



Project No. **283496**

transPLANT

Trans-national Infrastructure for Plant Genomic Science

Instrument: **Combination of Collaborative Project and Coordination and Support Action**

Thematic Priority: FP7-INFRASTRUCTURES-2011-2

D12.2

A web-based interface for a decision support system that evaluates and recommends on genome sequencing

Due date of deliverable: 31 August 2014

Actual submission date: 28 October 2014

Start date of project: 1.9.2011

Duration: 48 months

Organisation name of lead contractor for this deliverable: The Genome Analysis Centre

Project co-funded by the European Commission within the Seventh Framework Programme (2011-2014)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

1 | Introduction

1.1 Background

Assembling plant genomes is difficult. No current technology produces all the required sequencing data, and no current algorithm solves the problem completely with the data at hand. While generating a good quality draft of a mammalian genome (i.e. a human genome) is within the reach of current technologies and approaches, the same does not hold true for plant genomes which can be many times larger, consist of over 80% repeat sequence and may exhibit high ploidy levels. Plant researchers attempt to tackle these challenges using combinations of technologies and elaborate processing pipelines, with varying results. To date, there is not a single answer on how best to design and execute plant genome sequencing and assembly projects.

There is currently no complete solution to the whole genome assembly problem, even after large scale evaluations such as the Assemblathons (Earl, Bradnam, et al. 2011 Bradnam, Fass, et al. 2013) and GAGE (Salzberg, Phillippy, et al. 2012). Several short-read assemblers (Simpson, Wong, et al. 2009 Luo, Liu, et al. 2012 Gnerre, MacCallum, et al. 2011) are in use by the plant community to produce draft-quality references, with different degrees of success. It is typical of plant genome assembly projects to tweak tools, create new ones, and combine them in non-standard ways. Results are sometimes quite satisfactory and useful, but the whole process is time-consuming, error prone and demands a high level of understanding.

Genome diversity is one of the main explanations for the current status of the plant assembly problem. Plant genomes can range from small and compact like that of *A. thaliana* to large, polyploid and repetitive, such as the genome of the hexaploid *T. aestivum*, and everything in between. While there is a key set of challenges that remain more-or-less constant for most plant genomes, genome organisation and characteristics produce different challenges for different plants.

The planning stage of a plant genome sequencing and assembly project can be confusing because of the variety of genome compositions, strategies and tools available. Experienced researchers often favour approaches and tools with which they are familiar, even when using new data types. The best approach, in the current scenario where only particular cases of the assembly problem have been solved, would be to check for similar cases and research how different strategies

have performed. However plant genomes are rarely the focus of assembly tools, and as such few test cases may be available.

We have created a repository of examples, with a complete description of how the data has been processed, and a set of guidelines for common best practices. This constitutes a key milestone on the road to simplifying assembly choices for plant genomes. The repository provides information on how to analyse and assemble the data, and also explores data generation and quality, including whether examples will be valid for any given genome.

1.2 The Assembly KB

As sequencing technologies become cheaper and more widely available, more projects try to sequence and assemble different plant species. The assembly Knowledge Base (KB), comprising an open access repository of examples and proposing a unified view of genome assembly across the community, will reduce the duplication of effort and increase the success rate of plant genome projects, providing a place to exchange results and assess assemblies according to different objectives.

Most plant assembly projects produce draft references which are fragmented, with collapsed repeats, chimeric sequences, and are an incomplete representation of the genome. The different types of errors or problems in the assembly are usually trade-offs between each other, imposing further complexity to the problem. A particular genome assembly might be useful for one type of analysis but less appropriate for another, depending on how the particular weaknesses and errors affect each analysis.

Given the large number of assembly projects and the fast-moving pace of the technologies and methods, allowing user contributions is the only way to remain current and ensure quality and credibility of the results. It also allows the whole community to benefit from the latest techniques, and will encourage people to share their methods, as a highly successful method will increase its relevance once it is included in the benchmarks.

2 | Using the Assembly KB

The Assembly KB ideally represents a single go-to resource to learn about current techniques on plant genome assembly and their results. While this is a very ambitious goal, we believe the current implementation is already a useful start, and we will continue to work towards this aim.

Due to the diversity of the data and the results, the main driver of the assembly KB is Quality Control (QC). QC data is presented for all of the libraries (i.e. GC bias, Kmer Spectra, Quality Score Distributions) and datasets (i.e. content comparison among different libraries), and instructions are provided to generate this same QC data for the user's own datasets. This will allow the researchers to check the validity of the examples for their project. Assembly QC, while still a very open research topic, provides comparative metrics, including contiguity and content based metrics, and measures of internal coherence such as mapability of the dataset's reads to the assemblies. The ability of the Assembly KB to generate rankings based on different metrics will be an aid towards choosing a strategy that provides the desired results.

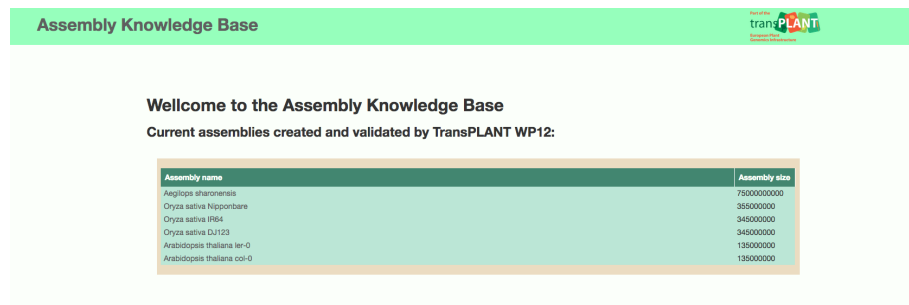


Figure 2.1: Welcome page with list of species contained in the Assembly KB.

Figure 2.2 shows a proposed workflow to use the Assembly KB during a project. Comparison with the example datasets can be used at every step of the project to validate results and consider next steps, as shown in this section.

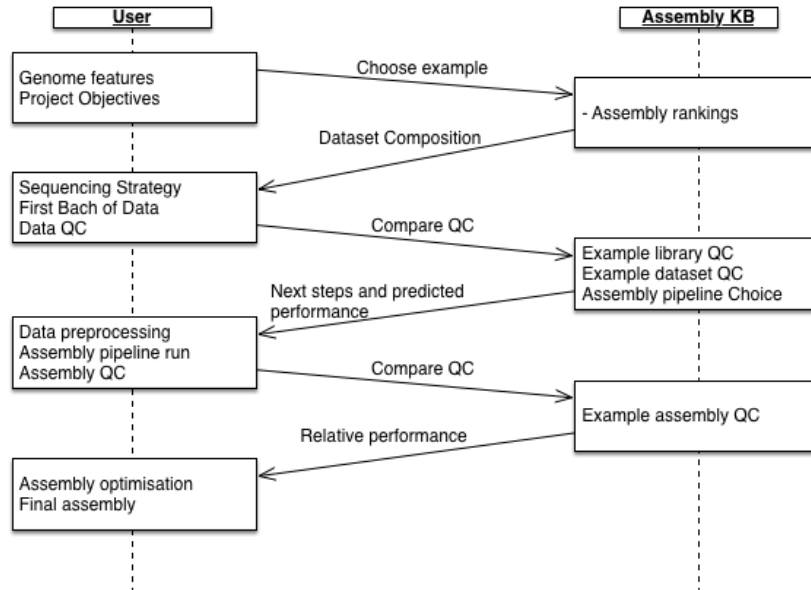


Figure 2.2: A proposed workflow when using the Assembly KB to aid in plant genome assembly.

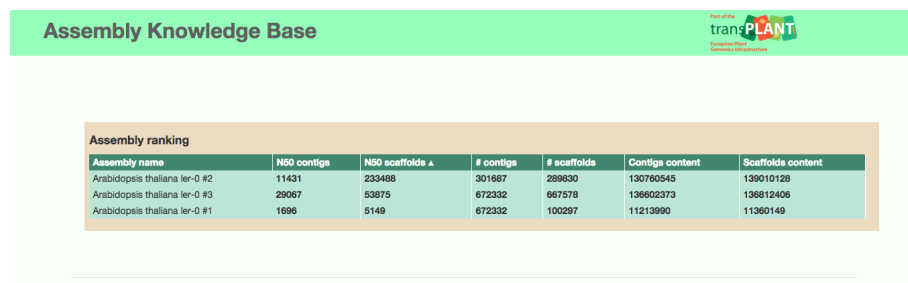
2.1 The planning stage: setting targets and picking examples

To know which type of output is best for a particular project, the questions to be answered using the genome assembly are the most important source of information: Projects looking for SNPs will need good base-by-base accuracy and enough contiguity to map reads from different samples onto the reference; Projects looking for comparative structural information will benefit from greater contiguity; Projects looking at genes might benefit from a good assembly of the usually less repetitive gene-space, whilst projects looking at certain features such as resistance genes which are usually interspersed with repetitions will need to at least untangle some repeats. By providing comparable QC among different strategies and datasets for the same or closely related species, the Assembly KB allows the researcher to check whether a certain approach is likely to produce a result that is usable for a given objective.

The Assembly KB can help inform each step when planning a genome sequencing and assembly project. By looking at a similar genome in the KB, datasets produced with different technologies and the assemblies produced by different processing pipelines can be compared. This not only allows the user to

consider different approaches but provides a complete example to follow while processing their data, at least on a first-pass basis.

A user of the Assembly KB may begin the exploration from a number of entry points. Whether it is from comparing their available libraries to those of an example dataset, looking for a genome sharing characteristics with their target species, or searching for an assembly with certain QC values, the interface has been designed to be unified and completely browsable.



The screenshot shows the 'Assembly Knowledge Base' header with the transPLANT logo. Below it is a table titled 'Assembly ranking'.

Assembly name	N50 contigs	N50 scaffolds ▲	# contigs	# scaffolds	Contigs content	Scaffolds content
Arabidopsis thaliana ler-0 #2	11431	233468	301667	289630	130760545	139010128
Arabidopsis thaliana ler-0 #3	29067	53875	672332	667576	136602373	136812406
Arabidopsis thaliana ler-0 #1	1696	5149	672332	100297	11213990	11360149

Figure 2.3: A simple assembly ranking, sorted by scaffold N50.

2.2 Coverage, fragment and read length, and wet-lab protocols

A successful genome assembly project starts by selecting an appropriate sequencing strategy and producing good quality data. Unfortunately, while mammalian-type genome assembly has successful recipes like "the allpaths recipe" that generates data to be used with the ALLPATHS-LG algorithm, there is no equivalent recipe that covers the whole diversity of plant genomes.

Choosing the right coverage of the correct type of data needs some careful considerations on the wet-lab side: different protocols require different types of input material and some work better or worse according to specific sequence characteristics such as GC content. While the generation of Long Mate Pair (LMP) reads is an unavoidable requirement for most plant genomes, the protocols for these libraries have limited reproducibility. This emphasises the need for QC and comparison of results.

The availability of PCR-free Paired End (PE) sequencing protocols (including Illumina's own standard kit for PCR-free sequencing) makes it easy to generate this kind of data. The absence of PCR bias provides a more uniform distribution of coverage, which is relevant for any genome that contains extreme GC sections.

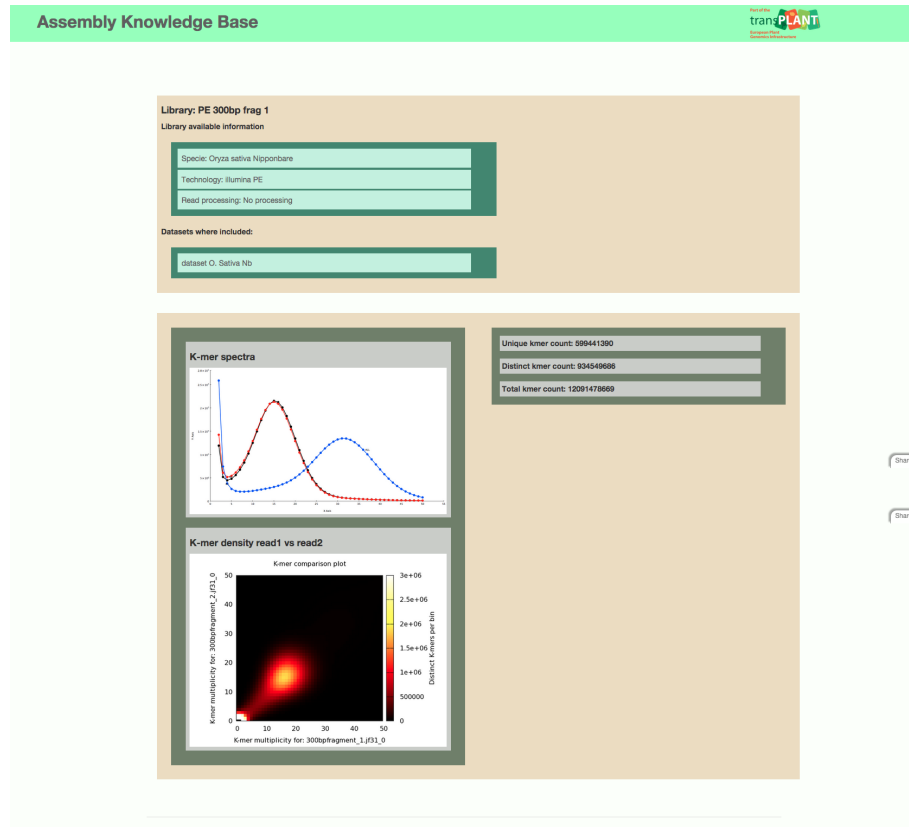


Figure 2.4: A library page showing QC results.

2.3 Generating data incrementally

Once an appropriate strategy for sequencing has been chosen, data generation can begin. While it is very common to generate all the data at the same time and then try to assemble it, generating the data incrementally and performing exhaustive QC and partial assemblies at each stage may be more efficient and provide guidance as to which data should be generated. We are exposing and encouraging this methodology via the assembly KB by generating datasets that incrementally add different libraries for each individual genome.

A good starting point to assess a novel genome is to generate around 50x of Illumina Paired End data, trying to get long fragment sizes in the range of 600bp-800bp and the longest read size that is economically reasonable for the project. Currently the longest Illumina reads are generated by the MiSeq (2x300bp) though the throughput is lower and cost per base higher than the HiSeq 2500 or 1 Terabase HiSeq..

By creating PE-only assemblies we can QC the libraries and the assembled

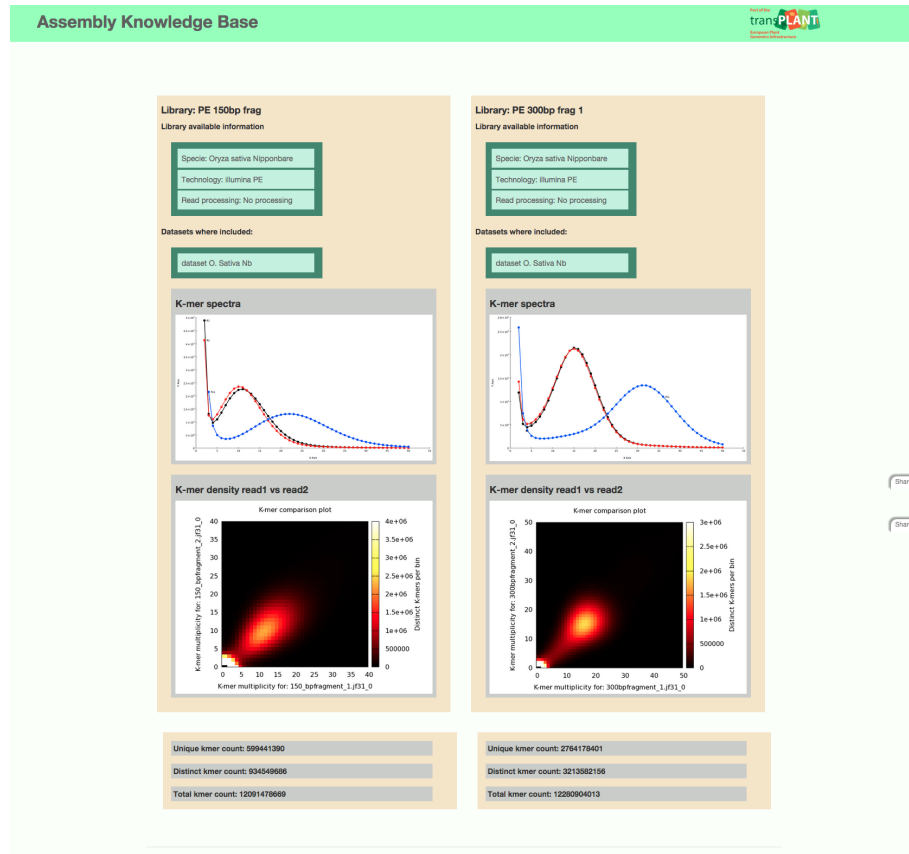


Figure 2.5: Side-by-side library comparison.

sequence, and make an informed decision about which step to take next. Then we can sequence more PE data, add incrementally-longer LMP data, add complementary data such as Pacbio or optical restriction maps, and then repeat the QC process to make the next choice.

2.4 Data QC, preparation, and comparison with the guideline data

Performing data QC is a challenge for Next Generation Sequencing, as while some technical metrics are universal, the characteristics of the genome being sequenced have a big impact on them. The Assembly KB will help here by providing data QC for example datasets, and allowing the user to evaluate how their data looks in comparison.

We are currently using KAT (<http://www.tgac.ac.uk/kat/>) kmer spectra

metrics as a starting point for library QC, because that enables cross-library coherence metrics. We are currently working on the integration of some of the FASTQC reports, and considering the addition of PreQC reports.

We provide the users with instructions on how to generate the QC reports comparable with those in the Assembly KB. A bundled software + script package will soon be available as a single download binary for linux and will allow the users to run all of the tools and create all of the reports in one step. We will then be able to offer the user the possibility of displaying their output side-by-side with the Assembly KB QC data, for an easier comparison.

As with the assembly quality metrics, the library quality metrics reflect some of the trade-offs in data generation. Even when a library has a better set of metrics than others, it won't necessarily result in better assemblies because the algorithms in the data processing software (data preparation software, assemblers and scaffolders) may not take advantage of these features. This is a key point for genomic assembly: the library quality needs to be evaluated in conjunction with a processing pipeline and its results.

2.5 Choosing a processing pipeline as a starting point, and comparing the results with the guidelines

Aided by the target metrics, the genome characteristics and the data QC reports, a user can choose a processing pipeline that is predicted to produce adequate results by looking at how different pipelines have performed on similar datasets. The Assembly KB has a complete description of how to reproduce the processing including precise versions and parameters for all of the software. An executable version of the pipeline will be available to download in the near future.

We plan to expand the control points to be able to follow each individual step of the pipeline and detect where any deviations are occurring. This helps when troubleshooting assembly processes, and the failure in a particular step is usually easier to link to either genome characteristics or data properties.

We would like to provide a comparison system to automatically check user submitted QC against the database and present the most representative datasets and suggest next steps to the users to guide them forward.

The Assembly KB is a good resource for example datasets and provides, a starting point for *de novo* plant genome assembly projects. However, there will remain a level of detailed parameter tuning will need to be performed by the bioinformatician as part of the assembly process to achieve optimal results.

2.6 Feeding back results: user submissions

Sequencing technologies, and the algorithms to process the data they generate, evolve rapidly. We have taken that into account when creating the Assembly

KB, rather than producing a one-off report that would rapidly become obsolete. We plan to keep the resource up to date allowing users to contribute their datasets and results, enabling a richer set of approaches and trials to be considered. To maintain consistency and quality, we will create a submission process for datasets and processing techniques that ensures reproducibility by reprocessing the data. This is a challenge both in terms of the human and computing time and resources required.

We are currently implementing a 3-tier structure to classify user assemblies. Users will submit their datasets by pointing to runs on the already existing read archives (EBI/NCBI/JGI), and uploading their QC results both for the reads and the assembled sequences, along with a detailed description of their processing pipeline.

Table 2.1 shows a detail of how user submitted assemblies will be divided. Class 3 assemblies will only require a reference to the input data, and the upload of data QC and assembly QC results. When a class 3 assembly is of sufficient interest for the Assembly KB, and provided that the assembled sequences have been uploaded, the QC for both the input data and the assembly will be run in the standard Assembly KB pipelines and uploaded replacing the user submission: this will constitute a class 2 assembly. An interesting class 2 assembly can become a class 1 assembly once the full assembly pipeline has been executed on the servers, checked, and corrected where needed; and its resulting assembly has replaced the user submitted assembly both for sequence availability and QC.

Class	Data QC	Pipeline	Assembly	Assembly QC
1	AKB	AKB checked and run	AKB	AKB
2	AKB	User description	User	AKB
3	User/AKB	User description	None/User	User

Table 2.1: Assembly classes being implemented to allow user submissions, according to the origin and level of detail available for the reported result.

Defining which assemblies are of sufficient interest will be initially the responsibility of the Assembly KB maintainers. We are considering the implementation of a user voting system to allow the community to choose assemblies on classes 2 and 3 for promotion to the next class. Depending on the demand this promotion scheme imposes on our resources, we will be able to choose a varying amount of the most interesting assemblies.

Assembly Knowledge Base



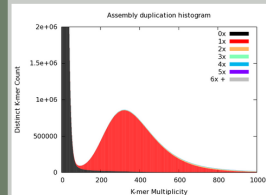
Assembly settings for assembled O. Sativa Nb 1

Species: *Oryza sativa* Nipponbare
 Dataset: dataset O. Sativa Nb
 Assembly run: Run1 O. Sativa Nb

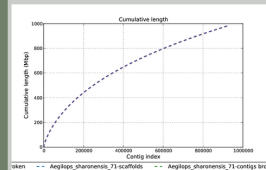
Dataset: dataset O. Sativa Nb

Lib name	Lib type	Pre-processing	Coverage	Read size	Fragment size
PE 150bp frag	Illumina PE	No processing	58	100	150
PE 300bp frag 1	Illumina PE	No processing	49	100	300
PE 300bp frag 2	Illumina PE	No processing	20	100	300
LMP 2kb jump 1	Illumina LMP	No processing	563	100	2000
LMP 2kb jump 2	Illumina LMP	No processing	12	100	2000
LMP 2kb jump 3	Illumina LMP	No processing	14	100	2000
LMP 5kb jump	Illumina LMP	No processing	483	100	5000

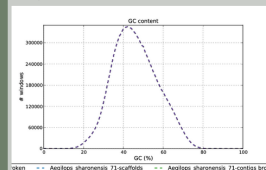
Contigs base content



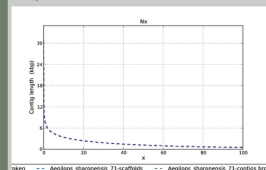
Length report



GC report



Nx report



N50 contigs: 881
 N50 scaffolds: 857
 # > 1k contigs: 48497
 # > 1k scaffolds: 49399
 Contigs content: 1577017421
 Scaffolds content: 1428260044
 Contigs 1k content: 67289555
 Scaffolds 1k content: 66328749
 # contigs: 7373247
 # scaffolds: 9625676

Share

Share

Share

Figure 2.6: Assembly page showing QC results.

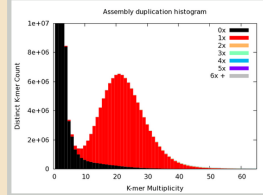
Assembly Knowledge Base



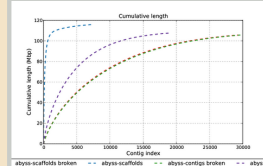
Assembly settings for assembled A. Thaliana col-0 1

Species: Arabidopsis thaliana col-0
 Dataset: dataset A, Thaliana col-0
 Assembly run: Run1 A, Thaliana col-0

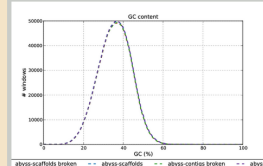
scaffolds base content



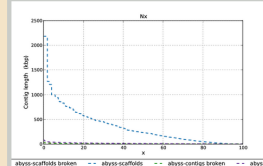
Length report



GC report



Nx report



NS0 contigs: 29067

NS0 scaffolds: 53875

> 1k contigs: -1

> 1k scaffolds: -1

Contigs content: 136602373

Scaffolds content: 136812406

Contigs 1k content: 104770536

Scaffolds 1k content: 105929227

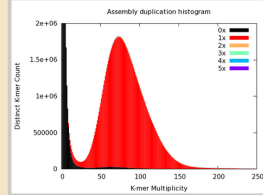
contigs: -1

scaffolds: -1

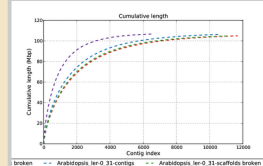
Assembly settings for assembled A. Thaliana ler-0 3

Species: Arabidopsis thaliana ler-0
 Dataset: dataset A, Thaliana ler-0
 Assembly run: Run3 A, Thaliana ler-0

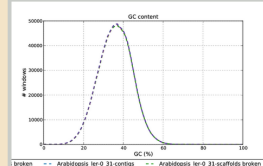
scaffolds base content



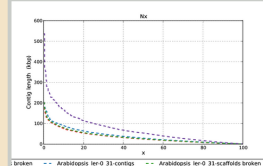
Length report



GC report



Nx report



NS0 contigs: 11431

NS0 scaffolds: 233496

> 1k contigs: 15032

> 1k scaffolds: 4108

Contigs content: 130760545

Scaffolds content: 139010128

Contigs 1k content: 301667

Scaffolds 1k content: 289630

contigs: -1

scaffolds: -1

Figure 2.7: Side-by-side assembly comparison.

3 | Assembly Knowledge Base Behind the Scenes

3.1 Implementation details

The current version of the Assembly KB is based on web2py, using the native SQLite3 persistence model for the data. The assembly sequence files are stored locally in an Apache web server that also acts as a proxy for the web2py dynamic site. While this is not the most scalable solution, this is unlikely to be a problem in the mid-term future. In case of an architectural change the data will be available easily for migration to any new platform.

We based our data schema on a flexible definition of datasets, processing and QC described in figure 3.1. This schema is flexible enough to allow us to introduce QC at different points in the assembly process, and describe the effects of any tool. We also defined the QC values in such a way that allows the automated creation of rankings and comparisons.

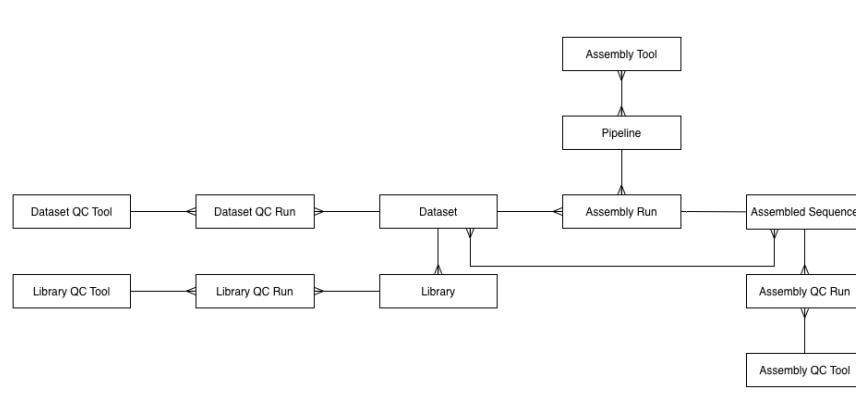


Figure 3.1: A simplified version of the database schema showing the data, processing and QC classes.

3.2 Reproducing the processing pipelines

To allow reproducibility on the assemblies, we have decided to keep complete records of all the software run and parameter settings. This includes not only assembly software, but also the read-preprocessing and assembly finishing step. While each run of assembly software can produce slightly different results, the complete description should allow a good level of reproducibility. We are considering adding intermediate QC metrics on the datasets when they are prepared for assembly, such as QC on the trimmed or error corrected reads.

At TGAC we have the advantage of a large computing facility, including multi-terabyte single-memory-image computers, which allow us to run and reproduce user generated assemblies. Where proprietary software is involved to which we do not have access we may leave the assemblies as class 2 or 3. However, if there is sufficient demand we will attempt to work with companies to include the software in a benchmarking and guideline knowledge base.

3.3 Rankings and their use as best-current-method assessments

An important feature of the Assembly KB is the ability to create assembly rankings using different quality metrics for the same species. We have defined this in a generic way, which allows us to either use the metrics independently or create rankings with combinations of metrics. We believe the use of rankings, especially for class 1 assemblies where the processing pipelines are checked and run in an unbiased way, will enable a real and healthy competition among groups creating different software and pipelines to assemble plant genomes. These rankings can be restricted to a specific dataset, thus making it just a comparison of assembly methods, but we think the best situation is indeed when different datasets and processing pipelines can be evaluated and compared in the same species.

We plan, in the future, to extend our ranking system to work combining different species rankings, and individual pipeline steps, allowing the creation of reports such as "the ranking for full assembly pipelines on grass genomes" or "the ranking for scaffolders on polyploid genomes". This will also allow us to attempt an individual evaluation of how each characteristic of a genome affects the different methods, trying to differentiate for instance the effects of ploidy from the associated growth in genome size.

3.4 Current status of the assembly KB

The assembly KB is currently available on <http://assemblykb.tgac.ac.uk>. All the screenshots in this document correspond to already implemented features, and the site is accessible for users to enter.

So far only class 1 assemblies are stored in the knowledge base, created to bootstrap the database with usable information. Table 3.1 shows a list of the currently available species with a summary of available data and assemblies. We will continue to generate data for this first batch of initial assemblies, and we will open the user submissions for class 3 assemblies in the near future.

Species	PE		LMP			Assemblies
	Libs	Cov	Libs	Sizes	Libs.	
<i>A. thaliana ler-0</i>	3	75x			3	3
<i>A. thaliana col-0</i>	2	20x	1	3kbp	5x	5
<i>O. Sativa Nipponbare</i>	3	180x	2	2-5kbp	900x	3
<i>O. Sativa IR64</i>	3	450x	2	2kbp	600x	3
<i>O. Sativa DJ123</i>	3	250x	3	2-5kbp	900x	3
<i>A. Sharonensis</i>	2	400x	2	3-8kb	200x	3

Table 3.1: Available species, data and assemblies as of October 2014.

We are in the process of submitting the read files for data which has not yet been made public (mainly datasets generated for this project) to the read archives, and the assemblies are available directly from the Assembly KB.

Instructions for how to generate QC metrics to compare user datasets to the Assembly KB datasets are provided on the website. We are building a script to run all of the required QC at once, thus making it easier for the users in the future.

3.5 Maintainability and future directions

We have designed the Assembly KB to be simple and require little maintenance, but the task of data curation and administration will require some time and dedication. It is in TGAC's best interest to maintain the database both as a tool for assembly projects and as a standard benchmark for new techniques, given its role as an early technology adopter for sequencing. We are considering applying for additional funding to further develop this resource in the future.

We are currently working on the following points, which we plan to finish before the end of the TransPLANT project:

- First version of the website: we are still finishing some details, such as the layout of metrics and detail pages for sequence sets.
- Extra metrics: we are adding new metrics both for datasets and assemblies.
- Improved rankings: we are working on the ranking system to create composed metrics and composite rankings.
- QC scripts: We need to create a package to expose these internal scripts and make it simple for users to run.

- User submissions: we need to create the user submission section of the website.

Longer term goals which may depend on us securing further funding include:

- Rampart pipelines: we are already using Rampart (<http://www.tgac.ac.uk/rampart/>) internally to run most of our assemblies, so we plan to integrate the pipeline system into the Assembly KB. This will allow users to download pipeline definitions and will give us a way to store the pipelines "exactly as run".
- User project tracking: we are planning to enable users to create a project entry for their own assembly projects as they are executing them. This feature will allow them to save step-by-step QC of their project and compare it to the guideline data, and receive suggestions of what the next steps should be.
- User interaction: once users have project entries, it will make sense to allow users to make their projects available for the community to comment, ask for help, etc.
- Full assembly performance prediction: ultimately, a complete analysis of the datasets on the Assembly KB should allow us to create accurate automated assembly predictions for a dataset.
- Inclusion of metrics based on performance of downstream analyses: while this will add yet another layer of complexity, it would be ideal for researchers looking for typical downstream analysis such as SNP calling. Because of the difficulty of reproducing downstream analysis results with different tool combinations we will need to choose example pipelines and report specific results.
- Extension into transcriptomic assembly: once the genomic assembly features are well built and robust, an extension to transcriptomic assembly might be achievable. This is an even more complex problem and will require considerable work.

Bibliography

- Bradnam, Keith R, Joseph N Fass, et al. (2013). “Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species.” In: *GigaScience* 2.1, p. 10.
- Earl, D, K Bradnam, et al. (2011). “Assemblathon 1: A competitive assessment of de novo short read assembly methods”. In: *Genome Research* 21.12, pp. 2224–2241.
- Gnerre, Sante, Iain MacCallum, et al. (2011). “High-quality draft assemblies of mammalian genomes from massively parallel sequence data.” In: *Proceedings of the National Academy of Sciences of the United States of America* 108.4, pp. 1513–1518.
- Luo, Ruibang, Binghang Liu, et al. (2012). “SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler.” In: *GigaScience* 1.1, p. 18.
- Salzberg, S L, A M Phillippy, et al. (2012). “GAGE: A critical evaluation of genome assemblies and assembly algorithms”. In: *Genome Research* 22.3, pp. 557–567.
- Simpson, J T, K Wong, et al. (2009). “ABYSS: A parallel assembler for short read sequence data”. In: *Genome Research* 19.6, pp. 1117–1123.