

Intro to ML & Bedtools

Michael Schatz

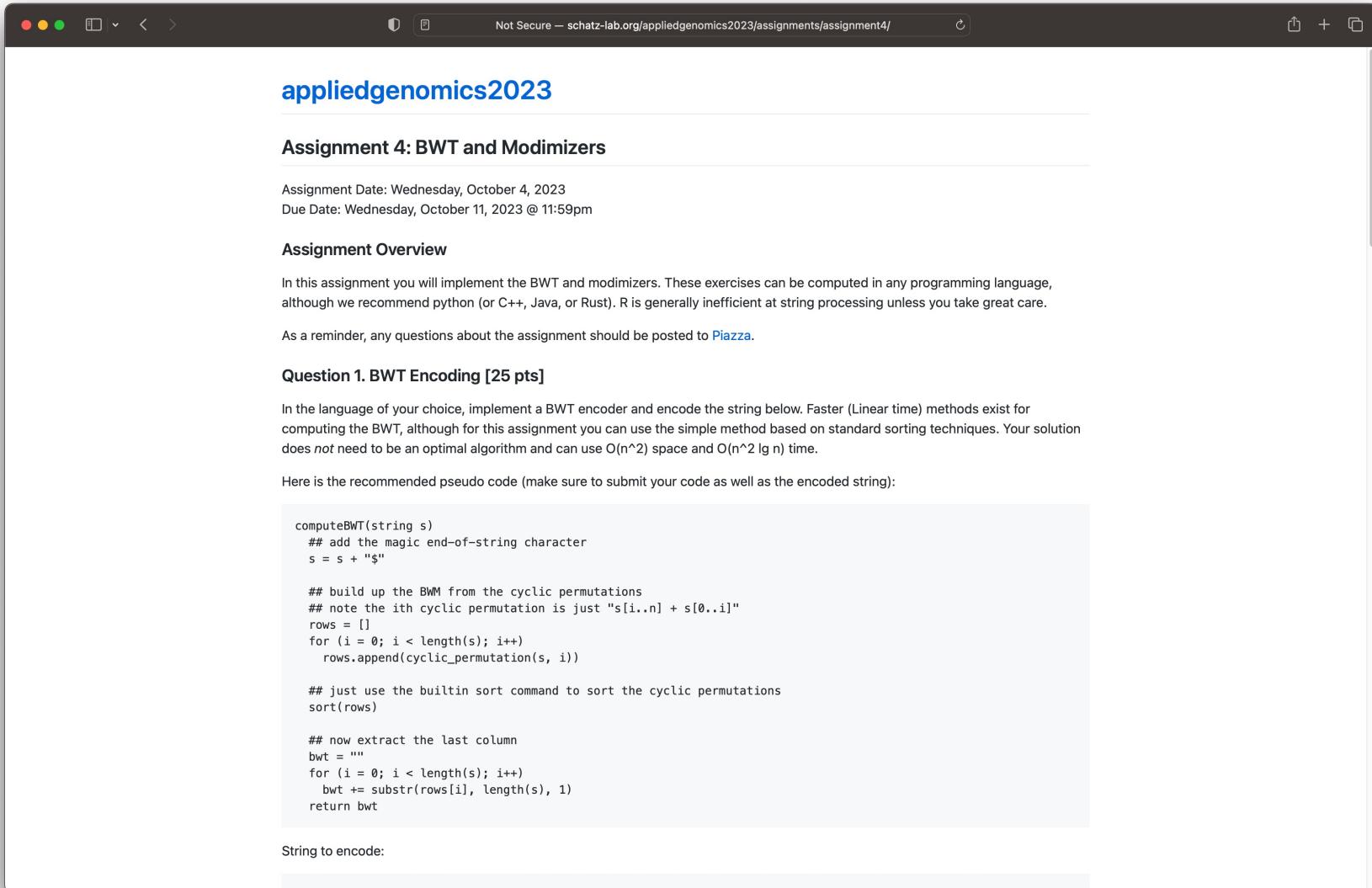
October 11, 2023

Lecture 13. Applied Comparative Genomics



Assignment 4: BWT and Modimizers

Due Wednesday Oct 11 by 11:59pm



The screenshot shows a web browser window with the URL [Not Secure – schatz-lab.org/appliedgenomics2023/assignments/assignment4/](https://schatz-lab.org/appliedgenomics2023/assignments/assignment4/). The page title is "appliedgenomics2023" and the section title is "Assignment 4: BWT and Modimizers". The assignment date is Wednesday, October 4, 2023, and the due date is Wednesday, October 11, 2023 @ 11:59pm. The "Assignment Overview" section states: "In this assignment you will implement the BWT and modimizers. These exercises can be computed in any programming language, although we recommend python (or C++, Java, or Rust). R is generally inefficient at string processing unless you take great care." It also notes: "As a reminder, any questions about the assignment should be posted to [Piazza](#)". The "Question 1. BWT Encoding [25 pts]" section contains pseudo code for implementing the BWT encoder:

```
computeBWT(string s)
    ## add the magic end-of-string character
    s = s + "$"

    ## build up the BWT from the cyclic permutations
    ## note the ith cyclic permutation is just "s[i..n] + s[0..i]"
    rows = []
    for (i = 0; i < length(s); i++)
        rows.append(cyclic_permutation(s, i))

    ## just use the builtin sort command to sort the cyclic permutations
    sort(rows)

    ## now extract the last column
    bwt = ""
    for (i = 0; i < length(s); i++)
        bwt += substr(rows[i], length(s), 1)
    return bwt
```

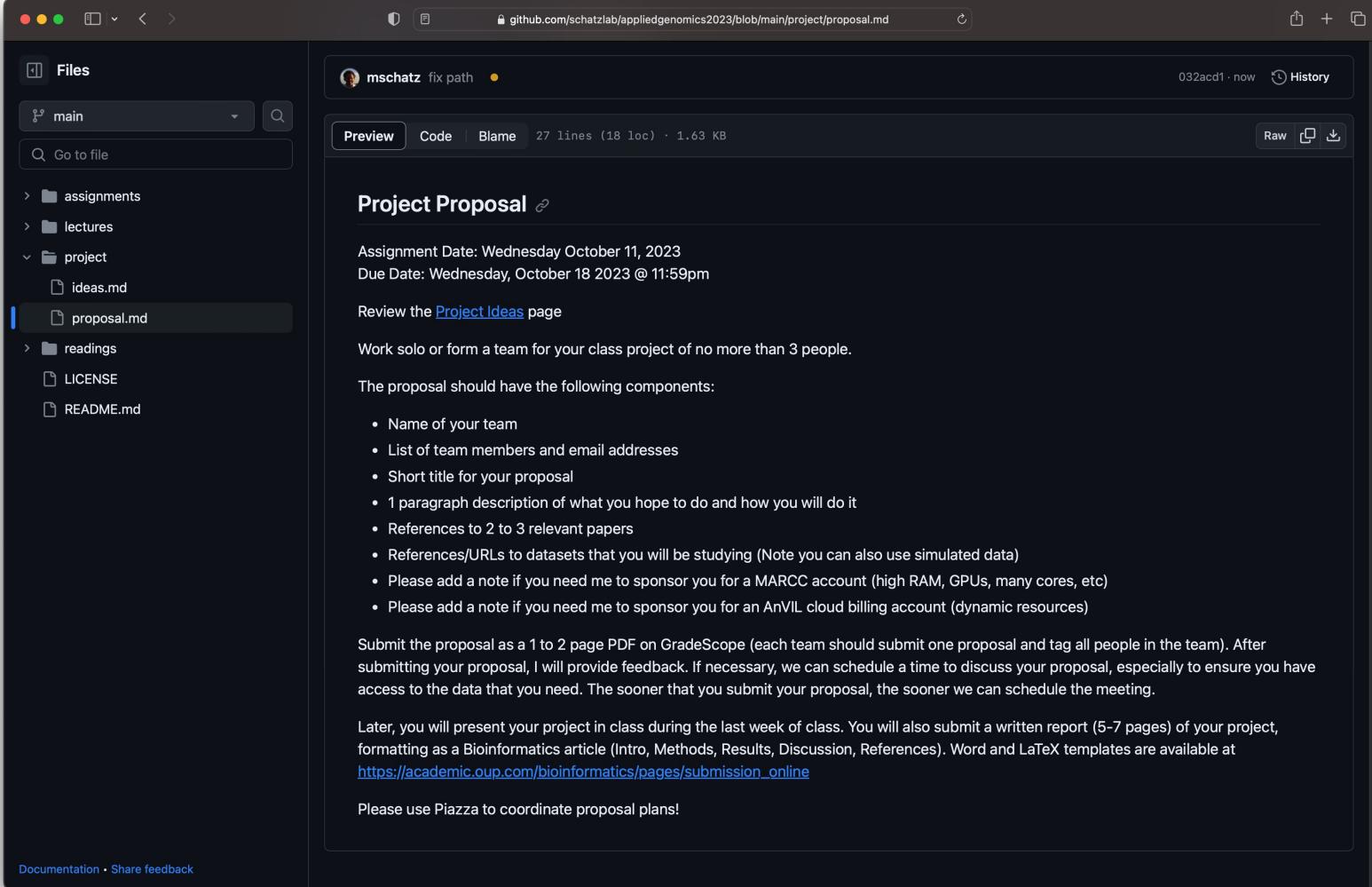
String to encode:

<https://github.com/schatzlab/appliedgenomics2023/tree/main/assignments/assignment4>

Check Piazza for questions!

Project Proposal

Due Wednesday Oct 18 by 11:59pm



The screenshot shows a GitHub repository interface. On the left, the file tree displays a directory structure: main, assignments, lectures, project (which contains ideas.md and proposal.md), readings, LICENSE, and README.md. The proposal.md file is selected. The main pane shows the content of proposal.md:

Project Proposal

Assignment Date: Wednesday October 11, 2023
Due Date: Wednesday, October 18 2023 @ 11:59pm

Review the [Project Ideas](#) page

Work solo or form a team for your class project of no more than 3 people.

The proposal should have the following components:

- Name of your team
- List of team members and email addresses
- Short title for your proposal
- 1 paragraph description of what you hope to do and how you will do it
- References to 2 to 3 relevant papers
- References/URLs to datasets that you will be studying (Note you can also use simulated data)
- Please add a note if you need me to sponsor you for a MARCC account (high RAM, GPUs, many cores, etc)
- Please add a note if you need me to sponsor you for an AnVIL cloud billing account (dynamic resources)

Submit the proposal as a 1 to 2 page PDF on GradeScope (each team should submit one proposal and tag all people in the team). After submitting your proposal, I will provide feedback. If necessary, we can schedule a time to discuss your proposal, especially to ensure you have access to the data that you need. The sooner that you submit your proposal, the sooner we can schedule the meeting.

Later, you will present your project in class during the last week of class. You will also submit a written report (5-7 pages) of your project, formatting as a Bioinformatics article (Intro, Methods, Results, Discussion, References). Word and LaTeX templates are available at https://academic.oup.com/bioinformatics/pages/submission_online

Please use Piazza to coordinate proposal plans!

<https://github.com/schatzlab/appliedgenomics2023/blob/main/project/proposal.md>

Check Piazza for questions!

Algorithm Overview

1. Split read into segments

Read
CCAGTAGCTCTCAGCCTTATTTACCCAGGCCTGTA

Read (reverse complement)
TACAGGCCTGGGTAAAATAAGGCTGAGAGCTACTGG

Policy: extract 16 nt seed every 10 nt

Seeds

+ , 0: CCAGTAGCTCTCAGCC	- , 0: TACAGGCCTGGGTAAA
+ , 10: TCAGCCTTATTTACC	- , 10: GGTAAAATAAGGCTGA
+ , 20: TTTACCCAGGCCTGTA	- , 20: GGCTGAGAGCTACTGG

2. Lookup each segment and prioritize

Seeds

+ , 0: CCAGTAGCTCTCAGCC	→	Ungapped alignment with FM Index	→	Seed alignments (as B ranges)
+ , 10: TCAGCCTTATTTACC		\$ a c a a c g a a c g \$ a c a c a a c g \$ a c g \$ a c a I c e - g - a I c e - g - a g \$ a c a a c		{ [211, 212], [212, 214] } { [653, 654], [651, 653] } { [684, 685] } { } { } { }
+ , 20: TTTACCCAGGCCTGTA				{ [624, 625] }
- , 0: TACAGGCCTGGGTAAA				
- , 10: GGTAAAATAAGGCTGA				
- , 20: GGCTGAGAGCTACTGG				

3. Evaluate end-to-end match

Extension candidates

SA:684, chr12:1955	→	SIMD dynamic programming aligner	→	SAM alignments
SA:624, chr2:462				r1 0 chr12 1936 0
SA:211: chr4:762				36M * 0 0
SA:213: chr12:1935				CCAGTAGCTCTCAGCCTTATTTACCCAGGCCTGTA
SA:652: chr12:1945				II

(Langmead & Salzberg, 2012)

Similarity metrics

- Hamming distance
 - Count the number of substitutions to transform one string into another

MIKESCHATZ

| | X | | XXXX |

MICESHATZZ

5

- Edit distance

- The minimum number of substitutions, insertions, or deletions to transform one string into another

MIKESCHAT-Z

| | X | | X | | | X |

MICES-HATZZ

3

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3	2	1	2	2	3	4	5	6
A	4	3	2	1	2	2	3	4	5
C	5	4	3	2	1	2	2	3	4
A	6	5	4	3	2	1	2	3	3
C	7	6	5	4	3	2	1	2	3
A	8	7	6	5	4	3	2	2	2

$$D[AGCACACA, ACACACTA] = 2$$

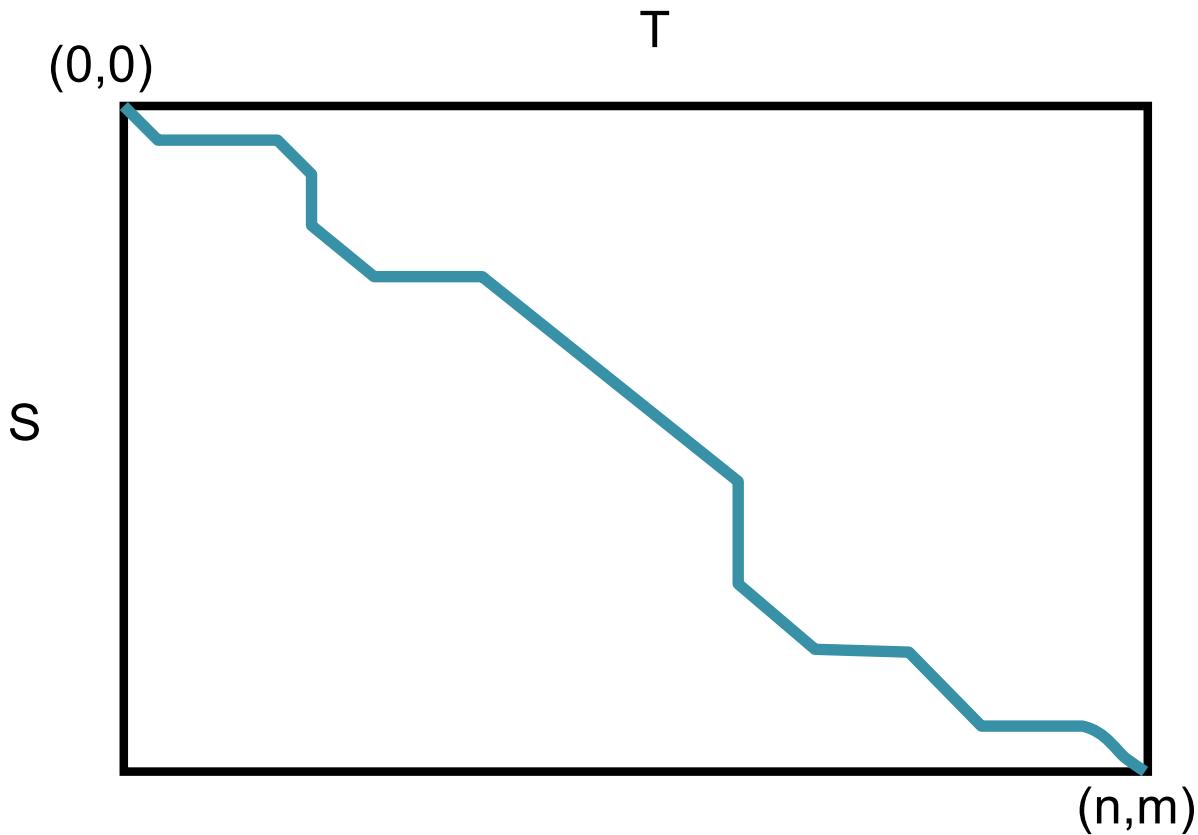
AGCACAC-A

| * | | | | * |

A-CACACTA

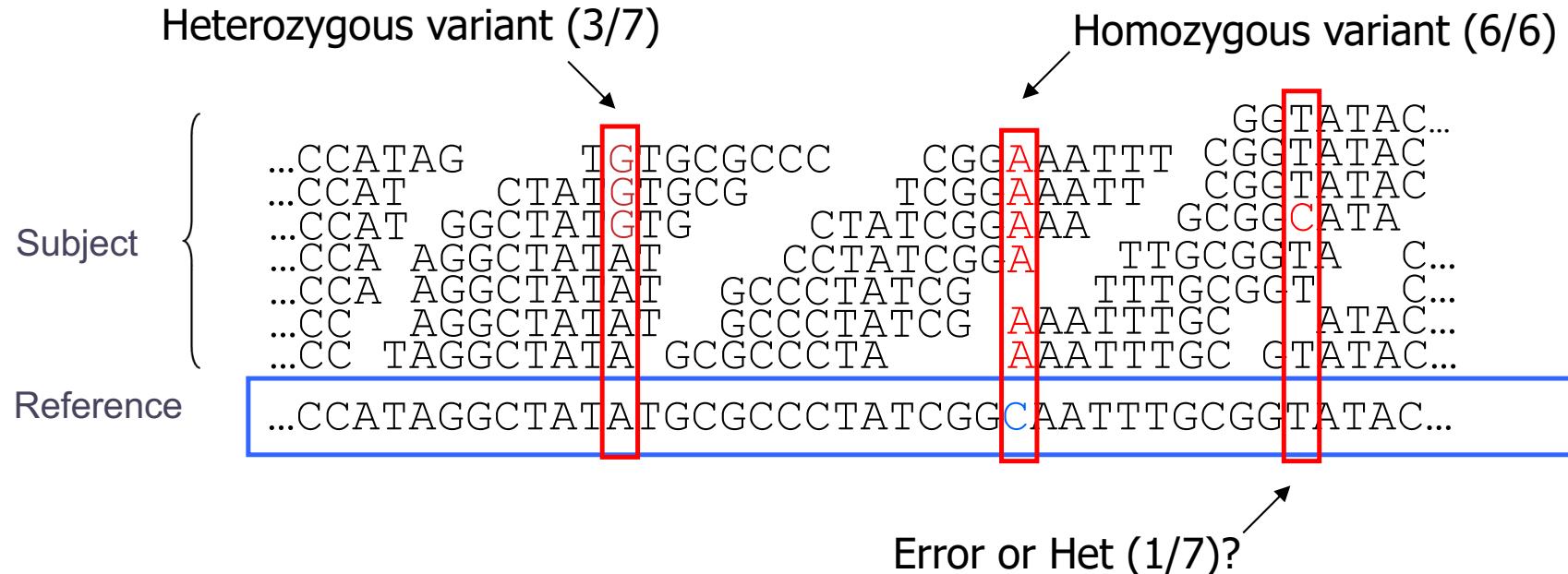
[Can we do it any better?]

Global Alignment Schematic

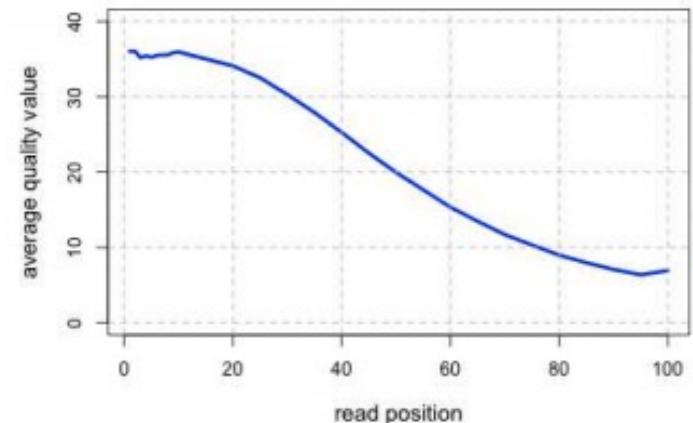


- A high quality alignment will stay close to the diagonal
 - If we are only interested in high quality alignments, we can skip filling in cells that can't possibly lead to a high quality alignment
 - Find the global alignment with at most edit distance d : $O(2dn)$

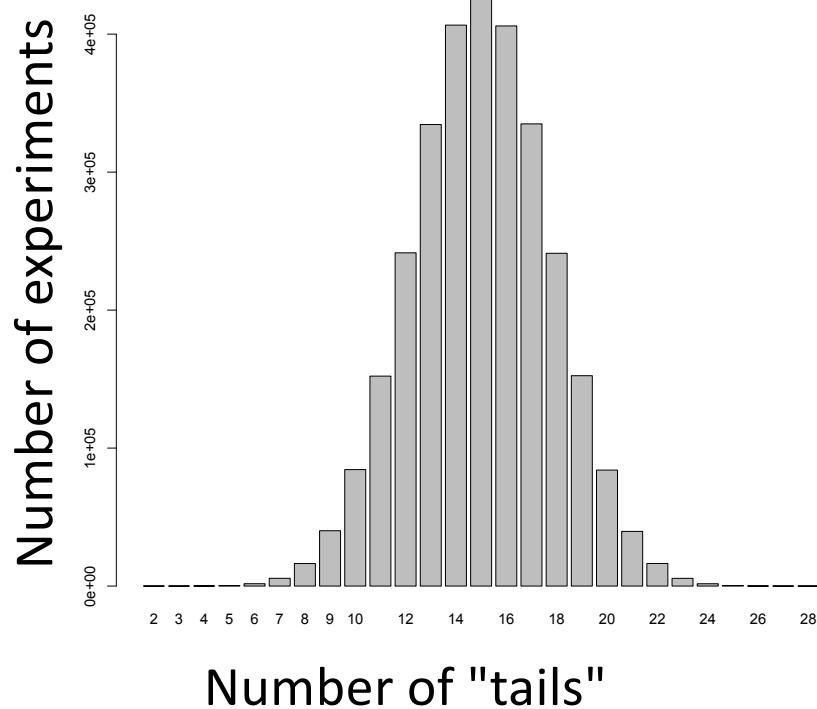
Genotyping Theory



- If there were no sequencing errors, identifying SNPs would be very easy: any time a read disagrees with the reference, it must be a variant!
- Sequencing instruments make mistakes
 - Quality of read decreases over the read length
- A single read differing from the reference is probably just an error, but it becomes more likely to be real as we see it multiple times



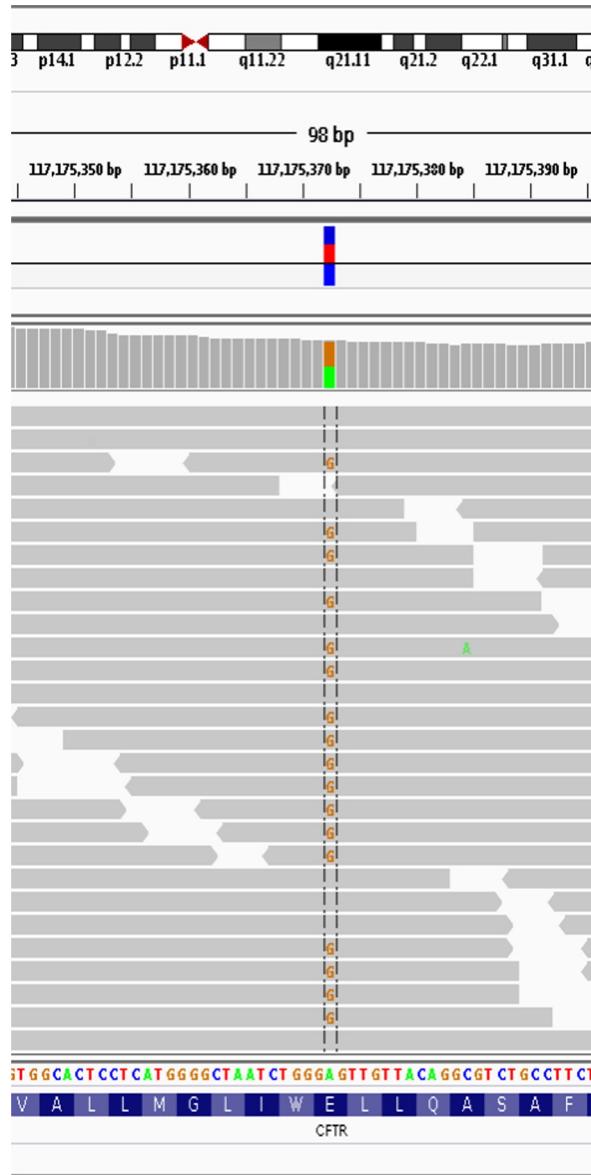
So, with 30 tosses (reads), we are much more likely to see an even mix of alternate and reference alleles at a heterozygous locus in a genome



This is why at least a "30X" (30 fold sequence coverage) genome is recommended: it confers sufficient power to distinguish heterozygous alleles and from mere sequencing errors

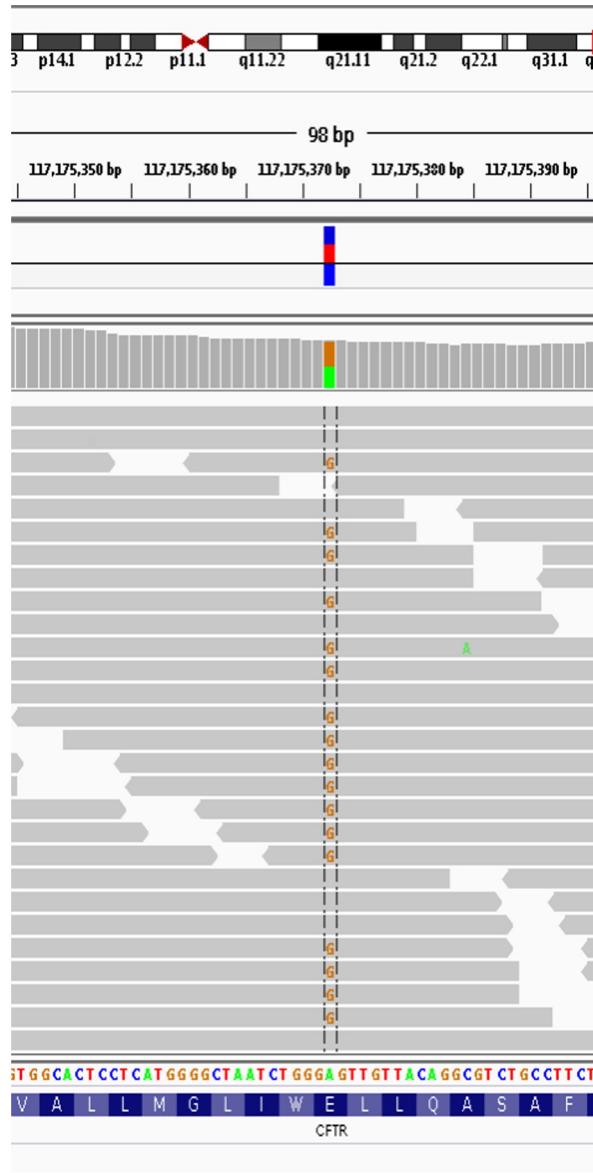
$P(3/30 \text{ het}) <?> P(3/30 \text{ err})$

What information is needed to decide if a variant exists?



- Depth of coverage at the locus
- Bases observed at the locus
- The base qualities of each allele
- The strand composition
- Mapping qualities
- Proper pairs?
- Expected polymorphism rate

Bayesian SNP calling



$$P(\text{SNP} | \text{Data}) = \frac{P(\text{Data} | \text{SNP}) * P(\text{SNP})}{P(\text{Data})}$$

PolyBayes: The first statistically rigorous variant detection tool.

letter

© 1999 Nature America Inc. • <http://genetics.nature.com>

A general approach to single-nucleotide polymorphism discovery

Gabor T. Marth¹, Ian Korf¹, Mark D. Yandell¹, Raymond T. Yeh¹, Zhijie Gu², Hamideh Zakeri², Nathan O. Stitzel¹, LaDeana Hillier¹, Pui-Yan Kwok² & Warren R. Gish¹

Bayesian posterior probability

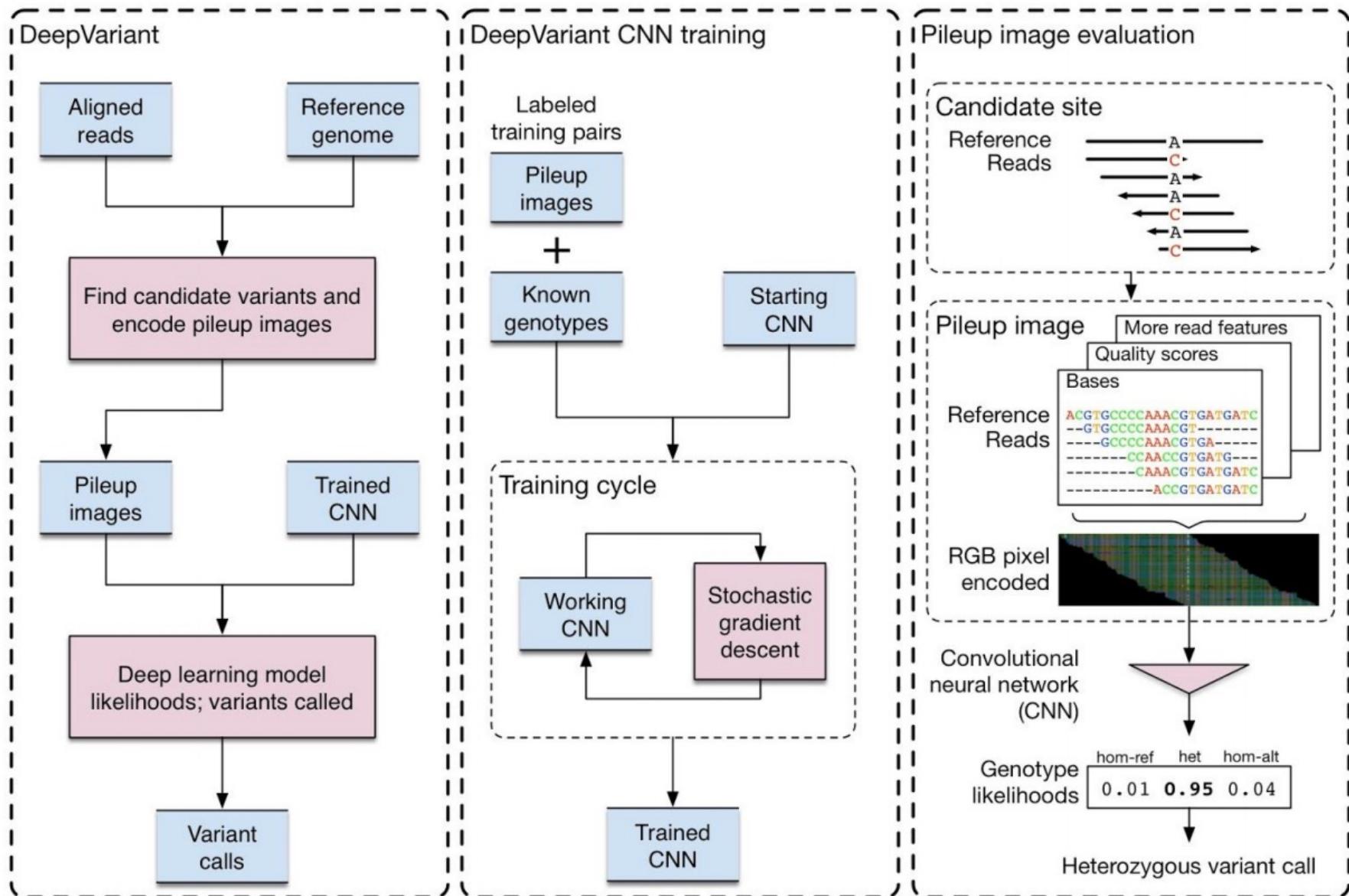
$$P(\text{SNP}) = \sum_{\text{all variable } S} \frac{\frac{P(S_1 | R_1) \dots P(S_N | R_N)}{P_{\text{Prior}}(S_1) \dots P_{\text{Prior}}(S_N)} \cdot P_{\text{Prior}}(S_1, \dots, S_N)}{\sum_{S_{i_1} \in [A,C,G,T]} \dots \sum_{S_{i_N} \in [A,C,G,T]} \frac{P(S_{i_1} | R_1) \dots P(S_{i_N} | R_1)}{P_{\text{Prior}}(S_{i_1}) \dots P_{\text{Prior}}(S_{i_N})} \cdot P_{\text{Prior}}(S_{i_1}, \dots, S_{i_N})}$$

Probability of observed base composition
(should model sequencing error rate)

Base call +
Base quality

Expected (prior)
polymorphism rate

DeepVariant



Creating a universal SNP and small indel variant caller with deep neural networks

Poplin et al. (2018) Nature Biotechnology. <https://www.nature.com/articles/nbt.4235>

1 Generate some noisy data

2 Fitting Lines to (multiple) points

2.1 Given any two points I can draw a straight line between them!

2.2 Given any three points, I can draw a quadratic between them!

2.3 Given any four points, I can draw a cubic between them!

2.3.1 First, 1/3, 2/3, last

2.3.2 Shifted First, 1/3, 2/3, last

2.4 Given n points, I can draw a n-1 degree polynomial between them!

3 Linear and Polynomial Regression

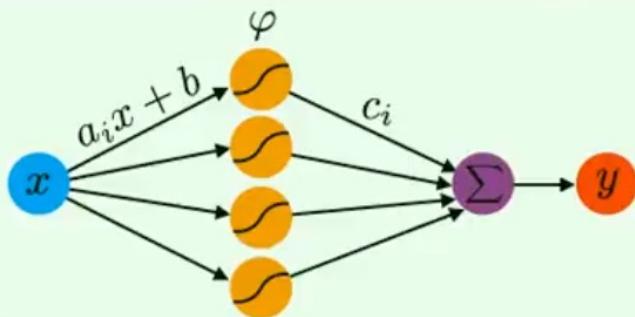
```
model3$coefficients[2]*q +  
model3$coefficients[1]  
lines(q, model3.y, col="green", lwd=3)
```

Observed data

The figure shows a scatter plot of 'noisy.y' versus 'q'. The x-axis ranges from -15 to 15, and the y-axis ranges from -1000 to 1000. The data points are scattered around a central peak at q ≈ -3. A green cubic curve is fitted to the data, passing through the first, middle, and last points of the observed data sequence.

2.3.2 Shifted First, 1/3, 2/3, last

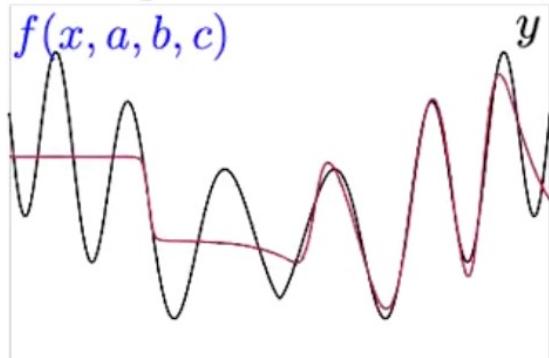
<https://schatz-lab.org/appliedgenomics2023/lectures/13.overfitting.html>



1 hidden layer perceptron:

$$y \approx f(x, a, b, c) \stackrel{\text{def.}}{=} \sum_{i=1}^p c_i \varphi(a_i x + b_i)$$

$p = 6$ neurons



$p = 20$ neurons



Approximation by Superpositions of a Sigmoidal Function*

G. Cybenko†

Abstract. In this paper we demonstrate that finite linear combinations of compositions of a fixed, univariate function and a set of affine functionals can uniformly approximate any continuous function of n real variables with support in the unit hypercube; only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single hidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks.

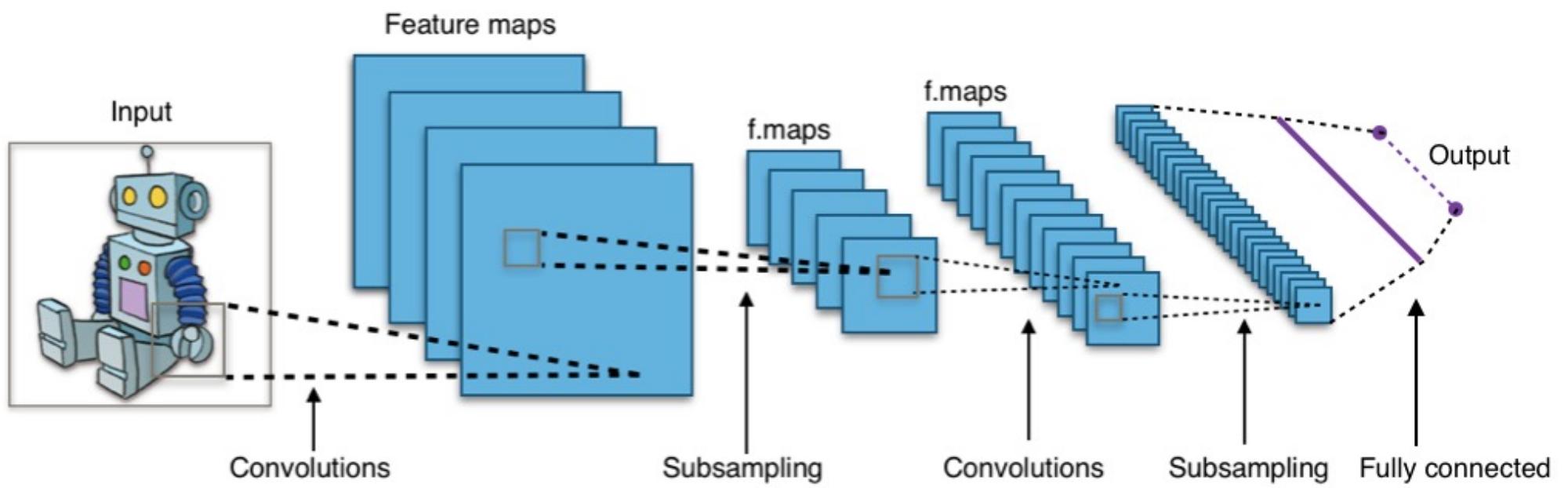
Key words. Neural networks, Approximation, Completeness.

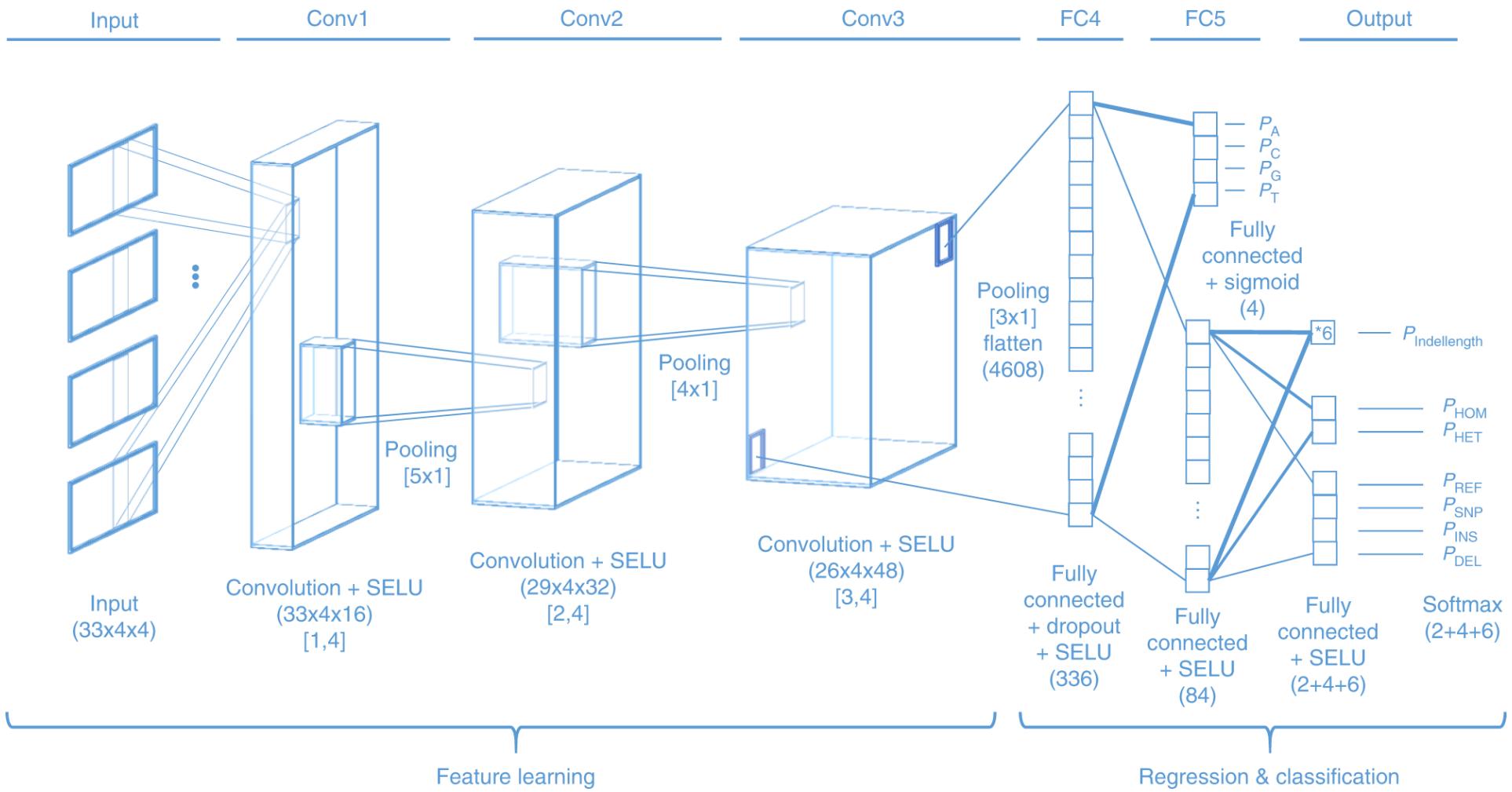


George Cybenko

Approximation by superpositions of a sigmoidal function

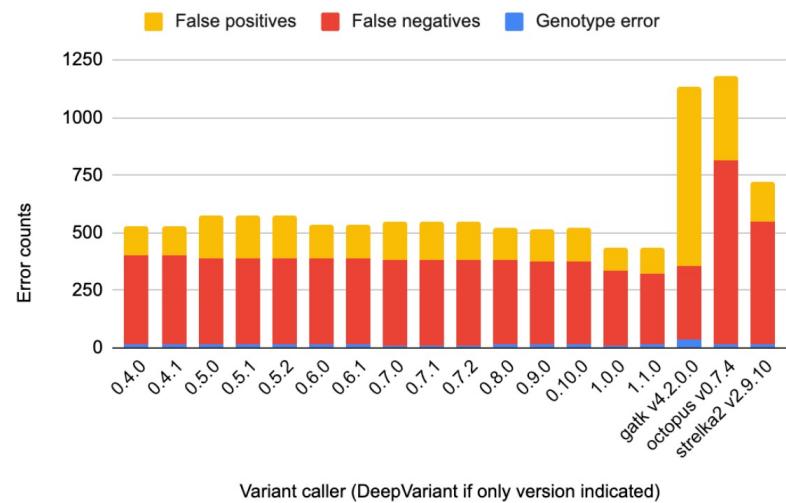
Cybenko, G. (1989) Mathematics of Control Signal Systems doi: 10.1007/BF02551274



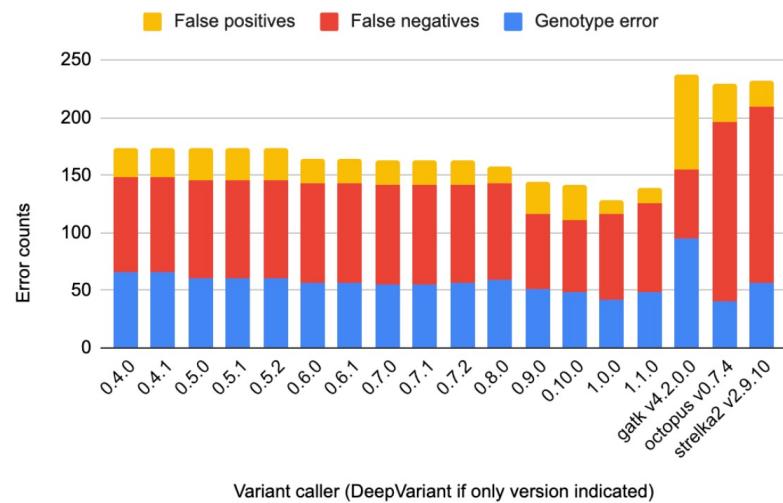


A multi-task convolutional deep neural network for variant calling in single molecule sequencing
 Luo et al. (2019) Nature Communication. <https://doi.org/10.1038/s41467-019-09025-z>

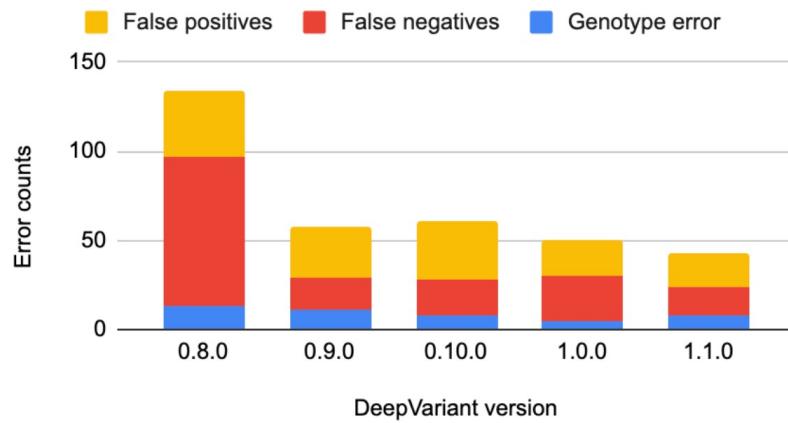
WGS SNP error counts (HG003)



WGS INDEL error counts (HG003)



PacBio SNP error counts (HG003)



PacBio INDEL error counts (HG003)

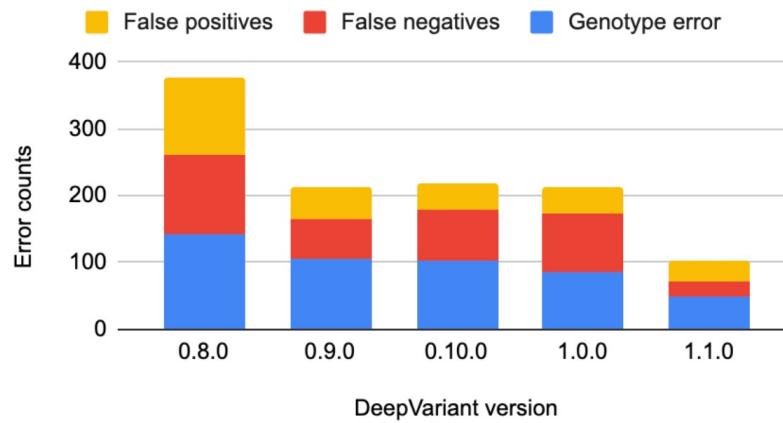


Figure 2: Error counts over the years for HG003. For Illumina WGS, we use a HiSeqX PCR-free dataset at 30x coverage. For PacBio HiFi, we use the same BAM as the one in our [case study](#).

DeepVariant over the years

<https://google.github.io/deepvariant/posts/2021-06-08-deepvariant-over-the-years/>