Computational Biomedicine

Lecture 1: Course Overview

I. Background
  A. Basics of Genomics
    1. Length of genomic segment: G
    2. Number of reads: N
    3. Length of each read: L
    4. Coverage: $\frac{NL}{G}$
  B. Fundamentals of Genetics
    1. Gene: The basic physical and functional unit of heredity
    2. Allele: One version of a Gene
    3. Trait: In genetics, a trait refers to any genetically determined characteristic
    4. Phenotype: The outwards expression of the genotype
    5. Genotype: The collection of genes in one individual
    6. Homozygous: A genetic condition where an individual inherits the same alleles for a particular gene from both parents.
    7. Heterozygous: Having inherited different forms of a particular gene from each parent.

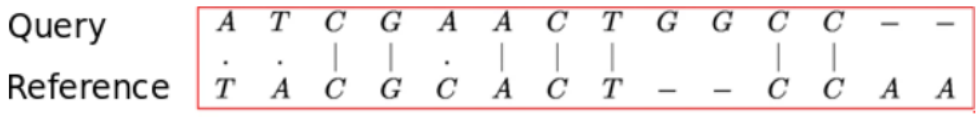
Lecture 2: Efficient Search and Alignment

I. Overview on high throughput sequencing data
  A. Genome Sequencing: the automatic determination of the complete DNA sequence of one or many given genomes
  B. Typical High-throughput Sequencing Experiment
    1. Illumina Sequencing
      a) Throughput: millions of reads that is 36 - 300 base pair long
      b) Reads may have position-wise varying quality
      c) Quality correspond to error probability that is encoded by
      $$q = -10 * log_{10}(P)$$
      Eg: $p = 10^{-3}$ per base corresponds to $q = 30$
      d) Typical read error probability for Illumina reads $\approx 1\,\%$
    2. FASTQ file format
      a) Typical output format from the sequencing platform
      b) 4 lines of text
        (1) Header: always begins with a “@” and is followed by a sequence identifier

(2) Sequence: The raw sequence letters
(3) Usually is just a "+", sometimes is followed by the same sequence identifier
(4) Phred quality score
(a) 0 to 40 quality score is encoded using ASCII 33 to 73
(b) Must be of the same number as the Sequence

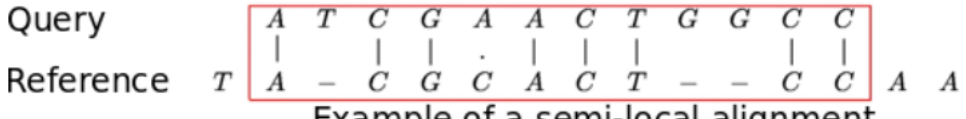
C. Computational Problems on Sequencing Data
1. Genome Assembly: The problem of reconstruct contigs or full chromosomes from short/long sequencing data
2. Read Mapping/Alignments: map/align reads back to a known genome
3. Variant Calling: Detection of positions varying doom a reference population
4. Challenges of these computational problems
a) Hundreds of millions of reads of short length
b) Computational challenge that needs **efficient** algorithms
c) Cost of analysis soon surpasses cost of sequencing as sequencing gets cheaper and cheaper

D. Read Analysis — Mapping
1. Read mapping problem
a) Problem: For each read find its target regions on the reference genome such that there are at most k mismatches/indels between the read and the target
(1) Global alignment: align the two query sequence so that they start and end at the same time



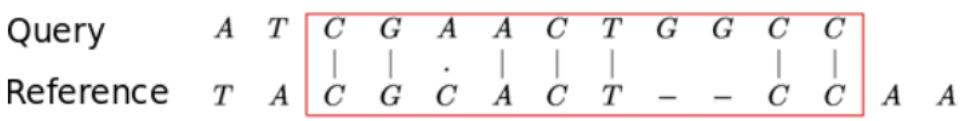
Example of a global alignment

(2) Semi-local alignment: align the whole query sequence to the reference genome



Example of a semi-local alignment

(3) Local alignment: align the part of the query sequence with high quality



Example of a local alignment

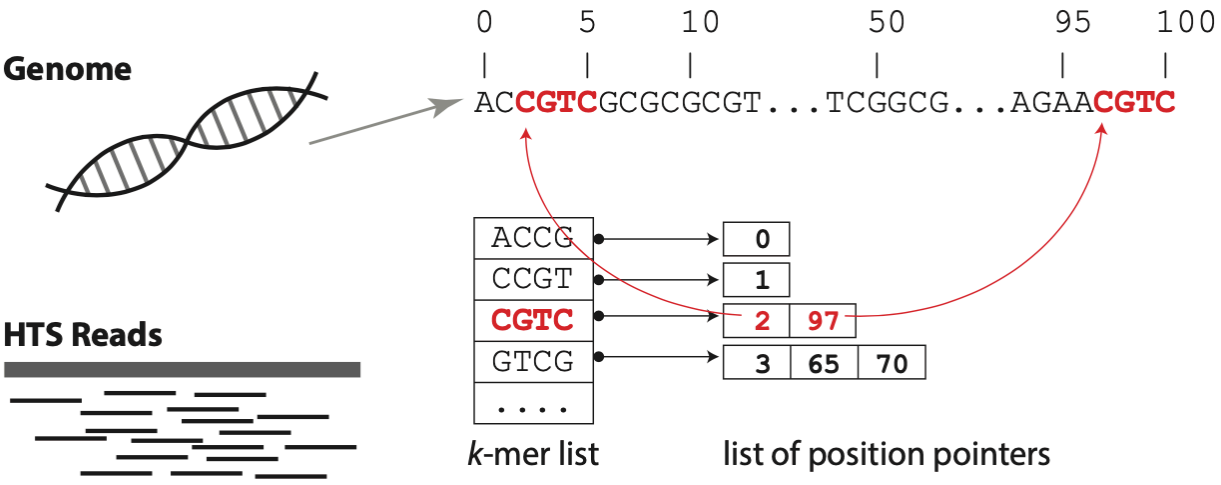
II. Strategies for efficient search in large genomes
   A. Global/local alignment of whole reads is **computationally prohibitive**
   B. Seed-and-extend alignment: build alignment from seed regions
      1. First search the seed using some indexing strategy
         a) Spaced seeds
         b) Suffix trees/arrays
         c) Burrows-Wheeler
      2. Extend the alignment from left or right, stop when the score is below some threshold
      3. Two main problem of seed-and-extend alignment
         a) Identify seed regions for each read
         b) Extend local candidate seeds into full alignment and choose optimal ones

III. String indexing
   A. Common Index Data Structures

	k-mer index	Suffix Tree	Suffix Array	BWT & FM Index
Space	$4L + \|\|\Sigma\|\|^k$	$\approx 12L$	$\approx 4L$	$\ll L$
Search Time	$O(p)$	$O(p \log L)$	$O(p \log L)$	$O(p)$

L target (genome) length
p pattern (read) length
$|\Sigma|$ alphabet size (4 for DNA)
k fixed constant $\ll L$

   B. (Spaced) k-mer index
      1. Steps to generate a k-mer index
         a) Take the genome as a linear string of alphabet A,C,G,T
         b) Generate a list of all words of length k in the genome string (k-mers) and make it unique
         c) Store for each k-mer all genomic positions of occurrence



2. Spaced Seeding*: Use the same number of positions k, but
   a) Introduce gaps in the word

```
0     5     10          50          95    100
|     |     |           |           |     |
ACCGTCGCGCGCGT...TCGGCG...AGAACGTC
AC--TC--GC
ACC-----GC
A--GTCG--C
```

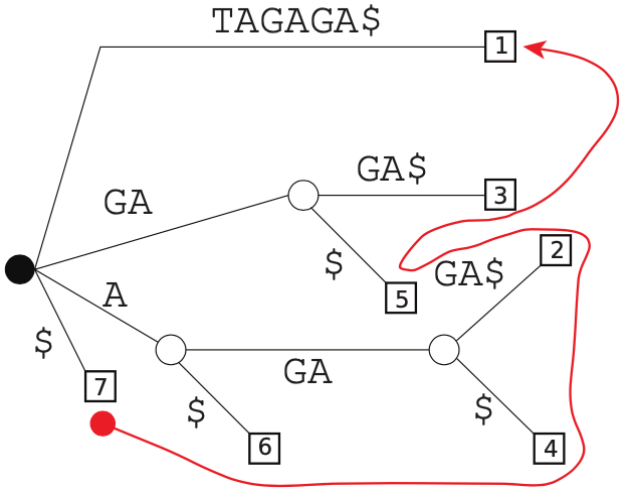
   b) May use different gap patterns to increase sensitivity

```
0     5     10          50          95    100
|     |     |           |           |     |
ACCGTCGCGCGCGT...TCGGCG...AGAACGTC
AC--TC--GC
 CC--CG--CG
  CG--GC--GC
       ...
```

只要 CG -- GC -- GC 就行
eg: cG AA GC AA GC
cG TT GC AT GC 都可以

C. Suffix-Trees
   1. Construction
      a) Start with an empty root and a full string S
      b) Add successively each suffix, such that edge labels on path from root to leaf spell the suffix, label leaf with starting position
      c) Use existing edges where possible
      d) Contract edges that have no branches and concatenate labels
      e) Repeat for all edges
      f) The number of leaves does not change
   2. Matching
      a) Start at root
      b) Traverse tree according to query pattern
      c) Stop, if leaf or end of pattern is reached
      d) Find start positions of P in S in the leaves below of current sub-tree
   3. Complexity
      a) Construction for string S of length n
         (1) $O(n)$ for a constant-size alphabet
      b) Tasks:
         (1) Search in S for a pattern P of length p: $O(p)$
         (2) Find matching statistics for P with p longest match between S and P: $\Theta(p)$
   4. **Useful Property: Search complexity is defined by the length of the pattern!**

D. Suffix-Arrays
   1. Definition: Sorted list of all suffixes of a string S
   2. Generation:
      a) Generate list of suffixes, sort them in lexicographical order and obtain the starting position of them
      b) Depth first traversal of a suffix tree

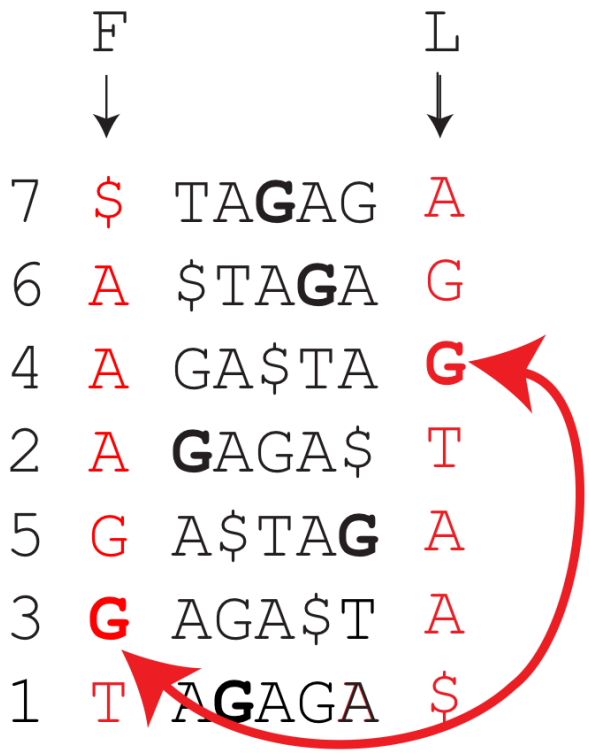


$	7	Suffix Array A
A$	6	
AGA$	4	
AGAGA$	2	
GA$	5	
GAGA$	3	
TAGAGA$	1	

3. Pattern Matching
   a) The Pattern Matching Problem
      (1) Suffix Array A: $A[k]$ is the start positions of k-th smallest suffix in $\{S_1, S_2, \ldots, S_n\}$, where S_i is the suffix starting on position I
      (2) Search Pattern P: substring of length $p \leq n$
      (3) Left Boundary $L_p = min(k : P \leq S_{A[k]} \text{ or } k = n + 1)$
         (a) The smallest suffix that is larger than P
         (b) One suffix can have P as a substring only when it is larger or equal than P
      (4) Right Boundary $R_p = max(k : S_{A[k]} < Q \text{ or } k = 0)$ where $Q = P\#$ and # being a character lexicographically larger than any other characters in S or P
         (a) The largest suffix that is smaller than Q
         (b) One suffix can have P as a substring only when it is smaller than Q
4. Search and Complexity*
   a) L_p and R_p fully define all matches of P in S
   b) Boundaries can be efficiently computed with binary search, pattern matching in $O(|P| log |S|)$
   c) Construction and storage of suffix array A in $O(|S|)$
      (1) Linear construction of suffix array is an advanced topic

E. Burrows-Wheeler Transform and FM-index
1. Construction
   a) Start with the full genome string
   b) Generate all rotations of the genome string
   c) Sort the matrix lexicographically
   d) Keep only the last column L and the index of the original string
2. Properties
   a) String L shows long stretches of the same symbol
   b) (Which) Allows for very efficient compression
   c) String L and index of original string are sufficient to reconstruct the original string
   d) After sorting, indices are equivalent to suffix array
3. Full-Text Minute-Space Index
   a) Construction
      (1) Add $-symbol to the genome end ($ is smaller than any other symbol, so no suffix can be the prefix of another suffix)
      (2) Apply Burrows-Wheeler Transform to string

b) Important Property: The k-th occurrence of character c in L corresponds to the k-th occurrence of character c in F



$L[3] = F[6] = G^{(2)}$

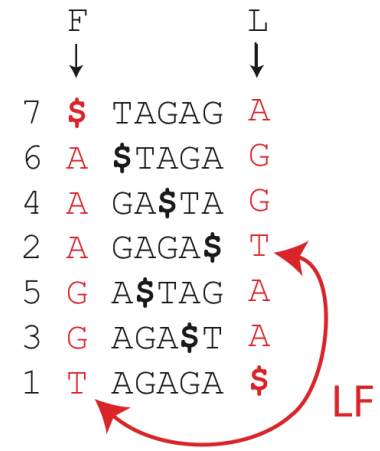
c) Pattern Matching with FM-indexing

(1) Compute LF

(a) C: C[k] = total number of occurrences of the characters < c in L

(b) Occ(c, k) = number of times c occurs in L[1,k]

(c) LF(i) = C[L(i)] + Occ(L[i], i)



Array C

$	A	G	T
0	1	4	6

Matrix Occ

	A	G	G	T	A	A	$
	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

$LF(4) = C[L[4]] + Occ(L[4], 4)$

$LF(4) = C[\mathbf{T}] + Occ(\mathbf{T}, 4)$

$LF(4) = 6 + 1$

$LF(4) = 7$

(2) The alignment algorithm*

(a) i <- p, c <- P[p]

(b) sp <- C[c] + 1

(c) ep <- C[c + 1]

(d) While ($sp \leq ep$ and $(i \geq 2)$ do

i) c <- P[i-1]

ii) sp <- C[c] + Occ(c, sp-1) + 1

iii) ep <- C[c] + Occ(c, ep)

iv) i <- i - 1

(e) End while

Locate Pattern Range:

$i \leftarrow p, \quad c \leftarrow P[p]$

$sp \leftarrow C[c] + 1$

$ep \leftarrow C[c + 1]$

while $(sp \leq ep)$ **and** $(i \geq 2)$ **do**

$c \leftarrow P[i - 1]$

$sp \leftarrow C[c] + Occ(c, sp - 1) + 1$

$ep \leftarrow C[c] + Occ(c, ep)$

$i \leftarrow i - 1$

end while

Array C

$	A	G	T
0	1	4	6

Matrix Occ

	1	2	3	4	5	6	7
$	0	0	0	0	0	0	1
A	1	1	1	1	2	3	3
G	0	1	2	2	2	2	2
T	0	0	0	1	1	1	1

Example

P = **AGA**, $p = |P| = 3$

Init:

$c \leftarrow P[3]$ =**A**

$sp \leftarrow C$[**A**] + 1 = 2

$ep \leftarrow C$[**A**+1] = C[**G**] = 4

Iteration 1:

$c \leftarrow$ **G**

$sp \leftarrow C$[**G**] + Occ[**G**, 1] + 1 = 5

$ep \leftarrow C$[**G**] + Occ[**G**, 4] = 6

Iteration 2:

$c \leftarrow$ **A**

$sp \leftarrow C$[**A**] + Occ[**A**, 5] + 1 = 3

$ep \leftarrow C$[**A**] + Occ[**A**, 6] = 4

d) LF-Mapping for Search*

(1) Search algorithm provides hit-range for pattern P in BWT-MAtrix index

(2) Translation into original Coordinates for S necessary

e) Complexity: O(|P|) for search where |P| is the length of search pattern

F. Summary for time complexity

1. K-mer indexing : O(p)
2. Suffix trees and Suffix arrays: O(plogL)
3. BWT and FM index: O(p)
4. L is genome length, p is the pattern length

Lecture 3:Alignment Algorithms

I. Formal definition of alignments, cost models and distance

A. Informal definition of alignments : Goal of a pairwise alignment is to identify the amount of commonality/similarity of two (sub)-sequences or to search for a given pattern in one or many target strings.

B. Formal definition of alignment

1. Definition of Alignment:Given sequences $A = a_1, \ldots, a_m \in \Sigma^*$ and $B = b_1, \ldots, b_n \in \Sigma^*$ a (global) alignment is a pair of sequences $U = u_1, \ldots, u_h$ and $L = l_1, \ldots, l_h$ of length $h \in [max(m, n), \ldots, m + n]$ such that
   a) U contains A and L contains B as a subsequence, respectively
   b) U contains the “gap” character “-” (h-m) times
   c) L contains the “gap” character “-“ (h-n) times
2. Cost: The cost C of an alignment (U,L) is defined as
$$C(U, L) = \sum_{i=1}^{h} c(u_i, l_i)$$
   where the cost-model c(alb) is the cost for aligning a and b
3. Edit Distance: Given two sequences A and B and a cost model c, we define their edit distance as the minimum alignment cost:
$$D(A, B) = \min_{(U,L) \in \mathbb{A}(A,B)} C(U, L)$$
   where $\mathbb{A}$ is the set of all possible alignments between A and B
   a) Edit distance can vary based on different cost models

Example I

Sequences A = `semester` and B = `minister` have edit distance 4 under the cost model $c(a, b) = 0$, if $a = b$ and 1 otherwise (Levenshtein distance). The only alignment (U, L) with minimum distance is

```
U = s e m e s t e r
L = m i n i s t e r
```

4

Example II

Sequences A = `semester` and B = `minister` have edit distance 6 under the cost model $c(a, -) = c(-, b) = 1$, $c(a, b) = 0, \forall a = b$ and $c(a, b) = 2$, otherwise. A possible alignment (U, L) with minimum distance is

```
U = s e m e - - s t e r
L = - - m i n i s t e r
```

II. Alignment algorithms

A. Needleman-Wunsch global alignment algorithm

1. Given two sequences $A = a_1, \ldots, a_m \in \Sigma^*$ and $B = b_1, \ldots, b_m \in \Sigma^*$, let us denote with $d_{i,j}$ the distance $D(a_1 \ldots a_i, b_1, \ldots, b_j)$ and with $d_{m,n} = D(A, B)$
2. For all $0 \leq i \leq m$ and $0 \leq j \leq n$ with $i + j > 0$, the edit distance $d_{i,j}$ can be computed using the following recurrence

$$d_{i,j} = min \begin{cases} d_{i-1,j-1} + c(a_i, b_j) \\ d_{i-1,j} + c(a_i, -) \\ d_{i,j-1} + c(-, b_j) \end{cases}$$

by initialising $d_{0,0} = 0$ and interpreting $d_{i,j}$ as ∞, if $i < 0$ or $j < 0$

Example I

Sequences A = `semester` and B = `minister` have edit distance 4 under the cost model $c(a, b) = 0$, if $a = b$ and 1 otherwise.

		s	e	m	e	s	t	e	r
	0	1	2	3	4	5	6	7	8
m	1	1	2	2	3	4	5	6	7
i	2	2	2	3	3	4	5	6	7
n	3	3	3	3	4	4	5	6	7
i	4	4	4	4	4	5	5	6	7
s	5	4	5	5	5	4	5	6	7
t	6	5	5	6	6	5	4	5	6
e	7	6	5	6	6	6	5	4	6
r	8	7	6	6	7	7	6	5	4

cost model:

```
c(a, b) = 0, if a == b
c(a, b) = 1, if a != b
c(a, -) = 1
c(-, b) = 1
```

```
U = s e m e s t e r
L = m i n i s t e r
```

Example II

Sequences A = `semester` and B = `minister` have edit distance 6 under the cost model $c(a, -) = c(-, b) = 1$, $c(a, b) = 0, \forall a = b$ and $c(a, b) = 2$, otherwise.

		s	e	m	e	s	t	e	r
	0	1	2	3	4	5	6	7	8
m	1	2	3	2	3	4	5	6	7
i	2	3	4	3	4	5	6	7	8
n	3	4	5	4	5	6	7	8	9
i	4	5	6	5	6	7	8	9	10
s	5	4	5	6	7	6	7	8	9
t	6	5	6	7	8	9	6	7	8
e	7	6	5	6	7	8	7	6	7
r	8	7	6	7	8	9	8	7	6

cost model:

```
c(a, b) = 0, if a == b
c(a, b) = 2, if a != b
c(a, -) = 1
c(-, b) = 1
```

```
--M-INISTER
  |  ||||
SEME---STER

--MI-NISTER
  ||   ||||
SEM-E--STER

--MIN-ISTER
  |||  ||||
SEM--E-STER

--MINISTER
  |  ||||
SEM-E-STER

--MINISTER
  |  ||||
SEME--STER

--MINI-STER
  |||  ||||
SEM---ESTER
```

3. Complexity: The time and space complexity of dynamic programming algorithm to fill the matrix we just saw is $\Theta(mn)$

B. Alignment in linear space: Hirschberg algorithm

1. Based on the observation that only the preceding, directly neighbouring cells are needed, one can derive an algorithm using only O(m) space, assuming that |A| = m, |B| = n and $m \leq n$
2. Time is still quadratic O(nm)

		s	e	m	e	s	t	e	r
	0	1	2	3	4	5	6	7	8
m	1	1	2	2	3	4	5	6	7
i	2	2	2	3	3	4	5	6	7
n	3	3	3	3					
i									
s									
t									
e									
r									

] -> Delete them

3. To trace back, a divide-and-conquer strategy is based on observation that if

$$(U, L) = NW(A, B)$$

is the optimal alignment of (A,B), and $A = A^l + A^r$ is an arbitrary partition of A, there exists a partition $B^l + B^r$ of B such that

$$NW(A, B) = NW(A^l, B^l) + NW(A^r, B^r)$$

C. Banded Alignment

1. If an upper bound for the edit distance, space and running time can be bound to $O(dm)$, assuming that $m \leq n$.
2. Each time the optimal path changes the diagonal, a cot of c(a,-) or c(-,b) is incurred.
3. Given a threshold t and the cost for insertion/deletion of Δ, then the diagonal zone to be considered is

$$Z = [-\lceil t/(2\Delta) - (n-m)/2\rceil \ldots \lceil t/(2\Delta) + (n-m)/2\rceil]$$

Example for diagonal zone *Z*

		s	e	m	e	s	t	e	r
	0	1	2						
m	1	1	2	2					
i	2	2	2	3	3				
n		3	3	3	4	4			
i			4	4	4	5	5		
s				5	5	4	5	6	
t					6	5	4	5	6
e						6	5	4	6
r							6	5	4

cost model:

```
c(a, b) = 0, if a == b
c(a, b) = 1, if a != b
c(a, -) = 1
c(-, b) = 1
```

recurrence:

$$d_{i,j} = \min\{d_{i-1,j-1} + c(a_i,b_j),\ d_{i-1,j} + c(a_i,-),\ d_{i,j-1} + c(-,b_j)\}$$

With a threshold $t = 4$ and an indel cost of $\Delta = 1$, the diagonal zone can be restricted to $Z = [-2 \ldots 2]$.

D. Approximate matching

1. Often it is not required to match the full strings but identify occurrences of a shorter pattern P in a longer target string S.
2. Initialising the first row in the dynamic programming matrix to 0 allows for multiple starting positions in S
3. All occurrences of P can be found in O(mn) time and O(m) space
4. Example:

Example

		b	a	n	a	n	a	s
	0	0	0	0	0	0	0	0
a	1	1	0	1	0	1	0	1
n	2	2	1	0	1	0	1	1
a	3	3	2	1	0	1	0	1

cost model:

```
c(a, b) = 0, if a == b
c(a, b) = 1, if a != b
c(a, -) = 1
c(-, b) = 1
```

E. Dual problem of global alignment

1. Instead of finding a minimum distance alignment under a given cost model, for biological sequences one often solves the dual problem
2. Global alignment:
   a) Given sequences $A = a_1 \dots a_m \in \Sigma^*$ and $B = b_1 \dots b_n \in \Sigma^*$ and a scoring model s, we search an alignment with maximal score

$$S(A,B) = \max_{(U,I)\in\mathbb{A}(A,B)} W(U,L)$$

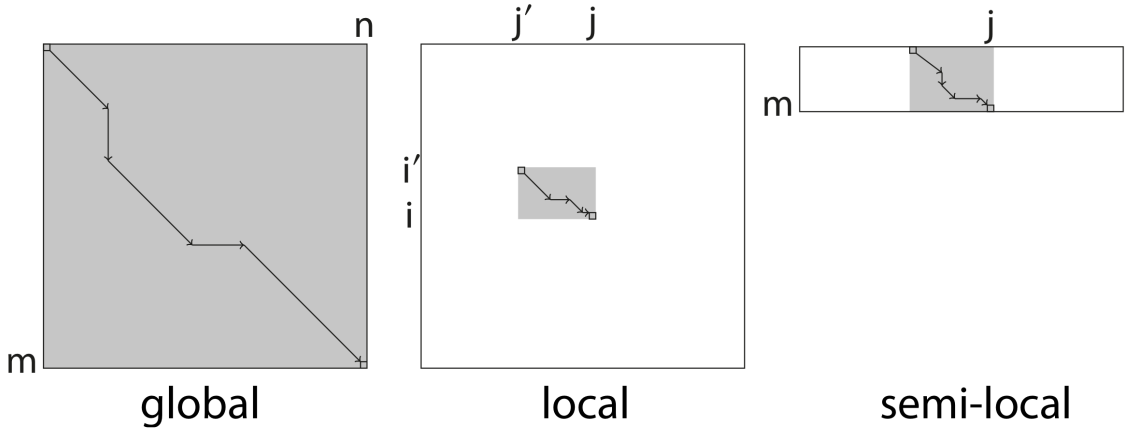
Where $\mathbb{A}(A,B)$ denotes all valid alignments of A and B and $W(U,L) = \sum_{i=1}^{h} s(u_i, l_i)$ denotes the score of an alignment

3. s(-,b) and s(a, -) are often set to a negative constant $-\delta$, also denoted as **gap penalty**
4. Initialisation of dual problem :

$$\begin{cases} s_{0,0} = 0 \\ s_{i,0} = -i\sigma, \forall 1 \le i \le m \\ s_{0,j} = -j\delta, \forall 1 \le j \le n \end{cases} \qquad s_{i,j} = \begin{cases} s_{i-1,j-1} + x(a_i, b_j) \\ s_{i-1,j} - \delta \\ s_{i,j-1} - \delta \end{cases}$$

F. Local and semi-local alignments

1. local alignment: find the local matches of two substrings of two given sequences A and B
2. semi-local alignments finds the occurrences of a shorter substring B in a long target string A (for instance a sequencing read in a genome)



3. local alignment:

a) Given sequences $A = a_1 \ldots a_m \in \Sigma^*$ and $B = b_1 \ldots b_n \in \Sigma^*$ and a scoring model s, the **local alignment** L(A,B) is the maximum scoring global alignment over all substrings of A and B such that

$$L(A,B) = \max\{S(A_{i'..i}, B_{j'..j}) \,|\, \forall i', i \in [1..m], j', j \in [1,n]\}$$

b) We can compute the local alignment score $l_{i,j}$ with a similar recurrence as before

$$l_{0,j} = l_{i,0} = 0, (\forall i \in [0,m], j \in [0,n])$$

$$l_{i,j} = \max \begin{cases} 0 \\ l_{i-1,j-1} + s(a_i, b_j) \\ l_{i-1,j} - \sigma \\ l_{i,j-1} - \sigma \end{cases}$$

Example

		b	a	n	a	n	a	s
	0	0	0	0	0	0	0	0
n	0	0	0	1	0	1	0	0
a	0	0	1	0	2	1	2	1
n	0	0	0	2	1	3	2	2
a	0	0	1	1	3	2	4	3

scoring model:

```
s(a, b) = 0, if a != b
s(a, b) = 1, if a == b
s(a, -) = -1
s(-, b) = -1
```

Instead of max, one can also keep the best top-scoring alignments.

Question: What is a sufficient cutoff? What does top-scoring mean?

One can design the scoring scheme so that local alignments with scores less than zero are not statistically significant.

III. Scoring functions

A. Assume that string A and B were generated from a **random model** R with probability q_a for a character $a \in \Sigma$, Then the joint probability of sequence A and B under R is

$$P(A,B \,|\, R) = \prod_i q_{a_i} q_{b_i}$$

B. Under the assumption that a and b have a common ancestor, consider the **match model** M, assigning a joint probability p_{ab} to the substitution $a \to b$

C. The probability of a trivial alignment between A and B is then

$$P(A,B\,|\,M) = \prod_i p_{a_i b_i}$$

D. Odds Ratio

1. The ratio of match model and random model is also known as the **odds ratio**

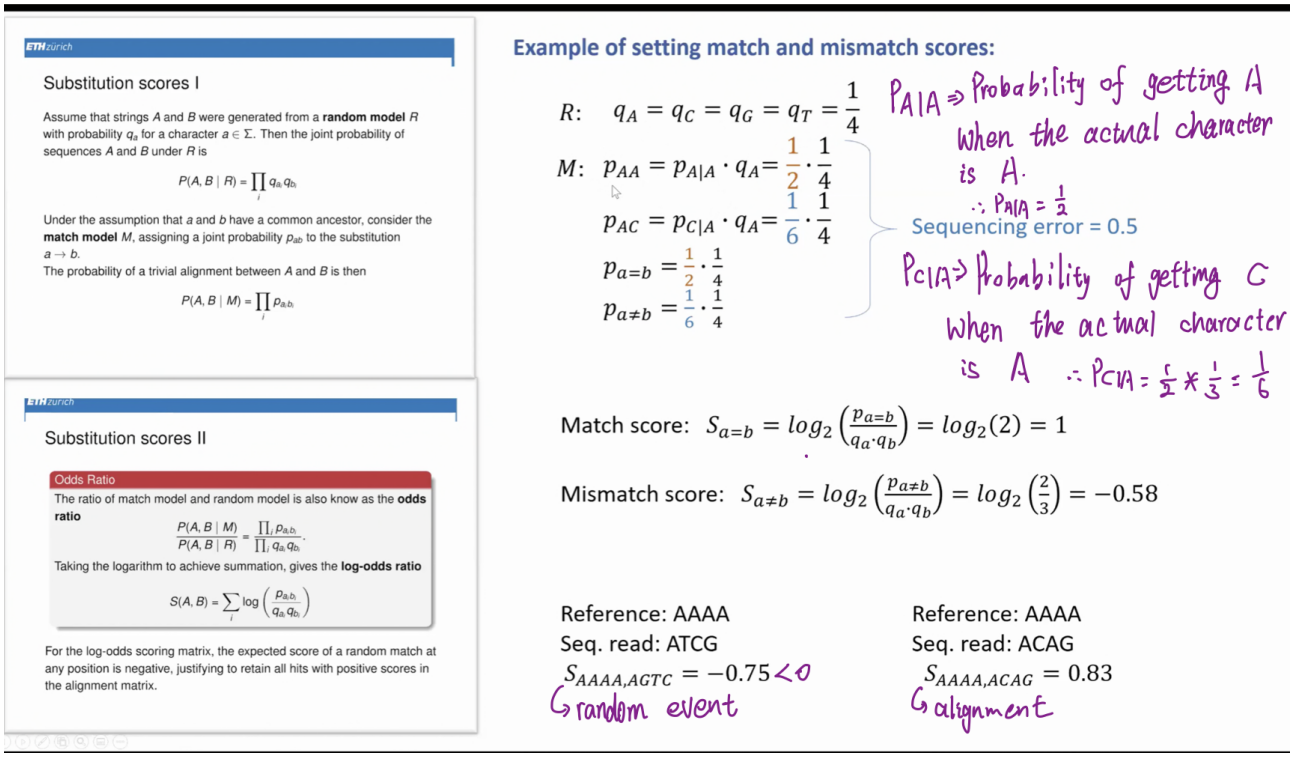
$$\frac{P(A,B\,|\,M)}{P(A,B\,|\,R)} = \frac{\prod_i p_{a_i b_i}}{\prod_i q_{a_i} q_{b_i}}$$

2. Taking the logarithm to achieve summation, gives the **log-odds ratio**

$$S(A,B) = \sum_i \log\left(\frac{p_{a_i b_i}}{q_{a_i} q_{b_i}}\right)$$

3. For the log-odds scoring matrix, the expected score of a random match at any position is negative, justifying to retain all hits with positive scores in the alignment matrix

E. Example of setting match and mismatch scores



IV. Local alignment with arbitrary gap cost

A. In bioinformatics, it makes sense to introduce different penalties for opening “gaps” and for extending them

B. Construction of the dynamic programming score matrix $L = [l_{i,j}]$
   1. Initialising $l_{0,j} = l_{i,0} = 0, \forall 0 \leq i \leq m, 0 \leq j \leq n$
   2. Computing the local alignment score $l_{i,j}$ with a similar recurrence as for the linear gap cost with a small change:

$$l_{i,j} = \max \begin{cases} 0 \\ l_{i-1,j-1} + s(a_i, b_j) \\ l_{i-1,j} - \sigma \\ l_{i,j-1} - \sigma \end{cases} \quad \text{becomes} \quad l_{i,j} = \max \begin{cases} 0 \\ l_{i-1,j-1} + s(a_i, b_j) \\ \max_{1 \leq k \leq i}(l_{i-k,j} - \gamma(k)) \\ \max_{1 \leq k \leq j}(I_{i,j-k} - \gamma(k)) \end{cases}$$

   3. Where $\gamma(k) \geq 0$. usually affine, but can be logarithmic or quadratic:

$$\gamma(k) = \begin{cases} \alpha + \beta(k-1) \\ \alpha + \beta ln(k) \\ \alpha + \beta(k-1)^2 \end{cases}, \ \alpha \geq \beta \geq 0$$ are gap-open & gap-extension penalties

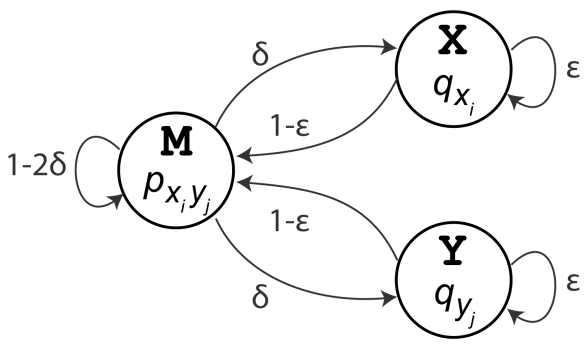
C. Complexity:
   1. Initially O(mn) space and $O(m^2 n)$ time, where m and n $(m \leq n)$ are lengths of the query and target texts.
   2. Since then optimised to $O(mn)$ time for an affine gap penalty
   3. Space complexity was reduced to O(n)

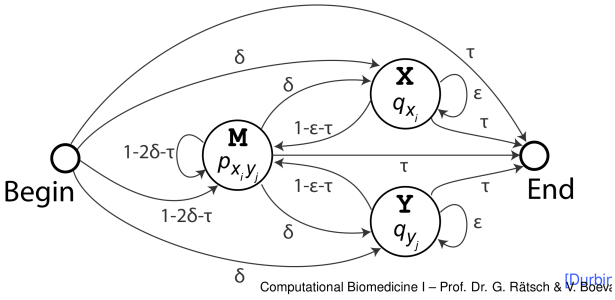
D. Needleman-Wunsch global alignment algorithm
   1. The Neeedleman-Wunsch global alignment algorithm can be also similarly extended for the arbitrary gap cost

V. Alignment as a probabilistic model*

A. An alignment can also be seen as a finite state automaton, with a matching state M and insertion and deletion states X and Y.

B. Using the probabilistic models M and R for the emission probabilities $p_{x_i y_i}, q_{x_i}$ and q_{y_i}, and a set of probabilities for state transitions, one can derive a simple probabilistic model and ultimately a full Pair Hidden Markov Model. To find the best alignment (the best path through the graph): Veterbi algorithm for pair HMM.





VI. Messages From Tutorial

A. Early realisation of affine gap penalty algorithm

Smith-Waterman-Gotoh alignment algorithm

- Smith-Waterman only works when gap open (δ_O) and gap extension (δ_E) penalties are equal
- Gotoh (1982) allowed for affine gap penalties. Every few years, a better implementation of this algorithm comes out.
- Additional matrices *e* and *f* which store the best scores when the last step was a gap in the query and reference sequence, respectively
- Need to store additional information to follow correct traceback path

$$l_{i,j} = max \begin{cases} 0 \\ l_{i-1,j-1} + s(a_i, b_j) \\ e_{i,j} \\ f_{i,j} \end{cases}$$

$$e_{i,j} = max \begin{cases} l_{i,j-1} - \delta_O \\ e_{i,j-1} - \delta_E \end{cases}$$

$$f_{i,j} = max \begin{cases} l_{i-1,j} - \delta_O \\ e_{i-1,j} - \delta_E \end{cases}$$

B. Advanced banding strategy

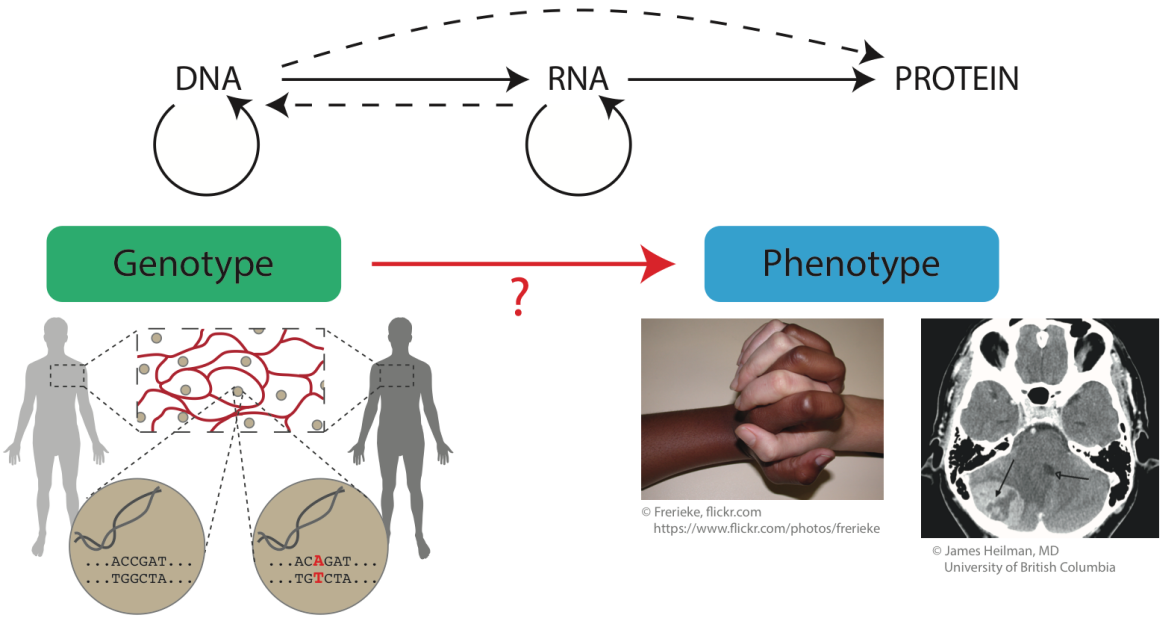
1. X-drop: set all cells whose scores are more than X away from the best score to $-\infty$
2. Z-drop : cut off the alignment if the score along the diagonal is more than Z away from the best score.ß

Lecture 4: Alignment with Variation and Graph Data Structures

---

I. Motivation from Genotype to Phenotype

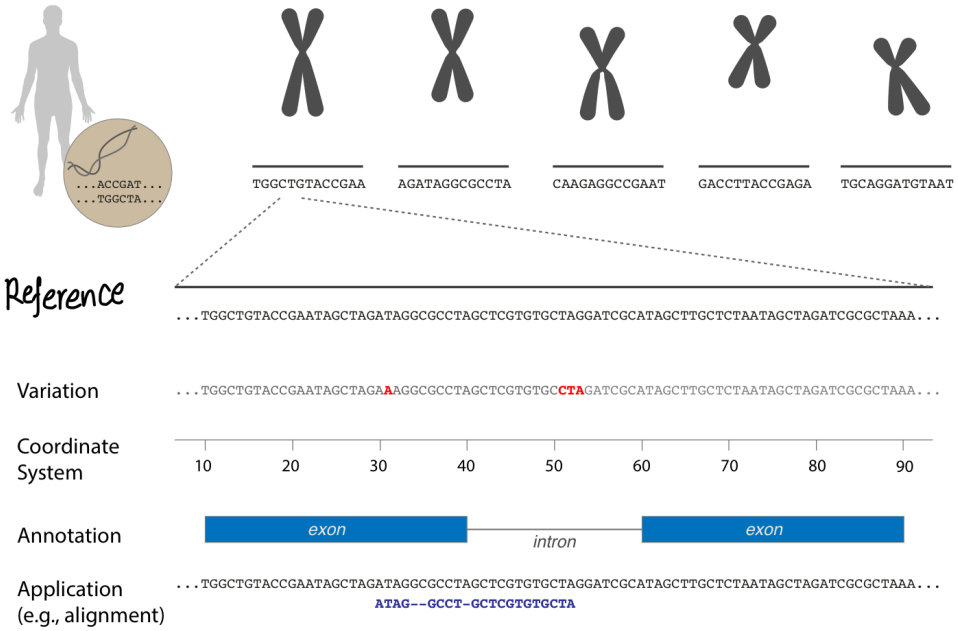
A. Central Dogma of Molecular Biology



II. Motivation Linear reference strings as a tool

A. We use the linear reference strings to

1. Find and document variations
2. Annotated the genome
3. Develop Coordinate System
4. Develop further applications like alignment



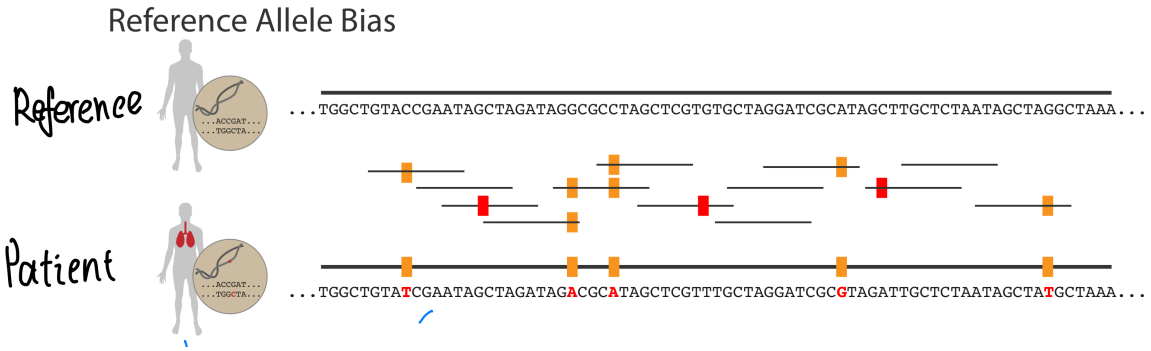
B. Limitations of linear approach: poor definition of reference genome
   1. A reference genome can
      a) originate from a single individual
      b) originate from a consensus of a population
      c) only contain a subset, e.g. functional, elements
      d) consist of the superset of sequences ever discovered
   2. Practical problems
      a) Many possibilities to express the same sequence variant relative to a reference
      b) Different versions change the whole coordinate system
      c) Allele biases and scalability issues

III. Motivation : Genome sequencing today is cheap and easy
   A. The price of sequencing is dropping at a higher speed than the Moore's Law

IV. Motivation: Alignment is at the heart of many applications

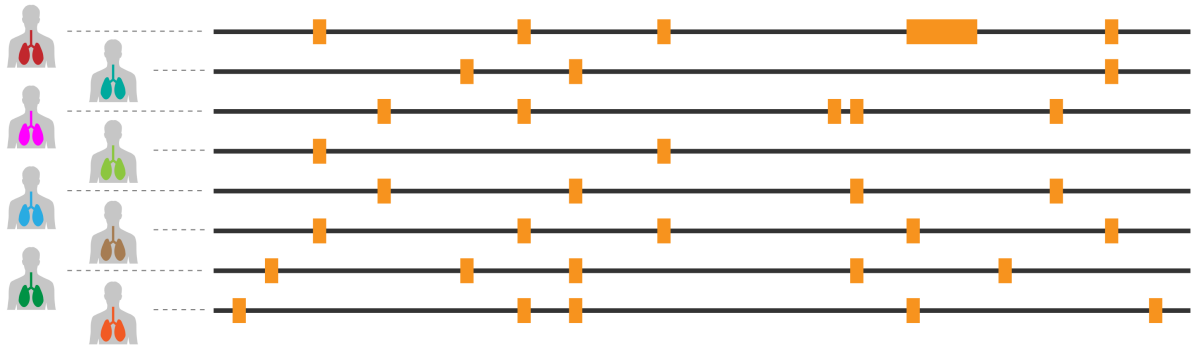
———————————————Motivation———————————————

V. Current Limitations of alignment
   A. Reference Allele Bias



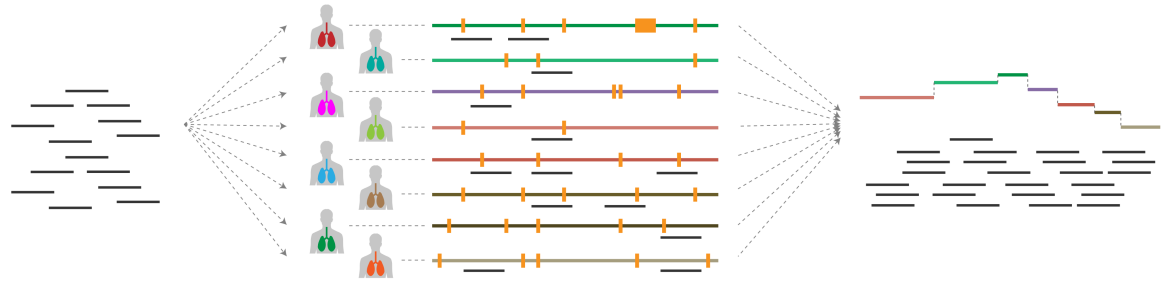
   1. The reference genome we are using now has some variations that is not relevant to the disease that the patient has (marked in orange)
   2. It would be better to just align the sequenced region to the individual genome of the patient so that it can get rid of the reference allele bias

   B. Scalability and Complexity



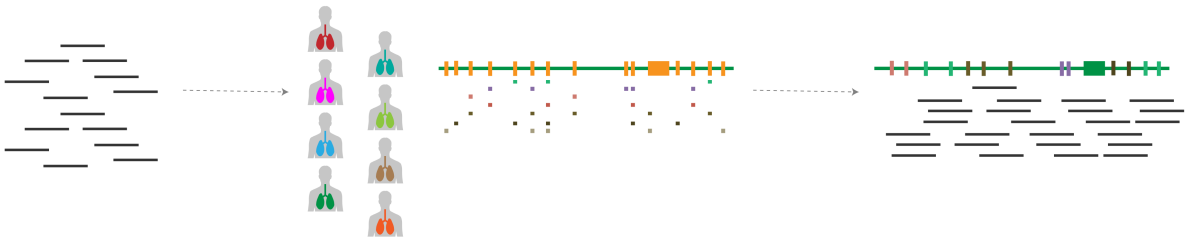
VI. Possible Solutions

A. Alignment to individual reference sequences

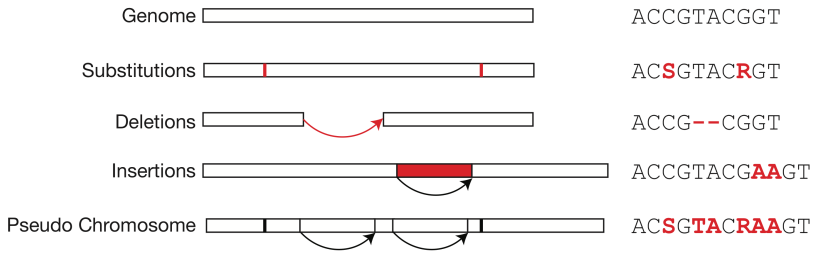


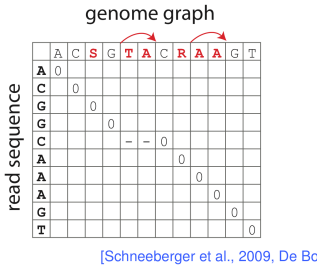
1. Instead of having a single reference genome, we have a collection of individual reference genomes
2. When a new patient/individual get sequenced, we align the reads to the collection of individual reference genomes and find the optimal alignment
3. Infer the genotype of the new patient/individual from the genotypes that we already have
4. Very very expensive
   a) population is large
   b) each genome is large
   c) alignment to all of them is therefore very very expensive

B. Alignment to single reference + variation (reference based compression)



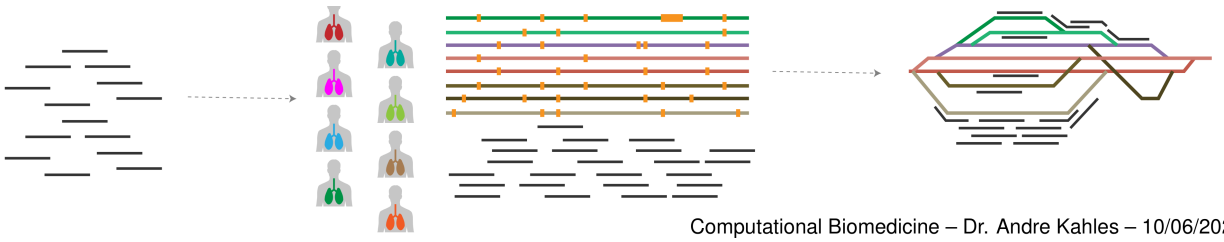
1. Most of the positions in the genome are identical within the population we have
2. Encode the rest by storing only one reference and variations based on that reference
3. For alignment, we align to the reference and a subset of the variations encoded in that region
4. Sequence alignment with variation
   a) Use IUPAC ambiguity codes to represent substitutions
   b) Integrate deletions and insertions as zero-cost gaps
   c) Collapse all variations into graph-like structure
   d) Dynamic alignment against local variation graph





5. Limitations
   a) Still Quiet Expensive
   b) Combinatorial explosion for complex events
   c) local graph needs to be rebuild for each alignment
   d) seed index not aware of variation
   e) difficult to encode long range haplotypes

C. Alignment to a pan-genome(direct sequence compression)
   1. Instead of building a local graph every time there's an alignment, we can build a graph that encodes the variation of the population
   2. Then align the new individual to that graph

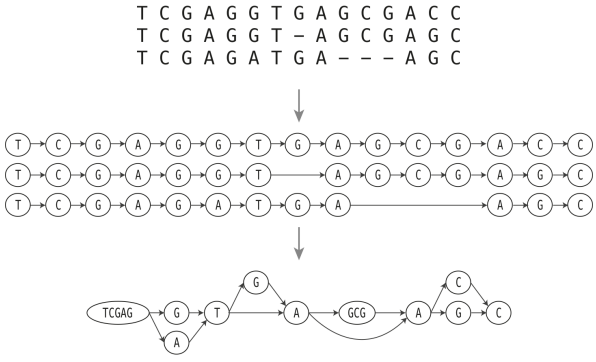


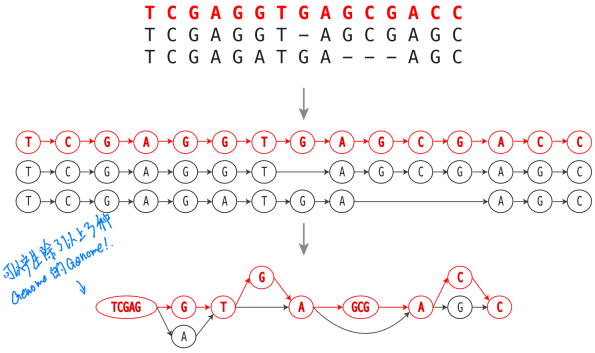
VII. Genome Graphs

A. Idea of a genome graph is to represent large and complex sequence information while
   1. compressing redundant information
   2. allowing for fast decompression on the fly
   3. allowing for efficient exact and inexact search (alignment)

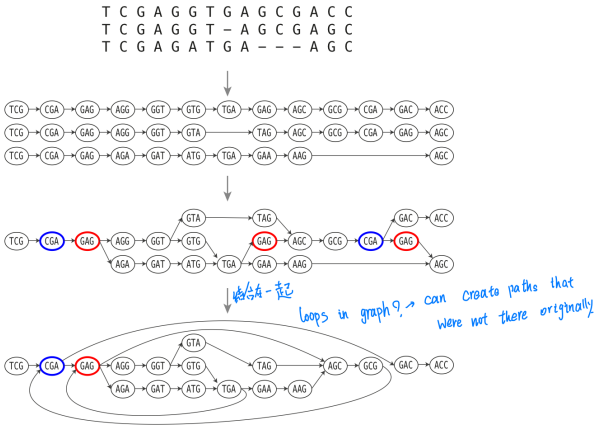
B. Types of genome graphs
   1. string graphs (variation graphs, overlap graphs, …)
   2. k-mer graphs

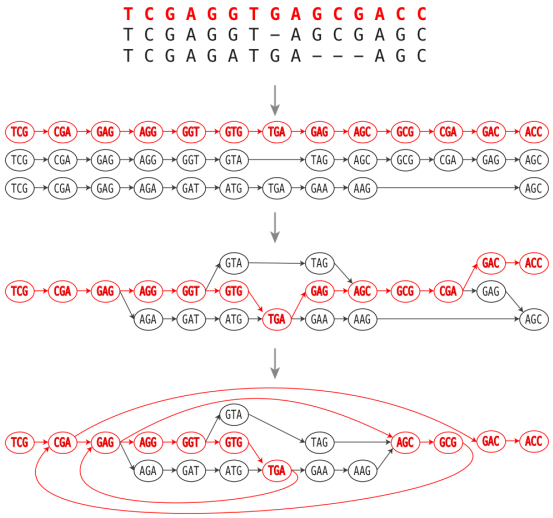
C. String graphs





D. K-mer Graphs

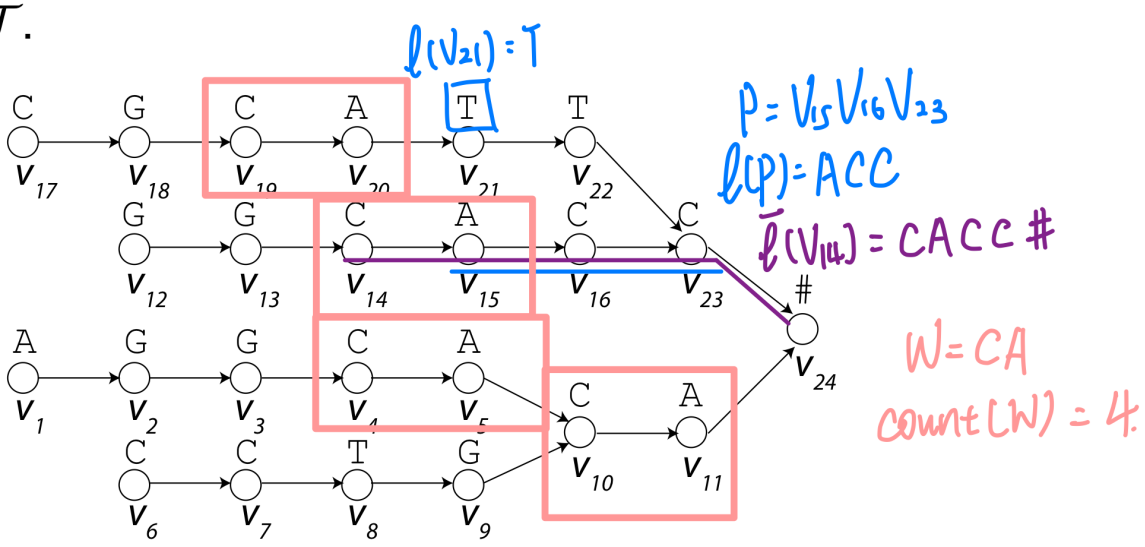




VIII. Burrows-Wheeler Transform on Trees

A. Given a rooted, labeled tree $T = (V, E, \Sigma)$, with directed edges (u, v) from children to parents with each node labeled $\ell(v_i) \in \Sigma$, we define a **path** $P = v_i, v_{i+1}, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ and a **path label** $\ell(P) = \ell(v_i) \cdot \ell(v_{i+1}) \cdots \ell(v_k)$. We further define the **extended label** $\bar{\ell}(v_i)$ as $\ell(v_i, v_{i+1} \ldots, v_r)$ and assume a **total order** between all children u, v of any given node, implying a total order on T

on $\mathcal{T}$.



B. Important Observations

1. If T were a chain, $\ell(v_i)$ would be a suffix of T
2. A string W can be the label of multiple paths, we use $count(W)$ to represent this number

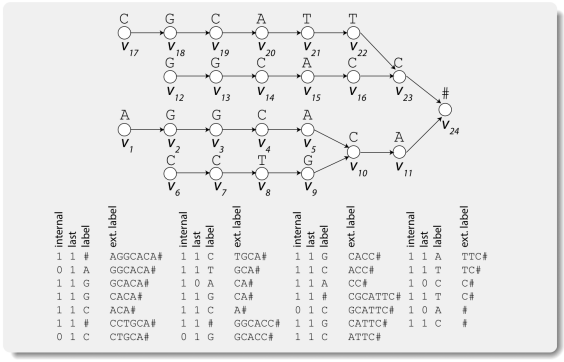
3. A string W is the prefix of the extended label $\ell(v_i)$ of $count(W)$ many nodes v_i

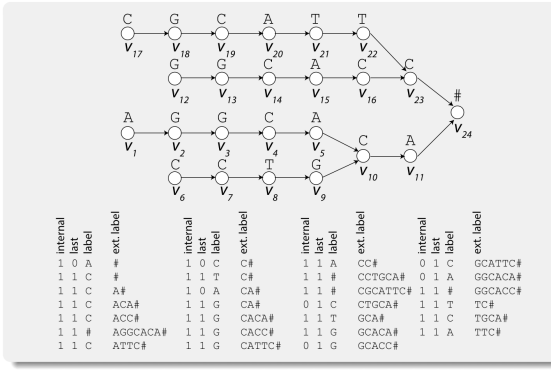
C. For each node $v \in V$ we define

1. $last(v) = \begin{cases} 1, & \text{iff v is greater than any of its siblings} \\ 0, & \text{otherwise} \end{cases}$
2. $internal(v) = \begin{cases} 1, & \text{iff v is not a leaf} \\ 0, & \text{otherwise} \end{cases}$
3. We now collect for all $v \in V$ and for all $(u, v) \in E$ in the order implied through $\prec^*$ a list L of 4-tuples:

$$L[i] = \begin{cases} (\ell(u), \bar{\ell}(v), last(u), internal(u)), & \textbf{if v is internal} \\ (\#, \ell(v), 1, 1) & \textbf{otherwise} \end{cases}$$

4. and apply a stable sort on L over the second element (the extended labels) of the tuples





D. Burrows Wheeler index for trees

1. For a given labeled tree T, its **Burrows-Wheeler index** BWT_T consists of a data structure containing
   a) the array of characters labels
   b) the binary vector last
   c) the binary vector internal
   d) the first column of $ext.labels$ encoded as offset vector C
2. Observations
   a) nodes v that share a prefix of their $\bar{\ell}(v)$ are in a contiguous interval in BWT_T
   b) the in-degree of v is directly encoded in $last$
   c) the k-th occurrence of c in labels corresponds to the $k-th$ (unique) extended label beginning with c

E. Tree BWT Traversal - top-down

1. Top-down Traversal

a) Given the range of node v in BWT_T as $\bar{v} = [p\,..\,q]$, we can compute the range of its k-th child with label c as

$$child(\bar{v}, c, k) = [select_q(last, LF(p-1,c) + k - 1) + 1..$$
$$select_1(last, LF(p-1,c) + k)]$$
$$\text{with} \quad LF(i,c) = Count[c] + rank_c(label, i)$$

2. Left-extension of suffix W

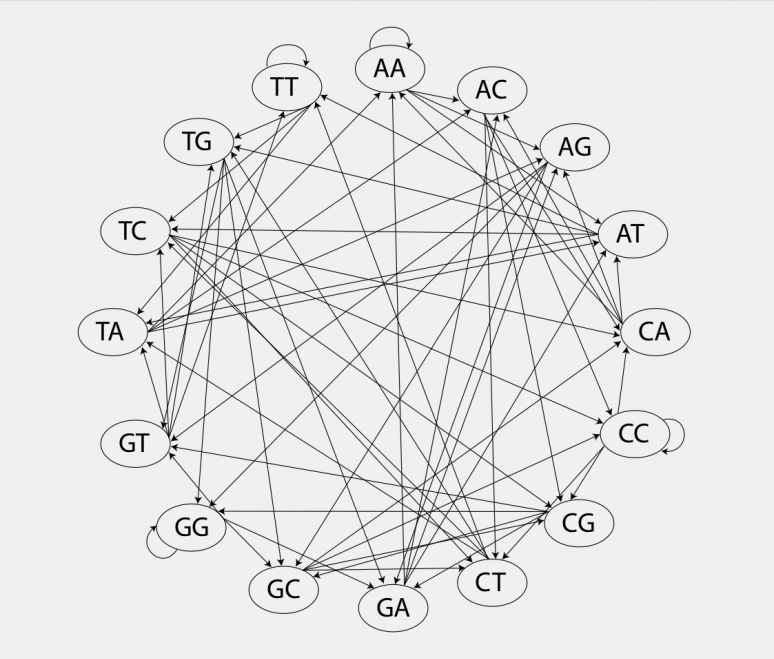
a) Given a range $\bar{W} = [p\,..\,q]$ of nodes that all have string W as prefix for their extended label, we can compute the range of the left extension cW as

$$extendLeft(\bar{W}, c) = [select_1(last, LF(p-1,c)) + 1..$$
$$select_1(last, LF(q,c))]$$

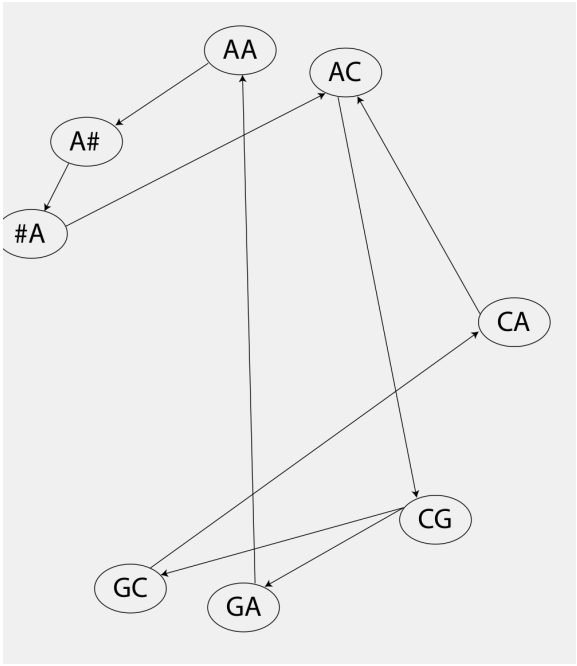
IX. BWT on De Bruijn Graphs

A. De Bruijn Graphs

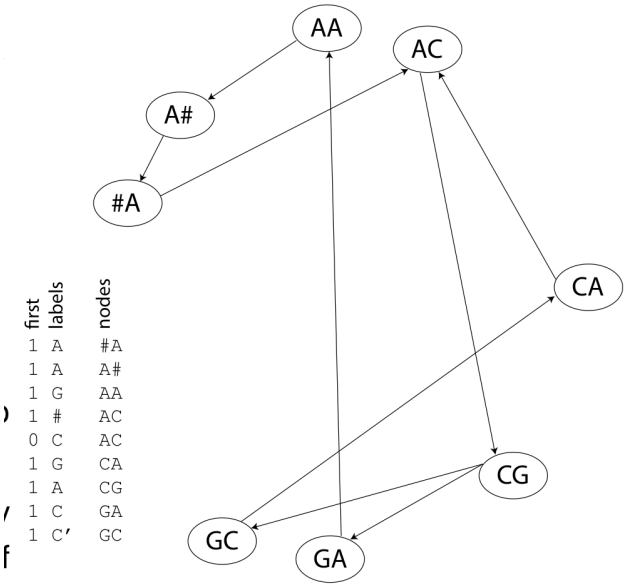
1. De Bruijn Graph: For a fixed integer k and an alphabet Σ, the graph (V, E) with $V = \Sigma^{k-1}$ (the set of all distinct strings from Σ of length $k-1$) and $E = \{(v,w) \,|\, v, w \in V \text{ and } v[2..k] = w[1..k-1]\}$ is called a **de Bruijn graph** of degree k: $DBG_{\Sigma,k}$. Edge (v,w) describes a string of length k: $DBG_{\Sigma,k}$. Edge (vow) describes a string of length k.
2. We will further only consider a special case, namely the sub-graph $DBG_{T,k}$, which contains only edges (v,w) that form a k-mer string that is a substring of a given circular string T
3. Example: fully connected de Bruijn graph



4. Example: Connections of de Bruijn graph $DBK_{T,3}$ with $T = ACGCACGAA\#$

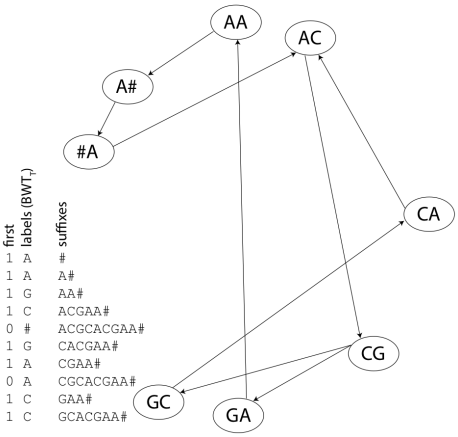


B. BWT on De Bruijn Graphs
   1. Sort nodes lexicographically by labels
   2. assign incoming labels of incoming edges to nodes
   3. mark first position of every interval with identical node labels
   4. augment alphabet in labels to mark last outgoing edges
   5. Representation only stores topology of the graph, not the frequency of occurrence in T



first	labels	nodes
1	A	#A
1	A	A#
1	G	AA
1	#	AC
0	C	AC
1	G	CA
1	A	CG
1	C	GA
1	C′	GC

C. frequency-aware DBG
   1. store the frequency of each k-mer in T
   2. Generate BWT_T of string T
   3. for a given k, set to 1 all fields in an array $first$ that mark the first item in the range of common $k-1$ prefixes in the list of suffixes

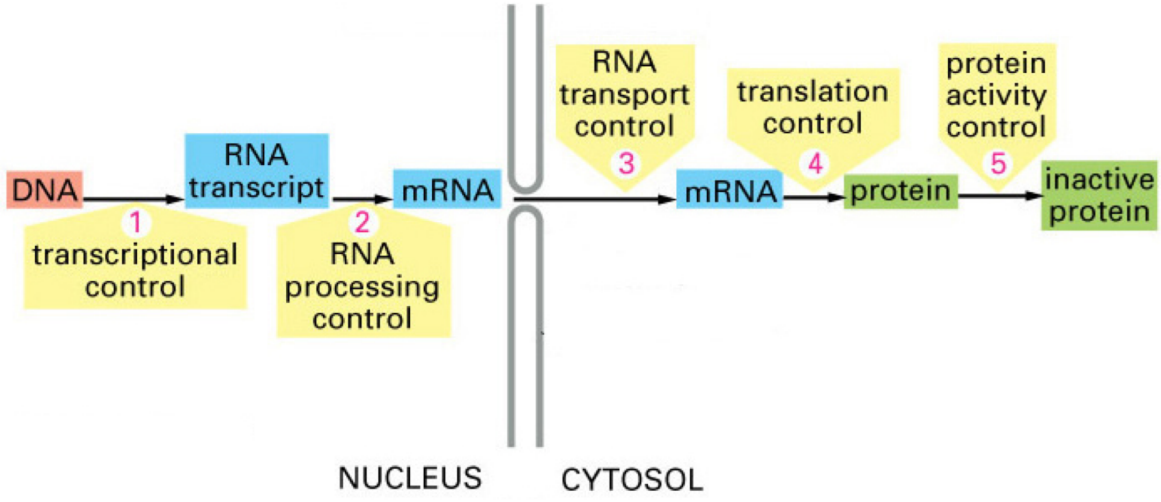


first	labels (BWT_T)	suffixes
1	A	#
1	A	A#
1	G	AA#
1	C	ACGAA#
0	#	ACGCACGAA#
1	G	CACGAA#
1	A	CGAA#
0	A	CGCACGAA#
1	C	GAA#
1	C	GCACGAA#

Lecture 5: RNA-Sequencing & Gene Expression Quantification

I. Motivations

A. Learning about the genome



1. Goal: Learn to predict what these processes accomplish
   a) Given the DNA, …, predict all gene products
      (1) Not only the sequence but also how many of them
      (2) $f(DNA,123) = RNA \quad F(RNA,4\ \&\ 5) = protein$
      (3) Estimating f, F amounts to cracking the codes of transcription, epigenetic, splicing ,… , protein interactions,….
2. Transcription & RNA Processing in Eucaryotes
   a) Newly synthesised pre-mRNA is capped with a 5'-cap
   b) Introns are spliced from ore-mRNA
   c) A polyA-tail is added to the 3' terminus of pre-mRNA
3. Key Research Questions
   a) Characterise an organism's full complement of genes
      (1) Find new, possibly noncoding genes → regulatory function
      (2) Compare genes among organisms, strains, individuals, tissues
      (3) Identify alternative splice forms or transcript ends
         (a) alternative splice forms are the reason for the polymorphism when cutting introns
   b) Understand (post-) transcriptional regulation
      (1) Monitor transcriptome changes between tissues or in response to environmental changes
      (2) Knock-out/knowck-down analysis of regulators
         (a) identify regulated targets with significant expression changes

(3) Identify regulatory binding sites (e.g. ChIP-seq)

c) Understand differences between populations, individuals, cells

(1) connect transcriptome-phenotype to underlying genotype

(2) Effect of rare or somatic mutations.

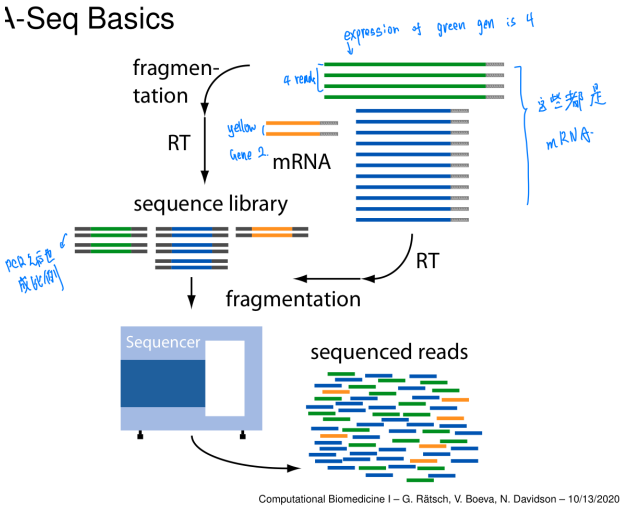
(a) Germline mutations: mutations you get when you're born

(b) Somatic mutations: mutations you get during your life time

i) only a part of your cells can have somatic mutations

II. RNA-sequencing mechanism

A. RNA-Seq Basics



1. RNA-Seq : How much of a certain gene is expressed in a sample
2. Fragmentation: Chop the mRNA into simi-uniform pieces like 50bp or 100 bp.
3.Then it goes to Reverse Transcription (RT)
(a)PCR
4.You can either do fragmentation first or do Reverse Transcription(PCR) first, but in the end you always have a library of sequences hopefully in the same proportion as in the cell
5. Then you put them into sequncer and then we are able to see the sequenced reads in the same proportion as in the cell

B. Biological Question ⇒ Analysis

1. New genome & variation (qualitative)
   a) Sequence DNA ⇒ assemble [⇒ align ⇒ detect]
   b) Sequence ⇒ align to DNA ⇒ detect
2. New gene / transcript (qualitative)
   a) Sequence RNA ⇒ assemble [⇒ align ⇒ detect]
   b) Sequence RNA ⇒ align to DNA ⇒ detect
3. Gene/transcript expression (quantitative)
   a) Sequence RNA ⇒ align to known RNAs ⇒ count read depth
   b) Sequence RNA ⇒ align to DNA ⇒ count
4. DNA.RNE-Protein binding (quant-&qualitative)
   a) Binding assay ⇒ Sequencing ⇒ align ⇒ identify peaks
   b) ChIP-seq

C. RNA Transcripts and Library Preparation

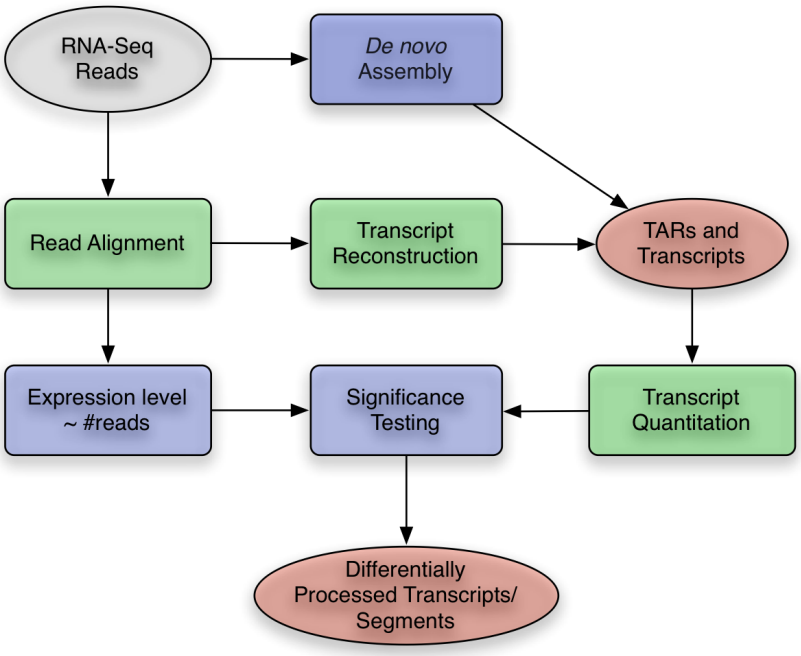
1. There are many different kinds of RNAs:
   a) Protein-coding mRNAs
   b) Noncoding RNAs
      (1) Structural RNAs (e.g. rRNAs, tRNAs,…)

(2) Small RNAs (e.g. miRNAs, endogenous siRNAs, …)
(3) Antisense / promoter-associated transcripts

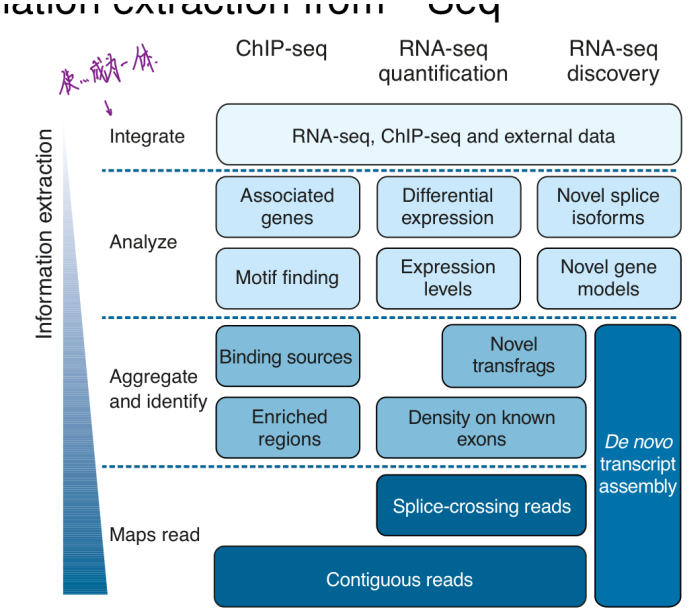
2. Analysis of biological sample starts with sample/library preparation. Depending on which RNAs should be targeted, different preparation strategies have to be used.

D. RNA-Seq Biases
1. Biases due to
   a) Samp prep: cDNA library construction
   b) Sequencer: Sequencing
   c) Computational pipeline: Read mapping

E. Common RNA-Seq Analysis Steps

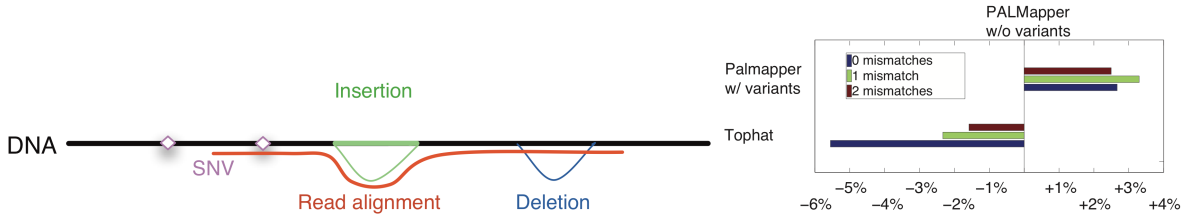


F. Information extraction from *-Seq

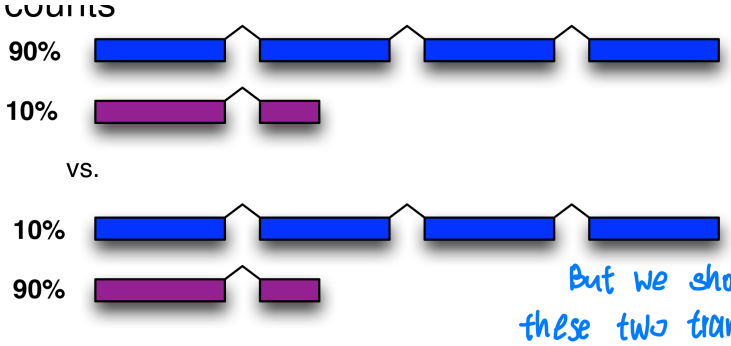


[Pepke et al., 2

III. Gene expression counting
  A. Gene Expression Estimation
    1. Idea: Use the number of reads mapping to a gene as an estimate for the gene expression
    2. Problem: Read number scales with the total number of reads and transcript length
    3. Approach: Normalize read count, by the
      a) Length of the transcript (sum of exotic regions in kilo bases)
      b) Total number of reads (in millions)
      c) Reads per kilobase per million mapped reads (RPKM)
    4. Alternative quantity for paired end sequencing (2 reads / fragment)
      a) Fragments per kilobase per million mapped reads
  B. Caveats of gene expression estimation
    1. RPKM/FPKM values are strongly dependent on the expression level of the highest expressed genes (largest fraction of reads, e.g. rRNA contamination)
    2. Effect of genomic variation



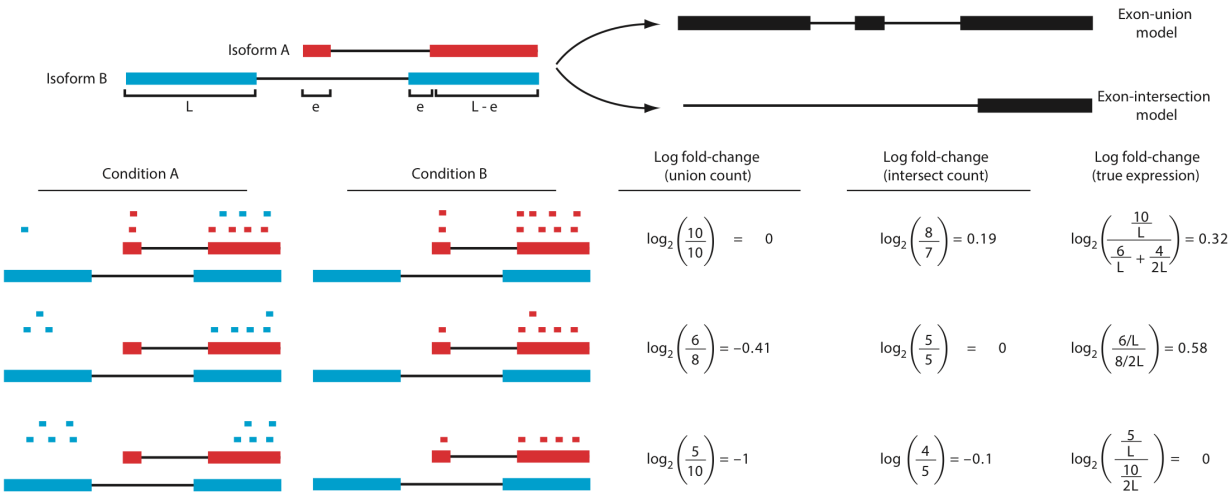
    3. Alternative transcripts/RNA-processing may lead to differential read counts



2 transcript of 1 Gene.
If we simply sum the number of reads on the Gene body, these two would look exactly the same

But we should be interested in which one of these two transcripts are actually active.

4. Caveats of gene expression estimation (Examples)

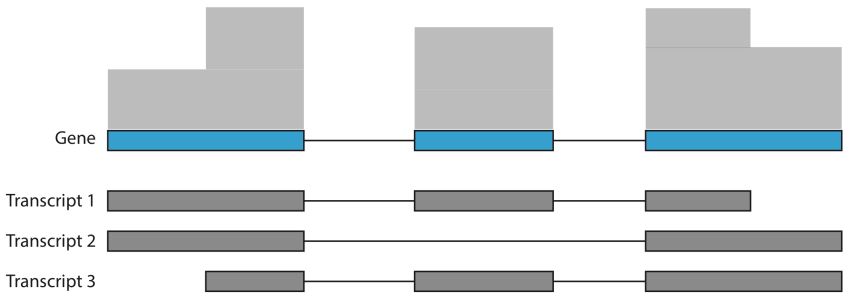


   a) The same gene can contain multiple, partially overlapping transcripts
   b) ignoring the transcript structure can lead to estimation biases (depending on the gene model used for counting Union/ intersection model)

IV. Determination of splice forms

A. Quantification of Transcripts

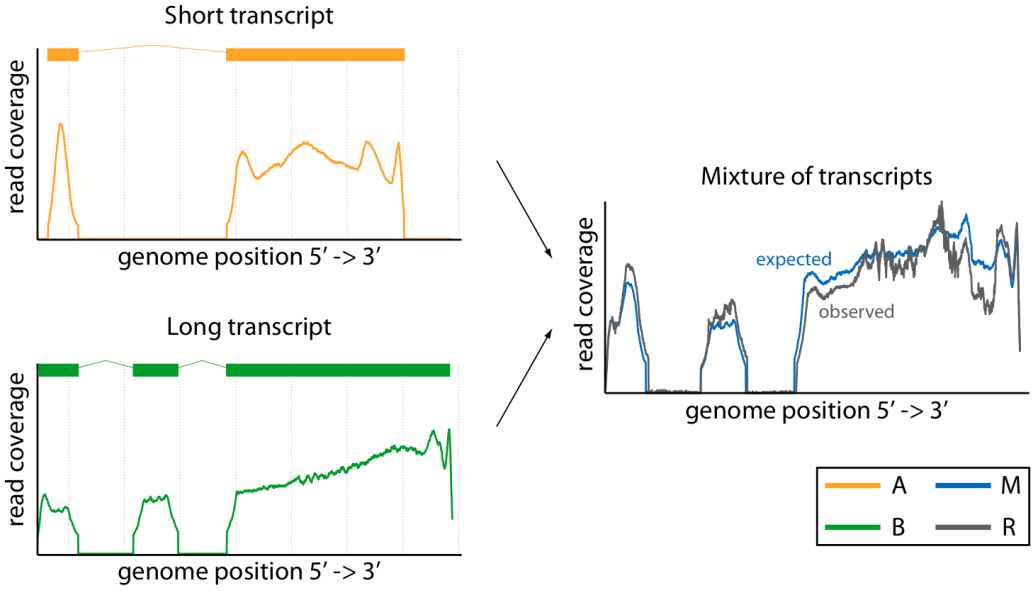
1. An alternative approach is the quantification of single transcripts. Then the total gene expression can be derived as the sum of transcript expressions.
2. Alternatively, one can also use individual transcripts for further analysis. Then different scenarios are possible
   a) the gene expression is different as a whole and transcript expression changes accordingly
   b) the gene expression as a whole remains constant but the expression ratio of transcript forms changes
3. This leads us to the central question of transcript quantification:
   a) Given short read alignments and a set of known transcripts, can we assign each transcript its correct expression value?



4. Solve an optimisation problem:
   a) Optimising weights w_t for each transcript $t = 1,...,T$
   b) Exploiting the additive nature of the read cover
   c) Minimising residual error (e.g., squared error)

$$(w_1, \ldots, w_T) = \underset{w_1,\ldots,w_T \geq 0}{\operatorname{argmin}} \sum_{p \in P} \left(R_p - \sum_{t=1}^{T} w_t D_{t,p} \right)^2$$

   d) With
      (1) P: set of considered genomic positions
      (2) R_p: observed read coverage (number of reads covering pos.p)
      (3) $D_{t,p}$: expected read coverage for transcript t at position p
5. Different approaches share similar ideas with different models on using read count differences and optimisation techniques
   a) Poisson distributions
   b) Absolute differences using a flow-network
   c) Squared differences using QP
   d) (approximate) Negative Binomial distribution
   e) These are all models to estimate the $D_{t,p}$
6. Problems:
   a) Abundances cannot be unambiguously determined with single-end reads (use paired-end reads)
   b) Solution may be unstable: a small change in reads can cause large changes in estimated abundances
   c) Read coverage is not uniform over the transcript
7. rQuant



$M_p = w_A D_{A,p} + w_B D_{B,p} \quad \Rightarrow \quad \min_{w_A, w_B} \sum_p \ell\left(M_p, R_p\right)$

a) Iterative Algorithm

(1) Optimise transcript weights w_t: $\min_{w} \sum_{p \in P} \ell(\sum_{t} w_t D_{t,p}, R_p)$

(a) shown in the low left

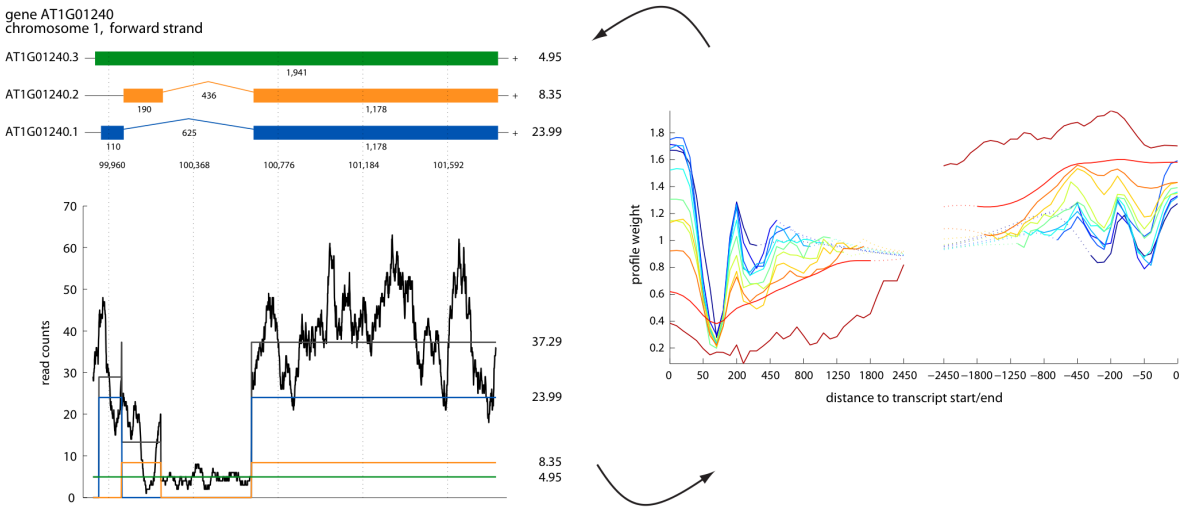
(b) optimise over the weight of each transcript (straight line)

(2) Optimise profile weights $D_{t,p}$: $\min_{w} \sum_{p \in P} \ell(\sum_{t} w_t D_{t,p}, R_p)$

(a) shown in the lower right

(b) optimise over profile weights (not straight line

(3) Repeat 1. and 2. until convergence



b) Verification

(1) Idealised simulation

(a) Simulate reads from annotated transcripts

(b) Use reads' origins as alignments

(c) Use alignments for quantification

(2) More realistic setting

(a) Simulate reads from annotated transcripts

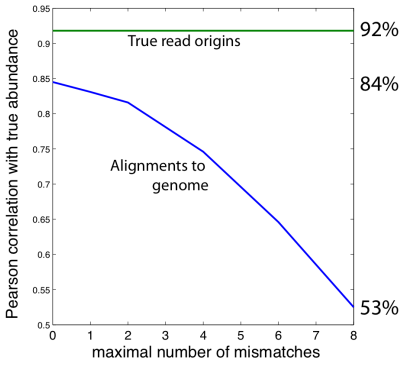
(b) Add errors to reads

(c) Align reads to genome with 0-8 mismatches (many alignments)

(d) Use alignments for quantification

(3) False alignments, multi-mappers etc. lead to weaker result

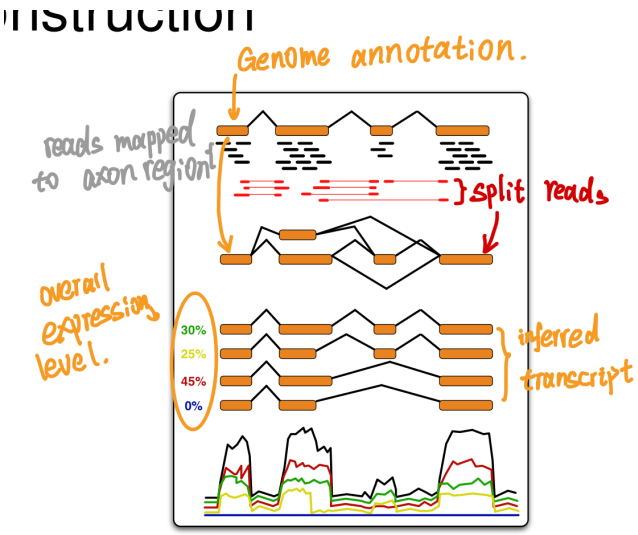
False alignments, multi-mappers etc. lead to weaker results



B. Transcript Identification Strategies
   1. Segmentation strategies: Based on read coverage only
      a) Little assumptions
      b) Find non-coding transcripts
      c) Varies with expression
   2. Trade-off between reads and genomic sequence features
      a) If we use genomic sequence features, we would be able to
         (1) Find a comprehensive set of genes
         (2) More complete due to additional information

V. Alignments & Transcript Reconstruction

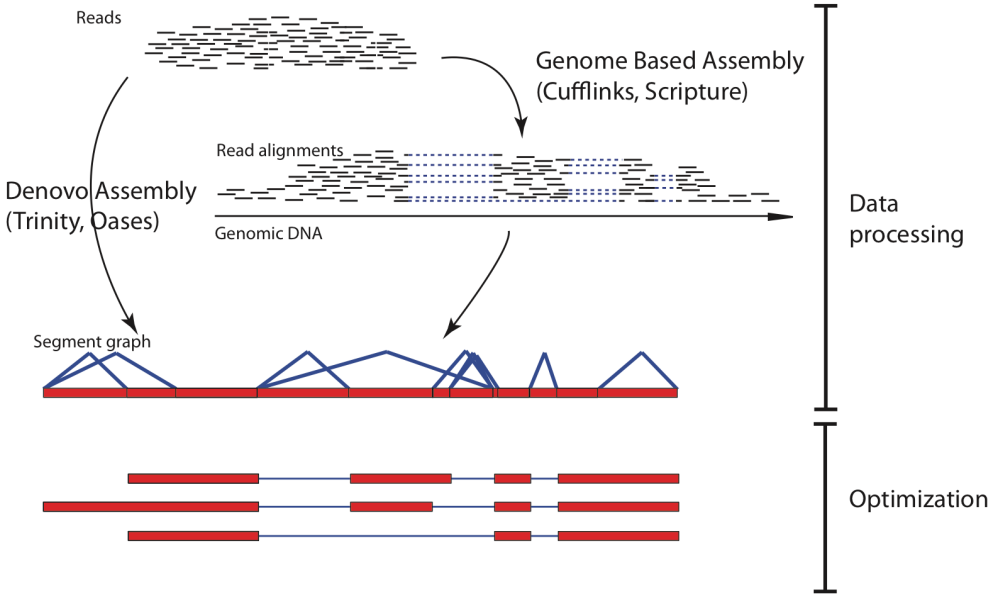
A. Transcript reconstruction:
   1. Identify axons and introns
   2. Relationship in transcripts
      a) e.g. which transcript is the most highly expressed and does this vary based on different conditions



B. Most approaches
   1. Build a transcript/splicing graph
   2. Choose paths as transcripts
   3. Quantify chooses transcripts

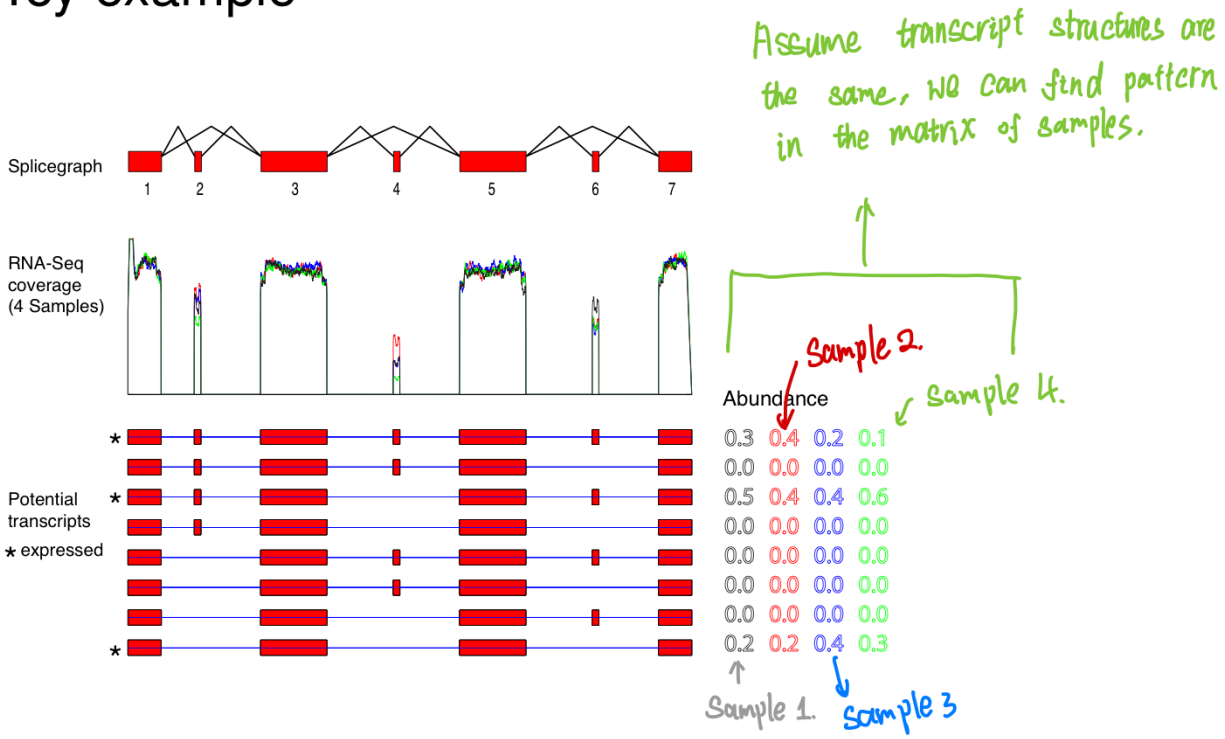
C. Strategy fails if graph is incorrect
   1. We want to be real sure about edges in the splice graph
   2. Filtering

D. Transcript Reconstruction with RNA-Seq



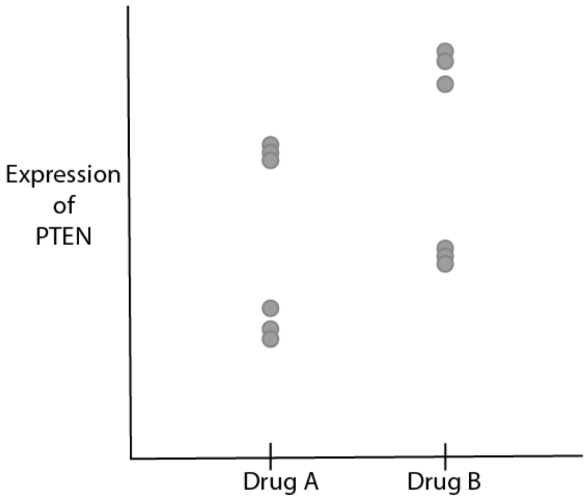
1. If there are 108 possible transcripts, there are 10^{28} possible subsets of transcripts
   a) Not every single possible of subsets of transcripts are true
   b) How do we get around this?
2. For a given splice graph and all of its potential transcripts, we can estimate the abundance of each transcripts based on the sample data. Assuming that all the sample data has the same transcript structures, we can find pattern in the matrix of abundance of samples.

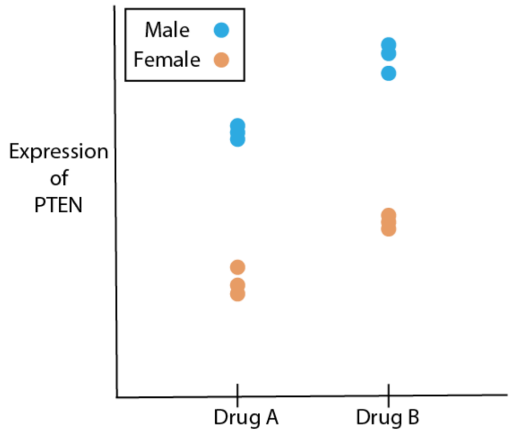
Toy example

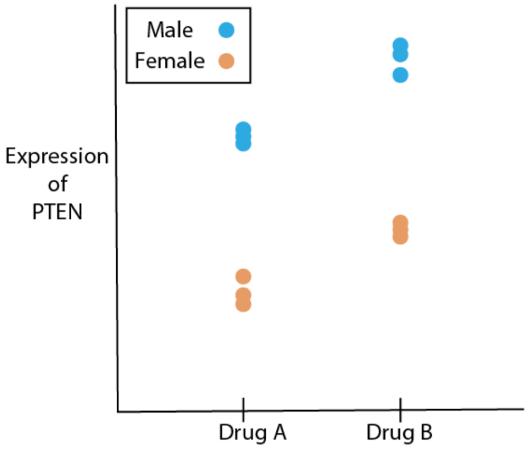


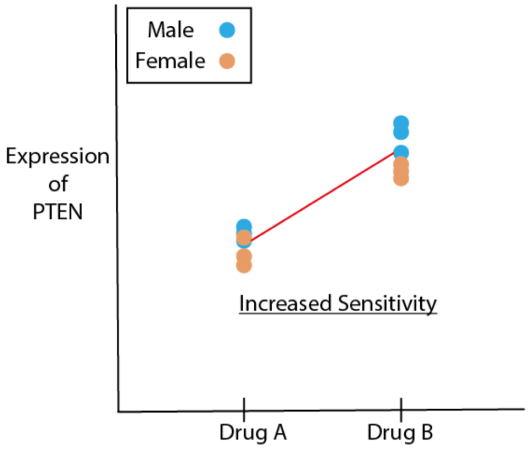
Lecture 6: Analysis of genomic data

I. Motivation
  A. Differential gene expression
    1. Example: Analyse gene expression under two conditions
    2. Condition1: 10M reads in total, Condition 2: 20M reads in total
    3. Gene 1: condition 1 has 10 reads, condition 2 has 20 reads
    4. Gene 2: condition 1 has 20 reads, condition 2 has 20 reads
    5. Gene 3: condition 1 has 20 reads, condition 2 has 10 reads
    6. Gene 4: condition 1 has 20,50,60 reads, condition 2 has 15,40,65 reads
    7. which of the genes are differentially expressed?
      a) Gene 2 & Gene 3 & Gene 4
  B. Differential gene expression and Confounding Factors
    1. Is there a significant effect of drug A vs drug B?









a) From the above figure we can see that for every drug there are two subgroups which can be explained by the gender of participants. However, we are not interested in how gender affects the expression of PTEN. So gender in this case has become a confounding factor. If we remove the confounding factor(by including the confounding factor in our linear model), we can see that there is an increase in the sensitivity.

II. Concepts for linear models

A. Generalised linear models

1. Versatile tool to model a wide range of problems
2. Reasonable results on small sample sizes
3. Scalable to work on large problem settings
4. Probabilistic interpretations
5. Statistical testing of significance

B. Step 1: Simple linear models

$$Y = \beta_0 + \beta_1 x + \epsilon$$

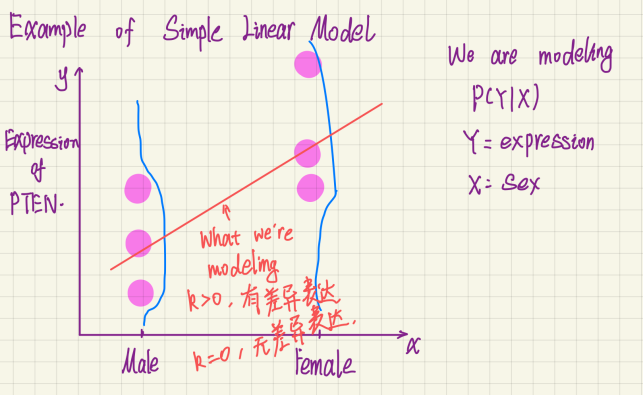
$$\epsilon_i \sim_{iid} N(0,\sigma^2)$$

In matrix notation:

$$Y = X\beta + \epsilon$$

Example:

Y = Gene expression of PTEN across n samples



$$x = \begin{cases} 1 & \text{if Male} \\ 0 & \text{if Female} \end{cases}$$

C. Simple Linear Model: least squares

1. Simplest case: Find an offset (β_0) and a slope (β_1), which best fits all (Y_i, x_i) pairs. In other words, minimise the error

$$Q = \sum_i^n (Y_i - \hat{Y})^2$$

$$= \sum_i^n (Y_i - (\beta_0 + \beta_1 x_i))^2$$

Expand equation, basic calculus, we can get

$$\beta_1 = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_i^n (x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

2. In matrix notation we have:

$$Y = X\beta + \epsilon$$

$$\Rightarrow \epsilon = Y - X\hat{\beta}$$

Sum of squared errors:

$$\begin{aligned}\epsilon^T \epsilon &= (Y - X\hat{\beta})^T (Y - X\hat{\beta}) \\ &= Y^T Y - Y^T (X\hat{\beta}) - (X\hat{\beta})^T Y + (X\hat{\beta})^T (X\hat{\beta}) \\ &= Y^T Y - 2\hat{\beta}^T X^T Y + \hat{\beta}^T X^T X \hat{\beta}\end{aligned}$$

Take derivative to find minimum:

$$\begin{aligned}0 - 2X^T y + 2X^T X\hat{\beta} &= 0 \\ X^T y &= X^T X\hat{\beta} \\ \hat{\beta} &= (X^T X)^{-1} X^T Y\end{aligned}$$

D. Simple linear model - probabilistic frame work

1. Density function of the normal distribution

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

Since

$$P(Y \mid X = x) = \prod_i^n p(y_i \mid x_i) = \prod_i^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(y_i - (\beta_0 + \beta_1 x_i))^2}{2\sigma^2}}$$

The log likelihood is

$$\begin{aligned}\log \prod_i^n p(y_i \mid x_i) &= \sum_i^n \log(p(y_I \mid x_i) \\ &= -\frac{n}{2}\log(2\pi) - n\log(2\sigma^2) - \frac{1}{2\sigma^2}\sum_i^n (y_i - (\beta_0 + \beta_1 x_i))^2\end{aligned}$$

After standard calculus (derivative on parameter and setting to zero):

$$\beta_1 = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_i^n (x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\sigma^2 = \frac{1}{n}\sum_1^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

E. Simple linear model - simple example
   1. Does height depend on shoe size or specific mutation?
   2. Assume we want to explain height by a specific mutation

$$\begin{pmatrix}187\\190\\160\\170\end{pmatrix} = \beta_0 + \beta_1 \begin{pmatrix}male\\male\\female\\female\end{pmatrix} + \beta_2 \begin{pmatrix}not-mutated\\mutated\\not-mutated\\mutated\end{pmatrix} + \epsilon$$

   3. Parameter estimate quantify the size of the effect of the factor (β_2) on explaining Y.
   4. Null hypothesis: Assume, mutation has no effect on height ($\beta_2 = 0$)
   5. Alternate hypothesis: Mutation has not 'no effect' on height ($\beta_2 \neq 0$)
   6. Why do we include gender in this example?
      a) Because Gender can be an important confounding factor in this problem since it is generally assumed that male is taller than female.
      b) If there are genes that are specific to female, if we do not include the gender in our model, we would likely to be tricked into that the gene can affect height since male is generally taller than female when the truth is simply male do not have the gene due to them being male.
   7. Goal: What is the probability, given that we have a model assuming the mutation has no effect on height, to observe the estimated $\hat{\beta}_2$ or more extreme one?

F. Simple linear model - Basic concepts
   1. p-value (informal): Probability that under a specific model, a specific statistical summary is equal or more extreme than its observed value in the sample data.
   2. p-value incorporate effect-size, sample size and variance
   3. p-values are easily misinterpreted
   4. p-value common misconceptions
      a) p-value do not make a statement on the truth of the null hypothesis
      b) p-value also do not measure support for the alternative hypothesis
      c) p-value threshold of 0.05 is only a convention
      d) p-value close to 0.05 can have error rates between 25 - 50%!

G. Simple linear model - testing
   1. Let's assume a simple linear model from above

$$Y = \beta_0 + \beta_1 x + \epsilon \text{ with } \epsilon_i \sim_{iid} N(0,\sigma^2)$$

$$H_0 : \beta_1 = 0 \text{ and } H_1 : \beta_1 \neq 0$$

2. Likelihood ratio test
   a) Under assumption H_0, one can show that statistic variable D

$$D = -2\log\left(\frac{L(H_0)}{L(H_a)}\right) = 2*(\log(L(H_a)) - \log(L(H_0))) \sim \chi^2_{n-k}$$

   b) $n-k$ is the degree of freedom, and can be calculated as the difference between the number of parameters in the general and in the nested model. (see computational biology lecture 7)
   c) Recall the likelihood above, we have

$$H_a\text{: } L(\hat{\beta}_0, \hat{\beta}_1, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\hat{\sigma}^2) - \frac{n\hat{\sigma}^2}{2\sigma^2} = -\frac{n}{2}(1 + \log 2\pi + \log\hat{\sigma}^2)$$

$$H_0\text{: } L(\hat{\beta}_0, \hat{\beta}_1 = 0, \hat{\sigma}^2) = -\frac{n}{2}(1 + \log 2\pi + \log(\sum_i^n (Y_i - \bar{Y})^2))$$

   d) Plugging in the likelihood in D:

$$D = \frac{n}{2}\log\left(\frac{\sum_i^n (Y_i - \bar{Y})^2}{\frac{1}{n}\sum_1^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2}\right)$$

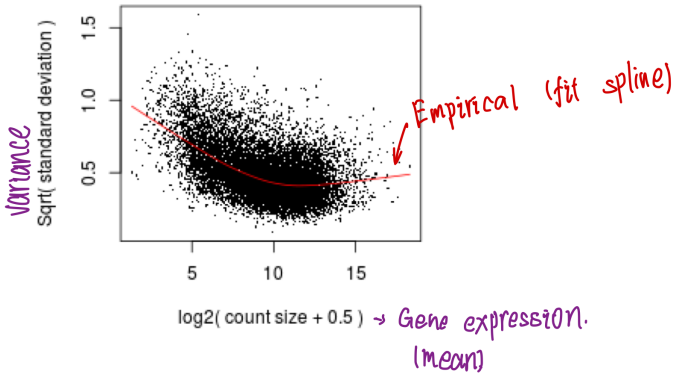
$$= \frac{n}{2}\log\left(\frac{s_y^2}{\hat{\sigma}^2}\right)$$

   e) Under H_0, one can show that

$$\frac{n}{2}\log\left(\frac{s_y^2}{\hat{\sigma}^2}\right) \sim \chi^2_1$$

H. Simple linear model - assumptions
   1. Normality of residuals
   2. Homoscedasticity (Variance does not depend on X)
   3. Independence of residuals
   4. Linearity and additivity

I. Simple linear model - Mean variance relationships



› be made:

1. The above figure describes a relationship between the mean and the variance of the genes in real time, which does not always satisfy the 4 conditions mentioned above, so we can
   a) Decouple mean from variance
   b) Empirical (fit spline) or theoretical (variance stabilising transforms)
      (1) Instead of fitting on the raw data. we can fit on the square root of the raw data, the log transform of the raw data and so on …
2. Instead of changing the data to fit the above strong assumption we can simply generalise the assumption of simple linear model and obtain the generalised linear model

J. Generalised linear model - probability distributions
1. All distributions are modelled for specific use-cases! What do the following distribution have in common?
   a) Gaussian; Multinomial; Bernoulli; Binomial;Gamma; Poisson; Exponential;Beta
   b) ⇒ They are all members of the exponential family
   c) we can write down the general form of PDF of exponential family in

$$p(x\,|\,\eta) = h(x)\exp(\eta^T t(x) - a(\eta))$$
$$a(\eta) = \log \int h(x)\exp(\eta^T t(x))dx$$

K. Generalised linear model - Logistic regression example
1. Binary output, so intuitively a Bernoulli distribution

$$p(x\,|\,\pi) = \pi^x(1-\pi)^{1-x}$$

2. We can rewrite the Bernoulli distribution as a standard form of the exponential family like so

$$t(x) = \log\frac{\pi}{1-\pi} \quad \eta = \log\frac{\pi}{1-\pi} \quad h(x) = 1$$
$$a(\eta) = -\log(1-\pi)$$

This can be rewritten as

$$\pi = \frac{1}{1+e^{-\eta}} \quad a(\eta) = \log(1+e^{\eta})$$
$$p(x\,|\,\eta) = \sigma(-\eta)e^{-\eta x}$$

3. The great thing about choosing the exponential family is that for every generalised linear model, we can just plot the data, choose the distribution and write it as a standard form of exponential family then plugin the result to an existing algorithm framework to calculate the likelihood etc.
   a) one algorithm fits all.

L. Generalised linear model - Parameter estimation

1. Likelihood of generalised linear model

$$I(\theta) = \log(\prod_{n}^{N} h(y_n)\exp(\eta_n y_n - a(\eta_n)))$$

$$= \sum_{n}^{N} \log(h(y_n)) + \sum_{n}^{N} (\eta_n y_n - a(\eta_n))$$

2. Take derivative to get gradient of log-likelihood function

M. Modeling count data

1. Often genomic data is count data
2. We are interested in the question: What is the probability to observe x reads mapping onto small region of the genome?
   a) Probability that a read maps onto a well defined region (gene j) is Bernoulli distributed

$$f(x) = \begin{cases} p & \text{if read maps onto gene } j \\ 1-p & \text{if read does not map to gene } j \end{cases}$$

   b) Assuming reads are iid, what is the probability that k out of n reads map onto gene j (Binomial)

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$E(k) = np \quad Var(k) = p(1-p)/n$$

3. Binomial vs. Poisson Distribution
   a) Let's assume $\lambda = np \rightarrow p = \lambda/n$
   b) Substitute p in Binomial Distribution

$$\lim_{n\to\infty} P(X=k) = \lim \frac{n!}{k!(n-k)!} \left(\frac{\lambda}{n}\right)^k \left(1-\frac{\lambda}{n}\right)^{n-k}$$

$$= \left(\frac{\lambda^k}{k!}\right) \lim \frac{n!}{(n-k)!} (\frac{1}{n^k} (1-\frac{\lambda}{n})^n (1-\frac{\lambda}{n})^{-k}$$

$$\lim \frac{n!}{(n-k)!} (\frac{1}{n^k}) = \lim \frac{n}{n} \frac{n-1}{n} \frac{n-2}{n} \dots \frac{n-k+1}{n} = 1$$

$$\lim (1-\frac{\lambda}{n})^n = \exp(-\lambda)$$

plugging in we have

$$\lim_{n\to\infty} P(X=k) = \left(\frac{\lambda^k}{k!}\right) e^{-\lambda} \sim \text{Poisson Distribution}$$

c) This means as we sequence more and more read, we can model the reads using the Poisson distribution.

4. Features of Poisson distribution

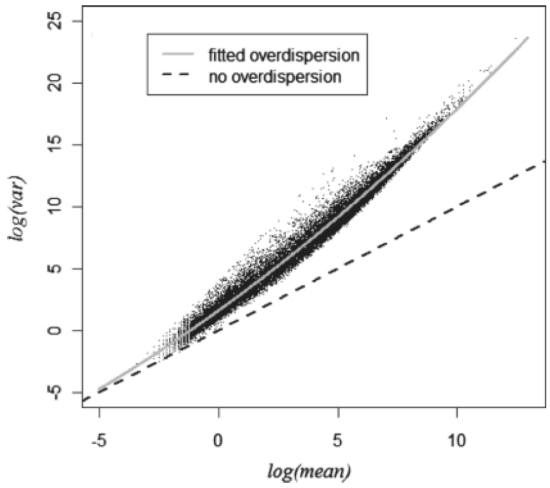
a) $P(X) = \left(\frac{\lambda^k}{k!}\right) e^{-\lambda}$

b) Mean = Variance = λ

(1) This means the data is no longer Homoscedastic meaning mean and variance are no longer independent. This is fine because Homoscedasticity is not a requirement in generalised linear model

(2) However, this do imply that the mean and variance are the same and this is not usually the case.

c) Overdispersion: $Var(Y) > E(Y)$



d)In the figure on the left, we can see that theoretically, the mean and the variance should be on the dashed line if there is no over dispersion. However, that is not the case and we see the variance is usually higher than the mean

e)Variance stabilising transformations

f)Directly lean the correlation between mean and variance and embed them into the model

N. Overdispersion

1. Even though we assume the variance and the mean should be the same, they are often in the real world not. and this is called the problem of overdispersion.
2. Potential causes of overdispersion:
   a) Excess zeros (many many zeros)
   b) Correlation of samples
   c) Grouping in samples → confounding factors
   d) Unobserved variables
3. Overdispersion does not change parameter estimates but increase their errors and thus effects testing

O. Addressing overdispersion

1. Variance Stabilising Transform

a) A transformation on some data X, such that Var(X) is independent of E(X)

b) E.g. If x is Poisson with $E(X) = Var(x) = \lambda$, then $y = \sqrt{x}$ should make var(x) independent of E(x)

2. Model overdispersion directly

a) Assume $\phi E(X) = Var(X)$ (Quasi-Likelihood model) then estimate $\hat{\phi} = \frac{1}{n-p} \sum_i \frac{(X_i - \hat{\mu}_i)^2}{Var(X_i)}$

P. Model overdispersion as mixture

1. We started to consider generalised linear model because there is a correlation of mean and variance which simple linear model does not allow.
2. Then we have this overdispersion which still requires variance stabilising transform or modelling, it's like we are in square one.
3. So instead of modelling the reads using Poisson distribution we can slightly change the distribution to Poisson-Gamma Distribution
4. Let's assume that $Y \mid \theta$ follows a Poisson with parameter μ_θ

   Intuitively θ could control something like higher or lower read count than expected
5. Assume θ follows a Gamma Distribution with parameter α and β where $E(\theta) = \frac{\alpha}{\beta}$ and $Var(\theta) = \frac{\alpha}{\beta^2}$ so $\alpha = \beta = \frac{1}{\sigma^2}$
6. The marginal is $P(Y) = \frac{(\alpha + y)}{y!(\alpha)} \frac{\beta^\alpha \mu^y}{(\mu + \beta)^{\alpha+y}}$ (Poisson-Gamma Distribution)

   a) Poisson-Gamma Distribution has some interesting feature with negative binomial!
7. Now with $\alpha = \beta = \frac{1}{\sigma^2}$ it follows that

   $E(Y) = \mu$ and $Var(Y) = \mu(1 + \sigma^2 \mu)$

Q. DESeq (An example of overdispersion as mixture)

1. DESeq makes similar assumption for read counts

   a) $K_{ij} \mid R_{ij}$ is Poisson with rate $s_j r_{ij}$ for gene i and sample j

   b) R_{ij} has mean q_{ip} and variance v_{ip}

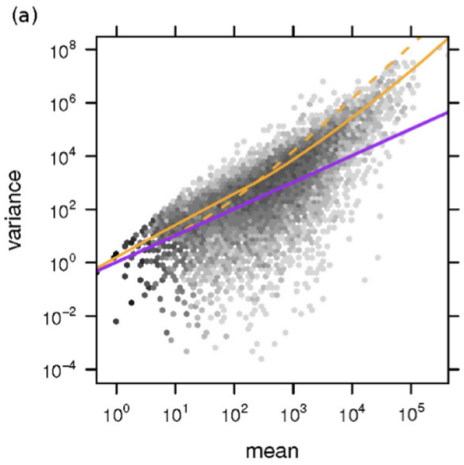
   c) Thus the marginal $K_{ij} \sim NB(\mu_{ij}, \sigma_{ij}^2)$ has mean μ_{ij} and $\sigma_{ij}^2 = \mu_{ij} + s_j^2 v_{i,p(j)}$

      (1) we can see that the variance is a function of the mean and this is exactly the case with data with overdispersion and this

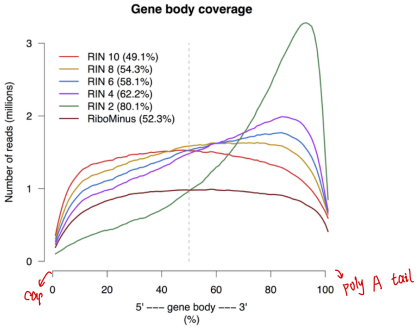
is why negative binomial or poisson gamma distribution is used for count data!

2. Three additional modelling assumption
   a) μ_{ij} is expected value of observed counts for gene i in condition j p(j) multiplied by a size factor s_j (geometric mean) to account for library size
   b) $\sigma_{ij}^2 = \mu_{ij} + s_j^2 v_{i,p(j)}$ or 'shot-noise' + 'variance' allowing to model higher variances with increasing mean expression
   c) $v_{i,p(j)} = v_p(q_{i,p(j)})$. In other words, the variance is estimated based on a function learned on all genes with similar expression.
3. Result

- Purple:Poisson fit
- Yellow:NB fit
- Yellow (dotted): Normal with variance stabilizing transform



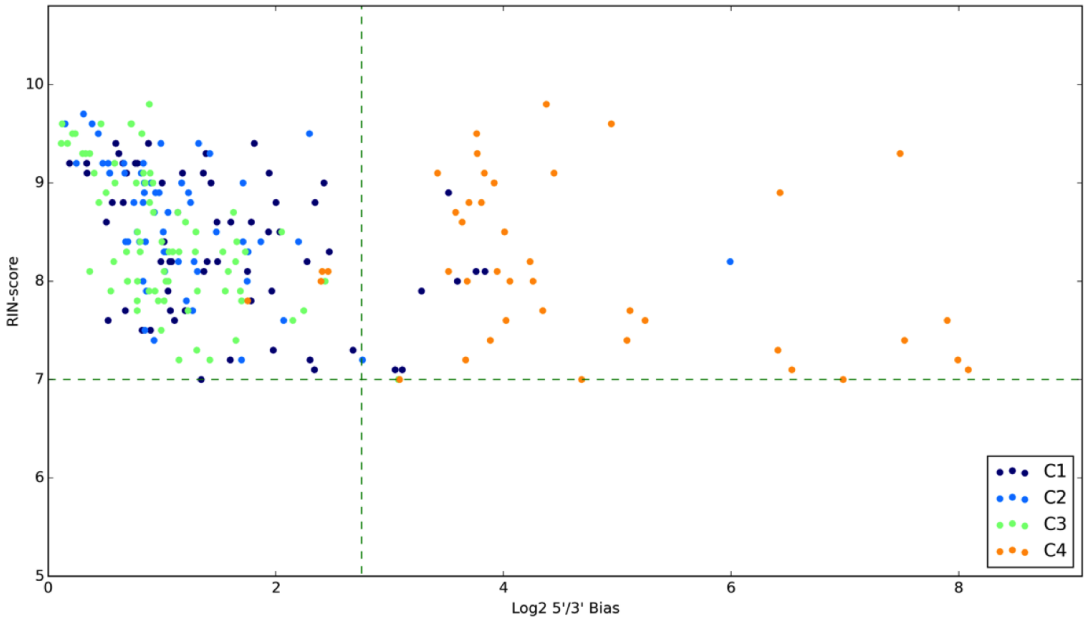
III. Confounding factors of genomic data
   A. RNA Degradation
      1. Certain genes will have lower read count due to RNA Degradation



2. The distribution of reads in the gene body is not identical and it depends hugely on what scoring system you are using (RIN 10, RIN 8, etc.) So it is important that you take these things into account.

B. Batch Correction

1. If two people are doing sequencing one day and one person happen to be better at sequencing that day than the other, this can also cause confounding factor



2. Even though we have high RIN Score, we can still have a lot of bias in the sequence. For example if we are to compare C4 and C1, we are pretty likely to find lots of differential expressed genes, but this is simply due to sample biases or batch effects.
3. Batch effects and experimental biases can lead to false conclusion
4. Worst cased: False treatment decisions

IV. Poisson-Gamma Distribution and Negative Binomial Distribution

$$pmf(x;\alpha,\beta) = \int_0^\infty Pois(x;\lambda)\cdot Gamma(\lambda;\alpha,\beta)d\lambda$$

$$= \int_0^\infty \frac{\lambda^x}{x!}e^{-\lambda}\cdot\frac{\lambda^{\alpha-1}e^{-\frac{\lambda}{\beta}}}{\Gamma(\alpha)\beta^\alpha}d\lambda$$

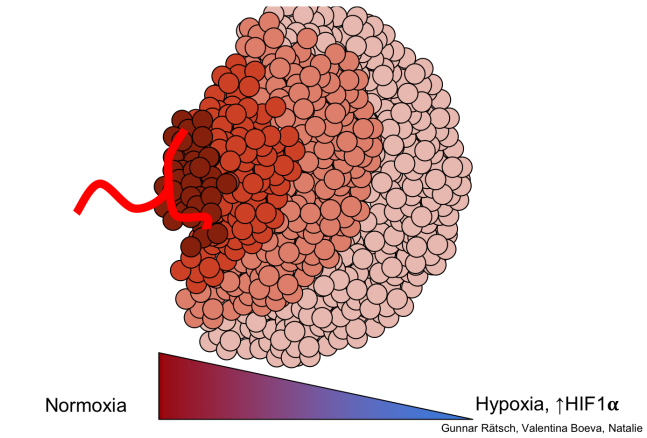
$$= \left(\frac{a+x-1}{x}\right)\left(\frac{1}{\beta+1}\right)^\alpha\left(1-\frac{1}{\beta+1}\right)^x$$

$$= \left(\frac{r+x-1}{x}\right)p^r(1-p)^x$$

$$r=\alpha; p=\left(\frac{1}{\beta+1}\right)$$

Lecture 7: Single-Cell Transcriptomics

I. Motivation
   A. Case study 1: Hypoxia of a cancer cell
      1. Hypoxia: The cell status when cell does not have enough oxygen, gene $HIF1_{\alpha}$ is expressed under this scenario. Hypoxia is important for a cancer patient because it is related to the transferability of the cancer cell.
      2. Image we are looking at a tumour which are connected by only one blood vessel. As it grows larger and larger and the cell grows further and further away from the blood vessel, we can see that the amount of oxygen the cell gets is less and less until we hit a point that the cell is Hypoxia as shown below.

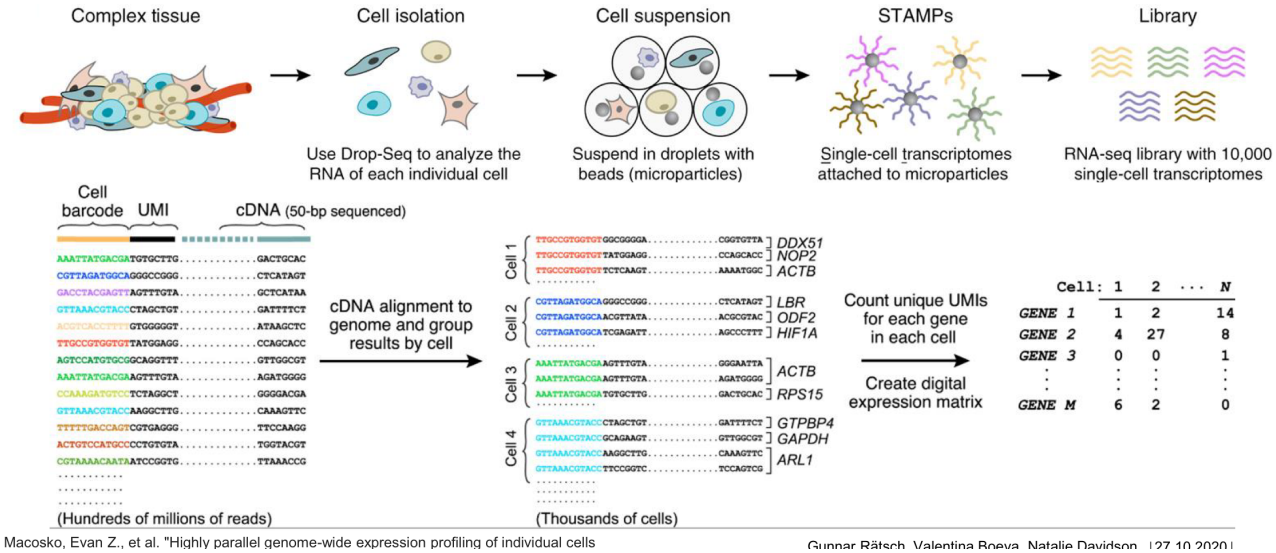


      3.If we do a traditional Bulk RNA-Seq, it will only get the mean expression of $HIF1_{\alpha}$ across the entire sample
      4.it can no longer tell if there exists a hypoxic region

   B. Case study 2: Multiple Cell Types
      1. When we obtain a sample of a tumour from a patient, we are not looking at only cancer cells, we are also exposed with different kinds of healthy cells like T cell, B cell etc. If we want to figure out the heterogeneity of the tumour and / or identify rare tutor cell populations, we can not do a bulk RNA-Seq, we must do it to a single cells' resolution.

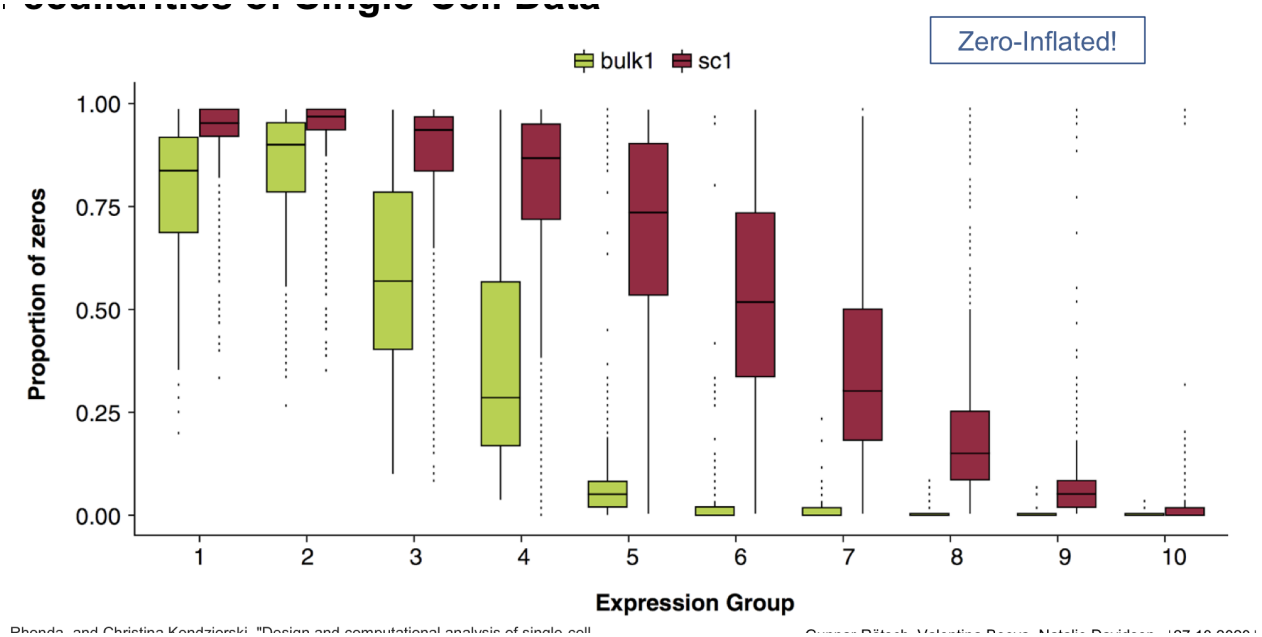
II. scRNA-Seq technology
   A. Drop-Seq
      1. Shown below are the workflow of Drop-Seq, the idea of Drop-Seq is that we separate each cell into a little droplets with beads, and then we attach each molecule of each droplets with an identifier which has three components:
         a) PCR handle: This is usually the same for every cell(every droplets)
         b) Cell barcode: same cell barcode means the same cell
         c) UMI: Unique molecule identifier, after PCR, we use UMI to find the same molecule in the cell.

2. The result of the Drop-Seq is a matrix. The 0 inside the matrix does not necessarily mean that gene is silent in that cell. It just means that the expression outcome of the gene was not captured by Drop-Seq. The sensitivity of catching an expression outcome is 10%.



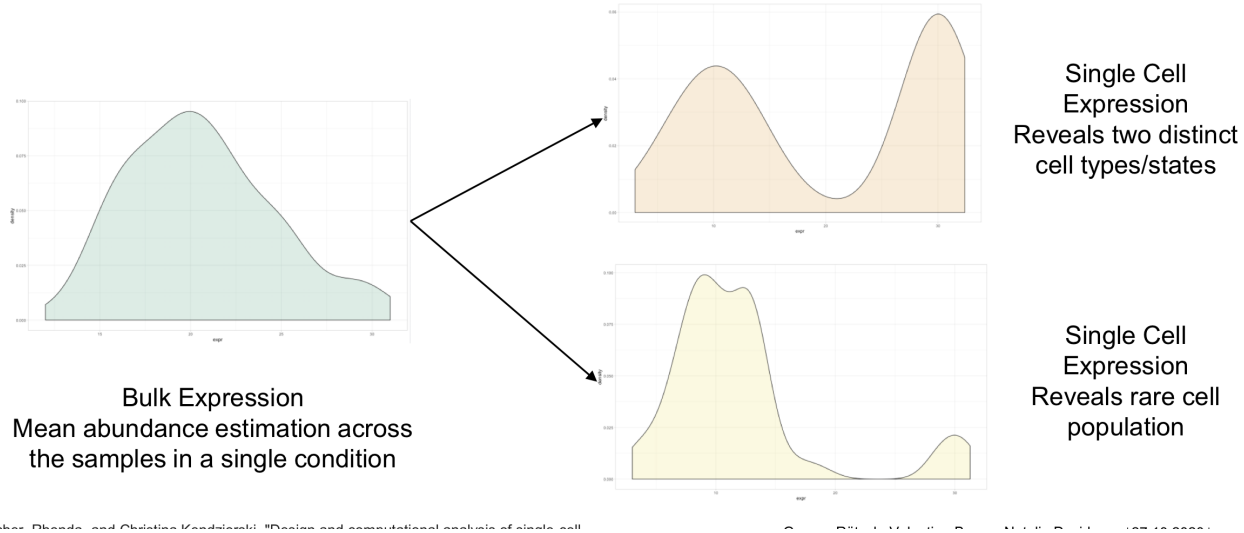
Macosko, Evan Z., et al. "Highly parallel genome-wide expression profiling of individual cells

3. This 0 inflated problem should be kept in mind when we do the analysis

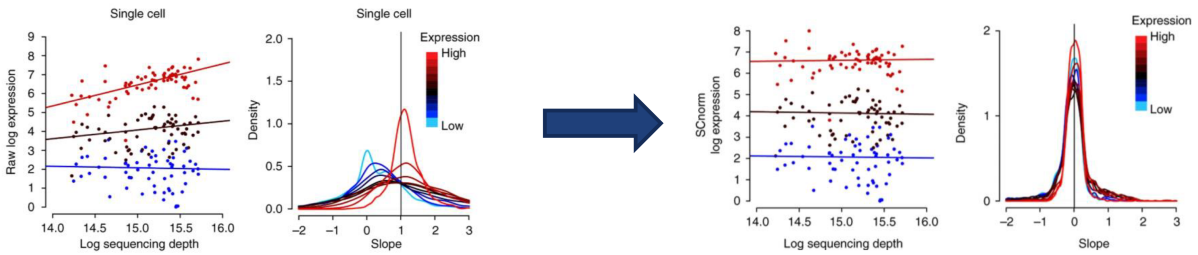


4. Shown above is a graph to show the zero inflated problem of single cell RNA-Seq. This means that even for highly expressed genes like expression group 6 or 7, we can still observe a lot of 0s while the traditional bulk seq does not have 0s.
5. Also, the variance of single cell data is larger compared to bulk RNA-Seq. This may due to

a) In bulk seq, all of the rare cell types are grouped together and generating an averaging effect of the expression. e.g. if one gene is not expressed at all in one cell while in another cell it is expressed a lot. In bulk seq this looks completely normal while single cell technology can capture this difference and this difference contribute to the increased variance of single cell

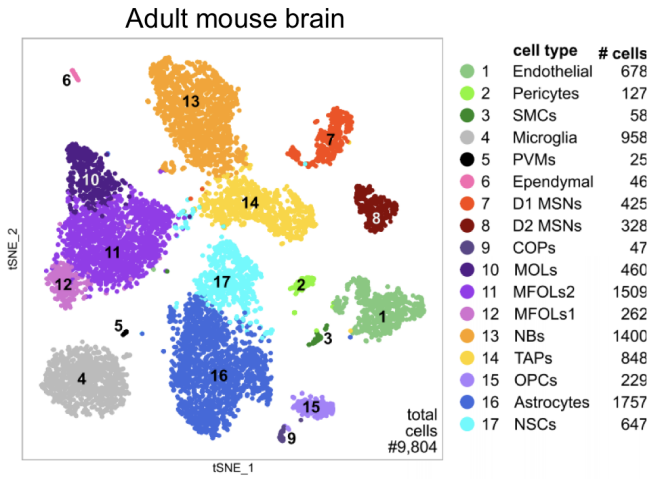


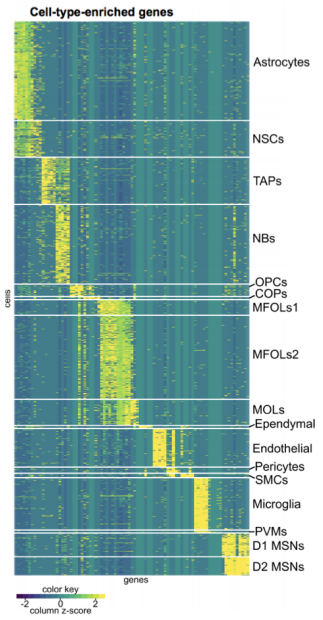
6. Library Size Normalization
   a) In the bulk RNA seq example, some samples will be sequenced at a deeper depth than other samples. The slope of read depth difference is the same for low expression genes, medium expressed genes and for highly expressed genes. This means that we only need a global correction factor to normalise the library size this is often called RPKM or FPKM
   b) In the single cell technology however, this is not the case and we can not use one global correction factor.
   c) scNorm
      (1) scNorm uses quantile regression to estimate the dependence of transcript expression on sequencing depth for every gene
      (2) Genes with similar dependence are the ngrouped, and a second quantile regression is used to estimate scale factors within each group
      (3) Within-group adjustment for sequencing depth is then performed using the estimated scale factors to provide normalised estimates of expression.



III. Analysis complications beyond bulk RNA-Seq
  A. Single Cell Visualisations
    1. Understand what cell types exist in our sample
      a) Do we have new cell types?
      b) How heterogeneous is our sample?
    2. Can we distinguish treated cells from non-treated cells?
      a) Batch correction
      b) Cell-type effects
    3. Example of visualisation of adult mouse brain
      a) Heat map (lower right)
        (1) this is the most straight forward way of visualisation
        (2) After single celling the cells in the brain of an adult cell, cells with similar expression levels are grouped together using hierarchical clustering and genes that are highly expressed are. marked in yellow.

Visualization of cell types





    4. Common methods
      a) PCA
        (1) Orthogonal linear transformation
        (2) Designed such that the first component explains the largest variance. Every component afterward explains less variance, and is uncorrelated with the previous component.
        (3) Given n points in R^p, principal components analysis consists of choosing a dimension $k < p$ and then finding the affine space of dimension k with the property that the squared distance of the points to their orthogonal projection onto the space is minimised

(4) PCA often can not capture non-linear relationships within the data.

b) tSNE: t-distributed Stochastic Neighbour Embedding

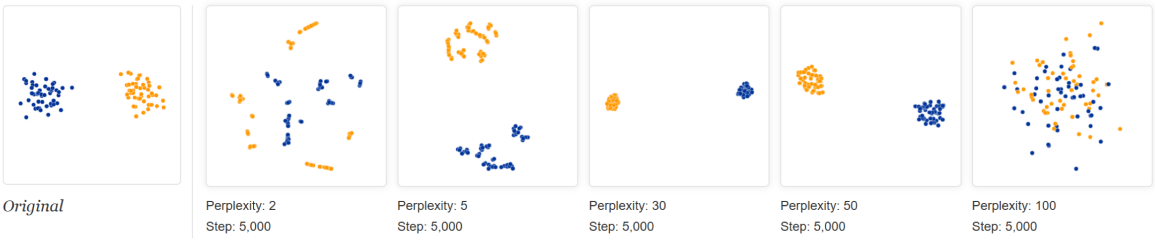
(1) tSNE: nonlinear dimensionality reduction technique, converts similarities between data points to joint probabilities and tries to minimise the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high dimensional data.

(2) tSNE has a cost function that is not convex. with different initialisations we can get different results.

(3) tSNE models the similarity of the higher dimensional data using normal distribution and the σ of the normal distribution is related to one user-specified perplexity parameter. Then it uses t-distribution with one degree of freedom to model the similarity on low-dimensional data. Then it tries to minimise the Kullback-Leibler divergence between the joint probabilities.

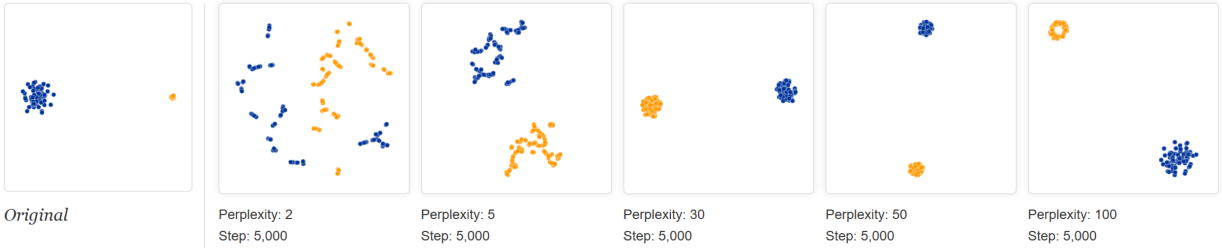
(4) Different choice of the perplexity parameter would have different result on tSNE. the perplexity parameter tells the tSNE how far it can sense the datapoints near it.

- tSNE method's the most important parameter:
  - **Perplexity**, scikit-learn recommended range: [5, 50], default: 30

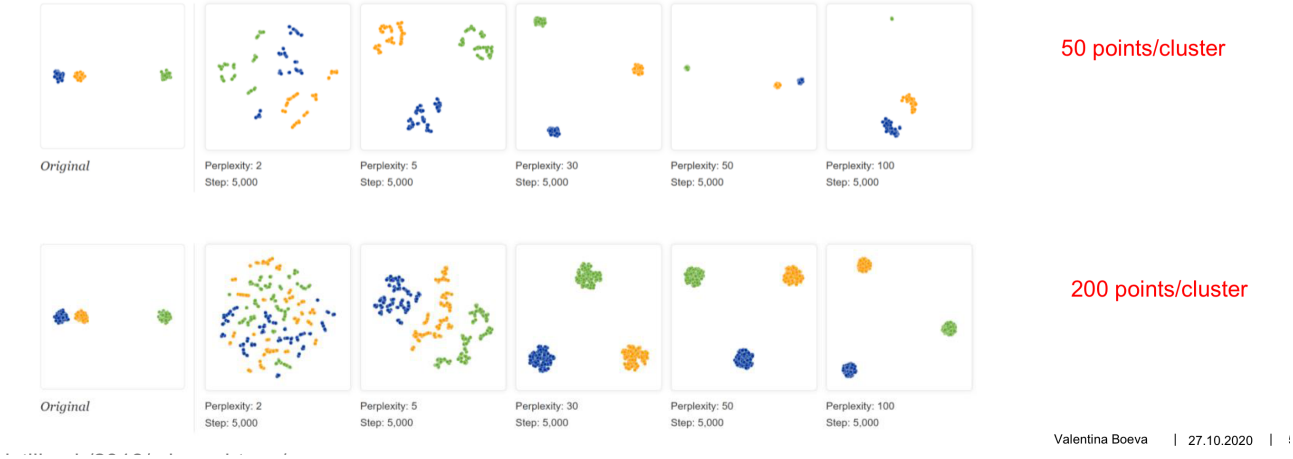


(5) tSNE particularities

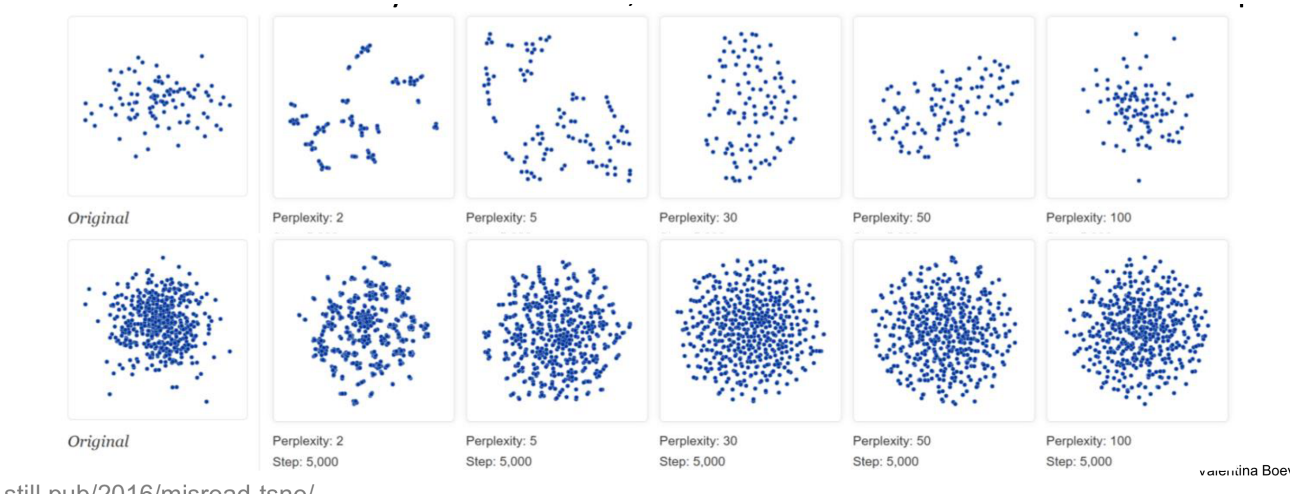
(a) Cluster sizes in a t-SNE plot mean nothing



(b) Distances between clusters might not mean anything



(c) Random noise does not always look random, and sometimes one can see some shapes.



c) UMAP: Uniform Manifold Approximation and Projection

(1) UMAP also measures the distance in the higher dimension using a joint probability. However, instead of using a normal distribution, any distance can be plugged into UMAP, not only euclidean distances.

(2) For the lower space, UMAP uses a family of curves similar to student t-distribution but not exactly the same

(3) Instead of minimising the Kullback-Leibler divergence, UMAP minimises the cross-entropy between two distributions.

(4) Keep in mind that distributions are not normalised - is one of the steps that makes UMAP much faster than t-SNE

(5) Properties of UMAP

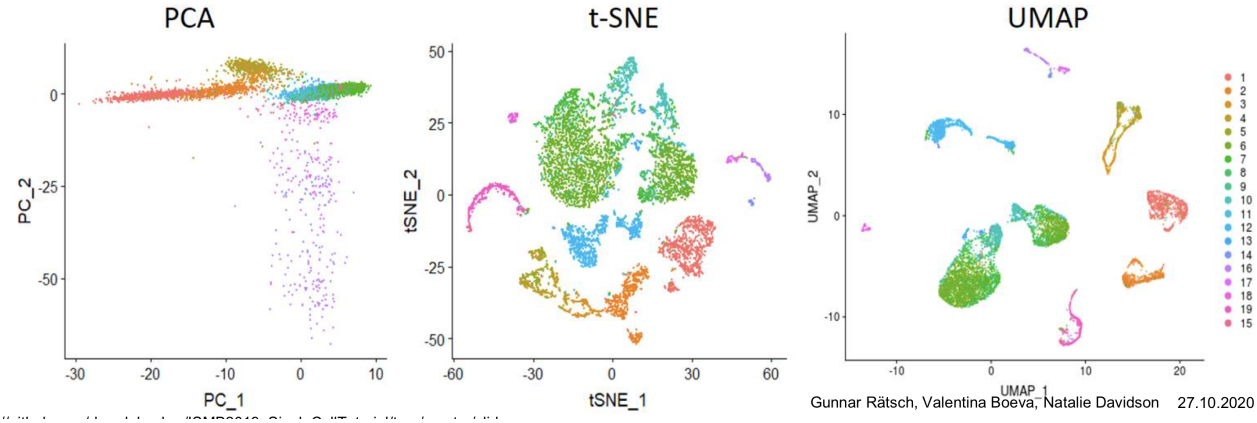
(a) Much faster

(b) Not limited to the first 2-3 dimensions like t-SNE

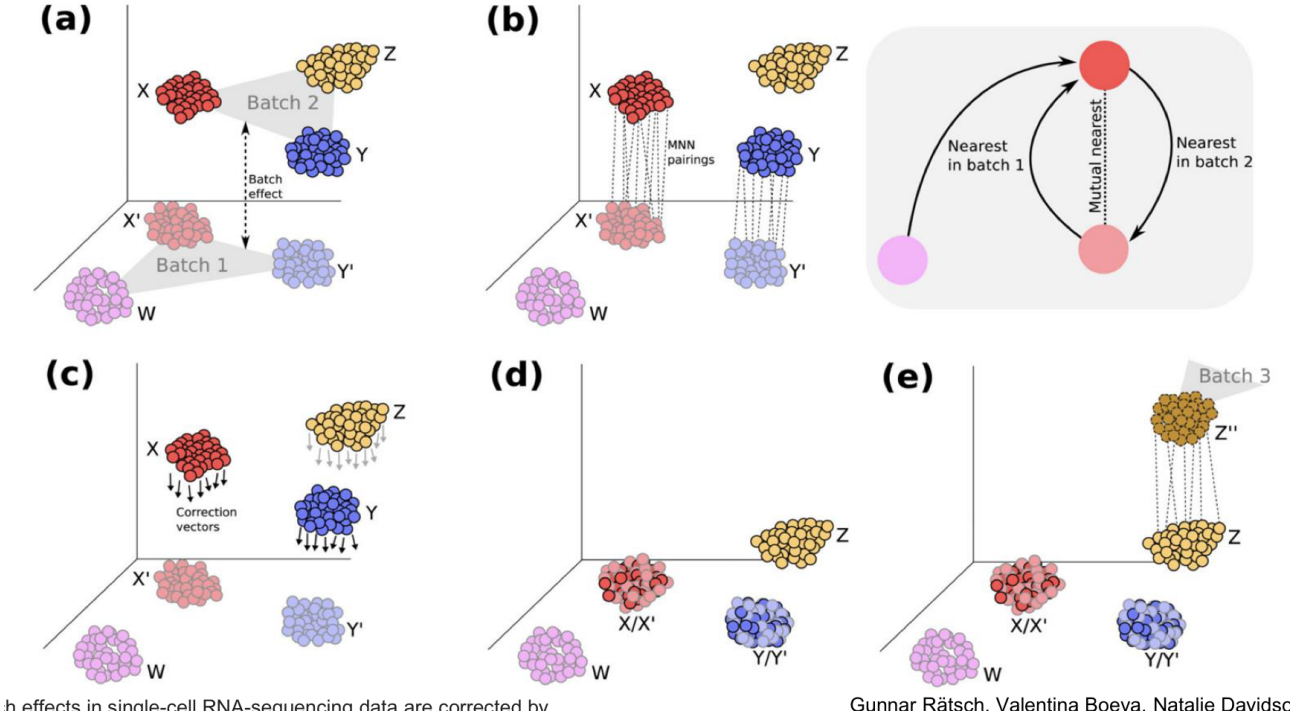
i) not only useful for visualisation but also useful for dimension reduction

(6) Better preserves global structure
(7) Uses the number of nearest neighbours instead of perplexity

d) Most important parameter of UMAP:
(1) min_dist, Controls how tightly the embedding is allowed compress points together. Larger values ensure embedded points are more evenly distributed, while smaller values allow the algorithm to optimise more accurately with regard to local structure. Sensible values are in the range 0.001 to 0.5, with 0.1 being a reasonable default
(2) n_neighbors, determines the number of neighbouring points used in local approximations of manifold structure. Larger values will result in more global structure being preserved at the loss of detailed local structure. In general this parameter should often be in the range 5 to 50, with a choice of 10 to 15 being a sensible default.

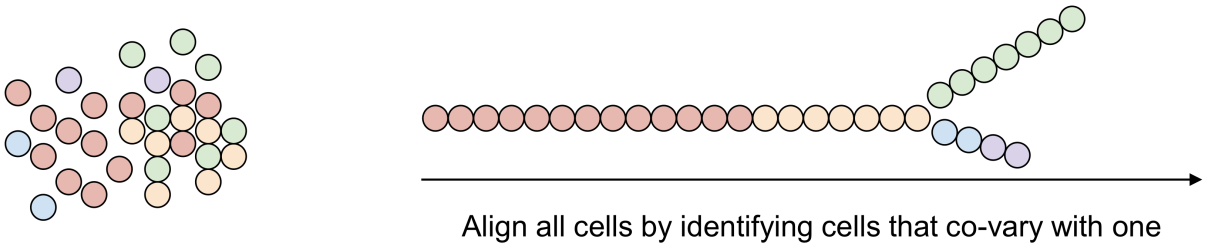


B. Single Cell Data Alignment
1. How do we combing data from different experiments?
a) Experiments can have different cell type populations
b) Experiments can come from other tissues
c) There may exist technical artefacts, such as different sequencing and preparation methods
2. How is it done for bulk RNA-Seq
a) Typically use a linear model to regress out the confounding effect
b) This does not work for differing numbers of cell populations per batch
3. MNNCorrect
a) iterative matching of mutual nearest neighbours



h effects in single-cell RNA-sequencing data are corrected by Gunnar Rätsch, Valentina Boeva, Natalie Davidsc

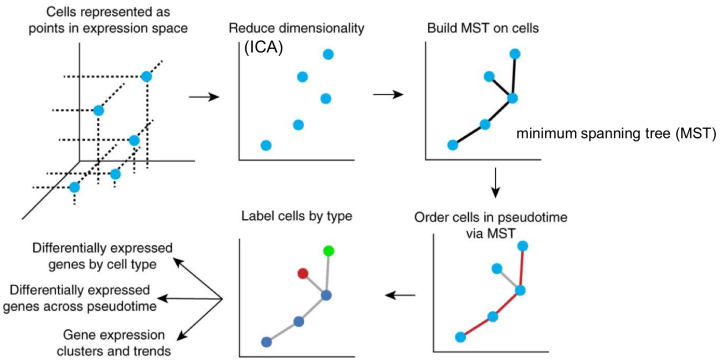
IV. Analysis applications beyond bulk RNA-Seq
  A. Single Cell Trajectory Estimation
    1. When do cells transition from one type to another and how many cell types exist?
      a) Using chemicals that induce differentiation, we can identify individual cells and their states and track them over time.
      b) This is not always possible, so we would like to estimate the trajectory given the status of the body in one single time point
        (1) Since cells are not synchronised in their state, can we order them based on their differentiation trajectory instead of the time they were sampled.

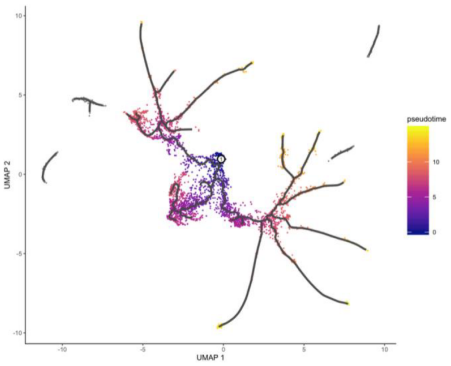


Align all cells by identifying cells that co-vary with one another across all genes.
Identify ***pseudotime*** ordering

    2. Monocle workflow
      a) Identify significant clusters
      b) Identify genes that are differentially expressed between clusters
        (1) These are out cell-state markers
      c) Construct Trajectories

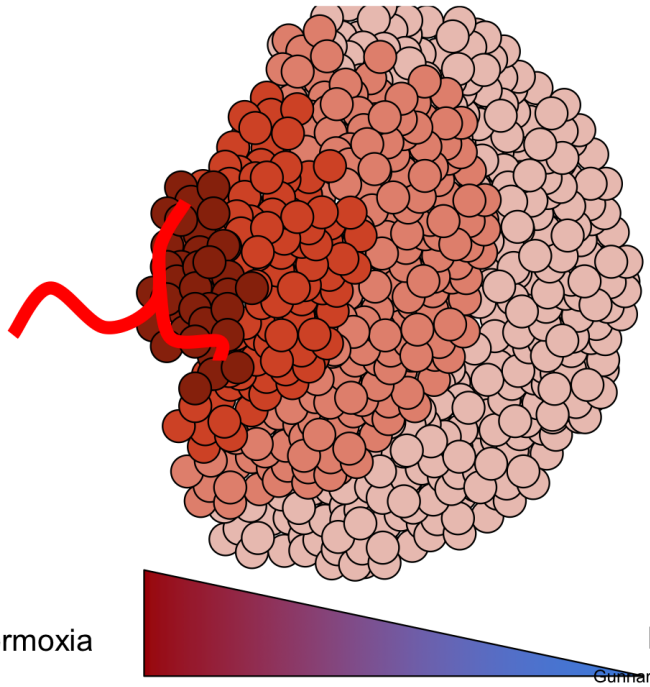
d) Identify groups of genes that vary similarly over pseudotime.





B. Spatial Transcriptomics

1. Gene expression is spatially distinct and we want to reconstruct that

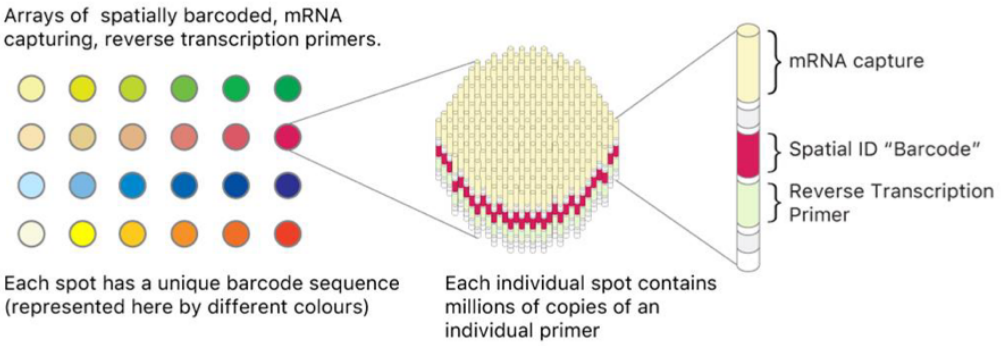


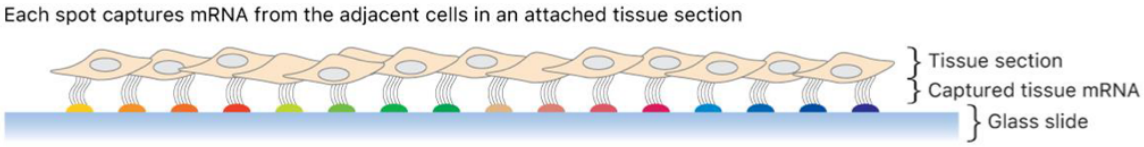
Gene expression is spatially distinct.

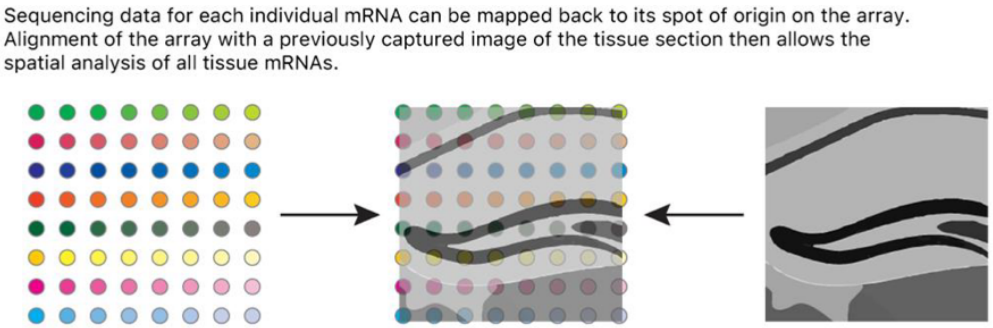
Normoxia

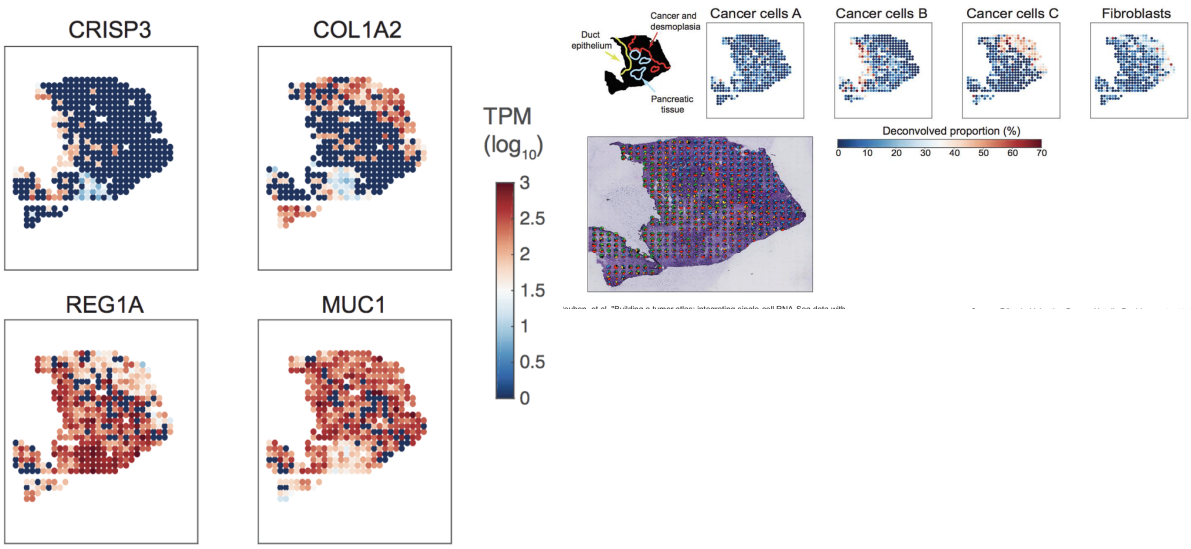
Hypoxia, ↑HIF1α

iptomics


Arrays of spatially barcoded, mRNA capturing, reverse transcription primers.
mRNA capture
Spatial ID "Barcode"
Reverse Transcription Primer
Each spot has a unique barcode sequence (represented here by different colours)
Each individual spot contains millions of copies of an individual primer


Each spot captures mRNA from the adjacent cells in an attached tissue section
Tissue section
Captured tissue mRNA
Glass slide


Sequencing data for each individual mRNA can be mapped back to its spot of origin on the array. Alignment of the array with a previously captured image of the tissue section then allows the spatial analysis of all tissue mRNAs.


CRISP3
COL1A2
REG1A
MUC1
TPM (log10)
3
2.5
2
1.5
1
0.5
0
Duct epithelium
Cancer and desmoplasia
Pancreatic tissue
Cancer cells A
Cancer cells B
Cancer cells C
Fibroblasts
Deconvolved proportion (%)
0 10 20 30 40 50 60 70

Lecture 8: Variant Calling

I. Understanding the problem setting and application
   A. What is a genomic variant/mutation?
      1. Working definition
         a) A genomic variant is a continuous substring of a genome which differs from a given reference genome substring at the same genomic location
      2. Type of variants
         a) Single nucleotide variants
         b) Multiple nucleotide variants
         c) Insertions/Deletions
         d) Homozygous vs. heterozygous
         e) Rare vs common
   B. Typical applications of variant calling
      1. Disease risk prediction
      2. Diagnosis
      3. Personalised drug selection (mutation specific drugs)
      4. Treatment recommendations
   C. Germline variant calling
      1. Assumptions
         a) Human genome has two copies (diploid) of every chromosomes (autosomes)
         b) Each copy represents the genome from one parent
         c) This limits the number of potential different in one sample at one position to at most 2
         d) After alignment, ideally each position will be covered by a mixture of reads from either of the two chromosomes
         e) In a perfect world, a homozygous position would show only reads with one letter, and a heterozygous position will have a 50/50 mixture of reads with both possibilities.
      2. Simplest variant caller
         a) Preparation
            (1) Align your reads to reference genome
            (2) Filter out bad reads
            (3) Find region of mismatches
            (4) Count number of differences
         b) Cases
            (1) If no difference: No mutation
            (2) If all reads consistently differ: Call a homozygous mutation based on this difference
            (3) If observe mix of reads supporting two sets of variants: Call a heterozygous variant based on these differences

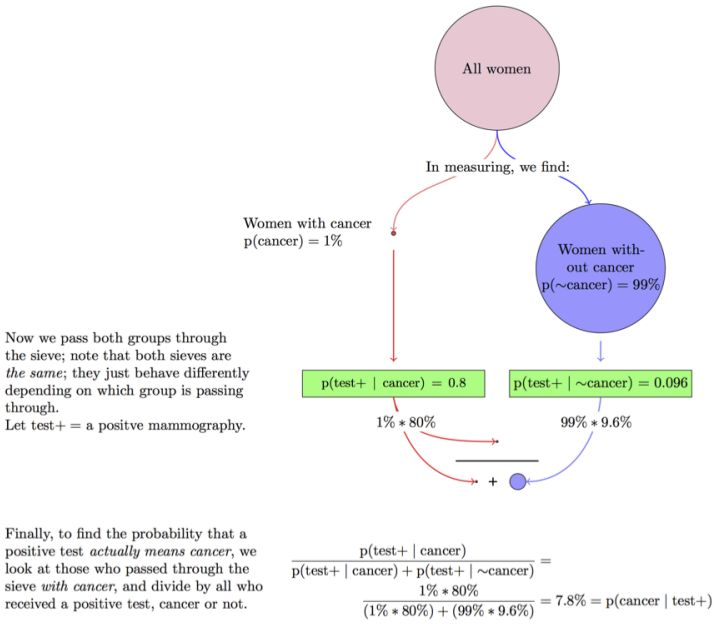
(4) Note: In case of heterozygous variant, both alleles may not match the reference.

c) However
   (1) reads can contain error with an error-rate dependent on sequencing technology
   (2) Errors are not uniformly distributed
   (3) Read coverage is not uniformly distributed
   (4) Reference bias
   (5) Alignment itself may not be uniform. (e.g: sequence context)

d) Consequence to consider for variant calling:
   (1) A heterozygous variant may not have 50% read support for either/both alleles
   (2) An observed difference may be an actual difference or a read error (sequence depth can help here)
   (3) Depend on sequence context mutation or error are more likely

3. Information we can use
   a) Typical things we now when calling a variant
      (1) Base qualities
      (2) Read quality (at nucleotide level)
      (3) Alignment quality (depending on aligner)
      (4) Genomic position (genomic context)
   b) Ideally we like an algorithm taking these uncertainties about variant calling into account. Let's focus on single nucleotide for now.

4. Bayes Theorem and Example
   a) $$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_i P(B \mid A_i)P(A_i)}$$



D. The MAQ Algorithm

1. MAQ is an aligner which calls variants during that process. It is one of the first broadly used variant callers
   a) Find ungapped match with lowest mismatch score for a specific read only considering positions that have less or equal than two mismatches in the first 28bp
   b) Assign quality of alignment PHRED score
      $$Q_s = -10\log_{10} P[\text{read is wrongly mapped}]$$
   c) Ambiguous read are mapped randomly with quality 0
   d) Create consensus genotype sequence inferred from Bayesian model
   e) Each consensus genotype receive thread quality
   f) SNP detection via comparing consensus sequence to reference and filtering to refine final set.
2. MAQ alignment quality score
   a) Given a read z and a reference sequence x:
      (1) $p(z|x,u)$: Probability of read z coming from position u on reference sequence x
      (2) Let's assume error are independent at site of the read, thus $p(z|x,u)$ is just the product of the error probabilities
      (3) Given two mismatches with thread quality 20 and quality 10, $p(z|x,u) = 10^{(-20+10)/10} = 0.001$
      (4) More general, if $p(u|x)$ is uniform:
      $$p_s(u|x,z) = \frac{p(z|x,u)p(u|x)}{\sum_{v=1}^{L-l+1} p(z|x,v)p(v|x)}$$
      L = Length of reference
      l = Length of read
      It follows that $Q_s(u|x,z) = -10\log_{10}[1 - p_s(u|x,z)]$
   b) In practice: Summing over reference sequence omitted for some well chosen constants.
3. MAQ genotype calling
   a) we can only have two alleles (a/b) (diploid genome) with $a, b \in A, C, G, T$, so three genotype possibilities
   b) Assume we observe k nucleotides b and $n-k$ nucleotides with allele a. Let $\alpha_{n,n-k}$ be the probability of observing $n-k$ errors from n bases and $< b, b >$ the true genotype:

$P(D| < b, b >) = \alpha_{n,n-k}$ accounting for n-k errors

$P(D| < a, a >) = \alpha_{n,k}$ accounting for k errors

$$P(D| < a, b >) = \binom{n}{k}(0.5)^k(1-0.5)^{n-k} = \alpha'_{nk}$$

Assuming the true genotype was <a,b> with a unequal b

c) Let's assume heterozygous calls have a different chance than homozygous once then we have the following priors:

$$p(<a,b>) = \begin{cases} (1-r)/2, \text{ if a unequal b (het)} \\ r, \text{ if a equal b (hom)} \end{cases}$$

d) MAQ now calls the genotype with

$\hat{g} = \text{argmax} p(g \mid D)$ and $Q_g = -10 \log_{10}[1 - P(\hat{g} \mid D)]$

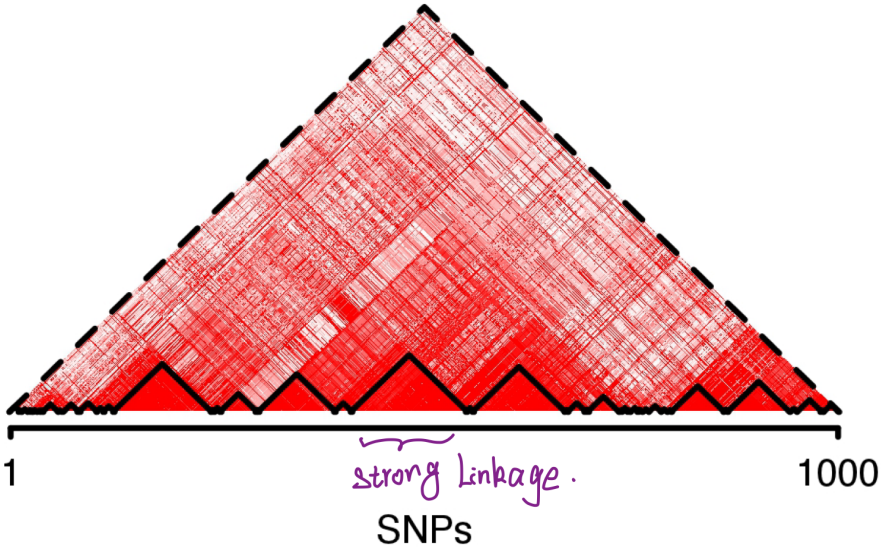
$$p_g(<b,b> \mid D) = \frac{\alpha_{n,n-k} P(<b,b>)}{\alpha_{n,n-k} P(<b,b> + \alpha'_{nk} P(<a,b>) + \alpha_{n,k} P(<a,a>}$$

$$p_g(<a,b> \mid D) = \frac{\alpha'_{nk} P(<a,b>)}{\alpha_{n,n-k} P(<b,b> + \alpha'_{nk} P(<a,b>) + \alpha_{n,k} P(<a,a>}$$

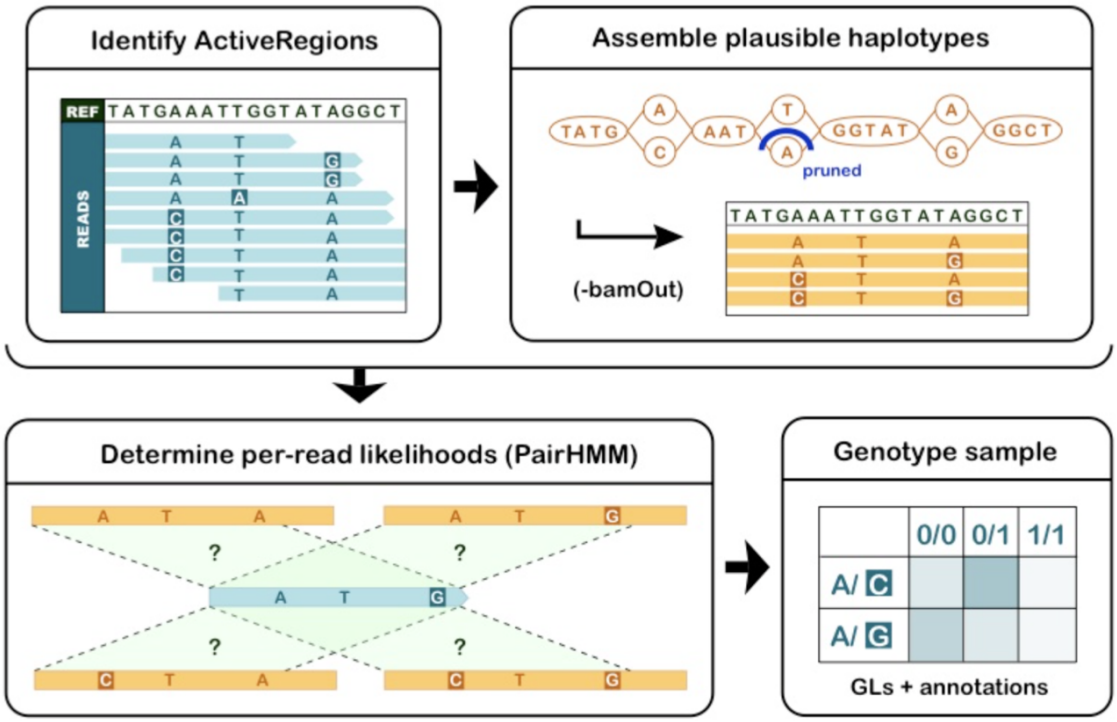
$$p_g(<a,a> \mid D) = \frac{\alpha_{n,k} P(<a,a>)}{\alpha_{n,n-k} P(<b,b> + \alpha'_{nk} P(<a,b>) + \alpha_{n,k} P(<a,a>}$$

4. Practical considerations
   a) Assuming error rates are independent and identical we can set $\alpha_{n,k} = P(n,k,\epsilon) = \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$ with the ϵ being the read error rate
   b) r typically set to 0.2 for known variants and 0.001 for new callset
   c) Report genotype maximising posterior probability
   d) Probability of k errors in n nucleotide $(\alpha_{n,k})$ is non-trivial unless assume independence.

E. Linkage blocks
   1. Correlation of SNPs across individuals in the population



II. Haplotype caller

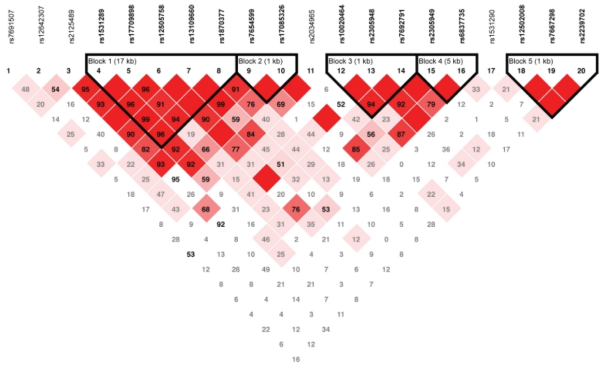


er Auweral

1. More detail at lecture 8 p20 - 30

Lecture 9: Linking genotypic information to clinical phenotypes Genome wide Association study.

I. General Idea
   A. Trying to understand whether there is a correlation between a phenotype and a genetic marker (SNP)
II. Interpreting variants
   A. Number of pair-wise differences between humans is about 3 million positions
   B. How do we gain knowledge about the effect of these differences?
      1. Variant effect prediction
      2. Genome wise association studies
III. Genome wide Association Study (GWAS
   A. Aim:
      1. Detect alleles (nucleotide values at SNP positions) significantly different between cases and controls
      2. Understand biological significance (why are these SNPs associated with the phenotype?)
   B. Linkage disequilibrium (LD)



1.Linkage disequilibrium (LD) is the non-random association of alleles at different loci in a give n population

2.Linkage disequilibrium is influenced by many factors, including the rate of genetic recombination, mutation rate and population structure.

   C. Advantages and disadvantages of GWAS
      1. No family tree needed, but just bulk genotyping data (usually SNP arrays or targeted sequencing for discovered variants)
      2. Translatable to clinic quickly: risk prediction, disease sub typing, drug development, drug toxicity
      3. Limited to large effects and common variants (Most relevant variants are likely rare)
   D. Testing for association
      1. What is our statistics to calculate the association between the allele genotype (SNP value) and the phenotype?
      2. Let's assume we have variants for both populations:

3. Contingency tables - Fisher's Exact Test

Allele	Cases (with AMD)	Controls (without AMD	Total Alleles
C	a	b	a + b
T	c	d	c+d
Total Alleles	a + c	b + d	a+b+c+d

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}$$

a,b,c,d are values in a contingency table
n is the total frequency

4. Contingency tables -χ^2 test

Allele	Cases (with AMD)	Controls (without AMD	Total Alleles
C	a	b	a + b
T	c	d	c+d
Total Alleles	a + c	b + d	a+b+c+d

$$E_1 = \frac{(a+b)(a+c)}{(a+b+c+d)} \quad \chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

$$Df = (2rows - 1) \times (2columns - 1) = 1$$

5. Simple χ^2 test on 3×2 table:

	H_z. reference	Heterozygous	H_z. alternate	Total
Case	N_{00}	N_{01}	N_{02}	N_0
T	N_{10}	N_{11}	N_{12}	N_1
Total Alleles	$N_{.0}$	$N_{.1}$	$N_{.2}$	N

$$\chi^2 = \sum_{j=1}^{m} \sum_{k=1}^{r} \frac{N_{jk} - \frac{N_{j.} - N_{.k}}{N}}{\frac{N_{j.} - N_{.k}}{N}}$$ with (m-1)(r-1) degrees of freedom

6. Alternatively use linear regression formulation:

$Y = \beta_0 + X_1\beta_1 + \epsilon \quad H_0 : \beta_1 = 0$ vs $H_a : \beta_1 \neq 0$

Y = phenotype {0,1} or $\mathbb{R}$

X_1= allele {0,1,2} for AA,AB, and BB

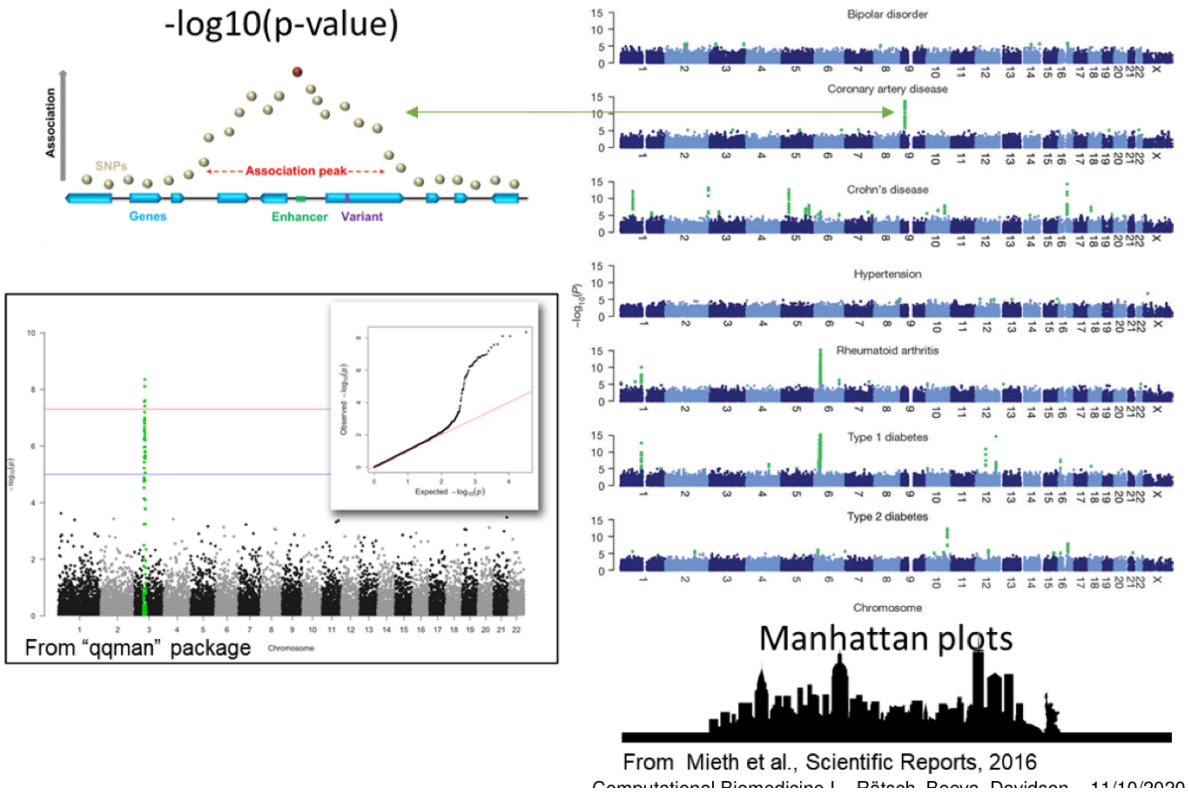
P-value calculation via linear regression slope T-test statistic:

$$t_{N-2} = \frac{\hat{\beta}_1 - B_0}{s_{\beta_1}}$$

where $B_0 = 0$ if $H_0 : \beta_1 = 0$, N is the sample size, $\hat{\beta}_1$ is the estimation of β_1, and s_{β_1} is the standard error of $\hat{\beta}_1$ and

$$s_{\beta_1} = \sqrt{\frac{1}{n-2} \cdot \frac{\sum (y_i - \hat{y}_i)^2}{\sum (x_i - \bar{x})^2}}$$

7. Typical visualisation : Manhattan plots



E. What distribution do GWAS p-values follow under the Null Hypothesis of no association?

1. Let $F_0(t)$ be the cumulative Distribution Function of T under $H_0 : F_0(t) = P(T \leq t \,|\, H_0)$
2. p = p-value corresponding to the observed value t of the testing statistics.

$$p = P(T > t \,|\, H_0) = 1 - P(T \leq t \,|\, H_0) = 1 - F_0(t)$$

$$F_0(t) = P(T \leq t \,|\, H_0) = P(F_0(t) \leq F_0(t))$$

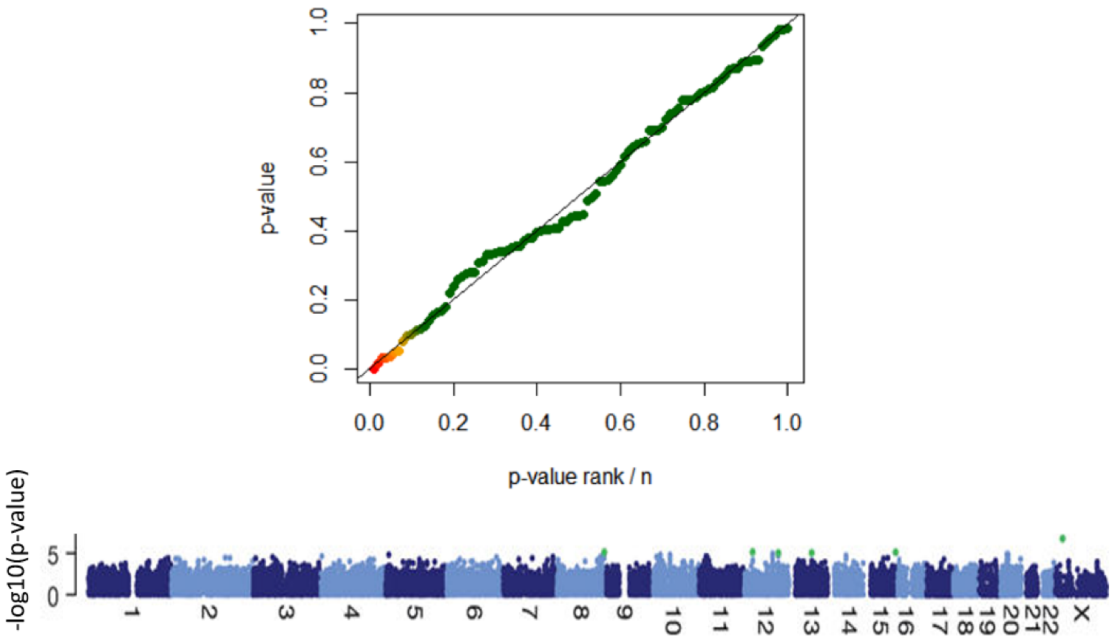
it follows

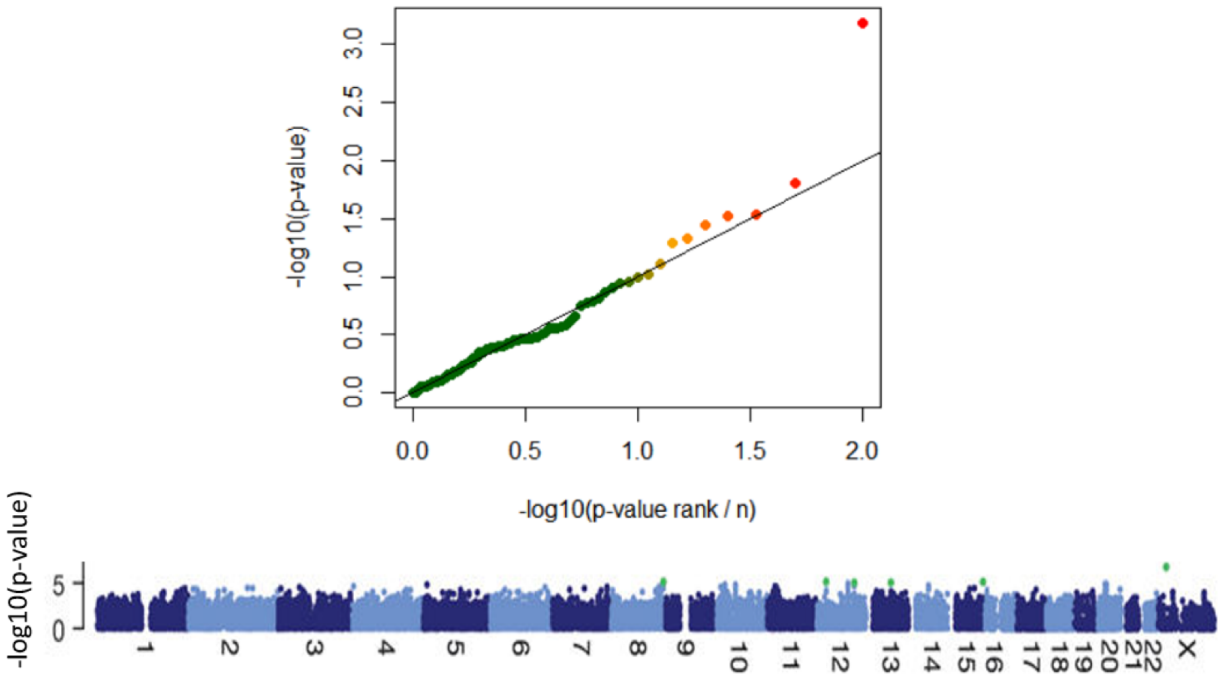
$$P(F_0(t) \leq F_0(t)) = F_0(t) \quad \text{or } P(F_0(T) \leq x) = x$$

3. This is essentially the definition of the uniform distribution. If $Z \sim U(0,1)$ then, $P(Z \leq x) = x$. So, $F_0(T) \sim U(0,1)$. But then, $1 - F_0(T) \sim U(0,1)$ as well.
4. Therefore, we can conclude that under the H_0, p-values are distributed $\sim U(0,1)$

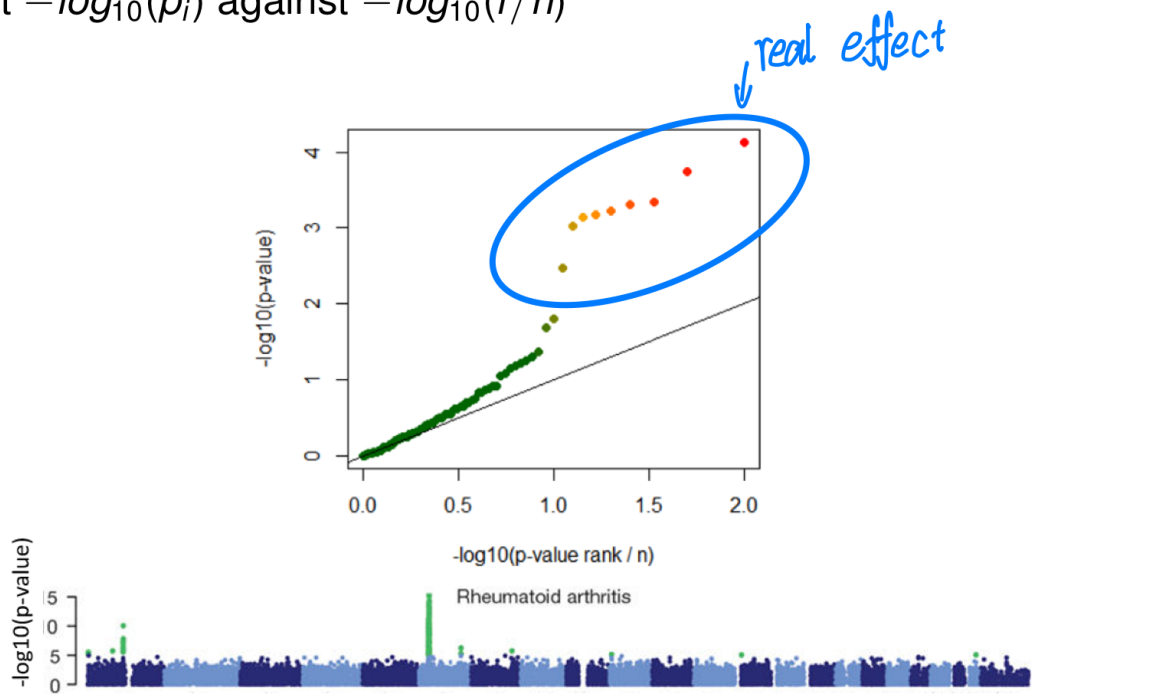
F. Visualisation of p-values on the uniform (quantile-quantile) QQ-plot

1. Rank data (p-values) in ascending order
2. Plot p_i against (i/n) or plot $-log_{10}(p_i)$ against $-\log_{10}(i/n)$
3. the result should be a diagonal straight line





Plot $-log_{10}(p_i)$ against $-log_{10}(i/n)$

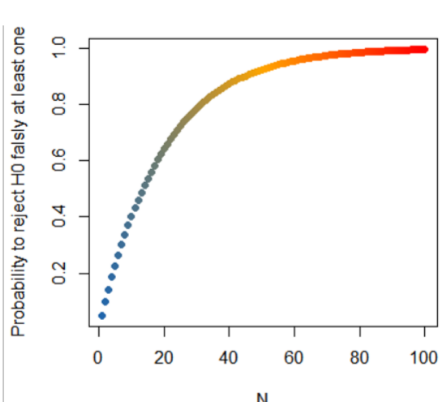


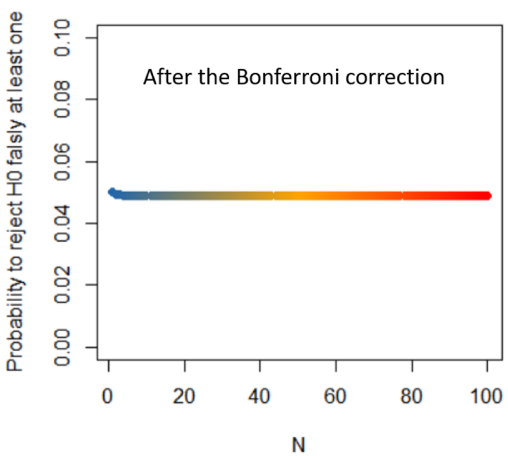
G. Multiple testing correction

1. Type 1 error: Reject H_0 if H_0 was actually true
2. If we set significance at 0.05, Type 1 error is at most 5% by definition
3. If we test independently N times, we will reject in exception $N \times 0.05$ tims
4. Family wise error rate: probability of making at least one false discovery
5. P(reject at lest once) = 1 - P(do not reject) $= 1 - 0.95^N$
6. Bonferroni approach for multiple test correction:

Given $p_1, \ldots, p_m$ values, then we reject the Null hypothesis for each $p_i \leq \frac{\alpha}{m}$

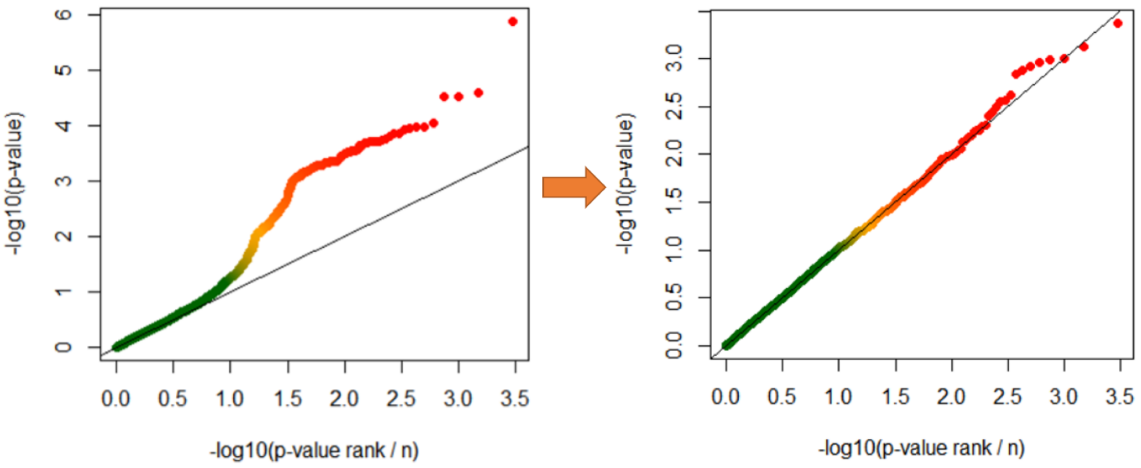
Main assumption: All tests are independent





H. What are the issues with GWAS?
   1. The studied trait may not be heritable
   2. Typically population can stem from different geographic regions
   3. Linkage will make it difficult. to identify specific causal variant

I. Population structure
   1. Wearing shapes is clearly not heritable
   2. Nevertheless significant difference in usage between populations
   3. All Russian population specific variants will likely be significant
   4. Addressing population structure
      a) Let X be a genotype matrix, then let $K = X^T X$. Do principle component analysis (PCA) on X: $K = W \wedge W^T, T = XW$. Where T is the matrix with principle components (PCs)



b) Linear mixed models: accounting for structure between individuals

$$y = X\beta + u + \epsilon \text{ with } \epsilon \sim N(0,\sigma_\epsilon^2 I) \text{ and } u \sim N(0,\sigma_K^2 K)$$

Where u is a vector of polygene background effects, and K kinship or relatedness matrix.

For example K can be genetic similarity as defined earlier. Let's estimate the variance component as well as $\sigma_\epsilon^2; \sigma_K^2$

$$Var(y) = V = \sigma_K^2 K + \sigma_\epsilon^2 I$$

$$H = K + \delta I \text{ with } \delta = \frac{\sigma_\epsilon^2}{\sigma_K^2}$$

$$l(y;\beta,\sigma_K,\delta) = \frac{1}{2}[-n\log(2\pi\sigma_K^2) - \log|H| - \frac{1}{\sigma_K^2}(y - X\beta)^T H^{-1}(y - X\beta)]$$

Likelihood $l(y;\beta,\sigma_K,\delta)$ is maximised for :

$$\hat{\beta} = (X^T H^{-1} X)^{-1} X^T H^{-1} y$$

$$\hat{\sigma}_K^2 = \frac{R}{n} \text{ with } R = (u - X\hat{\beta})^T H^{-1}(y - X\hat{\beta})$$

and δ has to be optimised numerically through spectral decomposition of H. The test statistics then is $F-$ statistic:

————————————F statistics ————————————————————

$$F_{n-q}^p = \frac{(M\hat{\beta})^T (M(X^T \hat{V}^{-1} X)^{-1} M^T)^{-1} (M\hat{\beta})}{p}$$

for full-rank $p \times q$ matrix M to test $H_0 : M\beta = 0$

The F-statistic asymptotically follows a χ_p^2 distribution unless the estimated variation component meets the boundary of parameter space.

For our specific example, the F statistics can be simplified to

$$F_{n-q}^p = \frac{(\hat{\beta})^T ((X^T \hat{V}^{-1} X)^{-1})^{-1} (\hat{\beta})}{p}$$

where q is the dimensionality of β and n the number of individuals tested.

————————Likelihood ratio test————————————————————

Likelihood Ratio $LR = -2[l(y;\beta_1 = 0,\sigma_K,\delta) - l(Y;\beta_1,\sigma_K,\delta)]$

$LR \sim \chi_q^2$ where q is the number of parameters in β set to 0.

J. Missing heritability
   1. From twin-studies and other approaches we have a good idea of heritability of some traits
      a) Height is roughly 80% heritable
      b) Found variants can only explain small fraction (45%)

K. Meta analysis
   1. Many resources have been pooled into studying the genetic association of various phenotypes. One may want to reuse such resources. Combining p-values for a given SNP from k studies.

$$\chi^2_{2k} = -2\sum_{i=1}^{k} ln(p_i)$$

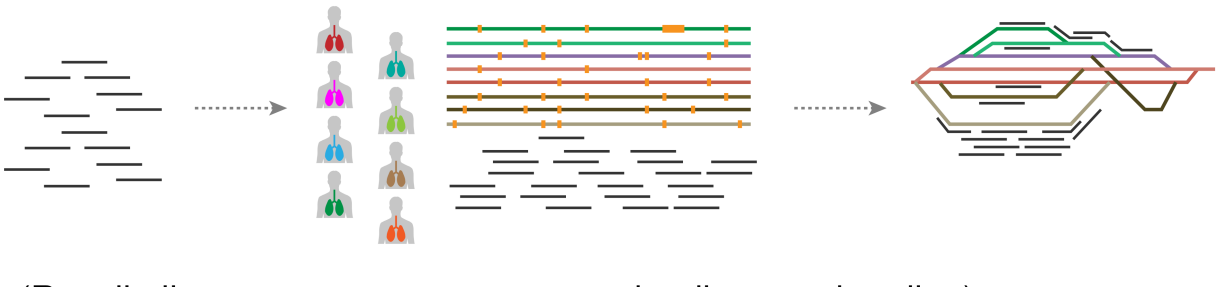
   2. k is the number of studies that you are trying to combine. The above equations works for one single SNP at a time. That means you fix the position of the genome, and you look through all the p-values k studies show, then you plug them into the above equation
   3. Why does it work?
      a) A log of a uniform follows an exponential distribution. Factor 2 yields chi-squared. If $X \sim U(0,1)$, then $-2\log(X) \sim \chi^2_2$.
      b) Issue: p-value combination does not take effect direction into account.

IV. Summary
   A. GWAS allows for the detection of variants of small and medium effect
   B. GWAS can detect numerous genetic risk factors, even those diffusely distributed across the genome
   C. GWAS can detect variants located in poorly understood regions of the genome.
   D. GWAS is highly reproducible
   E. Linkage disequilibrium may make it difficult to find the causing variant; also it results in SNPs not being independent (statistical issues)
   F. We can use QQ-plots, Manhattan plots and multiple test correction of p-values to identify variants associated with the phenotype
   G. Often it is difficult to biologically interpret the detected variants. (i.e. understand why they cause the phenotype)
   H. In GWAS on mixed populations, we should correct for the population structure.

Lecture 10: Germline Variant Calling and Variant Effect Prediction

---

I. Graph genomes for variant calling
   A. observations:
      1. Most calling methods discussed so far rely on read alignment to linear reference genome
      2. Calling quality depends on quality of the underlying alignment (and underlying reference genome)
      3. Form intermediate graph representation to express local variant
      4. Calling of long insertions borrows techniques from genome assembly
      5. Why not directly use sequence graphs to call genomic variants
   B. Strategies
      1. We can distinguish two different general strategies for graph based variant calling:
      2. Reference based: methods encode known genetic variant in a reference coordinate system using a graph representation. Remaining procedures are similar to the approaches using a linear reference.
      3. Reference free: methods only rely on the raw sequencing data and use graph data structures to represent detected variation ad hoc.
      4. Depending on the method, different graph data structures are used. We will give examples for both string graphs and de Bruijn graphs.
   C. Reference based variant calling with string graphs
      1. Strategy is similar to methods based on a linear reference, only encoding additional variation as a graph.



      2. General steps
         a) Transform reference sequence variation into graph representation
         b) align read data to the graph
         c) augment graph based on alignment support with additional insertions and deletions
         d) re-map the reads to augmented graph for variant quantitation

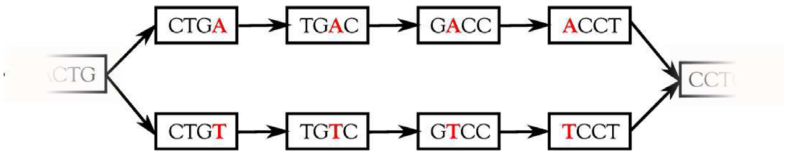
e) variant normalisation and filtering

3. Advantages:
   a) less reference bias and less ambiguity for read mapping
   b) incorporation of prior knowledge helps find rare variation
   c) variants can be reference by reference coordinates

D. Reference free variant calling with de Bruijn graphs

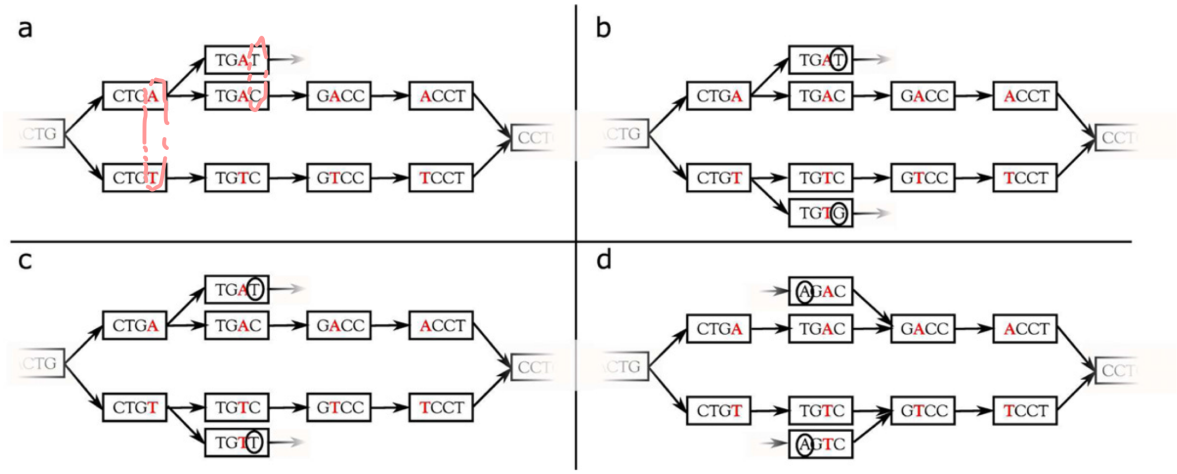
1. Recall that a de Bruijn graph represents input sequences are unique k mer nodes that share a directed edge, if two nodes share a substring of length k-1.
2. Single nucleotide variations in the input will form bubbles in the graph that have length k.
   a) sequencing errors would also cause bubbles

S1: . . . CTGACCT . . .
S2: . . . CTGTCCT . . .

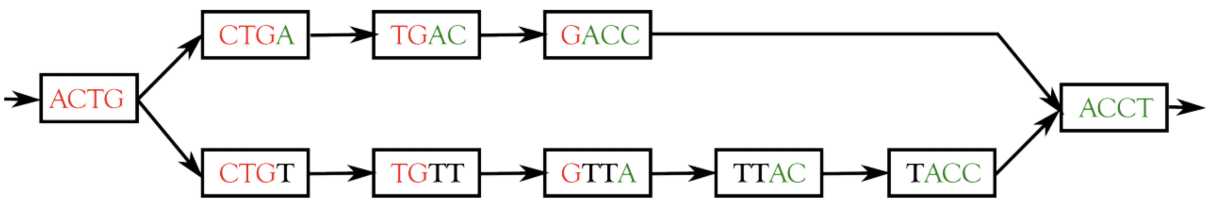


3. Graph can be walked to collect bubbles and identify SNPs/SNVs.
4. Calling is becoming more difficult if variants have a distance less than k to each other

each other

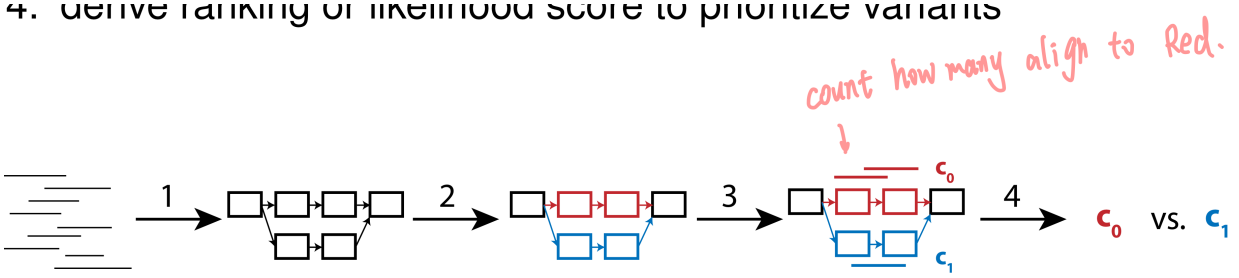


5. or long insertions are handled



6. General steps:
   a) assemble input sequencing data into (coloured) de Bruijn graph
   b) identify local variants as bubbles in the graph
   c) compute path quantitations for bubbles on the read data
   d) derive ranking or likelihood score to prioritise variants

4. derive ranking or likelihood score to prioritize variants



7. Path quantitation:
   a) Count number of reads that are compatible with each of the paths(not alignment in the classical sense)
      (1) one read could be compatible for both paths
   b) allow for no or a only very small number of mismatches
8. Ranking / Scoring:
   a) Several possible strategies:
      (1) Simply compute ratio of alternative path counts $\frac{c_0}{c_1}$ and use fixed thresholds to determine genotype
      (2) maximise likelihood under a give distribution assumption and incorporating error model.
   b) Possible likelihood estimation using a binomial model:

$$P(c_0, c_1 \mid 0/0, \rho, \alpha) = (1-\alpha)^{c_0} \cdot \alpha^{c_1} \cdot \binom{c_0 + c_1}{c_0} \cdot \frac{1-\rho}{2}$$

$$P(c_0, c_1 \mid 1/1, \rho, \alpha) = \alpha^{c_0} \cdot (1-\alpha)^{c_1} \cdot \binom{c_0 + c_1}{c_0} \cdot \frac{1-\rho}{2}$$

$$P(c_0, c_1 \mid 1/0, \rho, \alpha) = \left(\frac{1}{2}\right)^{c_0+c_1} \cdot \binom{c_0 + c_1}{c_0} \cdot \rho$$

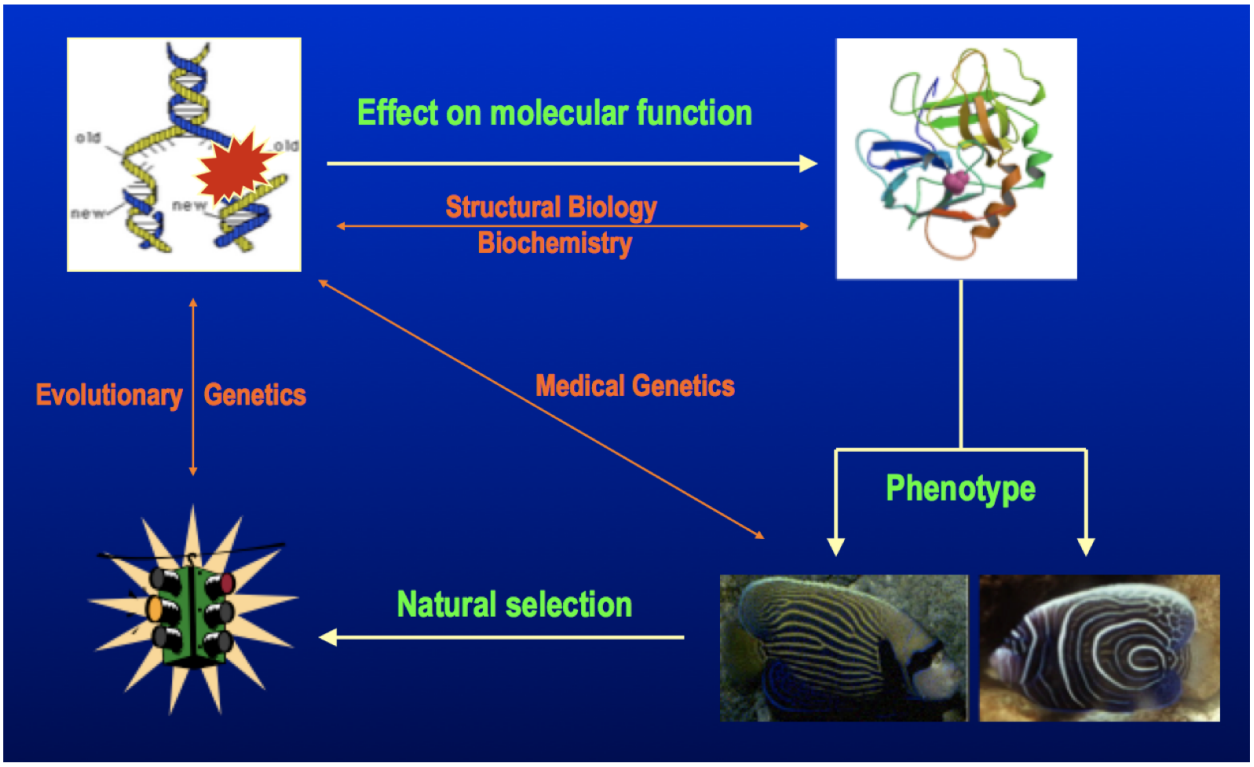
where α is the read error probability (e.g., 0.01), c_0 and c_1 are the two path counts and ρ is the prior probability for a heterozygous genotype.

Bayes theorem gives us the probabilities of genotypes given the data.

$$g = \underset{g}{\text{argmax}}\, p(g \mid c_0, c_1, \rho, \alpha)$$

II. Variant interpretation

A. Motivation

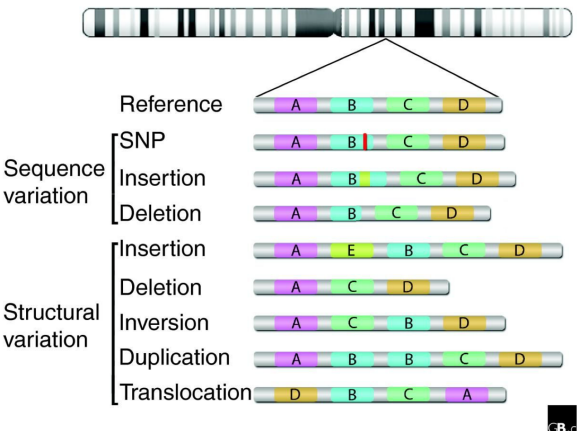


B. Interpreting variants

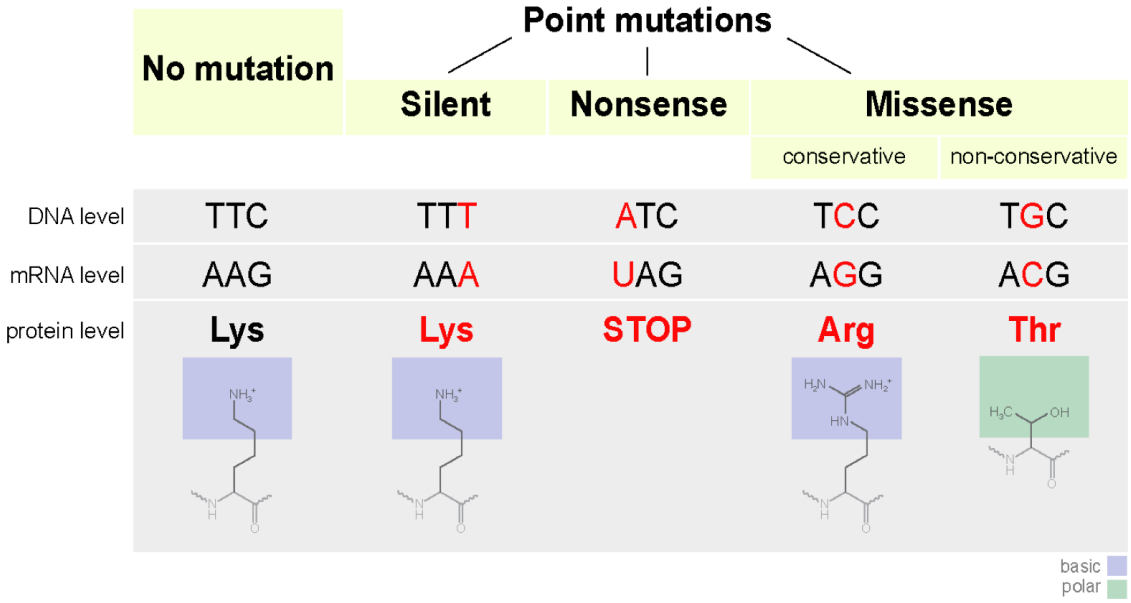
1. Number of pair-wise differences between human's about 3 million loci!
2. How do we gain knowledge about the effect of these differences
   a) Genome-wide-association analysis
   b) Variant effect prediction

C. Type of mutations

1. Genetic Variants' frequency in population should be large in order for it to be picked up in the association studies in populations.
2. rare variants typically have larger phenotypic effect than common variants but it can not be picked up by the association studies. But you can study the rare variants using linkage studies in one's family



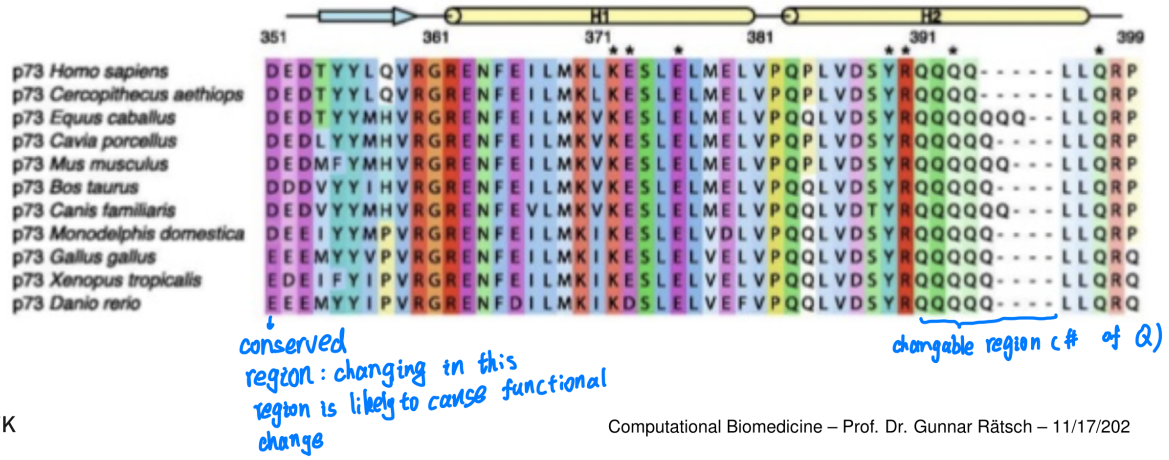
D. Type of mutation effect on point mutations
   1. point mutations
      a) Silent: no phenotypic effect of the mutation
         (1) on protein level, they encode the same amino acid.
      b) Nonsense: after mutation the codon changed to STOP codon
      c) Missense: changes codon so that a different protein is created
         (1) conservative: the properties of the amino acid remain the same
         (2) non-conservative: the properties of the amino acid does not remain the same



      d) There are other ways to categorise effect of mutations (function/ fitness/inheritance etc.)

E. What do we need to know to build a model of mutation effect
   1. What information can we leverage and what matters (feature generation)
   2. What classes are relevant (Multi-class?)
   3. What is a good training data set of relevant and non-relevant mutations (if going with ML)

F. SIFT (Sorting Intolerant From Tolerant)
   1. Generate alignment of different proteins across different species
   2. Identify protein which overlaps mutational position of interest
   3. Homology search (Find all similar protein sequences) using PSI_BLAST (position weight based)
   4. Multiple sequence alignment from PSI-BLAST
   5. Calculate probabilities



6. p_{ca} = probability of amino acid a at position c

$$p_{ca} = \frac{N_c}{N_c + B_c} g_{ca} + \frac{B_c}{N_c + B_c} f_{ca}$$

N_c is the total number of sequences in alignment

g_{ca} sequence-weighted frequency

f_{ca} pseudocounts drawn from a Dirichlet

B_c total number of pseudocounts $= exp(D_c)$ where $D_c = \sum_a (r_a g_{ca})$

with r_a is the rank of amino acid a substitution score with respect to the reference amino-acid

Normalise to $p^s_{ca} = p_{ca} / \max(p_{c.})$ cut-off at $p^s_{ca} < 0.05$ are deleterious

   a) Do not need to memorise the equations but generally the probability p_{ca} will be low if position a is highly conservative and will be high if the position a is not highly conservative.

7. Characteristics of SIFT
   a) It is build on sequence similarity of proteins
   b) It is not trained
   c) "Magic" happens on two ends
      (1) Finding set of similar proteins
      (2) Calculating probabilities

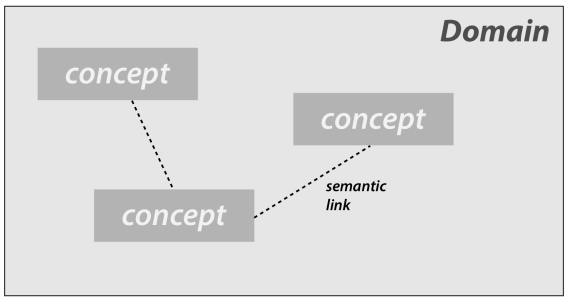
G. Focus on coding areas of the genome only?
   1. Direct functional link between mutation and amino-acid change onto phenotype
   2. Leverage insights into protein structure knowledge
   3. Ignores larger set of non-coding and synonymous coding mutations

H. Polyphen-2 Calssifier

1. In the Polyphen classifier, we have certain genome structures which is then used with a classifier naive bayes or logistic regression to predict the probability
   a) labeled training set build structures, multiple sequence alignments, a few sequence annotations which are available for protein domains and etc.
   b) Based on the features we can learn a classifier
   c) classifiers are usually pretty simple since the labels training set is usually very small.

Lecture 11: Ontology Systems and their Applications

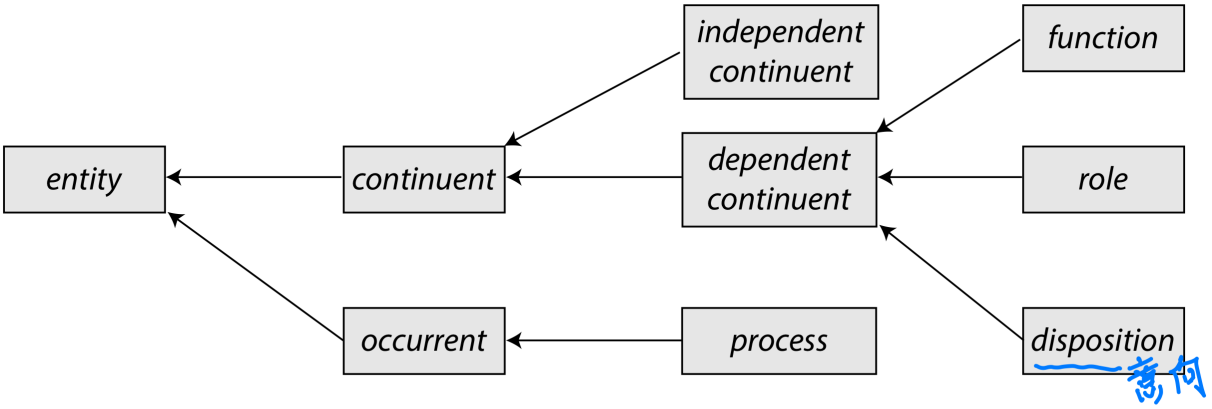
I. General Definition
   A. Goal : to build semantic models for the domains of reality. Items of the real world are represented by concepts and their relation as semantic links



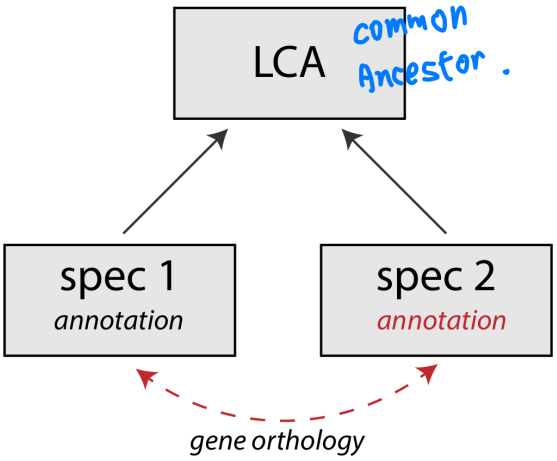
II. Applications
   A. Ontologies have a wide range of possible applications. Biomedicine is one of the most prolific areas of specialisation of ontologies
   B. semantic aware search
   C. standardisation (controlled vocabulary)
   D. data integration (over inhomogeneous sources)
   E. automatic reasoning within a domain (to generate / uncover knowledge → **inference**)
   F. error/contradiction detection (e.g., in medical databases)
   G. automatic classification
   H. over-representation analysis over a given set

III. Upper-Level Ontologies
   A. Abstract descriptions of ontologies in general
   B. Basic Formal Ontology(BFO)
      1. top level ontology for most biomedical ontologies
      2. guideline on how to transform reality into coherent abstract concepts (categorisation)
   C. BFO has influenced Gene Ontology, Protein Ontology, Cell Ontology and many more.
   D. Defines **universals** (types/classes) and **particulars** (instances/ individuals)
   E. Further distinguishes between **continuants** and **occurrents.**
   F. **Continuants vs Occurents**
      1. **Continuant** are persistent objects that preserve their identity over time (despite changes of properties)
         a) bearer of quality: e.g. cooler
         b) bearer of a realisable entity: e.g. ability of an entity

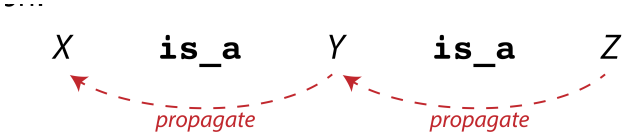
2. **Occurrent** is an entity that happens / develops through time and describes an event that continuants participate in.



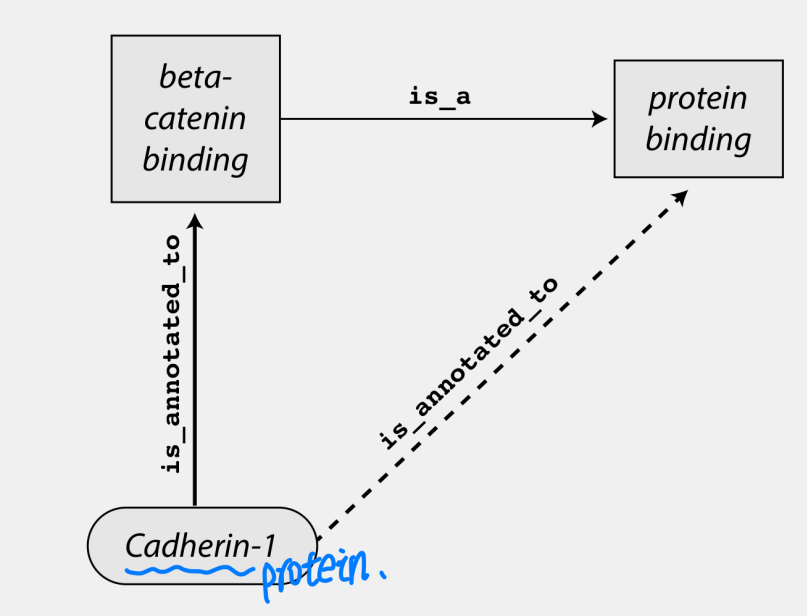
IV. Gene-Ontology
   A. Represents a structured vocabulary for the annotation of genes and their products
      1. Goal to explicit definitions for each and every biological concept
      2. originally structured as tree, intermediary as DAG (Directed acyclic graph) and recently as full graph
   B. Sub-ontologies
      1. The Gene Ontology has three sub-ontologies
      2. molecular function: describes the biochemical activity of a product
         a) enzymatic reaction; specific binding
      3. biological process: describes a biological objective
         a) change of cell state, regulation
      4. cellular component: describes location inside the cell where the product is active.
         a) nucleus, cytosole
   C. Relations
      1. Relational links between the Gene Ontology concepts form a graph structure that can be used for annotation propagation or inference
      2. is_a → sub-class relation (hierarchy)
      3. part_of → part-whole relation
      4. instance_of → class instantiation (speciation)
      5. regulates → direct effect between two concepts of the process sub-ontology
   D. Annotation propagation
      1. Graph structure of the GO us used to propagate annotations along links and infer new labels.



2.similar genes have similar annotation
3.use gene orthogology across species (link to LCA)
4.propagate annotations across links
5.In GO notation:

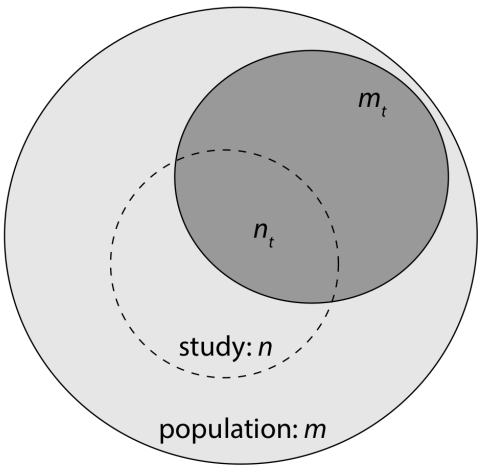


E. Example from gene ontology
   1. From Cadherin-1 being beta-catenin binding and beta-catenin binding being a specialisation of protein binding, it is inferred that Cadherin-1 is also protein binding



V. Overrepresentation Analysis
   A. Central Question: Does an ontology term (e.g., GO term) annotate more members of a given set (e.g., genes) than expected by chance?
   B. In a biomedical context often used for hypothesis generation for experimental follow up and testing
   C. Common strategy: term-for-term testing

VI. Term-for-term Testing
   A. Perform Fisher's Exact Test on each term individually
   B. M … size of population e.g. all of the genes we are looking at
   C. N … study set. E.g. subset of M, genes found by differential expression analysis



D.$M_t \subseteq M$ … subset of M with annotation t

E.$N_t \subseteq N$ … subset of N with annotation t

F.|M| = m; |N| = n; $|M_t|$ = m, |N| = n

$|M_t| = m_t; |N_t| = n_t$

G.We use **hypergeometric test** to compute whether our observation represents a significant enrichment

VII. Hypergeometric Test

A. Based on the hypergeometric distribution:

$$X_t \sim h(k \mid m; m_t; n) := P(X_t = k) = \frac{\binom{m_t}{k}\binom{m - m_t}{n - k}}{\binom{m}{n}}$$

B. Chance to find exactly k items labeled with t in a study set of size n, if m_t of m elements are in the population M are labelled with t

C. Hypotheses:

1. H_0: no positive association of term t and study set n
2. H_1: there is an overrepresentation of t in the study set

D. One-sided test:

$$P(X_t \geq n_t \mid H_0) = \sum_{k=n_t}^{\min(m_t,n} \frac{\binom{m_t}{k}\binom{m - m_t}{n - k}}{\binom{m}{n}}$$

this equation gives us the p-value of the hypergeometric test

E. Multiple Testing Correction

1. The Gene Ontology has several ten thousand terms annotated. Testing each term individually incurs a large multiple-testing burden.
2. Strategies for prevention / correction:
   a) Test only terms that annotate at least one item in the study set.
   b) Use corrective measures on the resulting p-values (Bonferroni, Benjamini-Hochberg)
3. Problems:
   a) Number of terms to test might still be large -despite reduction

b) Methods for multiple-testing correction assume **independence**, which is not true for many ontology terms that share a relationship through propagated annotations.

VIII. Gene Set Enrichment Analysis (GSEA)

A. When doing the term-for-term testing, we have to specify a threshold in order for the study set to be formed. e.g. we have to specify a threshold for p-value of the differential expression analysis. We do not want that.

B. Task: Given a list L of n items pre-ranked by a feature of interest (e.g., genes by differential expression between two samples), assess whether distribution of terms annotating a subset S of L is associated with the given ranking.

C. Approach: Compare fractions of items in S vs. fraction of items not in S relative to their ranks r_j up to a given position i in the ranked list L.

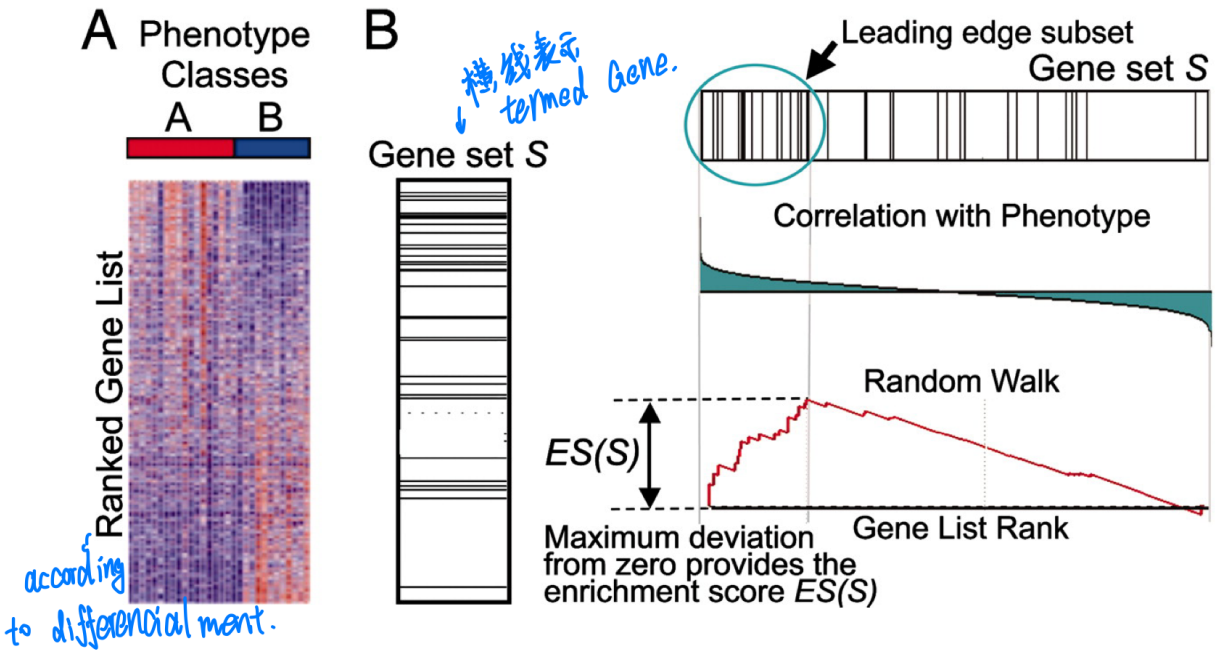
Compute an **enrichment Score** $(ES = \max_i(|P_{hit} - P_{miss}|))$:

$$P_{hit}(S,i) = \sum_{g_j \in S \ \ \forall j \le i} \frac{|r_j|^p}{n_R}, \ \textbf{with } n_R = \sum_{g_j \in S} |r_j|^p$$

$$P_{miss}(S,i) = \sum_{g_j \notin S \ \ \forall j \le i} \frac{1}{n - n_S}, \ \textbf{with } n_S = |S|$$

1. When p is set to 1, it is the fraction of gene that have the annotation vs. the fraction of gene that does not have annotation.

GSEA Approach:



D. Significance Assessment

1. To assess statistical significance of ES of a given gene set, a permutation test is used
   a) Generate k random gene sets M_i with k typically > 1000
   b) compute empirical distribution of $ES(M_i)$ from the random set
   c) assess significance of $ES(S)$ relative to empirical distribution

$$P = \frac{|\{i \,|\, ES(M_i) \geq ES(S)\,|\,\}}{k}$$

2. In genomics gene set enrichment analysis is a standard tool for functional assessment of ranked gene lists(e.g. from differential expression assays)

assessment of ranked gene lists (e.g., from differential expression assays).

[Mootha et al., 2003]

E. Summary 1

1. ontologies are an important tool for data standardisation and knowledge representation
2. Formal setup allows for automatic reasoning and inference
3. Biological applications use ontologies for knowledge generation and enrichment analyses.

F. Goals of Ontologies in Biomedicine

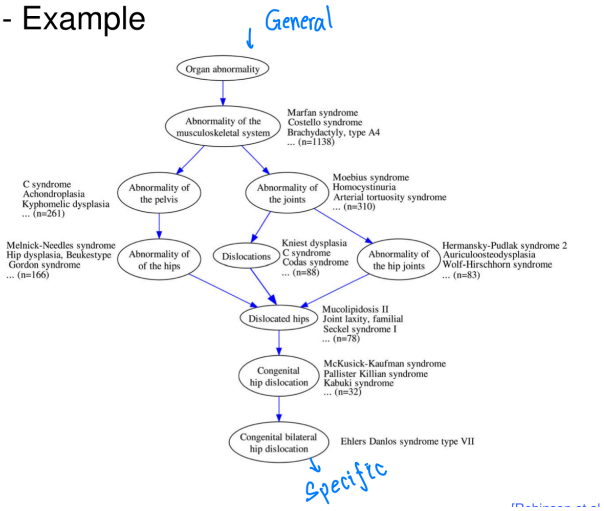
1. So far, we look at common motivations for ontologies in a genomic setting, such as enrichment analyses and functional annotations using Gene Ontology
2. In biomedicine one also needs to :
   a) Generate and maintain a controlled diagnostic vocabulary
   b) Integrate with other data sources (also ontologies) for knowledge generate

c) perform comparative analysis between instances depending on ontology terms and their relationships

G. Coordination and Unification
1. Over the past years, many different ontologies have developed in the biomedical field. In addition to defining relationships between entities, they are commonly used as a source of vocabulary
2. WIth the many different ontologies arising, problems of interoperability and standardisation increased, counteracting the principle of an ontology
3. The Open Biological and Biomedical Ontology (OBO Foundry is an initiative to standardise existing ontologies and set the on an equal theoretical footing following principles f the BFO

H. Human Phenotype Ontology(HPO)
1. important tool for semantic analyses in biomedicine
2. Main purpose of the HPO:
   a) connect between gene families and phenotypic disease families
   b) structure and organise phenotypic features of hereditary diseases
3. Construction of the HPO:
   a) Medicine has a clear need for controlled vocabularies / ontologies
   b) Syntactic ambiguity: some terms are identical but they are used simultaneously. For example: Generalised amyotrophy and generalised muscle atrophy etc.
   c) Creating the HPO included lots of manual curation:
      (1) transformation of all OMIM terms into HPO based on OBO-edit
      (2) all features with occurrence > 1 in OMIM became HPO term
      (3) adapted Smith-Waterman alignment between single occurrence OMIM terms and HPO hierarchy to identify synonyms and parent/child relationships (is_a())
   d) Example

O - Example



4. Similarity Measures
   a) Comparative analysis need the definition of similarity measures between ontology terms
   b) Observations
      (1) terms closer to the root represent more general concepts
      (2) information content of a term can be estimated through annotation frequency

      ⇒Sharing a more specific term will imply a higher similarity
5. Similarity of two terms t_1, t_2 sharing ancestors $A(t_1, t_2)$:

$$sim(t_1, t_2) = \max_{a \in A(t_1, t_2)} -\log p(a)$$

where $p(a)$ is the probability of term a measured as its frequency of annotation over all diseases in the database.

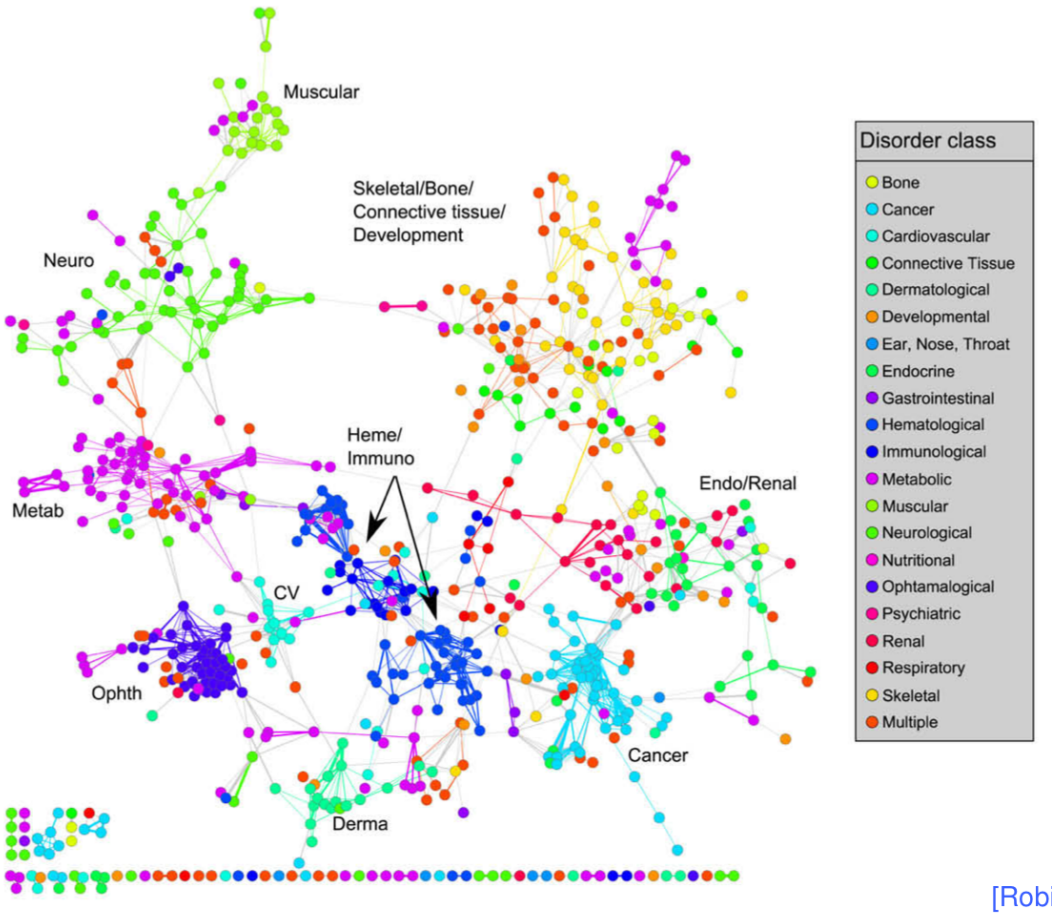
6. Based on the similarity of terms, we can define the similarity of two diseases d_1 and d_2

$$sim(d_1 \rightarrow d_2) = \text{avg}\left[\sum_{s \in d_1} \max_{t \in d_2} sim(s, t)\right]$$

   a) To break the asymmetry of the distance:

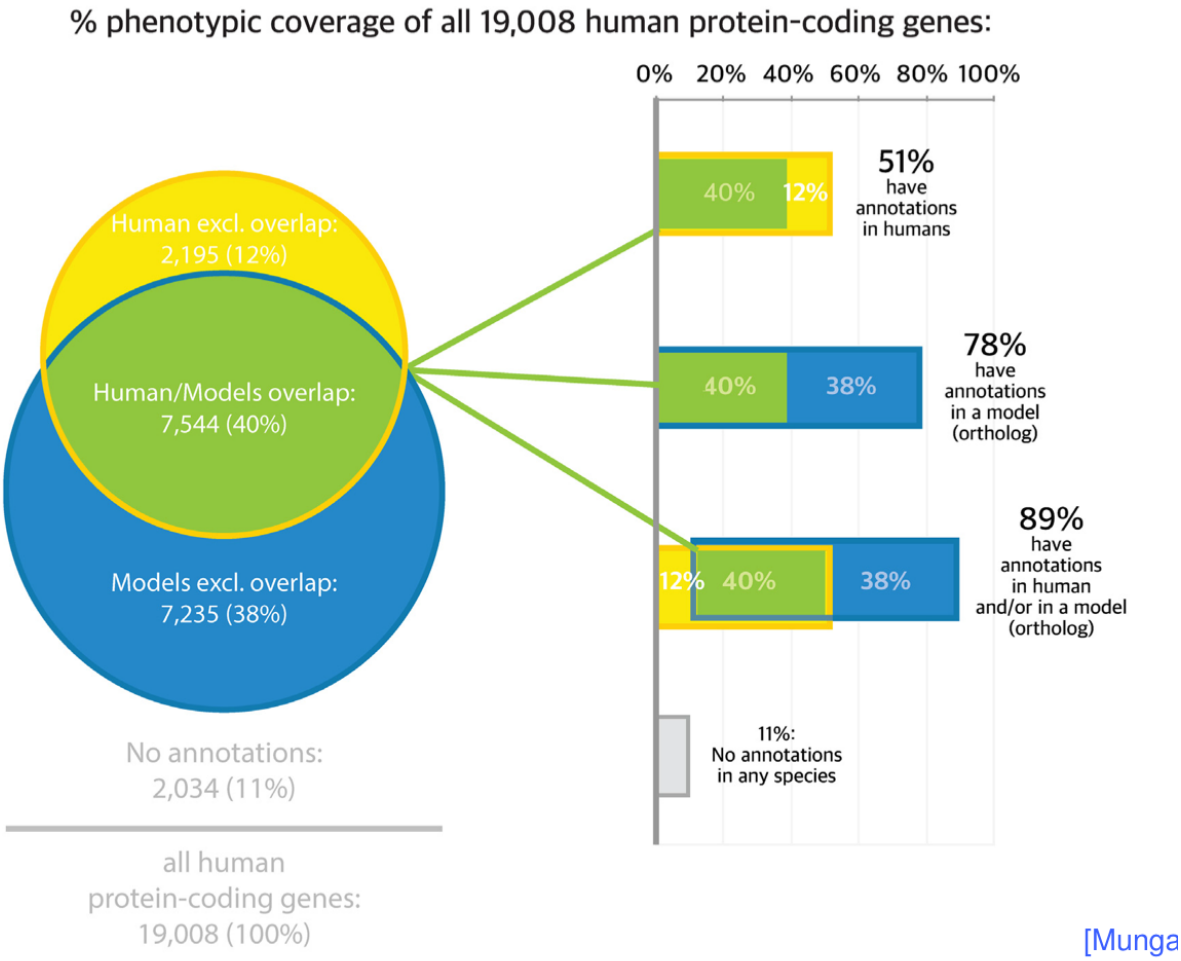
$$sim(d_1, d_2) = \frac{sim(d_1 \rightarrow d_2) + sim(d_2 \rightarrow d_1)}{2}$$

7. Example : Diseases clustered by similarity



[Robinson et

IX. The monarch Initiative
- A. The monarch Initiative is an open science, collaborative effort to
  1. semantically integrate genotype-phenotype data from many species and sources
  2. develop an integrated knowledge base, analytical tools and web services
  3. support precision medicine and disease modelling
- B. Especially bridging the gap to model-organism databases drastically increases annotation coverage of the human genome
- C. Annotation coverage increases from 50% to almost 90% of human coding genes
- D. Model Organism Annotations



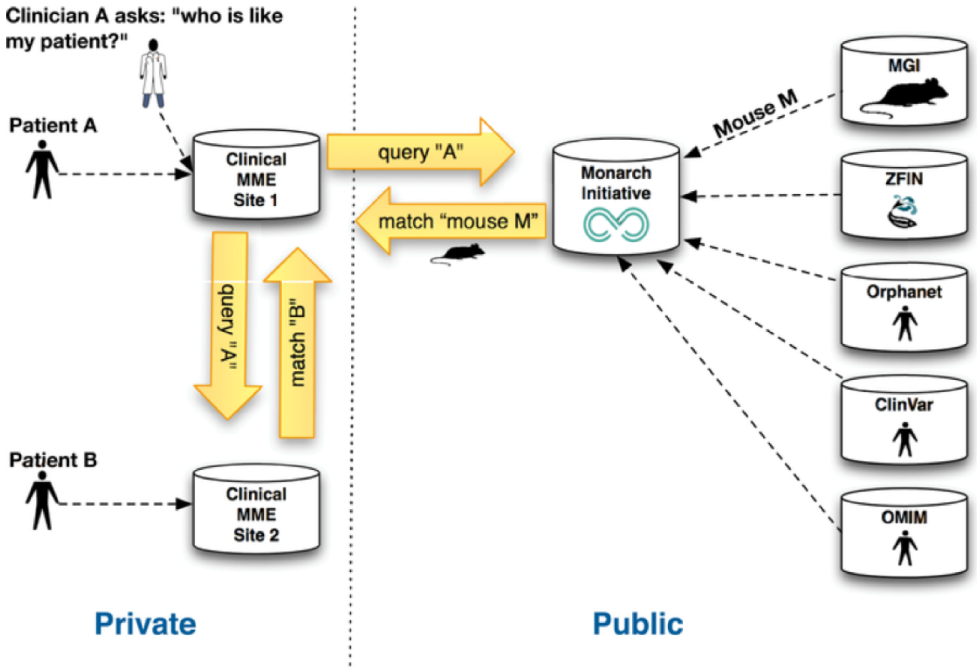
[Munga

- E. Missing Disease Associations
  1. Large fraction (>70%) of variants with a predicted effect of near-complete depletion of protein product have no disease phenotype assigned
  2. The Monarch Initiative concept:

a) enables phenotype comparisons within and between species based on computational reasoning
b) exploits that similarities in genetic aberrations often result in similarities of phenotypic aberrations.
c) exploits semantic similarities between annotation sets to characterise (rare) Mendelian diseases

F. Data Integration
1. At the core of Monarch approach stands the concept of data integration via a multi-step approach
   a) Ingest data through mapping into Resource Description Framework (RDF) graphs
   b) integration od data from other ontologies in the Open Biological Ontologies set
   c) build relationships of data elements from the Relationship Ontology
   d) aggregate into a unified ontology (knowledge graph)
   e) provide API for additional services to query and operate on the graph

G. Matchmaker Exchange
1. One of the interfaces is an API that given a patients' genotypic or phenotypic profile helps to discover:
   a) a cohort of similar patients
   b) existing data on human disease etiology
   c) existing data on model organisms
2. Needs to solve a number of administrative problems in addition, including
   a) providing legal constraints for re-identification
   b) authorisation and authentication of users
   c) policies for informed consent
   d) etc…



[Mungall et al., 201