The Portable Cray Bioinformatics Library

James Long[1]

Keywords: benchmarking, bioinformatics, compression, endian, library

1 Introduction.

The original Cray Bioinformatics Library (CBL) is a low level set of library routines using proprietary Cray hardware to implement some common nucleotide/protein sequence manipulations typical in a bioinformatics context. Written in Fortran and Cray assembly language (most are callable from C), the original CBL was coded and optimized on a Cray SV1 vector machine. Cray also has a port for their new X1™.

The Portable CBL is the open source version [1] written in C that implements the computational primitives in a generic fashion with little regard to specific hardware. The CBL routines facilitate performance by operating on compressed data whenever possible. In the case of nucleotide data, for example, it is sufficient to represent each of the four nucleotides with only two bits, and thus a 64-bit word can contain a sequence of 32 nucleotides instead of the normal 8. The CBL search routine compares whole words of a compressed query against a compressed database, realizing a significant performance increase. In addition to 2-bit compression, CBL supports 4 bit and 5 bit levels for larger alphabets. The CBL will continue to grow as additional biological computational primitives are identified and implemented [2].

2 Version 1.0 Routines.

cb_amino_translate_ascii - translate nucleotides to amino acids
cb_compress - compresses nucleotide or amino acid ASCII data
cb_copy_bits - copy contiguous sequence of memory bits
cb_countn_ascii - counts A, C, T, G, and N characters in a string
cb_fasta_convert - restructure the memory image of a FASTA file
cb_free - frees memory allocated with cb_malloc in Cray version
- simply calls free() in portable version
cb_irand - generates an array of random bits
cb_malloc - allocate block aligned memory region in Cray version
- simply calls malloc() in portable version
cb_read_fasta - loads data from a FASTA file into memory arrays
cb_repeatn - find short tandem repeats in a nucleotide string
cb_revcompl - reverse complements compressed nucleotide data
cb_searchn - gap-free nucleotide search allowing mismatches
cb_uncompress - uncompress nucleotide or amino acid data to ASCII
cb_version - returns the version number of libcbl

[1] Arctic Region Supercomputing Center, PO 756020, Fairbanks, AK 99775-6020
E-mail: `jlong@arsc.edu`

3 Performance.

A benchmark option in v1.0 exercises seven of the routines. Platforms used:

800 MHz Cray X1, running in both MSP and SSP mode
1.3 GHz Intel Itanium 2, 1.5 MB L3, intel 7.1 (icc) and gcc 2.96 compilers
1.4 GHz AMD Athalon MP 1600+, 256 KB cache, intel 7.1 and gcc 3.2.2 compilers
1.7 GHz IBM P4, 32 MB shared L3, 64-bit mode
2.8 GHz Intel Xeon, 512 KB cache, intel 8.0 and gcc 3.2.2

	Cray CBL		Portable CBL								
CBL	**800 MHz**		**800 MHz**		**1.3 GHz**		**1.4 GHz**		**1.7 GHz**	**2.8 GHz**	
Function	**X1**		**X1**		**Itanium2**		**AMD**		**IBM**	**Xeon**	
	MSP	**SSP**	**MSP**	**SSP**	**icc**	**gcc**	**icc**	**gcc**	**P4**	**icc**	**gcc**
cb_amino_tran	8	27	90	156	31	46	63	87	36	25	64
cb_compess/un	5	10	44	56	24	59	64	70	38	31	39
cb_copy_bits	3	4	1	1	8	27	45	45	10	19	18
cb_count_ascii	4	15	5	23	16	44	59	62	23	23	27
cb_repeatn	45	55	122	142	42	55	48	49	26	25	27
cb_revcompl	3	12	19	33	20	80	148	138	35	53	63
cb_searchn	23	85	36	65	92	94	129	167	40	78	127

Table 1: Benchmark times in seconds.

4 Roadmap.

The Portable CBL will follow the roadmap for Cray's implementation (now at 2.0). Developers interested in contributing to the roadmap should consult the author.

Coming in version 1.1:
cb_swa_fw - compute Smith-Waterman cell scores with ASCII input

Coming in version 1.2
cb_isort & cb_isort1 - unsigned integer radix sort with and w/o index array
cb_cghistn - histograms of cg density in a string
cb_swn_fw & cb_swn4_fw- same as cb_swa_fw, except with 2- or 4-bit nucleotide input
cb_nmer - creates up to 64-bit-length short sequences from each starting point in the input string.

Coming in version 2.0
cb_sort – multi-pass sort routine for compressed data

References

[1] http://cbl.sourceforge.net
[2] Long, J. 2003. The Portable Cray Bioinformatics Library. *Proceedings of the 45th Cray User Group Conference,* http://www.arsc.edu/support/technical/html/200305.OpenCBL/jlong_cbl.htm