Supplementary 1

Bloocoo: read error corrector

Bloocoo is a k-mers-spectrum based read error corrector developed with GATB. It relies on k-mers frequency to discriminate between correct solid k-mers and k-mers containing sequencing errors. This is a traditional approach used by many read correctors. Bloocooo distinguishes itself by requiring an order of magnitude less memory than other state-of-the art correctors, while still providing equivalent correction. This is achieved thanks to a constant-memory k-mers counting algorithm, and a bloom filter to store solid k-mers.

The read correction workflows consists in 3 main steps as follows:

- k-mers counting.
- Insertion of solid k-mers in a bloom filter.
- Multi-stage read correction

The first stage is the constant-memory k-mers counting step provided by the GATB library. It outputs on disk the list of solid k-mers, i.e. k-mers with a high enough abundance. The second step is the insertion of all solid k-mers in a bloom filter. This probabilistic data structure allows very low memory requirement implementation (about 11 bits per solid k-mers) the main operation required by the correction steps: know if a given k-mer from a read is solid or non solid. Most other software use a hash table to store solid k-mers and their abundance. Here the abundance information is lost, we only know if a k-mer abundance is above or below a given threshold. Moreover the bloom filter may introduce false positive solid k-mers.

The correction step is a multi-step approach largely similar to the Musket correction algorithm by Liu et al. All k-mers of a read are queried in the bloom filter and classified as solid or non-solid k-mers. Then errors are corrected through multiple iterations of two-sided conservative, one-sided aggressive and voting-based correction algorithm. To neutralize the effect of false positive solid k-mers coming from the bloom filter, errors are corrected only if several solid k-mers cover the corrected nucleotide, greatly reducing the risk that all of them are false-positives and induce false correction.

	Musket	Bloocoo
Time (s)	5330	1390
Memory (MB)	12190	740
Recall	90.92%	90.28%
Precision	97.86%	96.93%

Table 1: Indicative results on a simulated dataset with 1% error rate from human chr1, at 70x coverage. Times are real time reported by time command. Both software were running with 8 threads, on an Intel Xeon E5-2640

The correction procedure is parallelized by dispatching blocks of reads among several threads. The bloom filter used is a blocked bloom filter that greatly increases performance thanks to higher cache coherence.

References

Liu Y, Schroder J, Schmidt B (2013) Musket: a multistage k-mer spectrum-based error corrector for Illumina sequence data. Bioinformatics, 29(3):308-315