

複合ソート法による高速な 全ペア類似度検索

津田 宏治

産総研CBRC / JST ERATO

Collaboration with 田部井靖生、清水佳奈、伊東純一、
富井健太郎、杉山将、宇野毅明

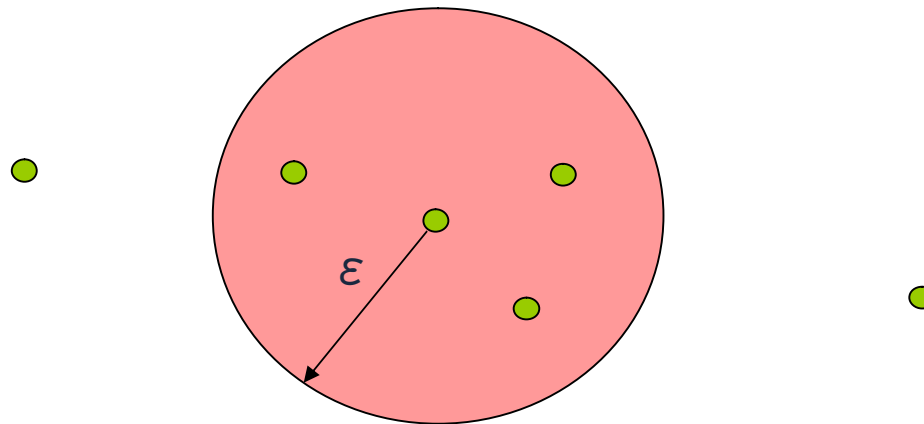
全ペア類似度検索

§ 近傍グラフ

§ 半教師つき学習、スペクトラルクラスタリングなどで必要

§ ϵ -近傍グラフの作成問題

§ Find all pairs (i, j) , $i < j$, that $\Delta(x_i, x_j) \leq \epsilon$



§ 全ペアの距離を計算すると $O(n^2)$

アウトライン

- § ソーティングによって、高速に全ペア類似度検索を行う方法「複合ソート法」を提案
- § ハミング距離に基づく全ペア類似度検索
- § コサイン距離に基づく全ペア類似度検索
 - § 画像、シグナル
- § 応用例

ハミング距離がd以下のペア発見

1:	1011	1111	0011	1110
2:	1101	0111	0111	0001
3:	1100	1000	1101	1100
4:	0100	0001	0111	1101
5:	1010	0010	1110	1010
6:	1111	0011	1001	0111
7:	0000	0001	0011	1110
8:	0101	1001	0111	1000
9:	1101	1000	1101	1110
10:	1001	0011	1001	0111

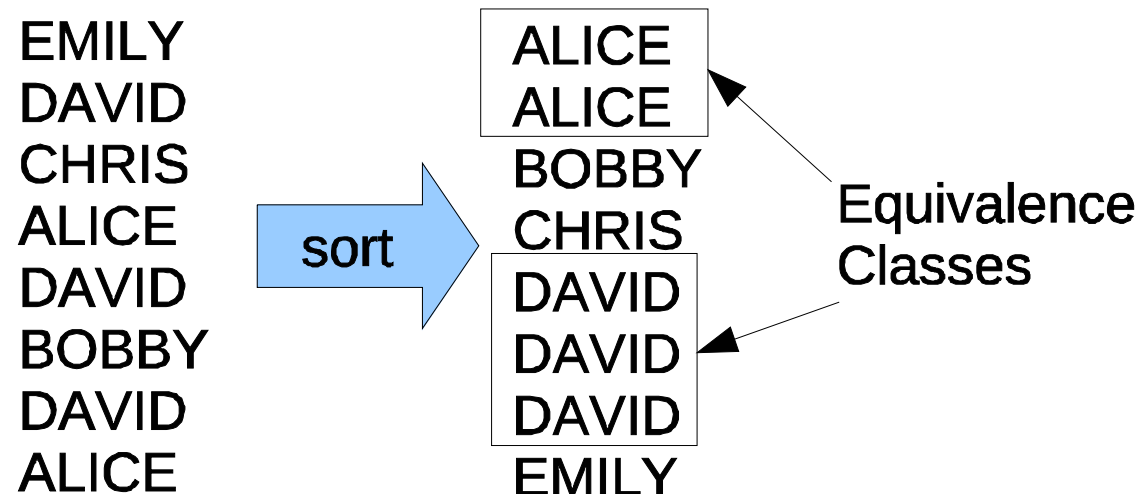
複合ソート法 (Multiple Sorting Method, Uno 2008)

- § 同じ長さ l の文字列が n 個与えられている
- § ハミング距離 d 以内のペアの全列挙
- § 距離 d 以内のペアの数を m とする
- § 基数ソートを再帰的に繰り返して、全ペアを $O(n+m)$ で列挙可能

- § この計算量を達成する方法(文字マスク)は、定数が大きい
ため実際には低速
- § ブロックマスクの導入による高速化

Special Case: 全く同じ文字列ペアの発見 ($d=0$)

- § 基数ソートの後, 文字列をEquivalence classに分割: $O(n)$
- § Equivalence class内の全ペアにエッジを張る: $O(m)$
- § 計算量: $O(n+m)$



複合ソート法 (文字マスク)

§ d個の文字を全通り選んでマスクする

§ 基数ソートを $\binom{\ell}{d}$ 回繰り返す

§ 計算量はdに対して指数、lに対しても多項式

§ しかし、文字列の数nに関しては線形のまま $O(n+m)$

7:000	0001	0011	11	0	7:0	0	0001	0011	1110
4:010	0001	0111	11	1	4:0	0	0001	0111	1101
8:010	1001	0111	10	0	8:0	1	1001	0111	1000
10:100	0011	1001	01	1	5:1	0	0010	1110	1010
5:101	0010	1110	10	0	3:1	0	1000	1101	1100
1:101	1111	0011	11	0	6:1	1	0011	1001	0111
2:110	0111	0111	00	1	10:1	1	0011	1001	0111
3:110	1000	1101	11	0	2:1	1	0111	0111	0001
9:110	1000	1101	11	0	9:1	1	1000	1101	1110
6:111	0011	1001	01	1	1:1	1	1111	0011	1110

ブロックマスク

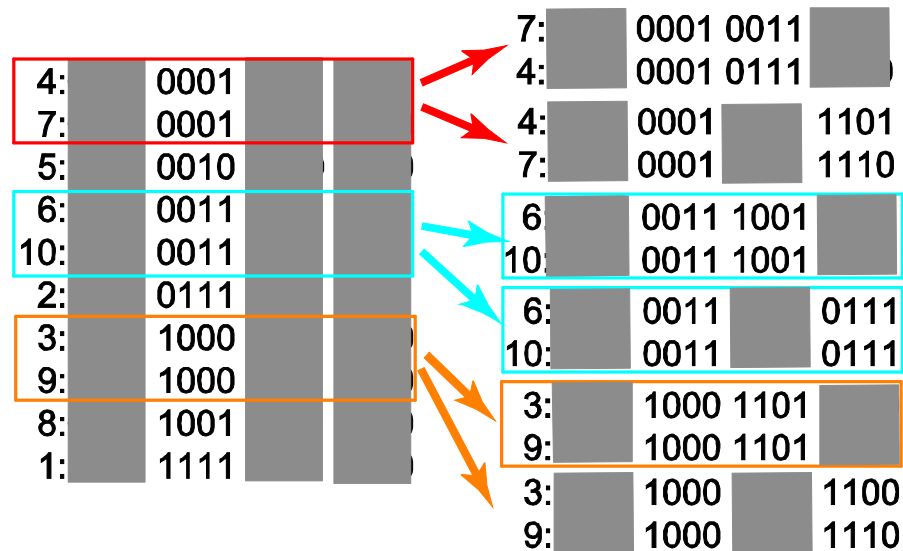
- § 文字列をk個のブロックに分割する
- § d個のブロックを、全通りマスクする
- § ソートの回数が劇的に減る
- § 近傍でないペアも検出されてしまう
 - § 実際に検出されたペアのハミング距離を計算して排除

7:0000 0001			7:0000		0011		7:0000			1110
4:0100 0001			4:0100		0111		4:0100			1101
8:0101 1001			8:0101		0111		8:0101			1000
10:1001 0011			10:1001		1001		10:1001			0111
5:1010 0010			5:1010		1110		5:1010			1010
1:1011 1111			1:1011		0011		1:1011			1110
3:1100 1000			3:1100		1101		3:1100			1100
2:1101 0111			2:1101		0111		2:1101			0001
9:1101 1000			9:1101		1101		9:1101			1110
6:1111 0011			6:1111		1001		6:1111			0111

7:		0001 0011		4:		0001 1101		1:		0011 1110
4:		0001 0111		7:		0001 1110		7:		0011 1110
5:		0010 1110		5:		0010 1010		2:		0111 0001
6:		0011 1001		6:		0011 0111		8:		0111 1000
10:		0011 1001		10:		0011 0111		4:		0111 1101
2:		0111 0111		2:		0111 0001		6:		1001 0111
3:		1000 1101		3:		1000 1100		10:		1001 0111
9:		1000 1101		9:		1000 1110		3:		1101 1100
8:		1001 0111		8:		1001 1000		9:		1101 1110
1:		1111 0011		1:		1111 1110		5:		1110 1010

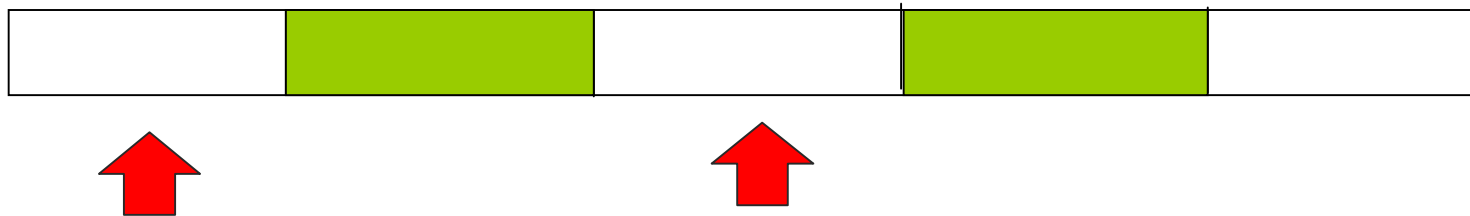
再帰アルゴリズム

- § まず第一のブロックをソートし、Equivalence classを発見する
- § 各々のEquivalence classに対して、次のブロックを付け加えて、ソートする。
- § k-d個のブロックが繋がったら、各Equivalence Classに入っているペアに対して「重複排除」を行う
- § 生き残ったペアに関して、ハミング距離を実際に計算



重複排除

- § ブロック列に、Lexicographical Orderを導入
- § あるブロック列に関して完全一致のペアが見つかったも、ブロック列が「最小」でなければ、出力しない
- § 最小性判定



ブロック列中の最右のブロックよりも、左にある空ブロックが完全一致
= 最小でない

Algorithm 1 Multiple Sorting Method. d : Hamming distance threshold, k : number of blocks.

```
1: function MULTIPLESORTINGMETHOD
2:    $I \leftarrow \{1, \dots, n\}$ 
3:    $B \leftarrow \phi$ 
4:   RECURSION( $I, B$ )
5:   return
6: end function
7: function RECURSION( $I, B$ )
8:   if  $|B| = k - d$  then
9:     for  $(i, j) \in I \times I, i < j$  do
10:      if  $s_i^b \neq s_j^b$  for all  $b < \max(B), b \notin B$  then
11:        if  $\text{HamDist}(s_i, s_j) \leq d$  then
12:          Report  $(i, j)$  to output file
13:        end if
14:      end if
15:    end for
16:    return
17:  end if
18:  for  $b$  in  $(\max(B) + 1) .. (k + |B| + 1)$  do
19:     $J \leftarrow$  Sorted indices based on  $b$ -th block  $\{s_i^b\}_{i \in I}$ 
20:     $T \leftarrow$  Intervals of equivalence classes in  $\{s_j^b\}_{j \in J}$ 
21:    for each interval  $(x, y) \in T$  do
22:      RECURSION( $J[x : y], B \cup b$ )
23:    end for
24:  end for
25:  return
26: end function
```

複合ソート法の疑似コード

ブロック数が $k-d$ に達した際のペアの数え上げ

Equivalence Class
の再帰的展開

コサイン距離での全ペア類似度検索

- § 画像、音声などは、実数値をもつ特徴ベクトルとして表現
- § Locality sensitive hashingを用いて、ベクトルを二値の文字列に変換 (スケッチ).
 - § 類似したベクトルは、類似した文字列になる
 - § 符号付きランダム射影
 - § 他の方法を用いれば、ユークリッド距離、Jaccard係数などによる検索も可能
- § Missing edge ratio (ペアを逃す確率) を 10^{-6} 以下に抑える
- § Cover tree (Beygelzimer et al., ICML2006) より10倍以上高速
- § 160万個の画像データを用いて実験

ベクトルを、0/1の文字列に射像する

§ コサイン距離
$$\Delta(x_i, x_j) = 1 - \frac{x_i^\top x_j}{\|x_i\| \|x_j\|}.$$

§ Locality sensitive hashing

§ D次元のベクトルを長さlの0/1文字列にする

§ $N(0, 1)$ からのサンプルで構成された $D \times l$ 行列を使用

§ 射影先のハミング距離の期待値が、元の空間でのコサイン距離に対する単調増加関数

Cosine LSH

§ $R \in \mathbb{R}^{D \times \ell}$: $N(0, 1)$ からサンプルされたランダム行列

§ 写像

$$s_{ik} := \text{sign}(\mathbf{r}_k^\top \mathbf{x}_i), k = 1, \dots, \ell$$

§ 非衝突確率は、角度に比例

$$\Pr(s_{ik} \neq s_{jk}) = \frac{\theta_{ij}}{\pi}, \quad \forall k,$$

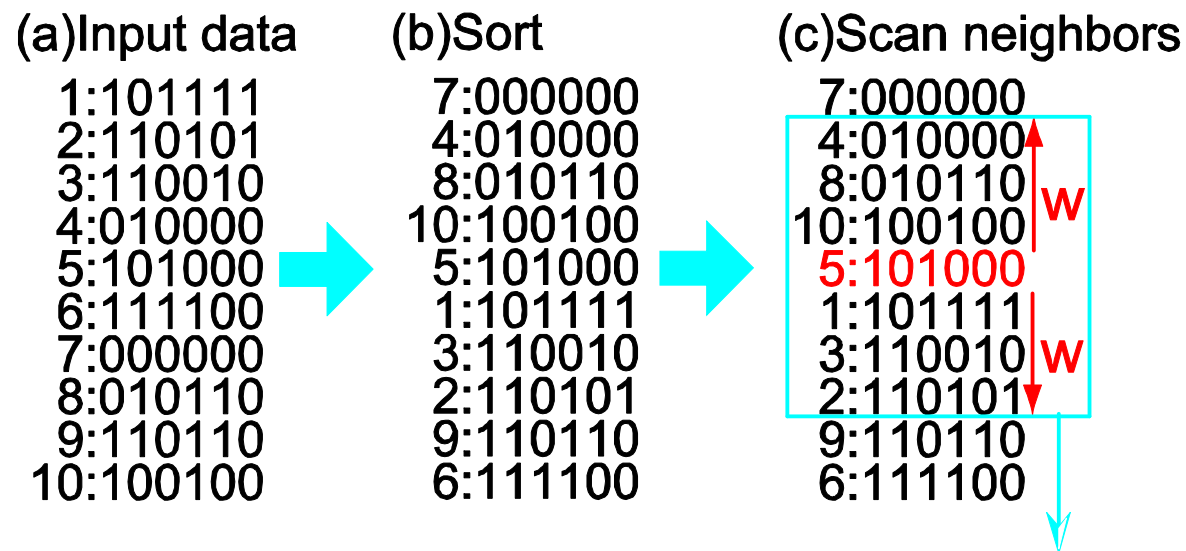
$$\theta_{ij} = \arccos\left(\frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}\right).$$

§ ガウシアンカーネルベースのスケッチを用いれば、ユークリッド距離でも可能 (Raginsky and Lazebnik, NIPS 2009)

§ 近年、多種多様なスケッチが提案されている

単純ソート法 (Chariker, 2002)

- § Sketchから、全ペア類似度検索を行う方法
- § ソートして、あるウィンドウ幅 w でスキャンする
- § 同じウィンドウに入ったペアに関して、コサイン距離を計算



SketchSort

- § 基本アイデア: ベクトルを文字列にして、複合ソート法適用
- § Not good: 長い文字列に、複合ソート法を適用する
- § チャンクへの分割
 - § 長さ ℓ の文字列プールを Q 個つくる
 - § 複合ソート法を各プールに適用する

$$E_q = \{(i, j) \mid HamDist(\mathbf{s}_i^q, \mathbf{s}_j^q) \leq d, i < j\}.$$

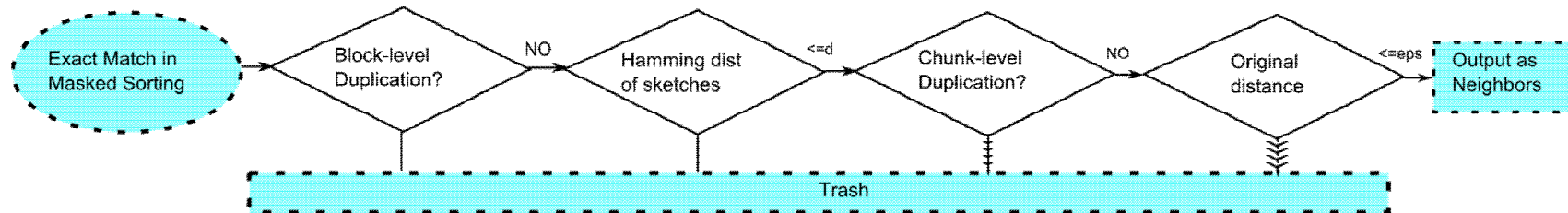
- § 全ての出力セットを併せる

$$E = E_1 \cup \dots \cup E_Q.$$

- § 中間結果 E の中で、 $\Delta(x_i, x_j) \leq \epsilon$ を満たすものを出
力

チャンク単位の重複排除

- § 異なるチャンクで同じペアが見つかり、重複が発生する
- § チャンク q で、ハミング距離 d 以内のペアが見つかった場合には、チャンク $1, \dots, q-1$ でハミング距離 d 以内のものがない場合だけ出力
- § 3重のチェック体制: できるだけコサイン距離の計算を避ける



SketchSortの疑似コード

各チャンクに対する呼び出し

```
1: function SKETCHSORT( $x_1, \dots, x_n$ )
2:   Use LSH to obtain sketches  $\{s_{1i}, \dots, s_{qi}\}_{i=1}^n$  from
   data  $\{x_i\}_{i=1}^n$ 
3:    $I \leftarrow \{1, \dots, n\}$ 
4:   for  $q = 1 : Q$  do
5:      $B \leftarrow \phi$ 
6:     RECURSION( $I, B, q$ )
7:   end for
8:   return
9: end function
10: function RECURSION( $I, B, q$ )
11:   if  $|B| = k - d$  then
12:     for  $(i, j) \in I \times I, i < j$  do
13:       if  $s_{qi}^b \neq s_{qj}^b$  for all  $b < \max(B), b \notin B$  then
14:         if  $\text{HamDist}(s_{qi}, s_{qj}) \leq d$  then
15:           if  $\text{HamDist}(s_{ri}, s_{rj}) > d$  for all  $r < q$ 
16:             then
17:               if  $\Delta(x_i, x_j) \leq \epsilon$  then
18:                 Report  $(i, j)$  to output file
19:               end if
20:             end if
21:           end if
22:         end if
23:       end if
24:     end for
25:     return
26:   end if
27:   for  $b$  in  $(\max(B) + 1) .. (k + |B| + 1)$  do
28:      $J \leftarrow$  Sorted indices based on  $b$ -th block  $\{s_{qi}^b\}_{i \in I}$ 
29:      $T \leftarrow$  Intervals of equivalence classes in  $\{s_{qj}^b\}_{j \in J}$ 
30:     for each interval  $(x, y) \in T$  do
31:       RECURSION( $J[x : y], B \cup b, q$ )
32:     end for
33:   end for
34: end function
```

ペアの数え上げ
(三重のチェック)

Equivalence Class
の再帰的展開

二種類のエラー

- § 真にエッジセット E^* , SketchSortによる中間結果 E
- § False positive: 近傍ではないペアが、1つ以上のチャンクでハミング距離 d 以内となる事象

$$F_1 = \{(i, j) \mid (i, j) \in E, (i, j) \notin E^*\}.$$

- § False negative: 近傍ペアが、全てのチャンクでハミング距離 $d+1$ 以上となる現象

$$F_2 = \{(i, j) \mid (i, j) \notin E, (i, j) \in E^*\}.$$

False negative rateの上限: Missing edge ratio

§ False negativeの方が致命的

§ False positiveは、コサイン距離計算によって除かれる

§ Missing edge ratio (False negative rate) は次のようにバウンドされる

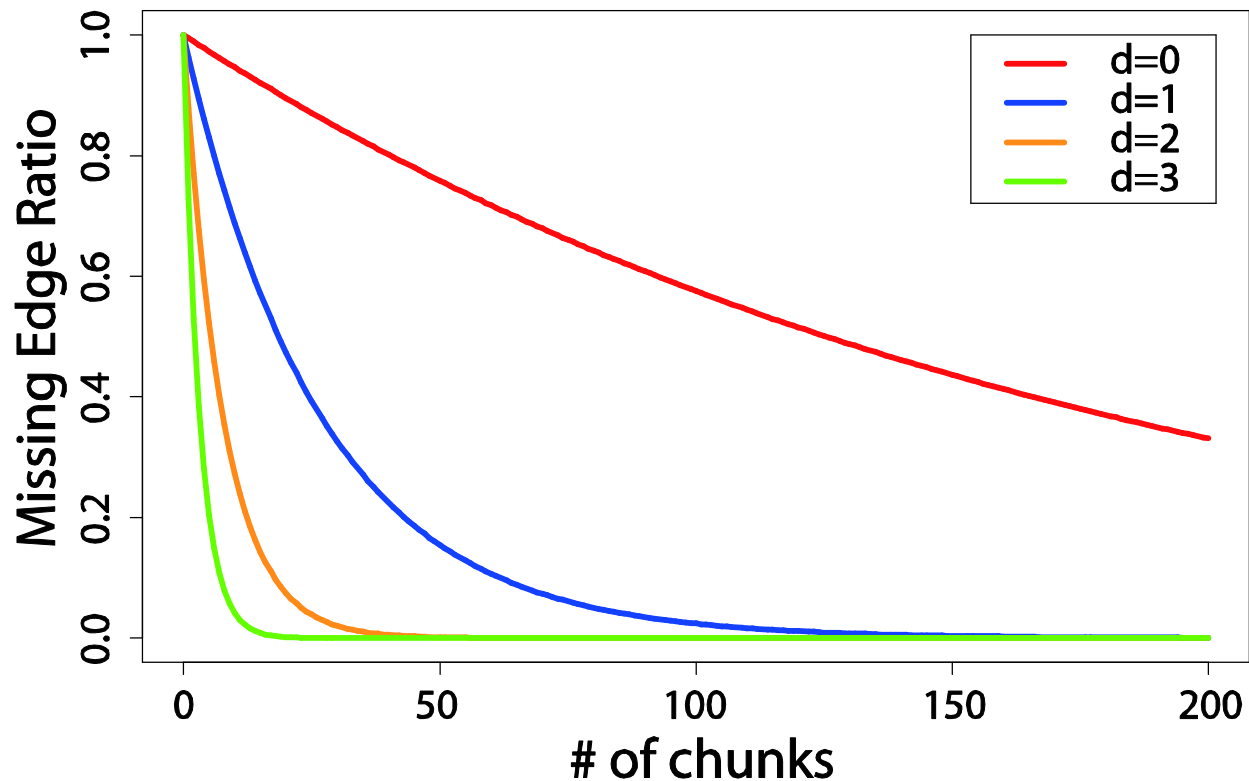
$$E \left[\frac{|F_2|}{|E^*|} \right] \leq \left(1 - \sum_{k=0}^{\lfloor d \rfloor} \binom{\ell}{k} p^k (1-p)^{\ell-k} \right)^Q,$$

ここで、 p は、LSHの非衝突確率の上限である

$$p = \frac{\arccos(1 - \epsilon)}{\pi}.$$

チャンクの数 Q に対する、Missing edge ratioの減少

§ チャンクの完全一致を用いる方法($d=0$ 、元祖LSH)では、十分にMissing edge ratioを減らせない



単純ソート、Lanczos Bisectionとの比較実験

§ 二つのデータセット

§ MNIST (60,000 points, 748 dims)

§ TinyImage (100,000 points, 960 dims)

§ Missing Edge Ratio計算のため、ダウンサンプリング

§ コサイン距離の閾値: 0.15π

§ 各チャンクは32ビット

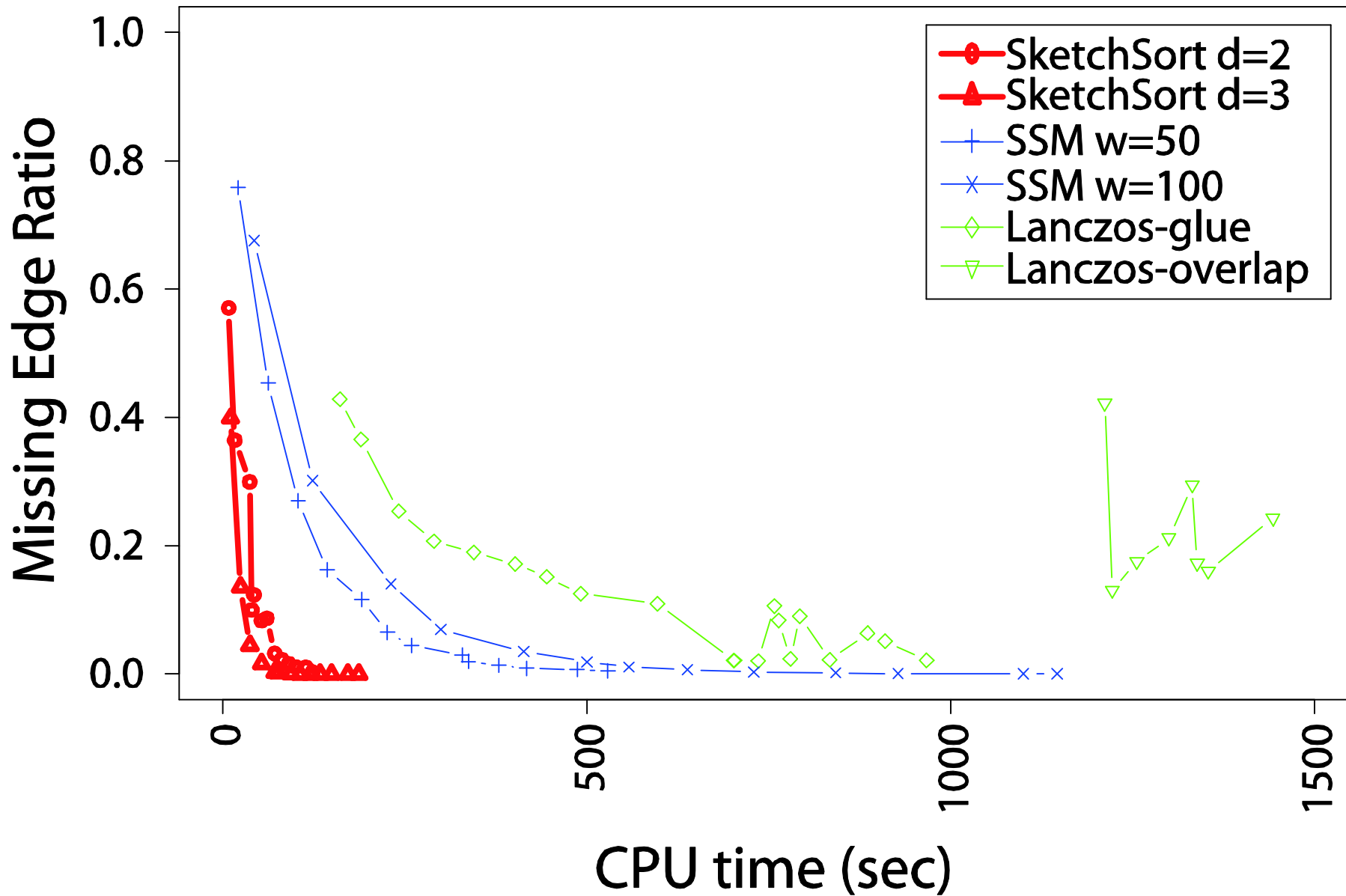
§ 複合ソートのハミング距離とブロック数: (2,5), (3,6)

§ チャンク数: 2, 6, 10, ..., 50

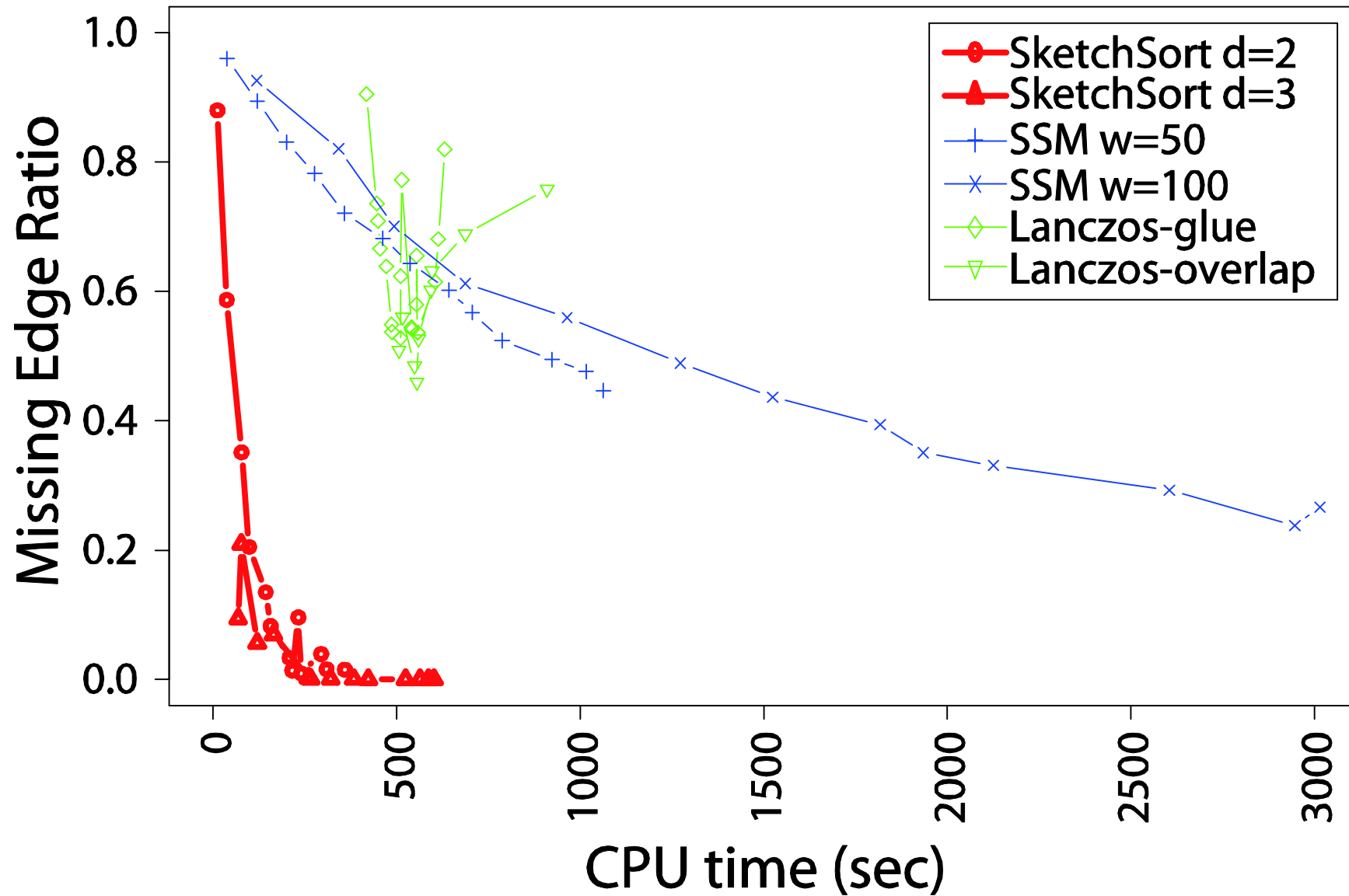
§ Lanczos Bisection (JMLR, 2009)とも比較

§ Lanczos法を使って空間を、再帰的に2分割する方法

MNIST, 閾値 0.15π



TinyImage, 閾值 0.15π



Nearest Neighbor検索実験

§ 複合ソートを改造

§ 最後にコサイン距離の閾値で排除する代わりに、各点で best-h の近傍を保持

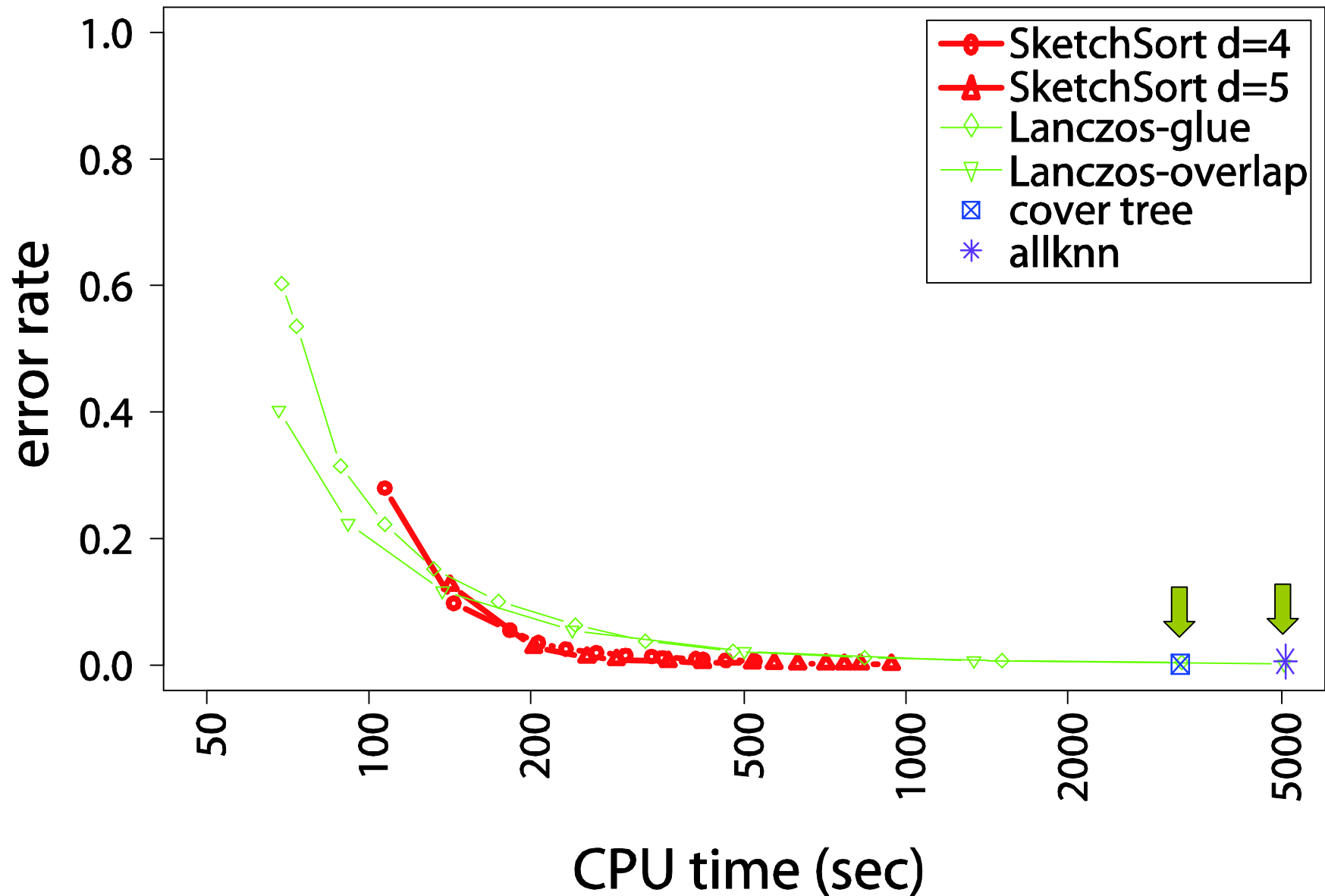
§ 全ての処理が終了したあと、近傍グラフ上の距離2のノードから、最も近いk個を選び出す

§ 追加する比較手法

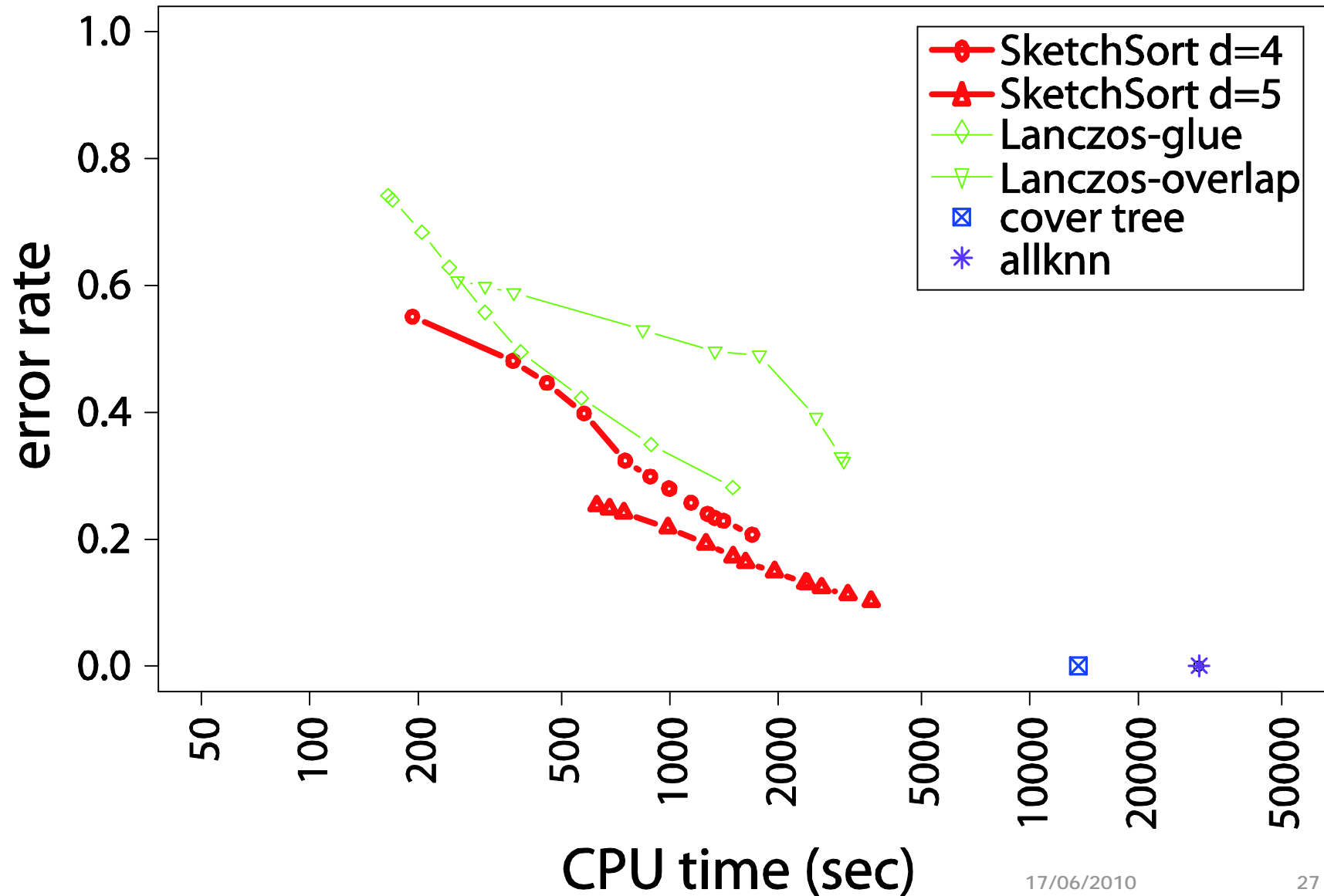
§ Cover Tree (Beygelzimer et al., ICML 2006)

§ AllKnn (Ram et al., NIPS 2009)

5-Nearest Neighbor検索 (MNIST)



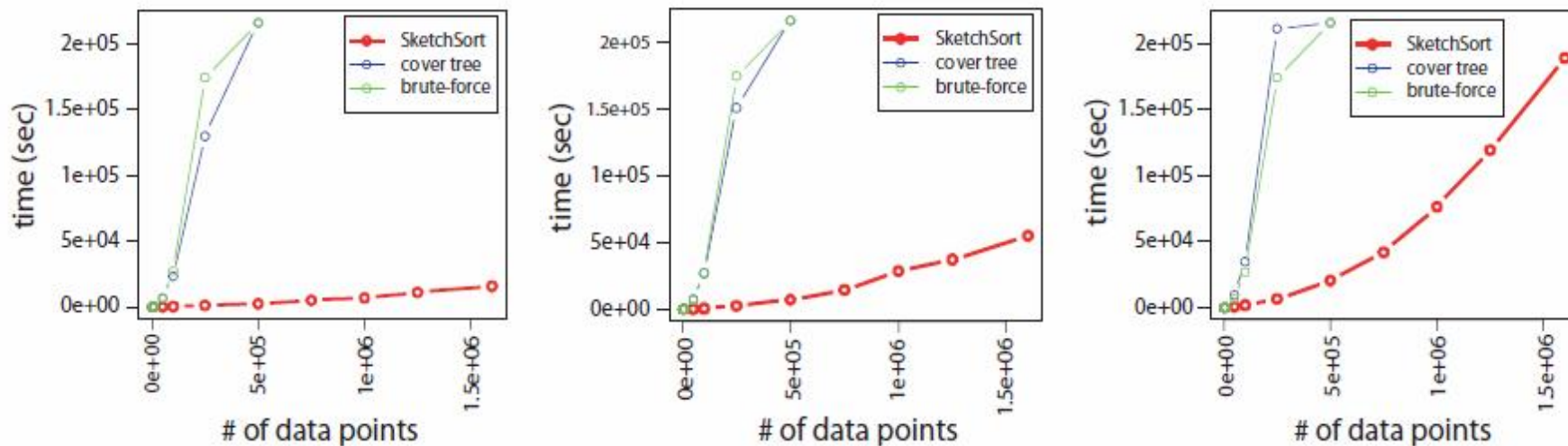
5-Nearest Neighbor検索 (TinyImage)



160万画像での実験

§ False positive rate $< 1.0 \times 10^{-6}$

§ 160万画像の全ペア類似度検索を、4.3時間で処理
(0.05π)



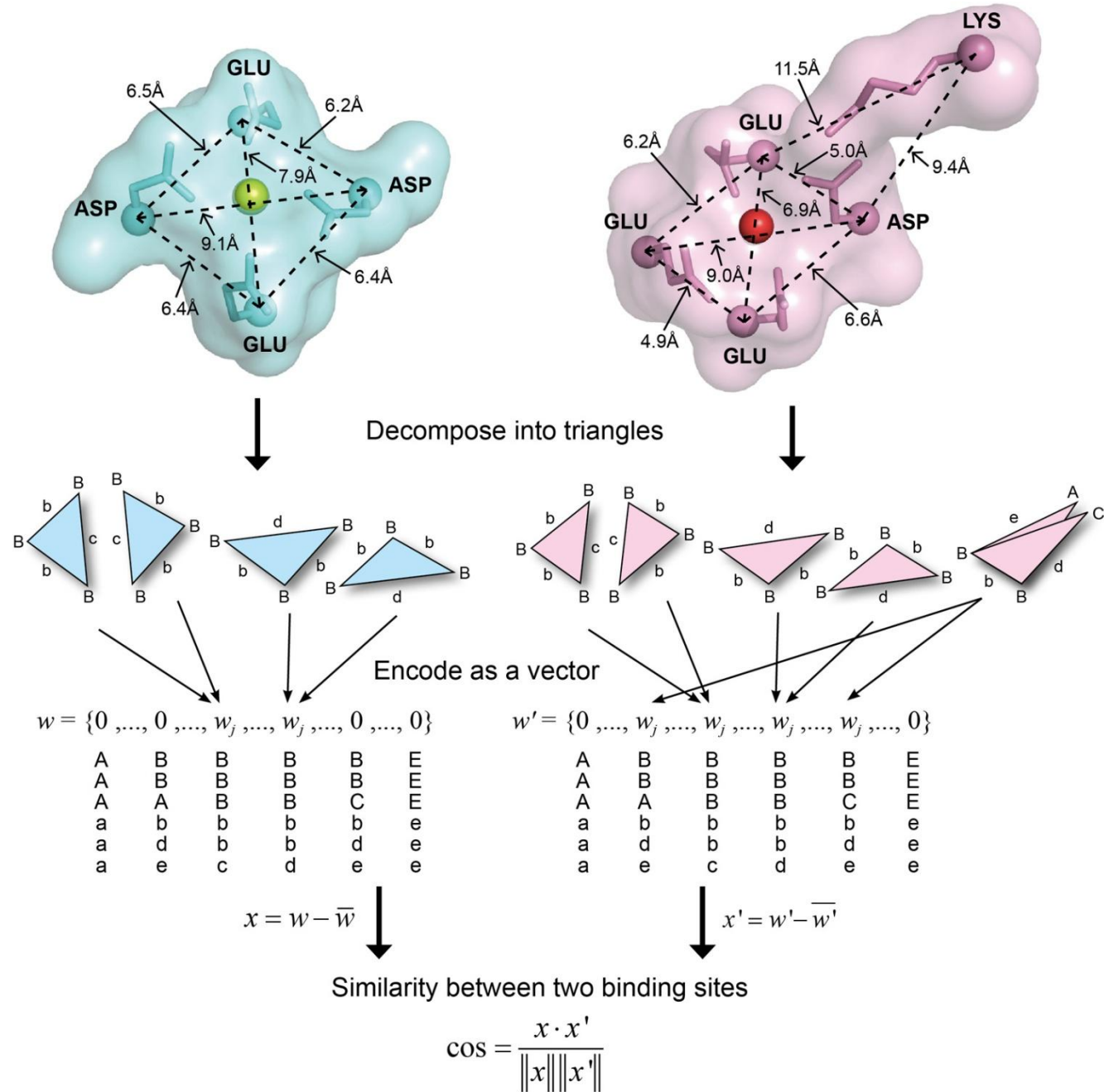
Near duplication detection in up to 1.6 million images at thresholds 0.05π (left), 0.10π (middle) and 0.15π (right)

タンパク質のリガンド結合可能部位の大規模解析

- § PDBに登録されているタンパク質三次元構造
- § 1,260,627個のリガンド結合可能部位(ポケット)を抽出
 - § その中の約20万個は、実際にリガンドが結合
- § SketchSortで類似するペアを列挙

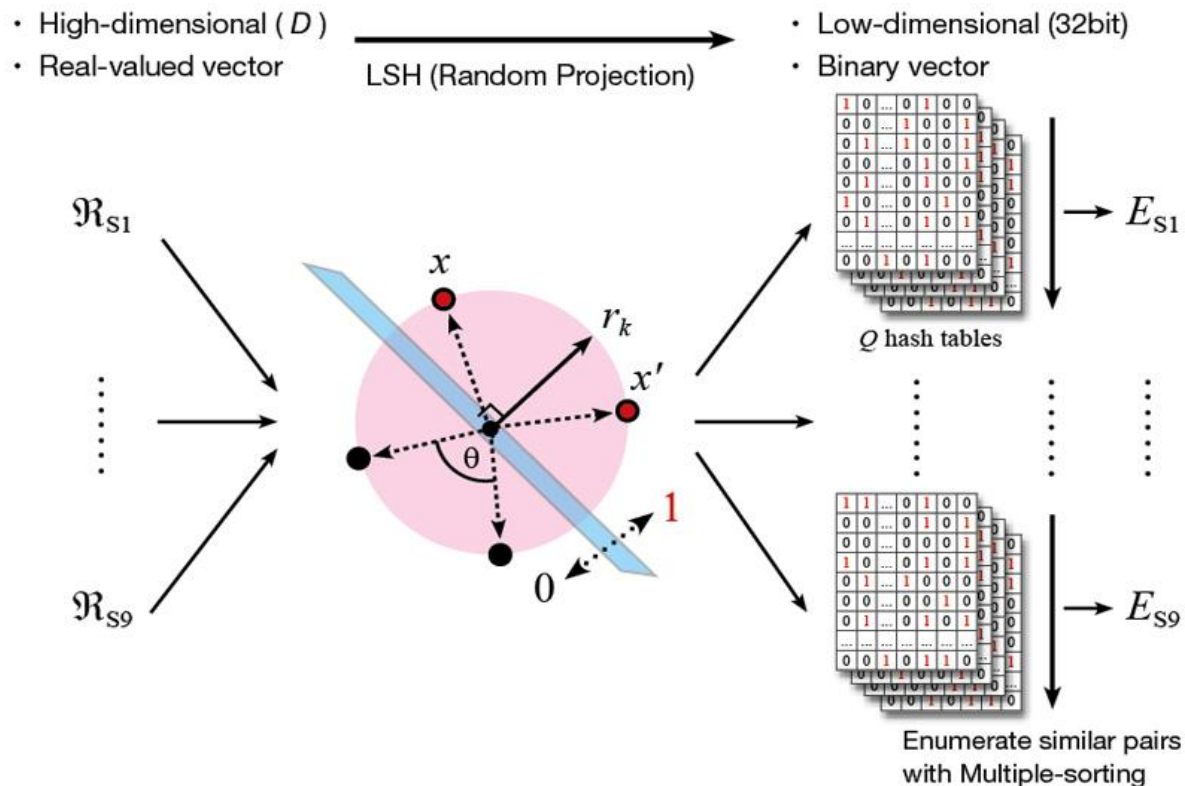
- § 通常は、アミノ酸配列の類似性によって、結合部位を発見 (Homolog)
- § しかし、配列が違っていても、三次元構造が同じであれば結合する可能性 (Analog)
- § 構造ゲノムプロジェクト等からもたらされるタンパク質立体構造からの機能推定や、ドラッグ設計の分野におけるスクリーニングに利用

幾何的 特徵抽出



SketchSortの適用

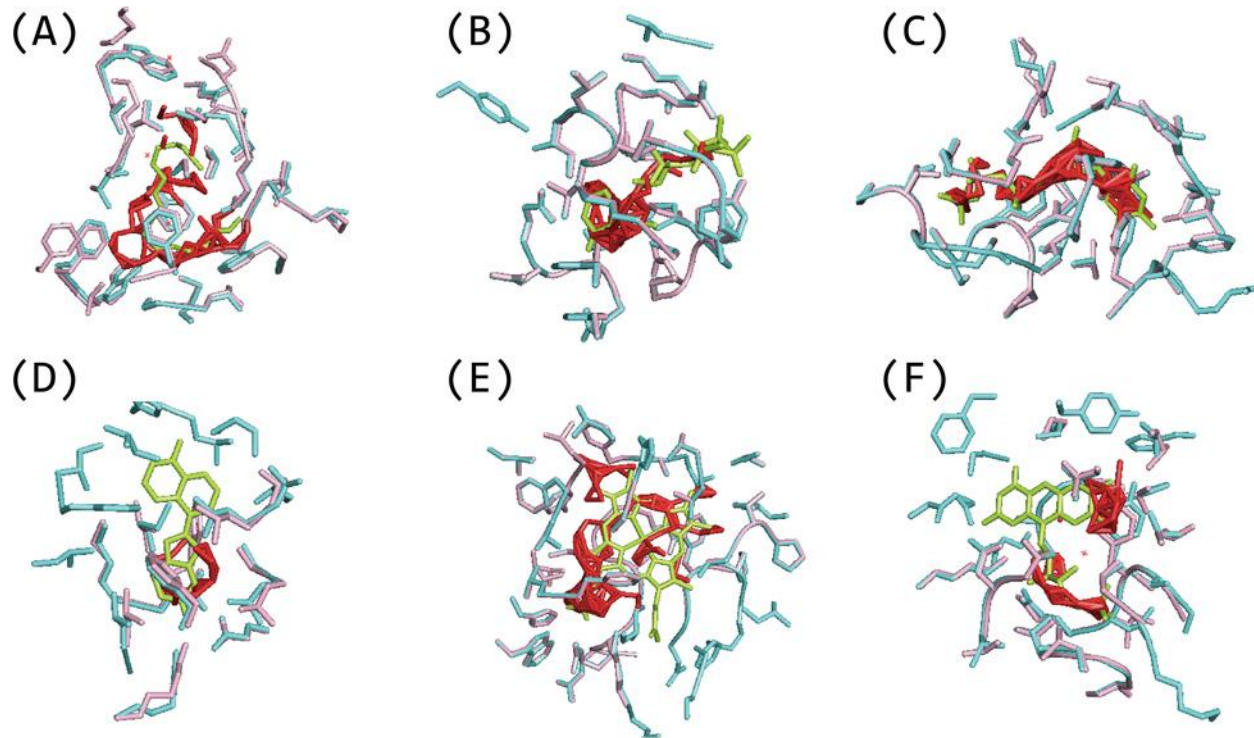
- § 9種類の異なった特徴空間を用意
- § 各々にSketchSortをかける: 全部で19時間
- § コサインの閾値0.85で15,966,080個のペアを抽出



ペアの内訳

- § (A) 空のポケットと空のポケットのペア: 8,219,689
- § (B) CATH_codeのアノテーションが付いていないペア:
4,378,432
- § (C) Homologousなペア{両端が同一のCATH_code}:
1,762,036
- § (D) Analogousなペア{両端が異なるCATH_code }:
1,605,923

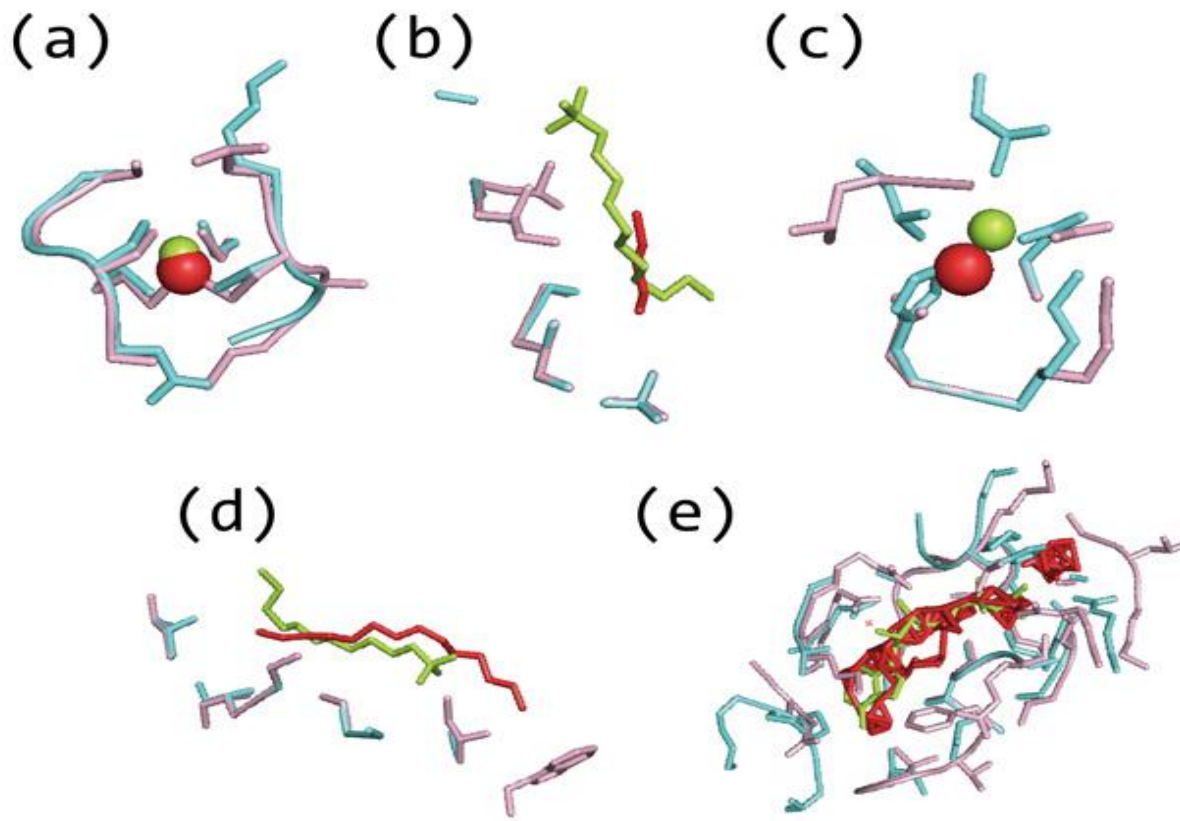
Homologous Sites



青と紫：重ね合わさった二つのタンパク質のリガンド結合部位
緑：青のリガンド結合部位へ結合しているリガンド
赤：リガンド結合部位予測プログラム (Ghecom) によって予測されたポケット

Analogous Sites

- § 異なるリガンドが結合するサイトが類似している例
- § 配列は異なる



終わりに

- § 高速な全ペア類似度検索法を提案
- § 簡単に数千万点のデータが扱える
- § 様々な機械学習における応用が考えられる
- § 今後の課題
 - § ペアを列挙したとしても、メモリに保持できない
オンライン学習？

- § コードはこちら <http://code.google.com/p/sketchsort/>