Upload image

Jun 23, 2020

Reference-independent analysis of RADseq data from a single sample

Bradley Till[1,2], Claudia Osorio[3], Rachel Howard-Till[1,4]

[1]Centro de Genómica Nutricional Agroacuícola; [2]https://orcid.org/0000-0002-1300-8285;
[3]Instituto de Investigaciones Agropecuarias; [4]https://orcid.org/0000-0003-2568-7980

1 Works for me dx.doi.org/10.17504/protocols.io.bdtji6kn

Bradley Till
Centro de Genómica Nutricional Agroacuícola; https://orcid.o...

ABSTRACT

This protocol describes the use of freely available software for the reference-independent analysis of RADseq data produced from a single sample. It is useful to perform a pilot experiment on a single sample for species lacking a reference genome sequence in order to facilitate proper experimental design for a larger-scale experiment. The example data used in this protocol is from a wild Berberis darwinii plant growing in Chile. The protocol produces data that provides an estimation of the number of unique base pairs recovered from a size selection of restriction endonuclease digested genomic DNA, and also an estimation of the frequency of heterozygous polymorphisms in the plant. Evaluation of sequence quality also provides a validation of the methodologies used for tissue collection, DNA extraction, restriction digestion, and size selection (see https://dx.doi.org/10.17504/protocols.io.bdg9i3z6). In addition, the discovered heterozyygous polymorphisms serve as markers to evaluate natural propagation (e.g. self-fertilization versus outcrossing versus clonal propagation) and also for use in tracking the success of human-directed crosses for breeding.

DOI

dx.doi.org/10.17504/protocols.io.bdtji6kn

PROTOCOL CITATION

Bradley Till, Claudia Osorio, Rachel Howard-Till 2020. Reference-independent analysis of RADseq data from a single sample. **protocols.io**
dx.doi.org/10.17504/protocols.io.bdtji6kn

KEYWORDS

RADseq, reference-independent, GBS, Berberis darwinii, RESCAN, Stacks, BBMap

LICENSE

CREATED

Mar 17, 2020

LAST MODIFIED

Jun 23, 2020

PROTOCOL INTEGER ID

34379

GUIDELINES

All data analysis steps described in this protocol were carried out using a 64-core 128 Gb RAM machine running Linux Debian. Pre-processing steps were tested on the Linux Ubuntu operating system (18.04 LTS) (https://ubuntu.com/download/desktop) on an quad-core desktop with 8 Gb RAM. When using this machine, step 3.3 (ustacks) failed due to insufficient memory. If you are new to bioinformatics and command-line tools, it is advised to test the tools first on a Linux PC prior to moving to a shared server.

MATERIALS TEXT

Raw fastq example data used to prepare this protocol can be found at https://www.ncbi.nlm.nih.gov/sra/?term=PRJNA613471. The software needed for this protocol are: RestrictionDigest, FastQC, Stacks, BBmap, and a text editor. Further details are provided in the main text.

BEFORE STARTING

Download, install and test all software described in this protocol.

1 Prepare genomic DNA for sequencing.

The extraction of genomic DNA of suitable quality and quantity is essential for a successful experiment. We have prepared a detailed protocol for low-cost tissue collection, desiccation, genomic DNA extraction, and concentration that provides DNA suitable for next generation sequencing (https://dx.doi.org/10.17504/protocols.io.bdg9i3z6).

1.1 Choose experimental parameters for the single sample pilot experiment part 1: Select size ranges and the amount of input genomic DNA needed for NGS library preparation.

For reduced-representation genome sequencing, an experimental design is developed using available genome size and sequence data, and the desired sequencing chemistry and read-length, in order to choose restriction enzyme(s) and molecular weight range of DNA fragments to produce the desired number of unique base pairs. For example, there is no reference genome available for Berberis darwinii, but scaffold sequences are available for the closely related Berberis thunbergii (https://www.ncbi.nlm.nih.gov/assembly/GCA_003290165.1/). The Perl module RestrictionDigest (https://github.com/JINPENG-WANG/RestrictionDigest) can be used to model the percentage of the genome represented in a specific molecular weight range when digested with a particular restriction endonuclease. The assumption with this approach is that the two genomes have a similar GC content and produce a similar genomic fraction under the same digestion and size selection parameters.

RestrictionDigest 0.01
source by Jinpeng Wang

A. Download and install RestrictionDigest software following the instructions in the readme file.

B. Download the reference genome assembly of a closely related species (genomic fasta as .fa or .fna).

C. Place the genomic fasta file in the RestrictionDigest directory.

D. Edit the Perl script file to select your enzyme(s), genome and output directory. (see note below).

Edit the below text using Ubuntu Text Editor. Save the file with a unique name (in this example, bthunbergii_haeIII.pl) and place this file into the RestrictionDigest directory. Edit the third line for the path and the name of your genome fasta file (the example is a fasta file titled thunbergii.fna that is in the folder titled RestrictionDigest located in the home directory). Edit line 9 for your restriction nucleases used (this example uses HaeIII). Edit line 10 for the size range. Edit line 12 with the output directory (in this example the directory is titled HaeIII_thunbergii and is located in the RestrictionDigest folder). Copy all text below. Line 1 should contain text "use RestrictionDigest".

```
use RestrictionDigest;
my $single_digest=RestrictionDigest::SingleItem::Single->new();
$single_digest->add_ref(-reference=>"/home/bradleytill/RestrictionDigest/thunbergii.fna");
$single_digest->new_enzyme(-enzyme_name=>"EcoRI", -recognition_site =>"G|AATTC");
$single_digest->new_enzyme(-enzyme_name=>"AluI", -recognition_site =>"AG|CT");
$single_digest->new_enzyme(-enzyme_name=>"HaeIII", -recognition_site =>"GG|CC");
$single_digest->new_enzyme(-enzyme_name=>"NlaIII", -recognition_site =>"CAGT|");
$single_digest->new_enzyme(-enzyme_name=>"FatI", -recognition_site =>"|CATG");
$single_digest->add_single_enzyme(-enzyme =>"HaeIII");
$single_digest->change_range(-start =>350,-end =>700);
$single_digest->change_lengths_distribution_parameters(-front =>100,-behind =>1200,-step
=>50);
$single_digest->add_output_dir(-
output_dir=>"/home/bradleytill/RestrictionDigest/HaeIII_thunbergii");
$single_digest->single_digest();
$single_digest->frags_in_range_coverage_ratio();
$single_digest->all_frags_coverage_ratio();
```

E. Execute the Perl script by opening a terminal window and changing the directory (cd) so that you are within the RestrictionDigest directory, and typing "perl bthunbergii_haeIII.pl" without the quotation marks and changing the name of the .pl file accordingly (in the command below the file is called yourperlscript.pl).

RestrictionDigest

```
perl yourperlscript.pl
```

Perl script for in silico evaluation of length ranges and genome fractions of DNA digested with one or two defined restriction endonucleases.

F. Multiple files will be produced. Open the file starting with "digestion_summary..."

G. Review the length ranges and fragment ranges, and choose a suitable range and starting DNA concentration for the experiment. For example, when digested with HaeIII, it is expected that a 350 to 700 bp range would represent 0.179 of the B. thunbergii genome. The assumption is made that the GC content between B. thunbergii and B. darwinii are similar, and that 350 to 700 bp fragments of the B. darwinii genome digested with Hae III will represent a genome fraction similar to 0.179. This fraction is used to determine the starting amount of genomic DNA for the restriction digestion. For example, if 200 nanograms of starting material is required for library preparation, a minimum of 200 ng /0.179 = 1.1 micrograms of starting genomic DNA is required. A larger starting amount is advised as genome fraction estimations may vary from the true fraction, and the recovery of the selected DNA molecular weight range from an agarose gel will be less than 100%.

1.2 Choose experimental parameters for the single sample pilot experiment part 2: Estimate the number of unique base pairs that will be sequenced based on the range of fragments selected, the library preparation method and the sequencing method, and determine required sequencing throughput.

Citation: Bradley Till, Claudia Osorio, Rachel Howard-Till (06/23/2020). Reference-independent analysis of RADseq data from a single sample. https://dx.doi.org/10.17504/protocols.io.bdtji6kn

For example, literature searches suggest Berberis darwinii has an expected genome size of approximately 1.5 Gb. Therefore, sequencing of the entire isolated molecular weight range of 350-700 bps would produce about (1.5 x 0.179) x 268 Mbps of unique sequence. For the pilot experiment a 2x150PE sequencing approach was chosen with a library preparation utilizing 200 ng of starting material that included a size enrichment for fragments of 350 bps. Library preparation was outsourced to a sequencing provider and resulted in 75% of DNA fragments having a 395 bp or lower insert size. This was used to adjust the rough estimation of the number of unique base pairs to decide the minimal sequencing throughput to purchase for a successful experiment. For example, in our experiment, the fraction of the genome from 350 - 400 bp, plus an additional 25% to compensate for the fragments falling outside of this range, is calculated at 83.75 Mbs. Using this estimation, 1 Gb of raw sequencing data was purchased as it should provide a maximum of 11.9x coverage for a single sample. For comparison, if all 268 Mbps were sequenced, 1Gb of sequencing data would produce an average coverage of 3.73x. Actual coverage was expected to fall within the range of 11.9 and 3.73x, and therefore 1 Gb was considered suitable to evaluate SNP discovery.

1.3 Isolate selected size ranges of genomic DNA and sequence.

The wet bench methods for genomic DNA isolation and concentration suitable for restriction digestion can be found here: https://dx.doi.org/10.17504/protocols.io.bdg9i3z6. DNA was isolated from an agarose gel to produce the data described in this protocol. Library preparation, DNA sequencing, de-multiplexing (if required) and trimming is carried out by the DNA sequencing service provider.

2 Evaluate fastq data.

Most sequencing service providers will deliver raw sequencing data in a compressed fastq format (fastq.gz or fq.gz). If you have performed a paired-end sequencing, you will get two files, one for each read. It is useful to evaluate the quality if this data prior to proceeding further. The pipeline used in this protocol is also simplified if each read file from a single sample is combined (interleaved) after pre-processing of the data.

2.1 Evaluate fastq data with FastQC.

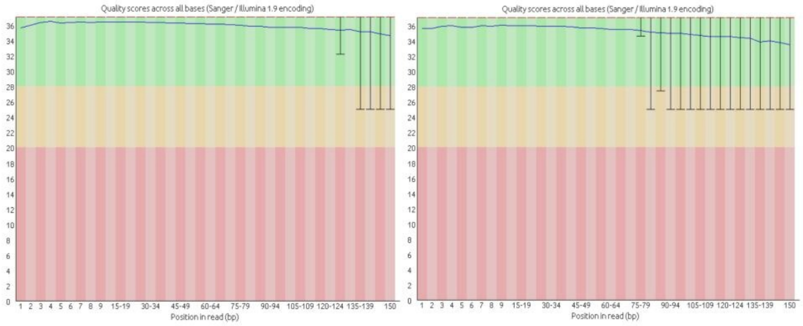
FastQC 0.11.9
by Simon Andrews

00:05:00 8 Gb RAM, quad-core processor

Follow instructions for installing and using the graphical user interface (GUI) software. Sequence quality scores should match expectations for the sequencing chemistry and read length used. Note that some metrics will produce warnings when evaluating RADseq type data (e.g. over-represented sequences).

Figure shows FastQC data for quality scores across all bases for read 1(left) and read 2 (right) of the 2x150PE run.

Citation: Bradley Till, Claudia Osorio, Rachel Howard-Till (06/23/2020). Reference-independent analysis of RADseq data from a single sample. https://dx.doi.org/10.17504/protocols.io.bdtji6kn



FastQC output of quality scores across all bases for read 1(left) and read 2 (right) of the 2x150PE run.

3 Perform the reference-independent Stacks pipeline on a single sample.

For the single Berberis sample example described in this protocol, the Stacks pipeline was chosen. Stacks was specifically developed to work with sequence data from restriction enzyme digested DNA to build genetic maps and perform population genomic analysis (see http://catchenlab.life.illinois.edu/stacks/). For the single Berberis sample, each step in the pipeline was executed independently.

Download and install Stacks following the instructions (http://catchenlab.life.illinois.edu/stacks/manual/).

Stacks 2.52

by Julian Catchen, Nicolas Rochette, Angel Amores, Paul Hohenlohe, Bill Cresko

If running Ubuntu, you may need to install a C++ compiler and zlib before it is possible to install Stacks.

Installing C++ compiler on Ubuntu.

```
sudo apt install gobjc++
```

Ubuntu 18.04

Install zlib on Ubuntu.

```
sudo apt-get install zlib1g-dev
```

Ubuntu 18.04

Citation: Bradley Till, Claudia Osorio, Rachel Howard-Till (06/23/2020). Reference-independent analysis of RADseq data from a single sample. https://dx.doi.org/10.17504/protocols.io.bdtji6kn

3.1 Process the fastq files for Stacks.

Move the fq.gz files into the Stacks directory. Create an output directory within the Stacks directory (here called radtagsoutputdirectory). Open a terminal window and enter (cd) into the Stacks directory. Make sure to define the restriction enzyme used (in the protocol example it is haeIII). This program includes a check for the presence of the enzyme cut site and provides a filtration based on quality. If you are using the test data, replace read1.fq.gz and read2.fq.gz with W2_L1_1.fq.gz and W2_L1_2.fq.gz, respectively.

Process radtags for Stacks.

```
process_radtags -1  read1.fq.gz -2 read2.fq.gz  -o ./radtagsoutputdirectory -c
-q -r -e haeIII
```

3.2 Interleave the two fastq files from a paired end read of a single sample.

BBmap 38.79
source by Brian Bushnell at the Joint Genome Institute

Follow the installation guide at https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/installation-guide/. Place the processed fastq.gz files from step 3.1 into the bbmap directory (do not take the files with "rem" in the title; you should move only two files for a single sample paired end read). Enter (cd) the bbmap directory and then execute the reformat software. (note if you are using the example data, read1.fastq.gz is W2_L1_1.1.fa.gz and read2.fastq.gz is W2_L1_L2.2.fq.gz).

00:05:00 Performed on 8 Gb RAM quad-core desktop

Interleave fastq files.

```
./reformat.sh in1=read1.fastq.gz in2=read2.fastq.gz out=interleaved.fq.gz
```

It is useful to choose a short and descriptive name for your output file. For the Berberis example described in this protocol, we chose W2pi.fq.gz. W2 for the unique plant and sample name, pi for processed radtags & interleaved.

3.3 Create loci with ustacks.

Move the interleaved.fq.gz file (titled W2pi.fq.gz in the example) to the Stacks directory. Create an output directory within the Stacks directory titled stacksout. Open a terminal window and move (cd) to the Stacks directory. Then execute ustacks.

ustacks

```
ustacks -f  W2pi.fq.gz -o ./stacksout -i 1 -p 64
```

Note that -p defines the number of threads. Check the number of processors on your computer and adjust accordingly.

3.4 Create a popmap file and run cstacks to create a catalog of loci.

For cstacks to run, a population map file must be prepared. This is a plain text file. For the single Berberis sample, this was a single row, two column tab delimited text file in the following format:

W2pi 1

This can be created using Text Edit in Ubuntu. It needs to be a plain text tab delimited file and may not work properly if it contains special characters that are inserted by some word processing programs.

This file is saved with the title popmap and placed in the Stacks directory. This is followed by running cstacks from the command line while in the Stacks directory.

cstacks

```
cstacks -P ./stacksout -M ./popmap -n 4 -p 64
```

3.5 Search the consensus loci created with ustacks against the catalog from cstaks using sstacks. The following command is executed in the terminal window while in the Stacks directory.

sstacks

```
sstacks -P ./stacksout -M ./popmap -p 64
```

3.6 Create a BAM formatted alignment file using tsv2bam.

tsv2bam

```
tsv2bam -P ./ustacksout -M ./popmap/popmap -t 64
```

Note that the thread option changes from -p to -t in this command.

3.7 Identify SNPs with gstacks.

gstacks

```
gstacks -P ./ustacksout -M ./popmap/popmap -t 64
```

Citation: Bradley Till, Claudia Osorio, Rachel Howard-Till (06/23/2020). Reference-independent analysis of RADseq data from a single sample. https://dx.doi.org/10.17504/protocols.io.bdtji6kn

3.8 Create a standard Variant Call Format (VCF) file with the populations tool.

Create a VCF file using populations.

```
populations -P ./ustacksout --vcf -t 64
```

VCF files are compatible with many tools for the analysis of nucleotide variants from sequencing data.

3.9 Create a reference sequence in fasta format using the populations tool.

Create a fasta file using populations.

```
populations -P ./ustacksout --fasta-loci t 64
```

The fasta file contains the genomic sequence from the experiment in a format compatible with many bioinformatics tools. This file can be used with the fastq data to perform additional analyses.

4 Preliminary analysis of data.

The goal of evaluating a single sample is to calculate the number of unique base pairs produced from the chosen experimental design (restriction enzyme, size range, sequencing chemistry, and read length), determine read coverage, and to obtain an initial estimation of heterozygous nucleotide variation.

Results can be found in log files produced by the various Stacks commands. The gstacks.log file provides the number of loci and mean loci size that can be used to calculate the number of base pairs in the experiment. The file also provides mean coverage for the experiment. The populations.log file reports unique base pairs (this should be similar to that calculated from gstacks.log) and also heterozygous variation (all variants reported in the Berberis darwinii example are heterozygous because a single diploid sample was evaluated). This experiment produced 84 Mbps of unique sequence with a mean coverage post-filtering of 8.7x. More than 25,000 candidate SNP variants were also discovered.

Citation: Bradley Till, Claudia Osorio, Rachel Howard-Till (06/23/2020). Reference-independent analysis of RADseq data from a single sample. https://dx.doi.org/10.17504/protocols.io.bdtji6kn