GEne COmpression Z (GECOZ) Formats.

The GECOZ file formats is a set of file formats for efficient storing of genetic information.

It includes the Reference Genome file format (*.gcz) which contains compressed genome reference as a chain of blocks that consist of headers and nucleotide sequences.

This way it represents a compact FASTA format representation based on the Burrows–Wheeler Transformed (BWT) sequence. The BWT sequence is stored as a Huffman Shaped Wavelet Tree (HSWT) which automatically provides Huffman compression. Given that compression of genetic information doesn't improve much with higher order entropy encoding (H_0 = 1.974, H_4 = 1.910 bit/nct), further HSWT nodes compression via RLE, RRR found inefficient (Grossi, Vitter, & Xu, 2011). Using pre-calculated ranks for HSWT nodes extract and locate operations may be performed in a constant time. The implemented Succinct Suffix Array index is a variation of FM-index and is broadly used for indexing genomic information in many bioinformatics tools such as BWA (Li & Durbin, 2009) and Bowtie2 (Langmead & Salzberg, 2012).

In addition to the compressed Reference Genome ***.gcz** file, there is a corresponding Sparse Suffix Array index file (***.gcx**). The size of the index file depends of the sampling rate. The reasonable parameters used by other bioinformatics tools are 16, 32, 64 and 128 (Ferragina, González, Navarro, & Venturini, 2009).

GECOZ SEQUENCE REFERENCE FORMAT

The ***.gcz** file consists of a sequence blocks. Any block corresponds to continuous sequence[s] (i.e. chromosome[s] or contigs) and consists of a header and the data section (Table 1).

"GecozBWT"	"magic" string	8 bytes
version	1	1 byte
size	block size	8 bytes
len	the sequence string **S** length	8 bytes
headers	FASTA header1	len (string)
	terminating \0	1 byte
	FASTA header2	len (string)
	terminating \0	1 byte
	headers terminating \0	1 byte
Deflate Length Table	RFC 1951 3.2.7 encoded Huffman table	
HSWT	Huffman Shaped Wavelet Tree, a sequence of bit vectors.	

Table 1. BWT Data block format

The block may contain more than one sequence. Every FASTA header is terminated by the '**\0**' byte. The last FASTA header is terminated by the double '**\0**'.

The data block contains the Deflate Length Table (as described in RFC 1951 3.2.7) and the HSWT nodes placed one after another using pre-ordered left-to-right traversal algorithm. The Huffman Codes are also generated according RFC 1951.

For fast O(1) extract/locate operations HSWT nodes include pre-calculated ranks:

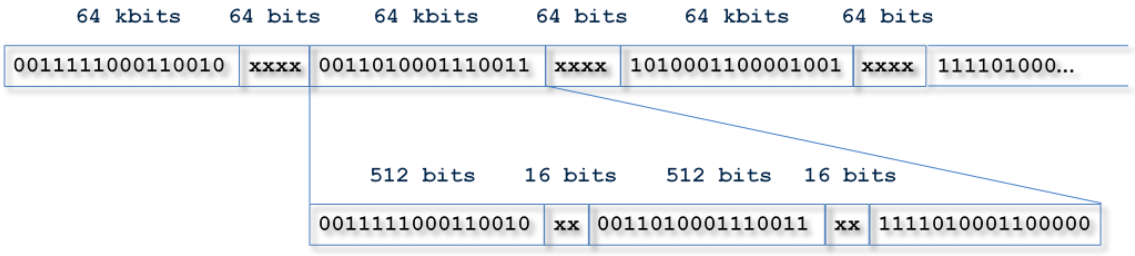

Figure 1. Ranked bit vector format

GECOZ SUCCINCT SUFFIX ARRAY INDEX FORMAT

The Succinct Suffix Array data is stored in the ***.gcx** file and consists of a ranked bit vector of stored Suffix Array indexes followed by the indexes themselves.

“GecozSSA”	“magic” string	8 bytes
version	1	1 byte
len	the index size	8 bytes
hash	the headers hash	8 bytes
SSA bit vector	bit vector that marks stored SSA indexes	
SSA indexes	wavelet table (bit vectors) that stores permuted SSA indexes	

Table 2. SSA Index block format

Bit vectors use the same format as on the Figure 1. SSA indexes are stored as a wavelet table where each row holds a bit vector of sorted index bit values (Table 3). Find and get index operations take log(n) × rank() time, where n is a number of indexes.

06 0	10 0	11 0	04 0	08 0	14 0	00 0	12 0	02 0	05 0	16 1	03 0	07 0	01 0	09 0	15 0	13 0
00110	01010	01011	00100	01000	01110	00000	01100	00010	00101	10000	00011	00111	00001	01001	01111	01101
0 06 0	0 10 1	0 11 1	0 04 0	0 08 1	0 14 1	0 00 0	0 12 1	0 02 0	0 05 0	0 03 0	0 07 0	0 01 0	0 09 1	0 15 1	0 13 1	1 16 0
0 06 1	0 04 1	0 00 0	0 02 0	0 05 1	0 03 0	0 07 1	0 01 0	1 10 0	1 11 0	1 08 0	1 14 1	1 12 1	1 09 0	1 15 1	1 13 1	0 16 0
0 00 0	0 02 1	0 03 1	0 01 0	1 06 1	1 04 0	1 05 0	1 07 1	0 10 1	0 11 1	0 08 0	0 09 0	1 14 1	1 12 0	1 15 1	1 13 0	0 16 0
0 00 0	0 01 1	1 02 0	1 03 1	0 04 0	0 05 1	1 06 0	1 07 1	0 08 0	0 09 1	1 10 0	1 11 1	0 12 0	0 13 1	1 14 0	1 15 1	0 16 0

Table 3. Suffix Array permutation.
Vectors’ bits (blue) are used for the next vector sorting (red bits).

References

Ferragina, P., González, R., Navarro, G., & Venturini, R. (2009, February). Compressed Text Indexes: From Theory to Practice. *J. Exp. Algorithmics, 13*, 12:1.12--12:1.31. doi:10.1145/1412228.1455268

Grossi, R., Vitter, J. S., & Xu, B. (2011). Wavelet Trees: From Theory to Practice. *Proceedings of the 2011 First International Conference on Data Compression, Communications and Processing* (pp. 210-221). Washington: IEEE Computer Society. doi:10.1109/CCP.2011.16

Langmead, B., & Salzberg, S. L. (2012, April). Fast gapped-read alignment with Bowtie 2. *Nat Meth, 9*, 357-359. Retrieved from http://dx.doi.org/10.1038/nmeth.1923

Li, H., & Durbin, R. (2009, July). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England), 25*(14), 1754-1760. doi:10.1093/bioinformatics/btp324