Syst Synth Biol (2010) 4:215–225
DOI 10.1007/s11693-010-9062-3

RESEARCH ARTICLE

The GeneOptimizer Algorithm: using a sliding window approach to cope with the vast sequence space in multiparameter DNA sequence optimization

David Raab · Marcus Graf · Frank Notka · Thomas Schödl · Ralf Wagner

Received: 19 October 2009 / Revised: 16 July 2010 / Accepted: 2 August 2010 / Published online: 1 September 2010

Abstract One of the main advantages of de novo gene synthesis is the fact that it frees the researcher from any limitations imposed by the use of natural templates. To make the most out of this opportunity, efficient algorithms are needed to calculate a coding sequence, combining different requirements, such as adapted codon usage or avoidance of restriction sites, in the best possible way. We present an algorithm where a "variation window" covering several amino acid positions slides along the coding sequence. Candidate sequences are built comprising the already optimized part of the complete sequence and all possible combinations of synonymous codons representing the amino acids within the window. The candidate sequences are assessed with a quality function, and the first codon of the best candidates' variation window is fixed. Subsequently the window is shifted by one codon position. As an example of a freely accessible software implementing the algorithm, we present the Mr. Gene web-application. Additionally two experimental applications of the algorithm are shown.

Keywords Gene synthesis · Codon optimization · Sequence optimization algorithm · Synthetic genes · Expression optimization

D. Raab · M. Graf · F. Notka · T. Schödl · R. Wagner (✉)
Geneart AG, Im Gewerbepark B35,
93059 Regensburg, Germany
e-mail: ralf.wagner@geneart.com

R. Wagner
Institute for Medical Microbiology and Hygiene,
University Regensburg, Regensburg, Germany

Introduction

In many cases de novo gene synthesis has become the preferred access route to biological DNA sequences. As the prices for synthetic genes have dropped considerably over the past years, today gene synthesis often economically outcompetes the classic genetic engineering methods. Another great advantage over the use of naturally occurring templates is the fact that the synthetic gene can literally be designed on an "electronic drawing board". Not only is it possible to freely add genetic elements such as a promoter or restriction sites flanking the coding region of a gene, but also to optimize the coding sequence itself for specific experimental requirements. This is possible due to the fact, that nearly all amino acids can be encoded by up to six different codons. Therefore, the DNA sequence can be altered without changing the corresponding amino acid sequence.

One of the most common applications is adaption of the used codons to the specific codon usage of a heterologous expression host. In the simplest form of optimization, a plain backtranslation is performed, where each amino acid is represented by the specific synonymous codon most frequently used in highly expressed genes of the production host. Obviously this procedure can lead to the generation of undesired restriction sites, expression constraining sequence elements, repetitive base stretches, etc. On the other hand, it may be desirable to introduce certain DNA motifs or avoid similarities to naturally occurring sequences. So the challenge is to find *the* sequence, which represents the best compromise between different and sometimes conflicting requirements. Without doubt, the best solution would be to generate all possible combinations of codons representing a given amino acid sequence, assess all of them with the help of a quality function and finally choose the one with the highest quality score.

Unfortunately the number of possible combinations is in the range of 10e47 even for a rather small protein of 100 amino acids, making the outlined approach impossible to perform in practice.

One possibility to reduce the sequence space, that has to be evaluated is to rely on statistical optimization methods. In multiple iterations synonymous codons are exchanged at random, while the choice of a certain synonymous codon may be controlled by a probability distribution based on the codon usage of the host organism (Villalobos et al. 2006). In each iteration the resulting sequence is assessed with respect to the desired parameters, and if the codon change leads to an improvement of the overall quality score, the changes are kept, otherwise they are discarded. In a variation of the method, known as "simulated annealing", also codon changes leading to a worse sequence score are allowed. In this case, the probability of acceptance for a new sequence is controlled by a "temperature parameter"-dependent Boltzmann distribution. This means that in the beginning of the optimization process, when the "temperature" is high, changes leading to a worse overall score are more likely accepted than in later iterations with a lower temperature parameter (Hoover and Lubkowski 2002; Rocha et al. 2008).

While these methods often lead to sequences with well balanced overall properties, they also suffer from several drawbacks.

Many of the optimization parameters to be taken into account represent rather local sequence properties, spanning a region of a few dozen bases, than global phenomena. This is obvious for short sequence motifs, such as restriction sites, splice site recognition patterns, etc.

Regarding properties like the GC content it is normally much less important to achieve a certain overall GC content than to avoid spikes in GC distribution, i.e. short sequence stretches with either very high or very low GC content.

The former assertion can also be extended to inverse repetitions, where long distance inverse repetitions are considered to be biologically less important than neighbouring ones, which can form stable hairpin loops, and several other features. As Monte Carlo Methods take only a tiny fraction of the whole sequence space into account, in most cases a less than optimal solution with respect to the theoretically ideal combination of codons representing the desired properties will be found in finite time.

Although it is impossible to assess all possible codon combinations representing a given amino acid sequence, it becomes clear from the aforesaid, that it is acceptable for many sequence features to reduce the search space by performing an exhaustive search for the best solution only inside a small "variation window", which is moved along the whole sequence.

Materials and methods

Sequence optimization

All shown or described sequence optimizations were performed with the GeneOptimizer software suite, which was developed in-house by the Geneart Corporation.

Transient expression of Mip1-alpha in 293T cells

MIP1-alpha alleles were cloned via 5′ XhoI—EcoRI-3′ into pcDNA3.1 (Invitrogen). Recombinant plasmids were produced in *E. coli* (XL10 Gold) and purified using the Endo-free Qiagen Maxi Kit (Qiagen) according to the manufacturers instructions. Ten µg of each plasmid were transfected using the Ca-Phosphate method (Graham and van der Eb 1973) into 293T cells. Cell culture supernatants were harvested 48 h post transfection. Amounts of secreted MIP1-alpha were determined using a commercial sandwich ELISA Kit (R&D Systems, mouse CCL3/MIP-1α).

Protein expression in *E. coli*

Standard and methylation site-optimized expression constructs were transformed into *E. coli* BL21(DE3). Two independent colonies were inoculated into 0.5 ml Luria–Bertani (LB) broth containing kanamycin (50 µg/ml), and grown overnight at 30°C with shaking at 160 rpm. Overnight cultures were then diluted in 50 ml of freshly prepared Luria–Bertani broth containing kanamycin (50 µg/ml). Cells were grown to an OD_{600} between 0.5 and 0.7 at 37°C and induced with 1 mM IPTG. After induction, cells were shifted to 30°C and continued to grow for 4 h. Cells were harvested by centrifugation at 4,000*g* for 10 min, resuspended in 5 ml lysis buffer (PBS, 1% Triton X100, 20 µl ProteaseInhibitor), and flash freeze in liquid N_2. Lysis was performed by one freeze/thaw cycle, followed by the addition of 20 µl lysozyme (in a concentration of 0.5 mg/ml), incubation on ice for 10 min and sonication for 30 s.

Quantification of expression

Protein concentration was measured using DC Protein Assay (Bio-Rad) and equal amounts were loaded on 4–20%-SDS–PAGE-gels (Invitrogen) for Western Blot analysis. Western Blot signals were detected using BM Chemiluminescence Western-Blotting-Substrate (POD) (Roche) or SuperSignal West-Femto-Maximum-Sensitivity-Substrate (ThermoScientific) and quantified using GelProAnalyzer-Software6 (INTAS). Corresponding standard and methylation site-optimized constructs were analyzed in triplicates on the same gel using α-Penta-His

antibodies. Quantified results were averaged and the ratio standard versus methylation site-optimized construct was determined. Lysate from *E. coli* cells transformed with the empty expression construct, served as negative controls for analysis.

Results

Presentation of the algorithm

To explain the algorithm, a coding DNA sequence of N codons is considered (Fig. 1). It is assumed that a first part of the sequence, comprising codon positions 1 to $i-1$, has already been optimized by the algorithm. The codon positions i to $i+m-1$ are defined as the so-called variation window. In an iterative step, all possible combinations of synonymous codons for the m codon positions—corresponding to m amino acids—are generated. For each generated combination a test-sequence is built, which consists of a section of the already optimized sequence previous to the variation window and the sequence formed by the respective codon combination. Each test sequence is assessed by a quality function with respect to the given optimization parameters, and the codon with the highest scoring test sequence corresponding to position i in the coding DNA sequence is considered a result codon, and becomes part of the optimized sequence. The length of the added section previous to the variation window will depend on the type of the quality function. Then the variation window is shifted by one codon position and the aforementioned steps are performed. This cycle is repeated until the start position of the variation window has reached position $N-m+1$. At this point, test sequences are again built and assessed, but now all variation window codons of the best test sequence can be taken as result codons simultaneously and added to the already optimized part. The number of varied codons m will normally be in the range of three to ten codons, larger numbers being better, but also leading to a larger calculation time, as the quantity of generated test sequences will grow exponentially. Nevertheless, it can be observed empirically that the quality score of the optimized sequence will only improve marginally when a variation window covering more than about four codons is used.

The total quality function generally takes the form of a linear combination of the weighted scores Score_q from several individual quality functions, each evaluating the test sequence or part of it with respect to a different optimization parameter.

$$\text{TotalScore} = \sum_{q=1}^{\text{Number of criteria}} G_q \cdot \text{Score}_q.$$

The weighting factor G_q allows one to differentiate between more and less important parameters in the optimization for a certain experimental set up. For example, in one application it may be more important to achieve a very high codon adaption index (CAI) of the optimized sequence and some repetitive stretches can be tolerated, while in a different application, repetitions shall be avoided as far as possible at the cost of a worse CAI. In most situations, the weighting factors will be chosen in such a way that the optimized sequence represents a well-balanced compromise between different experimental requirements.

The performance of the algorithm can be improved by the stepwise calculation of the individual quality functions, adding their score to the total score and comparing the latter with the best total score already achieved by a previously assessed test sequence. If the remaining quality functions can only contribute negatively to the total score by definition (e.g. the GC content score) and a higher total

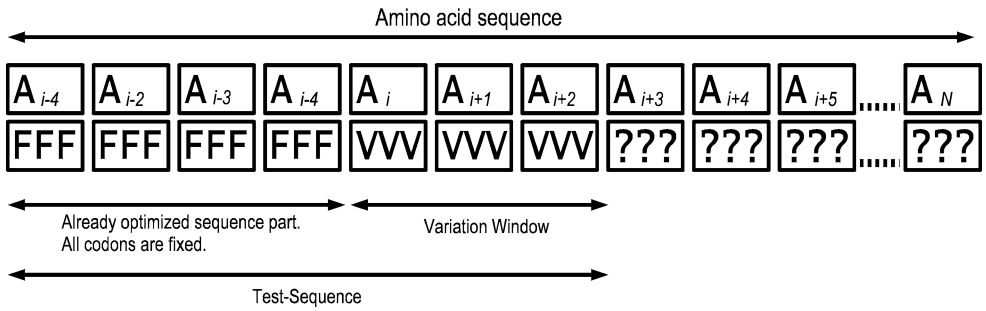

Fig. 1 *Schematic representation* of terms used in the explanation of the algorithm. The *boxes* represent amino acids and their corresponding codon positions. In the shown iteration the first five codon positions have been occupied with resulting codons from previous iterations of the algorithm and together represent the already optimized part of the sequence. The variation window covers the next three codon positions. Each nucleotide sequence resulting from all possible combinations of synonymous codons inside the window is added to the already optimized sequence part to form the test sequence. *A* amino acid, *FFF* fixed codon, *VVV* variable codon

score has already been reached with a different test sequence, the calculation of the remaining quality functions can be omitted.

Suitable methods for the calculating of the individual Score_q values for important optimization criteria will be explained below.

Codon usage

Referred to as degeneracy of the genetic code, nearly all amino acids can be encoded by more than one codon. Nevertheless, not all synonymous codons—i.e. representing the same amino acid—are used with equal frequency, but especially in highly expressed genes certain codons are significantly more frequently used than others. These frequencies—also referred to as "codon usage"—are also correlated with the corresponding tRNA levels in the cell and can strongly differ from organism to organism (Ikemura 1981). This is of great importance for heterologous expression, for example when the gene of interest contains codons that are only rarely used in the expression host. As codon usage is one of the most important factors in procaryotic gene expression, the occurrence of rare codons can significantly reduce expression or even inhibit it (Lithwick and Margalit 2003; Welch et al. 2009; Kane 1995). The codon usage cu_{ij} for a certain codon can be expressed as the absolute frequency of occurrence c_{ij} of the codon j representing the amino acid i in a defined set of genes, divided by the sum of c_{ij} over all n_i synonymous codons:

$$\text{CU}_{ij} = \frac{C_{ij}}{\sum_{j=1}^{n_i} C_{ij}}.$$

The codon usage is readily available for different organisms via the internet, for example from the Kazusa Codon Usage Database (Nakamura et al. 2000).

One important aim of sequence optimization for heterologous expression is therefore to take into account the codon usage of the host organism by reencoding the amino acid sequence with codons having a high c_{ij} in the expression host. Simple single parameter gene optimization is often done by choosing the codon with the highest c_{ij} for each amino acid. As soon as additional optimization parameters shall be considered, the c_{ij} values are no longer suitable for use in the quality function, because they do not allow one to compare the quality of codons encoding different amino acids. This is due to the fact, that the number of synonymous codons is not equal for each amino acid. For example, glutamate is encoded by two codons, while leucine has six and the second frequently used glutamate codon may have a c_{ij} of 0.4, surpassing the "best" leucine codon with a c_{ij} of 0.25.

To take the different number of synonymous codons into account, one can define the "Relative synonymous codon usage" (RSCU_{ij}), which describes the relation between the observed codon usage and the statistically expected fraction for equally frequently used codons (Sharp and Li 1987).

$$\text{RSCU}_{ij} = \frac{\text{CU}_{ij}}{1/n_i}$$

Alternatively, the "Relative Adaptiveness" w_{ij} may be used, which is defined as:

$$w_{ij} = \frac{C_{ij}}{C_{ij\,\max}}$$

where c_{ij} max denotes the codon usage of the most frequently used codon j for a certain amino acid i.

The geometric mean of the w values for L codons is known as the codon adaption index CAI, which indicates how well a coding sequence represents the codon usage of a certain organism. For example, a sequence where only the most frequently used codon for each amino acid is used, would have a CAI value of 1.

$$\text{CAI} = \left(\prod_{k=1}^{L} w_k\right)^{1/L}$$

The CAI can be used directly a score factor in the quality function for assessing a test sequence, calculating the CAI for the codons in the variation window is sufficient.

DNA motifs

It is a very common requirement, that the optimized sequence must not contain certain DNA motifs. One reason can be, that they would impede further processing of the synthetic gene, which is e.g. true for internal restriction sites. But unintentionally introduced sequence patterns can also have an undesired effect in the biological system itself, such as promoter recognition sequences, internal ribosomal entry sites, splice sites and so on. On the other hand, it may be part of the requirements to computationally introduce DNA sequence motifs into the optimized sequence. This can be in combination with a positional constraint allowing one to cut the DNA at a certain position with a restriction enzyme. Another task can be to generate a sequence with as many CpG motifs as possible to enhance its immunogenicity or expression in a mammalian system (Notka et al. 2007).

Many different algorithms and methods have been developed for the prediction of biologically active sequence patterns. In the simplest case, a regular expression search can be performed, which gives a yes or no answer as a result. This is however only possible for highly

conserved motifs, like restriction sites. In this case, the scoring function will examine a 3-prime part of the test sequence, and deliver the number of sequence segments matching the regular expression, multiplied with a motif specific "penalty" or "bonus". To ensure that all occurrences are found, the length of the considered part of the test sequence must match the longest motif.

For the recognition of motifs showing a higher degree of variation, position specific weight matrices can be employed. These 2-dimensional arrays represent the probability of finding a certain nucleotide at each base position of the target motif. The method can be enhanced by introducing a "consensus index" to account for the fact, that e.g. positions with a nearly uniform distribution of the four nucleotides are less significant than positions where predominantly one specific nucleotide is observed (Quandt et al. 1995). The search algorithm scans a DNA sequence and returns a matrix similarity value for each sequence position. High similarity values indicate that the matching sequence part will probably exhibit a high functional potential in the biological system. Again, the quality score for the test sequence can be calculated by the number of matches—having a sufficiently high similarity value—multiplied with their respective similarity value $F_{i,j}$ itself and a motif specific penalty factor g_i:

$$\text{Score}_{\text{Motives}} = \sum_{i=1}^{\text{Number Motives}} \left(g_i \cdot \sum_{j=1}^{\text{Number occurrences}} F_{i,j} \right)$$

The "confidence values" of other methods for motif recognition, like neural networks, can be included into the quality function in a similar fashion (Hebsgaard et al. 1996). As indicated above, the score for the occurrence of desired motifs is counted positive while the occurrence of unwanted patterns is scored negatively in the quality function.

GC content

The GC content is an important characteristic of a DNA sequence. DNA with extremely high or low GC content is more difficult to handle with standard molecular biological techniques such as PCR or sequencing, and also the gene synthesis process itself, which is often based on the correct hybridization of oligonucleotides with a subsequent PCR, may be aggravated (Sanli et al. 2001). Also in the biological system high deviations from an equal GC/AT distribution can lead to genetic instability of the constructs, for example (Lee et al. 2002).

It is nevertheless not sufficient to observe the overall GC content of the sequence, but to avoid spikes of very low or high GC content in the distribution along the sequence.

A suitable quality function for the GC content can be based on the negatively counted absolute difference between the desired GC content and the GC content of a defined end piece of the test sequence:

$$\text{Score}_{\text{GC}} = -\left|\overline{\text{GC}} - \text{GC}_{\text{desired}}\right|$$

In a multiparameter quality function the influence of a certain parameter in relation to the others is determined by the maximum difference in its score, which can be reached by the choice of different codons in the variation window. However, in the above formula the contribution by the nucleotides before the variation window, and the invariant nucleotides within the window, to the GC score will be neglected. Instead, it is desirable that in a case where these nucleotides cause a high deviation from the desired GC content, the GC parameter quality function should highly contribute to the total quality function. This can be accomplished by introducing an exponent into the formula:

$$\text{Score}_{\text{GC}} = -\left|\overline{\text{GC}} - \text{GC}_{\text{desired}}\right|^{p}$$

Repetitions, inverse repetitions

DNA stretches of high similarity to each other ("repetitions") in a gene can lead to genetic instability, as recombination events are fostered (Bzymek and Lovett 2001; Chen et al. 1987). Also gene synthesis methods employing a batch-Polymerase Extension Reaction of overlapping oligonucleotides will be hampered, since the partially similar oligonucleotides will cause false hybridization events to occur (Czar et al. 2009). The same is true for inverted repeats, which can, when they are sufficiently near to each other, lead to quite stable hairpin loops, which in turn can contribute to low energetic mRNA-secondary structures. Especially at the 5′ end, stable mRNA secondary structures can significantly reduce expression by hampering translation initiation (Kudla et al. 2009; Griswold et al. 2003).

The most universal—albeit time consuming—approach for detecting sequence similarities is to perform a local alignment of a terminal part of a test sequence with itself. For performance reasons only the alignment score of the highest scoring alignment is used in the quality function. Since concerning the invariant nucleotides before and inside the variation window the same applies as described above regarding the GC score, the quality function takes the form:

$$\text{Score}_{\text{Repetitions}} = -\text{Alignmentscore}_{\text{max}}{}^{p}$$

In a similar fashion the test sequences can be checked for inverted repeats. However, in this case a terminal part of the test sequence is at first inverted and then the complementary sequence is calculated. The resulting sequence is aligned with the complete test sequence and the alignment score used in the quality function:

$$\text{Score}_{\text{InvRepetitions}} = -\text{Alignmentscore}_{\max}{}^{p}$$

Homologies to reference sequences

A third application of sequence alignment in the quality function is the avoidance of similarities to a given reference sequence. This can be important for the development of DNA vaccines, where recombination events between the vaccine and the wildtype virus must be avoided. Another example is siRNA resistant genes, which are used to restore the gene function after the original gene has been silenced. When the original phenotype can be restored, the change in phenotype can be attributed to the silenced gene with higher confidence than with siRNA silencing alone (Dong-Ho and Rossi 2003). They can be optimized for increased expression and at the same time reduced homology to the wildtype gene. Again, a local alignment between a terminal part of the test sequence and the reference sequence can be used in the quality function.

Exemplary effects of various quality functions

To exemplify the effect of various quality functions, the DNA sequence coding for the green fluorescent protein from *Aequorea victoria* (GenBank: X83960.1) is at first optimized only to the highest possible CAI with respect to *E. coli K12* codon usage (Kazusa). In subsequent optimizations a two parameter quality function is used, which also accounts for a desired GC content of 50% (within a 40 nucleotide window), and the weighting factor for this parameter is stepwise increased. To visualize the properties of the resulting sequences, the used codons are classified by their $w_{ij} \times 100$ value and histogrammed, the GC content is calculated within a window of 40 bases and plotted against the sequence position.

As can be seen in Fig. 2 the sequence optimized solely for a high CAI comprises only codons where $w = 1$. However, the GC content run is rather heterogeneous with values reaching over 70% and below 30%. As soon as the GC content is factored in with a weighting factor of one, the curve begins to smoothen, and a weighting factor of 2 seems to yield the best compromise between high CAI and an even GC distribution around 50%. Increasing the weighting factor to 5 gives a slight additional improvement of the GC distribution at the expense of some rare codons.

These snapshots of consecutive optimization/analysis cycles illustrate a typical approach in determining a suitable set of weighting parameters for the quality functions to achieve a sequence representing the desired characteristics. However, once a suitable set of parameter values has been determined, these values can be used as a good starting point for similar optimizations, e.g. "optimization for expression in *E. coli*" and then often the optimization will yield a satisfying sequence with the first run.

Runtime analysis

In each step of the optimization algorithm, $O(k^m)$ codon combinations are examined, where k represents the maximum number of synonymous codons for an amino acid. On each combination, the user-chosen quality functions are evaluated. Since the algorithm performs $O(N\text{-}m)$ steps, one obtains a running time of:

$$O(k^m \cdot (N - m) \cdot m)$$

if only quality functions are considered that take linear time to evaluate on the sliding window. In the case where quality functions are used that are based on scores obtained by aligning the sliding window with the whole previous sequence, one obtains a running time of:

$$O(k^m \cdot m \cdot N^2)$$

The reason for this is that the dominating running time in each step is given by the running time for computing the alignment. Since the alignment of two sequences of length l_1 and l_2 takes time $O(l_1 \cdot l_2)$, and since the sliding windows of length m are aligned with sequences of lengths m, $m+1,\ldots,N$, the overall running time amounts to:

$$O(k^m \cdot (m \cdot m + \cdots + m \cdot N)) = O(k^m \cdot m \cdot N^2)$$

It can also be seen from the above expressions that an increase in the number of codon positions m within the window has the largest influence on the runtime of the algorithm. However, personal experience from many optimizations shows that an increase beyond a value of 5 for m hardly ever has any influence on the final optimized sequence.

As the tendency in processor development is towards multicore processors rather than a further increase in clock speed, it is noteworthy, that the algorithm offers several opportunities for parallelization. This may, for example, be done by splitting up the calculation of the quality function for a certain position of the variation window to several processor cores, i.e., dealing with an equal number of window position-specific codon combinations on each core. It is important that the threads can exchange information about the best total score obtained for one of the already assessed combinations, so that the calculation of quality functions can still be performed stepwise and cancelled when obvious that a certain combination will no longer reach a better score than the already established best score.

To give an example of the actual time requirements, a typical optimization run was performed on a coding sequence comprising 738 codons. All needed algorithms were implemented in vb.net and executed on a standard personal computer (Windows XP, AMD Athlon 64 X2

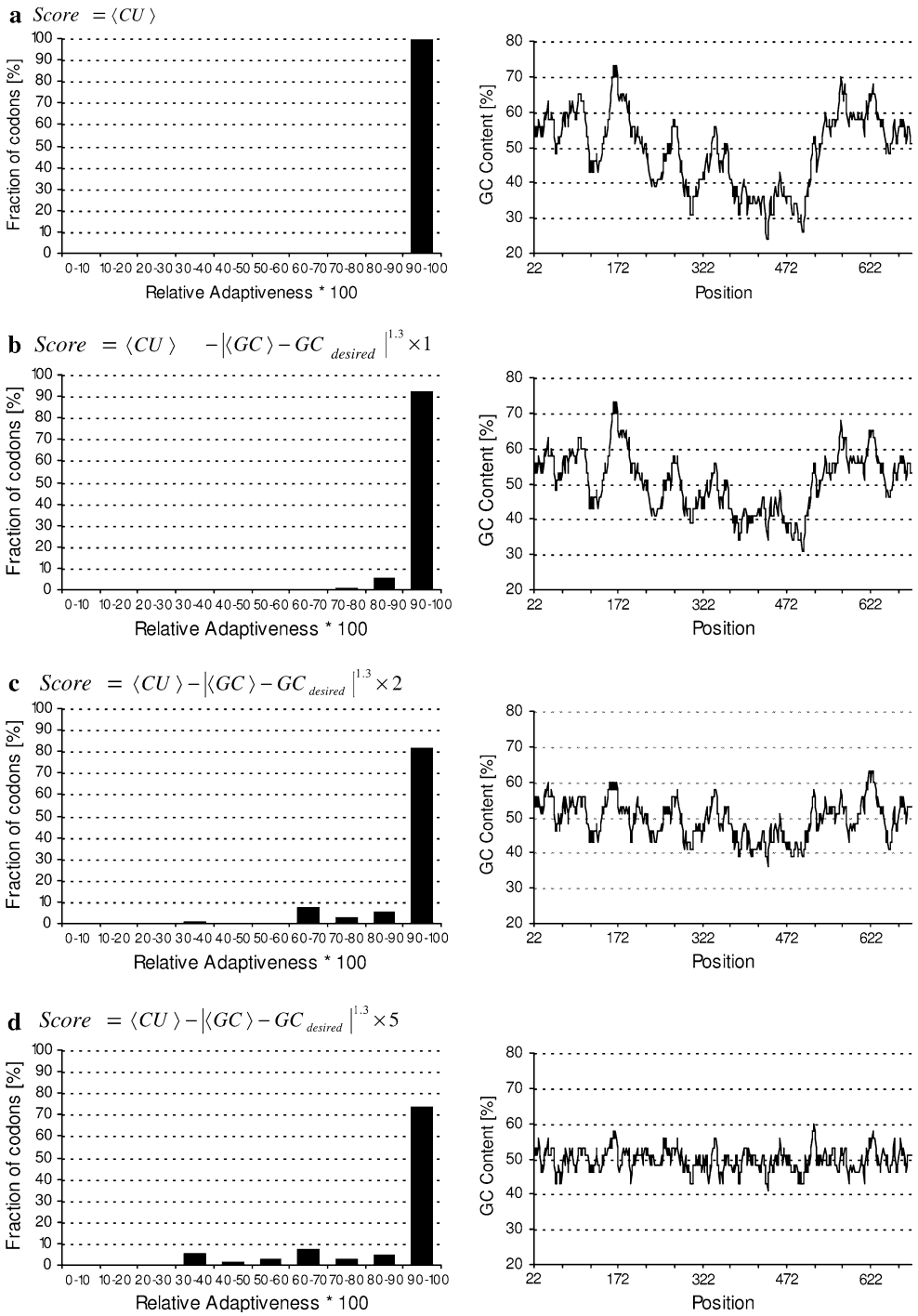

Fig. 2 Codon usage and GC content distribution of the GFP gene sequence when optimized with different quality functions. **a** Only the CAI of the variation window is used in the quality function. **b–d** Additionally the GC content of the last 40 bases of the test sequence is included using an increasing weighting factor

Dual Core Processor 5200+, 2.60 GHz, 3 GB RAM). For the optimization, a *homo sapiens* codon usage table was employed, in which codons with a relative adaptiveness $w < 0.30$ were not taken into consideration. When a quality function comprising the codon quality, GC-content and the check for sequence motifs was used, with $m = 5$, a runtime of 70 s was measured. With inclusion of the check for repetitions, the runtime increased to 96 s.

Application example Mr. Gene

The algorithm has been successfully used for the optimization of several ten thousand genes with the GeneOptimizer suite, a software package used in-house by the Geneart Corporation. To make the algorithm more accessible to the community of researchers interested in gene synthesis and sequence optimization, the Mr. Gene web

application has been developed. Like most modern software, it uses an "assistant guided" workflow to lead the user through the few steps from the original sequence to the optimized gene (Fig. 3). The latter can be directly ordered from the Mr. Gene company or may be used for other purposes. Every optimization is regarded as a separate project, that can be saved together with all associated parameters for later retrieval.

The workflow starts with the input of the original sequence, which can either be provided as DNA or amino acid sequence. The user may also choose between several optimization templates for different common organisms or opt for proceeding without optimization. The templates not only provide for the correct codon usage table, but also control the other optimization parameters, such as which DNA motifs to exclude from the optimized sequence by default, or the weight of the different optimization goals in the quality function. This especially helps non-experts to successfully design and optimize their genes on their own.

Besides the coding sequence, the user can also provide 5-prime and 3-prime untranslated regions and an optional cloning vector.

In the next step additional motifs to be excluded from the optimized sequence can be chosen, either from a repository of common motifs—like as restriction sites—or by manually entering a individual nucleotide pattern. In this step also the organism specific codon usage table may be altered.

The provided data is now used to compute the optimized sequence, which is then presented to the user on the next screen of the assistant. In case some undesired DNA patterns—such as recognition sites or extended repetitive elements—could not be eliminated automatically from the sequence, these sites are listed on the upper part of the screen. In the lower part, the sequence is presented inside a codon editor, which allows each used codon to be altered individually by the user. This is done by simply clicking on the relevant sequence position and choosing a synonymous codon from the appearing drop-down list. The list of problematic sequence parts is also coupled with the codon editor, where the selected sequence part is highlighted. After the sequence has been edited manually, it can be re-checked anytime as to whether problematic sites are still present.

A further screen provides a graphical comparison of the properties of the original sequence to the optimized sequence. Included are a codon usage histogram and two plots showing the distribution of codon usage and GC content along the sequence.

The final sequence can either be directly ordered for synthesis at the Mr. Gene company, saved as project, or exported as a PDF file.

Application examples of the algorithm

It has already been pointed out that a distinct feature of the algorithm is the ability to introduce defined DNA motifs into the optimized sequence, where the number of generated patterns can be controlled by the relative weighting of the optimization parameters (e.g. codon usage versus bonus per introduced motif). This greatly facilitates studying the relationship between the number of certain DNA patterns in a coding sequence and a supposedly associated effect.

As an example we would like to present an experiment, deriving from the area of HIV vaccine research. At first an amino acid derivative of the HIV-1 gp41 protein was designed. Its corresponding DNA sequence was optimized for expression in *E. coli* and the presence of dam/dcm methylation sensitive DNA patterns. Also extensive repetitions and other potential expression-inhibiting motifs where avoided. Different weighting of the optimization parameters yielded four sequence variants, comprising 0, 4, 11 and 20 methylation sites. Expression analysis of the different genes in *E. coli* showed that protein levels increased significantly with the decreasing number of methylation sites (Fig. 4).

A second experiment was designed to illustrate the versatility of the algorithm in implementing different literature-known optimization strategies. Also their effects on the expression of one exemplary protein, the murine macrophage inflammatory protein Mip1-alpha, is demonstrated.

For the first sequence variant, a simple backtranslation was performed by using a solely CAI based quality function in the optimization, i.e. for each amino acid, the synonymous codon with the highest relative adaptiveness value.

The second variant was designed to represent the strategy of "codon harmonization", which means that the codon usage of the optimized sequence should reflect the tabulated codon usage of the host organism as closely as possible (Angov et al. 2008). Therefore, a special quality function was developed, that calculates from each test sequence a "codon usage table" for the occurring represented amino acid types. The accumulated differences between the observed codon frequencies and the tabulated organism-specific frequencies are used in the quality function.

$$\mathrm{Score}_{\mathrm{Harmonization}} = -\left(\sum_{i=1}^{\substack{\text{Occuring amino}\\ \text{acid types}}} \sum_{j=1}^{\substack{\text{Synonymous}\\ \text{codons}}} \left| \mathrm{CU}_{\mathrm{ij\ observed}} - \mathrm{CU}_{\mathrm{ij\ tabulated}} \right| \right)^{p}$$

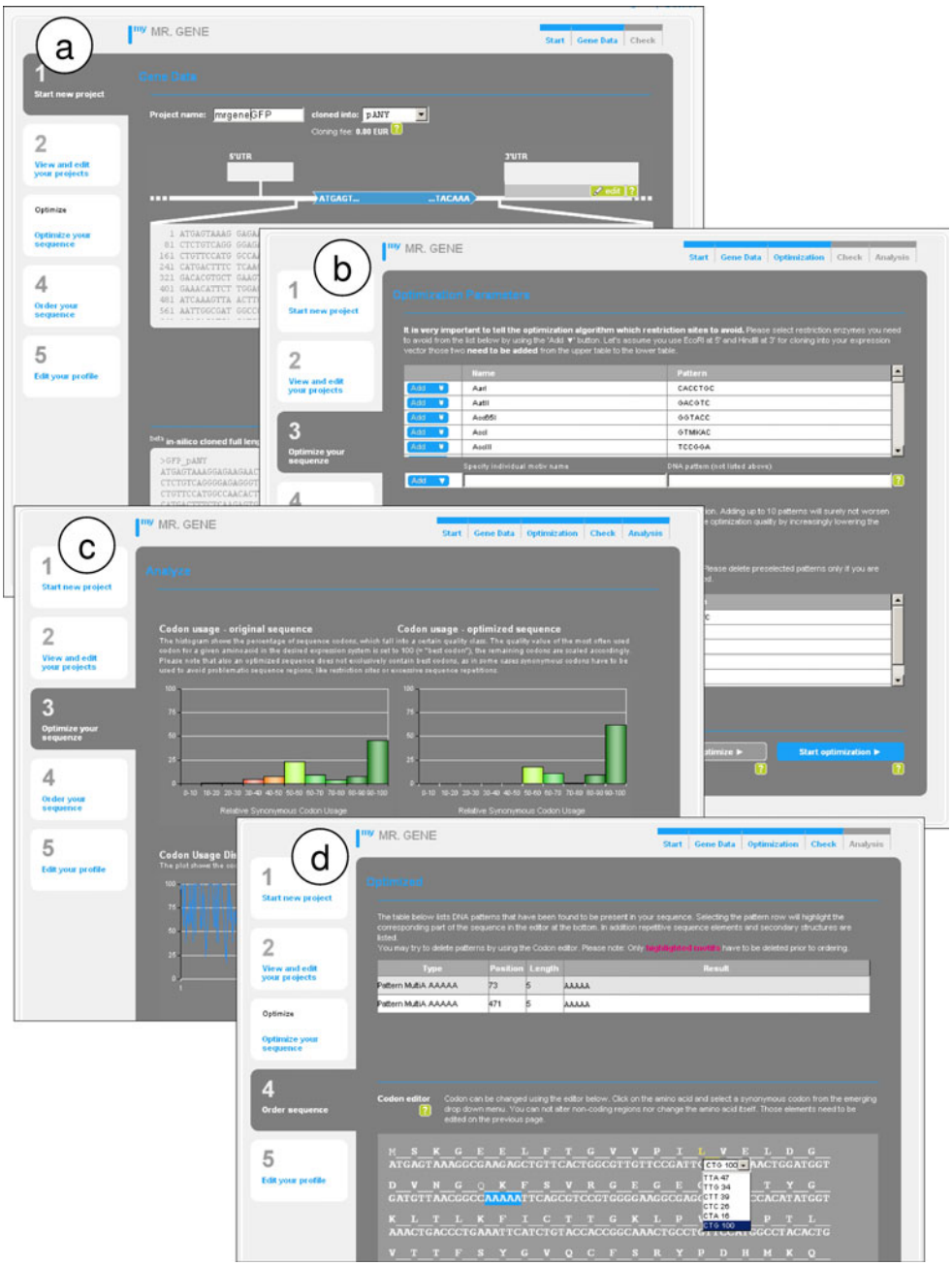

Fig. 3 Screenshots showing different steps in the optimization process of a gene using the Mr. Gene web application. From *top* to *bottom* **a** Entry of the native sequence, including optional UTRs and choice of the expression host. **b** Selection of DNA motifs to be excluded from the optimized sequence.**c** Graphical comparison of the properties of the native vs. optimized sequence. **d** List of undesired sequence patterns and codon editor. The selected motif is highlighted in the sequence, which can be altered manually by choosing alternative codons

The third strategy involved a CAI based quality function with respect to the codon usage, combined with quality scores for (inverse) repetitions, GC content and potentially inhibitory motifs.

The variants including the wildtype sequence were synthesized by Geneart and 293T cells were used to transiently express the genes. The amounts of secreted protein were measured using a commercially available ELISA.

Interestingly, the "harmonization" variant yielded even worse expression levels than the wildtype gene. The "backtranslate" optimization scored second best, while the "combined" approach delivered the best expression levels (Fig. 5).

Discussion

We have presented a deterministic algorithm for the optimization of a coding sequence, which has significant advantages over stochastic methods especially as far as local sequence properties are concerned. This is most obvious with the task of introducing a defined motif into

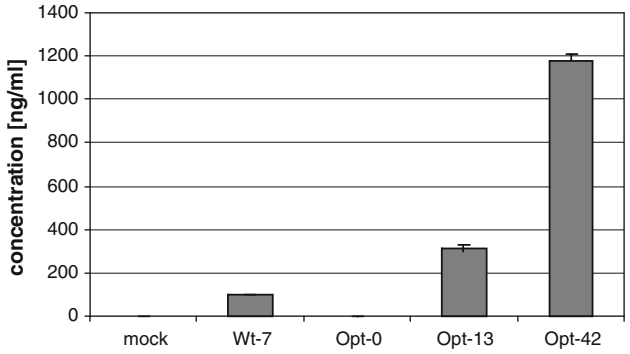

Fig. 4 Expression analyses of a HIV-1 gp41 derivative. *E. coli BL21*(*DE3*) expressing a HIV-1 gp41-derived variant (optimized with different amounts of methylation sites) were tested for their respective protein expression levels as determined by Western blot analyses and subsequent quantification using GelProAnalyzer-Software6. The graph displays the result of 3 independent experiments. Quantified results were averaged and the ratio of standard optimization (set at 100%) versus methylation site optimized construct was determined

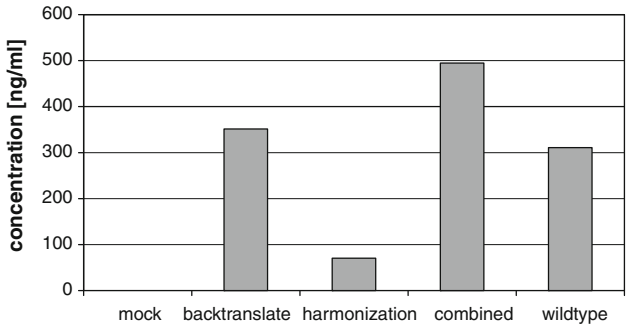

Fig. 5 Expression analysis of MIP1-alpha. 293T cells expressing different MIP1-alpha DNA sequence variants, optimized based on different strategies were tested for their respective protein expression levels as determined by ELISA analyses. *Backtranslate*: for each amino acid the synonymous codon with the highest relative adaptiveness value was used. *Harmonization*: the codon frequencies in the optimized sequence reflect the tabulated codon usage of homo sapiens. *Combined:* a multiparamater optimization was performed, accounting for a high CAI and avoidance of potentially expression limiting motifs, as well as spikes in GC content, and extensive repetitions

the sequence, which may only be possible with one specific combination of codons within the variation window. On the other hand, it is possible to find a codon combination that eliminates a weight matrix-defined motif by changing the nucleotides most important for the biological activity, often without compromising other important sequence properties.

It might be argued, that the directionality of the algorithm (i.e. optimizing the sequence from the 5-prime start to the 3-prime end) is a disadvantage compared to stochastic methods, which normally take a more global approach in assessing the sequence within each iteration of the optimization process. For example, consider an amino acid sequence that contains two identical sequence parts, one at the beginning and one at the end of the sequence. When the aim is to achieve a good CAI and at the same time avoid extensive repetitions in the optimized DNA sequence, the presented algorithm can only eliminate the repetition by introducing worse synonymous codons (with respect to their codon usage) in the second sequence part, while a stochastic algorithm is able to distribute the worse codons evenly between the two parts. However, it has been shown that the occurrence of rare codons can be much better tolerated at the 3-prime part of the sequence than at the beginning, actually turning the supposed disadvantage into an advantage (Goldman et al. 1995; Vervoort et al. 2000).

While we have shown how a number of important sequence properties can be accounted for in the quality function, it is obvious that further optimization parameters, for example the consideration of codon pairings (Gutman and Hatfield 1989), can easily be included in the calculation of the total quality score.

Acknowledgments The authors would like to thank Thomas Hofmeister, Andreas Wolf and Wolfgang Strenzl for their efforts in implementing the algorithm in different software applications. DR thanks Jörg Enderlein for his support in the course of the development of the GeneOptimizer software application. This work was supported by the Bundesministerium für Bildung und Forschung (BMBF) through grants 0313850 ("OLIGO" to R.W.) and 0313687 ("BioChancePLUS" to R.W.).

Conflict of interest The authors declare competing financial interests: Geneart AG performs gene design optimization as a free service with the genes that it sells.

References

Angov E, Hillier CJ, Kincaid RL, Lyon JA (2008) Heterologous protein expression is enhanced by harmonizing the codon usage frequencies of the target gene with those of the expression host. PLoS ONE 3(5):e2189. doi:10.1371/journal.phone.0002189

Bzymek M, Lovett ST (2001) Instability of repetitive DNA sequences: the role of replication in multiple mechanisms. PNAS 98(15):8319–8325. doi:10.1073/pnas111008398

Chen CW, Tsai JFY, Chuang S (1987) Intraplasmid recombination in Streptomyces lividans 66. Mol Gen Genet 209:154–158

Czar MJ, Anderson JC, Bader JS, Peccoud J (2009) Gene synthesis demystified. Trends Biotechnol 27(2):63–72. doi:10.1016/j.tibtech.2008.10.007

Dong-Ho K, Rossi JJ (2003) Coupling of RNAi-mediated target downregulation with gene replacement. Antisense Nucleic Acid Drug Dev 13(3):151–155. doi:10.1089/108729003768247619

Goldman E, Rosenberg AH, Zubay G, Studier FW (1995) Consecutive low-usage leucine codons block translation only when near the 5′ end of a message in *Escherichia coli*. J Mol Biol 245(5):467–473

Graham FL, van der Eb AJ (1973) A new technique for the assay of infectivity of human adenovirus 5 DNA. Virology 52(2): 456–467

Griswold KE, Mahmood NA, Iverson BL, Georgiou G (2003) Effects of codon usage versus putative 5′-mRNA structure on the expression of fusarium solani cutinase in the *Escherichia coli* cytoplasm. Protein Expr Purif 27:134–142

Gutman GA, Hatfield GW (1989) Nonrandom utilization of codon pairs in *Escherichia coli*. Proc Natl Acad Sci USA 86: 3699–3703. doi:10.1073/pnas.86.10.3699

Hebsgaard SM, Korning PG, Tolstrup N, Engelbrecht J, Rouzé P, Brunak S (1996) Splice site prediction in Arabidopsis thaliana pre-mRNA by combining local and global sequence information. Nucleic Acids Res 24(17):3439–3452

Hoover DM, Lubkowski J (2002) DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis. Nucleic Acids Res 30(10):e43

Ikemura T (1981) Correlation between the abundance of *Escherichia coli* transfer RNAs and the occurrence of the respective codons in its protein genes: a proposal for a synonymous codon choice that is optimal for the *E. coli* translational system. J Mol Biol 151:389–409. doi:10.1016/0022-2836(81)90003-6

Kane JF (1995) Effects of rare codon clusters on high-level expression of heterologous proteins in *Escherichia coli*. Curr Opin Biotechnol 6:494–500. doi:10.1016/0958-1669(95)80082-4

Kudla G, Murray AW, Tollervey D, Plotkin JB (2009) Coding-sequence determinants of gene expression in *Escherichia coli*. Science 324:255–258

Lee SG, Kim DY, Hyun BH, Bae YS (2002) Novel design architecture for genetic stability of recombinant Poliovirus: the manipulation of G/C contents and their distribution patterns increases the genetic stability of inserts in a Poliovirus-based RPS-vax vector system. J Virol 76(4):1649–1662

Lithwick G, Margalit H (2003) Hierarchy of sequence-dependent features associated with prokaryotic translation. Genome Res 13:2665–2673. doi:10.1101/gr.1485203

Nakamura Y, Gojobori T, Ikemura T (2000) Codon usage tabulated from the international DNA sequence databases: status for the year 2000. Nucleic Acids Res 28:292

Notka F, Leikam D, Bauer A, Raab D, Graf M, Wagner R (2007) Computer aided multi-parameter gene design: impact of synthetic DNAs on protein expression enhancement. BMC Syst Biol 1(Suppl 1):60. doi:10.1186/1752-0509-1-S1-P60

Quandt K, Frech K, Karas H, Wingender E, Werner T (1995) MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. Nucleic Acids Res 23(23):4878–4884

Rocha M, Maia Paulo, Mendes R, Pinto JP, Ferreira EC, Nielsen J, Patil KR, Rocha I (2008) Natural computation meta-heuristics for the in silico optimization of microbial strains. BMC Bioinformatics 9:499. doi:10.1186/1471-2105/9/499

Sanli G, Blaber SI, Blaber M (2001) Reduction of wobble-position GC bases in Corynebacteria genes and enhancement of PCR and heterologous expression. J Mol Microbiol Biotechnol 3(1): 123–126

Sharp PM, Li WH (1987) The codon adaption index–a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Res 15(3):1281–1295

Vervoort EB, van Ravestein A, van Peij NME, Heikoop JC, van Haastert PJM, Verheijden GF, Linskens MHK (2000) Optimizing heterologous expression in Dictyostelium: importance of 5′ codon adaption. Nucleic Acids Res 28(10):2069–2074

Villalobos A, Ness JE, Gustafsson C, Minshull J, Govindarajan S (2006) Gene Designer: a synthetic biology tool for constructing artificial DNA segments. BMC Bioinformatics 7:285. doi: 10.1186/1471-2105-7-285

Welch M, Govindarajan S, Ness JE, Villalobos A, Gurney A, Minshull J, Gustafsson C (2009) Design parameters to control synthetic gene expression in *Escherichia coli*. PLoS ONE 4(9):7002. doi:10.1371/journal.pone.0007002