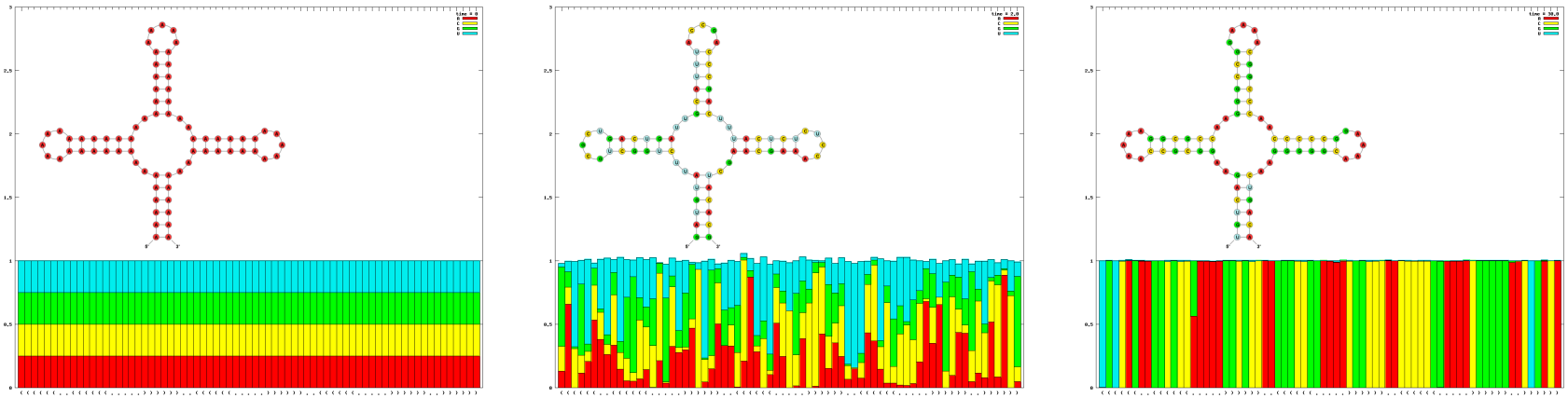
Continuous sequence representations and differentiable dynamic programming for RNA sequence design

Marco Matthies, University of Hamburg
Andrew Torda Lab

Benasque, 11.08.2022

Introduction: continuous RNA sequence optimisation



- sequence represented by $4 \times n$ matrix (prob. distribution for A,C,G,U at each position)
- start at equidistribution, end at "one-hot" sequence
- why? number of sequences grows as 4^n, distance from center of sequence prob. dist. to a corner grows as $\sqrt{n}$
- solve discrete optimisation problem by solving a continuous one

Differentiable optimisation of RNA sequence distributions

- differentiable design scoring function (heuristic)
  - positive design: $\mathbb{E}_s[\Delta G(s, \omega_t)]$, expected free energy over sequence prob. dist. and target structure ω_t
  - negative design: mean-field interactions for unwanted base pairs and local sequence heterogeneity term
- optimise sequence probabilities by dynamical simulated annealing
- works ok, but room for improvement:
  - generates many sequences quickly, but not all are good
  - fine-tuning of negative and positive design terms depending on target structure
  - approach doesn't easily extend to other design tasks (e.g. mRNA design)

Matthies, Bienert, Torda. Dynamics in Sequence Space for RNA Secondary Structure Design. *J. Chem. Theory Comput.*, 2012.

A less ad-hoc design criterion for continuous sequences

- expected probability of target structure

$$\mathbb{E}_s[p(\omega_t|s)] = \mathbb{E}_s[\frac{e^{-\beta\Delta G(s,\omega_t)}}{Q(s)}] \approx \frac{\mathbb{E}_s[e^{-\beta\Delta G(s,\omega_t)}]}{\mathbb{E}_s[Q(s)]}$$

- expected Boltzmann factor

$$\mathbb{E}_s[e^{-\beta\Delta G(s,\omega_t)}] = \mathbb{E}_s[\prod_L e^{-\beta\Delta G_L(s_L)}] = \prod_L \mathbb{E}_{s_L}[e^{-\beta\Delta G_L(s_L)}]$$

Notes

- expectation values are over sequence probability distribution $\mathbb{E}_s[f(s)] = \sum_s p(s)f(s)$
- sequence probability distribution is assumed to be independent at each site $p(s) = \prod_i p(s_i)$
- additive loop contributions to free energy $\Delta G(s,\omega_t) = \sum_{L\in\omega_t} \Delta G_L(s_L)$

$\mathbb{E}_s[Q(s)]$ for the Nussinov-Jacobson energy model

$$Q(i,j) = Q(i,j-1) + \sum_k Q(i,k-1)Q(k+1,j-1)e^{-\beta b(k,j)}$$

$$\begin{aligned}\mathbb{E}_s[Q(i,j)] = &\mathbb{E}_s[Q(i,j-1)] \\ &+ \sum_k \mathbb{E}_s[Q(i,k-1)]\,\mathbb{E}_s[Q(k+1,j-1)]\,\mathbb{E}_s[e^{-\beta b(k,j)}]\end{aligned}$$

- this uses the expectation semiring which has been used in natural language processing (Eisner 2001, Goodman 1999)
- expected base pair probabilities can be computed only approximatively due to divisions in the calculation
- as seq. prob. become "pure" error of approximations converges to zero

Eisner. Expectation semirings: Flexible EM for learning finite-state transducers. *FSMNLP*, 2001.
Goodman. Semiring Parsing. *ACL*, 1999.

Differentiable dynamic programming

- gradients of these dynamic programming algorithms with respect to sequence probabilities via automatic differentiation
- gradient-based optimisation of sequence probabilities, surface seems to be smooth
- has worked well in small-scale testing in the Nussinov-Jacobson model
- can combine differentiable dynamic programming algorithms with other differentiable models, such as neural networks for 5'-UTRs in mRNAs

Differential dynamic programming has been used in natural language processing, e.g. (Mensch & Blondel 2018).

Mensch, Blondel. Differentiable Dynamic Programming for Structured Prediction and Attention. *ICML*, 2018.