

「分子プログラミング」 分子コンピューティングの展開

萩谷 昌己

東京大学

大学院情報理工学系研究科

コンピュータ科学専攻

生体分子コンピューティングとは

- 生体分子 = 情報処理装置
 - 化学反応の自律的制御 自身にコード化
 - 微小・省エネルギー
 - 超並列
 - 分子の物理化学的性質
- 分子コンピューティングの目標
 - 分子反応の持つ潜在的計算能力の理学的解明
 - 分子反応に基づく新しい計算機能の工学的実現
- 参考文献
 - 萩谷, 横森編: DNAコンピュータ, 培風館, 2001
 - 萩谷編著: 分子コンピュータの現状と展望 – 分子プログラミングへの展開, サイエンス社, 2004

分子コンピューティングの目標

- 分子反応の持つ計算能力の解析と
- その応用
 - 分子による計算を活用した分子計測技術
バイオテクノロジーへの応用
 - プログラムされた自己組織化や分子マシン
ナノテクノロジーへの応用
 - 分子による進化的計算
分子進化学への応用
- 分子反応に基づく新しい計算パラダイム

分子コンピューティングの展開

- 分子反応に基づく計算モデルの提案と実装
 - 計算能力の解析
 - 計算可能性
 - 計算量 --- 時間、領域
 - エラーや生成量 --- 確率的な解析
 - 実装技術
 - 応用
- 計算モデルを実現する分子システムの設計論
 - ad hocなtrial&errorからシステムティックな設計論へ
特に計算論的な側面 **分子プログラミング**
 - 分子設計 = 配列設計
 - 反応設計

様々な計算モデルとその解析

- Adleman-Lipton
 - DNAの選択的ハイブリダイゼーションを利用した解の候補のランダムな生成
 - データ並列計算による解の抽出
 - Suyama --- dynamic programming
 - Sakamoto-Hagiya --- SAT Engine
 - Head-Yamamura --- aqueous computing
- Seeman-Winfrey
 - 各種形態のDNA分子の自己組織化 (self-assembly)
 - 自己組織化による計算

様々な計算モデルとその解析

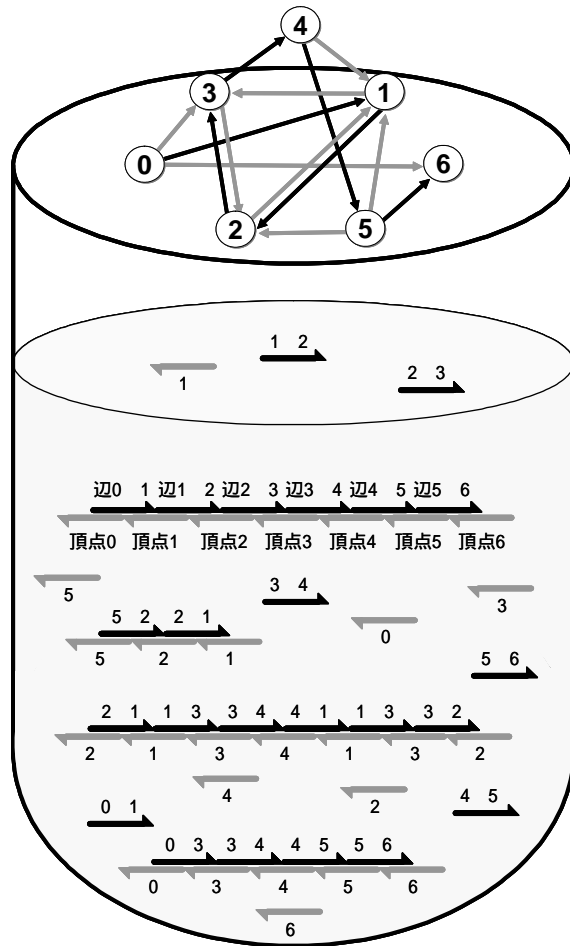
- Head
 - 遺伝子の組み換えによる言語の生成
 - Ogihara-Ray
 - ブール回路の並列計算
 - Hagiya-Sakamoto
 - 状態機械 (Whiplash)
 - Shapiro
 - 有限オートマトン
- } 分子マシンのところで

Adleman-Liptonパラダイム

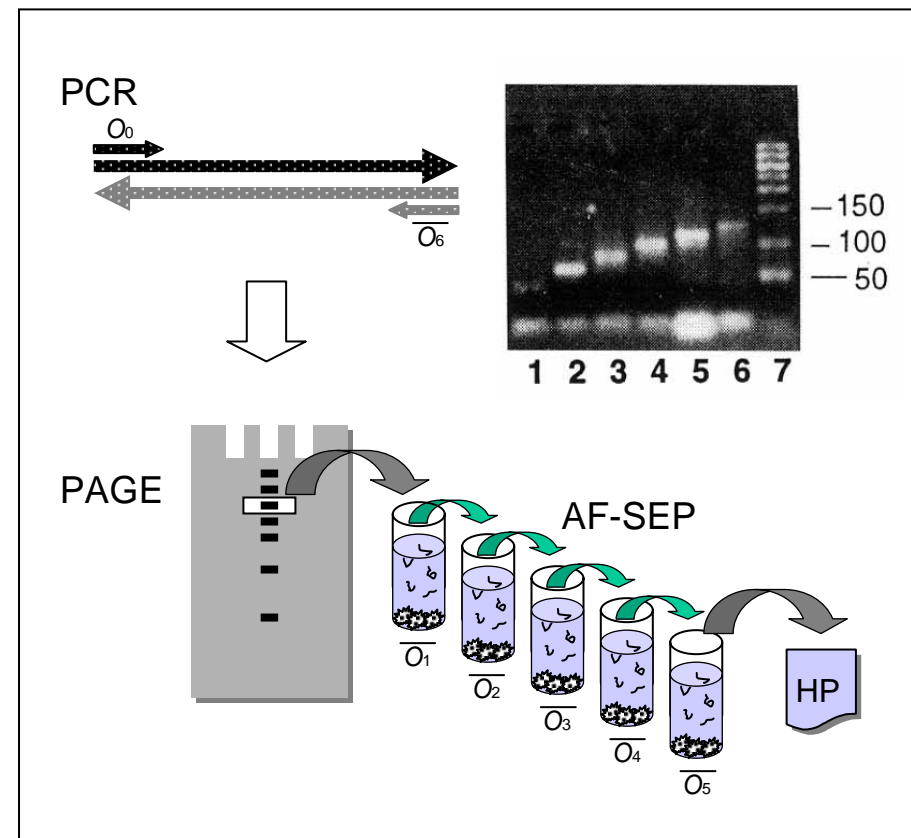
- Adleman (*Science* 1994)
 - ハミルトン経路問題をDNAを用いて解く。
- Lipton, *et al.*
 - SAT問題をDNAを用いて解く。
- 分子による超並列計算
 - 主として組み合わせ的最適化
 - DNAの自己会合によるランダムな生成
 - 解の候補 = DNA分子
 - 生物学実験技術を駆使した解の抽出

Adlemanの最初のDNAコンピュータ

Generation of a pool of
all possible solutions



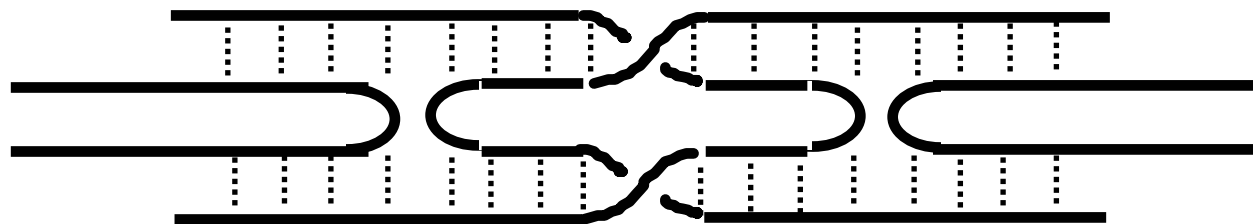
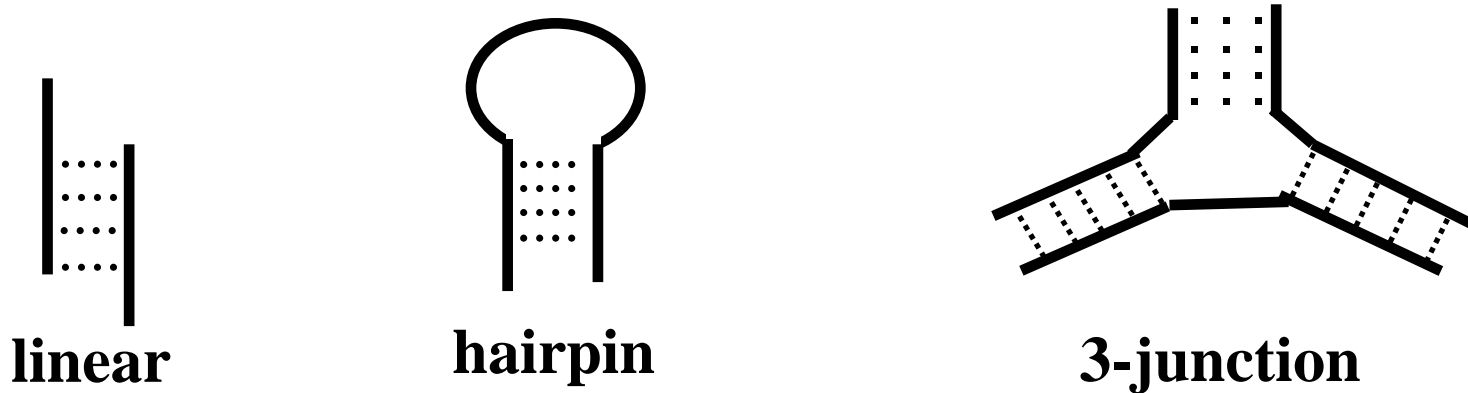
Selection of
correct answers



Adleman-Liptonパラダイムに関する 現在のコンセンサス

- 電子コンピュータを凌駕するには程遠い。
 - スケールアップ問題
- 「分子が計算する」ことの
proof of conceptとしては重要。
- 少なくとも、バイオテクノロジーの
ベンチマークとして使うことができる。
- さらに、遺伝子計測への応用 (Suyama)。

Seeman-Winfreeの DNAの自己組織化による計算

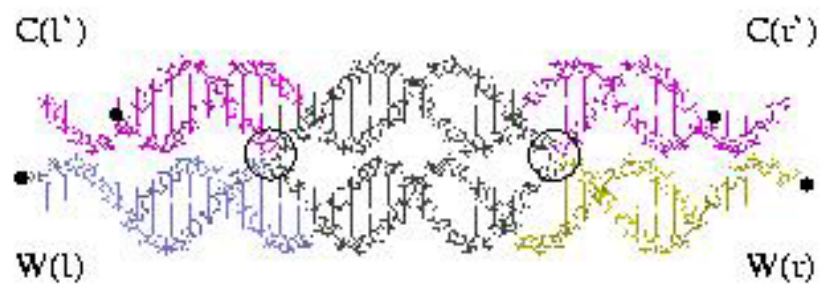
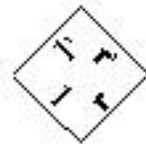


DX (double crossover)

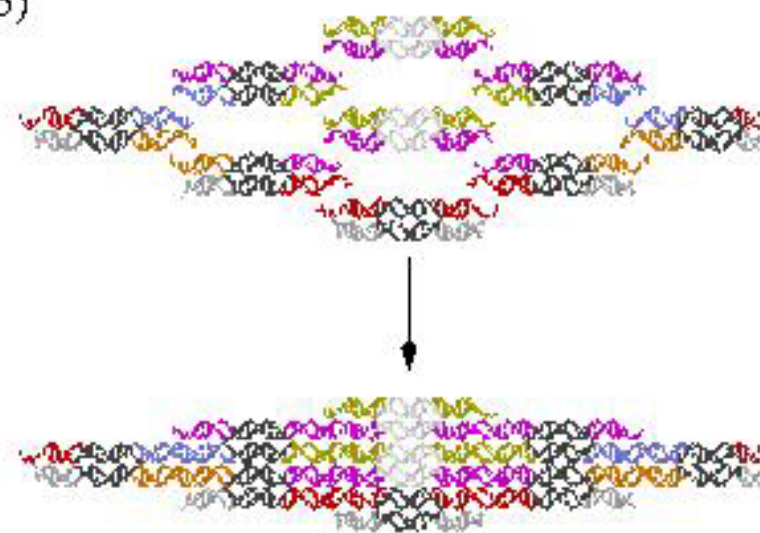
様々なDNA構造分子

The DNA representation of Wang tiles.

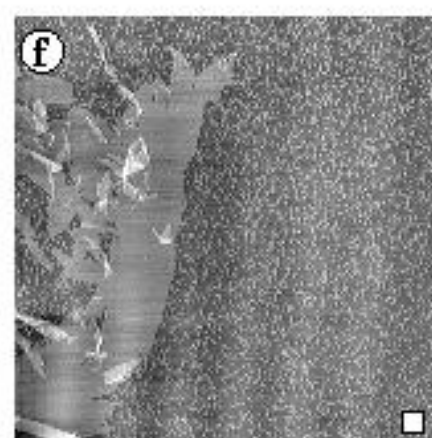
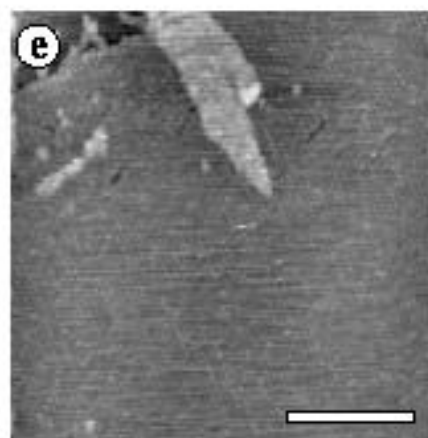
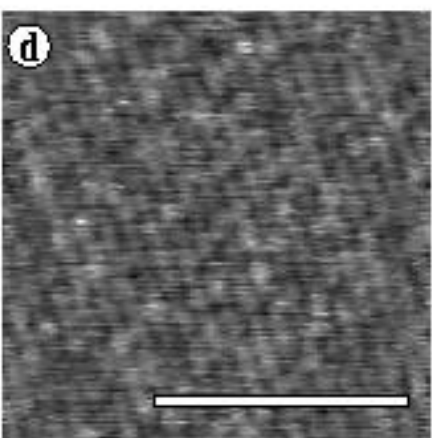
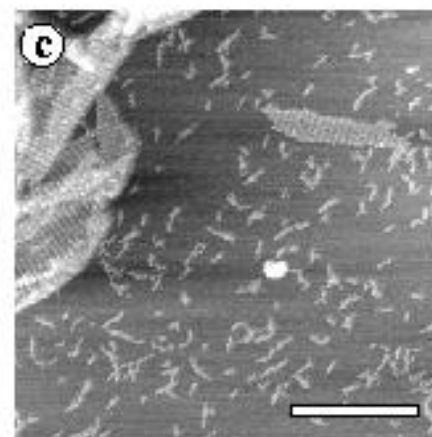
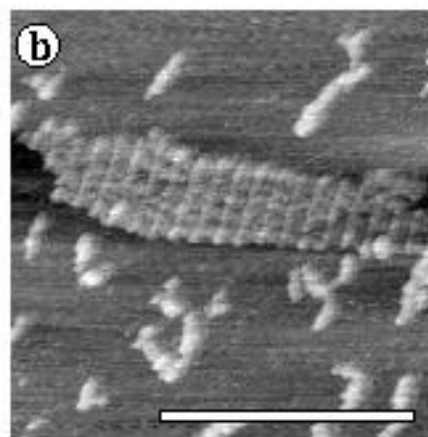
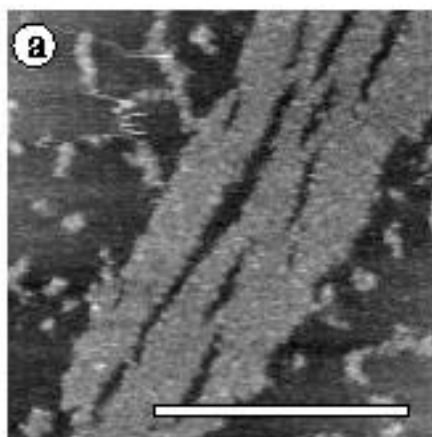
(a)



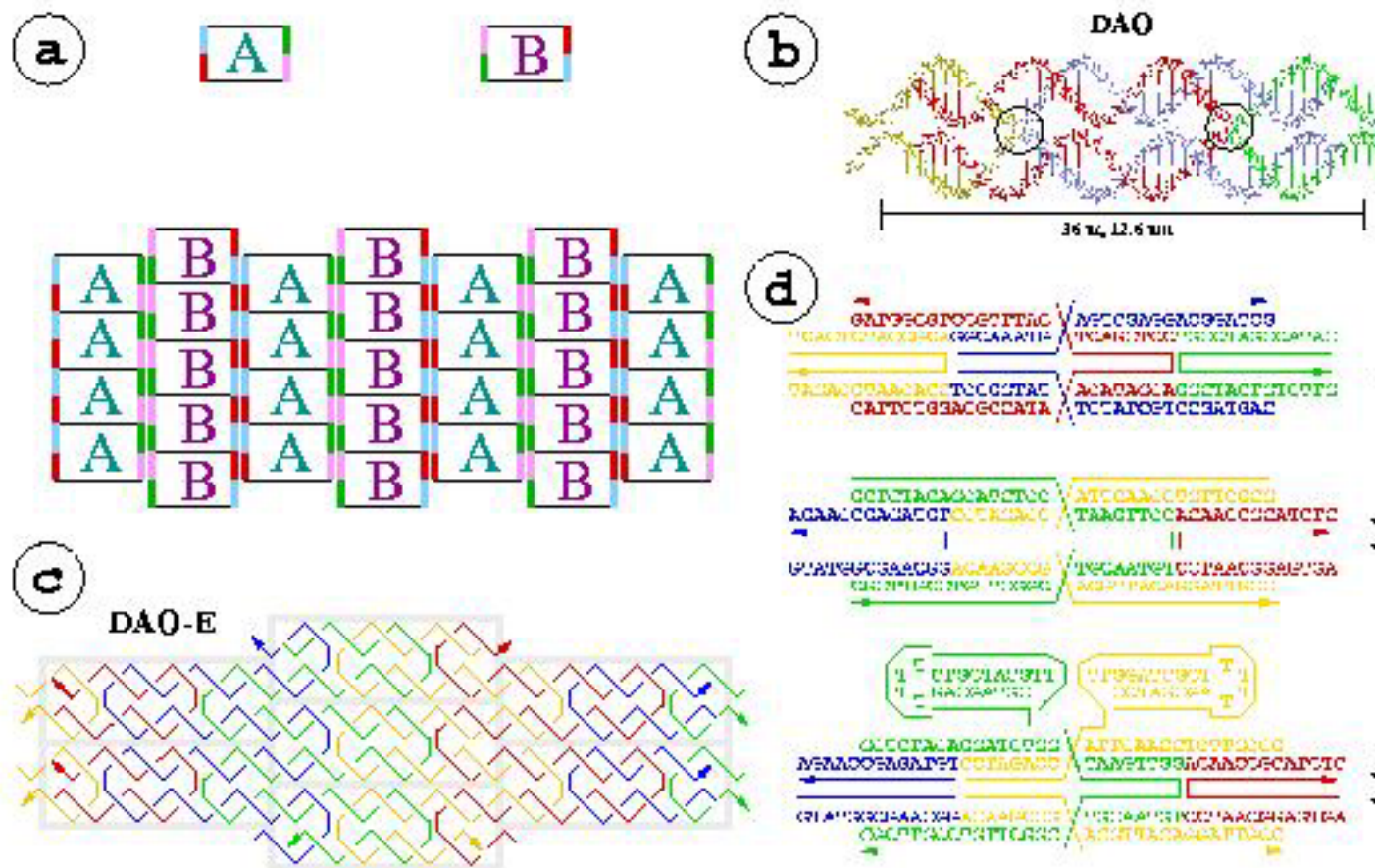
(b)



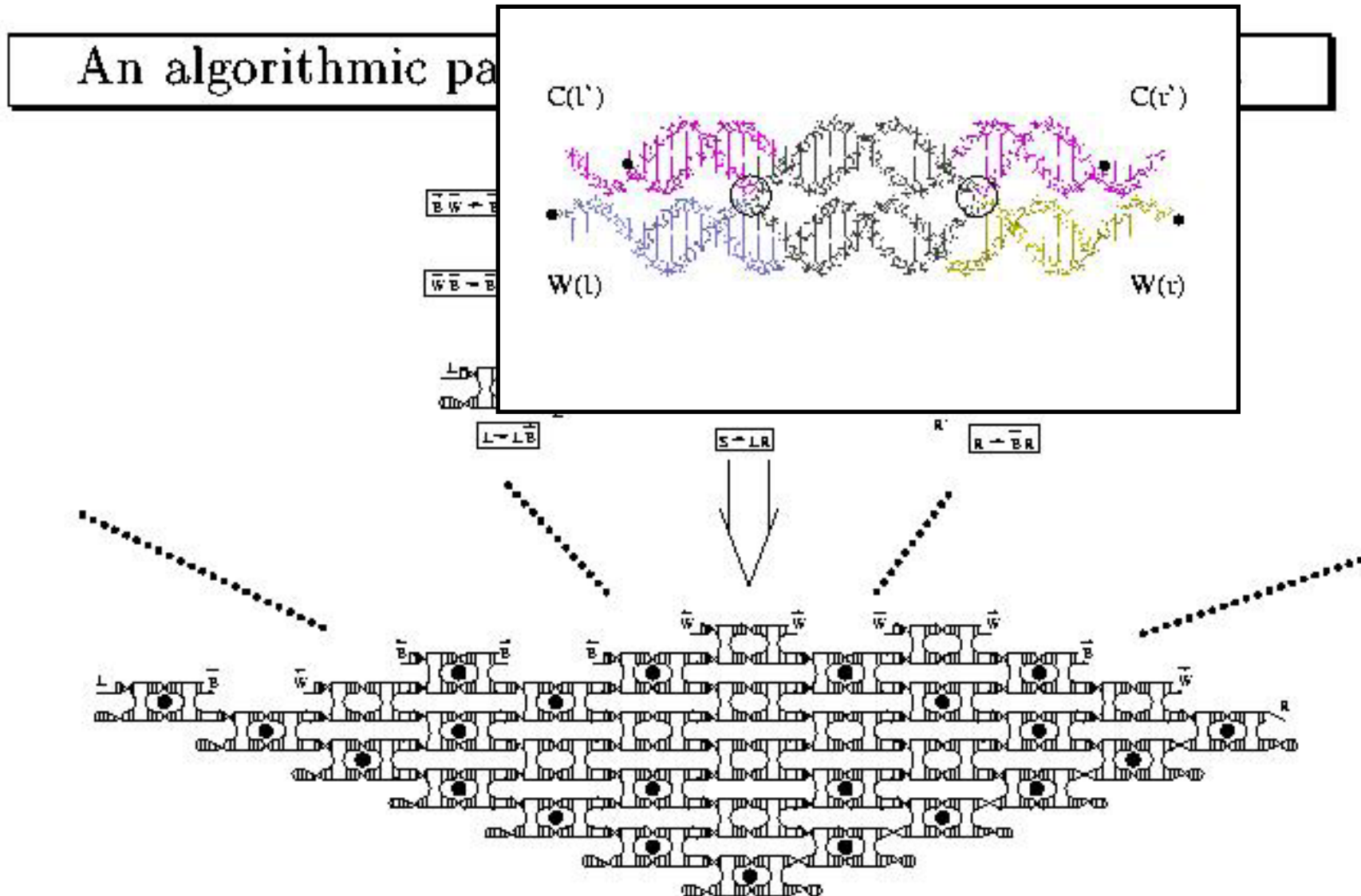
AFM images of unmodified DAO-E AB lattice.



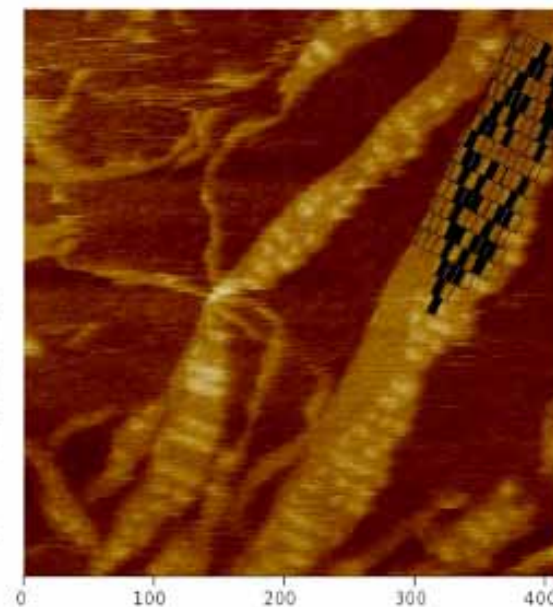
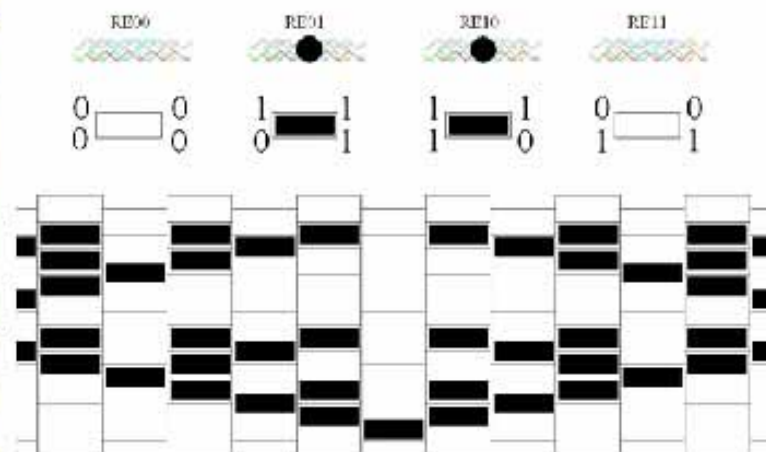
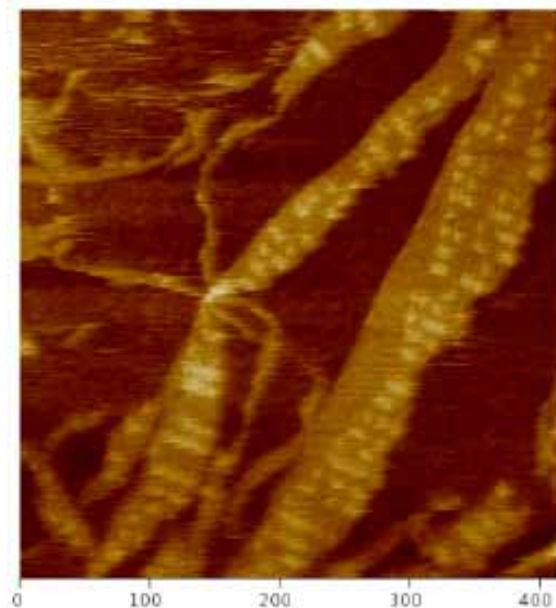
Design of DX molecular structure.



Winfreeのタイリング



Sierpinskiの三角形



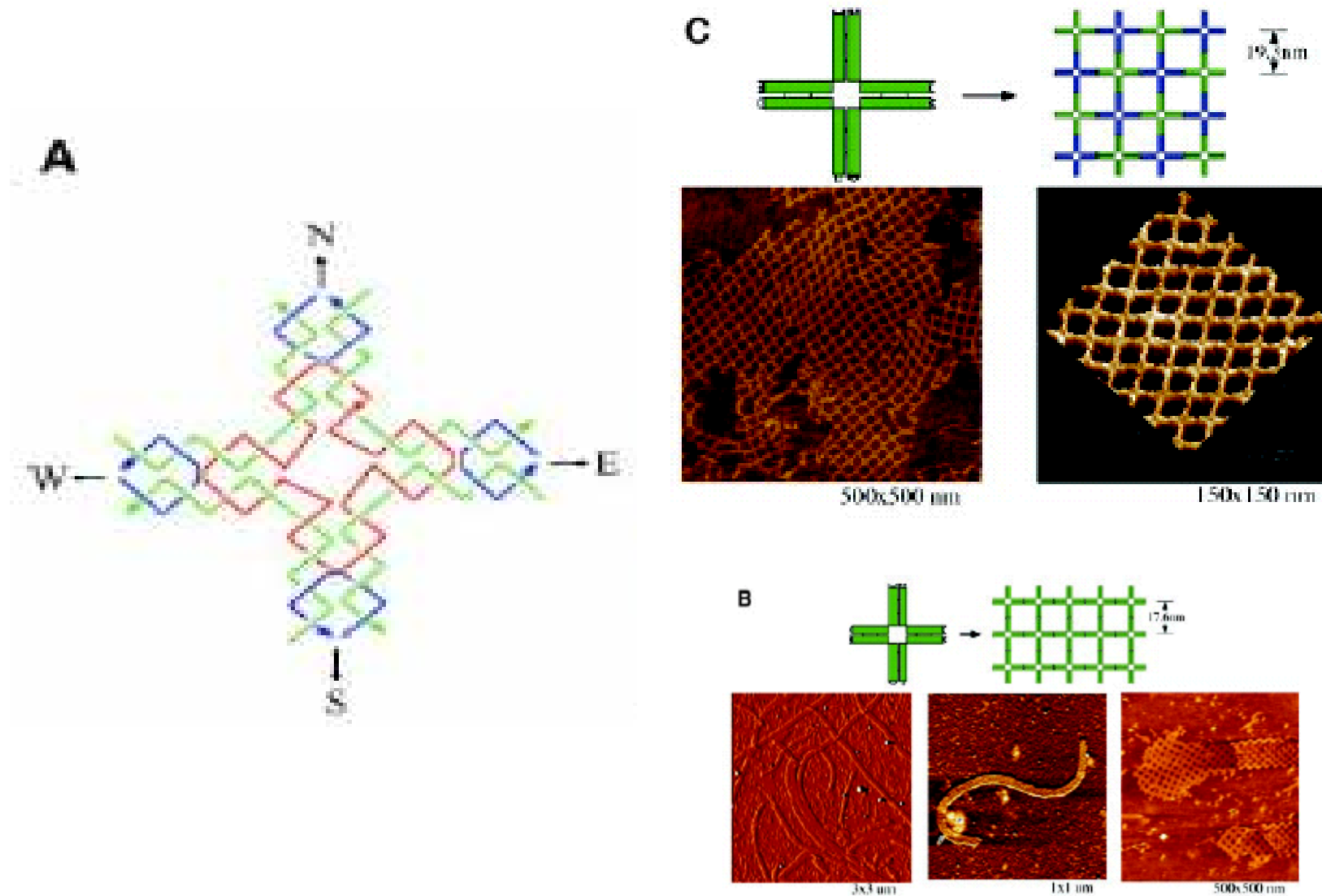
DNA9, 2003

計算能力に関するWinfreeの結果

- (線形) 構造分子によって生成される言語族
= 正則言語族
- (線形 + ヘアピン + 3 分岐) 構造分子によって生成される言語族
= 文脈自由言語族
- (線形 + DX) 構造分子によって生成される言語族
= 帰納的可算言語族
= チューリング計算可能

2次元のタイリングは
チューリング完全

LaBean's Four Four-arm Junctions



Seeman-Winfreeパラダイム...

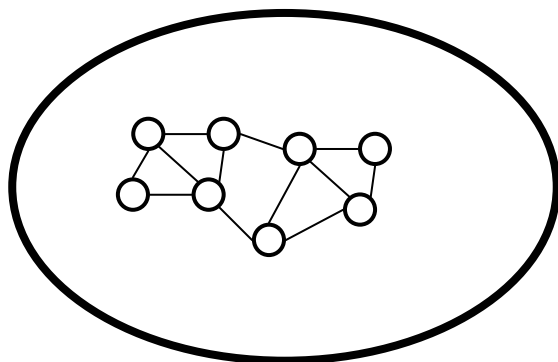
- 分子コンピューティング研究の主流に
 - International Society for Nanoscale Science, Computation and Engineering
- 複雑な構造の自己組織化
- 静的な構造から動的な構造(分子マシン)へ
ナノロボティクス
- システマティックな設計論が必要
分子プログラミング
 - DNAの場合 --- 配列設計
 - 構造 \Rightarrow 配列 (inverse folding)

分子マシン(ナノロボティクス)

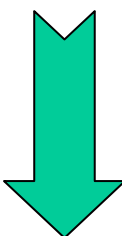
- アクチュエータとしてのマシン
 - モーター
 - トランスポータ
- 抽象的なマシン --- 有限状態機械(オートマトン)
 - 有限の状態を持つ。
 - その状態を自律的もしくは入力に従って遷移させる。
 - 出力を行うかもしれない。
 - 汎用コンピュータへの第一歩
 - 多くの応用
 - スイッチ・センサー
 - メモリ(記憶保持・アドレッシング)
- 両者はまだ渾然一体としている。

分子システムによる有限状態機械

分子システム



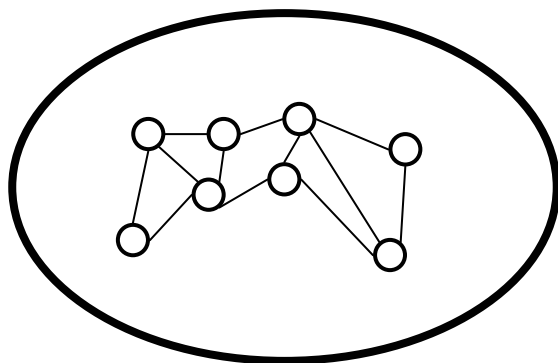
情報処理



計算

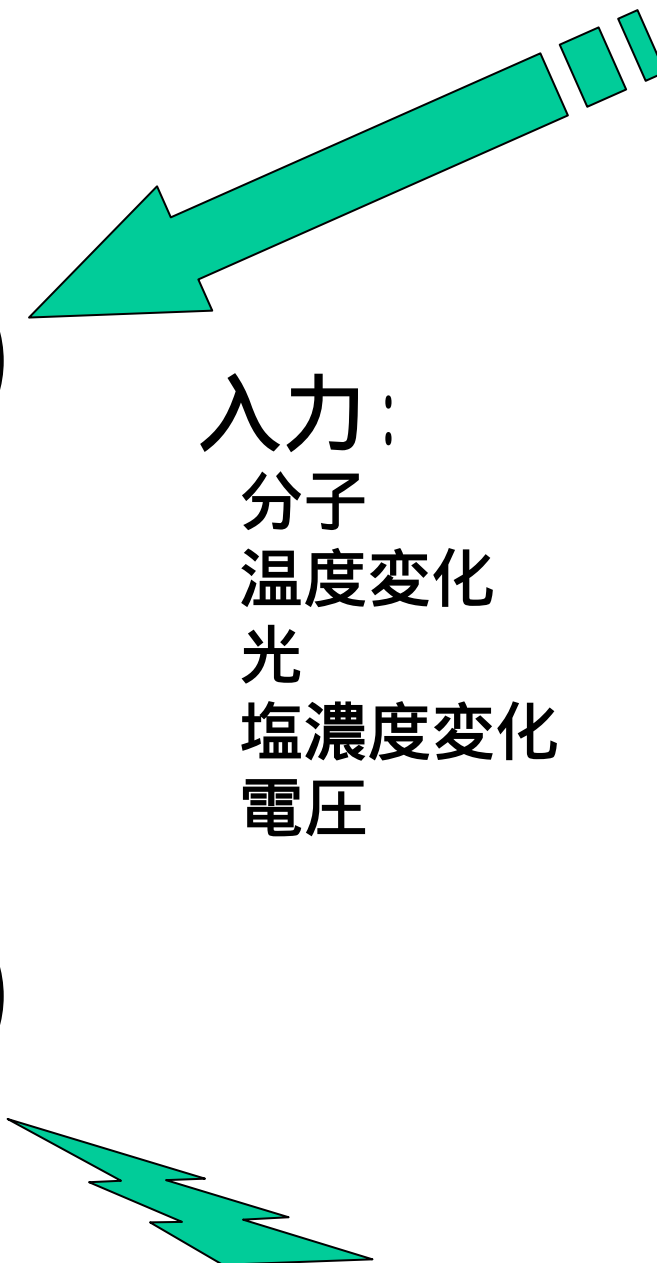
出力:

移動
形態変化
構造形成
光
電気
熱



入力:

分子
温度変化
光
塩濃度変化
電圧

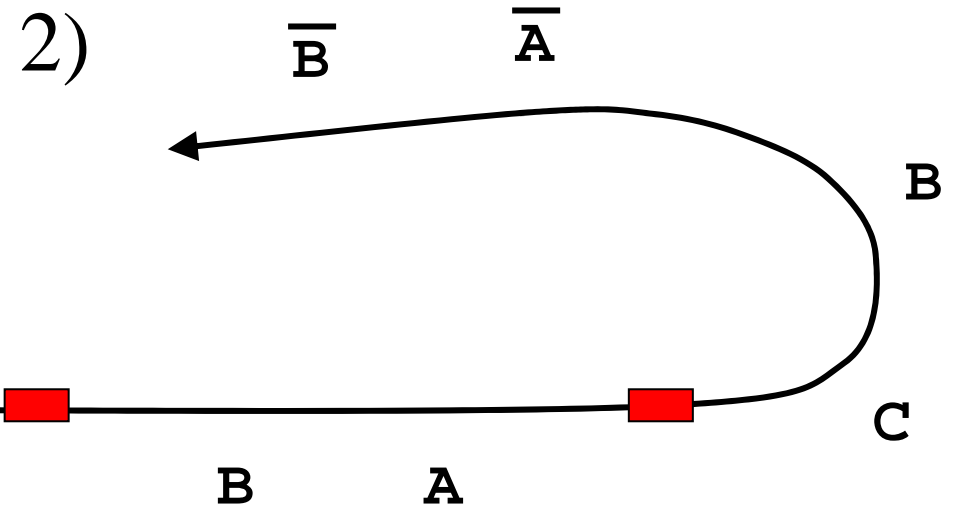
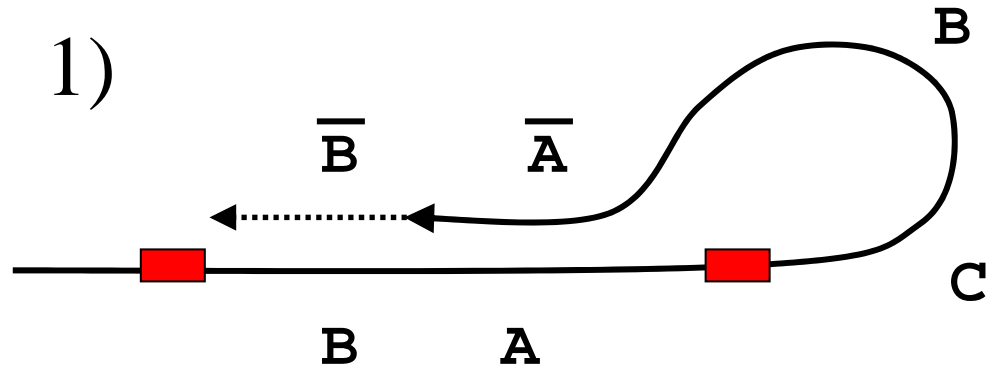


分子(DNA)状態機械

- 末端配列機械
 - 末端配列が状態を表現
 - Whiplash PCR (鞭打ちPCR)
 - 状態が遷移するにつれ長くなる。
 - Shapiroのオートマトン
 - 状態が遷移するにつれ短くなる。
- 形態機械 (Conformational Machine)
 - 分子の形態が状態を表現
 - YurkeのMolecular Tweezers (分子ピンセット)
 - SeemanのPX-JX₂ Switch
 - 我々のHairpin-Based Machine

Whiplash PCR (WPCR)

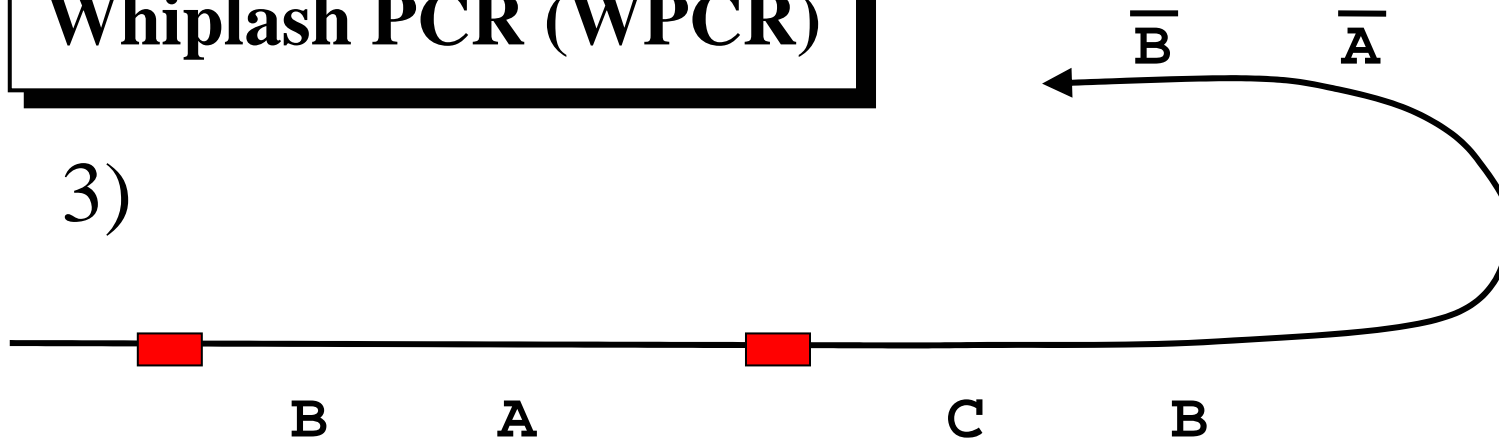
■ : stopper sequence



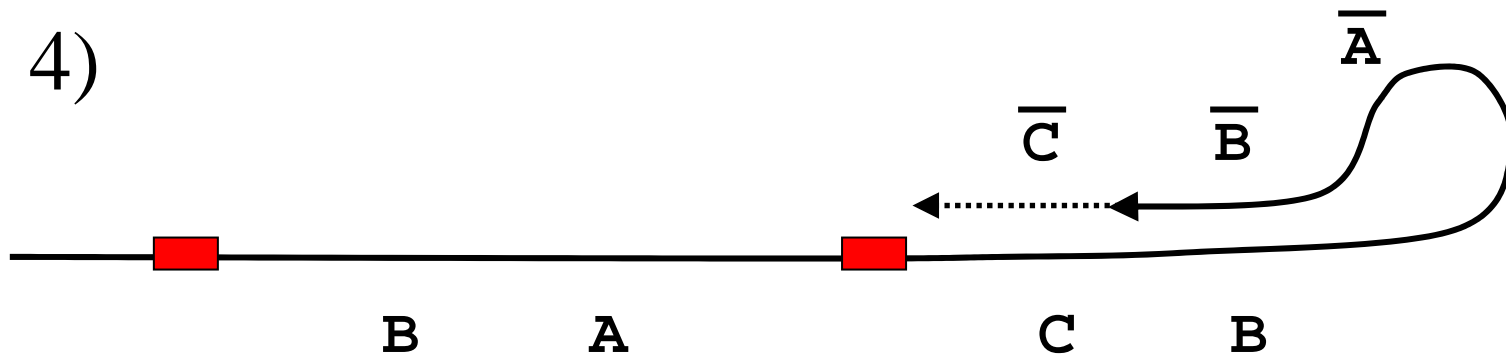
Komiya *et al.*

Whiplash PCR (WPCR)

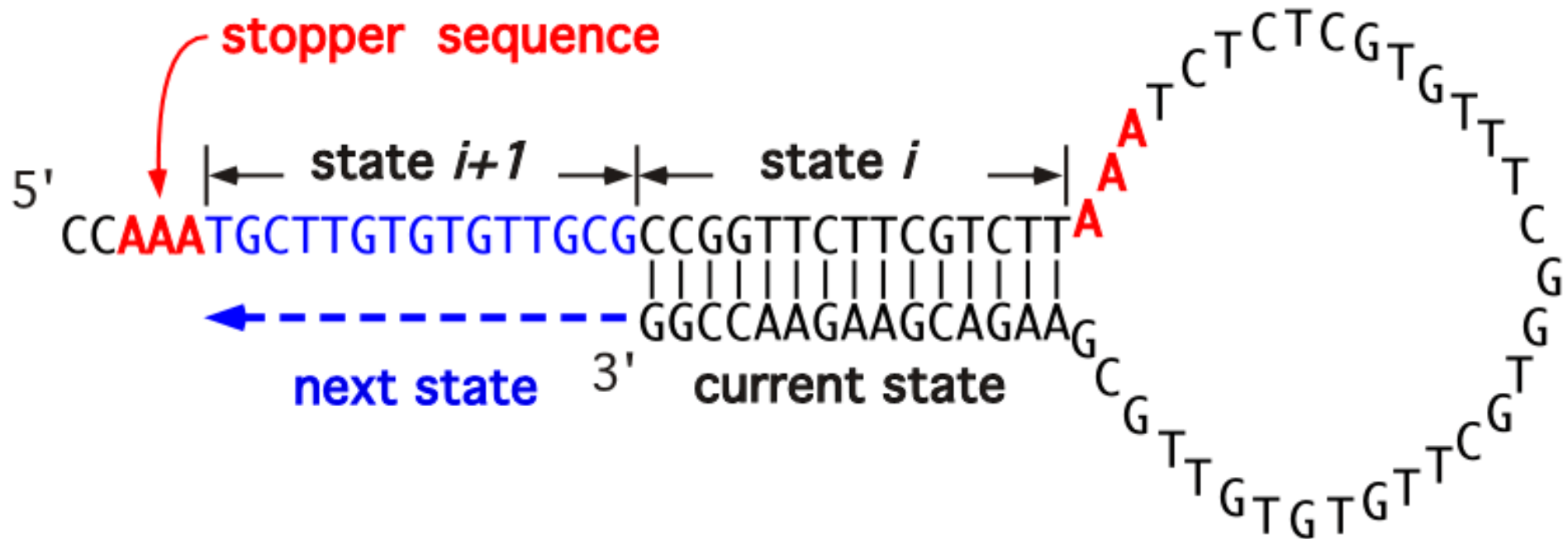
3)



4)



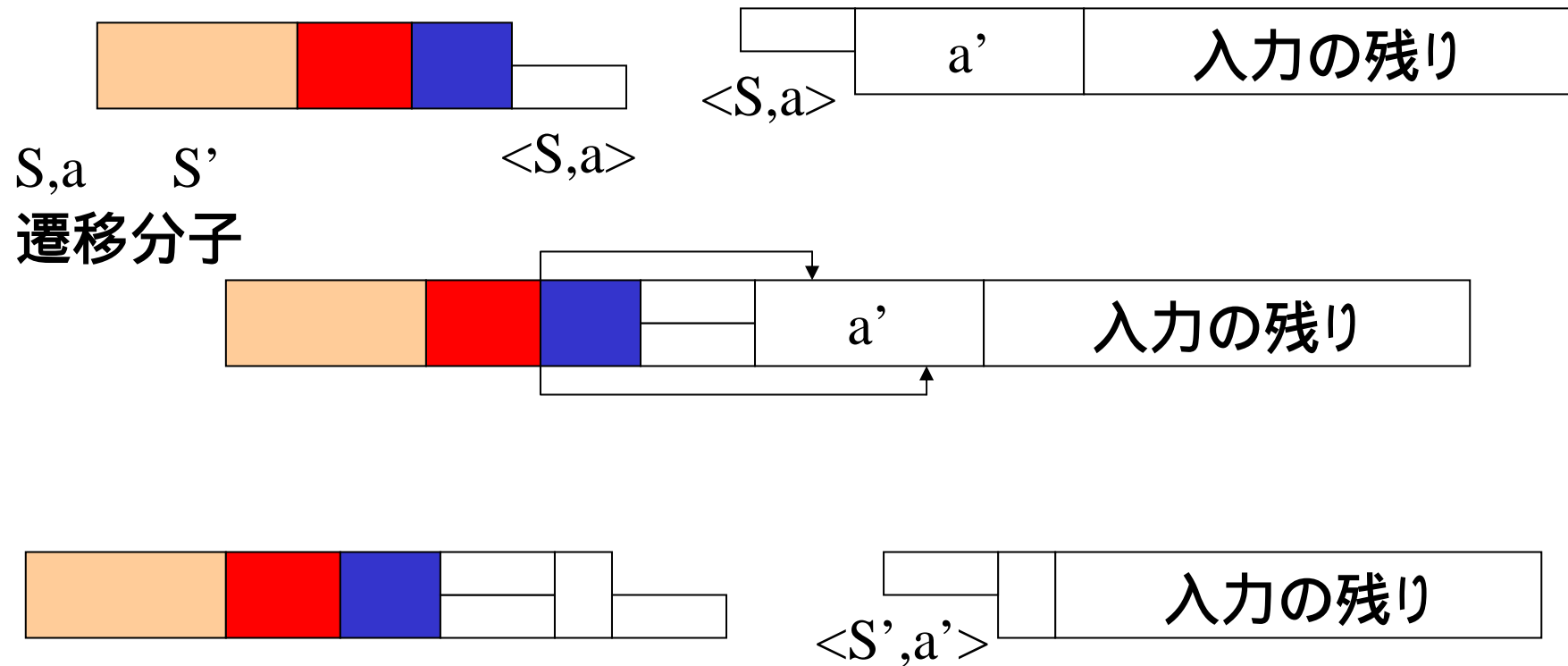
Polymerization Stop



Polymerization by DNA polymerase
with dATP, dCTP, dGTP

ShapiroのDNAオートマトン

IIS型制限酵素
認識部位 スペース



入力文字 a' の配列は、各 S' に対して $\langle S', a' \rangle$ を含んでいる。
遷移分子は、スペースを調整して、適切な個所で切断する。

ShapiroのDNAオートマトン

- *Nature* 2001
- 2入力文字、2状態
- *FokI*

a=CTGGCT

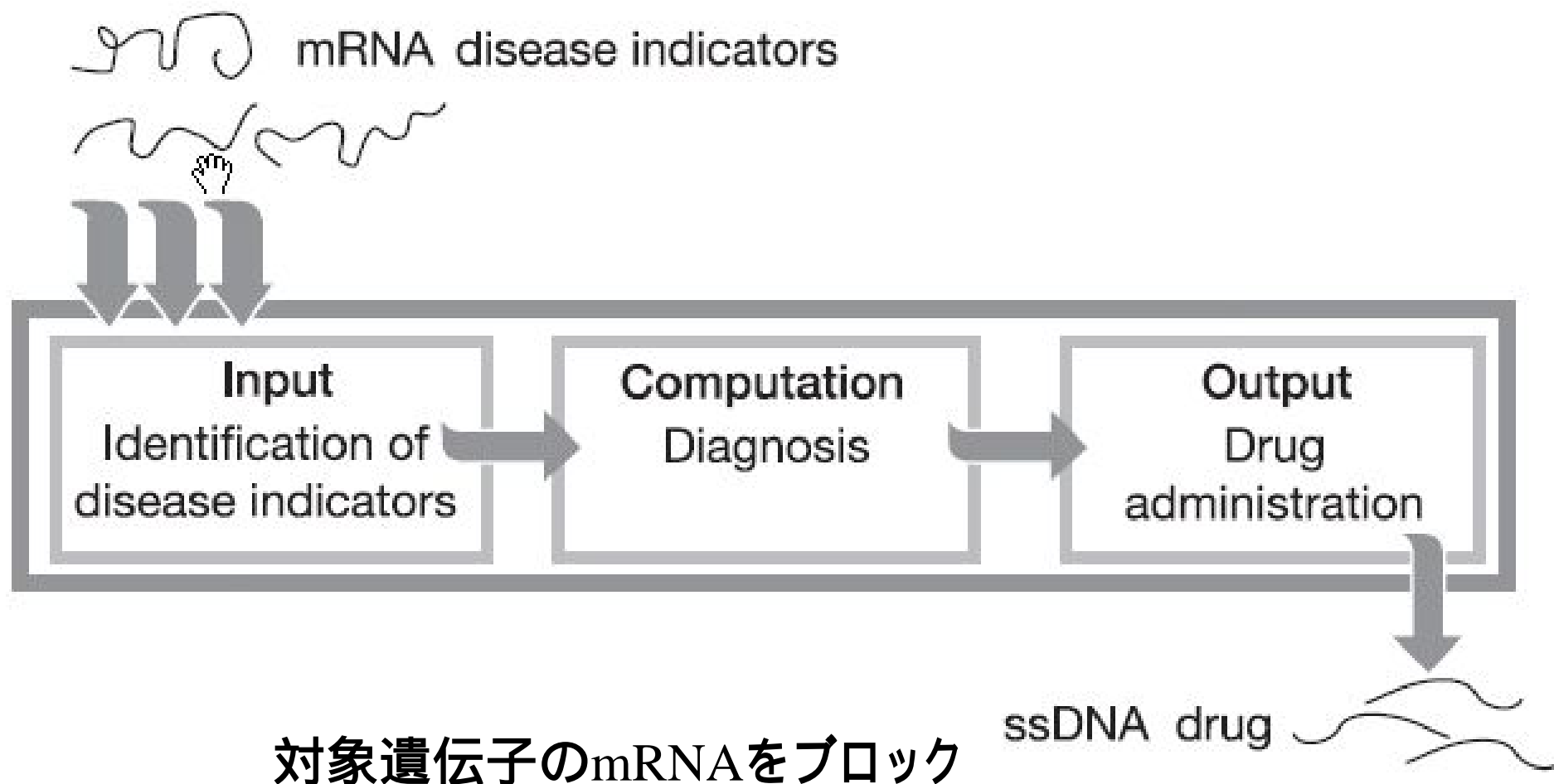
b=CGCAGC

5' -p...22...GGATGTAC
3' -GGT...22...CCTACATGCCGAp S0,a S0

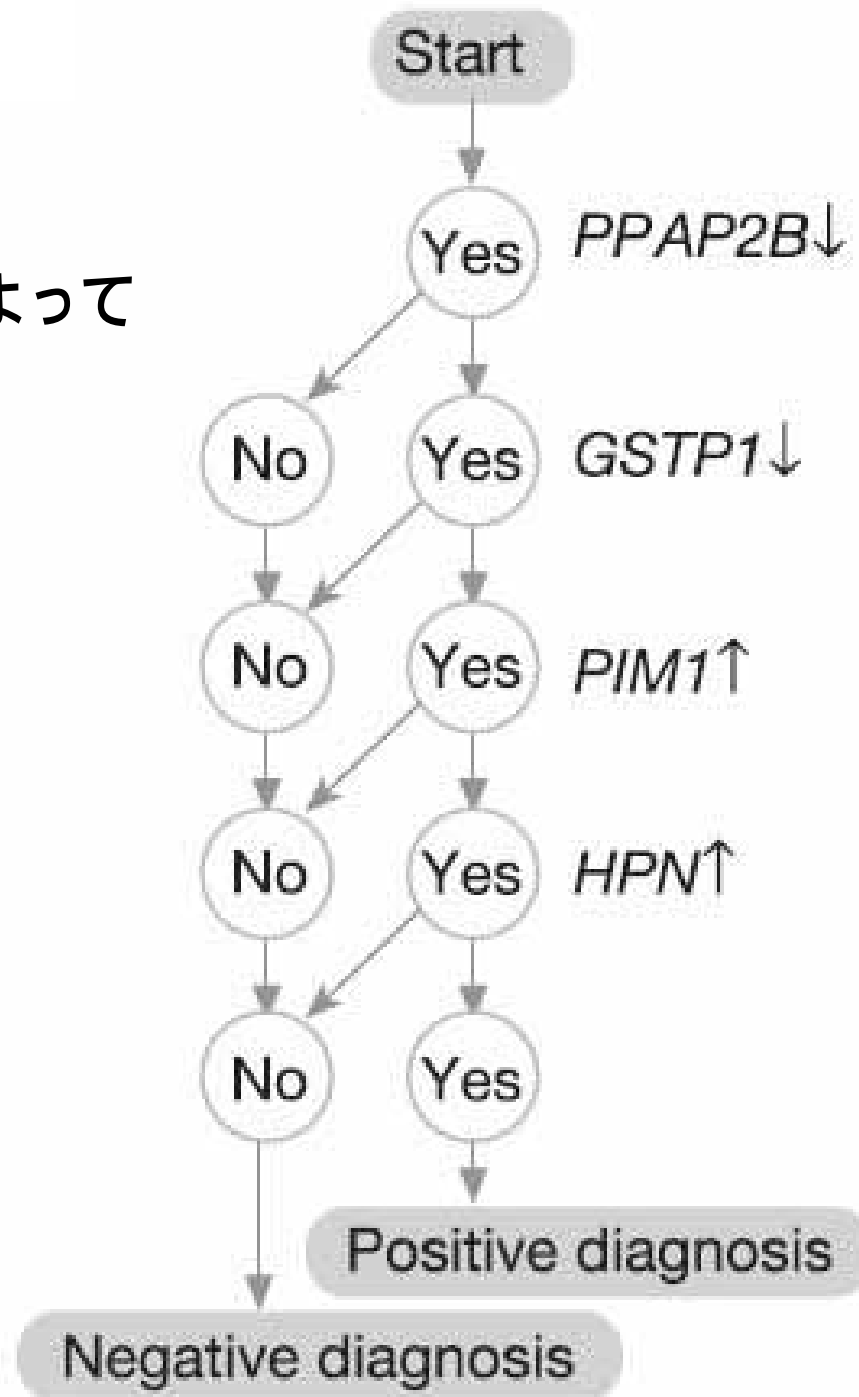
5' -p...22...GGATGACGAC
3' -GGT...22...CCTACTGCTGCCGAp S0,a S1

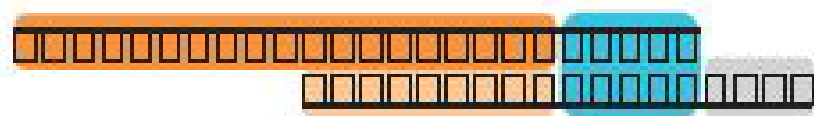
体内における投薬制御への応用

- *Nature* 2004

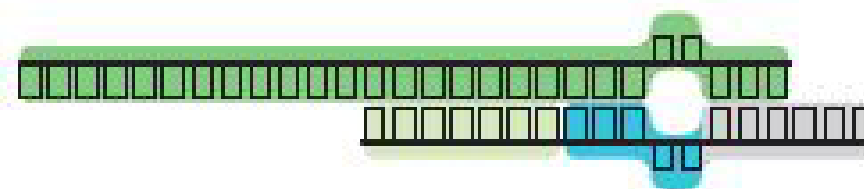


二分木プログラムを実現。



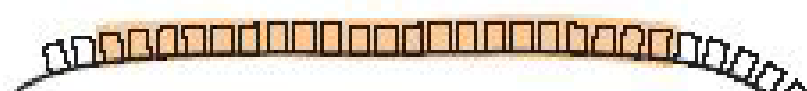


Active Yes $\xrightarrow{PIM1 \uparrow}$ No



Inactive Yes $\xrightarrow{PIM1 \uparrow}$ Yes

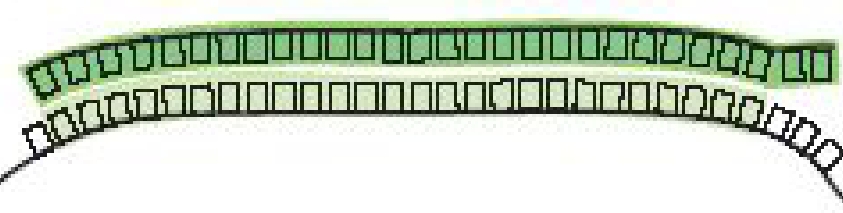
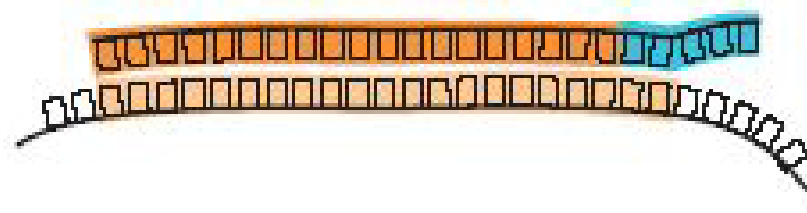
+



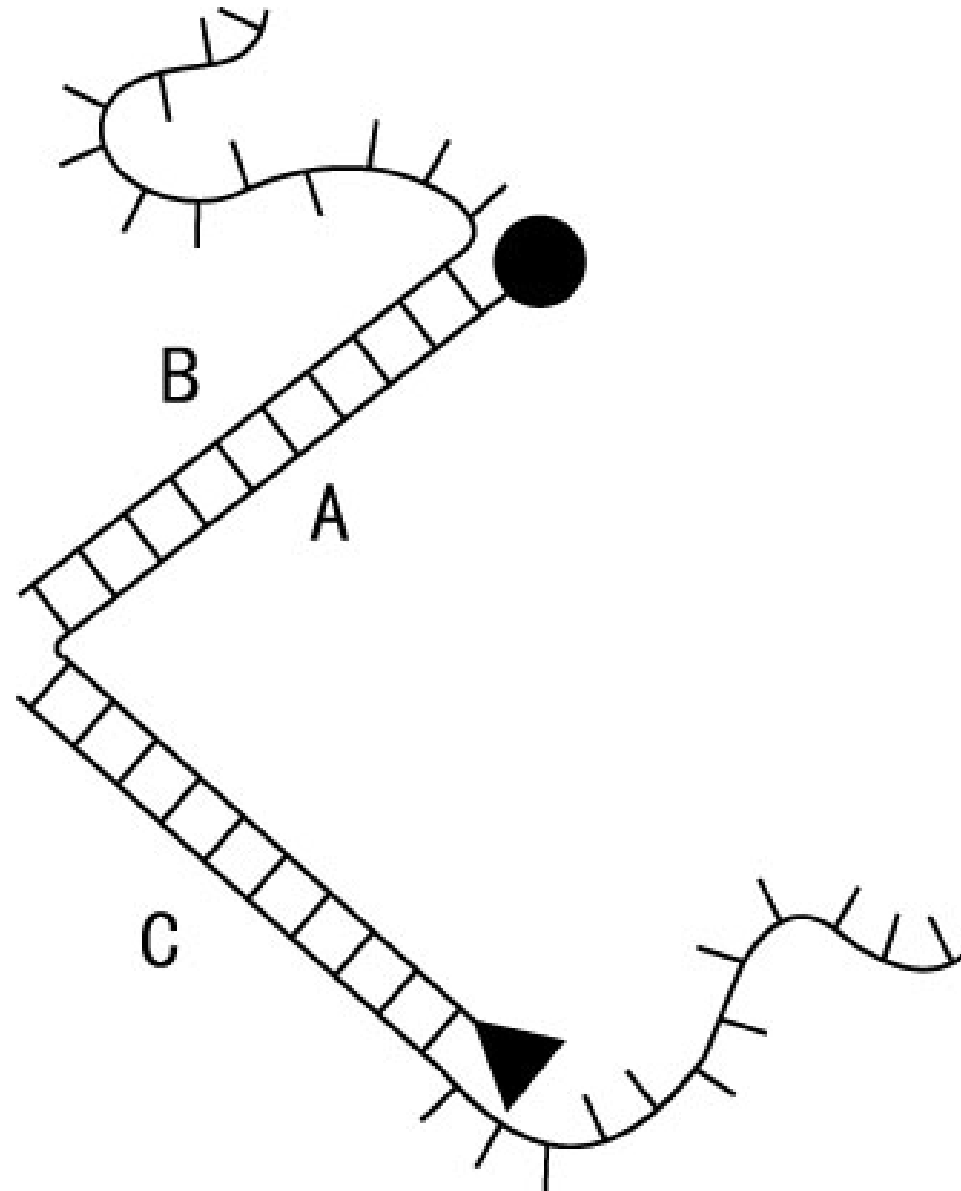
PIM1 inactivation tag

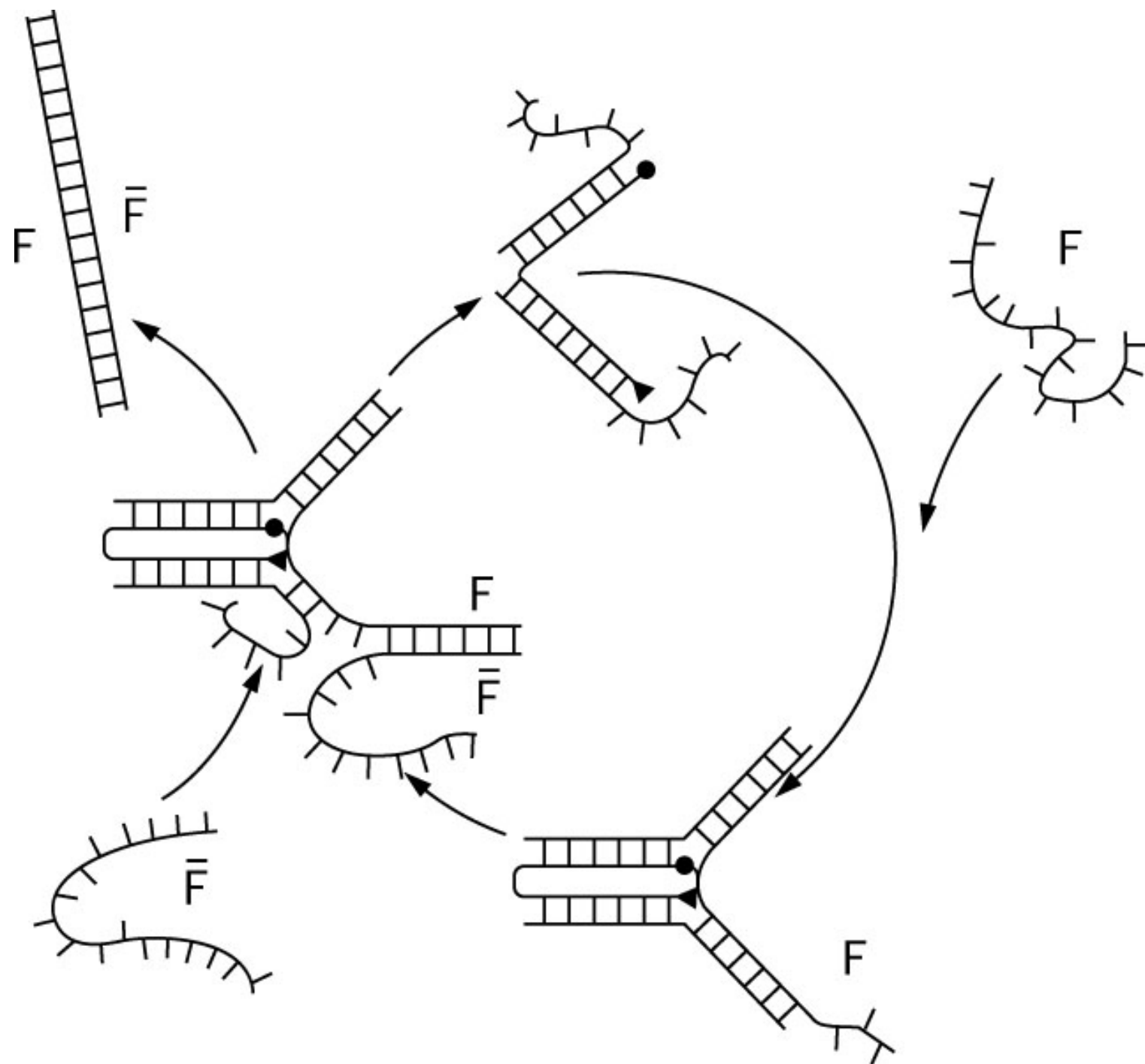


PIM1 activation tag



YurkeのDNAピンセット





分子コンピューティングの将来性

- 分子による超並列計算 --- 当初とは異なる方向
 - 分子レベルの情報処理：
遺伝子計測へ応用 知的なDNAチップ
 - 並列計算：
進化計算としての分子進化
GAなどの進化計算の知見を分子進化へ応用
- DNAの自己組織化
 - DNA = 糊・ブロック
 - ボトムアップ・ナノテクノロジーの主流に
 - 実用化には、トップダウン・ナノテクノロジーとの融合が不可欠

分子コンピューティングの将来性

- 分子マシン
 - スイッチやセンサー
 - メモリ(記憶保持・アドレッシング)
 - 自己組織化との融合 ナノ工場
 - 生体内の情報処理 遺伝子制御・投薬制御
- 分子マシンから細胞マシンへ
 - 細胞による情報処理
 - 人工遺伝子ネットワーク
- 新しい計算パラダイム
 - 膜コンピューティング
 - アモルファス・コンピューティング

分子システムの設計論

- ad hocなtrial&errorから
システムティックな設計論へ
- 特に計算論的な側面 **分子プログラミング**
- 分子の設計
 - DNAの場合 = 配列設計
 - 構造 配列 (inverse folding)
 - 自己組織化パターンの設計
 - 分子マシンの設計
- 分子反応の設計
 - 反応条件の設定
 - 実験操作のスケジューリング

配列設計

- 配列セットの評価

- ミスハイブリダイゼーションの回避

- ハミング距離

- エネルギー計算

mfold (Zuker)、Viennaパッケージ

- 一様な T_m (融解温度、melting temperature)

- 配列セットの探索

- 遺伝的アルゴリズム

- 符号理論

意図しない構造を
作らない配列セットの生成

- 逆問題

- 構造 配列 (inverse folding)

- Viennaグループ

意図した構造を
作る配列の設計

Inverse Folding

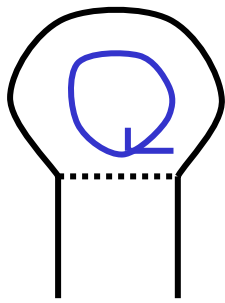
- Viennaグループ
- コスト関数の最小化による配列の探索

$$\Xi(x) = E(x, \Omega) - G(x) = -RT \ln p$$

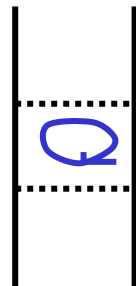
- Ω : 目的の構造
- x : 配列
- $E(x, \Omega)$: x における Ω の自由エネルギー
- $G(x)$: 配列 x の集団自由エネルギー (McCaskill)
- p : x における Ω の確率

DNA/RNAの二次構造

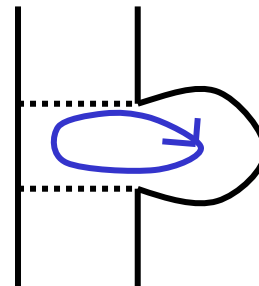
- ベースペアの i, j の集合
- k -loop --- k 個のベースペアで囲まれたループ



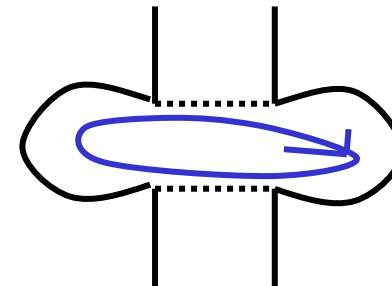
hairpin



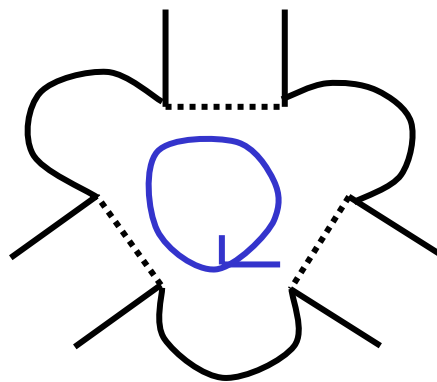
stack



bulge

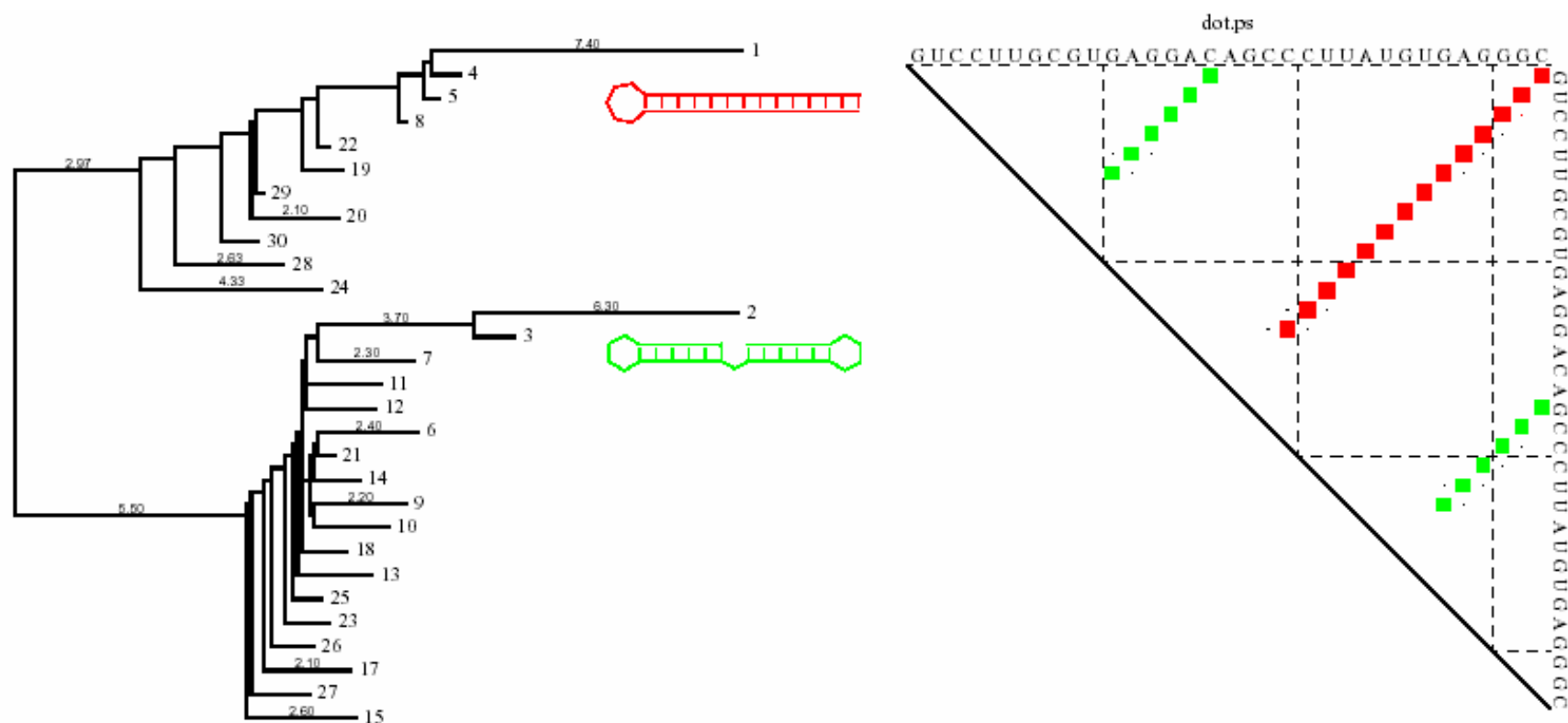


interior

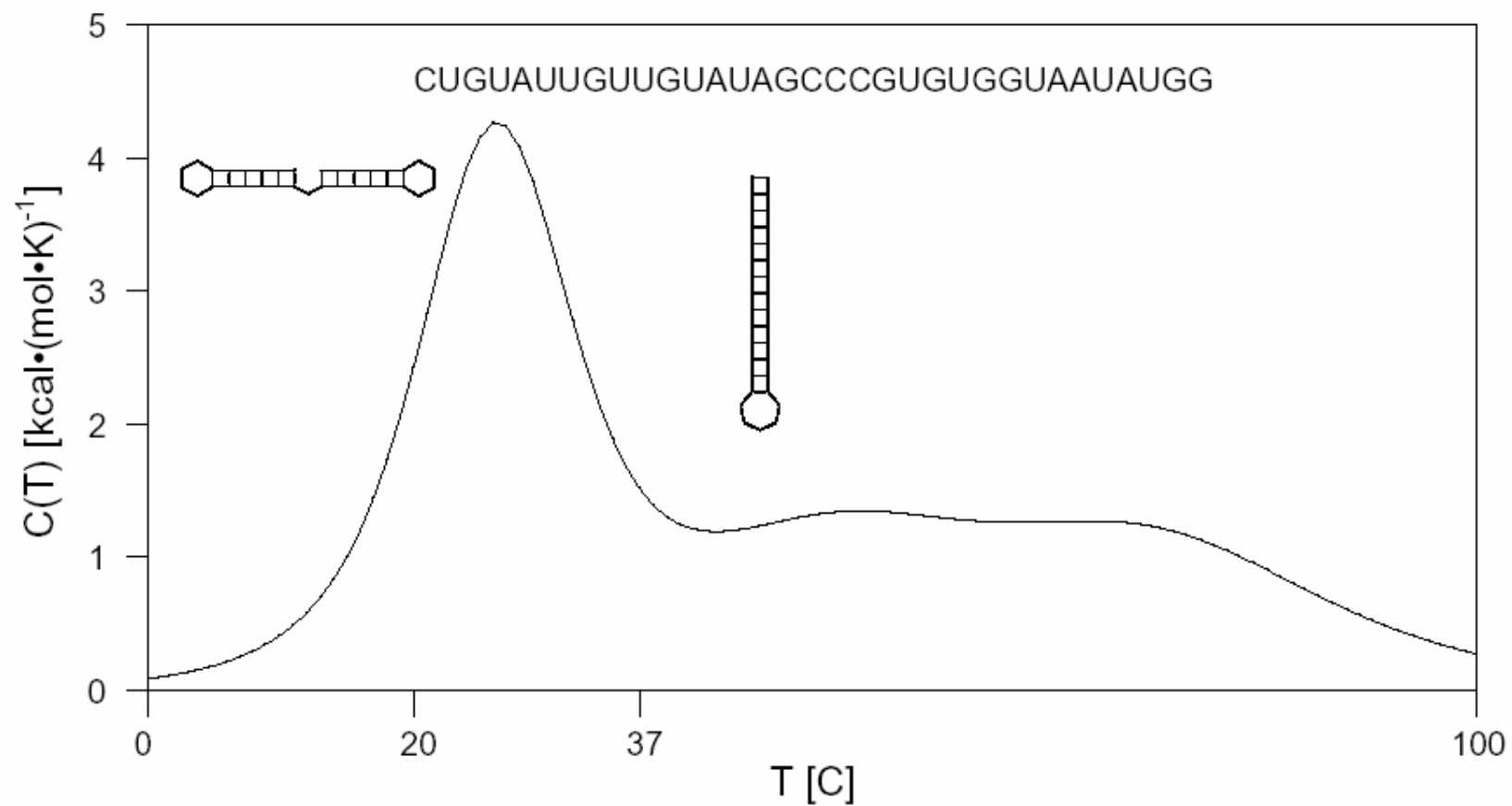


3-loop

- 各種のループに対して
エネルギー・パラメータを割り当てる
(nearest neighbor model)
- 動的計画法により、自由エネルギーが
最小の構造を計算

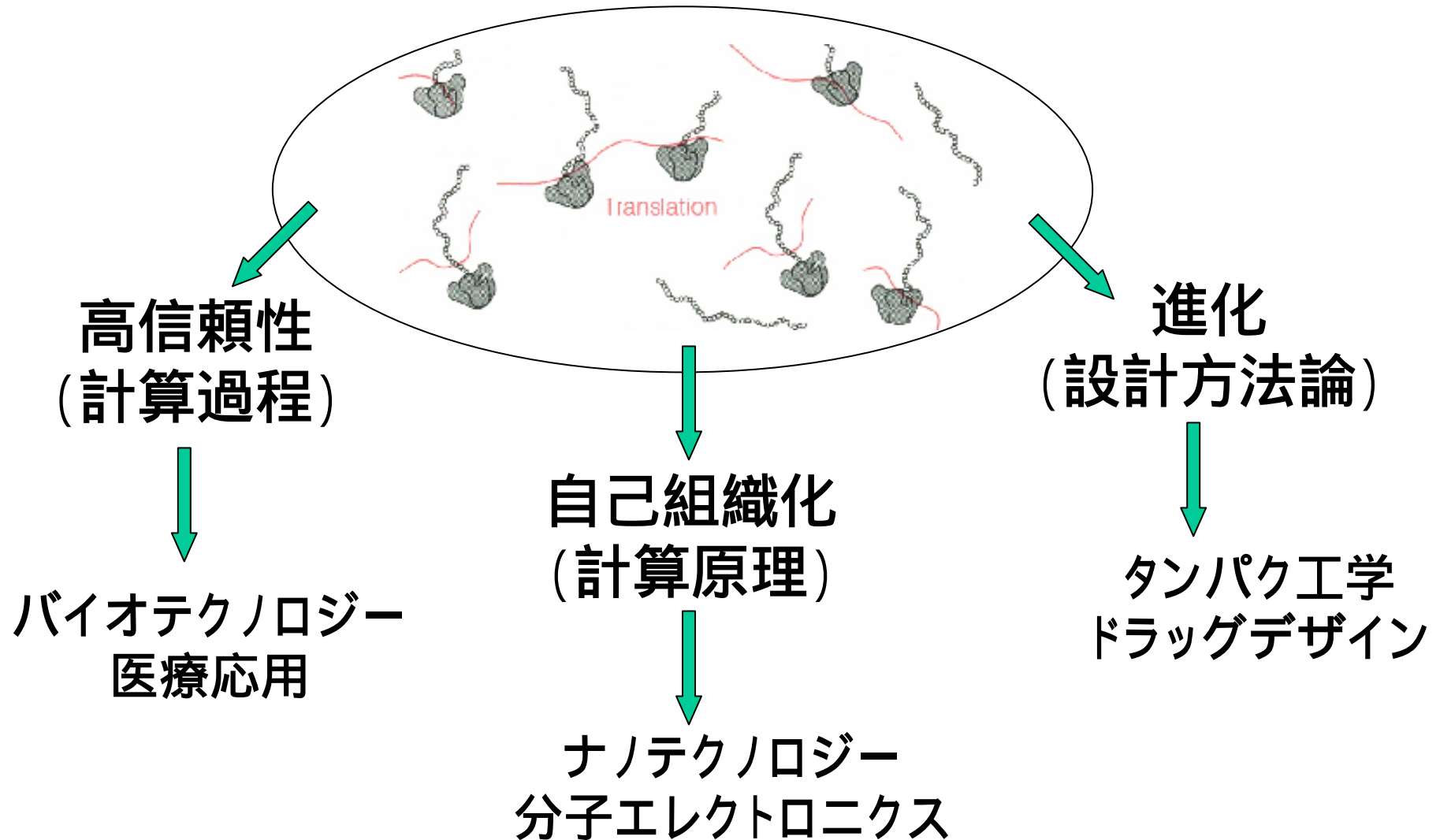


$$\Xi(x) = E(x, \Omega_1) + E(x, \Omega_2) - 2G(x) + \xi \left(E(x, \Omega_1) - E(x, \Omega_2) \right)^2$$

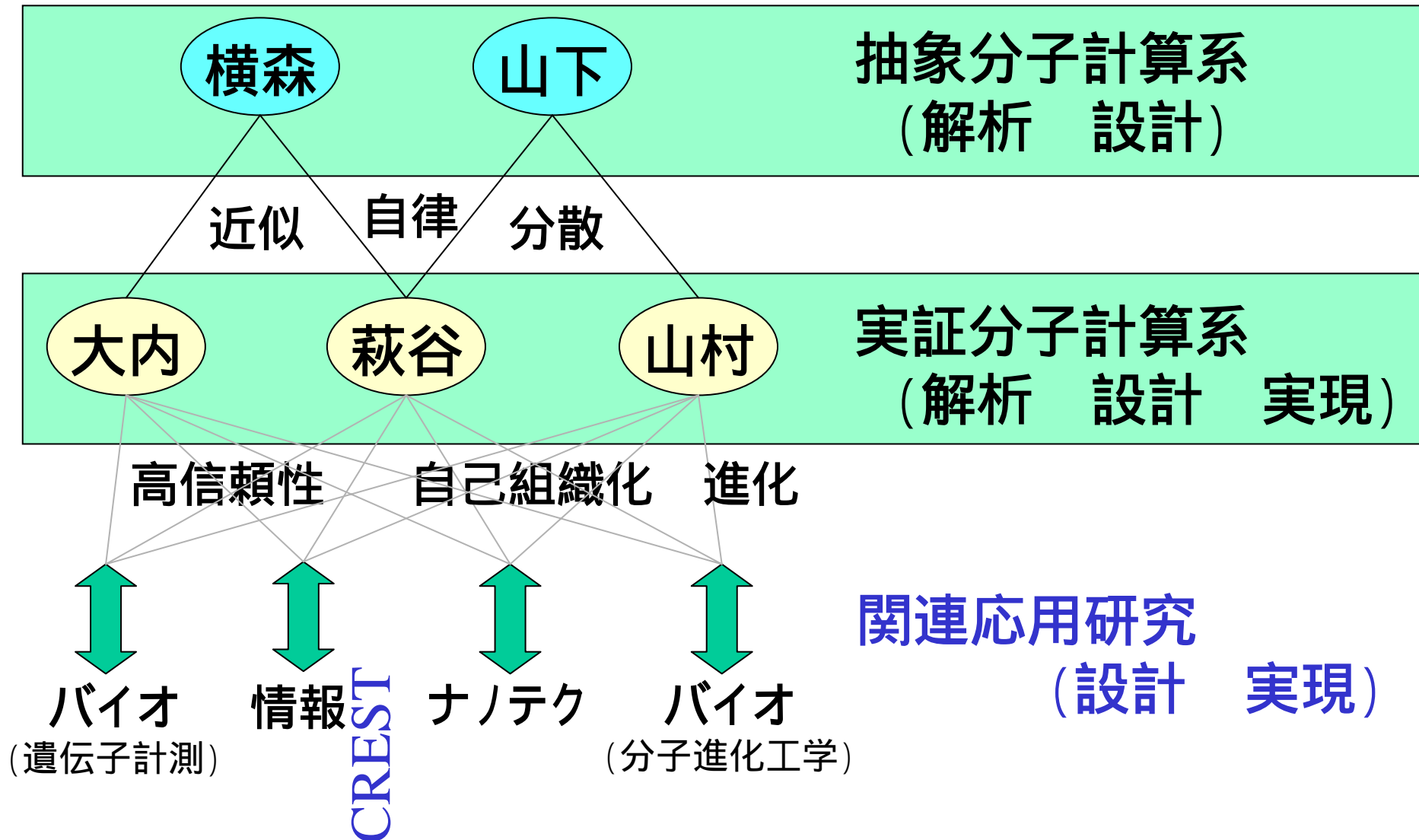


$$\begin{aligned} \Xi(x) = & \left(E_{T_1}(x, \Omega_1) - G_{T_1}(x) \right) + \left(E_{T_2}(x, \Omega_2) - G_{T_2}(x) \right) \\ & + \xi \left\{ \left(E_{T_1}(x, \Omega_1) - E_{T_1}(x, \Omega_2) \right) + \left(E_{T_2}(x, \Omega_2) - E_{T_2}(x, \Omega_1) \right) \right\} \end{aligned}$$

特定領域「分子プログラミング」 設計論のための3つの切り口



計画研究の組織と関連



総括班

- 総括班
 - 有川(九大)
 - 佐藤(京大)
 - 米澤(東大)
 - 小林(東工大)
 - 小長谷(北陸先端大・理研)
 - 都甲(九大) 自己組織化に関する助言
 - 塩谷(東大) 超分子に関する助言

抽象分子計算系

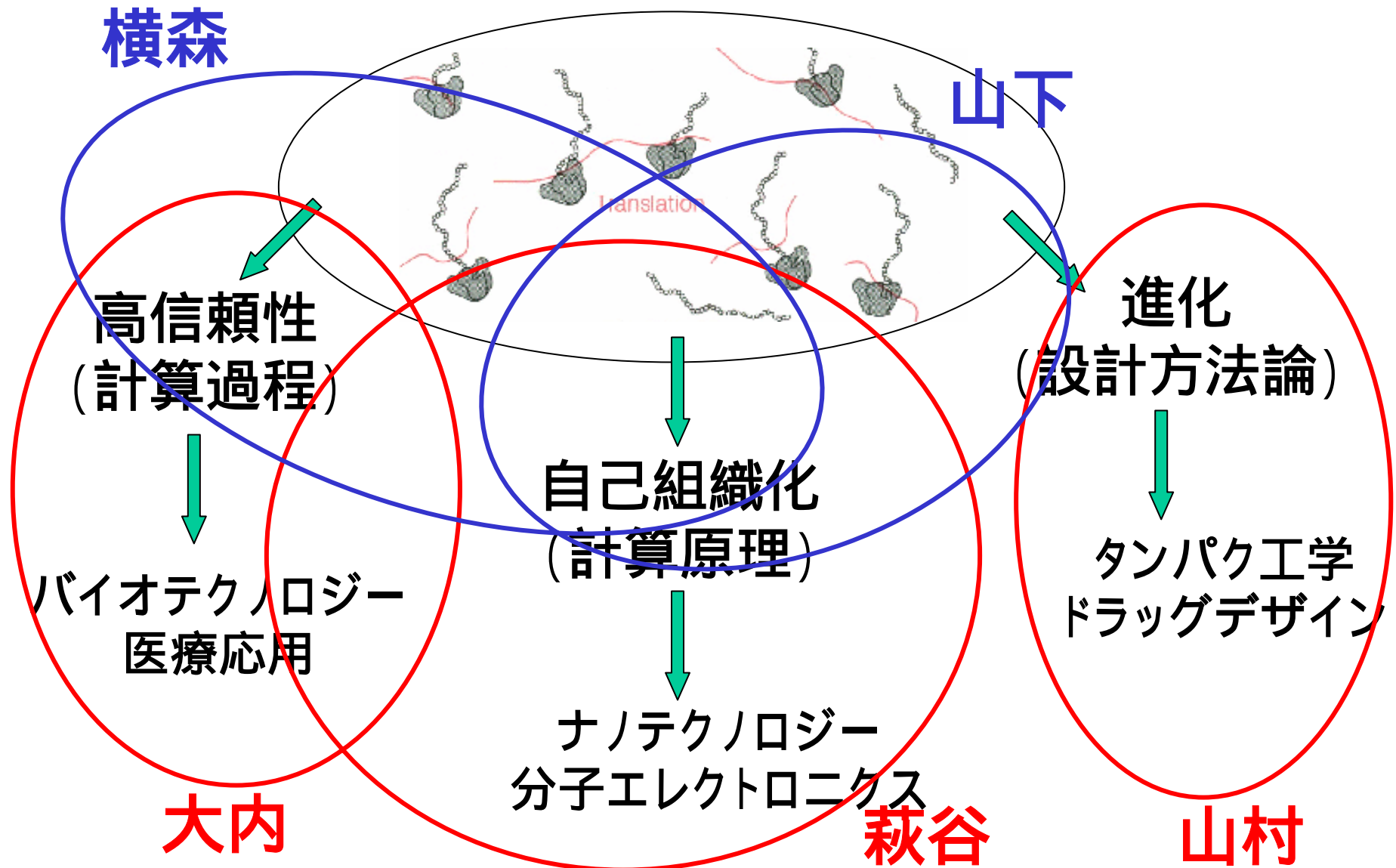
- 横森班: 分子計算のための形式言語理論・
膜計算・近似計算・並行計算
 - 横森・上田(早大)
 - 鈴木(東京医科歯科大)他
- 山下班: 分子計算のための分散計算
 - 山下(九大)他

アルゴリズム論的観点からの
反応解析・反応設計

実証分子計算系

- 大内班: 分子計算反応の高信頼化
バイオテクノロジーへの応用
 - 大内・山本・川村(北大)
- 萩谷班: 分子の設計・分子反応の設計
ナノテクノロジー・
バイオテクノロジーへの応用
- 山村班: 進化計算のタンパク工学への応用
 - 山村(東工大)
 - 坂本(東大)
 - 伏見(埼玉大) 他

設計論のための3つの切り口



これまでの主要な研究成果

- 分子設計 = 配列設計

- 構造を作らない設計 小林_(横森班)・朝廣_(山下班)
- 構造を作る設計 村田_(萩谷班)
- 構造を変化させる設計 萩谷・小野_(山下班)

- 反応設計

- 反応の並列化 萩谷
- 反応の精密化 大内
- 進化のための反応 山村

構造を作らない設計

小林 (横森班)

- 構造非形成検証問題 $SFTP(C, T)$
上の言語 R C が T に関して
構造非形成のとき yes、そうでないとき no
- C : 上の言語クラス
 $M = \{S^+ \mid n \geq 1, S \in M_n\}$
 Reg 正則言語のクラス
- T : 二次構造のクラス
 T_{pnf} 線形な二次構造のクラス
 T_{lin} シュードノットのない二次構造のクラス

未解決問題 [Condon03]

$SFTP(Reg, T_{pnf})$ は効率よく解けるか？

↓ まず、簡単な場合を考察

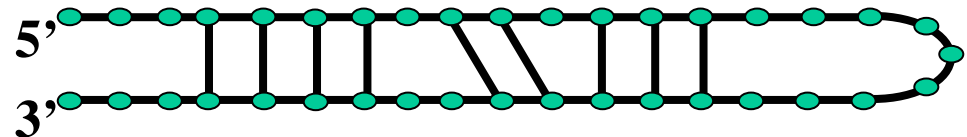
M : 与えられた配列セット S に属する配列の連結

$$M = \{S^+ \mid n = 1, S = n\}$$

T_{lin} : 線形な二次構造のクラス

$SFTP(M, T_{lin})$ は $O(m^6 n^6)$ 時間で解ける。
(ただし, $m = |S|$, n は S の要素の長さ)

T_{lin} : 線形な二次構造

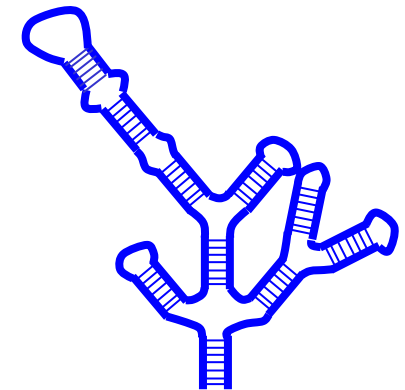


最新の結果

Reg : 正則言語のクラス

$SFTP(Reg, T_{pnf})$ は $O(m^8)$ 時間で解ける。
(ただし, m は 有限オートマトンの状態数)

T_{pnf} : シュードノットのない二次構造



今後の課題

- オートマトンのグラフ構造が acyclic な場合にアルゴリズムを高速化する
- 配列設計システムへの応用

構造を作らない設計

朝廣(山下班)

- 配列セットの設計における種々の貪欲法の有効性
 - ハミング距離による重なりをなるべく大きくとるGreedyPack、
 - ハミング距離の順に配列を網羅するGreedyHD
 - グレイコードに従って配列を数え上げるGreedyGray
- GreedyPack、GreedyHD、GreedyGrayの順に、実行時間は小さくなるが、得られる配列セットの大きさも小さくなる
- 実際に貪欲法で配列セットを生成することは頻繁に行われている。

構造を作る設計

村田(萩谷班)

- 自己集合するDNAタイルの新しい誤り抑制手法
 - プロテクタをDNAタイルの上に重層
 - DNAタイルの自己組織化が正しく行われたときのみ
プロテクタがはずれて、さらなる自己組織化が可能になる。
- 誤った自己組織化はほとんど進まない。
 - 結果として、正確性と効率の両方を
実現することができた。
- 今後は、DNAを用いて実装を行う計画

Layered Tile Model (LTM)

original tiles



corner tile
(seed tile)

protective tiles



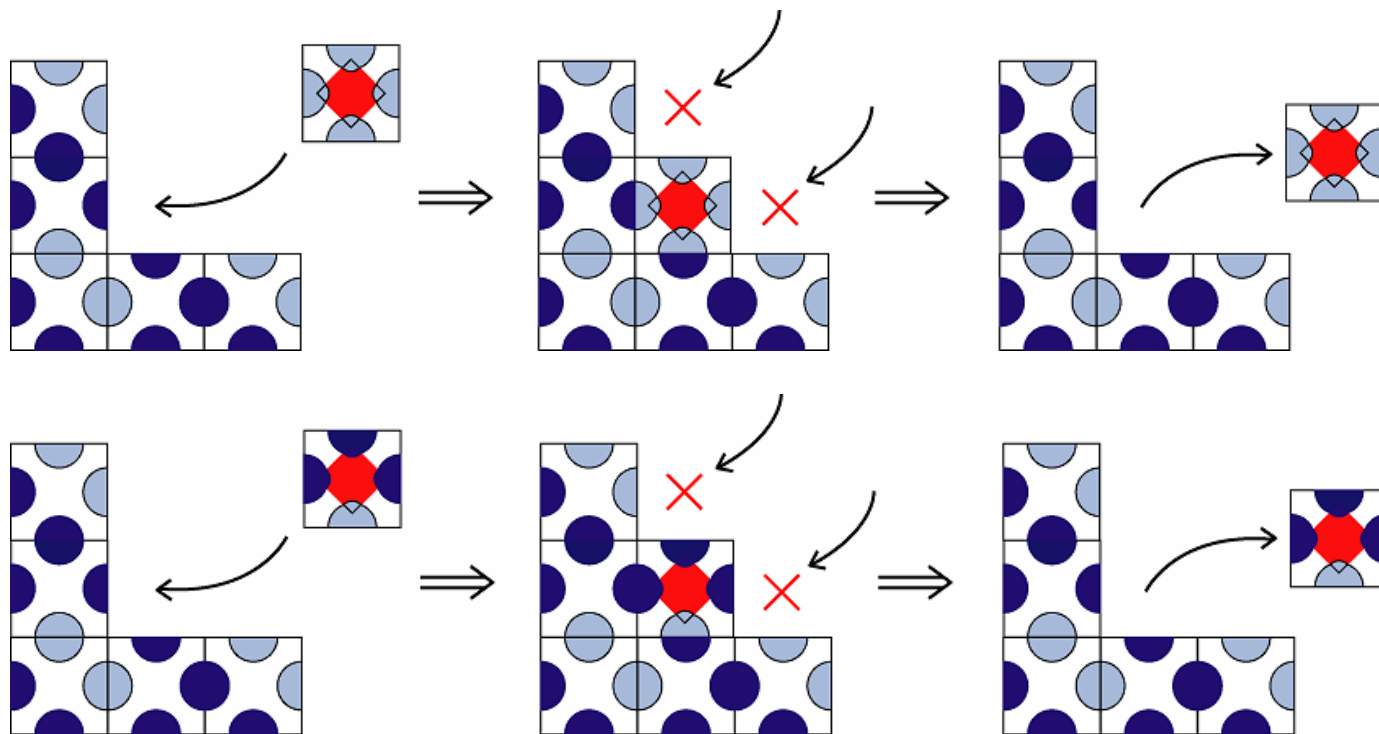
layered tiles



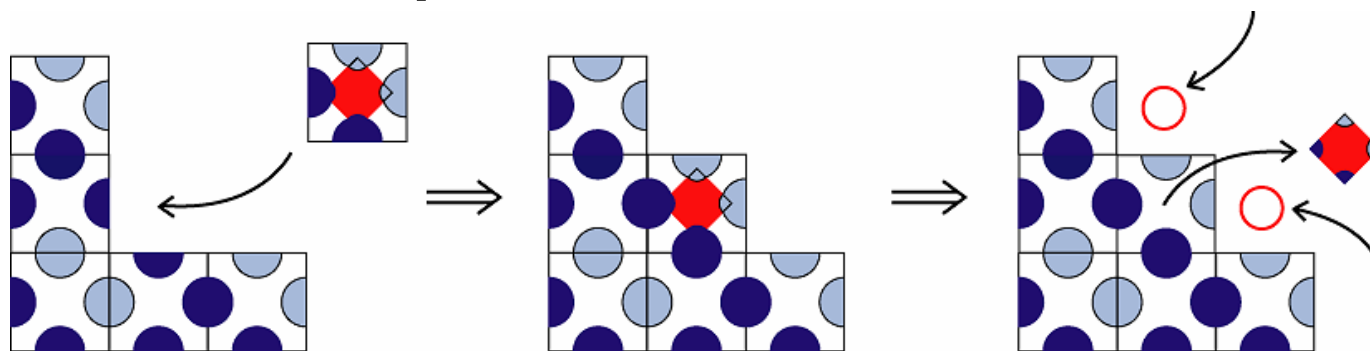
corner tile
(seed tile)

- original tileとprotective tileを重ねる

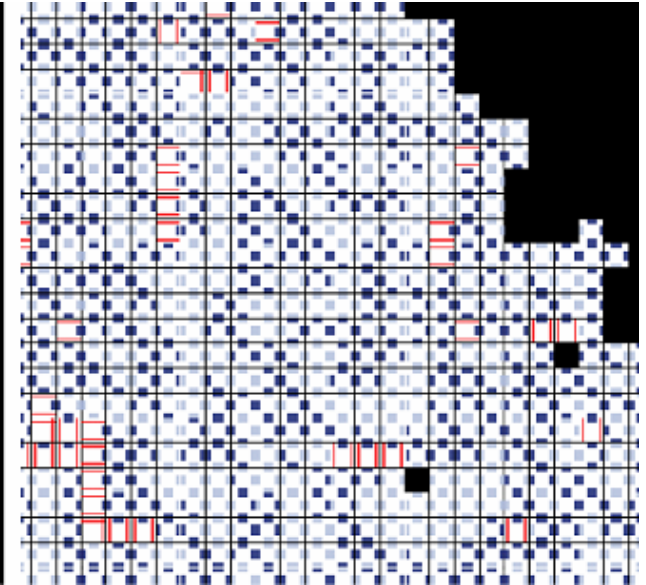
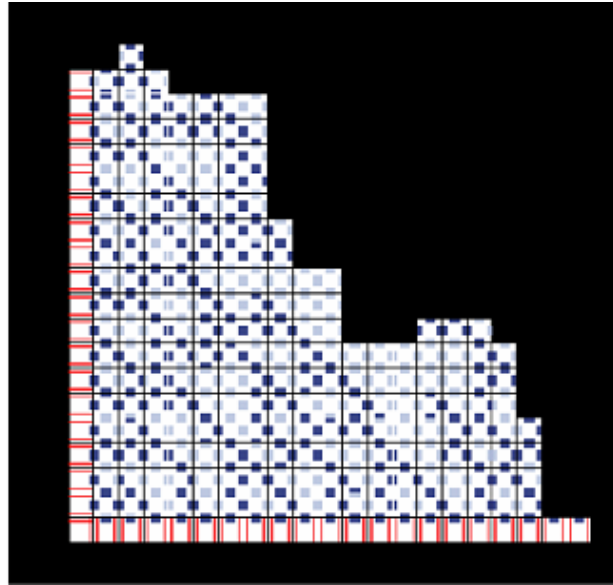
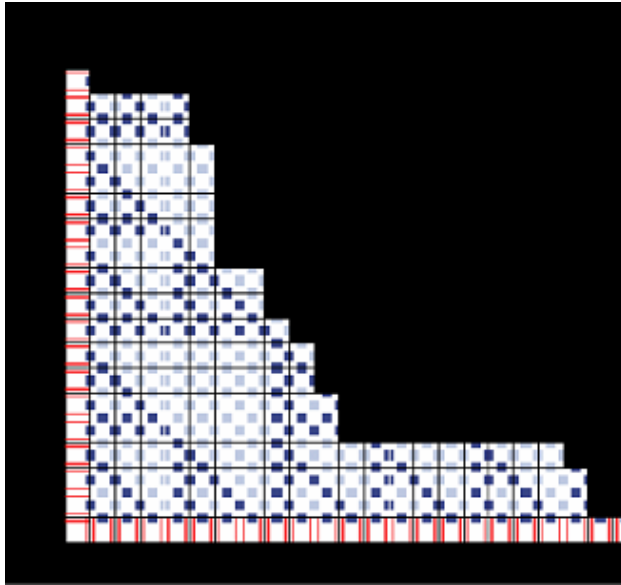
- 0, 1辺マッチの場合・・・二層のタイル(layered tile)ごと解離する



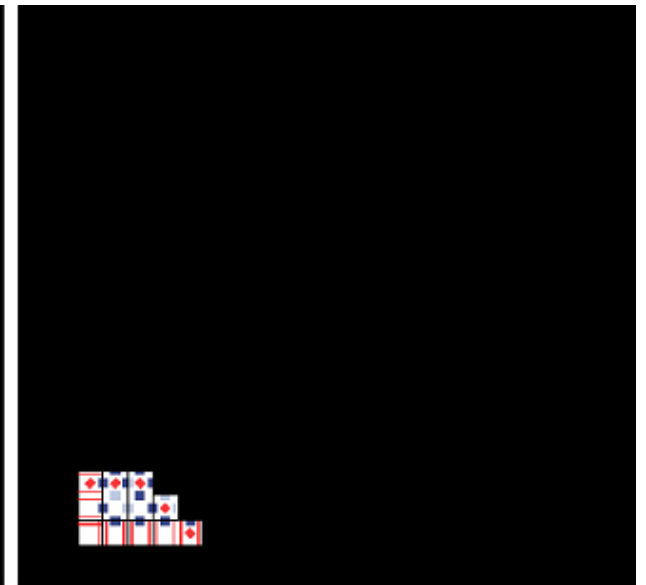
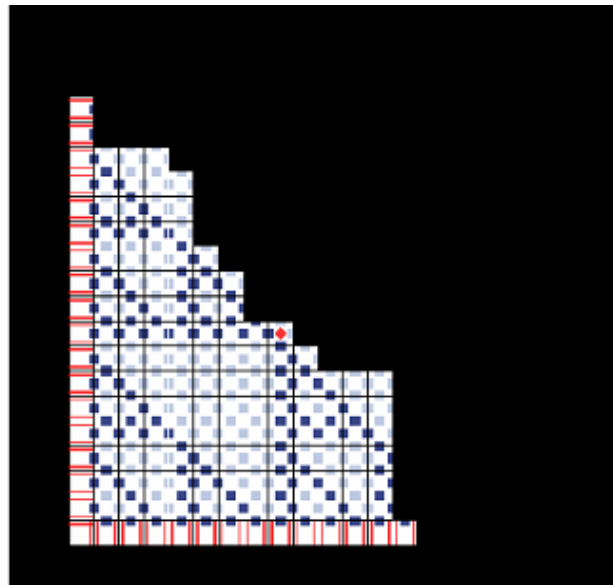
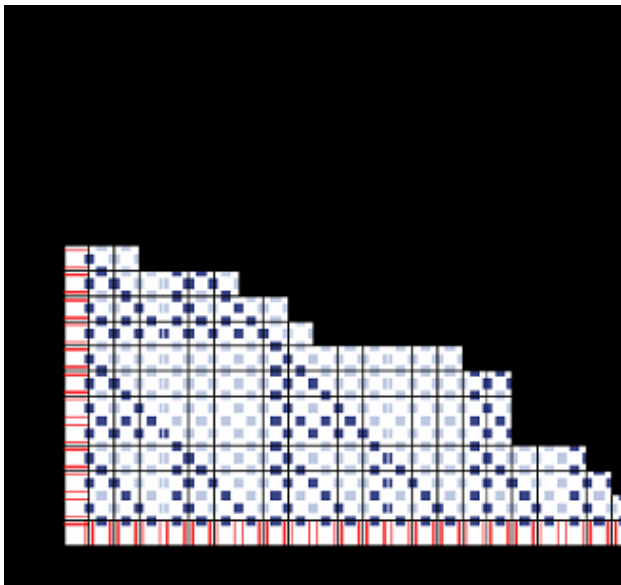
- 2辺マッチの場合・・・protective tileのみ解離する



OTM (Original Tile Model) …左から = 1.41, 1.12, 0.82



LTM (Layered Tile Model) …左から = 1.41, 1.12, 0.82



Layered Tile Modelの特長

Winfreeの

Proof Tile Model

- エラー率・成長速度の改善
 - $r_g = 10$ (2 x 2 PTMの10倍の性能)
- ロバスト性の向上
 - タイルの濃度と温度の変化に強い
- 少ない種類の塩基配列で実装が可能
 - オリジナルの約2倍の配列数, 配列も類似
- 不要なタイルの結合を防止できる
 - protective tileの導入により, モノマー同士の相互作用や集合体同士の相互作用を抑制できる
 - $0 < \dots < 1$ の領域において, 成長を抑制できる

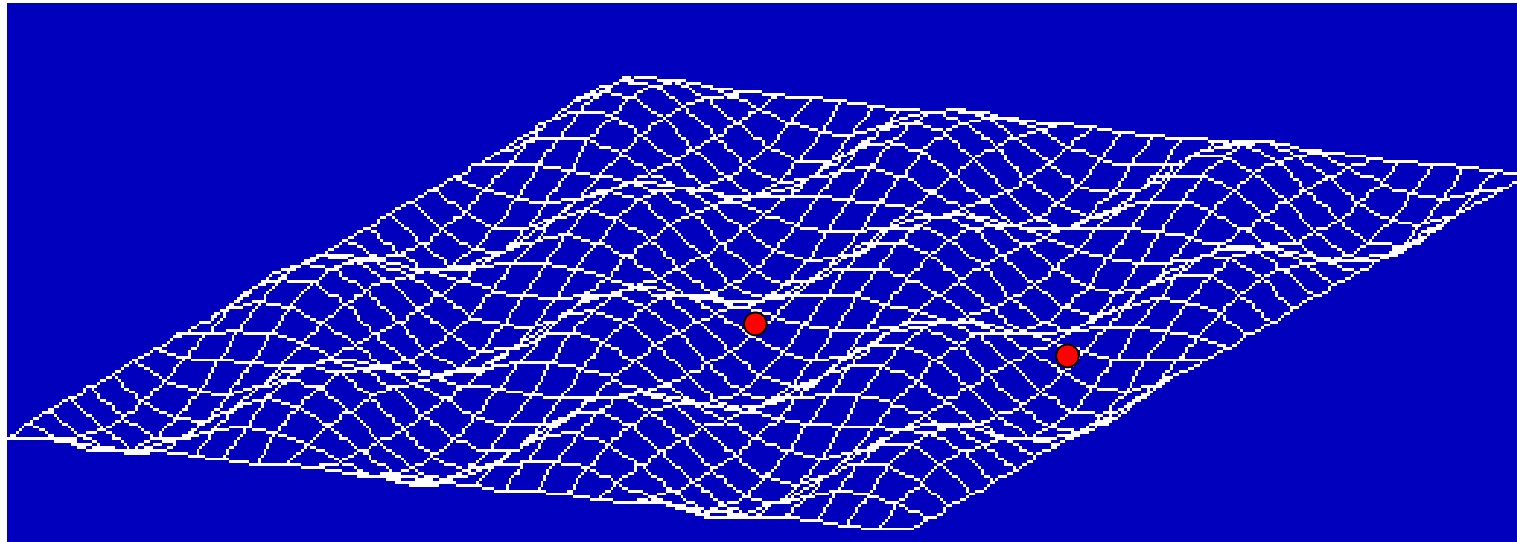
構造を変化させる設計

萩谷・小野(山下班)

- 二次構造の作るエネルギー地形の解析
 - 特に二つの二次構造間のエネルギー障壁
- 萩谷：最小流域アルゴリズム
 - 与えられた二つの構造を結ぶエネルギー最小流域
 - minimax path(エネルギー障壁を含む経路)を含む
 - ヘアピン・マシンの実験結果と比較
 - エネルギー障壁が求まらない場合もある。
 - 一般的に、配列が長くなると実行時間が指数的に増大
- 小野(山下班)：局所探索
 - 近似的にはあるが、エネルギー障壁をより効率的に計算するアルゴリズム

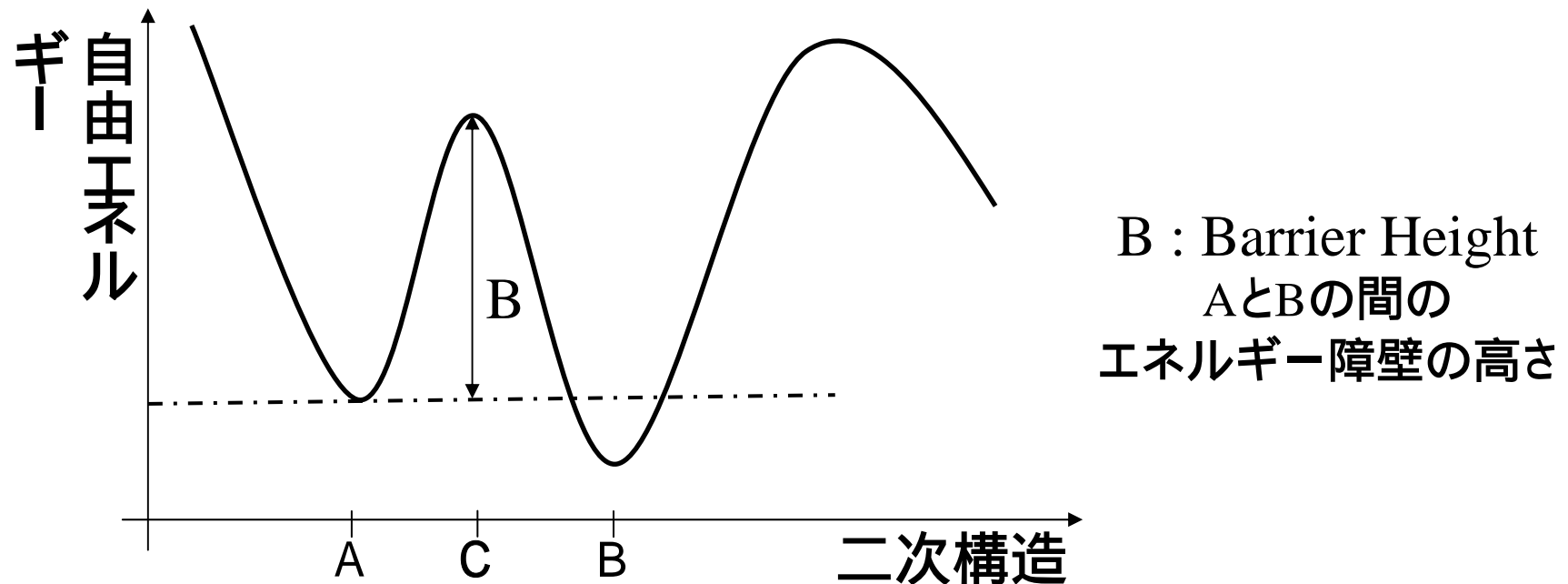
二次構造の作るエネルギー地形

- 地形グラフ $G(V, E, f)$
- 頂点(二次構造)の集合 V
- 接続関係 E (塩基対の形成/解離)
- エネルギーのマッピング $f: V \rightarrow \mathbf{R}$



エネルギー障壁

- DNA分子のある形態Aから、ある他の形態Bへの形態変化経路には、エネルギー障壁の形態Cがあると考えることができる。
- できるだけ低い障壁の経路で反応が進む確率が高いと予想される。



Minimum Basin Algorithm (最小流域アルゴリズム)

- 2点を含む最小の流域を求めるグラフ探索法。
- 始点から終点までのエネルギー地形(集合B)を、低い点を選択しながら探索・生成。(最小の流域)
- Prim-Dijkstra法的一种。

1. 始点・終点は与えられている。
2. 集合B、集合Nは空集合。(Basin, Neighbor)
3. 始点を集合Bに加える。
4. 集合Bに隣接する点を集合Nに加える。
5. 集合Nの最も低いエネルギーの要素を集合Bに加える。
(複数ある時は全てBに加える)
6. Step4に戻る。Step5で、加える点が終点であったら終了。

出力は集合B。その最大の要素が障壁になる。

集合Bの解析が形態変化経路予測として有効。

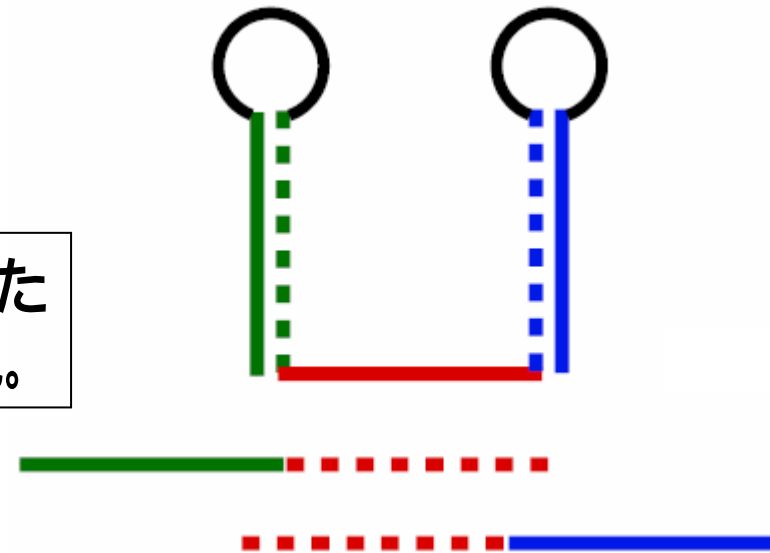
単純な実装では長い配列の解析が困難。

効率化

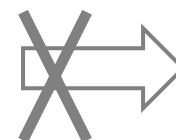
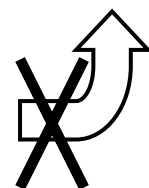
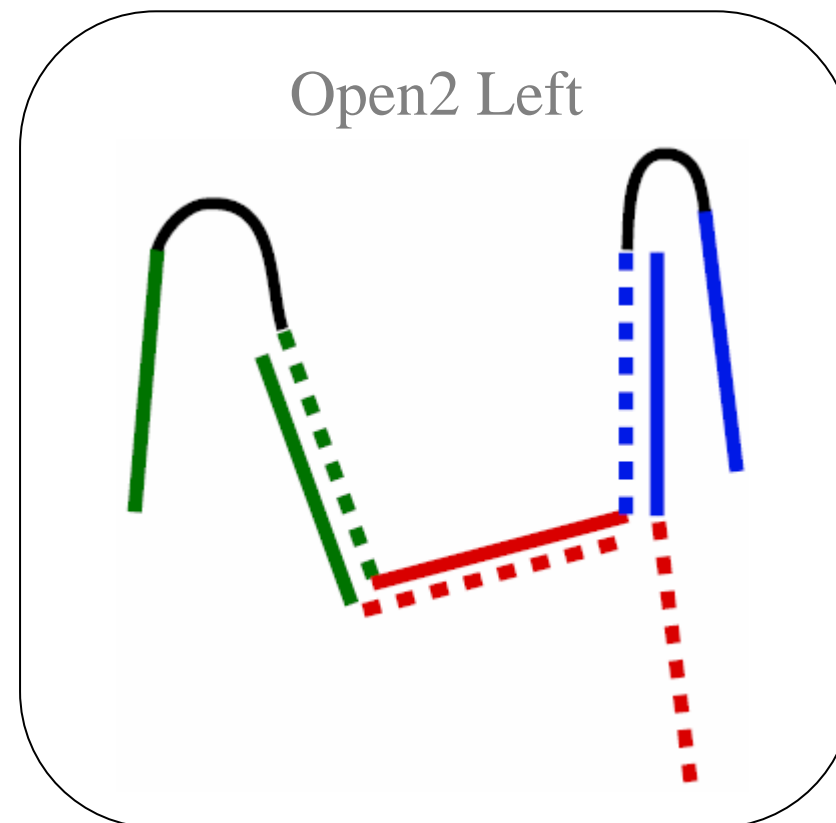
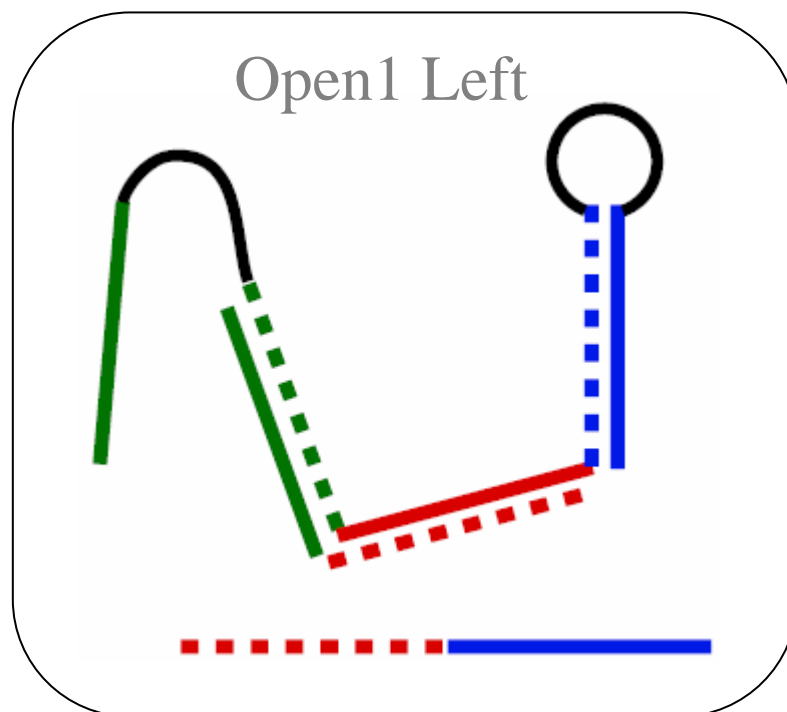
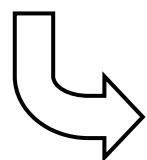
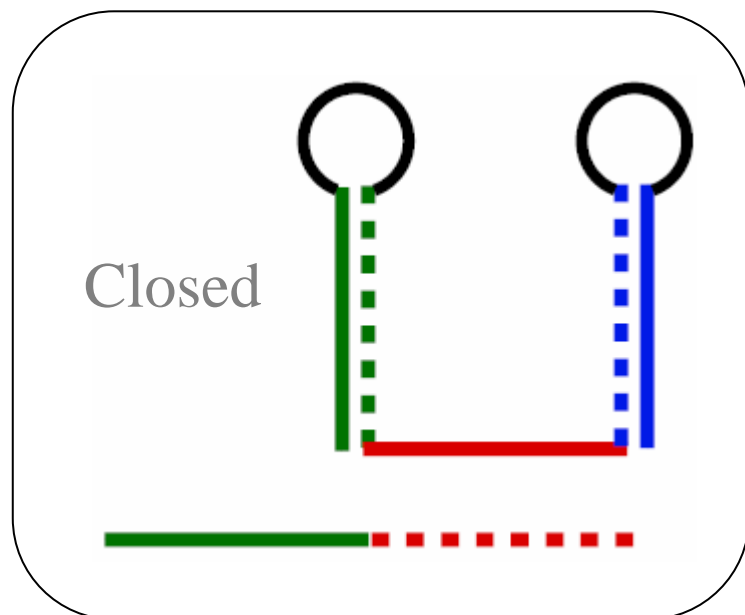
状態が分岐できる分子マシン

- 1分岐できるhairpin構造の配列を求める
- 実験で確かめる
- 計算との対応を見る

どちらかのヘアピンループが開いた形態(構造)で分岐した状態を表現。

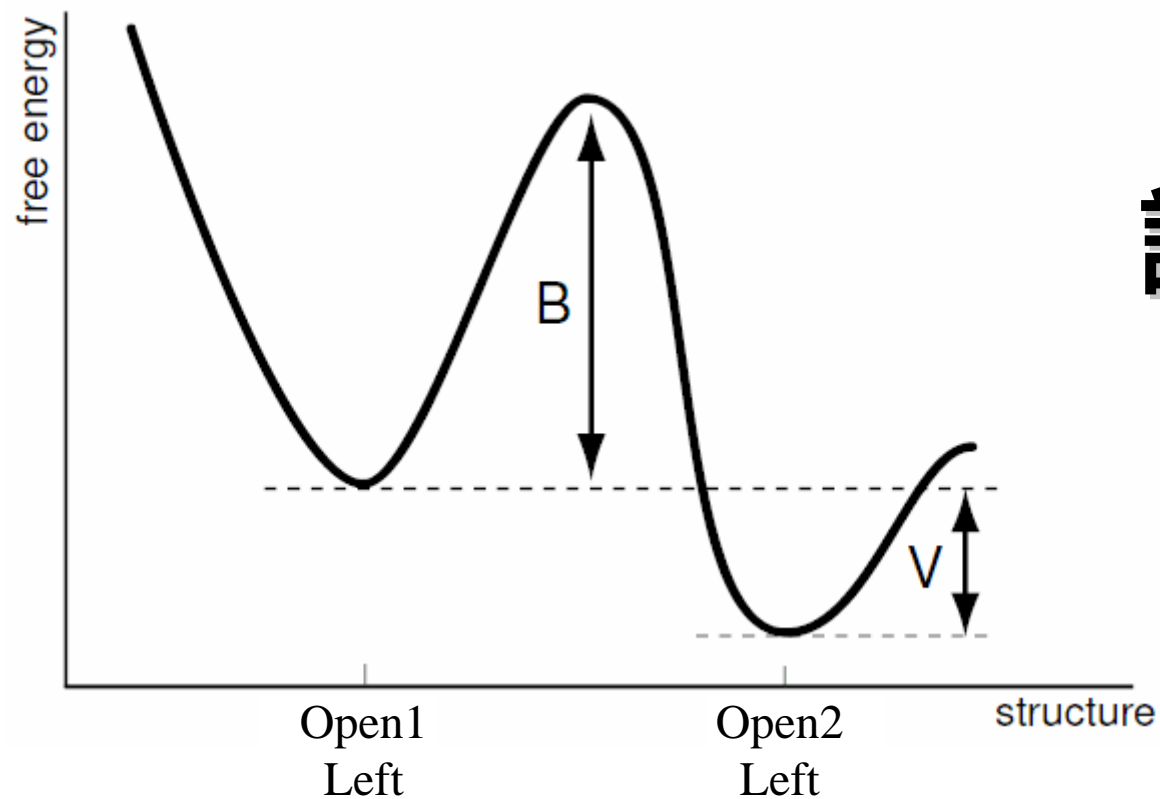
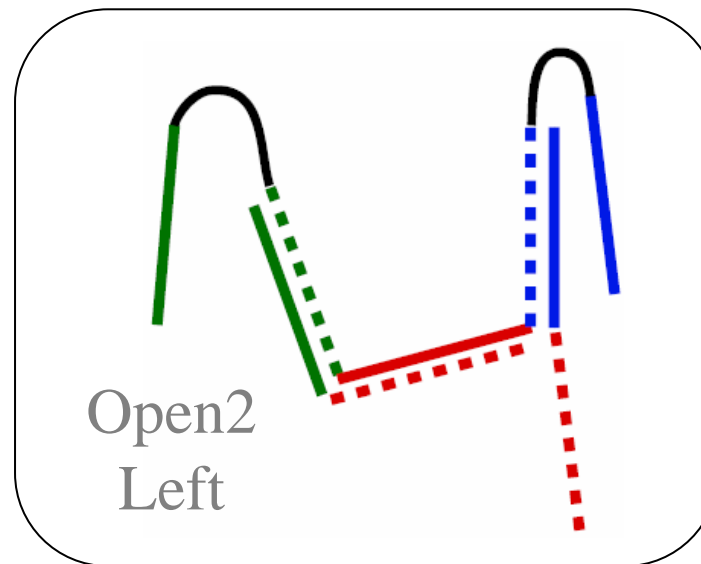
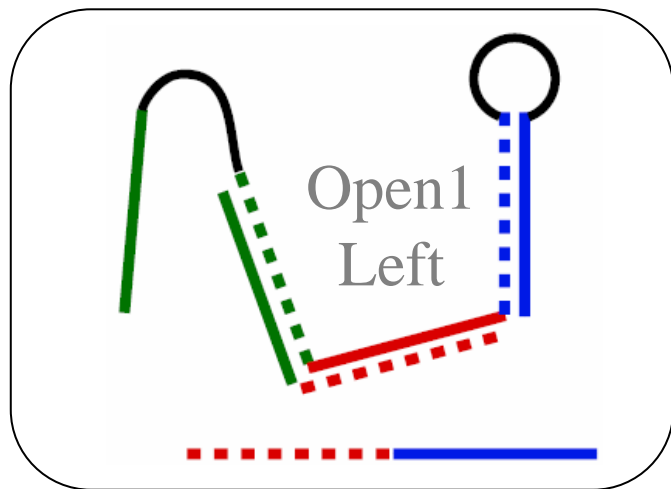


- H. Uejima and M. Hagiya:
“Secondary Structure Design of Multi-state DNA Machine Based on Sequential Structure Transition”, DNA9 Preliminary Proc., 92-96, (2003).
- A. Kameda, M. Yamamoto, H. Uejima, M. Hagiya, K. Sakamoto, A. Ohuchi:
“Conformational addressing using the hairpin structure of single-strand DNA”, DNA9 Preliminary Proc., 197-201, (2003).



(mfe構造 etc)

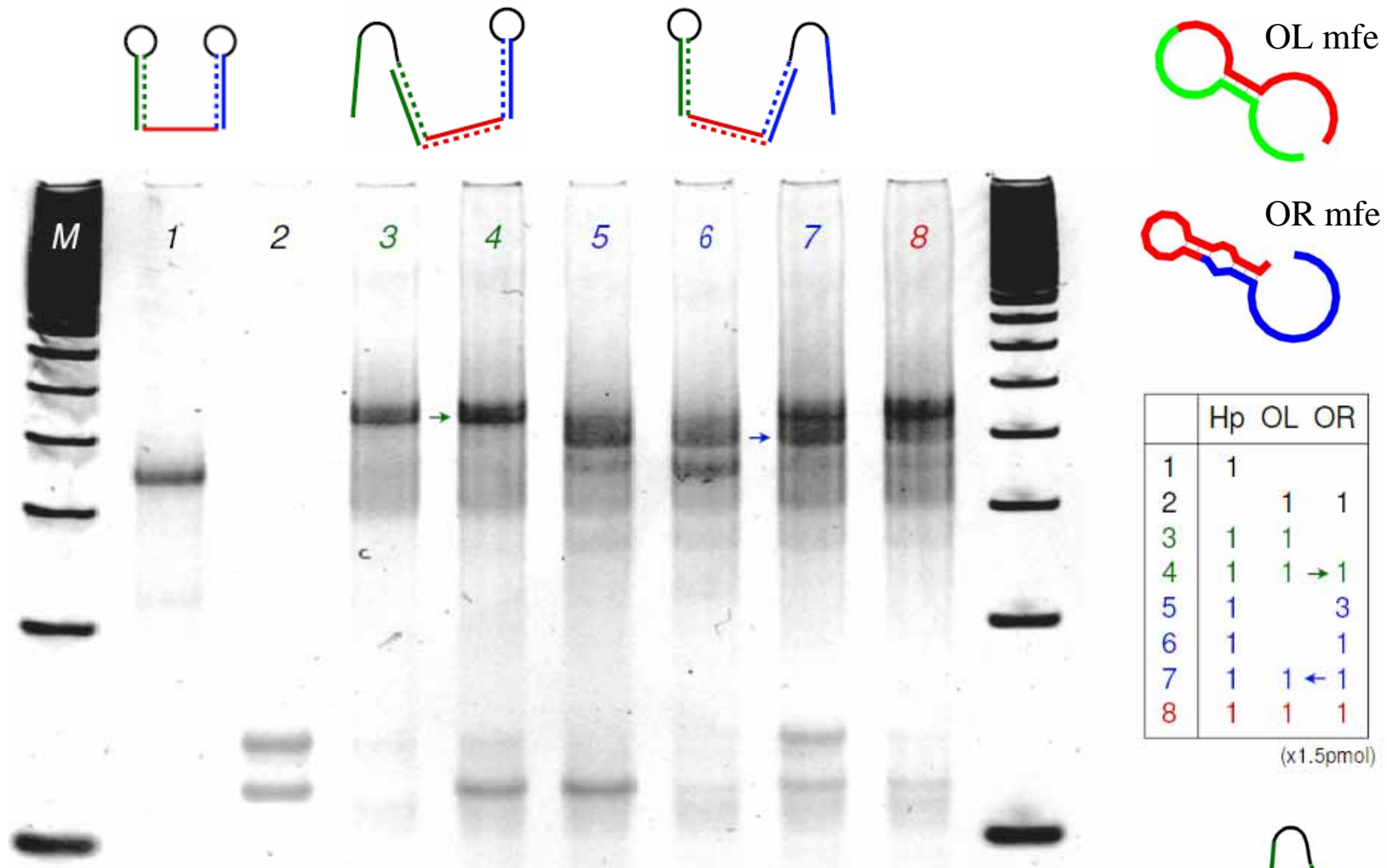
目標とする動作



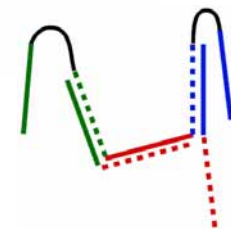
設計の基準

- maximize B
- minimize V

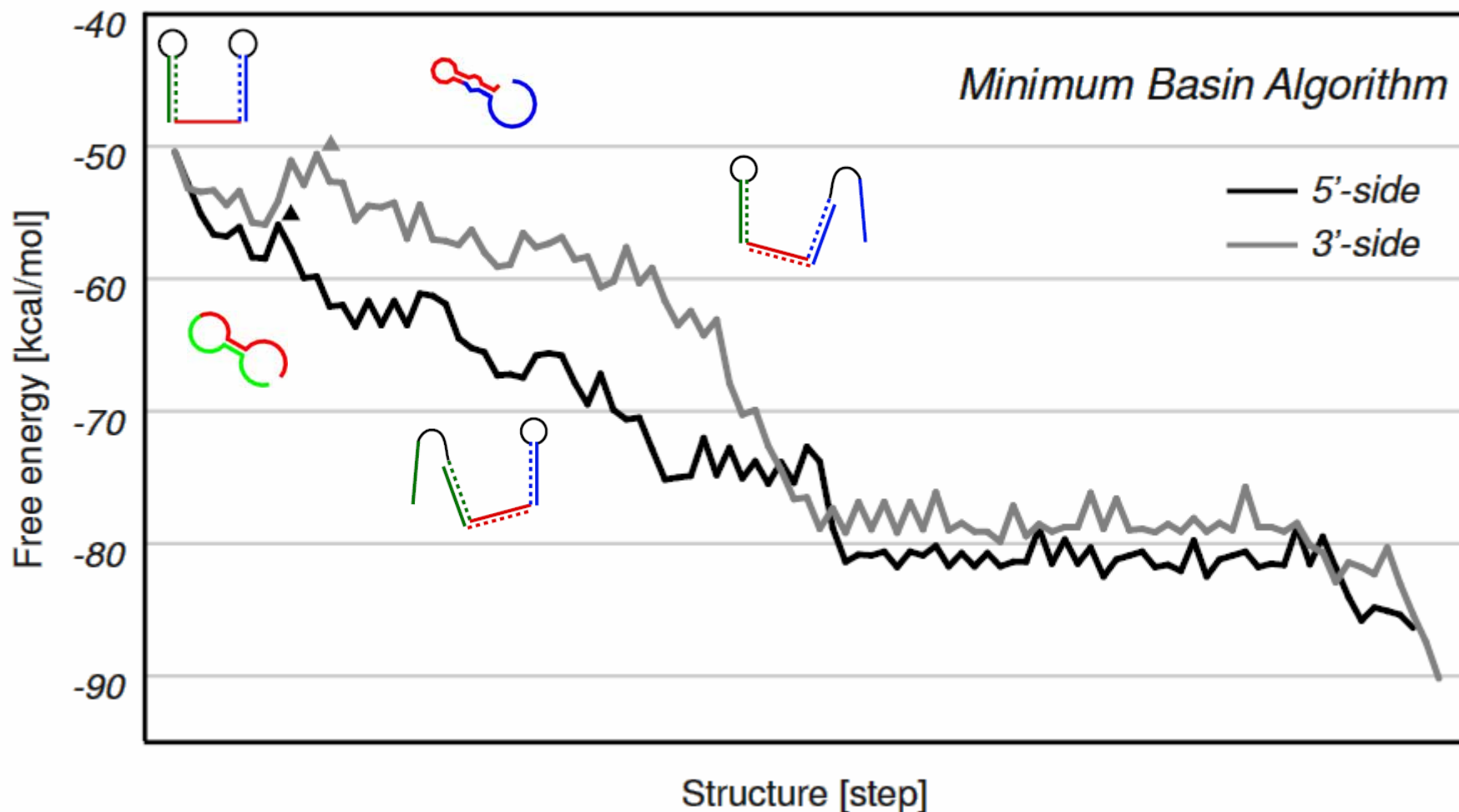
実験結果 と 想定外の構造



入力DNAの順序によって異なる形態をとる



最小流域法による経路 (minimax path)



障壁だけでなく経路についての議論ができる。

8 ~ 9sec. (CPU time), 200MB (memory), x86 500MHz, Windows2000

エネルギー障壁見積もり問題の特徴

- 構造数が莫大
 - 一般には配列長の指数サイズの構造数
 - 反応経路の候補自体が指数サイズ
- 構造のエネルギー値は高速に計算可能
 - ループ, スタックなどの部分構造から定まるエネルギー値
 - 塩基配列から直接障壁を求めるのは難しい(?)

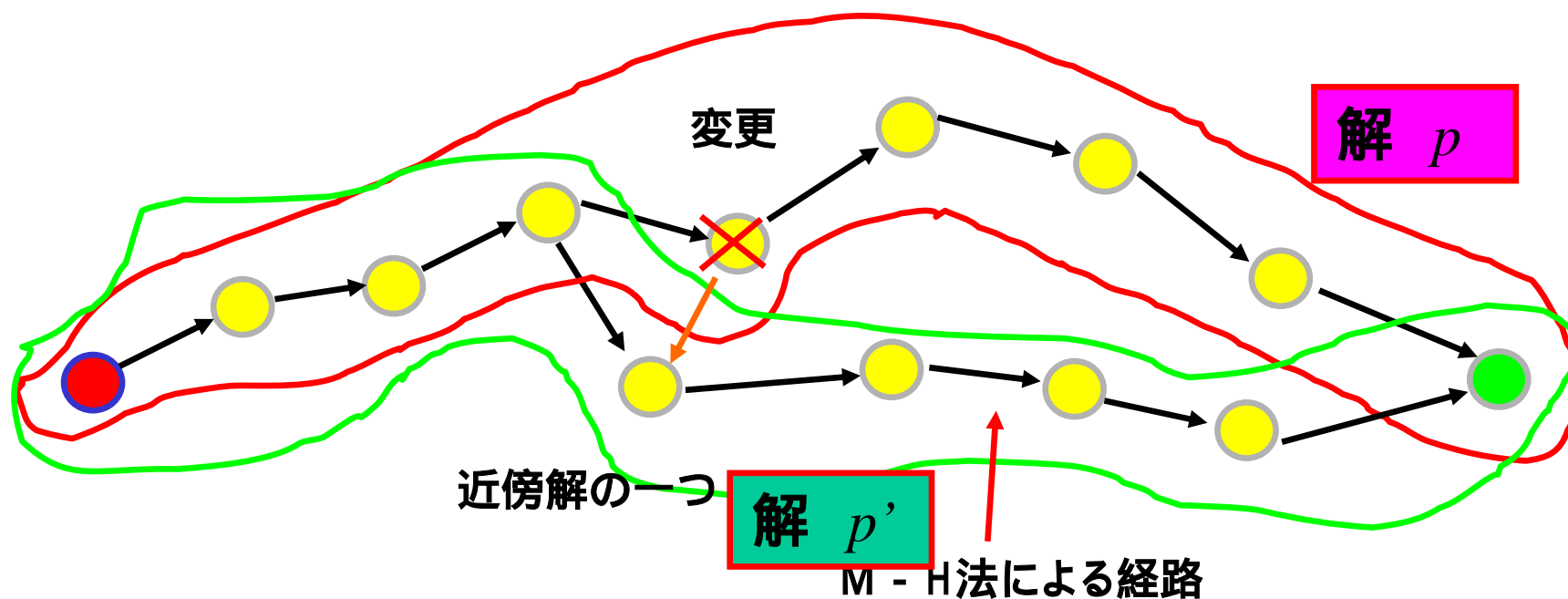
従来法

- Morgan-Higgs法 (MH法)
 - 厳密アルゴリズムではない
 - なるべく多くの塩基対を保つ経路を見つける
 - 経路長は最短
 - 高速
 - 解精度が不安定
- Flooding型 (最小流域法など)
 - 厳密アルゴリズム
 - エネルギー値を水位に見立てて流れるまで水位を上げるイメージ
 - 常に正しい解を返す
 - 計算時間が不安定 (問題によっては計算時間が爆発)

局所探索

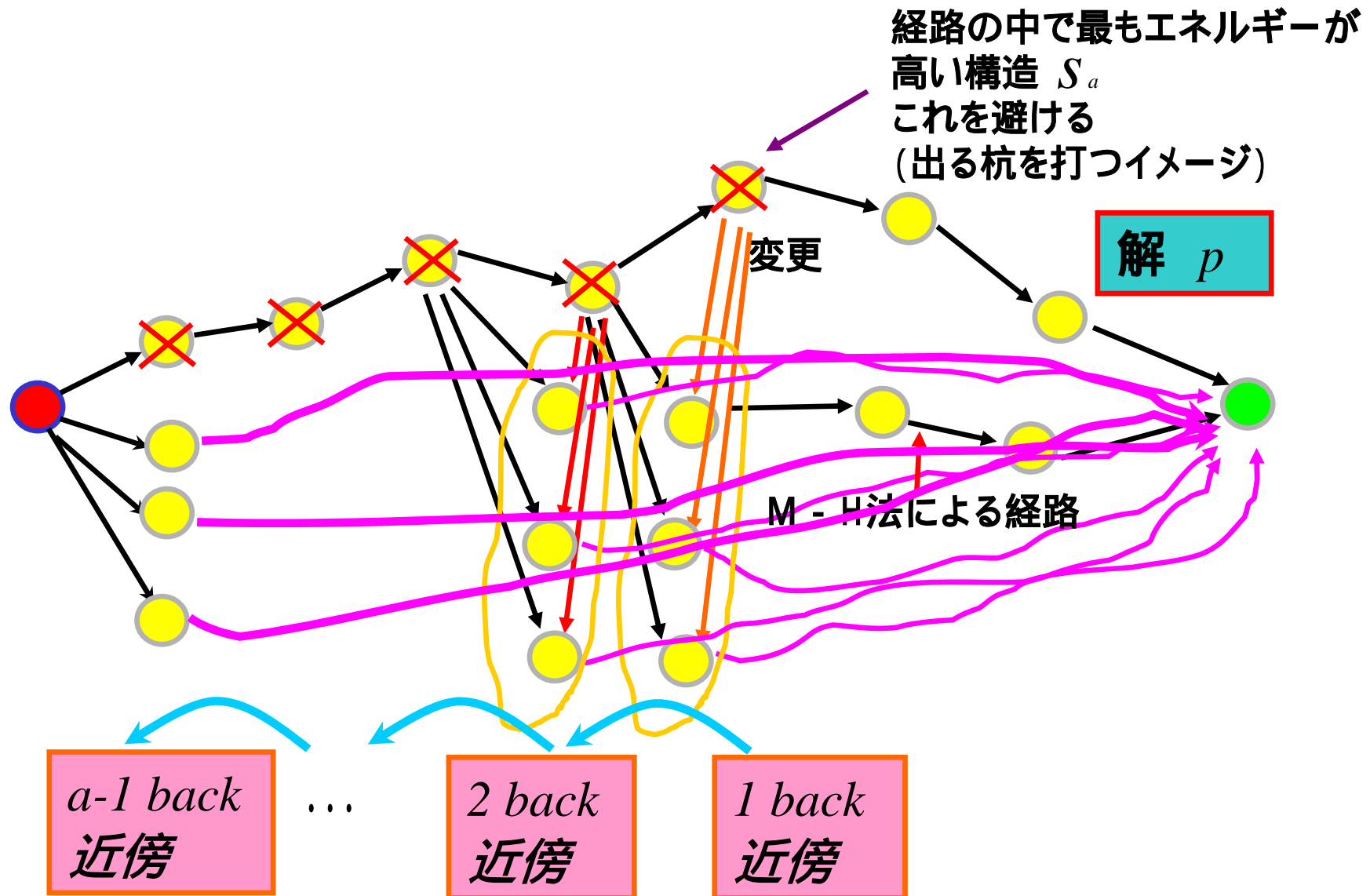
- より「ロバストな」アルゴリズムを
 - 現実的な計算時間で悪くない精度の解を見つける
- 反応障壁を見つける近似アルゴリズムの提案
 - NP困難な組合せ最適化との類似性
 - 局所探索に基づくアルゴリズム
- 局所探索の設計
 1. 初期解(反応経路)を生成する
 - MH法, Greedy法
 2. 近傍(経路の集合)を探索する
 - MH経路による近傍の定義
 - 1-back 近傍, k-back 近傍, Full-back近傍
 3. 改善解がなければ終了, 解(経路)を出力

近傍解



反応経路内のある構造を別の構造に変更することにより
新たな反応経路を生成

探索空間

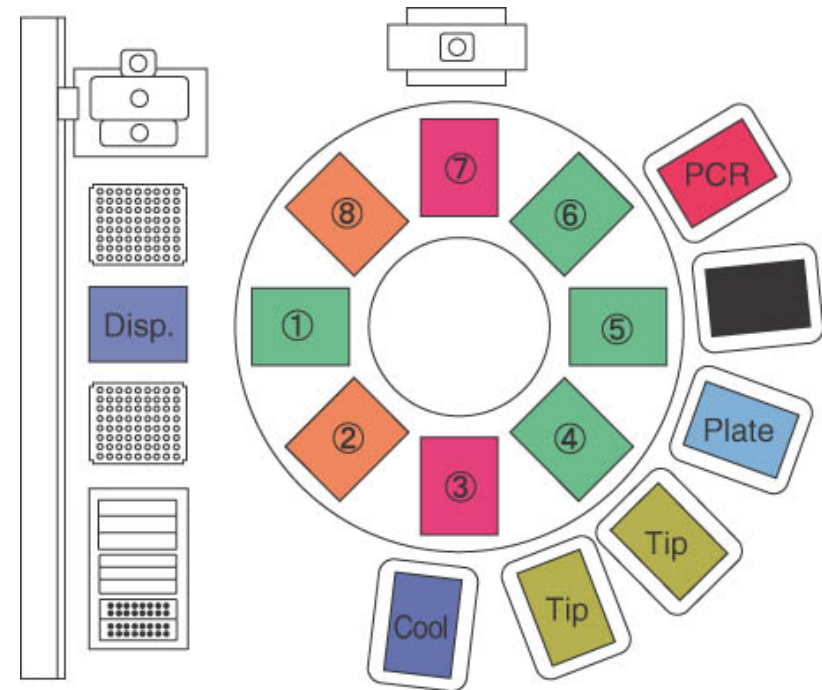


実験結果

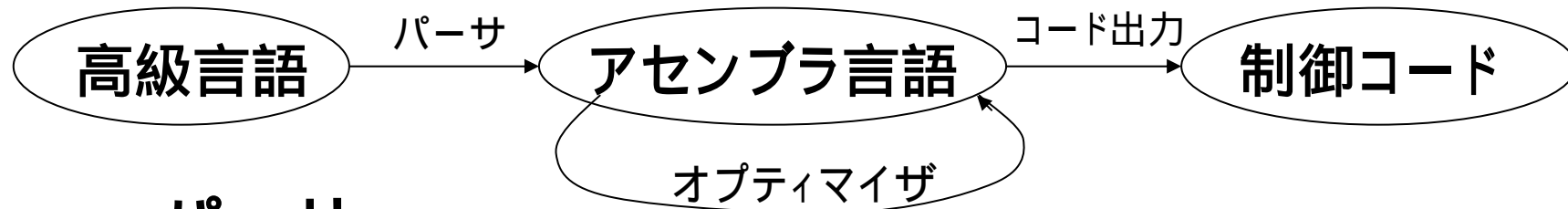
- 厳密解法との比較
 - とにかく解が出る (厳密解法はそうではない)
 - 精度面
 - 最適値を出力できているものも多い
 - 計算時間
 - 現実的な計算時間 (大体的場合, かなり早い)
 - 厳密解法よりも遅い場合もある
 - 探索する近傍サイズの大きさが原因
- 初期解生成法による違い (MHかGreedy)
 - 解精度
 - 大体においてMH法の方がよい性能
 - 計算時間
 - 大体においてMH法の方がよい性能
- 近傍による違い
 - k-back近傍が最も妥当な性能

反応の並列化 萩谷

- 陶山(萩谷班)のハイブリッド型DNAコンピュータ
- 4つの並列モジュール
 - IMU (=Integrated Magtration Unit)
 - ディスペンサ
 - コンベイヤ
 - ストッカ
- 自動化への要望
 - 具体的なテーブル番号を意識したくない。
 - 必要のないプレートやチップは自動的に廃棄して欲しい。
 - 温度制御などに関して、時間的な制約を意識したくない。
 - できる限りモジュールの並列性を活かしたい。



DNA コンパイラの構成



- パーサ

- 高級言語プログラムの意味的な正しさを検証し、正しいが逐次的なアセンブラコードを出力。
ここでは、テーブルや IMU といったリソースは無限に存在すると仮定。

- オプティマイザ

- パーサの出力したアセンブラコードに対して、テーブル等の有限なリソースの割り当てと、命令スケジューリングを行い、並列化された最適なアセンブラコードに変換。

- コード出力

- アセンブラ言語を実際の制御コードに変換。

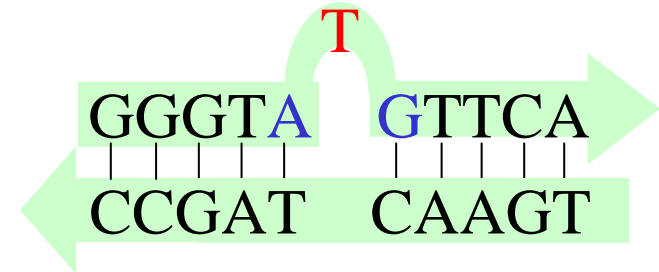
オプティマイザ

- 従来のコンパイラとの類似性
 - テーブルの割り当て = レジスタの割り当て
 - 制御命令のスケジューリング =
パイプラインのための命令スケジューリング
- 本研究
 - DNAコンピュータにも応用できるような汎用的なマシン記述言語
 - 整数線形計画法を用いて、テーブル等のリソースの割り当てと命令スケジューリングを同時に最適化
 - 特許申請中

反応の精密化

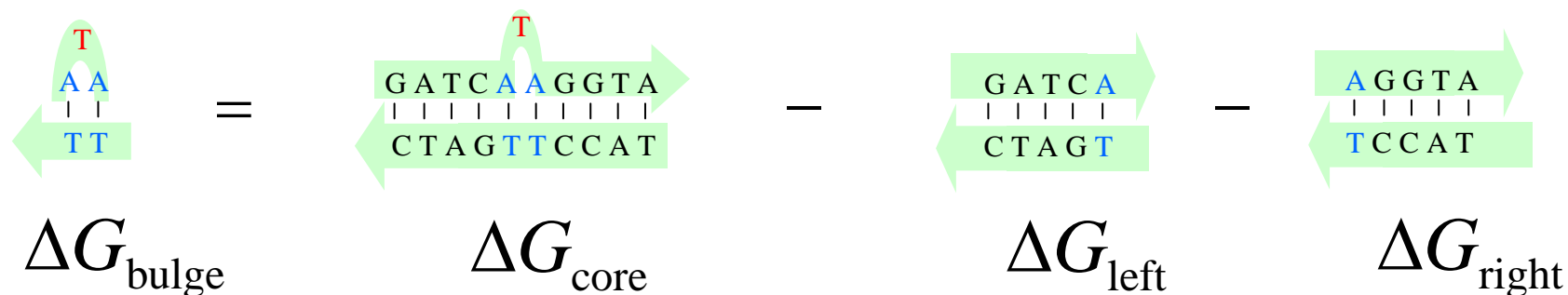
大内

- ハイブリダイゼーション反応のより精密な設計
- 一塩基バルジループ構造の安定性について検討



- Nearest-Neighborモデルの拡張
 - 自由エネルギーを実験データ(融解曲線)に基づいて算出
 - 64種類の部分配列のすべてについて実験
- バルジループ構造をとる可能性のあるDNA塩基配列の融解温度の予測をより精密に行うことができるようになった。

バルジループの G 算出手順

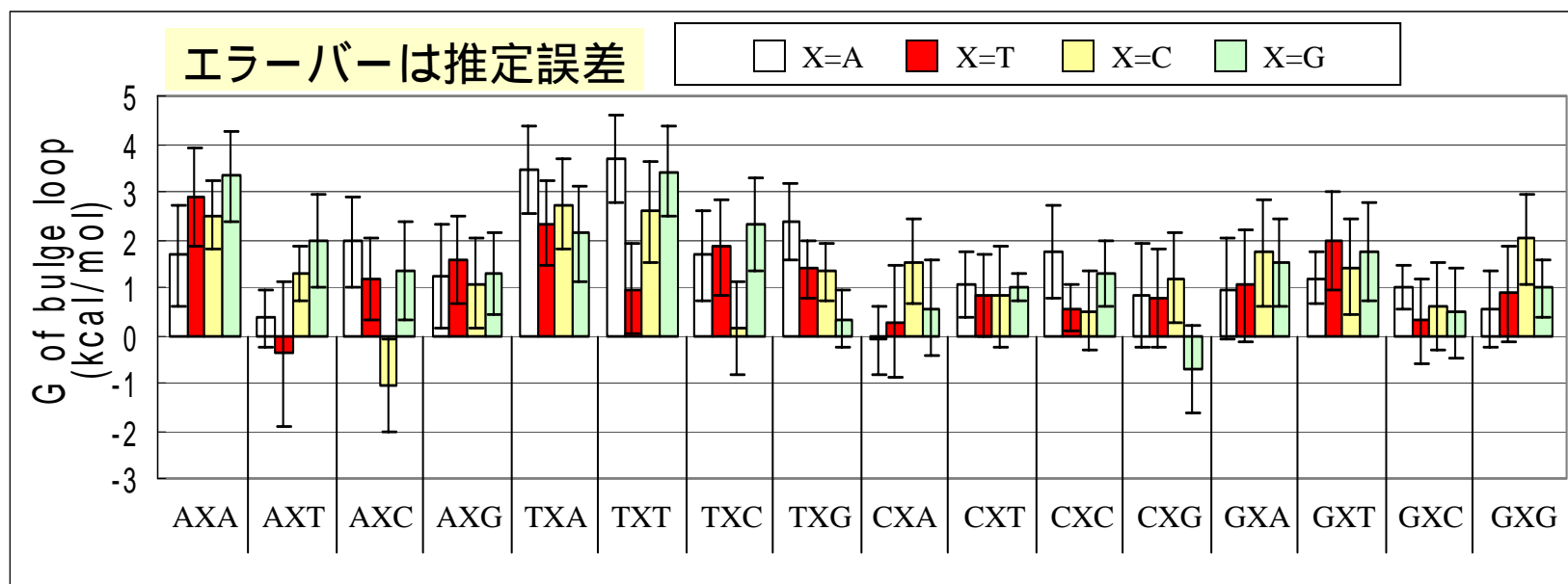
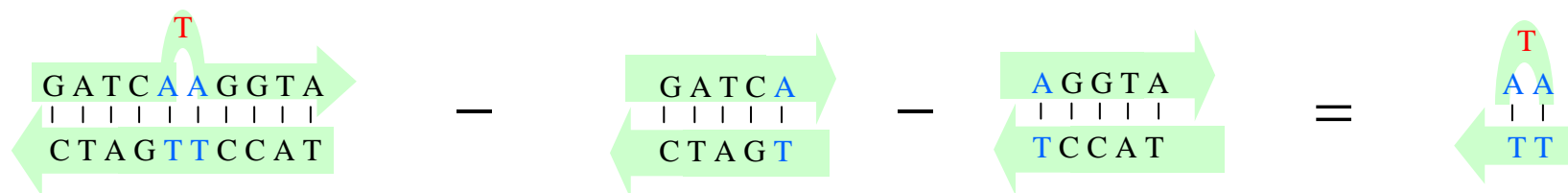


1. バルジループを含む配列の ΔG_{core} を融解曲線より算出
2. スタック対の NN パラメータを算出 $\Delta G_{\text{left}}, \Delta G_{\text{right}}$ の算出

AA/TT	-1.04	TA/AT	-0.60	CG/GC	-2.25
AT/TA	-0.91	TC/AG	-1.35	GC/CG	-2.28
AC/TG	-1.49	TG/AC	-1.51	init_AT	1.19
AG/TC	-1.33	CC/GG	-1.90	init_GC	1.12

3. ΔG_{bulge} を算出

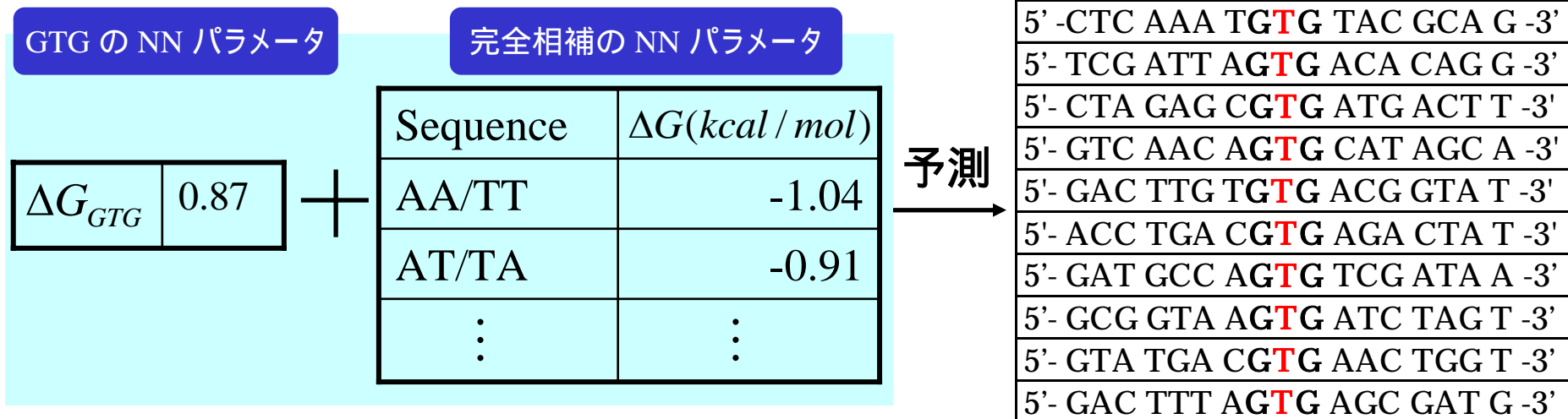
一塩基バルジループの NN パラメータ



最も不安定なものは TAT bulge (3.69 kcal/mol)

最も安定なものは ACC bulge (-1.05 kcal/mol)

GTG-bulge の NN パラメータの評価



予測誤差

	GTG-bulge	完全相補	
		Hatim 97	Sugimoto 96
G の予測誤差	3.3 %	3.9 %	4.8 %



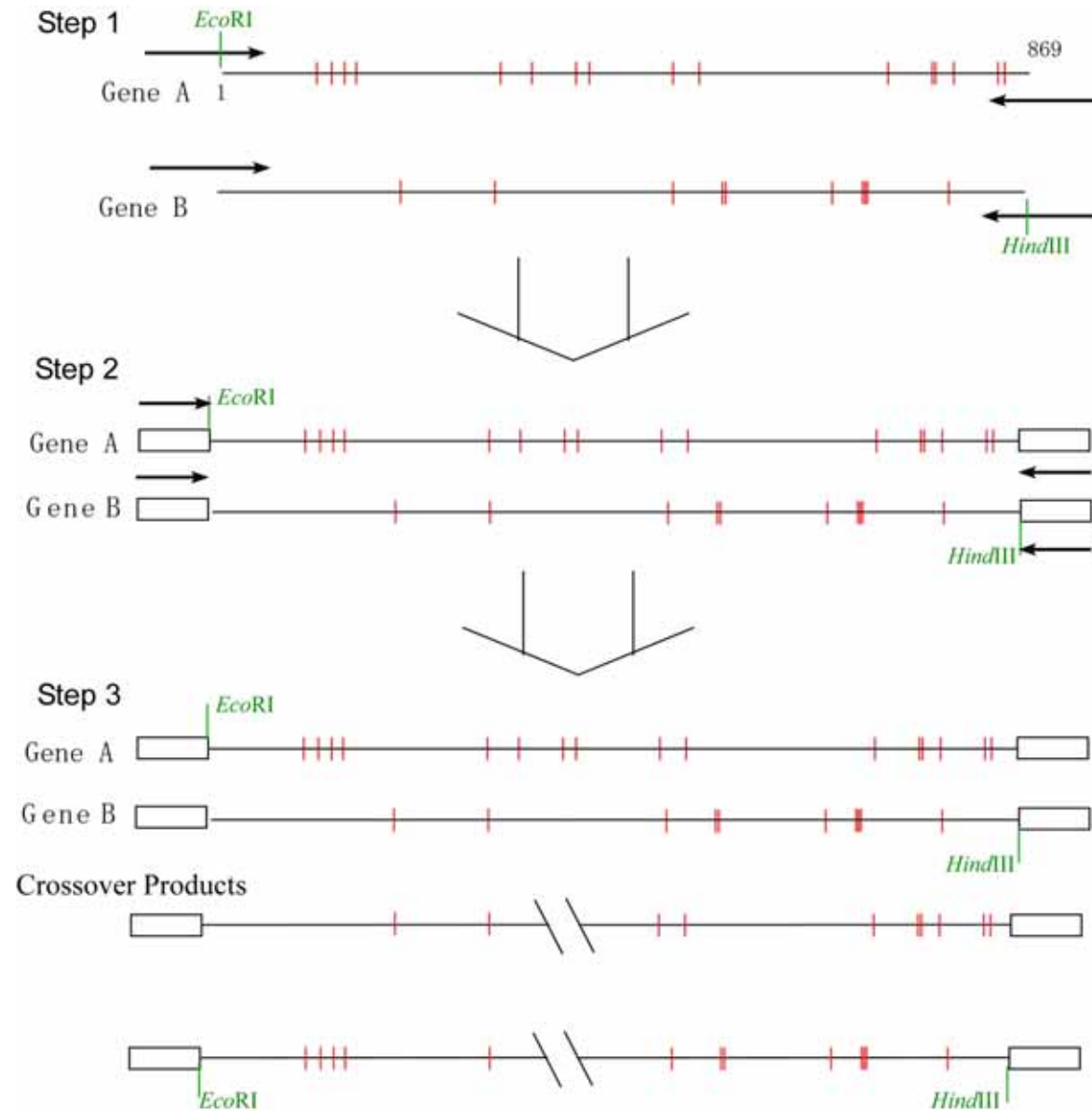
バルジグループを含む配列の ΔG は NN モデルで適切に予測できる

Biochemistry, Vol. 43, No. 22, pp.7143-7150, 2004 に掲載

進化のための反応 山村

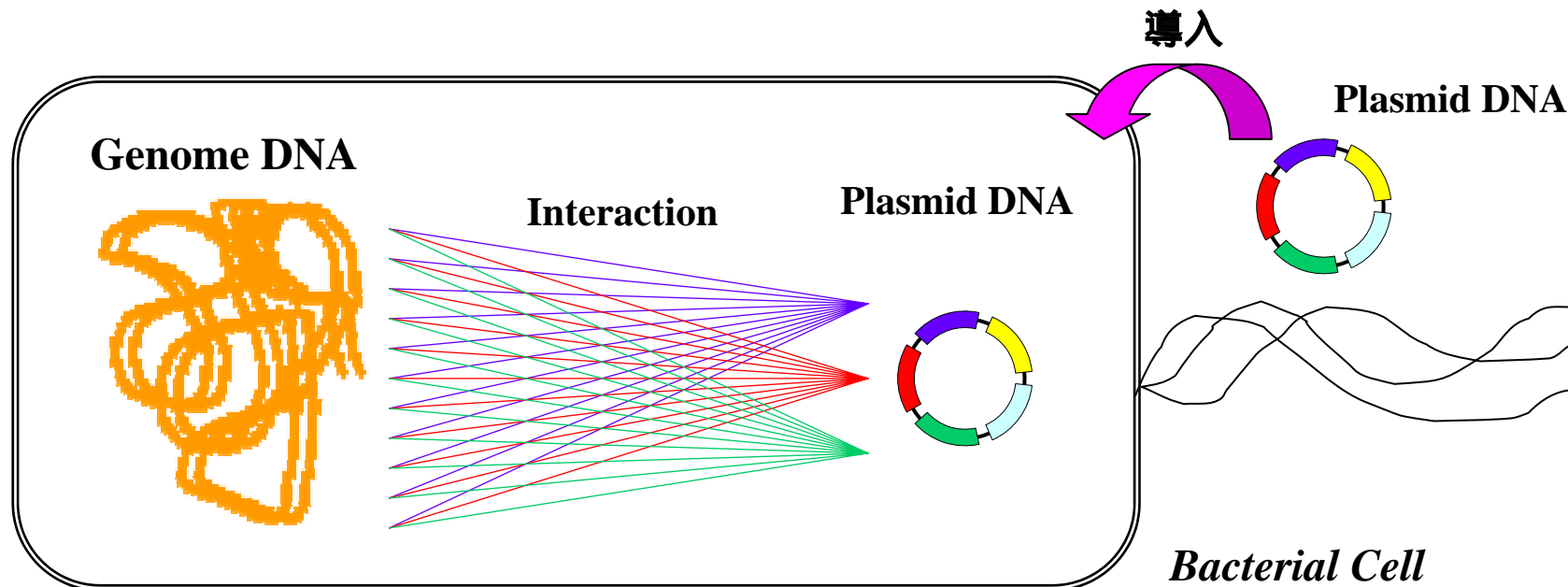
- ウェットGAによるチロシルtRNA合成酵素の基質改変
 - 困難な適応度地形を持つたんぱく質の進化
 - 計算機上のGAによる困難な最適化の知見を移転。
 - 点突然変異 + 一点交叉 + 熱力学的世代交代。
 - テストベッド = アミノアシル化tRNA合成酵素
 - 最古のタンパクのひとつ
 - 多くのローカルミニマ
 - シミュレーションによる困難な地形での動作確認。
 - 一点交叉の分子生物学実験上の実現 (次のスライド)
- ウェットGAによる細胞内ネットワークの進化

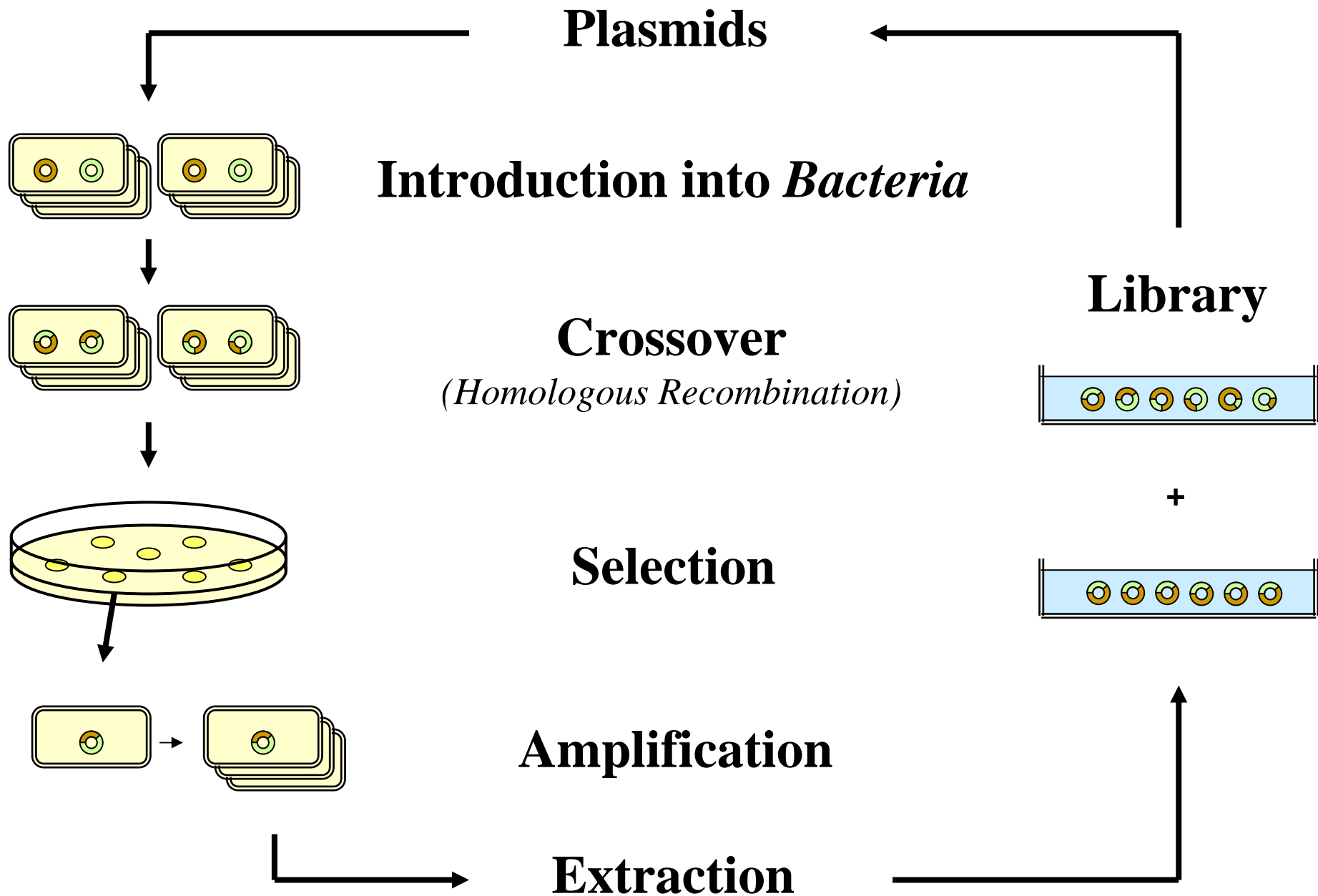
一点交叉の分子生物学実験上の実現



ウェットGAによる細胞内ネットワークの進化

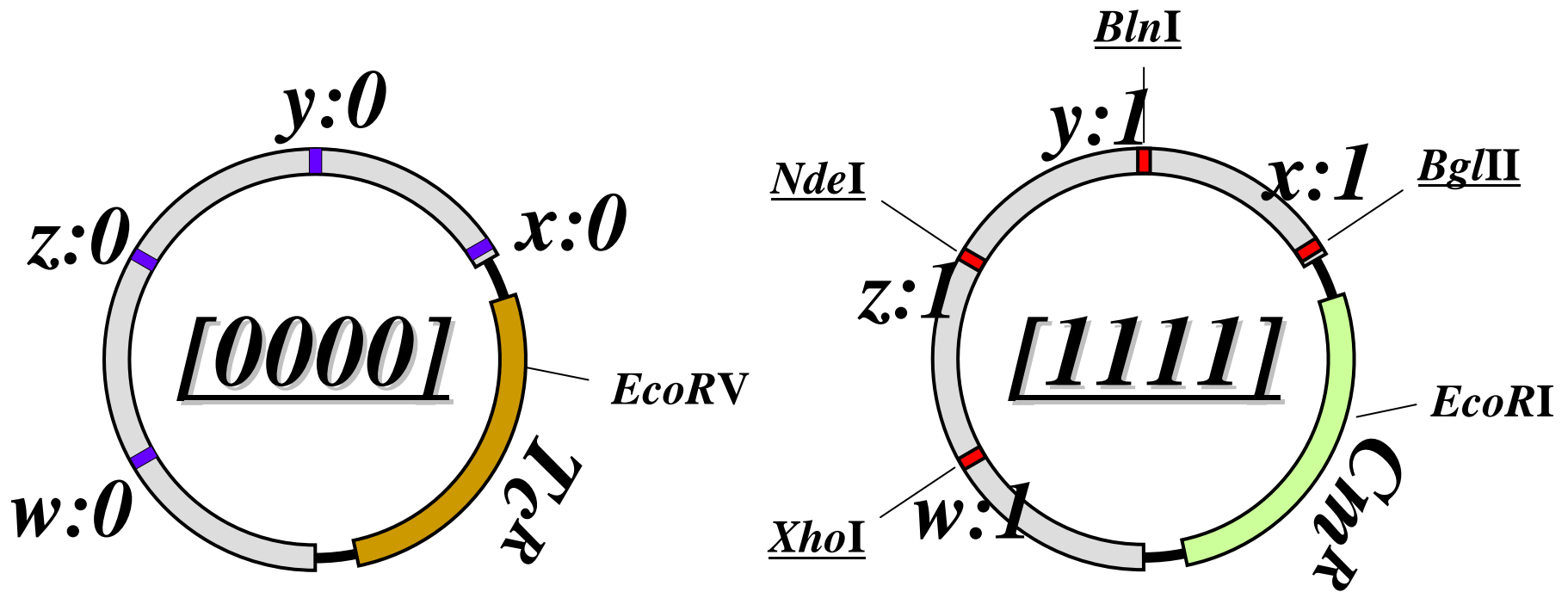
- プラスミド上の遺伝子の(発現量)最適化
 - 様々な発現プロファイルを持つプラスミドを並列に細胞導入し、設定した評価基準をもとに進化計算を行う。
 - 最適なプロファイルを持つプラスミドを進化的に獲得。
- 予備実験
 - 4遺伝子からなるプラスミドのin Vivoでの交叉を実現。



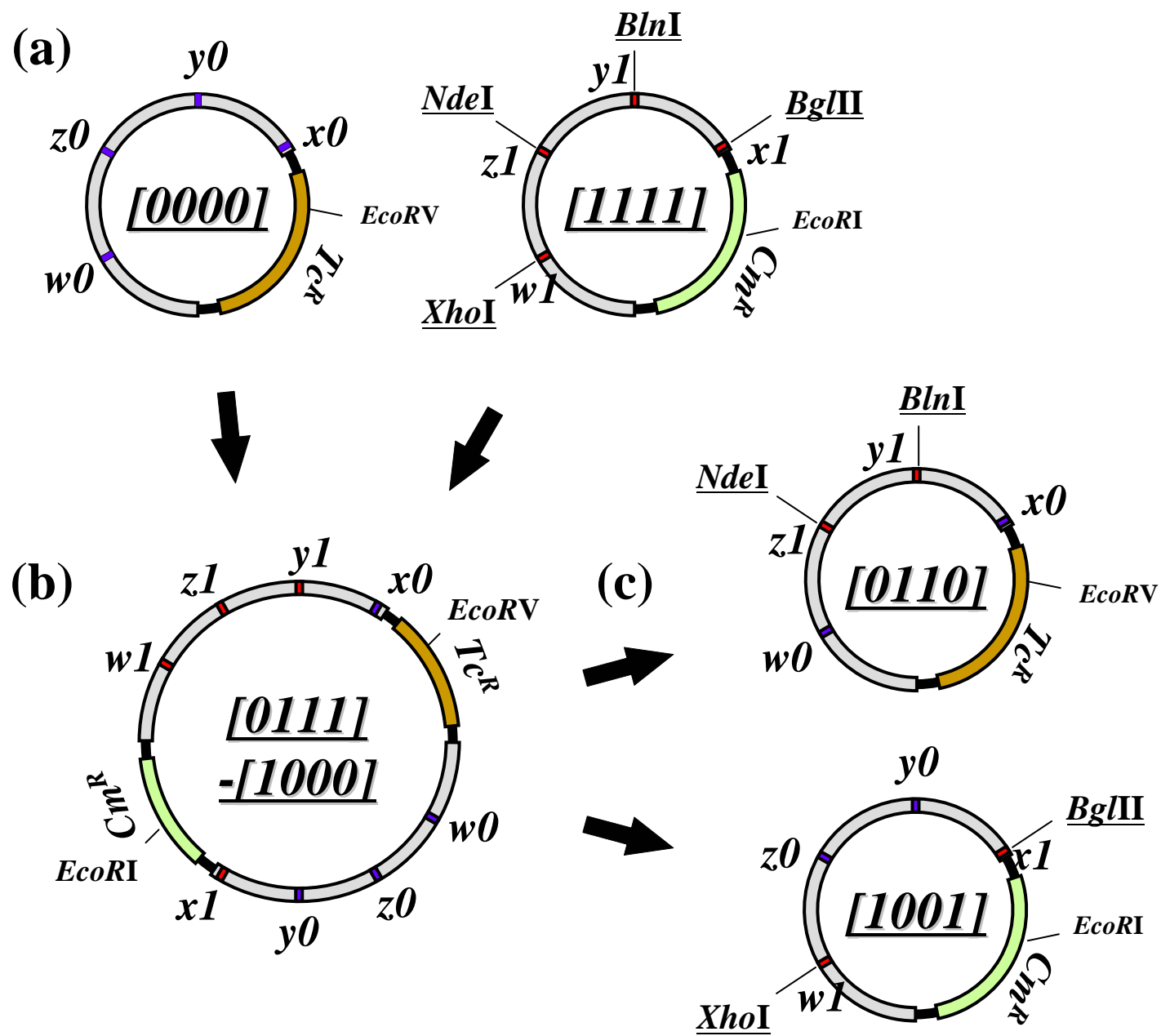


予備実験：4ビットプラスミド

「交叉」の実行可能性を調べるために、4ビットのプラスミドを作製した。



ビットの検出方法：制限酵素 (0:切れない / 1:切れる)



Crossover

Plasmids (pBR-0000, pBR-1111)

introduction

E. coli ES1301

over night incubation (+Tc, Cm)

Extraction

introduction

E. coli ES1301

over night incubation (+Tc, Cm)

Extraction

Detection

Plasmids (after crossover)

enzymatic digestion

E. coli DH5

plating, over night incubation (+Tc / +Cm)

Picking up colonies

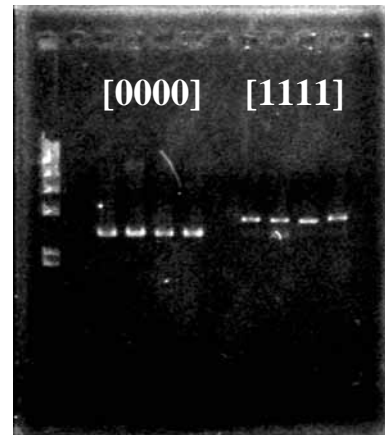
over night incubation (+Tc / +Cm)

Extraction

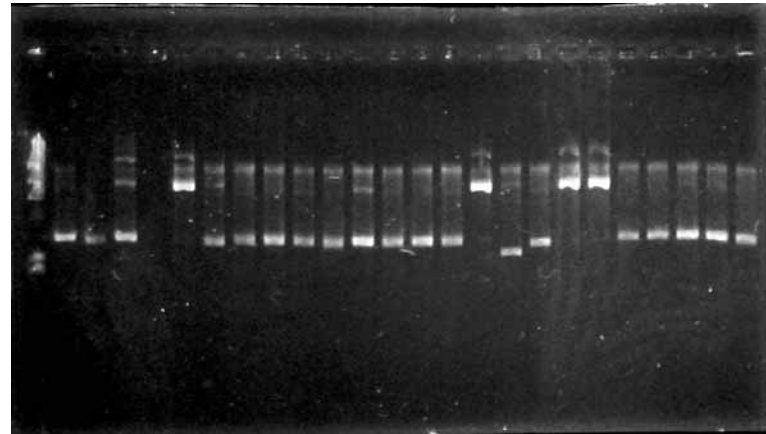
add enzyme (+*Bgl*III / +*Bln*I / +*Nde*I / +*Xho*I)

Gel Electrophoresis

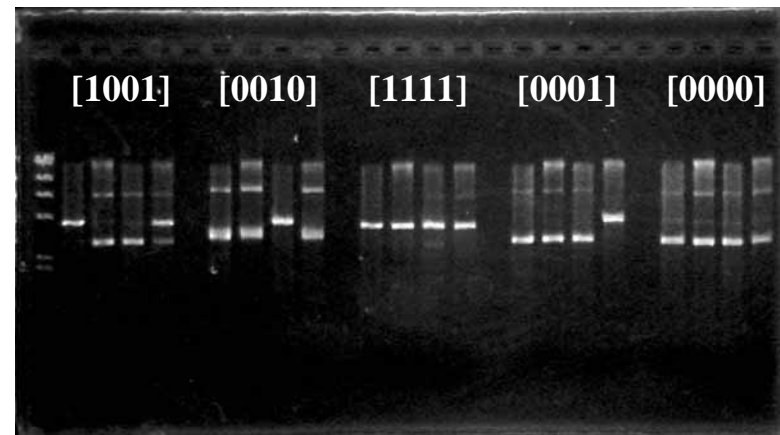
(a) Parents



(b) Fusion plasmids



(c) Children



最後に

- 新しい反応スキームが続々と提案されている
- 新しい要素分子が続々と創製されている
 - 化学者の努力の賜物
 - 例えば、様々な非天然塩基
- 分子システムの設計原理が待望されている
 - システマティックな設計論
 - 計算論的な側面

分子プログラミング