master's project notes

Contents

1 Evaluation of NumPy vs CuPy in npstructures and BioNumpy

To initiate exploration of what CuPy does well in comparison to NumPy, I am carefully benchmarking each step of the k-mer counting pipeline, going from an input fasta file of 20 million reads to the finished counter object, in order to fully understand which parts of the npstructures and BioNumpy code is lacking relative performance using both NumPy and CuPy as the backend array module.

This pipeline, as is performed in the BioNumpy library, can be abstracted into n parts: 1) reading and formatting a chunk of reads from the input fasta file, resulting in a ragged-array containing a single read per row, where each base is represented as a single unsigned 8-bit integer. 2) Converting the chunk of reads into a compact bit-array representation, only using 2 bits per base. 3) retrieving each 31-mer from the bit-array and representing each 31-mer as a 64-bit integer. 4) ...

1.1 Reading fasta file and creating chunks of reads

Practically all of the time spent when preparing each chunk in the chunk generator is spent in OneLineBuffer's get_data() method.