
Bayesian optimization for synthetic gene design

Javier González^{2,3}, Joseph Longworth², David C. James² and Neil D. Lawrence^{1,3}

¹Department of Computer Science

²Department of Chemical and Biological Engineering

³Sheffield institute for translational neuroscience. University of Sheffield

{j.h.gonzalez, j.longworth, d.c.james, n.lawrence}@sheffield.ac.uk

Abstract

We address the problem of synthetic gene design using Bayesian optimization. The main issue when designing a gene is that the design space is defined in terms of long strings of characters of different lengths, which renders the optimization intractable. We propose a three-step approach to deal with this issue. First, we use a Gaussian process model to emulate the behavior of the cell. As inputs of the model, we use a set of biologically meaningful gene features, which allows us to define optimal *gene designs rules*. Based on the model outputs we define a multi-task acquisition function to optimize simultaneously several aspects of interest. Finally, we define an *evaluation function*, which allow us to rank sets of candidate gene sequences that are coherent with the optimal design strategy. We illustrate the performance of this approach in a real gene design experiment with mammalian cells.

1 Introduction

Synthetic biology concerns with the design and construction of new biological elements of living systems and the re-design of existing ones for useful purposes [4]. In this context, there is a current interest in the development of new methods to engineer living cells in order to produce compounds of interest [11]. A modern approach to this problem is the use of synthetic genes, which once ‘inserted’ in the cells can modify their natural behavior activating the production of proteins useful for further pharmaceutical purposes.

We present the first approach for gene design based on Bayesian optimization (BO) principles. The BO framework [14, 3, 10, 6] allows us to explore the gene design space in order to provide rules to build genes with interesting properties, such as genes that are able to produce proteins of interest, or genes able to act on the cell lifespan. We use a Gaussian process [12] to emulate the complex behavior of the cell across the different gene designs. An acquisition function is optimized to deal with exploration-exploitation trade-off. To provide, not only rules for gene design, but current gene sequences candidates, we introduce the concept of *evaluation function*. The goal of this functions is to avoid the bottleneck of optimizing over the sequences by providing rules to rank biologically feasible genes coherent with the obtained gene design rules.

Although in this work we focus on the optimization of the translational efficiency of the cells, this framework can be generalized to multiple synthetic biology design problems, such as the optimization of media design, or the optimization of multiple gene knock-out strategies.

2 Rewriting the genetic code

Broadly speaking, in molecular biology it is assumed that a gene contains the information to encode a mRNA molecule. The production of such molecules is called *transcription* and takes place in

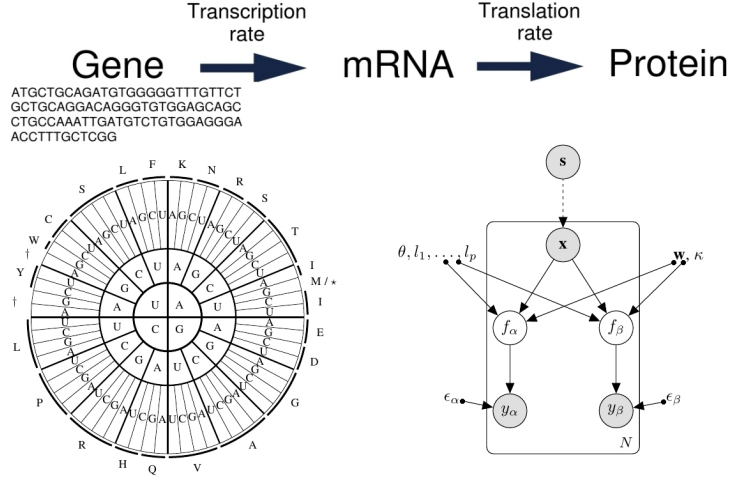


Figure 1: *Top*: Central dogma of molecular biology. Each gene contains the information to encode an mRNA molecule which is used by the cell to produce proteins. Both, mRNA molecules and proteins are produced at certain rate. The goal is to design genes sequences able to increase the rates while encoding the same protein. *Bottom left*: Graph of codons redundancy. Letters inside the circle represent DNA basis and letters outside represent amino acids. The arcs outside the circle cover the paths of redundant codons. † and * represent special codons. *Bottom right*: Graphical model of the multi-output Gaussian process used to emulate the cell behavior in this work.

the cell nucleus at certain rate y_α . Later on, the mRNA molecules are used to produce proteins at a different rate y_β , in what is called the *translation* phase. A one-to-one correspondence between genes and proteins is assumed. Each gene, itself, is also made up of a sequence \mathbf{s} of several hundreds of bases (A, T, G, C), triplets of which, form the so-called codons. We can interpret the codons like the ‘words’ in which the genetic code is written. The 64 possible codons encode 20 amino acids, which are the fundamental elements that the cell uses to produce proteins. This means that the genetic code is redundant: the same aminoacid can be encoded by different codons and therefore there exist multiple ways of encoding the same protein. See Figure 1 (top and bottom left) for an illustration of this process. A fundamental of gene design is that redundant codons choices do not affect the type of protein that is being encoded but they may affect the rates y_α, y_β , and therefore the efficiency at which it is produced.

Consider a p -dimensional representation $\mathbf{x} \in \mathbb{R}^p$ of a gene sequence \mathbf{s} . Such a representation will typically be the frequency of the different codons but it may also include other variables like the length of the sequence, or the times a certain pattern is repeated across the gene. Denote by $f_\alpha, f_\beta : \mathcal{X} \rightarrow \mathbb{R} \times \mathbb{R}$ the functions representing the expected transcription and translation rates given a sequence with features $\mathbf{x} \in \mathcal{X}$. We want to solve the global optimization problem of finding the sequence that maximizes both rates. However, this requires optimization across the all possible sequences. This is infeasible due to the high dimensionality. Instead, we aim to solve the surrogate multi-objective problem of finding $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}} (f_\alpha(\mathbf{x}), f_\beta(\mathbf{x}))$, to later connect \mathbf{x}^* with a particular gene design.

3 Method used

3.1 Multi-output Gaussian processes as cell-behavior surrogate model

Let $\{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ be a set of gene sequences. Consider a p -dimensional feature representation of the sequences given by $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ where $\mathbf{x}_i \in \mathbb{R}^p$. Let $\mathbf{y}_\alpha, \mathbf{y}_\beta \in \mathbb{R}^N$ be the observed transcription and translation rates. Our first goal is to learn from a model of the combined data $\mathcal{D} = \{\mathcal{D}_\alpha, \mathcal{D}_\beta\}$, where $\mathcal{D}_\alpha = \{\mathbf{x}_i, y_{\alpha,i}\}_{i=1}^N$ and $\mathcal{D}_\beta = \{\mathbf{x}_i, y_{\beta,i}\}_{i=1}^N$, how to predict the value of the output functions $f_\alpha(\mathbf{x})$ and $f_\beta(\mathbf{x})$ at any $\mathbf{x} \in \mathbb{R}^p$. For simplicity we assume here that both rates are available for all the sequences, but this assumption can be easily relaxed.

Algorithm 1 Bayesian optimization for gene design

Extract features $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ from the available gene sequences $\{\mathbf{s}_1, \dots, \mathbf{s}_N\}$.
Take $\mathcal{D}_1 = \{\mathcal{D}_\alpha, \mathcal{D}_\beta\}$, where $\mathcal{D}_\alpha = \{\mathbf{x}_i, y_{\alpha,i}\}_{i=1}^N$ and $\mathcal{D}_\beta = \{\mathbf{x}_i, y_{\beta,i}\}_{i=1}^N$.
for $t = 1, 2, \dots$ **do**
 Fit a multi-output GP model using \mathcal{D}_t .
 Obtain design rules by taking $\mathbf{x}_{t+1} = \arg \max_{\mathbf{x} \in \mathcal{X}} \text{acq}(\mathbf{x}|\mathcal{D}_t)$.
 Generate a set of candidate gene sequences \mathcal{S} .
 Rank the sequences in \mathcal{S} and select $\mathbf{s}_{t+1} = \arg \min_{\mathbf{s} \in \mathcal{S}} \text{eval}(\mathbf{s}|\mathbf{x}_{t+1})$.
 Run experiment using \mathbf{s}_{t+1} and extract features \mathbf{x}_{t+1} from the sequence \mathbf{s}_{t+1} .
 Augment the data $\mathcal{D}_{t+1} = \{\mathcal{D}_t, (\mathbf{x}_{t+1}, (y_{\alpha,t+1}, y_{\beta,t+1}))\}$.
end for
Returns: Optimal gene design \mathbf{s}^* .

A Gaussian process (GP) is a stochastic process with the property that each linear finite-dimensional restriction is multivariate Gaussian [12]. GPs are typically used as prior distribution over functions. In the simple output case, the random variables are associated to a single process f evaluated at different \mathbf{x} but, this can be easily generalized to multiple outputs, where the random variables are associated to different processes $\{f_l\}_{l=1}^d$. In our case we have $d = 2$, which correspond to the two rates of interest. We work therefore with the vector-value function $\mathbf{f} := (f_\alpha, f_\beta)$, which is assumed to follow a GP $\mathbf{f} \approx \mathcal{GP}(\mathbf{m}, \mathbf{K})$ where \mathbf{m} is a 2-dimensional vector whose components are the mean functions m_α, m_β of each output and \mathbf{K} is a positive matrix valued function that acts directly on input example and tasks indices. The entries $(\mathbf{K}(\mathbf{x}, \mathbf{x}'))_{l,l'}$ in $\mathbf{K}(\mathbf{x}, \mathbf{x}')$ represent the covariance between $f_\alpha(\mathbf{x})$ and $f_\beta(\mathbf{x}')$. Under a Gaussian likelihood assumption, the predictive distribution for a new vector \mathbf{x}_* is taken to be Gaussian such that $p(\mathbf{f}(\mathbf{x}_*)|\mathcal{D}, \mathbf{f}, \mathbf{x}_*, \phi) = \mathcal{N}(\mathbf{f}_*(\mathbf{x}_*), \mathbf{K}_*(\mathbf{x}_*, \mathbf{x}_*))$ where $\mathbf{f}_*(\mathbf{x}_*)$ and $\mathbf{K}_*(\mathbf{x}_*, \mathbf{x}_*)$ are close expressions that depend on the set of input \mathbf{X} and the kernel \mathbf{K} . See [12, 1] for details. ϕ represents all the parameters of the kernel, which can be built following various strategies. In this work we use a combination of the linear and the intrinsic coregionalization models [16, 1]. We take $\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B}_1 \otimes K_{lin}(\mathbf{X}, \mathbf{X}) + \mathbf{B}_2 \otimes K_{se}(\mathbf{X}, \mathbf{X})$, where K_{lin} is a linear kernel used to account for the different levels of the rates and K_{se} a square exponential kernel with a different lengthscale per dimension. $\mathbf{B}_{lin}, \mathbf{B}_{se}$ are the coregionalization matrices, which are parametrized as $\mathbf{B}_{lin} = \mathbf{w}_{lin} \mathbf{w}_{lin}^T + \kappa_{lin} \mathbf{I}_2$ and $\mathbf{B}_{se} = \mathbf{w}_{se} \mathbf{w}_{se}^T + \kappa_{se} \mathbf{I}_2$ for $\mathbf{w}_{lin}, \mathbf{w}_{se}, \kappa_{lin}, \kappa_{se} \in \mathbb{R}^2$ and \mathbf{I}_2 is the identity matrix of dimension 2. \otimes represents the Hadamard product. See Figure 1 (bottom right) for a graphical description of the model.

3.2 Acquisition and evaluation functions

In multi-task optimization problems a typical issue is to deal with potential conflicting objectives, or tasks that cannot be optimized simultaneously; in our case this means that both rates cannot be optimized simultaneously using the same sequence. Following previous work in multi-task Bayesian optimization [15], here we focus on an acquisition function that maximizes the average of the tasks. The predictive mean and variance of the average objective are $\bar{m}(\mathbf{x}) = \frac{1}{2} \sum_{l=\alpha,\beta} \mathbf{f}_*(\mathbf{x})$, and $\bar{\sigma}^2(\mathbf{x}) = \frac{1}{2^2} \sum_{l=\alpha,\beta} \sum_{l'=\alpha,\beta} (\mathbf{K}_*(\mathbf{x}, \mathbf{x}))_{l,l'}$. Both $\bar{m}(\mathbf{x})$ and $\bar{\sigma}^2(\mathbf{x})$ can be used in a standard way using any acquisition function $\text{acq}(\mathbf{x})$, such as the expected improvement (EI) [14].

Consider the optimal gene design given by $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}} \text{acq}(\mathbf{x})$. Assume that we are interested in the production of a certain protein whose sequence is \mathbf{s}_k . To improve the sequence \mathbf{s}_k according to the optimal design rules \mathbf{x}^* without changing the nature of the protein, we can interchange redundant codons, that is, codons encoding the same aminoacid. See Figure 1 (bottom right). Given a set of sequences satisfying this criteria, we introduce an evaluation function to rank them in terms of their coherence with the optimal design. In particular we choose $\text{eval}(\mathbf{s}|\mathbf{x}^*) = \sum_{j=1}^p w_j |\mathbf{x}_j - \mathbf{x}_j^*|$ where \mathbf{s} is a ‘coherent’ sequence, the \mathbf{x}_j are the features of \mathbf{s} , \mathbf{x}_j^* are the features of the optimal design and w_j are weights that we choose to be the inverse lengthscales of the K_{se} . See Algorithm 1.

4 Gene optimization in mammalian cells

In this experiment we use Bayesian Optimization to design genes able to optimize the transcription and translation rates of mammalian cells. We use the dataset in [13] in which the rates of 3810

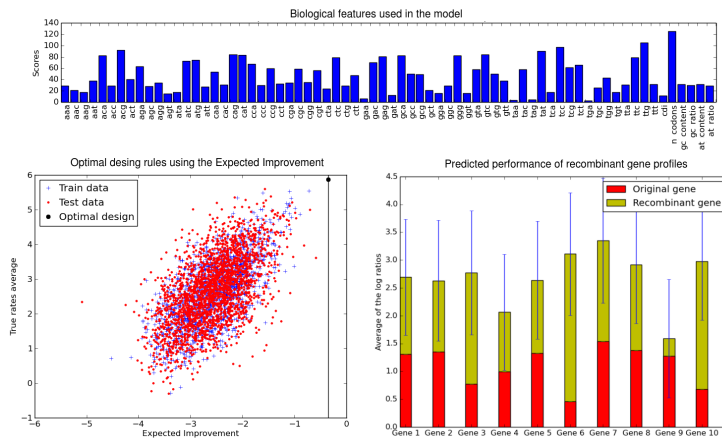


Figure 2: *Top*: Inverse lengthscales of the ARD component of the model. *Bottom left*: Optimal design rules selected using EI in the context of the true performance of 3810 genes in mammalian cells. *Bottom right*: comparison of the true performance of 10 genes with the predicted performance of recombinant genes selected among 1,000 random generated sequences. 95% confident intervals for the predictions are shown in blue.

genes cells are available. The associated sequences were extracted from <http://www.ensembl.org> using the first ENSEMBL identifier of the database. As features of the model, we used the frequency of appearance the 64 codons, together with the length of the gene, the GC-content, the AT-content, the GC-ratio and the AT-ratio. We randomly sampled 1500 genes that we used to train the model described in Section 3. We fit the hyper-parameters of the model by the standard method of maximizing training-set marginal likelihood, using L-BFGS [9] for 1,000 iterations and selecting the best of ten random restarts. We select the optimal gene design by means of the expected improvement, which we optimize across all the available genes (both train and test sets) in order to have a way of evaluating the coherence of the result with real experimental data. In Figure 2 (bottom left) we show the scatter plot of the EI evaluated in all gene features vs. the true average of the ratios. The best possible design is selected by the EI criteria. Next, we select 10 difficult-to-express genes by selecting ten random genes among those whose average log ratio is smaller than 1.5. By taking their sequence as a reference we generated 1,000 random sequences (for each gene) able to encode the same protein. All across the sequences, we replace each codon with a redundant one, which is sampled uniformly from the set of codons encoding the same aminoacid. Using the evaluation function in Section (3.2) we ranked the sequences and selected the top rated. in Figure 2 (bottom right) we show the true performance of the sequence (experimental value) versus the predicted value of best recombinant sequences. In the ten cases the recombinant sequence outperforms the original one.

5 Conclusions and challenges

We have shown that Bayesian optimization principles can be successfully used in the design of synthetic genes. One of the most important aspects in this process is to have a good surrogate model for the cell behavior able to lead to appropriate acquisition functions. Considering future models, the fact that the cell is a extremely complex system will be a key aspect to take into account. To optimize certain features of the cell, massive amounts of data will be required, which will require the used of sparse Gaussian processes. Regarding the optimization aspects of the problem, in this work we have worked with a set of features extracted from the gene sequences, which we have used to obtain gene design rules rather than optimal sequences. The use of more features will potentially lead to better and more specific gene design. This will require, however, the development of scalable Bayesian optimization methods able to work well in high dimensions in the line of some recent works [17, 5, 2, 7]. An alternative approach is to focus directly on the optimization on the sequences rather than on extracted features by omitting any previous biological knowledge. This seems feasible from the modeling point of view by means of the use of string or related kernels [8] but the optimization of the acquisition functions derived from this models remains challenging.

Acknowledgments: The authors would like to thank BRIC-BBSRC for the support of this project (No BB/K011197/1).

References

- [1] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266, March 2012.
- [2] James Bergstra, Rémy Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *NIPS'2011*, 2011.
- [3] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010.
- [4] Paul S. Freemont, Richard I. Kitney, Geoff Baldwin, Travis Bayer, Robert Dickinson, Tom Ellis, Karen Polizzi, Guy-Bart Stan, and Richard I. Kitney. *Synthetic Biology - A Primer*. World Scientific Publishing, 1 edition, July.
- [5] Roman Garnett, Michael A Osborne, and Philipp Hennig. Active learning of linear embeddings for gaussian processes. In *Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, 2014.
- [6] Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13, 2012.
- [7] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, LION'05, pages 507–523, Berlin, Heidelberg, 2011. Springer-Verlag.
- [8] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, March 2002.
- [9] Jorge Nocedal. Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation*, 35(151):773–782, 1980.
- [10] M. Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, PhD thesis, University of Oxford, 2010.
- [11] Michael Pedersen and Andrew Phillips. Towards programming languages for genetic engineering of living cells. *Journal of the Royal Society Interface*, April 2009.
- [12] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [13] Björn Schwanhäusser, Dorothea Busse, Na Li, Gunnar Dittmar, Johannes Schuchhardt, Jana Wolf, Wei Chen, and Matthias Selbach. Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–342, May 2011.
- [14] Jasper Snoek, Hugo Larochelle, and Ryan Prescott Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, 12/2012 2012.
- [15] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2004–2012. Curran Associates, Inc., 2013.
- [16] Yee Whye Teh and Matthias Seeger. Semiparametric latent factor models. *Workshop on Artificial Intelligence and Statistics 10*, 2005.
- [17] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando de Freitas. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conferences on Artificial Intelligence (IJCAI) - Distinguished Paper Award*, 2013.