

Differentiation in RNA sequence Design



Kiyoshi Asai

Department of Computational Biology Research Center & Life Science Data Research Center
University of Tokyo



2009

Acknowledgement

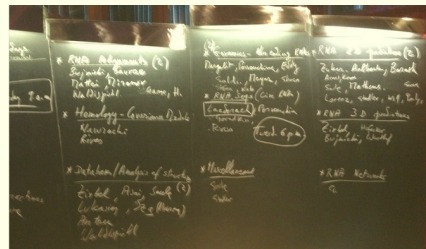
RNA informatics team in CBRC  



JBIC / AIST / NEDO / MHIR

2015

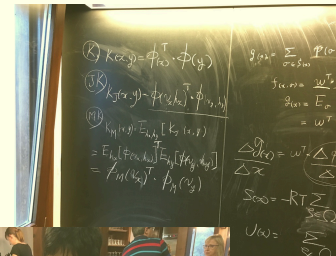
Organizational meeting



Cooking in a hotel

2018

Some formula produced



Excellent Jamón and wine found

2024

Fresh seafood in supermercado!!



Upgraded cooking in an apartment

I love dynamic programming (on RNA)

Scarna	2006	heuristic DP	Structural alignment
Murlet	2007	Sankoff	Structural alignment
McCaskill-MEA	2007	averaged BPP	Common 2D structure
MXSCARNA	2008	heuristic DP	Structural alignment
Rfold	2008	banded SCFG	Exact local BPP
CentroidFold	2009	γ -centroid (MEA)	2D structure prediction
CentroidHomfold	2009	γ -centroid (MEA)	2D structure prediction using homologs
CentroidAlign	2009	γ -centroid (MEA)	Structural alignment
CentroidAlign	2011	γ -centroid (MEA)	Common 2D structure
Rchange	2012	banded SCFG	Effects of mutation on 2D
CapR	2014	banded SCFG	2D structural profile

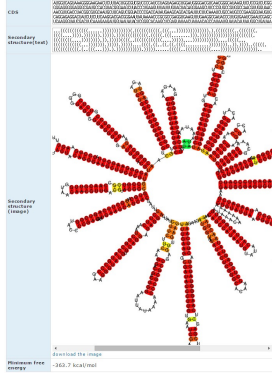
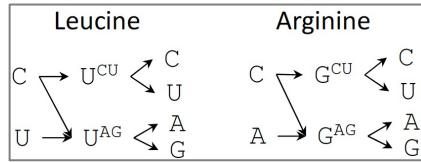
mRNA design

Dynamic Programming

CDSfold (2015) the most stable 2D structure

$$F^{n_i, n_j}(i, j) =$$

$$\min \begin{cases} \min_{n_{i+1} \in N_i | n_i} [F^{n_{i+1}, n_j}(i+1, j)] \\ \min_{n_{j-1} \in N_j \wedge n_j} [F^{n_i, n_{j-1}}(i, j-1)] \\ C^{n_i, n_j}(i, j) \\ \min_{\substack{i+1 < k < j-1 \\ n_k \in N_k \\ n_{k+1} \in N_{k+1} | n_k}} [F^{n_i, n_k}(i, k) + F^{n_{k+1}, n_j}(k+1, j)] \end{cases}$$

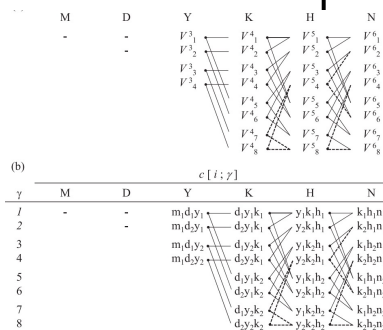


Terai G, Kamegai S., and Asai K, Bioinformatics 32(6), 826-834 (2015)

Genetic algorithm
for multiple
copies of same gene

COSMO (2020) multi-criteria codon optimization

$$V_{\gamma}^k = \begin{cases} \emptyset, & \text{if } \exists \sigma \in M, \sum_{j=1}^k N_{\text{end}}^{\sigma}(j) \neq 0, \\ \sum_{j=1}^k \tau(c_1, \dots, c_j), & \text{otherwise,} \end{cases}$$

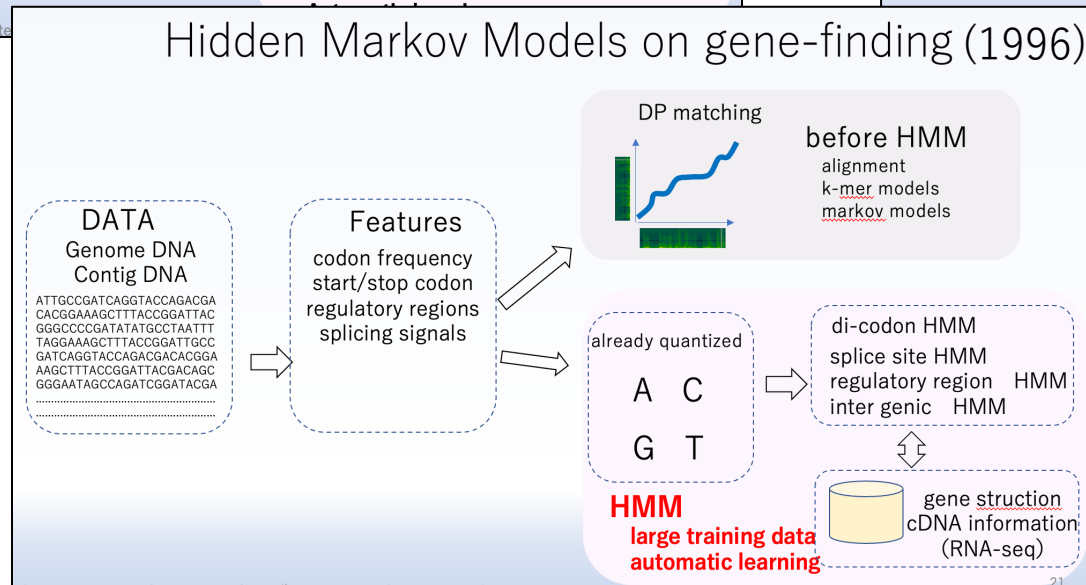
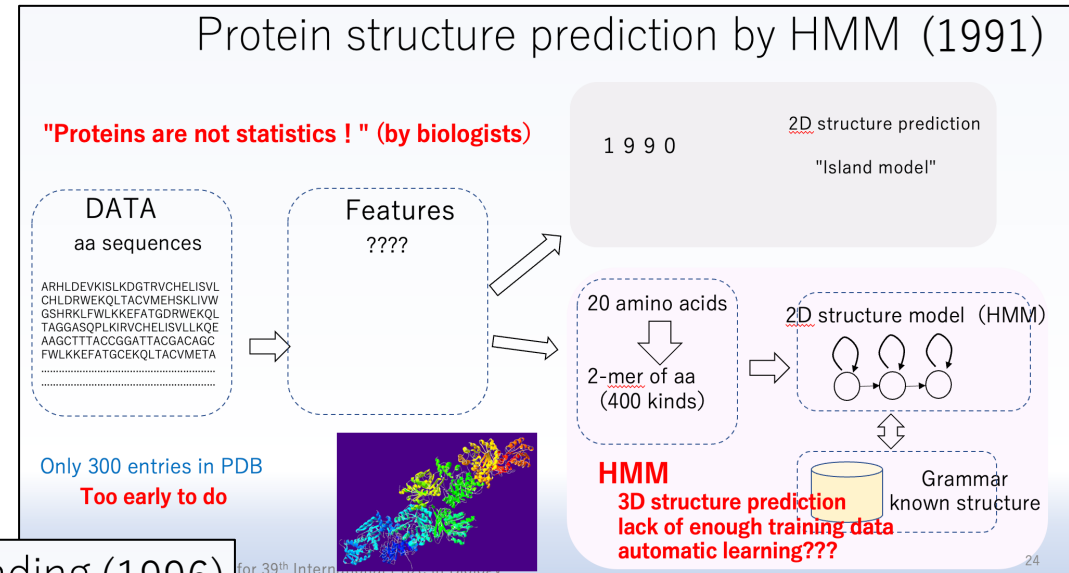
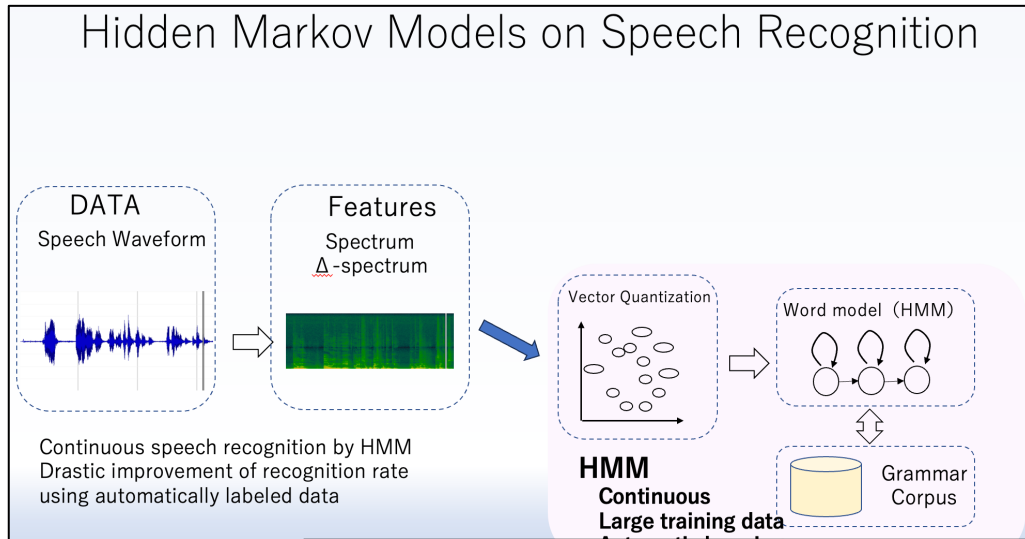


Taneda A and Asai K, Computational and Structural Biotechnology
Journal 13, 1811-1818 (2020)

Goro Terai, Satoshi Kamegai, Akito Taneda, Kiyoshi Asai, Evolutionary design of multiple genes encoding the same protein. Bioinformatics 33(11) 1613-1620

Not only DP

I also loved stochastic LANGUAGE models



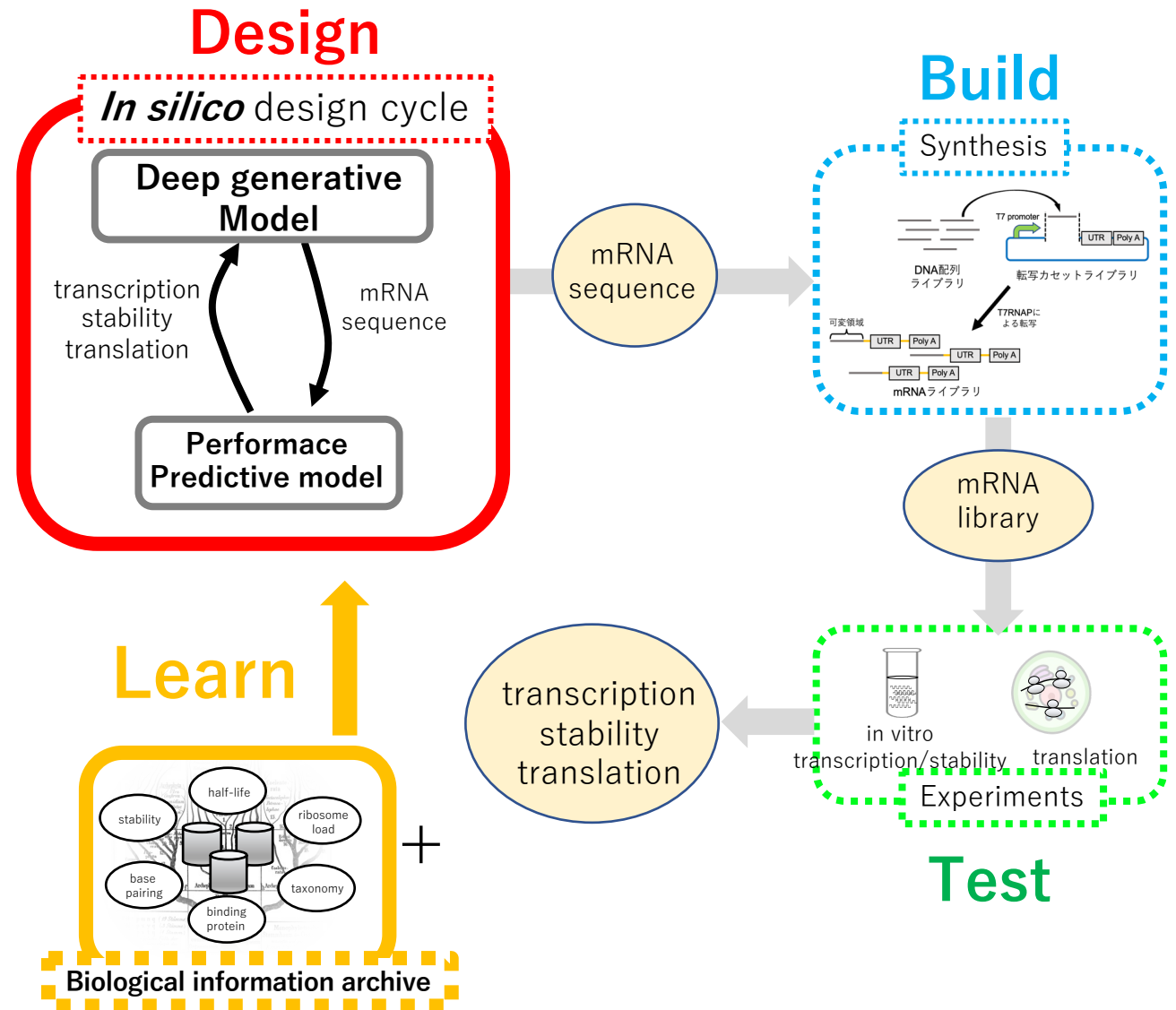
Now I want to use Large Language Models

mRNA design using deep generative models (including LLMs)

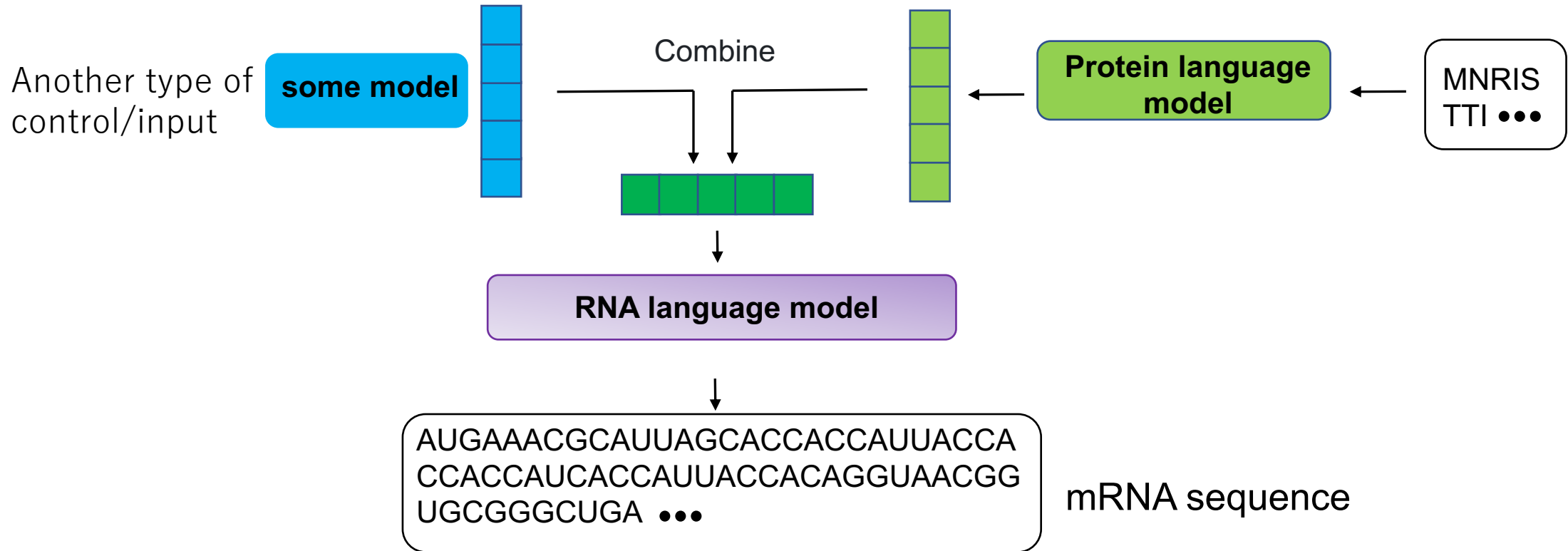
Pre-training by public database
(biological information archive)
Generative & predictive models

Design 1st stage mRNA
of the **target protein**
using *in silico* design cycle

Using the result of Build & Test
Fine-tuning of the models
on the target protein

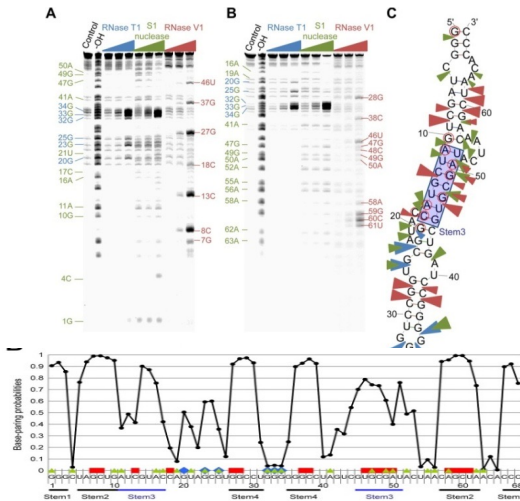


Language models for mRNA design



I also love marginal probabilities

Rnase on probability



1D projection

Base-pairing probability (BPP)

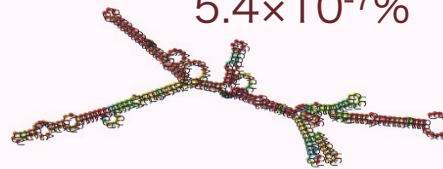
Adachi H et al., *Biochimie*, 93(7):1081-8(2011).

Probability of MFE structure
(the best)

$$P(\sigma | x) = \frac{1}{Z(x)} \exp \frac{-E(\sigma, x)}{kT}$$

Some stable local structures

$5.4 \times 10^{-7}\%$



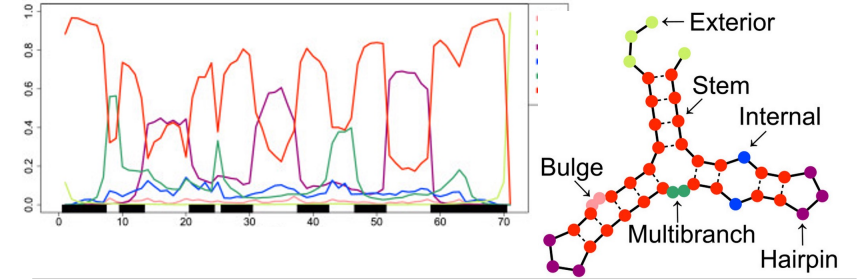
CentroidFold

Maximizes marginalized accuracy measures (MEA)

In CentroidFold, any energy model that can produce BPPs is applicable

Hamada M et al. *Bioinformatics* 25(4)4, 465-473 (2009).

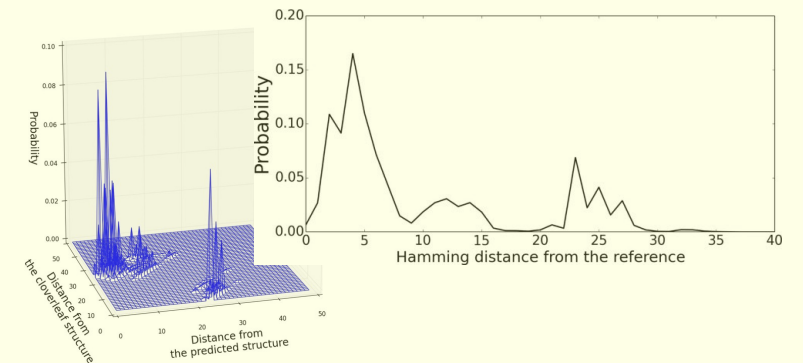
2D structural profile



Probs of local structures are not necessarily small

Fukunaga, T., *Genome Biol* 15, R16 (2014)

Distribution on Hamming distance



Mori, R., *BMC Genomics* 15 (Suppl 10), S6 (2014).

Takizawa, H, *BMC Bioinformatics* 21, 210 (2020).

Marginalized activity by thermodynamic fluctuation

Benasque 2018 model

$$g(x) = E_{\sigma}[f(x, \sigma)] = \sum_{\sigma} p(\sigma|x) f(x, \sigma)$$

x : sequence, σ : structure

$f(x, \sigma)$: activity function of x and σ

(depends to the sequence and the structure)

Linear model for activity function $f(x, \sigma)$

$$f(x, \sigma) = w^T \cdot \phi(x, \sigma)$$

w : weights vector for features

$\phi(x, \sigma)$: feature vector

$$g(x) = E_{\sigma}[w^T \cdot \phi(x, \sigma)]$$

$$= w^T \cdot E_{\sigma}[\phi(x, \sigma)] = w^T \cdot \phi_M(x)$$

$\phi_M(x) = E_{\sigma}[\phi(x, \sigma)]$: marginalized feature

||

Feature vector of **marginalized kernel**

Marginalized Kernel of HMM

$$\log p(x, h|A, E) = \sum_{i=1}^m \sum_{j=1}^m N_{i,j}^a(h) \log a_{i,i} + \sum_{i=1}^m \sum_{d=1}^D N_{i,d}^e(x, h) \log e_{i,d}$$

$$\phi(x, h) = \{N^a, N^e\}$$

$$E_h[N_{i,j}^a] = \frac{1}{p(x|\theta)} \sum_{t=1}^T f_{t-1,i} a_{i,j} e_{j,x_t} b_{t,j}$$

$$E_h[N_{i,d}^e] = \frac{1}{p(x|\theta)} \sum_{t \in \{x_t=d\}} f_{t,i} b_{t,i}$$

Marginalized Kernel of SCFG

$$\log p(x, h|T, E) = \sum_{i=1}^m \sum_{j=1}^m N_{i,j,k}^t(h) \log t_{j,k|i} + \sum_{i=1}^m \sum_{d=1}^D N_{i,d}^e(x, h) \log e_{i,d}$$

Marginalized Kernel of 2D structure distribution

$$P(\sigma|x) = \frac{1}{Z(x)} \exp[-E(x, \sigma)/RT]$$

$$\log P(\sigma|x) = \sum_{\xi \in \{\text{types of loops}\}} N_{\xi}(x, \sigma) E(L_{\xi}^{\text{type}}) - \log Z(x)$$

$$\mathbb{E}[N_{\xi}] = \frac{1}{\partial E(L_{\xi}^{\text{type}})} \partial \log Z(x)$$

HMM: Tsuda K, Kin T, Asai, K, *Bioinformatics* 18 Suppl 1:S268-75 (2002)

RNA-SCFG: Kin T, Tsuda K, Asai K, *Genome Informatics* 13, 112-122 (2002)

Marginalized activity by thermodynamic fluctuation

Benasque 2018 model

$$g(x) = E_{\sigma} [f(x, \sigma)]$$

x : sequ

$f(x, \sigma)$:

(depend

Linear model for

$$f(x, \sigma) = w \cdot \phi(x, \sigma)$$

w : weig

$\phi(x, \sigma)$: feature vector

$$g(x) = E_{\sigma} [w^T \cdot \phi(x, \sigma)]$$

$$= w^T \cdot E_{\sigma} [\phi(x, \sigma)] = w^T \cdot \phi_M(x)$$

$\phi_M(x) = E_{\sigma} [\phi(x, \sigma)]$: marginalized feature

||

Feature vector of **marginalized kernel**

Marginalized Kernel of HMM

Derivative of partition function is very important

For relationship between Transformer and Kernel methods,

Transformers as Support Vector Machines

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, Samet Oymak

<https://arxiv.org/abs/2308.16898> (2023)

$$\sum_{d=1}^m \sum_{h=1}^D N_{i,d}^e(x, h) \log e_{i,d}$$

$$E_h [N_{i,j}^e] = \frac{1}{p(x|\theta)} \sum_{t=1}^T f_{t-1,i} a_{i,j} e_{j,x_t} b_{t,j}$$

$$E_h [N_{i,d}^e] = \frac{1}{p(x|\theta)} \sum_{t \in \{x_t=d\}} f_{t,i} b_{t,i}$$

$$k|i + \sum_{i=1}^m \sum_{d=1}^D N_{i,d}^e(x, h) \log e_{i,d}$$

structure distribution

$$P(\sigma|x) = \frac{1}{Z(x)} \exp[-E(x, \sigma)/RT]$$

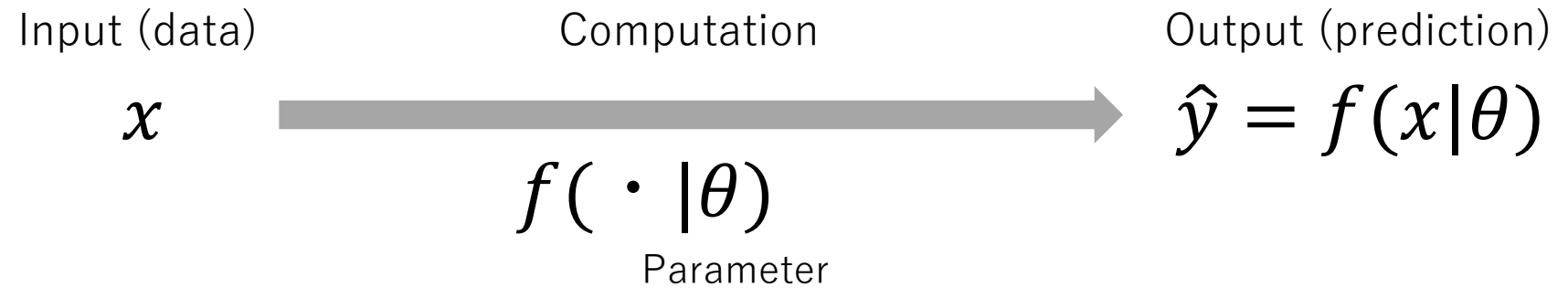
$$\log P(\sigma|x) = \sum_{\xi \in \{\text{types of loops}\}} N_{\xi}(x, \sigma) E(L_{\xi}^{type}) - \log Z(x)$$

$$\mathbb{E}[N_{\xi}] = \frac{1}{\partial E(L_{\xi}^{type})} \partial \log Z(x)$$

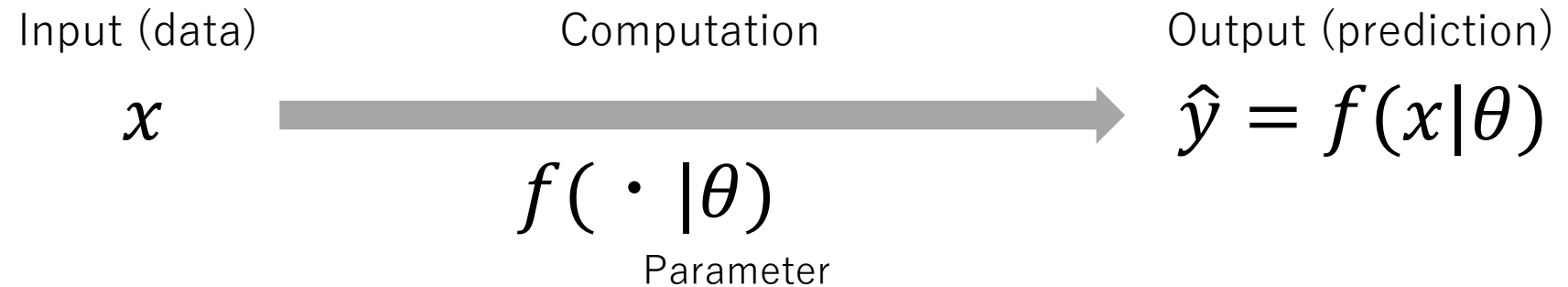
HMM: Tsuda K, Kin T, Asai, K, *Bioinformatics* 18 Suppl 1:S268-75 (2002)

RNA-SCFG: Kin T, Tsuda K, Asai K, *Genome Informatics* 13, 112-122 (2002)

A prediction system with parameters



Optimizing parameters of prediction systems



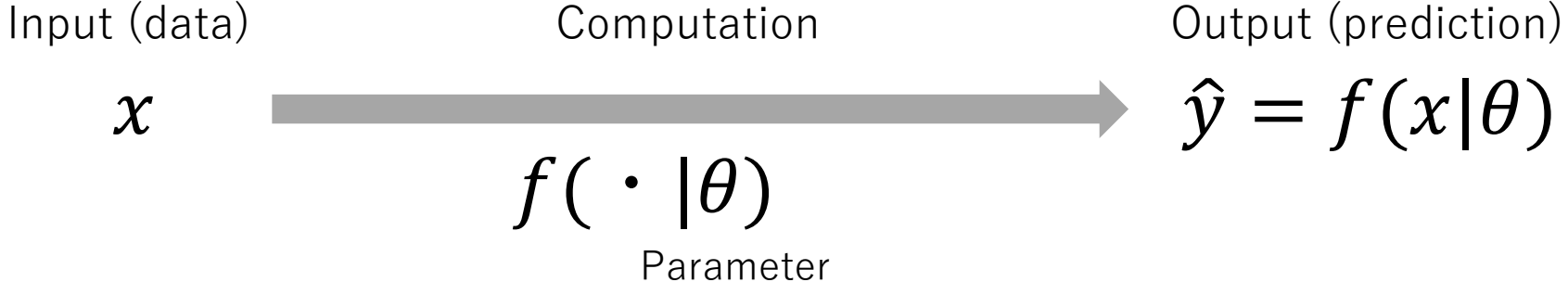
Gradient decent on parameters

Loss function

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}$$

$$\mathcal{L}(y, \hat{y})$$

Optimizing parameters of prediction systems



Gradient decent on parameters

Loss function

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}$$

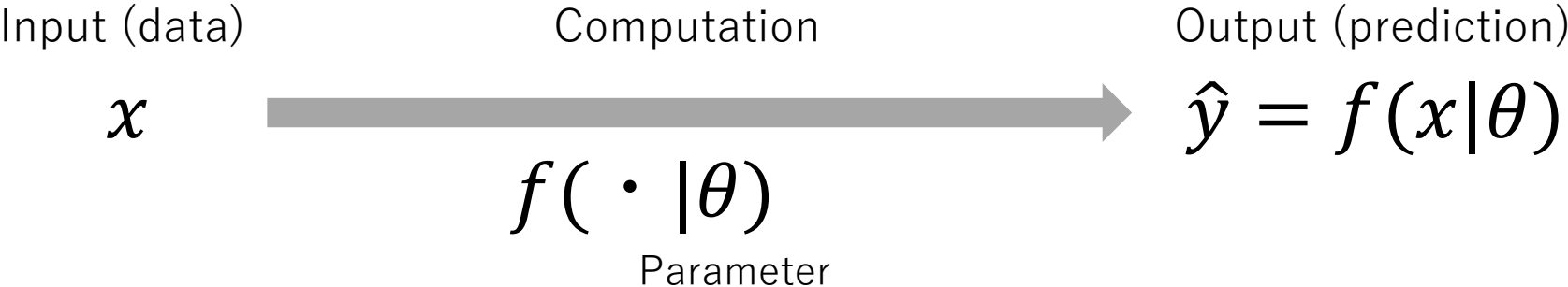
$$\mathcal{L}(y, \hat{y})$$

if $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial f(x, \theta)} \frac{\partial f(x, \theta)}{\partial \theta} = 2\{\hat{y} - y\} \partial_{\theta} f(x, \theta)$$

$$\partial_{\theta} \equiv \frac{\partial}{\partial \theta}$$

Optimizing parameters of prediction systems

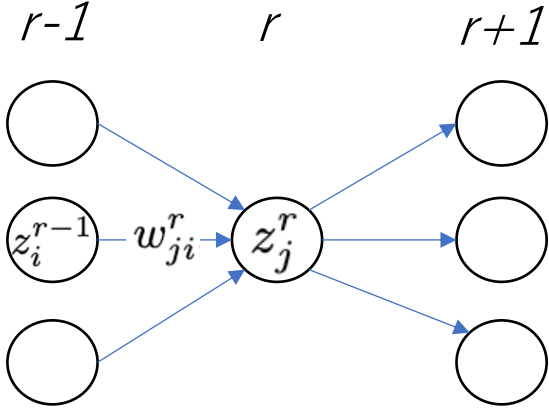


Gradient decent on parameters

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}$$

Artificial neural networks

$$w_{ji}^{(t+1)} \leftarrow w_{ji}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ji}}$$



$$a_j^r = \sum_i w_{ji}^r z_i^{r-1}$$

$$z_j^r = h(a_j^r)$$

Optimizing parameters of prediction systems

Input (data)

x

Computation

$$f(\cdot | \theta)$$

Parameter

Output (prediction)

$$\hat{y} = f(x | \theta)$$

Back-propagation

$$\delta_j^r \equiv \frac{\partial \mathcal{L}}{\partial a_j^r} = \sum_k \frac{\partial z_j^r}{\partial a_j^r} \frac{\partial a_k^{r+1}}{\partial z_j^r} \frac{\partial \mathcal{L}}{\partial a_k^{r+1}}$$

$$= h'(a_j^r) \sum_k w_{kj}^{r+1} \delta_k^{r+1}$$

$$\delta_j^L = \frac{\partial \mathcal{L}}{\partial a_j^L} = \frac{\partial \mathcal{L}(\hat{y} - y)}{\partial \hat{y}}$$

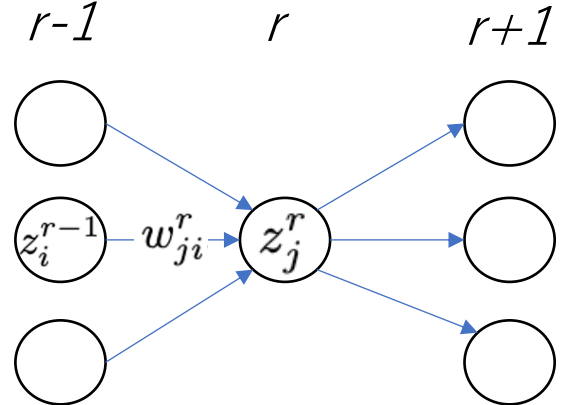
$$\frac{\partial \mathcal{L}}{\partial w_{ji}^r} = \frac{\partial \mathcal{L}}{\partial a_j^r} \frac{\partial a_j^r}{\partial w_{ji}^r} = \frac{\partial \mathcal{L}}{\partial a_j^r} z_i^{r-1} = \delta_j^r z_i^{r-1}$$

Gradient decent on parameters

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}$$

Artificial neural networks

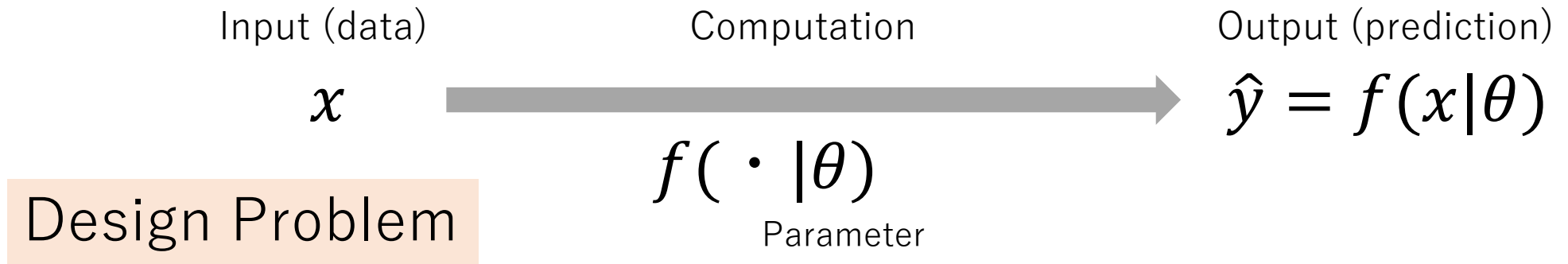
$$w_{ji}^{(t+1)} \leftarrow w_{ji}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ji}}$$



$$a_j^r = \sum_i w_{ji}^r z_i^{r-1}$$

$$z_j^r = h(a_j^r)$$

Optimizing **input** of prediction systems



Gradient decent on parameters

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \theta}$$

Gradient decent on inputs

$$x^{(t+1)} \leftarrow x^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial x}$$

Fixing parameters already optimized

Optimizing **input** of prediction systems

Input (data)

x

Computation

$f(\cdot | \theta)$

Parameter

Output (prediction)

$\hat{y} = f(x | \theta)$

Design Problem

Back-propagation

Gradient decent on inputs

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial a_j^1} \frac{\partial a_j^1}{\partial x_i} = \sum_j \delta_j^1 \frac{\partial a_j^1}{\partial x_i} = \sum_j \delta_j^1 w_{ji}^1$$

$$x^{(t+1)} \leftarrow x^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial x}$$

$$\begin{aligned} \delta_j^r &\equiv \frac{\partial \mathcal{L}}{\partial a_j^r} = \sum_k \frac{\partial z_j^r}{\partial a_j^r} \frac{\partial a_k^{r+1}}{\partial z_j^r} \frac{\partial \mathcal{L}}{\partial a_k^{r+1}} \\ &= h'(a_j^r) \sum_k w_{kj}^{r+1} \delta_k^{r+1} \end{aligned}$$

Optimizing **input** of prediction systems

Input (data)

x

Computation

$f(\cdot | \theta)$

Parameter

Output (prediction)

$\hat{y} = f(x | \theta)$

Design Problem

Back-propagation

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial a_j^1} \frac{\partial a_j^1}{\partial x_i} = \sum_j \delta_j^1 \frac{\partial a_j^1}{\partial x_i} = \sum_j \delta_j^1 w_{ji}^1$$

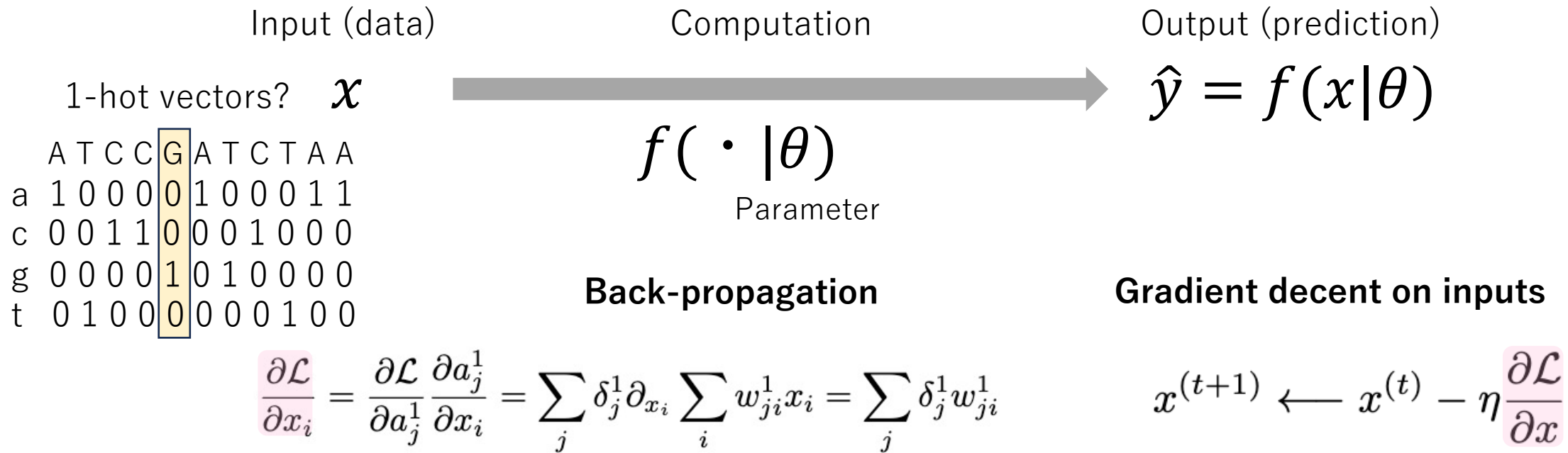
Gradient decent on inputs

$$x^{(t+1)} \leftarrow x^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial x}$$

$$\begin{aligned} \delta_j^r &\equiv \frac{\partial \mathcal{L}}{\partial a_j^r} = \sum_k \frac{\partial z_j^r}{\partial a_j^r} \frac{\partial a_k^{r+1}}{\partial z_j^r} \frac{\partial \mathcal{L}}{\partial a_k^{r+1}} \\ &= h'(a_j^r) \sum_k w_{kj}^{r+1} \delta_k^{r+1} \end{aligned}$$

Inputs are not differentiable
in biological sequence design

Optimizing **input** of prediction systems

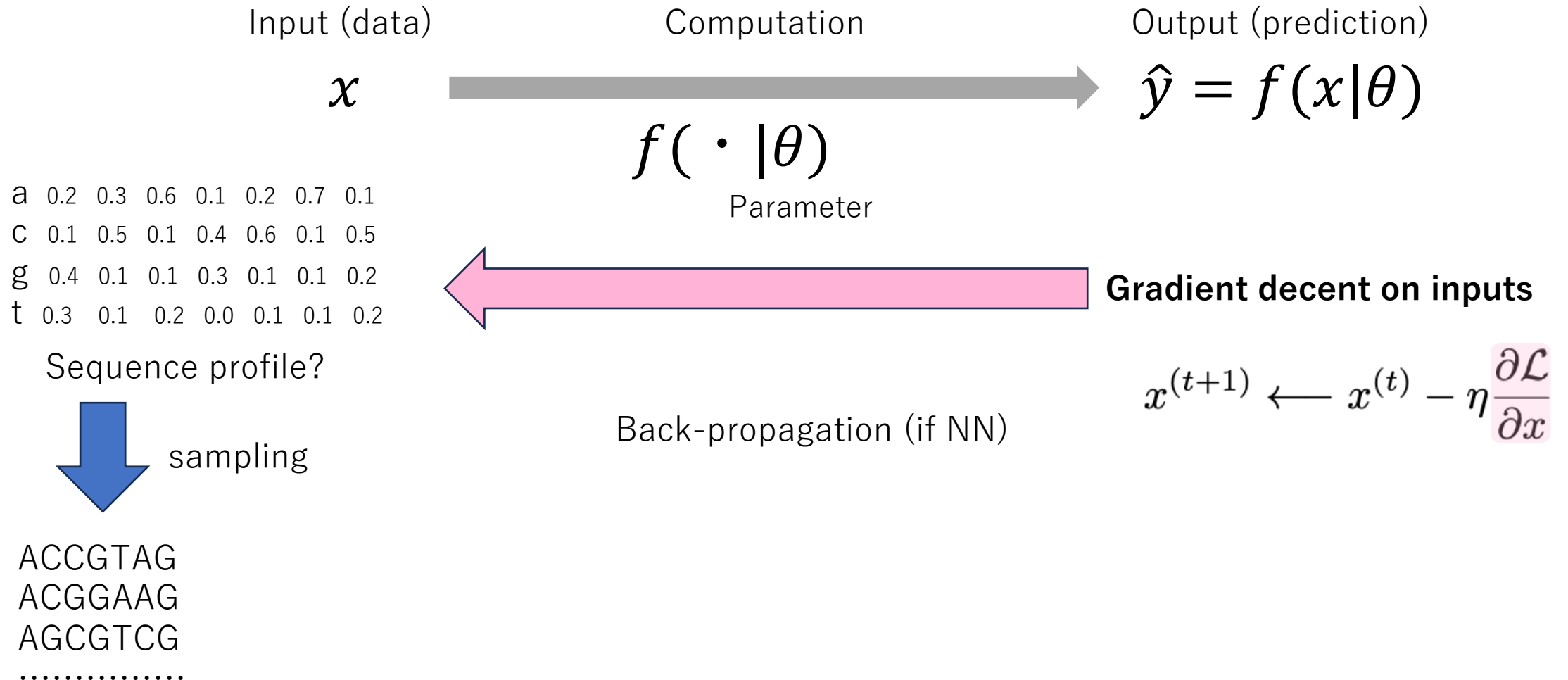


You can treat 1-hot vectors as if real valued vectors

$$\begin{aligned} \delta_j^r &\equiv \frac{\partial \mathcal{L}}{\partial a_j^r} = \sum_k \frac{\partial z_j^r}{\partial a_j^r} \frac{\partial a_k^{r+1}}{\partial z_j^r} \frac{\partial \mathcal{L}}{\partial a_k^{r+1}} \\ &= h'(a_j^r) \sum_k w_{kj}^{r+1} \delta_k^{r+1} \end{aligned}$$

Optimizing **input** of prediction systems

Design Problem



Upstream NN includes information covariation ?

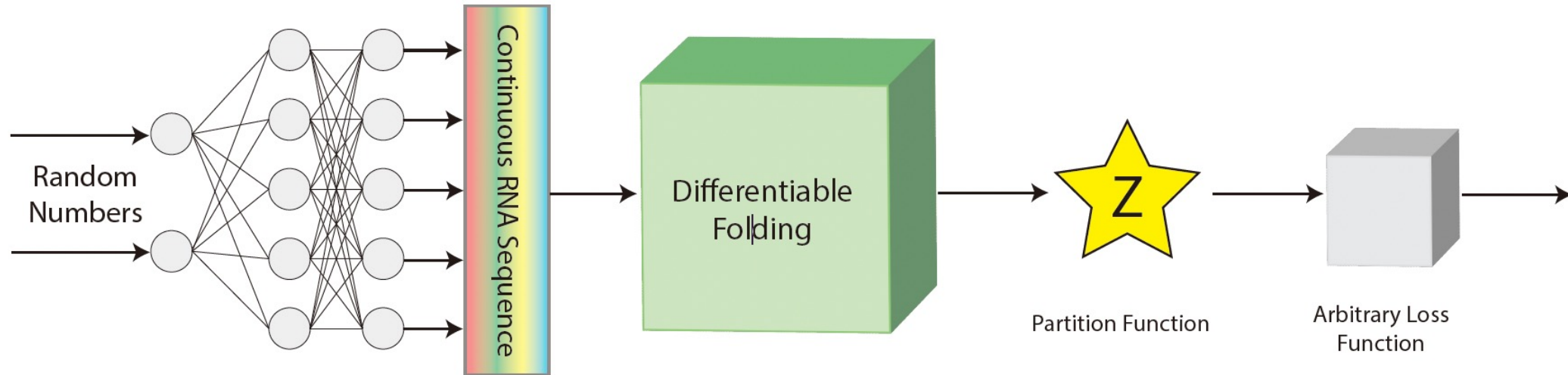
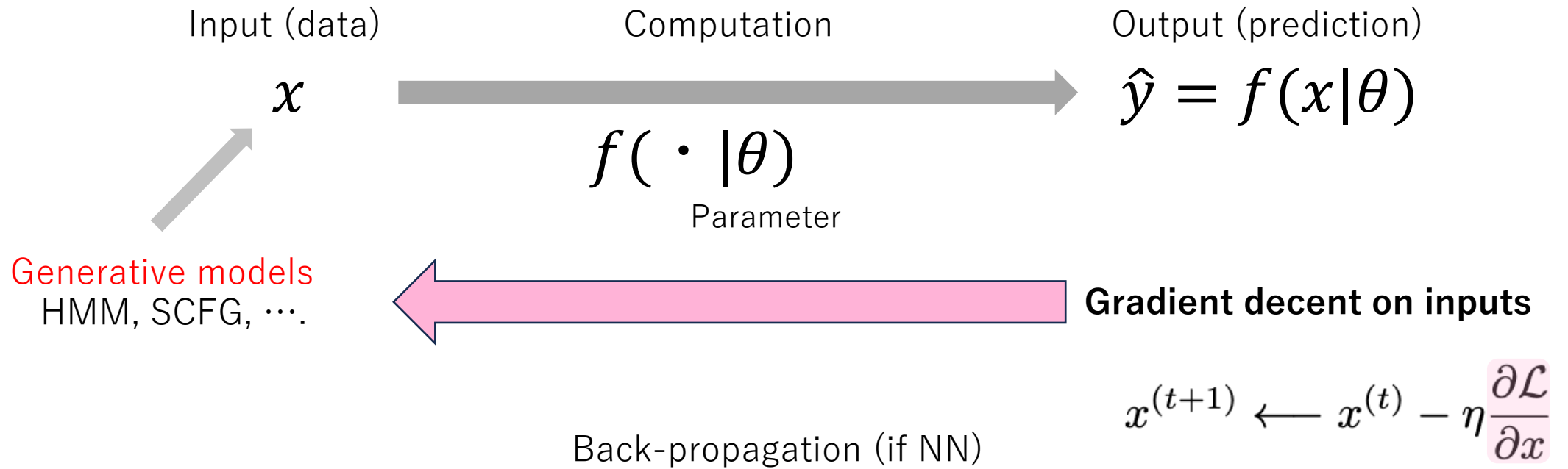


Figure 3: An overview of our method to train a generative model that produces probabilistic sequences that minimize a continuous loss function based on the partition function. We overparameterize the optimization problem by training a generative network to produce a sequence distribution that minimize the objective function. Once trained, discrete sequences can be sampled from the predicted sequence distribution.

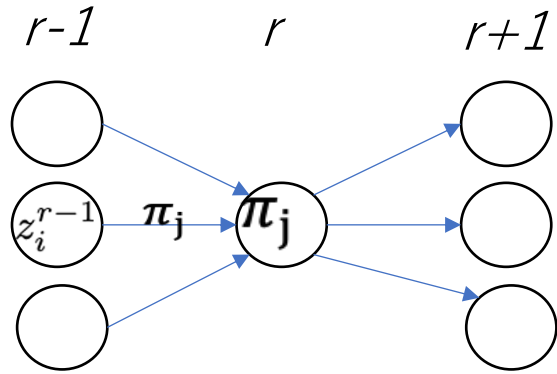
Ryan Krueger and Max Ward
Scalable Differentiable Folding for mRNA Design
<https://www.biorxiv.org/content/10.1101/2024.05.29.594436v1.full.pdf>

Optimizing **input** of prediction systems

Design Problem



Iterations of Neural Network and HMM



Neural Network

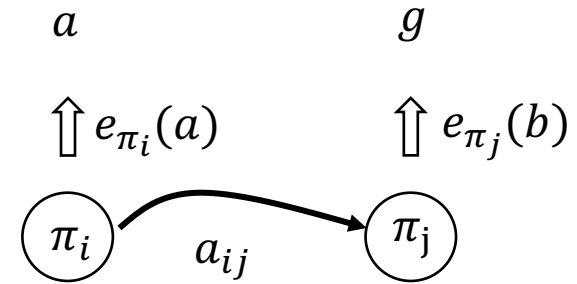
$$a_j^r = \sum_i w_{ji}^r z_i^{r-1}$$

$$z_j^r = h(a_j^r)$$

Same shape of formula,
substituting $h(x)=x$

$$f_j^t = \sum_i f_i^{t-1} w_{ji}^t$$

$$w_{ji}^t \equiv a_{ij} e_j(x_t)$$



HMM

Total probability of the sequence x
calculated by forward algorithm

Initialization:

$$f_0^0 = 1, \quad f_j^0 = 0 (j \neq 0)$$

Iteration:

for $t = 0, \dots, T - 1$

$$f_j^{t+1} = \sum_i f_i^t a_{ij} e_j(x_{t+1})$$

Termination:

$$p(x) = \sum_j f_k^L$$

Iterations of neural network and HMM

$$p(y) = \exp \left\{ \sum_i \theta_i y_i - \psi(\theta) \right\}$$

Canonical form of exponential family

$$p(\pi|x) = \exp \left\{ \sum_{ijt} n_{ij}^t \log w_{ji}^t - \log p(x) \right\}$$

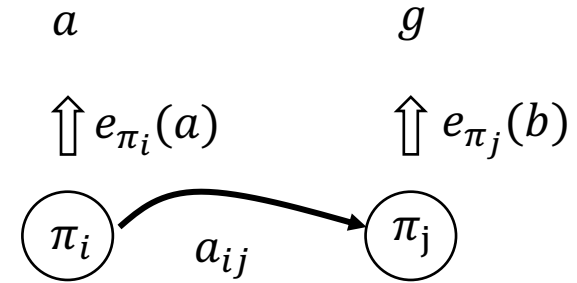
HMM posterior probability (conditional probability)

$$n_{ij}^t = \begin{cases} 1 & \text{if } \pi_{t-1} = i \text{ and } \pi_t = j \\ 0 & \text{otherwise} \end{cases}$$

Using this general property of exponential family,

$$\frac{\partial \psi(\theta)}{\partial \theta_i} = E[y_i]$$

$$\begin{aligned} E[n_{ij}^t] &= \frac{\partial \log p(x)}{\partial \log w_{ji}^t} \\ &= \frac{\partial \log p(x)}{\partial p(x)} \frac{\partial w_{ji}^t}{\partial \log w_{ji}^t} \frac{\partial p(x)}{\partial w_{ji}^t} \\ &= \frac{1}{p(x)} w_{ij}^t \frac{\partial p(x)}{\partial f_j^t} \frac{\partial f_j^t}{\partial w_{ji}^t} \\ &= \frac{1}{p(x)} w_{ij}^t \delta_j^t f_i^t \end{aligned}$$



HMM

$$f_j^{t+1} = \sum_i f_i^t a_{ij} e_j(x_{t+1})$$

$$f_j^t = \sum_i f_i^{t-1} w_{ji}^t \quad w_{ji}^t \equiv a_{ij} e_j(x_t)$$

$$\delta_j^t \equiv \frac{\partial p(x)}{\partial f_j^t} = \sum_k \frac{\partial p(x)}{\partial f_k^{t+1}} \frac{\partial f_k^{t+1}}{\partial f_j^t} = \sum_k \delta_k^{t+1} w_{kj}^{t+1}$$

Backward algorithm of HMM is a backpropagation of forward algorithm

Jason Eisner. [Inside-Outside and Forward-Backward Algorithms Are Just Backprop](#) (tutorial paper). In Proceedings of the Workshop on Structured Prediction for NLP, pages 1–17

EM algorithm

Optimize parameter θ of joint probability

$$p(x, \pi)$$

of observable x and unobservable π

E step: $p(\pi|x, \hat{\theta}^{(t)})$ is calculated

M step $\hat{\theta}^{(t+1)} = \operatorname{argmax}_{\theta} Q(\theta|\hat{\theta}^{(t)})$ is calculated

$$Q(\theta|\hat{\theta}^{(t)}) \equiv E_{\pi|x, \hat{\theta}^{(t)}} [\log p(\pi, x|\theta)]$$

$$p(\pi|x) = \exp \left\{ \sum_{ijt} n_{ij}^t \log w_{ji}^t - \log p(x) \right\}$$

$$E[n_{ij}^t] = \frac{1}{p(x)} w_{ij}^t \delta_j^t f_i^t$$

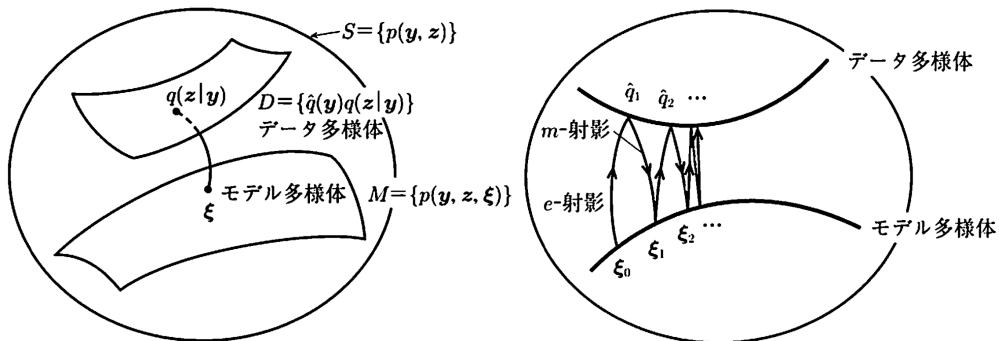


図 12.2 モデル多様体とデータ多様体.

図 12.3 EM アルゴリズム.

甘利俊一 情報幾何の新展開、数理科学別冊 (2014)

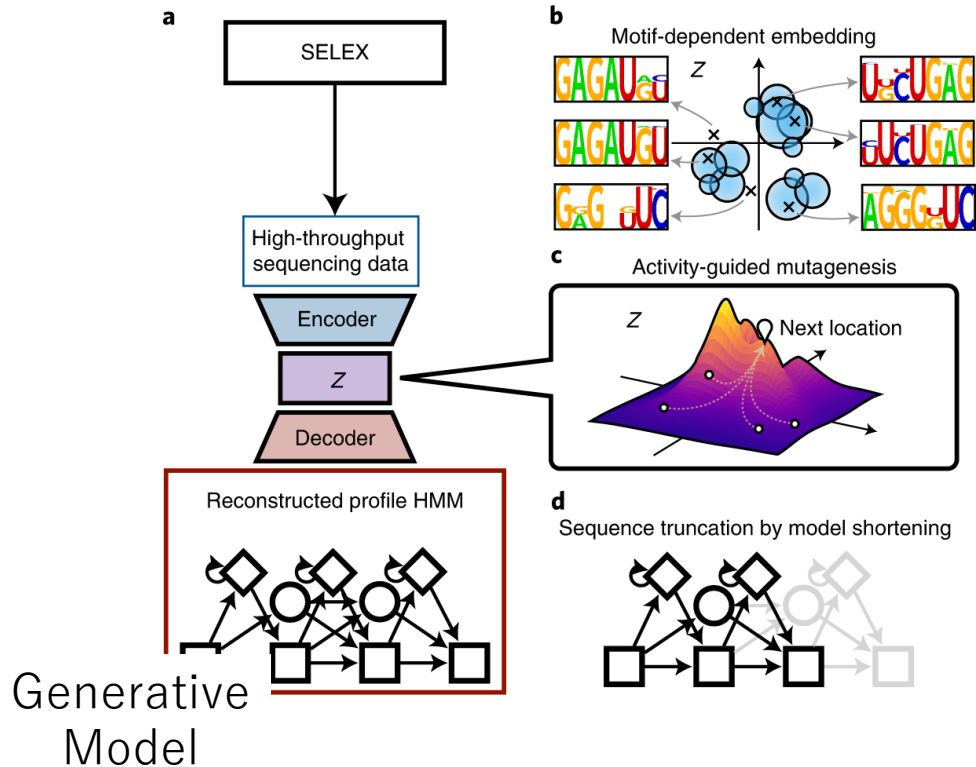
How about gradient descent?

$$\frac{\partial p(x)}{\partial w_{ji}^t} = \frac{\partial p(x)}{\partial f_j^t} \frac{\partial f_j^t}{\partial w_{ji}^t} = \delta_k^j f_i^t$$

No, but you can backpropagate at least.

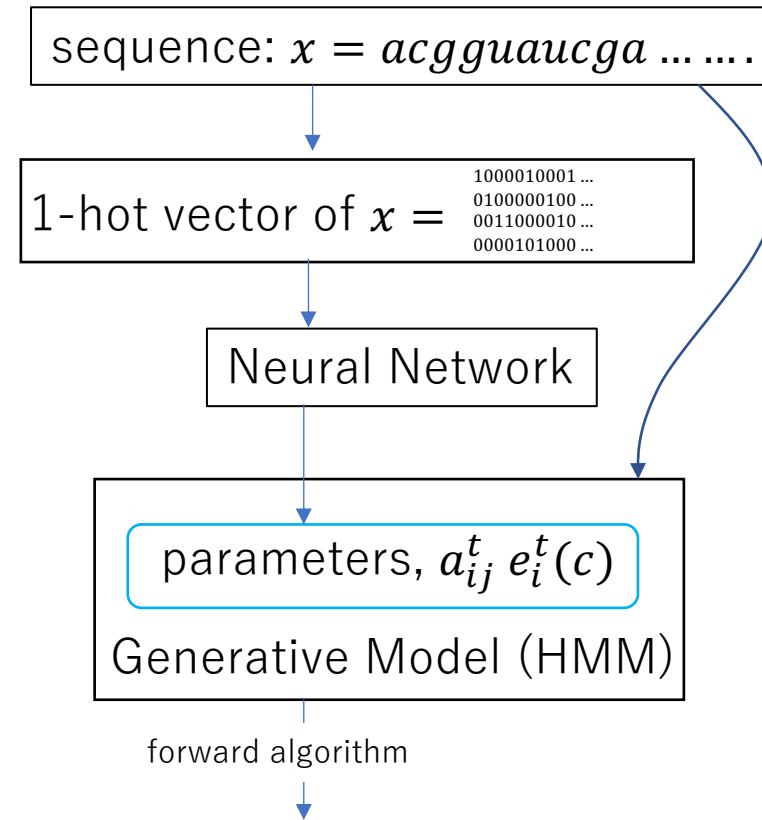
Combining generative model to NN

RaptGen for RNA aptamer



Iwano, N., Adachi, T., Aoki, K. *et al.* Generative aptamer discovery using RaptGen. *Nat Comput Sci* **2**, 378–386 (2022).

Connecting the output of NN to parameters of HMM you can further backpropagate on NN.



$$\frac{\partial \mathcal{L}}{\partial w_{ji}^r} = \delta_j^r z_i^{r-1}$$

$$\delta_j^r = h'(a_j^r) \sum_k w_{kj}^{r+1} \delta_k^{r+1}$$

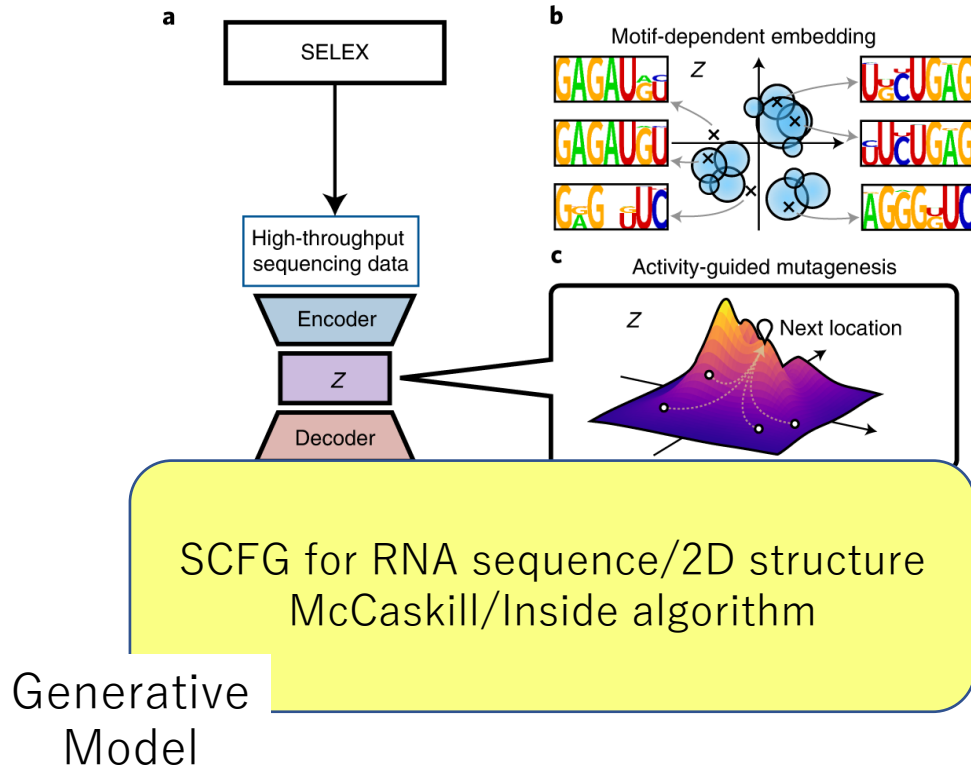
$$\frac{\partial p(x)}{\partial w_{ji}^t} = \frac{\partial p(x)}{\partial f_j^t} \frac{\partial f_j^t}{\partial w_{ji}^t} = \delta_k^j f_i^t$$

$$\delta_j^t \equiv \frac{\partial p(x)}{\partial f_j^t} = \sum_k \delta_k^{t+1} w_{kj}^{t+1}$$

backpropagation

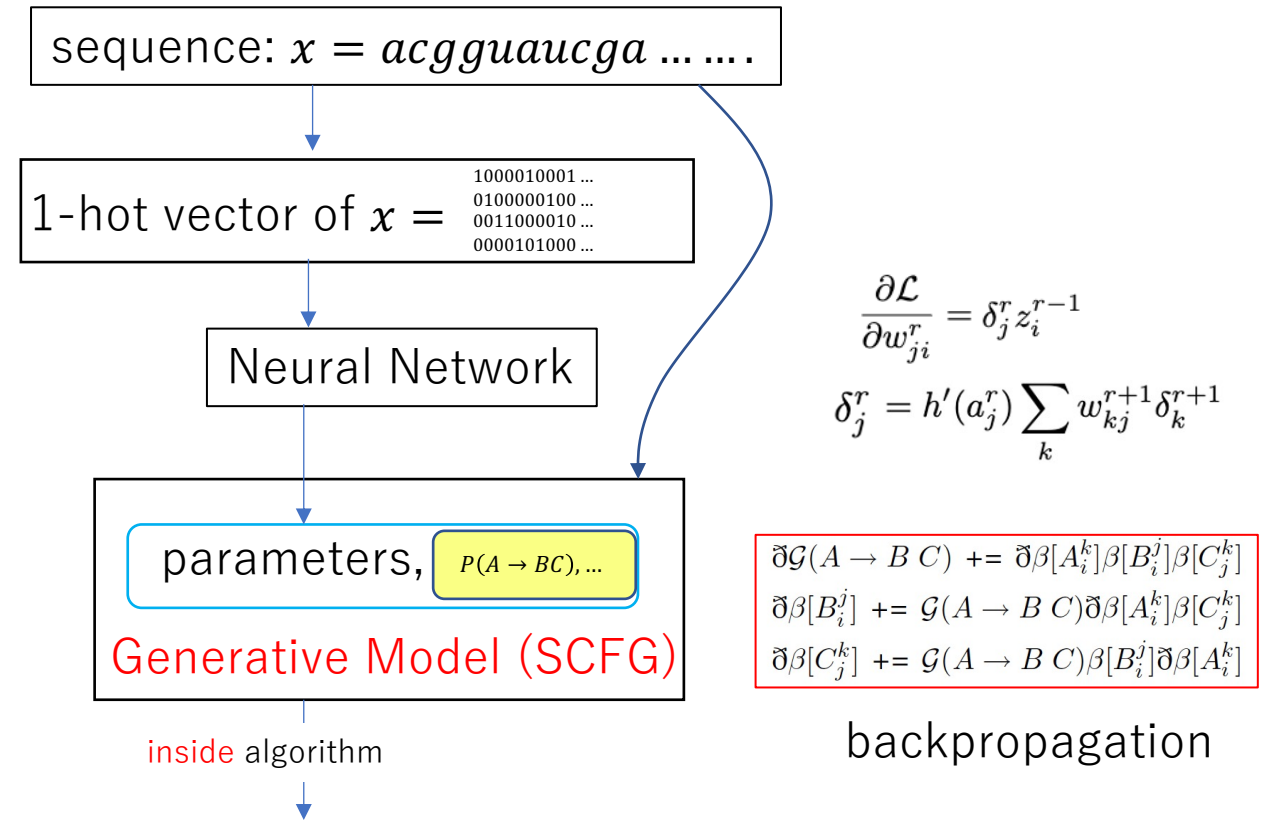
Combining generative model to NN

RaptGen for RNA aptamer

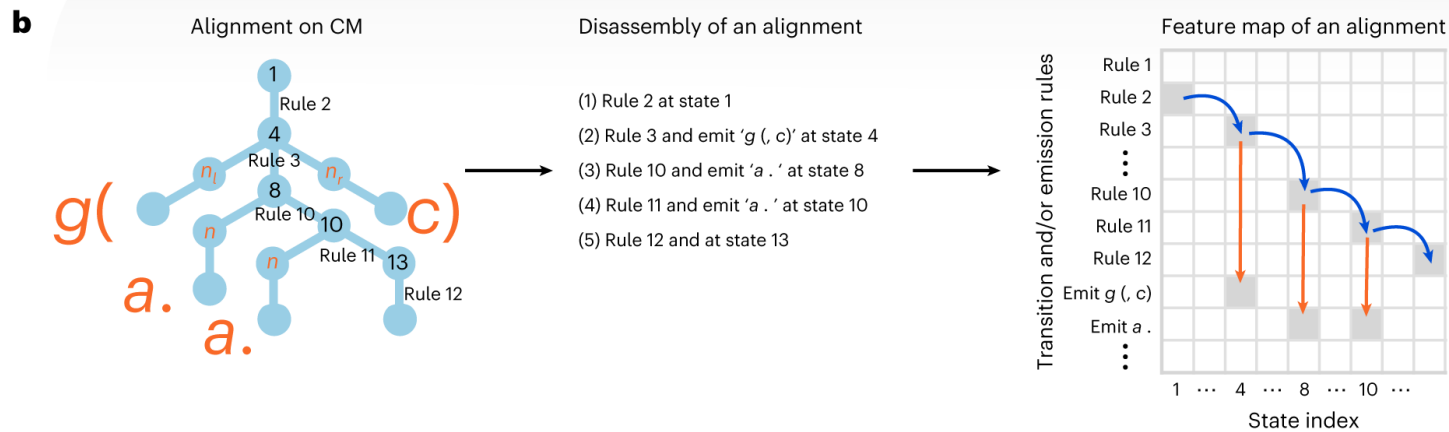
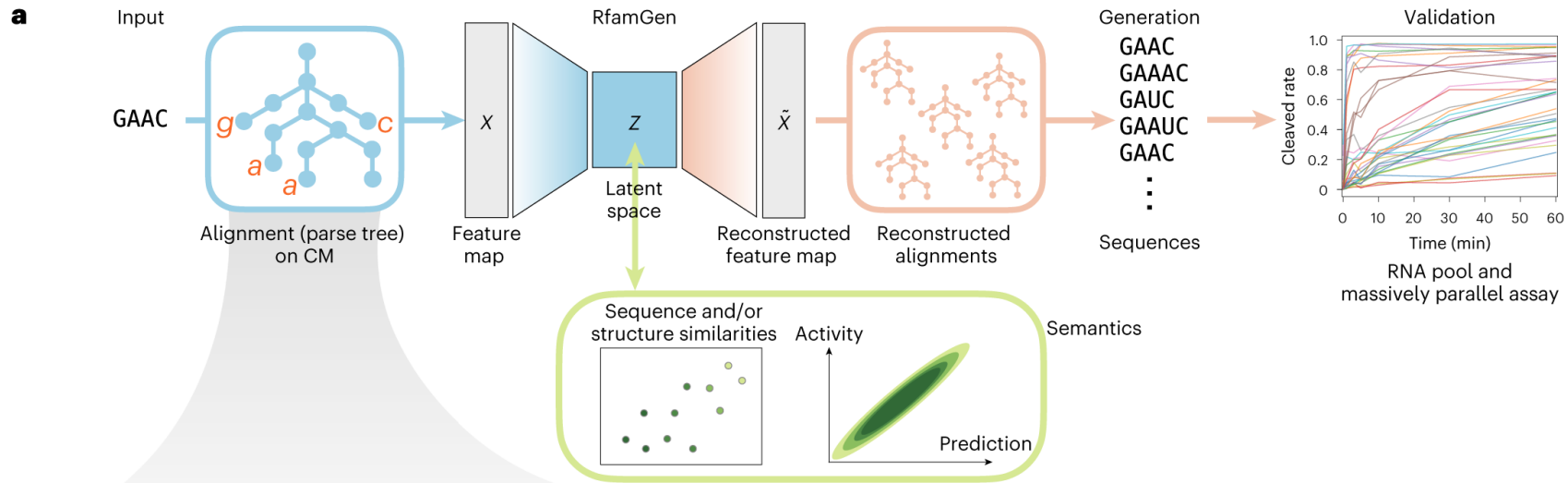


Iwano, N., Adachi, T., Aoki, K. *et al.* Generative aptamer discovery using RaptGen. *Nat Comput Sci* **2**, 378–386 (2022).

Connecting the output of NN to parameters of HMM you can further backpropagate on NN.



RfamGen



Sumi, S., Hamada, M. & Saito, H. Deep generative design of RNA family sequences. Nat Methods 21, 435–443 (2024)

Inverse Folding, BPP as the target

1. 2D structure as the target

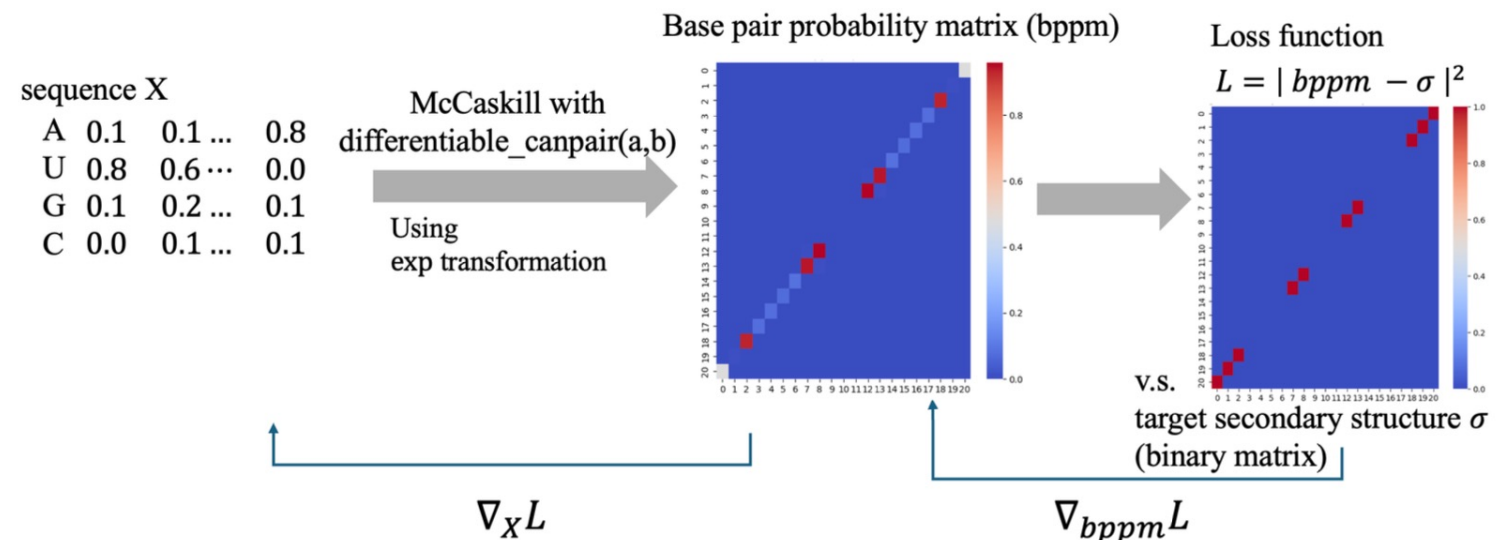
Derivative of max operation is necessary (both in MFE and MEA).

Sum of the probabilities of similar structures are ignored.

2. BPP as the target

McCaskill algorithm is naturally differentiable (by the loss or parameters)

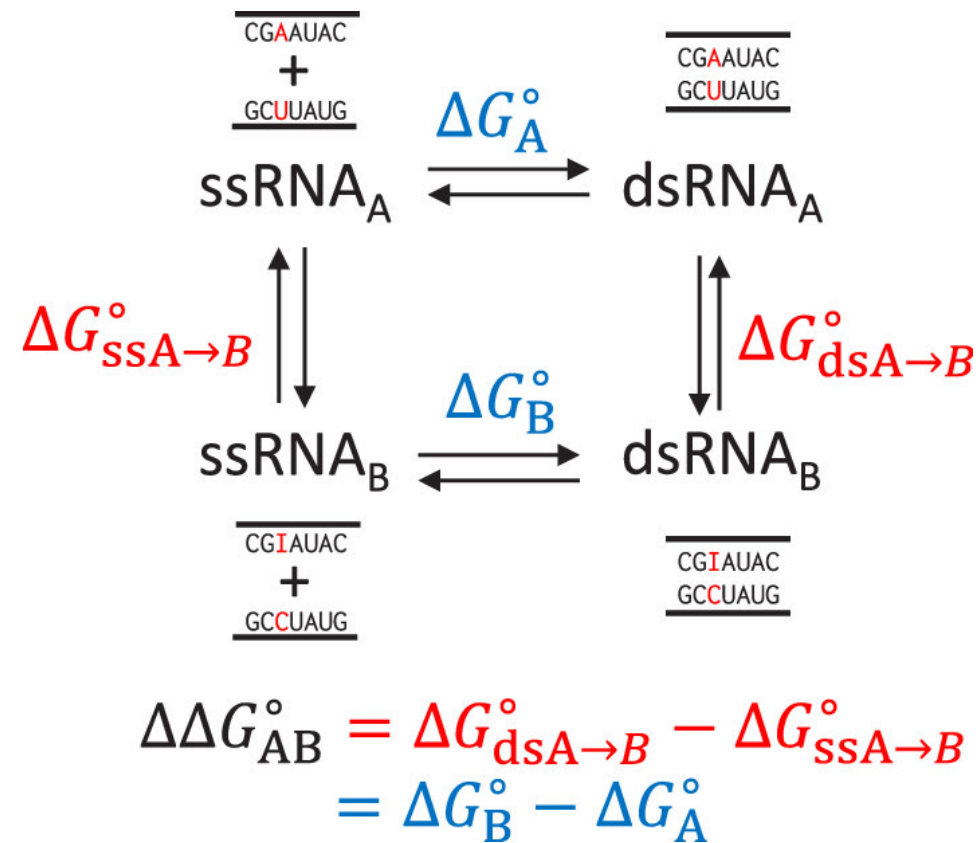
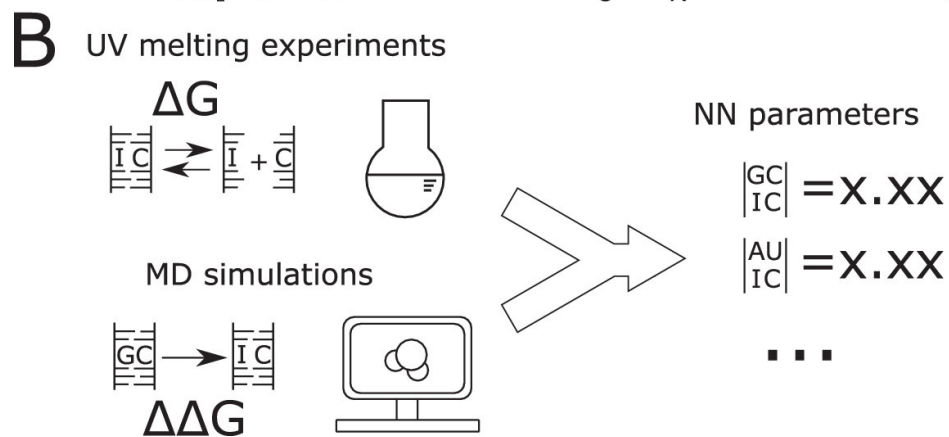
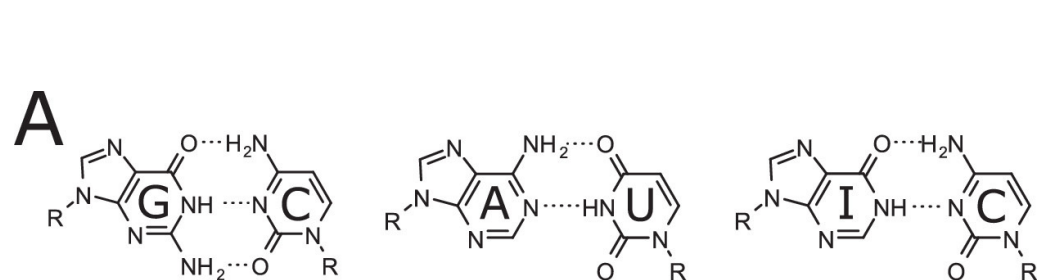
However, **differentiation of McCaskill by the sequence, is not easy**



Design of RNA with modified bases

- Modified RNA may have preferable features.
- Design of RNA sequence with modified bases
 - 2D structure control in design requires de novo prediction although there are methods for 2D prediction using SHAPE data.
 - Determination of energy parameters of modified bases is desired.
- For determination of RNA energy parameters including modified bases
 - Energy parameters of Inosine and m6A have been determined by combination of absorbance measure measurement of small number of pairs of complementary sequences and molecular dynamics (MD) calculation

Free-Energy Calculation of Ribonucleic Inosines and Its Application to Nearest-Neighbor Parameters



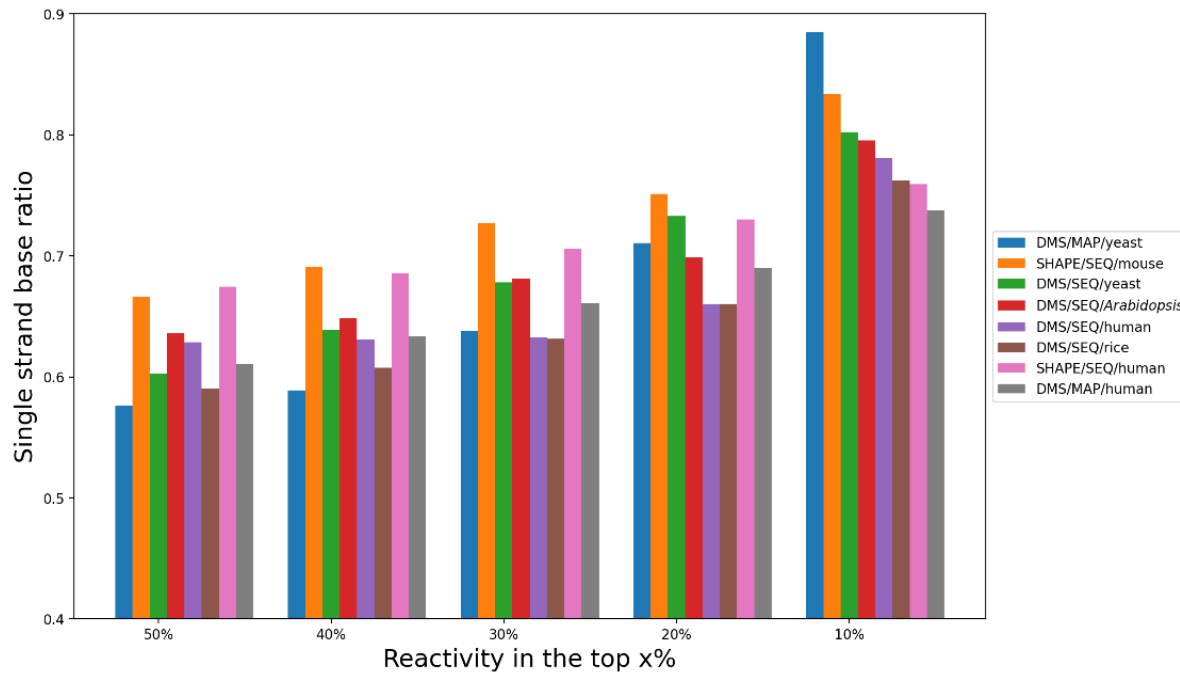
Design of RNA with modified bases

- Modified RNA may have preferable features.
- Design of RNA sequence with modified bases
 - 2D structure control in design requires de novo prediction although there are methods for 2D prediction using SHAPE data.
 - Determination of energy parameters of modified bases is desired.
- For determination of RNA energy parameters including modified bases
 - Energy parameters of Inosine and m6A have been determined by combination of absorbance measure measurement of small number of pairs of complementary sequences and molecular dynamics calculation
 - **Gradient decent by differentiation of BPPs by energy parameters**

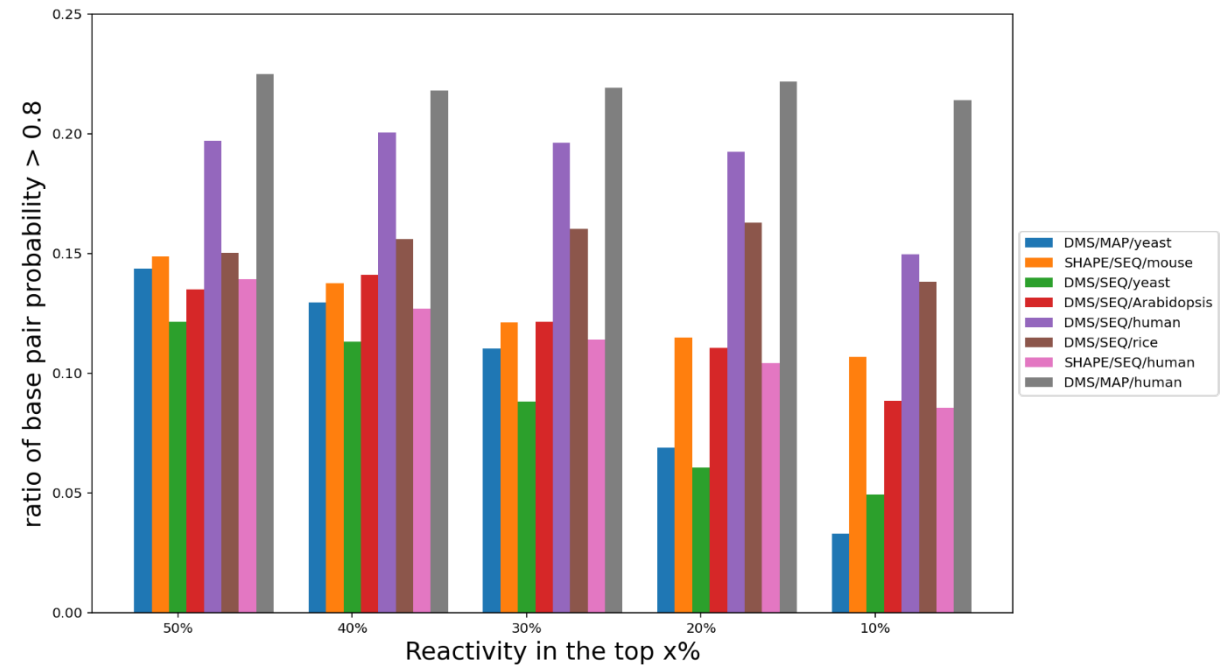
Chemical probing data and base-pairs

High reactivities correspond to free bases well,
but low reactivities do not necessarily mean base-paired.
Pseudo-free energy fits well to this observation.

$$\Delta G_{\text{total}} = \Delta G_{\text{thermodynamic}} + \Delta G_{\text{SHAPE}}$$



Reactivity vs annotation



Reactivity vs BPP

Upstream NN includes information covariation ?

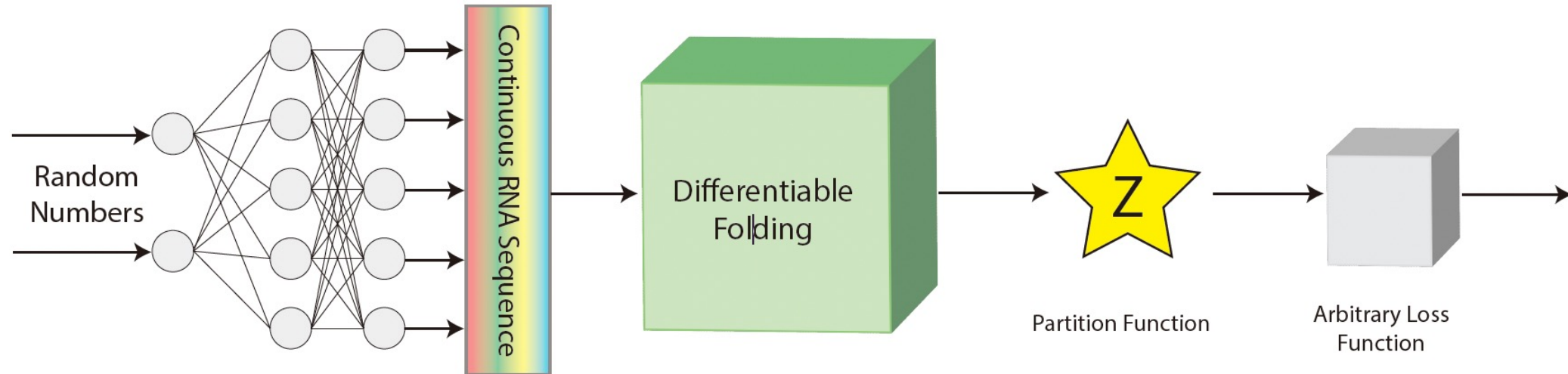


Figure 3: An overview of our method to train a generative model that produces probabilistic sequences that minimize a continuous loss function based on the partition function. We overparameterize the optimization problem by training a generative network to produce a sequence distribution that minimize the objective function. Once trained, discrete sequences can be sampled from the predicted sequence distribution.

Ryan Krueger and Max Ward
Scalable Differentiable Folding for mRNA Design
<https://www.biorxiv.org/content/10.1101/2024.05.29.594436v1.full.pdf>

Concluding Remarks

- Derivative by parameters may give a gradient descent optimization
 - Energy parameters of modified bases may be determined using SHAPE data.
 - Derivative by inputs may solve the input design problem.
- In RNA sequence design, optimizing discrete sequences require tricks.
 - VAE, LLM, sequence profile, HMM/SCFG(CM)
 - **Differentiable partition function is a great help for further research.**
- Modeling position dependency in probability distribution is one of the problems to be solved in applications of differential partition function.
- Theoretical note
 - Forward-backward/Inside-outside correspond to backpropagation in NN.
 - Attention in Transformer is not ad-hoc innovation, but related to SVM.

Acknowledgements

RNA-dojo members

Kengo Sato

Michiaki Hamada

Hisanori Kiryu

Yutaka Saito

Goro Terai

Junichi Iwakiri

Shun Sakuraba

Akito Taneda

Tsukasa Fukunaga

Takumi Otagaki

Kazuteru Yamamura

IPknot, MXfold2, CASP

RaptGen, RfamGen

Rfold, Raccess

LLM for mRNA

miRRim, QRNAstruct)

CASP

MD for energy parameters

COSMO

CapR

← folding, PK visualization, , CASP

SHAPE data analysis , CASP

Experiments

Hiroshi Abe

Tetsuro Hirose

Mikiko Shiomi

Akihiko Kondo

and many others

