Response to the reviewers – *Indexing labeled sequences*

Dear Editor,

We thank you, and the reviewers, for your positive appreciations on our manuscript `#CS-2017:09:20302:1:0:NEW` entitled *"Indexing labeled sequences"*.

Reviewer 1

There is [...] an even more broad generalization of adding an XML tree on top of a sequence [1], which encompasses linear labelling as a special case. That article is not targeted to a general audience, so reporting this special case with a tailored (and different) solution is completely desirable.

We appreciate this valuable comment. We integrated the reference at the end of the introduction to give a wider perspective.

Reviewer 2

Some of the bounds cited are not the best known, and several of the references are outdated.

We are not sure what bounds and references the reviewers had in mind. However we noticed that a 2016 paper by Munro *et al* improved the wavelet tree construction. We have integrated it to the paper.

It seems fairly easy to build an $O(n \log n)$-bit index such that, given a pattern P, a label x and a position i in P, we can reasonably quickly find all the positions where P occurs in T with its ith character labelled x. Offhand, however, I don't see how to reduce the space to $O(n \log \sigma)$. A solution to that problem might make the article more interesting for researchers familiar with pattern matching.

We are grateful to the reviewer for this interesting comment. We recall that our approach is in $O(n \log \sigma)$ bits. We added a discussion on how we could slightly modify our approach to search the label at another position than the first one of the occurrence (starting at lines 143 in the PDF).

Reviewer 3

The implementation is based on the SDSL library but important details, i.e. which SA sampling strategy was used, are missing in the current version of the article.

We discussed the sampling distance in the "Evaluation procedure" section. However the discussion was focused on the simulated datasets. Now we also comment on what the sampling strategy involves for the real dataset.

It is also unclear why the author opted for `rrr_vector` *to represent the bit vectors. It is expected that* `sd_vector` *is superior to* `rrr_vector` *for long labels.*

This is a very good point and we agree with the reviewer that `sd_vector` is more suitable with low proportion of 1s ($< 10\,\%$). We added a discussion on that matter at the end of the "Conclusions" section. However we do not expect the B_A or B_D bit vector to dominate the space consumption of our index but rather the FM-index and the wavelet tree. Therefore the gain in space consumption would be only small.

Other modifications

For the sake of clarity we introduced a new table (Table 1 in the manuscript) to summarize the time complexities of the queries for the two indexes we introduce as well as the baseline solution.

We also corrected several typos as pointed out by the reviewers. We would like to thank them again for their careful reading of the manuscript.

The authors