BITS :: Call for Abstracts 2022 - Oral communication

Type	Oral communication
Session	Metagenomics
Title	Refined Classification of Metagenomic Long reads with Overlap Graphs
All Authors	M.Luciani, M.Cavattoni, M.Comin
Affiliation	

Department of Information Engineering, University of Padova, Padova

Motivation

Current technologies allow the sequencing of microbial communities directly from the environment without prior culturing. The major problem when analyzing a metagenomic sample is to taxonomically annotate its reads to identify the species they contain.
Most of the methods currently available focus on the classification of reads using a set of reference genomes and their k-mers. While in terms of precision these methods have reached percentages of correctness close to perfection, in terms of recall (the actual number of classified reads) the performances fall at around 50%.
One of the reasons is the fact that the sequences in a sample can be very different from the corresponding reference genome.

Methods

To address this problem, in this paper we propose ClassGraph 2, a metagenomic taxonomy refinement tool that makes use of reads overlap information from the reads overlap graph, to refine the results of existing tools to classify unlabelled reads. ClassGraph 2 needs two types of input: one is the reads overlap graph and the other is the output of a binning tool. At this stage the graph is stored in a data structure, where each node/read is associated with a label given by the binning tool. The arcs in the graph are weighted based on the overlap of the two reads. Connected reads are more likely to be from the same species, thus we refine the node labels in the reads overlap graph. This procedure is performed to search for nodes that are mislabelled by the binning tool. The RefineLabel algorithm can eliminate incorrectly assigned labels, then a LabelPropagation algorithm expands the correct labels. The RefineLabel algorithm counts neighboring nodes with the same label as the node under examination. If the label of the node is different from most of the labels of neighboring nodes, then this label must be removed. In the label propagation phase each labeled node sends its label to its neighbor, along with the weight of the arc connecting the two nodes. The receiving node will choose its label maximizing the score of the associated arcs. This process is repeated until all nodes connected in the graph are labeled.

Results

We tested ClassGraph 2 on three simulated datasets of long reads, created using SimLoRD with 8, 20 and 50 species, and a real marine metagenome with 5000 species, from the CAMI2 challenge. We chose Kraken 2, which is the state of the art, for the taxonomic classification of reads. We compared the classification performance of Kraken 2 with ClassGraph and ClassGraph 2. Sensitivity, precision, F1-Score and PCC were used to assess the accuracy of the classifications. Instead, time and memory were used to assess the running costs of the tools.
From the results in the table it can be seen that after running ClassGraph the classification accuracy, in terms of F-measure, for all datasets increases slightly, mostly due to a better sensitivity. With ClassGraph 2 there is a further increase in the classification accuracy for all datasets, with a substantial increase of both sensitivity and precision.
From the results it can be observed that, although Kraken 2 is one of the best binning tools, it cannot classify all reads, in fact the sensitivity on the most complex datasets is 57%, and the precision ranges in [70%-80%]. With ClassGraph, and its Label Propagation algorithm, these labels can be expanded, increasing the number of classified reads, while preserving a similar precision. In ClassGraph 2, with the new Refine Label algorithm before the Label Propagation, the classification accuracy further increases. The performance of ClassGraph 2 confirms that some of the classifications produced by Kraken2 are incorrect. However, it is possible to recognise them and then expand only the correct ones, thus improving both sensitivity and precision.
It can also be seen that the execution times of Kraken 2 and ClassGraph 2 are of the same order of magnitude. However, the memory required to run ClassGraph 2 is less than that required to run Kraken 2.

Info

In the attach pdf you can find more information about the method and the experiments.
The tool is available at: https://github.com/MattiaLuciani/ClassGraph2

filename	abs.png

Figure

	Sim - 8	Sim - 20	Sim - 50	Marine
Kraken2	Sens: 0.765628 Prec: 0.861202 **F1: 0.810608** PCC: 0.996426 Time: 00:11:49 Memory: 47.45 GB	Sens: 0.629978 Prec: 0.776203 **F1: 0.695488** PCC: 0.923726 Time: 00:12:37 Memory: 47.47 GB	Sens: 0.570512 Prec: 0.704239 **F1: 0.630361** PCC: 0.946056 Time: 00:28:06 Memory: 47.53 GB	Sens: 0.577274 Prec: 0.806809 **F1: 0.673009** PCC: 0.989780 Time: 00:12:31 Memory: 47.43 GB
ClassGraph	Sens: 0.789763 Prec: 0.864731 **F1: 0.825549** PCC: 0.995107 Time: 00:01:07 Memory: 2.25 GB	Sens: 0.666136 Prec: 0.780303 **F1: 0.718714** PCC: 0.933724 Time: 00:04:26 Memory: 11.15 GB	Sens: 0.596122 Prec: 0.682097 **F1: 0.636218** PCC: 0.952594 Time: 00:10:58 Memory: 28.56 GB	Sens: 0.697363 Prec: 0.814654 **F1: 0.751459** PCC: 0.989609 Time: 00:14:00 Memory: 26.18 GB
ClassGraph 2.0	Sens: 0.993279 Prec: 0.994949 **F1: 0.994113** PCC: 0.999963 Time: 00:01:28 Memory: 2.72 GB	Sens: 0.869446 Prec: 0.961895 **F1: 0.913337** PCC: 0.977861 Time: 00:05:46 Memory: 13.22 GB	Sens: 0.769968 Prec: 0.827907 **F1: 0.797887** PCC: 0.983608 Time: 00:13:36 Memory: 33.91 GB	Sens: 0.795554 Prec: 0.909551 **F1: 0.848742** PCC: 0.989793 Time: 00:17:27 Memory: 29.01 GB

Availability	http://www.dei.unipd.it/~ciompin/ClassGraph2.pdf

Corresponding Author

Name, Surname	matteo, comin
Email	comin@dei.unipd.it
Submitted on	29.04.2022