Structure-based Realignment of Non-coding RNAs in Multiple Whole Genome Alignments.

by

Michael Ku Yu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author .....
Department of Electrical Engineering and Computer Science
May 20, 2011

Certified by .....
Bonnie Berger
Professor of Applied Mathematics and Computer Science
Thesis Supervisor

Accepted by .....
Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Structure-based Realignment of Non-coding RNAs in Multiple Whole Genome Alignments

by

Michael Ku Yu

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2011, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Computer Science and Engineering

Abstract

Whole genome alignments have become a central tool in biological sequence analysis. A major application is the *de novo* prediction of non-coding RNAs (ncRNAs) from structural conservation visible in the alignment. However, current methods for constructing genome alignments do so by explicitly optimizing for sequence similarity but not structural similarity. Therefore, *de novo* prediction of ncRNAs with high structural but low sequence conservation is intrinsically challenging in a genome alignment because the conservation signal is typically hidden. This study addresses this problem with a method for genome-wide realignment of potential ncRNAs according to structural similarity. Doing so reveals thousands of new high-confidence ncRNA predictions with particularly low sequence conservation from an alignment of 12 *Drosophila* genomes and hundreds from an alignment of 28 vertebrate genomes in the Encode project.

Thesis Supervisor: Bonnie Berger
Title: Professor of Applied Mathematics and Computer Science

Acknowledgments

Isaac Newton is often quoted for saying "If I have seen further it is only by standing on the shoulder of giants". While the "giants" typically refer to intellectual predecessors, usually with no personal relations to the one quoting, I interpret the "giants" to be metaphors for all of the individuals upon whose personal support I've been able to mature and learn.

This thesis represents my largest single academic project so far, and likewise this is the first significant "Acknowledgemets" piece I've had to write. Executing this thesis required not only particular familiarity in computational biology but also the maturity to follow a long project from start to finish. I acquired the latter over a lifetime of influential experiences from mentors, friends, and family members. Therefore, I will take advantage of this opportunity to make up for the lack of due credit I should have given throughout my life to the folks who have not only made possible this thesis but have shaped who I am today.

I thank my supervisor Bonnie Berger for having me in her group for almost four years now since the time of my arrival in the summer after my freshman year at MIT. By having me in group meetings, seminar talks, and fostering interactions with other members of the group, she's given me a strong foundation in computational biology for my future career.

I thank Sebastian Will, a postdoctoral fellow in Bonnie's group, an RNA expert, and a seasoned bioinformatician, who supervised me through the work presented in this thesis. We spent countless hours thoroughly discussing every idea presented in this thesis to painstaking detail. I appreciate his excellent mentorship and guidance during our exploration of misaligned non-coding RNAs in whole genome alignments.

I thank Michael Baym, my first mentor in Bonnie's group and in computational biology, for teaching me so much about research as well as life. He guided me through my first project in computational biology where we investigated compensatory mutations driven by protein-protein interactions. The ideas and problems we encountered here eventually led us to problems with current whole genome alignments. He also

taught me the basics of probability theory, to be confident in my own abilities, to appreciate traveling, and, most importantly, to enjoy life. His influence on me can only be understated here.

I thank everyone I've gotten to know in Bonnie's group over the years. I give special thanks to Patrice Macaluso for being an awesome and caring adminstrative assistant. It has been my pleasure to have gotten to know Allen Bryan, Leonid Chindelevitch, Eric Eisner, Raghavendra Hosur, Luke Hutchison, Irene Kaplow, Alex Levin, Po-Ru Loh, Andrew McDonnell, Oaz Nir, Charlie O'Donnell, Nathan Palmer, Daniel Park, Vinay Pulim, Patrick Schmid, Michael Schnall-Levin, Rohit Singh, Jason Trigg, George Tucker, Jerome Waldispuhl, and Shannon Wieland.

I thank Anke van Zuylen and Frans Schalekamp for being the most hospitable supervisors I could have asked for when I spent my summer of 2009 doing research at the Institute of Theoretical Computer Science at Tsinghua University, Beijing. Not only were they so kind and patient with my work, but they also took me around Beijing. Because of them, I always become nostalgic of this summer whenever I see a picture or hear a story about modern Beijing.

Professor Jonathan (Steve) Alexander's molecular and cellular biology lab at LSUHSC-S was the first research group I was a part of. It was a really exciting time for me, the first time I could tackle uncharted territory in science. I thank Steve for giving me the opportunity to work in his lab. Even though I had no prior experience, he treated me with so much respect as if we were intellectual equals. I thank Merilyn Jennings for teaching me so many essential protocols and lab practices. I thank Cidgem (Ci Ci) Yilmaz for being an always cheerful officemate and introducing me to the beauty of classical guitar music.

I am very greatful for my friends Albert Chang, Ana Chen, David Chen, Nadia Elkordy, Corinna Hui, Nasly Jimenez, Irene Kaplow, Itamar Kimchi, Jeremy Lai, Michele Lee, Grace Li, Victoria Lo, Kevin Luu, Daniel Mokrauer-Madden, Karl Rieb, Sharon Tam, Jason Trigg, Albert Wang, Jeff Xing, and Di Ye for their support over the years as we lived in the same or near floors of Next House as MIT undergraduates. I don't regret a single time we just hung out together, even though most of the time

we were drowning in work to be done, because years from now these will become my fondest memories of my MIT. I hope to be blessed with their continued friendships.

I thank my closest friends from high school: Vikram Agarwal, Charlotte Gates, Bennett Hailey, Claire Kendig, Sophia Kostelanetz, Austin Roger, Erin Smith, and Kevin Wilkes. Without them, I would not have survived the emotional ups and downs that normally come during teenage years.

Lastly, I would not have had the fortune to be at MIT, let alone be doing this M.Eng., if it were not for my parents and two older brothers, Christopher and Robinson. I never show enough appreciation for them, but they are family. If everyone else has left me, they will still be there standing by me.

5/20/2011

Michael Ku Yu

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rise of high-throughput sequencing technologies in the past decade has marched in a new era in biology characterized by a wealth of raw sequence and expression analysis data. Not surprisingly, increasing attention is being given to new methods to decipher this sea of information. Currently at hand are the complete genomes of hundreds of species, and the number is expected to climb exponentially in time. One of the most widely used tools for analyzing them is the sequence alignment of multiple whole genomes. This thesis addresses the current challenges of constructing whole genome alignments and focuses on improving the structural alignment of non-coding RNAs.

1.1 Sequence Alignment

The concept of a sequence alignment is among the oldest and important in the history of computational biology. Simply put, a sequence alignment is a matching of the individual characters in two or more strings to identify shared biology represented by the strings. In general the strings are amino acid sequences of proteins or nucleic acid sequences of DNA or RNA. Sequence alignments can be constructed so as to reflect evolutionary conservation and divergence among the sequences or to establish functional similarities such as sequence patterns that give rise to molecular structure or targeted binding sites. For example, the strings may represent the DNA sequences

of genes that have similar functions in closely related species, and the genes could be aligned so as to indicate which subregions of the genes descended from a common ancestral gene and which are mutations unique to a species.

Algorithmically, typical methods for constructing alignments will optimize objective functions. In sequence-similarity based methods, the objective function reflects evolutionary conservation on the sequence level by favoring the alignment of identical or similar characters. Much progress in sequence-similarity methods has been made in the past two decades through tools like Muscle [14], Clustalw [29], T-Coffee [21], and Probcons [12], to name only a handful, that can align multiple short sequences in seconds.

In structural-similarity based methods, the objective function reflects conservation of molecular structure and favors the alignment of positions that may not necessarily contain similar characters but do give rise to similar structures. For example, consider RNA secondary structure where the building block of structure is the base pairing of nucleotides either through Watson-Crick pairs ("A" with "U", or "C" with "G") or wobble pairs ("G" with "U"). Suppose that in a given RNA, two positions with nucleotides "A" and "U" form a base pair. Suppose further that in an evolutionarily related RNA, the two positions have mutated simultaneously into the nucleotides "C" and "G", respectively, but still remain a base pair. A structural-similarity method aligning these two RNAs may favor aligning "A" to "C" and "U" to "G" to match base pairs even though the characters do not match.

Alignment methods can be both structural and sequence-similarity based. This combination is clearly seen from the "simultaneous sequence alignment and folding" algorithm of Sankoff [27] for aligning RNAs. Under the right parameters, the Sankoff algorithm can detect conservation on both the sequence and structural level. It is therefore regarded as more accurate than "sequence" methods that rely only on sequence-similarity based and "folding" methods that rely only on structural-similarity. However, the original Sankoff algorithm runs in $O(n^6)$ time and is infeasible for large sequences. Variations of Sankoff's approach of "simultaneous alignment and folding" has been strongly improved during the last years, e.g. by the LocARNA

tool [35], making large scale structural alignment of RNAs possible.

When sequence identity is high, sequence-similarity methods will accurately align structural elements in sequences. However, the inverse is not necessarily true. Under low sequence identity, sequence-similarity methods cannot identify compensatory mutations, i.e. character mutations which change the sequence but preserve structure. In the most recent Bralibase study [36], a benchmark to assess methods for aligning structural RNAs, the performance of sequence-similarity methods broke down when the sequence identity of RNAs to be aligned was below 60%. On the other hand, structural-similarity methods can still accurately align these RNAs.

1.2 Whole Genome Alignments

The sequence alignment of multiple whole genomes, more compactly known as a "whole genome alignment" (WGA), has become an essential tool for comparative genomics. Genome alignments have a number of biological applications. Aligned regions showing a high degree of conservation is the basis of prediction programs for functional genes and non-coding RNAs. Genome alignments are also used for inferring evolutionary history such as phylogeny and rates of evolutionary processes.

1.2.1 Challenges in construction

Aligning genomes, due to their size and heterogenous make-up, is more challenging computationally than aligning short sequences such as individual genes. A major challenge is the sheer size of genomes because it prevents direct application of commonly used tools for short sequences. Another source of difficulty is biological complexity. A genome is a heterogenous composition of discrete regions, each of which may encode different biological functions and undergo different evolutionary pressures. Delineating regions is a difficult task in itself, let alone adapting the alignment method according to special characteristics of the region. Assessing genome alignment methods is also an open problem because there does not exist any established benchmark or accepted genome alignment on which to train or compare against.

1.2.2 Current methods

Standard construction of genome alignments follow two stages. First, a large-scale syntenic map is made. Because of genome rearrangements events, in the form of duplications and transpositions, two genomes cannot be aligned simply by inserting gaps in both. Instead, a genome must be sliced beforehand into smaller subsequences, each of which is matched with slices from other genomes to form a syntenic block. Second, each block is individually aligned.

First generation whole genome alignment tools, like Enredo/Pecan [23], Mercator/Mavid [6, 11], MULTIZ [5], and MLAGAN [7] have become widely used. The continued use of multiple genome alignments suggests that it is still an open problem without consensus. Recent studies have suggested that indeed there is still room for improvement. For example, Prakash and Tompa [25] identified "suspicious regions", composing 9.7% of a MULTIZ alignment of human chromosome 1 to 17 vertebrates, where in each suspicious region at least one sequence appeared to be forced into alignment with unrelated sequences. Chen and Tompa [8] also found a significant presence of such regions in four different alignments (Pecan, Mavid, MULTIZ, and MLAGAN) of the same Encode regions [19] to 28 verterbrates genomes.

A limitation of these current tools is the sole use of sequence-similarity alignment models. As remarked in Section 1.1, such models inevitably lead to misalignments of sequences with low enough identity. In intergenic regions and introns which are regions known to have low sequence conservation, Chen and Tompa [8] also found greater disagreement among the four alignments than in exons which are known to have higher sequence conservation.

1.3 *De novo* prediction of non-coding RNAs

Traditional dogma in molecular biology has been that protein synthesis is the most prominent code stored in a genome. This has been uprooted in recent decades with the discovery of an extensive presence of functional transcripts that do not code for proteins. Collectively known as non-coding RNAs (ncRNAs), these transcripts

include, but are not limited to, miRNAs, riboswitches, snoRNAs, tRNAs, and RNase Ps. Interest in ncRNAs has given rise to the development of *de novo* prediction of ncRNAs genes from genomic sequences. Among the most widely used tools are RNAz [3, 26, 33, 32], EvoFold [33, 24], and CMFinder [30, 34]. These tools predict ncRNAs based on the observation that many classes of functional non-coding RNAs, but not all, exhibit conservation of secondary structure.

RNAz and EvoFold rely on accurate input alignment to detect structural conservation. Thus, they cannot predict ncRNAs that are sufficiently misaligned. CMFinder is not constrained to an input alignment, however it is much slower than RNAz and EvoFold. In order to run CMFinder on an alignment of the Encode regions, Torarinsson et al. [30] applied a number of heuristics, such as a limitation to two stems, that simplified the possible structures that could be found. The heuristics notwithstanding, the Encode alignment is only composed of 1% of the human genome and is much smaller than some genome alignments that RNAz [26] and EvoFold [24] have been applied to. For an excellent review of *de novo* ncRNA prediction, see [17].

1.4 Motivation and aims

This thesis addresses two intertwined problems discussed earlier: 1.) Current genome alignments are constructed with only sequence-similarity methods, leading to potential misalignment on a structural level, and 2.) efficient *de novo* RNA prediction tools depend on alignment accuracy on a structural level. To solve these two problems simultaneously, we present a general method for realigning ncRNAs in a whole genome alignment according to both sequence and structural-similarity. The goal of realignment is two-fold. First, latent ncRNAs which were previously undetected due to misalignment can now be predicted. Second, the whole genome alignment can be improved by patching it according to the ncRNA realignments.

Uncovering previously unknown ncRNAs through realignment, however, can be a circular cat-and-mouse problem. If a ncRNA is not yet known, how can it be targeted in genomes for realignment? Morever, using a structural-similarity model to realign

motifs that are not ncRNA and whose molecular structure is not important could worsen the already existing alignment. We workaround this problem as follows. After realigning a site from an existing genome alignment, we apply a *de novo* predictor to decide whether there is an ncRNA. If so, then we keep the realignment, and otherwise, we keep the original alignment.

The relatively expensive computation of structural-similarity methods than those that only model sequence-similarity has hitherto prevented applying them on a genome-wide level. In Chapter and 2 and 3, we present a computational pipeline that solves this problem by incorporating two novel features, a stability filter and constrained realignment. In Chapter 4, we demonstrate the effectiveness of this pipeline on the alignments of 12 Fly genomes and an Encode alignment.

Chapter 2

A pipeline for genome-wide realignment of structural non-coding RNAs.

In this chapter, we present a novel pipeline that takes as input a whole genome alignment (WGA) and realigns potential structural non-coding RNAs according to sequence and structural similarity. The end product is a set of ncRNA predictions and an improved WGA alignment formed by replacing the original alignments of the ncRNAs with their realignments. The pipeline can be applied to patch any WGA and does not rely on external genome annotations.

As an overview, there are three main stages of the pipeline. First, a stability filter reduces the computational time of the pipeline by filtering only for sites in the WGA where individual sequences have sufficient structural stability in order to form functional RNAs. An important property of the filter is that it does not depend on local alignment quality and is therefore robust against local inaccuracies in the WGA. Second, the RNAz tool is used to make *de novo* ncRNA predictions based on the original WGA and realignments. Finally, the WGA is patched.

2.1 The pipeline step-by-step

The procedure is outlined in Figure 2-1. In the following description of the pipeline, we number the single steps to correspond with the figure.

The pipeline processes a whole genome alignment (WGA), which consists of a set of colinear segments, to identify potential sites for non-coding RNA. We follow the RNAz screens of [26] in the way we slice the segments into windows and merge overlapping windows into loci. In Step (1), each segment is sliced along both strand orientations into windows of length 120 alignment columns at every 40 columns. For each window, sequences are removed from the window if they contain more than 25% gaps, more than 25% GC or AU content, or contain a large number of masked characters. Only window with at least two sequences remaining are kept. We distinguish between windows that span the same columns but are read in opposite directions. For this step, we employ the script `rnazwindows.pl` from the RNAz 2.0 package [3]. In deviation from its default settings, we run the script without an explicit reference sequence, by setting the –no-reference option, and do not set limits on the minimum sequence identity, by assigning 0 to the –min-id option.

In Step (2), we filter windows for thermodynamic stability in the individual sequences. Thermodynamic stability is the major alignment-independent feature for the decision of de-novo ncRNA predictors. Note that unlike the RNAz p score, which [26] used to filter the windows, the stability of individual sequences does not depend on details of the alignment. Filtering by stability, therefore, retains misaligned potential non-coding RNAs for further analysis. For each window, we assess its stability by applying RNAz-2.0 [3] to estimate the average z-score of the minimum free energy of the sequences. Only windows with an average z-score below a fixed stability threshold $\theta_{\text{stability}}$ are kept. Therefore, filtering by stability removes many windows unlikely to contain ncRNAs because of insufficient stability.

In step (3) windows are merged and reassembled into continuous alignment blocks referred to as "loci". In a WGA with sequences numbered $1, \ldots, k$, two windows A and B containing sequences $S_A, S_B \in \{1, \ldots, k\}$ are merged into a locus if they are

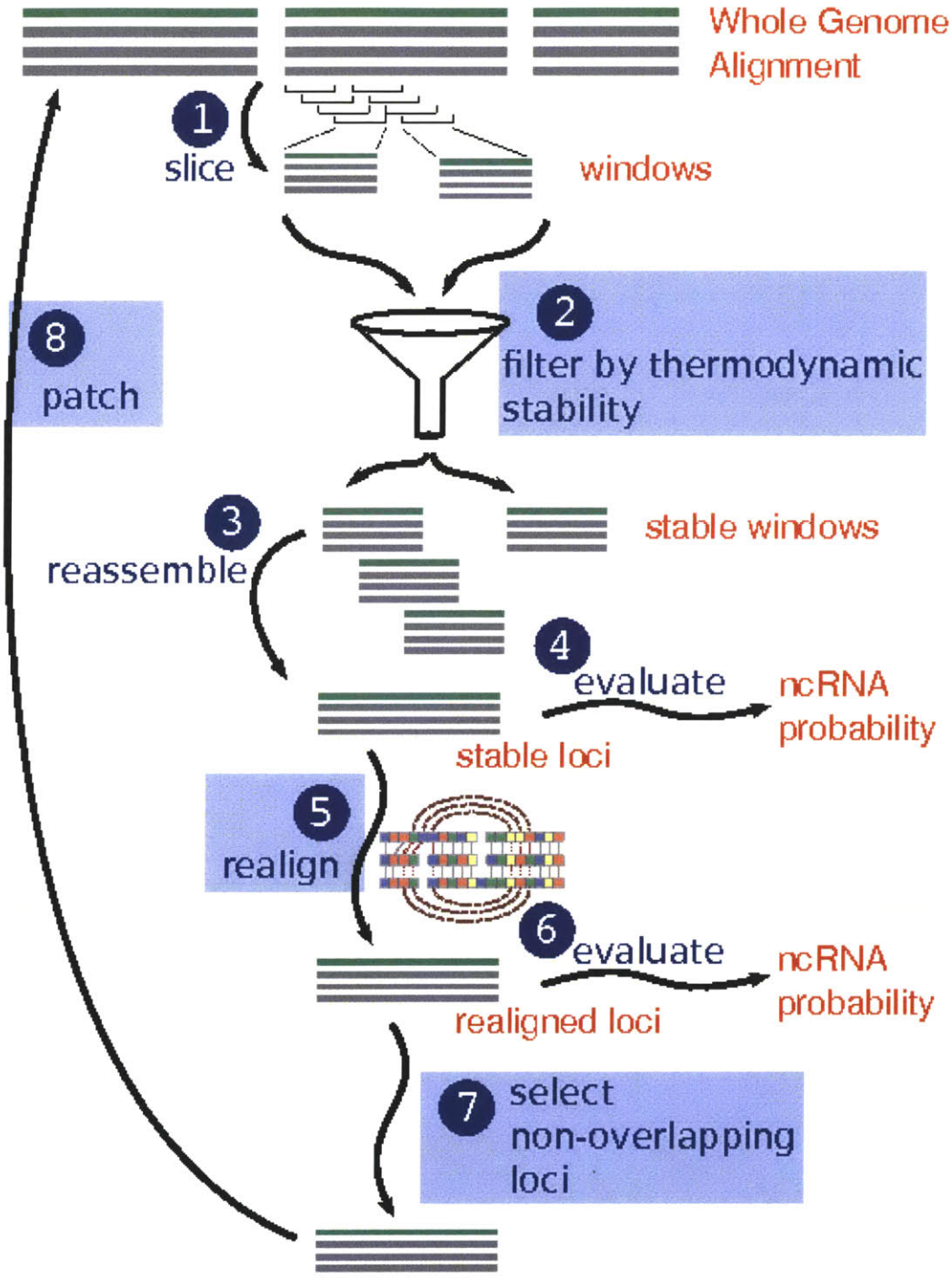


Figure 2-1: The realignment pipeline. Components of the pipeline share similarities to previous RNAz screens. We highlight our new, essential contributions by blue boxes. 1) The whole genome alignment is sliced into equally spaced windows of length 120 nt. 2) Potential sites for structural RNA are selected by filtering the windows by their thermodynamic stability. 3) Stable windows that overlap are assembled into loci. 4) The loci are evaluated for their likelihood of representing non-coding RNA. 5) Loci are realigned according to sequence and structural-similarity. 6) The realigned loci are evaluated for their non-coding RNA potential again. 7.) A non-overlapping subset of loci is selected. 8) For each realigned locus, patch the WGA depending on evaluations (4) and (6).

located in the same colinear segment, are oriented in the same direction, and overlap in one of the following two ways

1. A and B overlap at 80 alignment columns.

2. A and B overlap at 40 alignment columns or are positioned side-by-side. In this case, A and B must share a sufficient number of species defined by the condition $|S_A \cap S_B|/|S_A \cup S_B| \geq 0.5$.

A set of more than two windows is merged all together into one locus if there is a transitive chain of merging, i.e. A and B merge, B and C merge, etc. The locus is formed by re-slicing from the WGA the alignment block spanned by the windows and keeping sequences which appeared in one of the windows.

In step (4), we realign each locus, optimizing for sequence and structural similarity. In order to introduce only limited changes to the original alignment and maintain efficiency of the whole approach, we limit the deviation from the original alignment in a specified way. For this purpose, we apply a novel extension of the LocARNA multiple RNA alignment algorithm, which is introduced in Chapter 3. Before and after realignment, in Steps (4) and (6), we assess the likelihood that the locus contains structural non-coding RNA by de-novo ncRNA prediction. In our study, we estimate this likelihood by RNAz 2.0, i.e. we slice the locus into windows as above and determine the maximal RNAz ncRNA class probability P of the constituent windows. Before realignment, RNAz is applied in its non-structural alignment model, which is trained for sequence alignments produced by Clustal [29]. After realignment, we evaluate with RNAz in its structural alignment model [3] of RNAz, which was recently trained on the structural alignments of the LocARNA variant LocARNATE [22]. Applying RNAz after realignment can reveal potential non-coding RNAs that were misaligned in the whole genome alignment.

Since windows and hence loci were distinguished by strand orientation in earlier steps, some loci may be overlapping in genomic position. In step (7), we select a subset of the loci that are non-overlapping in position. We greedily consider adding loci one at a time to a non-overlapping set according to the maximum p score of a

locus before and after realignment. A locus is added to the set only if no other loci already in the set overlaps with it in position.

Finally, we propose to patch back locus alignments in step (8), replacing the original alignments of the loci with realignments. We use the evaluation of a realigned locus from earlier steps to decide where to patch the locus back. In this way, we can control the patching process and minimize the patching of loci that are not structural ncRNAs.

Chapter 3

Constrained realignment

In this chapter, we present a general algorithmic framework for realigning loci in step (5) of the pipeline. In this framework, we constrain the realignment of a locus such that it does not deviate too much from the original WGA. Doing so is faster than aligning the locus from scratch. Combined with the speedup of the stability filter in step (2), this framework allows the pipeline to be feasible for running over a whole genome alignment.

First, we motivate the idea of constrained realignment. One possibility for realigning a locus is to simply disregard the original WGA and align the locus completely from scratch. However, this would not take advantage of the WGA which, even if not perfect, may still provide useful information about the desired correct alignment. Some sequence positions of the locus may already be aligned correctly or almost correctly in the WGA. Instead, consider using the WGA as a reference and constraining the realignment of a locus around it.

Formally, we define this constraint with the notion of Δ-deviation from a reference alignment. We show how constraining the realignment provides a complexity speedup, how to apply it to any dynamic programming-based alignment algorithm, and finally its particular implementation in LocARNA [22], an efficient algorithm for simultaneous sequence and structural alignment of RNAs.

3.1 Preliminaries and terminology

A *sequence* S is a word of fixed alphabet Σ. We reserve a special symbol $- \notin \Sigma$ called *gap*. A word T of alphabet $\Sigma \cup \{-\}$ is called *gapped sequence*. We write $T_{|-}$ for the sequence that is obtained by removing all gap symbols from T. The *length of a word* w is denoted by $|w|$ and its *i-th character* by $w[i]$.

A *q-wise (multiple) alignment $\mathcal{A}$ of length m* $=: |\mathcal{A}|$ is a matrix $\mathcal{A} \subseteq (\Sigma \cup \{-\})^{q \times m}$ with $q =: \text{rows}(A)$ rows. We denote the x-th row of $\mathcal{A}$ by A_x and associate it with the gapped sequence $A_x[1] \dots A_x[m]$. A q-wise alignment $\mathcal{A}$ is an alignment of the sequences $S_1, \dots, S_q$ if and only if

1. for all $1 \leq x \leq q$: $S_x = A_{x|-}$.

2. no column in $\mathcal{A}$ is *gap-only*, i.e. consists only of symbols $-$.

For a q-wise alignment $\mathcal{A}$ and a p-tuple $\mathcal{I} = (x_1, \dots, x_p)$ of distinct integers in $\{1, \dots, q\}$, the *projection $\mathcal{A}\langle\mathcal{I}\rangle$ of $\mathcal{A}$ onto $\mathcal{I}$* corresponds to the sub-alignment implied by $\mathcal{A}$ on the sequences indexed by $\mathcal{I}$. It is constructed by taking the matrix $(A_{x_1} \dots A_{x_p})^{\mathrm{T}}$ and deleting all gap-only columns. A q-wise alignment $\mathcal{A}$ is an alignment of $\mathcal{A}^1$ and $\mathcal{A}^2$ if and only if $\mathcal{A}^1 = \mathcal{A}\langle(1, \dots, q_1)\rangle$ and $\mathcal{A}^2 = \mathcal{A}\langle(q_1 + 1, \dots, q)\rangle$ for some $1 \leq q_1 < q$.

A *(pairwise) cut* is a vector $c \in \mathbb{N}^2$. For a pairwise alignment $\mathcal{A}$, a cut $c = (c_1, c_2)$ is called *cut of $\mathcal{A}$ at column i* ($0 \leq i \leq |\mathcal{A}|$) iff $|(A_x[1] \dots A_x[i])_{|-}| = c_x$ for $x = 1, 2$. A pairwise alignment $\mathcal{A}$ of sequences S_1 and S_2 is uniquely described by its *set of cuts* $\text{cuts}(\mathcal{A})$.

We define mappings between positions and columns in an alignment. For a gapped sequence T and $0 \leq j \leq |T_{|-}|$, translate from positions in T (i.e. alignment columns) to positions in $T_{|-}$ by $\text{ctpl}_T(j) := |(T[1] \dots T[j])_{|-}|$. Note that in the case where $T[j] = -$, $\text{ctpl}_T(j)$ points to the position left of the gap. For $0 \leq i \leq |T|$, $\text{ptc}_T(i)$ inversely translates from positions to columns, i.e. $\text{ptc}_T(i)$ is the unique j where $\text{ctpl}_T(j) = i$.

Example. We consider the triplewise multiple alignment

$$\mathcal{A} = \begin{pmatrix} \mathtt{C} & \mathtt{-} & \mathtt{-} & \mathtt{T} & \mathtt{A} \\ \mathtt{-} & \mathtt{G} & \mathtt{T} & \mathtt{T} & \mathtt{-} \\ \mathtt{C} & \mathtt{G} & \mathtt{T} & \mathtt{T} & \mathtt{-} \end{pmatrix} \tag{3.1}$$

of the sequences $S_1 = \mathtt{CTA}$, $S_2 = \mathtt{GTT}$ and $S_3 = \mathtt{CGTT}$. The rows of $\mathcal{A}$ are $A_1 = \mathtt{C--TA}$, $A_2 = \mathtt{-GTTC}$ and $A_3 = \mathtt{CGTTA}$. $\mathcal{A}$ is an alignment of $\mathcal{A}^1 = \begin{pmatrix} \mathtt{C} & \mathtt{T} & \mathtt{A} \end{pmatrix}$ and $\mathcal{A}^2 = \begin{pmatrix} \mathtt{-} & \mathtt{G} & \mathtt{T} & \mathtt{T} \\ \mathtt{C} & \mathtt{G} & \mathtt{T} & \mathtt{T} \end{pmatrix}$. The cuts of $\mathcal{A}^2$ are $(0,0), (0,1), (1,2), (2,3)$, and $(3,4)$ at respective columns 0 to 4. There is no other alignment of S_2 and S_3 with the same set of cuts. Furthermore, $\mathrm{ptc}_{A_2}(3) = 4$, $\mathrm{ctpl}_{A_1}(5) = 3$, and $\mathrm{ctpl}_{A_1}(3) = 1$. △

3.1.1 RNA secondary structure

A *base pair* is a pair $a = (i, j) \in \mathbb{N}^2$. We call $i =: a^\ell$ its *left end* and $j =: a^\mathrm{r}$ its *right end.* An *(RNA) structure P for length n* is a set of *base pairs* (i, j), $1 \leq i < j \leq n$, where no two different base pairs share a common end, i.e. for all $(i, j), (i', j') \in P$: $i = i' \implies j = j'$ and $j \neq i'$. We call P *crossing* if and only if there exist two base pairs $(i, j), (i', j') \in P$ such that $i < i' < j < j'$. Otherwise, P is called *non-crossing* or *nested.* In this thesis, all RNA structures are non-crossing.

3.2 Simultaneous Alignment and Folding (SA&F)

Following [18] and [35], we define a *sequence-structure similarity score for an alignment $|\mathcal{A}|$ and an RNA structure P for length $|\mathcal{A}|$.* In the case of a pairwise alignment

$\mathcal{A}$, this similarity score is of the form

$$\mathfrak{S}(S_1, S_2, A_1, A_2, P) = \tag{3.2}$$

$$\sum_{\substack{(i,j)\in P \\ A_1[i]\neq -, A_1[j]\neq - \\ A_2[i]\neq -, A_2[j]\neq -}} \tau^{S_1,S_2}(\mathrm{ctpl}_{A_1}(i), \mathrm{ctpl}_{A_1}(j), \mathrm{ctpl}_{A_2}(i), \mathrm{ctpl}_{A_2}(j)) \quad \text{(structural similarity)}$$

$$+ \sum_{\substack{1\leq i\leq n, \\ i \text{ unpaired in } P, \\ A_1[i]\neq -, A_2[i]\neq -}} \sigma^{S_1,S_2}\big(\mathrm{ctpl}_{A_1}(i), \mathrm{ctpl}_{A_2}(i)\big) \quad \text{(sequence similarity)}$$

$$+ \sum_{k>0} \gamma(k) N_k^{A_1,A_2}, \quad \text{(affine gap cost)}$$

where σ^{S_1,S_2} is a sequence similarity and τ^{S_1,S_2} is a structural similarity function, $\gamma(k) = \gamma_o + k\gamma_e$, and $N_k^{A_1,A_2}$ is the number of maximal subsequences of k gaps in A_1 and A_2. For the definition of σ^{S_1,S_2} and τ^{S_1,S_2} confer [35]. We generalize this to the q-wise case by sum-of-pairs, i.e. we define for $q \geq 2$,

$$\mathfrak{S}(\mathcal{A}, P) = \sum_{1\leq x<y\leq q} \mathfrak{S}(S_x, S_y, A_x, A_y, P).$$

Given sequences $S_1, \ldots, S_q$, the problem of *simultaneous alignment and folding* (SA&F) is

$$\underset{\mathcal{A} \text{ of } S_1,\ldots,S_q,\ P \text{ for length } |\mathcal{A}|}{\arg\max} \mathfrak{S}(\mathcal{A}, P).$$

An efficient algorithm to solve this specific problem for the pairwise case ($q = 2$) was introduced in [18] and significantly improved in [35]. Whereas LocARNA [35] (and therefore our implementation) supports affine gap cost, we keep presentation simple, by describing only linear gap cost, where each gap costs γ (i.e. $\gamma_o = 0, \gamma_e = \gamma$).

The pairwise LocARNA-algorithm has parameters $(n, m, \sigma, \tau, \gamma)$, where n and m are sequence lengths, σ denotes sequence similarity, τ structural similarity, and γ gap cost. We assume for our description w.l.o.g that the algorithm aligns two sequences

S_1 and S_2 of respective lengths n and m. The algorithm evaluates the recursion

$$
\begin{aligned}
M_{i\,i-1;k\,k-1} &= 0 \\
M_{i\,j;k\,l} &= \max \begin{cases} M_{i\,j-1;k\,l-1} + \sigma(j,l) \\ M_{i\,j-1;k\,l} + \gamma \\ M_{i\,j;k\,l-1} + \gamma \\ \max_{j'l'} M_{i\,j'-1;k\,l'-1} + D_{j'\,j;l'\,l} \end{cases} \\
D_{i\,j;k\,l} &= M_{i+1\,j+1;k-1\,l-1} + \tau(i,j;k,l)
\end{aligned}
\tag{3.3}
$$

for $1 \leq i < j \leq n$ and $1 \leq k < l \leq m$. The matrix entries $M_{i\,j;k\,l}$ are defined as the maximal similarity score of an alignment of subsequences $S_1[i]..S_1[j]$ and $S_2[k]..S_2[l]$. $D_{i\,j;k\,l}$ is the maximal similarity score of such an alignment where base pairs (i,j) and (k,l) are matched.

In this way, the pairwise LocARNA-algorithm solves the alignment problem for sequence S_1 and S_2 when parametrized by $(|S_1|, |S_2|, \sigma^{S_1,S_2}, \tau^{S_1,S_2}, \gamma)$. The maximal sequence-structure similarity is obtained as $M_{1\,1;n\,m}$ and the actual alignment is obtained by trace back from the dynamic programming matrices.

The same algorithm can be employed in a progressive alignment scheme to compute multiple alignments [18, 35]. There the algorithm computes an alignment $\mathcal{A}$ of two alignments $\mathcal{A}^1$ and $\mathcal{A}^2$. For this reason the algorithm is parametrized by $(|\mathcal{A}^1|, |\mathcal{A}^2|, \sigma^{\mathcal{A}^1,\mathcal{A}^2}, \tau^{\mathcal{A}^1,\mathcal{A}^2}, \gamma)$. Details on how to construct $\sigma^{\mathcal{A}^1,\mathcal{A}^2}$ and $\tau^{\mathcal{A}^1,\mathcal{A}^2}$ according to the sum-of-pairs idea are given in [18] and [35]. In our representation, we assume that the algorithm actually aligns two "consensus" sequences $\hat{S}_1$ and $\hat{S}_2$ of respective alignments $\mathcal{A}^1$ and $\mathcal{A}^2$. For these sequences, we require only that $\hat{S}_1$ and $\hat{S}_2$ have respective lengths $|\mathcal{A}^1|$ and $|\mathcal{A}^2|$. Their actual composition is arbitrary, since all similarity information is encoded by σ and τ, which depend only on sequence positions. By traceback, one obtains a pairwise alignment $\mathcal{A}_\mathrm{p}$ of $\hat{S}_1$ and $\hat{S}_2$. This alignment $\mathcal{A}_\mathrm{p}$ *induces* a multiple alignment $\mathcal{A}$ of $\mathcal{A}^1$ and $\mathcal{A}^2$, write $\mathcal{A} := [\![\mathcal{A}_\mathrm{p}]\!]^{\mathcal{A}^1}_{\mathcal{A}^2}$, which is optimal, among all alignments of $\mathcal{A}^1$ and $\mathcal{A}^2$, due to the defined sum-of-pairs

score.

3.3 SA&F constrained to limited deviation from a reference alignment

We modify the pairwise LocARNA algorithm in order to compute optimal alignments in a limited neighborhood around a reference alignment. The idea is inspired by a common ad-hoc heuristic applicable to dynamic programming sequence-structure alignment or pure sequence alignment. It is thus instructive, to recall simple Needleman-Wunsch-style alignment of sequences S_1 and S_2 that evaluates $M_{ij} = \max\{M_{i-1,j-1} + \sigma(S_1[i], S_2[j]), M_{i-1,j} + \gamma, M_{i,j-1} + \gamma\}$ for $1 \leq i \leq |S_1|$ and $1 \leq j \leq |S_2|$ (after initializing $M_{00} = 0$, $M_{i0} = i\gamma$, $M_{0j} = j\gamma$). Instead of filling the whole $\mathcal{O}(|S_1||S_2|)$ matrix, this heuristic computes only $\mathcal{O}(|S_1|\Delta)$ entries M_{ij}, where $|i - j| \leq \Delta$ and (virtually) sets all other cells to $-\infty$. This arbitrarily restricts the search space to alignments $\mathcal{A}$, where $|i - j| \leq \Delta$ for all possible decompositions of $\mathcal{A}$ into an alignment $\mathcal{A}^{\text{pre}}$ of prefixes $S_1[1 \ldots i]$ and $S_2[1 \ldots j]$ and an alignment $\mathcal{A}^{\text{suf}}$ of suffixes $S_1[i \ldots |S_1|]$ and $S_2[j \ldots |S_2|]$. For SA&F, such a heuristic has been introduced with dynalign [20]. In fact, our method will subsume this and similar ad-hoc heuristics as a special case where, e.g. in this special case and for $i < j$, the reference alignment $\mathcal{A}^{\text{R}}$ is

$$\begin{array}{cccccccc} A_1 & A_2 & \ldots & A_i & \ldots & - \\ B_1 & B_2 & \ldots & B_i & \ldots & B_j. \end{array}$$

3.3.1 Δ-deviation from an alignment

We are going to define a deviation of an alignment $\mathcal{A}$ from an alignment $\mathcal{A}^{\text{R}}$, called the reference alignment. This deviation is directly understood as a deviation of the cuts in $\mathcal{A}$ from cuts in $\mathcal{A}^{\text{R}}$. It will allow efficient optimization over all alignments in a limited deviation from $\mathcal{A}^{\text{R}}$.

First, we define a distance on pairwise cuts $c = (c_1, c_2)$ and $c' = (c'_1, c'_2)$ as their Manhattan distance $\|c - c'\|_1 = |c_1 - c'_1| + |c_2 - c'_2|$. Then, for pairwise alignments $\mathcal{A}$

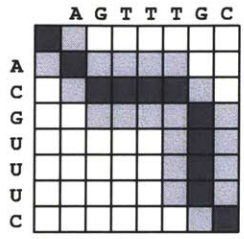


Figure 3-1: Matrix visualization of cut sets. Each cut (i, j) is represented as a matrix entry (i, j). The pairwise cuts of the alignment $\mathcal{A}^{\mathrm{R}}{=}^{t}$(`AC---GUUUC`, `AGTTTG---C`) (dark gray) and the cuts in 1-deviation from $\mathcal{A}^{\mathrm{R}}$ (dark and light gray).

and $\mathcal{A}^{\mathrm{R}}$ we define

$$\mathrm{d}_{\mathcal{A}^{\mathrm{R}}}(\mathcal{A}) = \max_{c \text{ cut of } \mathcal{A}} \Big(\min_{c^{\mathrm{R}} \text{ cut of } \mathcal{A}^{\mathrm{R}}} \|c - c^{\mathrm{R}}\|_1 \Big).$$

Now, we generalize this to a deviation of alignments $\mathcal{A}$ from $\mathcal{A}^{\mathrm{R}}$ of the same q sequences ($q \geq 2$) by

$$\mathrm{d}_{\mathcal{A}^{\mathrm{R}}}(\mathcal{A}) = \max \Big\{ \mathrm{d}_{\mathcal{A}^{\mathrm{R}}\langle i,j \rangle}(\mathcal{A}\langle i, j \rangle) \mid 1 \leq i < j \leq k \Big\}.$$

Fig. 3-1 visualizes the set of all cuts with (maximal) deviation 1 from a given example alignment.

Given sequences $S_1, \ldots, S_k$, the *SA&F problem with limited deviation* Δ *from a reference alignment* $\mathcal{A}^R$ is

$$\underset{\substack{\mathcal{A} \text{ of } S_1, \ldots, S_k \text{ in} \\ \Delta\text{-deviation from } \mathcal{A}^{\mathrm{R}}, \\ P \text{ structure for length } |\mathcal{A}|}}{\arg\max} \mathfrak{S}(\mathcal{A}, P).$$

3.3.2 Pairwise algorithm

The input of the pairwise algorithm consists of two sequences S_1 and S_2 and a pairwise reference alignment $\mathcal{A}^{\mathrm{R}}$ of these sequences. The pairwise alignment problem in deviation Δ from $\mathcal{A}^{\mathrm{R}}$ will be solved by a variant of the pairwise LocARNA-algorithm in Eq. 3.3 parametrized by $(|S_1|, |S_2|, \sigma^{S_1,S_2}, \tau^{S_1,S_2}, \gamma)$.

Due to the limitation, we change the semantics of the matrix entries $M_{ij;kl}$ and $D_{ij;kl}$ in a way that they contain the maximal score only over subalignments of

alignments with limited deviation Δ from $\mathcal{A}^{\mathrm{R}}$. Let $\mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta) \subseteq \{0, \ldots, |S_1|\} \times \{0, \ldots, |S_2|\}$ denote the set of cuts in Δ-deviation from $\mathcal{A}^{\mathrm{R}}$. It is defined such that for all alignments of the sequences of $\mathcal{A}^{\mathrm{R}}$ holds that

$$\mathrm{cuts}(\mathcal{A}) \subseteq \mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta) \text{ iff } \mathrm{d}_{\mathcal{A}^{\mathrm{R}}}(\mathcal{A}) \leq \Delta.$$

By definition of d_2, this is equivalent to

$$\mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta) := \left\{ (i,j) \;\middle|\; \begin{array}{l} 0 \leq i \leq |S_1|, 0 \leq j \leq |S_2|, \\ c_R \in \mathrm{cuts}(\mathcal{A}^{\mathrm{R}}) \colon \|c_R - c\|_1 \leq \Delta \end{array} \right\}.$$

Due to the definition of the matrix entries, we need to compute entries $M_{i\,j;k\,l}$ only if the optimal alignment can be derived from an alignment of subsequences $S_1[i] \ldots S_1[j]$ and $S_2[k] \ldots S_2[l]$, i.e. only if the cuts $(i-1, k-1)$ and (j, l) are in $\mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta)$. $D_{i\,j;k\,l}$ needs to be computed only when i can be matched to k and j can be matched to l, i.e. $(i,k), (i-1, k-1), (j,l)$, and $(j-1, l-1)$ are in $\mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta)$. Furthermore, the computation of $M_{i\,j;k\,l}$ is restricted to indices i and k that can match, i.e. where also $(i-2, k-2) \in \mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta)$ (with the exception of case $(i,k) = (1,1)$).

We will now describe an algorithm for constructing $\mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta)$. Since by definition for each $0 \leq i \leq |\mathcal{A}^{\mathrm{R}}|$, $\mathcal{C}_2(\mathcal{A}^{\mathrm{R}}, \Delta)$ contains consecutive elements $(i, j_i^{\min}) \ldots (i, j_i^{\max})$, the set can be conveniently described in terms of $j_i^{\min}$ and $j_i^{\max}$. This is useful, both for constructing the set and when restricting the alignment algorithm.

By iterating over the cuts of $\mathcal{A}^{\mathrm{R}}$, these values are efficiently computed (for $0 \leq i \leq |\mathcal{A}^{\mathrm{R}}|$) as

$$j_i^{\min} := \min_{(i_R, j_R) \text{ cut of } \mathcal{A}^{\mathrm{R}}} \begin{cases} \max(0, j_R - \Delta) & \text{if } i_R = i \\ i_R & \text{if } j_R - j \leq \Delta \end{cases}$$

and

$$j_i^{\max} := \max_{(i_R, j_R) \text{ cut of } \mathcal{A}^R} \begin{cases} \min(|S_2|, j_R + \Delta) & \text{if } i_R = i \\ i_R & \text{if } j - j_R \leq \Delta. \end{cases}$$

3.3.3 Progressive alignment heuristic

We devise a progressive alignment scheme to heuristically solve the problem of multiple alignment in deviation Δ from a reference alignment $\mathcal{A}^R$. The elementary operation of this scheme is computing an alignment $\mathcal{A}$ of two alignments $\mathcal{A}^1$ and $\mathcal{A}^2$ restricted by a set of permissible cuts $\mathcal{C}(\mathcal{A}^R, \Delta)$. The algorithm is analogous to the pairwise case. However, the definition and construction of this set is more involved than in the pairwise case, since cuts are required for a pairwise alignment, although the reference alignment is multiple. The algorithm is parametrized as in unrestricted progressive alignment.

Let $\mathcal{A}^1$ and $\mathcal{A}^2$ be alignments of respective sequences $S_1, \ldots S_{\text{rows}(\mathcal{A}^1)}$ and $S_{\text{rows}(\mathcal{A}^1)+1}, \ldots, S_q$. W.l.o.g. let $\mathcal{A}^R$ be a multiple alignment of the sequences $S_1, \ldots, S_q$. Recall that in our presentation, the algorithm computes an alignment $\mathcal{A}_p$ of consensus sequences $\hat{S}_1$ and $\hat{S}_2$ of respective $\mathcal{A}^1$ and $\mathcal{A}^2$, such that the optimal alignment of $\mathcal{A}^1$ and $\mathcal{A}^2$ is induced by $\mathcal{A}_p$. We define a set of permissible cuts $\mathcal{C}(\mathcal{A}^R, \Delta)$ for the alignment $\mathcal{A}_p$ of $\hat{S}_1$ and $\hat{S}_2$, such that the $\mathcal{A}_p$ with $\text{cuts}(\mathcal{A}_p) \subseteq \mathcal{C}(\mathcal{A}^R, \Delta)$ describe exactly the alignments whose induced alignments are in Δ-deviation from the reference alignment, i.e.

$$\text{cuts}(\mathcal{A}_p) \subseteq \mathcal{C}(\mathcal{A}^R, \Delta) \text{ iff } \mathrm{d}_{\mathcal{A}^R}([\![\mathcal{A}_p]\!]^{\mathcal{A}^1}_{\mathcal{A}^2}) \leq \Delta.$$

Due to the definition of the deviation as "maximum-of-pairs", we can compute this set as intersection of pairwise cut sets $\mathcal{C}(\mathcal{A}^R, x, y, \Delta) \subseteq \{0, \ldots, \text{rows}(\mathcal{A}_1)\} \times \{0, \ldots, \text{rows}(\mathcal{A}_2)\}$ $(1 \leq x \leq |\mathcal{A}^1|, |\mathcal{A}^1| + 1 \leq y \leq q)$ that guarantee the Δ-deviation

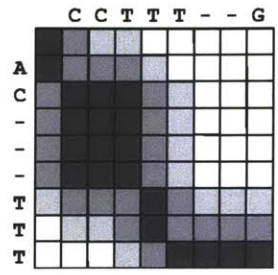


Figure 3-2: Example of cut sets $\mathcal{C}(\mathcal{A}^{\mathrm{R}}, x, y, \Delta)$. Matrix visualization as in Figure 3-1. Sets for $\Delta = 0$ (dark gray), $\Delta = 1$ (dark+medium gray), and $\Delta = 2$ (dark+medium+light gray). ${A^1}_x$ = `AC---TTT`, ${A^2}_{(y'-\mathrm{rows}(\mathcal{A}_1))}$ = `CCTTT--G`, ${A^{\mathrm{R}}}_x$ = `AC---TTT-`, ${A^{\mathrm{R}}}_y$ = `-CCTT-TG`

for the pairwise alignment of sequences S_x and S_y, i.e.

$$\mathrm{cuts}(\mathcal{A}_{\mathrm{p}}) \subseteq \mathcal{C}(\mathcal{A}^{\mathrm{R}}, x, y, \Delta)$$
$$\text{iff } \mathrm{d}_{\mathcal{A}^{\mathrm{R}}\langle x,y\rangle}([\![\mathcal{A}_{\mathrm{p}}]\!]^{\mathcal{A}^1}_{\mathcal{A}^2}\langle x, y\rangle) \leq \Delta.$$

Let $y' = y - \mathrm{rows}(\mathcal{A}_1)$ denote the index of sequence S_y in $\mathcal{A}_2$. The set $\mathcal{C}(\mathcal{A}^{\mathrm{R}}, x, y, \Delta)$ from the gapped sequences ${A^1}_x$, ${A^2}_{y'}$, ${A^{\mathrm{R}}}_x$, ${A^{\mathrm{R}}}_y$ is generated as follows. A cut c_p of A_p corresponds to a cut $c = [\![c_p]\!]^{{A^1}_x}_{{A^2}_{y'}} = (\mathrm{ctpl}_{{A^1}_x}(i), \mathrm{ctpl}_{{A^2}_{y'}}(j))$ of $[\![\mathcal{A}_{\mathrm{p}}]\!]^{\mathcal{A}^1}_{\mathcal{A}^2}\langle x, y\rangle$. For each cut c of the alignment $\mathcal{A}^{\mathrm{R}}\langle x, y\rangle$, we generate the set of cuts c' in distance Δ. Then for each such cut c', we generate the cuts c_p, where $c' = [\![c_p]\!]^{{A^1}_x}_{{A^2}_{y'}}$, and add them to the set $\mathcal{C}(\mathcal{A}^{\mathrm{R}}, x, y, \Delta)$. As in the pairwise case, this is conveniently computed in terms of boundaries $j_i^{\min}$ and $j_i^{\max}$ for "matrix rows" $1 \leq i \leq |\mathcal{A}_1|$. Figure 3-2 provides examples of cut sets $\mathcal{C}(\mathcal{A}^{\mathrm{R}}, x, y, \Delta)$.

All multiple alignments generated by a progressive alignment scheme built on this strict definition of permissible cuts $\mathcal{C}(\mathcal{A}^{\mathrm{R}}, \Delta)$ will have at most deviation Δ from the reference alignment. However, due to potential misalignments in previous progressive steps, this strategy can fail to produce an alignment at all. We remark that such potential inconsistencies are unavoidable in a method that guarantees the maximal deviation Δ and obeys the principles of progressive alignment, i.e. that each single progressive alignment step sees only the local information from the input alignments and alignments generated in previous steps are not changed ("once a gap, always a

gap"). In our experiments such inconsistencies are rare and such rare events can be tolerated.

Relaxed cut sets. For applications that require guaranteed success, we propose a relaxation of the method that avoids inconsistencies by relaxing the distance constraints in an optimal way. By dynamic programming, we compute a relaxed cut set $\mathcal{C}_{\text{relaxed}}(\mathcal{A}^{\text{R}}, \Delta)$ that has a size limited by Δ and minimizes the distance to the sets $\mathcal{C}(\mathcal{A}^{\text{R}}, x, y, \Delta)$. $\mathcal{C}_{\text{relaxed}}(\mathcal{A}^{\text{R}}, \Delta)$ is computed as set of cuts in Δ- deviation from an alignment $\mathcal{A}'$ of $\hat{S}_1$ and $\hat{S}_2$. $\mathcal{A}'$ minimizes $\sum_{c \in \text{cuts}(\mathcal{A}')} \text{cost}(c)$, where the cost of a cut of $\mathcal{A}'$ is defined by

$$\text{cost}(c) := \sum_{\substack{1 \le x \le \text{rows}(\mathcal{A}_1), \\ \text{rows}(\mathcal{A}_1) < y \le \text{rows}(\mathcal{A})}} \min_{c' \in \mathcal{C}(\mathcal{A}^{\text{R}}, x, y, 0)} \|c - c'\|_1.$$

The alignment $\mathcal{A}'$ is obtained by traceback from the dynamic programming matrix C evaluating $\text{C}(0,0) = \text{cost}(0,0), \text{C}(i,0) = \text{cost}(i,0) + \text{C}(i-1,0), \text{C}(0,j) = \text{cost}(0,j) + \text{C}(0,j-1)$, and

$$\text{C}(i,j) = \text{cost}(i,j) + \min\{\text{C}(i-1,j-1), \text{C}(i-1,j), \text{C}(i,j-1)\}$$

for $1 \le i \le |\mathcal{A}_1|, 1 \le j \le |\mathcal{A}_2|$. Finally, we set $\mathcal{C}_{\text{relaxed}}(\mathcal{A}^{\text{R}}, \Delta) := \mathcal{C}(\mathcal{A}', \Delta)$. Clearly, a heuristic based on this relaxation will not guarantee Δ-deviation from $\mathcal{A}^{\text{R}}$. However, by construction, it will favor low deviation and limit the computational cost and deviation in each progressive alignment step by Δ.

Chapter 4

Results

In this chapter, we demonstrate the effectiveness of the realignment pipeline (Chapter 2) in conjunction with constrained realignment (Chapter 3). We show that LocARNA is more powerful as the method for realignment in the pipeline than is Muscle as measured by the resulting prediction of new putative ncRNAs.

We ran our pipeline on the whole genome alignments of 12 fly genomes compiled by the *Drosophila* Twelve Genomes Consortium [1, 9] and of genomic regions selected from 28 vertebrates for the ENCODE project [19]. We shall refer to these alignments simply as "Fly" and "Encode", respectively. Both alignments were constructed using the Pecan tool [23]. These sets of genomes were chosen because of their use in previous *de novo* ncRNA prediction screens: Fly in [26] and Encode in [30, 33]. Moreover, they have become regarded in the biological community as test models for comparative sequence analysis of closely related species.

Loci were realigned in step (5) of the pipeline with either LocARNA [35] or Muscle [14]. LocARNA realignments were constrained to Δ-deviations of 5, 10, or 20. Muscle served as a control against LocARNA to distinguish between realignment effects that are reproducible just from the act of realignment versus those uniquely stemming from LocARNA's explicit optimization for structural similarity. In BraliBase benchmark reviews [36], Muscle was shown to be one of the most accurate sequence-similarity tools for aligning structural RNAs.

RNAz p scores of loci and windows were evaluated multiple times: once with

respect to the original WGA in step (4), and again after realignment with LocARNA or Muscle in step (6) of the pipeline. Note that only windows were directly evaluated by RNAz to generate p scores. The p score of a locus was defined to be the maximum p score among the windows composing the locus.

Following the convention of previous genome-wide RNAz screens [26, 33], we predicted ncRNA sites by applying a lower threshold of 0.5 or 0.9 on the p scores. Windows and loci with a p score of at least 0.5 are labeled "low confidence" hits, and those with a p score of at least 0.9 are labeled "high confidence" hits.

4.1 Construction of the stability filter

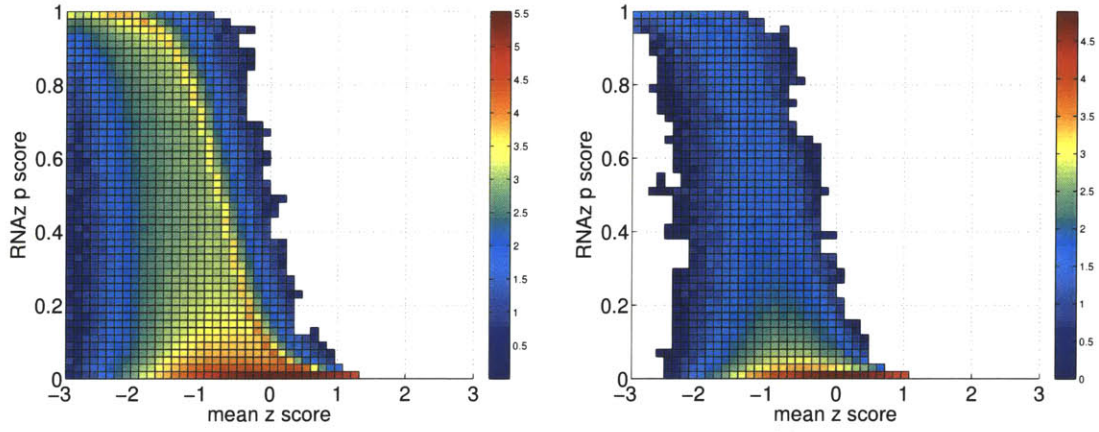
Execution of the thermodynamic stability filter in step (2) of the pipeline first requires the selection of a stability threshold on which to filter windows containing structurally stable sequences. Recall from Chapter 2 that the stability of a window is assessed by the average z-score of the mean free energy (MFE) of the window's sequences. A window passes the filter if its average z-score is below a fixed threshold θ_{stable}. Note that this measurement of stability depends only on which sequences are in a window and not by how the sequences are aligned. This filter is therefore robust to local alignment inaccuracies within windows in the original WGA.

The goal of the stability filter is to improve the overall run-time of the pipeline by avoiding further processing of windows that are unlikely to produce RNAz hits due to insufficient thermodynamic stability while simultaneously retaining most of the windows that will produce hits. Selecting a stability threshold, therefore, involves a trade off between the pipeline's computational tractability and sensitivity for ncRNAs in the form of RNAz hits. If the stability filter is weak (a high θ_{stable}), then it will pass more windows that will not form hits at the end of the pipeline due to insufficient structural stability in the windows' sequences. Thus, the total computation of the pipeline is unnecessarily increased for these windows. On the other hand if the filter is strict (a low θ_{stable}), fewer windows will pass, but now the filter will prematurely remove more windows that would otherwise form hits. Thus, the threshold needs to

be carefully chosen in order to optimize for this performance trade off.

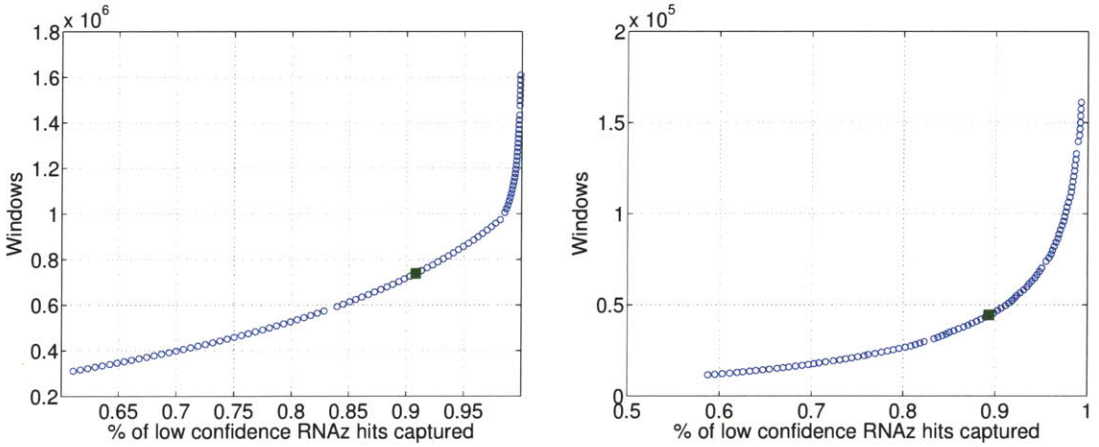
Since the end result of the pipeline is the prediction and patching of ncRNA sites, at this stage of the pipeline we cannot completely predetermine the sensitivity of the stability filter under a given threshold. In lieu of this cat-and-mouse problem, we offer a workaround as follows. We first run RNAz over all windows in the original WGA returned by step (1) to identify low-confidence hits ($p \geq 0.5$) and their average z-scores. Note that these calculations are not shown explicitly in Figure 2-1 but are regarded as implicit pre-processing before applying the stability filter. We then assume that the distribution of these hits according to their average z-scores is an approximation of the distribution of actual ncRNAs in the genomes. Thus for a fixed threshold θ_{stable}, we estimate the filter's sensitivity for ncRNAs to be the fraction of the hits that would pass under θ_{stable}.

Under this approximation framework, Figure 4-1 illustrates that simply setting a threshold is an effective means for constructing a stability filter. The figure is a 2D histogram of the windows returned by step (1) according to their average z-scores and to their p scores in the original WGA. While in general all windows are symmetrically distributed around an average z-score of about 0, most low-confidence hits ($p \geq 0.5$) have a low average z-score. Figure 4-2 plots the total number of windows and the fraction of low confidence-hits that would pass under various thresholds. This curve exhibits a desirable region of thresholds where the filter retains most hits but removes many windows. Higher thresholds beyond this region provide only marginal increases in hit sensitivity while the total number of passing windows quickly increases. Inversely, lower thresholds only have marginal filtering of windows but see a quick drop in hit sensitivity. Thus, we chose a threshold from this region (Table 4.1), in particular $\theta_{\text{stable}} = -1$ for our implementation of the pipeline. Windows passing the filter under this threshold were then merged and reassembled in step (3) into 33846 loci in Encode and 503830 loci in Fly.



(a) Fly (b) Encode

Figure 4-1: 2D histogram of windows in the original WGA according to average z-scores of MFE and to RNAz p scores. Each discrete square counts the number of windows that fall within an interval of 0.1 in the p scores and average z-scores. Counts are color-coded on a base 10 logarithmic scale. Most low-confidence RNAz hits ($p \geq 0.5$) have a low average z-score.



(a) Fly (b) Encode

Figure 4-2: Selecting a stability threshold. Each point on the curve plots, for a fixed threshold θ_{stable}, the number of windows and the fraction of low-confidence RNAz hits ($p \geq 0.5$) that would pass the stability filter. θ_{stable} presents a trade off between the run-time of the pipeline and sensitivity for ncRNAs. The green square marks the chosen threshold $= -1$.

4.2 RNAz p scores after realignment

Realignment noticeably changes the RNAz p scores of loci. Figure 4-3 is a 2D histogram of loci according to p scores evaluated before and after realignment. Located

Table 4.1: Performance of the stability filter under the chosen threshold $\theta_{\text{stable}} = -1$. Listed are the total number of windows and low-confidence RNAz hits ($p \geq 0.5$) obtained after slicing the original WGA (step (1)) and the number that remains after being filtered for stability under $\theta_{\text{stable}} = -1$ (step (2)). The stability filter reduced the set of total windows while being sensitive for hits.

	Fly			Encode		
	Sliced	Passing	(%)	Sliced	Passing	(%)
Total windows	5598326	737928	13.2%	829518	44328	5.3%
Low-confidence RNAz hits	348648	316623	90.8%	9615	8588	89.3%

along the left strips of 4-3(c)-(d), many loci start off with a very low p score in the original WGA but acquire a very p high score after realignment with LocARNA constrained to $\Delta = 20$. On the other hand, such loci are relatively absent after realignment with Muscle.

In general, p scores changed more so after realignment with LocARNA, as exhibited by the cloud of loci around both sides of the diagonals in Figure 4-3(c)-(d), than they did after realignment with Muscle, where most loci are remain concentrated exactly along the diagonal in Figure 4-3(a)-(b).

Some loci even experienced a drop in p score after realignment with LocARNA, but we believe that this can be mostly explained by the use of differently trained models of RNAz. In the original WGA and Muscle realignments, loci were evaluated with the sequence-similarity model trained over Clustal alignments, but in the LocARNA alignments, loci were evaluated with the more appropriate structural-similarity model trained over LocARNATE alignments. We believe that RNAz, when ran over the same or similar alignments, tends to assign a lower p score in the structural model than the sequence model in order to compensate for false structural conservation that is more likely produced with LocARNATE than with Clustal. Therefore, we believe that the p score of a locus probably decreases even when the LocARNA realignment does not significantly change the original alignment. This may happen, for example, if structural-similarity is already nearly optimized in the original alignment, or if the

sequence identity of the locus is too high to permit significant alignment change.

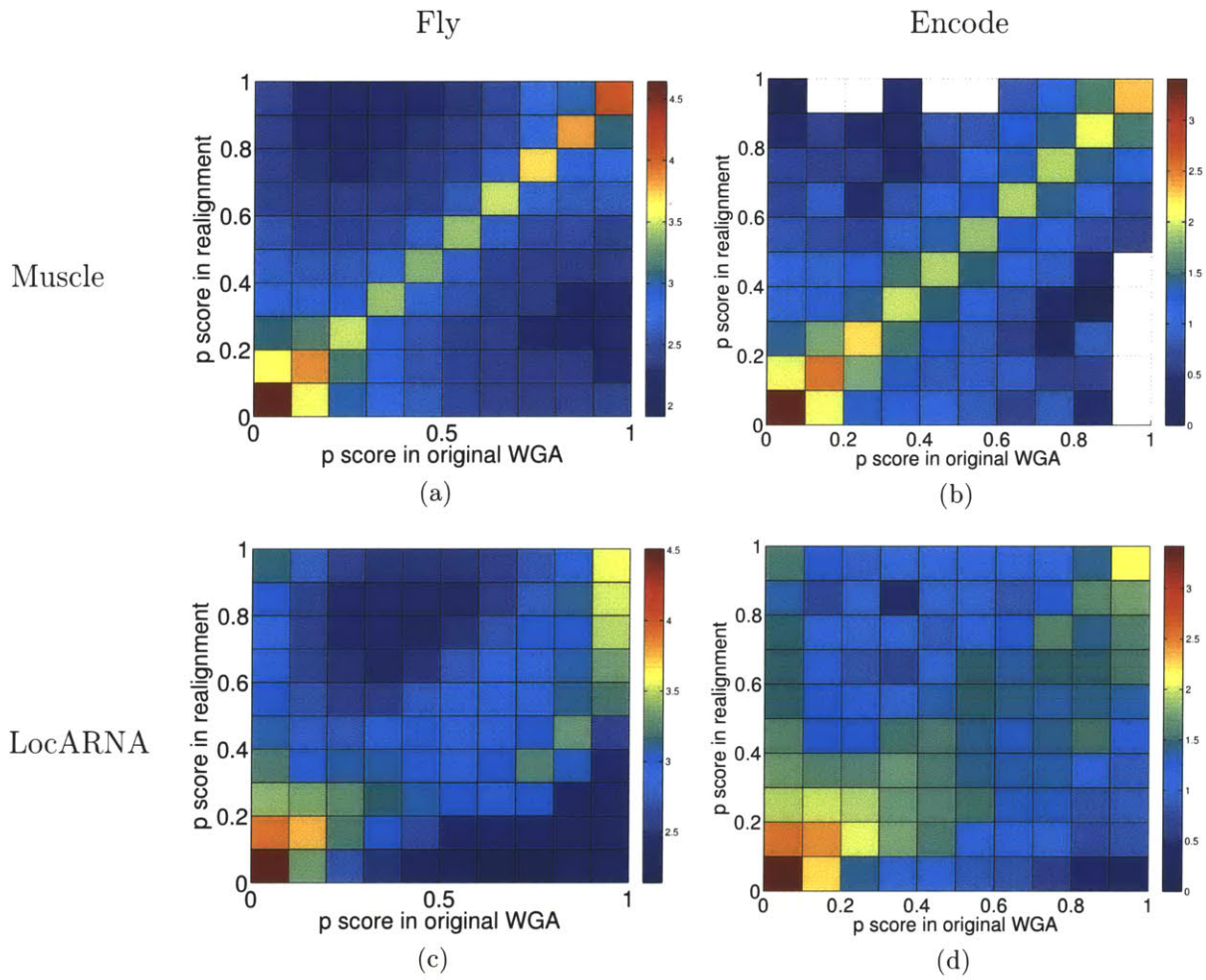


Figure 4-3: RNAz p scores of loci evaluated in the original WGA and after realignment. Loci were counted according to fixed intervals of 0.1 in the p scores before and after realignment with Muscle or LocARNA constrained to $\Delta = 20$. The number of loci is color-coded on a base 10 logarithmic scale. The diagonals count the loci experiencing less than 0.1 change in score. The loci in the left strip of the LocARNA histograms start with a very low score in the original WGA but acquire a very high score after realignment. By comparison, the score after realignment with Muscle is relatively unchanged.

The change from a locus's original alignment to its realignment was measured with the COMPALIGNP tool [36]. Briefly, COMPALIGNP computes the fraction of the original alignment that agrees with the realignment. A COMPALIGNP score of 1 indicates identical alignments and a 0 indicates complete disagreement. Figure 4-4 shows that a large increase in the p score after realignment with LocARNA is often seen when

realignment was extensive. This strong correlation suggests that some loci are "hidden" from prediction in the original WGA because of a low p score, but nevertheless they can be "uncovered" by a sufficiently different realignment.

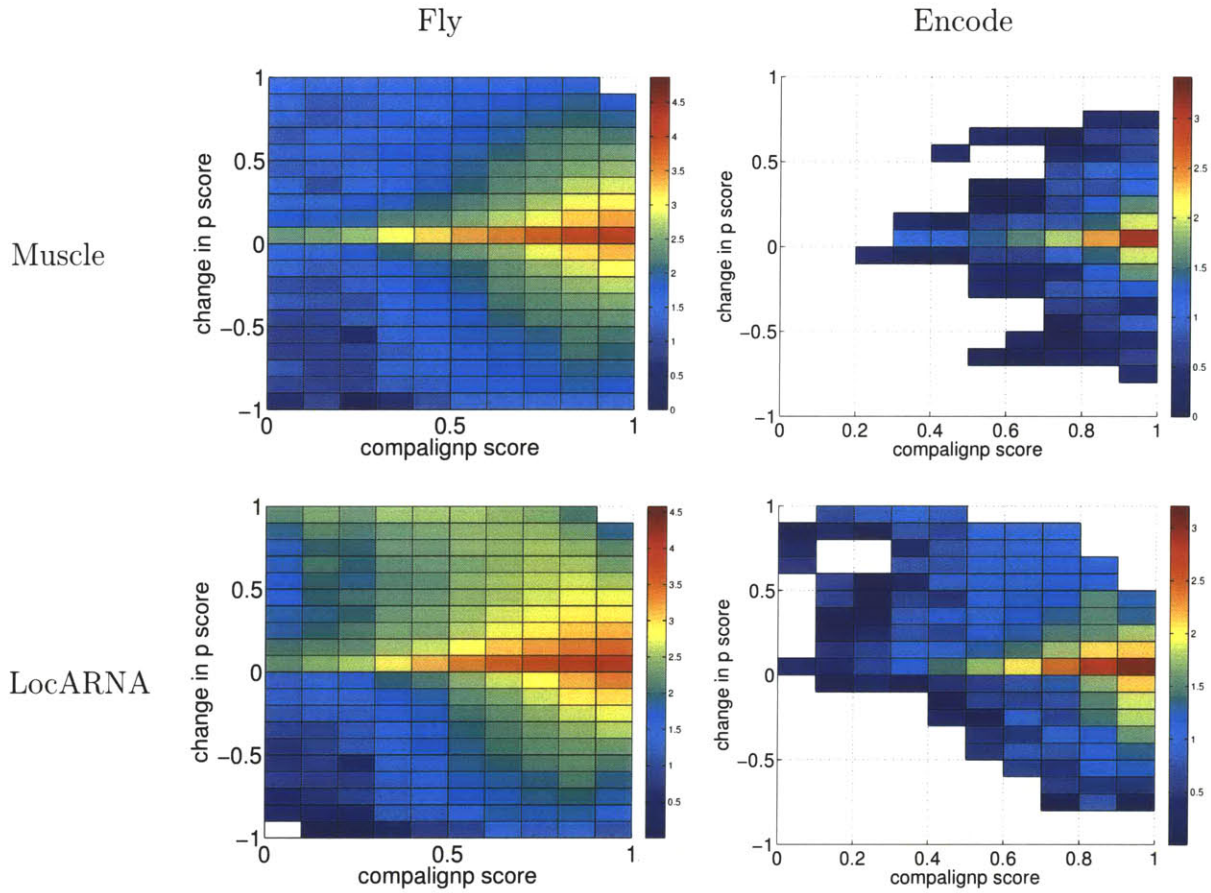


Figure 4-4: COMPALIGNP score vs. change in RNAz p scores.

4.3 New hits discovered after realignment

For each realignment method, a non-overlapping subset of the loci was selected (step (7)). For the rest of this chapter, we shall refer to these subsets of the loci by default unless otherwise stated.

For each p score threshold of 0.5 or 0.9, we delineated loci according to whether they were RNAz hits before realignment in the original WGA and/or after realignment. We refer to loci that are hits before and after realignment as "common hits",

loci that are hits only after realignment as "new hits", and those that are hits before realignment but are no longer so after realignment as "old hits". Table 4.2 summarizes the breakdown of RNAz hits according to these distinctions.

All methods of realignment revealed thousands of new hits among the 12 fly genomes and hundreds among the 28 vertebrates. These hits were not detected from the original WGA. In this regard of discovering more ncRNA sites, realigning with Muscle was the least effective, and realigning with LocARNA under an allowed deviation of $\Delta = 20$ was the most effective. Realigning with Δ-deviations smaller than 20 did not result in as many new hits, suggesting that some loci require more extensive realignment than others to be discovered *de novo*. Moreover, the increase in new hits from $\Delta = 10$ to 20 is much smaller than the increase from $\Delta = 5$ to 10. It is likely that this reflects a diminishing marginal returns in new hits as Δ increases. Therefore, constraining the LocARNA realignments by setting Δ to a sufficiently high value should reveal the same number of hits as by alignment with no constraints. This supports the use of the framework for constrained realignment presented in Chapter 3 to make the pipeline run faster.

4.4 Sequence identity of new hits

The average pairwise sequence identity (APSI) of every locus was measured with respect to the original WGA using the ALISTAT tool [13]. New hits after realignment had lower sequence identities on the mean and in the median than all of the hits discovered in the original WGA (Table 4.3). Figure 4-5 shows the shift in the sequence identity distribution. With respect to decreasing identity, the realignment methods ranked in the same order of effectiveness as they did with respect to the total number of new hits. This is also consistent with the most recent Bralibase benchmark study [36] that showed that as identity drops, constructing accurate structural alignments becomes more difficult for tools purely based on sequence-similarity. In particular, Wilm et al. [36] identified a "twilight" zone of $\leq 60\%$ identity where the performance of these methods is outperformed by methods that explicitly optimize for structural-

Fly

		New Hits (%)		Common Hits (%)		Old Hits (%)		All Hits	All Loci
$p \geq 0.5$	Muscle	**6305**	(8.4%)	61539	(82.3%)	6943	(9.3%)	74787	157400
	LocARNA, $\Delta = 5$	**10287**	(13.0%)	51587	(65.1%)	17339	(21.9%)	79213	158158
	LocARNA, $\Delta = 10$	**14076**	(16.6%)	53085	(62.5%)	17786	(20.9%)	84947	162582
	LocARNA, $\Delta = 20$	**16000**	(18.6%)	52708	(61.1%)	17514	(20.3%)	86222	161391
$p \geq 0.9$	Muscle	**3986**	(10.5%)	28994	(76.7%)	4815	(12.7%)	37795	157400
	LocARNA, $\Delta = 5$	**5826**	(14.6%)	21419	(53.5%)	12777	(31.9%)	40022	158158
	LocARNA, $\Delta = 10$	**8339**	(19.2%)	21960	(50.4%)	13236	(30.4%)	43535	162582
	LocARNA, $\Delta = 20$	**9914**	(22.2%)	21756	(48.6%)	13059	(29.2%)	44729	161391

Encode

		New Hits (%)		Common Hits (%)		Old Hits (%)		All Hits	All Loci
$p \geq 0.5$	Muscle	**144**	(9.9%)	1141	(78.8%)	163	(11.3%)	1448	5426
	LocARNA, $\Delta = 5$	**284**	(16.5%)	1008	(58.4%)	433	(25.1%)	1725	6597
	LocARNA, $\Delta = 10$	**401**	(21.7%)	1038	(56.2%)	409	(22.1%)	1848	6607
	LocARNA, $\Delta = 20$	**460**	(24.5%)	1027	(54.7%)	391	(20.8%)	1878	6410
$p \geq 0.9$	Muscle	**70**	(13.1%)	401	(75.2%)	62	(11.6%)	533	5426
	LocARNA, $\Delta = 5$	**133**	(20.1%)	331	(50.1%)	197	(29.8%)	661	6597
	LocARNA, $\Delta = 10$	**206**	(28.5%)	331	(45.8%)	186	(25.7%)	723	6607
	LocARNA, $\Delta = 20$	**243**	(32.4%)	331	(44.1%)	177	(23.6%)	751	6410

Table 4.2: Loci were evaluated with RNAz before realignment in the original WGA and after realignment with Muscle or LocARNA. "All loci" refers to the non-overlapping set of loci selected in step (7) of the realignment pipeline. Of those loci, "new hits" are the ones newly predicted only after realignment, "common hits" are the ones predicted before and after realignment, and "old hits" are the ones predicted only before realignment in the original WGA. .

similarity. The enrichment of lower sequence identity in the new hits, especially after applying LocARNA, therefore suggests that not only is realignment revealing new hits, but these hits were inherently more challenging to align accurately during the construction of the original WGA.

4.5 Validation with known ncRNAs

Known ncRNAs from Flybase and Rfam served as independent validation of RNAz hits. Table 4.4 lists the number of loci in Fly that matched at least one *D. melanogaster* ncRNA listed in Rfam or Flybase. A locus was considered to match an annotation if the locus contains a *D. melanogaster* sequence and there is a non-zero overlap in genomic position, disregarding strand orientation. Note that we did not distinguish between a match where a locus is completely contained inside a larger annotation, or

Fly

		New Hits		Common Hits		Old Hits		All Hits		All loci	
		mean	median	mean	median	mean	median	mean	median	mean	median
$p \geq 0.5$	Muscle	**0.77**	**0.78**	0.92	0.97	0.81	0.82	0.90	0.96	0.84	0.86
	LocARNA, $\Delta = 5$	**0.75**	**0.76**	0.91	0.96	0.91	0.98	0.89	0.95	0.84	0.86
	LocARNA, $\Delta = 10$	**0.74**	**0.74**	0.91	0.96	0.92	0.98	0.88	0.95	0.84	0.87
	LocARNA, $\Delta = 20$	**0.73**	**0.73**	0.91	0.96	0.92	0.98	0.88	0.94	0.84	0.87
$p \geq 0.9$	Muscle	**0.81**	**0.82**	0.94	0.98	0.85	0.88	0.91	0.97	0.84	0.86
	LocARNA, $\Delta = 5$	**0.76**	**0.76**	0.92	0.96	0.93	0.98	0.90	0.96	0.84	0.86
	LocARNA, $\Delta = 10$	**0.74**	**0.73**	0.92	0.97	0.93	0.98	0.89	0.96	0.84	0.87
	LocARNA, $\Delta = 20$	**0.72**	**0.72**	0.92	0.97	0.93	0.98	0.88	0.95	0.84	0.87

Encode

		New Hits		Common Hits		Old Hits		All Hits		All loci	
		mean	median	mean	median	mean	median	mean	median	mean	median
$p \geq 0.5$	Muscle	**0.69**	**0.71**	0.81	0.81	0.71	0.75	0.79	0.79	0.75	0.75
	LocARNA, $\Delta = 5$	**0.68**	**0.68**	0.82	0.83	0.77	0.76	0.78	0.78	0.75	0.75
	LocARNA, $\Delta = 10$	**0.67**	**0.67**	0.81	0.82	0.76	0.76	0.77	0.77	0.75	0.75
	LocARNA, $\Delta = 20$	**0.66**	**0.67**	0.81	0.82	0.77	0.75	0.77	0.77	0.75	0.75
$p \geq 0.9$	Muscle	**0.69**	**0.72**	0.83	0.82	0.74	0.76	0.80	0.79	0.75	0.75
	LocARNA, $\Delta = 5$	**0.66**	**0.67**	0.83	0.85	0.78	0.77	0.78	0.79	0.75	0.75
	LocARNA, $\Delta = 10$	**0.65**	**0.67**	0.82	0.84	0.78	0.77	0.77	0.77	0.75	0.75
	LocARNA, $\Delta = 20$	**0.65**	**0.66**	0.82	0.83	0.79	0.77	0.76	0.76	0.75	0.75

Table 4.3: Sequence identity of loci. The average pairwise sequence identity (APSI), computed with the ALISTAT tool on every locus with respect to the original WGA, served as a measure of sequence identity. New hits after realignment (in bold) are more concentrated in lower sequence identity levels.

vice versa, and a match with only partial overlap.

Several new hits discovered after realignment match known ncRNAs. More of such hits were seen with LocARNA realignments at higher values of Δ than at lower Δ's or with Muscle realignments. This is consistent with the relative number of new hits for each realignment methods. While there are also many validated old hits, they are still outnumbered by the validated new hits.

4.6 False Discovery Rate of the pipeline

We estimated the false discovery rate (FDR) of our realignment pipeline according to the procedure described in Chapter 6.5. Table 4.5 compares the FDR for predictions made from the original WGA and after each realignment method. We find that the FDR does increase after realignment but remains about the same. In fact, the FDR

Flybase

		New hits	Common hits	Old hits	All hits	All loci
$p \geq 0.5$	Muscle	11	202	9	222	329
	LocARNA, $\Delta = 5$	**23**	235	12	270	379
	LocARNA, $\Delta = 10$	**23**	233	15	271	383
	LocARNA, $\Delta = 20$	**24**	241	14	279	389
$p \geq 0.9$	Muscle	13	156	7	176	329
	LocARNA, $\Delta = 5$	**26**	174	11	211	379
	LocARNA, $\Delta = 10$	**24**	174	11	209	383
	LocARNA, $\Delta = 20$	**28**	181	11	220	389

Rfam

		New hits	Common hits	Old hits	All hits	All loci
$p \geq 0.5$	Muscle	3	162	2	167	186
	LocARNA, $\Delta = 5$	**3**	183	1	187	206
	LocARNA, $\Delta = 10$	**3**	171	1	175	193
	LocARNA, $\Delta = 20$	**4**	179	1	184	202
$p \geq 0.9$	Muscle	6	134	2	142	186
	LocARNA, $\Delta = 5$	**10**	145	7	162	206
	LocARNA, $\Delta = 10$	**11**	135	5	151	193
	LocARNA, $\Delta = 20$	**13**	143	5	161	202

Table 4.4: Validation with known ncRNAs. Loci were matched against annotations of known *D. melanogaster* ncRNAs in Flybase and Rfam. A locus was counted if it contains a *D. melanogaster* sequence and overlaps in genomic position, disregarding strand orientation, with an least one annotation. Several new hits after realignment with LocARNA (in bold) were validated, more so than the new hits with Muscle.

after realignment with LocARNA at any Δ is slightly lower in Fly. Thus, we believe that the new hits discovered after realignment contain about the same fraction of true ncRNAs as the hits from the original WGA.

Fly

		Original WGA (tot. windows = 5598326)		Randomized WGA (tot. windows = 5461614)		FDR
		Window hits	Hit %	Window hits	Hit %	
$p \geq 0.5$	Original WGA	316623	5.66%	211241	3.87%	**0.68%**
	Muscle	310390	5.54%	207867	3.81%	**0.69%**
	LocARNA, $\Delta = 5$	251306	4.49%	154685	2.83%	**0.63%**
	LocARNA, $\Delta = 10$	263011	4.70%	161404	2.96%	**0.63%**
	LocARNA, $\Delta = 20$	271213	4.84%	166437	3.05%	**0.63%**
$p \geq 0.9$	Original WGA	132464	2.37%	77876	1.43%	**0.60%**
	Muscle	129652	2.32%	76901	1.41%	**0.61%**
	LocARNA, $\Delta = 5$	95808	1.71%	49192	0.90%	**0.53%**
	LocARNA, $\Delta = 10$	103058	1.84%	52819	0.97%	**0.53%**
	LocARNA, $\Delta = 20$	108621	1.94%	56273	1.03%	**0.53%**

Encode

		Original WGA (tot. windows = 829518)		Randomized WGA (tot. windows = 815524)		FDR
		Window hits	Hit %	Window hits	Hit %	
$p \geq 0.5$	Original WGA	8588	1.04%	4102	0.50%	**0.49%**
	Muscle	8442	1.02%	4179	0.51%	**0.50%**
	LocARNA, $\Delta = 5$	7559	0.91%	3712	0.46%	**0.50%**
	LocARNA, $\Delta = 10$	8364	1.01%	4057	0.50%	**0.49%**
	LocARNA, $\Delta = 20$	8904	1.07%	4381	0.54%	**0.50%**
$p \geq 0.9$	Original WGA	2922	0.35%	1150	0.14%	**0.40%**
	Muscle	2910	0.35%	1176	0.14%	**0.41%**
	LocARNA, $\Delta = 5$	2544	0.31%	977	0.12%	**0.39%**
	LocARNA, $\Delta = 10$	2888	0.35%	1150	0.14%	**0.41%**
	LocARNA, $\Delta = 20$	3191	0.38%	1318	0.16%	**0.42%**

Table 4.5: False discovery rates of predictions

4.7 Examples

Figure 4-6 shows the alignments and consensus structures of an example locus, located on columns 9363200 to 9368080 in the syntenic block X_3665964_3708413 of Fly. Consensus secondary structures were computed and drawn with RNAALIFOLD [4].

The locus is contained with Flybase gene roX1 (ID = FBtr0070634), a long ncRNA that increases the expression of the X chromosome in *D. melanogaster* to compensate for the presence of only one X chromosome in male individuals [28]. The locus was a high-confidence hit after realignment with LocARNA constrained at $\Delta = 20$ ($p = 0.93$) but not in the original WGA ($p = 0$) and was only a low-confidence hit after realignment with Muscle ($p = 0.72$). This example suggests that new hits after realignment with LocARNA include many other functional ncRNAs.

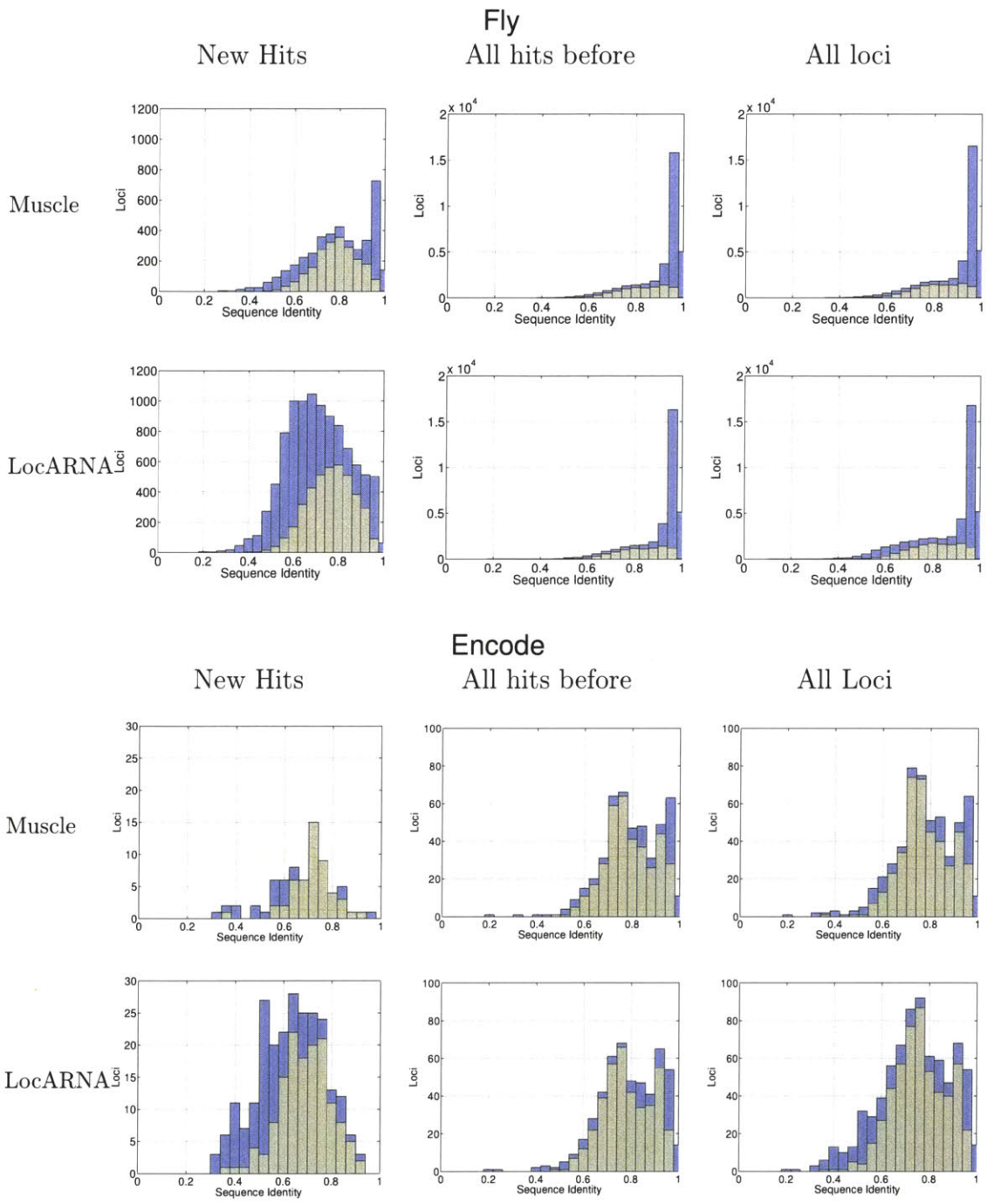
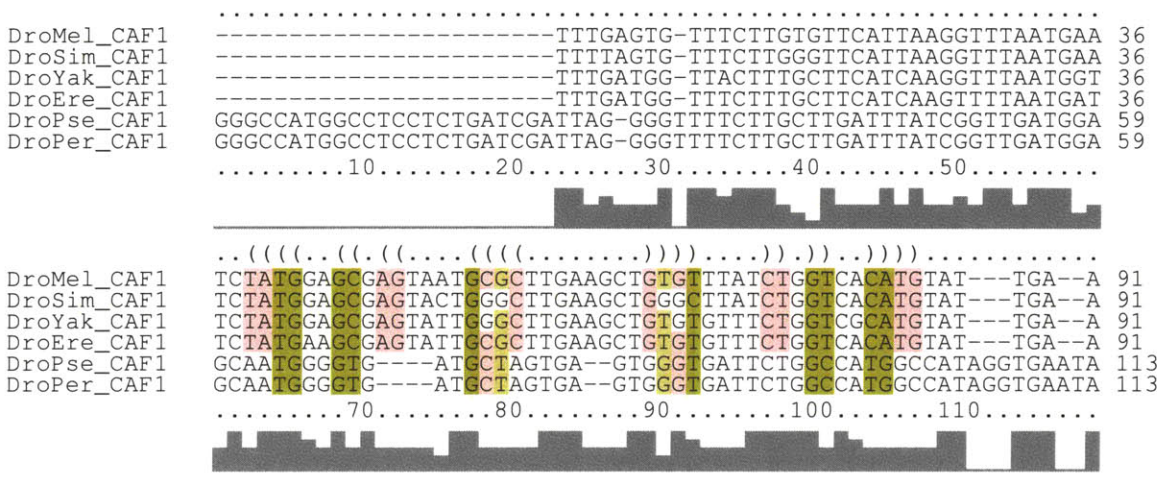


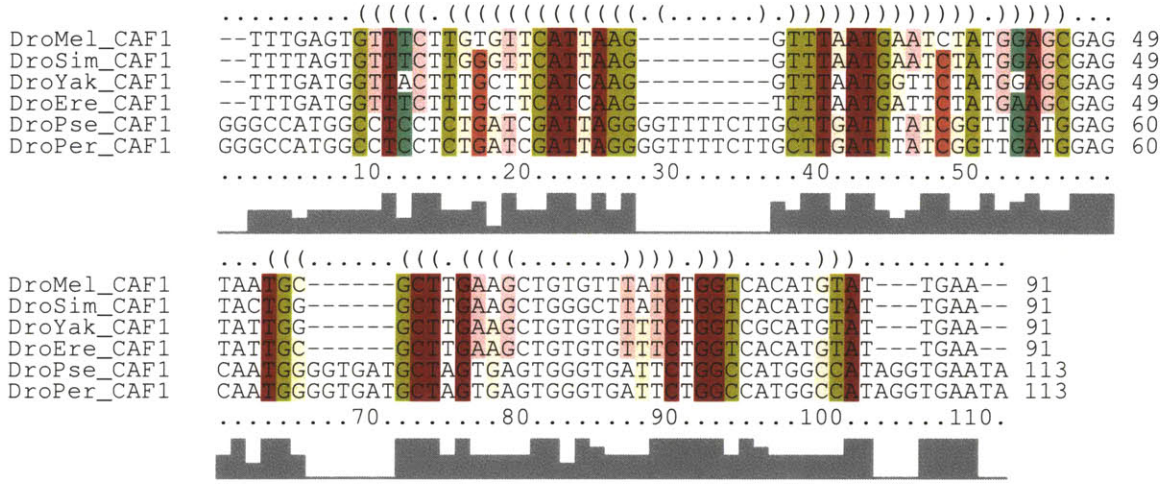
Figure 4-5: Distribution of sequence identity. The number of loci falling within fixed intervals of sequence identities is shown in blue, and the subset of those loci with more than two sequences is shown in light green. The distribution of new hits is shifted towards lower identity levels than the loci discovered before realignment in the original WGA ("All hits before"). Loci with only two sequences had a noticeably different distribution, especially in Fly where they account for most of the high sequence identity instances. Even when filtering out these loci, the shift in the distributions is still apparent.



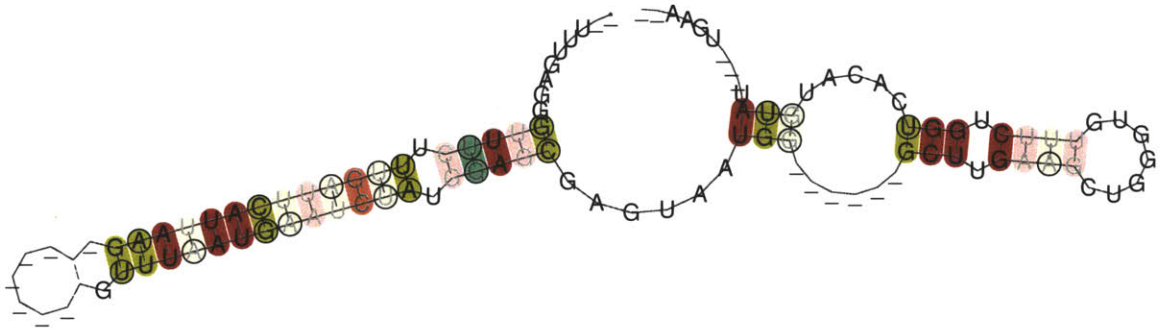
(a) Alignment

(b) Consensus secondary structure

Figure 4-6: Example locus in the original Fly whole genome alignment

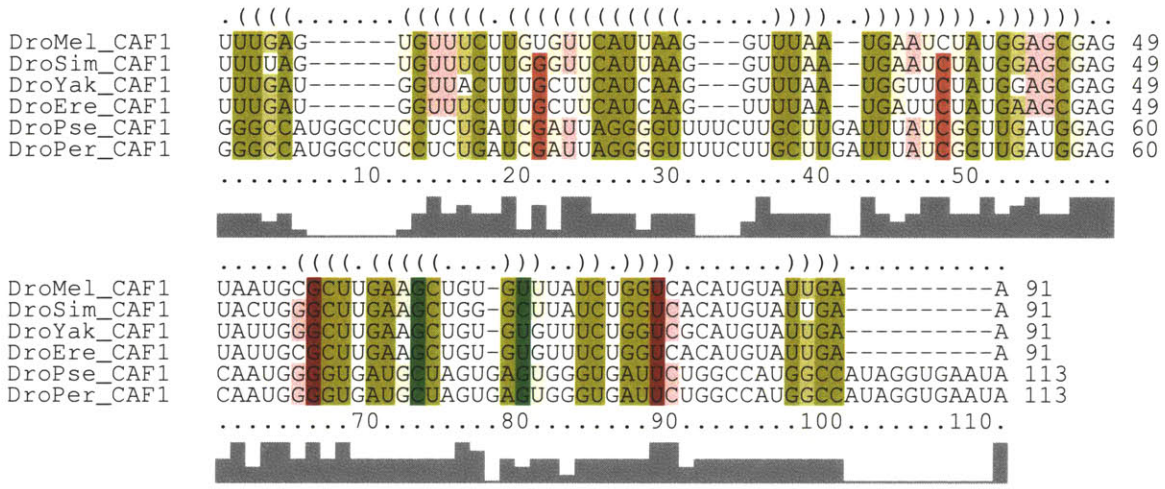


(c) Alignment

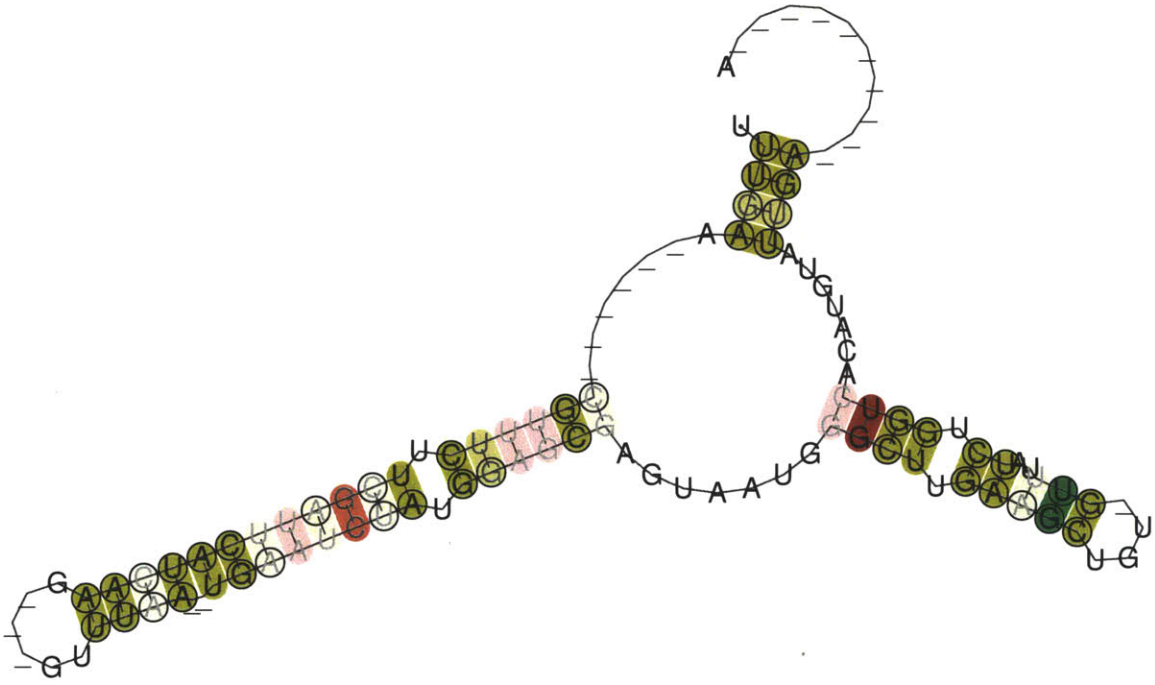


(d) Consensus secondary structure

Figure 4-6: Example locus realigned with Muscle



(e) Alignment



(f) Consensus secondary structure

Figure 4-6: Example locus realigned with LocARNA constrained at $\Delta = 20$

Chapter 5

Discussion

We have demonstrated the first genome-wide means for realigning ncRNAs according to sequence and structural similarity. We introduced two techniques that make our realignment pipeline computationally feasible. First, the stability filter removes windows with sequences that are probably too unstable to form structure. Second, constructing an alignment with limited deviation from a reference alignment by constraining the dynamic programming search space is faster than creating an alignment from scratch. We use this technique to make structural alignment methods like LocARNA that are otherwise too expensive relative to sequence-similarity methods amenable for large-scale use.

Multiple levels of evidence in the Fly and Encode experiments show that our pipeline enhances *de novo* ncRNA prediction capabilities and addresses the misalignment of ncRNAs. Realignment of loci with Muscle or LocARNA reveals thousands of new ncRNA candidates predicted by RNAz in Fly and hundreds in Encode. FDR estimations show that the pipeline has a similar fraction of false predictions compared to just applying RNAz on the original WGA. Following the arguments of previous screens [26, 33] for the reliability of RNAz predictions, we claim that a significant fraction of the new candidates represent true ncRNAs, several of which in Fly are verified by known annotations. The sequence identity distribution of the new candidates with respect to the original WGA are also lower than the candidates found from the original WGA. Sequences with lower identity are inherently more difficult

to align so as to reveal underlying structural conservation. Therefore, the discovery of a new candidate after realignment suggests that it was originally misaligned.

In addition to speed improvements, our framework for constrained realignment also has the advantage of ensuring that only conservative changes are made to the original WGA. All though our pipeline is careful in patching the original WGA by replacing only sites predicted to contain ncRNAs, it still suffers from a high FDR rate found in all *de novo* prediction studies [17]. In such cases the structural realignment of a locus may actually be more inaccurate than its original alignment. By controlling the realignment deviation Δ, we limit the extent to which patching has the opposite effect of misalignment.

In all of the above measurements, LocARNA outperforms Muscle as a realignment method for the pipeline. As both LocARNA and Muscle optimize for sequence similarity, we believe that the extra leverage of LocARNA stems mainly from the explicit additional optimization for structural similarity rather than minor details of the alignment algorithm such as gap and substitution costs and guide tree construction. This feature is neither present in the Pecan tool used in the Fly and Encode alignments here nor in other whole genome alignment methods. Structural alignment is therefore important for the accurate alignment and *de novo* prediction of ncRNAs in WGAs.

Chapter 6

Methods

6.1 Genome alignments

The alignment of 12 fly genomes was the same used by [26]. The genome sequences were taken from the Comparative Analysis Freeze 1 (CAF1) genome assemblies which were compiled by the *Drosophila* Twelve Genomes Consortium [1, 9] and include Release 4 of *D. melanogaster*. The alignment was constructed with Mercator [10, 11] to identify syntenic blocks and Pecan [23] to align each block. The alignment was originally downloaded from the site `http://www.sanger.ac.uk/Users/td2/pecan-CAF1`; however, we were unable to locate an existing download mirror.

We used the alignment of 28 vertebrates to 1% of the human genome selected by the Encode project [19]. The alignment was downloaded from `http://www.ebi.ac.uk/~bjp/pecan/encode_sept_pecan_mfas_proj.tar.bz2`.

6.2 Annotations

Annotations of known ncRNAs were obtained from Flybase Release 5.36 [31] and Rfam 10.0 [15]. The genomic position of the annotations are referenced with respect to Release 5 of the *D. melanogaster* genome. The coordinates were converted to Release 4, based on the MAPPING.SQL file of the BDGP Release 5 notes, in order to directly identify positional overlaps between loci and annotations. Annotations that

could not be mapped into exactly one assembly block in Release 4 were removed from consideration.

6.3 Alignment tools

LocARNA can be downloaded from `http://www.bioinf.uni-freiburg.de/Software/LocARNA/`. Constrained realignment can be enabled with the '–max-diff' and '–max-diff-aln' options.

Muscle was downloaded `http://www.drive5.com/muscle/`. The default settings were used.

The Alistat tool for calculating the average pairwise sequence identity was downloaded from the squid package [13] at `http://selab.janelia.org/software.html`. Compalignp [36] (`http://www.biophys.uni-duesseldorf.de/bralibase/`) is a "paranoid" version of the squid package's Compalign tool. It computes the same functions but makes extra checks that input sequences are in the correct order.

RNAalifold [4] was downloaded from `http://www.tbi.univie.ac.at/~ivo/RNA/` as part of the Vienna RNA package 2.0. The options '-old', '-color', '-p' were used to generate Figure 4-6.

6.4 RNAz package

RNAz 2.0[3] and its accompanying Perl scripts including rnazWindow.pl was downloaded from `http://www.tbi.univie.ac.at/~wash/RNAz/`. RNAz was ran with the '-d' option to use the improvements in using dinucleotide shuffling for calculating MFE z-scores. RNAz's default setting is to use the sequence-similarity trained model, and the '-l' option was included to use the structural-similarity trained model. In the pipeline described in Chapter 2, rnazWindow.pl was ran with the '–no-reference' option in order to filter sequences independently according to gap content and nucleotide content. No limit was imposed by rnazWindow.pl on the sequence identity of a sequence or the maximum number of sequences in a window by setting the parameters

'–min-id=0' and '–max-seqs=100', respectively.

6.5 Estimating the false discovery rate

An inherent challenge in measuring the false discovery rate (FDR) of *de novo* ncRNA prediction tools like RNAz is the lack of a set of sequences that are established as true negatives, i.e. sequences that do not contain structural ncRNAs. Previous RNAz studies [3, 26] have worked around this problem by generating a set of randomized alignments assumed to contain no structural conservation, and hence are suitable as a set of negatives. For each window sliced from a genome alignment, they generated a randomized variation of the window that preserved local base-composition, gap pattern, and conservation pattern while presumably not containing any structural conservation. We followed a similar idea of randomizing alignment to estimate the FDR of predictions from our realignment pipeline. First, we constructed a randomized variation of the input WGA. We took the set of windows returned by step (1) of the pipeline, skipped the stability filter and immediately merged and reassembled the windows in step (3). We shall refer to the re-assemblies simply as "blocks" instead of "loci" to avoid confusion with terminology in the rest of this section. We skipped the stability filter in step (2) because the windows passing the filter would be biased towards local base compositions and conservation patterns that are more likely to be associated with conserved structure. Hence such a set of windows is not representative of the sequence features of the original WGA. Through visual inspection, we found that the blocks were in general longer than the loci formed by normal execution of the pipeline because the removal of windows by the stability filter results in a more discontinuous set of windows. For each locus, we randomized non-overlapping windows of length 120 nt starting from the lowest column position. If the length of a locus is not a multiple of 120, then the window in the last 120 columns was also randomized. Following [3], a window was randomized by the shuffling of columns with MULTIPERM [2] if the window's entropy is below 0.5 or by re-sampling nucleotides with SISSIz [16] if the entropy is ≥ 0.5. In this way, we generate a set of randomized

loci. By considering each locus as a sytenic block of the genomes, the blocks form a randomized variation of original WGA that we treat as negative set of ncRNAs. Finally, we run the entire realignment pipeline on the blocks as normal and count the predictions.

By definition, the FDR is the expected ratio of false negatives, i.e. a site is predicted as an RNAz hit but does not actually contain ncRNAs, to the total set of predictions. For a given RNAz p score threshold, we will estimate

$$FDR = \Pr[\text{true negative}|\text{predict positive}] \tag{6.1}$$

$$= \frac{\Pr[\text{predict positive}|\text{true negative}]\,\Pr[\text{true negative}]}{\Pr[\text{predict positive}]} \tag{6.2}$$

where the second equation follows from Bayes's rule. Letting $\Pr[\text{true negative}] \leq 1$, we have the upper bound

$$FDR = \frac{\Pr[\text{predict positive}|\text{true negative}]}{\Pr[\text{predict positive}]}$$

From the execution of the pipeline on the original WGA, we calculate the ratio of the number of windows composing the loci after step (6) and are RNAz hits to the total number of windows considered after step (1). This gives an estimate of $Pr[\text{predict positive}]$. To estimate $\Pr[\text{predict positive}|\text{true negative}]$, we calculate the same ratio but over the execution of the pipeline on the randomized blocks. Note that windows rather than loci were counted in these ratios because the aforementioned sampling bias of loci due to the stability filter. By counting windows, we estimate the FDR of the entire pipeline rather than the FDR conditioned on stable loci.

Bibliography

[1] Assembly/alignment/annotation of 12 related drosophila species. *http://rana.lbl.gov/drosophila/*.

[2] P. Anandam, E. Torarinsson, and W. L. Ruzzo. Multiperm: shuffling multiple sequence alignments while approximately preserving dinucleotide frequencies. *Bioinformatics*, 2009.

[3] Stephan Washietl Ivo L. Hofacker Andreas R. Gruber, Sven Findeiss and Peter F. Stadler. RNAZ 2.0: IMPROVED NONCODING RNA DETECTION. In *PSB10*, volume 15, pages 69–79, 2010.

[4] Stephan Bernhart, Ivo Hofacker, Sebastian Will, Andreas Gruber, and Peter Stadler. Rnaalifold: improved consensus structure prediction for rna alignments. *BMC Bioinformatics*, 9(1):474, 2008.

[5] Mathieu Blanchette, W. James Kent, Cathy Riemer, Laura Elnitski, Arian F.A. Smit, Krishna M. Roskin, Robert Baertsch, Kate Rosenbloom, Hiram Clawson, Eric D. Green, David Haussler, and Webb Miller. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Research*, 14(4):708–715, 2004.

[6] Nicolas Bray and Lior Pachter. Mavid: Constrained ancestral alignment of multiple sequences. *Genome Research*, 14(4):693–699, 2004.

[7] Michael Brudno, Chuong B. Do, Gregory M. Cooper, Michael F. Kim, Eugene Davydov, NISC Comparative Sequencing Program, Eric D. Green, Arend Sidow, and Serafim Batzoglou. Lagan and multi-lagan: Efficient tools for large-scale multiple alignment of genomic dna. *Genome Research*, 13(4):721–731, 2003.

[8] Xiaoyu Chen and Martin Tompa. Comparative assessment of methods for aligning multiple genome sequences. *Nature Biotechnology*, 28(6):567–572, May 2010.

[9] Drosophila 12 Genomes Consortium. Evolution of genes and genomes on the drosophila phylogeny. *Nature*, 450(7167):203–218, 2007.

[10] Colin N. Dewey. *Whole-genome alignments and polytopes for comparative genomics.* PhD thesis, University of California, Berkeley, 2006.

[11] Colin N. Dewey. Aligning multiple whole genomes with Mercator and MAVID. *Methods Mol Biol*, 395:221–36, 2007.

[12] C Do, M Mahabhashyam, M Brudno, and S Batzoglou. Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15:330–340, 2005.

[13] Sean Eddy. Squid - c function library for sequence analysis. 2005.

[14] R Edgar. Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5:113, 2004.

[15] Paul P. Gardner, Jennifer Daub, John G. Tate, Eric P. Nawrocki, Diana L. Kolbe, Stinus Lindgreen, Adam C. Wilkinson, Robert D. Finn, Sam Griffiths-Jones, Sean R. Eddy, and Alex Bateman. Rfam: updates to the RNA families database. *Nucleic Acids Research*, 37(Database issue):D136–40, 2009.

[16] Tanja Gesell and Stephan Washietl. Dinucleotide controlled null models for comparative RNA gene prediction. *BMC Bioinformatics*, 9:248, 2008.

[17] Jan Gorodkin, Ivo L. Hofacker, Elfar Torarinsson, Zizhen Yao, Jakob H. Havgaard, and Walter L. Ruzzo. De novo prediction of structured rnas from genomic sequences. *Trends in Biotechnology*, 28(1):9 – 19, 2010.

[18] I. L. Hofacker, S. H. Bernhart, and P. F. Stadler. Alignment of RNA base pairing probability matrices. *Bioinformatics*, 20(14):2222–7, 2004.

[19] Elliott H. Margulies, Gregory M. Cooper, George Asimenos, Daryl J. Thomas, Colin N. Dewey, Adam Siepel, Ewan Birney, Damian Keefe, Ariel S. Schwartz, Minmei Hou, James Taylor, Sergey Nikolaev, Juan I. Montoya-Burgos, Ari Lytynoja, Simon Whelan, Fabio Pardi, Tim Massingham, James B. Brown, Peter Bickel, Ian Holmes, James C. Mullikin, Abel Ureta-Vidal, Benedict Paten, Eric A. Stone, Kate R. Rosenbloom, W. James Kent, Gerard G. Bouffard, Xiaobin Guan, Nancy F. Hansen, Jacquelyn R. Idol, Valerie V.B. Maduro, Baishali Maskeri, Jennifer C. McDowell, Morgan Park, Pamela J. Thomas, Alice C. Young, Robert W. Blakesley, Donna M. Muzny, Erica Sodergren, David A. Wheeler, Kim C. Worley, Huaiyang Jiang, George M. Weinstock, Richard A. Gibbs, Tina Graves, Robert Fulton, Elaine R. Mardis, Richard K. Wilson, Michele Clamp, James Cuff, Sante Gnerre, David B. Jaffe, Jean L. Chang, Kerstin Lindblad-Toh, Eric S. Lander, Angie Hinrichs, Heather Trumbower, Hiram Clawson, Ann Zweig, Robert M. Kuhn, Galt Barber, Rachel Harte, Donna Karolchik, Matthew A. Field, Richard A. Moore, Carrie A. Matthewson, Jacqueline E. Schein, Marco A. Marra, Stylianos E. Antonarakis, Serafim Batzoglou, Nick Goldman, Ross Hardison, David Haussler, Webb Miller, Lior Pachter, Eric D. Green, and Arend Sidow. Analyses of deep mammalian sequence alignments and constraint predictions for 1 *Genome Research*, 17(6):760–774, 2007.

[20] David H. Mathews and Douglas H. Turner. Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *Journal of Molecular Biology*, 317(2):191–203, 2002.

[21] Cdric Notredame, Desmond G. Higgins, and Jaap Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205 – 217, 2000.

[22] Wolfgang Otto, Sebastian Will, and Rolf Backofen. Structure local multiple alignment of RNA. In *Proceedings of German Conference on Bioinformatics (GCB'2008)*, volume P-136 of *Lecture Notes in Informatics (LNI)*, pages 178–188. Gesellschaft für Informatik (GI), 2008.

[23] Benedict Paten, Javier Herrero, Kathryn Beal, Stephen Fitzgerald, and Ewan Birney. Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs. *Genome Res*, 18(11):1814–28, 2008.

[24] Jakob Skou Pedersen, Gill Bejerano, Adam Siepel, Kate Rosenbloom, Kerstin Lindblad-Toh, Eric S Lander, Jim Kent, Webb Miller, and David Haussler. Identification and classification of conserved rna secondary structures in the human genome. *PLoS Comput Biol*, 2(4):e33, 04 2006.

[25] Amol Prakash and Martin Tompa. Measuring the accuracy of genome-size multiple alignments. *Genome Biology*, 8(6):R124, 2007.

[26] Dominic Rose, Jorg Hackermuller, Stefan Washietl, Kristin Reiche, Jana Hertel, Sven Findeiss, Peter F. Stadler, and Sonja J. Prohaska. Computational RNomics of drosophilids. *BMC Genomics*, 8:406, 2007.

[27] David Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45(5):810–825, 1985.

[28] Tobias Straub and Peter Becker. Dosage compensation: the beginning and end of generalization. *Nat. Rev. Genet.*, 8:47–57, Jan 2007.

[29] J Thompson, D Higgins, and T Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl Acids Res*, 22:4673–4680, 1994.

[30] E. Torarinsson, Z. Yao, E. D. Wiklund, J. B. Bramsen, C. Hansen, J. Kjems, N. Tommerup, W. L. Ruzzo, and J. Gorodkin. Comparative genomics beyond sequence-based alignments: RNA structures in the ENCODE regions. *Genome Res.*, 18:242–251, Feb 2008.

[31] S. Tweedie, M. Ashburner, K. Falls, P. Leyland, P. McQuilton, S. Marygold, G. Millburn, D. Osumi-Sutherland, A. Schroeder, R. Seal, H. Zhang, W. Gelbart, L. Bitsoi, M. Crosby, A. Dirkmaat, D. Emmert, L. S. Gramates, K. Falls, R. Kulathinal, B. Matthews, M. Roark, S. Russo, A. Schroeder, S. St Pierre, H. Zhang, P. Zhou, M. Zytkovicz, M. Ashburner, N. Brown, P. Leyland, P. McQuilton, S. Marygold, G. Millburn, D. Osumi-Sutherlan, R. Stefancsik, S. Tweedie, M. Williams, T. Kaufman, K. Matthews, J. Goodman, G. Grumbling, V. Strelets, and R. Wilson. FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucleic Acids Res.*, 37:D555–559, Jan 2009.

[32] Stefan Washietl, Ivo L. Hofacker, and Peter F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA*, 102(7):2454–9, 2005.

[33] Stefan Washietl, Jakob S. Pedersen, Jan O. Korbel, Claudia Stocsits, Andreas R. Gruber, Jorg Hackermuller, Jana Hertel, Manja Lindemeyer, Kristin Reiche, Andrea Tanzer, Catherine Ucla, Carine Wyss, Stylianos E. Antonarakis, France Denoeud, Julien Lagarde, Jorg Drenkow, Philipp Kapranov, Thomas R. Gingeras, Roderic Guigo, Michael Snyder, Mark B. Gerstein, Alexandre Reymond, Ivo L. Hofacker, and Peter F. Stadler. Structured RNAs in the ENCODE selected regions of the human genome. *Genome Res.*, 17(6):852–864, June 2007.

[34] Zasha Weinberg, Jeffrey E. Barrick, Zizhen Yao, Adam Roth, Jane N. Kim, Jeremy Gore, Joy Xin Wang, Elaine R. Lee, Kirsten F. Block, Narasimhan Sudarsan, Shane Neph, Martin Tompa, Walter L. Ruzzo, and Ronald R. Breaker. Identification of 22 candidate structured rnas in bacteria using the cmfinder comparative genomics pipeline. *Nucleic Acids Research*, 35(14):4809–4819, 2007.

[35] Sebastian Will, Kristin Reiche, Ivo L. Hofacker, Peter F. Stadler, and Rolf Backofen. Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. *PLOS Computational Biology*, 3(4):e65, 2007.

[36] Andreas Wilm, Indra Mainz, and Gerhard Steger. An enhanced rna alignment benchmark for sequence alignment programs. *Algorithms for Molecular Biology*, 1(1):19, 2006.