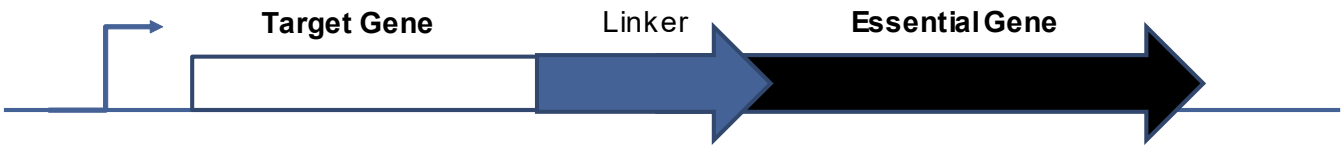
Staubility Enhancer User Guide

sTAUbility – TAU Israel 2020 iGEM Team

Welcome to the Staubility Enhancer! We hope that by using our product, you will be able to easily design synthetic sequences which have higher stability and expression levels.

A key challenge in the field of synthetic biology is genomic instability of introduced genes. Once a gene is inserted into a host organism, it can cause an additional metabolic load, significantly reducing the host's fitness. Therefore, mutations that damage the introduced gene are likely to be selected for, diminishing its expression. These mutations may render synthetic biology products obsolete, thus requiring constant maintenance.

We propose interlocking a target gene to the N-terminus of an essential gene in the host's genome, under the same promoter. This way, mutations on the target gene are likely to affect the expression of the essential gene, leading to the mutated host's mortality.



To implement this idea, we developed a software called the sTAUbility Enhancer. Given a target gene, the Enhancer provides you with the best matching conjugated gene candidates. After you select the gene and preferred linker type, the software yields the best linker candidates. Following the selection of a linker, the software scans the construct for mutational and epigenetic hotspots. Using these sites and additional optimization parameters, it optimizes the combined construct for efficient gene expression and increased stability, providing you with a sequence tailored to your needs.

In summary, **our software works in three steps:**

1. Choose the best fitting conjugated gene
2. Select the fitting linker
3. Optimize the final construct for efficient translation and increased stability

Accordingly, the software consists of three separate windows, each allows to control the appropriate parameters for the specific step.

For further reading on the theoretical background, please refer to the documentation in our website: https://2020.igem.org/Team:TAU_Israel/Model and https://2020.igem.org/Team:TAU_Israel/Software

For questions and suggestions, contact the developers at: igem.tau.2019@gmail.com. Please mention "The Staubility Enhancer V1" in the title.

In the following sections, we will provide usage instructions for these components.

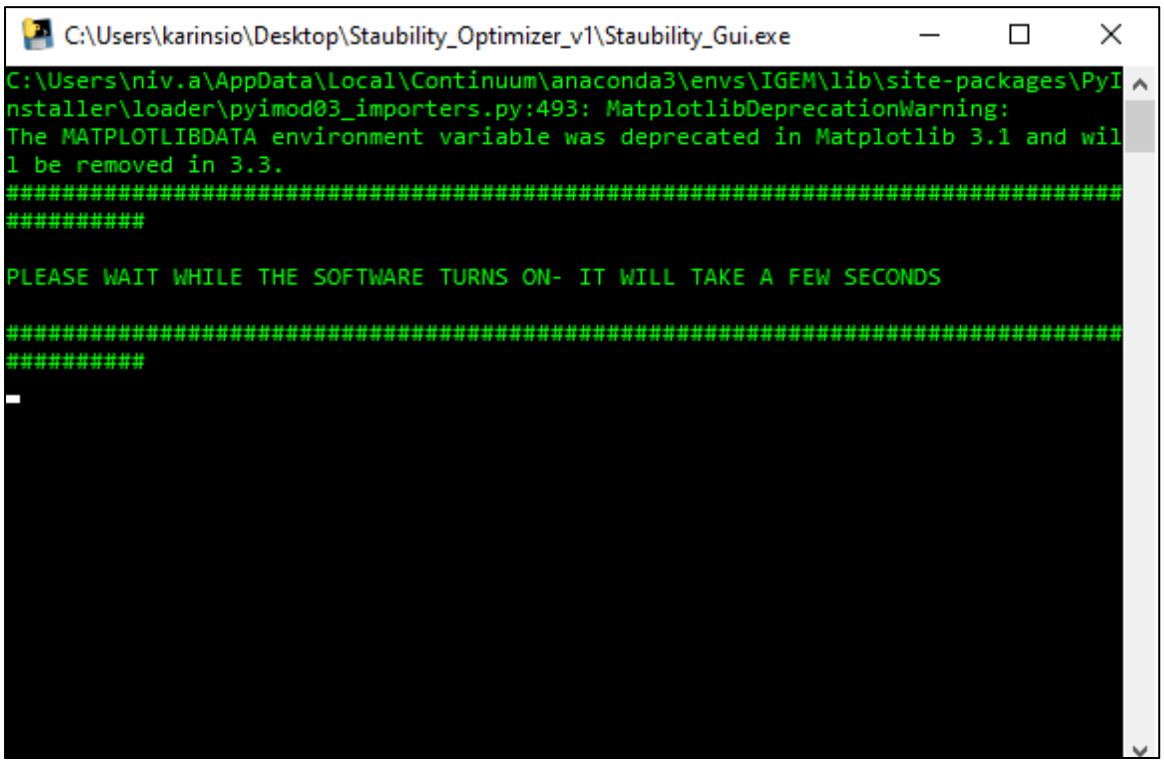
Installation instructions – Beta Version

Our beta version is available in the “Staubility_Enhancer_v1" folder. The folder includes the actual software in the "GUI.exe" file, another subfolder called "In", and this 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.

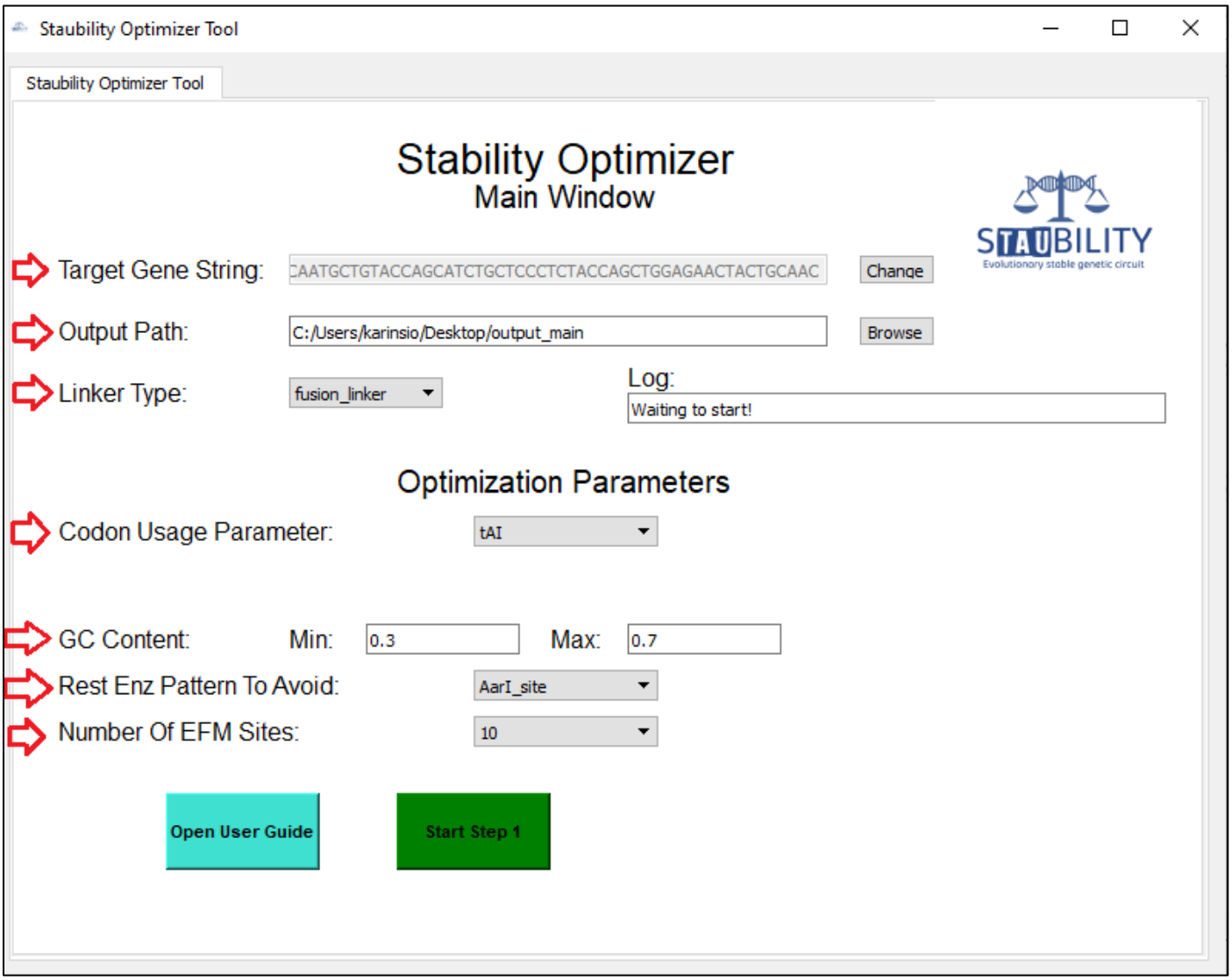
To launch the software:

1. After downloading the “Staubility_Enhancer_v1" folder with all its content, run the "GUI.exe" file **from the same folder.**
2. A CMD window should open. This is a black window, where notes appear while the software’s processes continue, allowing you to track the generation progress of the optimal construct.



3. The software's main window should launch a few moments after the CMD opens.

Main Window



Target Gene String:

In this field, you insert your target gene as a string. Note, **the string length must be divisible by three for division to codons, and must be composed of the letters ATGC.** If a sequence is legal, after pressing the “insert” button, it will change to upper case and turn gray to indicate that it is locked in. From this point on, you are required to press “change” in order to replace it. In case the sequence is not legal, pressing the "insert" button will raise an error. The software cannot proceed until the sequence is fixed.

Note: The stop codon of the target gene needs to be eliminated in order to transcriptionally interlock it to an essential gene. If you provide a target gene's sequence that includes a stop codon, our software will delete it automatically. Accordingly, the output optimized sequence of the target gene will not have a stop codon.

Output Path:

In this field, you define in which directory to output the software’s results. For further reading about the content of each output file, see the Output folder content section

Linker Type:

In this drop-down menu, you can select whether you prefer to use a fusion linker or 2A, and receive linker recommendations accordingly.

Log:

In this field, the software's progress is displayed. It is a concise description; additional information can be seen in the CMD window at each step of the process.

Codon Usage Parameter:

There are several possible objectives for codon usage optimization, which can be selected using this drop-down menu. They are as follows:

1. "Fraction" – this objective corresponds to codon relative frequency matching between the construct and the host genome. When selected, an additional drop-down menu of codon usage optimization methods becomes available (see description below).
2. "tAI" – this objective corresponds to the **tRNA Adaptation Index**, which is a biophysical measure of codon usage bias. This measure assumes that tRNA types which recognize the codons in highly expressed constructs are more abundant and available. Thus, codons with available tRNA are considered more optimal, as more codon-recognizing tRNAs in the cell lead to faster translation elongation.
3. "nTE" – this objective corresponds to the **normalized Translation Efficiency**, which is an improved version of tAI. It considers not only the supply of tRNA, but also the demand, i.e. the number of codons and mRNA copies in the genome which are translated by the same tRNA.
   An optimal codon is characterized by a large supply/demand ratio.
4. "TDR" – this objective corresponds to the **typical decoding rate.** For each codon, this measure is defined as its empirical decoding time's reciprocal.
   The codon empirical decoding time is an estimation of how long a ribosome typically stays on each of the 61 codons during translation based on ribo-seq experiments. The longer the ribosome stays, the slower the translation. This measure is similar to the previously described optimality measures, only it is based on direct experimental measurements. For the calculation of TDR, temporal statistics for each codon are collected from all transcripts. Afterwards, the codon temporal distribution is calculated, and the expected value extracted. This value is estimated as the typical empirical decoding time of the codon, and is the reciprocal of the translation rate.

Codon Usage Optimization Method:

After selecting "fraction" as the codon usage objective, three common codon optimization methods are offered:

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:

In these two fields, you can limit the frequency of GC nucleotides in the optimized sequence within a specified range. This is important for various biological purposes, including maintaining a high expression level and genomic stability.

Rest Enz Pattern to Avoid:

In many biological processes, restriction enzymes are used. In this case, it is imperative that a restriction site will not appear in unexpected locations. In this drop-down menu, you may select a restriction enzyme you are using, and respective restriction sites are avoided.

Number of EFM sites:

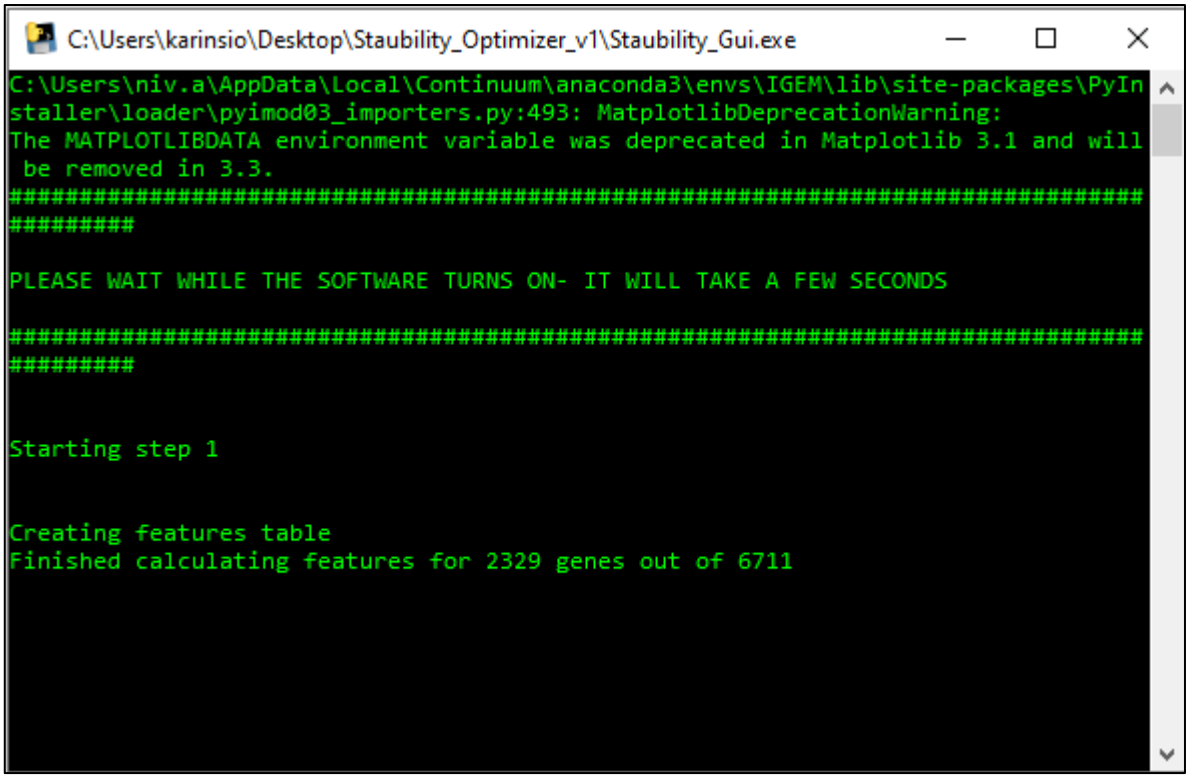
EFM stands for Evolutionary Failure Mode. Our EFM model finds the most unstable mutational hotspots of the types “Simple Sequence Repeats” and “Recombination Mediated Deletions”. Our optimization process can avoid these sites to increase stability. In this drop-down menu, you may select how many sites of each type to avoid. This parameter defines a **trade-off between stability and expression** – the more hotspots considered, the less degrees of freedom are reserved for optimizing expression.

Open User Guide:

This button opens the user guide you are currently reading.

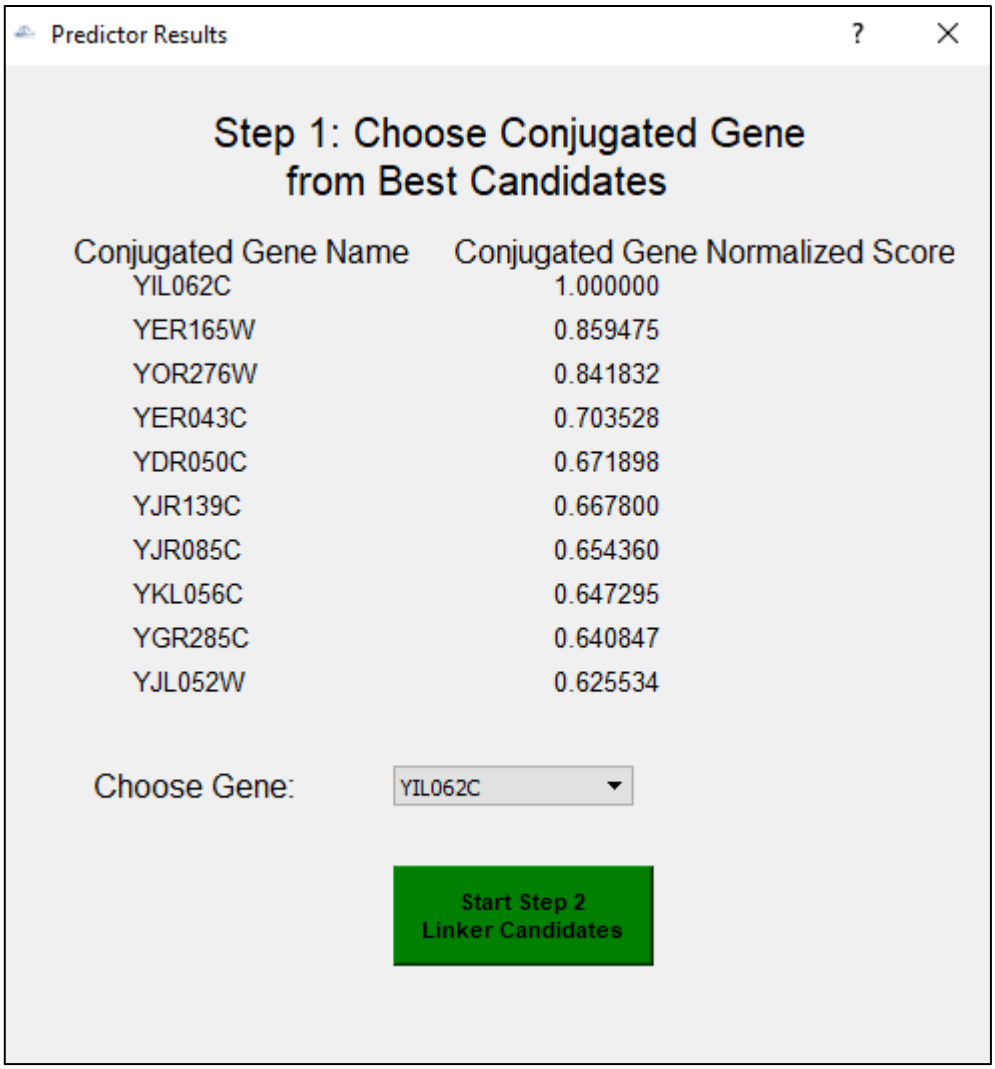
Start Step 1:

This button initiates the conjugated gene selection process. This may take a few moments, but you can view the progress in the CMD window previously opened:



Step 1: Choose Conjugated Gene from Best Candidates

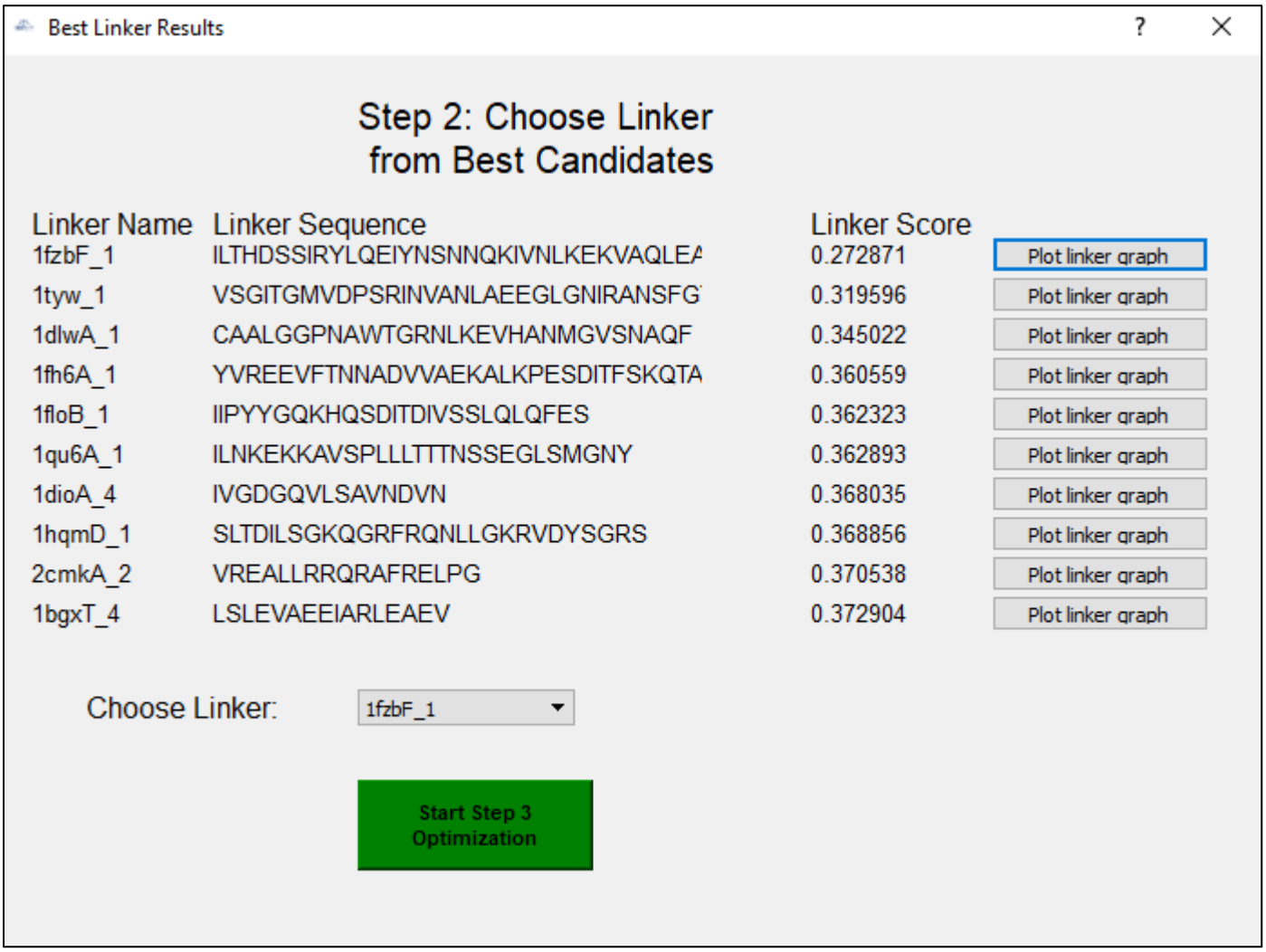
At this stage, our stability predictor has calculated the expected stability for each conjugated gene and normalized it according the top candidate. The top ten candidates are displayed, as in many biological experiments, you may want to try out several constructs for best results. This way, you can choose different genes and get different optional constructs.



To select a conjugated gene, select it from the drop-down menu titled “Choose Gene”. Click the “Start Step 2” button to initiate the linker selection process.

Step 2: Choose Linker from Best Candidates

At this stage, our linker model has calculated the score for each linker. This score represents the folding change that is caused to the conjugated protein and the target protein, following the fusion process. **A low score indicates that the protein's folding was less affected by the fusion, thus it is preferable**. You can choose to see the relative disorder profiles of the proteins before and after the fusion as a graph, which will indicate the linker's quality.

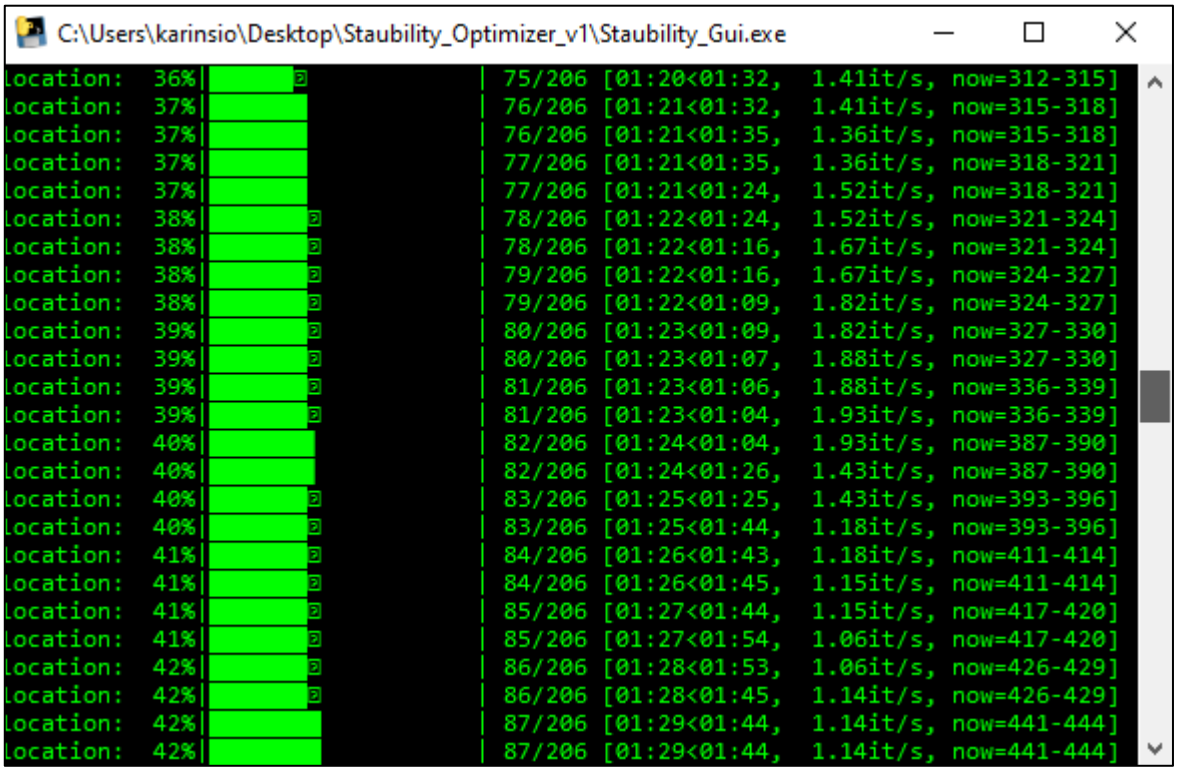


The top ten candidates are displayed, along with their amino acid sequence. Once again, you may want to try out several constructs for best results. This way you can select different linkers and get different optional constructs.

To choose a linker, select it from the drop-down menu titled “Choose Linker”. Click the “Start Step 3” button to initiate the optimization process.

Step 3: Optimization

In this final stage, the optimization process previously described is implemented. **This can take several moments**, but you can track the progress in the CMD window.



The combined construct is optimized for:

1. avoidance of EFM sites
2. weak mRNA folding at the start of the target gene (first 15 codons)
3. GC content
4. codon usage

This process provides you with a sequence optimized for expression and stability. The output results will be described in the next section.

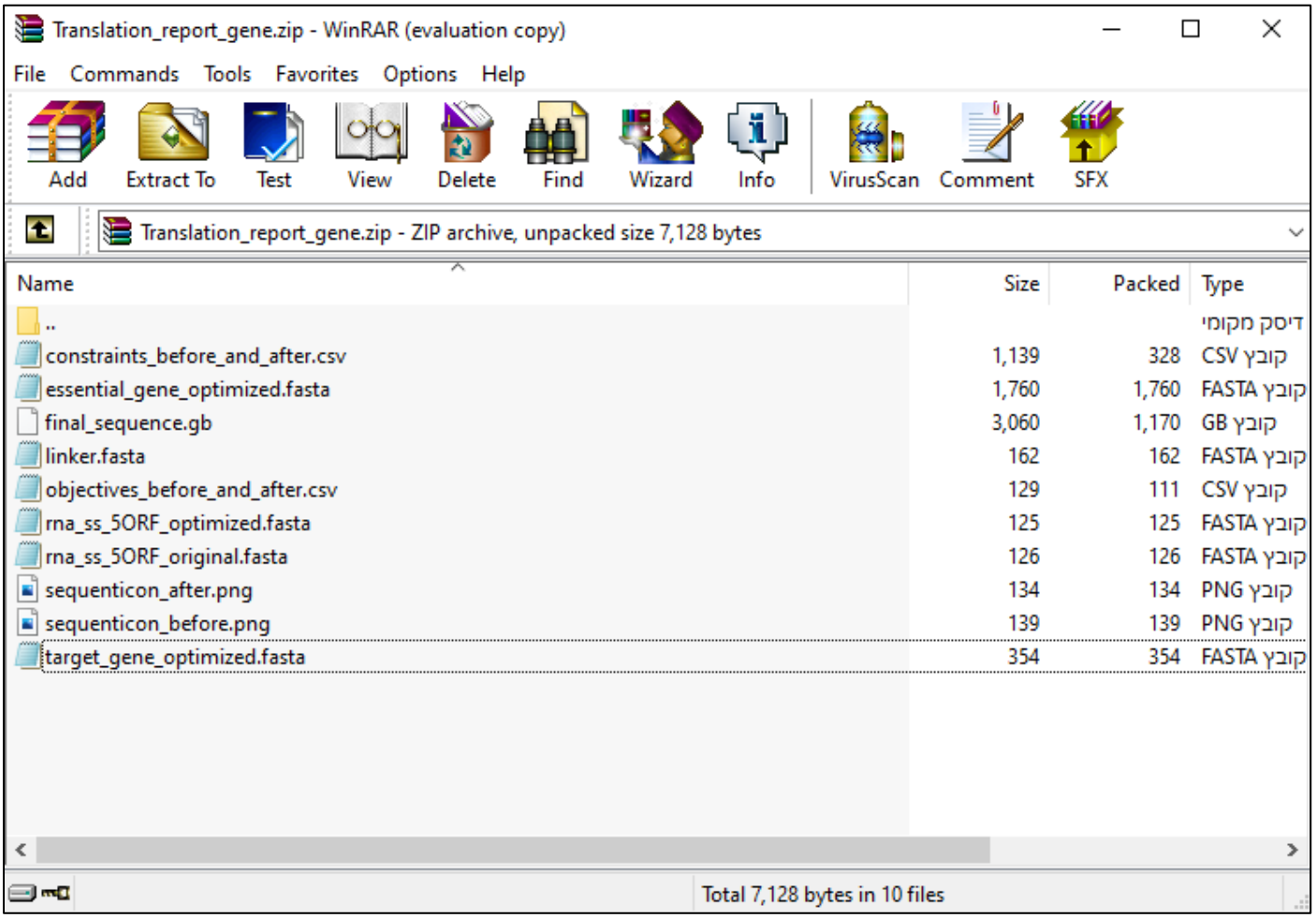
Output folder content

The output folder should include several files at the end of the process:

1. Disorder profile PNG files, displaying graphs of the target and conjugated proteins' disorder profiles. These graphs are also available within the software itself.
2. Translation report – a zip file which contains a summary of the optimization steps, and the final optimized sequences: the target gene, the conjugated gene, and the linker. Below is a detailed description of each file within the zip.

Translation report – zip file

Each zip file contains the following files:



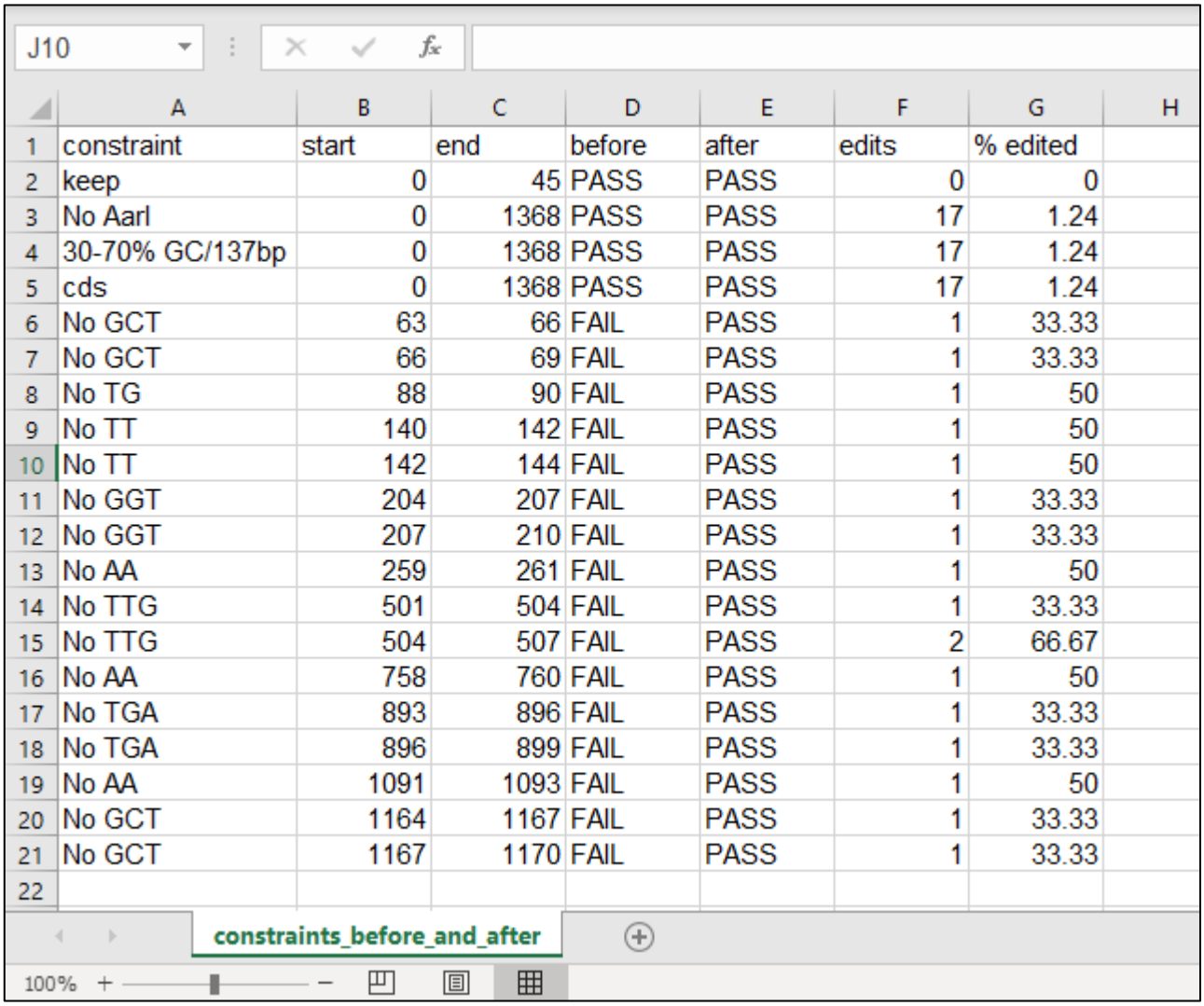
Files description:

1. **Objective_before_and_after.csv –**
   Reports the objective score, in csv format. For example:

	A	B	C	D	E	F	G	H	I
1	objective	boost	start	end	before	after	edits	% edited	
2	best-codon-optimize (s_cerevisiae)	1	45	1368	-11.1	-24.69	17	1.28	
3									
4									
5									

The objective function in this example was to optimize the codon usage based on S. Cerevisiae as the host organism. Note that the start position is 45 because during the optimization process, the first 15 codons (45 nucleotides) are optimized for weak folding and mRNA, and not for codon usage.

2. **Contraints_before_and_after.csv** –
Contains the objective and the constraints that were successfully maintained, in csv format. For example,



	constraint	start	end	before	after	edits	% edited
2	keep	0	45	PASS	PASS	0	0
3	No AarI	0	1368	PASS	PASS	17	1.24
4	30-70% GC/137bp	0	1368	PASS	PASS	17	1.24
5	cds	0	1368	PASS	PASS	17	1.24
6	No GCT	63	66	FAIL	PASS	1	33.33
7	No GCT	66	69	FAIL	PASS	1	33.33
8	No TG	88	90	FAIL	PASS	1	50
9	No TT	140	142	FAIL	PASS	1	50
10	No TT	142	144	FAIL	PASS	1	50
11	No GGT	204	207	FAIL	PASS	1	33.33
12	No GGT	207	210	FAIL	PASS	1	33.33
13	No AA	259	261	FAIL	PASS	1	50
14	No TTG	501	504	FAIL	PASS	1	33.33
15	No TTG	504	507	FAIL	PASS	2	66.67
16	No AA	758	760	FAIL	PASS	1	50
17	No TGA	893	896	FAIL	PASS	1	33.33
18	No TGA	896	899	FAIL	PASS	1	33.33
19	No AA	1091	1093	FAIL	PASS	1	50
20	No GCT	1164	1167	FAIL	PASS	1	33.33
21	No GCT	1167	1170	FAIL	PASS	1	33.33

In this example:

- The first constraint is to keep the 15 codons of the target gene, following their optimization for mRNA weak folding. The "start" and "end" column specify the python-wise indices[1] within the sequence that should maintain this constraint.
- The second constraint is "no AarI" which is the restriction site to avoid. The "before" and "after" contains the constraint status before and after optimization. The sequence did not include the AarI restriction site before optimization (PASS) and it was kept that way.
- The third constraint is "CDS", which is a short for Coding Sequence, meaning the amino-acid translation was preserved.
- the fourth constraint is to keep the GC content in the range of 30%-70% in windows of 137 base-pairs.

[1] In python, the first index is zero, the second index is one, and so forth. When one specifies a range of 0-45, it means that he does not take the 45th position; thus, the sequence is of length 45.

- From the fifth constraint you may see the mutational hotspots which 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 construct (composed of target gene, linker and conjugated gene) in GenBank format, not annotated.

4. **Essential_gene_optimized.fasta, linker.fasta, target_gene_optimized.fasta –**
   The optimized components of the final constructs in FASTA format.

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.

7. **Rna_ss_5ORF_original.fasta –**
   The first 15 codons of the target gene with the mRNA structure and the local minimum folding energy, before optimization, in FASTA format. Using this file, you may compare the original structure to the optimized one.

8. **Rna_ss_5ORF_optimized.fasta** –
   The first 15 codons of the target gene with the mRNA structure and the local minimum folding energy, after optimization, in FASTA format.

In some computers, the software might 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