PennState

Paul Medvedev
Professor
Department of Computer Science and Engineering
Department of Biochemistry and Molecular Biology
The Pennsylvania State University

506B Wartik Lab
University Park, PA 16802-5807
pzm11@psu.edu
814-865-0193

Report of Ondřej Sladký's bachelor thesis titled
"Masked superstrings for efficient k-mer set representation and indexing"

June 17, 2023

This bachelor thesis introduces the concept of masked superstrings to the field of k-mer set representations of bioinformatics data. Given a set of fixed length strings (called k-mers), the goal is to represent this set in a way the minimizes the total space taken and allows for fast membership queries. A popular previous approach is to represent the set by a set of longer strings such that a k-mer is in the set of strings if and only if it is in the original set. This thesis generalizes this approach, and others like it, under a unified masked superstring framework. It combines theoretical results about this new framework together with a implementation and experimental results.

The quality of the bachelor thesis is outstanding and on par with an excellent MSc thesis. I recommend its acceptance without any reservations. The idea of masked superstrings will move the field forward and I believe will be adopted by the research community. The thesis also has a nice blend of theory and practice – theoretically driven results that perform well in practice. The thesis is for the most part well written as well.

Below I will list a few comments and a question.

Questions

1. What could be possible data-driven ways to choose between the various mask-optimization strategies described in Section 3.2? By data-driven, I mean an algorithm that would take the dataset for which the masks will be optimized and quickly make an educated guess as to which optimization strategy would result in the least space. Would your approach have any provable guarantees? What types of datasets would your heuristic work well and for which would it work badly?

Comments

1. Theorem 2 is tight up to a constant factor, as the thesis states; however, the constant factor is 2, which is significant. For example, if K is the set of k-mers from the human genome, then Theorem 2 only applies for k of around 32 or more. But in some applications k could be much lower, e.g. 21. My sense is that proof could be strengthened to remove this constant factor.
2. On a related note, the thesis misses an important citation to the PhD thesis of John Kececioglu.[1] In particular, the RECONSTRUCT problem with epsilon = 0 is similar to the problem considered in the thesis, with the difference being that RECONSTRUCT has no length constraint. Importantly, RECONSTRUCT also accounts for reverse

[1] Available for PDF download here: https://repository.arizona.edu/handle/10150/185673?show=full

complements. Could elements of the proof used for Theorem 1.1 of Kececioglu's thesis be useful in strengthening the proof of Theorem 2?

3. The thesis is missing important citations to work on variable order de Bruijn graphs (e.g. the first paper in this line of work is "Variable-Order de Bruijn Graphs", appearing in DCC 2015). Variable order de Bruijn graphs seem very relevant to constructing superstrings when allowing overlaps of lengths smaller than k-1.
4. It seems to me that the description of the Global Greedy Algorithm in Section 3.1.3 could be simplified by using the bidirected de Bruijn graph framework (introduced in Medvedev and Brudno WABI 2007 but better described in Rahman and Medvedev, RECOMB/Genome Research 2022).
5. On page 2, the authors mention in footnote 1 that this approach could create a k-mer and its reverse complement in the resulting string. Has the notion of transitive edge reduction been considered (Myers, ECCB 2005)? It might be useful to avoid this situation and could possible improve speed/memory by reducing the number of edges in the overlap graph.
6. The approach of Section 5 is poorly motivated. The section jumps into defining the notion f-MS but without explaining what the goal is. This makes it difficult to understand the value of the contribution. For example, my initial thought was that the intention of these functions was to make the mask more compressible; further on in the text, I understood that this is not actually the intention. Then, what is being optimized with this approach? Let me describe what my best interpretation was so you can see where the reader may have misunderstandings and/or what holes need to be filled in the text.

   The setting is that you have multiple datasets represented with the FMS-index and you would like to generate a FMS-type-index for a new dataset that is defined as function of set-theoretic operations on the original datasets. You wish to make use of the original FMS-indices, rather than constructing a new one. A naïve way to do this is to store a tree of the operations to be performed (e.g. (A union B) intersect C could be the operations on three datasets). Then a query Q would be performed on the FMS-indices of A, B, and C separately and the appropriate logic will be applied to the result. What is being proposed is to improve on this naïve approach. In particular, you do not want to store the history of operations (though its not clear to me what is so bad about doing that). Instead, you want to encode the history of the operations in a function to interpret the occurrence functions.

7. It could also be noted at the start of Section 5 when the suggested approach is useful and when it is not. For example, if I want to enumerate the set K from the f-MS, using a function besides "or" would, I imagine, make the algorithm memory inefficient (because one would need to keep track of the occurrence lists somehow as one is scanning through the superstring).
8. On a related note, a major limitation of this approach is that the space taken by a merge is the sum of the prior space usages. This can be drastically inefficient when, for example, taking the union between sets of k-mers with large overlaps.
9. An additional list of minor comments appears below my signature.

The comments above focus on ways that the work could be improved and should not in any way take away from what is an outstanding piece of research. I enjoyed reading the thesis and wish Mr. Sladký continued success in his research endeavors.

Sincerely,

Medvedev

Paul Medvedev
Professor
Department of Computer Science and Engineering
Department of Biochemistry and Molecular Biology
Director, Center for Computational Biology and Bioinformatics
The Pennsylvania State University

PS: Below are a list of some possible typos I found and possible places to improve the text.

1. In section 1.3, the statement "k-mer sets are not independent" is confusing. I believe what is meant is that the k-mers within a set are not indendent.
2. In section 1.3, the statement "Since unitigs….are quite efficient" is confusing. In what sense are unitigs efficient? I imagine you are referring to space-efficiency rather than construction or query time efficiency, but this can be clarified.
3. In section 1.5.1, the statement "….EVERY state has also a fail function f…" is confusing, because there might not exist a proper suffix of s that is also a valid state. For example, in Figure 1.2, there is no arrow going out of the node AC.
4. The thesis cites CLJ+14 for the term unitig in several places. However, CLJ+14 did not introduce the term. It would be more appropriate to either cite the correct paper (though I am not sure myself what that paper is) or simply not giving a citation and treating it as a "folklore" term.
5. Definition 4: it wasn't immediately clear from the definition that the lambda sequence is ordered in increasing order of i. This might be nice to mention.
6. Table 5.1. It would be useful to use the caption to clarify the norm notation used in the table. If my guess is correct, |Lambda|_0 refer the L0 norm, etc…
7. Observation 7: "functions" should be "function"