

BOSTON COLLEGE

DOCTORAL THESIS

On the use of Coarse Grained
Thermodynamic Landscapes to
Efficiently Estimate Kinetic Pathways
for RNA Molecules

Author:

Evan SENTER

Supervisor:

Dr. Peter CLOTE

*A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy*

in the

Clote Lab

Department of Biology

Sunday 7th June, 2015

Declaration of Authorship

I, Evan SENTER, declare that this thesis titled, 'On the use of Coarse Grained Thermodynamic Landscapes to Efficiently Estimate Kinetic Pathways for RNA Molecules' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

BOSTON COLLEGE

Abstract

Dr. Peter Clote
Department of Biology

Doctor of Philosophy

**On the use of Coarse Grained Thermodynamic Landscapes to Efficiently
Estimate Kinetic Pathways for RNA Molecules**

by Evan SENTER

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

Declaration of Authorship	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	vi
List of Tables	vii
1 Chapter Title Here	1
1.1 Welcome and Thank You	1
1.2 Learning L ^A T _E X	1
1.2.1 A (not so short) Introduction to L ^A T _E X	2
1.2.2 A Short Math Guide for L ^A T _E X	2
1.2.3 Common L ^A T _E X Math Symbols	2
1.2.4 L ^A T _E X on a Mac	3
1.3 Getting Started with this Template	3
1.3.1 About this Template	3
1.4 What this Template Includes	4
1.4.1 Folders	4
1.4.2 Files	5
1.5 Filling in the ‘Thesis.cls’ File	6
1.6 The ‘Thesis.tex’ File Explained	7
1.7 Thesis Features and Conventions	8
1.7.1 Printing Format	8
1.7.2 Using US Letter Paper	8
1.7.3 References	9
1.7.4 Figures	10
1.7.5 Typesetting mathematics	11
1.8 Sectioning and Subsectioning	12
1.9 In Closing	12
2 Ribofinder	13

2.1	Pipeline	13
2.1.1	Candidate Selection	13
2.1.1.1	Detecting Aptamers with Infernal	14
2.1.1.2	Detecting Expression Platforms with TransTermHP	14
2.1.2	Notation for Representing Abstract RNA Shapes	15
2.1.3	Coarse-Grained Filtering	16
2.1.3.1	Constrained Folding to Predict Switch Structures	16
2.1.4	Fine-Grained Filtering	18
2.1.4.1	Using FoldAlign to Rank Candidates	18

Bibliography**19**

List of Figures

1.1	An Electron	10
2.1	Histogram displaying the distribution of genome sizes across the RefSeq data analyzed, comprising 5,172 bacterial genomes. Genome size is shown using a \log_{10} scale.	15
2.2	<i>Left:</i> the computationally predicted gene-off conformation of sequence NC_014019.1 1427313-1427501, using <code>RNAfold</code> from the ViennaRNA 2.1.8 suite, with dangles disabled and the Turner 1999 energies. This sequence is located upstream of the <code>xpt</code> gene in <i>Bacillus megaterium</i> QM B1551. <i>Right:</i> the gene-on conformation.	18

List of Tables

Chapter 1

Chapter Title Here

1.1 Welcome and Thank You

Welcome to this L^AT_EX Thesis Template, a beautiful and easy to use template for writing a thesis using the L^AT_EX typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L^AT_EX is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L^AT_EX is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L^AT_EX to make them look stunning.

1.2 Learning L^AT_EX

L^AT_EX is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L^AT_EX is actually a simple, plain text file that contains *no formatting*. You tell L^AT_EX how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the

`\textit{}` command and put the text I want in italics in between the curly braces. This means that L^AT_EX is a “mark-up” language, very much like HTML.

1.2.1 A (not so short) Introduction to L^AT_EX

If you are new to L^AT_EX, there is a very good eBook – freely available online as a PDF file – called, “The Not So Short Introduction to L^AT_EX”. The book’s title is typically shortened to just “lshort”. You can download the latest version (as it is occasionally updated) from here:

<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page:

<http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L^AT_EX by creating several, small ‘test’ documents. Making the effort now means you’re not stuck learning the system when what you *really* need to be doing is writing your thesis.

1.2.2 A Short Math Guide for L^AT_EX

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, “A Short Math Guide for L^AT_EX”. It can be found online here:

<http://www.ams.org/tex/amslatex.html>

under the “Additional Documentation” section towards the bottom of the page.

1.2.3 Common L^AT_EX Math Symbols

There are a multitude of mathematical symbols available for L^AT_EX and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page:

<http://www.sunilpatel.co.uk/latexsymbols.html>

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the L^AT_EX command for the symbol you need.

1.2.4 L^AT_EX on a Mac

The L^AT_EX package is available for many systems including Windows, Linux and Mac OS X. The package for OS X is called MacTeX and it contains all the applications you need – bundled together and pre-customised – for a fully working L^AT_EX environment and workflow.

MacTeX includes a dedicated L^AT_EX IDE (Integrated Development Environment) called “TeXShop” for writing your ‘.tex’ files and “BibDesk”: a program to manage your references and create your bibliography section just as easily as managing songs and creating playlists in iTunes.

1.3 Getting Started with this Template

If you are familiar with L^AT_EX, then you can familiarise yourself with the contents of the Zip file and the directory structure and then place your own information into the ‘Thesis.cls’ file. Section 1.5 on page 6 tells you how to do this. Make sure you read section 1.7 about thesis conventions to get the most out of this template and then get started with the ‘Thesis.tex’ file straightaway.

If you are new to L^AT_EX it is recommended that you carry on reading through the rest of the information in this document.

1.3.1 About this Template

This L^AT_EX Thesis Template is originally based and created around a L^AT_EX style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his original thesis style file at his site, here:

<http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/>

My thesis originally used the ‘ecsthesis.cls’ from his list of styles. However, I knew L^AT_EX could still format better. To get the look I wanted, I modified his style and also created a skeleton framework and folder structure to place the thesis files in.

This Thesis Template consists of that modified style, the framework and the folder structure. All the work that has gone into the preparation and groundwork means that all you have to bother about is the writing.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations.

1.4 What this Template Includes

1.4.1 Folders

This template comes as a single Zip file that expands out to many files and folders. The folder names are mostly self-explanatory:

Appendices – this is the folder where you put the appendices. Each appendix should go into its own separate `.tex` file. A template is included in the directory.

Chapters – this is the folder where you put the thesis chapters. A thesis usually has about seven chapters, though there is no hard rule on this. Each chapter should go in its own separate `.tex` file and they usually are split as:

- Chapter 1: Introduction to the thesis topic
- Chapter 2: Background information and theory
- Chapter 3: (Laboratory) experimental setup
- Chapter 4: Details of experiment 1
- Chapter 5: Details of experiment 2
- Chapter 6: Discussion of the experimental results
- Chapter 7: Conclusion and future directions

This chapter layout is specialised for the experimental sciences.

Figures – this folder contains all figures for the thesis. These are the final images that will go into the thesis document.

Primitives – this is the folder that contains scraps, particularly because one final image in the 'Figures' folder may be made from many separate images and photos, these source images go here. This keeps the intermediate files separate from the final thesis figures.

1.4.2 Files

Included are also several files, most of them are plain text and you can see their contents in a text editor. Luckily, many of them are auxiliary files created by L^AT_EX or BibT_EX and which you don't need to bother about:

Bibliography.bib – this is an important file that contains all the bibliographic information and references that you will be citing in the thesis for use with BibT_EX. You can write it manually, but there are reference manager programs available that will create and manage it for you. Bibliographies in L^AT_EX are a large subject and you may need to read about BibT_EX before starting with this.

Thesis.cls – this is an important file. It is the style file that tells L^AT_EX how to format the thesis. You will also need to open this file in a text editor and fill in your own information (such as name, department, institution). Luckily, this is not too difficult and is explained in section 1.5 on page 6.

Thesis.pdf – this is your beautifully typeset thesis (in the PDF file format) created by L^AT_EX.

Thesis.tex – this is an important file. This is the file that you tell L^AT_EX to compile to produce your thesis as a PDF file. It contains the framework and constructs that tell L^AT_EX how to layout the thesis. It is heavily commented so you can read exactly what each line of code does and why it is there. After you put your own information into the '**Thesis.cls**' file, go to this file and begin filling it in – you have now started your thesis!

vector.sty – this is a L^AT_EX package, it tells L^AT_EX how to typeset mathematical vectors. Using this package is very easy and you can read the documentation on the site (you just need to look at the '**vector.pdf**' file):

<http://www.ctan.org/tex-archive/macros/latex/contrib/vector/>

lstpatch.sty – this is a L^AT_EX package required by this LaTeX template and is included as not all T_EX distributions have it installed by default. You do not need to modify this file.

Files that are *not* included, but are created by L^AT_EX as auxiliary files include:

Thesis.aux – this is an auxiliary file generated by L^AT_EX, if it is deleted L^AT_EX simply regenerates it when you run the main '**.tex**' file.

Thesis.bbl – this is an auxiliary file generated by BibT_EX, if it is deleted, BibT_EX simply regenerates it when you run the main tex file. Whereas the '**.bib**' file contains

all the references you have, this ‘.bbl’ file contains the references you have actually cited in the thesis and is used to build the bibliography section of the thesis.

Thesis.blg – this is an auxiliary file generated by BibTeX, if it is deleted BibTeX simply regenerates it when you run the main ‘.tex’ file.

Thesis.lof – this is an auxiliary file generated by L^AT_EX, if it is deleted L^AT_EX simply regenerates it when you run the main ‘.tex’ file. It tells L^AT_EX how to build the ‘List of Figures’ section.

Thesis.log – this is an auxiliary file generated by L^AT_EX, if it is deleted L^AT_EX simply regenerates it when you run the main ‘.tex’ file. It contains messages from L^AT_EX, if you receive errors and warnings from L^AT_EX, they will be in this ‘.log’ file.

Thesis.lot – this is an auxiliary file generated by L^AT_EX, if it is deleted L^AT_EX simply regenerates it when you run the main ‘.tex’ file. It tells L^AT_EX how to build the ‘List of Tables’ section.

Thesis.out – this is an auxiliary file generated by L^AT_EX, if it is deleted L^AT_EX simply regenerates it when you run the main ‘.tex’ file.

So from this long list, only the files with the ‘.sty’, ‘.bib’, ‘.cls’ and ‘.tex’ extensions are the most important ones. The other auxiliary files can be ignored or deleted as L^AT_EX and BibTeX will regenerate them.

1.5 Filling in the ‘Thesis.cls’ File

You will need to personalise the thesis template and make it your own by filling in your own information. This is done by editing the ‘Thesis.cls’ file in a text editor.

Open the file and scroll down, past all the ‘\newcommand...’ items until you see the entries for ‘University Name’, ‘Department Name’, etc....

Fill out the information about your group and institution and ensure you keep to block capitals where it asks you to. You can also insert web links, if you do, make sure you use the full URL, including the ‘http://’ for this.

The last item you should need to fill in is the Faculty Name (in block capitals). When you have done this, save the file and recompile ‘Thesis.tex’. All the information you filled in should now be in the PDF, complete with web links. You can now begin your thesis proper!

1.6 The ‘Thesis.tex’ File Explained

The `Thesis.tex` file contains the structure of the thesis. There are plenty of written comments that explain what pages, sections and formatting the \LaTeX code is creating. Initially there seems to be a lot of \LaTeX code, but this is all formatting, and it has all been taken care of so you don’t have to do it.

Begin by checking that your information on the title page is correct. For the thesis declaration, your institution may insist on something different than the text given. If this is the case, just replace what you see with what is required.

Then comes a page which contains a funny quote. You can put your own, or quote your favourite scientist, author, person, etc. . . Make sure to put the name of the person who you took the quote from.

Next comes the acknowledgements. On this page, write about all the people who you wish to thank (not forgetting parents, partners and your advisor/supervisor).

The contents pages, list of figures and tables are all taken care of for you and do not need to be manually created or edited. The next set of pages are optional and can be deleted since they are for a more technical thesis: insert a list of abbreviations you have used in the thesis, then a list of the physical constants and numbers you refer to and finally, a list of mathematical symbols used in any formulae. Making the effort to fill these tables means the reader has a one-stop place to refer to instead of searching the internet and references to try and find out what you meant by certain abbreviations or symbols.

The list of symbols is split into the Roman and Greek alphabets. Whereas the abbreviations and symbols ought to be listed in alphabetical order (and this is *not* done automatically for you) the list of physical constants should be grouped into similar themes.

The next page contains a one line dedication. Who will you dedicate your thesis to?

Finally, there is the section where the chapters are included. Uncomment the lines (delete the ‘%’ character) as you write the chapters. Each chapter should be written in its own file and put into the ‘Chapters’ folder and named ‘`Chapter1`’, ‘`Chapter2`’, etc. . . Similarly for the appendices, uncomment the lines as you need them. Each appendix should go into its own file and placed in the ‘Appendices’ folder.

After the preamble, chapters and appendices finally comes the bibliography. The bibliography style (called ‘`unsrtnat`’) is used for the bibliography and is a fully featured

style that will even include links to where the referenced paper can be found online. Do not under estimate how grateful your reader will be to find that a reference to a paper is just a click away. Of course, this relies on you putting the URL information into the BibTeX file in the first place.

1.7 Thesis Features and Conventions

To get the best out of this template, there are a few conventions that you may want to follow.

One of the most important (and most difficult) things to keep track of in such a long document as a thesis is consistency. Using certain conventions and ways of doing things (such as using a Todo list) makes the job easier. Of course, all of these are optional and you can adopt your own method.

1.7.1 Printing Format

This thesis template is designed for single sided printing as most theses are printed and bound this way. This means that the left margin is always wider than the right (for binding). Four out of five people will now judge the margins by eye and think, “I never noticed that before.”

The headers for the pages contain the page number on the right side (so it is easy to flick through to the page you want) and the chapter name on the left side.

The text is set to 11 point and a line spacing of 1.3. Generally, it is much more readable to have a smaller text size and wider gap between the lines than it is to have a larger text size and smaller gap. Again, you can tune the text size and spacing should you want or need to. The text size can be set in the options for the ‘`\documentclass`’ command at the top of the ‘`Thesis.tex`’ file and the spacing can be changed by setting a different value in the ‘`\setstretch`’ commands (scattered throughout the ‘`Thesis.tex`’ file).

1.7.2 Using US Letter Paper

The paper size used in the template is A4, which is a common – if not standard – size in Europe. If you are using this thesis template elsewhere and particularly in the United States, then you may have to change the A4 paper size to the US Letter size. Unfortunately, this is not as simple as replacing instances of ‘`a4paper`’ with ‘`letterpaper`’.

This is because the final PDF file is created directly from the L^AT_EX source using a program called ‘pdfTeX’ and in certain conditions, paper size commands are ignored and all documents are created with the paper size set to the size stated in the configuration file for pdfTeX (called ‘pdftex.cfg’).

What needs to be done is to change the paper size in the configuration file for pdfTeX to reflect the letter size. There is an excellent tutorial on how to do this here:

http://www.physics.wm.edu/~norman/latexhints/pdf_papersize.html

It may be sufficient just to replace the dimensions of the A4 paper size with the US Letter size in the `pdftex.cfg` file. Due to the differences in the paper size, the resulting margins may be different to what you like or require (as it is common for Institutions to dictate certain margin sizes). If this is the case, then the margin sizes can be tweaked by opening up the `Thesis.cls` file and searching for the line beginning with, ‘\setmarginsrb’ (not very far down from the top), there you will see the margins specified. Simply change those values to what you need (or what looks good) and save. Now your document should be set up for US Letter paper size with suitable margins.

1.7.3 References

The ‘natbib’ package is used to format the bibliography and inserts references such as this one [1]. The options used in the ‘Thesis.tex’ file mean that the references are listed in numerical order as they appear in the text. Multiple references are rearranged in numerical order (e.g. [2, 3]) and multiple, sequential references become reformatted to a reference range (e.g. [1–3]). This is done automatically for you. To see how you use references, have a look at the ‘Chapter1.tex’ source file. Many reference managers allow you to simply drag the reference into the document as you type.

Scientific references should come *before* the punctuation mark if there is one (such as a comma or period). The same goes for footnotes¹. You can change this but the most important thing is to keep the convention consistent throughout the thesis. Footnotes themselves should be full, descriptive sentences (beginning with a capital letter and ending with a full stop).

To see how L^AT_EX typesets the bibliography, have a look at the very end of this document (or just click on the reference number links).

¹Such as this footnote, here down at the bottom of the page.

1.7.4 Figures

There will hopefully be many figures in your thesis (that should be placed in the ‘Figures’ folder). The way to insert figures into your thesis is to use a code template like this:

```
\begin{figure}[htbp]
  \centering
  \includegraphics{Figures/Electron.pdf}
  \rule{35em}{0.5pt}
  \caption[An Electron]{An electron (artist’s impression).}
  \label{fig:Electron}
\end{figure}
```

Also look in the source file. Putting this code into the source file produces the picture of the electron that you can see in the figure below.



FIGURE 1.1: An electron (artist’s impression).

Sometimes figures don’t always appear where you write them in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so \LaTeX puts it at the top of the next page. Positioning figures is the job of \LaTeX and so you should only worry about making them look good!

Figures usually should have labels just in case you need to refer to them (such as in Figure 1.1). The ‘`\caption`’ command contains two parts, the first part, inside the square brackets is the title that will appear in the ‘List of Figures’, and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

The ‘`\rule`’ command is optional and simply puts an aesthetic horizontal line below the image. If you do this for one image, do it for all of them.

The L^AT_EX Thesis Template is able to use figures that are either in the PDF or JPEG file format.

1.7.5 Typesetting mathematics

If your thesis is going to contain heavy mathematical content, be sure that L^AT_EX will make it look beautiful, even though it won’t be able to solve the equations for you.

The “Not So Short Introduction to L^AT_EX” (available [here](#)) should tell you everything you need to know for most cases of typesetting mathematics. If you need more information, a much more thorough mathematical guide is available from the AMS called, “A Short Math Guide to L^AT_EX” and can be downloaded from:

<ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>

There are many different L^AT_EX symbols to remember, luckily you can find the most common symbols [here](#). You can use the web page as a quick reference or crib sheet and because the symbols are grouped and rendered as high quality images (each with a downloadable PDF), finding the symbol you need is quick and easy.

You can write an equation, which is automatically given an equation number by L^AT_EX like this:

```
\begin{equation}
E = mc^2
\label{eqn:Einstein}
\end{equation}
```

This will produce Einstein’s famous energy-matter equivalence equation:

$$E = mc^2 \tag{1.1}$$

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by L^AT_EX. If you don't want a particular equation numbered, just put the command, '`\nonumber`' immediately after the equation.

1.8 Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. L^AT_EX automatically builds a table of Contents by looking at all the '`\chapter{}`', '`\section{}`' and '`\subsection{}`' commands you write in the source.

The table of Contents should only list the sections to three (3) levels. A '`\chapter{}`' is level one (1). A '`\section{}`' is level two (2) and so a '`\subsection{}`' is level three (3). In your thesis it is likely that you will even use a '`\subsubsection{}`', which is level four (4). Adding all these will create an unnecessarily cluttered table of Contents and so you should use the '`\subsubsection*`' command instead (note the asterisk). The asterisk (*) tells L^AT_EX to omit listing the subsubsection in the Contents, keeping it clean and tidy.

1.9 In Closing

You have reached the end of this mini-guide. You can now rename or overwrite this pdf file and begin writing your own '`Chapter1.tex`' and the rest of your thesis. The easy work of setting up the structure and framework has been taken care of for you. It's now your job to fill it out!

Good luck and have lots of fun!

Guide written by —
Sunil Patel: www.sunilpatel.co.uk

Chapter 2

Ribofinder

2.1 Pipeline

At the time of our retrieval (2014-11-25 at 09:14), the RefSeq database hosted by NCBI comprised 5,121 complete bacterial genomes with corresponding genomic annotations. In order to both detect putative full riboswitches across this collection of data as well as filter the candidates down to a number tractable for experimental validation, we developed a novel pipeline which takes a three-tiered approach to candidate selection. Our approach is to *a)* identify a pool of candidate riboswitches across genomic data; *b)* perform a coarse-grained filtering of the candidate pool based on structural characteristics; and finally *c)* fine-grained curation of the candidates based on a collection of measures and pairwise similarity.

In the following discussion, we describe the application of **Ribofinder** to identify unannotated G-box purine riboswitches; guanine-sensing cis-regulatory elements which modulate the expression of genes involved in purine biosynthesis.

2.1.1 Candidate Selection

The RefSeq data we used for analysis contains 5,121 annotated bacterial genomes across 2,732 different organisms, totaling over $9.5 * 10^9$ bases. We used the program **Infernal** to determine the coordinates of putative aptamer structures within the RefSeq genomes, and **TransTermHP** to locate candidate rho-independent transcription terminators.

2.1.1.1 Detecting Aptamers with **Infernal**

Infernal uses a stochastic context-free grammar (SCFG) with a user-provided multiple sequence alignment (MSA) to efficiently scan genomic data for RNA homologs, taking into consideration both sequence and structural conservation. Using the purine aptamer MSA from RFam 12.0 (RF00167), **Infernal** (v1.1.1, default options) detects 1,537 significant hits having E-value ≤ 0.01 . Because **Infernal** leverages the concept of a ‘local end’—a large insertion or deletion in the alignment at reduced cost—it is possible for the software to return a significant hit whose aligned structure does not have the canonical three-way junction observed in all purine riboswitches. **Ribofinder** prunes these truncated **Infernal** hits by converting the alignment structure into a parse tree, and only permitting trees of sufficient complexity to contain a multiloop (described further in 2.1.2). The pyrimidine residue abutted next to the P1 stem in the J3-1 junction differentiates between guanine and adenine-sensing riboswitches by binding the complimentary purine ligand; for our interest in G-box riboswitches exclusively we require the presence of a cytosine at this residue. In total, using **Infernal** with these additional filters yields 1,280 G-box aptamers across 555 unique organisms (note: here and elsewhere I define a ‘unique organism’ as having a unique taxonomy ID).

2.1.1.2 Detecting Expression Platforms with **TransTermHP**

TransTermHP detects rho-independent terminators in bacterial genomes in a context-sensitive fashion by leveraging the protein annotations available in PTT data. These terminator sequences canonically have a stable hairpin loop structure immediately preceding a run of 5+ uracil residues, the combination of which causes the ribosomal machinery to stall and dissociate from the transcript. **TransTermHP** performs a genomic scan to determine candidate loci with this motif, and returns scored hits. The scoring system considers both structural homology and the genomic contextual information available in the PTT file. Across our collection of bacterial genomes acquired from NCBI RefSeq data, **TransTermHP** identified 2,752,469 rho-independent terminators using the default filters.

Due to the spatially-mediated structural regulation of purine riboswitches, whereby ligand interaction with the aptamer domain induces local structural rearrangement in the expression platform, we paired aptamers with corresponding terminators by minimizing the genomic distance, with an upper bound of 200 nucleotides between the end of the aptamer domain and start of the terminator. This approach yields 577 candidate riboswitches, 81 of which have multiple rho-independent terminators within range

of a putative aptamer produced by **Infernal**. For these, we simply pair the closest **TransTermHP** hit with the aptamer domain.

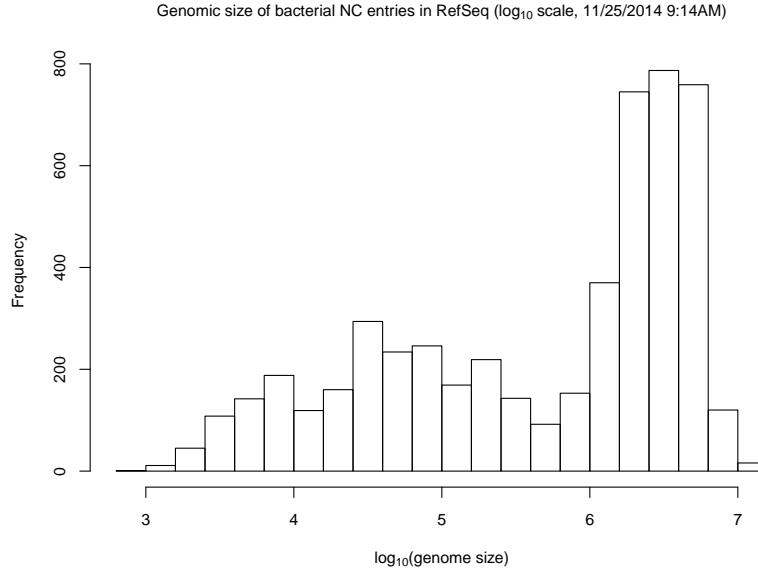


FIGURE 2.1: Histogram displaying the distribution of genome sizes across the RefSeq data analyzed, comprising 5,172 bacterial genomes. Genome size is shown using a \log_{10} scale.

2.1.2 Notation for Representing Abstract RNA Shapes

Given an RNA sequence $\mathbf{s} = a_1, a_2, \dots, a_n$, where positions a_i are drawn from the collection of single-letter nucleotide codes, i.e. $\{A, U, G, C\}$, it is possible to describe a corresponding secondary structure \mathcal{S} compatible with \mathbf{s} using the dot-bracket notation. In this notation, each nucleotide a_i has a corresponding state s_i , where s_i is denoted as a \cdot if unpaired and a $($ [resp. $)$] if the left [resp. right] base in a basepair. Given any two basepairs (i, j) and (k, l) in \mathcal{S} , then $i < k < j \iff i < l < j$; pseudoknots are not permitted in the structure. A secondary structure taking this form is said to have balanced parentheses, and can additionally be represented using a context-free grammar such as:

$$S \rightarrow S \cdot \mid \cdot S \mid (S) \mid SS \mid \epsilon \quad (2.1)$$

The grammar from (2.1) can be used to generate a parse tree \mathcal{T} for \mathcal{S} . The benefit of working with \mathcal{T} over \mathcal{S} is that the parse tree offers an abstract representation of secondary structure shape independent of sequence length, permitting us to classify and eventually constrain a large collection of sequences having variable length which are all

expected to have the same abstract tree shape. This is analogous to what the Giegerich lab refers to as their ‘type 5’ structural abstraction using the **RNAshapes** tool. Every node in \mathcal{T} represents a helix in \mathcal{S} , and internally tracks the indices of both its beginning (i, j) and closing (k, l) basepair. We use a level-order naming convention to refer to helices within the parse tree, whereby a position \mathbf{p}_1 references the first child of the root node, $\mathbf{p}_{1,2}$ references the second child of \mathbf{p}_1 , and generally $\mathbf{p}_{i_1, i_2, \dots, i_n}$ refers to the i_n^{th} child of $\mathbf{p}_{i_1, i_2, \dots, i_{n-1}}$. To reference specific nucleotides in the context of their location relative to a helix, we use the opening and closing basepairs (i, j) and (k, l) as landmarks. Thus, $\mathbf{p}_1(l)$ is the index in \mathcal{S} of the right-hand side closing basepair of \mathbf{p}_1 . We use the notation \mathbf{t}_i to refer to the subtree of \mathcal{T} whose root is \mathbf{p}_i .

Finally, we introduce the concept of a tree signature. The tree signature for a tree \mathcal{T} is a list of the node depths when traversed in a depth-first pre-order fashion. To provide a concrete example, consider the following experimentally validated xpt G-box riboswitch from *Bacillus subtilis* subsp. *subtilis* str. 168 (NC_000964.3 2320197-2320054) with corresponding gene-off structure:

```
ACACUCAUAUAUAUACGCGUGGAUAUAGGCACGCAAGUUUCUACCGGGCACCGUAAAUGUCCGACUAUGGGUGAGCAAUGGAACCGCACGUGUACGGUUUUUUUGUGAUUUCAGCAUUGCUUGCUCUUUAUUUGAGC
.(((((((.....((((.....)))))).....((((.....))))))..)))))).(((.(((((((.....))))))))......((((((((((((.....))))))
```

The **RNAshapes** ‘type 5’ representation for this structure is `[[[]][[]][[]]` (note the coalesced left bulge in the hairpin immediately downstream the closing multiloop stem, at helix \mathbf{p}_2) and the tree signature for this parse tree of the structure is `[0, 1, 2, 2, 1, 1]`.

2.1.3 Coarse-Grained Filtering

We leverage the notion of abstract structural filtering initially to ensure that all **Infernal** aptamer hits have a tree signature of `[0, 1, 2, 2]`, which represents a three-way junction, and that the binding site for the guanine ligand $\mathbf{p}_1(l-1) = \text{C}$. These filters, in combination with the proximal terminator hairpins produced by **TransTermHP** yield the aforementioned 577 candidate guanine riboswitches for which we then try to produce reasonable gene-on and off structures.

2.1.3.1 Constrained Folding to Predict Switch Structures

To restrict our search to unannotated G-box riboswitches, and further ensure that we are not re-detecting sequences based off the RFam covariance model provided to **Infernal**, we constrain our search to those RefSeq organisms not represented in the RFam seed

alignment. 503 of the 577 candidates, or 87.18% represent putative unannotated riboswitches not represented by RF00167.

The gene-off structure \mathcal{S}_{off} for a G-box riboswitch is the easier of the two to find computationally, since the terminator loop is exceptionally thermodynamically stable. In the gene-on conformation \mathcal{S}_{on} , the P1 stem of the multiloop partially dissociates and an anti-terminator loop forms between the region immediately 3' of the P1 stem and what was the left-hand side of the terminator loop. This truncated P1 stem, which closes the three-way junction in the aptamer, is exceptionally unstable based on present energy models available for structural folding, and requires special treatment to reconstitute in our final structures.

The software **RNAfold** (v2.1.8) allows for the folding of RNA molecules with ‘loose’ constraints. In this model of constrained folding, the resulting structure produced by the software guarantees not to explicitly invalidate any user-provided constraints, but does not guarantee all constraints will be satisfied in the resulting structure. For each of the candidate guanine riboswitches, having $\mathcal{T}_{\text{Infernal}}$ and $\mathcal{T}_{\text{TransTermHP}}$, we build the following constraint masks:

G-box gene-off constraint mask	G-box gene-on constraint mask
Prohibit basepairing upstream of $\mathbf{p}_1(i)$ and downstream of $\mathbf{p}_2(l)$.	
Force basepairs and unpaired regions in \mathbf{t}_1 , with the exception of \mathbf{p}_1 .	
Prohibit formation of \mathbf{p}_1 stem, which closes the three-way junction.	
Force basepairs and unpaired regions in \mathbf{t}_2 .	Require m nucleotides starting from $\mathbf{p}_1(l + 3)$ to pair to the right, where $m = \text{len}(\mathbf{p}_2)$, and require the left-hand side of the \mathbf{p}_2 helix to pair to the left. Disallow pairing downstream of $\mathbf{p}_2(j)$.

These constraint masks are run using the command-line flags `-d 0 -P rna.turner1999.par` to disable dangles and use the Turner 1999 energies respectively. Experimental evidence using inline probing suggests that the ‘on’ conformation of the G-box riboswitch has a reduced P1 stem length of 3 base pairs; in practice we were unable to force **RNAfold** to respect this constraint regardless of command-line options specified. For this reason we reconstitute the P1 stem in both structures after constrained folding, having length equivalent to it the **Infernal** P1 stem (resp. 3 basepairs) in the gene-off (resp. gene-on) structure.

This difficulty with **RNAfold** can be shown by using the constraint-produced structures as exhaustive constraints themselves. All unpaired nucleotides in \mathcal{S}_{off} and \mathcal{S}_{on} are notated

by a ‘**x**’ and all base pairs by ‘**C**’ for the 5’ and 3’ side of the pair respectively to form new constraints mask \mathcal{C}_{off} and \mathcal{C}_{on} , having all bases’ state explicitly specified. By refolding all 577 candidate sequences with \mathcal{C}_{off} and \mathcal{C}_{on} using the same options as before, only 463 (or 80.24%) of the resulting structures from \mathcal{C}_{off} have the tree signature prefix $[\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{2}, \mathbf{1}]$, and just 21 (or 3.64%) of the \mathcal{C}_{on} structures correctly re-fold their multiloop.

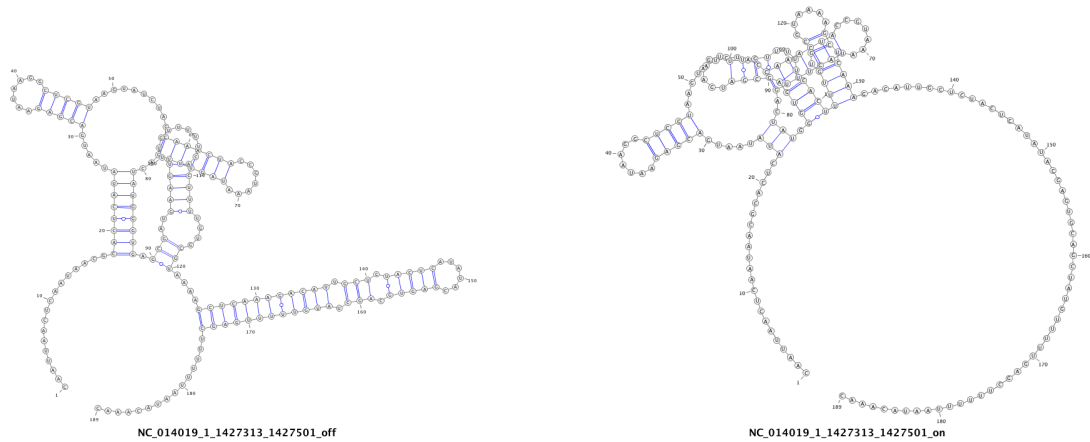


FIGURE 2.2: *Left*: the computationally predicted gene-off conformation of sequence NC_014019.1 1427313-1427501, using **RNAfold** from the ViennaRNA 2.1.8 suite, with dangles disabled and the Turner 1999 energies. This sequence is located upstream of the xpt gene in *Bacillus megaterium* QM B1551. *Right*: the gene-on conformation.

2.1.4 Fine-Grained Filtering

Until now, we have described our approach for generating the 503 guanine riboswitch candidates in RefSeq, alongside their gene-on and off structures. Unfortunately the experimental validation of all 503 candidates is not tractable, so it was necessary to reduce this collection again to a more manageable size, while only keeping the most promising candidates.

2.1.4.1 Using FoldAlign to Rank Candidates

Bibliography

- [1] A. S. Arnold, J. S. Wilson, and M. G. Boshier. A simple extended-cavity diode laser. *Review of Scientific Instruments*, 69(3):1236–1239, March 1998. URL <http://link.aip.org/link/?RSI/69/1236/1>.
- [2] Carl E. Wieman and Leo Hollberg. Using diode lasers for atomic physics. *Review of Scientific Instruments*, 62(1):1–20, January 1991. URL <http://link.aip.org/link/?RSI/62/1/1>.
- [3] C. J. Hawthorn, K. P. Weber, and R. E. Scholten. Littrow configuration tunable external cavity diode laser with fixed direction output beam. *Review of Scientific Instruments*, 72(12):4477–4479, December 2001. URL <http://link.aip.org/link/?RSI/72/4477/1>.