DEVELOPMENT OF BIOINFORMATIC TOOLS AND EPIGENETIC APPROACHES FOR THE STUDY OF *SCHMIDTEA MEDITERRANEA*

by

Sofia Maria Cristina Robb

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Neurobiology and Anatomy

The University of Utah

August 2011

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Sofia Maria Cristina Robb**

has been approved by the following supervisory committee members:

Alejandro Sánchez Alvarado , Chair **05/23/11**
Date Approved

Brad Cairns , Member **05/23/11**
Date Approved

Shannon Odelberg , Member **05/23/11**
Date Approved

Monica Vetter , Member **05/23/11**
Date Approved

Mark Yandell , Member **05/23/11**
Date Approved

and by **Monica Vetter** , Chair of the Department of **Neurobiology and Anatomy**

and by Charles A. Wight, Dean of The Graduate School.

ABSTRACT

Schmidtea mediterranea is being established as a standard model system for studying regeneration and adult stem cells (ASCs). This is largely due to the developmental plasticity of the planarian and the abundant distribution and experimental accessibility of the ASCs. Techniques such as whole mount *in situ* hybridization (WISH), dsRNA-mediated interference (RNAi), utilization of halogenated thymidine analogs, and fluorescence activated cell sorting (FACS) allow for studying ASCs *in vivo*. The development of bioinformatic tools is also required for this to be a reality. Before the genome was sequenced, the main bioinformatic resource was a web-accessible database containing a large collection of expressed sequence tags (ESTs), up-to-date annotations, and expression data (SmedDb). With the sequencing of the genome and genome assembly available, it was feasible to create and use tools to annotate (MAKER) and distribute the genome, annotations and experimental data (SmedGD). With these tools in place it was possible to proceed with investigating biological questions, specifically epigenetic states and regulation of ASCs. Given the role chromatin architecture plays in defining the genomic output of a given cell, the multipotentiality found in the stem cells of *S. mediterranea* is likely to be no different. Canonical histones, H3, H4, H2A, H2B, the linker H1 and the variants H3.3, H2A.X, H2A.Z are present in the *S. mediterranea* genome. Immunocytochemistry has confirmed differential levels of histone posttranslational modifications in different cell types, specially acetylation and methylation of histone H3. My aims were to develop the necessary bioinformatic tools, establish protocols for studying epigenetics in planarian, to determine if any epigenetic modifying enzymes are localized to ASCs, and identify if any of these enzymes have specific roles in ASC function. I assisted in the development of systems to keep SmedDb up-to-date, in the annotation (MAKER) of the genome, and the web accessibility of the genome and associated data (SmedGD). I created a library of genes for approximately 90 epigenetic modifying enzymes. The expression patterns of these genes were visualized with WISH and their function was perturbed with RNAi. Six genes were identified that are likely to have key roles in stem cell self-renewal, maintenance, and differentiation.

This work is dedicated to my mom, dad, and sisters, for the love, help, guidance, patience, and excitement they have shared with me. And to Alejandro for all the opportunities he has created for me.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

There is a growing need for all biologists to use bioinformatic tools, much like running a gel or doing PCR. In 2008 there were 99,116,431,942 basepairs (bp) and 98,868,465 sequences in GenBank (GenBank Statistics, 2009). In fact, the number of sequences is growing exponentially each year. There are approximately 250 sequenced eukaryote genomes, making up ~120Gb (gigabases) and ~4000 bacteria and viruses with ~5 Gb (Lander, 2011). There are terabytes of published RNA-seq and microarray data, the annotations of noncoding RNAs, conserved noncoding elements, transposons, epigenomic maps, information on the three-dimensional structures of the genome, splice sites, splice variants, protein domains, crystal structures, and folding motifs. It would be unthinkable to let all this information sit on hard drives around the world and not use it to formulate testable hypotheses. Biologists need to understand how to use these databases and the tools that make structural and functional predictions. We need to be able to search this wealth of information, sift through gigabytes of results to create informed and manageable lists used to define experimentally accessible questions.

It is in the phrase “sift through gigabytes of results,” that a disconnect exists between bioinformatics and experimental bench biology. How does a biologist with no programming experience or a programmer with no biological knowledge isolate the highly interpretable and “important” information from blast results of hundreds or more genes? How would a list of hundreds of upregulated transcripts be managed in which the upstream 1000bp is needed to search for regulatory elements and the Gene Ontology (GO) categorization (Ashburner et al., 2000) and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway (Kanehisa et al., 2010) information is required for the transcripts? An individual that is capable of accomplishing the steps required to interrogate intersecting datasets and filter them based on logical and biological rules is needed for such tasks. After a first look at a filtered list, a researcher is likely to realize that the original filtering criteria need to be altered because they may be too lax or too stringent, or a factor that is evident after reviewing a resulting list needs to be taken into consideration. Once these results are taken to the bench, new information may elucidate additional criteria for limiting the data. Many iterations of this analysis and filtering is likely to occur and are often necessary for the production of meaningful results. Such an involved process is better accomplished with someone who has an understanding of the computation and the biology required to create these lists for experimentation and the time to interact with the researcher. It seems that there is no one better suited for this task than the individual researcher.

The need for bioinformatics became palpably evident with the growing numbers of genome sequencing projects. The increase in data collected and the need for the development of resources for the storage, analysis, and sharing of this data are always pushing current technology to the limit. And no group was in greater need of bioinformatic tools than those working to understand the biology of the human genome and human disease biology.

It has been 10 years since the first draft of the human genome was released. Previously, only 100 disease genes had been identified. Now approximately 3000 genes causing disease via Mendelian inheritance and more than 1100 loci contributing to common polygenic disorders have been found (Lander, 2011). The obtained genomic information has been exploited to carry out high-throughput studies such as genome-wide association studies, DNA microarrays, and RNA-seq in both research and clinical trials. The overarching goal of these efforts has been to identify single nucleotide polymorphisms (SNPs) and transcriptional profiles of patients for disease diagnosis, predisposition, and personalized treatments. For instance, a 6.9 million-feature oligonucleotide array of the human transcriptome allows a broad assessment of gene expression. This array was also designed to provide information on alternative splicing as well as identification of SNPs and noncoding transcripts (Xu et al., 2011). Another approach to mine the complexity of the human genome has been to perform genome-wide association studies and this is being used to identify associations between specific chromosomal loci and complex human diseases (Hardy and Singleton, 2009). Nevertheless, even with such great accomplishments, not many advances in disease prevention or treatment have arisen in the past decade (Green et al., 2011). This is in part due to an incomplete but ever growing understanding of biology of the genomes and the biology of diseases and a need for furthering medical diagnostic techniques (Green et al., 2011).

Model organisms were recognized very early on to be necessary for a complete understanding of the human genome, as is evidenced in the report on the mapping and sequencing of the human genome by the National Academy of Sciences/National Research Council, due to the availability of experimental techniques and the ability to formulate testable hypothesizes that are not possible in humans (Alberts et al., 1988). Model organisms such as *Drosophila melanogaster* and *Caenorhabditis elegans* have proven to be very effective systems to study countless biological questions. Their ease of care and amenability to experimentation has allowed many techniques to be developed. These organisms have complete genome sequences, forward genetics, reverse genetics, transgenesis and thousands of researchers contributing to the collective knowledgebase, which includes but is not limited to genome organization, gene and protein function, epigenetics and noncoding regulatory elements.

Despite the long list of benefits of using these model organisms, they have only a small number or a complete absence of adult somatic stem cells (ASCs) (Micchelli and Perrimon, 2005; Pearson and Alvarado, 2009). ASCs are undifferentiated cells that reside in adult tissues that can self-renew or differentiate into a variety of predetermined cell types. The primary role of these cells is to maintain, repair, and/or replace damaged cells of the tissue in which they are housed. In humans, ASCs can be found in the bone marrow, blood vessels, skeletal muscle, liver, skin, teeth, heart, gut, testis and ovarian epithelium. These cells are few in number and difficult to access and to culture (Stem Cell Basics, 2010).

Many questions about the biology of ASCs exist, such as what factors influence the undifferentiated ASC to proliferate or differentiate, how do they “know” what cell types to become, what keeps them undifferentiated, how is epigenetics involved in the decisions that lead to proliferation and/or differentiation of ASC, where do ASC cells come from, do ASC reside in a niche, and if so, what cells types make up this niche? Many of these questions are particularly challenging to address in *C. elegans* or *D. melanogaster*.

Schmidtea mediterranea, a fresh water, nonparasitic planarian, possess an abundance of ASCs that are distributed throughout their body. These cells enable the worm to replace lost tissues via regeneration and to be an excellent model for tissue homeostasis. The full extent of the abilities of ASCs can be easily witnessed in this animal with a simple slice of a blade. A cephalically amputated worm will regenerate the head along with all missing structures, including the complete brain, photoreceptors, any sensory cells, and neurons in an amazingly short time period of 7 days (Figure 1.1). The amputated head will in turn regenerate the lost portions of the gastrovascular system, the ventral cords and commissural neurons and any other missing structures.

Many experimental methodologies have been developed that enable the interrogation of the biology of ASCs in planaria amenable. These techniques include dsRNA-mediated inference (RNAi) (Sánchez Alvarado and Newmark, 1999; Newmark, 2003) to disrupt specific gene products, whole mount *in situ* hybridization (Pearson et al., 2009) to identify expression domains of mRNAs, and fluorescence activated cell sorting (FACS) (Reddien, 2005; Higuchi et al., 2007) for isolating ASCs.

What the planarian community was lacking was a foothold in bioinformatic tools. Since the introduction of bioinformatic tools, many accomplishments have helped to establish *Schmiditea mediterranea* as a model organism for the study of regeneration and ASC biology. A large-scale RNAi screen was preformed to identify genes needed for regeneration, stem cell function, and tissue homeostasis (Reddien et al., 2005). This screen used RNAi to perturb the function of 1065 genes that were pulled from the collection of expressed sequence tags (ESTs) that were curated in SmedDb (Sánchez Alvarado, 2002). This collection of ESTs was also used to print the first cDNA microarray which was key in identifying markers for stem cell lineage analysis (Eisenhoffer et al., 2008). Use of SmedGD has been instrumental in the identification of *beta-catenin* and *wnt* pathway components (Gurley et al., 2010).

In the next few chapters I will describe the bioinformatic tools that I helped to develop for *Schmidtea mediterranea*. These tools include a semiautomated system for updating annotations of ESTs (Orendt et al., 2006) that are stored and organized in SmedDb (Sánchez Alvarado, 2002), a pipeline for gene annotations of genome sequence, MAKER (Cantarel et al., 2008), and a genome browser, SmedGD (Robb et al., 2007). I will also describe the biological experiments I performed to address questions of epigenetics in ASCs that were made possible with the utilization of these bioinformatic tools.

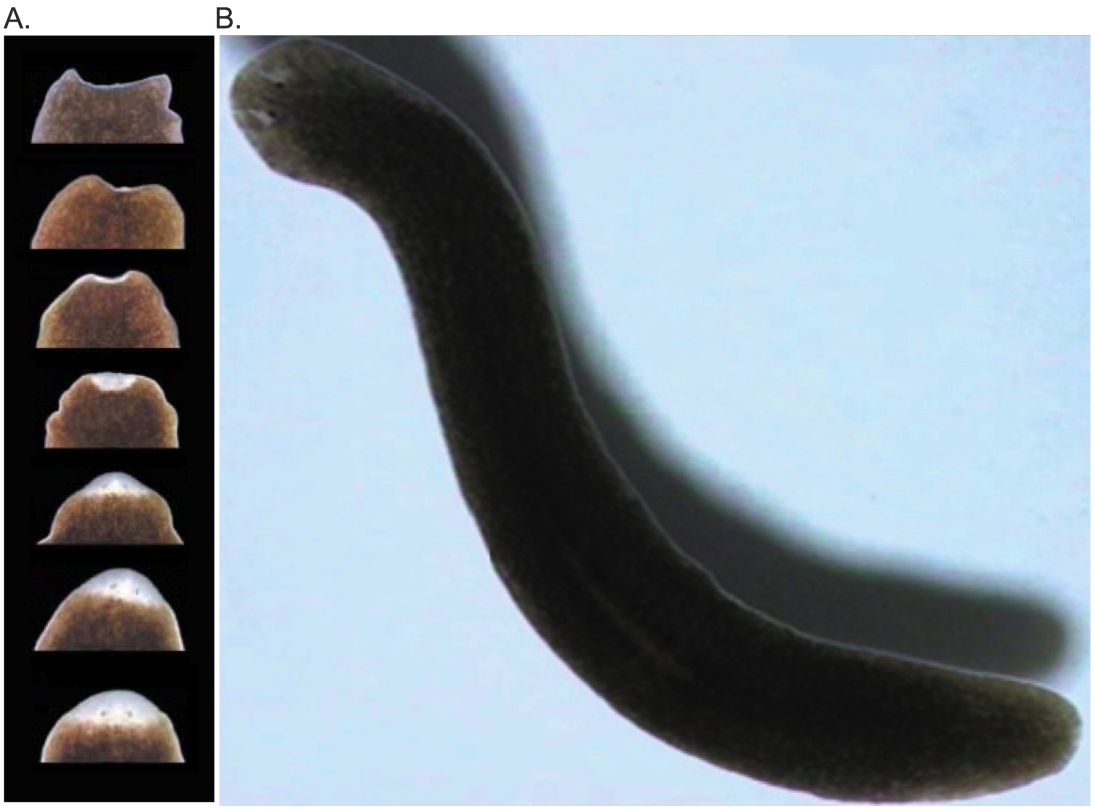


Figure 1.1: *Schmidtea mediterranea* and regeneration. (A) A worm that has been cephalically amputated at time 0, top image in series, takes seven days, bottom image in series, to regenerate all missing structures. (B) An intact *Schmidtea mediterranea.*

References

Alberts, B. M., Botstein, D., Brenner, S. E., Cantor, C. R., Doolittle, R. F., Hood, L., McKusick, V. A., Nathens, D., Olsen, M. V., Orkin, S. et al. (1988) Report on the Committee on the Mapping and Sequencing of the Human Genome. Washington, D.C.: Board on Basic Biology, Commision on Life Sciences, National Reserach Council.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T. et al. (2000) 'Gene ontology: tool for the unification of biology. The Gene Ontology Consortium', *Nat Genet* 25(1): 25-9.

Cantarel, B. L., Korf, I., Robb, S. M., Parra, G., Ross, E., Moore, B., Holt, C., Sánchez Alvarado, A. and Yandell, M. (2008) 'MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes', *Genome Res* 18(1): 188-96.

Eisenhoffer, G. T., Kang, H. and Sánchez Alvarado, A. (2008) 'Molecular Analysis of Stem Cells and Their Descendants during Cell Turnover and Regeneration in the Planarian Schmidtea mediterranea', *Cell Stem Cell* 3(3): 327-339.

GenBank Statistics (2009) GenBank Statistics, http://www.ncbi.nlm.nih.gov/genbank/genbankstats.html, accessed April 21 2011.

Green, E. D., Guyer, M. S., Manolio, T. A. and Peterson, J. L. (2011) 'Charting a course for genomic medicine from base pairs to bedside', *Nature* 470(7333): 204-213.

Gurley, K. A., Elliott, S. A., Simakov, O., Schmidt, H. A., Holstein, T. W. and Sánchez Alvarado, A. (2010) 'Expression of secreted Wnt pathway components reveals unexpected complexity of the planarian amputation response', *Developmental Biology* 347(1): 24-39.

Hardy, J. and Singleton, A. (2009) 'Genomewide Association Studies and Human Disease', *The new england journal of medicine* 360: 1759-68.

Higuchi, S., Hayashi, T., Hori, I., Shibata, N., Sakamoto, H. and Agata, K. (2007) 'Characterization and categorization of fluorescence activated cell sorted planarian stem cells by ultrastructural analysis', *Development, Growth & Differentiation* 49(7): 571-581.

Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M. and Hirakawa, M. (2010) 'KEGG for representation and analysis of molecular networks involving diseases and drugs', *Nucleic Acids Res* 38(Database issue): D355-60.

Lander, E. S. (2011) 'Initial impact of the sequencing of the human genome', *Nature* 470(7333): 187-197.

Micchelli, C. A. and Perrimon, N. (2005) 'Evidence that stem cells reside in the adult Drosophila midgut epithelium', *Nature* 439(7075): 475-479.

Newmark, P. A. (2003) 'Ingestion of bacterially expressed double-stranded RNA inhibits gene expression in planarians', *Proceedings of the National Academy of Sciences* 100(90001): 11861-11865.

Orendt, A. M., Haymore, B., Richardson, D., Robb, S. M. C., Sánchez Alvarado, A. and Facelli, J. C. (2006) Design, Implementation and Deployment of a Commodity Cluster for Periodic Comparisons of Gene Sequences. in L. T. Yang and M. Guo (eds.) *High-Performance Computing : Paradigm and Infrastructure*. Hoboken, NJ: John Wiley and Sons, Inc.

Pearson, B. J. and Alvarado, A. S. (2009) 'Regeneration, Stem Cells, and the Evolution of Tumor Suppression', *Cold Spring Harbor Symposia on Quantitative Biology* 73(0): 565-572.

Pearson, B. J., Eisenhoffer, G. T., Gurley, K. A., Rink, J. C., Miller, D. E. and Sánchez Alvarado, A. (2009) 'Formaldehyde-based whole-mount in situ hybridization method for planarians', *Developmental Dynamics* 238(2): 443-450.

Reddien, P. W. (2005) 'SMEDWI-2 Is a PIWI-Like Protein That Regulates Planarian Stem Cells', *Science* 310(5752): 1327-1330.

Reddien, P. W., Bermange, A. L., Murfitt, K. J., Jennings, J. R. and Sánchez Alvarado, A. (2005) 'Identification of genes needed for regeneration, stem cell function, and tissue homeostasis by systematic gene perturbation in planaria', *Dev Cell* 8(5): 635-49.

Robb, S. M. C., Ross, E. and Alvarado, A. S. (2007) 'SmedGD: the Schmidtea mediterranea genome database', *Nucleic Acids Research* 36(Database): D599-D606.

Sánchez Alvarado, A. (2002) 'The Schmidtea mediterranea database as a molecular resource for studying platyhelminthes, stem cells and regeneration', *Development* 129(24): 5659-5665.

Sánchez Alvarado, A. and Newmark, P. A. (1999) 'Double-stranded RNA specifically disrupts gene expression during planarian regeneration', *Proceedings of the National Academy of Sciences, USA* 96: 5049–5054.

Stem Cell Basics (2010) Stem Cell Basics: What are adult stem cells?, http://stemcells.nih.gov/info/basics/basics4.asp, accessed April 21 2011.

Xu, W., Seok, J., Mindrinos, M. N., Schweitzer, A. C., Jiang, H., Wilhelmy, J., Clark, T. A., Kapur, K., Xing, Y., Faham, M. et al. (2011) 'Human transcriptome array for high-throughput clinical studies', *Proceedings of the National Academy of Sciences* 108(9): 3707-3712.

CHAPTER 2

DESIGN, IMPLEMENTATION AND DEPLOYMENT OF A COMMODITY CLUSTER FOR PERIODIC COMPARISONS OF GENE SEQUENCES[1]

[1] Authors: Anita M. Orendt, Brian Haymore, David Richardson, Sofia Robb, Alejandro Sánchez Alvarado, and Julio C. Facelli
Chapter 39 of High-Performance Computing: Paradigm and Infrastructure By Laurence Tianruo Yang, Minyi Guo
Reprinted with permission from Mr. Brenton R. Campbell - Coordinator, Global Rights - John Wiley & Sons, Inc.

Abstract

Before the genome of *Schmidtea mediterranea* was sequenced the community possessed a collection of ESTs stored in a web-accessible database, *Schmidtea mediterranea* Database (SmedDb) (Sánchez Alvarado et al., 2002). SmedDb integrates *in situ* hybridization expression data and homology inferred from BLAST searches. ESTs are key reagents for printing DNA microarrays, carrying out large-scale spatial expression pattern studies, and the functional characterization of proteins. The utility of an EST collection depends in great part on determining the homology of the sequence, as these types of comparisons allow for the refinement of functional characterization and experimental design. The databases housing sequence information at the NCBI are ever growing and the task of keeping the sequence homology information on thousands of EST sequences most current can be cumbersome. In attempts to overcome this issue, we devised a semiautomatic system with a low-cost, commodity-based cluster computer that integrates SmedDb and BLAST results from thousands of BLAST searches, performed only when the NCBI databases change. The following manuscript is Chapter 39 from "High-Performance Computing: Paradigm and Infrastructure," in which I contributed the work required to specifically integrate SmedDb with the semi-automated updating system. I also contributed the section entitled "Integration with SmedDb".

CHAPTER 39

Design, Implementation and Deployment of a Commodity Cluster for Periodic Comparisons of Gene Sequences

ANITA M. ORENDT, BRIAN HAYMORE, DAVID RICHARDSON,
SOFIA ROBB, ALEJANDRO SANCHEZ ALVARADO,
and JULIO C. FACELLI

39.1 INTRODUCTION

In 1982, the number of sequences deposited in GenBank was 606, comprised of only 680,338 base pairs. By 1996, the number of sequences reached a little over one million with a total of nearly 652 million base pairs. Today, GenBank is made up of over 22 million sequences representing almost 29 billion base pairs (http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html). Behind the precipitous rise of sequencing data during the past four years has been the equally rapid improvement of DNA sequencing methodologies. Such improvements made possible what in the mid-1990s was thought to be nearly impossible: to completely sequence the genomes of multicellular organisms, including *Homo sapiens* [1]. In fact, in the next two years alone genome sequences for the chimpanzee, rhesus macaque, mouse, dog, cow, pig, chicken, zebrafish, the frog *Xenopus tropicalis,* sea urchin, honeybee, and the planarian *Schmidtea mediterranea* will be obtained and deposited in GenBank (http://www.genome.gov/page.cfm?pageID=10002154). The combined sequence data from these animals is expected to total 27 billion nucleotides, or roughly the same amount of sequence information available in GenBank today. It is clear, therefore, that the exponential growth of GenBank is unlikely to abate any time soon.

Because genetic sequences do not provide any direct biological information, researchers have to conduct elaborate experiments to determine the relationships between a genetic sequence and its biological function. This process can be significantly accelerated if one takes into account that there are important similarities

High-Performance Computing: Paradigm and Infrastructure. Edited by L. T. Yang and M. Guo

between genetic sequences associated with similar biological functionality across different species. Therefore, when determining the biological functionality of a new sequence it is of great value to find homologous sequences from other organisms in which their functionality is better understood.

The tools used to find sequence similarities between a researcher's protein or DNA sequence and the entries in GenBank are the suite of BLAST (Basic Local Alignment Search Tool) programs (*blastx, blastn, blastp, tblastx*) [2] available from the National Center for Biotechnology Information (NCBI) of the National Library of Medicine of the National Institutes of Health. These programs are described in detail on the NCBI's main BLAST Web site (http://www.ncbi.nlm.nih.gov/BLAST). The NCBI also maintains a number of databases available for download that are derived from the sequences deposited in GenBank. The NCBI site offers several search methods (webblast, networkblast, and blast URL API) that are sufficient for the needs of many laboratories working on a relatively small number of genes. NCBI also offers the tools necessary for individuals and institutions to maintain their own Web site with local copies of the various NCBI databases; this can be used in order to relieve part of the high demand on the NCBI-maintained Web site. The Center for High Performance Computing (CHPC) has done this at the University of Utah. However, for laboratories in need of analyzing thousands of individual sequences on a regular basis, these tools are not sufficient. For these cases, the NCBI offers also the standalone BLAST executables necessary for any individual to establish an in-house system as described in this chapter.

Our lab has been engaged in the identification of genes that are active during a variety of biological processes such as tissue regeneration and stem cell biology in the freshwater planarian *Schmidtea mediterranea* [3]. A total over of 6500 unique cDNA sequences of expressed genes, also known as expressed sequence tags (ESTs), have been accumulated. ESTs are key reagents for printing DNA microarrays, carrying out large-scale spatial expression pattern studies, and the functional characterization of proteins. The utility of an EST collection depends in great part on determining if the obtained sequence has been identified in other organisms, as these types of comparisons allow for the refinement of functional characterization and experimental design. However, comparing over 6500 ESTs to the NCBI databases on an individual basis would take hours if not days of supervised activity, not only during the performance of the BLAST searches themselves, but also in the archiving of the results of the searches into a laboratory database known as the *Schmidtea mediterranea* Database (SmedDb) [3].

In order to overcome the difficulties described above, the SmedDb used for the storage and organization of our sequences and their BLAST results has been integrated with a low-cost, commodity-based cluster computer system that can semiautomatically process thousands of BLAST searches when the NCBI databases change. The end result for the investigator is a dynamic database that is regularly and automatically updated to obtain the most up-to-date sequence comparisons available.

Cluster computing [4] has always provided an attractive approach to provide computer resources to scientific problems. In recent years, the advent of commodity clusters using the LINUX operating system has provided special impetus to the use of cluster architectures in technical and scientific environments, including grid computing [5]. Most reports in the literature address the design and implementation of these types of architectures as general-purpose systems with an intended workload encompassing many scientific applications. Nonetheless, the ample configuration space available when designing a system using commodity hardware allows for specialization when desired. This chapter describes the design, implementation, and deployment of a computer cluster dedicated to perform periodic BLAST searches and the manner in which the output of these searches is integrated into a laboratory database, SmedDb.

39.2 SYSTEM REQUIREMENTS AND DESIGN

The design of the system proceeded using the following guiding principles derived from the scientific considerations described above, along with the financial and operational constraints:

1. All the components of the cluster should be off the shelf to keep costs within the budget typically available at most biomedical labs.
2. The initial computing capacity of the search engine for the cluster should allow processing in parallel of most of the updates to SmedDb in less than 48 hours.
3. The cluster should be scalable, so the computing capacity can be increased as the size of the SmedDb and NCBI databases increase.
4. Considering that CHPC is a central university facility, the scalability of the cluster should allow the addition of computing capacity to support other users with similar requirements.
5. The download and processing of updated NCBI databases should proceed without interfering with ongoing searches. This is important because, when performing searches lasting more than 24 hours, there is a considerable probability that updates may be occurring during the processing time.
6. Database updates, process scheduling, submission of searches and retrieval of the results should be as automatic as possible to allow for high throughput without human intervention. But the system should allow for intervention when necessary to make judicious evaluations of the results.
7. Although in this implementation the parallel cluster search system has been integrated to the SmedDb, the design should be flexible enough to permit its integration to diverse laboratory-specific data management systems.

At the very beginning of the design process, it became apparent that the fundamental constraint in the design was the data management scheme needed to update and distribute the database files containing the approximately 15 Gbytes of data deposited in the major NCBI databases. The key issue was how to move this relatively large amount of data across the system in a manner that does not create bottlenecks that may affect the scalability or run-time goals of the project. Although it was obvious that the performance targets were achievable using more expensive proprietary solutions, their use would conflict with the desire to use commodity hardware to keep the cost of the system down.

In the design of the system, there were two choices for the location of the databases: (1) to have a global repository for the data files or (2) to replicate the files locally on each compute node of the system. There are obvious disadvantages to both of these options; the first may produce an IO bottleneck when several processors try to access the same files, whereas the second may require significant time for the data migration of the files to all the nodes on the cluster and significantly increase the total amount of disk space required to implement the system. To better understand the requirements, extensive tests of the IO behavior of BLAST searches in a cluster environment were performed. The results showed that neither of the models was totally satisfactory and that the results were highly dependent on the nature of the scheduled search. We decided to implement a hybrid model in which two generations of the formatted NCBI data files are kept in globally accessible space. Searches have the option of using these copies of the databases or replicating the necessary database files on a disk local to the compute nodes at run time. If the submitted job will be running over the entire course of the nightly database update, a process that takes about three hours, the user must make a local copy; however short runs do not have to waste the computer time that making a copy takes. The use of a copy of the database on local disk also permits a user to run multiple searches on different nodes using different versions of the databases. The current implementation uses remote copy to transfer the files from the global space to the individual nodes assigned to a given search, requiring multiple reads of the globally stored files. Although this has not been a bottleneck in our relative small system, in the near future we will change this implementation by using GridFTP (http://www.globus.org/research/papers.html#Data%20Grid%20Components) for replicating the files, hopefully decreasing the load on the file server and, consequently, increasing the scalability as we increase the number of search nodes to meet the needs of additional researchers.

The next area of concern was the downloading of the daily updates of the NCBI database files without degrading the performance of the rest of the system that was running searches. To solve this problem we decided to use a separate node to process the download of the datasets. Taking advantage of the low cost of commodity disks, it was possible to acquire the large disk space that allows keeping multiple copies of the databases while using a clever update scheme that precludes interference between the updates and the searches. This scheme is explained in detail below.

The computational capacity of the cluster can be easily scaled to meet the time requirements by adding processors to the system. Note that by decomposing the input search streams as described below, it is possible to obtain almost perfect linear parallel performance improvement of the searches. This scheme makes the searches embarrassingly parallel, isolating scalability issues to the data management and IO schemes discussed above.

39.2.1 Hardware Configuration

The final hardware configuration for the cluster (see Figure 39.1) consists of eight dual-processor AMD Athlon MP 2000+ search nodes, each with 2 GB of RAM and a moderate (60 GB) amount of local disk provided for the option of using local space for storage of the database files. The core file server, used to provide the global disk space, is also a dual AMD Athlon MP 2000+ with 1 GB of RAM and 240 GB of usable space in a RAID array configuration, optimized for NFS read performance. This node is also used to provide cluster services like scheduling and accounting. A specialized node was added for processing the daily updates of the database files. The performance of this node is not relevant, as most of the delays in the downloading of the databases are introduced by the network and source host constraints. The local space in this node is used to hold the newly downloaded NCBI database files and to provide space to *untar* the files before migrating them to the global file server. The intent of this design is to keep the nightly database updates from impacting the load on the global file server as much as possible. Finally, an interactive node was added to allow users to gain a login shell access to interact with the queuing system as needed. All of the nodes in the cluster are internally connected via a GigE network using a Foundry Big Iron 15000 switch supporting jumbo frames. The internal connection of the nodes via a private vLAN makes them inaccessible from outside the cluster. The interactive node (sequence) and the node

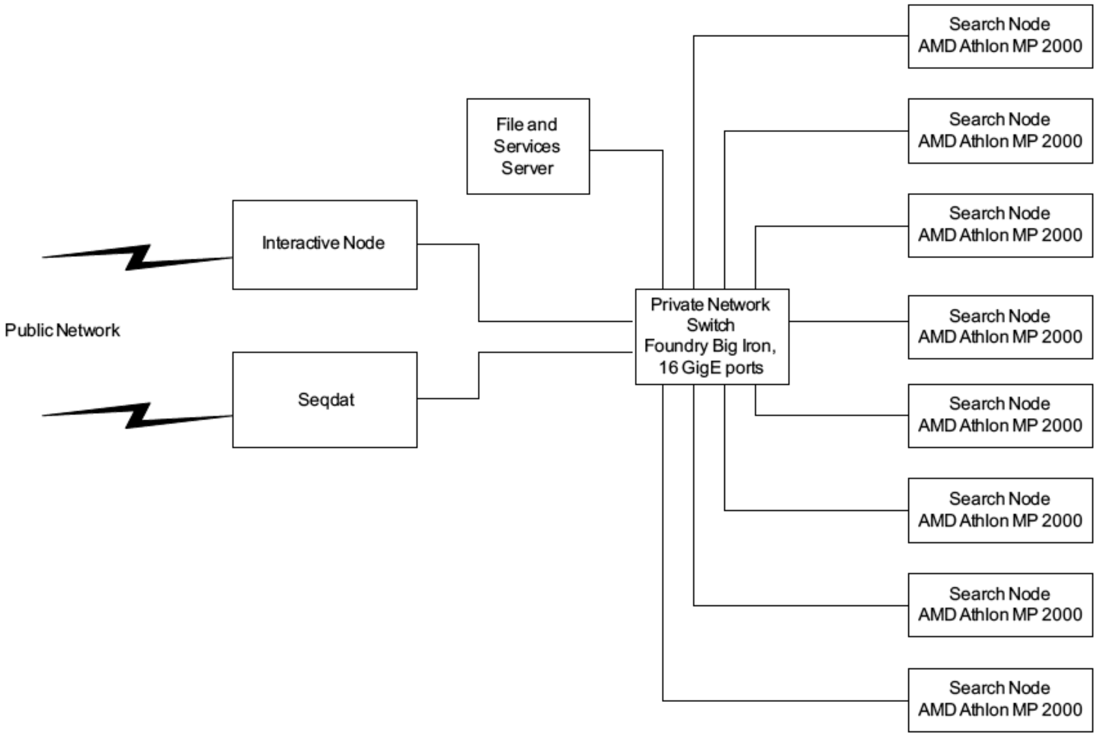

Search Node AMD Athlon MP 2000
File and Services Server
Search Node AMD Athlon MP 2000
Interactive Node
Search Node AMD Athlon MP 2000
Private Network Switch Foundry Big Iron, 16 GigE ports
Public Network
Search Node AMD Athlon MP 2000
Seqdat
Search Node AMD Athlon MP 2000
Search Node AMD Athlon MP 2000
Search Node AMD Athlon MP 2000
Search Node AMD Athlon MP 2000

Figure 39.1 Architecture of the cluster for BLAST searches.

for database updates (seqdat) are multiported, connected to both to the campus-wide area network and to the private network with the rest of the nodes. This scheme provides a higher degree of security by concentrating the access points in two nodes and eliminating the possibility of external threats on the rest of the nodes of the cluster.

39.2.2 Software Configuration

All the nodes of the cluster are running LINUX Redhat 8.0 as the operating system. OpenPBS 2.3.16 (http://www.openpbs.org) is used for resource management, Maui 3.2.6 (http://www.supercluster.org) for scheduling the jobs, and QBank 2.10 (http://www.emsl.pnl.gov/docs/mscf) for the accounting. In its current implementation, with the system being used only by one research group, it would be possible to manage the job of scheduling manually or by some easy-to-implement *cron* job. However, OpenPBS and Maui were still implemented as feature-rich queue systems (6), allowing for a scalable software infrastructure that can be used to manage the workflow of numerous research groups when needed.

All nodes are able to run the full BLAST suite of programs that were downloaded from the NCBI site and made available via NFS. After receiving notice from the NCBI mailing list on software updates, the programs are updated manually. After installation and testing of the newest code, the *std* link is changed from the old to the new version. Using this update mechanism, there is no need to make any changes in the scripts, which use the *std* link nomenclature, preventing interference with already running jobs and providing the flexibility to run searches using previous versions of the software.

39.2.3 Database Files Refreshing Scheme

As discussed above, the task of maintenance of the database files is delegated to a dedicated node of the cluster, seqdat. The procedure to update the local databases from the NCBI ftp site is performed on a nightly basis as a *cron* job. The large database files are present on this ftp site as preformatted files in FASTA format. On seqdat, three database directories are available: ~/db, ~/db_backup, and ~/db_source. The ~/db_source directory, located on a disk local to seqdat as it is only needed for the download process, contains the compressed *tar* files as received from the NCBI ftp site. The major advantage of having this directory on a local disk versus being on a disk that is NFS-mounted across the entire cluster is that the file transfer process is extremely slow when there is a search in progress on the rest of the cluster due to the contention between reads from searches and writes from downloads to the same file system. Our tests show that moving this directory to a local disk increased the overall transfer rate by an order of magnitude, from approximately 100 to 1000 kbytes/sec. The remaining two directories, ~/db_backup, and and ~/db, are available to seqdat via a NFS mount from the file server. The directory ~/db contains the current files that are being used in the searches and are maintained in the NFS-mounted space accessible to all nodes of the cluster. The directory ~/db_backup, also on the core file server, contains the version of the databases from the previous day.

A virtual *std* link pointing to the ~/db directory with the latest NCBI database files indicates the source of the ~/db files to use in any searches. The first task of the *cron* job is to make a copy of this directory, to ~/db_backup, and transfer the *std* link to this copy. Therefore any searches running from this copy that are in progress or that may start during the nightly update will continue accessing the last static copy of the database. After making this copy, the ftp transfer is done using *ncftpget* (http://www.ncftp.com), which compares the existing compressed tar files in ~/db_source with those available for download, proceeding to download a file when they are different. Once all of the updated database files available are downloaded, the remaining task is to unpack the compressed *tar* files into the ~/db directory and move the link back to the newly created database directory. When a user starts a search that may run through a database update, the first step of the job is to make a copy, either on global or local scratch, of the databases that the user will then own. This copy takes just over three hours if the user needs all of the maintained databases for the search. The user's search is then completed on this copy of the database. This allows a user to have a static database for a search that takes multiple days in the event of an update of the NCBI databases during the duration of the run.

39.2.4 Job Parsing, Scheduling and Processing

The search sequences provided by the user are given as a set of input files. Every input files contains a number of individual nucleotide sequences, in FASTA format, each of which need to be compared to the sequences in the databases. The input files are named according to the search to be done, according to the following format: ###X12, where ### is the file index currently ranging from 1 to 164 and X = n, x, t signifies a *blastn, blastx,* or *tblastx* search, respectively. *blastn* is the standard BLAST search in which a nucleotide sequence is compared to sequences in a nucleotide database, *blastx* searches protein databases to find proteins similar to a translated form of the nucleotide query sequence and, finally, *tblastx* compares the translated nucleotide query to translated nucleotide database entries. The *blastx* searches use the nonredundant peptide sequence database, whereas the *blastn* searches against the non-redundant nucleotide, est, sts, gss, and htgs databases as provided by the NCBI. The initial search only uses *blastn* and *blastx,* and is designated as the stage-one search. Currently, this search is done on an approximately weekly basis. For sequences that do not have any hits found during the first stage of similarity searching, a second search needs to be completed. Currently, this is the case for approximately 1800 of the 6500 sequences being analyzed for similarities. This second search is the *tblastx* matching using the nonredundant nucleotide, est, sts, gss, and htgs databases, and is designated as the stage-two search. This search, due to the translation of both the query sequence and the database entries takes a significantly longer time and it is performed only on a monthly basis. For the current input files, the stage-one search takes approximately 30 total node hours, whereas the second-stage search on the 1800 “No hits found” sequences takes over 620 total node hours.

Before starting a new search, a decision has to be made as to whether or not there has been a database update since the last search, so as to not repeat an identical search. As stated above, checks are made for updates to the database available at the NCBI ftp site on a nightly basis; however, updates are not always available. If there are difficulties present at the NCBI FTP site, several days or even longer can pass between updates becoming available. In addition, a decision on the number of nodes that are available for the search needs to be made. A *perl* script was developed to create the necessary PBS script files that distribute the searches among the available nodes in the cluster. This script is based on a file structure in which there is a directory $HOME/search which contains the input files. The $HOME/search directory also contains one additional file, searchlist.in, which is a list of the filenames, one per line, of the files containing sequences on which the stage-one search must be completed. Each of these input files must exist in the $HOME/search directory. The researchers may add new search sequences by either adding them to an existing input file or creating a new input file. In the later case, the filename, which must match the existing convention and have an n or x in the name, must be added to the searchlist.in file.

The output of each of the searches is stored in a new directory inside of $HOME/search. The name of the directory of a given search is based on the date on which the search was started and is of the form yyyy-mm-dd. In each output directory the results of the search are stored, in this case as html files. There is one output file for each input file that is searched. If only stage one is being done in the search, these output files are the entire contents of the output directory. If a stage-two search is also being completed, the input files generated for this search are also stored in the $HOME/search/yyyy-mm-dd directory, along with a directory TBLASTX, which will contain the output of the second stage.

The execution of the *perl* script first checks the date of the current databases, the one to which the *std* link points. This is accomplished by running the *fastacmd-I* on the nonredundant peptide sequence database. This command returns the date of the database, *date_db.* This date is compared to the date of the databases used in the last search, *date_last.* This date is obtained by taking the last of the date directories (comparing the directory names as numbers and looking for the largest), and looking at the date posted in the output of this last search on the 1X12 input file. If *date_db* is not greater than the *date_last,* the user is told that there has been no database update since last search and that they should try again not earlier than the next day as database updates are checked on a nightly basis. The script is then exited. If *date_db* is greater than *date_last,* then the script proceeds. The user is then prompted as to whether he wishes to perform a first-stage search only or a first- and second-stage search. This is the last input required from the user. At this point, the directories under $HOME/search where the output will be stored are created—$HOME/search/yyyy-mm-dd and, if necessary, $HOME/search/yyyy-mm-dd/TBLASTX.

The remainder of the script writes all the necessary PBS script files and submits them for processing. First, the number of nodes available for the search must be determined. This is done by issuing a *showq* command and using *grep* and *cut* to pull out the number of free compute nodes, allowing for the job to be completed with as many nodes as are available in order to minimize the wall time it takes for the search to be completed. This number, #nodes, can range from zero to eight on our current cluster. If zero, the search will default to run on one node. This simplistic mechanism for choosing how many nodes to use will need to be rewritten in the advent of multiple groups using the system. In this case, there will need to be a maximum number of nodes made available to any given job, allowing for the sharing of the resources.

The *perl* script then generates the #nodes scripts, named SCRIPT-1 through SCRIPT-#nodes. The necessary PBS headers are written to each of these script files, along with the necessary environmental variables and links. In addition, the first portion of each of these scripts involves cleaning up the local scratch space from previous searches followed by installing the necessary databases for the current search onto the local scratch system of each node. This is followed by a round-robin process dividing up the searches among the #node script files, with the first input file name in searchlist.in going to SCRIPT-1, the second to SCRIPT-2, and so on, until the last PBS script file is reached, then returning to SCRIPT-1 and repeating the cycle until the end of searchlist.in is reached. Once the PBS script files are generated, they are submitted by the *perl* script. In principle, this method of dividing the workload can lead to significant unbalanced loads on the nodes, but because most of the searches being performed take the same amount of time, this static job distribution introduces only minor load imbalance. Future implementations will use a client–server model in which nodes will be able to request new files for processing from a server node when they finish their scheduled tasks. The server node will then need to maintain an updated database categorizing the individual searches processed, being processed, and waiting to be processed.

In each of these PBS scripts, the stage-one search is processed for a given input file followed by, when requested by the user, creation of the second-stage input files by parsing of the stage-one output to extract the input sequences for which the "No hits found" message is received, and then starting the second-stage search. These input sequences produced in the parsing step are collected in a new second-stage input file with the same naming scheme, but with the x replaced with a t, and the second-stage search, using *tblastx,* is then started. After all requested searches are complete on a given input file, the PBS job continues with the next input file. At any time, the user can monitor the output as well as the status of the batch queues on the system in order to track the progress of the search.

39.2.5 Integration with SmedDb

In order to become useful, the searches produced have to be uploaded and integrated to the SmedDb system. Although the implementation of the search system described above is quite general and can be integrated into any laboratory system, the integration of the search output with the SmedDb database is quite specific, tailored to the needs of the research group. Therefore, this integration is presented as an example that can be used when integrating the search system to other laboratories. SmedDb is a Web-accessible database that contains *S. mediterranea* sequences, BLAST results, reading-frame diagrams, and in situ hybridization images in one output report. Any homologous sequences are displayed with links to NCBI's Entrez and Pubmed.

The architecture of SmedDb as it relates to our cluster search system is given in Figure 39.2. The FASTA files generated from SmedDb containing the sequences (ESTs) that need to be compared with the most recently updated version of NCBI databases are uploaded manually to the cluster search system using secure copy (*scp*). Using the script described above, the required BLAST searches are executed and the output files are downloaded to the SmedDb system also using *scp*. These search results are stored with the corresponding EST as well as being parsed by a *Bioperl* [3] script. The *Bioperl* script is used to pull out the highest-scoring (lowest expectation value) match for each of the searches along with its associated description. This information is then used to update the status of the sequences in SmedDb (having a significant match, no significant match, or no match at all). If a given input sequence has a significant match in the current

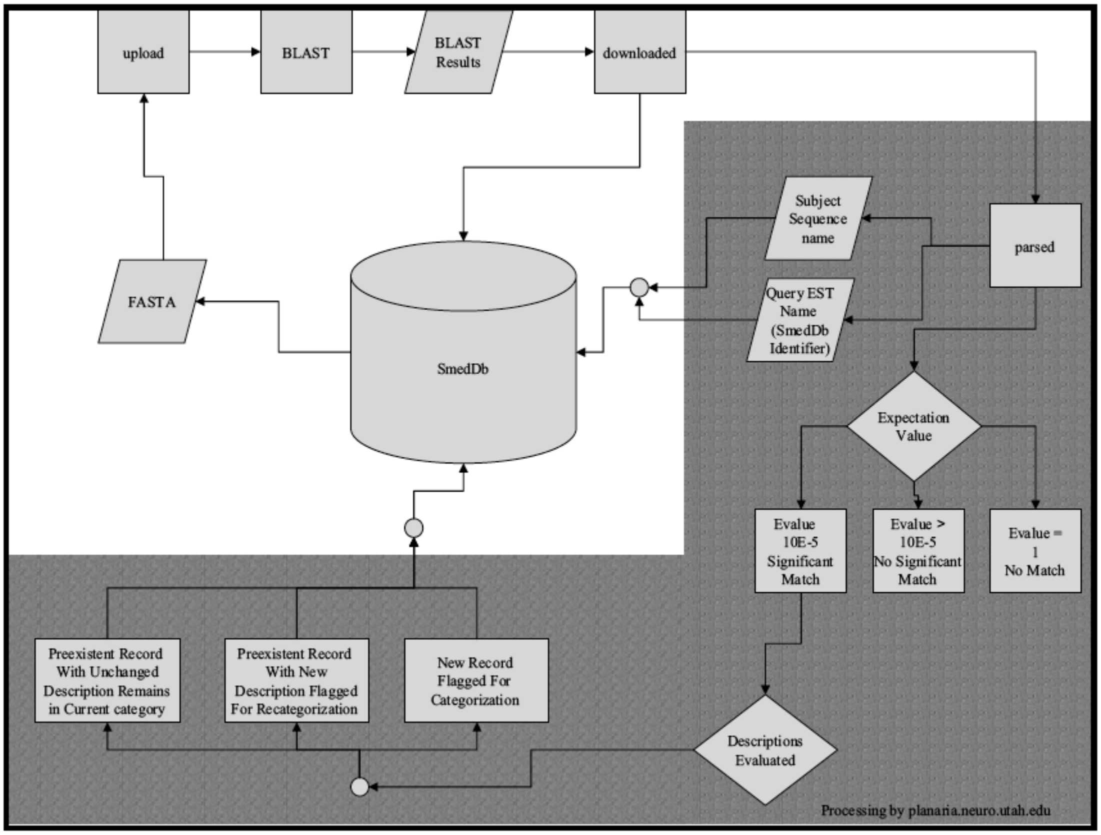

upload
BLAST
BLAST Results
downloaded
Subject Sequence name
parsed
FASTA
SmedDb
Query EST Name (SmedDb Identifier)
Expectation Value
Evalue 10E-5 Significant Match
Evalue > 10E-5 No Significant Match
Evalue = 1 No Match
Preexistent Record With Unchanged Description Remains in Current category
Preexistent Record With New Description Flagged For Recategorization
New Record Flagged For Categorization
Descriptions Evaluated
Processing by planaria.neuro.utah.edu

Figure 39.2 Schema showing the integration of the cluster-based BLAST search engine with the laboratory information system SmedDb.

search, further processing needs to be completed. If previous searches have not found a significant match, the search output is marked for review and possible classification by the researcher. If previous searches have already resulted in finding a significant match, the current result is compared to the old result. If the old and new descriptions are the same, no change is made in the SmedDb for the given sequence entry. If the description is different, this SmedDb record is flagged and a message is posted requesting manual intervention of the researcher to decide on its reclassification.

39.3 PERFORMANCE

The system described above is fully functional and is operating in a production environment. The system as it exists meets the design principles and requirements established in the planning stage. The total cost of the system has been estimated at $26,000, with all of the components being commodity parts.

The nightly database download process takes just over three hours for databases totaling about 15 GBytes. The copy of the necessary database files to local scratch for the project being described is approximately an hour for ~/db needed totaling about 12.5 GBytes. The database updates are performed so as not to interfere with any searches that may be in progress.

As described above, the current search is performed on over 6500 input sequences. Of these, currently about 1800 are candidates for the second-stage search. The time required for a first-stage search only is approximately 30 node hours, whereas a complete first- and second-stage search takes about 620 node hours, or slightly over 3 days if all eight nodes are used. This time is slightly longer than the criteria of most searches being done in 48 hours; however, the majority of searches are stage-one only.

39.4 CONCLUSIONS

This chapter reports a case study on the development of a dedicated commodity-based cluster for the periodic update of gene sequence comparisons. The project has been able to meet the turnaround time goals and eliminate a great deal of human labor in the periodic update of the SmedDb system. Our experience shows that it is possible to use commodity components to design and deploy a cluster with a configuration optimized for a particular task. The judicious use of low-cost hardware combined with a clever update of the databases permits the continuous operation of the system, avoiding interference among the updates and long-running searches. The cluster described here presents a low-cost model for biomedical labs requiring substantially more BLAST searches than can be reasonably performed using the existing NCBI services. As the system is highly scalable, it is possible to use this architecture to deploy systems serving from individual labs to departmental and even institutional BLAST search engines.

ACKNOWLEDGMENTS

This work was partially funded by the National Institutes of Health, National Institute of General Medical Sciences RO-1 GM57260 to ASA, and National Center for Research Resources 1 S10 RR17214-01 to JCF.

REFERENCES

1. R. Sachidanandam, D. Weissman, S. C. Schmidt, J. M. Kakol, L. D. Stein, G. Marth, S. Sherry, J. C. Mullikin, B. J. Mortimore, D. L. Willey, S. E. Hunt, C. G. Cole, P. C. Coggill, C. M. Rice, Z. Ning, J. Rogers, D. R. Bentley, P. Y. Kwok, E. R. Mardis, R. T. Yeh, B. Schultz, L. Cook, R. Davenport, M. Dante, L. Fulton, L. Hillier, R. H. Waterston, J. D. McPherson, B. Gilman, S. Schaffner, W. J. Van Etten, D. Reich, J. Higgins, M. J. Daly, B. Blumenstiel, J. Baldwin, N. Stange-Thomann, M. C. Zody, L. Linton, E. S. Lander, and D. Altshuler, A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms, *Nature, 409,* 928–933, 2001.
2. S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, *Nucleic Acids Research, 25,* 3389–402, 1997.
3. A. Sánchez Alvarado, P. A. Newmark, S. M. Robb, and R. Juste, The *Schmidtea mediterranea* database as a molecular resource for studying platyhelminthes, stem cells and regeneration, *Development, 129,* 5659–5665, 2002.
4. G. F. Pfister, *In Search of Clusters,* Prentice-Hall PTR, Upper Saddle River, NJ, 1998.
5. F. Berman, G. Fox, and T. Hey, *Grid Computing: Making The Global Infrastructure a Reality,* Wiley, 2003.
6. D. B. Jackson, B. Haymore, J. C. Facelli, and Q. O. Snell, Improving Cluster Utilization Through Set Based Allocation Policies, presented at Proceedings of the International Conference on Parallel Computing, Valencia, Spain, 2001.

CHAPTER 3

MAKER: AN EASY-TO-USE ANNOTATION PIPELINE DESIGNED FOR EMERGING MODEL ORGANISM GENOMES[2]

Brandi L. Cantarel, Ian Korf, Sofia M.C. Robb[3], Genis Parra, Eric Ross, Barry Moore, Carson Holt, Alejandro Sánchez Alvarado, and Mark Yandell

[2] This article is reprinted with permission from Cold Spring Harbor Press

[3] My contribution to this manuscript was to assist in formatting the output of MAKER such that it is able to be loaded without any reformatting into the GMOD genome browser, GBrowse. I also wrote the section entitled "Creating SmedGD".

Resource

MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes

Brandi L. Cantarel,[1] Ian Korf,[2] Sofia M.C. Robb,[3] Genis Parra,[2] Eric Ross,[4] Barry Moore,[1] Carson Holt,[1] Alejandro Sánchez Alvarado,[3,4] and Mark Yandell[1,5]

[1]Eccles Institute of Human Genetics, University of Utah, Salt Lake City, Utah 84112, USA; [2]Department of Molecular and Cellular Biology and Genome Center, UC Davis, Davis, California 95616, USA; [3]Department of Neurobiology & Anatomy, University of Utah School of Medicine, Salt Lake City, Utah 84132, USA; [4]Howard Hughes Medical Institute, University of Utah School of Medicine, Salt Lake City, Utah 84132, USA

We have developed a portable and easily configurable genome annotation pipeline called MAKER. Its purpose is to allow investigators to independently annotate eukaryotic genomes and create genome databases. MAKER identifies repeats, aligns ESTs and proteins to a genome, produces ab initio gene predictions, and automatically synthesizes these data into gene annotations having evidence-based quality indices. MAKER is also easily trainable: Outputs of preliminary runs are used to automatically retrain its gene-prediction algorithm, producing higher-quality gene-models on subsequent runs. MAKER's inputs are minimal, and its outputs can be used to create a GMOD database. Its outputs can also be viewed in the Apollo Genome browser; this feature of MAKER provides an easy means to annotate, view, and edit individual contigs and BACs without the overhead of a database. As proof of principle, we have used MAKER to annotate the genome of the planarian *Schmidtea mediterranea* and to create a new genome database, SmedGD. We have also compared MAKER's performance to other published annotation pipelines. Our results demonstrate that MAKER provides a simple and effective means to convert a genome sequence into a community-accessible genome database. MAKER should prove especially useful for emerging model organism genome projects for which extensive bioinformatics resources may not be readily available.

[Supplemental material is available online at www.genome.org.]

Genome annotation, not genome sequencing, is becoming the bottleneck in genomics today. New genomes are being sequenced at a far faster rate than they are being annotated. As of 2007, there are 126 completely sequenced, but unpublished genomes, and the backlog of unpublished and unannotated genomes continues to grow (Liolios et al. 2006). Eukaryotic genomes are particularly at risk as their large size and intron-containing genes make them difficult substrates for annotation. There are currently more than ~800 Eukaryotic genome projects under way (Liolios et al. 2006). Many of them belong to emerging model organisms (http://grants.nih.gov/grants/guide/pa-files/PA-04-135.html), and are represented by relatively small research communities. Annotating these genomes and distributing the results for the benefit of the larger biomedical community is proving difficult for many of these communities, as they often lack bioinformatics experience. One solution to this problem is to outsource the annotation to one of the major annotation databases such as Ensembl (Stabenau et al. 2004) or VectorBase (Lawson et al. 2007). This has proven a fruitful strategy for several groups (c.f. VectorBase), but the numbers of sequenced genomes far exceeds the capacity and the stated purview of these projects; Ensembl, e.g., is restricted to vertebrate genomes and VectorBase to insect vectors of human disease.

In an attempt to ameliorate this problem, many sequencing centers, data repositories, and model organism databases make their annotation software available to the public (http://www.broad.mit.edu/tools/software.html; http://www.tigr.org/software/genefinding.shtml) (Stabenau et al. 2004). However this is not their primary mission, and they usually only make subsets of their internal systems available—and these generally require significant in-house bioinformatics support (Lawson et al. 2007). Thus, despite the best efforts of the bioinformatics community, large numbers of unannotated genomes continue to accumulate, underscoring an urgent need for simpler, more portable annotation pipelines.

Developing an easy-to-use annotation pipeline imposes several design constraints. First, it must be easy to configure and run, requiring minimal bioinformatics and computer resources. In other words, external executables and software need to be minimal, and installation must be routine, even for users with only rudimentary UNIX skills. Second, an easy-to-use pipeline must also provide both a compute and an annotation engine. In practical terms, it must be able to identify repeats, to align ESTs and proteins to the genome, and to automatically synthesize these data into feature-rich gene annotations, including alternative splicing and UTRs, as well as attributes such as evidence trails, and confidence measures. Third, every genome is different and an easy-to-use annotation pipeline must be, therefore, easily configurable and trainable. If not, the evidence gathered by the compute pipeline will be of poor quality, and the annotation process will be compromised.

Another essential feature of an easy-to-use annotation pipeline is that its output formats must be both comprehensive and database ready. This task has been simplified by the Generic Model Organism Database (GMOD) project (http://www.gmod.org), which provides a generic genome database schema and ge-

[5]**Corresponding author.**
E-mail myandell@genetics.utah.edu; fax (801) 585-3214.

Article published online before print. Article and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.6743907.

18:188–196 ©2008 by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/08; www.genome.org

nome visualization tools. GMOD, however, does not provide a means to produce the contents of a database; these must be created by an external annotation pipeline. Therefore, to take advantage of GMOD tools, annotation pipelines must write their outputs in GMOD-compatible Generic Feature Format (GFF3; www.sequenceontology.org/gff3.shtml). However, creating GFF3 files containing all of the information necessary to populate a GMOD database is a complex task. These files must contain descriptions of EST and protein alignments, repeats, and gene predictions. They must also include EST and protein alignments not associated with any annotation, so that false negatives can be identified. Without such data, downstream automated and manual annotation management is seriously compromised.

Finally, to qualify as truly user-friendly, an annotation pipeline should provide an easy means to annotate, view, and edit individual contigs and BACs. This allows users to analyze partial genome assemblies and to independently annotate regions of interest using their own data sets, ideally without the overhead of a database and with only minimal compute resources such as a laptop computer.

We have designed an easy-to-use annotation tool called MAKER in an attempt to meet all of these design criteria. Our goal was to provide emerging genome projects with the means to independently annotate protein-coding genes and to create a GMOD database. MAKER identifies repeats, aligns ESTs and proteins to a genome, makes gene predictions, and integrates these data into protein-coding gene annotations. Moreover, its outputs can be loaded directly into GMOD browsers and databases with no post-processing. As proof of principle, we have used MAKER to annotate the genome of the planarian *Schmidtea mediterranea* and to create a new genome database, SmedGD (http://smedgd.neuro.utah.edu). We have also compared MAKER's performance to other published annotation pipelines as part of the nGASP contest hosted by WormBase (http://www.wormbase.org/wiki/index.php/NGASP). Our results demonstrate that MAKER provides a simple-to-use, yet effective means to annotate an individual contig or BAC or to convert an entire genome sequence into a community-accessible genome database. MAKER is not exhaustive: it does not identify noncoding RNA genes, nor is it intended as a comprehensive solution to every problem in genome annotation. Rather, MAKER is designed to jump-start genomics in emerging model organisms by providing a robust first round of database-ready protein-coding gene annotations.

Results

Benchmarking MAKER on *Caenorhabditis elegans*

In order to obtain a performance benchmark, we ran MAKER on a 10-megabase (Mb) portion of the *C. elegans* genome, as part of the nGASP competition (http://www.wormbase.org/wiki/index.php/NGASP). nGASP provided two annotated 10-Mb regions of the *C. elegans* genome, one for training, and the other for testing. We trained MAKER using the boot-strap procedure outlined in the Methods section and then compared MAKER's performance on the testing region to three other nGASP participants: SNAP (Korf 2004), Augustus (Stanke et al. 2006), and Gramene—an Ensembl-based pipeline (Stabenau et al. 2004) managed by the Gramene group (www.gramene.org). SNAP was run in its ab initio gene prediction mode; Gramene is an evidence-based annotation pipeline that assembles its own computational evidence; and Augustus is a gene-prediction algorithm that can be used to produce either ab initio or evidence-based predictions when provided with an external GFF3 file of EST and protein alignment data. The evidence-based Augustus annotations summarized in Table 1 used GFF3 files of aligned ESTs and proteins provided by nGASP.

Overall, MAKER's performance on the *C. elegans* genome was comparable to that of Gramene and to Augustus when run in the evidence-based mode. All three programs had very similar sensitivity and specificity values for genomic overlap—a measure of the percentage of genes overlapped by an annotation. MAKER's genomic overlap sensitivity (89.81%) was greater than that of Gramene's (88.74%) and less than that of Augustus' (97.05%), indicating that ~90% of annotated *C. elegans* genes were overlapped by at least a portion of a MAKER annotation. MAKER's genomic overlap specificity (91.69%) was also intermediate between those of Augustus (89.47%) and Gramene (93.49%).

When considering the remaining categories in Table 1, it should be kept in mind that these refer to the subset of annotations (32%) that WormBase denoted as complete and confirmed WB160 genes. The low specificities reported for all three programs reflect this fact.

MAKER's weakest performance was in the exon nucleotide accuracy and exon overlap and categories. For all genes, its exon level nucleotide accuracy is 61.82%, Gramene's is 70.8%, and Augustus' is 70.83% and 77.62% (evidence-based). For confirmed

Table 1. MAKER's performance on the *C. elegans* genome

Performance category	Ab initio		Evidence based		
	Snap	Augustus	Maker	Gramene	Augustus
Genomic overlap (gene)					
SP	82.48%	88.09%	91.69%	93.49%	89.47%
SN	95.44%	96.78%	89.81%	88.74%	97.05%
Exon overlap					
SP	18.88%	22.87%	25.58%	27.38%	23.54%
SN	87.63%	93.09%	91.17%	94.84%	96.19%
Exact transcript					
SP	3.92%	7.51%	6.01%	3.52%	8.65%
SN	12.22%	18.64%	14.97%	10.59%	22.20%
Full exact transcript					
SP	0.41%	1.02%	1.91%	0.39%	1.17%
SN	1.22%	2.34%	4.58%	1.02%	2.95%
Exact UTR5					
SP	1.38%	2.27%	4.41%	4.43%	3.38%
SN	5.80%	8.04%	11.20%	9.98%	10.08%
Exact UTR3					
SP	6.40%	9.86%	11.75%	8.05%	11.40%
SN	31.36%	44.20%	40.53%	23.63%	46.03%
Exact all exons					
SP	19.02%	22.08%	22.44%	34.08%	24.19%
SN	93.48%	98.98%	95.62%	91.24%	98.57%
Start stop					
SP	7.05%	12.97%	12.69%	11.87%	17.79%
SN	35.95%	51.83%	47.76%	34.42%	72.51%

SP, specificity; SN, sensitivity. Genomic overlap is based upon all annotations; other categories are for complete, confirmed genes only. Overlap indicates that prediction overlaps reference annotation on the same strand; exact, coordinates of prediction are identical to reference annotation; full exact transcript, all exons match reference annotation coordinates, as do the start and stop codons. Gramene data are from ensembl.gff; Augustus ab initio results are for augustus_cat1v2.gff; Augustus evidence-based results are from augustus_cat3v1.gff. SNAP and MAKER data are from snap.gff, and makerv2_testset.gff, respectively. All data are from files available at http://www.wormbase.org/wiki/index.php/NGASP. WormBase release WB160 was used as the reference. Sensitivity and specificity were calculated using EVAL (Keibler and Brent 2003).

genes, MAKER's exon level overlap specificity is similar to that of the other programs (Table 1), but its sensitivity is still 3.67% less than that of Gramene and 5.02% less than that of Augustus when run in its evidence-based mode. On confirmed genes, MAKER's exact all exon (Table 1) accuracy is similar to those of the other two evidence-based programs. MAKER fell squarely between Gramene and Augustus in correctly annotating entire transcripts (*Exact Transcript*) but outperformed the other two programs when the start and stop of translation is also taken into account (*Full Exact transcript*). MAKER also outperformed the other two programs in accurately identifying 5′ UTRs (*Exact UTR5*). MAKER was more effective at precisely identifying 3′ UTRs (*Exact UTR3*) than was Gramene and was only slightly less accurate than Augustus. The last category in Table 1, *Start Stop*, provides a measure of how well MAKER did at identifying start and stop codons. Once, again, MAKER's performance is comparable to the other programs. MAKER outperformed Gramene in this category, though Augustus was the clear winner. In total then, the data in Table 1 demonstrate that MAKER's overall performance on the *C. elegans* genome is in most instances comparable to that of Gramene and Augustus.

A proof-of-principle collaboration

In order to demonstrate MAKER's suitability as an annotation tool for the genomes of emerging model organisms, we partnered with the *S. mediterranea* genome project to annotate its genome and create a GMOD-based genome database. *S. mediterranea* is a model planarian species, known for its ability to regenerate complete animals from miniscule fragments of its body (Randolph 1897; Morgan 1898). *S. mediterranea* is an emerging model organism for regeneration studies following demonstrations that it is amenable to modern cell (Robb and Sánchez Alvarado 2002), molecular (Sánchez Alvarado et al. 2002), and RNAi (Sánchez Alvarado and Newmark 1999) techniques. Its annotated genome will provide a central resource for the planarian and regenerative medicine research community.

The *S. mediterranea* genome and assembly

The *S. mediterranea* genome was sequenced and assembled by the Washington University Genome Sequencing Center (St. Louis, MO). The final assembly is 902,775,852 nucleotides in length, consistent with Cot and nuclear volume analyses carried out prior to sequencing, which place the *S. mediterranea* genome at ~850 Mb (http://genome.wustl.edu/ancillary/data/whitepapers/Schmidtea_mediterranea_WP.pdf). *S. mediterranea* was sequenced to a depth of ~10×. The assembly's contig length distributions are similar to those of the human and *Drosophila* genomes (data not shown). Its super-contigs, however, are shorter, as technical issues precluded the construction of a BAC library for this organism; thus, no BAC end reads were available during the assembly process; 89.30% of the genome is in super-contigs 10 kb or longer and 44.62% is in super-contigs longer than 50 kb. The final assembly contains 43,673 contigs with a median length of 11,260 bp. The genome has a high AT content (67%).

ESTs

At time of compute, there were 78,101 ESTs from *S. mediterranea*. These were derived from a variety of libraries (see Methods), and consist of both 5′ and 3′ reads. As the *S. mediterranea* EST collection was quite redundant, we collapsed the ESTs into contigs using the CAP3 program (Huang and Madan 1999). This process yielded 15,011 contigs. MAKER aligned 13,026 (88%) of the EST contigs to the genome, using the splice-site aware Exonerate algorithm (Slater and Birney 2005). Of the remainder, about half were not found in the assembly, and low sequence complexity prohibited alignment of the other half. These numbers provide an estimate of 90% for the overall completeness of the assembly, a finding consistent with the experimental estimates of genome size (http://genome.wustl.edu/ancillary/data/whitepapers/Schmidtea_mediterranea_WP.pdf) and the size of the assembly.

S. mediterranea repeats

In total, RepeatMasker (http://repeatmasker.org) flagged 22% of the *S. mediterranea* genome as low-complexity sequence. MAKER also uses BLASTX together with an internal library of transposon and virally encoding proteins to identify mobile-elements (see Architecture of MAKER section). This process masked an additional 4.18% of the genome. Finally, we used Muscle (Edgar 2004) and PILER (Edgar and Myers 2005) to identify additional *S. mediterranea* specific and highly divergent repeats, missed by the previous processes. MAKER used these as a RepeatMasker library. This masked another 1.2% of the genome. In total, 27.4% of the genome was identified as repetitive. The creation of a custom library for use with MAKER is optional but recommended.

The *S. mediterranea* high-confidence gene set

In order to produce a maximally inclusive set of compute data and annotations for our downstream analyses, we ran MAKER over every contig in the *S. mediterranea* genome assembly regardless of size. The resulting data are summarized in Supplemental Table 1. Following procedures similar to those used to annotate other eukaryotic genomes (Rubin et al. 2000; Venter et al. 2001), we next sought to assemble a high-confidence (HC) gene set from among the 65,563 MAKER genes. To do so, we took advantage of the MAKER Quality Indices generated for each transcript, which document the number of exons confirmed by EST, and/or protein evidence (see Methods; Table 2). We included in the HC set every gene having at least one transcript confirmed by an EST alignment with at least one canonical splice site. In total, there were 12,620 genes that met this criteria. We also included in the HC gene set any MAKER annotation with protein homology (BLASTX $E < 1 \times 10^{-6}$) to the Swiss-Prot database (Bairoch and Apweiler 2000); 24,209 MAKER genes met this criterion. We then used RPS-BLAST (http://web.csb.ias.edu/blast/rpsblast.txt) to screen the annotations for Pfam (Bateman et al. 2004) domains ($E < 1 \times 10^{-3}$; minimum coverage >40%). This identified 15,702 domain-containing annotations. We also screened the 128,339 SNAP predictions not overlapping a MAKER annotation for protein homology with SWISS-PROT (Bairoch and Apweiler 2000) and for Pfam (Bateman et al. 2004) domains using the same significance thresholds; 378 of them were homologous to SWISS-PROT proteins, and 1633 had one or more domains. In

Table 2. Maker quality index summary

Length of the 5′ UTR
Fraction of splice sites confirmed by an EST alignment
Fraction of exons that overlap an EST alignment
Fraction of exons that overlap EST or Protein alignments
Fraction of splice sites confirmed by a SNAP prediction
Fraction of exons that overlap a SNAP prediction
Number of exons in the mRNA
Length of the 3′ UTR
Length of the protein sequence produced by the mRNA

total, this gave us a set of 31,955 protein-coding genes supported by combinations of EST, protein, and domain homology.

Protein-coding gene numbers

Our purpose in assembling the HC gene set was to produce a set of gene models suitable for comparison to other annotated eukaryotic genomes. Gene number is one such comparison. Though protein-coding gene numbers have been a subject of controversy, most annotated model Eukaryotes contain on the order of 15,000–25,000 protein-coding genes (for discussion, see Yandell et al. 2005). *Drosophila*, e.g., is believed to contain fewer than 15,000 protein coding genes (Yandell et al. 2005), and the WS160 WormBase release puts the number of *C. elegans* genes at slightly less than 20,000. The latest Ensembl (Stabenau et al. 2004) release of the human genome contains 21,724 known protein-coding genes.

Although there is no a priori reason to assume that *S. mediterranea* might not contain 31,955 protein-coding genes (the number of genes in the HC set), this possibility is not well supported by available experimental evidence. We therefore sought to determine what percentage of the annotations might be split across the short super-contigs characteristic of the *S. mediterranea* genome assembly. To do so, we cloned and sequenced 31 high-molecular-weight *S. mediterranea* mRNAs without recourse to the MAKER annotations. We aligned each mRNA to the genome assembly (30 were found in the assembly) and found that 28 corresponded to MAKER annotations, nine of these (30%) were split across multiple contigs, and four (14%) were annotated as multiple genes on a single *S. mediterranea* contig. By comparison, only 12 of the mRNAs were overlapped by SNAP ab initio predictions, and three of these were split. Though these are small numbers, they suggest that 90.3% of *S. mediterranea* genes correspond to at least one MAKER annotation, 30% of *S. mediterranea* genes are split among multiple contigs, and MAKER has incorrectly split ~14% of *mediterranea* genes into two or more annotations. Taking these percentages as indicative of the genome as a whole would place the *S. mediterranea* protein-coding gene number at 15,570, a number in good agreement with the annotated gene numbers in other model animals.

Evaluating MAKER's performance on *S. mediterranea*

The absence of a large corpus of known *S. mediterranea* genes and mRNAs makes it difficult to assess MAKER's performance by comparison to known *S. mediterranea* gene structures. Instead we have used the protein domains to gain a rough indication of overall annotation completeness and quality. Domain data also provide a measure of how much MAKER's synthesis procedure improved upon SNAP's ab initio predictions for this genome.

We used RPS-BLAST (http://web.csb.ias.edu/blast/rpsblast.txt) and Pfam (Bateman et al. 2004) to identify protein domains in *S. mediterranea* predicted and annotated proteins. In total, 21.54% of MAKER annotations and 38% of HC annotations contain at least one known domain. We used the same procedure to identify domains in the annotated proteomes of other animals and found that 35.5% of *Drosophila melanogaster*, 31.9% of *C. elegans*, and 36.4% human annotated proteins contain known domains. Thus, the percentage of protein domains in the HC set is comparable to those of other annotated animal proteomes. We further categorized the annotations using the Gene Ontology (http://www.geneontology.org) classifications of the domains they encode and used these data to compare the *S. mediterranea* annotations to other annotated animal genomes. These data are shown in Supplemental Table 2 and demonstrate that the HC genes contain an unbiased, diverse, and comprehensive sampling of the *S. mediterranea* proteome.

MAKER improves upon SNAP's ab initio predictions

As a control, we ran a version of SNAP trained for the AT-rich genome of *C. elegans* over the *S. mediterranea* genome: Only 3.49% of those predictions contained domains, whereas when trained for *S. mediterranea* using the universal gene-based procedure (for details, see Methods), 5.17% of SNAP ab initio predictions contained domains. By comparison, 21.54% of MAKER *S. mediterranea* annotations and 38% of HC annotations contain at least one domain (Supplemental Table 1). Of the 128,339 ab initio SNAP predictions not overlapping MAKER annotations, only 1.27% contained domains.

In contrast to *S. mediterranea*, MAKER's training and synthesis procedures produced only modest increases in gene-level specificity for *C. elegans*, with most of the improvements to accuracy coming from refinements to transcript structures. In *C. elegans*, MAKER's overall genomic overlap and exon overlap accuracies were similar to SNAP's (Table 1). Real gains, however, were observed in the other categories. MAKER's exact transcript sensitivity was 6.01% compared with SNAP's 3.92%; exact transcript specificity also showed gains, rising from SNAP's 12.22% to 14.97%. Likewise, full exact transcript sensitivity and specificity increased by a factor of four. The synthesis process also improved the accuracy of UTR annotation; in fact, Table 1 somewhat misrepresents the nature of the improvement, as EVAL (Keibler and Brent 2003) considers stop codons to be part of the 3′ UTR; likewise, it also considers an incomplete codon preceding the annotated translation to be 5′ UTR. Excluding UTRs of less than four nucleotides in length, the Exact UTR5 and UTR3 values are 0% for SNAP. MAKER's synthesis procedures also improved upon SNAP's ability to correctly identify start and stop codons; for these features, specificity rose from 7.05% to 12.69% and sensitivity from 35.95% to 47.76%.

Improvements were greater for the emerging genome

MAKER's synthesis procedure resulted in a far greater enrichment for protein-domain containing annotations in *S. mediterranea* than it did in *C. elegans*. In total, only 5.17% of *S. mediterranea* ab initio SNAP predictions encode proteins with domains, whereas 21.54% of the MAKER annotations do. In *C. elegans*, by comparison, 38.65% of ab initio SNAP predictions and 44.81% of MAKER annotations have domains. Hence, the enrichment (16%) of MAKER annotations compared with SNAP predictions for domains in *C. elegans* is modest compared with 315% enrichment seen in *S. mediterranea*. The difference appears to be due primarily to the lower specificity of SNAP on the *S. mediterranea* compared with the *C. elegans* genome. Three lines of evidence support this conclusion. First, the ratio of SNAP to MAKER transcripts is much higher in *S. mediterranea* than *C. elegans*, 2.5× compared with 1.3×, respectively. Second, there is a greater correspondence between SNAP predictions and MAKER annotations in *C. elegans* than there is in *S. mediterranea*—only 22.39% of SNAP predictions do not overlap a MAKER annotation in *C. elegans*, whereas 76% fail to do so in *S. mediterranea*. Third, of the SNAP predictions not overlapping an *S. mediterranea* MAKER annotation, only 1.27% contain domains (RPS blast to PFAM $E < 1 \times 10^{-3}$; percent coverage > 40%), again suggesting that many are false positives. Despite these differences, 38% of genes in the *S. mediterranea* HC gene-set encode protein domains, a value quite similar to the 32% in the WB160 release of the *C. elegans* proteome. These facts demonstrate that the ability of MAKER to screen and improve upon SNAP's ab initio predictions is greatest in emerging model organisms such as *S. mediterranea* for which there is limited training data.

SmedGD

We used the GFF3 output from MAKER to jump-start SmedGD, a publicly available resource for the planarian and regeneration research communities. SmedGD houses the *S. mediterranea* genome assembly, its MAKER annotations, and their associated computational evidence (Robb et al. 2007). Because SmedGD conforms to GMOD specifications (http://www.gmod.org), its contents can be queried and viewed over the Web using GMOD tools such as GBrowse (http://www.gmod.org). The use of GMOD schemas ensures interoperability of database contents, allowing them to be shared and compared with the contents of other GMOD databases such as FlyBase, WormBase, and SGD. Figure 1 shows a screen shot from SmedGD, showing a MAKER annotation and its accompanying compute data. SmedGD is located at http://smedgd.neuro.utah.edu.

Discussion

We used MAKER on the genomes of both an established and an emerging model organism. Our results for the *C. elegans* genome demonstrate that the accuracy of MAKER on a model organism genome is comparable to that of other annotation pipelines, whereas our work on the *S. mediterranea* genome shows that MAKER provides an effective means to annotate an emerging genome and to create a genome database.

MAKER's performance on established genomes

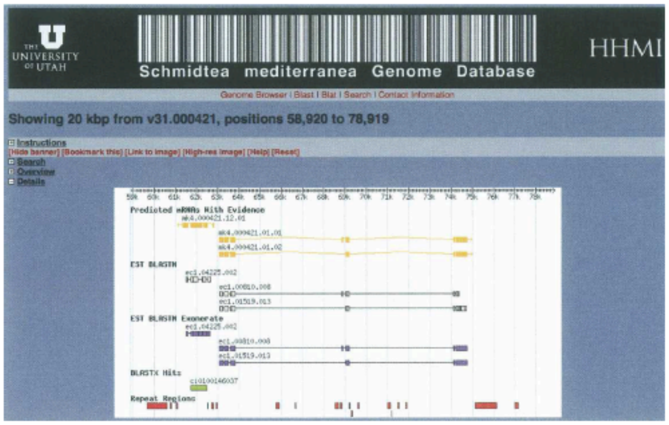
We compared MAKER's sensitivity and specificity to those of Augustus (Stanke et al. 2006) and Gramene (www.gramene.org)—an Ensembl-based pipeline (Stabeneau et al. 2004) in eight different categories using the EVAL program (see Table 1) (Keibler and Brent 2003). Two categories, *genomic overlap* and *exon overlap*, give an indication of what percentage of annotated *C. elegans* genes were overlapped by a MAKER annotation. MAKER's performance in the first category was similar to the other two programs; its accuracy was 90.75% compared with 91.12% and 93.26% for Gramene and Augustus, respectively. However, MAKER faired worse in the Exon overlap category, exhibiting a slight tendency to drop exons, resulting in a 3.67% and 5.02% underperformance for sensitivity in this category relative to Gramene and Augustus, respectively. MAKER was the most effective of the three annotation pipelines at calling entire transcripts, including start and stop codons (*full exact transcript*; Table 1). Its performance in the remaining categories was comparable to Gramene and Augustus.

MAKER's performance on emerging genomes

Emerging genomes place particular demands on annotation pipelines. The differences in the *C. elegans* and *S. mediterranea* annotations illustrate many of the challenges unique to annotating an emerging genome. Our results make it clear that good performance on an established genome is no guarantee of similar performance on an emerging genome. Poor ab initio gene-finder performance—even when retrained—makes an evidence-based process to inform and filter gene predictions absolutely crucial. As judged by domain content, MAKER was only able to improve upon SNAP's *C. elegans* ab initio predictions by 16%, whereas in *S. mediterranea* MAKER's annotations are enriched 315% for protein domains compared with the SNAP ab initio predictions. The differences are due to ab initio gene finder performance. In *C. elegans,* the ab initio predictors did well, and the evidence assembled by the compute pipeline usually did little more than confirm their structure. In *S. mediterranea*, the situation was very different, and MAKER's synthesis procedure played a much greater role. These facts demonstrate the necessity of a trainable, evidence-based process to inform and filter gene predictions when annotating emerging genomes; MAKER's quality indices proved instrumental in this regard, both for training and for assembling the HC gene set.

MAKER outputs are GMOD and Apollo compliant

The *S. mediterranea* genome project used MAKER's GFF3 outputs to create SmedGD, a GMOD database (http://www.gmod.org) for the *S. mediterranea* genome. SmedGD is available at http://smedgd.neuro.utah.edu and is intended to provide a basic resource for planarian functional and comparative genomics. In total, less than 60 d were required to convert the *S. mediterranea* genome assembly into a genome database. SmedGD thus demonstrates the power of MAKER to jump-start genomics in emerging model organisms by providing a first round of database-ready, protein-coding gene annotations.



Schmidtea mediterranea Genome Database

Showing 20 kbp from v31.000421, positions 58,920 to 78,919

Figure 1. SmedGD, the GMOD-based *S. mediterranea* genome database constructed directly from MAKER's outputs (http://smedgd.neuro.utah.edu).

MAKER is ideal for smaller projects

MAKER can also be used to annotate individual contigs and BACs. For *S. mediterranea*, MAKER ran on a single-core MAC laptop (2 GHz CPU with 2 GB RAM) at a rate of 4.1 h/Mb of sequence; this means that a 100 KB BAC can be annotated on a laptop computer in less than half an hour. Furthermore, the outputs can be immediately viewed and edited in Apollo (see Fig. 3) (Lewis et al. 2002) without the added overhead of a genome database. These features make MAKER ideal for small-scale applications and will prove useful for researchers working in emerging model organisms for which only partial assemblies are available.

Future improvements to MAKER

At present, MAKER uses only a single ab initio gene predictor and creates only protein-coding annotations. MAKER's modular structure means that any gene predictor can be integrated into its architecture with minimal software development. To date, we have focused on integrating SNAP, as it was designed with easy trainability in mind (Korf 2004), but additional predictors could be integrated as well. Augustus is also trainable and comes with an optimization script that tries to find values for the metaparameters, such as splice window sizes (Stanke and Morgenstern 2005). MAKER should be able to manufacture this information automatically as part of an extended training procedure, and we are currently exploring the feasibility of doing so. Extending MAKER to produce ncRNA annotations is another area of development. Tools for tRNA gene prediction exist (Lowe and Eddy 1997), as do ncRNA gene-finders (Holmes 2005; Rivas and Eddy 2001). These improvements will make for more complete genome databases and help end the annotation bottleneck.

Methods

Architecture of MAKER

MAKER has a modular architecture that abstracts sequence analyses in a standardized object model. MAKER uses the CGL (Yandell et al. 2006) common object model, which extends the Bioperl (http://www.bioperl.org) GenericHit and GenericHSP classes with methods that facilitate comparative analyses and automatic annotation. MAKER's modular construction allows it to break the annotation process down into a series of five discrete activities that are easily interoperable: *compute*, *filter/cluster*, *polish*, *synthesis*, and *annotate* (Fig. 2). MAKER performs these actions on sequences of any length by automatically cutting the input sequence into series of chunks (default is 100 kb), running each compute, and then merging the results.

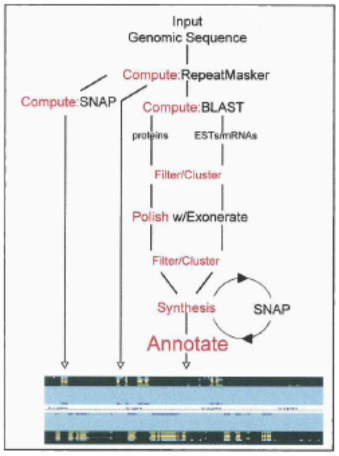


Figure 2. MAKER Overview. MAKER uses four external executables: RepeatMasker, BLAST, SNAP, and Exonerate. Actions corresponding to the five basic steps of automatic annotation are shown in red.

Step 1: Compute phase

A battery of sequence analysis programs is run on input genomic sequence. The purpose of these computes is to identify and Mask repeats and to assemble protein EST and mRNA alignments that will be used to inform MAKER's gene-annotation process, which is outlined in steps 4 and 5 below. The default MAKER configuration uses four external programs: RepeatMasker (http://repeatmasker.org), BLAST (Altschul et al. 1990; Korf et al. 2003), Exonerate (Slater and Birney 2005), and SNAP (Korf 2004). Each is publicly available and free for academic use. All four programs are also easy to install and run on UNIX, Linux, and OS X.

Unless repeats are effectively masked, gene predictions and gene annotations will contain portions of transposons and viruses. MAKER uses a two-tier process to avoid this problem. First, RepeatMasker is used to screen the genome for low-complexity repeats; these are then "soft-masked," e.g., transformed to lowercase letters rather than to Ns. Soft masking excludes these regions from nucleating BLAST alignments (Korf et al. 2003) but leaves them available for inclusion in annotations, as many protein-coding genes contain runs of low complexity sequence. MAKER also uses BLASTX together with an internal library of transposon and virally encoding proteins to identify mobile-elements. This process has been shown to substantially improve repeat masking as it identifies genome regions that are distantly related to the protein coding portions of transposons and viruses; these tend to be missed by RepeatMasker's nucleotide-based alignment process, even when genome specific repeat libraries are available (Smith et al. 2007). Repeat regions identified in this process are masked to Ns. MAKER performs all of the actions automatically.

BLAST is used throughout the compute phase, first for repeat identification with RepeatMasker (as described above) and then to identify EST, mRNAs, and proteins with significant similarity to the input genomic sequence. Because BLAST does not take splice sites into account, its alignments are only rough approximations. MAKER therefore uses Exonerate (Slater and Birney 2005), a splice-site aware alignment algorithm to realign, or polish, sequences following filtering and clustering (see steps 2 and 3, below). Exonerate's ability to align both protein and nucleotide sequences to the genome make it an economical choice for this task.

Step 2: Filter/cluster

Filtering consists of identifying and removing marginal predictions and sequence alignments on the basis of scores, percent identities, etc. Filtering criteria for each external executable are set by modifying the text-based maker_bopts.*ctl* file (see configuration README distributed with MAKER). New users are not expected to edit this file, but advanced users may do so to change the behavior of the program. After filtering, the remaining data are then clustered against the genomic sequence to identify overlapping alignments and predictions. Clustering has two purposes. First, it groups diverse computational results into a single cluster of data, all of which support the same gene or transcript. Second, it identifies redundant evidence. For example, highly expressed genes may be supported by hundreds if not thousands of identical ESTs. Clustering criteria are set in the maker_bopts.*ctl* file, which instructs MAKER to keep some maximum number of members within each cluster, sorted on some series of filtering attributes such as score or fraction of the hit-sequence aligned. The default parameters are appropriate for most applications but can be easily modified.

Step 3: Polish

This step realigns BLAST hits using a second alignment algorithm to obtain greater precision at exon boundaries. MAKER uses Exonerate (Slater and Birney 2005) to realign matching and highly similar ESTs, mRNAs, and proteins to the genomic input sequence. Because Exonerate takes splice-sites into account when generating its alignments, they provide MAKER with information about splice donors and acceptors. This information is especially useful in the synthesis and annotation steps (see below). The thresholds in the maker_bopts.*ctl* file earmark BLAST hits for polishing and are suitable for most applications but can be easily altered if desired (see configuration README distributed with MAKER).

Step 4: Synthesis

MAKER synthesizes information from the polished and clustered EST and protein alignments to produce evidence for annotations. To do so, it identifies ESTs that it judges correspond to the same alternatively splice transcript. This is accomplished by comparing the coordinates of each polished sequence alignment on the genomic sequence in the same way that a human annotator might, e.g., by looking for internal exons with differing boundaries. Next MAKER identifies those protein alignments whose coordinates are consistent with each of the EST splice forms. Once a set of EST and protein alignments—all consistent with the same spliced transcript—has been identified, positions on the genomic input sequence upstream and downstream of the alignments are labeled as possible intergenic regions. Those bases on the genomic sequence that fall between exons are labeled as putative introns, and bases overlapping the protein alignments are labeled as putative translated sequence. MAKER then calculates a score for each of these nucleotides on the query sequence based upon the percentage of similarity of the alignment, type of alignment, and a query nucleotide's position within the alignment. These scores together with their putative sequence types, e.g., Intergenic, Coding, Intron, and UTR, are then passed to SNAP. Based upon this information, SNAP then modifies its internal Hidden Markov Model (HMM). In the absence of any supporting EST or protein alignments, MAKER uses SNAP's ab initio prediction (for additional details, see Training SNAP).

Step 5: Annotate

MAKER also post-processes the synthesis-generated SNAP predictions and recombines them with evidence to generate complete annotations. Each synthesis-generated SNAP prediction is checked against all ESTs and mRNAs, and 5′ and 3′ UTRs consistent with the prediction are identified based upon their coordinates relative to the predicted coding exons. The coordinates of the SNAP prediction are then altered to include these regions. This process is repeated for each of the synthesis-based predictions. Finally, compute evidence supporting each exon is added, and alternatively spliced forms are documented.

Additional details regarding MAKER's architecture and implementation can be found in the release materials. All MAKER source code is publicly available; the current release along with installation instructions and documentation is available at http://www.yandell-lab.org/maker.

Inputs and outputs

The input to MAKER is a genomic sequence (of any length) in fasta format and three configuration files describing external executable, sequence database locations, and various compute parameters (see configuration README distributed with MAKER). MAKER also uses four sequence database files during the compute phase: a *transposons* file, an optional *repeatmasker database* file, a *proteins* file, and an *ESTs/mRNAs* file. Each file is in fasta format. The *transposons* file is bundled with MAKER and contains a selection of known transposon and virally encoded protein sequences. This file is used to identify and mask repeats missed by RepeatMasker, as this has been shown to substantially improve accuracy (Smith et al. 2007). In cases where no organism-specific repeat library is available, MAKER will automatically use the *transposon* file to mask mobile elements and the RepeatMasker program to identify and mask low-complexity sequences. The *repeatmasker* file is an optional fasta file containing organism specific repeat sequences, if available. The *proteins* file contains any proteins users would like aligned to the genome. We recommend they use the latest version of the SWISS-PROT database for this purpose (Bairoch and Apweiler 2000). Finally, users should also supply a file of ESTs and/or mRNAs sequences derived from the organism being annotated. Assembling these into contigs is helpful, but it is not required.

MAKER outputs GMOD-compliant annotations in GFF3 format (http://www.sequenceontology.org/gff3.shtml) containing alternatively spliced transcripts, UTRs, and evidence for each gene's annotated transcript and protein sequences. This file can be directly imported into genome browsers and databases that adhere to Sequence Ontology (Eilbeck et al. 2005) and GMOD (http://www.gmod.org) standards. For convenience, MAKER also outputs multifasta files of transcripts and protein sequences for both annotations and ab initio SNAP predictions.

MAKER also writes a GAME XML file (http://www.fruitfly.org/annot/apollo/game.rng.txt) containing the same contents as the corresponding GFF3 file (http://www.sequenceontology.org/gff3.shtml); this file can be directly viewed in the Apollo genome browser (Figure 3) (Lewis et al. 2002). Apollo can also be used to directly edit annotations and to save them to GFF3 format, thus changes to MAKER annotations can be saved prior to uploading them into a GMOD browser or database. Apollo can also directly export the revised transcripts and protein sequences in fasta format. This is an especially useful feature for those seeking to annotate a single contig or BAC rather than an entire genome, as it circumvents the overhead associated with creating and maintaining a GMOD database. Figure 3 shows a portion of an annotated contig viewed in Apollo genome browser. Compute evidence assembled by MAKER is shown in the top panel; its resulting annotation, below. This figure demonstrates how MAKER synthesizes data gathered by its compute pipeline into evidence-

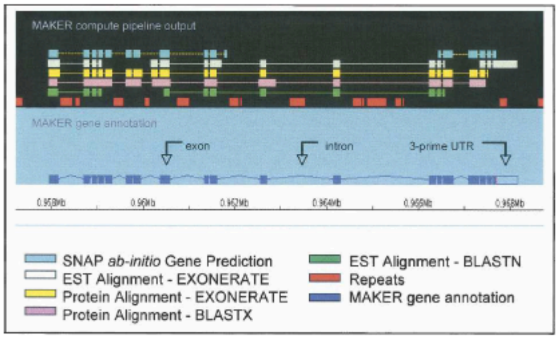

MAKER compute pipeline output
MAKER gene annotation
exon
intron
3-prime UTR
0.958Mb
0.96Mb
0.962Mb
0.964Mb
0.966Mb
0.968Mb
SNAP ab-initio Gene Prediction
EST Alignment - EXONERATE
Protein Alignment - EXONERATE
Protein Alignment - BLASTX
EST Alignment - BLASTN
Repeats
MAKER gene annotation

Figure 3. Apollo view of a MAKER gene annotation and its associated evidence. Evidence gathered by MAKER's compute pipeline (*upper* panel) is synthesized into the resulting MAKER annotation (*lower* panel).

informed gene annotations; while SNAP produced *two* ab initio predictions in this region, the EST and protein alignments clearly support a single gene. Note too the 3′ UTR on the MAKER annotation derived from the EST alignments.

The MAKER mRNA quality index

Compute data are essential for discriminating real genes from false positives. To simplify the quality evaluation process, each MAKER-annotated transcript has an associated quality index included in its GFF3 and GAME XML outputs. This is a nine-dimensional summary (Table 2) of a transcript's key features and how they are supported by the data gathered by MAKER's compute pipeline. The quality index associated with the mRNA shown in Figure 3 is QI:0|0.77|0.68|1|0.77|0.78|19|462|824.

Quality indices play a central role in training MAKER for a particular genome, where they are used to identify transcripts that are well supported by EST and protein evidence but poorly supported by ab initio SNAP predictions. These cases are used to retrain SNAP via the bootstrap procedure outlined below. MAKER's quality indices also provide an easy means to sort and rank transcripts by key features such as number of exons, presence or absence of UTR, or degree of computational support. Quality indices were used to assemble the HC *S. mediterranea* genes described in the Results section.

Training MAKER

For optimal accuracy, a gene finder must be trained for a specific genome (Korf 2004), generally using several hundred existing gene-annotations drawn from a body of experimental data gathered over many years. Unfortunately, many emerging genomes do not have a history of experimental molecular biology. It has therefore become a common practice to use gene finders trained in one genome to predict genes in another—a far from optimal solution to the problem (for discussion, see Korf 2004). Information gathered from ab initio predictions is essential for the annotation process, even when other evidence is available. Moreover, in the absence of experimental evidence and sequence similarities, the probabilistic models produced by ab initio gene prediction programs offer the best guesses at gene structure. The SNAP (Korf 2004) gene finder was designed from the outset to be easily configured for any genome; hence its use in MAKER.

MAKER is trained for a genome using a two-step process. First, SNAP is trained by aligning a set of universal genes to the input genome (Parra et al. 2007). These universal genes are highly conserved in all eukaryotes and can be identified using pairwise and profile-HMM alignment methods. The resulting gene structures are used to create a first-pass version of SNAP for use in the next stage of the training process. This initial stage of the training procedure is automated, and complete details of the process can be found in the MAKER README. More extensive documentation is provided by Parra et al. (2007).

The genome-specific HMM produced in the first stage of SNAP training is further refined with a second stage of training. This is accomplished by running MAKER on a few megabases of genomic sequence (enough to result in a few hundred annotations). The resulting GFF3 outputs are then used as inputs to a script called maker2zff.pl, whose output is a ZFF file that can be used to automatically create a revised HMM. The maker2zff.pl script uses the quality index MAKER attaches to each annotation to identify a set of gene models with intron-exon structures that are unambiguously supported by EST alignments and protein homology. These genes are then used to further refine the SNAP HMM. The maker2fzff.pl script is bundled with MAKER, and programs necessary to create the HMM are included in the SNAP package. To train MAKER for the *S. mediterranea* genome, we first trained SNAP using the universal gene set as outlined above. We then ran MAKER on a randomly selected 100-Mb portion of the *S. mediterranea* genome (~10% of the entire genome). The resulting GFF3 files were used as inputs to maker2zff.pl, and the refined SNAP-HMM was used in the final annotation run.

Manufacturing an *S. mediterranea* specific RepeatMasker database

Repeat sequences were identified for the *S. mediterranea* genome by two methods. First, RepeatRunner (Smith et al. 2007) identified and masked sequences that had similarity to previously identified repeated elements. Second, PILER-DF (Edgar and Myers 2005) was used to find novel dispersed repeats. Settings for the various programs in the PILER suite are as follows: PALS was run with the parameter length = 150 (minimum hit length) and pctid = 94 (minimum percentage identity). PILER was run with the parameter famsize = 10 (minimum size of the repeat family). MUSCLE (Edgar 2004) was run with maxiters = 1 and diags = 1 as recommended in the documentation for PILER. There were 295 repeat families found by this method; most were helitrons (Kapitonov and Jurka 2001).

Manufacturing EST contigs from *S. mediterranea* ESTs

The 78,101 EST sequences from *S. mediterranea* were clustered into 15,011 contigs using CAP3 (Huang and Madan 1999).

Manufacturing the protein database

The reference *proteins* file consisted of proteome sequences from seven organisms and all known Platyhelminthes proteins. The *C. elegans (W160)*, *D. melanogaster (v4.3)*, *Escherichia coli (NC_000913)*, *Homo sapiens (v36.1)*, *Mus musculus (v36.1)*, and *Saccharomyces cerevisiae* (08/2006) proteome sequences were downloaded from NCBI (http://ftp.ncbi.nih.gov/genomes). The *Ciona intestinalis* proteome (v1.0) was downloaded from the Joint Genome Institute downloads site (http://genome.jgi-psf.org/ciona4/ciona4.home.html). Platyhelminthes protein sequences were downloaded from NCBI's Entrez in August 2006.

Compute times

We clocked MAKER on a 2.236-Mb sequence. On a 32 GB-RAM machine, with eight dual-core 2-GHz processors, the annotation took MAKER 549 min on one processor and 299 min using two processors. When external programs, such as BLAST are pre-run, the process time for MAKER on one processor was 31.33 min. For this test, MAKER produced a 3.7-Mb GFF3 file, a 60 MB GAME XML document, and four fasta files totaling 560 kilobytes. For *S. mediterranea*, MAKER ran on a single-core MAC laptop (2 GHz CPU with 2 GB RAM) at a rate of 4.1 h/Mb of sequence—this value includes all compute steps, e.g., compute phase, filter/cluster, polish, synthesis, and annotate.

Downloading and installing MAKER

MAKER is available for download from http://www.yandell-lab.org/downloads/maker/maker.tar.gz. Once downloaded, the MAKER package should be unzipped and untared. Full installation and usage instructions are available in the file called README.

Creating SmedGD

The GFF3 output files generated by MAKER were used to populate SmedGD. The files were uploaded into a mySQL database, using a standard Bioperl (http://www.bioperl.org) loading script, bp_seqfeature_load.pl. This script converts GFF3 formatted annotations to Bio::SeqFeatureI objects, which are stored in the mySQL database. GBrowse, a tool distributed by GMOD (http://www.gmod.org) implementing a Bio::DB::SeqFeature::Store database adaptor, accesses and displays rows of data or tracks that are mapped to specific locations in the genome. SmedGD consists of MAKER annotations as well as project specific features, such as additional protein homology, human curated genes, and RNA interference phenotype data. The database is available at http://smedgd.neuro.utah.edu.

Acknowledgments

We thank Bret Pearson for the cloning of the 31 *S. mediterranea* mRNAs discussed in the Results section. This work was supported in part by NIH grant K22-HG0064 to I.K. S.M.C.R. is supported by NIH Genetics Training Grant 5 T32 GM007464. A.S.A. is a Howard Hughes Medical Institute investigator.

References

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. 1990. Basic local alignment search tool. *J. Mol. Biol.* **215:** 403–410.

Bairoch, A. and Apweiler, R. 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.* **28:** 45–48. doi: 10.1093/nar/28.1.45.

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., et al. 2004. The Pfam protein families database. *Nucleic Acids Res.* **32:** D138–D141. doi: 10.1093/nar/gkh121.

Edgar, R.C. 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32:** 1792–1797. doi: 10.1093/nar/gkh340.

Edgar, R.C. and Myers, E.W. 2005. PILER: Identification and classification of genomic repeats. *Bioinformatics* **21:** i152–i158.

Eilbeck, K., Lewis, S.E., Mungall, C.J., Yandell, M., Stein, L., Durbin, R., and Ashburner, M. 2005. The Sequence Ontology: A tool for the unification of genome annotations. *Genome Biol.* **6:** R44. doi: 10.1186/gb-2005-6-5-r44.

Holmes, I. 2005. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics* **6:** 73. doi: 10.1186/1471-2105-6-73.

Huang, X. and Madan, A. 1999. CAP3: A DNA sequence assembly program. *Genome Res.* **9:** 868–877.

Kapitonov, V.V. and Jurka, J. 2001. Rolling-circle transposons in eukaryotes. *Proc. Natl. Acad. Sci.* **98:** 8714–8719.

Keibler, E. and Brent, M.R. 2003. Eval: A software package for analysis of genome annotations. *BMC Bioinformatics* **4:** 50. doi: 10.1186/1471-2105-4-50.

Korf, I. 2004. Gene finding in novel genomes. *BMC Bioinformatics* **5:** 59. doi: 10.1186/1471-2105-5-59.

Korf, I., Yandell, M., and Bedell, M. 2003. *BLAST: An essential guide to the basic local alignment search tool.* O'Reilly & Associates, Inc., Sebastopol, CA.

Lawson, D., Arensburger, P., Atkinson, P., Besansky, N.J., Bruggner, R.V., Butler, R., Campbell, K.S., Christophides, G.K., Christley, S., Dialynas, E., et al. 2007. VectorBase: a home for invertebrate vectors of human pathogens. *Nucleic Acids Res.* **35:** D503–D505.

Lewis, S.E., Searle, S.M., Harris, N., Gibson, M., Lyer, V., Richter, J., Wiel, C., Bayraktaroglir, L., Birney, E., Crosby, M.A., et al. 2002. Apollo: A sequence annotation editor. *Genome Biol.* **3:** doi: 10.1186/gb-2002-3-12-research0082.

Liolios, K., Tavernarakis, N., Hugenholtz, P., and Kyrpides, N.C. 2006. The Genomes On Line Database (GOLD) v.2: A monitor of genome projects worldwide. *Nucleic Acids Res.* **34:** D332–D334. doi: 10.1093/nar/gkj145.

Lowe, T.M. and Eddy, S.R. 1997. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.* **25:** 955–964.

Morgan, T.H. 1898. Experimental studies of the regeneration of *Planaria maculata*. *Arch. Entw. Mech. Org.* **7:** 364–397.

Parra, G., Bradnam, K., and Korf, I. 2007. CEGMA: A pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics* **23:** 1061–1067.

Randolph, H. 1897. Observations and experiments on regeneration in planarians. *Arch. Entw. Mech. Org.* **7:** 352–372.

Rivas, E. and Eddy, S.R. 2001. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* **2:** 8. doi: 10.1186/1471-2105-2-8.

Robb, S.M. and Sánchez Alvarado, A. 2002. Identification of immunological reagents for use in the study of freshwater planarians by means of whole-mount immunofluorescence and confocal microscopy. *Genesis* **32:** 293–298.

Robb, S.M., Ross, E., and Sánchez Alvarado, A. 2007. SmedGD: The *Schmidtea mediterranea* genome database. *Nucleic Acids Res.* doi: 10.1093/nar/gkm684.

Rubin, G.M., Yandell, M.D., Wortman, J.R., Gabor Miklos, G.L., Nelson, C.R., Hariharan, I.K., Fortini, M.E., Li, P.W., Apweiler, R., Fleischmann, W., et al. 2000. Comparative genomics of the eukaryotes. *Science* **287:** 2204–2215.

Sánchez Alvarado, A. and Newmark, P.A. 1999. Double-stranded RNA specifically disrupts gene expression during planarian regeneration. *Proc. Natl. Acad. Sci.* **96:** 5049–5054.

Sánchez Alvarado, A., Newmark, P.A., Robb, S.M., and Juste, R. 2002. The *Schmidtea mediterranea* database as a molecular resource for studying platyhelminthes, stem cells and regeneration. *Development* **129:** 5659–5665.

Slater, G.S. and Birney, E. 2005. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* **6:** 31. doi: 10.1186/1471-2105-6-31.

Smith, C.D., Edgar, R.C., Yandell, M.D., Smith, D.R., Celniker, S.E., Myers, E.W., and Karpen, G.H. 2007. Improved repeat identification and masking in *Dipterans*. *Gene* **389:** 1–9.

Stabenau, A., McVicker, G., Melsopp, C., Proctor, G., Clamp, M., and Birney, E. 2004. The Ensembl core software libraries. *Genome Res.* **14:** 929–933.

Stanke, M. and Morgenstern, B. 2005. AUGUSTUS: A web server for gene prediction in eukaryotes that allows user-defined constraints. *Nucleic Acids Res.* **33:** W465–W467. doi: 10.1093/nar/gki458.

Stanke, M., Tzvetkova, A., and Morgenstern, B. 2006. AUGUSTUS at EGASP: using EST, protein and genomic alignments for improved gene prediction in the human genome. *Genome Biol.* **7:** S11–S18.

Venter, J.C.M.D., Adams, E.W., Myers, P.W., Li, R.J., Mural, G.G., Sutton, H.O., Smith, M., Yandell, C.A., Evans, R.A., Holt, J.D., et al. 2001. The sequence of the human genome. *Science* **291:** 1304–1351.

Yandell, M., Bailey, A.M., Misra, S., Shu, S., Wiel, C., Evans-Holm, M., Celniker, S.E., and Rubin, G.M. 2005. A computational and experimental approach to validating annotations and gene predictions in the *Drosophila melanogaster* genome. *Proc. Natl. Acad. Sci.* **102:** 1566–1571.

Yandell, M., Mungall, C.J., Smith, C., Prochnik, S., Kaminker, J., Hartzell, G., Lewis, S., and Rubin, G.M. 2006. Large-scale trends in the evolution of gene structures within 11 animal genomes. *PLoS Comput. Biol.* **2:** e15. doi: 10.1371/journal.pcbi.0020015.

Received May 25, 2007; accepted in revised form September 18, 2007.

CHAPTER 4

SMEDGD: THE *SCHMIDTEA MEDITERRANEA* GENOME DATABASE[4]

Sofia M.C. Robb, Eric Ross and Alejandro Sánchez Alvarado

[4] Reprinted with permission from Oxford University Press

Nucleic Acids Research, 2007, **1–8**
doi:10.1093/nar/Gkm684

SmedGD: the *Schmidtea mediterranea* genome database

Sofia M.C. Robb, Eric Ross and Alejandro Sánchez Alvarado*

Department of Neurobiology and Anatomy, Howard Hughes Medical Institute, University of Utah School of Medicine, Salt Lake City, UT 84132, USA

Received July 24, 2007; Accepted August 20, 2007

ABSTRACT

The planarian *Schmidtea mediterranea* is rapidly emerging as a model organism for the study of regeneration, tissue homeostasis and stem cell biology. The recent sequencing, assembly and annotation of its genome are expected to further buoy the biomedical importance of this organism. In order to make the extensive data associated with the genome sequence accessible to the biomedical and planarian communities, we have created the *Schmidtea mediterranea* Genome Database (SmedGD). SmedGD integrates in a single web-accessible portal all available data associated with the planarian genome, including predicted and annotated genes, ESTs, protein homologies, gene expression patterns and RNAi phenotypes. Moreover, SmedGD was designed using tools provided by the Generic Model Organism Database (GMOD) project, thus making its data structure compatible with other model organism databases. Because of the unique phylogenetic position of planarians, SmedGD (http://smedgd.neuro.utah.edu) will prove useful not only to the planarian research community, but also to those engaged in developmental and evolutionary biology, comparative genomics, stem cell research and regeneration.

INTRODUCTION

Schmidtea mediterranea is a freshwater planarian of the phylum Platyhelminthes that is rapidly becoming a model system for the investigation of regeneration, tissue homeostasis and stem cell biology (1). Interest is being spurred on by the remarkable biology of these animals and the expanding repertoire of tools available to interrogate their biology (2). Like other non-parasitic flatworms, *S. mediterranea* has the ability to regenerate complete animals from small, excised body fragments. If a planarian is decapitated, both head and trunk fragments regenerate the missing body parts, i.e. the body and the head, respectively. The process takes seven days and results in the full, functional integration of the newly regenerated tissues and organs to the pre-existing structures. This remarkable developmental plasticity is made possible by a population of somatic stem cells known as neoblasts found throughout the body of planarians. Because of their abundance and characteristic undifferentiated state, neoblasts are both easy to identify and amenable to experimental manipulation. In addition, planarians belong to the Lophotrochozoa, a large group of understudied animals that is sister to the Ecdysozoa (e.g. *Drosophila* and *Caenorhabditis elegans*) and the Deuterostomes (e.g. non-mammalian and mammalian vertebrates). Presently, the Lophotrochozoa are poorly represented among currently sequenced genomes, and much of their molecular and developmental biology remain unexplored. Hence, the study of *S. mediterranea* is likely to both complement ongoing studies in available model systems and to expand our knowledge in a large number of long-standing and fundamental problems relevant to human health and biology (e.g. tissue homeostasis and regeneration) not readily studied in well-established model systems such as *Drosophila* and *C. elegans*.

Recently, many methodologies have been introduced to analyze planarian biology in depth. Prominent among these are the availability of ~78 000 ESTs, the study of gene function via robust and reproducible RNAi methodologies (3,4) and a sequenced, assembled and annotated genome. Because these growing resources were devoid of an integrative tool capable of coordinating the inflow of genomic and functional genomic data, we set out to create an easy to use, yet comprehensive database to house and mine this information. The result of this effort is the *Schmidtea mediterranea* Genome Database (SmedGD), which was constructed using tools from the Generic Model Organism Database (GMOD; http://www.gmod.org) project, and populated with GMOD-compliant annotation data from MAKER (5), as well as information collected from a wide variety of sources such as Gene

*To whom correspondence should be addressed. Tel: +1 801 581 3548; Fax: +1 801 585 5171; Email: sanchez@neuro.utah.edu

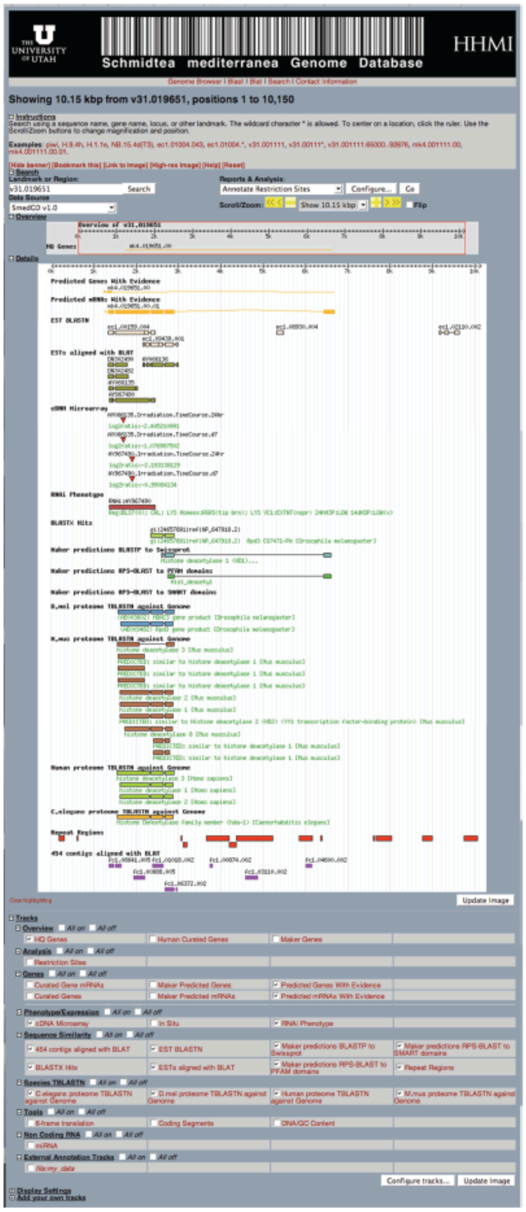
Ontology (GO), PFAM, SwissProt, SMART and others. Deploying GMOD tools to construct SmedGD facilitates and ensures: (1) interoperability of SmedGD with other genome databases (e.g. WormBase and FlyBase) to allow comparative genomic studies; (2) short- and long-term curation of gene models using Apollo (6); and (3) future expansions to include microarray gene expression data, *in situ* expression patterns, and PubMed references, for example. In its present form, members of the biomedical and planarian research communities can use SmedGD to find genes of interest and their homologs in other species, download sequences, link to other databases, and find RNAi phenotypes. SmedGD should prove a key resource in furthering the development and integration of *S. mediterranea* as an important model system for current studies of metazoan biology and human disease.

SmedGD architecture

In order to achieve interoperability between SmedGD and other organism databases such as FlyBase and WormBase, we constructed SmedGD using tools and components from GMOD, a NIH-funded effort aimed at providing generic software to build new genome/organism databases. SmedGD consists of two major components: a GFF database and a GBrowse generic framework (7) capable of autogenerating a generic genome browser from the database.

Data in GFF3 format and compliant with Sequence Ontology (8), is uploaded to a database using a perl script distributed with Bioperl (bp_seqfeature_load.pl) (9). The script converts GFF3 formatted annotations to Bio::SeqFeatureI objects, and generates a database schema that is GMOD-compliant. We used this mySQL-based schema instead of the Chado schema provided by GMOD because of the length of time it took to both load the *S. mediterranea* data into Chado (days instead of hours) and to query the resulting PostgresSQL database (minutes instead of seconds). Attempts were made to increase the efficiency of loading and querying, but not enough improvement was noted. Although Chado is a very robust relational database that has been successfully implemented in the development of BeetleBase (10) and ParameciumDB (11), we ascribe the underperformance of this schema in our hands to the sheer size of the *S. mediterranea* data (~900 Mb), i.e. 12.5× the size of ParameciumDB and 4.5× the size of BeetleBase.

We made sure that the mySQL database generated from the GFF3 files conformed to GMOD standards. Therefore, we parsed the database using GBrowse, which implemented a Bio::DB::SeqFeature::Store database adaptor to access and display rows of data or tracks that are mapped to specific locations in the genome (Figure 1). Customizations to the standard GMOD distribution were also made to accommodate additional database searching and sequence retrieval. CGI scripts that interface with SmedGD's mySQL database using DBI, a perl module, enable specialized queries of GO terms and RNAi phenotypes, and uniquely formatted protein homology search results. Changes made to the GBrowse configuration file allow for the linking to a CGI


THE UNIVERSITY OF UTAH
Schmidtea mediterranea Genome Database
HHMI
Genome Browser | Blast | Blat | Search | Contact Information
Predicted Genes With Evidence
Predicted mRNAs With Evidence
EST BLASTN
ESTs aligned with BLAT
cDNA Microarray
RNAi Phenotype
BLASTX Hits
Maker predictions BLASTP to Swissprot
Maker predictions RPS-BLAST to PFAM domains
Maker predictions RPS-BLAST to SMART domains
D.mel proteome TBLASTN against Genome
M.mus proteome TBLASTN against Genome
Human proteome TBLASTN against Genome
C.elegans proteome TBLASTN against Genome
Histone DeAcetylase family member (hda-1) [Caenorhabditis elegans]
Repeat Regions
454 contigs aligned with BLAT
Clear highlighting
Update Image
Configure tracks...
Update Image

Showing 10.15 kbp from v31.019651, positions 1 to 10,150

⊟ Instructions
Search using a sequence name, gene name, locus, or other landmark. The wildcard character * is allowed. To center on a location, click the ruler. Use the Scroll/Zoom buttons to change magnification and position.

Examples: piwi, H.9.4h, H.1.1e, NB.15.4d(T3), ec1.01004.043, ec1.01004.*, v31.001111, v31.00111*, v31.001111:65000..92876, mk4.001111.00, mk4.001111.00.01.

[Hide banner] [Bookmark this] [Link to Image] [High-res Image] [Help] [Reset]

⊟ Search
Landmark or Region: v31.019651 Search
Data Source: SmedGD v1.0
Reports & Analysis: Annotate Restriction Sites Configure... Go
Scroll/Zoom: Show 10.15 kbp □ Flip

⊟ Overview
Overview of v31.019651
HD Genes: mk4.019651.00

⊟ Details

⊟ Tracks
⊟ Overview □ All on □ All off
☑ HD Genes | □ Human Curated Genes | □ Maker Genes
⊟ Analysis □ All on □ All off
□ Restriction Sites
⊟ Genes □ All on □ All off
□ Curated Gene mRNAs | □ Maker Predicted Genes | ☑ Predicted Genes With Evidence
□ Curated Genes | □ Maker Predicted mRNAs | ☑ Predicted mRNAs With Evidence
⊟ Phenotype/Expression □ All on □ All off
☑ cDNA Microarray | □ In Situ | ☑ RNAi Phenotype
⊟ Sequence Similarity □ All on □ All off
☑ 454 contigs aligned with BLAT | ☑ EST BLASTN | ☑ Maker predictions BLASTP to Swissprot | ☑ Maker predictions RPS-BLAST to SMART domains
☑ BLASTX Hits | ☑ ESTs aligned with BLAT | ☑ Maker predictions RPS-BLAST to PFAM domains | ☑ Repeat Regions
⊟ Species TBLASTN □ All on □ All off
☑ C.elegans proteome TBLASTN against Genome | ☑ D.mel proteome TBLASTN against Genome | ☑ Human proteome TBLASTN against Genome | ☑ M.mus proteome TBLASTN against Genome
⊟ Tools □ All on □ All off
□ 6-frame translation | □ Coding Segments | □ DNA/GC Content
⊟ Non Coding RNA □ All on □ All off
□ miRNA
⊟ External Annotation Tracks □ All on □ All off
□ file:my_data

⊞ Display Settings
⊞ Add your own tracks

Figure 1. Screen capture of SmedGD displaying genomic contig v31.019651. This contig has only one predicted gene, which has 5 exons and a 3′UTR. The tracks displayed include the gene model, its corresponding predicted transcript, and the relevant biological evidence associated with this model (see text for detailed explanations of each track). From this data, users can see the details of the gene model and its evidence (all of the predicted exons are supported by EST and protein evidence), and that the gene is likely coding for a histone deacetylase. Double-stranded RNA has been used to silence this gene and the resulting phenotypes are listed. cDNA Microarray data is not yet available, but a sample of how this information will be viewed is presented. An arrow pointing down indicates down-regulation of the gene in the experimental group.

script that uses 'fastacmd', free software distributed by the NCBI (http://www.ncbi.nlm.nih.gov), to retrieve sequences from specially formatted fasta files and displays them when the 'Name' of an mRNA is selected in the mRNA 'Details Page' (see below). The web page displaying the retrieved sequence includes a link to the NCBI BLAST web server, which when selected will auto fill the NCBI BLAST forms with the retrieved planarian sequence.

Physically, SmedGD is housed on an Apple Xserve G5 computer with 3 GB of RAM. The operating system used is the Mac OS X Server operating system—version 10.4 and the web server software used is Apache—version 1.3.

SmedGD contents

SmedGD is a storehouse of valuable data that is easily accessible and conveniently mapped to the planarian genome sequence. The layers of data mapped to the genome sequence are MAKER annotations, ESTs, 454 cDNA sequences, protein homology, protein domains, Gene Ontology terms, RNAi phenotypes, mRNA expression patterns, human curated genes and microRNAs (Table 1).

Assembly and annotation data

Sequencing and assembly data was provided by the Washington University Genome Sequencing Center in St. Louis, MO. The current assembly version 3.1 (v31), consists of ~900 Mb of sequence distributed over 43 294 supercontigs. SmedGD also houses the annotation of v31, which was performed using a recently developed automated annotation pipeline named MAKER (5). MAKER implements an algorithm that uses protein homology and collectively assembled ESTs from various *S. mediterranea* EST projects (12,13) using CAP3 (14) to predict genes, many with 5′ and 3′ UTRs. The predicted genes (31 955) and the accompanying transcripts and splice variants are mapped to the genome sequence. The protein homology and the EST alignments that were used for gene prediction are also mapped to the genome to give the investigator the ability to judge the accuracy of the gene prediction.

cDNA data

Additional evidence is provided to encourage critical analysis of the predictions. The ~78 000 ESTs that were assembled and used by MAKER were individually aligned to the genome sequence using BLAT (15). We also aligned ~9000 mRNAs sequenced using 454 technology (16). These alignments enable the user to identify the genomic location of previously published ESTs and grade the positions of exons and splice sites.

Protein homology data

To assist in determining gene function, homologs of the MAKER-predicted protein sequences are included in SmedGD. This set of data was obtained by comparing the MAKER proteins to the PFAM (17) and SMART (18) portions of the CDD database (19), and to Swiss-Prot (20) using RPS-BLAST (21) and BLAST (22). To further validate the predicted genes and identify missing exons, the proteomes of *C. elegans*, *Drosophila melanogaster*, *Homo sapiens*, and *Mus musculus* were individually subjected to TBLASTN comparisons (22) against the planarian genome translated into six frames.

Table 1. Types and number of entries that comprise SmedGD

Data type	Count
Sequence type	
Genomic contigs	43 294
Predicted genes:	31 955
MAKER	30 437
SNAP	1518
Predicted mRNAs	32 448
Human curated genes	1000
ESTs	78 101
EST contigs	15 043
454 contigs	9071
MicroRNAs	71
Protein homology	
Swissprot	25 733
C. elegans Proteome	5319
H. Sapiens Proteome	10 746
M. musculus Proteome	14 713
D. melanogaster	4301
BLASTX with reference proteomes	88 293
Protein Domains	
PFAM	14 442
SMART	5711
GO terms	
Molecular Function	25 075
Cellular Location	12 294
Biological Process	23 842
Experimental data	
RNAi phenotypes	303
mRNA expression patterns	123

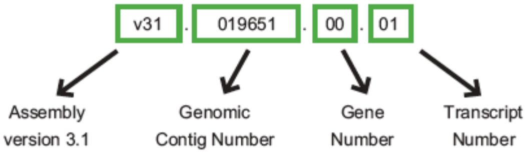
Gene ontology

Gene Ontology terms (23–25) associated with each of the homologous database entries were extracted and entered into SmedGD. The addition of GO terms makes searching the database for genes with a specific association to a biological process, molecular function or cellular location possible (see below).

RNAi and whole-mount *in situ* hybridization data

In 2005, the results of an extensive RNAi screen in *S. mediterranea* were published (3). This work has been incorporated into SmedGD. RNAi phenotypes can be searched and the results link to the genome browser. The 'Details Page' displays images and descriptions of phenotypes. In addition to RNAi experimental data, mRNA expression patterns are also included in the database. If an *in situ* hybridization of an mRNA exists, images are correlated with the position of the corresponding gene in the genome and can be accessed for viewing directly from the genome browser (see below).

Table 2. Standardized nomenclature strategy to denote individual entries in SmedGD



Type	Example name	Description of field			
Genomic contigs	v31.019651	Assembly version 3.1			Contig No*
Genes	mk4.019651.00	Maker Run 4	Contig No.*		Gene No.**
Transcripts	mk4.019651.00.01	Maker Run 4	Contig No.*	Gene No.**	Transcript No.**
EST contigs	ec1.00159.004	EST contig Version 1	EST contig No.***		No. of reads ****
454 contigs	fc1.08641.005	454 contig Version 1	454 contig No.***		No. of reads ****

*padded to 6 digits.
**padded to 2 digits.
***padded to 5 digits.
****padded to 3 digits.

Additional data

Additional features mapped to the genome include human-curated gene annotations and mature microRNA sequences (26). Because of the modular nature of GMOD, it will not be difficult to add additional data as it is collected. Currently, we are testing implementations to incorporate microarray expression data into SmedGD.

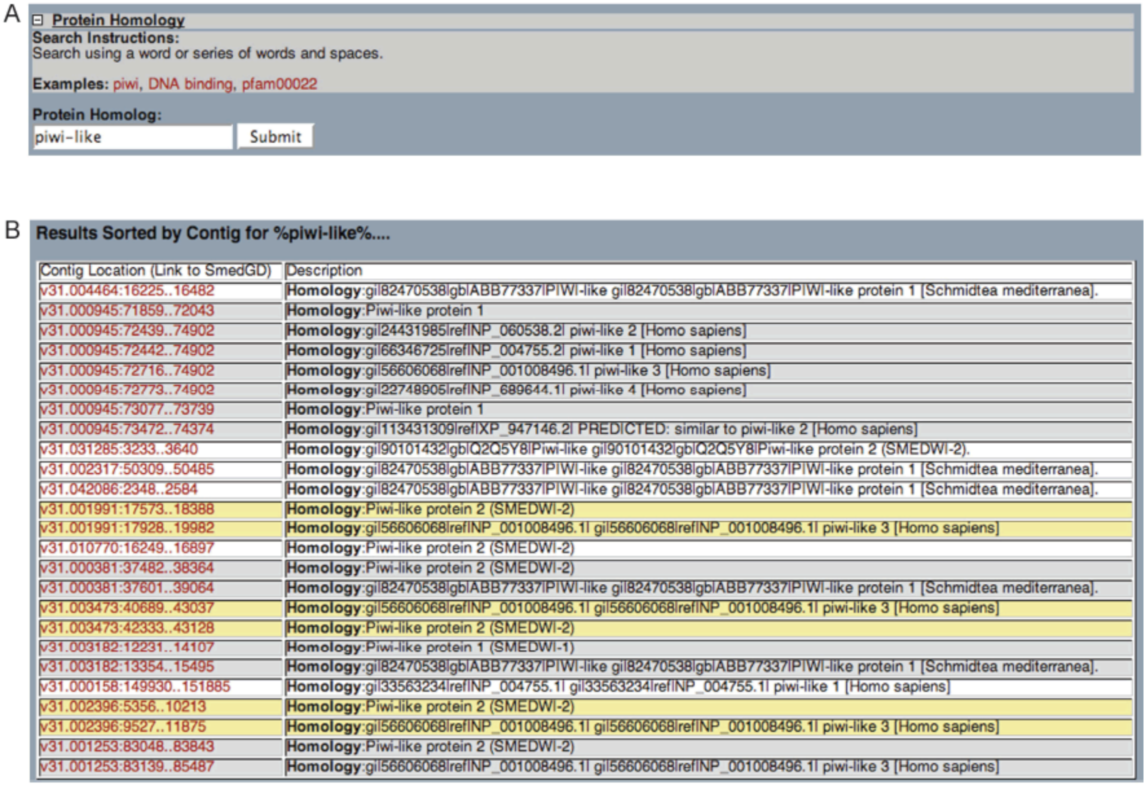
Using SmedGD

Overview. SmedGD can be navigated from the top tool bar, where links to the 'Genome Browser', sequence search interfaces 'Blast', 'Blat' and the text search interface 'Search' can be found (Figure 1). In order to facilitate large-scale, high-throughput mining of the data in SmedGD, we established a standard naming nomenclature for all genes and associated data (Table 2). Each name consists of up to four fields separated by a period. The first field identifies the type of data being viewed (assembly version, annotation run, EST or 454 cDNA contigs). The second field indicates the number of the genomic or EST/454 contig in which the data can be found. The third and fourth fields indicate the identification number of the feature (e.g. gene, transcript) associated with the genomic contig. A description of the nomenclature with examples is presented in Table 2.

Genome browser page. From within the genome browser the 'Search Landmark or Region' function can be used to query the database to find features mapped to specific regions of the genome. Examples of search terms are protein homology (piwi), contig names (v31.019651), and gene names (mk4.019651.00). The 'Overview Panel' provides a simplified view of the entire genomic contig (Figure 1). The red box indicates the area of the contig which is being viewed in the 'Details' panel. The red box can be recentered on a different area of the contig by clicking in the 'Overview' panel. The size of the box and therefore the number of base pairs viewed in the details panel can be altered by changing the parameters in the 'Scroll/Zoom' drop down menu or by adding the desired region to be viewed in the 'Search Landmark or Region' box of the contig. (e.g. v31.019651:1375..2445). The 'Details' panel contains the information that aligns to the genome, ranging from predicted genes to RNAi phenotypes. The information displayed in the 'Details' panel is controlled by the selection of tracks. Tracks are biologically and bioinformatically obtained data that have been aligned to the genomic contigs. The types of data are subdivided into the groups 'Genes', 'Phenotype/Expression', 'Sequence Similarity', 'Species TBLASTN' and 'Non-Coding RNA'.

Genes. Within this group, genes/mRNAs predicted by MAKER that are supported by High Quality evidence (e.g. ESTs and protein homology), and 'Human-Curated Genes/mRNAs' (human-edited genes/mRNAs) are displayed. Presently, and using Apollo (6), only one investigator has manually curated gene models (Dr Alejandro Sánchez Alvarado). To identify the provenance of the curation, the edited gene models are identified by the three letter designation, e.g. ASA, followed by a gene number and transcript number (ASA.00084.01). It is expected that others will join efforts in the curation of genes. Such edits will be identified by a unique three letter designation corresponding to the name of the person or laboratory responsible for the editing.

Phenotype/expression. Display of RNAi Phenotypes and *in situ* data are controlled by this group of tracks. The RNAi Phenotypes appear as a bar that spans the length of the EST used to disrupt gene function by RNA interference. The description of the resulting phenotype is located under the bar (RNAi:AY967490). If an



⊟ Protein Homology

Search Instructions:
Search using a word or series of words and spaces.

Examples: piwi, DNA binding, pfam00022

Protein Homolog:
piwi-like Submit

Results Sorted by Contig for %piwi-like%....

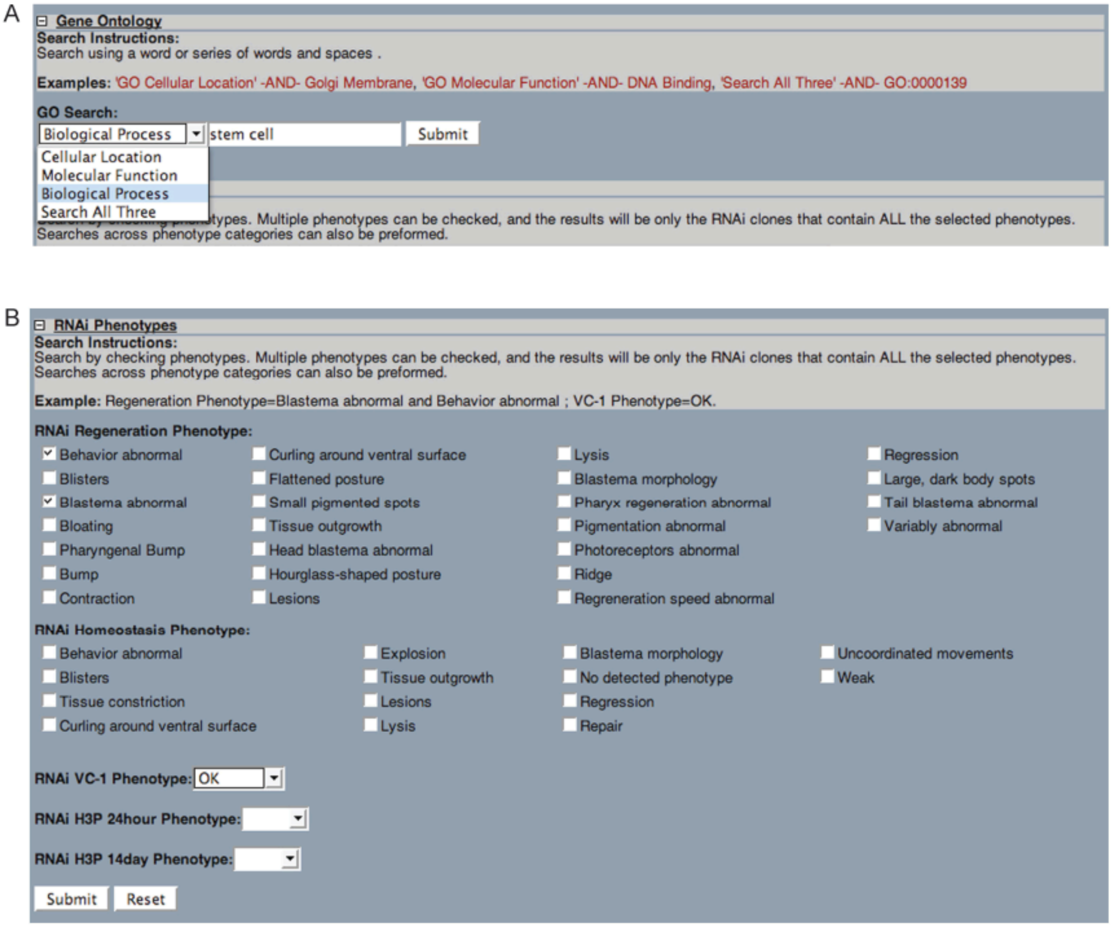
Contig Location (Link to SmedGD)	Description
v31.004464:16225..16482	**Homology**:gi|82470538|gb|ABB77337|PIWI-like gi|82470538|gb|ABB77337|PIWI-like protein 1 [Schmidtea mediterranea].
v31.000945:71859..72043	**Homology**:Piwi-like protein 1
v31.000945:72439..74902	**Homology**:gi|24431985|ref|NP_060538.2| piwi-like 2 [Homo sapiens]
v31.000945:72442..74902	**Homology**:gi|66346725|ref|NP_004755.2| piwi-like 1 [Homo sapiens]
v31.000945:72716..74902	**Homology**:gi|56606068|ref|NP_001008496.1| piwi-like 3 [Homo sapiens]
v31.000945:72773..74902	**Homology**:gi|22748905|ref|NP_689644.1| piwi-like 4 [Homo sapiens]
v31.000945:73077..73739	**Homology**:Piwi-like protein 1
v31.000945:73472..74374	**Homology**:gi|113431309|ref|XP_947146.2| PREDICTED: similar to piwi-like 2 [Homo sapiens]
v31.031285:3233..3640	**Homology**:gi|90101432|gb|Q2Q5Y8|Piwi-like gi|90101432|gb|Q2Q5Y8|Piwi-like protein 2 (SMEDWI-2).
v31.002317:50309..50485	**Homology**:gi|82470538|gb|ABB77337|PIWI-like gi|82470538|gb|ABB77337|PIWI-like protein 1 [Schmidtea mediterranea].
v31.042086:2348..2584	**Homology**:gi|82470538|gb|ABB77337|PIWI-like gi|82470538|gb|ABB77337|PIWI-like protein 1 [Schmidtea mediterranea].
v31.001991:17573..18388	**Homology**:Piwi-like protein 2 (SMEDWI-2)
v31.001991:17928..19982	**Homology**:gi|56606068|ref|NP_001008496.1| gi|56606068|ref|NP_001008496.1| piwi-like 3 [Homo sapiens]
v31.010770:16249..16897	**Homology**:Piwi-like protein 2 (SMEDWI-2)
v31.000381:37482..38364	**Homology**:Piwi-like protein 2 (SMEDWI-2)
v31.000381:37601..39064	**Homology**:gi|82470538|gb|ABB77337|PIWI-like gi|82470538|gb|ABB77337|PIWI-like protein 1 [Schmidtea mediterranea].
v31.003473:40689..43037	**Homology**:gi|56606068|ref|NP_001008496.1| gi|56606068|ref|NP_001008496.1| piwi-like 3 [Homo sapiens]
v31.003473:42333..43128	**Homology**:Piwi-like protein 2 (SMEDWI-2)
v31.003182:12231..14107	**Homology**:Piwi-like protein 1 (SMEDWI-1)
v31.003182:13354..15495	**Homology**:gi|82470538|gb|ABB77337|PIWI-like gi|82470538|gb|ABB77337|PIWI-like protein 1 [Schmidtea mediterranea].
v31.000158:149930..151885	**Homology**:gi|33563234|ref|NP_004755.1| gi|33563234|ref|NP_004755.1| piwi-like 1 [Homo sapiens]
v31.002396:5356..10213	**Homology**:Piwi-like protein 2 (SMEDWI-2)
v31.002396:9527..11875	**Homology**:gi|56606068|ref|NP_001008496.1| gi|56606068|ref|NP_001008496.1| piwi-like 3 [Homo sapiens]
v31.001253:83048..83843	**Homology**:Piwi-like protein 2 (SMEDWI-2)
v31.001253:83139..85487	**Homology**:gi|56606068|ref|NP_001008496.1| gi|56606068|ref|NP_001008496.1| piwi-like 3 [Homo sapiens]

Figure 2. (**A**) The 'Protein Homology' Search interface. In this example, the search term 'piwi-like' is being submitted. Each of the hits from SwissProt, SMART, PFAM and the species-specific databases are searched for the user query term. (**B**) Results of the search are sorted by genomic contig and location. When more than one result is found on one contig, the matches are grouped and the background will be similarly colored. When there is more than one protein match for one genomic location, it is often due to this sequence matching more than one database. When there is only one result per contig the background is colored white. The contig and location are hyperlinked to the genome browser for further inspection.

annotation has an mRNA *in situ* hybidization associated with it, a hyperlinked thumbnail is displayed that can be selected to obtain more detailed information (AY967481:*in situ*).

Sequence similarity. EST and 454 cDNA contigs, individual ESTs and protein sequences have been aligned to the genome and their display is controlled in this section of the 'Tracks'. In the '454 contigs aligned with BLAT', 454 contigs prepared from 454 sequencing reads were aligned to the genome using the Blat algorithm and standard psl output. BLAT is designed to quickly find sequences of 90% and greater similarity with a score of 30. In the 'BLASTX Hits track' nucleotide to protein comparisons, via WU-BLASTX six-frame translation (http://blast.wustl.edu) was used to find similarity hits of genomic sequence searched against a reference protein dataset comprising of the proteomes of *E. coli*, *Saccharomyces cerevisiae*, *D. melanogaster*, *C. elegans*, *Platyhelminthes*, *Ciona intestinalis*, mouse and human. Statistical cutoffs of 40% identity and expectation value of 1*e*−5 were used. The 'EST BLASTN' track is made up of contigs of *S. mediterranea* ESTs that were aligned to the *S. mediterranea* genome using cutoffs of 85% identity and expectation value of 1*e*−10. Similarly, 'ESTs aligned with BLAT' is made up of all the individual *S. mediterranea* ESTs aligned to the *S. mediterranea* genome using WU-BLAST formatted output from BLAT with expectation value cutoffs of 1*e*−95 and with at least 45% alignment of the EST.

There are three tracks that align whole proteins or protein domains to the MAKER predictions. The first of these, 'BLASTP to Swissprot' are alignments of predicted proteins from MAKER gene models against Swissprot. Next is the 'MAKER predictions RPS-BLAST to PFAM'. These are alignments of predicted proteins from MAKER gene models against PFAM using NCBI RPS-Blast to identify known functional motifs. The last of the three tracks is the 'MAKER predictions RPS-BLAST to SMART'. These are alignments of predicted proteins from MAKER gene models against SMART using NCBI RPS-Blast. An expectation value cutoff of 1*e*−3 and 40%



A

⊟ Gene Ontology

Search Instructions:
Search using a word or series of words and spaces .

Examples: 'GO Cellular Location' -AND- Golgi Membrane, 'GO Molecular Function' -AND- DNA Binding, 'Search All Three' -AND- GO:0000139

GO Search:
Biological Process | stem cell

...types. Multiple phenotypes can be checked, and the results will be only the RNAi clones that contain ALL the selected phenotypes.
Searches across phenotype categories can also be preformed.

B

⊟ RNAi Phenotypes

Search Instructions:
Search by checking phenotypes. Multiple phenotypes can be checked, and the results will be only the RNAi clones that contain ALL the selected phenotypes.
Searches across phenotype categories can also be preformed.

Example: Regeneration Phenotype=Blastema abnormal and Behavior abnormal ; VC-1 Phenotype=OK.

RNAi Regeneration Phenotype:

☑ Behavior abnormal
☐ Blisters
☑ Blastema abnormal
☐ Bloating
☐ Pharyngenal Bump
☐ Bump
☐ Contraction
☐ Curling around ventral surface
☐ Flattened posture
☐ Small pigmented spots
☐ Tissue outgrowth
☐ Head blastema abnormal
☐ Hourglass-shaped posture
☐ Lesions
☐ Lysis
☐ Blastema morphology
☐ Pharyx regeneration abnormal
☐ Pigmentation abnormal
☐ Photoreceptors abnormal
☐ Ridge
☐ Regeneration speed abnormal
☐ Regression
☐ Large, dark body spots
☐ Tail blastema abnormal
☐ Variably abnormal

RNAi Homeostasis Phenotype:

☐ Behavior abnormal
☐ Blisters
☐ Tissue constriction
☐ Curling around ventral surface
☐ Explosion
☐ Tissue outgrowth
☐ Lesions
☐ Lysis
☐ Blastema morphology
☐ No detected phenotype
☐ Regression
☐ Repair
☐ Uncoordinated movements
☐ Weak

RNAi VC-1 Phenotype: OK

RNAi H3P 24hour Phenotype:

RNAi H3P 14day Phenotype:

Figure 3. (**A**) 'Gene Ontology' Search interface. Users can search for terms, such as 'stem cell' in one of three Gene Ontology (GO) categories (e.g. Biological Process). Any protein homology hit to the genome that had GO terms associated with it will be searched and the corresponding genomic contig and location will be returned in a fashion similar to the 'Protein Homology' results page. (**B**) The 'RNAi Phenotype' search interface is used by selecting phenotypes from the five categories listed. More information about these categories and phenotypes can be found in (3). The selections are additive, therefore each of the records returned will contain all of the chosen phenotypes.

alignment requirement with the protein hit was used in filtering the results of these three protein homology searches. The final option in the 'Sequence Similarity' grouping is 'Repeat Regions', which when selected displays areas of the genome containing interspersed repeats and low complexity DNA sequences as identified by RepeatMasker (http://www.repeatmasker.org/) using a *S. mediterranea*-specific repeat library (5).

Species TBLASTN. This track displays alignments of the proteomes of individual species to the planarian genome using TBLASTN. The proteomes of *C. elegans*, *D. melanogaster*, *H. sapiens* and *M. musculus* were aligned and results with expectation values equal to or less than 1e−5 and 30% alignment were selected for display in the browser. These pre-computed sets of data allow users to determine quickly if homologs to the gene of interest exist in the genomes of established genetic model systems.

Non-coding RNA. This track maps published mature miRNA sequences (26) to the genome (sme-miR-2b, for example). Efforts are underway to populate this track with miRNA gene predictions as well.

In all of the above cases, the images of the features in the 'Details' area of the genome browser that align to the genome can be selected to retrieve a 'Details Page'. The information in the 'Details Page' of different feature types displays feature-specific information and links to tools and associated websites. For example, in the mRNA 'Details Page', the mRNA name is hyperlinked and selecting it will retrieve protein and nucleotide sequence, while in the miRNA 'Details Page', the name will link to the selected record in miRBase (http://microrna.sanger.ac.uk).

BLAST, BLAT and Search pages. The Search tools provide methods for users to directly query the databases of SmedGD. Sequences can be used to find homologs in *S. mediterranea* by using BLAST and BLAT. Both nucleotide and protein sequences can be used with BLAST, while BLAT will search the whole genome with nucleotide sequences only. The BLAT result page contains a button that will link the results to an auto-generated track in the browser, such that the queried sequence will be aligned visually to the genome and will be in correct alignment with the other data tracks. In the Search page a user can query the text data stored in SmedGD. Protein homology is queried in the Swissprot, SMART, PFAM and proteome hits to the genome (Figure 2A and B). Gene Ontology terms can be searched by cellular location, molecular function or biological process (Figure 3A). Finally RNAi phenotypes can also be queried with an assortment of checkboxes and drop down menus in an additive 'AND' fashion (Figure 3B).

Curation and future expansions of SmedGD

Since SmedGD is based on GMOD tools, updates, expansion and gene model curation are significantly streamlined. This is a key feature of a GMOD-compliant organism/genome database as genome annotation and the generation of evidence to curate such models is a community-wide ongoing effort. As such, curation of individual genes and data associated with their function will need to be updated regularly. Apollo (http://fruitfly.org/annot/apollo/) will be used to edit gene annotations, since this software has been adopted by GMOD as its annotation workbench, and its outputs are GMOD compliant.

Currently under development is the incorporation of microarray data to SmedGD. Expression data associated with over 10 000 unique ESTs will be mapped to each corresponding alignment on the genome, allowing users of SmedGD to correlate genome location, expression pattern and functional data in a single window. In addition, we are exploring the implementation of Textpresso (www.textpresso.org) (27) for literature searching of both current and historical publications associated with planarians, regeneration, tissue homeostasis and stem cells. Finally, as the information accrued increases in complexity, we will also implement a BioMart-based data management system in order to facilitate data exports and simplify database management tasks.

ACKNOWLEDGEMENTS

Lincoln Stein, Scott Cain and other GMOD developers and users for assistance in the setting up and running of SmedGD; Mark Yandell and Brandi Cantarel for the annotation data that served to populate SmedGD; Marc Friedlaender and Nikolaus Rajewsky for the miRNA data; Juancarlo Añez and members of the Sánchez Alvarado laboratory and the planarian and biomedical community, in particular Voichita Marinescu, Kyle Gurley, Peter W. Reddien, Phillip A. Newmark and Tatjana Piotrowski for making suggestions to improve the content and user interface. This work was supported by NIH Genetics Training Grant 5 T32 GM007464 to S.M.C.R. A.S.A. is an Investigator of the Howard Hughes Medical Institute. Funding to pay the Open Access publication charges for the article was provided by the Howard Hughes Medical Institute.

Conflict of interest statement. None declared.

REFERENCES

1. Sánchez Alvarado,A. and Tsonis,P.A. (2006) Bridging the regeneration gap: genetic insights from diverse animal models. *Nat. Rev. Genet.*, 7, 873–884.
2. Sánchez Alvarado,A. (2006) Planarian regeneration: its end is its beginning. *Cell*, **124**, 241–245.
3. Reddien,P.W., Bermange,A.L., Murfitt,K.J., Jennings,J.R. and Sánchez Alvarado,A. (2005) Identification of genes needed for regeneration, stem cell function, and tissue homeostasis by systematic gene perturbation in planaria. *Dev. Cell*, **8**, 635–649.
4. Newmark,P.A., Reddien,P.W., Cebria,F. and Sánchez Alvarado,A. (2003) Ingestion of bacterially expressed double-stranded RNA inhibits gene expression in planarians. *Proc. Natl Acad. Sci. USA*, **100(Suppl)**, 11861–11865.
5. Cantarel,B., Korf,I., Robb,S.M.C., Parra,G., Ross,E., Moore,B., Holt,C., Sánchez Alvarado,A. and Yandell,M. (2007). MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. *In press*.
6. Lewis,S.E., Searle,S.M., Harris,N., Gibson,M., Lyer,V., Richter,J., Wiel,C., Bayraktaroglir,L., Birney,E. *et al.* (2002) Apollo: a sequence annotation editor. *Genome Biol.*, **3**, RESEARCH0082.
7. Stein,L.D., Mungall,C., Shu,S., Caudy,M., Mangone,M., Day,A., Nickerson,E., Stajich,J.E., Harris,T.W. *et al.* (2002) The generic genome browser: a building block for a model organism system database. *Genome Res.*, **12**, 1599–1610.
8. Eilbeck,K., Lewis,S.E., Mungall,C.J., Yandell,M., Stein,L., Durbin,R. and Ashburner,M. (2005) The sequence ontology: a tool for the unification of genome annotations. *Genome Biol.*, **6**, R44.
9. Stajich,J.E., Block,D., Boulez,K., Brenner,S.E., Chervitz,S.A., Dagdigian,C., Fuellen,G., Gilbert,J.G., Korf,I. *et al.* (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.
10. Wang,L., Wang,S., Li,Y., Paradesi,M.S. and Brown,S.J. (2007) BeetleBase: the model organism database for Tribolium castaneum. *Nucleic Acids Res.*, **35**, D476–D479.
11. Arnaiz,O., Cain,S., Cohen,J. and Sperling,L. (2007) ParameciumDB: a community resource that integrates the Paramecium tetraurelia genome sequence with genetic data. *Nucleic Acids Res.*, **35**, D439–D444.
12. Zayas,R.M., Hernandez,A., Habermann,B., Wang,Y., Stary,J.M. and Newmark,P.A. (2005) The planarian Schmidtea mediterranea as a model for epigenetic germ cell specification: analysis of ESTs from the hermaphroditic strain. *Proc. Natl Acad. Sci. USA*, **102**, 18491–18496.
13. Sánchez Alvarado,A., Newmark,P.A., Robb,S.M. and Juste,R. (2002) The Schmidtea mediterranea database as a molecular resource for studying platyhelminthes, stem cells and regeneration. *Development*, **129**, 5659–5665.
14. Huang,X. and Madan,A. (1999) CAP3: a DNA sequence assembly program. *Genome Res.*, **9**, 868–877.
15. Kent,W.J. (2002) BLAT–the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.
16. Emrich,S.J., Barbazuk,W.B., Li,L. and Schnable,P.S. (2007) Gene discovery and annotation using LCM-454 transcriptome sequencing. *Genome Res.*, **17**, 69–73.
17. Bateman,A., Coin,L., Durbin,R., Finn,R.D., Hollich,V., Griffiths-Jones,S., Khanna,A., Marshall,M., Moxon,S. *et al.* (2004) The Pfam protein families database. *Nucleic Acids Res.*, **32**, D138–D141.

18. Ponting,C.P., Schultz,J., Milpetz,F. and Bork,P. (1999) SMART: identification and annotation of domains from signalling and extracellular protein sequences. *Nucleic Acids Res.*, **27**, 229–232.
19. Marchler-Bauer,A., Anderson,J.B., Derbyshire,M.K., DeWeese-Scott,C., Gonzales,N.R., Gwadz,M., Hao,L., He,S., Hurwitz,D.I. *et al.* (2007) CDD: a conserved domain database for interactive domain family analysis. *Nucleic Acids Res.*, **35**, D237–D240.
20. Boeckmann,B., Bairoch,A., Apweiler,R., Blatter,M.C., Estreicher,A., Gasteiger,E., Martin,M.J., Michoud,K., O'Donovan,C. *et al.* (2003) The SWISS-PROT protein knowledge-base and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, **31**, 365–370.
21. Gowri,V.S., Tina,K.G., Krishnadev,O. and Srinivasan,N. (2007) Strategies for the effective identification of remotely related sequences in multiple PSSM search approach. *Proteins*, **67**, 789–794.
22. McGinnis,S. and Madden,T.L. (2004) BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res.*, **32**, W20–W25.
23. Camon,E., Magrane,M., Barrell,D., Binns,D., Fleischmann,W., Kersey,P., Mulder,N., Oinn,T., Maslen,J. *et al.* (2003) The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Res.*, **13**, 662–672.
24. Camon,E., Magrane,M., Barrell,D., Lee,V., Dimmer,E., Maslen,J., Binns,D., Harte,N., Lopez,R. *et al.* (2004) The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res.*, **32**, D262–D266.
25. Mulder,N.J., Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Binns,D., Bradley,P., Bork,P., Bucher,P. *et al.* (2005) InterPro, progress and status in 2005. *Nucleic Acids Res.*, **33**, D201–D205.
26. Palakodeti,D., Smielewska,M. and Graveley,B.R. (2006) MicroRNAs from the Planarian Schmidtea mediterranea: a model system for stem cell biology. *Rna*, **12**, 1640–1649.
27. Muller,H.M., Kenny,E.E. and Sternberg,P.W. (2004) Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol.*, **2**, e309.

CHAPTER 5

DEVELOPMENT OF EPIGENETIC APPROACHES FOR *SCHMIDTEA MEDITERRANEA* AND IMPLEMENTATION OF LARGE SCALE BIOLOGICAL ANALYSIS

Abstract

Planaria are an ideal model for studying adult stem cells (ASCs) but were lacking in the accessibility of genomic information. However, in recent years, bioinformatic tools such as annotations created by MAKER (Cantarel et al., 2008) and a genome browser, SmedGD (Robb et al., 2007), have facilitated the investigation of biological questions, in particular the epigenetics of ASCs. Given the role chromatin architecture is likely to play in defining the genomic output of a given cell, the multipotentiality found in the stem cells of *S. mediterranea* is no different. Canonical histones, H3, H4, H2A, H2B, and H1 and the histone variants H3.3, H2A.X, H2A.Z are all present in the *S. mediterranea genome*. We have identified conserved epitopes for commercially available anti-histone antibodies and have tested their cross-reactivity with planarian chromatin. Western blot analyses and immunocytochemistry have confirmed, for example, the existence of acetylated and methylated posttranscriptional modifications of histone H3. The levels of these modifications have been found to be different in different cell types and may even be useful for identifying subsets within heterogeneous populations. Histone modifying enzymes have been identified, cloned, visualized with whole mount *in situ* hybridization and perturbed with RNAi. My aims were to determine if any epigenetics modifying enzymes are localized to ASCs or if any have specific roles in ASC self-renewal or maintenance. Six genes were identified that are likely to have key roles in stem cell self-renewal, maintenance and differentiation; *Smed-HDAC1-1, Smed-SETD8-1, Smed-RL7A, Smed-NHP2L1, Smed-RPL30* and *Smed-RPS12*.

Introduction

DNA is organized inside the nucleus into chromatin. The basic unit of chromatin is a complex of 147 base pairs of DNA wrapped around an octamer of histones, two each of histones H3, H4, H2A and H2B. The histone tails, or the 20-30 amino acids of the amino terminus of these histones extend out of the core nucleosome and can become posttranslationally modified. Specific modifications have been observed to correspond to specific transcriptional and cellular events. For example, histone H3 serine 10 becomes phosphorylated (H3P) during the G2 to M transition of the cell cycle, tri-methylation of histone H3 lysine 9 (H3K9-Me3) is typically found in regions of the genome which are transcriptionally inactive or silenced, while histone H3 lysine 9, lysine 14 acetylation (H3K9K14-Ac) is found in transcriptionally active regions of the genome (Peterson and Laniel, 2004). Modifications to the histones such as these make up the foundation of epigenetics.

Epigenetics is the interaction of genetic and environmental factors. This interaction produces an observable characteristic or trait, i.e., a phenotype. Epigenetic mechanisms regulate gene expression, but they do not change the actual DNA sequence. Some examples of epigenetics are posttranslational modifications of the amino acids of the histones and DNA methylation (Figure 5.1). Posttranslational modifications, such as H3K9-Me3, lead to heterochromatin formation. Fission yeast utilizes heterochromatin formation at its centromeres. DNA methylation is also known to be involved in imprinting. An example of imprinting is illustrated by the expression of only the paternal copy of the gene encoding Insulin-like growth factor 2 (DeChiara et al., 1991). Histone posttranslational modifications and DNA methylation are used in conjunction for events such as X-inactivation. H3K9-Me3 and DNA methylation both need to occur for X-inactivation. The coloration of the fur of a calico cat (Figure 5.2) is a visible manifestation of X-inactivation. The two fur coloration alleles, "black" and "orange" are found on the X chromosome. The inactivation of one or the other gene results in a patch of fur that is the color produced by the other active gene.

Most studies of epigenetic regulation in stem cells have been carried out in embryonic stem (ES) cells. These studies show that epigenetic regulation has a very important role in ES cell function. When ES cells are exposed to retinoic acid they begin to differentiate and express genes that are instrumental to differentiation. Four days after exposure to retinoic acid the mRNA levels of Hoxb1 rise and by day 10 they drop. The acetylation levels at the promoter region of this gene have a similar pattern. The increase in Histone H3 lysine 9 acetylation (H3K9-Ac) and Histone H3 lysine 4 methylation (H3K4-Me) is coupled with the increase in Hoxb1 expression and a decrease in these modifications is found when gene expression drops. The inverse is seen with H3K9-Me. This indicates that histone posttranslational modifications or chromatin organization is closely associated with biological processes such as differentiation (Chambeyron and Bickmore, 2004).

It has also been demonstrated that a specific pattern of modifications found in ES cells, termed ‘‘bivalent domains’’, help to maintain pluripotency. This bivalent domain consists of a pairing of H3 lysine 27 tri-methylation (H3K27-Me3) and H3 lysine 4 tri-methylation (H3K4-Me3) which silence developmental genes in ES cells but leaves them “poised” for expression during differentiation (Bernstein et al., 2006).

ES cells are a product of the collection and culturing of cells from the inner cell mass (ICM) of the mammalian blastocyst. Are cultured ES cells and the *in vivo* ICM cells experimentally the same? Can the data collected from these ES cell experiments be transposed onto *in vivo* studies? A study comparing the chromatin states of ES and ICM cells demonstrate that *in vitro* data is not directly applicable to *in vivo* biology (O'Neill et al., 2006). Typically *in vivo* ICM cells are too few for conventional chromatin immunoprecipitation (ChIP), but modifications in the ChIP methodology can be used that eases the limitations on cell number. Histone posttranslational modification levels of a set of genes known to be expressed in ICM and ES cells (*Nanog* and *Oct4*) and genes not expressed in these cell types (*Cdx2*, *Hhex*, *Gapdh*) were compared. The two cell types have many differences in posttranslational modifications especially in H3K4-Me. There is as much as a fivefold increase in H3K4-Me in the population of *in vitro* cultured ES cells in regions of *Nanog*. In ES cells, H3K4-Me is present on regions of *Cdx2*, a gene expressed in intestinal epithelium, but is undetected in ICM cells. These results alone clearly indicate that fundamental differences are likely to exist between cultured ES cells and the cells found *in vivo* in the ICM and that the *in vitro* data should not be used blindly to infer *in vivo* states.

How different are ES cells and ASCs? Can information learned from ES cells be used interchangeably with ASCs? ASCs are not as commonly used to study epigenetics as ES cells are due to the small numbers, inaccessibility and difficulties of culturing (Eilertsen et al., 2008). It is becoming increasingly evident that the two types of stem calls are very different. For instance, dissimilarity in the importance of specific genes to stem cell self renewal and maintenance have been identified. Specifically, *Nanog* and *Oct4* are necessary for ES cell maintenance, but are not important for ASCs. *Bmi-1* has been identified as necessary for adult stem cell self renewal but not ES cells (Molofsky et al., 2004).

Planarians are well known for their regenerative capacities. Understanding the biological principles guiding this remarkable attribute is likely to shed light on our understanding of how cellular pluripotentiality may be regulated. We want to study epigenetics in ASCs, and we postulate that planarians are an ideal model system for these studies. These worms can enable studies that circumvent the common issues of ASC studies. For instance, the worm has a large population of neoblasts, or planarian ASCs, that are distributed throughout their body. This allows for *in vivo* studies and permits us to overcome any issues with small cell numbers or artifacts of culturing conditions that is commonplace in other organisms. We also have the ability to study the progression of the stem cell lineage due to the identification of molecular markers for stem cells, stem cell early progeny and the late progeny (Figure 5.3) (Eisenhoffer et al., 2008). An additional benefit of using *S. mediterranea* as a model organism for studying epigenetics is that there is an excellent collection of bioinformatic tools to make many studies possible. These tools include the genome assembly, annotations (Cantarel et al., 2008), genome browser (Robb et al., 2007) and the ability to develop custom bioinformatic tools as needed.

Results

Identification and Placement of Histone Sequences in the Planarian Genome

All eukaryotic organisms have and use highly conserved histone proteins to organize DNA into chromatin (Malik and Henikoff, 2003), but the ways in which these proteins are regulated are different (Marino-Ramirez et al., 2006). How many copies of each of the core histone are present? Which histone variants exist in the genome, and how many? Do histone genes form clusters in the genome? These are the questions that I first wanted to address.

Due to the high degree of amino acid identity for histone proteins across species (Mariño-Ramírez et al., 2005), it was straightforward to identify the large numbers of annotations for all four core histones, variants, and the linker histone, H1, in *S. mediterranea* (Tables 5.1, and 5.2).

The numbers of histones are comparable to numbers I identified in other organisms (Table 5.3). I also searched for histone variants and identified variants of histone H3 and histone H2A. The H3 variant present in the planarian genome is H3.3. This variant replaces canonical H3 during transcription. Both H2A.X and H2A.Z are also present. H2A.X has a serine in the C terminal tail that is phosphorylated in response to double-stranded breaks. H2A.Z, on the other hand, is often found at heterochromatic boundaries, and it is thought that this histone enables the DNA to resist condensation and keeps heterochromatin from spreading into euchromatic regions (Malik and Henikoff, 2003).

During my analysis of histone genes in the planarian genome, I found that many of the genes appear in clusters (Table 5.4) (Figure 5.4), a feature likely to be the result of evolutionary conservation (Rooney et al., 2002) (Figure 5.5). Many species have histones in clusters, though the chromosomal organization is not always the same (Sittman et al., 1981). In *S. mediterranea*, nine histone gene clusters can be identified. The largest cluster is made up of five genes, two H3.3 sequences, two H2B sequences, and one H2A sequence. The other clusters are smaller and are made up of only two genes each. In the examination of the genome, neither H4 nor the linker histone H1 were found to be part of any histone cluster.

Characterization of Histone Posttranslational Modifications in *Schmidtea meditrranea*

My investigation of sequence conservation indicated that many commercially available antibodies generated against epitopes of histone posttranslational modifications in other species should work in planarians. *S. mediterranea* Histone H4 is identical to human H4 and Histone H3 has only two amino acid differences (Figure 5.6) I tested various histone antibodies by Western blots (Figure 5.7) and in whole animals with variable success (Table 5.5). The antibody against H3P was already established to work robustly in whole animals (Figure 5.8A), while the antibodies against H3K9K14-Ac and H2K9-Me3 were not previously tested in planaria. When tested in whole-mount fluorescent immunohistochemistry assays, these two antibodies produced discrete, cell-specific patterns throughout the animals (Figure 5.8B and 5.8C).

I also wished to test how well the identified antibodies could be used in other methodologies employed to study epigenetic regulation. One such method is Chromatin immunoprecipitation (ChIP), a standard technique for ascertaining the presence or absence of specific posttranslational modifications on a given gene. The antibody against H3K9K14-Ac worked particularly well (Figure 5.9). These findings open the possibility of carrying out genome-wide studies for identifying specific histone marks associated with specific gene functions in planarian cells subjected to various experimental conditions (e.g., regeneration, starvation).

The use of mass spectrometry is a very powerful tool for the identification of novel histone posttranslational modifications and characterization of all the modifications present in a purified protein extract (Zhang et al., 2003). Using this methodology can be very informative in identifying modifications or combinations of modifications that may only be present in specific cell types. I generated a protocol for isolation of histone proteins for separation by reverse phase liquid chromatography (RPLC) and identification by mass spectrometry analysis. (Figure 5.10). I encountered insurmountable technical difficulties in the implementation of identification of posttranslational modifications with the Mass Spectrometry Core Facility at the University of Utah. However, this is an avenue of investigation that I would like to continue in the future and given the ease of histone purification afforded by the protocol I have developed, plans have been made for the continuation of these studies at a different facility.

Analysis of Histone H3 Acetylation in Various Cell Types of *Schmidtea mediterranea*

In an attempt to introduce robust and reliable assays to gauge chromatin states in the undifferentiated ASCs and their division progeny, we visualized cell type and histone modifications via immunofluorescence and microscopy. Cells were collected from whole worm dissociations, and the levels of histone H3K9K14-Ac were measured by a combination of immunocytochemistry and fluorescent microscopy. Different cell types were identified by visual morphology and the levels of H3K9K14-Ac were measured. The 39 DAPI (nuclear stain) positive cells, from three experiments, #3 (Figure 5.11), #8 and #11, were imaged and could be categorized into six groups depending on the ratio of the acetylation fluorescence intensity to DAPI intensity (AC/DAPI) (Table 5.6). The majority of the cells were found to have low levels of AC/DAPI (Figure 5.12 A,B, and E-L). However, high levels of AC/DAPI were present in a few cells (Figure 5.12 C,D, and M). This becomes readily apparent when the levels of acetylation versus DNA content (DAPI intensity) are plotted (Figure 5.13). The three outliers are three cells with a very high AC/DAPI ratio (Figure 5.12 C,D, and M).

I also identified cell types with different AC/DAPI ratios but with similar morphologies (Figure 5.14). Each of these cells may be a neoblast due to the large nuclear to cytoplasmic ratio and small cell size (5-8microns). The majority of the neoblast-like cells have a very low AC/DAPI ratio of 0.1.

Immunohistochemistry of Cells with Antibodies Against Various Histones with Posttranslational Modifications

Fluorescence activated cell sorting (FACS) was used to sort stem and differentiated cells from wild type worms in order to determine if these two populations possess different modifications. Sorting these populations is possible because the stem cells are the only dividing cells in asexual planarians and when the worms are exposed to irradiation the stem cells die, effectively substracting them away from the samples. Hence, we use DNA content (>2n) as a general readout to identify which cell population on the FACS plot are stem cells. When irradiated animals are compared to wild type animals, the wild type cells that correspond to the missing irradiated cells are identified. The subset of these cells that possess an increase in DNA content, signifying cell division (>2n) are termed “X1”. A second population of cells, the undifferentiated progeny termed “X2”, are not dividing but are also sensitive to irradiation because they are produced by the stem cells and disappear 4 to 5 days after irradiation. The third population of cells is unaffected by irradiation and are called “Xins”. The differentiated cell types of the planarian are found in this population.

Using FACS, I collected stem and differentiated cells, stained with antibodies, and counter stained with DAPI to label their nuclei. I found that only 13% of stem cells had detectable levels of signal for the antibody against H3K9-

M3, in contrast to 76% in differentiated cells; while H3K27-Me3 was heavily detected in both populations (Table 5.7 and Figure 5.15).

Large Scale Screen of Epigenetic Modifying Enzymes

I wished to explore the roles of epigenetic modifying enzymes in ASC function to determine whether any enzymes are expressed in stem cells, or if any have a role in stem cell maintenance, self-renewal or differentiation. Using annotated protein collections of other species I compiled a list of sequences for epigenetic modifying enzymes, identified homologues in planaria and cloned these genes with ~96% success (Table 5.8). This is notable, as it served to independently validate the bioinformatics generated gene models in the MAKER annotation. I searched for each of these genes independently in the genome sequence and in MAKER annotations and found that every gene but three (*Smed-HDAC5-2*, *Smed-HDAC1-3*, and *Smed-MYST4-1*) of the 87, which were successfully cloned, were present in MAKER annotations.

I then sought to determine the expression pattern for each of the 87 genes. Approximately, 61% or fifty-three of the cloned genes gave us unambiguous expression patterns (Figure 5.16). I performed whole-mount *in situ* hybridization (WISH) and obtained a variety of discrete patterns and an absence of any ubiquitously expressed genes. Some expression patterns are shared by multiple classes of enzymes. For example, I observed a stem cell-like pattern and a nervous system-like pattern (Figure 5.17). The stem cell-like pattern is characterized by the expression of mRNA between the branches of the gut

(Figure 5.17A) and an absence of expression anterior to the photoreceptors and in the pharynx. This pattern was found for a histone deacetylase (*Smed-HDAC1-1*), a histone methyltransferase (*Smed-SETD8-1*), and two putative DNA demethylase (*Smed-NHP2L1 and Smed-RL7A*). The nervous system-like pattern is characterized by mRNA expression in the brain down the ventral cords and at the distal tip of the pharynx. This pattern is found in three of the histone deacetylases (*Smed-HDAC5-1*, *Smed-HDAC8-1* and *Smed-Sin3-2*), five of the histone acetyltransferases (*Smed-CBP-5*, *Smed-CBP-6*, *Smed-Ep300-3*, *Smed-Ep300-1*, *Smed-CBP-4*), two of the histone demethylases (*Smed-JARID-2* and *Smed-JARID-3*) and three of the histone methyltransferases (*Smed-TRR-1*, *Smed-DOT1L-1* and *Smed-SUV92-2*).

I also wanted to use RNAi to disrupt gene function and determine if any of the epigenetic modifying enzymes have a role in stem cell function. There is a characteristic phenotype that arises when a gene that is involved in stem cell function is perturbed which includes regression of the head, curling of the epithelium onto the ventral surface, and no regeneration when amputated (Reddien et al., 2005) (Figure 5.18A). Six genes gave me this phenotype when silenced by RNAi: *Smed-HDAC1-1* (Figure 5.18B), a histone deacetylase, *Smed-SETD8-1* (Figure 5.18C), a histone methyltransferase, and four putative DNA-demethylases, *Smed-RL7A* (Figure 5.18D), *Smed-RP30*, *Smed-NHP2L1*, and *Smed-RPS12*.

Loss of L7Ae Domain-Containing Genes and Their Effects on a Stem Cell Lineage

The four genes *Smed-NHP2L1, Smed-RL7A, Smed-RPS12,* and *Smed-RPL30* were selected as part of the screen because of the possibility that these molecules may play a role in DNA demethylation (Barreto et al., 2007a). These genes all contain the L7Ae domain, which is also found in the Gadd45 family of proteins. The members of this family are Gadd45a, Gadd45b, and GADD45g; the first two are believed to be involved in DNA demethylation (Barreto et al., 2007a; Ma et al., 2009; Sytnikova et al., 2011). Despite being very short genes and mainly only composed of the L7Ae domain, similar to the Gadd45 family, these four planarian genes are very dissimilar in amino acid sequence (Figure 5.19).

Smed-RL7A, *Smed-RPS12*, *Smed-RPL30*, and *Smed-NHP2L1* have a stem cell defective phenotype when the gene function is disrupted and *Smed-NHP2L1* and *Smed-RL7A* have stem cell-like expression patterns. Now that I have identified epigenetic modifying genes that may have a role in stem cell function, I perturbed their function and assayed the resulting effects on a previously characterized planarian stem cell lineage (Figure 5.3) (Eisenhoffer et al., 2008).

The worms were fed bacteria containing dsRNA on days 0, 4 and 7 (Figure 5.20). I amputated the head and the tail on day 10 after the first feeding. I fixed intact worms 3, 7 and 9 days after the first feeding (D3, D7 and D9). D3 and D7 worms had been fed only once, while D9 worms had been fed twice. The cut fragments were fixed on D11 through D17 after the first feeding, also noted as regeneration days 1 through 7 (RD1 through RD7). Worms that were cut on D10 and fixed on D11 had been fed three times. The fixed animals were used for WISH with probes against markers for stem cell lineage analysis, *smedwi-1* (stem cells), *Smed-NB.21.11e* (early progeny) and *Smed-AGAT-1* (late progeny). An antibody, anti-H3P was used to identify dividing stem cells. There is a noticeable reduction in the expression of all four markers over time in each of the four genes of the L7Ae family; however, the order of disappearance of the markers is different.

Smed-RL7A has homology to a gene that encodes the 60S subunit, a component of the ribosome and was found to be expressed in stem cells (Figure 5.16). *smedwi-1* expression levels in *Smed-RL7A(RNAi)* (Figure 5.21) begin to diminish between D7 and D9 and continues to disappear slowly though it is still lowly expressed by D17, or RD7. *Smed-NB.21.11e* expression is affected very early. There is a noticeable decrease as early as D3 and levels are completely absent in RD1 animals. *Smed-AGAT-1* expression appears to begin to decrease by D7 and steadily drops in expression until it is completely absent by RD3. Mitotic figures begin to decrease by D3 and continue to decline until approximately RD3, when there appears to be very few if any anti-H3P-labled cells (Figure 5.21 and Figure 5.22).

By homology, *Smed-NHP2L1* is a member of the H/ACA small nucleolar ribonucleoproteins (snoRNPs) gene family. snoRNPs are involved in various aspects of rRNA processing and modification and are also components of the telomerase complex. In the event that any of the proteins of this family is depleted both 18S rRNA production and rRNA pseudouridylation become impaired (Trahan et al., 2010). *Smed-NHP2L1(RNAi)* animals have a large deficit very early on in the expression of the marker for the stem cells and the early progeny. *smedwi-1* is missing by D9 and *Smed-NB.21.11e* is almost entirely absent by the same day. *Smed-AGAT-1* expression and mitotic figures begin to decrease by D7 (Figure 5.23). The mitotic figures continue to decline until about RD3 when there appears to be very few if any anti-H3P-labled cells (Figure 5.23 and Figure 5.24).

Smed-RPS12 is homologous to *RPS12,* a gene that encodes a ribosomal protein that is a component of the 40S subunit. The protein belongs to the S12E family of ribosomal proteins. Outside of the other Gadd45 family members *Gadd45a* has the most similarity to the S12 family of ribosomal proteins (Koonin, 1997) (Figure 5.25). To determine the effects of the loss of *Smed-RPS12* on the stem cell lineage, *Smed-RPS12(RNAi)* animals were fixed and as for the genes described above, WISH was performed with probes that are markers for stem cells, early progeny, and late progeny (Figure 5.26). *Smed-RPS12(RNAi)* animals have an increase in *smedwi-1* expression followed by a marked decrease. The increase in *smedwi-1* is paired with an increase in mitotic figures (Figure 5.27) and a dramatic decrease in *Smed-NB.21.11e* expression. *Smed-AGAT-1* shows a decrease in expression beginning at D9.

The last L7Ae domain-containing gene tested (*Smed-RPL30*) is homologous to RPL30, a gene encoding a ribosomal protein that is a component of the 60S subunit. The protein belongs to the L30E family of ribosomal proteins.

In the stem cell lineage analysis, I found that *Smed-RPL30(RNAi)* animals have a notable decrease in all four markers (Figure 5.28). However, this decrease is preceeded with a slight but quantifiable increase in the number of mitotic figures at D3 followed by a decrease by D7 (Figure 5.29). This increase is paired with a slight increase in *smedwi-1* expression visible at D7 and expression begins to decrease at D9. A decline can be observed starting at approximately D7 for *Smed-NB.21.11e*. *Smed-AGAT-1* expression begins to become depleted at D9.

Stem Cell Lineage Analysis on *Smed-HDAC1-1(RNAi)* Animals

Smed-HDAC1-1 is the only histone deacetylase of the fourteen that I cloned, which displays a stem cell expression pattern (Figure 5.30A and 5.31B). It is also the only deacetylase that initiated a stem cell defective phenotype (Figure 5.18). I wanted to determine what kind of effect the loss of *Smed-HDAC1-1* has on the stem cell lineage. I performed RNAi feedings in the same manner as with the putative DNA demethylases and fixed intact animals on D4, D7, D10, D12, D15 and regenerating worms at RD1 through RD6. These worms were examined using WISH and immunostaining with stem cell lineage markers and a mitosis marker. The RNAi was very effective in reducing the expression of the Smed-HDAC1-1 (Figure 5.30C).

In *Smed-HDAC1-1(RNAi)* animals *smedwi-1* expression begins to decrease by D7. There is a detectable reduction by D10 of *Smed-NB.21.11e* followed by a decrease at D12 of *Smed-AGAT-1* (Figure 5.31). *smedwi-1* is almost completely depleted in RD2 animals, *Smed-NB.21.11e* by RD4 and

Smed-AGAT-1 by RD6 (Figure 5.32). There is a quantifiable reduction in mitotic figures by D10 (Figure 5.33).

To determine more precisely the affect that the loss of *Smed-HDAC1-1* is having on the stem cell lineage, I dissociated the *Smed-HDAC1-1(RNAi)* worms for FACS and counted the numbers of stem cells, early progeny and late progeny. I used D12 worms and discovered that stem cells were still present (Figure 5.34). This day was selected because the animals appear healthy yet the stem cell marker is severely depressed, the early progeny marker has started to disappear, and the late progeny marker is beginning to disappear.

Stem Cell Lineage Analysis on Smed-SETD8-1(RNAi) Animals

SETD8 is a methyltransferase that is responsible for the methylation of H4K20 and is involved in transcriptional silencing (Peterson and Laniel, 2004). It has been shown to be required for S phase progression (Huen et al., 2008). H4K20-Me is localized to planarian stem cells (Guo et al., 2006). *Smed-SETD8-1(RNAi)* animals have a slight reduction in *smedwi-1* by D4. There is a detectable reduction by D7 of *Smed-NB.21.11e* followed by a decrease at D10 of *Smed-AGAT-1*. (Figure 5.35) There is a significant reduction in mitotic figures as early as D4 (Figure 5.36).

Discussion

From the studies that I have done, it is evident that planarians provide a unique opportunity and an unexplored paradigm for studying epigenetics in ASC.

Planarians have the core histones, histone variants and many of the widely studied posttranslational modifications. They also possess experimentally accessible ASC and methodologies for studying their biology *in vivo*. It is also clear that due to the large numbers of core histones, the presence of variants, and the existence of histone clusters that there is intricate regulation of histone gene expression. This is likely an interesting avenue of study in planarians especially in ASC and differentiated cell types.

For example, analyses of H3K9K14-Ac, H3K9-Me3 and H3K27-Me3 in cells show very promising results for identification of specific cell populations and specific stages of the cell cycle. Examination of H3K9K14-Ac in a mixed cell population indicates that cells with the visual characteristics of stem cells have different levels of acetylation (Figure 5.14). Also a small percentage (13%) (Table 5.7 and Figure 5.15) of sorted stem cells have high levels of H3K9-Me3. This is interesting because we can label the stem cells as they transition from G2 to M with anti-H3P (Van Hooser et al., 1998), and my new findings may allow us to further separate the mixed-staged cycling stem cell population by accessing H3K9K14-Ac (Figure 5.14) and H3K9-Me3 levels. Additionally, a combination of histone mark detections will allow for higher resolution studies of cell cycle dynamics. This is because specific modifications occur during specific cellular events. For example, H3K9K14-Ac is present in areas of the genome that are being actively transcribed (Berger, 2007). Therefore when levels of this modification are high, it should indicate times during the cell cycle when many genes are being transcribed. Transcription increases from G1 through S and peaks during G2 followed by an almost complete cessation of transcription during mitosis (White et al., 1995). H3K9-Me3 is found in areas of the genome that are transcriptionally silent (Berger, 2007), and its levels have been shown to increase during mitosis (Park et al., 2011). Use of antibodies against these three sets of modifications, H3P, H3K9K14-Ac and H3K9-Me3 should make it possible to identify cell cycle stage specific stem cells.

With the bioinformatic tools and a basis and methods for using planarians to study epigenetics, I wanted to create a library of clones of epigenetic modifying enzymes in which I could determine if they have roles in ASC function. The screen resulted in expression patterns for genes in many different classes of epigenetic modifiers. Two common patterns are a nervous system and stem cell-like pattern. Four genes resemble a stem cell-like pattern, *Smed-HDAC1-1* a histone deacteylase, *Smed-SETD8-1* a histone methyltransferase, and two DNA demethylases *Smed-NHP2L1* and *Smed-RL7A*. The other common expression pattern was a nervous system-like pattern and was found in thirteen genes of four enzyme classes. When the function of each of the genes in my library was perturbed, six genes in three different epigenetic modifying classes produced a stem cell-like deficient phenotype. Four are the same genes that gave stem cell-like expression patterns and the remaining two are in the same family as *Smed-RL7A* and *Smed-NHP2L1*, putative DNA demethylases.

The number of genes with expression patterns in the nervous system may be due to the numerous roles that epigenetics play in the nervous system. It has been shown that epigenetic regulation of gene expression is indispensible for neural lineage differentiation, memory and learning (Feng et al., 2007; Taniura et al., 2007). Therefore, changes in the epigenetic state may contribute to phenotypes associated with neural aging and neurodegenerative disorders, such as Alzheimer's disease (Peleg et al., 2010; Rumbaugh and Miller, 2011; Stilling and Fischer, 2011) It would be interesting to determine if the activities detected in the planarian nervous system may reflect similar changes and serve, perhaps, as experimental paradigms to study these processes.

Smed-NHP2L1, Smed-RL7A, Smed-RPS12, and *Smed-RPL30* have homology to ribosomal proteins and were selected as part of the screen due to the potential that they are DNA demethylases. These genes all contain the L7Ae domain, which is also found in DNA demethylases (Koonin, 1997; Barreto et al., 2007b; Ma et al., 2009; Sytnikova et al., 2011). It is not uncommon for ribosomal proteins to have extraribosomal functions (Wool, 1996; Warner and McIntosh, 2009) but it still remains to be determined if the planarian L7Ae domain-containing genes are DNA demethylases. However, when the function is perturbed for the members of the L7Ae family of genes they each have very interesting and different effects on stem cell lineage progression (Figure 5.37).

Smed-RPL30(RNAi) has the canonical, if not expected effect on the lineage. Under normal conditions, the stem cells divide with some of the daughter cells moving on to differentiate into early progeny, and the early progeny moving on to differentiate into late progeny, which in turn are thought to go on to become a variety of differentiated cell types. If the function of the stem cell is affected, such as their ability to maintain their numbers, detectable reductions in the markers for both the early and late progeny will become readily apparent. Hence, the most parsimonious explanation for the observed results in *Smed-RPL30(RNAi)* animals is that a disruption at the top of the lineage occurred: the stem cells are unable to replenish the early progeny (disappearance of *smedwi-1* signal), the early progeny are no longer made and become depleted as they go on to become late progeny, and likewise the late progeny population cannot be maintained as they continue to differentiate.

In marked contrast to *Smed-RPL30*, the other three L7Ae family members have different effects on the lineage when their function is perturbed. *Smed-RL7A(RNAi)* animals have a slow reduction in *smedwi-1* and *Smed-AGAT-1*, but an almost immediate deficit of *Smed-NB.21.11e*. This might indicate a role in the routing of the stem cell decision to become early progeny expressing *Smed-NB.21.11e* and therefore when *Smed-RL7A* is missing, the stem cell progeny are funneled into a separate, as yet undetermined lineage responsible for the generation of a different set of early progeny cells. Hence, it is likely that a detailed characterization of *Smed-RL7A(RNAi)* stem cells with halogenated thymidine analogs, followed by sorting and global gene expression patterns either by deep sequencing or microarrays will help uncover a new developmental lineage in these organisms.

Smed-RPS12(RNAi) animals display an equally interesting lineage defect. Abrogation of *Smed-RPS12* results in an increase in the stem cell marker *smedwi-1*, followed by its dramatic reduction. There is an almost immediate absence of the early progeny marker *Smed-NB.21.11e* and a gradual loss of

Smed-AGAT-1, even though the late progeny seem to be initially unaffected by the loss of *Smed-RPS12*. This could be explained by a misregulation of stem cell specification, such that the resulting daughter cells after cell division fail to commit to an early progeny fate which then may be followed by either their death or by their differentiation into cells for which we have not yet identified specific markers.

RNAi of *Smed-NHP2L1* (the remaining L7Ae domain containing gene tested) produced a rapid depletion of *smedwi-1* and *Smed-NB.21.11e* and a more gradual reduction in *Smed-AGAT-1*. By homology this gene is a member of the H/ACA small nucleolar ribonucleoproteins (snoRNPs) gene family. snoRNPs are involved in various aspects of rRNA processing and modification and are also components of the telomerase complex. In the event that any of the proteins of this family is depleted, both 18S rRNA production and rRNA pseudouridylation are impaired (Trahan et al., 2010). *Smed-NHP2L1* may be playing an important role in the translation of mRNAs in stem cells and the resulting postmitotic early progeny or some other basic function of these two cell types, but not in the late progeny.

At this point in time, it is unknown what exactly is happening to the cells in which the markers are declining. Are the cells dying or are they differentiating? If they are differentiating, are they making the proper cell fate decisions? Additional experimentation is required to determine the answers to these questions, such as, cell death assays, RNAi of these genes followed by FACS, pulse-chase studies with halogenated thymidine analogs, and WISH with additional lineage markers.

Another molecule causing marked defects when silenced by RNAi is *Smed-HDAC1-1*, a class I histone deacetylase. Class I histone deactylases are usually found in the nucleus, are expressed in most tissues and cell lines in humans (Verdin et al., 2003) and cause embryonic lethality when eliminated in mice (Dovey et al., 2010). The HDAC1 homolog in *C. elegans*, *hda-1*, is also ubiquitously expressed and it too results in embryonic lethality when mutated in these animals (Dufourcq et al., 2002). HDAC inhibitors have been shown to result in cell cycle arrest in cancer cell lines (Tonelli et al., 2006). In planaria, *Smed-HDAC1-1* is not ubiquitously expressed and the loss of this protein is likely causing cell cycle arrest. In *Smed-HDAC1-1(RNAi)* animals the stem cell marker decreases first, followed by a concomitant decrease in the early progeny first, and the late progeny second. Closer examination of the stem cells by FACS (Figure 5.34) at D12, a day in which the stem cell marker was almost completely absent, showed that the stem cells are in fact still present in normal numbers. This is accompanied by a loss of mitotic figures, measured by the absence of anti-H3P labeling (Figure 5.31). This indicates that the loss of *Smed-HDAC1-1* causes the cell cycle to arrest. Since there is a loss of phosphorylation of the H3 serine 10, the cell cycle is not progressing from G2 to M. And because one of our criteria for identifying stem cells by FACS is an increase in the concentration of DNA content (>2n), the cells are likely to be arrested somewhere between G1 and G2, or in one of the phases of mitosis. *Smed-HDAC1-1* is likely responsible for the regulation of genes that are required for the decision of stem cells to self-renew and/or to differentiate. With the loss of *Smed-HDAC1-1*, the stem cells are halted; they are no longer dividing and no longer differentiating into early progeny. The lack of stem cell differentiation is proposed because they are not being depleted even as there is a subsequent reduction in the expression of *Smed-NB.21.11e* and *Smed-AGAT-1*, presumably due to their differentiation or death.

Smed-SETD8-1 is a homolog of human *SET8* and *D. melanogaster pr-set7*. This protein is responsible for the methylation of H4K20 and is involved in transcriptional silencing (Peterson and Laniel, 2004). SET8 in humans is required for S phase progression and in mice and flies its loss results in embryonic lethality (Nishioka et al., 2002; Huen et al., 2008). H4K20-Me is localized to planarian stem cells (Guo et al., 2006). *Smed-SETD8-1* has a stem cell-like expression pattern (Figure 5.16), and *Smed-SETD8-1(RNAi)* animals have the characteristic stem cell deficient phenotype (Figure 5.18C). It is likely that the loss of *Smed-SETD8-1* causes the stem cells to be unable to enter or progress into the S phase of the cell cycle. This would result in insufficient numbers of stem cells to maintain the lineage.

My aims were to determine if any epigenetics modifying enzymes are localized to ASCs or if any of these enzymes have specific roles in ASC function. Six genes were identified that are likely to have key roles in stem cell self-renewal, maintenance and/or differentiation; *Smed-HDAC1-1, Smed-SETD8-1, Smed-RL7A, Smed-NHP2L1, Smed-RPL30* and *Smed-RPS12*. Future studies will attempt to uncover a more in-depth understanding of the specific roles these genes have in stem cell fate decisions.

Methods

Identification and Placement of Histone Sequences in *Schmidtea mediterrana*

I used BLAST (Altschul et al., 1990) and custom perl scripts that implement BioPerl (Stajich et al., 2002) to identify the histones in the *Schmidtea mediterrana* assembly and MAKER annotations. Histone sequences from other species were acquired from http://research.nhgri.nih.gov/histones/ (Mariño-Ramírez et al., 2005). In summary I performed the following steps (all scripts mentioned can be found in the Appendix: Custom Scripts):

1. For each of the four histones I created a fasta file of sequences from the histone database, each named accordantly in the following format, H1.histoneDB.fasta.
2. I compared all open reading frames of the planarian genome and MAKER amino acid annotations to each histone fasta file using BLASTP.

   ```
   blastall -p blastp -i smed_mk4.aa -d "H1.histoneDB.fasta" -o
   H1.histoneDB.fasta.blast.out
   ```
3. For each resulting blast result file I converted the standard output to a tabular format using a script named blast2table_desc.pl that I modified from blast2table.pl (Korf et al., 2003).

```
blast2table_desc.pl -e 1e-10 H1.histoneDB.fasta.blast.out >
H1.histoneDB.fasta.blast.b2t
```

4. I parsed each resulting blast2table file for the top hits using a custom script that I created called parseBlastTable_topHit.pl.

```
parseBlastTable_topHit.pl H1.histoneDB.fasta.blast.b2t > H1.tophit
```

5. From each file containing top hits I created a new file that contains only the S.med sequence name and the histone top hit description.

```
awk ‘{print $1}’ H1.tophit > mk4.H1.hits
```

6. I then collected the amino acid sequence for each planarian histone in histone specific files.

```
fastacmd -d /common/data/smed_maker_v4.aa -i mk4.H1.hits >
mk4.H1.fasta
```

7. I then formatted each planarian histone file such that it could function as a blast database.

```
formatdb -i mk4.H1.fasta -p T -o T
```

8. I searched for redundant sequences by first blasting each new planarian histone database against itself.

```
blastall -p blastp -i mk4.H1.fasta -d mk4.H1.fasta -o mk4.H1.redundant
-F F
```

9. The BLAST results from the redundant search were parsed by a custom script that identifies the redundant matches.

```
parseBlast_redundantSearch.pl mk4.H1.redundant > H1.copies
```

10. I then made a fasta file of the unique planarian histone sequences.

```
fastacmd -i mk4.H1.redundant.summary.redundantSearch.out -d
mk4.H1.fasta > H1.unique.fasta
```

Histone Counts in Other Organisms

I used TBLASTN (Altschul et al., 1990) to identify histones from each of the five HistoneDB fasta fies in the genomic reference sequences for *C. elegans* (Database: C elegans genome release WS189, 7 sequences; 100,281,426 total letters), *D. melanogaster* (Database: Drosophila melanogaster genome (reference only) 6 sequences; 120,381,546 total letters), and *H. sapiens* (Database: human build 36.3 reference and alternate assembly genomic Scaffolds; 11,546 sequences; 8,709,484,211 total letters). I parsed out and counted each unique HSP with a hit that had an e-value of 1e-15 or better.

Chromatin Immunoprecipitaion

I homogenized worms in 200ul of Calcium magnesium free media (CMF) with a pestle and filtered the solution with a 53-micron filter and added 36.5% formaldehyde to a final concentration of 1% and incubated for 15 minutes at room temperature with rotation. Glycine was added to a final concentration of 0.125M to stop the cross linking and incubated 5 minutes at room temperature with rotation. The tube was centrifuged at 700 x g for 4 minutes at 4°C. The pellet was washed two times by resuspending in ice cold Phosphate-buffered saline (PBS) with 1ug/ml protease inhibitor (Sigma P2714). The homogenate was pelleted as before and the nuclei were resuspended in 200ul of nuclei lysis buffer (1% SDS; EDTA pH 8.0; 50mM Tris pH 8.1; 1ug/ml protease inhibitor). The nuclei were incubatated on ice 10-20 minutes and were sonicated to shear the DNA into pieces between 1000-500bps (Branson Sonifier 200; duty 90%, output

30-40% with 9 pulses at 10sec each on ice with 1 minute incubation on ice between each pulse). The tube was centrifuged for 10 minutes at 13,000rpm at 4^{o}C. The supernatant was collected and split into two 1.5ml centrifuge tubes (tube 1 for antibody, tube 2 for no antibody control). ChIP dilution buffer (0.01% SDS; 1.1% Triton x-100; 16.7mM Tris pH 8.1; 167mM NaCl; 1ug/ml protease inhibitors) was added to to each tube so final volume was 10-fold the starting volume. To reduce nonspecific background, pre-clear by adding 80ul of Salmon Sperm DNA/Protein A agarose 50% slurry (Upstate 16-157C) to each tube and incubate 30 minutes at 4^{o}C with agitation. The agarose was pelleted by a brief centrifugation (700-1000 rpm ~1 minutes) and the supernatant was collected into new 1.5ml centrifuge tubes. The antibody was to each 1ml of supernatant and incubated overnight at 4^{o}C with rotation. 60ul of Salmon sperm DNA/Protein A Agarose Slurry was added and incubated for 1 hour at 4^{o}C with rotation to each tube. The agarose was pelleted by gentle centrifugation (700-1000 rpm ~1 minutes) and the supernatant was removed (contains the unbound DNA). The agarose/antibody/histone complex was washed 3-5 minutes on a rotating platform at 4^{o}C with 0.5ml of each of the following cold solutions Low Salt Immune Complex Wash Buffer (0.1% SDS; 1% Triton x-100; 2mM EDTA pH 8.0; 20mM Tris pH 8.1; 150mM NaCl), High Salt Immune Complex Wash Buffer (0.1% SDS; 1% Triton x-100; 2mM EDTA pH 8.0; Tris pH 8.1;500mM NaCl), LiCl Immune Complex Wash Buffer (0.25M LiCl;1% NP-40; 1% deoxycholic acid (w/v); 1mM EDTA pH 8.0; 10mM Tris pH 8.1). Two final washes with TE Buffer (1mM EDTA pH 8.0; 10mM Tris pH 8.1) were performed.

Nuclei Enrichment

Worms were homogenized in cold PBS with a pestle and spun at 250 x g for 5 minutes at 4^{o}C. The pellet was resuspended in 200ul of cell lysis buffer (5mM PIPES pH 8.0; 85mM KCl; 0.5% NP-40; 1ug/ml protease inhibitors) and incubated on ice for 5 minutes. The mixture was gently agitated and incubated for an additional 5 minutes on ice. The nuclei were pelleted by spinning at 5,000 rpm for 5 minutes at 4^{o}C.

Western Blotting of Histones

Western blotting of histones was preformed as previously described (Thiriet and Albert, 1995).

Histone Extraction

Whole worms were homogenized in CMF with a pestle, spun down at 250 x g for 5 minutes, resuspended in lysis buffer (0.01M Tric-HCl pH 7.5; 0.001M MgCl2; 0.5% NP40) and spun down at 10,000 x g for 3 seconds. The pellet was resuspended in 200ul of extraction solution (0.5M HCl; 10% glycerol; 0.1M 2-Mercaptoethylamine-HCl) and centrifuged at 10,000 x g for 5 minutes. The supernatant containing the histones and was collected.

Histone Separation with Reverse Phase HPLC

Histone extracts were purified by reverse phase HPLC with a C18 column (Vydac). The gradient was composed of mobile phase A, 0.1% Trifluoroacetic acid (TFA) in water and mobile phase B, (0.1% TFA in acetonitrile. Mobile phase B was linearly increased 1% per minute from 35-55%.

Immunocytochemistry, Imaging and Image Processing for Unsorted Cell Populations

S. mediterranea were dissociated by dicing 20 worms as finely as possible and the pieces were nutated in CMF plus trypsin for 1 hour at room temperature. The resulting cells were filtered using a sterile 53 micron filter and pelleted by centrifuging at 250 x g for 5 minutes. The pellet was resuspended in 2 mls of CMF and spotted on microscope slides.

The cells were fixed in 4% formaldehyde in 1x PBS for 15 minutes, rinsed and permeabilized in 0.25% triton X100 in PBS for 5 minutes, and blocked with 4% bovine serum albumin (BSA) in PBS for 1 hour followed by a PBS wash. The cells were incubated in a humidity chamber for 2 hours with a primary antibody. The slides were rinsed and incubated, as above, for 1 hour with secondary antibody. The slides were washed and Vectasheild with DAPI was applied and covered with a glass coverslip.

The stained cells were visualized on an Olympus IX70 fluorescent microscope at 60x magnification and imaged with a CCD camera. Image J and Adobe Photoshop were used to analyze the images. In Image J, the 16-bit images were first converted to 8-bit then DAPI fluorescence, which labels the nucleus, was adjusted by pixel threshold and the resulting binary image was subtracted from the original DAPI image and the image of H3K9K14-Ac fluorescence. The remaining particles, after subtraction, were analyzed, and the intensity of DAPI and H3K9K14-Ac fluorescence were obtained in units of average pixel number per particle, or cell. H3K9K14-Ac fluorescence (antibody staining) and nucleus fluorescence (DAPI staining) are reported in units of the average number of pixels per cell. The level of acetylation per cell is calculated by dividing H3K9K14-Ac fluorescence by the overall nucleus fluorescence.

Immunocytochemistry, Imaging and Image Processing for Sorted Cell Populations

The FACS sorted cells were fixed in 4% formaldehyde in 1x PBS for 5 minutes, rinsed three times in 1x PBS and permeablelized in 0.1% triton X100 in 1x PBS for 5 minutes. The slides were rinsed three times in 1x PBS then blocked with 4% BSA in 1x PBS for 1 hour followed by a 1x PBS wash. The cells were incubated in a humidity chamber for 2 hours with aprimary antibody. The slides were rinsed three times in 1x PBS and incubated, as above, for 1 hour with a secondary antibody. The slides were washed three times and Vectasheild with Dapi was applied and covered with a glass coverslip.

The cells were imaged at the University of Utah Microscopy core facility, on the automated BD Pathway 885 high-throughput confocal imager. I wrote perl scripts to create ImageJ scripts that counted all DAPI and antibody positive cells in my images and to format the output into tab delimited files. These scripts are included in the Appendix: Custom scripts.

Gene Finding and Cloning

For every protein of interest, such as HDAC, I generated a fasta file containing every sequence in Swissprot (Boeckmann et al., 2003) that contained the description I was interested in locating in the *S. mediterranea* genome.

1. grep "|HDAC" swissprot.aa | awk ‘{print $1}’ | awk -F "\|" ‘{print $5}’ > HDAC.swprt.list
2. fastacmd -i HDAC.swprt.list -d swissprot.aa > HDAC.swprt.fasta.aa

I used BLASTX to identify homologous planarian sequences and a custom script, parseBlast_hsp.pl to create the largest contiguous sequence from all blast alignments on the same genomic contig.

Once a reciprocal BLAST with the possible planarian sequences was performed,I used a Primer3 (Koressaar and Remm, 2007) and a custom script, runPrimer3.pl to create primers (Table 5.8). I prepared a 7-day regeneration library (Gurley et al., 2008) and PCR amplified products, purified and cloned into the pPR-T4P vector, which was obtained from J. Rink, and transformed into DH5a. Each clone identity was verified by sequencing. For RNAi experiments, the cloned vector was transformed into HT115.

Whole Mount *in situ* Hybridization and Immunostaining

Probe synthesis and WISH was carried out as previously detailed (Pearson et al., 2009).

The animals for immunostaining were processed exactly as the WISH animals until the application of the antibody (anti-phospho-histone H3 ser10

MC463 rabbit monoclonal antibody from Upstate Cell Signaling Solutions, used at 1:300) to detect mitotic activity (Robb and Sánchez Alvarado, 2002). The animals were incubated overnight at 4^{o}C, washed 6x with 1x PBS with 0.1% triton x100, and incubated with Alexa Fluor 555 goat anti-rabbit 555 (Invitrogen A21429).

Double Stranded RNA interferace (RNAi)

RNAi “soft-serve” food was prepared as described previously (Gurley et al., 2008) with the following modifications: all RNAi constructs for each gene were two times more concentrated. All animals were fed 3 times, on day 0, 4 and 7 and cut on day 10 after the first feeding.

Image Capture and Processing of Immunostaining and *in situ* Hybridizations

Images of immunostained animals and of whole-mount *in situ* hybridization in which NBT-BCIP was used as part of the development procedure were captured on a Zeiss Lumar V12 stereomicroscope using an Axiocam HRc camera.

Fluorescence Activated Cell Sorting (FACS)

The dissociation of planarians, cellular labeling, and isolation of cells by FACS were performed as described previously (Reddien, 2005), but using a Becton Dickinson FACSAria

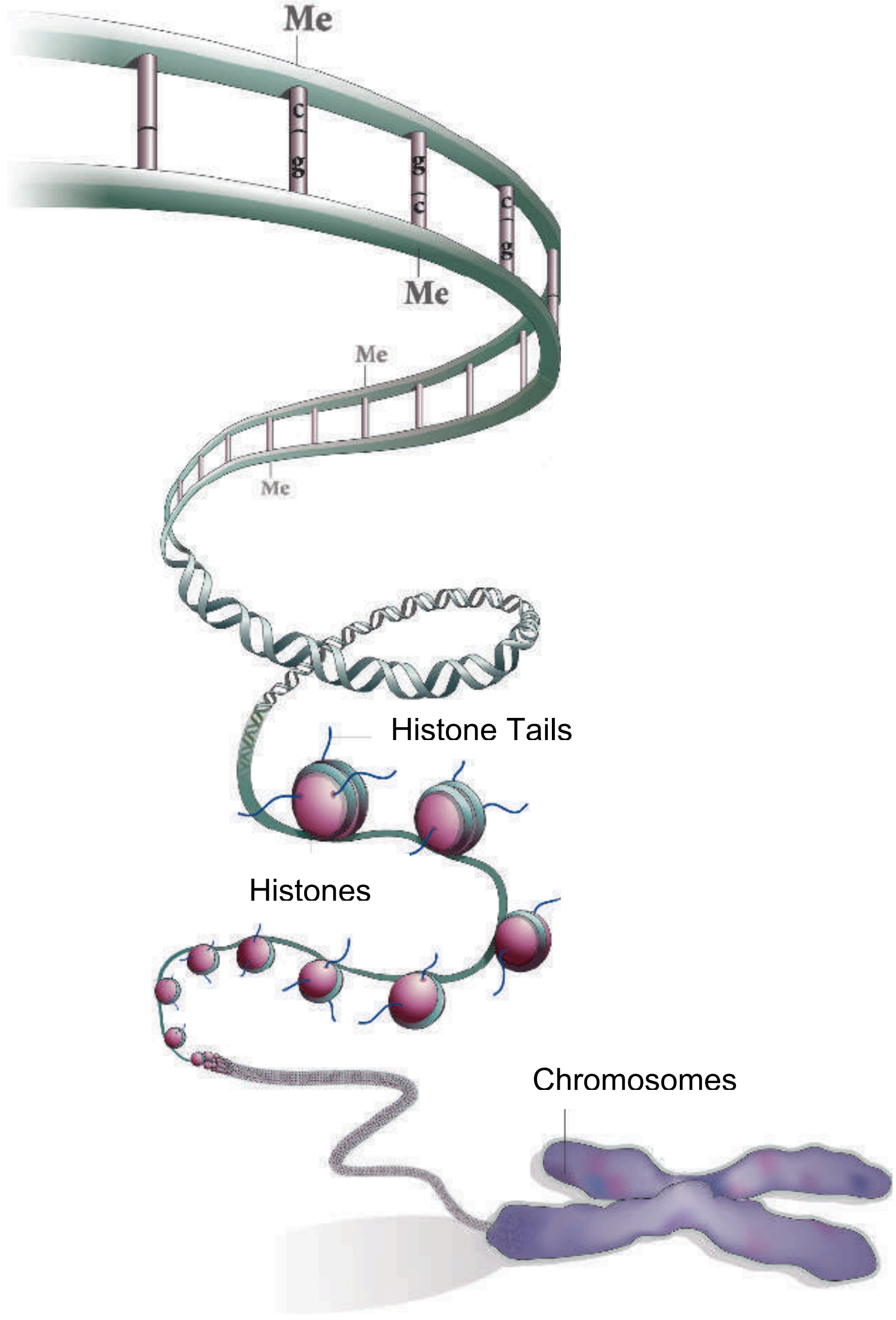


Figure 5.1: The two main components of the epigenetic code, DNA methylation and histone posttranslational modification (Qiu, 2006).

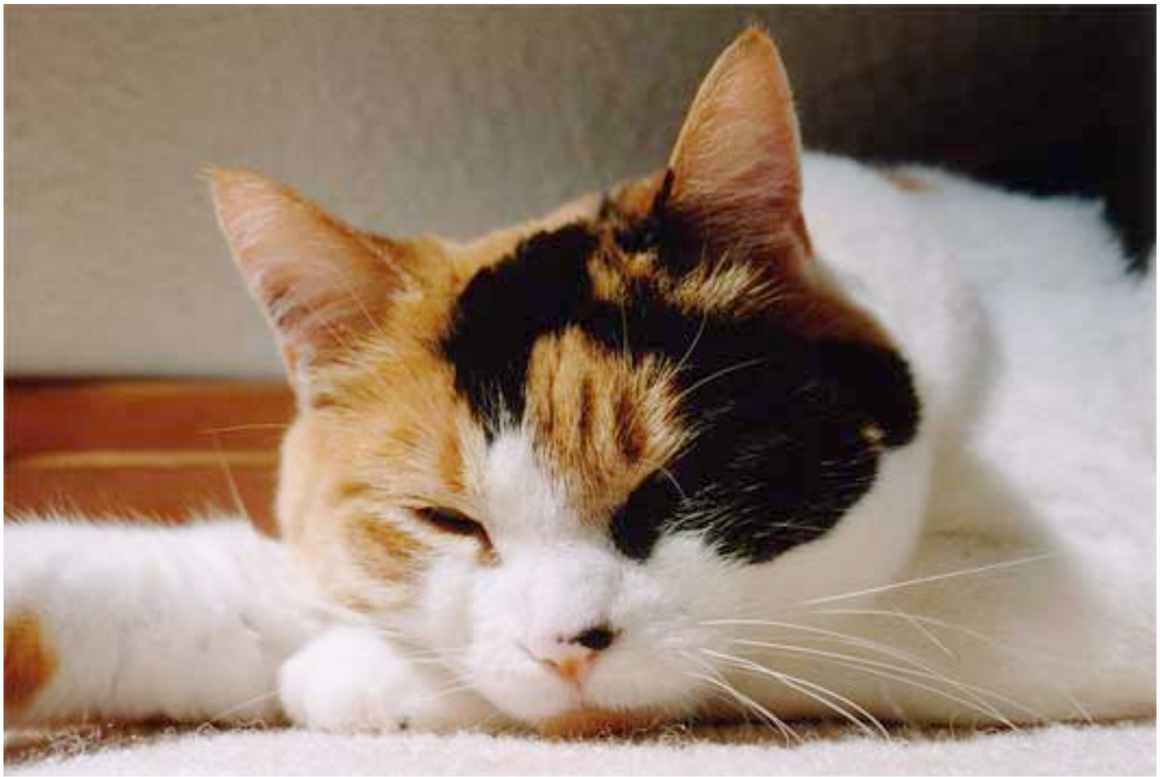


Figure 5.2: The coat coloration of a calico cat is a visible manifestation of X-inactivation (image is from http://www.great-pictures-of-cats.com/calico-cats.html).

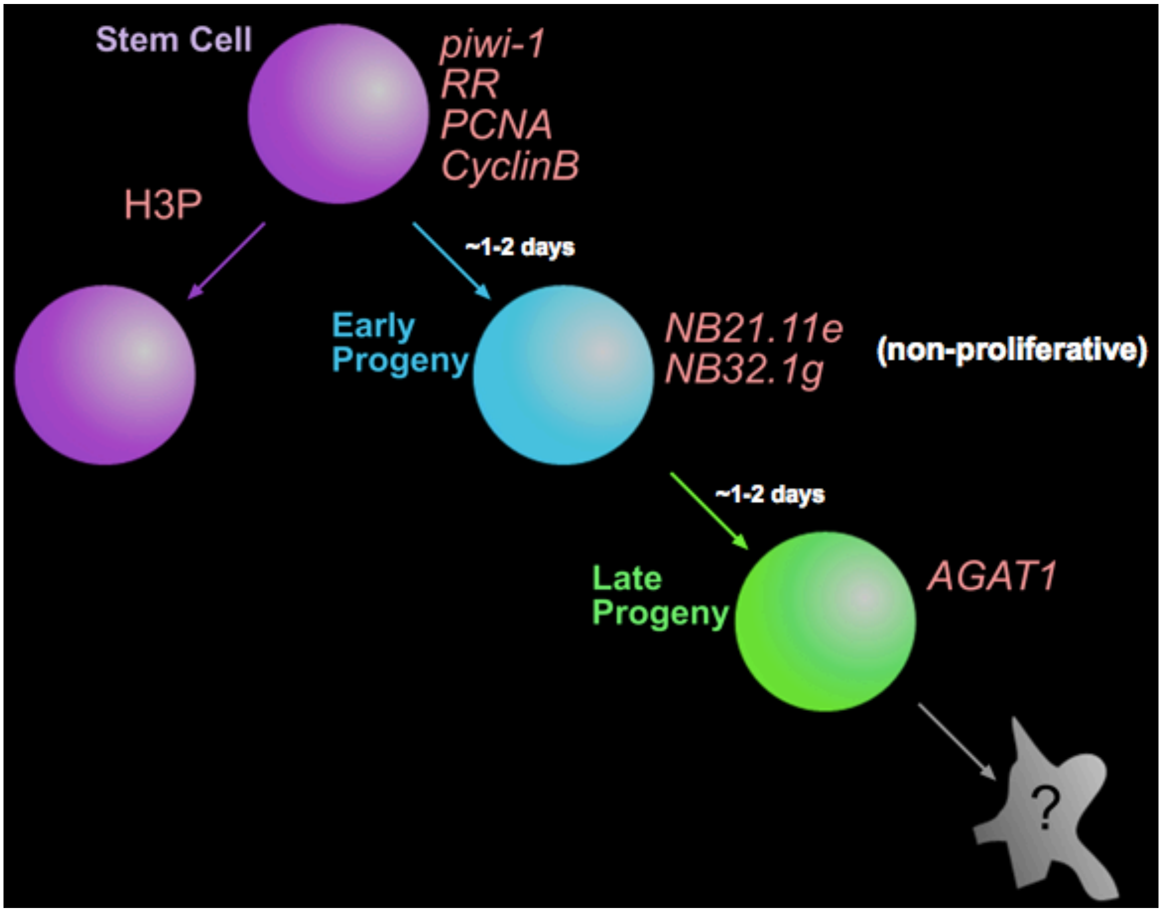


Figure 5.3: Molecular markers for studying stem cell lineage. Stem cells have been found to express the following molecular markers, *piwi-1* (*smedwi-1*), *RR*, *PCNA*, and *CyclinB*. Stem cells, which are the only mitotically active cells can also be labeled with an antibody against H3P as they are entering into mitosis. About one to two days after a stem cell divides one of the daughters differentiates into an early progeny cell. This non-proliferating cell type expresses *NB.21.11e* (*Smed-NB.21.11e*) and *NB.32.1g*. Then about one to two days later this cell type differentiates into a late progeny, which expresses *AGAT-1* (*Smed-AGAT-1*) and it goes on to differentiate into differentiated cell types (used with permission from Dr. Bret Pearson).

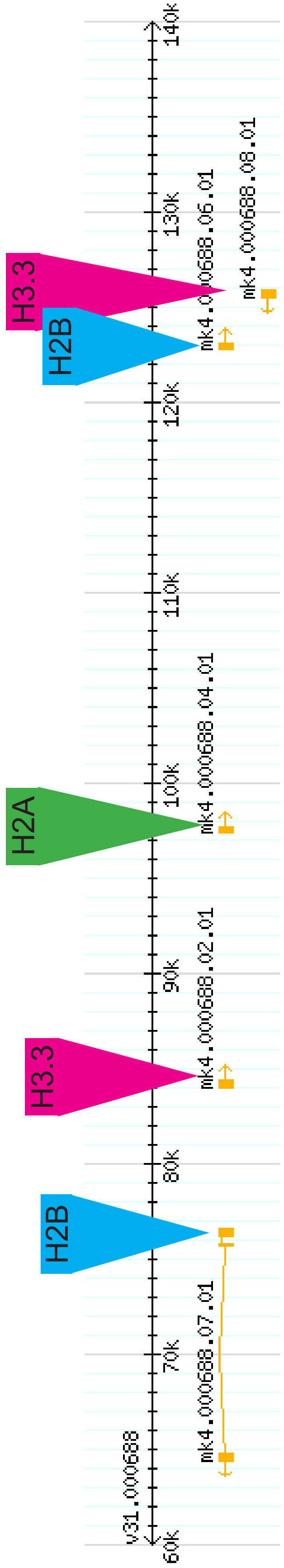

v31.000688
60k
70k
80k
90k
100k
110k
120k
130k
140k
mk4.000688.07.01
H2B
H3.3
mk4.000688.02.01
H2A
mk4.000688.04.01
H2B
H3.3
mk4.000688.06.01
mk4.000688.08.01

Figure 5.4: Largest identifiable *Schmidtea mediterranea* histone gene cluster. This is a screen capture from SmedGD of genomic contig v31.000688. It is the largest identifiable histone cluster and contains two H2B genes, two H3.3 genes, and one H2A gene. Only the first exon of the MAKER transcript mk4.000688.07.01 is the single exon H2B gene.

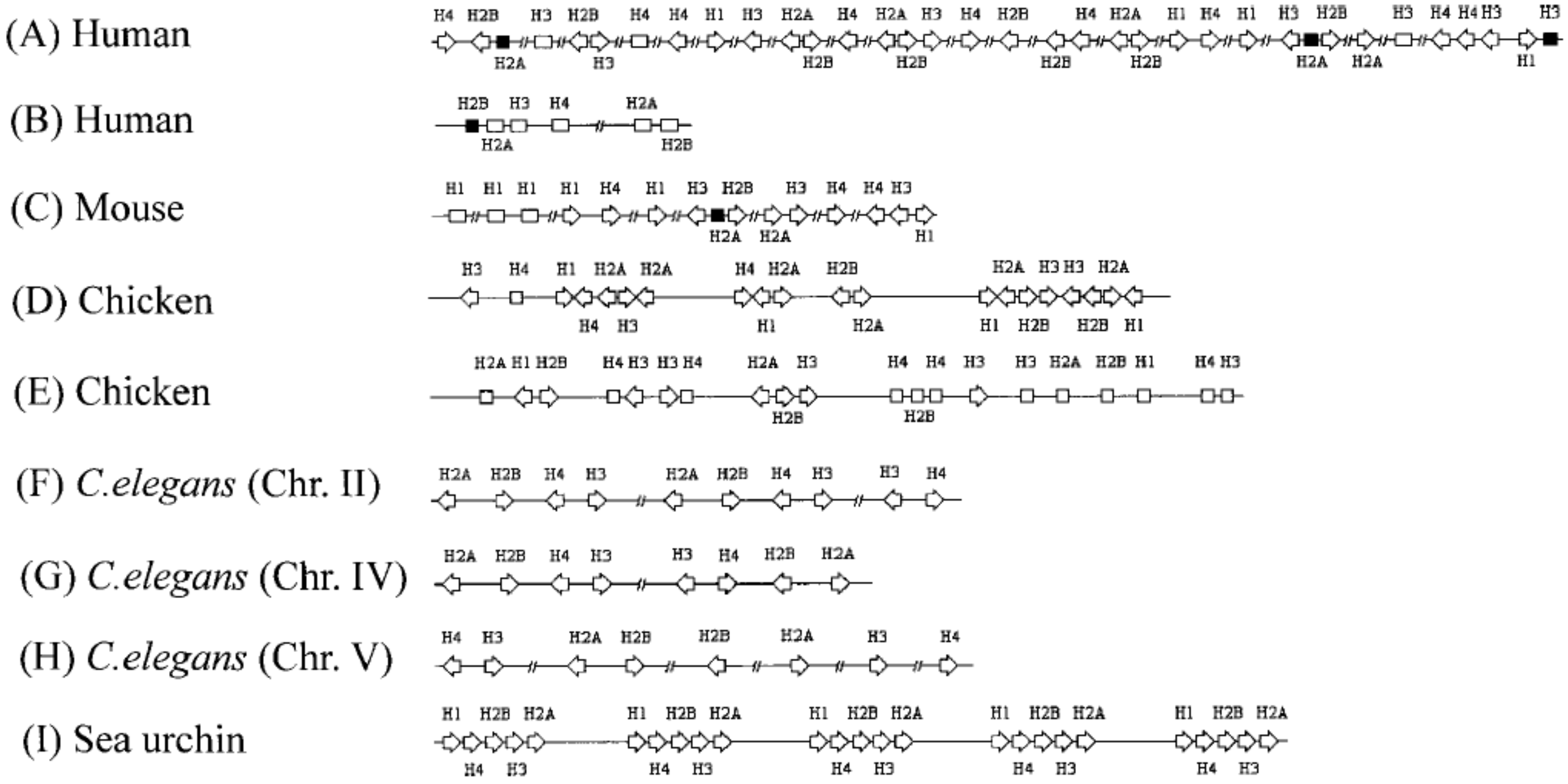

(A) Human
H4 H2B H3 H2B H4 H4 H1 H3 H2A H4 H2A H3 H4 H2B H4 H2A H1 H4 H1 H3 H2B H3 H4 H4 H3 H3
H2A H3 H2B H2B H2B H2B H2A H2A H1
(B) Human
H2B H3 H4 H2A
H2A H2B
(C) Mouse
H1 H1 H1 H1 H4 H1 H3 H2B H3 H4 H4 H3
H2A H2A H1
(D) Chicken
H3 H4 H1 H2A H2A H4 H2A H2B H2A H3 H3 H2A
H4 H3 H1 H2A H1 H2B H2B H1
(E) Chicken
H2A H1 H2B H4 H3 H3 H4 H2A H3 H4 H4 H3 H3 H2A H2B H1 H4 H3
H2B H2B
(F) C.elegans (Chr. II)
H2A H2B H4 H3 H2A H2B H4 H3 H3 H4
(G) C.elegans (Chr. IV)
H2A H2B H4 H3 H3 H4 H2B H2A
(H) C.elegans (Chr. V)
H4 H3 H2A H2B H2B H2A H3 H4
(I) Sea urchin
H1 H2B H2A H1 H2B H2A H1 H2B H2A H1 H2B H2A H1 H2B H2A
H4 H3 H4 H3 H4 H3 H4 H3 H4 H3

Figure 5.5: Comparison of histone clusters in various species. This is a diagram of histone clusters containing histone H3 in a variety of organisms (Rooney et al., 2002). These clusters illustrate the conserved concept of histone clusters for organization but the gene order and direction of transcription (indicated with arrows) is not consistent across species.

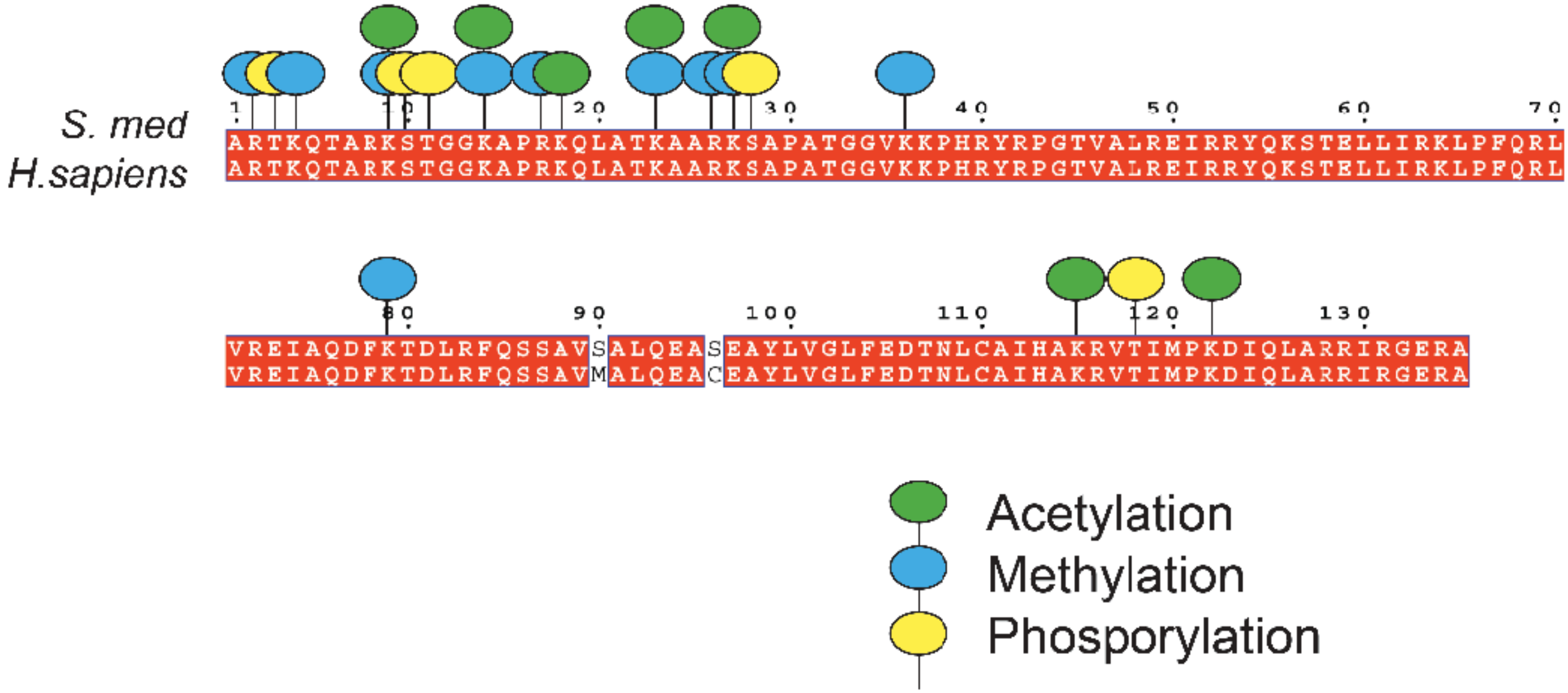

S. med
H.sapiens
1
10
20
30
40
50
60
70
ARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRYRPGTVALREIRRYQKSTELLIRKLPFQRL
ARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRYRPGTVALREIRRYQKSTELLIRKLPFQRL
80
90
100
110
120
130
VREIAQDFKTDLRFQSSAVSALQEASEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA
VREIAQDFKTDLRFQSSAVMALQEACEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA
Acetylation
Methylation
Phosporylation

Figure 5.6: Amino acid alignment of histone H3 in *H. sapiens* and *S. mediterranea*. This is an alignment created with ClustalW (Thompson et al., 1994), Planarian H3 is on top and human below. Locations of select posttranslational modifications are marked with colored lollipops. There are only two amino acid differences between *H. sapiens* and *S. mediterranea*.

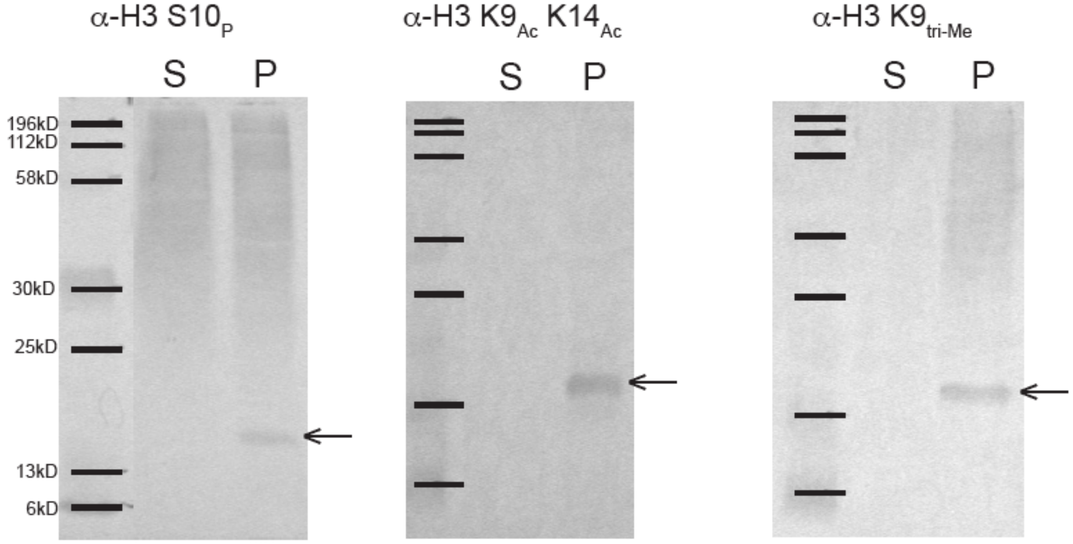


Figure 5.7: Histone antibodies cross react with planarian epitopes. Nuclei were enriched and the supernatant (S) and pellet (P) were run on a 12% SDS polyacrylamide gel, transferred to nitrocellulose and immunostained. Bands of the appropriate size were obtained (arrows).

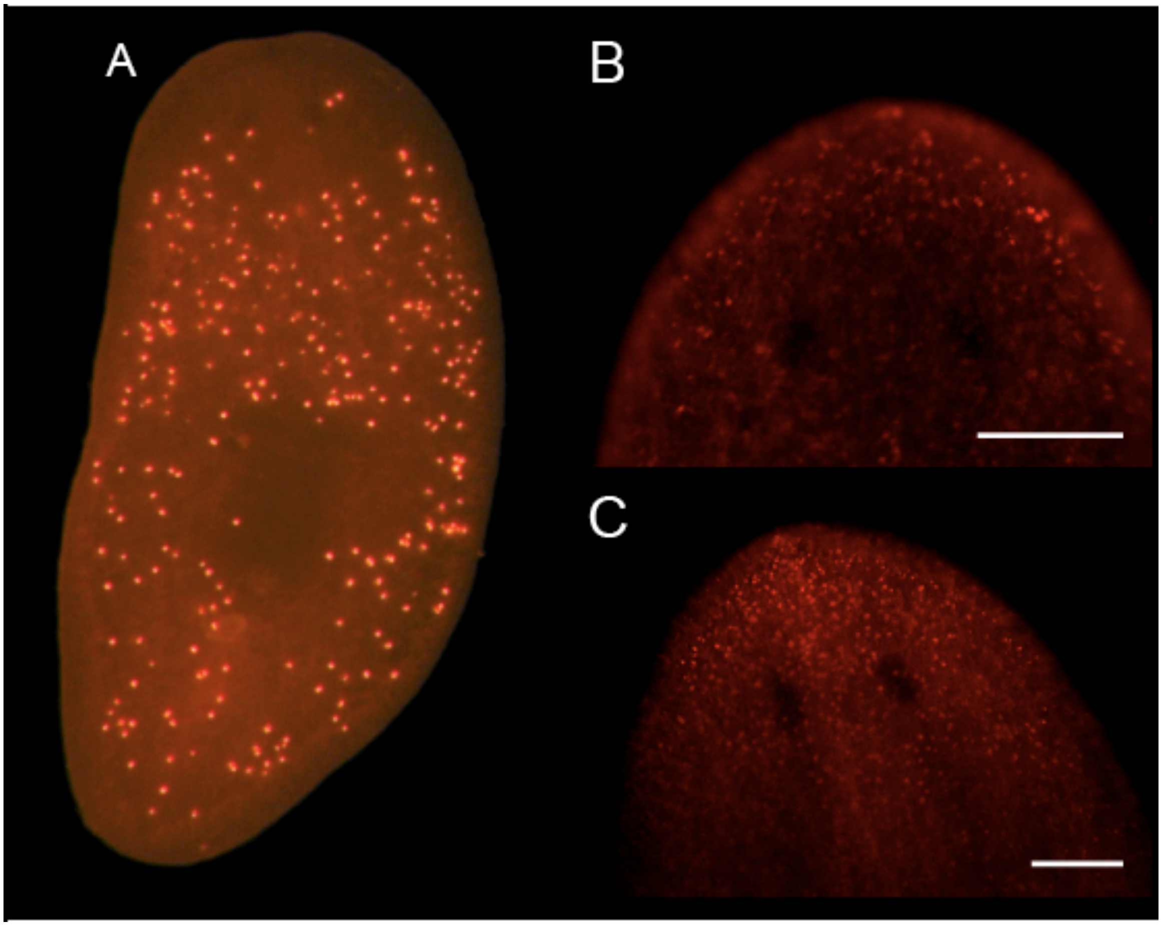


Figure 5.8 Whole-mount immunohistochemistry. (A) Animals immunostained with antibodies against (A) H3P, (B) H3K9K14-Ac, and (C) H3K9-Me3.

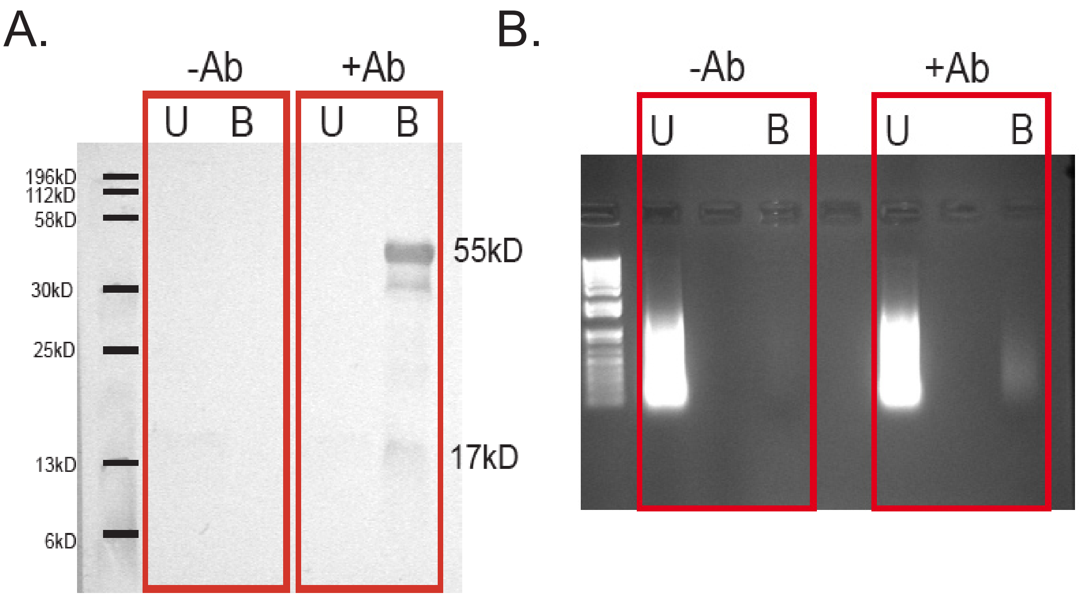
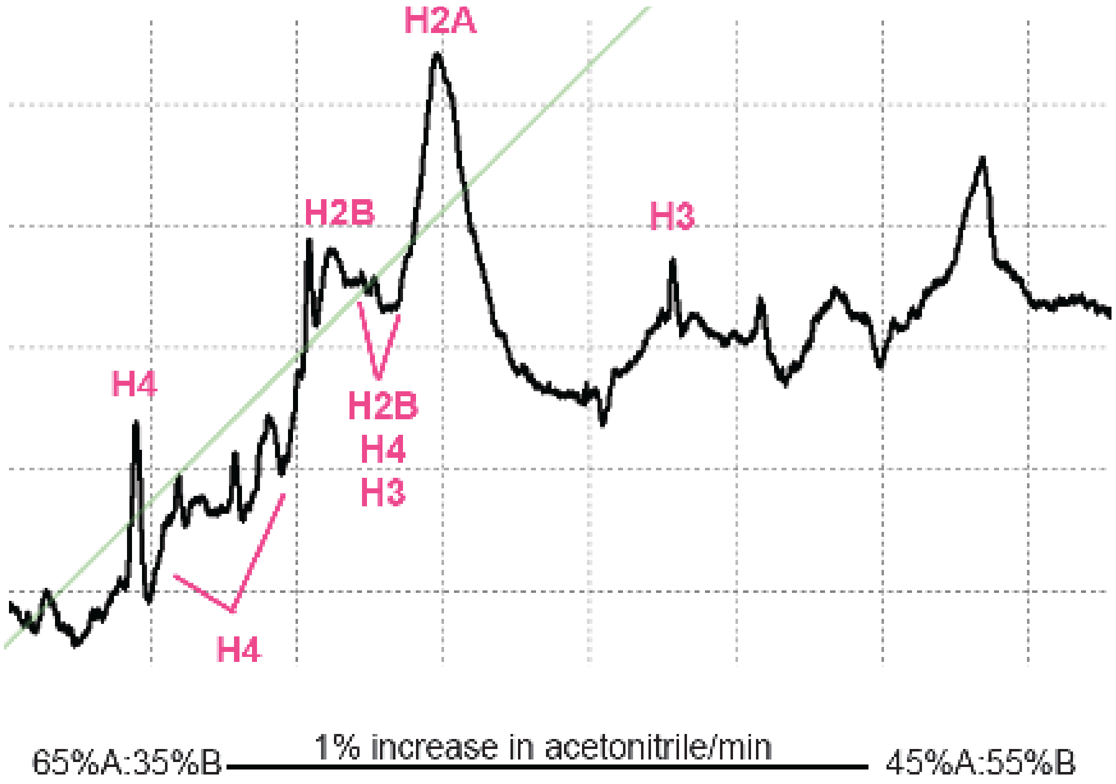
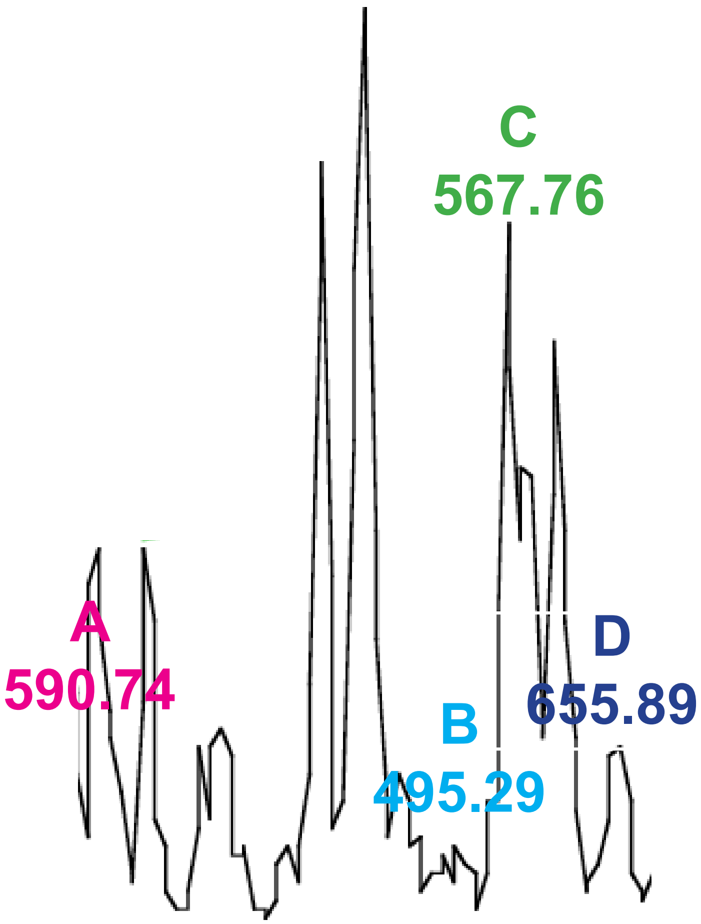


Figure 5.9: ChIP in planaria. An antibody against H3K9K14-Ac was used for ChIP. (A) An immunoblot of the no antibody control (-Ab) and the experimental H3K9K14-Ac pull down (+Ab) with unbound (U) and bound (B) fractions. A 17kD band in the experimental bound (+Ab, B) fraction verifies that the correct protein was precipitated. (B) A DNA agarose gel to verify that DNA had been bound in the experimental pull down in the plus antibody bound (+Ab, B) fraction.

A.


H2A
H2B
H3
H4
H2B
H4
H3
H4
65%A:35%B
1% increase in acetonitrile/min
45%A:55%B

B.



>Smed-HistoneH4

MSGRGKGGKGLGKGGAKRHR
KVLRDNIQGITKPAIRRLAR
RGGVKR**ISGLIYEETR**GVLK
VFLENVIR**DAVTYTEHAK**RK
TVTAMDVVYALKRQGRTLYG
FGG

Figure 5.10: Purification and identification of histone proteins. (A) RPLC was conducted with mobile phase B (0.1% trifluoroacetic acid, 90% acetonitrile) initially set at 35% and linearly increased to 55% over a 20 minute period (1% change per minute). The flow rate was held at a constant 4ml/min. (B) Histone H4, from a RPLC fraction, was identified by MS by the masses of four trypsin digested fragments, indicated in the mass charge chromatogram by peaks A,B,C and D. Peak A (pink) with a mass of 590.74(+2) Da, peak B (light blue) 495.29(+2) Da, peak C (green) 567.76(+2) Da, and peak D (dark blue) 655.89(+2) Da positively match the predicted masses of four trypsin digested peptides of Smed-HistoneH4.

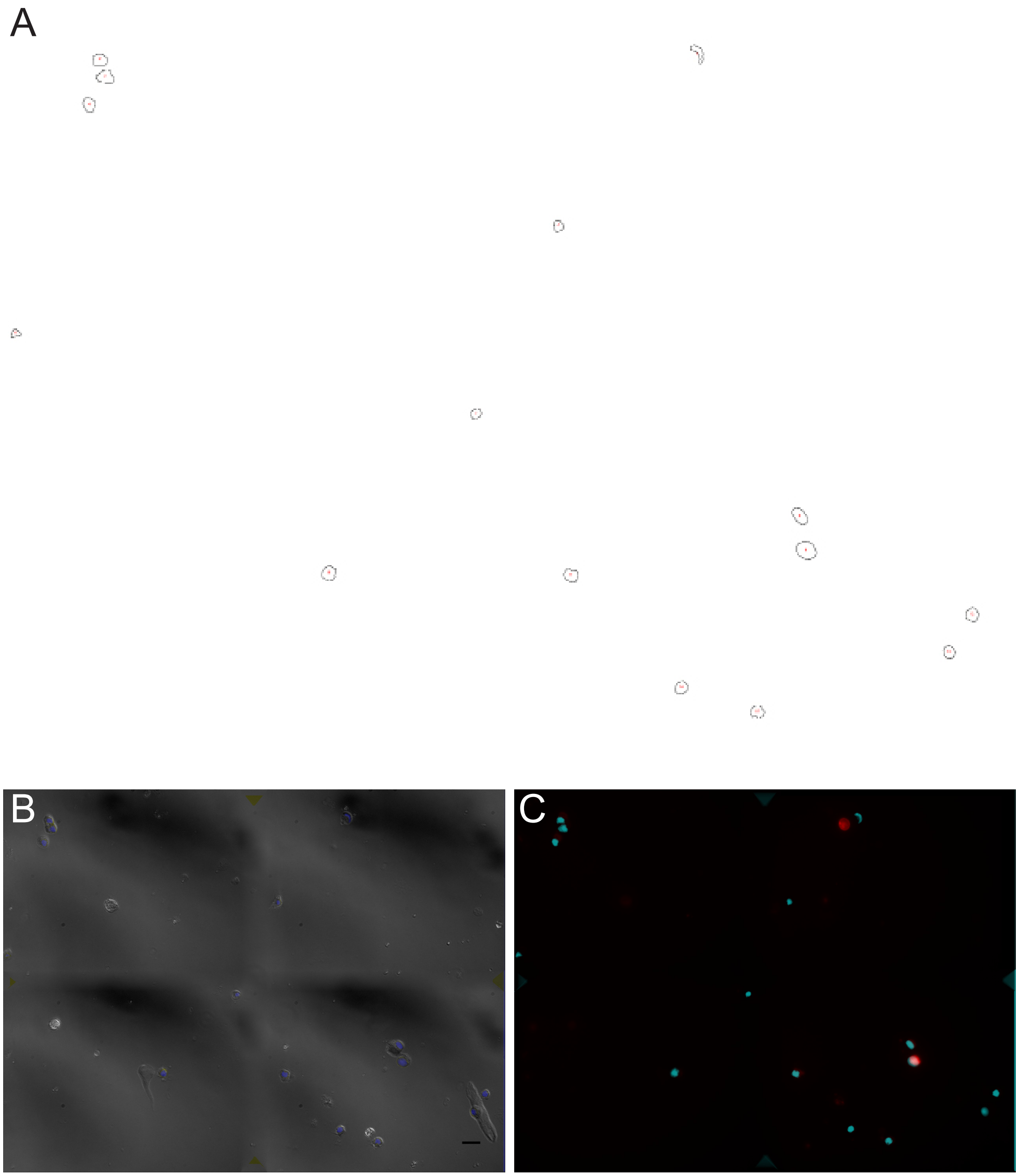


Figure 5.11: Image analysis of fluorescently labeled cells. Slides containing cells with nuclei labeled with DAPI and H3K9K14-Ac were imaged. (A) The resulting images for the experiment #3 slide were stitched together, masked based on DAPI postive nuclei, positive nuclei counted and an image map was created. (B) A stitched transmitted light image was overlayed with the DAPI (blue) image. (C) The DAPI (blue) and H3K9K14-Ac (red) are overlayed.

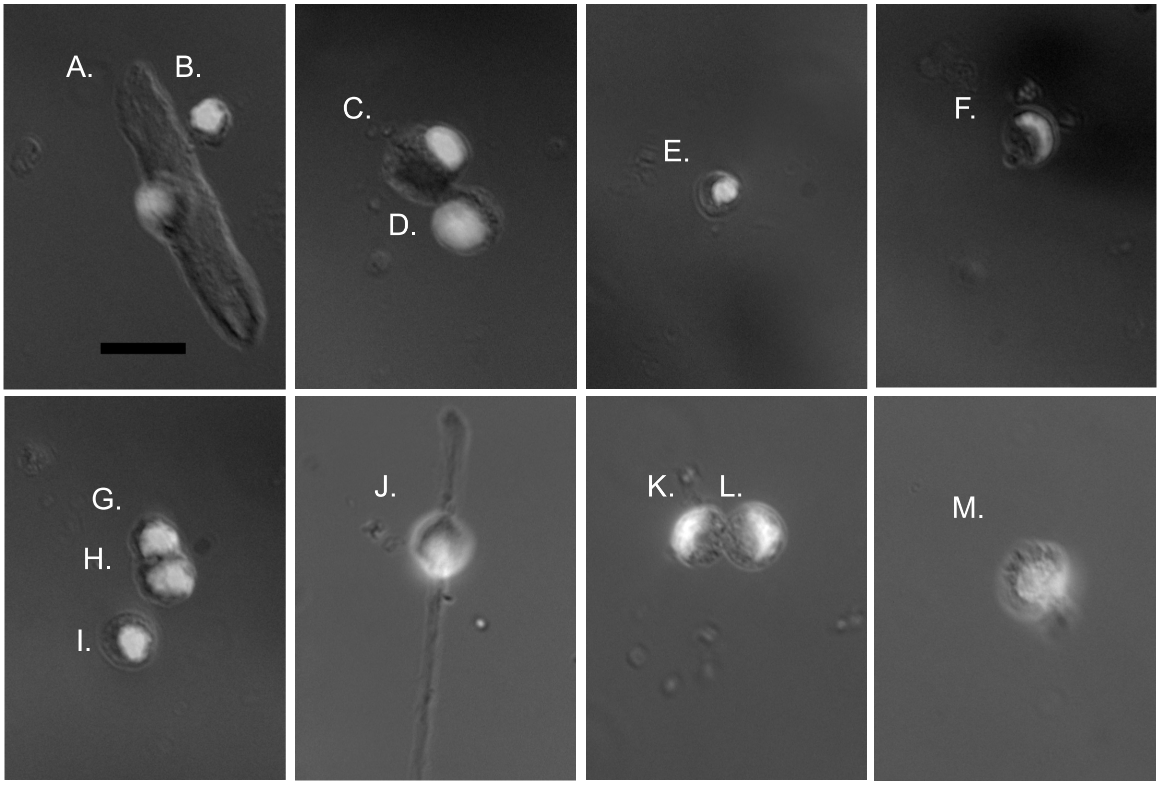


Figure 5.12: Sampling of cell types. These thirteen cells are a sampling of the 39 cells imaged and analyzed in experiments #3, #8 and #11. The scale bar is 10 microns. (A) A muscle cell (B, C, D, E, F, I and M) Potential stem cells (G, H, K, and L) Dividing stem cells. (J) A neuron.

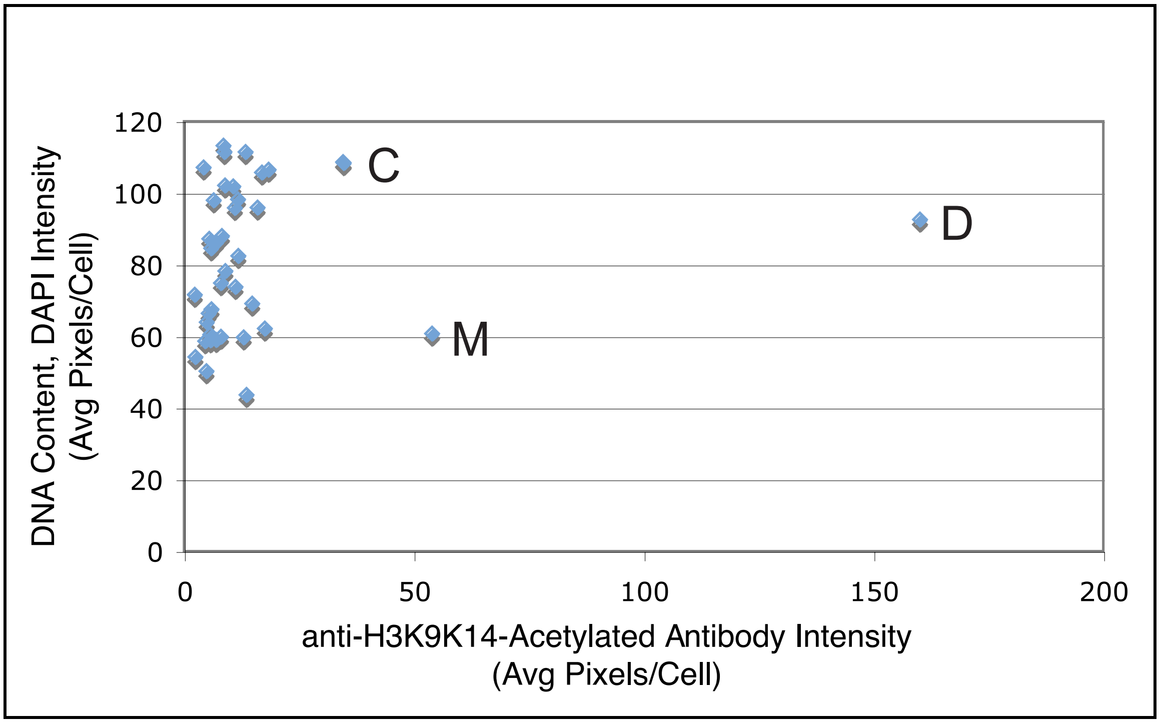


Figure 5.13: Plot of H3K9K14-Ac versus DNA content. The letters labeling the three outliers refer to the cells identified in Figure 5.12 D, C and M.

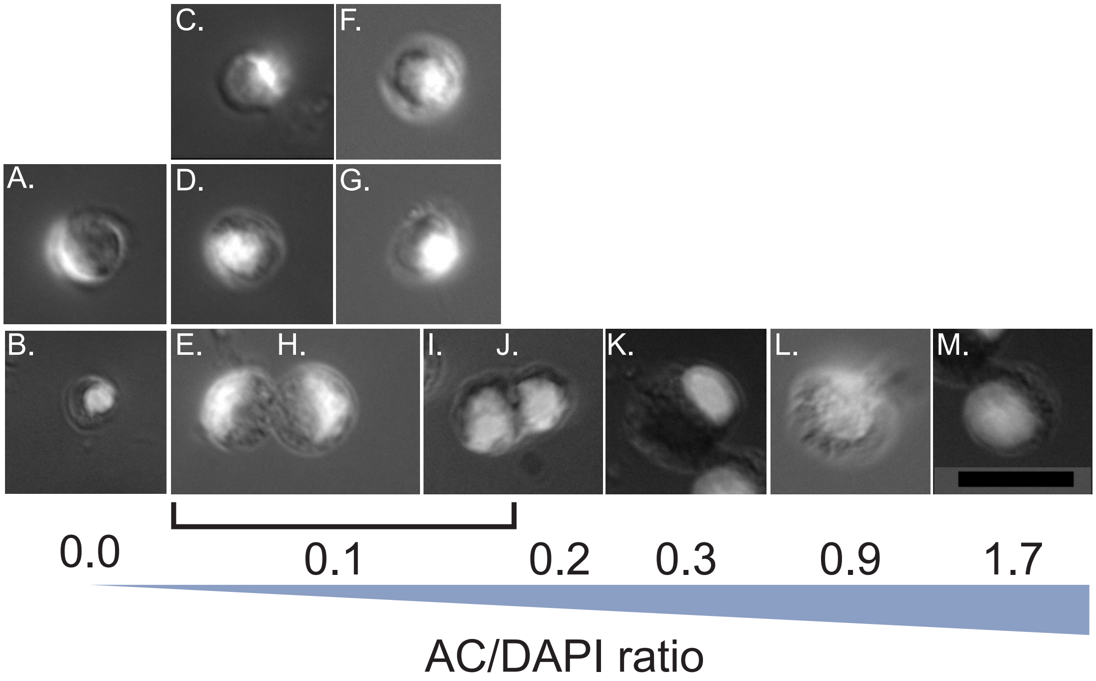


Figure 5.14: Stem Cells and AC/DAPI ratios. The scale bar is 10 microns. (A-B) Stem cells with AC/DAPI ratio of 0.0. (C-I) Most stem cells have a ratio of 0.1. (J-K) Stem cells with a ratio of 0.2 and 0.3. (L-M) Stem cells with a much higher ratio, 0.9 and 1.7.

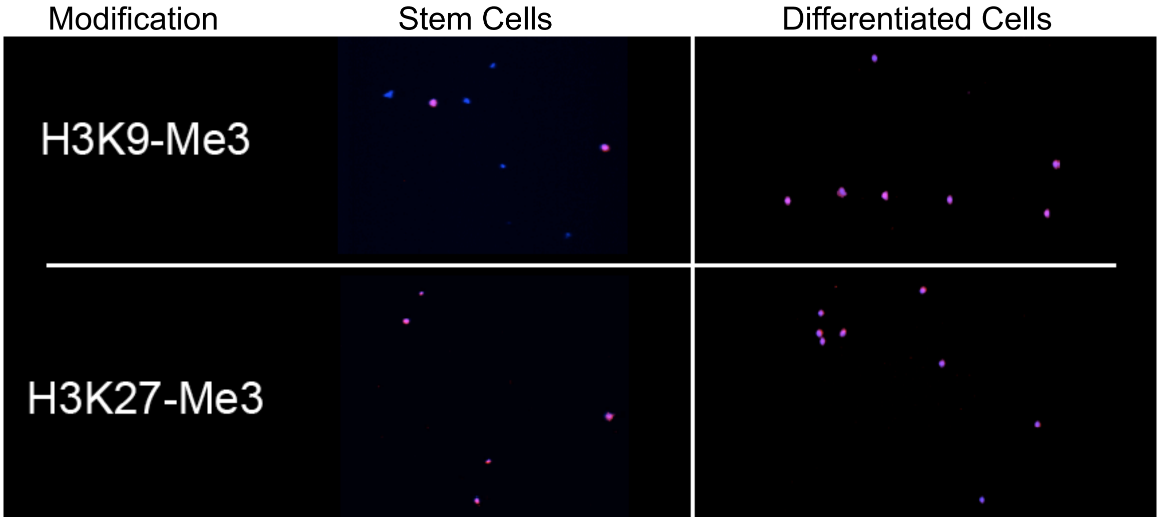


Figure 5.15. H3K9-Me3 and H3K27-Me3 in stem and differentiated cells. These are images of the stem cells and differentiated cells stained with H3K9-Me3 and H3K27-me3 (red) and DAPI (blue).

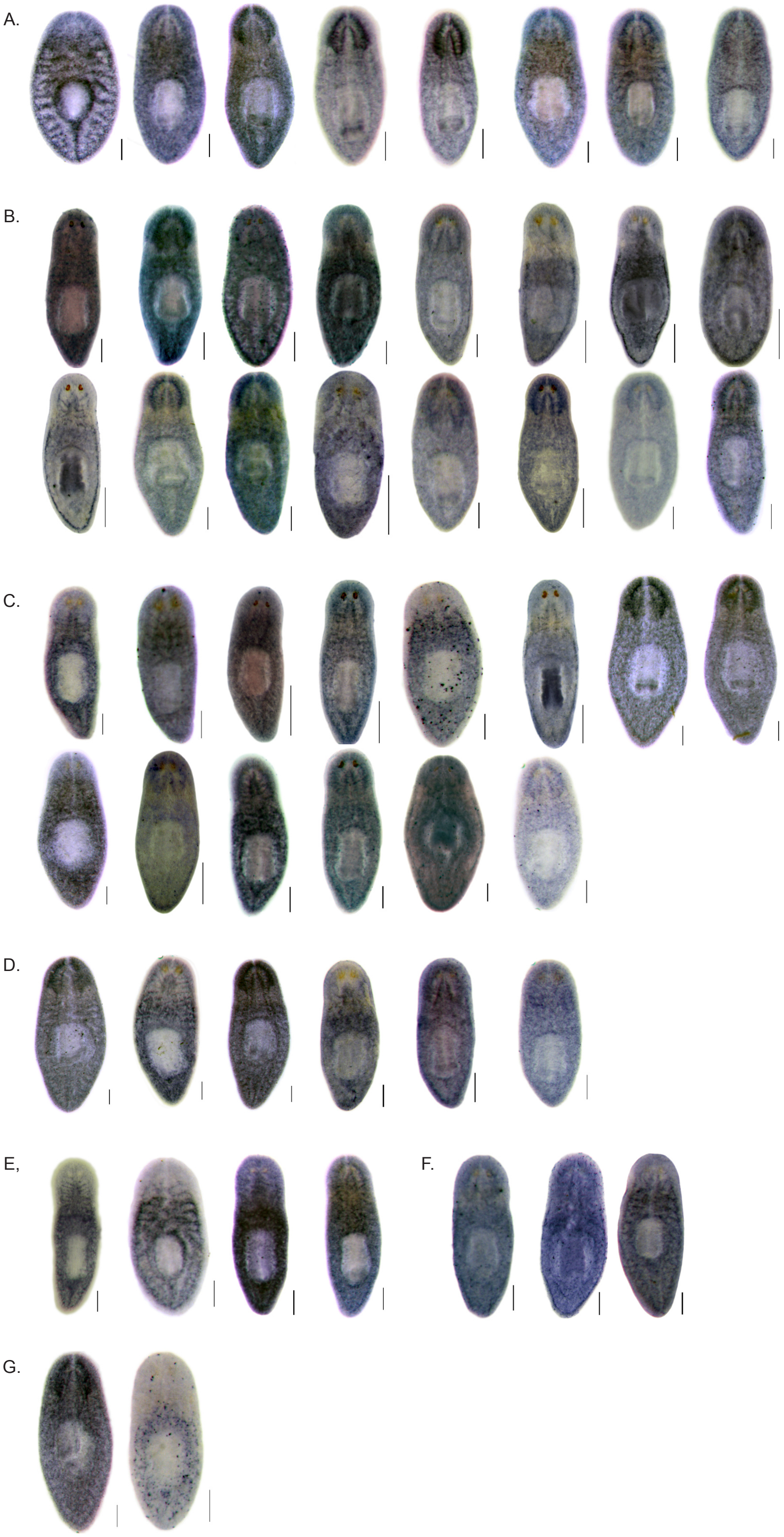

A.
B.
C.
D.
E,
F.
G.

Figure 5.16: Whole-mount *in situ* hybridization expression patterns of epigenetic modifying enzymes. (A) Histone deacetylases; *Smed-HDAC1-1, Smed-HDA2-1, Smed-HDAC5-1, HDAC8-1, Smed-Sin3-1, Smed-SAP18-1, Smed-SAP180-1,* and *HDAC1-3*. (B) Histone acetyltransferases; *Smed-ELP3-1, Smed-NCOAT-2, Smed-CBP-2, Smed-YEATS4-1, Smed-CBP-4, Smed-NCOAT-1, Smed-MYST3-1, Smed-MYST4-2, Smed-MYST2-2, Smed-Ep300-2, Smed-MYST4-1, Smed-HAT-1, Smed-CBP-5, Smed-CBP-6, Smed-Ep300-3* and *Smed-Tip60-1.* (C) Histone methyltransferases; *Smed-SETD8-1, Smed-SUV42-1, Smed-SETB1-1, Smed-CARM1-1, Smed-NSD2-2, Smed-SUV92-1, Smed-TRR-1, Smed-DOT1L-1, Smed-ANM8-1, Smed-ANMX-1, Smed-ANM1-1, Smed-ASH2L-1, Smed-SETB1-2* and *Smed-SUV92-2*. (D) Histone demethylases; *Smed-JARID-3, Smed-JmjC-1, Smed-JARID-2, Smed-JmjC-6, Smed-JmjC-4* and *Smed-JmjC-7.* (E) Putative DNA demethylases; *Smed-RL7A, Smed-NHP2L1, Smed-RPS12* and *Smed-RPL30.* (F) Endonucleases; Smed-XPG-1, Smed-XPG-2 and Smed-FEN1. (G) Histone phosphorylases; *Smed-TAF1-2 and Smed-TAF1-3*.

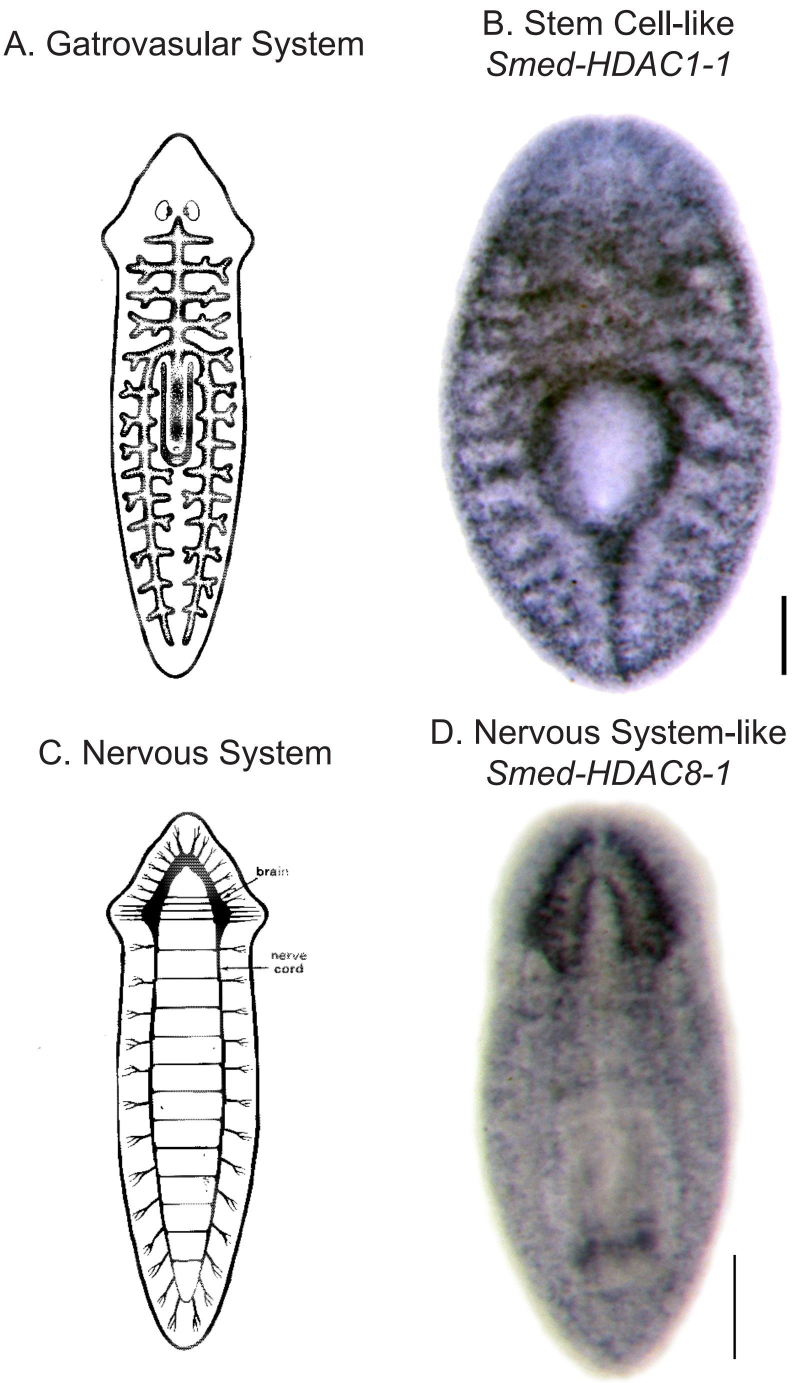


Figure 5.17: Two common whole-mount *in situ* hybridization expression patterns obtained. Scale bar is 200 microns. (A) An illustration of the gastrovasular system (Pearse et al., 1997). (B) The stem cell-like pattern is the expression of mRNA between the branches of the gut and an absence of expression anterior to the photoreceptors and in the pharynx as is exemplified by *Smed-HDAC1-1*. (C) An illustration of the nervous system (Pearse et al., 1997). (D) The nervous system-like pattern is found around the brain down the ventral cords and at the distal tip of the pharynx as can be seen in *Smed-HDAC8-1*.

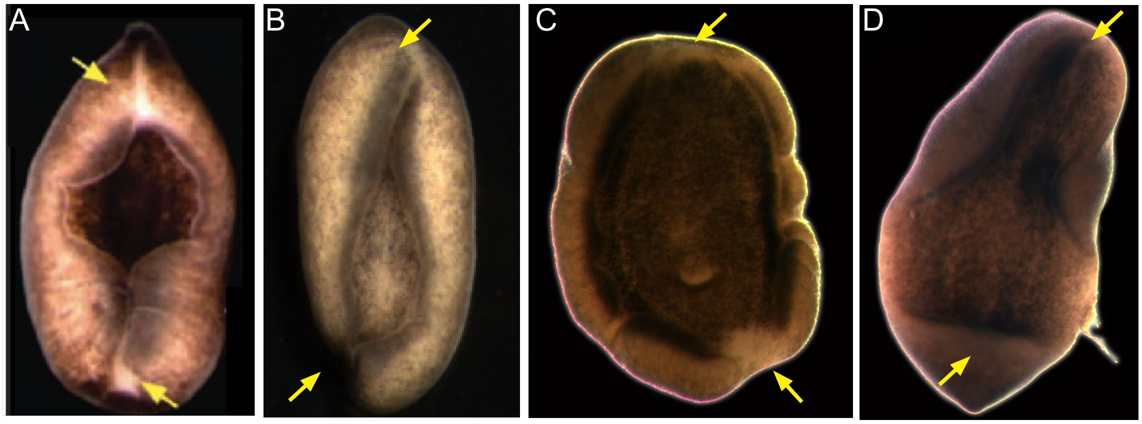


Figure 5.18: Stem cell deficient phenotype. This phenotype is characterized by a curling of the epithelium onto the ventral surface of the worm and by a lack of blastemas and regeneration, indicated by yellow arrows, when amputated. (A) 6K rad irradiated wildtype worm. (B) *Smed-HDAC1-1(RNAi)* animal. (C) *Smed-SETD8-1(RNAi)* animal. (D) *Smed-RL7A(RNAi)* animal.

```
CLUSTAL W (1.83) multiple sequence alignment

Smed-RL7A       PNNKAKLLAKKKKSVKKVAPVPEIIKSKTIKKKIVNPLFESRPKNFAIGQ 50
Smed-NHP2L1     XPLLNE-----------------------LTTKII-----NVI------- 15
Smed-RPL30      PTKKVK------------------------EENIG-----SRL------- 14
Smed-RPS12      KSVDS---------------------------LVQ-----SVL------- 11
                                                 :      .

Smed-RL7A       DIQSKRDLTRFVKWPKYVRLQRQKSILNKRLKMPAVINQFNNAIDKPNAK 100
Smed-NHP2L1     --------------------------------------------QKA--- 18
Smed-RPL30      --------------------------------------------ALA--- 17
Smed-RPS12      --------------------------------------------KDA--- 14
                                                              .

Smed-RL7A       SIVKLLSKYRPESKQEKVARLKLRASERVEGKPETVETKPALIRYGVREI 150
Smed-NHP2L1     -------------------------------------SSLSQIKKGANEA 31
Smed-RPL30      -------------------------------------VKTGKYCLGLNQT 30
Smed-RPS12      -------------------------------------VSVRGLTCGIHQT 27
                                                      .      * .:

Smed-RL7A       TTLVEQKKAQMVVIAHDVDPIEI--VLHLPALCRKMGVPYCILKGKAILG 198
Smed-NHP2L1     TKALNRGRAQLIVMAADAEPLEI--LLHLPLLCEDKNVPYVFIPSKTALG 79
Smed-RPL30      LKTIRSGKAELIIVAKNANPLVKSQVEYYSML-SRKGIHH-FDGNNSDLG 78
Smed-RPS12      TKALESQKALLCILAKNCDDANY--TKLIEVLCKEHNIPLLKVDSNKKLG 75
                 . :.  :* : ::* : :            *    .:      .:  **

Smed-RL7A       QIVRKK-T--CSAVALVNVASEDKAPLNKLTEIMMTNFNNRGDEIRKHVG 245
Smed-NHP2L1     RACGVSRH--VIACAITES---------------------TGSEVTPLV- 105
Smed-RPL30      TACGKLFK--VSXXAI---------------------------------- 92
Smed-RPS12      EYAGLCKLDKDLNPRKTVS---------------------CSSVVISNI- 103


Smed-RL7A       GGIMGL 251
Smed-NHP2L1     QGIHTA 111
Smed-RPL30      --TDPG 96
Smed-RPS12      PNNCTN 109
```

Figure 5.19: Alignment of the L7Ae family of genes in *S. mediterranea*. The amino acids colored in bold green make up the L7Ae domain.

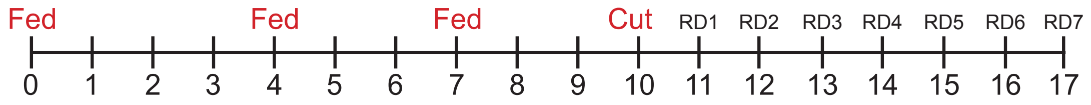


Figure 5.20: RNAi feeding and amputation schedule. The worms were fed on day 0, 4 and 7 (D0, D4 and D7). The amputation was administered on D10. Animals were allowed to attempt regeneration for up to 7 days after amputation. The days past amputation are labeled as RD1 through RD7, or regeneration day 1 through regeneration day 7.

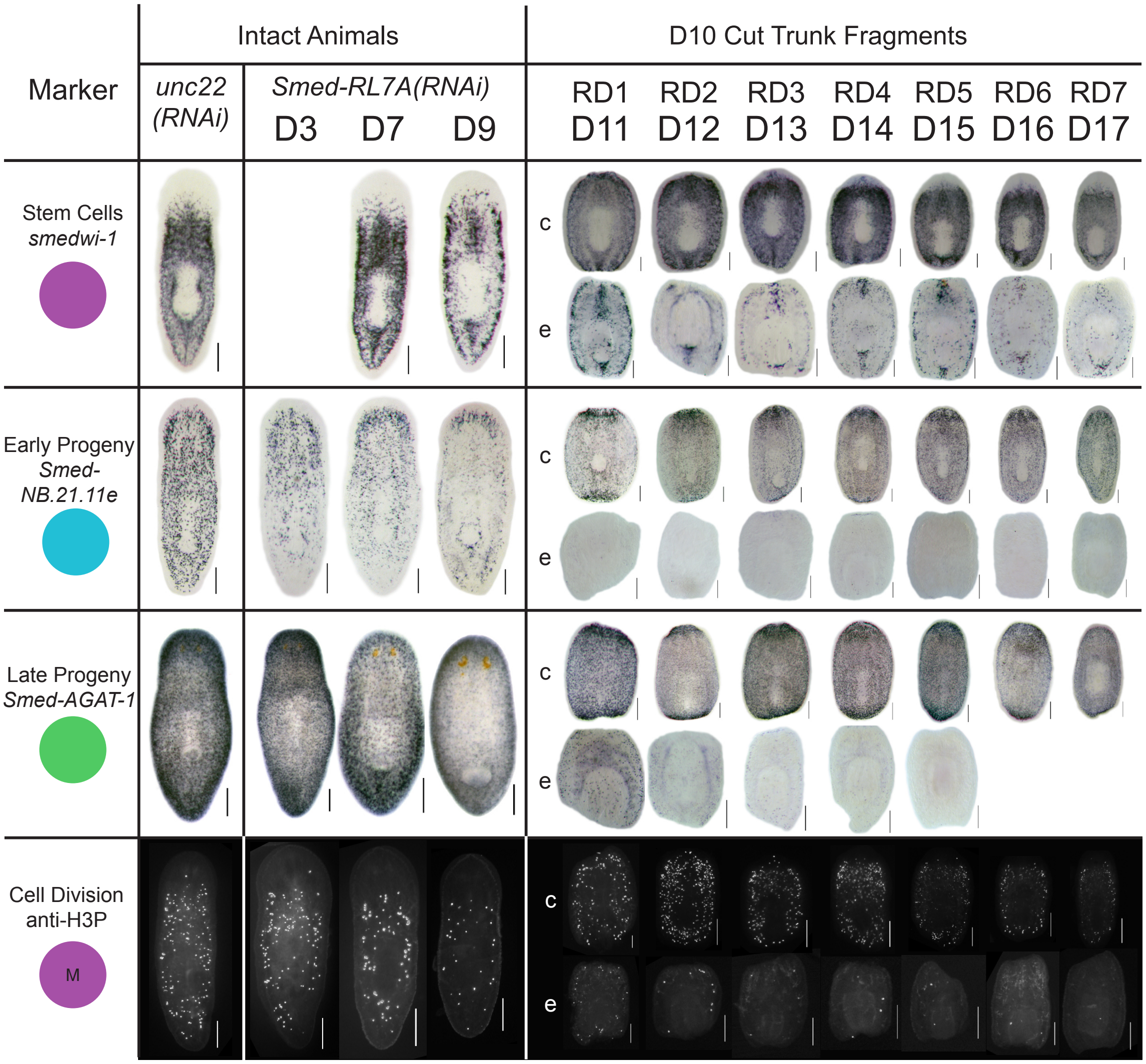


Figure 5.21: Stem cell lineage analysis of *Smed-RL7A(RNAi)* animals. Intact *Smed-RL7A(RNAi)* animals were fixed on D3, D7 and D9 and amputated *Smed-RL7A(RNAi)* animals (labeled “e” for experimental) were fixed on D11 through D17 (RD1 through RD7) and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control for intact animals and for amputated animals (labeled “c” for control).

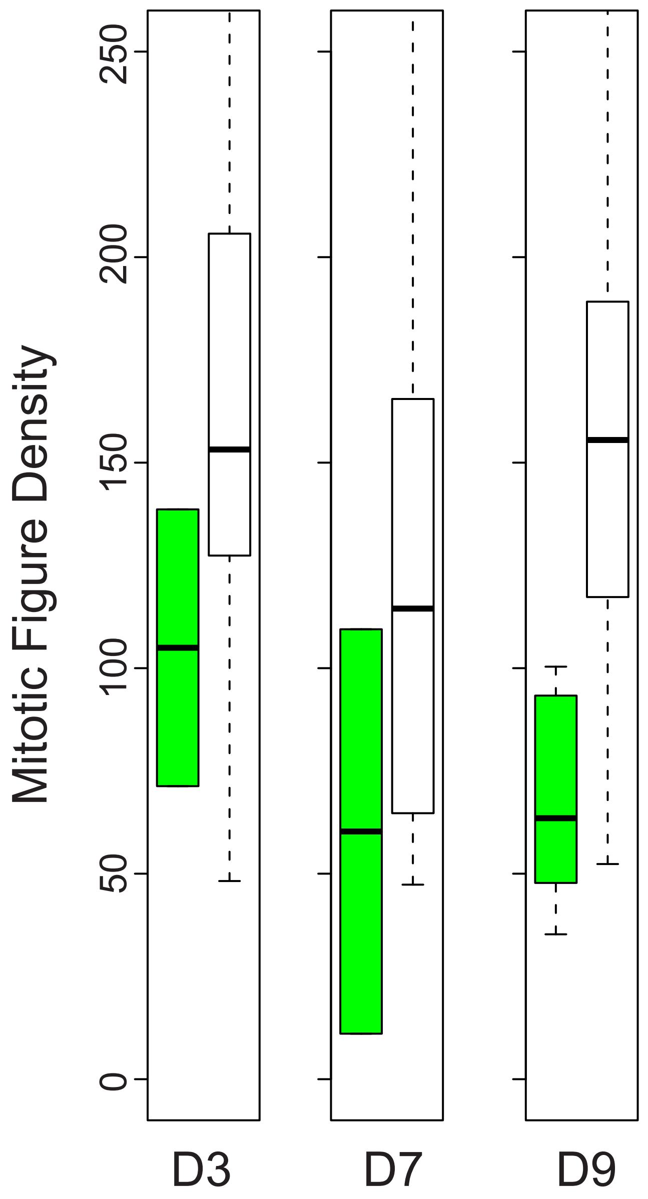


Figure 5.22: Mitotic stem cell density in *Smed-RL7A(RNAi)* animals. For *Smed-RL7A(RNAi)* (green) and *unc22(RNAi)* (white) animals on D3, D7, and D9 mitotic figures were identified, counted and compared to the volume of the worm to produce a mitotic figure density. This graph is a boxplot, the box itself contains the middle 50% of the data. The upper edge of the box indicates the 75th percentile of the data set, and the lower edge indicates the 25th percentile. The line in the box indicates the median value of the data. The ends of the vertical lines indicate the minimum and maximum data values, and the points outside the ends of the vertical lines are outliers or suspected outliers.

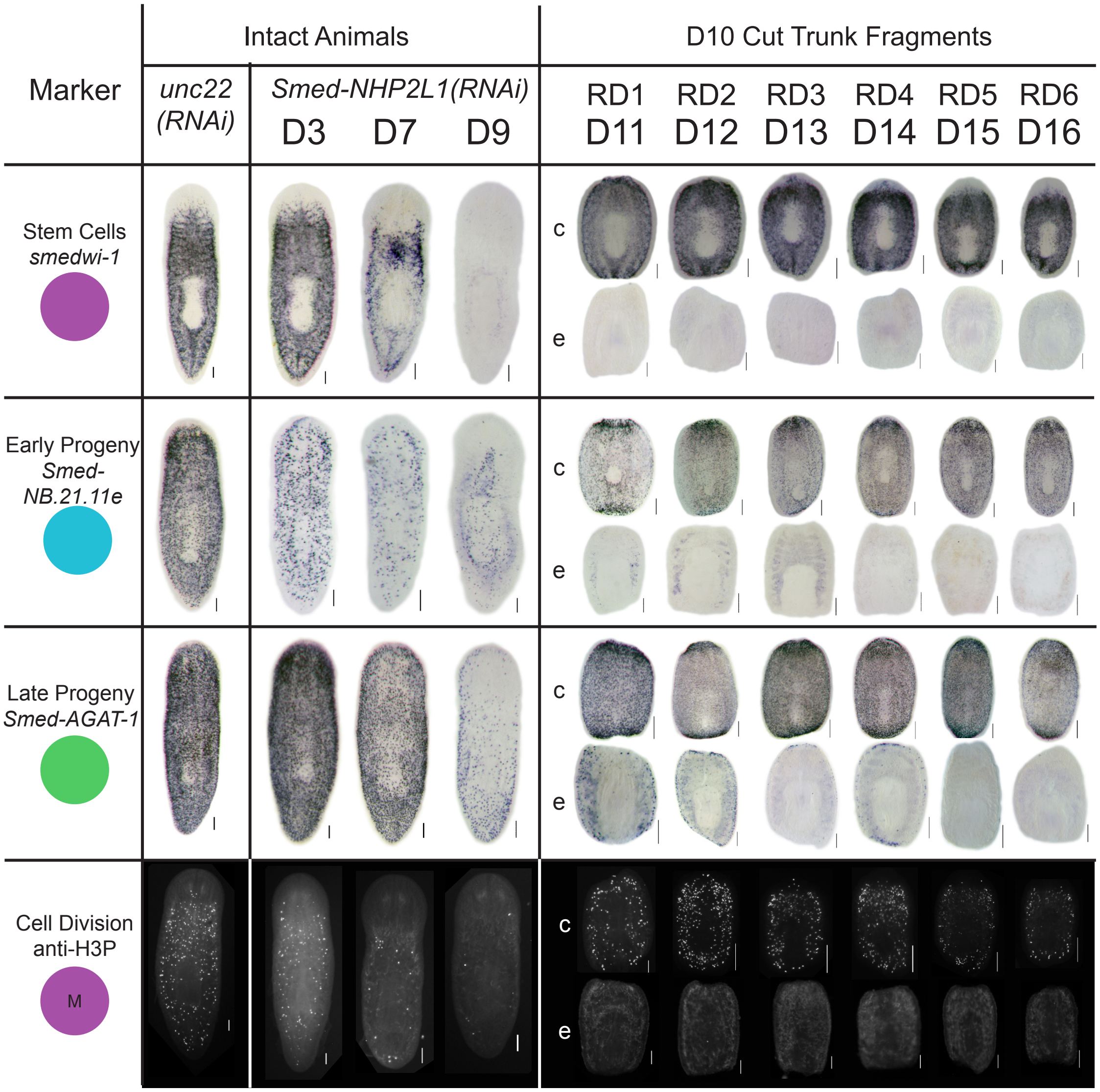


Figure 5.23: Stem cell lineage analysis of *Smed-NHP2L1(RNAi)* animals. Intact *Smed-NHP2L1(RNAi)* animals were fixed on D3, D7 and D9 and amputated *Smed- NHP2L1(RNAi)* animals (labeled “e” for experimental) were fixed on D11 through D16 (RD1 through RD6) and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control for intact animals and for amputated animals (labeled “c” for control).

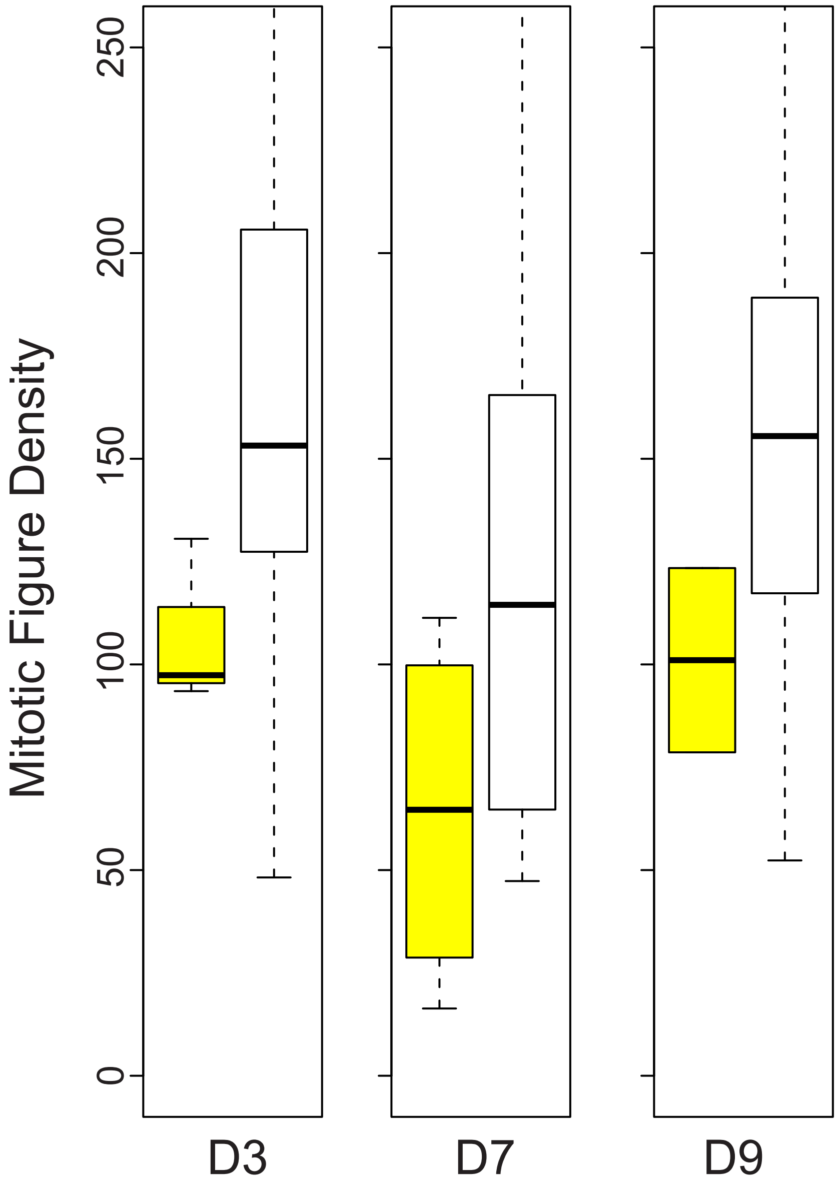


Figure 5.24: Mitotic stem cell density in *Smed-NHP2L1(RNAi)* animals. For *Smed-NHP2L1(RNAi)* (yellow) and *unc22(RNAi)* (white) animals on D3, D7, and D9 mitotic figures were identified, counted and compared to the volume of the worm to produce a mitotic figure density. This graph is a boxplot, the box itself contains the middle 50% of the data. The upper edge of the box indicates the 75th percentile of the data set, and the lower edge indicates the 25th percentile. The line in the box indicates the median value of the data. The ends of the vertical lines indicate the minimum and maximum data values, and the points outside the ends of the vertical lines are outliers or suspected outliers.

```
CLUSTAL W (1.83) multiple sequence alignment

Smed-RPS12               -------------------EE-----KSVDSLVQSVLKDAVSVRGLTCGI 26
GA45A_HUMAN              MTL-EEFSA------G---EQKTERMDKVGDALEEVLSKALSQRTITVGV 40
RS12_PIG                 MAE-EGIAA------G---GV-----MDVNTALQEVLKTALIHDGLARGI 35
RS12_YEAST               MSDVEEVVEVQEETVVEQTAE-----VTIEDALKVVLRTALVHDGLARGL 45
                                                     :   :: **  *:    :: *:

Smed-RPS12               HQTTKALES--QKALLCILAKNCDDAN------YTKLIEVLC--KEHNIP 66
GA45A_HUMAN              YEAAKLLNVDPDNVVLCLLAADEDDDRDVALQIHFTLIQAFC--CENDIN 88
RS12_PIG                 REAAKALDK--RQAHLCVLASNCDEPM------YVKLVEALC--AEHQIN 75
RS12_YEAST               RESTKALTR--GEALLVVLVSSVTEAN------IIKLVEGLANDPENKVP 87
                          :::* *     :. * :*. .  :          .*:: :.   *:.:

Smed-RPS12               LLKVSFRLTAIKNLENMQAYANLTRIVDSNKKLGEYAGLCKLDKDLNPRK 116
GA45A_HUMAN              ILRVS------------------------NPGRLAEL---LLLETDAGPAA 112
RS12_PIG                 LIKVD-----------------------DNKKLGEWVGLCKIDREGKPRK 102
RS12_YEAST               LIKVA-----------------------DAKQLGEWAGLGKIDREGNARK 114
                         :::*                        .  :*.*      :: :   .

Smed-RPS12               TV------SCSSVVISNIPNNCTNWEALMS-------------------E 141
GA45A_HUMAN              SEGAEQPPDLHCVLVTNPH-SSQWKDPALSQLICFCRESRYMDQWVPVIN 161
RS12_PIG                 VV------GCSCVVVKDYGKESQAKDVIEE-------------------Y 127
RS12_YEAST               VV------GASVVVVKNWGAETDELSMIME-------------------H 139
                                 .   *::.:   .    .   .

Smed-RPS12               V-KKQ 145
GA45A_HUMAN              L-PER 165
RS12_PIG                 FKCKK 132
RS12_YEAST               F-SQQ 143
                         .  ::
```

Figure 5.25: Alignment of the proteins of human Gadd45a, Smed-RPS12, and pig and yeast RS12.

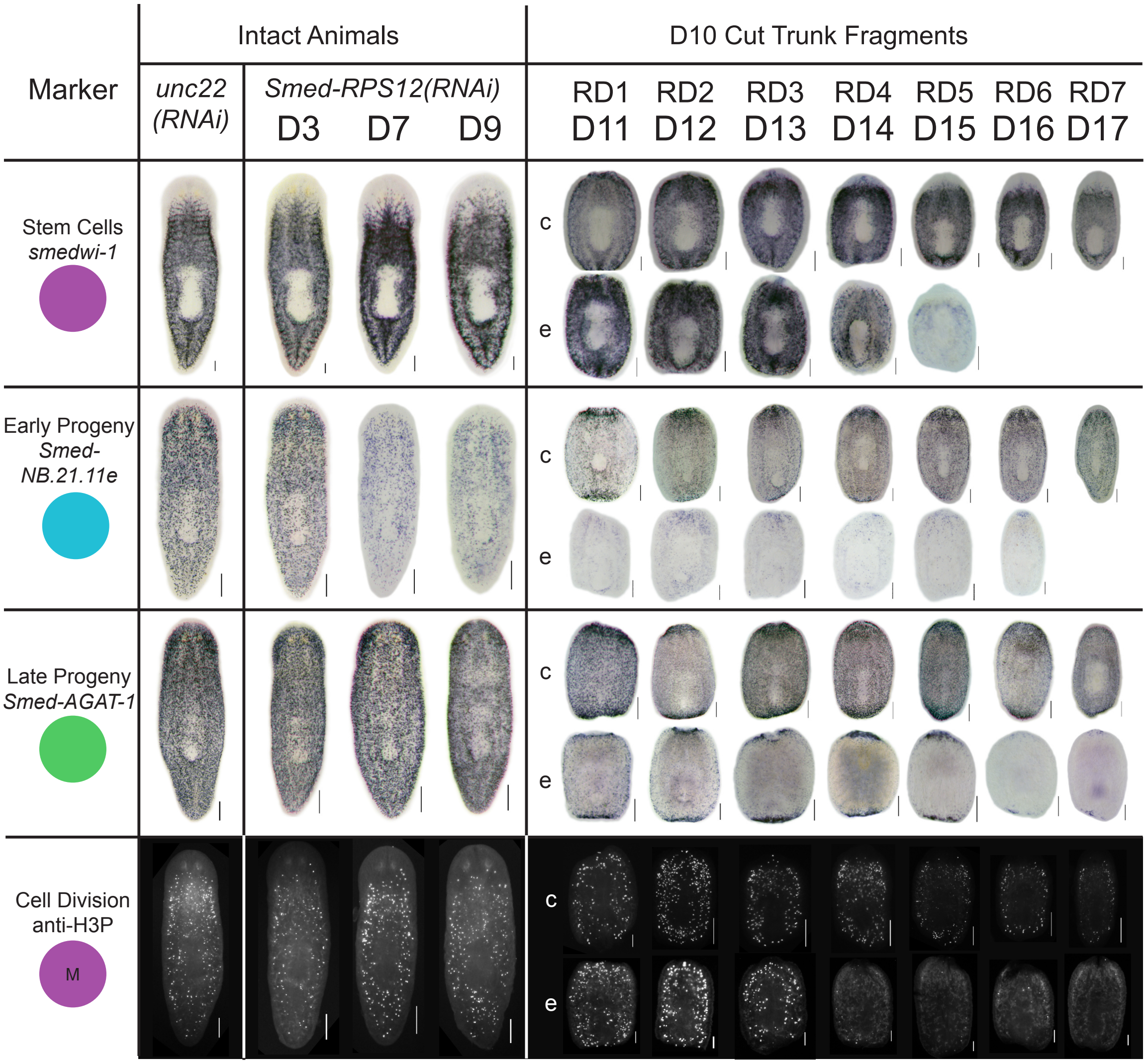


Figure 5.26: Stem cell lineage analysis of *Smed-RPS12(RNAi)* animals. Intact *Smed-RPS12(RNAi)* animals were fixed on D3, D7 and D9 and amputated *Smed-RPS12(RNAi)* animals (labeled “e” for experimental) were fixed on D11 through D17 (RD1 through RD7) and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control for intact animals and for amputated animals (labeled “c” for control).

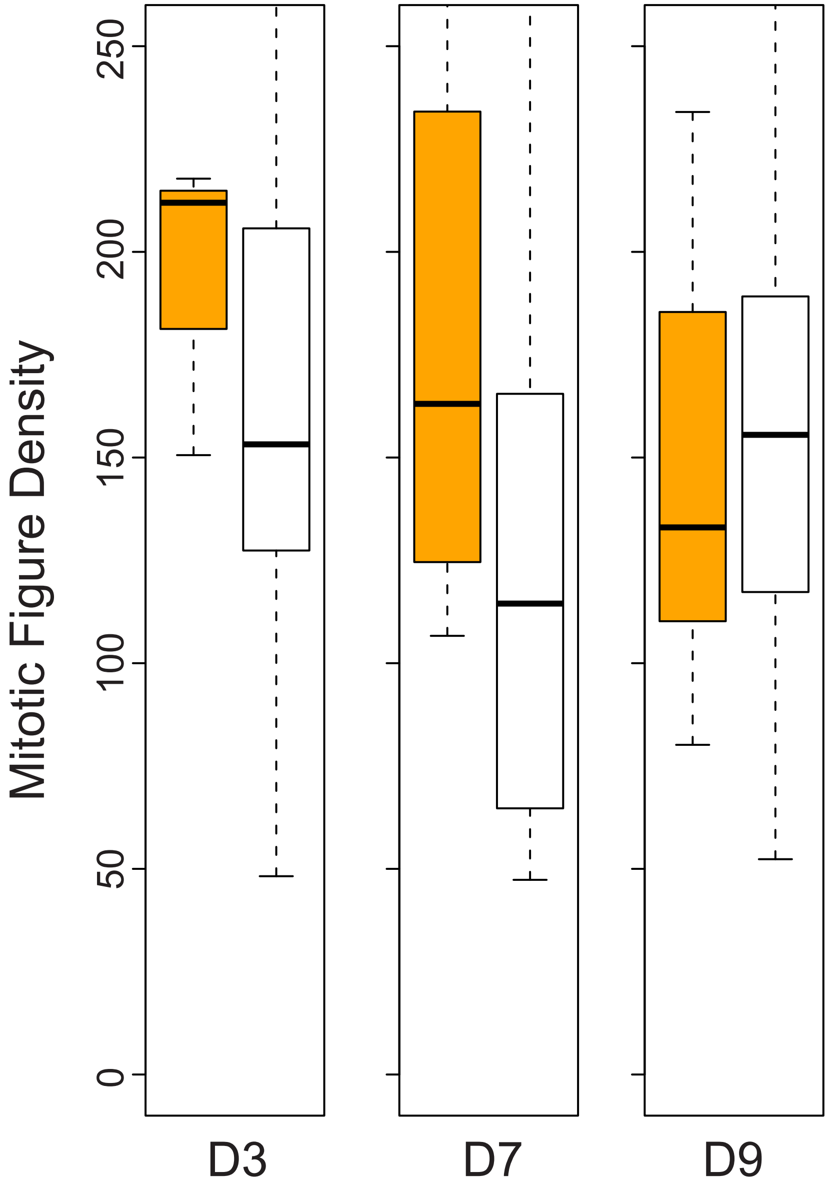


Figure 5.27: Mitotic stem cell density in *Smed-RPS12(RNAi)* animals. For *Smed-RPS12(RNAi)* (orange) and *unc22(RNAi)* (white) animals on D3, D7, and D9 mitotic figures were identified, counted and compared to the volume of the worm to produce a mitotic figure density. This graph is a boxplot, the box itself contains the middle 50% of the data. The upper edge of the box indicates the 75th percentile of the data set, and the lower edge indicates the 25th percentile. The line in the box indicates the median value of the data. The ends of the vertical lines indicate the minimum and maximum data values, and the points outside the ends of the vertical lines are outliers or suspected outliers.

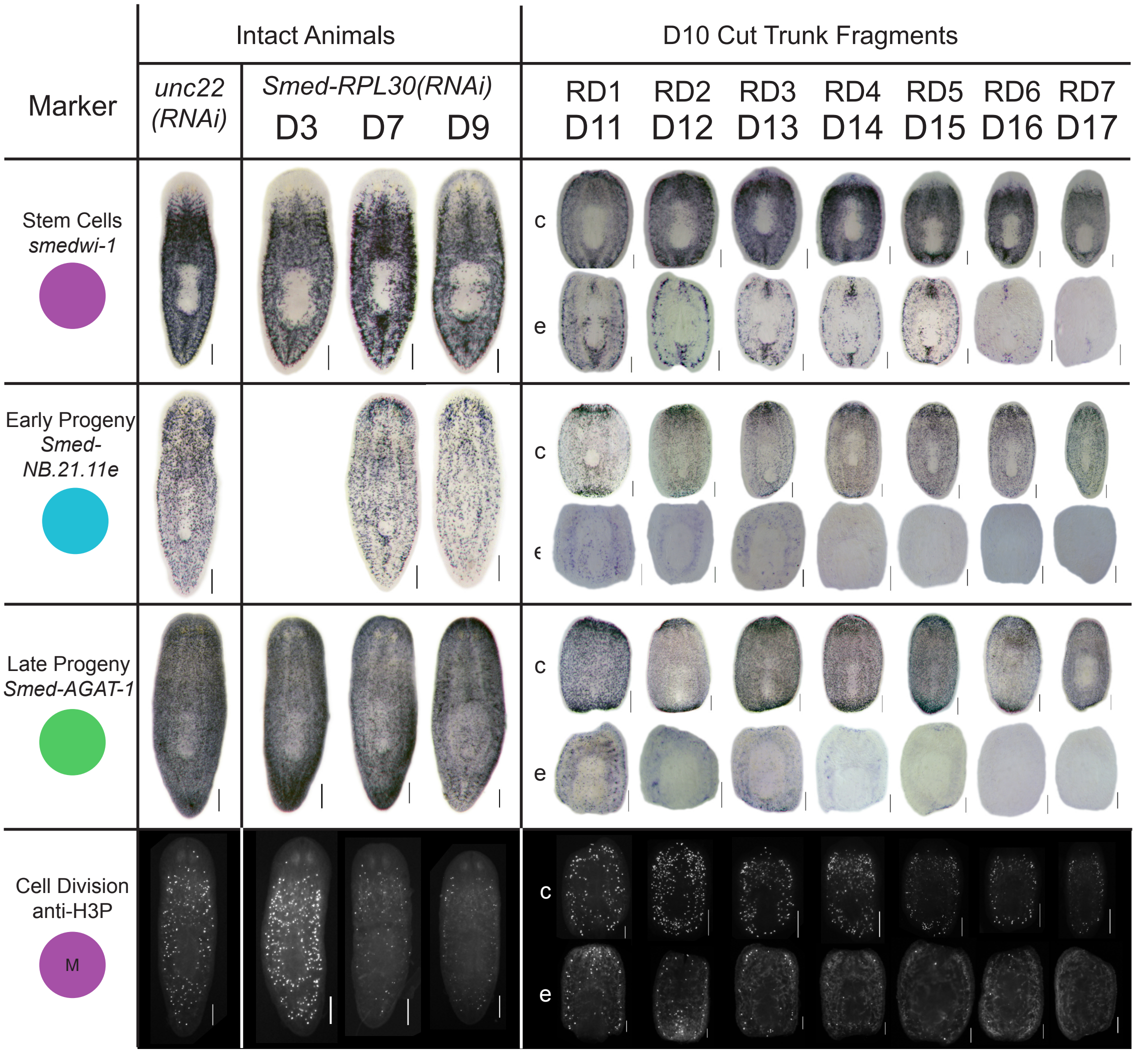


Figure 5.28: Stem Cell Lineage Analysis of *Smed-RPL30(RNAi)* animals. Intact *Smed-RPL30(RNAi)* animals were fixed on D3, D7 and D9 and amputated *Smed-RPL30(RNAi)* animals (labeled “e” for experimental) were fixed on D11 through D17 (RD1 through RD7) and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control for intact animals and for amputated animals (labeled “c” for control).

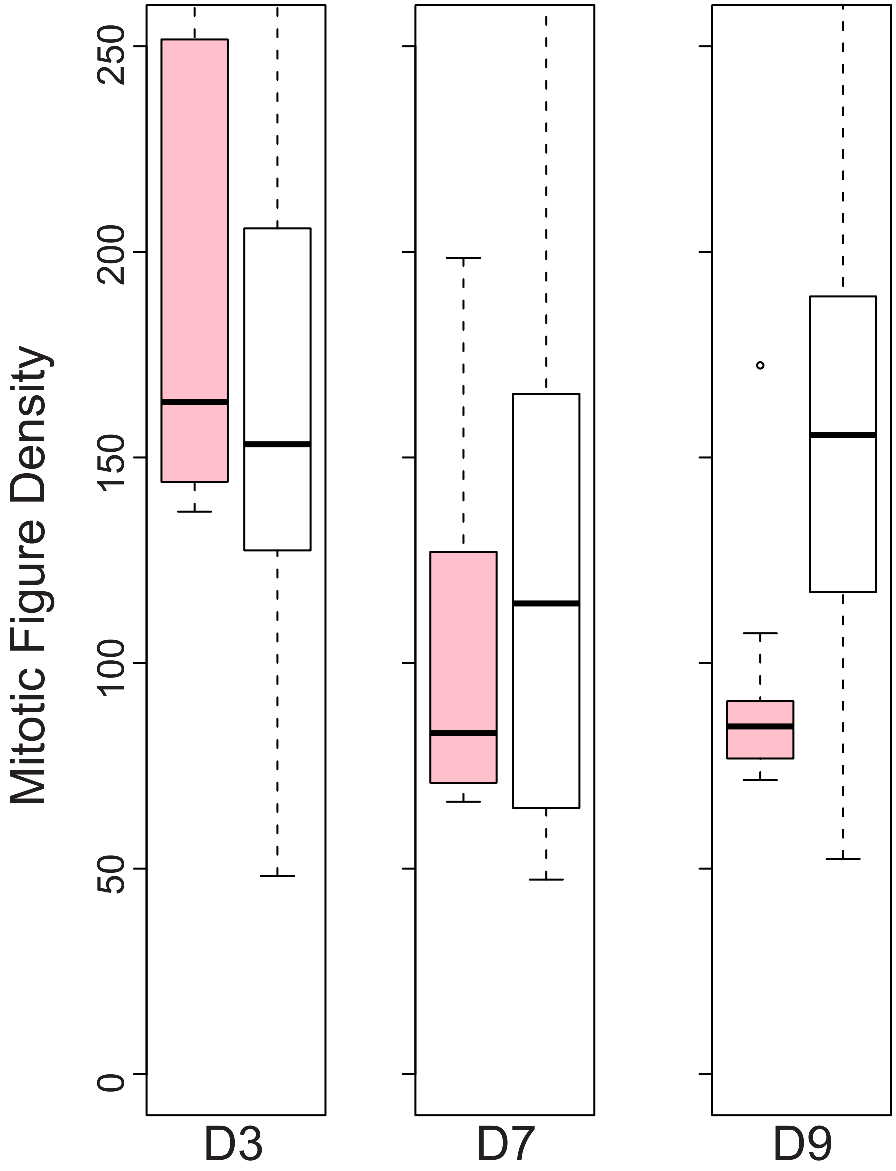


Figure 5.29: Mitotic stem cell density in *Smed-RPL30(RNAi)* animals. For *Smed-RPL30(RNAi)* (pink) and *unc22(RNAi)* (white) animals on D3, D7, and D9 mitotic figures were identified, counted and compared to the volume of the worm to produce a mitotic figure density. This graph is a boxplot, the box itself contains the middle 50% of the data. The upper edge of the box indicates the 75th percentile of the data set, and the lower edge indicates the 25th percentile. The line in the box indicates the median value of the data. The ends of the vertical lines indicate the minimum and maximum data values, and the points outside the ends of the vertical lines are outliers or suspected outliers.

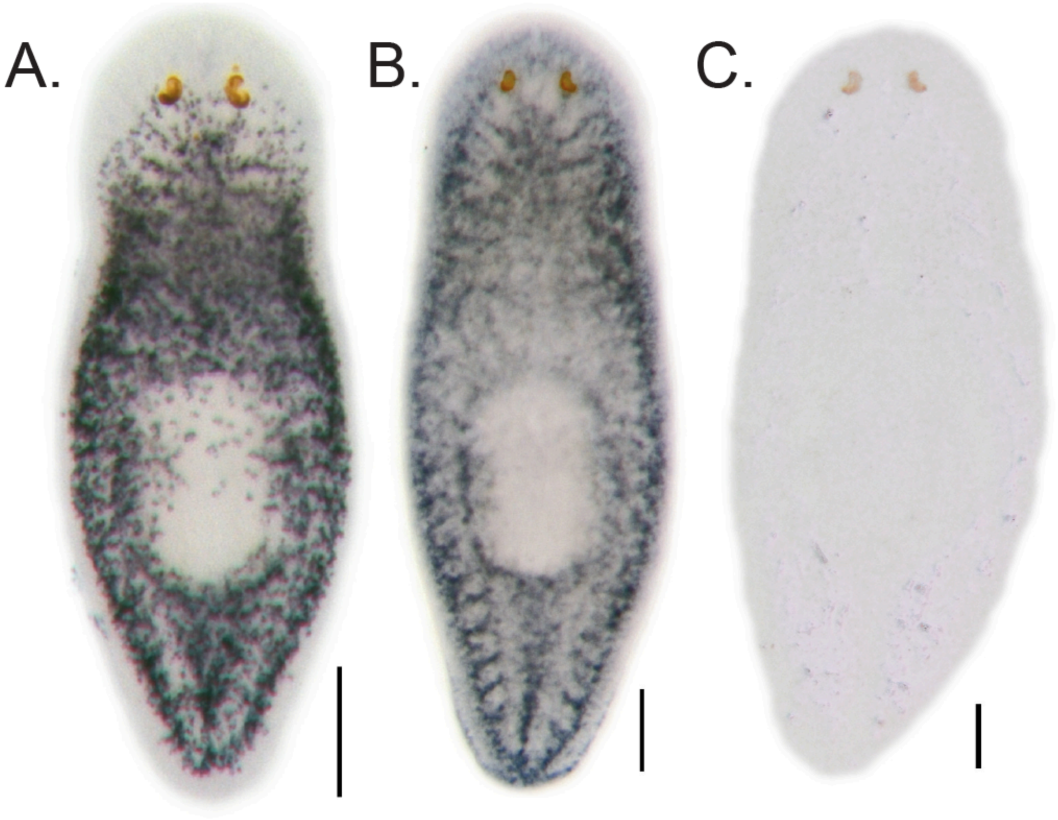


Figure 5.30: The stem cell-like expression pattern of *Smed-HDAC1-1* disappears with RNAi. (A) Canonical stem cell-like expression pattern can be seen in *smedwi-1*. (B) The expression pattern of *Smed-HDAC1-1* resembles the stem cell-like pattern, but does have some expression anterior to the photoreceptors. (C) The *Smed-HDAC1-1* expression disappears in a *Smed-HDAC1-1(RNAi)* animals. This WISH was preformed at D12.

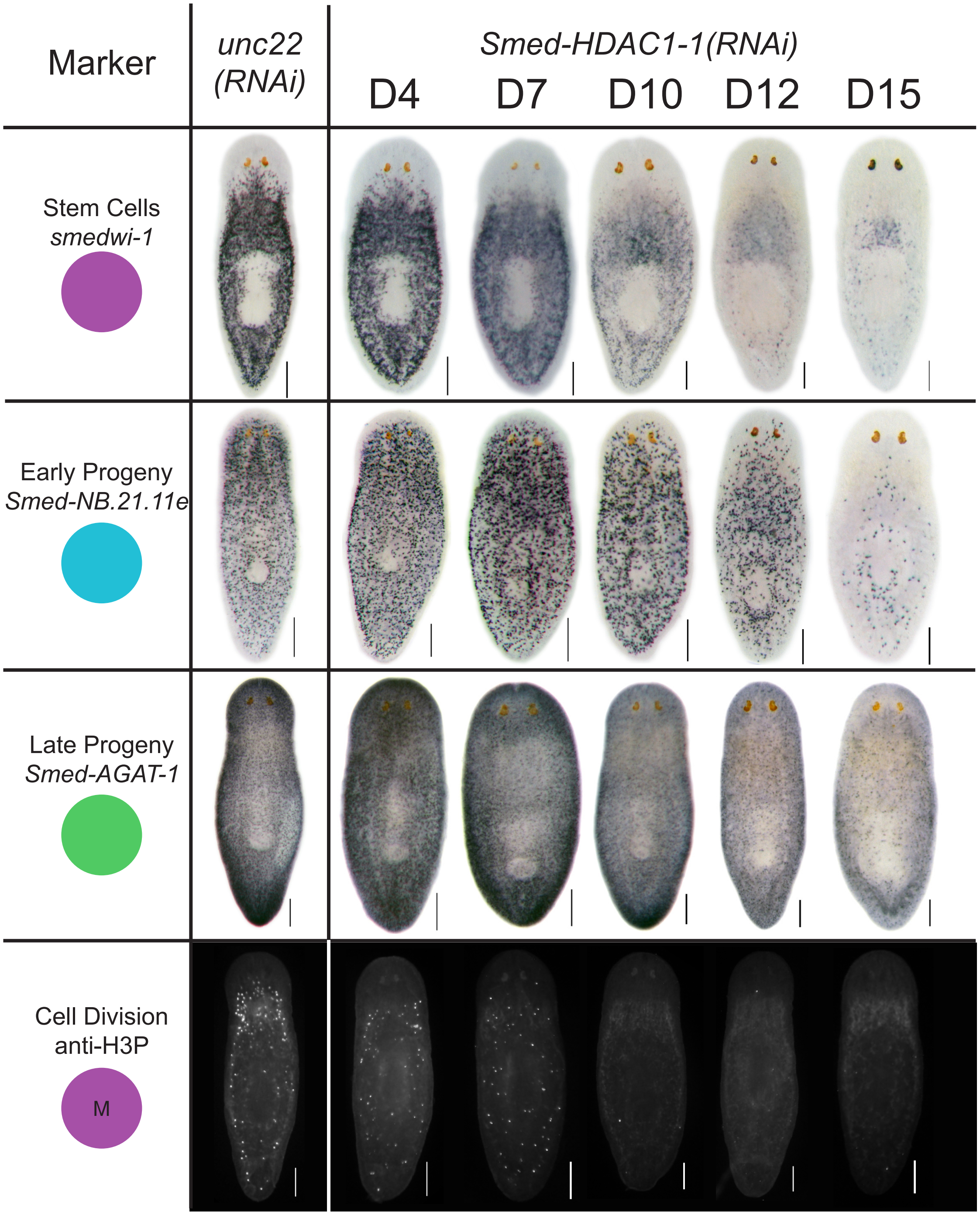


Figure 5.31: Stem cell lineage analysis of intact *Smed-HDAC1-1(RNAi)* animals. Intact *Smed-HDAC1-1(RNAi)* animals were fixed on D4, D7, D10, D12 and D15 and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control.

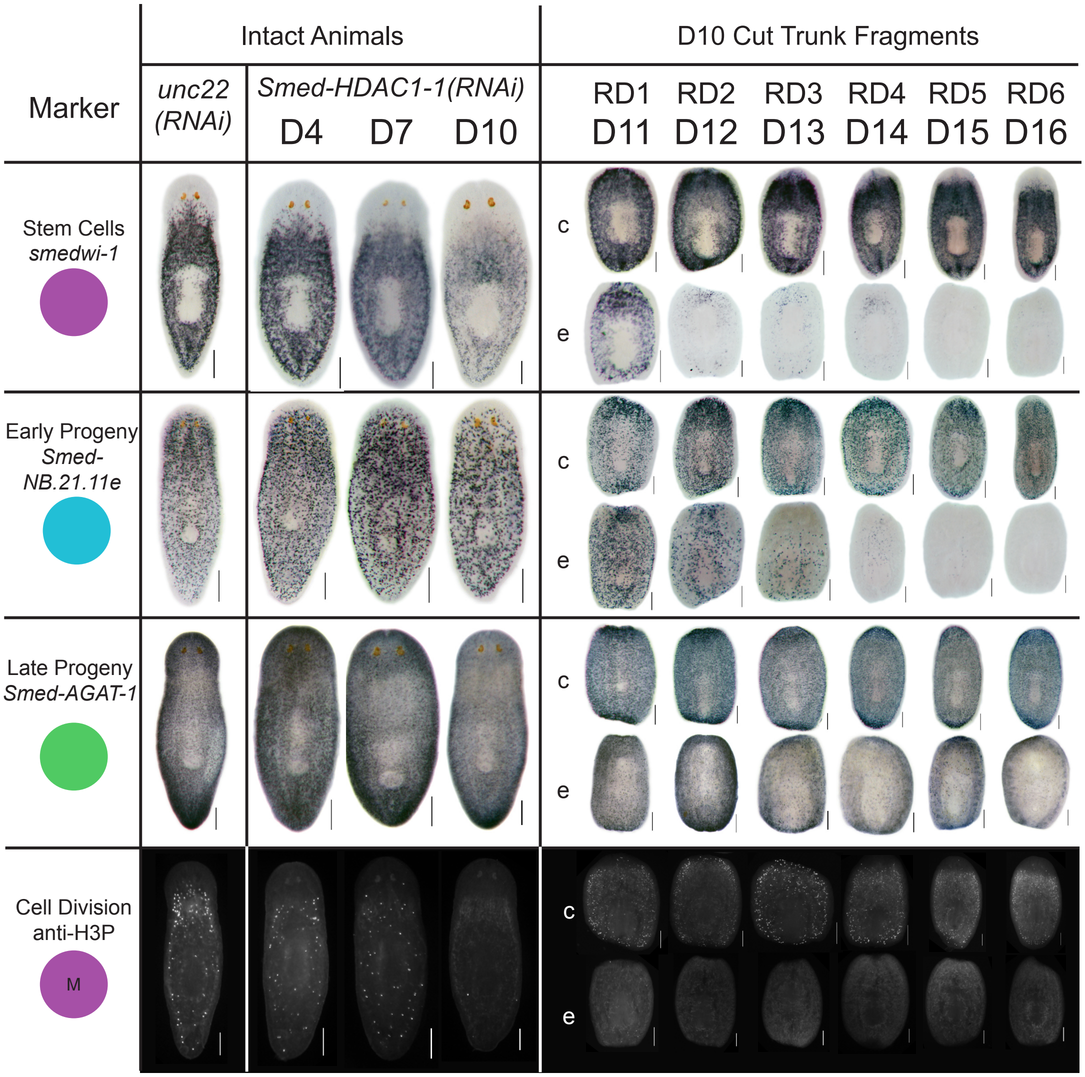


Figure 5.32: Stem Cell Lineage Analysis of *Smed-HDAC1-1(RNAi)* intact and amputated animals. Intact *Smed-HDAC1-1(RNAi)* animals were fixed on D4, D7 and D10 and amputated *Smed-HDAC1-1(RNAi)* animals (labeled “e” for experimental) were fixed on D11 through D16 (RD1 through RD6) and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control for intact animals and for amputated animals (labeled “c” for control).

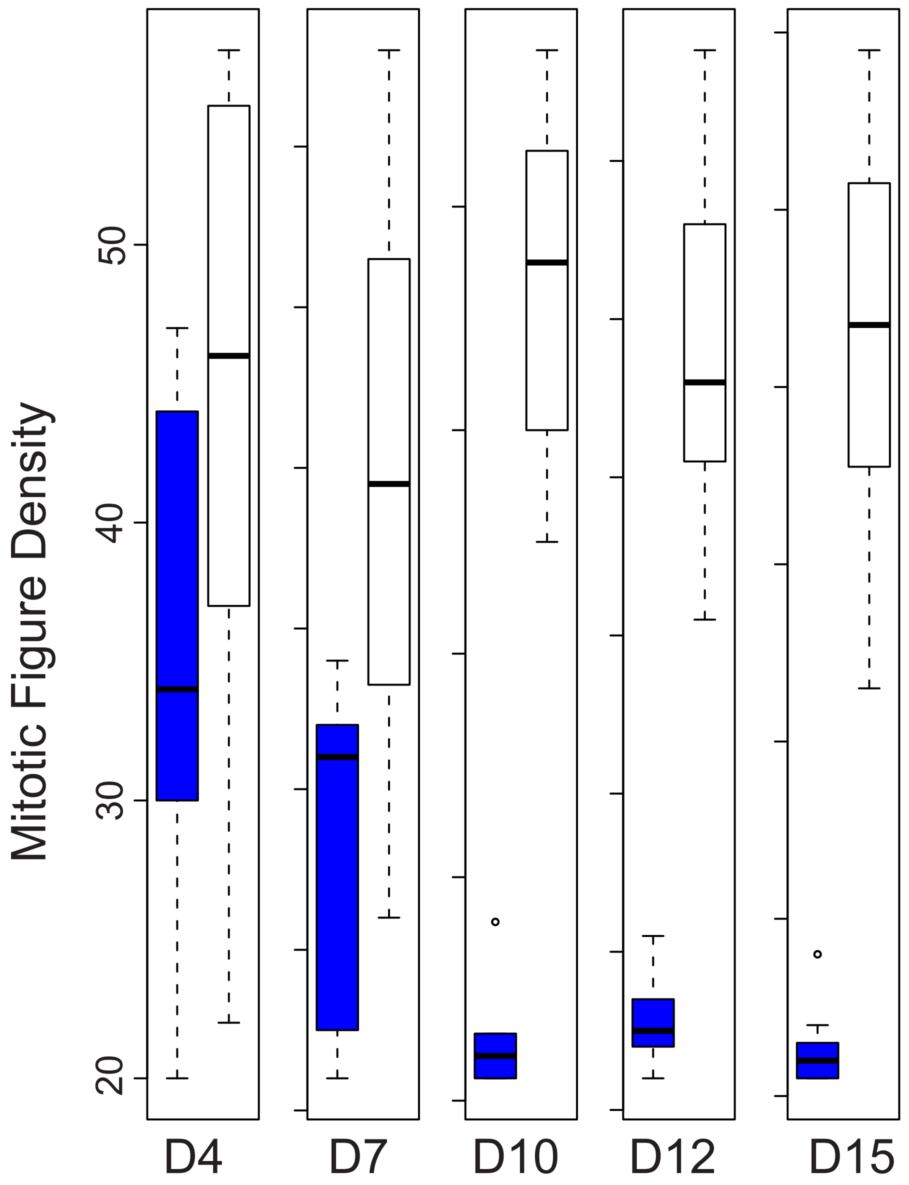


Figure 5.33: Mitotic stem cell density in *Smed-HDAC1-1(RNAi)* animals. For *Smed-HDAC1-1(RNAi)* (blue) and *unc22(RNAi)* (white) animals on D4, D7, D10, D12 and D15 mitotic figures were identified, counted and compared to the volume of the worm to produce a mitotic figure density. This graph is a boxplot, the box itself contains the middle 50% of the data. The upper edge of the box indicates the 75th percentile of the data set, and the lower edge indicates the 25th percentile. The line in the box indicates the median value of the data. The ends of the vertical lines indicate the minimum and maximum data values, and the points outside the ends of the vertical lines are outliers or suspected outliers.

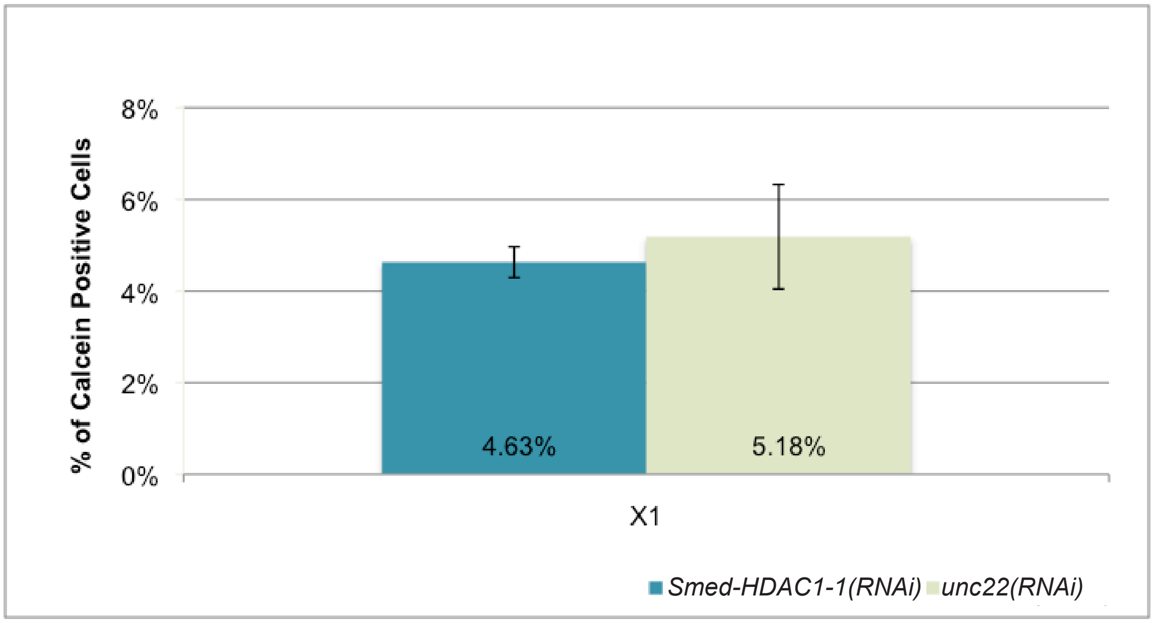


Figure 5.34: Stem cells are still present in D12 *Smed-HDAC1-1(RNAi)* animals. D12 *Smed-HDAC1-1(RNAi)* animals and unc22(RNAi) animals were dissociated and the stem cells were sorted and counted. The numbers reported are an average of a 220,000 total events over three separate sorts for *Smed-HDAC1-1(RNAi)* and for *unc22(RNAi)* animals.

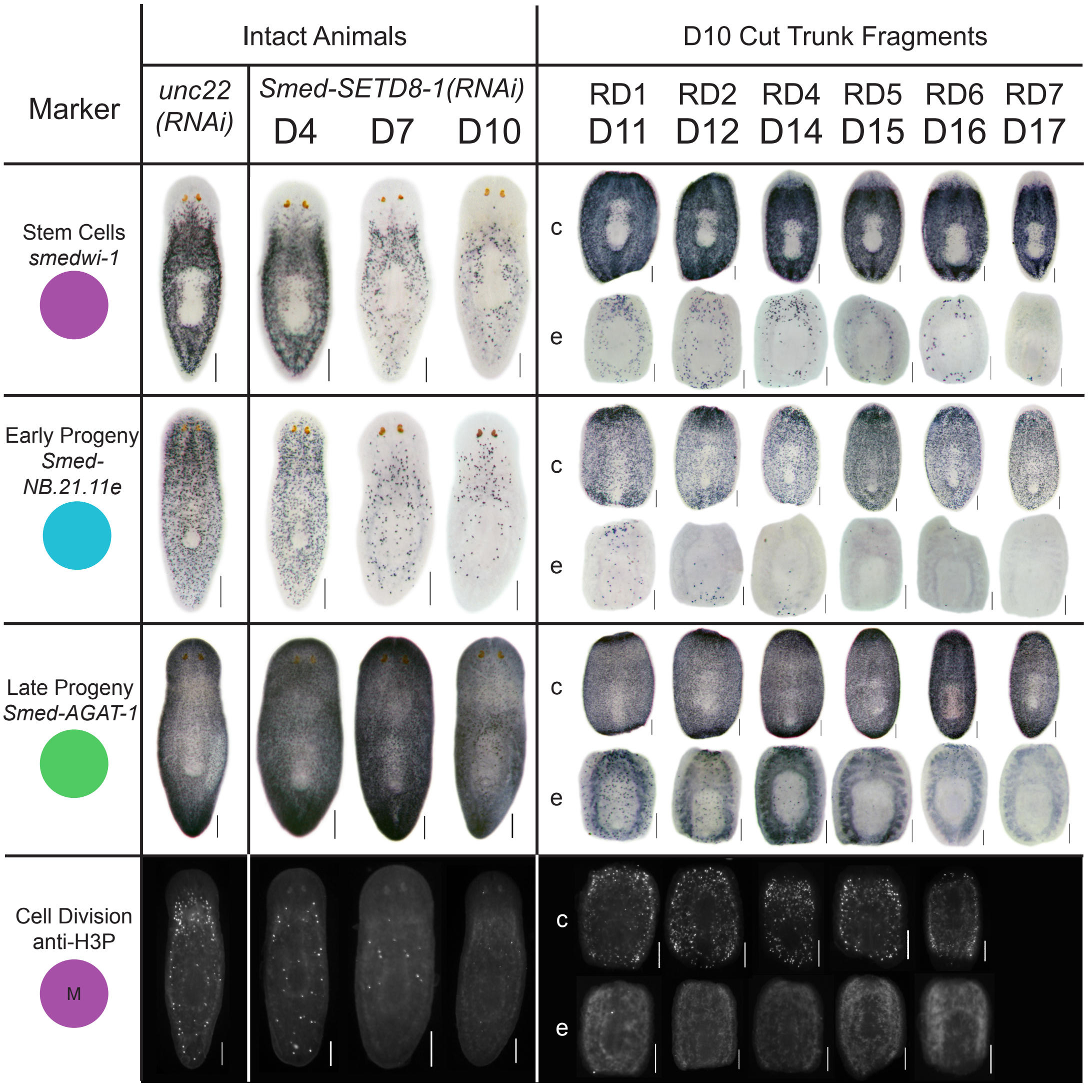


Figure 5.35: Stem Cell Lineage Analysis of *Smed-SETD8-1(RNAi)* intact and amputated animals. Intact *Smed-SETD8-1(RNAi)* animals were fixed on D4, D7 and D10 and amputated *Smed-SETD8-1(RNAi)* animals (labeled “e” for experimental) were fixed on D11 through D16 (RD1-2 and RD4 through RD7) and tested for the presence of the four markers *smedwi-1*, *Smed-NB.21.11e*, *Smed-AGAT-1* and anti-H3P. *unc22(RNAi)* animals were used as a control for intact animals and for amputated animals (labeled “c” for control).

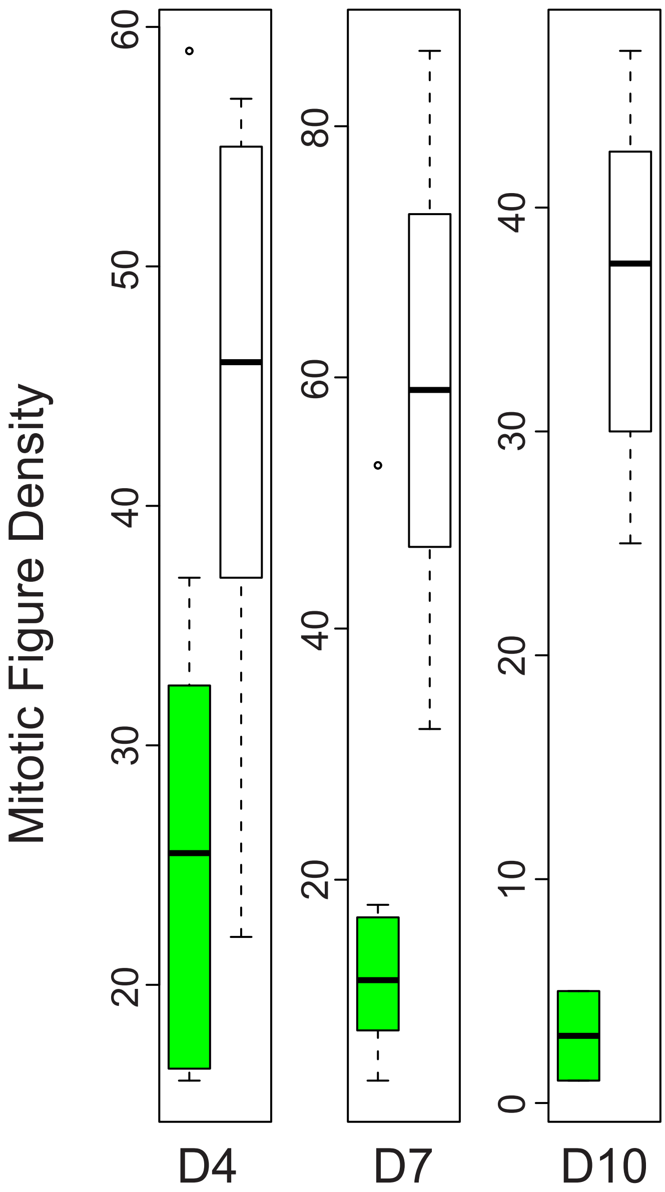


Figure 5.36: Mitotic stem cell density in *Smed-SETD8-1(RNAi)* animals. For *Smed-SETD8-1(RNAi)* (green) and *unc22(RNAi)* (white) animals on D4, D7 and D10 mitotic figures were identified, counted and compared to the volume of the worm to produce a mitotic figure density. This graph is a boxplot, the box itself contains the middle 50% of the data. The upper edge of the box indicates the 75th percentile of the data set, and the lower edge indicates the 25th percentile. The line in the box indicates the median value of the data. The ends of the vertical lines indicate the minimum and maximum data values, and the points outside the ends of the vertical lines are outliers or suspected outliers.

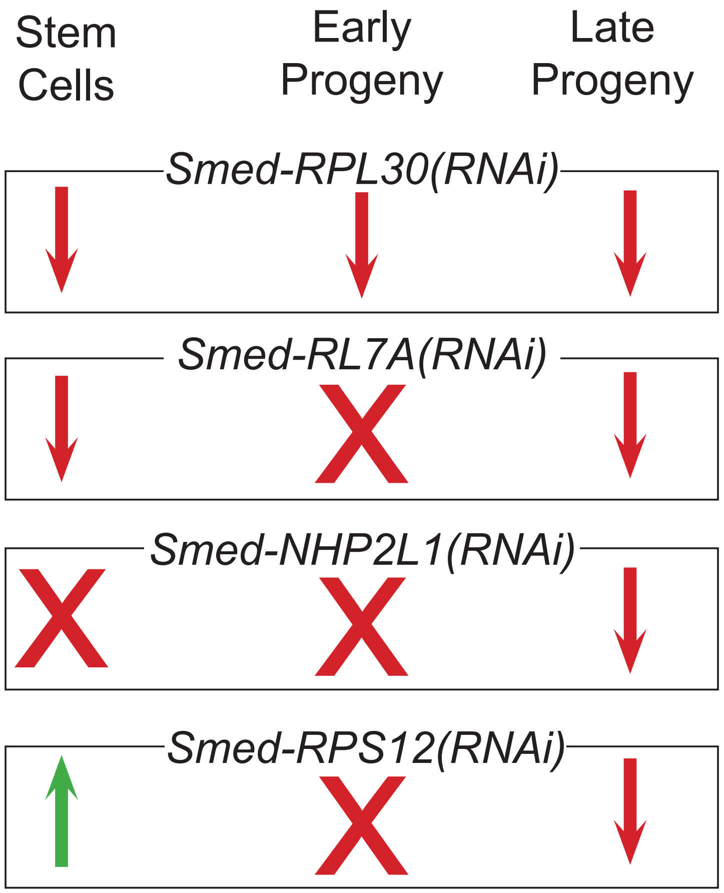


Figure 5.37: Summary of the effects of the loss of planarian L7Ae family of genes on the stem cell lineage. Red downward pointing arrows depict a gradual loss, while a red “X” describes very rapid loss of the indicated cell marker. A green upward pointing arrow is for the increase of the indicated marker.

Table 5.1: Histone numbers in *Schmidtea mediterranea*. The total unique nucleotide sequences of each of the four core histones, the linker histone H1, and histone variants are totaled in the column labeled ‘Count’.

Histone	Count
H1	6
H2A	247
H2B	18
H3	13
H4	9
H3.3	2
H2A.X	1
H2A.Z	2

Table 5.2: Catalog of each histone sequence in the *Schmidtea mediterranea* genome. The name of the MAKER annotation or the genomic location, if no MAKER exists, is listed for each unique nucleotide sequence of the core histones, the linker histone and histone variants. If a duplicate nucleotide sequence was identified it is listed as a "Duplicate Nucleotide Sequence".

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H1	mk4.004990.01.01	
H1	mk4.006907.05.01	
H1	mk4.014818.00.01	
H1	mk4.017587.00.01	
H1	mk4.018508.01.01	
H1	mk4.018999.00.01	
H2A	mk4.000031.07.01	
H2A	mk4.000038.12.01	
H2A	mk4.000045.11.01	
H2A	mk4.000056.00.01	
H2A	mk4.000056.09.01	
H2A	mk4.000059.14.01	
H2A	mk4.000066.09.01	
H2A	mk4.000086.10.01	
H2A	mk4.000100.09.01	
H2A	mk4.000102.21.01	
H2A	mk4.000103.01.01	
H2A	mk4.000114.12.01	mk4.000114.12.02
H2A	mk4.000124.05.01	
H2A	mk4.000129.03.01	
H2A	mk4.000160.09.01	
H2A	mk4.000167.01.01	
H2A	mk4.000268.06.01	
H2A	mk4.000301.11.01	
H2A	mk4.000337.10.01	
H2A	mk4.000350.12.01	
H2A	mk4.000371.06.01	
H2A	mk4.000389.01.01	
H2A	mk4.000422.11.01	
H2A	mk4.000444.07.01	
H2A	mk4.000468.01.01	
H2A	mk4.000480.03.01	
H2A	mk4.000503.19.01	
H2A	mk4.000546.21.01	
H2A	mk4.000573.01.01	
H2A	mk4.000586.04.01	
H2A	mk4.000610.06.01	

Table 5.2 continued

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H2A	mk4.000628.07.01	
H2A	mk4.000653.05.01	
H2A	mk4.000661.08.01	
H2A	mk4.000696.08.01	
H2A	mk4.000696.10.01	
H2A	mk4.000772.06.01	
H2A	mk4.000813.08.01	
H2A	mk4.000861.06.01	
H2A	mk4.000876.00.01	
H2A	mk4.000899.10.01	
H2A	mk4.000900.01.01	
H2A	mk4.000901.00.01	
H2A	mk4.000910.04.01	
H2A	mk4.000928.10.01	
H2A	mk4.000973.08.01	
H2A	mk4.000985.16.01	
H2A	mk4.001021.00.01	
H2A	mk4.001252.05.01	
H2A	mk4.001262.06.01	
H2A	mk4.001285.04.01	
H2A	mk4.001297.02.01	
H2A	mk4.001314.04.01	
H2A	mk4.001397.00.01	
H2A	mk4.001436.00.01	
H2A	mk4.001438.00.01	
H2A	mk4.001455.03.01	
H2A	mk4.001499.01.01	
H2A	mk4.001519.05.01	
H2A	mk4.001581.02.02	
H2A	mk4.001587.02.01	
H2A	mk4.001642.00.01	
H2A	mk4.001713.04.01	
H2A	mk4.001761.02.01	
H2A	mk4.002046.01.01	
H2A	mk4.002052.00.01	
H2A	mk4.002088.01.01	
H2A	mk4.002221.00.01	
H2A	mk4.002225.01.01	
H2A	mk4.002368.03.01	
H2A	mk4.002373.00.01	
H2A	mk4.002455.01.01	
H2A	mk4.002568.06.01	
H2A	mk4.002596.01.01	
H2A	mk4.002800.00.01	
H2A	mk4.002847.03.01	
H2A	mk4.002898.02.01	
H2A	mk4.002922.00.01	mk4.000688.04.01

Table 5.2 continued

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H2A	mk4.002959.02.01	
H2A	mk4.003116.02.01	
H2A	mk4.003169.00.01	
H2A	mk4.003365.00.01	
H2A	mk4.003443.01.01	
H2A	mk4.003555.01.01	
H2A	mk4.003568.03.01	
H2A	mk4.003682.00.01	
H2A	mk4.003696.00.01	
H2A	mk4.003804.04.01	
H2A	mk4.004173.01.01	
H2A	mk4.004225.02.01	
H2A	mk4.004315.05.01	
H2A	mk4.004364.00.01	
H2A	mk4.004520.00.01	
H2A	mk4.004629.01.01	
H2A	mk4.004660.01.01	
H2A	mk4.004754.04.01	
H2A	mk4.004793.01.02	mk4.004793.01.01
H2A	mk4.004850.06.01	
H2A	mk4.004866.01.01	
H2A	mk4.004873.02.01	
H2A	mk4.005181.01.01	
H2A	mk4.005310.01.01	
H2A	mk4.005494.07.01	
H2A	mk4.005504.00.01	
H2A	mk4.005523.00.01	
H2A	mk4.005542.00.01	
H2A	mk4.005699.02.01	
H2A	mk4.005994.01.01	
H2A	mk4.006188.02.01	
H2A	mk4.006492.01.01	
H2A	mk4.006729.01.01	
H2A	mk4.006761.00.01	
H2A	mk4.006861.00.01	
H2A	mk4.006950.00.01	
H2A	mk4.007436.00.01	
H2A	mk4.007648.00.01	
H2A	mk4.007699.00.01	
H2A	mk4.007728.00.01	
H2A	mk4.007909.00.01	
H2A	mk4.008019.00.01	
H2A	mk4.008168.04.01	
H2A	mk4.008251.01.01	
H2A	mk4.008399.02.01	
H2A	mk4.008811.02.01	
H2A	mk4.008818.01.01	

Table 5.2 continued

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H2A	mk4.008902.00.01	
H2A	mk4.008943.00.01	
H2A	mk4.009041.00.01	
H2A	mk4.009142.04.01	
H2A	mk4.009269.01.01	
H2A	mk4.009270.00.01	
H2A	mk4.009396.00.01	
H2A	mk4.009608.02.01	
H2A	mk4.009656.00.01	
H2A	mk4.009756.00.01	
H2A	mk4.009795.00.01	
H2A	mk4.010006.01.01	
H2A	mk4.010066.03.01	
H2A	mk4.010068.00.01	mk4.009889.01.01
H2A	mk4.010079.02.01	
H2A	mk4.010335.01.01	
H2A	mk4.010360.00.01	
H2A	mk4.010442.00.01	
H2A	mk4.010468.00.01	
H2A	mk4.010787.00.01	
H2A	mk4.010814.00.01	
H2A	mk4.010870.00.02	
H2A	mk4.011130.01.01	
H2A	mk4.011239.01.01	
H2A	mk4.011360.00.01	
H2A	mk4.011416.01.01	
H2A	mk4.011499.00.01	
H2A	mk4.011619.01.01	
H2A	mk4.011929.03.01	
H2A	mk4.012034.07.01	
H2A	mk4.012112.00.01	
H2A	mk4.012122.00.01	
H2A	mk4.012497.00.01	
H2A	mk4.012657.00.01	
H2A	mk4.012803.01.01	
H2A	mk4.012857.00.01	
H2A	mk4.013070.01.01	
H2A	mk4.013297.00.01	
H2A	mk4.013376.00.01	
H2A	mk4.013824.00.01	
H2A	mk4.013851.07.01	
H2A	mk4.013901.00.01	
H2A	mk4.014132.00.01	
H2A	mk4.014261.00.01	
H2A	mk4.014794.00.01	
H2A	mk4.014851.00.01	
H2A	mk4.014899.00.01	

Table 5.2 continued

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H2A	mk4.015024.00.01	
H2A	mk4.015042.00.01	
H2A	mk4.016020.00.01	
H2A	mk4.016231.00.01	
H2A	mk4.016263.01.01	
H2A	mk4.016797.01.01	mk4.051622.01.01
H2A	mk4.016810.05.01	
H2A	mk4.016918.01.01	mk4.018403.01.01
H2A	mk4.017079.00.01	
H2A	mk4.017302.00.02	
H2A	mk4.017365.01.01	
H2A	mk4.017535.00.01	
H2A	mk4.017677.02.01	
H2A	mk4.018087.00.01	
H2A	mk4.018219.01.01	
H2A	mk4.018403.01.01	
H2A	mk4.018669.01.01	mk4.011515.01.01
H2A	mk4.018714.00.01	
H2A	mk4.018759.00.01	
H2A	mk4.018961.00.01	
H2A	mk4.019478.00.01	
H2A	mk4.019624.00.01	
H2A	mk4.019679.01.01	
H2A	mk4.019733.00.01	
H2A	mk4.019915.02.01	
H2A	mk4.020443.00.01	
H2A	mk4.020573.01.01	
H2A	mk4.020659.03.01	
H2A	mk4.020750.00.01	
H2A	mk4.020889.01.01	
H2A	mk4.020967.00.01	
H2A	mk4.021210.00.01	
H2A	mk4.021383.00.01	
H2A	mk4.021421.01.01	
H2A	mk4.021772.02.01	
H2A	mk4.021972.02.01	
H2A	mk4.022214.00.01	
H2A	mk4.022413.00.01	
H2A	mk4.023072.00.01	
H2A	mk4.023376.00.01	
H2A	mk4.024058.00.01	
H2A	mk4.024106.01.01	
H2A	mk4.024181.00.01	
H2A	mk4.024306.01.01	
H2A	mk4.024390.00.01	
H2A	mk4.024448.00.01	
H2A	mk4.024604.00.01	

Table 5.2 continued

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H2A	mk4.024869.00.01	
H2A	mk4.025746.01.01	
H2A	mk4.026348.00.01	
H2A	mk4.026528.01.01	
H2A	mk4.027196.00.01	
H2A	mk4.027685.00.01	
H2A	mk4.027700.00.01	
H2A	mk4.028562.01.01	
H2A	mk4.028756.00.01	
H2A	mk4.030165.00.01	
H2A	mk4.030341.00.01	
H2A	mk4.030583.00.01	
H2A	mk4.030979.00.01	
H2A	mk4.032105.00.01	
H2A	mk4.032362.00.01	
H2A	mk4.032859.00.01	
H2A	mk4.032889.00.01	
H2A	mk4.033241.00.01	
H2A	mk4.035649.00.01	
H2A	mk4.040346.00.01	
H2A	mk4.041017.00.01	
H2A	mk4.043068.01.01	
H2A	mk4.043126.00.01	
H2A	mk4.043795.00.01	
H2A	mk4.049150.00.01	
H2A	mk4.051622.01.01	
H2A.X	mk4.045984.00.01	mk4.002385.04.01
H2A.Z	mk4.005139.02.01	
H2A.Z	mk4.005676.02.01	
H2B	mk4.000063.03.01	
H2B	mk4.000124.04.01	
H2B	mk4.000525.05.01	
H2B	mk4.000572.09.01	
H2B	mk4.000688.06.01	
H2B	mk4.000688.07.01	
H2B	mk4.000756.08.01	
H2B	mk4.002212.01.01	
H2B	mk4.006479.00.01	
H2B	mk4.011360.01.01	
H2B	mk4.012806.01.01	
H2B	mk4.013070.00.01	
H2B	mk4.014609.00.01	
H2B	mk4.014939.01.01	
H2B	mk4.015823.02.01	
H2B	mk4.019784.02.01	
H2B	mk4.021105.00.01	
H2B	mk4.023146.01.01	mk4.005020.01.01

Table 5.2 continued

Histone Name	MAKER ID or Genomic Location	Duplicate Nucleotide Sequences
H2B	v31.001736:20312..20722	
H2B	mk4.045597.00.01	mk4.008285.00.01
H3	mk4.000166.18.01	
H3	mk4.000192.00.01	
H3	mk4.000503.06.01	
H3	mk4.001251.06.01	
H3	mk4.001274.01.01	
H3	mk4.002039.00.01	
H3	mk4.002106.05.01	
H3	mk4.003160.00.01	
H3	mk4.006511.01.01	
H3	mk4.009200.01.01	
H3	mk4.009630.00.01	
H3	mk4.017017.00.01	
H3	mk4.018913.00.01	
H3.3	mk4.003583.00.01	mk4.001027.00.01
H3.3	mk4.005509.00.01	mk4.002008.04.01,mk4.005767.02.01,mk4.003741.00.01,mk4.000688.08.01,mk4.000688.02.01
H4	mk4.000809.00.01	mk4.003127.04.01,mk4.013216.00.01,mk4.017320.01.01
H4	mk4.000858.05.01	
H4	mk4.001066.07.01	mk4.003127.04.01,mk4.013216.00.01,mk4.017320.01.01
H4	mk4.001345.01.01	
H4	mk4.003066.03.01	
H4	mk4.003904.01.01	
H4	mk4.005158.02.01	
H4	mk4.006244.00.01	
H4	mk4.021260.01.01	

Table 5.3: Comparative species histone gene numbers. Total number of histone sequences identified in the genomes of *S. mediterranea*, *H. sapiens*, *D. melanogaster*, and *C. elegans* with the use of Blast and custom scripts.

	S. mediterranea	***H. sapiens***	***D. melanogaster***	***C. elegans***
H1	6	26	24	8
H2A	249	116	26	19
H2B	18	105	23	17
H3	15	184	49	28
H4	9	50	25	16

Table 5.4: Histone Clusters in *Schmidtea mediterranea*. Catalog of identifiable histone clusters with the genomic contig ID in which the cluster was found, the histones present in each cluster, and the MAKER ID for each histone.

Genomic Contig ID	Histone Name	MAKER ID
v31.000056	H2A	mk4.000056.00.01
	H2A	mk4.000056.09.01
v31.000114	H2A	mk4.000114.12.01
	H2A	mk4.000114.12.02
v31.000124	H2B	mk4.000124.04.01
	H2A	mk4.000124.05.01
v31.000503	H3	mk4.000503.06.01
	H2A	mk4.000503.19.01
v31.000688	H3.3	mk4.000688.02.01
	H2A	mk4.000688.04.01
	H2B	mk4.000688.06.01
	H2B	mk4.000688.07.01
	H3.3	mk4.000688.08.01
v31.000696	H2A	mk4.000696.08.01
	H2A	mk4.000696.10.01
v31.004793	H2A	mk4.004793.01.01
	H2A	mk4.004793.01.02
v31.11360	H2A	mk4.011360.00.01
	H2B	mk4.011360.01.01
v31.013070	H2B	mk4.013070.00.01
	H2A	mk4.013070.01.01

Table 5.5: Antibodies tested in whole animals.

Antibody	Catalog #	1^{o} Ab dilution	Positive for Signal
H3K9K14-Ac	Upstate 06-599	1:500	Yes
H3K27-Me3	Upstate 07-449	1:100	No
H3K4-Me3	Abcam ab8580	1:500	No
H3K4-Me2	Upstate 07-030	1:100	Yes
H3K9-Me3	Upstate 07-442	1:500	Yes
H3S10-P	Upstate 05-817	1:200	Yes
H3.3S31-P	Upstate 07-679	1:500	No
H3	Upstate 06-755	1:500	Yes
H4	Abcam ab10158	1:100	No
H2A	Upstate 07-146	1:100	No
H2BK12-Ac	Cell Siganling #2575	1:100	No
H2B	Abcam ab1790	1:100	No
H1	Upstate 05-629	1:500	Yes
5-Methylcytidine	Eurogentec BI-MECY	1:500	Yes
HDAC1	Upstate 06-720	1:100	Yes
HDAC2	Upstate 05-814	1:100	Yes
HDAC3	Upstate 05-813	1:100	Yes
HP1	Abcam ab9057	1:100	No

Table 5.6: Visualization and analysis of H3K9K14-Ac in a mixed population of cells. Three experiments (#3, #8 and #11) resulted in 39 DAPI positive cells that were imaged and tracked by Cell ID (ex. 3.09, experiment #3 cell number 9). The pixels per cell for the fluorescence of the H3K9K14-Ac and DAPI signal were measured and recorded and a ratio of these two values was taken. Image of the actual cells that were used for measurements can be found in the figures indicated in the Figure ID column.

	Figure ID		Average Fluorescence (pixels/cell)		Pixel Ratio
Cell ID	5.12	5.14	H3K9k14-Ac	DAPI	AC/DAPI
3.09	D	M	160.1	92.6	1.7
8.03	M	L	53.9	60.7	0.9
3.11			34.7	108.4	0.3
3.08	C	K	34.6	108.6	0.3
11.11			13.6	43.7	0.3
11.08			17.6	62.2	0.3
11.04			13	59.7	0.2
8.06	J		14.8	69.2	0.2
3.15			18.4	106.5	0.2
8.12			16	95.9	0.2
3.02	G	J	17	105.7	0.2
8.07			11.2	73.8	0.2
8.04			11.8	82.5	0.1
11.03			8.1	59.8	0.1
3.04	I		13.4	111.4	0.1
3.03	H	I	11.7	98.2	0.1
11.09			7	59	0.1
8.05			11.1	95.8	0.1
11.02			9	78.2	0.1
11.1			8	74.9	0.1
3.1			10.7	101.8	0.1
11.12	L	H	5.8	58.9	0.1
11.17		C	4.9	50.2	0.1
11.05			5.7	60.5	0.1
8.09		F	8.2	88	0.1
11.01		D	6	67.5	0.1
8.11		G	8.9	102.1	0.1
3.01	F		7	86.2	0.1
8.08			5.3	66.4	0.1
3.12	A		8.8	111.5	0.1
11.14			4.6	58.7	0.1
11.13	K	E	4.9	63.9	0.1
3.14			8.6	113.3	0.1
3.06			5.9	84.6	0.1
3.05			6.4	98	0.1
3.13	B		5.5	87.2	0.1
11.06		A	2.5	54.2	0
3.07	E	B	4.2	107.2	0
8.01			2.3	71.6	0

Table 5.7: H3K9-Me3 and H3K27-Me3 in stem cells and differentiated cells.

	Stem Cells	**Differentiated Cells**
H3K9-Me3	13% (31/233)	76% (158/208)
H3K27-Me3	88% (244/278)	94% (171/182)

Table 5.8: Classes and counts of epigenetic modifying enzyme genes in *Schmidtea mediterranea*. Counts of classes of enzymes found in the genome and the counts of the genes that were positively cloned and sequenced.

Genes	Found in Genome	Cloned and Sequenced
DNA De/Methylases	11	11
Histone Phosphorylases	3	3
Histone Acetyltransferases	26	25
Histone Methyltransferases	24	23
Histone Deacetylases	14	14
Histone Demethylases	12	11
Histone Deubiquinases	1	0
Count	91	87

Table 5.9: Primers used to clone *S. mediterranea* histone modifying enzymes.

Gene ID	Gene Name	Forward Primer	Reverse Primer
1	Smed-HDAC1-1	ATTGTCCTGTATTTGATGGG	ACATTCTTACGGTTGTCACC
2	Smed-DNMT1AP-1	TTCTTCATAACATATGGGGG	CGTAAAATCCAACTGGAGAG
3	Smed-DNMT2-1	TATTTGGTACGCCAAACTG	GTACAAACTGATGTCGTTGC
5	Smed-HDAC2-1	AGACTGCATGTTTTCGATTC	GTCAGACTTAGACGACCGAG
6	Smed-HDAC5-2	AATCTCCAATGACTGGTCC	AGATGAGGCTGGTATTTTCC
7	Smed-HDAC1-2	TAAAGATATTGGTGCTGGTG	ATGCACAGCTTCATGTACTC
8	Smed-HDAC5-1	GACGTTTAGCTATGGTTTGG	CATTACTACTCGACCTTCGG
9	Smed-Sin3-1	GATAATCGAGAGATTGCCAG	ACAAAAAGTGTAGTGGACGC
10	Smed-HDAC8-1	GCTAATTTGTCGAATGATCG	GAGTCCAAACCTTAGCAGTG
11	Smed-Sin3-2	ACACCAGGATTTGATTGAAG	CTTTCCATTCGACATCTTTC
12	Smed-HDAC3-1	ATATATCAACCGTTTCGAGC	ACACCAAGTCTATCACAGCC
13	Smed-SAP18-2	TTTATCATCTGCTGTTGTGG	CTTCCACAAAATCTCCAATC
14	Smed-SAP18-1	AATGAGTGATAAGAAGTCGG	TGCATAAGATTAATCACCGC
15	Smed-SAP180-1	AACTTCTGAGGTGATTGTGG	AATTACTACCAACAGTTCCGAC
16	Smed-HDAC6-1	ATTCTCAATCCCATGTTCAG	GTACCGTTACCATGATGGAC
18	Smed-HDAC1-3	TTAGTCTGTAGGTTTGCATCAC	TATTTTTCTTTCCCAGCAGC
19	Smed-RL7A	TCCTAACAACAAAGCCAAAC	TTAAGTCCCATAATTCCACC
20	Smed-NHP2	CATGGCTCCAGAAAAATTAG	GGTTGCGGTAGGTTAAGAAC
21	Smed-SEBP2	ATTAGTTGCCAATAACGACG	CTCTTTCACATAGGTTTCGG
22	Smed-NH2PL1	ACCCCTTAGCTAAAAACGAG	GCAGTGTGAATTCCTTGAAC
23	Smed-RPS12	GACCGAAGAGAAAAGTGTTG	AGTTTGTGCAGTTATTTGGG
24	Smed-RPL30	ACCGAAGAAAAAGGTTAAGG	TCCTGGATCAGTAATTGCC
26	Smed-GCN5-1	TGTTGTTAATGCGATGAGAG	CGTTTCATTATAAACTCGGC
27	Smed-ELP3-1	GTCAGCCAGTGGAGTAATTC	ATTTATCAGTGGGCAACAAC
28	Smed-MYST1-1	GGACTCGTTTTTGTGTCATC	GCTAATTTGCAACTCCTTTC
29	Smed-NCOAT-2	GTGGCGTGAGTTATACAGTG	TCAGACCTAATCCAATGAGC
30	Smed-CBP-2	GTCAACAATAAAGCGGAAAC	CAATCGACAGTCAATGTTTC
31	Smed-TAF5-1	ATGTTAACTGGCATCGTACC	CAATGAAAAGCGAAGAGAAC
32	Smed-Ep300-1	AGAAGCCAATGCTAATGAAG	CAACTGTTTGCATGTCTCAC
33	Smed-YEATS4-1	ATGGTAGAACTCACGAGTGG	CCATATCTTCCAAAAACGC
34	Smed-GCN5-2	ATCTTTCAACGGGATATCAG	CTGGCATATTGTAAACCTCG
35	Smed-CBP-4	GAGAGATGGATTCTATTGCG	AAAATACGGAAGCTCAGTTG
36	Smed-CBP-1	TTATCCCACTGGAAAAGATG	CCAATTCTACCCGCTGACATTAGGTTCGTCATCC
37	Smed-CBP-3	ATACAGAACCATGGGACATC	ACATCTGGCAATCCTCATAG
38	Smed-NCOAT-1	GACATTTATTCAACGCAAGG	AAAACTCCCCGTAATTTCTC
39	Smed-MYST3-1	GCCTTCTTTGTAACAATTCG	TGCTCAGCATTGAGTGTTAG
40	Smed-MYST4-2	AGCTCATCCCACATGTTTAC	GAATTCGACTTTTTGCTGTC
41	Smed-MYST2-1	AATTGTTTTCAGAGAGTGCG	CTAACCACAATAACGTCGTG
42	Smed-Ep300-2	CTGGAGTTAATGGAAAAACG	CTCAACATTTGTGCTTGTTG

Table 5.9 Continued

Gene ID	Gene Name	Forward Primer	Reverse Primer
43	Smed-MYST4-1	TTCCACCTGGTAATGAAATC	ACGTTAGGATTGATGAGGG
44	Smed-HAT-1	TACACATCAGTTTTATGGCG	CATGCCTTCAACAACATTC
45	Smed-CBP-5	TGCCTACTCATACAATTCCC	CAATCGACAGTCAATGTTTC
46	Smed-CBP-6	AGATGTCAAAGGCAAGAATC	TAGCCGAGTTTCTTCATCAG
47	Smed-Ep300-3	GAATATGAGCAAAAACTGGC	CCTTGGAAGTTTTCTTGTTG
48	Smed-ELP2-1	ATCAGTTTACCGAGCATCAG	ATTGGACCCTAACAACTGTG
49	Smed-Tip60-1	ATCAGGACATCGTCACTAGG	GGCCTTTGCTGTAGTAGATG
50	Smed-MAK3-1	TGCTGAAACTGTGATTGAAG	ATAGCCGTTTATCTCGAATG
51	Smed-SUV42-1	GGATGAACGATTCAAGAGAC	GTAATTGTACATTCCCCGAC
52	Smed-SETB1-1	CAGTGTTGATAATGACGTGC	TCTGTACATGTAATCCCACG
53	Smed-CARM1-1	ACAGAAATGCGTAGAATTGG	TGTAACTTTGCATGAAGTGC
54	Smed-ANM1-2	TGGATACAAATGACATGACG	TCACCTCTTTTAACAGGGAC
55	Smed-ANM8-2	CCATGACGAAATGCTAAAAG	TCCGGGTTTAACCTCTATTC
56	Smed-NSD3-1	TTCTGGGGTAAGTCACAATC	AGCAACTTCCACAGAAAATC
57	Smed-NSD1-2	GGAGGAGAATTACTGTGCTG	ACTCGGATTCAGTAGTGTCG
58	Smed-NSD2-2	AGAAAAAGCAGCGAGTAAAC	ATTTACCTCGACAGAACCAG
59	Smed-SUV92-1	TTAGTGGTTGTCGATGTTTG	TATCCGAGAATTCTGTCACC
60	Smed-NSD2-1	GTAATAATGGCAAGGTCTCG	AGTATCACCGTTGACTGACC
61	Smed-SETD2-1	ATTCCCTTTAGCTCCAAAAC	TTTCCCACAATGTTATCCTC
62	Smed-TRR-1	CGGAATCGTTTGCTATTATC	CACAAACACGGGATTTTATC
63	Smed-NSD1-1	TTCTCAGGGATTTACCATTG	GAAGTTCAGGAAATTTGTGC
64	Smed-MLL3-1	ATTCCTGTGTTACTTGGTCG	TGTGACGATCTCTCACACAC
66	Smed-DOT1L-1	AAAACCAAGACAACTCCAAC	GTTATGCGAAACTTCAGAGG
67	Smed-ANM8-1	CGATGAAAACAATCTATCCG	ACTGAATTGTCCGAGTGTTC
68	Smed-SETD8-1	GATTTGTGGGCTAACTTACG	CGTTAATATTTCAGCCAAGG
69	Smed-SETD8-2	TCGTGGTGTTGAAAGTACAG	CCCGATCTCCATAATCATAC
70	Smed-ANMX-1	GCCCTGAAACAAGTTATACG	AACCAATGATCTCTCCAATG
71	Smed-ASH1L-1	ATCTAAACGAGAAGCAGCAG	AAAATAAGCCTCCTTCCATC
72	Smed-ANM1-1	ATTGAAGAATATGGAGCACG	CTGAATGTTTCCTCCATTTG
73	Smed-ASH2L-1	TTTGCTAGCATGTGTCAAAC	CTGACATTAGGTTCGTCATCC
74	Smed-SETB1-2	TCCGACTAAAGTTGAAAAGG	ACGAATATCTACCTCGAACG
75	Smed-JARID-3	TTACGTGTTTCGAATGTGAC	TCGAGTAACTCCAGTGATCC
77	Smed-LSD1-1	AATCAGCTAATCAAGCAAGC	TGCAACATTTTCTACGTCTG
78	Smed-JmjC-1	GCTATGAATTATGCCAGCTC	TTACTGCAACAGTATCGTCG
79	Smed-JARID-2	TCAATTTACTCCACGAGTCC	AGATCCATTTTCACGAACAC
80	Smed-JmjC-5	TTCCAATTCACTAAAGCCAC	ATCCATTGATCCAGTTATGG
81	Smed-JmjC-6	TGGGAATATTTTGTCTGTGG	ATTTCACAGAGATCTGGTCG
82	Smed-JmjC-2	CTTCCACACGATGATTTTTC	TTATTTTCTCCTCGACATCC
83	Smed-JmjC-4	ACCAGTTTATCGTCCAACAG	TGGTTTCTTATTCTTCTCGG
84	Smed-JmjC-7	ATTACGGATGTTACGACTGC	CATAGGCCTTGAAATTCTTG
85	Smed-SUV92-2	CGCTTGATTTCCAGTTTATC	TCTTCGCCTGCTTTAATATC
86	Smed-JmjC-4	ATTCTGAAATGCTCACACTG	GAGTACATTGTGAAAGAACGC
87	Smed-TAF1-1	TCAAGTGAATTTAAGGTGGC	AAGCAAGGAGAAGAAGAAGG
88	Smed-TAF1-1	TCAATATCGACTCATTTCCC	CTTCCTCTTTTTATAGGGGG

Table 5.9 Continued

Gene ID	Gene Name	Forward Primer	Reverse Primer
89	Smed-TAF1-2	CCATCATCGGATGTTAATTC	TCTCACAAATTGAAACCCTC
90	Smed-TAF1-3	CCTCTACGAATGTTACCGAC	GTTTTCCCTTGCGTATTTC
92	Smed-XPG-1	CACAGTACTGGCAGAGATATTG	TCGTAGTATTCATTCTCGGG
93	Smed-XPG-2	TCTCTCGAAATGACGGTTAC	TGAGACTGATCGATTTCAGG
94	Smed-FEN	AGCAGAACAATGCTGTAGTG	CAATAATCGCAACCGAGTAG

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990) 'Basic local alignment search tool', *J Mol Biol* 215(3): 403-10.

Barreto, G., Schafer, A., Marhold, J., Stach, D., Swaminathan, S. K., Handa, V., Doderlein, G., Maltry, N., Wu, W., Lyko, F. et al. (2007a) 'Gadd45a promotes epigenetic gene activation by repair-mediated DNA demethylation', *Nature* 445(7128): 671-5.

Barreto, G., Schäfer, A., Marhold, J., Stach, D., Swaminathan, S. K., Handa, V., Döderlein, G., Maltry, N., Wu, W., Lyko, F. et al. (2007b) 'Gadd45a promotes epigenetic gene activation by repair-mediated DNA demethylation', *Nature* 445(7128): 671-675.

Berger, S. L. (2007) 'The complex language of chromatin regulation during transcription', *Nature* 447(7143): 407-12.

Bernstein, B. E., Mikkelsen, T. S., Xie, X., Kamal, M., Huebert, D. J., Cuff, J., Fry, B., Meissner, A., Wernig, M., Plath, K. et al. (2006) 'A Bivalent Chromatin Structure Marks Key Developmental Genes in Embryonic Stem Cells', *Cell* 125(2): 315-326.

Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M. C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I. et al. (2003) 'The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003', *Nucleic Acids Res* 31(1): 365-70.

Cantarel, B. L., Korf, I., Robb, S. M., Parra, G., Ross, E., Moore, B., Holt, C., Sánchez Alvarado, A. and Yandell, M. (2008) 'MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes', *Genome Res* 18(1): 188-96.

Chambeyron, S. and Bickmore, W. A. (2004) 'Chromatin decondensation and nuclear reorganization of the HoxB locus upon induction of transcription', *Genes Dev* 18(10): 1119-30.

DeChiara, T. M., Robertson, E. J. and Efstratiadis, A. (1991) 'Parental imprinting of the mouse insulin-like growth factor II gene', *Cell* 64(4): 849-59.

Dovey, O. M., Foster, C. T. and Cowley, S. M. (2010) 'Histone deacetylase 1 (HDAC1), but not HDAC2, controls embryonic stem cell differentiation', *Proc Natl Acad Sci U S A* 107(18): 8242-7.

Dufourcq, P., Victor, M., Gay, F., Calvo, D., Hodgkin, J. and Shi, Y. (2002) 'Functional requirement for histone deacetylase 1 in Caenorhabditis elegans gonadogenesis', *Mol Cell Biol* 22(9): 3024-34.

Eilertsen, K. J., Floyd, Z. and Gimble, J. M. (2008) 'The epigenetics of adult (somatic) stem cells', *Crit Rev Eukaryot Gene Expr* 18(3): 189-206.

Eisenhoffer, G. T., Kang, H. and Sánchez Alvarado, A. (2008) 'Molecular Analysis of Stem Cells and Their Descendants during Cell Turnover and Regeneration in the Planarian Schmidtea mediterranea', *Cell Stem Cell* 3(3): 327-339.

Feng, J., Fouse, S. and Fan, G. (2007) 'Epigenetic regulation of neural gene expression and neuronal function', *Pediatr Res* 61(5 Pt 2): 58R-63R.

Guo, T., Peters, A. H. and Newmark, P. A. (2006) 'A Bruno-like gene is required for stem cell maintenance in planarians', *Dev Cell* 11(2): 159-69.

Gurley, K. A., Rink, J. C. and Sánchez Alvarado, A. (2008) 'B-Catenin Defines Head Versus Tail Identity During Planarian Regeneration and Homeostasis', *Science* 319(5861): 323-327.

Huen, M. S., Sy, S. M., van Deursen, J. M. and Chen, J. (2008) 'Direct interaction between SET8 and proliferating cell nuclear antigen couples H4-K20 methylation with DNA replication', *J Biol Chem* 283(17): 11073-7.

Koonin, E. V. (1997) 'Cell cycle and apoptosis: possible roles of Gadd45 and MyD118 proteins inferred from their homology to ribosomal proteins', *J Mol Med* 75(4): 236-8.

Koressaar, T. and Remm, M. (2007) 'Enhancements and modifications of primer design program Primer3', *Bioinformatics* 23(10): 1289-91.

Korf, I., Yandell, M. and Bedell, J. (2003) *BLAST*, Sebastopol, CA: O'Reilly & Associates.

Ma, D. K., Guo, J. U., Ming, G. L. and Song, H. (2009) 'DNA excision repair proteins and Gadd45 as molecular players for active DNA demethylation', *Cell Cycle* 8(10): 1526-31.

Malik, H. S. and Henikoff, S. (2003) 'Phylogenomics of the nucleosome', *Nature Structural Biology* 10(11): 882-891.

Mariño-Ramírez, L., Hsu, B., Baxevanis, A. D. and Landsman, D. (2005) 'The histone database: A comprehensive resource for histones and histone fold-containing proteins', *Proteins: Structure, Function, and Bioinformatics* 62(4): 838-842.

Marino-Ramirez, L., Jordan, I. K. and Landsman, D. (2006) 'Multiple independent evolutionary solutions to core histone gene regulation', *Genome Biol* 7(12): R122.

Molofsky, A. V., Pardal, R. and Morrison, S. J. (2004) 'Diverse mechanisms regulate stem cell self-renewal', *Current Opinion in Cell Biology* 16(6): 700-707.

Nishioka, K., Rice, J. C., Sarma, K., Erdjument-Bromage, H., Werner, J., Wang, Y., Chuikov, S., Valenzuela, P., Tempst, P., Steward, R. et al. (2002) 'PR-Set7 is a nucleosome-specific methyltransferase that modifies lysine 20 of histone H4 and is associated with silent chromatin', *Mol Cell* 9(6): 1201-13.

O'Neill, L. P., VerMilyea, M. D. and Turner, B. M. (2006) 'Epigenetic characterization of the early embryo with a chromatin immunoprecipitation protocol applicable to small cell populations', *Nature Genetics* 38(7): 835-841.

Park, J. A., Kim, A. J., Kang, Y., Jung, Y. J., Kim, H. K. and Kim, K. C. (2011) 'Deacetylation and methylation at histone H3 lysine 9 (H3K9) coordinate chromosome condensation during cell cycle progression', *Mol Cells* 31(4): 343-9.

Pearse, V., Pearse, J., Buchsbaum, M. and Buchsbaum, R. (1997) *Living Invertebrates*, Boston, MA: Blackwell Scientific Publications.

Pearson, B. J., Eisenhoffer, G. T., Gurley, K. A., Rink, J. C., Miller, D. E. and Sánchez Alvarado, A. (2009) 'Formaldehyde-based whole-mount in situ hybridization method for planarians', *Developmental Dynamics* 238(2): 443-450.

Peleg, S., Sananbenesi, F., Zovoilis, A., Burkhardt, S., Bahari-Javan, S., Agis-Balboa, R. C., Cota, P., Wittnam, J. L., Gogol-Doering, A., Opitz, L. et al. (2010) 'Altered histone acetylation is associated with age-dependent memory impairment in mice', *Science* 328(5979): 753-6.

Peterson, C. L. and Laniel, M. A. (2004) 'Histones and histone modifications', *Curr Biol* 14(14): R546-51.

Qiu, J. (2006) 'Epigenetics: Unfinished symphony', *Nature* 441(7090): 143-145.

Reddien, P. W. (2005) 'SMEDWI-2 Is a PIWI-Like Protein That Regulates Planarian Stem Cells', *Science* 310(5752): 1327-1330.

Reddien, P. W., Bermange, A. L., Murfitt, K. J., Jennings, J. R. and Sánchez Alvarado, A. (2005) 'Identification of genes needed for regeneration, stem cell function, and tissue homeostasis by systematic gene perturbation in planaria', *Dev Cell* 8(5): 635-49.

Robb, S. M. C., Ross, E. and Alvarado, A. S. (2007) 'SmedGD: the Schmidtea mediterranea genome database', *Nucleic Acids Research* 36(Database): D599-D606.

Rooney, A. P., Piontkivska, H. and Nei, M. (2002) 'Molecular evolution of the nontandemly repeated genes of the histone 3 multigene family', *Mol Biol Evol* 19(1): 68-75.

Rumbaugh, G. and Miller, C. A. (2011) 'Epigenetic changes in the brain: measuring global histone modifications', *Methods Mol Biol* 670: 263-74.

Sittman, D. B., Chiu, I. M., Pan, C. J., Cohn, R. H., Kedes, L. H. and Marzluff, W. F. (1981) 'Isolation of two clusters of mouse histone genes', *Proc Natl Acad Sci U S A* 78(7): 4078-82.

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G., Korf, I., Lapp, H. et al. (2002) 'The Bioperl toolkit: Perl modules for the life sciences', *Genome Res* 12(10): 1611-8.

Stilling, R. M. and Fischer, A. (2011) 'The role of histone acetylation in age-associated memory impairment and Alzheimer's disease', *Neurobiol Learn Mem*.

Sytnikova, Y. A., Kubarenko, A. V., Schafer, A., Weber, A. N. and Niehrs, C. (2011) 'Gadd45a is an RNA binding protein and is localized in nuclear speckles', *PLoS ONE* 6(1): e14500.

Taniura, H., Sng, J. C. and Yoneda, Y. (2007) 'Histone modifications in the brain', *Neurochem Int* 51(2-4): 85-91.

Thiriet, C. and Albert, P. (1995) 'Rapid and effective western blotting of histones from acid-urea-Triton and sodium dodecyl sulfate polyacrylamide gels: two different approaches depending on the subsequent qualitative or quantitative analysis', *Electrophoresis* 16(3): 357-61.

Thompson, J. D., Higgins, D. G. and Gibson, T. J. (1994) 'CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice', *Nucleic Acids Research* 22(22): 4673-4680.

Tonelli, R., Sartini, R., Fronza, R., Freccero, F., Franzoni, M., Dongiovanni, D., Ballarini, M., Ferrari, S., D'Apolito, M., Di Cola, G. et al. (2006) 'G1 cell-cycle arrest and apoptosis by histone deacetylase inhibition in MLL-AF9 acute myeloid leukemia cells is p21 dependent and MLL-AF9 independent', *Leukemia* 20(7): 1307-10.

Trahan, C., Martel, C. and Dragon, F. (2010) 'Effects of dyskeratosis congenita mutations in dyskerin, NHP2 and NOP10 on assembly of H/ACA pre-RNPs', *Hum Mol Genet* 19(5): 825-36.

Van Hooser, A., Goodrich, D. W., Allis, C. D., Brinkley, B. R. and Mancini, M. A. (1998) 'Histone H3 phosphorylation is required for the initiation, but not maintenance, of mammalian chromosome condensation', *J Cell Sci* 111 (Pt 23): 3497-506.

Verdin, E., Dequiedt, F. and Kasler, H. G. (2003) 'Class II histone deacetylases: versatile regulators', *Trends Genet* 19(5): 286-93.

Warner, J. R. and McIntosh, K. B. (2009) 'How common are extraribosomal functions of ribosomal proteins?', *Mol Cell* 34(1): 3-11.

White, R. J., Gottlieb, T. M., Downes, C. S. and Jackson, S. P. (1995) 'Cell cycle regulation of RNA polymerase III transcription', *Mol Cell Biol* 15(12): 6653-62.

Wool, I. G. (1996) 'Extraribosomal functions of ribosomal proteins', *Trends Biochem Sci* 21(5): 164-5.

Zhang, L., Eugeni, E. E., Parthun, M. R. and Freitas, M. A. (2003) 'Identification of novel histone posttranslational modifications by peptide mass fingerprinting', *Chromosoma* 112(2): 77-86.

CHAPTER 6

DISCUSSION

Schmidtea mediterranea is becoming a well-established model organism system for studying regeneration, tissue homeostasis, and adult stem cell biology. Many tools and techniques have been engineered, including bromodeoxyuridine incorporation, high-throughput in situ hybridization, dsRNA-mediated gene perturbation, microarray mRNA expression analysis, and fluorescence activated cell sorting. In fact, many vital findings have been made over the years to define planaria as an excellent model system but little or no work had been done on the study of epigenetic regulation of adult stem cell (ASC) function.

Before my thesis work, the only bioinformatic tool in use was a web accessible database of EST sequences and relevant homology and expression data, SmedDb (Sánchez Alvarado, 2002). This database was instrumental in the printing of our first microarray platform, and data obtained from experiments using this microarray were key in identifying markers for characterizing and studying the first planarian stem cell linage ever reported (Eisenhoffer et al., 2008). Data mined from SmedDb were also used in the first large-scale RNAi screen performed in the entire Platyhelminth phyllum (Reddien et al., 2005).

Prior to the genome assembly, I assisted every lab member in their search of genes of interest in the millions of shotgun sequence trace reads. With the genome assembly in place, I was fortunate to have assisted in the completion of the first release of MAKER, an easy-to-use annotation pipeline for genome sequence (Cantarel et al., 2008). My input resulted in the ability to directly load the output into the GMOD (http://gmod.org) genome browser, Gbrowse (Stein et al., 2002). This contribution was not just significant for the planarian community, but also for many emerging model organism research communities with a sequenced genome that lacked the ability to create a tool to share the genome and annotations with the world. I also created our genome browser, SmedGD (Robb et al., 2007) which integrates the genomic information with MAKER annotations, ESTs, WISH expression patterns and RNAi phenotypes and is accessed by individuals across the globe. In addition, I have assisted my fellow lab members by creating tools for batch sequence retrieval, batch nucleotide translation and longest open reading frame finding, batch primer design and organization, batch gene finding, and batch parsing of BLAST results.

I also contributed to the field of epigenetics in planarian. My work yielded a collection of optimized protocols for the examination of chromatin states, a catalog of almost 90 genes of epigenetic modifying enzymes with expression patterns and RNAi phenotypes, and a characterization of modifications in different cell populations.

There are many well-studied histone posttranslational modifications. These modifications typically correspond to a specific cellular event, such as double stranded DNA breaks, transcription and stages of the cell cycle. They are dynamic and possess a predictable nature. The use of antibodies against these modifications may be an informative method for separating the mixed staged cycling stem cell population. H3K9K14-Ac levels compared to DNA content may be a sliding scale indicator to the phase of the cell cycle (Figure 5.14). H4K20-Me could be used as a signal of the entrance into S phase, H3P marking the G2 to M transition, and an increase in H3K9-Me3 corresponding to DNA condensation of the M phase.

I also closely investigated six genes of enzymes in three different classes of epigenetic modification: histone deacetylation (*Smed-HDAC1-1*), histone methylation (*Smed-SETD8-1*) and putative DNA demethylation (*Smed-RL7A, Smed-NHP2L1, Smed-RPS12*, and *Smed-RPL30*). The disruption of the function of these six genes affected the progression of the stem cell lineage in different ways. *Smed-HDAC1-1(RNAi)* animals have stem cells that are likely unable to progress into the M phase. *Smed-SETD8-1(RNAi)* animals are possibly unable to enter S phase. The four L7Ae family members differentially affect the stem cell lineage. *Smed-RPL30(RNAi)* animals seem to be defective in stem cell maintenance. *Smed-RL7A* is likely involved in the stem cell selection of differentiation lineage pathway. Abrogation of *Smed-RPS12* results in an excess of mitotic figures, which could be due to the over proliferation of the stem cells and an inability to differentiate. *Smed-NHP2L1* may be playing an important role in the translation of mRNAs in stem cells and the resulting postmitotic early progeny or some other basic function of these two cell types, but not in the late progeny.

Further analysis is required to understand the mechanics of the changes to the stem cell lineage in each of these genes, including FACS, whole-mount immunohistochemsitry, immunocytochemistry, cell death assays, incorporation of halogenated thymidine analogs, global gene expression patterns either by deep sequencing or microarrays, and real time quantitative PCR.

Adult stem cells are an understudied population of cells and *S. mediterranea* is an ideal model that makes the investigation of ASC much more amenable. This is mainly due to the abundance and experimental accessibility of ASCs in this organism. This is especially true in the study of epigenetics of ASCs, as is attested by the variety of roles epigenetic modifying enzymes seem to have in stem cell function. With many methodologies needed for studying chromatin state in place, such as, but not limited to ChIP, and the ability to measure the changes of modifications in individual cells, and the bioinformatic tools I help generate, great strides can now be made in the field of ASC epigenetics in planaria. Large-scale studies such as ChIP-Seq could be used to discover genes differentially modified in ASC and their progeny. Whole genome tiling arrays using chromatin immunoprecipitated DNA is an excellent technique for the identification of regulatory regions of genes that are essential for stem cell maintenance, self-renewal and differentiation. There are many proteomic approaches that can be used to better characterize epigenetics in ASC as well. Mass spectrometry analysis of histone proteins from ASCs and other cell types could yield novel histone posttranslational modifications or combinations of modifications that are unique to ASCs.

In conclusion this body of work has encompassed many disciplines and has significantly contributed to the advancement of planarian research. Tools have been created such that high-throughput, large scale biological experimentation can be implemented (Orendt et al., 2006; Robb et al., 2007; Cantarel et al., 2008). It has been demonstrated that histone posttranslational modification levels may be used to differentiate between cell types and in staging stem cells in the cell cycle (Figure 5.14 and 5.15). Histone modifying enzymes are not ubiquitously expressed (Figure 5.16), though genes such as *HDAC1* are ubiquitously expressed in other organisms such as humans, mouse and *C. elegans* (Dufourcq et al., 2002; Verdin et al., 2003; Dovey et al., 2010). Gene perturbations can be administered in the adult planarian, which are animals that as adults have ASCs, regulate cell numbers and possess mechanisms for tissue homeostasis. In many other experimental organisms cell number regulation and tissue turn-over only occurs during developmental stages and many of these genes cause embryonic lethality when eliminated. In the future, planarians will incontestably have a major impact on novel findings in the biology of epigenetics in ASCs.

References

Cantarel, B. L., Korf, I., Robb, S. M., Parra, G., Ross, E., Moore, B., Holt, C., Sánchez Alvarado, A. and Yandell, M. (2008) 'MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes', *Genome Res* 18(1): 188-96.

Dovey, O. M., Foster, C. T. and Cowley, S. M. (2010) 'Histone deacetylase 1 (HDAC1), but not HDAC2, controls embryonic stem cell differentiation', *Proc Natl Acad Sci U S A* 107(18): 8242-7.

Dufourcq, P., Victor, M., Gay, F., Calvo, D., Hodgkin, J. and Shi, Y. (2002) 'Functional requirement for histone deacetylase 1 in Caenorhabditis elegans gonadogenesis', *Mol Cell Biol* 22(9): 3024-34.

Eisenhoffer, G. T., Kang, H. and Sánchez Alvarado, A. (2008) 'Molecular analysis of stem cells and their descendants during cell turnover and regeneration in the planarian Schmidtea mediterranea', *Cell Stem Cell* 3(3): 327-339.

Orendt, A. M., Haymore, B., Richardson, D., Robb, S. M. C., Sánchez Alvarado, A. and Facelli, J. C. (2006) Design, Implementation and Deployment of a Commodity Cluster for Periodic Comparisons of Gene Sequences. in L. T. Yang and M. Guo (eds.) *High-Performance Computing : Paradigm and Infrastructure*. Hoboken, NJ: John Wiley and Sons, Inc.

Reddien, P. W., Bermange, A. L., Murfitt, K. J., Jennings, J. R. and Sánchez Alvarado, A. (2005) 'Identification of genes needed for regeneration, stem cell function, and tissue homeostasis by systematic gene perturbation in planaria', *Dev Cell* 8(5): 635-49.

Robb, S. M. C., Ross, E. and Alvarado, A. S. (2007) 'SmedGD: the Schmidtea mediterranea genome database', *Nucleic Acids Research* 36(Database): D599-D606.

Sánchez Alvarado, A. (2002) 'The Schmidtea mediterranea database as a molecular resource for studying platyhelminthes, stem cells and regeneration', *Development* 129(24): 5659-5665.

Stein, L. D., Mungall, C., Shu, S., Caudy, M., Mangone, M., Day, A., Nickerson, E., Stajich, J. E., Harris, T. W., Arva, A. et al. (2002) 'The generic genome browser: a building block for a model organism system database', *Genome Res* 12(10): 1599-610.

Verdin, E., Dequiedt, F. and Kasler, H. G. (2003) 'Class II histone deacetylases: versatile regulators', *Trends Genet* 19(5): 286-93.

APPENDIX

CUSTOM SCRIPTS

blast2table_desc.pl modified from blast2table.pl (Korf et al., 2003).

```perl
#!/usr/bin/perl -w
use strict;
use Getopt::Std;
use vars qw($opt_p $opt_b $opt_e $opt_m $opt_n);
getopts('p:b:e:m:n:');
my $PERCENT = $opt_p ? $opt_p : 0;
my $BITS    = $opt_b ? $opt_b : 0;
my $EXPECT  = $opt_e ? $opt_e : 1e30;
my $START   = $opt_m ? $opt_m : 0;
my $END     = $opt_n ? $opt_n : 1e30;

my ( $Query, $Sbjct );
my $HSP = "";
while (<>) {
    if (/^Query=\s+(\S+)/) { outputHSP(); $Query = $1 }
    ## original line
    #elsif (/^>(\S+)/)         {outputHSP(); $Sbjct = $1}
    ##modifed by Sofia M.C. Robb to include description
    elsif (/^>(\S+ .+)/) { outputHSP(); $Sbjct = $1 }
    elsif (/^ Score = /) {
        outputHSP();
        my @stat = ($_);
        while (<>) {
            last unless /\S/;
            push @stat, $_;
        }
        my $stats = join( "", @stat );
        my ($bits)   = $stats =~ /(\d\S+) bits/;
        my ($expect) = $stats =~ /Expect\S* = ([\d\.\+\-e]+)/;
        $expect = "1$expect" if $expect =~ /^e/;
        my ( $match, $total, $percent ) =
          $stats =~ /Identities = (\d+)\/(\d+) \((\d+)%\)/;
        my $mismatch = $total - $match;

        $HSP = {
            bits     => $bits,
            expect   => $expect,
            mismatch => $mismatch,
            percent  => $percent,
            q_begin  => 0,
            q_end    => 0,
            q_align  => 0,
            s_begin  => 0,
            s_end    => 0,
            s_align  => ""
        };
    }
    elsif (/^Query:\s+(\d+)\s+(\S+)\s+(\d+)/) {
        $HSP->{q_begin} = $1 unless $HSP->{q_begin};
        $HSP->{q_end} = $3;
        $HSP->{q_align} .= $2;
    }
    elsif (/^Sbjct:\s+(\d+)\s+(\S+)\s+(\d+)/) {
        $HSP->{s_begin} = $1 unless $HSP->{s_begin};
        $HSP->{s_end} = $3;
        $HSP->{s_align} .= $2;
    }
}
outputHSP();

sub outputHSP {
    return unless $HSP;
    return if $HSP->{percent} < $PERCENT;
    return if $HSP->{bits} < $BITS;
    return if $HSP->{expect} > $EXPECT;
    return if ( $HSP->{q_begin} < $START or $HSP->{q_end} < $START );
    return if ( $HSP->{q_begin} > $END or $HSP->{q_end} > $END );
    foreach my $field (
        'percent', 'q_align', 'mismatch', 's_align', 'q_begin', 'q_end',
        's_begin', 's_end',   'expect',   'bits'
      )
    {
        print "$field not defined\n" if not defined $HSP->{$field};
    }
    print join( "\t",
        $Query,
        $Sbjct,
        $HSP->{percent},
        length( $HSP->{q_align} ),
        $HSP->{mismatch},
        countGaps( $HSP->{q_align} ) + countGaps( $HSP->{s_align} ),
        $HSP->{q_begin},
        $HSP->{q_end},
        $HSP->{s_begin},
        $HSP->{s_end},
        $HSP->{expect},
        $HSP->{bits} ),
      "\n";
    $HSP = "";
}

sub countGaps {
    my ($string) = @_;
    my $count = 0;
    while ( $string =~ /\-+/g ) { $count++ }
    return $count;
}
```

parseBlastTable_topHit.pl: parses the output of blast2table_deac.pl. Takes a e-value cut off and returns a reformatted list of all the top hits that are better than given e-value cut off or the default e-value of 1.

```
#!/usr/bin/perl -w

use strict;

my $infile = $ARGV[0];
open INFILE, "$infile";
my %hash;
my $evalue_cutoff = $ARGV[1] ? $ARGV[1] : 1;

while ( my $line = <INFILE> ) {
    chomp $line;
    my @line   = split /\t/, $line;
    my $id     = $line[0];
    my $hit    = $line[1];
    my $evalue = $line[10];
    if ( !defined( $hash{$id} ) ) {
        $hash{$id}{hit}    = $hit;
        $hash{$id}{evalue} = $evalue;

    }
    elsif ( $hash{$id}{evalue} > $evalue ) {
        $hash{$id}{hit}    = $hit;
        $hash{$id}{evalue} = $evalue;
    }

}

foreach my $id ( keys %hash ) {
    my $hit    = $hash{$id}{hit};
    my $evalue = $hash{$id}{evalue};
    print "$id\t$hit\t$evalue\n" if $evalue <= $evalue_cutoff;
}
```

parseBlast_redundantSearch.pl: identifies non-self named matches in a blast of a fasta file against itself.

```
#!/usr/bin/perl -w

#Parsing Blast reports

#Parse an existing Blast report from file:
use Bio::SearchIO;

$inBlast = $ARGV[0];
$outFile = "$inBlast.summary.redundantSearch.out";
open OUTFILE, ">$outFile";

my $io = new Bio::SearchIO(
    -format => 'blast',
    -file   => $inBlast
);
my $hash;
my $allCopies = "";
while ( my $result = $io->next_result() ) {
    my $q_ID     = $result->query_name();
    my @temp     = split /\|/, $q_ID;
    my $uniqueID = pop @temp;
    my $copies   = "";
    while ( my $hit = $result->next_hit ) {

        # process the Bio::Search::Hit::HitI object
        while ( my $hsp = $hit->next_hsp ) {

            # process the Bio::Search::HSP::HSPI object
            my $hsp_identity = $hsp->frac_identical;
            my $hit_name     = $hit->name;
            my $queryLen     = $result->query_length();
            my $hitLen       = $hit->length();
            if ( $hsp_identity == 1 ) {
                if ( $hit_name ne $uniqueID ) {
                    if ( $queryLen <= $hitLen )
                    {    ## if query length is > or = hit lenght keep query
                            #if not already in copy list add id
                        if ( $allCopies !~ /$uniqueID/ ) {
                            $hash{$uniqueID} .= $hit_name . ","
                              if $allCopies !~ /$hit_name/;
                            $allCopies .=
                              $hit_name . ",";    # if $copies !~ /$hit_name/ ;
                        }
                    }
                    else {    #delete it if shorter hit is already a key
                        if ( defined $hash{$hit_name}
                            and length $hash{$hit_name} > 1 )
                        {
                            $hash{$uniqueID} .= $hash{$hit_name};
                            delete $hash{$hit_name};
                        }
                    }
                }
                elsif ( !defined $hash{$uniqueID} ) {
                    $hash{$uniqueID} = '';
                }
            }
        }
    }
}
foreach my $ID ( sort keys %hash ) {

    my $copies = $hash{$ID} ? $hash{$ID} : '';
    if ($copies) {
        $copies =~ s/$ID,?//;
        my @copies = split /,/, $copies;
        foreach my $copy (@copies) {
            delete $hash{$copy};
        }
    }
}
foreach my $ID ( sort keys %hash ) {
    $hash{$ID} =~ s/$ID,?//;
    $hash{$ID} =~ s/,$//;
    print "$ID\t$hash{$ID}\n";
    print OUTFILE "$ID\n";
}
```

printImageJMacro.pl: creates ImageJ macro files for a directory of directories of directories. Each top directory can contain many sub-directories. Each sub-directory is a for an individual snap shot and contains two directories one containing a DAPI image file and one containing a Texas Red image file. The macro will convert the images to 8-bit, set thresholds, create masks, and produce summary files.

```
#!/usr/bin/perl -w

use strict;

my $headDir = shift;
$headDir =~ s/\/$//;
my @sub_dirs = `ls -d  $headDir/*/`;

foreach my $dir (@sub_dirs) {
    chomp $dir;
    $dir =~ s/\/$//;
    next
      unless -e "$dir/DAPI - n000000.tif"
          and -e "$dir/Texas Red - n000000.tif"
    ;    #next dir unless these two files exist

    print '
DAPI = 0;
TEXAS = 0;
MASK=0;
open("', $dir, '/DAPI - n000000.tif");
run("8-bit");
setThreshold(14, 255);
run("Analyze Particles...", "size=10-1000 circularity=0.00-1.00 show=Masks
display exclude clear include summarize record");
run("Invert");
title=getTitle();
if (title == "Mask of DAPI - n000000.tif"){
        DAPI = 1;
}
open("', $dir, '/Texas Red - n000000.tif");
run("8-bit");
setThreshold(14, 255);
run("Analyze Particles...", "size=10-1000 circularity=0.00-1.00 show=Masks
display exclude clear include summarize record");
run("Invert");
title=getTitle();
if (title == "Mask of Texas Red - n000000.tif"){
      TEXAS = 1;
}
if (DAPI){
    if (TEXAS){
        imageCalculator("Add create", "Mask of DAPI - n000000.tif","Mask of
Texas Red - n000000.tif");
          //run("Image Calculator...", "image1=[Mask of DAPI - n000000.tif]
operation=Add image2=[Mask of Texas Red - n000000.tif] create");
        setThreshold(0, 254);
        run("Analyze Particles...", "size=10-1000 circularity=0.00-1.00
show=Masks display exclude clear include summarize record");
        title=getTitle();
        if (title == "Mask of Result of Mask"){
            MASK = 1;
        }
        selectWindow("Mask of DAPI - n000000.tif");
        close();
        selectWindow("Mask of Texas Red - n000000.tif");
        close();
    }
}
if (MASK){
       selectWindow("Mask of Result of Mask");
       saveAs("Tiff", "', $dir, 'Mask of Result of Mask.tif");
       close();
}
run("RGB Merge...", "red=[Texas Red - n000000.tif] green=*None* blue=[DAPI -
n000000.tif] keep");
selectWindow("RGB");
saveAs("Tiff", "', $dir, '_mergeRGB.tif");
while (nImages>0) {
          selectImage(nImages);
          close();
      }
selectWindow("Summary");
saveAs("Text" ,"', $dir, '_Summary.txt");
run("Clear Results");
selectWindow("Summary");
run("Close");
';

    print "\n\n\n";

}
```

convertImageJSummary2tab.pl: converts a directory of ImageJ summary output files into one tab-delimited standard output that can be redirected into a new file.

```
#!/usr/bin/perl -w

use strict;

my $dir = shift;

$dir =~ s/\/$//;

my @files = <$dir/*_Summary.txt>;
my %cells;
foreach my $file (@files) {
    if ( $file =~ /$dir\/(.+)_Summary\.txt/ ) {
        my $fileName = $1;
        my ( $dapiCount, $positiveCount ) = ( 0, 0 );
        open INFILE, $file;
        while ( my $line = <INFILE> ) {

            chomp $line;
            if ( $line =~ /DAPI/ ) {
                my @fields = split "\t", $line;
                $dapiCount = $fields[1];
            }
            if ( $line =~ /Result of Mask/ ) {
                my @fields = split "\t", $line;
                $positiveCount = $fields[1];
            }
        }
        my $percentPositive =
          $positiveCount > 0 ? $positiveCount / $dapiCount : 0;
        print "$fileName\t$dapiCount\t$positiveCount\t$percentPositive\n";
    }

}
```

parseBlast_simple.pl: parses blastn, blastp, blastx, tblastn, tblastx text output files. This script takes the following options: -f blast output file, -e e-value cutoff –c the count of hit summaries wanted for each result.

```
#!/usr/bin/perl -w

#Parsing Blast reports

#Parse an existing Blast report from file:
use Bio::SearchIO;
use Getopt::Std;

%options = ();
my $optString = 'h:e:f:c:';

sub init() {
    getopts( $optString, \%options ) or usage();

    # like the shell getopt, "d:" means d takes an argument

    if ( defined $options{h} ) {
        usage();
    }

    if ( defined $options{f} ) {
        print "-f $options{f}\n";
    }
    else {
        usage();
    }
    print "Unprocessed by Getopt::Std:\n" if $ARGV[0];
}

sub usage() {
    print STDERR <<EOF;
This program does ...
usage: $0 [-h] [-e e-value] [-f file]

       -h     : this (help) message
       -e     : the cutoff e-value (default is 1)
       -c     : the number of hits you want (default is 1)
       -f file      : blast output file

       example: $0 -e 1e-5 -c 5 -f blastOut
EOF
    exit;
}

init();

my ( $evalue, $inBlast, $userHitCount );

if ( defined $options{e} ) {
    $evalue = $options{e};
}
else {
    $evalue = 1;
}
print "-e $evalue\n";
```

```
if ( defined $options{c} ) {
    $userHitCount = $options{c};
}
else {
    $userHitCount = 1;
}
print "-c $userHitCount\n";
$inBlast = $options{f};
$outFile = "$inBlast.parsed.out";
open OUTFILE, ">$outFile";

my $io = new Bio::SearchIO(
    -format => 'blast',
    -file   => $inBlast
);

my %hits;

print OUTFILE
"id\tqueryDesc\tquerylength\thit_name\thit_desc\thitlength\tevalue\tidentity\ts
imilarity\n";
while ( my $result = $io->next_result() ) {
    my $count = 0;
    while ( my $hit = $result->next_hit ) {
        my $hit_signif = $hit->significance;
        if ( $hit_signif <= $evalue ) {
            my $hit_name    = $hit->name;
            my $hit_desc    = $hit->description;
            my $id          = $result->query_name();
            my $queryDesc   = $result->query_description();
            my $querylength = $result->query_length;
            my $hitlength   = $hit->length;

            #my $hsp = $hit->next_hsp;
            while ( my $hsp = $hit->next_hsp ) {

                #last if $hsp->evalue >  $evalue;
                my $identity        = $hsp->frac_identical();
                my $numberConserved = $hsp->num_conserved;
                my $similarity      = $hsp->frac_conserved;
                my $frame           = $hsp->query->frame;
                my $hstart          = $hsp->start('hit');
                my $hstop           = $hsp->end('hit');
                my $strand          = $hsp->strand('query');
                $strand = $strand > 0 ? '+' : '-';
                $frame++;
                $frame = $strand . $frame;
                print OUTFILE
"$id\t$queryDesc\t$querylength\t$hit_name\t$hit_desc\t$hitlength\t$hit_signif\t
$identity\t$similarity\n";
                last    #if only want hsp
            }    #end HSP loop
            $count++;
            last if $count == $userHitCount;

            #$userHitCount=0 if you want just the top hit
        }
    }
}
```

parseBlast_hsp.pl: parses blastx output files, specifically of amino acid sequences blasted against *S. med* genome assembly v31. It takes all the genome aligned sequence from multiple protein blasts and combines the nucleotide sequence to make the longest assembly from one genomic contig.

#!/usr/bin/perl -w

```
#Parsing Blast reports

#Parse an existing Blast report from file:
use strict;
use Bio::SearchIO;
use Getopt::Std;
use Number::Range;

my %options   = ();
my $optString = 'h:e:f:t:';

sub init() {
    getopts( $optString, \%options ) or usage();

    # like the shell getopt, "d:" means d takes an argument

    if ( defined $options{h} ) {
        usage();
    }

    if ( defined $options{f} ) {
        print "-f $options{f}\n";
    }
    else {
        usage();
    }
    print "Unprocessed by Getopt::Std:\n" if $ARGV[0];
}

sub usage() {
    print STDERR <<EOF;
This program does ...
usage: $0 [-h] [-e e-value] [-t 0|1] [-f file]

       -h     : this (help) message
       -e     : the cutoff e-value (default is 1)
       -t     : blast output type is standard(1)  or m8(0) (default is 1)
       -f file      : blast output file

       example: $0 -e 1e-5 -f blastOut
       example: $0 -e 1e-5 -t 1 -f blastOut
       example: $0 -e 1e-5 -t 0 -f blastOut
EOF
    exit;
}

init();

my ( $evalue, $inBlast, $type );
```

```
if ( defined $options{e} ) {
    $evalue = $options{e};
}
else {
    $evalue = 1;
}
print "-e $evalue\n";

if ( defined $options{t} ) {
    $type = $options{t};

    #if $type is 1 then it will be blast
    #else it will be blasttable
    $type = $type ? 'blast' : 'blasttable';
}
else {
    $type = 'blast';
}
print "-t $type\n";

$inBlast = $options{f};
my $outFile = "$inBlast.parsed.out";
open OUTFILE, ">$outFile";

my $io = new Bio::SearchIO(
    -format => $type,
    -file   => $inBlast
);

my %hits;
my %seqs;
while ( my $result = $io->next_result() ) {
    while ( my $hit = $result->next_hit ) {
        my $hit_signif = $hit->significance;
        if ( $hit_signif <= $evalue ) {
            my @seq_ranges;
            my $hit_name    = $hit->name;
            my $hit_desc    = $hit->description;
            my $id          = $result->query_name();
            my $queryDesc   = $result->query_description();
            my $hitDesc     = $hit->description();
            my $querylength = $result->query_length;
            my $hitlength   = $hit->length;
            while ( my $hsp = $hit->next_hsp ) {
                my $hsp_evalue = $hsp->evalue;
                my ( $start, $stop ) = $hsp->range('hit');
                my $fasta =
`fastacmd -d /common/data/smed_assembly_v31.nt -s $hit_name -L  $start,$stop`;
                my @fasta = split /\n/, $fasta;
                shift @fasta;
                my $seq = join '', @fasta;
                $seqs{$hit_name}{ $start . ".." . $stop } = $seq;
            }     #end hsp while loop
        }    #end hit while loop
    }
}

foreach my $id ( sort keys %seqs ) {
    my @sequence;
    foreach
      my $range ( sort { ( split /\.\./, $a )[0] <=> ( split /\.\./, $b )[0] }
        keys %{ $seqs{$id} } )
    {
        my $seq = $seqs{$id}{$range};
        my @seq = split //, $seq;
        my ( $start, $stop ) = split /\.\./, $range;
        my $length = scalar @seq;
        my $i      = $start;
        foreach my $bp (@seq) {
            $sequence[$i] = $bp;
            $i++;
        }
    }
    my $sequence;
    my $i = 0;
    my $ranges;
    my $last;    #0th element will always be undef,cause seqs start with 1.
    foreach my $bp (@sequence) {
        if ($bp) {
            $sequence .= $bp if $bp;
        }
        if ( !$last and $bp ) {
            $ranges .= $i . "..";
        }
        elsif ( $last and !$bp ) {
            $ranges   .= ( $i - 1 ) . ",";
            $sequence .= "NNN";
        }
        elsif ( $bp and $i + 1 == scalar @sequence ) {
            $ranges .= $i;
        }
        $last = $bp;
        $i++;
    }
    print OUTFILE ">$id($ranges)\n$sequence\n";
}
```

dna2orfFasta.pl: produces a fasta file with the 6 frame translation of any number of nucleotide sequences given in fasta format

```
#!/usr/bin/perl -w

use Bio::SeqIO;
use strict;

######################################################################
#
#this first section will translate each fasta entry into 6 reading
#frames. The translated seq will be printed to a file called
#$file.6frameTranslation.fasta
#
#Can change the min orf length by changing $minORFLen
#
#######################################################################

#first get file from command prompt
my $ORIGINAL_file = shift @ARGV;
my $file;

if ( $ORIGINAL_file =~ /\/common\/data\/(.+)/ ) {
    $file = $1;
    print "$file\n";
}
else {
    $file = $ORIGINAL_file;
}

my $minORFLen = 90;    #nucleotide numbering
open OUTFASTA, ">$file.ORF.fasta";

my $outfile = "$file.6frameTranslation.fasta";
open TRANSLATION, ">$outfile";

#input file will be a fasta file of nucleotides
my $in = Bio::SeqIO->new( -file => $ORIGINAL_file, '-format' => 'Fasta' );

#loop thru each sequence in fasta file
while ( my $seq = $in->next_seq() ) {
    my $id        = $seq->id;
    my $seqLength = length( $seq->seq );
    my ( $description, $translation );
    my @frames = ( 0, 1, 2 );

    foreach my $frame (@frames) {
        $translation = $seq->translate( undef, undef, $frame )->seq;

        #save new description
        $description = "Frame +" . ( $frame + 1 );

        #writting to the output file
        print TRANSLATION
          ">$id $description totalBP=$seqLength\n$translation\n";

        $translation = $seq->revcom->translate( undef, undef, $frame )->seq;

        #save new description
        $description = "Frame -" . ( $frame + 1 );
        #writting to the output file
        print TRANSLATION
          ">$id $description totalBP=$seqLength\n$translation\n";

    }

}
#######################################################################
#
#The second part of this program will open the file with the 6 frame
translations
#and print each ORF to a new fasta file
#
########################################################################

#open protienFasta.txt
my $protein_in = Bio::SeqIO->new( -file => "$outfile", '-format' => 'Fasta' );

print "ProteinInFile = $outfile\n";

my $seqID;

#loop thru each protein in this fasta
while ( my $seq = $protein_in->next_seq() ) {

    #get the id
    $seqID = $seq->id;

    #get frame and total sequence length
    my ( $frame, $seqLen );
    my $description = $seq->description;
    if ( $description =~ /Frame ([+|-]\d) totalBP=(\d+)/ ) {
        $frame  = $1;
        $seqLen = $2;
    }

    #get the sequence
    my $protein = $seq->seq;

    while ( $protein =~ /\*?([\w]+\*?)/g ) {
        my $orf    = $1;
        my $orfLen = length($1);
        if ( $orfLen >= ( $minORFLen / 3 ) ) {

            #pos returns end of m//
            my $orfStart = ( pos($protein) - $orfLen );

            #convert computer postion to biologically sig postion
            $orfStart = $orfStart + 1;

            my ( $bp_start, $bp_stop ) =
              aa2bp( $orfStart, $orfLen, $frame, $seqLen );

            print OUTFASTA
              ">$seqID(ORF:$bp_start..$bp_stop) Frame $frame\n$orf\n";
        }
    }
}

sub aa2bp {
    my ( $ORFstart, $ORFlen, $frame, $seqLen ) = @_;
    my $ORFstop = $ORFstart + $ORFlen - 1;
    my ( $bp_start, $bp_stop );
    my $aaX3_start = $ORFstart * 3;
    my $aaX3_stop  = $ORFstop * 3;

    if ( $frame eq "+1" ) {
        $bp_start = $aaX3_start - 2;
        $bp_stop  = $aaX3_stop + 0;
    }
    elsif ( $frame eq "+2" ) {
        $bp_start = $aaX3_start - 1;
        $bp_stop  = $aaX3_stop + 1;
    }
    elsif ( $frame eq "+3" ) {
        $bp_start = $aaX3_start + 0;
        $bp_stop  = $aaX3_stop + 2;
    }
    elsif ( $frame eq "-1" ) {
        $bp_stop  = $seqLen - $aaX3_start + 3;
        $bp_start = $seqLen - $aaX3_stop + 1;
    }
    elsif ( $frame eq "-2" ) {
        $bp_stop  = $seqLen - $aaX3_start + 2;
        $bp_start = $seqLen - $aaX3_stop + 0;
    }
    else {
        $bp_stop  = $seqLen - $aaX3_start + 1;
        $bp_start = $seqLen - $aaX3_stop - 1;
    }

    if ( $bp_stop <= 0 ) {
        $bp_stop = 1;
    }
    elsif ( $bp_start == 0 ) {
        $bp_start = 1;
    }
    elsif ( $bp_stop > $seqLen ) {
        $bp_stop = $seqLen;
    }
    elsif ( $bp_start > $seqLen ) {
        $bp_start = $seqLen;
    }

    return ( $bp_start, $bp_stop );
}
```

dna2longest_orfFasta.pl: takes a multi sequence nucleotide fasta and returns the longest open reading frame amino acid fasta. The longest ORF is defined as the longest the sequence can be without including a stop (/*?([\w]+*?)/).

```
#!/usr/bin/perl -w

use Bio::SeqIO;
use strict;
use Data::Dumper;

######################################################################
#
#this first section will translate each fasta entry into 6 reading
#frames. The translated seq will be printed to a file called
#$file.6frameTranslation.fasta
#
#Can change the min orf length by changing $minORFLen
#
#######################################################################

#first get file from command prompt
my $ORIGINAL_file = shift @ARGV;
my $file;

if ( $ORIGINAL_file =~ /\/common\/data\/(.+)/ ) {
    $file = $1;
    print "$file\n";
}
else {
    $file = $ORIGINAL_file;
}

my $minORFLen = 90;    #nucleotide numbering
open OUTFASTA, ">$file.ORF.fasta";

my $outfile = "$file.6frameTranslation.fasta";
open TRANSLATION, ">$outfile";

#input file will be a fasta file of nucleotides
my $in = Bio::SeqIO->new( -file => $ORIGINAL_file, '-format' => 'Fasta' );

#loop thru each sequence in fasta file
while ( my $seq = $in->next_seq() ) {
    my $id        = $seq->id;
    my $seqLength = length( $seq->seq );
    my ( $description, $translation );
    my @frames = ( 0, 1, 2 );

    foreach my $frame (@frames) {
        $translation = $seq->translate( undef, undef, $frame )->seq;

        #save new description
        $description = "Frame +" . ( $frame + 1 );

        #writting to the output file
        print TRANSLATION
          ">$id $description totalBP=$seqLength\n$translation\n";

        $translation = $seq->revcom->translate( undef, undef, $frame )->seq;
        #save new description
        $description = "Frame -" . ( $frame + 1 );

        #writting to the output file
        print TRANSLATION
          ">$id $description totalBP=$seqLength\n$translation\n";

    }

}
close TRANSLATION;
######################################################################
#
#The second part of this program will open the file with the 6 frame
translations
#and print each ORF to a new fasta file
#
#######################################################################

#open protienFasta.txt
my $protein_in = Bio::SeqIO->new( -file => "$outfile", '-format' => 'Fasta' );

print "ProteinInFile = $outfile\n";

my $seqID;
my %longestFrame;

#loop thru each protein in this fasta

while ( my $seq = $protein_in->next_seq() ) {

    #get the id
    $seqID = $seq->id;

    #get frame and total sequence length
    my ( $frame, $seqLen );
    my $description = $seq->description;
    if ( $description =~ /Frame ([+|-]\d) totalBP=(\d+)/ ) {
        $frame  = $1;
        $seqLen = $2;
    }

    #get the sequence
    my $protein = $seq->seq;

    while ( $protein =~ /\*?([\w]+\*?)/g ) {
        my $orf    = $1;
        my $orfLen = length($1);
        if ( $orfLen >= ( $minORFLen / 3 ) ) {

            #pos returns end of m//
            my $orfStart = ( pos($protein) - $orfLen );

            #convert computer postion to biologically sig postion
            $orfStart = $orfStart + 1;

            my ( $bp_start, $bp_stop ) =
              aa2bp( $orfStart, $orfLen, $frame, $seqLen );

            if ( !defined $longestFrame{$seqID} ) {
                $longestFrame{$seqID} = {
                    'frame'    => $frame,
                    'seq'      => $orf,
                    'length'   => $orfLen,
                    'bp_start' => $bp_start,
                    'bp_stop'  => $bp_stop,
                    'seqLen'   => $seqLen,
                };
            }
            else {
                if ( $longestFrame{$seqID}{length} < $orfLen ) {
                    $longestFrame{$seqID} = {
                        'frame'    => $frame,
                        'seq'      => $orf,
                        'length'   => $orfLen,
                        'bp_start' => $bp_start,
                        'bp_stop'  => $bp_stop,
                        'seqLen'   => $seqLen,
                    };
                }
            }

        #print OUTFASTA ">$seqID(ORF:$bp_start..$bp_stop) Frame
$frame\n$orf\n";
        }
    }
}

#print Dumper(\%longestFrame);
foreach my $seqID ( keys %longestFrame ) {
    my $bp_start = $longestFrame{$seqID}{bp_start};
    my $bp_stop  = $longestFrame{$seqID}{bp_stop};
    my $seqLen   = $longestFrame{$seqID}{seqLen};
    my $orfLen   = $longestFrame{$seqID}{length} * 3;

    print OUTFASTA
">$seqID(ORF:$bp_start..$bp_stop) $orfLen of $seqLen  Frame
$longestFrame{$seqID}{frame}
$seqID(ORF:$bp_start..$bp_stop)\n$longestFrame{$seqID}{seq}\n";
}

sub aa2bp {
    my ( $ORFstart, $ORFlen, $frame, $seqLen ) = @_;
    my $ORFstop = $ORFstart + $ORFlen - 1;
    my ( $bp_start, $bp_stop );
    my $aaX3_start = $ORFstart * 3;
    my $aaX3_stop  = $ORFstop * 3;

    if ( $frame eq "+1" ) {
        $bp_start = $aaX3_start - 2;
        $bp_stop  = $aaX3_stop + 0;
    }
    elsif ( $frame eq "+2" ) {
        $bp_start = $aaX3_start - 1;
        $bp_stop  = $aaX3_stop + 1;
    }
    elsif ( $frame eq "+3" ) {
        $bp_start = $aaX3_start + 0;
        $bp_stop  = $aaX3_stop + 2;
    }
    elsif ( $frame eq "-1" ) {
        $bp_stop  = $seqLen - $aaX3_start + 3;
        $bp_start = $seqLen - $aaX3_stop + 1;
    }
    elsif ( $frame eq "-2" ) {
        $bp_stop  = $seqLen - $aaX3_start + 2;
        $bp_start = $seqLen - $aaX3_stop + 0;
    }
    else {
        $bp_stop  = $seqLen - $aaX3_start + 1;
        $bp_start = $seqLen - $aaX3_stop - 1;
    }

    if ( $bp_stop <= 0 ) {
        $bp_stop = 1;
    }
    elsif ( $bp_start == 0 ) {
        $bp_start = 1;
    }
    elsif ( $bp_stop > $seqLen ) {
        $bp_stop = $seqLen;
    }
    elsif ( $bp_start > $seqLen ) {
        $bp_start = $seqLen;
    }

    return ( $bp_start, $bp_stop );
}
```

runPrimer3.pl: is a wrapper script for converting a fasta file of nucleotide sequences into Primer3 input format, running Primer3, and parsing the Primer3 output into a tab-delimited format.

```
#!/usr/bin/perl -w

# this script will call writePrimer3input.pl and parsePrimer3.pl
# runPrimer3.pl fasta.nt
# input a fasta file
# output a tab delimited file will all possible primers

use strict;

my $file = shift;

print "Primer3 is finding primers for the seqeunces in $file\n";
print "But first a proper IN FILE must be formated: $file.primer3.in\n";
`~/scripts/writePrimer3input.pl $file > $file.primer3.in`;
print "Primer3 is running on $file.primer3.in\n";
print
"With this command: cat $file.primer3.in  | primer3_core >
$file.primer3.out\n";
`cat $file.primer3.in  | primer3_core > $file.primer3.out`;
print "Your Primers are in $file.primers\n";
print `~/scripts/parsePrimer3.pl $file.primer3.out > $file.primers`;
```

writePrimer3input.pl: takes a fasta file of nucleotide sequences and creates Primer3 input.

```
#!/usr/bin/perl -w

#takes fasta and writes PRIMER3 input file

use strict;
use Bio::SeqIO;

my $file  = shift;
my $inseq = Bio::SeqIO->new(
    -file   => $file,
    -format => 'FASTA'
);

while ( my $seqIO = $inseq->next_seq ) {
    my $id  = $seqIO->id;
    my $seq = $seqIO->seq;
    my $len = length($seq);

    #my $optimal_len = $len < 1800 ? $len : 1800;
    my $optimal_len = $len;
    my $len_75      = int( $optimal_len * .75 );
    my $len_50      = int( $optimal_len * .50 );
    my $len_25      = int( $optimal_len * .25 );
    print "PRIMER_SEQUENCE_ID=$id
SEQUENCE=$seq
PRIMER_PRODUCT_SIZE_RANGE=$len_75-$optimal_len $len_50-$optimal_len $len_25-
$optimal_len 100-$len
PRIMER_GC_CLAMP=1
PRIMER_MIN_GC=40
PRIMER_OPT_GC_PERCENT=45
PRIMER_MAX_GC=80
PRIMER_OPT_SIZE=20
PRIMER_MIN_SIZE=18
PRIMER_MAX_SIZE=25
PRIMER_OPT_TM=55.0
PRIMER_MIN_TM=50.0
PRIMER_MAX_TM=65.0
=\n";
}
```

parsePrimer3.pl: pasrses Primer3 output into a tab-delimited file.

```
#!/usr/bin/perl

##parse primer3 output
##output tab delimited file

use strict;

my $file = shift;
open IN, $file;
my $id;
my $count;
my %primer;
while ( my $line = <IN> ) {
    if ( $line =~ /PRIMER_SEQUENCE_ID=(.+)/ ) {
        $id    = $1;
        $count = 0;
    }
    elsif ( $line =~ /^SEQUENCE=(.+)/ ) {
        $primer{$id}{1}{SEQUENCE} = $1;
    }
    elsif ( $line =~ /PRIMER_PRODUCT_SIZE_RANGE=.+\d+\-(\d+)$/ ) {
        $primer{$id}{1}{SEQ_LENGTH} = $1;
    }
    elsif ( $line =~ /PRIMER_LEFT_(\d*_)?SEQUENCE=(.+)/ ) {
        $count++;
        $primer{$id}{$count}{PRIMER_LEFT_SEQUENCE} = $2;
    }
    elsif ( $line =~ /PRIMER_RIGHT_(\d*_)?SEQUENCE=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_SEQUENCE} = $2;
    }
    elsif ( $line =~ /PRIMER_PAIR_PENALTY(_\d*)?=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_PAIR_PENALTY} = $2;
    }
    elsif ( $line =~ /PRIMER_LEFT(_\d*)?=(\d+),(\d+)/ ) {
        $primer{$id}{$count}{PRIMER_LEFT_START} = $2;
        $primer{$id}{$count}{PRIMER_LEFT_LEN}   = $3;
    }
    elsif ( $line =~ /PRIMER_RIGHT(_\d*)?=(\d+),(\d+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_START} = $2;
        $primer{$id}{$count}{PRIMER_RIGHT_LEN}   = $3;
    }
    elsif ( $line =~ /PRIMER_LEFT(_\d*)?_TM=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_LEFT_TM} = $2;
    }
    elsif ( $line =~ /PRIMER_RIGHT(_\d*)?_TM=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_TM} = $2;
    }
    elsif ( $line =~ /PRIMER_LEFT(_\d*)?_GC_PERCENT=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_LEFT_GC} = $2;
    }
    elsif ( $line =~ /PRIMER_RIGHT(_\d*)?_GC_PERCENT=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_GC} = $2;
    }
    elsif ( $line =~ /PRIMER_PRODUCT_SIZE(_\d*)?=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_PRODUCT_SIZE} = $2;
    }
    else {
        next;
    }
}
```

```
open OUTFASTA, ">$file.primerProduct.fasta";
print
"ID\tSEQLength\tprimerSetNum\tprimerOrient\tproduct_size\tstart\tlen\ttm\tgc%\t
primerPairPenalty\n";
foreach my $id ( keys %primer ) {
    my $seqLen   = $primer{$id}{1}{SEQ_LENGTH};
    my $sequence = $primer{$id}{1}{SEQUENCE};
    foreach my $count ( sort { $a <=> $b } keys %{ $primer{$id} } ) {
        next if $count == 0;
        my $leftSeq       = $primer{$id}{$count}{PRIMER_LEFT_SEQUENCE};
        my $rightSeq      = $primer{$id}{$count}{PRIMER_RIGHT_SEQUENCE};
        my $leftStart     = $primer{$id}{$count}{PRIMER_LEFT_START};
        my $leftLen       = $primer{$id}{$count}{PRIMER_LEFT_LEN};
        my $rightStart    = $primer{$id}{$count}{PRIMER_RIGHT_START};
        my $rightLen      = $primer{$id}{$count}{PRIMER_RIGHT_LEN};
        my $leftTM        = $primer{$id}{$count}{PRIMER_LEFT_TM};
        my $rightTM       = $primer{$id}{$count}{PRIMER_RIGHT_TM};
        my $leftGC        = $primer{$id}{$count}{PRIMER_LEFT_GC};
        my $rightGC       = $primer{$id}{$count}{PRIMER_RIGHT_GC};
        my $primerPairPen = $primer{$id}{$count}{PRIMER_PAIR_PENALTY};
        my $productSize   = $primer{$id}{$count}{PRIMER_PRODUCT_SIZE};

        my $product = substr( $sequence, $leftStart, $productSize );
        print
"$id\t$seqLen\t$count\tleft\t$productSize\t$leftStart\t$leftLen\t$leftTM\t$left
GC\t$leftSeq\n";
        print
"$id\t$seqLen\t$count\tright\t$productSize\t$rightStart\t$rightLen\t$rightTM\t$
rightGC\t$rightSeq\n";
        print OUTFASTA
">$id(primerSet:$count) productSize=$productSize leftTM=$leftTM leftGC=$leftGC
leftSeq=$leftSeq rightTM=$rightTM rightGC=$rightGC
rightSeq=$rightSeq\n$product\n";
    }
}
```

runPrimer3_web.pl: a web version of runPrimer3.pl. This script takes multi-sequence nucleotide fasta from a web text box, creates Primer3 input, runs Primer3 and runs pasrePrimer3.pl.

```
#!/usr/bin/perl -w

# this script will write a primer3 input file and will run parsePrimer3.pl
# runPrimer3.pl -f fasta.nt  (Note -f is required)
# input a fasta file
# running parsePrimer3.pl will parse output and generate a tab delimited file
# with all possible primers

# may need to change path to parsePrimer3.pl: /common/bin/parsePrimer3.pl
# may need to change path to primer3 application: /common/bin/primer3_core

# this script uses temp files in the following lines
## open OUTFILE,  ">>/tmp/$time.in_primer3";
## `cat /tmp/$time.in_primer3 | /common/bin/primer3_core >
##            /tmp/$time.out_primer3`;
## my $output = `/common/bin/parsePrimer3.pl /tmp/$time.out_primer3` ;
## unlink ("/tmp/$time.in_primer3","/tmp/$time.out_primer3");

use strict;
use Getopt::Std;
use Bio::SeqIO;
use CGI ':standard';
use IO::String;

if ( !param ) {

    print header;
    print
      start_html('Run Primer3'),
      h1('Run Primer3 and get tab-delimited output'),

      start_multipart_form,

      #textbox for seq
"Input one or more sequences in <a
href='http://en.wikipedia.org/wiki/FASTA_format'>FASTA format</a><br>",

      textarea( -name => 'fasta', -rows => 5, -cols => 90 ),

      br,

      #end textbox

      br,
      br,
      "<u>Primer Length</u> ",
      "&nbsp;",
      "&nbsp;",
      "opt:", "&nbsp;",
      textfield( -name => 'primerLenOpt', -value => 20, -size => 3 ),
      "&nbsp;",
      "&nbsp;",
      "min:", "&nbsp;",
      textfield( -name => 'primerLenMin', -size => 3, -value => 10 ),
      "&nbsp;",
      "&nbsp;",
      "max:", "&nbsp;",
      textfield( -name => 'primerLenMax', -size => 3, -value => 25 ),
      br,
      br,
      "<u>Primer Tm</u>",
      "&nbsp;",
      "&nbsp;",
      "opt:", "&nbsp;", textfield( -name => 'tmOpt', -size => 3, -value => 55
),
      "&nbsp;",
      "&nbsp;",
      "min:", "&nbsp;", textfield( -name => 'tmMin', -size => 3, -value => 50
),
      "&nbsp;",
      "&nbsp;",
      "max:", "&nbsp;", textfield( -name => 'tmMax', -size => 3, -value => 65
),
      br,
      br,
      "<u>Primer GC Content</u>",
      "&nbsp;",
      "&nbsp;",
      'opt:', "&nbsp;", textfield( -name => 'gcOpt', -size => 3, -value => 50
),
      "&nbsp;",
      "&nbsp;",
      'min:', "&nbsp;", textfield( -name => 'gcMin', -size => 3, -value => 45
),
      "&nbsp;",
      "&nbsp;",
      'max:', "&nbsp;", textfield( -name => 'gcMax', -size => 3, -value => 60
),
      br,
      br,
      "<u>GC Clamp</u>",
      "&nbsp;",
      "&nbsp;",
      "&nbsp;", textfield( -name => 'gcClamp', -size => 3, -value => 1 ),
      br,
      br, "<u>Optimal Product Length (bp)</u>", "&nbsp;", "&nbsp;", "&nbsp;",
      textfield(
        -name  => 'optimalProductLength',
        -size  => 25,
        -value => "As long as possible"
      ),
      br,
      br,

      submit( -name => 'primer3', -value => 'Get Primers' ), end_form,
end_html;

}

#####################
#
# reading, storing cmd line variables
#
###################
if (param) {
    print header;
    print start_html('Your Primers');
    my $fasta = param('fasta');
    my $time  = time();
####################
    #
    # tests
    #
####################

    #test for fasta format
    my $found = $fasta =~ />/;

    if ( !$found ) {
        $fasta = ">UnknownSequnce\n$fasta";
    }

####################
    #
    #  stores command line variables into program variables
    #    -- or --
    #  if no cmd line variables are given it stores default values
    #
####################

    my ( $gcOpt, $gcMin, $gcMax ) =
      ( param('gcOpt'), param('gcMin'), param('gcMax') );
    my ( $tmOpt, $tmMin, $tmMax ) =
      ( param('tmOpt'), param('tmMin'), param('tmMax') );
    my ( $primerLenOpt, $primerLenMin, $primerLenMax ) =
      ( param('primerLenOpt'), param('primerLenMin'), param('primerLenMax') );
    my $gcClamp              = param('gcClamp');
    my $optimalProductLength = param('optimalProductLength');
####################
    #
    #  1.parsing of the required fasta file
    #  2.writing the primer3 input file
    #
####################
    my $stringfh = new IO::String($fasta);
    my $inseq    = Bio::SeqIO->new(
        -fh     => $stringfh,
        -format => 'fasta'
    );

    my $count = 0;
    while ( my $seqIO = $inseq->next_seq ) {
        my $id  = $seqIO->id;
        my $seq = $seqIO->seq;
        my $len = length($seq);
        if ( $optimalProductLength !~ /long/ ) {
            $len = $optimalProductLength;
        }

        #my $len = length($seq);
        my $len_75 = int( $len * .75 );
        my $len_50 = int( $len * .50 );
        my $len_25 = int( $len * .25 );

        if ( $len_25 <= $primerLenMax ) {
            print
"** WARNING ** $id sequence is too short. No primers will be made for this
sequence.<br><br>";
            next if $len_25 <= $primerLenMax;
        }
        open OUTFILE, ">>/tmp/$time.in_primer3";

        print OUTFILE

          "PRIMER_SEQUENCE_ID=$id
SEQUENCE=$seq
PRIMER_PRODUCT_SIZE_RANGE=$len_75-$len $len_50-$len $len_25-$len 100-$len
PRIMER_GC_CLAMP=$gcClamp
PRIMER_MIN_GC=$gcMin
PRIMER_OPT_GC_PERCENT=$gcOpt
PRIMER_MAX_GC=$gcMax
PRIMER_OPT_SIZE=$primerLenOpt
PRIMER_MIN_SIZE=$primerLenMin
PRIMER_MAX_SIZE=$primerLenMax
PRIMER_OPT_TM=$tmOpt
PRIMER_MIN_TM=$tmMin
PRIMER_MAX_TM=$tmMax
=\n";
        $count++;
    }

    if ($count) {  #proceed only if 1 or more appropriate sequences are
provided

####################
        #
        # running the primer3
        #
###################

        print "Primers for $count seqeunce(s).\n\n";

`cat /tmp/$time.in_primer3  | /common/bin/primer3_core >
/tmp/$time.out_primer3`;

        #print "Your Primers are in $file.primers.txt\n\n";

####################
        #
        # running the parser
        #
###################

        my $output = `/common/bin/parsePrimer3.pl /tmp/$time.out_primer3`;

        my @lines = split /\n/, $output;
        print "<table><tr><td>";
        print "<table border=1 align=left cellpadding=2>";

        foreach my $line (@lines) {
            print "<tr align=center>";
            my @fields = split /\t/, $line;
            foreach my $field (@fields) {

                #print "<td><pre>$field</pre></td>";
                print "<td><font face=courier size=2>$field</font></td>";
            }
            print "</tr>";
        }
        print "</table>";
        print "</td></tr>";
        #print "<tr><td>";
        #print "<br>tab-delimited format:<br>";
        #print "<pre>", $output , "</pre>";
        #print "</tr></tr>";
        print "</table>";
    }
    close OUTFILE;
    unlink( "/tmp/$time.in_primer3", "/tmp/$time.out_primer3" );
    print end_html;
}
```

parsePrimer3.pl: slightly modified version to work with the web version of the runPrimer3_web.pl

```
#!/usr/bin/perl

##parse primer3 output from the web version of runPrimer3_web.pl
##output tab delimited file

use strict;

my $file = shift;
open IN, $file;
my $id;
my $count;
my %primer;
while ( my $line = <IN> ) {
    if ( $line =~ /PRIMER_SEQUENCE_ID=(.+)/ ) {
        $id    = $1;
        $count = 0;
    }
    elsif ( $line =~ /^SEQUENCE=(.+)/ ) {
        $primer{$id}{1}{SEQUENCE} = $1;
    }
    elsif ( $line =~ /PRIMER_PRODUCT_SIZE_RANGE=.+\d+\-(\d+)$/ ) {
        $primer{$id}{1}{SEQ_LENGTH} = $1;
    }
    elsif ( $line =~ /PRIMER_LEFT_(\d*_)?SEQUENCE=(.+)/ ) {
        $count++;
        $primer{$id}{$count}{PRIMER_LEFT_SEQUENCE} = $2;
    }
    elsif ( $line =~ /PRIMER_RIGHT_(\d*_)?SEQUENCE=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_SEQUENCE} = $2;
    }
    elsif ( $line =~ /PRIMER_PAIR_PENALTY(_\d*)?=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_PAIR_PENALTY} = $2;
    }
    elsif ( $line =~ /PRIMER_LEFT(_\d*)?=(\d+),(\d+)/ ) {
        $primer{$id}{$count}{PRIMER_LEFT_START} = $2;
        $primer{$id}{$count}{PRIMER_LEFT_LEN}   = $3;
    }
    elsif ( $line =~ /PRIMER_RIGHT(_\d*)?=(\d+),(\d+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_START} = $2;
        $primer{$id}{$count}{PRIMER_RIGHT_LEN}   = $3;
    }
    elsif ( $line =~ /PRIMER_LEFT(_\d*)?_TM=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_LEFT_TM} = $2;
    }
    elsif ( $line =~ /PRIMER_RIGHT(_\d*)?_TM=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_TM} = $2;
    }
    elsif ( $line =~ /PRIMER_LEFT(_\d*)?_GC_PERCENT=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_LEFT_GC} = $2;
    }
    elsif ( $line =~ /PRIMER_RIGHT(_\d*)?_GC_PERCENT=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_RIGHT_GC} = $2;
    }
    elsif ( $line =~ /PRIMER_PRODUCT_SIZE(_\d*)?=(.+)/ ) {
        $primer{$id}{$count}{PRIMER_PRODUCT_SIZE} = $2;
    }
    else {
        next;
    }
}

#open OUTFASTA , ">$file.primerProduct.fasta";
print
"ID\tSEQLength\tprimerSetNum\tprimerOrient\tproduct_size\tstart\tlen\ttm\tgc%\t
primerSeq\n";
foreach my $id ( keys %primer ) {
    my $seqLen   = $primer{$id}{1}{SEQ_LENGTH};
    my $sequence = $primer{$id}{1}{SEQUENCE};
    foreach my $count ( sort { $a <=> $b } keys %{ $primer{$id} } ) {
        next if $count == 0;
        my $leftSeq       = $primer{$id}{$count}{PRIMER_LEFT_SEQUENCE};
        my $rightSeq      = $primer{$id}{$count}{PRIMER_RIGHT_SEQUENCE};
        my $leftStart     = $primer{$id}{$count}{PRIMER_LEFT_START};
        my $leftLen       = $primer{$id}{$count}{PRIMER_LEFT_LEN};
        my $rightStart    = $primer{$id}{$count}{PRIMER_RIGHT_START};
        my $rightLen      = $primer{$id}{$count}{PRIMER_RIGHT_LEN};
        my $leftTM        = $primer{$id}{$count}{PRIMER_LEFT_TM};
        my $rightTM       = $primer{$id}{$count}{PRIMER_RIGHT_TM};
        my $leftGC        = $primer{$id}{$count}{PRIMER_LEFT_GC};
        my $rightGC       = $primer{$id}{$count}{PRIMER_RIGHT_GC};
        my $primerPairPen = $primer{$id}{$count}{PRIMER_PAIR_PENALTY};
        my $productSize   = $primer{$id}{$count}{PRIMER_PRODUCT_SIZE};

        my $product = substr( $sequence, $leftStart, $productSize );
        print
"$id\t$seqLen\t$count\tleft\t$productSize\t$leftStart\t$leftLen\t$leftTM\t$left
GC\t$leftSeq\n";
        print
"$id\t$seqLen\t$count\tright\t$productSize\t$rightStart\t$rightLen\t$rightTM\t$
rightGC\t$rightSeq\n";

#print OUTFASTA ">$id(primerSet:$count) productSize=$productSize leftTM=$leftTM
leftGC=$leftGC leftSeq=$leftSeq rightTM=$rightTM rightGC=$rightGC
rightSeq=$rightSeq\n$product\n";
    }
}
```

References

Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G., Korf, I., Lapp, H. et al. (2002) 'The Bioperl toolkit: Perl modules for the life sciences', *Genome Res* 12(10): 1611-8.

Korf, I., Yandell, M. and Bedell, J. (2003) *BLAST*, Sebastopol, CA: O'Reilly & Associates.