An Analysis of Salmonid RNA Sequences and Implications for Salmonid Evolution

by

Gordon David Brown
B.Sc., University of Victoria, 1990
M.Sc., University of Victoria, 1998

A Dissertation Submitted in Partial Fullfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
in the Department of Computer Science

© Gordon David Brown, 2008
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
An Analysis of Salmonid RNA Sequences and Implications for Salmonid Evolution

by

Gordon David Brown
B.Sc., University of Victoria, 1990
M.Sc., University of Victoria, 1998

Supervisory Committee

Dr. Valerie King, Supervisor
(Department of Computer Science)

Dr. Ben F. Koop, Co-supervisor
(Department of Biology)

Dr. Ulrike Stege, Departmental Member
(Department of Computer Science)

Dr. Chris Upton, Outside Member
(Department of Biochemistry and Microbiology)
**Supervisory Committee**

Dr. Valerie King, Supervisor  
(Department of Computer Science)

Dr. Ben F. Koop, Co-supervisor  
(Department of Biology)

Dr. Ulrike Stege, Departmental Member  
(Department of Computer Science)

Dr. Chris Upton, Outside Member  
(Department of Biochemistry and Microbiology)

**Abstract**

This work addresses two areas of computational biology: automation of sequence processing and an assessment of the evidence for a hypothesized salmonid genome based on an analysis of a set of expressed sequence tags.

Three problem areas in sequence processing are addressed in the first half of the work. Chapter 3 describes an accurate technique for trimming of vector, adapter and poly(A) sequence. Chapter 4 suggests methods for verifying the accuracy of assembled mRNA transcripts despite a large number of chimeras in the cDNA clone libraries. Chapter 5 is concerned with the problem of estimating the number of transcripts in a tissue or cDNA library, concluding that computational and statistical techniques are inadequate to estimate the quantity accurately.

The hypothesized salmonid genome duplication has been widely accepted since 1984. If it occurred, it should have left evidence in the form of many paralogous pairs of genes, all at approximately the same degree of sequence divergence. To assess
this question, several hundred thousand ESTs were assembled into transcripts, compared to each other to find homologs, and the evolutionary distances of the homologs represented as a histogram. Evidence of a single evolutionary event was not seen. The same procedure was applied to *Xenopus laevis*, which has a well-established recent genome duplication, and *Danio rerio*, which is known not to have had one. In those cases, the evidence for or against a genome duplication appeared exactly as predicted. The conclusion is that if the salmonid genome duplication occurred, some force altered its evolutionary development subsequently to mask the duplication, but also that a genome duplication is not necessary to explain the observed pattern of homolog distances.
# Table of Contents

Supervisory Committee ii

ABSTRACT iii

Table of Contents v

List of Tables viii

List of Figures ix

1 Introduction 1
  1.1 Sequence Processing ................................................................. 2
  1.2 Salmonid Evolution ..................................................................... 3
  1.3 Dissertation Overview ................................................................. 9

2 Sequence Processing 11
  2.1 Base Calling ............................................................................... 13
  2.2 Vector Masking .......................................................................... 14
  2.3 Adapter and Poly(A) Tail Detection and Masking ......................... 15
  2.4 Contamination Screening ............................................................. 15
  2.5 Low Quality and Low Complexity Filtering ................................. 15
  2.6 Transcript Assembly .................................................................... 18
  2.7 Contig Annotation ......................................................................... 20
  2.8 Screening Contigs for Trustworthiness ........................................ 22

3 Detecting and masking vector, adapters and poly(A) tails in ESTs 24
  3.1 Detecting adapters ....................................................................... 25
  3.2 Detecting Poly(A) Tails ................................................................. 29
    3.2.1 Polyadenylation Signals .......................................................... 40
  3.3 Discussion and future work ............................................................ 42
B Raw Data
# List of Tables

2.1 Trimer frequencies in salmonid mRNA .............................. 19

3.1 Detecting adapters ................................................. 27
3.2 Finding the end of the adapter .................................. 28
3.3 Nucleotide frequencies in poly(T) heads versus other RNA .... 37
3.4 Statistics on poly(T) head lengths ................................ 39
3.5 Adapters and poly(A) sequences found in reads ................. 44
3.6 Frequency of over-represented motifs near poly(A) tails ...... 45
3.7 Correlation of presence of TGTNTT with standard poly(A) signals .. 45

4.1 Criteria for chimera/non-chimera discrimination .................. 55

5.1 Estimates of transcriptome size .................................... 63

6.1 Input sequences for phylogeny ..................................... 68
6.2 Orthologs to Atlantic salmon transcripts .......................... 74
6.3 Paralogs in salmon, zebrafish and clawed frog .................... 81

7.1 Haploid Genome Sizes ............................................. 85
## List of Figures

1.1 Genome duplications in evolutionary context ........................................... 8
2.1 Low complexity DNA ...................................................................................... 17
3.1 An easily-recognized poly(A) tail ................................................................. 30
3.2 A difficult poly(A) tail ................................................................................. 31
3.3 A simple, not very effective algorithm for detecting poly(T) heads .......... 33
3.4 The global alignment algorithm ................................................................... 34
3.5 The local alignment algorithm ................................................................. 35
3.6 The prefix alignment algorithm ................................................................. 36
4.1 A string $S$ and overlapping substrings $S_i$ of $S$ ..................................... 49
4.2 Cartoon of real and chimeric substrings of $S$ ......................................... 50
4.3 Chimeric fragment joins two mRNAs ......................................................... 52
6.1 UTR versus coding distances in Orthologs .............................................. 70
6.2 UTR versus coding distances in Paralogs .................................................. 71
6.3 Apparent Orthologs ....................................................................................... 73
6.4 Atlantic Salmon Orthologs in Other Species ........................................... 75
6.5 The expected pattern of a genome undergoing re-diploidization follow- ing a genome duplication ................................................................. 77
6.6 Overcounting paralogs .................................................................................. 78
6.7 Histogram of paralog distances considering only the best match for each transcript ................................................................................................. 79
6.8 Non-overlapping contigs .............................................................................. 80
6.9 Atlantic salmon paralogs ............................................................................. 81
7.1 Paralogs and Orthologs .................................................................................. 87
7.2 Number of salmon orthologs for zebrafish sequences ................................. 89
A.1 The central dogma of molecular biology .................................................. 110
A.2  A mature messenger RNA  .................................................. 111
A.3  A sketch of a cDNA clone  .............................................. 114
Chapter 1

Introduction

Note: Appendix A is an introduction to basic concepts and terminology in molecular biology and evolution that are assumed in this work.

A genome duplication is a mutation in which the entire genome of an organism is copied, leaving the organism with twice as much DNA as it previously had. Salmonids are widely thought to have undergone a genome duplication after they diverged from closely related taxa such as smelt and pike. Recently, several hundred thousand expressed sequence tags (ESTs) from salmonids have become available (Rexroad III et al. 2003, Rise et al. 2004).

This work covers two main areas. The first is the automation of EST processing, as a prerequisite for the analysis of the very large sets of genetic data which are now available. The second is an analysis of the salmonid ESTs to assess their support for the salmonid duplication, and the evolution of the genome since then. In addition, as a side benefit of obtaining a large sample of salmonid messenger RNA (mRNA) sequences, we attempt to estimate the number of transcripts in the complete Atlantic salmon and rainbow trout genomes.
1.1 Sequence Processing

With the ongoing increase in the efficiency of DNA sequencing, automated downstream processing of generated sequences becomes essential. With current sequencing machines capable of running in excess of 3800 samples per day,\(^1\) screening and trimming of generated sequences would be prohibitive if done by hand.

Automated processing of DNA sequencing output is well established. The basic task of assembling a set of overlapping reads into a single contiguous sequence, or contig, is also well established. The Phred/Phrap pair of programs (Ewing et al. 1998, Ewing & Green 1998, Green n.d.), together with the associated graphical interface Consed (Gordon et al. 1998), provide the capability to assemble a DNA sequence from raw chromatograms. Several other packages serve the same purpose, with minor variations; CAP3 (Huang & Madan 1999), Euler (Pevzner et al. 2001b) and SeqMan (DNASTAR 2006) are popular tools that solve this problem.

The problem addressed in this work is somewhat different: rather than assemble a single sequence from a set of reads, the task is to assemble many mRNA sequences from a set of EST reads. As part of the Genomic Research on Atlantic Salmon Project (GRASP), about 430,000 Atlantic salmon and 125,000 Rainbow trout ESTs were collected and analyzed. Most were sequenced as part of GRASP (Rise et al. 2004); others were acquired from other salmonid researchers (Rexroad III et al. 2003, Hoyheim n.d.). The task is to assemble them into accurate, correctly trimmed mRNA transcripts.

Some parts of that task are similar to the assembly of a single contig: detecting overlaps between reads efficiently, discriminating between mismatches caused by sequencing errors and multiple occurrences of near-identical repeated motifs, and the

---

\(^1\)Applied Biosystems reports that their 3730xl model DNA sequencer can sequence 3840 samples per day (Applied Biosystems 2006).
construction of consensus sequences from possibly-conflicting reads of varying quality. Some problems are unique to the assembly of mRNAs: discriminating between chimeric reads and splice variants, identifying the ends of transcripts, and trimming poly(A) tails accurately, for example.

\section*{1.2 Salmonid Evolution}

We first introduce a number of terms to describe evolutionary events related to gene and genome duplications.

\textbf{diploid} A genome is diploid if it has two copies of each chromosome. Most animals are diploid.

\textbf{disome} A disome is a pair of chromosomes which are evolving together, i.e. the usual pair of copies that diploid organisms have. The phrase “evolving together” indicates that if a mutation occurs in one, it will likely be copied to the other sooner or later, so that the copies remain similar over millions of years. Genes which occur on disomic chromosomes undergo \textit{disomic} inheritance.

\textbf{polyploid} A genome is polyploid if it incorporates multiple copies of the whole genome of an ancestor species, i.e. if its genome has been duplicated at least once. In this case it has more than two copies of each chromosome.

\textbf{tetraploid} A genome is tetraploid if it has 4 copies of each chromosome, i.e. has had exactly one genome duplication from a diploid ancestor.

\textbf{tetrasome} A tetrasome is a group of 4 chromosomes which are evolving together. During meiosis, tetrasomes appear as visible structures called \textit{quadrivalents} in the cell; such structures are used as evidence of the existence of tetrasomes. Genes occurring on tetrasomic chromosomes undergo tetrasomic inheritance.
**pseudo-tetraploidy** A genome is pseudo-tetraploid if it has had a genome duplication, but the related chromosomes are diverging; the end result (after millions of years) is an organism with roughly twice as much DNA as its ancestor, but diploid. The process of returning to a diploid state is called “re-diploidization”. Immediately following a duplication, genes may undergo tetrasomic inheritance if some chromosomes remain in tetrasomes.

**paralog** Genes are paralogs if they arose from a single ancestor gene via a gene-sized or larger duplication within a single species. In other words, there are two or more copies in the species’ genome, but in some ancestor organism there was only one copy.

**ortholog** Genes are orthologs if they arose from the same ancestor gene via speciation, that is, their most recent common ancestor gene is a single gene in the most recent common ancestral species from which they arose.

**homolog** Genes are homologs if they had a common ancestor gene, whether they arose due to speciation or gene duplication or multiple events of either type.

A series of salmonid genome duplications was first suggested by Svardson (1945); he described several duplications within the salmonids, based on different numbers of chromosomes in various species. His proposal was shown to be incorrect by Rees (1964), but a later proposal by Ohno et al. (1968) of a single genome duplication prior to the ancestral salmonid’s speciation into many taxa remains the accepted theory. Allendorf & Thorgaard (1984) summarized the evidence, concluding that the duplication is well-supported; the issue has been viewed as largely resolved since then, though the date at which the event occurred was not known except extremely roughly.² Recently, several hundred thousand expressed sequence tags (ESTs) from

---

²Allendorf and Thorgaard reviewed evidence which suggest anything from 25 to 100 million years ago, but the upper bound was based on weak evidence, so the most that could be said with confidence
salmonids have become available (Rexroad III et al. 2003, Rise et al. 2004). In this work, we analyze these ESTs for evidence of the genome duplication, an indication of the approximate age of the event, and for information about the course of subsequent changes in the genome, that is, the process of re-diploidization.

The history of the salmonid genome is of interest as part of widespread research into the evolution of genes and genomes. Gene duplications were first observed as far back as 1936, when Bridges identified a duplicated region in the fruit fly *Drosophila melanogaster* genome using chromosome banding patterns (Bridges 1936). Evidence for gene duplication as an important element of evolution accumulated over the next decades, to the point that Ohno wrote in 1968 that “gene duplication now emerges as the prime factor of evolution” (Ohno et al. 1968, page 169). With the publication of his book *Evolution by Gene Duplication* (Ohno 1970), gene duplication became firmly entrenched in the mainstream of research into the mechanisms of evolution.

Gene duplication is a fundamental component of evolution because it is the primary mechanism for the creation of new genes (Ohno et al. 1968). The probability of an arbitrary sequence of DNA undergoing random changes and becoming a useful gene is extremely low. It is much more likely that a gene is copied, and that one of the two copies takes on a new function. In general, there are three possible fates for a newly copied gene:

1. Additional mutations may prevent one copy from being transcribed or translated, making it a non-functional *pseudogene*;

2. One copy may take on a new function (*neofunctionalization*);

3. The two copies may become specialized for different aspects of the gene’s original function (*subfunctionalization*).

is “probably at least 25 million years ago, and after the divergence from neighboring diploid taxa.”
Since the majority of mutations are deleterious (Graur & Li 2000), the first possibility is the most likely (Lynch & Conery 2000). However, even the relatively unlikely other possibilities are much more likely than a series of random mutations creating a functional and useful gene from non-coding DNA. While individual duplications may play a significant role in evolution, whole genome duplications create a large pool of duplicated pairs, allowing for significant increases in specialization of genes, and organism complexity.

Ohno proposed that two genome duplications have occurred in the process of evolution from primitive chordates to mammals, based on the quantity of DNA in the nucleus of the cells of various taxa, the number of chromosome arms in the genomes of those taxa, and the existence of many gene families (Ohno et al. 1968). Debate has ensued regarding the likelihood of two duplications (Escriva et al. 2002, Sidow 1996), just one (McLysaght et al. 2002, Guigo et al. 1996), or none (Hughes & Friedman 2004, Friedman & Hughes 2001), as well as the probable course of events post-duplication. The cited works are a tiny sample of an extensive literature on this subject. We do not propose to address this controversy; we mention it to make it clear that establishing the occurrence or non-occurrence of a hypothesized genome duplication is, or can be, controversial, and that the consequences of a genome duplication (the process of re-diploidization) are still poorly understood.

The occurrence of genome duplications in animals has been clearly demonstrated in some cases. For example, the evidence for a genome duplication early in the teleost lineage is strong (Jaillon et al. 2004). Examining another region of the evolutionary tree, Bisbee et al. (1977) suggested that Xenopus laevis, the African clawed frog, underwent a relatively recent genome duplication; Tymowska (1991) and Hughes & Hughes (1993) provide extensive evidence for it, as well as for several other polyploid genomes in the Xenopus genus. A genome duplication also occurred in yeast (Saccharomyces cerevisiae) approximately 100 million years ago (Wolfe & Shields 1997, Kellis...
et al. 2004). Figure 1.1 places several of the duplications mentioned here in their evolutional context.

There are four main lines of argument supporting an early salmonid genome duplication, as described by Allendorf & Thorgaard (1984).

1. Salmonids have approximately twice the DNA content per cell as closely related fish.

2. The number of chromosome arms is approximately 100 (Allendorf & Thorgaard 1984, Hartley 1987), as compared to 48 for many fish species (Gold et al. 1980, Sola et al. 1981).

3. Quadrivalents have been observed during meiosis.

4. Many paralogous pairs of genes have been observed in salmonids.

Most researchers have accepted these arguments as essentially conclusive.

Our original intent was to examine the newly-available set of ESTs with the goal of assembling a set of mRNAs, then detecting Atlantic salmon–Rainbow trout orthologs, plus salmon–salmon and trout–trout paralogs. Then we could measure the evolutionary distance of the orthologous pairs and the paralogous pairs, with the objective of dating both the speciation into Rainbow trout and Atlantic salmon, and the genome duplication. We expected to find that

1. the orthologous pairs had all diverged by roughly the same amount, corresponding with the speciation, and that

2. the paralogous pairs had likewise diverged by roughly the same amount, corresponding with the genome duplication.

By analyzing the 3’ UTR region of the transcripts, which is subject to less purifying selection than coding sequence, we expected the pattern of evolutionary distances to be relatively clear. In the case of orthologs, the result was exactly as expected: a
Figure 1.1: An evolutionary tree showing several real or hypothesized genome duplications in their historical context. Circles 1 and 2 are Ohno’s hypothesized duplications, still considered controversial. Circles 3 and 5 are the ancient teleost duplication and the recent *Xenopus laevis* duplication, both relatively well-confirmed. Circle 4 is the salmonid duplication proposed by Ohno. This tree was constructed using information from NCBI’s taxonomy database (National Center for Biotechnology Information 2007), except for the relative positioning of *Esox* and *Osmerus*, which is based on (Ishiguro et al. 2003). Common names of taxa are in normal type; scientific names are in italics. Note that branch lengths are not to evolutionary scale.
histogram of ortholog distances showed a well-defined peak at a distance of approximately 0.06 substitutions per nucleotide (see Chapter 6 for details). In the case of paralogs, the result was not as expected. Instead of a relatively well-defined peak in a histogram of paralog distances, we obtained a wide range of distances with no discernible peak which could correspond to a genome duplication, but rather a gradual decline from extremely similar to dissimilar. In addition, many paralogs were more closely related than orthologs, which does not support the genome duplication hypothesis well.

There is no doubt that there has been extensive duplication in the salmonid genome. The question is how it occurred. Regarding the four arguments summarized by Allendorf and Thorgaard, the second point specifically supports a single genome duplication, as opposed to an alternate hypothesis of many chromosome-sized or smaller duplications. The first supports a single duplication, but only weakly, because the “approximately” in “approximately twice the DNA content” is quite approximate. The third and fourth arguments, and to some extent the first, support the alternate hypothesis at least as well as the accepted one. Considering our discovery that the ages of the paralogs varies widely, we shall consider this alternate hypothesis in some detail.

1.3 Dissertation Overview

Chapter 2 describes the overall sequence processing pipeline. Chapter 3 provides details on the trimming process. Chapter 4 covers issues related to assembly in the presence of chimeras, in particular why it is a bigger problem for the assembly of mRNAs than it is for single contigs.

Chapter 5 addresses the problem of determining the transcriptome size from a sample of ESTs.
Chapters 6 and 7 discuss the detection of orthologs and paralogs, the measurement of evolutionary distance between them, and their consequences for the hypothesis of a salmonid genome duplication versus multiple smaller duplications.

Chapter 8 summarizes our results, offers some conclusions, and suggests future avenues of research.
Chapter 2

Sequence Processing

The path from raw chromatograms to useful, annotated mRNAs is long and tedious, including filtering out contamination, trimming of vector from reads, assembly and annotation. With the large quantity of data generated by modern DNA sequencing machinery, automation is essential. Since EST processing differs in several respects from the assembly of single sequences, any EST processing pipeline must be specialized for the task.

Two well-known EST processing software packages are “TranscriptAssembler” from Striking Development (formerly Paracel), and “stackPACK” from the South African National Bioinformatics Institute (Miller et al. 1999, Burke et al. 1999). TranscriptAssembler’s major flaw is easily stated: it is no longer available. stackPACK has two flaws:

1. it is available only for out-of-date operating systems, and is difficult or impossible to install on newer ones\(^1\), and

2. because it uses the \(d^2\) algorithm (Hide et al. 1994) for clustering, it is very

\(^1\)An experienced system administrator in the UVic Department of Biology was unable to get it fully working on a newer version of Red Hat Linux than version 7.3, on which it was developed.
sensitive to the presence of chimeras in the input data.

The second point is worthy of elaboration. stackPACK uses a two-stage clustering algorithm. In the first stage, $d^2$ is used to find sequences which show any significant overlap at all. Sequences which overlap are placed into the same cluster, along with any other sequences which have been found to overlap with either.\(^2\) Once initial clustering is complete, the Phrap sequence assembler (Green n.d.) is used to make one or more contigs from each cluster. In the presence of chimeras, ESTs which do not truly overlap will be placed in the same cluster, if they overlap with regions of a chimera which came from different original transcripts. In the presence of a substantial number of chimeras, in the extreme case the majority of transcripts could end up in the same cluster, making the initial clustering step relatively ineffective. This effect was observed in the salmonid EST data analyzed in this work, making stackPACK relatively unhelpful.

A custom solution was developed, using a combination of existing tools and new software. Well-known tools were used for base calling, assembly, vector screening and annotation; new tools were developed for accurate trimming of extraneous sequence from reads, low complexity screening, and filtering assembled sequences for reliability. This chapter describes the process in general terms; details of the new components are described in subsequent chapters.

The sequence processing pipeline begins with a set of chromatograms from the sequencing of clones from salmonid cDNA libraries. The output is a set of annotated DNA sequences representing mRNAs or fragments of mRNAs. Annotation of a sequence includes some or all of the orientation of the sequence ($3' \rightarrow 5'$ or $5' \rightarrow 3'$), the position of start and stop codons, the name of the protein it encodes (or the

\(^2\)This approach is reminiscent of R. E. Tarjan’s disjoint set union/find data structure as described in (Tarjan 1975, Tarjan 1983), though $d^2$ does not appear to use Tarjan’s algorithm.
one most similar to it), and whether a poly(A) tail was detected. Given a set of chromatograms, the processing steps are:

1. base calling,
2. masking of vector,
3. detection and masking of adapters and poly(A) tails,
4. removal of contamination,
5. detection and discarding of low quality sequences,
6. assembly of reads into contigs,
7. contig annotation, including identification of contig orientation and location of coding and UTR regions (if possible), and
8. analysis of contigs for trustworthiness.

These steps are described in the following sections.

2.1 Base Calling

Base calling is the translation of a chromatogram into a DNA sequence, as well as a quality sequence which records, for each nucleotide in the sequence, the likelihood that the base calling software gave the right answer. We use the widely-used Phred program (Ewing et al. 1998, Ewing & Green 1998) for base calling, using default parameters. For each nucleotide, Phred generates a quality value $q = -10 \log_{10} P_e$, where $P_e$ is the probability that the base call is incorrect. A quality value of 13 corresponds with a 5% probability of error; a value of 20 corresponds with a 1% probability of error. These quality values are used at several points in the processing
pipeline, to resolve conflicts between reads (in the case of sequencing errors) or to confirm that apparently different nucleotides in multiple sequences really are different.

2.2 Vector Masking

Vector masking is the process of replacing any vector DNA in a read with Xs, so that it won’t interfere with later processing. The Phrap software package (Green n.d.) includes a program called cross_match which can, given a DNA sequence and a set of vector sequences, find all regions of the DNA sequence which match some part of the vector sequence, and replace them with X’s. The default parameters for cross_match of minmatch=14 and minscore=30 are tuned for fast but approximate masking, suitable for input to the phrap sequence assembly program; they find regions which include at least one exact 14-nucleotide match, and score at least 30 overall using a score model of +1 for a match, -2 for a mismatch, -2 for an indel. (See Section A.3 for a summary of alignment and alignment scoring.) Phrap is designed to disregard short non-matching sequences at the beginning and end of a read, so the vector-screening step can afford to miss short fragments of vector. We use more sensitive parameters, since we are interested in detecting even short vector sequences: minmatch=8 and minscore=15 find vector sequences scoring as little as 15, with at least one exact match of 8 nucleotides, using the same score model. Some reads are entirely vector, because the insert was not correctly incorporated into the vector. In this case the vector masking process replaces the entire read with X’s; such reads are discarded.

Since the first few and last few nucleotides in a read are typically very low quality, masking software usually does not recognize it as vector. For ease of subsequent processing, low quality leading and trailing (before the leading vector, and after the trailing vector, if any) are replaced with Xs, so that every retained read has exactly one non-vector region.
2.3 Adapter and Poly(A) Tail Detection and Masking

In addition to the vector and the actual sequence of the EST, reads usually include adapter sequences, and possibly a poly(A) tail. These extraneous sequences must be identified and masked. Chapter 3 describes the method in detail.

Short inserts contribute relatively little information, so we discard sequences which (after masking of vector, adapters and poly(A) tails) are less than 100 nucleotides long.

2.4 Contamination Screening

Occasionally, an insert is sequenced that is not from the species of interest. Typically, it is *Escherichia coli*, a bacterium used in sequencing. Detection of contamination is straightforward, if the full DNA sequence of the contaminating organism is known, as it is in the case of *E. coli*. We use NCBI’s BLAST (Altschul et al. 1990, Altschul et al. 1997) to identify sequences which are highly similar to contaminant species; such sequences are marked as contamination and excluded from further processing.

2.5 Low Quality and Low Complexity Filtering

Some chromatograms are not useful, because either the read is not of high quality overall, or the sequence is all or mostly low complexity sequence. Low complexity sequences are those consisting mainly of short repeating patterns (typically 2 or 3 nucleotides long). These reads can be the result of problems in the sequencing process, or may be real repetitive DNA.

A low quality read can result from any number of problems with the physical
process of sequencing. If, after removing low quality leading and trailing regions of a read, the read includes at least 25% nucleotides with quality values less than 13 (more than 5% probability of error), the read is discarded. In addition, if this remaining region is less than 100 nucleotides long, the read is discarded.

The intent of this low complexity screening step is to filter out sequences which are low complexity due to sequencing problems, that is, to identify and reject entire reads which are artefacts, rather than the more common problem of masking low complexity regions of real DNA. Unfortunately it is not always possible to distinguish them.

One problem in sequencing is called sequencer stutter: if the true sequence is, for example, AGACATAC, the output of the sequencer might include several copies of each nucleotide:

```
AAAAAAAAGGGGAAAAAAAACCCCCCACAATT TT T T TAAAACCCCCCCCC
```
or something similar. This sort of chromatogram may have adequate quality values according to the base calling software, but it is not real. Other sequencing problems produce long repeating di- or tri-nucleotide repeats such as “GAGAGAGAGA...”. Figure 2.1 shows an example of DNA which has adequate quality values, but low complexity. If these repeating patterns are real DNA, they might be of interest; if not, they should be discarded. Unfortunately there does not appear to be a good way to tell the difference, so they are all discarded.

BLAST includes a low complexity filter called DUST (Tatusov & Lipman n.d.). We do not use it for two reasons. First, at the time this work was done, DUST was not easily separable from the BLAST program (the author tried, but gave up after expending considerable effort). Second, DUST is designed to detect and mask real DNA which is low complexity, whereas we wish to filter incorrect outputs from DNA sequencers. Though the problems are similar, we chose to abandon efforts to apply DUST and use our own approach, described below.
Figure 2.1: Low complexity DNA. The leading X’s are masked vector sequence.

We detect low complexity sequences by counting the frequency of trimers in each read, and comparing the frequencies to typical mRNA. A table of typical trimer frequencies was initially constructed by counting trimers in several thousand salmonid mRNA sequences (results shown in Table 2.1). The statistic chosen is the sum of squares of the difference between the typical frequencies and the frequencies in a single sequence. Given the set $T$ of all trimers, the typical frequencies $B_t$ for trimers $t \in T$, and the trimer frequencies $S_t$ in a given sequence $S$, we compute

$$S = \sum_{t \in T} |B_t - S_t|^2 \quad (2.1)$$

To decide at what threshold a sequence should be considered low complexity, the trimer frequencies of 1500 low complexity and 24,000 normal sequences were computed. Of the 1500 low complexity sequences, all but 159 had scores greater than

---

3Early experiments using dimers rather than trimers were not successful. The differences between dimer frequencies in some low complexity sequences and normal mRNA were not sufficiently clear to allow for good filtering.

4Sequences were classified as normal or low complexity by visual inspection. The overwhelming majority of sequences can be classified at a glance, so visual inspection is not as implausible as it might seem; only a few sequences required careful examination.
0.02. Of the 24,000 normal sequences, all but 195 had scores less than 0.02. Therefore a threshold value of 0.02 was chosen (no other threshold had a lower total of false positives plus false negatives). Sequences with scores less than the threshold are retained, and the others discarded.

### 2.6 Transcript Assembly

Sequence assembly is the process of discovering the nucleotide sequence of a DNA molecule from a collection of overlapping reads. Since technological limitations currently prevent us from directly sequencing more than a few hundred nucleotides of a DNA molecule at one time, the standard approach to sequencing long molecules is to sequence short overlapping fragments, then put the fragments together. Given sufficiently many reads from randomly chosen positions in the original sequence, all or most of the original sequence can be reconstructed. The reconstructed sequences, found by successively overlapping matching reads and assembled sub-sequences, are called *contigs* (from *contiguous sequence*).

In the case of cDNA libraries, the problem is to assemble many mRNA transcripts from a set of reads, without prior information about which reads are part of the same transcript, or even how many transcripts there are. One complication which occurs in both regular sequence assembly and assembly of a set of mRNA transcripts is the presence in cDNA libraries of clones which are composed of fragments from two or more different mRNA transcripts. During the library construction process, it is possible for unrelated fragments of RNA to join end-to-end, as if they were one longer fragment. These clones are called *chimeric* clones or just *chimeras*. The assembly process, including the complications which arise as a result of the presence of chimeras, is described in detail in Chapter 4.
<table>
<thead>
<tr>
<th>Trimer</th>
<th>Freq.</th>
<th>Trimer</th>
<th>Freq.</th>
<th>Trimer</th>
<th>Freq.</th>
<th>Trimer</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>0.03711</td>
<td>AAC</td>
<td>0.01903</td>
<td>AAG</td>
<td>0.01714</td>
<td>AAT</td>
<td>0.02392</td>
</tr>
<tr>
<td>ACA</td>
<td>0.02981</td>
<td>ACC</td>
<td>0.01223</td>
<td>ACG</td>
<td>0.00545</td>
<td>ACT</td>
<td>0.01575</td>
</tr>
<tr>
<td>AGA</td>
<td>0.01885</td>
<td>AGC</td>
<td>0.01361</td>
<td>AGG</td>
<td>0.01534</td>
<td>AGT</td>
<td>0.01703</td>
</tr>
<tr>
<td>ATA</td>
<td>0.02133</td>
<td>ATC</td>
<td>0.01442</td>
<td>ATG</td>
<td>0.01979</td>
<td>ATT</td>
<td>0.02242</td>
</tr>
<tr>
<td>CAA</td>
<td>0.02252</td>
<td>CAC</td>
<td>0.01746</td>
<td>CAG</td>
<td>0.02195</td>
<td>CAT</td>
<td>0.02219</td>
</tr>
<tr>
<td>CCA</td>
<td>0.01770</td>
<td>CCC</td>
<td>0.01141</td>
<td>CCG</td>
<td>0.00425</td>
<td>CCT</td>
<td>0.01486</td>
</tr>
<tr>
<td>CGA</td>
<td>0.00424</td>
<td>CGC</td>
<td>0.00368</td>
<td>CGG</td>
<td>0.00437</td>
<td>CGT</td>
<td>0.00564</td>
</tr>
<tr>
<td>CTA</td>
<td>0.01111</td>
<td>CTC</td>
<td>0.01500</td>
<td>CTG</td>
<td>0.01839</td>
<td>CTT</td>
<td>0.01760</td>
</tr>
<tr>
<td>GAA</td>
<td>0.01748</td>
<td>GAC</td>
<td>0.01214</td>
<td>GAG</td>
<td>0.01370</td>
<td>GAT</td>
<td>0.01317</td>
</tr>
<tr>
<td>GCA</td>
<td>0.01503</td>
<td>GCC</td>
<td>0.00960</td>
<td>GCG</td>
<td>0.00399</td>
<td>GCT</td>
<td>0.01239</td>
</tr>
<tr>
<td>GGA</td>
<td>0.01375</td>
<td>GGC</td>
<td>0.01031</td>
<td>GGG</td>
<td>0.01241</td>
<td>GGT</td>
<td>0.01272</td>
</tr>
<tr>
<td>GTA</td>
<td>0.01368</td>
<td>GTC</td>
<td>0.01272</td>
<td>GTG</td>
<td>0.01468</td>
<td>GTT</td>
<td>0.01595</td>
</tr>
<tr>
<td>TAA</td>
<td>0.01983</td>
<td>TAC</td>
<td>0.01453</td>
<td>TAG</td>
<td>0.01197</td>
<td>TAT</td>
<td>0.01863</td>
</tr>
<tr>
<td>TCA</td>
<td>0.02155</td>
<td>TCC</td>
<td>0.01497</td>
<td>TCG</td>
<td>0.00426</td>
<td>TCT</td>
<td>0.01920</td>
</tr>
<tr>
<td>TGA</td>
<td>0.01954</td>
<td>TGC</td>
<td>0.01344</td>
<td>TGG</td>
<td>0.01699</td>
<td>TGT</td>
<td>0.02162</td>
</tr>
<tr>
<td>TTA</td>
<td>0.01891</td>
<td>TTC</td>
<td>0.01799</td>
<td>TTG</td>
<td>0.01893</td>
<td>TTT</td>
<td>0.02769</td>
</tr>
</tbody>
</table>

Table 2.1: Trimer frequencies in salmonid mRNA
2.7 Contig Annotation

This section describes the annotation of assembled mRNA sequences. Annotation serves two main purposes: it provides information to biologists about the identity and completeness of the mRNA sequences being produced, and it identifies the coding and untranslated regions of the sequences. This latter information is important for making meaningful evolutionary comparisons, since rates of evolution differ between coding and non-coding DNA (and between UTR and intergenic DNA, though that is not at issue here since all our sequences are from transcribed sequences, i.e. there is no intergenic DNA in the data set). Chapter 7 includes some results regarding the relative rates of evolution in coding and 3′ UTR regions. It concludes that UTR sequences provide more reliable results, as expected.

The basic annotation tool is BLAST, the Basic Local Alignment Search Tool (Altschul et al. 1990, Altschul et al. 1997). BLAST compares a sequence to a database of sequences, returning matches which exceed a specified statistical significance, based on the score model and the size of the database. By finding high-quality matches with known protein sequences, it is possible to determine the orientation of an RNA sequence, the probable start and end of the coding sequence, and a putative identification of the gene corresponding with this RNA. Protein databases can be searched for matches to nucleotide sequences by first constructing all possible translations of the nucleotide sequence into amino acids, then searching for matches as usual. BLASTX translates the query in all six reading frames, so the nucleotide-sequence-to-protein-database comparison is straightforward. BLAST generates alignment scores, bit scores and e-values. Alignment scores are as described in Appendix A. A bit score is an alignment score which has been normalized to a standard score model. The e-value of a match is the expected number of matches of equal or higher score one could expect to find by chance when searching a database of the size of the one
used, using a random query string of the same length as the input query sequence. This implies that smaller e-values are more significant.

Three public databases of protein sequences are used for annotation: the NCBI maintains a non-redundant protein sequence database, commonly referred to as “NR” (Benson et al. 2006), which contains essentially all publicly available protein sequences; the Gene Ontology Consortium and the UniProt Consortium maintain smaller but more highly curated collections of protein sequences (Ashburner et al. 2000, Gene Ontology Consortium 2001, Apweiler et al. 2004, Bairoch et al. 2005), referred to as the GO and UniProt databases. Using BLAST, it is straightforward to find protein sequences in these databases that are similar to the contigs resulting from our sequence assembly process, if such proteins are in the database. This general idea of using alignment against known proteins to annotate cDNA sequences, which we have used for several years, and which has been used by many others, was recently described by Min et al. (2005). I provide here the details of our method.

The first step is to find the best-quality matches in the protein database for each contig. We use BLAST to compare each sequence with the NR, GO and UniProt databases, keeping the three highest-scoring matches from each database. A match with an e-value of at most $10^{-15}$ is considered useful; anything with a higher e-value is discarded.\(^5\) The orientation of the sequence is determined by the reading frame of the translation which matched any given protein sequence. If the orientations deduced by two hits conflict (that is, one matches in a positive frame and the other in a negative frame), the orientation is considered unknown.

If the orientation is determined as above, and the matched region of the protein

---

\(^5\) The threshold of $10^{-15}$ represents a compromise between the biologists on the project and the author. The biologists wanted a threshold of $10^{-5}$ so that more sequences would be annotated; the author found that too many spurious matches were being taken as real, so he argued for a more stringent threshold on the grounds that no annotation is better than an incorrect annotation.
sequence includes the end of the sequence, and there is a stop codon in the correct reading frame within 10 nucleotides of the end of the match, it is assumed to be the true stop codon. At this point the coding sequence and 3′ UTR have been identified. If a start codon is found within 10 nucleotides of the beginning of the alignment, and the alignment includes the beginning of the protein sequence, it is considered to be the true start codon.

Further annotations step are to determine whether the contig is, or may be, chimeric, and whether a poly(A) tail was found, indicating that the contig extends to the end of the 3′ UTR. These steps are described in chapters 3 and 4.

### 2.8 Screening Contigs for Trustworthiness

Chimeric reads cannot in general be identified without comparing them to other sequences. One manual approach is to compare the sequence to public databases of protein sequences: if a read matches different proteins in different regions, it may be chimeric. Unfortunately, it is difficult to automate this approach, since it may be difficult to determine whether two matches really are to different proteins, or to conserved domains which occur in many proteins, or to variants of the same protein, etc.

Chapter 3 addresses this problem by considering two sources of information: the structure of the contig itself, and matches with other assembled contigs. If the contig incorporates multiple reads from multiple cDNA libraries at all points along its length, it is unlikely to be chimeric; if it matches a contig from another organism along most

---

6BLAST uses local alignment for matching the sequences. If two sequences are reasonably similar, but differ by a few amino acids at the end of the sequence, the alignment will not include them. For this reason we allow the stop codon to be a short distance downstream of the end of the alignment.

7There is one exception, briefly mentioned in Chapter 3: if poly(A) tail detection discovers a poly(A) tail on both ends of a read, it may be assumed to be chimeric.
or all of its length, it is likewise unlikely to be chimeric. In the absence of such confirmation, it must be assumed to be unreliable.
In addition to the actual sequence of the EST, sequencing reads usually include fragments of vector and adapter sequence, and possibly a poly(A) tail. These extraneous sequences, which are not part of the original gene, must be identified and removed. Removal is done by masking: replacing the sequence to be removed by X’s, which are recognized by most computational biology software tools as indicating sequence which should be ignored.¹

¹The alternative to masking is trimming: removing unwanted nucleotides from the sequence altogether. We mask rather than trim because masking does not change the length of the sequence, so the corresponding quality file does not need to be adjusted to match the trimmed sequence. If we trimmed sequences, the quality files would need the corresponding quality values removed as well, to keep the quality values aligned with the right nucleotides.
to that consensus sequence, since there will in general be other reads that override the extraneous fragments. Assembly software is designed to disregard non-matching nucleotides at the ends of reads. In the case of ESTs, though, each read is, or might be, important, since only one or two instances of rare transcripts may be found in shotgun sequencing. We need to maximize the amount of accurate information extracted from each read by removing nucleotides which are not part of the original gene. Failing to remove vector, adapters and poly(A) sequences can cause

1. misleading sequence annotations due to irrelevant matches with vector and adapter sequences in the public databases,

2. errors in assembly due to incorrect joins of reads based on matching garbage sequence in unrelated reads, and

3. errors in phylogenetic analyses.

Detecting and masking of vector is a well-established process, as mentioned in the previous chapter. The crossmatch program from the Phrap package (Green n.d.) was used in this work, with parameters “minmatch=8 minscore=15” to increase sensitivity to short fragments of vector sequence. Adapter and poly(A) sequence detection are less well studied; our techniques are described in the next two sections.

### 3.1 Detecting adapters

Many adapters are 6–8 nucleotides long (based on examination of sequences in GenBank as well as those sequenced as part of the GRASP project), too short to produce misleading BLAST hits\(^2\), or distort phylogenetic analyses significantly. However, an 8-nucleotide sequence will not be found in nucleotide or protein blast hits, because an exact match of 11 nucleotides is required for a nucleotide search, and 3 amino acids (9 nucleotides) for a translated protein search. (NCBI Blast Home Page n.d.)
some are 20–21 nucleotides long, and can cause problems. Masking them is important. Even for the shorter ones, accurate masking makes detection of ambiguous poly(A) sequences easier, because there is less extraneous sequence to confound the poly(A) detection mechanism.

Since the analyses in this work incorporate sequences collected from many research groups, an additional problem arises: the adapters may not be known. Though the people who constructed each cDNA library presumably know the adapter sequences, it is often not possible to find out who they were, to contact them if they are known, or (in the author’s experience) to extract that information from them, if they can be contacted. A first step, then, in trimming the adapters is determining what they are.

One simple approach to adapter detection is looking for over-represented motifs immediately following the leading vector, and immediately preceding the trailing vector (if any). Referring to Figure A.3, the typical read will begin with some vector sequence and the adapter, prior to the beginning of the insert. If all reads were perfect quality, the identification of adapters would be trivial: mask the vector, then the next few nucleotides should be the same in all sequences. Those few identical nucleotides are the adapter. The first non-identical position is the beginning of the insert. Of course, reads are not perfect. The adapter could be concealed by a variety of imperfections. For example, there may not be enough vector to be recognized, so it is not clear where the adapter should start. Alternatively, the adapter may be in the leading or trailing very low quality region at the beginning or end of the read, hence unrecognizable. Yet another possibility is that the adapter was corrupted in the process of sequencing. Notwithstanding those possibilities, assuming that sequencing was done in such a way as to include the adapter in the read, the true adapter should occur much more frequently immediately after the leading vector and before the trailing vector than any other sequence. Given a minimum adapter length of, say, 5 nucleotides, and a maximum of 22 (based on the adapters used in the GRASP
Table 3.1: Detecting adapters by counting over-represented prefixes and suffixes in 5451 sequences from a single cDNA library. The observed frequency of tail sequences is expected to be lower than that for heads, because many inserts are too long to be sequenced in one read, hence the tail adapter is not present in the read.

<table>
<thead>
<tr>
<th>Position</th>
<th>Length</th>
<th>Sequence</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>head</td>
<td>5</td>
<td>GCACG</td>
<td>0.31</td>
</tr>
<tr>
<td>tail</td>
<td>5</td>
<td>CGTGC</td>
<td>0.15</td>
</tr>
<tr>
<td>head</td>
<td>6</td>
<td>GCACGA</td>
<td>0.30</td>
</tr>
<tr>
<td>tail</td>
<td>6</td>
<td>TCGTGC</td>
<td>0.15</td>
</tr>
<tr>
<td>head</td>
<td>7</td>
<td>GCACGAG</td>
<td>0.30</td>
</tr>
<tr>
<td>tail</td>
<td>7</td>
<td>CTCGTGC</td>
<td>0.14</td>
</tr>
<tr>
<td>head</td>
<td>8</td>
<td>GCACGAGG</td>
<td>0.30</td>
</tr>
<tr>
<td>tail</td>
<td>8</td>
<td>CCTCGTGC</td>
<td>0.14</td>
</tr>
</tbody>
</table>

project), we can count the frequency of prefixes and suffixes of length 5–22, reporting those which are more frequent than expected. Sequences which appear more frequently than expected are probably the adapters or substrings of them; the longest ones which are well-conserved are (probably) the adapters. Table 3.1 shows the results of counting over-represented words in an Atlantic salmon pyloric caecum cDNA library. In this case, the head and tail adapters are the same (or rather reverse complements of each other), but in general this cannot be assumed. Table 3.2 illustrates one additional complication: despite the fact that several 12-nucleotide sequences are over-represented, there are many fewer of them than of the single 11-nucleotide sequence. It is clear in this case that the 11-nucleotide sequence is the true adapter, since the 12th nucleotide is not well conserved.

In some cases, because either

1. the library has too few representative sequences, or

2. the majority of ESTs from the library were sequenced using primers which removed the adapter sequences,
Table 3.2: Finding the end of the adapter. Though the 12-nucleotide sequences are over-represented, it is clearly because of the conserved 11-nucleotide prefix. In general, only a single leading and single trailing adapter sequence are expected in a single library. (Calculated on 44154 reads from a Rainbow trout mixed tissue library.)

<table>
<thead>
<tr>
<th>Position</th>
<th>Length</th>
<th>Sequence</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>head</td>
<td>5</td>
<td>CCACG</td>
<td>0.80</td>
</tr>
<tr>
<td>head</td>
<td>6</td>
<td>CCACGC</td>
<td>0.80</td>
</tr>
<tr>
<td>head</td>
<td>7</td>
<td>CCACGCG</td>
<td>0.79</td>
</tr>
<tr>
<td>head</td>
<td>8</td>
<td>CCACGCGT</td>
<td>0.79</td>
</tr>
<tr>
<td>head</td>
<td>9</td>
<td>CCACGCGTCTC</td>
<td>0.75</td>
</tr>
<tr>
<td>head</td>
<td>10</td>
<td>CCACGCGTCC</td>
<td>0.72</td>
</tr>
<tr>
<td>head</td>
<td>11</td>
<td>CCACGCGTCCG</td>
<td>0.26</td>
</tr>
<tr>
<td>head</td>
<td>12</td>
<td>CCACGCGTCCGG</td>
<td>0.28</td>
</tr>
</tbody>
</table>

it not possible to determine the adapters. Relatively few sequences are affected in either case, though: in the first, because there are few sequences, in the second, because the adapters are not present in the reads and so do not need to be trimmed.

Having identified the adapter sequences (where possible), masking is straightforward. It is helpful to allow for a small number of errors in the adapter to allow for sequencing errors, as well as a handful of nucleotides preceding it (or trailing it, in the case of trailing adapters) in case a short vector sequence was not correctly detected and masked. A simple approach is to align the true adapter against a prefix (or suffix, for the trailing adapter) of the sequence somewhat longer than the adapter, using a score model in which gaps at the ends of the sequences are not penalized. In this way a handful of nucleotides preceding the adapter will not interfere with detection, and by choosing the match, mismatch and indel parameters appropriately, we can ensure any required degree of fidelity of the putative adapter in the sequence to the true adapter.
Among various parameter choices explored, allowing up to two errors and searching for the adapter in a region of length twice the length of the adapter turns out to be a good compromise between detecting corrupted adapters on the one hand, and avoiding the classification of non-adapters as adapters on the other (though in fact no special harm comes from this, except that a few too many nucleotides are masked from these sequences).

Table 3.5 shows the results of screening selected cDNA libraries for adapters.

3.2 Detecting Poly(A) Tails

It is useful to be able to recognize and mask off poly(A) tails, or rather the remnants of them which appear in reads, for two main reasons:

1. if they are not masked, the assembly process tends to make false joins between unrelated sequences on the basis of similarity of long sequences of A residues, and

2. the presence of a poly(A) tail indicates that the read includes the end of the mRNA, which is relevant information in itself, and also useful for determining the orientation of the insert in the vector (3′ → 5′ or 5′ → 3′).

An additional serendipitous benefit of accurate poly(A) tail detection is that if well-defined poly(A) tails are detected on both ends of a clone, we can conclude that the clone is chimeric and discard it. Approximately 1300 clones were identified as probably chimeric due to having an excessive number of poly(A) tails.

A note on terminology: a poly(A) tail is always a sequence of A’s appended to the 3′ end of an mRNA. However, since clones may be inserted into the vector in either orientation, a poly(A) tail may appear as a sequence of A’s at the end of the insert, or as a sequence of T’s at the start of the insert: a poly(T) head. When referring to
poly(A) tail detection in general, we refer to both poly(A) tails and poly(T) heads. In some cases we refer specifically to poly(T) heads, and in some cases specifically to poly(A) tails, but the intent should be clear from context whether “poly(A) tail” refers only to poly(A) tails, or to both.

There is no correct model for poly(A) tail detection, i.e. one which correctly identifies all poly(A) tails, and does not identify any non-tail as a tail, because there may not be enough information in a read to determine whether a handful of nucleotides is a tail or not. A true poly(A) tail may be about 200 nucleotides long, which would be easy to identify. However, as mentioned in Appendix A, the cDNA library construction and sequencing processes typically truncate the poly(A) tail to about 20 nucleotides (with considerable variation, from 2 or 3 to 100 or more, based on examination of reads sequenced as part of GRASP). Even so, a well-defined poly(A) tail is easy to recognize: the standard poly(A) signal AATAAA (Strachan & Read 1996, page 20), followed by 10-25 arbitrary nucleotides, followed by 18–20 A’s, followed immediately by an adapter sequence and vector. Figure 3.1 shows an example which adheres extremely closely to the ideal. Figure 3.2 shows a less easily recognized example.

Whether it is a true poly(A) tail or not is uncertain, given only this much information: it is adjacent to vector sequence, but is short and lacks the usual signals. In fact, it is a true poly(A) tail (or rather the remnant of one), confirmed by alignment with other copies of the same mRNA that have more clearly defined tails, but that fact cannot be determined from the sequence itself.
...GGTTTCAAGCCACCCGATAACCACCTGCGACCTTTCTCCATAAAAAAAAXXXXXXXXXX...

Figure 3.2: A difficult poly(A) tail: no known poly(A) signal, no adapter sequence, and only 9 adenylate residues.

Though the discussion above is described in terms of poly(A) tails, it is usually more convenient to search for poly(T) heads, since the beginning of a read is higher quality and therefore more likely to match well.\(^3\) Finding poly(A) tails can easily be done by performing the same poly(T) head search on the reverse complement of a read, but since the quality of the tail of the read is lower, a well-defined poly(A) tail is less likely to be found.

A few methods of poly(A) tail identification are mentioned in the literature. The processing pipeline for TIGR’s Gene Indices recognize poly(A) tails based solely on the presence of a known polyadenylation signal, specifically AATAAA or ATTAAA (Frequently Asked Questions About the TIGR Gene Indices n.d.). Telles & da Silva (2001) report using BLAST to detect long poly(A) tails (by searching a database of their sequences with a long poly(A) sequence as the query), and additionally consider as poly(A) tails those sequences within 30 nucleotides of the end of a read that score at least 30 (with respect to an unstated scoring model, presumably the BLAST defaults). Scheetz et al. (2003), looking for poly(T) heads, search for a region of high T density within a few nucleotides of an adapter sequence. Mao et al. (2003), also searching only for poly(T) heads, use a sliding window with user-specifiable parameters for window size, number of allowable non-T nucleotides per window, and minimum length of the longest run of T in the window.

None of these techniques is completely satisfactory. TIGR’s approach is not very effective for salmonid fishes, since only about 80% of salmonid ESTs have a known

---

\(^3\)The quality of a chromatogram is highest near the start, and degrades over the length of the read, so the most accurate data are near the start.
poly(A) signal (see Section 3.2.1). Telles and da Silva’s technique misses short poly(A) sequences (less than 30 nucleotides, which implies that it would miss most of those present in our data). Scheetz et al. and Mao use variations on a sliding window; both suffer from the problem of determining exactly where the poly(T) head ends. Scheetz in particular makes no attempt to determine the exact endpoint of the region, except in cases where it is extremely clear.

We experimented with other ad-hoc techniques which also turn out not to work well in recognizing complete poly(T) heads, that is, they don’t detect them at all, or detect part of the sequence but not the whole thing. A naive first attempt was to search for the adapter sequence followed by a sequence of at least 18 T symbols. (In this and all following discussion, we assume that vector has been masked off, and that the sequence is oriented 3’ → 5’, hence the search is for a poly(T) head.) This pattern finds perfect poly(T) heads, but misses any sequence in which the adapter is not present or is corrupted, or that includes one or more incorrect nucleotides in the first 18 T’s. In practice, this simplistic model detects 793 of 71852 poly(T) heads in a set of 220820 Atlantic salmon reads, which is clearly inadequate.

A second attempt incorporated two enhancements: the adapter can match with up to 1 error, and the poly(T) head does not stop at the first non-T position. Instead, it repeatedly searches for a pattern beginning with T and stopping after the first run of non-T positions (see Figure 3.3 for details). The process is repeated until the occurrence of such a pattern with less than 60% T positions (the figure of 60% was provided by a nearby biologist). This approach is an improvement over the naive one, but tends not to find complete poly(T) heads. One major problem is that short low quality fragments can halt the search: “...TTCTCTT...” causes the algorithm to stop searching after “...TTCTC” and declare the poly(T) head complete.

Experimentation with various parameters (altering the required percentage of T positions in a run, allowing a small number of arbitrary nucleotides before the poly(T)
Input: a reverse-complemented mRNA with vector and adapter trimmed
Output: the number of nucleotides making up the poly(T) head

seq = the input mRNA sequence
pattern = “^T+[^T]*”
headLength = 0
head = prefix of seq matching pattern

while head is at least 60% T do
    add length of head to headLength
    remove head from seq
    set head = prefix of seq matching pattern
end

return headLength

Figure 3.3: A simple, not very effective algorithm for detecting poly(T) heads.

head was considered to have begun, and so on) did not produce good results.

An alignment-based approach appears to work well. Referring to the sequence terminology introduced in Section A.3, prefix alignment is a blend of global and local alignment which, given strings S and T, finds prefixes $S_{1..i}$ and $T_{1..j}$ and an alignment of these prefixes such that no higher-scoring alignment exists for any prefixes of S and T. A prefix alignment of a sequence against a sufficiently long sequence of T symbols, with an appropriate score model, detects poly(T) heads with good accuracy.

Prefix alignment uses the initialization steps and recurrence relation of global alignment (Figure 3.4), but follows local alignment (Figure 3.5) in beginning the traceback from the highest-scoring point in the alignment. It is summarized in Figure 3.6.

The question remains of how to pick a score model. Consider an ordinary scoring model for alignment, as described in (Durbin et al. 1998, page 15) for example. There are two models to consider: the random or unrelated model $R$, and the match model $M$. In the first, nucleotides occur in aligned pairs with a frequency based only on their frequency in the two sequences. Define $q_a$ to be the overall frequency of nucleotide
**Input:** a pair of sequences $S$ and $T$

**Output:** an alignment $A$ and a score $F(A)$

**Notation:**
- $\Sigma = \{A, C, G, T\}$, the usual DNA alphabet
- $s(x, y)$ = score of an aligned pair $x, y \in \Sigma$
- $g$ = indel score
- $M$ = dynamic programming matrix
- $D$ = traceback matrix (whether to go Up, Diagonal, or Right in traceback)

**Initialization:**
- $M_{0,i} = gi$ and $D_{0,i} = R$, for $i \in \{1..|S|\}$
- $M_{j,0} = gj$ and $D_{j,0} = U$, for $j \in \{1..|T|\}$

**Filling in the matrix:**
\[
\text{for } i \text{ from } 1 \text{ to } |S| \text{ do } \\
\quad \text{for } j \text{ from } 1 \text{ to } |T| \text{ do } \\
\quad \quad M[i, j] = \max \begin{cases} \\
M[i, j - 1] + g & \text{(Up)} \\
M[i - 1, j - 1] + s(S_i, T_j) & \text{(Diagonal)} \\
M[i - i, j] + g & \text{(Right)} \\
\end{cases} \\
\quad D[i, j] = U, D, \text{ or R according to which case above was highest} \\
\text{end}
\text{end}
\]

$F(A) = M[|S|, |T|]$

**Traceback:**
- $i = |S|, j = |T|$
- $Q, R = \text{empty strings}$
\[
\text{while } i \geq 0 \text{ and } j \geq 0 \text{ do } \\
\quad \text{switch } D[i,j] \text{ do } \\
\quad \quad \text{case U: } Q = \text{“-”} | Q, R = T_j | R, j = j - 1 \\
\quad \quad \text{case D: } Q = S_i | Q, R = T_j | R, i = i - 1, j = j - 1 \\
\quad \quad \text{case R: } Q = S_i | Q, R = \text{“-”} | R, i = i - 1 \\
\quad \text{end}
\text{end}
\]
- $A = (Q, R)$

Figure 3.4: The global alignment algorithm of Needleman and Wunsch. This example uses linear gap scoring and a function $s(x, y)$ for the score of an aligned pair of non-gap symbols. “||” is the symbol for string concatenation.
Input: a pair of sequences $S$ and $T$

Output: an alignment $A$ and a score $F(A)$

Notation:
- $\Sigma = \{A, C, G, T\}$, the usual DNA alphabet
- $s(x, y) =$ score of an aligned pair $x, y \in \Sigma$
- $g =$ indel score
- $M =$ dynamic programming matrix
- $D =$ traceback matrix (Up, Diagonal, or Right or Stop in traceback)

Initialization:
- $M_{0,i} = 0$ and $D_{0,i} = S$, for $i \in \{1..|S|\}$
- $M_{j,0} = 0$ and $D_{j,0} = S$, for $j \in \{1..|T|\}$

Filling in the matrix:
for $i$ from 1 to $|S|$ do
  for $j$ from 1 to $|T|$ do
    $M[i,j] = \max \begin{cases} M[i-1,j] + g & \text{(Up)} \\ M[i,j-1] + s(S_i, T_j) & \text{(Diagonal)} \\ M[i-1,j-1] + g & \text{(Right)} \\ 0 & \text{(Stop)} \end{cases}$
  end
  $D_{i,j} =$ U, D, R, or S according to which case above was maximum
  if $M_{i,j} > m$ then
    $m = M_{i,j}$, $mi = i$, $mj = j$
  end
end
$F(A) = M[|S|, |T|]$  

Traceback:
- $i = mi$, $j = mj$
- $Q, R =$ empty strings
- while $i \geq 0$ and $j \geq 0$ and $D_{i,j} \neq S$ do
  switch $D_{i,j}$ do
    case U: $Q = "-"||Q$, $R = T_j||R$, $j = j - 1$
    case D: $Q = S_i||Q$, $R = T_j||R$, $i = i - 1$, $j = j - 1$
    case R: $Q = S_i||Q$, $R = "-"||R$, $i = i - 1$
  end
end
$A = (Q, R)$

Figure 3.5: The local alignment algorithm of Smith and Waterman. Differences between this and global alignment are marked with line numbers.
Input: a pair of sequences \( S \) and \( T \)
Output: an alignment \( A \) and a score \( F(A) \)

Notation:
\( \Sigma = \{A, C, G, T\} \), the usual DNA alphabet
\( s(x, y) = \) score of an aligned pair \( x, y \in \Sigma \)
\( g_p \) = leading indel score
\( g_i \) = internal indel score
\( M \) = dynamic programming matrix
\( D \) = traceback matrix (whether to go Up, Diagonal, or Right in traceback)

Initialization:
\( M_{0,i} = g_p i \) and \( D_{0,i} = R \), for \( i \in \{1..|S|\} \)
\( M_{i,0} = g_p j \) and \( D_{j,0} = U \), for \( j \in \{1..|T|\} \)

Filling in the matrix:
for \( i \) from 1 to \( |S| \) do
  for \( j \) from 1 to \( |T| \) do
    \[ M[i, j] = \max \left\{ \begin{array}{ll}
      M[i-1, j-1] + g_i & (\text{Up}) \\
      M[i-1, j] + s(S_i, T_j) & (\text{Diagonal}) \\
      M[i, j-1] + g_j & (\text{Right})
    \end{array} \right. \]
    \[ D[i, j] = U, D, \text{ or } R \text{ according to which case above was highest} \]
    if \( M_{i,j} > m \) then \( m = M_{i,j}, m_i = i, m_j = j \)
  end
end
\( F(A) = M[|S|, |T|] \)

Traceback:
\( i = m_i, j = m_j \)
\( Q, R = \) empty strings
while \( i \geq 0 \) and \( j \geq 0 \) do
  switch \( D[i, j] \) do
    case \( U \): \( Q = \text{“-”}||Q, R = T_j||R, j = j - 1 \)
    case \( D \): \( Q = S_i||Q, R = T_j||R, i = i - 1, j = j - 1 \)
    case \( R \): \( Q = S_i||Q, R = \text{“-”}||R, i = i - 1 \)
  end
end
\( A = (Q, R) \)

Figure 3.6: The prefix alignment algorithm. Note that the initialization and matrix-filling steps reflect global alignment, but the traceback starts at the highest-scoring point in the matrix, as in local alignment.
Table 3.3: Frequency of nucleotides in non-poly(T) mRNA, and in poly(T) head.
Non-poly(T) frequencies were computed based on 17499 masked contigs (total length 17.1 Mb). Poly(T) frequencies computed on 34961 poly(T) heads (total length 1.3 Mb).

<table>
<thead>
<tr>
<th>Nucleotide</th>
<th>Non-Poly(T)</th>
<th>Poly(T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0.27</td>
<td>0.950</td>
</tr>
<tr>
<td>non-T</td>
<td>0.73</td>
<td>0.046</td>
</tr>
<tr>
<td>A</td>
<td>0.27</td>
<td>0.012</td>
</tr>
<tr>
<td>C</td>
<td>0.23</td>
<td>0.016</td>
</tr>
<tr>
<td>G</td>
<td>0.23</td>
<td>0.013</td>
</tr>
<tr>
<td>N</td>
<td>0.0007</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Accordingly, the probability $p_R(a, b)$ of any aligned pair $a$ and $b$ of nucleotides is $p_R(a, b) = q_a q_b$, and the probability of an alignment $A = (S, T)$ is

$$P(A|R) = \prod_{i=1}^{\vert A \vert} p_R(S_i, T_i).$$

(3.1)

In the second, they occur in aligned pairs with some frequency $p_M(a, b)$ which depends upon the degree of divergence between the sequences, and the probability of the alignment under the match model is

$$P(A|M) = \prod_{i=1}^{\vert A \vert} p_M(S_i, T_i).$$

(3.2)

To compare the two probabilities, the statistic of choice is their ratio, the so-called odds ratio. To simplify calculation, the log of this ratio is considered, so the actual score of an alignment is

$$S(A) = \sum_{i=1}^{\vert A \vert} \frac{p_M(S_i, T_i)}{p_R(S_i, T_i)}.$$

(3.3)

the log-odds ratio. The individual terms $\frac{p_M(a, b)}{p_R(a, b)}$ are usually calculated beforehand, to make up a scoring matrix. Standard protein scoring matrices such as the PAM (Dayhoff et al. 1978), BLOSUM (Henikoff & Henikoff 1992) and Gonnet (Gonnet et al. 1992) matrices are calculated in this manner.
Now consider alignment of a sequence against a reference sequence composed of a single repeated nucleotide. Following a similar approach to that of the previous paragraph, by examining typical poly(T) heads and typical non-heads, we can define two models: poly(T) and non-poly(T). In the non-poly(T) model, we expect a frequency of each nucleotide of approximately their background frequency in mRNAs overall. In a poly(T) head, we expect a much higher proportion of T residues, along with some small number of other residues (due to errors in polyadenylation, in sequencing or in base-calling). Table 3.3 lists the frequencies of nucleotides in non-poly(T) mRNA sequences, and in poly(T) heads. Based on those frequencies, and treating all non-T nucleotides as one class, we can construct a score model along the lines of the log-odds scoring models described in the previous paragraph. Define two models $P$ and $N$ for poly(T) head and non-poly(T) head, each with two possible symbols $T$ and $V$ (chosen because $V$ is the IUPAC standard code for a non-T nucleotide (Cornish-Bowden 1985)). Then $P_T$ is the frequency of $T$ in poly(T) head, $P_V$ is the frequency of non-$T$ in poly(T) head, and similarly for $N_T$ and $N_V$. The log-odds score model has just two entries:

$$s(T) = \log(P_T/N_T) = \log(0.95/0.27) = 1.8$$

$$s(V) = \log(P_V/N_V) = \log(0.046/0.73) = -3.9$$

Rounding to integers provides a substitution score model of match score $m = 2$ and mismatch score $s = -4$. In this case, of course, we are not modeling an evolutionary process, but rather the alterations to the poly(A) tail introduced by the library construction and sequencing processes.

As described so far, the model does not require an alignment algorithm. It could be implemented by a single pass along a sequence, incrementing a running score by $s(T)$ at each T nucleotide, and decrementing it by $s(V)$ at each non-T nucleotide. Then the poly(T) head is the region from the beginning of the sequence to the highest-
Table 3.4: Statistics on poly(T) head lengths and preceding low quality sequence

<table>
<thead>
<tr>
<th></th>
<th>S. salar</th>
<th>O. mykiss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poly(T) heads</td>
<td>34961</td>
<td>11341</td>
</tr>
<tr>
<td>Maximum length</td>
<td>118</td>
<td>135</td>
</tr>
<tr>
<td>Mean length</td>
<td>35.9</td>
<td>33.8</td>
</tr>
<tr>
<td>Heads with leading garbage</td>
<td>3624</td>
<td>3223</td>
</tr>
<tr>
<td>Percentage with leading garbage</td>
<td>10.3%</td>
<td>28.4%</td>
</tr>
<tr>
<td>Mean length of leading garbage</td>
<td>16.0</td>
<td>16.6</td>
</tr>
</tbody>
</table>

scoring point in the sequence (inclusive). However, one complication exists. Table 3.4 provides some statistics on typical lengths and numbers of poly(T) heads. It also provides some statistics on the amount of preceding non-poly(T) sequence we typically find; this leading sequence is vector or adapter sequence which is too garbled or too short to be recognized and masked. We need to allow for this as well when applying the model.

The problem is to decide where the leading garbage characters end and the poly(T) head starts. We need some model of its length with respect to the length of the poly(T) head, so as to decide how much leading sequence followed by how much poly(T) sequence should be considered a “real” poly(T) head, as opposed to a chance run of T nucleotides near the start of an insert. The approach we take is to use prefix alignment rather than a simple sequence scan, but to assign a low (but non-zero) cost to a leading indel. In this way, some leading non-poly(T) sequence is allowed for, but too much will push the score down and cause the sequence not to be recognized as having a poly(T) head. Since it is not immediately obvious how to convert the numbers in Table 3.4 into meaningful gap scores, we include an ad-hoc leading indel penalty of $g_p = -1$, allowing for quite a few leading garbage characters if the poly(T) head is long, and very few if it is short. Internal gaps are not meaningful in this application of alignment, so the we assign a large value to the internal indel penalty.
of \( g_i = -8 \) (ensuring that no internal gaps will occur in an optimal alignment, since the cost of a gap of length \( k \) is higher than the cost of \( k \) mismatches).

Prefix alignment of a sequence with a sufficiently long\(^4\) poly(T) sequence, combined with the score model described above, is an effective detector of poly(T) heads. Its one drawback is that it is slower than simpler searches.

The only remaining question is at what score to consider a prefix of a sequence to be a poly(T) head. As discussed early in this section, there is no correct answer. Based on consultation with members of the GRASP project, a leading sequence of 8 or more T residues (with a clear adapter sequence and no extraneous nucleotides) was considered to be a poly(T) head. Accordingly, we consider a prefix to be a poly(T) head if it scores 16 or better (the match score of \( m = 2 \) for 8 nucleotides) in a prefix alignment against a sequence of T residues.

Results of this algorithm were assessed by manual inspection of several hundred sequences, covering a wide variety of possible cases, as well as by searches for remaining long poly(A) or poly(T) sequences (potential false negatives: unmasked tails) in the result sequences. Virtually all poly(A) tails longer than 8 nucleotides were correctly detected.

### 3.2.1 Polyadenylation Signals

Polyadenylation signals provide another clue to the presence of a poly(A) tail. The most common polyadenylation signal is AATAAA, the same as that found in other higher organisms. ATTAAA (also well-known in other organisms) is also common (Manley 1988). In salmonid sequences, TGTNTT (where “N” indicates “any nucleotide”) also occurs more frequently than expected, in no particular relation to sequences con-

\(^4\)How long is “sufficiently long”? The correct answer is “longer than the longest poly(T) head that is possible”, but we do not know in advance how long that is. The simple upper bound is “as long as the input sequence”, which is guaranteed to be long enough.
taining either of the two standard signals (and in no particular position relative to them or to the poly(A) tail). All three motifs were found using MotifSampler (Thijs et al. 2001). The first two were expected, being well-known signals. TGTNTT was not; it is not clear what role it plays. Table 3.6 reports statistics on the frequency and position of these motifs in the 50 nucleotides upstream of the poly(A) tail in several thousand salmon and trout ESTs.

All three motifs occur statistically more frequently in the 50 nucleotides upstream of the poly(A) tail than in randomly selected 50-nucleotide fragments of salmonid RNA. In 38142 50-nucleotide fragments chosen at random from Atlantic salmon contigs, 3.7% contained AATAAA, 2.4% contained ATTAAA, and 9.8% contained TGTNTT. These motifs occur much more frequently near the poly(A) tail. Fisher exact tests (details not shown) confirm that these motifs are statistically very significantly over-represented in the last 50 nucleotides of salmonid mRNAs. No other motifs were reported by MotifSampler.

Table 3.6 provides some statistics on the frequency of these motifs. 8864 Atlantic salmon and 3152 trout contigs with clear poly(T) heads did not include any of these signals within 50 nucleotides of the poly(A) tail; a motif search on just these contigs did not reveal any other motifs.

Table 3.7 shows the results of a Fisher exact test of the hypothesis that there is a correlation between the presence of TGTNTT and A[AT]TAAA, concluding that they occur independently.

The detection process described herein does not take advantage of the presence of poly(A) signals to improve identification of otherwise-ambiguous tails. They are not included because the process as given seems to be accurate enough for the purposes of the GRASP project. It could probably be made more accurate if this information were incorporated, at least for the sequences containing a polyadenylation signal. Future work includes incorporating this information, or at least examining it to see
how much benefit it provides.

### 3.3 Discussion and future work

Table 3.5 shows the results of adapter and poly(A/T) detection on several cDNA libraries. Some of the patterns are informative. For example, libraries rgb and rgb2 are not different libraries (they have different names for administrative reasons) but do not show the same pattern of poly(T) head or tail detection. The reason is that the great majority of sequences from rgb were sequenced with primers that remove the poly(T) head, while sequences from rgb2 were not. Note that neither one shows any adapter trimming at all; the adapter detection mechanism did not find adapters for these libraries. That is understandable for rgb, since the primers which cut off the poly(T) head also cut off the adapter; it is less clear for rgb2. Manual examination of a number of sequences shows that these libraries were made without adapters, or with very short ones (less than 5 nucleotides).

Library srkc shows many leading adapters trimmed, but very few poly(A/T) sequences or trailing adapters. The implication is that it was sequenced with a primer that includes the adapters, but the library creation process removes the bulk of poly(A/T) sequences. The absence of trailing adapters implies that the majority of inserts are longer than a single read, so the trailing adapters are never seen. On the other hand, the relatively large number of trailing adapters detected in the evf library suggests that the inserts for this library are relatively short.

It appears that most or all the elements of sequence masking could be handled cleanly using a hidden Markov model. Factors which are currently disregarded or handled in an ad-hoc manner could be incorporated in a straightforward manner: the likelihood of leading unmasked sequence prior to a poly(T) head, the presence of an adapter sequence, the presence of a poly(A) signal, and so on. Given the consider-
able information we have regarding the frequency of features like adapter sequences, polyadenylation signals and leading low-quality sequence, a fairly accurate model could be constructed. Since this idea occurred to the author too late to implement it, we leave it as future work.
<table>
<thead>
<tr>
<th>Library</th>
<th>Count</th>
<th>Adapter (head)</th>
<th>Poly(T)</th>
<th>Poly(A)</th>
<th>Adapter (tail)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Atlantic salmon</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>brh</td>
<td>1161</td>
<td>1</td>
<td>1063</td>
<td>43</td>
<td>61</td>
</tr>
<tr>
<td>evd</td>
<td>4752</td>
<td>1990</td>
<td>2402</td>
<td>2141</td>
<td>2154</td>
</tr>
<tr>
<td>eve</td>
<td>4413</td>
<td>2127</td>
<td>2178</td>
<td>1508</td>
<td>1716</td>
</tr>
<tr>
<td>evf</td>
<td>4994</td>
<td>2485</td>
<td>2357</td>
<td>1979</td>
<td>2036</td>
</tr>
<tr>
<td>ga</td>
<td>2308</td>
<td>0</td>
<td>2276</td>
<td>89</td>
<td>1237</td>
</tr>
<tr>
<td>lna</td>
<td>1379</td>
<td>4</td>
<td>1193</td>
<td>43</td>
<td>940</td>
</tr>
<tr>
<td>mgd</td>
<td>1287</td>
<td>3</td>
<td>1156</td>
<td>18</td>
<td>326</td>
</tr>
<tr>
<td>mge</td>
<td>1903</td>
<td>12</td>
<td>1708</td>
<td>161</td>
<td>106</td>
</tr>
<tr>
<td>nwh</td>
<td>3832</td>
<td>35</td>
<td>2358</td>
<td>148</td>
<td>1310</td>
</tr>
<tr>
<td>ob</td>
<td>2625</td>
<td>3</td>
<td>2606</td>
<td>74</td>
<td>1343</td>
</tr>
<tr>
<td>pitl</td>
<td>2883</td>
<td>19</td>
<td>2731</td>
<td>90</td>
<td>816</td>
</tr>
<tr>
<td>phnb</td>
<td>12435</td>
<td>5253</td>
<td>5061</td>
<td>1631</td>
<td>2454</td>
</tr>
<tr>
<td>rgb</td>
<td>25839</td>
<td>0</td>
<td>558</td>
<td>517</td>
<td>0</td>
</tr>
<tr>
<td>rgb2</td>
<td>97171</td>
<td>0</td>
<td>40525</td>
<td>27066</td>
<td>0</td>
</tr>
<tr>
<td>shc</td>
<td>1644</td>
<td>17</td>
<td>1204</td>
<td>27</td>
<td>41</td>
</tr>
<tr>
<td>sjb</td>
<td>4780</td>
<td>2517</td>
<td>2262</td>
<td>2229</td>
<td>2424</td>
</tr>
<tr>
<td>srkc</td>
<td>1627</td>
<td>1424</td>
<td>85</td>
<td>29</td>
<td>55</td>
</tr>
<tr>
<td>tc</td>
<td>2038</td>
<td>416</td>
<td>1272</td>
<td>481</td>
<td>617</td>
</tr>
<tr>
<td><strong>Rainbow trout</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>groo1</td>
<td>3677</td>
<td>3362</td>
<td>651</td>
<td>698</td>
<td>2678</td>
</tr>
<tr>
<td>pit</td>
<td>3682</td>
<td>1703</td>
<td>1502</td>
<td>210</td>
<td>232</td>
</tr>
<tr>
<td>rblb</td>
<td>19395</td>
<td>9064</td>
<td>7772</td>
<td>4051</td>
<td>3974</td>
</tr>
<tr>
<td>rDNA</td>
<td>3837</td>
<td>12</td>
<td>3064</td>
<td>62</td>
<td>1620</td>
</tr>
<tr>
<td>rtdh</td>
<td>1096</td>
<td>1021</td>
<td>57</td>
<td>158</td>
<td>728</td>
</tr>
<tr>
<td>rtwh</td>
<td>44154</td>
<td>37784</td>
<td>743</td>
<td>9601</td>
<td>79</td>
</tr>
<tr>
<td>swel1</td>
<td>2952</td>
<td>2681</td>
<td>550</td>
<td>592</td>
<td>2066</td>
</tr>
<tr>
<td>tcaa</td>
<td>1231</td>
<td>182</td>
<td>455</td>
<td>25</td>
<td>786</td>
</tr>
<tr>
<td>tcab</td>
<td>1034</td>
<td>113</td>
<td>6</td>
<td>336</td>
<td>563</td>
</tr>
<tr>
<td>tcac</td>
<td>4110</td>
<td>3888</td>
<td>1928</td>
<td>210</td>
<td>1622</td>
</tr>
<tr>
<td>tcad</td>
<td>6370</td>
<td>5774</td>
<td>3241</td>
<td>596</td>
<td>2616</td>
</tr>
<tr>
<td>tcav</td>
<td>3426</td>
<td>2958</td>
<td>1095</td>
<td>98</td>
<td>1230</td>
</tr>
<tr>
<td>tcay</td>
<td>26051</td>
<td>23485</td>
<td>9585</td>
<td>1760</td>
<td>8969</td>
</tr>
</tbody>
</table>

Table 3.5: Number of adapters and poly(A/T) sequences found in reads from selected cDNA libraries.
Table 3.6: Frequency of over-represented motifs within 50 nucleotides of poly(A) tails.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Count&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Frequency</th>
<th>Mean Position&lt;sup&gt;b&lt;/sup&gt;</th>
<th>Std. Dev.&lt;sup&gt;c&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlantic salmon</td>
<td>AATAAA</td>
<td>33584</td>
<td>67%</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>ATTAAA</td>
<td>8710</td>
<td>17%</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>TGTNTT</td>
<td>11738</td>
<td>24%</td>
<td>31</td>
</tr>
<tr>
<td>Rainbow trout</td>
<td>AATAAA</td>
<td>13132</td>
<td>63%</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>ATTAAA</td>
<td>3654</td>
<td>18%</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>TGTNTT</td>
<td>4546</td>
<td>22%</td>
<td>31</td>
</tr>
</tbody>
</table>

<sup>a</sup>Count is based on 49783 salmon and 20879 trout ESTs with unambiguous poly(T) heads (length at least 20 nucleotides). The 50 nucleotides upstream of the poly(T) head were searched for common motifs.

<sup>b</sup>Position is measured as number of nucleotides between the 5′ end of the signal and the polyadenylation site.

<sup>c</sup>Standard deviation of the distance from the polyadenylation site.

Table 3.7: Fisher’s exact two-tailed test for the hypothesis that there is a correlation (positive or negative) between the presence of TGTNTT and either AATAAA or ATTAAA motifs within 50 nucleotides of the poly(A) tail. The low p-value indicates that there is no apparent correlation between the motifs.
Chapter 4

Assembling messenger RNA in the presence of chimeras

4.1 Introduction

Sequence assembly, the process of discovering the nucleotide sequence of a DNA molecule from a collection of overlapping fragments, is a fundamental tool in molecular biology. Since technological limitations currently prevent biologists from directly sequencing more than a few hundred (typically 600–800) nucleotides of a DNA molecule at one time, there is no alternative to sequencing larger molecules in small fragments, then putting the fragments together. One approach, primer walking, involves sequencing one end of a molecule, then the next few hundred nucleotides, then the few hundred after that and so on, in order, until the complete sequence is known. This approach is reliable, but expensive and time-consuming. The currently more popular approach, shotgun sequencing, is to make many copies of the molecule of interest, break them up into small fragments, sequence a random set of fragments in parallel (hundreds at a time), and use overlaps between fragments to reconstruct the
original sequence. (In practice, these methods are used together: shotgun sequencing until most of the sequence is known, then primer walking to fill in the gaps.)

Until recently, improvements in sequencing technology produced longer and better quality reads, making the assembly process easier. However, the new direction for sequencing is very large numbers of relatively short reads. For example, ABI’s “SOLiD” technology produces up to 6 gigabases of sequence in a single run, in fragments of 25–35 nucleotides, and 454 Life Sciences’ technology produces up to 25 megabases, in reads of up to 100 nucleotides. Considering this trend, the sequence assembly problem is likely to become more important, and more challenging, rather than less so.

Shotgun sequencing of mRNA is a similar problem, with some additional complications. Instead of sequencing a single large molecule, the problem is to sequence fragments of many mRNA molecules, and reconstruct the set of mRNA transcripts from the fragments (not knowing in advance which fragments are parts of the same molecule).

Two complications make the problem harder than assembly of a single sequence. First, the same DNA sequence may give rise to more than one type of mRNA sequence, via alternative splicing: mRNAs are altered after being transcribed from DNA, by having sections cut out (“spliced”). If different sections are cut out, the final mRNAs will be identical in some parts but completely different in others (“splice variants”). Second, the laboratory processes involved in sequencing DNA or mRNA may create chimeras, sequences which appear to be one sequence but are in fact composed of pieces of two (or more) unrelated sequences. Chimeras are a problem for normal sequencing, as described below, but not a significant one. In mRNA sequencing, they may be impossible to distinguish from splice variants, and so are a bigger problem.

One can, to some extent, ignore the problem of chimeras by considering as reliable only sequences for which there are at least two or three fragments covering each region.
(Since chimeras occur randomly, it is highly unlikely that two chimeras will agree; hence if we have multiple agreeing fragments, they are likely not chimeric.) However, since mRNA transcripts occur in widely varying frequencies (the most frequent may be five or six orders of magnitude more common than the least frequent), and since low-frequency transcripts are often of great interest, we need to find some way to preserve as much data as possible.

One solution is to look elsewhere for reinforcing sequences: similar transcripts from closely related species (or the same species), homologs from other organisms, available via public databases, and so on. This chapter describes the analysis of salmon and trout mRNAs, and the steps taken to validate as many sequences as possible.

4.2 Assembling a single string

Given the standard nucleotide alphabet $\Sigma$ and a string $S \in \Sigma^*$, the assembly problem is to reconstruct $S$ given a set of substrings $s_i$ of $S$. In its simplest form, the characters of $S$ are selected uniformly at random from $\Sigma$, $|S| \gg |s_i|$, $\Sigma$ is finite, the $s_i$ are all approximately the same length, and are distributed uniformly at random in $S$. $|S|$ and $|s_i|$ are known (approximately). Typical values for $|\Sigma|$, $|S|$ and $|s_i|$ are 4, 100,000 and 500 characters, respectively.

In this form, the problem is simple. If we observe that the last $k$ characters of $s_i$ equal the first $k$ characters of $s_j$, then the probability that $s_i$ and $s_j$ do not overlap is $p = (1/|\Sigma|)^k$. By choosing $k$ appropriately, $p$ can be made sufficiently small that an incorrect assembly is unlikely. $S$ can be reconstructed by finding overlapping pairs of fragments and building up progressively longer fragments of $S$. Figure 4.1 shows a sketch of a partially assembled sequence. More substrings will be needed to complete the assembly. In general, we will not have enough $s_i$ to fully reconstruct
The assembly of DNA sequences from overlapping fragments is a challenging problem. Figure 4.1 illustrates a string $S$ and its overlapping substrings $S_i$.

During the shotgun sequencing phase, we aim to reconstruct the sequence $S$ from the fragments we have. Given some simplifying assumptions, it can be shown (Ewens & Grant 2001) that if the cumulative length of the $s_i$ is approximately $|S| \times 4.6$, then the expected proportion of the sequence which is covered is about 0.99. Other analyses (Arratia et al. 1991, Schbath 1997, Schbath et al. 2000) remove some of these simplifying assumptions.

Three complications make the problem harder. One is that the $s_i$ are not perfect substrings of $S$. Each character of $s_i$ is associated with a “quality value” reflecting the probability that it is correct. It turns out that taking the quality values into account in computing the likelihood that an overlap is real is relatively straightforward. (It is also the case that the members of $\Sigma$ are not in general equally frequent in the target string; this does not affect the difficulty of assembly, but calls for minor modifications to the probability calculation above.) The second problem is that $S$ may contain regions of very high similarity known as repeats. Repeats are interesting in their own right (see Achaz et al. 2002, Rocha et al. 1999, Jurka 1998, Heringa 1998, Cox & Mirkin 1997 for a small sampling), but also make assembly difficult, because apparently overlapping substrings may not in fact belong together. Although an interesting topic, we will not address it further here, beyond noting that Pevzner et al. (Pevzner et al. 2001a) provide an interesting solution. Rather than attempt to distinguish between copies of a repeated element, they merge them, so that the assembled sequence is represented as a cyclic directed graph. The loops are unraveled at the end of the process, using the lengths of individual clones to determine the order.
Chimeras are the third complication. As a consequence of the laboratory processes used in sequencing, some percentage of substrings will incorporate non-adjacent fragments of $S$. Figure 4.2 shows a cartoon of real and chimeric substrings of $S$. More formally, a real substring $s_i = S_{j..k}$ where $1 \leq j \leq k \leq |S|$. A chimeric substring is $S_{j..k,x..y}$ where $1 \leq j \leq k \leq |S|$ and $1 \leq x \leq y \leq |S|$ and $x - k \neq 1$ (in other words $S_k$ and $S_x$ are not adjacent in $S$). The problem is that chimeric substrings can lead to an incorrect reconstruction of $S$. Note that there is no way of identifying a substring as chimeric in isolation; it looks just like any other substring. Only by comparing it with others can we determine that there is an inconsistency. The frequency of chimeras varies depending on details of the laboratory processes, from near zero to perhaps 10% seen in some cDNA libraries we have analyzed; in one notorious case more than 50% of the sequences were chimeric (Anderson 1993).

Chimeras can be detected computationally provided there are enough real substrings. Sooner or later one will conflict with the chimera; at that point it is clear that one or the other is chimeric, but it may not be clear which one. However, since (eventually) multiple real substrings will cover any given region of $S$, sooner or later other substrings will reinforce the correct substring and contradict the chimera. The problem is slightly more complicated than stated, because an apparent conflict may also indicate a repeated region. The same logic applies, though: real substrings will

---

**Figure 4.2:** Cartoon of real and chimeric substrings of $S$. The real substring is shown as a solid line, and the chimeric substring is shown as a dotted line.
sooner or later be reinforced by others. In the end, the consequences of chimeric substrings are that the assembly process must be able to reject a substring as chimeric if it conflicts with other substrings (as opposed to being strictly greedy in pairing up substrings), and in general more substrings will be needed to ensure that $S$ is correctly reconstructed. Side note: the ends of the string can be sequenced directly, so we do not need to worry about chimeras which join one end of the string to some other part; we can detect those immediately.

### 4.3 Assembling messenger RNA

Suppose instead of having a single string $S$, we have a pool of many strings $S_1, \ldots, S^k$. As before, we are given substrings $s_i$, but in this case we do not know to which string a given substring belongs, or how many strings we are trying to assemble. This is the problem of reconstructing mRNA sequences after RNA has been extracted from a tissue sample and sequenced. Although we still know the approximate lengths of the $s_i$, we no longer know the lengths of the $S^i$, although in general they will range from a few hundred to a few thousand nucleotides.

One complication is removed: repeats are infrequent (although not completely unknown) in mRNA.

A new complication is added: many genes have more than one *splice variant*: two mRNA sequences from the same gene might differ in various ways: one might be truncated at one or both ends of the string; they might share identical ends, but differ completely in the middle of the string; they might share one end, but differ completely at the other end (and so on). (See, for example, (Strachan & Read 1996) or any general molecular biology text for a discussion of the biology behind splice variants.) Chimeras may be hard to distinguish from splice variants.

One complication is made more serious: we can no longer assume that enough
good data will eventually contradict every chimera. If a chimera joins the very end (or start) of a string with an unrelated sequence from another string, no amount of additional data will produce a conflict. Figure 4.3 shows how two separate mRNAs may be joined by a chimeric fragment. No further fragment will contradict the join (unless there is also a splice variant here); we will conclude that $S^1$ and $S^2$ are part of one mRNA.

One further difficulty is that the frequency of different strings varies considerably. The most frequent strings might be 5 to 6 orders of magnitude more frequent than the least. This is particularly irksome in that it is often the least frequent (hence least likely to have lots of substrings to give us confidence in the string) which are the most interesting.

Yet another issue is that some chimeric sequences may in fact be real transcripts, produced from more than one gene. Recent work by Akiva et al. (2006) and Parra et al. (2006) shows that adjacent genes can be transcribed as a single RNA sequence, parts of the sequence spliced out, and the remainder used to create a functional protein which is constructed from two genes. This phenomenon turns out to be sufficiently common that Kim et al. (2006) developed ChimerDB, a public database to store information about them. Other work by Unneberg & Claverie (2007) raises the possibility of transcripts being formed from parts of genes on different chromosomes, if they are in physical proximity during transcription. Lacking a complete genome
for salmonids, we cannot easily distinguish between chimeric ESTs that are artefacts of the cDNA library construction, and real but chimeric transcripts. As discussed in the next section, the only hope for distinguishing these unusual transcripts from chimeras is the presence of reinforcing data, such as multiple copies of the transcript, or similar transcripts in other organisms.

4.4 Identifying non-chimeras

The basic solution to identifying chimeras in assembly mRNAs is the same as for single strings: look for reinforcing data. However, since there are many strings for which we only find a small number of substrings, we need to look elsewhere for reinforcement. Rather than ask which substrings are chimeric (by noting that multiple substrings disagree with it), ask which strings we can be confident are not chimeric.

The initial step is to examine the assembled sequence itself. It may be covered by enough fragments that it is clearly real. We can count the “blocks” in an assembled sequence: the regions which are sufficiently well-covered that we can be confident they are reliable. If there is only one, the sequence is not chimeric.

Beyond that, there are several sources of confirmation. In the experiments described in this work, we have sequences from two species, Atlantic salmon and Rainbow trout. These species are near relatives, so it is straightforward to identify homologous sequences via BLAST or other comparison methods. If we find that putative strings from the two species are highly similar for their whole length, then we can have confidence that they are both real. In addition, salmon and trout are pseudotetraploid: having undergone a whole-genome duplication several million years ago, they have multiple copies (different enough to be distinguished, but similar enough to be identifiable) of most genes; if the orthologs arising from the genome duplication are both present, and sufficiently similar, we can conclude that they are not chimeric.
More distantly related organisms may also help. By blasting against the GenBank non-redundant nucleotide and protein databases and the curated Gene Ontology (GO) protein database, we can confirm the real status of more sequences. Comparison with the Gene Ontology database is beneficial for at least two reasons, beyond merely supporting the non-chimeric status of a sequence:

1. Because entries in the database are curated, a match against one of them is better confirmation that the query sequence is real than a similar match against a sequence in NR (which may be a predicted protein with no experimental evidence, and may never have been reviewed by a person).

2. The association of sequences in GO with gene ontology terms allows us to assign putative functions to genes, and also to tissues or cell types. For example, Rise et al. (2004) compared the GO terms associated with sequences from a pyloric caeca cDNA library with GO terms associated with sequences from other tissues. They found that the pyloric caeca library had a statistically significantly higher frequency of sequences associated with redox control and the prevention of systemic uptake of xenobiotics, and concluded that these are possible functions of the pyloric caeca.
<table>
<thead>
<tr>
<th></th>
<th>Salmon</th>
<th>Trout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragments</td>
<td>99747</td>
<td>127099</td>
</tr>
<tr>
<td>Contigs</td>
<td>24549</td>
<td>37617</td>
</tr>
<tr>
<td>1Block</td>
<td>4345</td>
<td>8324</td>
</tr>
<tr>
<td>Paralogs</td>
<td>6500</td>
<td>4302</td>
</tr>
<tr>
<td>Orthologs</td>
<td>17194</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Table 4.1: Results of various criteria for establishing the non-chimeric status of assembled sequences. 1Block: how many contigs form a single block. Paralogs: how many contigs are reinforced by a paralog. Orthologs: how many sequences are reinforced by a contig in the other species.
Chapter 5

Estimating the Transcriptome Size

The *transcriptome* of an organism is the set of all messenger RNA transcripts the organism can produce. Various researchers have addressed the question of the size of the transcriptome of an organism. Harrison et al. (2002) refer to the proteome, but in fact examine the number of identifiable open reading frames in the genome of an organism to estimate the number of transcripts, then assume that the proteome is accurately estimated by the number of transcripts. This approach is applicable only to organisms for which full genomic sequence is available, but serves to illustrate that the question is of interest. There is not a one-to-one mapping between genes and transcripts, due to alternate splicing (multiple transcripts from one gene). There is also not a one-to-one mapping between transcripts and proteins, due to post-translational modifications of proteins, such as the transcription of multiple peptides in a single mRNA which are cleaved into multiple functional molecules after translation. However, in the absence of other sources of information, the size of the transcriptome is a reasonable estimate of the size of the proteome, and if splice variants are merged, it may also estimate the number of genes reasonably well.

Unlike the genome, but like the proteome, one may meaningfully refer to the
transcriptome of a tissue, cell type, life stage of an organism, and so on, since the transcriptome reflects the set of proteins being manufactured, which varies with the criteria above (among many others). An even simpler question of interest is how many distinct mRNAs a cDNA library contains. If a library represents a single tissue or cell type, or a single life stage, and so on, the number of distinct transcripts is of interest as a measure of the complexity of this tissue, cell type, etc, but even if not, it gives the researcher an estimate of what percentage of transcripts in the library have been found, hence how many remain unsequenced. We would like to estimate this number (without additional lab work), based on the number of copies of each transcript seen so far.

Laboratory-based approaches to finding the size of the transcriptome include massively parallel signature sequencing (MPSS) (Brenner et al. 2000) and serial analysis of gene expression (SAGE) (Velculescu et al. 1995, Saha et al. 2002). MPSS and SAGE are laboratory processes which produce a characteristic (or “signature”) nucleotide sequence for many transcripts simultaneously. MPSS generates up to millions of 16-20 nucleotide signatures; SAGE generates thousands of 9-10 nucleotide signatures; LongSAGE (a variant of SAGE) generates thousands of 17-nucleotide signatures. In the case of MPSS, a single experiment will, with high (> 99%) probability, include at least one representative of each transcript (Brenner et al. 2000). Computing the size of the transcriptome is then a matter of counting distinct signatures.

There are a few drawbacks to these techniques. One is that a 20 nucleotide fragment may not be enough to distinguish transcripts (since there is considerable sequence conservation across genes, especially in taxa with many paralogs, such as salmonids). Another is that some genes do not contain the necessary 4-nucleotide sequence to initiate the process; those genes will systematically be missed by all of these techniques. Lynx Therapeutics finds that 4% to 5% of genes in humans and rodents are missed by MPSS due to one of these reasons (Lynx Therapeutics n.d.).
Since SAGE and LongSAGE typically generate far fewer signatures than MPSS, one may assume that more transcripts will be missed. In all cases, the process is expensive and requires extensive lab work. Considering that EST projects typically involve sequencing a large number of ESTs anyway (for example to construct a microarray), it would be helpful to estimate the size of the transcriptome using these data, without extra lab work or expense.

It is not always possible to classify ESTs correctly. Sequencing errors may cause two ESTs to appear to represent distinct (although very similar) transcripts. Conversely, distinct transcripts may be similar enough that any differences are incorrectly dismissed as sequencing errors. In addition, it might be impossible to recognize that two ESTs are fragments of the same transcript if they do not overlap. A typical EST is much shorter (say, 500 nucleotides) than a typical transcript (typically 1500 to 2500 nucleotides, but possibly as long as several thousand nucleotides), so it is likely that two ESTs from the same transcript will not overlap. This analysis does not address these issues; we simply note in passing that it is much more likely that we will treat two ESTs representing the same transcript as different, than treat distinct ESTs as the same. This bias will tend to skew our estimates to the high side.

A formal description of the problem is given in section 5.1. Sections 5.2 and 5.3 apply two approaches to estimating transcriptome size to a set of salmonid EST sequences. Results of applying these techniques to several cDNA libraries are reported in section 5.4.

5.1 Problem Formalization

Assume a population of size \( N \), composed of members of \( K \) classes (arbitrarily labeled with integers from 1 to \( K \)). Class \( i \) has population frequency \( p_i \) (so \( \sum_{i=1}^{K} p_i = 1 \)). We are given a random sample of size \( n \) (with \( n \ll N \), so we can assume sampling with
replacement) from the population, in which class $i$ is represented $f_i$ times. Define an indicator variable $I_i = \begin{cases} 1 & \text{if } f_i > 0, \\ 0 & \text{otherwise}. \end{cases}$ Then $k = \sum_{i=1}^{K} I_i$ is the number of classes represented in the sample. $N$, $K$ and the $p_i$ are unknown; $n$, $k$, and the $f_i$ are known. For the classes not represented in the sample, $f_i = 0$; we do not know how many such classes there are, of course. The problem is to estimate $K$ given $n$, $k$, and the $f_i$.

Some additional notation will be helpful: $n_i$ is the number of classes represented $i$ times in the sample (so $n_0$ is the number of classes not represented in the sample). Clearly $n = \sum_{i=1}^{n} in_i$, $k = \sum_{i=1}^{n} n_i$, and $K = k + n_0$, so an equivalent formulation of the problem is to estimate $n_0$ given the $n_i$ ($i > 0$).

It will not be possible in general to solve the problem as stated. Consider the sample coverage $C = \sum_{i=1}^{K} p_i I_i$, the proportion of the population which is represented in the sample (in the sense of having at least one member of its class in the sample). The frequency in the population of the $n_0$ remaining classes together is $1 - C$, but it might be distributed among arbitrarily many classes, if their frequencies are arbitrarily low.

### 5.2 The Coupon Collector’s Problem

One simplifying assumption is that the population distribution of transcripts is uniform, that is, that $p_i = 1/K$ for all $i$. Although this assumption is incorrect, its results might be useful if the true frequencies of transcripts are not too far from uniform. Soares et al. (1994) report that normalization of an mRNA library can result in a difference of frequencies between the most and least frequent transcripts of 10 to 1 in some cases, which is (perhaps) near enough to uniform.

With this assumption, the problem can be viewed as a variant of the coupon collector’s problem. Suppose that $K$ different coupon types can be found in cereal boxes, with equal probability of finding each type in any given box. One might
naturally ask how many boxes of cereal she will probably have to buy to collect at least one coupon of each type, the probability of finding all $K$ in $n$ boxes, and so on. Motwani & Raghavan (1995, pages 57–63) provide a detailed analysis of this problem.

The present problem is the inverse coupon collector’s problem, in which we know the number of samples and wish to estimate the number of coupon types. Harris (1968) among others provides a conceptually straightforward solution (the notation given here is based on Dawkins (1991)). For a given $n$ and $K$, the probability $p(k, n, K)$ of seeing exactly $k$ classes in the sample is

$$p(k, n, K) = \binom{K}{k} \frac{1}{K^n} \sum_{j=0}^{k} \binom{k}{j} (k-j)^n (-1)^j$$

(5.1)

so the maximum likelihood $\hat{K}$ is the one which maximizes the value of the expression, for fixed $k$ and $n$. This expression is difficult to evaluate directly for $k$ and $n$ in the range of interest. Langford & Langford (2002) suggest considering the ratio of successive terms (as $K$ increases): treat $K$ as real and solve $p(k, n, K+1)/p(k, n, K) = 1$ for $K$ (with fixed $k$ and $n$). The maximum likelihood $\hat{K}$ is one of the two integers on either side of the solution (which will not be an integer). Since there is guaranteed to be exactly one maximum in the range of interest, this technique gives the correct result. However, Harris (1968) provides simpler and computationally tractable approximations. The maximum likelihood $\hat{K}$ is the solution to

$$k = K(1 - e^{-n/K})$$

(5.2)

with asymptotic variance

$$\text{Var}(\hat{K}) = K/(e^{n/K} - (n/K) - 1) .$$

(5.3)

The results of applying these formulae to several EST data sets are reported in Table 5.1.
5.3 Coverage of a Biased Sample

An alternative approach is to estimate the bias of the $p_i$ from the bias of the sample. Good (1953) describes an estimator $\hat{C}$ (suggested by Alan M. Turing) for coverage of a sample from a biased population:

$$\hat{C} \approx 1 - \frac{n_1}{n}.$$  (5.4)

Chao & Lee (1992) estimate the bias of the population frequencies from the bias of the samples, then use the bias and coverage to correct the estimate of $K$. Using a preliminary estimate $\hat{K}_1 = k/\hat{C}$ (that is, that the frequencies of transcripts are the same), they suggest the following estimates of $\gamma$ (the coefficient of variation of the population frequencies) and $K$:

$$\hat{\gamma}^2 = \max \left\{ \hat{K}_1 \frac{\sum_{i=1}^n i(i-1)n_i}{n(n-1)} - 1, 0 \right\}$$  (5.5)

$$\hat{K}_2 = \frac{k}{\hat{C}} + \frac{n(1 - \hat{C})}{\hat{C}} \hat{\gamma}^2$$  (5.6)

If the bias in the population is large, they recommend using $\hat{K}_2$ to refine the estimate (by replacing $\hat{K}_1$ in equation 5.5 with the expression for $\hat{K}_2$). The resulting expressions for $\tilde{\gamma}^2$ and $\hat{K}_3$ are:

$$\tilde{\gamma}^2 = \max \left\{ \hat{\gamma}^2 \left( 1 + \frac{(1 - \hat{C}) \sum_{i=1}^N i(i-1)n_i}{(n-1)\hat{C}} \right), 0 \right\}$$  (5.7)

$$\hat{K}_3 = \frac{k}{\hat{C}} + \frac{n(1 - \hat{C})}{\hat{C}} \tilde{\gamma}^2$$  (5.8)

It should be noted that these statistics are intended to compensate for relatively small biases in the population; the stronger the bias, the more samples necessary to produce good results. See Chao & Lee (1992) for details; they demonstrate cases in which $4K$ samples are necessary to get accurate estimates of $K$. Since the biases in our cDNA libraries is relatively large, we should suspect the accuracy of these statistics.
A simpler approach might give useful results. If \( m = \min(p_i) \) (the population frequency of the least frequent class) is known, an upper bound \( \hat{K}_m \) on \( K \) is

\[
K = k + n_0 \leq k + \frac{1 - C}{m} = \hat{K}_m.
\]

(5.9)

In other words, all remaining uncounted transcripts are as rare as possible. Davidson & Britten (1979) report that the ratio of most frequent to least frequent transcripts can range from 300:1 to \( 10^5:1 \) under normal conditions, and may reach \( 10^6:1 \) in unusual tissues or conditions. Clearly the accuracy of the estimate for \( m \) will have a dramatic effect on the upper bound.

We can estimate \( m \), given a set of transcripts which are known to occur very infrequently (approximately as infrequently as the least frequent transcripts), by considering how many of them are represented in the sample. Given \( d \) such transcripts, if \( f \) individuals of these types exist in the sample, then the estimate of \( m \) is \( \hat{m} = \frac{f}{nd} \), with variance \( \frac{1}{df} \cdot \hat{m}(1-\hat{m}) / n \) (treating the data set as \( n \) Bernoulli trials with \( p = dm \)).

There are two drawbacks to this approach: first, since the transcripts of interest are (by definition) very rare, they may not occur in the sample set, or not in sufficient quantity to get useful results; second, there does not appear to be a well-known set of rare genes on which to apply this technique (based on informal conversations with a non-random sample of biologists and biochemists, and on a search of the relevant literature).

Rough estimates of \( \min(p_i) \) are used in Table 5.1. Using Davidson & Britten’s estimate of \( 10^5:1 \), and observing the frequency of the most frequent transcripts in the samples, a rough approximation of the frequency of the least frequent transcripts is 5 orders of magnitude less than that. Including the assumption that normalization reduces the difference in frequencies by up to 3 orders of magnitude (as Soares et al. claim), the frequency of the least frequent transcript in a normalized library ought to be about \( 1/100 \) that of the most frequent. (Three of the libraries described in the
Table 5.1: Estimates of $K$ using the coupon collector (CC) assumption ($\hat{K}_{CC}$), Chao & Lee's estimators $\hat{K}_2$ and $\hat{K}_3$, and $\hat{K}_m$ and $\hat{K}_{ml}$, the bounds based on $\min(p_i)$, for four cDNA libraries.

<table>
<thead>
<tr>
<th></th>
<th>rgb</th>
<th>rtwh</th>
<th>groo</th>
<th>evd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>87109</td>
<td>43714</td>
<td>2128</td>
<td>2623</td>
</tr>
<tr>
<td>$k$</td>
<td>32571</td>
<td>26643</td>
<td>1693</td>
<td>2182</td>
</tr>
<tr>
<td>$n_1$</td>
<td>15137</td>
<td>18782</td>
<td>1457</td>
<td>1863</td>
</tr>
<tr>
<td>$\max(f_i)$</td>
<td>51</td>
<td>191</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>$\hat{C}$</td>
<td>0.83</td>
<td>0.57</td>
<td>0.32</td>
<td>0.29</td>
</tr>
<tr>
<td>$\hat{m}$</td>
<td>$5.9 \times 10^{-6}$</td>
<td>$3.6 \times 10^{-5}$</td>
<td>$8.9 \times 10^{-8}$</td>
<td>$2.3 \times 10^{-5}$</td>
</tr>
<tr>
<td>$\hat{m}_{ml}$</td>
<td>$1.1 \times 10^{-5}$</td>
<td>$2.3 \times 10^{-5}$</td>
<td>$4.7 \times 10^{-4}$</td>
<td>$3.8 \times 10^{-4}$</td>
</tr>
<tr>
<td>$K_{CC}$</td>
<td>35676</td>
<td>40180</td>
<td>4469</td>
<td>6899</td>
</tr>
<tr>
<td>Std.Dev.($K_{CC}$)</td>
<td>67</td>
<td>213</td>
<td>221</td>
<td>289</td>
</tr>
<tr>
<td>$\hat{K}_2$</td>
<td>61879</td>
<td>143343</td>
<td>12369</td>
<td>9690</td>
</tr>
<tr>
<td>$\hat{K}_3$</td>
<td>85112</td>
<td>411382</td>
<td>27517</td>
<td>12153</td>
</tr>
<tr>
<td>$\hat{K}_m$</td>
<td>61385</td>
<td>38540</td>
<td>7642142</td>
<td>28788</td>
</tr>
<tr>
<td>$\hat{K}_{ml}$</td>
<td>47380</td>
<td>45440</td>
<td>3140</td>
<td>4044</td>
</tr>
</tbody>
</table>

(next section are normalized; one is not.) These estimates are given as $\hat{m}$ in Table 5.1.

A simple upper bound on $\min(p_i)$ is the frequency of the least frequent transcript in the observed data. Since every library contains many singletons, if $n$ clones have been sequenced, an upper bound on $\min(p_i)$ is $1/n$. In Table 5.1, this estimate is $\hat{m}_{ml}$, the “minimum sample frequency”.

Due to the sensitivity of the result to these estimates of $m$, the main conclusion regarding this technique is that it is not useful without much better estimates of $\min(p_i)$.
5.4 Experimental Results

Results of applying these statistics to four cDNA libraries are shown in Table 5.1. Libraries rgb (Atlantic salmon mixed tissue: spleen, kidney, brain) and rtwh (Rainbow trout mixed tissue: spleen, kidney, brain, gill, liver, muscle) were chosen because they are the largest data sets, hence contain the most information from which to compute statistics. They are both normalized (though rgb seems to be more normalized than rtwh, based on \( \max(f_i) \), the number of occurrences of their most common transcripts). groo (Rainbow trout whole tissue) is the non-normalized library for which the largest number of sequences is available. evd (Atlantic salmon thymus) is included to demonstrate the statistics on a normalized library for which relatively few clones are available.

The “coupon collector” assumption produces plausible-looking results for the libraries for which we have many clones, but they are almost certainly underestimates. It is clearly underestimating the libraries for which we have fewer data. This is as expected. Chao & Lee’s statistics turn out to be very sensitive to the bias of the sample (as they were intended to be), but perhaps too sensitive. The results for rtwh are implausible, presumably due to the existence of several clones which occur in very high copy numbers in the library. (Equations 5.5 and 5.7 are very sensitive to a small number of high-count classes, due to the \( \sum_{i=1}^{n} i(i-1)n_i \) summation: if there is a class with \( i \) much larger than most others, \( i(i-1) \) will dominate the summation, even though there may be only a single class with many instances.) The estimates based on \( \min(p_i) \) demonstrate mainly that without accurate estimates of \( \min(p_i) \), the technique is not useful: groo’s estimates are clearly outlandish; rtwh’s “upper” bound is lower than its “lower” bound; only rgb’s estimates are plausible-looking.
5.5 Summary

The pure problem of estimating $K$ from only $n$, $k$ and the $f_i$ is unsolvable. The assumption that the distribution of transcripts types in the population is uniform allows a calculation to be made, but it is biased low (very low for non-normalized libraries, or those with few clones sequenced).

Attempting to account for the bias in the population requires an estimate of the probability of the least probable transcript, or specific assumptions about the distribution of the $p_i$. The accuracy of this estimate having such a dramatic effect on the result, and the (un)likelihood of getting an accurate estimate make this technique of limited value at present.
Chapter 6

Orthologs and Paralogs

If a genome duplication occurred in the ancestral salmonid, followed by re-diploidization over the subsequent millions of years, a relatively clear signal should exist in the genome: a larger than usual number of paralogous pairs of genes, all of similar sequence divergence. Ancient genome duplications are difficult to detect reliably, as demonstrated by the ongoing controversy regarding the existence of genome duplications in the human lineage mentioned in Chapter 1. However, at least some recent ones show clear signals, as we show later in this chapter, and the hypothesized salmonid duplication cannot be older than the divergence date of salmonids and their nearest neighbors, so it ought to show such a signal. For context in the following analysis, our null hypothesis is that no genome duplication occurred, and our hypothesis is that a duplication did occur.

With the large number of ESTs now available for salmonids, it is possible to investigate this hypothesis. Our experiment is to examine the ages of paralogs in salmonids, and compare them to the ages of salmon-whitefish orthologs and salmon-smelt orthologs. If a significant majority of paralogous pairs in, for example, Atlantic salmon are older than orthologous pairs in salmon and whitefish, but younger than
orthologous pairs in salmon and pike or smelt, then there is support for the hypothesis that a single duplication could account for those pairs. On the other hand, if a significant fraction of paralogs do not occur along that specific segment of the evolutionary tree, then the hypothesis that one genome duplication event gave rise to the majority of salmonid paralogs must be called into question, or additional events must be inferred to account for these observations.

This is the first study of the hypothesized salmonid genome duplication using a large number of transcripts from across the salmonid genome. Previous studies such as (Shiina et al. 2005, McKay et al. 2004) compared individual genes or relatively short regions of the genome. Using a large set of ESTs, we can detect genome-wide patterns which cannot be observed by examining only small regions.

This chapter describes the identification of putative paralogous and orthologous pairs (and larger clusters) of transcripts, and the patterns of orthologous and paralogous duplications in salmonids. Comparative analyses of *Xenopus laevis*, the African clawed frog, a species with a well-established recent duplication, and *Danio rerio*, zebrafish, a species without a recent duplication, act as positive and negative controls on the detection of patterns arising from duplications.

### 6.1 Input Data

Table 6.1 briefly summarizes the data on which the following sections are based. ESTs were processed and assembled as described in Chapters 2-4. Lists of the accession numbers of the sequences involved are found in Appendix B.
6.2 Finding Homologs

The basic problem of identifying recent homologs is relatively straightforward; any of a variety of sequence comparison packages will suffice. We use the BLAST program blastn (Gish & States 1993) to identify homologs among our sequences.

To ensure that measurements of genetic divergence are consistent, we would like to ensure that we are always compare the same region of the transcript. It is well established that coding DNA accumulates mutations more slowly than non-coding DNA, and also at more widely varying rates. Non-coding DNA accumulates changes more rapidly, and at a more consistent rate, due to a reduced rate of purifying selection. Considering the nature of the data set in question, with a choice of coding or 3’ UTR sequence, we would prefer to use 3’ UTR sequence for all comparisons. This approach has the drawback that a great deal of data must be discarded, since many contigs do not include enough 3’ UTR sequence, or the stop codon cannot be identified unambiguously. The benefit is that measurements are consistent and comparable. Unfortunately, too few contigs in the data set have enough 3’ UTR sequence, so the analyses presented in this chapter use whatever homologous regions are available, whether coding or UTR.

To determine whether the results are significantly distorted by using mixed cod-

<table>
<thead>
<tr>
<th>Species</th>
<th>ESTs</th>
<th>Contigs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlantic salmon</td>
<td>434384</td>
<td>85157</td>
</tr>
<tr>
<td>Rainbow trout</td>
<td>127099</td>
<td>45115</td>
</tr>
<tr>
<td>Lake whitefish</td>
<td>10313</td>
<td>6446</td>
</tr>
<tr>
<td>Zebrafish</td>
<td>400000</td>
<td>71361</td>
</tr>
<tr>
<td>Clawed frog</td>
<td>275215</td>
<td>75697</td>
</tr>
</tbody>
</table>

Table 6.1: A summary of the inputs to the clustering and phylogenetic algorithms discussed in this chapter. The “Contigs” column includes both singletons and non-singletons;
ing and UTR sequence, we compare Atlantic salmon paralogous pairs using distances based on only 3’ UTR sequence versus distances using any paralogous regions, and Atlantic salmon / Rainbow trout orthologs with the same criteria. Coding and UTR regions are identified using blastx against GenBank’s nonredundant protein database. If there is no match for a transcript, it is discarded. Otherwise, the top match is used to identify the orientation of the sequence, and the stop codon. Then we discard any sequence for which we do not have at least 200 nucleotides downstream of the stop codon. These 200 nucleotide fragments are our basis for comparison.

Figures 6.1 and 6.2 show the results of comparing distances based only on 3’ UTR data with “mixed” distances based on any available homologous cDNA sequence. Figure 6.1 shows that distances which include coding sequence are slightly more conserved than UTR-only sequences, entirely as expected. However, the difference is relatively small. Somewhat surprisingly, Figure 6.2 does not show any appreciable difference between the two calculations. However, the curves are based on relatively few data, especially in the UTR-only case, so it is possible that a distinction would emerge if more data were available. Alternatively, the relaxation of purifying selection which has been observed following gene duplications may allow paralogous loci to accumulate changes in coding sequence more rapidly than in orthologous loci (Lynch & Conery 2000, Ohta 1994). Yet another possibility is that some of these paralogs may be pseudogenes which are still transcribed, but not translated, hence subject to relatively little purifying selection. Whichever is the case, since the difference is relatively small between UTR-only and mixed distances, we proceed using all available data, on the assumption that the benefit of a great deal more data overrides any possible loss of accuracy.

1If there is at least one in-frame stop codon within 30 nucleotides upstream or downstream of the end of the aligned region, the furthest upstream one is considered to be the true stop codon. If there are none, the sequence is discarded.
Figure 6.1: The difference between Atlantic salmon / Rainbow trout ortholog distances calculated based on coding sequence and UTR sequence mixed, versus based only on 3' UTR sequence. The histogram shows 22107 “mixed” distances against 5115 UTR-only distances, normalized so that the areas under the curves are the same.
Figure 6.2: The difference between Atlantic salmon paralog distances calculated based on coding sequence and UTR sequence mixed, versus based only on 3’ UTR sequence. The histogram shows 8165 “mixed” distances against 1279 UTR-only distances, normalized so that the areas under the curves are the same.
6.3 Identifying Orthologs

Finding homologous pairs between Atlantic salmon and nearby species is not hard, since they are similar enough that BLAST is more than adequate for identifying them. The challenge is to identify true orthologs, that is, genes that arose from the same ancestral locus as a result of the speciation of the most recent common ancestor of the species in question. Two problems may arise: first, a pair may have arisen from a paralogous duplication prior to the speciation; second, paralogous duplications may have occurred in one or both taxa after speciation, making it difficult to choose a particular pair of sequences as the true orthologs.

The first problem will bias the apparent speciation to appear older than it is. If two sequences, one from each species, have diverged to a distance of 0.11, while most putative orthologs are near 0.05, is it because they have diverged more rapidly than other pairs, or is it because they are not true orthologs, but rather paralogs? Figure 6.3 provides a sketch of this possibility. If

1. an ancestral gene $A$ is duplicated, giving rise to genes $A_1$ and $A_2$ in the ancestral genome,

2. then a speciation occurs,

3. then $A_1$ is lost in one lineage while $A_2$ is lost in the other,

the remaining pair may appear to be orthologs, though they are not.

The second problem is relatively trivial. Though there may be more than one ortholog in species $A$ to some gene in species $B$, it will in general be safe to choose as a representative any ortholog in $A$ which maximize similarity between itself and the gene in $B$. We can be confident that we will not underestimate divergence (disregarding the relatively rare effect of convergent evolution). If a pair is identified as orthologous, it is certain that the true orthologs (which may be this pair, or some more closely related
Figure 6.3: A sketch showing a paralogous duplication followed by a speciation, and loss of different paralogs in the two lineages. In this scenario, genes A1 and A2, apparent orthologs in salmon and trout, are in reality paralogs arising from the earlier paralogous duplication, making the speciation appear older than it is.
species count avg. dist. median dist. std. dev. avg. length

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Trout</td>
<td>22061</td>
<td>0.057</td>
<td>0.044</td>
<td>0.051</td>
<td>641</td>
</tr>
<tr>
<td>Whitefish</td>
<td>4502</td>
<td>0.088</td>
<td>0.077</td>
<td>0.053</td>
<td>626</td>
</tr>
<tr>
<td>Smelt</td>
<td>5347</td>
<td>0.269</td>
<td>0.258</td>
<td>0.096</td>
<td>679</td>
</tr>
</tbody>
</table>

Table 6.2: A summary of orthologs of Atlantic salmon transcripts found in 3 other species. Distances are Felsenstein’s F84 distance, calculated using the `dnadist` program from the Phylip package (Felsenstein 2005) on the alignable region of orthologous pairs, with gap positions removed. “Std. dev” is the standard deviation of the distances. The average length of the alignable region of the pairs is also provided, to give some indication of how much data the values are based on. Sequences are considered orthologs if they are more similar to each other than either is to any other sequence in the other species, the alignable region is at least 300 nucleotides long, and the distance is at most 0.40. More distant pairs are considered to be members of a gene family, and so are not included in the chart.

Table 6.2 summarizes the number, mean distance, and mean alignment length of orthologs to Atlantic salmon transcripts found in 3 other species. The distance is Felsenstein’s F84 distance (Kishino & Hasegawa 1989, Felsenstein & Churchill 1996), which is similar to Kimura’s classic 2-parameter model (Kimura 1980), but allows for varying frequencies of nucleotides as well as Kimura’s differing rates of transitions and transversions. Figure 6.4 displays histograms of these distances. As can be seen from the histogram, there is a wide variation in orthologous distances, particularly between Atlantic salmon and smelt. It is very likely that most of the more divergent putative orthologs between Atlantic salmon on the one side and Rainbow trout or whitefish on the other are not true orthologs, as discussed in Section 6.3. However, with no clear criteria for discriminating between them, we consider them all orthologs. With
Figure 6.4: Histogram of the evolutionary distance of transcripts in various species to their orthologs in Atlantic salmon. Distances are Felsenstein’s F84 distance, calculated as in Table 6.2. The X axis is the distance; the Y axis is the frequency of orthologous pairs at that distance.
respect to the hypothesized salmonid genome duplication, the expectation is that the majority of salmonid paralogs will occur between the salmon-whitefish and salmon-smelt ortholog curves, since that is the period of evolutionary history in which the genome duplication, by hypothesis, occurred. The next section provides some results on paralogs, and the following chapter discusses the implications.

### 6.4 Identifying Paralogs

Identification of paralogs begins with the same steps as identifying orthologs, with the distinction that in this case we are comparing a set of contigs against itself, rather than against another set. Any sequence which shows similarity to another via a `blastn` comparison is a potential paralog. Pairwise alignment can then be used to identify pairs which are alignable over a sufficient distance to be considered useful.²

Once putative paralogs have been identified, a distribution of their distances might provide evidence for or against a genome duplication. The expected pattern of paralogous pairs in a genome undergoing re-diploidization after a recent genome duplication is sketched in Figure 6.5. Given this pattern, identification of paralogs arising from the genome duplication should be simple. The techniques we use will be applied to three species: *Xenopus laevis*, the African clawed frog, which has a well-documented recent genome duplication; *Danio rerio*, the zebrafish, which has *not* had a recent genome duplication; and *Salmo salar*, Atlantic salmon, which may or may not have had a recent duplication.

A simple approach to determining this distribution is to calculate the evolutionary distance of the pairs identified as described in the previous paragraph, and draw a histogram of the results. This technique suffers from an overcounting problem.

²Paralogs are considered interesting if they can be aligned for at least 300 nucleotides, with at least 60% identities.
Figure 6.5: A sketch of the expected pattern of paralog distances in a genome undergoing re-diploidization after a recent genome duplication. The high percentage of extremely recent paralogs is normally seen in all genomes, as described by Lynch & Conery (2000) and others. When the data are derived from transcripts, as is the case in this work, it is probably a combination of recent paralogs which have not yet been silenced and alleles; these cases cannot be distinguished unambiguously in the absence of genomic DNA against which to compare the sequences. The peak labelled “approximate point of genome duplication” reflects the large number of paralogs of roughly similar divergence times which arose as a result of the duplication.
Consider the situation diagrammed in Figure 6.6: A paralogous duplication occurs at some point, but instead of leading to a single pair of genes, very recent duplications create many copies of each original paralog. If there are $n$ copies along one branch, and $m$ copies along the other, we will count $nm$ copies at approximately the distance from one group to the other, and $n(n-1)/2 + m(m-1)/2$ very recent duplications, rather than $(n-1) + (m-1)$. In the sketch, we will count all $3 \times 4 = 12$ pairs at a distance of approximately $2d$, when in truth there was only one duplication at that distance. We will also count $3 + 6 = 9$ very recent duplications rather than $2 + 3 = 5$. This overcounting greatly distorts the histogram, if there are many groups of very similar contigs; the data studied in this work do indeed suffer from this problem.

A second effort is to consider only the best match for each transcript, rather than all matches. This method will undercount. Consider again Figure 6.6. The closest match to each contig is a very near neighbour, and only those matches will be counted. The main paralogous duplication in the tree will not be counted at all. However, if enough transcripts do not have very near paralogs, perhaps the expected pattern will
Figure 6.7: Simple analysis of paralog distances, in which the best match for each transcript is considered. Distances are Felsenstein’s F84 distance.

be visible. Figure 6.7 shows an analysis of the three species of interest. The African clawed frog clearly shows the pattern of duplications expected after a recent genome duplication. The zebrafish shows the pattern of a large number of very recent paralogs steeply falling off to a low level, as expected of an organism without a recent genome duplication. The Atlantic salmon shows an unexpected pattern, with no immediately obvious explanation.

To count each duplication exactly once, it will be necessary (or at any rate convenient) to construct a phylogenetic tree for each group of paralogs. Consider again the sketch in Figure 6.6. Given a rooted phylogenetic tree representing that cluster, it is straightforward to perform a post-order depth-first traversal\(^3\) of the tree, calculating the mean distance between the leaves on each side of the node, and recording that as the distance of this particular duplication. In this way, each duplication is recorded

\(^3\)That is, a traversal which visits all of the nodes below a given node before visiting that node.
Figure 6.8: A sketch of non-overlapping contigs in a cluster. To align them successfully, at least one contig must be discarded, and the others trimmed.

exactly once. The drawback is that the construction of phylogenetic trees requires us to construct a multiple alignment of the cluster, which is time-consuming and may require the discarding of some sequences or some parts of some sequences.

To ensure that the clusters are alignable, each is filtered as follows. The contigs are sorted by length, then from longest to shortest, sequences are retained if and only if they are alignable with each contig in the retained set, and discarded otherwise. Contigs are trimmed to the length of the shortest one. Once filtered, clusters are multiply aligned using MAFFT (Katoh et al. 2002), columns with gaps are removed, Phylip’s dnahist program is used to construct distance matrices, and trees are constructed using the UPGMA algorithm as implemented in Phylip’s neighbor program (Sokal & Michener 1958, Nei 1975). UPGMA is used because it generates rooted trees, which are necessary for calculating the distance between paralogous sequences.

The pattern of paralogous duplications in Atlantic salmon does not appear very much like expected. The method appears sound, since the expected patterns are seen in zebrafish and the African clawed frog. In the next chapter, some possible explanations are advanced, and the evidence for and against a salmonid genome
<table>
<thead>
<tr>
<th>Species</th>
<th>Count</th>
<th>Mean Dist.</th>
<th>Std. Dev.</th>
<th>Trees</th>
<th>Multiple</th>
<th>Mean Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>salmon</td>
<td>8164</td>
<td>0.098</td>
<td>0.064</td>
<td>6788</td>
<td>922</td>
<td>574</td>
</tr>
<tr>
<td>zebrafish</td>
<td>4014</td>
<td>0.077</td>
<td>0.077</td>
<td>2851</td>
<td>547</td>
<td>520</td>
</tr>
<tr>
<td>clawed frog</td>
<td>7369</td>
<td>0.091</td>
<td>0.059</td>
<td>6338</td>
<td>628</td>
<td>583</td>
</tr>
</tbody>
</table>

Table 6.3: Summary of paralogs found in salmon, zebrafish and African clawed frog. Only pairs at a distance of at least 0.02 are counted. “Count” is the number of paralogs detected. “Mean Dist.” is the mean distance between them, using Felsenstein’s F84 distance as usual. “Std. Dev.” is the standard deviation of the distances. “Trees” is the number of distinct trees. “Multiple” is the number of trees with more than one duplication (that is, including at least 3 paralogs). “Mean Length” is the mean length of the sequences on which distances were calculated. The actual clusters and trees are available at [http://web.uvic.ca/grasp/gdbrown/dissertation_data/](http://web.uvic.ca/grasp/gdbrown/dissertation_data/).

Figure 6.9: Histogram of the evolutionary distance of paralogs in salmon, African clawed frog and zebrafish. This diagram shows the same pattern as Figure 6.7, indicating that the pattern is strong enough to show even when undercounting.
duplication is considered.

6.5 A Brief Aside

One early theory of salmonid evolution proposed that four independent genome duplications occurred within the salmonids (Schmidtke & Kandt 1981). The argument is based on reassociation kinetics, the rate at which DNA strands hybridize after being heated to the point of denaturation. Based on their work and earlier work by Obruchev (1967), they observed that the rate of cross-species hybridizations was slower than hybridizations of putative duplicated regions within a species, indicating that the intra-specific duplicated regions were more similar than the orthologous regions between species. This result indicated that the duplications were more recent than the speciations, contrary to the ancient single genome duplication hypothesis.

We mention this now-discredited work only because our own results suggest why they might have gone wrong: from our measurements of paralog and ortholog distances, as seen in Figure 7.1, it is clear that many paralogs are, in fact, more similar than expected, and indeed more similar than their orthologs in other taxa. Perhaps it was these paralogs that Schmidtke and Kandt observed.
Chapter 7

Evolution of the Salmonid Genome

The hypothesis of a salmonid genome duplication has gained widespread acceptance. However, the expected pattern of many paralogs of roughly similar divergence arising from the duplication, as seen in *Xenopus laevis* in the previous chapter, does not occur.

Two possibilities are:

1. A genome duplication occurred, but some other force or event(s) obscured the expected pattern, or

2. a genome duplication did not occur, but many smaller duplications account for the large genome size and large number of paralogs.

These two scenarios, and the evidence supporting or not supporting them, are described in this chapter.

As described earlier in this work, the widely accepted hypothesis is the occurrence of a genome duplication after the salmonids branched off from their nearest evolutionary neighbours, but before the salmonids themselves began to diverge into the many species now present. There are four main arguments, presented in Allendorf and Thorgaard’s classic 1984 paper (Allendorf & Thorgaard 1984), and summarized here:
1. salmonids have approximately twice the DNA content of related fish,

2. salmonids typically have about 100 chromosome arms, roughly twice as many as related fish,

3. multivalents have been observed in meiosis in several species, and

4. salmonids show high incidence of duplicated loci (paralogs).

All of these points support a genome duplication. However, the absence of the expected pattern of paralog distances calls the conclusion into question. Assuming the duplication occurred, two possibilities which account for the evidence are:

1. The tetrasomes formed during the genome duplication remained intact for some time after the duplication, preventing paralogs from diverging. Then disomic inheritance was re-established at different times for different chromosomes or regions of chromosomes, leading to the observed pattern.

2. Disomic inheritance was re-established quickly, but frequent gene conversions keep the paralogs from diverging.

Another hypothesis which is consistent with all the above arguments is a series of smaller duplications, perhaps chromosome-sized or smaller, over a long time. It accounts for salmonids having a great deal more DNA than neighbouring species, the existence of multivalents during meiosis, the high incidence of paralogs, and for salmonids having many more chromosome arms than closely related taxa. It is also consistent with a wide variation of paralog distances, with no noticeable peak of duplication. These points will be considered in the following sections.
<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Osmeridae</td>
<td>0.71</td>
<td>0.08</td>
<td>5</td>
</tr>
<tr>
<td>Esocidae</td>
<td>1.13</td>
<td>0.16</td>
<td>14</td>
</tr>
<tr>
<td>Salmonidae</td>
<td>2.82</td>
<td>0.38</td>
<td>87</td>
</tr>
</tbody>
</table>

Table 7.1: Mean haploid genome sizes in picograms, for smelt (*Osmeridae*), pike (*Esocidae*) and salmonids (*Salmonidae*), from the Animal Genome Size Database (Gregory 2007). “Count” is the number of data points for each family.

### 7.1 Genome size

Thorgaard and Allendorf use the fact that the salmonid genome is “approximately” twice the size of related species as evidence for a genome duplication. In fact, the salmonid genome is significantly more than double the size of both the smelt and pike genomes.

Table 7.1 shows the mean and standard deviation in picograms of reported haploid genome sizes reported for smelt, pike, and salmonid species. It is immediately clear that the mean for salmonids is more than double the other figures. Student’s T-tests for equal means between the smelt (multiplied by 2) and salmonid values, and between the pike (multiplied by 2) and salmonid values give a probability $p$ of the null hypothesis (of equal means) of $p < 0.0001$, indicating that salmonids have statistically significantly more than double the DNA of smelt and pike. Interestingly, pike also have statistically significantly more DNA than smelt ($p < 0.0001$ of equal means).

We conclude from this brief analysis that even if a genome duplication occurred, a significant number of other duplications must have occurred as well in the salmonid lineage. In addition, since pike also have significantly more DNA than smelt, it may be that a high rate of duplication began before the pike and salmonids diverged (but tapered off in the pike fairly quickly).
7.2 Presence of multivalents

Multivalents have been observed in several salmonid species, including Atlantic salmon and rainbow trout, implying homology between the involved chromosomes (Allendorf & Thorgaard 1984). Ohno observed 3 quadrivalents during meiosis in a male Rainbow trout (Ohno et al. 1968). However, the Atlantic salmon and rainbow trout karyotypes are very different: 54–58 chromosomes and 72–74 chromosome arms in the Atlantic salmon, versus 58–64 chromosomes and 104 chromosome arms in rainbow trout. These differences imply significant chromosomal rearrangements in one or both lineages. The possibility that 3 sets of chromosomes would remain evolving as a tetrasome, and so still be able to form quadrivalents, seems low. Very recent chromosomal duplications, on the other hand, might quite reasonably be expected to form quadrivalents.

7.3 Many paralogs

Though there are clearly many more paralogs in salmonids than would be expected in ordinary diploid organisms, the expected pattern of evolutionary distances among paralogs does not occur, as seen in the previous chapter. Instead, paralogs appear at widely varying distances, with a strong trend towards increasing numbers of paralogs at short evolutionary distances. Figure 7.1 plots the distance of Atlantic salmon paralogs on the same scale as salmon-trout, salmon-whitefish, and salmon-smelt orthologs. By the genome duplication hypothesis, most of the paralogs should fall between the whitefish and smelt curves. It is clear that there is no peak of paralog distances between those two curves. Because the smelt sequences in particular vary so widely in distance from their salmon orthologs, it is not very meaningful to consider how many Atlantic salmon paralogs fall between the means of the whitefish and smelt
Figure 7.1: Evolutionary distance of Atlantic salmon paralogs compared to the distance of orthologs between Atlantic salmon and other taxa.

curves. However, it happens that approximately 45% do fall into this range.

One further point is that 922 trees (of 6788), as reported in Table 6.2, have more than two paralogs, rather than the two which might be expected from a genome duplication. Note that this considers only sequences which have a distance of at least 0.02, to avoid counting extremely recent duplicates, alleles, and so on.

### 7.4 Number of Chromosome Arms

Allendorf and Thorgaard hypothesized that the ancestral salmonid prior to the genome duplication had 48 acrocentric chromosomes, which then duplicated to 96. Hartley (Hartley 1987) suggests that the ancestral karyotype may have undergone some changes before the duplication, though, since most taxa have 100-104 chromosome arms, and few have 96.
Though most taxa have around 100 chromosome arms, the range is from 72 (Atlantic salmon) to 170 (European grayling), according to Phillips and Ráb’s survey (Phillips & Ráb 2001). In addition, only two genera are very consistent: *On-co-rhynchus* at 100–104 arms and *Salvelinus* at 98–100. Others vary more widely: *Coregonus* ranges from 92 to 106, *Thymallus* from 146 to 170, *Salmo* from 72 to 104. Support for a single duplication is not especially strong, unless one assumes an enormous amount of rearrangement after the duplication (while retaining enough tetrasomies to form quadrivalents during meiosis).

### 7.5 Salmon Orthologs to Zebrafish Sequences

One additional line of evidence which might support a whole genome duplication in salmonids is the number of orthologs in salmonids, relative to a non-duplicated out-group. Figure 7.2 shows the number of orthologs in Atlantic salmon for approximately 24,000 zebrafish sequences. A genome duplication should reveal itself by showing a distinct peak at 2 orthologs, that is, there should be 2 salmon orthologs for many zebrafish sequences. The figure does not show this pattern. Instead, there is only one ortholog for many zebrafish sequences, two for some, and more than two for many more, suggesting that there have been multiple small duplications in salmon, rather than one genome-wide duplication.

### 7.6 Summary

With regard to the genome duplication hypothesis, the large genome size of salmonids implies a large amount of duplication. However, since it is significantly more than twice the size of the pike and smelt genomes, it is clear that a genome duplication alone is not enough to explain the data. The sorts of mutations which would generate
Figure 7.2: The number of orthologs in Atlantic salmon for 24,152 zebrafish sequences. Zebrafish amino acid sequences were obtained from the zebrafish sequencing project. Salmon orthologs were found using tblastn. Salmon sequences were counted as orthologs only if they were distinct from other salmon sequences by at least 0.05 substitutions per site (i.e. alleles etc. were not counted as orthologs).
the extra genetic material are known to be frequent: several fish have been karyotyped and found to have extra copies of chromosomes, without any apparent loss of health or fertility. If we must already assume a number of smaller duplications, it is not clear that we need a genome duplication.

The presence of a small number of multivalents is weak evidence of a whole genome duplication, made weaker by the unlikelihood of their survival through the significant genome rearrangements which have occurred. On the other hand, it appears to support the alternative of a small number of recent chromosome duplications.

The number of paralogs supports a genome duplication, but the pattern of divergence does not.

The number of chromosome arms is the one point which leans strongly to a single event, since the majority of taxa do have approximately 100 chromosome arms.

One issue which must be raised is autotetraploidy versus allotetraploidy. Allotetraploids arise when two species begin to diverge, followed by cross-breeding leading to a tetraploid organism, with one copy of each species’ genome. In this case, since the species have already begun to diverge, it is unlikely that chromosomes will form tetrasomes, so disomic inheritance should begin immediately. *Xenopus laevis* is thought to be an allotetraploid. Autotetraploidy occurs when a single species undergoes a genome duplication, acquiring two copies of the same genome. In this case, the formation of tetrasomes may prevent the copies from diverging for some time. The hypothesized salmonid genome duplication is thought to be autotetraploid, so the possibility raised earlier of a genome duplication followed by gradual breakup of tetrasomes is more likely than it would be for an allotetraploid duplication.

We argue that though the genome duplication hypothesis has not been rejected, it has also not been strongly supported. The next chapter suggests some avenues for future work which could resolve this issue.
Chapter 8

Summary and Future Work

8.1 Automated Sequence Processing

Though the sequence processing pipeline developed in the course of this work shares many similarities with others, there are three areas in which it is distinguished. Accurate trimming of ESTs was more important than in the usual assembly problem, because the presence of vector or adapters could distort evolutionary distances, cause joins between unrelated reads, and produce misleading BLAST hits, and because the presence or absence of a poly(A) tail is relevant in determining whether a read reaches the end of the transcript. Low complexity screening was necessary to remove sequences which were artefacts of the sequencing process, rather than real DNA. Beyond these steps, some automated assessment of the reliability of contigs was found to be necessary, due to the relatively large number of chimeric ESTs in the data set.

Standard methods were used to identify and screen vector sequence, though the parameters required some tuning to identify short fragments which would otherwise interfere with the subsequent identification of adapters and poly(A) tails. A low complexity screening method was developed based on statistically unusual counts
of trimers in a sequence. Note that the intent was not to identify real but low complexity DNA; rather, the task was to identify sequences which were artefacts, but which appeared in other respects to be of reasonably good quality. By examining the characteristic trimer frequencies of real and spurious sequences, a threshold model was developed which successfully screened out the great majority of the artefacts. Assembled contigs were analyzed for robustness, based on several criteria:

1. whether all or nearly all of the contig was supported by at least two clones (hence unlikely to be chimeric),

2. whether the reads included in the contig had consistent poly(A) tails, i.e. all sequences\(^1\) with tails had them on the same end of the contig,

3. whether translated BLAST hits were consistent with the structure of the contig, i.e. there were not strong matches on both strands, and the endpoints of the alignment were consistent with a non-chimeric contig.

One further refinements would be useful. The accuracy of poly(A) tail detection could be improved by considering other ESTs in a contig: if a relatively short run of A’s the end of the read (or T’s at the start) was not recognized as a poly(A) tail, but other ESTs had a poly(A) tail which started at that point, the likelihood that this fragment represented a real tail would be higher. Given that approach, the initial detection criteria could be relaxed, in the hope of avoiding incorrect identification of non-poly(A) sequence as poly(A).

It may also be useful to consider the entire trimming process in a more unified way. Currently, each stage of screening is independent: vector, then adapters, then

\(^1\)In fact, aggressive trimming of poly(A) sequence occasionally leads to adenine-rich sequences being identified incorrectly as poly(A) tails and trimmed. If a contig was otherwise sound, and only one or two reads conflicted with the majority in this regard, the contig was considered reliable.
poly(A) tails. Some means of considering the probabilities of the different components together, such as a hidden Markov model, could increase screening accuracy considerably: for example, if clear vector sequence was identified, the next few nucleotides are more likely to be adapter than otherwise, and so on. In addition, the incorporation of information about the presence or absence of polyadenylation signals would be relatively straightforward. In this way, higher accuracy could be achieved for relatively little effort.

8.2 Transcriptome Size

Estimation of the transcriptome size from a sample of ESTs would be useful both as a guide to sequencing, to estimate how many transcripts remain to be discovered in a cDNA library, and as a biological guide to the complexity of a tissue or organism. Unfortunately, as stated in Chapter 5, the problem does not appear to be solvable with available data.

Consider the set of transcript types as a population, and the sequencing of ESTs from a cDNA library as sampling with replacement from that population. Statistical techniques which are designed to estimate the size of a population from such a sample have been developed. If the probability of choosing any member of the population is equal, i.e. all transcripts are equally frequent, a reasonable job of estimation can be done. Some techniques tolerate some degree of bias in the population, i.e. when the probability of sampling different members of the population is not equal. However, none appear to be capable of accurate estimation when the difference between the frequency of the most and least probable members is 5 or 6 orders of magnitude, as is the case with transcript frequencies.

A rough upper bound would be calculable, if the frequency of the least frequent items relative to the most were known, and if at least a small number of those least
frequent transcripts occurred in the sample. It does not appear that it will be possible to know this difference with adequate accuracy, however. The other approach is an improved knowledge of the distribution of transcripts. If it were found, say, to follow some well-known distribution, then the estimation would become relatively straightforward. There does not appear to be any reason why they would follow such a distribution, though.

Considering the development of new laboratory technologies such as MPSS and SoLID, mentioned in Chapter 5, I predict that it will remain more feasible to answer this question through laboratory experiments than a purely computational approach.

\section*{8.3 Salmonid Evolution}

Salmonids have long been thought to have undergone a genome duplication, and to be in the process of rediploidization. With the objective of refining the age of the genome duplication, I set out to

1. identify paralogs within the salmonids and orthologs both within the salmonids and in nearby taxa,

2. examine the distribution of evolutionary distances to find a set of paralogs of roughly the same age, and

3. refine the previous estimate of the age of the genome duplication of “25 to 100 million years” to something more precise.

Surprisingly, the distribution of distances did not show any clustering of paralogs of roughly similar distance. The technique worked as expected in both positive and negative controls (\textit{Xenopus laevis} and \textit{Danio rerio}, respectively), so it lends credence to the methodology.
If the genome duplication occurred, we are forced to assume the existence of something which distorted the expected pattern, such as frequent gene conversions, or the preservation of tetrasomic inheritance for millions of years after the duplication, and varying points of divergence from tetrasomic inheritance across the genome. If the genome duplication did not occur, then many smaller duplications must have occurred over millions of years. The pattern of paralog distances is consistent with both models. In addition, it is known that salmonids tend to be tolerant of the presence of extra copies of some chromosomes (*aneuploidy*). The available evidence does not confirm or refute the occurrence of a genome duplication, but it presents some unexpected results which must be accounted for.

How can the question be resolved? If most regions of the genomes of most salmonids occur in homologous pairs, then a genome duplication gains credibility. If some regions are duplicated, others occur only once, and others occur more than twice, then the possibility of many smaller duplications gains credibility.

Given complete genomic sequence from several salmonids, it would be possible to detect the homologous regions, and address the question above. Similarly, genomic information from pike or smelt could be matched with salmonid genomes, to see if there are approximately two regions in salmonids corresponding with each region in the neighbouring taxa. In the absence of complete genomes, locating a large number of paralogous pairs on the genome via fluorescent *in situ* hybridization might allow some estimate of the number of copies of the duplicates, and the degree of rearrangement of the genome post-duplication. Consideration of tree topologies is also relevant, as mentioned in Chapter 7, but the available data present a rather mixed message thus far.

One further avenue of study is to consider other autotetraploid species. The catostomids, or suckers, the other large group of polyploid fish, are thought to be allopolyploid, so they are not a good choice. Two families of frogs, *Leptodactylidae*
and *Hylidae*, demonstrate autopolyploidy frequently (Becak & Kobashi 2004), but relatively few sequences are available for comparison at this time.

Whether or not a genome duplication occurred, it seems clear that *something* not well understood has created a remarkably varied and unusual collection of genomes in the salmonids, which merit considerable further research.
Bibliography


Felsenstein, J. (2005), *PHYLIP (Phylogeny Inference Package)*. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.


Hoyheim, B. (n.d.), ‘The salmon genome project’, URL: [http://www.salmongenome.no/cgi-bin/sgp.cgi](http://www.salmongenome.no/cgi-bin/sgp.cgi).


Appendix A

Molecular Biology Background

This appendix provides a brief summary of DNA, genes, RNA, DNA sequencing and evolution, to provide enough terminology to understand this work. It is neither complete nor rigorous; the intent is solely to provide computer scientists with the relevant vocabulary. This summary is loosely based on material drawn from several sources (Strachan & Read 1996, Felsenstein 2004, Graur & Li 2000), and on personal communications with members of Dr. Ben Koop’s research lab in the Centre for Biomedical Research, University of Victoria.

A.1 Basic Definitions

**nucleotide** a molecule which is a component of DNA or RNA. There are five types of nucleotides, for simplicity referred to by the first letters of their names: \( A \), \( C \), \( G \), \( T \), and \( U \).

**DNA** a DNA molecule is a chain made up of \( A \), \( C \), \( G \), and \( T \) nucleotides. For our purposes, it can be viewed as a string or sequence on a four-character alphabet \( \Sigma_D = \{A,C,G,T\} \), corresponding with the four nucleotides. DNA is a two-stranded molecule, with the strands in a complementary relationship: where one strand has an \( A \), the other has a \( T \); where one strand has a \( C \), the other has a \( G \). The *reverse complement* of a sequence is the sequence formed when each nucleotide is replaced with its complement, and the result reversed. For example, \( TTATTCGG \) and \( CCGAATAA \) are reverse complements of each other.

**RNA** an RNA molecule is a chain made up of \( A, C, G, \) and \( U \) nucleotides, represented as a string on the alphabet \( \Sigma_R = \{A,C,G,U\} \). RNA is a single-stranded molecule.

**dimer, trimer, k-mer** dimers, trimers and k-mers are 2-, 3-, and k-nucleotide strings. Trimers are also called *triplets*. 
**orientation** DNA and RNA strands have a well-defined direction, an “upstream” and a “downstream”, based on their molecular structure. The upstream direction is referred to as the 5′ (“five-prime”) end; the downstream is the 3′ (“three-prime”) end. The labeling of the ends as 3′ or 5′ arises from the chemical structure of DNA and RNA (details omitted). The two strands of a DNA sequence are in an antiparallel relationship: the 5′ → 3′ orientation of one is opposite to that of the other. DNA and RNA sequences are normally written in the 5′ → 3′ direction.

**amino acid** amino acids are the molecular building blocks of proteins, among other functions. There are 20 amino acids which are constituents of proteins.

**protein** a protein is a chain of amino acids. For our purposes, it is a string from a 20-character alphabet.

**genome** the genome of an organism is the set of DNA sequences in the nucleus of each of its cells that makes up its genetic content. It is also common to refer to the genome of a species as a whole: although each individual’s genome is slightly different, they are all much more like each other than they are like the genomes of members of other species.

**chromosome** a chromosome is a DNA sequence that makes up part of a genome. Typically, chromosomes are very long molecules, in the millions of nucleotides. A genome may be made up of 1 or several chromosomes. Chromosomes include a region called a centromere, which is not important in this context except that its position in the chromosome serves to classify chromosomes:

- *metacentric* chromosomes have the centromere roughly in the middle of the chromosome;
- *submetacentric* chromosomes have the centromere significantly closer to one end than the other;
- *acrocentric* chromosomes have the centromere extremely close to one end.

Metacentric and submetacentric chromosomes are said to have two arms; acrocentric chromosomes have one (interesting) arm. This detail is significant because the number of chromosome arms is often better-conserved during evolution than the number of chromosomes, and is sometimes used as a guide to evolutionary relatedness.

**karyotype** a genome’s karyotype is the number and type of chromosomes. For example, the human genome’s karyotype includes 5 metacentric chromosomes, and
either 13 submetacentric plus 5 acrocentric (female) or 12 submetacentric plus 6 acrocentric (male) chromosomes.

**banding pattern** By staining a chromosome with an appropriate chemical, a characteristic pattern of light and dark regions can be seen. This pattern is called a banding pattern, and serves to distinguish chromosomes which might otherwise appear very similar. For example, two acrocentric chromosomes of approximately the same length might have quite distinct banding patterns.

**genetic code** a genetic code is a function from triplets to amino acids, used to build proteins from information in RNA sequences. For example, the triplet ACG signifies the amino acid threonine. Most organisms use the same genetic code, referred to as the *standard* genetic code. See Figure A.1 and the definition of *translation* below for information on the use of genetic codes.

**codon** a triplet, used in the context of genetic codes.

**gene** a gene is a substring of a chromosome that encodes the sequence of amino acids in a protein, via a genetic code, plus upstream and downstream sequences which are needed for transcription and translation. An occurrence of a gene at a particular position in a genome is called a *locus* (plural *loci*). (There may be multiple copies of a particular gene in a genome.)

**allele** Different versions of a gene are referred to as *alleles*.

**transcription** a gene is transcribed into RNA by copying the gene’s DNA sequence into an equivalent RNA sequence (substituting U for T). A gene’s 5’ → 3’ orientation matches that of the corresponding RNA: the 5’ end corresponds with the 5’ end of the RNA, and the 3’ with the 3’ end of the RNA.

**translation** an RNA sequence is translated into an amino acid sequence according to a genetic code, using successive non-overlapping triplets to determine the sequence of amino acids in the protein. There are three possible translations of any given sequence, depending on which triplets are chosen. These choices are called the *reading frames*. Reading frame 1 uses the first three nucleotides as the first codon, and continues with non-overlapping triplets to the end of the translated region. Reading frames 2 and 3 start with the 2nd and 3rd nucleotides, respectively. Since the triplets are offset with respect to reading frame 1, the resulting amino acid sequences also differ. The 3 reading frames on the reverse complement of the sequence are referred to as frames -1, -2 and -3, so a given sequence has a total of 6 reading frames, hence 6 possible translations. For example, the nucleotide sequence “ATGTTGCCCCAAA” has these translations:
splicing an RNA sequence is spliced by removing segments of it and joining the remaining fragments together.

intron a region of a gene which will be removed by splicing.

exon a region of a gene which will be retained during splicing.

polyadenylation an RNA sequence is polyadenylated by adding approximately 200 A molecules to the 3’ end. These nucleotides are referred to as “capping” the mRNA. The poly(A) tail is thought to facilitate exporting of the mRNA from the nucleus, and to delay degradation of the molecule (Strachan & Read 1996, page 20).

messenger RNA a messenger RNA (mRNA) is an RNA sequence which has been transcribed from a gene, had its introns removed, and had a poly(A) tail added. A completed or “mature” mRNA may be referred to as a transcript.

Figure A.1 shows the so-called central dogma of molecular biology: a gene is transcribed into RNA; the RNA is modified by splicing and polyadenylation to make an mRNA; the mRNA is translated into an amino acid sequence, which folds into a three-dimensional structure to form a functional protein (folding step not shown). The structure of the post-modification mRNA is the most relevant part of Figure A.1 for present purposes. Figure A.2 shows an mRNA molecule in more detail. The four main sections are

- the 5’ untranslated region (5’ UTR),
- the coding sequence, which is translated into amino acids according to a genetic code, and which begins with a start codon (the codon ATG) and ends with a stop codon (one of TAA, TAG, and TGA).
Figure A.1: The central dogma of molecular biology: a gene is *transcribed* into a “pre-messenger” RNA molecule; the RNA molecule has its introns removed via *splicing*, and is *capped* by a poly(A) tail (a sequence of about 200 A nucleotides), resulting in a mature *messenger RNA* (mRNA); the coding region of the mRNA is *translated* into an amino acid sequence.
Figure A.2: A mature messenger RNA. The start codon ATG marks the beginning of the coding sequence; a stop codon (TAA, TAG, or TGA) marks the end. A polyadenylation signal may be present to indicate the point at which to append a poly(A) tail; if present, it is 10 to 30 nucleotides upstream of the beginning of the tail.

- the 3’ untranslated region (3’ UTR), usually including a polyadenylation signal, either AATAAA or ATTAAA, between 10 and 30 nucleotides upstream of the end of the 3’ UTR (see Section 3.2.1 for details), and
- the poly(A) tail.

The 5’ and 3’ untranslated regions are so called because they are not translated into amino acids (nor is the poly(A) tail). Although the presence of the start and stop codons is almost universal, one or the other of the polyadenylation signals shown is present in only about 80% of mRNAs, as discussed in Chapter 3.

All of the sequences which are analyzed as part of this work are complete mRNA transcripts, or fragments of them, as shown in Figure A.2; determining what part, if any, of a given sequence is coding, what part is UTR (5’ or 3’), and what part is poly(A) tail is an important step in the analysis.

A.2 Sequencing Terminology

Sequencing is the process of discovering the sequence of nucleotides which makes up a DNA or RNA molecule.

complementary DNA a complementary DNA (cDNA) sequence is a double-stranded DNA sequence copied from an mRNA. One strand will reflect the exact sequence of the mRNA; the other will be the reverse complement of it. cDNAs are made from mRNAs during RNA sequencing, because the sequencing process requires a double-stranded molecule. Since this work is concerned entirely with RNA transcripts, we will refer to the sequencing of cDNA rather than DNA.
clone a cDNA molecule which is to be sequenced.

vector a vector is a circular DNA molecule (circular string) with a defined insert site, at which the molecule is broken and a cDNA clone is inserted. Once the clone is inserted, it is known as an insert.

library a library is, in this context, a population of clones, inserted into vector molecules and ready for sequencing.

adapter an adapter is a short DNA molecule used to link the ends of a cDNA molecule with the vector, when inserting the cDNA into the vector. Some library construction methods use adapters; others do not.

Given the above terminology, sequencing mRNA involves several stages:

1. extracting mRNA from the tissue,
2. converting the mRNAs into cDNAs,
3. inserting the cDNAs into vector molecules to make a cDNA library, usually truncating the poly(A) tail to approximately 20 nucleotides, and
4. sequencing clones chosen at random from the library.

The details of the laboratory processes in steps 1–3 are not relevant; the result is diagrammed in Figure A.3, showing the vector, the adapters, and the insert. Sequencing begins at a primer site, a specific DNA pattern which initiates the sequencing process, typically a short distance upstream of the insert site, and continues for several hundred nucleotides. The output of sequencing is a chromatogram. The format of a chromatogram is not relevant here; all that matters is that it is translated into a DNA sequence by base calling software.

Base calling is the translation of a chromatogram into a DNA sequence, as well as a quality sequence which records, for each nucleotide in the sequence, the likelihood that the base calling software gave the right answer. We use the widely-used Phred program (Ewing et al. 1998, Ewing & Green 1998) for base calling, using default parameters. For each nucleotide, Phred generates a quality value $q = -10 \log_{10} P_e$, where $P_e$ is the probability that the base call is incorrect. A quality value of 13 corresponds with a 5% probability of error; a value of 20 corresponds with a 1% probability of error.

A read is a DNA sequence resulting from a single sequencing operation on a clone. The read typically includes some vector, the leading adapter, and some or all of the insert, as can be seen in Figure A.3.
It is also possible to sequence from the other end of the insert, in which case the read will include the reverse complement of a few nucleotides of the trailing vector, the trailing adapter, and the end of the insert. If the insert is short enough, the forward and reverse reads will overlap (in reverse complement orientation to each other). Other possibilities exist. For example, the adapter sequence may incorporate a primer site, so that sequencing could begin in the adapter. In this case, the complete adapter would not be detectable in the resulting read, since sequencing would have started part way through it. As another example, many clones sequenced in the GRASP project were sequenced using a run of T nucleotides as the primer, with the intent of sequencing only those inserts which were inserted in 3' → 5' orientation in the insert, and which had a clear poly(A) tail (in this case a poly(T) head, since the sequences were oriented 3' → 5', i.e. were reverse complements of the original mRNA). The advantage is that fewer nucleotides of vector, adapter and poly(A) tail are included in the read, so the read incorporates more useful nucleotides than otherwise (since vector, adapter and poly(A) tail sequences do not provide meaningful information about an mRNA).

It is important to note that the orientation of the insert with respect to the vector is not necessarily 5' → 3' with respect to the original mRNA, but may be reversed, in which case sequencing from the primer site will produce the reverse complement of the true mRNA sequence.\footnote{Depending on the details of the cDNA library construction, most sequences may be 5' → 3', most may be 3' → 5', or there may be no particular bias. However, even when an attempt is made to insert the cDNA in a particular orientation, the desired orientation is achieved at most about 90–95% of the time, based on analysis of the cDNA libraries used in GRASP.} In addition, the insert is usually not a complete mRNA; it may be just the 5' end, just the 3' end, or a fragment from the middle of the original sequence. If the insert is short, both leading and trailing vector and adapter may be included in a single read.

A single read from a cDNA clone is called an expressed sequence tag or EST, because it indicates that the corresponding sequence of genomic DNA is expressed, or transcribed into RNA. Since vector, adapter, and poly(A) tail sequences should not be included in further processing, they are masked: replaced by X's in the sequence. Chapter 3 details the post-processing steps to detect and mask the vector, adapters and poly(A) tail in reads.

Given the reads and quality values from a set of chromatograms, sequence assembly is the process of discovering the nucleotide sequence of a DNA molecule from a collection of overlapping reads. Since technological limitations currently prevent us from directly sequencing more than a few hundred nucleotides of a DNA molecule at one time, the standard approach to sequencing long molecules is to sequence short overlapping fragments, then put the fragments together. Given sufficiently many reads from randomly chosen positions in the original sequence, all or most of the
Figure A.3: A sketch of a cDNA clone, including the vector (thin black line), adapters (medium black lines) and the insert (thick black line). The sequencing process reads the gray region: a small amount of the vector sequence, the adapter, and some or all of the insert. If the insert is relatively short, the sequence may include the trailing adapter and some trailing vector.
original sequence can be reconstructed. The reconstructed sequences, found by successively overlapping matching reads and assembled sub-sequences, are called contigs (from contiguous sequence). In the case of cDNA libraries, the problem is to assemble many mRNA transcripts from a set of reads, without prior information about which reads are part of the same transcript, or even how many transcripts there are. One complication which occurs in both regular sequence assembly and assembly of a set of mRNA transcripts is the presence in cDNA libraries of clones which are composed of fragments from two or more different mRNA transcripts. During the library construction process, it is possible for unrelated fragments of RNA to join end-to-end, as if they were one longer fragment. These clones are called chimeric clones or just chimeras. The assembly process, including the complications which arise as a result of the presence of chimeras, is described in detail in Chapter 4.

A.3 Sequence Alignment

Consider DNA sequences as strings over the alphabet $\Sigma = \{A, C, G, T\}$. $|S|$ denotes the length of a DNA sequence $S$. Characters of $S$ are numbered from 1 to $|S|$, indicated by subscripts: $S_i$ is the $i$'th character of $S$. Substrings of $S$ are indicated as ranges of subscripts: $S_{i..j}$ is the substring of $S$ from the $i$'th to the $j$'th character, inclusive. A gap is added to a sequence by inserting one or more gap characters “-”. An indel is a single gap position, i.e. a position which has a gap character in one sequence rather than a nucleotide. A gap is a maximal consecutive run of gap characters in a sequence. An alignment $A$ of two sequences $S$ and $T$ is a pair of sequences $S'$ and $T'$, formed from $S$ and $T$ by inserting gap characters in one or both until they are the same length. The $i$'th position in $A$ is a pair of characters $S'_i$ and $T'_i$.

A score function assigns a score to an alignment. For example, a simple function $F$ assigns a score to an alignment $A$ as follows:

$$F(A) = \sum_{i=1}^{|A|} \begin{cases} 
-2 & \text{if } S'_i = "-" \text{ or } T'_i = "-" \\
1 & \text{if } S'_i = T'_i \text{ and } S'_i \neq "-" \\
-1 & \text{otherwise}
\end{cases}$$

(A.1)

This is an example of a linear score function, so called because the score of a gap is a linear function of its length. Many variations on score functions exist. The popular affine model scores a gap of length $k$ with a function $g(k) = o + ek$, where $o$ is the score for beginning or “opening” a gap, and $e$ is the score for each position in the gap (alternatively, for extending the gap). Another variation uses a per-pair function $s(x, y)$ for characters $x$ and $y \in \Sigma$, rather than fixed match and mismatch scores. Many score models combine these two features.
The basic problem of alignment is to find an arrangement of gaps in the two input sequences which maximizes the score of the alignment, with respect to some relevant score model. “Relevant” in this case refers to the purpose of the alignment. Usually an alignment is intended to reflect an evolutionary relationship between sequences, so a score function which reflects the likelihood of evolutionary changes is appropriate. However, other cases exist. In Chapter 3, we describe a score function suitable for detecting poly(A) tails.

A **global** alignment is an alignment of two complete strings. Needleman & Wunsch (1970) described an algorithm for finding an optimal (highest scoring) global alignment of two strings. Detailed descriptions of this algorithm can be found in any introductory computational biology textbook, for example (Durbin et al. 1998, chapter 1) or (Gusfield 1997, chapter 11).

A **local** alignment is an alignment of substrings of two strings. Smith & Waterman (1980) gave an algorithm for local alignment, which finds a pair of substrings \( S_{i..j} \) and \( T_{k..l} \) from \( S \) and \( T \) and an optimal alignment of \( S_{i..j} \) and \( T_{k..l} \), such that no other pair of substrings from \( S \) and \( T \) has a higher-scoring optimal alignment. If there are multiple equally high-scoring points in the matrix, which is found depends on details of the implementation, or in some cases all equally high-scoring solutions might be given.

Both algorithms are described in Chapter 3.

### A.4 Evolution Terminology

Evolution occurs as changes to a species’ genome accumulate over time, and as species diverge into multiple distinct species. Various changes occur, from very small to very large:

**substitution** Individual nucleotides may be replaced by others. The replacement of a purine (A or G) with another purine, or a pyrimidine (C or T) with another pyrimidine, is called a *transition*. Any other substitution is called a *transversion*. This detail is of interest because transitions are more probable than transversions.

**insertion and deletion** Fragments of DNA ranging in size from one or two nucleotides up to millions of nucleotides may be added to or removed from the genome.

**reversal** Regions of DNA may be cut from the genome and re-inserted at the same spot, but in the opposite orientation.
translocation A region of DNA may be cut from one part of the genome and reinserted elsewhere.

duplication A region of DNA of almost any size may be copied and inserted elsewhere in the genome. In particular, a gene duplication is a duplication of a region of DNA containing a complete gene. A species’ genome may include several copies of a gene, which have diverged over time and taken on new or specialized functions; such groups of related genes are called gene families. Whole chromosomes or the entire genome may also be duplicated.

Changes to DNA are called mutations. In general, any given mutation occurs in one individual initially, then either

1. becomes lost because the individual does not reproduce, or not enough of its descendants transmit the mutation to their descendants, or

2. propagates through the population over many generations, until all members of the species carry it.

In the second case, the mutation is said to have become fixed in the population. The first case is always more likely than the second, but not all mutations have an equal probability of achieving fixation. Mutations may have no effect on the organism’s fitness for survival and reproduction, a negative effect, or a positive effect. In general, mutations with a positive effect on fitness have a higher likelihood of fixation; those with a negative effect on fitness a lower one. Mutations which have a strongly negative effect, such as making the organism infertile or dead, are subject to purifying selection: the mutation will never be passed on to offspring, and so can never become fixed in the population. Even somewhat bad mutations tend to suffer strong purifying selection. Most mutations are neutral; some are harmful; very few are beneficial.

Speciation occurs when a single species diverges via evolution into multiple species. Since species (almost) never merge again, the evolutionary history of species can be described as a tree: the well-known evolutionary tree. Speciation may occur because of geographical separation of populations, for a variety of reasons related to changes in the genome.

Species and named groups of species are referred to as taxa (singular taxon). Taxa at some levels have generic names: in order from smallest to largest, classification levels include species, genus, family, order, class, phylum, kingdom. There are many intermediate classification levels as well.
Taxa of interest

This section briefly describes some taxa that are relevant to this work.

*Salmonidae* This family includes all salmonid fishes, such as salmon, trout and whitefish.

*Esocidae* The pike family includes pike and related taxa such as pickerel, now thought to be the most closely related family to salmonids.

*Osmeridae* Smelt and related taxa, formerly thought to be the family most closely related to salmonids.

*Teleostei* This group incorporates the majority of the so-called “ray-finned” fishes, and includes most of the commonly-known fishes other than sharks. Teleosts are thought to have undergone a genome duplication shortly after branching off from the rest of the evolutionary tree.

*Danio rerio* The zebrafish is one of the best-studied of all fishes, with full genomic sequence available, as well as a large set of ESTs. It participated in the Teleost genome duplication, but has not undergone a duplication since then. Because so much sequence data is available, it makes a convenient test model for a species without a recent duplication.

*Xenopus* This genus includes *Xenopus laevis*, the African clawed frog, and *Xenopus tropicalis*, the Western clawed frog. These species are of interest because *X. laevis* has recently undergone a genome duplication relative to *X. tropicalis*, hence is a useful model on which to test theories of genome duplication.

*Vertebrata and Chordata* The vertebrates include all animals with a spinal cord. The larger group of chordates includes additional species which have a primitive “notocord”, the predecessor to a spinal cord. These taxa are of relevance to this work only because controversy exists regarding whether they underwent genome duplications or not.

Phylogenetic relationships among these taxa are shown in Figure 1.1.
Appendix B

Raw Data

This chapter lists the ranges of accession numbers, where available, of the raw EST traces used in this work. Note that the ranges do not include all accessions between the first and last; rather, they include all ESTs from GenBank’s Trace Archive for the given species, in that range. This is done simply to reduce the number of pages needed to describe these inputs.

For approximately 29,000 Rainbow trout sequences, accessions are not available since the sequences have not been submitted to GenBank. However, all raw data including unsubmitted sequences are available at:

http://web.uvic.ca/grasp/gdbrown/dissertation_data/
Atlantic salmon (*Salmo salar*)

<table>
<thead>
<tr>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJ424083 – AJ425876</td>
<td>CA767613 – CA770910</td>
<td>DR695341 – DR696698</td>
<td></td>
</tr>
<tr>
<td>AM041408 – AM042538</td>
<td>CB498694 – CB518126</td>
<td>DT317730 – DT317740</td>
<td></td>
</tr>
<tr>
<td>AM049439 – AM049915</td>
<td>CK873405 – CK900547</td>
<td>DV105941 – DV107615</td>
<td></td>
</tr>
<tr>
<td>AM083343 – AM083946</td>
<td>CK990213 – CK991215</td>
<td>DV644005 – DV644481</td>
<td></td>
</tr>
<tr>
<td>AM397478 – AM397622</td>
<td>CN181049 – CN181464</td>
<td>DV670507 – DV671022</td>
<td></td>
</tr>
<tr>
<td>BE518459 – BE518636</td>
<td>CN442486 – CN442560</td>
<td>DW005391 – DW007928</td>
<td></td>
</tr>
<tr>
<td>BF228468 – BF228712</td>
<td>CO469580 – CO472623</td>
<td>DW177413 – DW183431</td>
<td></td>
</tr>
<tr>
<td>BG933736 – BG936682</td>
<td>CV428771 – CV429168</td>
<td>DW338860 – DW340703</td>
<td></td>
</tr>
<tr>
<td>BI468016 – BI468193</td>
<td>CX153890 – CX154306</td>
<td>DW468731 – DW473883</td>
<td></td>
</tr>
<tr>
<td>BI544051 – BI544217</td>
<td>CX281915 – CX282499</td>
<td>DW531365 – DW584513</td>
<td></td>
</tr>
<tr>
<td>BM413686 – BM414535</td>
<td>CX352775 – CX358392</td>
<td>DW588064 – DW592397</td>
<td></td>
</tr>
<tr>
<td>BM446607 – BM446608</td>
<td>CX727320 – CX727664</td>
<td>DY467526 – DY467913</td>
<td></td>
</tr>
<tr>
<td>BQ035314 – BQ037059</td>
<td>CX984300 – CX984374</td>
<td>DY691308 – DY741432</td>
<td></td>
</tr>
<tr>
<td>BQ630874 – BQ630874</td>
<td>DN047365 – DN048367</td>
<td>DY802071 – DY802200</td>
<td></td>
</tr>
<tr>
<td>BU693964 – BU694199</td>
<td>DN138771 – DN140628</td>
<td>EG755023 – EG943493</td>
<td></td>
</tr>
<tr>
<td>BU808708 – BU808708</td>
<td>DN162055 – DN167138</td>
<td>FC072704 – FC072888</td>
<td></td>
</tr>
<tr>
<td>BU965588 – BU965906</td>
<td>DN550291 – DN550634</td>
<td>U079963 – U079971</td>
<td></td>
</tr>
<tr>
<td>CA036414 – CA064598</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Zebrafish (Danio rerio)**

<table>
<thead>
<tr>
<th>Accession Numbers</th>
<th>Start Date</th>
<th>End Date</th>
<th>Accession Numbers</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA494559 – AA495494</td>
<td>AW170810 – AW171628</td>
<td>CK138264 – CK144292</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA497144 – AA497402</td>
<td>AW173804 – AW175576</td>
<td>CK145448 – CK148642</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA542435 – AA542678</td>
<td>AW184039 – AW184771</td>
<td>CK235961 – CK240389</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA545709 – AA545724</td>
<td>AW202563 – AW203173</td>
<td>CK352167 – CK357381</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA605608 – AA606234</td>
<td>AW231973 – AW233766</td>
<td>CK359975 – CK363563</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA658585 – AA658812</td>
<td>AW279609 – AW282147</td>
<td>CK394854 – CK401113</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI330328 – AI332297</td>
<td>AW305386 – AW306107</td>
<td>CK666569 – CK700712</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI384137 – AI387493</td>
<td>AW342597 – AW344345</td>
<td>CK703311 – CK704301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI395665 – AI397484</td>
<td>AW419474 – AW423272</td>
<td>CK705759 – CK706131</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI415743 – AI416403</td>
<td>AW466377 – AW466864</td>
<td>CK707576 – CK714742</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI436854 – AI439493</td>
<td>AW500778 – AW501032</td>
<td>CK864382 – CK873404</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI444118 – AI444540</td>
<td>AW566517 – AW567569</td>
<td>CN013347 – CN024941</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI461280 – AI461395</td>
<td>AW594754 – AW595856</td>
<td>CN167208 – CN180764</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI476823 – AI478024</td>
<td>AW777210 – AW778723</td>
<td>CN315426 – CN331900</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI522337 – AI522810</td>
<td>AW826183 – AW827062</td>
<td>CN500080 – CN513464</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI558267 – AI558895</td>
<td>BE015622 – BE017947</td>
<td>CN831330 – CN844104</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI584192 – AI585238</td>
<td>BE200550 – BE202257</td>
<td>CO245082 – CO250218</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI588088 – AI588836</td>
<td>BE556693 – BE558190</td>
<td>CO390363 – CO390366</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI601277 – AI601868</td>
<td>BE605258 – BE606174</td>
<td>CO392748 – CO406224</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI626134 – AI626875</td>
<td>BF156051 – BF158632</td>
<td>CO794910 – CO815827</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI629052 – AI629398</td>
<td>BF717268 – BF718230</td>
<td>CO913421 – CO936964</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI641018 – AI641780</td>
<td>BF937050 – BF938948</td>
<td>CO956969 – CO960794</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI668667 – AI676790</td>
<td>CA469634 – CA477797</td>
<td>CV072870 – CV081151</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI721287 – AI723335</td>
<td>CA480561 – CA480638</td>
<td>CV102363 – CV121766</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI793345 – AI794648</td>
<td>CA495796 – CA496800</td>
<td>CV126123 – CV127283</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI882672 – AI884302</td>
<td>CA787299 – CA787661</td>
<td>CV480209 – CV491838</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI931001 – AI931129</td>
<td>CA974743 – CA975855</td>
<td>CV576051 – CV576845</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI942493 – AI943405</td>
<td>CD750680 – CD760424</td>
<td>DN596774 – DN601413</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI957385 – AI959771</td>
<td>CD766660 – CD766720</td>
<td>DN764125 – DN772611</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI964842 – AI965327</td>
<td>CF243194 – CF243278</td>
<td>DN799483 – DN799814</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI974130 – AI974210</td>
<td>CF265213 – CF266490</td>
<td>DNS32278 – DNS36863</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI979342 – AI979376</td>
<td>CF347086 – CF348806</td>
<td>DR714146 – DR730530</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW018432 – AW019868</td>
<td>CF549558 – CF550675</td>
<td>DT054337 – DT082442</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW058679 – AW059511</td>
<td>CF594364 – CF595418</td>
<td>DT317743 – DT319101</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW076504 – AW078464</td>
<td>CF995861 – CF999999</td>
<td>DT862496 – DT883563</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW115513 – AW117172</td>
<td>CK000001 – CK000807</td>
<td>DV584231 – DV600733</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW127719 – AW128833</td>
<td>CK015385 – CK019446</td>
<td>DY544190 – DY576214</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW133501 – AW134404</td>
<td>CK024236 – CK030496</td>
<td>EE299434 – EE331820</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW165023 – AW165432</td>
<td>CK126185 – CK127952</td>
<td>EE684204 – EE695310</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
African clawed frog (*Xenopus laevis*)

<table>
<thead>
<tr>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>AW147173 – AW148291</td>
<td>BQ723779 – BQ737634</td>
<td>CF270174 – CF271888</td>
</tr>
<tr>
<td>AW164864 – AW165022</td>
<td>BQ884353 – BQ884431</td>
<td>CF282467 – CF290820</td>
</tr>
<tr>
<td>AW198612 – AW200663</td>
<td>BQ885923 – BQ886036</td>
<td>CF341986 – CF342435</td>
</tr>
<tr>
<td>AW460243 – AW460989</td>
<td>BQ897323 – BQ897491</td>
<td>CF519315 – CF522766</td>
</tr>
<tr>
<td>AW570626 – AW570642</td>
<td>BQ900459 – BQ900810</td>
<td>CF546553 – CF549557</td>
</tr>
<tr>
<td>AW595857 – AW595915</td>
<td>BU899326 – BU902015</td>
<td>CK363564 – CK363567</td>
</tr>
<tr>
<td>AW762295 – AW783940</td>
<td>BU903031 – BU917048</td>
<td>CK367859 – CK367863</td>
</tr>
<tr>
<td>AW871698 – AW872331</td>
<td>CA787662 – CA794201</td>
<td>CK740333 – CK743195</td>
</tr>
<tr>
<td>BE025536 – BE027132</td>
<td>CA971157 – CA974742</td>
<td>CK796493 – CK800757</td>
</tr>
<tr>
<td>BE131692 – BE132413</td>
<td>CA980748 – CA983640</td>
<td>CK803267 – CK807870</td>
</tr>
<tr>
<td>BE188655 – BE190144</td>
<td>CA984701 – CA988641</td>
<td>CN315439 – CN31905</td>
</tr>
<tr>
<td>BE191742 – BE192420</td>
<td>CB179912 – CB180945</td>
<td>CO382191 – CO390362</td>
</tr>
<tr>
<td>BE490830 – BE491740</td>
<td>CB185102 – CB185107</td>
<td>CO553867 – CO554655</td>
</tr>
<tr>
<td>BE505157 – BE509591</td>
<td>CB196813 – CB201494</td>
<td>CV072865 – CV081164</td>
</tr>
<tr>
<td>BE575320 – BE576803</td>
<td>CB205879 – CB209720</td>
<td>CV102381 – CV121752</td>
</tr>
<tr>
<td>BE669150 – BE669413</td>
<td>CB558150 – CB565877</td>
<td>CX129928 – CX135761</td>
</tr>
<tr>
<td>BE677954 – BE681493</td>
<td>CB590939 – CB593345</td>
<td>DR714412 – DR730528</td>
</tr>
<tr>
<td>BF024719 – BF025649</td>
<td>CB755779 – CB756884</td>
<td>DT054336 – DT082441</td>
</tr>
<tr>
<td>BF047067 – BF049564</td>
<td>CB941050 – CB945391</td>
<td>DY544192 – DY576216</td>
</tr>
<tr>
<td>BF071739 – BF072407</td>
<td>CB983178 – CB984346</td>
<td>EB461409 – EB484066</td>
</tr>
<tr>
<td>BF231535 – BF232617</td>
<td>CB985962 – CB985965</td>
<td>EB643659 – EB647671</td>
</tr>
<tr>
<td>BF426221 – BF428512</td>
<td>CD085815 – CD101655</td>
<td>EB727227 – EB739623</td>
</tr>
<tr>
<td>BF610635 – BF616026</td>
<td>CD252635 – CD257086</td>
<td>EB800794 – EB801413</td>
</tr>
<tr>
<td>BG016127 – BG023719</td>
<td>CD300791 – CD306335</td>
<td>EC275258 – EC277773</td>
</tr>
<tr>
<td>BG037214 – BG038977</td>
<td>CD324639 – CD330177</td>
<td>EE296939 – EE297447</td>
</tr>
<tr>
<td>BG159741 – BG163215</td>
<td>CD360092 – CD363323</td>
<td>EE299460 – EE331817</td>
</tr>
<tr>
<td>BG232839 – BG235069</td>
<td>CD519688 – CD520062</td>
<td>EG565451 – EG587992</td>
</tr>
<tr>
<td>BG264120 – BG264814</td>
<td>CD811081 – CD811472</td>
<td></td>
</tr>
</tbody>
</table>

Rainbow trout (*Oncorhynchus mykiss*)

<table>
<thead>
<tr>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
<th>Accession Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>BX072585 – BX088607</td>
<td>CB485850 – CB498693</td>
<td>CN442492 – CN442499</td>
</tr>
<tr>
<td>BX295261 – BX295518</td>
<td>CK991216 – CK991358</td>
<td>CX351194 – CX351507</td>
</tr>
<tr>
<td>BX296569 – BX321855</td>
<td>CN181470 – CN181471</td>
<td>CX662670 – CX662670</td>
</tr>
<tr>
<td>CA341564 – CA388587</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lake whitefish (*Coregonus clupeaformis*)

CB483540 – CB484653  CX349194 – CX349770  EV365920 – EV370540
CN442488 – CN442488  EV249701 – EV253700