BIOL 266: Computational Biology

Final Exam Review

Unit 0: Introduction

- **What is Computational Biology**
  - The application of computational tools to solve biological problems
  - Under computational biology: **Bioinformatics** - More emphasis on analysis of high-throughput data
- **Tasks of Computational Biology**
  - Pattern discovery -- Learn patterns from biological data
  - Prediction -- Use patterns to predict biological function
  - Integration -- Develop models that connect levels of information
  - Simulation -- Model behavior of biological systems on a computer
  - Engineering -- Design novel biological systems for specific purposes
  - Therapy -- Design molecular therapeutics to combat disease

Unit 1: Molecular Biology and Evolution

- **The genetic material -- DNA**
  - DNA polymers are specific sequences of nucleotides
    - Each nucleotide differs by the nitrogenous base it contains
  - All of the organism's DNA-based genetic instructions make up the **genome**
  - Genome is composed of **genes**, which are DNA instructions for making **proteins**
- **Central Dogma of Molecular Biology**
  - DNA is transcribed into RNA via RNA polymerase
  - RNA is translated into proteins by ribosomes
- **RNA**
  - 3 Types:
    - mRNA - messenger RNA
    - tRNA - transfer RNA
    - rRNA - ribosomal RNA
  - RNA, like DNA, can be single or double stranded, linear or circular
  - Unlike DNA, RNA can exhibit **different conformations**
    - Different conformations permit the RNAs to carry out specific functions in the cell
  - Contains uracil (U) instead of thymine (T)
- **Gene Expression**
  - Use DNA to make mRNA and proteins
  - RNA polymerases look for **promoter sequences** to recognize beginning of genes
  - Prokaryotes use positive and negative regulation for transcription
  - Eukaryotes are much more complex – promoters and enhancers
- **Open Reading Frames (ORF)**
  - Long stretches of DNA that are **un-interrupted by stop-codons** therefore encode protein
  - Gene = ORF + additional regulatory information
  - Start at AUG start codon, run until stop codon (UAG, UAA, UGA)

        - Stop codons are 3/64, or expected about one every 20th codon
- **Protein Structure**
    - **Primary structure**: from sequence and chemical properties of the amino acids
        - **Hydrophobic**: A I L M P V F W
        - **Hydrophilic (polar)**: C N Q S T Y G
        - **Charged**: (-) D E, (+) K R H
- **Sequencing**
    - Determining the exact nucleotide sequence of DNA
    - Methods
        - Maxam-Gilbert Method – chemical degradation
        - Dideoxy (Sanger) Method – chain termination
        - Next-generation (high-throughput) – many types
            - Next-generation (high-throughput) – many types
- **Evolution**
    - Changes in inherited characteristics of biological populations over successive generations, caused by mutations
    - Generates diversity of **genotype** & **phenotype**
    - Types of mutations
        - Point mutation
        - Duplication
        - Insertion
        - Deletion
- **Homology**
    - similarity due to common ancestry
    - The genes and genomes of different species share significant similarity due to homology (common ancestry)
    - Evolutionarily related implies homologous

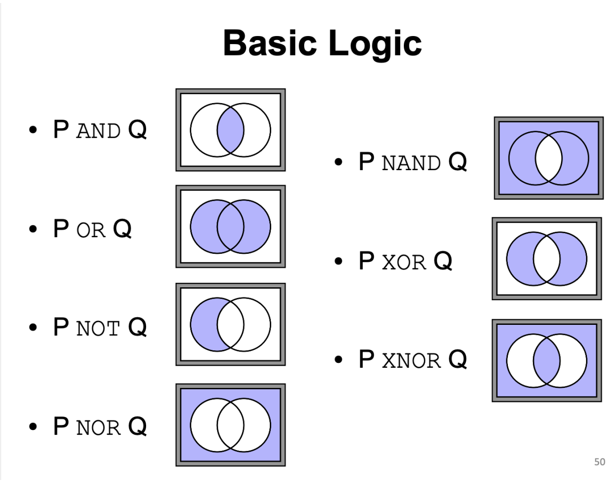
Unit 2: Sequence and Database

- **History of Sequencing**
  - First complete **protein sequence**: in 1955, insulin
  - **Nucleotide sequence**: Development of cloning and then later PCR greatly increased sequencing of DNA
  - **Genome sequence**:
    - Bacteriophage Φ X174 (5386 bp) Sanger et al. 1977
    - First Bacterial genome was sequenced in 1995 (*Haemophilus influenzae*) at TIGR (1.8 Mb)
    - First Eukaryotic genome: *Saccharomyces cerevisiae*
    - Draft of Human Genome (2001)
- **Flow of Information**
  - The data is **curated**, **annotated** and **released** to the public
    - **Core data**: key information in the database entry and minimal information required to identify the data
    - **Annotations**: all additional information, secondary information, may change over time
  - Data may be **re-organized** or **re-annotated** to make it more accessible to users
- **Storage of Data**
  - **Flat File**:
    - Data (e.g. sequences) are stored as a **text file** or a collection of text files
    - flat, as in a sheet of paper → Flat File
  - **Relational File**:
    - Data stored within a number of tables linked together by a shared field, the **key**, to handle large amount of data
  - **.fasta File**:
    - Contains header followed by raw data
- **NCBI/Genbank Nucleotide Database**

- 3 parts: Header, features, sequence (each with **assertion number** and **version number**)
    - **Assertion Number**: Identifiers for specific DNA and protein sequence records
    - Features can be assigned to different regions. It can also include a **feature key** (a keyword indicating the functional group)

- **NCBI/UniPortKB Protein Database**
  - Almost all protein sequences are derived from translation of nucleotide information

- **Accessing the Database**
  - Databases composed of **entries**
  - Programs are designed to match your **query** with entries that are relevant
    - **Query:** name, features, identifier, etc.
    - Yet, using names may not be consistent due to different input
  - In many molecular biology databases you can impose **limits** on your searches
    - Use of **Logic Operators**



Basic Logic

- P AND Q
- P OR Q
- P NOT Q
- P NOR Q
- P NAND Q
- P XOR Q
- P XNOR Q

  - **Controlled vocabularies** or **ontologies** can also narrow searches by clearly defining boundaries

- **Data Quality**
  - Databases are screened to **reduce redundancy** and improve search **efficiency**
  - Databases are under **automatic and manual quality control**

Unit 3: Pairwise and Local Sequence Alignment

- **Sequence Alignment**
    - identification of character matches preserving character order → fundamental tool of bioinformatics
    - A true alignment of nucleotide or amino acid sequences is one that reflects **evolutionary relationship** between two or more sequences that **share a common ancestor** → **homology**
    - Two types of alignment:
        - **Global**: attempts to align the **entire sequence**
        - **Local**: stretches of sequence with **highest density of matches** are aligned
    - Important for discovering functions, structural information, evolutionary information; the result reveals similarity, conservation, and evolutionary relationships
- **Scoring Alignments**
    - Good alignment will have many matches, few mismatches, and few gaps
    - The higher the score, the better the match
    - We use a **scoring matrix** to assign alignment scores
        - Common nucleotide matrix: Identity matrix; BLAST matrix; Transition/transversion matrix
        - Common protein matrix: BLOSUM62
        - Gap may have **origin penalty** and **extension penalty**
- **Computing Alignments**
    - We can look through all alignment possibilities → takes a very long time → **exhaustive approach**
    - Instead, we use **dynamic programing** → solves the problem by breaking it down
- **The Needleman-Wunsch Algorithm for GLOBAL Alignments**
    - http://experiments.mostafa.io/public/needleman-wunsch/
    - Initialize first row and column by multiply gap penalty
    - For a particular cell

        - Take value in cell immediately above add this value to the gap penalty (vertical moves imply a gap)=score
        - Take value in cell immediately to the left add this value to the gap penalty (horizontal moves imply a gap) =score
        - Take value in cell at immediate diagonal and add a match bonus or a mismatch penalty IF the residues match or mismatch respectively=score (e.g. if match =1 then add 1, if mismatch=0 then add 0)
    - Choose the direction that had the highest score and that equals the path that the alignment will go

- **The Smith-Waterman Algorithm for LOCAL Alignment**
    - https://gtuckerkellogg.github.io/pairwise/demo/
    - Very similar to Needleman-Wunsch
    - Uses a **harsher penalty for mismatches**
        - example: match = 1, mismatch = -1, gap = -1
    - One more possibility is added:
        - if the score is **negative**, put in a **zero** instead
    - Find the maximum value in the table, and go backwards from there until you reach a zero

Unit 4: Database Homology Search

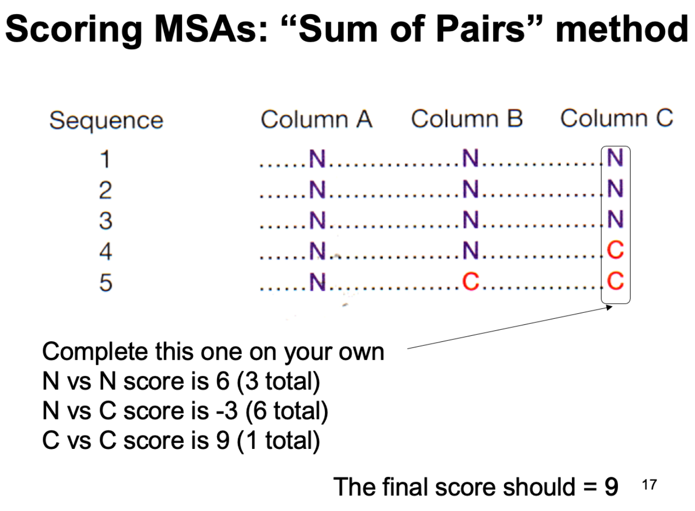
- **Database search**
    - Find homology using one sequence query → align it against all target sequences in the database
    - Each target must be given a score reflecting **degree of similarity**
        - **Bit Score** → score obtained for local alignment → higher = better
    - We then need to estimate the **probability** that the match could of occur by chance (ie: statistical significance)
        - **E Value** → The number of matches with scores **equivalent to or better than S** (bit score) that are expected to occur in a database search **by chance**
        - The closer E Value is to 0, the better
        - Usually**, E < 0.01** (borderline significant),…, E < 1e-10 (highly significant)
        - Significant sequence similarity indicates homology; Yet, non-significant sequence similarity <u>does not indicate lack of homology</u>
    - One method: **SSEARCH**
        - Use S-W against all sequences, and sort by score and probability
        - Problem: speed
- **BLAST - Basic Local Alignment Search Tool**
    - http://www.ncbi.nlm.nih.gov/BLAST/
    - A heuristic procedure → avoids looking at all possibilities
    - **Word-based method (k-tuples)** that initially finds ungapped, locally optimal sequences alignments
        - Could also have larger word length but **permits inexact matches** between words
        - Length is usually 3 for protein and 16 for nucleotide
    - Many types of BLAST from NCBI
        - **blastp**: <u>protein</u> query against <u>protein</u> database (db)
        - **blastn**: <u>nucleotide</u> query against <u>nucleotide</u> db

        - **blastx**: translated nucleotide query against protein db
        - **tblastn**: protein query against translated nucleotide db
        - **tblastx**: translated nucleotide query against translated nucleotide db
        - **PSI-Blast**: detection of remote protein homology using profiles
    - It also conders different **reading frames** → forward 3 and reverse 3 = total 6
    - Simplified procedure:
        - Break query into words
        - Search for (exact) word matches in db
        - Extend the match in both directions until alignment score falls below a fixed threshold (called High Scoring Pairs, HSP)
        - Merge HSPs into longer segments and allow gaps
        - Report E Values and S Values (hit scores)
- **Artifacts about Database Search**
    - Longer sequence = higher score (since more possible matches)
    - Query with repeats, low complexity regions, and short query may also cause problems
    - Always question about your search result!

Unit 5: Multiple Sequence Alignment

- **Multiple Sequence Alignment (MSAs)**
    - Alignment of **more than 2 sequences** at the same time
    - Basics of phylogenetics reconstructions (family trees) to find conservation and variation
    - More complicated as the number of sequence increases, yet it gets more accurate
    - It is computationally difficult
        - **Insertion** of a nucleotide in one sequence requires that a **gap** be added to every other sequence → Causes problem when scoring
        - **Order** in which sequences are added to an MSA can also affect end result
- **Computing MSAs**
    - Challenges:
        - Finding the best alignment that takes into accounts mutations/gaps for **ALL sequences**
        - **Scoring** entire alignment
        - Placement and scoring of **gaps**
        - Can't simply extend N-W or S-W → it will be very slow
    - We use **progressive approach**
        - A form of heuristic approach
        - Build up alignment, and add one sequence at a time
        - Start with **most closely related sequences**
            - May not be the most correct/optimal one, but we hope it is close enough
    - Ex: **ClustalW Algorithm**
        - Align all possibilities using pair-wise alignment → called "**all by all**"
            - Ex: for abc, do ab, ac, and bc
        - Calculate alignment score for each and create a **guide tree** based on scores – closest sequence will be neighbours

        - Progressively align everything based on the location in the guide tree

- **Scoring MSAs**
    - Using **sum of pairs** method for the overall alignment
    - Add the score for all pairs for each column, then sum all the score
    - 

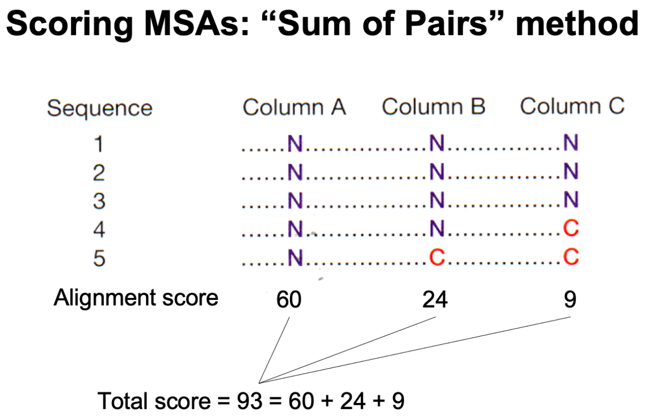
Scoring MSAs: “Sum of Pairs” method

Sequence	Column A	Column B	Column C
1	......N................	N..............	N
2	......N................	N..............	N
3	......N................	N..............	N
4	......N................	N..............	C
5	......N................	C..............	C

Complete this one on your own
N vs N score is 6 (3 total)
N vs C score is -3 (6 total)
C vs C score is 9 (1 total)

The final score should = 9

17

    - 

Scoring MSAs: “Sum of Pairs” method

Sequence	Column A	Column B	Column C
1	......N................	N..............	N
2	......N................	N..............	N
3	......N................	N..............	N
4	......N................	N..............	C
5	......N................	C..............	C
Alignment score	60	24	9

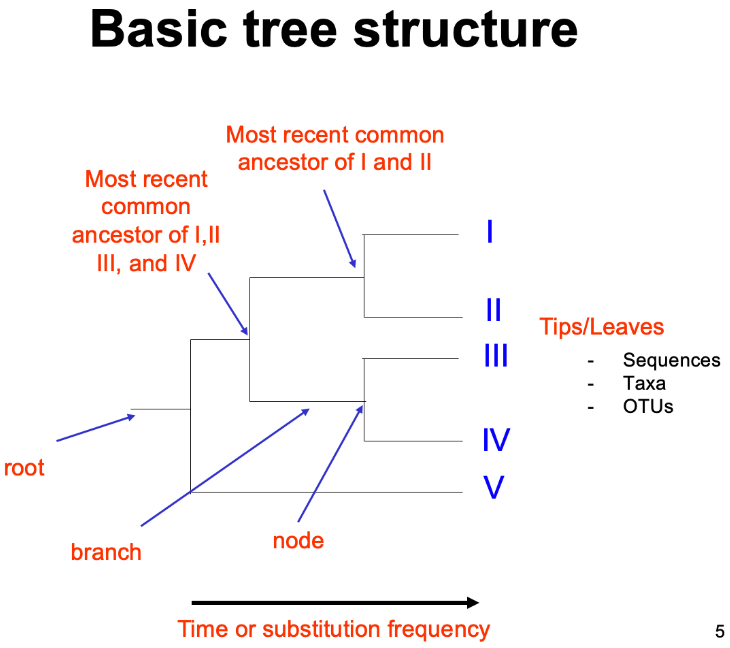
Total score = 93 = 60 + 24 + 9

- **Visualizing MSAs**
    - Colouring by **property** or **conservation**
    - Conservation
        - Evolutionary conservation is plotted for each column
        - Regions of high conservation may be particularly important
        - Variable regions are often less important (but not always, since they may underline evolutionary changes in function)
        - Can done by **conservation profile** or **consensus sequence**

    - Visualizing alignment as **logos**
        - Convenient way of visualizing patterns in a MSA without looking at full MSA
        - Each column of an alignment is represented as **stacked letters**
        - **Height** of letter reflects its evolutionary conservation (more specifically its information content) in the alignment → taller = better conserved
        - Easy to see what regions are conserved/unconserved
- **MSA Programs**
    - **Clustal**: web-based http://www.ebi.ac.uk/Tools/msa/clustalo/
    - Note: MSA and PSA programs will align anything you give it, but it does not always mean that there is anything significant → garbage in, garbage out
- **MSA Databases**
    - We can **precompute** and store alignments of sequences
    - Ex: Uniref50 and Uniref90
        - Clustered sequences with **no more than** 50% and 90% identity
    - Other databases:
        - PFAM – A database of protein alignments
            - Database of protein domain families based on the protein profile of Hidden Markov Models (HMMs)
                - Proteins are often composed of multiple “domains” that are structurally, functionally, and evolutionarily distinct
        - DFAM – A database of DNA alignments
        - RFAM – A database of RNA alignments

Unit 6: Phylogenetics

- **Phylogeny**
    - Hypothesis of the evolutionary history of a group
    - All life forms are related together by descent
        - Use phylogeny to explain diversity
    - A **phylogenetic tree** is a graphical summary of the history evolution (phylogeny)
- **Phylogenetics**
    - Study of **evolutionary relationships** using gene sequences
    - A phylogenetic analysis of a family of sequences may provide information on how the family **diversified during evolution**
- **Species Tree**
    - **Structure of a basic tree**



Basic tree structure

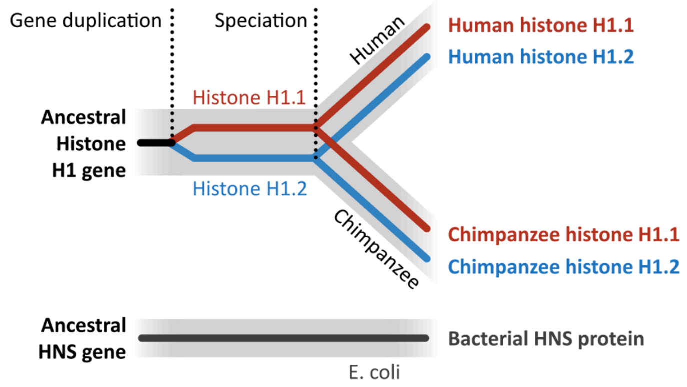
Most recent common ancestor of I and II
Most recent common ancestor of I,II III, and IV
I
II
III
IV
V
Tips/Leaves
- Sequences
- Taxa
- OTUs
root
branch
node
Time or substitution frequency
5

Consists of root, branch, node, and tips

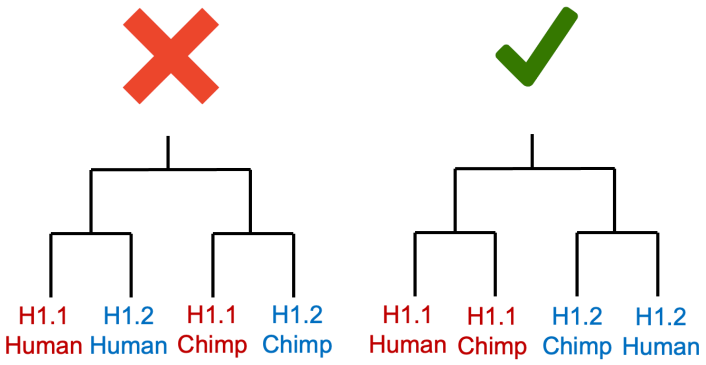
- **Species Tree**
    - phylogenetic tree that represents the evolutionary pathways of a group of species
    - **Nodes** represent **common ancestors**

        - **Bifurcations** (splits from nodes) represent **speciation events**
    - **Scaled** and **Unscaled Trees**
        - **Phylogram**: length means evolutionary distance (scaled)
        - **Cladogram**: only the structure is important, branch length is not (unscaled)
    - **Rooted** and **Unrooted Tress**
        - **Unrooted** if we cannot find the root of the tree
            - we can force the root to be anywhere to produce a **rooted tree** → but that rooting can be right or wrong!
        - To root correctly → use **outgroups**
            - Root is on the branch leading to the outgroup
- **Gene Tree**
    - Based on **molecular phylogenies**
        - Traditionally, phylogenies based on **morphological (phenotypic) traits**
            - Yet, similar phenotype does not always mean homology → might be due to **convergent evolution**
        - Molecular phylogeny is based on DNA/protein alignment across species
            - More reliable and contain more data
    - Gene tree models evolution of a gene family
        - (split from) nodes could represent:
            - **Speciation events**, OR
            - **Gene duplication events**
        - Gene trees can be used to infer species tree

Gene duplication & speciation



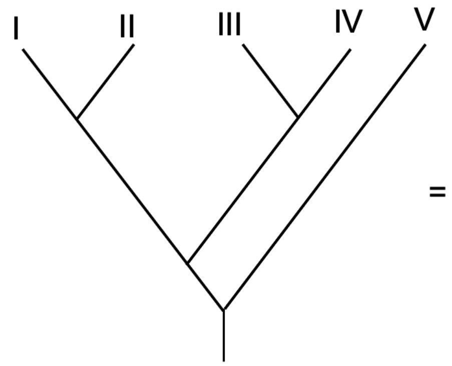
- , and when grouping by homology (share a common ancestry)



-

- **Homologs**
  - Sequences that share a common ancestry (ie: homologous sequences)
  - Types of homologs:
    - **Orthologs**
      - Related by **speciation events** → same gene in different species
    - **Paralogs**
      - Related by **gene duplication** → within or between species
    - **Xenologs**
      - Related by **lateral gene transfer**
  - We want to use **orthologs** to infer phylogeny of a species
    - Usually use **rRNA** – a universally conserved orthologs

- **Store Phylogenetic Trees**
    - Use **Newick Format**

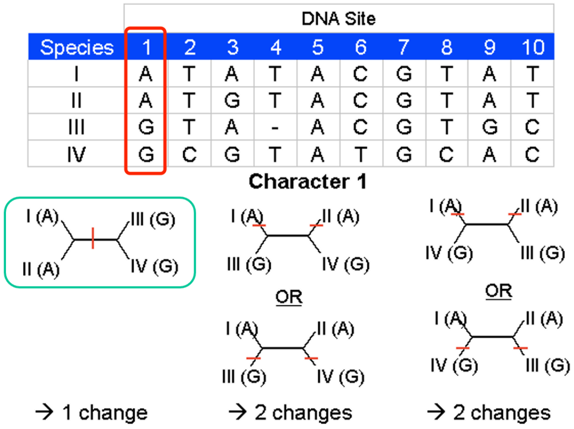


= (((I,II),(III,IV)),V)

    - Order does not matter, only groups does
- **Methods for Tree Reconstruction**
    - We start from **multiple sequence alignment** → the better the alignment, the better the tree
    - **Distance Matrix methods** - compute evolutionary distances and constructs tree based on distances → **Distance based**
    - **Maximum Parsimony methods** - search for shortest pathway leading to character states (tree with shortest length) → **Character based**
    - **Maximum Likelihood methods** - compute trees based on model of evolution and best tree is the one with highest maximum likelihood score → **Character based**
    - No method is guaranteed to produce the correct tree
        - Since results are only **hypotheses**
        - Use multiple methods to compare the results
        - Yet, if both distance and character-based methods produce similar trees, the trees are likely to be of high quality
- **Distance Matrix Methods**
    - Use a distance matrix
    - **UPGMA - Unweighted Pair Group Method with Arithmetic Mean**
        - Developed in the 1950s for analyzing morphological characters (not for tree reconstruction!)

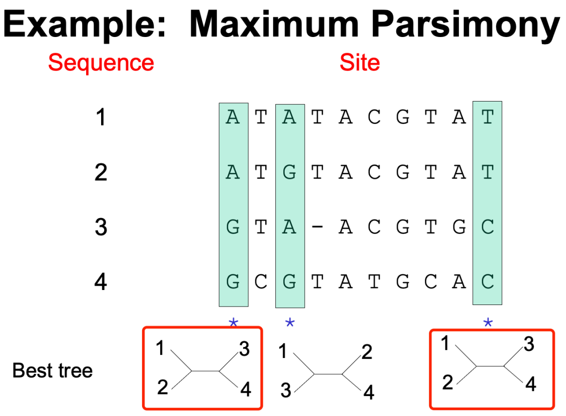
            - Takes data and creates a table of “distances” for all pairwise comparisons
                - % differences between sequences → **P distance** (# difference/# sites)
            - Then reconstruct based on the table
            - Steps
                - Examine sequence alignment and create a pairwise distance matrix of **number of non-matching nucleotides**
                - Find the **smallest distance**, group them
                - Create a new matrix, with the new group in place
                    - The new score is the average of the old ones
                        - Ex: the new group is DE, then A-DE is (D-A + E-A)/2
                - Repeat until all taxa is combined into a tree
            - A clustering method → we are making a lot of assumptions
                - No implication of underlying evolutionary mechanism
                - Tree produced <u>not</u> guaranteed to have the **smallest total branch length**
        - **Neighbour Joining (NJ) Method**
            - Based on **minimal evolution principle**
                - Fewest evolutionary steps are most likely
                - Also used in **maximum parsimony method**
            - Improvement over UPGMA
                - attempts to produce the tree with the smallest sum of branch lengths
            - Among all possible pairs of OTUs, the one that gives the smallest sum of branch lengths is chosen.
            - These OTUs are then regarded as **a single OTU** and pairwise comparisons are done again to create a new distance matrix
- **Character Based Methods**
    - Use multiple sequence alignment directly

- **Maximum Parsimony Method**
    - Find the tree with the fewest changes → **minimal evolution principle**
    - Evaluates **many possible trees** to find which tree(s) are consistent with the fewest # changes
    - We use **phylogenetically informative sites** to find the better trees
        - **invariant sites** (completely conserved) are **uninformative** → those sites do not tell us which tree is better
        - Informative sites, in general, must have:
            - at least **two different** characters (nucleotides or AAs)
            - each character has to be **present more than once**
    - Steps:
        - Identify **how many possible trees** exist for the data set (4 taxa = 3 unrooted trees)
        - Examine each **informative site** and determine which tree is preferred
            - Preferred: fewest number of ancestral substitutions



	DNA Site									
Species	1	2	3	4	5	6	7	8	9	10
I	A	T	A	T	A	C	G	T	A	T
II	A	T	G	T	A	C	G	T	A	T
III	G	T	A	-	A	C	G	T	G	C
IV	G	C	G	T	A	T	G	C	A	C

            -
        - For all informative sites in the entire alignment tally the number of times each tree is preferred



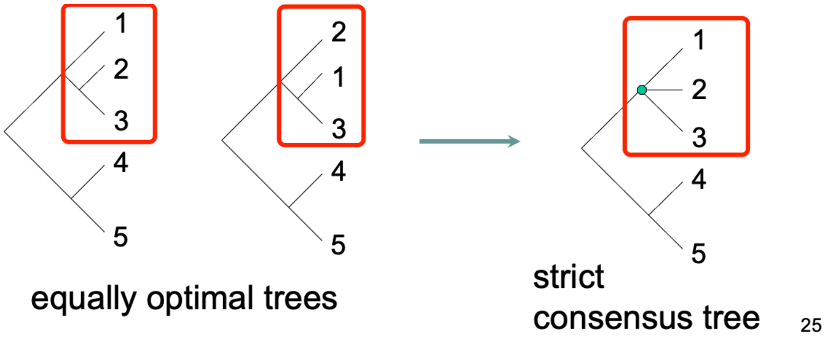
Example: Maximum Parsimony

Sequence	Site									
1	A	T	A	T	A	C	G	T	A	T
2	A	T	G	T	A	C	G	T	A	T
3	G	T	A	-	A	C	G	T	G	C
4	G	C	G	T	A	T	G	C	A	C
	*		*							*

Best tree

-
- **The one with the greatest number in the tally is the most parsimonious tree**
    - you can have more than one solution
- **Maximum-Likelihood Methods and Bayesian Methods**
    - Both look through many possible trees to find the best one
    - **ML:** Finds the tree that maximizes the **probability of the alignment** (probability of data given a certain tree)
    - **Bayesian:** Finds the tree that maximizes the **probability of the tree** given the data

- **Consensus Trees**
    - Parsimony and, to a lesser extent Likelihood methods can sometimes produce **many <u>equally</u> optimal trees**
    - Consensus tree **combines ambiguous nodes** within equally optimal trees
        - **strict consensus** (all equivalent trees agree), or
        - **'majority rule' consensus** (more than half the trees agree)

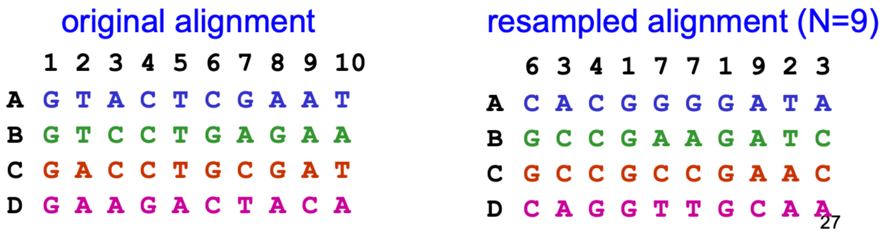


equally optimal trees

strict consensus tree

-

- **Quality of Trees**
  - **Bootstrap Confidence**
    - Steps:
      - 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



original alignment

	1	2	3	4	5	6	7	8	9	10
A	G	T	A	C	T	C	G	A	A	T
B	G	T	C	C	T	G	A	G	A	A
C	G	A	C	C	T	G	C	G	A	T
D	G	A	A	G	A	C	T	A	C	A

resampled alignment (N=9)

	6	3	4	1	7	7	1	9	2	3
A	C	A	C	G	G	G	G	A	T	A
B	G	C	C	G	A	A	G	A	T	C
C	G	C	C	G	C	C	G	A	A	C
D	C	A	G	G	T	T	G	C	A	A

27

    - Record Bootstrap confidence level:
      - the **percentage of times** that clade is present in the collection of trees → you want as close to 100 as possible
      - The less supported (low bootstrap score) groupings can be "collapsed" (ungrouped) so that we don't make unsupported claims about their order of splitting

Unit 7: Structural Biology

- **Protein Structure**
  - Crucial to understanding how a protein works, and provides a framework for explaining molecular biology
  - Organization of structures
    - **Primary:** linear sequence of amino acids
    - **Secondary:** α-helix (A), β-sheet (B), β-turn (C)
    - **Tertiary:** overall three-dimensional shape of a polypeptide chain
    - **Quaternary:** two or more polypeptide chains held together by non-covalent forces, in precise ratios with a precise 3D configuration
  - **Sidechains** of Amino Acids
    - What makes protein unique and determines the fold
    - Vary in size, charge, polarity, and shape
  - **Hydrophobicity**
    - One of the governing principles of protein structure
    - Non-polar side chains are similar to oil-like solutes
      - Interaction with water is unfavourable
    - **Hydrophobic collapse:** folding nuclei formed by core hydrophobic residues
      - Charged AAs are often excluded from the protein interior
      - Exterior is mostly charged, yet you still do find a lot of charged side chains (about 1:1 charged: uncharged)
- **Protein Data Bank**
  - Stores .pdb files → **3D atomic resolution** of a molecule and a 4 digit identifier
  - Different structural visualizations:
    - **Ball and stick**
      - **main chain bolder** than side chains
      - Sometimes represented as "ball and stick"
      - Gives a lot of information

        - **Ribbon**
            - course of the chain is represented by **smooth interpolated curve**
            - chevrons indicate chain direction
            - only gives the information of the backbone
        - **Cartoon**
            - cylinders represent helices
            - arrows represent strands of sheet
            - easy visual information about secondary structure
- **Structural Determination**
    - Proteins generally fold into **single, stable 3D structures** based on their sequence
    - Lowest energy state, most stable → **native state**
    - Determined by **favorable interactions** within/between amino acids
    - Structure can be determined **experimentally** and in some cases reasonably approximated **in silico** (using computers)
    - **Experimental Determination**
        - X-ray crystallography
        - Nuclear magnetic resonance (NMR)
        - Cryo-electron microscopy (cryo-EM)
    - **Prediction of Structure**
        - **Ab initio** → from first principles
            - **Energy minimization** → Compute energies associated with structures → computationally very challenging
            - Issues: **Local minima traps**, single domain vs multi-domain, energy functions
        - **Comparative modeling**
            - **Template-based**, uses existing fold information from PDB
            - **threading** (fold-recognition) and **homology modeling**
- **Ab Initio**
    - Based on first principle of **Energy minimization**

    - find the most stable conformation (3D position of all atoms) based on **energy functions (CHARMM)**
    - Parameters include **bond angles** and **interactions between atoms**:
        - Primary chemical bonds
        - Weaker interactions
    - Find the “folding tunnel”
    - **Computational issues**
        - Are the parameters complete/accurate?
        - Search space is massive
    - Simple proteins can usually be modeled based on few parameters
    - Larger proteins usually have **too many parameters** to examine exhaustively, therefore heuristic approaches must be used
    - **Local Minima Issue**
        - Not looking at all possibilities (heuristic methods) means falling into local optimums
        - Impossible to distinguish from global optimum (if optimum is unknown)
- **Solution to Local Minima Issue**
    - **Steepest descent**
        - estimates energy at current conformation
        - changes coordinates to **move directly down gradient**
        - Stop when can’t go any lower -> no global
    - **Conjugate gradient**
        - uses **two successive gradients** to make an intelligent guess at the location of the minimum
    - **Newton-Raphson**
        - gradient of the gradient (**second derivative**)
        - works well, but computationally expensive
    - **Monte Carlo Procedure**
        - Uses random search

        - Useful for finding the minimum of a **function of many variables**
        - Steps:
            - Generate **random set of values** for variables (i.e., a random conformation)
            - Perturb variables to generate a **neighbouring conformation**
            - Calculate the energy of the new conformation
            - Decide whether or not to accept the change or try another one
                - If energy **decreases** (i.e., the step creates a better state), then <u>accept</u> (the perturbed conformation becomes the new current conformation)
                - If the energy **increases** or **stays the same**, <u>sometimes accept</u> the new conformation
                    - this helps avoid local optimum solutions
                    - Allows the temporary movement to 'worse' solutions
            - Go back to step 2 and repeat until exit condition
    - **Levinthal's Paradox**
        - Does nature really explore all possible protein folds until it finds the lowest energy one? Because that would take very long… yet protein folding is FAST
- **Comparative modeling**
    - **Homology Modelling**
        - Most reliable method of modeling protein structure
        - Requires detectable **sequence homology** to existing structures
            - These structures are used as **templates**
        - At least 40-50% identity required
            - But higher identities are much better (e.g., 75% +)
        - **E-value** must be significant as well (of course)
        - Use of **multiple template** can increase accuracy

        - Structurally reliable alignments rely on **sequence identity and length:**
            - Shorter sequence needs higher minimum identity
        - Steps:
            - Template selection (e.g., top BLAST match from PDB)
            - Align target to template
            - **Generate backbone as template**
            - Loop modeling (insertion/deletion)
                - Variations between the template and target sequences are most likely in loop regions
                - Deletions easier than insertions → just remove it
                - Insertions modeled as loops → energy minimization
            - Side-chain modeling
            - Model optimization
            - Model validation: if poor quality, go back to (1) or (2)!
    - **Fold Recognition - Threading**
        - When template is not present
        - There are only about 2000 ways that a protein can fold → same fold can occur for many different proteins
        - Basic idea:
            - For each possible fold structure
                - pull string of amino acids (target) through fold
                - **examine (score) the compatibility** of each amino acid with that fold
            - If score is significantly high, **template is assumed to fold in much the same** way as that structure
        - try many alignments and try all templates, to see which model is the best
- **Evaluating Model Quality**
    - **Force Fields**
        - Residues in energetically unfavoured environment; energy minimization

    - **Ramachandran plot**
        - Main chain structure can be approximated using the sequence of 3 angle values for each amino acid
            - N-Caplha (phi)
            - Calpha-C (psi)
            - Angle of rotation around the peptide bond (either trans or cis)
        - The plot separates into areas of **possible and preferred conformations** for amino acid residues
        - Areas of the plot indicate likelihood of alpha-helices and beta-sheets
        - See if the structure falls into expected region of bond angles
- **Structural Alignment**
    - How similar our structure is to other structures
    - This can sometimes, **but not always**, be inferred using sequence homology (i.e., BLAST)
        - **Structure is more conserved** than sequence information throughout evolution
            - Sometimes, structures might be unidentifiable at the sequence level, but still have similar structure
        - Sometimes aligning sequence information without structural data is misleading
            - Can miss homologies, especially if distantly related
    - Distantly related proteins can be detected based on conserved spatial contact patterns between residues
    - Successful in finding very **distant evolutionary relationships**
    - Two commonly used tools:
        - DALI and VAST (**Vector Alignment Search Tool**)

Unit 8: Genomics

- **Genome**
  - full complement of genetic material within an organism or cell/tissue sample of interest
  - Genome sequencing encompasses:
    - Organelles, plasmids, viruses, prokaryotes, eukaryotes
    - Single cell sequencing, cancer genomics
    - Environmental DNA samples (metagenomics – Collection of genomes)
  - Fundamental problem: A fundamental problem of genomics is the Genotype-to-Phenotype problem: still largely unsolved
  - Steps in genome analysis:
    - Selecting an organism → collect sample → sequencing → genome assembly → genome annotation
- **Selecting the Genome**
  - The selection is based on:
    - Genome size
    - Cost
    - Relevance (disease, biological question, agriculture, etc.)
  - Can also sequence **1 individual** or **multiple individuals**
    - Multiple individuals example: **1000 Genome Project**, look for genetic diversity by examining 1000 individuals' whole genome
- **Sequencing the Genome**
  - 2 main approaches:
    - **Whole-genome shotgun sequencing (WGS)**
      - first done by Sanger on Bacteriophage Φ X174 and then used by Venter and Colleagues (Celera)
    - **Hierarchical shotgun sequencing (more traditional)**
      - Divide the genome up to regions and line them up
  - Terminologies

  - **Read** – an individual sequence fragment (output by sequencer) – often short
  - **Contig** – set of overlapping clones/sequences/reads from which a longer sequence can be obtained. Contigs are derived from assembling the reads (but not necessarily the whole genome)
  - **Scaffold** – ordered set of contigs placed on the chromosome (may contain missing sequences and gaps)
  - **Draft sequence** – incomplete sequence of the genome (more sequencing still in progress) (most in NCBI database are draft sequences)
  - **Finished sequence** - genome is completely sequenced with no gaps
  - **STSs: Sequence-tagged sites**
    - short segments of unique DNA sequence on a chromosome
    - usually defined by a pair of PCR primers that amplified only one segment of the genome
    - used as 'road markers' on the chromosome → orientation
  - **ESTs: Expressed Sequence Tags**
    - unlike STSs, ESTs are from transcribed regions (regions that made mRNA)
    - short segments (<500 bp) from cDNA
    - identify coding regions
  - **RNA-seq**: related approach that sequences the full complement of expressed transcripts in a sample
- **Shotgun Sequencing**
  - **Random fragmentation** of genome by shearing or restriction
  - Universal primer used to sequence random selection of fragments
  - Sequences assembled into contigs
  - Gaps are targeted for additional specific sequencing; Overlaps are the original sequences
  - Can only work alone → since otherwise there will be repetition work

- **Hierarchical Genome Sequencing**
    - Also called: top-down, map-based, ordered clone, clone-by-clone
    - **Breaks down genome** into smaller and smaller pieces → Divide into large segments of known orders
    - Allows for:
        - assembly of <u>high resolution</u> physical and genetic maps
        - <u>global groups</u> to work together without repetition
- Example: **Human Genome Project**
    - Used **hierarchical sequencing**
    - **Restriction enzymes** used to chop chromosomes into pieces
    - Pieces inserted into vectors, for replication in
        - *E. coli* : Bacterial Artificial Chromosomes (BACs), about 150 kb
        - Yeast : Yeast Artificial Chromosomes (YACs), 150 kb to 1.5 Mb
    - Restriction maps and common STSs used **to identify overlapping BACs and YACs**
    - Assembled into contiguous overlapping segments of DNA (**contigs**)
    - STSs used to locate contigs on chromosome
    - **Public and Private Genomes**
        - There were two draft versions of the human genome,
            - public (Human Genome Project) and private (Celera)
        - Public database is more accessible (i.e. free)
        - Private used public data as well
        - Private effort likely ‘motivated’ public effort
        - Few differences between the initial versions
        - Full human genome sequence completed April 2003

- **Finishing the Genome Assembly**
    - Raw genomic information are submitted to NCBI through the HTG sequence division and sequences are categorized into 4 phases
        - 0,1,2 = unfinished; 3 = finished

- Genome is finished when 5-10 fold **coverage** (but much higher these days)
    - **Coverage**: average time that each base is covered by the reads
- Greatest difficulty is repetitive elements

- **Genome Annotation**
    - the process by which the key features of the genome are described
    - Includes:
        - Basic **genome stats**: Genome size, # chromosomes, GC content, etc.
        - **Location** of non-coding region
        - **Location** of protein-coding genes (introns/exons)
            - *De novo* or *Ab-initio* methods
            - Empirical
                - EST/mRNA based
                - Homology-based (ex: blast)
        - transcription start sites, promoters, RNAs, regulatory elements, repetitive elements, etc.
        - What are the **functions** of the genes and other genomic elements?
- **Prokaryotic Genome Annotation**
    - First, look to **non-coding regions**
        - e.g., rRNAs, tRNAs – common structure → tend to be easier to find
        - Remaining sequence can then be scanned for protein-coding genes
        - **rRNA genes**
            - can have many copies in the genome
            - **well characterized** that they are easy to distinguish
        - **tRNA genes** (often >50)
            - The complement of tRNA genes indicates **codon preferences**, which makes protein coding gene detection easier
    - **Detection of tRNA genes using tRNAscan**
        - tRNA genes have **highly conserved structure**
        - Algorithm developed using **alignment of many tRNA sequences**

      - identifying regions of high sequence and structure conservation
    - Uses a **decision tree** – see if that sequence is consistent with the tRNA pattern
      - at each step in the procedure the sequence has to pass a test
      - in tRNA, the paring sites are **very conservative** since they hold the structure together → invariant bases
      - Also has allowable insertion sites → variable length
      - tRNAscan looks for pre-defined feature → once if failed, it shifts the sequence and tries again
      - The question gets more specific as you move on
    - Effective:
      - Predicts 97.5 % of tRNA genes
    - Accurate:
      - one false positive/3 million bases
    - very good for prokaryotes
    - error rate too high for eukaryotes → modified algorithm for eukaryotes
  - Gene density is high with prokaryotes → about 85% - 88% nucleotides are within coding regions
  - # of genes varies (several usually thousands), yet minimal set of genes for absolute survival is usually from 30 – 150
  - Genes with **related functions** are often grouped within an **operon**
    - several genes with **one shared promoter**
    - one RNA transcript for all genes in operon (**polycistronic RNA**)
- **Looking for Genes in Prokaryotic Genome**
  - Relatively easy compared to eukaryotes
    - <u>Lack of introns</u> simplifies process of gene finding
  - Genomes are **circular** and there is typically **one gene for each KB** of genomic DNA
  - Matches to simple conserved promoter sequences

- Features used to fine genes:
  - **Open reading frames**
    - ORFs are stretches of DNA with **no stop codons** for a particular reading frame
    - The <u>longer</u> the potential ORF, the <u>more likely</u> it is to really be a gene
    - One **stop codon** is **expected every 20-25 codons** in random sequence
      - The likelihood of internal stop codons occurring in a random sequence increases with its length
      - ORFs longer than 60 codons have <5% chance of being a result of chance
    - Defined by a **start codon** (typically AUG) and a **stop codon** (UAA, UAG, UGA)
      - There are exceptions to standard codons (e.g., E. coli uses GTG for 9% and TTG for 0.5% of start codons)
  - **Sequence motifs/patterns indicative of genes**
    - **Shine-Dalgarno sequence**
      - upstream of start codon
      - May find multiple in frame start sites
      - identifying a <u>ribosome binding site</u> can be an important indicator of likely start site
      - In bacteria, it is a sequence that is complementary to the 3’ end of the SSU rRNA (5’-AGGAGGU-3’)
    - **Transcription initiation sequences**
      - Pribnow box **(-10)** sequence: **TATAAT consensus**
      - **-35** sequence: TTGACA consensus
  - **Codon Usage**

            - Protein-coding genes possess a **distinct codon usage profile** (“signature”) that can distinguish them from non-coding DNA
            - the frequency occurrences of different amino acid codons in genes and intergenic (non-coding) DNA are different
            - Can be used as a **gene-prediction feature**
        - **Homology to known genes**
            - **Putative genes** (predicted ORFs) can be compared to databases
                - BLAST against NCBI, etc.
            - Becomes more effective as databases get larger
    - Pitfalls with Prokaryotes Gene Predictions
        - Difficult to distinguish whether <u>short ORFs</u> are genuine ORFs or are false positives
        - Partial genes
            - Sequencing errors?
            - Pseudogenes?
            - Frameshifts?
        - It is relatively easy to find genes in prokaryotic genomes, but can be much **harder to assign them function**
- **Eukaryotic Gene Annotation**
    - **Differences between Euk and Prok**
        - <u>Scale</u> of analysis is much larger
        - <u>Gene structure</u> causes eukaryotic gene detection to be much harder
            - Eukaryotic genes contain **introns** and **exons** due to **splicing**
            - <u>Length of the exons</u> is on average <u>smaller</u> than in prokaryotes making ORF recognition more difficult
        - Lower gene density
            - E.g., 98.5% of human genome is non-coding DNA → coding sequences are rarer and harder to detect
        - Abundance of **repetitive sequences**

    - “junk DNA” → These can lead to errors in gene prediction and genome annotation
- **Introns and Exons**
  - Most protein coding introns follow GU-AG rule:
    - **start** of intron is 5’-GU-3’
    - **end** of intron is 5’-AG-3’
    - additional recognition sites within intron also available
  - Length <u>minimum</u> is **~60 bp**, no real upper bound
  - Introns are less common in simple eukaryotes
  - About 95% human genes contain introns
  - Exons are shorter than that of prok, but both the length of introns and exons can vary
- **Alternative Splicing**
  - Majority of eukaryotic genes appear to be processed into a single mRNA
  - However, **over 50%-75% of human genes alternatively spliced**
  - Alternative splicing - depends on a cell type and other regulatory factors, one gene can produce different mRNA to make different proteins
- **Repetitive Elements**
  - Many DNA regions contain **repetitive sequences**
    - can be removed from dataset to simplify gene finding
  - Typically, large repetitive chunks are divided into:
    - **tandemly repeated DNA** (ex: 5’ CTCTCTCT 3’)
      - **Satellite DNA**
        - long, simple sequences (up to 10mbp) with skewed nucleotide compositions
        - repeating fragments of up to 2,000bp
      - **Minisatellite DNA**
        - not as long as satellites (up to 20kbp)
        - copies of sequences of up to 25bp

- **Microsatellite DNA**
  - shorter than minisatellites (up to 150bp)
  - up to 100 copies of sequences of up to 5bp (typically 2-3)
  - “TAGTAGTAGTAGTAGTAGTAG..."
  - Example: humans, **‘CA’ repeats**
    - occur once every 10,000bp
    - make up 0.5% of human genome
- repeats that are interspersed throughout the genome (e.g., LINE and SINE elements)
- **Eukaryotic Gene Regulation**
  - Eukaryotic **promoters** more variable in composition and position
    - TATA box and CCAAT box – RNAP recognition
  - Eukaryotic genes are also regulated by **enhancers**
    - Enhancers may be close to OR far away (sometimes megabases) from the gene
    - May be upstream or downstream or even within introns
    - This makes them hard to predict
- **Important Eukaryotic Genome Annotations**
  - **cDNAs** – reverse transcribed from mRNAs
  - **ESTs** - expressed sequence tags (short segments of cDNAs)
  - **RNA-seq** sequences the pool of cDNA extracted from a sample
    - Very valuable in understanding transcript
      - can be used to identify intron/exon boundaries
      - can be correlated with structure of other genes

Unit 9: Transcriptomics

- **Functional Genomics**
  - Includes Transcriptomics, proteomics, and other omics
  - To understand the function of genomes, instead of individual gene only → **multigene process**
  - **Genome-wide expression analysis**
    - Two major perspectives (& there are more):
      - mRNA transcript abundance - **transcriptomics**
        - **Microarrays and RNA-seq**
      - protein abundance – **proteomics**
    - Unlike the genome (static), transcriptomes and proteomes are dynamic
      - Diverse behavior in different cells/tissues/conditions
      - many more transcripts and proteins than genes
    - A lot more info than just looking at genome
- **Transcriptomics**
  - full set of **mRNA transcripts** expressed in a sample of interest → organism, cell, tissue, etc.
  - Reflects the **biological state** of the sample and **pattern of gene regulation**
    - Stage of development, growth, death; Cell cycle; Diseased vs. healthy; Response to therapy or stress
  - By comparing transcriptomes you can **detect changes in transcription levels** for all genes in a genome
- **Microarray analysis of gene expression**
  - mRNA isolation → prepare cDNA from mRNA → Fluorescent labelling → hybridization to the assay
  - **One-colour technique**
    - A single sample is hybridized to each microarray after it has been labeled with a **single fluorophore**
    - Allows for comparison across many microarrays

- **Two-colour technique**
  - A single sample is hybridized to each microarray after it has been labeled with a **two fluorophores**
  - Produce different colours based on the reaction
- **Determination of Expression Level**
  - Brightness is **proportional** to amount of cDNA bound to spot on chip
  - Colour is due to relative expression levels between control and experimental
  - Raw data are signal intensities
- **Data Processing**
  - Initial data processing:
    - Subtract 'background' signal detected for each spot on the array (reflects noise)
    - Minimize noise variation in data **by log-transforming raw signal intensities**
  - Normalization or standardization
    - Adjust data **to fit a predefined distribution** (e.g., gaussian distribution) that is suitable for statistical analysis
      - There is often a skewed observation of high intensity spots, yet in general we should expect normal distribution
  - Expression data from different samples can be **centered** to have the same median level and **transformed** to have a similar distribution (**between sample normalization by mean centering**)
  - Outlier removal
- **Data Normalization**
  - Normalized expression values for every gene calculated as **ratio of experimental and control expression**
    - Called the **“fold change”**

            - E.g., Cy5 (red) labeled probe from healthy tissue used as a control for expression profile in a Cy3 (green) labeled probe from a tumor
        - But these values are not symmetric around 1
            - To solve this: take **logarithms** of the ratios
            - **+ve values** will reflect experimental up-expression relative to control
            - **-ve values** will reflect experimental down-expression relative to control
            - **This will make the distribution symmetric around 0**
        - Log is commonly used as a **relative** measure
            - **Semi-quantitative data**
                - Easy to distinguish presence/absence
                - Absolute levels beyond current methods
                - Relative levels are difficult but possible
                    - Especially after normalization of data

- **RNA-seq**
    - More modern solution for problems previously addressed by microarray
    - applies **NGS (next generation sequencing)** to sequence all mRNAs (cDNAs) within a sample of interest
    - NGS produces FASTQ Files, and then apply quality control removes poor data
    - Each transcript is sequenced at a different coverage
    - Coverage indicates gene "expression level" → high abundance gets more coverage
    - Complexities and considerations
        - RNA-seq may be difficult without a reference transcriptome or genome to map reads to
        - How to handle multi-mapped reads (reads that align to multiple regions)?
        - How to distinguish splice isoforms? (a gene with multiple splice forms)

        - When comparing between samples, it is often assumed that the **total mRNA abundance is the same** (yet often not true)
            - To solve this → use a **negative control** of known amount, and normalize to the amount of control, rather than overall sample

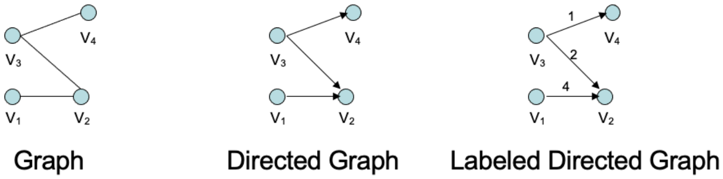
- **Normalization of gene expression levels**
    - Simply counting the number of reads that pile up over a gene will be inaccurate → longer gene will have more reads simply due to its length, even though its expression might be low → so, we have to **normalize and account for the length of genes** → ex: use number of reads per base
    - **RPKM: reads per kilobase million**
        - Account for length (kilobase) and size of data (per million reads mapped)
        - Count <u>total # reads</u> in sample and <u>divide by 1 million</u>
            - Gives you **"per million scaling factor"**
        - Count <u># reads that map to a gene</u> and divide this by the <u>per million scaling factor</u>
            - Gives you **reads per million (RPM)**
        - Divide <u>RPM value</u> by <u>length of gene (in kb)</u>
            - Gives you RPKM
    - **TPM**: number of transcripts per million reads → 10^6 * RPKM/(sum RPKM)
- **Transcriptomic data analysis**
    - Two main quantitative analyses are performed:
        - Detection of **differentially expressed genes (DEGs)** between samples
            - T-test, from <u>repeated</u> experiments
            - If there are a lot of genes → apply **Multiple hypothesis correction** (Bonferroni adjustment and False-discovery rates (FDR))
            - Top DEG candidates will **have logFCs (Fold changes) of high magnitude AND will be statistically significant**
                - Often DEGs are ranked by *p*-value
            - Usually above a horizontal line on **VOLCANO PLOT**

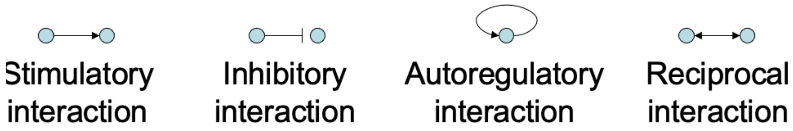
        - Cluster analysis of **co-expressed gene** sets
            - Hierarchical Clustering, PCA (visualization)
                - cluster genes based on their expression profiles across samples and/or cluster samples based on their gene expression profiles
                - ie: what genes have similar expressions? This might suggest that they are functionally linked. Or another reason, what samples have similar expressions?

- **Hierarchical Clustering**
    - Matrix of **genes vs samples** (derived from multiple microarray or RNA-seq experiments)
        - Samples may be different tissues, conditions, time points, etc.
        - Values can be FPKM, TPM, relative expression levels (e.g., fold changes)
    - Matrix can be clustered by rows or columns
    - Values Colored as heat map (usually: red = up regulation; green = down regulation)
    - **Clustering of Experimental Data**
        - A measure of similarity between expression pattern is needed
        - Can compute the **correlation coefficient** (-1.0 to 1.0) between any two expression profiles
        - Use this as a distance/similarity measure between genes, with 1.0 being an exact match and -1.0 being negatively correlated
        - Apply UPGMA to cluster data, and generate a **similarity tree** for genes

Unit 10: Network and System Biology

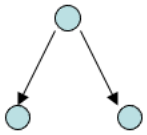
- **System Biology**
    - Extremely difficult to define → Meant many things over the last 50 years
    - Institute for Systems Biology:
        - Systems biology is the study of an organism, viewed as an **integrated and interacting network** of genes, proteins and biochemical reactions which give rise to life.
    - Instead of analyzing individual components or aspects of the organism, such as sugar metabolism or a cell nucleus, systems biologists focus **on all the components and the interactions among them**, all as part of one system.
- **Network**
    - A **biological system** is its **components** and their **interactions**
    - This information can be represented as a **network**
    - By examining a biological system as a network of interacting components, we can <u>view the big picture</u>
    - 2 elements in a network:
        - **Node:** Gene, Protein, Neuron, Species
        - **Edge**: Physical interaction; Regulatory interaction; Functional interaction; Electrical signaling
    - Biological networks include Protein Interaction Networks, Gene Regulatory Networks, Metabolic Networks (ex: KEGG Database), Cell, Organisms, Ecosystems
- **Important Terminologies**
    - Each **edge** is specified by a pair of **vertices (nodes)**
    - In a **directed graph,** the edges are **ordered pairs** of vertices
    - In **a labeled graph**, there are <u>values</u> associated with <u>each edge</u>
    - An **undirected unlabeled graph** specifies <u>connectivity without orientation</u>



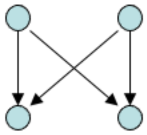


Stimulatory interaction, inhibitory interaction, autoregulatory interaction, reciprocal interaction

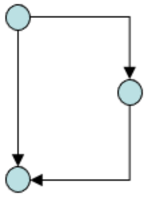
- **Common Network Motifs in Biological Networks**
    - **Fork**
        - Single-input motif, one incoming signal, multiple outputs (can amplify signal / **cascade**)
        - Effective for activations of large sets of genes from a single impulse



    - **Scatter**
        - Multiple-input motif
        - Can function as an OR operation
        - Both downstream impulses are activated by either upstream element



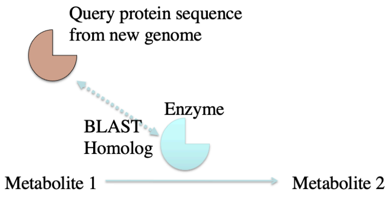
    - **One-two punch**
        - Feed-forward loop
        - If both paths are needed, it operates as an AND
        - Can filter out 'noisy' stimuli



- **Structure vs. Dynamics**
    - Modeling of a biological network requires knowledge of its:
    - **Structure – this is static**

        - Can be retrieved from **databases**
            - E.g., Known structure of human metabolism
        - Can be **inferred**
            - E.g., construct gene regulatory network by connecting **coexpressed genes** (Pearson correlation > K – the threshold)
            - Connect two proteins if there is significant evidence of a **physical interaction**
            - Extracted from **published literature** (text-mining)
    - **Dynamics**
        - How does the network **change over time**, in response to various cell types, pressures, perturbations, etc.
        - Requires **experimental data**
        - Enzyme kinetics, binding coefficients, concentrations, etc.
- **The String Database**
    - Infer network structure
    - Combines eight types of evidence to support and interaction between two proteins:
        - **Gene Neighborhood** → Interacting genes tend to be clustered in the genomes
        - **Gene fusion** → Fusions indicate that those genes are interacting in some way
        - **Co-occurrence** → Genes that appear together across many species (it might suggest that one requires the other, or some pathways require both)
        - **Co-expression** → Genes expressing together
        - **Experiments** → Usually high throughput – many proteins against many other proteins → most credible source
        - **Textmining** → use programs that detects word associations across big databases

    - **Database** → existing information about the structure
    - **Homology** → similar proteins might interact with each other (usually)
  - Connects proteins based on **total score** (specified threshold)
  - Additional information:
    - Can also add **protein functions** to the network (**Gene ontology (GO) functions**) → hubs
      - Map the functional annotation by, for example, colouring
    - Add **linkages/hubs** AND **subcellular localization** data
    - Results in a more realistic computational model of the cell
- **Hubs**
  - Proteins that participate in the **same functional module** (e.g., complex) are organized into hubs
  - many proteins all interacting with <u>each other</u> or with <u>a central protein</u>
  - Two types:
    - **Party Hub**
      - Members of hubs interact with <u>each other</u> most of the time
    - **Date Hub**
      - Interact with <u>different partners</u> at different times and locations
  - Analyze networks to identify nodes → important proteins with many connections → **hub proteins**
- **Inferring Pathways from Genomes**
  - By using **Homology**
    - Define known pathways with **reference enzymes** for each reaction
    - Use **homology** (e.g., BLAST) to detect presence/absence of **homologous protein** for species of interest


Query protein sequence
from new genome
Enzyme
BLAST
Homolog
Metabolite 1
Metabolite 2