[figure1.eps]

caption: “Figure 1: Detailed results for the serotyping and virulence factor subtyping task. While data storage in Spfy is graph-based, a familiar tabular structure is presented to users. The genome file, GCA 001911825.1 ASM191182v1 genome.fna, was analyzed with the determined serotype associated with the file, and virulence factors associated with the contigouous DNA sequences they were found on. The Start/Stop positions on the contig., are provided along with the Percent Identity (Cutoff) used in the analysis.”

[figure2.eps]

caption: “Figure 2: E. coli genomes of serotype O157 (903 genomes) compared against genome of serotype O26 (291 genomes), for statistically significant differences in the carriage of 129 AMR genes. Fisher’s Exact Test is used by Spfy for these comparisons.”

[figure3.eps]

caption: “Figure 3: Structure of the Spfy graph database. Brackets highlight the source of different data points and the software it was generated from. Data are added in as the analysis modules complete, at varying times, and the overall connections are inferred by the database. Non-bracketed sections are sourced from the uploaded genome files or user-supplied metadata. Figure was generated using http://www.visualdataweb.de/webvowl/.”

[figure4.eps]

caption: “Figure 4: The results interface for submitted tasks. Cards represent individual tasks submitted by the user, such as checking the Database status, subtyping of single genomes, population comparisons, or subtyping of multiple genomes.”

[figure5.eps]

caption: “Figure 5: The Docker containers used in Spfy. Arrows represent the connections between different containers, and the entire platform can be recreated with a single command using its Docker-Compose definition. Users access the platform using the ReactJS based website, which makes requests to the Flask API. Any requested analysis task is distributed to the Redis Task Queue and data files are stored in a Docker volume. MongoDB stores a hash table for efficient duplicate checking of results in Blazegraph.”

[figure6.eps]

caption: “Figure 6: Runtimes of Fisher’s Exact Test depending on the number of nodes/attributes involved in the comparison. A: Runtimes as the number of genomes increased for a fixed (107) number of targets per genome. B: Runtimes as the number of targets increased for a fixed (116) number of genomes per target. C: Overall runtimes as the total number of targets retrieved increased; the total number of targets was calculated as follows: (Number Genomes Group A + Number Genomes Group B) x Number Targets per Genome. In all cases, a linear increase in runtime was observered as the number of targets or genomes increased.”

[figure7.eps]

caption: “Figure 7: Total runtimes of Spfy’s analysis modules for batches of files. The blue line indicates the actual time to completion after accounting for parallelization; 50 files are analyzed in 45 minutes and 100 files in 89 minutes.”

[table1.csv]

caption: “Table 1: Comparison of four bioinformatic pipelines and their underlying database. Functionally, Spfy integrates different analysis modules as in BAP while also merging large datasets as in PATRIC.”