EFM Optimizer User Guide

sTAUbility – TAU Israel 2020 iGEM Team

Welcome to the Evolutionary Failure Mode (EFM) Optimizer! We hope that using our product, you will be able to easily design synthetic sequences with higher stability and expression level.

Our basic service will allow you to easily find and rank **simple sequence repeats (SSR)** and **repeat mediated deletions (RMD)** sites for many files at once. These are mutational hotspots, which reduce genetic stability significantly [1-2].

In mammalian or insectoid cells, methylation sites become significant factors in the host's stability, as they are epigenetic inheritance hotspots. For these cells, we offer the ability to find **methylation sites** as well.

Finally, we believe that working with a list of problematic sites is inconvenient. We offer a further service of **optimizing the input sequence**. It is optimized for the avoidance of these sites, preservation of GC content specified by the user, and codon usage. In essence, you will provide input genetic sequences and receive an equivalent sequence, optimized for stability and expression level. At the end of the optimization process, the software will produce a zip report, including the annotated final sequence with the marked changes and CSV files that lists the successful constraints and objectives.

For the theoretical background, please refer to the documentation in our website: https://2020.igem.org/Team:TAU_Israel/Contribution

For questions and suggestions, contact the developers at igem.tau.2019@gmail.com and mention "EFM Optimizer V1" in the title.

In the following sections, we will provide instructions on the use of these components.

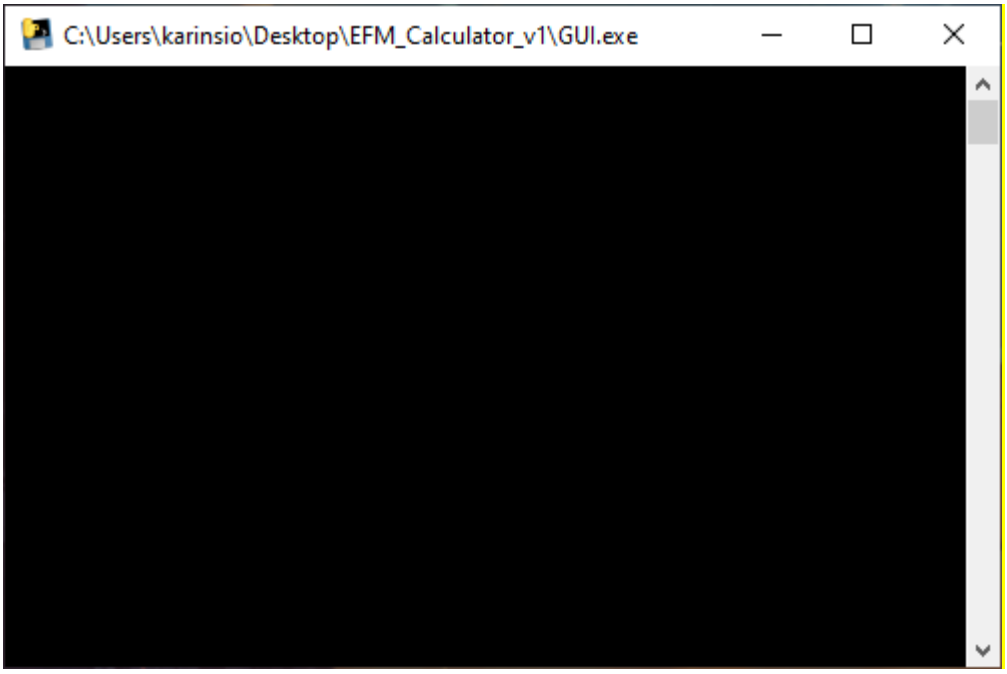
Installation instructions – Beta Version

Our beta version is available in the "EFM_Calculator_v1" folder. The folder includes the actual software in the "GUI.exe" file and another subfolder called "In", as well as the user guide.

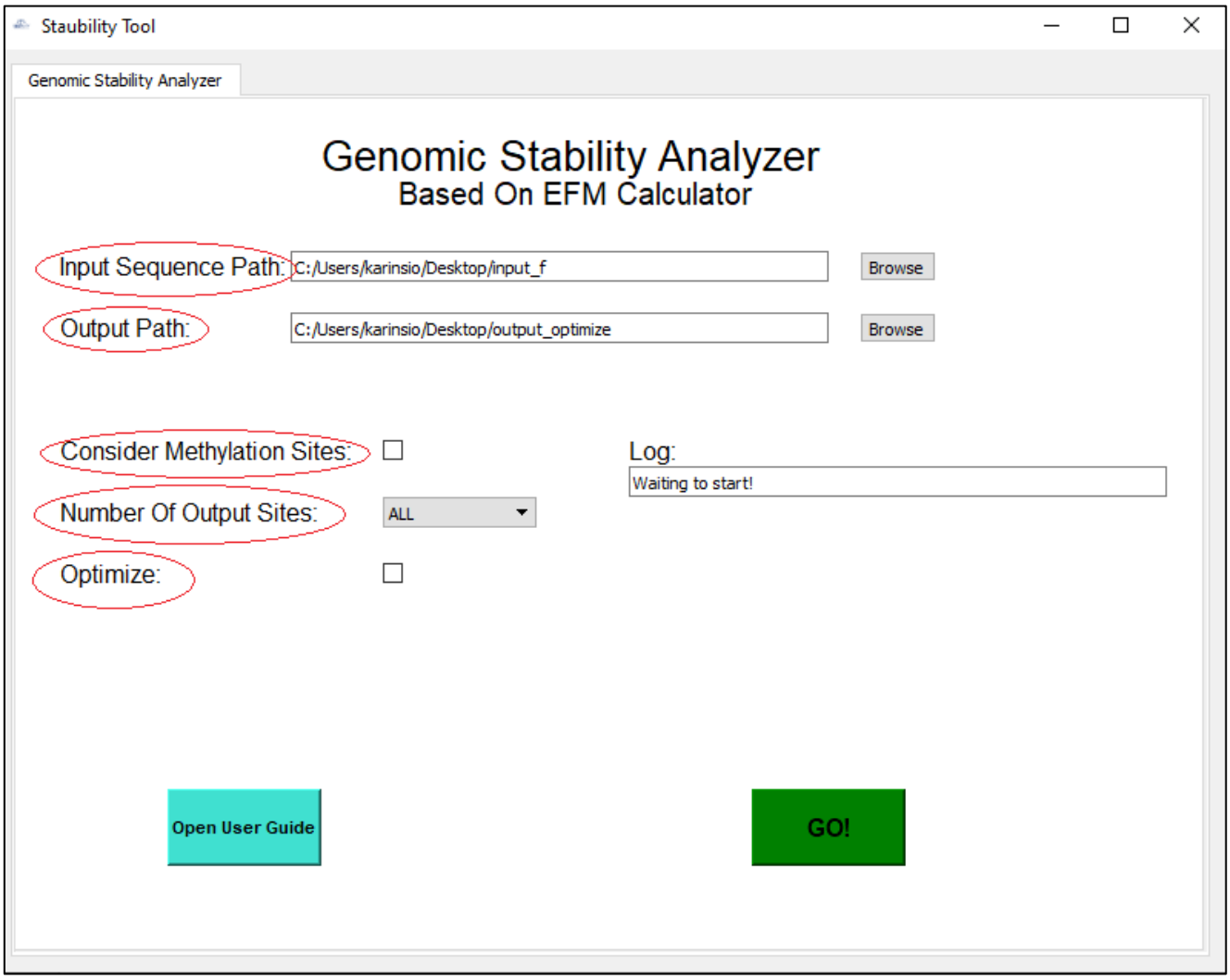
Please make sure that you keep these components in the same folder! Otherwise, the software will not work. Do not change the folder names.

To launch the software:

1. After downloading the "EFM_Calculator_v1" folder with all its content, run the "GUI.exe" file **from the same folder**.
2. Your operating system may ask for permissions to run the program, grant them.
3. A CMD window will open. This is a black window, where notes will appear while the processes continue, allowing you to track the progress of the mutational-sites detection and optimization step for each inserted FASTA file.
4. The main software window will launch a few minutes after the CMD.


C:\Users\karinsio\Desktop\EFM_Calculator_v1\GUI.exe

Main Window:


Staubility Tool
Genomic Stability Analyzer
Genomic Stability Analyzer
Based On EFM Calculator
Input Sequence Path:
C:/Users/karinsio/Desktop/input_f
Browse
Output Path:
C:/Users/karinsio/Desktop/output_optimize
Browse
Consider Methylation Sites:
Log:
Waiting to start!
Number Of Output Sites:
ALL
Optimize:
Open User Guide
GO!

Input Sequence Path:

In this field, you define which genetic sequences you wish to analyze. Use the "browse" button to select a directory in which your FASTA files appear. The selected directory and all its subdirectories will be searched for **FASTA** and **fasta.gz** files. The analysis will be performed on all sequences contained within.

Output Path:

In this field, you define in which directory to output the software's results. Each input file's relative path is defined as a directory within the output path, and the results for the file are exported to this directory. This ensures the relative ordering between inputs is maintained.

For example, if your **input** directory contains the following the following setup:

- input_folder_1
  - sequences_1A.fasta (one sequence within)
  - sequences_1B.fasta.gz (one sequence within)
- input_folder_2
  - sequences_2A.fasta (two sequences within)

Then your specified **output directory** will include:

- output_folder
  - input_folder_1
    - sequences_1A
      - Optimization_report_0.zip
      - methylation_sites.csv
      - recombination_sites.csv
      - slippage_sites.csv
    - sequences_1B
      - Optimization_report_0.zip
      - methylation_sites.csv
      - recombination_sites.csv
      - slippage_sites.csv

 input_folder_2

 sequences_2A

 Optimization_report_0.zip

 Optimization_report_1.zip

 methylation_sites.csv

 recombination_sites.csv

 slippage_sites.csv

Where  and  are new folders that were created to match the relative input order. Each subfolder  contains the per-sequence output as will be described below.

Please note that a single FASTA file can contain **many sequences**, as in "sequences_2A.fasta" in the example. In this case, the software will refer to the sequences with an **index**, starting from 0. For instance, 'sequences_2A.fasta' contained two sequences, so the first will be assigned with index '0' and the second will be assigned with '1'.

We will introduce the content of each output file after further explanation of the parameters in the GUI.

Consider Methylation Sites:

With this tick box, you can select whether to find and avoid methylation sites.

These sites are major hotspots within mammalian and insectoid cells [3-6]. For these cells, searching for methylation sites is vital.

However, methylation sites take significantly more computing power to find, resulting in much longer runtimes. Thus, do not tick this box for other types of cells.

Number of output sites:

In this field, you define how many sites of each type to avoid, where the most unstable hotspots are avoided first. This parameter defines a **trade-off between stability and expression** – the more hotspots considered, the fewer degrees of freedom are reserved for optimizing expression.

Optimize:

With this tick box, you can select whether to perform optimization on the input sequence and return a new sequence, ready for use. **This is highly recommended, as improving the sequence manually normally takes much effort.**

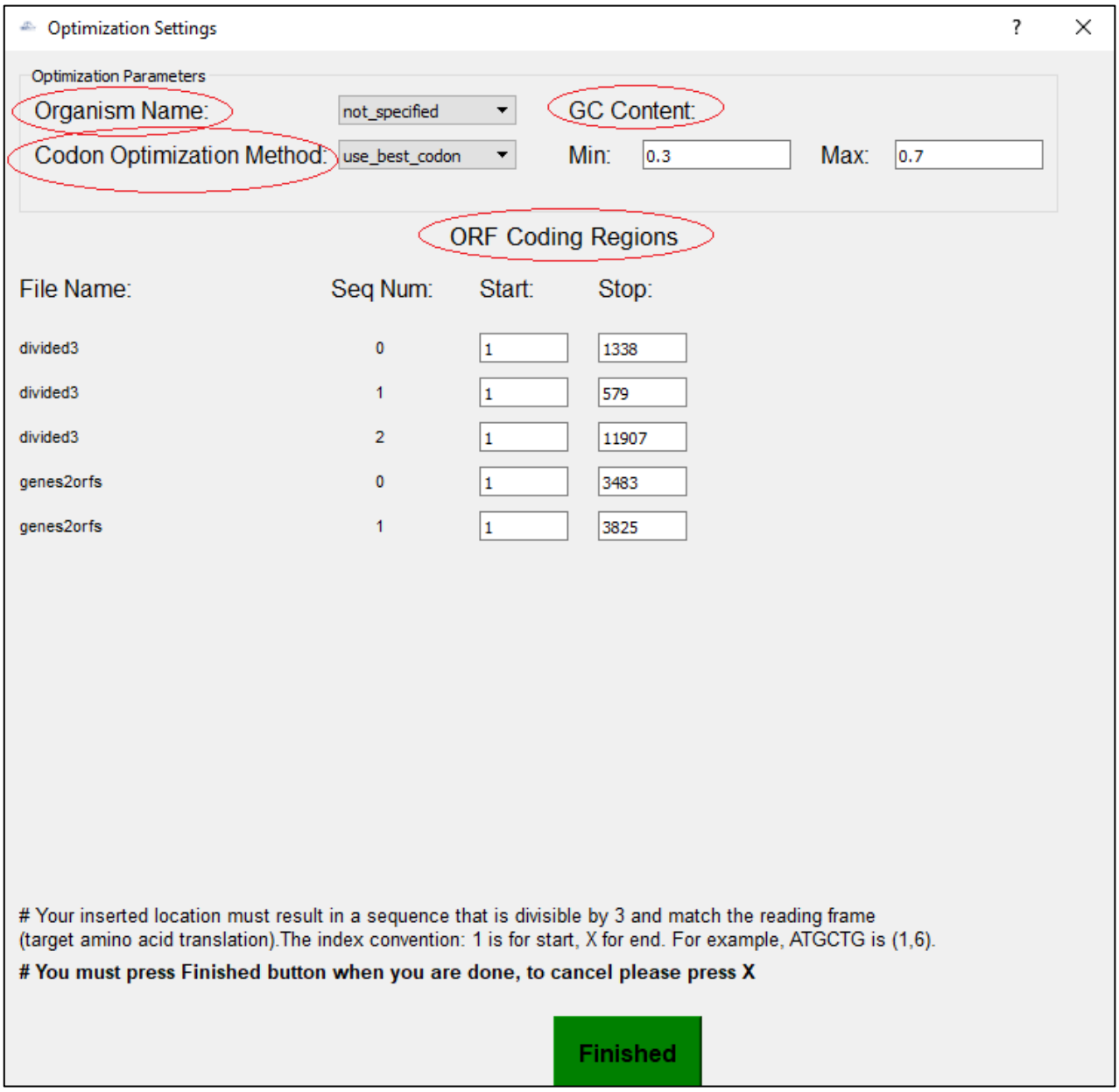
This optimization is performed for three objectives:

1. Avoidance of the hotspots found – improves genomic stability. For computational considerations, we only offer to avoid the first ten sites from each type – ten RMD ('recombination') sites, ten SSR ('slippage') sites, and ten methylation sites when relevant.
2. GC content – maintaining the frequency of GC nucleotides within a specified range. This is important for various biological purposes, including maintaining a high expression level and genomic stability.
3. Codon usage – replacing the codons used to generate amino acids in order to match the relative codon frequency within the host organism. The underlying assumption is that the genome of the host went through selective pressure for stability and expression in some form. Thus, by matching the sequence to the host, it will likely have higher levels of stability and expression as well.

When the optimize option is ticked, a menu of optimization parameters will open, which will now be introduced.

Note: at this stage, our software is limited to optimizing ten sequences. If there are more than ten sequences within the input directory, the optimization module will not be available.

Optimization Window:



Organism name:

In this drop-down menu, it is possible to select a host organism. The codon usage optimization of the sequence would attempt to match this organism. The default value is 'not specified', meaning the host organism is not available and codon optimization objective will not be defined during optimization. The remaining optimization modules will be EFM constraints (mutational hotspots) and GC content.

If your host organism is not within the proposed list, it is currently not supported. Please contact us if this is the case (the contact information can be found at the end of this document).

Codon optimization method:

In this drop-down menu, different optimization methods can be selected for codon usage. **The default value is Use best codon**, and as it is the most popular method in literature, will often serve your needs well.

These are the supported optimization methods:

1. "Use best codon" – in this method, each codon is replaced by the "optimal" (i.e. most frequent) synonymous codon in the target organism. This is equivalent to the Codon Adaptation Index (CAI) optimization.
2. “Match codon usage” – in this method, the optimization process matches the final sequence's codon usage profile to the target species’ profile.
3. “Harmonize RCA” – in this method, each codon in the sequence is replaced by a synonymous codon. Two codons are considered synonymous if the first codon’s frequency in the final sequence is approximately the second codon’s frequency in the host organism.

GC content:

You can select the minimal and maximal allowed values for GC content. This value determines the frequency of G and C within a genetic frequency, normalized between zero and one for no GC and all GC, respectively.

It is assumed that for lower values of GC content, a sequence is more stable, since it has been proven that genes with high GC content had a substantially elevated rate of mutations, both single-base substitutions and deletions [5]. On the other hand, allowing only low GC content values would limit the optimization’s degrees of freedom, and perhaps allow lower maximal expression.

The default values that allow reasonable optimization are 0.3-0.7.

ORF coding regions:

For each file and sequence within the file, you will be requested to provide an open reading frame (ORF) region to be codon optimized. This region is defined by a start and end index.

All ORF regions must have lengths divisible by 3. Otherwise, they cannot be divided into codons.

The indexing convention used is:

1. The first index is 1.
2. The last index is included in the ORF.

For example, an ORF with indexes (1, 6) would be the first six nucleotides of the sequence.

The default values for the start and end indexes are 1 and the last index in the sequence. **The start index must be an integer greater than 0. The end index must be an integer, and at most, the last index in the sequence.**

The optimization preserves the coding sequence (CDS, amino-acid translation of the reading frame). Thus, it is very important to carefully fill the indices, otherwise the wrong ORF will be chosen and a completely different protein will be produced.

Finish:

When you have inserted all the parameters, click on the "finish" button. You will be back in the main window. At any time, you can return to the optimization window by using the 

button that now appears in the main window next to the Optimize thick box. The parameters you entered are kept each time you use the optimization window.

The content of the output folder

As stated above, the output folder will include a list of folders, where each folder maps to a single input FASTA file, matching the setup of the input folder.

Inside each sub-folder, you will find:

1. A list of the mutational hotspots by category (CSV format). Usually, there are two such files, but when 'consider methylation' is specified, there will be an additional CSV file that lists the methylation sites.
   Each of these files includes a 'sequence_number' column corresponding to the sequence index within the FASTA file.
2. An optimization report (zip format) when 'optimize' is specified.

Mutational report – csv files

The hotspots found are divided into types: Simple Sequence Repeats, Repeat Mediated Deletions, and methylation. Each of these types are summarized in a different csv file.

Simple Sequence Repeats (SSR)

These sites are composed of repeating base units. In the translation process, this could cause a polymerase slippage mistake, which would add or remove a base unit. These hotspots are ranked according to their instability in a file named slippage_sites.csv, which includes the following columns:

Start/end: the start/end index of the site within the respective sequence. The convention used is that the first index in a sequence is 0, the start index is included in the site, and the end is excluded.

Length_base_unit: the length of the repeating base unit.

Sequence: the nucleotide sequence of the SSR site.

Num_base_units: how many repeating units are within the site.

log10_prob_slippage_ecoli: the instability score of a site. It is linear with the empirical mutational probability in E. Coli. While the mutational probabilities for other organisms differ, they will be monotone with this score, and thus this score is a meaningful instability measure.

Sequence_number: the sequence index within the file. 0 corresponds to the first sequence, 1 to the second sequence, and so on.

Repeat Mediated Deletions (RMD)

RMD sites are long, identical sequences appearing in different locations within the genome. This could lead to a recombination error, where the intermittent genetic code is deleted.

These hotspots are ranked according to their instability in a file named recombination_sites.csv, which includes the following columns:

Start/end_1/2: the start/end index of the first/second site with the identical sequence. The convention used is that the first index in a sequence is 0, the start index is included in the site, and the end is excluded.

Sequence: the repeating nucleotide sequence of the RMD sites.

Location_delta: the distance between the end of the first site to the start of the second site. The closer they are, the more likely the error.

Site_length: the repeating sequence's length. The longer it is, the more likely the error.

log10_prob_recombination_ecoli: the instability score of a site. It is linear with the empirical mutational probability in E. Coli. While the mutational probabilities for other organisms differ, they will be monotone with this score, and thus this score is a meaningful instability measure.

Sequence_number: the sequence index within the file. 0 corresponds to the first sequence, 1 to the second sequence, and so on.

Methylation Sites

Methylation sites are significant epigenetic inheritance hotspots in mammalian and insectoid cells. They increase DNA folding and reduce expression levels. For these types of cells, it is imperative to avoid methylation sites as well.

In order to detect these sites, we used a database containing 313 reported **methylation motifs** by Wang. et al 2019 [4]. We compare each subsequence within your gene with each methylation site in a probabilistic manner and ranking the subsequences by their likelihood of being a methylation site.

These hotspots are ranked according to their likelihood to be methylation sites in a file named methylation_sites.csv, which includes the following columns:

Actual site: sequence matched.

Actual_site_rev_conj: the reverse conjugate of the site matched. Relevant, as a methylation site can be found on the reverse conjugate as well.

Matching_motif: methylation site detected.

Start/end_index: the start/end index of the methylation site. The convention used is that the first index in a sequence is 0, the start index is included in the site, and the end is excluded.

log10_site_match: the probabilistic likelihood of the subsequence matching the methylation site, measured in log scale. A score of 0 means a perfect match, and the lower the score, the lesser the match, and the less likely the site is to perform methylation.

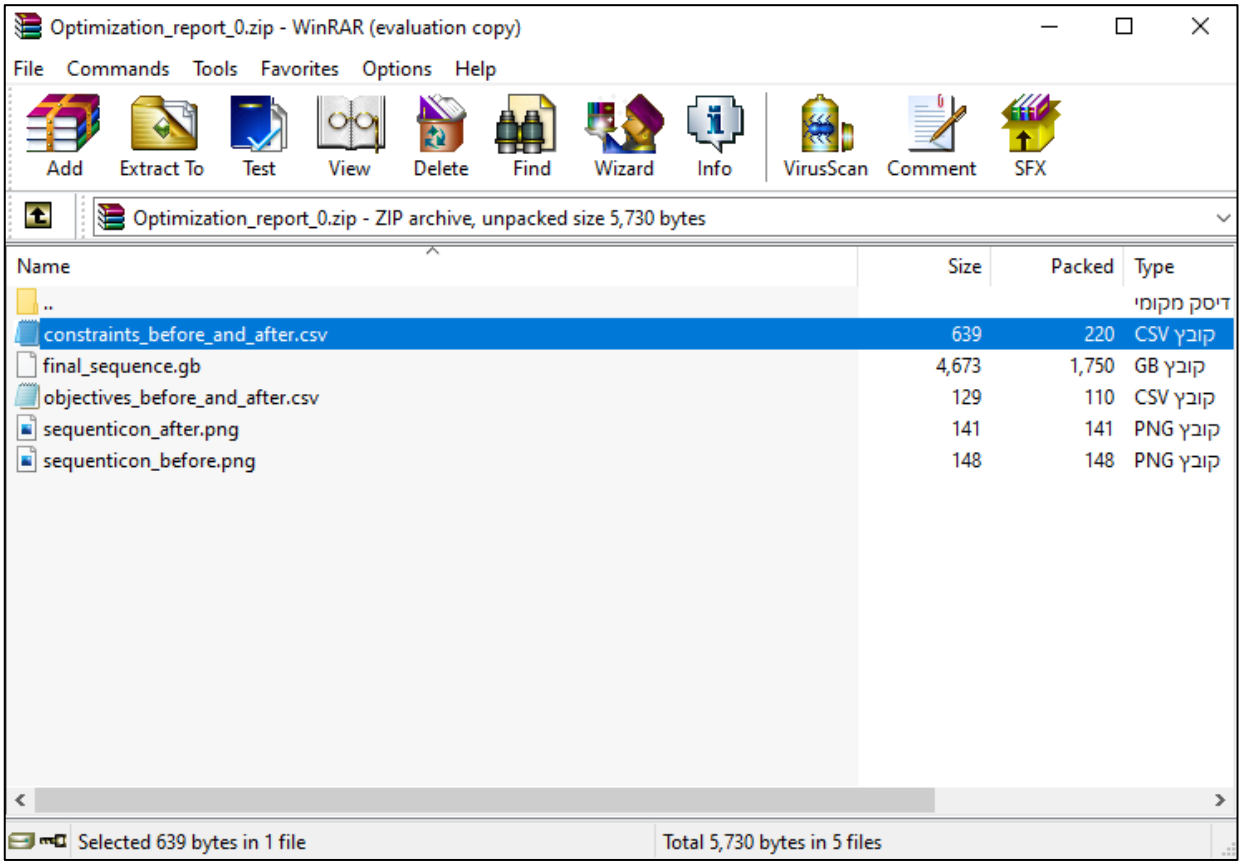
Num_nucleotides: how many nucleotides are matched by the methylation site.

0-13: the probability of each nucleotide for each position along the methylation site.

Sequence_number: the sequence index within the file. 0 corresponds to the first sequence, 1 to the second sequence, and so on.

Optimization report – zip file

Each zip file contains the following files:



Files description:

1. **Objective_before_and_after.csv –**

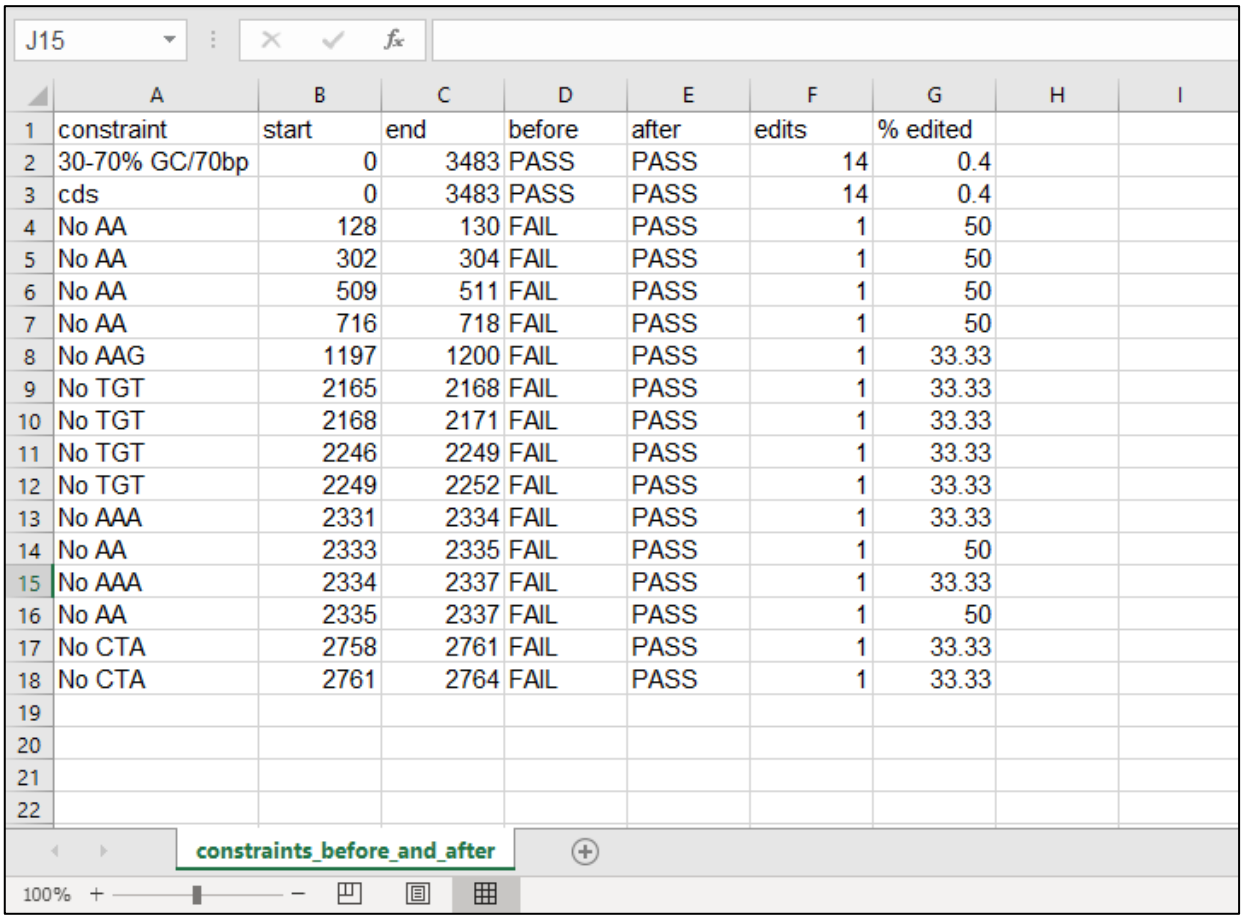
   Reports the objective score in CSV format. For example:

	A	B	C	D	E	F	G	H
1	objective	boost	start	end	before	after	edits	% edited
2	best-codon-optimize (s_cerevisiae)	1	0	3483	-114.38	-122.82	14	0.4
3								
4								
5								

   The objective function in this example was to optimize the codon usage based on S. Cerevisiae as the host organism.
   When the organism name is "not_specified", this file will be empty, since unlike the constraints, the only objective in our software is the codon optimization.

2. **Contraints_before_and_after.csv** –

   contains the objective and the constraints that were successfully maintained in CSV format. For example:



	A	B	C	D	E	F	G	H	I
1	constraint	start	end	before	after	edits	% edited		
2	30-70% GC/70bp	0	3483	PASS	PASS	14	0.4		
3	cds	0	3483	PASS	PASS	14	0.4		
4	No AA	128	130	FAIL	PASS	1	50		
5	No AA	302	304	FAIL	PASS	1	50		
6	No AA	509	511	FAIL	PASS	1	50		
7	No AA	716	718	FAIL	PASS	1	50		
8	No AAG	1197	1200	FAIL	PASS	1	33.33		
9	No TGT	2165	2168	FAIL	PASS	1	33.33		
10	No TGT	2168	2171	FAIL	PASS	1	33.33		
11	No TGT	2246	2249	FAIL	PASS	1	33.33		
12	No TGT	2249	2252	FAIL	PASS	1	33.33		
13	No AAA	2331	2334	FAIL	PASS	1	33.33		
14	No AA	2333	2335	FAIL	PASS	1	50		
15	No AAA	2334	2337	FAIL	PASS	1	33.33		
16	No AA	2335	2337	FAIL	PASS	1	50		
17	No CTA	2758	2761	FAIL	PASS	1	33.33		
18	No CTA	2761	2764	FAIL	PASS	1	33.33		
19									
20									
21									
22									

   In this example, the first constraint is to keep the GC content in the range of 30%-70% in windows of 70 base-pairs. The "start" and "end" columns specify the indices within the sequence that should maintain this constraint. The "before" and "after" contains the constraint status before and after optimization – the GC content was 30%-70% before optimization (PASS) and was kept that way.

   The second constraint is "CDS", which is short for Coding Sequence, meaning the amino-acid translation was preserved.

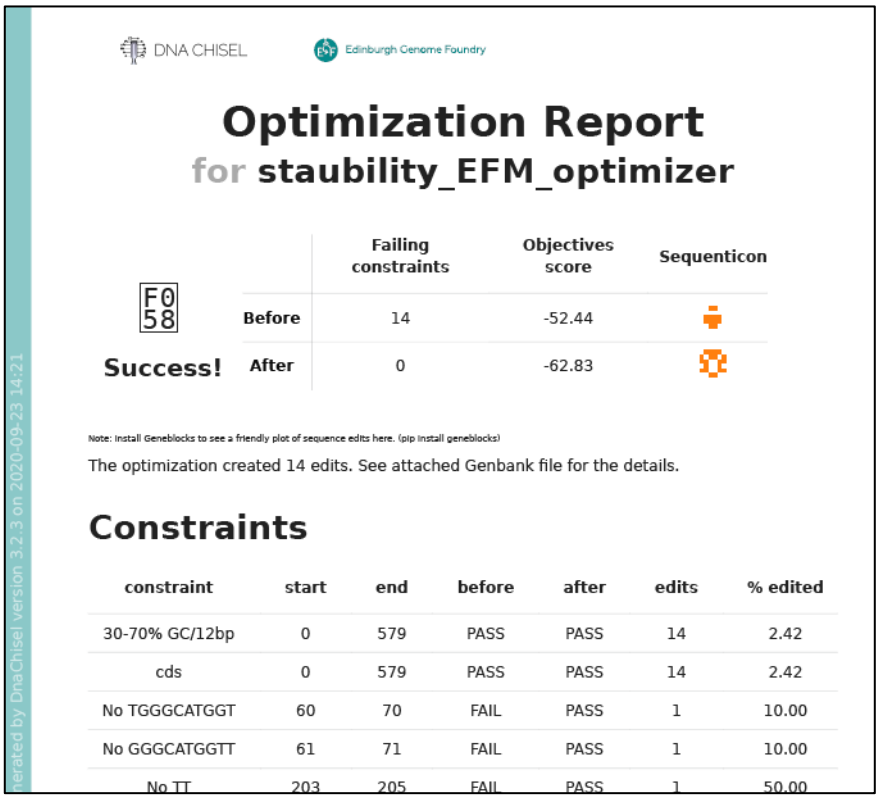
   From the third constraint, we see the mutational hotspots that were detected and avoided. For example, "no AA" represents avoidance of the AA repeating sequence in this region, as it increases the probability of a polymerase slippage error.

3. **Final_sequence.gb** –
   the optimized sequence in GenBank format, not annotated.
4. **Final_sequence_with_edits.gb** –
   the optimized sequence in GenBank format, annotated with the edits during optimization. Thus, you can conveniently track the changes during the process.
5. **Sequenticon_before.png** –
   Sequenticon is an icon unique to the final output sequence. This is an important feature, especially when dealing with large sets of input sequences (which are often renamed or updated), because it enables the user to differentiate between sequences which otherwise might get confused with one another. Sequenticons provide a simple visual way to know that two sequences are different

(different identicons) or very probably the same (same identicon). The "Sequenticon_before.png" refers to the sequence before optimization.

6. **Sequenticon_after.png** –
   refers to the final sequence, after optimization.

In some computers, the software will also produce a PDF report. This property requires installation of the WeasyPrint package. An example of such a report appears below. It contains the information presented in the other output files, displayed in a single PDF file. If you are interested in obtaining a PDF report automatically as well, refer to this link.



DNA CHISEL Edinburgh Genome Foundry

Optimization Report
for staubility_EFM_optimizer

		Failing constraints	Objectives score	Sequenticon
F058	Before	14	-52.44	
Success!	After	0	-62.83	

Note: Install Geneblocks to see a friendly plot of sequence edits here. (pip install geneblocks)

The optimization created 14 edits. See attached Genbank file for the details.

Constraints

constraint	start	end	before	after	edits	% edited
30-70% GC/12bp	0	579	PASS	PASS	14	2.42
cds	0	579	PASS	PASS	14	2.42
No TGGGCATGGT	60	70	FAIL	PASS	1	10.00
No GGGCATGGTT	61	71	FAIL	PASS	1	10.00
No TT	203	205	FAIL	PASS	1	50.00

Generated by DnaChisel version 3.2.3 on 2020-09-23 14:21

References

[1] Jack, B. R., Leonard, S. P., Mishler, D. M., Renda, B. A., Leon, D., Suárez, G. A., & Barrick, J. E. (2015). Predicting the Genetic Stability of Engineered DNA Sequences with the EFM Calculator. ACS synthetic biology, 4(8), 939–943. https://doi.org/10.1021/acssynbio.5b00068

[2] Renda, B. A., Hammerling, M. J., & Barrick, J. E. (2014). Engineering reduced evolutionary potential for synthetic biology. Molecular bioSystems, 10(7), 1668–1678. https://doi.org/10.1039/c3mb70606k

[3] Greenberg, M.V.C., Bourc'his, D. The diverse roles of DNA methylation in mammalian development and disease. *Nat Rev Mol Cell Biol* **20,** 590–607 (2019). https://doi.org/10.1038/s41580-019-0159-6

[4] Curradi, M., Izzo, A., Badaracco, G., & Landsberger, N. (2002). Molecular mechanisms of gene silencing mediated by DNA methylation. *Molecular and cellular biology*, *22*(9), 3157–3173. https://doi.org/10.1128/mcb.22.9.3157-3173.2002

[5] Newell-Price, J., Clark, A. J., & King, P. (2000). DNA methylation and silencing of gene expression. *Trends in endocrinology and metabolism: TEM*, *11*(4), 142–148. https://doi.org/10.1016/s1043-2760(00)00248-4

[6] Baylin, S. DNA methylation and gene silencing in cancer. *Nat Rev Clin Oncol* **2,** S4–S11 (2005). https://doi.org/10.1038/ncponc0354

[7] Hale, R. S., & Thompson, G. (1998). Codon Optimization of the Gene Encoding a Domain from Human Type 1 Neurofibromin Protein Results in a Threefold Improvement in Expression Level inEscherichia coli. Protein Expression and Purification, 12(2), 185-188.

[8] Claassens, N. J., Siliakus, M. F., Spaans, S. K., Creutzburg, S. C., Nijsse, B., Schaap, P. J., ... & Van Der Oost, J. (2017). Improving heterologous membrane protein production in Escherichia coli by combining transcriptional tuning and codon usage algorithms. PloS one, 12(9), e0184355.