

# 分子コンピューティングにおける Reservoir Computer の実装とリザーバー性能の自動探索

理学専攻・情報科学コース 伊藤美賀 (指導教員: Natanael Aubert-Kato)

## 1 はじめに

本研究では分子コンピューティングのためのモジュラーフレームワークである, PEN DNA toolbox [1] を Reservoir Computing (RC) に応用する. リカレント特性を持っていれば物理的なりザーバーとして置き換えることが可能であることから, 化学反応ネットワーク (CRN) を使用した RC を実装した. CRN はネットワークサイズが大きくなれば表現力が上がる一方で, 物理的な再現が難しくなる. リザーバー特性とネットワークサイズのトレードオフの関係を解明するために, 品質多様性アルゴリズム [2] を用いてリザーバー性能空間の自動探索を行う. また, 品質多様性アルゴリズムを適用したシミュレーションの計算コストが高いという課題に対して, サロゲートモデルを作成することで同等の性能を保ったまま計算時間を削減する.

## 2 実験手法

### 2.1 Reservoir Computing

RC は, 入力層, 中間層, 出力層の3層で構成されており, 出力層でのみ学習が行われる. (図1左)

具体的には, 入力信号  $u(n) \in \mathbb{R}^I$  に対して, 目標出力は  $y^{target}(n) \in \mathbb{R}^O$  である.  $I$  と  $O$  はそれぞれ入力と出力の数であり,  $N$  はリザーバーのノード数である. 典型的な更新式は以下の通りである:

$$x(n) = f(W^{in} \cdot u(n) + W \cdot x(n-1)), \quad (1)$$

ここで  $x(n) \in \mathbb{R}^N$  はリザーバー内のニューロン活性化ベクトルであり  $W^{in} \in \mathbb{R}^{N \times I}$  と  $W \in \mathbb{R}^{N \times N}$  は入力重み行列と結合重み行列を表し  $f$  は活性化関数である.

出力層は以下のように定義される:

$$y(n) = W^{out} \cdot x(n)^T \quad (2)$$

ここで  $y(n) \in \mathbb{R}^O$  はネットワーク出力であり  $W^{out} \in \mathbb{R}^{O \times N}$  は出力重み行列を表している.

### 2.2 CRN の Reservoir Computing への適用

PEN DNA Toolbox で実装可能な CRN をリザーバーとして使用した (図1右). PEN DNA Toolbox では, モジュール間の相互作用として機能する. 本研究では, 分子流入の一つをシステム全体の時系列入力として変化させ, 流入によって直接影響を受けるシグナル種, インヒビター, テンプレートの濃度をリザーバー自身の出力として用いる. そして, これらの出力に出力重み行列  $W_{out}$  を掛けて, システム全体の出力  $y(n)$  を求める (式2). 図2は, ネットワークからの出力が目標関数に追従しようとしていることを示している.

### 2.3 Reservoir Computing の性能指標

最適化を行う上でリザーバーが達成すべき性能指標として, 以下の4つを目的関数として採用した [3].

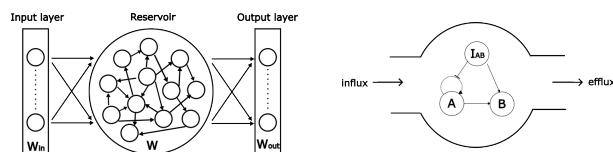


図1: 従来型 Reservoir Computing のモデル (左) と CRN を使用した Reservoir Computing のモデル (右)

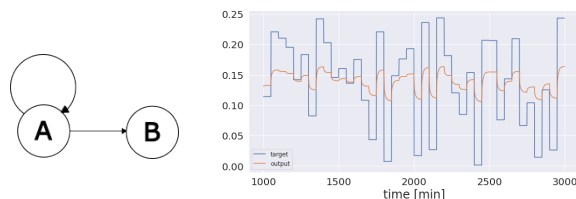


図2: シンプルな CRN をリザーバーとして用いたときの目的関数 (青) と出力 (オレンジ) の値

**記憶容量 (MC)** 記憶容量とは入力情報をどれくらいの期間記憶しておくかということである. 入力信号  $u(n)$  に対して,  $u(n-k)$  を記憶するタスクで学習された出力信号  $y_k(n)$  と,  $k$  によって遅延された入力信号  $u(n)$  の間の二乗相関係数で計算される.

**カーネル特性 (KR)** カーネル特性とは, リザーバーが異なる入力パターンを区別する能力の尺度である. カーネル特性を測るには, 分散の大きいデータを入力信号として用い, リザーバー内の状態  $x(n)$  を特異値分解した値を計算する. この値が大きいほうが高いカーネル特性を持つ.

**汎化能力 (GR)** 汎化能力とは, 類似した入力情報が与えられた場合の一般化能力のことである. 汎化能力が劣っていると, 訓練データに対しては学習がされているが, 未知データに対しては適合できていないという状況に陥ってしまう. 汎化能力を測るには, ノイズの大きいデータを入力信号として用い, リザーバー内の状態  $x(n)$  を特異値分解した値を計算する. この値が小さいほうが高い汎化能力を持つ.

**CRN サイズ** CRN サイズとはネットワークのノード数, すなわちシグナル種の数である.

CRN サイズが小さいものほど物理的な実験と計算機上のシミュレーション結果での差異が小さくなるため, 大きな記憶容量, 高いカーネル特性, 高い汎用能力を持つ最小のリザーバーを得たい.

### 2.4 最適化

品質多様性アルゴリズムの一つである MAP-Elites [4] を用いて, CRN を最適化する. メモリ容量をフィットネスとして用い, カーネルランクと汎化ランクを特徴量とみなしている. この方法は, 高性能な CRN を探索し, そのトポロジーとシグナル種の安定性をパラ

メータ値として変化させることを意図している。異なるサイズのトポロジーを同時に探索することでトポロジーの複雑さとのトレードオフを考慮する。

## 2.5 サロゲートモデルの実装

品質多様性アルゴリズムを使った最適化によるシミュレーションは計算コストの面で安価とは言えず、多くのシミュレーションを必要とする解析は膨大な計算コストのために現実的ではない。そこで、シミュレーションの出力を正確に近似するサロゲートモデルを構築する。サロゲートモデルの実装には、ニューラルネットワークを使用する。入力としての CRN と、対応する出力としてのメモリ容量を学習データセットに組み込み、これらを回帰問題として扱うことで得られたデータセットを基に機械学習モデルを構築することが可能になる。今回は問題の単純化のためネットワークサイズは5に限定し、インヒビターは考慮しない。

## 3 実験結果

記憶容量とサイズの関係に注目すると、ネットワークのサイズが小さいとき記憶容量が緩やかに上昇し、プラトーになった後、下降していることがわかる。これはネットワークが大きくなるにつれて、適合させるべきパラメータの数が増えることに起因していると考えられる。ネットワークサイズが小さいものの良い性能を持つ個体を発見することができたことがわかる。

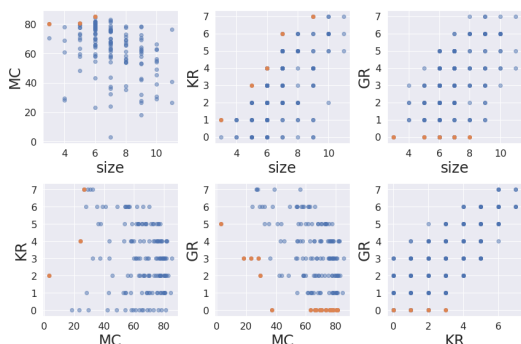


図 3: 最適化プロセスによって得られた最終個体。オレンジの個体は Pareto Front を表している。

また、サロゲートモデルの有用性を示すために、構築したサロゲートモデルがどの程度正確に記憶容量を推定できるかを調べた。完全なモデルとサロゲートモデルで得られた記憶容量を比較すると、両者にはある程度の相関があることがわかった (相関係数=0.42166385, 図 4 右)。次の検証では、最初の 2000 回の評価をサロゲートモデルを用いて最適化し、残りの 8000 回の評価を完全なモデルで最適化する。このとき、サロゲートモデルによる最適化で得られた最終個体は、完全なモデルでの初期個体として再評価される。

評価回数に対して  $QDscore$  (性能と多様性の両方を評価した値) と記憶容量がどのように変化していくかを示したのが図 4 左である。評価回数が 10000 回の完全なシミュレーションに対し、評価回数 8000 回のシミュレーションのメモリ容量はほぼ同じレベルである。サロゲートモデルによる記憶容量推定は非常に高速であるため、サロゲートモデルを用いることでシミュレ

表 1: 最適化プロセスの計算時間と、最終個体分布の比較のための Mann-Whitney U 検定による p 値。

計算時間		Mann-Whitney U
full	srg + full	p 値
487m48.938s	407m23.597s	0.28582

シヨンの評価回数を減らすことができた場合、約 17% の計算時間短縮に寄与する (表 1)。さらに、完全なシミュレーションと組み合わせて得られた最終個体について Mann-Whitney U 検定を行った結果、得られた個体は同等であることが分かった (表 1)。

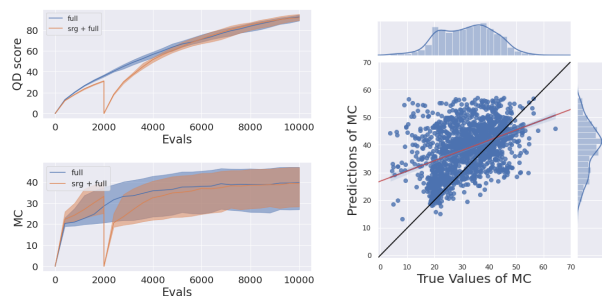


図 4: 左:  $QDscore$  とメモリ容量の進化。中央の太線は中央値、透明部分は Q1, Q3 を表す。右: 完全モデル ( $x$  軸) とサロゲートモデル ( $y$  軸) のメモリ容量の値の比較。線形回帰を赤で、 $x = y$  を黒で示す。

## 4 まとめと今後の課題

4つの性能指標に焦点を置き、品質多様性アルゴリズムを用いて RC の性能向上に向けて実験を行った。小さいサイズで高い性能を持つ多様なリザバーを生成することができ、目的やタスクに合わせて適切なりザバーを選択することができるだろう。またサロゲートモデルの適用により、完全なモデルに匹敵する性能を保ちつつ計算時間を削減することができた。今後はサロゲートモデル自体の性能の向上、記憶容量以外の性能指標におけるサロゲートモデルを構築などに着手したい。

## 参考文献

- [1] K. Montagne, R. Plasson, Y. Sakai, T. Fujii, and Y. Rondelez. Programming an in vitro dna oscillator using a molecular networking strategy. *Molecular systems biology*, Vol. 7, No. 1, p. 466, 2011.
- [2] L. Cazenille. Qdpy: A python framework for quality-diversity. <https://gitlab.com/leo.cazenille/qdpy>, 2018.
- [3] M. Dale, J.F. Miller, S. Stepney, and M.A. Treftzer. A substrate-independent framework to characterise reservoir computers. arxiv 2018. *arXiv preprint arXiv:1810.07135*.
- [4] J.B. Mouret and J. Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.