BCB410 Assessment 3

Yunyi Cheng

Outline of the R package I would like to develop

1. What is the objective(s) of the tool you plan to develop? What does it improve?

The objective of the tool is to facilitate the seamless design of the mutagenizing oligonucleotides required for Tile Region Exchange Mutagenesis (T-Rex), which is a crucial step for deep mutation scanning (DMS), by harnessing data-optimized assembly design (DAD). It improves both the accuracy and efficiency of designing genic tiles.

The following paragraphs provide extra information about deep mutational scanning, Tile Region Exchange Mutagenesis and data-optimized assembly design for a better understanding of the objectives.

DMS is a quantitative framework used to analyze the effects of genetic variants on protein function simultaneously. The typical pipeline of a DMS study is the systematic and saturated mutagenesis of a gene, followed by a selection assay modelled on a specific phenotype to determine the functional impact of each variant. This is coupled with high-throughput DNA sequencing to functionally score variant effects, which are then typically superimposed onto a heat map. To facilitate high-throughput production of gene variants, we need to perform T-Rex based on Golden Gate assembly, which allows simultaneous assembly of multiple DNA fragments. It uses a Type IIS restriction enzyme to cleave near enzyme recognition sequences on destination vector and gene cassettes containing desired mutations, yielding digested destination vector and mutagenic cassettes, all flanked by 4 bp overhangs. Then, DNA fragments are joined by DNA ligase. Golden Gate assembly uses Type IIS restriction endonucleases which cut DNA outside their recognition site, providing freedom to customize overhang sequences and enabling seamless assembly in a single experiment. Correctly assembled final products do not contain enzyme recognition sites; thus, the reaction is irreversible by nature. One of the challenges of Golden Gate assembly is determining the final construct of DNA segments, which depends on overhang sequences generated by Type IIS enzyme digestion. Choosing appropriate overhang sequences is vital for Golden Gate assembly, as poorly designed overhangs are prone to undesired ligation of non-complimentary overhangs. To maximize the success rate of assembly, we need to follow general guidelines of design:

1. Each cassette must lie between codon boundaries, and cassettes must be adjacent, ensuring the effective mutations of amino acids.
2. Palindromic overhangs and duplicate overhangs are forbidden to avoid incorrect ligations.
3. The length of oligonucleotides should be limited to approximately 120 bp since the fidelity of oligonucleotides decreases exponentially as the length increases.

According to previous studies, outcomes of Golden Gate assemblies do not comply with sheer rules based on observed assembly events. Thus, designing overhangs by hand is error-prone and time-consuming. To tackle this challenge, I will develop an *in silco* tool to streamline the procedure. Each overhang has been tested for reactivity against all other overhangs so we can identify overhang sets expected to have high fidelity with the help of bioinformatic analysis. DAD also makes large assemblies possible: using DAD tools, the fidelity starts to decrease only when the number of overhangs reaches a certain threshold, which adds a great advantage, especially when mutagenizing large genes.

Briefly, the tool begins by assigning a score to tile locations. This score is a function of the tile sizes, whether tiles at particular positions would generate palindromic or repeating overhangs after BsaI restriction digest, and ligation fidelity as determined by New England Biolabs. In the study by New England Biolabs, the authors analyzed the propensity for a specific 4 bp overhang to ligate to all other possible 4 bp overhangs. The fraction of reads observed with correct or incorrect ligations is given in the study and forms the basis of our ligation fidelity score. Ligation fidelity is calculated by the On-Target ligation minus the Off-Target ligations to all other overhangs produced by the current tile locations. Tile locations with more desirable qualities (such as high-fidelity overhangs) are given a higher score.

The optimal positions for the tile boundaries cannot be easily obtained. This is because modifying the boundaries of a single tile will strongly affect the optimality of other tiles (due to the ligation score and the tile size score). I believe that this problem reduces to the well-known clique problem in computational complexity and is therefore NP-complete (no efficient solution can be found). However, finding the optimal tile positions is not compulsory; it is adequate to find decent solutions for assembly design. Thus, I will develop a gradient descent algorithm that randomly ‘wiggles’ tile positions, accepting solutions only if they increase the global score.

2. ## What type(s) of biological data will be analyzed using your tool? Clearly explain scientific jargon. Do not use undefined acronyms.

The tool takes the target gene as a string containing the open reading frame (ORF) of a gene, as well as two codons before and after the ORF for overhang extraction. The gene could be hardcoded as a string or imported from a FASTA file that contains multiple strings of interest for high-throughput design. It also takes an overhang fidelity chart from the study by New England Biolabs and an oligonucleotide cost chart (both are Excel files) to determine the gene tile design that is the most budget-friendly and has the highest assembly success rate.

3. ## Are there similar packages that have already been developed? Do a search. Provide a list and some details of similar packages. What aspect of data analysis does your R package permit to perform, that has not been done to date?

After a thorough search, there are no similar packages in R. But there are similar web tools such as NEBridge ® Golden Gate Assembly Tool and Golden Gate Assembly Wizard. They all automate sticky end design and display the final construct of the gene plasmid. On top of these functionalities, my package is capable of performing DAD that optimizes the design further to reduce off-reactions during assembly and to limit the cost at a reasonable level. Furthermore, my package will enable visualization during

optimization.

4. ## Who is the targeted audience of your package? Explain.

   The targeted audience of my package would be researchers who perform Tile Region Exchange Mutagenesis for some genes of interest and need an efficient and reliable streamline to produce mutagenizing oligonucleotides since designing the assembly by hand would be time-consuming and error-prone.

5. ## What are some of the assumptions of your package? Explain. If you are unclear, see Assessment 2, section 2. j.

   1. Users are using BsaI restriction enzymes which create 4 bp overhangs.
   2. The target gene sequence must be provided in a string (hardcoded or imported from a FASTA file), containing only the open reading frame and one codon before and after the open reading frame.

6. ## Approximately how many functions would be available to user? Explain the purpose of these functions.

   These functions will be available to users:

   1. getOverhangs: Get the pair of overhangs that are generated by the given tile.
   2. obtainScore: Obtain the score for one tile position based on on-target reactivity, length, and palindromicity.
   3. getAllOverhangs: Get all overhang sequences given a list of positions.
   4. caculateScores: Obtain the global score for the complete list of positions based on the sum of local scores, off-target reactivity, and duplication.
   5. optimizePosition: For a specific list of positions, optimize a single position by scanning for positions among neighbourhoods, according to the selected mode.
   6. iterPlot: Generate plots of change of score during the optimization process.

7. ## What type of a visual output might help perform the exploratory analysis of your package? Explain. How would you convert this idea into a function for visualization, as part of your package?

   A graph illustrating the optimization process (the change in the overall score for the design as optimization takes place) would help perform the exploratory analysis of my package. I would include this idea into a function in my package called iterPlot to perform real-time visualization of my optimization tool.

8. **Of the total number of functions in your package, how many will be dedicated to visualization?**

   Two will be dedicated to visualization. One is iterPlot and the other one would be the helper for iterPlot. The number of helper functions might increase as implementation takes place.

9. **For help documentation of your R package, you are required to include examples that show utility of functions using data. Will you be providing your own dataset? If yes, explain the format of the data. Otherwise, you will need to use data from another package. In that case, which package would your data come from? You are welcome to use both cases as well.**

   I will be providing my own dataset. It will likely be a FASTA file containing a list of genes of interest.

10. **What other packages would your package depend on? List these and explain their purpose and why they are used in your package.**

    Currently, my package will depend on the following other packages (might add new ones along with implementation):

    1. Biostring: for manipulation of targeted genes
    2. ggplot2: for visualization
    3. dplyr: for data frame operations
    4. reticulate: R interface to Python modules, classes and functions for better optimization

11. **Are you planning to use Generative Artificial Intelligence (AI) tools for developing your R package? If yes, explain for what aspects of the package this will be used for and why it is necessary to use generative AI. If not, explain your position.**

    I am not planning to use Generative AI tools for developing my R package since I will be capable of testing and debugging without its help. Moreover, Generative AI is highly dependent on the data fed to it so it is prone to errors.

12. **How can you ensure equity, diversity inclusion (EDI) practices are respected in your work on R package? Explain. What are some challenges or what maybe beyond your control? Explain.**

    First, I will not use/define data, variables and functions that do not ensure EDI. I will comment on my code carefully and include an informative README for the GitHub repository of the package to ensure that it is open-source and easy to use for anyone including people from underrepresented groups. However, there are challenges beyond my control. I am only developing the tool, but the way of using it and the dataset fed to it are also important aspects of ensuring EDI. For example, if the dataset fed to the R package is biased, the result would be biased as well without users knowing it.

13. **How can you ensure reproducibility in your work on R package? Explain.**

    I will make it open-source on GitHub with the necessary files and an informative README so that it can be reproduced by any interested researchers who want to utilize this library.

14. **What is the name of your R package? Describe the name you selected.**

    My R package would be named TRexDAD for it helps Tile Region Exchange Mutagenesis by harnessing data-optimized assembly design.

15. **In your GitHub account, you must use your full name as provided in the course. This can be checked/changed by going to Settings → Public profile → Name. E.g., Kevin Smith should not use “Kev S" as the GitHub name. “Kevin Smith" must be used, if that is the name provided in the course. State that you have ensured this step is complete and your full name as provided in BCB410 is visible on GitHub.**

    I have ensured this step is complete and my full name as provided in BCB410 is visible on GitHub.

16. **Create a GitHub repository with the package name. Provide the name and link. E.g., Name of the package is mixGaussian. GitHub link: https://github.com/anjalisilva/mixGaussian. Note: Your project must be housed in this repository.**

    Name of the package: TRexDAD

    GitHub link: https://github.com/yunyicheng/TRexDAD

References

1. Pryor JM, Potapov V, Kucera RB, Bilotti K, Cantor EJ, et al. (2020) Enabling one-pot Golden Gate assemblies of unprecedented complexity using data-optimized assembly design. PLOS ONE 15(9): e0238592. https://doi.org/10.1371/journal.pone.0238592