

The large-scale blast score ratio (LS-BSR) pipeline: a method to rapidly compare genetic content between bacterial genomes

Jason W. Sahl^{1,2,4}, J. Gregory Caporaso², David A. Rasko³ and Paul Keim^{1,2,4}

¹ Division of Pathogen Genomics, Translational Genomics Research Institute, Flagstaff, AZ, USA

² Department of Biological Sciences, Northern Arizona University, Flagstaff, AZ, USA

³ Department of Microbiology and Immunology, Institute for Genome Sciences, University of Maryland School of Medicine, Baltimore, MD, USA

⁴ Center for Microbial Genetics and Genomics, Northern Arizona University, Flagstaff, AZ, USA

ABSTRACT

Background. As whole genome sequence data from bacterial isolates becomes cheaper to generate, computational methods are needed to correlate sequence data with biological observations. Here we present the large-scale BLAST score ratio (LS-BSR) pipeline, which rapidly compares the genetic content of hundreds to thousands of bacterial genomes, and returns a matrix that describes the relatedness of all coding sequences (CDSs) in all genomes surveyed. This matrix can be easily parsed in order to identify genetic relationships between bacterial genomes. Although pipelines have been published that group peptides by sequence similarity, no other software performs the rapid, large-scale, full-genome comparative analyses carried out by LS-BSR.

Results. To demonstrate the utility of the method, the LS-BSR pipeline was tested on 96 *Escherichia coli* and *Shigella* genomes; the pipeline ran in 163 min using 16 processors, which is a greater than 7-fold speedup compared to using a single processor. The BSR values for each CDS, which indicate a relative level of relatedness, were then mapped to each genome on an independent core genome single nucleotide polymorphism (SNP) based phylogeny. Comparisons were then used to identify clade specific CDS markers and validate the LS-BSR pipeline based on molecular markers that delineate between classical *E. coli* pathogenic variant (pathovar) designations. Scalability tests demonstrated that the LS-BSR pipeline can process 1,000 *E. coli* genomes in 27–57 h, depending upon the alignment method, using 16 processors.

Conclusions. LS-BSR is an open-source, parallel implementation of the BSR algorithm, enabling rapid comparison of the genetic content of large numbers of genomes. The results of the pipeline can be used to identify specific markers between user-defined phylogenetic groups, and to identify the loss and/or acquisition of genetic information between bacterial isolates. Taxa-specific genetic markers can then be translated into clinical diagnostics, or can be used to identify broadly conserved putative therapeutic candidates.

Submitted 11 February 2014

Accepted 14 March 2014

Published 1 April 2014

Corresponding author

Jason W. Sahl, jsahl@tgen.org

Academic editor

Jiayan Wu

Additional Information and
Declarations can be found on
page 10

DOI 10.7717/peerj.332

© Copyright
2014 Sahl et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Bioinformatics, Computational Biology, Genomics, Microbiology
Keywords Genomics, Bioinformatics, Microbiology, Pathogens, Comparative genomics

INTRODUCTION

Whole genome sequence (WGS) data has changed our view of bacterial relatedness and evolution. Computational analyses available for WGS data include, but are not limited to, single nucleotide polymorphism (SNP) discovery (*DePristo et al., 2011*), core genome phylogenetics (*Sahl et al., 2011*), and gene based comparative methods (*Hazen et al., 2013*; *Sahl et al., 2013*). In 2005, a BLAST score ratio (BSR) method was introduced in order to compare peptide identity from a limited number of bacterial genomes (*Rasko, Myers & Ravel, 2005*). However, the “all vs. all” implementation of this method scales poorly with a larger number of sequenced genomes.

Here we present the Large Scale BSR method (LS-BSR) that can rapidly compare gene content of a large number of bacterial genomes. Comparable methods have been published in order to group genes into gene families, including OrthoMCL (*Li, Stoeckert & Roos, 2003*), TribeMCL (*Enright, Van Dongen & Ouzounis, 2002*), and GETHOGs (*Altenhoff et al., 2013*). Although grouping peptides into gene families is not the primary focus of LS-BSR, the output can be parsed to identify the pan-genome (*Tettelin et al., 2008*) structure of a species; scripts are included with LS-BSR that classify coding sequences (CDSs) into pan-genome categories based on user-defined identity thresholds.

Pipelines have also been established to perform comprehensive pan-genome analyses, including the pan-genome analysis pipeline (PGAP) (*Zhao et al., 2012*), which requires specific gene annotation from GenBank and complicates the analysis of large numbers of novel genomes. PGAP also doesn't allow for the screen of specific genes of interest against query genomes in order to identify patterns of distribution. GET_HOMOLOGUES (*Contreras-Moreira & Vinuesa, 2013*) is a recently published tool that can be used for pan-genome analyses, including the generation of dendrograms based on the presence/absence of homologous genes; by only using presence/absence based on gene homology, more distantly related gene relatedness cannot be fully investigated. The integrated toolkit for the exploration of microbial pan-genomes (ITEP) toolkit (*Benedict et al., 2014*) was recently published and performs similar functions to LS-BSR, including the identification of gene gain/loss at nodes of a phylogeny. ITEP relies on multiple dependencies and workflows, which are available as a pre-packaged virtual machine. The authors of ITEP report that an analysis of 200 diverse genomes would take ~6 days on a server with 12 processors and scales quadratically with additional genomes.

MATERIALS AND METHODS

The LS-BSR method can either use a defined set of genes, or can use Prodigal (*Hyatt et al., 2010*) to predict CDSs from a set of query genomes. When using Prodigal, all CDSs are concatenated and then de-replicated using USEARCH (*Edgar, 2010*) at a pairwise identity of 0.9 (identity threshold can be modified by the user). Each unique CDS is then translated with BioPython (www.biopython.org) and aligned against its nucleotide

sequence with TBLASTN (Altschul *et al.*, 1997) to calculate the reference bit score; if BLASTN or BLAT (Kent, 2002) is invoked, the nucleotide sequences are aligned. Each query sequence is then aligned against each genome with BLAT, BLASTN, or TBLASTN and the query bit score is tabulated. The BSR value is calculated by dividing the query bit score by the reference bit score, resulting in a BSR value between 0.0 and 1.0 (values slightly higher than 1.0 have been observed due to variable bit score values obtained by TBLASTN). The results of the LS-BSR pipeline include a matrix that contains each unique CDS name and the BSR value in each genome surveyed. CDSs that have more than one significant BSR value in at least one genome are also identified in the output. A separate file is generated for CDSs where one duplicate is significantly different than the other in at least one genome; these regions could represent paralogs and may require further detailed investigation. Once the LS-BSR matrix is generated, the results can easily be visualized as a heatmap or cluster with the Multiple Experiment Viewer (MeV) (Saeed *et al.*, 2006) or R (R Core Team, 2013); the heatmap represents a visual depiction of the relatedness of all peptides in the pan-genome across all genomes. The Interactive Tree Of Life project (Bork *et al.*, 2008) can also be used to generate heatmaps from LS-BSR output and correlate heatmap data with a provided phylogeny. A script is included with LS-BSR (compare_BSR.py) to rapidly compare CDSs between user-defined sub-groups, using a range of BSR thresholds set for CDS presence/absence. Annotation of identified CDSs can then be applied using tools including RAST (Aziz *et al.*, 2008) and prokka (<http://www.vicbioinformatics.com/software/prokka.shtml>). LS-BSR source code, unit tests, and test data can be freely obtained at <https://github.com/jasonsahl/LS-BSR> under a GNU GPL v3 license.

RESULTS AND DISCUSSION

LS-BSR algorithm speed and scalability

To determine the scalability of the LS-BSR method, 1,000 *Escherichia coli* and *Shigella* genomes were downloaded from Genbank (Benson *et al.*, 2012); *E. coli* was used as a test case due to the large number of genomes deposited in Genbank. Genomes were sub-sampled at different depths (100 through 1000, sampling every 100) with a python script (<https://gist.github.com/jasonsahl/115d22bfa35ac932d452>) and processed with LS-BSR using 16 processors. A plot of wall time and the number of genomes processed demonstrates the scalability of the method (Fig. 1A) using three different alignment methods. To demonstrate the parallel nature of the algorithm, 100 *E. coli* genomes were processed with different numbers of processors. The results demonstrate decreased runtime of LS-BSR with an increase in the number of processors used (Fig. 1B).

Improvements on a previous BSR implementation

The LS-BSR method is an improvement on a previous BSR implementation (<http://bsr.igs.umaryland.edu/>) in terms of speed and ease of use. The former BSR algorithm (Rasko, Myers & Ravel, 2005) requires peptide sequences and genomic coordinates of CDSs to run. LS-BSR only requires genome assemblies in FASTA format, which is the standard

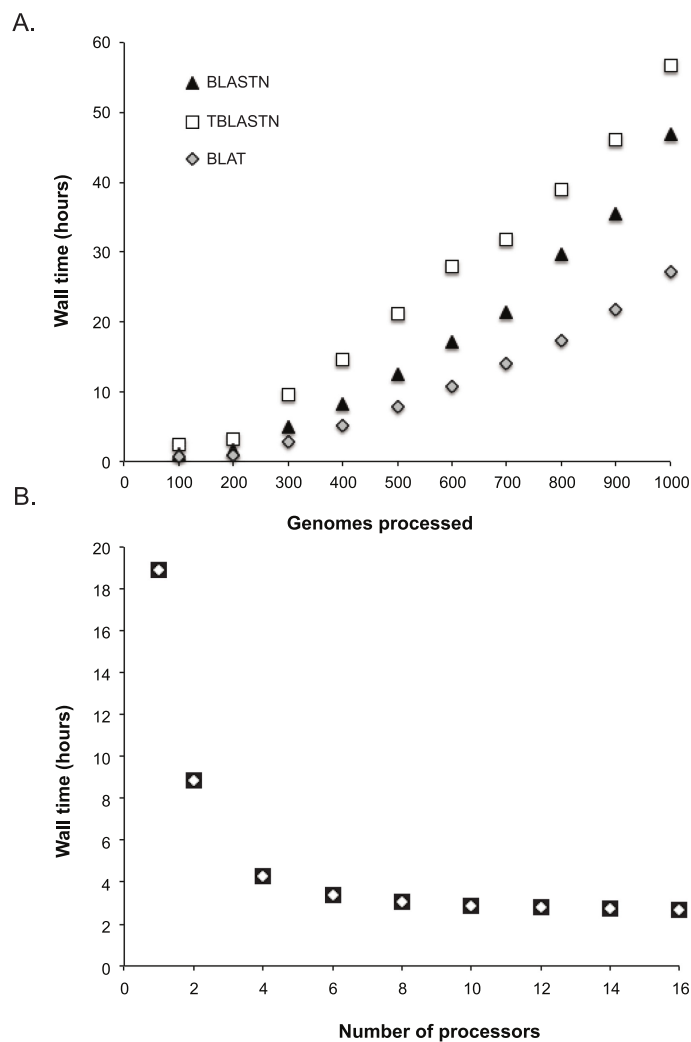


Figure 1 Time performance of the LS-BSR pipeline. (A) 1000 *Escherichia coli* and *Shigella* genomes were randomly sub-sampled and analyzed using default LS-BSR parameters and 16 processors. Wall time was plotted against the number of genomes analyzed. The results demonstrate that the LS-BSR pipeline scales well with increasing numbers of genomes. (B) The same set of 100 *E. coli* genomes was processed with different numbers of processors and the wall time was plotted. The results demonstrate that using additional processors decreases the overall run time of LS-BSR.

output of most genome assemblers. To test the speed differences between methods, 10 *E. coli* genomes (Table S1) were processed with both methods. Using the same number of processors ($n = 2$) on the same server, the original BSR method took ~ 14 h (wall time) to complete, while the LS-BSR method, using TBLASTN, took ~ 25 min to complete (wall time). Because the original BSR method is an “all vs. all” comparison and the LS-BSR method is a “one vs. all” comparison, this difference is expected to be more pronounced as the number of genomes analyzed increases.

Test case: analysis of 96 *E. coli* and *Shigella* genomes

To demonstrate the utility of the LS-BSR pipeline, a set of 96 *E. coli* and *Shigella* genomes were processed (Table S1); these genomes are in various stages of assembly completeness and have been generated with various sequencing technologies from Sanger to Illumina. The BSR matrix was generated with TBLASTN in 2 h 34 min from a set of ~20,000 unique CDSs using 16 processors. In addition to the LS-BSR analysis, a core genome single nucleotide polymorphism (SNP) phylogeny was inferred on 96 genomes using methods published previously (Sahl et al., 2011); the SNP phylogeny with labels is shown in Fig. S1. Briefly, all genomes were aligned with Mugsy (Anghiuoli & Salzberg, 2011) and the core genome was extracted from the whole genome alignment; the alignment file was then converted into a multiple sequence alignment in FASTA format. Gaps in the alignment were removed with Mothur (Schloss et al., 2009) and a phylogeny was inferred on the reduced alignment with FastTree2 (Price, Dehal & Arkin, 2010).

The compare_BSR.py script included with LS-BSR was used to identify CDS markers that are unique to specific phylogenetic clades (Fig. 2). Identified CDSs had a BSR value ≥ 0.8 in targeted genomes and a BSR value < 0.4 in non-targeted genomes; the gene annotation of all marker CDSs is detailed in Table S2. The conservation and distribution of all clade-specific markers was visualized by correlating the phylogeny with a heatmap of BSR values (Fig. 2). This presentation provides an easy way for the user to highlight features conserved in one or more phylogenomic clades.

E. coli and *Shigella* pathogenic variants (pathovars) are delineated by the presence of genetic markers primarily present on mobile genetic elements (Rasko et al., 2008). The conservation of these markers was used as a validation of the LS-BSR method. A representative sequence from each pathovar-specific marker (Table S2) was screened against the 96-genome test set and the BSR values (Table S3) were visualized as a heatmap (Fig. 2). The BSR matrix demonstrates that pathovar-specific genes were accurately identified in each targeted genome (Table S3, Fig. 2). For example, the *ipaH3* marker was positively identified in all *Shigella* genomes and the Shiga toxin gene (*stx2a*) was conserved in the clade including O157:H7 *E. coli* (Fig. 2). A sub-set of these 96 *E. coli* genomes is included with LS-BSR as test data to characterize the conservation and distribution of pathovar specific genes.

Finally, the BSR values were used to cluster all 96 genomes with an average linkage algorithm implemented in MeV and the structure of the resulting dendrogram was compared to the core SNP phylogeny. The BSR based clustering method incorporates both the core and accessory genome, while the SNP phylogeny relies on core genomic regions alone. A comparison of the tree structures demonstrates that while *Shigella* genomes share a diverse evolutionary history (Fig. 3A), they all cluster together based on gene presence and conservation (Fig. 3B). This result was also observed using a k-mer frequency method (Sims & Kim, 2011), which uses all possible k-mer values to infer a phylogeny and validates the findings of the LS-BSR pipeline. The dendrogram also differed from the core SNP phylogeny in other genomes, which could represent either assembly problems, or more likely the acquisition of accessory genomic regions that are not a product of direct descent.

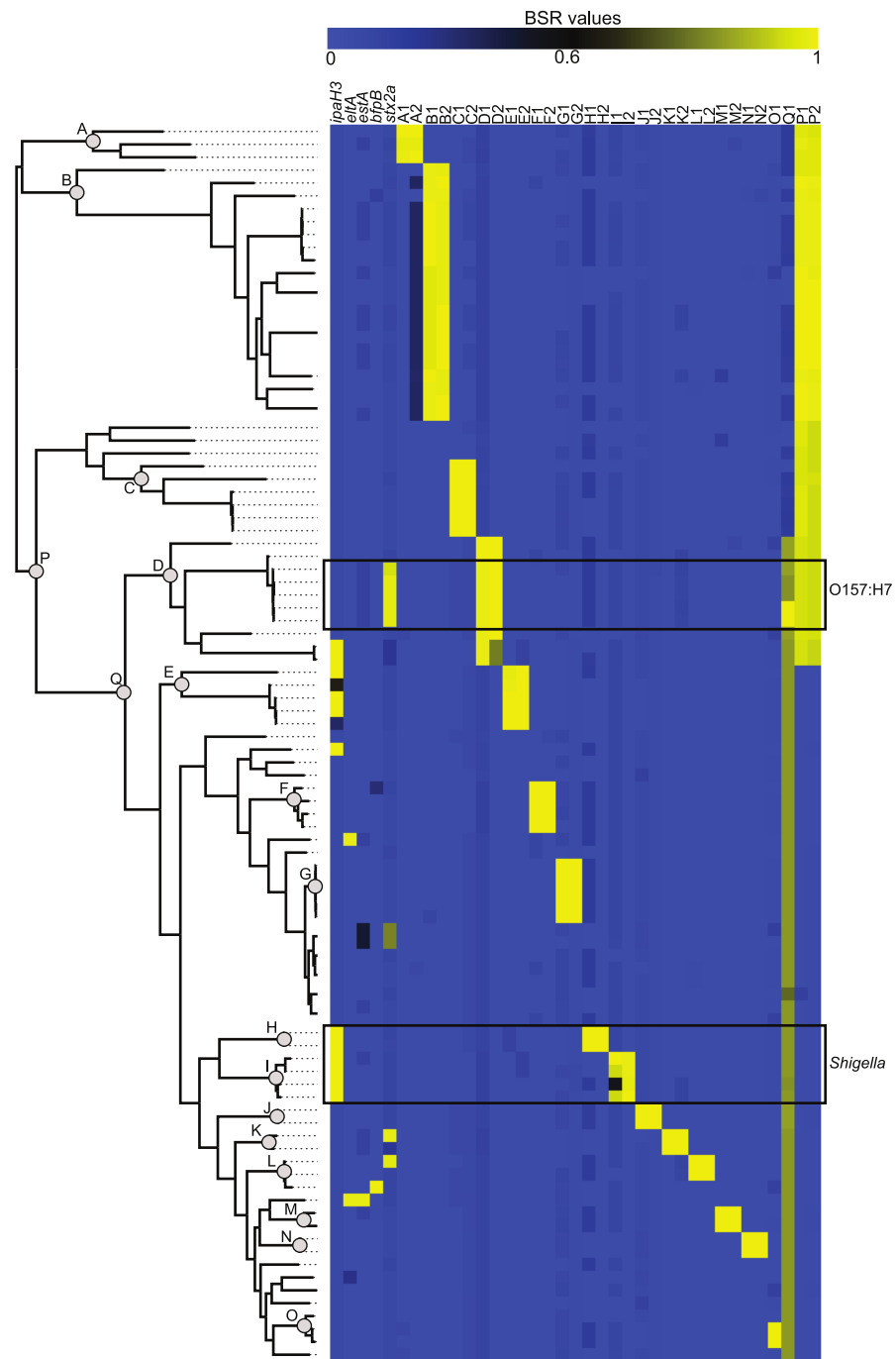


Figure 2 The distribution of virulence factors and phylogenomic markers associated with a core single nucleotide polymorphism (SNP) phylogeny. The core SNP phylogeny was inferred from a whole genome alignment produced by Mugsy (Angiuoli & Salzberg, 2011). Known virulence genes (Table S2) were screened against 96 *Escherichia coli* and *Shigella* genomes using BLASTN within LS-BSR. Clade specific markers were identified at defined nodes in the phylogeny (A through Q). Gene annotations for these markers are detailed in Table S2.

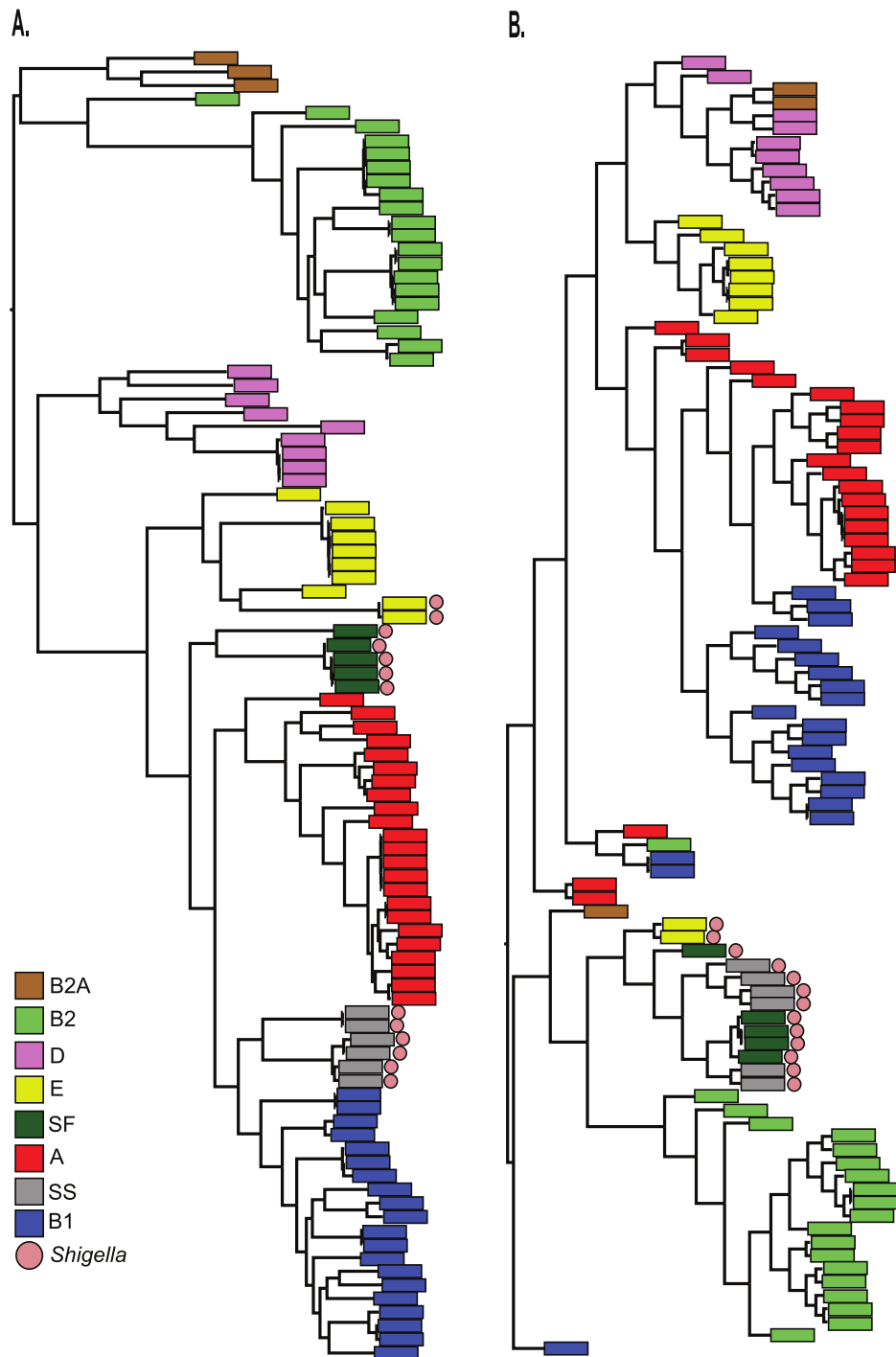


Figure 3 Comparison of LS-BSR cluster with core genome SNP phylogeny. A comparison of 96 *Escherichia coli*/*Shigella* genomes between (A) a core single nucleotide polymorphism (SNP) phylogeny or (B) a cluster generated with the Multiple Experiment Viewer (Saeed et al., 2006) from BLAST Score Ratio (BSR) values that include the entire pan-genome. Colors applied to each classical *E. coli* phylogroup were applied to the SNP phylogeny and transferred to the BSR cladogram. *Shigella* genomes are marked with a red circle.

Table 1 Comparison of four pan-genome methods on a test set of 11 *Streptococcus pyogenes* genomes.

	LS-BSR	GET_HOMOLOGUES	PGAP	ITEP
Clusters orthologs?	Yes	Yes	Yes	Yes
Open source?	Yes	Yes	Yes	Yes
Pan-genome calculation?	Yes	Yes	Yes	Yes
Lineage specific gene identification?	Yes	Yes	Yes	Yes
Functional annotation?	No	Yes	Yes	Yes
Analyzes user-defined genes?	Yes	No	No	Yes
Input files	“.fasta”	GenBank or “.faa”	“.faa”, “.fna”, “.ppt”	GenBank
Supported platforms	linux, OSX	linux/OSX	linux	linux/OSX
Core genome size	1318, 1350, 1426 ^a	1232, 1234 ^b	1332, 1366 ^{c,d}	1342
Time (2 cores), only runtime	5 m 59 s, 1 m 53 s, 1 m 17 s ^a	25 m 14 s	29 m 59 s, 199 m 58 s ^{c,d}	24 m 22 s

Notes.^a TBLASTN, BLASTN, BLAT.^b COG, MCL.^c MP, GF.^d Taken from publication.

The functionality of LS-BSR was compared to recently released pan-genome software packages including GET_HOMOLOGUES (*Contreras-Moreira & Vinuesa, 2013*), ITEP (*Benedict et al., 2014*), and PGAP (*Zhao et al., 2012*). A set of 11 *Streptococcus pyogenes* genomes was chosen for the comparative analysis, as it was also used as a test set in the PGAP publication; the comparative analysis and results are shown in [Table 1](#). Overall, the size of the core genome was comparable between methods, with LS-BSR (BLASTN) and GET_HOMOLOGUES calculating differing core genome numbers compared to the other methods. However, small differences were expected due to differing thresholds and clustering algorithms. Based on these results, LS-BSR represent a significant improvement in terms of speed and ease of use compared to comparable methods, while having comparable utility.

Pan-genome analyses

One application in comparative genomics is the analysis of the pan-genome, or the combined genome, of isolates within a species. Post matrix-building scripts are available to visualize the pan-genome of a given dataset. One script (BSR_to_PANGP.py) creates a matrix compatible with PanGP (*Zhao et al., 2014*), for visualization of pan-genome statistics. The pan_genome_stats.py script provides data that can be used to visualize the conservation of CDSs at different genome depths ([Fig. 4A](#)). An additional script randomly subsamples the CDS distribution at all depths and produces data that can be plotted to visualize core genome convergence ([Fig. 4B](#)), accumulation of CDSs ([Fig. 4C](#)), and the number of unique CDSs for each genome analyzed ([Fig. 4D](#)). All analyses were conducted on a set of 100 *E. coli* genomes, with 100 iterations.

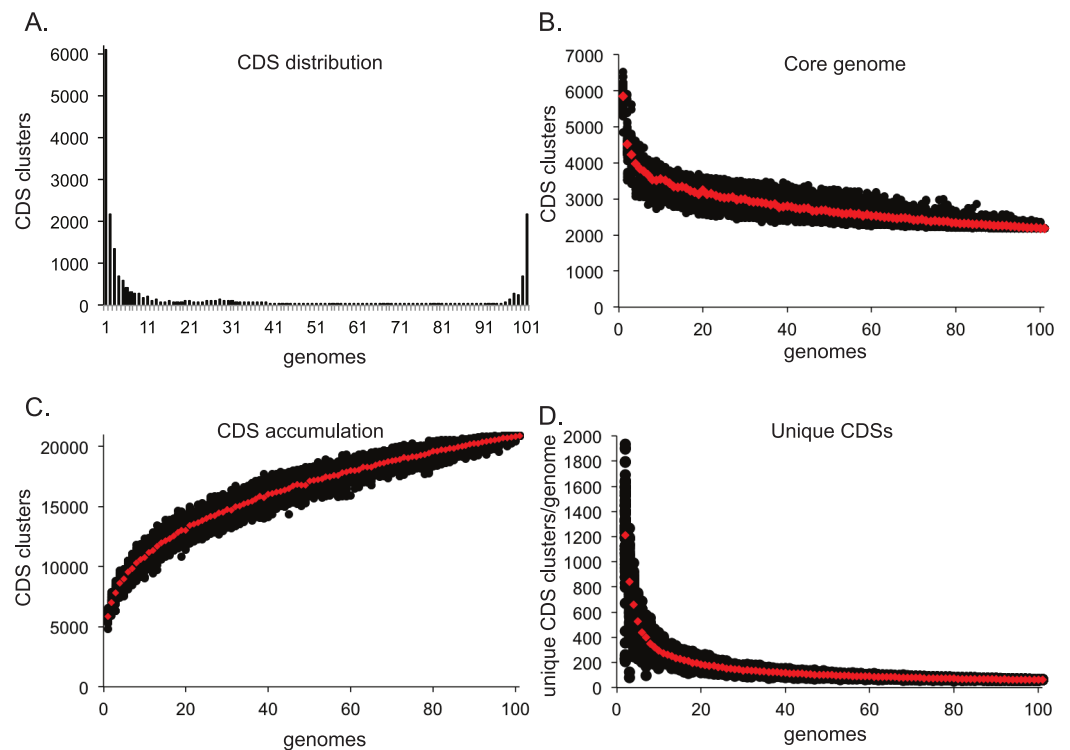


Figure 4 Pan-genome plots generated from LS-BSR output. Analyses were conducted on a set of 100 *Escherichia coli* genomes. The distribution of coding region sequences (CDSs) across the set of genomes surveyed is shown in A. A supplemental script can be used to better understand the convergence of the core genome (B), the accumulation of CDSs (C), and the number of unique CDSs for each genome analyzed (D); each analysis was conducted with 100 random sum-samplings and means are depicted with red diamonds.

CONCLUSIONS

The LS-BSR method can rapidly compare the gene content of a relatively large number of bacterial genomes in either draft or complete form, though with more fragmented assemblies LS-BSR is likely to perform sub-optimally. As sequence read lengths improve, assembly fragmentation should become less problematic due to more contiguous assemblies. LS-BSR can also be used to rapidly screen a collection of genomes for the conservation of known virulence factors or genetic features. By using a range of peptide relatedness, instead of a defined threshold, homologs and paralogs can also be identified for further characterization.

LS-BSR is written in Python with many steps conducted in parallel. This allows the script to scale well from hundreds to thousands of genomes. The LS-BSR method is a major improvement on a previous BSR implementation in terms of speed, ease of use, and utility. As more WGS data from bacterial genomes become available, methods will be required to quickly compare their genetic content and perform pan-genome analyses. LS-BSR is an open-source software package to rapidly perform these comparative genomic workflows.

ACKNOWLEDGEMENTS

Thanks to Darrin Lemmer for his critical review of the LS-BSR code.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was funded by the NAU Technology and Research Initiative Fund (TRIF). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
NAU Technology and Research Initiative Fund (TRIF).

Competing Interests

Jason W. Sahl is employed by The Translational Genomics Research Institute, and Paul S. Keim is the Director of The Translational Genomics Research Institute.

Author Contributions

- Jason W. Sahl conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, reviewed drafts of the paper.
- J. Gregory Caporaso analyzed the data.
- David A. Rasko conceived and designed the experiments, contributed reagents/materials/analysis tools.
- Paul Keim conceived and designed the experiments.

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj.332>.

REFERENCES

- Altenhoff AM, Gil M, Gonnet GH, Dessimoz C. 2013. Inferring hierarchical orthologous groups from orthologous gene pairs. *PLoS ONE* 8:e53786 DOI 10.1371/journal.pone.0053786.
- Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* 25:3389–3402 DOI 10.1093/nar/25.17.3389.
- Angiuoli SV, Salzberg SL. 2011. Mugsy: fast multiple alignment of closely related whole genomes. *Bioinformatics* 27(3):334–342 DOI 10.1093/bioinformatics/btq665.
- Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, Formsma K, Gerdes S, Glass EM, Kubal M, Meyer F, Olsen GJ, Olson R, Osterman AL, Overbeek RA, McNeil LK, Paarmann D, Paczian T, Parrello B, Pusch GD, Reich C, Stevens R, Vassieva O, Vonstein V, Wilke A, Zagnitko O. 2008. The RAST Server: rapid annotations using subsystems technology. *BMC Genomics* 9:75 DOI 10.1186/1471-2164-9-75.

- Benedict MN, Henriksen JR, Metcalf WW, Whitaker RJ, Price ND. 2014. ITEP: an integrated toolkit for exploration of microbial pan-genomes. *BMC Genomics* 15:8 DOI 10.1186/1471-2164-15-8.
- Benson DA, Karsch-Mizrachi I, Clark K, Lipman DJ, Ostell J, Sayers EW. 2012. GenBank. *Nucleic Acids Research* 40:D48–D53 DOI 10.1093/nar/gkr1202.
- Bork P, Hugenholtz P, Kunin V, Raes J, Harris JK, Spear JR, Walker JJ, Ivanova N, von Mering C, Bebout BM, Pace NR. 2008. Millimeter-scale genetic gradients and community-level molecular convergence in a hypersaline microbial mat. *Molecular Systems Biology* 4:198–198.
- Contreras-Moreira B, Vinuesa P. 2013. GET_HOMOLOGUES, a versatile software package for scalable and robust microbial pangenome analysis. *Applied and Environmental Microbiology* 79:7696–7701 DOI 10.1128/AEM.02411-13.
- DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, Rivas MA, Hanna M, McKenna A, Fennell TJ, Kernytsky AM, Sivachenko AY, Cibulskis K, Gabriel SB, Altshuler D, Daly MJ. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics* 43:491–498 DOI 10.1038/ng.806.
- Edgar RC. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26:2460–2461 DOI 10.1093/bioinformatics/btq461.
- Enright AJ, Van Dongen S, Ouzounis CA. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research* 30:1575–1584 DOI 10.1093/nar/30.7.1575.
- Hazen TH, Sahl JW, Fraser CM, Donnenberg MS, Scheutz F, Rasko DA. 2013. Refining the pathovar paradigm via phylogenomics of the attaching and effacing *Escherichia coli*. *Proceedings of the National Academy of Sciences of the United States of America* 110:12810–12815 DOI 10.1073/pnas.1306836110.
- Hyatt D, Chen GL, Locascio PF, Land ML, Larimer FW, Hauser LJ. 2010. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* 11:119 DOI 10.1186/1471-2105-11-119.
- Kent WJ. 2002. BLAT—the BLAST-like alignment tool. *Genome Research* 12:656–664 DOI 10.1101/gr.229202. Article published online before March 2002.
- Li L, Stoeckert Jr CJ, Roos DS. 2003. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research* 13:2178–2189 DOI 10.1101/gr.1224503.
- Price MN, Dehal PS, Arkin AP. 2010. FastTree 2—approximately maximum-likelihood trees for large alignments. *PLoS ONE* 5:e9490 DOI 10.1371/journal.pone.0009490.
- Rasko DA, Myers GS, Ravel J. 2005. Visualization of comparative genomic analyses by BLAST score ratio. *BMC Bioinformatics* 6:2 DOI 10.1186/1471-2105-6-2.
- Rasko DA, Rosovitz MJ, Myers GS, Mongodin EF, Fricke WF, Gajer P, Crabtree J, Sebahia M, Thomson NR, Chaudhuri R, Henderson IR, Sperandio V, Ravel J. 2008. The pangenome structure of *Escherichia coli*: comparative genomic analysis of *E. coli* commensal and pathogenic isolates. *Journal of Bacteriology* 190:6881–6893 DOI 10.1128/JB.00619-08.
- R Core Team. 2013. R: a language and environment for statistical computing. Vienna: R Foundation for Statistical Computing. Available at <http://www.R-project.org>.
- Saeed AI, Bhagabati NK, Braisted JC, Liang W, Sharov V, Howe EA, Li J, Thiagarajan M, White JA, Quackenbush J. 2006. TM4 microarray software suite. *Methods in Enzymology* 411:134–193.

- Sahl JW, Gillece JD, Schupp JM, Waddell VG, Driebe EM, Engelthaler DM, Keim P. 2013.** Evolution of a pathogen: a comparative genomics analysis identifies a genetic pathway to pathogenesis in *Acinetobacter*. *PLoS ONE* **8**:e54287 DOI [10.1371/journal.pone.0054287](https://doi.org/10.1371/journal.pone.0054287).
- Sahl JW, Steinsland H, Redman JC, Angiuoli SV, Nataro JP, Sommerfelt H, Rasko DA. 2011.** A comparative genomic analysis of diverse clonal types of enterotoxigenic *Escherichia coli* reveals pathovar-specific conservation. *Infection and Immunity* **79**:950–960 DOI [10.1128/IAI.00932-10](https://doi.org/10.1128/IAI.00932-10).
- Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, Sahl JW, Stres B, Thallinger GG, Van Horn DJ, Weber CF. 2009.** Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology* **75**:7537–7541 DOI [10.1128/AEM.01541-09](https://doi.org/10.1128/AEM.01541-09).
- Sims GE, Kim SH. 2011.** Whole-genome phylogeny of *Escherichia coli*/*Shigella* group by feature frequency profiles (FFPs). *Proceedings of the National Academy of Sciences of the United States of America* **108**:8329–8334 DOI [10.1073/pnas.1105168108](https://doi.org/10.1073/pnas.1105168108).
- Tettelin H, Riley D, Cattuto C, Medini D. 2008.** Comparative genomics: the bacterial pan-genome. *Current Opinion in Microbiology* **11**:472–477 DOI [10.1016/j.mib.2008.09.006](https://doi.org/10.1016/j.mib.2008.09.006).
- Zhao Y, Jia X, Yang J, Ling Y, Zhang Z, Yu J, Wu J, Xiao J. 2014.** PanGP: a tool for quickly analyzing bacterial pan-genome profile. *Bioinformatics* **2014**:btu017v1-btu017 DOI [10.1093/bioinformatics/btu017](https://doi.org/10.1093/bioinformatics/btu017).
- Zhao Y, Wu J, Yang J, Sun S, Xiao J, Yu J. 2012.** PGAP: pan-genomes analysis pipeline. *Bioinformatics* **28**:416–418 DOI [10.1093/bioinformatics/btr655](https://doi.org/10.1093/bioinformatics/btr655).