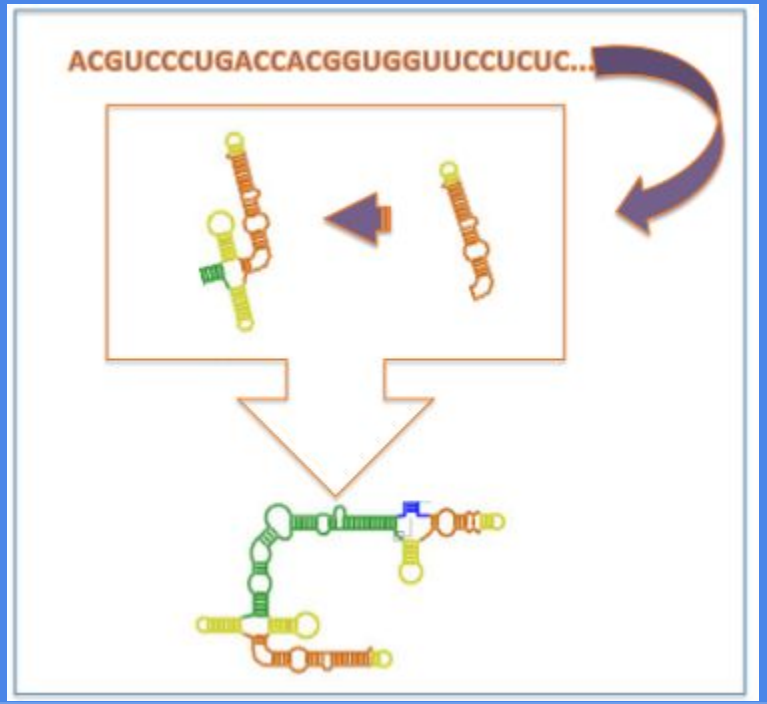
RNA Folding



Vishal Patel, Ashton Berger, & Katie Fields

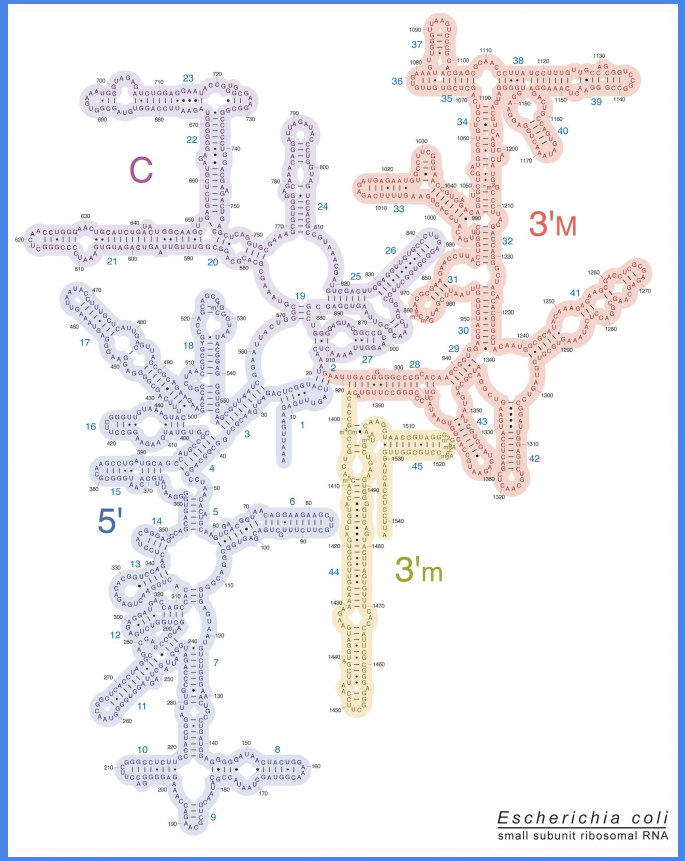
Presentation Layout

1. Motivation
2. Background
3. Materials and Methods
   - 3.1. Designing the algorithm
   - 3.2. Software development
4. Results
   - 4.1. Visualization results
   - 4.2. Software accomplishments
   - 4.3. Secondary structure prediction results
5. Future Goals

Motivation

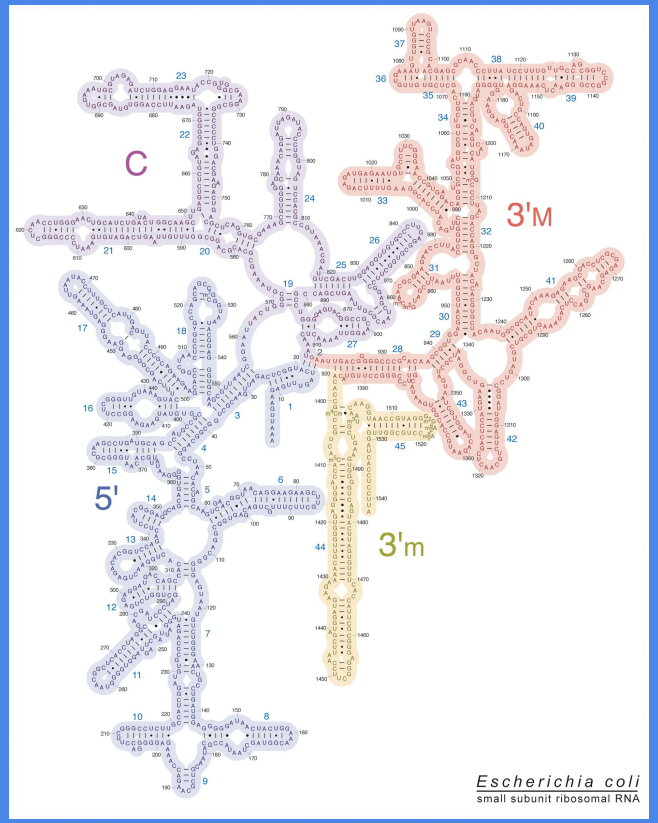
Motivation

- E. coli 16s rRNA

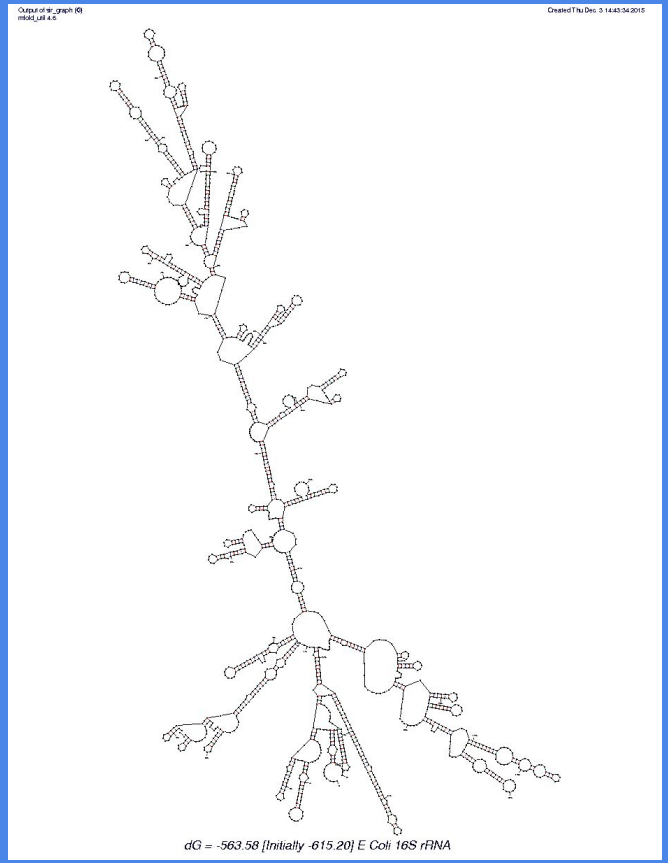


Motivation

- E. coli 16s rRNA

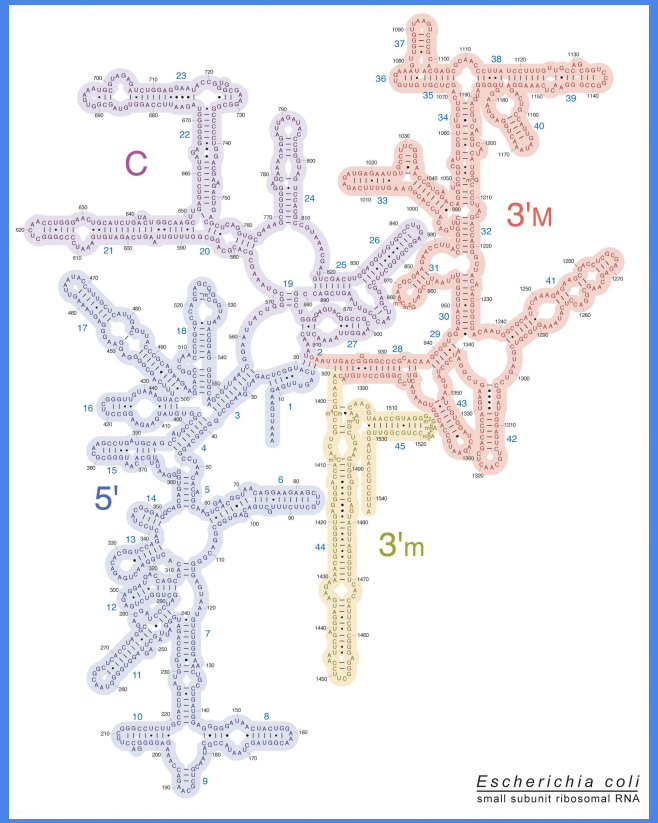


- Mfold

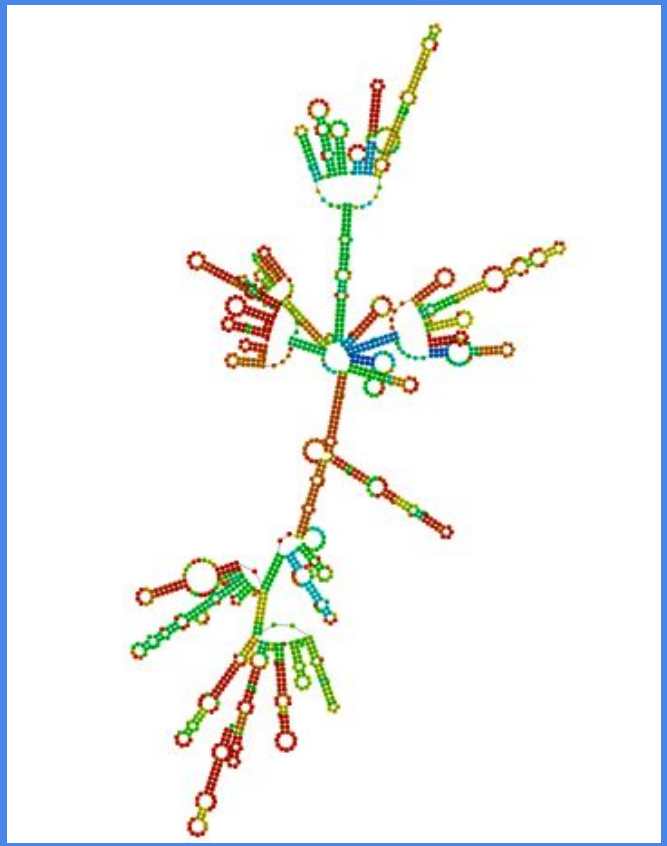


Motivation

- E. coli 16s rRNA

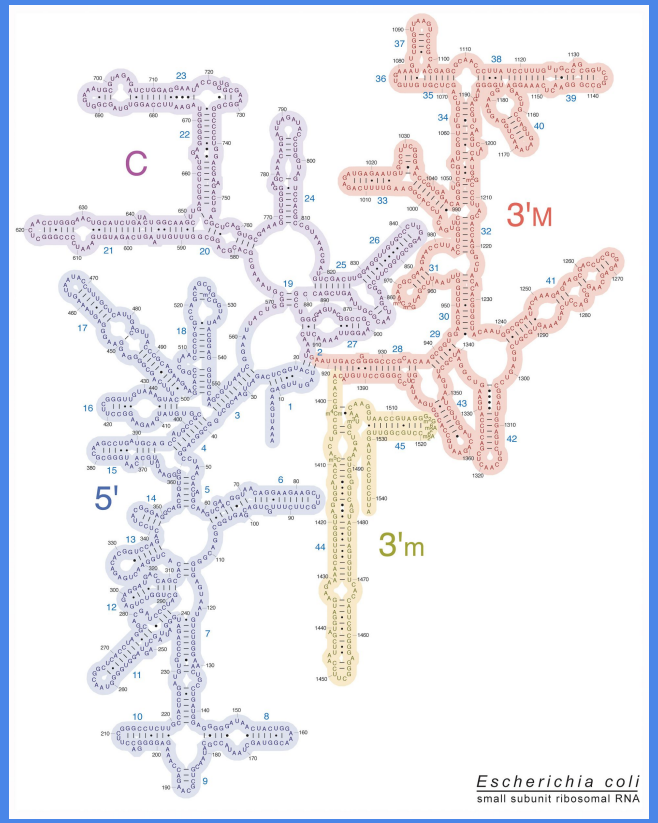


- RNAfold

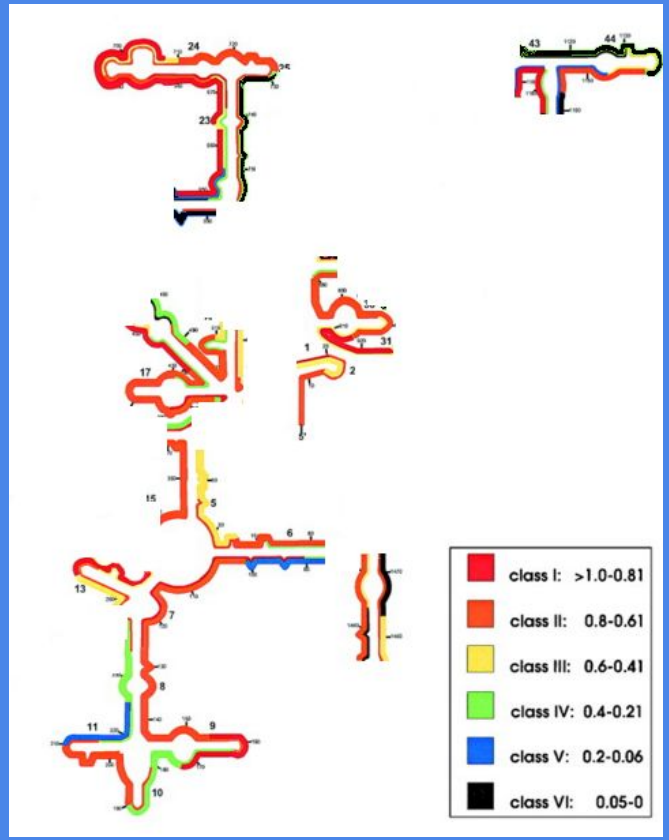


Motivation

- E. coli 16s rRNA

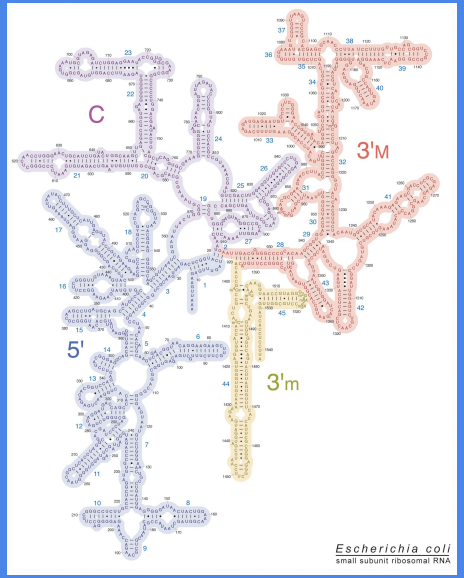


- Behrens et al 2003

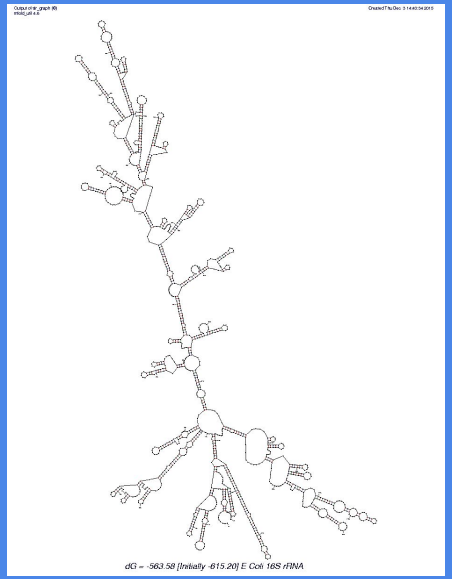


Motivation

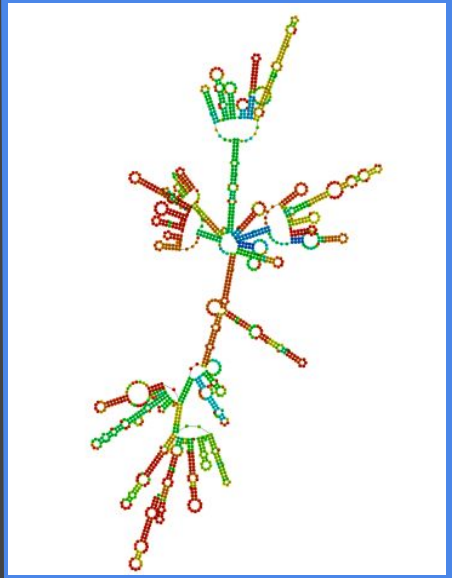
- E. coli 16s rRNA



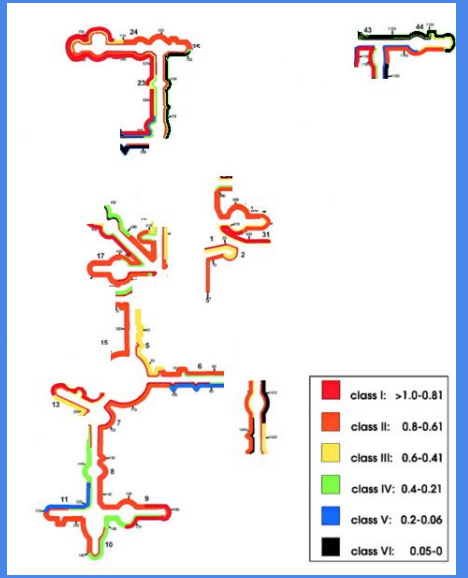
- Mfold



- RNAfold



- Behrens



Motivation

- Currently, an RNA secondary structure prediction algorithm does not exist that can correctly fold RNA sequences into their known structures
- Lots of software currently available based on various methods, but no solution that is 100% accurate in folding an RNA sequence into its correct secondary structure
- Despite this, much research is still performed with and based on predictions that are given from inaccurate software
    - Over 60,600 journal publication results returned from searching “RNA secondary structure” in Google Scholar… just from 2015 alone
    - If majority of these results are publications based on predictions made by inaccurate algorithms, this is a massive waste of resources and efforts!

Motivation

We aim to accomplish several majors goals with our project:

1. Illustrate the inherent difficulty of the problem and our attempts using computer-generated visualizations
2. Solve the RNA folding problem once and for all by developing a new, biologically-relevant algorithm that can correctly predict the secondary structure for RNA sequences
3. Implement the algorithm within documented software than can continue to be used easily and efficiently by anyone who chooses to work on the project in the future

Background

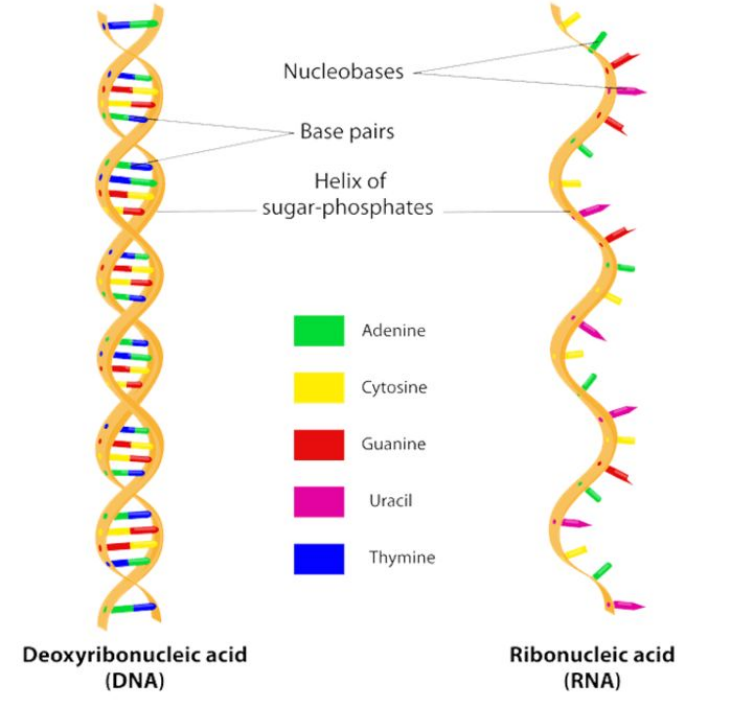
Background - RNA

- One of the “three essential macromolecules of life”

- Is a nucleic acid composed of nucleotide monomers
  - RNA nucleotides: guanine, uracil, adenine, cytosine

- Has many functions, including, but not limited to
  - encoding genetic information
  - cellular signaling
  - reaction catalysis
  - protein synthesis
  - etc

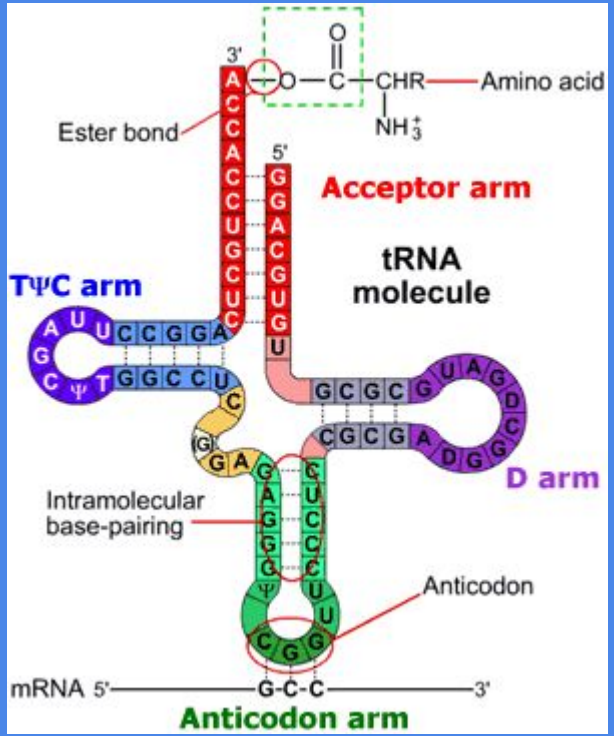
DNA helix vs. single stranded RNA



http://www.aboutthemcat.org/images/biology/dna-vs-rna.png

Background - RNA

- Why is secondary structure important?
  - RNA secondary structure can help us infer what the function of a specific RNA is
  - tRNA
  - Secondary structure can predict whether an mRNA will have problems during translation
  - Secondary structure can help with predictions of RNA tertiary structure

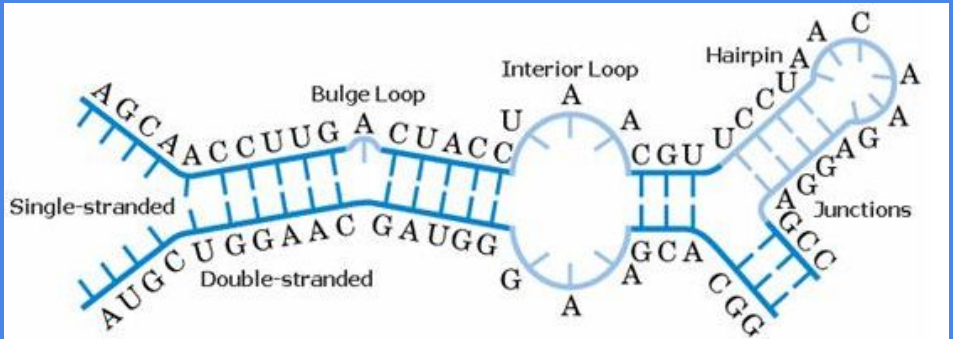


http://higheredbcs.wiley.com/legacy/college/boyer/0471661791/structure/tRNA/trna_diagram_small.gif

Background - RNA

- Nucleotides on a strand of RNA can base pair with other nucleotides from the same strand within proximity, leading to overall structures and helices
  - Hairpin Loops
  - Bulges
  - Interior Loops
- The final folded sequence and its resulting structures is the secondary structure

Examples of various structures



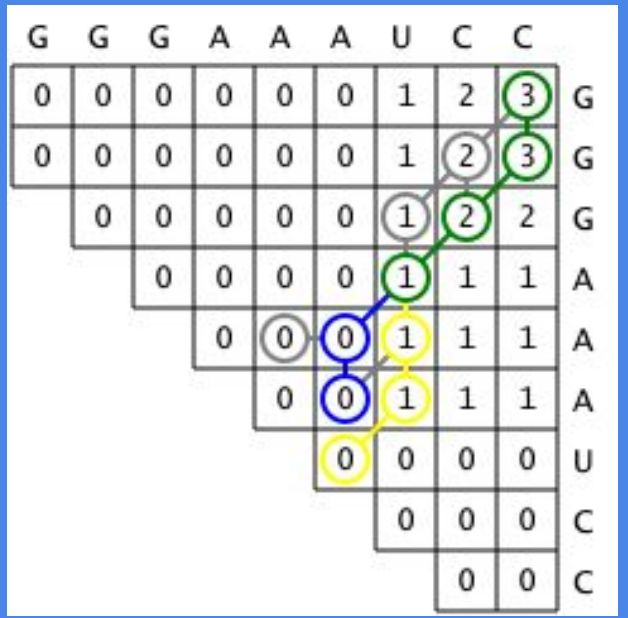
http://www.http://www.gene-quantification.de/image-rna-structure-1.gif

Background - Currently available software

Old Programs/ Algorithms

- Nussinov-Jacobsen Algorithm
  - Several current folding algorithms are based on this model, but this model is incorrect
  - We know this because the predictions based on this algorithm are not accurate!
- Other methods that current folding algorithms use:
  - supervised learning models
  - free energy minimization
  - energy-weighted partitioning of structures
  - simulated annealing
  - centroid estimization
  - etc...

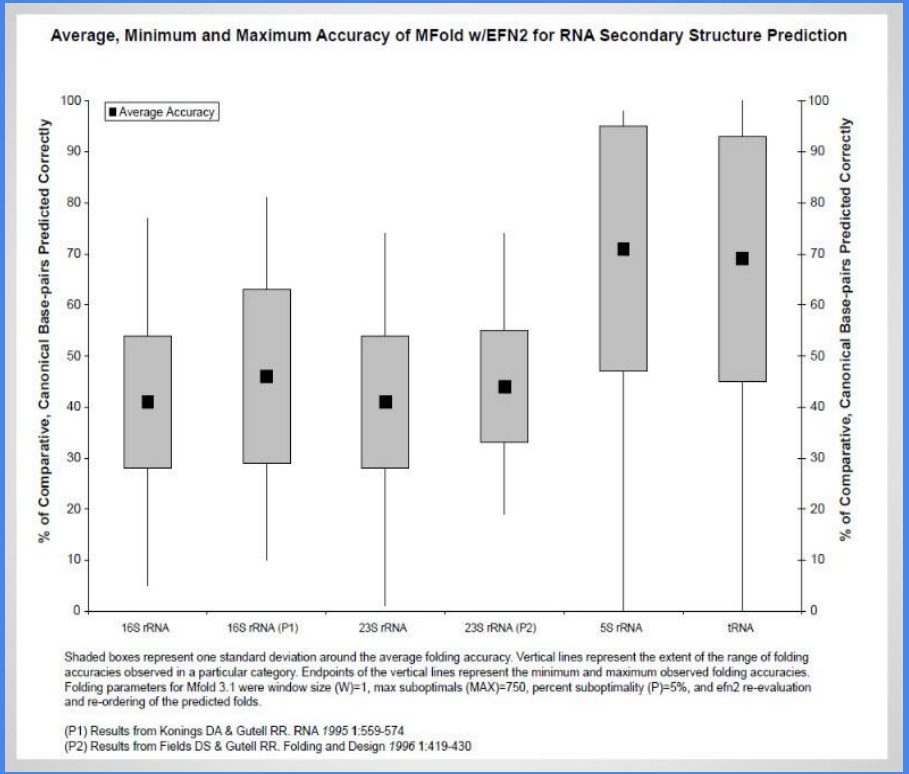
Nussinov-Jacobsen basis



http://datamech.com/devan/images/traceback_cmp.png

Background - Currently available software

- **Mfold**
  - Low accuracy in predictions
  - As the size of the sequence increases the accuracy of the predictions decreases



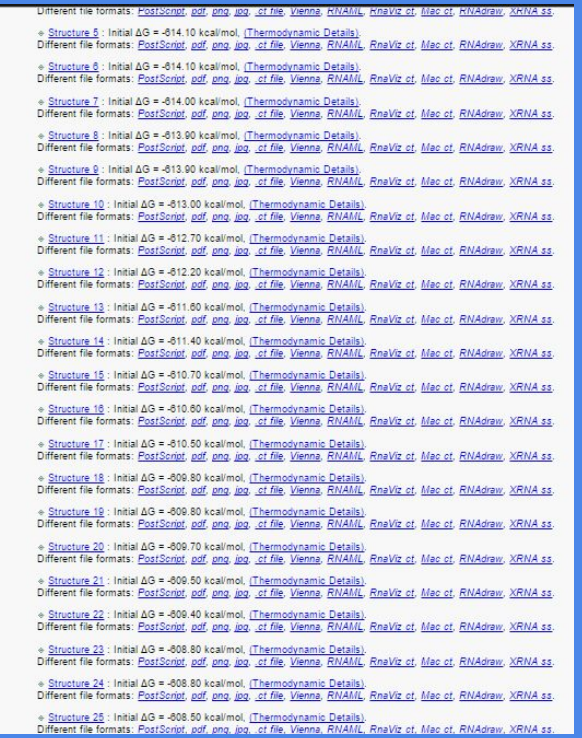
Background - Currently available software

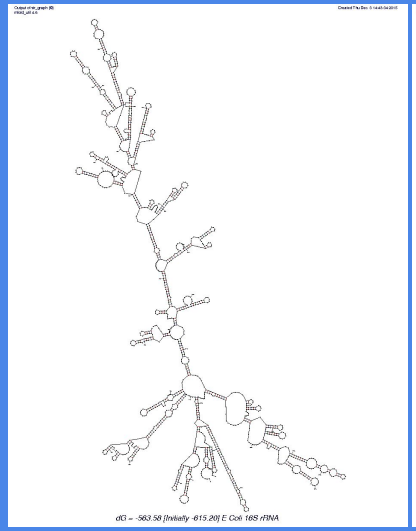
- MFold

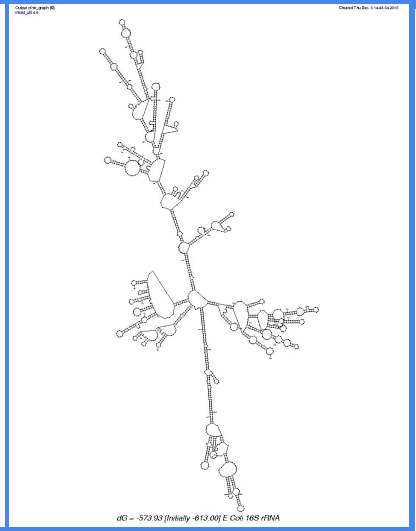
List of Mfold predictions for E. Coli 16s rRNA structure

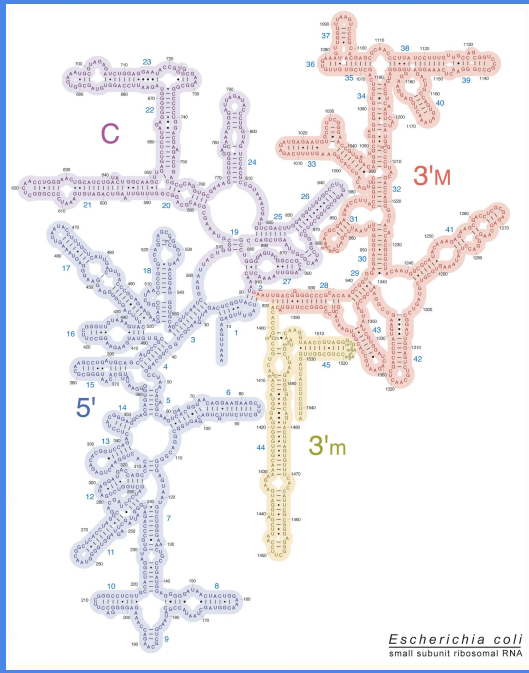
Some of the predicted structures for 16s

Correct E. Coli 16s rRNA structure







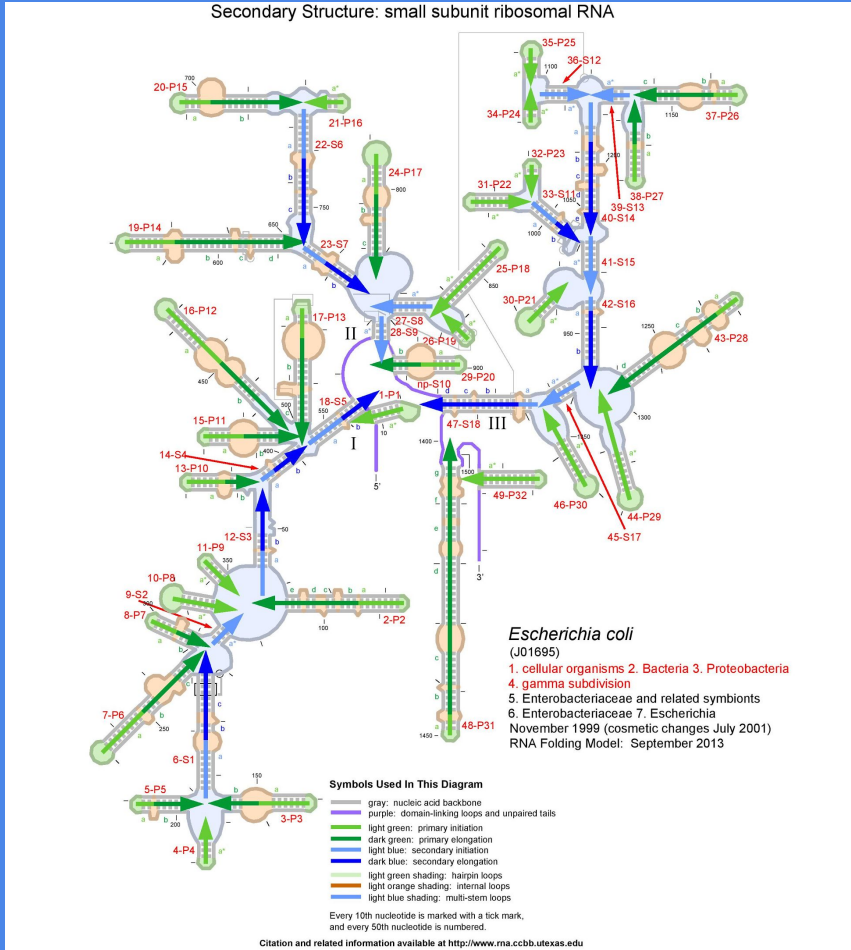


http://rna.ucsc.edu/rnacenter/images/figs/ecoli_16s.jpg

Methods I:
The Algorithm

Materials and Methods - How does RNA fold?

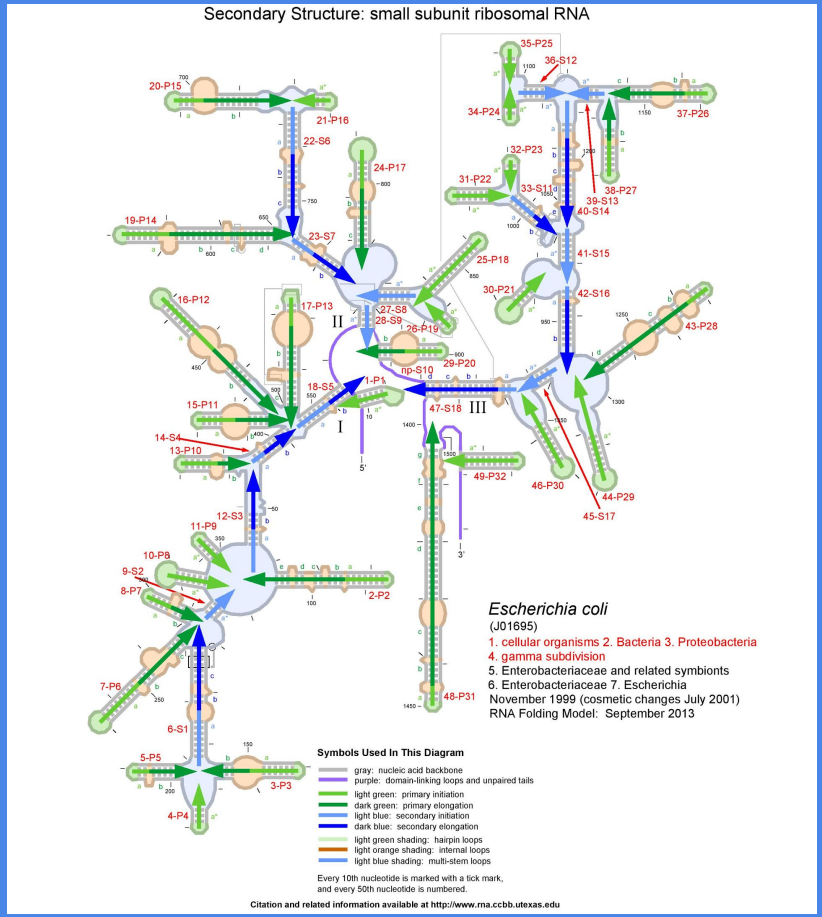
- **Visualize using PieSie**
  - format developed by The Gutell Lab
  - Compound Helix is made up of helices and various other structures
    - Pi: Primary Initiation
    - Pe: Primary Elongation
    - Si: Secondary Initiation
    - Se: Secondary Elongation
  - Pi helices form and then Pe(s) elongate from the Pi
  - Si helices from at junctions where several Pe's meet and Se(s) elongate from there
  - PieSie helps visualize how secondary structure biologically forms, let's use it to help design our algorithm!



Materials and Methods - Algorithm

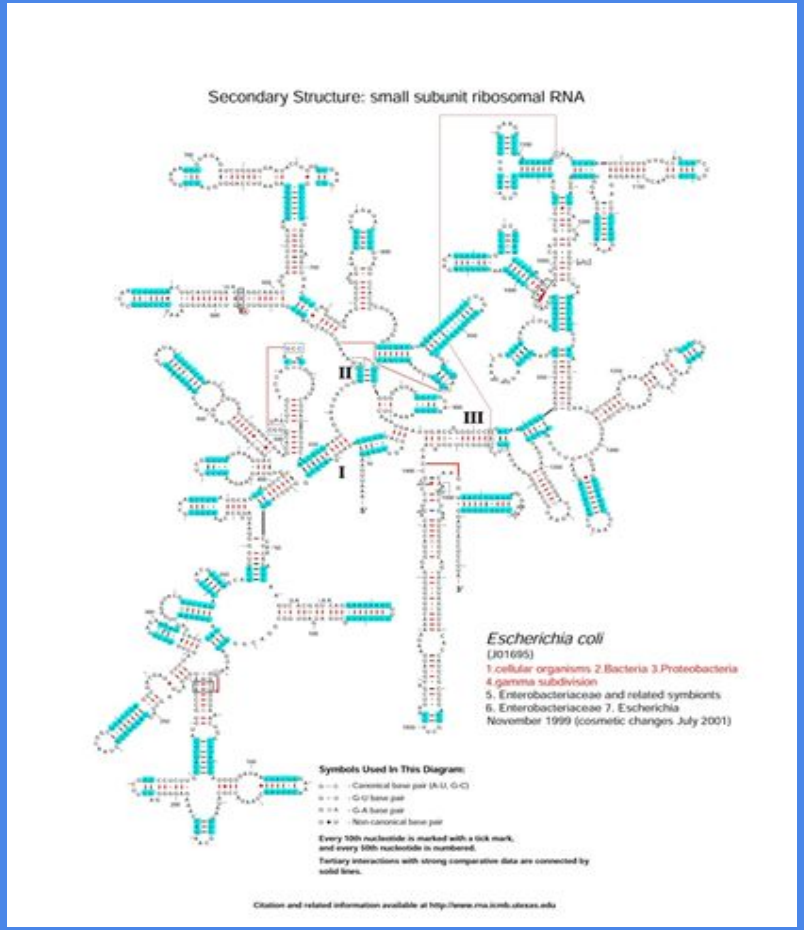
Proposed structure of the overall algorithm:

1. identify potential Pis in order from 5’ to 3’
2. extend the potential Pis with Pe helices
3. dynamically adjust/recalculate helices based on conditional distances/feature scoring/modified energies
4. identify potential Sis from Pe junctions
5. extend the potential Sis with Se helices
6. dynamically adjust/recalculate helices based on conditional distances/feature scoring/modified energies
7. take the resulting collection of compound helices and structures and perform optimization step to combine into a final predicted secondary structure



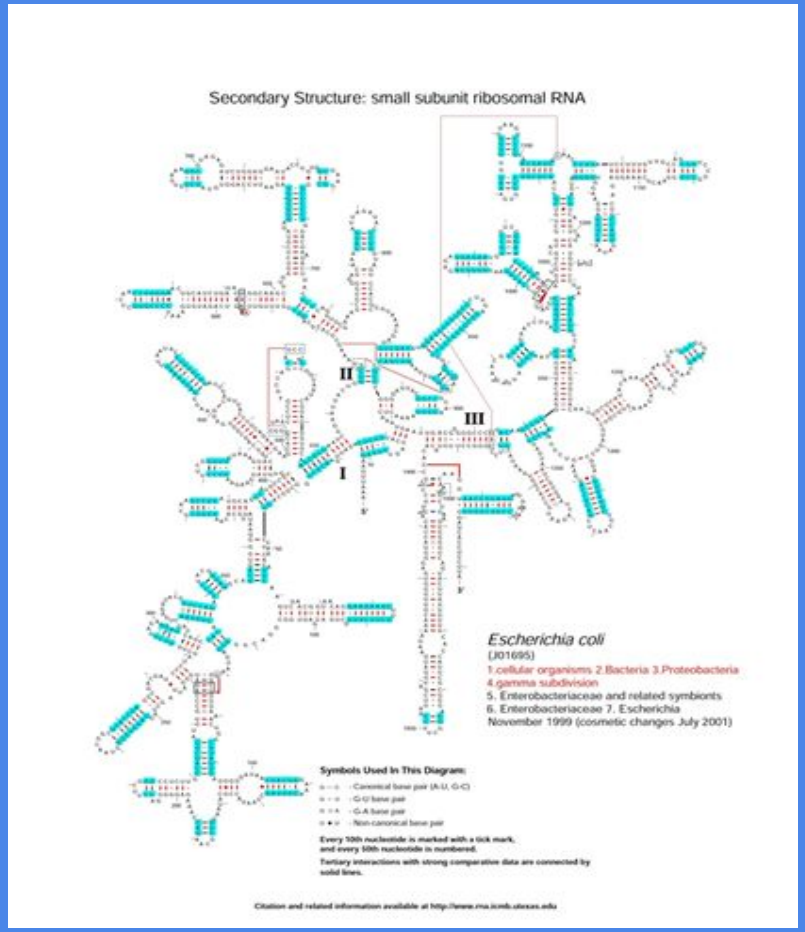
Materials and Methods - Algorithm

- **Major algorithm goal: developing the extension algorithm**
  - Very complex and difficult
    - RNA/DNA “breathes”
      - 5e-12 s
    - many, many structures are possible
      - $4.3 * 10^{393}$
        - possible helices for E. coli 16s
      - 10^{85}
        - estimated number of fundamental particles in the observable universe
  - Develop algorithm and implement in software that, given Pi and Si helices, predicts Pe and Se helices (Primary and Secondary Elongation)
  - Target structures of interest highlighted in blue on the right



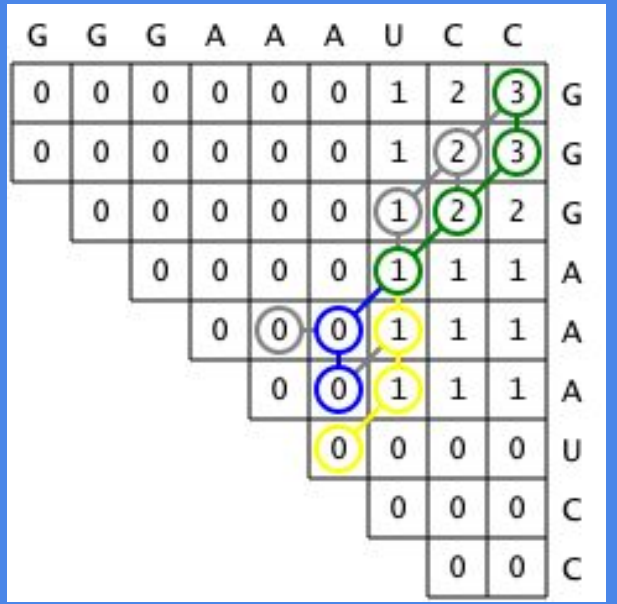
Materials and Methods - Algorithm

- **Where do we start?**
  - Start with a given: predict from known PIs and SIs and try to perfect the extension of the compound helix
  - i.e. successfully predict all of the PEs and SEs for a structure given the starting points
  - Target structures of interest for E. coli 16s highlighted in blue on the right
- Why?
  - The extension algorithm is vital to the prediction of majority of RNA secondary structure
    - 231 / 1542 nucleotides in E. coli 16s rRNA belong to PIs and SIs
    - Majority of the remaining 1311 base pairs are apart of PEs and SEs



Materials and Methods - Extension Algorithm

- **Extension Algorithm Principles - Simple Distance First!**
  - Helices that are made up of nucleotides that are closest together should form first
  - Helix formation should begin at the diagonal and work outwards towards the corner
  - Helices that form should minimize the distance between helices
  - The most stable helices should form within a window of 10-15 nucleotides from the previous helix
    - This reflects what is seen via comparative analysis of known RNA structures

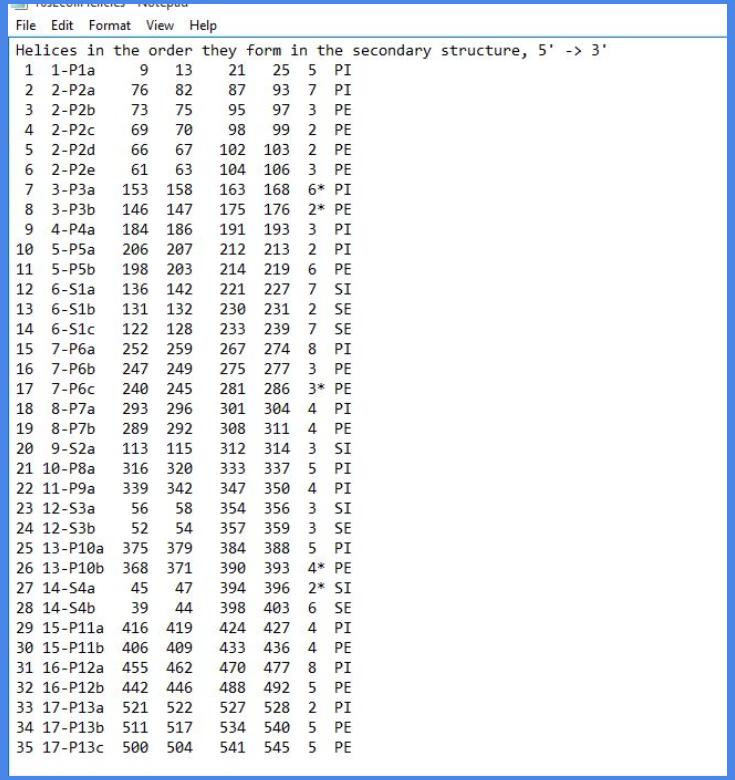


http://datamech.com/devan/images/traceback_cmp.png

Materials and Methods - Extension Algorithm

Primary Elongation (PE) and Secondary Elongation (SE) Implementation

1. Read in a piesie file containing all correct helices and store these in the program
2. Get all the PI and SI helices and iterate through them and predict the elongation for each of these initial helices

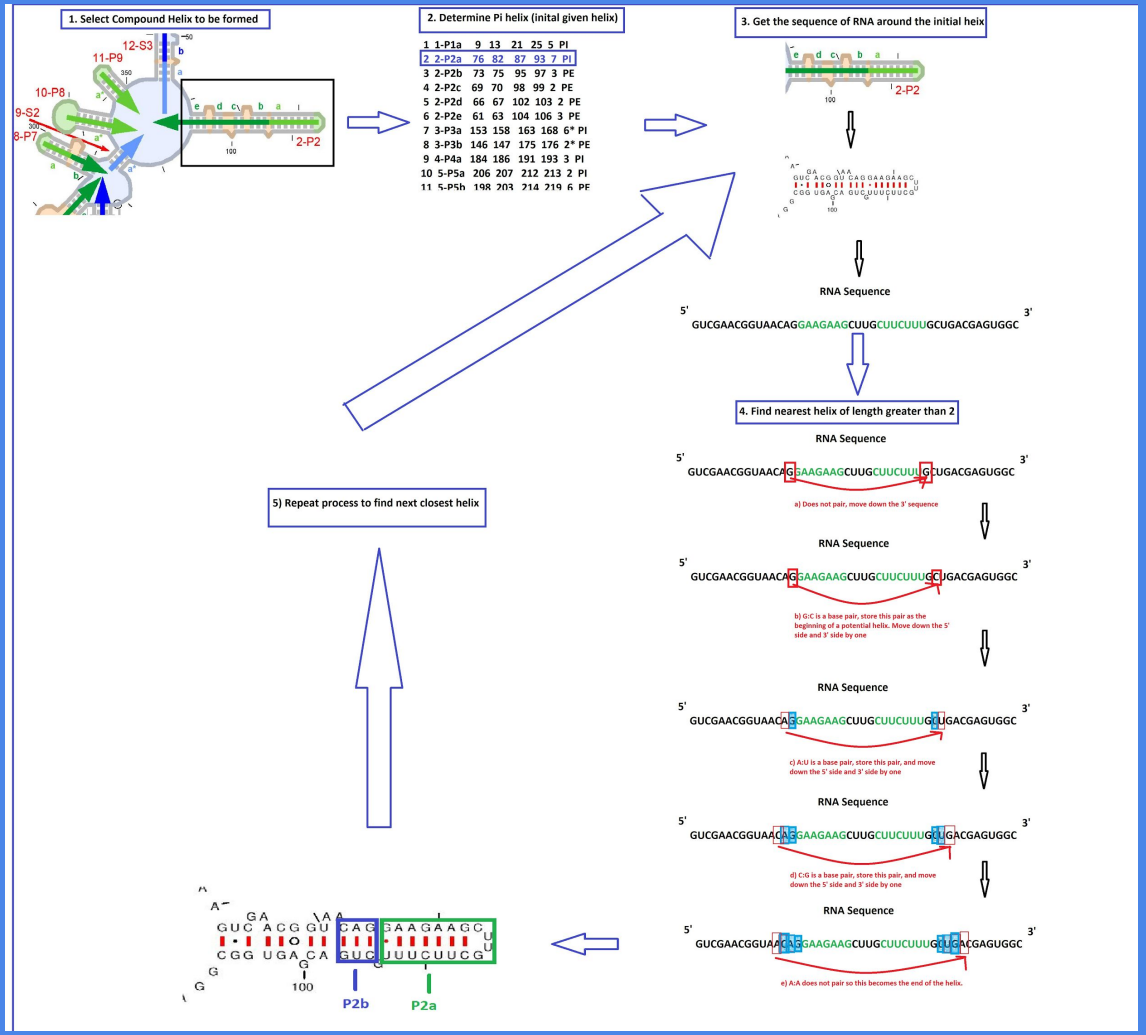


File Edit Format View Help

Helices in the order they form in the secondary structure, 5' -> 3'

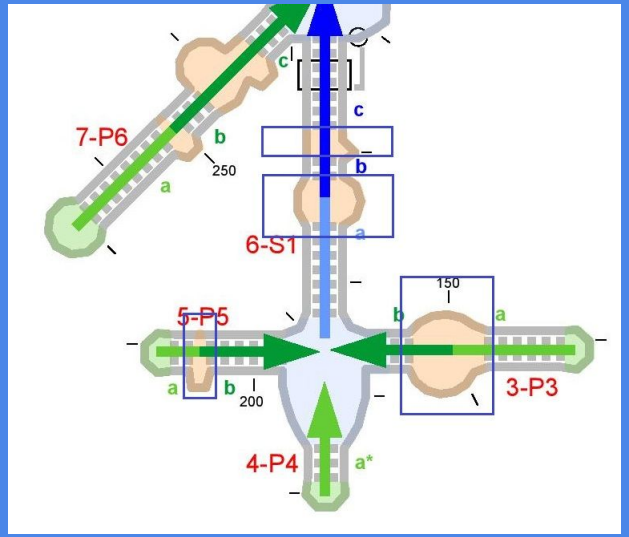
1	1-P1a	9	13	21	25	5	PI
2	2-P2a	76	82	87	93	7	PI
3	2-P2b	73	75	95	97	3	PE
4	2-P2c	69	70	98	99	2	PE
5	2-P2d	66	67	102	103	2	PE
6	2-P2e	61	63	104	106	3	PE
7	3-P3a	153	158	163	168	6*	PI
8	3-P3b	146	147	175	176	2*	PE
9	4-P4a	184	186	191	193	3	PI
10	5-P5a	206	207	212	213	2	PI
11	5-P5b	198	203	214	219	6	PE
12	6-S1a	136	142	221	227	7	SI
13	6-S1b	131	132	230	231	2	SE
14	6-S1c	122	128	233	239	7	SE
15	7-P6a	252	259	267	274	8	PI
16	7-P6b	247	249	275	277	3	PE
17	7-P6c	240	245	281	286	3*	PE
18	8-P7a	293	296	301	304	4	PI
19	8-P7b	289	292	308	311	4	PE
20	9-S2a	113	115	312	314	3	SI
21	10-P8a	316	320	333	337	5	PI
22	11-P9a	339	342	347	350	4	PI
23	12-S3a	56	58	354	356	3	SI
24	12-S3b	52	54	357	359	3	SE
25	13-P10a	375	379	384	388	5	PI
26	13-P10b	368	371	390	393	4*	PE
27	14-S4a	45	47	394	396	2*	SI
28	14-S4b	39	44	398	403	6	SE
29	15-P11a	416	419	424	427	4	PI
30	15-P11b	406	409	433	436	4	PE
31	16-P12a	455	462	470	477	8	PI
32	16-P12b	442	446	488	492	5	PE
33	17-P13a	521	522	527	528	2	PI
34	17-P13b	511	517	534	540	5	PE
35	17-P13c	500	504	541	545	5	PE

3. Begin by selecting a helix based on shortest CD and a helix that has most balanced unpaired region



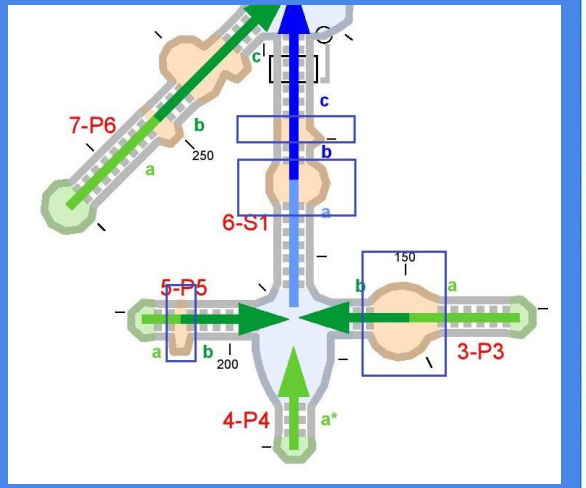
Statistics for Unpaired Regions

```
Compound Helix: P1
Distance = 0

Compound Helix: P2
P2a -P2b  5' distance: 0 || 3' distance: 1 || CD: 1  ||  5' end RNA sequence:            3' end RNA sequence: G          ||           5' sequence compos
P2b -P2c  5' distance: 2 || 3' distance: 0 || CD: 2  ||  5' end RNA sequence: AA         3' end RNA sequence:            ||           5' sequence compos
P2c -P2d  5' distance: 1 || 3' distance: 2 || CD: 3  ||  5' end RNA sequence: G          3' end RNA sequence: GA         ||           5' sequence compos
P2d -P2e  5' distance: 2 || 3' distance: 0 || CD: 2  ||  5' end RNA sequence: GA         3' end RNA sequence:            ||           5' sequence compos
Average:  5' distance: 1.25 | 3' distance: 0.75 | CD: 2.0
Average:  5' composition: A: 37.5% C: 0.0% G: 37.5% U: 0.0  ||  3' composition: A: 12.5% C: 0.0% G: 37.5% U: 0.0
Average:      composition: A: 25.0% C: 0.0% G: 37.5% U: 0.0%

Compound Helix: P3
P3a -P3b  5' distance: 5 || 3' distance: 6 || CD: 11  ||  5' end RNA sequence: GAUAA      3' end RNA sequence: CUAAUA      ||           5' sequence compo
Average:  5' distance: 5.0 | 3' distance: 6.0 | CD: 11.0
Average:  5' composition: A: 60.0% C: 0.0% G: 20.0% U: 20.0  ||  3' composition: A: 50.0% C: 17.0% G: 0.0% U: 33.0
Average:      composition: A: 55.0% C: 8.5% G: 10.0% U: 26.5%

Compound Helix: P4
Distance = 0

Compound Helix: P5
P5a -P5b  5' distance: 2 || 3' distance: 0 || CD: 2  ||  5' end RNA sequence: GA         3' end RNA sequence:            ||           5' sequence compos
Average:  5' distance: 2.0 | 3' distance: 0.0 | CD: 2.0
Average:  5' composition: A: 50.0% C: 0.0% G: 50.0% U: 0.0  ||  3' composition: A: 0.0% C: 0.0% G: 0.0% U: 0.0
Average:      composition: A: 25.0% C: 0.0% G: 25.0% U: 0.0%

Compound Helix: S1
S1a -S1b  5' distance: 3 || 3' distance: 2 || CD: 5  ||  5' end RNA sequence: UGC        3' end RNA sequence: AU         ||           5' sequence compos
S1b -S1c  5' distance: 2 || 3' distance: 1 || CD: 3  ||  5' end RNA sequence: AA         3' end RNA sequence: G          ||           5' sequence compos
Average:  5' distance: 2.5 | 3' distance: 1.5 | CD: 4.0
Average:  5' composition: A: 50.0% C: 16.5% G: 16.5% U: 16.5  ||  3' composition: A: 25.0% C: 0.0% G: 50.0% U: 25.0
Average:      composition: A: 37.5% C: 8.25% G: 33.25% U: 20.75%

Compound Helix: P6
P6a -P6b  5' distance: 2 || 3' distance: 0 || CD: 2  ||  5' end RNA sequence: AG         3' end RNA sequence:            ||           5' sequence compos
```



Statistics for Unpaired Regions

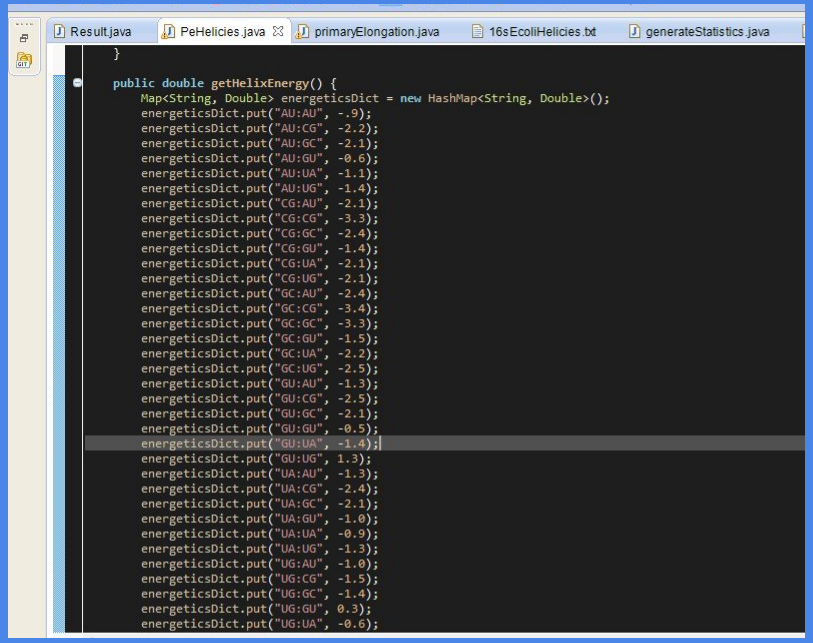
- Averages
- CD < 3
- These unpaired regions mainly consist of As and Gs
- The length of the 5’ region is very close to the length of the 3’ region
- Balance = 2

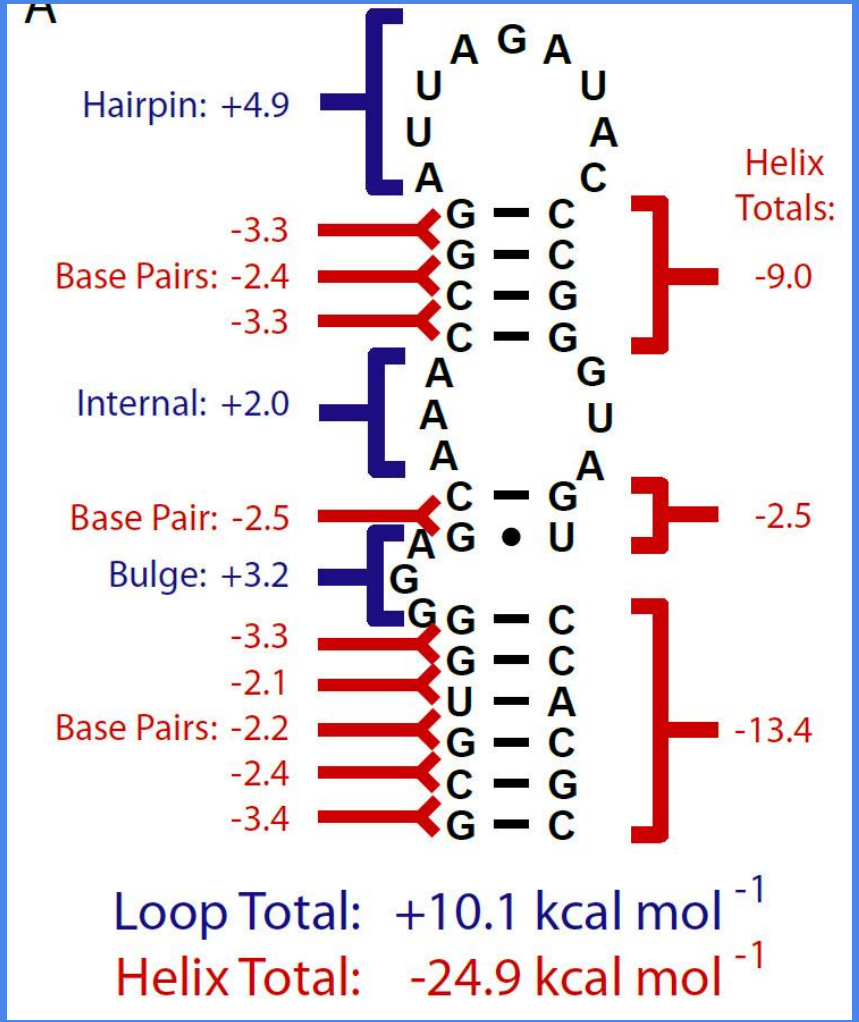



```
Average:     composition: A: 16.665% C: 36.165% G: 16.665% U: 13

Compound Helix: P31
P31a-P31b 5' distance: 2 || 3' distance: 1 || CD: 3  ||  5' end
P31b-P31c 5' distance: 1 || 3' distance: 0 || CD: 1  ||  5' end
P31c-P31d 5' distance: 4 || 3' distance: 3 || CD: 7  ||  5' end
P31d-P31e 5' distance: 2 || 3' distance: 2 || CD: 4  ||  5' end
P31e-P31f 5' distance: 1 || 3' distance: 1 || CD: 2  ||  5' end
P31f-P31g 5' distance: 3 || 3' distance: 4 || CD: 7  ||  5' end
Average:  5' distance: 2.17 | 3' distance: 1.83 | CD: 4.0
Average:  5' composition: A: 76.33% C: 5.5% G: 12.5% U: 5.5  ||
Average:     composition: A: 57.58% C: 8.335% G: 20.835% U: 4.83

Compound Helix: P32
Distance = 0


Whole Sequence Statistics:
Max Distance between helicies: 5' end: 8 | 3' end: 10
Average:  5' distance: 1.33 | 3' distance: 1.34 | CD: 2.67
Average:  5' composition: A: 20.88% C: 4.43% G: 14.55% U: 3.64
Average:     composition: A: 45.16% C: 8.9% G: 22.56% U: 9.67%
```

4. Determine the modified energy of the predicted helix

- ME = OE/CD
- OE is determined by next door neighbor energies

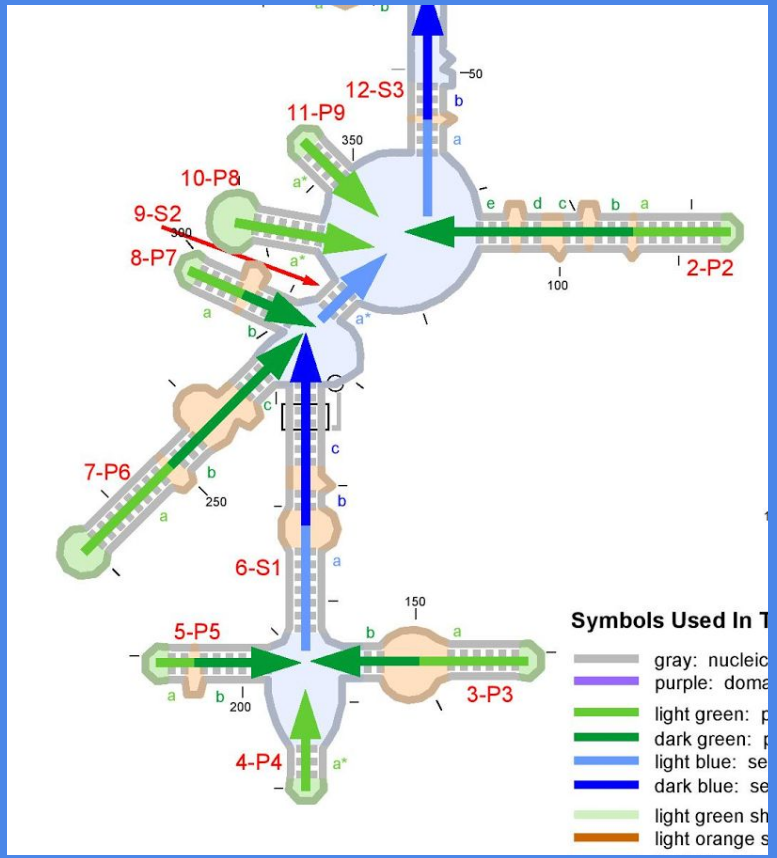




Competition between helices

5.

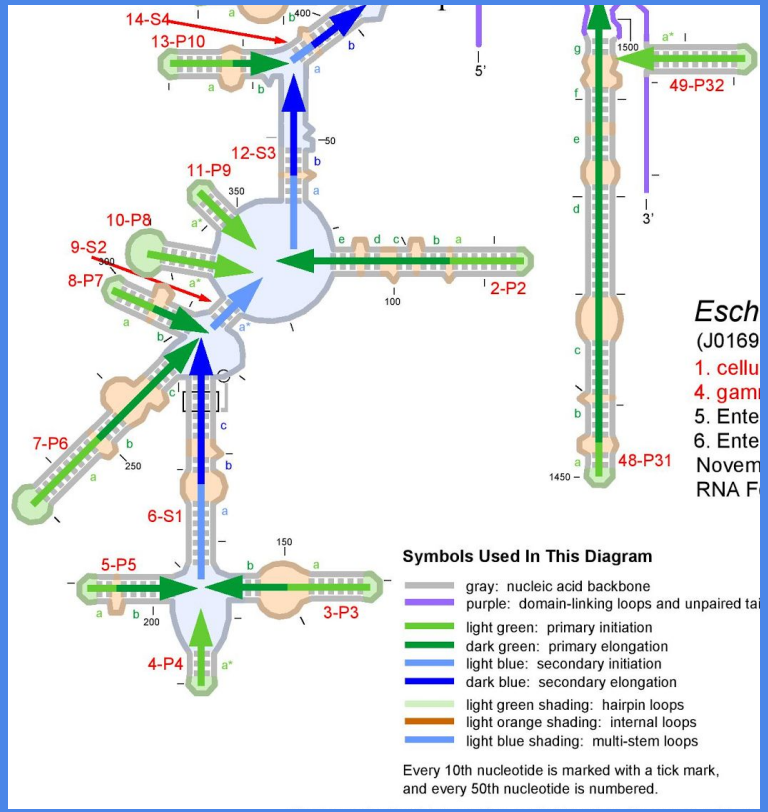
- Once an initial stable helix is formed based on shortest CD, a more stable helix within 5-10 nucleotides can form
- Current Window = 10
- If this helix is more stable, the entire previous helix is replaced or parts of it
- Program then keeps searching for a more stable helix
- Once it cannot find one, the current helix is the predicted helix



Termination of Elongation

6.

- Terminate elongation of compound helix whenever the following conditions are met:
  - no other helices can be found within the window
  - no other helices found before reaching another already predicted compound helix
  - no other helices can be found within the window that are stable enough to form
  - no other helices found before reaching a nearby Pi or Si helix



Output

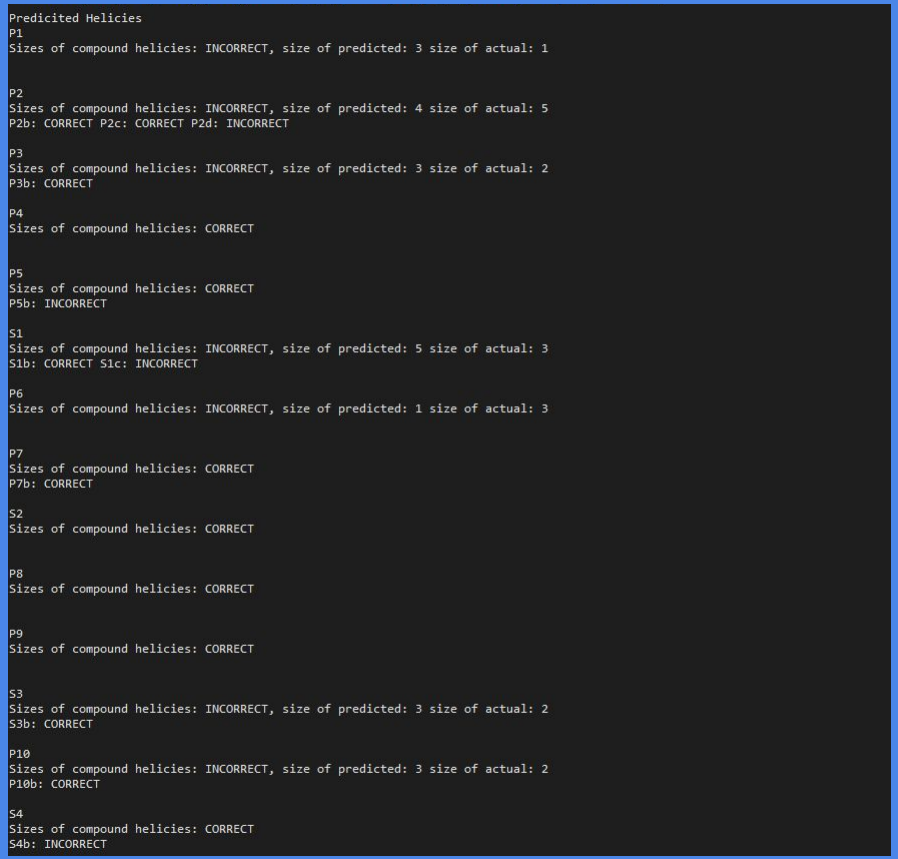
7.

```
P2
P2a
-9.700000000000001
76:G 77:A 78:A 79:G 80:A 81:A 82:G
93:U 92:U 91:U 90:C 89:U 88:U 87:C

P2b
-4.2
73:C 74:A 75:G
97:G 96:U 95:C

P2c
-2.2
69:G 70:U
99:C 98:A

P2d
-5.4
61:G 62:U 63:C 64:G
106:C 105:G 104:G 103:U


P3
P3a
-9.100000000000001
153:C 154:U 155:A 156:C 157:U 158:G
168:G 167:A 166:U 165:G 164:G 163:C

P3b
0.0
146:G 147:G
176:C 175:C


P4
P4a
-4.6
184:G 185:U 186:C
193:C 192:A 191:G


P5
P5a
-3.3
206:C 207:C
213:G 212:G

P5b
-9.600000000000001
199:A 200:G 201:G 202:G 203:G 204:G
219:U 218:U 217:C 216:U 215:C 214:C


S1
S1a
-12.399999999999999
136:C 137:U 138:G 139:A 140:U 141:G 142:G
227:G 226:G 225:C 224:U 223:A 222:C 221:C

S1b
-2.2
131:A 132:C
231:U 230:G

S1c
-14.3
121:U 122:G 123:U 124:C 125:U 126:G 127:G 128:G
240:G 239:U 238:A 237:G 236:A 235:C 234:C 233:C

S1d
-2.2
118:U 119:A 120:A
246:A 245:U 244:U

S1e
-2.1
116:A 117:G
249:U 248:C


P6
P6a
-14.6
252:U 253:A 254:G 255:G 256:U 257:G 258:G 259:G
274:A 273:U 272:C 271:C 270:A 269:C 268:U 267:C


P7
P7a
-5.9
293:G 294:U 295:C 296:U
304:U 303:A 302:G 301:G

P7b
-7.6
289:G 290:C 291:U 292:G
311:C 310:G 309:A 308:C


S2
S2a
-4.300000000000001
113:G 114:U 115:G
314:C 313:A 312:C


P8
P8a
-9.9
316:C 317:U 318:G 319:G 320:A
337:G 336:A 335:C 334:C 333:U


9
9a
7.8
39:C 340:U 341:C 342:C
50:G 349:A 348:G 347:G


3
3a
5.5
6:U 57:G 58:C
56:A 355:C 354:G

3b
4.300000000000001
2:C 53:A 54:C
59:G 358:U 357:G

3c
2.1
8:C 49:U
61:G 360:G


10
10a
8.5
75:U 376:G 377:G 378:G 379:C
88:G 387:U 386:C 385:C 384:G

10b
7.6
68:U 369:G 370:C 371:A
93:A 392:C 391:G 390:U

10c
2.1
65:U 366:A 367:U
99:G 398:U 397:A


4
4a
6.699999999999999
5:G 46:G 47:C
96:C 395:C 394:G

4b
9.1
9:G 40:C 41:G 42:G
03:C 402:G 401:C 400:C


11
11a
6.3
```

Check correctness of predicted helicies

8.

- Check correctness for each compound helix predicted
- Determines if the size of the compound helix is correct
- Determines if the helices within each compound helix is correct
- In order for a helix to be correct, every base pair within a helix must be correct

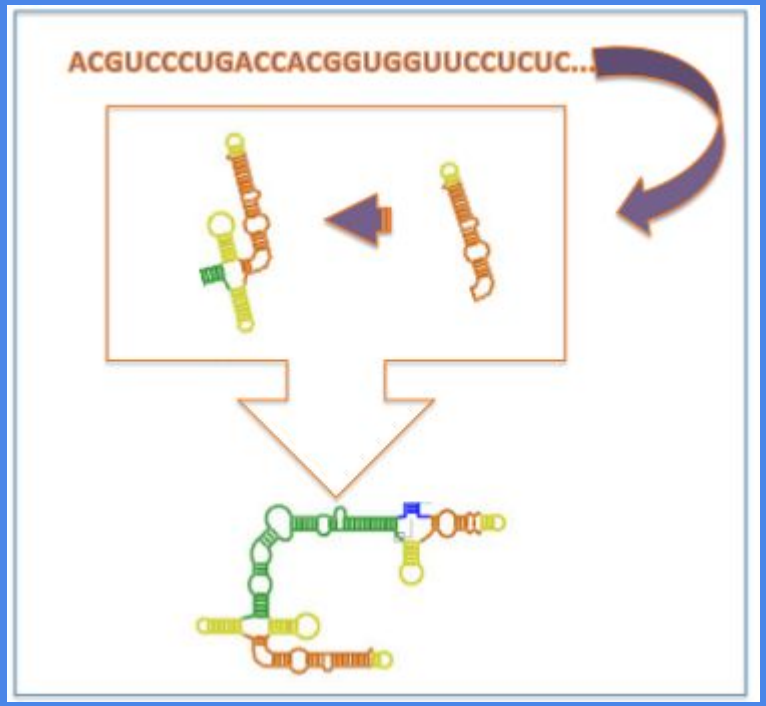


Materials and Methods - Algorithm

- What about predicting PIs? 4 main steps...
  - **1:** Divide the sequence into 2-15 nucleotide windows and attempt to pair with each other to get all potential helices from 2 to 15 bps in length
    - Very expensive but thorough
      - ~ 270,000 potential PI helices predicted for E. coli 16s rRNA
        - (only 32 PI helices in known structure)
  - **2:** Reduce the PIs based on various criteria and then extend the remaining helices to try and determine which are correct
      - Helices with hairpin loops <= 15 nucleotides in length are selected from the total pool
      - Any helices with dG values > 0 are removed
      - dG values are then modified by hairpin loop size and nucleotide identity in loop
        - e.g. GNRA, UUCG, CUUG motifs are more common and thus presumably more stable in RNA secondary structure
  - **3:** The resulting helices are ranked by the modified energies to prioritize extension with extension algorithm
  - **4:** Use the elongation algorithm on PIs to determine which structures will ultimately remain

Materials and Methods

- Now how do you go from sequence to structure?



Methods II:
The Software

Materials and Methods - Software

- STEP 1:
  - Get .fasta file for sequence of interest
    - .fasta is a file with a header line beginning with '>' that describes the sequence, which follows the header line
    - 3 ways to get .fasta for this project:
      - extract from .bpseq via Comparative RNA Web http://www.rna.icmb.utexas.edu/DAT/
      - use GenBank Nucleotide DB search script
      - create by hand

```
1  >Organism=Escherichia coli, Accession Number=V00336.1, Sequence=5s-rRNA
2  TGCCTGGCGGCCGTAGCGCGGTGGTCCCACCTGACCCCATGCCGAACTCAGAAGTGAAACGCCGTAGCGCCGATGGTAGTGTGGGGTCTCCCCATGCGAGAGTAGGGAACTGCCAGGCAT
3
```

Materials and Methods - Software

- STEP 2 (optional):
  - Get .bpseq file for sequence
    - .bseq is a file containing the known base pairings within a secondary structure for a sequence
    - they are produced by the Gutell Lab and are available via the Comparative RNA web http://www.rna.icmb.utexas.edu/DAT/
    - may not be available for every sequence
    - optional because it is only required for comparing predictions with actual known helices - accuracy checking

```
Filename: d.5.b.E.coli.nopbpseq
Organism: Escherichia coli
Accession Number: V00336
Citation and related information available at http://www.rna.ccbb.utexas.edu
1 U 119
2 G 118
3 C 117
4 C 116
5 U 115
6 G 114
7 G 113
8 C 112
9 G 111
10 G 110
11 C 0
12 C 0
13 G 0
14 U 0
15 A 0
```

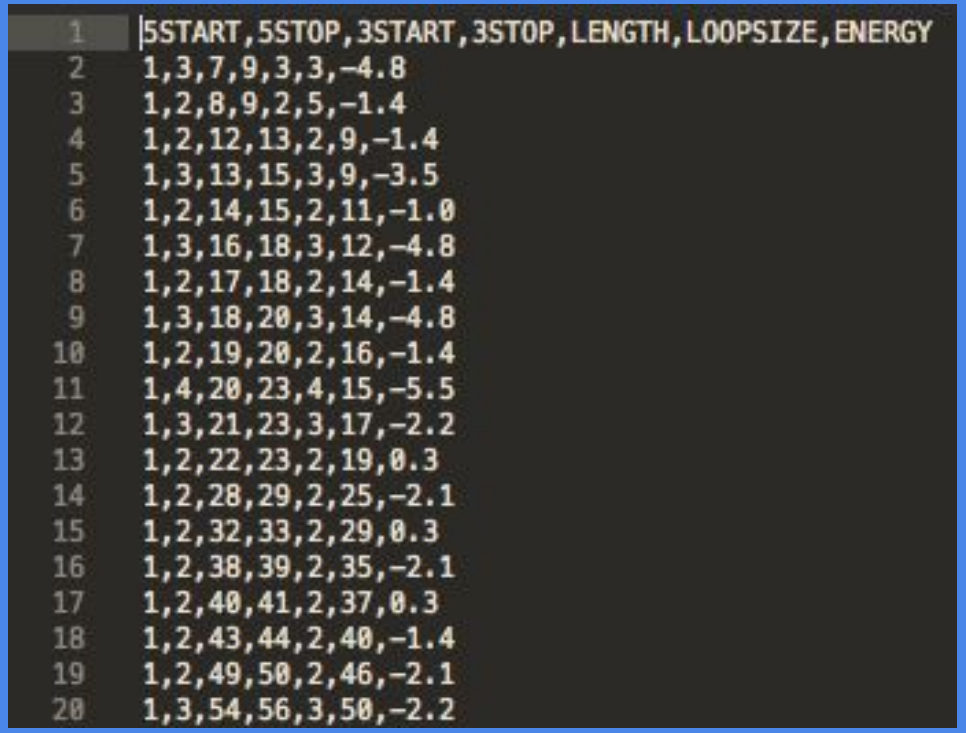
Materials and Methods - Software

- STEP 3 (optional):
  - Get .piesie file for sequence
    - .piesie is a tab-delimited file containing the Pie-Sie helix annotations for the structures formed from a sequence
    - they are also produced by the Gutell Lab
    - 2 options for getting .piesie:
      - use automated script that converts .bpseq -> .piesie
      - make by hand

	NAME	5START	5STOP	3START	3STOP	TYPE
2	1-P1a	9	13	21	25	PI
3	2-P2a	76	82	87	93	PI
4	2-P2b	73	75	95	97	PE
5	2-P2c	69	70	98	99	PE
6	2-P2d	66	68	101	103	PE
7	2-P2e	61	63	104	106	PE
8	3-P3a	153	158	163	168	PI
9	3-P3b	144	147	175	178	PE
10	4-P4a	184	186	191	193	PI
11	5-P5a	206	207	212	213	PI
12	5-P5b	198	203	214	219	PE
13	6-S1a	136	142	221	227	SI

Materials and Methods - Software

- STEP 4:
  - Get .energetics file for sequence
    - .energetics is comma-separated-value file containing information for all possible helices that can form using the input sequence and constraints
      - current options are:
        - minimum loop size
        - maximum loop size
        - minimum helix length
        - maximum helix length
    - must be produced by script
      - script can take .fasta or a manually entered sequence as inputs



```
5START,5STOP,3START,3STOP,LENGTH,LOOPSIZE,ENERGY
1,3,7,9,3,3,-4.8
1,2,8,9,2,5,-1.4
1,2,12,13,2,9,-1.4
1,3,13,15,3,9,-3.5
1,2,14,15,2,11,-1.0
1,3,16,18,3,12,-4.8
1,2,17,18,2,14,-1.4
1,3,18,20,3,14,-4.8
1,2,19,20,2,16,-1.4
1,4,20,23,4,15,-5.5
1,3,21,23,3,17,-2.2
1,2,22,23,2,19,0.3
1,2,28,29,2,25,-2.1
1,2,32,33,2,29,0.3
1,2,38,39,2,35,-2.1
1,2,40,41,2,37,0.3
1,2,43,44,2,40,-1.4
1,2,49,50,2,46,-2.1
1,3,54,56,3,50,-2.2
```

Materials and Methods - Software

- STEP 5:
  - Get .predicted file for sequence
    - .predicted is comma-separated-value data frame containing information for all predicted helices resulting from the prediction pipeline, sorted by lowest to highest modified energy
    - requires .fasta and .energetics to produce
    - will annotate designate structures is a .piesie is supplied as well
    - is formatted so that it can be analyzed directly with other data-manipulating software, e.g. Excel, MATLAB, R

```
1   5START,5STOP,3START,3STOP,LENGTH,LOOPSIZE,ENERGY,5SEQ,3SEQ,LOOPSEQ,LOOP_SCORE,E_MOD,IN_MODEL
2   79,86,90,97,8,3,-15.6,UGUGGGG,CCCAUGC,UCU,0.0,-5.2,1
3   81,86,90,95,6,3,-13.4,UGGGG,CCCAU,UCU,0.0,-4.46666666667,0
4   80,86,90,96,7,3,-13.1,GUGGGG,CCCAUG,UCU,0.0,-4.36666666667,0
5   82,86,90,94,5,3,-12.0,GGGG,CCCA,UCU,0.0,-4.0,0
6   10,14,18,22,5,3,-10.7,CCGU,CGGU,AGC,0.0,-3.56666666667,0
7   83,86,90,93,4,3,-9.9,GGG,CCC,UCU,0.0,-3.3,0
8   11,14,18,21,4,3,-8.2,CGU,CGG,AGC,0.0,-2.73333333333,0
9   79,85,91,97,7,5,-12.3,UGUGGG,CCAUGC,GUCUC,0.0,-2.46,0
10  84,86,90,92,3,3,-6.6,GG,CC,UCU,0.0,-2.2,0
11  102,105,110,113,4,4,-5.6,UAG,UGC,GGAA,2.5,-2.025,0
12  81,85,91,95,5,5,-10.1,UGGG,CCAU,GUCUC,0.0,-2.02,0
13  80,85,91,96,6,5,-9.8,GUGGG,CCAUG,GUCUC,0.0,-1.96,0
14  69,71,75,77,3,3,-5.8,CC,GU,GAU,0.0,-1.93333333333,0
15  20,23,28,31,4,4,-7.6,GUG,ACC,GUCC,0.0,-1.9,0
```

Materials and Methods - Software

- Examples of 5s PI helix predictions data manipulation:
  - viewing the 2 helices that are in the model

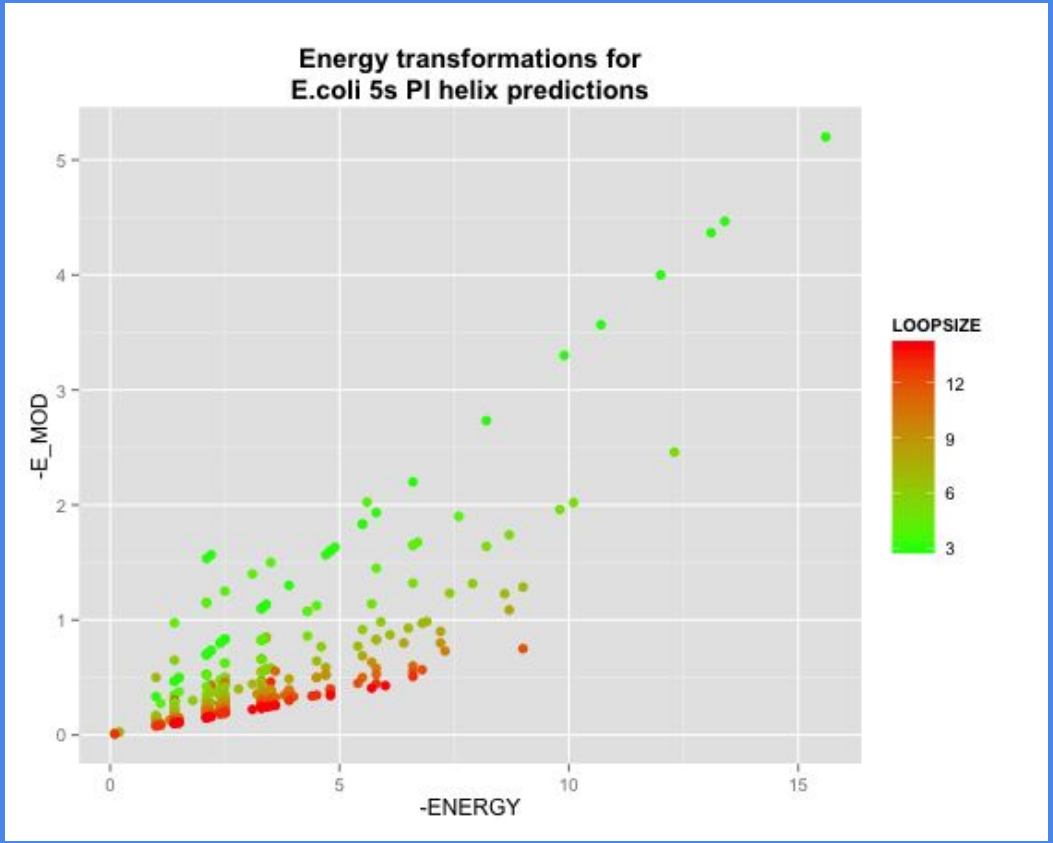
```
> df[df$IN_MODEL == 1,]
    X5START X5STOP X3START X3STOP LENGTH LOOPSIZE ENERGY   X5SEQ   X3SEQ        LOOPSEQ LOOP_SCORE      E_MOD IN_MODEL
1        79     86      90     97      8        3  -15.6 UGUGGGG CCCAUGC           UCU          0 -5.2000000        1
117      31     34      48     51      4       13   -6.6     UGA     CAG CCCCAUGCCGAAC          0 -0.5076923        1
```

  - viewing the helices with hairpin loops of size 10

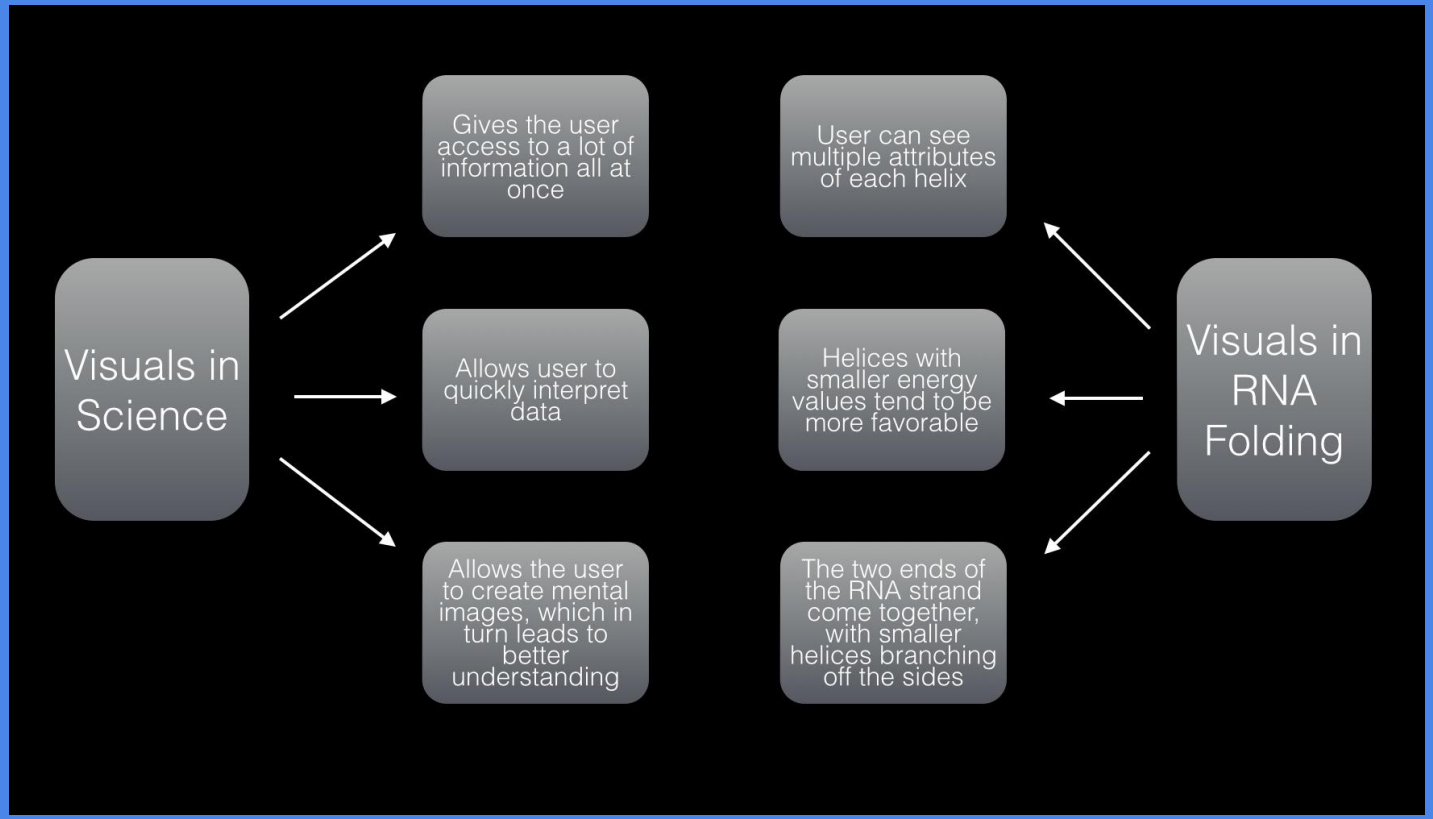
```
> df[df$LOOPSIZE == 10,]
    X5START X5STOP X3START X3STOP LENGTH LOOPSIZE ENERGY X5SEQ X3SEQ    LOOPSEQ LOOP_SCORE E_MOD IN_MODEL
80        9     12      23     26      4       10   -7.3   GCC   GUC GUAGCGCGGU        0.0 -0.73        0
99       10     12      23     25      3       10   -5.8    CC    GU GUAGCGCGGU        0.0 -0.58        0
119      67     68      79     80      2       10   -2.5     C     U GCCGAUGGUA        2.5 -0.50        0
152      67     69      80     82      3       10   -3.9    CG    GU CCGAUGGUAG        0.0 -0.39        0
153      68     70      81     83      3       10   -3.9    GC    UG CGAUGGUAGU        0.0 -0.39        0
175      11     12      23     24      2       10   -3.3     C     G GUAGCGCGGU        0.0 -0.33        0
176      23     24      35     36      2       10   -3.3     G     C UCCCACCUGA        0.0 -0.33        0
215      69     70      81     82      2       10   -2.5     C     U CGAUGGUAGU        0.0 -0.25        0
233      28     29      40     41      2       10   -2.1     A     G CCUGACCCCA        0.0 -0.21        0
262      75     76      87     88      2       10   -1.5     G     C UAGUGUGGGG        0.0 -0.15        0
```

Materials and Methods - Software

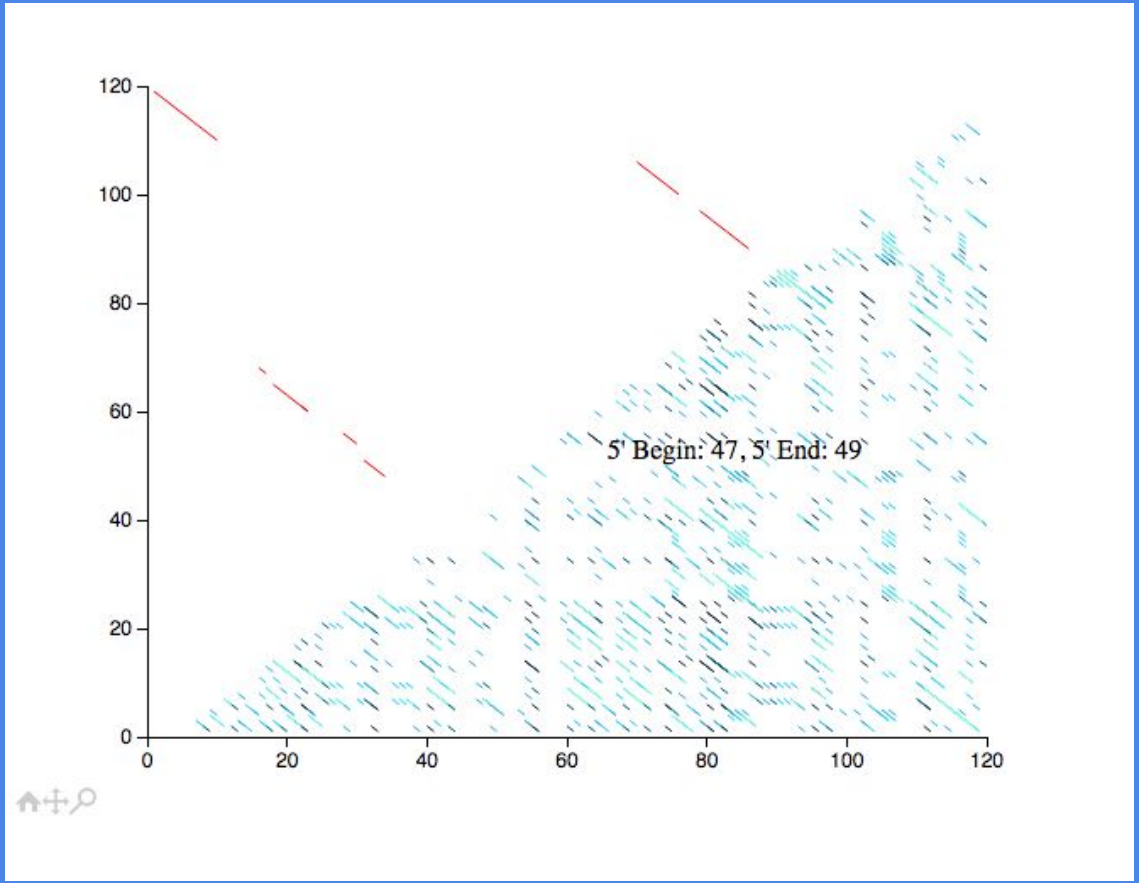
- Examples continued:
  - Example of directly plotting resulting data frame in R
  - plot displays modified energies of the predicted helices vs. original energy
  - colored by hairpin loop size to demonstrate the effect loop size has on new modified energies



Materials and Methods - Software



Materials and Methods - Software



Materials and Methods - Software

Goal

- Create a visual that shows all of the possible and correct helices
- Make the visual interactive, allowing the user to view specific helices based on size, energetics, etc

Materials and Methods - Software

Option 1
Build on program from previous class

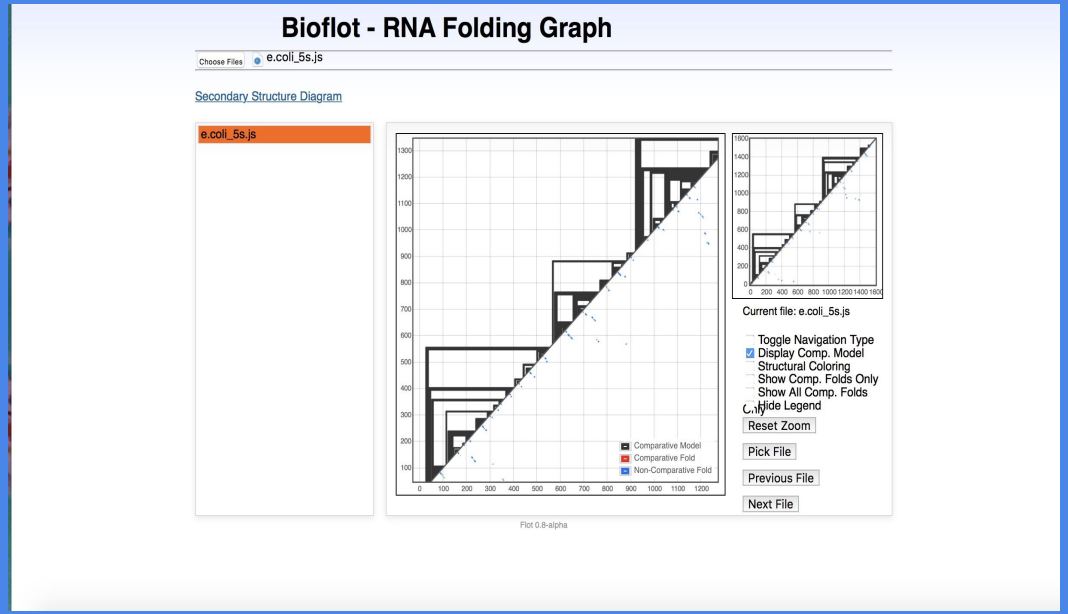
Pros:

1. Start with a solid base
2. Can spend time adding different functions

Cons:

1. Don't know javascript
2. Code is difficult to follow
3. Creates better base for future

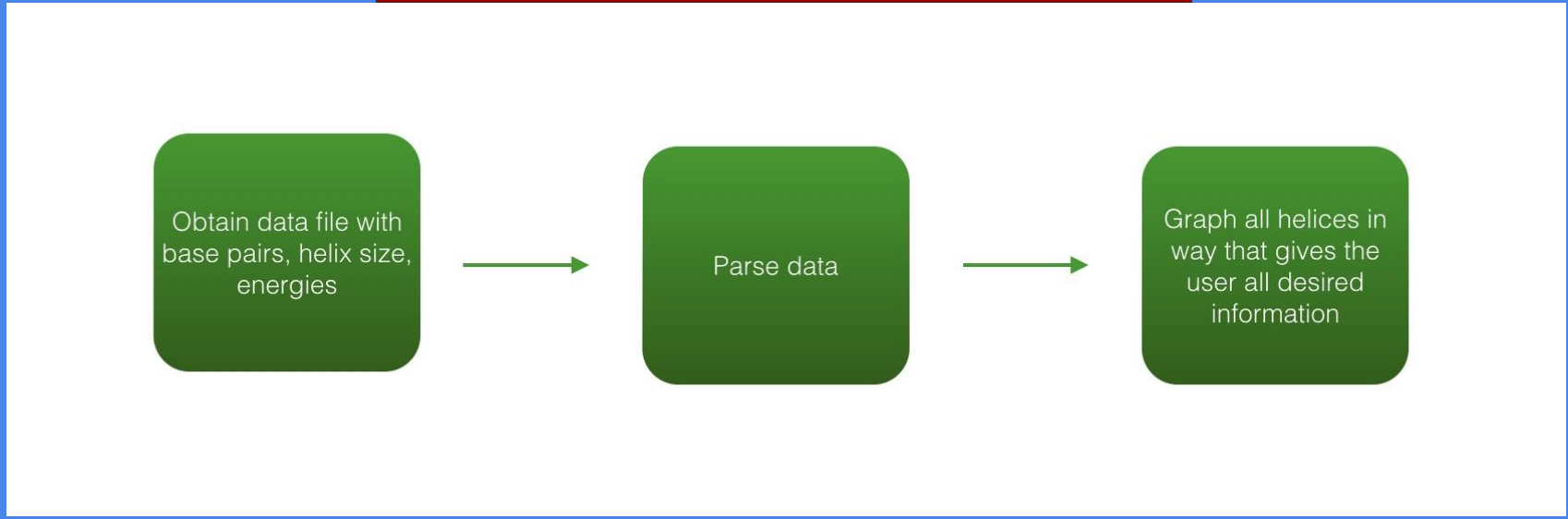
Materials and Methods - Software



- Each point represents a base pair
- The smaller triangles represent the pairs that formed first
- Gives the user options but doesn't give specific information about each base pair

Materials and Methods - Software

Option 2
Start from scratch



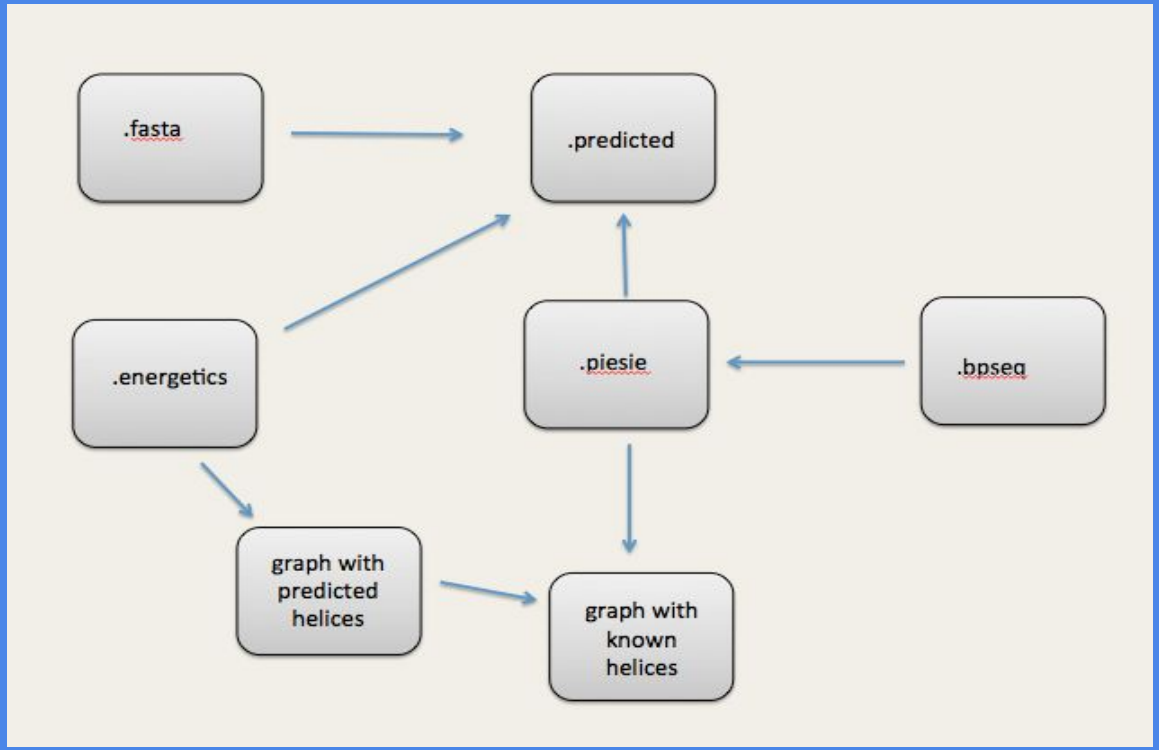
Materials and Methods - Software

Resources

RNAF.py Program	Matplotlib	MPLD3
Program from previous class - outputs all *possible* helices	Python graphics library - well documented	Python package - combines the graphing ability of Matplotlib and interactive features of Javascript

Materials and Methods - Software

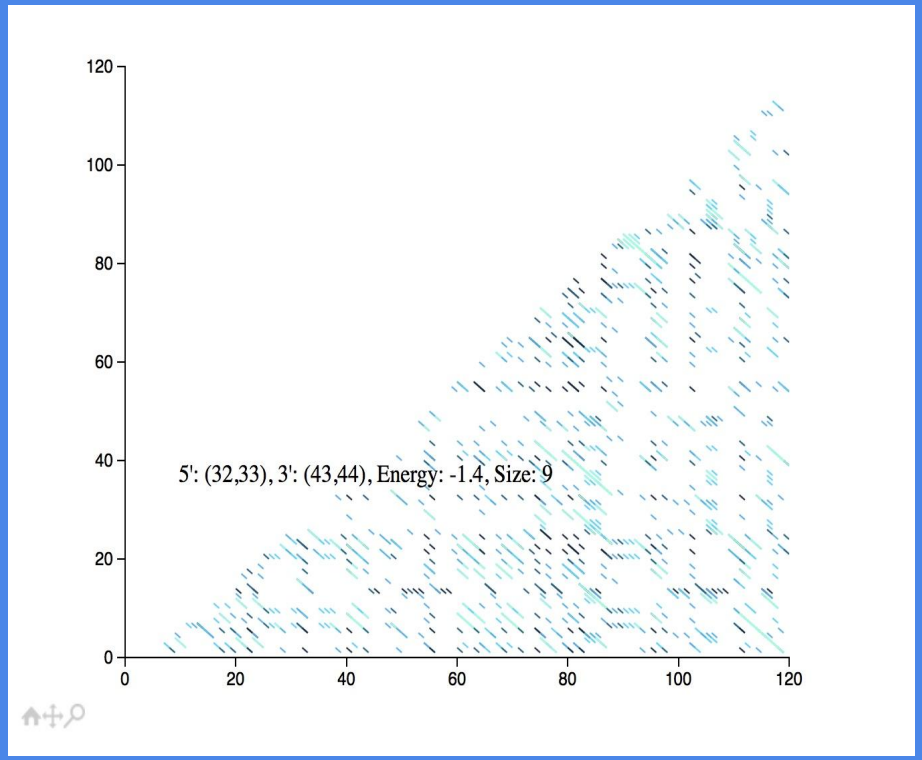
- Software dependencies visualization:

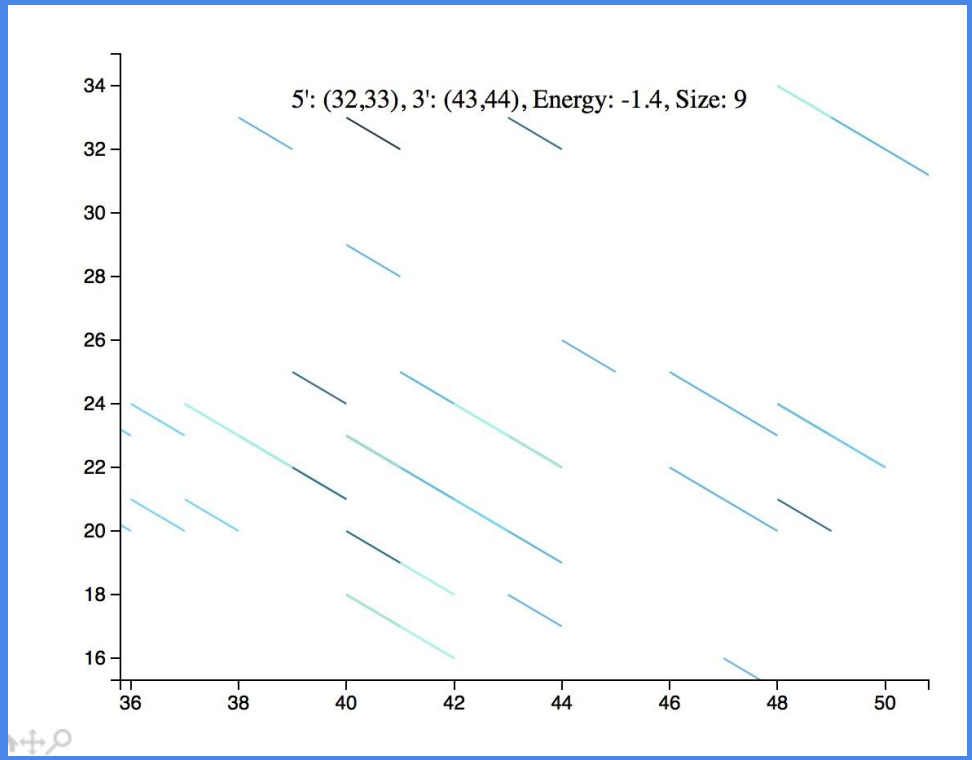


Results I: Visualization

Results - Visualization

Hover Tool

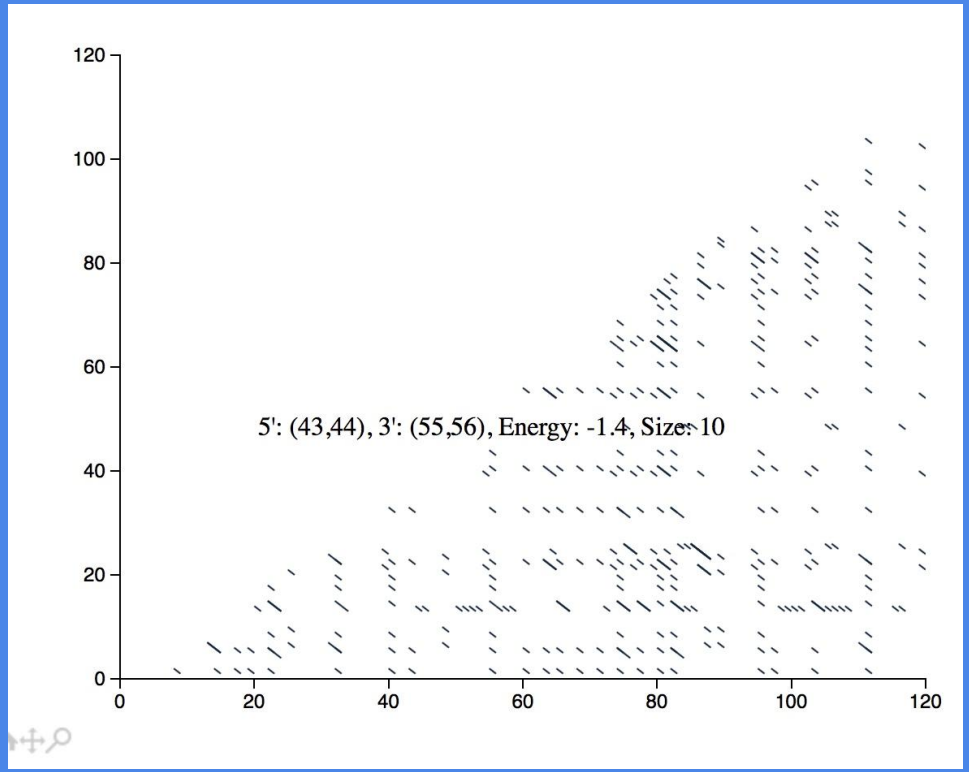




Results - Visualization

Sorting Options

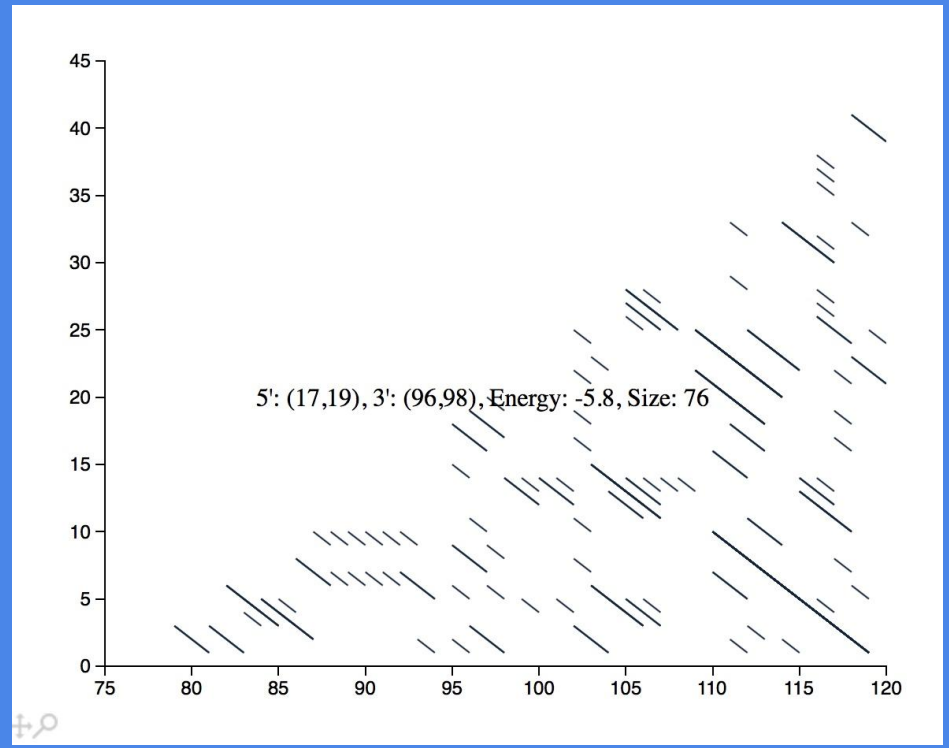
```
genAllHelices_5-15-14 — python rnaplot.py ecoli.in — 80×24
Last login: Thu Dec 10 17:47:28 on ttys000
Kathryns-MBP:~ kathrynfields$ cd Downloads/genAllHelices_5-15-14/
Kathryns-MBP:genAllHelices_5-15-14 kathrynfields$ python rnaplot.py ecoli.in
Would you like to sort by energy or length? Enter 'None' to view all: energy
Enter a number: -2
Would you like to view helices with energies greater than or less than this valu
e? greater
Serving to http://127.0.0.1:8888/    [Ctrl-C to exit]
127.0.0.1 - - [10/Dec/2015 17:52:02] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2015 17:52:02] "GET /d3.js HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2015 17:52:02] "GET /mpld3.js HTTP/1.1" 200 -
```



Results - Visualization

Sorting Options

```
genAllHelices_5-15-14 — python rnaplot.py ecoli.in — 80×24
Last login: Thu Dec 10 17:51:47 on ttys000
Kathryns-MBP:~ kathrynfields$ cd Downloads/genAllHelices_5-15-14/
Kathryns-MBP:genAllHelices_5-15-14 kathrynfields$ python rnaplot.py ecoli.in
Would you like to sort by energy or length? Enter 'None' to view all: length
Enter a number: 75
Would you like to view helices longer or shorter than this number? greater
Serving to http://127.0.0.1:8888/    [Ctrl-C to exit]
127.0.0.1 - - [10/Dec/2015 17:54:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2015 17:54:41] "GET /d3.js HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2015 17:54:41] "GET /mpld3.js HTTP/1.1" 200 -
```



Results II: Software Accomplishments

Results - Software Accomplishments

- Nussinov-Jacobsen Implementation
  - Implemented in the same way many other folding algorithms are implemented
  - Recursion with traceback step
  - Usually $O(n^6)$
  - Slow for large sequences
- Extension algorithm - over 1000 lines of code!
  - Java software implementation
  - generates statistics for a given sequence of RNA and its PieSie file
  - Outputs predicted helices for a RNA sequence
  - Compares prediction results with correct secondary structure

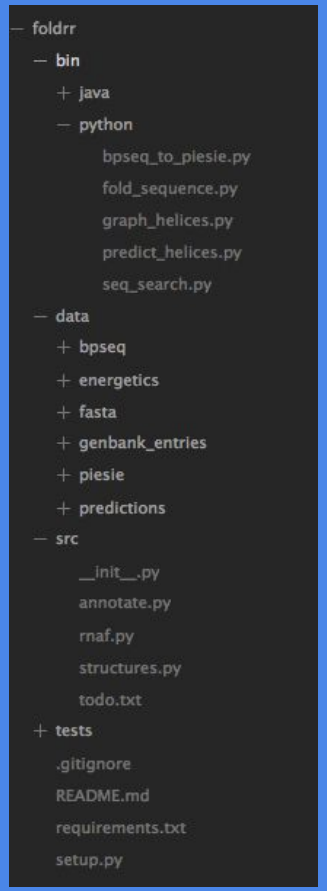
Nussinov - Jacobsen Implementation

RNAsequence.txt | NussinovJacobsonAlgorithm.java

```
import java.util.Scanner;

public class NussinovJacobsonAlgorithm {
	/* This program takes an input RNA sequence and finds the optimal base pairs.
	 * It then outputs the optimal number of base pair locations.
	 */

	public static void main(String []args) throws FileNotFoundException {

		//read in input file with RNA sequence and get length of seq
		String sequence = fileIn();
		int len = sequence.length();

		//creates a matrix of size len x len with zeros
		int [][] matrix = new int [len][len];

		for (int i = 0; i < len; i++) {
			for (int j = 0; j < len; j++) {
				matrix[i][j] = 0;
			}
		}

		//traverse the matrix and replace zeros with results of algorithm
		//for each diagonal from 1 to len
		for (int diagonal = 1; diagonal < len; diagonal++) {
			//for each row from 0 to (N-diagonal)
			for (int row = 0; row < len - diagonal; row++) {
				int col = row + diagonal;
				int w = isBasePair(sequence.charAt(row), sequence.charAt(col));
				int v1 = matrix[row + 1][col - 1] + w;
				int v2 = matrix[row + 1][col];
				int v3 = matrix[row][col-1];
				int v4 = getMaxK(matrix, row, col);
				matrix[row][col] = Math.max(Math.max(v1, v2), Math.max(v3, v4));

			}
		}

		findBasePairs(matrix, sequence, 0, len-1);

G(0) base pairs with C(5)
G(1) base pairs with C(4)
U(6) base pairs with A(12)
U(7) base pairs with A(11)
```

Results - Software Accomplishments

- foldrr
  - Python RNA folding software application - over 1500 lines of code!
  - complete with automated build files, licensing, and documentation
  - localized in Bitbucket remote repository for sharing and version control
    - https://bitbucket.org/ashtoncb/foldrr
  - performs all functionality (besides extension of PIs and SIs in Java)
    - .fasta retrieval from GenBank Nucleotide DB
    - Convert .bpseq to .fasta
    - Convert .bpseq to .piesie annotation file
    - Predict potential helices and .energetics file generation
    - Filtering potential helices to select PI helices
    - Formats potential helices for Java extension algorithm
    - Prepares final predicted structures in .predictions file for data analysis
    - Graphs the visualization component
    - and more!

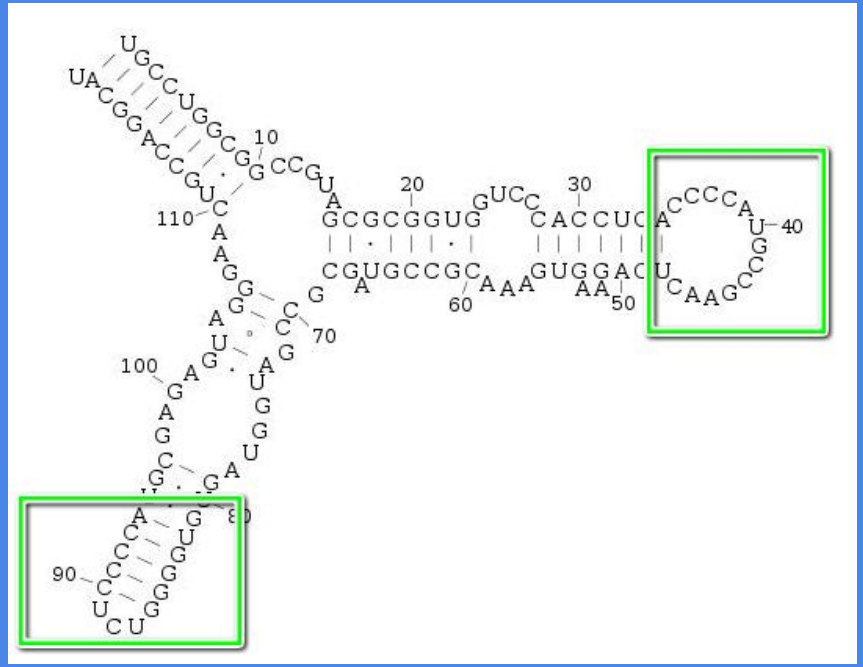


Results III:

Secondary Structure Prediction

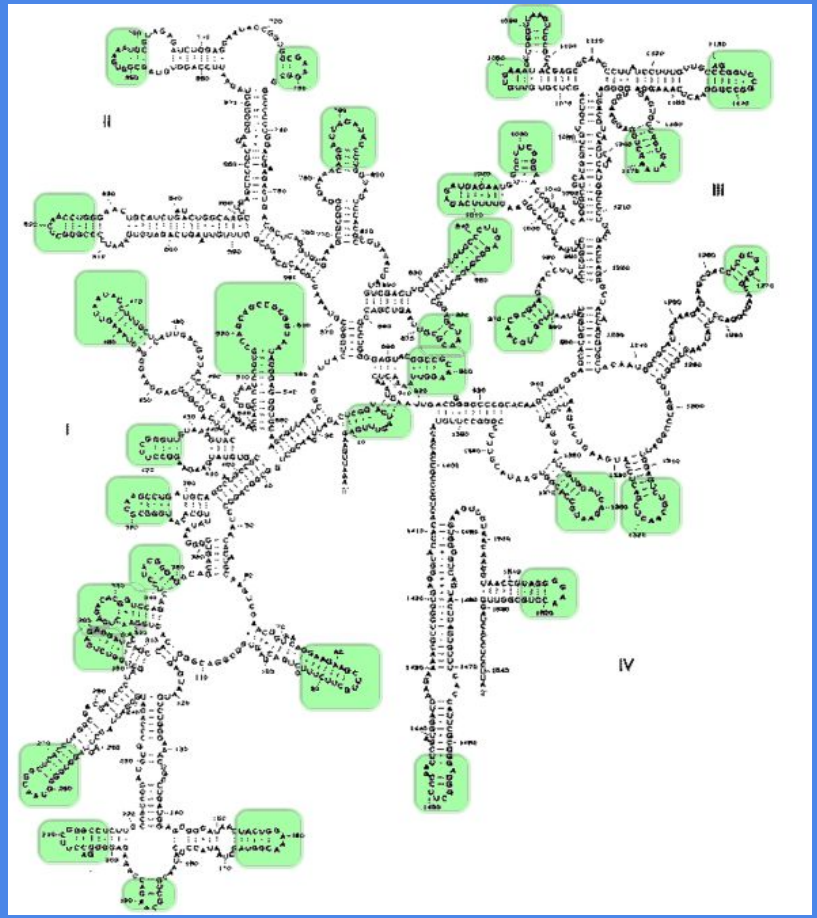
Results - PI Predictions

- E. coli 5s rRNA:
  - Length of sequence:
    - 120
  - Number of predicted PI structures:
    - Before filter: 1490
    - After filter: 296
    - Decrease: 80.13%
  - Number of actual PI structures predicted:
    - 2 of 2



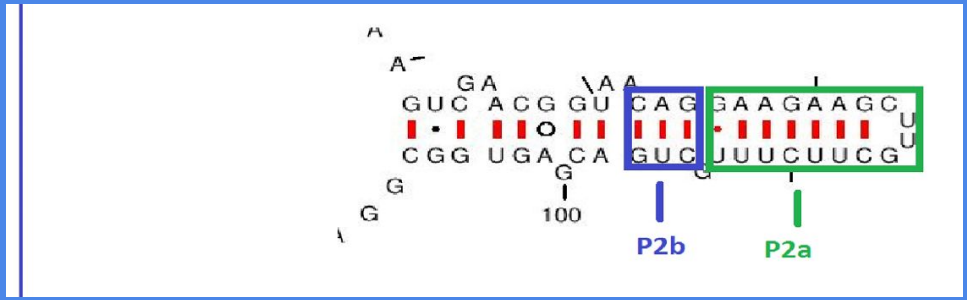
Results - PI Predictions

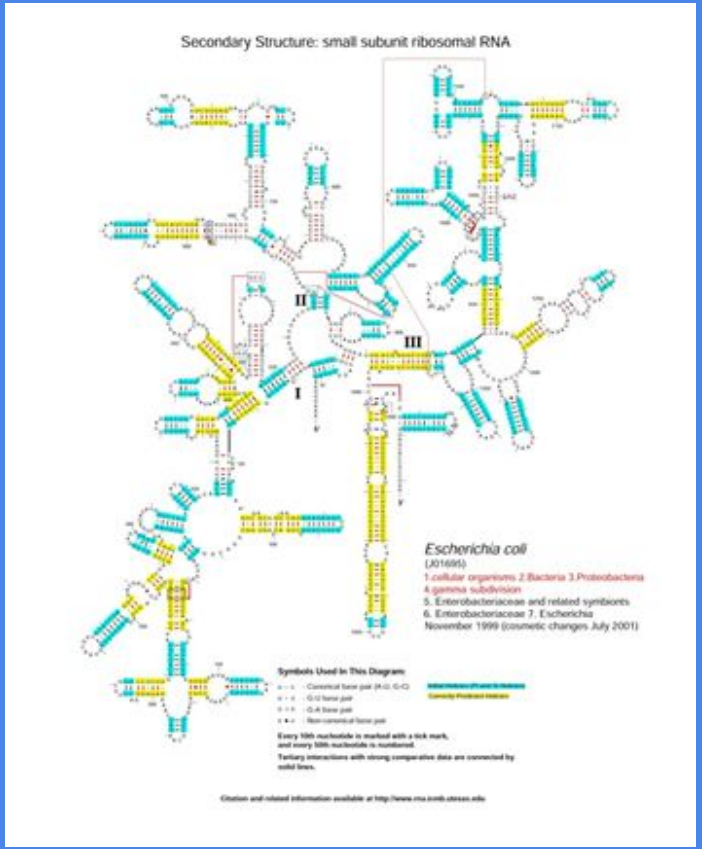
- E. coli 16s rRNA:
  - Length of sequence:
    - 1542
  - Number of predicted PI structures:
    - Before filter: 268870
    - After filter: 4068
    - Decrease: 98.49%
  - Number of actual PI structures predicted:
    - 32 of 32



Results - Elongation

- Primary and Secondary Elongation Results
  - 54% of helices predicted correctly
    - A helix is considered correct when every base pair within a helix is correct
    - Off by one error considered incorrect
  - 90% of base pairs predicted correctly
    - Total number of base pairs in E. Coli 16S secondary structure = 711





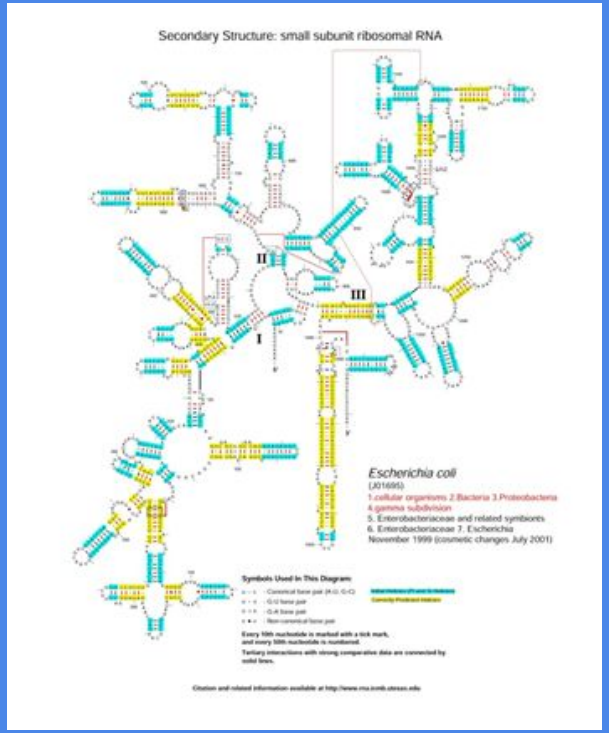
Results - Overall Table

Structure Type:	**E. coli 5s rRNA** (150 bp)	**E. coli 16s rRNA** (1542 bp)
Primary Initiation (PI)	Actual helices: 2 Predicted helices: 296 Correct helices: 2 Correct nucleotides in correct structures: 40	Actual helices: 32 Predicted helices: 268870 Correct helices: 32 Correct nucleotides in correct structures: 325
Primary Elongation (PE)	Actual helices: 4 Predicted helices: 3 Correct helices: 3 Correct nucleotides in correct structures: 54	Actual helices: 33 Predicted helices: 31 Correct helices: 13 Correct nucleotides in correct structures: 424
Secondary Initiation (SI)	Actual helices: 1 Predicted helices: 0 Correct helices: 0 Correct nucleotides in correct structures: 0	Actual helices: 18 Predicted helices: 0 Correct helices: 0 Correct nucleotides in correct structures: 0
Secondary Elongation (SE)	Actual helices: 0 Predicted helices: 0 Correct helices: 0 Correct nucleotides in correct structures: 0	Actual helices: 17 Predicted helices: 18 Correct helices: 9 Correct nucleotides in correct structures: 264
Total	Total % of helices correctly predicted: 71.43% Total % nucleotides correctly predicted: 62.67%	Total % of helices correctly predicted: 54.00% Total % nucleotides correctly predicted: 65.69%

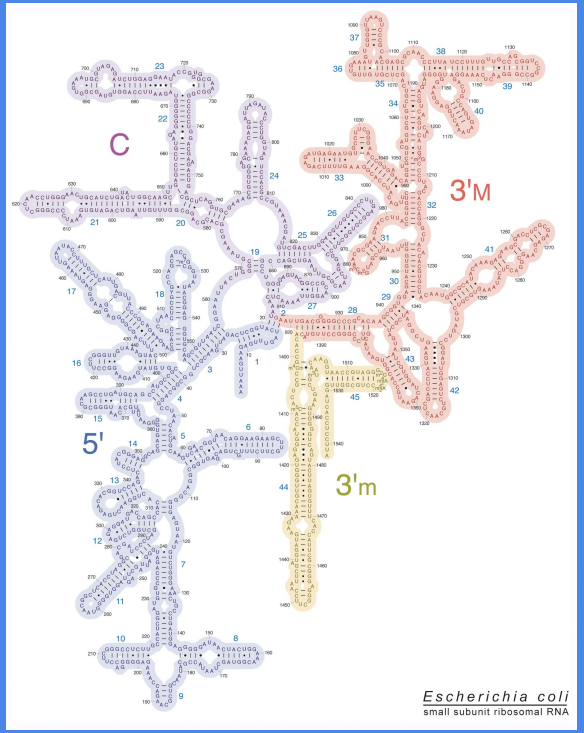
Results - Comparison

- E. Coli 16s rRNA overall predictions

Simple Distance First Algorithm



Correct Structure



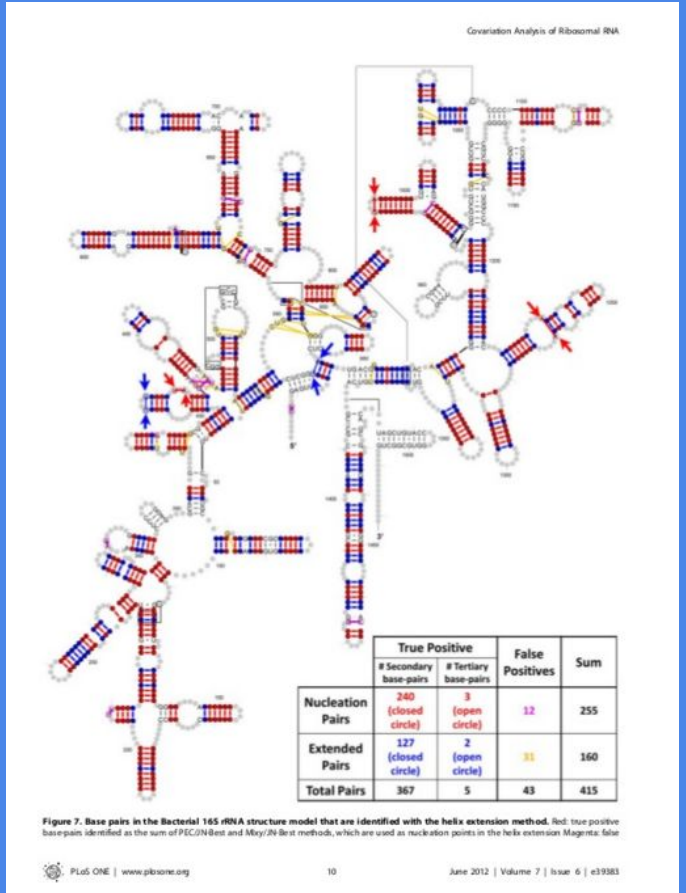
http://rna.ucsc.edu/rnacenter/images/figs/ecoli_16s.jpg

Future Goals

Future Goals

- Extension algorithm:
  - Improve accuracy of predicting elongation of compound helices
  - Develop a method to stop extension for SI predictions to occur
- PI/SI predictions:
  - Further narrow down the PI predictions that are made and
  - Extend from our predicted PI helices instead of the known PI helices
  - Try to predict the SIs
- Overall algorithm:
  - After extending from our Pi helices, develop a method to predict overall structure based on the resulting compound structures
- Visualization:
  - Improve hover tool, work on new graph

Questions?



	True Positive		False Positives	Sum
	# Secondary base-pairs	# Tertiary base-pairs		
Nucleation Pairs	240 (closed circle)	3 (open circle)	12	255
Extended Pairs	127 (closed circle)	2 (open circle)	31	160
Total Pairs	367	5	43	415

Figure 7. Base pairs in the Bacterial 16S rRNA structure model that are identified with the helix extension method. Red: true positive base-pairs identified as the sum of PEC/JN-Best and Mixy/JN-Best methods, which are used as nucleation points in the helix extension Magenta: false

End