Software demo

Benasque July 2026

Hackaton Prague 2026

Bogdan Schneider

Nucleic Acid Structure Annotation & Prediction Hackathon 2026 took place in Prague in May 27-29. It was supported by ELIXIR as an activity of 3D-Bioinfo community implementation plan and organized by Lada Biedermannova and Bohdan Schneider from the Czech Academy of Sciences.

About 25 participants worked in two workstreams (https://github.com/na-hackathon/na-hackathon-2026/tree/main/workstreams):

Workstream 1 — Unified annotation & validation workflow
Integrate key annotation and validation steps into a single workflow with consistent, standardized outputs, demonstrated on agreed case studies.

Workstream 2 — RNA prediction and non-canonical base pairs
Define or refine interoperable 2D + mmCIF representations of canonical and non-canonical base pairs, and use them to support benchmarking and case studies for prediction approaches.

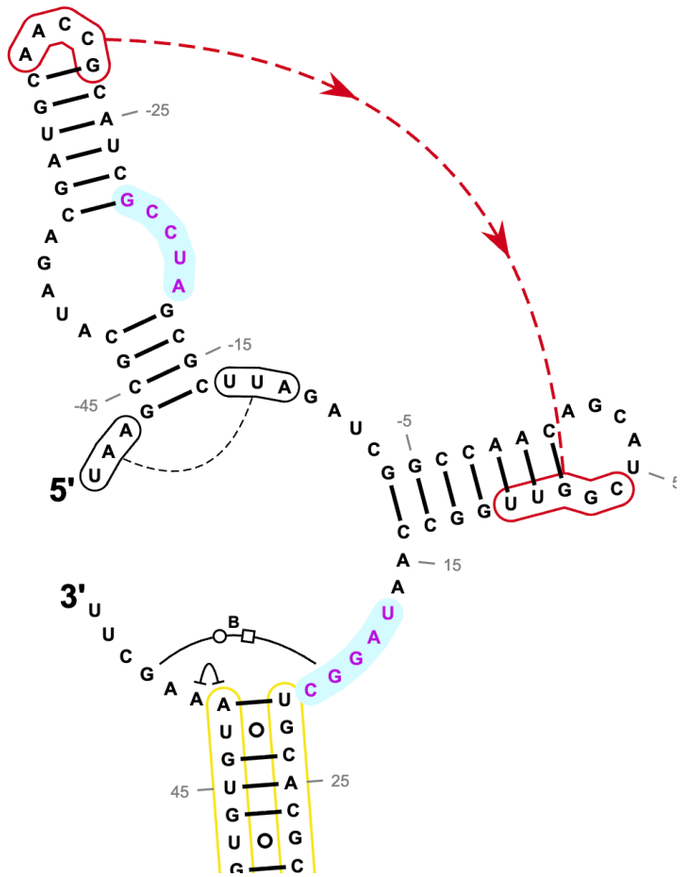
In both workstreams, we've achieved a significant progress in understanding the processes allowing more robust data sharing. Developed tools will be summarized in a report, our progress can be viewed at github (see above).

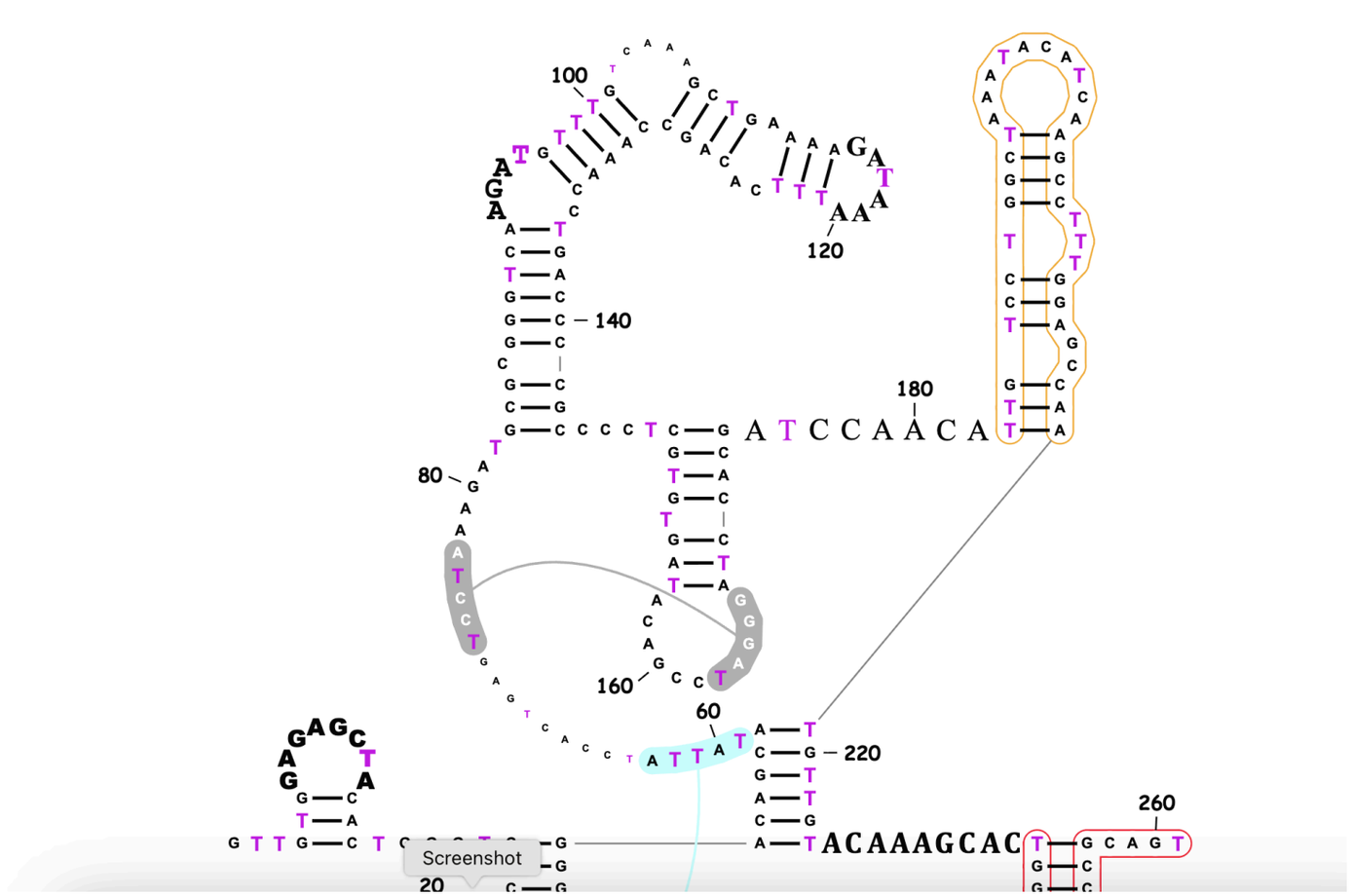
RNACanvas

Philip Johnson

"Highly interactive 2D RNA structure drawing and editing" with a link to RNAcanvas: https://rnacanvas.app

rnacanvas.app
RNAcanvas
A web app for drawing and exploring nucleic acid structures.





Webserver interactive ensemble RNA plotter (ARNy Plotter)

Samuela Pasquali

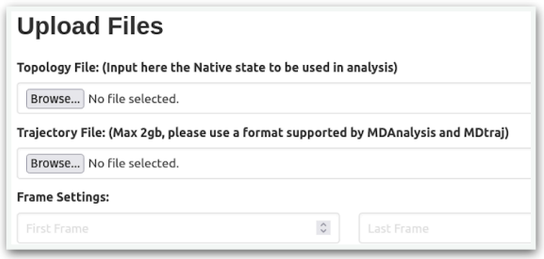
https://arny-plotter.rpbs.univ-paris-diderot.fr/

Input panel

Upload structural ensemble

Choose analysis

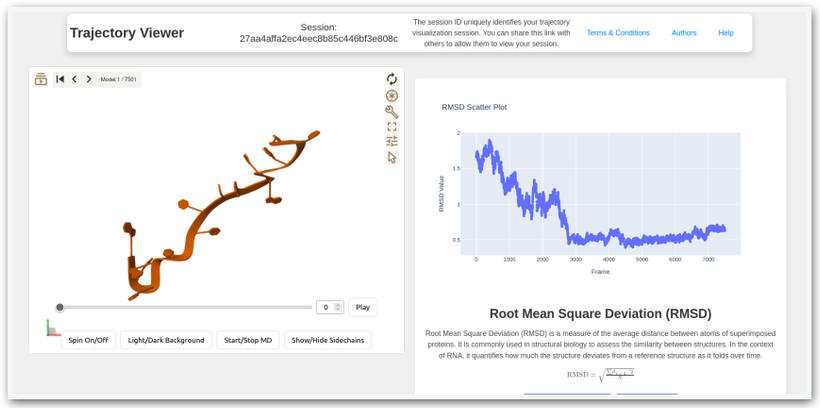
Retrive previous results

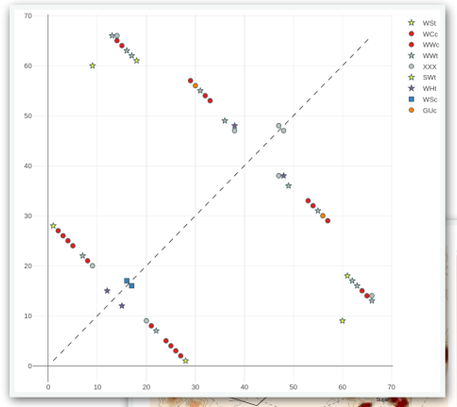


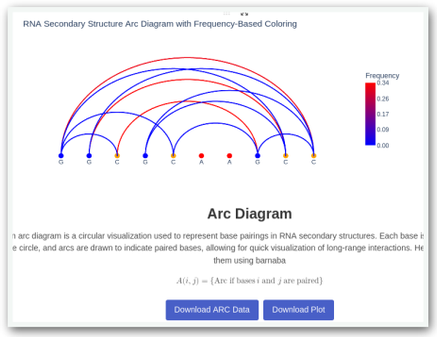
Documentation and examples

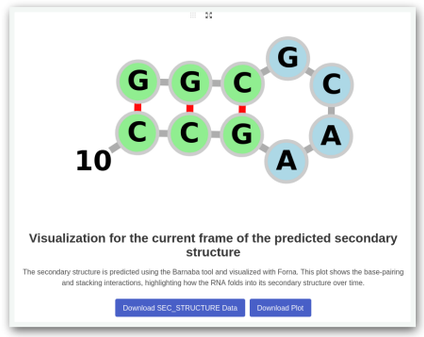
Processing Panel

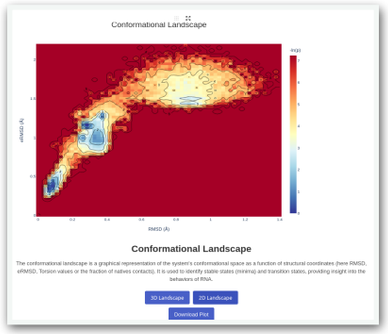
Results Panel

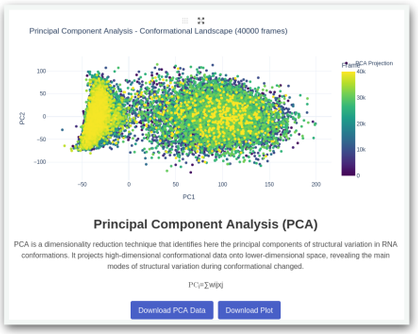


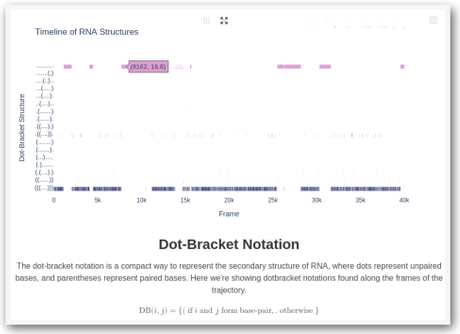




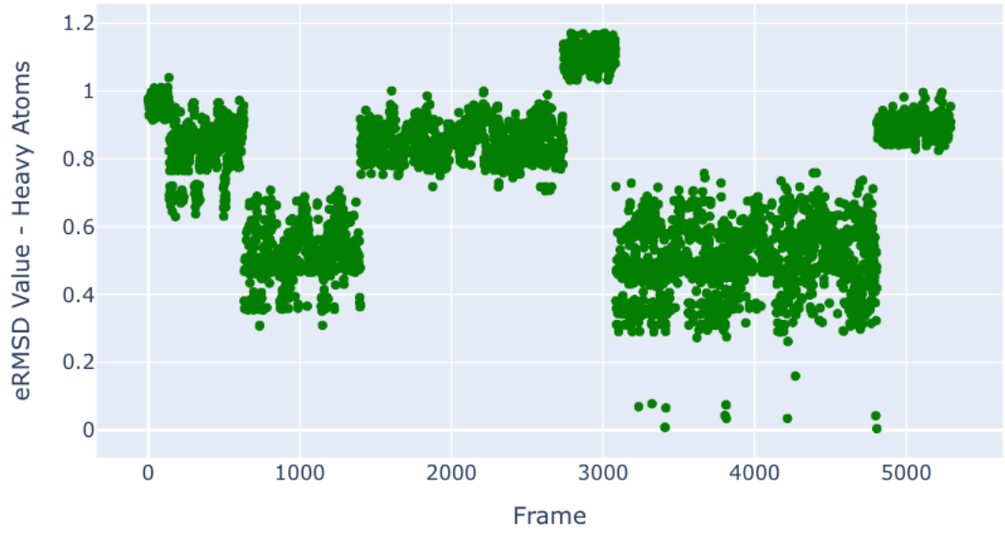


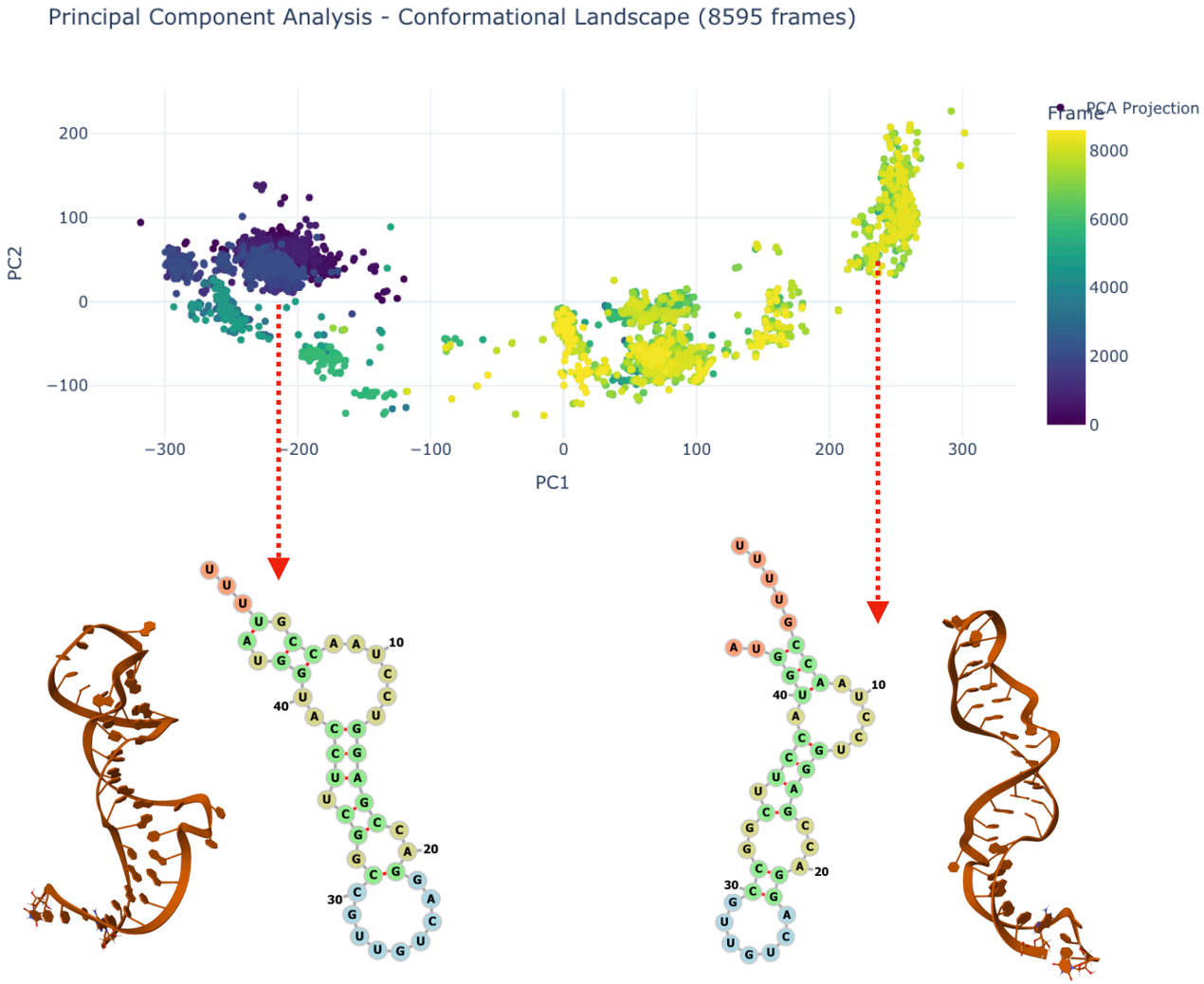






Meuret, Pasquali, PLOS Computational Biology (2026), https://doi.org/10.1371/journal. pcbi.1013873





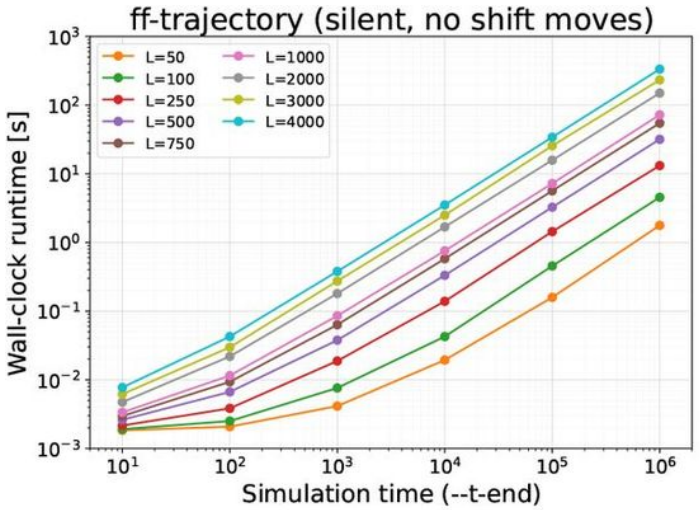
Interactive setup
Selecting a point on any plot will show the corresponding structure and updte all single structure analysis (contact map, 2D, etc)

Fuzzyfold

Stefan Badelt

The fuzzyfold workspace

Fast, modular, and extensible stochastic RNA folding



cargo install fuzzyfold

https://github.com/bad-ants-fleet/fuzzyfold

contributions welcome

Current fuzzyfold crates:

fuzzyfold v0.4.2 | ff_structure v0.3.1 | ff_energy v0.4.1 | ff_kinetics v0.4.2

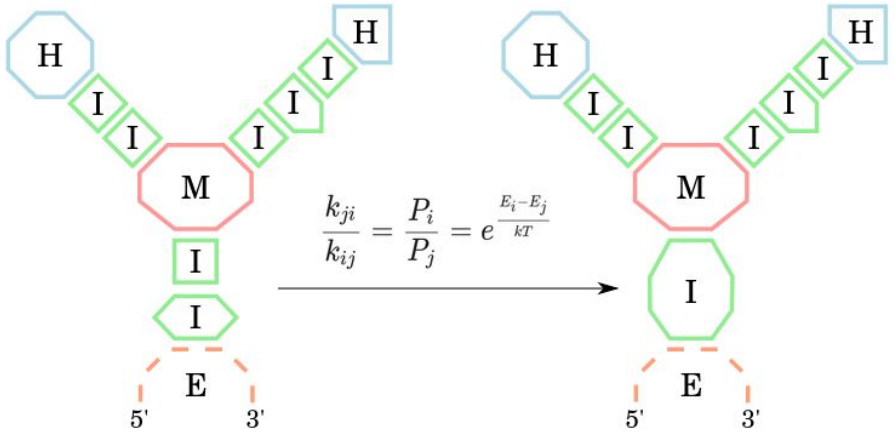
- ff_structure: Common secondary structure data structures.
- ff_energy: Secondary structure free energy evaluations.
- ff_kinetics: Stochastic folding kinetics for nucleic acids.
- fuzzyfold: Command-line interfaces.

universität wien

tbi

Stochastic RNA folding kinetics

Choose a **move set** and a **rate model** that satisfies detailed balance, e.g.:



Simulate stochastic folding trajectories as given by the master equation

$$\frac{dP_i(t)}{dt} = \sum_{i \neq j} (P_j(t)k_{ji} - P_i(t)k_{ij})$$

⇒ yields the correct equilibrium distribution

⇒ whether kinetics is correct depends on the rate model!

universität wien

ff-trajectory: single stochastic simulations

GCGUUUCCAGGGUUUAGACGGACGGGUGUGACUCGCCCAGCCCCGACCUC	energy	arrival-time	waiting-time	mean-waiting
.....	0.00	0.00000000e0	5.48838308e-1	1.15990269e0
.....(.....).....	2.10	5.48838308e-1	7.42186429e-1	6.00461698e-1
.....(.....).....(.....).....	4.50	1.29102474e0	6.62519714e-1	3.57034800e-1
.....(.....).....	2.40	1.95354445e0	8.70308936e-3	5.15932279e-1
.....	0.00	1.96224754e0	2.24827455e0	1.15990269e0
.....(.....).....	1.90	4.21052209e0	1.08942365e0	4.14255646e-1
.....((.....)).....	0.10	5.29994574e0	3.93454939e-1	5.63963354e-1
.....(.((.....)).).....	-0.20	5.69340068e0	1.06769149e-1	4.52730587e-1
.....(.(.((.....)).).).....	1.80	5.80016983e0	4.33243095e-1	3.16296938e-1
.....(((.((.....)).))).....	-2.40	6.23341293e0	1.28813977e0	8.21431625e-1
.....(.(((.((.....)).)))).....	-1.60	7.52155270e0	2.45415452e-1	2.40731917e-1
.....(.(.(((.((.....)).)))))....	-1.50	7.76696815e0	6.94202383e-2	4.10851300e-1
.....(((.(((.((.....)).))))))....	-5.60	7.83638839e0	3.47571920e0	3.26611240e0
.....(((..((.((.....)).)).)))....	-2.90	1.13121076e1	1.55874453e0	4.13337266e-1
.....(((.(((.((.....)).))))))....	-5.60	1.28708521e1	2.36328447e0	3.26611240e0
.....(((.(((.((.....)).)))))).....(.....).....	-2.10	1.52341366e1	7.41356896e-3	4.84982503e-1
.....(((.(((.((.....)).)))))).((...))....	-4.70	1.52415502e1	2.00829892e1	1.23961105e1
..(.....(((.(((.((.....)).)))))).((...)).....)..	-0.50	3.53245394e1	1.26546682e0	7.99897165e-1
.....(((.(((.((.....)).)))))).((...))....	-4.70	3.65900062e1	4.58067121e1	1.23961105e1

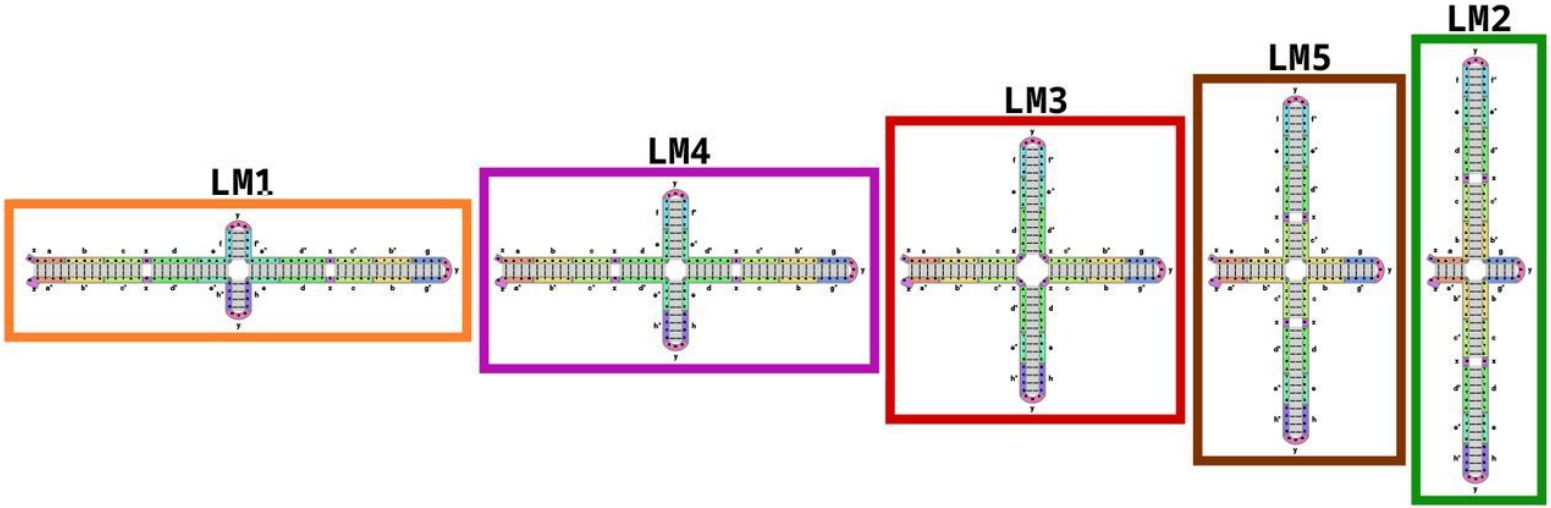
universität wien

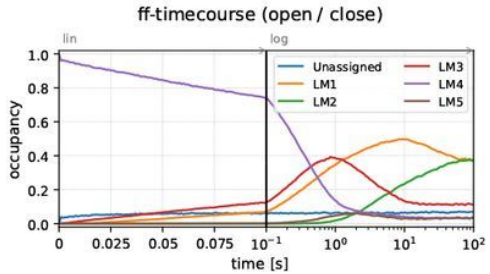
ff-trajectory: single stochastic simulations

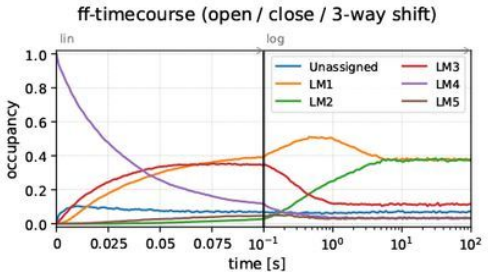
```
[I] (base) stefan@brain20:~/Work/Rust/fuzzyfold$ ff-trajectory --help
Stochastic Simulation Algorithm for RNA folding

Usage: ff-trajectory [OPTIONS] [INPUT]

Arguments:
  [INPUT]  Input file (FASTA-like), or "-" for stdin [default: -]

Options:
      --fasta          Read input as standard FASTA (DNA alphabet, multi-line sequence, no structure)
      --t-end <T_END>  Simulation time at the full-length molecule [default: 1]
      --t-ext <T_EXT>  Optional simulation time per nucleotide during co-transcriptional folding
      --silent         Do not print trajectory, only last structure
  -h, --help           Print help
  -V, --version        Print version

Energy model parameters:
      --celsius <CELSIUS>   Temperature in Celsius [default: 37.0]
      --rna [<RNA_PRESET>]  Built-in RNA parameter set (default: turner2004ext) [possible values: turner2004ext, turner2004, andronescu2007]
      --dna [<DNA_PRESET>]  Built-in DNA parameter set (default: mathews2004) [possible values: mathews2004]

Kinetic model parameters:
      --k0 <K0>      Rate constant for add/delete moves [default: 100000]
      --k3ws <K3WS>  Rate constant for three-way shift moves (optional, default = off)
      --k4ws <K4WS>  Rate constant for four-way shift moves (optional, default = off)
```

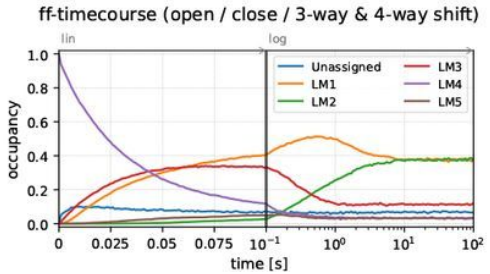
universität wien

ff-timecourse: ensemble dynamics









universität wien

ff-timecourse: ensemble dynamics

```
[I] (base) stefan@brain20:~/Work/Rust/fuzzyfold$ ff-timecourse --help
Stochastically simulated nucleic acid ensembles over time.

Usage: ff-timecourse [OPTIONS] --output <FILE> [INPUT]

Arguments:
  [INPUT]  Input file (FASTA-like), or "-" for stdin [default: -]

Options:
  -n, --num-sims <NUM_SIMS>    [default: 1]
      --macrostates <FILE>...
  -o, --output <FILE>
  -t, --title <TITLE>
  -h, --help                   Print help
  -V, --version                Print version

Simulation parameters:
      --t-ext <T_EXT>  The last time point of the linear scale [default: 0.04]
      --t-end <T_END>  Simulation stop time [default: 1]
      --t-lin <T_LIN>  Number of time points on the linear scale: [0..t-ext] [default: 100]
      --t-log <T_LOG>  Number of time points on the logarithmic scale: [t-ext..t-end] [default: 100]

Energy model parameters:
      --celsius <CELSIUS>   Temperature in Celsius [default: 37.0]
      --rna [<RNA_PRESET>]  Built-in RNA parameter set (default: turner2004ext) [possible values: turner2004ext, turner2004, andronescu2007]
      --dna [<DNA_PRESET>]  Built-in DNA parameter set (default: mathews2004) [possible values: mathews2004]

Kinetic model parameters:
      --k0 <K0>      Rate constant for add/delete moves [default: 100000]
      --k3ws <K3WS>  Rate constant for three-way shift moves (optional, default = off)
      --k4ws <K4WS>  Rate constant for four-way shift moves (optional, default = off)
```

universität wien

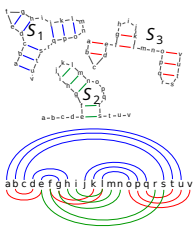
INFRARED

Sebastian Will

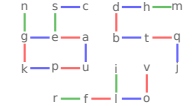
Summary of Infrared

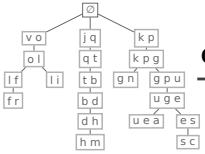
Weighted Constraint Satisfaction Problem (CSP)

$(\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F})$



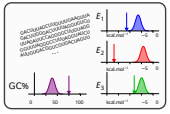
Constraints $\mathcal{C}$: S_1, S_2, S_3
Functions $\mathcal{F}$: GC%, E_1, E_2, E_3





Tree decomposition
Treewidth w

Sampling



Optimization

GCCGCGGUAGCU · · ·

Time Complexity
$\mathcal{O}(d^{w+1} \cdot (|\mathcal{X}| + |\mathcal{C}| + |\mathcal{F}|))$

Optimization

Objective function

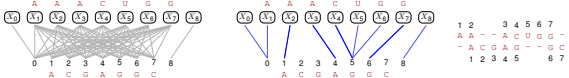
Positive Design

***De novo* xrRNA design with Infrared: talk by Leonhard Sidl at Friday afternoon virus session.**

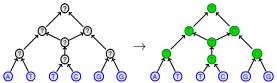
Infrared xrRNA

Infrared: Applications

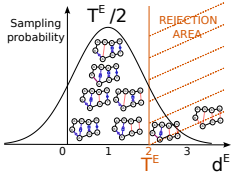
- Sequences (structure) alignment, $w \approx 3$



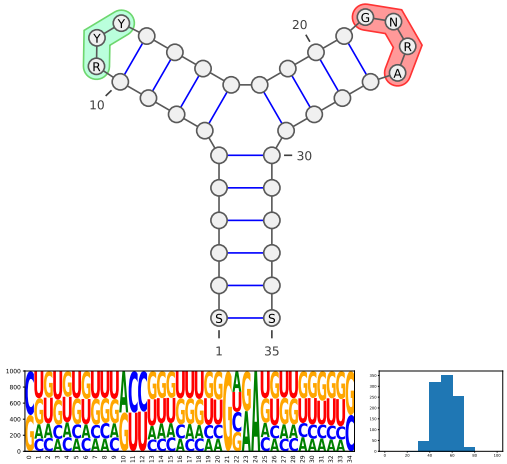
- Parsimony for phylogenetic reconstruction, $w \approx 4$



- Explore of 3D motif variations, $w \approx 3$ (T. Boury *et al.*, 2023)



Single structure design



[H-T. Yao *et al.*, RNA Folding, 2024 (Book chapter)]

```python
import infrared as ir
import infrared.rna as rna

model = ir.Model(35, 4)

target = "((((((((((...))))))((((....))))))))))"
model.add_constraints(rna.BPComp(i, j)
    for (i, j) in rna.parse(target))

iupac_seq = "SNNNNNNNNNRYYNNNNNNNNGNRANNNNNNNNNS"
for i, x in enumerate(iupac_seq):
    model.add_constraints(
        ir.ValueIn(i, rna.iupacvalues(x)))


sampler = ir.Sampler(model)
samples = [sampler.sample() for _ in range(1000)]
```

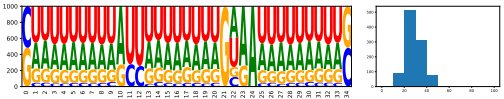
0 : A 1 : C 2 : G 3 : U

AU, CG, · · ·

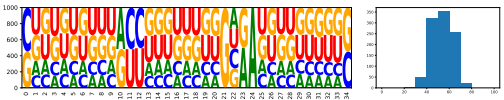
N : ACGU S : CG R : AG Y : CU

Control GC

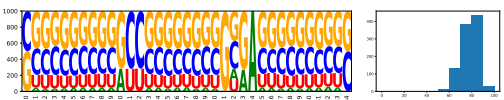
$\alpha = -1$



$\alpha = 0$



$\alpha = +1$



Method 1:

CG : 1 AU : 0

```
model.add_functions([rna.GCCont(i))
    for i in range(n)], 'gc')
model.set_feature_weight(α, 'gc')

sampler = ir.Sampler(model)
samples = [sampler.sample() for _ in range(1000)]
```

Method 2 (Targeted sampling):

```
sampler = ir.Sampler(model)
sampler.set_target( 0.75 * n, 0.01 * n, 'gc' )
samples = [sampler.targeted_sample()
    for _ in range(1000)]
```

Automatically update α

LinearCapR:
Linear-time computation of per-nucleotide structural-context probabilities of RNA without base-pair span limits

Takumi Otagaki, Hiroaki Hosokawa, Tsukasa Fukunaga, Junichi Iwakiri, Goro Terai and Kiyoshi Asai

Institute of Science Tokyo, PhD student

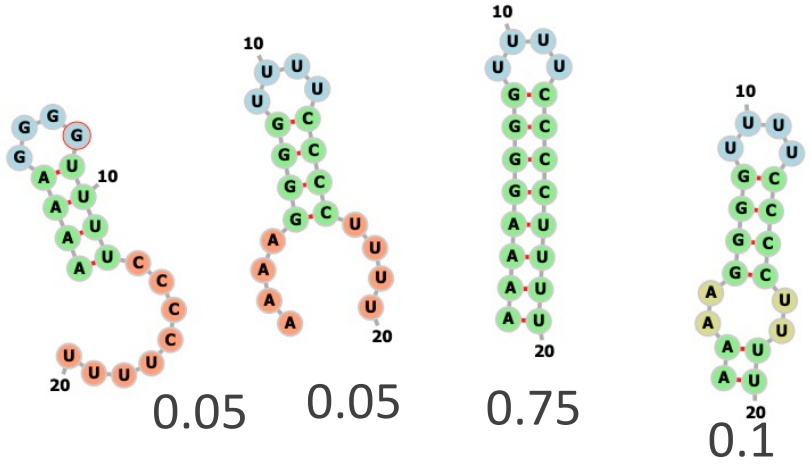
Benasque 2026

Otagaki, Takumi, Hiroaki Hosokawa, Tsukasa Fukunaga, Junichi Iwakiri, Goro Terai, and Kiyoshi Asai. 2026.
"LinearCapR: Linear-Time Computation of per-Nucleotide Structural-Context Probabilities of RNA without Base-Pair Span Limits." *Bioinformatics (Oxford, England)* 42 (6): btag295.

RNA structure is an ensemble

- The structure of RNA is not so rigid.
- Considering the ensemble of RNA structures enables us to understand more.
- Marginalized Probability is “some” probability considering the ensemble.

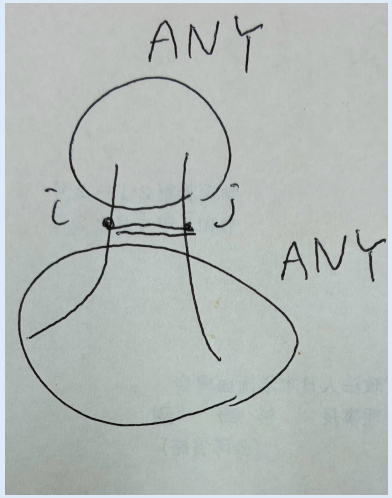
$AAAAGGGGUUUUCCCCUUUU \rightarrow$



Marginalized Probability; Base Pairing Probability and Accessibility

Base pairing probability
is the probability that
i-th and j-th nucleotides form a base pair.

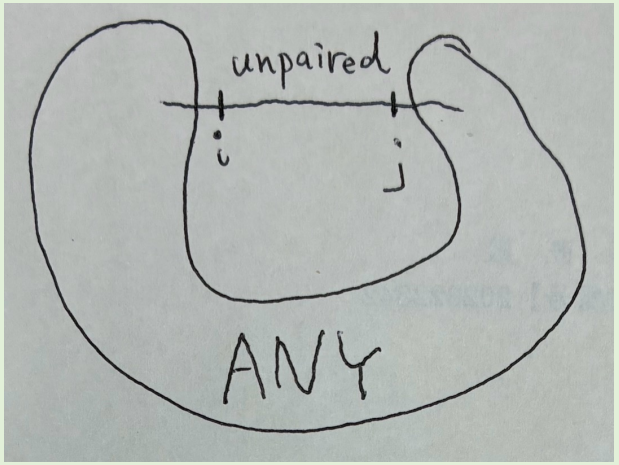
$$bpp(i,j) = \sum_{\sigma \in S} 1\{\text{base pair } (i,j) \in \sigma\}\, p(\sigma)$$



S: set of all possible structures

Accessibility
is (often) the probability that
a window $[i:j]$ is all-unpaired.

$$p_{\text{acc}}(i,j) = \sum_{\sigma \in S} 1\left\{\begin{array}{c}\forall\, k\ (i \le k \le j) \\ \text{forms no base pair} \in \sigma\end{array}\right\} p(\sigma)$$



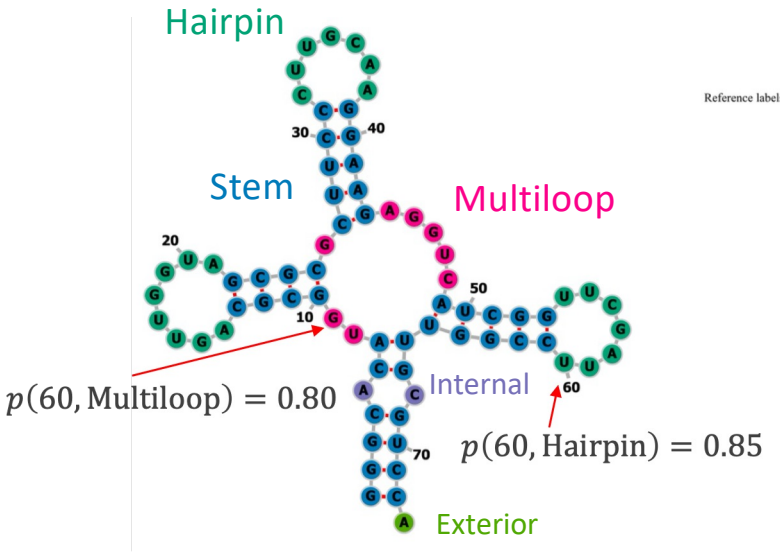
Marginalized Probability; **Loop Profile**

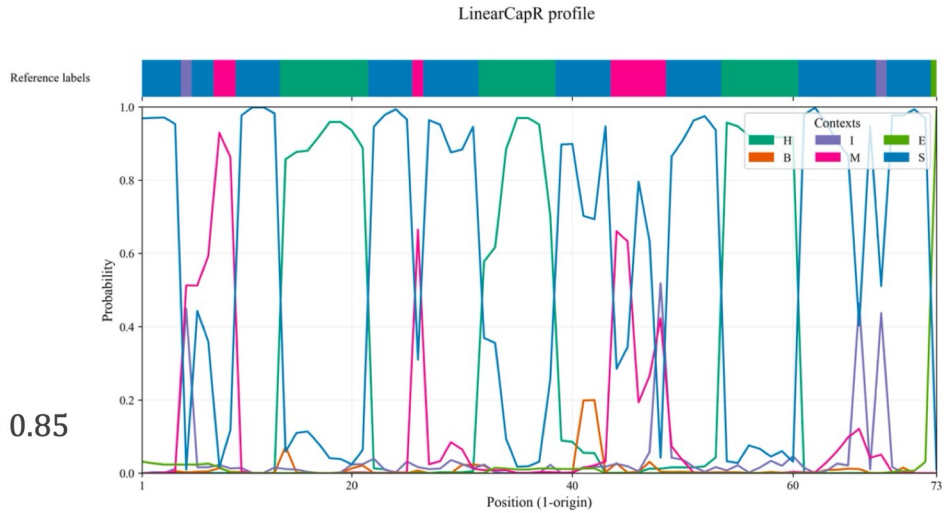
Loop profile is the probability that i-th nucleotide is located in the loop δ.

$$p(i, \delta) = \sum_{\sigma \in S} 1\{i \text{ is in the loop } \delta \in \sigma\}\, p(\sigma)$$

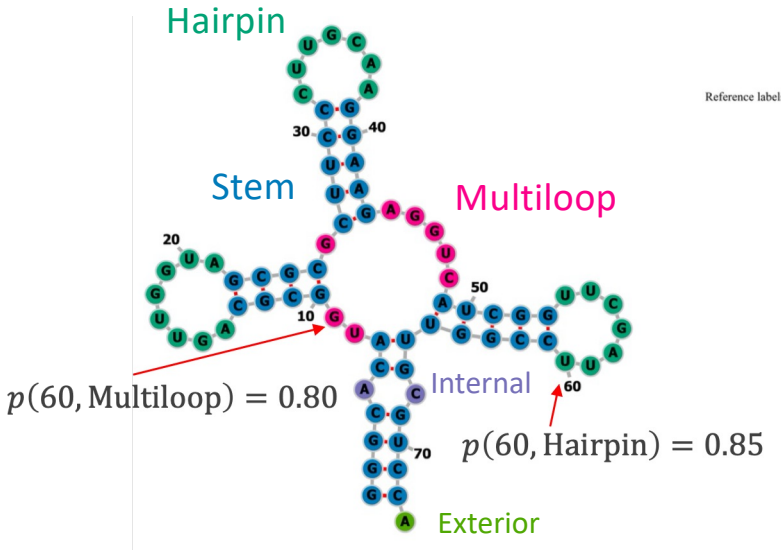
S: set of all possible structures

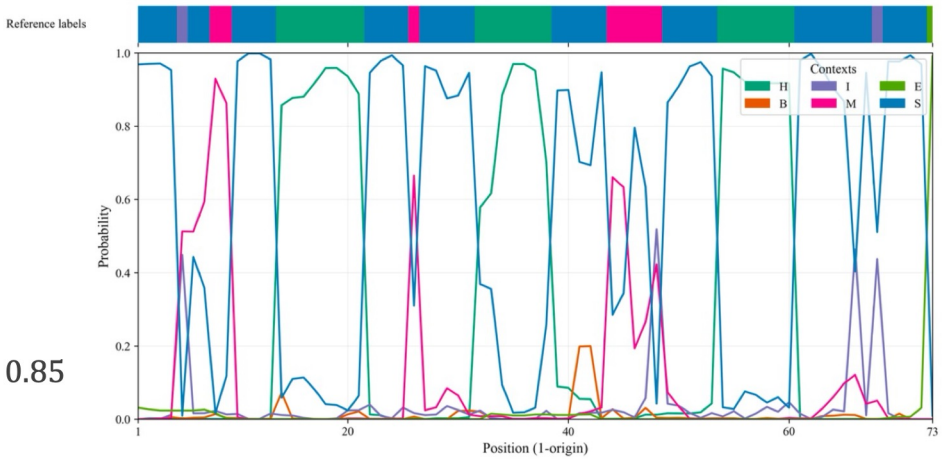
$\delta \in \{\text{Stem, Hairpin, Internal , Bulge, Multiloop, Exterior}\}$





Loop profile





- Loop profile helps us with understanding the structure ensemble in terms of the loop-type.
- CapR (2014, Fukunaga et al.) calculates the loop profile.
- Can be used for the RBP binding site analysis, virus genome functional analysis

How can we scale loop-profile computation?

CapR (previous work) imposes the **max base-pair distance W**, and calculate the loop profile in

$$\text{Time: } O(NW^2), \text{Space: } O(NW)$$

where, N is the length of the input sequence.

However, max base-pair distance W discards long range base pairings.

Proposed software; LinearCapR

LinearCapR paper↓

→We propose LinearCapR, which

- **is based on the beam pruning**
- **can incorporate the long-range base pair**
- **calculates the loop profile in the linear time**

Same Idea with LinearPartition

LinearPartition calculates base pair probability in **linear-time** by beam pruning, **without excluding the long-range base pairs**.

Otagaki, Takumi, Hiroaki Hosokawa, Tsukasa Fukunaga, Junichi Iwakiri, Goro Terai, and Kiyoshi Asai. 2026.
"LinearCapR: Linear-Time Computation of per-Nucleotide Structural-Context Probabilities of RNA without Base-Pair Span Limits." *Bioinformatics (Oxford, England)* 42 (6): btag295.

1. Concept of beam pruning in the structure-ensemble analysis.

Every RNA secondary structure except for pseudoknots can be generated from the Stochastic Context-Free Grammar (SCFG) framework.

$$\begin{aligned}
\text{Outer} &\rightarrow \epsilon \mid \text{Outer} \cdot a \mid \text{Outer} \cdot \text{Stem} \\
\text{Stem} &\rightarrow b_< \cdot \text{Stem} \cdot b_> \mid b_< \cdot \text{StemEnd} \cdot b_> \\
\text{StemEnd} &\rightarrow s_n \mid s_m \cdot \text{Stem} \cdot s_n \; (m+n>0) \mid \text{Multi} \\
\text{Multi} &\rightarrow s_n \cdot \text{MultiBif} \\
\text{MultiBif} &\rightarrow \text{Multi1} \cdot \text{Multi2} \\
\text{Multi1} &\rightarrow \text{MultiBif} \mid \text{Multi2} \\
\text{Multi2} &\rightarrow \text{Stem} \cdot s_n
\end{aligned}$$

Table 1. Non-terminal symbols used in the SCFG.

Name	Description
`Outer`	Exterior unpaired region
`Stem`	Helix extension
`StemEnd`	Terminates a helix into a hairpin, bulge/internal loop, or multiloop region
`Multi`	Completed multiloop
`MultiBif`	Bifurcation into two or more branches
`Multi1`	One-or-more list of branches
`Multi2`	One branch followed by a 3'-side run of unpaired bases

1. Concept of beam pruning in the structure-ensemble analysis.

The core idea is the pruning for each state (Outer, Stem, …) for each index j.

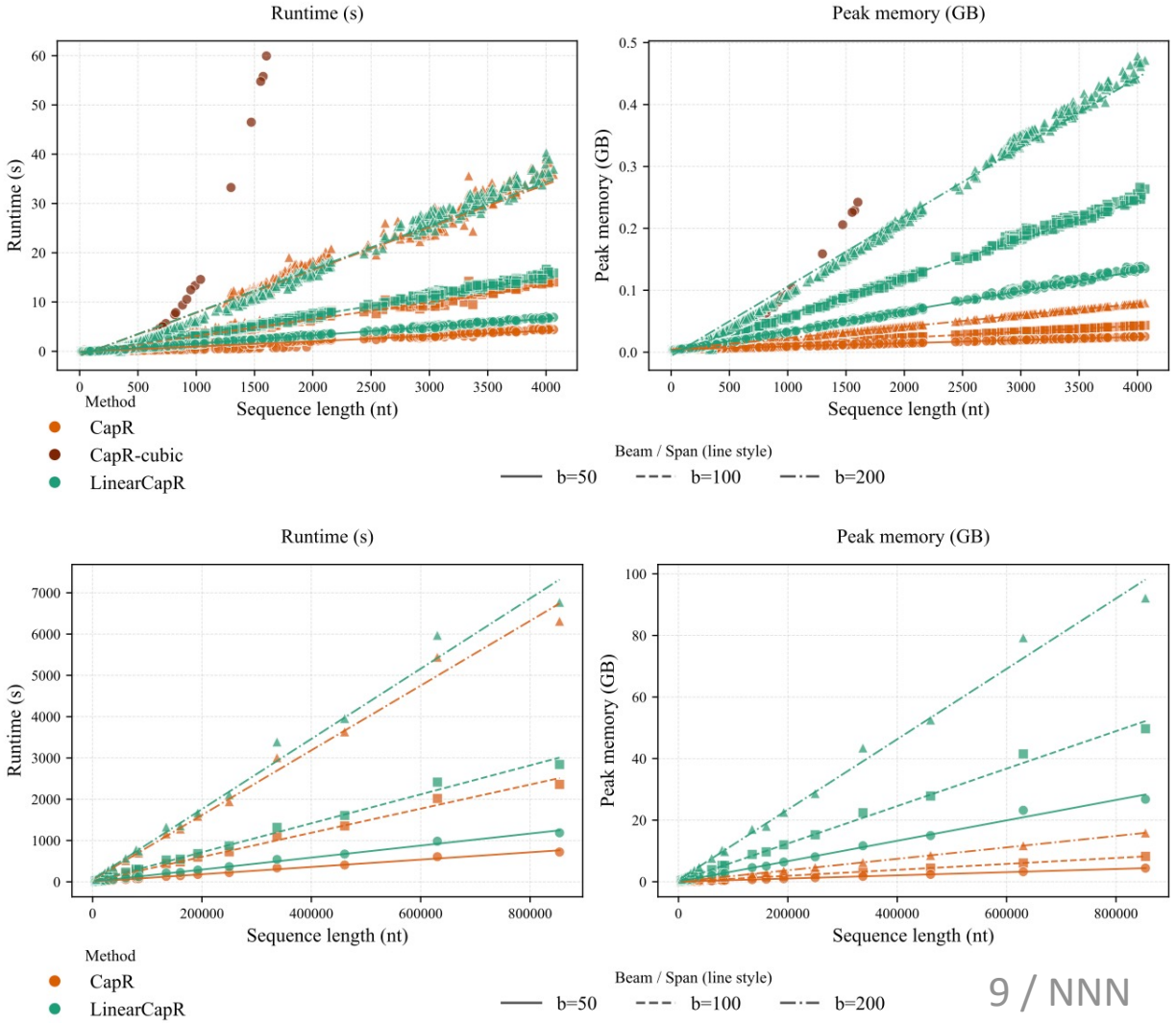
procedure INSIDE ALGORITHM IN LINEARCAPR
  for $j = 1 \ldots N$ **do**
    prune$(\alpha_{\text{Stem}}, j)$ ←pruning the Stem state with index j
    for all i s.t. $[i, j] \in \alpha_{\text{Stem}}$ **do** ← summing up for the remained j
      $\alpha_{\text{Stem}}(i-1,\ j+1)\ \mathrel{+}= \alpha_{\text{Stem}}(i,j) \cdot t(\text{Stem} \to b_< \cdot \text{Stem} \cdot b_>)$ # Stem
      $\alpha_{\text{Outer}}(j)\ \mathrel{+}= \alpha_{\text{Outer}}(i-1) \cdot \alpha_{\text{Stem}}(i,j) \cdot t(\text{Outer} \to \text{Outer} \cdot \text{Stem})$
      for $n = 0 \ldots C$ **do**
        $\alpha_{\text{Multi2}}(i,\ j+n)\ \mathrel{+}= \alpha_{\text{Stem}}(i,j) \cdot t(\text{Multi2} \to \text{Stem} \cdot s_n)$ # Multiloop
      for all p, q s.t. $p \le i < j \le q,\ 0 < (i-p) + (q-j) \le C$ **do**
        $\alpha_{\text{StemEnd}}(p,q)\ \mathrel{+}= \alpha_{\text{Stem}}(i,j) \cdot t(\text{StemEnd} \to s_{i-p} \cdot \text{Stem} \cdot s_{q-j})$ # Internal/Bulge
        ⋮

Results

Time/memory Comparison of CapR (max base-pair span) and LinearCapR (beam pruning)

- CapR $O(NW^2)$
- CapR-cubic $O(N^3)$
- LinearCapR $O(Nb^2)$

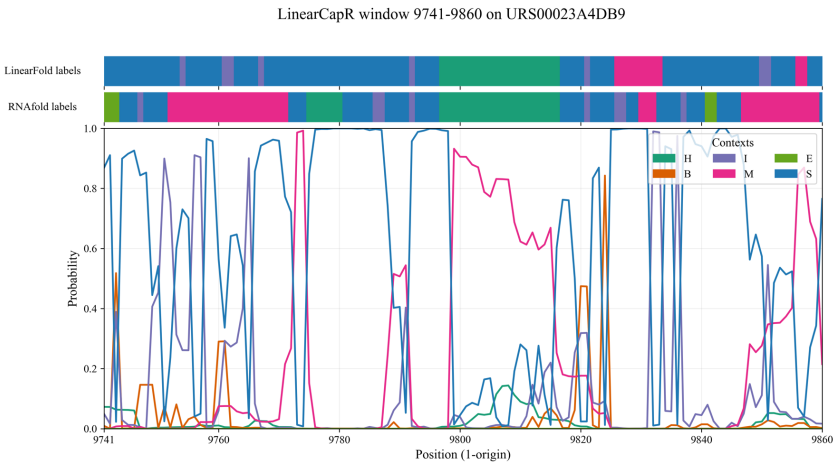
- Runtime and memory scale approximately linearly once b or W is fixed.
- LinearCapR is span-unrestricted but keeps more active states.
- Memory, rather than runtime, becomes the practical limiter for very long RNAs.

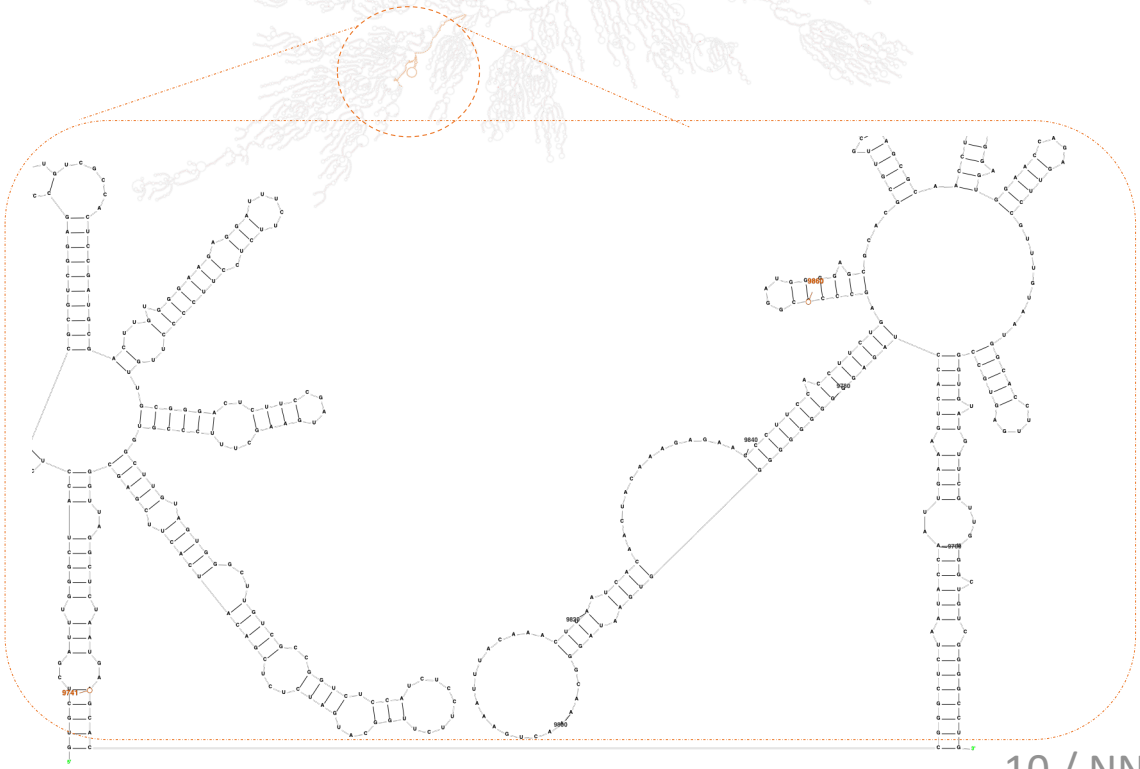


Results

Long RNA (Zea mays lncRNA from RNAcentral, ~30kbp)

Without discarding long range base pairing,
LinearCapR computes the
Loop profile for longer RNA.



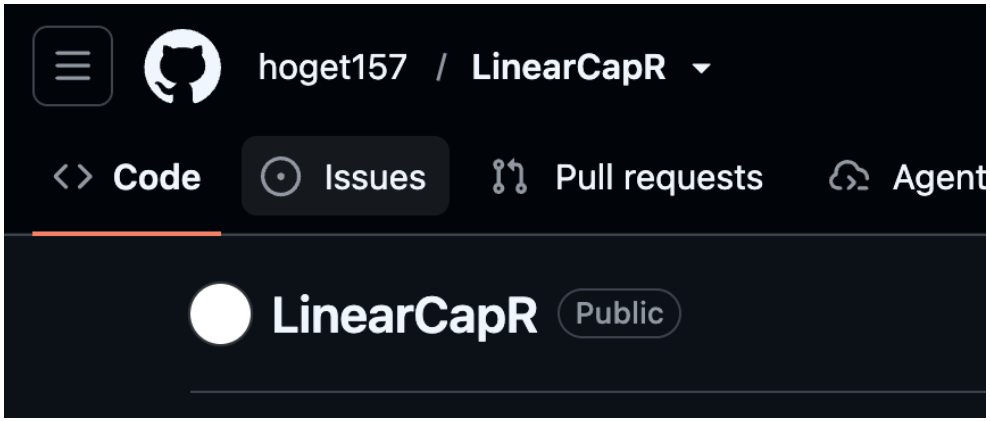


Let’s Demo!

git clone git@github.com:hoget157/LinearCapR.git

cd LinearCapR

make

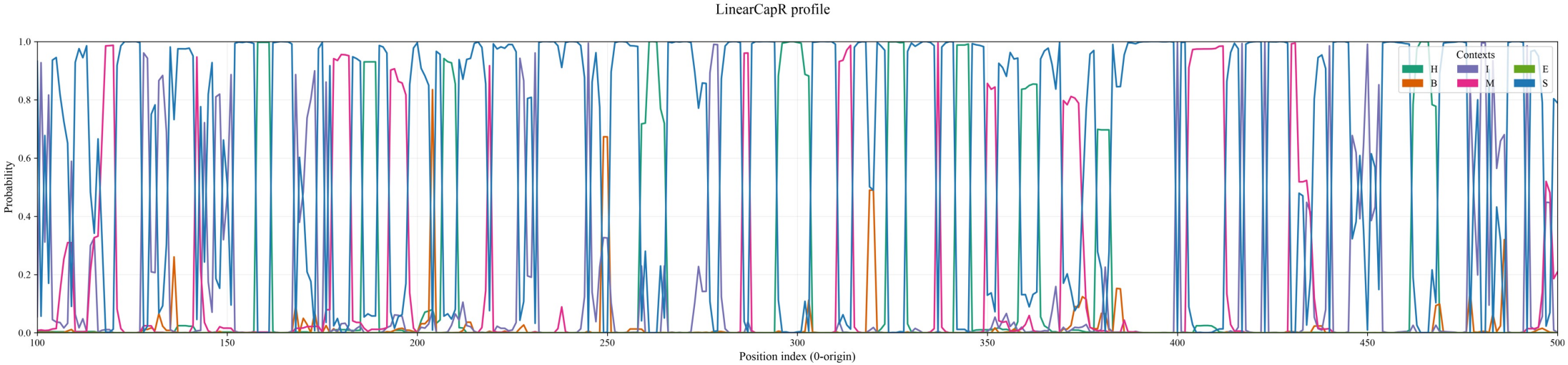


Demo

16S ribosomal RNA from Escherichia coli (PDB 8QK7, chain 2) 1.54kbp

AAAUUGAAGAGUUUGAUCAUGGCUCAGAUUGAACGCUGGCGGCAGGCCUAACACAUGCAAGUCGAACGGUAACAGGAAGA
AGCUUGCUUCUUUGCUGACGAGUGGCGGACGGGUGAGUAAUGUCUGGGAAACUGCCUGAUGGAGGGGGAUAACUACUGG
AAACGGUAGCUAAUACCGCAUAACGUCGCAAGACCAAAGAGGGGGACCUUCGGGCCUCUUGCCAUCGGAUGUGCCCAGAUG
GGAUUAGCUAGUAGGUGGGGUAACGGCUCACCUAGGCGACGAUCCCUAGCUGGUCUGAGAGGAUGACCAGCCACACUGGA
ACUGAGACACGGUCCAGACUCCUACGGGAGGCAGCAGUGGGGAAUAUUGCACAAUGGGCGCAAGCCUGAUGCAGCCAUGC
CGCGUGUAUGAAGAAGGCCUUCGGGUUGUAAAGUACUUUCAGCGGGGAGGAAGGGAGUAAAGUUAAUACCUUUGCUCAU
UGACGUUACCCGCAGAAGAAGCACCGGCUAACUCCGUGCCAGCAGCCGCGGUAAUACGGAGGGUGCAAGCGUUAAUCGGAA
UUACUGGGCGUAAAGCGCACGCAGGCGGUUUGUUAAGUCAGAUGUGAAAUCCCCGGGCUCAACCUGGGAACUGCAUCUGA
UACUGGCAAGCUUGAGUCUCGUAGAGGGGGGUAGAAUUCCAGGUGUAGCGGUGAAAUGCGUAGAGAUCUGGAGGAAUAC
CGGUGGCGAAGGCGGCCCCCUGGACGAAGACUGACGCUCAGGUGCGAAAGCGUGGGGAGCAAACAGGAUUAGAUACCCUG
GUAGUCCACGCCGUAAACGAUGUCGACUUGGAGGUUGUGCCCUUGAGGCGUGGCUUCCGGAGCUAACGCGUUAAGUCGAC
CGCCUGGGGAGUACGGCCGCAAGGUUAAAACUCAAAUGAAUUGACGGGGGCCCGCACAAGCGGUGGAGCAUGUGGUUUAA
UUCGAUGCAACGCGAAGAACCUUACCUGGUCUUGACAUCCACGGAAGUUUUCAGAGAUGAGAAUGUGCCUUCGGGAACCG
UGAGACAGGUGCUGCAUGGCUGUCGUCAGCUCGUGUUGUGAAAUGUUGGGUUAAGUCCCGCAACGAGCGCAACCCUUAU
CCUUUGUUGCCAGCGGUCCGGCCGGGAACUCAAAGGAGACUGCCAGUGAUAAACUGGAGGAAGGUGGGGAUGACGUCAA
GUCAUCAUGGCCCUUACGACCAGGGCUACACACGUGCUACAAUGGCGCAUACAAAGAGAAGCGACCUCGCGAGAGCAAGCG
GACCUCAUAAAGUGCGUCGUAGUCCGGAUUGGAGUCUGCAACUCGACUCCAUGAAGUCGGAAUCGCUAGUAAUCGUGGAU
CAGAAUGCCACGGUGAAUACGUUCCCGGGCCUUGUACACACCGCCCGUCACACCAUGGGAGUGGGUUGCAAAAGAAGUAG
GUAGCUUAACCUUCGGGAGGGCGCUUACCACUUUGUGAUUCAUGACUGGGGUGAAGUCGUAACAAGGUAACCGUAGGGG
AACCUGCGGUUGGAUCACCUCCUUA

https://rnacentral.org/rna/URS00000ABFE9/562


LinearCapR profile
Probability
Position index (0-origin)
Contexts
H
I
E
B
M
S
0.0
0.2
0.4
0.6
0.8
1.0
100
150
200
250
300
350
400
450
500

Discussion and future work

Discussion:

- LinearCapR computes CapR-style structural-context posterior probabilities in linear time while retaining long-range base-pair interactions.
- The main practical cost is increased memory usage.

Future work:

- compressed state encodings and adaptive beam control
- integration with SHAPE reactivity and CLIP-seq data
- viral-genome and lncRNA-scale applications
- differentiable loop profiles for energy-parameter estimation

RNApolis / cli2rest-bio

Tomasz Żok

RNApolis

A Python toolkit for RNA structural bioinformatics

📦 `pip install rnapolis` · tzok/rnapolis-py · 📖 Docs

MIT License · 23★ · 6 forks · 594 commits · by Tomasz Żok et al.

What it does

Parse, annotate, filter and cluster RNA 3D structures from PDB/mmCIF files — all from a Python library or the command line.

Key tools

Category	CLI utilities
Annotation	`annotator` (3D → 2D structure), `adapter` (parses FR3D/DSSR/RNAView/... outputs)
Analysis	`clashfinder`, `motif-extractor`, `metareader`
Transformation	`transformer`, `unifier`, `splitter`, `quick-filter`, `molecule-filter`
Clustering	`distiller` (PCA / nRMSD + hierarchical or affinity propagation)
Prediction	`rfam-folder` (Rfam consensus folding via Infernal)

Highlights

- Detects canonical & non-canonical base pairs (Leontis-Westhof + Saenger), stacking, base-ribose and base-phosphate interactions
- Exports BPSEQ, CSV, JSON, DOT and dot-bracket (first-class support for pseudoknots)
- Fast 3D structure clustering with dimensionality reduction
- One-call mmCIF filtering, splitting and metadata extraction
- Jupyter notebooks with examples

cli2rest-bio

Bioinformatics CLI tools turned into REST APIs

tzok/cli2rest-bio · 🚀 Powered by cli2rest

Companion project to **RNApolis-py** & **cli2rest** · 285 commits · 19 tool images · by Tomasz Żok et al.

What it does

Provides ready-to-build Dockerfiles and CI/CD pipelines that wrap popular RNA / structural-bioinformatics command-line tools in reproducible, version-locked container images served via a REST interface.

Highlights

- **Reproducible builds**: exact source pinning, digest-tracked base images, transitive dependency lock-files (`uv` / `pip-compile --generate-hashes`)
- **Auto-publishing**: GitHub Actions build & push to GHCR on every update; Dependabot keeps dependencies fresh
- **Hardened Dockerfiles**: multi-stage builds, wheelhouse pattern, builder-stage toolchains removed for minimal runtime attack surface
- **Unified interface**: every tool speaks the same REST contract via `cli2rest`

Python Interface

```python
import fuzzyfold as ff

# Co-transcriptional mode modfications and shift moves:
seq = "UGCCUAGAGAGPCAGGPGAU"
ssa = ff.Simulator(params = "rna_extended", k0 = 1, k3ws = 1, k4ws = 1)
print(f"{seq}")
for ss, en, gtime, _tinc, _wtime in ssa.simulate(seq, None, t_ext = 40, t_end = 40):
    print(f"{ss} {en/100:6.2f} {gtime:12.8e}")
print(f"{seq}")
```

Preprint available: