The BAllC Format Specification

Version 0.1

Wei Tian and Joseph R Ecker

BAllC stands for Binary All Cytosine format, storing DNA 5mC methylation status particularly from whole genome sequencing. It's conceived as a binary alternative to the AllC format, which is a TAB-delimited text format. BAllC not only reduces file sizes and speeds up read/write operations but also eliminates redundancies prevalent in the AllC format, especially for single cell DNA methylation data, leading to a more streamlined and efficient data representation.

Background on the AllC Format

Before diving into the details of BAllC, it's essential to understand its predecessor, the AllC format. AllC is widely used in bioinformatics for representing cytosine-specific methylation data. The AllC format is a TAB-delimited text, and the typical fields in an AllC file are:

Index	Field name	Example	Note
1	ref	chr12	
2	pos	18283342	1-based
3	strandness	+	either + or -
4	c_context	CGT	3 bases starting with “C”; can use bases defined by IUPAC (eg. H=A,T,orC, N=A,T,C, or G, etc)
5	mc	18	count of reads supporting methylation
6	cov	21	read coverage
7	methylated	1	indicator of significant methylation (1 if no test is performed)
8	(optional) num_matches	3,2,3	number of matched base calls at context nucleotides
9	(optional) num_mismatches	0,1,0	number of mismatches at context nucleotides

The BAllC Solution

One of the primary inefficiencies of AllC lies in the repeated storage of strandness and c_context information, particularly for single cell studies. For a given position in a specified genome assembly, the strandness and cytosine context remain consistent across single cells. Storing this redundant information for each cell is wasteful in terms of both storage and processing time.

BAllC addresses the redundancies of the AllC format by segregating the consistent metadata of every cytosine, like strandness and c_context, into a separate CMeta (cytosine meta) file. The fields 7-9 of the AllC format, which are rarely used in practice, are omitted from BAllC as well.

Furthermore, BAllC adopt a binary representation instead of a text-based one to reduce the file size further.

Format Structures

Field			Description	Type	Value
header					
	magic		magic string	char[6]	"BALLC\1"
	version		major version of the BAllC format	uint8_t	e.g. "0"
	version_minor		minor version of the BAllC format	uint8_t	e.g. "1"
	sc		flag of single cell BAllC format	uint8_t	"0" or "1"
	l_assembly		length of assembly text	uint32_t	
	assembly_text		user specified assembly info	char[l_assembly]	e.g. "mm10", "hg38", "donor1"
	l_text		length of header text	uint32_t	
	header_text		note for this ballc file	char[l_text]	
	n_refs		number of references (chromsomes)	uint16_t	
	refs		list of refs (n=n_refs)		
		ref			
		l_name	length of reference name	uint32_t	
		ref_name	reference name	char[l_name]	e.g. "chr1"
		ref_len	length of reference	uint32_t	e.g. "248956422"
mc_records			list of mc_records		
	mc_record				
	pos		position of cytosine	uint32_t	
	ref_id		reference id, 0 ≤r ef_id < n_refs	uint16_t	
	mc		number of counts supporting methylation call	uint16_t when sc=1; uint8_t when sc=0	
	cov		number of counts covering the position	uint16_t when sc=1; uint8_t when sc=0	

The BAllC file starts with a header that encapsulates essential metadata:

1. Magic String (magic): A signature for the BAllC format. It is always set to "BALLC\1".
2. Version and Version Minor: These fields denote the major and minor versions of the BAllC format, respectively. For instance, "0" and "1" represent version 0.1.
3. Single Cell Flag (sc): A flag that indicates if the BAllC format is for single-cell data. It can be "0" (not single-cell) or "1" (single-cell).
4. Assembly Text: This field provides user-specified assembly information, such as "mm10", "hg38", or "donor1". The length of the assembly text is determined by the l_assembly field.
5. Header Text: This is a note for the BAllC file. The length of this text is determined by the l_text field.
6. Number of References (n_refs): This field denotes the number of references (chromosomes/contigs) in the genome assembly.
7. References (refs): This is a list containing details of each reference. Each reference has a name (like "chr1"), length of which determined by “l_name” and a length of the reference (for example, "248956422" for “chr1” of human).

Following the header, the file contains data blocks that encapsulate the actual methylation data:

1. Position (pos): This indicates the position of the cytosine in the reference.
2. Reference ID (ref_id): This ID links the data block to one of the references specified in the header.
3. Methylation Count (mc): Represents the number of counts supporting the methylation call. Its data type varies based on the sc flag.
4. Coverage (cov): Denotes the number of counts covering the position. Like mc, its data type depends on the sc flag.

CMeta file

The CMeta file serves as a companion to the BAllC format, efficiently storing the consistent metadata of each cytosine. It is essentially a TAB-delimited text file that includes following fields:

Index	Field name	Example	Note
1	ref	chr12	
2	pos	18283342	1-based
3	strandness	+	either + or -
4	c_context	CGT	3 bases starting with "C"; can use bases defined by IUPAC (eg. H=A,T,orC, N=A,T,C, or G, etc)

By segregating this information, which remains static across single cells for a given genomic position, into the CMeta file, BAllC effectively minimizes redundancy and optimizes storage. When using BAllC in tandem with CMeta, researchers can achieve the same comprehensive view of methylation data yet without the inefficiencies associated with the AllC format.

Working with BAllC

TODO…