# Chapter 5

# SNP Discovery Using Next Generation Transcriptomic Sequencing

## Pierre De Wit

## Abstract

In this chapter, I will guide the user through methods to find new SNP markers from expressed sequence (RNA-Seq) data, focusing on the sample preparation and also on the bioinformatic analyses needed to sort through the immense flood of data from high-throughput sequencing machines. The general steps included are as follows: sample preparation, sequencing, quality control of data, assembly, mapping, SNP discovery, filtering, validation. The first few steps are traditional laboratory protocols, whereas steps following the sequencing are of bioinformatic nature. The bioinformatics described herein are by no means exhaustive, rather they serve as one example of a simple way of analyzing high-throughput sequence data to find SNP markers. Ideally, one would like to run through this protocol several times with a new dataset, while varying software parameters slightly, in order to determine the robustness of the results. The final validation step, although not described in much detail here, is also quite critical as that will be the final test of the accuracy of the assumptions made in silico.

There is a plethora of downstream applications of a SNP dataset, not covered in this chapter. For an example of a more thorough protocol also including differential gene expression and functional enrichment analyses, BLAST annotation and downstream applications of SNP markers, a good starting point could be the "Simple Fool's Guide to population genomics via RNA-Seq," which is available at http://sfg.stanford.edu.

**Key words** RNA-Seq, SNP, Transcriptome assembly, Bioinformatics, Alignment, Population genomics, NGS, Illumina

## 1 Introduction

### *1.1 Historical Background: From Sanger to RNA-Seq*

Since the advent of DNA sequencing methods, and the discovery of genetic variation [1], there has been an interest in using this variation to understand evolutionary processes such as genetic drift, natural selection, and the formation of new species. Early on, gel electrophoretic markers such as AFLPs [2] and allozymes [3] provided some interesting insights into the genetic structures of populations. Later, the development of microsatellite markers further improved our understanding of neutral genetic variation in natural populations [4]. However, these markers are usually few

Sarah J. Bourlat (ed.), *Marine Genomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1452,
DOI 10.1007/978-1-4939-3774-5_5, 

and it cannot be known if they are representative of the genome as a whole. In addition, they are generally assumed to not be under any selection pressure [5]. Only recently, with the advent of high-throughput DNA sequencing methods, have we begun to gain insights into the genome-wide distribution of polymorphisms and the effects of natural selection on genome architecture.

### 1.2 Why Focus on the Transcriptome?

Even with the latest DNA sequencing technologies, putting together a genome sequence into full-length chromosomes from short read data is very difficult. The number of available genome sequences is ever increasing, but the list of well-assembled ("complete") genomes is to date still restricted to a few model taxa. Thus, it is many times desirable to focus on parts of the genome that contain the information of interest. There are many methods to do this, but they all fall within two categories: random and targeted methods. An example of a random method is RAD sequencing [6], in which the genomic DNA is fragmented using a restriction enzyme and regions flanking the restriction site are sequenced. These types of methods are useful for studying genome-wide distributions of genetic variation or for finding loci exhibiting interesting patterns. However, unless there is a well-annotated genome of the species of interest, it can be very difficult to gain an understanding of the function of the observed pattern. An example of a targeted method is RNA-Seq [7], whereby mature mRNAs are isolated and sequenced, usually with a poly-A binding method. While this method does not provide genome-wide observations, it focuses on the part of the genome that contains a large proportion of the functionally relevant information (how much is still an active debate, however). One might also argue that protein-coding sequences also have a larger chance of being affected by natural selection (both balancing and disruptive), while third codon positions and UTRs could be freer to evolve neutrally.

One very useful aspect of expressed sequence data is the relative ease of functional annotation due to the very conserved nature of protein evolution—by BLASTing to public databases one can in many cases gain an understanding of the function of an unknown sequence even in nonmodel systems where no genome sequence is available.

### 1.3 Issues with Transcriptomic SNPs

While characterizing the genetic variation present in and around protein coding regions allows for studies of natural selection and population genetics, there are some issues to keep in mind. First, the potential for background stabilizing selection can pose problems (even in UTRs and third codon positions linked to selected loci), as this process tends to disguise weak population structure [8]. Also, the assumptions of outlier analyses might be violated if most of the loci used in an analysis are under stabilizing selection [9]. Second, the very nature of mRNA can pose problems as there is great variation in transcript abundance, so in low-frequency transcripts it can be hard to separate sequencing errors from true SNPs [10]. In

addition, patterns of allele-specific expression (ASE) can bias allele frequency estimates on pooled samples [11], or even cause incorrect genotyping if the difference in expression between alleles is too high [12]. It can also be difficult to separate out different isoforms of the same transcript from transcripts from paralog genes [13].

## 2 Materials

### 2.1 RNA Extraction Using Phenol/Chloroform

1. Solution for RNA stabilization and storage or liquid nitrogen for tissue preservation.
2. 1.5 ml Eppendorf tubes.
3. Trizol.
4. Chloroform.
5. Ball bearing beads.
6. 100 % isopropanol.
7. High salt buffer: 0.8 M Na citrate and 1.2 M Na chloride.
8. 75 % ethanol.
9. 4 °C centrifuge.
10. 55 °C heat block.
11. Tissue lyser or vortex mixer.

### 2.2 cDNA Library Preparation Using Illumina's TruSeq RNA Sample Prep Kit

1. Illumina TruSeq RNA sample preparation kit.
2. Magnetic beads for DNA purification (also called SPRI beads for solid-phase reversible immobilization).
3. Magnetic 96-well plate.
4. Reverse transcriptase.
5. Agilent Bioanalyzer or TapeStation.
6. QuBit high-sensitivity DNA assay.

### 2.3 Sequencing

1. A sequencing facility with access to Illumina sequencing machines.

### 2.4 Bioinformatics

1. Computer (Mac/Linux) with software installed: fastx toolkit, trinity, bwa, samtools (or access to a remote server with this software installed). Custom-made Python and bash scripts, GATK v 2.5 and Picard MarkDuplicates available on GitHub at: (https://github.com/DeWitP/SFG/tree/master/scripts/).

### 2.5 Validation

1. Primer 3 software.
2. PCR reagents: Oligonucleotides, dNTPs, BSA, $MgCl_2$, Water, Taq polymerase.
3. A sequencing facility for Sanger sequencing.

## 3 Methods

### 3.1 RNA Extraction

All Trizol steps should be done in a fume hood.

1. Thaw tissue (should be flash-frozen at time of sampling and stored at −80 °C, alternatively stored in RNA stabilization solution at −20 °C) on ice.
2. Cut tissue into small pieces with a clean razor blade, blot with tissue paper, and place in a 1.5 ml Eppendorf tube (*see* **Note 1**).
3. Add ball bearing beads, then 1 ml Trizol (in fume hood). Shake in a Tissue lyser (or Vortex on high speed) for 2 min until tissue has been homogenized.
4. Incubate at room temperature for 5 min.
5. Spin for 10 min at 12,000 × *g*, 4 °C. Transfer liquid to clean tube.
6. Add 200 μl chloroform and shake vigorously for 15 s by hand.
7. Incubate for 2–3 min at room temperature.
8. Spin for 15 min at 12,000 × *g*, 4 °C, then transfer the top phase (RNA) to a clean tube. DNA and proteins are in the bottom phase and can be stored at −20 °C until validation of markers is required.
9. If contamination occurs (part of the inter- or bottom phase are transferred), add 100 μl chloroform, shake for 15 s by hand, then repeat **step 8**.
10. Add 250 μl 100% isopropanol and 250 μl high salt buffer, shake.
11. Precipitate at room temp for 5–10 min.
12. Spin for 10 min at 12,000 × *g*, 4 °C, then discard supernatant.
13. Wash pellet in 1 ml 75% ethanol. Spin for 5 min at 7500 × *g*, 4 °C.
14. Discard supernatant, air dry for 5–10 min (30 s on 55 °C heat block).
15. Resuspend in nuclease-free water (12 μl) and incubate for 10 min at 55–60 °C. 1 μl can be used for QuBit concentration measurement and to examine RNA integrity (*see* **Note 2**).
16. Flash freeze in liquid nitrogen and store at −80 °C overnight, or continue directly with Subheading 3.2.

### 3.2 cDNA Library Prep

1. Standardize the amount of starting material, usually about 1 μg of total RNA produces good results.
2. Follow exactly the manual of the TruSeq kit (*see* **Note 3**).
3. Determine the fragment size distributions in the samples with an Agilent Bioanalyzer or TapeStation.
4. Measure the DNA concentration using a QuBit high-sensitivity DNA assay (the TapeStation measurements are usually not accurate enough).

5. The molarity can then be calculated as follows:

   $\text{Molarity} = \text{Concentration (ng/ml)} / (0.66 \times \text{mean fragment length (bp)})$.

6. Pool the samples equimolarly by calculating the required volume of each sample required so that the number of moles in each sample is identical. Illumina sequencing machines typically require a pool DNA molarity of 2–10 nM. The final pool volume should ideally be at least 20 μl (*see* **Note 4**).

### 3.3 Sequencing

1. Choose a sequencing center (*see* **Note 5**).
2. Send samples on ice, providing the center with information on DNA concentration and fragment size distribution.

### 3.4 Data Download and QC

1. Make a safety backup of the data, and upload the data to the location where you will be doing the analyses. This can either be on your local computer if it has enough capacity or preferably on a remote computer cluster.
2. Once the data is located in the right place, we want to control the quality (*see* **Note 6**). In this chapter, we assume that you are working in your home folder and have your data located in a subfolder called "data" and the Python scripts in a subfolder called "scripts." If you change this, please adjust the following instructions accordingly.
3. Move into the "data" folder:

   ```
   cd ~/data
   ls
   ```

4. Execute the bash script TrimClip.sh (*see* **Note 7**) by typing:

   ```
   sh ../scripts/TrimClip.sh
   ```

   while in the folder containing your data. Make note of how many reads are being trimmed and clipped through the screen output.

5. Calculate the fraction of duplicate and singleton reads, using the bash script CollapseDuplicateCount.sh (*see* **Note 8**), by typing:

   ```
   sh ../scripts/CollapseDuplicateCount.sh
   ```

   while in the folder containing your data. Results will be located in text files named with your original file name with *.duplicate-count.txt* appended.

6. Summarize quality score and nucleotide distribution data, then plot, by typing:

   ```
   sh ../scripts/QualityStats.sh
   ```

   in order to summarize your data files. Then execute the plotting software by typing:

   ```
   sh ../scripts/Boxplots.sh
   ```

the software creates individual .png files for each sample, then combines them into one file called "Boxplots.pdf" (see Note 9).

### 3.5 Assembly (See Note 10)

1. Concatenate the sample files into one, using the cat command:
   ```
   cat *.trimmed.clipped.fastq > assembly_ready.fastq
   ```
2. Run Trinity to create a de novo assembly (*see* **Note 11**):
   ```
   Trinity.pl --seqType fq --JM 1G \
   --single assembly_ready.fastq --output assembly
   ```
3. Summarize the statistics of the assembly, using the count_fasta.pl script, by typing:
   ```
   ../scripts/count_fasta.pl ./assembly/Trinity.fasta \ > assembly/trinityStats.txt
   ```
4. Examine the statistics of the assembly (*see* **Notes 12** and **13**) by typing:
   ```
   nano assemblyTest/trinityStats.txt
   ```

### 3.6 Mapping (See Note 14)

1. Open the BWAaln.sh script in nano, by typing:
   ```
   nano ../scripts/BWAaln.sh
   ```
   The default parameters are currently set as:

   –n .01 –k 5 –l 30 –t 2

   You can change them to something else if you like (*see* **Note 15**).
2. Execute the BWAaln.sh script (*see* **Note 16**) by typing:
   ```
   sh ../scripts/BWAaln.sh
   ```
3. Convert your .sam files to .bam (*see* **Note 17**), sort and remove duplicate reads, by executing the script convert_to_bam_and_dedup.sh (*see* **Note 18**). Type:
   ```
   sh ../scripts/convert_to_bam_and_dedup.sh
   ```

### 3.7 SNP Detection and Filtering (See Notes 19 and 20)

1. Create a tab-delimited text file called rg.txt, which is located along with your data files. This file provides critical information for GATK to keep the individuals apart in the merged file (*see* **Note 21**). It should be formatted like this (new line for each sample):
   ```
   @RG    ID:READ_GROUP    SM:SAMPLE_NAME    PL:Illumina
   ```
2. Merge your deduplicated .bam files:
   ```
   samtools merge -h rg.txt merged.bam *dedup.bam
   ```

3. Index your merged .bam file so that GATK will be able to search through it:
   ```
   samtools index merged.bam
   ```
4. Realign around InDels using GATK, by typing (*see* **Note 22**):
   ```
   sh ../scripts/realigner.sh
   ```
5. Detect variant sites, using the script SNP-detection.sh, by typing (*see* **Note 23**):
   ```
   sh ../scripts/SNP_detection.sh
   ```
6. Recalibrate the SNPs, using the GATK VQSR algorithm, by typing (*see* **Note 24**):
   ```
   sh ../scripts/VQSR.sh (see Note 25)
   ```
7. Extract genotypes of all individuals at all variable sites from the .vcf file into a format useable by Microsoft Excel, using a genotype quality threshold, by typing (*see* **Note 26**):
   ```
   python ../scripts/getgenosfromvcf.py VQSR_
   PASS_SNPS.vcf \ Genotypes.txt rows 20
   ```
8. Use the bash command 'grep' to create a new file with only SNPs with high-quality genotypes for all samples:
   ```
   grep -v "\." Genotypes.txt > genotypes_
   shared_by_all.txt
   ```

### *3.8 In Silico Validation*

1. Test for deviations from Hardy–Weinberg equilibrium, especially for cases where all individuals are heterozygotes (*see* **Note 27**).
2. Another way is to use phase information to examine contigs for linkage disequilibrium—long linked stretches with fixed nucleotide differences could be signs of paralogous genes (but could also be a sign of a selective sweep).

### *3.9 Validation: Designing Primers, Sanger Sequencing (See Note 28)*

1. Design primers by copy-pasting your protein-coding DNA sequence into the online portal Primer3 (http://bioinfo.ut.ee/primer3-0.4.0/) (*see* **Note 29**).
2. Make sure that the primer binding site does not contain any nucleotide variation.
3. Once you have sequences, you can easily order primers online.
4. Conduct a PCR using the annealing temperature specified by Primer3 (*see* **Note 30**) (Table 1).
5. Send off the PCR product to a sequencing facility for Sanger sequencing.
6. Confirm the genotypes using the Sanger chromatograms.

**Table 1**
**Example of enzyme amounts to use for a 20 μl PCR reaction**

| Reagent | ×1 | ×4 |
|---|---|---|
| $ddH_2O$ | 9.8 | 39.2 |
| 10× buffer (comes with Taq) | 2 | 8 |
| BSA | 2 | 8 |
| $MgCl_2$ | 1.6 | 6.4 |
| F primer | 1 | 4 |
| R primer | 1 | 4 |
| dNTPs | 0.4 | 1.6 |
| Taq polymerase | 0.2 | 0.8 |
| Template DNA | 2 | 8 |
| Total | 20 | 80 |

## 4 Notes

1. Make sure that the lab space used is very clean. It is good to wash benches with RNAse-away or a similar RNAse remover beforehand.
2. Integrity of the RNA can be determined using denaturing MOPS agarose gels or a Bioanalyzer.
3. The Illumina TruSeq kits come with positive controls, which can be used to investigate where things have gone wrong during library preparation. These known sequences, if used, will have to be removed bioinformatically postsequencing.
4. Depending on the desired sequencing depth per sample, samples can in most cases be combined in one sequencing reaction. In this case, it is essential to use the barcoded adapters provided with the kit, and to not mix two samples with the same barcode.
5. Illumina sequencing is with few exceptions conducted by a sequencing center. When choosing which sequencing center to use, there are three important considerations: (a) Communication—do the technical staff answer to emails within a reasonable time? (b) Queue—how long will it take before your data will be available? (c) Price—is the sequencing possible considering the available budget?
6. There are many different potential quality control protocols, but the most important is to examine the distribution of base call qualities along the short Illumina reads, and to remove any artifacts from the sample preparation procedure. Artifacts can consist of either remains of adapter sequences or as PCR duplicates.

The objectives of this section are to (a) remove all bases with a Phred quality score of less than 20, (b) remove any adapter sequences present in the data, (c) graph the distributions of quality scores and nucleotides, and (d) calculate the fractions of duplicate and singleton reads in the data.

7. The bash script TrimClip.sh first invokes the quality trimmer, which scans through all reads, and when it encounters a base with a quality score of less than 20, trims off the rest of the read and then subsequently removes reads shorter than 20 bases. A temporary file is created, which is then used as an input file for the adapter clipper. The clipper removes any read ends that match the defined adapter sequences and then removes reads that after clipping are shorter than 20 bases.
8. The bash script CollapseDuplicateCount.sh first uses fastx_collapser to combine and count all identical reads. A temporary FASTA-formatted file called YOURFILE_collapsed.txt is created, which is then used as an input file for a python script (fastqduplicatecounter.py) that calculates the fractions of duplicate reads and singletons. This file is removed at the end of the program since it was just an intermediate step.
9. The easiest way to view the plots is by copying this file to your local drive and opening it there. The plots should look something like Fig. 1a, b. If the mean quality scores are low throughout or if the nucleotides are nonrandomly distributed, something could have gone wrong during sample preparation or sequencing.
10. RNA-Seq reads represent short pieces of all the mRNA present in the tissue at the time of sampling. In order to be useful, the reads need to be combined—assembled—into larger fragments, each representing an mRNA transcript. These combined sequences are called "contigs," which is short for "contiguous sequences." A de novo assembly joins reads that overlap into contigs without any template (i.e., no reference genome/transcriptome).
11. Building a de novo assembly is a very memory-intensive process. There are many programs for this, some of which are listed later. We are using Trinity [14] in this section, an assembler that is thought to work very well for transcriptomes, as opposed to others that are optimized for genome assembly. Trinity uses *De Bruijn graphs* to join reads together (*see* Fig. 2a). De Bruijn graphs summarize sequence variation in a very cost-effective way, speeding up the assembly process. Nevertheless, it is a very memory-intensive step, and having access to a computer cluster might be necessary if the number of reads is high.
12. When comparing the lengths and numbers of contigs acquired from de novo assemblies to the predicted number of transcripts from genome projects, the de novo contigs typically are shorter and more numerous. This is because the assembler cannot join contigs together unless there is enough overlap and coverage in the reads, so that several different contigs will match one

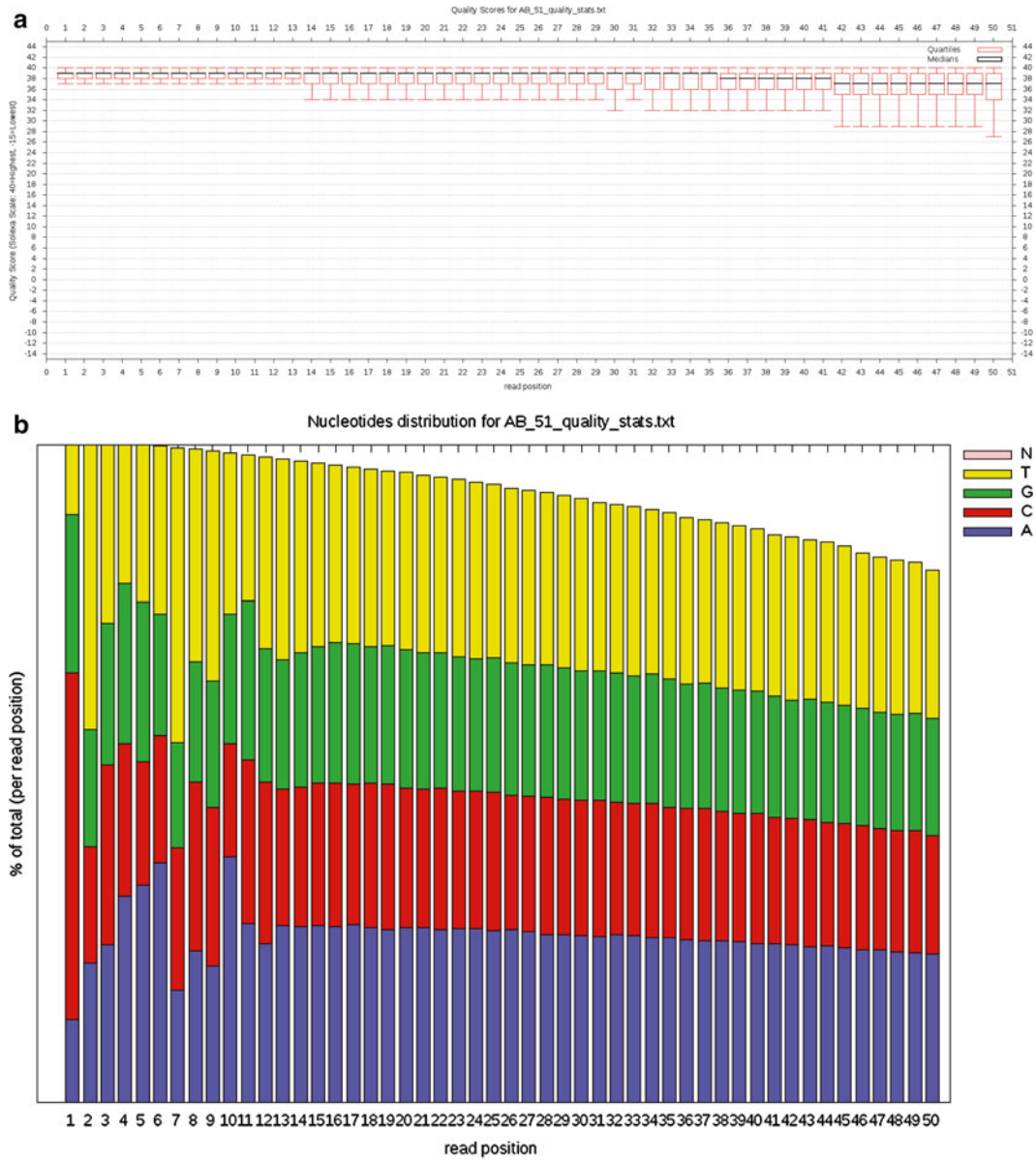


**Fig. 1** (**a**) Quality score boxplot of 50-bp Illumina reads (after quality trimming, $Q < 20$), summarized by read position. Lower scores at the beginning of the reads are due to an artifact of the software used to calculate base quality scores. (**b**) Nucleotide distribution chart of 50-bp Illumina reads, summarized by read position. A nonrandom distribution in the first 12 bases is common and is thought to be an artifact of the random hexamer priming during sample preparation

mRNA transcript. Biologically, alternative splicing of transcripts also inflates the number of contigs when compared to predictive data from genome projects. This is important to keep in mind, especially when analyzing gene expression data based on mapping to a de novo assembly. To minimize this issue, we want to use as many reads as possible in the assembly

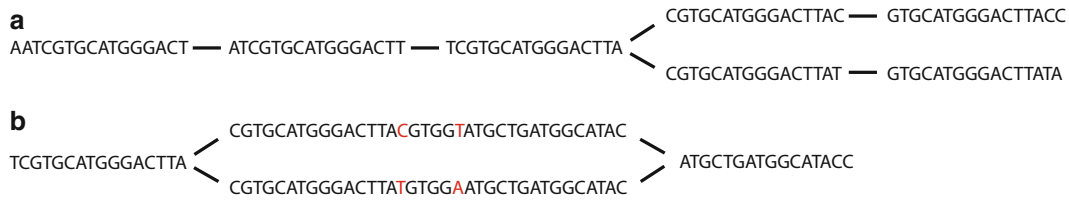


**Fig. 2** (**a**) An example De Bruijn graph with k-mer size 16 and 5 nodes. (**b**) A bubble caused by two SNPs or sequencing errors. Shorter k-mers will decrease bubble size, but could increase fragmentation if coverage is not high enough

to maximize the coverage level. We therefore pool the reads from all our samples, which means that no information about the individual samples can be extracted from the assembly. In order to get sample-specific information, we need to map our reads from each sample individually to the assembly once it has been created (next section).

13. There are several parameters one can vary when assembling a transcriptome or genome. Perhaps the most important one is the k-mer (word) length of the De Bruijn graphs. Longer k-mers can help resolve repeat regions in genome assemblies and can be useful to resolve homeolog genes in polyploid species, whereas shorter one can increase performance in polymorphic sequences (*see* Fig. 2b). As Trinity focuses on transcriptome assembly, the k-mer length is preset to 25. In other assemblers, it can vary considerably.
14. Mapping refers to the process of aligning short reads to a reference sequence, whether the reference is a complete genome, transcriptome, or a de novo genome/transcriptome assembly. The program that we will utilize is called BWA [15], which uses a Burrow's Wheeler Transform method, with the goal of creating an alignment file also known as a Sequence/Alignment Map (SAM) file for each of your samples. This SAM file will contain one line for each of the reads in your sample denoting the reference sequence (genes, contigs, or gene regions) to which it maps, the position in the reference sequence, and a Phred-scaled quality score of the mapping, among other details [16].
15. There are several parameters that can be defined for the alignment process, including: the number of differences allowed between reference and query (–n), the length (–l) and number of differences allowed in the seed (–k), the number allowed and penalty for gap openings (–o, –O), and the number and penalty for gap extensions (–e, –E). Changing these parameters will change the number and quality of reads that map to reference and the time it takes to complete mapping a sample. For a complete list of the parameters and their default values, go to http://bio-bwa.sourceforge.net/bwa.shtml.

16. We will map the reads from each of your trimmed and clipped FASTQ files to the de novo reference assembly that you created in the previous section. Specifically, we will (a) create an index for the reference assembly (just once), which will help the aligner (and other downstream software) to quickly scan the reference; (b) for each sample, map reads to the reference assembly; and (c) convert the resulting file into the SAM file format and append "read group" names to the SAM file for each sample. Steps b and c are "piped," or put together feeding the output of one program in as the input for the next program. The read groups, which can have the same names as your sample names, will be appended to each file and will become critical for the downstream SNP detection step. The read group name in each SAM file will connect the reads back to individual samples after files have been merged for SNP detection. All of the earlier steps for all samples can be "batch" processed at once by editing the bash script BWAaln.sh. We then want to remove all duplicate reads, for which we need to use the MarkDuplicates program from the software package "Picard." Picard uses the binary equivalent of SAM files, BAM, as input, so first we need to convert the files using SAMtools. These steps are performed by the convert_to_BAM_and_dedup.sh bash script.
17. From now on, we will work with the binary equivalent of the SAM file format: BAM. BAM files take up less place on a hard drive and can be processed faster. Most SNP detection software are made to process BAM files. The drawback is that they cannot be examined directly in a text editor. Our first task is to remove any duplicate reads from the alignments, for which we also need to sort our aligned reads by alignment position. Identical, duplicate reads can be a result of biology and represent highly expressed transcripts. However, they are also quite likely to be an artifact of the PCR step in the sample preparation procedure. Artifactual duplicates can skew genotype estimates so they must be identified for SNP estimation.
18. The convert_to_bam_and_dedup.sh script has two elements: (a) It converts the .sam file to a binary bam file and sorts the reads within it. (b) It marks and removes duplicate reads using the MarkDuplicates program from the Picard package.
19. For all the data processing steps within this section, I have chosen to follow the recommendations of the Broad Institute, created for the Genome Analysis Toolkit (GATK): http://www.broadinstitute.org/gatk/guide/topic?name=best-practices [17]. I highly recommend keeping an eye on the instructions of this site for more information and updated protocols. They also have an excellent forum for posting technical questions. The only step in their protocol that we do not use is the Base Quality Score recalibration, as this step requires a list of known variant sites as input. If you do have access to this

type of data, it is highly recommended to follow the instructions on the GATK site.

20. The objectives of this section are to (1) merge your alignment files and realign poorly mapped regions, (2) detect variant sites and filter out true sites from false positives, (3) extract genotype information for all individuals at all variant sites.

21. There are three major steps to this section of the protocol. First, we need to process our alignment files slightly. We start by merging all the deduplicated .bam files from Subheading 4 into one file called merged.bam, which will be our base for SNP discovery. At this step, it is crucial that the "read group" headings for your samples (which we specified in the previous section) are correct, as they will be used to keep track of the samples within the merged .bam file. We then index our merged .bam file and search through the file for areas containing indels, where the initial mapping might be of poor quality. By using information from all samples in the merged file in a realignment step, we improve our chances of correctly aligning these regions. The merged realigned .bam file is what we will use in the next step, variant (SNP) detection and genotyping. An initial search for only very high-quality variant sites outputs a .vcf file, which is a list of all variant sites and the genotypes of all individuals for those sites. For information about the vcf file format, *see* http://www.1000genomes.org/node/101. We will consider the high-quality variants "true" sites for further processing. An additional search for variant sites, now with a lower quality threshold, is then conducted and by using our "true" variant sites from the first search we can build a Gaussian mixture model to separate true variants from false positives using a log-odds ratio (VQSLOD) of a variant being true vs. being false: (http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_variantrecalibration_VariantRecalibrator.html). Following this, we can extract the genotype information for each individual from the .vcf file, while specifying a genotype quality threshold, and use this information to calculate allele and genotype frequencies. For simplicity we will use $Q=20$ ($p=0.99$) as a threshold.

22. There are two parts to the realigner.sh script: (a) call on RealignerTargetCreator to search for poorly mapped regions near indels and save those intervals in an output.intervals file. (b) Call on IndelRealigner to realign the intervals specified in step 1, and save the realigned file as merged_realigned.bam.

23. The SNP_detection.sh script has three elements: It calls on the GATK HaplotypeCaller to only call variant sites with a Phred scale quality of more than 20 (probability of being a true variant site >0.99). This will be used as a set of "true" variant sites to train the Gaussian mixture model used by the Variant Quality Score Recalibrator (VQSR) in the next step. The VQSR depends on a set of true variant sites, so if you are working with an organism for

which a validated set of variants exist, it is recommended to use that data here. However, as we are working with nonmodel organisms, we cannot assume that this data will always be available so let's assume that we have no prior knowledge in this case. You will want the quality threshold to be as high as possible at this point, but with out limited dataset, we will have to settle for $Q=20$ as a threshold. The script then calls on the HaplotypeCaller to call SNPs with a threshold that is largely determined by the sequencing depth. As we have low coverage due to our truncated fastq files, we will use a low-quality threshold here ($Q=3$). In reality, you would want to maximize this to reduce the chance of false positives. Finally, the script uses the VariantAnnotator to add annotations to the .vcf file output. The high-quality variant sites are stored in a file called: raw_snps_indels_Q20.vcf, while the variants that should be used for the final call set are in a file called: raw_snps_indels_Q3_annotated.vcf.

24. The VQSR.sh script has five elements: (a) It uses the high-quality SNP dataset to train a model that can be used for filtering the true SNPs from false positives in our call dataset. (b) It uses the high-quality InDel dataset to train a model that can be used for filtering the true InDels from false positives in our call dataset. (c) It applies the SNP model to the call data and flags all SNPs failing the filter. (d) It applies the InDel model and flags all InDels failing the filter. (e) It saves only the variant sites that have passed the VQSR into a new file called VQSR_PASS_SNPS.vcf.
25. If you get an error message when running the VQSR.sh script, try changing the settings for -percentBad, -minNumBad, and --maxGaussians in the first two commands of VQSR.sh using nano, then resaving and rerunning the script.
26. The final argument of the getgenosfromvcf.py script specifies a genotype Phred quality score cutoff of 20 (99% probability of being true). This parameter can be changed according to your needs. The "rows" argument specifies that SNPs will be output in rows, with two columns per individual, one for each allele (specifying "cols" would return the same output, but with SNPs as columns, two columns per SNP).
27. There are many different software and methods to do this, so I will not go into much detail here.
28. The true test of a putative SNP is whether it can be validated using different methods. There are a variety of methods available for this, but we will focus on a traditional way, which is to design primers and to amplify and sequence fragments using PCR and Sanger sequencing.
29. Design primers: RNA-Seq data does unfortunately not contain any information about intron–exon boundaries, so the safest place to design primers is within the coding regions. It is also

possible to do this outside of coding frames, but in this case it can be nice to have access to a genome of a closely related species, in order to minimize the risk of designing primers that span over an intron.

30. Choosing samples for Sanger sequencing validation: Use DNA preferably from individuals indicated as homozygotes for the reference and alternative alleles at the SNP site of interest. It is possible to use heterozygotes as well, with an expectation of a double peak in the Sanger chromatogram, but PCR artifacts can potentially obscure this pattern.

## References


1. Barton NH, Keightley PD (2002) Understanding quantitative genetic variation. Nat Rev Genet 3(1):11–21
2. Vos P, Hogers R, Bleeker M, Reijans M, Vandelee T, Hornes M, Frijters A, Pot J, Peleman J, Kuiper M, Zabeau M (1995) AFLP – a new technique for DNA-fingerprinting. Nucleic Acids Res 23(21):4407–4414
3. Richardson BJ, Baverstock PR, Adams M (1986) Allozyme electrophoresis: a handbook for animal systematics and population studies. Academic, San Diego, CA
4. Slatkin M (1995) A measure of population subdivision based on microsatellite allele frequencies. Genetics 139(1):457–462
5. Selkoe KA, Toonen RJ (2006) Microsatellites for ecologists: a practical guide to using and evaluating microsatellite markers. Ecol Lett 9(5):615–629
6. 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(10)
7. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10:57–63
8. Beaumont MA, Nichols RA (1996) Evaluating loci for use in the genetic analysis of population structure. Proc R Soc B Biol Sci 263(1377):1619–1626
9. Charlesworth B, Nordborg M, Charlesworth D (1997) The effects of local selection, balanced polymorphism and background selection on equilibrium patterns of genetic diversity in subdivided populations. Genet Res 70(2):155–174
10. Martin JA, Wang Z (2011) Next-generation transcriptome assembly. Nat Rev Genet 12(10):671–682
11. Konczal M, Koteja P, Stuglik MT, Radwan J, Babik W (2013) Accuracy of allele frequency estimation using pooled RNA-Seq. Mol Ecol Resour 14:381–392
12. Skelly DA, Johansson M, Madeoy J, Wakefield J, Akey JM (2011) A powerful and flexible statistical framework for testing hypotheses of allele-specific gene expression from RNA-seq data. Genome Res 21:1728–1737
13. De Wit P, Pespeni MH, Palumbi SR (2015) SNP genotyping and population genomics from expressed sequences - current advances and future possibilities. Mol Ecol 24(10):2310–2323
14. Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, Regev A (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol 29(7):644-U130
15. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25(14):1754
16. 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 25(16):2078
17. DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, Philippakis AA, del Angel G, 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. Nat Genet 43:491–498