Using KBase to Assemble and Annotate Prokaryotic Genomes

Authors

Benjamin Allen[1], Meghan Drake[1], Nomi Harris[2], Tarah Sullivan[3]

[1]Bioinformatics Group, Oak Ridge National Laboratory
[2]Environmental Genomics & Systems Biology Group, Lawrence Berkeley National Laboratory
[3]Department of Crop and Soil Sciences, Washington State University

Notice of Copyright: This manuscript has been co-authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

Abstract

The DOE Systems Biology Knowledgebase (KBase, http://kbase.us/) is an open access bioinformatics software and data platform for analyzing plants, microbes, and their communities. KBase enables scientists to create, execute, collaborate on, and share reproducible analyses of their biological data in the context of public data and private collaborator data. For microbiologists researching prokaryotes, KBase offers analysis tools for performing quality control and assessment of Next-Generation Sequencing reads, *de novo* assembly, genome annotation, and tools for analyzing structural and functional features of genomes. This unit demonstrates an example workflow for taking a comparative and iterative approach to assembly and annotation of prokaryotic genomes using KBase that can be used by microbiologists seeking to perform isolate analysis in a rapid and reproducible fashion.

Keywords: KBase, genomics, assembly, annotation, systems biology

Introduction

The advent of high-throughput sequencing technology has enabled more researchers working in smaller teams to generate large amounts and diverse types of biological data. Computational

biology software has developed alongside sequencing technology, with a variety of open-source tools available for researchers to run essential workflows for quality control, processing, analysis, and visualization of data. However, data fidelity, provenance, access, and reproducibility remain major challenges for open-science approaches to computational biology. The DOE Systems Biology Knowledgebase – KBase (Arkin et al., 2016; see Figure 1) – is a software and data platform designed to address these challenges within systems biology by enabling researchers to conduct experiments and perform analyses in a single environment with a unified data model. KBase is the first large-scale bioinformatics system that enables users to upload their own data, analyze it alongside collaborator and public data, and build increasingly realistic models, while sharing and publishing their workflows and conclusions within a single, integrated environment. By creating an environment for reproducible scientific analysis and facilitating collaboration, KBase seeks to accelerate the pace of systems biology research. KBase users have applied the system to address a range of scientific problems, including comparative genomics of plants, prediction of microbiome interactions, and metabolic modeling of environmental and engineered microbes.

For microbiologists, assembling prokaryotic Next-Generation Sequencing reads into contigs and constructing an annotated genome with functional information about the coding DNA sequences are necessary first steps for comparative genomics, phylogenetic analysis, or metabolic modeling. In KBase, users can quickly and easily perform *de novo* assembly on prokaryotic DNA reads using multiple assemblers, and then run two separate annotation tools on the assemblies, calling genes and other genomic features and assigning biological functions, to generate an annotated genome. This genome can be used in downstream analysis within KBase or downloaded and used with other software platforms.

Figure 1. *The KBase homepage (http://kbase.us) has links for creating an account, logging into KBase, reference documentation, research highlights, tutorials, and user support.*

Quality Assessment and *de novo* Assembly of Prokaryotic DNA Short Reads

This protocol demonstrates a simplified workflow for performing sequencing quality assessment, *de novo* assembly, and assembly assessment using some example paired-end sequencing reads generated from DNA isolated from a culture of an unknown *Rhodobacter* species sequenced on an Illumina platform. This workflow could be utilized by any researcher who has obtained either raw or quality controlled single- or paired-end prokaryotic DNA reads in FASTQ format from an

Illumina HiSeq or MiSeq platform. Instructions for uploading single- or paired-end reads to KBase from a personal computer are provided in Support Protocol 2.

The FastQC app in KBase enables users to perform quality assessment and control of sequence data. FastQC is an open source quality control tool that provides an overview and graphical assessments of sequencing reads. A common use of FastQC is to determine if any trimming or adapter removal is necessary for raw sequence data coming from high throughput sequencing platforms. For this example, our data has already undergone quality control and therefore does not require any additional trimming or adapter removal, although tools to perform these tasks are available in KBase. Detailed instructions for reading the FastQC reports can be found at: *https://www.bioinformatics.babraham.ac.uk/projects/fastqc/*

KBase has integrated many different open source *de novo* assemblers into its system by wrapping existing assembly algorithms into KBase apps. The two assembly apps used for this protocol are Velvet and SPAdes. Both Velvet and SPAdes are capable assemblers for prokaryotic reads, although SPAdes has been optimized for single cell data. Velvet is a de Bruijn graph based assembler that combines multiple algorithms for the construction, simplification, and error correction of assembly graphs (Zerbino & Birney, 2008). Velvet combines overlapping *k*-mers into nodes, traces paths between nodes, and then constructs the assembly graph by mapping overlapping reads hashed into *k*-mers to the corresponding paths. The assembler uses topological analysis to eliminate errors caused by biological variation or pre-sequencing artifacts.

SPAdes is also a de Bruijn graph based assembler unique in its utilization of multisized de Bruijn graphs to generate a final assembly from multiple *k*-mers combined with error and mismatch correction tools (Bankevich et al., 2012). SPAdes begins its assembly process by using multisized de Bruijn graphs for constructing the assembly graph while detecting and removing chimeric reads. Next, distances between the *k*-mers are estimated for mapping the edges of the assembly graph. Afterwards, a paired assembly graph is constructed and SPAdes outputs a set of contiguous DNA sequences (contigs). In KBase, running these assembly apps on a set of reads generates an Assembly object, which contains the assembled contigs and can be used in downstream analyses or downloaded as a FASTA file.

After performing multiple assemblies on a set of sequencing reads, it can be useful to compare the performance of each assembler by analyzing the quality of each assembly. QUAST is an open source assembly quality assessment tool that allow users to compare summary statistics of multiple assemblies (Gurevich, Saveliev, Vyahhi, & Tesler, 2013). These statistics can be used to determine which assembly is optimal for feeding into downstream annotation pipelines.

Materials

An internet-accessible computer with a recent version Chrome, Firefox, or Safari

A KBase account. KBase is free to sign up for and use. All functionality detailed in this protocol requires being logged in with a KBase account: http://kbase.us/sign-up-for-a-kbase-account/

Saved Narrative (Support Protocol 1)

1. Add the example data to the Narrative.

    a. Click the *Add Data* button in the Data Panel on the left of the screen.
    b. The Data Browser will slide out, with tabs that show several data sources. Select the *Example* tab and look at the *Example Sequence Assembly Inputs* heading.
    c. Find "*rhodo.art.q20.PE.reads*" and add this set of paired-end reads to the Narrative by mousing over it and clicking the *Add* button that appears at its left.
    d. Exit the Data Browser by clicking either the *Close* button at the bottom right of the browser window or the arrow at the top of the Data Panel.
    e. The Data Panel will update to show the *rhodo.art.q20.PE.reads* dataset added (Figure 2).

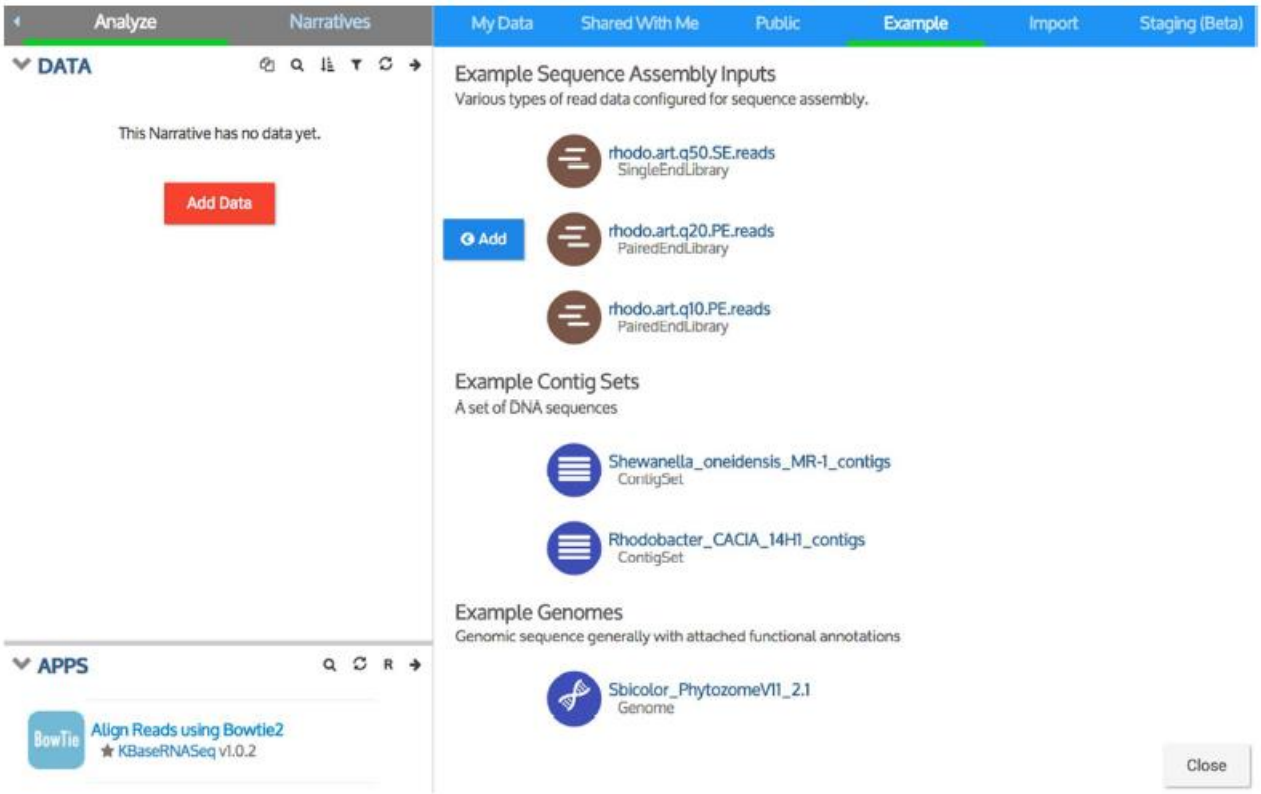

Figure 2. *Adding example* Rhodobacter *sp. reads to the Narrative.*

2. Assess the quality of the reads using the FastQC - Read Quality Analysis app.

    a. Locate the Apps panel near the bottom left of the Narrative. The Apps panel shows all the analysis tools available in KBase, sorted alphabetically by default.
    b. Scroll through the list to locate the FastQC - Read Quality Analysis app.
    c. Click the name of the app to load it into the Narrative.
    d. The FastQC - Read Quality Analysis app will now appear as a new cell within the Narrative.
    e. Load the example reads into the app by clicking the dropdown menu next to Reads under the section labeled Input Objects. This dropdown menu will be populated with all data in the Narrative that is compatible as input for the app. Select the reads labeled *rhodo.art.q20.PE.reads.*
    f. Click the green Run button near the top left of the app cell to start the analysis.
    g. After a few minutes, the analysis will complete and a FastQC report will be generated with statistics about the quality of the reads.

    The output of the FastQC app is an HTML report displaying graphical and statistical information about the sequencing reads (Figure 3). By default, this report appears in the "Result" tab of the app. However, it can also be displayed in a separate browser window by clicking the "View report in separate window" button that appears in the top right of the cell under "Report." For a paired-end read library, the first page of the report displays an overview for the forward reads, while the second page displays an overview for the reverse reads. Under the "Summary" section of the output report, there is a list of hyperlinked analysis modules that contain information about various aspects of the reads. Next to each analysis module is a symbol indicating how the module evaluated each aspect of the reads. A green circle with checkmark indicates that the reads matched expected values for acceptable quality for that module, while a red circle with an "X" means that the reads did not pass a quality check by that module and may need to undergo quality control. For instance, a red circle with an "X" next to "Per base sequence content" could mean that GC content of the read has gone above an expected threshold after a certain position, and that trimming is required on the ends of the reads. Additionally, FastQC will display a yellow circle with an exclamation point to indicate that the user should take note of a specific aspect of the reads for performing quality control. For example, this caution symbol make appear next to the "Overrepresented sequences" section to indicate that FastQC identified adapters that need to be removed before proceeding onto assembly. After running FastQC on this example data, it appears that these reads are of high quality based on the summary of the output report and can be used for assembly without further quality control.

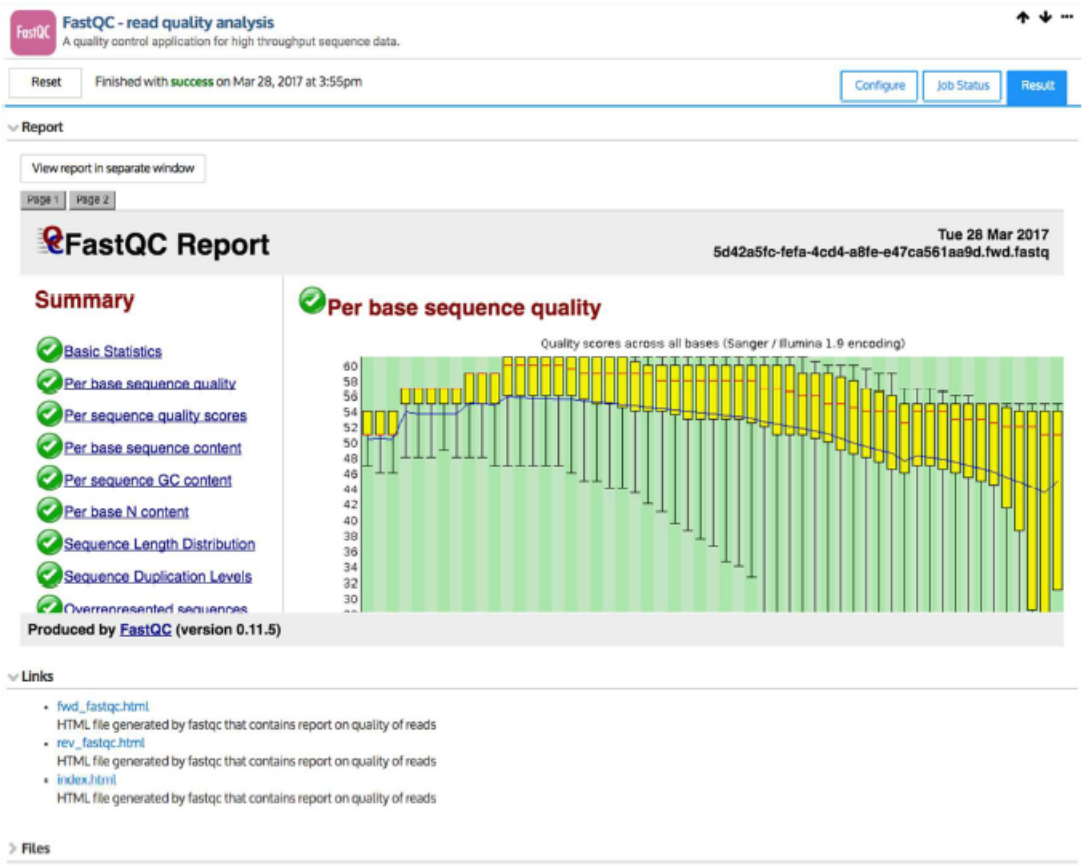

Figure 3. *FastQC Report for the unknown* Rhodobacter *sp. reads.*

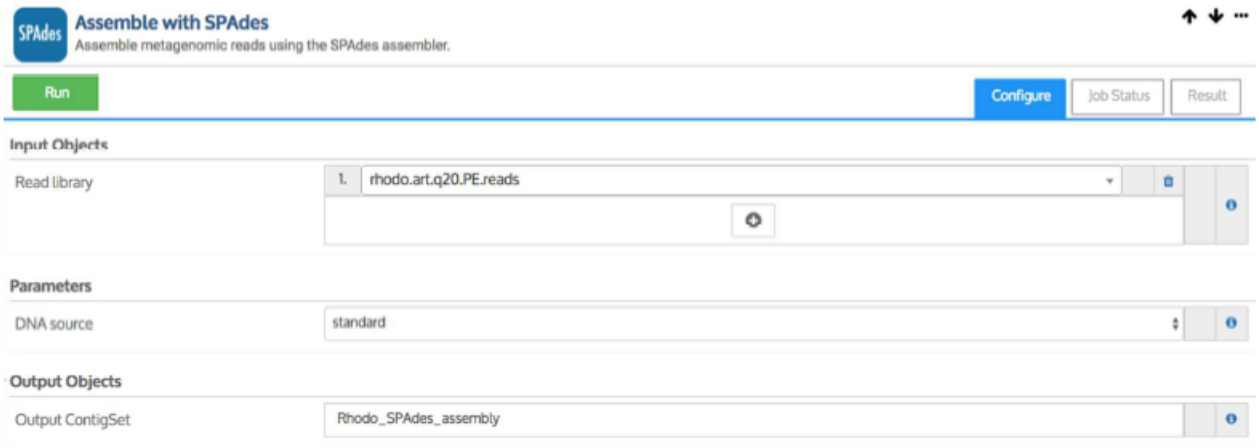

Figure 4. *Configuration of the Assemble with SPAdes app before running the assembly job.*

3. Perform *de novo* assembly using the SPAdes assembler app.

    a. Go to the Apps panel. Scroll through the list to locate the Assemble with SPAdes app (Figure 4).
    b. Click the name of the app to load it into the Narrative.

c. In the first parameter field, Read Library, select "*rhodo.art.q20.PE.reads*" from the dropdown menu.
d. For DNA Source, select "standard."
e. Provide a name for the output assembly. For instance, "Rhodo_SPAdes_assembly."
f. Click the green Run button to start the analysis. The app cell will show the job as queued, then running.
g. Following assembly by this app, an Assembly called "Rhodo_SPAdes_assembly" will be created and added to the Data Panel. An assembly statistics report will be generated in the "Results" tab of the app.

4. Perform *de novo* assembly using the Velvet assembler app (Figure 5).

a. Locate the Assemble with Velvet app in the Apps panel.
b. Click the name of the app to load it into the Narrative.
c. In the first parameter field, Read Library, select "*rhodo.art.q20.PE.reads*" from the dropdown menu.
d. Provide a name for the output assembly. For instance, "Rhodo_Velvet_assembly."
e. Click the green Run button to start the analysis. The app cell will show the job as queued, then running.
f. Following assembly by this app, an Assembly called "Rhodo_Velvet_assembly" will be created and added to the Data Panel. An Output cell summarizing the outcome of the assembly process will also be added to the Narrative.

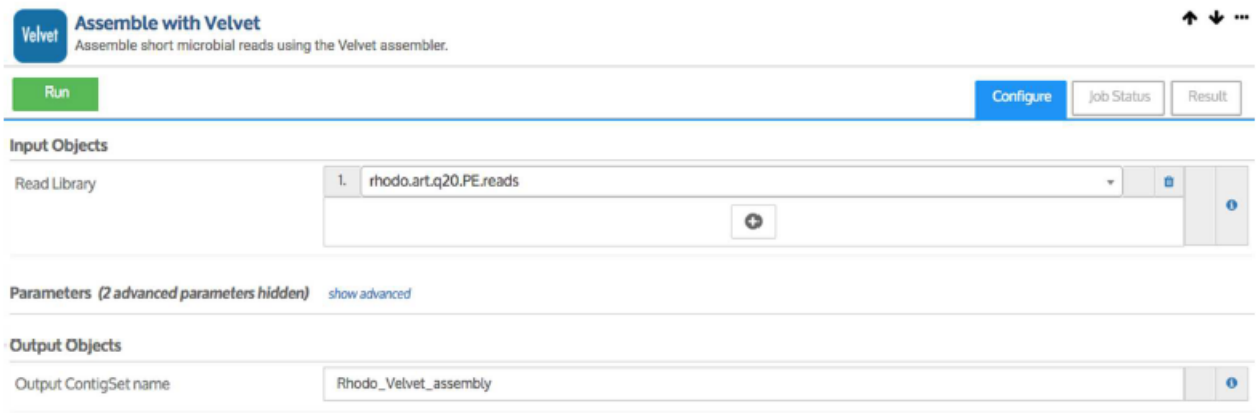

Figure 5. *Configuration of the Assemble with Velvet app before running the assembly job.*

5. Compare the assemblies using the QUAST assembly comparison and analysis app.

a. Locate the QUAST app in the Apps panel.
b. Click the name of the app to load it into the Narrative.

c. In the Assemblies field, click the "+" button to select the assemblies for comparison. Add the "Rhodo_SPAdes_assembly" and the "Rhodo_Velvet_assembly" generated in the previous steps.
d. Click the green Run button to start the analysis. The app cell will show the job as queued, then running.
e. Once the app is complete, an assembly quality assessment will be generated (Figure 6).

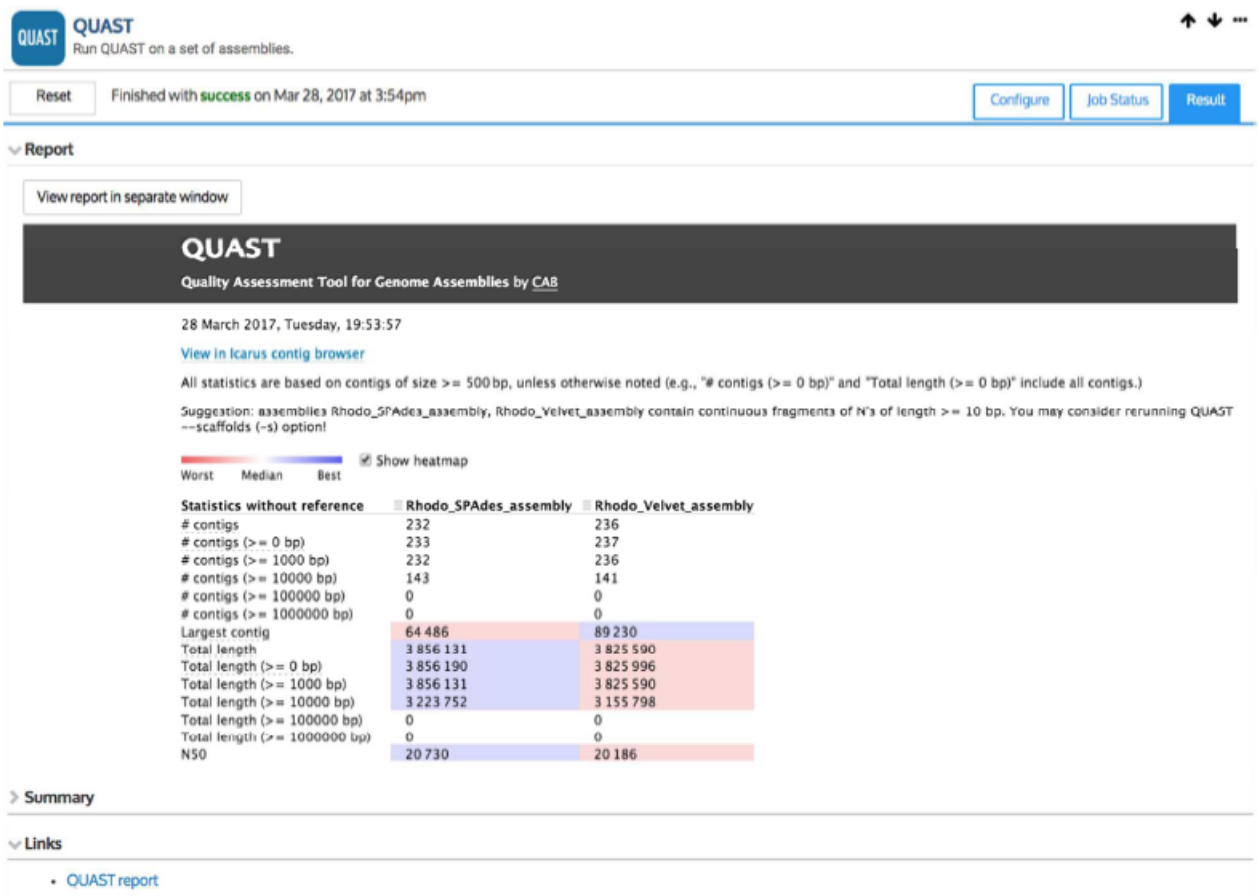

Figure 6. *QUAST Report with statistics and visualizations for assessing the assemblies.*

The output of QUAST contains important statistics for choosing the best assembly, such as the total number of contigs, number of genes, and the N50 values of the contigs. Additionally, the app provides an easy to read heatmap that highlights the "best" performing statistics in blue and the "worst" ones in red. An optimal assembly run would generally be the one with the fewest number of contigs, the highest N50 value, lowest L50, the largest number of genes predicted of a base pair length above a reasonable threshold (greater than or equal to 300 base pairs in coding DNA sequence length), and the fewest mismatches. The N50 value is a reference statistic that acts as a weighted median for evaluating assemblies. It means that half of the

assembly is contained in contigs equal to or larger than this value. L50 is a corollary statistic that equals the number of contigs whose lengths summed together is the N50 value. However, one should also consider the distribution of contig sizes throughout the assembly or whether an assembly has a high number of misreads relative to its contig sizes. After running QUAST on both the SPAdes and Velvet assemblies, it appears that SPAdes has the best statistics overall. The SPAdes assembly has the fewest number of contigs (232 SPAdes vs. 238 Velvet), the largest number of predicted genes greater than or equal to 300 base pairs in length (3352 SPAdes vs 3316 Velvet), comparable N50/L50 values (20730/58 SPAdes vs. 20768/58 Velvet), and the fewest number of mismatches (914 SPAdes vs 22151 Velvet). Because it produced the better of the two assemblies, the output of the SPAdes assembler will be used for downstream analysis.

Using Narratives in KBase

KBase users can create shareable, reproducible workflows called *Narratives* (Figure 7) that include data, analysis steps, results, visualizations and commentary. Users can access all the tools and data within KBase via Narratives. Please see the Narrative Quick Start (*http://kbase.us/narrative-quick-start/)* for a brief introduction to Narratives in KBase.

Figure 7. *Overview of a KBase Narrative. A Narrative is a shareable, reproducible computational experiment that can include data, analysis steps, results, visualizations, and commentary.*

Materials

An internet-accessible computer with a recent version Chrome, Firefox, or Safari

A KBase account. KBase is free to sign up for and use. All functionality detailed in this protocol requires being logged in with a KBase account: *http://kbase.us/sign-up-for-a-kbase-account/*

1. Sign up for a KBase account

2. Create a new Narrative. The first step for all types of analysis in KBase is to sign in, create a Narrative, provide it a name, and save the Narrative. This Narrative will contain all the data and tools used in this protocol.

    a. Go to the KBase homepage at *http://kbase.us/*.
    b. Click the Sign In button in the top right corner to log in to KBase
    c. After signing in, the Dashboard page will appear
    d. Click the blue "+ New Narrative" button to open a new Narrative
    e. A new window will open with an empty Narrative (Figure 8). Click the text at the top left of the screen that says "Untitled." This will open a dialog box to provide a name for the Narrative. Provide a name for the Narrative and click OK.
    f. Click the Save button at the top right of the screen to save the Narrative.

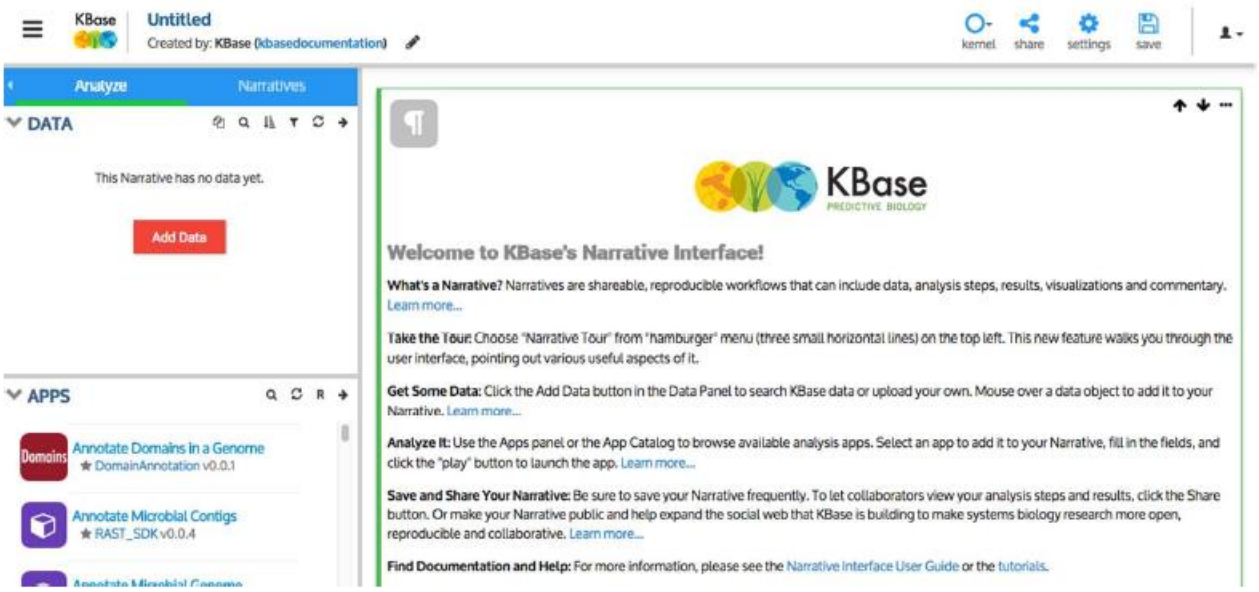

Figure 8. *An empty KBase Narrative ready for adding all the data and apps associated with an analysis.*

Uploading Short Reads to KBase from a Desktop Computer

Single-end and paired-end NGS short read libraries from Illumina sequencing platforms can be uploaded into KBase in FASTQ format for assembly.

Materials

An internet-accessible computer with a recent version Chrome, Firefox, or Safari

A KBase account. KBase is free to sign up for and use. All functionality detailed in this protocol requires being logged in with a KBase account: *http://kbase.us/sign-up-for-a-kbase-account/*

Short reads from an Illumina sequencing platform in FASTQ format.

1. Upload short reads from a FASTQ file
    a. Click the *Add Data* button in the Data Panel on the left of the screen.
    b. The Data Browser will slide out, with tabs that show several data sources. Select the *Import* tab.
    c. Choose *Short Reads* as the data type.
    d. Click the *Next* button.
    e. Select the type of file you want to upload:
        i. If paired-end reads, click the *PAIRED END READS* tab (note that this will prompt you for up to two files; only one is needed if you are uploading an interleaved file; see Figure 9).
        ii. If single-end reads, select the *SINGLE END READS* tab.
    f. Click the *Select File* button(s) to open a file browser allowing you to choose a file from your computer.
    g. Name the short reads you are importing.
    h. Click the *Import* button.
    i. After the import process has completed, the short reads will appear in the Data Panel.

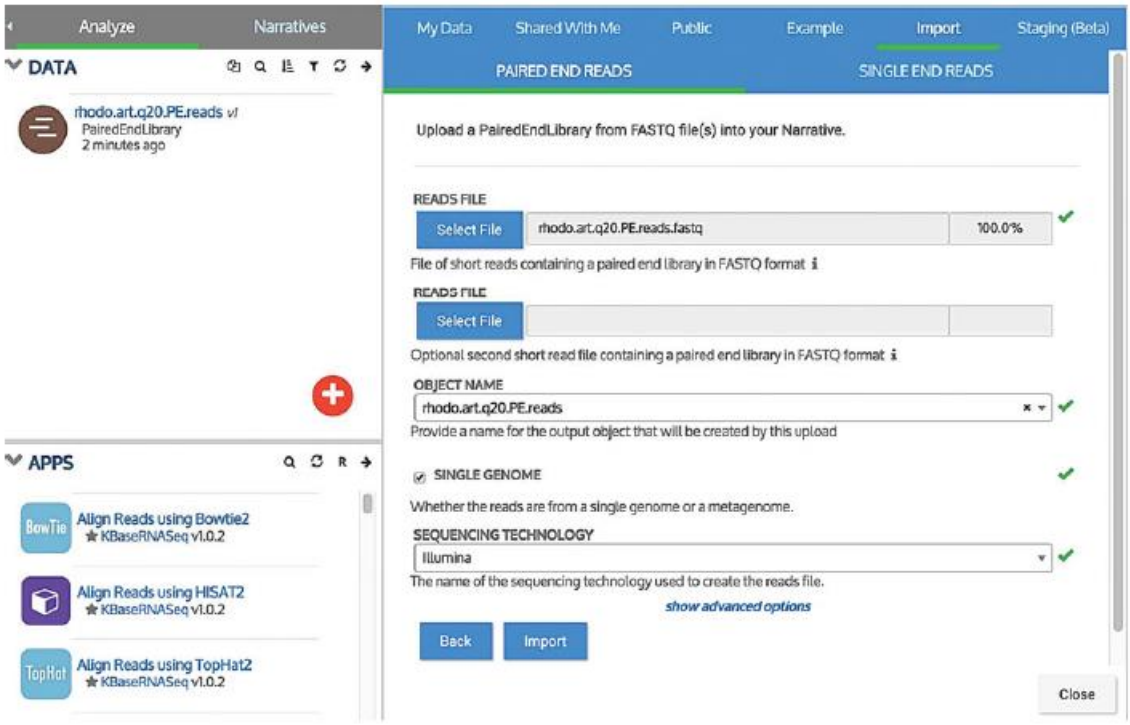

Figure 9. *Uploading an interleaved paired-end short read library as a FASTQ file.*

Genome Annotation and Comparison of Annotations

Genome annotation is a common and essential research activity for research biologists. Annotating the structural and functional features of an organism's genome allows for comparative analysis against other organisms' genomes and provides a reference for understanding the flow of information within a biological system. This protocol demonstrates how to annotate assembled contigs with structural and functional information. Prokaryotic genome annotation in KBase is provided by two apps: Annotate Microbial Contigs and Annotate Assembly with Prokka.

Annotate Microbial Contigs takes as input an Assembly object generated by one of the assembly apps or imported by the user and runs the sequence through a multi-step pipeline. First it applies two different algorithms, Prodigal and Glimmer3, to predict gene locations within the contigs. Next, the gene sequences are passed into the functional annotation pipeline in KBase, which is based on the RAST (Rapid Annotations using Subsystems Technology) toolkit (Overbeek *et al.*, 2014). This annotation pipeline assigns functions from the SEED Subsystems Ontology to genes using a fast *k*-mer based approach. Annotate Assembly with Prokka is a KBase app that calls the popular prokaryotic annotator Prokka. Prokka combines multiple open-source annotation tools in a quick and thorough annotation pipeline for prokaryotic sequences that can be applied iteratively when performing sequence analysis. First, Prokka uses Prodigal (Hyatt 2010) to

identify gene coordinates within the assembly. Then it uses a hierarchical system for calling gene annotations from existing databases, beginning with UniProt and then moving on to RefSeq. Finally, TIGRFAM and Pfam hidden Markov model databases are queried for domain-specific annotations.

The final output of both annotation apps is a genome data object that can be analyzed for specific biological features called by annotation, such as proteins or regulatory elements. This object can be explored in a tabular genome viewer that shows summary information about the Genome as well as a list of contigs and the genes that were annotated on each contig. This protocol will reveal how to look for some example genes related to the photosynthetic machinery of the example *Rhodobacter* species. The genome object can be used as input to other KBase analysis apps or be downloaded as a GenBank file.

Materials

An internet-accessible computer with a recent version Chrome, Firefox, or Safari

A KBase account. KBase is free to sign up for and use. All functionality detailed in this protocol requires being logged in with a KBase account: http://kbase.us/sign-up-for-a-kbase-account/

1. Annotate the SPAdes assembly using the Annotate Microbial Contigs app

    a. Locate the Annotate Microbial Contigs app in the Apps panel (Figure 10).
    b. Click the name of the app to add it to the Narrative.
    c. In the first parameter field, Assembled Contigs, select "Rhodo_SPAdes_assembly" from the dropdown menu.
    d. For Scientific Name, provide a name for the organism, for instance "Rhodobacter sp."
    e. Leave the Domain field set to "B (Bacteria)" and Genetic Code field set to "11 (Archaea, most Bacteria, most Virii, and some Mitochondria").
    f. Provide a name for the output annotated genome. For instance, "Rhodo_SPAdes_RAST."
    g. Click the green Run button to start the analysis. The app cell will show the job as queued, then running.
    h. Following annotation by this app, a Genome named "Rhodo_SPAdes_RAST" will be created and added to the Data Panel. An Output cell summarizing the outcome of the annotation process will have also been added to the Narrative (Figure 11).

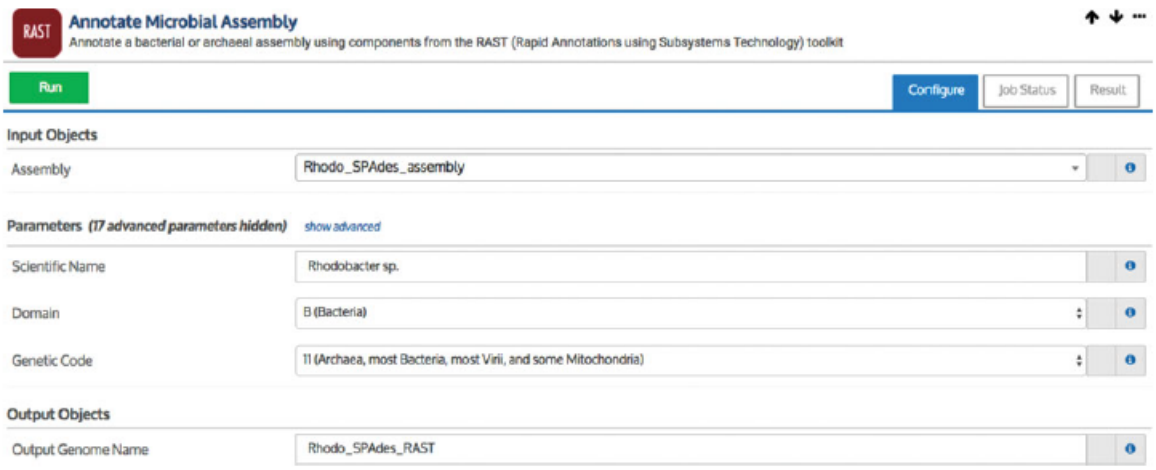

Figure 10. *Configuration of Annotate Microbial Assembly app before running the annotation job.*

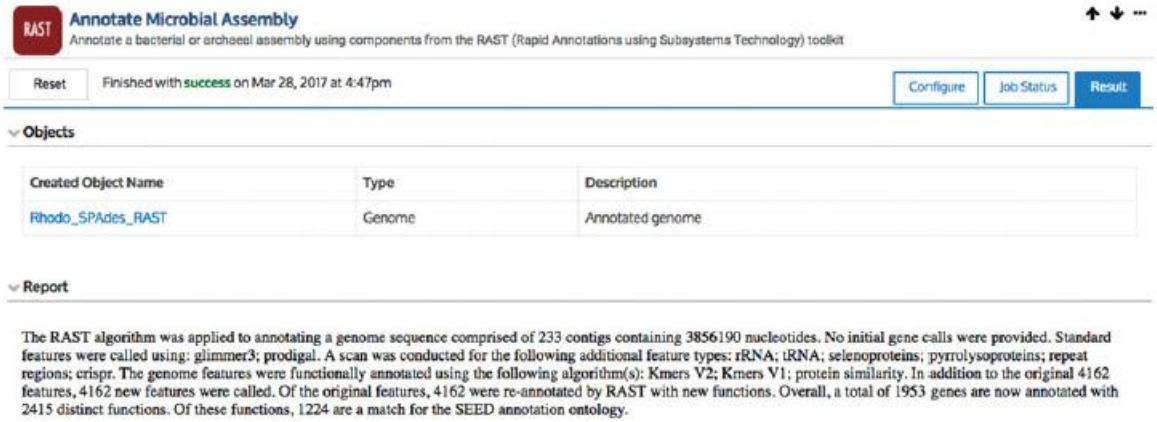

Figure 11. *Output of Annotate Microbial Assembly app after completing the annotation job.*

The output of the annotation step is an annotated Genome object, which is displayed below the input cell in a genome viewer. This viewer contains overview statistics about the Genome, as well as a list of contigs, and the genes that were called on each contig. Clicking the KBase Object Name in the viewer - "Rhodo_SPAdes_RAST" - will open a landing page for the Genome with additional information about data provenance, publications related to the Genome, and biological information derived from the assembly and annotation process. Running Annotate Microbial Contigs also generates a brief report with information about which algorithms were run to complete the annotation, the kinds of non-protein coding features were identified, the number of features annotated by RAST, the number of genes annotated with a distinct non-hypothetical function, and the number of genes with matches in the SEED system ontology.

2. Annotate the SPAdes assembly using the Annotate Assembly with Prokka app.

a. Locate the Apps panel in the bottom right of the Narrative. Scroll through the list to locate the Annotate Assembly with Prokka app (Figure 12).
b. Click the name of the app to add it to the Narrative.
c. In the first parameter field, Assembly, select "Rhodo_SPAdes_assembly" from the dropdown menu.
d. For Scientific Name, provide a name for the organism, for instance "Rhodobacter sp."
e. Provide a name for the output annotated genome. For instance, "Rhodo_SPAdes_Prokka."
f. Click the green Run button to start the analysis. The app cell will show the job as queued, then running.
g. Following annotation by this app, a Genome named "Rhodo_SPAdes_Prokka" will be created and added to the data panel. An Output cell summarizing the outcome of the annotation process will also be added to the Narrative (Figure 13).

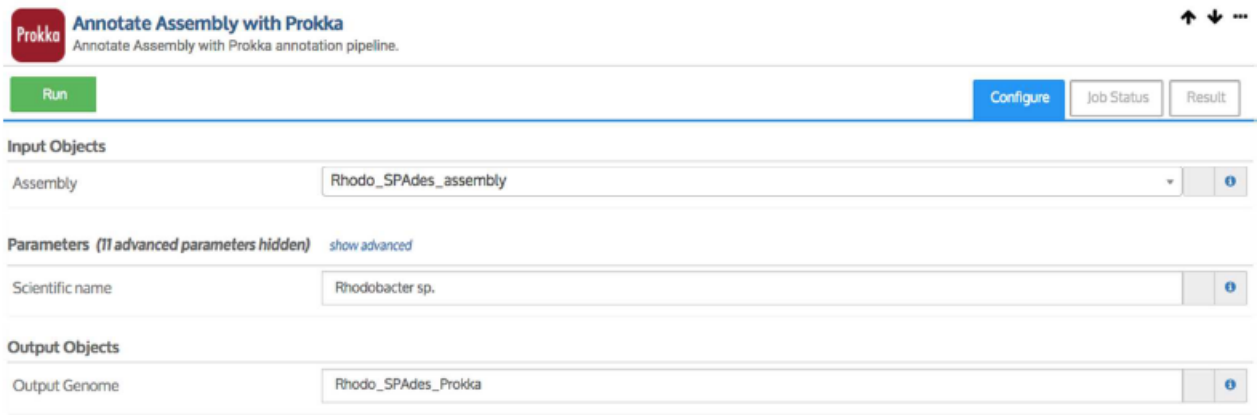

Figure 12. *Configuration of Annotate Assembly with Prokka app before running the annotation job.*

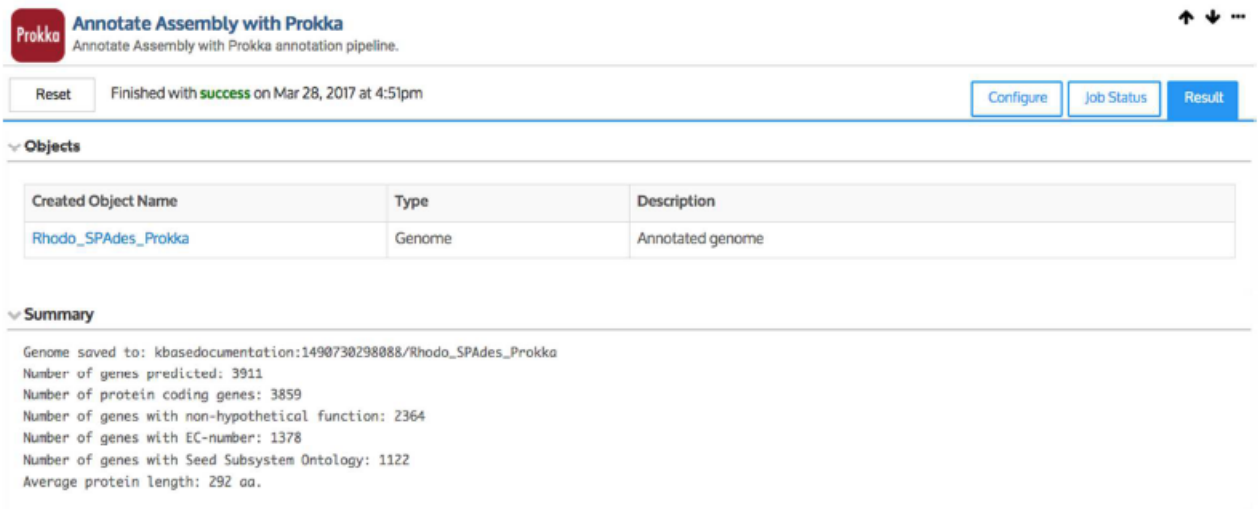

Figure 13. *Output of Annotate Assembly with Prokka app after completing the annotation job.*

The output of the annotation step is an annotated Genome object with a genome viewer that contains the same information outlined in the previous step. Running Annotate Assembly with Prokka also generates a brief report with information about which the number of genes predicted, the number of protein coding genes, the number of genes with non-hypothetical function, the number of genes with Enzyme Commission (EC) numbers, and the number of genes with matches in the SEED system ontology.

Notice that the output viewer for each genome has three tabs for reviewing the data: Overview, Browse Features, and Browse Contigs. The Overview section contains information about the assembled and annotated Genome. The Browse Features tab allows the user to explore the different biological functions annotated within the Genome. The Browse Contigs tab contains information about the length and the number of genes contained in each contig. These tables can be sorted by clicking on a column header for each field (e.g., Length). Clicking the same column header again will reverse the sort order. Users can even sort by more than one column simultaneously by clicking one column header and then Shift-clicking on others. Additionally, users can search for the length or number of features in a contig, or look for specific biological functions using the Search box in the top right corner of the viewer. To see more details about an entry under the Contigs and Features tabs, click on the entry to open an expanded view of it (Figure 14).

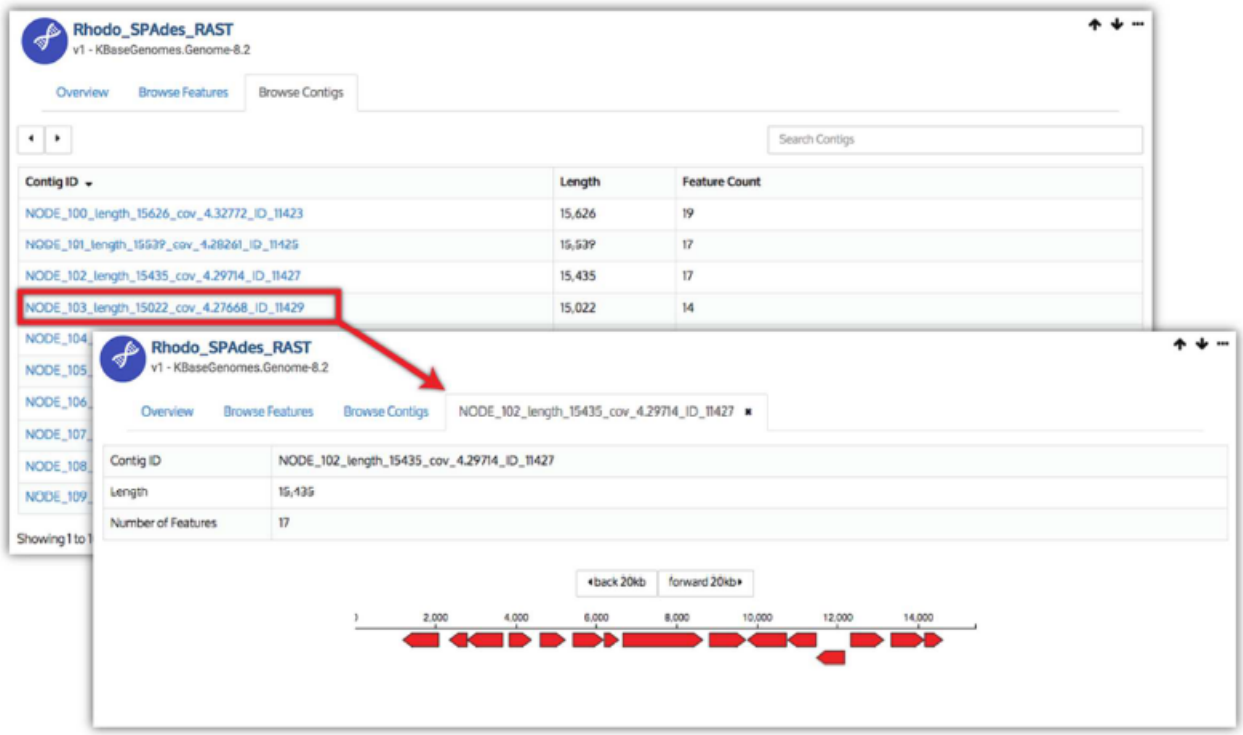

Figure 14. *Viewing a Contig using the Genome viewer. Click on a contig or gene name to open a panel showing more details about the contig or gene.*

3. Compare the results of the two annotation apps to select the better annotation.

Looking at the output reports for each annotated genome, we can see that there was variance in the total number of genes predicted and annotated with a non-hypothetical function. In general, it is assumed that genome with the most genes annotated with non-hypothetical functions will provide the most useful set of data for making biological inferences or feeding into downstream analysis. Overall, RAST (Figure 15) annotated more features in general (4162 RAST vs. 3911 Prokka), but RAST also annotated more genes with distinct non-hypothetical functions than Prokka (Figure 16; 2415 RAST vs. 2364). Therefore, it could be said that in a general sense, the RAST-annotated genome might be a better choice for continued analysis.

However, this is not absolutely the case in all circumstances. A researcher investigating the ability of an organism to perform a specific function would want to check for the presence or absence of annotations for the proteins or structures required for that function in the annotated genome object to determine if the genome was suitable for use in downstream analysis. For instance, it is known that in certain species of Rhodobacter, specific cytochrome complexes play a structural role in enabling efficient photosynthesis that is unique for these organisms via their organization into chromatophores (Scheuring et al., 2014). Therefore, searching for features annotated with "cytochrome" and looking at the differences between the annotations called by RAST and Prokka might aid in determining which of the two annotations is more suitable for continued investigation. Performing this search within the two annotations shows that RAST has annotated more features with functions related to cytochrome than Prokka (48 vs. 29 respectively). Therefore, RAST is likely to be the better annotation for evaluating this particular function within this organism. Thus, considering both the quantitative and qualitative configuration of any genomes generated using this protocol in KBase and consulting the literature regarding the specific organism or type of experiment are necessary for making appropriate decisions to get the most out of the data and tools in KBase.

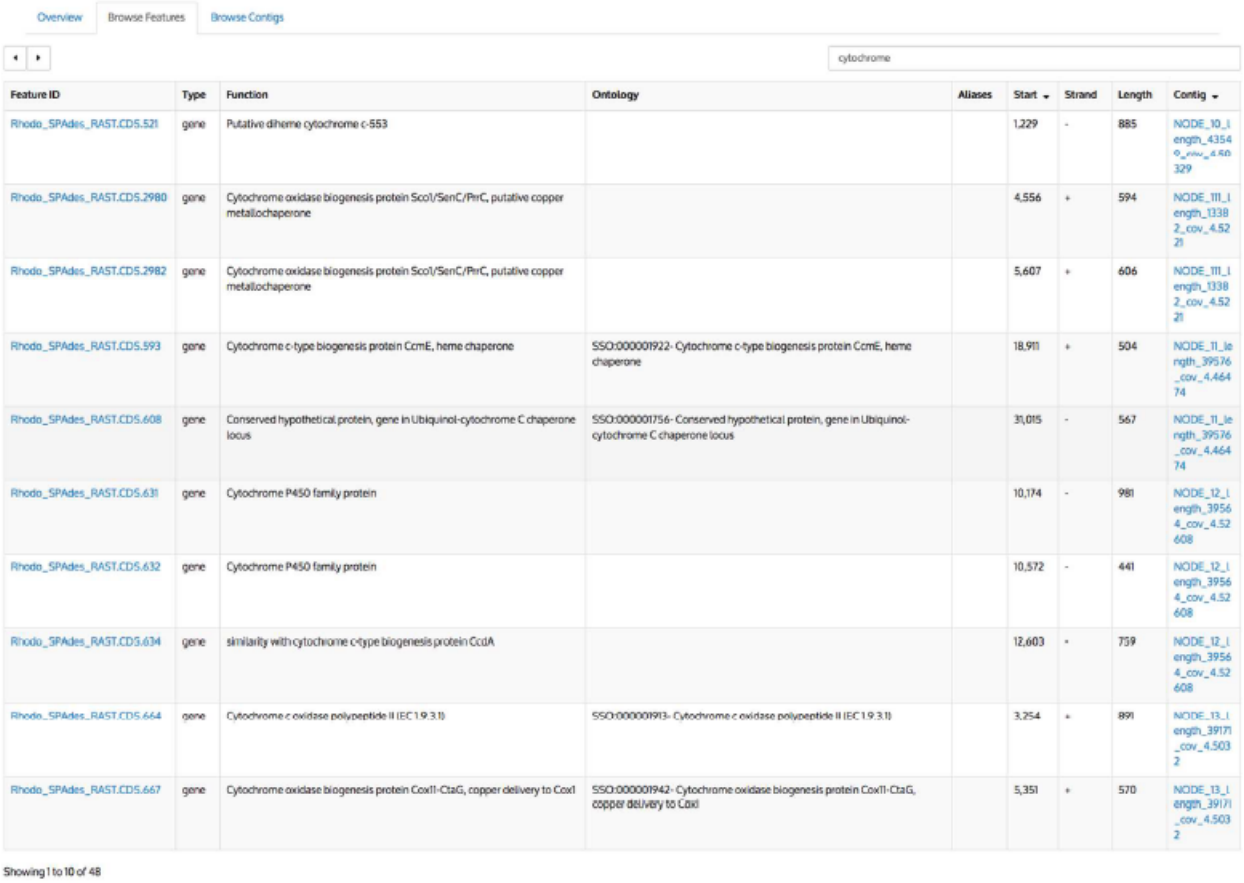

Figure 15. *Exploring the features annotated by RAST to understand differences between annotations. In this example, a search was performed for features annotated with "cytochrome" and the results were sorted first by contig and then by starting position.*

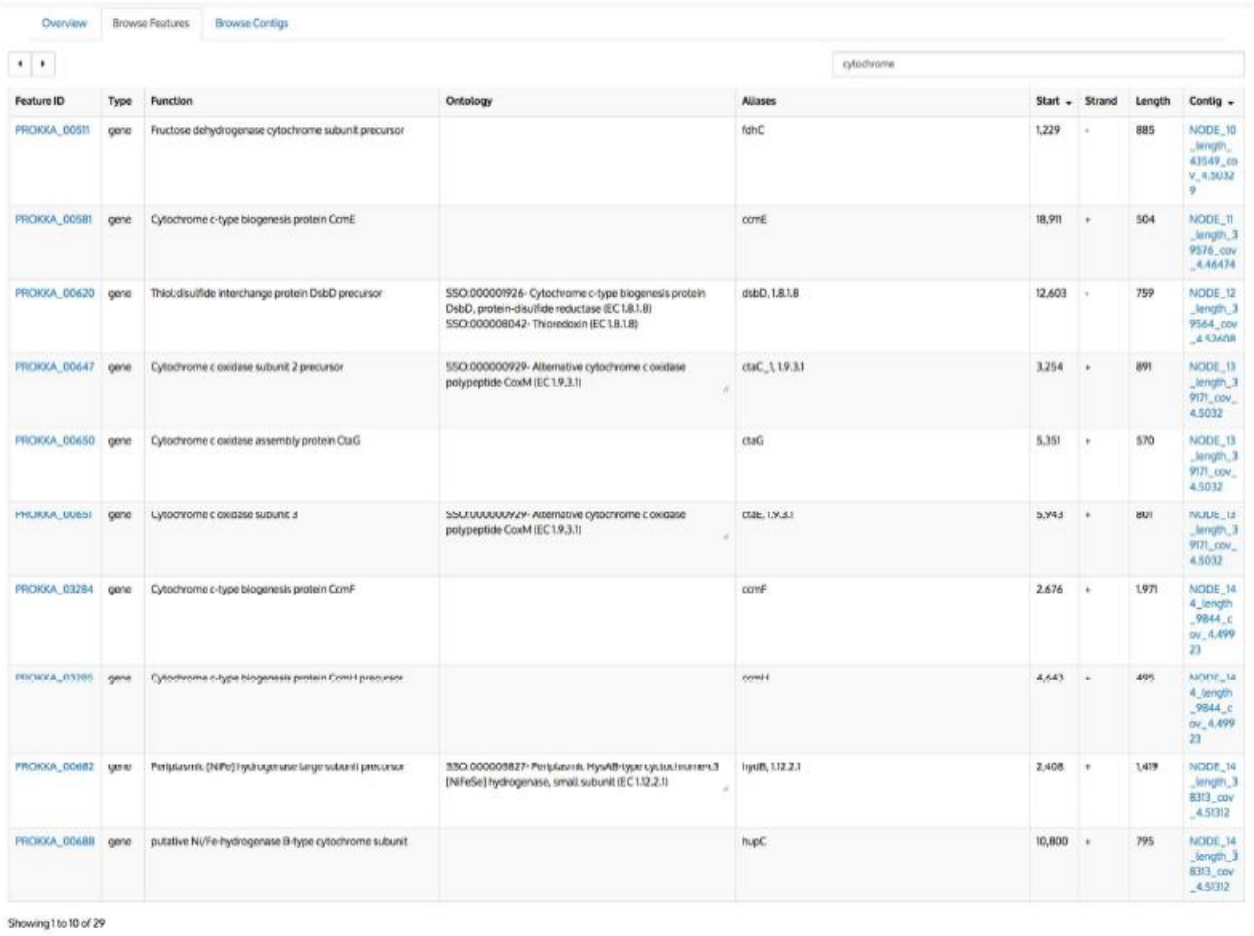

Figure 16. *Exploring the features annotated by Prokka to understand differences between annotations. In this example, a search was performed for features annotated with "cytochrome" and the results were sorted first by contig and then by starting position.*

Downloading Data from KBase to a Desktop Computer

Most data objects can be downloaded from KBase to a personal computer in common formats.

Materials

An internet-accessible computer with a recent version Chrome, Firefox, or Safari

A KBase account. KBase is free to sign up for and use. All functionality detailed in this protocol requires being logged in with a KBase account: http://kbase.us/sign-up-for-a-kbase-account/

1. Download the SPAdes assembly to a desktop computer.
    a. Locate the "Rhodo_SPAdes_assembly" assembly object in the Data Panel
    b. Hover the mouse over the assembly object and click the ". . ." to see several options for managing the data object.

c. Locate and click the icon labeled “Export / Download data” (see Figure 17).
d. A dialog labeled “Export As:” will appear below with options for formatting the data to be downloaded. Assembly objects can be downloaded as a FASTA file or JSON object. FASTA is a standard file type for assembly data, whereas the JSON format is specific to KBase and would likely only be useful for software developers.
e. Click the “FASTA” button to begin downloading the assembly. The FASTA file will be saved to the default download folder.

2. Download the RAST annotated genome to a desktop computer.

a. Locate the “Rhodo_SPAdes_RAST” Genome object in the Data Panel.
b. Hover the mouse over the Genome object and click the “. . .” to see several options for managing the data object.
c. Locate and click the icon labeled “Export / Download data” (see Figure 17).
d. A dialog labeled “Export As:” will appear below with options for formatting the data to be downloaded. Genome objects can be downloaded as a GenBank file or JSON object. GenBank is a standard file type for genomes, whereas the JSON format is specific to KBase and would likely only be useful for software developers.
e. Click the “GenBank” button to begin downloading the Genome. The GenBank (.gbk) file will be saved to the default download folder (Figure 18).

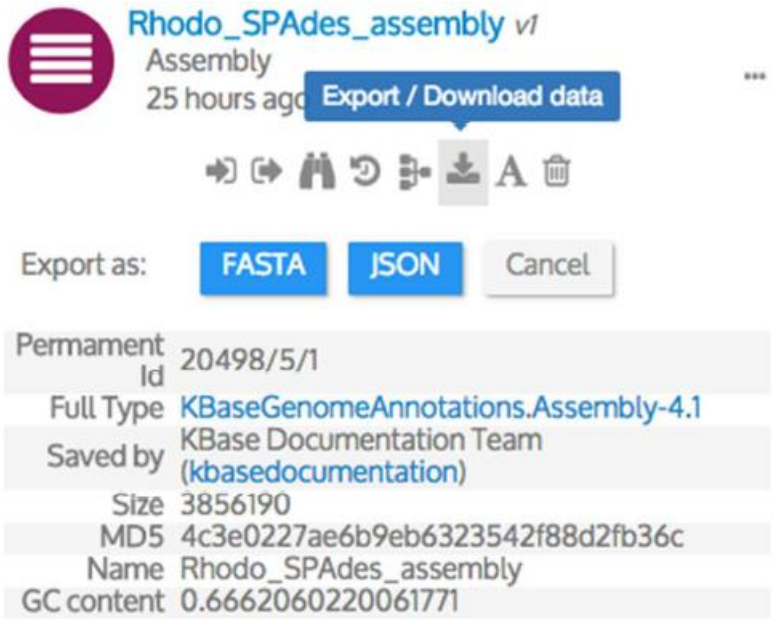

Figure 17. *Downloading the “Rhodo_SPAdes_assembly” assembly as a FASTA file.*

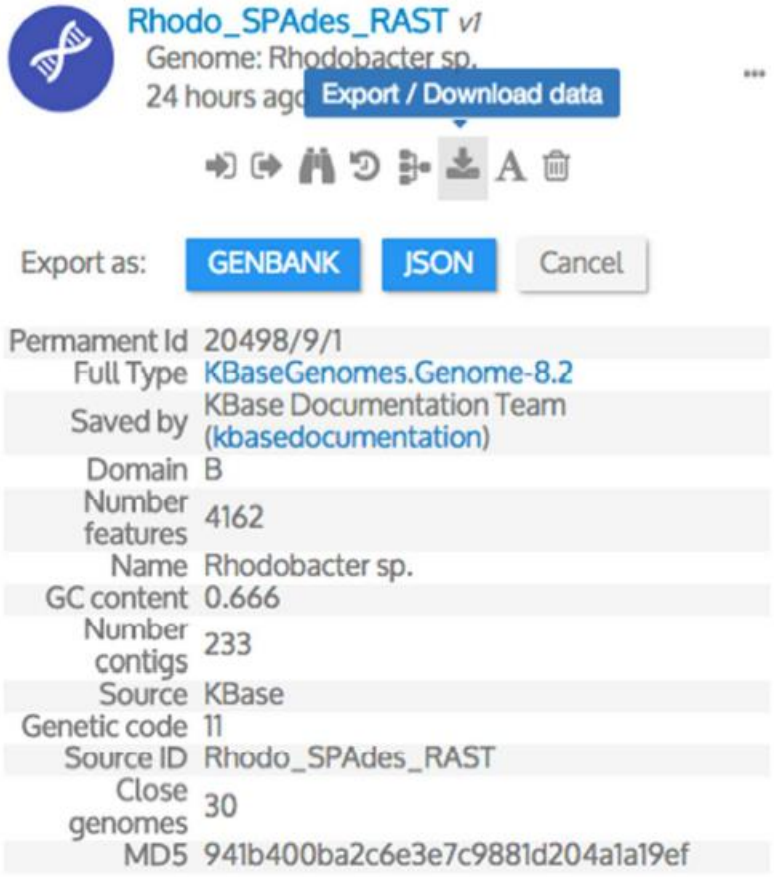

Figure 18. *Downloading the "Rhodo_SPAdes_RAST" genome as a GenBank file.*

Commentary

Background Information

KBase offers a streamlined user interface that integrates analysis tools and datasets to enable assembly, annotation, construction, and comparison of genomes. This protocol covers an essential and popular workflow in KBase that starts with DNA sequence reads, goes through pipelines for assembly and annotation, and generates a genome to be used in downstream analysis within KBase, or downloaded and analyzed in other software environments. Within KBase, these genomes can be used in apps for comparative genomics, phylogenetic analysis, and metabolic modeling.

Science Performed in KBase

An increasing number of researchers are using KBase to address research questions in systems biology. A list of publications that cite KBase in their methods can be found at

http://kbase.us/publications/. KBase users have applied the system to address a range of scientific problems, including comparative genomics of plants, prediction of microbiome interactions, and deep metabolic modeling of environmental and engineered microbes. The Narratives they have chosen to share publicly (see *http://kbase.us/narrative-library* for some examples) can be viewed, copied, and re-run, possibly with different parameters or new datasets.

Using KBase to Teach Genome Assembly and Annotation

The rapidly changing technological environment for biology has made it challenging to provide comprehensive training for graduate researchers. New scientists are faced with expectations to understand and apply the most up-to-date and relevant analytical pipelines and to create biologically relevant models from these data. KBase creates a platform for understanding, analyzing, and interpreting genomic data using the latest tools in a way that is transparent and reproducible for students and professors. The need to train these scientists in systems biology, while facilitating transparency and open collaborations between diverse scientific fields, led to the development of an intensive one-day workshop on assembly and annotation in KBase at Washington State University. The purpose was to communicate content knowledge of systems biology in general and genome assembly and annotation in particular and to increase confidence in working in these domains for a moderately sized group with diverse scientific backgrounds. Viability of this activity was demonstrated by pre- and post-workshop assessments soliciting both Likert scale (1- Strongly Disagree; 5-Strongly Agree) responses regarding attitudes and confidence and written responses to questions probing content knowledge regarding assembly and annotation. The Likert item data was analyzed using paired sample t-tests to assess significant differences between the pre- and post-test responses. For the 15 participants that completed the pre- and post-assessment, there were indicators that participating in the workshop about using KBase and running this protocol made students more confident in their capability to make informed decisions for assembly and annotation. When asked if they were able to make an informed decision about selecting an optimal quality *de novo* assembly, these students indicated that they were unable to do so confidently before the workshop (mean 1.4) while after the workshop they indicated that they were more able to make an informed decision (mean 3.7; p-value ≤ 0.05). A similar question was posed regarding the participants' capability to confidently explore and discover information about an organism based on functional annotations of its genome. Before the workshop student attitudes were neutral or negative (mean 2.87), while after the workshop they were more confident in their ability to meaningfully explore genome annotations (mean 4.14, p-value ≤ 0.05). The authors would like to compile more data to aid the development of curriculum for systems biology education using the KBase platform and are open to conducting these workshops at more universities.

User Feedback

KBase is a community-driven project, which makes getting feedback and priorities from users very important. Feedback from all types of users helps the team improve the user interface and functionality of the system, improve documentation and training, and better understand the priorities of the research community. All members of the community are encouraged to share their questions, comments and suggestions via the open Help Board (*https://kbase.us/contact-us/*).

Community Involvement

KBase was designed to be an extensible community resource. By allowing apps to be built in a standardized way, the KBase Software Development Kit (SDK) enables third-party developers to wrap new or existing open source analysis tools as KBase apps, and make them available in the Narrative Interface. Another way to contribute to KBase is to share Narratives that represent computational experiments (similar to interactive research papers), tutorials on how to perform specific workflows in KBase, or even "negative results" Narratives that document attempted experiments within the system.

Code Availability

KBase software, available at *https://github.com/kbase*, is open source and freely distributed under the MIT License.

Time Considerations

For a prokaryotic genome around 3Mbp long, running this whole protocol generally takes less than half an hour. *De novo* assembly is the longest step, with each assembler taking between 5 and 10 minutes to complete. Because these assembly and annotation tools are already installed and configured within KBase and utilize advanced computing resources accessible through a web interface, these tools typically run much faster than users would expect using a personal computer.

Acknowledgements

This work is supported as part of the Genomic Sciences Program DOE Systems Biology Knowledgebase (KBase) funded by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research under Award Numbers DE-AC02-05CH11231 (to Lawrence Berkeley National Laboratory), DE-AC02-06CH11357 (to Argonne National Laboratory), DE-AC05-00OR22725 (to Oak Ridge National Laboratory), and DE-AC02-98CH10886 (to Brookhaven National Laboratory).

References

Arkin, A. P., Stevens, R. L., Cottingham, R. W., Maslov, S., Henry, C. S., Dehal, P., … Yoo, S. (2016). The DOE Systems Biology Knowledgebase (KBase). *bioRxiv*. https://doi.org/10.1101/096354

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., … Pevzner, P. A. (2012). SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *Journal of Computational Biology*, *19*(5), 455–477. https://doi.org/10.1089/cmb.2012.0021

Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, *29*(8), 1072–1075. https://doi.org/10.1093/bioinformatics/btt086

Overbeek, R., Olson, R., Pusch, G. D., Olsen, G. J., Davis, J. J., Disz, T., … Stevens, R. (2014). The SEED and the Rapid Annotation of microbial genomes using Subsystems Technology (RAST). *Nucleic Acids Research*, *42*(D1), D206–D214. https://doi.org/10.1093/nar/gkt1226

Scheuring, S., Nevo, R., Liu, L.-N., Mangenot, S., Charuvi, D., Boudier, T., … Reich, Z. (2014). The architecture of Rhodobacter sphaeroides chromatophores. *Biochimica et Biophysica Acta (BBA) - Bioenergetics*, *1837*(8), 1263–1270. https://doi.org/10.1016/j.bbabio.2014.03.011

Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, *30*(14), 2068–2069. https://doi.org/10.1093/bioinformatics/btu153

Zerbino, D. R., & Birney, E. (2008). Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, *18*(5), 821–829. https://doi.org/10.1101/gr.074492.107

Internet Resources

FastQC Resource - *https://www.bioinformatics.babraham.ac.uk/projects/fastqc/*

KBase homepage - *http://kbase.us*