Leveraging Multiprocessing for Thermodynamic Optimization and Generation of Orthogonal DNA Libraries

Jean-Sebastien Paul April 14 2025

To Cover

- **Motivation**
- Nomenclature & Notes
- Overall Steps
- Symmetry Minimization & Character Similarity
- On-Target Optimization
- Off-Target Optimization
- Melting Temperature Optimization
- Conclusions

To preface: we will cover picked orthogonality criteria which is a bit dry. However for the DNA group as a whole, the multiprocessing aspect of calculating general thermodynamic values might be of the greatest interest as a research tool.

For the sake of simplicity I will only cover Duplex library generation here although single stranded libraries can be generated too.

Motivation

- Goal: to generate an orthogonal DNA duplex library.
  - Composed of sequences which when put together in a test tube only bind to their reverse complement if the reverse complement is available.
  - The sequences should not bind to other sequences, reverse complements or not, in the library.
- Useful across domains from DNA assembly, DNA strand displacement, HCR probe design and more
- Most approaches try to use predominantly string based methods like Sequence Symmetry Minimization.
  - Thermodynamic calculations are costly
- We propose using string based methods to generate an initial library which is optimized via thermodynamic calculations.
- Critically, we make thermodynamic calculations multiprocessed leveraging multiple cores on a CPU for speed.

To Cover

- Motivation
- **Nomenclature & Notes**
- Overall Steps
- Symmetry Minimization & Character Similarity
- On-Target Optimization
- Off-Target Optimization
- Melting Temperature Optimization
- Conclusions

Nomenclature

- We will write the following to represent expected concentration:
  - $E([(a,b)]\,|\,I_c, T, \Theta, \Psi)$
    - Where (a,b) is some complex composed of strands a and b.
    - I_c represents the set of initial concentrations for starting species.
    - T represents the temperature.
    - Θ represents other user-adjustable salt conditions when carrying out thermodynamic predictions (Na^+ & Mg^{2+} concentration).
    - Ψ represents the considered complexes possible from the initial concentrations I_c.
- Complexes can be represented by (a,b) for duplexes, or simply (a) for a single stranded system (given strands a and b). We will not consider complexes in size above a duplex for the sake of simplicity.
- Strand $\tilde{a}$ is the reverse complement of strand a.

Nomenclature

- A user will define:
  - Length of sequences l
  - Symmetry minimization criterion k
  - Some working concentration c
  - Some working temperature τ
  - Some standard salt conditions θ
  - Some thresholds $\eta_{SIM}, \eta_{ON}, \eta_{OFF}, \delta$ for various optimization steps
  - Γ which specifies the temperature space to explore in melting temperature optimization

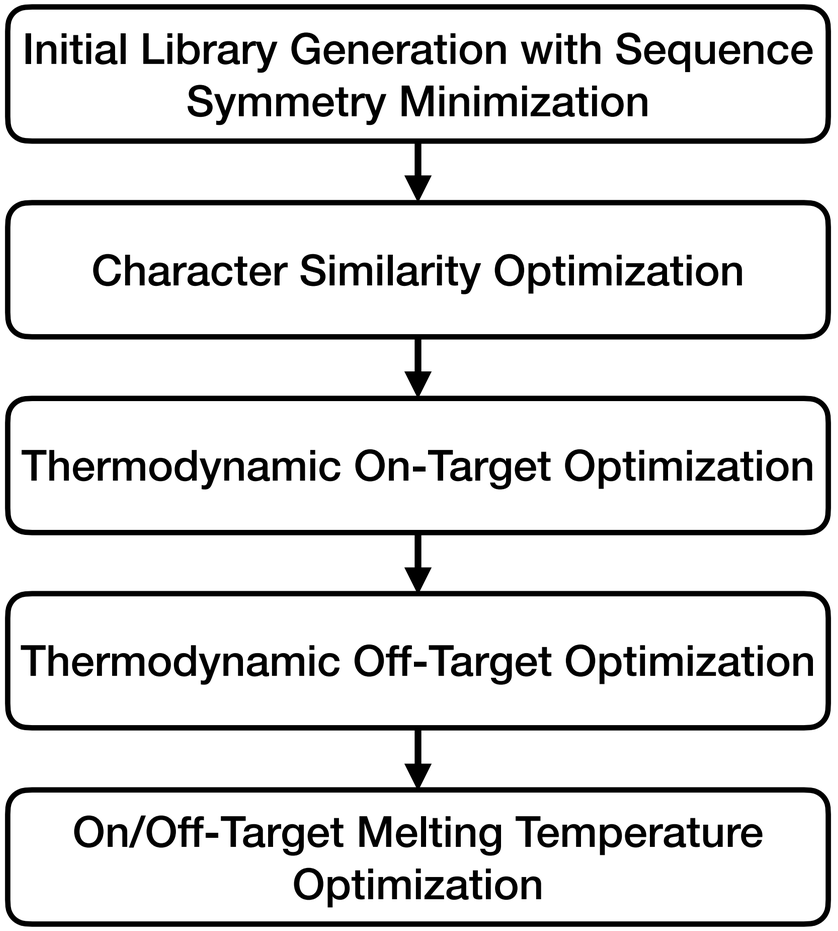
Note on runtimes

- For the sake of consistency and fair comparison, all runtimes reported here come from runs on a single computer (the Caltech Undergraduate Computer Science Club server).
  - CPU: 2023 14th generation Intel Core i9-14900 Raptor Lake-S Refresh 2.0GHz **24 Core** LGA 1700
- Note that this computer context switches between my tasks and the many other tasks of other students.

To Cover

- Motivation
- Nomenclature & Notes
- **Overall Steps**
- Symmetry Minimization & Character Similarity
- On-Target Optimization
- Off-Target Optimization
- Melting Temperature Optimization
- Conclusions

Overall Flowchart of Steps



To Cover

- Motivation
- Nomenclature & Notes
- Overall Steps
- **Symmetry Minimization & Character Similarity**
- On-Target Optimization
- Off-Target Optimization
- Melting Temperature Optimization
- Conclusions

Initial Library Processing

Sequence Symmetry Minimization & Character Similarity

- Sequence symmetry minimization defines a set of sequences where:
  - Each sequence is of length l
  - Each sequence shares no subsequence of length $k \leq l$ with any other sequence in the set
- We can generate the max size set (which also considers reverse complements of the sequences generated).
- We will call the current library of sequences L
  - In our notation, L will not contain reverse complements of its sequences which are instead contained in a separate set $\tilde{L}$.

Initial Library Processing

Sequence Symmetry Minimization & Character Similarity

- We then ensure character similarity between sequences is less than some threshold η_{SIM}
- Let $SIM(x, y)$ be a function that counts the number of characters in the same position between 2 strings x, y.
- Go through each $a, b \in L$ where $a \neq b$
  - Define $H_{ab} = H_{ba} = \max(SIM(a, b), SIM(\tilde{a}, b), SIM(a, \tilde{b}), SIM(\tilde{a}, \tilde{b}))$
- $H_{aa} = 0, \forall a \in L$
- Then, while $\max(H) > \eta_{SIM}$:
  - Given $a, b \in L$ where $H_{ab} = \max(H)$
  - Remove a if $\max(H_a - H_{ab}) > \max(H_b - H_{ab})$
  - Remove b if $\max(H_a - H_{ab}) < \max(H_b - H_{ab})$
  - Remove one of a, b at random

To Cover

- Motivation
- Nomenclature & Notes
- Overall Steps
- Symmetry Minimization & Character Similarity
- **On-Target Optimization**
- Off-Target Optimization
- Melting Temperature Optimization
- Conclusions

On-Target Thermodynamic Optimization

- Given a present library L whose size is denoted n, for each $a \in L$:
  - $\iota_c = \{[a] = c, [\tilde{a}] = c\}$
  - $\psi = \{(a), (\tilde{a}), (a, a), (\tilde{a}, \tilde{a}), (a, \tilde{a})\}$
  - Calculate $p_a = \dfrac{E([(a, \tilde{a})] \,|\, I_c = \iota_c, T = \tau, \Theta = \theta, \Psi = \psi)}{c}$
- We can treat each p_a as a separate calculation and can compute each independently with multiprocessing.
- We drop any a from the library L if $p_a < \eta_{ON}$ (the user defined On-Target threshold)

To Cover

- Motivation
- Nomenclature & Notes
- Overall Steps
- Symmetry Minimization & Character Similarity
- On-Target Optimization
- **Off-Target Optimization**
- Melting Temperature Optimization
- Conclusions

Off-Target Thermodynamic Optimization

- Given a present library L of size n, for each $a, b \in L$ where $a \neq b$:
  - For each $i \in \{a, \tilde{a}\}, j \in \{b, \tilde{b}\}$
    - $\iota_c = \{[i] = c, [j] = c\}$
    - $\psi = \{(i), (i, i), (j), (j, j), (i, j)\}$
  - Calculate $r_{ij} = \dfrac{E([(i, j)] \,|\, I_c = \iota_c, T = \tau, \Theta, \Psi = \psi)}{c}$
  - Then let $\omega_{ab} = \omega_{ba} = \max(r_{ij})$

- Given a present library L of size n, for each $a \in L$
  - For each $i \in \{a, \tilde{a}\}$
    - $\iota_c = \{[i] = 2c\}$
    - $\psi = \{(i), (i, i)\}$
  - Calculate $r_i = \dfrac{E([(i, i)] \,|\, I_c = \iota_c, T = \tau, \Theta, \Psi = \psi)}{c}$
  - Then let $\omega_{aa} = \max(r_i)$

- Once again, we can treat each ω entry as a separate calculation and can compute each entry independently with multiprocessing.

Off-Target Thermodynamic Optimization

- Once ω is generated, first go through each $a \in L$ and remove a from L and ω if $\omega_{aa} > \eta_{OFF}$ (the user defined Off-Target threshold)
- Then go through each $a, b \in L$ where $\omega_{ab} > \eta_{OFF}$
  - Remove a from L and ω, if $\sum_{i=1}^{n} \omega_{ai} > \sum_{i=1}^{n} \omega_{bi}$
  - Remove b from L and ω, if $\sum_{i=1}^{n} \omega_{ai} < \sum_{i=1}^{n} \omega_{bi}$
  - Else remove one of a, b from L and ω at random

On and Off Target Calculations are in Parallel

- We calculate p and ω together at the same time in a multiprocessed way.
  - Runtime:
    - 14mins 13sec for 3800 sequences including reverse complements to analyze (24 core)
    - 2hrs 42mins 30sec for 3794 sequences including reverse complements to analyze (1 core)
    - ~12x faster

To Cover

- Motivation
- Nomenclature & Notes
- Overall Steps
- Symmetry Minimization & Character Similarity
- On-Target Optimization
- Off-Target Optimization
- **Melting Temperature Optimization**
- Conclusions

Melting Temperature Optimization

Defining melting temperature

- Given a present library L of size n, for each $a, b \in L \cup \tilde{L}$:
  - $\iota_c = \{[a] = c, [b] = c\}$
  - $\psi = \{(a), (b), (a, b)\}$
  - Γ is a user defined set of temperatures to explore
  - Calculate $\mu_{ab} = \mu_{ba} = \underset{t \in \Gamma}{\operatorname{argmin}} \left| \frac{E([(a, b)] \,|\, I_c = \iota_c, T = t, \Theta = \theta, \Psi = \psi)}{c} - 0.5 \right|$
- We can use binary search across Γ to find μ_{ab} reducing the number of calculations
- Each μ_{ab} can be calculated independently over multiple processes.

Melting Temperature Optimization

Defining melting temperature

- Let $\lambda = \{\mu_{i,j} : i = \tilde{j}, i \in L, j \in L\}$
- Then for each $a, b \in L$
  - $m_{aa} = \max(\mu_{aa}, \mu_{\tilde{a}\tilde{a}})$
- For each $a, b \in L$
  - $m_{ab} = \max(\mu_{ab}, \mu_{a\tilde{b}}, \mu_{\tilde{a}b}, \mu_{\tilde{a}\tilde{b}})$
- Algorithm (defined next slide) will remove sequences from L (and correspondingly μ, m, λ) such that $\max(m) < \min(\lambda) - \delta$

Melting Temperature Optimization

Defining melting temperature

- Strategy 1:
  - Go through all a where $m_{aa} > \min(\lambda) - \delta$
    - Remove a from L, m, λ
  - Go through all a, b where $m_{ab} > \min(\lambda) - \delta$
    - Remove a from L, m, λ, if $\sum_{i=1}^{n} m_{ai} > \sum_{i=1}^{n} m_{bi}$
    - Remove b from L, m, λ, if $\sum_{i=1}^{n} m_{ai} < \sum_{i=1}^{n} m_{bi}$
    - Else remove one of a, b from L, m, λ at random
- Strategy 2:
  - Remove $a = \underset{i \in L}{\operatorname{argmin}} |\lambda_i|$ from L, m, λ
  - Repeat Strategy 1
- Carry out Strategy 1 and Strategy 2 independently.
  - If $|L|$ after strategy 1 is larger than or equal to after strategy 2, go with strategy 1.
  - If $|L|$ after strategy 1 is smaller than after strategy 2, repeat this full process again.

Melting Temperature

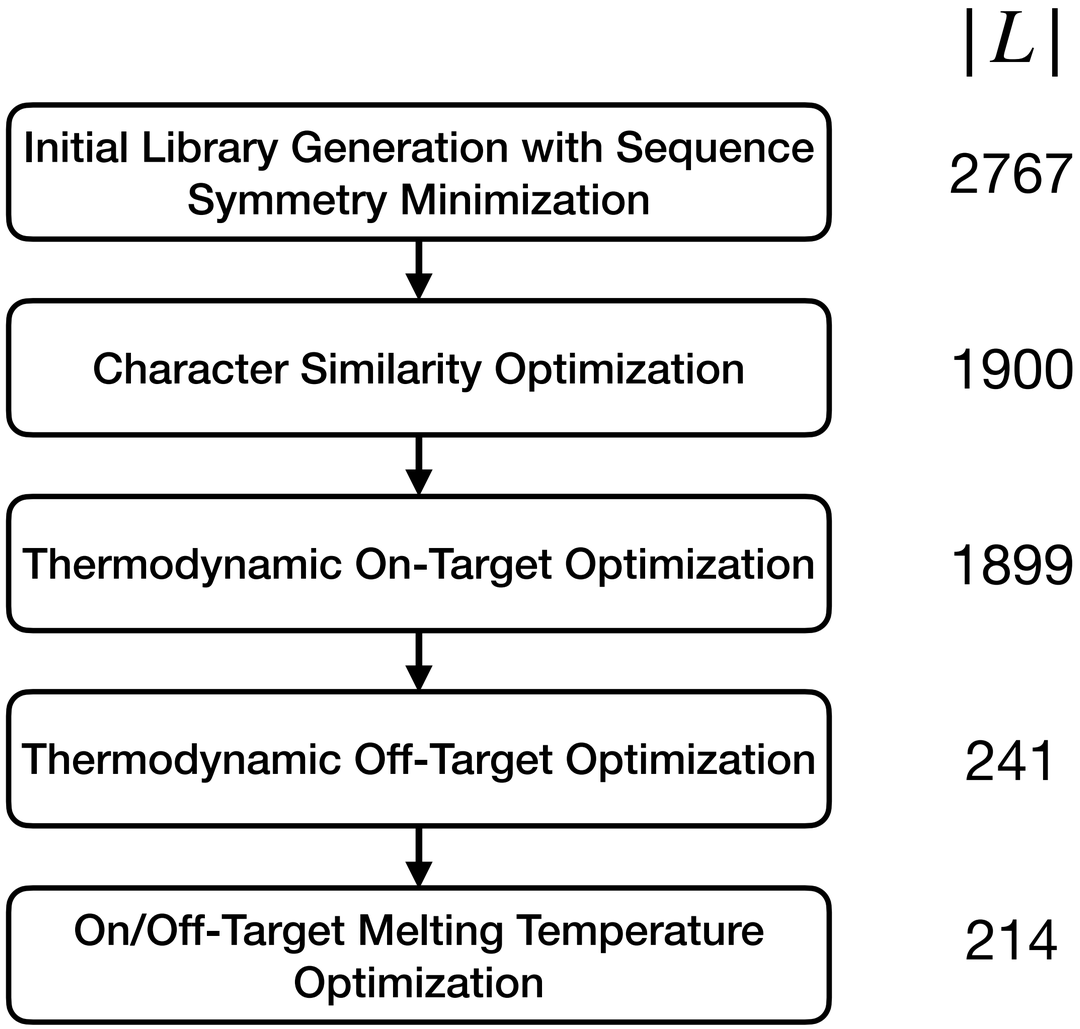
- Most intensive function.
- Runtime
  - 7mins 33sec for 482 sequences including reverse complements to analyze (24 core)
  - 1hr 49min 32sec for 482 sequences including reverse complements to analyze (1 core)
  - ~16x faster

To Cover

- Motivation
- Nomenclature & Notes
- Overall Steps
- Symmetry Minimization & Character Similarity
- On-Target Optimization
- Off-Target Optimization
- Melting Temperature Optimization
- **Conclusions**

Advantages of This Approach: Speed

- An example run with
  - Length of sequences $l = 16$
  - Symmetry minimization criterion $k = 8$
  - Some working concentration $c = 1\mu M$
  - Some working temperature $\tau = 20^oC$
  - Some standard conditions
    $\theta = \{[Na^+] = 1\mu M, [Mg^{2+}] = 0\mu M\}$
  - $\eta_{SIM} = 12, \eta_{ON} = 0.7, \eta_{OFF} = 0.3, \delta = 5^oC$
  - $\Gamma = \{30^oC, 31^oC, 32^oC \ldots, 80^oC\}$



An equivalent job on 1 core with same specifiers has Wall Clock Time: 4hrs 32mins 34sec

24 Core Wall Clock Time: 22mins 19sec

Advantages of This Approach: Modularity

- Easy to build other thermodynamic optimization methods leveraging general intermediate results or simply the existing general processor spawning code.
  - NUPACK intermediate analysis calculations can give other interesting metrics eg complex free energy
- Example 1: Bounds on ON-target melting temperature
  - We can use λ to optimize the library such that $\max(\lambda) - \min(\lambda) \leq \delta_{bound}$
- Example 2: Approximating thermodynamic equilibrium in a pot
  - We can change off target optimization to include selecting for each $i, j \in L, i \neq j$ against high:
    - $1 - \dfrac{E([(i,\tilde{i})] \,|\, I_c = \{[i] = c, [\tilde{i}] = c, [j] = c, [\tilde{j}] = c\}, T = \tau, \Theta, \Psi = \psi)}{c}$
  - (Where $\psi = \{(i,i), (i,\tilde{i}), (i,j), (i,\tilde{j}), (\tilde{i},\tilde{i}), (\tilde{i},j), (\tilde{i},\tilde{j}), (j,j), (j,\tilde{j}), (\tilde{j},\tilde{j}), (i), (\tilde{i}), (j), (\tilde{j})\}$)

These examples are existing methods already written

Disadvantages and Things to Work on

- Currently using *forkspawn* in process generation
  - Forks from an initial python parent process, not the current parent process.
  - This means we reimport different libraries and functions for each new process
  - Use pickle to pass objects from parent to child.
- This is due to an issue where using *fork* caused deadlocks when NUPACK was imported.
  - Python 3.13 might fix this with an experimental mode without the Python Global Interpreter Lock…
- Disadvantage: cannot be run in a Jupyter notebook.
  - Cells are not “main” scripts
  - But can execute python script separately and pickle and depickle intermediate calculations of interest into a notebook.