Aalto University
School of Science

CS-E4880
Machine Learning in Bioinformatics

Reinforcement Learning
(Introduction + Inverse Folding)

Mehdi Saman Booy

mehdi.samanbooy@aalto.fi

Contents

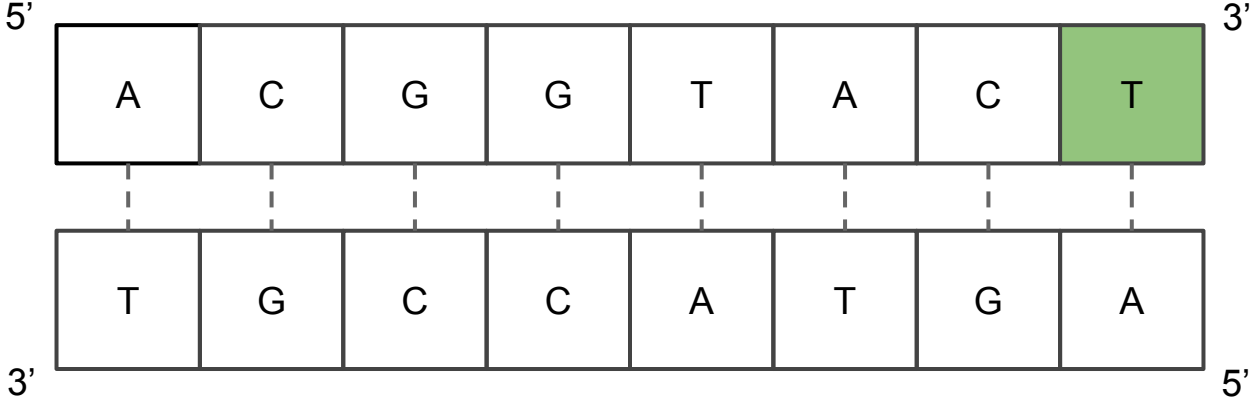
- RNA Nanotechnology (Biological background)
- Reinforcement Learning (Machine Learning background)
- RL in RNA Inverse Folding (Application)

DNA/RNA Structure Prediction

Aalto University
School of Science

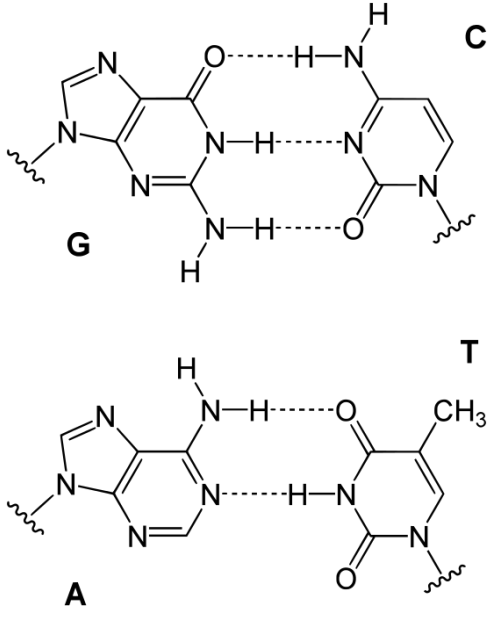
DNA

- Stands for Deoxyribonucleic acid
- (Probably) the carrier of genetic information
- Double strands
- Nucleotide (base)



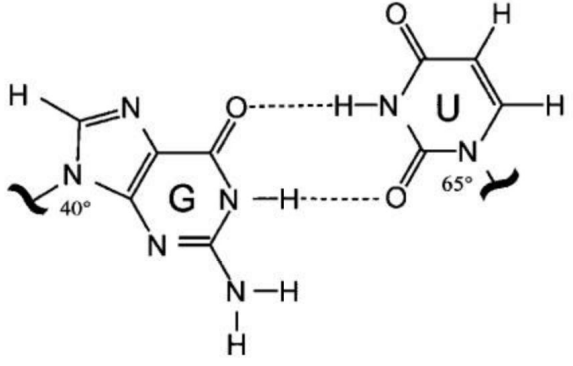
DNA (Nucleotides)

- 4 different nucleotides
  - Cytosine [C]
  - Guanine [G]
  - Adenine [A]
  - Thymine [T]
- Watson-Crick base pairing
  - C-G / A-T
- G-C is stronger (3 H-bonds vs. 2 in A-T)



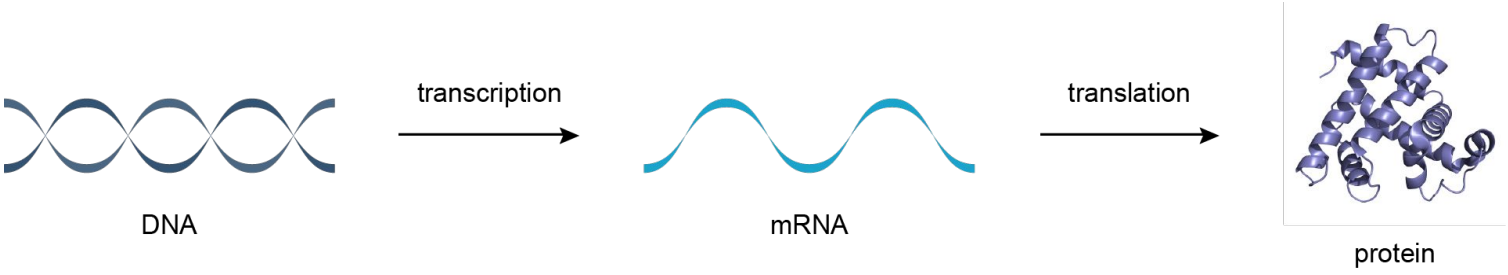
RNA

- DNA is transcribed into Ribonucleic Acid
- Single strand
- 4 nucleotides (A, Uracil, G, C)
- Base-pairing
  - W-C (A-U / G-C)
  - Wobble (U-G)



But what does base-pairing mean when RNA is single strand?
(in a couple of slides we'll see)

More information



Sequence -> Structure -> Function

But we are not always looking for the function!

DNA Nanotechnology

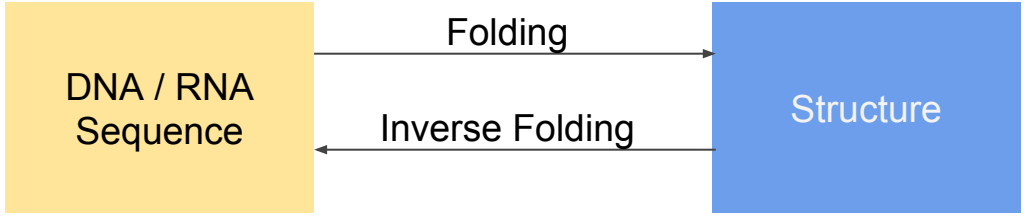
"The nucleic-acid "system" that operates interrestrial life is optimized (through evolution) chemistry incarnate. Why not use it? Not to make genetic manipulations … But to allow human beings to sculpt something new, perhaps beautiful, perhaps useful, certainly unnatural. As beautiful and unnatural as a Schubert song or the American Constitutions."

[DNA as Clay, Roald Hoffmann, 1994]

DNA / RNA Nanotechnology

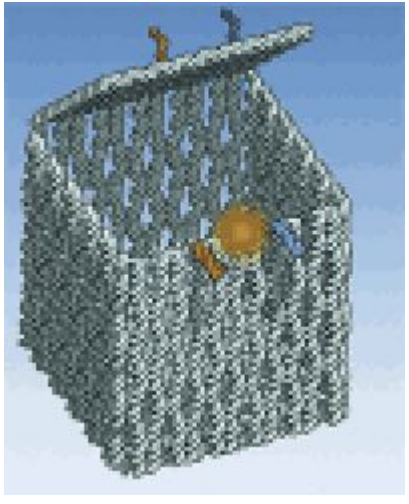
Using DNA / RNA as structural elements to build nano-scale objects

- We can synthesise them (Synthetic Biology)
- Then, we can fold them. Thanks to Self-assembly



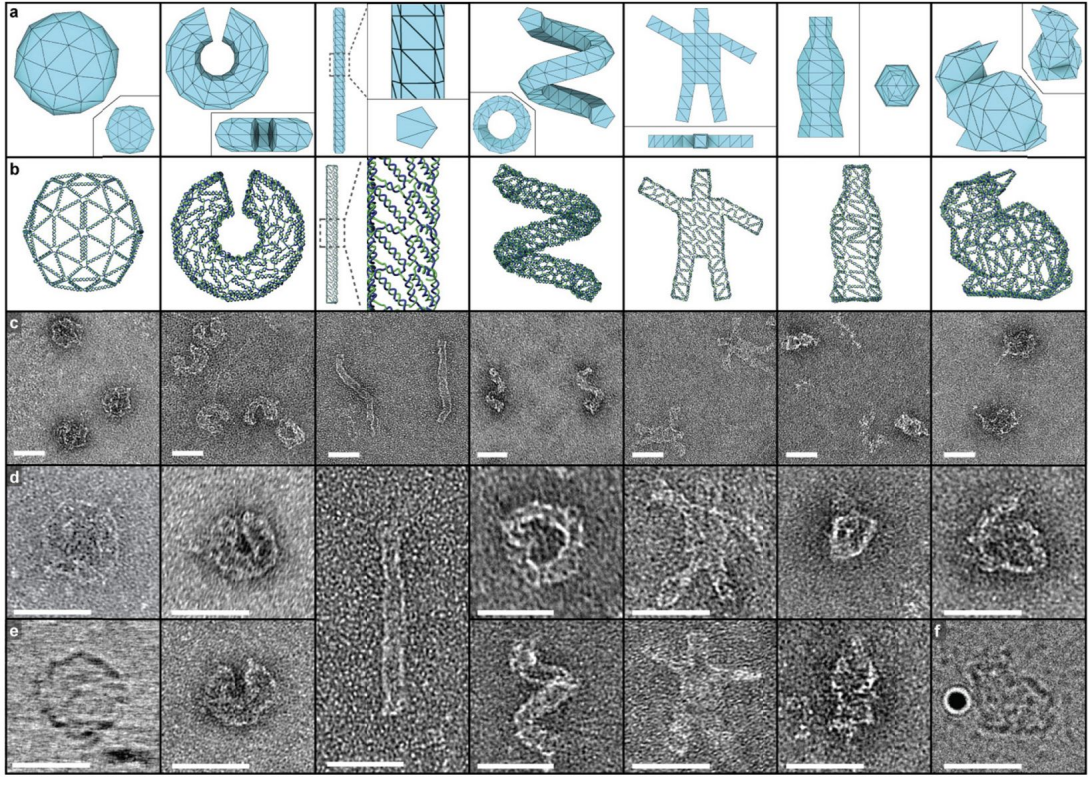
(We will focus on RNA …)

Some Cool Examples



Nano-scale box with a lock

Some Cool Examples (cont'd)



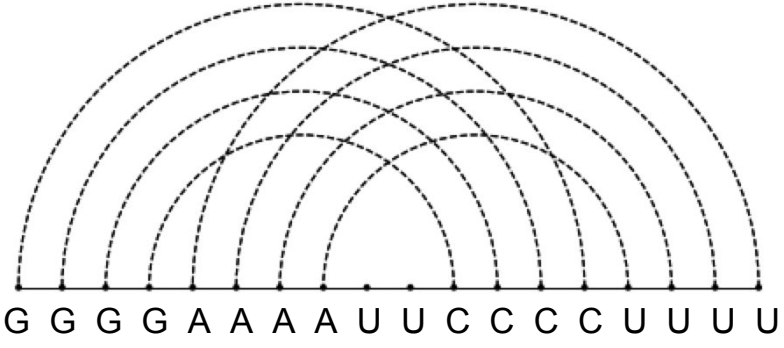
Benson, Erik; Mohammed, Abdulmelik; Gardell, Johan; Masich, Sergej; Czeizler, Eugen; Orponen, Pekka; Högberg, Björn | Nature, 2015]

Structural Level in RNA

Primary structure	Linear sequence of bases from 5’ to 3’
Secondary structure	Pairing arrangement of bases in primary structure
PK-Free Secondary structure	PKed Secondary structure
Tertiary structure	3D shape of secondary structure

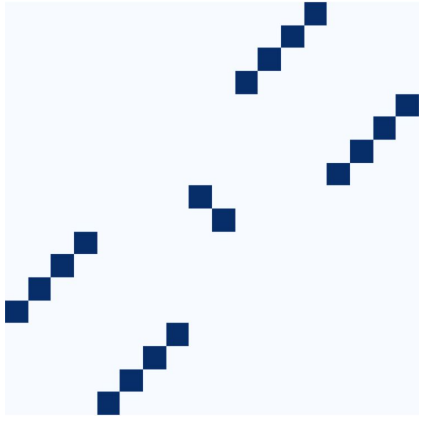
RNA Structure Presentation

For a sequence like GGGGAAAAUUCCCCUUUU



Arc Diagram

GGGGAAAAUUCCCCUUUU



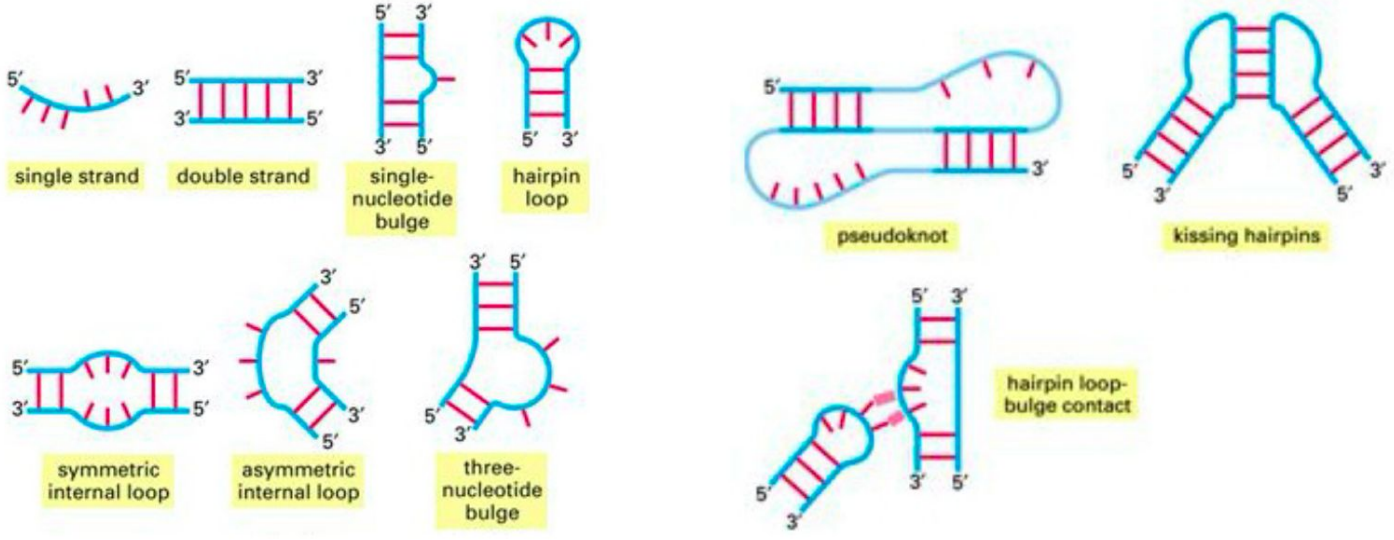
Pairing matrix

[13, 12, 11, 10, 17, 16, 15, 14, -1, -1, 3, 2, 1, 0, 7, 6, 5, 4] **Pairing list**

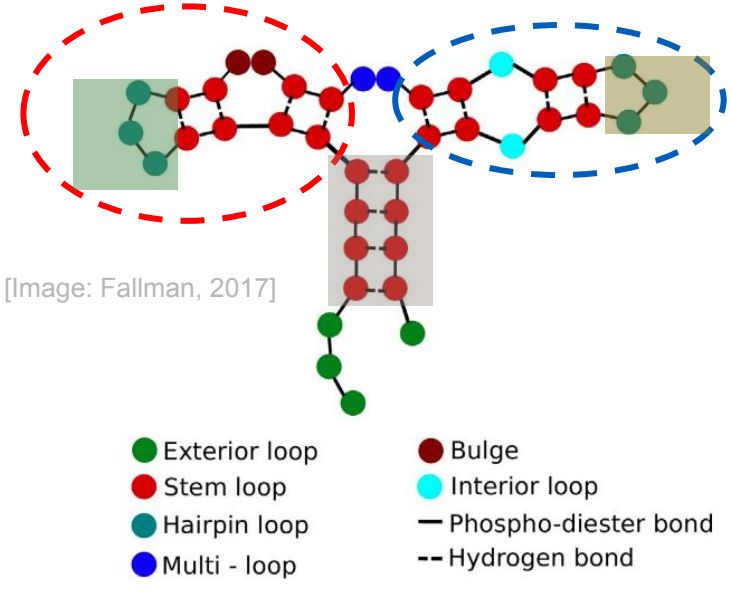
(((([[[[. .))))]]]] **Dot-bracket**

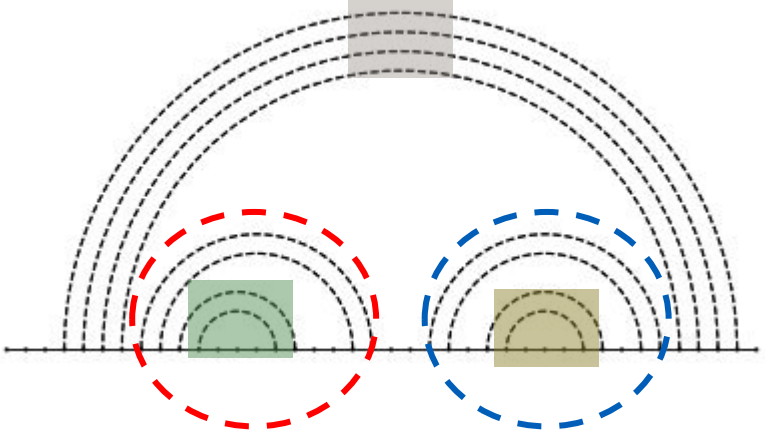
((((((((. .)))))))) **Dot-parenthesis** → **Not working when we have crossover**

RNA Secondary Structure Motifs



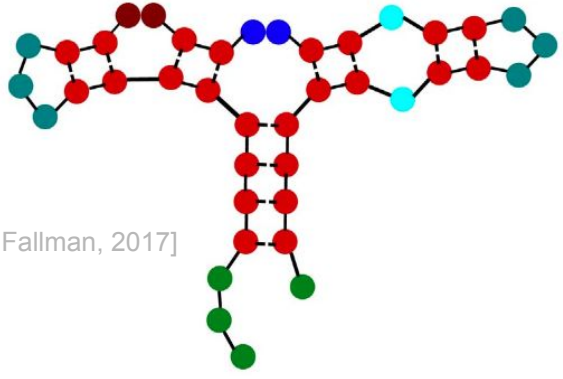
Pseudoknot-Free Secondary Structure



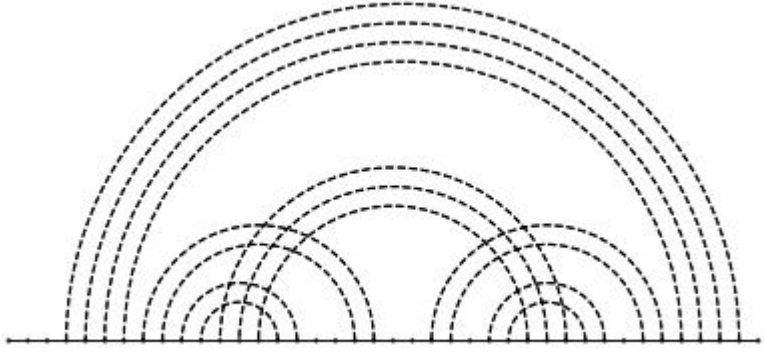


Pseudoknotted Secondary Structure

If we connect 2 hairpins together



[Image: Fallman, 2017]



No concrete definition, but "***Crossing in arc diagram***".
(or) If we cannot represent the structure using dot-parenthesis model.

Pseudoknotted Secondary Structure (cont'd)

Of course we can have mathematical definition of PK-ed structure.

For a valid structure s, $PK(s)$ is defined in $\{0,1\}$ in this way:

$$PK(s) = \begin{cases} 1 : \exists i,j,k \mid i < k < j \Rightarrow (s_k > j \vee s_k < i) \\ 0 : O.W. \end{cases}$$

But there is no standard definition.

RNA Folding

- An RNA strand folds (asymptotically) to its Minimum Free Energy (MFE) structure (native structure).
- Why?
  - Nature wants to decrease the FE
- What is FE?
  - Actually, we don’t know exactly!!!
  - But we have some experimental data (e.g. Turner model)

Finding MFE secondary structure is **NP-hard**.

Energy Models

- Each (sequence, structure) pair has a **Free Energy** value
- FE is additive (Tinoco, 1973)
  - FE(structure) = FE(stem) + FE(interior_loop) + FE(hairpin_loop) + …
- FE of each motif is calculated based on experimental tables

```
>Hairpin Loop Energies: size = 1,2,3,..,30
   0    0  570  560  560  540  590  560  640  650  660  670  680  690  690  700  710  710  720  720  730  730  740  740  750  750  750  760  760  770
>Bulge loop Energies: size = 1,2,3,..,30
 380  280  320  360  400  440  460  470  480  490  500  510  520  530  540  540  550  550  560  570  570  580  580  580  590  590  600  600  600  610
>Interior Loop Energies: size = 1,2,3,..,30
   0    0    0  170  180  200  220  230  240  250  260  270  280  290  300  300  310  310  320  330  330  340  340  340  350  350  360  360  360  370
```

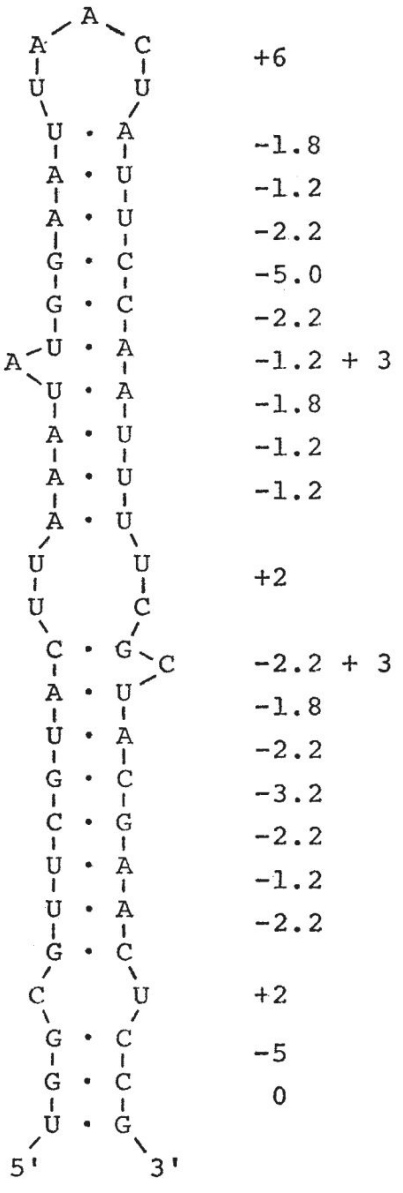
(About 4k other lines like these!)

Nearest Neighbor model

- Tinoco, 1973
- FE is additive
- Stems have negative FE
- Hairpins, bulges, interior-loops has positive

Table 1 Free Energies at 25° C for RNA Structures

Base paired regions	ΔG(kcalorie) ± 10%
-A-A- -U-U-	−1.2
-A-U- -U-A , -U-A- -A-U-	−1.8
-A-C- -U-G- , -C-A- -G-U- , -A-G- -U-C- , -G-A- -C-U-	−2.2
-C-G- -G-C-	−3.2
-G-C -C-G , G-G- C C-	−5.0
-G-U- -U-G-	−0.3
-G-X- -U-Y- , -X-G- -U-Y-	0



2 Useful Tools

NUPACK

NUPACK: For the analysis and design of nucleic acid structures, devices, and systems.

ViennaRNA: Prediction and comparison of RNA secondary structures.

FE calculation, Secondary Structure Prediction,
Sequence Design (without PK)

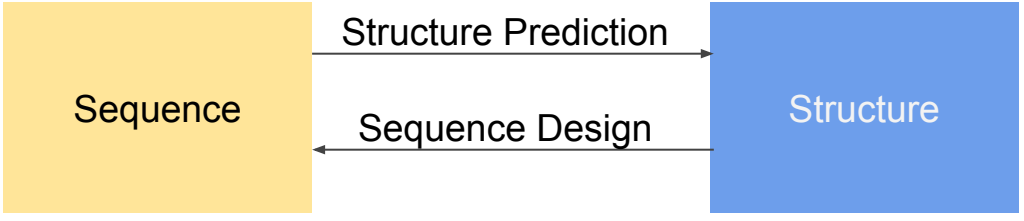
Problems in Computer Science

Structure Prediction:

Find MFE (secondary for now!) structure given a sequence

Sequence Design:

Find a sequence which its MFE is the target structure



Structure Prediction (Folding)

Inputs

Sequence *q* with length N

Energy model (*F(q) = fe*) -> (#basepairs / FE / …)

Output

Structure *s* w.r.t sequence *q* and energy model *F*

Structure Prediction (Folding) (cont'd)

- **Naive solution:**

  Search the whole space (all possible structures for q)

Search space is huge ~ O(n!!)

However, candidates are a few. For q seq with n=10, we have ~3.7m structures, but only 20 has FE<0 (FE of an totally unpaired structure is zero, so the Nature prefers to do nothing instead of having positive FE).

Structure Prediction (Folding) (cont'd)

- **Dynamic Programming**
- Firstly used by Nussinov and Zuker

AUCGUACUGAUCGUUCGUUCGAUCUGACUGUCGAUCAUGUCGAU

If we know the solution up to here
What would be the solution with adding this one

Not working in general in case of PK!

Sequence Design (Inverse Folding)

- Folding calculates the structure for a give sequence
- But, which sequence should we use to build our desire object?!
- first introduced in (Hofacker et.al., 1994).

- Inputs:
  - Target structure *s*
  - Folding function *F(q) = s'*
- Output:
  - Sequence *q* that its folded structure (*s'*) is the target *s* (or close to s with some distance measure)

Sequence Design (Inverse Folding) (cont'd)

Naive algorithm

Start from a random sequence

While the answer has not been found:

q' <- Randomly change 1 base

Check the MFE structure for q'

Search space is even exponentially huge! ~ O(4^n)

Reinforcement Learning

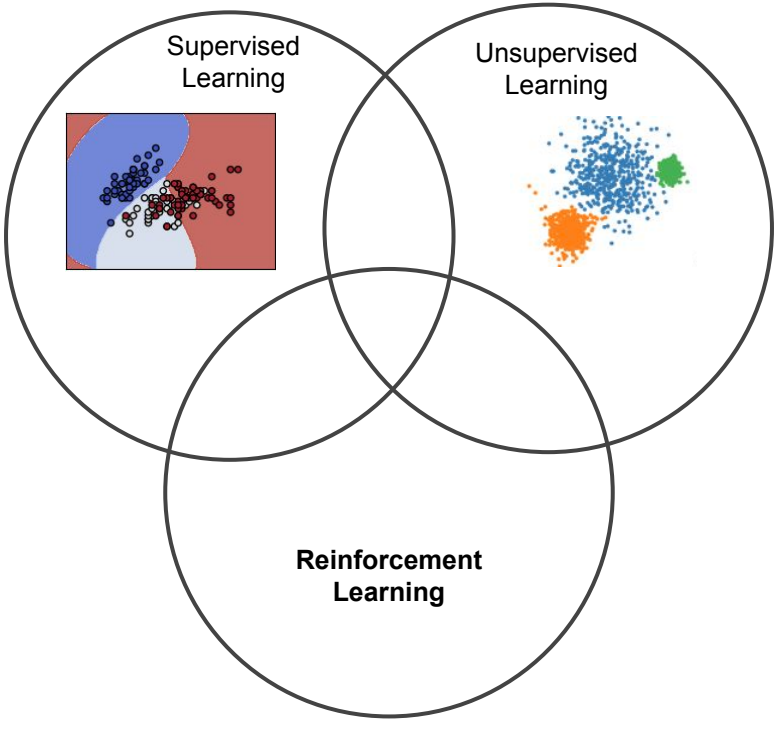
Aalto University
School of Science

2 Useful Resources

- *Reinforcement Learning: An Introduction*, Sutton and Barto, 2018
  - Revision (completely) on the old book
  - Available online!

- UCL Course on RL, David Silver, 2015
  - You can find lectures and slides

Machine Learning Branches



[images from scikit-learn]

Example of a successful story (Atari)

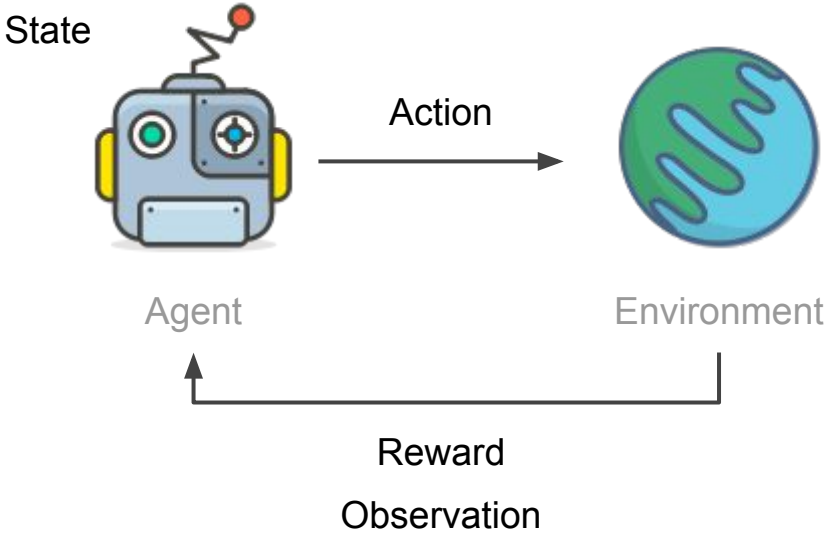

Ti
DEEPMIND AI
LEARNED HOW TO WALK

Why RL?

- No supervision, (in general) no data
  - Only reward
- It's like our learning procedure (trial and error)
- Sequential decision making (MDP*)

(* Markov Decision Process)

RL Elements



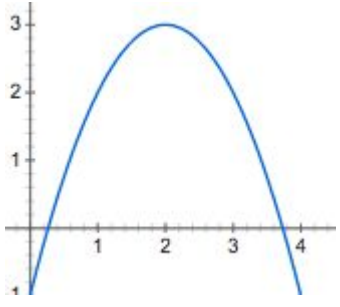
Can you guess the goal of the agent?

Reward

- R_t is a scalar feedback signal from the env. at time t
- We are modeling the goal as a reward function
- Agent wants to maximise cumulative reward
- Examples:
  - Chess: +1: win / 0: tie / -1: loss
  - Atari: -100: losing / +1: hitting each break

    or -1: losing (doesn't work!)

    or +1: hitting a break
  - Optimising F(x_1, x_2, …, x_n): F(x_1, x_2, …, x_n)
    - *It would choose x=2 in this example*

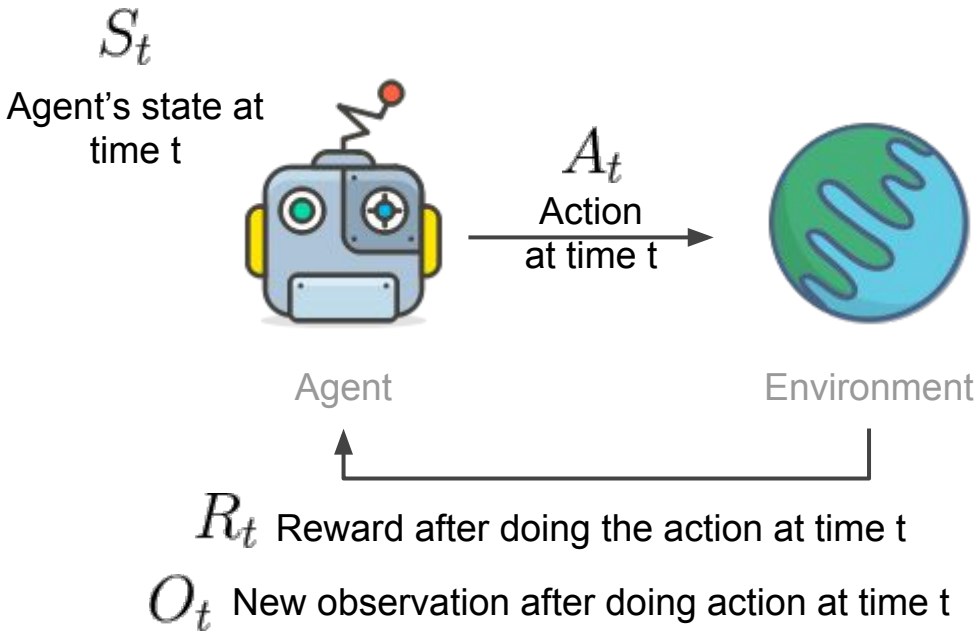


Reward Hypothesis

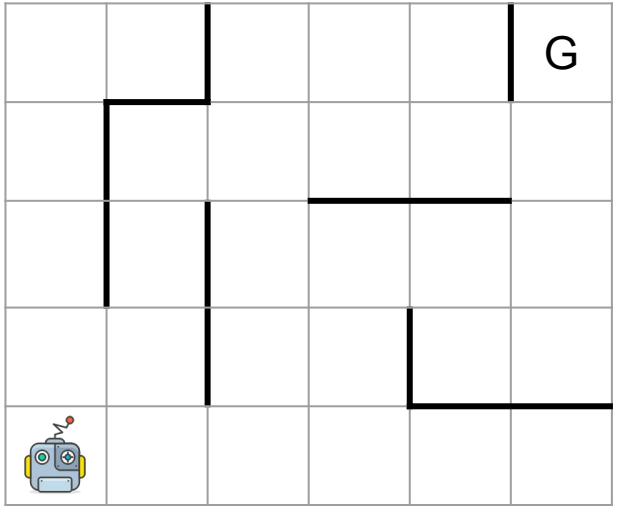
"All goals can be described by the maximization of expected cumulative rewards."

Important note: Expected Cumulative Rewards is not greedy!

Terminology



An example

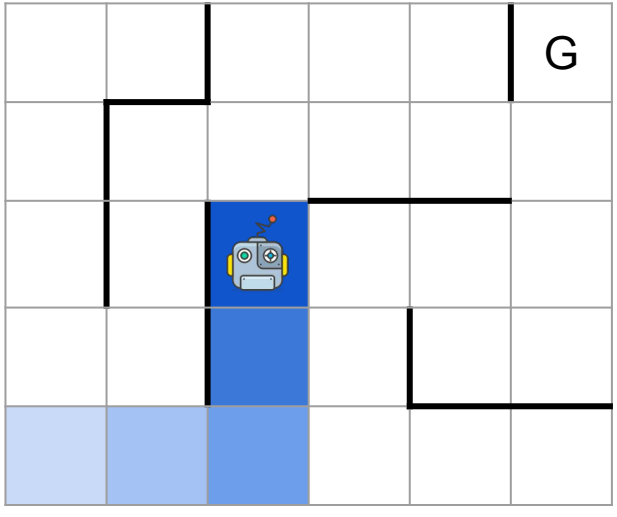


$$S = \{(i, j) : \forall 1 \leq i \leq 5, 1 \leq j \leq 6\}$$

$$A = \{L, R, U, D\}$$

$$R_t = \begin{cases} 100 & : S_t = (5, 6) \\ -1 & : O.W. \end{cases}$$

An example (cont'd)



When the agent is at state (3, 3) and wants to do the next action, does it matter where was it before?

NO, why? (next slide)

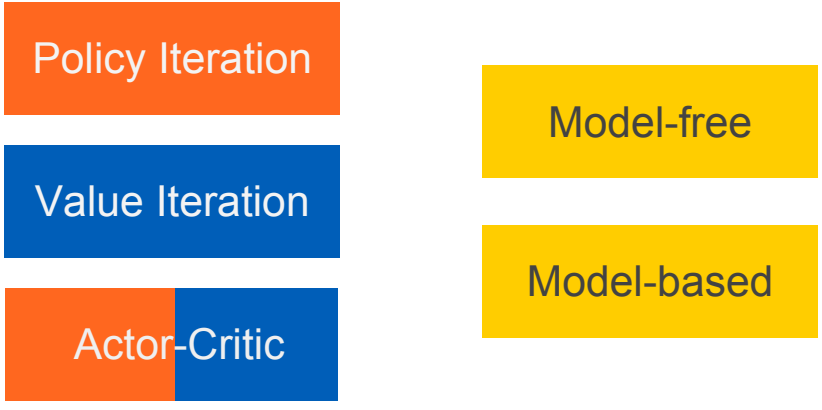
Markov Decision Process

$$\mathbb{P}(S_{t+1}|S_t, S_{t-1}, \ldots, S_1) = \mathbb{P}(S_{t+1}|S_t)$$

If our env. has MDP property, next state depends only on the current one!

Main Components of RL Agent

Agent's type



Policy (π)

- Agent's behavior
- Maps state to action
- Deterministic

$$a = \pi(s)$$

- Stochastic

$$\pi(a|s) = \mathbb{P}(A_t = a|S_t = s)$$

Value Function (V)

- Prediction of future rewards (accumulated rewards)
- How good / bad is a state

$$V_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s]$$

(Gamma is the discount factor)

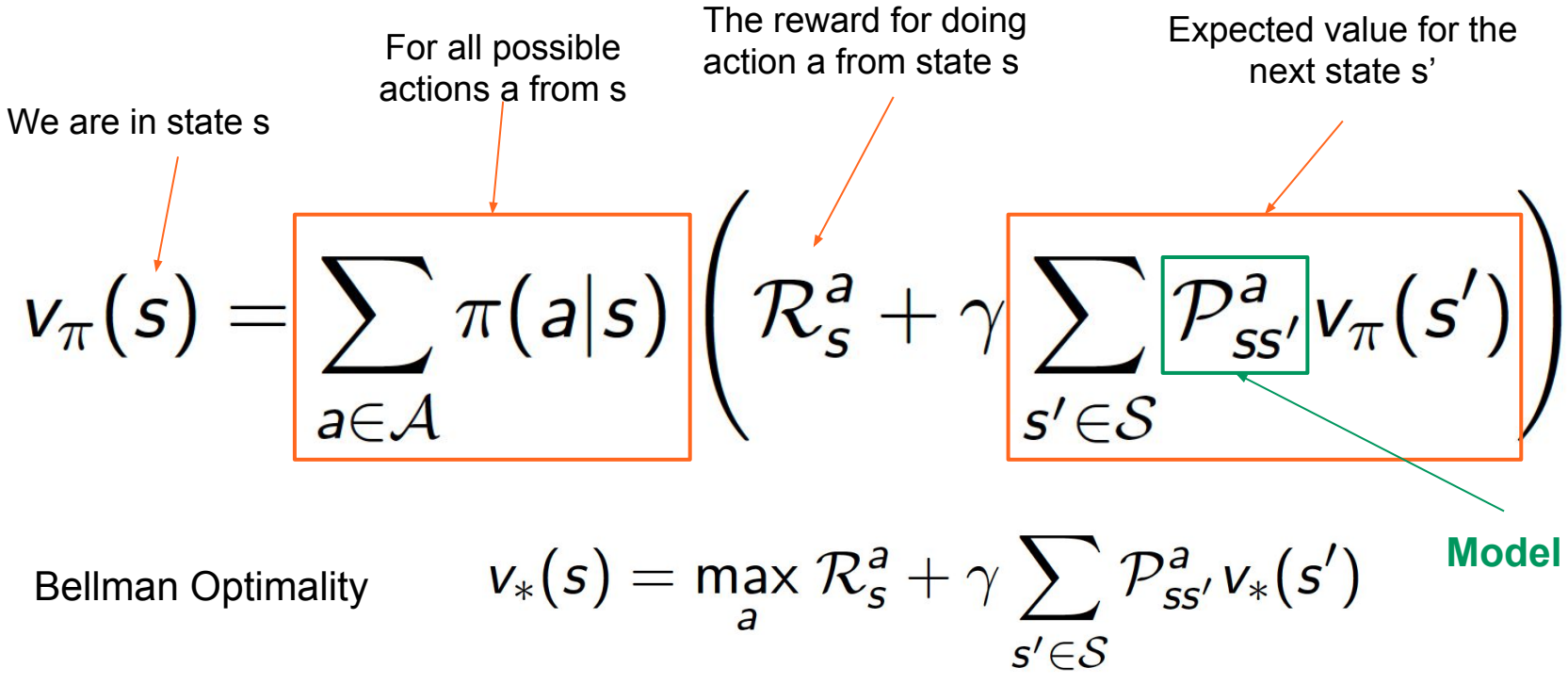
Bellman Equation

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right]$$

So, while true, {doing action, update V}

We should see each (state, action) several times.

Bellman Equation



Simple Example

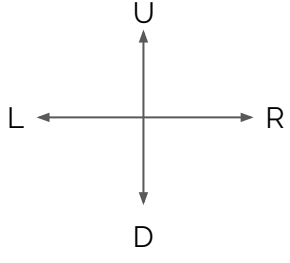
Parameters

R(G) = 90
Gamma = 0.9

Q is the value for (s, a) pairs

3	G
1	2

Environment with states



Actions

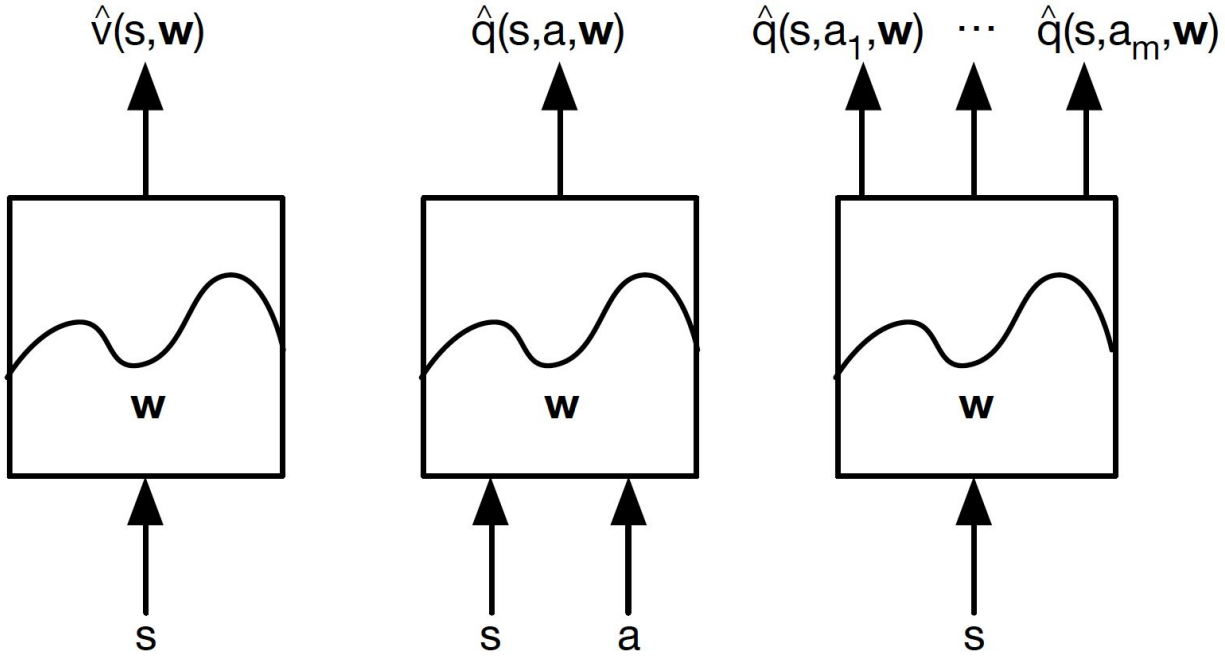
s	a	Q
1	L	71
1	R	81
1	U	80
1	D	70
2	L	63
2	R	64
2	U	89
2	D	70
3	L	70
3	R	90
3	U	80
3	D	63

Tabular vs. Approximation RL

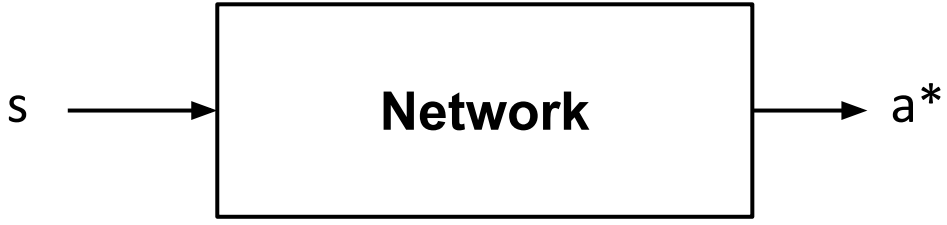
- What we saw before was tabular
  - Save values for each state (or state, action pair)
- What if the |S|x|A| was huge?
  - Continuous space
  - RNA (inverse) folding
  - Go (#states > #atoms in the universe!!)

Approximation methods try to tackle this issue with creating a *model*

Value Function Approximation



Policy Network



However, it can be any estimator (Linear combination of features, decision tree, etc.)

RNA Inverse Folding using RL

Aalto University
School of Science

Why RL?

- Lack of data -> RL learns from experience
- Algorithmic solutions solve each problem independently
  - Nature has the same rules for all folding problems!
- Some new patterns have been seen in previous applications (e.g. AlphaGo)

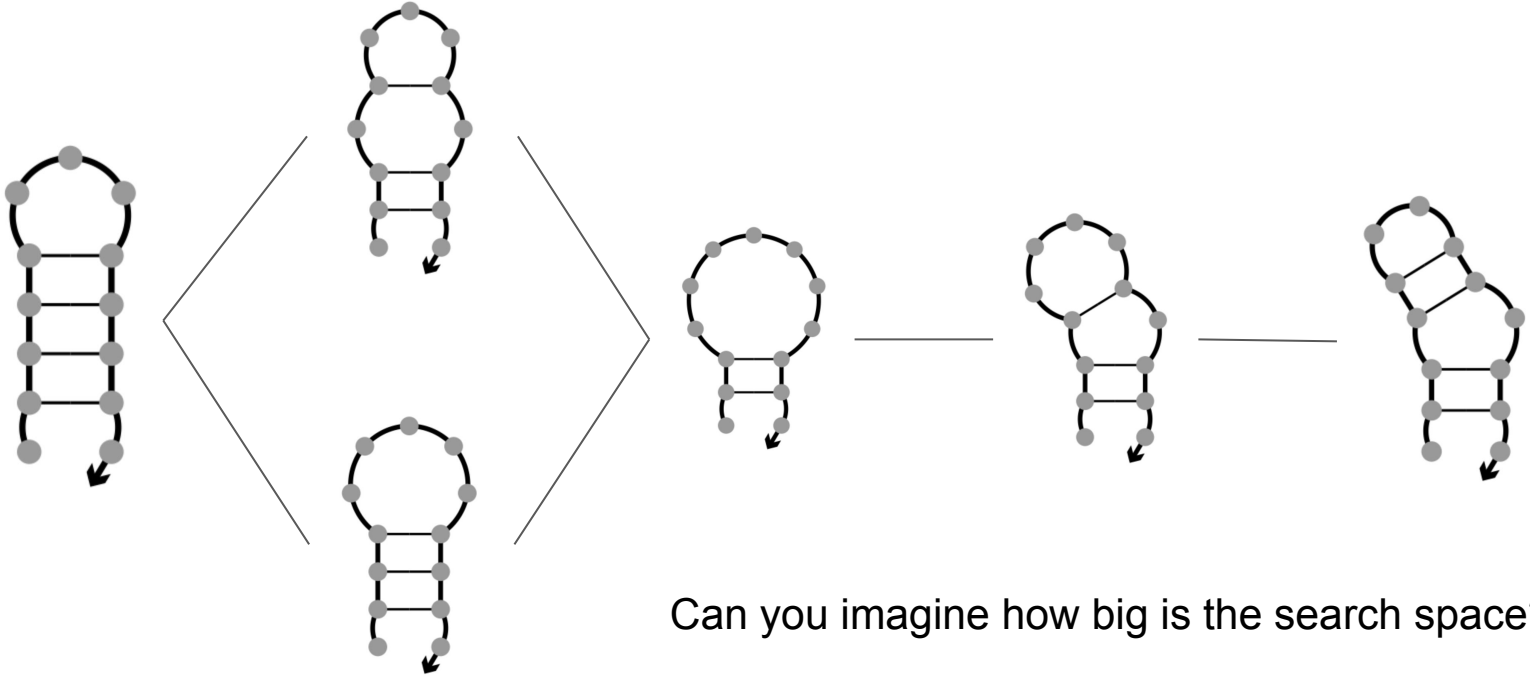
Reinforcement Learning (Folding)

States	Any structure
Actions	Pairing / Unpairing two basis
Valuation Function	Free energy

Reinforcement Learning (Inverse Folding)

States	Any sequence
Actions	Changing any base
Valuation Function	Free energy of target Distance measure for MFE structure

An Example for Folding



Can you imagine how big is the search space?!

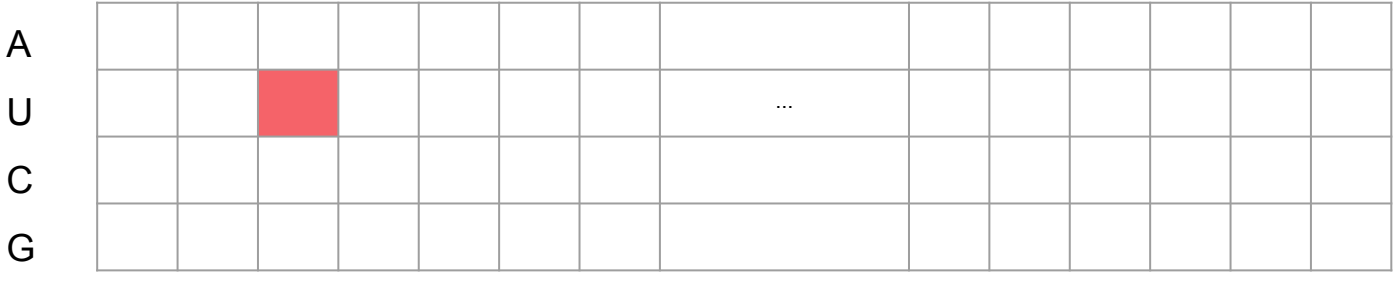
Solving the RNA design problem with reinforcement learning

Peter Eastman[1]*, Jade Shi[2], Bharath Ramsundar[3], Vijay S. Pande[1]

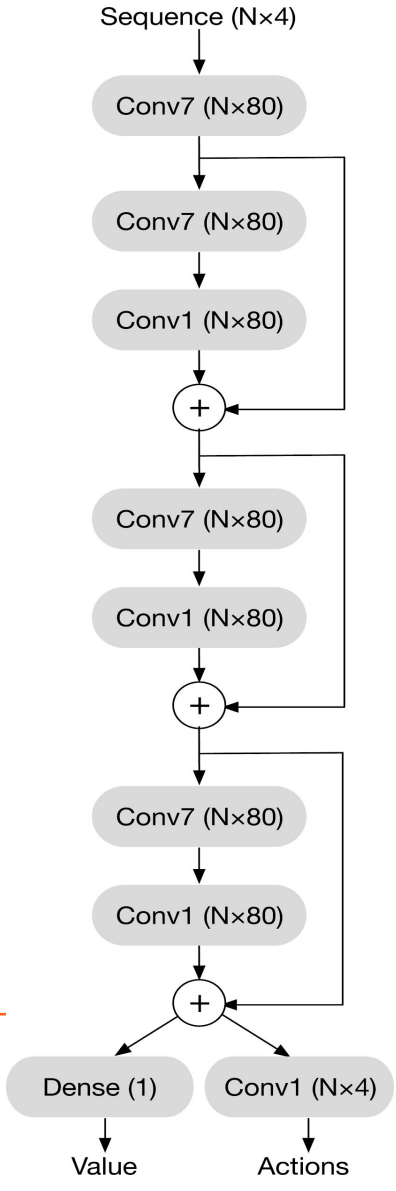
1 Department of Bioengineering, Stanford University, Stanford, CA, United States of America, **2** Department of Chemistry, Stanford University, Stanford, CA, United States of America, **3** Department of Computer Science, Stanford University, Stanford, CA, United States of America

Architecture

- **Input**
  - Nx4 (one-hot sequence)
- **Output**
  - Nx4 (probability)



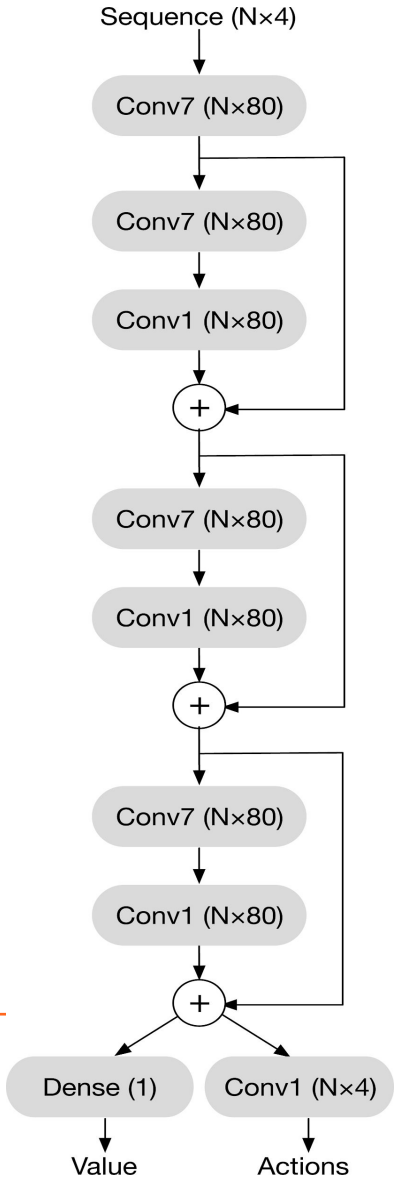
Probability of changing base number 3 to “U” (second nucleotide)



Architecture

- **Conv1**
  - Convolve each base separately
  - Merging kernels from last layer
- **Conv7**
  - Convolve [i-2:i+2]
  - and paired base from target structure
  - and current paired base from current structure

Actor-Critic (Policy + Value)



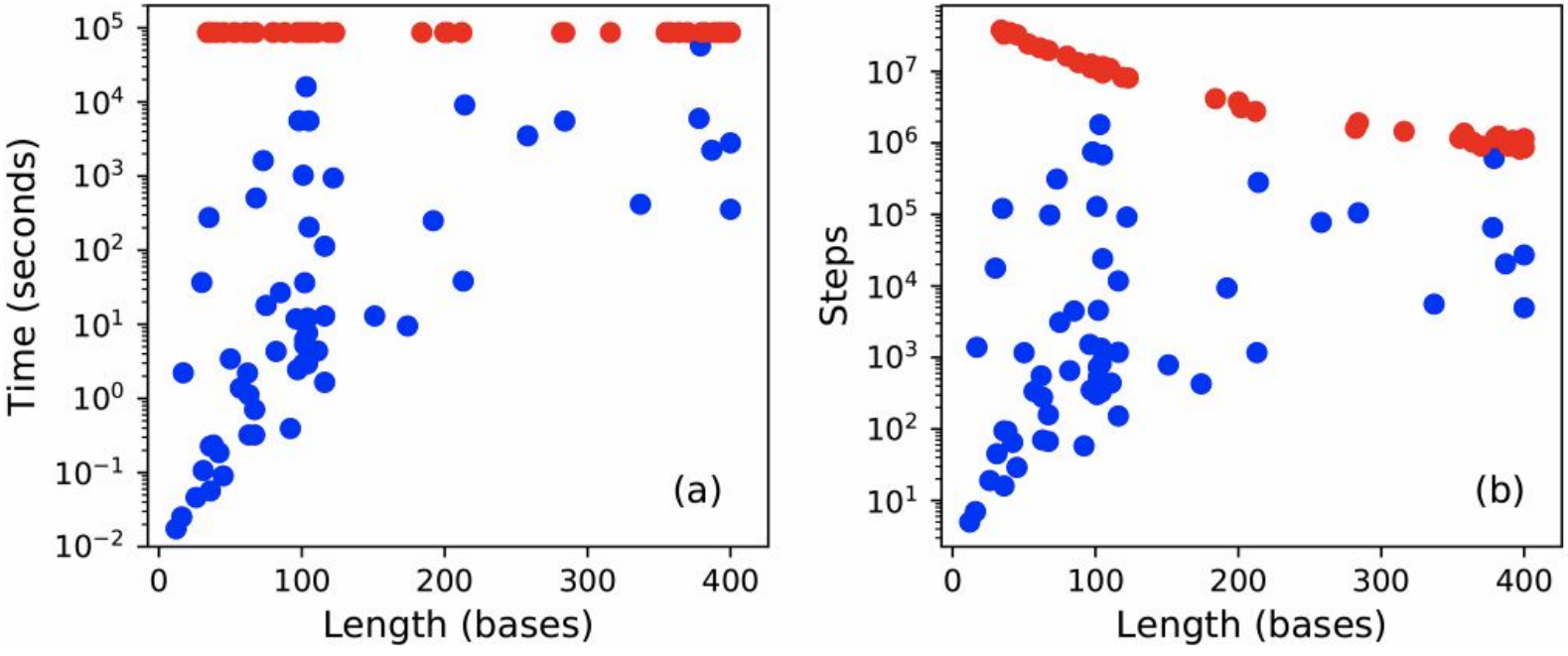
Training

- 100’000 RNA seq of len 32
  - 46’188 unique structures
- A3C (Asynchronous Advantage Actor-Critic) for RL
- 1.5 million episodes
  - First 0.5m : easier samples
  - Others : all except the 500 most difficult samples
  - Last 500 as evaluation set
- ViennaRNA for the folding part

Test

- Start with a random sequence
- Follow the policy until the solution will be find
  - Change base number *i* to *b*
- Or stop the algorithm after 24h

Results



Blue: Solved / Red: failure

Results

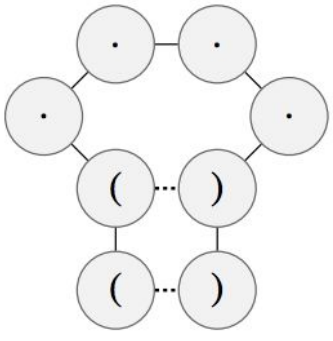
They could solve 60 / 100 samples from Eterna.

Published as a conference paper at ICLR 2019

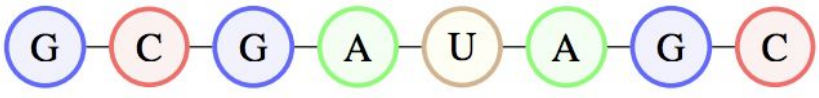
LEARNING TO DESIGN RNA

Frederic Runge*, Danny Stoll*, Stefan Falkner & Frank Hutter
Department of Computer Science
University of Freiburg
{runget,stolld,sfalkner,fh}@cs.uni-freiburg.de

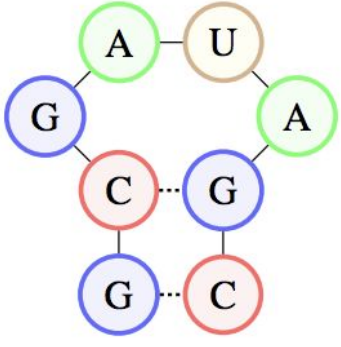
Problem definition (Inverse Folding)



(a) Target structure ω



(b) RNA sequence $\phi \in N^{|\omega|}$



(c) Fold $\mathcal{F}(\phi)$

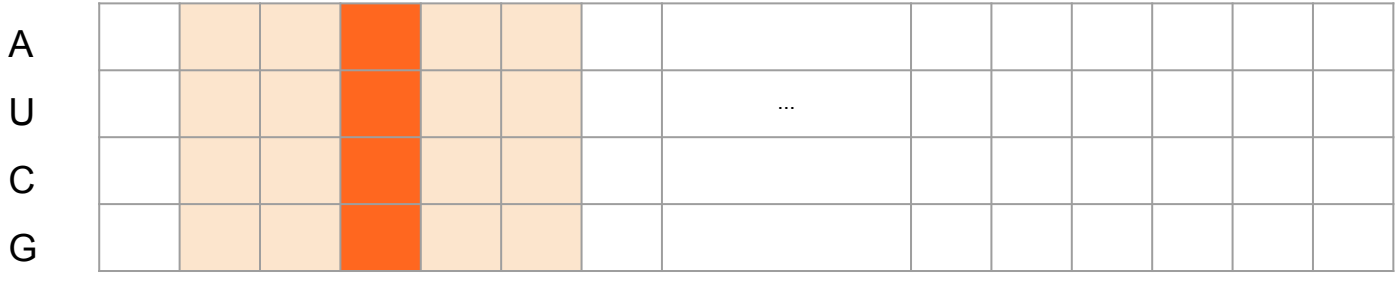
Action Space

$$\mathcal{A} := \{0, 1, 2, 3\} \equiv \begin{cases} \{A, G, C, U\} & \text{for } \mathcal{C}_\omega(t) = . \quad \text{["dot"]} \\ \{GC, CG, AU, UA\} & \text{for } \mathcal{C}_\omega(t) = (\quad \text{["opening bracket"]} \end{cases}$$

- Actions are 0, 1, 2, 3 but has different meaning
- If this base is paired or not in the target structure
- Action a at time t means decision making for t-th base

States

$$\mathcal{S} := \{0, 1, 2, 3\}^{2\kappa+1}$$



k is a hyperparameter to consider the locality

Reward Function

$$\mathcal{R}_\omega^T(\phi) := (1 - L_\omega(\mathcal{F}(\phi)))^\alpha$$

$$L_\omega(\mathcal{F}(\phi)) := \mathrm{d_H}(\mathcal{F}(\phi), \omega) \,/\, |\omega|$$

Hamming distance

Predicted structure for current sequence

Target structure

Alpha is hyperparameter

Network Architecture

It is dynamic consists of

- Input is binary encoded of dot-bracket or an embedding
- Optional CNN with at most 2 layers
- Optional LSTM with at most 2 layers
- Shallow FC with at most 2 layers
- Output is distribution over actions

Hyperparameters

Parameter Name	Type	Range	Prior
filter size in 1st conv layer	integer	$\{0\} \cup \{3, 5, \ldots, 17\}$	uniform
filter size in 2nd conv layer	integer	$\{0, 3, 5, 7, 9\}$	uniform
# filter in 1st conv layer	integer	$[1, 32]$	log-uniform
# filter in 2nd conv layer	integer	$[1, 32]$	log-uniform
# LSTM layers	integer	$[0, 2]$	uniform
# units in every LSTM layer	integer	$[1, 64]$	log-uniform
# fully connected layers	integer	$[1, 2]$	uniform
# units in fully connected layer(s)	integer	$[8, 64]$	log-uniform
state space radius κ	integer	$[0, 32]$	uniform
embedding dimensionality	integer	$[0, 4]$	uniform
batch size	integer	$[32, 128]$	log-uniform
entropy regularization	float	$[1 \cdot 10^{-5}, 1 \cdot 10^{-2}]$	log-uniform
learning rate for PPO	float	$[1 \cdot 10^{-5}, 1 \cdot 10^{-3}]$	log-uniform
reward exponent α	float	$[1, 10]$	uniform

Then do the **architecture search.**

Results

Method	Solved Sequences [%]		
	Eterna100	Rfam-Taneda	Rfam-Learn-Test
MCTS-RNA	57	79	97
antaRNA	58	66	**100**
RL-LS	59	62	62
RNAinverse	60	59	95
LEARNA	67	79	97
Meta-LEARNA	**68**	**83**	**100**
Meta-LEARNA-adapt	**68**	**83**	99

Conclusion

- RNA (DNA) can be used as an structural material to build nano-scale objects w.r.t self-assembly
- To this end, folding and inverse folding are two important challenges to think about (analyze, learn, predict, …)
- Folding and inverse folding are two interesting problems in CS
- RL is a powerful tool, yet not as well-defined as supervised tools
- RL is a perfect fit for folding and inverse folding problems due to lack of data and their complexity

Aalto University
School of Science

Thanks
More Questions?!

Mehdi Saman Booy - March 15th 2019