dDocent: a RADseq, variant-calling pipeline designed for population genomics of non-model organisms

Jonathan B. Puritz†, Christopher M. Hollenbeck, and John R. Gold

Marine Genomics Laboratory, Harte Research Institute, Texas A&M University-Corpus Christi, 6300 Ocean Drive, Corpus Christi, Texas 78412-5869

†Author to whom correspondence should be addressed.

Email: jonathan.puritz@tamucc.edu Phone: 361-825-3343 Fax: 361-825-2050

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014

ABSTRACT

Restriction-site associated DNA sequencing (RADseq) has become a powerful and useful approach for population genomics. Currently, no software exists that utilizes both paired-end reads from RADseq data to efficiently produce population-informative variant calls, especially for organisms with large effective population sizes and high levels of genetic polymorphism but for which no genomic resources exist. *dDocent* is an analysis pipeline with a user-friendly, command-line interface designed to process individually barcoded RADseq data (with double cut sites) into informative SNPs/INDELs for population-level analyses. The pipeline, written in BASH, uses data reduction techniques and other stand-alone software packages to perform quality trimming and adapter removal, *de novo* assembly of RAD loci, read mapping, SNP and INDEL calling, and baseline data filtering. Double-digest RAD data from population pairings of three different marine fishes were used to compare *dDocent* with *Stacks*, the first generally available, widely used pipeline for analysis of RADseq data. *dDocent* consistently identified more SNPs shared across greater numbers of individuals and with higher levels of coverage. This is most likely due to the fact that *dDocent* quality trims instead of filtering and incorporates both forward and reverse reads in assembly, mapping, and SNP calling, thus enabling use of reads with INDEL polymorphisms. The pipeline and a comprehensive user guide can be found at (http://dDocent.wordpress.com).

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014

INTRODUCTION

Next-generation sequencing (NGS) has transformed the field of genetics into genomics by providing DNA sequence data at an ever increasing rate and reduced cost (Mardis, 2008). The nascent field of population genomics relies on NGS coupled with laboratory methods to reproducibly reduce genome complexity to a few thousand loci. The most common approach, restriction-site associated DNA sequencing (RADseq), uses restriction endonucleases to randomly sample the genome at locations adjacent to restriction-enzyme recognition sites that, when coupled with Illumina sequencing, produces high coverage of homologous SNP (Single Nucleotide Polymorphism) loci. As such, RADseq provides a powerful approach for population level genomic studies (Ellegren, 2014;Narum et al., 2013;Rowe et al., 2011).

The original RADseq approach (Baird et al., 2008), and initial population genomic studies employing it (Hohenlohe et al., 2010), focused on SNP discovery and genotyping on the first (forward) read only. This is because the original RADseq method (Baird et al., 2008) utilized random shearing to produce RAD loci; paired-end reads were not of uniform length or coverage, making it problematic to find SNPs at high and uniform levels of coverage across a large proportion of individuals. As a result, the most comprehensive and widely used software package for analysis of RADseq data, *Stacks* (Catchen et al., 2013, 2011), provides SNP genotypes based only on first-read data. In contrast, RADseq approaches such as ddRAD (Peterson et al., 2012), 2bRAD (Wang et al., 2012), and ezRAD (Toonen et al., 2013) rely on restriction enzymes to define both ends of a RAD locus, largely producing RAD loci of fixed length (flRAD). Paired-end Illumina sequencing of flRAD fragments provides an opportunity to significantly expand the number of SNPs that can be genotyped from a single RADseq library.

Here, the variant-calling pipeline *dDocent* is introduced as a tool for generating population genomic data; a brief methodological outline of the analysis pipeline also is presented. *dDocent* is a wrapper script designed to take raw RADseq data and produce population informative SNP calls, taking full advantage of both paired-end reads. *dDocent* is configured for organisms with high levels of nucleotide and INDEL polymorphisms, such as found in many marine organisms (Guo et al., 2012;Keever et al., 2009;Sodergren et al., 2006;Waples, 1998;Ward et al., 1994). As input, *dDocent* takes paired FASTQ files for individuals and outputs raw SNP and INDEL calls as well as filtered SNP calls in VCF format. The pipeline and a comprehensive online manual can be found at (http://dDocent.wordpress.com). Finally, results of pipeline analyses, using both *dDocent* and *Stacks*, of populations of three species of marine fishes are provided to demonstrate the utility of *dDocent* compared to *Stacks,* the first and most comprehensive existing software package for RAD population genomics.

METHODS

Implementation and basic usage

The *dDocent* pipeline is written in BASH and will run using most Unix-like operating systems. *dDocent* is largely dependent on other bioinformatics software packages, taking advantage of programs designed specifically for each task of the analysis and ensuring that each modular component can be updated separately. Proper implementation depends on the correct installation of each third-party packages/tools. A full list of dependencies can be found in the user manual at (http://ddocent.wordpress.com/ddocent-pipeline-user-guide/) and a sample script to automatically download and install the packages in a Linux environment can be found at the *dDocent* repository (https://github.com/jpuritz/dDocent).

dDocent is run by simply switching to a directory containing the input data and starting the program. There is no configuration file; *dDocent* will proceed through a short series of command-line prompts, allowing the user to set up analysis parameters. After all required variables are configured, including an e-mail address for a completion notification, *dDocent* provides instructions on how to move the program to the background and run, undisturbed, until completion. The pipeline is designed to take advantage of multiple processing core machines and, whenever possible, processes should be invoked with multiple threads or occurrences. For most Linux distributions, the number of processing cores should be automatically detected. If *dDocent* cannot determine the number of processors, it will ask the user to input the value.

There are two distinct modules of *dDocent*: dDocent.FB and dDocent.GATK. dDocent.FB uses minimal, BAM-file preparation steps before calling SNPs and INDELs, simultaneously using FreeBayes (Garrison & Marth, 2012). dDocent.GATK uses GATK (McKenna et al., 2010) for INDEL realignment, SNP and INDEL genotyping (using HaplotypeCaller), and variant quality-score recalibration, largely following GATK Best Practices recommendations (Auwera & Carneiro, 2013;DePristo et al., 2011). The modules represent two different strategies for SNP/INDEL calling that are completely independent of one another. The remainder of this paper focuses on dDocent.FB; additional information on dDocent.GATK may be found in the user guide and results from dDocent.GATK can be found in Appendix S1.

Data input requirements

dDocent requires demultiplexed forward and paired-end FASTQ files for every individual in the analysis. A simple naming convention (a single-word locality code/name

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014

and a single-word sample identifier separated by an underscore) must be followed for every sample; examples are *LOCA_IND01.F.fq* and *LOCA_IND01.R.fq*. A sample script for using a text file with barcodes and sample names and *process_radtags* from *Stacks* (Catchen et al., 2013) to properly demultiplex samples and put them in the proper *dDocent* naming convention can be found at the *dDocent* repository (https://github.com/jpuritz/dDocent).

Quality trimming

After *dDocent* checks that it is recognizing the proper number of samples in the current directory, it asks the user if s/he wishes to proceed with quality trimming of sequence data. If directed, *dDocent can* use the program *Trim Galore!* (http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/) to simultaneously remove Illumina adapter sequences and trim ends of reads of low quality. By default, *Trim Galore!* looks for double-digest RAD adapters (Peterson et al., 2012) and trims bases with quality scores less than Phred 10. Typically, quality trimming only needs to be performed once on data, so the option exists to skip this step in subsequent *dDocent* analyses.

De novo assembly

Without reference material, population genomic analyses from RADseq depend on *de novo* assembly of a set of reference contigs. Inherently, not all RAD loci appear in all individuals due to stochastic processes inherent in library preparation and sequencing and to polymorphism in restriction-enzyme restriction sites (Catchen et al., 2011). Moreover, populations can contain large levels of within locus polymorphism, making generation of a reference sequence computationally difficult. *dDocent* minimizes the amount of data used for assembly by taking advantage of the fact that flRAD loci present in multiple individuals should have higher levels of exactly matching reads (forward and reverse) than loci that are only present in a few individuals. Caution is advised for unique reads with low levels of coverage throughout the data set as they likely represent sequencing errors or polymorphisms that are shared only by a few individuals.

During assembly, paired-end reads are reverse complemented and concatenated to forward reads. Unique paired reads are identified and their occurrences are counted in the entire data set. These data are tabulated into the number of unique reads per levels of 1X to 50X coverage; a graph is then generated and printed to the terminal. The distribution usually follows an asymptotic relationship (Figure 1), with a large proportion of reads only having one or two occurrences, meaning they likely will not be informative on a population scale. Highly polymorphic RAD loci still should have at least one allele present at the level of expected sequence coverage, so this can be used as a guide for informative data. The user chooses a cut-off level of coverage for reads to be used for assembly – note all reads are still used for subsequence steps of the pipeline.

After a cut-off level is chosen, remaining reads are returned in forward- and reverse-read files and then input directly into the RADseq assembly program *Rainbow* (Chong et al., 2012). The default parameters of *Rainbow* are used except that the maximum number of mismatches used in initial clustering should be changed from four to six. In short, *Rainbow* clusters forward reads based on similarity; clusters are then recursively divided, based on reverse reads, into groups representing single alleles. Reads in merged clusters are then assembled using a greedy algorithm (Pop & Salzberg, 2008). *dDocent* then selects the longest contig for each cluster as the representative reference sequence for that RAD locus. If the forward read does not overlap with the reverse read (almost always the case with flRAD), the forward read is concatenated to the reverse read with ten ‘N’ characters as padding. Finally, reference sequences are clustered based on overall sequence similarity (chosen by user, 90% by default), using the program *CD-HIT* (Fu et al., 2012;Li & Godzik, 2006). This final cluster step reduces the data set further, based on overall sequence identity after assembly. Alternatively, *de novo* assembly can be skipped and the user can provide a FASTA file with reference sequences.

Read mapping

dDocent uses the MEM algorithm (Li, 2013) of *BWA* (Li & Durbin, 2009, 2010) to map quality-trimmed reads to the reference contigs. Users can deploy the default values of BWA or set an alternative value for each mapping parameter (match score, mismatch score, and gap-opening penalty). The default settings are meant for mapping reads to the human genome, so users are encouraged to experiment with mapping parameters. BWA output is ported to SAMtools (Li et al., 2009), saving disk space, and alignments are saved to the disk as binary alignment/Map (BAM). BAM files are then sorted and indexed.

SNP and INDEL discovery and genotyping

dDocent uses a two-step process to optimize the computationally intensive task of SNP/INDEL calling. First, quality-trimmed forward and reverse reads are reduced to unique reads. This data set is then mapped to all reference sequences using the previously entered mapping settings (see *Read Mapping* above). From this alignment, a set of intervals is created using BEDtools (Quinlan & Hall, 2010). The interval set saves computational time by directing the SNP-/INDEL-calling software to examine only reference sequences along contigs that have high quality mappings. Second, the interval list is then split into a single file for each processing core, allowing SNP/INDEL calling to be optimized with a scatter-gather technique. The program *FreeBayes* (Garrison & Marth, 2012) is then executed multiple times simultaneously (one execution per processor and genomic interval). *FreeBayes* is a Bayesian-based, variant-detection software that uses assembled haplotype sequences to simultaneously call SNPs, INDELS, multi-nucleotide polymorphisms (MNPs), and complex events (e.g., composite insertion and substitution events) from alignment files; *FreeBayes* has the added benefit for population genomics of using reads across multiple individuals to improve genotyping (Garrison & Marth, 2012). *FreeBayes* is run with minimal changes to the default parameters; minimum mapping quality score and base quality score are set to PHRED 10. After all executions of *FreeBayes* are completed, raw SNP/INDEL calls are concatenated into a single variant call file (VCF), using VCFtools (Danecek et al., 2011).

Variant Filtering

Final SNP data-set requirements are likely to be highly dependent on specific goals and aims of individual projects. To that end, *dDocent* uses *VCFtools* (Danecek et al., 2011) to provide only basic level filtering, mostly for run diagnostic purposes. d*Docent* produces a final VCF file that contains all SNPs, INDELS, MNPs, and complex events that are called in 90% of all individuals, with a minimum quality score of 30. Users are encouraged to use VCFtools and vcflib (part of the *FreeBayes* package; https://github.com/ekg/vcflib) to fully explore and filter data appropriately.

Comparison between dDocent and Stacks

Two sample localities, each comprised of 20 individuals, were chosen randomly from unpublished RADseq data sets of three different, marine fish species: red snapper (*Lutjanus campechanus*), red drum (*Sciaenops ocellatus*), and silk snapper (*Lutjanus vivanus*). These three species are part of ongoing RADseq projects in our laboratory, and preliminary analyses indicated high levels of nucleotide polymorphisms across all populations. Double-digest

RAD libraries were prepared, generally following Peterson *et al.* (2012). Individual DNA extractions were digested with *Eco*RI and M*sp*I. A barcoded adapter was ligated to the *Eco*RI site of each fragment and a generic adapter was ligated to the *Msp*I site. Samples were then equimollarly pooled and size-selected between 350 and 400 bp, using a Qiagen Gel Extraction Kit. Final library enhancement was completed using 12 cycles of PCR, simultaneously enhancing properly ligated fragments and adding an Illumina Index for additional barcoding. Libraries were sequenced on three separate lanes of an Illumina HiSeq 2000 at the University of Texas Genomic Sequencing and Analysis Facility.

Demultiplexed individual reads were analyzed with *dDocent*, using three different levels of final reference contig clustering (90%, 96%, and 99% similarity) in an attempt to alter the most comparable analysis variable in *dDocent* to match analysis variables of *Stacks*. The coverage cut-off for assembly was 12 for red snapper, 13 for red drum, and nine for silk snapper. All *dDocent* runs used mapping variables of one, three, and five for match-score value, mismatch score, and gap-opening penalty, respectively. For comparisons, complex variants were decomposed into canonical SNP and INDEL representation from the raw VCF files, using *vcfallelicprimitives* from *vcflib* (https://github.com/ekg/vcflib).

For *Stacks*, reads were demultiplexed and cleaned using *process_radtags*, removing reads with 'N' calls and low-quality base scores. Because *dDocent* inherently uses both reads for SNP/INDEL genotyping, forward reads and reverse reads were processed separately with *denovo_map.pl* (*Stacks* version 1.08), using three different sets of parameters. The first set had a minimum depth of coverage of two to create a stack, a maximum distance of two between stacks, and a maximum distance of four between stacks from different individuals, with both the deleveraging algorithm and removal algorithms enabled. The second set had a minimum depth of coverage of three to create a stack, a maximum distance of four between stacks, and a maximum distance of eight between stacks from different individuals, with both the deleveraging algorithm and removal algorithms enabled. The third set had a minimum depth of coverage of three to create a stack, a maximum distance of four between stacks, and a maximum distance of 10 between stacks from different individuals, with both the deleveraging algorithm and removal algorithms enabled. SNP calls were output in VCF format.

For both *dDocent* and *Stacks* runs, VCFtools was used to filter out INDELs and SNPs that had a minor allele count of less than five. SNP calls were then evaluated at different individual-coverage levels: the total number of SNPs; the number of SNPS called in 75%, 90%, and 99% of individuals at 3X coverage; the number of SNPS called in 75% and 90% of individuals at 5X coverage; the number of SNPS called in 75% and 90% of individuals at 10X coverage; and the number of SNPS called in 75% and 90% of individuals at 20X coverage. Overall coverage levels for red snapper were lower and likely impacted by a few low-quality individuals; consequently, the number of 5X and 10X SNPs shared among 90% of individuals (after removing the bottom 10% of individuals in terms of coverage) were compared instead of SNP loci shared at 20X coverage. Results from two runs of *Stacks* (one using forward and one using reverse reads) were combined for comparison with *dDocent*, which inherently calls SNPs on both reads. All analyses and computations were performed on a 32-core Linux workstation with 128 GB of RAM.

RESULTS AND DISCUSSION

Results of SNP calling, including run times (in minutes) for each analysis (not including quality trimming), are presented in Table 1. Data from high coverage SNP calls, averaged over all runs for each pipeline, are presented in Figure 1. While *Stacks* called a larger number of low coverage SNPs, limiting results to higher individual coverage and to higher individual call rates revealed that *dDocent* consistently called more high-quality SNPs. Run times were equivalent for both pipelines.

At almost all levels of coverage in three different data sets, *dDocent* called more SNPs across more individuals than *Stacks*. Two key differences between *dDocent* and *Stacks* likely contribute these discrepancies: (i) quality trimming instead of quality filtering, and (ii) simultaneous use of forward and reverse reads by *dDocent* in assembly, mapping, and genotyping, instead of clustering as employed by *Stacks*. As with any data analysis, quality of data output is directly linked to the quality of data input. Both *dDocent* and *Stacks* use procedures to ensure that only high-quality sequence data are retained; however, *Stacks* removes an entire read when a sliding window of bases drops below a preset quality score (PHRED 10, by default), while *dDocent* via *Trim Galore!* trims off low-quality bases, preserving high-quality bases of each read. Filtering instead of trimming results in fewer reads entering the *Stacks* analysis (between 65%-95% of the data compared to *dDocent*; data not shown), generating lower levels of coverage and fewer SNP calls than *dDocent*.

dDocent offers two advantages over *Stacks*: (i) it is specifically designed for paired-end data and utilizes both forward and reverse reads for *de novo* RAD loci assembly, read mapping, variant discovery, and genotyping; and (ii) it aligns reads to reference sequence instead of clustering by identity. Using both reads to cluster and assemble RAD loci helps to ensure that portions of the genome with complex mutational events, including INDELs or small repetitive regions, are properly assembled and clustered as homologous loci. Additionally, using *BWA* to map reads to reference loci enables *dDocent* to properly align reads with INDEL polymorphisms, increasing coverage and subsequent variant discovery and genotyping. Clustering methods employed by *Stacks*, whether clustering alleles within an individual or clustering loci between individuals, effectively remove reads, alleles, and loci with INDEL polymorphisms because the associated frame shift effectively inflates the observed number of base-pair differences. For organisms with large effective population sizes and high levels of genetic diversity, such as many marine organisms (Waples, 1998;Ward et al., 1994), removing reads and loci with INDEL polymorphisms will result in a loss of shared loci and coverage.

CONCLUSION

dDocent is an open-source, freely available population genomics pipeline configured for species with high levels of nucleotide and INDEL polymorphisms, such as many marine organisms. The *dDocent* pipeline reports more SNPs shared across greater numbers of individuals and with higher levels of coverage than current alternatives. The pipeline and a comprehensive online manual can be found at (http://dDocent.wordpress.com) and (https://github.com/jpuritz/dDocent).

ACKNOWLEDGEMENT

We thank T. Krabbenhoft for beta testing and use of his data, and C. Bird and D. Portnoy for useful discussions and comments on the manuscript. Work was supported by the National Marine Fisheries Service under Saltonstall-Kennedy Award NA10NMF4270199, MARFIN Award NA12NMF4330093, and Cooperative Research Program Award NA12NMF4540082, and by an award from the NOAA Sea Grant Aquaculture Program.

REFERENCES

Auwera G, Carneiro M. 2013. From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline. *Current Protocols in Bioinformatics*: 1–33.

Baird NA, Etter PD, Atwood TS, Currey MC, Shiver AL, Lewis ZA, Selker EU, Cresko WA, Johnson EA. 2008. Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PloS ONE* 3: e3376.

Catchen JM, Amores A, Hohenlohe P, Cresko W, Postlethwait JH. 2011. Stacks: building and genotyping Loci de novo from short-read sequences. *G3 (Bethesda, Md.)* 1: 171–182.

Catchen J, Hohenlohe PA, Bassham S, Amores A, Cresko WA. 2013. Stacks: an analysis tool set for population genomics. *Molecular ecology* 22: 3124–3140.

Chong Z, Ruan J, Wu C. 2012. Rainbow : an integrated tool for efficient clustering and assembling RAD-seq reads. : 1–6.

Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R. 2011. The variant call format and VCFtools. *Bioinformatics (Oxford, England)* 27: 2156–2158.

DePristo MA, Banks E, Poplin R, Garimella K V, Maguire JR, Hartl C, Philippakis AA, Angel G del, Rivas MA, Hanna M, McKenna A, Fennell TJ, Kernytsky AM, Sivachenko AY, Cibulskis K, Gabriel SB, Altshuler D, Daly MJ. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature genetics* 43: 491–498.

Ellegren H. 2014. Genome sequencing and population genomics in non-model organisms. *Trends in ecology & evolution* 29: 51–63.

Fu L, Niu B, Zhu Z, Wu S, Li W. 2012. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics (Oxford, England)* 28: 3150–3152.

Garrison E, Marth G. 2012. Haplotype-based variant detection from short-read sequencing. : 1–9.

Guo B, Zou M, Wagner A. 2012. Pervasive indels and their evolutionary dynamics after the fish-specific genome duplication. *Molecular biology and evolution* 29: 3005–3022.

Hohenlohe PA, Bassham S, Etter PD, Stiffler N, Johnson EA, Cresko WA. 2010. Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. *PLoS genetics* 6: e1000862.

Keever CC, Sunday J, Puritz JB, Addison JA, Toonen RJ, Grosberg RK, Hart MW. 2009. Discordant distribution of populations and genetic variation in a sea star with high dispersal potential. *Evolution; international journal of organic evolution* 63: 3214–3227.

Li H, Durbin R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)* 25: 1754–1760.

Li H, Durbin R. 2010. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)* 26: 589–595.

Li W, Godzik A. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics (Oxford, England)* 22: 1658–1659.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)* 25: 2078–2079.

Li H. 2013. Aligning sequence reads , clone sequences and assembly contigs with BWA-MEM. 00: 1–3.

Mardis ER. 2008. Next-generation DNA sequencing methods. *Annual review of genomics and human genetics* 9: 387–402.

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, DePristo MA. 2010. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research* 20: 1297–1303.

Narum SR, Buerkle CA, Davey JW, Miller MR, Hohenlohe PA. 2013. Genotyping-by-sequencing in ecological and conservation genomics. *Molecular ecology* 22: 2841–2847.

Peterson BK, Weber JN, Kay EH, Fisher HS, Hoekstra HE. 2012. Double digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PloS one* 7: e37135.

Pop M, Salzberg S. 2008. Bioinformatics challenges of new sequencing technology. *Trends in Genetics* 24: 142–149.

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics (Oxford, England)* 26: 841–842.

Rowe HC, Renaut S, Guggisberg A. 2011. RAD in the realm of next-generation sequencing technologies. *Molecular ecology* 20: 3499–3502.

Sodergren E, Weinstock GM, Davidson EH, Cameron RA, Gibbs RA, Angerer RC, Angerer LM, Arnone MI, Burgess DR, Burke RD, Coffman JA, Dean M, Elphick MR, Ettensohn

CA, Foltz KR, Hamdoun A, Hynes RO, Klein WH, Marzluff W, et al. 2006. The genome of the sea urchin Strongylocentrotus purpuratus. *Science (New York, N.Y.)* 314: 941–952.

Toonen RJ, Puritz JB, Forsman ZH, Whitney JL, Fernandez-Silva I, Andrews KR, Bird CE. 2013. ezRAD: a simplified method for genomic genotyping in non-model organisms. *PeerJ* 1: e203.

Wang S, Meyer E, McKay JK, Matz M V. 2012. 2b-RAD: a simple and flexible method for genome-wide genotyping. *Nature methods* 9: 808–810.

Waples RS. 1998. Separating the wheat from the chaff: patterns of genetic differentiation in high gene flow species. *Journal of Heredity* 89: 438–450.

Ward RD, Woodwark M, Skibinski DOF. 1994. A comparison of genetic diversity levels in marine, freshwater, and anadromous fishes. *Journal of Fish Biology* 44: 213–232.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014

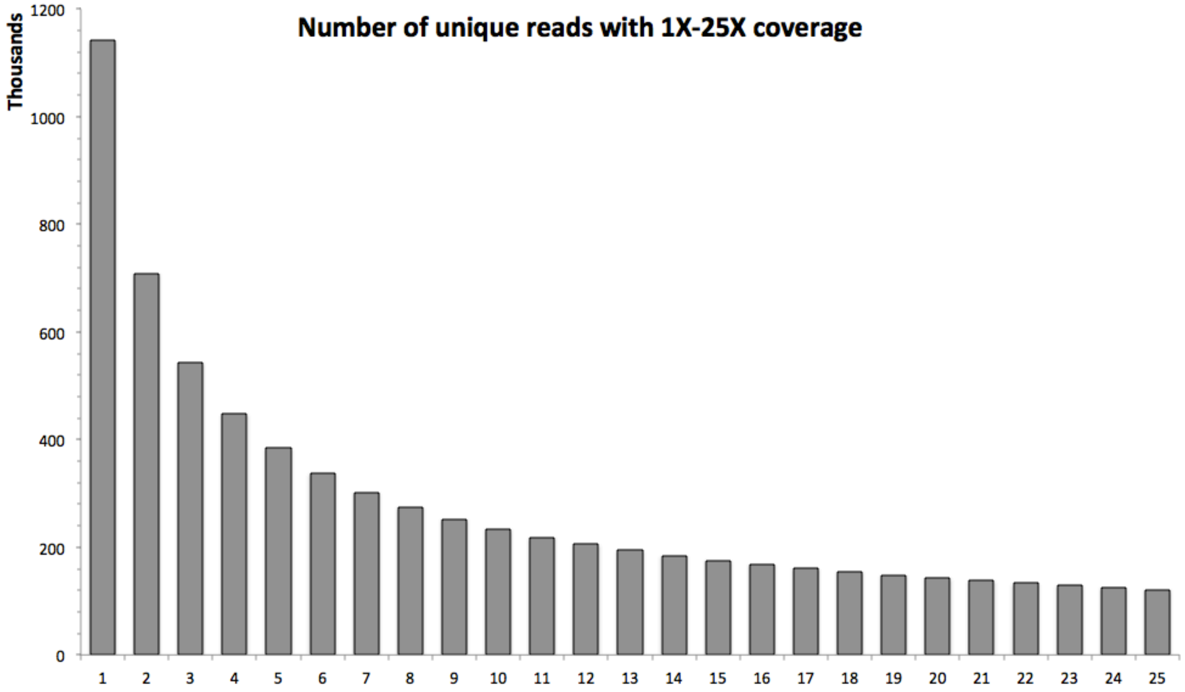
Table 1. Results from individual runs of *dDocent* and *Stacks. dDocent* runs varied in the level of similarity used to cluster reference sequences: A (90%), B (96%), and C (99%). For *Stacks*, forward reads and reverse reads were separately processed with *denovo_map.pl* (*Stacks* version 1.08), using three different sets of parameters: A, minimum depth of coverage of two to create a stack, a maximum distance of two between stacks, and a maximum distance of four between stacks from different individuals; B, minimum depth of coverage of three to create a stack, a maximum distance of four between stacks, and a maximum distance of eight between stacks from different individuals; and C, minimum depth of coverage of three to create a stack, a maximum distance of four between stacks, and a maximum distance of 10 between stacks from different individuals. SNP calls were evaluated at different individual coverage levels: (i) total number of SNPs; (ii) number of SNPS called in 75%, 90%, and 99% at 3X coverage; (iii) number of SNPS called in 75% and 90% of individuals at 5X coverage; (iv) number of SNPS called in 75% and 90% of individuals at 10X coverage; and, (v) number of SNPS called in 75% and 90% of individuals at 20X coverage. Results from forward and reverse reads of *Stacks* were combined for comparison with *dDocent* , which inherently calls SNPs on both reads.

	dDocent A	*dDocent* B	*dDocent* C	*Stacks* A	*Stacks* B	*Stacks* C
	Red snapper					
Total 3X SNPS	30,130	30,043	29,907	28,817	33,479	34,459
75% 3X SNPs	12,507	12,249	12,012	4,150	5,735	5,728
90% 3X SNPs	5,368	5,187	5,039	675	987	983
99% 3X SNPs	52	25	5	0	0	0
75% 5X SNPs	8,144	7,946	7,793	2,632	4,351	4,324
90% 5X SNPs	2,775	2,696	2,606	179	579	574

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014

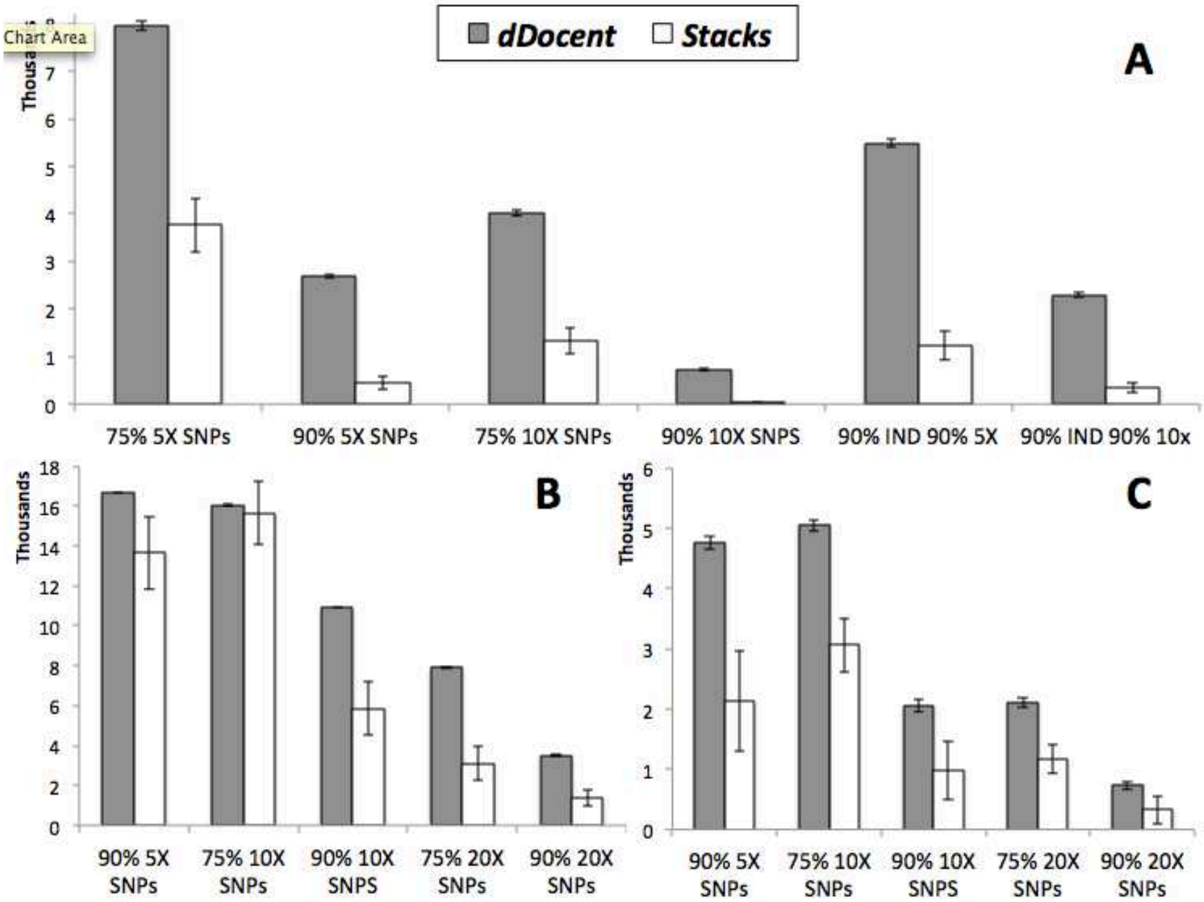
75% 10X SNPs	4,151	4,017	3,914	783	1,618	1,579
90% 10X SNPS	785	729	682	7	48	47
90% IND 90% 5X	5,625	5,499	5,332	806	1,807	1,079
90% IND 90% 10x	2,403	2,298	2,196	129	441	434
Run time	59	58	57	70	47	53
	Red drum					
Total 3X SNPS	27,263	27,329	27,295	45,792	50,821	52,366
75% 3X SNPs	23,339	23,328	23,226	24,134	28,991	28,981
90% 3X SNPs	20,764	20,704	20,586	13,439	17,946	17,874
99% 3X SNPs	7,121	7,022	6,937	828	1,264	1,259
75% 5X SNPs	20,015	20,009	19,946	21,021	26,526	26,464
90% 5X SNPs	16,739	16,680	16,588	10,494	15,282	15,207
75% 10X SNPs	16,078	16,042	15,970	12,928	17,018	16,983
90% 10X SNPS	10,988	10,942	10,842	4,159	6,734	6,705
75% 20X SNPs	7,975	7,933	7,824	2,276	3,538	3,516
90% 20X SNPs	3,534	3,512	3,455	243	1,974	1,961
Run time	55	55	53	58	55	65
	Silk snapper					
Total 3X SNPS	35,763	35,645	35,509	48,742	55,505	58,352
75% 3X SNPs	17,518	17,244	16,992	7,596	9,705	9,696
90% 3X SNPs	8,586	8,353	8,157	2,007	3,439	3,433
99% 3X SNPs	2,552	2,380	2,276	132	527	523
75% 5X SNPs	10,775	10,547	10,385	4,789	7,290	7,274
90% 5X SNPs	4,936	4,725	4,606	1,225	2,573	2,570
75% 10X SNPs	5,252	5,018	4,876	2,094	3,547	3,546
90% 10X SNPS	2,191	2,058	1,938	489	1,224	1,223
75% 20X SNPs	2,220	2,098	1,984	703	1,415	1,411
90% 20X SNPs	801	721	675	136	417	418
Run time	98	100	60	93	89	204

Figure 1. Levels of coverage for each unique read in the red snapper data set. The horizontal axis represents the minimal level of coverage and the vertical axis represents the number of unique paired reads in thousands.



PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014

Figure 2. SNP results averaged across the three different run parameters for *dDocent* and *Stacks*. (A) Red snapper, (B) Red drum, (C) Silk snapper (see Methods or Table 1 for SNP categories description). Error bars represent standard error.



PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.314v1 | CC-BY 4.0 Open Access | received: 28 Mar 2014, published: 28 Mar 2014