Modelling Sanger Sequencing Traces by Position and Context

November 9th, 2018

Part of multiEditR Project

Lab of Dr. Branden Moriarity

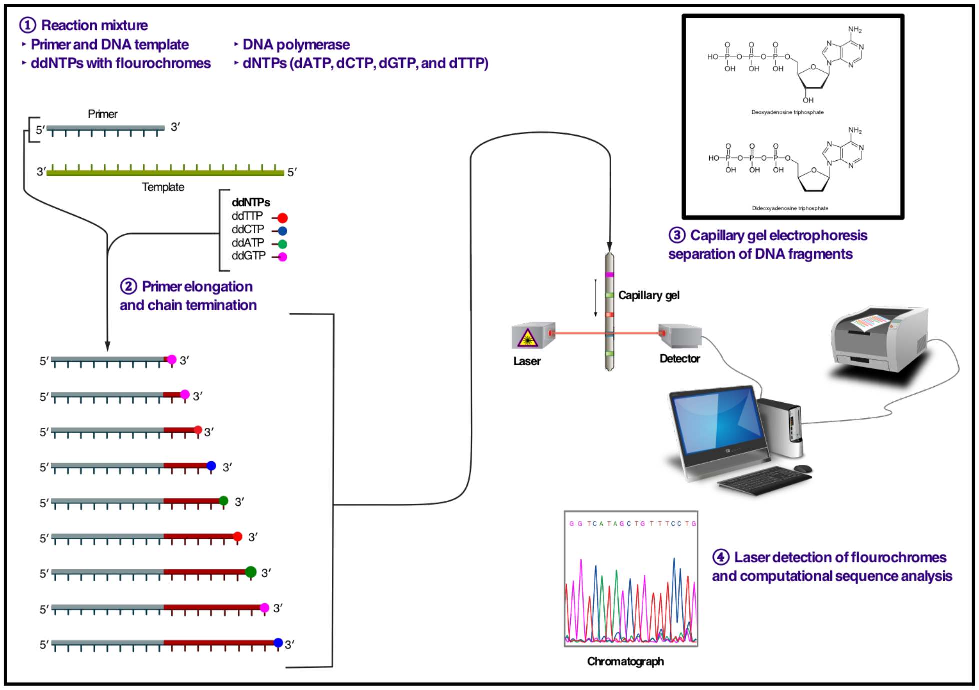
Mitchell Kluesner

Samantha Lee

Confidential, please do not distribute without permission.

Background

Sanger Dideoxy Sequencing



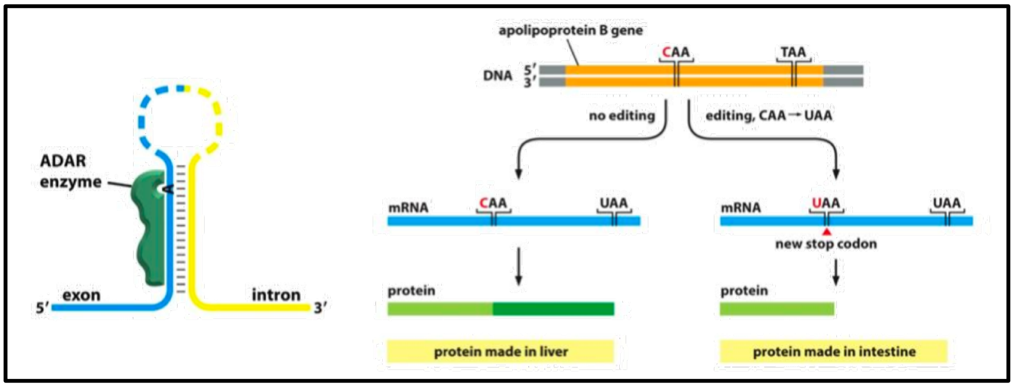
① Reaction mixture
‣ Primer and DNA template
‣ ddNTPs with flourochromes
‣ DNA polymerase
‣ dNTPs (dATP, dCTP, dGTP, and dTTP)
Primer
5′
3′
3′
5′
Template
ddNTPs
ddTTP
ddCTP
ddATP
ddGTP
② Primer elongation and chain termination
Deoxyadenosine triphosphate
Dideoxyadenosine triphosphate
③ Capillary gel electrophoresis separation of DNA fragments
Capillary gel
Laser
Detector
④ Laser detection of flourochromes and computational sequence analysis
G G T C A T A G C T G T T T C C T G
Chromatograph

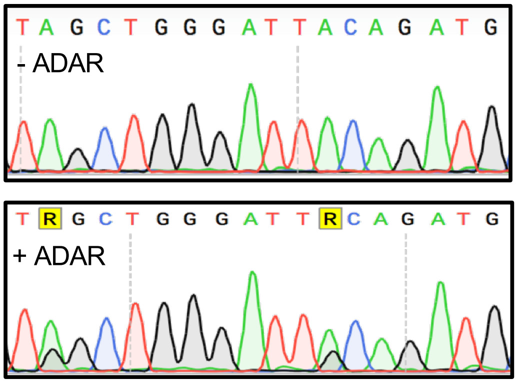
https://en.wikipedia.org/wiki/Sanger_sequencing

- Output of sanger sequencing is a chromatogram as a .ab1 or .scf file
- Sanger sequencing can be used in a quantitative fashion to measure percentage of indels and SNPs from a pool of amplicons.
- Our lab previously published a paper on quantifying SNPs from samples treated with the gene editing technology CRISPR-Cas9 Base Editors
    - Kluesner & Nedveck et al., 2018, CRISPR Journal
- Now we are developing a tool to quantify multiple editing events in RNA editing
    - Termed multiEditR

Endogenous RNA Editing

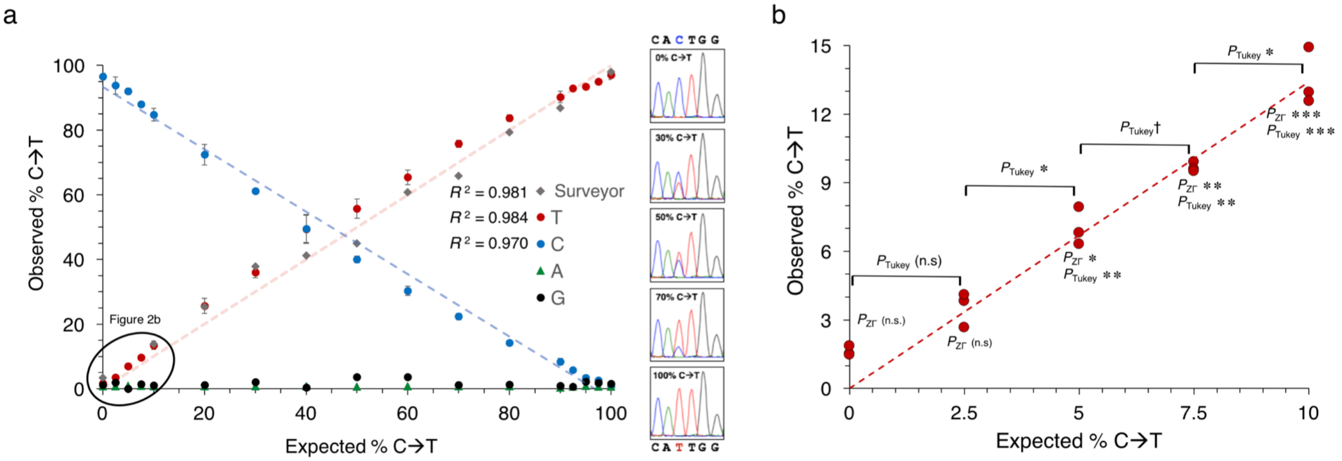
- Several enzymes have catalytic activity on mRNA that post-transcriptionally modifies mRNA sequences
- Operates via conversion of one nucleotide to another
- ADARs are adenosine deaminases that convert adenosine (A) to Inosine (I). In turn inosine behaves like guanine (G)
- APOBECs are cytosine deaminases that convert cytosines (C) to uridines (U). In turn uridine behaves like thymine (T)
- **Currently methods to quantify RNA editing are expensive, and time consuming**





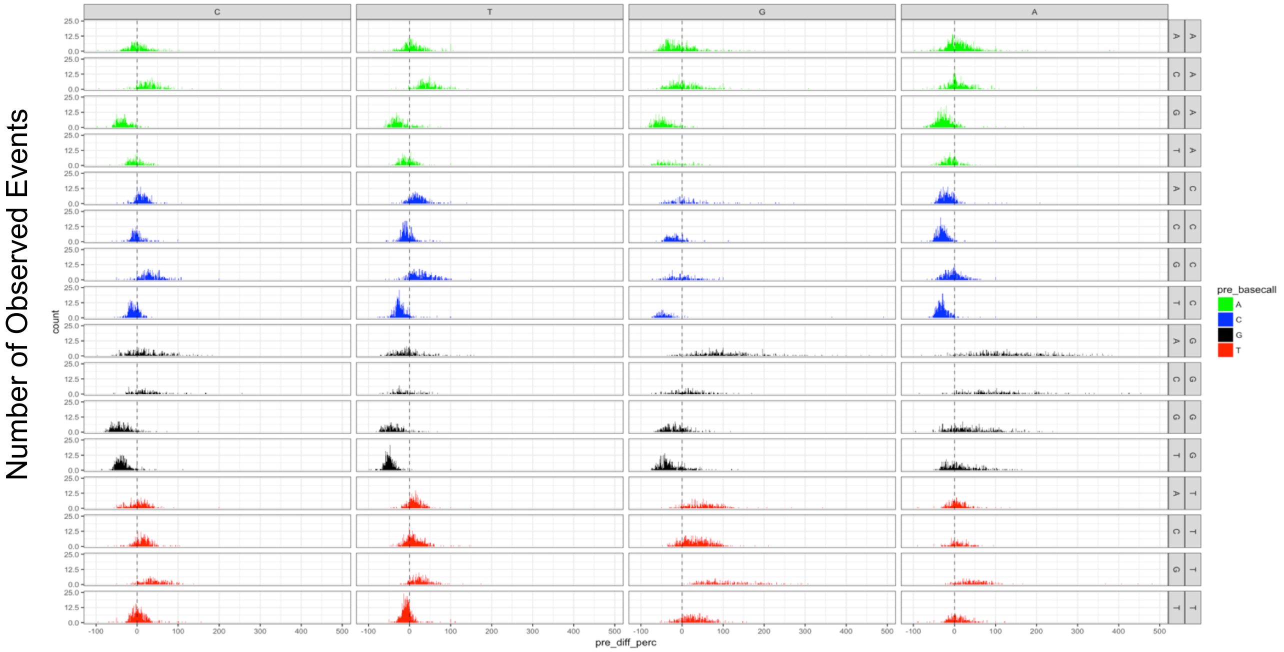
- On the left are two sanger sequencing chromatograms from the same transcript
- On the top, cells without ADAR, and on the bottom a cells with ADAR
- Presence of ADAR creates A>I (A>G) edits in the sequence observed in chromatogram

We can measure the percent editing at a single base from sanger sequencing for C>T



However, A>G appear less accurate and are especially influenced by sequence context

Percent change from previous base height by base identity and upstream two bases



Distribution of Percent Change from Previous base

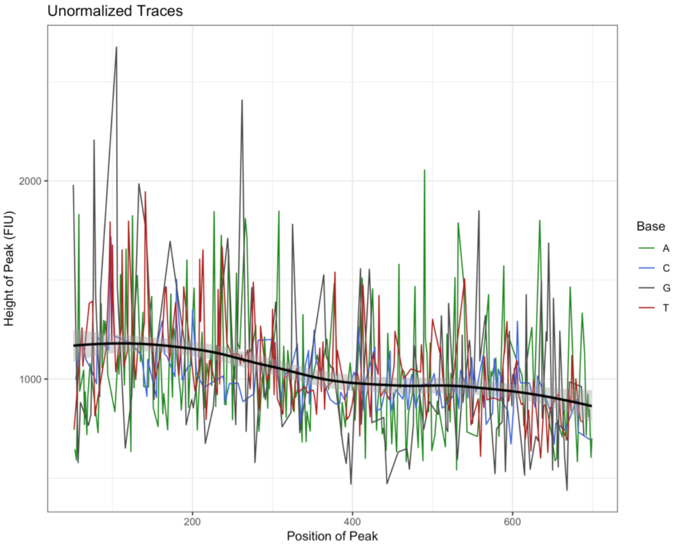
- Data from 31 different files, 19,579 bases
- Each column is looking at a different base, each row is a different preceding dinucleotide
- Clusters are colored by just the preceding base
- Distributions most central to the dashed line have the least amount of change, while those that are skewed from the line demonstrate a high degree of context dependency
- The skew is heaviest when either G or T is preceding the base being measured, while the skew is smallest when C or A is preceding the base being measured.

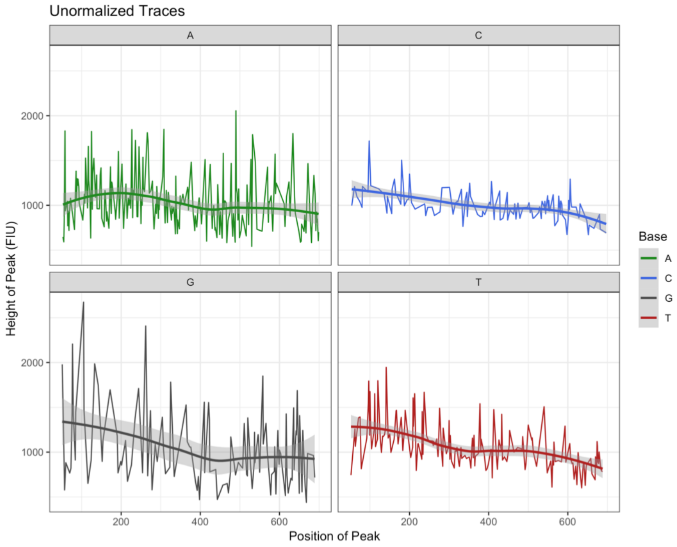
Can we use machine learning to normalize the height of each base by sequence context and position?
What models and approaches would be most adept to this problem?

Initial Considerations

Peak height decreases as the Sanger reactions proceeds

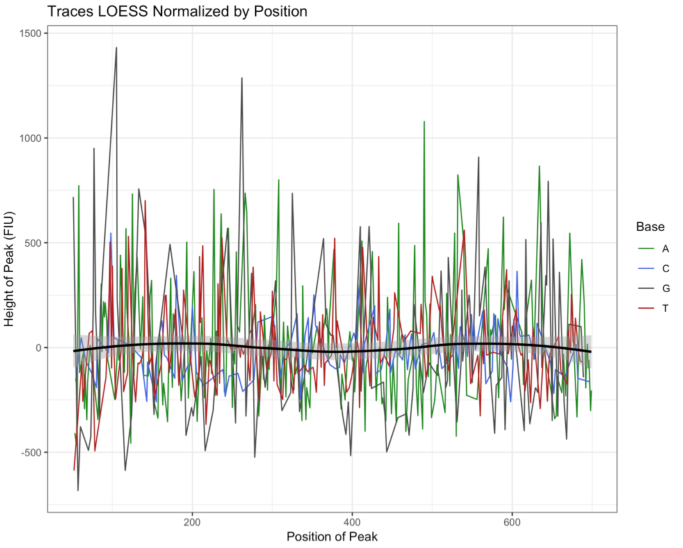
- As a result as the position of the base increases, the local average of height decreases
- On left is all the traces (A, C, G and T) superimposed, on the right is individual
- Regression line is a non-parametric locally weighted regression (LOESS), always over-fitted

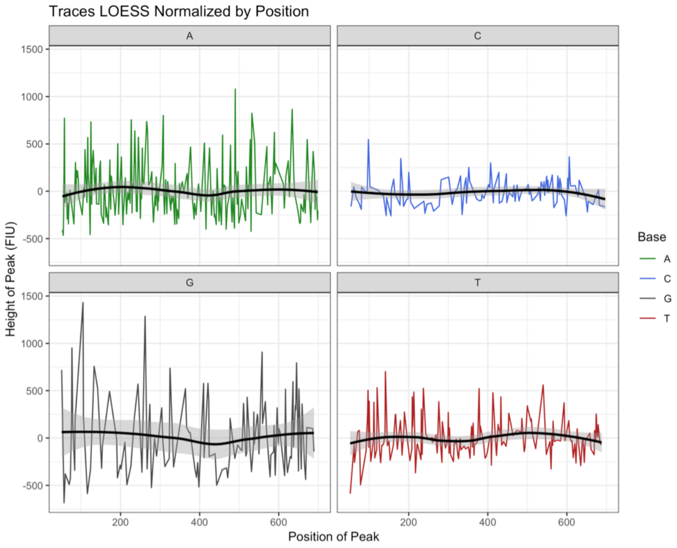




Can we normalize initial data with the LOESS regression, then use data for machine learning?

- Below is the same data, except with the predicted value from the LOESS regression subtracted from the initial value, generating a normalized value centered about 0
- Data may be more amenable to machine learning operation and may eliminate position effects
- I have tried to explore context effects further with this data without success using simple linear models





Further considerations

- All data analysis and visualization was done R
- Future publication would need a program all implemented in R for adaptation to R shiny web app
- Model could be built in Python or any other platform as long we can transfer what we learn into R
- If model can successfully adjust base height by context, then we would have a strong program to publish on
- **We have unique data on between 32,000 – 64,000 bases available to build a model**