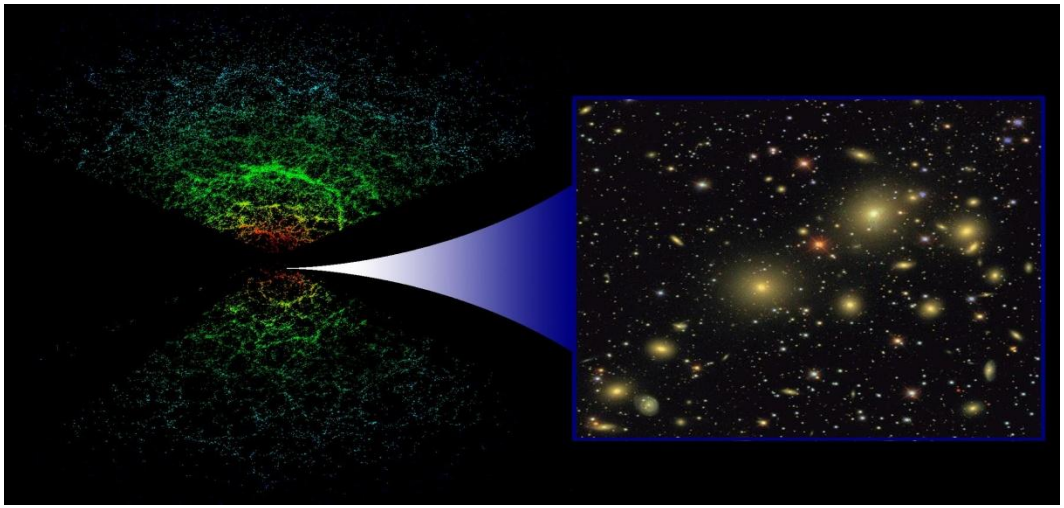
Inverse RNA Folding
Week 6

Amarendra Badugu
Tom Hyland
Elena Sizikova

Summary

- RNA Molecules
  - Folding Problem
    - RNA Folds
  - **Inverse Folding**
    - Given a structure, find the Sequence that it would fold into at a given temperature



Genetic Algorithm- Mimics Evolution

1. Generate Sequences (Initialisation)
2. Crossover substructures between individuals (Crossover)
3. Mutate positions(Mutation)
4. Select Fit individuals from overall population (selection)
5. Repeat from 1-4 until optimum solution is reached (Termination)

Our Implementation

- Single Target Structures
  - Generate Random Sequences. Base Pair according to Target structure. Fold them using Vienna Fold.
  - Mutate at Positions that differ
  - different types of Crossover between optimum substructures
  - Mix the mutants and crossovers with parents and pick the top individuals allowing some diversity or pick random individuals
  - Using various distance metrics to quantify how close two structures are .

Our Implementation

- Vienna fold takes majority of the computation time. We parallelized the function calls to increase performance by more than 10 fold.
- Initialisation works quite well picking up target structures in the initial guess
- A easy to use GUI with lots of parameters
- we have a large degree of control over our parameters allowing us to fine tune the performance

Our Implementation

- **Multiple Target Structure**
  - We want to design sequences that fold into different target structures at different temperatures (e.g at 28°C an 37°C)
  - Or perhaps we want to design a sequence that has the same target structure at our two different temperatures.
  - We can adapt our current genetic algorithm to deal with multiple target structures.

Chains of Dependencies

One Target Structure (. . . . .)

- For a single target structure only the positions that form a base pair depend on one another.
- So in this case just the first and last positions depend on one another.

Two Target Structures ((. . . .)) (. . .) (. . . . .) . (. . .)

- For 2 target structures there can be longer chains of dependencies.
- Position 1 and 8 must for a base pair in target structure 1. But position 1 must also form a base pair with position 7 in the second target structure. And position 7 in the target structure must form a base pair with position 2 in the first target structure.
- So we have a dependency chain 2 -> 7-> 1-> 8

Initialization

- Two target structures means we can have long dependency chains.
- Our Initialisation takes into account these dependency chains.
- We will do this by assigning random strings of complementary bases to our dependency chains.
- This saves us a lot of time as we are only considering solutions that have a chance of folding into the target structure.
- Chain 2->7->1->8, A->U->G->C

Initial Sequence= GA????UC???

Selection

We have used the same Elite selection method as before.

Our distance metric is based on the Zuker distance for single target structures.

Added punishment for sequences that could be too stable.

We also favour sequences that are close to matching both rather than just one.

a=ZukerDist from Target 1

b=Zuker Dist from Target 2

c=0.2 (if target strcutures are different)

Distance=(1+c) (a+b + (mod(a-b)/(a+b)))

Mutation

- Our previous mutation method no longer works due to the base chain dependencies mentioned before.
- We have adapted our mutation approach to deal with this problem.
- We pick a dependency chain at random and regenerate it int the same way as the initialization stage.

Mutation Example

- Target 1=(....)(....), Target 2=((........))
- Sequence AAGCAUGACAUU
- Folded at 20°C (….)....., Folded at 37°C (…….)
- Chains=[(7->12->1->6), (2->11),3, 4, 5, 8, 9, 10]
- Pick a chain to mutate (7->12->1->6) from G->U->A->U to A->U->G->C, new sequence GAGCACAACAUU

Crossover

- Our single target structure method extends well to two target structures.
- We simply find the lists of suitable crossover points, in the same way as before, for each target structure. We will then choose a crossover point at random from those points which occur in both lists.
- The probabilities of picking a particular crossover pointed are weighted so that crossovers that contain about 50% of the bases from each structure are most favorable
- Finally we perform our crossover in the usual way at the chosen point.

Benchmarking the Single Target Version

The Short Conclusion

We need to improve our technique to get higher success rates than MODENA

A little more detail...

- For comparing our results to that of other algorithms, we have used two data sets:
- 29 Rfam database structures, coming from the different RNA families
- A simulated dataset (ViennaGen) that was created by taking a list of sequences and folding them using Vienna RNA fold

Results

- The simulated dataset was not very useful in distinguishing performance of different methods as a lot of sequences were too short and hence ModeNA solved 100% of the structures,

while our algorithm gave correct results to between 90% and 95% sequences

Rfam Data

- MoDeNa solved 15/29 = 52% of the structures
- Our best run was 10/29 = 34% of the structures

Some Graphs

Conclusion:

We would probably benefit a lot by diversifying the fitnesses of the population, hence being able to make wiser choices on what the next generation will look like

One possible approach to this would be to use free energy and Boltzmann probabilities more extensively.

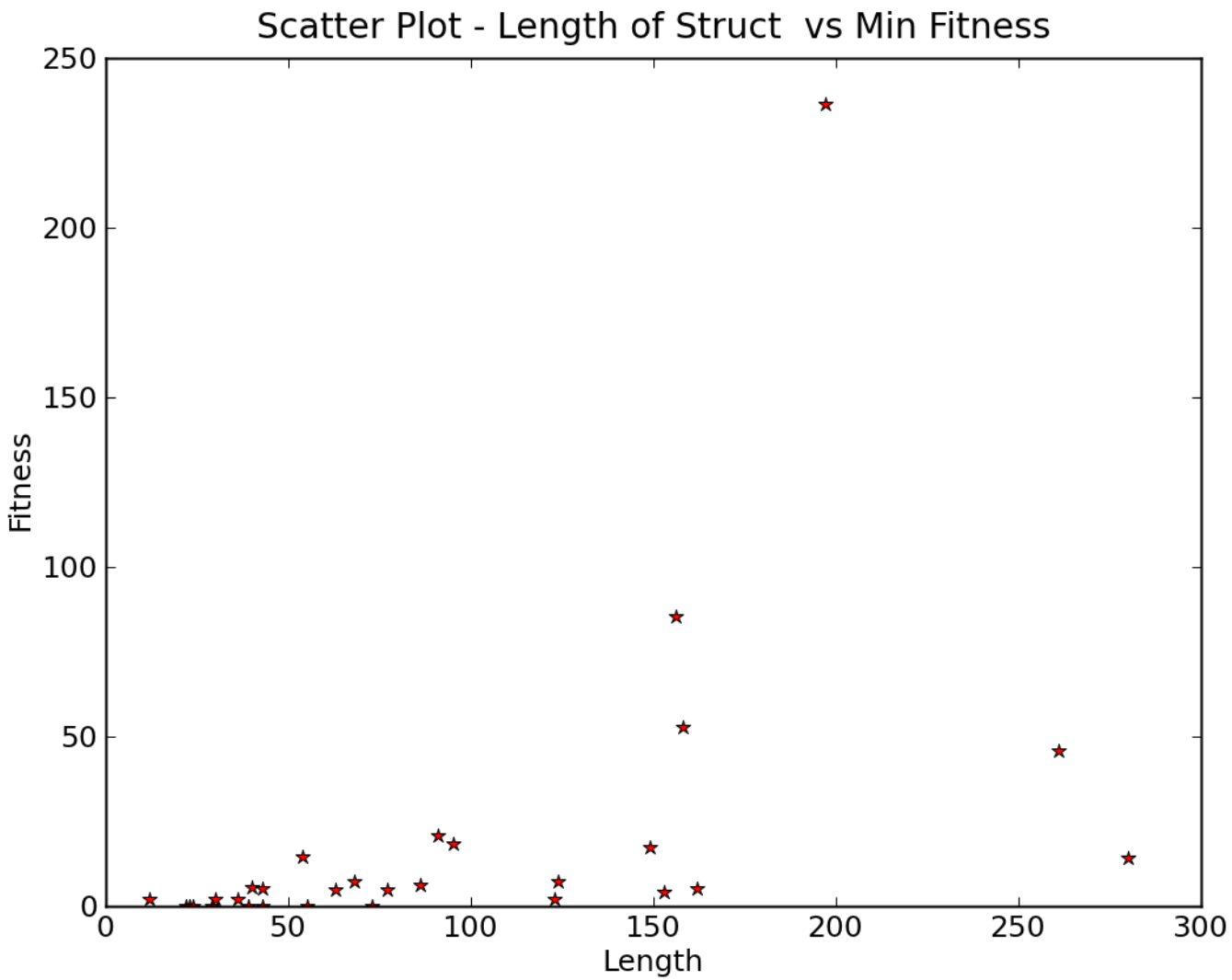
Benchmarks for Multi Target:

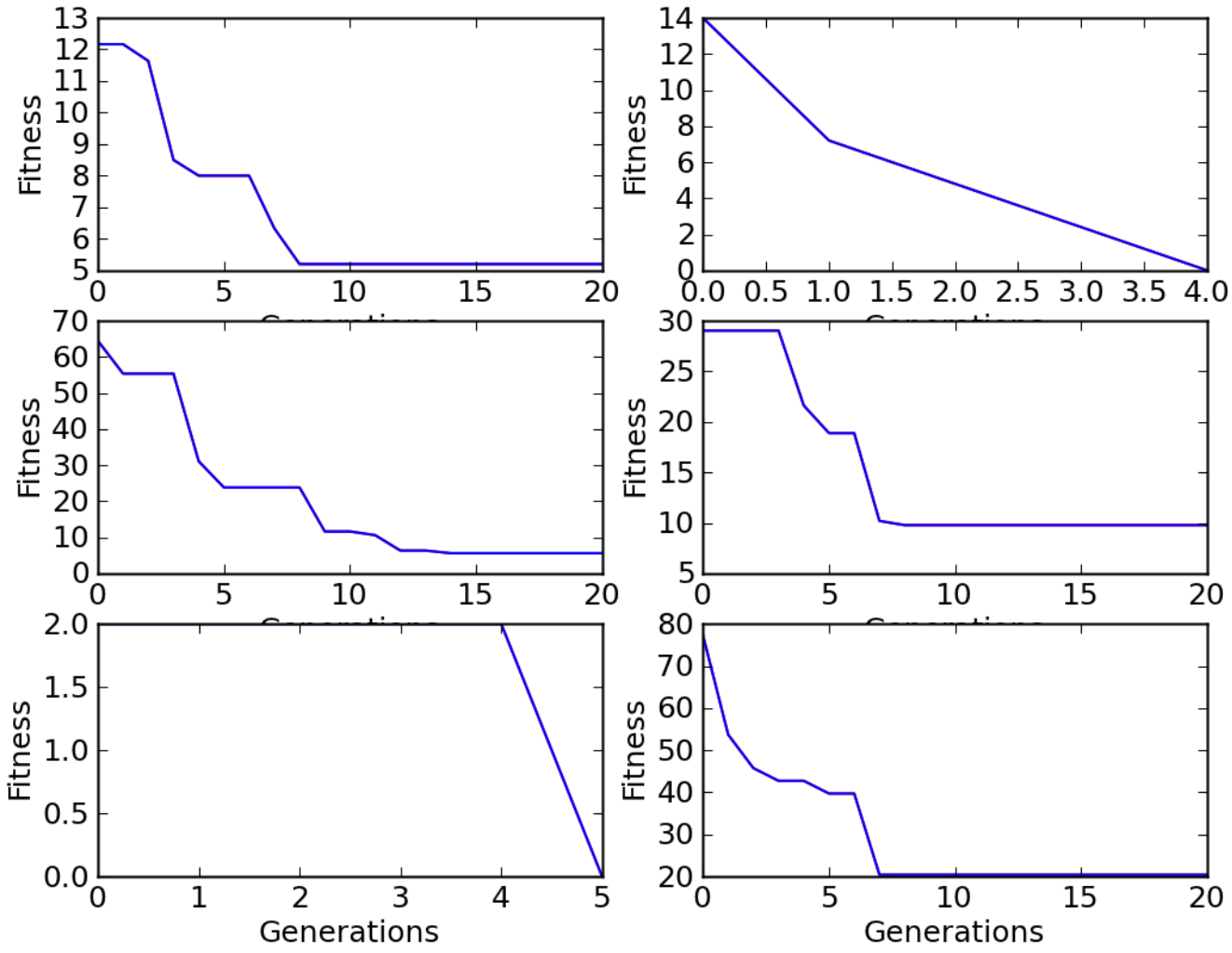
•Total Structures received from the Regulatory RNA group ~ 5000

• At 10:04 am, 19-08-11, after an estimated run time of 19 hours:

422 Correct out of 1127 with many more being very close to final structure

• Custom Dataset picking 30 sequences from the 5000 Structures with various scenarios for benchmarking purpose, we get about 30 % correct.


Scatter Plot - Length of Struct vs Min Fitness
Fitness
Length
0
50
100
150
200
250
300


Fitness
Generations

Acknowledgments:

Dr. Rune Lyngsoe
James Anderson
Dr. Adam Novak
Professor Jotun Hein
Dr. Steve Kelly for the server
Regulatory RNA Team for the Folding data

Questions