Received: 22 July 2019 | Revised: 21 October 2019 | Accepted: 23 October 2019

DOI: 10.1111/1755-0998.13108

RESOURCE ARTICLE

MOLECULAR ECOLOGY RESOURCES WILEY

Accuracy of *de novo* assembly of DNA sequences from double-digest libraries varies substantially among software

Melanie E. F. LaCava[1,2] | Ellen O. Aikens[1,3] | Libby C. Megna[1,4] | Gregg Randolph[5] | Charley Hubbard[1,6] | C. Alex Buerkle[1,6]

[1]Program in Ecology, University of Wyoming, Laramie, WY, USA

[2]Wildlife Genomics and Disease Ecology Laboratory, Department of Veterinary Sciences, University of Wyoming, Laramie, WY, USA

[3]Wyoming Cooperative Fish and Wildlife Research Unit, Department of Zoology and Physiology, University of Wyoming, Laramie, WY, USA

[4]Department of Zoology and Physiology, University of Wyoming, Laramie, WY, USA

[5]Genome Technologies Lab, University of Wyoming, Laramie, WY, USA

[6]Department of Botany, University of Wyoming, Laramie, WY, USA

Correspondence
Melanie E. F. LaCava, Department of Veterinary Sciences, University of Wyoming, 1000 E. University Ave., Laramie, WY 82071, USA.
Email: mlacava@uwyo.edu

Funding information
National Science Foundation, Grant/Award Number: NNX15AI08H; Wyoming NASA Space Grant Consortium; University of Wyoming

Abstract

Advances in DNA sequencing have made it feasible to gather genomic data for non-model organisms and large sets of individuals, often using methods for sequencing subsets of the genome. Several of these methods sequence DNA associated with endonuclease restriction sites (various RAD and GBS methods). For use in taxa without a reference genome, these methods rely on *de novo* assembly of fragments in the sequencing library. Many of the software options available for this application were originally developed for other assembly types and we do not know their accuracy for reduced representation libraries. To address this important knowledge gap, we simulated data from the *Arabidopsis thaliana* and *Homo sapiens* genomes and compared *de novo* assemblies by six software programs that are commonly used or promising for this purpose (ABySS, CD-HIT, Stacks, Stacks2, Velvet and VSEARCH). We simulated different mutation rates and types of mutations, and then applied the six assemblers to the simulated data sets, varying assembly parameters. We found substantial variation in software performance across simulations and parameter settings. ABySS failed to recover any true genome fragments, and Velvet and VSEARCH performed poorly for most simulations. Stacks and Stacks2 produced accurate assemblies of simulations containing SNPs, but the addition of insertion and deletion mutations decreased their performance. CD-HIT was the only assembler that consistently recovered a high proportion of true genome fragments. Here, we demonstrate the substantial difference in the accuracy of assemblies from different software programs and the importance of comparing assemblies that result from different parameter settings.

KEYWORDS

GBS, genomics, indels, paralogs, population, RAD, reference genome

1 | INTRODUCTION

Advances in DNA sequencing have made the laboratory portion of large studies of genomic variation among many individuals feasible and economical, even for non-model taxa (Andrews, Good, Miller, Luikart, & Hohenlohe, 2016; Benestan et al., 2016; Ekblom & Galindo, 2011; Narum, Buerkle, Davey, Miller, & Hohenlohe, 2013). Current research in population genomics is often based on sequencing of reduced representation DNA libraries for many individuals, rather than the equivalent amount of whole genome re-sequencing for a smaller set of individuals (Buerkle & Gompert, 2013; Fumagalli, 2013). Ecological and evolutionary research in diverse systems

often uses DNA sequencing to obtain genotypic information, regularly without a complete reference genome (draft genome assembly with long continguous sequences, for the focal or a related taxon). Instead, many studies rely on *de novo* assemblies of the subset of the genome contained in reduced representation libraries to use for reference-based mapping of reads. Despite the fact that the *de novo* assembly is prerequisite for further analysis and has strong potential to affect reference-based assemblies and genotyping, few studies have compared the efficacy of different software for sequence assembly from reduced representation libraries, including various protocols for genotyping-by-sequencing (GBS) and restriction associated DNA (RAD) sequencing (e.g., Baird et al., 2008; Elshire et al., 2011; Parchman et al., 2012; Peterson, Weber, Kay, Fisher, & Hoekstra, 2012), which collectively we will refer to as GBS hereafter.

We performed a literature review to assess which *de novo* assembly programs are commonly used for processing GBS data. We report the software that was used in those studies and the extent to which papers evaluated and reported different assemblies based on different software settings and parameters. Some software was designed specifically for assembling RAD sequences (e.g., STACKS; Catchen, Amores, Hohenlohe, Cresko, & Postlethwait, 2011; Rochette, Rivera-Colon, & Catchen, 2019). Other programs were designed for assembly of whole genomes, transcriptomes, or protein sequences, including assembling contiguous sequences (contigs) that are longer than raw reads from the sequencing instrument (e.g., VELVET; Zerbino & Birney, 2008). These assemblers are sometimes applied to reduced representation sequences despite being designed for a different type of data, and the appropriateness of this application has not been directly evaluated.

There are a variety of parameter options for a given assembly method. For example, the assembly module in the software STACKS allows users to vary over a dozen parameters, whereas the software CD-HIT requires users to define only a few parameters (Catchen et al., 2011; Li & Godzik, 2006). Although assembly parameters such as percent match (i.e., threshold required for reads to be clustered into a single contig) can significantly alter resulting assemblies (Harvey et al., 2015; Ilut, Nydam, & Hare, 2014), few studies report any attempts to optimize these parameters. Resources to compare assemblies with different parameter settings and optimize assembly performance are increasingly available, but so far are underutilized (Harvey et al., 2015; Ilut et al., 2014; Paris, Stevens, & Catchen, 2017; Puritz, Hollenbeck, & Gold, 2014).

In this study, we evaluated the performance of a sample of leading *de novo* assembly programs. We used simulations of sequence reads for a population of individuals to compare *de novo* assembly software programs in terms of the accuracy of the resulting assemblies, given known locations for sequence reads within each genome. As representative, well-assembled genomes, we used the human (*H. sapiens* hereafter; GRCh38 retrieved from `genome.ucsc.edu` in January 2014, Lander, Linton, Birren, Nusbaum, & Zody, 2001) and *Arabidopsis thaliana* (*A. thaliana* hereafter; TAIR 10 assembly, Lamesch et al., 2012) genomes and their expected fragmentation through double digestion with two restriction enzymes (EcoRI and MseI). We simulated different rates of mutation and different mutation types (single nucleotide polymorphisms and insertions/deletions) to evaluate assembler performance with different genome characteristics. Additionally, we investigated the sensitivity of the assemblers to two parameters that are used in their algorithms: percent match and k-mer length. We compared the six assemblers by quantifying different types of errors in their assemblies of our simulated data. We compared the completeness of the assemblies (fraction of all true genome fragments represented) and their degree of over-assembly (i.e., collapsing multicopy, paralogous loci into a single contig) and under-assembly (i.e., separating allelic variants at a single locus into different contigs).

2 | MATERIALS AND METHODS

2.1 | Literature review

We performed a literature search using the Web of Science database to quantify the frequency of use of different assemblers in current GBS studies. Our search terms were "double digest" or "genotyping by sequencing" or "restriction site-associated". We limited the search to papers from 1 January 2012 to 18 September 2017, a period that is relevant to GBS methods and generated an adequate sample size. We retained papers that presented new GBS data, performed a library preparation method that included digestion with two enzymes, and performed *de novo* assembly. We excluded single enzyme GBS studies because the assembly of single-digest fragments is a more complex problem, in part because their sequence reads are of either DNA strand surrounding the restriction site. We reviewed the papers in reverse chronological order of publication date and retained the first 100 papers that met these criteria to evaluate a reasonable subset of the relevant literature. For these 100 papers, we documented the *de novo* assembly software that was used and whether the authors varied assembler parameters, including percent match, k-mer length, or any other parameter.

2.2 | Assembler selection

We selected six software programs for assessment of their performance: ABYSS, CD-HIT, STACKS, STACKS2, VELVET, and VSEARCH. We chose a sample of assemblers that were presented in peer-reviewed publications, employed a variety of assembly strategies, and were freely available. Additionally, assemblers were only included in the assessment if they had adequate and up-to-date user resources available online (e.g., manual, tutorial, user help forums). Lastly, we included assemblers that were commonly used in the published literature, and therefore likely of interest to researchers currently performing *de novo* assembly. The six assembly programs we selected represent variations on two clustering algorithms (graph-based and greedy clustering algorithms), and together these programs were used in 55% of the papers in our literature review. Although our comparison is not a complete list of assemblers meeting the desired

criteria, our aim was to investigate variation in performance among a sample of currently available software.

We included four assemblers that use graph-based algorithms in our comparison: STACKS (version 1.46), STACKS2 (version 2.1), ABYSS (version 1.3.4) and VELVET (version 1.1) (Catchen et al., 2011; Rochette et al., 2019; Simpson et al., 2009; Zerbino & Birney, 2008). We evaluated both STACKS and STACKS2 due to significant changes in the software related to how insertion and deletion (indel) variation is treated (changes to STACKS2 since the version 2.1 we used have not included changes to *de novo* assembly). These four assemblers apply graph theory, whereby nodes represent unique reads and edges connect nodes that have sequence segments in common. Graphs are constructed using maximum likelihood to cluster reads into contigs (Catchen et al., 2011). All four assemblers rely on input parameters to vary assembly constraints, but little guidance exists for their use with GBS data, except for efforts to aid users in selecting parameters for STACKS and STACKS2 (Paris et al., 2017). Because each assembler includes some unique parameters, we set parameters that we were not explicitly testing to comparable values when possible. For example, in STACKS and STACKS2 we set the minimum depth of coverage required to create a contig at 1 to mimic other assemblers having no rule for minimum requirement and performed an ungapped alignment. We also allowed assemblers to optimize parameters when the option was available. STACKS, STACKS2, and a script for VELVET (VELVETOPTIMIZER) were used to optimize k-mer length (Gladman & Seeman, 2012). VELVETOPTIMISER is substantially more memory intensive than simply running VELVET so we were unable to use it for the *H. sapiens* simulations. We chose to include assemblers designed specifically for reduced representation data sets (i.e., STACKS, STACKS2), as well as assemblers designed for other applications that are sometimes utilized for reduced representation data sets (i.e., for whole genome assembly using short reads: ABYSS, VELVET).

We included two assemblers, CD-HIT (version 4.6.6) and VSEARCH (version 2.4.0), that use greedy clustering algorithms for assembly (Li & Godzik, 2006; Rognes, Flouri, Nichols, Quince, & Mah´e F, 2016). Greedy clustering algorithms group reads into clusters incrementally by optimizing similarity within contigs and dissimilarity between contigs. Although VSEARCH was intended for *de novo* assembly of metagenomic sequence data, it is also used for alignment and clustering in PYRAD, a program commonly used in GBS studies (Eaton, 2014). CD-HIT was developed for assembling protein sequences, but was later extended for nucleotide sequences, and it is used in the analysis pipeline DDOCENT (Puritz et al., 2014). We used `dDocent`'s data reduction step that retains only one copy of each unique sequence for assembly to reduce computational time (this script can be found in the Dryad repository: https://doi.org/10.5061/dryad.8tr03f8, LaCava et al., 2019).

2.3 | Parameter settings

We varied two parameters, percent match and k-mer length, across all assemblers and simulations to evaluate their influence on assembly. We selected 90%, 94%, and 98% for the minimum percent match to investigate how these interacted with allelic and paralogous variation in the simulated reads. Some assemblers use percent match as a parameter (e.g., CD-HIT), while others use raw mismatch numbers (e.g., STACKS), so an apparently identical parameter setting could produce varied results depending on this distinction. For example, a 100 bp read that includes a 6 bp barcode results in 94 bp reads at the assembly step. For our 98% match setting, the raw mismatch assemblers would allow 2 base pairs to mismatch, while the percent match assemblers would require 98% match, and 0.98 × 94 bp = 92.12, which allows only one base pair to differ. This is consistent across all three levels of percent match tested; for assemblers that use percent match, the mismatch allowed is 1 bp less than for assemblers that use raw mismatch. Table 1 identifies the version of this parameter used in each assembler, and lists base pair equivalents of the percent match used.

TABLE 1 Percent match and k-mer length values tested for each assembler. We tested a range of parameter values possible for each assembler. We also constructed assemblies using the assembler-optimized parameter values, or if the assembler did not have an optimization routine, we used the defaults. Optimized or default parameter values are in bold. For the match parameter, assemblers either use raw mismatch or percent match, and with 94 bp reads this means the same parameter values result in different raw mismatch values, indicated here

Assembler	K-mer length	Match parameter	Raw mismatch allowed
ABYSS	15, 31, **optimized**	Raw mismatch	10, 6, 2
CD-HIT	NA	Percent mismatch	9, 5, 1
STACKS	15, 31, **optimized**	Raw mismatch	10, 6, 2
STACKS2	15, 31, **optimized**	Raw mismatch	10, 6, 2
VELVET	15, 31, **optimized**[a]	Percent mismatch	9, 5, 1
VSEARCH	**8 (default)**, 15	Percent mismatch	9, 5, 1

[a]Note that for VELVET, only the *Arabidopsis thaliana* simulations had optimized k-mer length because the attempted optimization of *Homo sapiens* simulations exceeded available computational resources.

We compared k-mer lengths of 8–31 bp, as well as an assembler-optimized k-mer length, although some assemblers were unable to run with every k-mer length (Table 1). K-mer length represents the sequence length that the assembler algorithm uses to compare reads; that is, the algorithms do not consider the entire, intact sequence at once.

2.4 | Simulations

As representative genomes, we selected the *A. thaliana* (1.44×10^8 base pairs, including gaps and unknown bases, unambiguously

mapped to chromosomes, mtDNA, and cpDNA in TAIR10) and *H. sapiens* (3.85×10^9 base pairs, including gaps and unknown bases, unambiguously mapped to chromosomes and mtDNA in GRCh38) genomes to evaluate to what extent genome complexity influences assembler performance. These genomes differ in total genome size and amount and structure of the repetitive content in their genomes, potentially presenting different challenges for *de novo* assembly.

We used DDRADSEQTOOLS version 0.42 (https://github.com/GGFHF/ddRADseqTools, Mora-Márquez, García-Olivares, Emerson, & Lo´pez de Heredia, 2017) to create in silico ddRAD digests of the *A. thaliana* and *H. sapiens* genomes. From the ddRADseqTools package, we used the script *rsitesearch.py* to obtain fragments from restriction sites for the commonly used EcoRI and MseI restriction enzymes within each of the reference genomes. We then used the script *simddradseq.py* to set parameters and simulate 100 bp single-end reads, including barcodes, from the genome fragments. Because we were not interested in investigating additional complexity introduced by PCR error, we did not simulate PCR duplicates and did not use the PCR duplicate removal step in *pcrdupremoval.py*. We used *indsdemultiplexing.py* to demultiplex the simulated reads and *readstrim.py* to trim the barcodes from the reads, resulting in reads of 94 bp that all began with the EcoRI restriction site. Our wrapper functions for each simulation, modified from *run_ddradseq_chain.sh* in `ddRADseqTools`, are included in the Dryad repository: https://doi.org/10.5061/dryad.8tr03f8, LaCava et al. (2019).

We used the in silico ddRAD digests of *A. thaliana* and *H. sapiens* genomes to simulate sets of reads for each genome. We simulated 100 individuals for all simulations of both genomes. We set `minfragsize` = 350 and `maxfragsize` = 400 to simulate size selection of fragments between 350 and 400 bp (in practice, a larger range of fragments would mean more paralogous loci would be in the sequenced library). We set `minreadvar` = 1 and `maxreadvar` = 1 for all simulations so that read depth was approximately constant across all fragments. We set `locinum` to the number of fragments obtained by *rsitesearch.py* for each reference genome, and then set `readsnum` to a sufficiently high number that all fragments were sampled in the output reads. For *A. thaliana* we set `locinum` = 1,849 and `readsnum` = 3,698,000; for *H. sapiens* we set `locinum` = 45,190 and `readsnum` = 90,380,000.

We generated nine sets of simulated reads per genome with varying rates and types of mutations (Table 2). We varied mutation probability, maximum number of mutations per locus, mutation type (i.e., probability of nucleotide variation [SNP], or nucleotide insertion or deletion [indel]), and maximum mutation length (for indels). Mutations based on these parameters are randomly introduced within simulated reads using the Jukes-Cantor model of sequence evolution (assumes equal nucleotide frequencies and equal mutation rates across base pairs), and the location of mutations is conserved across loci and individuals within each simulated data set. The first simulation had `mutprob` set to zero, so this simulation recovered the original genome. The other eight simulations for each reference genome varied in the amount and types of mutations. We refer to the 350–400 bp reference genome sequences as "fragments", sequences produced by the simulator from the fragments as "reads", and sequences determined by the assemblers to be unique parts of the reference set as "contigs".

2.5 | Quantifying accuracy of assemblies

We constructed GBS assemblies using each assembler for the nine simulated data sets for *A. thaliana* and *H. sapiens*. We used a custom Perl script to compare the assembled contigs with the known fragments from the in-silico digestion of genomes to determine assembly accuracy using two metrics (this script can be found in the Dryad repository: https://doi.org/10.5061/dryad.8tr03f8, LaCava et al., 2019). To evaluate how completely each assembler recovered the original genome fragments (loci), we counted the number of true genome fragments that were represented in the assembly (completeness criterion). We used the simulation's record of what genome fragment a simulated read had been drawn from (recorded in the information line of reads in the simulated fasta data), regardless of whether the read corresponded to the ancestral or a mutated sequence. For the completeness criterion, we counted both assembled contigs that perfectly matched simulated sequences, as well as contigs that were at least 94 bp in length and contained the full length of a simulated sequence, but potentially contained additional bases to accommodate indel variation. Assemblies would be incomplete if some fraction of true fragments and their corresponding ancestral or mutated sequences were not represented in the contigs, because true fragments had been incorrectly subdivided and shortened.

Furthermore, whereas an assembly could be a complete representation of all fragments in the genome, those fragments could be under- or over-assembled relative to the true number of unique genomic regions. Thus, we also compared the number of true fragments in the genome to the number of assembled contigs (over-under assembly criterion). A correct assembly would produce exactly as many contigs as there are unique fragments, regardless of mutations. An under-assembled genome would contain more contigs than there are fragments (i.e., fragments incorrectly split into more contigs than is accurate; a ratio greater than one), whereas an over-assembled genome would contain fewer contigs than there are fragments (i.e., fragments collapsed into fewer contigs than is accurate; a ratio less than one). Over- and under- assembly can significantly affect downstream analyses, and while some assembly errors can be identified and accounted for in subsequent filtering steps, the reliance on filtering to remove assembly errors is not ideal. Therefore, software that avoids both over- and under-assembly is preferable.

3 | RESULTS

3.1 | Literature review

We reviewed a total of 665 papers to find 100 papers that met the desired criteria. The 100 selected papers spanned February 2015

TABLE 2 Parameter settings for in silico restriction enzyme digestions and simulated reads for GBS using `ddRADseqTools` (Mora-Márquez et al., 2017). The parameter `mutprob` sets the probability that a base pair in the locus will mutate. The maximum mutations allowed per locus is set by `locusmaxmut`. The probability that a mutation will be an insertion or deletion (indel) rather than a SNP is set by `indelprob`. The maximum length in base pairs of the indels is set by `maxindelsize`. We simulated reads for both the *Arabidopsis thaliana* and *Homo sapiens* reference genomes using each of these nine parameter combinations

Simulation Name	Description	mutprob	locusmaxmut	indelprob	maxindelsize
Original genome	No mutation in homozygous genome	0	3	0	1
A few SNPs	SNPs only, at a low probability	0.1	3	0	1
More SNPs	SNPs only, at high probability	0.2	3	0	1
Multiple SNPs per locus	High number of mutations per locus	0.1	5	0	1
Mostly SNPs, few indels	Most mutations are SNPs, 10% indels	0.1	3	0.1	1
50:50 SNPs:indels	Half of mutations are SNPs, half indels	0.1	3	0.5	1
Only 1 bp indels	Mutations are mostly 1 bp indels	0.1	3	0.99	1
1–3 bp indels	Mutations are mostly 1–3 bp indels	0.1	3	0.99	3
1–5 bp indels	Mutations are mostly 1–5 bp indels	0.1	3	0.99	5

to October 2017. Of the 100 papers, 39 used STACKS (the period we reviewed preceded the release of STACKS2), 19 used `UNEAK`, 11 used VSEARCH, and 14 used one of the following assemblers: `DNASTAR SeqMan`, `dDocent` (i.e., CD-HIT), or `AftrRAD`. The remaining 17 papers each used a unique assembler (see Table S1). Of 100 papers, 13 reported that they varied percent match, while 12 reported that they varied other assembler parameters. None of the reviewed papers reported varying k-mer length.

3.2 | Simulations

The *A. thaliana* genome contained 1,849 GBS fragments in the 350–400 bp size class, but when we simulated 100 bp reads from these fragments and removed the 6 bp barcode from the beginning of the read, this reduced to 1,813 unique DNA sequences. The *H. sapiens* genome contained 45,190 fragments in the 350–400 bp size class, corresponding to 43,160 unique sequences when trimmed to 94 bp. Since *de novo* assemblers cannot distinguish identical sequences from different parts of the genome, we used the unique 94 bp fragments to represent the expected number of contigs in our analyses of assembler performance.

3.3 | Recovery of genomes without simulated mutation

Across all k-mer length and percent match settings, CD-HIT, STACKS, STACKS2, and VSEARCH recovered at least 96% of the fragments from the unmutated *A. thaliana* genome (Figure 1, Table S2). VELVET assemblies recovered 83%–95.0% of the fragments from the unmutated *A. thaliana* genome, depending on the k-mer length setting (Figure 1). In the larger and more complex *H. sapiens* genome, CD-HIT, STACKS, STACKS2 and VSEARCH recovered at least 87% of the fragments from the unmutated genome across all percent match and k-mer length settings (high completeness; Figure 1). Of all assemblers, CD-HIT with a percent match setting of 98% recovered the highest proportion of fragments (98.3%) from the unmutated *H. sapiens* genome (Table S3). VELVET recovered 17% (k-mer length = 15) to 76% (k-mer length = 31) of the fragments from the unmutated *H. sapiens* genome, regardless of percent match setting (Figure 1). ABYSS failed to recover any full-length contigs that corresponded to fragments from the unmutated *A. thaliana* and *H. sapiens* genomes (Tables S2 and S3). Instead, ABYSS retained contigs that corresponded to fragmented sequence reads, and reported a large number of contigs that were shorter than, and lost information relative to, the simple set of unique sequence reads. Thus, we excluded ABYSS from further analysis.

None of the assemblers recovered the exact number of contigs expected. CD-HIT, STACKS, STACKS2 and VELVET over-assembled contigs to varying degrees (Figure 2). Assemblies from CD-HIT, STACKS and STACKS2 recovered 88%–98% of true fragments (contig/fragment ratios), whereas VELVET assemblies contained 77% (with k-mer length of 31) to 93% of the true number of fragments (k-mer length of 15), regardless of percent match setting. VSEARCH was the only software that under-assembled contigs when assembling the original, unmutated genomes (Figures S1 and S2). For VSEARCH, a k-mer length of eight resulted in approximately the correct number of contigs, but

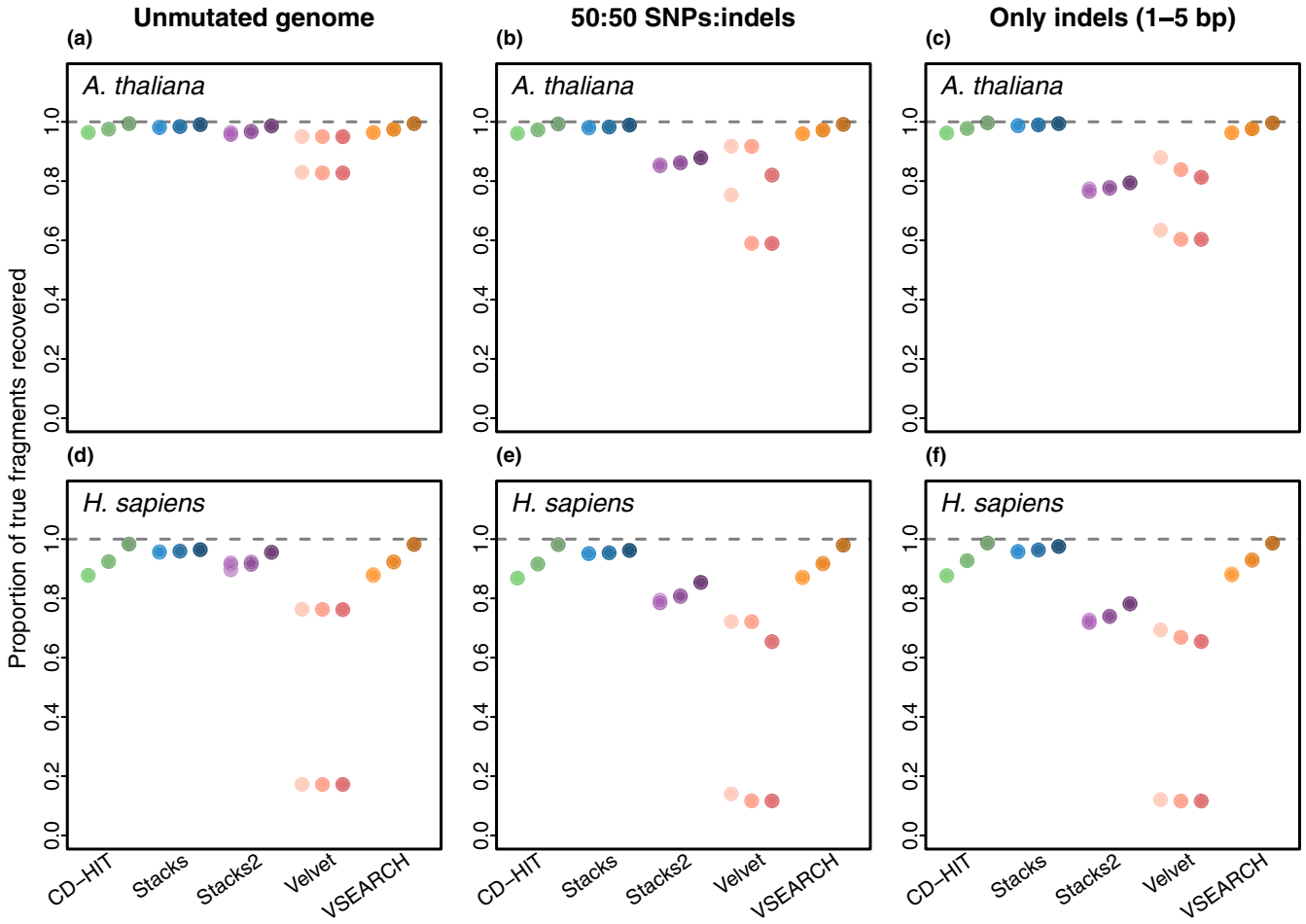

FIGURE 1 The completeness of assemblies in simulations of unmutated genomes (a, d), in simulations of an equal number of SNPs and indels (b, e), and simulations of 1–5 base pair indels (c, f). Simulations were derived from the *Arabidopsis thaliana* (a–c) and *Homo sapiens* (d–f) genomes. Completeness was calculated as the proportion of contigs that matched original genome fragments. A value less than one indicates that some of the assembled contigs were not found in the genome fragments. Values are reported for five assemblers: CD-HIT (green), Stacks (blue), Stacks2 (purple), Velvet (pink) and VSEARCH (orange). The hue of each color corresponds to the percent match parameter setting used in the assembly, with light hues corresponding to 90% match, medium hues corresponding to 94% match, and dark hues corresponding to 98% match. Assemblers have multiple dots in the same hue when k-mer length affected assembly outcome (see Tables S2 and S3 for details) [Colour figure can be viewed at wileyonlinelibrary.com]

a k-mer setting of 15 resulted in under-assembly of the *A. thaliana* genome, producing two times more contigs than the true number of unique fragments (Table S2). Under-assembling was more severe for VSEARCH with the more complex *H. sapiens* genome, resulting in over three times the number of contigs compared to the expected number (Table S3).

3.4 | Sensitivity to SNPs

Across all simulations containing only SNPs, CD-HIT, Stacks, and Stacks2 recovered a high proportion of true genome fragments (Figures S3 and S4) and produced approximately the expected number of contigs (Figures S1 and S2). CD-HIT under-assembled SNP simulations when we used a percent match parameter setting of 98%, and the magnitude of under-assembly was greatest for simulations that allowed up to 3–5 SNPs per locus (Figures S1 and S2), where variation in the simulated reads was higher than variation permitted by the percent match setting. Across all SNP simulations, Velvet assemblies with a k-mer length of 15 resulted in over-assembly (Figures S1 and S2), whereas a k-mer length of 31 resulted in more accurate assemblies, similar to those produced by CD-HIT, Stacks, and Stacks2 (Figures S1 and S2). VSEARCH assemblies of simulated *A. thaliana* data varied from slight over-assembly to considerable under-assembly depending on k-mer length and the probability of SNPs in the simulated data set (Figure S1). VSEARCH assemblies of simulated *H. sapiens* data containing SNPs consistently resulted in under-assembly (Figure S2).

3.5 | Sensitivity to indels

Across all three simulations that introduced only indels as mutations (Table 2), CD-HIT consistently recovered at least 96% of *A.*

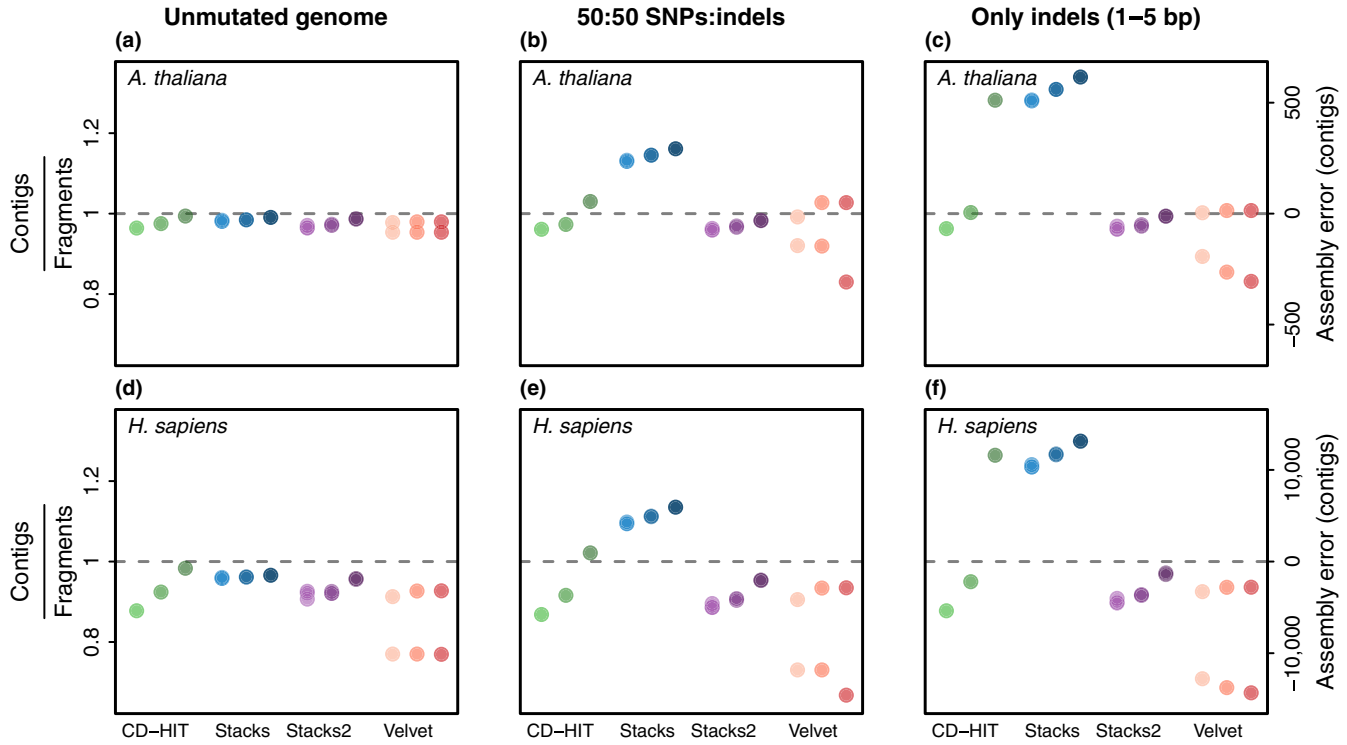

FIGURE 2 Measures of over- and under-assembly in simulations of unmutated genomes (a, d), in simulations of an equal number of SNPs and indels (b, e), and simulations of 1–5 base pair indels (c, f). Simulations were derived from the *Arabidopsis thaliana* (a–c) and the *Homo sapiens* (d–f) genomes. Over- and under-assembly are presented by the ratio of assembled contigs to true genome fragments (left vertical axis) and by absolute numbers (right vertical axis). A contigs:fragments ratio greater than one represents under-assembly and a ratio less than one represents over-assembly. Assembly results are shown for CD-HIT (green), Stacks (blue), Stacks2 (purple) and Velvet (pink) with variable percent match (light hues = 90%, medium hues = 94%, dark hues = 98%). Assemblers have multiple dots in the same hue when k-mer length affected assembly outcome (see Tables S2 and S3 for details). VSEARCH was omitted because its under-assembly was so much greater than other software (VSEARCH is included in Figures S1 and S2) [Colour figure can be viewed at wileyonlinelibrary.com]

thaliana fragments and at least 87% of *H. sapiens* fragments (Figures S3 and S4). CD-HIT only under-assembled with a percent match of 98% when indels were up to 3–5 bp in length, where variation in the simulated reads was higher than variation permitted by the percent match setting (Tables 2, S2 and S3). Stacks consistently recovered at least 95% of true fragments from both genomes, but Stacks also consistently under-assembled, with contig/fragment ratios over 1.2 (Figures S1–S4). This means that although all the contigs originated from true genome fragments, more contigs were produced than expected. In contrast, Stacks2 recovered only 71%–82% of true genome fragments, but produced contig/fragment ratios of 0.90–0.97, meaning that Stacks2 produced closer to the correct number of contigs, but fewer of these contigs were found in the simulated genome fragments (Figures S3 and S4).

Similar to SNP simulations, Velvet assemblies for indel simulations varied in accuracy across k-mer settings. A k-mer setting of 15 produced approximately as many contigs as expected, but as few as 12% of those contigs were found in the original fragments. In contrast, a k-mer setting of 31 produced a contig/fragment ratio as low as 0.72, but a higher percentage of contigs matched true genome fragments (Figures S1–S4). As with SNPs, VSEARCH performance varied between the *A. thaliana* and *H. sapiens* genomes. For *A. thaliana*, VSEARCH varied from slight over-assembly to considerable under-assembly depending on k-mer length and the length of indels simulated (Figure S1). Similar to SNP simulations, all indel simulations for *H. sapiens* resulted in under-assembly when using VSEARCH (Figure S2).

3.6 | Sensitivity to the combination of SNPs and indels

For assemblies of sequences that contained a combination of SNPs and indels (Table 2), CD-HIT, Velvet and VSEARCH performed similarly to simulations where SNPs or indels were introduced independently (Figures S1 and S2). The performance of Stacks with a combination of SNPs and indels, however, produced results intermediate to SNPs or indels independently. In the simulation with mostly SNPs and a few indels (Table 2), Stacks produced approximately the expected number of contigs (Figures S1 and S2). In the simulation with 50:50 SNPs and indels, Stacks consistently under-assembled for both genomes and across percent match settings (Figure 2). The degree of under-assembly by Stacks increased as more indels were introduced to simulations (Figure 2). Stacks2 assemblies were also intermediate for simulations with a combination of SNPs and indels compared to simulations with each mutation type independently.

As the proportion of indels increased, STACKS2 assemblies for both *A. thaliana* and *H. sapiens* were less complete, but the ratio of contigs to fragments remained close to 1 (Figures 1 and 2).

3.7 | Sensitivity to k-mer setting

K-mer length had almost no effect on the proportion of true genome fragments recovered or the number of contigs produced by STACKS or STACKS2 (Tables S2 and S3). However, k-mer length affected assembly outcome for both VELVET and VSEARCH across all simulations that included mutations (Figures S1–S4). For VELVET, k-mer length affected both the proportion of the true fragments recovered and the rate of over-assembly. Across all simulations for both *A. thaliana* and *H. sapiens* genomes, a k-mer length of 15 consistently reduced the completeness of assemblies when compared to assemblies of k-mer length of 31. In contrast, a k-mer length of 31 typically resulted in over-assembly from VELVET, which was more extreme in the complex *H. sapiens* genome. For VSEARCH, k-mer length had little effect on the rate of recovery of true fragments, but can lead to substantial under-assembly (Figures S1 and S2). CD-HIT does not permit users to vary k-mer length, so this parameter was not evaluated for that assembler.

3.8 | Sensitivity to percent match

STACKS and STACKS2 were the least sensitive to varying the percent match parameter setting (Tables S2 and S3). CD-HIT and VSEARCH assemblies were affected by the percent match parameter setting in an expected fashion; for example, increasing percent match to 98% resulted in increased under-assembly for simulations that produced reads that diverged from the original fragments by more than two base pairs (Figures S1 and S2). VELVET assemblies either varied little with the percent match parameter setting (in the case of k-mer lengths of 15) or varied in the opposite direction (in the case of k-mer lengths of 31); the 98% match often caused greater over-assembly than the 90% or 94% match settings (Tables S2 and S3).

4 | DISCUSSION

With any short read sequencing technology (commonly 100–250 bp), there is some ambiguity in the alignment or mapping of those reads because of sequence similarity due to paralogy or allelic variation (Harvey et al., 2015; Ilut et al., 2014). This applies to mapping reads to a high-quality reference genome (e.g., with `bwa`, Li et al., 2010), to the *de novo* assembly of reads as investigated here for GBS data, or to the related challenges for *de novo* assembly of transcriptomes. The ambiguity due to sequence paralogy is evident in the 1.9%–4.5% of GBS loci from the *A. thaliana* and *H. sapiens* genomes that were not distinguishable using 94 bp (assuming 6 bp of the 100 bp reads were used as molecular barcodes to distinguish samples). In typical molecular ecology studies, the problem is compounded by allelic variation in the sample used to construct the *de novo* reference genome. Consequently, some methods and models for identifying variants and calculating genotype likelihoods use mapping quality of reads (e.g., `FreeBayes`; Garrison & Marth, 2012), or use filtering steps to remove sites with low mapping quality scores. Longer reads, or pairs of reads that together are both longer and potentially separated in the genome by some length, will have a higher chance of mapping uniquely, but also have a higher chance of containing nucleotide variants relative to the reference genome or the alleles of other individuals. Thus, the problem of correctly mapping sequences to a reference or *de novo* assembly is general and not restricted to GBS data. In this study we have focused on the assembly problem in the context of GBS, because of the method's common usage (reviewed in Andrews et al., 2016; Benestan et al., 2016; Ekblom & Galindo, 2011; Narum et al., 2013) and its potentially attractive place in the trade-offs that exist among: (a) completeness of genome coverage (low for GBS, relative to whole genome sequencing [WGS]), (b) depth of sequencing at locus (can be optimized, potentially high relative to WGS; Buerkle & Gompert, 2013; Fumagalli, 2013), and (c) numbers of individuals (can be optimized, potentially high relative to WGS) for a finite amount of sequencing. Despite legitimate concerns about the adequacy of genome coverage by GBS-like methods for certain questions and in some systems (Lowry et al., 2016), for many applications in population genomics GBS-like methods are likely to remain attractive for some time (McKinney, 2016). Whereas several studies have examined the consequences of laboratory and bioinformatic methods for variant identification and other downstream analyses (Flanagan & Jones, 2018; Shafer et al., 2017; Warmuth & Ellegren, 2019), and others have suggested methods to optimize assembly parameters (Paris et al., 2017; Puritz et al., 2014), this investigation fills a gap in knowledge regarding the performance of *de novo* assembly software without the aid of additional steps.

Our literature review of 100 recently published papers indicates that STACKS has been the most commonly used *de novo* assembler for GBS data (39 of the reviewed studies), but also that a large variety of software programs are used. Our comparative simulation study showed that STACKS (and STACKS2) recovered true genomes well in the absence of allelic variation, but did less well than CD-HIT (used in only four of 100 reviewed papers) for both the *A. thaliana* and *H. sapiens* genomes when mutations were present (Table 3). In particular, insertion and deletion polymorphisms caused under-assembly of reads for STACKS (as previously demonstrated by Puritz et al., 2014) and a failure to recover a substantial fraction of true genome fragments for STACKS2 (presumably because polymorphisms led to fragmentation of contiguous sequences in the assemblies). CD-HIT was the only assembler that across simulations consistently recovered a high proportion of true genome fragments and its assemblies typically were close to the original genome fragments (with the expected exception in assemblies with a 98% minimum match percentage separating, in which allelic variants with >2% divergence were placed into separate contigs). Two of the other assemblers we considered, VELVET (used in one of 100 reviewed papers) and VSEARCH

TABLE 3 A summary of assembler performance. Assemblers are compared based on recovery of the original genome, assembly outcomes when presented with simulated data with differing degrees of mutations arising from SNPs and indels, and the impact of varying the k-mer and percent match parameter settings

Assembler	Genome recovery	Assembly of SNPs	Assembly of indels	Sensitivity to k-mer	Sensitivity to % match
ABySS	Worst	NA	NA	NA	NA
CD-HIT	Best	Best	Best	NA	Sensitive
Stacks	Best	Best	Poor	Relatively insensitive	Relatively insensitive
Stacks2	Best	Best	Poor	Relatively insensitive	Relatively insensitive
Velvet	Poor	Mixed	Mixed	Very sensitive	Sensitive
VSEARCH	Mixed	Worst	Worst	Very sensitive	Sensitive

(used in 11 of 100 reviewed papers), either performed relatively well at recovering all genome fragments, or at assembling reads into the correct number of genome fragments, but not both. Somewhat dependent on the k-mer setting and the simulation, Velvet assemblies failed to recover a substantial fraction of true fragments (sometimes with counter-intuitive sensitivity to assembler settings), yet over-assembled those fragments only modestly. Whereas VSEARCH assemblies recovered a high fraction of the true genome fragments across simulations, typically the assemblies were drastically under-assembled, particularly for the human genome. Finally, ABySS was poorly suited to *de novo* assembly of GBS reads (it was not used in any of the 100 reviewed studies), in that it resulted in contigs that were exclusively shorter than the original reads and its assemblies did not contain any true genome fragments.

We found that assembly method did not predict assembler performance in any consistent manner (Table 3). We included software that used either graph-based algorithms or greedy-clustering algorithms, and assemblers in each category varied in their performance. The highest performing assemblers, CD-HIT and Stacks2, used different algorithms, suggesting that assembly algorithm is not a useful criterion to select software for *de novo* assembly of GBS data.

For the top performing assemblers, CD-HIT and Stacks2, the challenges to obtaining correct assemblies were as expected: allelic polymorphism due to indel variation at a locus likely led to assembly of shorter tracts of true genome fragments into contigs (Stacks2; see Figures S3 and S4), and sequence divergence among paralogs and alleles made assemblies appropriately sensitive to the minimum percentage match of sequences within a contig. The choice of minimum match percentage that optimizes over- vs. under-assembly will remain a problem for *de novo* assembly until read lengths become much longer than paralogous sequences (allowing them to be placed uniquely in the genome). If not recognized, under- and over-assembly affect downstream analyses, including estimates of population heterozygosity and differentiation (Harvey et al., 2015; Willis, Hollenbeck, Puritz, Gold, & Portnoy, 2017). Of the two, over-assembly is likely preferable for many genomes, as its errors involve closely related, paralogous sequences, which are expected to be rarer than comparable allelic variation at individual loci. Downstream filtering of loci from population samples may identify likely over-assembled paralogs (O'Leary, Puritz, Willis, Hollenbeck, & Portnoy, 2018), though limiting over- and under-assembly from the onset is likely desirable. This post-assembly filtering includes excluding loci based on the distribution of read depth across loci and on improbably high heterozygosity given the allele frequencies at a locus (McKinney, Waples, Seeb, & Seeb, 2017). Ideally, this filtering would be combined with the systematic analysis and comparison of *de novo* assemblies using different percent matches (McCartney-Melstad, Gidi, & Shaffer, 2019; Willis et al., 2017). Tools and methods are available to compare assemblies obtained under different different percent matches (and other settings; McCartney-Melstad et al., 2019; Paris et al., 2017; Rochette & Catchen, 2017) and these should likely become a standard part of population genomics based on *de novo* assemblies. Additionally, some bioinformatic pipelines provide additional steps to improve the accuracy of assemblies, and may therefore result in more accurate assembled RAD loci than the assembler software would produce on its own (e.g., `dDocent`; Puritz et al., 2014).

Our study indicates CD-HIT is a good choice among currently available programs for *de novo* assembly with varying match percentages, and draws attention to the substantial differences among methods that will be beneficial in evaluating new tools for *de novo* assembly of GBS sequences (e.g., `RADProc`; Nadukkalam Ravindran, Bentzen, Bradbury, & Beiko, 2019).

ACKNOWLEDGEMENTS

This project was initiated as part of a computational biology practicum course taught by CAB. All computing was done with the support of the University of Wyoming's Advanced Research Computing Center, on its IBM System X clusters, Mount Moran (http://n2t.net/ark:/85786/m4159c) and Teton (https://doi.org/10.15786/M2FY47). EOA was supported by the National Science Foundation Graduate Research Fellowship Program and the Wyoming NASA Space Grant Consortium (NASA Grant #NNX15AI08H). MEFL was supported by the University of Wyoming Program in Ecology and her major advisor Holly Ernest's Wyoming Excellence Chair funds. T. Parchman provided helpful feedback at several stages of the project and provided valuable comments on a draft of the manuscript. C. Nice also provided valuable comments on a draft of the manuscript.

AUTHOR CONTRIBUTIONS

E.O.A. conducted simulations and compiled results. C.A.B. wrote the Perl scripts to compare assemblies to simulated data, conducted the literature review, and performed the CD-HIT assemblies. C.H. performed the VSEARCH assemblies. M.E.F.L. performed the STACKS, STACKS2, and ABYSS assemblies and conducted the literature review. L.C.M. conducted simulations. G.R. performed VELVET assembles. All authors wrote and reviewed the manuscript.

ORCID

Melanie E. F. LaCava https://orcid.org/0000-0001-7921-9184
Ellen O. Aikens https://orcid.org/0000-0003-0827-3006
Charley Hubbard https://orcid.org/0000-0003-3887-5729
C. Alex Buerkle https://orcid.org/0000-0003-4222-8858

DATA AVAILABILITY STATEMENT

Simulated reads, assembler outputs, and scripts for simulations, assembly, and analysis are available from the Dryad Digital Repository: https://doi.org/10.5061/dryad.8tr03f8 (LaCava et al., 2019).

REFERENCES

Andrews, K. R., Good, J. M., Miller, M. R., Luikart, G., & Hohenlohe, P. A. (2016). Harnessing the power of RADseq for ecological and evolutionary genomics. *Nature Reviews Genetics*, *17*, 81–92. https://doi.org/10.1038/nrg.2015.28

Baird, N. A., Etter, P. D., Atwood, T. S., Currey, M. C., Shiver, A. L., Lewis, Z. A., … Johnson, E. A. (2008). Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PLoS ONE*, *3*, e3376. https://doi.org/10.1371/journal.pone.0003376

Benestan, L. M., Ferchaud, A.-L., Hohenlohe, P. A., Garner, B. A., Naylor, G. J. P., Baums, I. B., … Luikart, G. (2016). Conservation genomics of natural and managed populations: Building a conceptual and practical framework. *Molecular Ecology*, *25*, 2967–2977. https://doi.org/10.1111/mec.13647

Buerkle, C. A., & Gompert, Z. (2013). Population genomics based on low coverage sequencing: How low should we go? *Molecular Ecology*, *22*, 3028–3035. https://doi.org/10.1111/mec.12105

Catchen, J. M., Amores, A., Hohenlohe, P., Cresko, W., & Postlethwait, J. H. (2011). Stacks: Building and genotyping loci de novo from short-read sequences. *G3: Genes, Genomes, Genetics*, *1*, 171–182.

Eaton, D. A. R. (2014). PyRAD: Assembly of de novo radseq loci for phylogenetic analyses. *Bioinformatics*, *30*, 1844–1849. https://doi.org/10.1093/bioinformatics/btu121

Ekblom, R., & Galindo, J. (2011). Applications of next generation sequencing in molecular ecology of non-model organisms. *Heredity*, *107*, 1–15. https://doi.org/10.1038/hdy.2010.152

Elshire, R. J., Glaubitz, J. C., Sun, Q., Poland, J. A., Kawamoto, K., Buckler, E. S., & Mitchell, S. E. (2011). A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. *PLoS ONE*, *6*, e19379. https://doi.org/10.1371/journal.pone.0019379

Flanagan, S. P., & Jones, A. G. (2018). Substantial differences in bias between single-digest and double-digest RAD-seq libraries: A case study. *Molecular Ecology Resources*, *18*, 264–280. https://doi.org/10.1111/1755-0998.12734

Fumagalli, M. (2013). Assessing the effect of sequencing depth and sample size in population genetics inferences. *PLoS ONE*, *8*, 1–11. https://doi.org/10.1371/journal.pone.0079667

Garrison, E., & Marth, G. (2012) Haplotype-based variant detection from short-read sequencing. *arXiv* preprint arXiv:1207.3907.

Gladman, S., & Seeman, T. (2012). *Velvet Optimizer*. https://github.com/tseemann/VelvetOptimiser

Harvey, M. G., Judy, C. D., Seeholzer, G. F., Maley, J. M., Graves, G. R., & Brumfield, R. T. (2015). Similarity thresholds used in DNA sequence assembly fromshort reads can reduce the comparability of population histories across species. *PeerJ*, *2015*, 1–16.

Ilut, D. C., Nydam, M. L., & Hare, M. P. (2014). Defining loci in restriction-based reduced representation genomic data from non-model species: Sources of bias and diagnostics for optimal clustering. *BioMed Research International*, *2014*, 675158. https://doi.org/10.1155/2014/675158

LaCava, M. E. F., Aikens, E. O., Megna, L. C., Randolph, G., Hubbard, C., & Buerkle, C. A. (2019). Data from: Accuracy of de novo assembly of DNA sequences from double digest libraries varies substantially among software. Dryad Digital Repository, https://doi.org/10.5061/dryad.8tr03f8

Lamesch, P., Berardini, T. Z., Li, D., Swarbreck, D., Wilks, C., Sasidharan, R., … Huala, E. (2012). The *Arabidopsis* information resource (TAIR): Improved gene annotation and new tools. *Nucleic Acids Research*, *40*, D1202–D1210. https://doi.org/10.1093/nar/gkr1090

Lander, E., Linton, L., Birren, B., Nusbaum, C., & Zody, M. C., Baldwin, J., … International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*, *409*, 860–921.

Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., … Wang, J. (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, *20*, 265–272. https://doi.org/10.1101/gr.097261.109

Li, W., & Godzik, A. (2006). CD-HIT: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, *22*, 1658–1659. https://doi.org/10.1093/bioinformatics/btl158

Lowry, D. B., Hoban, S., Kelley, J. L., Lotterhos, K. E., Reed, L. K., Antolin, M. F., & Storfer, A. (2016). Breaking RAD: An evaluation of the utility of restriction site associated DNA sequencing for genome scans of adaptation. *Molecular Ecology Resources*, *17*(2), 142–152.

McCartney-Melstad, E., Gidi, M., & Shaffer, H. B. (2019). An empirical pipeline for choosing the optimal clustering threshold in RADseq studies. *Molecular Ecology Resources*, *19*(5), 1195–1204. https://doi.org/10.1111/1755-0998.13029

McKinney, G. J. (2016). RADseq provides unprecedented insights into molecular ecology and evolutionary genetics: comment on Breaking RAD by Lowry et al. (2016). *Molecular Ecology Resources*, *17*(3), 356–361.

McKinney, G. J., Waples, R. K., Seeb, L. W., & Seeb, J. E. (2017). Paralogs are revealed by proportion of heterozygotes and deviations in read ratios in genotyping-by-sequencing data from natural populations. *Molecular Ecology Resources*, *17*, 656–669. https://doi.org/10.1111/1755-0998.12613

Mora-Márquez, F., García-Olivares, V., Emerson, B. C., & López de Heredia, U. (2017). DDRADSEQTOOLS: A software package for in silico simulation and testing of double-digest RADseq experiments. *Molecular Ecology Resources*, *17*, 230–246.

Nadukkalam Ravindran, P., Bentzen, P., Bradbury, I. R., & Beiko, R. G. (2019). RADProc: A computationally efficient de novo locus assembler for population studies using RADseq data. *Molecular Ecology Resources*, *19*, 272–282. https://doi.org/10.1111/1755-0998.12954

Narum, S. R., Buerkle, C. A., Davey, J. W., Miller, M. R., & Hohenlohe, P. A. (2013). Genotyping-by sequencing in ecological and conservation genomics. *Molecular Ecology*, *22*, 2841–2847. https://doi.org/10.1111/mec.12350

O'Leary, S. J., Puritz, J. B., Willis, S. C., Hollenbeck, C. M., & Portnoy, D. S. (2018). These aren't the loci you'e looking for: Principles of effective SNP filtering for molecular ecologists. *Molecular Ecology*, *27*, 3193–3206. https://doi.org/10.1111/mec.14792

Parchman, T. L., Gompert, Z., Mudge, J., Schilkey, F., Benkman, C. W., & Buerkle, C. A. (2012). Genome wide association genetics of an adaptive trait in lodgepole pine. *Molecular Ecology*, *21*, 2991–3005. https://doi.org/10.1111/j.1365-294X.2012.05513.x

Paris, J. R., Stevens, J. R., & Catchen, J. M. (2017). Lost in parameter space: A road map for stacks. *Methods in Ecology and Evolution*, *8*, 1360–1373.

Peterson, B. K., Weber, J. N., Kay, E. H., Fisher, H. S., & Hoekstra, H. E. (2012). Double digest RADseq: An inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PLoS ONE*, *7*, e37135. https://doi.org/10.1371/journal.pone.0037135

Puritz, J. B., Hollenbeck, C. M., & Gold, J. R. (2014). dDocent: A RADseq, variant-calling pipeline designed for population genomics of non-model organisms. *PeerJ*, *2*, e431.

Rochette, N. C., & Catchen, J. M. (2017). Deriving genotypes from 542 RAD-seq short-read data using Stacks. *Nature Protocols*, *12*, 2640–2659.

Rochette, N. C., Rivera-Colon, A. G., & Catchen, J. M. (2019). Stacks 2: Analytical methods for paired end sequencing improve RADseq-based population genomics. *Molecular Ecology*, *28*(21), 4737–4754. https://doi.org/10.1111/mec.15253

Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). VSEARCH: A versatile open source tool for metagenomics. *PeerJ*, *4*, e2584. https://doi.org/10.7717/peerj.2584

Shafer, A. B. A., Peart, C. R., Tusso, S., Maayan, I., Brelsford, A., Wheat, C. W., & Wolf, J. B. W. (2017). Bioinformatic processing of RAD-seq data dramatically impacts downstream population genetic inference. *Methods in Ecology and Evolution*, *8*, 907–917. https://doi.org/10.1111/2041-210X.12700

Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J., & Birol, I. (2009). ABySS: A parallel assembler for short read sequence data ABySS : A parallel assembler for short read sequence data. *Genome Research*, *19*, 1117–1123. https://doi.org/10.1101/gr.089532.108

Warmuth, V. M., & Ellegren, H. (2019). Genotype-free estimation of allele frequencies reduces bias and improves demographic inference from RADSeq data. *Molecular Ecology Resources*, *19*, 586–596. https://doi.org/10.1111/1755-0998.12990

Willis, S. C., Hollenbeck, C. M., Puritz, J. B., Gold, J. R., & Portnoy, D. S. (2017). Haplotyping RAD loci: An efficient method to filter paralogs and account for physical linkage. *Molecular Ecology Resources*, *17*, 955–965. https://doi.org/10.1111/1755-0998.12647

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

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: LaCava MEF, Aikens EO, Megna LC, Randolph G, Hubbard C, Buerkle CA. Accuracy of *de novo* assembly of DNA sequences from double-digest libraries varies substantially among software. *Mol Ecol Resour*. 2020;20:360–370. https://doi.org/10.1111/1755-0998.13108