Advanced NGS Analysis (Day 1)
Session II

Lyda Hill department of Bioinformatics 2022 Nanocourse Series

Date & Time: June 27-28: 9AM-5PM (NB2.100A)

Course Instructors: Bo Li, Daehwan Kim, Christopher Chaney, & **Micah Thornton (micah.Thornton@utsouthwestern.edu)**

UTSouthwestern
Medical Center

Day 1: RNA-Seq Analysis Using Pseudo/Quasi-Alignment and Expectation Maximization (Kallisto, Salmon).

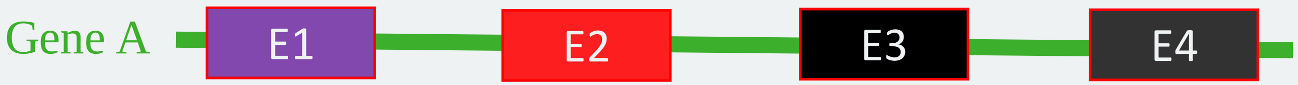
UTSouthwestern
Medical Center

Part 1: Pseudo and Quasi Alignment & Quantification Resolution

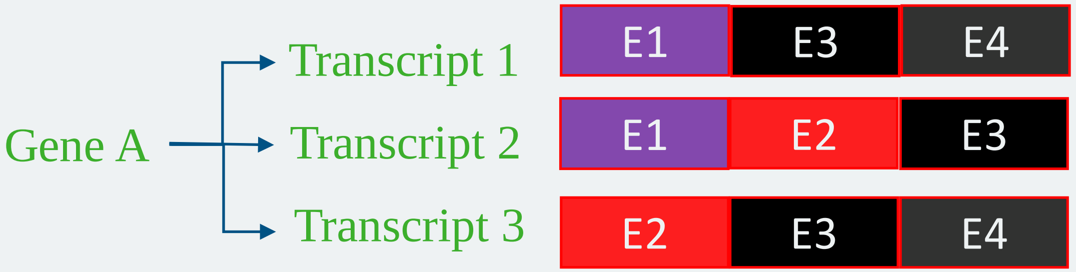
UTSouthwestern
Medical Center

RNA-Seq Experiments (Key Difference from DNA)

- In DNA-Seq, we usually align ‘NGS sequencing reads’ to a reference genome.
  - These sequences include all exons and introns, in fixed order.



- In RNA-Seq on the other hand we might try to align to a ‘transcriptome’.
  - These include a set of ‘transcripts’ which contain different possible sequences encoded by the same gene



UTSouthwestern
Medical Center

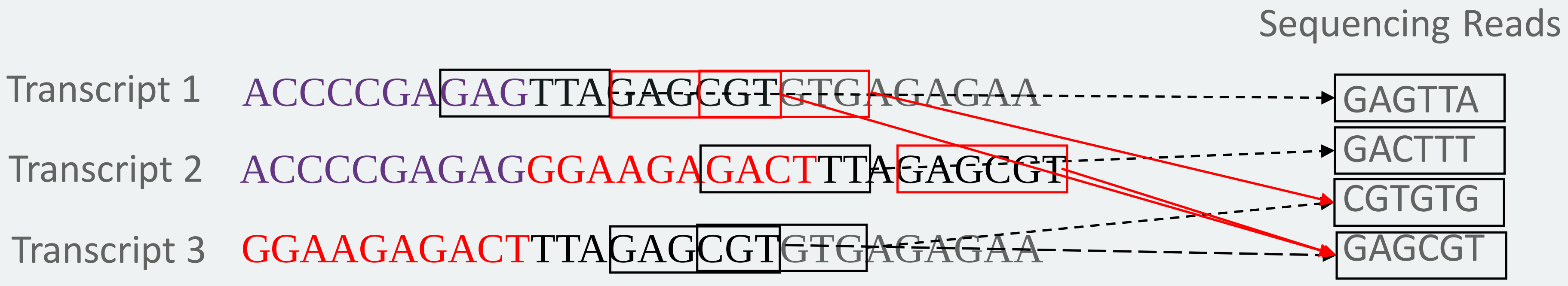
RNA-Seq Experiments (Key Questions)

- In RNA-Seq experiments researchers hope to answer questions such as:
  1. How do abundances of a particular gene transcripts in the same population vary?
     - Ex. Do subjects with a high abundance of transcript A, tend also to exhibit relatively high abundances of transcript B?
  2. Do the abundances of gene transcripts vary with some observable phenotypic characteristic?
     - Ex. Do patients with latent state tuberculosis exhibit more abundance of transcript A than those with active state?
  3. What constitutes the most likely genetic transcript profile for a particular subject?
     - Ex. Is Transcript A more abundant in this subject?
  4. Etc …

- So if we do not have exact *positional* alignments for reads, that is okay, as long as we are able to determine the most likely transcript that they came from. (Expectation Maximization for Multinomial Data)

RNA-Seq Experiments Overview

- In some RNA-Seq Experiments the *actual* alignment of reads may not be measurable in some cases.
- For instance, consider the following small example,


Sequencing Reads
Transcript 1
ACCCCGAGAGTTAGAGCGTGTGAGAGAA
Transcript 2
ACCCCGAGAGGGAAGAGACTTTAGAGCGT
Transcript 3
GGAAGAGACTTTAGAGCGTGTGAGAGAA
GAGTTA
GACTTT
CGTGTG
GAGCGT

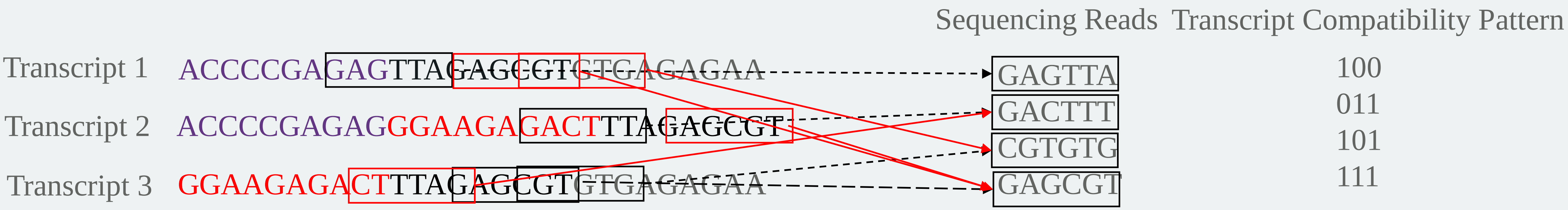
- It is possible that many RNA-Seq reads might be compatible with the same transcripts.
- Therefore for each of the RNA-Seq reads, the *actual* alignment is not directly recoverable.

RNA-Seq Experiments Overview

- Instead of working directly with the sequences to produce *positional* alignments (the exact position where the read came from), coarsened compatibility patterns may be observed for the transcripts the read is compatible with.
- From the previous example,

RNA-Seq Experiments Overview

- Instead of working directly with the sequences to produce *positional* alignments (the exact position where the read came from), coarsened compatibility patterns may be observed for the transcripts the read is compatible with.
- From the previous example,



Transcript 1 ACCCCGAGAGTTAGAGCGTGTGAGAGAA

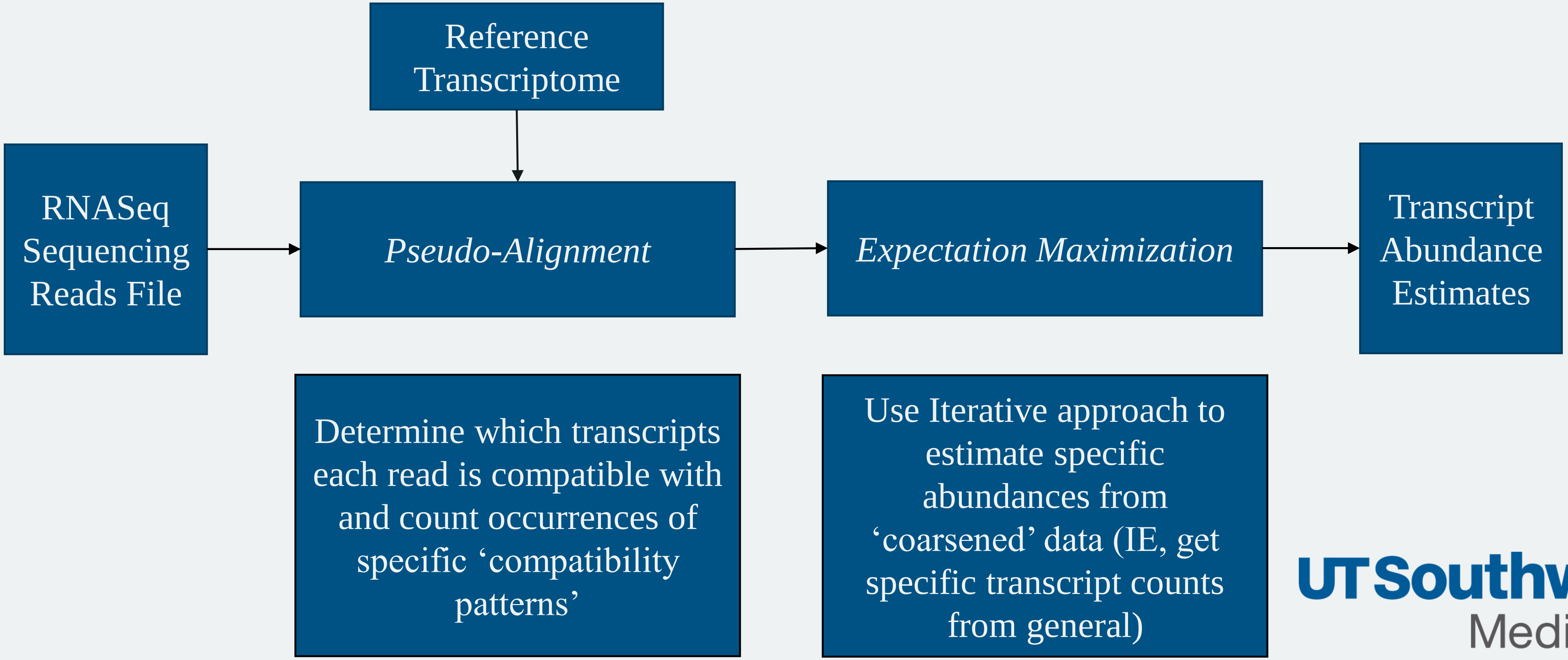
Transcript 2 ACCCCGAGAGGGAAGAGACTTTAGAGCGT

Transcript 3 GGAAGAGACTTTAGAGCGTGTGAGAGAA

Sequencing Reads	Transcript Compatibility Pattern
GAGTTA	100
GACTTT	011
CGTGTG	101
GAGCGT	111

RNA-Seq Experiments Overview

- Now suppose that there are 10 Million such reads aligning to the transcriptome.
- Since *positional* alignment information may not allow the discernment of transcripts,
  - Not necessary to determine and report such information.
- Instead, many RNA-Seq transcript quantification tools use the following general procedure:



Reference Transcriptome

RNASeq Sequencing Reads File

Pseudo-Alignment

Expectation Maximization

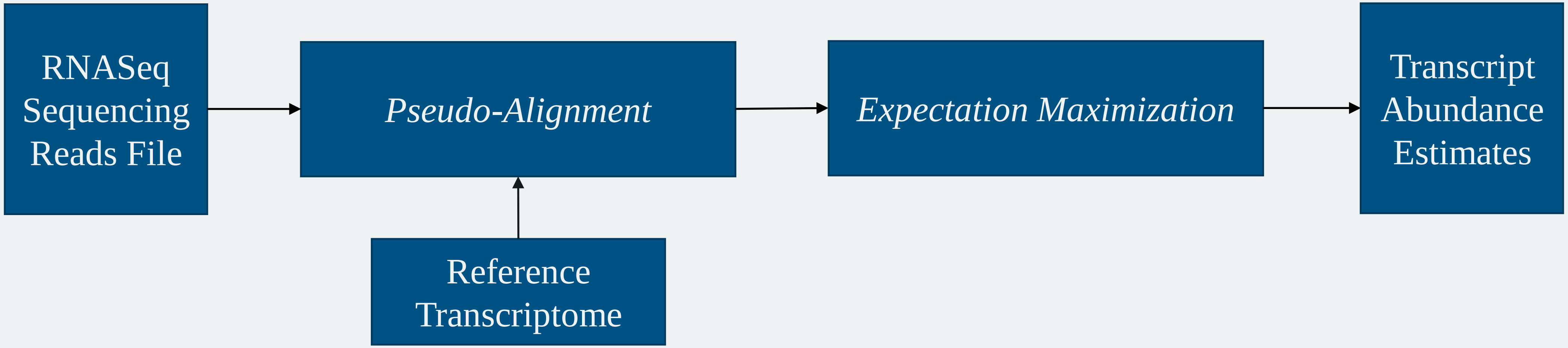
Transcript Abundance Estimates

Determine which transcripts each read is compatible with and count occurrences of specific ‘compatibility patterns’

Use Iterative approach to estimate specific abundances from ‘coarsened’ data (IE, get specific transcript counts from general)

UTSouthwestern Medical Center

RNA-Seq Experiments Overview


RNASeq Sequencing Reads File
Pseudo-Alignment
Expectation Maximization
Transcript Abundance Estimates
Reference Transcriptome

- First data is collected from the RNA of the subject(s) under study using NGS technologies (short read sequencing)
- Compatibility for each read with each transcript in a *Reference Transcriptome* is determined by some Pseudo-Alignment procedure.
- Expectation Maximization for abundances of the true transcript counts X_i are determined from the coarsened pattern counts Y_i.
- Estimates of the parameters of the distribution of Y_i (for $i = 1, \dots, N_t$), γ_i, are presented as abundance estimates, and multiplied by the number of reads N_r for expectations.

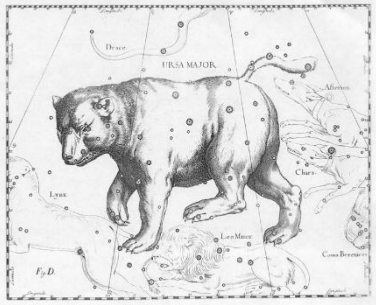
UTSouthwestern
Medical Center

RNA-Seq Abundance Estimation: Kallisto

- First data is collected from the RNA of the subject(s) under study using NGS technologies (short read sequencing)
- Compatibility for each read with each transcript in a *Reference Transcriptome* is determined by some Pseudo-Alignment procedure.
- Expectation Maximization for abundances of the true transcript counts X_i are determined from the coarsened pattern counts Y_i.
- Estimates of the parameters of the distribution of Y_i (for $i = 1, \dots, N_t$), γ_i, are presented as abundance estimates, and multiplied by the number of reads N_r for expectations.
- Sometimes TPM (Transcripts per million reads) are also presented.

- Two popular procedures which implement this approach are Salmon and Kallisto.
  - These will be presented and a demonstration given.
  - Tomorrow we will walk through installation of these software and their usage, and a third software will be demonstrated*.

Salmon



Kallisto

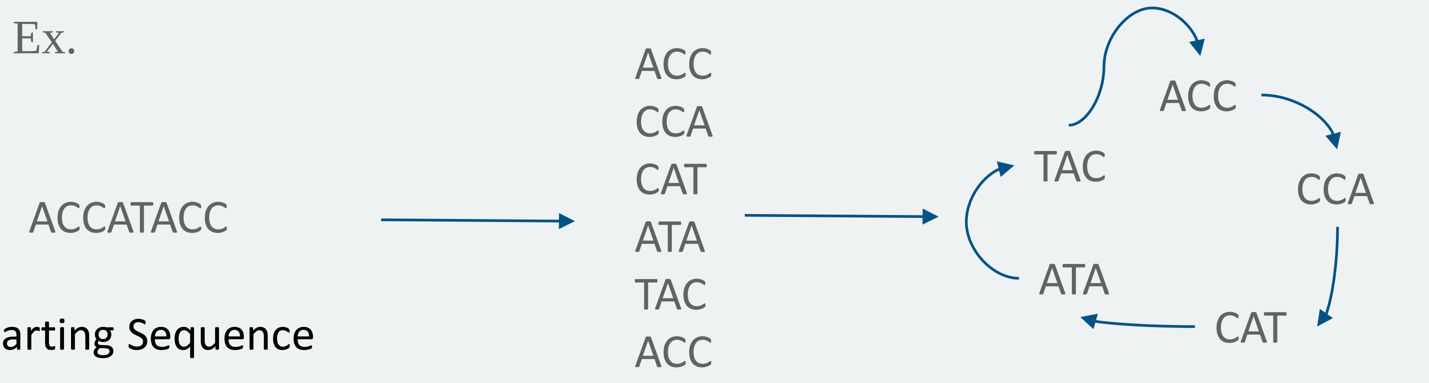
* we are creating a new tool called H2Q which takes into account SNP variability between alleles to provide allele specific transcript quantification results

Kallisto: Understanding the Algorithm

- As previously stated, the RNA-Seq quantification algorithms used in state of the art applications generally follow two steps:
    1. “Pseudo-Alignments” indicating the subset of transcripts which are ‘compatible’ with each read are determined.
    2. The Expectation Maximization algorithm is applied to determine the Maximum Likelihood Estimators for the Multinomial Proportions associated with each transcript.

- Kallisto uses a transcriptome de Brujin graph to determine which transcripts are compatible with each read (pair).
    - The De Brujin Graph represents sequences by connecting nodes of subsequences (in this case we call them k-mers, where k denotes the size.

    - Ex.



Starting Sequence

3-mers

Simple De Brujin Graph

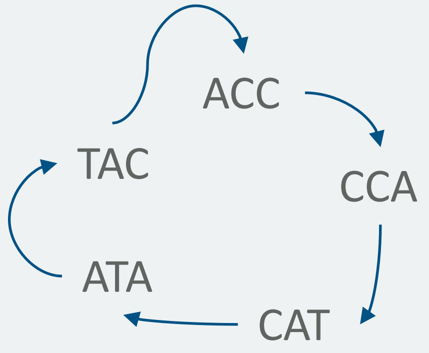
Kallisto: Understanding the Algorithm (Pseudo-Alignment)

ACCATACC

Starting Sequence

ACC
CCA
CAT
ATA
TAC
ACC

3-mers



Simple De Brujin Graph

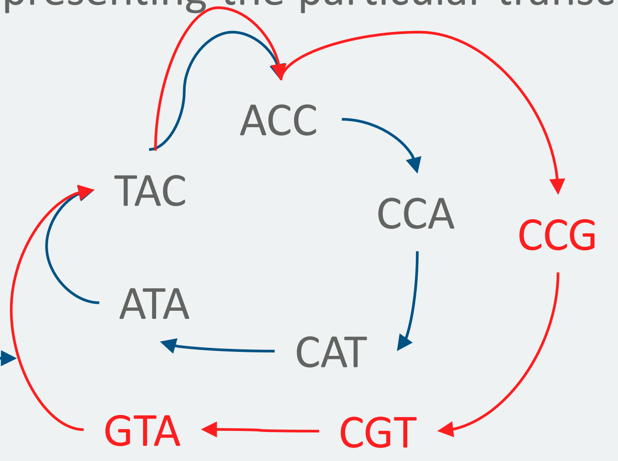
- Extending the De Brujin procedure to representing transcriptomes can be accomplished by adding “colors” or categorical weights to the edges of the De Brujin graph representing the particular transcript of alignment.

ACCATACC
ACCGTACC

Starting Transcriptome

3-mers

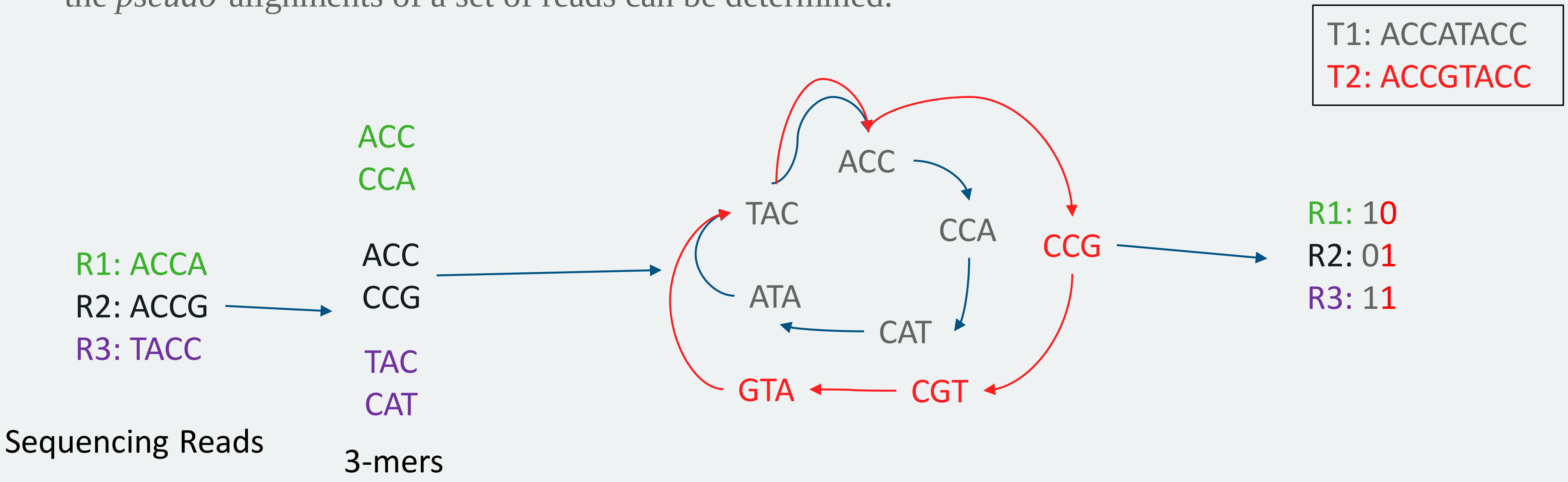
ACC
CCA
CCG
CAT
CGT
ATA
GTA
TAC
ACC



Simple De Brujin Graph

Kallisto: Understanding the Algorithm (Pseudo-Alignment)

- Given a new set of reads, the *k*-mers can be determined, and then using the De Brujin Graph Transcriptome, the *pseudo*-alignments of a set of reads can be determined.

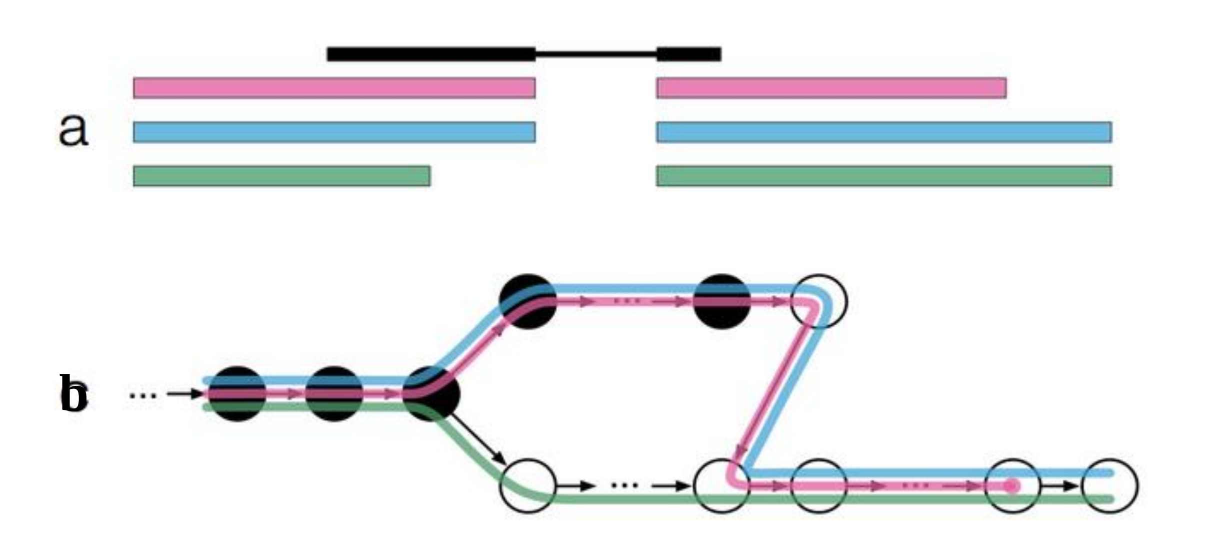


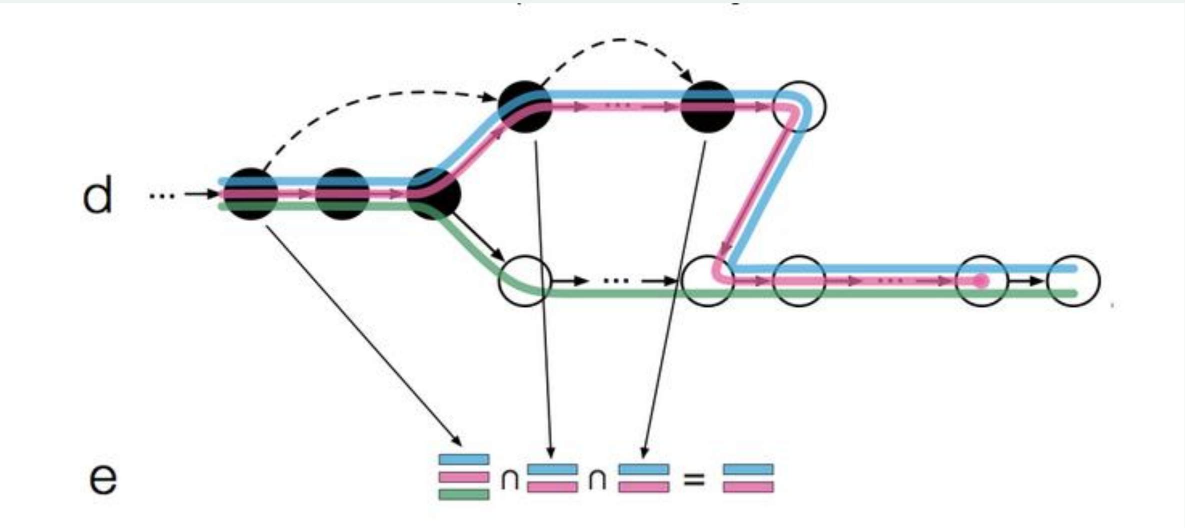
Comparison of 3-mers from reads to color DBG transcriptome allows the determination of compatibility counts.

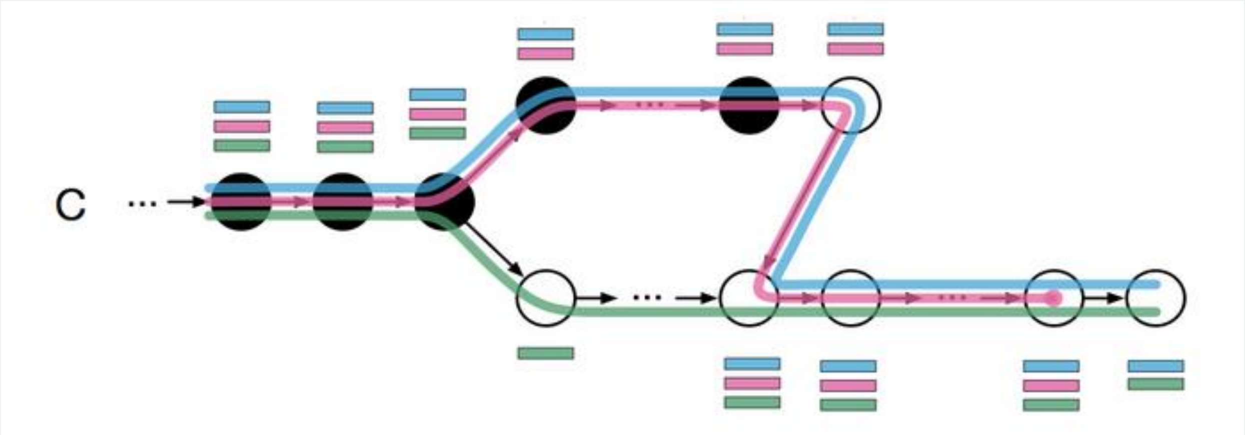
https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html

UTSouthwestern
Medical Center

Kallisto Pseudo-Alignment Example Backup





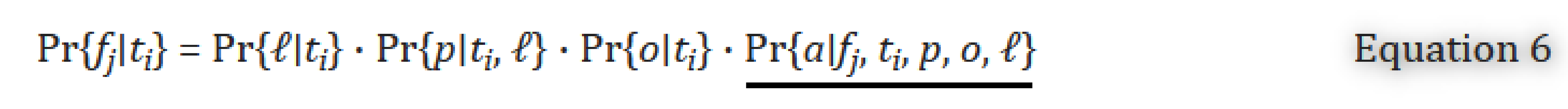


This is taken from Fong Chun Chan’s Blog post on “How Pseudoalignments Work in Kallisto.”

https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html

UTSouthwestern
Medical Center

Salmon: Understanding the Algorithm (Quasi-Mapping)



$$\Pr\{f_j|t_i\} = \Pr\{\ell|t_i\} \cdot \Pr\{p|t_i, \ell\} \cdot \Pr\{o|t_i\} \cdot \underline{\Pr\{a|f_j, t_i, p, o, \ell\}} \qquad \text{Equation 6}$$

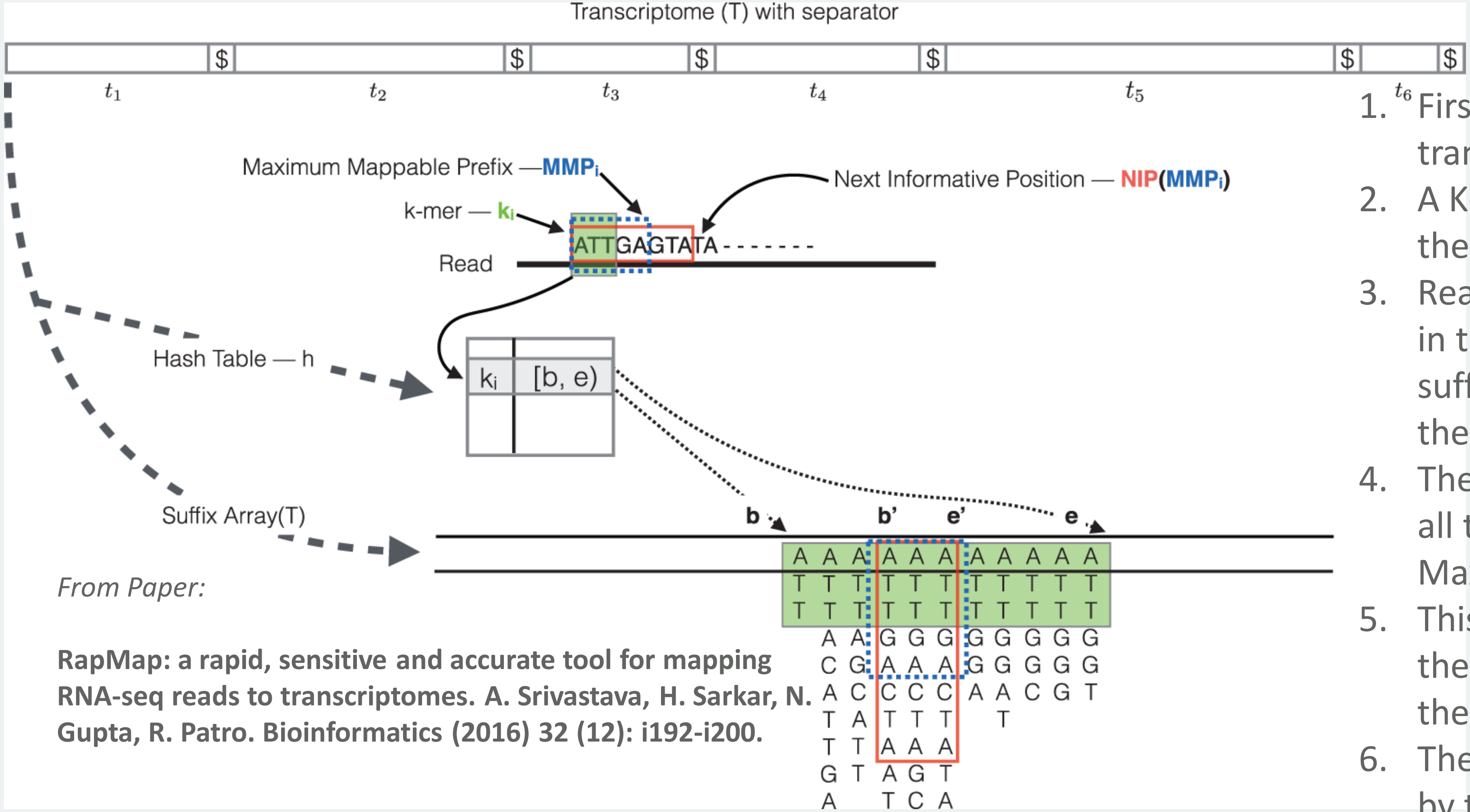
- If alignments are being used, then the Fragment-Transcript agreement model used by Salmon incorporates the true alignment information in this term (The Probability of generating alignment a of fragment f_j, being drawn from t_i, with position, orientation, and length, p, o, ℓ.
- In ‘Quasi-Mapping’ Mode however this term is fixed as 1.

- In Salmon, there are additional phases where parameters are calculated and utilized for determining a better transcript abundance quantification.

Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*. 2017;14(4):417-419. doi:10.1038/nmeth.4197

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5600148/#FD10

Salmon: Understanding the Algorithm (Quasi-Mapping)



From Paper:

RapMap: a rapid, sensitive and accurate tool for mapping RNA-seq reads to transcriptomes. A. Srivastava, H. Sarkar, N. Gupta, R. Patro. Bioinformatics (2016) 32 (12): i192-i200.

1. First a Suffix Array is built for the transcriptome
2. A K-mer Hash table for indexing into the the suffix array is constructed
3. Reads are scanned, and when k-mers in the hash table are found, all suffixes matching are extracted from the suffix array
4. The longest prefix in common among all the suffixes (called the MMP Maximal Matching Prefix) is found.
5. This is repeated for all kmers, and then the intersection of transcripts in the MMP
6. The Transcripts which are intersected by the associated MMP are provided as the compatible quasi-mappings

* From Article: **Quantification of transcript abundance using Salmon Introduction to bulk RNA-seq**

https://hbctraining.github.io/Intro-to-rnaseq-hpc-salmon-flipped/lessons/08_quasi_alignment_salmon.html

Questions about Pseudo-Alignment/Quasi-Mapping?

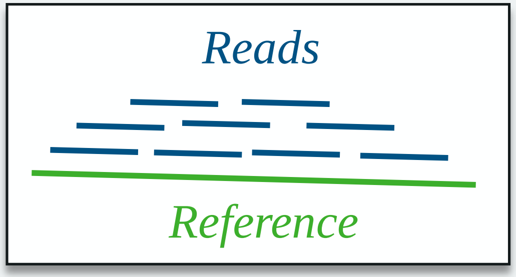
Break (10 Minutes)

Part 2: Expectation Maximization & Gene Transcript Quantification

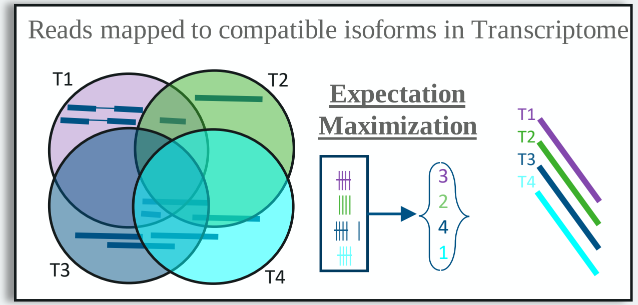
Pseudo/Quasi Alignment in RNA Experiments

- Sometimes the *exact* position of a sequencing read is not of critical import.
  - There are a few approaches for resolving the *approximate* location of a read.
  - Procedures work by determining the subset of *transcript isoforms* compatible with a read.
  - Two such approaches are known as:
    - Pseudo-Alignment
      - The Approach used by **Kallisto.**
      - Uses the De Brujin ('Deh-Broine') graph procedure.
    - Quasi-Alignment
      - The Approach used by **Salmon.**
      - Uses a K-mer Hash table and Suffix Array.

Typical 'DNA-Seq Like' Experiment



Typical 'RNA-Seq Like' Experiment



Recall that in most typical sequencing experiments we are dealing with a large collection of shorter subsequences called ***reads***, which we attempt to map to a larger sequence known as the ***reference***.

Resources – Kallisto (Pseudo-alignment)

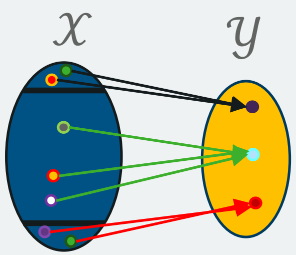
1. https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html (Higher Level Overview pseudo alignment)
2. https://www.youtube.com/watch?v=f-ecmECK7lw (Video Describing how To Build The De Brujin graph)
3. https://www.nature.com/articles/nbt.2023 (Nature Primer on Using De Brujin Graphs for Genomic Alignments).

Resources – Salmon (Quasi-alignment)

1. https://hbctraining.github.io/Intro-to-rnaseq-hpc-salmon-flipped/lessons/08_quasi_alignment_salmon.html (Higher Level Overview Quasi-Alignment)
2. https://academic.oup.com/bioinformatics/article/32/12/i192/2288985?login=true (RapMap Paper and Description).

Expectation Maximization (in general) – Incomplete Data & A Restricted Case

Many-to-one relationship



- Two general uses include:
  - determination of maximum likelihood estimates for parameters when missing data is present and
  - estimation of missing or otherwise incomplete data.

- In general, suppose that we would like to observe the values, $x_1, x_2, \dots x_n$, to determine something about the parameters of the random variable X which has sample space $\mathcal{X}$ as shown (top right).
  - However, we are only able to observe, $y_1, y_2, \dots y_n$, valuations of the random variable Y which has sample space $\mathcal{Y}$ onto which there exists a many-to-one mapping from $\mathcal{X}$.
    - In other words, there are multiple values possible to observe in $\mathcal{X}$ corresponding to the same value in $\mathcal{Y}$.

- Suppose, at first, that the distribution of $\boldsymbol{X}$ (note boldface indicates that $\boldsymbol{X}$ could be a vector quantity) is one of the exponential family of distributions generally denoted,

$$f_X(\boldsymbol{x}|\boldsymbol{\theta}) = b(\boldsymbol{x})e^{(\boldsymbol{\theta}\boldsymbol{t}(\boldsymbol{x})^{\boldsymbol{T}})}a(\boldsymbol{\theta})^{-1}$$

$\boldsymbol{\theta}$ is a parameter [column]-vector (of size r).
$\boldsymbol{t}(\boldsymbol{x})^{\boldsymbol{T}}$ is the sufficient statistic [row]-vector (of size r).
$a(\cdot), b(\cdot)$, are any arbitrary function.
e is the natural number.

See section II of the Dempster, Laird, Rubin paper mentioned below for more details about natural parameters.

UTSouthwestern Medical Center

These Expectation Maximization Notes Draw Heavily from “Maximum Likelihood from Incomplete Data via the EM Algorithm” by Dempster Rubin and Laird (https://www.jstor.org/stable/2984875)

Expectation Maximization (in general) – The Algorithm

- The “simple characterization” of the EM algorithm according to Dempster, Laird, and Rubin (DLR77) is:

(1) With $\theta^{(p)}$ indicating the estimate of θ at the p^{th} step of the algorithm, estimate the complete-data sufficient statistics $\boldsymbol{t}(\boldsymbol{x})$ by finding

$$\boldsymbol{t}^{(p)} = E\left(\boldsymbol{t}(\boldsymbol{x})\middle|\boldsymbol{y}, \boldsymbol{\theta}^{(p)}\right).$$

(2) Perform maximum likelihood estimation to determine $\theta^{(p+1)}$ from $\boldsymbol{t}^{(p)}$,

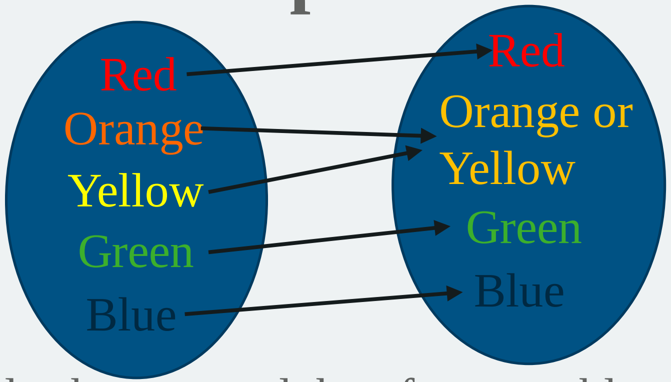
$$\boldsymbol{E}(\boldsymbol{t}(\boldsymbol{x})|\boldsymbol{\theta}) = \boldsymbol{t}^{(p)}.$$

- Proof of convergence to the maximum likelihood value is given the DLR77, as are details regarding further generalizations of the expectation maximization algorithm.
- The algorithm is broadly applicable in many cases, and not all of the applications have been discovered yet.

These Expectation Maximization Notes Draw Heavily from “Maximum Likelihood from Incomplete Data via the EM Algorithm” by Dempster Rubin and Laird (https://www.jstor.org/stable/2984875)

Expectation Maximization (in general) – A Multinomial Example

- Suppose that there are marbles of five colors in a bag.
  - Red marbles are denoted by ‘R’
  - Orange marbles are denoted by ‘O’
  - Yellow marbles by ‘Y’
  - Green marbles by ‘G’
  - Blue marbles by ‘B’



Red
Orange
Yellow
Green
Blue

Red
Orange or Yellow
Green
Blue

- Now, you personally cannot tell a difference between the orange and the yellow marbles by eye, and therefore are able to produce counts of four categories of marbles only (that is: “Red”, “Orange or Yellow”, “Green”, and “Blue”).

[EXAMPLE]

- Suppose it is known ahead of time that the proportions of the *actual* colors of each of the marbles are related via an unknown parameter π, such that for the unobservable true color of an arbitrarily selected marble i, denoted c_i (true color) given below induces a distribution on the observable o_i (observed color) follows this distribution:

$$P\left(c_i = \begin{pmatrix} \text{Red} \\ \text{Orange} \\ \text{Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}\right) = \begin{pmatrix} (1-\pi)/4 \\ \pi/4 \\ 1/2 \\ (1-\pi)/4 \\ \pi/4 \end{pmatrix} \Rightarrow P\left(o_i = \begin{pmatrix} \text{Red} \\ \text{Orange or Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}\right) = \begin{pmatrix} (1-\pi)/4 \\ 1/2+\pi/4 \\ (1-\pi)/4 \\ \pi/4 \end{pmatrix}$$

UTSouthwestern
Medical Center

These Expectation Maximization Notes Draw Heavily from “Maximum Likelihood from Incomplete Data via the EM Algorithm” by Dempster Rubin and Laird (https://www.jstor.org/stable/2984875)

Expectation Maximization (in general) – A Multinomial Example (Continued)

Suppose that we observe 197 marbles, and arrive at the following counts:

$$\begin{pmatrix} \text{R}-\text{Red: 18} \\ \text{OY}-\text{Orange or Yellow:125} \\ \text{G}-\text{Green: 20} \\ \text{B} - \text{Blue: 34} \end{pmatrix}$$

$$P\left(c_i = \begin{pmatrix} \text{Red} \\ \text{Orange} \\ \text{Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}\right) = \begin{pmatrix} (1-\pi)/4 \\ \pi/4 \\ 1/2 \\ (1-\pi)/4 \\ \pi/4 \end{pmatrix} \Rightarrow P\left(c_i = \begin{pmatrix} \text{Red} \\ \text{Orange or Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}\right) = \begin{pmatrix} (1-\pi)/4 \\ 1/2+\pi/4 \\ (1-\pi)/4 \\ \pi/4 \end{pmatrix}$$

$$x_j = \sum_{i=1}^{197} \mathbb{1}(c_i \equiv j) \qquad j \in \begin{pmatrix} \text{Red} \\ \text{Orange} \\ \text{Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}$$

$$y_t = \sum_{i=1}^{197} \mathbb{1}(o_i \equiv t) \qquad t \in \begin{pmatrix} \text{Red} \\ \text{Orange or Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}$$

- Let the *actual* color counts be denoted by the values $(x_1, x_2, x_3, x_4, x_5)$ such that x_1 corresponds to the count of marbles which were actually red, x_2 to those which were Orange, and so on…

- Let the observed color counts be denoted by the values (y_1, y_2, y_3, y_4) which are given in this example as (18,125,20,34).

- Furthermore, it is known that $y_2 = x_2 + x_3$.

- The Likelihood on π for the full data can be expressed as:

$$f(\boldsymbol{x}|\pi) = \frac{\left(\sum_{i=1}^{5} x_i\right)!}{\prod_{i=1}^{5}(x_i!)} \cdot \left(\frac{1-\pi}{4}\right)^{x_1} \cdot \left(\frac{\pi}{4}\right)^{x_2} \cdot \left(\frac{1}{2}\right)^{x_3} \cdot \left(\frac{1-\pi}{4}\right)^{x_4} \cdot \left(\frac{\pi}{4}\right)^{x_5}$$

- The *coarsened/incomplete* Likelihood on π for the full data can be expressed as:

$$g(\boldsymbol{y}|\pi) = \frac{\left(\sum_{i=1}^{4} y_i\right)!}{\prod_{i=1}^{4}(y_i!)} \cdot \left(1-\frac{\pi}{4}\right)^{y_1} \cdot \left(\frac{1}{2}+\frac{\pi}{4}\right)^{y_2} \cdot \left(1-\frac{\pi}{4}\right)^{y_3} \cdot \left(\frac{\pi}{4}\right)^{y_4}$$

UT Southwestern
Medical Center

DLR77 (https://www.jstor.org/stable/2984875)

Expectation Maximization (in general) – A Multinomial Example (E-Step)

- Clearly, due to the fact that a marble cannot *actually* be two colors simultaneously, there is no probability that any marble is *truly* both orange and yellow at the same time, therefore we may express the probability that a marble is orange or yellow as follows:

$$P\big(o_i = (\text{Orange or Yellow})\big) = P\left(c_i \in \binom{\text{Orange}}{\text{Yellow}}\right) = P(c_i = \text{Orange}) + P(c_i = \text{Yellow}) - P(c_i = \text{Yellow}\ \&\ c_i = \text{Orange})$$

$$= P(c_i = \text{Orange}) + P(c_i = \text{Yellow}) - 0 \ = \frac{\pi}{4} + \frac{1}{2}$$

- From here we can derive the expression for the maximum likelihood estimates of the unobserved counts for orange and yellow marbles (x_2, x_3) in terms of the observed count of "orange or yellow" marbles (y_2).

$$P\big(c_i = \text{Orange}\ |o_i = (\text{Orange or Yellow})\big) = \frac{P(o_i = (\text{Orange or Yellow})\ \&\ c_i = \text{Orange})}{P(o_i = (\text{Orange or Yellow}))} = \frac{P(c_i = \text{Orange})}{P\big(o_i = (\text{Orange or Yellow})\big)} = \frac{\frac{\pi}{4}}{\frac{\pi}{4} + \frac{1}{2}}$$

$$P\big(c_i = \text{Yellow}\ |o_i = (\text{Orange or Yellow})\big) = \frac{\frac{1}{2}}{\frac{\pi}{4} + \frac{1}{2}}$$

Therefore the conditional expectation of x_2 and x_3 are:

$$E(x_2|y_2) = y_2 \frac{\frac{\pi}{4}}{\frac{\pi}{4} + \frac{1}{2}} \quad \text{- and -} \quad E(x_3|y_2) = y_2 \frac{\frac{1}{2}}{\frac{\pi}{4} + \frac{1}{2}}$$

Suppose that we observe 197 marbles, and arrive at the following counts:

$$\begin{pmatrix} \text{R}-\text{Red: } 18 \\ \text{OY}-\text{Orange or Yellow:} 125 \\ \text{G}-\text{Green: } 20 \\ \text{B} - \text{Blue: } 34 \end{pmatrix}$$

These Expectation Maximization Notes Draw Heavily from "Maximum Likelihood from Incomplete Data via the EM Algorithm" by Dempster Rubin and Laird (https://www.jstor.org/stable/2984875)

UTSouthwestern
Medical Center

Expectation Maximization (in general) – A Multinomial Example (M-Step)

$$\Rightarrow x_1 + x_4 = x_2\hat{\pi} + x_5\hat{\pi} + x_1\hat{\pi} + x_4\hat{\pi}$$

- Recall that the full likelihood for the multinomial distribution was given by:

$$f(\boldsymbol{x}|\pi) = \frac{\left(\sum_{i=1}^{5} x_i\right)!}{\prod_{i=1}^{5}(x_i!)} \cdot \left(\frac{1-\pi}{4}\right)^{x_1} \cdot \left(\frac{\pi}{4}\right)^{x_2} \cdot \left(\frac{1}{2}\right)^{x_3} \cdot \left(\frac{1-\pi}{4}\right)^{x_4} \cdot \left(\frac{\pi}{4}\right)^{x_5}$$

$$\Rightarrow \log L(\pi|\boldsymbol{x}) = \log\frac{\left(\sum_{i=1}^{5} x_i\right)!}{\prod_{i=1}^{5}(x_i!)} + x_1\log\left(\frac{(1-\pi)}{4}\right) + x_2\log\left(\frac{\pi}{4}\right) + x_3\log\left(\frac{1}{2}\right) + x_4\log\left(\frac{(1-\pi)}{4}\right) + x_5\log\left(\frac{\pi}{4}\right)$$

- In this example, only x_2 and x_3 are unobservable, the rest are known:

$$\Rightarrow \frac{\partial \log L(\pi|\boldsymbol{x})}{\partial\,\pi} = x_1\left(\frac{4}{1-\pi}\right)\left(-\frac{1}{4}\right) + x_2\left(\frac{4}{\pi}\right)\left(\frac{1}{4}\right) + x_4\left(\frac{4}{1-\pi}\right)\left(-\frac{1}{4}\right) + x_5\left(\frac{4}{\pi}\right)\left(\frac{1}{4}\right) = \frac{x_1}{\pi-1} + \frac{x_2}{\pi} + \frac{x_4}{\pi-1} + \frac{x_5}{\pi}$$

$$\Rightarrow \frac{x_1}{\hat{\pi}-1} + \frac{x_2}{\hat{\pi}} + \frac{x_4}{\hat{\pi}-1} + \frac{x_5}{\hat{\pi}} = 0 \Rightarrow (x_2 + x_5)(1-\hat{\pi}) = (x_1 + x_4)\hat{\pi} \Rightarrow x_2 + x_5 - x_2\hat{\pi} - x_5\hat{\pi} = x_1\hat{\pi} + x_4\hat{\pi}$$

$$\Rightarrow x_2 + x_5 = (x_1 + x_2 + x_4 + x_5)\hat{\pi} \Rightarrow \hat{\pi} = \frac{(x_2 + x_5)}{x_1 + x_2 + x_4 + x_5}$$

$$\Rightarrow -x_1\hat{\pi} - x_4\hat{\pi} + x_1 + x_4 = x_2\hat{\pi} + x_5\hat{\pi}$$

$$\Rightarrow x_1 + x_4 = x_2\hat{\pi} + x_5\hat{\pi} + x_1\hat{\pi} + x_4\hat{\pi}$$

Suppose that we observe 197 marbles, and arrive at the following counts:

$$\begin{pmatrix} x_1 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 18 \\ 20 \\ 34 \end{pmatrix} \Rightarrow \hat{\pi} = \frac{x_2 + 34}{18 + x_2 + 20 + 34}$$

$$\begin{pmatrix} \text{R}-\text{Red: } 18 \\ \text{OY}-\text{Orange or Yellow:}125 \\ \text{G}-\text{Green: } 20 \\ \text{B}-\text{Blue: } 34 \end{pmatrix} \therefore \frac{1}{\hat{\pi}} = \frac{18 + x_2 + 20 + 34}{x_2 + 34} = 1 + \frac{38}{x_2 + 34} \Rightarrow \hat{\pi} = \frac{1}{1 + \frac{38}{x_2 + 34}}$$

UTSouthwestern Medical Center

Expectation Maximization (in general) – A Multinomial Example (Iteration)

- Taking the conditional expectations for the computation of x_2 and x_3 will depend on a particular estimation of π, an initial estimate ($\pi^{(0)}$) must be supplied to the algorithm to start the procedure, then conditional expectations for the missing (coarsened) data at the p^{th} step (where $p \in \{1,2,\ldots\}$) is given by:

[E – Step]

$$E_{(p)}(x_2|y_2) = y_2 \frac{\frac{\pi^{(p-1)}}{4}}{\frac{\pi^{(p-1)}}{4} + \frac{1}{2}} \text{ - and - } E_{(p)}(x_3|y_2) = y_2 \frac{\frac{1}{2}}{\frac{\pi^{(p-1)}}{4} + \frac{1}{2}}$$

[M – Step]

$$\widehat{\pi^{(p)}} = \frac{1}{1 + \frac{38}{E_{(p)}(x_2|y_2) + 34}}$$

- Convergence Criteria:
  - Generally we use relative convergence criteria (when the change in the parameters from step p to step $p + 1$ falls below a relative tolerance ε_R) to determine when to stop iterating, for instance, the iteration will continue until:

[Convergence]

$$\left(\frac{1}{1 + \frac{38}{E_{(p)}(x_2|y_2) + 34}} - \frac{1}{1 + \frac{38}{E_{(p-1)}(x_2|y_2) + 34}} \right)^2 \leq \varepsilon_R$$

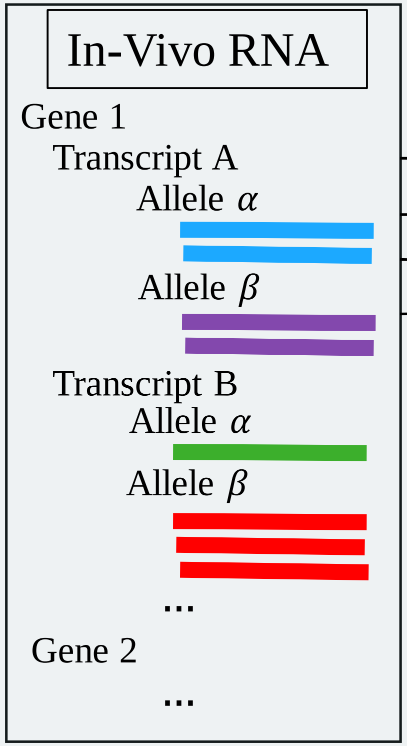
Expectation Maximization (Genetic Abundance Estimation)

- We observe N_r RNA-seq reads from an experiment involving a transcriptome of size T.
  - Each of the N_r reads came specifically from *only* one of the T categories.

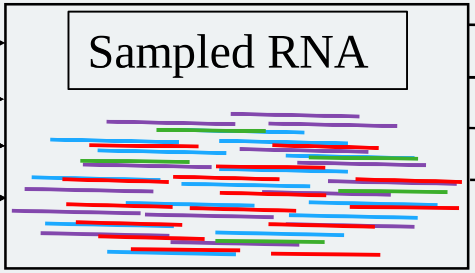
True Abundances

TX	CNT
1.A.α	x_1
1.A.β	x_2
1.B.α	x_3
1.B.β	x_4

We are interested in the vector of true abundances, and estimate these through EM.



Genes, transcripts, and alleles are initially differentially expressed in a subject of interest.



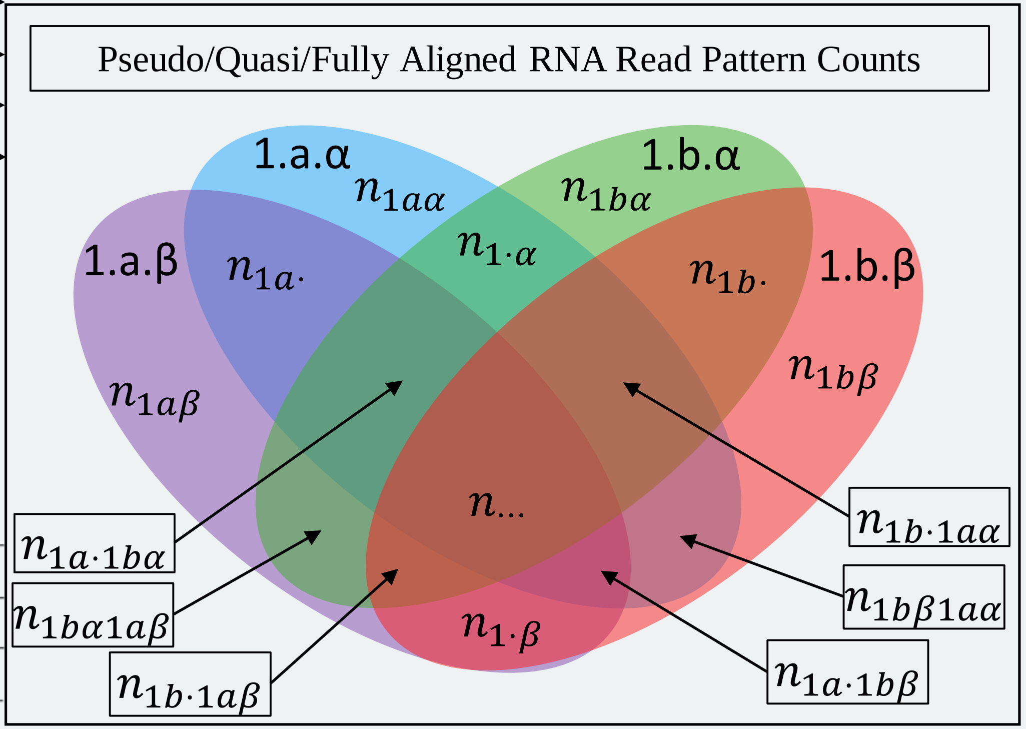
RNA reads are sampled from the original subject's genetic material.

Expectation Maximization

[E-Step] $E(\vec{\boldsymbol{x}} \mid \vec{\boldsymbol{n}})$

[M-Step] $\dfrac{\partial L(\hat{\boldsymbol{\gamma}} \mid \vec{\boldsymbol{x}})}{\partial \hat{\boldsymbol{\gamma}}} = \mathbf{0}$

Aligned Reads allow counts of compatible transcript patterns to be captured.



Where $\hat{\boldsymbol{\gamma}}$ is the vector of proportions associated with the transcriptome, and $\vec{\boldsymbol{n}}$ is the set of pattern counts from alignment.

UTSouthwestern
Medical Center

Expectation Maximization (Genetic Abundance Estimation) [M-Step]

- Clearly, the distribution of the reads among their true transcript sources can be modeled as multinomial.
  - The probability distribution of the count vector of the true abundances, $\vec{\boldsymbol{X}}$**,** is

$$\Pr\left(\vec{\boldsymbol{X}} = \vec{\boldsymbol{x}}\right) = \frac{\left(\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{x_i}\right)!}{\prod_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{x_i})!} \prod_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{\gamma_i^{x_i}}$$

- This probability distribution function doubles as the Likelihood for the parameter vector $\vec{\boldsymbol{\gamma}}$ under the observed data $\vec{\boldsymbol{x}}$.

$$L(\vec{\boldsymbol{\gamma}}|\vec{\boldsymbol{x}}) = \frac{\left(\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{x_i}\right)!}{\prod_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{x_i})!} \prod_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{\gamma_i^{x_i}} \Rightarrow \ell(\vec{\boldsymbol{\gamma}}|\vec{\boldsymbol{x}}) = \log \frac{\left(\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{x_i}\right)!}{\prod_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{x_i})!} + \sum_{i=1}^{N} x_i \log \gamma_i \Rightarrow \frac{\partial \ell(\vec{\boldsymbol{\gamma}}|\vec{\boldsymbol{x}})}{\partial \gamma_i} = \frac{x_i}{\gamma_i} \qquad \sum_{i=1}^{N} \gamma_i = 1$$

- Do not forget that there is an inherent constraint on the parameter space (the sum of all proportions must be one).
  - We must optimize $\ell(\vec{\boldsymbol{\gamma}}|\vec{\boldsymbol{x}})$ subject to the constraint: $\sum_{i=1}^{N} \gamma_i = 1$.
  - This is accomplished by using the method of Lagrange multipliers.

$$\ell'(\vec{\boldsymbol{\gamma}}, \lambda) = \ell(\vec{\boldsymbol{\gamma}}) + \lambda\left(1 - \sum_{i=1}^{N} \gamma_i\right) \Rightarrow \frac{\partial \ell'(\vec{\boldsymbol{\gamma}}, \lambda)}{\partial \gamma_i} = \frac{x_i}{\gamma_i} - \lambda \Rightarrow \frac{x_i}{\widehat{\gamma_i}} - \lambda = 0 \Rightarrow \widehat{\gamma_i} = \frac{x_i}{\lambda}$$

$$\widehat{\gamma_i} = \frac{x_i}{\lambda} \Rightarrow \sum_{i=1}^{N} \frac{x_i}{\lambda} = 1 \Rightarrow \frac{1}{\lambda} \sum_{i=1}^{N} x_i = 1 \Rightarrow \lambda = \sum_{i=1}^{N} x_i = N_r \Rightarrow \widehat{\gamma_i} = \frac{x_i}{N_r}$$

UTSouthwestern
Medical Center

Expectation Maximization (Genetic Abundance Estimation) [E-Step]

- The algorithm calculates the conditional expectation for missing x_i values during the E-Step.
  - If x_i is missing, we must first determine a valid estimate of x_i using the parameter estimated from the previous step (or the initial value used).
  - Instead of observing the vector $\vec{\boldsymbol{x}}$ directly, we observe the pattern count vector $\vec{\boldsymbol{n}}$.
  - Let the elements of $\vec{\boldsymbol{n}}$, $(n_1, n_2, \ldots, n_{N_k})$ be indexed by j, which runs from 1 to the number of unique compatibility patterns (N_k).
  - The conditional expectations of the missing components of $\vec{\boldsymbol{x}}$ are computed using the elements of $\vec{\boldsymbol{n}}$**,** which compose the counts of patterns including those same missing components.
    - For example, if x_1 is missing, but we determine there are reads which align to x_1 as well as others, say n_1, n_3, and, n_5 are compatible with transcript 1, then each of these quantities would be used to compute the conditional expectation of the missing value.
  - Note, we either begin with $\gamma^{(0)}$, or have iterated to the pth step, and have $\gamma^{(p-1)}$.
  - The conditional expectation of the missing value, x_i, is determined by considering the observations of those elements of $\vec{\boldsymbol{n}}$ which contain alignments to transcript i**.**
  - **Let the indicator $\boldsymbol{\psi_{ij}}$ be 1 if read $\boldsymbol{i}$ is present in compatibility pattern $\boldsymbol{j}$ and 0 otherwise.**

$$E_{(p)}(\, x_i \mid \vec{\boldsymbol{n}}\,) = \frac{\gamma_i^{(p-1)}}{\sum_{j=1}^{N_p} \psi_{ij} n_j \gamma_i^{(p-1)}} N_r$$

Expectation Maximization (Genetic Abundance Estimation)

- The EM algorithm amounts to applying these two operations in alternative order until there is convergence in the parameter vector.

[E-Step] $$E_{(p)}(\, x_i \mid \vec{\boldsymbol{n}}\,) = \frac{\gamma_i^{\widehat{(p-1)}}}{\sum_{j=1}^{N_k} \psi_{ij} n_j \gamma_i^{\widehat{(p-1)}}} n_j$$

[M-Step] $$\widehat{\gamma}_i^{\,(p)} = \frac{E_{(p)}(\, x_i \mid \vec{\boldsymbol{n}}\,)}{N_r}$$

- The EM Algorithm will achieve convergence when the change from step $p-1$ to p is below some user selected relative tolerance ε_r.

[Convergence Criteria] $$\widehat{\gamma}_i^{\,(p)} - \widehat{\gamma}_i^{\,(p-1)} \leq \varepsilon_r$$

- Note the Expectation Maximization algorithm for Multinomial count data (as above) can be applied in a general case. This algorithm is implemented in multiple software packages available for use, but we have created a general version (For a copy, please request via email at this stage).

UTSouthwestern
Medical Center

Expectation Maximization (Multinomial algorithm example)

Suppose we have true abundances
Transcript 1 : 500 (0.5)
Transcript 2 : 200 (0.2)
Transcript 3 : 300 (0.3)

But that we can only observe whether reads are in the following:
(T1,T3): 300+250 = 550
(T1,T2): 200+200 = 400
(T3): 50 = 50

It is typical to start with uniform probabilities for transcripts

$$\gamma_i^{(0)} = \frac{1}{N_r} \forall\, i$$

$$\boldsymbol{\gamma}^{(0)} = \left[\frac{1}{3}.\ \frac{1}{3}.\ \frac{1}{3}\right]$$

$$\begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}550\\ 400\\ 50\end{bmatrix} = \boldsymbol{n}$$

E Step (1)

$$E\left(\boldsymbol{x}\middle|\boldsymbol{n},\boldsymbol{\gamma}^{(0)}\right) = \begin{bmatrix} \dfrac{\left(\frac{1}{3}\right)}{\frac{1}{3}+\frac{1}{3}}(\mathbf{550}) + \dfrac{\left(\frac{1}{3}\right)}{\frac{1}{3}+\frac{1}{3}}(\mathbf{400}) \\ \dfrac{\left(\frac{1}{3}\right)}{\frac{1}{3}+\frac{1}{3}}(\mathbf{400}) \\ \dfrac{\left(\frac{1}{3}\right)}{\frac{1}{3}+\frac{1}{3}}(\mathbf{550}) + (\mathbf{1})(\mathbf{50}) \end{bmatrix} = \begin{bmatrix}\mathbf{475}\\ \mathbf{200}\\ \mathbf{325}\end{bmatrix}$$

$$\boldsymbol{\gamma}^{(1)} = \begin{bmatrix}\mathbf{0.475}\\ \mathbf{0.2}\\ \mathbf{0.325}\end{bmatrix}$$

M Step (1)

$$\boldsymbol{\gamma}^{(1)} = \left[\frac{475}{1000}.\ \frac{20}{100}.\ \frac{325}{1000}\right]$$

$$\begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}550\\ 400\\ 50\end{bmatrix} = \boldsymbol{n}$$

UTSouthwestern
Medical Center

Expectation Maximization (Multinomial algorithm example)

Suppose we have true abundances
Transcript 1 : 500 (0.5)
Transcript 2 : 200 (0.2)
Transcript 3 : 300 (0.3)

But that we can only observe whether reads are in the following:
(T1,T3): 300+250 = 550
(T1,T2): 200+200 = 400
(T3): 50 = 50

It is typical to start with uniform probabilities for transcripts

$$\gamma_i^{(0)} = \frac{1}{N_r} \forall\, i$$

$$E\left(\boldsymbol{x} \middle| \boldsymbol{n}, \boldsymbol{\gamma}^{(0)}\right) = \begin{bmatrix} \frac{(0.475)}{0.475 + 0.325}(550) + \frac{(0.475)}{0.475 + 0.2}(400) \\ \frac{0.2}{0.475 + 0.2}(400) \\ \frac{0.325}{0.325 + 0.475}(550) + (1)(50) \end{bmatrix} = \begin{bmatrix} 326.56 + 281.48 \\ 118.518 \\ 223.43 + 50 \end{bmatrix}$$

M Step (2)

$$\boldsymbol{\gamma}^{(2)} = \begin{bmatrix} \mathbf{0.607} \\ \mathbf{0.12} \\ \mathbf{0.273} \end{bmatrix}$$

Converged?

$$|\boldsymbol{\gamma}^{(2)} - \boldsymbol{\gamma}^{(1)}| = \left\| \begin{bmatrix} \mathbf{0.607} \\ \mathbf{0.12} \\ \mathbf{0.273} \end{bmatrix} - \begin{bmatrix} \mathbf{0.475} \\ \mathbf{0.2} \\ \mathbf{0.325} \end{bmatrix} \right\| \leq \boldsymbol{\varepsilon_R}$$

E Step (2)

$$\boldsymbol{\gamma}^{(1)} = \begin{bmatrix} \frac{475}{1000}. & \frac{20}{100}. & \frac{325}{1000} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 550 \\ 400 \\ 50 \end{bmatrix} = \boldsymbol{n}$$

Questions about Expectation Maximization?

Break (10 Minutes)

UTSouthwestern
Medical Center

Part 3: Genetic Transcript Abundance Quantification Software

UTSouthwestern
Medical Center

Software for Genetic Abundance Quantification Salmon & Kallisto

- In order to quantify the true abundances of transcripts within a given genomic RNA-Seq sample, we use the Expectation Maximization algorithm following pseudo or quasi read alignment.
- Two packages which implement this approach are Salmon and Kallisto
  - **Salmon** was developed by Rob Patro, Geet Duggal, Michael Love, Rafael Irizarry, and Carl Kingsford and **published in 2017**. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5600148/)
  - **Kallisto** was developed by Nicolas L. Bray, Harold Pimentel, Páll Melsted, and Lior Pachter and **published in 2016.** (https://www.nature.com/articles/nbt.3519/)

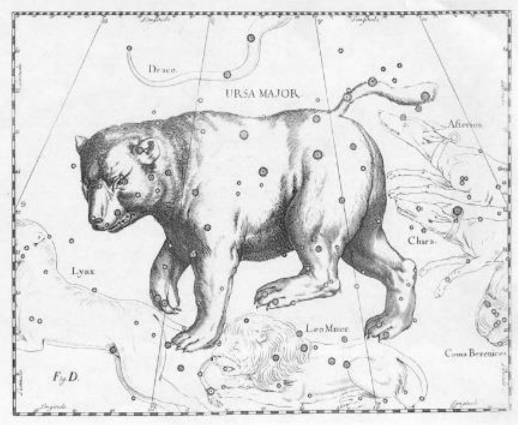


Image from Kallisto website.

- RSEM (RNA-Seq Expectation Maximization) was an earlier package which implemented expectation maximization on the incomplete/missing compatibility patterns:

Li B, Dewey CN. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*. 2011;12:323. Published 2011 Aug 4. doi:10.1186/1471-2105-12-323

Logo from SALMON website.

Genetic Transcript Abundance Software Kallisto & Salmon

- Today, just follow along on the screen with me, tomorrow we will work through getting Kallisto and Salmon on your personal device, and working through some example problems with them together.

- In order to run Kallisto and Salmon, you first need to have a transcriptome in the FASTA format (from which the sequencing reads file of interest is taken)
  - **Note if you do not have a transcriptome file, you might need to produce one by first parsing GTF/SNP/ or another Variant Call Format like file, and the reference sequence to which it corresponds.**
  - **For example, we created an RNA-Seq read simulator which will allow for us to produce a transcriptome file for GTF/SNP/FASTA (reference) files.**

- In this short demonstration We will use an example transcriptome from a subset of genes on human chromosome 22.

Genetic Transcript Abundance Software Kallisto & Salmon

- To access the help for either Salmon or Kallisto, you can use:

```
kallisto
```

```
kallisto 0.46.0

Usage: kallisto <CMD> [arguments] ..

Where <CMD> can be one of:

    index         Builds a kallisto index
    quant         Runs the quantification algorithm
    bus           Generate BUS files for single-cell
data
    pseudo        Runs the pseudoalignment step
    merge         Merges several batch runs
    h5dump        Converts HDF5-formatted results to
plaintext
    inspect       Inspects and gives information about
an index
    version       Prints version information
    cite          Prints citation information

Running kallisto <CMD> without arguments prints usage
information for <CMD>
```

```
Salmon -h
```

```
salmon v1.8.0

Usage:  salmon -h|--help or
        salmon -v|--version or
        salmon -c|--cite or
        salmon [--no-version-check] <COMMAND> [-h
| options]

Commands:
     index      : create a salmon index
     quant      : quantify a sample
     alevin     : single cell analysis
     swim       : perform super-secret operation
     quantmerge : merge multiple quantifications
into a single file
```

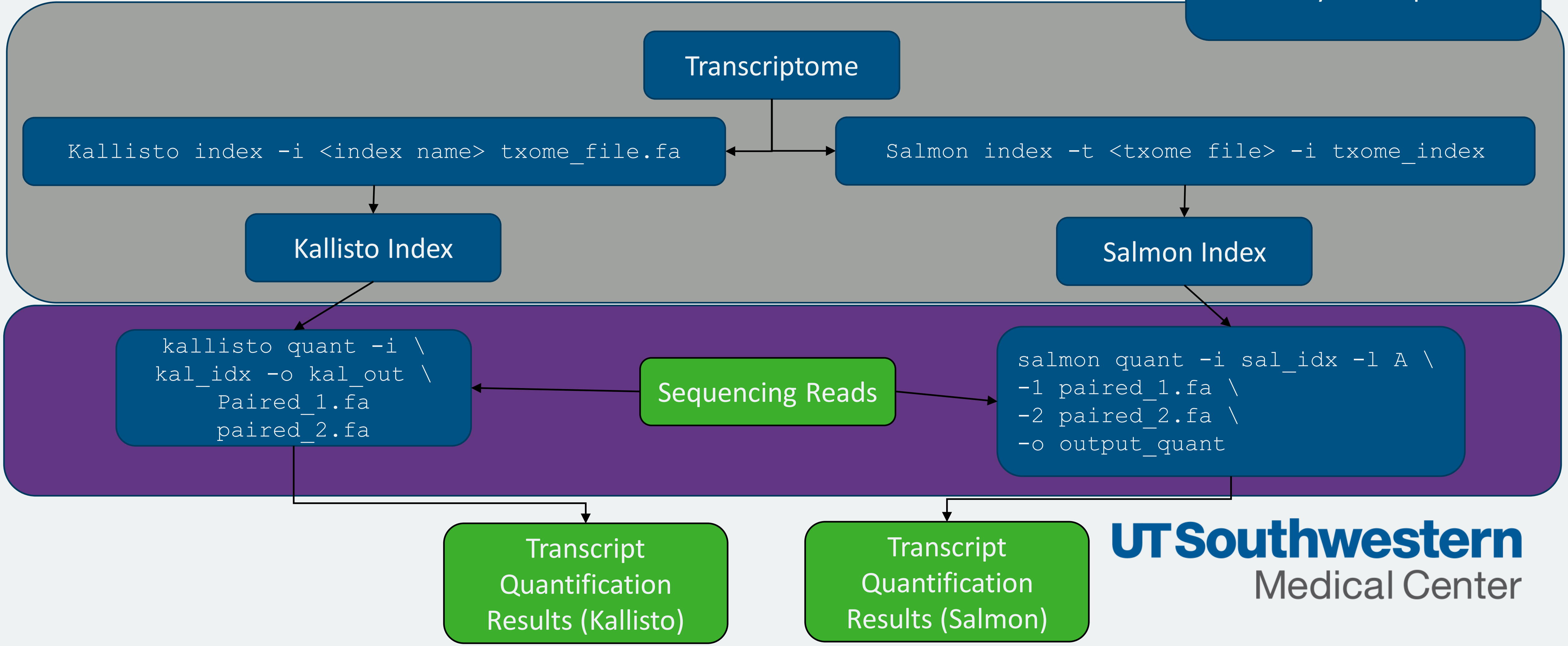
UTSouthwestern
Medical Center

Genetic Transcript Abundance Software Kallisto & Salmon

- As we can see, Salmon and Kallisto have many options which are subdivided into sub-commands, to access the help for a subcommand you can use `Kallisto <CMD>,` or `Salmon <CMD> -h`

- The basic flow of using the tools is as follows:

Length of Kmer can be set directly with options



Transcriptome

```
Kallisto index -i <index name> txome_file.fa
```

```
Salmon index -t <txome file> -i txome_index
```

Kallisto Index

Salmon Index

```
kallisto quant -i \
kal_idx -o kal_out \
Paired_1.fa
paired_2.fa
```

Sequencing Reads

```
salmon quant -i sal_idx -l A \
-1 paired_1.fa \
-2 paired_2.fa \
-o output_quant
```

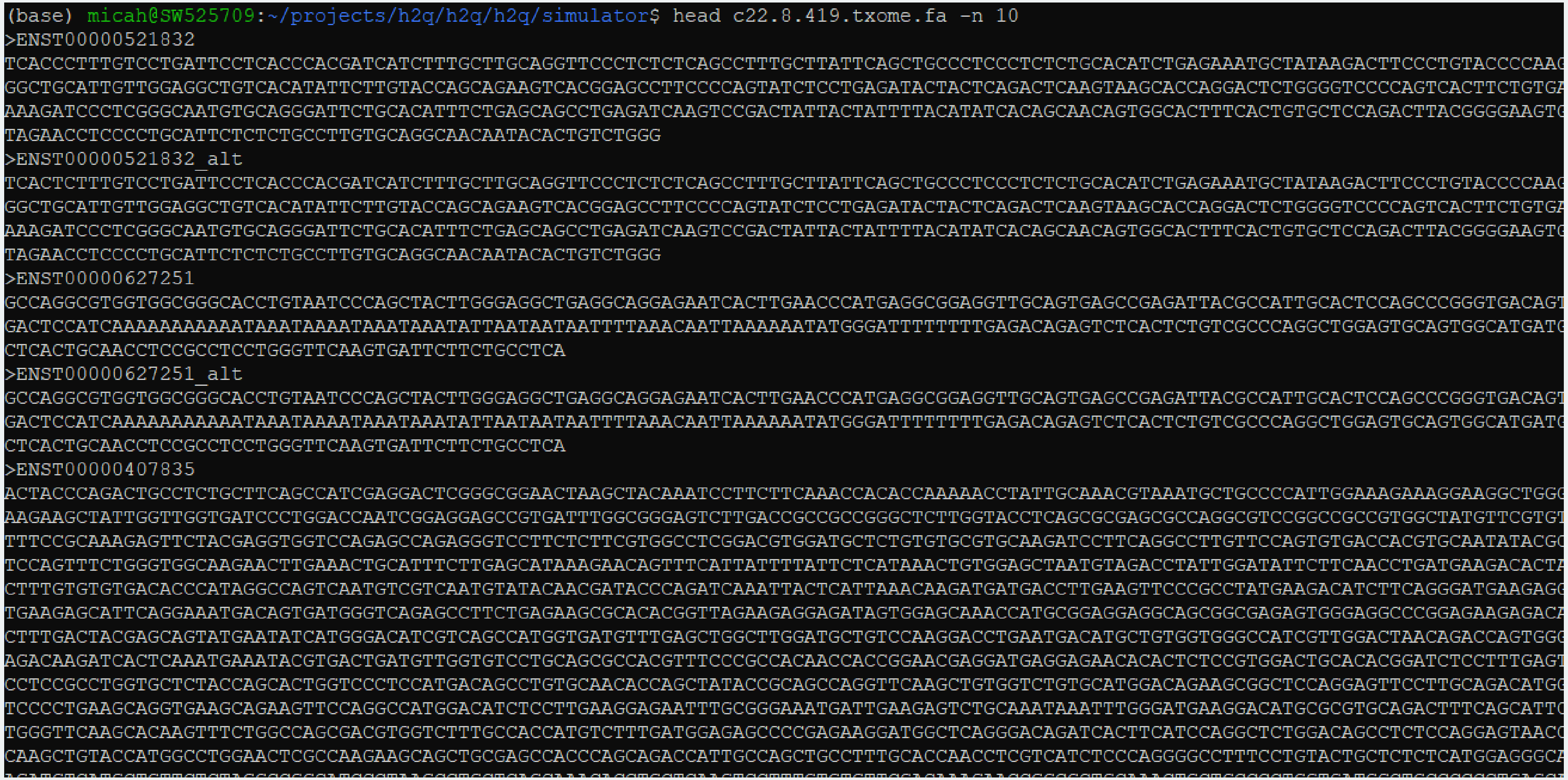
Transcript Quantification Results (Kallisto)

Transcript Quantification Results (Salmon)

UTSouthwestern Medical Center

Genetic Transcript Abundance Software Kallisto & Salmon

- Example Transcriptome file (Note Allele-Specific Names)



```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/simulator$ head c22.8.419.txome.fa -n 10
>ENST00000521832
TCACCCTTTGTCCTGATTCCTCACCCACGATCATCTTTGCTTGCAGGTTCCCTCTCTCAGCCTTTGCTTATTCAGCTGCCCTCCCTCTCTGCACATCTGAGAAATGCTATAAGACTTCCCTGTACCCCAAG
GGCTGCATTGTTGGAGGCTGTCACATATTCTTGTACCAGCAGAAGTCACGGAGCCTTCCCCAGTATCTCCTGAGATACTACTCAGACTCAAGTAAGCACCAGGACTCTGGGGTCCCCAGTCACTTCTGTGA
AAAGATCCCTCGGGCAATGTGCAGGGATTCTGCACATTTCTGAGCAGCCTGAGATCAAGTCCGACTATTACTATTTTACATATCACAGCAACAGTGGCACTTTCACTGTGCTCCAGACTTACGGGGAAGTG
TAGAACCTCCCCTGCATTCTCTCTGCCTTGTGCAGGCAACAATACACTGTCTGGG
>ENST00000521832_alt
TCACTCTTTGTCCTGATTCCTCACCCACGATCATCTTTGCTTGCAGGTTCCCTCTCTCAGCCTTTGCTTATTCAGCTGCCCTCCCTCTCTGCACATCTGAGAAATGCTATAAGACTTCCCTGTACCCCAAG
GGCTGCATTGTTGGAGGCTGTCACATATTCTTGTACCAGCAGAAGTCACGGAGCCTTCCCCAGTATCTCCTGAGATACTACTCAGACTCAAGTAAGCACCAGGACTCTGGGGTCCCCAGTCACTTCTGTGA
AAAGATCCCTCGGGCAATGTGCAGGGATTCTGCACATTTCTGAGCAGCCTGAGATCAAGTCCGACTATTACTATTTTACATATCACAGCAACAGTGGCACTTTCACTGTGCTCCAGACTTACGGGGAAGTG
TAGAACCTCCCCTGCATTCTCTCTGCCTTGTGCAGGCAACAATACACTGTCTGGG
>ENST00000627251
GCCAGGCGTGGTGGCGGGCACCTGTAATCCCAGCTACTTGGGAGGCTGAGGCAGGAGAATCACTTGAACCCATGAGGCGGAGGTTGCAGTGAGCCGAGATTACGCCATTGCACTCCAGCCCGGGTGACAGT
GACTCCATCAAAAAAAAAAATAAATAAAATAAATAAATATTAATAATAATTTTAAACAATTAAAAAATATGGGATTTTTTTTGAGACAGAGTCTCACTCTGTCGCCCAGGCTGGAGTGCAGTGGCATGATG
CTCACTGCAACCTCCGCCTCCTGGGTTCAAGTGATTCTTCTGCCTCA
>ENST00000627251_alt
GCCAGGCGTGGTGGCGGGCACCTGTAATCCCAGCTACTTGGGAGGCTGAGGCAGGAGAATCACTTGAACCCATGAGGCGGAGGTTGCAGTGAGCCGAGATTACGCCATTGCACTCCAGCCCGGGTGACAGT
GACTCCATCAAAAAAAAAAATAAATAAAATAAATAAATATTAATAATAATTTTAAACAATTAAAAAATATGGGATTTTTTTTGAGACAGAGTCTCACTCTGTCGCCCAGGCTGGAGTGCAGTGGCATGATG
CTCACTGCAACCTCCGCCTCCTGGGTTCAAGTGATTCTTCTGCCTCA
>ENST00000407835
ACTACCCAGACTGCCTCTGCTTCAGCCATCGAGGACTCGGGCGGAACTAAGCTACAAATCCTTCTTCAAACCACACCAAAAACCTATTGCAAACGTAAATGCTGCCCCATTGGAAAGAAAGGAAGGCTGGG
AAGAAGCTATTGGTTGGTGATCCCTGGACCAATCGGAGGAGCCGTGATTTGGCGGGAGTCTTGACCGCCGCCGGGCTCTTGGTACCTCAGCGCGAGCGCCAGGCGTCCGGCCGCCGTGGCTATGTTCGTGT
TTTCCGCAAAGAGTTCTACGAGGTGGTCCAGAGCCAGAGGGTCCTTCTCTTCGTGGCCTCGGACGTGGATGCTCTGTGTGCGTGCAAGATCCTTCAGGCCTTGTTCCAGTGTGACCACGTGCAATATACGC
TCCAGTTTCTGGGTGGCAAGAACTTGAAACTGCATTTCTTGAGCATAAAGAACAGTTTCATTATTTTATTCTCATAAACTGTGGAGCTAATGTAGACCTATTGGATATTCTTCAACCTGATGAAGACACTA
CTTTGTGTGTGACACCCATAGGCCAGTCAATGTCGTCAATGTATACAACGATACCCAGATCAAATTACTCATTAAACAAGATGATGACCTTGAAGTTCCCGCCTATGAAGACATCTTCAGGGATGAAGAGG
TGAAGAGCATTCAGGAAATGACAGTGATGGGTCAGAGCCTTCTGAGAAGCGCACACGGTTAGAAGAGGAGATAGTGGAGCAAACCATGCGGAGGAGGCAGCGGCGAGAGTGGGAGGCCCGGAGAAGAGACA
CTTTGACTACGAGCAGTATGAATATCATGGGACATCGTCAGCCATGGTGATGTTTGAGCTGGCTTGGATGCTGTCCAAGGACCTGAATGACATGCTGTGGTGGGCCATCGTTGGACTAACAGACCAGTGGG
AGACAAGATCACTCAAATGAAATACGTGACTGATGTTGGTGTCCTGCAGCGCCACGTTTCCCGCCACAACCACCGGAACGAGGATGAGGAGAACACACTCTCCGTGGACTGCACACGGATCTCCTTTGAGT
CCTCCGCCTGGTGCTCTACCAGCACTGGTCCCTCCATGACAGCCTGTGCAACACCAGCTATACCGCAGCCAGGTTCAAGCTGTGGTCTGTGCATGGACAGAAGCGGCTCCAGGAGTTCCTTGCAGACATGG
TCCCCTGAAGCAGGTGAAGCAGAAGTTCCAGGCCATGGACATCTCCTTGAAGGAGAATTTGCGGGAAATGATTGAAGAGTCTGCAAATAAATTTGGGATGAAGGACATGCGCGTGCAGACTTTCAGCATTC
TGGGTTCAAGCACAAGTTTCTGGCCAGCGACGTGGTCTTTGCCACCATGTCTTTGATGGAGAGCCCCGAGAAGGATGGCTCAGGGACAGATCACTTCATCCAGGCTCTGGACAGCCTCTCCAGGAGTAACC
CAAGCTGTACCATGGCCTGGAACTCGCCAAGAAGCAGCTGCGAGCCACCCAGCAGACCATTGCCAGCTGCCTTTGCACCAACCTCGTCATCTCCCAGGGGCCTTTCCTGTACTGCTCTCTCATGGAGGGCA
```

The transcriptome file can quickly become very large when dealing with many different transcripts of genes (and possible different allele-specific versions of the same transcript).

Produce by internal Python Simulator (just inserts mutations):

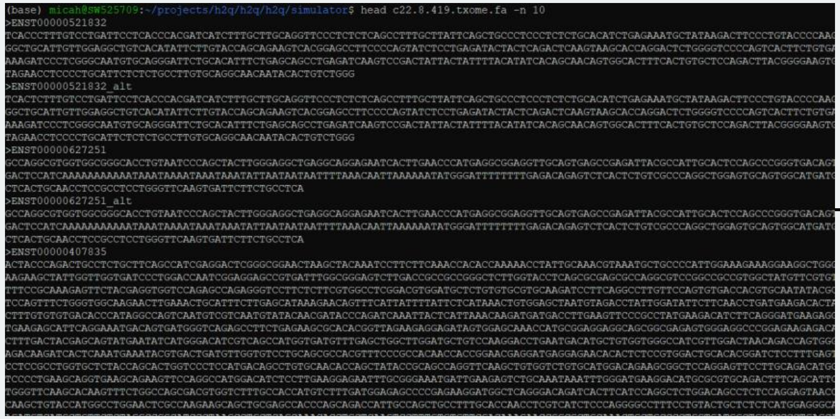
```
Python hisat2_simulate_reads -f ref.fa -g ref.gtf -s ref.snp -o ref.txome.fa
```

UTSouthwestern
Medical Center

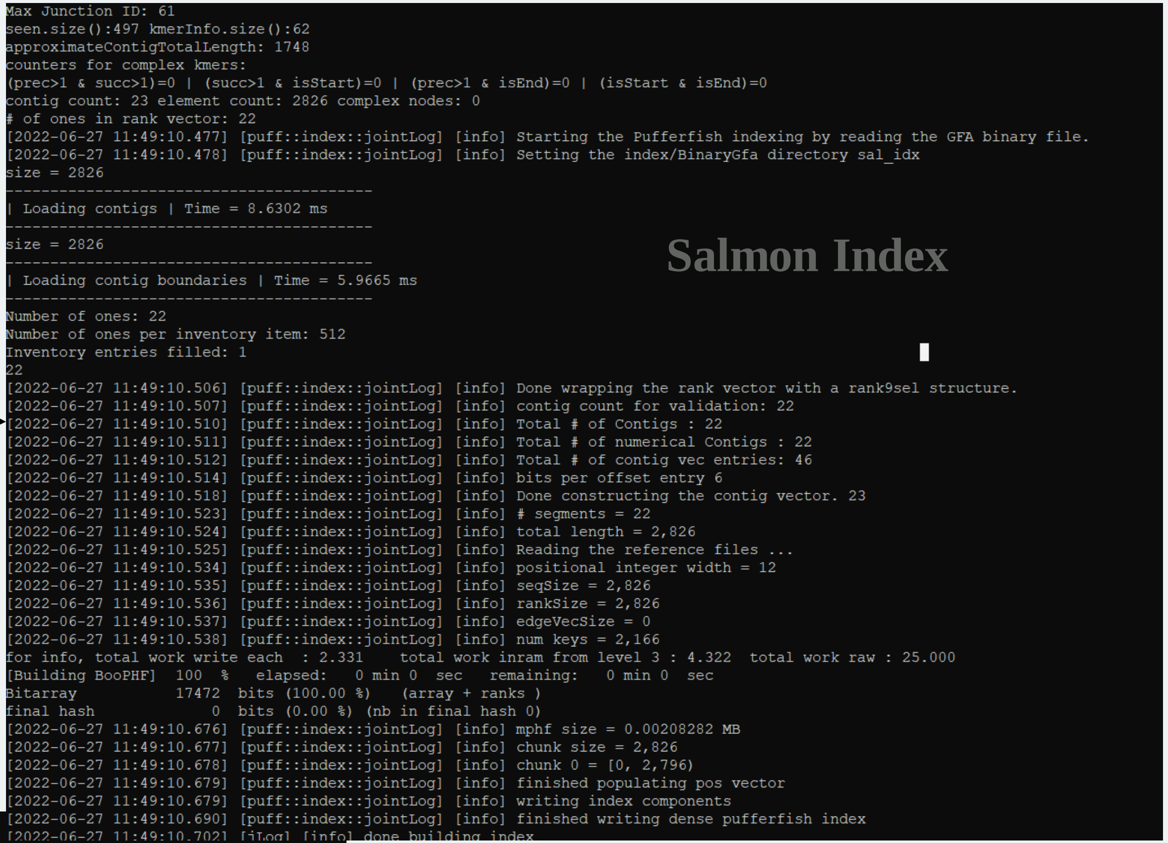
Genetic Transcript Abundance Software Kallisto & Salmon

- From the transcriptome FASTA, we can use Kallisto & Salmon to produce their respective indexes (Colored De Brujin graph for Kallisto & K-mer table + Suffix Array for Salmon)

Produce Index from Transcriptome

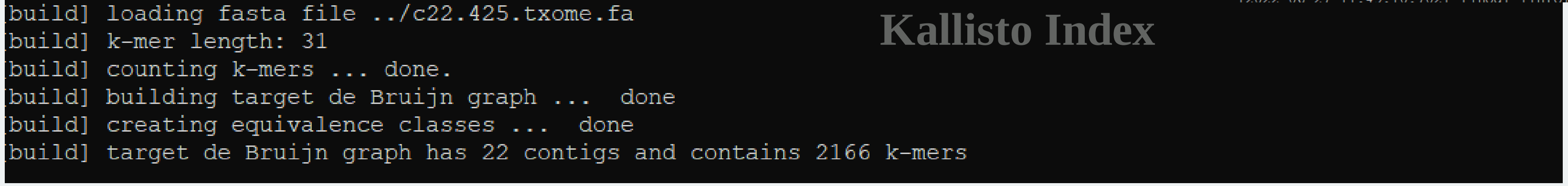


```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/simulator$ head c22.8.419.txome.fa -n 10
>ENST00000521832
TCACCCTTTGTCCTGATTCCTCACCCACGATCATCTTTGCTTGCAGGTTCCCTCTCTCAGCCTTTGCTTATTCAGCTGCCCTCCCTCTCTGCACATCTGAGAAATGCTATAAGACTTCCCTGTACCCCAAG
GGCTGCATTGTTGGAGGCTGTCACATATTCTTGTACCAGCAGAAGTCACGGAGCCTTCCCCAGTATCTCCTGAGATACTACTCAGACTCAAGTAAGCACCAGGACTCTGGGGTCCCCAGTCACTTCTGTGA
AAAGATCCCTCGGGCAATGTGCAGGGATTCTGCACATTTCTGAGCAGCCTGAGATCAAGTCCGACTATTACTATTTTACATATCACAGCAACAGTGGCACTTTCACTGTGCTCCAGACTTACGGGGAAGTG
TAGAACCTCCCCTGCATTCTCTCTGCCTTGTGCAGGCAACAATACACTGTCTGGG
>ENST00000521832_alt
TCACTCTTTGTCCTGATTCCTCACCCACGATCATCTTTGCTTGCAGGTTCCCTCTCTCAGCCTTTGCTTATTCAGCTGCCCTCCCTCTCTGCACATCTGAGAAATGCTATAAGACTTCCCTGTACCCCAAG
GGCTGCATTGTTGGAGGCTGTCACATATTCTTGTACCAGCAGAAGTCACGGAGCCTTCCCCAGTATCTCCTGAGATACTACTCAGACTCAAGTAAGCACCAGGACTCTGGGGTCCCCAGTCACTTCTGTGA
AAAGATCCCTCGGGCAATGTGCAGGGATTCTGCACATTTCTGAGCAGCCTGAGATCAAGTCCGACTATTACTATTTTACATATCACAGCAACAGTGGCACTTTCACTGTGCTCCAGACTTACGGGGAAGTG
TAGAACCTCCCCTGCATTCTCTCTGCCTTGTGCAGGCAACAATACACTGTCTGGG
>ENST00000627251
GCCAGGCGTGGTGGCGGGCACCTGTAATCCCAGCTACTTGGGAGGCTGAGGCAGGAGAATCACTTGAACCCATGAGGCGGAGGTTGCAGTGAGCCGAGATTACGCCATTGCACTCCAGCCCGGGTGACAGT
GACTCCATCAAAAAAAAAAATAAATAAAATAAATAAATATTAATAATAATTTTAAACAATTAAAAAATATGGGATTTTTTTTGAGACAGAGTCTCACTCTGTCGCCCAGGCTGGAGTGCAGTGGCATGATG
CTCACTGCAACCTCCGCCTCCTGGGTTCAAGTGATTCTTCTGCCTCA
>ENST00000627251_alt
GCCAGGCGTGGTGGCGGGCACCTGTAATCCCAGCTACTTGGGAGGCTGAGGCAGGAGAATCACTTGAACCCATGAGGCGGAGGTTGCAGTGAGCCGAGATTACGCCATTGCACTCCAGCCCGGGTGACAGT
GACTCCATCAAAAAAAAAAATAAATAAAATAAATAAATATTAATAATAATTTTAAACAATTAAAAAATATGGGATTTTTTTTGAGACAGAGTCTCACTCTGTCGCCCAGGCTGGAGTGCAGTGGCATGATG
CTCACTGCAACCTCCGCCTCCTGGGTTCAAGTGATTCTTCTGCCTCA
>ENST00000407835
ACTACCCAGACTGCCTCTGCTTCAGCCATCGAGGACTCGGGCGGAACTAAGCTACAAATCCTTCTTCAAACCACACCAAAAACCTATTGCAAACGTAAATGCTGCCCCATTGGAAAGAAAGGAAGGCTGGG
AAGAAGCTATTGGTTGGTGATCCCTGGACCAATCGGAGGAGCCGTGATTTGGCGGGAGTCTTGACCGCCGCCGGGCTCTTGGTACCTCAGCGCGAGCGCCAGGCGTCCGGCCGCCGTGGCTATGTTCGTGT
TTTCCGCAAAGAGTTCTACGAGGTGGTCCAGAGCCAGAGGGTCCTTCTCTTCGTGGCCTCGGACGTGGATGCTCTGTGTGCGTGCAAGATCCTTCAGGCCTTGTTCCAGTGTGACCACGTGCAATATACGC
TCCAGTTTCTGGGTGGCAAGAACTTGAAACTGCATTTCTTGAGCATAAAGAACAGTTTCATTATTTTATTCTCATAAACTGTGGAGCTAATGTAGACCTATTGGATATTCTTCAACCTGATGAAGACACTA
CTTTGTGTGTGACACCCATAGGCCAGTCAATGTCGTCAATGTATACAACGATACCCAGATCAAATTACTCATTAAACAAGATGATGACCTTGAAGTTCCCGCCTATGAAGACATCTTCAGGGATGAAGAGC
TGAAGAGCATTCAGGAAATGACAGTGATGGGTCAGAGCCTTCTGAGAAGCGCACACGGTTAGAAGAGGAGATAGTGGAGCAAACCATGCGGAGGAGGCAGCGGCGAGAGTGGGAGGCCCGGAGAAGAGACA
CTTTGACTACGAGCAGTATGAATATCATGGGACATCGTCAGCCATGGTGATGTTTGAGCTGGCTTGGATGCTGTCCAAGGACCTGAATGACATGCTGTGGTGGGCCATCGTTGGACTAACAGACCAGTGGG
AGACAAGATCACTCAAATGAAATACGTGACTGATGTTGGTGTCCTGCAGCGCCACGTTTCCCGCCACAACCACCGGAACGAGGATGAGGAGAACACACTCTCCGTGGACTGCACACGGATCTCCTTTGAGT
CCTCCGCCTGGTGCTCTACCAGCACTGGTCCCTCCATGACAGCCTGTGCAACACCAGCTATACCGCAGCCAGGTTCAAGCTGTGGTCTGTGCATGGACAGAAGCGGCTCCAGGAGTTCCTTGCAGACATGC
TCCCCTGAAGCAGGTGAAGCAGAAGTTCCAGGCCATGGACATCTCCTTGAAGGAGAATTTGCGGGAAATGATTGAAGAGTCTGCAAATAAATTTGGGATGAAGGACATGCGCGTGCAGACTTTCAGCATTC
TGGGTTCAAGCACAAGTTTCTGGCCAGCGACGTGGTCTTTGCCACCATGTCTTTGATGGAGAGCCCCGAGAAGGATGGCTCAGGGACAGATCACTTCATCCAGGCTCTGGACAGCCTCTCCAGGAGTAACC
CAAGCTGTACCATGGCCTGGAACTCGCCAAGAAGCAGCTGCGAGCCACCCAGCAGACCATTGCCAGCTGCCTTTGCACCAACCTCGTCATCTCCCAGGGGCCTTTCCTGTACTGCTCTCTCATGGAGGGCA
```



Salmon Index

```
Max Junction ID: 61
seen.size():497 kmerInfo.size():62
approximateContigTotalLength: 1748
counters for complex kmers:
(prec>1 & succ>1)=0 | (succ>1 & isStart)=0 | (prec>1 & isEnd)=0 | (isStart & isEnd)=0
contig count: 23 element count: 2826 complex nodes: 0
# of ones in rank vector: 22
[2022-06-27 11:49:10.477] [puff::index::jointLog] [info] Starting the Pufferfish indexing by reading the GFA binary file.
[2022-06-27 11:49:10.478] [puff::index::jointLog] [info] Setting the index/BinaryGfa directory sal_idx
size = 2826
-----------------------------------------
| Loading contigs | Time = 8.6302 ms
-----------------------------------------
size = 2826
-----------------------------------------
| Loading contig boundaries | Time = 5.9665 ms
-----------------------------------------
Number of ones: 22
Number of ones per inventory item: 512
Inventory entries filled: 1
22
[2022-06-27 11:49:10.506] [puff::index::jointLog] [info] Done wrapping the rank vector with a rank9sel structure.
[2022-06-27 11:49:10.507] [puff::index::jointLog] [info] contig count for validation: 22
[2022-06-27 11:49:10.510] [puff::index::jointLog] [info] Total # of Contigs : 22
[2022-06-27 11:49:10.511] [puff::index::jointLog] [info] Total # of numerical Contigs : 22
[2022-06-27 11:49:10.512] [puff::index::jointLog] [info] Total # of contig vec entries: 46
[2022-06-27 11:49:10.514] [puff::index::jointLog] [info] bits per offset entry 6
[2022-06-27 11:49:10.518] [puff::index::jointLog] [info] Done constructing the contig vector. 23
[2022-06-27 11:49:10.523] [puff::index::jointLog] [info] # segments = 22
[2022-06-27 11:49:10.524] [puff::index::jointLog] [info] total length = 2,826
[2022-06-27 11:49:10.525] [puff::index::jointLog] [info] Reading the reference files ...
[2022-06-27 11:49:10.534] [puff::index::jointLog] [info] positional integer width = 12
[2022-06-27 11:49:10.535] [puff::index::jointLog] [info] seqSize = 2,826
[2022-06-27 11:49:10.536] [puff::index::jointLog] [info] rankSize = 2,826
[2022-06-27 11:49:10.537] [puff::index::jointLog] [info] edgeVecSize = 0
[2022-06-27 11:49:10.538] [puff::index::jointLog] [info] num keys = 2,166
for info, total work write each  : 2.331    total work inram from level 3 : 4.322  total work raw : 25.000
[Building BooPHF]  100  %   elapsed:   0 min 0  sec   remaining:   0 min 0  sec
Bitarray           17472  bits (100.00 %)   (array + ranks )
final hash             0  bits (0.00 %) (nb in final hash 0)
[2022-06-27 11:49:10.676] [puff::index::jointLog] [info] mphf size = 0.00208282 MB
[2022-06-27 11:49:10.677] [puff::index::jointLog] [info] chunk size = 2,826
[2022-06-27 11:49:10.678] [puff::index::jointLog] [info] chunk 0 = [0, 2,796)
[2022-06-27 11:49:10.679] [puff::index::jointLog] [info] finished populating pos vector
[2022-06-27 11:49:10.679] [puff::index::jointLog] [info] writing index components
[2022-06-27 11:49:10.690] [puff::index::jointLog] [info] finished writing dense pufferfish index
[2022-06-27 11:49:10.702] [jLog] [info] done building index
```

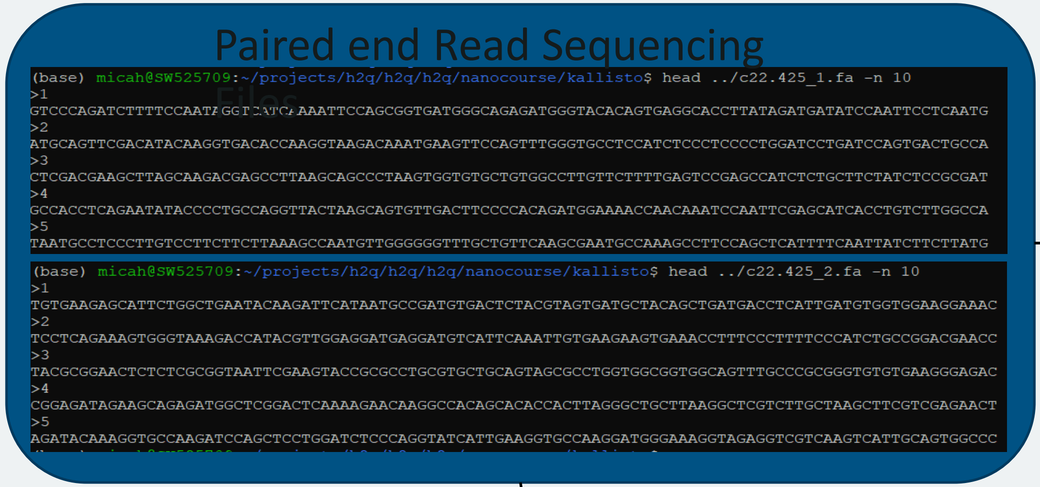


Kallisto Index

```
[build] loading fasta file ../c22.425.txome.fa
[build] k-mer length: 31
[build] counting k-mers ... done.
[build] building target de Bruijn graph ...  done
[build] creating equivalence classes ...  done
[build] target de Bruijn graph has 22 contigs and contains 2166 k-mers
```

Genetic Transcript Abundance Software Kallisto & Salmon

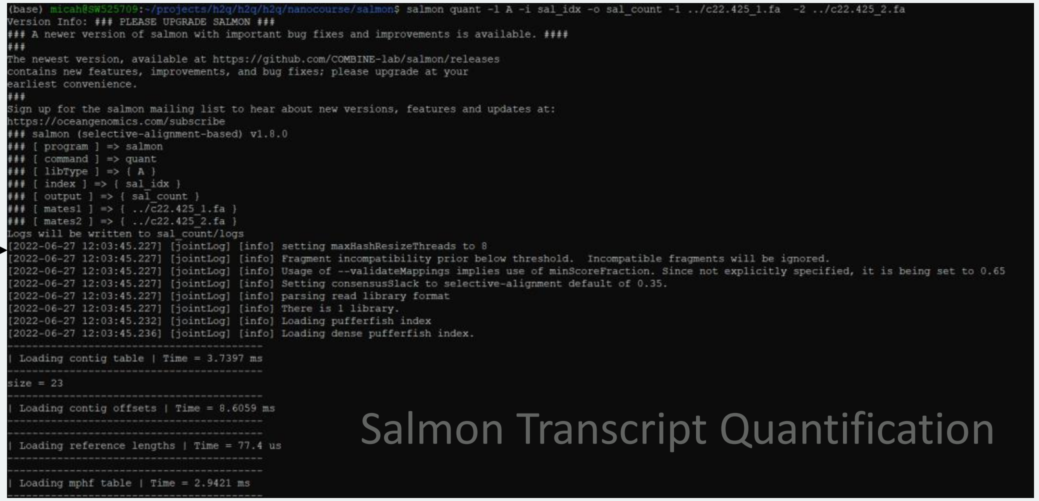
- Run the pseudo-alignment and quantification procedures using Kallisto and Salmon to produce the results (quantification of abundances).



Paired end Read Sequencing Files

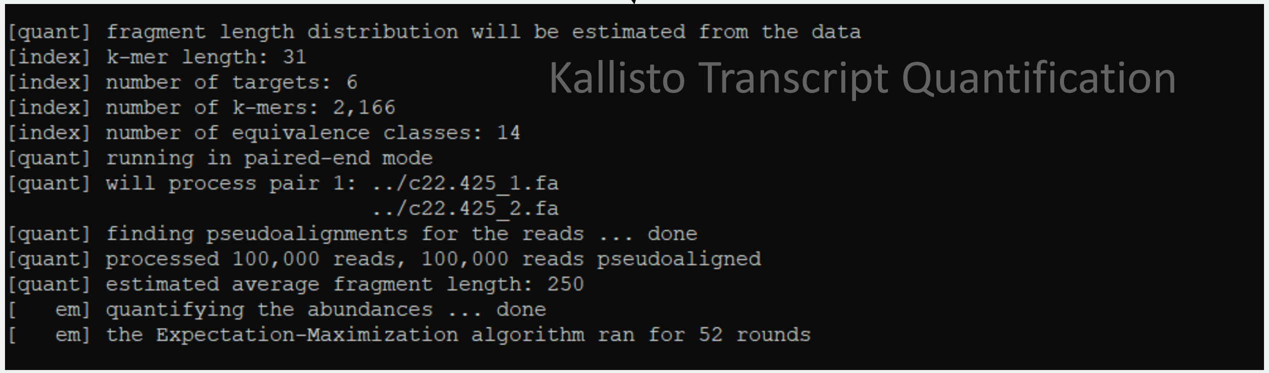
```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ head ../c22.425_1.fa -n 10
>1
GTCCCAGATCTTTTCCAATAGGTCATCAAAATTCCAGCGGTGATGGGCAGAGATGGGTACACAGTGAGGCACCTTATAGATGATATCCAATTCCTCAATG
>2
ATGCAGTTCGACATACAAGGTGACACCAAGGTAAGACAAATGAAGTTCCAGTTTGGGTGCCTCCATCTCCCTCCCCTGGATCCTGATCCAGTGACTGCCA
>3
CTCGACGAAGCTTAGCAAGACGAGCCTTAAGCAGCCCTAAGTGGTGTGCTGTGGCCTTGTTCTTTTGAGTCCGAGCCATCTCTGCTTCTATCTCCGCGAT
>4
GCCACCTCAGAATATACCCCTGCCAGGTTACTAAGCAGTGTTGACTTCCCCACAGATGGAAAACCAACAAATCCAATTCGAGCATCACCTGTCTTGGCCA
>5
TAATGCCTCCCTTGTCCTTCTTCTTAAAGCCAATGTTGGGGGGTTTGCTGTTCAAGCGAATGCCAAAGCCTTCCAGCTCATTTTCAATTATCTTCTTATG
```

```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ head ../c22.425_2.fa -n 10
>1
TGTGAAGAGCATTCTGGCTGAATACAAGATTCATAATGCCGATGTGACTCTACGTAGTGATGCTACAGCTGATGACCTCATTGATGTGGTGGAAGGAAAC
>2
TCCTCAGAAAGTGGGTAAAGACCATACGTTGGAGGATGAGGATGTCATTCAAATTGTGAAGAAGTGAAACCTTTCCCTTTTCCCATCTGCCGGACGAACC
>3
TACGCGGAACTCTCTCGCGGTAATTCGAAGTACCGCGCCTGCGTGCTGCAGTAGCGCCTGGTGGCGGTGGCAGTTTGCCCGCGGGTGTGTGAAGGGAGAC
>4
CGGAGATAGAAGCAGAGATGGCTCGGACTCAAAAGAACAAGGCCACAGCACACCACTTAGGGCTGCTTAAGGCTCGTCTTGCTAAGCTTCGTCGAGAACT
>5
AGATACAAAGGTGCCAAGATCCAGCTCCTGGATCTCCCAGGTATCATTGAAGGTGCCAAGGATGGGAAAGGTAGAGGTCGTCAAGTCATTGCAGTGGCCC
```



```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon$ salmon quant -l A -i sal_idx -o sal_count -1 ../c22.425_1.fa  -2 ../c22.425_2.fa
Version Info: ### PLEASE UPGRADE SALMON ###
### A newer version of salmon with important bug fixes and improvements is available. ####
###
The newest version, available at https://github.com/COMBINE-lab/salmon/releases
contains new features, improvements, and bug fixes; please upgrade at your
earliest convenience.
###
Sign up for the salmon mailing list to hear about new versions, features and updates at:
https://oceangenomics.com/subscribe
### salmon (selective-alignment-based) v1.8.0
### [ program ] => salmon
### [ command ] => quant
### [ libType ] => { A }
### [ index ] => { sal_idx }
### [ output ] => { sal_count }
### [ mates1 ] => { ../c22.425_1.fa }
### [ mates2 ] => { ../c22.425_2.fa }
Logs will be written to sal_count/logs
[2022-06-27 12:03:45.227] [jointLog] [info] setting maxHashResizeThreads to 8
[2022-06-27 12:03:45.227] [jointLog] [info] Fragment incompatibility prior below threshold.  Incompatible fragments will be ignored.
[2022-06-27 12:03:45.227] [jointLog] [info] Usage of --validateMappings implies use of minScoreFraction. Since not explicitly specified, it is being set to 0.65
[2022-06-27 12:03:45.227] [jointLog] [info] Setting consensusSlack to selective-alignment default of 0.35.
[2022-06-27 12:03:45.227] [jointLog] [info] parsing read library format
[2022-06-27 12:03:45.227] [jointLog] [info] There is 1 library.
[2022-06-27 12:03:45.232] [jointLog] [info] Loading pufferfish index
[2022-06-27 12:03:45.236] [jointLog] [info] Loading dense pufferfish index.
-----------------------------------------
| Loading contig table | Time = 3.7397 ms
-----------------------------------------
size = 23
-----------------------------------------
| Loading contig offsets | Time = 8.6059 ms
-----------------------------------------
-----------------------------------------
| Loading reference lengths | Time = 77.4 us
-----------------------------------------
-----------------------------------------
| Loading mphf table | Time = 2.9421 ms
-----------------------------------------
```

Salmon Transcript Quantification

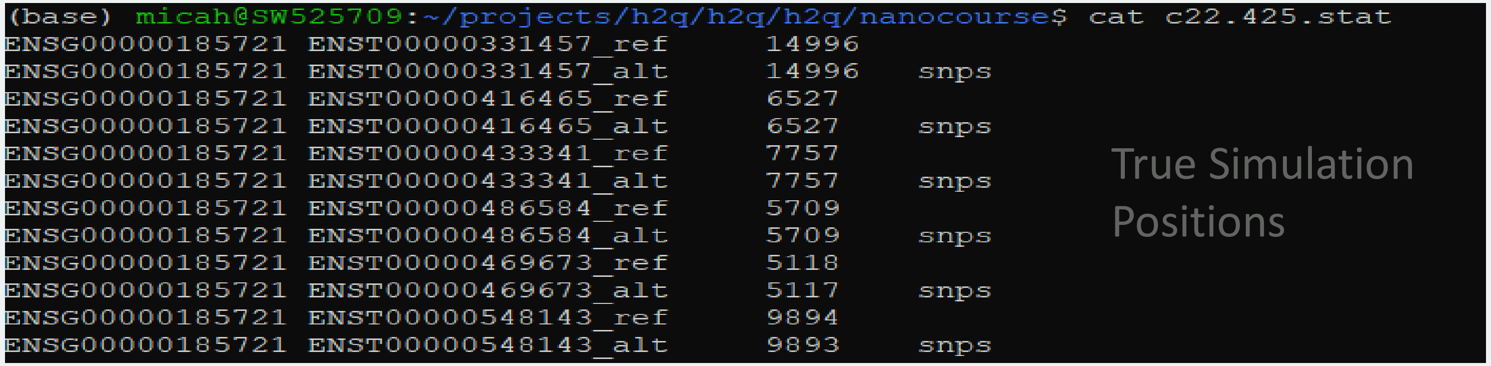


```
[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 31
[index] number of targets: 6
[index] number of k-mers: 2,166
[index] number of equivalence classes: 14
[quant] running in paired-end mode
[quant] will process pair 1: ../c22.425_1.fa
                             ../c22.425_2.fa
[quant] finding pseudoalignments for the reads ... done
[quant] processed 100,000 reads, 100,000 reads pseudoaligned
[quant] estimated average fragment length: 250
[   em] quantifying the abundances ... done
[   em] the Expectation-Maximization algorithm ran for 52 rounds
```

Kallisto Transcript Quantification

UTSouthwestern
Medical Center

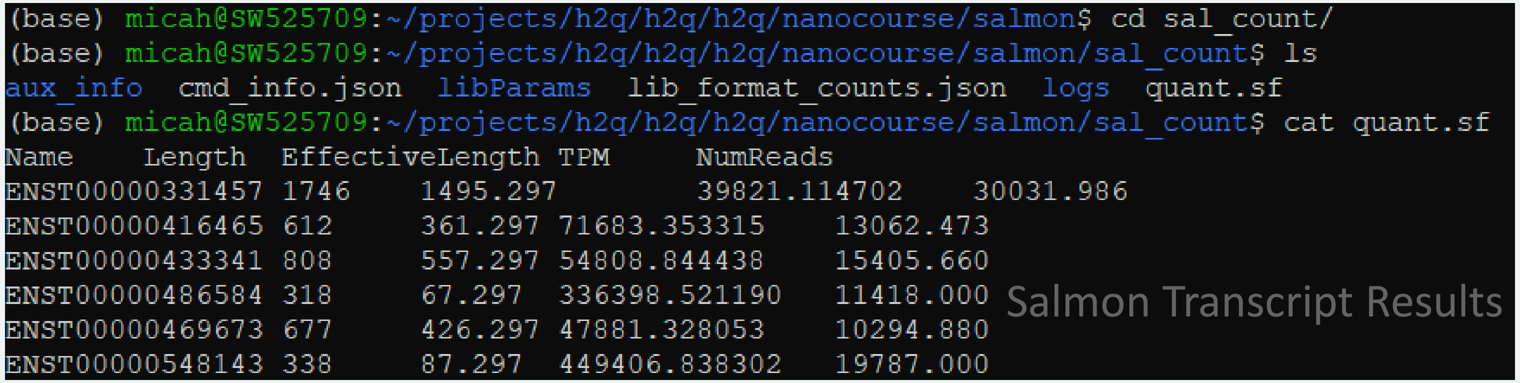
Genetic Transcript Abundance Results



```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse$ cat c22.425.stat
ENSG00000185721 ENST00000331457_ref      14996
ENSG00000185721 ENST00000331457_alt      14996    snps
ENSG00000185721 ENST00000416465_ref      6527
ENSG00000185721 ENST00000416465_alt      6527     snps
ENSG00000185721 ENST00000433341_ref      7757
ENSG00000185721 ENST00000433341_alt      7757     snps
ENSG00000185721 ENST00000486584_ref      5709
ENSG00000185721 ENST00000486584_alt      5709     snps
ENSG00000185721 ENST00000469673_ref      5118
ENSG00000185721 ENST00000469673_alt      5117     snps
ENSG00000185721 ENST00000548143_ref      9894
ENSG00000185721 ENST00000548143_alt      9893     snps
```

True Simulation Positions

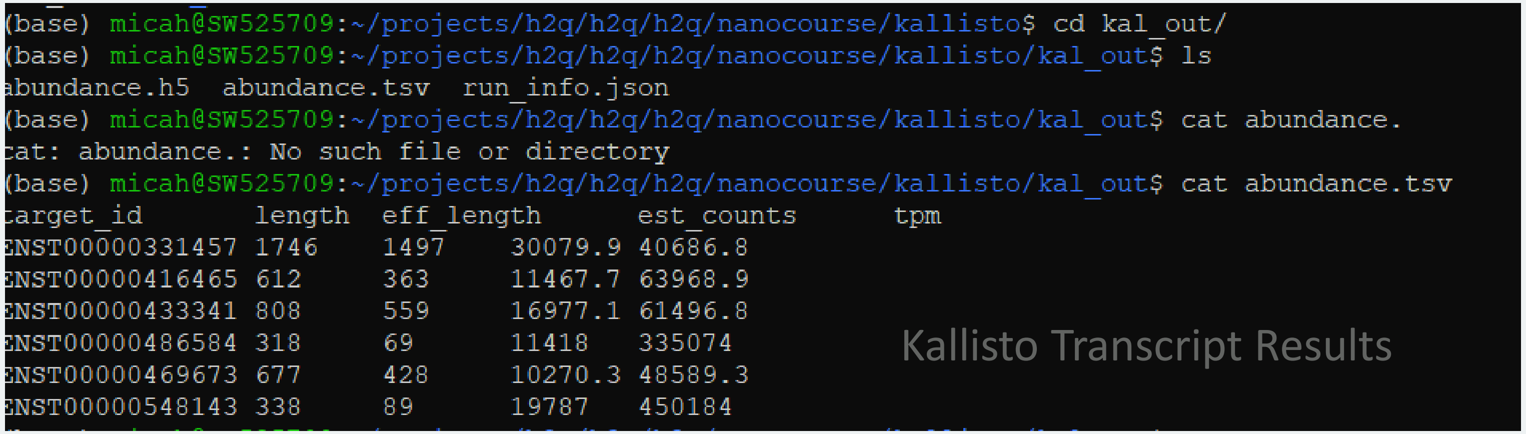
Reads were simulated from allele-specific (randomly mutated) versions of 6 transcripts of gene ENSG00000185721



```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon$ cd sal_count/
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon/sal_count$ ls
aux_info  cmd_info.json  libParams  lib_format_counts.json  logs  quant.sf
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon/sal_count$ cat quant.sf
Name    Length  EffectiveLength TPM     NumReads
ENST00000331457 1746    1495.297        39821.114702    30031.986
ENST00000416465 612     361.297 71683.353315    13062.473
ENST00000433341 808     557.297 54808.844438    15405.660
ENST00000486584 318     67.297  336398.521190   11418.000
ENST00000469673 677     426.297 47881.328053    10294.880
ENST00000548143 338     87.297  449406.838302   19787.000
```

Salmon Transcript Results

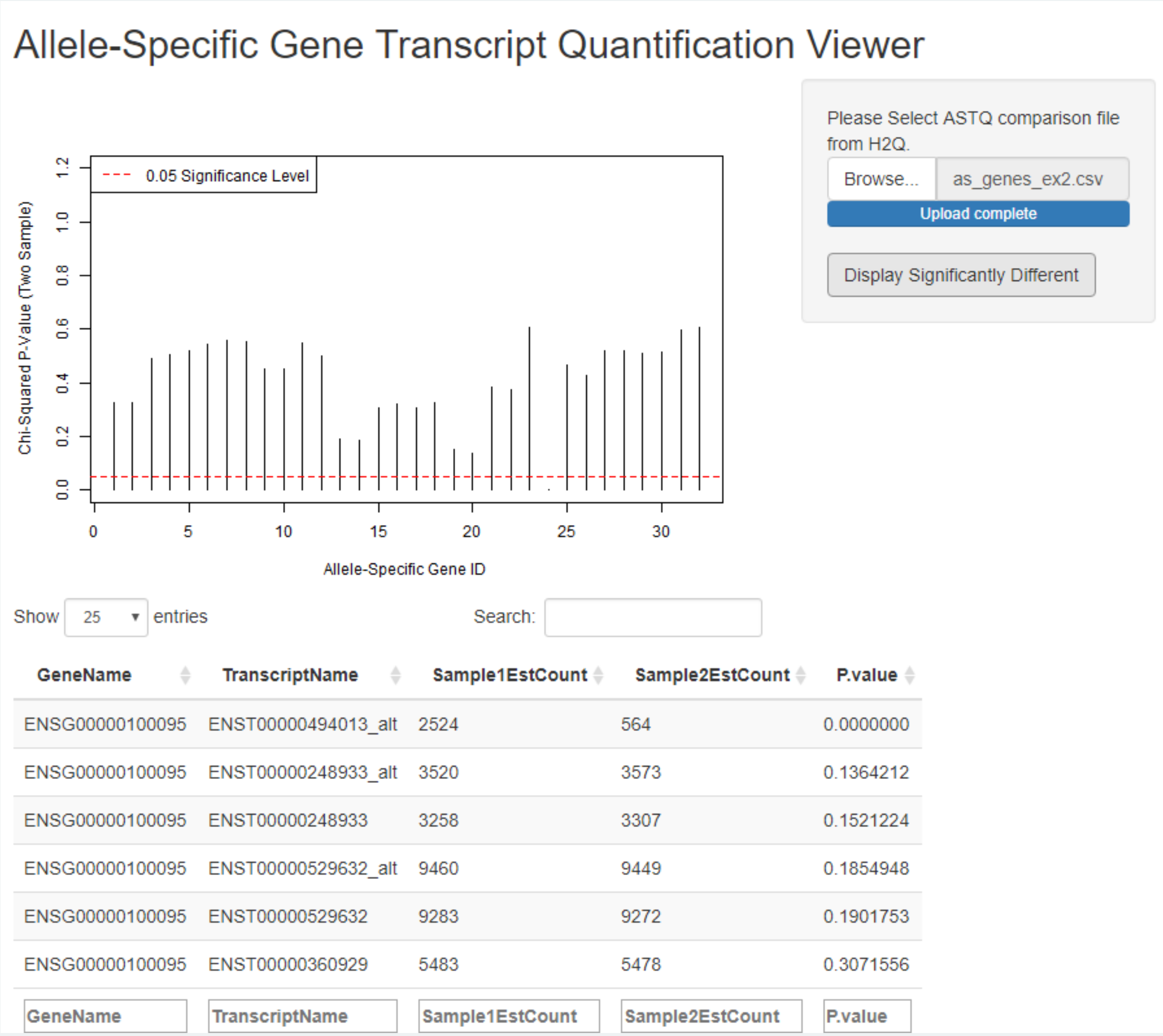
Since a non-allele specific transcriptome was used, Salmon and Kallisto cannot provide more specific quantification of these results than at the transcript level



```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ cd kal_out/
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto/kal_out$ ls
abundance.h5  abundance.tsv  run_info.json
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto/kal_out$ cat abundance.
cat: abundance.: No such file or directory
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto/kal_out$ cat abundance.tsv
target_id       length  eff_length      est_counts      tpm
ENST00000331457 1746    1497    30079.9 40686.8
ENST00000416465 612     363     11467.7 63968.9
ENST00000433341 808     559     16977.1 61496.8
ENST00000486584 318     69      11418   335074
ENST00000469673 677     428     10270.3 48589.3
ENST00000548143 338     89      19787   450184
```

Kallisto Transcript Results

Genetic Transcript (Allele-Specific Abundance Results with H2Q ~ Teaser)



Allele-Specific Gene Transcript Quantification Viewer

Please Select ASTQ comparison file from H2Q.

as_genes_ex2.csv

Upload complete

Show 25 entries

Search:

GeneName	TranscriptName	Sample1EstCount	Sample2EstCount	P.value
ENSG00000100095	ENST00000494013_alt	2524	564	0.0000000
ENSG00000100095	ENST00000248933_alt	3520	3573	0.1364212
ENSG00000100095	ENST00000248933	3258	3307	0.1521224
ENSG00000100095	ENST00000529632_alt	9460	9449	0.1854948
ENSG00000100095	ENST00000529632	9283	9272	0.1901753
ENSG00000100095	ENST00000360929	5483	5478	0.3071556
GeneName	TranscriptName	Sample1EstCount	Sample2EstCount	P.value

- By using the graph-alignment procedures of HISAT2, we are able to produce exact alignments about as quickly as Kallisto and Salmon produce Pseudoalignments.

- This also allows us to identify allelic markers more easily (without rewriting the allelic variants into a transcriptome ahead of time).

Questions about Kallisto, Salmon and H2Q?

Day 1 Session 2 Summary

1. Pseudo-alignment/Quasi-mapping
2. Expectation Maximization in Coarsened Multinomial Data
3. Salmon, Kallisto, and H2Q (Introduction)

Day 2 Session 4 Topics

1. Additional information about Salmon and Kallisto
2. Statistical Procedures for Comparing Quantification Results
3. Practice Problem and Environment Set-up

Back-up (Bonus Material) – Day 1

UTSouthwestern
Medical Center

Mathematical Background & Theory
Proof of Jensen’s Inequality
&
Derivation of Expectation Maximization (Chalk-Talk)

UTSouthwestern
Medical Center

Expectation Maximization for Coarsened Multinomial Data Software Example

UTSouthwestern
Medical Center