











REVIEW

Significantly improving the quality of genome assemblies through curation

Kerstin Howe ^{*}, William Chow , Joanna Collins , Sarah Pelan ,
Damon-Lee Pointon , Ying Sims , James Torrance , Alan Tracey  and
Jonathan Wood 

Tree of Life, Wellcome Sanger Institute, Cambridge CB10 1SA, UK

^{*}Correspondence address. Kerstin Howe, Tree of Life, Wellcome Sanger Institute, Cambridge CB10 1SA, UK.

E-mail: kerstin@sanger.ac.uk  <http://orcid.org/0000-0003-2237-513X>

Abstract

Genome sequence assemblies provide the basis for our understanding of biology. Generating error-free assemblies is therefore the ultimate, but sadly still unachieved goal of a multitude of research projects. Despite the ever-advancing improvements in data generation, assembly algorithms and pipelines, no automated approach has so far reliably generated near error-free genome assemblies for eukaryotes. Whilst working towards improved datasets and fully automated pipelines, assembly evaluation and curation is actively used to bridge this shortcoming and significantly reduce the number of assembly errors. In addition to this increase in product value, the insights gained from assembly curation are fed back into the automated assembly strategy and contribute to notable improvements in genome assembly quality. We describe our tried and tested approach for assembly curation using gEVAL, the genome evaluation browser. We outline the procedures applied to genome curation using gEVAL and also our recommendations for assembly curation in a gEVAL-independent context to facilitate the uptake of genome curation in the wider community.

Keywords: genome; assembly; curation; gEVAL

Assembly Curation Adds Significant Value

Despite the advances in sequencing and mapping technologies and the ever-increasing number of sophisticated algorithms and pipelines available, generating error-free eukaryotic genome assemblies in a purely automated fashion is currently not possible [1, 2]. Assembly software designed to generate continuous sequence from raw reads is confused by heterozygous or repeat-rich regions, introducing erroneous duplications, collapses, and misjoins. The same issues recur in subsequent scaffolding processes that aim to turn primary contigs into representations of chromosomal units. The fact that these tools are commonly applied in series rather than in parallel results in the passing of mistakes made from one process on to the next. As a result, even so-called high-quality or “platinum” assemblies can suffer from hundreds to thousands of duplications, collapses, misjoins

and missed joins. Because assemblies are often judged simply by their continuity rather than by their completeness and (structural) correctness, these errors go unnoticed. This affects research in many ways, making whole regions of the genome impossible to access or misleading researchers who misinterpret assembly artefacts as biological findings (C. Lee, unpublished data).

One way to address these shortcomings is in-depth analysis of discordances between the assembly that has been generated and the different data types available for the sequenced individual or species and subsequent resolution of these discordances. This can be performed at the sequence and the structural level. Many automated tools are available that assess sequence quality through read alignment, k-mer counting, gene finding, and other methods [3–6]. For structural quality assessment, several

Received: 12 August 2020; **Revised:** 17 November 2020; **Accepted:** 30 November 2020

© The Author(s) 2021. Published by Oxford University Press GigaScience. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

individual tools can be used, but these tend to analyse a single data type at a time rather than combining insights from analysis of several in parallel [7, 8].

We created gEVAL, the genome evaluation browser, to enable a user to visualize and evaluate discordances between an assembly and multiple sets of accompanying data at the same time [9]. gEVAL enables the identification of errors and simultaneously suggests ways to resolve them. Combined with manual assessment of the generated data by experienced curators and a pipeline that enables the curators to record changes and recreate the improved assembly accordingly, gEVAL provides a critical addition to strategies striving to produce assemblies of the highest possible quality.

Herein we outline the strategic design, achievements, and limitations of the gEVAL approach to assembly curation. gEVAL is tied into our local infrastructure and as such sadly not portable, yet fully publicly accessible at geval.org.uk. We therefore also provide detailed recommendations on how to create similar analyses that do not use gEVAL to promote the core, proven design concepts in gEVAL. This is especially timely in the context of emerging projects that aim to assemble the genomes of very large numbers of species to highest quality possible, including the Vertebrate Genomes Project (VGP), the Darwin Tree of Life Project (DTOL, darwintreeoflife.org), and the overarching Earth Biogenome Project [1, 10].

Checking for Assembly Coherence, Coverage, and Contamination

We recommend that every genome assembly be checked for coherence. This includes making sure that only data that belong to the relevant species are used for assembly in the first place. This is best checked before starting the assembly process by aligning all raw datasets with, e.g., mash [11] and checking that the data are in fact combinable (i.e., that they are likely to derive from the same underlying distribution of sequence). A major source of remaining technical error in assemblies is the retention of duplicated regions that result from failure to recognize that two sequences are in fact allelic. These false duplications have wide-ranging negative consequences for subsequent research, e.g., causing prediction of erroneous gene duplications [1]. False duplications are caused by either incorrect resolution of assembly graphs or failures in detection of haplotypic variation. They can be detected using simple read coverage plots or more sophisticated *k*-mer analyses (e.g., using the K-mer Analysis Toolkit [KAT] [4], the K-mer Counter [KMC] [12], or Merqury [6]). *K*-mer approaches also support the estimation of the completeness of the assembly (i.e., whether the assembly contains all the relevant *k*-mers present in the reads) and the ploidy of the genome [13]. False duplications can be removed, ideally after generation of the contigs, with tools that recognize partial and complete allele overlap, such as *purge_dups* [14]. In addition to duplications, assembly quality is also negatively affected by erroneous sequence collapses, mostly located in repetitive regions. Collapses are relatively easy to detect on the basis of increased read coverage, but harder to resolve because they require generation of new sequence. This can be performed through extraction of mapped reads and local reassembly under more stringent conditions, or with more sophisticated methods such as the Segmental Duplication Assembler (SDA) [15].

Assemblies are frequently polished after contig generation, using the bulk of data or particular high base accuracy data such as Illumina short reads, to correct remaining errors in the de-

rived consensus sequence. It is however possible to over-polish, such that rare repeat variants are replaced by the most abundant version, or where nuclear insertions of organellar genome fragments (nuclear mitochondrial transfers [NUMTs] and nuclear plastid transfers [NUPTs]) are polished to match the organelle sequence. For polishing, the target genome assembly therefore must include the organelle genomes. Organellar genomes are often missing from assemblies because assembly toolkits recognize and exclude them as repeat sequence or because they yield complex graphs that conflict with nuclear insertions. They can be assembled independently from the raw reads, e.g., using the mitoVGP pipeline [16]. Contigs/scaffolds that represent the organelle genomes should be identified and submitted as such to the International Nucleotide Sequence Database Collaboration (INSDC) archives.

A preliminary assembly of data from a target species can inadvertently include synthetic sequence from cloning or sequencing systems, contamination from species handled in the same laboratory or sequencing centre, or contamination from natural cobionts of the target (e.g., gut and skin microbiomes, unsuspected parasites). Decontamination serves to detect and mask or remove sequence not originating from the target species, and to separate organelle genomes from the primary assembly if not carried out previously. This includes identifying remaining vector and adapter contamination based on known sequence. Contaminating sequence can be detected with dedicated toolkits, such as BlobToolKit [17] or Anvi'o [18], or through individual sequence similarity searches using BLAST or Diamond against suitable databases (Table 1). Our in-house pipelines use automated detection of synthetic, laboratory, and natural contaminants, but include manual controls to preserve sequences that may be the product of horizontal gene transfer (described below).

Last, trailing Ns should be removed from all contigs and scaffolds.

Improving Structural Integrity

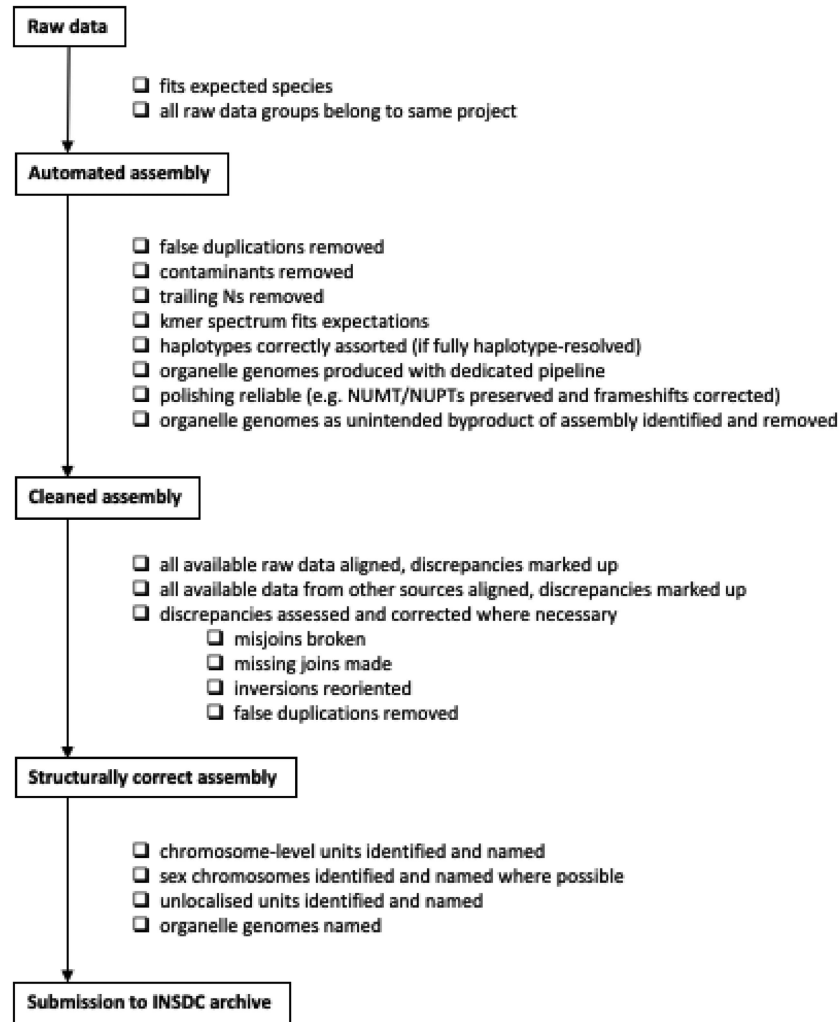
Because most assembly pipelines currently apply different scaffolding steps in series, errors in early steps can propagate through the process. To avoid compounding these errors, one could carry out a thorough curation process after every scaffolding step, but if many scaffolding steps are involved this will be very demanding on time and resources. Our experience has shown that structural integrity can be successfully improved after completion of a full, automated assembly process [1, 9].

The principle behind identification of assembly errors is simple: align all available (raw and other) data to the produced assembly, check for discordances, and then correct. Several tools that detect scaffolding issues with single data types are available, including *scaff10x* for 10X Chromium linked reads [28], *Access for Bionano maps* [7], and *HiGlass* [29], *pretext* [30], and *Juicebox* [8] for Hi-C data. ASSET evaluates multiple data types in parallel and is therefore an excellent tool to assess and visualize potential misassemblies [31]. Read coverage plots identify errors or problem regions through deviation from expected averages (indicating possibly problematic low-coverage regions, haploid regions, or regions of collapsed repeat) and sites where aligned reads are all clipped at the same site (suggesting that the assembly contains an erroneous join). Aligning the assembly against itself can be used to detect duplications.

Additional data not used in generating an assembly also provide critical information. Comparing the assembly to previ-

Table 1: Detecting contamination in assemblies, inspired by the processes carried out by GenBank's genome archive [19]

Contaminant	Software tools	Detection requirements	Database
Vector/adaptor sequence	Vecscreen [20]		UniVec [21]
Common contaminants	megaBLAST [22]	e-value $\leq 1e-4$, reporting matches $\geq 98\%$ sequence identity with match length 50–99 bp, $\geq 94\%$ with match length 100–199 bp, or $\geq 90\%$ with match length > 200 bp	Contamination in eukaryotes [23]
Organelle genomes	megaBLAST	e-value $\leq 1e-4$, sequence identity $\geq 90\%$, match length ≥ 500	RefSeq mitochondria [24] and plastid [25] assemblies
Other species	megaBLAST	e-value $\leq 1e-4$, match score ≥ 100 , sequence identity $\geq 98\%$; ignore regions also matching highly conserved rDNAs	Windowmasked [26] RefSeq genomes [27]


Figure 1: Recommended workflow for curation activities during assembly generation.

ous assemblies from the same species or to assemblies from closely related species can highlight areas of disagreement and thus areas that deserve closer attention during curation. Transcript evidence, as assembled cDNAs or long single-molecule reads, can be aligned to affirm joins across sequence gaps, identify local misassemblies, and detect false duplications. Protein sequences from the same or related species can serve the same purpose. Centromeres and telomeres can be identified in the assembly through sequence features [32, 33]. Long-range

structural data (such as karyotypes and fluorescence *in situ* hybridization mapping) and genetic mapping data (such as genetic map or radiation hybrid mapping data) can provide validation of the large-scale correctness of an assembly and, in particular, guide correct association and orientation of chromosomal arms with respect to telomeres and centromeres. Chromosome-wide patterns of repeat proportion and GC content can also be used to affirm completeness of chromosomal units.

Table 2: Examples of data types and analyses included in gEVAL and their ability to detect issues and errors

Data type	Software	Analysis type supported			
		Misjoins	Missed joins	Duplications	Collapses
Long reads	Minimap2 [43], winnowmap [44]	X	X	X	X
Bionano	Bionano Solve	X	X	X	X
10X linked reads	Break10x [28]	X			
cDNAs/gene sets	Blat [45], pblat [46]	X	X	X	
Self-alignments	Mummer [47]			X	
Other assemblies	Compara [40]	X	X		
Centromeres	Repeatmasker [40, 48], centromere db [32]	X	X		
Telomeres	Find_telomere [33] adapted to work with any sequence	X	X		
Genetic and other maps	EPCR [49], Blast [50]	X	X	X	

cDNA: complementary DNA.

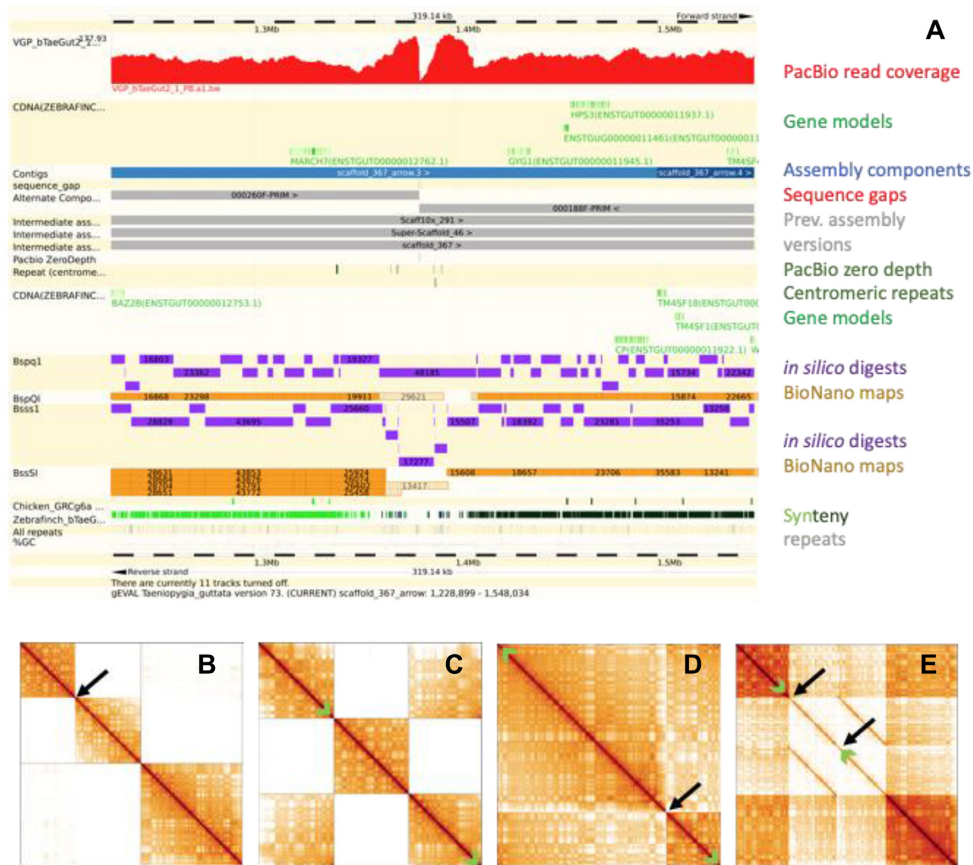


Figure 2: Examples of assembly error signatures in different data types. (A) Assembly issue identified in gEVAL in a bird genome (*Taeniopygia guttata*, VGP). Feature tracks (named on the right) are shown in the context of the assembly. A misjoin is visible in the middle of the example, indicated by the drop in Pacific Biosciences (PacBio) read coverage, discordance with the aligned (yellow indicates aligned, and beige, not aligned) Bionano maps, and the break in synteny. The alignments with intermediate assembly stages show that this error was introduced by the scaffolding step involving scaffold10x. (B–E) Assembly issues identified in HiGlass Hi-C 2D maps of a human assembly (HG002, varying assembly approaches). Scaffold boundaries are delineated in gray. (B) The first of the 2 scaffolds depicted here shows a misjoin (black arrow) that needs to be broken. The second scaffold reveals no structural issues. (C) The first and third of the 3 scaffolds shown here need to be joined as indicated by the green arrows. (D) The single scaffold depicted here has a misjoin (black arrow) that needs to be broken and rejoined as indicated by the green arrows. (E) This single scaffold contains a duplication, half of which needs to be excised (e.g., black arrows) and the scaffold rejoined (green arrows). The choice of the excised half can be based on phasing.

Once identified, errors should be corrected. We have found that whole-genome sequence editing tools, such as gap5 [34], are useful for this process. It is critical to record the corrections made so that the path from primary assembly to the final completed genome assembly is clear and justified.

Identifying and Naming Chromosome-Scale Scaffolds

The ultimate goal of genome assembly is the production of fully contiguous nucleotide sequences that represent each of the



Figure 3: Comparison of the *fbn2b* region in the *Danio rerio* (zebrafish) reference assemblies Zv9 (top), GRCz10 (middle), and GRCz11 (bottom) in gEVAL. The fragmented *fbn2b* locus (colour coded in orange and red) was adjusted for GRCz10 (colour coded in orange) and further improved by removing whole-genome shotgun contigs in favour of finished clone sequence for GRCz11. The final correct gene locus is indicated in green.

chromosomal units for the species, with an estimate of both overall and local quality, and with known sites that may have issues flagged. Long-range data, such as Hi-C contact maps, can reliably indicate which scaffolds correspond to chromosomal units, and these putative chromosomal assemblies can be reconciled with karyotypic information where available. Fully resolved chromosomal units (where all contigs and scaffolds are ordered and oriented) can be submitted to the the INSDC sequence archives (the INSDC partners: GenBank, European Nucleotide Archive [ENA], and DNA Data Bank of Japan [DDBJ]) as a “chromosome.” Scaffolds and contigs that are demonstrably associated with a chromosomal unit but that cannot be joined because of ambiguous order or orientation must be submitted as “unlocalized” for this chromosome. Scaffolds and contigs that cannot be associated with a chromosome and that also cannot be established as being separate chromosomes are deemed “unplaced.”

If a reference assembly for the same species or a karyotype with sequence-based anchors is available, chromosome naming should follow the precedent to ensure compatibility with previously reported results. Identification of sex chromosomes can be based on comparisons to related species or the location of marker genes. In heterogametic individuals, sex chromosomes will also be easily recognizable by their halved sequence coverage compared with autosomes. If no reference for chromosome naming is established, they should be named by size.

Last but not least, every assembly, together with all relevant raw and metadata, should be submitted to one of the INSDC archives (Genbank, ENA, or DDBJ [35]) to allow discoverability, ensure community access, and provide stability for future analyses.

Fig. 1 summarizes the above recommendations in a suggested workflow for assembly curation activities.

Assembly Curation for High-Throughput Projects

The aforementioned curation processes suffer from the same shortcoming as the assembly process itself: they are usually applied in series rather than in parallel. The benefits of a multitude of data types and approaches are also difficult to realize. Whilst the identification of many assembly issues can be automated, the actual decision to apply a change is still best made by an experienced curator, seemingly slowing the process to an extent that excludes it from any high-throughput project.

The Genome Reference Informatics Team (GRIT) assembly curation pipeline was established to deliver high-quality assembly curation for the Genome Reference Consortium (GRC [36]), the VGP, and DTOL. The pipeline automates the processes of data gathering and computational analysis for decontamination, validation, and correction of assemblies, sourcing all available data from in-house and public resources. The analyses are then presented for manual evaluation by experienced genome curators, who perform the evaluation and log required changes. The corrected assembly ready for submission is generated automatically. Central to this pipeline is gEVAL, the genome evaluation browser [9]. gEVAL enables visualization and evaluation of discordances between an assembly and multiple sets of accompanying data in parallel, enabling the simultaneous identification of errors and ways to resolve them [37]. The pipeline that GRIT deploys has much evolved since its first implementation [9] and is now so closely tied into the Wellcome Sanger Institute’s internal data structure that it cannot be ported, but is described here as an example of a successful implementation that mixes automated and manual processes and significantly improves genome assemblies in a time- and resource-sensitive way that allows its use within high-throughput projects. All as-

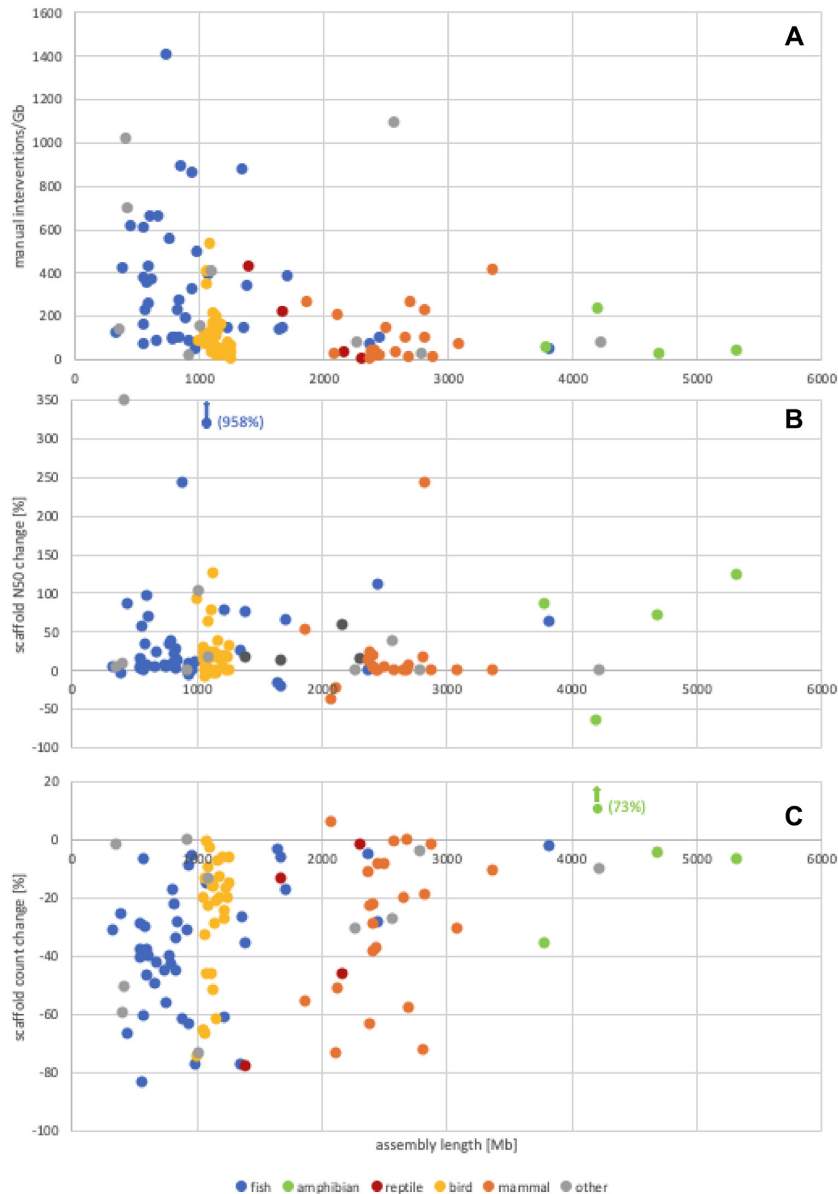


Figure 4: Changes to 111 assemblies from different clades through manual assembly curation by the Genome Reference Informatics Team at the Wellcome Sanger Institute. (A) Manual interventions (breaks, joins, removal of false duplications) as events per gigabase of assembly sequence. (B) Changes in scaffold N50 after curation. (C) Changes in scaffold counts after curation. The depicted assemblies were created with PacBio CLR, Chromium 10X, Bionano, and Hi-C data.

sembly projects loaded into gEVAL are publicly accessible at geval.org.uk.

The GRIT curation process usually starts with assemblies that have been purged of duplicates and most haplotypic segments, scaffolded with long-range data and polished. Before being loaded into gEVAL, all assemblies are run through a nextflow [38] pipeline that performs contamination detection and separation or removal as described in Table 1, combined with removal of trailing Ns [38]. Brief manual checking of the results prevents the erroneous removal of regions likely derived from horizontal gene transfer. This pipeline was inspired by the contamination checking process conducted by Genbank [39].

gEVAL analyses are collated in a database built on an Ensembl framework [40] that has been modified to visualize assembly quality rather than gene and feature annotation. Loading of the analyses into gEVAL and subsequent assembly analyses are

pipelined using snakemake and vr-runner [41, 42]. Which analyses are run and visualized depend on the availability of data, but typically include the types listed in Table 2. The alignments and placements are visualized in a genome browser as feature tracks and colour coded to indicate agreement or disagreement with the assembly (Fig. 2). The gEVAL process also generates lists that detail discordances between the assembly and the different data types. The process of analysis and loading into gEVAL requires up to 3 days for a 1-Gb assembly.

gEVAL automatically flags areas where the raw and other comparative data available are discordant with the presented assembly. Experienced curators use the gEVAL database and visualization, and (where available) Hi-C maps (generated outside the gEVAL pipeline and viewed in HiGlass [29] or pretext [30]), to check each listed discordance and decide whether and how to adjust the sequence on the basis of the available data (Fig. 2).

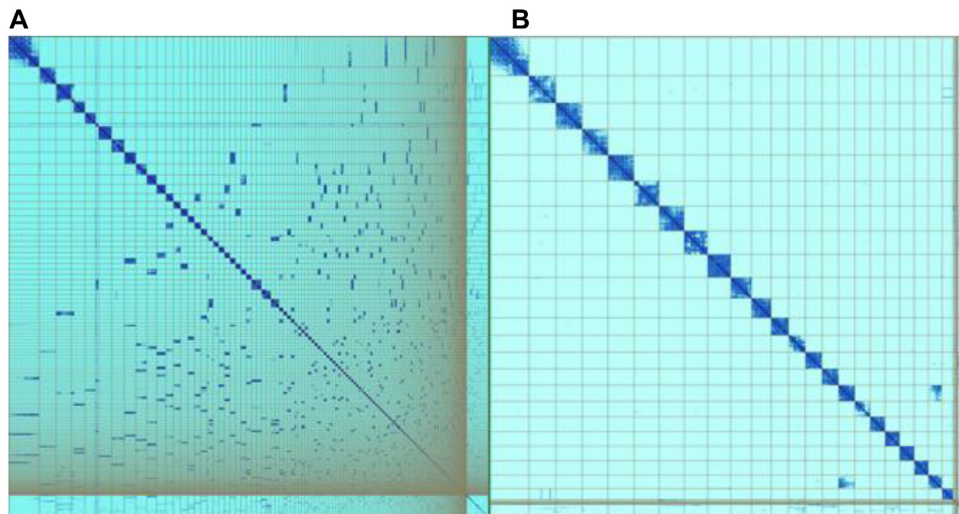


Figure 5: Hi-C maps (pretext) showing the *Asterias rubens* (starfish) genome assembly (sequenced as part of the Sanger Institute's 25 Genomes for 25 Years project) before (A) and after (B) curation. The curation corrected the initial assembly by making 75 breaks and 216 joins and removed 1 stretch of erroneously duplicated sequence. A total of 97% of the assembly sequence could be assigned to 22 chromosomes. The curated assembly (B) contains 1 scaffold that is known to be associated with a second one (off-diagonal signal at bottom right), but its order and orientation are ambiguous. This scaffold has been submitted as “unlocalized” for the relevant chromosome.

In rare cases, the information contained in gEVAL and the Hi-C maps is not sufficient to decide whether a change is warranted. The curators then use additional tools such as gap5 [34] for in-depth analysis of aligned reads or Genomicus for information on synteny with other species [51]. Curators propose a variety of interventions such as breaking or joining sequence regions, changing the order and orientation of scaffolds and contigs, and removing false duplications. Detangling sequence collapses is currently only possible where additional data can be used for local reassembly. In high-throughput projects such as DTOL or VGP, curation is usually restricted to a resolution of ~100 kb. This allows an experienced curator to complete curation of 1 Gb of sequence in ~3 days. For projects without immediate time constraints and aimed at single references, such as the genomes curated within the GRC, there is no resolution limit.

During the gEVAL build, assembly scaffolds are split into equally sized components, with their order and orientation recorded in a path file under version control, listing component name, scaffold name, and orientation. Should any rearrangement be necessary, the curators simply reorder/reorient the components in the path file. If necessary, components can be split with bespoke scripts that create new components and store them in the gEVAL database. After manual curation, the adjusted ordering and orientation of components and a list of scaffold-chromosome associations are processed automatically to generate the final assembly for submission. All milestones and metrics of the whole curation process are recorded in a tracking database.

Using gEVAL to Assess Published Assemblies

Above we have described the use of gEVAL to create high-quality assemblies. gEVAL can also be used to support research communities in verifying research results, ensuring that they are not based on assembly artefacts. For this, a gEVAL database is generated for publicly available assemblies, as, e.g., is the case for all GRC assemblies [37]. Here, gEVAL offers the same analyses as

detailed above, plus additional databases with other assemblies of the same species, such as previous versions of the current reference, including whole-genome alignments between them (Fig. 3). Combined with tutorials and documentation, this provides a valuable resource for users of the featured reference assemblies.

Impact of Assembly Curation for High Throughput-Projects

During curation of 111 assemblies (174 Gb sequence) for VGP and DTOL, on average 221 interventions per Gb of sequence were applied (67 breaks, 105 joins, and 49 removals of false duplications, Fig. 4). These changes led to a mean reduction in assembly length by 2% because the curation effort did not generate new sequence. However, mean scaffold N50 increased by 40% and scaffold number decreased by 29%. It is important to note that scaffold N50 changes differed for each assembly and that while the process improved N50 several hundred fold in initially fragmented assemblies it halved the N50 in over-scaffolded assemblies. On average 96% of assembly sequence was scaffolded to chromosome level (Fig. 5). The number and scale of changes to the assemblies necessary across the diversity of species analysed shows the persistent need for manual intervention on the path to high-quality genome assemblies. Our experiences in curating partially and fully haplotype-resolved genome assemblies for GRC, VGP, and DTOL have driven improvements in assembly software (e.g., purge.dups [14], salsa2 [52]), assembly pipelines (VGP, DTOL), and assembly assessment tools (e.g., Asset [31, 37]). Genome assembly generation is a fast-moving field, and we are constantly adapting the curation software and processes to include novel data types and novel ways of generating assemblies whilst being conscious of the need to maximize throughput. This ensures ongoing involvement of assembly curation in high-throughput projects to produce the best possible data for the community to base their research upon.

Abbreviations

BLAST: Basic Local Alignment Search Tool; bp: base pairs; DDBJ: DNA Data Bank of Japan; DTOL: Darwin Tree of Life Project; ENA: European Nucleotide Archive; Gb: giga base pairs; GC: guanine-cytosine; GRC: Genome Reference Consortium; GRIT: Genome Reference Informatics Team; Hi-C: high-throughput chromosome conformation capture; INSDC: International Nucleotide Sequence Database Collaboration; NUMT: nuclear mitochondrial transfer; NUPT: nuclear plastid transfer; VGP: Vertebrate Genomes Project.

Competing Interests

The authors declare that they have no competing interests.

Funding

This project is supported by the Wellcome Trust, WT206194.

Authors' Contributions

KH incepted and wrote the manuscript, and performed summary analyses of the curation effort. JT implemented the decontamination and postprocessing pipelines and performs decontamination and curation postprocessing. WC and YS implemented gEVAL and its analysis pipeline, and produce gEVAL databases together with JW and DLP. JC, SP, AT and JW curate assemblies.

Acknowledgements

We thank Mark Blaxter, Richard Durbin, and Shane McCarthy for their input on this project and many helpful discussions. Additional thanks go to Mark Blaxter for many constructive discussions around this manuscript. The genome curation project is heavily influenced by the Genome Reference Consortium, the Vertebrate Genomes Project, and the Darwin Tree of Life Project, and we are indebted to all the members for their engagement with the curation process.

References

- Rhie A, McCarthy SA, Fedrigo O, et al. Towards complete and error-free genome assemblies of all vertebrate species. *bioRxiv* 2020, doi:10.1101/2020.05.22.110833.
- Miga KH, Koren S, Rhie A, et al. Telomere-to-telomere assembly of a complete human X chromosome. *Nature* 2020;585:79–84.
- Yang L-A, Chang Y-J, Chen S-H, et al. SQUAT: a Sequencing Quality Assessment Tool for data quality assessments of genome assemblies. *BMC Genomics* 2019;19:238.
- Mapleson D, Garcia Accinelli G, Kettleborough G, et al. KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. *Bioinformatics* 2017;33:574–6.
- Seppy M, Manni M, Zdobnov EM. BUSCO: Assessing Genome Assembly and Annotation Completeness. *Methods Mol Biol* 2019;1962:227–45.
- Rhie A, Walenz BP, Koren S, et al. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. *Genome Biol* 2020;21:245.
- Chan S, Lam E, Saghbini M, et al. Structural variation detection and analysis using Bionano optical mapping. *Methods Mol Biol* 2018;1833:193–203.
- Durand NC, Robinson JT, Shamim MS, et al. Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom. *Cell Syst* 2016;3:99–101.
- Chow W, Brugger K, Caccamo M, et al. gEVAL - a web-based browser for evaluating genome assemblies. *Bioinformatics* 2016;32:2508–10.
- Lewin HA, Robinson GE, Kress WJ, et al. Earth BioGenome Project: Sequencing life for the future of life. *Proc Natl Acad Sci U S A* 2018;115:4325–33.
- Rhie A. Mash Pipeline. <https://github.com/VGP/vgp-assembly/tree/master/pipeline/mash>. Accessed 17 July 2020.
- van Haarst J, Plaza Oñate F, Karasikov M. KMSSDS. KMC. <https://github.com/refresh-bio/KMC>. Accessed 17 July 2020.
- Rhyker Ranallo-Benavidez T, Jaron KS, Schatz MC. GenomeScope 2.0 and Smudgeplot for reference-free profiling of polyploid genomes. *Nat Commun* 2020;11(1):1432.
- Guan D, McCarthy SA, Wood J, et al. Identifying and removing haplotypic duplication in primary genome assemblies. *Bioinformatics* 2020;36:2896–8.
- Vollger MR, Dishuck PC, Sorensen M, et al. Long-read sequence and assembly of segmental duplications. *Nat Methods* 2019;16:88–94.
- Formenti G, Rhie A, Balacco J, et al. Complete vertebrate mitogenomes reveal widespread gene duplications and repeats. *bioRxiv* 2020, doi:10.1101/2020.06.30.177956.
- Challis R, Richards E, Rajan J, et al. BlobToolKit - Interactive quality assessment of genome assemblies. *G3 (Bethesda)* 2020;10:1361–74.
- Eren AM, Murat Eren A, Esen ÖC, et al. Anvi'o: an advanced analysis and visualization platform for 'omics data. *PeerJ* 2015;3:e1319.
- Contamination in sequence databases. <https://www.ncbi.nlm.nih.gov/tools/vecscreen/contam/>. Accessed 17 July 2020.
- Hancock JM, Bishop MJ. VecScreen. In: *Dictionary of Bioinformatics and Computational Biology*. 2004, doi:10.1002/9780471650126.dob0783.pub2.
- UniVec. <https://www.ncbi.nlm.nih.gov/tools/vecscreen/uni-vec/>. Accessed 17 July 2020.
- Morgulis A, Coulouris G, Raytselis Y, et al. Database indexing for production MegaBLAST searches. *Bioinformatics* 2008;24:1757–64.
- Contamination in eukaryotes. <ftp://ftp.ncbi.nlm.nih.gov/pub/kitts/contam.in.euks.fa.gz>. Accessed 17 July 2020.
- RefSeq. RefSeq assemblies: mitochondria. <ftp://ftp.ncbi.nlm.nih.gov/refseq/release/mitochondrion/>. Accessed 17 July 2020.
- RefSeq. RefSeq assemblies: plastids. <ftp://ftp.ncbi.nlm.nih.gov/refseq/release/plastid/>. Accessed 17 July 2020.
- Morgulis A, Gertz EM, Schäffer AA, et al. WindowMasker: window-based masker for sequenced genomes. *Bioinformatics* 2006;22:134–41.
- O'Leary NA, Wright MW, Brister JR, et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res* 2016;44:D733–45.
- Ning ZHE. Scaff10X v4.2: Pipeline for scaffolding and breaking a genome assembly using 10x genomics linked-reads. <https://github.com/wtsi-hpag/Scaff10X>. Accessed 17 July 2020.
- Kerpedjiev P, Abdennur N, Lekschas F, et al. HiGlass: web-based visual exploration and analysis of genome interaction maps. *Genome Biol* 2018;19:125.
- Harry E. PretextView (Paired Read TEXTure Viewer): A desktop application for viewing pretext contact maps. <https://github.com/wtsi-hpag/PretextView>. Accessed 17 July 2020.

31. Guan D. Asset: An assembly evaluation tool. <https://github.com/dfguan/asset>. Accessed 17 July 2020.
32. Melters DP, Bradnam KR, Young HA, et al. Comparative analysis of tandem repeats from hundreds of species reveals unique insights into centromere evolution. *Genome Biol* 2013;**14**:R10.
33. Koren S. Find.telomere. <https://github.com/VGP/vgp-assembly/tree/master/pipeline/telomere>. Accessed 17 July 2020.
34. Bonfield JK, Whitwham A. Gap5—editing the billion fragment sequence assembly. *Bioinformatics* 2010;**26**:1699–703.
35. Karsch-Mizrachi I, Takagi T, Cochrane G, International Nucleotide Sequence Database Collaboration. The International Nucleotide Sequence Database Collaboration. *Nucleic Acids Res* 2018;**46**:D48–51.
36. Church DM, Schneider VA, Graves T, et al. Modernizing reference genome assemblies. *PLoS Biol* 2011;**9**:e1001091.
37. Genome Reference Informatics Team. gEVAL: The Genome Evaluation Browser. <https://geval.org.uk/>. Accessed 17 July 2020.
38. Di Tommaso P, Chatzou M, Floden EW, et al. Nextflow enables reproducible computational workflows. *Nat Biotechnol* 2017;**35**:316–9.
39. ContamFilter. GitHub. <https://github.com/NCBI-Hackathons/ContamFilter>. Accessed 12 November 2020.
40. Aken BL, Achuthan P, Akanni W, et al. Ensembl 2017. *Nucleic Acids Res* 2017;**45**:D635–42.
41. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics* 2018;**34**:3600.
42. Danecek P, McCarthy S, Randall JC, et al. vr-runner: A lightweight pipeline framework. <https://github.com/VertebrateResequencing/vr-runner>. Accessed 17 July 2020.
43. Li H. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* 2016;**32**:2103–10.
44. Jain C, Rhie A, Zhang H, et al. Weighted minimizer sampling improves long read mapping. *Bioinformatics* 2020;**36**(Supplement_1), doi:10.1093/bioinformatics/btaa435.
45. Kent WJ. BLAT—The BLAST-Like Alignment Tool. *Genome Res* 2002;**12**:656–64.
46. Wang M, Kong L. Pblat: A multithread Blat algorithm speeding up aligning sequences to genomes. *BMC Bioinformatics* 2019;**20**(1):28.
47. Kurtz S, Phillippy A, Delcher AL, et al. Versatile and open software for comparing large genomes. *Genome Biol* 2004;**5**:R12.
48. Bao W, Kojima KK, Kohany O. Repbase Update, a database of repetitive elements in eukaryotic genomes. *Mob DNA* 2015;**6**:11.
49. Shyu C, Foster JA, Forney LJ. Electronic polymerase chain reaction (EPCR) search algorithm. In: *Proceedings. IEEE Computer Society Bioinformatics Conference, Stanford, CA. 2002*, doi:10.1109/csb.2002.1039361
50. Altschul SF, Gish W, Miller W, et al. Basic Local Alignment Search Tool. *J Mol Biol* 1990;**215**:403–10.
51. Nguyen NTT, Vincens P, Roest Crollius H, et al. Genomicus 2018: karyotype evolutionary trees and on-the-fly synteny computing. *Nucleic Acids Res* 2018;**46**:D816–22.
52. Ghurye J, Pop M, Koren S, et al. Scaffolding of long read assemblies using long range contact information. *BMC Genomics* 2017;**18**:527.