ScaDS.AI
DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Topic: **Synthetic structured RNA - playing with genetic algorithms**

Speaker: **Christopher Klapproth**

Supervisor: **Prof. Peter F. Stadler**

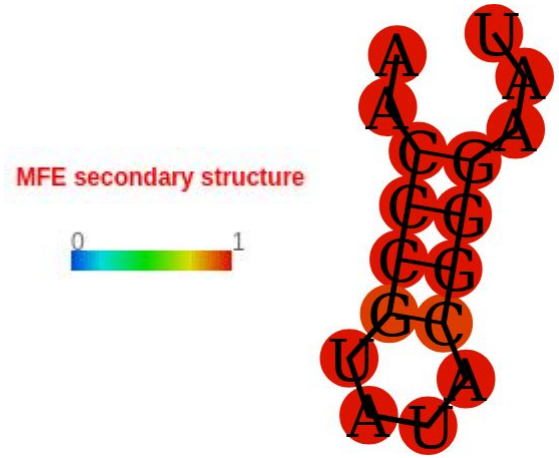
Contact:
christopher@bioinf.uni-leipzig.de

SACHSEN Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

A problem one might have

3’-AACCCGUAUACGGGAAU-5’

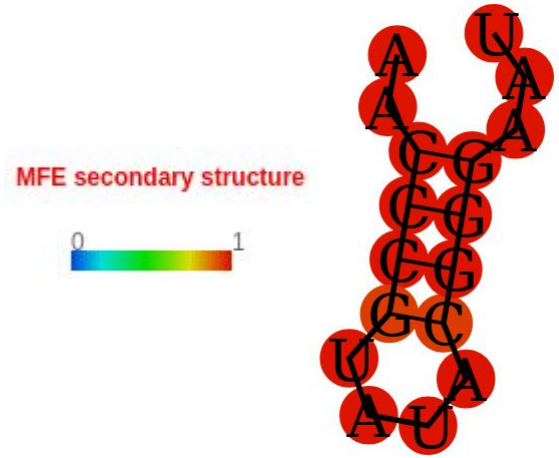
RNAfold, etc….

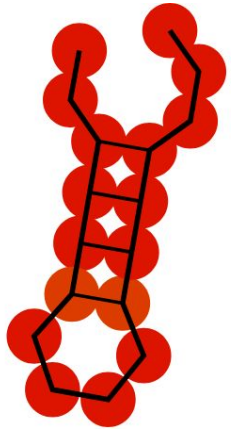


A problem one might have

3’-AACCCGUAUACGGGAAU-5’

RNAfold, etc….





Some form of secondary structure encoding

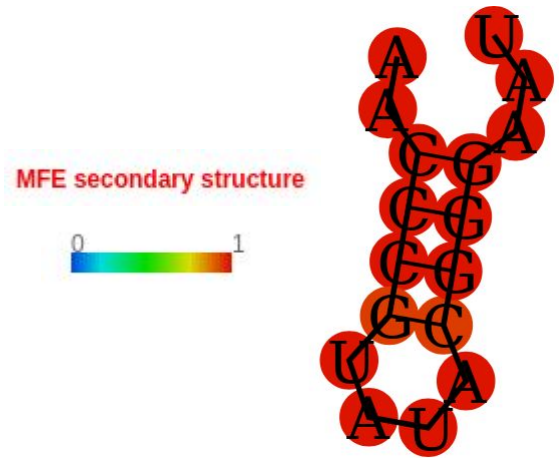
RNA design

3’-AACCCGUAUACGGGAAU-5’

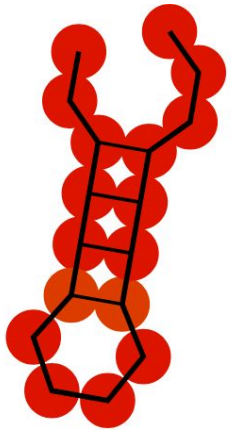
A problem one might have

3’-AACCCGUAUACGGGAAU-5’

RNAfold, etc….



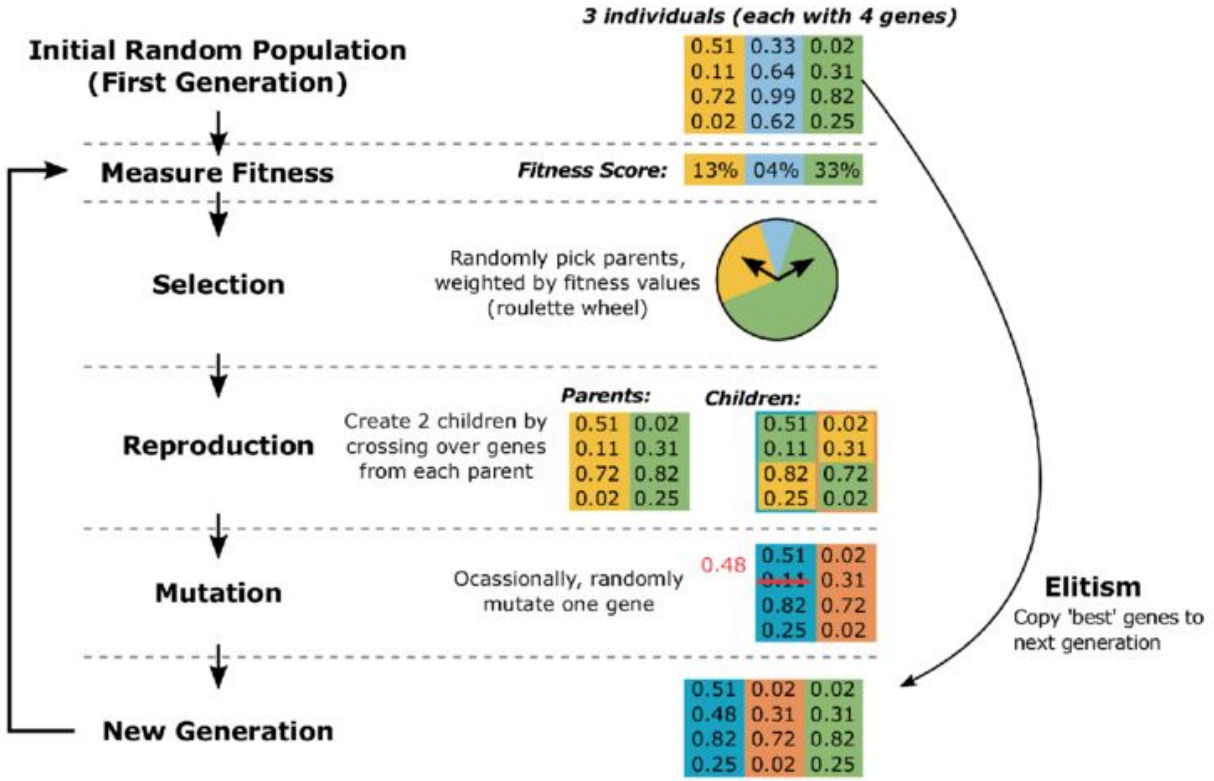

```
>s1
AACCGGAUAUGGCCAA
>s2
AUCCGGUAUACCGGAA
>s3
UAGGGUAAUUACCCAA
```



Structurally aligned synthetic RNA

3’-AACCCGUAUACGGGAAU-5’

Genetic algorithms - a quick recap



- **Basic concept:** Iterative optimization of a population using a fitness function
- Each step - or generation - randomly mutates some entities
- Creation of the next population pool is based on selecting the fittest entities and cross-breeding them

Figure edited from: Woodward, R. I., & Kelleher, E. J. (2016). Towards 'smart lasers': self-optimisation of an ultrafast pulse source using a genetic algorithm. *Scientific reports*, *6*(1), 37616.

Genetic algorithms - rules applied to RNA sequences

Population *P* of *N* nucleotide sequences in generation *g*:

$$P = \{s_1, s_2, \ldots, s_n\}, 1 < n \leq N$$

Nucleotide sequence of length *L* as single item in Population:

$$s_i = AUCGGACG, L_i = 8$$

Single Nucleotide as smallest mutable unit:

$$x_j \in \{A, U, C, G\}, 0 \leq j < L_i$$

'Intuitive' Genetic operators:

- **Point mutation**
- **Insertion**
- **Deletion**

- **Example:**
  With a population of 100 sequences of length 50, the explorable space (if one assumes constant length) has a cardinality of

  $$4^{50} \times 100 \sim 1.27 \times 10^{32}$$

  ...making a complete traversion practically impossible.

Genetic algorithms - Gene Operators

Point Mutation

Swapping exactly one randomly selected nucleotide for another one.

Insertion

Adding random nucleotide in an equally randomized position, thereby elongating the sequence by 1.

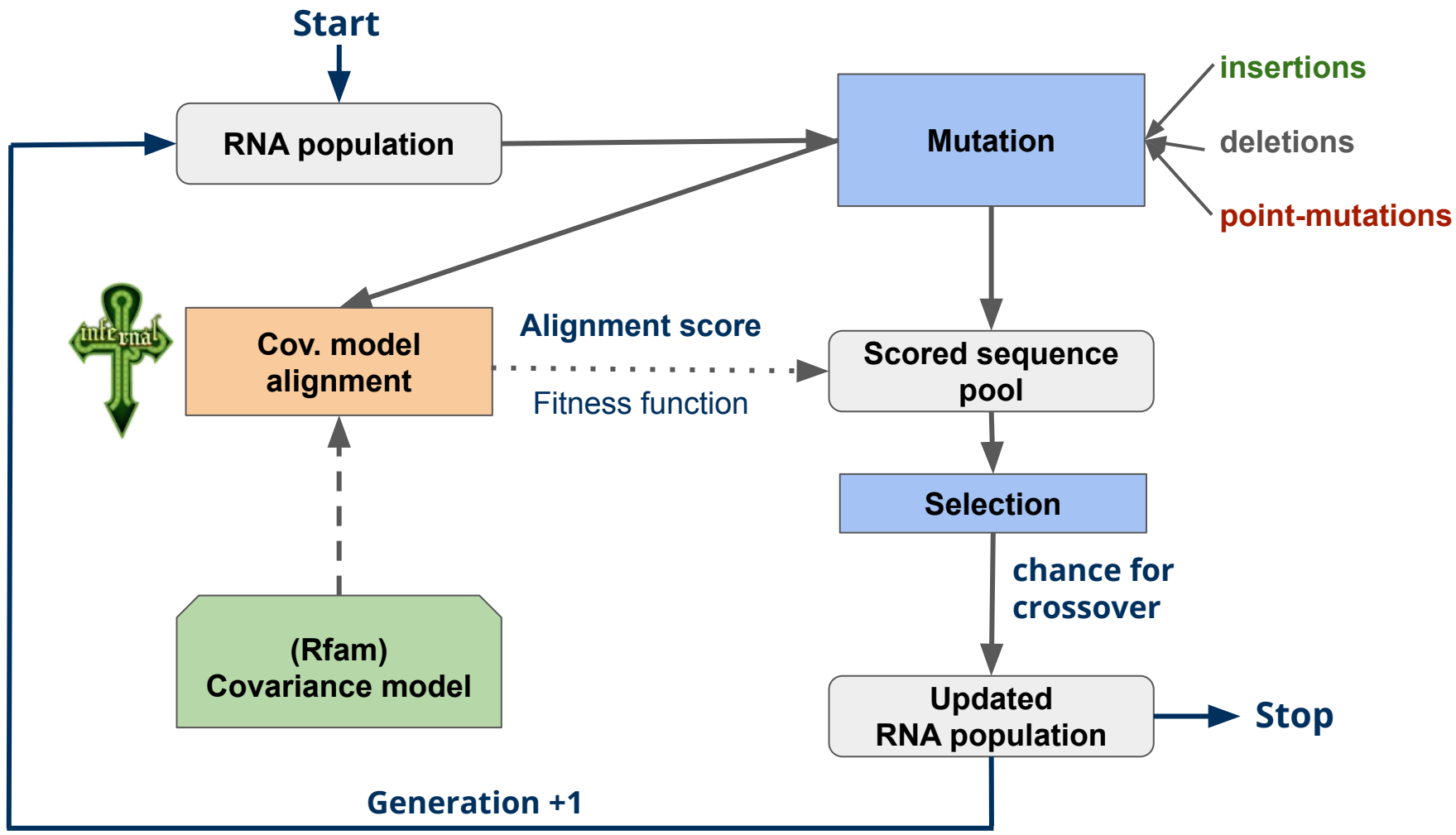
Deletion

Selecting a random sequence index and deleting the nucleotide, thereby shortening the sequence by 1.

Crossover

Exchanging two positions of equal length and identical relative indices between two sequences.

Genetic algorithms - basic implemented workflow



Quick word on covariance model scoring

To align a population of candidate sequence files to a given model, we use:

```
$ cmalign  [model] [sequences]
```

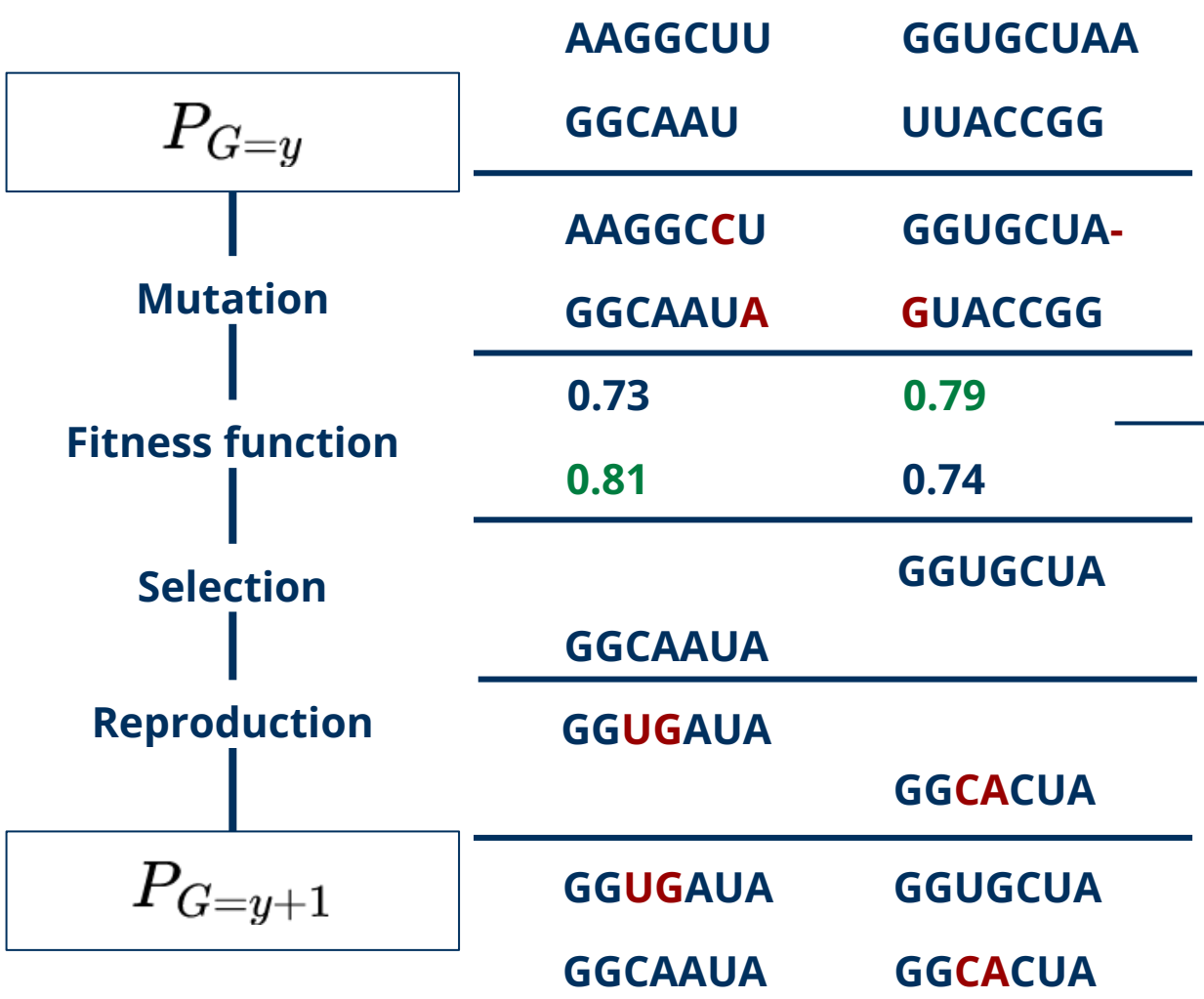
Infernal calculates a posterior probability for each nucleotide aligned to the model

$$pp(s) = \frac{1}{L_s} \sum_{i=0}^{L_s} P(N_i|C)$$

where *S* is the final score for sequence *s* and *L* is the length of the alignment. P(N|C) represents the probability of nucleotide *N* being correctly aligned given covariance model C.

- **Covariance models**
  Models currently used for testing are obtained from the **Rfam** data base.

Genetic algorithms - basic implemented workflow



(Rfam)
Covariance model

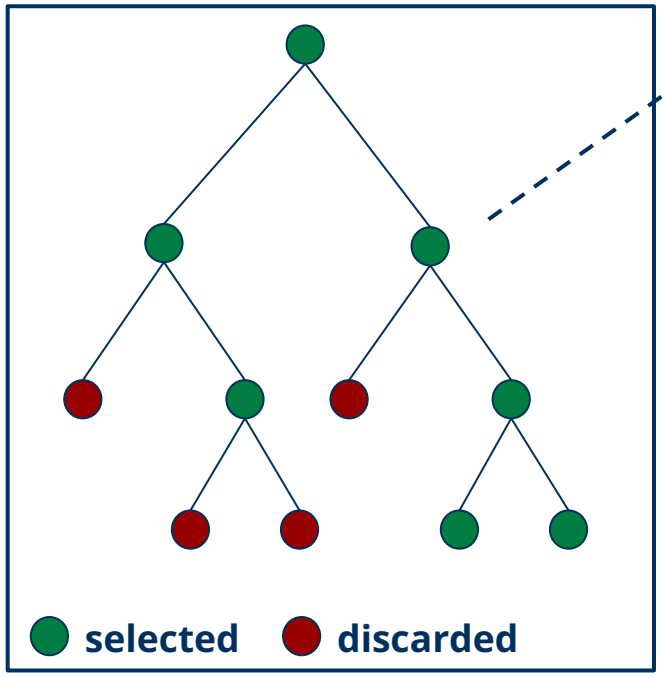
Evorna-struc
Evolve **RNA struc**tures - we will probably find a better name at some point ;)

- Initialize RNA populations
- Run Genetic Algorithm with user-set parameters and constraints
- Visualize results

Github:
https://github.com/chrisBioInf/evorna-struc

Tracking simulated generations as a tree structure

All generated sequences & their properties are saved in a tree-like structure:

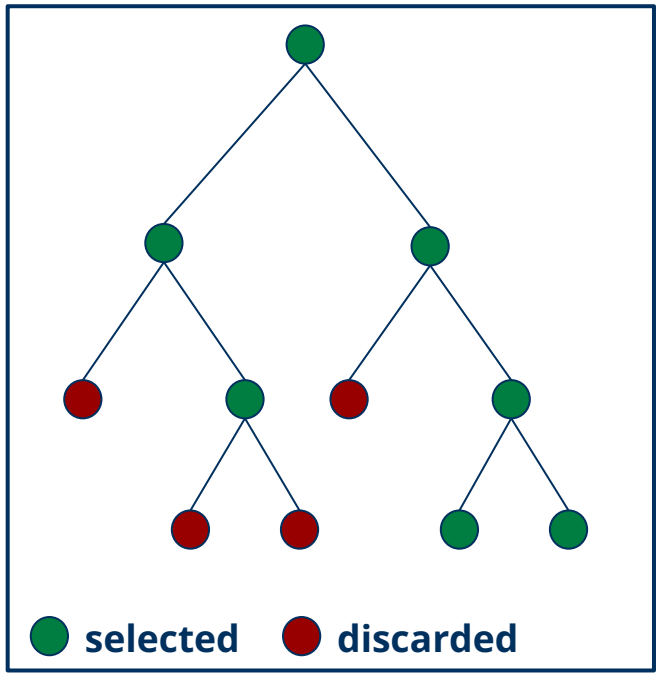


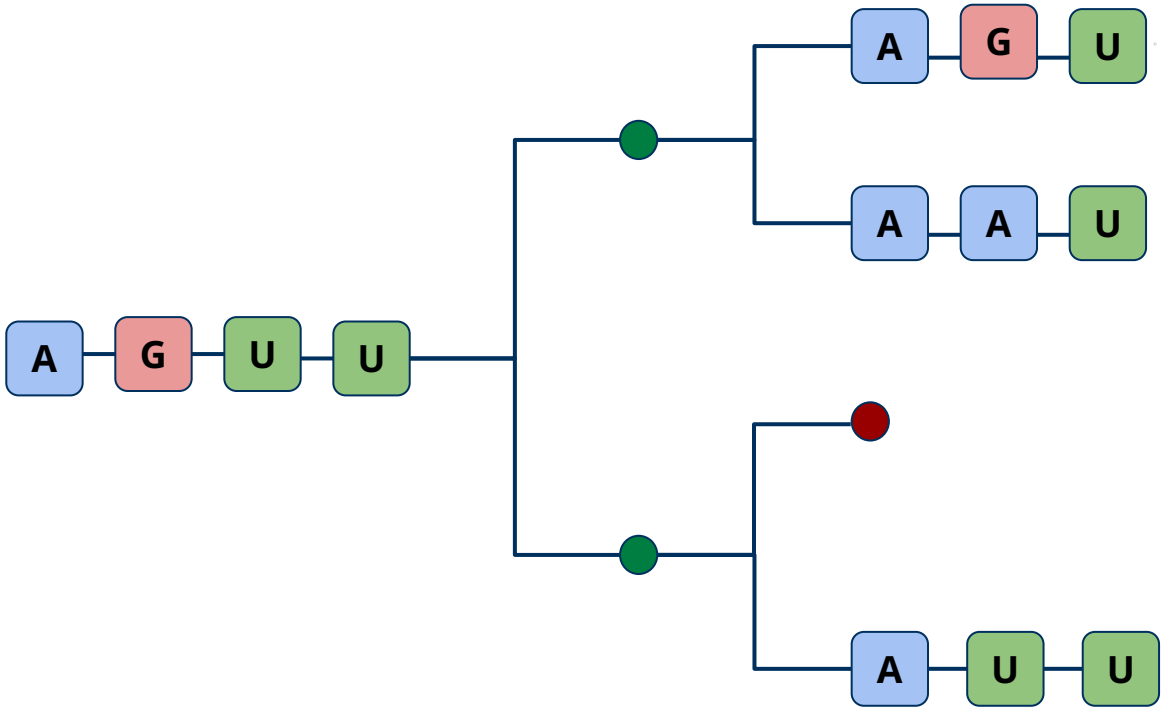

```
{
  "CG_content": 0.6,
  "children": [
    {
      "CG_content": 0.6,
      "children": [
        {
          "CG_content": 0.62,
          "cm_score": 0.43,
          "generation": 2,
          "hash": "810b3654ad025d7fcba0689d80a6a086",
          "id": "s_2_7",
          "mfe": -20.9,
          "seq": "UGTCUGGCGCGUCGGCGGGAGUGCGCCGUAGGGUCGGAUCAUAGACCUAUUCA"
        },
        {
          "CG_content": 0.58,
          "children": [
            {
              "CG_content": 0.58,
              "cm_score": 0.29,
              "generation": 3,
              "hash": "819a3fcd7f69ba59fe9030baef70a7ec",
              "id": "s_3_17",
              "mfe": -13.3,
              "seq": "UGTAAGCCGCGUCGGCGGGAGUGCGCCGUAGGGUCGGAUCAUAGAACUAUUCA"
            },
            {
              "CG_content": 0.57,
              "cm_score": 0.72,
              "generation": 3,
              "hash": "bfc57b3d61c2de2eb358850ee6f3f052",
              "id": "s_3_39",
              "mfe": -13.8,
              "seq": "UGTUCAUCGCGUCGGCGGGAGUGCGCCGUAGGGUCGGAUCAUAGAACUAUUCA"
            }
```


- **Exporting**
  Saved trees can then be accessed to:
1. ...get data in JSON format
2. ...extract sequences
3. ...filter by scores & features
4. trace 'evolutionary history'

Tracking simulated generations as a tree structure

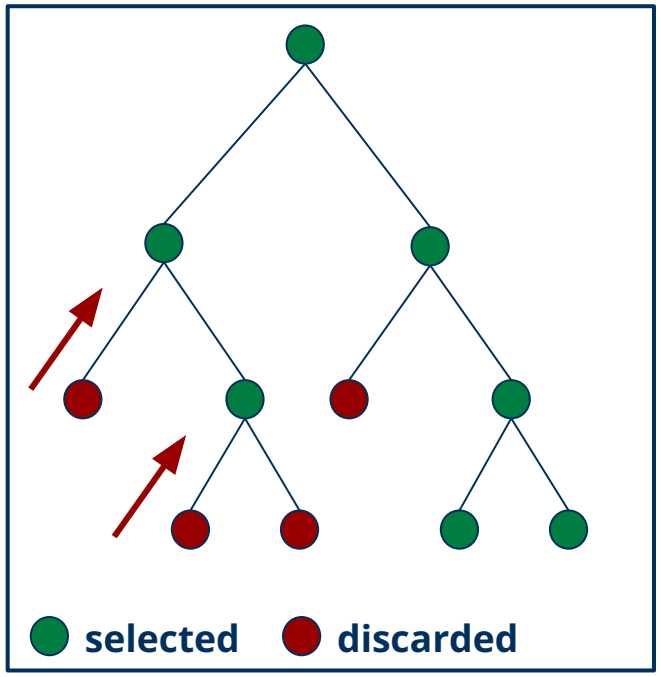
This allows, for example, do post-hoc determine last common ancestors of high scoring sequences

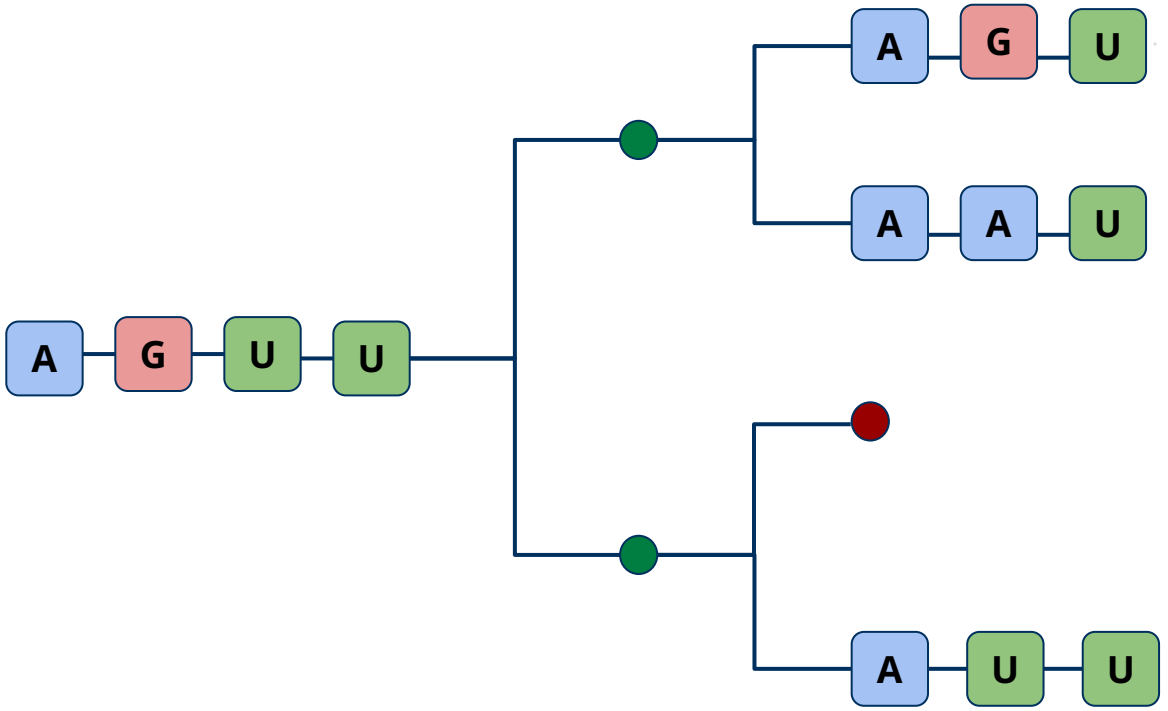




Tracking simulated generations as a tree structure

This allows, for example, do post-hoc determine last common ancestors of high scoring sequences

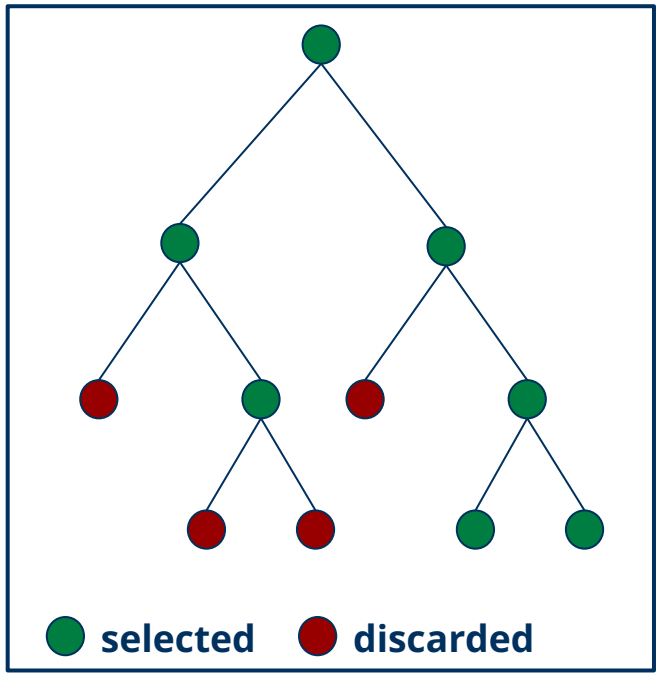




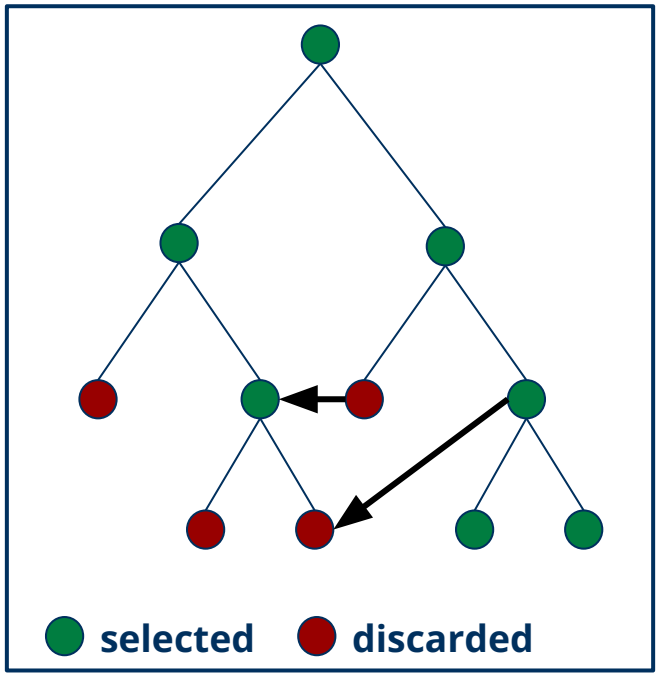
...this also potentially allows for backtracking on 'unfavorable' mutations...

Tracking simulated generations as a tree structure

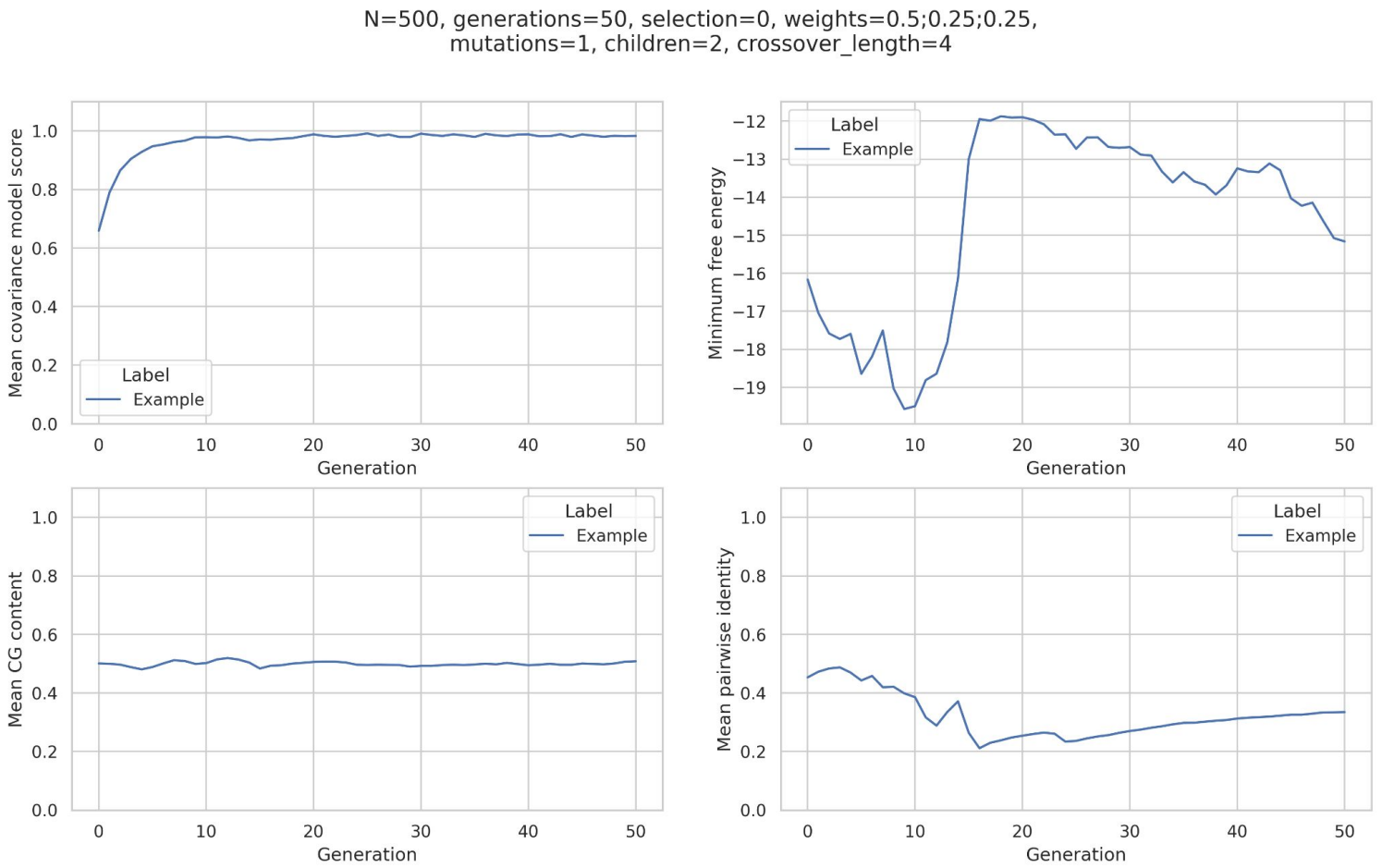
On principal, the approach would also allow for phylogenetic network-like structures



vs.

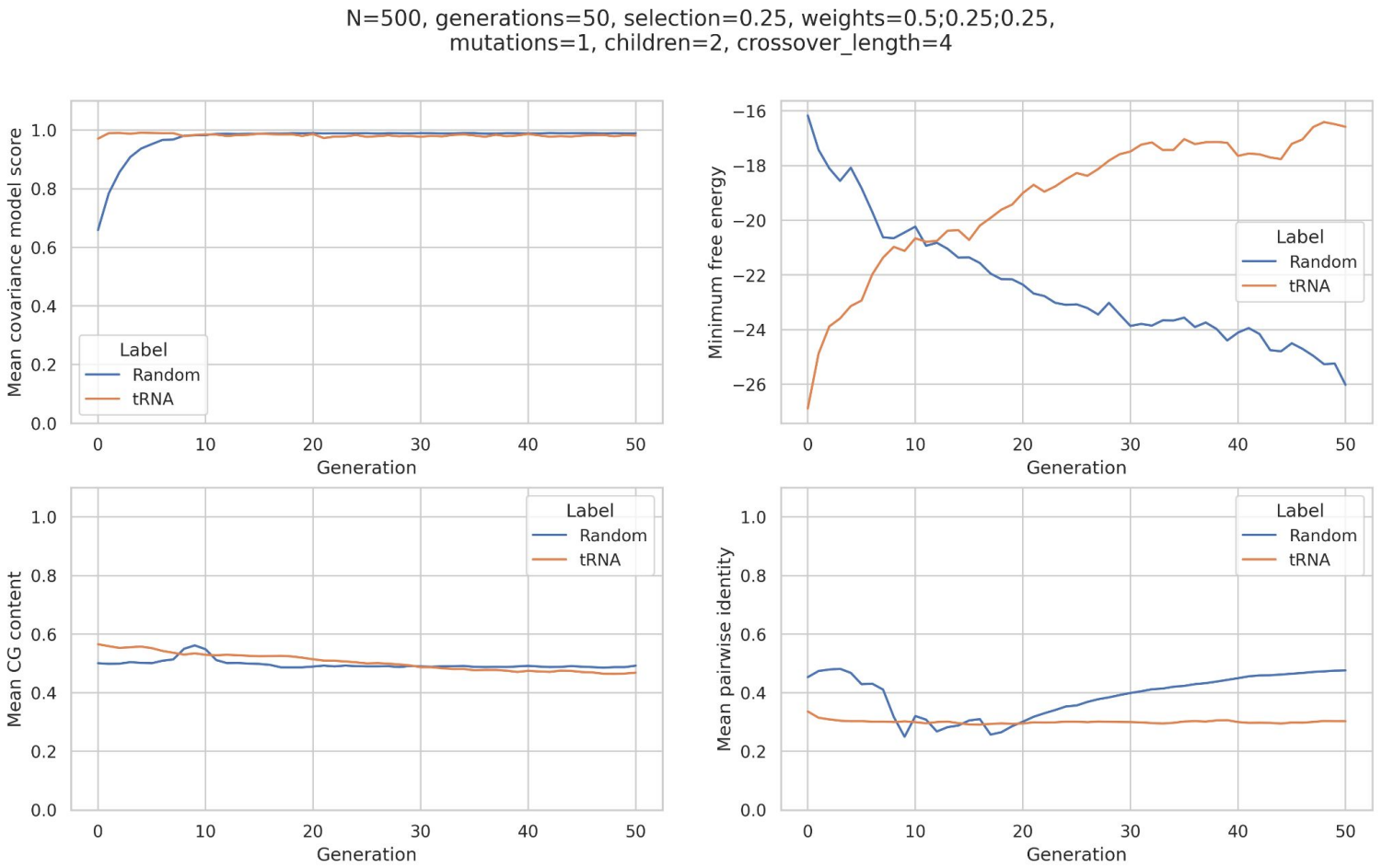


Evaluating simulation results



- **The ideal:**
  Ideally, we want an arbitrary starting population of RNAs to converge towards a common (given) secondary structure, while setting any sequence similarity threshold.

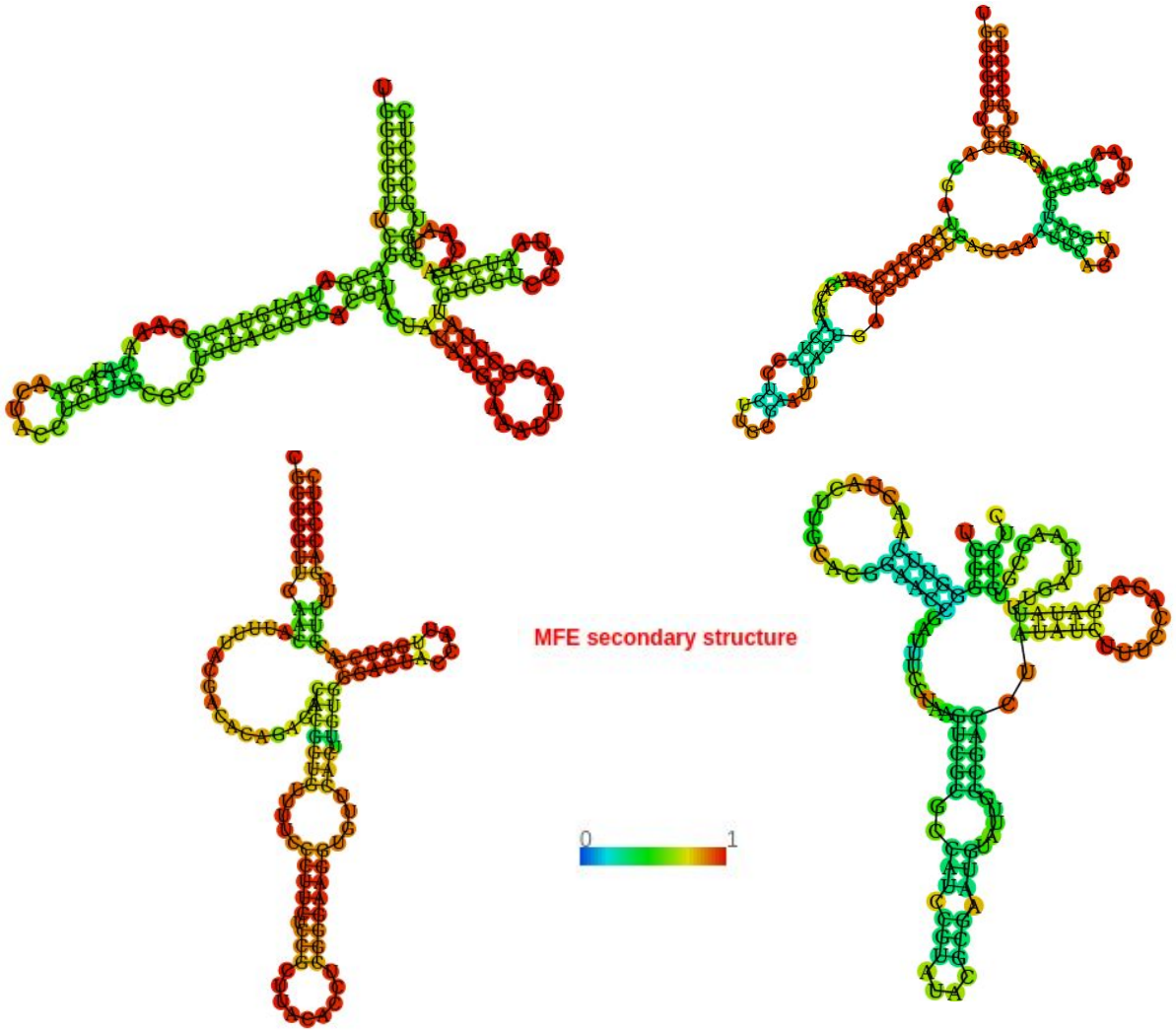
Example 1: Fun with tRNA structure learning



Label explanation

- **Random:**
  Initial set of randomized synthetic RNA sequences of lengths between 50-100
- **tRNA**
  An equal number of genuine tRNA sequences not in the covariance model data set.

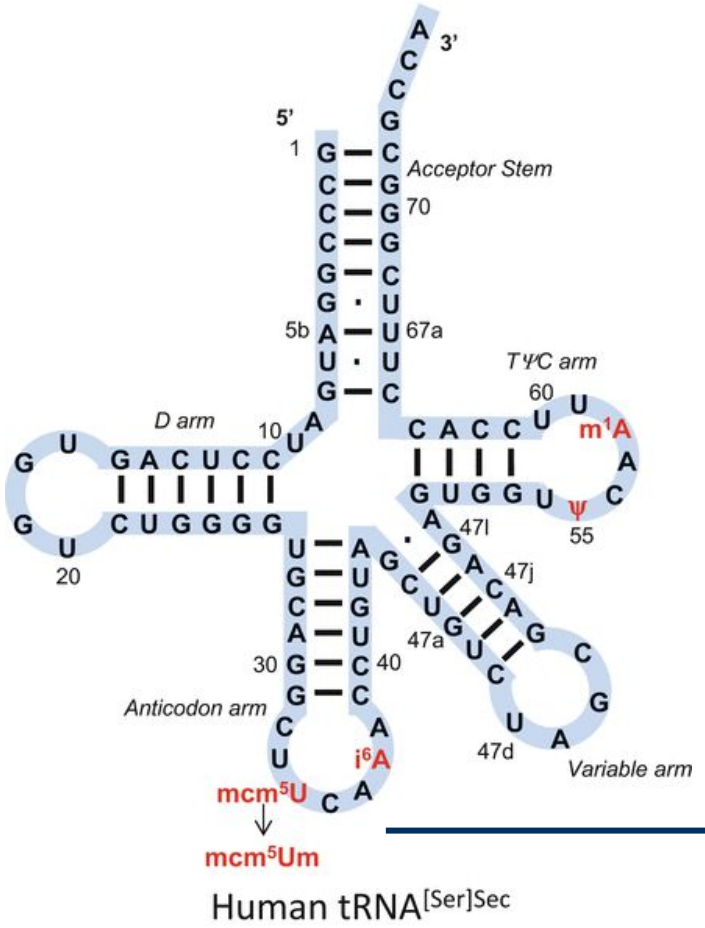
However - structural similarities?

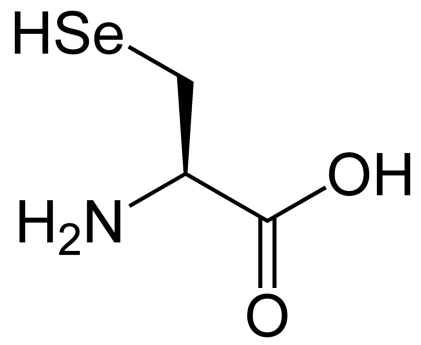


- **Things to maybe consider:**
  It may be smart, maybe even necessary, to preselect specific tRNAs in the initial covariance model… Cause these randomly evolved RNAs show a lot of variation!

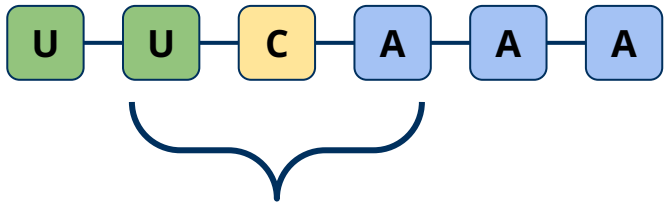
  Also: It is not like these guys have ‘valid’ anticodon loops -

Sequence constraints





extremely conserved region

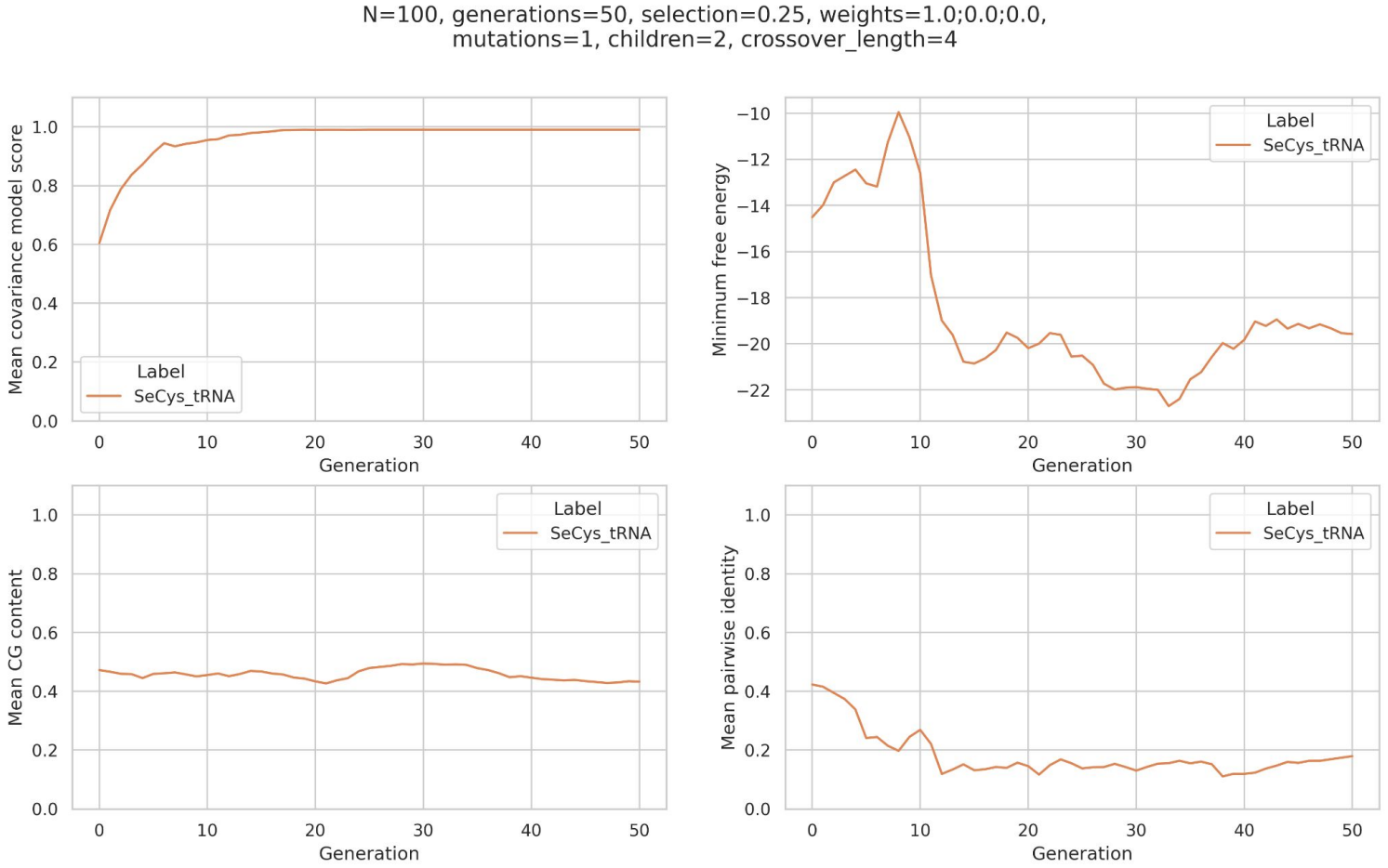


Anticodon

What does the simulation it look like, when the highly conserved region on the left is set as a **hard requirement**?

Figure edited from: Carlson, B. A., Lee, B. J., Tsuji, P. A., Tobe, R., Park, J. M., Schweizer, U., ... & Hatfield, D. L. (2016). Selenocysteine tRNA [Ser] Sec: from nonsense suppressor tRNA to the quintessential constituent in selenoprotein biosynthesis. *Selenium: its molecular biology and role in human health*, 3-12.

Sequence constraints



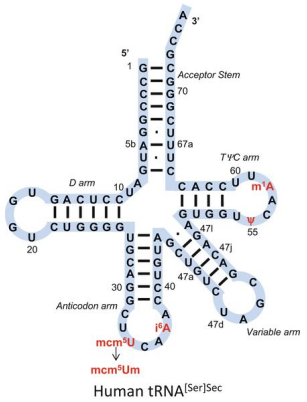
Starting population was initialized by generating random sequences with **length** between **70 and 85** and the motif **UUCAAA** between positions **33 and 39**.

Sequence constraints

Generated RNA population

cmbuild

Covariance model



BLAST

Rfam reference sequences

cmsearch

0 hits
('expected')

That could have gone better...

5 / 109 recalled
('could be more')

vs.
0 / 109 for a randomized training set

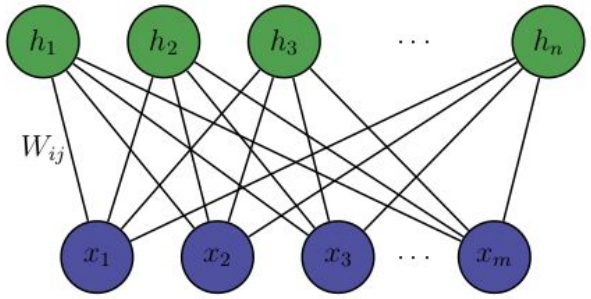
Things to do next

Experiment with different objective functions

Structural distance scores to an ‘ideal’ target structure (i.e. function of total and correctly paired bases), while retaining sequence constraints.

Structure learning with Boltzmann machines

Muntoni et al. recently introduced an application of restricted Boltzmann machines for the learning of biological sequence contact maps, technically allowing for a much more accurate representation of the baseline structure [1].



[1]: Muntoni, A. P., Pagnani, A., Weigt, M., & Zamponi, F. (2021). adabmDCA: adaptive Boltzmann machine learning for biological sequences. *BMC bioinformatics*, *22*(1), 1-19.

Other potential use cases

Synthetic data generation for machine learning applications

For many biological structures, only insufficient example data exists - synthetic sequences could supplement training data sets, to leave the *bona fide* examples for the (necessary) independent test set.

Homology search

Depending on the size of the search space, a constrained approach can be used to generate biologically plausible RNAs for homology search in other species, as was done with Riboswitches by *Murherjee et al*. [1].

Constrained simulation of RNA evolution

As most parameters are freely definable, the system could also be used to analyze convergence/divergence of sequence pools towards conserved elements , in particular where simulation of the whole structure space is not feasible.

[1]: Mukherjee, S., Retwitzer, M. D., Hubbell, S. M., Meyer, M. M., & Barash, D. (2023). A computational approach for the identification of distant homologs of bacterial riboswitches based on inverse RNA folding. *Briefings in Bioinformatics*, *24*(3), bbad110.

UNIVERSITÄT LEIPZIG

BIO Leipzig
‹INFORMATICS›

ScaDS.AI
DRESDEN LEIPZIG

Thaaaaaaanks for listening!

(this is the part where you tell me that all of this is nonsense)

Special thanks to:
Prof. Peter F. Stadler
Dr. Sven Findeiß

...and literally everyone @Bioinf