Gene Expression Data

DNA and RNA

- 4 primary nucleotides of DNA: A *(adenine)*, C *(cytosine)*, G *(guanine)*, T *(thymine)*
- *T pairs with A, and C pairs with G*
  - *In RNA we have A, C, G but T is replaced with U (uracil)*
  - *C in DNA is transcribed into G in RNA, G in DNA is transcribed into C in RNA etc*
  - *U is less stable than T — it is suspected than DNA evolved from U to T whereas RNA did not have the same selection pressure*
- *Pairing of bases provides backups while the DNA is stored as a double helix, and enable easy reproduction of the cell as when it divides into 2 cells each gets half the helix and can then recreate the other half from this (assuming no mutations happen while the DNA is unzipped…)*
- Each amino acid is coded for by a sequence of 3 RNA letters (a codon) (or equivalently DNA letters) — a whole sequence of amino acids makes a protein
  - *In principle there could be $4^3 = 64$ amino acids coded for, but there are actually only 20*
  - *There are 3 stop signals (TAA, TAG, TGA)*
  - *Start signals double up as codes for amino acids, so are at the start of every protein (other than those for which post-transcription modification splices it out of the mRNA)*
  - *The 61 only code only code for 20 amino acids as some amino acids have multiple codes for robustness against mutations*

Standard codon table

1st base	2nd base								3rd base
	U		C		A		G		
U	UUU	(Phe/F) Phenylalanine (np)	UCU	(Ser/S) Serine (p)	UAU	(Tyr/Y) Tyrosine (p)	UGU	(Cys/C) Cysteine (p)	U
	UUC		UCC		UAC		UGC		C
	UUA	(Leu/L) Leucine (np)	UCA		UAA	Stop (*Ochre*) *[note 2]	UGA	Stop (*Opal*) *[note 2]	A
	UUG ⇒		UCG		UAG	Stop (*Amber*) *[note 2]	UGG	(Trp/W) Tryptophan (np)	G
C	CUU		CCU	(Pro/P) Proline (np)	CAU	(His/H) Histidine (b)	CGU	(Arg/R) Arginine (b)	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	(Gln/Q) Glutamine (p)	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	(Ile/I) Isoleucine (np)	ACU	(Thr/T) Threonine (p)	AAU	(Asn/N) Asparagine (p)	AGU	(Ser/S) Serine (p)	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	(Lys/K) Lysine (b)	AGA	(Arg/R) Arginine (b)	A
	AUG ⇒	(Met/M) Methionine (np)	ACG		AAG		AGG		G
G	GUU	(Val/V) Valine (np)	GCU	(Ala/A) Alanine (np)	GAU	(Asp/D) Aspartic acid (a)	GGU	(Gly/G) Glycine (np)	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	(Glu/E) Glutamic acid (a)	GGA		A
	GUG ⇒		GCG		GAG		GGG		G

The central dogma

- **The Central Dogma of molecular biology: DNA is transcribed into RNA and RNA is translated into proteins** — DNA creates new DNA through replication, but otherwise this is a one way process — eg information in proteins is locked up so changes by prions do not affect DNA
    - *Viruses often replicate their RNA*
    - *Retroviruses such as HIV reverse transcribe RNA into DNA, but in most organisms the central dogma essentially holds*
- **DNA is transcribed into mRNA which becomes mature mRNA through post-transcription modification which is translated into proteins**
- DNA is “designed” for long term data storage — whereas RNA is a temporary data form while DNA is being “executed”

Protein coding DNA

- DNA and RNA are each read linearly end-to-end like a Turing machine tape, but most of it is not relevant to the cell in question (in fact plenty of it is completely left over from less evolved animals and never expressed in humans)
- **Exons = sections of the mRNA (or equivalently the corresponding sections of the DNA) that survive post-transcriptional modification (and so are translated to produce the protein)**
- **Introns = sections of the mRNA (or equivalently the corresponding sections of the DNA) that are spliced out by post-transcriptional modification (and so are not translated to produce the protein)**
- **Alternate transcripts can arise through alternate splicing: Evolution makes use of the same DNA for multiple proteins by varying the post-transcription modification** to vary what are exons and introns

Regulatory DNA

- Only coding regions of DNA get transcribed into RNA
- Non-coding regions of DNA are often regulatory sequences rather than junk
- Regulatory sequences control whether a coding region gets transcribed, and which alternate transcripts get produced in post-transcriptional modification

Gene expression data

- **Gene expression data refers to sequencing RNA not DNA**
- **It is more clinically relevant to see the extent to which genes are being transcribed** instead of what genes there are available to be transcribed
- DNA sequencing does occur, but in this course we focus on RNA sequencing

RNA Sequencing

- **We can only sequence a few hundred base pairs at a time — we have to fragment the RNA to be able to sequence it, then try to computationally combine the reads having lost the information of what order the fragments were in** (although some fragments are reused in the genome meaning we can't do this perfectly) — this is way easier to do by aligning (find the largest substring) to a reference genome, than to do as a de-novo assembly (by building up possible chains by merging overlaping reads) to create a reference genome
    - This is complicated by the fact that only the exons are in the RNA reads but the reference genome is for DNA and so our sequence is broken up in the reference by introns at unknown points
- STAR (Spliced Transcripts Alignment to a Reference) is a fast algorithm (at the cost of RAM usage) for aligning RNA-seq reads into a sequence

- FASTQ is the de facto standard file format for sequencer readouts
    - Uses ASCII characters (!-K) to efficiently (ie as a single character in the text file) encode the quality score for each read character — phred quality score (rarely exceeds 60) = −10 log_10(p) where p is the error probability
- Through an alignment algorithm, a FASTQ file can be converted into a sequence, a SAM (Sequence Alignment Map) file — a SAM can be compressed into a BAM (binary alignment map)

Differential expression analysis

- Most genes have the same expression data between samples — this is helpful as we can safely normalise overall (this is especially important when harmonizing data across experiments as sequencing depth etc may vary) before we attempt to identify the minority of genes which do vary with some label
- **Expression data has too high variance to fit well to a Poisson distribution (which requires that mean = variance) so we use a negative binomial distribution instead** — in particular there are a lot of 0 counts
- **Dispersion = variance/mean**
  - **Longer genes are more likely to be read due to the sampling nature of sequencing, so we have to normalise dispersion based on gene length**
  - **Due to the sampling nature of sequencing, gene expression data is heteroscedastic — variance depends <u>non-linearly</u> on mean**
- **MA plot: y-axis = M = log2(expression in group 1/expression in group 2)** = log2(expression in group 1) − log2(expression in group 2). **x-axis = A = ½ * log2(expression in group 1 + expression in group 2)**
  - **M = log ratio of expression levels, A = average log expression level**

Overrepresentation analysis

- **Overrepresentation analysis (ORA) models data using a hypergeometric distribution (balls in a bag approach)**
	Differentially expressed in our label	**Not differentially expressed in our label**
In cluster	**a**	**b**
Not in cluster	**c**	**d**

  The probability that this occurred by chance rather than because that cluster of genes (eg biological pathway) is actually differentially expressed is ((a+b)!(c+d)!(a+c)!(b+d)!)/(a!b!c!d!(a+b+c+d)!)
- Assumes more independence than is biologically plausible — some pairs of distinct clusters are more similar than others

Gene set enrichment analysis

- **Gene Set Enrichment analysis (GSEA): Create a ranked list of correlations ρ_i between gene expression and class label. Enrichment score of a set of genes S = maximum deviation from the origin encountered in the walk that goes through the ranked list in order and takes a forwards step** $\rho_i/(\sum_j \rho_j)$ **when the current gene i is in S and a backwards step** of size 1/(number of genes not in S) **when the current gene is not in S**
- **By shuffling the labels and recomputing the enrichment score, we get a null distribution** — in the null distribution, the correlation between class label and gene has been destroyed but the correlations between genes will still be present, thus by using the enrichment score in this as a baseline we can isolate in our data how responsible the genes are for the label specifically
  - p-value = number of trials out of 1000 trials of shuffling in which the enrichment score is higher in the shuffled data that in the actual data
- **Unlike overrepresentation analysis, does not make any assumption about the distribution** as we estimate the null distribution from the data instead

NCBI eUtils

Bulk download vs APIs

- **Advantage of bulk download (ie disadvantage of APIs): Having a local snapshot of the data at the start of your experiment means better repeatability**, this is not to say that you shouldn't try to stay up to date but at least you are in control and can easily track changes
- **Disadvantage of bulk download (ie advantage of APIs): Wastes resources of client and server to download extraneous data**, especially as have to check for updates periodically

Harmonization

- Mapping = converting from one schema to another
- Harmonization = combining separate datasets, such as by mapping between them

- **Accession number = synthetic primary key**

NCBI eUtils

- **eSearch**
- **ePost — Upload identifiers (for use with history server) if we have them already instead of needing to search**
- **eSummary — Obtain internal NCBI identifiers and basic metadata from eSearch identifiers — use history feature of eSearch to do this easily**
- **eLink — Map between NCBI databases e.g. convert between identifiers in the nucleotide (mRNA) database and identifiers in the gene (DNA) database**
- **eFetch — Obtain the full database entry for identifiers — eSummary is often sufficient**

Network Modelling

Network data science

- **Laplacian matrix = degree matrix − adjacency matrix**
  - **Degree matrix = diagonal matrix where each entry is the sum of the corresponding row in the adjacency matrix**

- **Degree centrality of a node = number of neighbours of that node — high degree = central hub**
- **Betweenness centrality of a node = fraction of shortest paths between pairs of nodes in the graph that pass through that node — high betweenness = connector hub**

Clustering

- **Community (modularity) clustering: Modularity score of a clustering = number of edges within clusters − expectation of number of edges within clusters if edges were random but each nodes still had the same degree. Start with each data point in its own cluster and greedily maximize modularity score**
- **Hierarchical (agglomerative) clustering: Start with each data point in its own cluster. Repeatedly find the two most similar clusters and merge them until some stopping condition**
    - If we allow it to run until all data is in a single cluster and keep records of each step, we obtain a tree which we can cut at various heights for various levels of clustering

Network fusion

- Multi-omic data allows us to average away more noise and so pick up on weaker signals
- **Early integration: Just concatenate all the omics together** — yes it should work out in principle, but it adds complexity when in actuality omics that are very different are not going to be any better off from being available to each other straight away
- **Late integration: Process the omics individually (as would have been done before we had multi at all), then aggregate the results**
- **Intermediate integration: Look at all the data straight away, but with the purpose of specifically "fuse"ing the data into a single pseudo omic then process this individually**
- **Similarity network fusion (SNF): Create a patient similarity network (nodes are patients, edge weights are similarity scores) for each data modality. Use an iterative process to converge upon a single patient similarity network that is a fusion of all the data**
- **Neighbourhood Based Multi-Omic Clustering (NEMO): Significantly simplifies SNF by removing need for iteration, while also extending to allow data for which some samples are missing data for some modalities**

SNF: Mathematical details

- *Let $W^{(k)}$ = matrix of edge weights (similarity scores) for the kth feature. Then, the generalised degree matrix $D^{(k)}$ is still the diagonal matrix with each entry the sum of all edge weights with an endpoint at that node. Compute $P^{(k)}_0 = (D^{(k)})^{-1}W^{(k)}$*
    - *As D is a diagonal matrix, its inverse is the diagonal matrix with diagonal entries the recripcoals of those in D. Thus, it acts to normalise so that each row in P_0 will sum to 1*
- *For each node find the top k (hyperparameter) nodes most similar to it according to W, consider these to be that node's (directed) neighbours. Compute $S^{(k)}$ the matrix where each element i, j is 0 if j is not a neighbor of i and $W^{(k)}(i,j)$/(sum over l neighbour of i (including i itself) of $W^{(k)}(i, l)$)*
    - *S simply zeroes out everything in P not in the local neighbourhood, and renormalized everything that is in the local neighbourhood (so that each row still sums to 1)*
- *t := 0; do until $\forall k\ |P^{(k)}_t - P^{(k)}_{t-1}| < \varepsilon$ { for k in features $\{W^{(k)}_{t+1} = S^{(k)}\langle P^{(v)}_t \rangle_{v \neq k} (S^{(k)})^T$ where $\langle\rangle_{v \neq k}$ is the mean over all the features other than k; $P^{(k)}_{t+1} = (D^{(k)}_{t+1})^{-1}W^{(k)}_{t+1}$ where D is computed in the same way as before} t += 1}*
    - *If early stopping occurs, used features are defined as the arithmetic mean $\langle P^{(k)}_t \rangle_{\forall k}$, however maths tells us that if the system is allowed to converge in t then $\forall_{i,j}\ P^{(i)}_t = P^{(j)}_t$*
    - *Ss (local similarities) do not evolve over time, only Ps (global similarities)!*
    - *At each step we update the global similarities for each feature by the mean of the current global similarities for all the other features filtered through the local similarities for this feature*

NEMO: Mathematical details

1. *As in SNF: Let $W^{(k)}$ = matrix of edge weights (similarity scores) for the kth feature. For each node find the top k (hyperparameter) nodes most similar to it according to W, consider these to be that node's (directed) neighbours*
2. *Compute $S^{(k)}$ the matrix where each element i, j is 0 if j is not a neighbor of i and ½[$W^{(k)}$(i,j)/(sum over l adjacent to i (including i itself) of $W^{(k)}$(i, l)) + $W^{(k)}$(i,j)/(sum over l adjacent to i (including i itself) of $W^{(k)}$(j, l))] — we need to check adjacency in both directions due to the directedness of the edges*
3. *Return the arithmetic mean over all k of $S^{(k)}$. For each entry i, j only include in the mean the ks for which there is data for both i and j*

- *Yes really, apparently the iteration of global information in SNF is not necessary, an immediate mean of (albeit slightly differently normalised as it is now what we are using as the output so needs to be symmetric) local information suffices!*

Ontologies

- **A terminology is a list of controlled vocabulary: a defined list of terms with standardised definitions**
- **An ontology extends a terminology by also formalising the relationships between <u>all</u> the concepts**
- **Ontologies have the transitivity property: Any property that a node has is also a property which all its children** (and so by induction its grandchildren and so on) have
  - Think of inheritance in OOP