Blue Distribution 1.1.3
23rd April 2015

Changes from release 1.1.2

This release fixes a few small bugs and improves performance. The source code has been extensively restructured for maintainability – especially in the files found in the CommonCode directory.

Most of the performance improvements come from tuning the hash table implementations and binary file IO. The following table shows that Tessel and GenerateMerPairs are most improved by these changes.

Pseudomomas -t 8	Tessel	GMP	Blue	
1.1.2	77.1	83.5	163.0	323.6
1.1.3	51.2	52.2	144.9	248.3
Improvement	34%	37%	11%	23%

Tessel is the only program with added functionality in this release. By default, the k-mers generated by Tessel are in ‘canonical’ form. A k-mer and its reverse complement are considered equal, and only the lexicographically lower of the two is kept in the hash tables and written out at the end. Two counts are maintained for each canonical k-mer – how many times it appeared, and how many times it was found in reverse-complement form. Tessel 1.1.3 introduces two new options to control this behaviour: ‘-canonical’ and ‘-asread’. The ‘-canonical’ option is the default (and the only previous behaviour). The ‘-asread’ option gets Tessel to maintain separate entries for each k-mer ‘as read’, and not to merge the counts for a k-mer and its reverse complement. This is most useful when tiling contigs/genomes, and the default canonical tiling is most useful for sequencing reads.

The canonical binary k-mers are written to a ‘.cbt’ file (as in previous releases), and using the ‘-asread’ option causes them to be written to a ‘.abt’ file. The format of these files is the same and has not changed from previous releases. Version 0 of this file format is:

Int32 ‘k’ length used. The top 8 bits of this Int32 are reserved for use as a file format marker (currently 0).

Sets of {uint64, int32, int32} triplets, one for each k-mer.
The uint64 value is a k-mer in packed binary format.

- Two bits per base (A=00, C=01, G=10, T=11).
- K-mers are left-adjusted within the 64-bit word.
- A k-mer can be no bigger than 32 bases long.

The first int32 is the number of times the corresponding k-mer was seen.
The next int32 is the number of times it was seen in its reverse complement form.

Tessel now also supports writing k-mers and their counts out to a text file. The ‘-text textFN’ option causes Tessel to read back the binary k-mers file and write out the k-mers and their counts in text (rather than binary) format. The ‘-textFormat’ option controls the format of this text file. The ‘sum’ and ‘faSum’ output formats are compatible with the text files generated from JellyFish. The available text formats are:

pairs	k-mer (tab) count (tab) rc-count (default)
sum	k-mer (tab) summed-count
faPairs	FASTA file. Header is >count rcCount. Sequence line is k-mer.
faSum	FASTA file. Header is >summed-count. Sequence line is k-mer.

Changes from release 1.1.0

All three programs now use buffered, asynchronous code for reading FASTQ files. This reduces IO time and improves overall performance.

Blue has had minor improvements to improve accuracy and performance.

GenerateMerPairs now dynamically calculates the pair gap, rather than using 16bp for all read length. This improves accuracy for longer reads. Pairs are also generated to the end of the read rather than just sampling from the first half of the read.

Changes from Blue 1.0.1 to 1.1.0

- Improved performance and scaling.

  Performance improved by about 40% for bacterial-like data, and much more for human data. Much of the latter improvement came from better handling of extremely deep-coverage reads, and not spending inordinate amounts of time trying to correct read artefacts that were never going to be successfully corrected anyway.

- Reduced memory usage.

  Better allocation of the memory used to hold the k-mer consensus tables.

- Added –*fixed* and –*variable* options.

  By default, Blue will always maintain the length of the reads it corrects. It does this by either padding or trimming those reads whose length has changes through insertions or deletions. The –*variable* option stops this happening and reads are allowed to grow or shrink.

- Added –*paired* and –*unpaired* options.

  By default, if Blue is asked to correct a set of files it will treat them as a set and maintain the read pairing between each file in the set when writing out the corrected reads. If the *good* option is used to discard poor reads after correction, then both members of a pair will be discarded if any one of them fails the sufficient-good-k-mers test. The –*unpaired* option stops this behaviour, and the –*good* option will only discard those reads that actually fail the specified goodness test.

- Keeping good-but-unpaired reads.

  If Blue is correcting a set of ‘paired’ reads, those reads that pass the goodness test but have mates that fail the test are now written to a ‘_singles’ file. The failing reads themselves will be written to a ‘_problems’ file if the –*problems* option is set.

- Added a –help option to display a fuller version of the command-line options

Changes from Tessel 1.0.1 to 1.1.0

- Reduced memory requirements and improved performance.

  Tessel is now lock-free, allowing it to scale better on multi-processor systems. The memory allocation algorithms have also been tuned to use less memory.

Changes from GenerateMerPairs 1.0.1 to 1.1.0

- Reduced memory requirements and improved performance.

  GenerateMerPairs is now lock-free, allowing it to scale better on multi-processor systems. The memory allocation algorithms have also been tuned to use less memory.