The student ID of the student whose paper you are grading *

24978168

Writeup

Readability and grammar of written report (5 points) *

	1	2	3	4	5	
Difficult to read and/or poor grammar	○	○	○	○	◉	Clearly written and excellent grammar

Level of written detail on comparison of R and C++ implementation and runtime (3 points) *

	0	1	2	3	
Did not write about a comparison of the R and C++ implementation	○	○	○	◉	Wrote a detailed comparison between the R and C++ implementations

R and C++ code

Review the code written by the author. If you aren't sure of the correctness of the implementation, that's fine, just give a grade and say so in the comments.

Correctly coded the parallelization of k-means and pairwise similarity in R/C++ (3 points) *

	0	1	2	3	
incorrect implementation	○	○	◉	○	seems correct to me

Comments on implementation of parallelization or the similarity measure?

The R implementation was not provided, but the C++ implementation looks correct.

Efficiency and practicality of R and C++ code (3 points) *

	1	2	3	
inneficient (e.g. repeated computations unnecessarily, saved objects unnecessarily, etc)	○	○	◉	very efficient and practical

Suggestions for improving *efficiency* of R and/or C++ code *

Perform the calculation of `corr` in log space to avoid overflow

Does the author satisfy the following code readability requirements? (3 points) *

- [x] Consistent spacing before and after variable assignment and addition symbols (" = ", " + "), and after commas (", ")
- [ ] No line of code exceeds 80 characters
- [x] Consistent variable naming (words always separated by one of "_" or ".")

Clarity of variable names (2 points) *

	0	1	2	
variable names are unclear and meaningless (eg `df`, `x`, `data2`, etc)	○	◉	○	variable names are helpful and unambiguous

Quality of code comments (2 points) *

	0	1	2	
there are almost no comments	○	○	◉	the comments explain clearly what is being done and why

Suggestions for improving *readability* of R code *

Fix the spelling errors in the comments, make sure the comments don't overflow 80 columns (I didn't think this was a requirement myself >.<)

Did the student provide all code necessary for recompiling their results AND report (note: you do not have to actually reproduce their report) (2 points) *

	0	1	2	
Incomplete code or no .Rnw/.Rmd file provided	○	◉	○	Everything was provided

Clarity of folder structure (2 points) *

	0	1	2	
The folder structure was very confusing	○	○	◉	It was clear what each file corresponded to and there were no surplus files floating around

Optional comments on folder structure and files provided (please provide comments if you docked points for any reason)

You could have provided the code for computing similarity in R as well as for generating the R vs C++ benchmarks reported in `kmeans_corr_results.txt`

Figures

Correctly produced Ben-Hur-type figures (3 points) *

	0	1	2	3	
Did not provide a figure like Ben-Hur	○	○	○	◉	Figures look correct

If the Ben-Hur figures do not look correct, what is wrong?

Quality of Ben-Hur Figure 3 replication figures (3 points) *

	0	1	2	3	
Did not provide a figure like Ben-Hur	○	○	○	◉	Provided clear and visually appealing figures

Discuss one (or more) things that you liked about the author's Ben-Hur figures *

Good use of ggplot!

Discuss one (or more) things that could be improved for the author's Ben-Hur figures *

In the legend of the CDF plots, use "number of clusters" instead of "k"

Justification of conclusions drawn from the Ben-Hur-type figures (3 points) *

	0	1	2	3	
Did not write about any conclusions drawn from the figures	○	○	○	◉	Clearly outlined interpretations of the figures and drew reasonable conclusions (e.g. found k = 3, or some other value, is the best and provides reasons why)

Comments on the conclusions and interpretations of the Ben-Hur type figures *

Both the histograms and CDF make this really clear!

Conclusion

Provide concluding comments

One or more things that you thought was well done overall *

Your code is really neat and modular, and your comments are very helpful.

One or more things that could be improved upon overall *

You could have included the code to reproduce the benchmarks.

Any other comments that you would like to add?