Performance optimization of an algorithm for DNA rewriting to achieve transgene packaging ability
Master Thesis Exposé

Jessica Kranz

February 3, 2022

1 Introduction

The therapeutic use of genetically engineered recombinases aims at the excision of a provirus from the host genome. This, however, requires the tissue-specific delivery of the specific transgene and its nuclear import [Nabel, 1999]. One approach to deliver a gene is the use of Adeno Associated Viral (AAV) vectors.

Packaging of the therapeutic transgene into the AAV capsid requires the presence of secondary DNA structures [Yan et al., 2005], known as inverted terminal repeats (ITRs), which flank the transgene sequence and function as packaging signals. However, it has been shown that secondary structures inside the transgene sequence similar to the ITRs can interfere with the packaging process [Cataldi and McCarty, 2013] and lead to aberrant encapsidation of truncated (and therefore non-functional) vector genomes.

ITR-like structures are also called hairpins or stable secondary structures due to their shape in 2D folding prediction. We refer to them as *secondary structures* in the following. A secondary structure is formed by a substring of a DNA sequence, composed of the known bases $\Sigma = \{A, C, G, T\}$. For this substring, it must hold that a reverse-complementary substring exists whose elements pair with those of the first subsequence. A *base pair* consists of two bases linked to each other by hydrogen bonds. These bonds, known as Watson-Crick pairings [Watson and Crick, 1953], form between adenine and thymine and between guanine and cytosine. We call the two partial sequences of pairing bases *stem* and the sequence of non-pairing bases between the original and the reverse complementary sequence *loop*. A base sequence is classified as a secondary structure if the stem length is at least the stem length specified by the user and the loop length remains below the loop length specified. Thus, the choice of the stem length and loop length parameters determines the number of structures in a sequence that are considered as secondary structures. We call a DNA sequence optimal (for our purpose) when it is free of secondary structures within the specified parameter limits.

Transgene sequences designed in a laboratory for a specific purpose often contain unwanted and potentially interfering secondary structures. One way to solve the issue is to consider one codon sequence of the transgene and construct an alternative codon sequence by exchanging codon triplets. However, it must be ensured that the resulting amino acid sequence remains unchanged to maintain the functionality of the transgene. In a preliminary work [Kranz, 2021], an algorithm was developed that first examines an initial sequence for the presence of secondary structures. Then, one randomly selected triplet involved in the stem of a secondary structure is replaced one-by-one with the aim to reduce the number of secondary structures. However, since replacing codons can also introduce new secondary structures, this is not guaranteed. If no secondary structure-free sequence is found within a given number of substitutions, the algorithm terminates.

While the resolution of a single, short secondary structure using this method was feasible, the correction of long open reading frames (ORFs) remained challenging.

2 Aim

As a prerequisite for the formal definition of the optimization objective we pursue with this work, we define the following terms.

Given an alphabet Σ, a string B is an ordered sequence of elements from Σ. For DNA strings, the order is defined by the order of bases within the amino acid sequence, read in 5' - 3' direction.

$$B = (b_0, ..., b_{l_{b-1}})$$

When we use the term base pair, we are referring to a relation that assigns each base b to its complementary base.

$$c(b) = \begin{cases} A & \text{if } b = T \\ T & \text{if } b = A \\ C & \text{if } b = G \\ G & \text{if } b = C \end{cases}$$

We define $c(b)$ to be applicable element-by-element to B, such that the complementary sequence c(B) of length l_B is the reverse ordered sequence of pairing bases.

$$B = (b_0, b_1, ...b_{l_{B-1}})$$
$$c(B) = (c(b_{l_{B-1}}),\ c(b_{l_B-2}),\ ...\ c(b_0))$$

By B' we refer to substrings of B of minimal length l_{min}. The total order of B is preserved in every substring B'. We say that B contains s secondary structure if there exists at least one B' in B for which also its complement c(B') exists in B, such B' and c(B') do not overlap in B. Any such pair is called a secondary structure. Intuitively, all secondary structures contained in B can be found by splitting B into sliding windows of length l_{min}, for each determining whether a complementary sequence exists.

$$S(B) = \{B' \ \subseteq \ B \mid S(B')\ (B' \wedge c(B'))\}$$

Let S(B) be the set of secondary structures in B. The aim of this thesis is to design and evaluate an efficient algorithm which, given a sequence B, finds a rewriting r(B) of B such $|S(r(B))|$ is small. We say a rewriting r is optimal for B if there exists no other rewriting r' such that $S(r'(B)) < S(r(B))$.

3 Approach

The solution space of the described problem is large, making an exhaustive enumeration off all possible rewritings infeasible. We propose to search good solutions with the help of metaheuristic methods. For the evaluation of a solution candidate, a cost function has already been established [Kranz, 2021]. Further, possible operations for generating solution candidates were defined. However, the present algorithm chooses operations randomly rather than in a goal-directed way. Section 3.1 gives an overview of possible starting points for generating promising solution candidates. Furthermore, a selection of metaheuristic approaches will be examined for suitability to solve the problem and the most appropriate approach will be selected. A subset of eligible metaheuristics is summarized in section 3.2.

3.1 Operations to generate solution candidates

A non-optimized solution contains at least one secondary structure to be resolved. With a minimal stem length of six, at least four codon triplets are involved in the formation of a secondary structure, each of which can be replaced by different triplets. This often results in a large number of possible rewritings to generate a new candidate solution.

A single triplet rewriting has an individual effect on the secondary structure under consideration and possibly on adjacent secondary structures. Thus, a substitution of a triplet may have a weak effect on resolving the secondary structure due to the location of the exchanged codon. Especially, the exchange of a triplet located close to non-pairing bases may remain effectless. The same holds true for triplet substitutions that only change one base. In such cases, the simultaneous replacement of multiple triplets is essential. Though, if there is a high density of secondary structure candidates, multiple exchanges may provide more stability to neighboring secondary structures, giving rise to the formation of new secondary structures.

When considering secondary structures with a high stem length, it may happen that a secondary structure is not resolved by the exchange of one triplet but e.g. split it into two secondary structues. One possible strategy could be to resolve all secondary structures in the analysed frame once a secondary structure has been identified to thereafter consider the next secondary structure. The other way might be to process alternating single secondary structures to be able to react to changes of neighboring codon sequences. The different approaches and their effect on the variance and quality of the candidate solutions will be investigated and a suitable implementation will be chosen.

3.2 Choosing a suitable metaheuristic approach

From the set of metaheuristic approaches, the following could address the needs of this project.

- **Independent parallel optimization runs**
  Based on the initial sequence, independent parallel optimization runs that start with the same input sequence and diverge with each step can be performed as well as methods using knowledge about already explored paths. Although independent runs will often get stuck in local optima, a wide variety of paths will be explored and good approximate solutions will be found using an appropriate number of runs.

- **Metaheuristic Methods**
  As a more targeted approach, search methods that accept worse neighbours can be used to keep going when stuck in local optima. In particular, it can be helpful here to choose methods that store the result of already explored paths and invest the computation time in investigating unexplored paths. Possible methods include the following.

  - **Tabu search**, designed to avoid cycles by tagging visited solutions as tabu and storing them in a memory structure, called tabu list [Glover, 1989], [Glover, 1990]
  - **Simulated Annealing**, designed to avoid getting stuck in local optima by accepting worse neighbours of a current solution while preferring better neighbours [Kirkpatrick et al., 1983]

In addition to considering different heuristics, care should also be taken to find a reasonable tradeoff between reevaluation of existing secondary structures and replacements, since evaluation is fairly resource-intensive.

If possible, the optimization problem should be extended by a further optimization objective, which has not been modeled so far, to limit the problem in its complexity. Specifically, a further goal is to orient the percentage distribution of triplets used to the codon usage of the target organism. This additional optimization goal contradicts the first one [Liu et al., 2013], because both goals limit the number of codon triplets to be used. Therefore, it should be examined to what extent consideration of the organism-specific codon usage is possible.

3.3 Evaluation

Using synthetic codon sequences with adaptable structural characteristics, the runtime of the existing implementation is to be tested, by changing one parameter at a time while keeping the others constant. The most influential parameters should then be selected and solutions developed to address the problems that cause the greatest effect on runtime. The developed scenarios will later be applied to implementations of different optimization approaches to support the process of selecting appropriate optimization methods. The findings collected using synthetic codon sequences will be validated on a selection of real sequences that fit into the carrying capacity of AAV which is between 4.1 and 4.9 kb [Dong et al., 1996]. Transgenes of practical relevance, suitable to ensure the algorithms applicability to real data include base editors, e.g. ABE7.10 [Gaudelli et al., 2020], hemophilia treatments using factor IX transgenes [George et al., 2017] and recombinases that recognize virus-specific sequences, e.g. Brec-1 [Karpinski et al., 2016].

4 Related Work

Among the DNA optimization algorithms that already exist there are algorithms that focus on individual optimization goals relevant to this work, which are namely codon usage optimization [Puigbò et al., 2007], [Chung and Lee, 2012] and secondary structure avoidance [Gaspar et al., 2013]. As non of these meet all the requirements of this project, we already implemented a suitable but non-runtime optimized single-objective secondary structure destruction algorithm [Kranz, 2021].

Besides the aforementioned single-objective algorithms, there are multi-objective algorithms, aiming to combine at least two, possibly conflicting, optimization goals; however, not necessarily the optimization goals relevant to us. For example, there is one multi-objective algorithm [Gaeta et al., 2021], which optimizes codon usage and can prevent clustered occurrence of GC sites but does not support the destruction of secondary structures. An algorithm that opts for codon usage optimization and the destruction of secondary structures is JCat [Grote et al., 2005]. However, in this work secondary structures are only detected and destroyed if they are followed by a poly-U stretch, because then a rho independent transcription terminator is formed, which is the actual optimization objective. Another multi-objective algorithm that includes codon usage and the avoidance of secondary structures among its optimization goals is COStar [Liu et al., 2013]. Here, the problem is decomposed into subproblems whose possible solutions are modeled as an acyclic graph in a way that the best solution corresponds to the path following the best partial solutions. This procedure cannot detect secondary structures that form across windows and its detection algorithm is based on a free energy approach, which is not detecting every occurrence of secondary structures.

In addition to concrete implementations, there are also software libraries that aim to provide a basis for specific implementations in the field of DNA optimization. A suitable implementation is DNA-Chisel [Zulkower and Rosser, 2019], which is a Python framework for multi-objective algorithms. The codebase contains a basic method for dealing with multiple optimization targets as well as an approach for the detection and destruction of secondary structures.

References

[Cataldi and McCarty, 2013] Cataldi, M. P. and McCarty, D. M. (2013). Hairpin end conformation of adeno-associated virus (aav) genome determines interactions with dna repair pathways. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3578132/.

[Chung and Lee, 2012] Chung, B. and Lee, D.-Y. (2012). Computational codon optimization of synthetic gene for protein expression. *BMC Systems Biology*, 6:134 – 134.

[Dong et al., 1996] Dong, J.-Y., Fan, P.-D., and Frizzell, R. A. (1996). Quantitative analysis of the packaging capacity of recombinant adeno-associated virus. *Human Gene Therapy*, 7(17):2101–2112. 10.1089/hum.1996.7.17-2101.

[Gaeta et al., 2021] Gaeta, A., Zulkower, V., and Stracquadanio, G. (2021). Design and assembly of DNA molecules using multi-objective optimization. *Synthetic Biology*, 6(1). https://doi.org/10.1093/synbio/ysab026.

[Gaspar et al., 2013] Gaspar, P., Moura, G., Santos, M. A. S., and Oliveira, J. L. (2013). mrna secondary structure optimization using a correlated stem–loop prediction. https://academic.oup.com/nar/article/41/6/e73/2902446.

[Gaudelli et al., 2020] Gaudelli, N. M., Lam, D. K., Rees, H. A., Solá-Esteves, N. M., Barrera, L. A., Born, D. A., Edwards, A., Gehrke, J. M., Lee, S.-J., Liquori, A. J., Murray, R., Packer, M. S., Rinaldi, C., Slaymaker, I. M., Yen, J., Young, L. E., and Ciaramella, G. (2020). Directed evolution of adenine base editors with increased activity and therapeutic application. *bioRxiv*. 10.1101/2020.03.13.990630.

[George et al., 2017] George, L. A., Sullivan, S. K., Giermasz, A., Rasko, J. E., Samelson-Jones, B. J., Ducore, J., Cuker, A., Sullivan, L. M., Majumdar, S., Teitel, J., McGuinn, C. E., Ragni, M. V., Luk, A. Y., Hui, D., Wright, J. F., Chen, Y., Liu, Y., Wachtel, K., Winters, A., Tiefenbacher, S., Arruda, V. R., van der Loo, J. C., Zelenaia, O., Takefman, D., Carr, M. E., Couto, L. B., Anguela, X. M., and High, K. A. (2017). Hemophilia b gene therapy with a high-specific-activity factor ix variant. *New England Journal of Medicine*, 377(23):2215–2227. 10.1056/NEJMoa1708538.

[Glover, 1989] Glover, F. (1989). Tabu search—part i. *Journal on Computing*, 1(3):190–206. https://doi.org/10.1287/ijoc.1.3.190.

[Glover, 1990] Glover, F. (1990). Tabu search—part ii. *Journal on Computing*, 2(1):4–32. https://doi.org/10.1287/ijoc.2.1.4.

[Grote et al., 2005] Grote, A., Hiller, K., Scheer, M., Münch, R., Nörtemann, B., Hempel, D. C., and Jahn, D. (2005). JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. *Nucleic Acids Research*, 33(suppl_2):W526–W531. https://doi.org/10.1093/nar/gki376.

[Karpinski et al., 2016] Karpinski, J., Hauber, I., Chemnitz, J., Schäfer, C., Paszkowski-Rogacz, M., Chakraborty, D., Beschorner, N., Hofmann-Sieber, H., Lange, U. C., Grundhoff, A., Hackmann, K., Schrock, E., Abi-Ghanem, J., Pisabarro, M. T., Surendranath, V., Schambach, A., Lindner, C., van Lunzen, J., Hauber, J., and Buchholz, F. (2016). Directed evolution of a recombinase that excises the provirus of most hiv-1 primary isolates with high specificity. https://www.nature.com/articles/nbt.3467.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680. h10.1126/science.220.4598.671.

[Kranz, 2021] Kranz, J. (2021). Development of an algorithm to identify and eliminate pseudo-itr structures from aav transgenes. Humboldt University Berlin, student project.

[Liu et al., 2013] Liu, X., Deng, R., Wang, J., and Wang, X. (2013). Costar: A d-star lite-based dynamic search algorithm for codon optimization. *Journal of theoretical biology*, 344. 10.1016/j.jtbi.2013.11.022.

[Nabel, 1999] Nabel, G. J. (1999). Development of optimized vectors for gene therapy. *Proceedings of the National Academy of Sciences*, 96(2):324–326. https://www.pnas.org/content/96/2/324.

[Puigbò et al., 2007] Puigbò, P., Guzmán, E., Romeu, A., and Garcia-Vallvé, S. (2007). OPTIMIZER: a web server for optimizing the codon usage of DNA sequences. *Nucleic Acids Research*, 35(suppl_2):W126–W131. https://doi.org/10.1093/nar/gkm219.

[Watson and Crick, 1953] Watson, J. D. and Crick, F. H. C. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738. https://doi.org/10.1038/171737a0.

[Yan et al., 2005] Yan, Z., Zak, R., Zhang, Y., and Engelhardt, J. F. (2005). Inverted terminal repeat sequences are important for intermolecular recombination and circularization of adeno-associated virus genomes. *Journal of Virology*, 79(1):364–379. https://journals.asm.org/doi/abs/10.1128/JVI.79.1.364-379.2005.

[Zulkower and Rosser, 2019] Zulkower, V. and Rosser, S. (2019). Dna chisel, a versatile sequence optimizer. 10.1101/2019.12.16.877480.