Algorithms for clustering highly conserved phylogenetic markers

A prospectus submitted in partial fulfillment of the degree of Doctor of Philosophy

Preliminary Oral Examination for
Christopher Mihcael Hill

Advisor:
Dr. Mihai Pop

Committee Members:
Dr. Atif Memon
Dr. Héctor Corrada Bravo

Department of Computer Science
University of Maryland, College Park, MD 20742
April 9th, 2014

Abstract

Insert abstract here.

Contents

List of Figures

1 Introduction

Microbes play a huge role in every aspect of our life. **digestion, disease, ...**.

Advances in sequencing technology allows us to capture a snapshot of the microbial community/better understand the role microbes play in our life.

Typical analyses involve clustering the sequences into similar operational taxonmic units (OTUs).

Clustering is a widely-studied problem in computer science.

Computing the pair-wise distance are expensive and impractical in practice.

Introduction here[21]. Wednesday 9th April, 2014

2 Related Work

The traditional approach for clustering 16S rRNA sequences has involved the use of a multiple sequence alignment (MSA) for all sequences.

Optimal multiple sequence alignment between a collection of sequences can be done with a dynamic programming algorithm.

Impractical for the large number of sequences. 454 technologies can produce millions of sequences.

Greedy methods can be used to iteratively build a rooted tree. Then a specific cutoff can be given to split the tree into clusters.

Alternatives include building a distance matrix between each pair of sequences. Once the distance matrix is built, clustering is typically done via hierarchical methods. Agglomerative methods involve a bottom-up approach, where each sequence starts as its own cluster then iteratively merged with other clusters. Divisive methods, on the other hand, work from a top-down approach, where all sequences belong to a single cluster then are iteratively split into smaller clusters.

While hierarchical methods requires less work than the multiple sequence alignment, calculating the pairwise distances generally requires cubic work, making it impracticable for a large number of sequences.

An alternative approach for clustering sequences involves selecting a sequence to become the cluster center. The center is then used to recruit the remaining sequences that fall within some given *distance* threshold to the center. The distance between two sequences can include edit distance (also called Levenshtein distance), k-mer distance, similarity, identity, ...

2.1 Greedy clustering paradigm

A commonly-used clustering paradigm for sequence clustering is to iteratively select a sequence to serve as a cluster center and then recruit all remaining sequences that fall within some given distance of the cluster center. This process is repeated until no more sequences remain or the predetermined list of centers is exhausted.

2.2 Cluster center selection

There are three strategies used for selecting potential cluster centers. In ***de novo*** clustering, centers are selected *only* from the set of input sequences. Strategies for selecting potential centers are described below. In **closed-reference** clustering, a list of predetermined centers is given, such as a collection of previously discovered OTUs[7, 27]. In **open-reference** clustering, sequences are first recruited to a list of predetermined cluster centers. Afterwards, the centers are chosen by the *de novo* methods.

Selecting which sequence to use as the cluster center is a difficult problem. One strategy is to select the remaining sequence with the longest length. An intuitive argument is that a longer sequence would more likely recruit shorter sequences. However, the more rigorous argument is that selecting the longest remaining sequence allows you certain mathematical guarantees when aligning and recruiting shorter sequences. For example, given 3 sequences: A, B, and C. If the length of A is less than B and C and we know the distance between A and B and A C, we can not say anything for certain about the distance between A and C. We do not have any data about the overlapping regions between B and C.

INSERT FIGURE OF THIS.

This center selection strategy is used by CD-HIT[23], DNACLUST[13].

Frequency of k-mers.

2.3 Sequence recruitment

A key part of any clustering algorithm is how the distance between two objects is computed. In the case of sequence clustering, we need to calculate the distance between two strings. Commonly-used distance metrics include:

- Edit (Levenshtein) distance
- K-mer
- Identity

2.3.1 Edit distance

The *edit distance* between a text string $t = t_1t_2...t_n$ and pattern string $p = p_1p_2...p_m$ is the minimum number of differences between them such that one string can be transformed into the other. A difference is one of the following:

1. A character of the text corresponds to a different character of the pattern.
2. A character of the text corresponds to no character (a gap) in the pattern.
3. A character of the pattern corresponds to no character (a gap) in the text.

Algorithms for $k - mismatches$ only satisfies differences of type 1.

We use the NeedlemanWunsch**NEED CITATION** dynamic programming algorithm to calculate the edit distance between two strings.

Algorithm to solve k-differences

$$M[i,j] = \begin{cases} M[i-1,j]+1 \\ M[i,j-1]+1 \\ M[i-1,j-1] + \begin{cases} 0, & \text{if } t[i] == p_j \\ 1, & \text{else} \end{cases} \end{cases} \tag{1}$$

Algorithm 1 Compute Edit Distance between two strings. $O(nm)$ work.

```
procedure ComputeEditDistance($a, b$)
    $n \leftarrow |a|$
    $m \leftarrow |b|$
    for $i = 0..n$ do
        $M(i,0) \leftarrow i$
    for $i = 0..m$ do
        $M(0,i) \leftarrow i$
    for $i = 1..n$ do
        for $j = 1..m$ do
            $row \leftarrow M(i-1,j)+1$                ▷ Number of edits with a gap inserted into $b$
            $col \leftarrow M(i,j-1)+1$                ▷ Number of edits with gap inserted into $a$
            $diag \leftarrow M(i-1,j-1)$               ▷ Number of edits with matching characters $a_i$ and $b_j$
            if $a_i \neq b_j$ then $diag \leftarrow diag + 1$
            $M(i,j) \leftarrow \min(row, col, diag)$
    return $M(n,m)$
```

If we want to find a global alignment within k-differences, we only need to worry about a $2k+1$ band along the diagonal. This reduces the amount of work from $O(n^2)$ to $O(nk)$. However, this assumes we are aligning the two sequences end-to-end.

3 Preliminary Work

3.1 Parallelizing sequence recruitment to a cluster center

Due to the large scale of sequencing data produced, clustering tools must utilize multiple processors to process the data in a timely manner.

Here, we present two parallel approaches for recruiting sequences to a cluster center. The first approach (naïve) is based on evenly partitioning the sequences among the processors. The second approach (work-based) involves partitioning the sequences based on the potential work that needs to be done when calculating the edit distance to the center. If the sequences are stored in a trie-like data structure, then it is beneficial to partition highly similar sequences together despite potentially assigning an uneven number of sequences to each processor.

We implement these parallel approaches in DNACLUST[13] show the speed-ups when clustering tens of millions of 16S rRNA sequences.

3.1.1 Naïve parallelization strategy

The second step of DNACLUST's algorithm involves recruiting all sequences that lie within a given distance of the current cluster center. Given p processors, we can evenly partition the database into p chunks such that each processor can calculate edit distance independently in parallel.

INSERT FIGURE

3.1.2 Work-based parallelization strategy

Since we reuse part of the dynamic programming matrix, evenly partitioning the sequences may split highly similar sequences into separate threads.

Instead of evenly splitting the number of sequences between threads, we can evenly split the amount of potential work (characters we need to examine in the trie).

This is done by counting the total number of characters on the edges in the trie (trie length) and dividing by the number of threads.

INSERT FIGURE

3.2 Efficient data structures for edit distance computation

Currently, we require $O(n^2)$ work to calculate the edit distance between two sequences. This cost is reduced to $O(nk)$ in the specific case of globally aligning two sequences within k edits.

In this section, we describe how how to further improve the runtime to $O(k^2)$ in the case of global alignment and $O(nk)$ for semi-global alignment.

3.2.1 Alternative representation of the dynamic programming matrix

When calculating the edit distance between sequences $A = a_1a_2..a_n$ and $B = b_1b_2..b_m$, entry (i, j) in matrix M represents the minimum edit distance between prefixes $A_{1,i}$ and $B_{1,j}$ (Algorithm 1).

An alternative way to view this alignment is to consider each diagonal d and edit e of M. The d-diagonal is equal to $i - j$. Let C be another matrix where entry (i, j) now refers to the furthest reaching row in M of diagonal d that contains e edits.

-Show that the matrix solves the problem.

-Show that we can write a recurrence to solve this problem.

-Show the algorithm pseudocode.

-LCP can be answered in constant time via a suffix tree (gusfield).

-Compare the two approaches in terms of cells of the dp matrix that need to be computed.

-More difficult to exploit the sorted sequences (possible future work).

-Future work includes actually implementing the O(1) LCP extension.

3.2.2 Suffix tree cluster center representation

3.3 Handling ambiguous reads

When a sequence is being recruited by a center, it is possible that this sequence is within some distance from another potential center. Henceforth, we refer to a sequence that lies within a given distance from multiple centers as *ambiguous*. Currently in DNACLUST, an ambiguous sequence is recruited by the first center that encounters. Depending on the number of ambiguous of sequences, this may affect the resulting cluster abundances. Furthermore, downstream analyses on analyzing these count matrices (such as detecting differentially abundant OTUs) could lead to incorrect results.

Here, we describe different methods for assigning ambiguous reads.

The first way is to simply discard any ambiguous reads and only consider reads that can be uniquely aligned to a single center.

Another way is to randomly assign the ambiguous read to the set of potential centers.

Similarly, instead of randomly assigning the reads, we can assign a fractional count to each center.

Lastly, we can assign a read based on the proportion of uniquely aligned reads to the center. In other words, if a read can align equally well to two different centers, but one center contains uniquely aligned reads and the other contains none, then it is more probable that the read came from the first center.

INSERT FIGURE

4 Proposed Work

4.1 Farrah's algorithm for SIMD edit distance computation

4.2 Streaming clustering

5 Timeline

Item 1	2 months
Item 2	3 months
Item 3	4-5 months
Item 4	2-3 months
TOTAL	11-13 months

Paper deadline goals:

- Conference, Month Year: Project 1
- Conference, Month Year: Project 2
- Conference, Month Year: Project 3

6 Conclusion

Insert conclusion here.

A Reading List

A.1 Area 1

1) Citation 1.

2) Citation 2.

3) Citation 3.

4) Citation 4.

5) Citation 5.

A.2 Area 2

1) Citation 1.

2) Citation 2.

3) Citation 3.

4) Citation 4.

5) Citation 5.

A.3 Area 3

1) Citation 1.

2) Citation 2.

3) Citation 3.

4) Citation 4.

5) Citation 5.

B Appendix A

References

[1] Mohamed Ibrahim Abouelhoda, Stefan Kurtz, and Enno Ohlebusch. "Replacing suffix trees with enhanced suffix arrays". In: *Journal of Discrete Algorithms* 2.1 (Mar. 2004), pp. 53–86. ISSN: 1570-8667. DOI: 10.1016/S1570-8667(03)00065-0. URL: http://www.sciencedirect.com/science/article/pii/S1570866703000650 (visited on 03/24/2014).

[2] Stephen F. Altschul et al. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs". In: *Nucleic Acids Research* 25.17 (Sept. 1, 1997). PMID: 9254694, pp. 3389–3402. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/25.17.3389. URL: http://nar.oxfordjournals.org/content/25/17/3389 (visited on 03/24/2014).

[3] Ergude Bao et al. "SEED: efficient clustering of next-generation sequences". In: *Bioinformatics* 27.18 (Sept. 15, 2011). PMID: 21810899, pp. 2502–2509. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btr447. URL: http://bioinformatics.oxfordjournals.org/content/27/18/2502 (visited on 03/24/2014).

[4] Jon L. Bentley and Robert Sedgewick. "Fast Algorithms for Sorting and Searching Strings". In: *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '97. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997, 360369. ISBN: 0-89871-390-0. URL: http://dl.acm.org/citation.cfm?id=314161.314321 (visited on 03/24/2014).

[5] Yunpeng Cai and Yijun Sun. "ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time". In: *Nucleic Acids Research* 39.14 (Aug. 1, 2011). PMID: 21596775, e95–e95. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkr349. URL: http://nar.oxfordjournals.org/content/39/14/e95 (visited on 03/24/2014).

[6] J. Gregory Caporaso et al. "QIIME allows analysis of high-throughput community sequencing data". In: *Nature Methods* 7.5 (May 2010), pp. 335–336. ISSN: 1548-7091. DOI: 10.1038/nmeth.f.303. URL: http://www.nature.com/nmeth/journal/v7/n5/full/nmeth.f.303.html (visited on 03/24/2014).

[7] T. Z. DeSantis et al. "Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB". In: *Applied and Environmental Microbiology* 72.7 (July 1, 2006). PMID: 16820507, pp. 5069–5072. ISSN: 0099-2240, 1098-5336. DOI: 10.1128/AEM.03006-05. URL: http://aem.asm.org/content/72/7/5069 (visited on 04/07/2014).

[8] Robert C. Edgar. "Search and clustering orders of magnitude faster than BLAST". In: *Bioinformatics* 26.19 (Oct. 1, 2010). PMID: 20709691, pp. 2460–2461. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/

bioinformatics/btq461. URL: http://bioinformatics.oxfordjournals.org/content/26/19/2460 (visited on 03/24/2014).

[9] Michael Farrar. “Striped SmithWaterman speeds database searches six times over other SIMD implementations”. In: *Bioinformatics* 23.2 (2007). PMID: 17110365, pp. 156–161. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btl582. URL: http://bioinformatics.oxfordjournals.org/content/23/2/156 (visited on 03/24/2014).

[10] Daniel Fasulo. *An Analysis of Recent Work on Clustering Algorithms.* 1999.

[11] Limin Fu et al. “CD-HIT: accelerated for clustering the next-generation sequencing data”. In: *Bioinformatics* 28.23 (Dec. 1, 2012). PMID: 23060610, pp. 3150–3152. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/bts565. URL: http://bioinformatics.oxfordjournals.org/content/28/23/3150 (visited on 03/24/2014).

[12] Zvi Galil and Kunsoo Park. “An Improved Algorithm for Approximate String Matching”. In: *SIAM J. Comput.* 19.6 (Nov. 1990), 989999. ISSN: 0097-5397. DOI: 10.1137/0219067. URL: http://dx.doi.org/10.1137/0219067 (visited on 03/24/2014).

[13] Mohammadreza Ghodsi, Bo Liu, and Mihai Pop. “DNACLUST: accurate and efficient clustering of phylogenetic marker genes”. In: *BMC Bioinformatics* 12.1 (June 30, 2011). PMID: 21718538, p. 271. ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-271. URL: http://www.biomedcentral.com/1471-2105/12/271/abstract (visited on 03/24/2014).

[14] Steven R. Gill et al. “Metagenomic Analysis of the Human Distal Gut Microbiome”. In: *Science* 312.5778 (June 2, 2006). PMID: 16741115, pp. 1355–1359. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1124234. URL: http://www.sciencemag.org/content/312/5778/1355 (visited on 03/24/2014).

[15] Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, May 28, 1997. 556 pp. ISBN: 9780521585194.

[16] Julia Handl, Joshua Knowles, and Douglas B. Kell. “Computational cluster validation in post-genomic data analysis”. In: *Bioinformatics* 21.15 (Aug. 1, 2005). PMID: 15914541, pp. 3201–3212. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/bti517. URL: http://bioinformatics.oxfordjournals.org/content/21/15/3201 (visited on 03/24/2014).

[17] Susan M Huse et al. “Ironing out the wrinkles in the rare biosphere through improved OTU clustering”. In: *Environmental Microbiology* 12.7 (July 2010). PMID: 20236171 PMCID: PMC2909393, pp. 1889–

1898. ISSN: 1462-2912. DOI: 10.1111/j.1462-2920.2010.02193.x. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2909393/ (visited on 03/24/2014).

[18] Victor Kunin et al. "Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates". In: *Environmental microbiology* 12.1 (2010). PMID: 19725865, pp. 118–123. ISSN: 1462-2920. DOI: 10.1111/j.1462-2920.2009.02051.x.

[19] G M Landau and U Vishkin. "Introducing Efficient Parallelism into Approximate String Matching and a New Serial Algorithm". In: *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*. STOC '86. New York, NY, USA: ACM, 1986, 220230. ISBN: 0-89791-193-8. DOI: 10.1145/12130.12152. URL: http://doi.acm.org/10.1145/12130.12152 (visited on 03/24/2014).

[20] Gad M Landau and Uzi Vishkin. "Fast parallel and serial approximate string matching". In: *Journal of Algorithms* 10.2 (June 1989), pp. 157–169. ISSN: 0196-6774. DOI: 10.1016/0196-6774(89)90010-2. URL: http://www.sciencedirect.com/science/article/pii/0196677489900102 (visited on 03/24/2014).

[21] Gad M. Landau and Uzi Vishkin. "Fast string matching with k differences". In: *Journal of Computer and System Sciences* 37.1 (Aug. 1988), pp. 63–78. ISSN: 0022-0000. DOI: 10.1016/0022-0000(88)90045-1. URL: http://www.sciencedirect.com/science/article/pii/0022000088900451 (visited on 03/24/2014).

[22] Weizhong Li and Adam Godzik. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences". In: *Bioinformatics* 22.13 (July 1, 2006). PMID: 16731699, pp. 1658–1659. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btl158. URL: http://bioinformatics.oxfordjournals.org/content/22/13/1658 (visited on 03/24/2014).

[23] Weizhong Li, Lukasz Jaroszewski, and Adam Godzik. "Clustering of highly homologous sequences to reduce the size of large protein databases". In: *Bioinformatics* 17.3 (Mar. 1, 2001). PMID: 11294794, pp. 282–283. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/17.3.282. URL: http://bioinformatics.oxfordjournals.org/content/17/3/282 (visited on 03/24/2014).

[24] Weizhong Li et al. "Ultrafast clustering algorithms for metagenomic sequence analysis". In: *Briefings in Bioinformatics* 13.6 (Nov. 1, 2012). PMID: 22772836, pp. 656–668. ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbs035. URL: http://bib.oxfordjournals.org/content/13/6/656 (visited on 03/24/2014).

[25] Udi Manber and Gene Myers. "Suffix Arrays: A New Method for On-Line String Searches". In: *SIAM Journal on Computing* 22.5 (Oct. 1993), pp. 935–948. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/0222058. URL: http://epubs.siam.org/doi/abs/10.1137/0222058 (visited on 03/24/2014).

[26] Sarah P. Preheim et al. "Distribution-Based Clustering: Using Ecology To Refine the Operational Taxonomic Unit". In: *Applied and Environmental Microbiology* 79.21 (Nov. 1, 2013). PMID: 23974136, pp. 6593–6603. ISSN: 0099-2240, 1098-5336. DOI: 10.1128/AEM.00342-13. URL: http://aem.asm.org/content/79/21/6593 (visited on 03/24/2014).

[27] C. Quast et al. "The SILVA ribosomal RNA gene database project: improved data processing and web-based tools". In: *Nucleic Acids Research* 41 (D1 2013), pp. D590–D596. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gks1219. URL: http://nar.oxfordjournals.org/content/41/D1/D590 (visited on 04/07/2014).

[28] Christopher Quince et al. "Accurate determination of microbial diversity from 454 pyrosequencing data". In: *Nature Methods* 6.9 (Sept. 2009), pp. 639–641. ISSN: 1548-7091. DOI: 10.1038/nmeth.1361. URL: http://www.nature.com/nmeth/journal/v6/n9/full/nmeth.1361.html (visited on 03/24/2014).

[29] Jens Reeder and Rob Knight. "Rapid denoising of pyrosequencing amplicon data: exploiting the rank-abundance distribution". In: *Nature methods* 7.9 (Sept. 2010). PMID: 20805793 PMCID: PMC2945879, pp. 668–669. ISSN: 1548-7091. DOI: 10.1038/nmeth0910-668b. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2945879/ (visited on 03/24/2014).

[30] Joo F. Matias Rodrigues and Christian von Mering. "HPC-CLUST: distributed hierarchical clustering for large sets of nucleotide sequences". In: *Bioinformatics* 30.2 (2014). PMID: 24215029, pp. 287–288. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btt657. URL: http://bioinformatics.oxfordjournals.org/content/30/2/287 (visited on 03/24/2014).

[31] Torbjrn Rognes. "Faster Smith-Waterman database searches with inter-sequence SIMD parallelisation". In: *BMC Bioinformatics* 12.1 (June 1, 2011). PMID: 21631914, p. 221. ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-221. URL: http://www.biomedcentral.com/1471-2105/12/221/abstract (visited on 03/24/2014).

[32] Patrick D. Schloss et al. "Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities". In: *Applied and Environmental Microbiology* 75.23 (Dec. 1, 2009). PMID: 19801464, pp. 7537–7541. ISSN: 0099-2240, 1098-5336. DOI: 10.1128/AEM.01541-09. URL: http://aem.asm.org/content/75/23/7537 (visited on 03/24/2014).

[33] E. Ukkonen. “On-line construction of suffix trees”. In: *Algorithmica* 14.3 (Sept. 1, 1995), pp. 249–260. ISSN: 0178-4617, 1432-0541. DOI: 10.1007/BF01206331. URL: http://link.springer.com/article/10.1007/BF01206331 (visited on 03/24/2014).

[34] Esko Ukkonen. “Approximate string-matching over suffix trees”. In: *Combinatorial Pattern Matching.* Ed. by Alberto Apostolico et al. Lecture Notes in Computer Science 684. Springer Berlin Heidelberg, 1993, pp. 228–242. ISBN: 978-3-540-56764-6, 978-3-540-47732-7. URL: http://link.springer.com/chapter/10.1007/BFb0029808 (visited on 03/24/2014).

[35] J. Craig Venter et al. “Environmental Genome Shotgun Sequencing of the Sargasso Sea”. In: *Science* 304.5667 (Apr. 2, 2004). PMID: 15001713, pp. 66–74. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1093857. URL: http://www.sciencemag.org/content/304/5667/66 (visited on 03/24/2014).

[36] John C. Wooley, Adam Godzik, and Iddo Friedberg. “A Primer on Metagenomics”. In: *PLoS Comput Biol* 6.2 (Feb. 26, 2010), e1000667. DOI: 10.1371/journal.pcbi.1000667. URL: http://dx.doi.org/10.1371/journal.pcbi.1000667 (visited on 03/24/2014).

[37] Yuzhen Ye. “Identification and quantification of abundant species from pyrosequences of 16S rRNA by consensus alignment”. In: *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Dec. 2010, pp. 153–157. DOI: 10.1109/BIBM.2010.5706555.

[38] Zejun Zheng, Stefan Kramer, and Bertil Schmidt. “DySC: software for greedy clustering of 16S rRNA reads”. In: *Bioinformatics* 28.16 (Aug. 15, 2012). PMID: 22730435, pp. 2182–2183. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/bts355. URL: http://bioinformatics.oxfordjournals.org/content/28/16/2182 (visited on 03/24/2014).