# Optimization of GenoCAD Design Based on AMMAS

Yingjie Wang[1] and Yafei Dong[1,2(✉)]

[1] School of Computer Science, Shaanxi Normal University, Xian 710119, China
dongyf@snnu.edu.cn
[2] College of Life Science, Shaanxi Normal University, Xian 710119, China

**Abstract.** By successively click on prescribed grammar captured in successful genetic designs to assemble a range of genetic parts for example BioBricks, large and complicated genetic systems composed of substantial functional blocks can be constructed. As the number of genetic parts increases, each category of genetic parts includes so many parts that the process of assembling a great deal of genetic parts is costly, time-consuming and error-prone. GenoCAD is a web-based application for synthetic biology to guide users through the design of artificial gene networks, protein expression vectors and other complex genetic constructs by continuously click on predefined grammars according to the notion of genetic parts. However, at the last step of a design in GenoCAD, it's difficult for users to determine which basic part will be taken in every category. On the basis of statistical language model, a probability distribution over a string S reflecting how frequently a string S will occur and a mathematical model to select basic genetic parts to form a genetic construct can be determined. After converting the parts assembly process into a mathematical model, adaptive maximum-minimum ant system (AMMAS) proposed in this paper can be applied to the mathematical model to figure out an optimal combination of parts of a design with maximum probability automatically within seconds. The adaptive maximum-minimum ant system (AMMAS) can not only optimize the parts selection process of a design but also can devise particular projects performing specific functions based on former successful parts assembly experience. Consequently, redundant operations can be reduced and cost as well as time spent in biological experiments can be minimized drastically.

**Keywords:** GenoCAD · AMMAS · Statistical language model · Grammars

## 1 Introduction

The rapid development of synthetic biology makes it essential to develop methodologies to streamline the design of custom genetic systems [1]. Gene expression network, metabolic engineering and protein expression vector are some of applications in this field [2, 3]. GenoCAD, a web-based application for synthetic biology, can satisfy the needs of scientific studies in synthetic biology and allow users to quickly devise genetic constructs based upon the notion of genetic parts [4]. GenoCAD is built upon a solid computational linguistic foundation and can guide users through genetic designs by successively click


H. Han et al. (Eds.): IDMB 2019, CCIS 1099, pp. 254–271, 2020.
https://doi.org/10.1007/978-981-15-8760-3_19

on prescribed grammars capturing design strategies of specific applications [4]. Users, who elect to create a personal account, can log in the system to engineer project-specific parts libraries, upload new parts into their workspace and save designs for later use [5]. Designers usually decompose large biological sequences into functional blocks as genetic parts to expand parts database including promoter, terminator, plasmid backbone, gene and ribosome binding site (RBS) which are necessary for designing genetic constructs [6]. The compelling vision of libraries of biological parts enabling a fast and cheap assembly of large biological systems is one of the foundations of synthetic biology [7, 8]. There are several assembly standards to follow when assembling a set of genetic parts into genetic constructs and the BioBrick Foundation (BBF) has been favorable for promoting the BioBrick standard. A BioBrick standard compliant part is a DNA fragment flanked by a prefix and a suffix segment having particular restriction sites [9, 10]. In addition, two BioBrick compliant parts can be assembled together using multiple specific ligations and restriction digestions independent of both parts sequences, which indicates that any number of parts compliant with a same assembly standard can be assembled into a new complex genetic construct by means of specific restriction digestions and ligations.

When assembling series of genetic parts into intricate genetic systems, users commonly are unsure of selecting an appropriate basic part in a part category. In Consequence, it's always costly, error-prone and time-consuming for biologists to determine a combination of parts of a genetic construct with biological experiments. For the sake of minimizing time and cost spent in the parts assembly process, researchers have developed robotic platforms to automate the parts assembly process which can be used to devise genetic systems by continuous click on the pre-defined grammars to convert the structure of genetic designs. Ultimately users will choose a genetic part from every parts category to fulfill their designs [11]. However, with the development of synthetic biology, an increasing number of genetic parts are developed, which makes users confused to select a suitable part from every parts category at the last step of a design (Fig. 1). Therefore, this study is carried out to settle the problem of selecting a reasonable genetic part from every parts category to build a project-specific genetic constructs. Above all, statistical language model (SLM) is introduced to facilitate the parts assembly process by transforming the parts assembly process into a mathematical model according to interaction between parts. Applications of statistical language model are as diverse as speech recognition, machine translation, word segmentation, part of speech tagging and other natural language applications. The established mathematical model in this paper can be solved by statistical parameters extracted from BioBrick standard parts downloaded from iGEM website and the proposed AMMAS to work out an optimal combination of parts with maximum probability to accomplish a genetic construct. Taking former successful iGEM part assemblies and resulted statistical parameters into account, our algorithm can be used to minimize cost and time spent in the parts assembly process. Our suggested scheme can not only select an optimized combination of parts at the last step of a genetic design in robotic platforms for example GenoCAD, but also can devise new projects performing specific functions based on former successful experience.

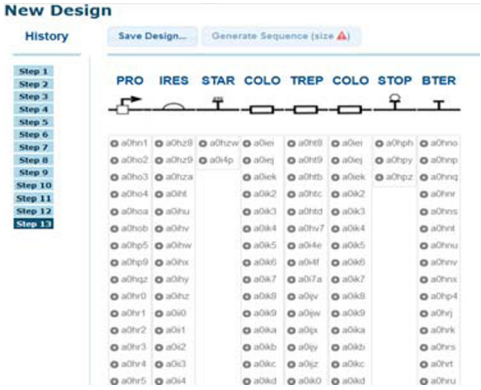


**Fig. 1.** Every icon has so many options.

## 2 Materials and Methods

A static snapshot of iGEM Registry content in June 2014 containing 7242 parts is available at http://parts.igem.org/das/parts/entry_points/ and Perl script is employed to parse out and analyze the content of each part at the link http://parts.igem.org/das/parts/features/?segment=part. Decomposed into a structured and unified data format, these parts can be imported into a relational database MySQL. Using SQL sentences, finally 75744 features sequences are acquired. Genetic parts comprise both basic parts which are unable to further divided into subparts such as promoter, terminator and ribosome binding site (RBS) [12, 13] and composed parts consisting of multiple basic parts such as system and device. Employing SQL sentences, basic parts and their usage frequencies in composed parts can be counted as well as composite parts and the usage frequencies of two adjacent basic parts (parts pair) in composed parts can be uniformly calculated. By querying the MySQL database, we extracted 1682 basic parts including 405 promoters, 57 terminators, 42 RBSs and 1178 genes, which can be used to build complicated genetic constructs. The experience of previous successful assembly of genetic constructs is utilized to guide us into the parts selection so that the resulted genetic constructs according to our algorithm will be reliable and scientific.

## 3 Mathematical Model

At the last step of a design in robotic platforms for example GenoCAD, there are so many choices in every parts category to complete the design making it a hard nut to crack to select the most suitable part from every part category (Fig. 1). It seems impossible and impractical to carry out exhaustive search method by testing all possible combinations of parts with wet biological experiments. And representing the structure of a genetic construct by using a reasonable mathematical model can simplify the process of choosing a suitable combination of parts [14]. To meet our expectations, statistical language model (SLM) is suggested to convert the parts assembly process into a mathematical model which takes the probability of occurrence of a sentence as a proof of its reasonableness.

A sentence S, denoted as a genetic construct, is composed of a strand of words which can be regarded as basic parts and the probability of a genetic construct can be evaluated accordingly.

$$S = part_1, part_2, \ldots, part_n \tag{1}$$

$$P(S) = P(part_1, part_2, \ldots, part_n) \tag{2}$$

In accordance with conditional probability formula, following formula (3) can be derived.

$$\begin{aligned} P(S) &= P(part_1, part_2, \ldots, part_n) \\ &= P(part_1) \cdot P(part_2|part_1). \\ &P(part_3|part_1, part_2). \\ &\ldots \cdot P(part_n|part_{n-1}, part_{n-2}, \ldots part_1) \end{aligned} \tag{3}$$

In above formula (3), $P(part_1)$ is the probability $part_1$ occurs in a design and $P(part_2|part_1)$ means the probability that $part_2$ appears with $part_1$ prior to it. Moreover, the probability $part_n$ occurs hinges on all the parts prior to it making it difficult to calculate $P(part_n|part_{n-1}, part_{n-2}, \ldots, part_1)$ compared with computing $P(part_1)$ and $P(part_2|part_1)$ owing to so many variables involved in it distinctly. According to Markov Hypothesis, the probability a part will occur is merely related to one or more parts prior to it. Hence formula (3) can be described in a simplified way as follows:

$$\begin{aligned} P(S) &= P(part_1, part_2, \ldots, part_n) \\ &= P(part_1) \cdot P(part_2|part_1) \cdot \\ &\ldots \cdot P(part_n|part_{n-1}) \end{aligned} \tag{4}$$

The formula (4) presented above is bi-gram of statistical language model which implies whether a part will appear in a design is simply concerned with one part prior to it. In line with the notion of conditional probability formula, conditional probability formulas involved in formula (4) can be deduced.

$$P(part_i|part_{i-1}) = \frac{P(part_{i-1}, part_i)}{P(part_{i-1})} \tag{5}$$

Utilizing usage frequencies of basic parts and parts pairs extracted from the downloaded feature sequences, we can estimate $P(part_i)$ and $P(part_i, part_{i-1})$ respectively.

$$P(part_i, part_{i-1}) \approx \frac{count(part_i, part_{i-1})}{count(all\_parts)} \tag{6}$$

$$P(part_i) \approx \frac{count(part_i)}{count(all\_parts)} \tag{7}$$

By means of above formulas (5–7), we can calculate conditional probability formulas involved in formula (4).

$$P(part_i|part_{i-1}) = \frac{count(part_{i-1}, part_i)}{count(part_{i-1})} \tag{8}$$

In this way, all components in formula (4) can be estimated meaning the combination of parts with maximum probability can be figured out. In accordance with the statistical language model (SLM), the combination of parts with maximum probability is the most meaningful and reasonable. There are too many candidate paths to accomplish a design and a path will lead to an $S$ (a path = an $S = part_1, part_2, \ldots, part_n$). The optimal path among all candidate paths can be represented by $PATH$.

$$\begin{aligned} PATH &= \arg\max_{all\_S}(P(S)) \\ &= \arg\max_{all\_S}((P(part_1) * \prod_{i=2}^{n} P(part_i|part_{i-1})) \end{aligned} \tag{9}$$

Since the database we use is extracted from a relatively sparse corpus, zero-frequency problem is inevitable when parts pairs never occur in the corpus meaning their corresponding *count* will be zero. This circumstance causes the deviation in calculating $P(S)$ and $PATH$. To overcome this difficulty, Add-k data smoothing technology is applied to settle the zero-frequency problem [15], and the corresponding conditional probability formulas involved in formula (4) can be gauged as follows.

$$P(part_i|part_{i-1}) = \frac{count(part_{i-1}, part_i) + k}{count(part_{i-1}) + k|W|} \tag{10}$$

Furthermore, the target function in this paper is represented by $f(S)$.

$$\begin{aligned} f(S) &= P(S) \\ &= P(part_1) * \prod_{i=2}^{n} P(part_i|part_{i-1}) \end{aligned} \tag{11}$$

$W$ is the total number of parts pairs and formula (10) is employed to replace the conditional probability formula (8). Hence all components in formula (4) can be gauged and the resulted $PATH$ is regarded as the optimal path (S) with maximum probability of occurrence among all candidate paths. After transforming the parts assembly process into a mathematical model, adaptive maximum-minimum ant system (AMMAS) can be exploited to automate this parts selection process efficiently.

## 4 Algorithms

The next step is to use the proposed algorithm to figure out a path, composed of a sequence of basic parts with the largest probability, in this lattice. This algorithm can direct us through the process of solving the target function $f(S)$.

The ant system algorithm, inspired by the observation of ant colonies in real world, was first proposed by Dorigo et al. in 1991 as a population-based approach to settle difficult combinatorial optimization problems [16–18]. Furthermore, an interesting and important behavior of ants is how to find a shortest path between their nest and food sources. While walking from the nest to food sources and vice versa, ants deposits a substance called pheromone on the ground, forming in this way a pheromone trail [19]. When deciding the direction to go, they choose paths which are marked by stronger

pheromone centration with higher probability. Ant's tendency to determine a specific path is positively correlated with the intensity of a found trail. The basic behavior is a foundation for a cooperative interaction relationship that results in the emergency of the shortest paths. The ant system algorithm has been applied to plenty of difficult combinatorial optimization problems such as the quadratic assignment problem [20, 21] and traveling salesman problem (TSP). The performance of ant system algorithm can be enhanced by introducing maximum and minimum trail strengths on arcs, named maximum and minimum ant system, to alleviate the problem concerning early stagnation. However, long runs of the maximum and minimum ant system (MMAS) still show stagnation behavior, despite of using minimum and maximum trail limits. Therefore, we raised three main changes to further improve its performance.

### 4.1 Simulated Annealing Mechanism

Beginning by randomly generating an initial solution, simulated annealing is a neighborhood search technique to resolve combinatorial problems. At each stage, a new solution taken from the neighborhood of the current solution will be accepted as a new current solution if it has a lower or equal cost; If it has a higher cost it will be accepted with a probability which decreases as the difference in cost between two solutions increases and as the temperature of the method decreases [22]. This temperature, simply a positive number, is reduced periodically according to the following formula, so that it can gradually from a relatively high value to near zero as the algorithm progresses. At the beginning of simulated annealing, most worsening moves are accepted, however, only improving ones are more likely to be accepted in the later stage of the algorithm. Furthermore, to enhance the intensity and diversity of simulated annealing searching procedure, when a solution doesn't show better performance within a prescribed number of continuous cooling process, the restarting solution mechanism, called tempering mechanism, is designed to generate new solutions for the further solution improving and the maximum number of tempering is $H_{\max}$.

$$T(N+1) = a \times T(N) \tag{12}$$

In addition, in order to avert premature convergence of the algorithm, the simulated annealing algorithm presented by the following formula (13) is carried out.

$$p = \begin{cases} \exp(-\frac{f(S_{global}) - f(S)}{T(t)}) & f(S) \leq f(S_{global}) \\ 1 & f(S) > f(S_{global}) \end{cases} \tag{13}$$

Where $S_{global}$ is the global optimal route and $S$ is a collection of paths resulted in this round. To make the performance of our proposed algorithm more robust, comparing the calculated $p$ by taking out solutions from $S$ one by one with a random number $\gamma$ within the interval [0, 1] is necessary. It is noted that a good quality solution can be confirmed in the case of $p = 1$ or $p > \gamma$; Otherwise a path defined as a bad solution will be determined. Moreover, the temperature reduction factor $a$ of 0.9 is chosen, which has been indicated to be satisfactory in the gradual temperature reduction process [23, 24].

### 4.2 Adaptive Pheromone Concentration Updating Mechanism

The pheromone updating mechanism is designed to allocate a great amount of pheromone concentration to short tours, in a sense, which is similar to the reinforcement learning schema. It is widely recognized that better solutions will get a higher reinforcement. The pheromone updating formula was intended to simulate the change of the amount of pheromone in virtue of both the addition of new pheromone deposited by ants on the visited edges and of pheromone evaporation. Ants have memory ability, however, as time goes on, information is lost. In order to prevent the algorithm from getting into local optimum due to large differences of pheromone density between the worst path and the best path, pheromone updating formula is designed to dynamically adjust the volatilization coefficient of pheromone as follows.

$$\rho(N+1) = 1/\log_2(1+N_c) \tag{14}$$

In this research, the volatilization coefficient of pheromone $\rho$ reduces gradually up to $1/\log_2(1+N_c) < \rho_{\min}$.

### 4.3 Adaptive Change of Weight Coefficient

As previously noted, to intensify and diversify the searching procedure and to make the solution found more robust, a dynamic change mechanism of weight coefficient $\alpha$ and $\beta$ is designed to fulfill the purpose when the current global optimal path does not change within 50 rounds. The idea is to maintain a high ability to search for new solutions and prevent algorithm from getting into local optimum not only by reducing the relative influence of pheromone, but also by increasing the relative influence of the heuristic information, presented as follows, thus the goals of intensification and diversification of the algorithm can be achieved.

$$\beta = sl/(3 \cdot sum\_column) \tag{15}$$

Where *sum_column* is the total number of columns of the built lattice. In the above formula, *sl* reflects the total number of ants which go through all edges of the optimal solution of one iteration.

### 4.4 State Transition Rules

To achieve better balance between using prior knowledge and exploring new paths, the pseudo-random rate rule is selected when ant k chooses next node $j$ from node $i$, specifically described as follows.

$$j = \begin{cases} \arg \max\limits_{u \in allowed_k} \left\{ [\tau_{iu}(t)]^{\alpha} * [\eta_{iu}(t)]^{\beta} \right\}, & if\ q \le q_0 \\ p_{ij}^{k}(t), & else \end{cases} \tag{16}$$

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{(\tau_{ij}(t))^{\alpha}(\eta_{ij}(t))^{\beta}}{\sum\limits_{l \in Allowed_k(i)} (\tau_{il}(t))^{\alpha}(\eta_{il}(t))^{\beta}} & j \in Allowed_k(i) \\ 0 & else \end{cases} \tag{17}$$

where $q$ is a random number with uniform distribution in [0, 1] and variable $q_0$, defined as below, determines the relative importance degree between using prior information and exploring new paths. $\alpha$ and $\beta$ emphasize the importance degree of pheromone concentration and heuristic information respectively while $\eta_j$ represents the heuristic information guiding the selection of the next node.

$$q_0 = sum\_column \Big/ sl \tag{18}$$

$$\begin{aligned} \eta_j &= P(part_j | part_i) \\ &= \frac{P(part_i, part_j)}{P(part_i)} \end{aligned} \tag{19}$$

### 4.5 The AMMAS Algorithm

Then the improved ant colony system is designed to solve the problem of biological parts selection and it consists of two steps as below.

First of all, a lattice is created. Each column, expressed by one icon, corresponds to one parts category and every node in each column refers to as a basic part. Part name is unique in the workspace.

Afterwards, the improved algorithm, composed of five steps, can be applied to solve the built statistical language models and the algorithm flow chart is as follows.

Step 1: The initial pheromone information $\tau_{ij} = \tau_{\max}$ (a constant), the iteration counter of the algorithm $N_c$, an adjustable parameter $q_0$ involved in the state transition rule, the maximum iteration $N_{\max}$, the maximum number of nodes $N_m$ an ant passes in each round are all to be initialized. The variable $n$ is the number of nodes one ant goes through and it is set to $n = 1$. To ensure that the pheromone distribution varies along the paths at the beginning of the algorithm, the threshold $U$ ($U \leq 10$) can be employed to determine whether ants choose next node according to formula (17) or not in the early stage of the algorithm. $m$ ants are randomly assigned to all nodes of the first column of the established lattice.

Step 2: If $N_c \leq U$, ant $k$ chooses next node in accordance with formula (17) and makes a step forward according to the selected node as well as places that node into its taboo list $Tabu_k$. If $N_c > U$, ants will choose appropriate node selection formula based on the comparison between $q_0$ and $q$.

Step 3: If the number of nodes ant $k$ passed in this round has not reached the given number $N_{\mathrm{m}}$, the algorithm will go to Step 2 to decide next node, on the contrary go to Step 6.

Step 4: When all ants complete this iteration, the route with maximum probability among all candidate paths in this iteration can be gained. And volatilization factor of pheromone $\rho$ can be changed based on formula (14). In addition, pheromone information on all paths can be updated globally, however, updating global pheromone information in this way can't well guide ants towards the global optimum. In view of this situation, the introduction of reward and punishment mechanism regarding better and worse solutions respectively is necessary. Different edges have diverse impacts on guiding ants towards global optimal solutions. Considering this characteristic, we are intended to allocate

more pheromone on good paths and less pheromone on bad paths as below. Updating pheromone concentration in this way can accelerate the convergence speed.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{good} \tag{20}$$

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) - \sum_{k=1}^{m} \Delta\tau_{ij}^{bad} \tag{21}$$

Where

$$\Delta\tau_{ij}^{good} = w \cdot f(s^{best}) \tag{22}$$

$$\Delta\tau_{ij}^{bad} = w \cdot f(s^{worst}) \tag{23}$$

Where $w(0 \leq w \leq m)$ expresses the total number of edge $(i, j)$ appearing in all candidate paths. Moreover, $f(s^{best})$ and $f(s^{worst})$ depict best and worst value of the target function in this round respectively.

Step 5: Compared with global optimal path, if the optimal path obtained in this iteration is better than the current global optimal one, the global optimum will be replaced. In order to jump out of the local optimal and expand the search range, $\beta$ can be further dynamically changed if global optimal path remains unchanged within fifty rounds.

The temperature $T$ should be updated according to $T \leftarrow aT, t \leftarrow t+1$. If $T \geq T_{\min}$, the proposed algorithm goes to Step 2 to start a new iteration. If $T < T_{\min}$ and $H = H_{\max}$, the algorithm outputs the saved global optimized path. If $T < T_{\min}$ and $H < H_{\max}$, then $H \leftarrow H+1, T \leftarrow T_{\max}$ and it goes to Step 2.

Ultimately, the optimal combination of parts to form a genetic construct can be worked out by the algorithm in seconds after entering the parts category sequences, which is also an optimal solution with maximum probability to the target function $f(S)$. In comparing the proposed algorithm with exhaustive algorithm, one must bear in mind that the running time of the former is on the order of seconds, which is acceptable in synthetic biology (Fig. 2).

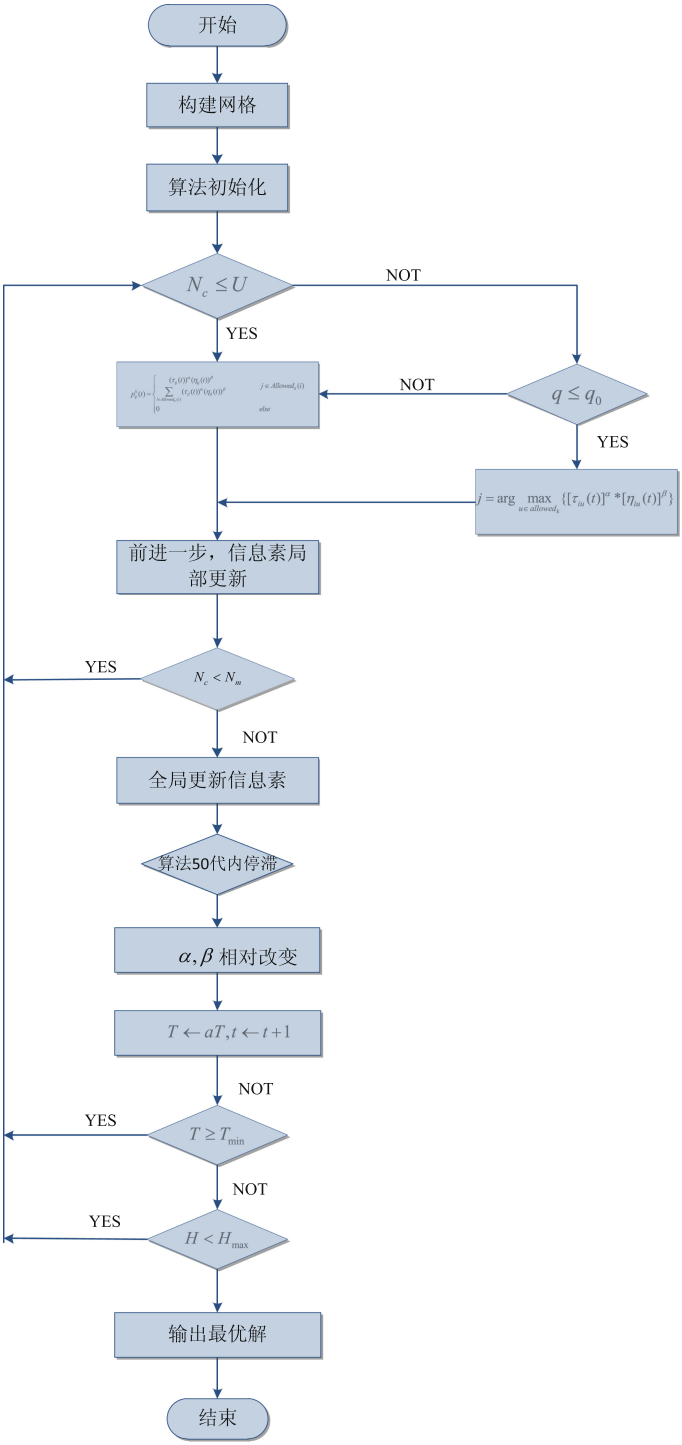


**Fig. 2.** The algorithm and flow chart.

## 5 Result

Developing grammars for modeling the structure of genetic constructs has become a routine tool in synthetic biology [25]. There is a need for simple and versatile design strategies to allow high throughput approaches in synthetic biology studies (rule-based design). Therefore, we implemented a set of rules to design genetic constructs based on the basic grammar which is PRO (promoter)-RBS (ribosome binding sites)-GEN (genes)-TERM (terminator) [26]. The full grammatical model, similar to the context-free grammar (CFG), used in this system is available in Table 1.

**Table 1.** Grammars used in this paper

| Rule code | Rule | Description |
|---|---|---|
| 1 | CAS → 2CAS | Convert a cassette into two cassettes (CAS) |
| 2 | CAS → Pro CIS TERM | Convert a cassette into a promoter (Pro), a cistron (CIS), a terminator (TERM) |
| 3 | CAS → Pro CIS | Convert a cassette into a promoter (Pro) and a cistron (CIS) |
| 4 | CIS → 2CIS | Convert a cistron into two cistrons (CIS) |
| 5 | CIS → RBS GEN | Convert a cistron into a rbs (RBS) and gene (GEN) |
| 6 | TERM → 2TERM | Convert a terminator into two terminators (TERM) |
| 7 | GEN → 2GEN | Convert a gene into two genes (GEN) |

To describe how to assemble series of parts compliant with BioBrick standard into a functional biosynthetic system by our algorithm, we select wintergreen odor biosynthetic system (http://parts.igem.org/Part:BBa_J45700), designed and implemented by MIT iGEM 2006. This system includes two expression cassettes: one can produce salicylate acid from cellular metabolic and the other can catalyze the conversion of the salicylate acid to methyl salicylate or wintergreen odor. We can perform the following grammatical model to direct users through the wintergreen odor biosynthetic system. Starting with a CAS and by means of rule1, the design becomes CAS-CAS and the following design is PRO-CIS-TERM-PRO-CIS-TERM by applying rule2 to both CAS. Employing rule5 to both CIS, the design turns into PRO-RBS-GEN-TERM-PRO-RBS-GEN-TERM and finally it becomes PRO-RBS-GEN-TERM-TERM-PRO-RBS-GEN-TERM-TERM according to rule6. After determining genes we want to express, our algorithm in Python language can be applied to choose an optimal parts combination automatically for the input parts category sequence which becomes PRO-RBS-J45004-TERM-TERM-PRO-RBS-J45017-TERM-TERM to form the system. The resulted combination of parts by our bi-gram model algorithm is R0040-B0032-J45004-B0010-B0012-R0010-B0032-J45017-B0010-B0012. Compared with the validated combination of parts of the wintergreen odor biosynthetic system R0040-B0032-J45004-B0010-B0012-R0011-B0032-J45017-B0010-B0012, the simulation result from our algorithm (Fig. 3) is very similar to that verified one of this system.

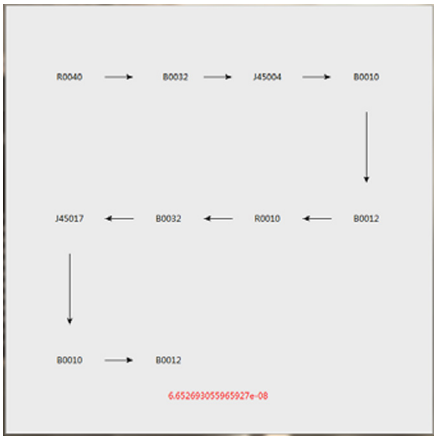


**Fig. 3.** The simulation results of the first system.

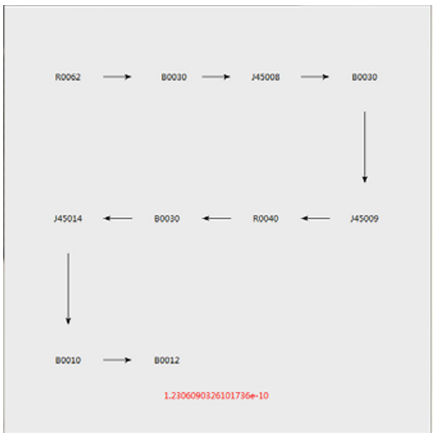


**Fig. 4.** The simulation results of the second system.

Moreover, for another example, we elect the banana odor biosynthetic system (http://parts.igem.org/Part:BBa_45900), implemented and designed by teams enrolling in iGEM in 2006. This system includes two expression cassettes: one can produce isoamyl alcohol with *BAT*2 and *THI*3 and the other can catalyze the conversion of the cellular metabolite leucine to isoamyl acetate or banana odor. We can implement the following grammatical model to direct users through the banana odor biosynthetic system. Starting with a CAS and by means of rule1, the design becomes CAS-CAS and the following design is PRO-CIS-PRO-CIS-TERM by applying rule3 to the first CAS and rule2 to the second CAS. Employing rule4 to first CIS and rule5 to the second CAS, the design turns into PRO-CIS-CIS-PRO-RBS-GEN-TERM and it becomes PRO-RBS-GEN-RBS-GEN-PRO-RBS-GEN-TERM according to rule5. Finally the design is PRO-RBS-GEN-RBS-GEN-PRO-RBS-GEN-TERM-TERM according to rule6. After determining genes to be expressed in this design, our algorithm in Python language can be utilized to pick out an optimal parts combination automatically for the input parts category sequence which becomes PRO-RBS-J45008-RBS-J45009-PRO-RBS-J45014-TERM-TERM to form the system. The resulted parts series of this design by our bi-gram model algorithm is R0010-B0030-J45008-B0030-J45009-R0040-B0030-J45014-B0010-B0012. Compared with the validated combination of parts of the banana odor biosynthetic system R0011-B0030-J45008-B0030-J45009-R0040-B0030-J45014-B0010-B0012, the simulation result from our algorithm (Fig. 4) is very close to that verified one of this system.

In addition, we select the design RBS.GFP + PBad CFP (http://parts.igem.org/Part:BBa_I13404), also designed and implemented by the team participating in iGEM in 2006, as another example to illustrate efficiency of our algorithm. Under the input design RBS-E0040-TERM-TERM-PRO-RBS-E0020–TERM-TERM, the bi-gram model algorithm proposed in this paper recommends the parts series B0034-E0040-B0010-B0012-R0010-B0034-E0020-B0010-B0012. As can be seen during comparison with the actual

combination of parts of the system B0034-E0040-B0010-B0012-I0500-B0034-E0020-B0010-B0012, our simulation result (Fig. 5) is quite close to the valid one of the system.

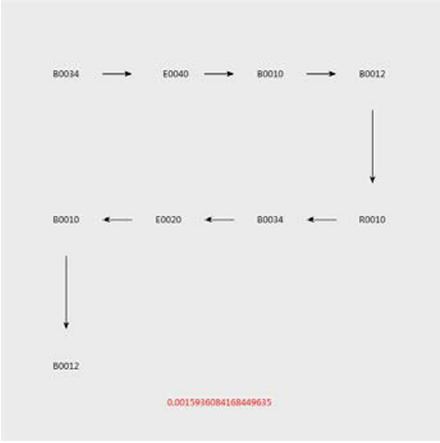


**Fig. 5.** The simulation results of the third system.

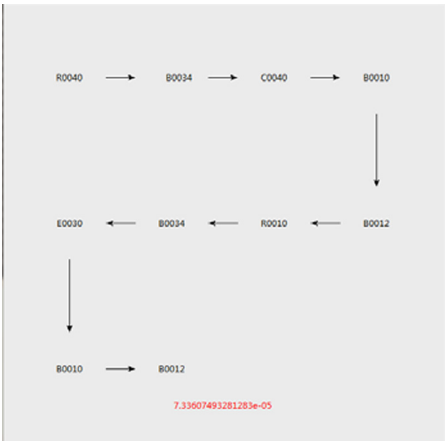


**Fig. 6.** The simulation results of the fourth system.

For the fourth example, we select the example I0500.Q04400.E0430 (http://parts.igem.org/Part:BBa_E0611) which was designed and implemented by the group taking part in iGEM in 2004. Under the input part category sequence PRO-RBS-C0040-TERM-TERM-PRO-RBS-E0030-TERM-TERM for this genetic construct, our bi-gram model algorithm recommends a combination of parts R0040-B0034-C0040-B0010-B0012-R0010-B0034-E0030-B0010-B0012. According to the comparison between simulation results and real results I0500-B0034-C0040-B0010-B0012-R0040-B0034-E0030-B0010-B0012, there are two basic parts that differ from the real parts series in simulation ones (Fig. 6).

Furthermore, we select a design from the link http://parts.igem.org/Part:BBa_S01664, designed and implemented by the team participating in iGEM in 2004, as the fifth example. Under the input design PRO-RBS-C0051-TERM-TERM-PRO-RBS-C0012-TERM-TERM-PRO, the bi-gram model algorithm suggests the parts sequence R0040-B0034-C0051-B0010-B0012-R0010-B0034-C0012-B0010-B0012-R0010. As is illustrated during comparison with the actual combination of parts of the system R0040-B0034-C0051-B0010-B0012-R0051-B0034-C0012-B0010-B0012-R0011, our simulation result (Fig. 7) is similar to the verified one of the system.

The example [TetR][rbs][LuxR][dblTerm][LuxPR] + [rbs][LacI][dblTerm] "AHL-dependent inverter" (http://parts.igem.org/Part:BBa_J23040), designed and implemented by the group participating in iGEM in 2006, is chosen as the sixth example. Based on the input parts category sequence of the design PRO-RBS-C0062-TERM-TERM-PRO-RBS-C0012-TERM-TERM, a set of parts can be figured out (R0040-B0034-C0062-B0010-B0012-R0010-B0034-C0012-B0010-B0012) by our algorithm according to the specifications required for this design. In comparison with the real parts

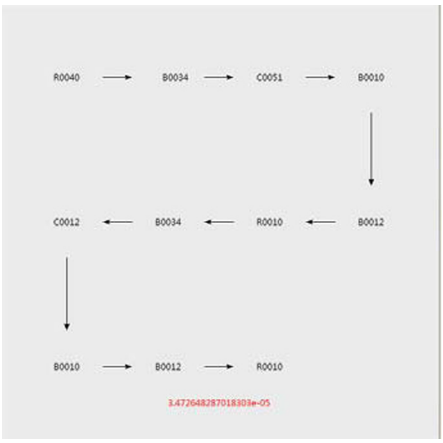


**Fig. 7.** The simulation results of the fifth system.

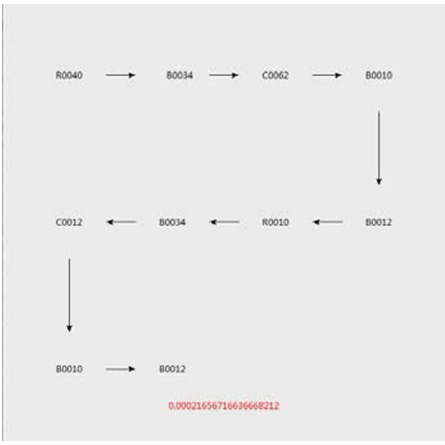


**Fig. 8.** The simulation results of the sixth system.

combination of the design R0040-B0034-C0062-B0010-B0012-R0062-B0034-C0012-B0010-B0012, our simulated result (Fig. 8) has only one different basic part from the real one.

We consider the example QPI Test Construct Intermediate (Q04121.E0430) (http://parts.igem.org/Part:BBa_I13021), designed and implemented by the group joining in iGEM in 2004, as the seventh example. To meet specific needs of the design, our algorithm presents a genetic construct B0034-C0012-B0010-B0012-R0010-B0034-E0030-B0010-B0012 within seconds on the basis of the input parts series RBS-C0012-TERM-TERM-PRO-RBS-E0030-TERM-TERM. It is obvious that our simulation result (Fig. 9) is pretty similar to the valid series of parts of the design B0034-C0012-B0010-B0012-R0011-B0034-E0030-B0010-B0012.

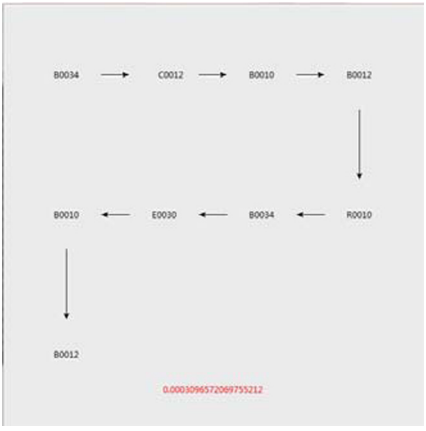


**Fig. 9.** The simulation results of the seventh system.

Taken together, simulation results of our algorithm AMMAS are exactly similar to the actual parts combination of genetic designs. Compared with the dynamic programming algorithm used in reference [26] to settle the 3-gram or 4-gram model, our bi-gram model algorithm can also figure out an assembly highly similar to the real parts combination of a design with fewer variables and less computation amounts, which prove the practical significance of our scheme. Moreover, dynamic programming algorithm is a classical implementation of the idea that sacrificing space to improve efficiency. When the size of problem $n$ increases, in theory, heuristic algorithms always present better performance in solving such combinatorial optimization problems. If some parts are known to express in a design, users can decide them first or evaluate them higher when entering the parts series. We can also exclude some parts and iterate over our algorithm to meet the needs of some other options needed for a design, which is instructive for synthetic biologists to design new projects. Using the extracted statistical parameters and the proposed adaptive maximum-minimum ant system (AMMAS) to solve the 2-gram mathematical model, the resulted optimal combination of parts is scientific and reliable.

In above cases, R0010 always appears at the same time as B0012. One of the reasons for this is that the database we use is sparse, and the other reason is that the bi-gram statistical language model can't well reveal the interaction relationships between parts categories. To address this issue, it is essential to adopt higher-gram model to depict the interaction relationships between parts categories.

## 6 Discussion

This paper presents an efficient algorithm AMMAS to guide users through the design of genetic constructs performing specific function by selecting an optimal parts combination for a genetic construct at the last step of a design in GenoCAD and to devise projects meeting specific requirements. Utilizing the concept of statistical language model and conditional probability, the parts assembly process can be converted into a mathematical model. The parts assembly process being transformed into a bi-gram model, adaptive maximum-minimum ant system (AMMAS) can be carried out to choose an optimal solution with the largest probability. In addition to selecting an optimized parts combination at the final step of a design in robotic platforms for example GenoCAD, this method can be used to automate DNA assembly process as well. We entering the parts category sequence of a design, our algorithm can work out a set of suitable parts to form the genetic construct automatically based upon the previous successful assemblies on iGEM website. In this way, redundant operations and time as well as cost spent in biological experiments can be minimized greatly. As depicted above, bi-gram of statistical language model proposed in this paper signifies that whether a part can be enabled in a design is simply related to one part prior to it, which can't well reveal the mechanism of DNA assembly process in real world. For example, whether a gene will be expressed efficiently is not only concerned with its promoter, but also its RBS and plasmids backbone as well as other regulating sequences. To simulate the parts assembly process in real world, higher-gram models should be introduced. Higher than 2-gram models indicate that one part involved in a design is related to more than one part prior to it. However, there are so many variables involved in these higher-gram models making calculating

conditional probability formulas a hard nut to crack. When developing higher-gram models, computers of high performance are necessary [27] though the accuracy of the results has improved greatly.

Since the dataset we used was extracted from a relatively sparse corpus, zero-frequency issue is inevitable when some parts pairs never appear. When calculating conditional probability formulas involved in the mathematical model, we employed Add-k smoothing technology to address the zero-frequency issue. However, it's not a good idea to utilize Add-k smoothing due to its disadvantages such as considerable amount of the probability space allocated to unseen events. It is used just for simplicity. Therefore, other smoothing techniques will be considered to improve the accuracy of the results such as Katz Backoff smoothing, Good-Turing smoothing, Witten-Bell and so on [28]. Some parts are likely to appear in any analysis so that the simulation results of the algorithm have a certain deviation from the real values. The reason may be that the existence of noisy data in the dataset results in the deviation. We intend to adopt the commonly used basic parts and parts pairs with high usage frequency (more than three times for basic parts and parts pair) in the corpus next. The commonly used basic parts and parts pairs with high usage frequency (more than three times for basic parts and parts pair) are regarded as successful words while the others are referred to as noisy data. In addition to improving data smoothing techniques, it is also of great importance to expand the database. However, expanding the corpus needs more operations to represent the notion and definition of parts and features in a unified format. That is, we should eliminate inconsistencies between features and redundant data in the corpus. The problem can be resolved by developing the ontology giving the community a controlled vocabulary to depict parts and features in a uniform format. And developing the synthetic biology open language (SBOL) will accelerate this process remarkably.

Based upon statistical language model, we present an efficient computational supplement for designs in robotic platforms of synthetic biology for example GenoCAD. In synthetic biology, it's an important question that too many choices are offered at the last step of a design. It's a matter of considerable interest to take the previous successful assemblies into account when we develop new projects. For those who don't have the expertise in synthetic biology, it's fairly difficult to elect a suitable part in a particular category. Users can choose a train of suitable parts to form a design according to a body of existing experiences by our algorithm. Our newly proposed method will facilitate the popularity of synthetic biology to a wider community and can help to eliminate inconsistencies in this field. In the future, further successful assemblies will be considered and we can devote ourselves to developing efficient algorithm to guarantee the reliability of results.

**Funding.** Natural Science Foundation of China [61572302].

**Conflict of Interest Statement.** None declared.

## References

1. Goler, J.A., Bramlett, B.W., Peccoud, J.: Genetic design: rising above the sequence. Trends Biotechnol. **26**(10), 538–544 (2008)

2. Graslund, S., Nordlund, P., Weigelt, J., et al.: Protein production and purification. Nat. Methods **5**(2), 135–146 (2008)
3. Ghaemmaghami, S., Huh, W.K., Bower, K., et al.: Global analysis of protein expression in yeast. Nature **425**(6959), 737–741 (2003)
4. Czar, M.J., Cai, Y., Peccoud, J.: Writing DNA with genoCAD. Nucleic Acids Res. **37**, W40–W47 (2009)
5. Cai, Y., Wilson, M.L., Peccoud, J.: GenoCAD for iGEM: a grammatical approach to the design of standard-compliant constructs. Nucleic Acids Res. **38**(8), 2637–2644 (2010)
6. Isaacs, F.J., Dwyer, D.J., Ding, C., et al.: Engineered riboregulators enable post-transcriptional control of gene expression. Nat. Biotechnol. **22**(7), 817–841 (2004)
7. Endy, D.: Foundations for engineering biology. Nature **438**(7067), 449 (2005)
8. Baker, D., Church, G., Collins, J., et al.: Engineering life: building a FAB for biology. Sci. Am. **294**(6), 44–51 (2006)
9. Arkin, A.: Setting the standard in synthetic biology. Nat. Biotechnol. **26**(7), 771–774 (2008)
10. Canton, B., Labno, A., Endy, D.: Refinement and standardization of synthetic biological parts and devices. Nat. Biotechnol. **26**(7), 787–793 (2008)
11. Coll, A., Wilson, M.L., Gruden, K., Peccoud, J.: Rule-based design of plant expression vectors using GenoCAD. PLoS ONE **10**(7), e0132502 (2015)
12. Gardner, T.S., Cantor, C.R., Collins, J.J.: Construction of a genetic toggle switch in Escherichia coli. Nature **403**(6767), 339 (2000)
13. Atkinson, M.R., Savageau, M.A., Myers, J.T., et al.: Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. Cell **113**(5), 597–607 (2003)
14. Ellis, T., Wang, X., Collins, J.J.: Diversity-based, model-guided construction of synthetic gene networks with predicted functions. Nat. Biotechnol. **27**(5), 465–471 (2009)
15. Bahaaddini, M., Hagan, P.C., Mitra, R., Hebblewhite, B.K.: Parametric study of smooth joint parameters on the shear behaviour of rock joints. Rock Mech. Rock Eng. **48**(3), 923–940 (2015)
16. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. EC **1**(1), 53–66 (1997)
17. Dorigo, M., Gambardella, L.M.: Ant colonies for the travelling salesman problem. Bio Syst. **43**(2), 73 (1997)
18. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B Cybern. **26**(1), 29 (2002). A Publication of the IEEE Systems Man & Cybernetics Society
19. Watanabe, I., Matsui, S.: Improving the performance of ACO algorithms by adaptive control of candidate set. In: Congress on Evolutionary Computation, pp. 1355–1362. IEEE (2003)
20. Maniezzo, V., Colorni, A., Dorigo, M.: The ant system applied to the quadratic assignment. IEEE Trans. Knowl. Data Eng. **11**(5), 769–778 (1994)
21. Gambardella, L.M., Taillard, E., Dorigo, M.: Ant colonies for the quadratic assignment problem. J. Oper. Res. Soc. **50**(2), 167–176 (1999)
22. Low, C., Yeh, J.Y., Huang, K.I.: A robust simulated annealing heuristic for flow shop scheduling problems. Int. J. Adv. Manuf. Technol. **23**(9–10), 762–767 (2004)
23. Van Laarhoven, P.J.M., Aarts, E.H.L., Lenstra, J.K.: Job shop scheduling by simulated annealing. Oper. Res. **40**(1), 113–125 (2007)
24. Low, C., Wu, T.H.: Mathematical modelling and heuristic approaches to operation scheduling problems in an FMS environment. Int. J. Prod. Res. **39**(4), 689–708 (2010)
25. Cai, Y., Hartnett, B., Gustafsson, C., et al.: A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. Bioinformatics **23**(20), 2760–2767 (2007)

26. Fang, G., Zhang, S., Dong, Y.: Optimizing DNA assembly based on statistical language modeling. Nucleic Acids Res. **45**(22), e182 (2017)
27. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press, Cambridge (1997)
28. Katz, S.M.: Estimation of probabilities from sparse data for the language model component of a speech recogniser. IEEE Trans. Signal Process. **35**(3), 400–401 (1987)