Botany 2012
Introduction to Next-Generation Sequencing Workshop
Practical Exercises

Prepared by
Shannon Straub
Postdoctoral Scholar - Liston Lab
Oregon State University
straubs@science.oregonstate.edu

The following exercises are intended to help you familiarize yourself with Illumina data and some of the basic kinds of analyses that are performed using these data. After the conclusion of the workshop, a great resource for finding answers to your questions about sample preparation, raw data handling, and basic and advanced genomics analyses is SEQanswers (http://seqanswers.com/).

Exercise 1 – Illumina Raw Data

Illumina raw data analyzed using Casava v. 1.8 pipeline are returned as fastq files. Depending on the number of reads obtained and the standards at your sequencing center, the read pool may be broken down into several smaller files of more manageable size. Some data analysis programs accept fastq files and utilize the quality information that they contain, while other programs will require the fastq files to be converted to fasta format.

A. The Anatomy of a fastq File

1. Use any text editor to open the example fastq file (example_fastq.fq) provided for the workshop in the "example_files" directory. Fastq formatted files returned from the Illumina pipeline contain 4 lines of information per record. Look at the example file to make sure you understand the information contained in each line, especially the Illumina header in line 1.

Line 1: This line starts with '@' and includes text to identify the sequence. The identifiers for Illumina reads contain several pieces of information separated by colons. The information for reads that have been analyzed using Illumina's Casava v. 1.8 will contain the following information: @instrument name:run id:flow cell id:lane number:tile number:x-coordinate for this cluster on tile:y-coordinate for this cluster on tile number of a pair(1 or 2):flag for whether this read has passed Illumina chastity and purity filter (N=pass, Y=fail):control bits:index(barcode) sequence. Many users will filter reads based on the Illumina quality flag.

Line2: This line contains the sequence information.

Line3: This line starts with '+' and may or may not include more information, such as a repetition of the information included in line 1.

Line4: This line contains the quality scores for each base of the sequence read in line 2. Illumina's Casava v. 1.8 encodes standard Sanger quality scores (Phred+33) using ASCII characters. If you have legacy Illumina data (pre-v. 1.8) be aware that that quality scores may be one of the older Illumina formats. Multiple scripts and programs are available to convert these older Illumina quality scores to Sanger quality scores. Many users prefer to filter or trim reads based on their quality scores prior to beginning their analyses. See http://www.usadellab.org/cms/index.php?page=trimmomatic for one such versatile example.

B. File format conversion

If your data analysis program of choice requires conversion of your fastq files to fasta there are several options to perform the conversion. Fasta formatted files are probably familiar to most of you. If not, you can open the example provided for the workshop in the "example_files" directory that was converted to fasta format from the example fastq file from part A using method #3 below (example_fasta.fa). These files simply contain a header beginning with a ">" and containing a name and description of the sequence separated by a space, this information is followed by one or more lines of sequence associated with that header.

1. Some analysis programs will perform the conversion prior to initiating, so check this first.

2. Use a script. An efficient perl script can be downloaded at: http://brianknaus.com/software/srtoolbox/fastq2fasta.pl

3. There are many ways to accomplish this conversion using the command line. Here's a linux example:

   ```
   sed 3~4d example_fastq.fq | sed 3~3d | sed 's/@/>/' > out.fa
   ```

Exercise 2 - Short Read Mapping

Short read mapping involves aligning reads to a reference sequence based on a set of criteria. Some reads will map to a single location, but others, such as those that originate from repetitive areas of a genome, may map to multiple locations and are called multimaps. Read mapping is useful for determining sequencing depth, as well as for calling single nucleotide polymorphisms (SNPs) between the sequenced sample and a reference sequence. BWA (Li and Durbin, 2009) and Bowtie (Langmead et al., 2009), both based on Burrows-Wheeler transform, are two popular short read mapping programs. Geneious also has a read mapping function, which we will use to map reads from a genome skim of *Asclepias boliviensis* to an annotated *Asclepias syriaca* reference sequence.

A. The reference sequence

1. View the provided *A. syriaca* reference plastid genome (plastome) in Geneious by clicking on the "read_mapping_de_novo_assembly" directory and choosing the name of the file (JF433943) in the document table pane. Take a few minutes to familiarize yourself with the types of annotations that will be present for a plastome downloaded from GenBank and how these annotations are displayed in Geneious.

B. Map the reads

1. To map the reads highlight both the read file (Asclepias_boliviensis_reads.fsa) and the reference file (JF433943) by clicking on each while holding down the shift key.

2. Next, select the "Align/Assemble" button from the Geneious menu and choose "Map to Reference…"

3. You should see the JF433943 file in the "Reference Sequence" box. The default settings are fine for this analysis. Check the boxes next to "Save assembly report" and "Save contigs." Click the "Ok" button at the bottom of the window to begin the read mapping analysis.

C. Estimate the chloroplast content of the *A. boliviensis* Illumina library

1. When the read mapping is complete, click on the assembly report in the Geneious document table pane.

2. From the assembly report we will estimate the chloroplast content of the *A. boliviensis* library and the sequencing depth of the plastome for this individual. Assuming the sequence of the *A. boliviensis* plastome is unknown, we can make estimates of each of these metrics using the number of reads that map to the *A. syriaca* plastome.

a. Chloroplast content calculation

(______ reads mapped to cp / ______ total reads) * 100 = _______ % cp

b. Sequencing depth calculation – For this calculation you will also need the read length, which can be found in the document table pane under min or max sequence length, and the genome size, in this case the plastome size for the *A. syriaca* reference (158,798 bp).

(_____ reads mapped * _____ bp read length)/_____ bp genome size
= _____× sequencing depth

D. Examine the read mapping results

1. Click on the file named "Asclepias_boliviensis_reads assembled to JF433943" in the document table pane to open a graphical view of the read mapping results. This first view will show the consensus sequence of the mapped reads, the coverage of each part of the plastome based on read mapping depth, and the annotated reference sequence. To view some more basic information about the analysis choose the Statistics panel by clicking the tab with a % sign at the right hand part of the window. How well does your sequencing depth estimate from Part C match up with the Geneious base by base coverage estimate?

2. Next, zoom in using the magnifying glass button above the Statistics panel to get a better look at the read mapping results. If you zoom in far enough you can see the actual sequence of each read. Take a few minutes to explore the results, perhaps by finding your favorite plastid gene using the reference sequence annotations.

3. Click on the display tab (screen icon) to the right side of the Statistics panel. Make sure that the box next to "Highlighting" is checked and that the next two boxes form the phrase "Disagreements to Reference." While exploring the mapped reads, you will notice that differences in the assembled reads and reference are now highlighted in each read. How can you distinguish SNPs from sequencing errors?

Other highlighted features you may encounter are areas of misalignment of reads. Also note that apparent sequencing errors could have a biological basis and originate from chloroplast pseudogenes residing in the mitochondrial or nuclear genomes, which have been sequenced at much lower depth than the plastid genes and retain high enough similarity to be mapped.

4. Geneious can aid in the search for SNPs. To find and annotate SNPs in coding regions, choose “Annotate & Predict” then select “Find Variations/SNPs.” Set the “Minimum Coverage” to 25 and “Minimum Variant Frequency” to 0.8. Next, choose “Only in CDS” from the dropdown menu next to “Find Polymorphisms.” Then click ok.

5. Explore the results by looking at the new track “Variations” that has been added to the “Annotations and Tracks” menu (yellow arrow tab to the right of the Contig View). Use the arrows to jump between SNPs. If you mouse over the orange SNP track markers in the Contig view you will be able to see detailed information about each SNP, such as whether it causes a synonymous or nonsynonymous amino acid change.

Exercise 3 – Plastome Assembly

The plastome of plants provides a good starting place for learning how to deal with the different types of assembly due to its tractable size and conserved gene content and organization.

A. Starting a de novo assembly in Geneious

De novo assemblies of genomes use only the sequences of the short reads themselves to create longer stretches of sequences (contigs). The two main approaches employed are overlap-layout-consensus and de Bruijn graph algorithms. Some of the recently popular programs available for de novo assembly include ABySS (Simpson et al., 2009), ALLPATHS-LG (Gnerre et al., 2011), SOAPdenovo (Li et al., 2010), and Velvet (Zerbino and Birney, 2008). See section 4.7.1 of the manual to learn more about the overlap-layout-consensus assembly algorithm employed by Geneious.

A de novo assembly of the *A. boliviensis* read pool will take approximately 30 min. to run in Geneious assuming an allocation of 2 GB of RAM to the process and a ~2.8 GHz processor. We will start this analysis then continue with the reference-guided assembly exercise while we wait for the results.

1. If you were able to allocate 2GB of RAM to Geneious for the de novo assembly prior to the workshop, you are ready to move on to step 2 to start the de novo assembly. If not, make sure the read_mapping_and_de_novo_assembly directory is still selected. Then choose File -> Import -> From File… and navigate to the workshop distribution files that you downloaded prior to the workshop. Open the "de_novo_assembly_low_RAM" directory, select all of the files, and click "Import." Use these files for the de novo assembly exercises after you complete the reference-guided assembly exercise that begins at the top of pg. 7.

2. To begin the de novo assembly, click to highlight the name of the *Asclepias boliviensis* read file (Asclepias_boliviensis_reads.fsa) in the top document table pane of Geneious (read_mapping_and_de_novo_assembly directory still selected). You should then see a graph showing the lengths of the reads in the file appear in the bottom pane. In this case, they are all 74 bp.

3. Next, choose the "Align/Assemble" button from the Geneious menu and select "De Novo Assemble…"

4. Click the box next to "Save assembly report" in the Results section. The other default settings are fine for a preliminary analysis, so then click "OK."

5. Go on to the reference-guided assembly section and we will return to the results of the de novo analysis after it has completed.

B. Reference-guided assembly

In reference-guided methods, a reference sequence is employed to aid genome assembly. This approach is especially amenable to the assembly of plant plastomes. A sequence from a closely related species is the best reference choice, but the majority of the plastome can be assembled using a reference in the same order as your species of interest (Straub et al., 2012). Some of the common assembly mistakes encountered in reference-guided assembly include incorrectly assembled or missing insertions and deletions relative to the reference and missed rearrangements relative to the reference sequence's orientation.

Alignreads (Straub et al., 2011) is an assembly pipeline developed in the Liston lab for reference-guided assembly. The pipeline incorporates YASRA (Yet Another Short Read Aligner) (Ratan, 2009), which handles indels well and performs even when the reference sequence and the sequence to be assembled are divergent. YASRA maps reads to a reference genome using the LASTZ algorithm (Harris, 2007) and refines their alignment using ReAligner (Anson and Myers, 1997). Genomic regions without coverage after the initial round of alignment are then masked, and the program attempts to assemble the missing sequence by tiling the remaining reads across the masked region. The resulting assembly is used as the reference for a subsequent round of alignment and tiling, and the process iterates until no further improvement is made. Next, NUCmer and delta filter from the MUMmer 3.0 suite (Delcher et al., 2002; Kurtz et al., 2004) are used to align the assembled YASRA contigs to a reference sequence, which can be either the reference sequence used for assembly or another reference of interest. Two Liston lab scripts, sumqual and qualtofa, are used to integrate the YASRA assembly information and the NUCmer alignment information and then format the results into a user-friendly fasta file, as well as apply filters based on user inputs, such as masking for sequencing depth. Alignreads can be downloaded from http://milkweedgenome.org/?q=node/28.

Alignreads runs on the linux operating system. For the purposes of this workshop, we will provide instructions on how to construct the command line for running the pipeline, but we will not actually be able to do an analysis. We will then view the output from Alignreads using Geneious, however this output is optimized for viewing in BioEdit (Hall, 1999).

1. The basic command line for running Alignreads, which is a python script, contains the following information:

   ```
   python alignreads.py [options] <reads in fasta format> <reference in fasta format>
   ```

   Alternatively, if you just want to change masking or alignment parameters without repeating the assembly, you can use the following command:

   ```
   python alignreads.py [options] <YASRA folder>
   ```

2. See the attached handout for detailed usage of alignreads.py. A typical command line for alignreads looks like this:

```
python  alignreads.py -t 454 -o linear -w 5 -x 25 0.8 Asclepias_boliviensis_reads.fsa
Asclepias_syriaca_reference_no_IR.fsa
```

This set of parameters will complete the plastome assembly pipeline for *A. boliviensis* using the *A. syriaca* plastome reference sequence with only one copy of the inverted repeat included (see below). The –t 454 option indicates that YASRA should use its 454 setting for longer reads. In the older versions of YASRA, the solexa (illumina) setting was optimized for very short reads (<40 bp), so the current read lengths of 80-150 bp were too long and are assembled better using the 454 setting for longer short reads. We will soon be releasing a new version of Alignreads that will be compatible with the latest YASRA release, where the solexa/illumina setting will work just fine for Illumina reads. The –o linear option is indicating that the reference sequence and the assembled sequence are expected to be linear. For reference guided assembly of the plastome, we have found that removing one copy of the inverted repeat to produce a linear, rather than circular reference, improves the assembly. The –w 5 option indicates that in the consensus sequence any base with a sequencing depth <5 should be masked (i.e. converted to 'N'). The –x 25 0.8 option indicates that for a SNP vs. the reference to be called, the sequencing depth must be at least 25 with 80% of those reads supporting the SNP.

3. Open the alignreads assembly output file in Geneious by choosing the "reference_guided_assembly" directory and clicking on Asclepias_boliviensis_alignreads_assembly.fsa in the document table. In the Alignment View you will see an identity graph followed by 34 sequences. The first sequence is the reference sequence. It is followed by the consensus sequence of the contigs in the assembly and then by the masked version of this sequence based on user criteria. The YASRA contigs aligned to the reference and consensus sequences make up the remaining sequences. Take a few minutes zoom in and explore the consensus and contig sequences. You should see the same SNPs that we discovered in the read mapping analysis, but indels may have been reconstructed in the reference-guided analysis that were not apparent from read mapping.

4. To look at a specific case involving an indel, check for differences in the consensus sequences for the reference-guided assembly and read mapping results for the the following stretch of sequence. For the assembly, click on the name of the reference sequence in the Alignment View. Then jump to position 17,580 by clicking the "Jump to a specific position in the sequence" button (red arrow pointing to a black bar) in the upper right corner of the Alignment View options. For the read mapping result, click on the consensus and then jump to base 17,723 to view the same location. It might be helpful for you to select one view and have your neighbor select the other view so that you can look at them both at the same time instead of flipping back and forth between views. For a better direct comparison of the assemblies for this area, you can view contigs from the Asclepias_boliviensis_YASRA_pileup.ace file. The contigs in this file are numbered and listed in the document table pane when the reference_guided_assembly directory is selected. These contigs come from a read pile-up output by YASRA that will give you a similar view to the read mapping

results. To view the same stretch of sequence, look at bases 8,539-8,583 in contig 3. In this view you will have to use slide the bar at the bottom of the Contig View window to reach the desired location. Which of the two reconstructions is more likely to be correct (i.e. Which method performed better?)?

Note that we used the default read mapping setting in Geneious, but choosing options for fine tuning the read mapping could lead to better handling of indels.

5. You may have noticed that several of the contigs have been aligned multiple times (e.g., contig 8). Multiple alignments occur when there is an indel event that is too large to be handled under our default alignment parameters or when repetitive sequences cause a misassembly or misalignment. You can manually adjust the alignment to incorporate this information or try adjusting the NUCmer options and re-running the latter part of the Alignreads pipeline to correct the alignments for indels. To do a manual adjustment on contig 8, start at the beginning of the first alignment of the contig and scan down the assembly until you see where this sequence is no longer consistent with the consensus sequence. It will be easier to see cases of disagreement if you select the “Use dots” option under “Highlighting” in the “Display” menu (screen icon) at the right hand side of the Alignment View window. Alternatively, jump ahead to base 50,212 after clicking on the name of the reference sequence in the Alignment View.

6. Next, copy the first 10-20 bp of the sequence that does not match with the consensus (If you selected the “Use dots” option, deselect it before copying.). Then click on the second alignment Contig 8 (2). Next click on “Annotate & Predict” in the Geneious menu bar. Choose “Search for Motifs…” from this menu. When the “Search for Motifs…” menu opens, paste the sequence you copied into the “Sequence or PROSITE motif” box. Uncheck the “Add results to track” box if it is checked. Then click “Ok” to find the copied sequence in the second alignment, as well as the reference and consensus sequences.

7. Now that you can clearly see where this stretch of sequence occurs, the alignment can be edited. To do so, click the “Allow Editing” button at the top of the Alignment View window. Now highlight the stretch of sequence you searched for in the first alignment of contig 8. Once it is highlighted you can click and slide it down so that it aligns with the occurrence of the sequence in the second alignment of contig 8 and the consensus sequences.

8. If you have the “Use dots” box selected, unclick it to view the full sequence. Looking at the full sequence you should be able to see that the multiple alignment was caused by a direct repeat in the *A. syriaca* reference that may be missing in *A. boliviensis.* Before accepting this explanation, the read pile-up should be checked. Go to contig 8 and look for position 3,108. You will see that there is not a single read

that spans the length of this sequence, so misassembly may be a more likely explanation than a missing repeat. In the latter case, we would expect to find reads that span the length of the sequence because a single copy of the repeat is shorter than the read length. Now the masked consensus sequence (the precursor of the final *A. boliviensis* plastome sequence) can be manually edited to reflect your discovery. To be conservative, masking (N's) should be added across the length of the repeat region to signify our uncertainty about the actual sequence in this region.

9. Explore additional aspects of the reference-guided assembly result until the de novo assembly analysis is complete. The other multiple alignments of contigs will provide more complex examples of problems that need to be corrected in a reference-guided assembly.

C. De novo assembly

Start by clicking on the directory that contains the de novo assembly results to view all of the result files in the document table pane of Geneious.

1. Click on the Assembly Report in the document table pane. You can use this report to explore the assembly statistics associated with your run. We will look at this information in more detail in the next section.

2. Next, click on the Sequence Length tab in the document table pane and sort the contigs from largest to smallest.

3. Click on Contig6, the longest contig recovered in the de novo assembly. Use the information reported about this contig to begin filling in Table C on pg. 11.

4. Now we will try to determine the genomic origin of the contig and some genes that might be contained in this contig using a Custom Blast in Geneious. To begin, click the Sequence Search Button.

5. Within the Sequence Search panel choose "selected_plant_genomes" as the database. This is the custom library you imported prior to the workshop. Then for Program, choose "Discontiguous Megablast." The remainder of the default options should be fine, so then click the Search button to start your BLAST search.

6. When the BLAST hits are returned, explore the results and use them to complete the rest of the table for Contig6.

7. Repeat this process for contigs 825, 50, and 119. Randomly choose some additional contigs and explore those as well.

Table C. Contig statistics and preliminary annotations for a de novo assembly of the *Asclepias boliviensis* genome

Contig Number	Contig Length	# Reads Assembled	Genome (nuc., cp, mt)	Preliminary Gene Annotations (based on annotations of best BLAST hits)
6				
825				
50				
119				

Based on the number of reads that assembled for contig 119, what can you say, in general, about its copy number in the genome? Considering the preliminary gene annotation, is this surprising?

8. Note that the chloroplast contigs in the de novo assembly could be ordered relative to one another using a reference sequence to produce a plastome sequence. Based on the information presented in the morning session, what factors do you think would be important in getting a successful plastome assembly using de novo techniques?

D. Assessing assembly quality

Assembly quality is assessed in several different ways. In general the more sequence in long contigs and fewer contigs (provided most of the reads are being used) the better. The N50 is a value often reported for genome assemblies and indicates that 50% of the assembly is in contigs of that size or greater, and again the higher the better. Although we cannot compare the de novo and reference-guided assemblies directly because one is an assembly of the whole genome and the other an assembly of only the plastome, we can explore these values to get an idea of assembly quality.

1. Return to the assembly report from the de novo assembly analysis to fill in values in Table D.
2. Use any text editor to open the Asclepias_boliviensis_contig_stats.txt file distributed for the workshop in the example_files directory. Use this information in combination with that available in the document table pane when the reference_guided_assembly directory is highlighted to complete the information about the reference-guided assembly.

Table D. Assembly statistics for de novo and reference guided assemblies of the *Asclepias boliviensis* read pool.

Assembly method	Number of reads used	Number of Contigs	Number of Contigs >1kb	Length of the longest contig	N50 (all contigs)	Length of the assembly (bp) in contigs > 1kb
de novo						
reference-guided	90,785					

Even though they are not directly comparable, which assembly appears to be better based on its stats in Table D? Why?

References Cited:

ANSON, E. L., and E. W. MYERS. 1997. ReAligner: a program for refining DNA sequence multi-alignments. *Journal of Computational Biology* 4: 369-383.

DELCHER, A. L., A. PHILLIPPY, J. CARLTON, and S. L. SALZBERG. 2002. Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Research* 30: 2478-2483.

GNERRE, S., I. MACCALLUM, D. PRZYBYLSKI, F. J. RIBEIRO, J. N. BURTON, B. J. WALKER, T. SHARPE, et al. 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences, USA* 108: 1513-1518.

HALL, T. A. 1999. BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. *Nucleic Acids Symposium Series* 41: 95-98.

HARRIS, R. S. 2007. Improved pairwise alignment of genomic DNA. Ph.D. dissertation, Pennsylvania State University, University Park, Pennsylvania, USA.

KURTZ, S., A. PHILLIPPY, A. L. DELCHER, M. SMOOT, M. SHUMWAY, C. ANTONESCU, and S. L. SALZBERG. 2004. Versatile and open software for comparing large genomes. *Genome Biology* 5: R12.

LANGMEAD, B., C. TRAPNELL, M. POP, and S. SALZBERG. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology* 10: R25.

LI, H., and R. DURBIN. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25: 1754-1760.

LI, R., W. FAN, G. TIAN, H. ZHU, L. HE, J. CAI, Q. HUANG, et al. 2010. The sequence and de novo assembly of the giant panda genome. *Nature* 463: 311-317.

RATAN, A. 2009. Assembly algorithms for next-generation sequence data. Ph.D. Dissertation, The Pennsylvania State University, University Park.

SIMPSON, J. T., K. WONG, S. D. JACKMAN, J. E. SCHEIN, S. J. M. JONES, and İ. BIROL. 2009. ABySS: a parallel assembler for short read sequence data. *Genome Research* 19: 1117-1123.

STRAUB, S. C. K., M. PARKS, K. WEITEMIER, M. FISHBEIN, R. C. CRONN, and A. LISTON. 2012. Navigating the tip of the genomic iceberg: next-generation sequencing for plant systematics. *American Journal of Botany* 99: 349-364.

STRAUB, S. C. K., M. FISHBEIN, T. LIVSHULTZ, Z. FOSTER, M. PARKS, K. WEITEMIER, R. C. CRONN, et al. 2011. Building a model: developing genomic resources for common milkweed (*Asclepias syriaca*) with low coverage genome sequencing. *Bmc Genomics* 12: 211.

ZERBINO, D. R., and E. BIRNEY. 2008. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research* 18: 821-829.

Alignreads Usage

```
Usage: python alignreads.py [options] <Reads in .fa file> <Reference>  OR...
       python alignreads.py [options] <YASRA folder>

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -H, --advanced-help   Display help information for all supported options
                        (Default: only basic options are shown)
  -z STRING, --import-options=STRING
                        specify the path to a Command_Line_Record.txt fie from
                        a previous run, or the folder that contains one. Any
                        other options used with this one are overwritten.
                        (Default: use options supplied)
  -Z, --debug           Save the debug log to the current working directory.
                        (Default: dont save)

  YASRA-Related Modifiers:
    -d, --silent        Nothing is printed to the screen (Default: print the
                        output of yasra to the screen)
    -t 454 or solexa, --read-type=454 or solexa
                        Specify the type of reads. (Default: solexa)
    -o circular or linear, --read-orientation=circular or linear
                        Specify orientation of the sequence. (Default:
                        circular
    -p same, high, medium, low or very low, --percent-identity=same, high, medium, low or very low
                        The percent identity (PID in yasra). The settings
                        correspond to different percent values depending on
                        the read type (-t). (Defalt: same)
    -a, --single-step   Activate yasra's single_step option (Default: run
                        yasra normally)
    -E FILEPATH, --external-makefile=FILEPATH
                        Specify path to external makefile used by YASRA.
                        (Default: use the makefile built in to runyasra)
    -Q, --no-dot-replace-reads
                        Do NOT replace N's with dots (.) in the microreads
                        file before running yasra/ (Default: replace dots)
    -I, --no-dos2unix-ref
                        Do NOT run dos2unix on the reference before running
                        yasra/ (Default: run dos2unix)

  NUCmer-Related Modifiers:
    -f STRING, --prefix=STRING
                        Set the output file prefix (Default: out)
    -b INT, --break-length=INT
                        Distance an alignment extension will attempt to extend
                        poor scoring regions before giving up (Default: 200)
    -j INT, --alternate-ref=INT
                        Specify a new reference to be used in the rest of the
                        alignment after yasra. (Default: use YASRA's
                        reference)
    -A mum, ref, or max, --anchor-uniqueness=mum, ref, or max
                        Specify how NUCmer chooses anchor matches using one of
                        three settings: mum = Use anchor matches that are
                        unique in both the reference and query, ref =  Use
                        anchor matches that are unique in the reference but
                        not necessarily unique in the query, max = Use all
                        anchor matches regardless of their uniqueness.
                        (Default = ref)
    -T INT, --min-cluster=INT
```

```
                        Minimum cluster length used in the NUCmer analysis.
                        (Default: 65)
  -D FLOAT, --diag-factor=FLOAT
                        Maximum diagonal difference factor for clustering,
                        i.e. diagonal difference / match separation used by
                        NUCmer. (Default: 0.12)
  -J, --no-extend       Prevent alignment extensions from their anchoring
                        clusters but still align the DNA between clustered
                        matches in NUCmer. (Default: extend)
  -F, --forward-only    Align only the forward strands of each sequence.
                        (Default: forward and reverse)
  -X INT, --max-gap=INT
                        Maximum gap between two adjacent matches in a cluster.
                        (Default: 90)
  -M INT, --min-match=INT
                        Minimum length of an maximal exact match. (Default:
                        20)
  -C, --coords          Automatically generate the <prefix>.coords file using
                        the 'show-coords' program with the -r option.
                        (Default: dont)
  -O, --no-optimize     Toggle alignment score optimization. Setting
                        --nooptimize will prevent alignment score optimization
                        and result in sometimes longer, but lower scoring
                        alignments (default: optimize)
  -S, --no-simplify     Simplify alignments by removing shadowed clusters.
                        Turn this option off if aligning a sequence to itself
                        to look for repeats. (Default: simplify)

Delta-Filter-Related Modifiers:
  -y INT, --min-identity=INT
                        Set the minimum alignment identity [0, 100], (Default:
                        80)
  -l INT, --min-align-length=INT
                        Set the minimum alignment length (Default: 100)
  -K FLOAT, --max-overlap=FLOAT
                        Set the maximum alignment overlap for -r and -q
                        options as a percent of the alignment length [0, 100].
                        (Default 100)
  -B, --query-alignment
                        Query alignment using length*identity weighted LIS.
                        For each query, leave only the alignments which form
                        the longest consistent set for the query. (Defualt:
                        global alignment)
  -R, --ref-alignment
                        Reference alignment using length*identity weighted
                        LIS. For each reference, leave only the alignments
                        which form the longest consistent set for the
                        reference. (Defualt: global alignment)
  -G, --global-alignment
                        Global alignment using length*identity weighted LIS
                        (longest increasing subset). For every reference-query
                        pair, leave only the alignments which form the longest
                        mutually consistent set. (this is the default)
  -U FLOAT, --min-uniqueness=FLOAT
                        Set the minimum alignment uniqueness, i.e. percent of
                        the alignment matching to unique reference AND query
                        sequence [0, 100]. (Default 0)

sumqual-Related Modifiers:
  -Y, --save-ref-dels
                        Save the sequence of the reference that corresponds to
                        empty gaps in the consensus in a fasta file. (Default:
```

```
                        dont save)

qualtofa-Related Modifiers:
  -c, --exclude-contigs
                        Dont include each contig on its own line (Default:
                        include contigs)
  -i, --no-match-overlap
                        Add deletions (i.e. -'s) to the reference to
                        accommodate any overlapping matches. (Default:
                        Condense all overlapping regions of the consensus into
                        IUPAC ambiguity codes.)
  -e, --no-overlap    Add deletions (i.e. -'s) to the reference to
                        accommodate any overlapping sequence, including
                        unmatched sequence. (Default: Condense all overlapping
                        regions of the consensus into IUPAC ambiguity codes.)
  -k, --keep-contained
                        Include contained contigs (Default: save sequences of
                        contained contigs to a separate file)
  -q INT, --end-trim-qual=INT
                        Trim all the bases on either end of all contigs that
                        have a quality value less than the specified amount
                        (Default: 0)
  -s, --dont-save-SNPs
                        Dont save SNPs to a .qual file(Default: Save SNP file)
  -W, --dont-align-contigs
                        Do NOT align contigs to the reference using '-'s at
                        the start of each contig; independent of the
                        consensus. (Default: align contigs)
  -N INT, --end-trim-num=INT
                        Trim the ends of the contigs by the specified number
                        of bases. (Default: 0)
  -L INT, --min-match-length=INT
                        Set minimum length of the matching region of the
                        contigs. (Default: 50)

Coverage and Call Proportion Masking:
  The following options take one integer argument and one decimal
  argument between 0 and 1, if the second is not supplied it is assumed
  to be 0. Do NOT put a comma between the entries.

  -m, --mask-contigs  Set minimum coverage depth and call proportion for
                        contig masking; independent of the consensus. Cannot
                        be used with the -c modifier.(Default: 0 0)
  -n, --mask-contig-SNPs
                        Set minimum coverage depth and call proportion for
                        contig SNP masking; independent of the consensus.
                        Cannot be used with the -c modifier.(Default: 0 0)
  -w, --mask-consensus
                        Set minimum coverage depth and call proportion for the
                        consensus; a new masked sequence will be added to the
                        output file. (Default: 0 0)
  -x, --mask-SNPs     Set minimum coverage depth and call proportion for
                        SNPs in the consensus; a new masked sequence will be
                        added to the output file. (Default: 0 0)
```