DISSERTATION

EXPERIMENTAL AND COMPUTATIONAL ANALYSIS OF CAENORHABDITIS ELEGANS SMALL RNAS

Submitted by

Kristen Brown

Graduate Degree Program in Cell and Molecular Biology

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2019

Doctoral Committee:

Advisor: Tai Montgomery

Dawn Duval
Ashok Prasad
Ann Hess

ABSTRACT

EXPERIMENTAL AND COMPUTATIONAL ANALYSIS OF CAENORHABDITIS ELEGANS SMALL RNAS

Caenorhabditis elegans contains twenty-five Argonautes, of which, only ALG-1 and ALG-2 are known to interact with microRNAs (miRNAs). ALG-5 belongs to the AGO subfamily of Argonautes that includes ALG-1 and ALG-2, but its role in small RNA pathways is unknown. We analyzed by high-throughput sequencing the small RNAs associated with ALG-5, ALG-1, and ALG-2, as well as changes in mRNA expression in *alg-5*, *alg-1*, and *alg-2* mutants. We show that ALG-5 defines a distinct branch of the miRNA pathway affecting the expression of genes involved in immunity, defense, and development. In contrast to ALG-1 and ALG-2, which associate with the majority of miRNAs and have general roles throughout development, ALG-5 interacts with only a small subset of miRNAs and is specifically expressed in the germline. *alg-5* is required for optimal fertility and mutations in *alg-5* lead to a precocious transition from spermatogenesis to oogenesis. Our results provide a near-comprehensive analysis of miRNA-Argonaute interactions in *C. elegans* and reveal a new role for miRNAs in the germline.

The small RNA field has grown rapidly since miRNAs were discovered to be conserved regulators of developmental timing. This growth occurred during a time when high-

throughput transcriptomic data from microarrays and next-generation sequencing became widely accessible. As a result, research projects dissecting small RNA pathways often produce sequencing data that can be complex and difficult to perform appropriate data analysis for without specialized or advanced computational knowledge. Many researchers end up only study a subset of small RNAs, outsourcing their analysis, or piecing together a pipeline using tools developed for mRNA sequencing. We aim to reduce this barrier to entry in the field and improve reproducibility by creating an open-source, user-friendly data processing pipeline for small RNA sequencing. To create a simple, reproducible pipeline, we utilized the Common Workflow Language (CWL) and Python, while otherwise minimizing dependencies. The pipeline reads a configuration file and sample sheets that can be easily modified by a user to run the complete analysis from raw fastq file to summary statistics and publication-ready plots. We present AQuATx (automated quantitative analysis of transcript expression) for small RNAs and the analysis of *C. elegans* germline tissue as an example data set. Our software will allow bench scientists with little to no computational knowledge to easily analyze their small RNA sequencing data. Overall, the final software will be a valuable tool for anyone interested in studying small RNAs.

PREFACE

Chapter 1 is an introduction to the small RNA field, a literature review discussing the necessary background and history of small RNA research in C. elegans, and a discussion of best practices in computational biology research and software development.

Chapter 2 describes the characterization of the Argonaute ALG-5 (previously named T23D8.7/*hpo-24*) as well as a comparison to the well-known Argonautes, ALG-1 and ALG-2.

Chapter 3 describes the development and implementation of a small RNA sequencing data analysis tool and guidelines for improvements called AQuATx

Chapter 4 is a brief discussion of the overall accomplishments of this body of work, the limitations, and a larger discussion around potential future directions.

Appendix I contains the supplementary material for Chapter 2.

Appendix II contains the supplementary material for Chapter 3.

TABLE OF CONTENTS

ABSTRACT ii
PREFACE iv

CHAPTER 1. INTRODUCTION 1
1.1 SMALL NON-CODING RNAS 1
1.1.1 ARGONAUTE PROTEINS 3
1.1.2 MICRORNA PATHWAY 6
BIOGENESIS OF MIRNAS 7
MIRNA FUNCTION 8
THE MIRNA-ASSOCIATED ARGONAUTES IN C. ELEGANS 10
BROAD ROLES FOR MIRNAS IN C. ELEGANS 11
1.1.3 SIRNA AND PIRNA PATHWAYS 13
BIOGENESIS OF SIRNAS AND PIRNAS 13
SMALL RNAS IN THE GERMLINE OF C. ELEGANS 14
1.1.4 HIGH THROUGHPUT SEQUENCING OF SMALL RNAS 17
1.2 *C. ELEGANS* AS A MODEL ORGANISM 19
GERMLINE DEVELOPMENT 19
1.3 REPRODUCIBLE RESEARCH IN COMPUTATIONAL BIOLOGY 21
BEST PRACTICES FOR BIOINFORMATICS SOFTWARE 23
WORKFLOW ENGINES 26
REFERENCES 30

CHAPTER 2. FUNCTIONAL SPECIALIZATION OF THE MICRORNA-ASSOCIATED ARGONAUTES IN C ELEGANS 40
2.1 INTRODUCTION 40
2.2 MATERIALS AND METHODS 42
2.3 RESULTS 51
2.3.1 ALG-5 IS REQUIRED FOR PROPER DEVELOPMENTAL TIMING IN THE GERMLINE 51
2.3.2 ALG-5 IS PRIMARILY EXPRESSED IN THE GERMLINE 54
2.3.3 ALG-5 FUNCTIONS IN THE MIRNA PATHWAY 58
2.3.4 ALG-5, ALG-1, AND ALG-2 INTERACT WITH DISTINCT SUBSETS OF MIRNAS 60
2.3.5 DIFFERENTIAL GENE EXPRESSION IN ALG-5, ALG-1, AND ALG-2 MUTANTS 64
2.3.6 FUNCTIONAL OVERLAP BETWEEN THE MIRNA-ASSOCIATED ARGONAUTES 69
2.4 DISCUSSION 73
REFERENCES 78

CHAPTER 3. AUTOMATED ANALYSIS OF SMALL RNAS WITH AQUATX 82
3.1 INTRODUCTION 82
3.2 IMPLEMENTATION 85
3.2.1 OVERALL WORKFLOW DEVELOPMENT 85
3.2.2 USER INPUT AND CONFIGURATION 87
3.2.3 THE AQUATX STANDARD WORKFLOW 88
3.2.4 TEST SUITE & CONTINUOUS INTEGRATION 97
3.2.5 STEPS TO BE IMPLEMENTED 98
3.2.6 CODE AVAILABILITY 104
3.2.7 ANALYSIS OF GERMLINE SMALL RNAS IN C. ELEGANS 105
3.3 DISCUSSION 109
REFERENCES 111

CHAPTER 4. DISCUSSION 113
4.1 SUMMARY 113
4.2 FUTURE DIRECTIONS 114
4.2.1 THE ROLE OF ALG-5 AND MIRNAS IN THE GERMLINE 114
4.2.2 WHY ARE ALG-2 MUTANTS LONG-LIVED? 117
4.2.3 RNAI AND THE STUDY OF SMALL RNA PATHWAYS 118
4.2.4 IMPROVEMENTS TO AQUATX 120
FINAL REMARKS 122
REFERENCES 123

APPENDIX I 124
S2.1 SUPPLEMENTARY FIGURES FOR CHAPTER 2 125
APPENDIX II 131
S3.1 AQUATX USER GUIDE V0.1 132
S3.1.1 MAJOR VERSION SUMMARIES 132
S3.1.2 PRE-REQUISITES AND INSTALLATION (V0.1) 133
S3.1.3 USING THE END-TO-END WORKFLOW 136
S3.1.4 USING INDIVIDUAL STEPS AS STANDALONE TOOLS 145
S3.2 BASIC WORKFLOW OUTPUTS 152
S3.3 ADDITIONAL CODE 154
S3.3.1 AQUATX HELPER SCRIPTS 154
S3.3.2 SHINYSEQBROWSER 155
S3.3.3 MISCELLANEOUS SCRIPTS 156

1. INTRODUCTION

Due to the interdisciplinary nature and scope of this work, the background information is quite broad. To provide context, this chapter is an introduction to the small RNA field[1] and a discussion of emerging best practices in computational biology research. More focus is given to the microRNA pathway in *Caenorhabditis elegans* studied in Chapter 2, but other small RNA pathways are discussed to provide broader context and motivations for Chapter 3.

1.1 SMALL NON-CODING RNAS

Small non-coding RNAs have emerged as a major regulator of gene expression throughout the development of plants and animals. Small RNAs have distinct temporal and spatial expression, which is important for their roles in controlling developmental timing and cell specification during development [1, 2]. In animals, small RNAs play important roles in regulating diverse processes such as the proper differentiation of stem cells, organ development, response to drug treatments and pathogen infections [2-5]. Additionally, misexpression of small RNAs have been linked to numerous diseases, including cancers, neurodegenerative diseases, and diabetes [2, 4-10]. In plant development, small RNAs play fundamental roles in growth, flowering, leaf patterning [11-13], modulating responses to viral infection and abiotic stresses such as

[1] The majority of this text is unique to this dissertation. Small portions were published as a part of perspective articles highlighting recently published primary research we reviewed:
- Brown, K.C. and Montgomery, T.A. (June 2018). The long and short of lifespan regulation by Argonautes. PLoS Genetics.
- Brown, K.C. and Montgomery, T.A. (May 2017) Epigenetic Inheritance: Perpetuating Transgenerational RNAi. Current Biology.

temperature and drought tolerance [12, 14]. As the roles of small RNAs span many pathways and the field is relatively young, there are still interesting and fundamental discoveries to be made.

Small RNAs typically act to downregulate or “turn off” the expression of messenger RNAs (mRNAs) through their interactions with a protein co-factor named Argonaute, a key component of the RNA-induced silencing complex (RISC). While the small RNA provides the regulatory target information through sequence complementarity with target mRNAs, it does not have regulatory activity outside the RISC. A minimal, functional RISC complex contains a small RNA and an Argonaute protein [15-17]. Downregulation of targeted mRNAs is thought to occur through direct cleavage of the transcript, recruitment of decay factors to degrade the transcript, or by blocking translation into protein [18-20]. Small RNAs are often used to experimentally dissect mRNA function through “knockdown” experiments that take advantage of sequence-specific targeting. Studying small RNAs across systems and pathways has improved our understanding of specific gene function and regulatory networks in plants and animals. Small RNAs also make attractive candidates for practical biotechnology and clinical applications for their potential to target and downregulate specific gene products. In particular, small RNAs have been used for developing novel therapeutics targeting disease-associated genes [21, 22] and to improve crop yield or resistance to various biotic and abiotic stresses [23, 24].

There are three major classes of small RNAs in animals: small interfering RNAs (siRNA), piwi-interacting RNAs (piRNAs), and microRNAs (miRNAs). Small RNAs can be classified by their size, precursor duplex structure, 5' nucleotides, the protein Argonautes they interact with, and/or their mode of target repression, as well as their genetic requirements [17, 25, 26]. siRNAs require perfect sequence complementarity with their mRNA targets for silencing and are most well known for their function in the RNA interference (RNAi) pathway [18, 25]. In animals, miRNAs only require partial complementarity, but are often near perfect complementarity in plants and play important roles in development [27, 28]. piRNAs are found in the germline of animals and have been characterized as a defense mechanism against transposable elements to maintain genomic integrity [25, 29].

1.1.1 ARGONAUTE PROTEINS

Argonaute proteins were initially identified in *Arabidopsis thaliana* and named for the squid-like appearance of leaves in loss-of-function mutant plants [30]. Argonautes have three main conserved domains that define them: the PIWI domain, the MID domain, and the PAZ domain (Figure 1.1). The PIWI domain is RNaseH-like and in some Argonautes has endonucleolytic activity as determined by a DDH motif. The PAZ domain interacts with the 3' end of the small RNA and the MID domain the 5' end. [31-33]. The loading of small RNAs can be specialized for each Argonaute based on specific features of the small RNA, such as terminal nucleotide, length, or secondary structure. For example, miRNA-associated Argonautes often have a U or A nucleotide bias at the 5' end [17, 34-

37]. This bias for specific nucleotides might be explained by the MID domain, where the 5' end is anchored and has been shown to have specific contacts for UMP/AMP in human AGO2 [38]. Beyond conferring regulatory activity of mRNA, Argonaute proteins have also been shown to stabilize overall expression of miRNAs and thus might help protect all small RNAs from turnover [39].

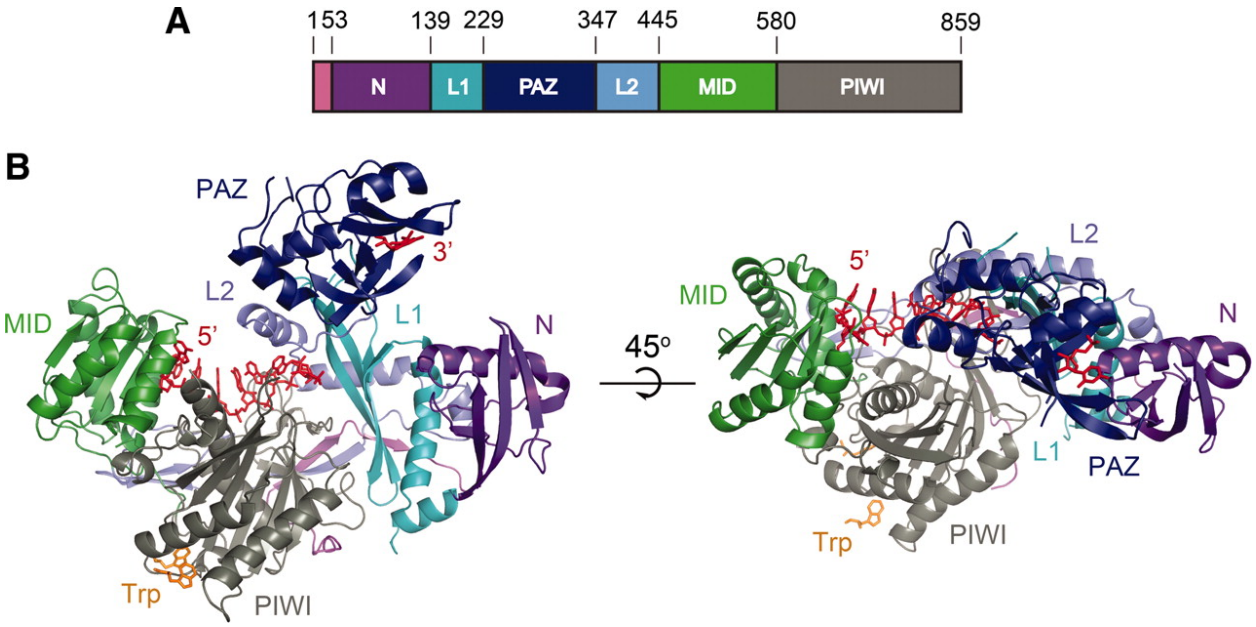


Figure 1.1 Structure of human Argonaute2. A) Ago2 primary sequence schematic B) Views of Ago2 structure: N (purple), PAZ (navy), MID (green), and PIWI (gray) domains and linkers L1 (teal) and L2 (blue). RNA (red) can be traced at 1 to 8 and 21 nt. Tryptophan (orange) binds to hydrophobic pockets within PIWI. Reprinted from [40] with permission (License #4575590550394).

In *C. elegans* there are ~25 Argonaute proteins divided into three distinct clades: the PIWI clade, the AGO clade, and the worm-specific AGO clade (WAGO) (Figure 1.2). The PIWI clade consists of proteins that are part of the piRNA pathway and interact with piRNAs. The WAGO clade consists of proteins that are a part of the RNAi pathway and mainly interact with siRNAs in *C. elegans*. The AGO clade is highly conserved across

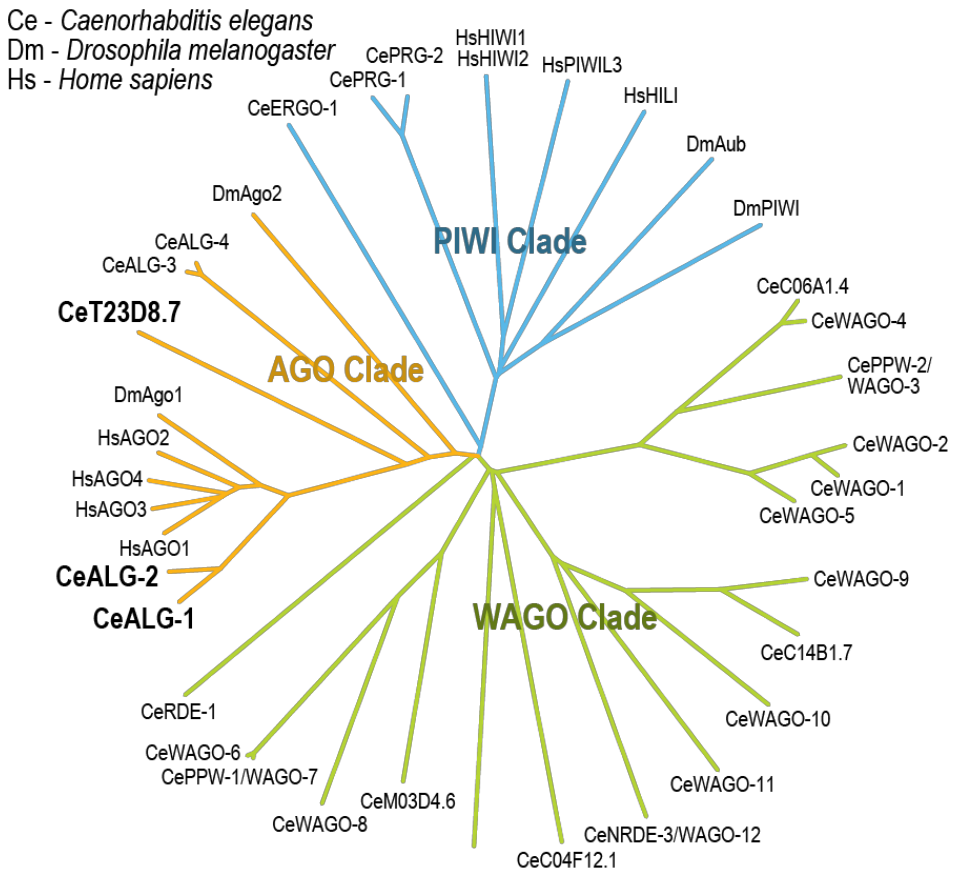


Figure 1.2 Argonaute Proteins. A phylogenetic tree of known and predicted Argonaute proteins. Adapted from [41]. Argonautes characterized as a part of this dissertation are in bold.

animals and contains Argonautes that interact with miRNAs and siRNAs [42-46]. Within the AGO clade of *C. elegans*, there are two Argonautes known to interact with miRNAs, ALG-1 and ALG-2 [47, 48]. Two other *C. elegans* specific AGO-clade Argonautes, ALG-3 and ALG-4, interact with 26-nt siRNAs that comprise a spermatogenesis-specific endogenous RNAi pathway [49]. As part of my dissertation, I discovered that a fifth *C. elegans* Argonaute in this clade, T23D8.7/HPO-24, also interacts with miRNAs, which I will describe in detail in this dissertation (Chapter 2). However, *hpo-24* was identified in a genetic screen of gene inactivations that lead to hypersensitivity to pore-forming toxins and was initially named for this phenotype (Hypersensitive to POre forming

toxins) [50]. We have since renamed this Argonaute gene to *alg-5* to reflect its relatedness to other Argonautes.

1.1.1 THE MICRORNA PATHWAY

The most well-studied class of small RNAs is the miRNA class, which controls almost every aspect of development in plants and animals. miRNAs are highly conserved among animals [2, 4]. The first miRNA, *lin-4*, was discovered in 1993 as a part of two studies of *C. elegans* developmental timing and thought to be a weird, nematode-specific phenomenon [51, 52]. Several years later, a second miRNA regulating developmental timing, *let-7,* was discovered in *C. elegans*, but this time the sequence was found to be highly conserved across many organisms, including humans [53, 54]. This sparked widespread interest in the study of small regulatory RNAs across fields and many new small RNAs have since been discovered and characterized. Interestingly, the small RNA field grew alongside the adoption of high-throughput transcriptomic technologies such as microarrays and next-generation sequencing due to decreasing costs (Figure 1.3). As a result, a good proportion of small RNA studies involve transcriptomic datasets. These studies often look at mutant or overexpression lines of individual small RNAs or proteins within the pathways, such as biogenesis factors or Argonautes.

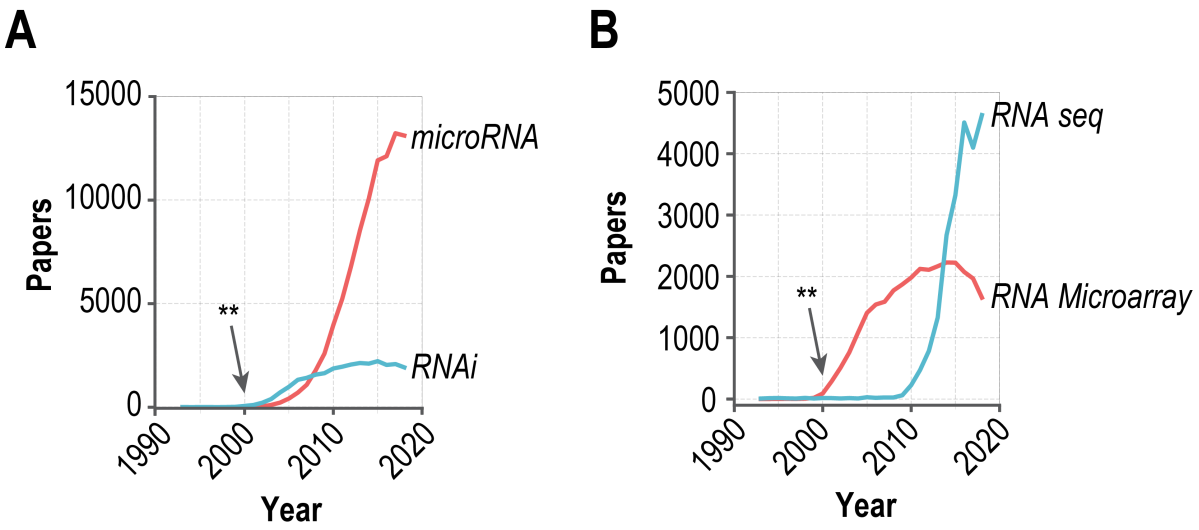


Figure 1.3 The rise of interest in small RNAs. A) The number of publications from 1993 to 2019 containing "microRNA" and "RNAi". B) The number of publications from 1993 to 2019 containing "RNA seq" and "RNA microarray". Trendlines were created using data from Medline [55]. **Discovery of the miRNA *let-7* [53, 54].

BIOGENESIS OF MIRNAS

The canonical biogenesis pathway of miRNAs is fairly well defined and conserved across animal species. Long, primary transcripts (pri-miRNAs) are transcribed by RNA polymerase II and form characteristic foldback structures resembling hairpins (Figure 1.4B) [56, 57]. The protein Drosha/Pasha recognizes the hairpin structure and processes it into a shorter pre-miRNA hairpin [58-61] that is then exported out of the nucleus by exportin-5 [62-64]. Next, within the cytoplasm, Dicer recognizes and processes the pre-miRNA from the stem regions of partially base-paired RNA hairpins into ~22-nt duplexes with 2-nt 3' overhangs (Figure 1.4B) [48, 60, 65, 66]. A similar pathway for miRNA biogenesis exists in plants as well [28]. There is evidence that suggests miRNAs also derive from a non-canonical biogenesis pathway, beginning with transcription by RNA polymerase III in intronic regions or downstream from tRNAs [67, 68]. Finally, miRNA duplexes form ribonucleoprotein complexes with effector proteins in

the Argonaute family, upon which one of the two strands, the passenger or so-called star strand, is ejected or degraded [69-71]. The miRNA strand retained in the complex acts as a sequence-specific guide to anchor the Argonaute to a target mRNA [72]. Historically, the “star strand” was thought to lack regulatory activity and was labeled by having the fewest reads of the duplex in initial sequencing datasets. However, evidence emerged indicating that these canonical passenger/star strands may also exhibit regulatory activity and be more highly expressed depending on the context [73-77]. The nomenclature was updated in miRBase since release 17, and miRNAs are now referred to by -3p or -5p arms to avoid regulatory implications [78, 79].

MIRNA FUNCTION

Each miRNA acts as a sequence-specific guide to direct an Argonaute protein to a target mRNA for silencing [17]. miRNAs regulate gene expression through inhibition of translation, direct cleavage, or recruitment of mRNA decay factors. In plants, the mode of repression is often direct target cleavage due to the high complementarity between miRNAs and target mRNAs, but translational repression has been shown to occur as well, at least in some contexts [28, 80]. In animals, the mode of silencing was initially thought to mainly occur as translational repression, but more recent evidence suggests that mRNA destabilization is actually the predominant and steady-state form of regulation. Translational repression seems to occur initially and rapidly, but is a weak, unstable mode of action and accounts for 10 – 25% of the effects of miRNAs [31, 80-84]. In worms, miRNAs are ~22 nucleotides long and target sequences matching a 6 – 8

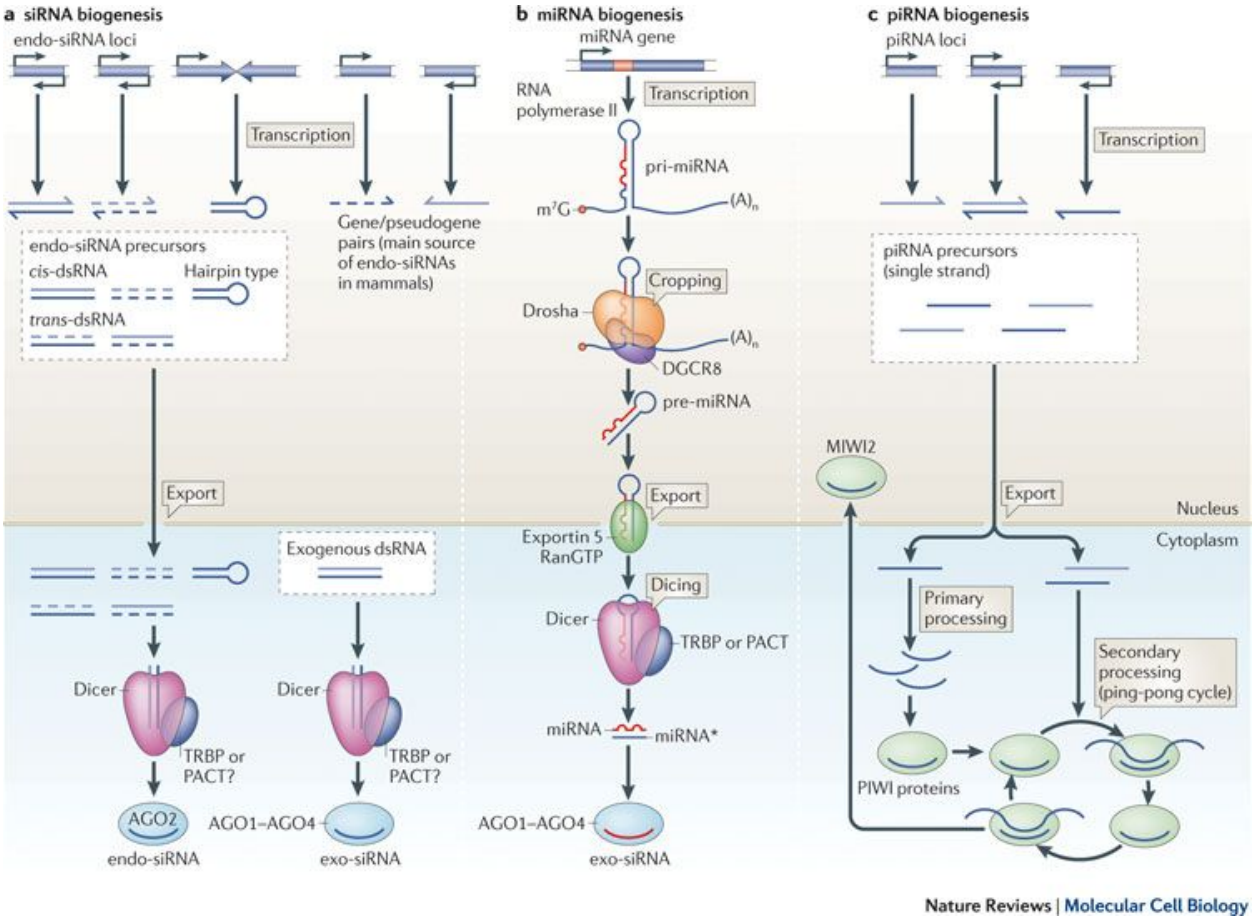


Figure 1.4 The biogenesis of small RNAs in animals. A) Endogenous small-interfering RNA biogenesis pathway. B) Canonical miRNA biogenesis pathway. C) The piRNA biogenesis pathway. Reprinted from [85] with permission (License #4577960960379)

nt “seed” sequence at the 5’ end, which is the defining, highly conserved region of the sequence. This seed sequence canonically targets the 3’ untranslated region (3’UTR) of complimentary mRNA sequences. The seed sequences are often present in multiple miRNAs and are used to place miRNAs into “families”, which have been shown to regulate the same genes redundantly [58, 72, 86]. With these parameters in mind, miRNAs are predicted to broadly target many mRNAs, but a short seed sequence naturally produces many false positives in software for target prediction [87]. There is

emerging evidence that predicts miRNAs can also utilize base pairing at the 3' end of the miRNA beyond the seed and target mRNAs within the coding region [88-90]. As we still do not fully understand the functions of miRNAs, the downstream effects on mRNAs can be difficult to study. Thus, studying individual miRNAs alongside the Argonaute proteins can be a fruitful method for dissecting miRNA function at a more general scale.

THE MIRNA-ASSOCIATED ARGONAUTES IN C. ELEGANS

In *C. elegans* there are two well-known miRNA-associated Argonautes, ALG-1 and ALG-2, both of which have essential roles in embryonic development. *alg-1* and *alg-2* are largely redundant with one another and loss of function in either, but not both, is permissible. Mutations in both *alg-1* and *alg-2* lead to embryonic lethality. Mutations in *alg-1* alone cause several developmental defects, while those in *alg-2* alone do not cause obvious defects [47, 91]. ALG-1 and ALG-2 seem to be fairly ubiquitously expressed throughout development and across tissue types, with specific expression occurring only in a couple tissue types [47, 92]. Because of their sequence relatedness and overlapping roles in many processes, one would predict that ALG-1 and ALG-2 function redundantly and thus might act synergistically in adult animals as well. However, recent studies of *alg-1* and *alg-2* expression and function in aging [92, 93], show that *alg-1* expression declines, while *alg-2* stays consistent through aging. One of these studies also indicates that *alg-1* and *alg-2* actually have opposing roles in longevity and aging [92]. While differences in miRNA binding and temporal or spatial expression of the two proteins likely account at least in part for their opposite roles, it is

possible that the two Argonautes also have distinct biochemical function from one another. ALG-1 and ALG-2 share ~75% amino acid similarity but diverge substantially at their N termini which could lead to different protein interactions or subcellular localization [47]. As part of my dissertation project, I characterized a third miRNA-associated Argonaute, T23D8.7/*alg*-5 (Chapter 2). ALG-5 only shares ~36% amino acid similarity with ALG-1 and ALG-2 and likely has distinct biochemical functionality [48]. There is additional evidence that some more "promiscuous" Argonautes exist in *C. elegans* which interact with more than just miRNAs. RDE-1 is one such Argonaute, primarily known for its function in the RNAi pathway, but has been shown to interact with a few miRNAs [94, 95]. Characterization of the many Argonautes in *C. elegans* and other organisms has given us important insights into the broader roles of small RNA pathways.

BROAD ROLES FOR MIRNAS IN C. ELEGANS

miRNAs play diverse roles in *C. elegans* that make it suitable for studying miRNA regulation of many interesting pathways. Individual miRNAs are not essential to *C. elegans* development or viability as demonstrated by their lack of developmental phenotypes in animals that have lost individual miRNA sequences [96]. This is reasoned to be due to the redundancy of seed sequences amongst miRNAs that groups them into families that likely target the same mRNAs. Even so, loss of most families of miRNAs are also not essential [97]. This could suggest that either there is additional redundancy or regulation among essential pathways or not all members of each family

have been discovered or characterized. However, the loss of individual miRNAs in a genetically sensitive background where animals have a loss-of-function mutation in *alg-1*, does lead to various phenotypes that may suggest specific functions for that miRNA [98]. Of specific interest, the *miR-35* family of miRNAs is one of the families which is essential to development and loss of this family leads to embryonic lethality [97], much like loss of *alg-1* and *alg-2* together [47, 91].

As with most genes, the expression of individual miRNAs varies during development, however, global downregulation of miRNAs often occurs in aging [99-103]. Individual miRNAs have important roles in aging and can both shorten and extend lifespan [101, 104, 105]. Specifically, *miR-71* seems to extend lifespan through the insulin signaling pathway, but by acting within neurons that transmit the signal [106] and *mir-239* seems to shorten lifespan [101]. Another interesting, perhaps related, mechanism miRNAs control in *C. elegans* is the response to changing environmental cues. Several miRNAs, including *mir-71* and *mir-239* regulate the response to heat stress. In particular, loss of *mir-71* leads to hypersensitivity and loss of *mir-239* leads to resistance [101, 107]. The miRNAs *let-7, mir-251, mir-67, mir-233,* and others also function in the innate immune response to pathogen infection [108-111]. In particular, *mir-67* seems to regulate avoidance behavior through the regulation of a neuronal adhesion molecule important in neuronal development [112]. There is more evidence that suggests miRNAs play important roles in neurons and neuronal development [113, 114]. For example, *mir-84* plays a role in synaptic remodeling of motor neurons [115] and *lsy-6* is key to

specification of the ASE neurons [114]. There are many miRNAs in *C. elegans* that have yet to be characterized and will surely provide additional insight into the complex regulatory networks of development, aging, and stress response.

1.1.2 SIRNA AND PIRNA PATHWAYS

miRNAs are probably the most widely studied class of small RNA, but to avoid missing interesting information in our studies, we need to take into account all classes of small RNAs. After the initial discoveries of miRNAs and their influence on developmental timing, the silencing of genes in plants and animals by short, ~22 nt RNAs called siRNAs, triggered by exogenous double-stranded RNA (dsRNA) was described [116, 117]. Exogenous RNAi provided a robust way for researchers to knockdown specific target genes by providing a long, dsRNA that was complimentary to the target sequence [118]. The first piRNAs were discovered in *Drosophila melanogaster* and shown to be crucial for fertility [85, 119, 120]. They have since been found in most animal gonads and to act as a "genome defense" mechanism against transposable elements [85, 119].

THE BIOGENSIS OF SIRNAS AND PIRNAS

siRNAs and piRNAs have similar, but distinct biogenesis pathways from miRNAs (Figure 1.4). The production of exogenous siRNAs is triggered by long dsRNA, either through infection by viral sources or experimental injection [116, 121]. Endogenous siRNAs are produced from various genomic loci, like miRNA primary transcripts, with perfect complementarity to produce long dsRNA [122]. Within the cytoplasm, these

longer transcripts are processed by Dicer into shorter duplexes with 2-nt 3' overhangs, which are then loaded into Argonaute proteins (Figure 1.4A) [25, 122, 123]. Unlike miRNAs and siRNAs, piRNAs are derived from single-stranded precursors and their biogenesis does not rely on Dicer. piRNAs are often found in clusters in the genome and seem to be restricted to germ cells of animals. In *D. melanogaster*, piRNAs undergo an amplification process, called the ping-pong cycle, to amplify expression of piRNAs from transposon mRNAs (Figure 1.4C) [25, 85, 119]. *C. elegans* also has a small RNA amplification process that utilizes a feed-forward mechanism with target mRNAs, but instead amplifies siRNAs [124-126].

SMALL RNAS IN THE GERMLINE OF C. ELEGANS

Much of the interesting and complex small RNA biology occurs in the germ cells of *C. elegans.* siRNAs and piRNAs interact with target mRNAs at germ granules as they are exported into the cytoplasm from nuclear pores (Figure 1.5). In *C. elegans* these germ granules are called P granules and contain many RNA regulatory proteins, including Argonautes and associated small RNAs, suggesting they might play a role in mRNA surveillance [127]. Adjacent to P granules are Mutator foci, which contain the Mutator protein complex that produces so-called secondary siRNAs, distinct from the canonical biogenesis pathway of primary siRNAs [128, 129]. piRNAs, in association with the PIWI protein PRG-1 and primary siRNAs produced from exogenous dsRNA, in association with the Argonaute RDE-1, trigger production of secondary siRNAs and serve to amplify the small RNA signal [43, 124, 125, 130-135]. The mRNA is guided to the Mutator

complex and used as a template, producing 22G siRNAs via RNA-dependent RNA polymerases (RDRPs) [43, 125, 126, 134, 136, 137]. The Argonaute HRDE-1/WAGO-9 binds to a subset of these secondary siRNAs and intercepts target transcripts in the nucleus, which through an unknown mechanism leads to H3K9 methylation and transcriptional gene silencing (Figure 1.5) [130, 133, 138].

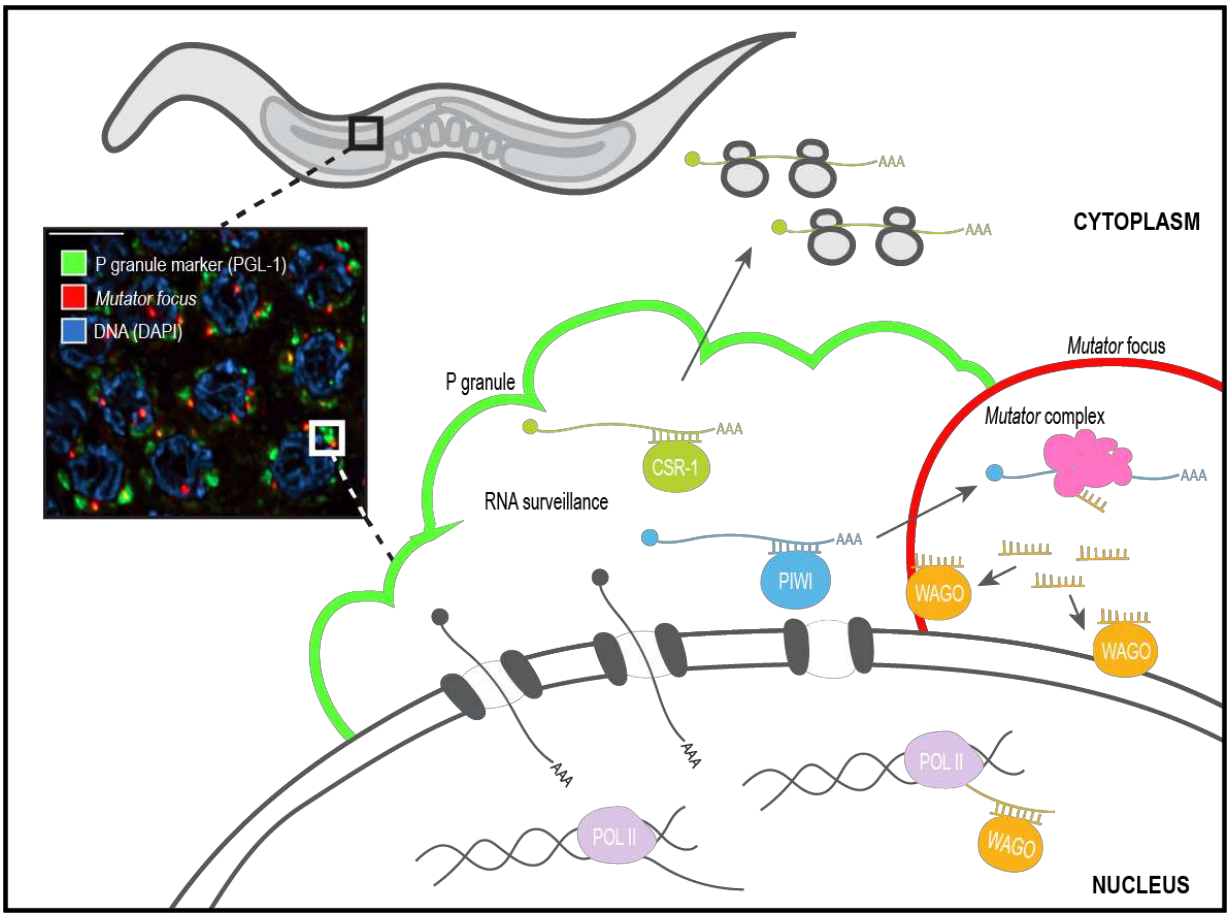


Figure 1.5 Small RNA biology of P granules. The various known small RNA pathways that act as an RNA surveillance mechanism of the germ cells of *C. elegans.* This figure was adapted from [129] and [139][2].

[2] This figure was also adapted from a presentation slide created by Tai Montgomery

Early studies exploring the mechanism of RNAi in worms revealed that gene silencing is sometimes transmitted from the parent receiving RNAi treatment to the untreated progeny [117]. RNAi directed against GFP transgenes, as well as endogenous genes, can persist for multiple generations before eventually petering out [140]. Because of their sequence specificity and ability to be transmitted from parent to progeny, small RNAs are attractive candidates for maintaining a cellular memory of environmentally-induced gene silencing from one generation to the next [126, 141]. Silencing of certain viruses and transgenes in *C. elegans* involves a small RNA signal that is epigenetic in nature and is heritably maintained for multiple generations [130, 133, 142, 143]. Interestingly, although loss of piRNAs in some animals results in immediate sterility, in worms bearing a mutation in *prg-1,* sterility occurs after several generations [144]. It is possible that misexpression of repetitive elements and mutations caused by transposons gradually accumulate in the absence of the piRNA pathway before reaching some threshold that is no longer tolerated. Interestingly, re-establishing endogenous RNAi in worms in which both the piRNA and endogenous RNAi pathways have been disabled causes immediate HRDE-1-dependent sterility. The cause of this sterility likely stems from aberrant silencing of essential genes by HRDE-1 when piRNAs are unavailable to provide the initial target information. Thus, secondary siRNAs seem to provide a “memory” of piRNA targets over generations [139, 145].

1.1.4 HIGH-THROUGHPUT SEQUENCING OF SMALL RNAS

In order to quantify the expression of small RNAs in an organism or tissue type, we perform next-generation sequencing. There are several key differences from the standard RNA-seq methods that need to be taken into account for both the preparation of sequencing libraries (Figure 1.6) and the analysis of resulting data. The biggest difference is that small RNA sequencing typically involves one or more size selection steps to make sure that a range of 15 – 36 nt sequences are captured. The sequencer read length is typically 50bp, meaning there will be adapters attached to true small RNAs in the resulting data. Reads without an adapter are longer than what was intended to be captured. Reads are also not reverse transcribed based on random hexamer priming, but by ligation to adapters at 5' and 3' ends. However, this can result in their own set of biases [146, 147]. Additionally, because of the size, we do not need to fragment small RNAs and their sequences are likely to be duplicated in the data. There is no method of de-duplicating the data as there are for mRNA sequencing, short of unique molecular identifier (UMI) sequencing [148]. When aligning the reads to a genome, small RNAs are also expected to map to multiple locations, but with mRNA sequencing these reads are often discarded. A more detailed description of library preparation is in Chapter 2.2 and the analysis in Chapter 3.

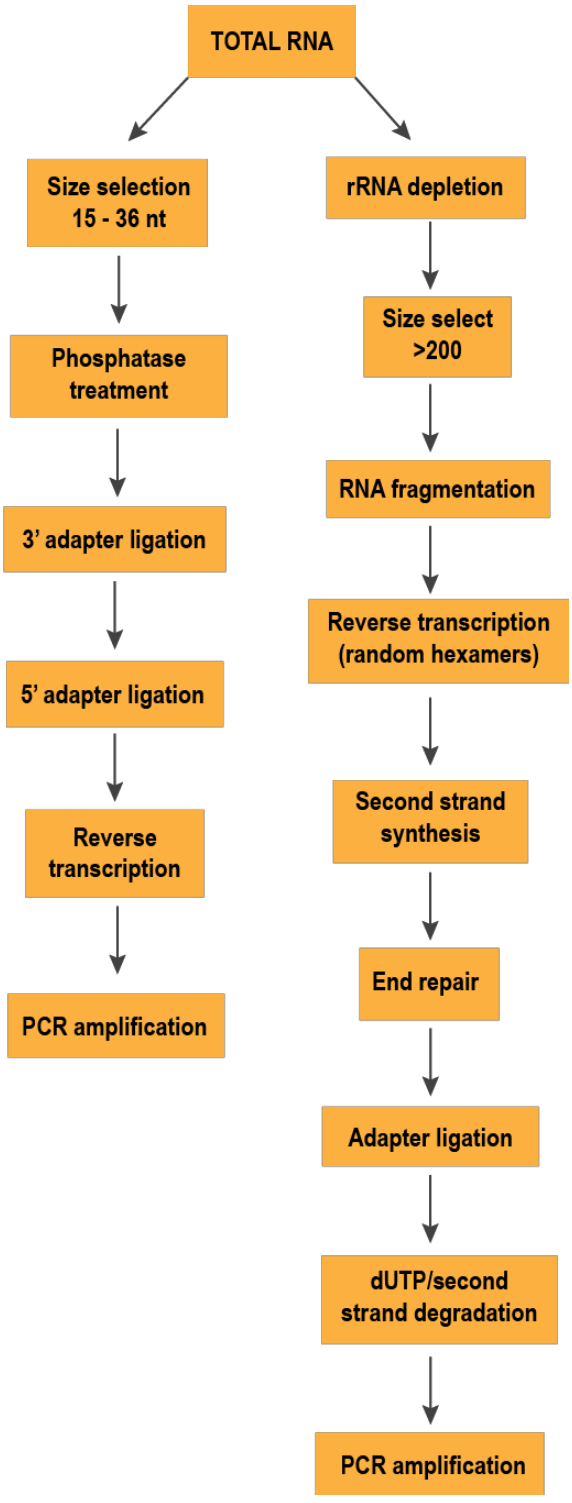


Figure 1.6 sRNA and mRNA sequencing library preparation. A standard library preparation workflow for small RNA sequencing (left) and mRNA sequencing (right). Both protocols and mRNA sequencing data analysis are described in detail in Chapter 2.2. Small RNA sequencing data analysis is described in Chapter 3, also see Figure 3.1.

1.2 C. ELEGANS AS A MODEL ORGANISM

C. elegans is a popular model organism across fields that has led to many important discoveries, including that of small RNAs [149] and important genes in aging [150] and apoptosis [151]. It is a powerful model organism because of its short generation time, genetic tractability, clear body, and a wealth of resources through the worm community (http://www.wormbase.org/). In the lab setting, worms eat *E. coli* and live in controlled environments at 15 – 25˚C [152]. Typically, enhanced phenotypes are observed at 25˚C, with normal growth and behavior occurring at 20˚C. The worm life cycle begins at the embryo stage, moves through four distinct larval stages (L1 – L4), and ends with adulthood. At 20˚C, this cycle occurs over 3 days, after which the worms reproduce over another 3-4 days, and then live for an additional 10 – 15 days. *C. elegans* is a hermaphroditic species, producing both sperm and oocytes for self-fertilization, so this cycle is fairly consistent in wild type animals. While males do exist in the population, they are rare [152].

GERMLINE DEVELOPMENT

Much of the interesting small RNA activity of the worm exists in the germline (Figure 1.6) and thus, many of the phenotypes resulting from disrupting this activity results in germline defects. Hermaphrodites develop two gonadal arms (Figure 1.8) and males only develop one [153, 154]. The germline is an immortal lineage, beginning with the P cell in embryos. The L1 stage has 4 germ cells, two of which are the precursor to the somatic gonad tissue. The germ cells proliferate rapidly over the next larval stages into

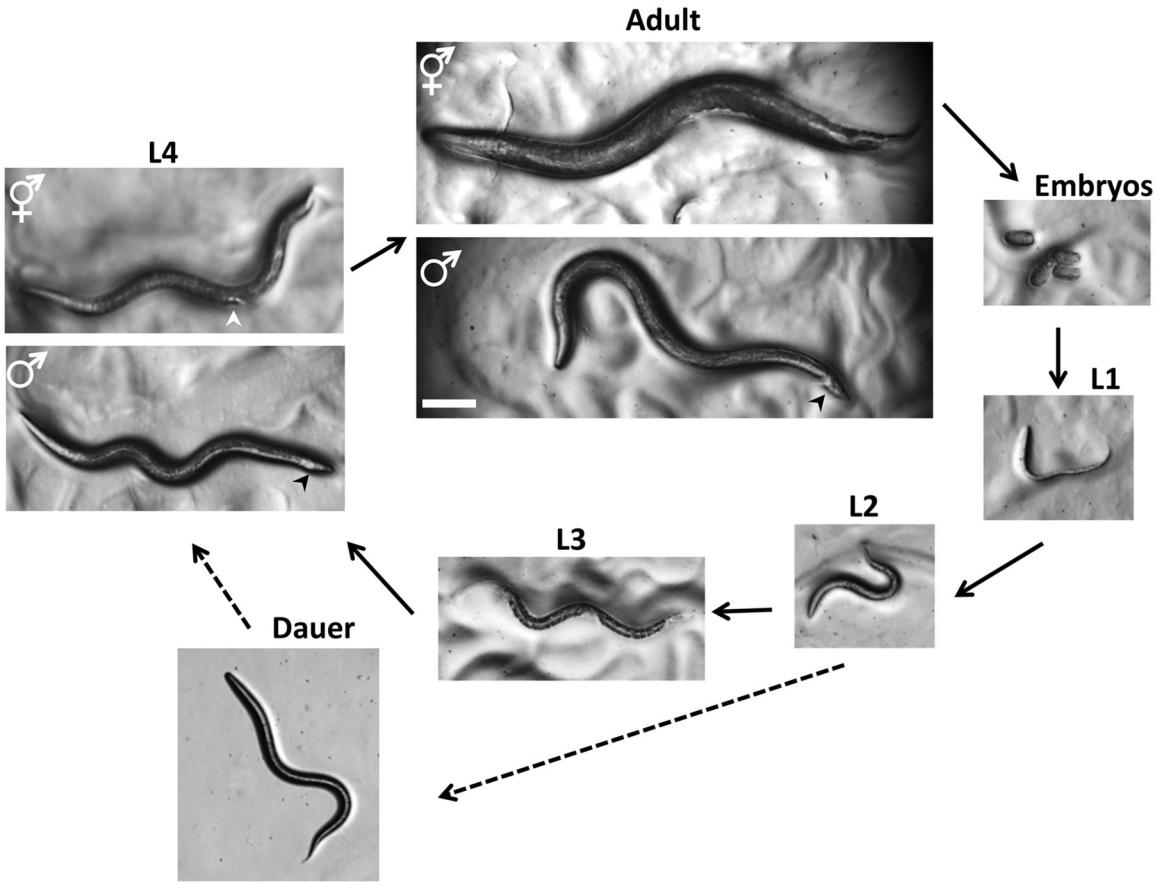


Figure 1.7 The C. elegans life cycle. Adult *C. elegans* lay embryos during their fertile period, which hatch into the first larval stage (L1). *C. elegans* undergoes 4 distinct larval molts (L1-L4) prior to a molt into fertile adulthood. Males can be distinguished from hermaphrodites by the tail structure (as indicated by the arrow in L4 and Adult male photos). A specialized larval stage, Dauer, occurs after L2 in unfavorable environments, such as starvation and allows the worms to survive for several months. Reprinted from WormBook [152, 155] under the Creative Commons License.

adulthood. At the L3 stage the worms begin producing mature sperm cells, which are then stored in the spermatheca, just at the proximal tip of the gonad [153]. The sperm produced at this time is the limiting factor in progeny production when there are no males to fertilize the hermaphrodites [156]. After the sperm is produced, the germ cells switch to the production of oocytes during the L4 stage, which continues through adulthood [153].

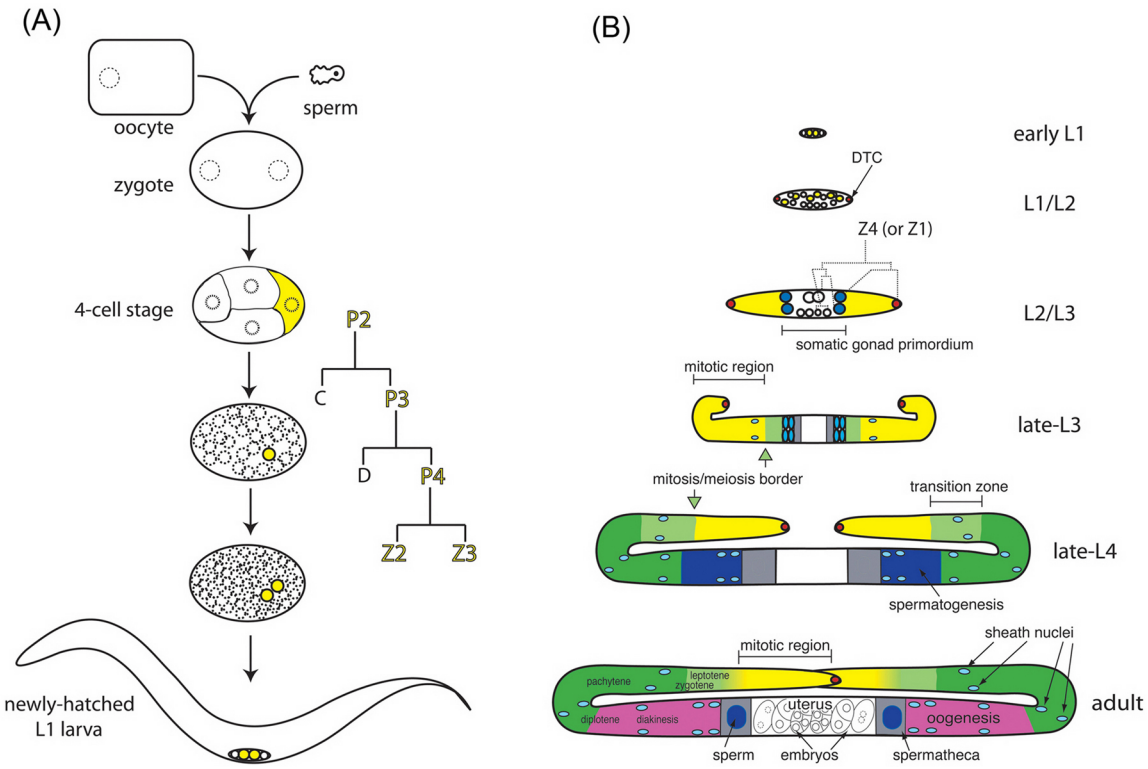


Figure 1.8 Germline development in *C. elegans.* A) Tracing the embryonic germline post-fertilization. Germline lineage is highlighted in yellow. B) The post-embryonic development of the germline. DTC: distal tip cells. The relative sizes are not to scale. Reprinted from WormBook [153] under the Creative Commons License.

1.3 REPRODUCIBLE RESEARCH IN COMPUTATIONAL BIOLOGY

The growth of quantitative and computational methods in the sciences stems from the need to analyze or simulate large amounts of data. Bioinformatics and computational biology are fields that apply statistical and computational approaches to biological problems [157]. In experimental biology, there has been an explosion of the production of next-generation sequencing datasets over the last decade. The field has since focused on analyzing the ever-growing amount of sequencing data being produced

[157-160]. Unfortunately, reproducibility has been difficult to manage or not even considered a concern for those new to the field. After all, computer software is not finicky, random, and prone to error, like experimental biology, right? In practice, as the software and analysis was done by humans, there are bound to be errors and nuances of performing analysis that might not be obvious [161, 162]. With the growing concern for reproducibility in science [160, 163], how can computational sciences be more reproducible?

Much of the methodology for analyzing sequencing data can be difficult to approach without a basic background in computational biology and next-generation sequencing [164-167]. Despite growing popularity and demand for bioinformatics skillsets, there is still a shortage of properly trained scientists for developing methods and software for the wider community. A lot of this has to do with the highly interdisciplinary nature of bioinformatics and the lack of formal training as the field grew through self-taught scientists. More recently, universities have been offering coursework and training in applied bioinformatics to train biologists in data analysis and computational skills. Currently the education in computational biology is sparse, mainly limited to graduate students, and sometimes not effective. The knowledge required to succeed in the field rapidly evolves and there can be disagreement among scientists about what core skills should be taught to biologists. However, even introductory coursework and basic skills can help improve bench scientists' ability to work with and utilize tools created by computational scientists [164-167].

BEST PRACTICES FOR BIOINFORMATICS SOFTWARE

As bioinformatics training focuses more on basic competencies and analyzing field-specific data, less focus is given to reproducibility and software engineering practices [168, 169]. This is likely due to the lack of developed training curriculum, the highly specific nature of research questions requiring computational methods, the lack of reward for robust software over results, and the amount of time that needs to be invested on top of solving the scientific problems at hand. Sometimes best practices and reproducibility are taken into account once there is a great need to reuse code, but often scientists will, understandably, prioritize making progress on the biological problem and results over spending time on the performance and engineering aspects of coding [168, 169]. At minimum, there is a growing interest in reproducibility and sharing data analysis code on platforms such as GitHub. However, not all journals require authors to share code associated with analysis and even when they do, it's often simply to denote the code is "available upon request", the effectiveness of which is insufficient [170]. Many authors cite not wanting to share "bad" code with judging eyes, despite the fact that community feedback can lead to improved code and the justification is absurd in the context of other fields [171, 172]. This habit can unfortunately lead to confusion and a lack of transparency around how specific analyses were performed and reduce confidence in the results. Errors may be present in the code and one might never know without code reviews and testing principles common in software engineering, but not in scientific computing. And when errors happen in scientific code, they can lead to false results, ultimately doing more harm than good [173]. Taking a software-engineering

guided approach from the beginning benefits the science, programmers, maintainers, and users [168, 169].

What constitutes good software engineering practices for bioinformatics? In an ideal world, every scientist doing computational work would follow all the standard software engineering best practices, work would be robust and no one would need to reinvent the wheel to repeat analyses. This becomes difficult in practice with a growing number of scientists relying on computational methods without enough time or resources for formal training in development methodologies. There are many established practices out there, but researchers often don't know where to start. Giving too many or too difficult practices to implement might scare off early-stage programmers. Outlined below are a few practices that seem "good enough" to balance best practices with limited time based on a few sources that make for good further reading on the topic, as the details of applying these practices are much too long for this introduction [168, 169, 174-180], but this list should provide a starting point to think on.

1. Publish the code online. If the code was good enough to produce publishable results, it's good enough to post online and should be subject to peer-review. Popular resources include posting to GitHub, with a repository for each publication, for example.
2. Project planning. Many issues with using code after the creation could be alleviated with some amount of planning the project beyond what problem it should solve. Questions to ask your group ahead of the project: How to organize the data, code,

model files, etc? What are the input and outputs of the workflow? How will someone else use this code in the future? How will we know the code is correct?

3. Version control. To keep a history of the changes made to code, version control systems like git are very useful and can be easily integrated with GitHub for sharing code. The main advantages of using version control is the ability to track all changes, revert bad ones, and to collaborate on the same codebase with many developers.

4. Documentation. There are two types of documentation important for code – one for users and one for developers. The user documentation describes how someone runs your code on their own machine. The developer documentation lives in the code and describes the code well enough for someone to take over or modify the code. Documentation is probably the single greatest influence on how often the code will be used vs. recreated by others.

By investing a little time in these practices, and more thorough ones discussed in [168, 169, 174-180], we can begin to make sure that as the field grows, it grows in a sustainable and robust way. However, without proper education in these areas, it will continue to be an insurmountable hurdle for many scientists, especially those new to programming in biology. These practices should be taught as a part of data analysis and bioinformatics courses and workshops.

Another component of reproducible scientific research is the idea of data provenance [181] – how exactly was this data produced? Data provenance in science is typically

good enough when the data can be reproduced using the methods provided. However, small changes in versions, parameters, random seeds, or state of the initial raw data can influence how reproducible the data are. Implementation of provenance tracking generally consists of explicitly recording the initial and final states of data through file checksums and the entire flow of data through the analysis performed with all associated metadata [181]. A simple method is to produce log files with the software each time it is run, but some data analysis tools, such as workflow engines, implement provenance tracking for the user in a more robust and complete way [182].

WORKFLOW ENGINES

With the increasing amount of biological data being produced and analyses becoming more standardized, scientific workflow languages emerged. A scientific workflow typically defines an entire analysis step-by-step such that another researcher can run the workflow using the same settings [183, 184]. Workflows can often be described as directed acyclic graphs (DAGs) to simplify the organization of the data flow. Workflow engines provide a platform for reproducible data analysis pipelines and are becoming heavily adopted in bioinformatics [183, 184]. Another layer of reproducibility is the adoption of Docker (https://www.docker.com/) or Singularity [185] containers that are similar to virtual machines, but take up far fewer resources. Containers significantly improve reproducibility in data analysis across domains through explicitly defining an environment in which the software is known to work, separate from other installations on the operating system (OS), thus reducing cross-platform issues. For bioinformatics

workflows, the cost of using a container is negligible [186]. A maximally reproducible and portable bioinformatics analysis would likely combine workflow languages and containers. Additional advantages of both include the simplicity of running tools on HPC clusters or in the cloud distributed across many samples and without having to install the dependencies explicitly [187].

There are now over 200 workflow engines/systems/languages across fields and industries, including many specifically for bioinformatics (https://s.apache.org/existing-workflow-systems). Several workflow systems became quite popular in bioinformatics over the last few years, including Snakemake [188], Nextflow [189], and WDL [190]. The issue with this is that there are now so many different ways to define workflows for bioinformatics that there are issues with sharing those workflows among everyone. Each of the implementations mentioned contains their own syntax and method to run a pipeline, a standard feature of a workflow engine [188-190]. When a researcher decides they would like to share their Nextflow pipeline, it's great for other researchers that either also use Nextflow or are able to have it installed on their system. However, if they instead use WDL, and their cluster only has Cromwell (the engine for running WDL), they are suddenly unable to use the Nextflow workflow. Generally speaking, issues with installation of the engine are simple to overcome, but there may be other reasons institutions avoid certain implementations (such as being unable to use GPL licensed software in their own workflows, which Nextflow was until very recently). Additionally, if the user wants to share only a portion of that workflow or wishes to contribute new

steps, they either have to choose learning an entirely new framework or rewriting the whole pipeline in the framework they already know. This is where the inconsistency among workflow language adoption becomes more of a problem and can interfere with collaboration.

In response to this, a standard is emerging called the Common Workflow Language (CWL) [191]. CWL aims to be implementation-agnostic, interoperable, and explicitly written such that it is simpler to create programmatically. This creates a less human-friendly framework, but allows developers of implementations to create a converter from CWL to their own workflow language or for their software to easily interpret and run CWL scripts. By maximizing interoperability, CWL can help alleviate this roadblock to collaboration among scientific workflows. Many implementations have already adopted CWL and have made great strides toward supporting it. In this way, implementations can still use CWL workflows while using their own definitions of “best practices” in running those implementations. Some implementations like CWLEXEC (https://github.com/IBMSpectrumComputing/cwlexec) only run CWL, but utilize the best practices for running workflows on an LSF cluster as defined by the developers of LSF at IBM. CWL also allows users to more easily upload their workflows onto cloud platforms such as DNANexus, which has their own app-based workflow system, but have created a CWL-to-DNANexus tool (https://github.com/dnanexus/dx-cwl). Nextflow, SnakeMake and Cromwell all have some capability of running CWL scripts and thus, creating tool definitions in CWL will likely be the most interoperable long-term option.

With software engineering best practices, workflow languages, and containers, biologists can also create robust and reproducible software for their analysis that can be used by many for years to come. Initial time investments in education and implementation may seem large, but the long-term impact on reproducibility, reducing future time spent fixing it, and the science performed makes it worthwhile.

REFERENCES

1. Carrington, J.C. and V. Ambros, *Role of MicroRNAs in Plant and Animal Development.* Science, 2003. **301**(5631): p. 336.
2. Kloosterman, W.P. and R.H. Plasterk, *The diverse functions of microRNAs in animal development and disease.* Dev Cell, 2006. **11**(4): p. 441-50.
3. Zhang, C., *Novel functions for small RNA molecules.* Current opinion in molecular therapeutics, 2009. **11**(6): p. 641-651.
4. Alvarez-Garcia, I. and E.A. Miska, *MicroRNA functions in animal development and human disease.* Development, 2005. **132**(21): p. 4653.
5. Bhaskaran, M. and M. Mohan, *MicroRNAs: history, biogenesis, and their evolving role in animal development and disease.* Veterinary pathology, 2014. **51**(4): p. 759-774.
6. Garofalo, M., G. Condorelli, and C.M. Croce, *MicroRNAs in diseases and drug response.* Curr Opin Pharmacol, 2008. **8**(5): p. 661-7.
7. Romano, G., et al., *Small non-coding RNA and cancer.* Carcinogenesis, 2017. **38**(5): p. 485-491.
8. Hébert, S.S. and B. De Strooper, *Alterations of the microRNA network cause neurodegenerative disease.* Trends in Neurosciences, 2009. **32**(4): p. 199-206.
9. Ma, L. and R.A. Weinberg, *Micromanagers of malignancy: role of microRNAs in regulating metastasis.* Trends in Genetics, 2008. **24**(9): p. 448-456.
10. Feng, J., W. Xing, and L. Xie, *Regulatory Roles of MicroRNAs in Diabetes.* International journal of molecular sciences, 2016. **17**(10): p. 1729.
11. D'Ario, M., S. Griffiths-Jones, and M. Kim, *Small RNAs: Big Impact on Plant Development.* Trends in Plant Science, 2017. **22**(12): p. 1056-1068.
12. Li, S., et al., *The functions of plant small RNAs in development and in stress responses.* The Plant Journal, 2017. **90**(4): p. 654-670.
13. Chen, C., et al., *Small RNAs, emerging regulators critical for the development of horticultural traits.* Horticulture Research, 2018. **5**(1): p. 63.
14. Guleria, P., et al., *Plant Small RNAs: Biogenesis, Mode of Action and Their Roles in Abiotic Stresses.* Genomics, Proteomics & Bioinformatics, 2011. **9**(6): p. 183-199.
15. Pratt, A.J. and I.J. MacRae, *The RNA-induced silencing complex: a versatile gene-silencing machine.* The Journal of biological chemistry, 2009. **284**(27): p. 17897-17901.
16. Rivas, F.V., et al., *Purified Argonaute2 and an siRNA form recombinant human RISC.* Nature Structural &Amp; Molecular Biology, 2005. **12**: p. 340.
17. Czech, B. and G.J. Hannon, *Small RNA sorting: matchmaking for Argonautes.* Nat Rev Genet, 2011. **12**(1): p. 19-31.
18. Rana, T.M., *Illuminating the silence: understanding the structure and function of small RNAs.* Nature Reviews Molecular Cell Biology, 2007. **8**: p. 23.
19. Carthew, R.W. and E.J. Sontheimer, *Origins and Mechanisms of miRNAs and siRNAs.* Cell, 2009. **136**(4): p. 642-655.
20. Wu, L. and J.G. Belasco, *Let me count the ways: mechanisms of gene regulation by miRNAs and siRNAs.* Mol Cell, 2008. **29**(1): p. 1-7.
21. Burnett, J.C. and J.J. Rossi, *RNA-based therapeutics: current progress and future prospects.* Chemistry & biology, 2012. **19**(1): p. 60-71.

22. Chakraborty, C., et al., *Therapeutic miRNA and siRNA: Moving from Bench to Clinic as Next Generation Medicine.* Molecular Therapy - Nucleic Acids, 2017. **8**: p. 132-143.
23. Kamthan, A., et al., *Small RNAs in plants: recent development and application for crop improvement.* Frontiers in plant science, 2015. **6**: p. 208-208.
24. Zhang, B. and Q. Wang, *MicroRNA, a new target for engineering new crop cultivars.* Bioengineered, 2016. **7**(1): p. 7-10.
25. Ghildiyal, M. and P.D. Zamore, *Small silencing RNAs: an expanding universe.* Nat Rev Genet, 2009. **10**(2): p. 94-108.
26. Kim, V.N., *Small RNAs: Classification, Biogenesis, and Function.* Mol. Cells, 2005. **19**(1): p. 1-15.
27. Lee, R., R. Feinbaum, and V. Ambros, *A short history of a short RNA.* Cell, 2004. **116**(2 Suppl): p. S89-92, 1 p following S96.
28. Axtell, M.J., J.O. Westholm, and E.C. Lai, *Vive la différence: biogenesis and evolution of microRNAs in plants and animals.* Genome Biology, 2011. **12**(4): p. 221.
29. Malone, C.D. and G.J. Hannon, *Small RNAs as guardians of the genome.* Cell, 2009. **136**(4): p. 656-68.
30. Bohmert, K., et al., *AGO1 defines a novel locus of Arabidopsis controlling leaf development.* The EMBO journal, 1998. **17**(1): p. 170-180.
31. Jonas, S. and E. Izaurralde, *Towards a molecular understanding of microRNA-mediated gene silencing.* Nat Rev Genet, 2015. **16**(7): p. 421-33.
32. Höck, J. and G. Meister, *The Argonaute protein family.* Genome biology, 2008. **9**(2): p. 210-210.
33. Cenik, E.S. and P.D. Zamore, *Argonaute proteins.* Curr Biol, 2011. **21**(12): p. R446-9.
34. Mi, S., et al., *Sorting of small RNAs into Arabidopsis argonaute complexes is directed by the 5′ terminal nucleotide.* Cell, 2008. **133**(1): p. 116-27.
35. Tomari, Y., T. Du, and P.D. Zamore, *Sorting of Drosophila small silencing RNAs.* Cell, 2007. **130**(2): p. 299-308.
36. Ghildiyal, M., et al., *Sorting of Drosophila small silencing RNAs partitions microRNA* strands into the RNA interference pathway.* RNA, 2010. **16**(1): p. 43-56.
37. Goh, E. and K. Okamura, *Hidden sequence specificity in loading of single-stranded RNAs onto Drosophila Argonautes.* Nucleic Acids Research, 2018. **47**(6): p. 3101-3116.
38. Frank, F., N. Sonenberg, and B. Nagar, *Structural basis for 5′-nucleotide base-specific recognition of guide RNA by human AGO2.* Nature, 2010. **465**(7299): p. 818-22.
39. Winter, J. and S. Diederichs, *Argonaute proteins regulate microRNA stability: Increased microRNA abundance by Argonaute proteins is due to microRNA stabilization.* RNA Biology, 2011. **8**(6): p. 1149-1157.
40. Schirle, N.T. and I.J. MacRae, *The crystal structure of human Argonaute2.* Science, 2012. **336**(6084): p. 1037-40.
41. Montgomery, T.A., et al., *PIWI associated siRNAs and piRNAs specifically require the Caenorhabditis elegans HEN1 ortholog henn-1.* PLoS Genet, 2012. **8**(4): p. e1002616.
42. Tolia, N.H. and L. Joshua-Tor, *Slicer and the argonautes.* Nat Chem Biol, 2007. **3**(1): p. 36-43.
43. Yigit, E., et al., *Analysis of the C. elegans Argonaute family reveals that distinct Argonautes act sequentially during RNAi.* Cell, 2006. **127**(4): p. 747-57.

44. Almeida, V.M., A.M. Andrade-Navarro, and F.R. Ketting, *Function and Evolution of Nematode RNAi Pathways.* Non-Coding RNA, 2019. **5**(1).
45. Hutvagner, G. and M.J. Simard, *Argonaute proteins: key players in RNA silencing.* Nat Rev Mol Cell Biol, 2008. **9**(1): p. 22-32.
46. Youngman, E.M. and J.M. Claycomb, *From early lessons to new frontiers: the worm as a treasure trove of small RNA biology.* Frontiers in genetics, 2014. **5**: p. 416-416.
47. Vasquez-Rifo, A., et al., *Developmental characterization of the microRNA-specific C. elegans Argonautes alg-1 and alg-2.* PLoS One, 2012. **7**(3): p. e33750.
48. Grishok, A., et al., *Genes and mechanisms related to RNA interference regulate expression of the small temporal RNAs that control C. elegans developmental timing.* Cell, 2001. **106**(1): p. 23-34.
49. Conine, C.C., et al., *Argonautes ALG-3 and ALG-4 are required for spermatogenesis-specific 26G-RNAs and thermotolerant sperm in Caenorhabditis elegans.* Proc Natl Acad Sci U S A, 2010. **107**(8): p. 3588-93.
50. Kao, C.Y., et al., *Global functional analyses of cellular responses to pore-forming toxins.* PLoS Pathog, 2011. **7**(3): p. e1001314.
51. Lee, R.C., R.L. Feinbaum, and V. Ambros, *The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14.* Cell, 1993. **75**(5): p. 843-54.
52. Wightman, B., I. Ha, and G. Ruvkun, *Posttranscriptional regulation of the heterochronic gene lin-14 by lin-4 mediates temporal pattern formation in C. elegans.* Cell, 1993. **75**(5): p. 855-62.
53. Reinhart, B.J., et al., *The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans.* Nature, 2000. **403**(6772): p. 901-6.
54. Pasquinelli, A.E., et al., *Conservation of the sequence and temporal expression of let-7 heterochronic regulatory RNA.* Nature, 2000. **408**(6808): p. 86-9.
55. Corlan, A.D. *Medline trend: automated yearly statistics of PubMed results for any query*. 2004 2019-02-20]; Available from: http://dan.corlan.net/medline-trend.html.
56. Lee, Y., et al., *MicroRNA maturation: stepwise processing and subcellular localization.* The EMBO journal, 2002. **21**(17): p. 4663-4670.
57. Lee, Y., et al., *MicroRNA genes are transcribed by RNA polymerase II.* The EMBO journal, 2004. **23**(20): p. 4051-4060.
58. Lim, L.P., et al., *The microRNAs of Caenorhabditis elegans.* Genes Dev, 2003. **17**(8): p. 991-1008.
59. Lee, Y., et al., *The nuclear RNase III Drosha initiates microRNA processing.* Nature, 2003. **425**(6956): p. 415-9.
60. Denli, A.M., et al., *Processing of primary microRNAs by the Microprocessor complex.* Nature, 2004. **432**(7014): p. 231-235.
61. Han, J., et al., *Molecular basis for the recognition of primary microRNAs by the Drosha-DGCR8 complex.* Cell, 2006. **125**(5): p. 887-901.
62. Bohnsack, M.T., K. Czaplinski, and D. Gorlich, *Exportin 5 is a RanGTP-dependent dsRNA-binding protein that mediates nuclear export of pre-miRNAs.* RNA (New York, N.Y.), 2004. **10**(2): p. 185-191.
63. Lund, E., et al., *Nuclear Export of MicroRNA Precursors.* Science, 2004. **303**(5654): p. 95.

64. Yi, R., et al., *Exportin-5 mediates the nuclear export of pre-microRNAs and short hairpin RNAs.* Genes & development, 2003. **17**(24): p. 3011-3016.
65. Hutvagner, G., et al., *A cellular function for the RNA-interference enzyme Dicer in the maturation of the let-7 small temporal RNA.* Science, 2001. **293**(5531): p. 834-8.
66. Ketting, R.F., et al., *Dicer functions in RNA interference and in synthesis of small RNA involved in developmental timing in C. elegans.* Genes Dev, 2001. **15**(20): p. 2654-9.
67. Borchert, G.M., W. Lanier, and B.L. Davidson, *RNA polymerase III transcribes human microRNAs.* Nature Structural &Amp; Molecular Biology, 2006. **13**: p. 1097.
68. Canella, D., et al., *Defining the RNA polymerase III transcriptome: Genome-wide localization of the RNA polymerase III transcription machinery in human cells.* Genome Research, 2010. **20**(6): p. 710-721.
69. Khvorova, A., A. Reynolds, and S.D. Jayasena, *Functional siRNAs and miRNAs exhibit strand bias.* Cell, 2003. **115**(2): p. 209-16.
70. Schwarz, D.S., et al., *Asymmetry in the assembly of the RNAi enzyme complex.* Cell, 2003. **115**(2): p. 199-208.
71. Liu, J., et al., *Argonaute2 is the catalytic engine of mammalian RNAi.* Science, 2004. **305**(5689): p. 1437-41.
72. Bartel, D.P., *MicroRNAs: target recognition and regulatory functions.* Cell, 2009. **136**(2): p. 215-33.
73. Okamura, K., et al., *The regulatory activity of microRNA* species has substantial influence on microRNA and 3' UTR evolution.* Nat Struct Mol Biol, 2008. **15**(4): p. 354-63.
74. Okamura, K., et al., *The Drosophila hairpin RNA pathway generates endogenous short interfering RNAs.* Nature, 2008. **453**(7196): p. 803-6.
75. Yang, J.S., et al., *Widespread regulatory activity of vertebrate microRNA* species.* RNA, 2011. **17**(2): p. 312-26.
76. Griffiths‐Jones, S., et al., *MicroRNA evolution by arm switching.* EMBO reports, 2011. **12**(2): p. 172.
77. Guo, L. and Z. Lu, *The Fate of miRNA* Strand through Evolutionary Analysis: Implication for Degradation As Merely Carrier Strand or Potential Regulatory Molecule?* PLOS ONE, 2010. **5**(6): p. e11387.
78. Desvignes, T., et al., *miRNA Nomenclature: A View Incorporating Genetic Origins, Biosynthetic Pathways, and Sequence Variants.* Trends in genetics : TIG, 2015. **31**(11): p. 613-626.
79. Kozomara, A. and S. Griffiths-Jones, *miRBase: annotating high confidence microRNAs using deep sequencing data.* Nucleic acids research, 2014. **42**(Database issue): p. D68-D73.
80. Iwakawa, H.-o. and Y. Tomari, *The Functions of MicroRNAs: mRNA Decay and Translational Repression.* Trends in Cell Biology, 2015. **25**(11): p. 651-665.
81. Djuranovic, S., A. Nahvi, and R. Green, *miRNA-mediated gene silencing by translational repression followed by mRNA deadenylation and decay.* Science, 2012. **336**(6078): p. 237-40.
82. Eichhorn, Stephen W., et al., *mRNA Destabilization Is the Dominant Effect of Mammalian MicroRNAs by the Time Substantial Repression Ensues.* Molecular Cell, 2014. **56**(1): p. 104-115.

83. Hendrickson, D.G., et al., *Concordant Regulation of Translation and mRNA Abundance for Hundreds of Targets of a Human microRNA.* PLOS Biology, 2009. **7**(11): p. e1000238.
84. Guo, H., et al., *Mammalian microRNAs predominantly act to decrease target mRNA levels.* Nature, 2010. **466**: p. 835.
85. Siomi, M.C., et al., *PIWI-interacting small RNAs: the vanguard of genome defence.* Nat Rev Mol Cell Biol, 2011. **12**(4): p. 246-58.
86. Lewis, B.P., et al., *Prediction of Mammalian MicroRNA Targets.* Cell, 2003. **115**(7): p. 787-798.
87. Pinzón, N., et al., *microRNA target prediction programs predict many false positives.* Genome research, 2017. **27**(2): p. 234-245.
88. Grimson, A., et al., *MicroRNA targeting specificity in mammals: determinants beyond seed pairing.* Mol Cell, 2007. **27**(1): p. 91-105.
89. Broughton, James P., et al., *Pairing beyond the Seed Supports MicroRNA Targeting Specificity.* Molecular Cell, 2016. **64**(2): p. 320-333.
90. Chipman, L.B. and A.E. Pasquinelli, *miRNA Targeting: Growing beyond the Seed.* Trends in Genetics, 2019. **35**(3): p. 215-222.
91. Tops, B.B., R.H. Plasterk, and R.F. Ketting, *The Caenorhabditis elegans Argonautes ALG-1 and ALG-2: almost identical yet different.* Cold Spring Harb Symp Quant Biol, 2006. **71**: p. 189-94.
92. Aalto, A.P., et al., *Opposing roles of microRNA Argonautes during Caenorhabditis elegans aging.* PLoS genetics, 2018. **14**(6): p. e1007379-e1007379.
93. Inukai, S., et al., *A microRNA feedback loop regulates global microRNA abundance during aging.* RNA (New York, N.Y.), 2018. **24**(2): p. 159-172.
94. Tabara, H., et al., *The rde-1 gene, RNA interference, and transposon silencing in C. elegans.* Cell, 1999. **99**(2): p. 123-32.
95. Correa, R.L., et al., *MicroRNA-directed siRNA biogenesis in Caenorhabditis elegans.* PLoS Genet, 2010. **6**(4): p. e1000903.
96. Miska, E.A., et al., *Most Caenorhabditis elegans microRNAs are individually not essential for development or viability.* PLoS Genet, 2007. **3**(12): p. e215.
97. Alvarez-Saavedra, E. and H.R. Horvitz, *Many families of C. elegans microRNAs are not essential for development or viability.* Curr Biol, 2010. **20**(4): p. 367-73.
98. Brenner, J.L., et al., *Loss of Individual MicroRNAs Causes Mutant Phenotypes in Sensitized Genetic Backgrounds in C. elegans.* Current Biology, 2010. **20**(14): p. 1321-1325.
99. Inukai, S., et al., *A microRNA feedback loop regulates global microRNA abundance during aging.* RNA, 2018. **24**(2): p. 159-172.
100. Ibanez-Ventoso, C., et al., *Modulated microRNA expression during adult lifespan in Caenorhabditis elegans.* Aging Cell, 2006. **5**(3): p. 235-46.
101. de Lencastre, A., et al., *MicroRNAs both promote and antagonize longevity in C. elegans.* Curr Biol, 2010. **20**(24): p. 2159-68.
102. Noren Hooten, N., et al., *microRNA expression patterns reveal differential expression of target genes with age.* PLoS One, 2010. **5**(5): p. e10724.
103. Inukai, S., et al., *Novel microRNAs differentially expressed during aging in the mouse brain.* PLoS One, 2012. **7**(7): p. e40028.

104. Kato, M., et al., *Age-associated changes in expression of small, noncoding RNAs, including microRNAs, in C. elegans.* RNA, 2011. **17**(10): p. 1804-20.
105. Sun, K. and E.C. Lai, *Adult-specific functions of animal microRNAs.* Nature reviews. Genetics, 2013. **14**(8): p. 535-548.
106. Boulias, K. and H.R. Horvitz, *The C. elegans microRNA mir-71 acts in neurons to promote germline-mediated longevity through regulation of DAF-16/FOXO.* Cell metabolism, 2012. **15**(4): p. 439-450.
107. Nehammer, C., et al., *Specific microRNAs Regulate Heat Stress Responses in Caenorhabditis elegans.* Scientific Reports, 2015. **5**: p. 8866.
108. Ren, Z. and V.R. Ambros, *Caenorhabditis elegans microRNAs of the let-7 family act in innate immune response circuits and confer robust developmental timing against pathogen stress.* Proc Natl Acad Sci U S A, 2015. **112**(18): p. E2366-75.
109. Sun, L., et al., *microRNAs Involved in the Control of Innate Immunity in Candida Infected Caenorhabditis elegans.* Scientific Reports, 2016. **6**: p. 36036.
110. Zhi, L., et al., *Molecular Control of Innate Immune Response to Pseudomonas aeruginosa Infection by Intestinal let-7 in Caenorhabditis elegans.* PLoS pathogens, 2017. **13**(1): p. e1006152-e1006152.
111. Dai, L.-L., et al., *mir-233 Modulates the Unfolded Protein Response in C. elegans during Pseudomonas aeruginosa Infection.* PLOS Pathogens, 2015. **11**(1): p. e1004606.
112. Ma, Y.-C., et al., *mir-67 regulates P. aeruginosa avoidance behavior in C. elegans.* Biochemical and Biophysical Research Communications, 2017. **494**(1): p. 120-125.
113. Sharifnia, P. and Y. Jin, *Regulatory roles of RNA binding proteins in the nervous system of C. elegans.* Frontiers in Molecular Neuroscience, 2015. **7**: p. 100.
114. Davis, G.M., M.A. Haas, and R. Pocock, *MicroRNAs: Not "Fine-Tuners" but Key Regulators of Neuronal Development and Function.* Frontiers in neurology, 2015. **6**: p. 245-245.
115. Thompson-Peer, K.L., et al., *HBL-1 Patterns Synaptic Remodeling in C. elegans.* Neuron, 2012. **73**(3): p. 453-465.
116. Mello, C.C. and D. Conte, *Revealing the world of RNA interference.* Nature, 2004. **431**(7006): p. 338-342.
117. Fire, A., et al., *Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans.* Nature, 1998. **391**(6669): p. 806-11.
118. Mocellin, S. and M. Provenzano, *RNA interference: learning gene knock-down from cell physiology.* Journal of Translational Medicine, 2004. **2**(1): p. 39.
119. Ozata, D.M., et al., *PIWI-interacting RNAs: small RNAs with big functions.* Nature Reviews Genetics, 2019. **20**(2): p. 89-108.
120. Aravin, A.A., et al., *Double-stranded RNA-mediated silencing of genomic tandem repeats and transposable elements in the D. melanogaster germline.* Current Biology, 2001. **11**(13): p. 1017-1027.
121. Kim, D. and J. Rossi, *RNAi mechanisms and applications.* BioTechniques, 2008. **44**(5): p. 613-616.
122. Piatek, M.J. and A. Werner, *Endogenous siRNAs: regulators of internal affairs.* Biochemical Society transactions, 2014. **42**(4): p. 1174-1179.
123. Kim, V.N., J. Han, and M.C. Siomi, *Biogenesis of small RNAs in animals.* Nat Rev Mol Cell Biol, 2009. **10**(2): p. 126-39.

124. Sijen, T., et al., *On the role of RNA amplification in dsRNA-triggered gene silencing.* Cell, 2001. **107**(4): p. 465-76.
125. Sijen, T., et al., *Secondary siRNAs result from unprimed RNA synthesis and form a distinct class.* Science, 2007. **315**(5809): p. 244-7.
126. Holoch, D. and D. Moazed, *RNA-mediated epigenetic regulation of gene expression.* Nat Rev Genet, 2015. **16**(2): p. 71-84.
127. Updike, D. and S. Strome, *P granule assembly and function in Caenorhabditis elegans germ cells.* Journal of andrology, 2010. **31**(1): p. 53-60.
128. Aoki, K., et al., *In vitro analyses of the production and activity of secondary small interfering RNAs in C. elegans.* EMBO J, 2007. **26**(24): p. 5007-19.
129. Phillips, C.M., et al., *MUT-16 promotes formation of perinuclear mutator foci required for RNA silencing in the C. elegans germline.* Genes Dev, 2012. **26**(13): p. 1433-44.
130. Ashe, A., et al., *piRNAs can trigger a multigenerational epigenetic memory in the germline of C. elegans.* Cell, 2012. **150**(1): p. 88-99.
131. Bagijn, M.P., et al., *Function, targets, and evolution of Caenorhabditis elegans piRNAs.* Science, 2012. **337**(6094): p. 574-8.
132. Lee, H.C., et al., *C. elegans piRNAs mediate the genome-wide surveillance of germline transcripts.* Cell, 2012. **150**(1): p. 78-87.
133. Shirayama, M., et al., *piRNAs initiate an epigenetic memory of nonself RNA in the C. elegans germline.* Cell, 2012. **150**(1): p. 65-77.
134. Gu, W., et al., *Distinct argonaute-mediated 22G-RNA pathways direct genome surveillance in the C. elegans germline.* Mol Cell, 2009. **36**(2): p. 231-44.
135. Pak, J. and A. Fire, *Distinct populations of primary and secondary effectors during RNAi in C. elegans.* Science, 2007. **315**(5809): p. 241-4.
136. Zhang, C., et al., *mut-16 and other mutator class genes modulate 22G and 26G siRNA pathways in Caenorhabditis elegans.* Proc Natl Acad Sci U S A, 2011. **108**(4): p. 1201-8.
137. Phillips, C.M., et al., *MUT-14 and SMUT-1 DEAD box RNA helicases have overlapping roles in germline RNAi and endogenous siRNA formation.* Curr Biol, 2014. **24**(8): p. 839-44.
138. Buckley, B.A., et al., *A nuclear Argonaute promotes multigenerational epigenetic inheritance and germline immortality.* Nature, 2012. **489**(7416): p. 447-51.
139. Phillips, C.M., et al., *piRNAs and piRNA-Dependent siRNAs Protect Conserved and Essential C. elegans Genes from Misrouting into the RNAi Pathway.* Dev Cell, 2015. **34**(4): p. 457-65.
140. Vastenhouw, N.L., et al., *Gene expression: long-term gene silencing by RNAi.* Nature, 2006. **442**(7105): p. 882.
141. Henikoff, S. and J.M. Greally, *Epigenetics, cellular memory and gene regulation.* Curr Biol, 2016. **26**(14): p. R644-8.
142. Rechavi, O., G. Minevich, and O. Hobert, *Transgenerational inheritance of an acquired small RNA-based antiviral response in C. elegans.* Cell, 2011. **147**(6): p. 1248-56.
143. Luteijn, M.J., et al., *Extremely stable Piwi-induced gene silencing in Caenorhabditis elegans.* EMBO J, 2012. **31**(16): p. 3422-30.
144. Simon, M., et al., *Reduced insulin/IGF-1 signaling restores germ cell immortality to caenorhabditis elegans Piwi mutants.* Cell Rep, 2014. **7**(3): p. 762-73.

145. de Albuquerque, B.F., M. Placentino, and R.F. Ketting, *Maternal piRNAs Are Essential for Germline Development following De Novo Establishment of Endo-siRNAs in Caenorhabditis elegans.* Dev Cell, 2015. **34**(4): p. 448-56.
146. Raabe, C.A., et al., *Biases in small RNA deep sequencing data.* Nucleic acids research, 2014. **42**(3): p. 1414-1426.
147. Wright, C., et al., *Comprehensive assessment of multiple biases in small RNA sequencing reveals significant differences in the performance of widely used methods.* bioRxiv, 2019: p. 445437.
148. Fu, Y., et al., *Elimination of PCR duplicates in RNA-seq and small RNA-seq using unique molecular identifiers.* BMC Genomics, 2018. **19**(1): p. 531.
149. Grishok, A., *Biology and Mechanisms of Short RNAs in Caenorhabditis elegans.* Adv Genet, 2013. **83**: p. 1-69.
150. Uno, M. and E. Nishida, *Lifespan-regulating genes in C. elegans.* Npj Aging And Mechanisms Of Disease, 2016. **2**: p. 16010.
151. Danial, N.N. and S.J. Korsmeyer, *Cell Death: Critical Control Points.* Cell, 2004. **116**(2): p. 205-219.
152. Corsi, A.K., B. Wightman, and M. Chalfie, *A Transparent Window into Biology: A Primer on <em>Caenorhabditis elegans</em&gt.* Genetics, 2015. **200**(2): p. 387.
153. Greenstein, E.J.A.H.a.D., *Introduction to the germ line*, in *WormBook*, T.C.e.R. Community, Editor., WormBook.
154. Emmons, S.W., *Male development*, in *WormBook*, T.C.e.R. Community, Editor., WormBook.
155. Corsi, A.K., B. Wightman, and M. Chalfie, *A Transparent window into biology: A primer on Caenorhabditis elegans*, in *WormBook*, T.C.e.R. Community, Editor., WormBook.
156. Cutter, A.D., *Sperm-Limited Fecundity in Nematodes: How Many Sperm Are Enough?* Evolution, 2004. **58**(3): p. 651-655.
157. Derome, N., et al., *A brief history of bioinformatics.* 2018.
158. Lowe, R., et al., *Transcriptomics technologies.* PLoS computational biology, 2017. **13**(5): p. e1005457-e1005457.
159. Marx, V., *The big challenges of big data.* Nature, 2013. **498**: p. 255.
160. Nekrutenko, A. and J. Taylor, *Next-generation sequencing data interpretation: enhancing reproducibility and accessibility.* Nature Reviews Genetics, 2012. **13**: p. 667.
161. Garijo, D., et al., *Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome.* PLOS ONE, 2013. **8**(11): p. e80278.
162. Dumas, G., Y.-M. Kim, and J.-B. Poline, *Experimenting with reproducibility: a case study of robustness in bioinformatics.* GigaScience, 2018. **7**(7).
163. Hothorn, T. and F. Leisch, *Case studies in reproducibility.* Briefings in Bioinformatics, 2011. **12**(3): p. 288-300.
164. Budd, A., et al., *Bioinformatics training: a review of challenges, actions and support requirements.* Briefings in Bioinformatics, 2010. **11**(6): p. 544-551.
165. Attwood, T.K., et al., *A global perspective on evolving bioinformatics and data science training needs.* Briefings in Bioinformatics, 2017. **20**(2): p. 398-404.

166. Mulder, N., et al., *The development and application of bioinformatics core competencies to improve bioinformatics training and education.* PLOS Computational Biology, 2018. **14**(2): p. e1005772.
167. Yanai, I. and E. Chmielnicki, *Computational biologists: moving to the driver's seat.* Genome Biology, 2017. **18**(1): p. 223.
168. Lawlor, B. and P. Walsh, *Engineering bioinformatics: building reliability, performance and productivity into bioinformatics software.* Bioengineered, 2015. **6**(4): p. 193-203.
169. Verma, D., Gesell, Jon, Siy, Harvey, Zand, Mansour *Lack of Software Engineering Practices in the Development of Bioinformatics Software*, in *ICCGI 2013 : The Eighth International Multi-Conference on Computing in the Global Information Technology*. 2013. p. 58-62.
170. Stodden, V., J. Seiler, and Z. Ma, *An empirical analysis of journal policy effectiveness for computational reproducibility.* Proceedings of the National Academy of Sciences, 2018. **115**(11): p. 2584.
171. LeVeque, R.J., *Top Ten Reasons to Not Share Your Code (and why you should anyway)* in *SIAM News*. 2013.
172. Barnes, N., *Publish your computer code: it is good enough*, in *Nature News*. 2010, Nature.
173. Miller, G., *A Scientist's Nightmare: Software Problem Leads to Five Retractions.* Science, 2006. **314**(5807): p. 1856.
174. Wilson, G., et al., *Good enough practices in scientific computing.* PLoS computational biology, 2017. **13**(6): p. e1005510-e1005510.
175. Taschuk, M. and G. Wilson, *Ten simple rules for making research software more robust.* PLOS Computational Biology, 2017. **13**(4): p. e1005412.
176. Karimzadeh, M. and M.M. Hoffman, *Top considerations for creating bioinformatics software documentation.* Briefings in bioinformatics, 2017. **19**(4): p. 693-699.
177. Pavelin, K., et al., *Bioinformatics Meets User-Centred Design: A Perspective.* PLOS Computational Biology, 2012. **8**(7): p. e1002554.
178. Leprevost, F.d.V., et al., *On best practices in the development of bioinformatics software.* Frontiers in genetics, 2014. **5**: p. 199-199.
179. Russell, P.H., et al., *A large-scale analysis of bioinformatics code on GitHub.* PLOS ONE, 2018. **13**(10): p. e0205898.
180. Rother, K., et al., *A toolbox for developing bioinformatics software.* Briefings in bioinformatics, 2012. **13**(2): p. 244-257.
181. Pasquier, T., et al., *If these data could talk.* Scientific data, 2017. **4**: p. 170114-170114.
182. Kanwal, S., et al., *Investigating reproducibility and tracking provenance – A genomic workflow case study.* BMC Bioinformatics, 2017. **18**(1): p. 337.
183. Leipzig, J., *A review of bioinformatic pipeline frameworks.* Briefings in bioinformatics, 2017. **18**(3): p. 530-536.
184. Spjuth, O., et al., *Experiences with workflows for automating data-intensive bioinformatics.* Biology Direct, 2015. **10**(1): p. 43.
185. Kurtzer, G.M., V. Sochat, and M.W. Bauer, *Singularity: Scientific containers for mobility of compute.* PLOS ONE, 2017. **12**(5): p. e0177459.
186. Di Tommaso, P., et al., *The impact of Docker containers on the performance of genomic pipelines.* PeerJ, 2015. **3**: p. e1273-e1273.

187. Schulz, W.L., et al., *Use of application containers and workflows for genomic data analysis.* Journal of pathology informatics, 2016. **7**: p. 53-53.
188. Köster, J. and S. Rahmann, *Snakemake—a scalable bioinformatics workflow engine.* Bioinformatics, 2012. **28**(19): p. 2520-2522.
189. Di Tommaso, P., et al., *Nextflow enables reproducible computational workflows.* Nature Biotechnology, 2017. **35**: p. 316.
190. Voss K, G.J.a.V.d.A.G., *Full-stack genomics pipelining with GATK4 + WDL + Cromwell [version 1; not peer reviewed]*, in *18th Annual Bioinformatics Open Source Conference*. 2017, F1000Research.
191. Peter Amstutz, M.R.C., Nebojša Tijanić (editors), Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, Matt Scales, Stian Soiland-Reyes, Luka Stojanovic, *Common Workflow Language, v1.0. Specificatio*, C.W.L.w. group, Editor. 2016.

2. FUNCTIONAL SPECIALIZATION OF THE MIRNA-ASSOCIATED ARGONAUTES IN *C. ELEGANS*

The majority of the work in this chapter focuses on the characterization of a new branch of the miRNA pathway, ALG-5, but also contains a comprehensive analysis of the more well-known miRNA-associated Argonautes, ALG-1 and ALG-2[1,2]. We found that ALG-5 associates with a subset of miRNAs, many of which have unknown function. In contrast to ALG-1 and ALG-2, we also discovered that ALG-5 is a germline-specific Argonaute that localizes to p-granules. Much of the work and data provided here forms a solid foundation for future studies of the miRNA-associated Argonautes of *C. elegans.*

2.1 INTRODUCTION

MicroRNAs (miRNAs) interact with target mRNAs to control the levels and timing of gene expression in plants and animals [1]. miRNAs are processed from the stem regions of partially base-paired RNA hairpins into ~22-nucleotide (nt) duplexes with 2-nt 3' overhangs [2, 3]. miRNA duplexes form ribonucleoprotein complexes with effector proteins in the Argonaute/Piwi family, upon which, one of the two strands is ejected or degraded [4-6]. The miRNA strand retained in the complex acts as a sequence-specific guide to anchor the Argonaute to a target mRNA, which in animals typically occurs via base-pairing between the seed region of the miRNA (nucleotides 2-8) and the 3' UTR of the mRNA [7]. miRNAs affect gene expression through two distinct modes - inhibition of

[1] This chapter was published as written:
Brown, K.C., Svendsen, J.M., Tucci, R.M., Montgomery, B.E., Montgomery, T.A. (June 2017) ALG-5 is a miRNA-associated Argonaute required for proper developmental timing in the *Caenorhabditis elegans* germline. *Nucleic Acids Research*, doi: 10.1093/nar/gkx536

[2] Thanks to Josh Svendsen, Rachel Tucci, Brooke Montgomery, and Tai Montgomery for contributions

translation or recruitment of mRNA decay factors. The individual contributions of these two modes of silencing can vary depending in part on the cellular context [8].

Small interfering RNAs (siRNAs) and piwi-interacting RNAs (piRNAs) are distinct classes of small RNAs related to miRNAs by their length (~20-30-nt) and their association with Argonaute/Piwi proteins [9]. The Argonautes can be classified into three subfamilies by their phylogenetic relatedness, which is often indicative of which of the three classes of small RNAs they bind. The AGO subfamily is conserved across eukaryotes and contains both miRNA and siRNA associated Argonautes, whereas Argonautes in the PIWI subfamily bind their namesake piRNAs. The WAGO subfamily is unique to nematodes and has thus far only been implicated in siRNA pathways. The nematode *Caenorhabditis elegans* contains each of the three broad classes of small RNAs, as well as 25 Argonautes spanning each of the three subfamilies [10]. Each *C. elegans* Argonaute is specialized for a particular class or subclass of small RNAs, with the majority binding to the extensive repertoire of *C. elegans* siRNAs, which come in multiple varieties with distinct molecular features and functions [11].

The AGO subfamily of *C. elegans* Argonautes is comprised of five members, two of which, ALG-1 and ALG-2, interact with miRNAs, while two others, ALG-3 and ALG-4, function within the spermatogenesis branch of the 26-nt 5'G-containing siRNA (26G-RNA) pathway [11]. RDE-1, which primarily associates with siRNAs and does not clearly fall within any of the Argonaute subfamilies, also binds a subset of miRNAs [12].

HPO-24 (hereafter referred to as ALG-5 because of its relatedness to ALG-1-4), the fifth AGO subfamily Argonaute, has yet to be linked to a small RNA pathway.

We used protein-RNA co-immunoprecipitation combined with high-throughput sequencing to identify the small RNA interactors of ALG-5, as well as those of ALG-1 and ALG-2. We show that ALG-5 binds a subset of miRNAs that partially overlaps with those bound by ALG-1 and ALG-2. *alg-5* is expressed in the germline and ALG-5 protein localizes, in part, to P granules. Loss of *alg-5* activity results in a modest reduction in fertility and an accelerated transition from spermatogenesis to oogenesis in hermaphroditic animals. Using RNA-seq of *alg-5*, *alg-1*, and *alg-2* mutants, we identified hundreds of mRNAs misregulated in the absence of each branch of the miRNA pathway. Of the mRNAs misregulated in *alg-5* mutants, genes involved in defense were most significantly enriched. The results implicate ALG-5 as a distinct germline-specific branch of the miRNA pathway and pave the way for functional analysis of the role of ALG-5 in immunity and development.

2.2 MATERIALS AND METHODS

Strains

N2 [wild type], VC446 [*alg-1(gk214)* X], WM53 [*alg-2(ok304)* II], WM159 [*alg-5(tm1163)* I], MT14119 [mir-35-41(nDf50)], and SS104 [*glp-4(bn2)* I] were obtained from the CGC. *RFP::pgl-1* was described in [13]. The *alg-5(tm1163)* allele was backcrossed to wild type an additional two times. New strains generated for this study are listed in

Supplementary Table S2.1. *alg-1::HA::alg-1, alg-2::HA::alg-2, alg-1::HA::alg-2, alg-5::HA::alg-5, alg-1::HA::alg-5* transgenes were generated using Life Technologies Multisite Gateway Technology. Individual promoter (~2,400-3,700 nt upstream of start codons), CDS (start to stop codons), and 3' UTR (~400-1,500 nt downstream of stop codons) sequences were PCR amplified from genomic DNA using Phusion polymerase (New England Biolabs). PCR products were cloned into entry vectors using Gateway BP recombination (Life Technologies). The HA epitope tag was PCR amplified from pENTR 3XHA-AGO1 [14] with primers that added a TEV tag and SpeI and NdeI restriction sites (Supplementary Table S2.2) and introduced into pENTR (Life Technologies). The 3XHA-TEV cassette was restriction digested from the pENTR plasmid using SpeI and NdeI and ligated into the *alg-1*, *alg-2*, and *alg-5* CDS entry clones. Individual fragments were recombined into destination vectors modified for Life Technologies Multisite Gateway Technology (pCFJ151, *alg-5::HA::alg-5*, and pCFJ178, *alg-1::HA::alg-1, alg-2::HA::alg-2, alg-1::HA::alg-2*) [15]. *alg-1::HA::alg-2* and *alg-1::HA::alg-5* were generated by recombining the *alg-1* promoter and 3'UTR sequences with the *alg-2* or *alg-5* CDS sequence, respectively. Constructs were sequence-verified and introduced into EG6699 [*ttTi5605* II; *unc-119(ed3)* III; oxEx1578] for integration on chromosome II (*alg-5::HA::alg-5, alg-1::HA::alg-5*) or EG5003 [*unc-119(ed3)* III; *cxTi10882* IV] for integration on chromosome IV (*alg-1::HA::alg-1, alg-2::HA::alg-2, alg-1::HA::alg-2*), using MosSCI [16]. *alg-5(ram1[GFP::3xFLAG::alg-5 + loxP]), alg-5(ram2[GFP::3xFLAG + loxp]),* and *alg-5(ram9[GFP::3xFLAG::alg-5^{tm1163} + loxP]),* were generated using CRISPR/Cas9 as described in [17, 18] using plasmids pDD162 and

pDD282 (AddGene). Guide RNAs were designed using http://crispr.mit.edu/. Primer sequences are in Supplementary Table S2.2. Unless noted otherwise, strains were grown under standard conditions at 20˚C [19].

Phylogenetics

AGO clade Argonaute protein sequences [10] were aligned using ClustalW2 2.1 with the Dayhoff-PAM weight matrix [20]. Protein maximum likelihood distances were calculated and the phylogenetic tree was drawn in Phylip 3.69 [21].

Co-immunoprecipitation

Animals were grown at 20˚C for 49 (GFP::ALG-5) or 68 (all HA::ALG strains) hrs following L1 synchronization. Animals were flash frozen in liquid nitrogen and lysed in 50 mM Tris-Cl, pH 7.4, 100 mM KCl, 2.5 mM MgCl2, 0.1% Igepal CA-630, 0.5 mM PMSF, and 1X Proteinase Inhibitor (Life Technologies, 88266). Cell debris was removed by centrifugation and cell lysates were incubated with anti-HA affinity matrix (Roche, 11815016001) or anti-GFP mAb-agarose (MBL, D153-8) for 1 hr. Following co-immunoprecipitation (co-IP), beads were washed 4 times in lysis buffer and split into RNA and protein fractions.

Protein isolation

Proteins were extracted from co-IPs or whole animals using Laemmli buffer. Embryos were extracted from gravid adults by hypochlorite treatment and incubated for ~1 hr in

M9. L1 animals were collected after hatching and incubation for ~24 hrs in M9. L2 animals were collected ~20 hrs after L1 synchronization, L3 animals were collected ~27 hrs after L1 synchronization. L4 animals were collected ~48 hrs after L1 synchronization. Gravid adults were collected ~68 hrs after L1 synchronization. For comparison of HA::ALG-5 levels in males and hermaphrodites, two replicates of 400 L4 stage animals of each sex were collected by hand picking animals ~48 hrs after L1 synchronization of F1 animals from a self-cross between *alg-5::HA::alg-5* transgenic animals to enrich for males. For comparison of HA::ALG-5 and HA::ALG-1 levels in animals wild type for or deficient in germline proliferation, animals were treated with control (L4440) or *glp-4* dsRNA [22] and collected ~68 hrs after L1 synchronization.

Western blots

Proteins were resolved on 4-12% Bis-Tris SDS polyacrylamide gels and transferred to nitrocellulose membranes (Life Technologies). Blots were blocked in PBST containing 5% milk and probed with anti-HA (Roche, 12013819001), anti-actin (Abcam, ab3280), or anti-GFP antibodies (Invitrogen, MA5-15256-HRP). SuperSignal West Femto Maximum Sensitivity Chemiluminescent Substrate (Life Technologies, 34096) was used for signal detection. Where applicable, signal intensity was quantified on a Bio-Rad ChemiDoc and HA-fusion protein levels were normalized to actin levels.

RNA isolation

RNA was isolated from whole animals after flash freezing in liquid nitrogen or from input and co-IP fractions using Trizol (Life Technologies, 15596018) followed by two chloroform extractions and isopropanol precipitation. RNA was diluted to 1.0 +/- 0.05 ug/ul prior to library preparation and qRT-PCR. For comparison of *alg-5*, *alg-1*, and *alg-2* mRNA levels in wild type and *glp-4(bn2)* mutants, three biological replicate pools (n=~18,000 each) were collected as stage-matched young adults prior to the appearance of embryos in the uterus. For comparison of *alg-5* mRNA levels in males and hermaphrodites, three replicates of 50 L4 stage animals of each sex were collected by hand picking animals ~48 hrs after L1 synchronization of F1 animals from a genetic cross between wild type animals to enrich for males.

Small RNA sequencing

Small RNAs in the 18-28-nt range were purified from total RNA by size selection using electrophoretic transfer from 17% polyacrylamide gels. Purified small RNAs were treated with RNA polyphosphatase (Illumina, RP8092H) or Tobacco Alkaline Phosphatase (Epicentre Biotechnologies, T81050) to reduce di and triphosphates to monophosphates to facilitate capture of 22G-RNAs by 5' adapter ligation. Phosphatase was deactivated and removed after 30 min by phenol:chloroform extraction. Preadenylated 3' adapter was ligated to small RNAs using T4 RNA Ligase 2 Truncated KQ (NEB, M0373S). 5' adapter was ligated using T4 RNA Ligase (Life Technologies, AM2140). Ligation reactions were done at 16˚C for 16-18 h. Adapter-ligated small RNAs

were size selected at each ligation step using electrophoretic transfer from 12 or 15% polyacrylamide gels. Adapter-bound small RNAs were reverse transcribed using SuperScript III (Life Technologies, 18080-044) and the Illumina TruSeq RT Primer. RT products were amplified using NEBNext 2X PCR Master Mix (NEB, M0541S) and the TruSeq forward primer and reverse primers containing index sequences. PCR products corresponding to 18-28-nt small RNAs (~136-146 bp) were size selected using electrophoretic transfer from 10% polyacrylamide gels. Samples were sequenced on an Illumina HiSeq 2000, HiSeq 2500, or NextSeq 500. For each Argonaute analyzed, small RNA sequencing from co-IPs was done at least twice, and although results were consistent across experiments, for simplicity only one dataset is described. Primer and adapter sequences are in Supplementary Table S2.2.

Small RNA sequencing data analysis

Small RNA sequences were parsed from adapters, filtered for quality, and aligned to the *C. elegans* genome (WS230) using CASHX 2.3 [23]. The numbers of reads sequenced, parsed, and mapped are described in Supplementary Table S2.3. Data analysis was done using R and custom Perl and Python scripts. miRNA annotation was based on miRBase release 20. Mutator class siRNA annotation was based on Phillips et al. [15]. CSR-1 class siRNA annotation was based on Claycomb et al. [24]. piRNA annotation was based on WormBase release WS230. New miRNAs were identified using miRDeep2 [25]. To identify GFP::ALG-5, HA::ALG-5, HA::ALG-1, and HA::ALG-2 interactors, we calculated the normalized reads (reads per million total genome-

matching reads in each library) in the small RNA libraries derived from the co-IP fractions relative to the cell lysate (input, in) fractions. HA::ALG-5 interactors were defined as miRNAs that were enriched in the co-IP fraction by ≥2 fold to account for presumed non-specific carryover from the cell lysates. Unless noted otherwise, a 1-fold cutoff was applied to HA::ALG-1, HA::ALG-2, and GFP::ALG-5 because these co-IPs had very little non-specific carryover from the cell lysates.

mRNA sequencing

Methodology for mRNA library preparation was adapted from the NEBNext Ultra Directional RNA Library Prep Kit and Zhang et al. [26]. RNA isolated from ~5,000 wild type, *alg-5(ram2), alg-1(gk214), and alg-2(ok304)* mutant L4 stage animals per replicate (3 replicates per strain) was depleted of rRNA using the Ribo-Zero Magnetic Kit (Illumina, MRZH116). rRNA-depleted RNA was enriched for RNA >200 nucleotides using the RNA Clean & Concentrator-5 Kit (Zymo Research, R1015) and fragmented to 200-350 bp by incubating in SuperScript III 5X first strand buffer (Life Technologies) for 2 min at 94˚C. First strand cDNA was synthesized from fragmented RNA using Superscript III RT and random hexamers (Life Technologies, 18080-093). Second strand cDNA was synthesized using the NEBNext Second Strand Synthesis Module (NEB, E7550S), which uses dUTP instead of dTTP to preserve strand information. Double-stranded cDNA was end repaired using NEBNext Ultra End Repair/dA-Tailing Module (NEB, E7442S). 200-350 bp double-stranded cDNA was size selected using AMPure XP Beads (Beckman Coulter, A63881). Adapters were ligated using T4 DNA

Ligase (NEB, M0202S). Uracils were excised from cDNA using USER enzyme (NEB, M5505S) and cDNA strands that had contained uracil were degraded to prevent capture of the antisense strands. cDNA libraries were amplified by PCR. cDNA and PCR products were purified using AMPure XP Beads. Samples were sequenced on an Illumina HiSeq 2500. Primer and adapter sequences are in Supplementary Table S2.2.

mRNA sequencing data analysis

Adapter sequences and low quality bases were trimmed from mRNA sequences using Trimmomatic 0.35 [27]. Trimmed sequences were aligned to the *C. elegans* WS230 genome using TopHat2 [28]. The numbers of reads sequenced, parsed, and aligned are described in Supplementary Table S2.3. Data processing and quality assessment were done using custom scripts in Python and R. Differentially regulated protein-coding genes were identified using Cuffdiff2 [29] and HTSeq-count followed by DESeq2 [30, 31]. rRNA, tRNA, and mtRNA were masked from the analysis. A 1.5 fold-change cutoff was applied when filtering significantly affected genes. DAVID 6.8 was used to identify significantly overrepresented functional annotations using a Benjamini- Hochberg adjusted p-value cutoff of 0.05 [32, 33]. Categories were collapsed and colored the same in plots if there was greater than 50% overlap of genes within the category containing fewer genes. Venn diagrams were generated using BioVenn [34]. Reads were plotted in IGV 2.3.67 [35, 36]. Volcano plots were drawn with CummeRbund [29]. miRNA target site abundance in differentially regulated genes was assessed using Targetscan Release 6.2 and custom scripts in Python and R [37, 38].

Quantitative RT-PCR

For qRT-PCR, Turbo DNase-treated total RNA (Life Technologies, AM1907) was subjected to reverse transcription with SuperScript III (Life Technologies, 18080-044) using an oligo(dT) primer to enrich for mRNA. qRT-PCR was done using iTaq Universal SYBR Green Supermix (Bio-Rad, 172-5122) and the primer sequences in Supplementary Table S2.2. Reverse transcription and qPCR were done according to manufacturers' specifications. qRT-PCR was done using a CFX96 Touch Real-Time PCR Detection System (Bio-Rad). Means and standard deviations were calculated for three biological replicates in each experiment. The 2^{-ddCT} method was used to quantify fold change differences between samples. *rpl-32* was used for normalization. P-values were calculated using ANOVA followed by either two-sample t-tests when making one comparison or Tukey HSD tests when making multiple comparisons.

RNAi assays

Synchronized L1 animals were fed *E. coli* HT115 expressing either an empty vector control (L4440), or *alg-1*, *alg-2*, or *glp-4* dsRNA [22].

Phenotype assays

Animals were grown at 20˚C on NGM plates containing live *E. coli* (OP50) unless noted otherwise. Brood size assays were done on individual animals over their entire lifetimes at 20˚C or 25˚C. Live progeny of each animal were counted and removed from the plates each day such that all hatched live animals were included in our counts. Dead

embryos were not included. P-values for brood size assays were calculated using the Wilcoxon Rank Sum test. The numbers of animals that burst or had protruding vulvas were counted at 96-120 hrs. The timing of oogenesis was assayed in three independent experiments. Animals were scored as oogenic if the germline clearly contained at least one oocyte as evidenced by appearing as a larger, single-row, and often rectangular germ cell next to the spermatheca. Otherwise, if the gonad was clearly visible and did not appear to contain oocytes, the animal was scored as non-oogenic.

Imaging

Imaging of live animals was done on a Zeiss Axio Imager Z2 upright microscope. Animals were immobilized in a 25 uM sodium azide solution on 1.5-2% Agarose pads. For assessing the presence or absence of oocytes, animals were imaged 56-61 hrs after L1 synchronization. For imaging GFP::ALG-5 and free GFP from *alg-5(ram2)*, the developmental stage was determined by the number of germ cells and the germline or whole animal morphology.

2.3 RESULTS

2.3.1 ALG-5 is Required for the Proper Developmental Timing in the Germline

ALG-5 is an AGO subfamily Argonaute most closely related in *C. elegans* to the miRNA-associated Argonautes ALG-1 and ALG-2 (~36% amino acid identity) (Figure 2.1A) [39]. Whereas the role of ALG-5 is unknown, ALG-1 and ALG-2 have roles throughout development. *alg-1(gk214)* mutants display a strong reduction in the number of viable

progeny they produce relative to wild type animals and *alg-2(ok304)* mutants display a more modest reduction in viable progeny (Figure 2.1B) [39, 40]. The publically available partial deletion allele, *alg-5(tm1163)*, results in the loss of 145 amino acids in ALG-5, however, the mRNA is produced at near wild type levels outside of the deleted region (Supplementary Figure S2.1A). The protein produced by the *alg-5(tm1163)* allele is predicted to have a truncated PAZ domain, which engages the 3' end of the small RNA, and to lack the linker 2 domain, which links the PAZ and PIWI lobes (Supplementary Figure S2.1B) [41]. It is unclear, however, whether the mutation would result in complete loss of function phenotype. Thus, using CRISPR-Cas9 we developed an open reading frame deletion of *alg-5*, *alg-5(ram2)* in which the coding region was replaced with GFP sequence (Supplementary Figure S2.1B). Similar to what we observed in *alg-2(ok304)* mutants, *alg-5(tm1163)* mutants produced a median ~24% fewer viable progeny than wild type animals ($p = 0.0015$, Supplementary Figure S2.1C) and *alg-5(ram2)* mutants produced ~15% fewer progeny ($p=0.025$, Figure 2.1C).

Aside from the modest reduction in brood size, neither *alg-5(tm1163)* nor *alg-5(ram2)* mutants displayed obvious developmental defects and in general appeared healthy, suggesting a specific requirement for ALG-5 in germline development or embryogenesis. Given the well-described heterochronic roles for small RNAs in *C. elegans*, we examined the timing of germ cell progression between spermatogenesis and oogenesis in wild type and *alg-5(ram2)* mutants. In each of three independent

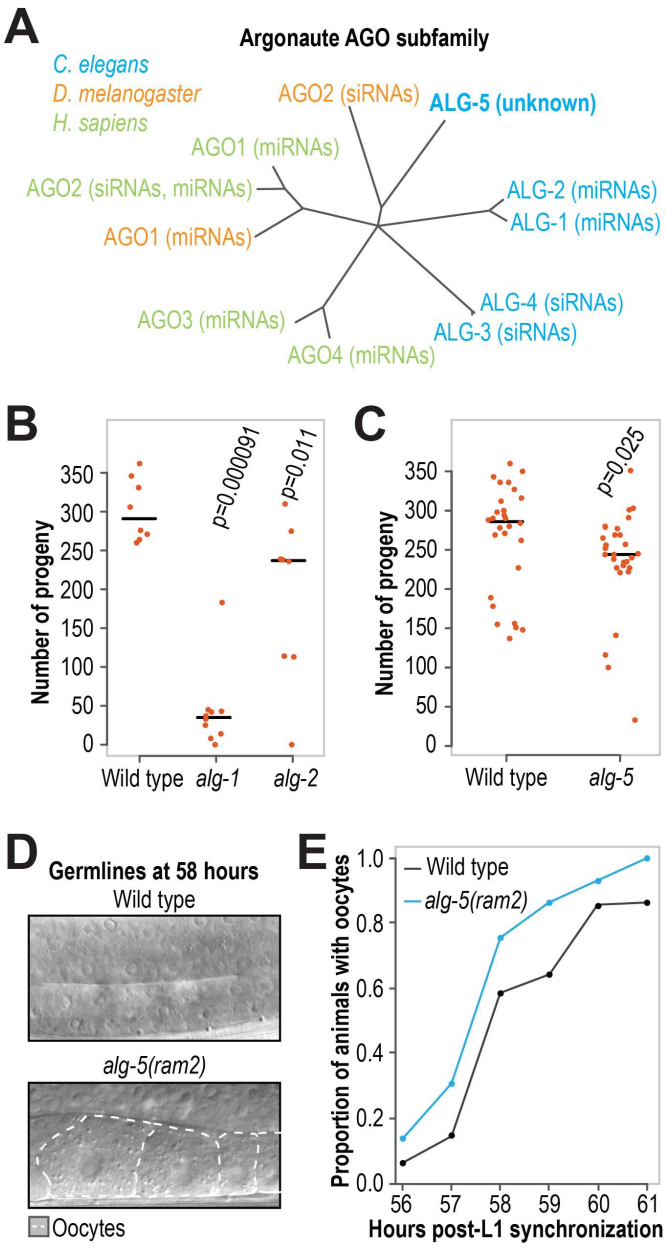


Figure 2.1. ALG-5 is required for optimal fertility and proper timing of oogenesis. (A) Phylogenetic tree of the AGO subfamily in worms, flies, and humans. (B) Numbers of viable progeny produced by wild type (n=8), *alg-1(gk214)* (n=10), and *alg-2(ok304)* (n=8) at 20°C. (C) Number of viable progeny produced by wild type (n= 28) and *alg-5(ram2)* (n=29) grown at 20°C. (D) Representative images of wild type and *alg-5(ram2)* mutant germlines at 58 hours post-L1 synchronization. The regions where oocytes form is shown. (E) Proportions of wild type and *alg-5(ram2)* mutant animals with oocytes formed at 56-61 hours post-L1 synchronization (n=~25-50). One of three independent experiments is shown (the other two experiments are shown in Supplementary Figure S2.1D). At 58 hours, the proportion of *alg-5(ram2)* mutant animals with oocytes is 17-35% higher than in wild type across the three experiments. See also Supplementary Figure S2.1.

experiments, *alg-5(ram2)* mutants displayed precocious development of oocytes, pointing to an accelerated transition from spermatogenesis to oogenesis (Figures 2.1D and E and Supplementary Figure S2.1D). Our results therefore suggest that ALG-5 is required for the proper timing of oogenesis. The number of sperm produced in *C. elegans* hermaphrodites prior to oogenesis limits overall fecundity [42, 43]. Thus, the premature switch to oogenesis in *alg-5* mutants presumably reduces the number of sperm available for fertilization, likely resulting in the observed reduction in progeny.

2.3.2 ALG-5 is Primarily Expressed in the Germline

To determine when ALG-5 is expressed during development, we made an *HA::alg-5* epitope fusion transgene containing the endogenous *alg-5* 5' and 3' regulatory sequences and introduced it into *C. elegans* using Mos1-mediated single copy integration [16]. We then crossed the transgene into the *alg-5(tm1163)* mutant strain and examined by Western blot analysis HA::ALG-5 levels at each of the major developmental stages. Because of their relatedness to ALG-5, we also examined HA::ALG-1 and HA::ALG-2 levels across developmental stages using single-copy transgene strains developed for this study (see Materials and Methods). HA::ALG-5 expression was highest during late stages of larval development and into adulthood (Figures 2.2A and B). HA::ALG-1 was abundant throughout development, consistent with a central role for ALG-1 in the miRNA pathway (Figure 2.2C) [39, 40, 44-48]. In contrast, HA::ALG-2 was predominantly expressed in embryos (Figure 2.2D).

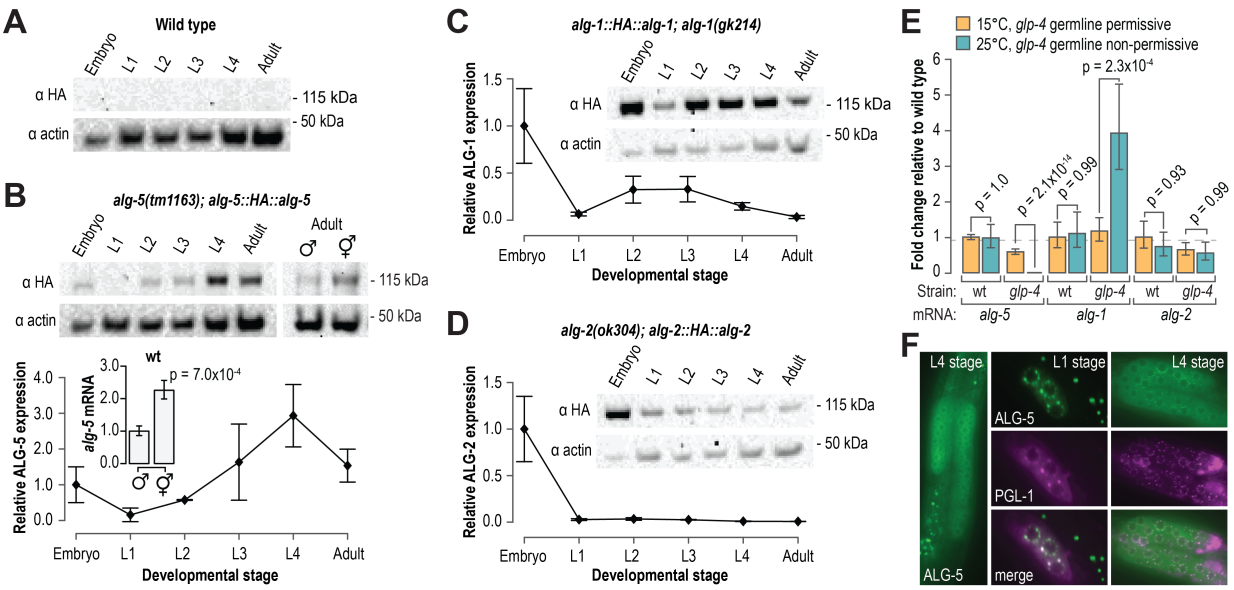


Figure 2.2. *alg-5* is specifically expressed in the germline. (A) Western blot assay of HA and actin in wild type animals across developmental stages. Non-transgenic wild type animals do not express the HA epitope and are included as a negative control. (B) Western blot assay and quantification of HA::ALG-5. A blot image of 1 of 2 biological replicates is shown. Points within the plot represent average signal intensity of HA normalized to actin (embryo sample arbitrarily set to 1.0). Error bars represent standard deviations from the mean. A western blot assay of HA::ALG-5 in hermaphrodites and males is also shown. The bar plot displays relative levels of endogenous *alg-5* mRNA in wild type animals, as determined by quantitative RT-PCR, in hermaphrodites and males. (C-D) Western blot assay and quantification of HA::ALG-1 (C) and HA::ALG-2 (D). Points within the plots represent average signal intensity of HA normalized to actin (embryo sample arbitrarily set to 1.0). Error bars represent standard deviations from the mean. (E) Average fold change in *alg-5, alg-1,* and *alg-2* transcript levels in wild type and *glp-4(bn2)* at 15°C (orange) and 25°C (teal), as determined by quantitative RT-PCR. Error bars represent standard deviations from the means for three biological replicates. (F) Representative images of GFP::ALG-5 and RFP::PGL-1. Images are of GFP or RFP fluorescence in the germline regions of living animals. See also Supplementary Figure S2.2.

The expression of HA::ALG-5 in late larval stages and adults (Figure 2.2B), the stages of development in which the *C. elegans* germline proliferates and matures, and the requirement of ALG-5 for the proper timing of oogenesis both point to a role for ALG-5 in germ cells. To determine if *alg-5* expression is elevated in germ cells relative to somatic cells, we measured by qRT-PCR *alg-5* mRNA levels in wild type and *glp-4(bn2)* mutant animals. When grown at the permissive temperature of 15˚C, the germlines of *glp-4* mutants develop normally, but when grown at the non-permissive temperature of 25˚C, the germlines fail to proliferate. Thus, a gene that is enriched in germ cells will be depleted in *glp-4* mutant animals grown at 25˚C relative to animals grown at 15˚C. *alg-5* levels were depleted ~400 fold in *glp-4* mutants grown at 25˚C relative to those grown at 15˚C ($p = 2.1x10^{-14}$) (Figure 2.2E). In contrast, *alg-1* mRNA levels were elevated >3 fold in *glp-4* mutants grown at 25˚C relative to those grown at 15˚C, indicating that *alg-1* is depleted in germ cells ($p =0.00023$) (Figure 2.2E). *alg-2* mRNA levels were not significantly different between *glp-4* mutants grown at 15˚C or 25˚C ($p =1.0$), suggesting that it is expressed in both somatic and germ cells (Figure 2.2E). Consistent with germline-specific expression, *alg-5* mRNA and protein levels were ~2 fold higher in hermaphrodites, which contain two gonad arms, than in males, which contain a single gonad arm (Figure 2.2B).

To examine the tissue and cellular localization of ALG-5, we used CRISPR-Cas9 to introduce GFP sequence at the 5' end of the coding sequence of the endogenous *alg-*5 locus in wild type animals (Supplementary Figure S2.1B) [17, 18]. Our *alg-5* deletion

allele*, alg-5(ram2)*, described above also provides a transcriptional readout for *alg-5* expression, as it contains the *alg-5* 5' and 3' regulatory sequences flanking GFP coding sequence (Supplementary Figure S2.1B). Free GFP expressed from the *alg-5(ram2)* allele was present throughout development but was restricted to germ cells (Supplementary Figure S2.2A). Similarly, the GFP::ALG-5 fusion protein was detectable throughout development but only detectable above background in germ cells (Figure 2.2F). GFP::ALG-5 appeared cytoplasmically diffuse but also formed distinct puncta at the nuclear periphery reminiscent of P granules, a germ cell-specific class of RNA granules that function in mRNA surveillance. P granules contain the piRNA-associated Piwi protein, PRG-1, and much of the siRNA pathway machinery [49]. GFP::ALG-5 foci overlapped with the P granule marker RFP::PGL-1 foci, indicating that, similar to many known piRNA and siRNA components, ALG-5 localizes to P granules (Figure 2.2F).

We also introduced GFP at the 5' end of *alg-5* coding sequence in *alg-5(tm1163)* to determine if the mutant allele produces a stable protein [17, 18]. Indeed, GFP::ALG-5^{tm1163} was expressed at similar levels to non-mutant GFP::ALG-5 protein and formed foci at the nuclear periphery (Supplementary Figure S2.2B). Because mutant ALG-5 produced from the *alg-5(tm1163)* allele could conceivably compete with other Argonautes for shared components of a small RNA pathway, it is important to interpret results obtained from the *alg-5(tm1163)* allele with caution.

2.3.3 ALG-5 Functions in the miRNA Pathway

To identify the small RNAs bound by ALG-5 and thus place ALG-5 within our current understanding of small RNA pathways, we co-immunoprecipitated GFP::ALG-5 and HA::ALG-5 protein complexes and subjected the associated small RNAs to high-throughput sequencing (Figure 2.3A and Supplementary Figure S2.3A). piRNAs, 22-nt 5'G-containing siRNAs (22G-RNAs), and 26G-RNAs were all depleted in both the GFP::ALG-5 and HA::ALG-5 co-immunoprecipitates (co-IPs), whereas miRNAs were enriched ~2-3 fold (Figures 2.3B and C; Supplementary Figures S2.3A and B). Although as a class miRNAs were enriched, the majority of individual miRNAs were depleted in the ALG-5 co-IPs, as were individual piRNAs and 22G-RNA clusters (Figure 2.3D and Supplementary Table S2.4). Of the 368 annotated miRNAs in *C. elegans*, including both strands of each miRNA duplex, only 24 yielded >10 normalized reads (reads per million total mapped reads) and were enriched >1 fold in the GFP::ALG-5 co-IP. Of these, 10 were enriched >25 fold, indicating that ALG-5 binds with high affinity a very small number of miRNAs (Figure 2.3E). Although GFP::ALG-5 was co-immunoprecipitated from L4 stage animals and HA::ALG-5 from adult animals, there was nonetheless a majority overlap in the associated miRNAs (Figure 2.3E).

We next used small RNA high-throughput sequencing to assess miRNA accumulation defects in the two *alg-5* mutant strains, *alg-5(tm1163)* and *alg-5(ram2)*. Only one miRNA, miR-250-3p, was depleted >3 fold in the *alg-5(tm1163)* mutant (Supplementary Table S2.5). miR-250-3p was the fourth most highly enriched miRNA in the HA::ALG-5

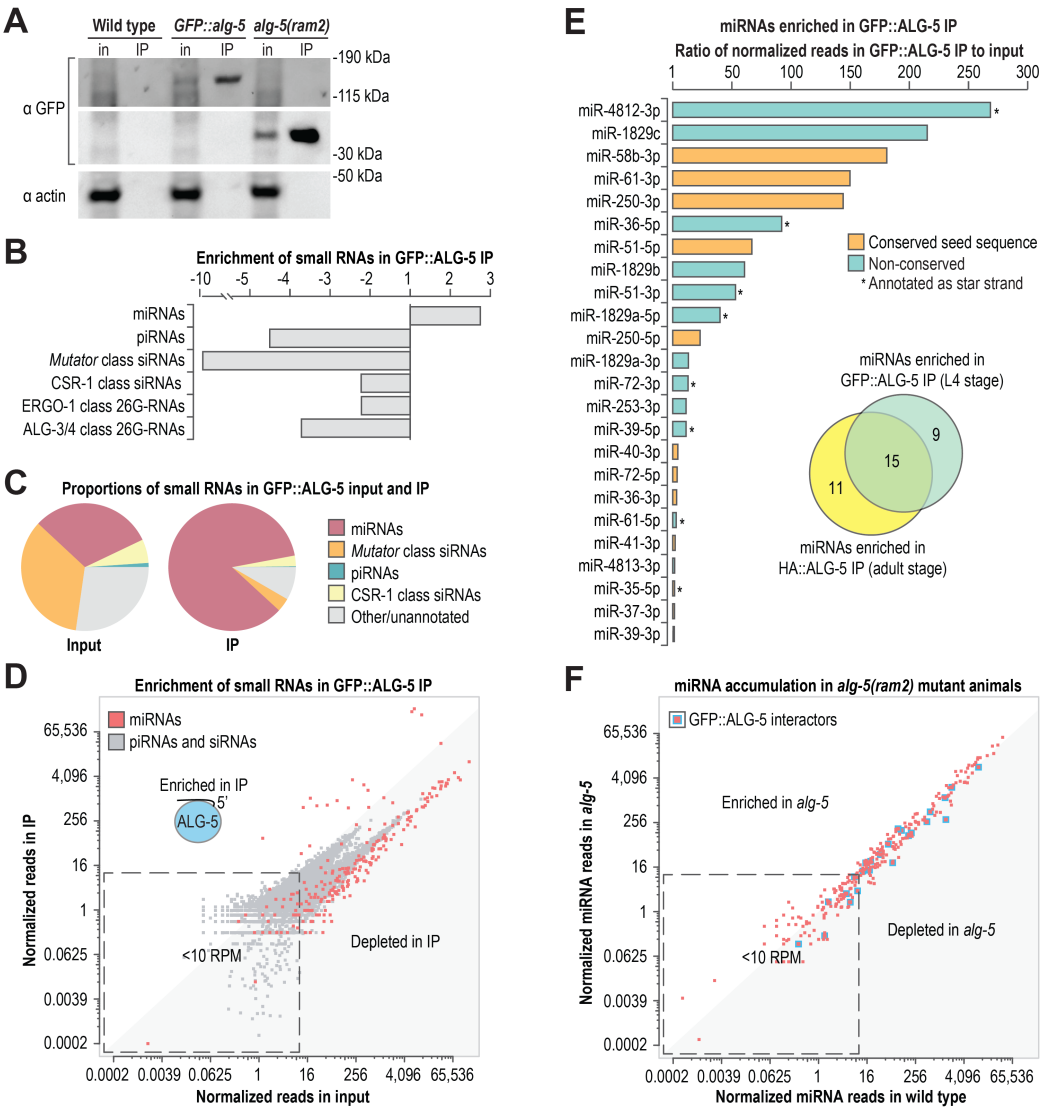


Figure 2.3. ALG-5 binds a subset of miRNAs. (A) Western blot assay of GFP::ALG-5 from cell lysates (input, in) and co-IPs (IP) used for small RNA isolation and sequencing. ~0.2% starting material equivalents for the input fractions and ~5% starting material equivalents for the co-IP fractions were run on the gels for Western blots. (B) Enrichment of miRNAs, piRNAs, and siRNAs in GFP::ALG-5 co-IP relative to input as determined by high-throughput sequencing. (C) The relative proportions of each class of small RNAs in input and co-IP fractions. (D) Normalized reads (reads per million total mapped reads) for each miRNA in GFP::ALG-5 co-IP versus input are shown in red. Normalized reads for other classes of small RNAs (piRNAs and siRNA loci) are shown in grey. (E) miRNAs enriched >1 fold in the GFP::ALG-5 co-IP relative to input. Colors indicate if the seed sequence (positions 2-8) is conserved in *Drosophila melanogaster* and/or *Homo sapiens*. Asterisks indicate if the sequence is annotated as a star strand in miRBase v. 20. The inset Venn diagram displays the overlap in miRNAs enriched in the GFP::ALG-5 (L4 stage animals) and HA::ALG-5 (adult animals) co-IPs. (F) Normalized reads for each miRNA in *alg-5(ram2)* versus wild type. See also Supplementary Figure S2.3 and Supplementary Tables S2.3-S2.5.

co-IP (~100 fold) and the fifth most highly enriched in the GFP::ALG-5 co-IP (~140 fold), and its levels were partially rescued in *alg-5(tm1163)* by the *HA::alg-5* transgene (Supplementary Figure S2.3C and Supplementary Table S2.4). Five miRNAs, including miR-250-3p, yielded >10 normalized reads and were depleted >3 fold in *alg-5(ram2)* mutants, only two of which were enriched in the GFP::ALG-5 co-IP (Figure 2.3F and Supplementary Table S2.5). Thus, the majority of miRNAs bound by ALG-5 are not dependent on ALG-5 for their overall stability, possibly because of association with other Argonautes, although they may be impacted specifically in the germline which might be missed in our whole animal-based approach.

2.3.4 ALG-5, ALG-1, and ALG-2 Interact with Distinct Subsets of miRNAs

To help determine the relatedness of ALG-5 to ALG-1 and ALG-2 within the miRNA pathway, we isolated small RNAs bound to HA-epitope fusions of ALG-1 and ALG-2 from adult animals and subjected them to high throughput sequencing (Supplementary Figure S2.4A). The majority of miRNAs were enriched in the HA::ALG-1 co-IP relative to the cell lysate (Figure 2.4A; Supplementary Figures S2.4B and C; Supplementary Table S2.4). Most miRNAs were also depleted in *alg-1(gk214)* mutants, although this may be due in part to developmental defects in *alg-1* mutants (Figure 2.4B and Supplementary Table S2.5) [44, 47]. Total miRNA levels were depleted by ~40% in *alg-1* mutants and were partially rescued by the *HA::alg-1* transgene (Supplementary Figure S2.4D). Similar to HA::ALG-1, HA::ALG-2 interacted with the majority of miRNAs (Supplementary Figures S2.4B and C; Supplementary Table S2.4). However, unlike

HA::ALG-1, which showed little bias for specific miRNAs, the HA::ALG-2 co-IP was strongly enriched for miR-35-42 family miRNAs, miR-51, and miR-1829a (~10-24 fold) (Figure 2.4C and Supplementary Table S2.4). *alg-2(ok304)* mutants displayed only modest enrichment or depletion in the levels of most miRNAs, although members of the miR-35-42 family were depleted by ~8-12 fold, except for miR-42 which was depleted by only ~2 fold (Figure 2.4D; Supplementary Figure S2.4D; Supplementary Table S2.5). miR-35-42 levels in *alg-2(ok304)* mutants were partially restored by the *HA::alg-2* transgene (Supplementary Figure S2.4D). The miR-35 and miR-51 families are required for embryogenesis [50]. Thus, the strong enrichment we observed for miR-35 family miRNAs and miR-51 in the HA::ALG-2 co-IP and the relatively strong expression of HA::ALG-2 in embryos points to a prominent role for ALG-2 in conferring robustness to the miRNA pathway during embryogenesis (Figures 2.2D and 2.4C). In support of this model, we were unable to isolate animals homozygous mutant for both *alg-2* and *mir-35-41* (the *mir-35-41* deletion mutant has only a partially penetrant embryonic lethality phenotype because it contains wild type mir-42), suggesting that *alg-2* enhances the *mir-35-41* mutant phenotype (Supplementary Figure S2.4E) [50].

Of the 159 miRNAs that yielded >10 normalized reads (reads per million total mapped reads) and were enriched >1 fold in the HA::ALG-1 or HA::ALG-2 co-IPs, 14 were uniquely bound by HA::ALG-2 and 40 were uniquely bound by HA::ALG-1, based on this enrichment criterion (Figure 2.4E). Of the 26 miRNAs enriched in the HA::ALG-5 co-IP >2 fold (a 2-fold cutoff was used because of relatively high carryover from the cell

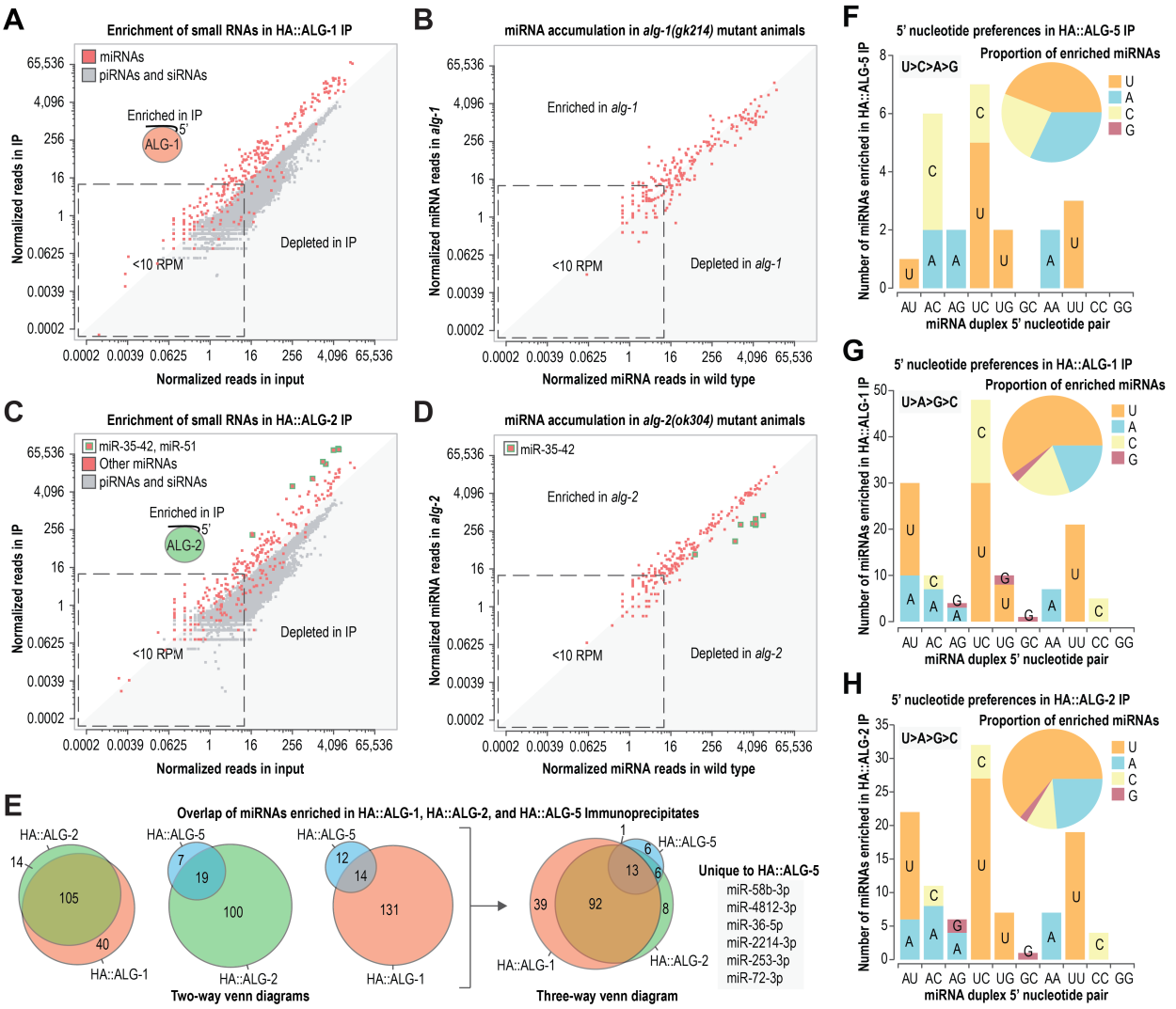


Figure 2.4. Overlap between miRNAs associated with ALG-5, ALG-1, and ALG-2. (A) Normalized reads for each miRNA in HA::ALG-1 co-IP versus input are shown in red. Normalized reads for other classes of small RNAs (piRNAs and siRNA loci) are shown in grey. (B) Normalized reads for each miRNA in *alg-1(gk214)* versus wild type. (C) Normalized reads for each miRNA in HA::ALG-2 co-IP versus input are shown in red. Normalized reads for other classes of small RNAs (piRNAs and siRNA loci) are shown in grey. (D) Normalized reads for each miRNA in *alg-2(ok304)* versus wild type. (E) Overlap of miRNAs enriched in HA::ALG-1 and HA::ALG-2 co-IPs >1 fold and HA::ALG-5 IP >2 fold (data from adult stage animals). (F-H) Numbers of miRNAs enriched in HA::ALG-5 (F), HA::ALG-1 (G), and HA::ALG-2 (H) co-IPs categorized by 5' nt. miRNAs are categorized by their 5' nt and the 5' nt of the opposing strand of the miRNA duplex. Only miRNA duplexes for which at least one strand was enriched in the corresponding co-IP are shown. Each bar corresponds to the total number of miRNA duplexes with each 5' nt combination and each 5' nt is shaded in a different color. See also Supplementary Figure S2.4 and Supplementary Tables S2.3-S2.5.

lysate in the co-IP), 6 were depleted in both HA::ALG-1 and HA::ALG-2 co-IPs and were thus unique to HA::ALG-5, at least in adult animals (Figure 2.4E and Supplementary Table S2.4). We were not able to identify unique sequence or structural features that might contribute to binding specificity amongst the Argonautes. Each of the three Argonautes preferentially bound miRNAs beginning with a uridine, although a greater proportion of miRNAs associated with HA::ALG-1 and HA::ALG-2 contained a 5' uridine compared to HA::ALG-5 (Figures 2.4F-H).

Our small RNA high-throughput sequencing datasets from the co-IPs of GFP::ALG-5, HA::ALG-1, and HA::ALG-2 provided us with the opportunity to search for new miRNAs that might normally be missed due to their low abundance in whole animal cell lysates. Despite strong enrichment for miRNAs in these datasets, we identified only three new miRNAs, indicating that through the numerous high-throughput sequencing efforts, *C. elegans* miRNA identification is approaching saturation. miR-candidate-1 is derived from a unique miRNA-generating locus and contains a novel seed sequence - positions 2-8, which are largely responsible for conferring miRNA-target recognition - placing it in a new miRNA family (Figure 2.5A) [7]. The other two miRNAs are derived from genomic loci antisense (miR-candidate-2) or adjacent (miR-candidate-3) to annotated miRNA loci (Figures 2.5B and C). miR-candidate-2 contains a novel seed sequence, thus defining a second new miRNA family (Figure 2.5B). miR-candidate-3 shares a seed sequence with the miR-58/bantam family, and although not validated, was previously predicted to be a miRNA and shown to be downregulated in aged animals (Figure 2.5C) [51].

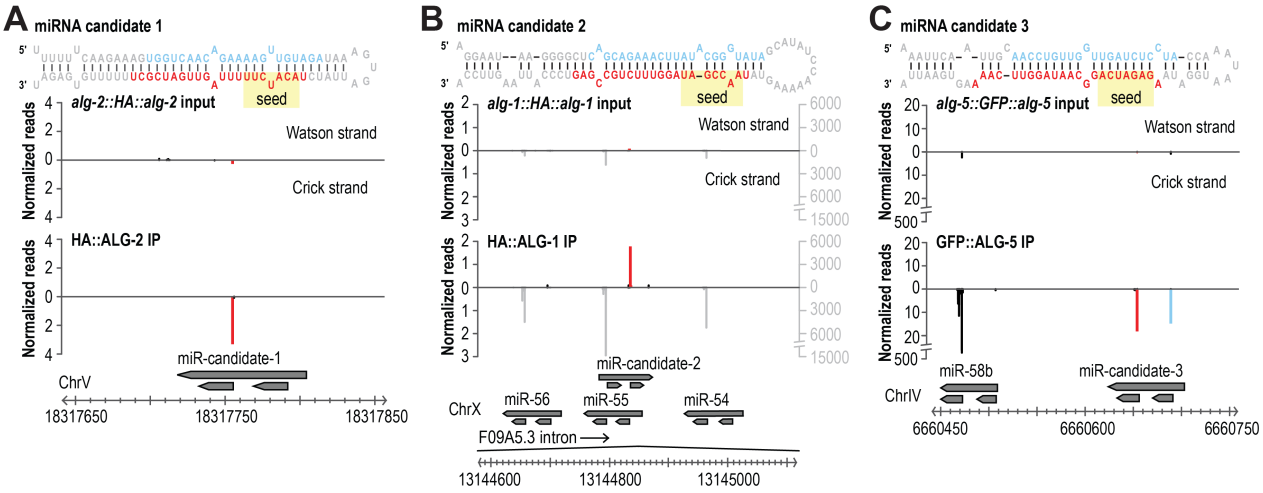


Figure 2.5. New miRNAs identified from Argonaute co-IPs. (A-C) miRNA candidates were identified by MirDeep2 using high-throughput sequencing data from GFP::ALG-5, HA::ALG-1, and HA::ALG-2 co-IPs. Small RNA distribution across each candidate locus in the co-IP library from which it was discovered and the corresponding input library.

2.3.5 Differential Gene Expression in *alg-5*, *alg-1*, and *alg-2* Mutants

To better understand the role of ALG-5 in regulating gene expression, we subjected total rRNA-depleted RNA from L4 stage wild type and *alg-5(ram2)* mutant animals to high-throughput mRNA sequencing. Differences in mRNA levels between wild type and mutant animals were quantified using Cuffdiff and HTSeq-count combined with DESeq [29-31] (Supplementary Tables S2.6 and S2.7). Applying a 1.5 fold change cutoff, we identified 88 upregulated and 235 downregulated genes in *alg-5(ram2)* using Cuffdiff (Figure 2.6A and Supplementary Tables S2.6 and S2.7). Because ALG-5 is expressed in the germline, in which several endogenous siRNA pathways function, we assessed whether the genes misregulated in *alg-5(ram2)* were targets of each of the germline siRNA pathways – Mutator, CSR-1, ALG-3/4, and ERGO-1. Amongst the

downregulated genes, ~26% are targets of siRNAs, representing a slight underrepresentation (1.6 fold) relative to what would be expected by chance, although Mutator targets were modestly enriched (~1.3 fold) (Figure 2.6B). Within the upregulated gene set, only CSR-1 targets were enriched (~1.2 fold) (Figure 2.6B). This was not unexpected given that ALG-5 functions in the germline and CSR-1 targets a large proportion of germline genes [24]. We next assessed using DAVID overrepresentation of specific cellular processes within the gene sets differentially regulated in *alg-5* mutants [32, 33]. In the set of downregulated genes, several gene ontology terms related to immunity and defense were significantly enriched ($p < 0.05$) (Figure 2.6C and Supplementary Table S2.8). No specific gene ontology terms were significantly enriched by DAVID analysis within the upregulated gene set, likely due in part to its small size (88 genes from our Cuffdiff analysis).

In *C. elegans,* miRNAs guide gene silencing by affecting decay or translational repression of mRNA targets. However, the individual contribution of these two modes of silencing is poorly understood [8]. It is possible that ALG-5 impacts gene expression through translational repression of its targets or that the genes that are misexpressed in *alg-5* mutants are downstream of the direct ALG-5 targets. Consistent with this possibility, we did not observe substantial enrichment for target sites (7-mers and 8-mers) of GFP::ALG-5-associated miRNAs within the mRNAs of genes up or downregulated in *alg-5* mutants (Supplementary Figure S2.5A). A similar lack of enrichment for targets sites was observed in miR-35 family mutants, in which genes that

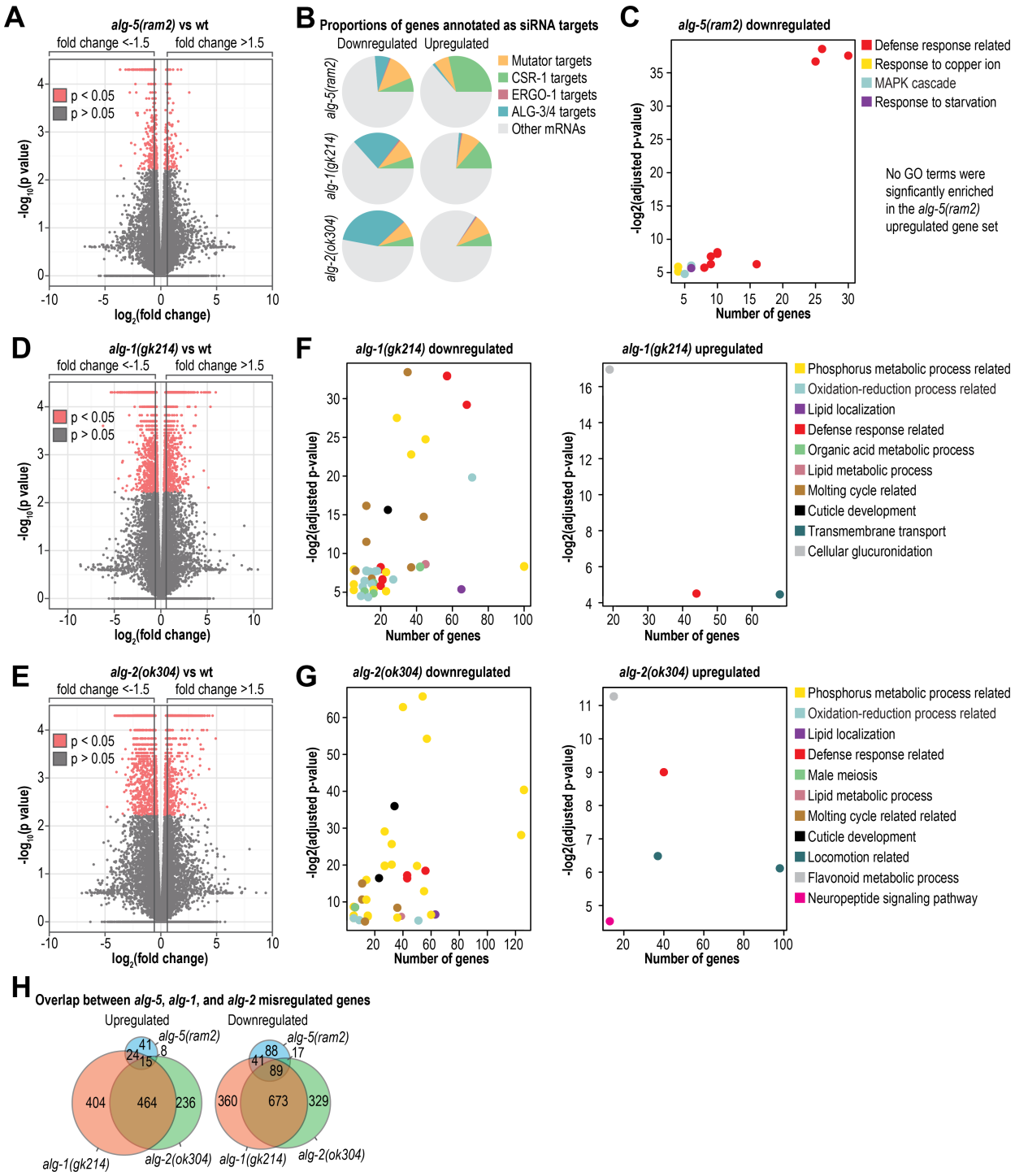


Figure 2.6. mRNA-seq analysis of differential gene expression in *alg-5*, *alg-1*, and *alg-2* mutants. (A) Volcano plot displaying differential gene expression between *alg-5(ram2)* mutants and wild type animals (n=3 replicate pools). (B) The proportions of genes misregulated in each of the Argonaute mutants that are also characterized as siRNA targets. (C) DAVID analysis of significantly enriched gene ontology terms amongst the genes misregulated in *alg-5(ram2)* mutants. Gene ontology categories are plotted as a function of the P value for enrichment and the number of genes associated with the

gene ontology term. Some gene ontology terms overlap in associated genes by more than 50% and were collapsed into a more general category, as indicated in the key (for example, 'Defense response related'). (D) Volcano plot displaying differential gene expression between *alg-1(gk214)* mutants and wild type animals (n=3 replicate pools). (E) Volcano plot displaying differential gene expression between *alg-2(ok214)* mutants and wild type animals (n=3 replicate pools). (F) Same as in C but *alg-1(gk214)*. (G) Same as in C but *alg-2(ok304)*. (H) Overlap in misregulated genes in each of the Argonaute mutants. See also Supplementary Figure S2.5 and Supplementary Tables S2.6-S2.16.

are misregulated are not enriched for miR-35 target sites [52]. Nonetheless, the results suggest a role for ALG-5, be it direct or indirect, in regulating genes involved in development and defense response pathways.

In parallel to *alg-5(ram2)*, we assessed changes in gene expression in *alg-1(gk214)* and *alg-2(ok304)* mutant L4 stage animals. The *alg-1(gk214)* allele is a 220 bp deletion-13 bp insertion that deletes an exon-intron junction at the 5' end of the coding sequence and would likely lead to a frame shift (Supplementary Figure S2.5B). The *alg-2(ok304)* allele is a 1,378 bp deletion spanning much of the open reading frame (Supplementary Figure S2.5C). Both mRNAs are still expressed, although at much lower levels than from the wild type alleles (Supplementary Figures S2.5B and C). In *alg-1(gk214)*, 907 genes were upregulated >1.5 fold and 1,163 genes were downregulated >1.5 fold (Figure 2.6D and Supplementary Tables S2.9 and S2.10). In total, 2,070 genes were misexpressed in *alg-1(gk214)*, representing nearly 10% of *C. elegans* protein coding genes. We were careful to stage match animals and removed developmentally delayed individuals from the pools of *alg-1(gk214)* mutants before collecting them for RNA

isolation, however, it is possible that some of the genes misregulated in our dataset are artifacts of developmental abnormalities in *alg-1(gk214).* In *alg-2(ok304)* mutants, which do not display obvious development abnormalities, we identified 1,831 genes that were misregulated by >1.5 fold, of which 723 genes were upregulated and 1,108 were downregulated (Figure 2.6 and Supplementary Tables S2.11 and S2.12). Numerous gene ontology terms were significantly enriched amongst the *alg-1(gk214)* and *alg-2(ok304)* misregulated gene sets, including defense response related terms (Figures 2.6F and G; Supplementary Tables S2.13-S2.16). However, the most highly enriched gene ontology terms identified amongst the *alg-1(gk214)* and *alg-2(ok304)* downregulated gene sets were related to phosphorus metabolism and protein phosphorylation and dephosphorylation (Figures 2.6F and G; Supplementary Tables S2.13 and S2.15). Classic targets of let-7 and lin-4, such as *lin-41* and *lin-14* respectively, were also amongst the genes significantly upregulated in *alg-1(gk214)* and *alg-2(ok304)* mutants (Supplementary Tables S2.10 and S2.12). Interestingly, and for reasons unclear to us, a large proportion of the genes downregulated in *alg-2(ok304)*, and to a lesser extent *alg-1(gk214)*, are also targets of the ALG-3/ALG-4 26G-RNA pathway that functions during sperm development (Figure 2.6B) [53-55].

There was substantial overlap in the genes misregulated in each of the miRNA-associated Argonaute mutants, particularly amongst the downregulated gene sets (Figure 2.6H). This is not unexpected given the overlap in miRNAs associated with each Argonaute (Figure 2.4E). However, given the differences we observed in expression of

the Argonautes across developmental stages and their presence or absence in the germline (Figure 2.2), it is possible that there is tissue and timing specificity for each Argonaute even in regulating overlapping gene sets. We did not observe substantial enrichment for 7-mer and 8-mer target sites of miRNAs associated with HA::ALG-1 and HA::ALG-2 in the gene sets upregulated in the corresponding mutants (Supplementary Figures S2.5D and E). However, ~75% of *C. elegans* genes are predicted to contain 7-mer or 8-mer target sites for miRNAs associated with HA::ALG-1 and HA::ALG-2 and thus there is very little room for enrichment (Supplementary Figures S2.5D and E) [37, 38].

It is likely that many miRNA targets were missed in our analysis because of functional redundancy amongst the Argonautes and because our whole animal approach may dilute cell or tissue specific effects. It is also possible that many targets cannot be identified by RNA-seq because in some instances miRNAs may function in translational repression and not in mRNA decay, as noted above.

2.3.6 Functional Overlap Between the miRNA-Associated Argonautes

ALG-1 and ALG-2 have overlapping roles in development [39, 44]. To determine if ALG-5 has an overlapping role with ALG-1 or ALG-2, we introduced the *alg-5(ram2)* mutation into *alg-1(gk214)* and *alg-2(ok304)* mutant animals. *alg*-5 did not enhance the brood size defects of *alg-1* or *alg-2* mutants, nor did we observe additional developmental abnormalities not observed in the single mutants (Supplementary Figures S2.6A and B).

It is nonetheless possible that there is redundancy between ALG-5 and the other Argonautes that would emerge from a more detailed analysis.

alg-2; *alg-1* double mutants arrest during embryogenesis [39, 44]. Suppressing *alg-2* by RNAi in *alg-1(gk214)* mutants during early larval stages also leads to developmental arrest, suggesting that ALG-1 and ALG-2 have overlapping roles during both embryo and larval development (Supplementary Figure S2.6C) [48]. *alg-1* mutants are much sicker than *alg-2* mutants, thus it is possible that *alg-2* lacks certain functionality possessed by *alg-1*. To test this possibility, we developed a chimeric construct that contains the HA epitope sequence fused to the *alg-2* coding sequence and *alg-1* 5' and 3' regulatory sequences (*alg-1::HA::alg-2*) and introduced it into *alg-1(gk214)* mutant animals. HA::ALG-2 expressed under the control of *alg-1* regulatory elements displayed a similar expression profile to that of HA::ALG-1 (Figures 2.2C and 2.7A). Both the *alg-1::HA::alg-1* and *alg-1::HA::alg-2* transgenes rescued the developmental defects of *alg-1(gk214)* mutants, indicating ALG-2 is functionally interchangeable with ALG-1 (Figure 2.7B). The small RNA repertoire of HA::ALG-2 expressed from *alg-1* regulatory sequences had greater overlap with miRNAs uniquely bound by HA::ALG-1 than those uniquely bound by HA::ALG-2 (Figure 2.7C). This indicates that the difference we observed in miRNA specificity between HA::ALG-1 and HA::ALG-2 (Figure 2.4) does not reflect miRNA sequence or structure preferences of the two Argonautes and instead is likely due to developmental differences in *alg-1* and *alg-2* expression (Figures 2.2C and D).

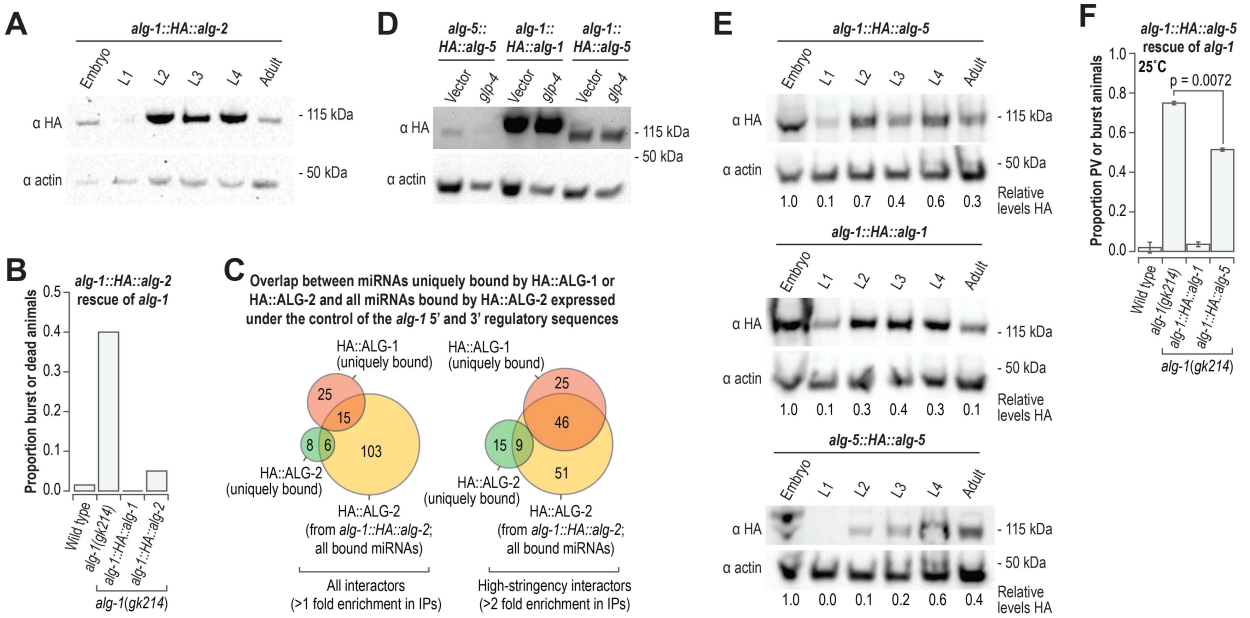


Figure 2.7. Functional overlap of *alg-5, alg-1,* and *alg-2*. (A) Western blot assay of HA::ALG-2 derived from a chimeric construct containing *alg-1* 5' and 3' regulatory sequence and *alg-2* coding sequence (*alg-1::HA::alg-2*). Actin is shown as a loading control. (B) Proportion of burst or dead animals. Wild type (n=152), *alg-1(gk214)* (n=114), *alg-1::HA::alg-1; alg-1(gk214)* (n=116), and *alg-1::HA::alg-2; alg-1(gk214)* (n=92) animals were grown at 20°C. (C) Overlap of miRNAs enriched >1 fold (left) and >2 fold (right) in co-IP of HA::ALG-2 derived from *alg-1::HA::alg-2; alg-1(gk214)* with miRNAs uniquely enriched in co-IPs from HA::ALG-1 (*alg-1::HA::alg-1; alg-1(gk214)*) or HA::ALG-2 (*alg-2::HA::alg-2; alg-2(ok304)*) co-IPs >1 fold (left) and >2 fold (right). (D) Western blot assay of HA::ALG-5 derived from a construct containing the authentic *alg-5* regulatory elements in the *alg-5(tm1163)* mutant background (*alg-5(tm1163); alg-5::HA::alg-5*) and a chimeric construct containing *alg-1* 5' and 3' regulatory sequences and *alg-5* coding sequence in the *alg-1(gk214)* mutant background (*alg-1::HA::alg-5; alg-1(gk214)*). HA::ALG-1 from *alg-1::HA::alg-1* in the *alg-1(gk214)* mutant background is also shown. Actin is shown as a loading control. *glp-4* RNAi was done to reduce germ cell proliferation during development. L4440 vector RNAi was done as a control. (E) A developmental time course of HA::ALG-5 from *alg-1::HA::alg-5; alg-1(gk214)* (upper panel) and *alg-5(tm1163); alg-5::HA::alg-5* (lower panel) and HA::ALG-1 from *alg-1::HA::alg-1; alg-1(gk214)* (middle panel). Actin is shown as a loading control. Numbers below blot images are signal intensities of HA normalized to actin (embryo samples arbitrarily set to 1.0). (F) Proportions of animals containing protruding or burst vulvas. Wild type (n=104-124), *alg-1(gk214)* (n=102-109), *alg-1::HA::alg-1; alg-1(gk214)* (n=109-114), and *alg-1::HA::alg-5; alg-1(gk214)* (n=102-116) animals were grown at 25˚C. See also Supplementary Figure S2.6.

Although we did not observe functional redundancy between *alg-5* and *alg-1*, we nonetheless tested whether ALG-5 is functionally interchangeable with ALG-1, as germline or somatic specificity in gene expression could preclude functional overlap during development. We developed a construct containing the HA epitope sequence fused to the *alg-5* coding sequencing and containing the *alg-1* 5' and 3' regulatory elements (*alg-1::HA::alg-5*) and introduced it by Mos1-mediated single copy integration into *C. elegans* [16]. We then crossed the *alg-1::HA::alg-5* transgene into *alg-1(gk214)* mutants. Given that *alg-5* is normally expressed primarily in the germline and *alg-1* is expressed in the soma (Figure 2.2), we tested whether *alg-5* would display somatic expression similar to *alg-1* when expressed from *alg-1* regulatory elements. To suppress germline development, we treated *alg-1::HA::alg-5*-transgenic animals with *glp-4* RNAi. As controls, we included *alg-1::HA::alg-1* and *alg-5::HA::alg-5* transgenic animals. ALG-5 levels in *alg-5(tm1163); alg-5::HA::alg-5* were moderately depleted upon treatment with *glp-4* RNAi compared to treatment with a vector control (Figure 2.7D). In contrast, the levels of ALG-5 produced from *alg-1::HA::alg-5; alg-1(gk214)* and ALG-1 from *alg-1::HA::alg-1; alg-1(gk214)* were unchanged between vector control RNAi and *glp-4* RNAi (Figure 2.7D). HA::ALG-5 protein was produced at higher levels when expressed from *alg-1::HA::alg-5* than when expressed from the authentic *alg-5* regulatory elements (*alg-5::HA::alg-5*), but did not appear to be produced at as high of levels as the HA::ALG-1 protein produced from *alg-1::HA::alg-1* (Figure 2.7D). Thus, it is likely that there are additional features that affect *alg-5* expression or the stability of the ALG-5 protein. The pattern of HA::ALG-5 expression from the *alg-1::HA::alg-5*

transgene across development was similar to that of HA::ALG-1 expressed from *alg-1::HA::alg-1* (Figure 2.7E).

Keeping in mind the caveat that *alg-1::HA::alg-5* does not produce as much protein as *alg-1::HA::alg-1*, we assessed whether *alg-1::HA::alg-5* would rescue the developmental defects in *alg-1(gk214)* mutants. A modest reduction in the proportion of animals displaying protruding or bursting vulvas was observed in *alg-1::HA::alg-5; alg-1(gk214)* animals relative to non-transgenic *alg-1(gk214)* animals when grown at 25˚C ($p = 0.0072$, Figure 2.7F). Thus, ALG-5 likely has some functional overlap with ALG-1 but differences in expression levels prevent us from drawing conclusions about the extent of such overlap.

2.4 DISCUSSION

2.4.1 ALG-5 as a Distinct Branch of the miRNA Pathway

ALG-5 likely defines a branch of the miRNA pathway largely distinct from that of ALG-1 and ALG-2. ALG-1 and ALG-2 bind overlapping and extensive sets of miRNAs and function redundantly during embryogenesis and larval development [39, 44, 48]. In contrast, ALG-5 binds a very narrow subset of miRNAs and does not appear to have substantial functional overlap with ALG-1 or ALG-2 despite the three Argonautes having many miRNA interactors in common. Unlike ALG-1 and ALG-2, with central roles in embryogenesis and larval development, ALG-5 appears to have a specific role in developmental timing in the germline. *alg-5* is expressed primarily, if not exclusively, in

the germline. *alg-5* mutants display a slight reduction in the number of progeny they produce and an accelerated transition from spermatogenesis to oogenesis. We identified several genes misregulated in *alg-5(ram2*) mutants that could contribute to the observed phenotype*,* including genes regulating the MAPK pathway, DNA-damage response, and apoptosis. Numerous genes involved in immunity and defense also emerged as being downregulated in *alg-5* mutants. Interestingly, *alg-5* was identified in a screen for gene inactivations that cause hypersensitivity to bacterial pore-forming toxins, hence its original name *hypersensitive to pore-forming toxin-24* (*hpo-24*) [56]. Consistent with this role, two genes required for pore-forming toxin defense, the activator protein-1 (AP-1) transcription factors *jun-1* and *fos-1* [56] were downregulated in *alg-5* mutants. It will be important in future studies to identify the direct targets of ALG-5 and its precise function in germline development and pathogen defense.

2.4.2 ALG-5 Localization to P Granules

Several Argonautes have been shown to associate with perinuclear germ granules called P granules in *C. elegans*, including the piRNA-associated Argonaute PRG-1 and two siRNA-associated Argonautes, CSR-1 and WAGO-1, where they have important roles in both gene licensing and silencing [13, 24, 57]. Our results demonstrate that ALG-5 also associates with P granules, indicating that miRNAs have a role in regulating gene expression in P granules as well. Interestingly, AIN-1, a GW182 protein orthologous to human TNRC6A, co-purifies with several P granule components [58]. GW182 proteins function as scaffolds between miRNA-associated Argonautes and

downstream effectors of miRNA-mediated silencing [8]. Based on sequence alignment of ALG-5 with human AGO2 and ALG-1, the amino acid residues in the tryptophan binding pockets that facilitate GW182-Argonaute interactions appear to be conserved (Supplementary Figure S2.1B) [59, 60]. Thus, AIN-1 may interact with ALG-5 within P granules to mediate RNA silencing.

2.4.3 Functional Similarity Between the miRNA-Associated Argonautes

ALG-1 but not ALG-2 is required for normal development, despite that the two genes are nearly identical in amino acid sequence (88% identity). ALG-2 expressed under the control of the *alg-1* regulatory sequences largely rescues developmental defects in *alg-1* mutants, indicating that differences in gene expression and not molecular functionality likely distinguish ALG-1 and ALG-2. ALG-5 shares only ~36% identity with ALG-1 and ALG-2 and is thus unlikely to have identical molecular functionality. ALG-5 lacks the conserved RNaseH residues that confer slicer activity and which are present in ALG-1 and ALG-2, pointing to at least one functional difference between the proteins [10]. Nonetheless, when introduced into an *alg-1* mutant under the control of *alg-1* non-coding regulatory sequences, we did observe partial rescue of the *alg-1* mutant phenotype by *alg-5*. The ALG-5 branch of the AGO subfamily is highly conserved across *Caenorhabditis* species estimated to be separated from *C. elegans* by at least 110 million generations [61, 62], pointing to ancient divergence of ALG-5 from ALG-1 and ALG-2.

2.4.4 A Near-Comprehensive miRNA-Argonaute Interactome

Our datasets provide a near-comprehensive analysis of miRNA-Argonaute interactions in *C. elegans*. The majority (~80%) of annotated miRNAs, including the strands often annotated as star or passenger, were identified in our analysis of Argonaute-small RNA interactions and were enriched in libraries from at least one Argonaute co-immunoprecipitate (co-IP), although many were present at levels below our stringent 10 normalized reads (reads per million total mapped reads) threshold. Many miRNAs were not enriched in any of the Argonaute co-IPs or were completely absent in all of our datasets. Several of the miRNAs not enriched in any of the Argonaute co-IPs are presumed miRNA stars. However, notably absent guide strand miRNAs include miR-261 and miR-264-miR-273, which were identified computationally [63] but have not been validated using sequencing-based approaches, and miR-4930-miR-4935, many of which are enriched in aged animals [64]. The three newly discovered miRNAs represent two new miRNA families and a new member of the miR-58/bantam family. Each of the miRNAs is expressed at relatively low levels, likely explaining why they were not identified or validated in previous analyses. Although it is likely that miRNA identification is approaching saturation in *C. elegans*, new miRNAs continue to be discovered and will likely continue to emerge from analyses of animals grown under non-standard laboratory conditions, such as drug treatment, pathogen exposure, and environmental stress.

The various roles of miRNAs and their specific functions in *C. elegans* gene regulation are still poorly understood. The identification of ALG-5 and the near comprehensive

analysis of miRNA-Argonaute interactions presented here will provide a valuable framework for discovering new roles for miRNAs in development and disease.

ACCESSION NUMBERS

All the high-throughput sequencing data described here has been deposited to the Gene Expression Omnibus (GEO) and is available under accession number GSE98935.

SUPPLEMENTARY DATA

Supplementary Data can be found in Appendix I beginning on page 124.

FUNDING

This work was supported by Colorado State University [laboratory startup funds to T.A.M.], the Boettcher Foundation [003614-00002 to T.A.M.], the NIH [1R35GM119775-01 to T.A.M.], and a Department of Education GAANN fellowship [to K.C.B.].

REFERENCES

1. Bartel, D.P., *MicroRNAs: genomics, biogenesis, mechanism, and function.* Cell, 2004. **116**(2): p. 281-97.
2. Lim, L.P., et al., *The microRNAs of Caenorhabditis elegans.* Genes Dev, 2003. **17**(8): p. 991-1008.
3. Lee, Y., et al., *The nuclear RNase III Drosha initiates microRNA processing.* Nature, 2003. **425**(6956): p. 415-9.
4. Khvorova, A., A. Reynolds, and S.D. Jayasena, *Functional siRNAs and miRNAs exhibit strand bias.* Cell, 2003. **115**(2): p. 209-16.
5. Schwarz, D.S., et al., *Asymmetry in the assembly of the RNAi enzyme complex.* Cell, 2003. **115**(2): p. 199-208.
6. Liu, J., et al., *Argonaute2 is the catalytic engine of mammalian RNAi.* Science, 2004. **305**(5689): p. 1437-41.
7. Bartel, D.P., *MicroRNAs: target recognition and regulatory functions.* Cell, 2009. **136**(2): p. 215-33.
8. Jonas, S. and E. Izaurralde, *Towards a molecular understanding of microRNA-mediated gene silencing.* Nat Rev Genet, 2015. **16**(7): p. 421-33.
9. Claycomb, J.M., *Ancient endo-siRNA pathways reveal new tricks.* Curr Biol, 2014. **24**(15): p. R703-15.
10. Tolia, N.H. and L. Joshua-Tor, *Slicer and the argonautes.* Nat Chem Biol, 2007. **3**(1): p. 36-43.
11. Grishok, A., *Biology and Mechanisms of Short RNAs in Caenorhabditis elegans.* Adv Genet, 2013. **83**: p. 1-69.
12. Correa, R.L., et al., *MicroRNA-directed siRNA biogenesis in Caenorhabditis elegans.* PLoS Genet, 2010. **6**(4): p. e1000903.
13. Gu, W., et al., *Distinct argonaute-mediated 22G-RNA pathways direct genome surveillance in the C. elegans germline.* Mol Cell, 2009. **36**(2): p. 231-44.
14. Montgomery, T.A., et al., *Specificity of ARGONAUTE7-miR390 interaction and dual functionality in TAS3 trans-acting siRNA formation.* Cell, 2008. **133**(1): p. 128-41.
15. Phillips, C.M., et al., *MUT-16 promotes formation of perinuclear mutator foci required for RNA silencing in the C. elegans germline.* Genes Dev, 2012. **26**(13): p. 1433-44.
16. Frokjaer-Jensen, C., et al., *Single-copy insertion of transgenes in Caenorhabditis elegans.* Nat Genet, 2008. **40**(11): p. 1375-83.
17. Dickinson, D.J., et al., *Engineering the Caenorhabditis elegans genome using Cas9-triggered homologous recombination.* Nat Methods, 2013. **10**(10): p. 1028-34.
18. Dickinson, D.J., et al., *Streamlined Genome Engineering with a Self-Excising Drug Selection Cassette.* Genetics, 2015. **200**(4): p. 1035-49.
19. Brenner, S., *The genetics of Caenorhabditis elegans.* Genetics, 1974. **77**(1): p. 71-94.
20. Larkin, M.A., et al., *Clustal W and Clustal X version 2.0.* Bioinformatics, 2007. **23**(21): p. 2947-8.
21. Felsenstein, J., *PHYLIP - Phylogeny Inference Package (Version 3.2).* Cladistics, 1989. **5**: p. 164-166.
22. Kamath, R.S., et al., *Systematic functional analysis of the Caenorhabditis elegans genome using RNAi.* Nature, 2003. **421**(6920): p. 231-7.

23. Fahlgren, N., et al., *Computational and analytical framework for small RNA profiling by high-throughput sequencing.* RNA, 2009. **15**(5): p. 992-1002.
24. Claycomb, J.M., et al., *The Argonaute CSR-1 and its 22G-RNA cofactors are required for holocentric chromosome segregation.* Cell, 2009. **139**(1): p. 123-34.
25. Mackowiak, S.D., *Identification of novel and known miRNAs in deep-sequencing data with miRDeep2.* Curr Protoc Bioinformatics, 2011. **Chapter 12**: p. Unit 12 10.
26. Zhang, Z., et al., *Strand-specific libraries for high throughput RNA sequencing (RNA-Seq) prepared without poly(A) selection.* Silence, 2012. **3**(1): p. 9.
27. Bolger, A.M., M. Lohse, and B. Usadel, *Trimmomatic: a flexible trimmer for Illumina sequence data.* Bioinformatics, 2014. **30**(15): p. 2114-20.
28. Kim, D., et al., *TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions.* Genome Biol, 2013. **14**(4): p. R36.
29. Trapnell, C., et al., *Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks.* Nat Protoc, 2012. **7**(3): p. 562-78.
30. Anders, S., P.T. Pyl, and W. Huber, *HTSeq--a Python framework to work with high-throughput sequencing data.* Bioinformatics, 2015. **31**(2): p. 166-9.
31. Love, M.I., W. Huber, and S. Anders, *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.* Genome Biol, 2014. **15**(12): p. 550.
32. Huang da, W., B.T. Sherman, and R.A. Lempicki, *Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.* Nat Protoc, 2009. **4**(1): p. 44-57.
33. Huang da, W., B.T. Sherman, and R.A. Lempicki, *Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists.* Nucleic Acids Res, 2009. **37**(1): p. 1-13.
34. Hulsen, T., J. de Vlieg, and W. Alkema, *BioVenn - a web application for the comparison and visualization of biological lists using area-proportional Venn diagrams.* BMC Genomics, 2008. **9**: p. 488.
35. Robinson, J.T., et al., *Integrative genomics viewer.* Nat Biotechnol, 2011. **29**(1): p. 24-6.
36. Thorvaldsdottir, H., J.T. Robinson, and J.P. Mesirov, *Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration.* Brief Bioinform, 2013. **14**(2): p. 178-92.
37. Jan, C.H., et al., *Formation, regulation and evolution of Caenorhabditis elegans 3'UTRs.* Nature, 2011. **469**(7328): p. 97-101.
38. Lewis, B.P., C.B. Burge, and D.P. Bartel, *Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets.* Cell, 2005. **120**(1): p. 15-20.
39. Grishok, A., et al., *Genes and mechanisms related to RNA interference regulate expression of the small temporal RNAs that control C. elegans developmental timing.* Cell, 2001. **106**(1): p. 23-34.
40. Bukhari, S.I., et al., *The microRNA pathway controls germ cell proliferation and differentiation in C. elegans.* Cell Res, 2012. **22**(6): p. 1034-45.
41. Yuan, Y.R., et al., *Crystal structure of A. aeolicus argonaute, a site-specific DNA-guided endoribonuclease, provides insights into RISC-mediated mRNA cleavage.* Mol Cell, 2005. **19**(3): p. 405-19.

42. Cutter, A.D., *Sperm-limited fecundity in nematodes: how many sperm are enough?* Evolution, 2004. **58**(3): p. 651-5.
43. Hodgkin, J. and T.M. Barnes, *More is not better: brood size and population growth in a self-fertilizing nematode.* Proc Biol Sci, 1991. **246**(1315): p. 19-24.
44. Vasquez-Rifo, A., et al., *Developmental characterization of the microRNA-specific C. elegans Argonautes alg-1 and alg-2.* PLoS One, 2012. **7**(3): p. e33750.
45. Zinovyeva, A.Y., et al., *Mutations in conserved residues of the C. elegans microRNA Argonaute ALG-1 identify separable functions in ALG-1 miRISC loading and target repression.* PLoS Genet, 2014. **10**(4): p. e1004286.
46. Zinovyeva, A.Y., et al., *Caenorhabditis elegans ALG-1 antimorphic mutations uncover functions for Argonaute in microRNA guide strand selection and passenger strand disposal.* Proc Natl Acad Sci U S A, 2015. **112**(38): p. E5271-80.
47. Tops, B.B., R.H. Plasterk, and R.F. Ketting, *The Caenorhabditis elegans Argonautes ALG-1 and ALG-2: almost identical yet different.* Cold Spring Harb Symp Quant Biol, 2006. **71**: p. 189-94.
48. Bouasker, S. and M.J. Simard, *The slicing activity of miRNA-specific Argonautes is essential for the miRNA pathway in C. elegans.* Nucleic Acids Res, 2012. **40**(20): p. 10452-62.
49. Billi, A.C., S.E.J. Fischer, and J.K. Kim, *Endogenous RNAi pathways in C. elegans*, in *WormBook*, T.C.e.R. Community, Editor., WormBook.
50. Alvarez-Saavedra, E. and H.R. Horvitz, *Many families of C. elegans microRNAs are not essential for development or viability.* Curr Biol, 2010. **20**(4): p. 367-73.
51. Kato, M., et al., *Age-associated changes in expression of small, noncoding RNAs, including microRNAs, in C. elegans.* RNA, 2011. **17**(10): p. 1804-20.
52. Massirer, K.B., et al., *The miR-35-41 family of microRNAs regulates RNAi sensitivity in Caenorhabditis elegans.* PLoS Genet, 2012. **8**(3): p. e1002536.
53. Conine, C.C., et al., *Argonautes ALG-3 and ALG-4 are required for spermatogenesis-specific 26G-RNAs and thermotolerant sperm in Caenorhabditis elegans.* Proc Natl Acad Sci U S A, 2010. **107**(8): p. 3588-93.
54. Conine, C.C., et al., *Argonautes Promote Male Fertility and Provide a Paternal Memory of Germline Gene Expression in C. elegans.* Cell, 2013. **155**(7): p. 1532-44.
55. Han, T., et al., *26G endo-siRNAs regulate spermatogenic and zygotic gene expression in Caenorhabditis elegans.* Proc Natl Acad Sci U S A, 2009. **106**(44): p. 18674-9.
56. Kao, C.Y., et al., *Global functional analyses of cellular responses to pore-forming toxins.* PLoS Pathog, 2011. **7**(3): p. e1001314.
57. Batista, P.J., et al., *PRG-1 and 21U-RNAs interact to form the piRNA complex required for fertility in C. elegans.* Mol Cell, 2008. **31**(1): p. 67-78.
58. Wu, E., et al., *A continuum of mRNP complexes in embryonic microRNA-mediated silencing.* Nucleic Acids Res, 2017. **45**(4): p. 2081-2098.
59. Schirle, N.T. and I.J. MacRae, *The crystal structure of human Argonaute2.* Science, 2012. **336**(6084): p. 1037-40.
60. Jannot, G., et al., *GW182-Free microRNA Silencing Complex Controls Post-transcriptional Gene Expression during Caenorhabditis elegans Embryogenesis.* PLoS Genet, 2016. **12**(12): p. e1006484.
61. Shi, Z., et al., *High-throughput sequencing reveals extraordinary fluidity of miRNA, piRNA, and siRNA pathways in nematodes.* Genome Res, 2013. **23**(3): p. 497-508.

62. Cutter, A.D., A. Dey, and R.L. Murray, *Evolution of the Caenorhabditis elegans genome.* Mol Biol Evol, 2009. **26**(6): p. 1199-234.
63. Grad, Y., et al., *Computational and experimental identification of C. elegans microRNAs.* Mol Cell, 2003. **11**(5): p. 1253-63.
64. de Lencastre, A., et al., *MicroRNAs both promote and antagonize longevity in C. elegans.* Curr Biol, 2010. **20**(24): p. 2159-68.

3. AUTOMATED ANALYSIS OF SMALL RNAS WITH AQUATX

Much of my time as a PhD student was spent performing data analysis as a part of collaborative projects. This highlighted a need for simpler, robust tools for bench scientists and small RNA researchers. This chapter describes the development of a small RNA sequencing data analysis tool utilizing best practices described in Chapter 1 called AQuATx. It also describes additional small RNA analysis that was performed using this pipeline[1].

3.1 INTRODUCTION

With more scientists looking to generate large, next-generation sequencing datasets, it is important that the tools they use are robust, reproducible, user-friendly, and well-documented. Despite the growing popularity of small RNA research, few tools exist to comprehensively and easily analyze associated sequencing datasets (Table 3.1). While popular, well-defined data analysis workflows exist for mRNA sequencing, there is more confusion surrounding the best practices in small RNA analysis workflows. We encountered several issues with existing tools in the literature: 1) Are single step or partial analysis, 2) have complex installation procedures, 3) no longer exist or maintained, 4) not actively open-source developed/responsive to feedback, 5) only focus on miRNAs, 6) utilize tools developed for mRNA or 7) other usability restrictions. Some of these tools are also aimed at observing differences in conditions such as disease states, whereas there is a lack of tools with the study of small RNA pathways

[1] Thanks to Meng Cao, Wen Zhou, Jay Breidt, Dustin Updike, Tai Montgomery, and Joshua Svendsen for contributions to the work this chapter.

themselves in mind. A comparison of these tools and features are presented in Table 3.1 and was used to guide development decisions for a new small RNA workflow. We include only multi-step workflows and steps which are part of the pipeline.

In some cases, web applications exist for various bioinformatics tools, but the users are required to upload their data, presenting a challenge for those with very large data sets or clinical data which needs to be on a secure server to avoid violating HIPAA regulations. Additionally, the actual analysis is often a “black box”, with the code hidden from the public, unavailable for public scrutiny and improvement, reducing confidence in the results or it forcec you to use specific genomes or perform pre-processing on your own. While less common, there are certain considerations and analyses we often perform that require custom or outdated tools. For instance, we expect that disrupting components of a small RNA processing pathway will create large changes in small RNA abundance which may not be suitable for standard differential expression analysis or normalization. We also often study Argonaute co-immunoprecipitations and look at enrichment of small RNAs in the co-IP over repeated experiments, but not replicates. Such experiments may also lead to large, sweeping differences in samples that standard normalization and detection methods may not be appropriate for.

To address the concerns and gaps regarding small RNA data analysis, we developed AQuATx (Automated Quantitative Analysis of Transcript Expression), a simple, user-friendly pipeline for analyzing small RNA sequencing data. In developing this pipeline it

Table 3.1 Existing tools for small RNA research. A comparison of features for small RNA tools. Not included are tools that are single-step or highly specific tools that could be folded into a workflow (such as miRDeep2).

Name	Ref	Adapter removal	Quality control report	Read mapping	Read counting	DEG Analysis	Plots	Interface	Recently Updated?	Open source practices	Notes
bcbio-nextgen	-	atropos	FASTQC	multiple	mirDeep seqbuster			bcbio	Yes	Yes	
SPAR	[1]				custom		Size, class, expression	Web	Yes	No	Only certain genomes
UEA sRNA workbench	[2]	custom	custom	Patman	custom	multiple	RNA fold, coverage	GUI	Yes	No	Separate modules
sRNAnalyzer	[3]	cutadapt		bowtie	custom			Web	No	No	miRNA focused
CASHx	[4]	custom		custom				CLT	No	No	Dead website
Oasis2	[5]	custom	custom	bowtie	custom	DESeq2	PCA, heatmap	Web	Yes	No	Only certain genomes
iSRAP	[6]	trimmo matic	FASTQC	Bowtie2	Bedtools	multiple	Heatmap, feature histogram, MA plots	Web	No	No	Trims bases. Dead website
sRNAPipe	[7]			bwa	multiple		Size, class, coverage	Galaxy	Yes	Yes	
iSmaRT	[8]	cutadapt	FASTQC	custom	custom	multiple	PCA, heatmap	GUI	No	No	Dead website
CPSS 2.0	[9]			bowtie	custom	custom	Class, size, network, histogram	Web	No	No	Only certain genomes
RAPID	[10]			bowtie2	bedtools	DESeq2	Size, heatmap, coverage, PCA, histogram	CLT	Yes	Yes	Separate modules

was important to us that it be usable on a personal computer, implements best practices for small RNA data and scientific reproducibility, incorporates software-engineering guided best practices, as well as being very simple for an end-user to install and run. We also present use-cases for analysis with new and existing datasets to demonstrate the functionality pipeline.

3.2 IMPLEMENTATION

The main workflow and individual steps are written in Python and the Common Workflow Language (CWL) [11]. Third-party tools used in the software were written in several languages, including Python, C/C++, and R. When we did not implement third-party tools, we created them using Python. The overall workflow is described in Figure 3.1. The main implementation is an end-to-end analysis run with cwltool (https://github.com/common-workflow-language/cwltool), but individual steps or partial workflows may be run as well. We also provide options to export workflows as CWL to run using a different implementation. Support for Nextflow [12] and SnakeMake [13] will be added as well, as they are supporting CWL implementations, but needs further testing and validation.

3.2.1 Overall workflow development

Python is one of the fastest growing general-purpose programming languages and is popular among developers [14, 15]. Due to the simplicity of the language and the popularity, it should be easier for other scientists to contribute to the software through

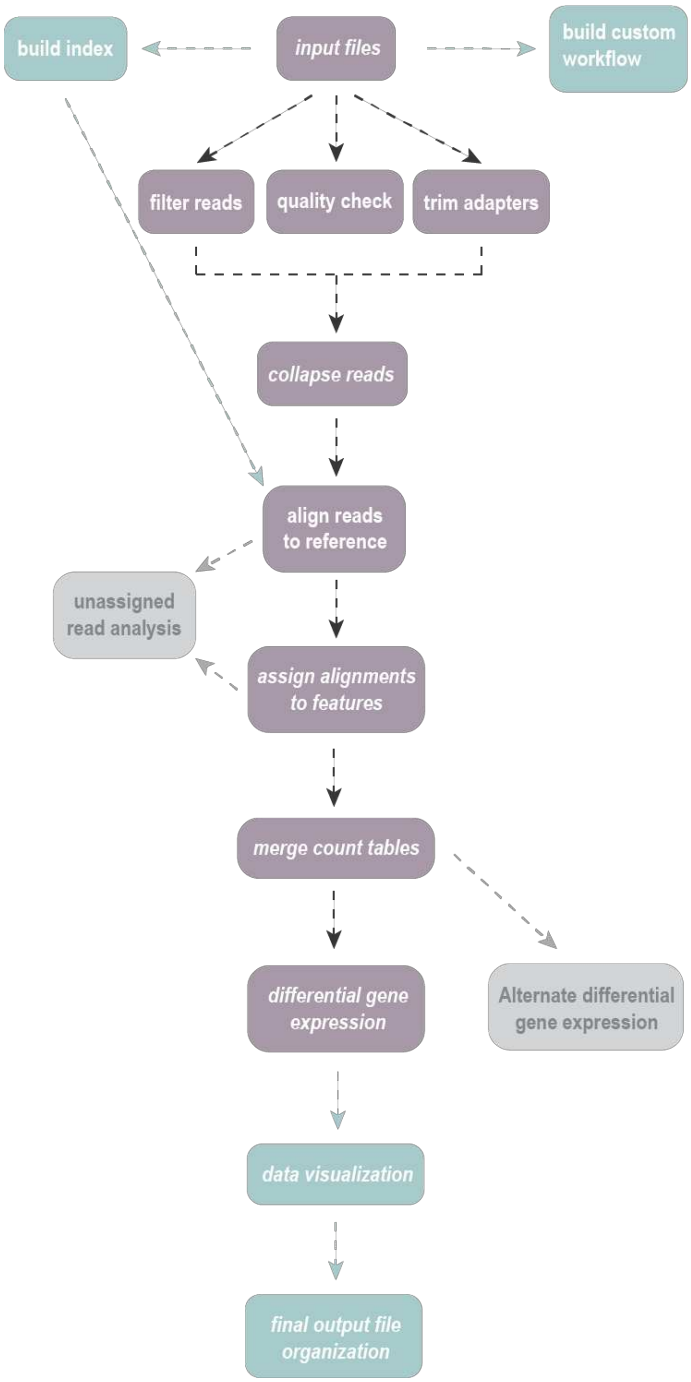


Figure 3.1 The AQuATx workflow. A flowchart describing the steps involved in a small RNA sequencing data analysis workflow. Steps are part of the CWL workflow in v0.1 (purple), v0.5 (green), and v1.0 (grey). Filter reads, quality check, and trim adapters (purple) are done using fastp as a single step. Align reads to reference (purple) and bowtie-build (green) is performed using bowtie. Differential gene expression (purple) is done with DESeq2. Unassigned read analysis (grey) will contain multiple tools. Alternate differential gene expression (grey) will be implemented as an R package. Everything else is implemented in Python or R specifically for AQuATx.

GitHub. Over the last few years, many workflow engines and implementations have been created to improve reproducibility among scientific fields with a lot of data to process [16-18]. This has become particularly popular within the bioinformatics field with the increasing amount of next-generation sequencing (NGS) data biologists produce. However, due to the sheer number of competing workflow engines, the interoperability of any particular lab's workflow may turn out to be quite low. CWL is an implementation-agnostic workflow language. Rather than compete with implementations, CWL aims to be a workflow standard from which other implementations can build off of [11]. There are already several implementations that can run CWL (cwltool for general-purpose use, CWLEXEC for LSF clusters, Arvados, Toil, Cromwell, etc), as well as workflow converters to take CWL to workflow languages such as Nextflow [12] and WDL [19], two other popular bioinformatics workflow languages [18]. Thus, we created the all-in-one workflow in CWL to maximize interoperability among users and institutions. CWL also forces very explicit definition of inputs, outputs, and steps, which creates a more reproducible workflow. The workflow is outlined in Figure 3.1. Additionally, CWL allows for developers to create workflows programmatically, so we are also using it to allow users to modularize or use subsets of the workflow if desired.

3.2.2 User input and configuration

We use a configuration file and sample sheets to set up each run, rather than individual command line arguments, which can get lengthy when you want to change many options. In order to force users to diligently keep track of each run's parameters, we

save a copy of configuration files and sample sheets, along with run time, date, user information in a separate folder per run. This allows a user to trace back any experiment's analysis without much hassle. The sample sheets are CSV format, which are easily edited in spreadsheet programs like Excel. The configuration file is a simple text file that allows a user to modify parameters at each step of the workflow, but otherwise sets defaults that we have deemed "best practices" for small RNA data. This allows the user to run a workflow without much input, but also allows them to choose more appropriate parameters if they wish. See Appendix II (Box S3.1, Tables S3.1-S3.2) for the configuration files and instructions on usage. The workflow code reads the configuration file (which is a standard YAML format) using the ruamel.yaml Python library and the csv inputs to figure out and update the settings for running with cwltool or other CWL runners. When the YAML is read by the code, all entries can be accessed as a normal dictionary and entries are updated based on the sample & reference sheets to reflect the CWL input format, and output to a new YAML file with the correct format.

3.2.3 The AQuATx Standard Workflow

The standard workflow performs an end-to-end analysis from raw fastq sequencing data to differential gene expression & visualizations. The steps and implementation are described in more detail below. To create a first-pass, simple workflow to process small RNA data, we evaluated and selected among tools that are commonly used at each step for processing NGS data. Some steps are implemented with custom tooling if no

tools seemed appropriate or were simpler to implement. See Appendix II for more information on running and installing the pipeline (User Guide).

1. Adapter trimming & quality filtering

The first step is pre-processing the sequence data, usually in a fastq format. This contains each sequence tied to quality scores per base. Due to the short length of small RNAs, the likelihood of adapters being a part of the sequence is very high, in fact, we expect that all sequenced small RNAs are attached to an adapter sequence. Thus, an adapter-removal step is essential. We settled on using fastp [20] for adapter trimming as the tool spans functionality of multiple useful pre-processing tools while remaining efficient. In particular, fastp also allows for quality filtering and trimming, filtering for a min/max length, adapter detection, as well as producing quality control statistics that can be used to evaluate sequencing quality [20]. We have contributed a general-purpose CWL wrapper for fastp to the workflow repository (https://github.com/common-workflow-language/workflows) in addition to providing a small RNA specific wrapper with the aquatx installation.

2. Counting & collapsing duplicate sequences

Next, due to the small size, the sequences are often duplicated in the data and we expect them to represent real expression instead of mainly PCR duplication for most datasets, unlike we expect with mRNA data. In order to avoid re-aligning the same sequence multiple times, we “collapse” the sequences down to a fasta file. Each entry is

a unique sequence with the header containing the number of duplicate reads in that sample. We implemented this step in Python using a simple approach. The sequence is read in 4 lines at a time for fastq files and added to a Counter dictionary – a specialized dictionary for counting entries. The file is then filtered for a threshold number of reads specified by the user and matching entries are written to a new fasta file. A potential improvement to be made is splitting files to count independently then merging for reduced memory usage and ability to process in parallel. This could be implemented using a split fastq file output from Fastp as an option.

3. Sequence alignment to a reference

The fasta files are then aligned to a reference genome using bowtie [21, 22]. We chose bowtie over other alignment programs because it is easily used on a personal computer, allowing users to process data without a high-memory machine, as well as the ability to force exact matches and report all alignments [21, 22]. We check all alignments due to the expected multi-mapping of small RNA, especially in transposable elements. We will also provide an option to use bowtie2 or STAR [23]for alignment, which may be more suitable in some instances or for realignment and comparisons. Initially bowtie2 was chosen, but bowtie was ultimately picked because of its ability to perform exact matches on the entire sequence where bowtie2 seems to only match on a seed. Matching the entire sequence exactly (or allowing for a specified number of mismatches) was one of the requirements I thought could be met with bowtie2, but could not. Unaligned files are saved to a separate fasta file for further evaluation, if desired. An improvement on this

step to be made is to add an optional realignment step of unaligned sequences allowing for a single mismatch or using a non-exact approach.

4. Assigning counts to small RNA features

The aligned sequences are then counted using Python and the HTSeq API [24]. We created a counter script for processing small RNA alignments in order to allow for multimappers, class counting, keeping track of unannotated reads, and feature masking. This code works by first reading in the reference files and creating independent reference arrays which assign features to positions from a reference GFF3 file and adding any overlapping “mask” features from the masked GFF3 file. A separate reference array is created for each reference-mask pair in the configuration. If no mask is specified, then an empty mask file is used. If a small RNA overlaps with another within the same feature set, the reads are not counted. For example, a small RNA transcribed from within an intron of a gene may be more likely to become the annotated miRNA over an overlapping siRNA, and in order to avoid assigning those counts to siRNAs or both or discarding them altogether, a miRNA GFF3 file can be passed as a “mask”. The alignments are then read into the script and processed as bundles – each unique sequence is counted as one “bundle” and the number of alignments for that unique sequence is then used to correct multimappers. For these multimapping reads, we simply divide the total counts, pulled from the sequence “name” where the collapser tracks the number of duplicates, by the number of genome-matching hits before assigning the counts to a particular feature to avoid over-inflating the value. Each

alignment within a bundle is then matched against each reference array for overlap. When no feature is found, the alignment counts are assigned to “unaligned” reads in the Counter dictionary. If the feature contains the “_mask” identifier, the counts are not added to that feature. If there is more than one feature within a region where the alignment falls, the counts are not added to any of the features. Different classes are counted alongside features in a separate Counter dictionary. If sequences match to more than one class, it is assigned to “ambiguous”. Class is determined by the third column in the GFF3 file. These counts and overall alignment/count statistics are then stored for later steps. Initially this step was written using popular libraries pandas and numpy to avoid yet another dependency, but the memory usage of pandas data frames was often quite large and slowed down processing due to the lack of vectorized operations for the task we were performing. Other numpy-only or pandas-only solutions were attempted, but ultimately the memory and time used was greater than desired. Using the HTSeq library allowed us to take advantage of the C-optimized data structures and code specific to this analysis task and use for-loops for processing while maintaining speed and minimizing memory usage. This also allows the code to be much less mysterious and easier to digest for those not familiar with pandas and numpy operations. Further improvements to this code include the addition of a “smart masker”, such that instead of reading in multiple references with a paired mask file for each then creating just as many arrays to sort through for each alignment, we can use a single reference file and no mask file. The user would then specify the “order of assignment”, such that the top choice is given preference in the case that there is overlap with

multiple features. If there is no preference given or features are within the same class then overlapping reads will not be counted. This should speed up the counting process, reduce overall memory usage, and simplify the configuration input as well.

5. Merging counts and statistics into tables

The count files are then merged into a larger table per sample for normalization, differential expression, and plotting. Advanced users may also simply run the counter script and pipe the counts into DESeq2 [25], as they might for mRNA-seq data, if they desire. Additionally, the single sample count files are saved in case the user wishes to discard certain poor-quality samples for reanalysis. This script reads in the single-sample files as pandas data frames, creates the new data frame of the correct size, then fills in the data frame per file in the set. This method of creating the data frame in the right final size ahead of time is more efficient with memory due to not needing to recreate and reallocate space for each intermediate data frame. This merge step can be performed for feature counts, class counts, and statistics tables.

6. Normalization & differential gene expression

For the first-pass workflow we have created an R wrapper script for DESeq2 [25] that performs a standard pairwise-comparison analysis across all samples in the dataset. The script can be run as an executable as aquatx-deseq with specified options. The lack of an argparse equivalent in base R led to the inclusion of a simple argument parser that is specific to the script and must be used exactly as specified. There is some

level of error-handling if the inputs are given incorrectly, but it is not as robust a method. The samples to compare are inferred by splitting the column names of the merged feature counts input file by "_replicate_" to determine group and replicate number. This format is automatically created by prior python scripts. If someone wishes to use this script outside of the aquatx pipeline, it would need to be formatted to reflect this. The table is integer rounded before creating the DESeq2 dataset. The script then returns a normalized count file and a table of differentially expressed gene statistics for each comparison. All comparisons are made by default. DESeq2 was created with mRNA sequencing data in mind. DESeq2 also expects integer counts and non-normalized data, but multi-mapping corrections can lead to fractional counts and without features fully annotated the library sizes may be skewed.

We will also provide a statistical module as an independent R package on Bioconductor (Cao et al, unpublished) and a wrapper in R for our workflow to process the data. This statistical method accounts for a zero-inflated dataset that is common in small RNA features, as well as being robust to large changes in different populations of small RNAs. Our lab and many others study small RNA pathways and sequencing data disrupting the small RNA processing pathways can often distribute the changes among small RNA features quite a lot – this can be difficult to perform proper differential expression analysis on using standard tooling. It also allows for fractional counts and calculating library size information for normalization purposes. DESeq2 may still be suitable for data comparing tissue types or treatment conditions, but we believe the

small RNA specific method is more robust for small RNA data in general. The workflow will likely contain the option to perform both or one of the two methods. Eventually, the method should be converted to Python to reduce excessive external dependencies on R and R packages.

7. Data visualization and summarization

The data is then summarized for the user with tables and plots. Several standard small RNA plots are included as PDF outputs that may be edited with a vector graphics tool such as Illustrator. We also provide a “smrna-light” stylesheet for matplotlib to create publication-ready plots that require little to no editing on the user’s part if desired. If the “smrna-light” theme is not installed then the “seaborn-poster” theme is used instead, but this can also be specified in configuration. The plots that are incorporated or to be incorporated for v0.5 include the QC plots from fastp output, size distribution plots of aligned reads, class bar and pie charts, enrichment barplots, scatter plots, PCA, and heatmaps. This code reads in the merged data tables as pandas data frames then creates plots based on the specified mode associated with the file given in the arguments. Each plot type is an independent function and future plots should be added as independent functions then called from the main function under each “mode”. The x and y axes should be inferred appropriately, either all comparisons are made or inferred using the same “_replicate_” structure used in the DESeq2 method.

8. Modularity and output file organization

While the implementation is presented as an “all-in-one” workflow to simplify data analysis for the scientist, advanced users may choose to perform individual steps in isolation. With CWL wrappers for each individual tool, users may also use just one step of the workflow or string together their own workflow building off of ours. The documentation in Appendix II provides more description of the toolkit for each step as well as an overall user guide. The workflow will allow users to update the configuration file to turn certain steps “on” or “off”. This allows the pipeline to be end-to-end by default, but modular as well. The default settings save all intermediate files created, but each step could have file saving turned off to maximize space. The workflow generator code is part of the v0.5 release and will use ruamel.yaml to read in the configuration as normal, but also to write the CWL workflow as specified. Each step is an entry in the overall workflow dictionary that can be output in YAML format. Thus, each step can be pasted together into a complete workflow depending on the configuration. For further analysis of unaligned and unannotated sequences, we recommend saving the “unaligned_files” and “annotation_tables” for downstream analysis. Additionally, the final step of the workflow will self-organize the output files into a “run” folder with the data and metadata information for the run, as well as outputs per step organized into folders. This could be implemented in either Python or CWL (with Javascript expressions).

3.2.4 Test Suite & Continuous Integration

In order to create a more “best practices” software package, we also implement tests for the software, including unit tests for individual functions, as well as tests on more involved steps using data we created. Bioinformatics software often lacks integrated test suites and instead relies on trusting that the developer has made no mistakes in the codebase or when updating the code. Software often contains errors ranging from simple typos in string or documentation that creates no issues to code that produces outputs that are incorrect. Testing is crucial to making sure the software is accurate at the time of creation, but more importantly that it *remains* correct as new maintainers come on board or code is refactored. Creating tests for bioinformatics can be challenging, but necessary. We use a small, exact, well-defined data set to run through the pipeline and tests to check if the results are as expected. We use continuous integration with TravisCI on GitHub in order to maximize maintainability. For contributors, this allows maintainers to quickly know whether or not the changes introduced also break the expected functionality or in the accuracy of the code & where that change occurred. As new features are added, we also encourage contributors to add tests with the unittest package or at the very least define the expected outputs for a particular input so that we may implement a new test. Integrated testing can also force good code style using PEP8 guidelines and a linter tool. We hope that these software-engineering guided principles will allow the software to be maintained for years to come and give users confidence in the robustness of the tool. We also clearly define a

template for submitting bugs and issues, feature suggestions, and contributing guidelines.

3.2.5 Steps to be implemented

At the time of this dissertation, not all the desired steps have been implemented. The standard workflow and testing suite represent the minimal viable product for performing the analysis and is the minimum portion that should be complete by submission. This workflow will be released as v0.5 on GitHub and added to BioConda for wider use and testing. The initial public commit to the current repository is considered v0.1 and has functionality through differential gene expression analysis. The project board for v0.5 contains all the specific tasks to be developed or that have been completed here: https://github.com/MontgomeryLab/aquatx-srna/projects/1. There are additional steps described here to be implemented prior to publication, but after submission of the dissertation. The additional steps described here will make the package overall better, more user-friendly, and provide more options. AQuATx will then be a more complete software tool for publication and could probably be released as v1.0, followed by smaller incremental releases. The description below includes the basic idea behind the step and guidelines for implementation, in case an additional developer is brought on to complete these.

1. Pre-configured Pipelines

In striving for maximum simplicity for the end-user, we will provide pre-defined configurations for model organisms and invite contributions in this area. Users may download these configurations individually. Each configuration provides the genome, bowtie index, and reference annotations for major classes of small RNAs. To fold it into the pipeline, the developer should upload the datasets to an individual GitHub repository or another data repository that includes versioning, which is essential in my opinion. Datasets made publically available should include versions with each update so that authors may reference specific versions to maximize reproducibility in future analyses. For folding into the AQuATx pipeline, the developer should include a method of fetching (online download tool, `aquatx-fetch homosapiens` for example, or allowing the user to download a compressed file that can be extracted by the program and installed directly) and installing this dataset, associated metadata should be stored and accessible, and inputs to the pipeline should be updated to reference datasets (keep track of file locations from install and use the absolute paths in the configuration files). First implementation and testing should be for *C. elegans* data as we have access to this to use immediately.

2. Options for non-model organisms

As small RNA sequencing becomes more widely integrated into a standard transcriptome analysis among biology labs, there is an increasing interest in analyzing small RNA data from non-model organisms, which lack well annotated genomes or

even proper reference genomes to align to. There are many possibilities for non-model organism analysis, but a few straightforward options for a standard analysis can be implemented for v1.0. Others will require more time and research to provide. Without a reference genome, one can look at the size and 5' nucleotide distribution of sequences to assess overall small RNA population among samples. A developer can implement this using the counter script already implemented, but adjusted for reading in a fasta file instead of a SAM/BAM file. This can then be folded into the workflow in CWL and a separate genome-free workflow can be created that runs through 1. Fastp, 2. Aquatx-collapse, 3. Counting, 4. Plots. At some point a de novo clustering algorithm to group sequences together as potential small RNA candidates would make a useful addition, but requires more computing power, algorithm testing, and development. Then a differential expression step per cluster could be incorporated for looking at generalized differences in sequences among samples. For those with reference genomes, size plots will also be produced, but after alignment as with other organisms. With a genome, but without annotation, we can use tools like mirDeep2 and piPipes to determine potential small RNA features, produce counts, and perform differential expression. These tools can be wrapped in CWL and folded into the pipeline as an optional step. A developer would need to incorporate the generation of this step into the code that generates workflows.

3. mRNA sequencing workflow

We also plan to package an independent workflow for processing mRNA data, similar to our small RNA pipeline. This should be uploaded to a separate GitHub repository, as there are different requirements that may not be desired for every user. However, the simplicity of use and consistent plot theme may be desirable to someone using our small RNA package. The best practices for this type of data are more commonly described and validated and we allow for more configuration in this regard. The chosen tools overlap with the small RNA workflow to minimize extra installations. The workflow should follow the following steps: 1. Preprocessing with fastp [20], 2. alignment that is configurable for HiSAT2 [26] or STAR [23] (must already be installed), 3. Feature counting with HTSeq [24], 4. Differential expression analysis with DESeq2 [25], 5. Plotting and summarization with Python. Each step should have a CWL wrapper around the tool and folded into the overall workflow. The DESeq2 wrapper should be usable as-is. Fastp should be adjusted to use default parameters for mRNA sequencing and perhaps allow for more configuration. File organization, workflow generation, and overall configuration can be implemented very similarly to the sRNA workflow and much of the existing code can likely be repurposed.

4. AQuATx Dashboard

Lastly, user-friendly interface to the workflow would be a valuable addition. Using the R shiny package (https://github.com/rstudio/shiny), we can create a web application built on top of the AQuATx software quickly and easily. Shiny has a simple interface and

large package ecosystem to deploy web applications for data using reactive programming concepts in R. A web application will also provide a way for users to install and run code on servers or personal laptops. The dashboard lives in a separate repository on GitHub (https://github.com/MontgomeryLab/aquatx-dashboard). A basic dashboard with the overall dashboard design was created to guide development. I believe that the best way to develop this dashboard is to first develop the user-interface (UI) aspect fully. Make sure that it makes sense from the user perspective each button and menu that is provided. A developer should first make sure that these are well-designed and allows them to think through how the backend of the app should work. At each new UI function the developer should keep track of what that UI element needs to be connected to and how the code should "react" to changes in the state of that object to carefully plan the backend. Next, each user input should then be connected to the appropriate function in the R-based backend. Each function should be tested one at a time as new functionality is added and use shinytest (https://github.com/rstudio/shinytest) to automate tests as necessary. Reactlog (https://github.com/rstudio/reactlog) should be used to assess reactivity is functioning as expected. The initial interface should simply provide a wrapper around the workflow and a method for configuration. The configuration can be implemented as a script in R through the dashboard and either downloaded or piped to system calls to aquatx. Improvements to the interface should include directly incorporating the post-counting steps into R (statistics & visualizations) instead of performing system calls for these steps through R. This will also allow for easier incorporation of interactive and editable

plots through shiny. Once incorporated, I believe the overall software will be very comprehensive and easy-to-use.

5. A maintenance plan

Bioinformatics tools are often developed by single graduate students or postdocs with no maintenance plan once the developer has moved on. Software maintenance is time-consuming and often unpaid in academic research, which leads to less interest in upkeep and addressing issues. Some tools remain popular due to the fact that they are actively maintained and responsive to the open-source community. Some developers leave tools on their personal GitHub to maintain a sense of responsibility for its future, but as the repository belongs to the Montgomery Lab organization and not my personal page, I have no incentive to maintain the software once I make a final "release" and have moved on to my next step. That isn't to say I would never contribute again, but it is less likely than working on my own separate tools and repositories. Thus, a maintenance plan is essential to the longevity of the software. On the README page of the main repositories, the authors & contributors of the tools are listed and this can grow as the tool grows. A separate section titled "Maintainers" should be added and list the current and former maintainers alongside contributors to make sure all contributing developers are given proper credit. For an initial period (Summer 2019) I can remain a maintainer while finishing the development of v0.5 - 1.0, but a new maintainer should be identified promptly. If a new maintainer cannot be identified at any point in the future, the README should be updated to reflect that the tool is no longer maintained. This

allows users to still use “at-their-own-risk”, but makes clear that no one is available to respond to issues that arise or new feature requests. This reduces frustrations the users might have with using the tool and unable to get help. The responsibilities of the maintainer should be as follows: 1) Keep track of all reported issues and discussions on the repository; 2) Prioritize and fix issues as needed, then close issues that are fixed; 3) Review all pull requests to the repository for code accuracy, style, tests, documentation, installation issues, and what it adds to the codebase; 4) Merge pull requests that pass review; 5) Be aware of the professional developer’s model of branching/merging/rebasing/releasing so that no breaking changes are pushed to the master branch; 6) Create project boards and timelines for releases that detail all new issues to fix for a specific release, new features to be added, who is doing what, outline tests, etc; 7) Update documentation appropriately and as needed; 8) Understand the codebase and tests on a deeper level so that maintenance is easier and responding to issues is simple; 9) Create and incorporate new tests as updates are made to the code. More than one maintainer can be identified to spread the work and can be used as a learning opportunity for students to maintain software on GitHub. As major changes and releases are made, informing the community of the new features and improvements could be published.

3.2.6 Code Availability

The full package for small RNA AQuATx is available freely on GitHub (https://github.com/MontgomeryLab/aquatx-srna). The software is open-source and

licensed under GPLv3 (due to redistribution of HTSeq, bowtie, which are GPLv3). We welcome and encourage contributions to improve the software over time alongside our own.

3.2.7 Analysis of germline small RNAs in *C. elegans*

In order to test the tool, we ran the pipeline on a dataset of dissected germline tissue from *C. elegans*. Germlines were hand-dissected from ~500 young adult worms per replicate. Each replicate occurred on a single day, with all samples being collected on separate days. Strains were alternated by rotation ~30 at a time and multiple sets of worms were staged to last through each day while remaining within a short window of timed development to reduce differences in age and batch effects. Small RNAs were isolated and sequencing libraries prepared as in Chapter 2.2 (Methods). Default settings for the pipeline, custom GFF3 files for feature assignment and masking, and the *C. elegans* WS230 genome for alignment were used. Sample data outputs for a subset of the data are in Tables 3.2 – 3.4. The data will be explored more thoroughly and used as a guide for working out more nuanced aspects and determining potential user-needs of the pipeline during development of v1.0. Sample plots for a subset of the run (Figure 3.2) were created for the baseline plots of v0.1, even though the plotter is not in the end-to-end analysis currently (See Appendix II for more details). Additional plots are being folded into the plotter code and finalized for v0.5. The matplotlib stylesheet or aesthetics of individual plots may also be updated, but the plots of Figure 3.2 represent the overall style being used.

Table 3.2 Run statistics for analysis of germline data. A file created by the feature count and merge steps after alignment to describe the run by alignments and features.

Alignment Statistics	**wild_type_replicate_1**	**ram2_replicate_1**	**ne219_replicate_1**	**wild_type_replicate_2**	**ram2_replicate_2**	**ne219_replicate_2**
_unique_sequences_aligned	844503	838324	860955	902756	887122	857535
_aligned_reads	6426321	6270901	6285110	6461905	6730497	6190121
_aligned_reads_unique_mapping	5554530	5413282	5451279	5565733	5774408	5370158
_aligned_reads_multi_mapping	871791	857619	833831	896172	956089	819963
Feature Statistics	**wild_type_replicate_1**	**ram2_replicate_1**	**ne219_replicate_1**	**wild_type_replicate_2**	**ram2_replicate_2**	**ne219_replicate_2**
_alignments_unique_features	819866	812904	836238	875411	861214	832718
_reads_unique_features	819866	812904	836238	875411	861214	832718
_ambiguous_alignments_classes	21303	21885	21322	23861	22362	21530
_ambiguous_reads_classes	210412	203408	199951	215556	212332	195634
_ambiguous_alignments_features	605	591	560	642	661	553
_ambiguous_reads_features	4161	4203	4218	4307	4423	3580

Table 3.3 Differential Gene Expression Table. The first ten entries of the output comparison of wild type to *alg-5(ram2)* germlines, fold changes in wild type. The final table contains all features compared by DESeq2 and is sorted by adjusted p-value.

	baseMean	**log2FoldChange**	**lfcSE**	**stat**	**pvalue**	**padj**
F58E10.1	306.0105332	-4.036027669	0.418495569	-9.644134768	5.20E-22	7.15E-19
cel-miR-250-3p	4153.345352	2.689126309	0.281450213	9.554536421	1.24E-21	8.52E-19
F17A9.2	171.2030294	3.774220256	0.509405126	7.409073963	1.27E-13	5.82E-11
21ur-54	39.00856391	8.747450925	1.544373191	5.66407846	1.48E-08	5.07E-06
H12C20.2	53.99435124	-2.37988989	0.52023709	-4.574625558	4.77E-06	0.001310041
21ur-12981	42.63541503	3.213410689	0.722045267	4.450428302	8.57E-06	0.001680929
C47E8.4	238.4613217	-2.106485099	0.471048204	-4.471909841	7.75E-06	0.001680929
Y54E10A.2	37.35180497	-2.695718178	0.621514738	-4.337335888	1.44E-05	0.002475178
Y56A3A.7	33.60763814	-2.678679846	0.637666244	-4.200755288	2.66E-05	0.004058375
cel-miR-61-3p	5969.81362	1.121549356	0.273236789	4.104679164	4.05E-05	0.005558953

Table 3.4 Normalized counts output table. The first five lines of the normalized counts (as determined by DESeq2) for the germline data.

	wild_type_replicate_1	ram2_replicate_1	ne219_replicate_1	wild_type_replicate_2	ram2_replicate_2	ne219_replicate_2
21ur-15479	0	0	0	0	0	0
21ur-13439	48.34989779	23.07207615	73.5284904	31.62334372	51.01731871	39.2640805
21ur-8411	80.2402559	68.29334541	160.6733679	54.42156827	61.22078245	78.528161
21ur-14112	0	0	0	0	0	0
21ur-15102	0	0	0	0	0	0

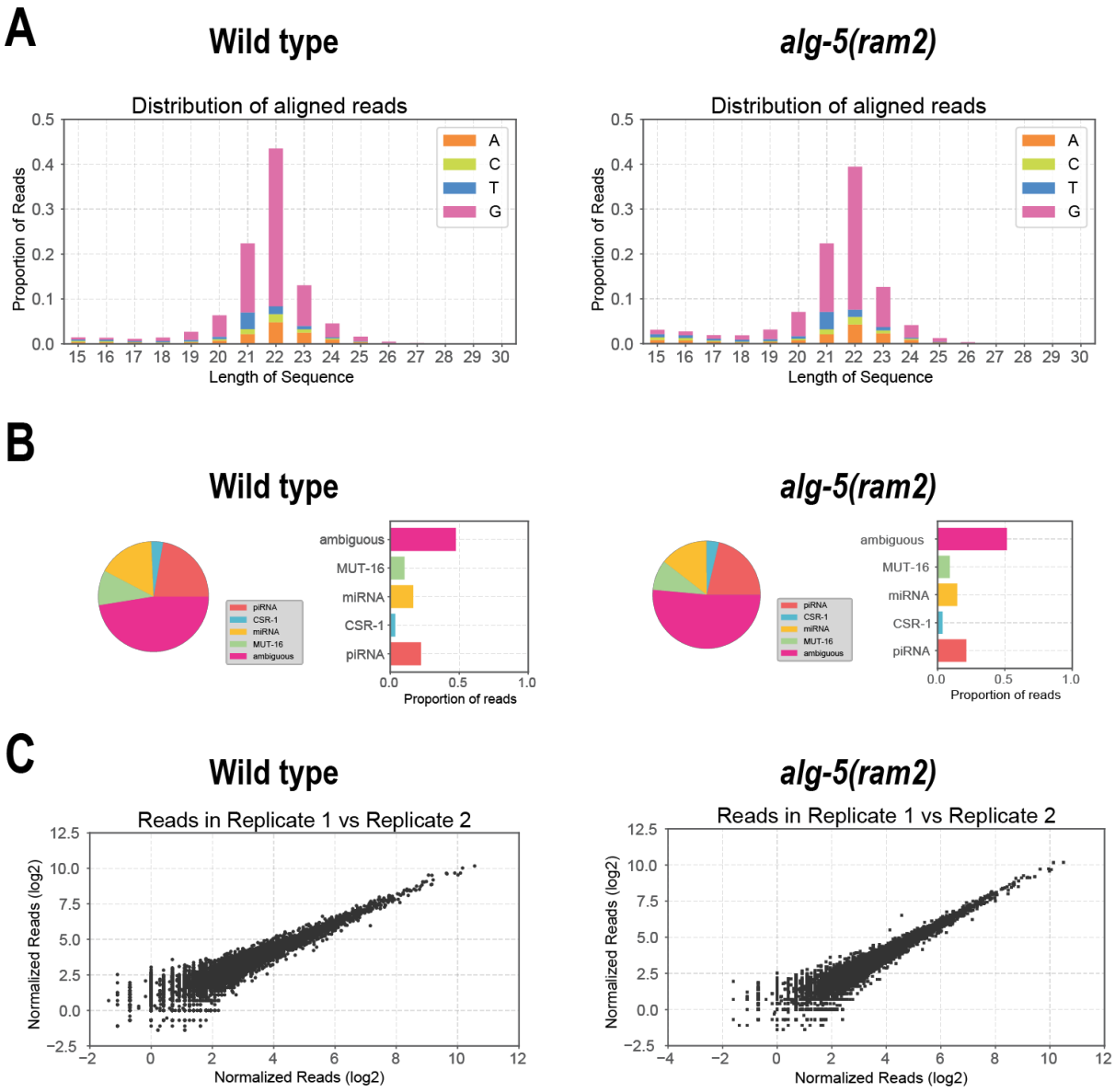


Figure 3.2 Sample plot outputs. A few sample plot outputs created by the initial plotter script using AQuATx v0.1 for a wild type and *alg-5(ram2)* sample. The generate_plots.py script was used to create these plots as it is not wrapped into the end-to-end pipeline yet, see Appendix II for more information on usage. A) Size distribution plots with 5' nucleotide information. Tools in Table 3.1 show only size as far as we could find. B) Pie and bar charts displaying reads aligning to various classes. Ambiguous is defined as aligning to more than one distinct class. C) Scatter plots (log2 normalized) of read counts in 2 replicates to display consistency in read alignments. These can also be produced between different biological samples or averaged reads.

3.3 DISCUSSION

We provide a simple, user-friendly tool for analyzing small RNA sequencing data. While other tools and workflows exist, there are some limitations to these tools that we addressed in our own workflow. Also, very few are maintained after publication or are actively developed. We chose to use CWL to maximize interoperability of the tool, such that the user can choose the implementation to run which is most appropriate for their system, rather than needing to install a very specific workflow engine to run it. We have created a simple, user-friendly interface that allows users unfamiliar with programming to run their analysis from start-to-finish through modification of sample sheets and configuration files and just a couple commands on the command line interface. We also implement some software-engineering guided best practices, such as open-source development on GitHub, thorough documentation, contribution guidelines, and user-guides, test-driven development and continuous integration. We will also integrate a small RNA specific statistical package for analysis of differential gene expression and improved options for model and non-model organisms. We provide modularity and configurability through generation of partial workflows or running of individual steps and allowing a choice of aligner and differential expression tool.

We recognize that there are many more features and improvements to be made in future versions of the software. We welcome suggestions, bug reports, and improvements through our GitHub repositories. The next major improvement will include integrating a simpler user-interface and interactive plotting tools to make exploratory

data analysis simpler and more engaging for users. Folding in more tools and allowing generation of more types of workflows with Nextflow [12] or SnakeMake [13] will also be implemented shortly.

We also present the analysis of small RNA and mRNA populations of dissected germline tissue in *C. elegans* as an example dataset. While miRNAs often dominate small RNA populations in many organisms, tissue specific expression and other organisms may have varying degrees of different classes of small RNAs. In *C. elegans* germline tissue we detect very few miRNAs overall, but many other classes of small RNA are present. By analyzing all the data together, we can more accurately describe and normalize the miRNA population of the germline.

REFERENCES

1. Valladares, O., et al., *SPAR: small RNA-seq portal for analysis of sequencing experiments.* Nucleic Acids Research, 2018. **46**(W1): p. W36-W42.
2. Paicu, C., et al., *The UEA sRNA Workbench (version 4.4): a comprehensive suite of tools for analyzing miRNAs and sRNAs.* Bioinformatics, 2018. **34**(19): p. 3382-3384.
3. Wu, X., et al., *sRNAnalyzer-a flexible and customizable small RNA sequencing data analysis pipeline.* Nucleic acids research, 2017. **45**(21): p. 12140-12151.
4. Fahlgren, N., et al., *Computational and analytical framework for small RNA profiling by high-throughput sequencing.* RNA, 2009. **15**(5): p. 992-1002.
5. Rahman, R.-U., et al., *Oasis 2: improved online analysis of small RNA-seq data.* BMC Bioinformatics, 2018. **19**(1): p. 54.
6. Quek, C., et al., *iSRAP - a one-touch research tool for rapid profiling of small RNA-seq data.* Journal of extracellular vesicles, 2015. **4**: p. 29454-29454.
7. Pogorelcnik, R., et al., *sRNAPipe: a Galaxy-based pipeline for bioinformatic in-depth exploration of small RNAseq data.* Mobile DNA, 2018. **9**(1): p. 25.
8. Rinaldi, A., et al., *iSmaRT: a toolkit for a comprehensive analysis of small RNA-Seq data.* Bioinformatics, 2016. **33**(6): p. 938-940.
9. Wan, C., et al., *CPSS 2.0: a computational platform update for the analysis of small RNA sequencing data.* Bioinformatics (Oxford, England), 2017. **33**(20): p. 3289-3291.
10. Karunanithi, S., M. Simon, and M.H. Schulz, *Automated analysis of small RNA datasets with RAPID.* bioRxiv, 2019: p. 303750.
11. Peter Amstutz, M.R.C., Nebojša Tijanić (editors), Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, Matt Scales, Stian Soiland-Reyes, Luka Stojanovic, *Common Workflow Language, v1.0. Specificatio*, C.W.L.w. group, Editor. 2016.
12. Di Tommaso, P., et al., *Nextflow enables reproducible computational workflows.* Nature Biotechnology, 2017. **35**: p. 316.
13. Köster, J. and S. Rahmann, *Snakemake—a scalable bioinformatics workflow engine.* Bioinformatics, 2012. **28**(19): p. 2520-2522.
14. Robinson, D., *The Incredible Growth of Python*. 2017, Stack Overflow: Stack Overflow.
15. Overflow, S. *Developer Survey Results*. 2018 [cited 2019; Available from: https://insights.stackoverflow.com/survey/2018/.
16. Spjuth, O., et al., *Experiences with workflows for automating data-intensive bioinformatics.* Biology Direct, 2015. **10**(1): p. 43.
17. Schulz, W.L., et al., *Use of application containers and workflows for genomic data analysis.* Journal of pathology informatics, 2016. **7**: p. 53-53.
18. Leipzig, J., *A review of bioinformatic pipeline frameworks.* Briefings in bioinformatics, 2017. **18**(3): p. 530-536.
19. Voss K, G.J.a.V.d.A.G., *Full-stack genomics pipelining with GATK4 + WDL + Cromwell [version 1; not peer reviewed]*, in *18th Annual Bioinformatics Open Source Conference*. 2017, F1000Research.
20. Chen, S., et al., *fastp: an ultra-fast all-in-one FASTQ preprocessor.* Bioinformatics (Oxford, England), 2018. **34**(17): p. i884-i890.
21. Langmead, B., et al., *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biology, 2009. **10**(3): p. R25.

22. Charles, R., et al., *Scaling read aligners to hundreds of threads on general-purpose processors.* Bioinformatics, 2018. **35**(3): p. 421-432.
23. Dobin, A., et al., *STAR: ultrafast universal RNA-seq aligner.* Bioinformatics (Oxford, England), 2013. **29**(1): p. 15-21.
24. Anders, S., P.T. Pyl, and W. Huber, *HTSeq--a Python framework to work with high-throughput sequencing data.* Bioinformatics, 2015. **31**(2): p. 166-9.
25. Love, M.I., W. Huber, and S. Anders, *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.* Genome Biol, 2014. **15**(12): p. 550.
26. Kim, D., B. Langmead, and S.L. Salzberg, *HISAT: a fast spliced aligner with low memory requirements.* Nature Methods, 2015. **12**: p. 357.

4. DISCUSSION

Taken together, the work in this dissertation furthers our understanding of small RNA pathways and the roles of Argonautes in *C. elegans*. Additional analysis of small RNA pathways was performed in collaboration with multiple labs and formed the justification and basis for creating a new automated small RNA data analysis pipeline. We hope that this work helps accelerates discovery in small RNA research across fields. This chapter summarizes the main results and provides a discussion of the potential next steps in this body of work.

4.1 SUMMARY

We performed a comprehensive characterization of three miRNA-associated Argonautes of *C. elegans,* ALG-1, ALG-2, and ALG-5. We characterized T23D8.7/ALG-5 for the first time, learning that it is a germ cell specific Argonaute that interacts with a subset of miRNAs. The small repertoire of miRNAs is likely due to the tissue-specific nature of expression. In normal conditions, *alg-5* is required for proper timing of the switch from spermatogenesis to oogenesis, which leads to a reduction in progeny produced. The work also provides a comprehensive set of data related to miRNA pathways of *C. elegans* that will facilitate future studies.

We created a simple set of tools and workflow for the automated analysis of small RNA sequencing data called AQuATx. It performs a full, end-to-end analysis aimed at users that are unfamiliar with bioinformatics tools, the command line, and/or programming.

This tool provides useful, basic functionality for exploration of small RNA sequencing data for bench scientists that are unable to dedicate the time to learn best practices for small RNA sequencing. The tool also provides modularity in terms of the scripts provided and can be run one step at a time. It is easily extended for additional functionality as a result. The tool development follows several software-engineering best practices that are often lacking in bioinformatics and this allows for simple future maintainability.

4.2 FUTURE DIRECTIONS

The work presented in this dissertation provide a foundation upon which future studies in our lab and others will build. First, there are many questions about the miRNA-associated Argonautes left to be answered experimentally. Second, we provide a reproducible data analysis pipeline, reducing the time and difficulty involved in small RNA research, but that could benefit from improvements. As for myself, I am moving more into computational work and am less likely to work on any of this in particular, but would be happy if someone reading this picked up one of these possible projects. If anything, I may continue to develop and improve the relevant computational tools.

4.2.1 The role of *alg-5* and miRNAs in the germline

There is growing interest in studying the function of miRNAs in the germline, where historically the focus has been on siRNAs and piRNAs. The *mir-35* family has classically been the poster child of the germline miRNAs because they are highly abundant during

oogenesis and embryogenesis [1-3]. Few other miRNAs have been described as germline-enriched [4]. Interestingly, no evidence has emerged to suggest that ALG-1 or ALG-2 are expressed in the germ cells themselves, but some studies suggest they are instead restricted to the somatic gonad, including ALG-1 acting in the distal tip cells [5-7]. ALG-5 might be the only miRNA-associated Argonaute specifically acting in proliferating germ cells. A recent study [8] showed that the *mir-35* family regulates apoptosis through the MAPK pathway in the germline. They also suggest that a P granule protein, CGH-1, is involved in inhibition of *ndk-1*, which is required for proper germline development. Specifically inhibiting CGH-1 leads to excessive germ cell death [8]. While ALG-5 does seem to interact with the *mir-35* family, it seems to preferentially bind other miRNAs. Although, one of the early hypotheses I had about the function of *alg-5*, involved the regulation of proper germ cell death because some miRNAs it bound and mRNAs that were differentially expressed in early data suggested it might target the MAPK apoptotic pathway that included many of the genes mentioned in the study. When I looked into performing apoptosis assays, the phenotypes were likely to be very subtle and normal conditions might make it difficult to quantify this role. Inducing genotoxic stress through ionizing radiation might "activate" ALG-5 and produce a measurable difference in its absence.

In addition to genotoxic stress response, I also wondered about ALG-5 playing a larger role as a "responsive" Argonaute and that is why the effects it has under normal conditions are more modest. Perhaps under more stressed conditions, more interesting

phenotypes to study would emerge and would suggest that under normal conditions ALG-5 is dispensable, but under certain conditions it might be required for differing responses to protect the integrity of germ cells. We tested a few of these in the lab, such as osmotic stress, but the results were inconclusive. More investment in this might be worthwhile moving forward.

Lastly, the actual targeting of ALG-5 might be an interesting area of study. ALG-5 does not have slicer activity like ALG-1 and ALG-2, although the requirement for this catalytic domain is questionable in *C. elegans* miRNA pathway. It would be interesting to see if anything changes in mutants where ALG-5 is given the catalytic DDH motif, in terms of the miRNAs it binds and what the effects are in the germline. Another recent, interesting study provides evidence that miRNAs of germ cells actually protect target mRNAs from degradation, differing from their normal function in the somatic tissue [9]. This provides an interesting avenue for exploring ALG-5 function, as this study also showed that target mRNAs are sequestered and localized next to P granules, dependent on GLH-1. And while this study did suggest that ALG-1 and ALG-2 were part of the germline miRISCs and were unable to detect ALG-5, they pulled down miRNAs enriched or depleted in *glp-4(bn2)* studies to determine what was “germ cell enriched” vs somatic [9]. This is not an ideal way to study the different tissues as enrichment might include differences in development of somatic gonad depending on the stage being compared. Thus, I believe that it would be interesting to study this phenomenon of mRNA localization by miRNA targeting through ALG-5, whether it interacts with GLH-1, and if

the mRNAs it targets are stabilized. This might also explain why few mRNAs are differentially expressed in *alg-5* mutant animals, if the mRNAs are normally being stabilized, but in a dynamic way as the germ cell progress and mature.

4.2.2 Why are *alg-2* mutants long-lived?

In the lab, we consistently observed that *alg-2(ok304)* mutants were long-lived at elevated temperatures. The Pasquinelli lab observed and published the same findings [7], but failed to identify a specific mechanism for the phenotype. In my time counting the brood sizes of the miRNA mutants, I noticed that *alg-2* mutants were particularly frustrating to count because they would often disperse all over the plate, rather than sticking to the lawn as much as the other mutants or wild type. While we never quantified this observation, I believe that it is still a plausible explanation for their long lives. If the assay for measuring this lawn-leaving phenotype were performed and it was confirmed as significantly different from wild type, this would have made a nice second project for me had I been able to pursue it.

In particular, I believe that the lawn-leaving phenotype is a direct consequence of the *alg-2* mutation and the longevity phenotype is actually an indirect consequence. This behavioral phenotype is likely through a loss of miRNA regulation in neurons where ALG-2 is highly expressed, and ALG-1 is not [6, 7]. The longevity increases because the animals are moving around more instead of staying in the food, thus they are eating less. It has been well-documented that caloric restriction and insulin signaling play

important roles in lifespan extension [10]. Thus, I believe that any insulin changes in the animals are indirectly caused by caloric-restriction.

The actual targets of ALG-2 could play a role in behavior and this is a much more interesting pathway for a miRNA-associated Argonaute to regulate because this indicates that gene expression is more responsive to environmental cues through small RNA regulation. In our mRNA-seq data for *alg-2(ok304)* mutants, there was an upregulation of neuropeptide-related genes, indicating that there are some potentially interesting genes ALG-2 could target. In particular, miR-71 has been shown to have roles in increasing lifespan through regulation within neurons [11]. Interestingly, a very recent study showed that miR-71 directly targets TIR-1 in olfactory neurons and affects odor perception, controlling chemotactic behavior in the animals [12]. We also showed that ALG-2 interacts with miR-71-5p (Chapter 2, Supplementary Table S2.4) and this could be, at least partly, why *alg-2* seems to regulate lifespan. And while this mechanism and phenotype seems to be explained, this could still be an attractive project that directly shows this and how Argonautes might respond to environmental cues.

4.2.3 RNAi and the study of small RNA pathways

High-throughput RNAi studies of phenotypes are not always consistent with loss-of-function mutation studies and it was recently shown that this is can be a greater concern with RNAi against small RNA pathway components [13]. This is consistent with a couple

of our own observations. First, *alg-1* and *alg-2* were identified as factors that shortened lifespan in original RNAi studies [14, 15], yet with a loss-of-function mutation in the gene, we observe that it is in fact long-lived [7]. However, this is more likely due to RNAi against *alg-2* targeting *alg-1* as well. Second, *alg-5* was identified in an RNAi screen for factors involved in Cry5b toxin sensitivity [16] as well as an unpublished RNAi screen for sensitivity to the antibiotic geneticin. In our own studies and other labs, *alg-5* loss-of-function mutants do not display this hypersensitivity to drug or toxin treatment. Now, there are a couple potential areas to explore as to why this is. One could look into specifically, the reason *alg-5* is only hypersensitive as a result of RNAi or into how RNAi against small RNA factors cause unintended consequences. One potential way to study this is to consider the fact that all systems are resource-limited. Multiple small RNA pathways utilize the same components for biogenesis and action, thus provide a limiting factor for each pathway. With so many different small RNA pathways co-existing in the germline, perhaps *alg-5* is normally competing with exogenous RNAi factors and this plays a role in why RNAi against *alg-5* leads to hypersensitivity to toxins, but a genomic deletion does not. In wild type worms, *alg-5* might act to protect against invading toxins alongside some other germline small RNA factor. When *alg-5* is deleted in a mutant, this other factor is able to pick up the slack in responding to toxins. However, when *alg-5* is depleted by RNAi, this other factor is resource-limited because exogenous RNAi "wins" the resources and becomes susceptible to toxin infection as a result. Studying this might require more complex understanding of stochasticity and dynamics of

biological systems to fully tease out. This, however, could make for an interesting project on how small RNAs of the germline are poised to react to their environment.

4.2.4 Improvements to AQuATx

The software presented at the time of this dissertation writing is a tool with fairly basic functionality. Implementation of software-engineering best practices was a large focus during development in order to allow future students and outside contributors to improve the software over time without much difficulty. Thus, there are several features and improvements that immediately come to mind that can be addressed within the time to final publication, not discussed here, as well as beyond that for any interested incoming students.

An obvious addition is to implement small RNA specific, third-party tools for: 1) small RNA discovery, 2) predicting mRNA targets, 3) creating sRNA-mRNA interaction networks, 4) RNA modification analysis – isoforms, tailing, etc. These tools are often desired in a more tailored analysis, but can be difficult to install or implement for many without the appropriate experience. They would be simple to fold into the existing pipeline as an extra, optional set of steps. For someone more interested in computational biology, this could be a good starter or rotation project to simply collect all the tools that exist in this area then install, use, and compare them among multiple standard datasets. Figuring out the best tools and the optimal parameters is an important step in workflow development. Adding in custom tools that fill a missing need

after this analysis can be another extension – including better, more specific analysis and visualizations. Such benchmarking and tool creation could even be a separate publication outside of folding these tools into the existing pipeline.

One simple, but time-consuming addition that has been in our minds from the beginning is to provide databases and pre-configuration for analyzing small RNAs from model organisms, so users don't have to provide their own small RNA annotations. Creating more options in terms of tools that can be run (such as STAR) in place of others would likely be beneficial to more experienced computational biologists. Creating tools for users that lack reference genomes would likely be a popular addition, but more difficult to implement properly. One option for de novo analysis of such datasets would include implementing clustering algorithms for sequences with associated downstream analysis and comparison to related, known small RNAs from miRBase [17], for example. Another improvement is to intelligently assign multi-mapping reads, which is potentially complex simply due to the nature of the problem – how do we know where a small RNA derives from, with any certainty, given two locations in the genome? If there are no additional clues, such as nearby single-mapping alignments, it seems unlikely that we can assign reads with a high-level of confidence.

Lastly, there are several bottlenecks in the pipeline that could be improved for performance through rewrite in C/C++, the removal of R dependencies, better parallelization, and more. The dashboard is built with R Shiny due to simplicity of

implementation, but this dependency on R could also be removed to make it easier for end-users to install. However, these tasks are more of an undertaking and likely of less interest to a biologist or even a computer scientist since it is not an interesting scientific problem, but a product-oriented software development one.

Final Remarks

There are many areas in the small RNA world to explore, both experimentally and computationally. If you spent much time with me in the lab, you know I had a tendency to go down hypothesis rabbit holes for nearly everything I did, so I hope to someday see some of these confirmed, ideally without having to do any of the work myself. Best of luck to future students in the field that might be reading this.

REFERENCES

1. Alvarez-Saavedra, E. and H.R. Horvitz, *Many families of C. elegans microRNAs are not essential for development or viability.* Curr Biol, 2010. **20**(4): p. 367-73.
2. Lau, N.C., et al., *An Abundant Class of Tiny RNAs with Probable Regulatory Roles in <em>Caenorhabditis elegans</em&gt.* Science, 2001. **294**(5543): p. 858.
3. Wu, E., et al., *Pervasive and cooperative deadenylation of 3'UTRs by embryonic microRNA families.* Mol Cell, 2010. **40**(4): p. 558-70.
4. McEwen, T.J., et al., *Small RNA in situ hybridization in Caenorhabditis elegans, combined with RNA-seq, identifies germline-enriched microRNAs.* Developmental biology, 2016. **418**(2): p. 248-257.
5. Bukhari, S.I., et al., *The microRNA pathway controls germ cell proliferation and differentiation in C. elegans.* Cell Res, 2012. **22**(6): p. 1034-45.
6. Vasquez-Rifo, A., et al., *Developmental characterization of the microRNA-specific C. elegans Argonautes alg-1 and alg-2.* PLoS One, 2012. **7**(3): p. e33750.
7. Aalto, A.P., et al., *Opposing roles of microRNA Argonautes during Caenorhabditis elegans aging.* PLoS genetics, 2018. **14**(6): p. e1007379-e1007379.
8. Tran, A.T., et al., *MiR-35 buffers apoptosis thresholds in the C. elegans germline by antagonizing both MAPK and core apoptosis pathways.* Cell Death & Differentiation, 2019.
9. Dallaire, A., P.-M. Frédérick, and M.J. Simard, *Somatic and Germline MicroRNAs Form Distinct Silencing Complexes to Regulate Their Target mRNAs Differently.* Developmental Cell, 2018. **47**(2): p. 239-247.e4.
10. Uno, M. and E. Nishida, *Lifespan-regulating genes in C. elegans.* Npj Aging And Mechanisms Of Disease, 2016. **2**: p. 16010.
11. Boulias, K. and H.R. Horvitz, *The C. elegans microRNA mir-71 acts in neurons to promote germline-mediated longevity through regulation of DAF-16/FOXO.* Cell metabolism, 2012. **15**(4): p. 439-450.
12. Finger, F., et al., *Olfaction regulates organismal proteostasis and longevity via microRNA-dependent signalling.* Nature Metabolism, 2019. **1**(3): p. 350-359.
13. Pinca, Ana Paula F., et al., *RNA interference may result in unexpected phenotypes in Caenorhabditis elegans.* 2019.
14. Kato, M., et al., *Age-associated changes in expression of small, noncoding RNAs, including microRNAs, in C. elegans.* RNA, 2011. **17**(10): p. 1804-20.
15. Samuelson, A.V., C.E. Carr, and G. Ruvkun, *Gene activities that mediate increased life span of C. elegans insulin-like signaling mutants.* Genes & development, 2007. **21**(22): p. 2976-2994.
16. Kao, C.Y., et al., *Global functional analyses of cellular responses to pore-forming toxins.* PLoS Pathog, 2011. **7**(3): p. e1001314.
17. Kozomara, A. and S. Griffiths-Jones, *miRBase: annotating high confidence microRNAs using deep sequencing data.* Nucleic acids research, 2014. **42**(Database issue): p. D68-D73.

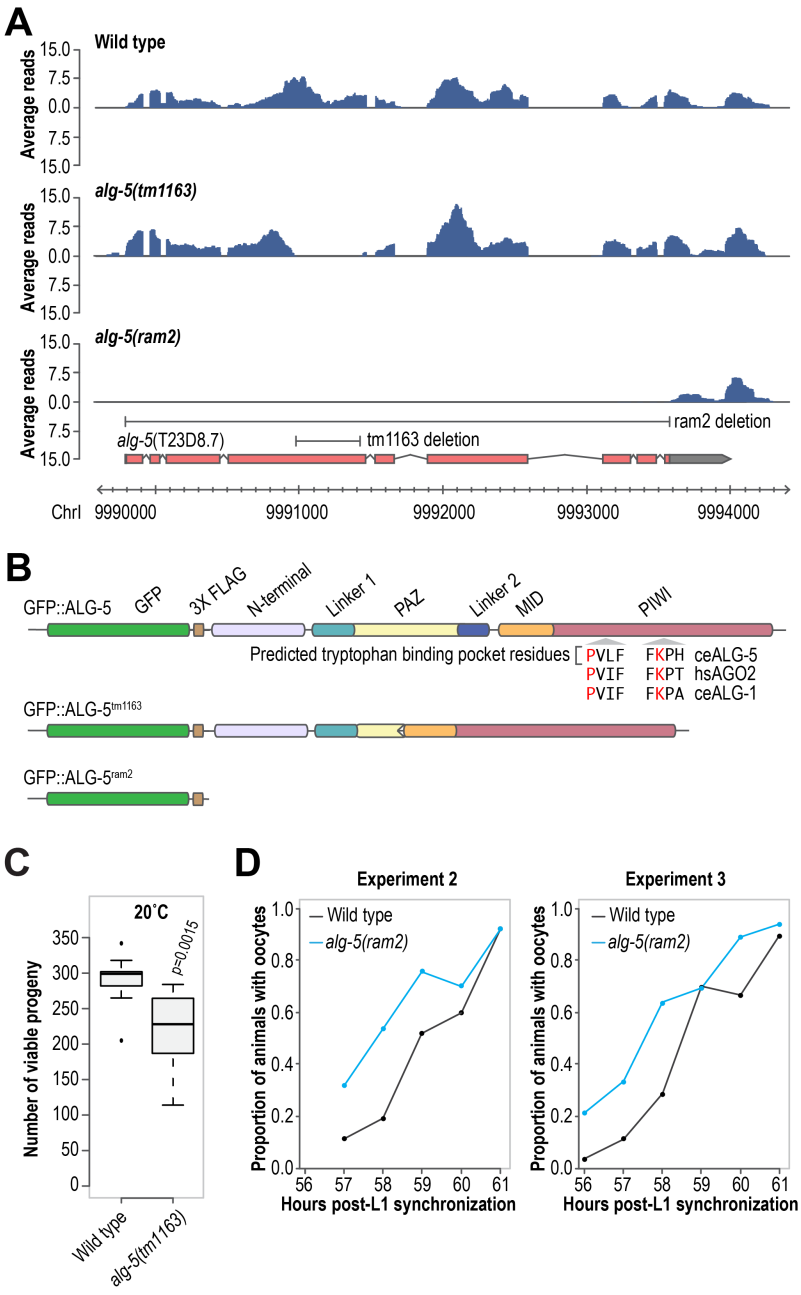
APPENDIX I.

S2.1 Supplementary Material for Chapter 2

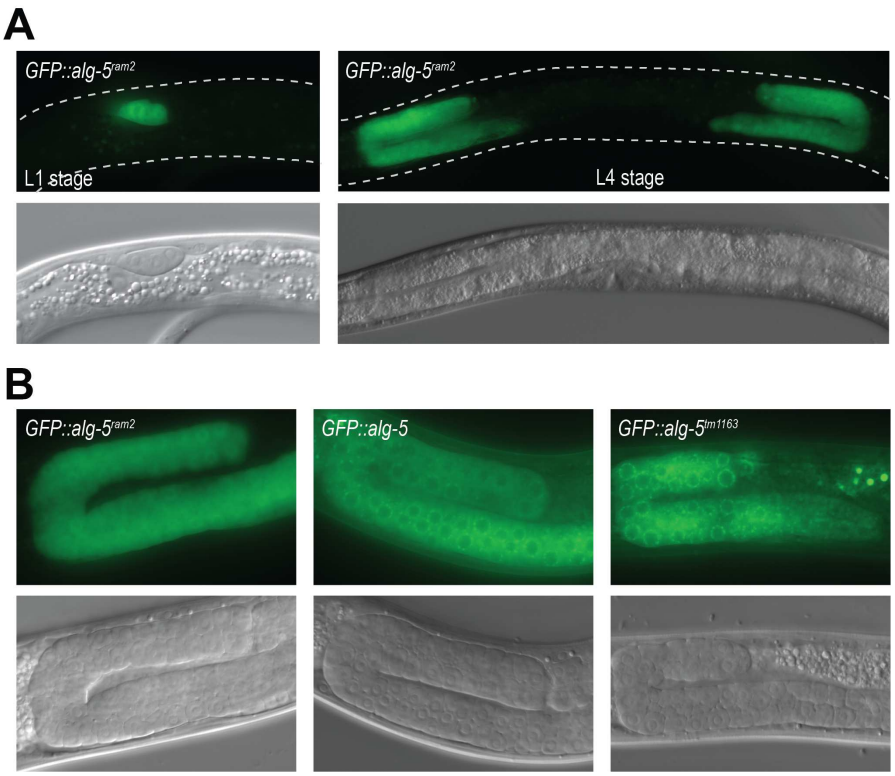
Sequencing data was published in GEO under accession GSE98935. Supplementary tables are attached online alongside this dissertation as an Excel spreadsheet. Appendix I contains all supplementary figures referenced in Chapter 2.

Supplementary data for Chapter 2 can also be found at NAR online:

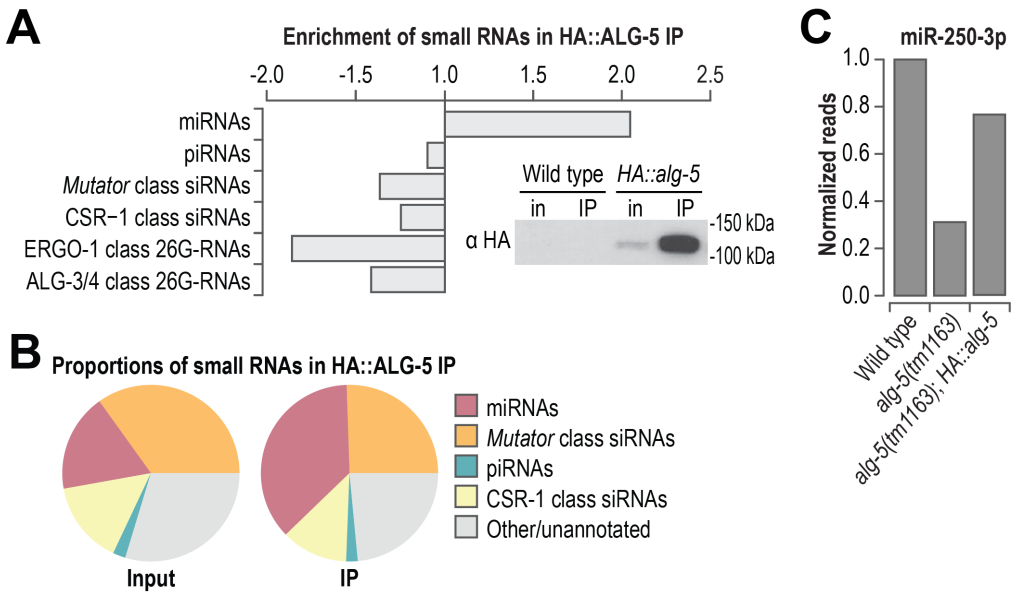
https://academic.oup.com/nar/article/45/15/9093/3883740



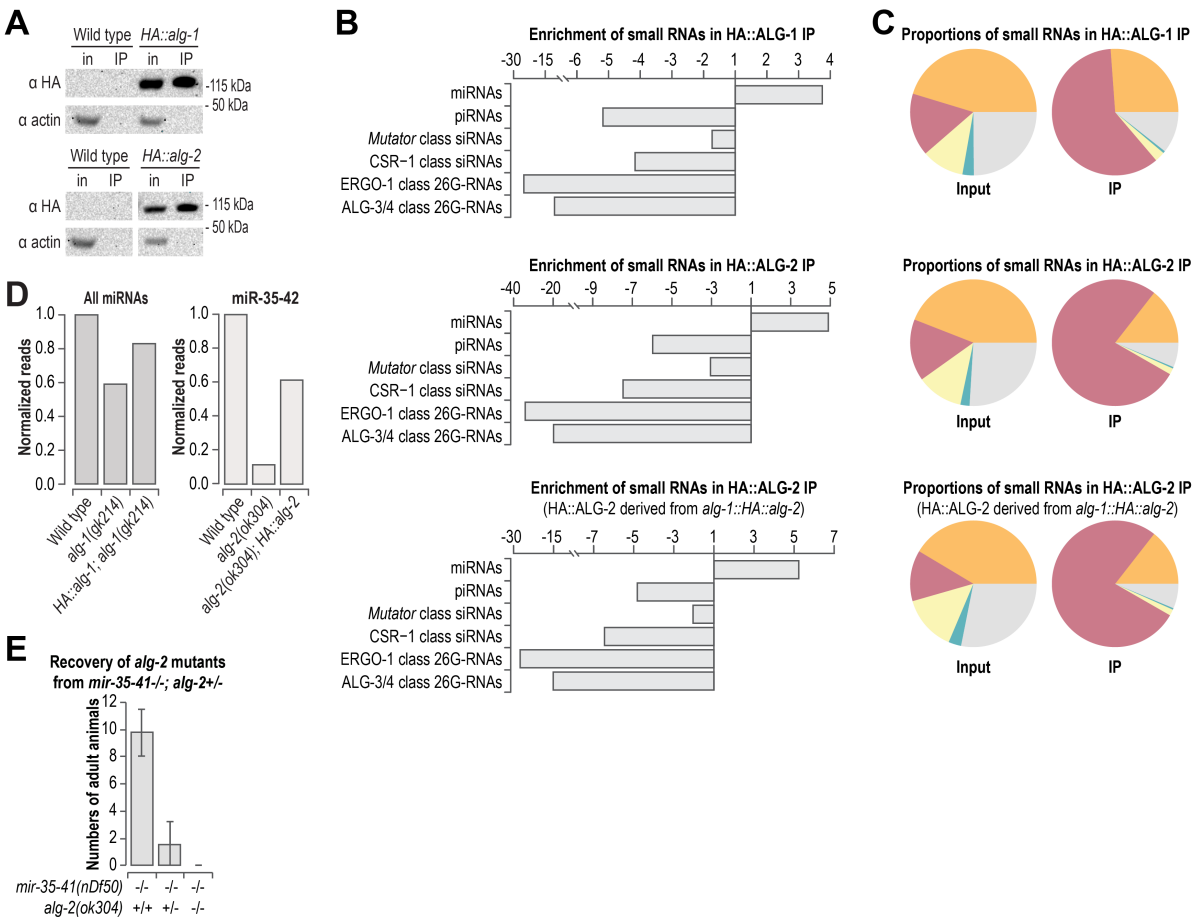
Supplementary Figure S2.1. ALG-5 is required for optimal fertility. (A) Average read distribution across *alg-5* in wild type and *alg-5(tm1163)* and *alg-5(ram2)* mutants. (B) Predicted protein domains of ALG-5 and ALG-5^{tm1163} and ALG-5^{ram2} (deletion allele) fused to GFP::3X-FLAG. (C) Number of viable progeny produced by *alg-5(tm1163)* (n=15) grown at 20°C. (D) Proportions of wild type and *alg-5(ram2)* mutant animals with oocytes formed at 56-61 hours post-L1 synchronization (n=~25-50). Two of three independent experiments are shown. Experiment 1 is in Figure 2.1.



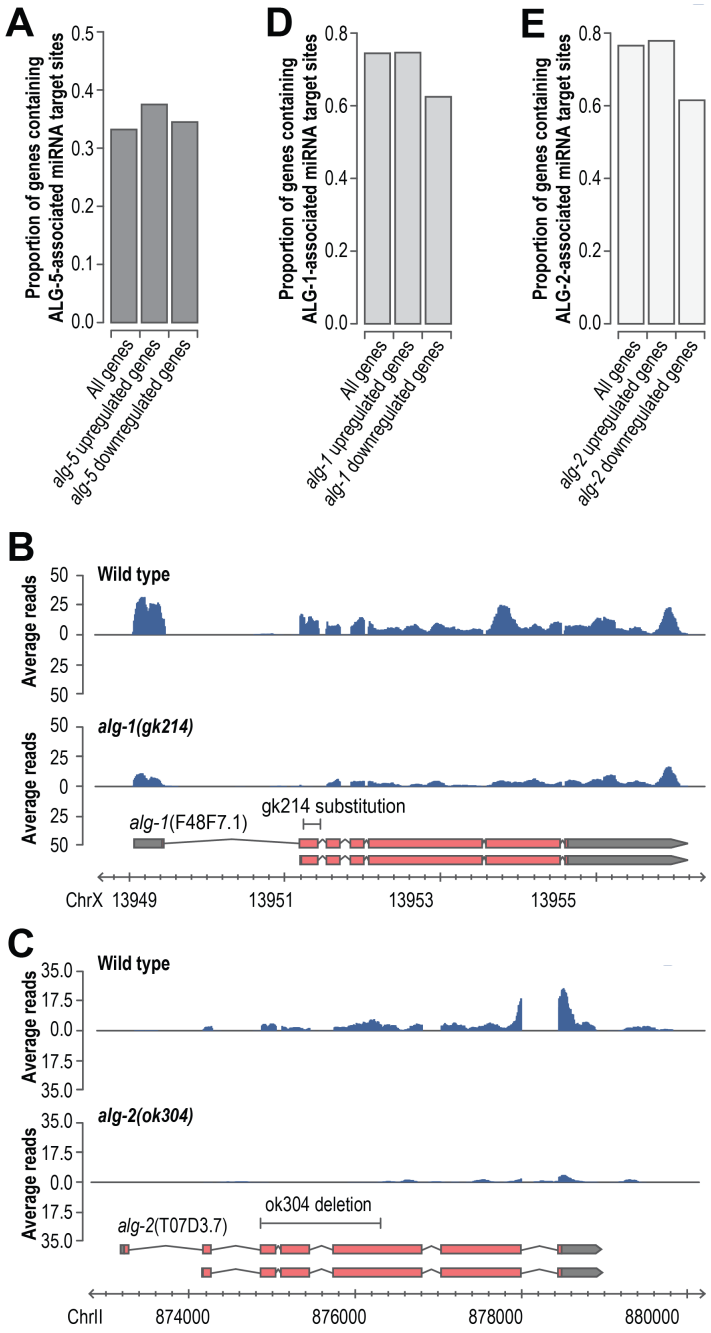
Supplementary Figure S2.2. *alg-5* is expressed in the germline. (A) Representative images of GFP expressed from a transcriptional reporter for *alg-5* activity (*alg-5(ram2)*) in which the endogenous *alg-5* coding sequence was replaced with GFP sequence. (B) The *alg-5(tm1163)* allele produces a stable protein when fused to GFP. Three different *alg-5* alleles are shown: the transcriptional reporter described in A (*GFP::alg-5*^{ram2}); a translational reporter for wild type ALG-5 (*GFP::alg-5*) in which GFP was introduced in frame with *alg-5* coding sequence in wild type animals; and a translational reporter for mutant ALG-5 produced from *alg-5(tm1163)* (*GFP::alg-5*^{tm1163}) in which GFP was introduced in frame with *alg-5* coding sequence in the partial deletion allele *alg-5(tm1163)*. Upper panels are GFP fluorescence and lower panels are DIC images of the germline regions of living animals.



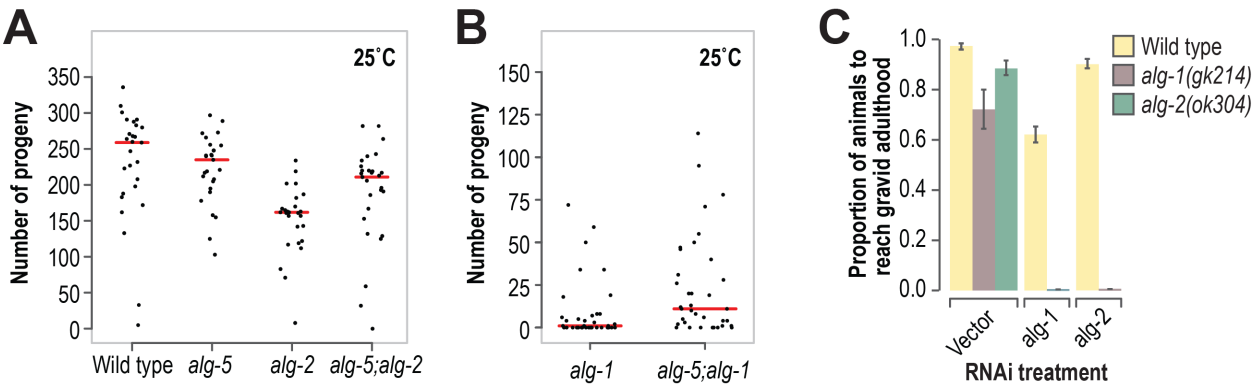
Supplementary Figure S2.3. HA::ALG-5-miRNA interactions. (A) Enrichment of miRNAs, piRNAs, and siRNAs in HA::ALG-5 co-IP relative to input as determined by high-throughput sequencing. The inset blot image is a western blot assay of HA::ALG-5 from cell lysates (input, in) and co-IPs (IP) used for small RNA isolation and sequencing. ~0.2% starting material equivalents for the input fractions and ~5% starting material equivalents for the co-IP fractions were run on the gel for the Western blot. (B) The relative proportions of each class of small RNAs in input and co-IP fractions. (C) miR-250-3p normalized reads (RPM) in *alg-5(tm1163)* and *alg-5(tm1163); HA::alg-5* relative to wild type.



Supplementary Figure S2.4. Small RNA interactors of HA:ALG-1 and HA::ALG-2. (A) A western blot assay of HA::ALG-1 and HA::ALG-2 from cell lysates (input, in) and co-IPs (IP) used for small RNA isolation and sequencing. ~0.3% starting material equivalents for the input fractions and ~5% starting material equivalents for the co-IP fractions were run on the gels for Western blots. (B) Enrichment of miRNAs, piRNAs, and siRNAs in co-IPs of HA::ALG-1, HA::ALG-2, and HA::ALG-2 under the *alg-1* regulatory elements relative to cell lysate input fractions as determined by high-throughput sequencing. (C) The relative proportion of each class of small RNAs in input and co-IP fractions. (D) Normalized reads (RPM) for all annotated miRNAs relative to wild type animals in *alg-1(gk214)* and *HA::alg-1; alg-1(gk214)* (left panel). RPM for all miR-35-42 family miRNAs in *alg-2 (ok304)* and *alg-2(ok304); HA::alg-2* relative to wild type animals (right panel). (E) *alg-2(ok304)* additivity with a mir-35 family mutant. Frequency of the *alg-2(ok304)* mutation in segregating animals. Animals containing the *mir-35-41(nDf50)* mutation were crossed with animals containing the *alg-2(ok304)* mutation. An F1 animal homozygous for *mir-35-41* and heterozygous for *alg-2* was selected and its progeny expanded over several generations and then individual adult animals were genotyped (n=45). Error bars represent standard deviations.



Supplementary Figure S2.5. mRNA-seq of *alg-1* and *alg-2* mutants. (A) The proportion of all genes and genes misregulated in *alg-5(ram2)* mutants that contain 7-mer or 8-mer target sites for GFP::ALG-5-associated miRNAs. (B) Average read distribution across *alg-1* in wild type and *alg-1(gk214)* mutants. (C) Average read distribution across *alg-2* in wild type and *alg-2(ok304)* mutants. (D) The proportion of all genes and genes misregulated in *alg-1(gk214)* mutants that contain 7-mer or 8-mer target sites for HA::ALG-1-associated miRNAs. (E) The proportion of all genes and genes misregulated in *alg-2(ok304)* mutants that contain 7-mer or 8-mer target sites for HA::ALG-2-associated miRNAs.



Supplementary Figure S2.6. Functional overlap of ALG-5, ALG-1, and ALG-2. (A) Numbers of viable progeny produced by wild type (n=27), *alg-5(ram2)* (n=27), *alg-2(ok304)* (n=27), and *alg-5(ram2); alg-2(ok304)* (n=27) at 25˚C. (B) Numbers of viable progeny produced by *alg-1(gk214)* (n=36) *alg-5(ram2); alg-1(gk214)* (n=36) at 25˚C. (C) Proportion of wild type, *alg-1(gk214)*, and *alg-2(ok304)* animals to reach gravid adulthood by 96 hours at 20°C when treated with vector (L4440), *alg-1,* or *alg-*2 RNAi. Bars depict average proportion in two replicates and error bars represent standard deviations.

APPENDIX II

Supplementary Material for Chapter 3

The full user and installation guide for AQuATx v0.1 is written here, as well as a short summary of v0.1, v0.5, and v1.0 major milestones. Example inputs, brief output descriptions (examples in Chapter 3), and short summaries with links to additional scripts that could be used or further developed are described. Appendix II contains all supplementary information for Chapter 3.

S3.1 AQUATX USER GUIDE (v0.1)

This user guide is for the currently published v0.1 of AQuATx online (https://github.com/MontgomeryLab/aquatx-srna). The user guide will be uploaded and available online accessible through the main GitHub repository README or within the wiki (https://github.com/MontgomeryLab/aquatx-srna/wiki). Chapter 3 also describes v0.5 of AQuATx, currently under development. The progress may be checked via the project board (https://github.com/MontgomeryLab/aquatx-srna/projects/1). All issues, feature enhancements, pull requests, and future projects will be available on the GitHub repository.

S3.1.1 MAJOR VERSION SUMMARIES

v0.1 represents a basic analysis pipeline with minimal testing. The data can be processed from raw fastq file inputs to differentially expressed gene tables, raw and normalized counts of features and classes, size and length distribution counts, and run statistics. Plots can be created via the plotter script separately.

v0.5 folds in the updated plotter into the end-to-end workflow. It also allows for additional file format support at various steps, more modular workflow configuration, sanity checks for configuration, better data management and organization, and robust testing for validity and robustness. Bioconda installation recipes will be created for v0.5.

v1.0 does not yet have specific milestones or checklists to completion, but suggested additions are described in Chapter 3 that could become part of v1.0. The main goal of a v1.0 is to release a well-tested, stable version that is feature rich and robust. Well-tested includes testing on multiple platforms, good code coverage of tests, testing with multiple organisms.

S3.1.2 PRE-REQUISITES AND INSTALLATION (v0.1)

The basic AQuATx pipeline for small RNA analysis depends on Python3 (tested for 3.6), numpy, pandas, matplotlib, HTSeq, fastp, bowtie, R, and DESeq2. To install the pipeline it is also easiest to install with conda, otherwise you'll likely need to install each tool separately. Conda will also allow you to set up a virtual environment to reduce version conflicts of dependencies. If you run into an xcrun error, such as an invalid active developer path on MacOS, please make sure you have XCode developer tools installed and try again.

Install conda

To install conda you can either download and follow instructions for:
Anaconda (https://www.anaconda.com/distribution/), which comes packaged with a few dependencies already or miniconda3 (https://docs.conda.io/en/latest/miniconda.html), which doesn't install unnecessary packages. If you already have conda installed and you run into issues with installation of the software, it might be an issue with the conda

install. Please reinstall conda and try again if you get errors related to: pip installation, packages not being found. After installation, please upgrade conda using:

```
# update conda
conda update –n base conda
```

(Optional) Set up a virtual environment

A suggested, optional step is to use conda to set up a virtual environment using:

```
# create a new environment for aquatx
conda create –n aquatx

# activate the environment for next steps
conda activate aquatx
```

If you have an error at this step, try the following command:

```
# activate the environment for next steps
source activate aquatx
```

Once this is complete, when using AQuATx you'll need to activate the environment and deactivate it at the end. This allows the tool to live in an environment separated from other tools and dependencies that might create conflicts. To deactivate the environment after running use:

```
# deactivate aquatx environment
conda deactivate aquatx
```

Install R & DESeq2

Since there are some known conflicts with installing R via conda, I recommend you install it yourself first from CRAN (https://www.r-project.org/) and DESeq2

following instructions in Bioconductor

(https://bioconductor.org/packages/release/bioc/html/DESeq2.html).

Install AQuATx

To install AQuATx & the remaining dependencies use the following code snippet:

```
# Clone the repository into a local directory
git clone https://github.com/MontgomeryLab/aquatx-srna.git
cd aquatx-srna

# Install dependencies using conda
conda install -c bioconda --file requirements.txt

# Run the aquatx setup script
python setup.py install
```

This code should install everything successfully and if it does not, please report

installation issues here: https://github.com/MontgomeryLab/aquatx-srna/issues

You might also want to remove the downloaded directory. First move to the parent

directory then remove the folder. You can also delete the folder from the OS directory

browser.

```
# Remove downloaded files
cd ..
rm –rf aquatx-srna
```

The version you have downloaded and installed will be the *latest developer version*, aka

the software which has not necessarily been released. Thus, if you are following this

guide post-v0.1, you won't be getting v0.1 and the remaining user guide may not relate

to the tool you just installed. This version will always be the one to clone for those that

wish to add new changes to the repository (and that have write permissions – please fork the repository if you do not). If we are in v0.5, please read the updated README on the GitHub repository for the *latest stable version* installation instructions.

S3.1.3 USING THE END-TO-END WORKFLOW

There are several ways to use the pipeline, the simplest of which is to run an *end-to-end analysis* of raw fastq sequencing data files. Each independent step can also be used and run independently of other steps, but this is a more complex interface to the tools in v0.1. First the *end-to-end analysis* workflow will be described, followed by how to use individual steps. To use the software, you need to modify the input file templates provided. The files can be downloaded from the GitHub page or can be copied fresh by running from the command line:

```
aquatx get-template
```

This will copy the 3 files needed for configuring a workflow run with the software. There is the main configuration file, a sample table, and a reference table.

1. Main configuration file

The main configuration file is written in a standard YAML format. The file contains the main options that are required to be modified before use, as well as optional changes to default settings that can be made. The document is separated with commented headers and descriptions of each option. A sample configuration is given in Box S3.1. Not all options are implemented in v0.1. Commented sections are not read by the software and

are denoted at each line beginning with the “#” symbol and are colored blue for this example in Box S3.1. YAML files are often used to store metadata and make a good standard format for configuration files because they use a key-value pair system (like JSON, but formatted differently) that can be read and parsed by Python tools as a dictionary. The keys in this file are highlighted with bolding, followed by a ‘:’. The formatting of the text is used in this appendix to allow for easy reading, but the actual input file is a plain text file. Some settings are optional and commented out, but may be uncommented if the user wishes to set them.

The first section “AQuATx Configuration”. This is the main metadata for the run to keep track of how the workflow was configured, an important aspect of data provenance and reproducibility. If this section is not filled out then the software automatically fills in the information it can gather, such as the run date and time, and uses that to build a run identifier. Once the configurations are modified, you can rename this file to have an appropriate name description for your experiment. If you do not rename the file, a copy will be created with a name using this “run_prefix” value. The next section is mandatory and points AQuATx to the other input files, the sample table, reference table, and reference files (genome and annotation). Beyond this section, defaults are set according to what we believe will optimally process small RNA sequencing data. These can be changed, but do not need to be. Additionally, some options are not to be modified (as noted) and will be filled in automatically by the software to use during the run. The tool will use a new input configuration that contains all the file information so that the run

metadata can be recorded. More options may be added by creating new, unused key-value pairs following this format.

Box S3.1 run_config_template.yml

```
######---------------------------- AQuATx Configuration ----------------------------######
#
# For reproducibility purposes, if you do not rename this file, we will make a copy
# with current run time and date information.
#
# Please rename this file with identifying information if you wish to trace back your run.
# If you would like to add run time information, you may add it here.
#
#
#
# 1. Add a username here to identify the person creating the runs, if desired for record keeping
# 2. Please add a final run directory to store files - otherwise it will be generated based on
#    initial run time and date (with added option username).
# 3. Please add a final run output prefix to label run-specific summary reports - otherwise
#    it will be generated based on initial run time and date (with added optional username).
#
######-------------------------------------------------------------------------------######

user: 'kristen'
run_directory: ' '
run_prefix: 'germline_seq'
run_date: ''
run_time: ''

######################  MAIN INPUT FILES FOR ANALYSIS ######################
#
# Edit this section to contain the sample sheets with file information to use in the
# workflow. If you want to use DEFAULT settings for the workflow, this is all you need to
# edit before running the workflow.
#
# Directions:
# 1. Fill out the sample sheet with files to process + naming scheme. [sample_sheet_template.csv]
# 2. Fill out the reference sheet with reference files and corresponding optional masks
[reference_sheet_template.csv]
# 3. Rename the files and list them here. If you DO NOT rename them, they will be renamed for you.
# 4. Add an output identifier for summary files and databases
#
######-------------------------------------------------------------------------------######

##-- Relative path to sample & reference sheets --##
sample_sheet_file: 'extras/sample_sheet_template.csv'
reference_sheet_file: 'extras/reference_sheet_template.csv'

##-- The prefix for your bowtie index, include relative path --##
```

```
ebwt: 'tests/testdata/references/c_elegans_ws230'

##-- If True: run bowtie-build to index the genome --##
run_idx: False

##-- Number of threads for multi-threaded programs --##
threads: 2

##-- Final output file prefixes for overall run --##
##-- If none given, run_prefix is used (default: date_time_aquatx) --##
output_prefix: ''

######----------TRIMMING AND QUALITY FILTER OPTIONS ----------######
#
# We use the program fastp to perform: adapter trimming (req), quality filtering (on),
# and QC analysis for an output QC report. See https://github.com/OpenGene/fastp for more
# information on the fastp tool. We have limited the options available to those appropriate
# for small RNA sequencing data. If you require an addition option, create an issue on the
# pipeline github: https://github.com/biokcb/smallRNA/issues
#
# We have specified default parameters for small RNA data based on our own "best practices".
# You may change the parameters here.
#
######--------------------------------------------------------------------------------######

##-- Adapter sequence to trim --##
adapter_sequence: 'auto_detect'

##-- Minumum & maximum accepted lengths after trimming --##
length_required: 15
length_limit: 30

##-- Minimum phred score for a base to pass quality filter --##
qualified_quality_phred: 15

##-- Minimum % of bases that can be below minimum phred score (above) --##
unqualified_percent_limit: 0

##-- Minimum allowed number of bases --##
n_base_limit: 1

##-- Compression level for gzip output --##
compression: 4

###-- Unused option inputs: Remove '#' in front to use --###
##-- Trim poly x tails of a given length --##
#trim_poly_x: false
#poly_x_min_len: 0

##-- Is the data phred 64? --##
#fp_phred64: False

##-- Turn on overrepresentation sampling analysis --##
```

```
#overrepresentation_sampling: 0
#overrepresentation_analysis: false

##-- If true: don't overwrite the files --##
#dont_overwrite: false

##-- If true: disable these options --##
#disable_quality_filtering: false
#disable_length_filtering: false
#disable_adapter_trimming: false

###-- These options are generated from sample sheet --###
# input fastq files
in_fq: ''
# output, cleaned fastq files
out_fq: ''
# output reports
report_title: ''

######------------------------- READ COLLAPSER OPTIONS -------------------------######
#
# We use a custom Python script for collapsing duplicate reads for now. There are only a
# couple options and we recommend using the default (keep all reads).
#
# We have specified default parameters for small RNA data based on our own "best practices".
# You may change the parameters here.
#
######-------------------------------------------------------------------------######

##-- Min. num of duplicate sequences required to keep --##
threshold: 1

##-- If True: write the sequences below threshold to a file --##
keep_low_counts: False

###-- These options are generated from sample sheet --###
# output filenames
uniq_seq_file: ''

######------------------------- BOWTIE2 ALIGNMENT OPTIONS -------------------------######
#
# We use bowtie2 for read alignment to a genome.
#
# We have specified default parameters for small RNA data based on our own "best practices".
# You may change the parameters here.
#
######-------------------------------------------------------------------------######

##-- Max allowed num of mismatches --##
end_to_end: 0

##-- If True: report all alignments --##
all: True
```

```
##-- Set a random seed for alignment --##
seed: 0

##-- If True: supress sam records for unaligned reads --##
no_unal: True

##-- If True: input files are fasta --##
fasta: True

##-- If True: output a sam file instead of stdout --##
sam: True

###-- Unused option inputs: Remove '#' in front to use --###
##-- If true: do not align to reverse-compliment reference --##
#norc: False

##-- If True: do not align to forward reference --##
#nofw: False

##-- If True: input quality scores are Phred64 --##
#bt_phred64: False

##-- If True: input files are fastq --##
#fastq: False

##-- Number of alignments to report --##
#k_aln: 100

##-- Number of bases to trim from 5' or 3' end of reads --##
#trim5: 0
#trim3: 0

##-- If True: input files are solexa or solexa 1.3 quality --##
#solexa: false
#solexa13: false

###-- These options generated from sample & reference sheet --###
# bowtie index files
bt_index_files: ''
# output alignment file names
outfile: ''
#unaligned read file names
un: ''

######---------------------------- FEATURE COUNTER OPTIONS ----------------------------######
#
# We use a custom Python script that utilizes the HTSeq API to count small RNA reads.
#
#
######-------------------------------------------------------------------------------######

##-- If True: save intermediate table with all information --##
```

```
intermed_file: False

###-- These options generated from sample & reference sheet --###
# output file prefix
out_prefix: ''
# reference gffs
ref_annotations: ''
# mask gffs
mask_annotations: ''
# turn on/off counting antisense alignments
antisense: ''

######------------------------- MERGE SAMPLES OPTIONS -------------------------######
#
# We use a custom Python script to merge outputs of the counter for further processing.
#
#
######-------------------------------------------------------------------------######

##-- These options are generated with output_prefix --##
output_file_stats: ''
output_file_counts: ''


######--------------------- NORMALIZATION AND STATISTICS OPTIONS -------------------######
#
# We use a custom Python script for read normalization and statistical analysis
# (differential gene expression) based on statistical methods developed in [ref]. If you
# do not want to use this method and would prefer to use a method such as DESeq2 in R,
# set use_smrna_stats to False and your output will end at the counts step.
#
# We have specified default parameters for small RNA data based on our own "best practices".
# You may change the parameters here.
#
######-------------------------------------------------------------------------######

##-- If True: use zero-inflated model for normalization and DEG calling --##
use_smrna_stats: False
use_deseq: True

######------------------------------ PLOTTING OPTIONS ------------------------------######
#
# We use a custom Python script for creating all plots. The default base style is called
# 'smrna-light'. If you wish to use another matplotlib stylesheet you may specify that here
# (i.e. ggplot or seaborn-white). You may also specify color palettes built into matplotlib
# if you do not wish to use 'smrna-light' defaults.
#
# We have specified default parameters for small RNA data based on our own "best practices".
# You may change the parameters here.
#
######-------------------------------------------------------------------------######

create_pdf: True
```

2. Sample table

The next main input is the sample table, where all the sample files to be processed are described. An example input is in Table S3.1 as it might look opened in a program like Microsoft Excel. The file is a comma-separated values (CSV) file that can be edited and saved in GUI applications like MS Excel, but can also be modified using code or other tools. The rows must correspond correctly for the tool to work. Each sample file listed in the first column needs to have a relative path from where the tool will be run or an absolute path. Then in the second column should be a descriptor for the sample, such as “wild_type” that can be used to group files into sample groups for later comparison. Column three should contain the replicate number for that sample so that biological replicates can be grouped for normalization and statistical analysis.

Supplementary Table S3.1 sample_sheet_template.csv

Input FastQ/A Files	Sample/Group Name	Replicate number
tests/testdata/gonad_seq_full_set/KB1_raw.fastq	wild_type	1
tests/testdata/gonad_seq_full_set/KB2_raw.fastq	ram2	1
tests/testdata/gonad_seq_full_set/KB3_raw.fastq	ne219	1
tests/testdata/gonad_seq_full_set/KB5_raw.fastq	wild_type	2
tests/testdata/gonad_seq_full_set/KB6_raw.fastq	ram2	2
tests/testdata/gonad_seq_full_set/KB7_raw.fastq	ne219	2

3. Reference table

The final input file is the reference table. This file tells AQuATx how to count small RNA features. Column one contains the paths (relative or absolute) to the reference annotation files in GFF3 files. The reference files should contain the class information in the third column, as this is used by AQuATx to determine class. The second column is used to point the tool to the "mask" reference GFF3 files, such that features are assigned appropriately. For instance, if a miRNA overlaps with a CSR-1 target feature (row 2), the alignment is not counted toward CSR-1, because the miRNA is used as a "mask". The third column is used to indicate if the features should count both antisense and sense read alignments, as some small RNA features might derive from both. The v1.0 version will likely have updated this input to allow for a single reference annotation input and have an alternate, automated method of masking and counting antisense reads.

Supplementary Table S3.2 reference_sheet_template.csv

Reference Annotation Files	Reference Mask Annotation Files	Also count antisense?
tests/testdata/fixed_21U.gff	None	FALSE
tests/testdata/fixed_CSR-1_Targets_All.gff	tests/testdata/fixed_miRNAs_4nt.gff	TRUE
tests/testdata/fixed_miRNAs_4nt.gff	None	FALSE
tests/testdata/fixed_mut-16_targets_ws230.gff	tests/testdata/fixed_miRNAs_4nt.gff	TRUE

Running the end-to-end analysis

Once the input files are modified they can then be used as input to the software. To run the end-to-end analysis, use the following command:

```
aquatx run --config <path/to/run_config_template.yml>
```

Where <path/to/run_config_template.yml> is replaced by the path to the run conflict file modified for the run. This command will attempt to run the entire pipeline based on the configuration file information. The pipeline first updates the configuration file given to contain all the right input lists for downstream tools. Then it runs the workflow using cwltool. If you wish to use a different workflow engine, you may generate the CWL files for the workflow and the tools using:

```
aquatx setup-cwl --config <path/to/run_config_template.yml>
```

This will copy all the necessary CWL files to the current directory under the folder “cwl”, with the workflow under “cwl/workflows” and tools under “cwl/tools” as well as create the input YAML file for running the workflow. AQuATx will also then print the configuration file name to stdout so you may find it. Together, these scripts and inputs can be used to run the entire workflow or independent steps using any CWL implementation, such as CWLEXEC.

S3.1.4 USING INDIVIDUAL STEPS AS STANDALONE TOOLS

If instead, you would like to run individual steps instead of the entire end-to-end workflow, you can get a copy of all the CWL files necessary to run each step using the

previously described command:

```
aquatx setup-cwl --config None
```

With the configuration specified as “None” to indicate you would like to configure the workflow differently. This will then simply create a copy of the tools in your current directory. Now you may use cwltool to run individual steps as well as create templates for inputs. This requires a little more knowledge on using CWL and/or cwltool described here: https://github.com/common-workflow-language/cwltool

To create a template for fastp from within the same directory the files are copied to, run the following command:

```
cwltool --make-template cwl/tools/fastp.cwl > fastp_input.yml
```

This will create a new file called “fastp_input.yml” that contains all possible inputs to the fastp.cwl wrapper. These will then need to be modified to contain appropriate settings for running fastp. This tool can be used to run on one input fastq file at a time. So to run many you will need to create an appropriate script to submit all as separate jobs or to run them over a loop. Once the configuration is setup, you can use the following command to run the tool with cwltool:

```
cwltool cwl/tools/fastp.cwl fastp_input.yml
```

This can be similarly done for each individual step where there is a CWL tool wrapper. CWL workflows can also be created or modified from the original end-to-end workflow if

you know CWL (user guide: https://www.commonwl.org/user_guide/)[1]. The RABIX composer (http://rabix.io/) can also be used to link tools together into a workflow using a GUI application for easier building of workflows. AQuATx v0.5 will have the ability to more easily run single steps across many samples and to specify sub-workflows. Additionally, as of a recent set of updates to Snakemake, individual CWL scripts can be run as part of Snakemake workflows as independent rules. Thus, if one wishes to just fold in one tool from AQuATx into their workflow, this should be straightforward with the CWL wrappers.

If one does not wish to use the CWL wrapper, each step in AQuATx we created can be called as a standalone tool as well. Third-party tools such as fastp and bowtie have their own user guides as well and can be run separately if desired. The inputs and outputs to AQuATx-specific tools should work with the inputs and outputs of third-party tools if one does not use fastp/bowtie for processing data. Each command has specified input arguments that can be found using a "help" flag. As long as the tool is installed as described above these commands will work, but each is also a Python script under the "aquatx/srna" folder of the GitHub repository and may be run using Python. All scripts can be imported as modules as well, for additional customization or running chunks of code within a Jupyter notebook, for example. Use Python's help() function to list all the functions in each module and their docstrings for information on using them.

[1] I also contributed to the user guide documentation of CWL for parallel processing of inputs and believe the explanations given in the user guide are generally sufficient for advanced users.

Counting & collapsing duplicate reads

This is the second step of the end-to-end workflow. The trimmed & filtered data is collapsed to unique sequences only, but duplicate counts are tracked in the output fasta header file.

Command to print help:

```
aquatx-collapse –h
```

Output of help command:

```
usage: aquatx-collapse [-h] -i FASTQFILE [-o OUTPUTFILE] [-t THRESHOLD]
                       [-k FILENAME]

optional arguments:
  -h, --help            show this help message and exit
  -i FASTQFILE, --input-file FASTQFILE
                        input fastq file to collapse
  -o OUTPUTFILE, --out-file OUTPUTFILE
                        output file name to use
  -t THRESHOLD, --threshold THRESHOLD
                        number of sequences needed to keep in final fasta file
  -k FILENAME, --keep-low-counts FILENAME
                        keep sequences not meeting threshold in a separate file
```

The functions of this script can also be imported within a Python script or Jupyter notebook using:

```
import aquatx.srna.collapser
```

Assigning counts to small RNA features

This tool assigns counts to reference features and classes. It also tracks the overall alignment and feature statistics. In the end-to-end workflow this step occurs immediately after alignment.

Command to print help:

```
aquatx-count –h
```

Output of help command:

```
usage: aquatx-count [-h] -i SAMFILE -r GTFFILE [GTFFILE ...]
                    [-m MASKFILE [MASKFILE ...]] [-o OUTPUTPREFIX]
                    [-a ANTISENSE [ANTISENSE ...]] [-t]

optional arguments:
  -h, --help            show this help message and exit
  -i SAMFILE, --input-file SAMFILE
                        input sam file to count features for
  -r GTFFILE [GTFFILE ...], --ref-annotations GTFFILE [GTFFILE ...]
                        reference gff3 files with annotations to count.
  -m MASKFILE [MASKFILE ...], --mask-file MASKFILE [MASKFILE ...]
                        reference gff3 files with annotations to mask from
                        counting.
  -o OUTPUTPREFIX, --out-prefix OUTPUTPREFIX
                        output prefix to use for file names
  -a ANTISENSE [ANTISENSE ...], --antisense ANTISENSE [ANTISENSE ...]
                        also count reads that align to the antisensestrand and
                        store in a separate file.
  -t, --intermed-file   Save the intermediate file containing all alignments
                        andassociated features.
```

The functions of this script can also be imported within a Python script or Jupyter notebook using:

```
import aquatx.srna.counter
```

Merging output files

This tool runs after aquatx-count in the end-to-end workflow and merges the outputs into a single table. There is more than one mode and it is run twice in the current v0.1 workflow.

Command to print help:

```
aquatx-merge –h
```

Output of help command:

```
usage: aquatx-merge [-h] -i FILE [FILE ...] -o OUTPUT -s NAMES [NAMES ...] -m
                    MODE

optional arguments:
  -h, --help            show this help message and exit
  -i FILE [FILE ...], --input-files FILE [FILE ...]
                        input count files to merge for processing
  -o OUTPUT, --output-file OUTPUT
                        output filename
  -s NAMES [NAMES ...], --sample-names NAMES [NAMES ...]
                        associated sample names for input files given
  -m MODE, --mode MODE  mode for merging: counts, stats, or unique seq files
```

The functions of this script can also be imported within a Python script or Jupyter notebook using:

```
import aquatx.srna.merge_samples
```

DESeq2 wrapper

This wrapper script can be used as an independent executable as well, but was created to specifically work with the outputs of the AQuATx tool. The command installed is:

```
aquatx-deseq
```

Running the command without input arguments will yield information about the inputs required. This is a custom, primitive argument parser written in R and while there is error-handling, it is best to be used with the workflow and following the instructions exactly.

```
No arguments given. The following arguments are accepted:

  --input-file <count_file>
        A text file containing a table of features x samples of the run to
              process by DESeq2. The count output of aquatx-merge is expected
        here.

  --outfile-prefix <outfile>
        Name of the output files to write. These will be created:
           1. Normalized count table of all samples
           2. Differential gene expression table per comparison
```

This tool expects that the input table columns are labeled following this format:

“samplegroup_replicate_num” in order to automatically make all pairwise comparisons

using DESeq2.

Plotting the outputs

The plotter tool will be assigned an entry-point command once folded into the full

workflow in v0.5 and will be similarly usable under the following command:

```
aquatx-plotter -h
```

Output of help command:

```
usage: aquatx-plotter [-h] -i DATAFILE [DATAFILE ...] -o OUTFILE -m MODE
                      [MODE ...] [-s PLOTSTYLE] [-c PLOTCOLOR]

optional arguments:
  -h, --help            show this help message and exit
  -i DATAFILE [DATAFILE ...], --input-files DATAFILE [DATAFILE ...]
                        input files with data from final merged tables
  -o OUTFILE, --out-prefix OUTFILE
                        prefix to use for output PDF files. If mode=all/by-
                        sample, sample.names will also be appended to the
                        prefix.
  -m MODE [MODE ...], --data-types MODE [MODE ...]
                        List of data types corresponding to input files.
```

```
                Options: raw_counts, norm_counts, degs, len_dist,
                class_counts
-s PLOTSTYLE, --style PLOTSTYLE
                plotting style to use. Default: smrna-light.
-c PLOTCOLOR, --color PLOTCOLOR
                color palette to use for data instead of default
```

For now, the plotter may be used with python as a script or run as a module. Functions can be imported directly within a Python script or Jupyter notebook using:

```
import aquatx.srna.generate_plots
```

However, as it is under development this script is not stable in v0.1.

S3.2 BASIC WORKFLOW OUTPUTS

The pipeline will produce all intermediate files by default in v0.1. We provide output tables and multiple visualizations in a vector-editable format. Some outputs are displayed in Chapter 3.

FastQ Quality Metrics HTML Report

fastp produces summary and quality statistics for each of your raw fastq files and outputs them to HTML format. The plots are created using plotly and are interactive. Unlike the tool, FASTQC, which produces a similar quality control report, fastp analyzes the entire fastq file to create this report. FASTQC only uses the first 200k sequences and is a tool separate from adapter trimming & quality filtering, so to look at the quality metrics before and after, it must also be run twice.

Size Distributions Table & Plot

After alignment, a size and 5'-nt distribution table and histogram is created. The distribution of lengths and 5'-nt can be used to assess the overall quality of your libraries. This can also be used for analyzing small RNA distributions in non-model organisms without annotations.

Alignment Statistics Table

The aquatx-count tool also produces a summary table of the alignment information. This includes the number of alignments, the number of alignments with more than one genome-hit, alignments without a feature, and ambiguous alignments matching more than one feature or class.

Class Count Tables and Plots

AQuATx produces pie and bar charts to display the class distribution of your aligned reads as well as a count table. Classes are defined from the reference GFF3 files (column 3).

Feature Count Tables and Plots

After read counting, count files for all features and samples are produced then merged into a single table. A row with the number of reads that were not assigned a feature is also included. Scatter plots can be created with the plotter tool for comparing raw

counts or normalized counts (output of aquatx-deseq) per feature between replicates or biological groups.

Differential Gene Expression Tables

The output of aquatx-deseq includes tables of all small RNAs, their fold change among comparisons, and associated p-values as determined by DESeq2. One table per comparison is created and the comparisons are inferred from the original sample sheet groups.

S3.3 ADDITIONAL CODE

A few additional scripts and software were created during the completion of this dissertation, but not necessarily generic or well-developed enough to be useful to those outside of our lab and a few others. They are briefly described here and linked for interested parties (either in assisting with development or using them).

S3.3.1 AQuATx helper scripts

I wrote helper scripts that might be useful to a future developer and could be generalized more. They are included as part of the main aquatx-srna repository on GitHub, but not installed for users.

aquatx/srna/process_annotations.py

To make sure that the chromosomes were labeled the same in the input GFF files and the genome fasta file/genome index files, I wrote a simple helper script to convert from one format to another. This is highly-specific for *C. elegans* genome, using the following dictionary to convert values:

```
chrdict = {'1': 'I', '2': 'II', '3': 'III', '4': 'IV', '5': 'V', '6': 'X', '7': 'MtDNA'}
```

This could be expanded to detect differences and automatically update them to match.

tests/create_test_data.py

To create validation tests for the software, I wrote a script to generate some randomized test data. The tool specifically works for *C. elegans*, but could easily be adapted to other organisms. It currently generates a dummy fastq file of an assortment of microRNAs to give to the pipeline and validate the outputs.

S3.3.2 shinySeqBrowser

As a personal side project, I created a standalone tool in R shiny similar to genome browsers, described here: https://github.com/biokcb/shinySeqBrowser. The app can be run from any computer with R and the appropriate dependencies described under the installation guide. This app allows for easy browsing of RNA-seq reads across transcripts. The user can upload a BAM file and an annotation GFF3 file. The backend will process these files to produce reads across regions of interest. The plots can be searched by chromosome from a drop-down menu & specified position or by searching for the transcript name. The plot colors can be adjusted and then exported as a vector-

editable PDF, allowing users to make further edits without difficulty. This project will likely continue to be updated as a side project as time allows.

S3.3.3 Miscellaneous scripts

https://github.com/biokcb/bio-data-analysis/blob/master/scripts/next_gen_sequencing/analyze_tails.py
This analysis looked at the trimming and tailing of mRNA sequencing data of small RNA pathway mutants. It requires TopHat2 and samtools. The tool uses the unaligned sequence file, then realigns them, trimming one nucleotide off at a time, until the sequence is aligned. The “tails” are recorded in the output file and can be analyzed and plotted using additional scripts like this one written in R for a very specific analysis I did: https://github.com/biokcb/bio-data-analysis/blob/master/scripts/next_gen_sequencing/2017-10-03_tailing_plots.R

My commonly used R functions were compiled into a “library” of sorts. This script was created before I learned “best practices” and contains hardcoded paths that must be updated. https://github.com/biokcb/bio-data-analysis/blob/master/scripts/next_gen_sequencing/kristens_common.R

For the curious, additional data analysis scripts created may be found here (https://github.com/biokcb?tab=repositories), though not everything was published online.