A Parallel Python Implementation of BLAST+ (PPIB) for Characterization of Complex Microbial Consortia

Amina Jackson
Naval Research Laboratory
4555 Overlook Avenue – SW
Washington, DC 20375
amina.jackson@nrl.navy.mil

W. Connor Horne
Naval Research Laboratory
4555 Overlook Avenue – SW
Washington, DC 20375
william.horne@nrl.navy.mil

Daniel Beall
Naval Research Laboratory
4555 Overlook Avenue – SW
Washington, DC 20375
daniel.beall@nrl.navy.mil

Kenneth Jiang
Naval Research Laboratory
4555 Overlook Avenue – SW
Washington, DC 20375
kenneth.jiang@nrl.navy.mil

W. Judson Hervey, IV*
Naval Research Laboratory
4555 Overlook Avenue – SW
Washington, DC 20375
judson.hervey@nrl.navy.mil

ABSTRACT

Basic Local Alignment Search Tool (BLAST) is an indispensable application among "-omics" sciences for putative functional inference of biomolecules. Here, we deploy BLAST on an HPC system for large-scale environmental microbiomes functional inference: microbial fuel cell Biocathodes and ship hull biofilms.

KEYWORDS

Bioinformatics, Computational Biology, Data-Intensive Parallel Algorithms, Broadly-Applicable Performance Optimization Techniques, Parallel Application Frameworks.

ACM Reference format:

A. Jackson, W. C. Horne, D. Beall, K. Jiang, W. J. Hervey, IV. 2017. A Parallel Python Implementation of BLAST+ (PPIB) for Characterization of Complex Microbial Consortia. In *Proceedings of ACM Supercomputing 2017 conference, Denver, Colorado USA, November 2017 (SC17)*. https://doi.org/0000001.0000001

1 INTRODUCTION

Rapid technological advancements among analytical biomolecular sequencing platforms have made "-omics" sciences (genomics, transcriptomics, and proteomics) data-driven endeavors. Since the completion of the human genome, data explosion has led to exponential growth in data volume, which has made high performance computing (HPC) appealing in "-omics". However, application deployment on new and emerging HPC architectures pose significant challenges. BLAST [1], essential "-omics" data analysis tool compares unknown biomolecular sequences to large-scale reference sequences of known origin to infer putative function(s). Although BLAST has had disparate modified implementations to improve performance over increasing data volume, these improvements [3-4] have neither been consistently maintained for deployment on emerging architectures nor applied to complex, "real-world" microbiomes. Additionally, implementations [3-4] often require substantial modifications to efficiently use HPC resources. Here, we compare BLAST+ performance of 2 configurations: BLAST+ and Parallel Python Implementation of BLAST+ (PPIB). Application code executed on HPC system, with 4 disparate inputs: subset 52 long sequences, subset 200 short sequences, Biocathode-MCL [5-6], and ship hull biofilms (SHB) [7-8]. NCBI non-redundant (NR) protein sequences served as the reference.

Although advanced technologies have increased computing power "-omics" is utilizing to minimize its growing big data problems, putative function assignments among complex microbiomes remains a challenge to address. In our study, BLAST+ could not scale based on computing power and did not complete functional assignments of our microbiomes [5-8] within resource limits while PPIB scaled by 135% and completed assignments with 164% completion and walltime advantage over BLAST+.

2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

2.1 HPC Hardware Configuration

The HPC system used, Thunder [9], is an SGI Ice X with Intel Xeon Intel E5-2699v3 standard compute nodes (36 cores/node).

2.2 Software Configuration

2.2.1 BLAST+ and NR reference sequences. Source code and NR reference sequences (96,287,830 protein entries) were downloaded from NCBI [10] and compiled as previously described [11].

2.2.2 Parallelization and Job Submission. mpi4py [12] installed with Anaconda2 [13] was used for PPIB's MPI implementation. Portable Batch System (PBS) submitted jobs at 336 hours extended to ensure completion for benchmarking purposes.

2.3 PPIB Implementation

2.3.1 Python Message Passing Interface (MPI). MPI implementation use collective methods Broadcast, Scatter, and Gather [12] to create subdirectories, break input file into sub-files distributed amongst cores, and assemble output into a single file.
2.3.2 Python Multiprocessing. Pool creates multiple query processes and instances searched simultaneously. Therefore, a job assigned 36 cores can have 36*100=3600 queries processing simultaneously.

2.4 Data Processed

2.4.1 Test Input. As sequence length influences processing time, tests on "long" 52 entries and "short" 200 entries aimed at determining job completion within walltime limits.
2.4.2 Biocathode-MCL. Microbial fuel cell biocathode system [5-6] contained 79,765 entries, determined from *de novo* metagenome sequencings where 2,807 entries were determined "long" (eg. >667 residues).
2.4.3 SHB. Microbiome previously characterized in-house contained 243,146 entries [7-8], 305 entries determined "long".

3 RESULTS AND DISCUSSION

3.1 Short and Long Test Data

BLAST results with 2 configurations on subset files, single compute node in Fig 1 shows 52 input-file at walltime 170 minutes BLAST+ and 33 minutes PPIB, took longer than 200 input-file. PPIB had 135% advantage at 10 minutes over BLAST+'s 159 minutes on 200 input-file.

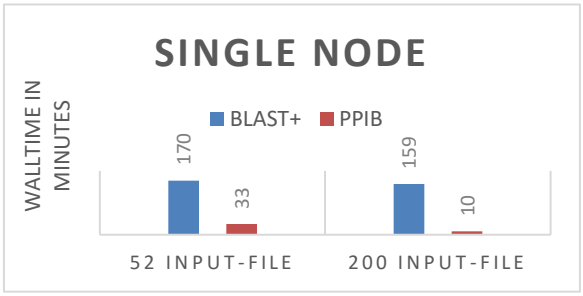

Figure 1: Effect of Protein Sequence Length on Walltime

Fig 2. shows impact from increasing computing power by increasing cores used to identify job completion solution within resource limits.

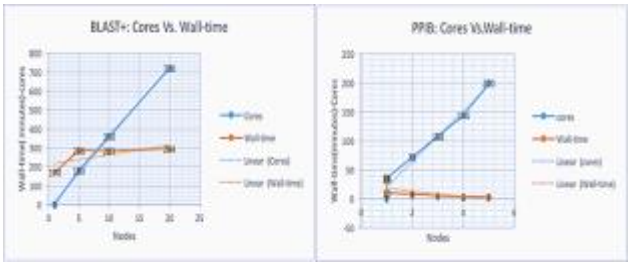

Figure 2: Effect of Resource Allocation on Walltime.

In BLAST+ serial-like behavior, increases in cores negatively impacts walltime. In PPIB parallelism, increases in cores decreases walltime. BLAST+'s single node walltime 170 minutes was lower than 5 nodes' 286 minutes. In comparison, PPIB single core walltime was 33 minutes, 36 cores 10 minutes and 200 cores lessened to 3 minutes.

3.2 Complex Microbiome Sample Input

3.2.1 Biocathode-MCL. With 79,765 predicted metaproteome entries [5-6], BLAST+ completed 14.4% in 336 hours. However, PPIB on single node completed the entire job in 206 hours. Increasing resources allocated to PPIB to 3 nodes decreased walltime to 71 hours (shown table 1 below).
3.2.2 SHB Microbiome. Of 243,146 predicted metaproteome entries [7-8], on a single node, BLAST+ completed 27%. However, PPIB processed 81% SHB entries over the same walltime and completed all entries in 160 hours when assigned 3 nodes.

3.3 Comparisons of BLAST+ with PPIB

By scattering tasks with MPI and creating parallel query processes on cores with multiprocessing, PPIB solves the BLAST+ serial-like behavior and lowers walltime cost. Table 1 below demostrates this PPIB advantage over BLAST+.

Table 1: Walltime Advantage of PPIB over BLAST+

	Data Type	*Total of queries*	*Wall-time*	*Completion %*	*Wall-time Cost advantage %*
BLAST+	Test subset A	52	170	100%	
PPIB	Test subset A	52	33	100%	135%
BLAST+	Test subset B	200	159	100%	
PPIB	Test subset B	200	10	100%	176%
BLAST+	Biocathode-MCL	79765	336	14%	
PPIB	Biocathode-MCL	79765	71	100%	130(+86) 216%
BLAST+	SHB	243146	336	27%	
PPIB	**SHB**	243146	160	100%	70(+73) 143%

4 CONCLUSIONS

We have demonstrated efficiency of a Parallel Python Implementation of BLAST+ (PPIB) which leverages HPC resources for putative functional assignments among complex microbiomes of up to nearly 250,000 proteins.

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research via NRL core funds. This work utilized resources allocated by the DoD's HPCMP. The opinions and assertions contained herein are those of the authors and are not to be construed as those of the U.S. Navy, military service at large or U.S. Government.

REFERENCES

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 3 (Oct. 1990), 403–410. DOI: http://doi.org/10.1016/S0022-2836(05)80360-2

[2] C. Camacho, G. Coulouris, V. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer and T. L. Madden. 2009. BLAST+: architecture and applications. *BMC Bioinformatics*. 10, 421 (Dec. 2009). 1471-2105. DOI: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-421

[3] A. E. Darling, L. Carey, and W. Feng. 2003. The Design, Implementation, and Evaluation of mpiBLAST. ClusterWorld Conference & Expo and the 4th International Conference on Linux Clusters: The HPC Revolution. LA-UR 03-2862 URL: http://pages.cs.wisc.edu/~darling/mpiblast-cwce2003.pdf

[4] S. E. Sawyer, B. Rekepalli, M. D. Horton, R. G. Brook. 2015. HPC-BLAST: Distributed BLAST for Xeon Phi Clusters. *ACM* 978, 1-4503-3853. (Sept. 2015), DOI: http://dl.acm.org/citation.cfm?doid=2808719.2811435

[5] Z. Wang Dagmar H. Leary, A. P. Malanoski, R. W. Li, W. J. Hervey IV, B. J. Eddie, G. S. Tender , S. G. Yanosky, G. J. Vora,a L. M. Tender, B. Lin and S. M. Strycharz-Glavena. 2015. A Previously Uncharacterized, Nonphotosynthetic Member of the *Chromatiaceae* Is the Primary CO2-Fixing Constituent in a Self- Regenerating Biocathode. *J. Applied and Environmental Microbiology*. 81, 2 (Jan 2015), 699-712. DOI: 10.1128/AEM.02947-14

[6] D. H. Leary, W. J. Hervey, A. P. Malanoski, Z. Wang, B. J. Eddie, G. S. Tender, G. J. Vora, L. M. Tender, B. Lin and S. M. Strycharz-Glaven. 2015. Metaproteomic evidence of changes in protein expression following a change in electrode potential in a robust biocathode microbiome. *PROTEOMICS*. 15, 20 (Oct. 2015), 3486-3496 DOI: 10.1002/pmic.201400585

[7] D. H. Leary, W. J. Hervey, IV, R. W. Li, J. R. Deschamps, A. W. Kusterbeck, and G. J. Vora. 2012. Method Development for Metaproteomic Analyses of Marine Biofilms. *Analytical Chemistry*. 84, 9 (Apr. 2012), 4006-4013 DOI: http://pubs.acs.org/doi/abs/10.1021/ac203315n

[8] D. H. Leary, R. W. Li, L. J. Hamdan, W. J. Hervey, IV, N. Lebedev, Z. Wang, J. R. Deschamps, A. W. Kusterbeck, G. J. Vora. 2014. *Biofouling*. 30, 10 (Nov. 2014), 1211-1223. DOI: http://dx.doi.org/10.1080/08927014.2014.977267

[9] AFRL The Air Force Research Laboratory. 2017. High Performance Computing System. Thunder URL: https://www.afrl.hpc.mil/hardware/index.html. (ACCESSED 17 JUL 2016)

[10] National Center for Biotechnology Information and U.S. National Library of Medicine. 2016. blast+ version 2.2.31 (Sept. 2015 Release). URL: ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/ (ACCESSED 03 SEPT 2016).

[11] G. Albert. BLAST for the Intel® Xeon Phi™ Coprocessor. 2015. Intel Software Developer Zone. (Jan. 2015). URL: https://software.intel.com/en-us/articles/blast-for-the-intel-xeon-phi-coprocessor (ACCESSED 03 SEPT 2016)

[12] L. Dalcin. 2015. MPI for Python (Release 2.0.0). CIMEC URL: http://pythonhosted.org/mpi4py/ (ACCESSED 15 APR 2017)

[13] Continuum Analytics. Anaconda 2017. ANACONDA DISTRIBUTION (Version 4.2.0). URL: https://www.continuum.io/downloads (ACCESSED 10 Mar 2017)