1) SNP Calling & Sampling Probability $P(E_i=1) = P(E_i=1 \mid G) = e$, $\begin{cases} P(S_i=A \mid G=AB) = S_A \\ P(S_i=B \mid G=AB) = S_B \end{cases}$

a) $P(O_i=A \mid G=AB) = P(O_i=A \mid E_i=0, S_i=A, G=AB) + P(O_i=A \mid E_i=1, S_i=B, G=AB)$

$\cdot P(E_i=0)$ $\cdot P(S_i=A)$ ← same as $P(E_i=0 \mid G=AB)$ $\cdot P(E_i=1)$ $\cdot P(S_i=B)$

$= 1 \cdot (1-e) \cdot S_A + 1 \cdot e \cdot (1-S_A)$

$= S_A(1-e) + (1-S_A)e$

$= S_A + e - 2eS_A$.

b) $P(O_i=B \mid G=AB) = P(O_i=B \mid E_i=0, S_i=B, G=AB) \cdot P(E_i=0) \cdot P(S_i=B)$

$+ P(O_i=B \mid E_i=1, S_i=A, G=AB) P(E_i=1) \cdot P(S_i=A)$

$= (1-e)(1-S_A) + e \cdot S_A$

$= 1 - e + 2eS_A - S_A$

d) $P(AB \mid O_1 \cdots O_N) = \dfrac{P(O_1 \cdots O_N \mid G=AB) P(G=AB)}{P(O_1 \cdots O_N \mid AB) P(AB) + P(O_1 \cdots O_N \mid BB) P(BB) + P(O_1 \cdots O_N \mid AA) P(AA)}$ → abbreviated as $P(AB)$

From a), b), likelihoods are: $P(O_1 \cdots O_N \mid AB) = \prod_{i:O_i=A} P(O_i=A \mid AB) \prod_{i:O_i=B} P(O_i=B \mid AB)$

N_A: # of A observed,

N_B: # of B observed $= N - N_A$

$= (S_A + e - 2eS_A)^{N_A} (1 - e + 2eS_A - S_A)^{N_B}$

and $P(O_1 \cdots O_N \mid AA) = \prod_{i:O_i=A} P(O_i=A \mid AA) \cdot \prod_{i:O_i=B} (O_i=B \mid AA) = e^{N_B}(1-e)^{N_A}$

$P(O_1 \cdots O_N \mid BB) = e^{N_A}(1-e)^{N_B}$

Thus $P(AB \mid O_1 \cdots O_N) = \dfrac{[(S_A + e - 2eS_A)^{N_A} (1 - e + 2eS_A - S_A)^{N_B}] P(AB)}{[(S_A + e - 2eS_A)^{N_A} (1 - e + 2eS_A - S_A)^{N_B}] P(AB) + e^{N_B}(1-e)^{N_A} P(AA) + e^{N_A}(1-e)^{N_B} P(BB)}$

$= \dfrac{[(S_A + e - 2eS_A)^{N_A} (1 - e + 2eS_A - S_A)^{N-N_A}] P(AB)}{[(S_A + e - 2eS_A)^{N_A} (1 - e + 2eS_A - S_A)^{N-N_A}] P(AB) + e^{N-N_A}(1-e)^{N_A} P(AA) + e^{N_A}(1-e)^{N-N_A} P(BB)}$

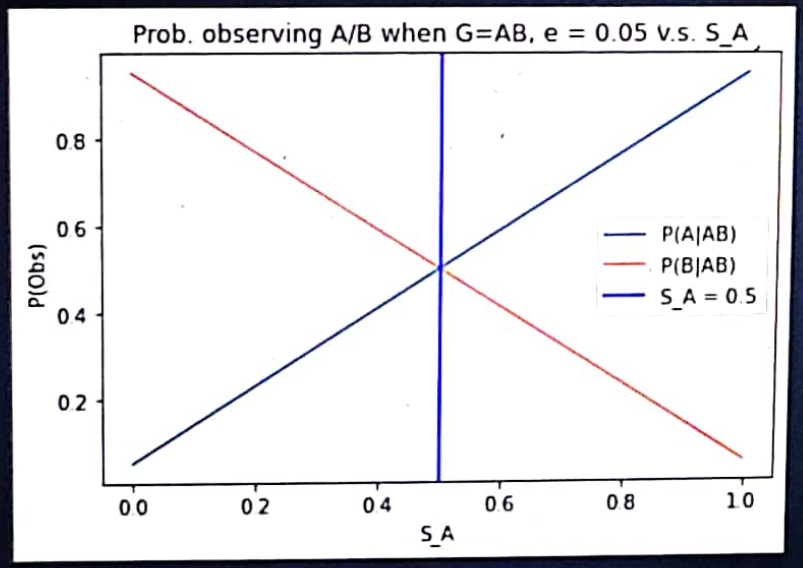
c), e) see codes.

```python
def PobsA_AB(S_A,err):
    return S_A + err - 2*err*S_A

def PobsB_AB(S_A,err):
    return 1 - err + 2*err*S_A - S_A #or 1 - PobsA_AB

S = np.linspace(0,1,100)
e = 0.05
ya = [PobsA_AB(sa,e) for sa in S]
yb = [PobsB_AB(sa,e) for sa in S]
plt.plot(S,ya,label = "P(A|AB)")
plt.plot(S,yb,label = "P(B|AB)")
plt.axvline(x = 0.5, color = 'b', label = 'S_A = 0.5')
plt.xlabel("S_A")
plt.legend()
plt.ylabel("P(Obs)")
plt.title("Prob. observing A/B when G=AB, e = 0.05 v.s. S_A")
```

✓ 0.1s

Pyth

```
Text(0.5, 1.0, 'Prob. observing A/B when G=AB, e = 0.05 v.s. S_A')
```



- When sampling rates are the same for A and B strands (=0.5), the probabilities of observing A and B are the same given the true genotype is AB
- As the chance of sampling the A strand goes up, the probability of observing A in reads, given the true genotype is AB and the error rates are the same for A&B strands, goes up; while the probability of observing B in the reads goes down.

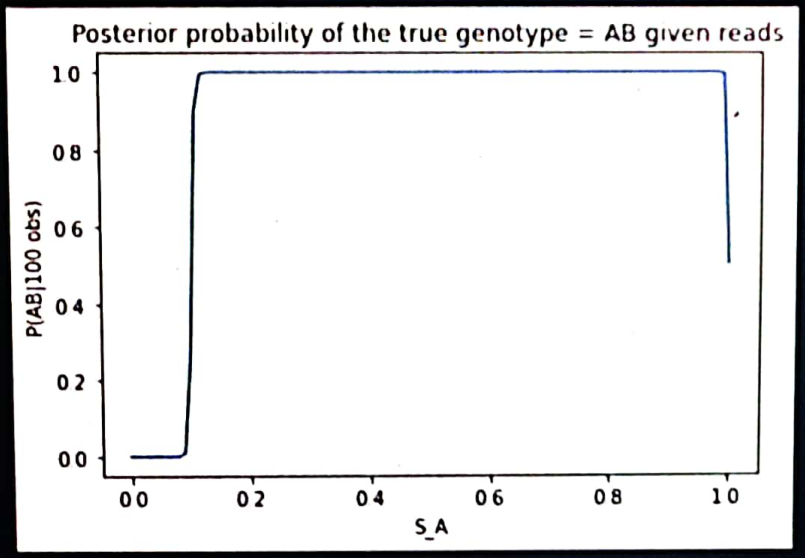
Given the assumptions and the reads, the prosterior probability of the true genotype G = AB is:

$$P(AB|O_1...O_{100}) = \frac{(S_A+0.05-2*0.05*S_A)^{60}*(1-0.05+2*0.05*S_A-S_A)^{40}}{(S_A+0.05-2*0.05*S_A)^{60}*(1-0.05+2*0.05*S_A-S_A)^{40}+0.05^{40}*0.95^{60}+0.05^{60}*0.95^{40}}$$

```
def post(S_A):
    ab = (S_A + 0.05 - 0.1 * S_A )**60 * (1 - 0.05 + 0.1 * S_A - S_A)**40
    aa = (0.05**40)*(0.95**60)
    bb = (0.05**60)*(0.95**40)
    return ab/(aa+ab+bb)

x = np.arange(0, 1.01, 0.01).tolist()
plt.plot(x,[post(a) for a in x])
plt.ylabel("P(AB|100 obs)")
plt.xlabel("S_A")
plt.title("Posterior probability of the true genotype = AB given reads")
```

[51] ✓ 0.2s Python

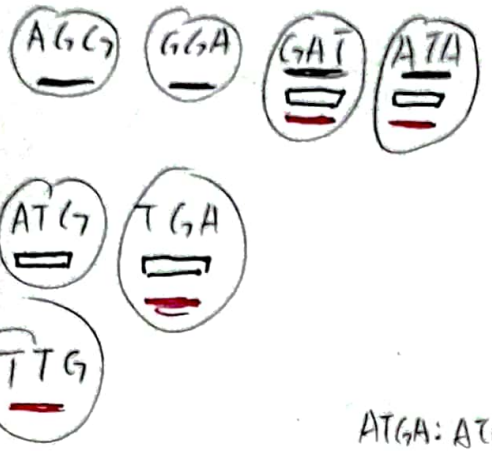
```
Text(0.5, 1.0, 'Posterior probability of the true genotype = AB given reads')
```



- Given 60 reads are A, 40 reads are B, and the error rates = 0.05 for both A and B strands sampled, the posterior probability of the true genotype = AB rises shaprly when P(S) > 0.1, then remains pretty much at 1 until the P(S) goes over 0.9.
- Intuitively, because the reads are pretty much half-and-half, and the error rates are very low, we could infer (which is what the posterior probability is doing) that the true genotype shouldn't be homozygous, unless the sampling rates are very biased. For example, if the chance of sampling B is very high if the true genotype is AB (S_A closes to 0), but we are still observing so many A in our reads tells us that the true genotype is more likely to be AA.

② Alignment (Pseudo)

a) k=3



b) length = 4, k=3

Matched:

AGGA, GGAT, GATA

ATGA, TTGA, TGAT.

Unmatched: $4^4 - 6 = 250$

No errors !!!

(if no skipping / backtracking are implemented when failed to match.

ATGA: ATG ∩ TGA = t_2 AGGA: AGG ∩ GGA = t_1

TTGA: TTG ∩ TGA = t_3 GGAT: GGA ∩ GAT = t_1

TGAT: TGA ∩ GAT = t_2, t_3 GATA: GAT ∩ ATA = t_1, t_2, t_3

c) if read length = 5

- It changed. Because t_1 & t_2 or t_1 & t_3 do not share a k-mer with the same sequence, while t_2 & t_3 does (TGATA).
- So the total number of equivalen classes would reduce to 5 (including unmatched), with the [t_1, t_2, t_3] label out of the picture
- The counts would also be different. All matched's become 1, while the unmatched increases by 2.

counts

Equivalen classes	b/	c/	d/
t_1	2	1	2
t_1, t_2, t_3	1	0	1
t_2	1	1	1
t_2, t_3	1	1	1
t_3	1	1	1
unmatched	250	1020	250
Total possible reads	256	1024	256

no errors.

d) k=4, length of read = 4

AGGA GGAT GATA

ATGA TGAT

TTGA

see the d/ column

c) Based on the table, a larger k for the same read length of 4 does not seem to change equivalence class counts / help disambiguate

② Generative models

G1 – S – G2 – M cell states, denoted by S.
↘ δ, P(death) => obs = 0

a) Let C be the total number of cells sampled. We decide for each cell to be generated:

① Cell types, from K possible classes • $T_i \sim$ Categorical(P_T)
↑ probability/proportion of each cell type

② Cell states, • $S_i \sim$ Categorical(P_S)
↑ probability of cells being at a certain cell state.
S = G1, S, G2, M.
*Assume: cell state distribution is independent of types.

③ Barcodes
• barcode #: $A_i \sim$ Poisson(λ_A) (average # of barcodes in a cell.)
• barcode: $B_i | A_i \sim$ Multi.(A_i, P_B) what barcodes get assigned follows multinomial where A_i is the total number of barcodes in the cell modeled by Poisson, and P_B is a vector of probabilities of a specific barcode being chosen.

④ Cell Death
*Assume: By the end of the process, M state cells randomly die with a probability of δ. Dead cells give no observations of gene expressions.

• $D_i \sim$ Bernoulli(δ) Probability of cell death of a M-stage cell.

• $D_i | S_i \sim \begin{cases} \text{Bernoulli}(\delta), & S_i = M \\ 0, & S_i \neq M \end{cases}$ is dependent on cell state

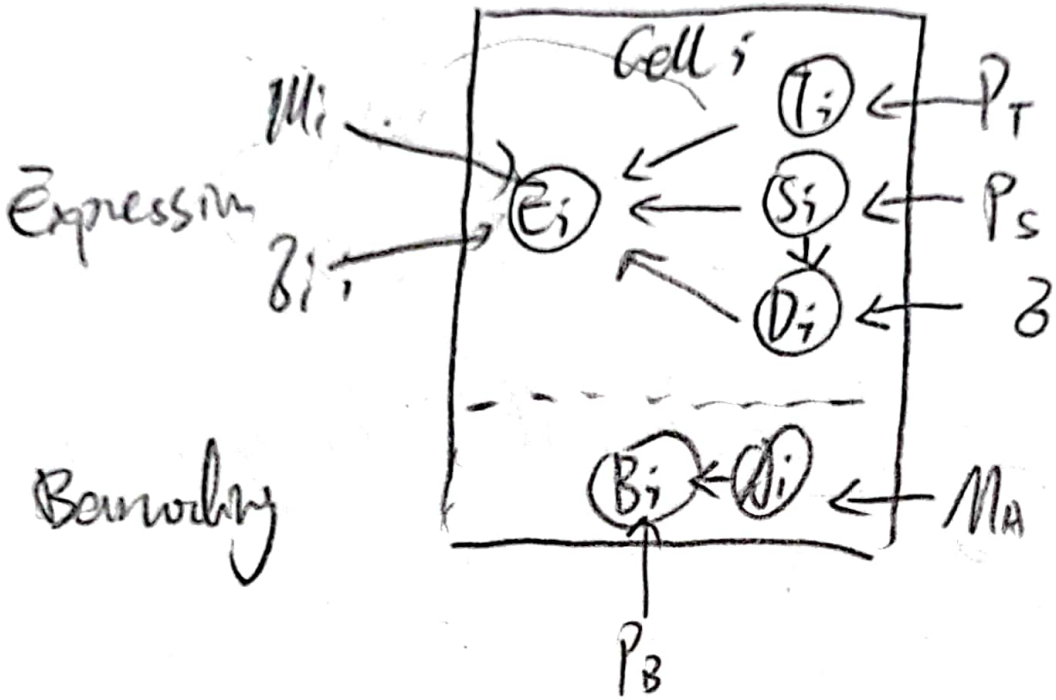
⑤ Gene expression,
would be zero if the cell dies otherwise would follow normal distributions depending on the cell states and the cell types

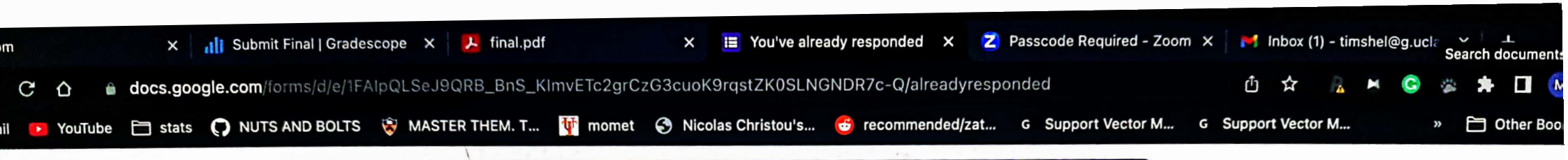
• $E_i | D_i \sim \begin{cases} 0, & D_i = 1 \\ N(\mu_{S_i}, \sigma^2_{S_i}), & D_i = 0 \end{cases}$
↑ affected by cell state & cell type.

b) $P(E_i) = P(E_i | D_i, S_i, T_i) P(D_i | S_i) P(S_i) P(T_i)$
$P(B_i) = P(B_i | A_i) P(A_i)$

E_i: gene expression, D_i: cell death, S_i: cell state, T_i: cell type, A_i: total available barcodes, B_i: barcode for the cell

c) Plate Model





You've already responded

You can fill out this form only once.

Try contacting the owner of the form if you think this is a mistake.