生物高次情報データベース
高速検索アルゴリズム

渋谷　哲朗

東京大学　医科学研究所

本日の話題

- **生物学データベースと検索アルゴリズム**
  - 様々なデータベース
  - タンパク質立体構造データベース検索アルゴリズム
- **今後の革新的アルゴリズム基盤構築に向けて**

生物学におけるデータベースと検索

- **配列データベース**
  - DNA・RNA・タンパク質配列データベース
  - モチーフ（パタン）データベース
  - 次世代シーケンサーデータ・アーカイブ
  - 個人ゲノム・ハプロタイプデータ

検索アルゴリズムの研究基盤はきわめて整備されている。
しかし、次世代シーケンサーの登場による**爆発的なデータ大規模化**への対応が急務
（こちらも勿論重要だが）

- **高次構造データベース**
  - 3次元・高次元データDB
    - **タンパク質立体構造DB**
    - 質量分析データベース
    - マイクロアレイ・データベース
  - 木・グラフ構造データベース
    - RNA2次構造データベース
    - 代謝経路データベース
    - タンパク間相互作用データベース

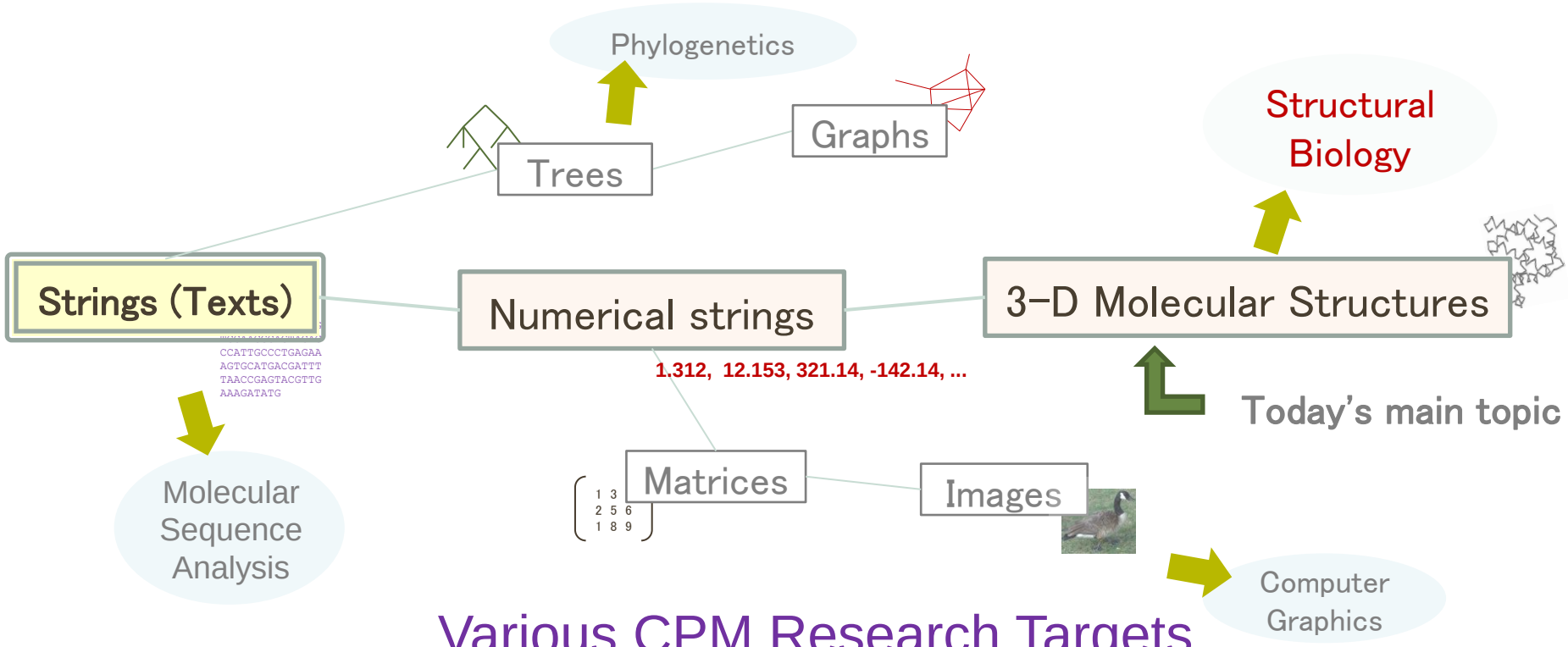
配列DBほどではないにしてもやはり指数関数的に大規模化しており、検索アルゴリズムの重要性が高まりつつある。
アルゴリズム基盤の構築が急務
（今日の話の重点はここ）

- **文献データベース**
  - 医療・生物学文献
  - 電子カルテ

自然言語処理分野

配列検索のためのアルゴリズム基盤

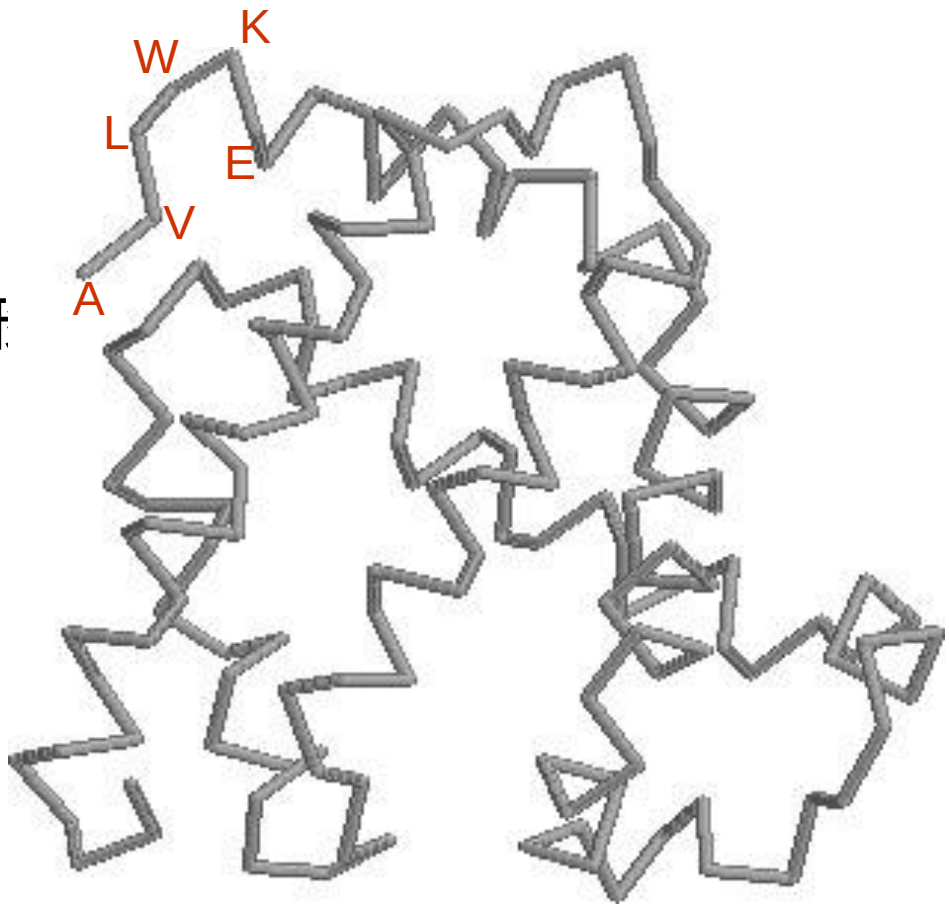
- **組み合わせパタンマッチング**
  - 離散構造の比較をするアルゴリズム研究分野
    - 多くの応用分野
  - 配列検索アルゴリズムの基盤
    - 接尾辞配列、接尾辞木、BW変換、圧縮索引、PatternHunter、 etc



Various CPM Research Targets

CPMの高次元構造DB検索への拡張

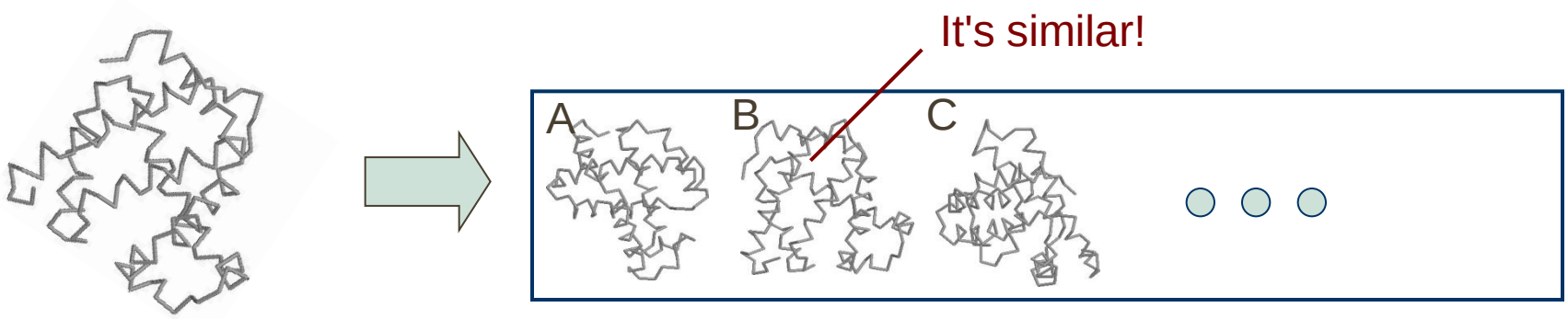
- **タンパク質3次元立体構造**
  - 20種類の塩基からなる鎖状分子
  - 3次元空間上で何らかの形をとる



タンパク質3次元立体構造DB検索

- **類似構造**
  - 機能も類似する　→　機能推定のために類似検索が必要
- **PDB (Protein Data Bank)**
  - 70,000～エントリー
  - 年率20%程度ずつ増加

→ **高速検索アルゴリズム基盤の構築が急務！**



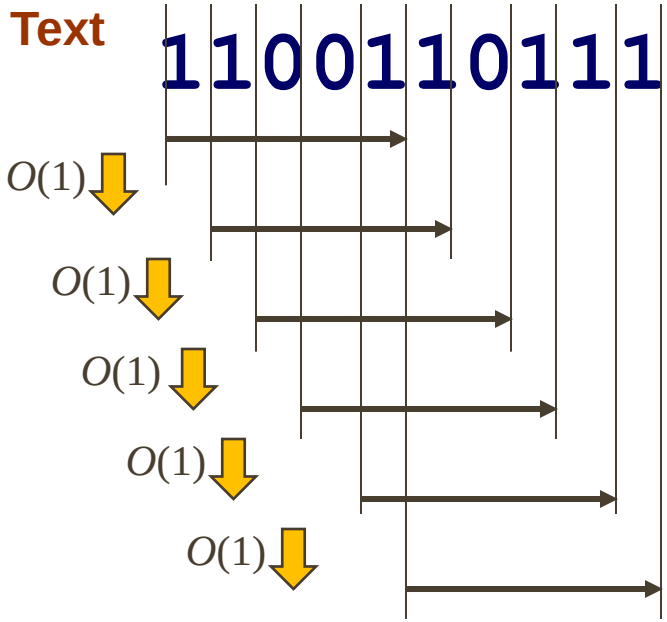
Query: Protein structure

Protein Structure Database

基本的な文字列CPMアルゴリズム

	Exact matching	Inexact matching
Comparison	(Trivial)	Hamming distance Alignment
Text searching without preprocessing texts	Knuth-Morris-Pratt Boyer-Moore Karp-Rabin Shif-Or Aho-Corasick etc.	FFT-based searching Smith-Waterman etc.
Text indexing (Preprocessing the texts before searching)	Suffix trees Suffix arrays Compressed suffix arrays Burrows-Wheeler transform Hashing etc.	BLAST FASTA BLAT PatternHunter etc.

Karp-Rabin (1981)



(16+8+1) mod 5 = 0

((0-1·16)·2+1) mod 5 = 4

((4-1·16)·2+0) mod 5 = 1

((1-0·16)·2+1) mod 5 = 3 ⇨ check → NO

((3-0·16)·2+1) mod 5 = 2

((2-1·16)·2+1) mod 5 = 3 ⇨ check → YES!

ハッシュ値の計算は全体で線形時間

ハッシュを用いて（平均的に）線形時間

Pattern 10111

(16+4+2+1) mod 5 = 3

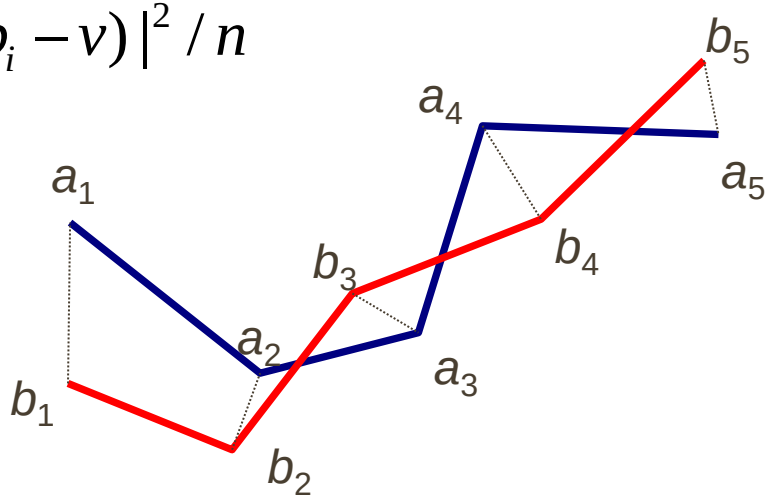
立体構造検索版Karp-Rabinは可能か？

- **そもそも何を検索するのか？**
  - 類似指標の定義が必要
- **ハッシュは使えるのか？**
  - 3次元構造をハッシュするのは難しい
    - 名前的にはGeometric hashingという手法はあるが、あくまでヒューリスティックなもの
- **平均性能とは？**
  - 文字列では「ランダム」な文字列を仮定することは易しい
  - 立体構造DB上で、「ランダムな（あるいは平均的な）立体構造」とは何か？

立体構造の比較指標

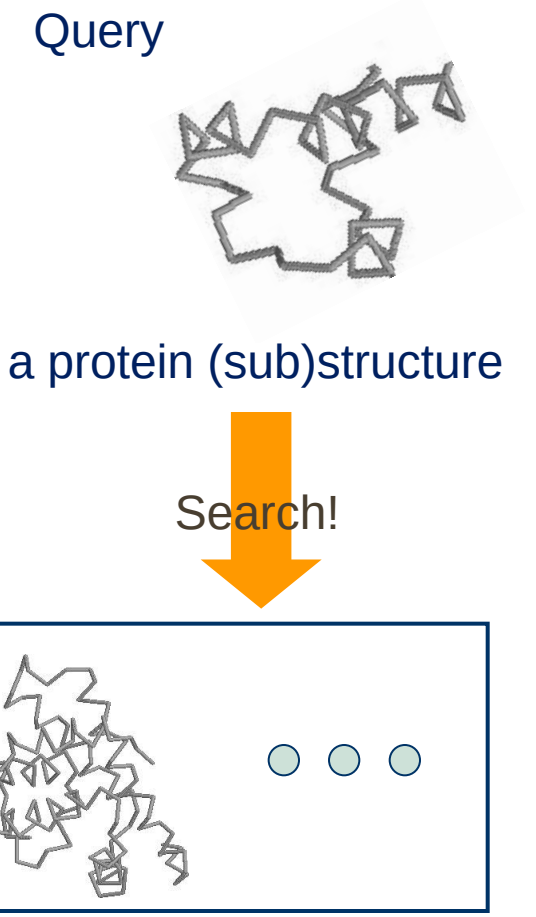
- **RMSD: Root Mean Square Deviation**
  - 最も一般的な指標
  - SVDを用いて $O(n)$ 時間で計算可能 [Kabsch '76]
    - n: chain length
    - Correspondence of atoms is given

$$RMSD(A,B) = \min_{R,v} \sqrt{\sum_{i=1}^{n} | a_i - R \cdot (b_i - v) |^2 / n}$$



基本的な問題

- **Database**
  - Protein 3-D structures in a database
- **Query**
  - A (sub)structure
- **Output**
  - All the similar substructures in the database
    - *i.e.,* RMSD ≤ some given bound c
    - No insertions/deletions



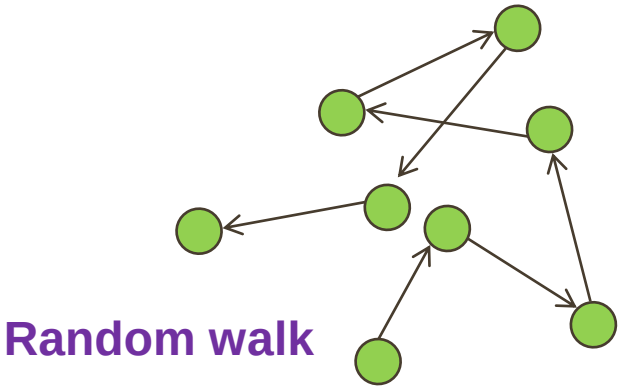
既存手法

- **単純な $O(Nm)$ のアルゴリズム**
  - Compute RMSDs for all the N-m+1 substructures of length m in the database
    - N: sum of lengths of all the structures in the database
    - m: query size
- **理論的には $O(N \log m)$ のアルゴリズムも存在**
  - Convolution technique based on FFT [Schwartz et al. '87]
  - An interesting algorithm, but it's practically not so faster than the naive algorithm

... That's all!

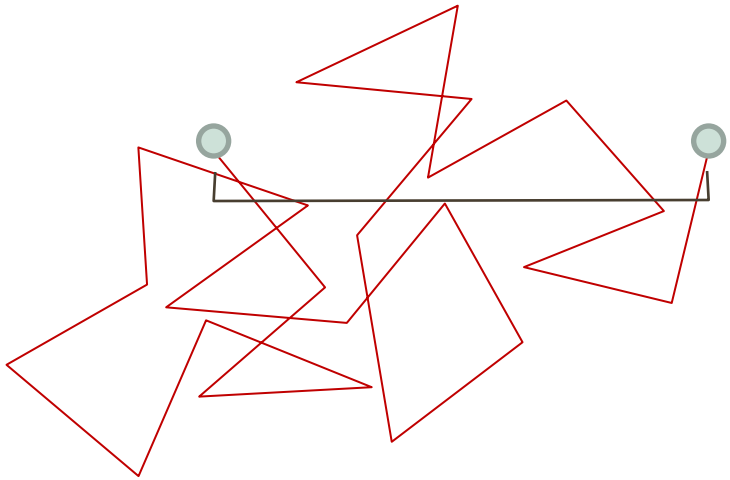
ランダムな鎖状分子のモデル

- **Freely-jointed chain (FJC) model**
  - 理論分子物理で最初に出てくる最も簡単なモデル
    - いわゆる3次元上のランダム・ウォーク
    - 単に '*Random-walk* model' とか '*Ideal chain* model' とも呼ばれる
  - 鎖状分子の振る舞いをよく反映する
    - ただし、実際には、もっと考慮するべきことは多い
      - 分子の衝突、結合角、α ヘリックス等の頻出構造、などなど



ランダムウォークの特徴

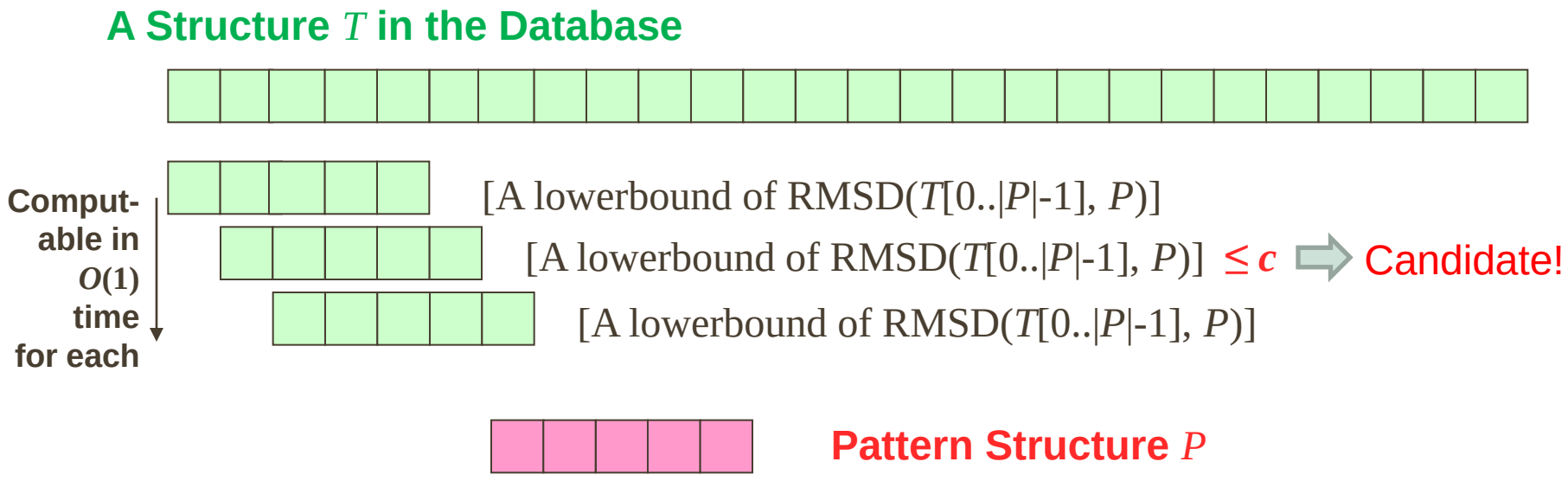
- **長さ n のランダムウォークの両端間の距離は $O(n^{1/2})$**
  - 次元に関わらず



3次元版Karp-Rabinアルゴリズム！

[Shibuya, RECOMB '09]
(Best Paper Award)

- **単純なフィルタリングによる高速化**
  - ハッシュのかわりにRMSDのlower boundを計算（線形時間）
  - Lower boundが閾値以下のものだけに対してきちんとRMSDを計算する



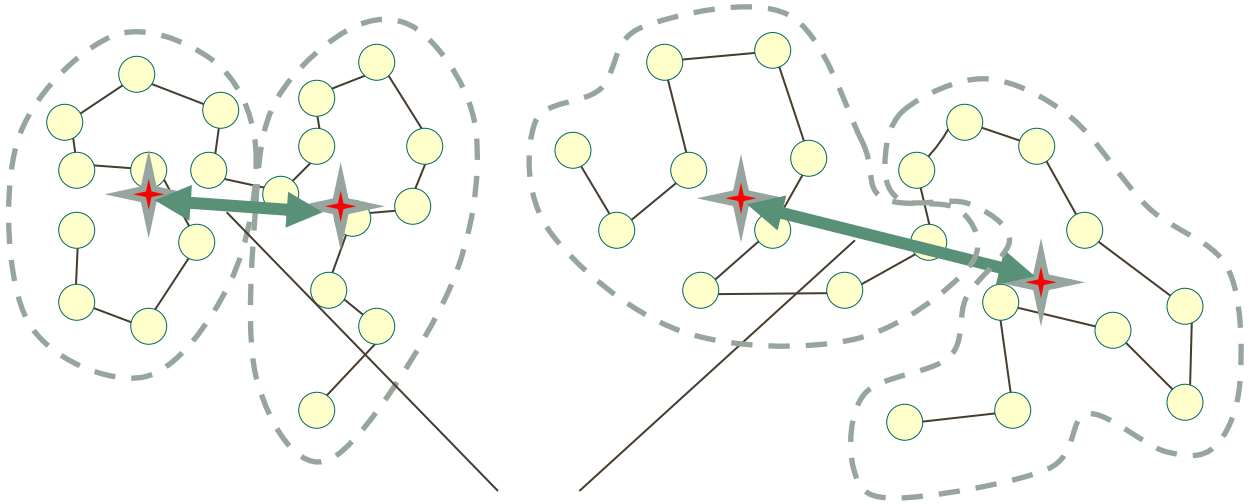
Compute the RMSD only when the lower bound is $\leq c$

キーとなるアイディア

- **フィルタリングで残った候補数の期待値が $O(N/m)$**
  - 候補すべてに対してRMSDを計算しても線形時間！
    - N: database size　　m: query size
- **Assumption**
  - FJCモデル
    - データベース中の構造がランダムウォークに従うことを仮定して、フィルタリング後の候補数を見積もる
    - クエリーはどんな構造でもよい

$O(N \cdot m^{1/2})$ を実現するlower bound

- **$D_1(P[1..m],Q[1..m])$　（m: even number）**
  - $|H(P[1..m]) - H(Q[1..m])| /2$
    - where $H(P) = |G(P[1..m/2]) - G(P[m/2+1..m])|$
    - $G(S)$ is the centroid (center of mass) of structure S
    - Consider n as an even number (to simplify the discussion)
  - It is always smaller than or equal to $RMSD(P, Q)$



Half of the difference of the two distances

実際にも速い！

- **Target database : The whole PDB (September 5th, 2008)**
  - 52,821 entries / 244,719 chains / 38,267,694 a.a.
- **Query**
  - 100 random substructures of each specified length, taken from PDB
  - Threshold: 1 Å
- **Computation Time (sec)**
  - Average computation time of 100 random queries
  - on 1 CPU of 1200MHz UltraSPARC III on SunFire 15K

Query Length	40	80	120	160	200
#Substructures	33,722,208	21,692,707	16,134,096	12,362,509	9,559,056
#Hits	38.1	32.9	27.3	16.0	23.2
D_1	98.9	92.4	75.6	59.4	60.0
D_2	58.9	36.4	32.8	27.3	25.7
D_3	74.5	25.5	17.3	14.2	12.9
Naive	447.0	442.0	415.2	378.9	342.5
FFT	531.9	463.1	399.8	330.6	293.0

(sec)

さらには

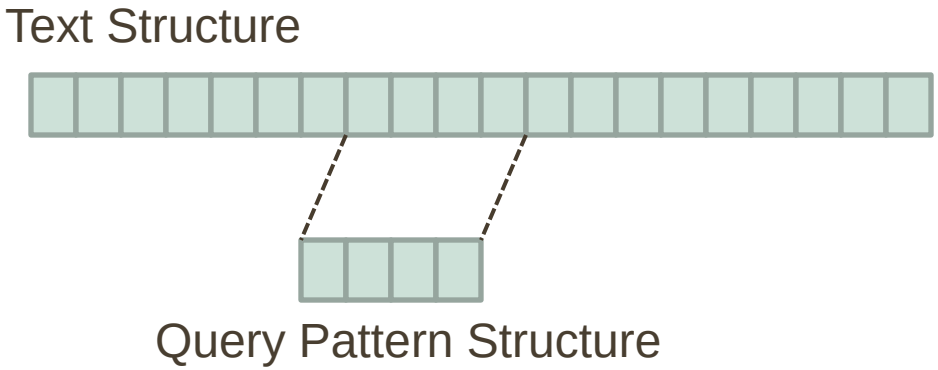
- **さまざまな拡張アルゴリズム**
  - $O(m + N / m^{1-\varepsilon})$ アルゴリズム
    - *cf.* Boyer–Moore averagely runs in $O(m+N/m)$ time
  - Indelを考慮した線形時間アルゴリズム
  - 索引化による高速化
    - さらに数十倍の高速化が可能

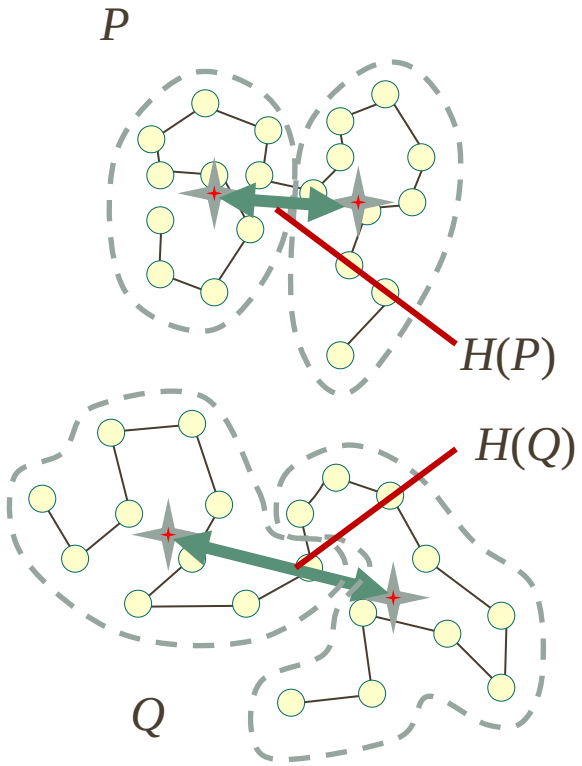
etc.

Geometric Suffix Array

[Shibuya, RECOMB '09]

- **D_1–based candidates can be searched with binary search!**
  - on a sorted array of centroid–centroid distances $H(T[i..i+m\text{-}1])$
  - for fixed–length queries





$$D_1 = |H(P) - H(Q)| / 2$$

3次元版の存在するCPMアルゴリズム

	Exact matching	Inexact matching
Comparison	(Trivial)	Hamming distance Alignment
Text searching without preprocessing texts	Knuth-Morris-Pratt Boyer-Moore Karp-Rabin Shif-Or Aho-Corasick etc.	FFT-based searching Smith-Waterman etc.
Text indexing (Preprocessing the texts before searching)	Suffix trees Suffix arrays Compressed suffix arrays Burrows-Wheeler transform Hashing etc.	BLAST FASTA BLAT PatternHunter etc.

高次検索アルゴリズムの基盤として何が必要か？

- **対象のふるまいの理解**
  - タンパク質立体構造データベースの場合、FJCモデルを仮定することで、新たなアルゴリズムの設計・性能評価が可能となった上に、実用上も極めて高速な（従来比で数十倍～数百倍）アルゴリズムが達成できた
  - 特に大規模なデータベースでは、対象の統計的なふるまいの理解が必要
    - 「平均計算量」の概念をデータベース毎に吟味する必要がある
      - 複雑すぎれば解析不能だが、簡単すぎれば実際にそぐわない。
- **発展可能なアルゴリズム・フレームワーク**
  - 単発の検索アルゴリズムを作っても、発展しない
  - タンパク質立体構造データベース検索の場合、CPM的な考え方を流用したため、様々な拡張が可能

まとめ

- **高次構造データベース検索**
  - タンパク質立体構造ではFJCモデル（ランダムウォーク）の性質をうまく活用した、超高速検索アルゴリズムの開発に成功
    - 様々な拡張も可能
- **今後の展望**
  - 検索が必要なデータベースに対して、その対象の統計的な振る舞いを利用したアルゴリズムの高速化の必要性
    - 質量分析DB
      - 鎖状分子の重さの分布
    - 立体構造シミュレーションDB
      - 分子の動き
    - 時系列発現データ
      - 発現は何らかのモデルに基づいて偏りがある
    - その他、高次元データ・構造データを含む大規模DB

Thank you!