<table>
<thead>
<tr>
<th>Title</th>
<th>SLIDESORT: Developing an exact method to find similar pairs with small edit-distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Shimizu, Kana; Tsuda, Koji</td>
</tr>
<tr>
<td>Citation</td>
<td>2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録. p.376.</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2011-06</td>
</tr>
<tr>
<td>Doc URL</td>
<td><a href="http://hdl.handle.net/2115/48368">http://hdl.handle.net/2115/48368</a></td>
</tr>
<tr>
<td>Type</td>
<td>conference presentation</td>
</tr>
<tr>
<td>Note</td>
<td>ERATO 湊離散構造処理系プロジェクトシンポジウム（第1回）第9回情報科学技術フォーラムイベント企画セッション 2010年9月8日（水）九州大学伊都キャンパス</td>
</tr>
</tbody>
</table>

04.FIT_tsuda.pdf

Hokkaido University Collection of Scholarly and Academic Papers: HUSCAP
**SLIDESORT:** Developing an exact method to find similar pairs with small edit-distance

Kana Shimizu, Koji Tsuda
Computational Biology Research Center, National Institute of Advanced Science and Technology, JAPAN
Contact: shimizu-kana@aist.go.jp

**ABSTRACT:** In this study, we propose an exact method that finds all similar pairs from a string pool in terms of edit distance without any duplication on small memory in $O(N)$. Using an efficient pattern growth algorithm, our method discovers chains of common $k$-mers to narrow down the search. Compared to existing methods based on single $k$-mer, our method is more effective in reducing the number of edit distance calculations. In large short read datasets, our method was 20-3000 times faster than the state-of-the-art method, BWA.

**Motivation**

Due to the dramatic improvement of DNA sequencing, it is required to evaluate sequence similarities among a huge amount of fragment sequences such as short reads. Finding similar pairs from a string pool is an important first step for many biological sequence analyses. In this study, we would like to address the problem of discovering all neighbor pairs in a large string pool in terms of edit distance. This problem is conventionally called "all pairs similarity search," that will help ... 

- **Shot read analysis**
  - preprocess of de nove assembly
  - Clustering reads (e.g. online tree construction, greedy clustering etc.)
  - genome analyses
  - Repeat analyses
  - Sequence pattern analyses etc...

**Method**

**Strategy of SLIDESORT**

Like other methods based on $k$-mer matching, SLIDESORT finds common substrings and verifies the matches. The key idea of SLIDESORT is to find longer common substrings to narrow down search space. We developed an efficient pattern growth algorithm inspired by multiple sorting algorithm (Uno 2008 PAKDD).

**Theory of SLIDESORT**

How SLIDESORT can find longer common substrings? Given two sequences that are divided into $k+1$ blocks, if edit distance of the two sequences is at most $c$, there exist at least $k$ blocks that are the same string patterns between the two within slide width $d/2$. Based on this theory, SLIDESORT regards concatenation of $k$-blocks as a common substring.

**Algorithm of SLIDESORT**

A pattern corresponds to a sequence of substrings. The space of all patterns is organized as a tree and systematically traversed. SLIDESORT uses radix sort to find equivalent strings in pattern growth.

**Experiments**

Comparison of number of candidate pairs. The evaluations were done on 100,000 short reads. The proposed method was examined with $k=1,...,5$. 'Neighbor pairs' represent the actual number of neighbor pairs in data. 'k-mer/nonredundant' and 'k-mer/redundant' represent two variants of the single seed method.

**Application**

One of an application of SLIDE-SORT is constructing a tree. The right graph is a minimum spanning forest of 112,995 short reads that was generated by processing a pair of sequences (an edge) one-vs-one. Most of the case, computation time of each process is smaller than interval time to find next pair.

**SLIDESORT is available from**

[www.cbrc.jp/~shimizu/slidesort/](http://www.cbrc.jp/~shimizu/slidesort/)