BIOL 365: Methods in Bioinformatics

Winter 2022 Review Notes

This course covers bioinformatics methods with an emphasis on analysis of high-throughput `-omics' data. Topics include analysis of genome-scale gene and protein expression, construction of species and gene trees from molecular sequence data, and analysis of biological systems using bioinformatics tools. Included will be practical experience with bioinformatics tools and datasets.

Table of Content

Lecture 1: Introduction to Bioinformatics

"Bioinformatics" → 1980's for analysis of biological sequence data

- biology + computer science
- Primary challenge
    - analyze the wealth of sequence data → structure, function, and evolution
- structure databases, interaction maps, and the '-omics' family (genomics, proteomics, transcriptomics, metabolomics, etc.)
- Compilation of molecular data in databases with global access

Important components of Bioinformatics

- Analysis and interpretation
- Algorithm development
- Tool development
- Information management

Evolution of Bioinformatics

- **Protein Sequencing**
    - 1955: first complete protein sequence (insulin, Ryle et al. 1955. Bioch. J. 60: 541-546)
    - 1965: ~20 proteins sequenced
    - 1980: ~1500 full sequences of proteins
    - The NCBI Refseq protein database as of Jan. 2021: 191,411,721 proteins
- **Nucleotide Sequencing**
    - Structure of DNA (Watson & Crick, 1953)
    - 1960's and 1970's – only small RNAs were sequenced
    - Development of Cloning and then later PCR greatly increased sequencing of DNA
    - Simpler techniques in sequencing and automation greatly increased ability to collect nucleotide data
    - NCBI Genbank as of Dec 2020: 221,467,827 sequences, 7.23 x 1011 bases
    - NCBI WGS as of Dec 2020: 1,517,995,689 sequences, 1.18 x 10^{13} bases

- Development of **molecular techniques** laid foundation for the "sequence revolution" of the 1980's and 1990's
- Development of **more efficient computer technology** allowed for generated sequence data to be stored and retrieved
- The combination of the above resulted in the birth of the field of **Bioinformatics**
- Another huge change has been the ongoing development of **high throughput sequencing methods**

Perspectives of Bioinformatics

- Cellular level: genome → transcriptome → proteome → cellular phenotype
- Organism level: time of development; physiological or pathological state; region of body

Questions of Bioinformatics

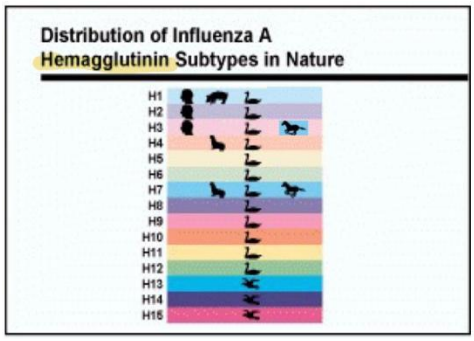
- Given a genetic sequence, what is the likely **function** of the gene or protein?
- What genes/proteins are known to be **involved in**... (a disease, a biochemical function, etc.)?
- How are a group of organisms (or genes/proteins) **related**? (phylogenetics)
- What is the **geographic/evolutionary** origin of a disease (e.g. HIV, Swine Flu)?
- Given a protein sequence, what might its **structure** look like and how is this related to function?

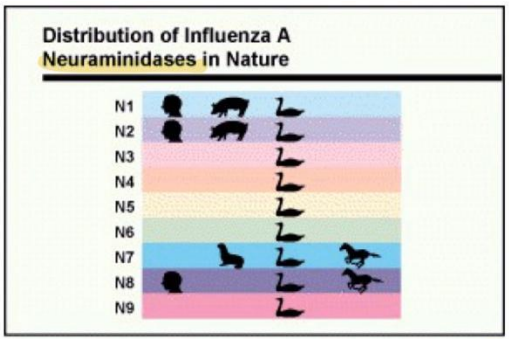
Example: H1N1

- deposition of newly sequenced genomes or nucleotide data
- retrieval and analysis of existing genome, gene, and protein data
    1. Sequence retrieval
    2. Given a sequence
        a. starting point: sequence alignments to previously characterized genes + proteins: BLAST searches
            - nucleotide vs nucleotide, translated nucleotide vs protein, similarity to other strains

Comparative protein structure modeling

- BLAST search of proteins with known structure (PDB)
    - Hemagglutinin is a major functional protein and immune target for H1N1
    - Hemagglutinin in Nature vs Neuraminidases in Nature





Phylogeny of H1N1

- Clustering and Phylogenetics
    - Clustering
        - grouping by similarity
        - assume changes are additive (changes accumulate along branches)
        - or assume changes are not additive
    - Phylogenetics
        - clustering nucleotide or protein sequences by similarity
        - more detailed than simple clustering
        - changes are additive
        - follows evolutionary model
        - can use different models and assumptions
- output is a grouping of data, often as a tree

Lecture 2: Databases

Databases

- Starting point for bioinformatic research
- Storage, sharing and describing of data + preliminary analysis through database-associated tools
- Simple to very complicated, local to international
- Databases are very large and continue to grow
- Managing much of this information is the responsibility of large consortiums (NCBI, EBI, DDBJ) → a huge industry in science
- Databases have many different structures. We will deal with:
    - **flat file databases**
        - Data (e.g., sequences) are stored as a **text file** or a collection of text files
            - flat, as in a sheet of paper
        - Easy to input, distribute, search, and retrieve data
    - **relational databases**
        - Most common type of biological database
        - Data stored within a number of tables linked together by a **shared field**, the **key**
            - a key must be unique to each record
        - Handles huge amounts of data
            - reducing data in memory
            - faster search and retrieval

Primary Data vs. Secondary Data

- **primary data** = data derived from experimental results
    - sequence data (genomic, ESTs, etc.)
    - gene expression data (microarray data, ESTs)
    - structural data (X-ray, NMR)
- **derived (secondary) data** = results utilizing primary data

    - conserved sequence motifs
    - conserved protein/domain families
    - functional annotations
    - signal peptides
    - binding/catalytic sites
    - many of these annotations are automated

Data Quality

- **redundant** vs. **non-redundant**
    - efficiency
        - the less duplication the faster the process
    - databases are screened to reduce redundancy
- Databases can be under automatic and/or manual quality control
    - UniProtKB/SWISS-Prot (highly annotated manual component)
    - UniProtKB/TrEMBL (automatically annotated)

Reliability of Databse

- Record may not always be what you are looking for
- **Computational error**
    - incorrect annotations
    - missed relationships (insufficient information extraction)
- **Human error**
    - Contributions of many different people with different accuracy standards and goals
    - Fragment of vector sequence included in sequence
    - PCR chimeras
    - Taxonomic misidentification
    - Trivial data entries
        - Example: Accession A00674: CACTAA, patented six nucleotides, unknown source/organism

        - By random chance this sequence is found in numerous sequences (since it is short). Entry is now deleted
- Although most data is accurate, errors are possible

International Nucleotide Sequence Database Collaboration (INSDC)

- Gene Bank (NCBI), ENA (EMBL-EBI), DDBJ
- each of these databases contain equivalent information (formats vary slightly)
- data is exchanged daily

Other databases

- **Swiss-Prot** - curated protein database
- **TrEMBL** - computer annotated supplement to Swiss-Prot
- **Swiss-Prot** and **TrEMBL** are available through **ExPASy**
    - **ExPASy** (Expert Protein Analysis System) contains many other useful bioinformatics tools and databases
- **Ensembl** - annotated genome browser
- **UCSC genome browser**
- **Protein structure database**

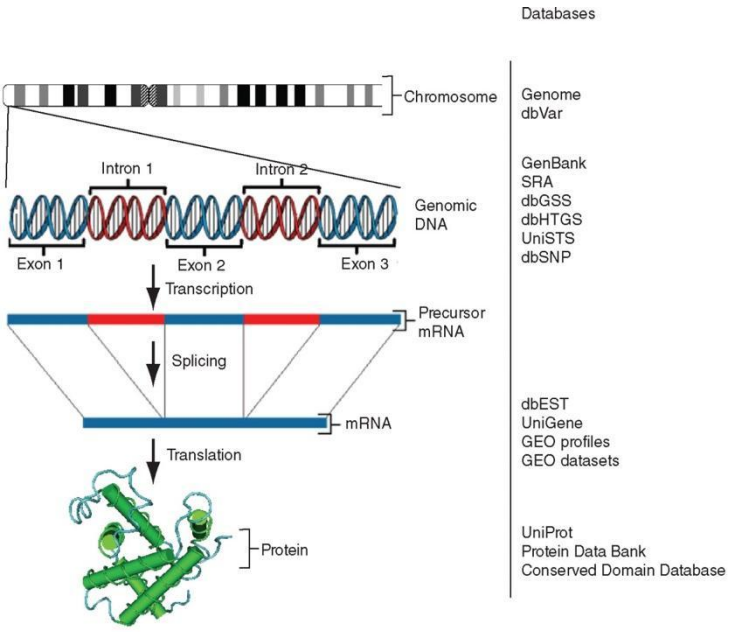
The NCBI Database

- **Type of Sequence Data**
    - A given target gene/protein typically has multiple records: **DNA, RNA (as cDNA), and protein**, and can be from numerous organisms and/or tissues
    - **Genomic and gene data (DNA)** → The target sequence can be
        - part of a chromosome
        - part of a large DNA fragment (BAC, YAC, or cosmid)
        - a gene, including introns, exons, and untranslated regions
        - **a Sequence Tagged Site (STS)**
            - small fragments of DNA, often ~500 bp, with known location within a genome.
            - Used to link genetic and physical maps.
    - **cDNA and ESTs**

        - a complete cDNA copy of an mRNA transcript
        - an EST - **Expressed Sequence Tag** - partial copies of RNA transcripts, often 300-800 bp
        - both mRNAs and ESTs can be tissue specific, identifying a gene/protein that is expressed at a certain time within a certain tissue
    - **Protein**
        - a sequenced protein (rare)
        - sequence translated from mRNA or DNA (common)
        - sequence corresponding to a known protein structure (PDB data)
- **Curated Sequence Data** - **RefSeq**
    - Best Representative Reference Sequence
    - goal: to unify sequence records and provide the most reliable sequence data
    - one RefSeq per gene per organism
    - As of January 2021, there are
        - 191,411,721 RefSeq proteins
        - 35,353,412 RefSeq transcripts
        - 106,581 Organisms in RefSeq
- **Accession Number**
    - provide a unique identifier for a specific sequence
    - accession number depends on data source
- Presented in flat file format
- **Features within Flat Files**
    - perform a biological function
    - affect or are the result of the expression of a biological function
    - interact with other molecules
    - affect replication of a sequence
    - affect or are the result of recombination of different sequences
    - are a recognizable repeated unit
    - have secondary or tertiary structure

- exhibit variation
- have been revised or corrected

- **Sequence Database in NCBI**



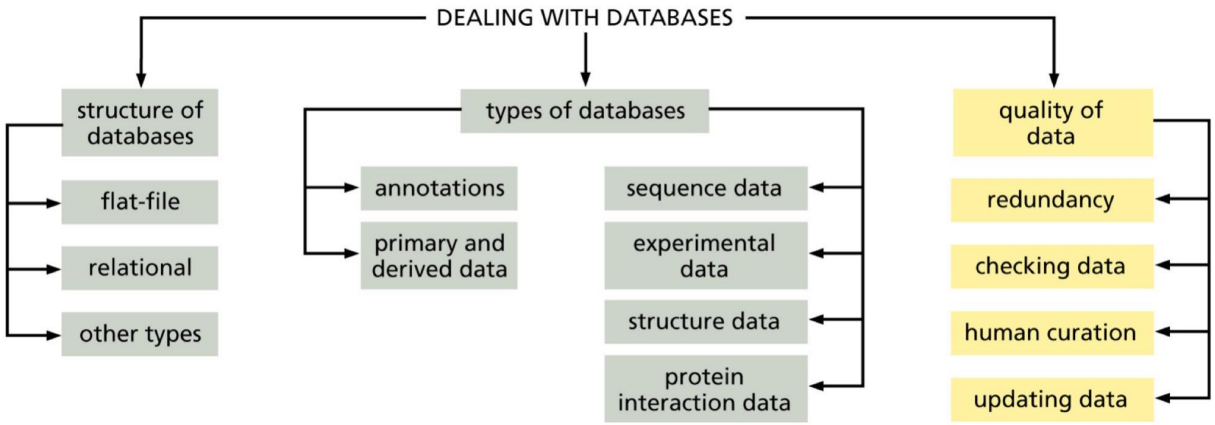
- **PubMed information Retrieval**
  - provides a gateway to **biomedical** literature
    - great for medicine, genetics, biochemistry, etc.
    - less useful for computer algorithms, astrophysics
  - PubMed includes some **out-of-scope** articles from **multidisciplinary journals (Science, Nature, PNAS, …)**
  - often provides direct links to journal articles
    - some are directly accessible, or you can access through UW library
  - can limit search by **Publication type** (e.g. review) or by **publication date** or **medical subject area** (MeSH), etc.
- **Google scholar**
  - good for an initial search as well (be careful about journal quality)
- **Controlled Vocabularies (CV) and Ontologies**
  - CV - defines **data types** (amino acid, mutation, etc)
  - Ontologies define relations: ‘is a’, ‘is attribute’, ‘part of’

- **MeSH – Medical subject headings**
    - **controlled vocabulary** indexing of PubMed articles
    - useful for refining research and using correct search terms
        - Ex: restrict to “human”: human[organism]
    - MeSH is **curated** and continually updated
    - provides general headings and subheadings
        - subheadings provide additional search terms
        - can be used to define a targeted search
    - displays a **summary** definition of the term
    - display a **tree of related items** in the subject headings
- **PubMed Links**
    - from search page, items can be added to clipboard
    - linked directly to article for some journals
    - summaries can be e-mailed
    - save citations to a file, readable by reference software e.g. EndNote
    - related articles
    - links to textbooks
- **Taxbrowser** - taxonomy search
    - use to search for **lineage** and scientific names of organisms
    - links to
        - taxonomy resources
        - genetic codes (alternate codons by species and organelle)
        - sequence data for extinct organisms
        - recent classification changes
    - provides **full lineage**, Kingdom, Phylum, …, sub-species
    - individual records provide links to **Entrez records**
        - nucleotide sequences
        - protein sequences

        - PopSet (population study data sets)

- **Finding Appropriate Database**
    - familiarity / previous use
    - major database centers (NCBI, EBI, Ensembl, etc.)
    - literature and internet
    - precedent
    - authoritative source (journal summaries)
        - nucleic acids research journal, database issue
- NAR database collection updates
    - Nucleic acids research
- **Automation, Curation, and Quality**
    - Redundancy, non-redundant databases
    - Automated data checking
    - Initial analysis is often automated
    - <u>Manually curated databases are highest quality</u>
    - All databases have some level of error
    - Error sources can be experimental, computational, inferential, or contextual

Summary:



Lecture 3. Pairwise Sequence Alignments

Importance of Sequence Alignment

- the most fundamental tool of bioinformatics
- used to determine **if two sequences are evolutionarily related** (homologous)
    - or more precisely, which **regions** may be homologous
- used to identify **common domains or motifs** between different proteins
- can uncover **patterns of conservation** and variability
- useful in **genome annotation**

Sequence Alignment

- Sequence alignment is the identification of **residue-residue** or **nucleotide-nucleotide** matches, preserving their order
    - Or stated differently: **pairwise matching** between characters in each sequence
- Need to define criteria → algorithm can choose the best alignment: **SCORE**
- **GAPS** are introduced to maximize overall score
- used to decide if sequences are homologous
    - if the alignment reflects an evolutionary relationship between two or more sequences → share a common ancestor
- sequence similarity implies **structural similarity**, indicate **related function**, and **infer evolutionary history**

Homology

- **Homologs**: sequences are from a common ancestral sequence
    - **Orthologs** - related sequences in different species that have diverged through **speciation**
        - EX: whale and horse myoglobin
    - **Paralogs** / **Paralogy** - sequences within a genome that share similarity due to **duplicate ancestral gene**
        - EX: human alpha and beta chains of hemoglobin
        - EX: Lactate dehydrogenase isoenzymes

    - **Xenologs** - relationship due to **lateral gene transfer**
        - EX: some types of antibiotic resistance in bacteria
- Homology is a '**yes or no**' question, sequences are homologous or they aren't. The degree of sequence identity (or similarity) between sequences can be used to answer this
- **Percent Identity** is the percent of exact matches within a pair of aligned sequences
- **Similarity** includes functionally similar amino acids as approximate matches
    - e.g. Valine and Isoleucine (hydrophobic) & Lysine and Glutamine (charged, hydrophilic)

Pairwise alignment

- The process of lining up two sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) for the purpose of assessing the degree of similarity and the possibility of homology
- It is possible for sequences to appear **highly similar** even if they are not homologous, e.g. as a result of **convergent evolution**. Typically these are shorter peptide sequences, repeated sequences, or low-complexity regions.

Changes in Homologous Sequences

- Changes that occur through **divergence**
- **Substitutions** (A to T, T to C etc. or amino acid changes)
- **Insertions** (one or more nucleotides/amino acids)
- **Deletions** (one or more nucleotides/amino acids)
- In regions where there is an insertion or deletion (**indel event**), **gaps** are inserted

Should You Be Aligning Amino Acid or Nucleotides?

- **protein alignments** may be more informative for **function**
    - 20 amino acids vs. 4 nucleotides (**less noise for AA**)
    - many amino acids have related **biophysical properties**
    - **codon degeneracy**: changes in the third position often do not alter the amino acid that is specified
    - **protein** sequences offer a **longer "look-back" time** (due to codon degeneracy)

        - DNA sequences can be translated into protein, and then used in pairwise alignments
    - in some cases, **DNA alignments** are more appropriate
        - to confirm the **identity** of a cDNA
        - to study **noncoding regions** of DNA
        - to study **DNA polymorphisms**
    - DNA sequences change at a faster rate than protein sequence
    - DNA may be more informative for **closely related species comparisons** or comparisons within a species
    - When in doubt, compare both

Best Alignment

- Need a way to examine all possibilities and determine which is best
- Alignment may not be unique in that several may give the same score

Global vs Local

- **Global**: attempts to align **entire** sequence
- **Local**: aligns regions with **recognizable** similarity (regions with score > similarity)

Multiple Domain Protein

- possible that only a subset of domains share homology
- in this case, a local alignment should be used, or subsets of the complete protein sequences selected
- global alignments can be useful for **aligning known domains of low similarity**, e.g. to a known structure

Multiple Sequence Alignment

- including **additional information** from other sequences can improve a given pairwise alignment

Pairwise sequence alignment methods

- can be **exact** (find optimal solution, eg. best score) → **Dynamic programming algorithms**
    - Smith and Waterman (local)
    - Needleman and Wunsch (global)

- or **Heuristic** (faster, but solution might not be optimal)
    - world or k-tuple methods (FASTA, BLAST, MegaBLAST)
- **graphical methods** for visualization (dot matrix)

Gaps

- gaps are included to obtain the best possible alignment between two sequences
- **insertion and deletion** (indel) events complicate sequence alignment
- **gap penalty** must be included for each gap introduced
    - because insertion and mutation are rare, but may affect nearby residues and nucleotides
    - simple score: match=1, mismatch=0, gap=-1
    - there may be a number of equally optimal alignments (ie: same score) between two sequences
    - Indels are rare, better alignment may be the one with fewer indels but **longer gaps**
        - Since there is no way to determine without knowing the original precursor sequence, there is no way to determine if a gap was caused by an insertion in one sequences or a deletion in another sequence
        - It is more likely that the difference in length for 2 sequences over a short region is **due to one indel than several**
        - Thus, we can bias the alignment scoring function to take this into account and gap penalty is divided into two parts
            - Gap origination penalty (G_O)
            - Length (or **extension**) penalty, G_L
            - Gap penalty = $G_O + nG_L$

Dynamic programming algorithms for sequence alignment

- Exhaustive alignment methods are not feasible
- Dynamic programming algorithms can **provide an optimal alignment between two sequences** (mathematically proven)
    - Compares every pair of characters in two sequences and generates an alignment

    - **Matches between identical or related characters are maximized** in the alignment
- **Global alignments**: Needleman-Wunsch algorithm (1970)
- **Local alignments**: Smith-Waterman algorithm (1981)

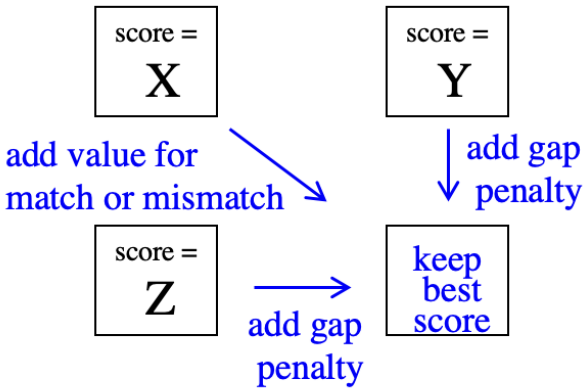
Dynamic Programming Algorithms

- break big problems into smaller ones
- calculate a score for the first several positions, and use that score as part of the score for the larger alignments.
    - Saves recalculating the entire score for each alignment

The Needleman and Wunsch Algorithm (for global alignment)

https://bioboot.github.io/bimm143_W20/class-material/nw/

- uses a **2D array** to both calculate and store values for partial sequence alignments with a scoring function.
- It is possible to get more than one optimal alignment, depending on both the sequences and the scoring function used
- each entry is the best score for the alignment up to that point
- built by filling in array from top to bottom, left to right.
- Reconstruct from **bottom right**
- **vertical** move = gap in seq1
- **horizontal** move = gap in seq2
- **diagonal** = alignment in that position



Local alignments

- global alignments depend on order of genes or domains, and local sequence
    - will not work well for large DNA comparisons or two multi-domain proteins that share only one common domain
- Local alignment recognize smaller regions of similar sequence
    - small sub-sequence within a coding region, or several regions in different order within complete sequences
- focus on regions with significant similarity
- **Smith-Waterman Algorithm** (for local alignment)

https://observablehq.com/@manzt/smith-waterman-algorithm

    - 2D array produces optimal alignments
    - recognize regions of **local similarity** within larger sequences
    - also recognize global similarity (if exists, eg. similar regions are in the same sequence order)
    - requires a **harsher penalty for mismatches**
    - similar as NW
        - new rules: **zero is the minimum value** for any entry in table
        - when tracing back through matrix to produce the alignment, start at the **maximum value** in the table (not lower right corner)
        - stop traceback when a **zero is reached**

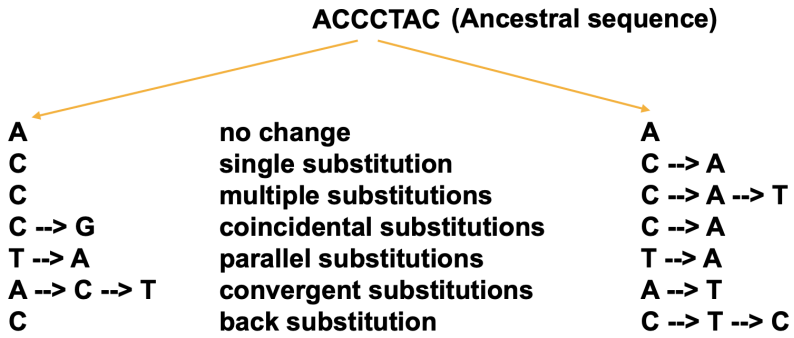
Approaches to Pairwise Alignment

- Choose two sequences
- Select an algorithm that generates a score
- Allow gaps (insertions, deletions)
- **Total score reflects degree of similarity**
- Alignments can be global or local
- Estimate probability that the alignment occurred by chance (e.g. using **E-values**)

Lecture 4: Scoring

Scoring Function

- scoring functions can be derived for nucleotides and (more often) for amino acids
- this can be viewed purely statistically:
    - the scoring function can be based on **observed frequencies of exchange**
- Or, for amino acids, it can be related to **biochemical properties** (amino acid frequency, codon frequency, physicochemistry, mutability), which largely explain observed frequencies
- For nucleotide ⟶ multiple substitutions are likely to occur and tend to average out, many scoring functions for **nucleotides database searches** are fairly **simple** (match, mismatch)



Amino Acid List

NONPOLAR, HYDROPHOBIC		
Alanine	(Ala)	A
Valine	(Val)	V
Leucine	(Leu)	L
Isoleucine	(Ile)	I
Methionine	(Met)	M
Phenylalanine	(Phe)	F
Tryptophan**	(Trp)	W
Proline	(Pro)	P

Glycine	(Gly)	G

POLAR, HYDROPHILIC		
Serine	(Ser)	S
Threonine	(Thr)	T
Cysteine	(Cys)	C
Tyrosine	(Tyr)	Y
Asparagine	(Asn)	N
Glutamine	(Gln)	Q

ACIDIC (Negative charge)		
Aspartic acid	(Asp)	D
Glutamic acid	(Glu)	E

BASIC (Positive charge)		
Lysine	(Lys)	K
Arginine	(Arg)	R
Histidine**	(His)	H

Possible Change in Amino Acid Sequence

- Observed changes in amino acid sequence due to changes in the corresponding nucleotide coding sequence
    - changes in nucleotide sequence
        - result in a change of amino acid
        - have no effect on AA (**synonymous** substitution)
        - result in **chain termination or elongation**
    - amino acids are often under functional constraint
        - the amino acid (or a similar amino acid) is important for the function or structure of the protein is under functional constraint → lower observed rate of change over time
- The observed probability of an amino acid exchange is a function of a number of factors, including:
    - functional constraint
    - frequency of amino acid
    - codon frequency and codon bias
    - mutability of amino acid
        - relative functional importance
        - relative chemical similarity
- A scoring function can be calculated from observed amino acid exchange rates in homologous proteins, effectively integrating the above factors

Substitution Matrices

- A substitution matrix contains values based on the probability that amino acid *i* mutates into amino acid *j* for all pairs of amino acids
- Substitution matrices are constructed by assembling a large and diverse sample of verified pairwise alignments (or multiple sequence alignments) of amino acids.
- Substitution matrices should reflect an **average of the probabilities for each possible amino acid change**

- Two major types of substitution matrices are:
    - based on **explicit evolutionary models** such as **PAM or GTR**
    - based on **observed alignments** to be used as **scoring functions** such as **BLOSUM**

Protein Scoring Matrices

- method of scoring is the same as with nucleotides
    - **BLOSUM62** = BLOcks SUbstitution Matrix 62%

Pairwise Alignment of Protein Sequence

- the same optimal alignment methods can be used:
    - Needleman-Wunsch (global)
    - Smith-Waterman (local)
- Typically match/mismatch scoring is replaced by **amino acid specific scores**, i.e. (L,L) = 6, (L,V) = 2, (L,D) = -4.
- Scores are based on **observed frequencies of substitution**
- **Gap penalties** are usually based on the **origination/extension model** (affine gap penalties), though linear gap penalties may also be used

Derivation of Scoring Matrices

- Scoring functions are based on a **probabilistic approach**, usually in the form of a comparison between alternate models.
- Two standard models used for comparison are
    i) a **non-random** (evolutionary) model (PAM or GTR)
    ii) a **random** model (BLOSUM)
- For any given position in a pairwise alignment, probabilities can be estimated for each model
- Models can be compared by the ratio of probabilities, or the **odds ratio**
- Many scoring systems use some variation of a **log-odds ratio**
- define $Q_{a,b}$ as the probability that an alignment of amino acids a and b occur in an **homologous sequence pair**
- define $P_{a,b}$ as the probability of **aligning amino acids**

- a and b in a random model, e.g. based on the background probabilities p_a and p_b for amino acids a and b

- the **odds ratio** for an alignment of a and b in a sequence is

$$\text{Odds ratio} = \frac{Q_{a,b}}{P_{a,b}}$$

$$\text{Odds ratio} = \frac{\text{the probability that an alignment is authentic}}{\text{the probability that the alignment was random}}$$

- If it is assumed that each column u in the **alignment is independent**, then the total odds ratio score is a product of each of the columns:

$$\text{Odds ratio (alignment)} = \prod_u \left(\frac{Q_{a,b}}{P_{a,b}}\right)_u$$

If we use log-odds, values can be added instead of multiplied:

$$S_{Total} = \sum_u log\left(\frac{Q_{a,b}}{P_{a,b}}\right)_u = \sum_u (s_{a,b})_u$$

PAM Matrices: Point-accepted mutation

- based on global alignments of **closely related proteins** (>85% sequence identity)
- An accepted point mutation is the replacement of one amino acid by another, and accepted by natural selection
- Dayhoff et al. constructed a mutation probability matrix M, where element Mij represents the probability of **amino acid j being replaced by amino acid i** over a given **evolutionary interval**
    - For the PAM1 matrix, this interval is equivalent to one change per 100 amino acids

BLOSUM Scoring

- designed for use with database search algorithms
- based on local alignments (after PAM with more data)

- derived from **BLOCKS database**, a curated database of a large number of **ungapped multiple sequence alignments**
- differences with PAM matrices:
    - contain quite dissimilar sequences (still homologous)
    - sequences are **compared directly** (no evolutionary model)
    - intent is to identify conserved regions in sequences
    - designed for database searches, **not models of evolution**

Derivation of BLOSUM Matrix

- based on observed alignments, but not extrapolated from comparisons of closely related proteins
- The BLOCKS database contains thousands of groups of multiple sequence alignments.
- BLOSUM62 is the default matrix in BLAST 2.0
- Though it is tailored for comparisons of **moderately distant proteins**, it performs well in **detecting closer relationships**
- A search for **distant relatives** might be more sensitive with a different matrix, but in general **BLOSUM62 is the best matrix to use for initial searches**
- One potential issue is that similar sequences in the training set will bias results. To address this, sequences were **weighted using similarity thresholds**
    - Similar sequences greater than a given threshold (80%, 62% etc.) were **clustered into a single sequence**.
    - e.g. for BLOSUM62, no two (averaged) sequences used have more than **62% identity**
        - WHY? $\rightarrow$ 62% works the best, as it balances size and evolutionary distances

Derivation of BLOSUM Matrices

Given a filtered database of homologous sequence alignments, the database can be reduced to a set of amino acid pair frequencies f_{ij}.

Observed probability for a given pair: $q_{ij} = \frac{f_{ij}}{\sum_{i=1}^{20}\sum_{j=i}^{20} f_{ij}}$

Given background probabilities p_i and p_j for each amino acid, the probability of a random match is:

$$e_{ii} = p_i p_j \quad \text{if } i = j$$
$$e_{ij} = p_i p_j + p_j p_i = 2 p_i p_j \quad \text{if } i \neq j$$

Derivation of BLOSUM Matrices

Odds ratio = $\frac{\text{the probability that an alignment is authentic}}{\text{the probability that the alignment was random}}$

Log odds ratio = $\log_2 q_{ij} / e_{ij}$

Score in BLOSUM matrix = S(i,j) = $2 \log_2 q_{ij} / e_{ij}$

where 2 is a scaling factor and values are rounded to the nearest integer.

Which Matrix Should We Use?

- The optimal matrix depends on the comparison being made
- **BLOSUM62** is the best default matrix for **database searching**, in that is optimal for sequences that are moderately distant, and also performs well on closely related (easy) sequences
- for more **divergent** comparisons, other matrices could be tried
    - For BLOSUM: high similarity = high threshold

BLOSUM 80	BLOSUM 62	BLOSUM 45
PAM 1	PAM 120	PAM 250
Less divergent	⟷	*More divergent*

Rat vs. mouse — Mouse vs. yeast

- Realistically, a better option is to use a more powerful method (such as multiple sequence alignments, profiles, or HMMs)

Gaps in Scoring Functions

- for good or bad, gap penalties were **NOT** initially derived as rigorously as scoring functions
- a **heuristic approach** is used, i.e. **default gap penalties** are chosen because they appear to produce good results
- we have used **linear gap penalties** in alignment algorithms:

$$g(n_{gap}) = - n_{gap}G, \text{ where G is gap penalty}$$

- **Affine gap penalties**, which have separate origination (G_O) and length (G_L) penalties, are used in most algorithms

$$g(n_{gap}) = - G_O - n_{gap}G_L$$

THE RESULT OF THE ALIGNMENT DEPENDS ON THE ALGORITHM USED, THE SCORING MATRIX, AND GAP PENALTIES

Lecture 5: BLAST

Database Searching

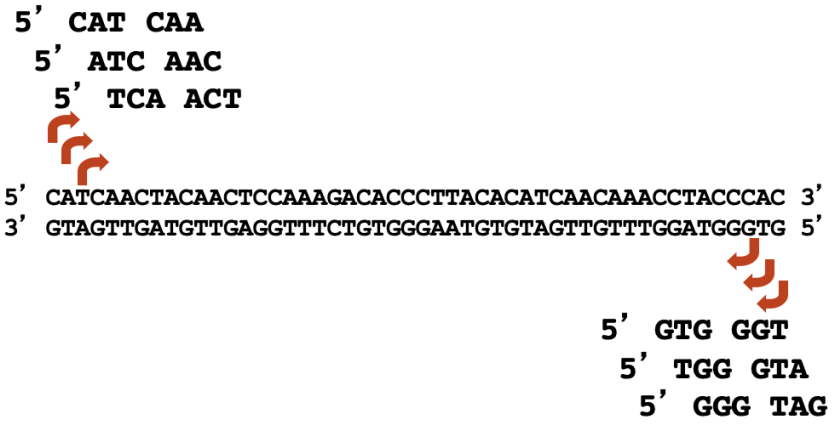
- Sequence similarity alone is not informative, we often want to **deduce function**
- BLAST is a set of algorithms to compare a query sequence against all sequences in a database
    - Similarity is measured by **aligning two sequences**
    - Each comparison is given a **score** reflecting degree of similarity, where (higher scores → greater similarity)
    - The chance (**expectation value**) that the match could have occurred as a random hit is estimated (smaller E value → more significant, similar top-values)
        - **E value**: number of random alignments expected to have a score equal or better than the query/subject alignment
        - Remember similarity does not always equal function!
- Sequentially align a query sequence to each subject sequence in a database
- Results are reported as a **ranked list** with scores and statistics
- **'Heuristic' (approximate) methods** are required due to the huge size of most databases. These are faster than optimal methods such as Smith-Waterman but are not guaranteed to find all results or the best result.
- Steps: BLAST program → Enter sequence data in FASTA or accession number → Choose Database → Adjust Parameter

Why Use BLAST

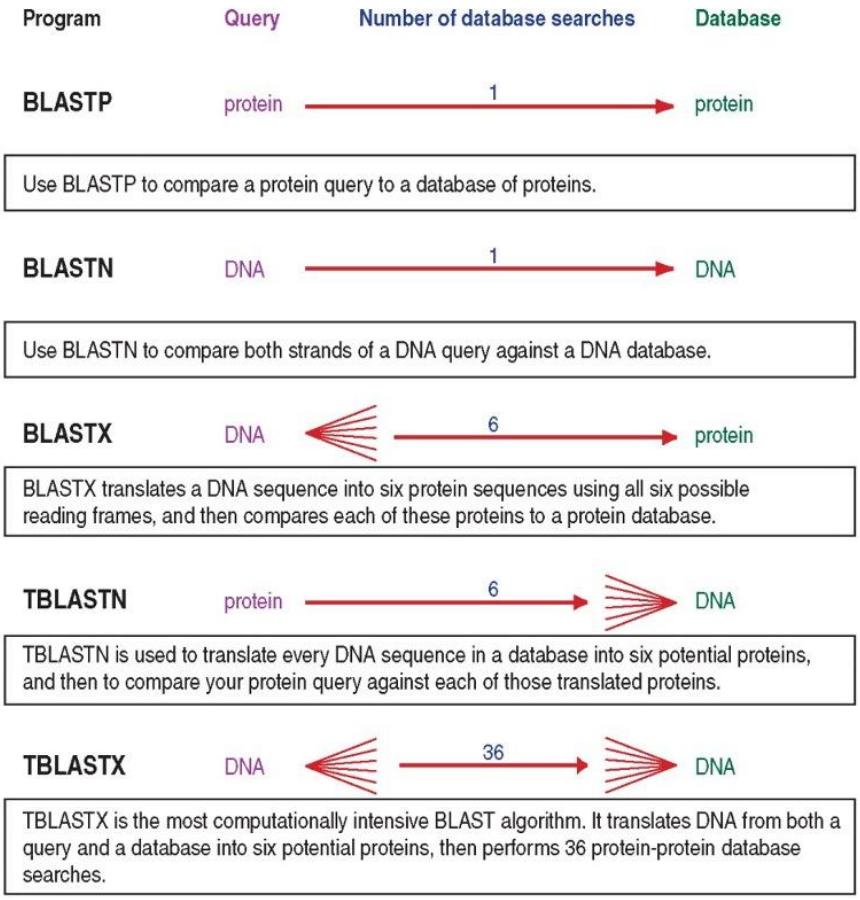
- **BLAST (Basic Local Alignment Search Tool)**
- allows rapid sequence comparison of a query sequence against a database. The BLAST 2.0 algorithm is fast, accurate, and web-accessible.
- **Applications include**
    - identifying **orthologs** and **paralogs**
    - discovering **new genes** or proteins
    - discovering **variants** of genes or proteins

- investigating **expressed sequence tags** (ESTs)
- exploring **protein structure** and function

DNA encodes for six protein sequence



BLAST Programs



Choosing a Database

- **Nucleotide Database**
    - Human genomic + transcript
    - Mouse genomic + transcript
    - nr = non-redundant nucleotide collection (most general database)
    - refseq_rna
    - refseq_genomes
    - 16S ribosomal RNA (bacteria and archaea)
    - est (expressed sequence tags)
    - htgs (high throughput genome seq.)
    - Human RefSeqGene
    - + others
- **Protein Database**
    - nr = non-redundant protein collection (most general database)
    - refseq_protein
    - Swissprot
    - patented sequences
    - PDB proteins (also in nucleotide db)
    - Metagenomic proteins
    - + others

BLAST: optional parameters

- choose the organism to search → e.g. restrict results to a taxonomic group
- turn **filtering** on/off
    - Low complexity regions can result in false positive hits. These can be filtered out.
- change the **substitution matrix**
- change the **expect (e) value cutoff**
- change the **word size**
- change the **output format**

Formatting Parameters

- Alignment view, Descriptions, Alignments...

BLAST Result

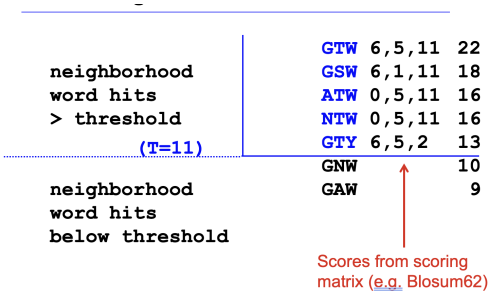
- Search ID (temporary)
- query
- database
- program
- taxonomy report
- Multiple alignment

Comments on BLAST Searching

- **protein alignments** are more useful for **inferring structure and function** from sequence
- look at the scores
    - E > 1 is likely to indicate a random alignment
    - E < 0.05 is often used to suggest **biological significance**
- low complexity regions can be similar without being related
    - usually blocked out by BLAST
- significant similarity most often means **homology**, but homology does always result in sequence similarity
- % identity = % **identical residues**
- % similarity = % **similar residues** (e.g. I <--> V)
- homology is still a yes or no question

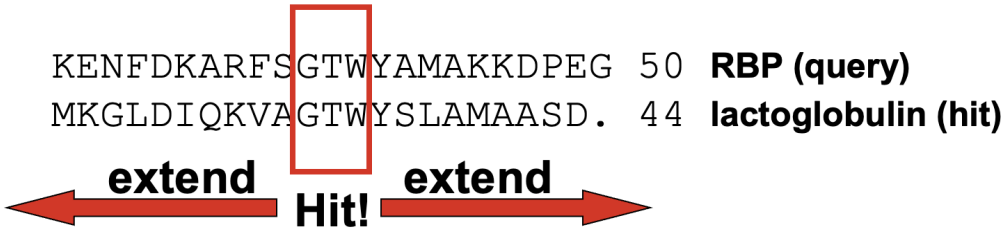
BLAST Algorithm

- **Phase 1**: compile a list of **word pairs** (w=3 in the example below) above **threshold T**



    - BLAST 2.0 uses default word size of 6 to reduce noise

- **Phase 2:**
    - Scan database for hits
    - **Index locations** of hits (uses a 'hash table')
    - Perform **gap-free extensions**
    - Then perform **gapped extensions**



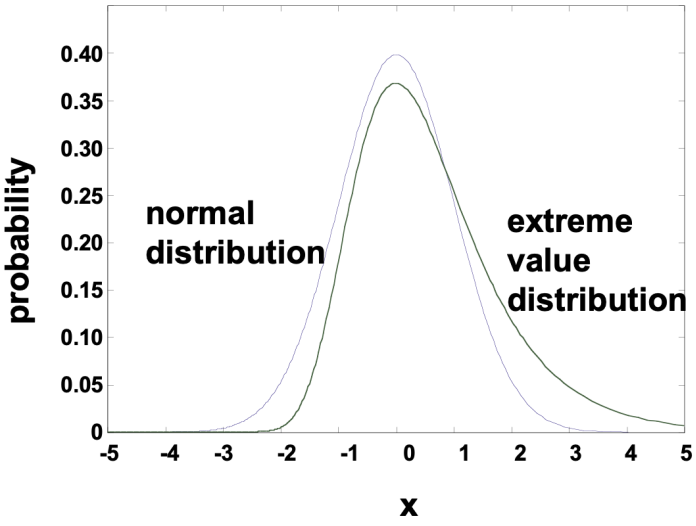
- **Phase 3**
    - Tidying up…
    - Calculate locations of **insertions**, **deletions**, matches
    - Apply **composition-based statistics**
    - Generate gapped alignment

Sensitivity and Search Speed

- High sensitivity = lower speed = lower T and/or large w
- Low sensitivity = higher speed = larger T and/or small w

Expect Value

- It is important to assess the statistical significance of search results.
- For **local alignments** (including BLAST search results), the scores follow an **extreme value distribution (EVD)** rather than a normal distribution.
    - Given two **random unrelated sequences**, A and B, the scores of random, ungapped alignments of a given length will be approximately normally distributed.
    - The highest scoring pair of these local alignments is the **maximum value** (extreme value) within the sampled normal distribution.
    - Aligning sequence A to several thousand random sequences B_i, the distribution of maximum scores (A,B_i) follows an **Extreme Value Distribution**



How to interpret a BLAST search: expect value

- The **expect value E** is the number of alignments with scores **greater than or equal to score S** that are expected to occur **by chance** in a database search.
- An E value is related to a **probability value p**, where

$$p = 1 - e^{-E}$$

- For values of **E and p < 0.05, E ≈ p**.
- Put another way,
  - E is the **expected number** of random hits with **score ≥ S**, and
  - p is the **probability** of one or more random hits with **score ≥ S**.
- **Very small E values** are very **similar to p values**.
  - E values of above 1 are far easier to interpret than corresponding p values (as large Es have similar P values).
- Given the cumulative distribution function of the extreme value distribution, the probability of a **random** score S being greater than or equal to a given threshold score x is

$$P(S \geq x) = 1 - \exp(-\exp[-\text{lambda}\ (x-u)])$$

- It was shown by Karlin and Altschul (1993) that the **parameter u** (the peak of the distribution) can be expressed as **a function of lambda**, the **size of the query sequence** (m) and **database** (n), and a scaling term K:

$$u = \ln (Kmn / lambda)$$

- AND

$$E = Kmn\, e^{-\text{lamba } S}$$

- lambda, K are the **Karlin Altschul statistics**, where **lambda** is a constant related to the **scoring function**, and K is a scaling factor related to the search space
- E increases linearly with m and n (**search space terms**)
    - larger search size = more random hits
- The value of **E decreases exponentially** with **increasing S**
    - higher S values correspond to better alignments
    - Very high scores correspond to very low E values.
- The **S value** for aligning **a pair of random sequences** must on average be **negative**
    - Otherwise, long random alignments would acquire large scores
- For **E=1**
    - one match with a similar score is expected to occur by chance <u>within a random distribution</u>
    - For larger or smaller databases, you would expect E to vary accordingly

Raw Score and Bit Scores

- There are two kinds of scores:
    - **raw scores** (calculated from a substitution matrix)
    - **bit scores** (normalized scores)
        - Bit scores are **comparable between different searches** because they are normalized to account for the use of different scoring matrices and different database sizes

$$S' = \text{bit score} = (lS - lnK) / ln2$$

- The E value corresponding to a given bit score is:

$$E = mn\, 2^{-S'}$$

- Bit scores allow you to compare results between different database searches, even **using different scoring matrices**

Lecture 6: More BLAST

Variations on BLAST

- Other databases can present BLAST results differently
- The **Ensembl** database is focused on **genomes** and can return searches showing **genome location**
    - Text output from the Ensembl database search

Evaluating BLAST Significance

- a **true positive** is defined as a **database match that is homologous** to the query sequence
    - e.g. descended from a common ancestral gene or protein
- Homology is inferred from **sequence similarity**, supported by **statistical evaluation** of the results (e.g. $E < 0.05$)
    - However, statistical results should not be relied upon **exclusively**
        - High E could mean homology; low E could not mean homology
    - E values are dependent on **sequence length**, **database size**, and **scoring function** used
    - **biological information** - e.g. structure, function, and multiple comparisons - can provide further evidence
        - proteins (or domains) are about the same size
        - share a common motif (GXW, lipocalin)
        - common to multiple sequence alignment
        - biochemical similarities
        - similar structure (possible to compare in this case)
        - reciprocal BLAST search returns similar matches

Dealing with too many results

- restrict to **RefSeq** (best representative sequence)
- restrict search space **by organism**
    - fewer but higher scoring hits

- search on a **subset** of the sequence
    - search for a domain if boundary is known
- change the scoring matrix (?)
- change **Expect value** (**lower** number => less hits. Note that this doesn't change # of significant hits → significant is E<= 0.05)

Dealing with too few results

- Some genes/proteins do not have significant database matches, e.g. many microbial proteins. Some strategies to improve results:
    - restrict the search space to a **given organism** (returns better **E-values**)
    - alternately, **remove database restrictions** in case closer matches exist
    - **raise** the **Expect value**: 10, 100, ... (does not change significance)
    - **change scoring matrix** to lower BLOSUM number, or higher PAM number
- use a **more sensitive approach**
    - multiple sequence alignments
    - PSI-BLAST
    - HMMs

Remote Homologies

- Depending on the particular research question being asked, the **goal** may be to identify
    - **remote homologs** of a gene or protein of interest
    - finding members of **a family of distantly related proteins**
    - recognition of a **remote homolog** that has a known structure
    - recognition of **all homologs** in a given species
    - discovery of **novel genes**/proteins
    - recognition of **conserved (functionally important) residues**
- If a given sequence has a possible (but remote) relation, **additional sequences** may be used to strengthen this

Position Specific Scoring Matrices (PSSM)

- In contrast to the general purpose PAM and BLOSUM matrices, **PSSMs** are specific to a **given multiple sequence alignment**, and are often **constructed from sequences homologous to an input query sequence**
- This approach captures **patterns of conservation and substitution** within a family of related proteins
- **position specific amino acid frequencies** are used to refine the scoring function

Position Specific Iterated BLAST - PSI-BLAST

- The purpose of PSI-BLAST is to look deeper into the database for matches to your query protein sequence by **employing a scoring matrix that is customized to your query**.
    - based on multiple sequence alignments
    - residue conservation integrated into scoring
- Steps:
    - [1] Select a query and search it against a protein database
    - [2] PSI-BLAST constructs **a multiple sequence alignment** then creates a "profile" or **specialized position-specific scoring matrix (PSSM)**
    - [3] The PSSM is used as a query against the database
    - [4] PSI-BLAST estimates statistical significance (E values)
    - [5] Repeat steps [3] and [4] iteratively (typically 4-5x). At each new search, **a new profile** is used as the query.
        - search is conducted using PSSM instead of initial query
        - E values are recalculated, and new hits returned based on PSSM

How is PSSM constructed

- for a query of length L, a PSSM of dimensions **L x 20** is created
    - Later versions expand this to include gaps
- in iteration #1, rows of the PSSM are essentially the **same as in the default matrix** (e.g. BLOSUM62)
- in later iterations, the **frequency of each amino acid at each position in the alignment is weighted** and contributes to the score

- **<u>redundant sequences (<~94% ID)</u> are listed but not used**
- A multiple sequence alignment →used as input to build a 'profile'
    - As with other BLAST scoring functions, a **log-odds** method is used
    - Instead of a generalized matrix, different scores are calculated for each position in the sequence or profile

$$m_{u,i} = \frac{1}{\lambda} \log \frac{q_{u,i}}{p_i}$$

    - Here, $q_{u,i}$ is the adjusted **observed frequency of residue i at position u** within the multiple sequence alignment (MSA), and p_i is the **background frequency** of residue type i
    - The frequencies $q_{u,i}$ are adjusted for missing data: $q_{u,i}$ is a linear combination of the observed frequencies in the MSA and 'pseudocount' frequencies of amino acids used in the BLOSUM matrices. lambda is a scaling factor.
- **More iteration = more results** (you are brings in stuff that might be ignored during the first iterations)

Iteration using PSI-BLAST

- iteration 1
    - is almost **identical to a BLAST search** may be slight differences, as PSI-BLAST corrects for amino acid composition
- iteration 2, a position specific scoring matrix is used to score alignments
    - in a example, **score has increased**, % id slightly less, sequence length increased, fewer gaps in the alignment
- iteration 3, the PSSM is further refined
    - score increases further, % id slightly less, fewer gaps, and lower E value (greater significance)
- Overall: higher bit score, lower E value, slightly lower identity, fewer gaps

PSI-BLAST: Performance assessment

- PSI-BLAST results were evaluated using a **database** in which protein structures have been solved and all proteins in a group share < 40% amino acid identity
- At a **false positive rate (FPR)** of 1/100,000, 23% of homologous structures were detected
- At FP ~ 1/1000, 44% of homologs were detected

Corruption of PSI-BLAST

- PSI-BLAST is useful to detect **weak but biologically meaningful relationships** between proteins
- The main source of false positives is the **spurious amplification of sequences not related to the query**
    - For instance, a query with a coiled-coil **motif** may detect thousands of other proteins with this motif that are not homologous
- Once even a single spurious protein is included in a PSI-BLAST search above threshold, **it will not go away**
- **Corruption** is defined as the **presence of at least one false positive alignment** with an E value **< 10^{-4}** after **five iterations**
- Three approaches to stopping corruption:
    - [1] Apply **filtering** of biased composition regions
    - [2] Adjust **E value from 0.005** (default) to **a lower value** such as E = 0.0001
    - [3] **Visually inspect the output from each iteration**. Remove suspicious hits by unchecking the box

Searching Pre-existing PSSMs - RPS-BLAST

- **RPS-BLAST** is the inverse operation to PSI-BLAST
    - [1] a single sequence is selected as a query
    - [2] The query sequence is compared to an **existing database of PSSMs** for known protein domains
    - [3] A BLAST-like procedure is used to **align the query to PSSMs** in the database and generate an alignment score
    - [4] RPS-BLAST estimates statistical significance (E values)

        - No iteration is required, PSSMs are already constructed.
- RPS-BLAST is the tool used to search the **Conserved Domain Database**, and is done by default with any **BLASTp** search. The search time is faster than BLASTp as the database is considerably smaller

Delta-BLAST

- **Delta-BLAST** combines PSI-BLAST and RPS-BLAST to identify remote homologs:
    - [1] a single sequence is selected as a query
    - [2] The query sequence is **compared to PSSMs in the Conserved Domain Database**
    - [3] A new PSSM for the whole protein is **constructed from a multiple alignment of CDD hits**
    - [4] The new PSSM is used to search the sequence Database
    - [5] matches are returned and statistical significance is estimated (E values)
        - No iteration is required as CDD hits are already constructed PSSMs.
- DeltaBLAST is a **faster version of PSI-BLAST** in many circumstances

Lecture 7: Genome

Overview of genome analysis

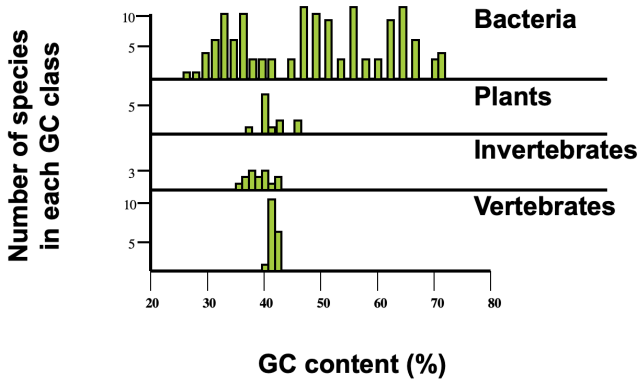
- Selection of genomes for sequencing
    - Based on criteria such as:
        - genome size (some plants are >>> human genome)
        - relevance to human disease (or other disease)
        - relevance to basic biological questions
        - relevance to agriculture
        - biological diversity
        - cost
- Sequence one individual genome, or several?
    - Initially, genome centers often targeted one chromosome from an organism
        - High throughput sequencing technology has made sequencing much more accessible
    - For viruses, thousands of isolates may be sequenced (to track evolution)
    - **Sequence variation** (SNPs, Copy number variation) may need to be considered
- How big are genomes?
    - Viral genomes: 1 kb to 350 kb (Mimivirus: 800 kb)
    - Bacterial genomes: 0.5 Mb to 13 Mb
    - Eukaryotic genomes: 8 Mb to 686 Gb
- How was a genome sequenced
    - 20 Genome sequencing centers contributed to the public sequencing of the human genome.
    - Many of these are listed at the **Entrez genomes website**
        - Sequencing has become much more accessible and now it is possible for individual labs to collect sufficient data for **draft genome** assembly.
    - **Full genome** assembly is more **difficult** as gaps need to be specifically targeted
        - error rate can accomplice genome sequencing

- There are two main strategies for sequencing genomes
    - **Whole Genome Shotgun** (from the NCBI website)
        - An approach used to decode an organism's genome by **shredding it into smaller fragments** of DNA which can be sequenced individually.
        - The sequences of these fragments are then ordered, based on overlaps in the genetic code, and finally reassembled into the complete sequence.
        - The 'whole genome shotgun' (WGS) method is applied to the **entire genome all at once**.
    - **Hierarchical shotgun method**
        - An ‘older’ method (early 2000’s)
        - Regions within a genome (such as a specific chromosome) are isolated and targeted for sequencing.
        - The full genome is obtained from the collection of targeted sequencing efforts
        - The sequence may be drafted or a finished sequence.
- When has a genome been fully sequenced?
    - An initial goal was to obtain **five to ten-fold coverage**
        - now 30x or more for short read technologies.
    - **Draft sequence**: clone sequences may contain several regions **separated by gaps**. The true order and orientation of the pieces may not be known
    - **Finished sequence**: a clone insert is contiguously sequenced with high quality standard of **error rate 0.01%**. There are usually **few gaps** in the sequence.
- Repositories for genome sequence data
    - Raw data from many early genome sequencing projects are stored at the trace archive at **NCBI or EBI**
    - More recent efforts include **BioProject** within NCBI and **Genome On Line Database (GOLD)**

- Genome annotation
    - Information content in genomic DNA includes:
        - repetitive DNA elements
        - nucleotide composition (GC content)
        - protein-coding genes, other genes

GC content varies across genomes

- Ex: High temp = more GC, as GC are more stable



General features of the eukaryotes when compared to prokaryotes

- eukaryotes include many multicellular organisms, in addition to unicellular organisms
- eukaryotes have [1] a membrane-bound nucleus, [2] intracellular organelles, and [3] a cytoskeleton
- Most eukaryotes undergo sexual reproduction
- The **genome size** of eukaryotes spans a **wider range** than that of most prokaryotes
- Eukaryotic genomes have a **lower density of genes**
- Prokaryotes are haploid; eukaryotes have **varying ploidy**
- Eukaryotic genomes tend to be organized into **linear chromosomes** with a centromere and telomeres.

C Value Paradox

- Why do eukaryotic genome sizes vary?
- **C value** = haploid genome size

- The range in C values does not correlate well with the **complexity of the organism**. This phenomenon is called the C value paradox.
- The solution to this "paradox" is that genomes are filled with large tracts of **noncoding, often repetitive DNA sequences that don't code for protein**, but still might have functional roles in **chromosome structure or gene regulation** .
- The haploid genome size of eukaryotes, called the C value, varies enormously.
- Small genomes include:
    - Encephalotiozoon cuniculi (a protozoan parasite), 2.9 Mb
    - A variety of fungi, ~10-40 Mb
    - Takifugu rubripes (pufferfish), 365 Mb (similar number of genes as other fish or as the human genome, but 1/10th the size)
- Large genomes include:
    - Pinus resinosa (Canadian red pine), 68 Gb
    - Protopterus aethiopicus (Marbled lungfish), 140 Gb
    - Amoeba dubia (amoeba), 690 Gb

Eukaryotic genome organization

- Organized into chromosomes
    - Can be visualized using DNA binding stains such as **Giemsa or Wright's stains**.
    - These stain the length of each chromosome and exhibit a **characteristic banding pattern**
- The **karyotype** refers to visualization of stained chromosomes, typically in **metaphase**.
- Visible features include **telomeres** (chromosome ends) and **centromeres** (can be located near the middle the chromosome or near the telomeres)

Human genome project

- The human genome is approximately **3 billion base pairs**
- We have a comparable number of genes to other animals and many plants. Humans have ~21,000 genes; sequenced plant and animal genomes typically have ~12,000 to 30,000 genes
- About **2-3% of the human genome codes for proteins**

- **>30 million single nucleotide polymorphisms** (SNPs) have been identified (NCBI dbSNP, build 132)
    - A random pair of haploid genomes differs at a rate of **1 base pair every 800** on average in aligned regions
    - Fewer than 1% of SNPs alter protein sequence
- **Segmental duplications** (a duplicated region with one or more similar copies elsewhere in the genome) are common

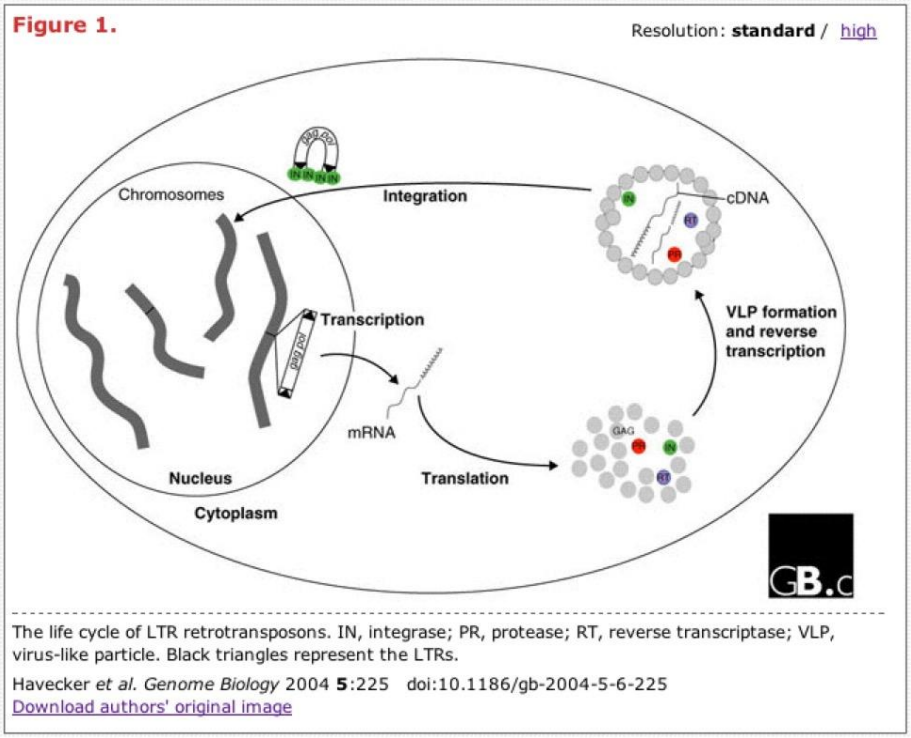
Repetitive DNA

- **interspersed repeats** (transposon-derived repeats)
    - eg. retrotransposed genes that **lack introns**
    - constitute ~45% of the human genome
    - They involve
        - **RNA intermediates** (retroelements) or **DNA intermediates** (DNA transposons, ~3% of human genome).
    - RNA-mediated:
        - **Long interspersed elements** (LINEs, ~21%); encode a **reverse transcriptase**
            - code for polymerase → copy itself on another genomic location
        - **Short interspersed elements** (SINEs, ~13%); include **Alu repeats**
            - no reverse transcriptase, rely on LINEs to copy themselves
            - Alu Repeats
                - There are 300,000 Alu repeats in the human genome
                - Each is **300 base pairs** and contains an ***AluI* restriction enzyme site**.
                - They occupy 3% of the genome.
                - Their distribution is non-random: they are retained in **GC-rich regions** and may confer some benefit

- **Long-terminal repeat transposons** (LTRs, ~8%)

Classes of interspersed repeat in the human genome

			Length	Copy number	Fraction of genome
LINEs	Autonomous	ORF1 ORF2 (pol) AAA	6–8 kb	850,000	21%
SINEs	Non-autonomous	A B AAA	100–300 bp	1,500,000	13%
Retrovirus-like elements	Autonomous	gag pol (env)	6–11 kb	450,000	8%
	Non-autonomous	(gag)	1.5–3 kb		
DNA transposon fossils	Autonomous	transposase	2–3 kb	300,000	3%
	Non-autonomous		80–3,000 bp		



- **Processed pseudogenes**
  - have a **stop codon** or **frameshift mutation**
  - **do not encode a functional protein**
  - They can arise from **retrotransposition**, or following **gene duplication** and subsequent gene loss

- **Simple sequence repeats** (micro-, minisatellites)
    - **Microsatellites**: from one to a dozen base pairs
        - Examples: $(A)_n$, $(CA)_n$, $(CGG)_n$
        - These may be formed by **replication slippage**
    - **Minisatellites**: a dozen to 500 base pairs
        - Simple sequence repeats of a **particular length**
        - composition occur preferentially in different species
    - Micro- and minisatellites comprise **3% of the genome**
    - In humans, an **expansion of triplet repeats** such as CAG is associated with at least 14 disorders, including Huntington's disease
- **Segmental duplications**
    - These are blocks of about **1 kilobase to 300 kb** that are copied intra- or inter-chromosomally.
    - Duplicated regions often share very **high (99%) sequence identity**.
    - Later work has demonstrated **copy number variation** is a major source of variability in the human genome
        - As an example, consider a group of lipocalin genes on human chromosome 9
    - The initial estimates in the 2001 human genome papers were that about 3.6% of the finished human genome sequence may represent segmental duplications, with lengths typically **10-50 kb**
        - now **closer to 5-10%**
- **blocks of tandem repeats** (e.g. at centromere and telomere)
    - These include **telomeric repeats** (e.g. TTAGGG inhumans) and **centromeric repeats** (e.g. a 171 base pair repeat of a satellite DNA in humans).
    - **span millions of base pairs**, and it is often **species-specific**

Genomic Rearrangement

- **Non-allelic homologous recombination (NAHR)** of similar genomic sequences can cause **unequal crossover events**

CpG Islands

- Dinucleotides of CpG are under-represented in genomic DNA, occuring at **one fifth the expected frequency**
- CpG dinucleotides are often **methylated on cytosine** (and subsequently may be deaminated to thymine)
- Methylated CpG residues are often associated with **house-keeping genes in the promoter** and exonic regions
- Methyl-CpG binding proteins recruit **histone deacetylases** and are thus responsible for **transcriptional repression**
- They have roles in **gene silencing**, genomic imprinting, and **X-chromosome inactivation**
- 50,267 CpG islands in human genome
- 28,890 after masking repeats with RepeatMasker
- 5-15 CpG islands per megabase (about <40 genes per megabase)

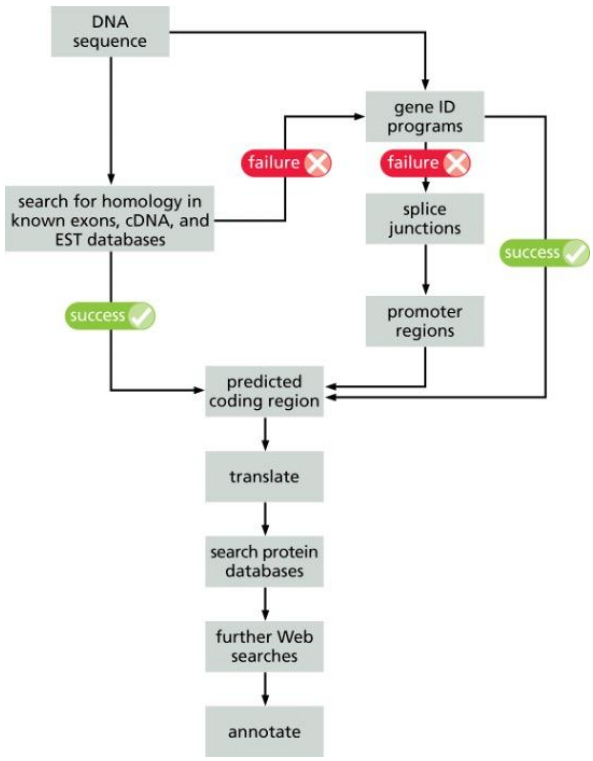
Software to detect repetitive DNA

- It is essential to identify repetitive DNA in eukaryotic genomes.
- **RepBase Update** is a **database** of known repeats and **low-complexity regions**.
- **RepeatMasker** is a **program** that searches DNA queries against **RepBase**. There are many RepeatMasker sites available online

Lecture 8: Gene Finding

Gene Finding Algorithm

- **Homology-based searches** ("extrinsic")
  - Relies on **comparison** with **previously identified genes** and 'external' data → comparative genomics
  - including **RNA-seq** and **ESTs**
- **Algorithm-based searches** ("intrinsic")
  - Investigate nucleotide composition, **open-reading frames,** and other intrinsic ('internal') properties of genomic DNA
  - Compare DNA in exons (unique codon usage) to DNA in introns (unique splice sites) and to noncoding DNA.
- These often be combined



Gene Annotation Program

- HMMGene
- tRNAScan
- ORFFinder
- Grail
- GenScan
- Procrustes
- ORPHEUS
- GLIMMER
- many others…

Initial Genome Examination

- Gene finding programs will use intrinsic and/or extrinsic methods
- Initial examination can identify protein coding regions, **repeat regions** and some **non-coding RNA genes** (tRNA, rRNA)

- Large genomes (i.e. eukaryotic genomes) can be **split** into **overlapping regions** of several megabases or more
    - but watch out for gene split

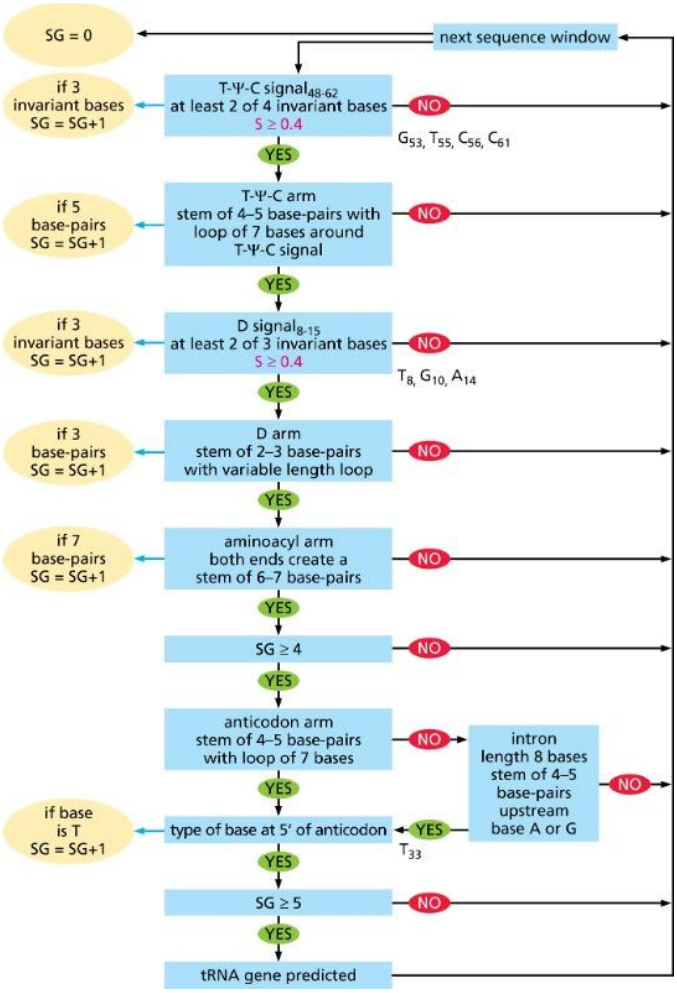
Finding tRNAs

- tRNAs and rRNAs are highly conserved and relatively easy to find.
- The program **tRNAscan** (for example) can find tRNAs with an error rate of <1 per 3 MB, suitable for **bacteria**
- A combination of methods is used in **tRNAscan-SE** for **eukaryotic** genomes
    - tRNAscan
    - RNA-polymerase III related (tRNA transcription)
    - covariance and secondary structure

Identifying Non-coding RNAs (ncRNAs)

- tRNAs (green) may be predicted based on comparisons with **known tRNA sequences** and base pairing patterns **independent of open reading frames**

tRNAscan Method



Bacteria and archaea: finding coding genes

- Four main features of microbial genomic DNA are useful:
  - **Open reading frame** length
  - Consensus for **ribosome binding** (Shine-Dalgarno)
  - Pattern of **codon usage**
  - **Homology** of putative gene to other genes
- **GLIMMER** for gene-finding in bacteria
- gene identification in prokaryotes (**Artemis**)
  - **green** in lower frame shows identified promoter region, -35 from start codon (Met)
  - note overlap of coding region of tolB and promoter

Finding genes in Eukaryotes DNA

- biggest challenges in understanding any eukaryotic genome
    - defining what a gene is
    - identifying genes within genomic DNA
    - annotation of genes and functional regions
- Types of genes include
    - protein-coding genes
    - pseudogenes
    - functional RNA genes
        - tRNA transfer RNA
        - rRNA ribosomal RNA
        - snoRNA small nucleolar RNA
        - snRNA small nuclear RNA
        - miRNA microRNA
- **tRNAscan-SE** identifies 99 to 100% of tRNA molecules, with a rate of **1 false positive per 15 gigabases**.
- Other RNA genes have diverse and important functions.
    - difficult to identify in genomic DNA, because they can be very **small**, and **lack ORF** for protein-coding genes
- In eukaryotes, **protein-coding gene** density is lower, and exons are interrupted by introns.
    - On the other hand, protein-coding genes are relatively easy to find in prokaryotes (high gene density, about one gene per kilobase)
    - There are several kinds of exons → Eukaryotic gene prediction algorithms distinguish those
        - noncoding
        - initial coding exons
        - internal exons
        - terminal exons

        - some single-exon genes are **intronless**

Gene identification in Euk.

- can be complicated
    - intron/exon structure
    - low gene density
    - much larger and more complex genomes
- comparison with other characterized genes (homology)
- iterative refinement of predictions

Features of Genes - ORFs and Start Codon

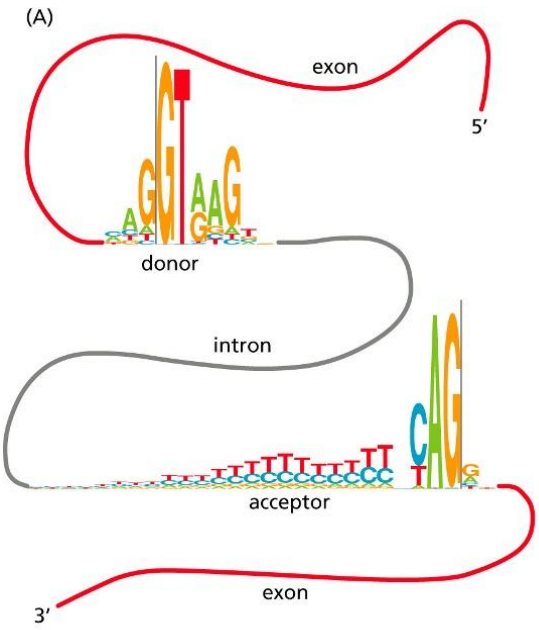
- **Open reading frame**, including start and stop codon
    - long open reading frames are highly likely to be within genes
    - slightly more difficult to separate short genes from non-coding regions
    - Eukaryote start codons are very frequently AUG (Methionine)
        - CUG may be used (rarely); mitochondria in humans use AUA and AUU
- **Stop codons** are TAG, TAA, TGA (DNA)
    - in terms of RNA, stop codons are UAG, UAA, UGA
    - stop codons are at the end of an ORF, by definition
    - mitochondria (except in plants) translate TGA as Tryptophan

Features of Genes - amino acid and codon biases

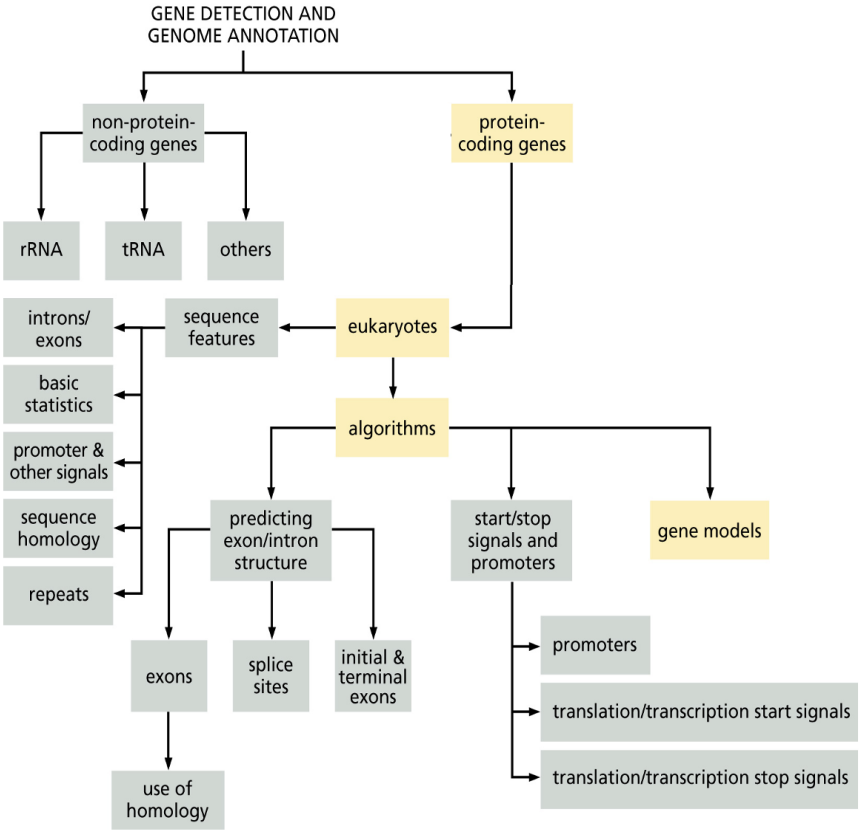
- amino acid codons have **different frequencies** in coding vs. non-coding regions
- codon frequencies themselves will vary, and are correlated to tRNA occurrence and/or level in specific organisms.

Features of Genes - ORF length

- ORF have different length distributions in coding and non-coding regions
- in eukaryotes
    - **exons** are typically **shorter** and are interspersed with introns
    - spice sites are frequently defined by GT - AG



Summary: Identifying Euk. Genes



Finding Genes in Euk. DNA

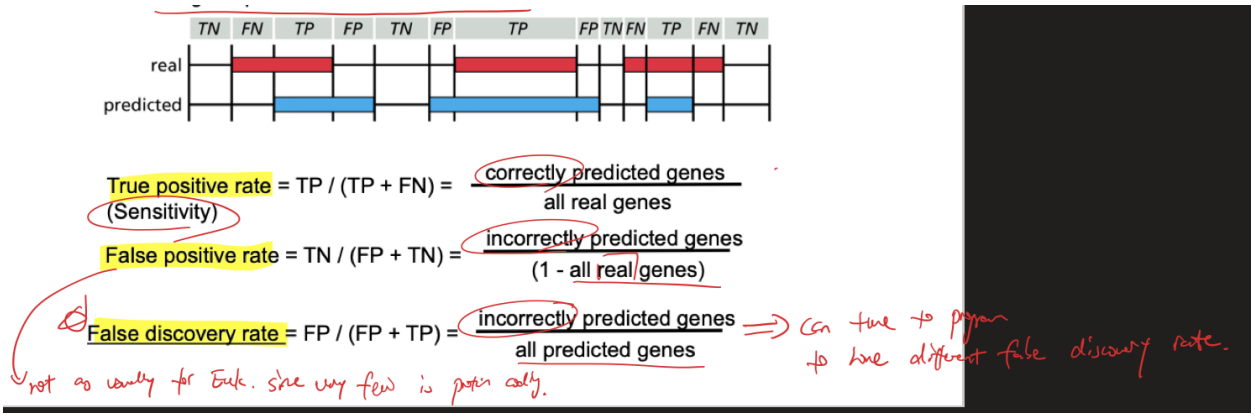
- Both intrinsic and extrinsic algorithms vary in their rates of false-positive and false-negative gene identification.
- Programs such as **GENSCAN** and **Grail** account for features such as the nucleotide composition of coding regions, and the presence of signals such as promoter elements.

Transcription Factor Database

- predict the presence of genomic DNA features: **promoter elements** and **GC content**
- predict **transcription factor binding sites** and related sequences

Estimating Gene Prediction Accuracy

- How can accuracy of gene prediction be assessed?
    - for uncharacterized genomes, not feasible
    - **a well characterized** genome can be used as a **reference** to test gene prediction software
- Commonly used:
    - True Positive Rate (sensitivity)
    - False Positive Rate
        - rarely use for Euk, as it depends on protein coding genes, which is not many in Euk
    - **False Discovery Rate (FDR)**
        - can use to tune the program to have a desired false discovery rate



Lecture 9: Multiple Sequence Alignment (MSA)

Definition

- a collection of **three or more protein or nucleic acid sequences** that are **partially or completely aligned**
- Homologous residues are aligned in columns across the length of the sequences
    - residues are homologous in an **evolutionary sense** or a **structural sense**

Properties

- not necessarily one "correct" alignment of a protein family
- protein sequences evolve…
    - ...the corresponding **three-dimensional structures of proteins** also evolve → homology might not be detected
    - may not be possible to identify amino acid residues that align properly (structurally) throughout a multiple sequence alignment

Features

- some aligned residues, such as **cysteines that form disulfide bridges**, may be **highly conserved**
- there may be **conserved motifs** (short signature regions)
    - e.g. 'kinase ATP binding'
- often contain **conserved secondary structure features**
    - there may be **regions** with consistent patterns of **insertions or deletions (indels)**

Uses

- MSAs are more **sensitive** than pairwise alignment to detect **homologs**
- BLAST output can take the **form of an MSA**, and can reveal **conserved residues or motifs**
- **Population data** can be analyzed in an MSA (PopSet)
- A single query can be searched against a **database of MSAs**
- **Regulatory regions** of genes may have **consensus sequences** identifiable by MSA

Remote Homology

- **MSAs** of more distantly related sequences are typically **better quality** than pairwise alignments

Sequences in MSA

- How many sequences to include?
    - **more is generally better**
    - MSAs of **three sequences** are of limited use
- Removing redundant sequences
    - highly similar sequences do not add much information
    - in some cases, it may be useful to choose **one representative from a set of similar sequences**, e.g. amino acid sequences with >90-95% ID
- Removing questionable sequences
    - sequences need **enough similarity** to produce a **quality alignment** (> 40-50% ID?)
    - depending on the goal, some **low ID sequences** may be included
        - e.g. sequences of **known 3D structure**
- Goal is to include members belonging to the **same 'group' of proteins**
    - Researcher knowledge of the proteins of interest
    - what is known about the **function of each aligned sequence** can help determine if it should be included
- Other considerations
    - are the sequences approximately the **same length**?
    - is there a known **domain structure**?
    - **redundant** sequences initially removed may be **added back to the MSA** (e.g. for a specific region of interest)
    - MSAs may be divided into **sub-groups**

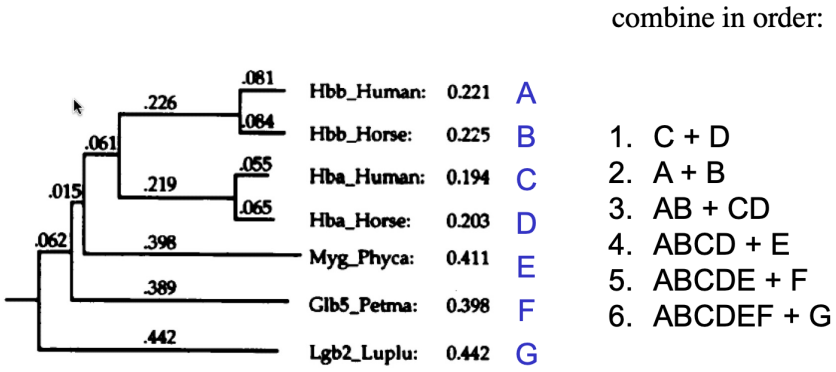
Methods for CreatingMSAs

- Different algorithm might produce slightly different results
- **Progressive alignment** (Feng & Doolittle, ClustalW).
    - sequences are globally aligned in a specified order
    - one of the first MSA methods
- **local/global weighting** (T-coffee)
    - uses information from **other sequences in pairwise alignments**
    - **weights locally similar regions**
- **Iterative re-alignment** (MUSCLE, MAFFT)
    - **sequence subsets** removed and re-aligned
    - uses a **tree-based hierarchy**
- **Context-based alignment** (Decipher)
    - use **secondary structure prediction** as part of alignment scores
    - uses **variable gap penalties** based on local sequence

Feng & Doolittle (Clustal) MSA occurs in 3 stages

- Do a set of **global pairwise alignments** (Needleman and Wunsch) → progressive
    - for N sequences, have **N(N-1)**/2 alignments → quadratic dependence of N
    - Inspection of results
        - check **sequence lengths** - sequences with dissimilar lengths can be edited
        - check for **redundant sequences** - highly similar sequences can be removed
- Create a **guide tree**
    - specifies **order of alignment** → most similar to least similar
    - Convert similarity scores to **rough distance scores**
        - e.g. D ~= 100 - %identity
    - A tree shows the **distance between objects**
    - Current version uses **neighbor joining** to construct the guide tree (**Cladogram** and **phylogram**).
    - **ClustalW** provides a syntax to describe the tree

- **Progressively** align the sequences
    - **dynamic** programming
    - Make a MSA based on the order in the guide tree
    - Start with the two **most closely related sequences**
        - Combine sequences or groups of sequences together based on the order specified in the guide tree
        - Continue until all sequences are added to the MSA
    - Rule: "**once a gap, always a gap**"
        - a **'greedy' algorithm**, no going back to fix things



Guide Trees and Phylogenetic Trees

- ClustalW produces a guide tree in building the MSA → Can this be used as a phylogenetic tree?
    - ideally **no** - it is a tree, but **not an optimized phylogenetic tree**.
    - guide tree is based on **pairwise similarity**, not similarity within the MSA
    - It's **an intermediate step** in analyzing the data for an accurate tree.

Multiple sequence alignment resources

- **Databases** of multiple sequence alignments
    - Text-based searches of **CDD**, **Pfam** (profile HMMs), **PROSITE**
    - Database searches using a **query sequence** with BLAST, CDD, PFAM
    - Other databases:
        - BLOCKS
        - CDD

        - DOMO
        - INTERPRO
        - iProClass
        - MetaFAM
        - PFAM
        - PRINTS
        - PRODOM
        - PROSITE
        - SMART
- Multiple sequence alignment by **manual input**
    - Decipher, CLUSTAL-Omega, MUSCLE, T-coffee, MAFFT

Notes on MSA

- It is possible that a progressive MSA could make an **initial error during pairwise alignment** → The initial error would be **retained** in the MSA
- Two possible solutions (there are others):
    - Iteratively **remove and re-align sequences** within the MSA (e.g. Muscle, MAFFT)
    - **weight alignments** using predicted secondary structure, use variable gap costs to optimize gap placement

Multiple Sequence Comparison using Log-Expectation (MUSCLE) MSA → Iterative re-alignment with sub-trees

- Steps:
    1. estimate a guide tree #1 (their own sequence similarity measure, K-mercounting)
    2. Use a progressive alignment to build MSA #1
    3. computer percent id for sequence pairs, build new **distance matrix**
    4. compute guide tree #2 based on (3.)
    5. build MSA #2
    6. **delete edges** in tree #2
    7. compute profiles for **two sub-trees**
    8. realign profiles from 7 to create new MSA

9. If score improves for MSA in 8, keep new MSA
10. Repeat steps 6 to 9 until convergence or limit reached

- Edges are deleted in order of decreasing distance from root of tree
    - this will effectively **re-align each individual sequence** to the rest of the MSA
- Sub-trees of several sequences are also re-aligned, following order in guide tree
- accuracy is comparable to other heuristic methods & also **very fast** (linear with N)

Lecture 10: Hidden Markov Models (HMMs)

Definition

- HMMs are **probabilistic models** based on **state transitions** in a sequence
- Similar to PSSMs in some ways, but are more versatile and can include gap probabilities
- can be used to:
    - represent a **profile**, similar to a multiple sequence alignment
    - define **changes in states in genes**, such as intron/exon boundaries
    - identify **changes in state in proteins**
        - EX: inside → transmembrane helix →outside for transmembrane proteins

A Simple HMMs Example

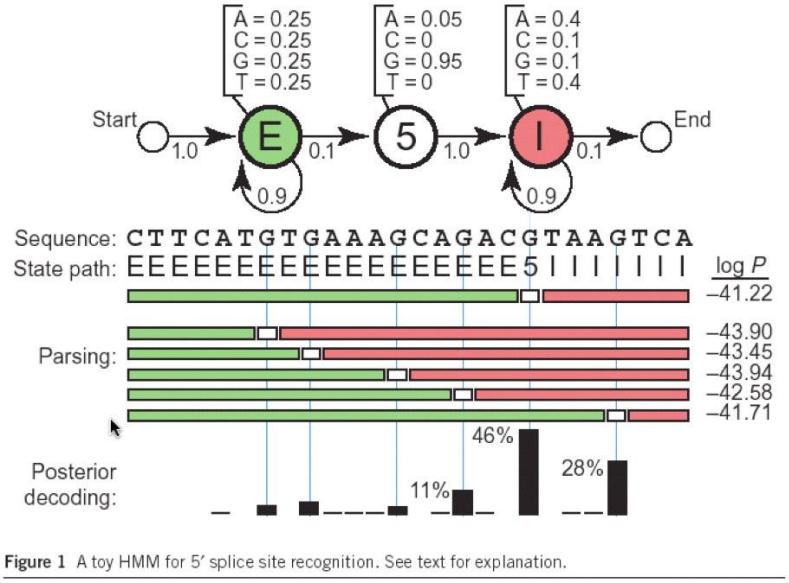


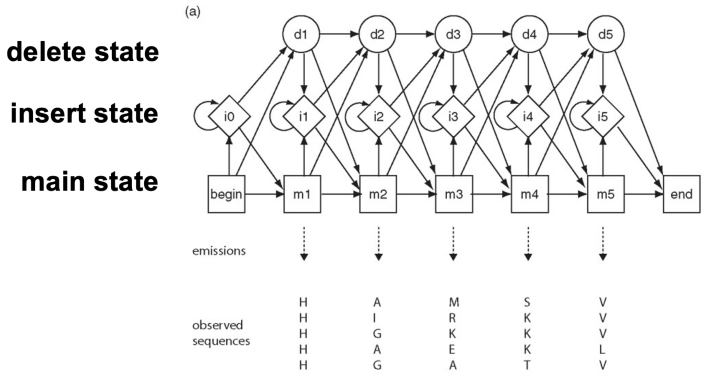
Figure 1 A toy HMM for 5′ splice site recognition. See text for explanation.

- This is a three-state model, with states for exon (E), transition (5), and intron (I).
- In model, bases have an equal probability within exons (A=C=G=T=25%), the transition state is nearly always G(95%) but sometimes A(5%), and the intron state is AT rich (A=T=40%; G=C=10%)
- The most probable state is found by the **Viterbi algorithm** (a dynamic programming algorithm similar to NW and SW).

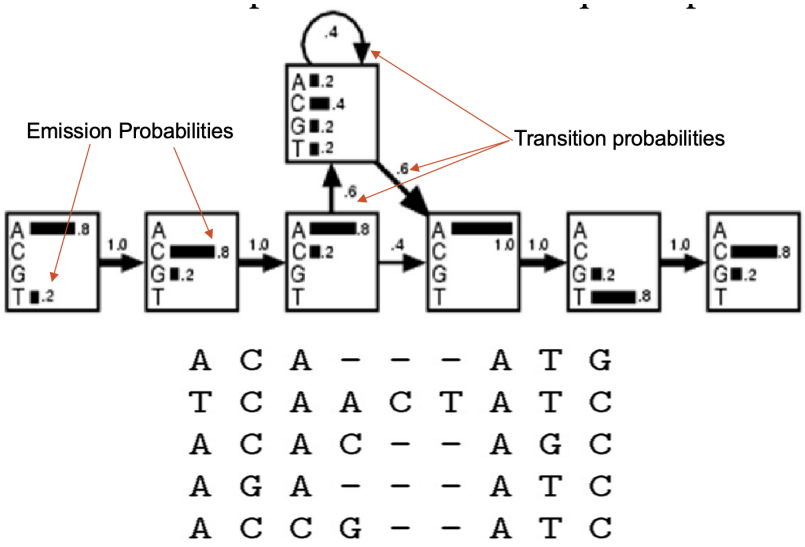
From MSA to HMM

- A profile Hidden Markov model has **"states"** that describe the probability of having a **particular amino acid** residue at arranged in a **column** of a **multiple sequence alignment**
- HMMs are **probabilistic models**, compared to PSSMs which are scoring matrices.
    - The end result can be similar however
- HMM can give **more sensitive alignments** than traditional techniques such as progressive alignments

Structure of Another HMM - a Profile HMM (Similar to an MSA)



- Can be constructed from a MSA & used to represent the MSA profile
    - Translate to **probability** (and log odds ratios) at each MSA site.
    - Model can also include insertion and deletion probabilities
    - Can also have **emission probability** and **transition probabilities**



Representing a MSA profile

- Keep MSA
- Use a **Regular Expression** ([AT][CG][AC][ACTG]*{0-3}A[TG][GC])
- Use a **consensus sequence** (A C A - - - A T C)

HMMs Based Database Search tool

- **HMMER Program** → Build HMM; BLAST HMM against a HMMER database
    - output includes **scores** and **E-values**
- **PFAM**
    - Database of HMM for protein families
    - **HMM logos** graphically depict the likelihood of observed amino acids

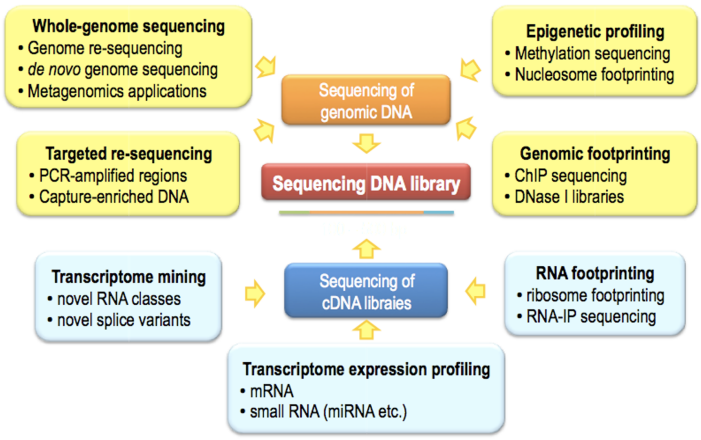
Lecture 11: High-throughput Sequencing

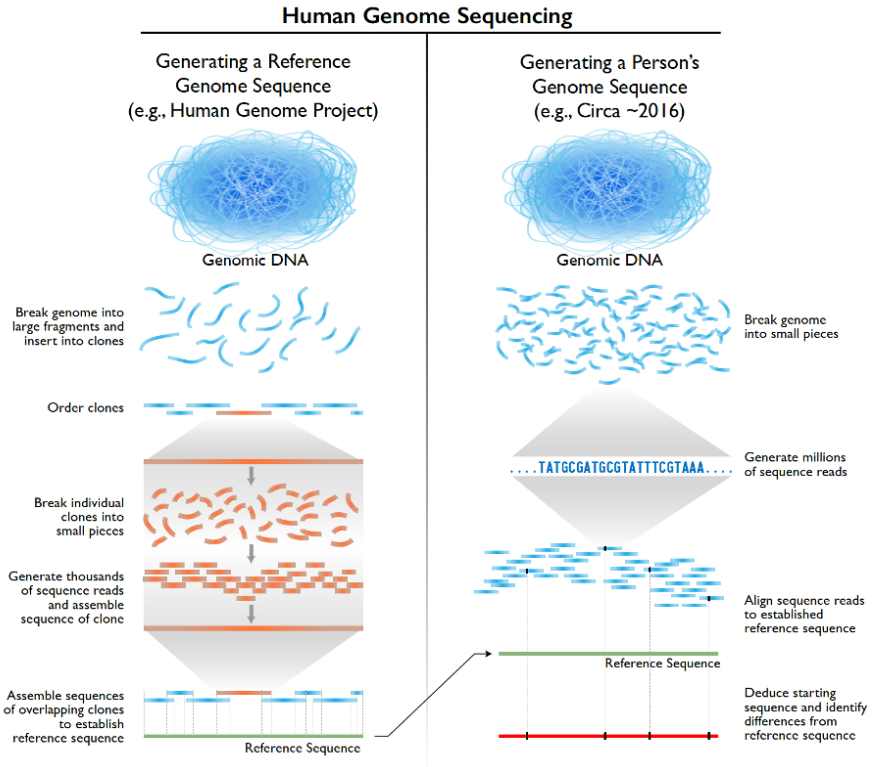
High-throughput Sequencing

- Techniques for **massively parallel sequencing** developed over the last two decades
- The '**next generation**' after Sanger sequencing, which gave us the **first human genome** sequence
- Comprised of several different sequencing techniques
    - Currently the dominant technology is **Illumina sequencing**
        - provides large numbers of sequence reads with relatively low error rates.
    - Other systems include **Pacific Biosciences**, **Oxford Nanopore**, **Ion torrent**, **SOLiD**, and **454 sequencing** (though some are being phased out)

Incentives for New Technology

- utility of **short-read sequencing** greatly increased due to existence of **reference genomes**
- more general technological innovations in microscopy, surface chemistry, polymerase engineering, data storage and analysis.
- sequencing isn't just for genomes…
    - genetic variations
    - RNA expression
    - protein-DNA interactions
    - chromosome conformation
    - epigenetics (DNA methylation etc)





- Need ordered clones then breakdown into smaller pieces for **generating reference genomes**
- You can then just **map newly sequenced genomes** to the reference genome

Implications

- greatly increased speed of sequencing
- greatly reduced cost
- accessible for individual investigators

- acceleration of biological and biomedical research
- analysis of genomes and transcriptomes can become routine
- personalized medicine (individual genome sequencing)
- comparative genomics
- genetic and genomic diversity

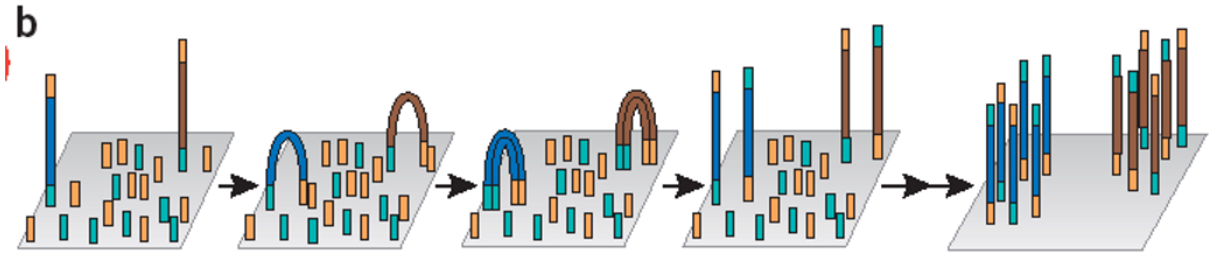
Sequencing Technologies

- **Approaches for DNA amplification**
    - Primer/adaptor sequences are used to bind sequences templates to the surface
    - Amplification can occur on **beads** (Ion torrent, 454, SOLiD) or on a **slide surface** (Illumina)
    - Some systems use **single molecule sequencing** (PacBio, Oxford Nanopore) and don't require amplification
- **Detection systems**
    - **Light based** – a fluorescence signal is detected
        - sanger sequencing; Illumia, PacBio
    - **Semiconductor based** – detects electric signal / ion current, or equivalently a change in pH

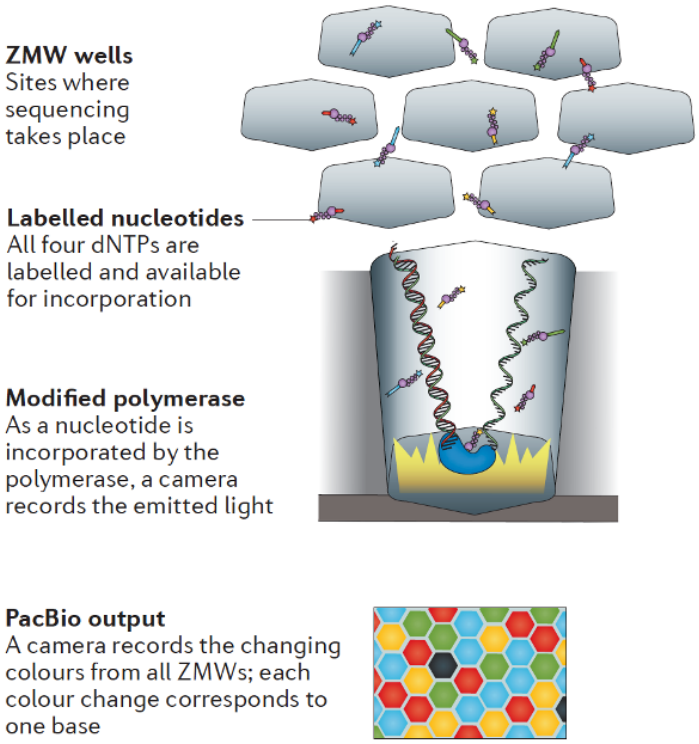
- **Sanger sequencing** – dideoxy chain termination. Uses fluorescently labelled dideoxy (**chain terminating**) nucleotides
    - in shotgun *de novo* sequencing, randomly sheared DNA fragments cloned into E. coli plasmid
    - in targeted resequencing, **PCR amplification** using **primers** flanking target
    - **Chain Termination**
        - Each sample has a **primer**, the four normal deoxy-nucleotides, DNA polymerase, and four **dideoxy-nucleotides** with different fluorescent labels are added in limited quantities (say ~ 1%)
        - chains are terminated with labeled dideoxynucleotides → then run **capillary electrophoresis** to get the sequence

- one Sanger run is typically 96 or 384 capillary separations/electrophoresis
- typically 500-1000 bp per read
- high-throughput cost (genome) ~ $0.50/ kilobase.
    - A good option for **sequencing individual genes**

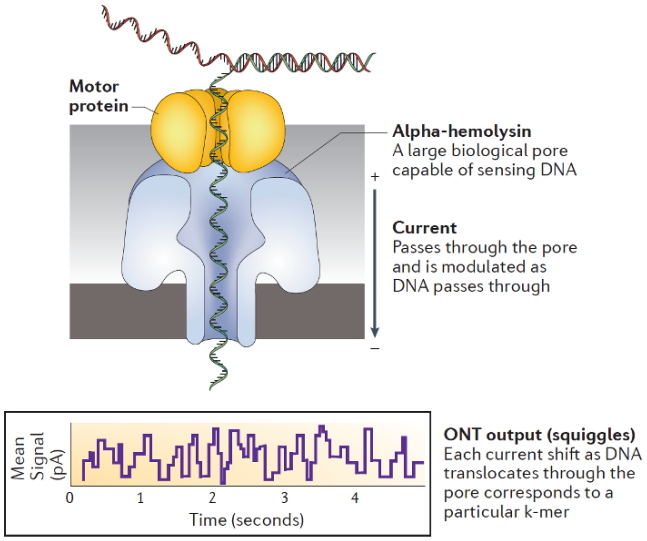
- **Sequencing by synthesis** – nucleotides are detected as they are added to a growing DNA molecule (e.g. Illumina, Pacific Biosciences).
    - Both Illumina and PacBio use cleavable fluorescently labelled nucleotides
    - **Illumina:**
        - DNA fragments ligated to **adaptors**
        - surface of substrate is densely coated with **adaptor primers**
        - initial **low-density binding** to substrate is followed by **bridge PCR** amplification, by adding unlabelled nucleotides and DNA polymerase
        - Strands are **denatured** to get single stranded DNA between amplification rounds
            - high density clusters: ~1000 copies (for single cluster)
            - If density is too high ⟶ you get overlapping clusters, which are not good for your signal
        - After clusters are generated, fluorescently labelled '**reversible terminator**' bases are added to amplified single strands
        - Each base G C A T has a **different fluorescent label**
        - Signal from terminators is recorded when added to the strand
        - **Label is removed** from terminator and cycle repeated
        - Can have <u>substitution error</u>



- **PacBio -** Long read single molecule sequencing
    - PacBio systems use **'SMRT'** - an <u>immobilized</u> **DNA polymerase** and fluorescently tagged nucleotides
    - Nucleotides are detected - added to a growing DNA molecule (Sequencing by synthesis)



- **Nanopore sequencing** – single stranded DNA is passed through a membrane-bound pore and ion current is measured (no fluorescent labels)
    - long read single molecule sequencing
    - Oxford nanopore systems (MioION, ProtION) use an <u>immobilized</u> **protein pore** which passes single stranded DNA.
    - Detection is through changes in **current through the pore**, dependent on the type of nucleotides in the pore
    - Can have **indel errors** if too many repeats → signal is too flat



- **Pyrosequencing (454)** – uses **luciferase** to detect pyrophosphate (PO_3)-O-(PO_3) and produce **light**

Analysis of High-throughput Data Sets

- Assembly of nucleotide sequence (genome or RNA)
- Mapping of gene sequence to reference genomes
- Mapping of peptide sequence to protein databases
- Identification of differentially expressed genes or proteins
- Integration of datasets
- Pathway mapping/enrichment

Summary of sequencing techniques

Summary comparison of systems (~2016 data)

Platform	Read length	GB / run	Run time	Error profile	Cost / GB
Illumina Hiseq 2500 v2	250 x 2	125-150	60 h	0.1%, substitution	$40
Illumina Miseq v3	300 x 2	15	56 h	0.1%, substitution	$100
Ion torrent S5 530 chip	400	8	4 h	1%, indels	$140
PacBio Sequel	~10,000+	3 - 6	< 6 h	~12%, random indels	$200
Oxford Nanopore, Fast mode	~10,000+	44	< 48 h	~12%, indels	$60 ?
SOLiD 5500xl	75	240	10 d	<0.1% AT bias	$70

Lecture 12: Sequence Assembly

Overview

- Whole genome shotgun sequencing uses **reads** sampled from all chromosomes of a given genome
- Reads are typically **short read** (100-250 bp) sequences collected via high-throughput sequencing, but may also include **longer sequences** from Nanopore or PacBio sequencing
- Assembly is possible through **over-sampling**, where reads will overlap with other reads and permit assembly
- *De novo* sequence assembly is the reconstruction of sequence up to chromosome length, without reference to an existing genome or transcriptome sequence assemblies

Issues With Assemblies

- General assumption: similar overlapping reads originate from the same location in the genome
    - This permits joining of reads into **contigs** (contiguous sequences)
- Genome sequences contain **repeat regions** (short repeats to two or more regions of several thousand base pairs of nearly identical sequence)
- **Errors** in sequencing will also occur, and may appear similar to **SNPs** or other genome features
- **Coverage** of the genome will also vary, resulting in some regions with high coverage and others with low or no coverage

Data inputs

- Available technologies have different error structures and read lengths, which will influence assembly methods
- **paired-end reads** (or 'mate-pairs') can help resolve local repeats.
    - Paired end reads are sequenced from either end of a longer sequence, say 100 bp from each end of an 700 bp segment, or 100 bp from each end of a 180 bp segment

- **Short jump reads** are similar to paired-end reads, except they are of greater total length (say ~3500 bp) and are derived from **circularized DNA fragments**

Assembly Methods

- **Greedy**
    - Assembler makes choice with greatest immediate benefit, such as joining best overlapping reads if consistent with existing assembly
    - Emphasizes local assembly
    - Uses **heuristics** to avoid misassembly of repeats
    - Examples: early assemblers (TIGR Assembler, Phrap) and some recent tools (VCAKE)
- **Overlap-layout consensus**
    - Identifies **all pairs of reads** which overlap sufficiently well
    - Creates a **graph** with a node for each read and edges connecting overlapping reads
    - May have high computational overhead for paired overlap calculations.
        - Requires **effective indexing** or other methodology to circumvent this
    - Examples: Celera assembler, SGA
- **De Bruijn graphs**
    - Uses exact substrings of length k
        - We usually want **odd number** of length, as it will not match in the middle
    - Creates a graph of **kmers** that overlap by k-1 letters
    - Typically uses reads to refine graph structure and remove inconsistencies
    - Better results if errors corrected prior to assembly
    - Examples: Velvet, SOAPdenovo, ALLPATHS

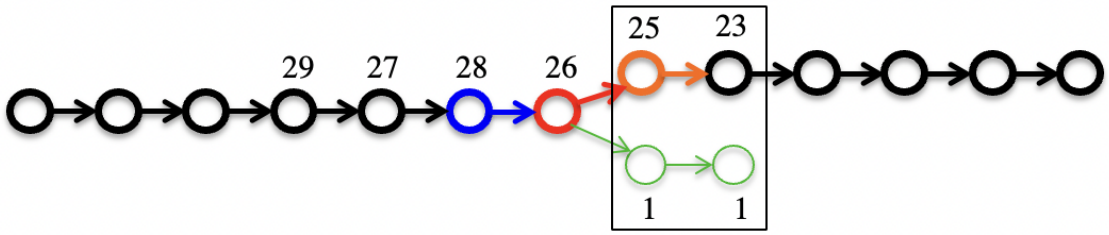
De Bruijn Graphs

- Collect sequence data, say **2 x 200 paired end reads**. Sequencing depth should be sufficient, say **~40x on average.**
- Split all reads into **k-mers**, say length 31. The number of possible kmers k = 4^31, or ~4 x 10^{18} (4 billion billion)

- These kmers are **long enough** that the large majority should be **unique** within the genome.
    - If they are unique, it is possible to put kmers into **long 'chains' or contigs.**

Errors in Genome Assembly

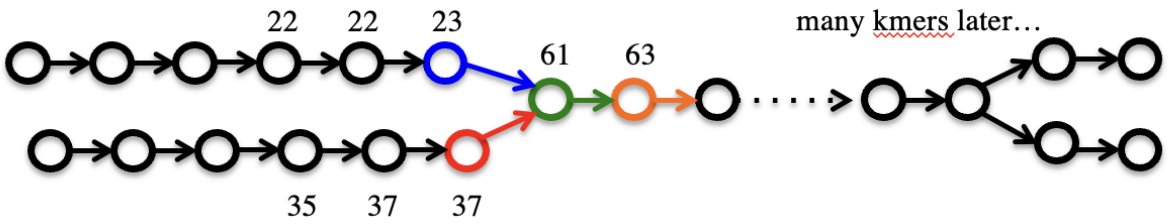
- If the genome was a random sequence with no repeats
    - The De Bruijn graph would be a **single long chain** (assuming all regions of the genome were sequenced, and there were no errors in sequencing)
- But, there are **errors**
    - An error near the **end of a read** could result in the De Bruijn graph looking something like this:



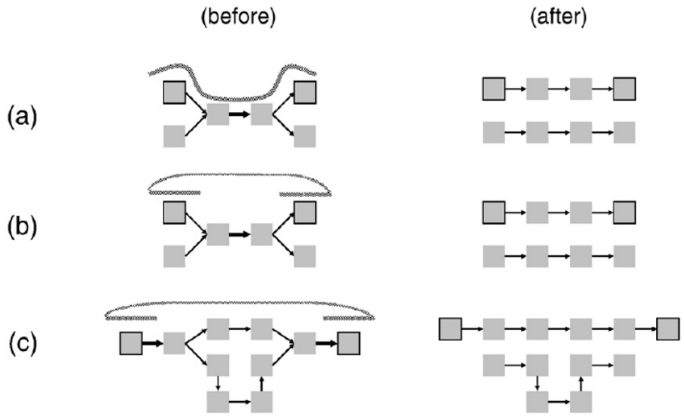
    - Errors usually occur at low frequencies → resolve by selecting the higher frequency one

Repeats in Genome Assembly

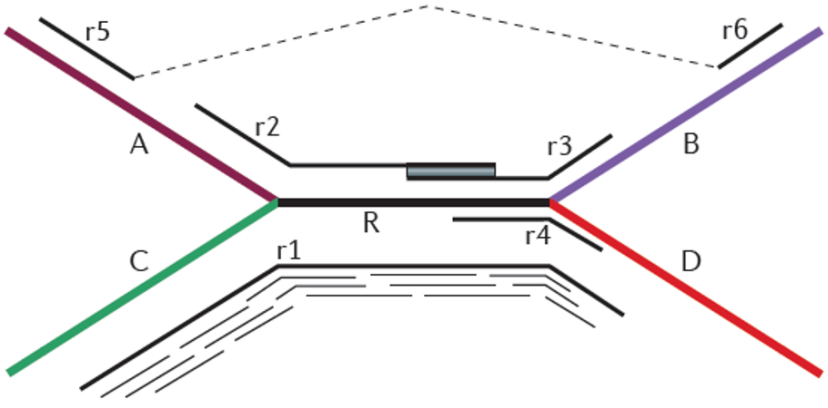
- Repeats cause a different issue.
- The kmers map to **more than one location** in the genome. The De Bruijn graph will contain a **'join'**, where two chains representing different parts of the genome join together:



- we need information from sequences that are longer than the kmer length → **reads** or **paired-end reads**
    - **Read Threading**
        - joins paths across repeats that are shorter than read lengths
    - **Mate threading**
        - joins paths across repeats shorter than paired-end distances
    - **Path following**
        - chooses the path if length fits paired-end constraints.
        - If it is not possible to select the correct path, the graph can be 'cut' into multiple contigs → path not resolved



- EXAMPLE: Split into ARB and CRD



- **Read** data and **paired end data** would be used to resolve this into **a set of linear paths** (ideally one path, though that may not be possible).

Why De Bruijn

- A competing method is **OLC** (**Overlap Layout Consensus**)
    - identify which reads overlap → If there are 100 million reads, this is very **computationally intensive** as there are ~ 10^8 x10^8 pairs.
    - **Heuristic (approximate) methods** are needed to speed up overlap calculations, but can also introduce errors (e.g.incorrectly identify repeat overlaps).
- A **De bruijn graph** removes the need to identify pairwise overlaps, and instead uses kmer counting and ordering.
    - It identifies the '**simple' paths**, and then the more complicated patterns (repeats etc) are resolved using data from complete reads.
- OLC and De Bruijn methods are both in use, and both can produce good assembly results
    - If sequencing techniques introduces high number of errors per read (ie: nanopore, as the reads are longer) → De bruijn fails since most k-mers will contain errors

Genome Assembly Validation

- patterns in the alignment: highlight the potential mis-assemblies
    - misoriented matches → possible misjoin between unrelated regions
    - unusual deep coverage → collapse of repeated region
    - weak join → possible misjoin between unrelated genomic regions

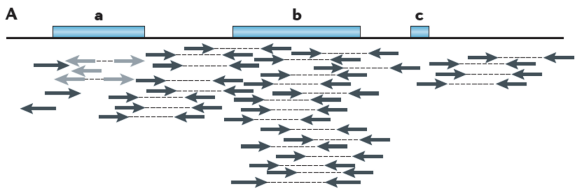


Figure 1 | **Methods for assembly validation.** **A** | Patterns in the alignment of reads along the assembled sequence, highlighting potential misassemblies: misoriented mate pairs, indicating a possible misjoin between unrelated genomic regions (**a**); a region with unusually deep coverage, indicating potential collapsed repeat (**b**); and a weak join, indicating a possible misjoin between unrelated genomic regions (**c**).

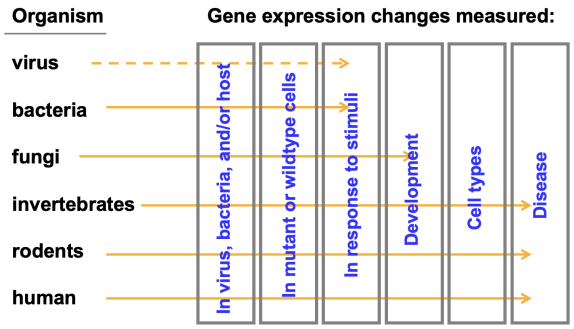
Lecture 13: Gene Expression

Gene expression studies

- The pattern of genes expressed in a cell = characteristic measure of the present state of the cell
- many differences in a cell state are correlated with changes in mRNA levels of genes
- Expression patterns of uncharacterized genes may provide clues to function
- Note however that gene expression does not capture all changes at the protein level
    - Environmental interactions, homeostasis control, etc. all play roles during translation
- Gene expression is **differentially regulated** under several basic conditions
    - by region (e.g. brain versus liver)
    - in development (e.g. fetal versus adult tissue)
    - in dynamic response to environmental signals
        - (e.g. immediate-early response genes)
    - in disease states
    - by gene activity (e.g. wild-type vs. mutant)

Potential impacts

- Identification of complex genetic disease
- Drug discovery and toxicology studies (drugs and gene interaction)
- Mutation/polymorphism detection
- Pathogen analysis (manufacture targets of antivirals)
- **Differential expression of genes over time**, between tissues, and in disease states.



Central Dogma

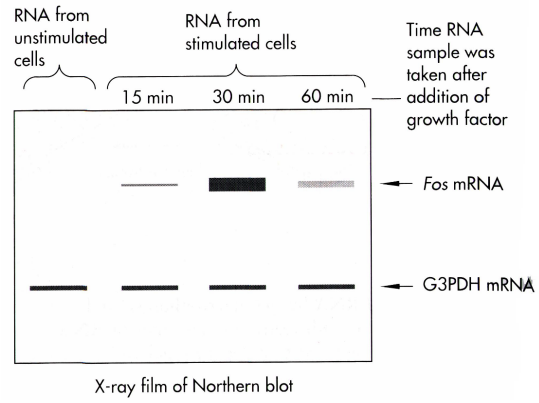
- **Complementary DNA (cDNA)**
    - more stable and more easily manipulated/amplified than RNA.
    - It provides a means to investigate RNA expression patterns within a cell.

mRNA - target of gene expression study

- **rRNA** - ribosomal RNA >70% of cellular RNA
    - But does not tell you too much about function (only about the type of organism)
- **tRNA** - transfer RNA >15% of cellular RNA
- **mRNA** - messenger RNA, 3-4% of cellular RNA
    - This is very few → you have to identify those first
- Gene regulation is mediated partly through RNA, by
    1. transcription
    2. RNA processing (splicing, polyadenylation - poly A tail)
    3. RNA export
    4. RNA surveillance (degradation - rate depends on function, targeting)
    5. siRNA (small interfering RNA)

Traditional gene expression analysis: Northern Blot

- detects specific RNAs
- determine steady-state level of a transcript in a specific RNA mixture
    - RNA is isolated from cells and separated using **electrophoresis**
    - **probed with labeled cDNA** from a specific gene (and a housekeeping gene as a reference)

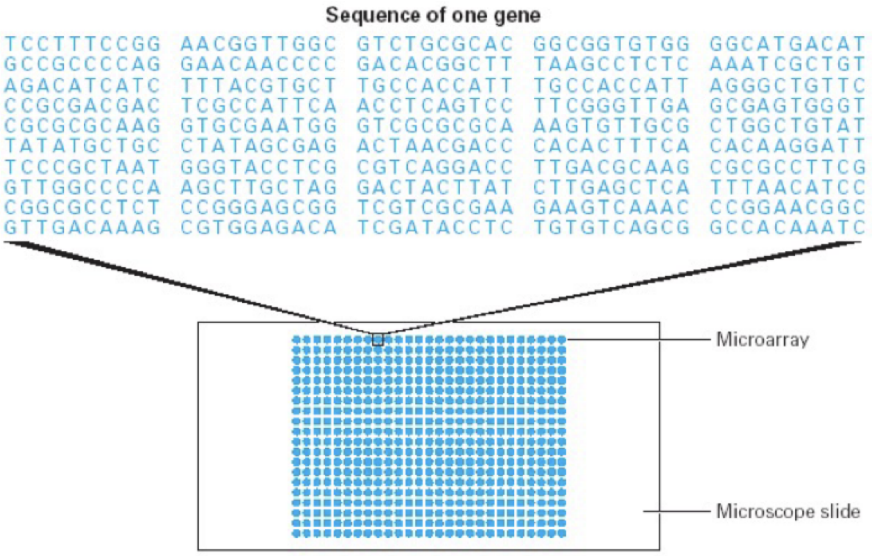


Analysis of Gene Expression in cDNA Library

- An established approach to studying gene expression is through **cDNA libraries**.
    - Isolate RNA (always from a specific organism, region, and time point)
    - Convert RNA to **complementary DNA**
    - Subclone into a vector
    - transform and select for in E. coli
    - Sequence the cDNA inserts.
- A more modern approach: sequence all mRNAs directly using newer methods

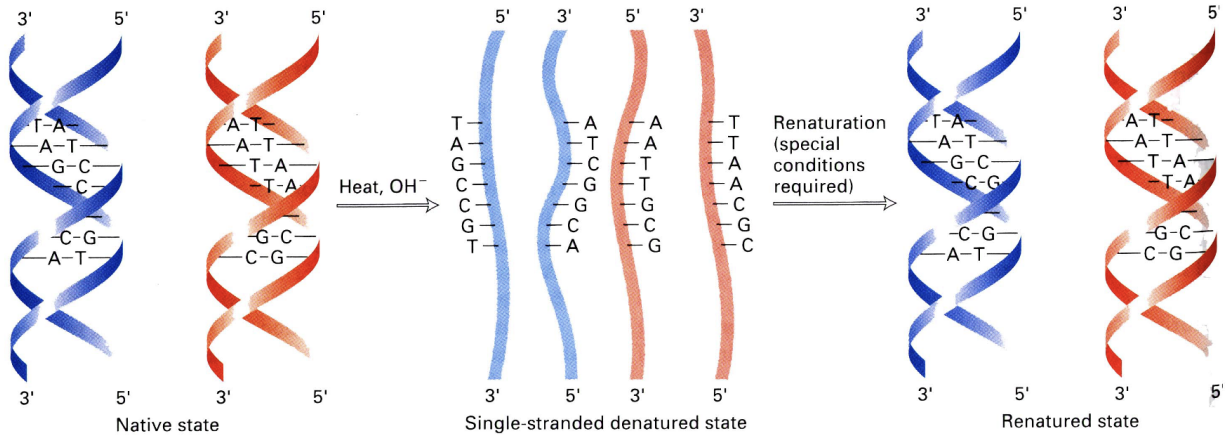
Microarray

- For gene expression study
- A microarray is a **solid support** (such as a membrane or glass microscope slide) on which DNA of known sequence is deposited in a grid-like array.
    - difficult to use microarray if you do not know the sequence of your RNA
- The most common form of microarray is used to measure gene expression.
- RNA is isolated from matched samples of interest. The RNA is typically **converted to cDNA**, labeled with **fluorescence** (or radioactivity), then **hybridized** to microarrays in order to measure the expression levels of thousands of genes.
- **EX: DNA Chip**



Hybridization

- **Base-pairing** and hybridization is underlying principle used by all (nucleotide) microarrays
- Heating for denaturation and then renaturation

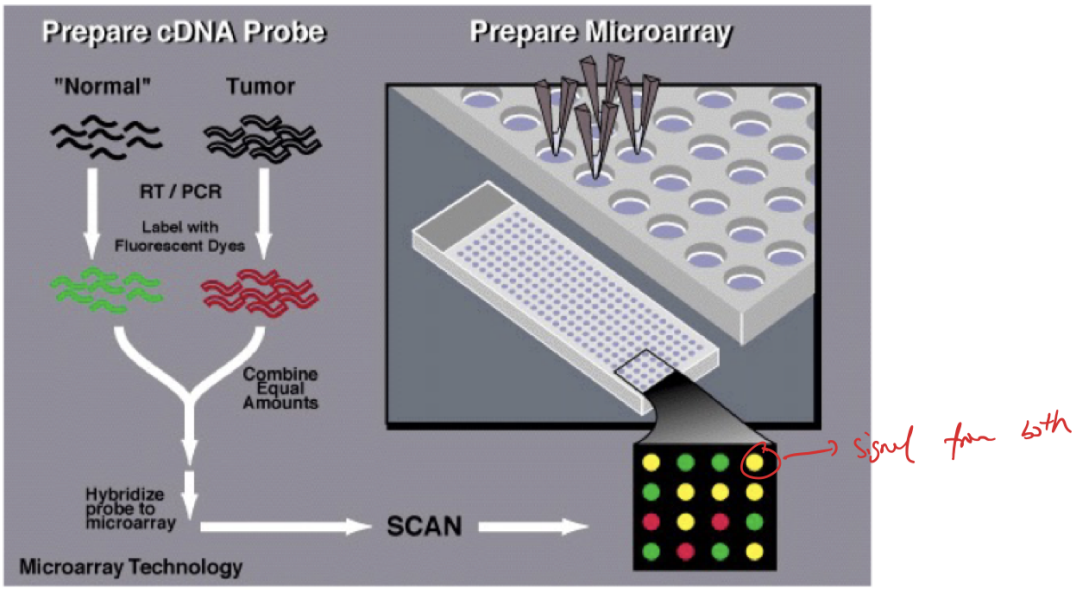


Type of Microarray

- **cDNA arrays**
  - longer segments, ~500 or more nucleotides
  - lower density ~10 to 30 thousand spots/chip → but still enough for all expression of human genes on one chip
- **Oligonuclotide arrays** (e.g. affymetrix)
  - very high density ~400,000 spots/chip
  - short DNA segments, 25-, 50-, 60-mers
  - Can be used to identify **SNPs and variants** and **expression**
- **Genome tiling arrays**
  - Much longer segments, ~100 kb +
  - Useful for identifying genomic location and copy number variations (CNV) (from segmental duplication)

Preparation

- mRNA isolation ⟶ Prepare cDNA ⟶ Fluorescent labeling ⟶ Hybridization



Two colour microarrays

- Two colour arrays are most common
- cDNA from cell state incorporates Cyanine fluorescent dye Cy5 (red) and cDNA from another cell state incorporates Cy3 (green)
- Two samples (one experimental, one control) are mixed and hybridized to an array where the labeled cDNA sequences bind to targets
- Each spot represents expression data for a given gene / sequence
- Total amount of fluorescence from both red and green is measured
- Brightness is **proportional** to amount of cDNA bound to spot on chip
- Colour is due to relative expression levels between control and experimental
    - **Yellow - mix of red and green**

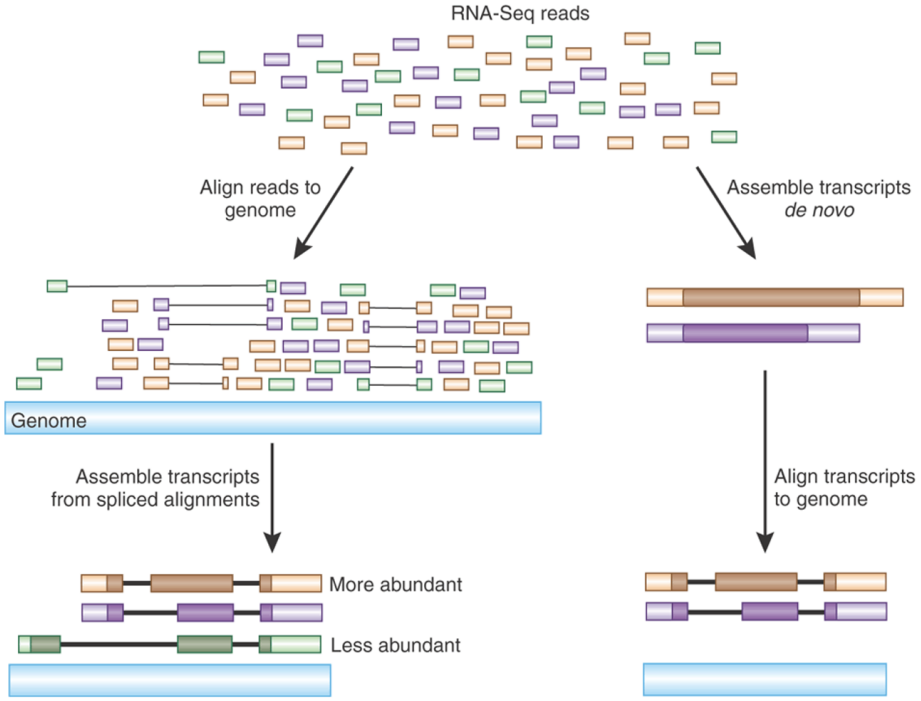
Data quality and Quantification

- Under proper conditions the **intensity of the spot** is proportional to the **expression level of a gene**
- The following should hold:
    - cDNA probe concentration is proportional to the mRNA in the tissue

- cDNA probe amount binding to each spot is proportional
- cDNA target spot on chip is constant
- No contamination
- Signal pixels are correctly identified during image analysis

RNA-Seq

- RNA-Seq is more recently developed high-throughput
- technique for **sequencing RNA** (mRNA or total RNA)
    - RNA is isolated from the target cells/organisms (always from a specific source and time point)
    - Typically RNA is converted into cDNA
    - **High-throughput sequencing** (Illumina, others) used to sequence all reads
    - Reads are mapped to genes/genomes and **reads counted** (more reads = more expression)
    - Can identify novel genes and splicing events
    - Can replace some microarray technologies
    - Data analysis becoming more standardized



Application

- The most common application for RNA-Seq and microarrays is **gene expression analysis** (analysis of the **transcriptome**)
- Other applications include:
    - genotyping and SNP detection
    - DNA mapping
    - CHiP-on-Chip (DNA binding sites of **transcription factors**)
    - comparative genome hybridization
    - splice variant analysis
    - other

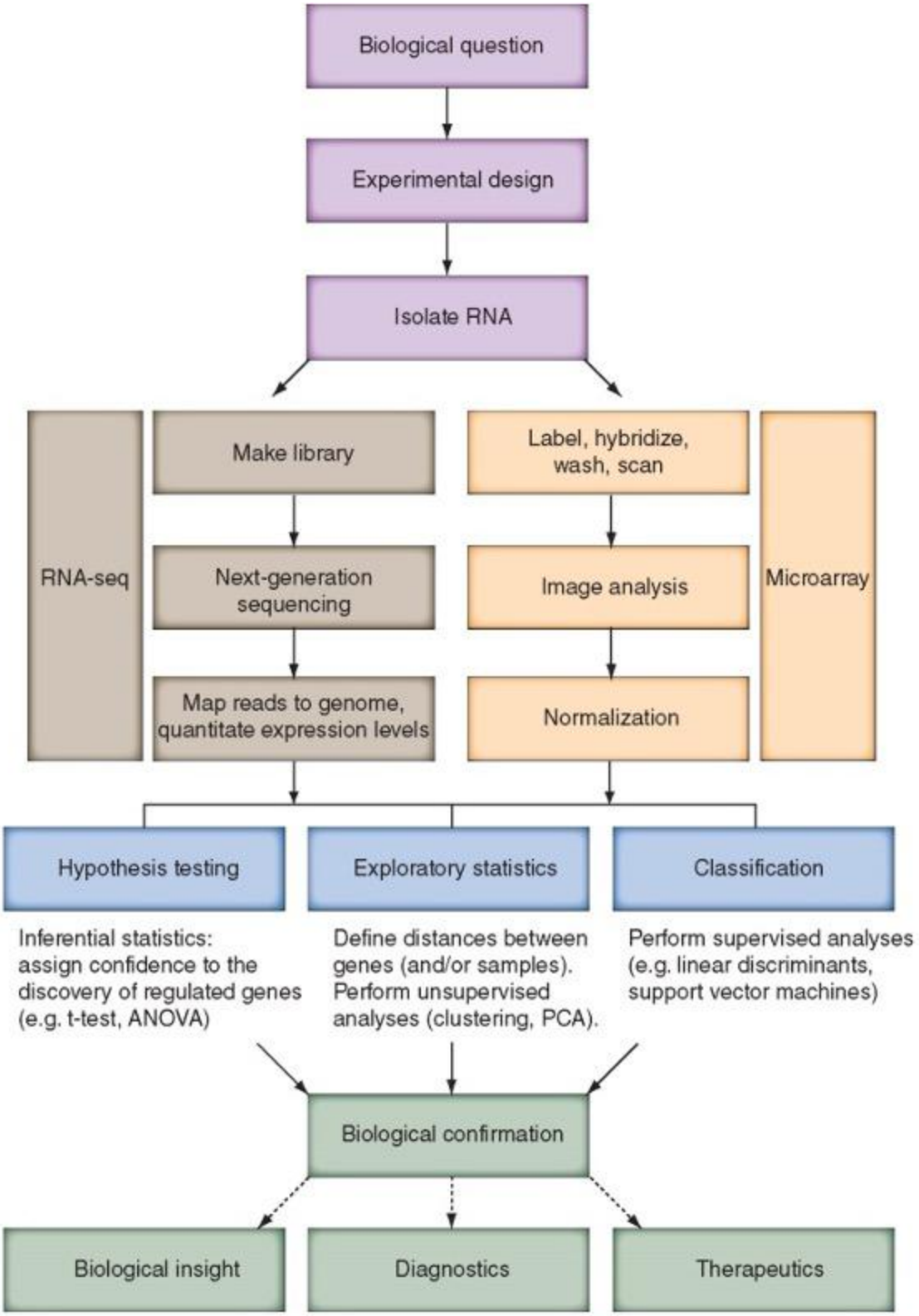
Data Normalization

- Normalization or standardization can separate true variation from variation due to experimental variability
- Common method:
    - E.g. counts (RNAseq) or fluorescence (Microarray) from **healthy tissue** used as a control for expression profile in RNA isolated from a tumor sample
- Expression values for every gene would then be calculated as **a ratio of experimental and control expression**
- Does not eliminate all variation

Absolute vs. Relative Amount

- Usually relative changes are more important than absolute changes
- Can use **log2 ratio** / log ratio to determine reduced expression / increased expression

Experimental design, Data Collection, and Data Analysis



Lecture 14: Expression Stats

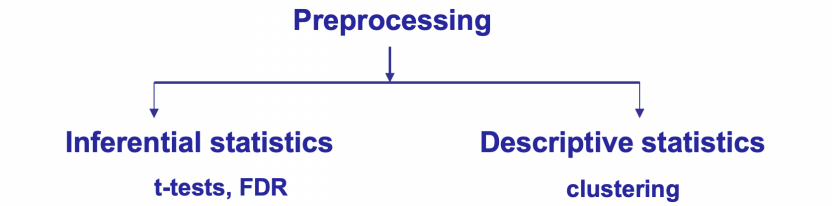
Why do we need expression stats?

- " Which genes were significantly up or down regulated in my experiment?"
    - answered by **inferential statistics**:
        - e.g. statistical testing, p-values, False Discovery Rates, assigning confidence to observed results
            - EX: 'gene X is upregulated 5-fold, p<0.0001.'
- " Can I classify my samples?"
- " What patterns of gene expression may be observed?"
    - answered by **descriptive statistics**:
        - e.g. statistics that provide description of results.
        - **classification** of gene expression patterns
        - **clustering** of samples by similar gene expression
        - **clustering** of genes by similar expression profile

Data Analysis - Overview

- begin with a **data matrix** (gene expression values versus samples)

	Control	Treated
gene AZ1	101	178
gene BA1	5032	4830
gene BC2	38	50



Data Preprocessing

- Observed differences in gene expression could be due to transcriptional changes, or they could be caused by **artifacts** such as:
    - variations in RNA purity or quantity
    - variations due to sample pretreatment, e.g rRNA removal

- different labeling efficiencies of Cy3, Cy5 (Microarrays)
- variations in detection efficiency
- biological variation (that naturally exists between samples)?

- The main goal of data preprocessing is to **remove the systematic bias** in the data as completely as possible, while preserving the variation in gene expression that occurs because of biologically relevant changes in transcription
- A basic assumption of most normalization procedures is that the <u>average gene expression level does not change in an experiment</u>
  - The total amount of mRNA should not change during an experiment → some up some down

Accuracy vs. Precision:

- We want both!

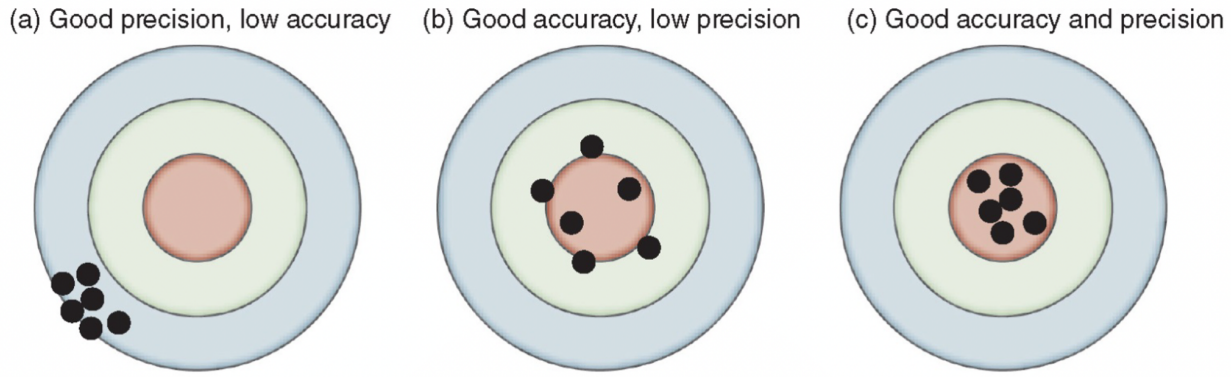


FIGURE 11.5 Accuracy and precision. (a) Good precision is characterized by reproducible results. It is assessed by repeated measurements of samples (technical replicates). (b) Good accuracy is characterized by measurements that correspond to an independently known result. It can be assessed by measurement of known ("spiked in") concentrations of RNA to an experiment, or by measuring dilutions of known concentrations of RNA. (c) A goal of preprocessing algorithms is to achieve both accuracy and precision.

Global Normalization

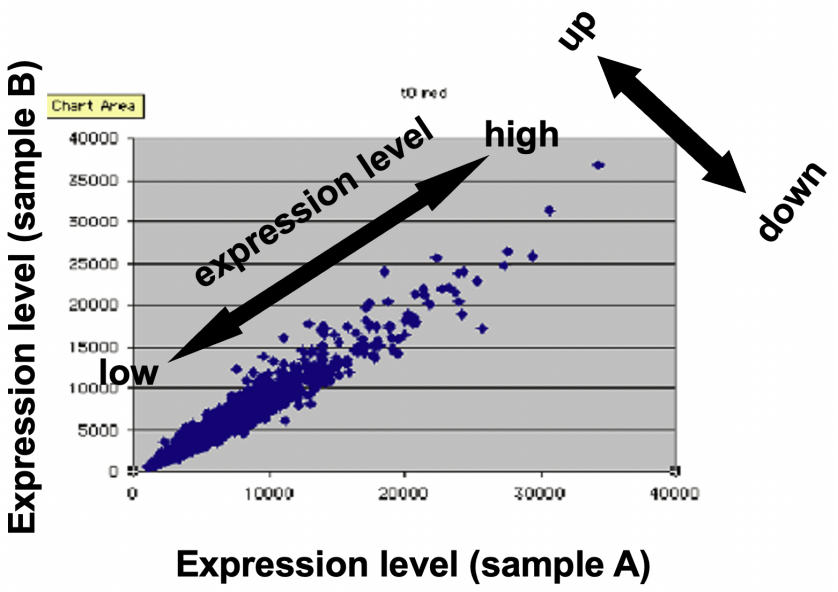
- used to correct two or more data sets
- Steps:
  - Step 1: subtract **background intensity values**, if present (e.g. for microarrays, use a blank region of the array)

- Step 2: globally normalized so that the **average ratio = 1** (apply this to 1-channel or 2-channel data sets) ⟶ remove any biases for low expression

- once globally normalized, data are usually **log transformed** ⟶ remove any noise for high expression
- **Proportional changes** are generally more important than **absolute changes** in biological data (e.g. 100-->200 vs. 1100-->1200)
- Another approach
  - Some researchers use **housekeeping genes** for global normalization
    - housekeeping genes are typically invariant
  - One highly cited source: "Human housekeeping genes revisited". E. Eisenberg and E.Y. Levanon, Trends in Genetics, 29 (2013)
    - Identified a set of 3804 human genes that are expressed uniformly across a panel of tissues.

Scatter Plots

- Useful to represent gene expression values from two samples (e.g. control, experimental)
- Each dot corresponds to a gene expression value
- Most dots fall along a **line**, ratio 1:1 after normalization
- **Outliers** represent up-regulated or down-regulated genes



Log-log transformation

- Log transformation also results in a more even distribution of data points, and is more suitable for statistical analysis (<u>variance is more uniform</u> over data range)

Ratio vs Mean Intensity

- To highlight differential expression, data may be plotted as **expression ratio vs. geometric mean intensity**.
- For two samples Con(control) and Exp(Experimental):
  - **X axis**: geometric mean intensity

$$X = \log_{10}[(\textbf{Con intensity x Exp intensity})^{1/2}]$$

$$= 0.5\ [\ \log_{10}(\text{Con intensity}) + \log_{10}(\text{Exp intensity})]$$

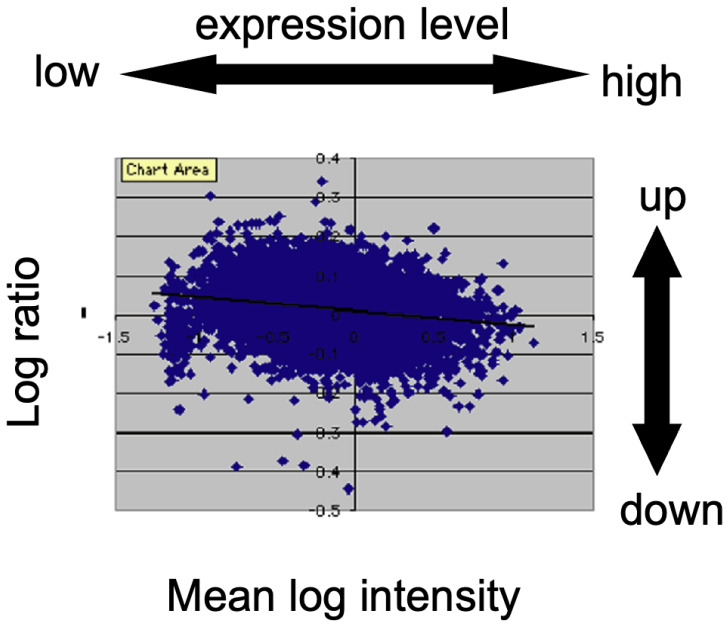
$$= \textbf{mean } \log_{10} \textbf{ intensity}$$

  - **Y axis**: log ratio of intensity

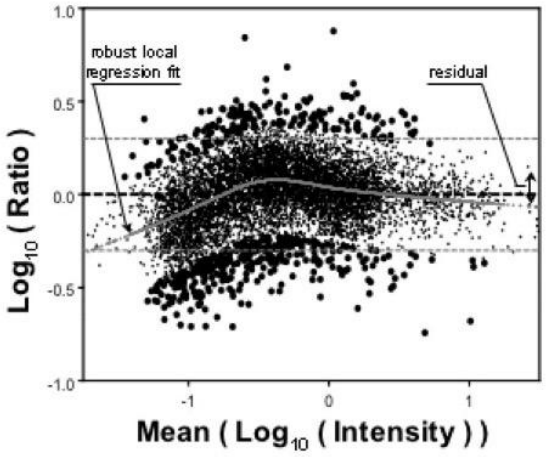
$$Y = \log_{10}(\textbf{Exp intensity / Con intensity})$$

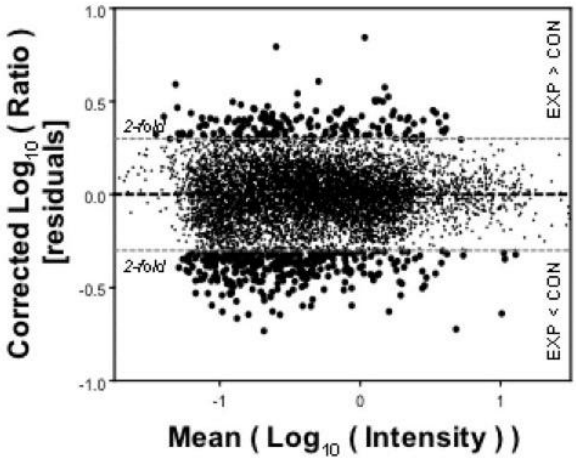
$$= \log_{10}(\text{Exp intensity}) - \log_{10}(\text{Con intensity})$$

- **Results**:
  - a regression line through the data set may be used to correct for intensity dependent variation
  - Ideally, <u>non-changing is on the line of y=0</u>



- **Software to correct local variance artifacts**
    - One method is **lowess normalization** (LOcally WEighted Scatterplot Smoothing)
        - e.g. Microarray analysis software - SNOMAD (http://snomad.org)

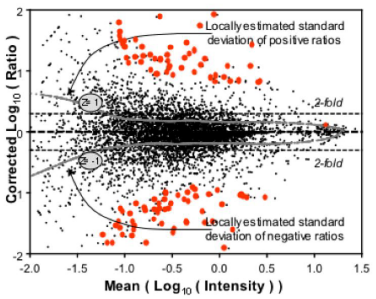


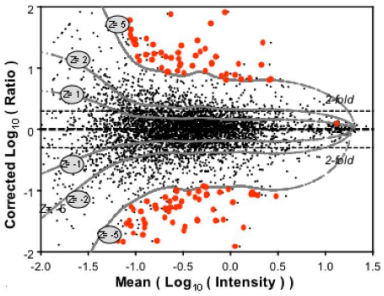


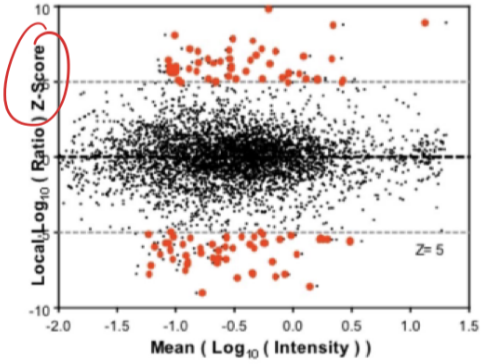
From Local Variance to Local Z-Score

- A Z-score is count or signal data expressed in terms of standard deviations (s) from the mean (m):

$$\text{Z-score}(x_1) = (x_1 - m)/s$$







Inferential Statistics

- used to make **inferences** about a population from a sample.
- **Hypothesis testing** is a common form of inferential statistics.
    - **Null hypothesis** "There is no difference in signal intensity for the gene expression measurements in normal and diseased samples."
    - The **alternative hypothesis** is that there is a difference.
- We use a **test statistic** to decide whether to accept or reject the null hypothesis. Using traditional statistics, a significance level, a, is often set to **$p < a = 0.05$**.

Hypothesis Testing

- Statistical **inference** (i.e. detecting differences) often uses a **null hypothesis**
    - i.e. a default hypothesis that can be proved wrong.
    - Say you take a measure the expression of a single gene in two populations. The **null hypothesis is that the expression is the same**:
        - H_0: $\mu_1 = \mu_2$, where:
        - H_0 = the null hypothesis
        - μ_1 = the mean of sample type 1, and
        - μ_2 = the mean of sample type 2.
- The alternate hypothesis is H_A: $\mu_1 \neq \mu$.
    - If you can show that the **null hypothesis is unlikely**, then the alternate hypothesis is likely to be true.
- Possible outcomes of hypothesis testing

	Null hypothesis (H_0) is true	**Null hypothesis (H_0) is false**
Reject null hypothesis	Type I error False positive	Correct outcome True positive
Fail to reject null hypothesis	Correct outcome True negative	Type II error False negative

- **p-values** can be used to estimate type I error, the chance of a false positive. So, if a p-value is reported as less than a threshold α (say $p < \alpha = 0.05$), it means the **chance of a**

false positive (incorrectly rejecting the null hypothesis) is less than 5%. Therefore, there is 5% chance of 'crying wolf'.

T-Testing

- A very common form of hypothesis testing is the t-test. It can be used to test if two means are different, or wherever the test statistic follows a **t-distribution**.
- For testing differences between means X_1 and X_2, assuming equal variance between samples:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1 X_2} \cdot \sqrt{\frac{2}{n}}}$$

- Where $n = n_1 = n_2$ is the number of samples, and S_{X1X2} is the pooled **variance** estimate.
- **IF $X_1 = X_2$, then t follows the t-distribution.**
- The t-distribution can be used to test the null hypothesis.
- If a single hypothesis is tested at $\alpha = 0.05$, then that hypothesis has a **5% chance of being a false positive**.
- If multiple hypotheses are tested at $\alpha = 0.05$, then each hypothesis has up to a 5% chance of being a false positive. Therefore, for **1000 hypothesis tests**, the expectation is that approximately **50 tests will be false positive**s if all are tested at $p < \alpha = 0.05$.
- Is it appropriate to set the significance level to $p < 0.05$ for high-throughput data?
    - If you hypothesize that a specific gene is up-regulated, you can set the probability value to 0.05.
    - Within a transcriptomics experiment, you might measure the expression of **10,000 genes** and hope that any of them are up- or down-regulated. But you would expect to see **5% (500 genes)** identified at the $p < 0.05$ level **by chance alone**.
    - A very strict option is to apply a **Bonferroni correction**. The level for statistical significance is **divided by the number of measurements**, and the criterion becomes:

$$p < (0.05)/10{,}000 \quad \text{or} \quad p < 5 \times 10^{-6}$$

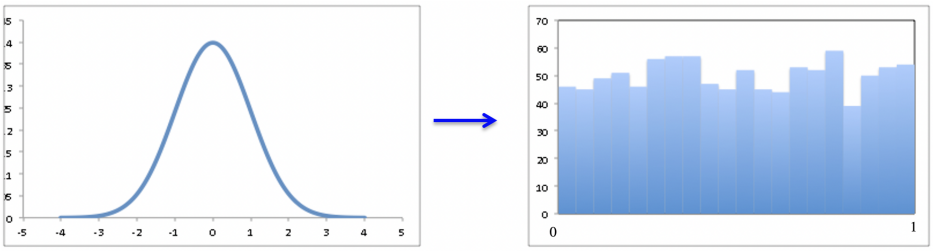
- The **Bonferroni correction** means that the chance of **any** false positive being present is < 0.05
- However, using a Bonferroni correction for multiple hypotheses will reject many true positive cases
    - led to widespread use of another measure, the **False Discovery Rate, or FDR**. Instead of setting a family-wise error rate, the rate of false discoveries is set to a given value

        FDR = Expected (# false positives / # total positives)

→ you can set how many false positives you want

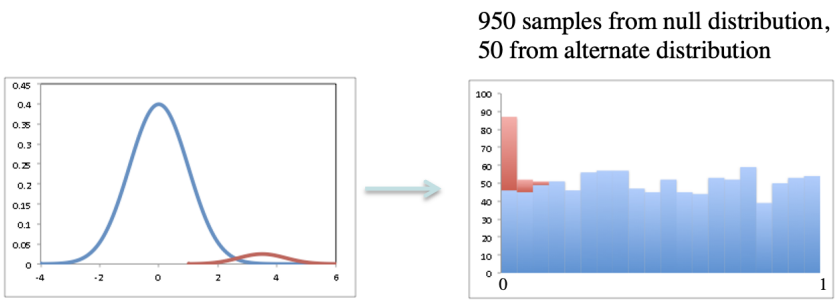
Null p-value distribution

- A p-value is the probability of seeing the observed effect under the null hypothesis.
    - If two groups of samples were randomly selected from a null distribution and tested, and the test was repeated many times we would expect an even distribution of p-values.
    - i.e. if we sampled two groups from a null distribution 1000 times and calculated p-values, approximately 50 p-values would be between 0 and 0.05.



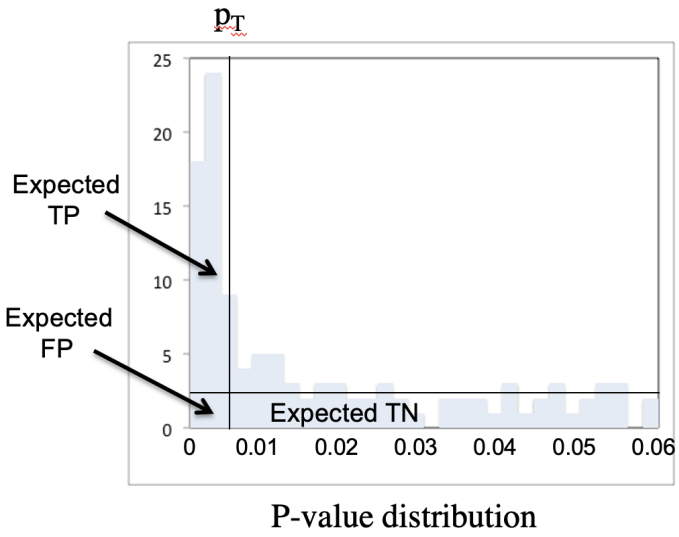
Background distribution P-value distribution

- If real effects are present, the **distribution of real effects should be different from the null distribution**.
- If the effects are **large** enough, **corresponding p-values will be small**.
- The resulting distribution of p-values will include both **null and alternate distributions**. Say in our experiment we test 1000 genes, and 50 of these are differentially expressed (from a different distribution)
- Using a p-value cutoff of 0.05 we would still include many false positives



FDR corrections

- Set the number of expected false positives to **some fraction of all detected positives**
  - e.g. q = 0.10.
    - Here, approximately **10% of all 'calls' (TP+FP) will be false positives**.
    - A threshold $\mathbf{p_T}$ can be set based on the observed p-values
    - The value of p_T will typically be << 0.05



Estimating false positive

- If our background distribution function is a good approximation, we can estimate the number of expected false positives for any p-value threshold, p_T.
- For example, for **m = 1000 hypotheses** and a **p-value threshold of p_T = 0.01**, the expected number of false positives is

$$E(N_{FP}) = m*p_T = 1000 * 0.01 = 10$$

Observed Positives (True Positives + False Positive)

- Assuming there are some cases where the **null hypothesis is not true** (e.g. some genes are differentially expressed), we want to identify these as different from the background distribution.
- The **greater the difference** from background, the **smaller the p-values**.
- If p-values are put in **rank order** from smallest to largest,

$$p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}$$

- then we can use a **p-value cutoff p_T** so that some observed number of positive calls i = N_{obs} are made.
    - e.g. say for $p < 0.01$ we see $i = N_{obs} = 30$ cases. This can include both true positives and false positives.

False Positive Rates

- We want to make a number of calls so that the **expected rate of false positives q is less than a specified amount**, say q = 0.10. (i.e. less than 10% false positives). Equivalently,

$$\mathbf{q} \geq \frac{\text{expected number of } \mathbf{false} \text{ positives}}{\text{total number of positives}}$$

- Our ordered list is a set of **observed p-values $p_{(i)}$** for each rank i =1 to m (from smallest to largest), where **m is the total number of tests** (genes).

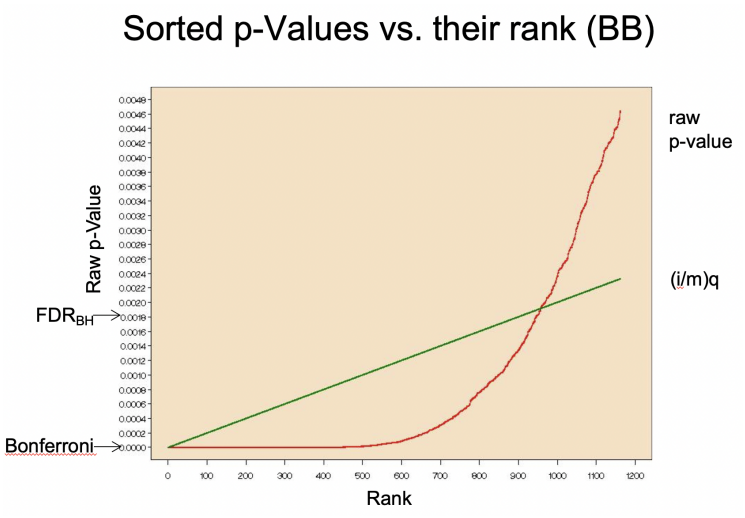
$$q \geq m * p_{(i)} / i$$

- Find the largest i such that:

$$p_{(i)} \leq (i / m) * q$$

FDR Controlling Procedures

- This gives us the Linear step up procedure of Benjamini and Hochberg (BH procedure)
    - **Order the p-values** $p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(m)}$, where m is the number of comparisons (i.e. genes)
    - Let $k = \max \{ i, \text{such that } p_{(i)} \leq (i/m)q \}$ where **q is the target FDR** (say 0.05)
    - Reject null hypotheses $H_{(0)}$, $H_{(1)}$, ... $H_{(k)}$ (if no such k exists don't reject any)



The intersection is the **FDR-BH Threshold**

Benefits of FDR

- We get more samples passing the significance threshold than Bonferroni
- We get fewer false positive than a single p-value of 0.05, and we know exactly how many percent (equals to q)

Lecture 15: Expression Clustering

Descriptive Statistics

- Gene expression data are highly dimensional: there are many thousands of measurements made from a small number of samples.
- **Descriptive (exploratory) statistics** help you to **find meaningful patterns i**n the data.
- The first step is to arrange the data in a **matrix**. Next, use a **distance metric** to define the **relatedness** of the different data points. Two commonly used distance metrics are:
    - **Euclidean distance**
    - **Pearson coefficient of correlation**

Euclidian distance

- In 3D space, the distance between two points (x_1, x_2, x_3) and (y_1, y_2, y_3) is

$$d_{xy} = [\ (x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2\ ]^{1/2}$$

- In n-dimensional space, the distance is

$$d_{xy} = [\ S_{i=1\ to\ n}\ (x_i-y_i)^2\ ]^{1/2}$$

- Numerical values are often **normalized** and variance corrected prior to calculating distances
    - e.g. transformed variable x' has **mean=0, st.dev.=1**.

Pearson Correlation Coefficient r

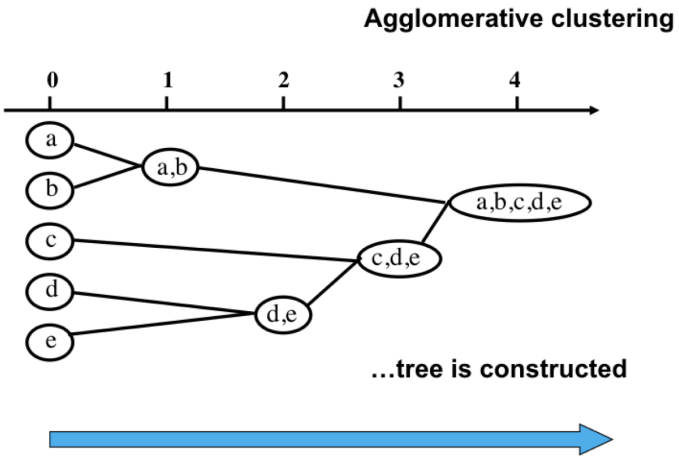
- For any two series of numbers $\{X_1, X_2, \ldots, X_N\}$ and $\{Y_1, Y_2, \ldots, Y_N\}$,

$$r = \left[\sum_{i=1\ to\ N} \frac{(X_i - X_{avg})}{\sigma_X} \frac{(Y_i - Y_{avg})}{\sigma_Y} \right] / (N-1)$$

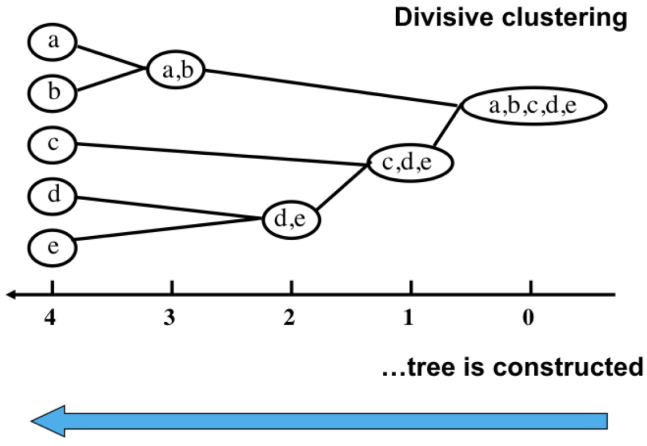
- where X_{avg} is the mean of all X values and s_X is the standard deviation of all X values.
- r varies from **r = 1** (two series **correlate exactly**) to **r = -1 (inverse correlation)**.
    - **r = 0** indicates X and Y are **uncorrelated**.
- This can be translated to a <u>rough</u> distance measure using **D = 1 - r** $\rightarrow$ **0 <= D <= 2**
    - Rough because distance for uncorrelation in phylogenetic is not necessarily 0

Clustering

- Clustering algorithms offer useful **visual descriptions** of gene expression data.
- Genes or samples, or both, may be clustered,
- **Hierarchical clustering** is often used
    - This may be **agglomerative**
        - building up the branches of a tree, beginning with the two most closely related objects → pairwise combination



    - or **divisive**
        - building the tree by finding the most dissimilar groups first → keep dividing the tree in half



- In each case, we end up with a tree having **branches** and **nodes**.

- Agglomerative and divisive clustering can produce somewhat different results, as shown here
    - but we do not know which is right ⟶ in both cases, we just get some description about the statistics

Unweighted Pair Group Method with Arithmetic Mean (UPGMA)

http://genomes.urv.cat/UPGMA/index.php?entrada=Example6

- At each iteration, the closest two groups are combined to form one new group.
- This process is continued until there is only one group, producing a **rooted tree**.
- UPGMA may be a suitable method for expression data, as expression data isn't typically additive and UPGMA doesn't assume additivity
    - change of gene expression does not always accumulate in the same direction
- Steps:
    - first calculate **pairwise distances** between taxa
        - can use the fraction of mismatches have been used
    - then combine closest pair of taxa
    - recalculate distance matrix with new group (take the average)
    - combine closest pair again
    - recalculate the matrix again to get distance
    - finish tree ⟶ the distance is divided in half for each branch

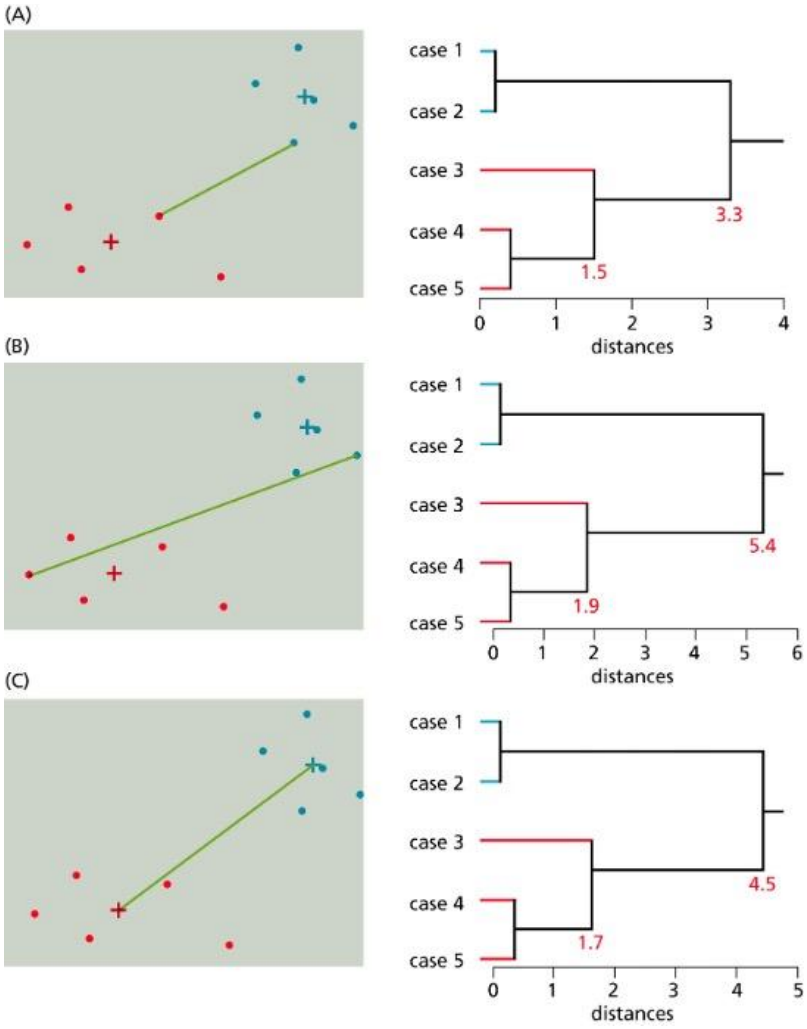
Distance Between Clusters

- Distances between clusters may be defined in several different ways.
- With UPGMA, the distance between clusters is typically the **average of all distances between cluster elements** ⟶ average linkage clustering

$$d_{XY} = \frac{1}{N_X N_Y} \sum_{i \in X,\, j \in Y} d_{ij}$$

- 3 ways to compute linkage distances:
    - **Single Linkage** (shortest distances)

$d_{XY} = \min \{ d_{ij} \}_{i \in X, j \in Y}$



    - **Complete Linkage** (largest distances)

$d_{XY} = \max \{ d_{ij} \}_{i \in X, j \in Y}$

    - **Centroid Clustering** (average distances)

$d_{XY} = d_{X\,Y}$, or the distance between the centroids of each cluster

Visual Representation of Expression Ratio

- changes in expression ratios can be represented on **a colour scale**, to enable visualization of large data sets
- red and green are often used for this as well
    - a red spot on microarray is also red on this scale (assuming control and expt markers were chosen accordingly) ⟶ red up green down
    - expression scale can be interpreted as **normalized fluorescence data** (corrected for spot intensity) ⟶ can be based on t = 0
- can use **log of expression ratios** to calculate a **correlation coefficient r** (-1.0 to 1.0)

- use this as a distance measure between genes, with 1.0 being an exact match and -1.0 being negatively correlated
- For clustering, Distance D = 1 – r, Range [0,2]

Clustering of Data

- hierarchical clustering can be used to successively combine the closest pair of genes, based on their similarity
- for newly combined pairs, one method of calculating the combined score is to average the correlation scores
- can generate a similarity tree for the genes

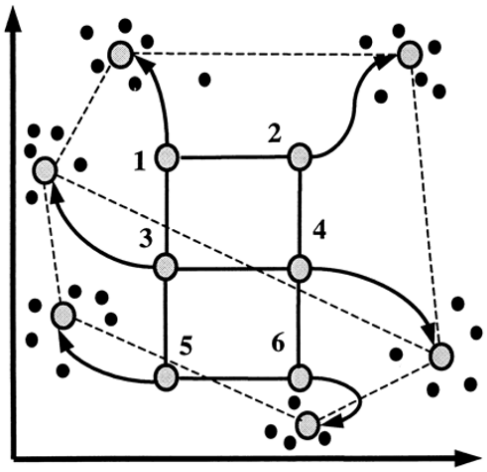
Non-hierarchical Clustering Method - K-Means Clustering

- Data are classified into **k groups**, where k is specified by the user
- useful if expected **number of clusters in known** (e.g. 3 treatments) → you can define this
- Method:
    - each object (gene) is initially **assigned a group** at random
    - **center** of each cluster is defined using a **distance metric**
    - objects are **iteratively reassigned** to clusters to **minimize the within-cluster sum of squared distances**
    - converges after many iterations
    - objects within a cluster contain **similar expression profiles**
- each object (gene) is initially assigned a group at random members are iteratively reassigned until convergence
    - Or, K genes are randomly selected, then all others initially assigned based on distance to the **initial K genes**
- Genes are iteratively reassigned based on group average, until convergence (no further changes at each step)

Non-hierarchical Clustering Method - Self-Organizing Maps (SOM)

- Like k-means clustering, but initial assignment is **not random**
- One chooses a **geometry of 'nodes'**-for example, a 3x2 grid

- The nodes are **mapped** into high dimensional space, and objects assigned to nodes. The **node locations are iteratively adjusted**, along with object assignments.
- Unlike k-means clustering, which is unstructured, SOMs allow one to **impose partial structure** on the clusters.
- The principle of SOMs is as follows
    - One chooses an initial geometry of “nodes” such as a 3 x 2 rectangular grid (indicated by solid lines in the figure connecting the nodes).
    - **Hypothetical trajectories** of nodes as they **migrate to fit data** during successive iterations of SOM algorithm are shown.
    - Data points are represented by black dots, six nodes of SOM by large circles, and trajectories by arrows.



- **Neighboring nodes** tend to define **'related' clusters.**
- An SOM based on a **rectangular grid** is analogous to an entomologist's specimen drawer in which adjacent compartments hold similar insects.
- SOMs may be applied to time-series data, grouping points according to **changes in expression over time** → same group has similar pattern
- there may be **considerable variation** in **number of members** per **node** and **variation of behaviours/expression within a node**

Producing Better SOMs

- **Variation Filtering**
    - Data were passed through a **variation filter** to eliminate those genes showing no significant change in expression across the k samples.
    - This step is needed to prevent nodes from being attracted to **large sets of invariant genes**.
- **Normalization**
    - The expression level of each gene was normalized across experiments.
    - This focuses attention on the **'shape' of expression** patterns rather than absolute levels of expression.

Phylogenetic Methods vs. Hierarchical Clustering

- Hierarchical clustering is quite similar to phylogenetic methods in some respects
    - but phylogenetic follows a stricter set of rules
- One important difference is that branch lengths are not additive for clustering
    - in phylogeny, mutations are cumulative → changes are usually additive and are usually in one direction
    - hierarchical are not cumulative

Which method to use?

- In most cases, this is data dependent → It is often the choice of the researcher → whatever works!
- With clustering there is no objectively correct answer. The goal is to find patterns in the data, and use methods that effectively describe the pattern(s)
- That being said, it is good to consider what assumptions are being built into the data analysis.

Lecture 16: Phylogenetics

Introduction

- Evolution is a process of mutation with selection
    - Molecular evolution: study of changes in genes and proteins throughout different branches of the tree of life
    - Phylogeny is the inference of evolutionary relationships
- Traditionally, phylogeny relied on the comparison of morphological features between organisms
- Today, most phylogenies are constructed using molecular sequence data

Historical Background

- Studies of molecular evolution began with the first sequencing of proteins, beginning in the 1950s.
- In 1953 Frederick Sanger and colleagues determined the primary amino acid sequence of insulin. (The accession number of human insulin is NP_000198)
- We can make a multiple sequence alignment of insulins from various species, and see conserved regions…
    - Mature insulin consists of an A chain and B chain heterodimer connected by disulphide bridges
        - The signal peptide and C peptide are cleaved, and their sequences display fewer functional constraints → changes more
- By the 1950s → amino acid substitutions occurred **non-randomly**
    - most amino acid changes in the insulin A chain are restricted to a disulfide loop region
        - **"neutral"changes** with no observable positive or negative impact
    - the rate of nucleotide (and of amino acid) substitution is about six- to ten-fold higher in the C peptide

Molecular Clock Hypothesis

- In the 1960s → sequence data were accumulated for small, abundant proteins such as globins, cytochromes c, and fibrinopeptides
    - Some proteins appeared to evolve slowly, while others evolved rapidly
- Linus Pauling, Emanuel Margoliash and others proposed the **hypothesis of a molecular clock**:
    - Hypothesis: For every given protein, the rate of molecular evolution is approximately constant in all evolutionary lineages
- EXAMPLE: data from three protein families: cytochrome c, hemoglobin, and fibrinopeptides
    - **x-axis** shows the divergence times of the species, estimated from paleontological data
    - y-axis shows m, the **corrected number of amino acid** changes per 100 residues.
    - n is the **observed number of amino acid changes** per 100 residues, and it is corrected to m to account for changes that occur but are not observed.
        - since we only see the last change → everything before is masked

$$\frac{n}{100} = 1 - e^{-(m/100)}$$

- Conclusion:
    - For each protein, the data lie on a straight line
        - the rate of amino acid substitution remained constant for each protein
    - The average rate of change **differs** for each protein.
        - The time for a 1% change to occur between two lines of evolution is 20 MY (cytochrome c), 5.8 MY (hemoglobin), and 1.1 MY (fibrinopeptides)
        - The observed variations in rate of change reflect **functional constraints** imposed by **natural selection**

Neutral Theory of Evolution

- An often-held view of evolution is that just as organisms propagate through **natural selection**, so also DNA and protein molecules are selected for.
- Motoo Kimura's 1968 **neutral theory of molecular evolution**
    - the vast majority of DNA changes are not selected for in a Darwinian sense
    - The main cause of evolutionary change is **random drift** of mutant alleles that are selectively neutral (or close to neutral)
    - Positive Darwinian selection does occur, but it has a **limited role**
- As an example, the divergent C peptide of insulin changes according to the neutral mutation rate

Goal of Molecular Phylogeny

- How many genes are related to my favorite gene?
- Was the extinct quagga more like a zebra or a horse?
- How did humans disperse over the earth?
- How related are whales, dolphins & porpoises to cows?
- Where and when did a virus (HIV, Zika, …) originate?
- What is the history of life on earth?

Molecular phylogeny in bioinformatics

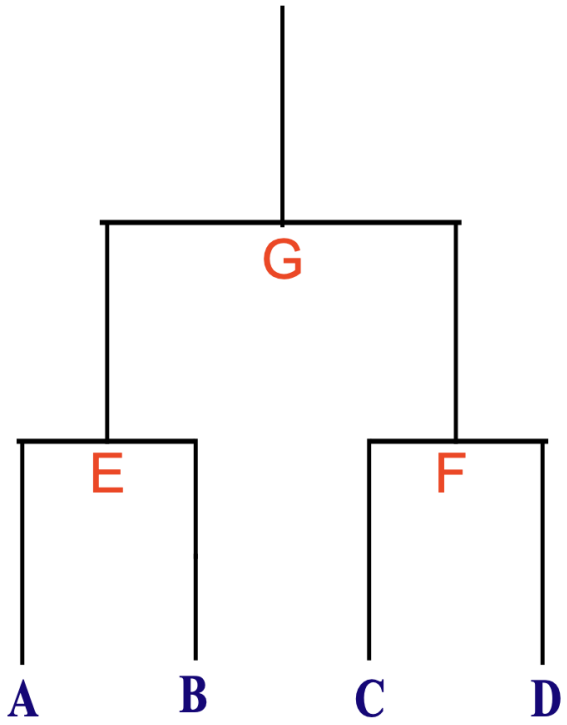
- Many of the topics we have discussed so far involve **explicit** or **implicit models** of evolution.
- Dayhoff et al. (1978) describe the **PAM scoring matrices**:
    - An accepted point mutation in a protein is a replacement of one amino acid by another, accepted by natural selection. It is the result of two distinct processes: the first is the occurrence of a mutation in the portion of the gene template producing one amino acid of a protein; the second is the acceptance of the mutation by the species as the new predominant form
- Feng and Doolittle (1987, p. 351) use the **Needleman-Wunsch algorithm**
    - to achieve the multiple alignment of a set of protein sequences and to construct an evolutionary tree depicting their relationship. The sequences are assumed a

priori to share a common ancestor, and the trees are constructed from different matrices derived directly from the multiple alignment

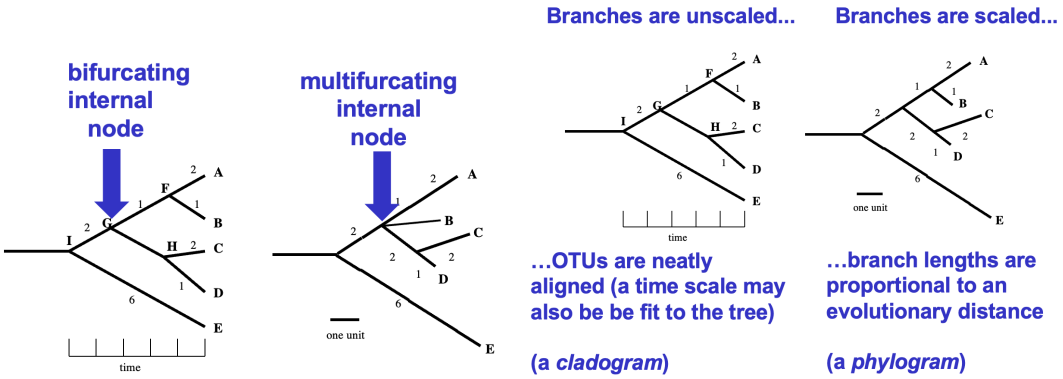
- There are two main kinds of information inherent to any tree: **topology** and **branch lengths**

Nomenclature

- Molecular phylogeny uses **trees** to depict evolutionary relationships among organisms, based upon DNA and protein sequence data
- **Topology**
  - branching pattern of tree
- **Taxon (ABCDEFG)**
  - a distinct group of organisms; a node on the tree
- **OTU (operational taxonomic units)**
  - taxa under consideration = external nodes (ABCD)
- **Internal nodes**
  - hypothetical ancestors (EFG)



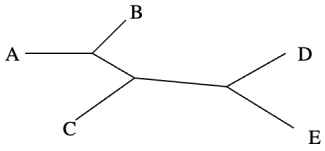
- **Root**
  - **common ancestor** of all OTUs under study (G)
  - **unrooted tree**: no specified common ancestor
  - A tree can be rooted using an outgroup (that is, a taxon known to be d**istantly related** to all other OTUs, but not expected to be within the target group).
- Two types: **bifurcating** internal node (common in evolution) or **multifurcating** internal nodes (uncommon, or represent that we do not know the order)
- Branch can be scaled (**phylogram**) or unscaled (**cladogram**)
- The more OTUs we have, the possible number of rooted and unrooted trees increases exponentially



Tree topology: Newick format

- Graphical representations of trees are the most intuitive
- However, not the best means to compactly store topological and distance data
- One common method for storing tree data is **Newick format**, where topology and distances are recorded as a set of nested brackets specifying a **hierarchy**
- tree topology is specified by nested parentheses

((C,(A,B)),(D,E));



- for **multifurcating nodes**, more than two groups can be included within a bracket:

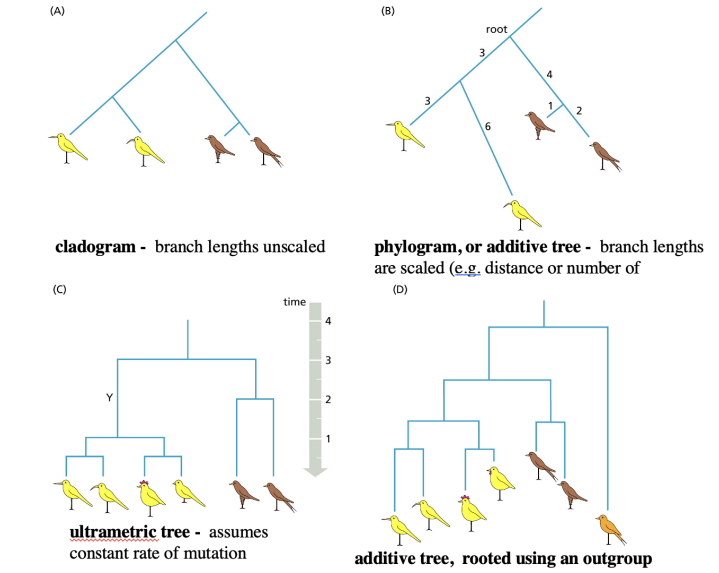
(A,(B,C,D),E)

- branch lengths can be included by following a node with a colon and length

((C:2,(A:1.5,B:1.5)):1,(D:2,E:2):3);

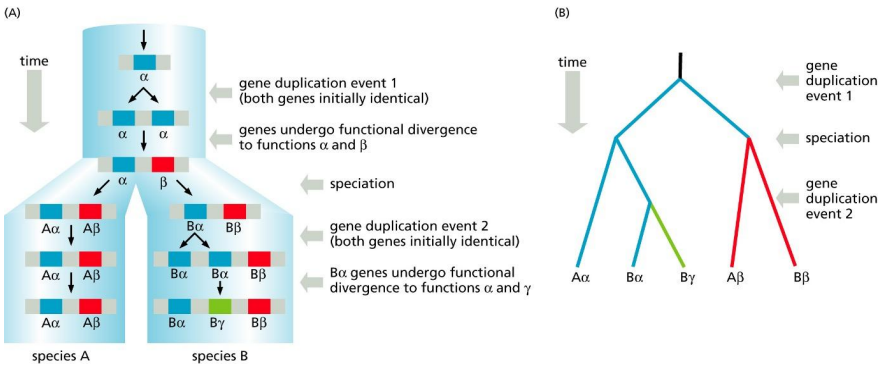
Types of Trees

- cladogram, phylogram (additive tree), ultrametric tree, additive tree



Species trees versus gene/protein trees

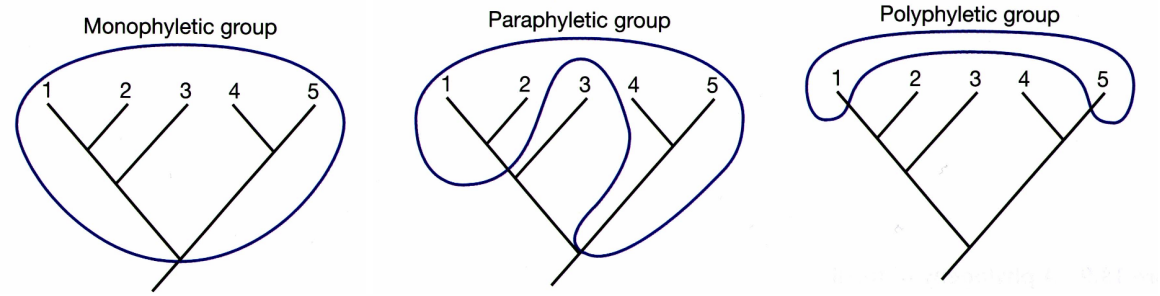
- Molecular evolutionary studies can be complicated by the fact that **both species and genes evolve**.
    - **Speciation** usually occurs when a species becomes reproductively isolated. In a species tree, each **internal node** represents a speciation event.
- **Genes** (and proteins) may **duplicate** or otherwise evolve before or after any given speciation event.
- The topology of a gene (or protein) based tree may differ from the topology of a species tree
- Gene loss and missing data can affect results



Lecture 17: Phylogenetic Trees

Additional Background

- **Monophyletic** - derived from a common ancestor
- **Polyphyletic** - derived from more than one common ancestor → bad!
- **Paraphyletic** - if taxa are derived from a common ancestor but the group does not contain all descendant taxa of the same common ancestor



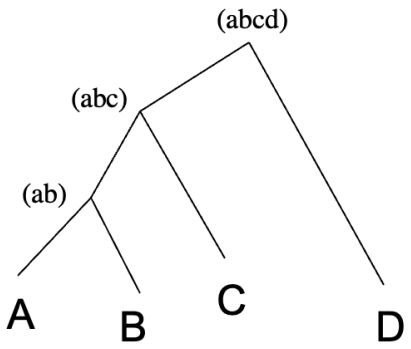
Example of Polyphyletic and Paraphyletic Groups

- **polyphyletic groups** can share similar features as a result of convergent evolution (e.g. winged animals)
- **Paraphyletic groups** are not acceptable in Linnean/ evolutionary taxonomies (try searching for 'Reptilia' in NCBI TaxBrowser)
    - 'Reptiles' are a paraphyletic group
- In general, taxonomists want everything to be in **monophyletic groupings**

Additive and Non-additive Trees

- **additive tree**
    - attempts to preserve the sum of distances between terminal elements (leaves or OTUs) within the tree
- Whether a tree is additive or non-additive depends on the data set
- **Phylogenetic trees** should be additive
    - An issue with UPGMA is that it is not additive

- an example of a non-additive tree would be clustering of species composition data in ecology (no assumption of common ancestor between clustered elements)



$$d(A,B) = d(A,ab) + d(B,ab)$$

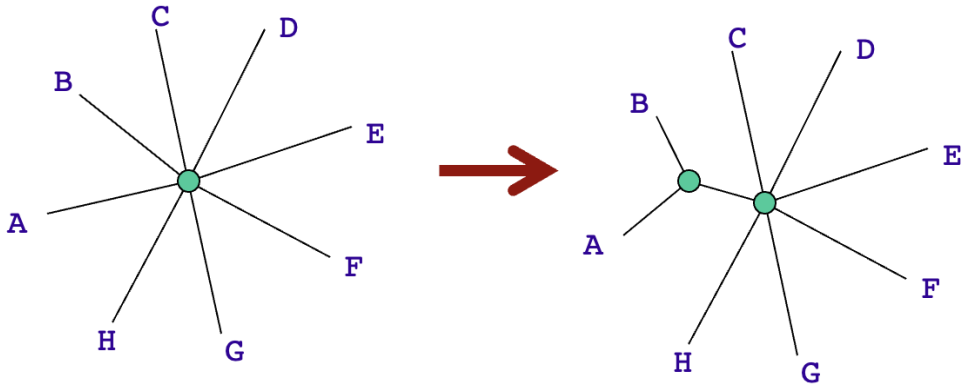
$$d(B,C) = d(A,C) - d(A,ab) + d(B,ab)$$

$$d(A,D) + d(B,C) = d(A,C) + d(B,D)$$

etc.

Neighbour-Joining Methods - Tree Building Method

- attempt to minimize the total length of branches on the tree
    - combine the two closest group, yet the furthest from everything else first
- start with a star-shaped tree
- Among all possible pairs of OTUs (Operational Taxonomic Units), the one that gives the smallest sum of branch lengths is chosen



- These OTUs are regarded as a single OTU and pairwise comparisons are done again to create a new distance matrix
- at each step, all remaining pairs from central node are considered
- repeated until entire tree is generated

- MATH:
    1. for each node, calculate u_i ~ approximate distance from **node to rest of tree**:

$$u_i = \sum_k D_{i,k} / (n-2)$$

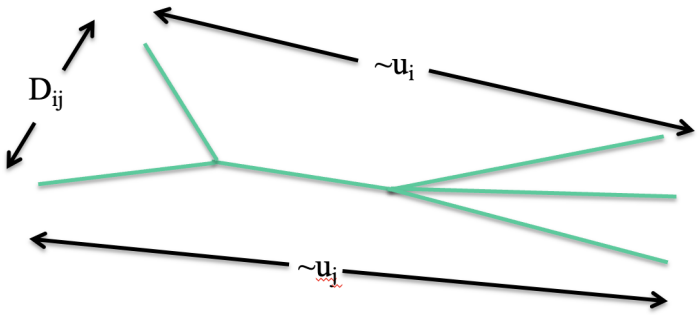
    2. join i, j make a new node (i j), for which $D_{i,j} - u_i - u_j$ is a minimum (minimize distance between two points). This provides the largest reduction in total branch length.
    3. calculate distance for i, j to new node (i j):

$$D_{i,(i\,j)} = 1/2\ D_{i,j} + 1/2\ (u_i - u_j), \quad D_{j,(i\,j)} = 1/2\ D_{i,j} + 1/2\ (u_j - u_i)$$

    4. calculate distance from other nodes to new node (i j) (external distance). The -1/2 Di,j is to ensure **sum is additive**

$$D_{(i\,j),\,k} = 1/2\ (\ D_{i,k} + D_{j,k}\ ) - 1/2\ D_{i,j}$$

    5. remove i and j from tables, recalculate values of u for all remaining nodes.
    6. if there are more than 2 nodes left, repeat steps 1 to 5. Otherwise, connect two remaining nodes with branch of length $D_{i,j}$.
- Criteria for minimization of branch lengths at each step is to choose nodes i and j so that $D_{i,j} - u_i - u_j$ is a minimum.
    - This step reduces the branch length of the tree by the largest amount.



- Depending on the data, UPGMA and Neighbor-joining can combine taxa in **a different order**
- Properties
    - very fast (greedy algorithm)
    - can exactly reproduce tree topology and branch lengths if input data provides exact branch lengths

    - fits well with statistical models of evolution
    - accounts for variation in branch length (evolutionary rate)
    - can produce negative branch lengths
    - not perfect however, due to loss of information in producing distance matrix (like all distance methods)
- Many trees include numbers that estimate confidence in the branch assignment instead of branch lengths

Character based method: Parsimony → Do not build trees, only compare a variety of possible trees

- 'parsimonious' = stingy
- Merriam-Webster definition:
    - exhibiting or marked by parsimony;
    - especially : frugal to the point of stinginess
- in phylogenetic studies, parsimony methods look for the fewest number of changes in an evolutionary tree
    - equivalent to a tree of minimum length
- Similar to all character-based methods, in that a target model is being minimized (here, equal weighting for all changes)
- underlying assumption:
    - mutations are very rare events
    - the fewer rare events, the more likely the model is correct
- like distance based methods, starting point is a **multiple sequence alignment**
- sites within the alignment are defined as
    - **informative** (possessing useful parsimony information)
    - or **uninformative** (no useful information) - e.g. invariant sites are uninformative sites
- EX: Given one site within **four aligned sequences**, there are three possible unrooted trees
- informative sites favour some trees (with fewer mutation events) over others. in general,

    - informative sites need to have at least two different nucleotides
    - and each of these has to be present two or more times
- uninformative sites do not favour any trees
    - can give some different results when compared to distance based methods!
- With more than 4 sequences
    - identify informative sites as before
    - a simple approach is to consider all trees, and count the number of mutations required to produce each tree
    - keep the tree that has the fewest mutations.
    - there are problems with this:
        - the large number of possible trees as you increase number of alignments
        - individual sites (for 5+ sequences) can support alternative trees
        - calculating the number of substitutions for a tree gets complicated

Exploring Tree Space

- In general, character-based methods such as parsimony do not provide a way to build trees, only to compare them
- Other methods are needed to generate news trees to rank and compare (lower **parsimony score**, or shorter branch lengths in an evolutionary model)
- A generalized approach:
    1. Build a starting tree using a fast method (NJ, UPGMA, etc)
    2. Make changes to that tree by moving branches or nodes
    3. Over time, keep trees that have an improved score

Bootstrapping - Measurement of Confidence

- Often used with rapid phylogenetic methods such as neighbour joining
- Computational technique for estimating the confidence level of a phylogenetic hypothesis
- Randomly generates new data sets from the original set (100 or 1000 replicates is common)

- Computes the number of times that a particular grouping (or branch) appeared in the tree
- Calculating Bootstrap Scores
    - start with a **multiple sequence alignment**
    - divide the alignment into a set of **N ordered sites**
    - randomly **choose N sites** from the alignment, with replacement (can choose a particular site more than once)
    - recalculate tree, often 1000 times or more
    - determine the **frequency of each node** within the replicates
- **Bootstrap confidence level**
    - in very common usage
    - useful for interpreting results
    - still should be used with some caution
        - can over- or underestimate confidence (can be corrected for)
    - may provide a better representation than just the 'best' tree
    - **>70% is often used as a threshold**

Lecture 18: Phylogenetic Models

Molecular Phylogeny Models

- Phylogenetic reconstruction would be relatively simple, if:
    - a constant rate of mutation was seen in all branches → so that UPGMA will work
    - sequences are only moderately divergent, so multiple changes are not seen at individual sites
- Unfortunately, neither of these conditions hold for most data sets.
- The following should apply to all reconstructions:
    - an accurate multiple sequence alignment is used
    - an appropriate model of evolution is selected
    - for species trees, aligned sequences are true orthologs
- Ideally,
    - reconstruction method should be appropriate for the data
    - different models of evolution are compared
    - quality of the tree is assessed
- Possible factors in models of molecular evolution:
    - different substitution preferences
        - nucleotide transitions vs. transversions
        - amino acid substitution patterns
    - different rates at different sequence positions
        - different codon positions (1st, 2nd, 3rd) → wobble base at third position
        - different degrees of conservation
    - different rates on different branches of the tree

Measuring Evolutionary Distances

- **p-distance** → simplest model for measuring evolutionary distances
    - aka. fractional alignment distance

$$p = D / L$$

        - D is the number of observed changes

- L is the length of the sequence

- Problems with this measure:
  - for short sequence lengths, statistical variation in p-distance is high
  - does not account for multiple substitutions
    - as the number of mutations per site increase, p-distance levels of in the PAM model → fail to capture the real distance
  - does not account for different rates between sequences

- **The Poisson distance d_p**
  - accounts for multiple substitutions at individual sites.
  - Assuming there is a fixed rate of mutation per unit time, the probability of one position changing is

$$p_{1/2} = 1 - e^{-rt}$$

  - and one of two aligned positions changing is $p = 1 - e^{-2rt}$
  - If distance is defined as $d = 2rt$, then $p = 1 - e^{-d}$, then rearranging and solving for d, $d_p = - \ln (1 - p)$
  - this agrees with p-distance if there aren't many changes, say $p \lesssim 0.2$
  - Poisson distance d_p gets larger as p-distance gets closer to one

Nucleotide models

- the goal for the model is to effectively represent nucleotide changes within a set of sequences
- There is no one 'best' model for all sequence data sets
  - However, the constraints on different sequences will vary, and different models may work best with different data sets.
  - For example, very small data sets might need a simple model, whereas a large data set that is highly functionally constrained may need a more complicated model.
- Can be very simple or complex
  - **Jukes and Cantor** → single parameter model
  - **Kimura** → Two parameter model
  - **Tamura** → Multi-parameter

- A number of models are considered and the one that provides the best match with the data is selected
- **Jukes-Cantor Model**
  - assumption:
    - all sites are independent
    - rates of evolution are the same at all sites
    - all substitutions are equally likely
      - change occur at rate a → chance of the change is a
      - chance of not change: 1 - 3a
    - The change per unit time can be represented in the **rate matrix**

	A	G	C	T
A	-3α	α	α	α
G	α	-3α	α	α
C	α	α	-3α	α
T	α	α	α	-3α

    - **distance d_{JC}** can be derived as $d_{JC} = -3/4 \ln [1 - (4/3) p]$, p=D/L
- **Kimura Model (K2P)**
  - uses different rates for transitions (rate a) (A ⇄ G, C ⇄ T) and transversions (rate b) (A ⇄ C, A ⇄ T, C ⇄ G, G ⇄ T).
    - Transversions are expected to occur at a lower rate (b < a)

	A	G	C	T
A	-2β-α	α	β	β
G	α	-2β-α	β	β
C	β	β	-2β-α	α
T	β	β	α	-2β-α

  - Here, the corrected distance can be calculated as

$$d_{K2P} = -1/2 \ln(1 - 2P - Q) - 1/4 \ln(1 - 2Q)$$

    - P is the observed number of aligned transitions
    - Q is the observed number of aligned transversions
    - total probability of change is p = P+Q
- **HKY85 model** (Hasegawa, Kishino, and Yano, 1985)
  - Both the Jukes-Cantor and K2P model assume a 1:1:1:1 ratio of nucleotide composition. → not always true, as G-T content depends on the environment
    - corrected in HKY85 model
  - Here, the ratios of nucleotides are p_A, p_C, p_G, p_T, and the matrix is

	A	G	C	T
A	$(-2\beta-\alpha)\pi_A$	$\alpha\pi_G$	$\beta\pi_C$	$\beta\pi_T$
G	$\alpha\pi_A$	$(-2\beta-\alpha)\pi_G$	$\beta\pi_C$	$\beta\pi_T$
C	$\beta\pi_A$	$\beta\pi_G$	$(-2\beta-\alpha)\pi_C$	$\alpha\pi_T$
T	$\beta\pi_A$	$\beta\pi_G$	$\alpha\pi_C$	$(-2\beta-\alpha)\pi_T$

- Distances may also be derived from this model, and for more complicated models with different substitution constants
- $p_A + p_C + p_G + p_T = 1$ (three free parameters)
    - so there are five free parameters including a and b

- **Generalized Time-Reversible (GTR) Model**
    - parameters for nucleotide composition p_A p_G p_C p_T and six different rates for all possible reversible transitions and transversions r_{AG}, $r_{AC,}$ r_{AT}, $r_{GC,}$ r_{GT}, r_{CT} .
    - The rate matrix is

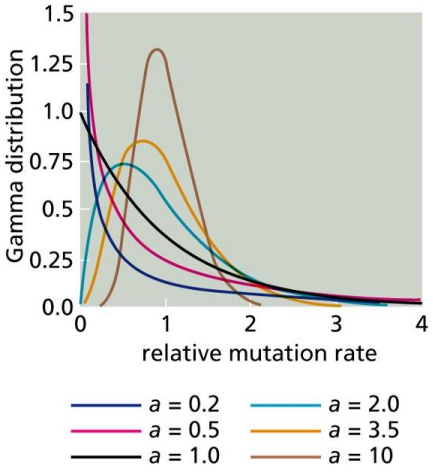
	A	G	C	T
A	$-r_{AG}\pi_G - r_{AC}\pi_C - r_{AT}\pi_T$	$r_{AG}\pi_G$	$r_{AC}\pi_C$	$r_{AT}\pi_T$
G	$r_{AG}\pi_A$	$- r_{AG}\pi_A - r_{GC}\pi_C - r_{GT}\pi_T$	$r_{GC}\pi_C$	$r_{GT}\pi_T$
C	$r_{AC}\pi_A$	$r_{GC}\pi_G$	$-r_{AC}\pi_A - r_{GC}\pi_G - r_{CT}\pi_T$	$r_{CT}\pi_T$
T	$r_{AT}\pi_A$	$r_{GT}\pi_G$	$r_{CT}\pi_C$	$-r_{AT}\pi_A - r_{GT}\pi_G - r_{CT}\pi_C$

    - Distances may also be derived from this model.
    - (p_A , p_C , p_G , p_T) = three free parameters
    - (r_{AG}, $r_{AC,}$ r_{AT}, $r_{GC,}$ r_{GT}, r_{CT}) = six parameters, may be shown as a,b,γ,δ,ε,ζ

- **Common Nucleotide Model**
    - We can use an algorithm to determine the best matrix to fit the data

Model name	Base composition	Different rates of transition/ transversion	Identical transition rates	Identical transversion rates	Reference	Parameters
JC (JC69)	1:1:1:1	No	Yes	Yes	Jukes and Cantor, 1969	rate α
Felsenstein 81 (F81)	variable	No	Yes	Yes	Felsenstein, 1981	rate α comp π_A π_G π_C π_T
K2P (K80)	1:1:1:1	Yes	Yes	Yes	Kimura, 1980	rates α, β
HKY85	variable	Yes	Yes	Yes	Hasegawa et al, 1985	rates α, β comp π_A π_G π_C π_T
Tamura-Nei (TN)	variable	Yes	No	Yes	Tamura and Nei, 1993	rates α1, α2, β comp π_A π_G π_C π_T
K3P (K81)	variable	Yes	Yes	No	Kimura, 1981	rates α, β1, β2 comp π_A π_G π_C π_T
SYM	1:1:1:1	Yes	No	No	Zharkikh, 1994	rates α,β,γ,δ,ε,ζ
REV (GTR)	variable	Yes	No	No	Rodriguez et al, 1990	rates α,β,γ,δ,ε,ζ comp π_A π_G π_C π_T

Correcting for Different Rates at Different Position



- These models still don't account for <u>different rates at different positions</u> (say third position within a codon vs. a highly conserved site).
- This can be corrected using a **Gamma distribution**, which models variation across different sequence positions. This can be applied to a variety of different models, and a corresponding distance measure, the **Gamma distance, d_G** (Uzell and Corbin, 1971) can be calculated.
- For example, for the Jukes-cantor model, $d_{JC} = -3/4 \ln [1 - (4/3)p]$

$$\text{becomes } d_{JC+G} = 3/4\ a([\ 1 - (4/3)p]^{-1/a} - 1)$$

- where a is the gamma parameter
- Different gamma parameter will give you different relative mutation rate (to overall rate of change → is it slower, the same, or faster)

Models of Protein Evolution

- Similar to the previously described nucleotide models, models have been generated for protein sequence data
- Common practice is to use empirically derived substitution matrices, similar to the PAM and BLOSUM matrices → except that those are for phylogenetics, not for searching databases
- Commonly used matrices for phylogenetic studies include **JTT** (Jones, Taylor, Thornton, 1992), **WAG** (Whelan and Goldman, 2001), and **LG** (Le and Gascuel, 2008)

Which is the most accurate tree?

- In the previous example, different values of the gamma parameter resulted in different trees. Some features are present in all the trees, but branch lengths and some groupings are different
- Fortunately, there is a 'best' choice for the gamma distribution – the value that best fits the data, i.e. one that best represents the observed variability within the given set of sequences

Which is the best model to use?

- For our substitution models, there is no universal model that will work best with all data sets, as the parameter weights and which parameters to include are dependent on the data → balance between accuracy and reliability
    - Too few parameters can lead to **inaccuracy**, convergence upon the wrong tree (inconsistency)
    - Too many parameters can **reduce statistical power**, the ability to reject an hypothesis → not reliable
- Fortunately, it is possible to compare the models and determine which is the best quality.
    - Here, '**quality**' is a balance between goodness-of-fit of the model and model complexity.
- Many phylogenetic analysis programs will contain algorithms to assess models (e.g. **Modeltest** and **Prottest** for comparing nucleotide and protein data respectively).

- This is done using some measure of quality, such as the **AIC or BIC (Bayesian Information Criterion)**.
- **The Akaike Information Criterion (AIC)**
    - AIC for each model = -2 ln L + 2k
    - Where k is the number of free parameters and L is the likelihood for the model/data. The model with the lowest AIC is selected

Lecture 19: Maximum Likelihood

Steps for Phylogenetic Data Analysis

1. Selection of sequences
2. Multiple Sequence Alignment
    - simple match/mismatch for alignment of nucleotides?
    - alignment method for protein sequences?
    - how to handle indel events (delete all gaps?)
3. Tree building
    - distance based methods
    - character based methods
4. Tree Evaluation
    - are there any xenologs (horizontal transfer)
    - or substituted paralogs?

Step 1: Selection of sequences

- For **species trees**, selected sequences should be <u>true orthologs</u>.
    - This can be done by using established data sets (e.g. clusters of orthologous groups, COGs, KOGs), or reciprocal BLAST hits, where selected sequences are **reciprocal top matches** between species
    - This can also be verified by inspection of the final tree

Maximum Likelihood

- Likelihood Theory based on work of Fisher (1922)
    - proposed for molecular data by Felsenstein (1973)
- uses a model of <u>sequence evolution</u>
    - what is the probability of <u>obtaining the data</u> (sequence alignment) given the model (tree and model of evolution)
        - **P(data|tree,model)** → "probability of observing the data, given a tree and evolutionary model"

- another way of saying this:
    - given a model of sequence evolution, find the tree and parameters which maximizes the likelihood of the observed data (alignment)
    - a model is always implicit when considering the probability of events

Likelihood vs. Probability

- Given a system, **probability** is the set of possible results given the model
    - eg. if you flip a coin ten times and there is a 50% chance of heads (p=0.50), then you get a distribution of possible results
    - Probability is the predicted result, based on an assumed model
        - e.g. what is the probability of 6 of 10 heads in a coin toss, based on the model P(heads) = 0.5?
- **Likelihood** is based on observed results
    - eg. if you observe 7 of 10 heads, likelihood estimates the parameters, e.g. what is the likelihood of p(Heads) = 0.50?
    - Likelihood predicts the model (parameters) based on an observed set of results.
        - e.g. what is the likelihood of P(heads) = 0.5, given that we have observed 6 of 10 heads in a coin toss?

What is needed for Maximum Likelihood (ML)?

- model of evolution
    - How is your data going to behave?
- JC, Tamura-Nei, GTR, etc.
    - parameters: transitions/transversions, nucleotide changes
- Tree
    - we determine the likelihood for a subset of trees (heuristic) and output the 'best tree'
    - local optimum (unless doing an exhaustive search)

How does likelihood work?

- Because ML assumes independence of sites, the total likelihood is the product of the likelihoods of each column in the alignment

- For a given time period, say the probability of a nucleotide change = 0.05
    - transitions and transversion probabilities are the same (e.g. $P_{A>G}$= αt = 0.05)
    - nucleotide frequencies (Π_A, Π_G, Π_C, Π_T) are included
- The likelihood of a sequence changing from GATCG to GAACA can be calculated with a Probability matrix

	A	G	C	T
A	0.85	0.05	0.05	0.05
G	0.05	0.85	0.05	0.05
C	0.05	0.05	0.85	0.05
T	0.05	0.05	0.05	0.85

$$L(GATCG > GAACA) = \Pi_G P_{G>G} \times \Pi_A P_{A>A} \times \Pi_T P_{T>A} \times \Pi_C P_{C>C} \times \Pi_G P_{G>A}$$

$$L(GATCG > GAACA) = (0.25)(0.85) \times (0.25)(0.85) \times (0.25)(0.05) \times (0.25)(0.85) \times (0.25)(0.05)$$

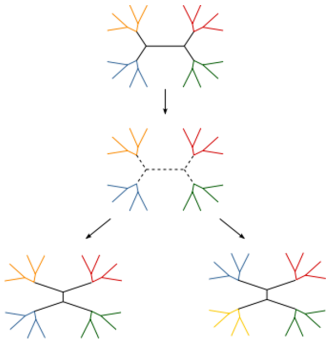
$$L(GATCG > GAACA) = 0.0000014993$$

Related to a tree

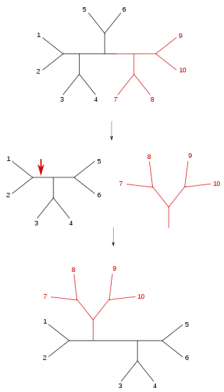
- The example on the previous slide is going from one sequence to another
- When you have more than 3 sequences,
    - each term in the likelihood calculation is expanded to incorporate tree topologies and branch lengths
- It becomes a very large calculation
- **Branch lengths**, or time of evolutionary divergence, have an influence on the likelihood values
    - models of base pair evolution can incorporate this information
- The probability of base pair changes on a very short branch length is low
- extremely long branch lengths eventually see a reduction in the likelihood
    - multiple substitutions at the same site
- How is the tree generated?
    - Maximum likelihood doesn't require a specific algorithm
        - **it compares trees** but doesn't generate them → similar to parsimony
    - There are a variety of algorithms for generating trees for testing
    - Most algorithms utilize a method where new trees are generated from previous trees, such as a '**branch swapping**' algorithm

Exploring Tree Space

- In general, character-based methods such as maximum likelihood (and parsimony) do not provide a way to build trees, only to compare them
- Other methods are needed to generate new trees to rank and compare (e.g shorter branch lengths in an evolutionary model, or a better fit with the observed data)
- A generalized approach:
    - 1.Build a starting tree using a fast method (NJ, UPGMA, etc)
    - 2.Make changes to that tree by moving branches or nodes
    - 3.Over time, keep trees that have an improved score
- Strategy: **Nearest neighbour interchange (NNI)**
    - Each internal branch is connected to four groups within a tree
    - NNI changes how an internal branch is connected (three possibilities total)
    - If the score is better, keep the new tree (or alternately, keep the new tree with some probability, based on the score)

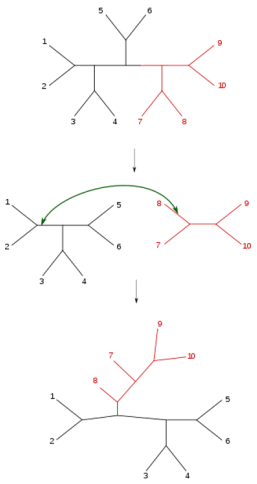


- Strategy: **Subtree Pruning and Regrafting (SPR)**
    - A subtree is cut from the original tree and moved to a new location
        - i.e. the pruned subtree is added to a different branch
    - If the score is better, keep the new tree (or keep the new tree with some probability, based on the score)



-

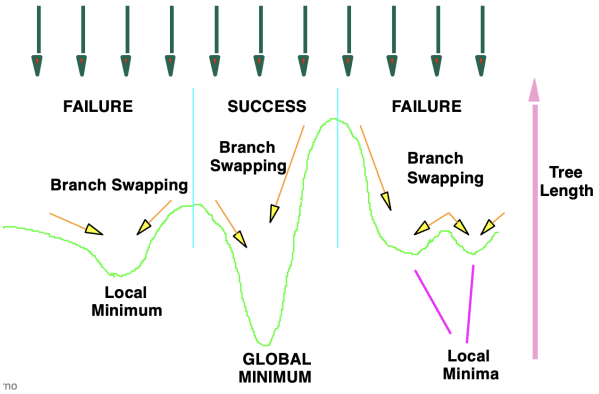
- Strategy: **Tree Bisection and Reconnection (TBR)**
  - Starting tree is split into two
  - All edge combinations for subtrees are recombined, generating a set of new trees
  - As with other methods, if the score is better, keep a new tree (or keep a new tree with some probability, based on the score)



- **Parsimony Ratchet**
  - One issue with tree exploration is that all methods may get stuck in '**local minima**' – there may be a better answer, but you can't get there from here.
  - The parsimony ratchet uses a bootstrap-like approach:

1. Get a reasonable starting tree
2. Re-weight a subset (10-20%) of columns in your MSA
3. Explore new tree space (using TBR or another method) and keep best tree
4. Reset to original weights and explore tree space again
5. Keep the best tree so far.
6. Repeat steps 2-5 (100 times or more)

Tree space may be populated by local minima and islands of optimal trees



Strength and Weakness of Maximum Likelihood

- strengths
    - correct tree given the correct model (and enough time to search all trees)
    - accurate branch lengths
    - uses all data (all data are informative)
- weaknesses
    - computationally intensive
    - inconsistent if wrong model is chosen
- The likelihood of a tree is the **product of the site likelihoods**. Taken as natural logs, the site likelihoods can be summed to give the log likelihood: the tree with the highest likelihood (highest lnL) is the ML tree.

Long branched attraction

- Using parsimony, the model of evolution is simply the number of observed changes. This can result in errors if there are many changes along a specific branch. → ie: data attracted by long branch
- Under the correct model, Maximum likelihood will account for parallel changes and back mutations
- The most appropriate model → USE AIC

General time-reversible (GTR+I+G) likelihood parameters

Branch-lengths: (2n-3 free parameters on unrooted trees)

Substitution rates: α_{AC}, α_{AG}, α_{AT}, α_{CG}, α_{CT}, α_{GT} (5 free parameters)

Base frequencies: π_A π_G π_C π_T (3 free parameters)

Proportion of invariant sites: I (1 free parameter)

Shape of the Γ distribution: α (1 free parameter)

Programs supporting maximum likelihood analysis

- PHYLIP (http://evolution.gs.washington.edu/phylip/software.html)
- PAUP* (http://paup.csit.fsu.edu)
- PHYML (http://atgc.lirmm.fr/phyml/)
- PAML (http://abacus.gene.ucl.ac.uk/software/paml.html)
- TREE-PUZZLE (http://www.tree-puzzle.de)
- DAMBE (http://aix1.uottawa.ca/~xxia/software/software.htm)
- Phangorn package in R (https://cran.r-project.org/web/packages/phangorn/index.html)
- Bayesian methods: Mr.Bayes, BEAST

ML vs Bayes

- Maximum likelihood
    - maximizes the chance of seeing the data given the tree (P(Data|Tree))
    - pick the best tree and see which can reproduce the data
- Bayesian
    - a newer 'framework' for likelihood models → fit the best tree to data
    - maximizes the chance of seeing the tree given the data (P(Tree|Data))
- Note that both require exploring many trees.

Importance of phylogenies

- Phylogenies can also be constructed for gene families, identifying both speciation and duplication events
- general understanding of evolution
- implications for medicine
    - if antibiotic A works well on one pathogen, it is likely to work well on closely related pathogens
- implications for agriculture
    - transfer of disease resistance factors between related species
- implications for ecology/environment
    - Is a population a distinct species? if yes, it may be easier to argue for its protection

Lecture 20: Copy Number Variants (CNVs)

The HapMap Project

- Goals:
    - Define patterns of genetic variation across human genome (SNP and CNV)
    - Guide selection of SNPs efficiently to "tag" common variants
    - Public release of all data (assays, genotypes)
- Phase I: 1.3 M markers in 269 people
- Phase II: +2.8 M markers in 270 people
- Phase III: 1.6 M markers in 1184 people
- **Result**: 10 million common SNPs, haplotype tag SNPs (blocks that are inherited with high relation to SNPs) ~ 300,000-600,000

Origin of Samples - Phase I and II

- African, Asian, European individuals, 270 total
- Populations from HapMap project
    - HapMap project is to characterize common human haplotypes
    - HapMap data used to maximize utility (relate to SNP data)
- 270 individuals, From U.S.A, Nigeria, China, and Japan
    - Yoruba people, Ibadan, Nigeria- 30 trios (2 parents+child) → trio means good for quality control
    - Japan, 45 unrelated Japanese from Tokyo
    - China, 45 unrelated Han Chinese from Beijing
    - U.S. 30 trios from Utah, northern and western European ancestry
- Cell lines created from blood samples, transformed lymphoblastoid cell lines → this cell line is useful for amplification, but might cause quality control issues

Phase III Samples

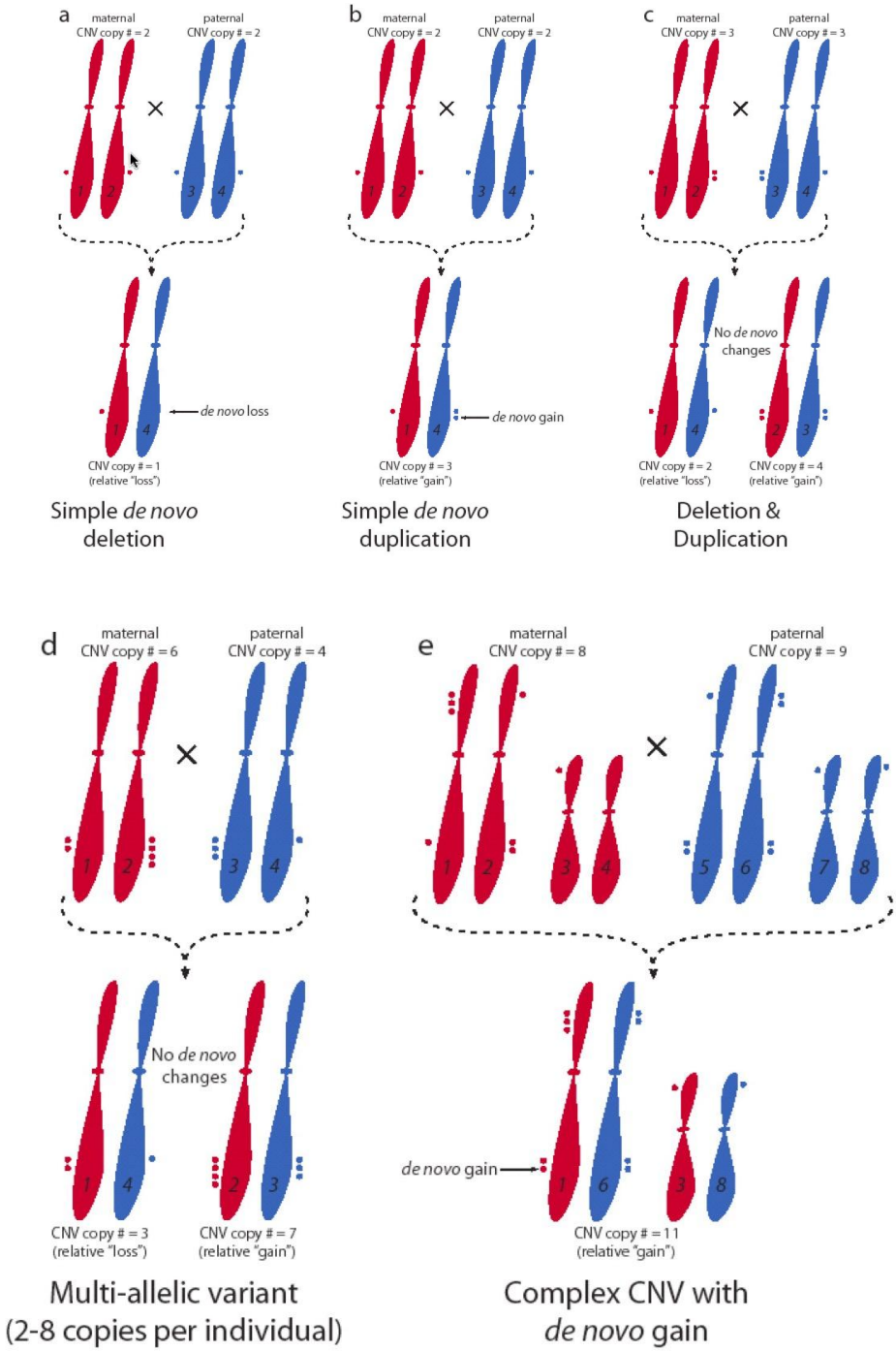
label	population sample	# samples	QC+ Draft 1
ASW*	African ancestry in Southwest USA	90	71
CEU*	Utah residents with Northern and Western European ancestry from the CEPH collection	180	162
CHB	Han Chinese in Beijing, China	90	82
CHD	Chinese in Metropolitan Denver, Colorado	100	70
GIH	Gujarati Indians in Houston, Texas	100	83
JPT	Japanese in Tokyo, Japan	91	82
LWK	Luhya in Webuye, Kenya	100	83
MEX*	Mexican ancestry in Los Angeles, California	90	71
MKK*	Maasai in Kinyawa, Kenya	180	171
TSI	Toscans in Italy	100	77
YRI*	Yoruba in Ibadan, Nigeria	180	163
		1,301	**1,115**

* Population is made of family trios

Copy Number Variant

- CNVs defined as a region 1kb or larger, present in a variable number vs. a reference genome
- can be simple structure (**tandem duplication**) or more complex
- Related to **segmental duplications** identified in reference genome
    - Segmental duplications defined as sequences in the reference genome with >90% similarity over > 1 kb with **another region** in the reference genome
    - segmental duplications defined within a single genome are somewhat arbitrary, as they may represent fixed **duplications** (100% occurrence) or **CNVs** (variable frequency)

Classes of CNVs



Technology for CNV Typing

- Two methods used → both are microarray
    - **SNP genotyping array** → identify SNPs → more SNP = more signal
        - Affymetrix GeneChip Human Mapping 500K array (500K EA)
        - 474,642 SNPs (2 arrays x ~6.5 million 'features' = spots/array)
        - Detected as 25-mer probe quartets, 6 or 10 quartets per SNP
        - Quartet is match + mismatch for each allele
        - $250 per sample
        - Average distance between tagged SNPs - 5.8 kb
    - **Clone-based comparative genomic hybridization** →look for bigger portion from hybridization of DNA
        - Whole Genome Tile Path (WGTP) array
        - 26,574 large insert clones,
        - Coverage is 93.7% of euchromatic portion (easily sequenced portion) of genome
        - average size ~170 kb per clone
        - Human WGTP array designed for this study (Fiegler et al, 2006. Genome Research, Epub Nov. 22)

Quality control

- Samples are from cell lines, it is possible that there are changes in the chromosomal structure within the cell line
    - Cells were **karyotyped** - example below has 5 trisomies (Chromosomes 4,7,9,14,21)
- Chromosomal aberrations appear in CNV assessment (4,7,9,14,21)
    - 30 cell lines of 268 lines had chromosomal abnormalities, r**emoved aberrant chromosomes**
- Investigated HapMap trios for signs of somatic deletions
    - Looked for **SNP deletions** in parental cell lines

        - Deletion in parental cell line identified as cluster of SNPs in offspring but not in either parent
    - In 120 trio parents, 17 of 4758 CNVs consistent with somatic deletions → Suggests ~0.5% of deletions are somatic artefacts
- Additional details
    - dye swap experiments → eliminate dye specific changes
    - in 500K EA chips, probes with central 21 bases matching more than one genomic location were removed
    - statistical analysis, false positive rate set to 5%

Comparison of platforms

- 500K EA platform (SNP genotyping) better at detecting small CNVs
    - average of 24 CNVs in one experiment (one genome)
    - mean size 206 kb
    - median size 81 kb
- WGTP platform better at detecting CNVs in duplicated genomic regions where 500K EA coverage is poorer
    - average of 70 CNVs in one experiment (two compared genomes)
    - mean size 341 kb
    - median size 228 kb
    - larger median size due to overestimation of CNV boundaries (~170 kb is average size of BAC clone)
    - cannot accurately detect CNVs <100 kb

Finding

- 1447 copy number variable (CNV) regions identified
    - 360 Megabases
    - represents ~12% of genome
    - contain hundreds of genes, disease loci, functional elements, segmental duplications
    - much greater nucleotide content than SNPs

    - significant variation in copy number between populations
- 24% are associated with previously identified segmental duplications in reference genome
- Two causes:
    - non-allelic homologous recombination (**gene rearrangement**)
    - previously identified segmental duplications are actually CNVs
- Distribution is overall scattered throughout the chromosomes
- Different CNV types:
    - deletion
    - duplication
    - deletion and duplication
    - multi-allelic
    - complex

Importance of CNVs

- influence expression, phenotypic variation, adaptation
- can disrupt genes, alter gene dosage
- relation to disease:
    - microduplication or microdeletion disorders
    - can be associated with mutations (SNPs) relevant to disease
        - e,g, Parkinson's, Alzheimer's
- major source of genetic diversity, more frequent than SNPs
- genomic distribution
    - distribution of CNVs is not uniform
    - greater density of CNVs near centromeres and sub telomeric regions
    - gaps in sequence are mostly centromeric regions for which no data is available
    - CNV coverage somewhat variable between chromosomes

genomic impact of CNVs

- CNVs contain genes, non-coding RNAs, and conserved elements
- biased away from genes (fewer than expected based on a random distribution) → what gene types?

Gene types in CNVs

- Functional categories enriched:
    - cell adhesion
    - perception of smell and chemical stimulus
    - neurophysiological processes
- Functional categories underrepresented: → fewer change than expected
    - cell signalling
    - cell proliferation
    - kinases
    - phosphorylases
- Underrepresented categories could be dosage sensitive
    - developmentally important
    - oncogenes or tumour suppressor genes

Differences Between Populations

- V_{st} is a measure of population differentiation (similar to F_{st}) → 0 = undifferentiated, 1 = population specific
- used 67 genotyped biallelic CNVs
- average Fst, Vst ~ 0.11, comparable to SNPs, Fst ~ 0.13 → so about **undifferentiated**
- some CNVs are more frequent in certain populations
- none are mutually exclusive → somewhat present in every population