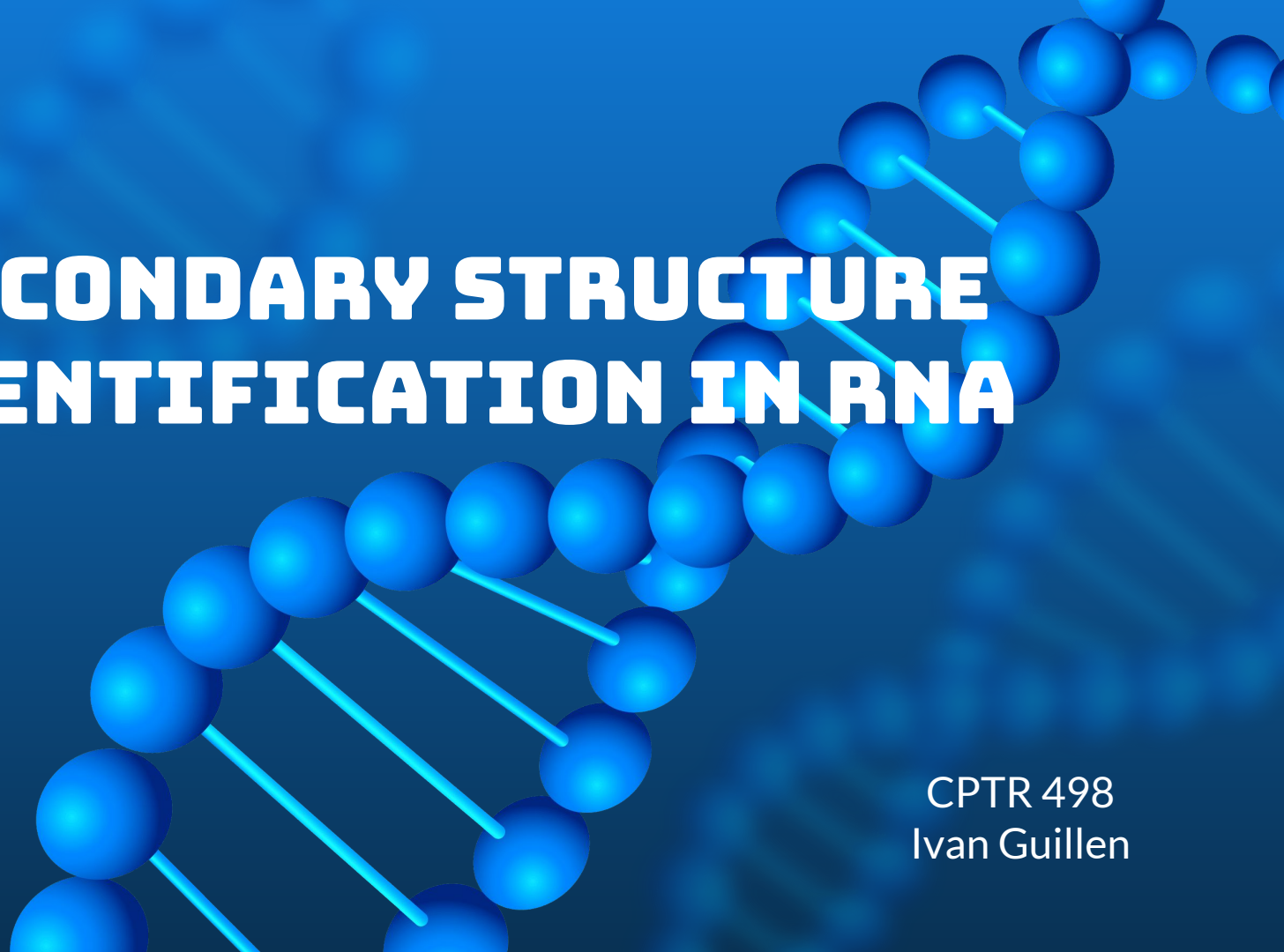
SECONDARY STRUCTURE IDENTIFICATION IN RNA



CPTR 498
Ivan Guillen

ORIGINAL GOAL

“The end goal for the project would be to have a program that could take a FASTA file with genetic sequences, and another file of potential editing sites in that FASTA file and mark which potential editing sites could form a double stranded structure with another part of the sequence.”

- Dr. Kirt Onthank

INPUT

FASTA

- List of genetic sequences
- Each item in file has:
  - Sequence ID
  - Sequence

```
>lcl|TRINITY_DN13654_c4_g1_i1:c99-22
TTGTGTCTAGTTAAATTACTAGTTTCAGAGAATGCTTTA
CCACAGATTTCACATTCATATGGTTTCTCTCCTGTATGA
```

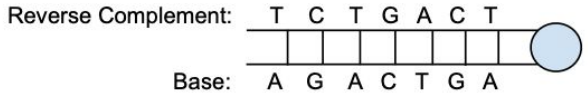
CSV

- List of potential edits
- Each item in file has:
  - Sequence ID
  - Edit site

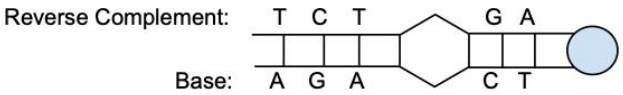
```
orf,pos,mrna_con
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,59,T
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,26,A
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,48,T
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,20,A
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,74,A
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,40,C
lcl|TRINITY_DN13654_c4_g1_i1:c99-22,65,C
```

WHAT IS A SECONDARY STRUCTURE?

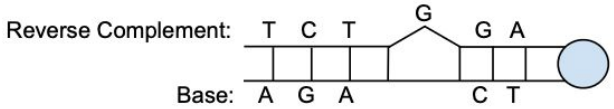
Hairpin Loops are a common type of secondary structure that are created when a sequence of RNA folds upon itself and forms base pairs with another section of the same sequence.



Internal Loops are similar, but feature a short sequence of unpaired bases within a larger sequence of paired bases.



Bulges are also similar, but feature regions on one side of the folded structure that have extra bases with no corresponding bases on the opposite side.



- Nucleic acid secondary structures are base-pairing interactions that occur within the same sequence.
- 3 common types:
  - Hairpin Loops
  - Internal Loops
  - Bulges

MODIFIED GOAL

The end goal for this project is to create a program that could take a FASTA file with genetic sequences and a CSV file of potential editing sites in that FASTA file, and **identify the longest secondary structure possible for each edit**.

OUTPUT

Hairpin Loops	pos	len	base	base_loc	rev_comp	rev_comp_loc
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	26	6	AGAGAA	[26, 31]	TTCTCT	[63, 68]
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	20	5	ACTAG	[17, 21]	CTAGT	[6, 10]
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	65	6	TTCTCT	[63, 68]	AGAGAA	[26, 31]
Internal Loops						
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	48	7	ATTT.AC	[45, 51]	GT.AAAT	[9, 15]
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	40	6	ACCA.A	[38, 43]	T.TGGT	[57, 62]
Bulges						
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	26	6	CAGAGA	[25, 30]	TCT.CTG	[66, 72]
lcl|TRINITY_DN13654_c4_g1_i1:c99-22	48	5	ATTT.A	[45, 50]	TAAAT	[11, 15]

ITERATIONS

Symmetrical Search Outwards from Edit Site:

- Each iteration, check left and right. Add if reverse complement exists.
- Issue: What if adding a base to the left limits the search to the right (or vice versa)?

ITERATIONS

Search All the Way One Way, then Increase Length by Searching the Other Way:

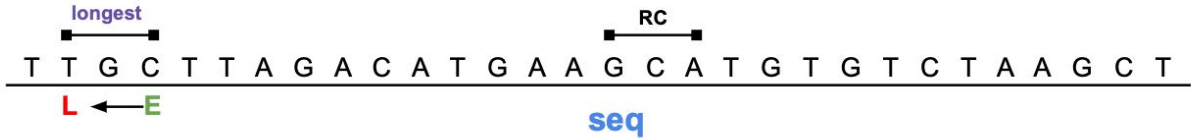
- Issue: What if searching one way first limits the search the other way?

Current Implementation - Step 1

Our goal is to find the longest reverse complement at each edit site. Thus, given a sequence (**seq**) and edit site (**E**):

Step 1: Find Longest Reverse Complement to the Left

- Set **L** = **E**
- While the reverse complement of **seq[L:E]** exists in either **seq[:L]** or **seq[E:]**,
  - Decrement **L** by 1
- Set **longest** to **seq[L:E]**



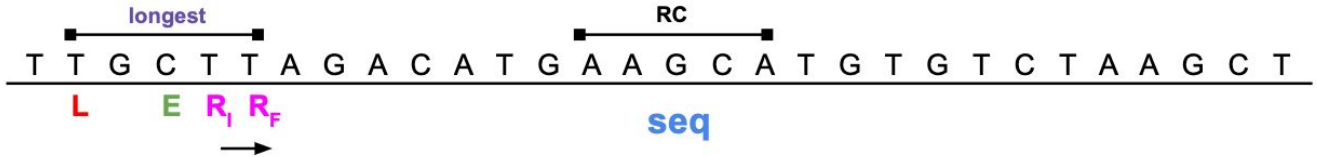
Current Implementation - Step 2a

Step 2: Find Longest Reverse Complement

- Set R = E + 1
- While L ≤ E,
  - While the reverse complement of seq[L:R] exists in either seq[:L] or seq[R:],
    - Set longest to seq[L:R]
    - Increment R by 1
  - Increment L and R by 1
- Return longest

CURRENT IMPLEMENTATION - STEP 2B

Iteration 1:



Iteration 2:

Iteration 3:

MODIFICATIONS

- While the base algorithm is similar for all three types, additional modifications were needed to account for unpaired/extra bases in internal loops and bulges.
- Regular expression searches were used instead of Python's built-in string.find() and periods (".") were used to simulate gaps between paired bases.
- A boolean flag was set if a jump over the unpaired/extra bases could be made. The jump would only be made if necessary.

FUTURE ADDITIONS

- Additional support for different types of secondary structures
  - Pseudoknots
- Additional search parameters
- Multi-thread capacity