Optimizing Lempel Ziv Welch for DNA Compression

A Thesis

Presented to

The Division of Mathematics and Natural Sciences

Reed College

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Arts

Caden Corontzos

May 2023

Approved for the Division
(Computer Science)

Eitan Frachtenberg

Acknowledgements

I want to thank a few people.

Preface

This is an example of a thesis setup to use the reed thesis document class (for LaTeX) and the R bookdown package, in general.

List of Abbreviations

LZW Lempel Ziv Welch

Table of Contents

List of Tables

List of Figures

Abstract

The Lempel Ziv Welch compression algorithm is a lossless data compression algorithm used for numerous applications, including the Unix file compression utility `compress` and the GIF image format. Storing, reading, and transferring enormous amounts of data is often an issue in the biological field, especially when concerning DNA. This thesis explores the application of Lempel Ziv Welch to the compression of DNA. A variety of different optimization of the original LZW algorithm are explore included palatalizing, multiple dictionaries, and some other cool thing here broh.

Dedication

You can have a dedication here if you wish.

Introduction

When dealing with DNA, it

Chapter 1

Backround and Motivations

This thesis deals with some high level topics and uses language specific to compression research. This chapter tries to give brief summaries and examples of the relevant topics to be discussed so readers of all experience levels can put our results into context.

1.1 What is information?

Suppose you had an idea that you wanted to share with another person. Humans have many ways to communicate information; you could send a text message, you could tell them with words, you could tell them with sign language. But regardless of the medium, you have some idea that you want to get across. Does it matter if the other person gets your message exactly? Or can it be part of the message? If someone asks you "Where library", despite the lack of prepositions you still understand what they mean. So did that person convey any less information than a person who asks "Where is the library?" Clearly, information is fundamental to how humans interact and how they understand the world, but defining it proves difficult. For our purposes, let us assume that information is something that can be interpreted to glean information that you didn't know before.

1.2 Compression: A history

1.3 Compression Metrics

1.3.1 Compression Ratio

Compression Ratio is the measure of size reduction achieved by a compression algorithm. It is typically expressd as a ratio of the size of the uncompressed data (OS) to the size of the compressed data ({CS}).

$$CR = \frac{OS}{CS}$$

So a higher compression ratio means a more effective compression algorithm, and means that we were able to store more data in less space, allowing for easier storage and transfer.

1.3.2 Runtime

The runtime is also an important part of evaluating the effectiveness of a compression algorithm. If you have the option of two compression algorithms, one with a compression ratio of 2.0, and another with a compression ratio of 2.15 but takes twice as long as the other, you may opt for a lower compression ratio to save time.

1.3.3 Memory Usage

Memory usage is closely tied with runtime when it comes to compression algorithms. Memory generally refers to information that programs track as they are running on a computer. So do reduce our runtime and make a more effective compression algorithm, we want to be saving only the most important data that our algoritm needs in order to reduce our memory usage.

1.4 Lossless vs. Lossy Compression

1.4.1 Lossy

Lossy compression is based on the idea that not all information is vital. For instance, when saving a picture on your computer, your computer may save it in the .jpeg format to save space. Jpegs lose some of the information in the original picture and

produce an overall lower quality picture, but the general information in the picture is preserved. Another example

1.4.2 Lossless

Lossless compression is the compression of data with the goal of preserving all the information in the data. As a result, lossless compression algorithms usually don't compress as well as their lossy counterparts. Examples of lossless compression algoritms are Huffman Encoding and Lempel Ziv Welch, which is the focus of this thesis.

1.5 Examples of Compression Algorithms

1.5.1 Run Length Encoding

Run Length Encoding (RLE) is on of the simplest and most intuitive forms of compression. We can take advantage of redundant runs of characters in a sequence by just giving the number of times each character appears. Suppose you want to send the following message

AAGCTTTTTTTTGGGGGCCCT

Even if this message did mean something, we can get the information across without repeating ourselves. When writing a grocery list, you don't write "egg egg egg egg", you say "4 eggs". RLE uses this same strategdy.

2A1G1C8T5G3C1T

We could compress this even further if we omit the 1 on characters that only appear once.

1.5.2 Huffman

Huffman Encoding is a strategdy that assigns variable length code to certain symbols in the data. The goal is to assign short codes to frequently appearing symbols and longer codes to less frequent symbols.

Put example here

1.5.3 Arithmetic

Arithmetic encoding is another lossless compression algorithm that uses probability to asssign codes to symbols in the message. Unlike Huffman, arithmetic enoding assins

a single code to the whole message, rather than seperate codes for each symbol.

Here is a simple example. Say we want to encode a string of characters "ACCGGGGTTT". The probability of each symbol in the message are

- P(A) = 1/10
- P(C) = 2/10
- P(G) = 4/10
- P(T) = 3/10

We want to represent the message as a fractional number between 0 and 1. We will divide the interval [0,1] into sub intervals using the probabilities of each character in the message. That way, each symbol is reprsented by the sub-interval that corresponds to its probablity.

Arithmetic encoding can have a better compression ratio that Huffman in some cases, but the computation time is often not worth the payoff.

1.5.4 Lempel Ziv Welch

Lempel Ziv Welch is another lossless compression algorithm. When compressing, LZW builds a dictionary of codewords, where codewords represent strings previously seen in the message. As it compresses the message, the dictionary grows. The compression algorithm leaves behind the codewords and some of the original characters, allowing the decompression algorithm to build up the same dictionary as it decompresses the message.

Here is a simple example. We may be sending messages with the characters {'A', 'C', 'T', 'G'}, so I will start with those in my dictionary. Say we want to send the message

"AAGGAATCC"

When we compress, we start at the beginning of the message and scan through.

Chapter 2

Optimizing LZW: Approach

To restate the goal of this thesis, we seek to optimize LZW for use in compression of DNA. I chose to write in C++.

2.1 Supporting Research

There has been several attempts to optimize LZW by computer science researchers.

2.2 A Starting Point

2.3 Techniques Used

I used several programming tools in order to access and optimize my code.

2.3.1 Valgrind and Callgrind

2.3.2 gprof

2.3.3 perf

2.4 Trying Different Dictionaries

2.4.1 Direct Map

2.4.2 Multiple Indexed Dictionaries

Chapter 3

Graphics, References, and Labels

3.1 Figures

If your thesis has a lot of figures, *R Markdown* might behave better for you than that other word processor. One perk is that it will automatically number the figures accordingly in each chapter. You'll also be able to create a label for each figure, add a caption, and then reference the figure in a way similar to what we saw with tables earlier. If you label your figures, you can move the figures around and *R Markdown* will automatically adjust the numbering for you. No need for you to remember! So that you don't have to get too far into LaTeX to do this, a couple **R** functions have been created for you to assist. You'll see their use below.

In the **R** chunk below, we will load in a picture stored as `reed.jpg` in our main directory. We then give it the caption of "Reed logo", the label of "reedlogo", and specify that this is a figure. Make note of the different **R** chunk options that are given in the R Markdown file (not shown in the knitted document).

```
include_graphics(path = "figure/reed.jpg")
```

Figure 3.1: Reed logo

Here is a reference to the Reed logo: Figure 3.1. Note the use of the `fig:` code

here. By naming the **R** chunk that contains the figure, we can then reference that figure later as done in the first sentence here. We can also specify the caption for the figure via the R chunk option `fig.cap`.

Below we will investigate how to save the output of an **R** plot and label it in a way similar to that done above. Recall the `flights` dataset from Chapter **??**. (Note that we've shown a different way to reference a section or chapter here.) We will next explore a bar graph with the mean flight departure delays by airline from Portland for 2014.

```
mean_delay_by_carrier <- flights %>%
  group_by(carrier) %>%
  summarize(mean_dep_delay = mean(dep_delay))
ggplot(mean_delay_by_carrier, aes(x = carrier, y = mean_dep_delay)) +
  geom_bar(position = "identity", stat = "identity", fill = "red")
```

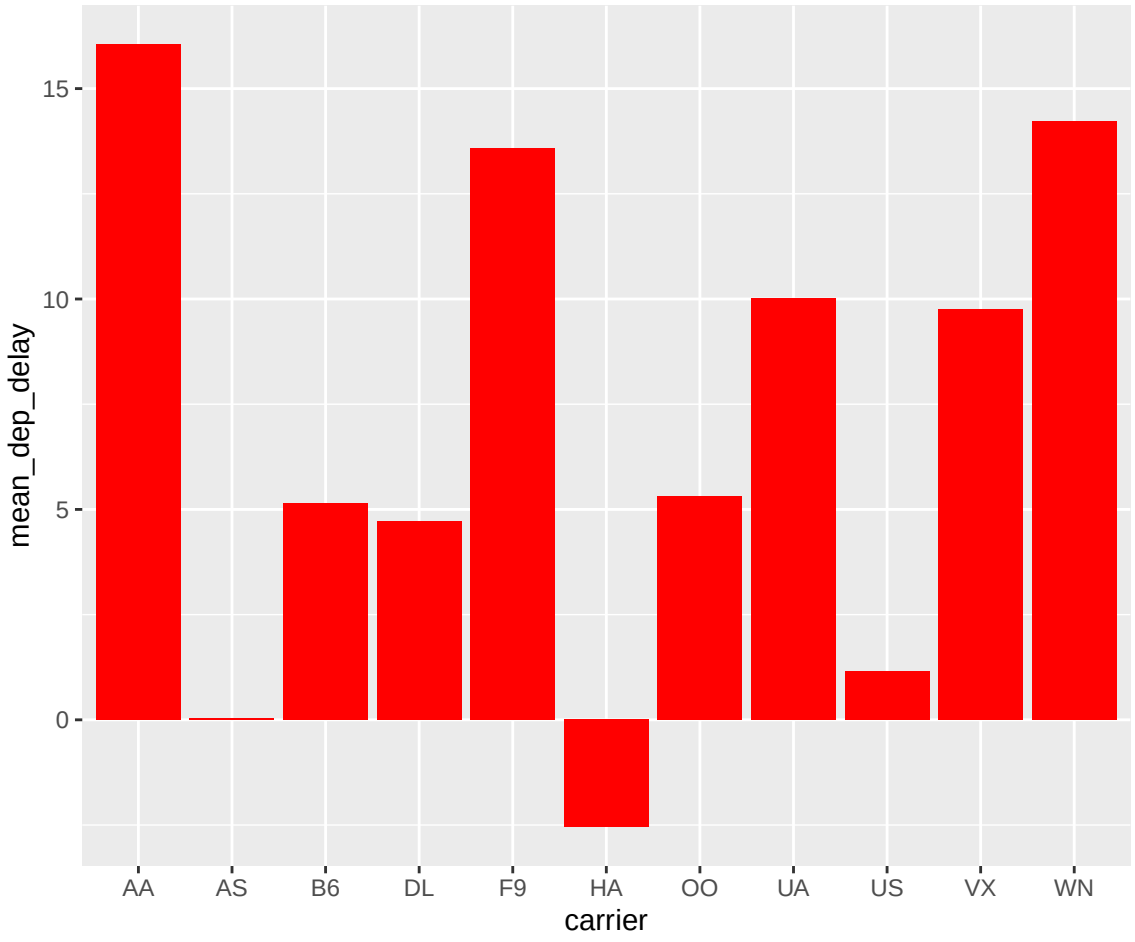

Figure 3.2: Mean Delays by Airline

Here is a reference to this image: Figure 3.2.

A table linking these carrier codes to airline names is available at `https://github.com/ismayc/pnwflights14/blob/master/data/airlines.csv`.

Next, we will explore the use of the `out.extra` chunk option, which can be used to shrink or expand an image loaded from a file by specifying `"scale= "`. Here we use the mathematical graph stored in the "subdivision.pdf" file.

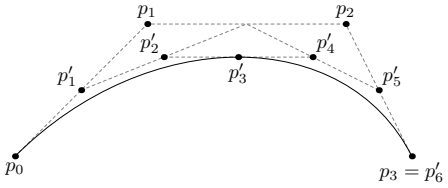

Figure 3.3: Subdiv. graph

Here is a reference to this image: Figure 3.3. Note that `echo=FALSE` is specified so that the **R** code is hidden in the document.

More Figure Stuff

Lastly, we will explore how to rotate and enlarge figures using the `out.extra` chunk option. (Currently this only works in the PDF version of the book.)

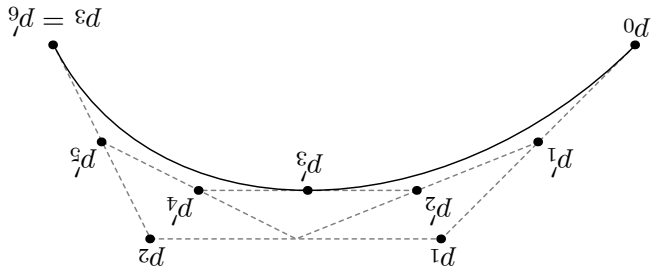

Figure 3.4: A Larger Figure, Flipped Upside Down

As another example, here is a reference: Figure 3.4.

3.2 Footnotes and Endnotes

You might want to footnote something. [1] The footnote will be in a smaller font and placed appropriately. Endnotes work in much the same way. More information can be found about both on the CUS site or feel free to reach out to `data@reed.edu`.

3.3 Bibliographies

Of course you will need to cite things, and you will probably accumulate an armful of sources. There are a variety of tools available for creating a bibliography

[1] footnote text

database (stored with the .bib extension). In addition to BibTeX suggested below, you may want to consider using the free and easy-to-use tool called Zotero. The Reed librarians have created Zotero documentation at `https://libguides.reed.edu/citation/zotero`. In addition, a tutorial is available from Middlebury College at `https://sites.middlebury.edu/zoteromiddlebury/`.

R Markdown uses *pandoc* (`https://pandoc.org/`) to build its bibliographies. One nice caveat of this is that you won't have to do a second compile to load in references as standard LaTeX requires. To cite references in your thesis (after creating your bibliography database), place the reference name inside square brackets and precede it by the "at" symbol. For example, here's a reference to a book about worrying: (Molina & Borkovec, 1994). This `Molina1994` entry appears in a file called `thesis.bib` in the `bib` folder. This bibliography database file was created by a program called BibTeX. You can call this file something else if you like (look at the YAML header in the main .Rmd file) and, by default, is to placed in the `bib` folder.

For more information about BibTeX and bibliographies, see our CUS site (`https://web.reed.edu/cis/help/latex/index.html`)[2]. There are three pages on this topic: *bibtex* (which talks about using BibTeX, at `https://web.reed.edu/cis/help/latex/bibtex.html`), *bibtexstyles* (about how to find and use the bibliography style that best suits your needs, at `https://web.reed.edu/cis/help/latex/bibtexstyles.html`) and *bibman* (which covers how to make and maintain a bibliography by hand, without BibTeX, at `https://web.reed.edu/cis/help/latex/bibman.html`). The last page will not be useful unless you have only a few sources.

If you look at the YAML header at the top of the main .Rmd file you can see that we can specify the style of the bibliography by referencing the appropriate csl file. You can download a variety of different style files at `https://www.zotero.org/styles`. Make sure to download the file into the csl folder.

Tips for Bibliographies

- Like with thesis formatting, the sooner you start compiling your bibliography for something as large as thesis, the better. Typing in source after source is mind-numbing enough; do you really want to do it for hours on end in late April? Think of it as procrastination.
- The cite key (a citation's label) needs to be unique from the other entries.

[2]Reed College (2007)

- When you have more than one author or editor, you need to separate each author's name by the word "and" e.g. `Author = {Noble, Sam and Youngberg, Jessica},`.
- Bibliographies made using BibTeX (whether manually or using a manager) accept LaTeX markup, so you can italicize and add symbols as necessary.
- To force capitalization in an article title or where all lowercase is generally used, bracket the capital letter in curly braces.
- You can add a Reed Thesis citation[3] option. The best way to do this is to use the phdthesis type of citation, and use the optional "type" field to enter "Reed thesis" or "Undergraduate thesis."

3.4 Anything else?

If you'd like to see examples of other things in this template, please contact the Data @ Reed team (email `data@reed.edu`) with your suggestions. We love to see people using *R Markdown* for their theses, and are happy to help.

[3]Noble (2002)

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{-}` attribute.

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In the main Rmd file

```
# This chunk ensures that the thesisdown package is
# installed and loaded. This thesisdown package includes
# the template files for the thesis.
if (!require(remotes)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("remotes", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste('You need to run install.packages("remotes")",
            "first in the Console.')
    )
  }
}
if (!require(thesisdown)) {
  if (params$`Install needed packages for {thesisdown}`) {
    remotes::install_github("ismayc/thesisdown")
  } else {
    stop(
      paste(
        "You need to run",
```

```
        'remotes::install_github("ismayc/thesisdown")',
        "first in the Console."
      )
    )
  }
}
library(thesisdown)
# Set how wide the R output will go
options(width = 70)
```

In Chapter 3:

```
# This chunk ensures that the thesisdown package is
# installed and loaded. This thesisdown package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if (!require(remotes)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("remotes", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("remotes")',
        "first in the Console."
      )
    )
  }
}
if (!require(dplyr)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("dplyr", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("dplyr")',
        "first in the Console."
```

```
      )
    )
  }
}
if (!require(ggplot2)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("ggplot2", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("ggplot2")',
        "first in the Console."
      )
    )
  }
}
if (!require(bookdown)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("bookdown", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("bookdown")',
        "first in the Console."
      )
    )
  }
}
if (!require(thesisdown)) {
  if (params$`Install needed packages for {thesisdown}`) {
    remotes::install_github("ismayc/thesisdown")
  } else {
    stop(
      paste(
        "You need to run",
        'remotes::install_github("ismayc/thesisdown")',
```

```
        "first in the Console."
      )
    )
  }
}
library(thesisdown)
library(dplyr)
library(ggplot2)
library(knitr)
flights <- read.csv("data/flights.csv", stringsAsFactors = FALSE)
```

Appendix B

The Second Appendix, for Fun

References

Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. Pr

Sayood, K. (2017). Introduction to data compression. Academic Press.

Witten, I. H., Neal, R. M., & Cleary, J. G. (1987). Arithmetic coding for data c

Sayood, K. (2017). Introduction to data compression. Academic Press.

Angel, E. (2000). *Interactive computer graphics : A top-down approach with OpenGL*. Boston, MA: Addison Wesley Longman.

Angel, E. (2001a). *Batch-file computer graphics : A bottom-up approach with QuickTime*. Boston, MA: Wesley Addison Longman.

Angel, E. (2001b). *Test second book by angel*. Boston, MA: Wesley Addison Longman.

Molina, S. T., & Borkovec, T. D. (1994). The Penn State worry questionnaire: Psychometric properties and associated characteristics. In G. C. L. Davey & F. Tallis (Eds.), *Worrying: Perspectives on theory, assessment and treatment* (pp. 265–283). New York: Wiley.

Noble, S. G. (2002). *Turning images into simple line-art* (Undergraduate thesis). Reed College.

Reed College. (2007). LaTeX your document. Retrieved from `https://web.reed.edu/cis/help/LaTeX/index.html`