is reached, the algorithm is stopped and returns the best individual found.

Initialization: The algorithm is initialized by defining the number of individuals M that the population contains and a value $N_i \leq N_M$ for each individual, which indicates the maximum number of genes a chromosome can initially have. Chromosomes can be built in two ways. If a random initialization strategy is used, transformations are selected according to

$$\mathbf{c} = \begin{cases} r_i \in R^+ & \text{if } N = 1, \\ [r_i, r_j], r_i \in R^1, r_j \in R^N & \text{if } N = 2, \\ [r_i, \ldots, r_k, \ldots, r_j], r_i \in R^1, r_k \in R^*, r_j \in R^N & \text{if } N > 2, \end{cases} \quad (1)$$

where every gene r is a transformation randomly selected from the corresponding set. When a valid initialization strategy is employed, transformations are selected according to

$$\mathbf{c} = \begin{cases} r_i \in R^+ & \text{if } N = 1, \\ [r_i, \ldots, r_{j-1}, r_j], r_i \in R^1, r_j \in R^*/s_j = p_{j-1} & \text{if } N \geq 2. \end{cases} \quad (2)$$

All transformations that are already part of the chromosome are not selected again to be inserted into the sequence. When valid initialization is forced to set a chromosome with N_x genes, it will be called fixed-length valid initialization, and the chromosome will be completed up to contain the specified number of genes.

Selection: In this algorithm the traditional roulette method was used [9]. This method is based on assigning a value f to each individual that is proportional to its contribution to the fitness mean of the population. Tournament selection with 3, 5 and 7 competitors was also tested. In an independent experiment, six searches were performed, each one with 30 runs. Results show that roulette uses searching times significantly lower than tournament ($p < 0.05$). Elitism is employed to guarantee the preservation of the fittest individuals in each generation. The parameter ε determines the number of individuals that will be preserved and that will pass to the following generation without any modification.

Crossover: This operator presents an important modification compared to the classical one, because it promotes the formation of valid bonds of the genetic material. The crossover point ϕ for each parent $(\mathbf{c}_1, \mathbf{c}_2)$ is randomly selected from a set containing pairs of positions (ϕ_1, ϕ_2) that satisfy $\delta(s_i, p_j) = 1$, where δ is the Kronecker delta function, which takes value 1 when $s_i \in \mathbf{c}_1$ and $p_j \in \mathbf{c}_2$ are equal. Fig. 1 shows a diagram of the functioning of the operator in the case of two parents that are not completely valid (see definition of validity later). Each block represents a gene and codes a chemical reaction in which letters represent the substrates and products of each relation. It can be noticed that if a simple crossover method is applied without considering the sequence of reactions, the validity of the generated offspring would probably diminish. However, if the crossover is carried out in one of the highlighted pairs of positions (ϕ_1, ϕ_2), the validity of the offspring will increase. When there is not a feasible crossover point, two different progenitors are selected.

Mutation: This operator replaces a chromosome gene by another one where s or p of the new gene is p of the previous gene or s of the following gene, respectively. Each chromosome has a probability μ of being mutated in a single randomly selected position. The new gene is obtained according to

$$mut(r_i) = \begin{cases} r \in R^+ & \text{if } N = 1, \\ r \in R^1 & \text{if } N > 1 \wedge i = 1, \\ r \in R^N & \text{if } N > 1 \wedge i = N, \\ r \in R^*/p = s_{i+1} & \text{if } N > 1 \wedge 1 < i < N \wedge u \leq 0.5, \\ r \in R^*/s = p_{i-1} & \text{if } N > 1 \wedge 1 < i < N \wedge u > 0.5, \end{cases} \quad (3)$$

where s and p are, respectively, the substrate and product of the new gene r; s_{i+1} is the substrate of the gene that is located in the position next to the mutated gene r_i, and p_{i-1} is the product of the gene that is in the position previous to the mutated gene. Value u is randomly selected in the range [0,1]. Fig. 2 presents a diagram of the functioning of this operator. The choice of a new gene can result in a wide range of possible chromosomes. It is observed that in the chromosome placed at the top of the figure the gene selected to mutate has a valid relation with the previous gene, but not with the next one. Transformations that replace the gene to mutate should use the compound r as a substrate, the compound z as a product or both in the best case (bottom of the figure).

If a classical mutation operator is applied, there is high probability that the validity of the chromosome will diminish, because the new gene would not establish a valid union with their neighboring genes. On the contrary, if a valid mutation strategy is applied like in (3), it is more probable that the chromosome will increase its validity, as it can be seen in the chromosome at the bottom of the figure. Although in some cases the validity of the new chromosome might decrease, it is the cost to pay to be able to explore distant regions of the search space and leave local minima.

2.3. Fitness function

To evaluate the desired characteristics of the solutions, a fitness function f was built taking into account features of biochemical reactions, and it was employed to direct the search. The function f for the chromosome $\mathbf{c}$ is defined as $f(\mathbf{c}) = \alpha[V(\mathbf{c}) + \beta E(\mathbf{c}) + Q(\mathbf{c}) + I(\mathbf{c})]$, where $\alpha = 1/(3+\beta)$ is a normalization constant driving the function to the range [0,1] and β determines the relative contribution of

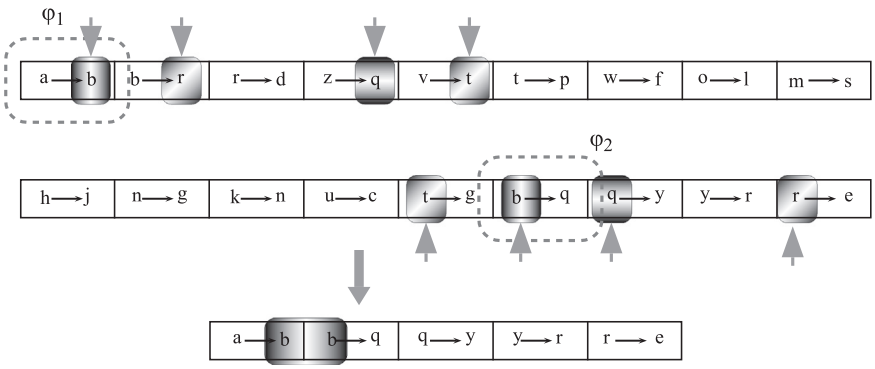

Fig. 1. Diagram of the functioning of the crossover operator. Each block corresponds to a gene encoding a transformation. In each gene, substrate and product are represented by letters at left and right of each arrow, respectively. Shaded elements indicate pairs of positions (ϕ_1, ϕ_2) where a valid crossover can be made.

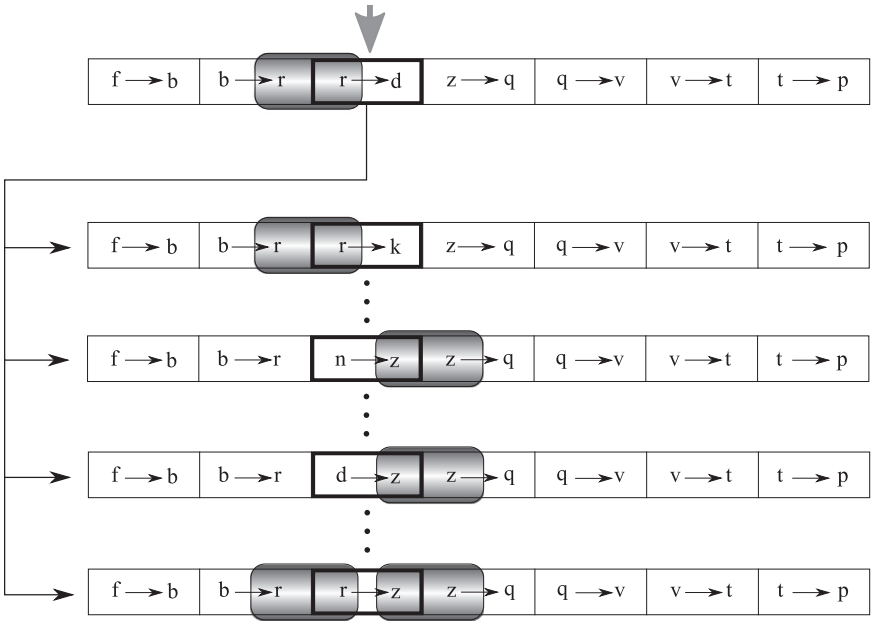

Fig. 2. Diagram of the functioning of the mutation operator. The selected gene to mutate is marked with a black rectangle. Top: Chromosome with a selected gene to be mutated. Bottom: Possible resulting chromosomes after mutation.

E measure. This function takes value 1 when a valid metabolic pathway, without trivial cycles, that produces $\hat{p}$ from $\hat{s}$ is found. In case of having information about the relative abundance of compounds, this function could be modified to weight the reactions according to the probability of occurrence, which is directly associated with the abundance of the compounds involved. The four measurements that are part of the function are described below.

Validity (V): It quantifies the number of valid concatenations present in the chromosome, defining them as those consecutive pairs of transformations where the product p_i of r_i is the substrate s_{i+1} of the transformation r_{i+1}. Based on this, validity is calculated as

$$V(\mathbf{c}) = \frac{\delta(\hat{s}, s_1) + \delta(p_N, \hat{p}) + \sum_{i=1}^{N-1} \delta(s_{i+1}, p_i)}{N+1}. \quad (4)$$

It varies in the range [0,1], being 1 when all operators are concatenated and the compounds s_1 and p_N are the ones desired.

Valid extremes (E): This term evaluates the transformations r_1 and r_N to verify they contain the desired $\hat{s}$ and $\hat{p}$ compounds. The calculation is done according to $E(\mathbf{c}) = \frac{1}{2}[\delta(\hat{s}, s_1) + \delta(p_N, \hat{p})]$. This term varies in the range [0,1] and reaches its maximum value when the compounds s_1 and p_N are the ones desired. This plays an important role when the size of the metabolic pathways exceeds N_M.

Unique reactions rate (Q): This term penalizes the repetition of transformations in the chromosome. Function φ is defined for its calculation, which evaluates a sequence and returns the number of unique elements present in it. The rate is calculated as $Q(\mathbf{c}) = (\varphi(\mathbf{c}) - 1)/(N-1)$ and it is defined $Q(\mathbf{c}) = 0$ when $N = 1$. It varies in the range [0,1] and reaches its minimum value when the sequence contains a unique element repeated N times ($\varphi(\mathbf{c}) = 1$).

Unique compound rate (I): This term penalizes the repetition of compounds in the pathway. The rate is calculated as $I(\mathbf{c}) = (\varphi(\mathbf{q}) - 2)/(N-1)$ and it is defined $I(\mathbf{c}) = 0$ when $N = 1$. It varies in the range [0,1] and reaches its minimum value when the chromosome contains transformations that only produce s_i or p_i (chromosomes containing repetitions of the transformations of $s_i \rightarrow p_i$ and $p_i \rightarrow s_i$). For example, in the metabolic pathway codified in the chromosome $\mathbf{c} = [a \rightarrow b][b \rightarrow a][a \rightarrow b][b \rightarrow d]$ the number of reactions is $N = 4$ and the sequence of states associated to it is $\mathbf{q} = [\underline{a}, \underline{b}, a, b, \underline{d}]$ (see definition on page 2) where only three compounds are unique ($\varphi(\mathbf{q}) = 3$). In consequence, $I(\mathbf{c}) = \frac{1}{3}$.

To illustrate the calculation of the fitness function, the lower chromosome in Fig. 1 will be used. If it is assumed that $\hat{s} = a$ and $\hat{p} = e$, then $V(\mathbf{c}) = 1$ and $E(\mathbf{c}) = 1$ because the substrate of each reaction is produced by the previous reaction and the terminal compounds (a and e) are the compounds to relate. $Q(\mathbf{c}) = 1$ since each gene in the chromosome is unique; and given that the sequence of compounds is $\mathbf{q} = [a, b, q, y, r, e]$, results $\varphi(\mathbf{q}) = 6$ and $I(\mathbf{c}) = 1$, since all the compounds appear only once in q. Therefore, $f(\mathbf{c}) = 1$. Instead, if we now assume that the third gene of the chromosome is $[b \rightarrow q]$, then $V(\mathbf{c}) = \frac{2}{3}$ and $E(\mathbf{c}) = 1$ because there are 2 invalid reactions (between genes 2–3 and 3–4) and the terminal compounds are the same; $Q(\mathbf{c}) = \frac{3}{4}$ since there is a gene repeated in the chromosome (third gene); given that now $\mathbf{q} = [\underline{a}, \underline{b}, \underline{q}, q, \underline{r}, \underline{e}]$, results $\varphi(\mathbf{q}) = 5$ and $I(\mathbf{c}) = \frac{3}{4}$. Therefore, $f(\mathbf{c}) \sim 0.792$.

3. Results and discussion

This section presents the results obtained in the evaluation of EAMP and its comparison with two classical search methods. First, the data used in the experiments are described. Then, the measures employed to compare the algorithms are presented. Afterwards, EAMP behavior is analyzed using different mutation rates. Finally, a comparison is made between the measures obtained with the different algorithms during the search of metabolic pathways.

The set of valid compounds to generate metabolic pathways and the possible chemical reactions among these compounds were extracted from the KEGG database.[5] This source was used because it is widely cited in the literature, and it contains information of a wide variety of organisms. In KEGG each compound is identified by means of a single code and chemical reactions store the information of the transformations using this coding. In addition, chemical reactions are stored as sets of "substrate-product" binary relations, where each one is labeled according to the function

[5] http://www.genome.jp/kegg/.

which the pair fulfills inside the chemical reaction. The relations labeled "main" were the ones selected to conduct the experiments since they contain the information about the transformations $s \rightarrow p$ [28]. The dataset used in the experiments was built by extracting and filtering all main pairs in order to preserve only one copy of each one. Each one was splitted into a backward and a forward reaction. After processing the data, 14,346 transformations relating 5936 compounds were extracted from the KEGG database.[6] Moreover, there is a subset of compounds ($\approx 5\%$ of dataset) with a high branching factor that reaches values up to 118.

Suppose that, for example, two clusters are found from a dataset with a data mining tool, and that we want to find a metabolic pathway relating two compounds belonging to each of those clusters. Then, we could use an algorithm for searching metabolic pathways and a set of possible transformations to link them. In this work, the dataset used consisted of metabolic and transcriptional profile data from tomato fruits cultivated under controlled conditions and harvested during maturation process [29]. A data mining tool based on a neural model [30] was selected to obtain the clusters. This tool is based on a self-organized map and uses the principle of "guilt-by-association" to build clusters of metabolites and transcripts with high probability of participating in the same biological process. It is important to remark that the use of a data mining tool is not a requirement of the algorithm. Compounds to relate could be selected using other criteria, for example, they could be manually selected. Transformations extracted from KEGG were used for linking compounds. Pairs of compounds belonging to each cluster were used to perform the search and test the algorithms. Table 1 presents details of the compounds belonging to the clusters. Isomers detailed in this table were considered as different compounds, so cluster A contains 6 compounds and cluster B contains 12 compounds. To simplify the notation, each compound code will be used without considering the letter and zeros preceding the number.

Table 1
Clusters selected to search for metabolic pathways.

Cluster A			Cluster B		
Compounds	Isomers		Compounds	Isomers	
	I	II		I	II
Arginine	C00062	C00792	Asparagine	C00152	C01905
Glycerate	C00258		Glycine	C00037	
Lysine	C00047		Histidine	C00135	
Ornithine	C00077	C00515	Isoleucine	C00407	
			Serine	C00065	C00740
			Tyrosine	C00082	
			Threonine	C00188	C00820
			Valine	C00183	C06417

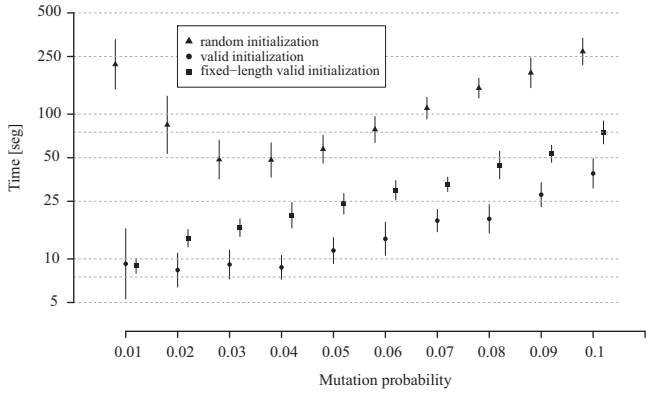

Fig. 3. Influence of the mutation rate on the searching time required by EAMP to find a solution using three different initialization strategies. Triangles, circles and squares indicate the median time of the dataset for each initialization and vertical bars represent the corresponding 95% confidence interval. Triangles indicate random initialization, circles indicate valid initialization and squares indicate fixed-length valid initialization. Times are plotted in logarithmic scale.

3.1. Performance measures

To compare the results obtained with different algorithms, the time required to find a metabolic pathway (t) was measured,[7] as well as the number of transformations (L) that the pathway contains and the number of compounds (ψ) belonging to the cluster of the pair of compounds which are part of the pathway. In the case of the EAMP, the number of generations used to find a pathway (G) was also evaluated. For each pair of compounds, 20 runs were carried out. For each run the maximum value (indicated by subscript M), the minimum value (indicated by subscript m) and the median (indicated by the symbol indicated by the symbol "$\hat{}$") for several measures were determined. These runs were carried out to evaluate the diversity of pathways found with different algorithms. In most measurements the median was used instead of the mean since it is a more robust measure for asymmetric distributions, as it happens in these cases.

To evaluate the proportion of cluster compounds in the metabolic pathway, we define the explanation rate of the cluster as

$$\Lambda = \frac{\max_k\{\psi_k\}}{|\Psi|}, \tag{5}$$

where k indicates the run number, ψ_k is the number of cluster compounds included in the pathway found in the run k, and $|\Psi|$ is the total number of cluster compounds. This rate varies in the range [0,1] and indicates the proportion of cluster compounds present in the metabolic pathway. Values of Λ close to 1 indicate that the pathway relates a great number of cluster compounds. The Wilcoxon signed-rank test was used to perform the statistical analysis of data [31].

3.2. Mutation rates and initialization strategies

To evaluate the influence of the mutation rate, this parameter was studied in the range 0.01–0.1, for the compounds marked in Table 1. Applying elitism ($\varepsilon = 1$) the best solution in each generation was preserved. In all runs a population of $M = 1000$ individuals, a maximum chromosome size of $N_M = 100$ genes, an initial maximum number of genes per chromosome of $N_i = 50$, and a crossover probability $\gamma = 0.8$ were used. Preliminary tests were performed with values of γ in the range [0.5, 0.9] and a step of 0.05. The best results were obtained for $\gamma = 0.8$. Random initialization, valid initialization and fixed-length valid initialization, were tested employing those parameters. For this last one, initial number of genes was set to $N_x = N_M$.

Results for the searching time and the number of generations were similar in both cases. To evaluate normality of the distributions of the time data, the Jarque–Bera test [32] was used. It performs an hypothesis test to compare a metric relating symmetry and kurtosis values of the distribution tested versus a normal distribution. In all cases, data showed non-normal distributions. Given that a median value is better than an average value to characterize asymmetric distributions, the first one was employed for each distribution of the time data and a confidence interval using bootstrap technique [33] was estimated. Confidence intervals for each distribution data are shown in Fig. 3. The symbols (triangles, circles and squares)

[6] The last free available version of KEGG data was downloaded in May, 2011.
[7] Experiments were conducted using a PC INTEL Pentium IV 3 GHz.

correspond to the median of the time values for each initialization strategy and the vertical lines represent the corresponding 95% confidence interval calculated by the bootstrap technique. Triangles correspond to results for random initialization, circles indicate valid initialization and squares correspond to fixed-length valid initialization. All times are shown in logarithmic scale. The minimum searching time for each strategy was associated to the lowest value of the median for all mutation rates. In addition, it was considered that overlapped confidence intervals did not present statistically significant differences.

Fig. 3 shows that valid initialization and fixed-length valid initialization have only small differences between them but they produce significant improvements compared to random initialization. For all the initialization strategies a solution was found for every run. Minimum searching time for random initialization was found at $p_m=0.04$ with no significant differences in the range 0.02–0.05. For valid initialization strategy the minimum was reached at 0.02 and there were no significant differences in the range 0.01–0.05. Fixed-length valid initialization strategy only has a lowest searching time at $p_m=0.01$.

These results may be due to how the mutation operator works. In the context of random initialization, two phases for the mutation operator can be identified. In the first phase, the operator helps building blocks of genes to increase the average population validity. In the second phase, when there are a few blocks per individual, these blocks are modified by the mutation operator in a way that allows the crossover operator to generate a feasible solution. As a consequence, the first phase can be delayed by low probabilities, while the second phase can be slowed by high probabilities because they would introduce a high variability in the searching process. Therefore, it is expected that intermediate probability values produce the best results.

When both valid initialization strategies are analyzed, the effect of the mutation rate on them is different and high mutation values slow the search. Thus, given that chromosomes are completely valid at the beginning and due to crossover operations also produce valid chromosomes, the incorporation of a high number of mutations slows the searching of solutions. Therefore, although mutations prevent the search from stopping at a local minimum, they drive it away from the region of the initial feasible solutions.

The differences between the valid initialization strategies could be explained by the initial number of genes in the population. Since fixed-length valid initialization has a larger number of genes at the beginning, small increases in the mutation rate produce larger effects. This is because the number of genes to mutate is proportional to the number of genes in the population. In addition, given that at the beginning all chromosomes are valid, the number of cross points increases and the speed at which unnecessary genes are eliminated is reduced.

An exponential increase in the searching time, in the range 0.05–0.1, was seen for random and valid initialization strategies because of the variability effect introduced by elevated mutation rates. The range of probabilities for fixed-length valid initialization was broader probably because the initial number of genes to mutate is larger than for the other strategies, increasing the influence of high mutation rates. In reference to the metrics for the evaluation of the pathways (L, ψ and Λ), similar values were found among the three strategies. This indicates that the use of any valid initialization strategy improves searching time without producing significant effects on the characteristics of the solutions found.

In summary, the use of a valid initialization strategy significantly improves the searching time, driving exploration of the search space toward where it is more probable to find a solution. Since shorter searching times were obtained with a valid initialization, this strategy will be used for future comparisons between EAMP and other techniques. Although for fixed-length valid initialization strategy searching time at $p_m=0.01$ was similar to valid initialization in the range 0.01–0.05, this last one will be used because it is less sensitive to changes in the mutation rate. Finally, $p_m=0.04$ will be employed because the major difference between the three strategies is achieved with this value.

3.3. Comparison between classical search strategies and the proposed evolutionary algorithm

To compare the performance of EAMP against classical BFS and DFS algorithms, measures obtained for the search of metabolic pathways were compared. The EAMP was evaluated applying the parameters values set in the previous subsection.

BFS and DFS algorithms were modified to incorporate a control of repeated states, that allows to discard operators producing compounds previously generated in the pathway. In the case of the DFS, the maximum depth for the search was limited to 100 transformations to apply the same restriction as the one set for the EAMP.

Both classical algorithms explore the state space by building a search tree, where each node is a compound and the connections represent possible reactions among them (parent nodes are the substrates and child nodes are the products). Therefore, a pathway consists of a set of compounds and reactions that link them. However, while BFS explores all nodes with depth d(pathways with $d-1$ reactions) before those of depth $d+1$, DFS takes every pathway until it reaches a leaf node.

Since the set of solutions found by BFS and DFS algorithms is influenced by the order in which operators are applied, a single search may generate biased solutions due to their initial ordering. To avoid this effect, we use as many randomizations or randomized traversals as pathways searched by EAMP. For each randomization, the first pathway found was considered. Metrics for each algorithm were calculated as an average value over randomizations.

Results regarding the search time and the diversity in the lengths generated with the three methods are shown in Fig. 4. Boxplots for search times are shown in white and the length of the pathways found are shown in gray. In each diagram, the body of the box contains the central 50% of the data and the complete diagram reflects the variability of the analyzed measure. The box bottom and top limits correspond to the quartiles Q_1 and Q_3 respectively, and the segment dividing each box into two halves shows the position of the quartile Q_2 (median). As it can be observed in the figure, DFS found pathways in times close to 1 s, but the length of the pathways was biased to the maximum allowed value. Because metabolic pathways relating only two compounds and containing 100 transformations have no biological

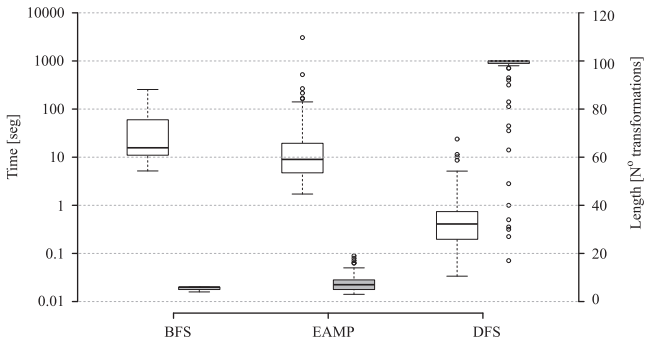

Fig. 4. Boxplot showing the search time and the length of the pathways for the results of EAMP, BFS and DFS algorithms. Search times are expressed in seconds and are shown in white; the length of the pathways found are expressed in number of transformations and are shown in gray. Time is plotted in logarithmic scale. Each box represents the central 50% of the data. Circles indicate atypical measurements. Segments extending outside the boxes indicate the maximum and minimum limits from which values are considered atypical.

Table 2
Comparison between BFS and EAMP. Time $\hat{t}$ is expressed in seconds and L in number of transformations. $|\Psi|$ indicates the number of compounds in each cluster.

Search ⟶		1	2	3	4	5	6
$\|\Psi\|$⟶			6			12	
Compounds to relate ⟶		62–47	258–77	47–258	37–82	135–65	135–82
$\hat{t}$	EAMP	17.6	8.6	8.3	3.4	15.2	8.4
	BFS	16.1	77.5	142.0	10.2	12.6	12.9
L_M	EAMP	13	11	17	9	19	18
	BFS	4	5	5	3	5	5
$\hat{L}$	EAMP	6.5	7	8	5	9	6
	BFS	4	5	5	3	5	5
L_m	EAMP	4	5	5	3	5	5
	BFS	4	5	5	3	5	5
ψ_M	EAMP	3	3	2	4	3	4
	BFS	2	2	2	3	2	2
$\overline{\psi}$	EAMP	2.4	2.1	2.0	2.3	2.1	2.1
	BFS	2.0	2.0	2.0	2.4	2.0	2.0
Λ	EAMP	0.50	0.50	0.33	0.33	0.33	0.33
	BFS	0.33	0.33	0.33	0.25	0.17	0.17

interest (high energetic inefficiency), results obtained with this algorithm were excluded from subsequent analyses.

Fig. 4 shows similar values for searching time, with mean values slightly lower for BFS (45.2 s) than for EAMP (51.2 s). This result is mainly due to a few cases where EAMP used searching times higher than 500 s. For practical purposes, they could be considered equivalents. EAMP found metabolic pathways with intermediate lengths to those of BFS and DFS. This result is interesting because, apart from finding the shortest pathways between compounds, it also found pathways with more diversity of sizes, offering alternatives that might be of interest for a biological analysis. For example, if the shortest pathway has 5 reactions, pathways with greater length could incorporate additional compounds of interest to produce the desired final compound through alternative pathways.

In order to analyze the results generated by EAMP and BFS in more detail, Table 2 shows measures obtained with each algorithm for each pair of compounds. The rows in the table correspond to the median of the search time ($\hat{t}$); the maximum, minimum, and median number of transformations (L_M, L_m and $\hat{L}$ respectively); the maximum and mean number of cluster compounds incorporated into the pathway (ψ_M and $\overline{\psi}$ respectively); and the explanation rate of the cluster (Λ). Columns indicate different pairs of compounds numbered from 1 to 6; the number of compounds in each one of the clusters previously found; and the compounds used in every run.

Table 2 shows that BFS uses similar or higher times than EAMP. Only for Searches 2, 3 and 4 these differences were significant ($p < 0.05$). The minimum pathways size was similar for BFS and EAMP showing that both algorithms can find the shortest pathways. Besides, EAMP found larger pathways than BFS, leading to the median be significantly greater ($p < 0.001$). This is desirable because indicates that EAMP finds metabolic pathways with reactions not included in the shortest pathways, providing information about alternative mechanisms to synthesize a metabolite. Measures related to the number of cluster compounds included into the pathways were similar for both algorithms, differing only in the Search 1 where EAMP related more cluster compounds than BFS ($p < 0.01$). The explanation rate of the clusters reflects that none of the algorithms presented a preference for incorporating cluster compounds into the solutions. However, EAMP presented slightly higher values for some cases as a result of the variability in the number of transformations of the solutions.

For a preliminary biological evaluation of the algorithms, the search of a metabolic pathway linking compounds C00103 (α−D−glucose−1P) and C00631 (glycerate-2P), both characteristic of glycolysis, was performed. Fig. 5 shows the classical glycolysis pathway and those found by EAMP (short dashed lines) and BFS (long dashed lines). Initial and final compounds are drawn as bold hexagons. Large gray rectangles indicate different pathways. Compounds (circles) and reactions (arrows) are shown with the KEGG codification. Dotted circles indicate compounds that are present in more than one pathway.

The pathway found by BFS employs five reactions, one less than the standard pathway of glycolysis, by taking a shortcut through the pentose pathway. The glycolysis and the pathway found with EAMP use six reactions for linking the compounds of interest. However, reactions for this last one belong to two different known pathways. Furthermore, only one reaction belonging to the glycolysis pathway (not participating in the sequence linking the desired compounds) is used by EAMP. When comparing both pathways found, it can be seen that although the pathway found by BFS contains fewer steps, it just replaces three of the six original reactions belonging to glycolysis. Instead, the pathway found with EAMP replaces all reactions from the standard pathway by reactions belonging to alternative routes, and uses only one reaction from glycolysis. The last one is very interesting because it provides a novel pathway employing a set of reactions different to the well-known glycolysis. Furthermore, this result indicates that the production of a compound may occur through different mechanisms than those already known.

Although solutions found by EAMP are biologically possible, the incorporation of additional information would lead the search towards solutions of particular interest. For example, the use of "atom tracking" as presented in [17] would allow to find metabolic pathways where a predefined fraction of the substrate has to be conserved into the product.

Finally, given that the fitness function and many parts of the algorithm can be easily modified, it could be applied to a wide range of applications. For example, it could be used to eliminate gaps in metabolic pathways [26,34,35] or the construction of heterologous pathways in metabolic engineering [36–38].

4. Conclusions and future work

This paper proposed an evolutionary algorithm for searching metabolic pathways between two compounds. The structure of the chromosomes was defined as a sequence of chemical transformations. Specific operators of crossover and mutation for the application domain were defined, to favor concatenation of related transformations. Different metrics were also proposed to compare the performance of the algorithms and assess the features of pathways found. It was noted that the use of a valid initialization reduces the search time, and that mutation rates lower than 0.05 favor this decrease. In the comparison of EAMP with BFS and DFS it was noted that the proposed algorithm used similar times to BFS. Although DFS is the fastest, as it would be expected, in most cases this algorithm founds pathways with the maximum allowed length, which strongly influences their usefulness. It must be highlighted that the EAMP generated a higher dispersion of lengths in the pathways, with intermediate values to those of the solutions found by the other two algorithms. From a biological point of view, this result is interesting because it shows that our algorithm can explore alternative mechanisms for the production of compounds. These alternatives include reactions belonging to

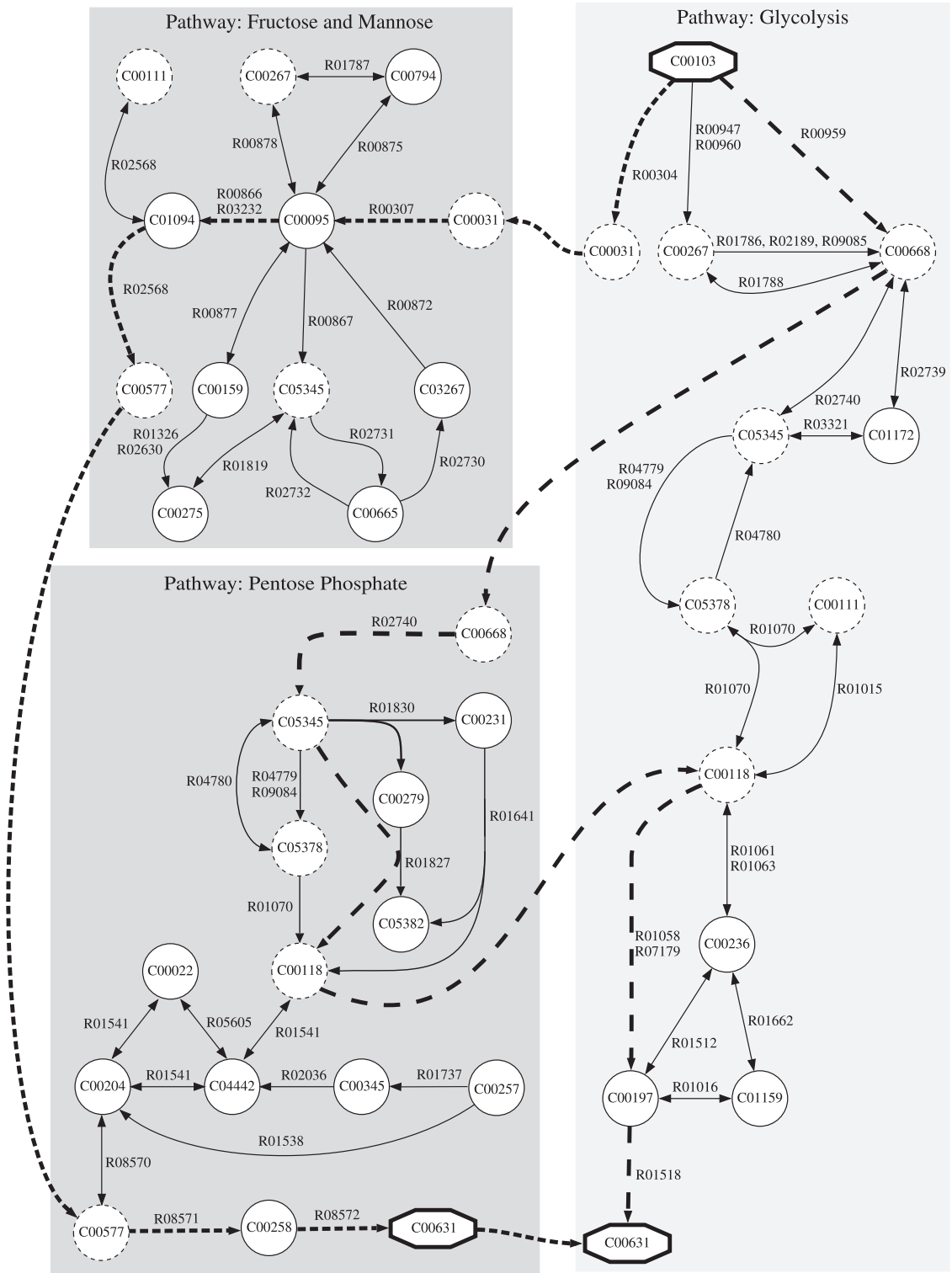

Fig. 5. Metabolic pathways linking compounds C00103 and C00631 found using EAMP (short dashed line) and BFS (long dashed line). Initial and final compounds are drawn as bold hexagons. Large gray rectangles indicate different pathways and their compounds (circles) and reactions (arrows), shown with the KEGG codification.

different metabolic pathways, which is well-known that could be involved in the metabolism.

As future work, we are considering the use of an initialization strategy that builds the chromosome starting from both compounds to relate. Furthermore, next steps will involve finding metabolic pathways linking more than two compounds, also incorporating information of enzymes and taking into account non-trivial cycles. In addition, other multi-objective evolutionary strategies will be considered.

Conflict of interest statement

None declared.

References

[1] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed, Prentice Hall, 2010.
[2] M. Kanehisa, S. Goto, KEGG: kyoto encyclopedia of genes and genomes, Nucleic Acids Res. 28 (1) (2000) 27–30.
[3] J. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975.
[4] I. Rechenberg, Evolutionsstrategie: Optimierung Technischer Systeme und Prinzipien der biologischen Evolution, Frommann-Holzboog, 1973.
[5] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.
[6] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, 1966.
[7] T. Bäck, U. Hammel, H. Schwefel, Evolutionary computation: comments on the history and current state, IEEE Trans. Evol. Comput. 1 (1997) 3–17.
[8] W. Langdon, R. Poli, Foundations of Genetic Programming, Springer, 2010.
[9] T. Bäck, D. Fogel, Z. Michalewicz, Evolutionary Computation I: Basic Algorithms and Operators, Institute of Physics Publishing, 2000.
[10] U. Morbiducci, A. Tura, M. Grigioni, Genetic algorithms for parameter estimation in mathematical modeling of glucose metabolism, Comput. Biol. Med. 35 (2005) 862–874.
[11] L. Zou, Z. Wang, Y. Wang, F. Hu, Combined prediction of transmembrane topology and signal peptide of β–barrel proteins: using a hidden Markov model and genetic algorithms, Comput. Biol. Med. 40 (2010) 621–628.
[12] S. Yu, M. Lee, Bispectral analysis and genetic algorithm for congestive heart failure recognition based on heart rate variability, Comput. Biol. Med. 42 (2012) 816–825.
[13] H. Ogata, S. Goto, W. Fujibuchi, M. Kanehisa, Computation with the KEGG pathway database, BioSystems 47 (1998) 119–128.
[14] J. Easton, L. Harris, M. Viant, A. Peet, T. Arvanitis, Linked metabolites: a tool for the construction of directed metabolic graphs, Comput. Biol. Med. 40 (2010) 340–349.
[15] D. Croes, F. Couche, S. Wodak, J. van Helden, Metabolic pathfinding: inferring relevant pathways in biochemical networks, Nucleic Acids Res. 33 (2005) W326–W330.
[16] D. McShan, S. Rao, I. Shah, PathMiner: predicting metabolic pathways by heuristic search, Bioinformatics 19 (2003) 1692–1698.
[17] A. Heath, G. Bennett, L. Kavraki, Finding metabolic pathways using atom tracking, Syst. Biol. 26 (2010) 1548–1555.
[18] C. Trinh, A. Wlaschin, F. Srienc, Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism, Appl. Microbiol. Biotechnol. 81 (2009) 813–826.
[19] M. Terzer, J. Stelling, Large-scale computation of elementary flux modes with bit pattern trees, Bioinformatics 24 (2008) 2229–2235.
[20] V. Acuña, A. Marchetti-Spaccamela, M. Sagot, L. Stougie, A note on the complexity of finding and enumerating elementary modes, Biosystems 99 (2010) 210–214.
[21] P. Carbonell, A. Planson, D. Fichera, J. Faulon, A retrosynthetic biology approach to metabolic pathway design for therapeutic production, BMC Syst. Biol. 5 (2011) 122.
[22] G. Stegmayer, D. Milone, L. Kamenetzky, M. López, F. Carrari, Neural network model for integration and visualization of introgressed genome and metabolite data, in: Proceedings of the 2009 International Joint Conference on Neural Networks, 2009.
[23] K. Saito, M.Y. Hirai, K. Yonekura-Sakakibara, Decoding genes with coexpression networks and metabolomics – 'majority report by precogs', Trends Plant Sci. 13 (2008) 36–43.
[24] E. Urbanczyk-Wochniak, et al., Parallel analysis of transcript and metabolic profiles: a new approach in systems biology, EMBO Rep. 4 (10) (2003) 989–993.
[25] R. Küffner, R. Zimmer, T. Lengauer, Pathway analysis in metabolic databases via differential metabolic display (DMD), Bioinformatics 16 (9) (2000) 825–836.
[26] P. Kharchenko, D. Vitkup, G. Church, Filling gaps in a metabolic network using expression information, Bioinformatics 20 (2004) i178–i185.
[27] M. Kotera, Y. Okuno, M. Hattori, S. Goto, M. Kanehisa, Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions, J. Am. Chem. Soc. 126 (2004) 16487–16498.
[28] K. Faust, D. Croes, J. van Helden, Metabolic pathfinding using RPAIR annotation, J. Mol. Biol. 388 (2) (2009) 390–414.
[29] F. Carrari, C. Baxter, B. Usadel, E. Urbanczyk-Wochniak, M. Zanor, A. Nunes-Nesi, V. Nikiforova, D. Centero, A. Ratzka, M. Pauly, L. Sweetlove, A. Fernie, Integrated analysis of metabolite and transcript levels reveals the metabolic shifts that underlie tomato fruit development and highlight regulatory aspects of metabolic network behavior, Plant Physiol. 142 (2006) 1380–1396.
[30] D.H. Milone, G. Stegmayer, L. Kamenetzky, M. López, J. Lee, J. Giovannoni, F. Carrari, *omeSOM: a software for clustering and visualization of transcriptional and metabolite data mined from interspecific crosses of crop plants, BMC Bioinf. 11 (2010) 438.
[31] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.
[32] D. Ruppert, Statistics and Data Analysis for Financial Engineering, Springer, 2011.
[33] B. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, 1993.
[34] J. Orth, B. Palsson, Systematizing the generation of missing metabolic knowledge, Biotechnol. Bioeng. 107 (2010) 403–412.
[35] L. Liu, R. Agren, S. Bordel, J. Nielsen, Use of genome-scale metabolic models for understanding microbial physiology, FEBS Lett. 584 (2010) 2256–2264.
[36] D. Na, T. Kim, S. Lee, Construction and optimization of synthetic pathways in metabolic engineering, Curr. Opinion Microbiol. 13 (2010) 363–370.
[37] J. Blazeck, H. Alper, Systems metabolic engineering: genome-scale models and beyond, Biotechnol. J. 5 (2010) 647–659.
[38] T. Osterlund, I. Nookaew, J. Nielsen, Fifteen years of large scale metabolic modeling of yeast: developments and impacts, Biotechnol. Adv. 30 (2012) 979–988.