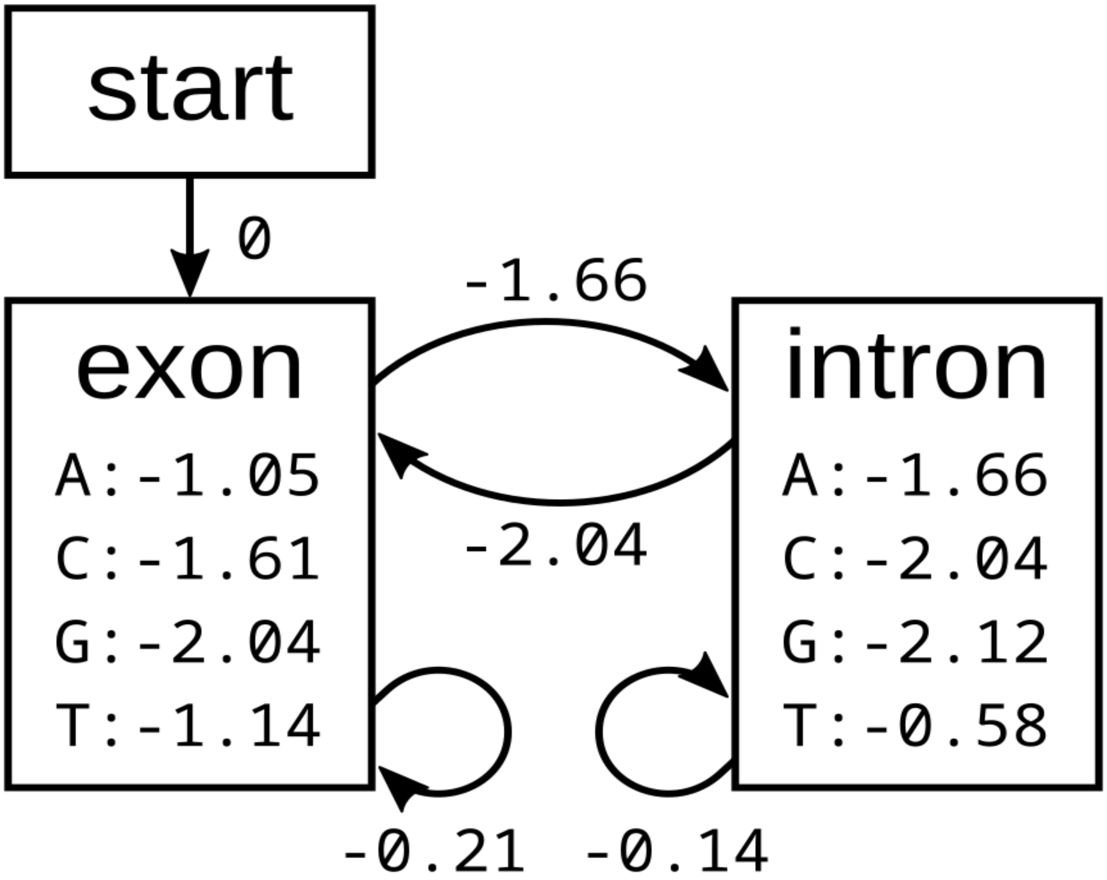
Class Test 02 (Question 2)

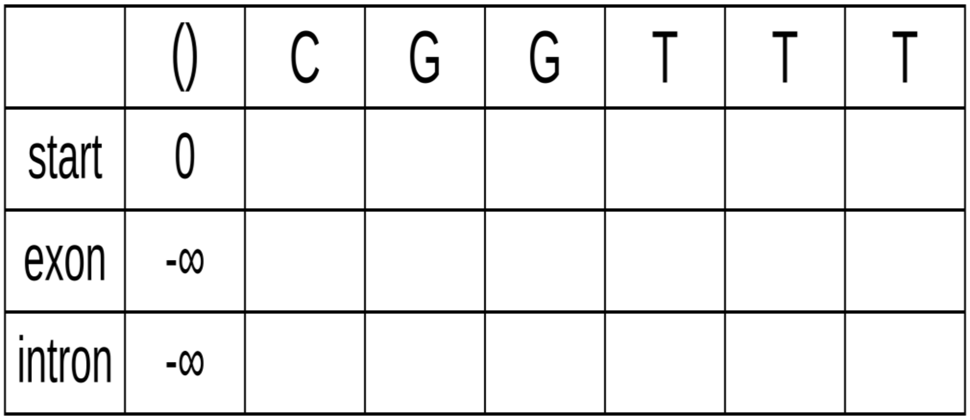
1)

The probabilities of the model have the corresponding log-probabilities, to two decimal places:



Let's apply this simple model to the toy sequence CGGTTT.

Draw up a table and fill in the probabilities of the states when the sequence is empty: 0 log-probability (100% probability) for being in the start state at the start of the sequence, and negative infinity (0% probability) for not being in the start state at the start of the sequence:



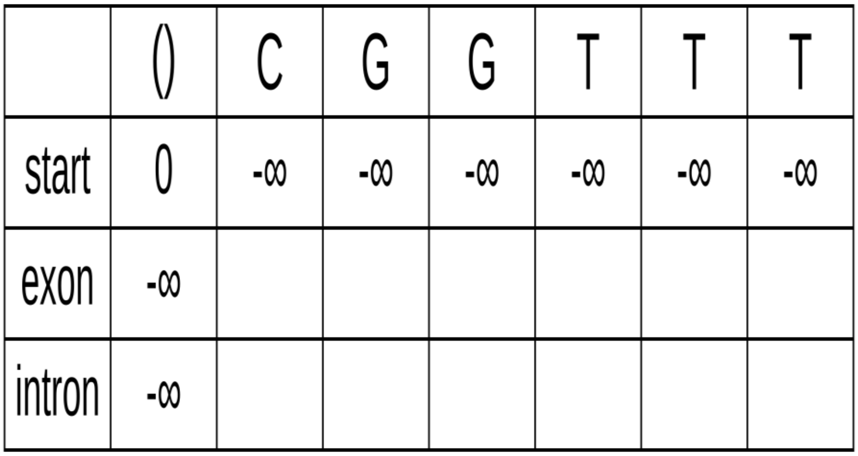
| | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | | | | | | |
| exon | -∞ | | | | | | |
| intron | -∞ | | | | | | |

We will refer to every element of the matrix as $v_{k,i}$ where $k$ is the hidden state, and $i$ is the position within the sequence. $v_{k,i}$ is the maximum log joint probability of the sequence and any path up to $i$ where the hidden state at $i$ is $k$:

$v_{k,i} = \max_{\text{path1..}i\text{-1}}(\log P(\text{seq}_{1..i}, \text{path}_{1..i\text{-}1}, \text{path}_i = k))$.

This log joint probability is equal to the maximum value of $v_{k',i\text{-}1}$ where $k'$ is the hidden state at the previous position, plus the transition log-probability $t_{k',k}$ of transitioning from the state $k'$ to $k$, plus the emission log-probability $e_{k,i}$ of the nucleotide (or amino acid for proteins) at $i$ given $k$. We find this value by calculating this sum for every previous hidden state $k'$ and choosing the maximum.

The transition log probability from any state to the start state is -∞, so for any value of $i$ from 1 onwards, $v_{\text{start},i} = -\infty$. Go ahead and fill those in to save time:



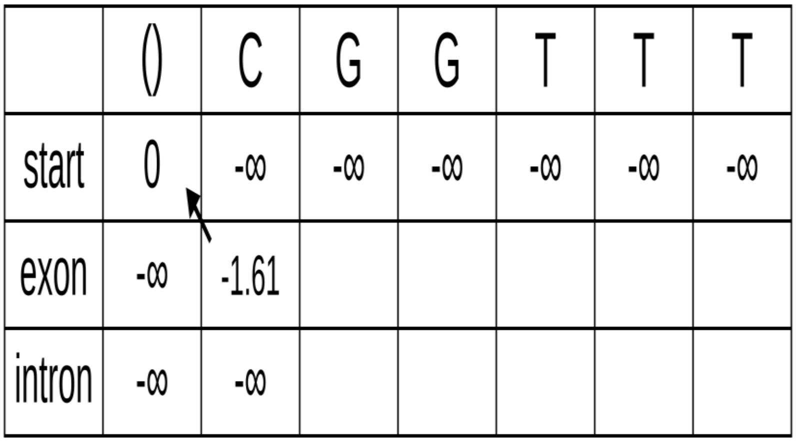
| | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ |
| exon | -∞ | | | | | | |
| intron | -∞ | | | | | | |

For the next element $\mathrm{v}_{\mathrm{exon},1}$ we only have to consider the transition from the start state to the exon state, because that is the only transition permitted by the model. Even if we do the calculations for the other transitions, the results of those calculations will be negative infinities because the Viterbi probability of non-start states in the first column are negative infinities. The log-probability at $\mathrm{v}_{\mathrm{exon},1}$ is therefore:

- $v_{\mathrm{exon},1} = v_{\mathrm{start},0} + t_{\mathrm{start,exon}} + e_{\mathrm{exon},1} = 0 + 0 + -1.61 = -1.61$

The log-probability of $v_{\mathrm{intron},1}$ is negative infinity because the model does not permit the state at the first sequence position to be an intron. This can be effected computationally by setting the $t_{\mathrm{start,intron}}$ log-probability to negative infinity. Then regardless of the Viterbi and emission log-probabilities, the sum of $v$, $t$ and $e$ will be negative infinity.

Fill in both values for the first position of the sequence (or second column of the matrix), and add a pointer from the exon state to the start state:



|  | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ |
| exon | -∞ | -1.61 |  |  |  |  |  |
| intron | -∞ | -∞ |  |  |  |  |  |

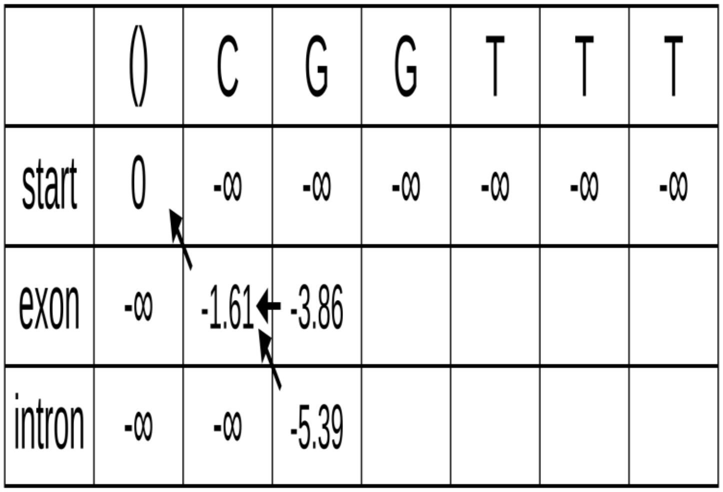
Once we get to $v_{\mathrm{exon},2}$, we only have to consider the exon to exon transition since the log-probabilities for the other states at the previous position are negative infinities. So this log-probability will be:

- $v_{\mathrm{exon},2} = v_{\mathrm{exon},1} + t_{\mathrm{exon,exon}} + e_{\mathrm{exon},2} = -1.61 + -0.21 + -2.04 = -3.86$

And for the same reason to calculate $v_{\mathrm{intron},2}$ we only have to consider the exon to intron transition, and this log-probability will be:

- $v_{\mathrm{intron},2} = v_{\mathrm{exon},1} + t_{\mathrm{exon,intron}} + e_{\mathrm{intron},2} = -1.61 + -1.66 + -2.12 = -5.39$

So fill on those values, and add pointers to the only permitted previous state, which is the exon state:



| | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ |
| exon | -∞ | -1.61 | -3.86 | | | | |
| intron | -∞ | -∞ | -5.39 | | | | |

For the next position, we have to consider all transitions between intron or exon to intron or exon since both of those states have finite log-probabilities at the previous position. The log-probability of $v_{\text{exon},3}$ will be the maximum of:

- $v_{\text{exon},2} + t_{\text{exon,exon}} + e_{\text{exon},3}$ = -3.86 + -0.21 + -2.04 = -6.11
- $v_{\text{intron},2} + t_{\text{intron,exon}} + e_{\text{exon},3}$ = -5.39 + -2.04 + -2.04 = -9.47

The previous hidden state that maximizes the Viterbi log-probability for the exon state at the third sequence position is therefore the exon state, and the maximum log-probability is -6.11. The log-probability of $v_{\text{intron},3}$ will be the maximum of:

- $v_{\text{exon},2} + t_{\text{exon,intron}} + e_{\text{intron},3}$ = -3.86 + -1.66 + -2.12 = -7.64
- $v_{\text{intron},2} + t_{\text{intron,intron}} + e_{\text{intron},3}$ = -5.39 + -0.14 + -2.12 = -7.65

The previous hidden state that maximizes the Viterbi log-probability for the intron state at the third sequence position is therefore also the exon state, and the maximum log-probability is -7.64.

Fill in the maximum log-probabilities for each hidden state *k*, and also draw pointers to the previous hidden states corresponding to those maximum log-probabilities:

|  | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ |
| exon | -∞ | ↖ -1.61 | ← -3.86 | ← -6.11 |  |  |  |
| intron | -∞ | -∞ | ↖ -5.39 | ↖ -7.64 |  |  |  |

The rest of the matrix is filled in the same way as for the third position:

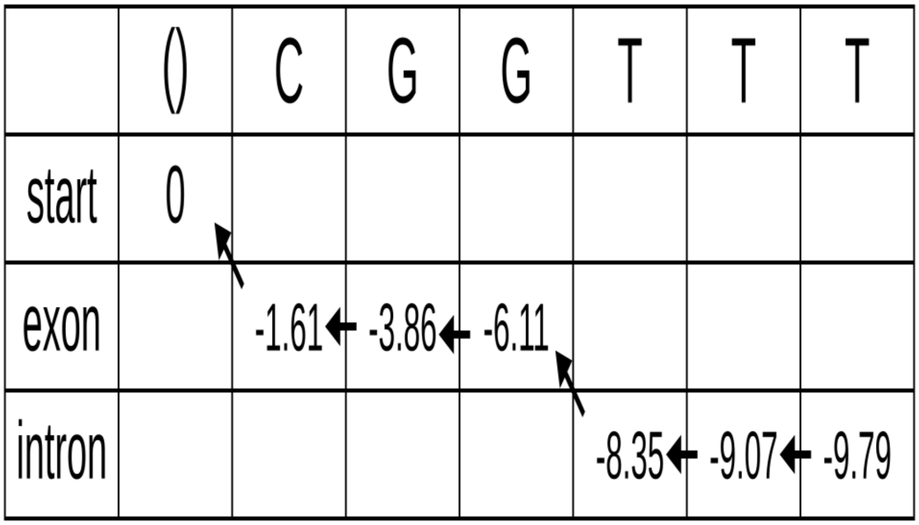
|  | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | -∞ | -∞ | -∞ | -∞ | -∞ | -∞ |
| exon | -∞ | ↖ -1.61 | ← -3.86 | ← -6.11 | ← -7.46 | ← -8.81 | ← -10.16 |
| intron | -∞ | -∞ | ↖ -5.39 | ↖ -7.64 | ↖ -8.35 | ← -9.07 | ← -9.79 |

The maximum log joint probability of the sequence and path is the maximum out of $\mathrm{v}_{k,L}$, where $L$ is the length of the sequence. In other words, if we calculate the log joint probability

$\mathrm{v}_{k,L} = \max_{\mathrm{path}1..L\text{-}1}(\log\mathrm{P}(\mathrm{seq}_{0..L}, \mathrm{path}_{0..L\text{-}1}, \mathrm{path}_L = k))$.

for every value of $k$, we can identify the maximum log joint probability unconditional on the value of $k$ at $L$. The path is then reconstructed by following the pointers backwards from the maximum log joint probability. In our toy example, the maximum log joint probability is -9.79 and the path is:

**2)**



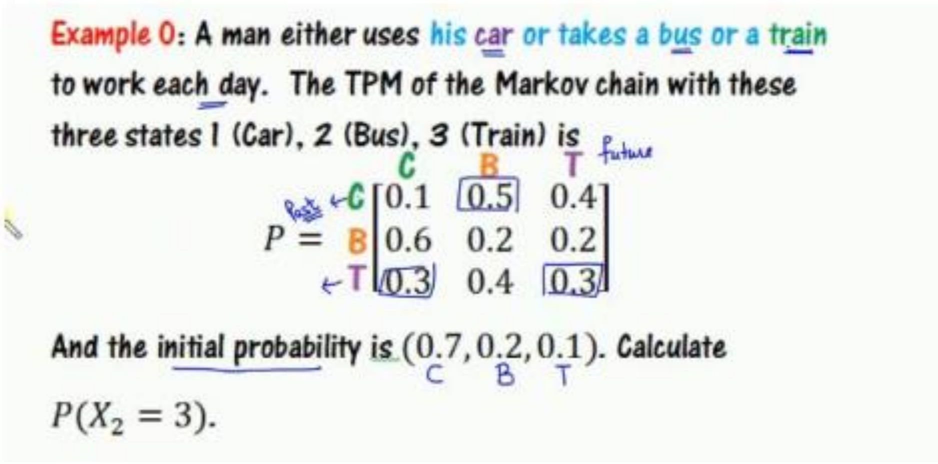
| | () | C | G | G | T | T | T |
|---|---|---|---|---|---|---|---|
| start | 0 | | | | | | |
| exon | | ↖ -1.61 | ← -3.86 | ← -6.11 | | | |
| intron | | | | | ↖ -8.35 | ← -9.07 | ← -9.79 |

Or, ignoring the start state, exon-exon-exon-intron-intron-intron.

The basic Viterbi algorithm has a number of important properties:

- Its space and time complexity is O(*Ln*) and O($Ln^2$) respectively, where *n* is the number of states and *L* is the length of the sequence
- It returns a point estimate rather than a probability distribution
- Like Needleman–Wunsch or Smith–Waterman it is exact, so it is guaranteed to find the optimal[1] solution, unlike heuristic algorithms, and unlike an MCMC chain run for a finite number of steps[2]
- The probability is the (log) joint probability of the *entire* sequence (e.g. nucleotides or amino acids) **and** the *entire* path of unobserved states. It is *not* identifying the most probable hidden state at each position, because it is not marginalizing over the hidden states at other positions.

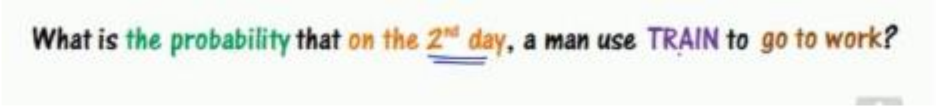
**2)**



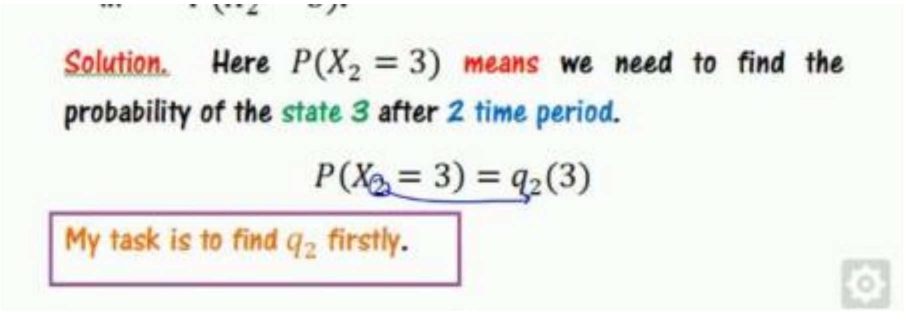
Example 0: A man either uses his car or takes a bus or a train to work each day. The TPM of the Markov chain with these three states 1 (Car), 2 (Bus), 3 (Train) is

$$P = \begin{array}{c|ccc} & C & B & T \text{ future} \\ \text{Past} \leftarrow C & 0.1 & \boxed{0.5} & 0.4 \\ B & 0.6 & 0.2 & 0.2 \\ \leftarrow T & \boxed{0.3} & 0.4 & \boxed{0.3} \end{array}$$

And the initial probability is $(\underset{C}{0.7}, \underset{B}{0.2}, \underset{T}{0.1})$. Calculate $P(X_2 = 3)$.



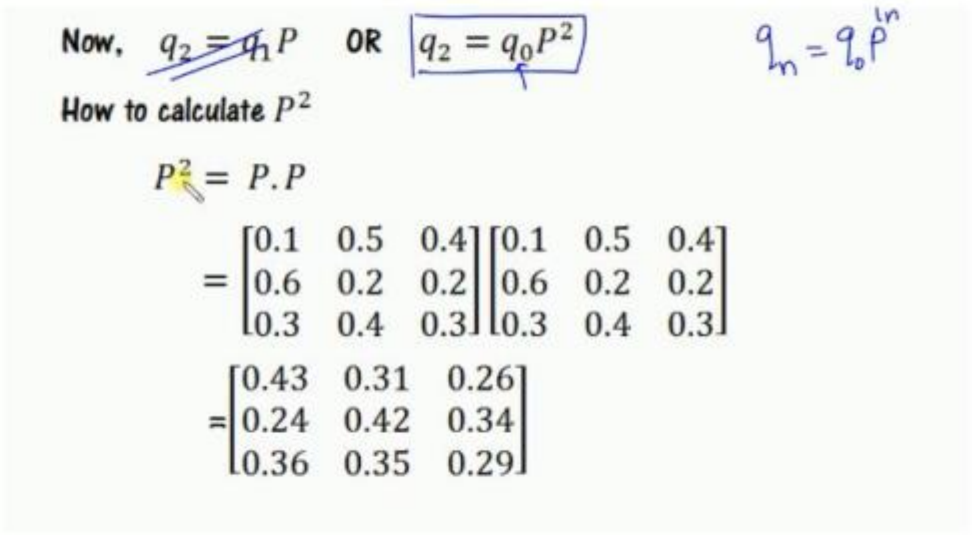
What is the probability that on the 2nd day, a man use TRAIN to go to work?



Solution. Here $P(X_2 = 3)$ means we need to find the probability of the state 3 after 2 time period.

$$P(X_2 = 3) = q_2(3)$$

My task is to find $q_2$ firstly.



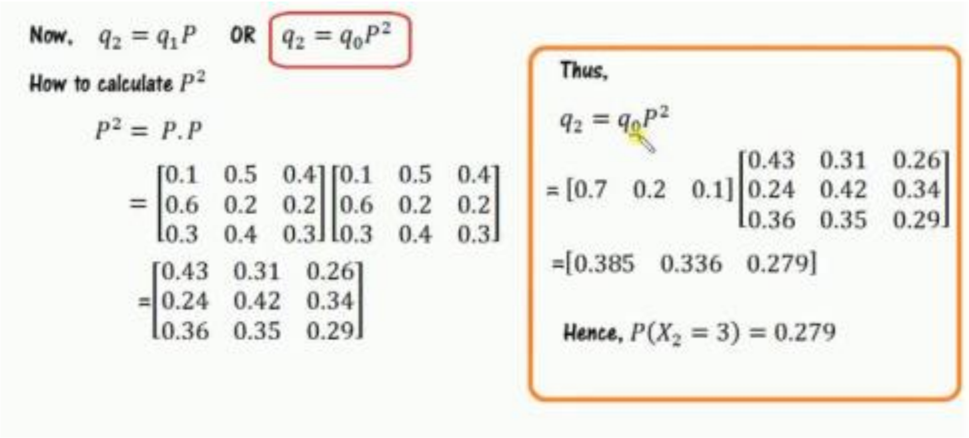
Now, $\cancel{q_2 = q_1} P$ OR $\boxed{q_2 = q_0 P^2}$ $\qquad q_n = q_0 P^{n}$

How to calculate $P^2$

$$P^2 = P.P$$

$$= \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$$

$$= \begin{bmatrix} 0.43 & 0.31 & 0.26 \\ 0.24 & 0.42 & 0.34 \\ 0.36 & 0.35 & 0.29 \end{bmatrix}$$



Now, $q_2 = q_1 P$ OR $\boxed{q_2 = q_0 P^2}$

How to calculate $P^2$

$$P^2 = P.P$$

$$= \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \begin{bmatrix} 0.1 & 0.5 & 0.4 \\ 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \end{bmatrix}$$

$$= \begin{bmatrix} 0.43 & 0.31 & 0.26 \\ 0.24 & 0.42 & 0.34 \\ 0.36 & 0.35 & 0.29 \end{bmatrix}$$

Thus,

$$q_2 = q_0 P^2$$

$$= [0.7 \quad 0.2 \quad 0.1] \begin{bmatrix} 0.43 & 0.31 & 0.26 \\ 0.24 & 0.42 & 0.34 \\ 0.36 & 0.35 & 0.29 \end{bmatrix}$$

$$=[0.385 \quad 0.336 \quad 0.279]$$

Hence, $P(X_2 = 3) = 0.279$

## Class Test 03(Question 2,3)

Q2a, What are the significances of studying protein-protein interactions in bioinformatics?

Studying protein-protein interactions (PPIs) in bioinformatics is of significant importance because it provides valuable insights into the complex molecular mechanisms that underlie various biological processes. The study of PPIs is crucial for several reasons:

i. **Understanding Biological Function**: Proteins rarely work in isolation; they often interact with other proteins to perform specific functions. By studying PPIs, we can unravel the underlying biological pathways and processes, including signal transduction, metabolic pathways, and cellular responses to external stimuli.

ii. **Drug Discovery and Development**: Many diseases are caused by disruptions in PPIs. Understanding these interactions can lead to the identification of potential drug targets and the development of therapeutic interventions. Targeted drug design is often based on disrupting or enhancing specific PPIs.

iii. **Network Biology**: PPIs form intricate networks, and these networks can reveal the organization and functioning of biological systems. Network analysis can provide insights into key regulatory proteins, hubs, and bottlenecks in these networks.

iv. **Functional Annotation**: By identifying the interacting partners of a protein, we can infer its potential function. This is particularly useful for studying proteins with unknown functions. Knowing the context in which a protein operates helps annotate its function.

v. **Disease Mechanisms**: Many diseases, including cancer and neurodegenerative disorders, are associated with disrupted PPIs. Understanding these interactions can shed light on the underlying disease mechanisms and lead to the development of diagnostic and therapeutic approaches.

vi. **Evolutionary Biology**: PPIs can change over evolutionary time, and comparing PPI networks across species can provide insights into the evolution of biological processes and the emergence of new functions.

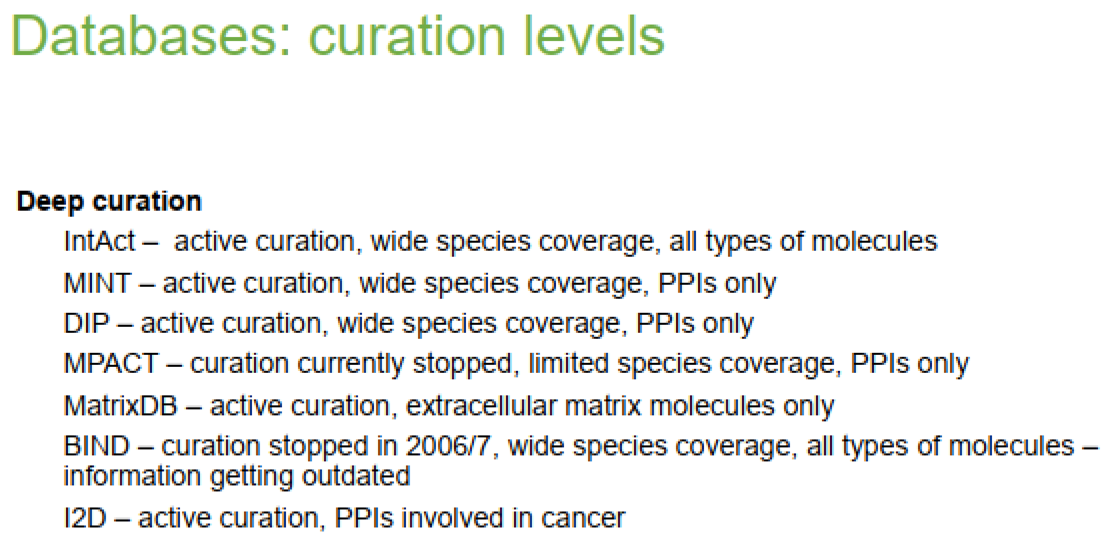
vii. **Protein Engineering and Biotechnology**: Knowledge of PPIs is essential in designing proteins for various biotechnological applications, such as enzyme engineering, antibody development, and the production of biopharmaceuticals.

viii. **Structural Biology**: Determining the three-dimensional structures of protein complexes involved in PPIs is important for understanding the atomic-level details of these interactions. This information can inform drug design and mechanistic studies.

ix. **Biomarker Discovery**: PPIs can be used to identify potential biomarkers for disease diagnosis and prognosis. Changes in PPIs can be indicative of disease or other physiological states.

x. **Systems Biology**: The integration of PPI data with other biological data types (e.g., gene expression, metabolomics) is critical for a systems-level understanding of living organisms. This holistic approach can lead to the discovery of emergent properties and regulatory mechanisms.

b. Write the name and function of four databases those are exist in deep curation level. How can you collect and preprocessing the biological datasets from NCBI?



Databases: curation levels

**Deep curation**

- IntAct – active curation, wide species coverage, all types of molecules
- MINT – active curation, wide species coverage, PPIs only
- DIP – active curation, wide species coverage, PPIs only
- MPACT – curation currently stopped, limited species coverage, PPIs only
- MatrixDB – active curation, extracellular matrix molecules only
- BIND – curation stopped in 2006/7, wide species coverage, all types of molecules – information getting outdated
- I2D – active curation, PPIs involved in cancer

Collecting and preprocessing biological datasets from the National Center for Biotechnology Information (NCBI) can involve several steps, depending on the type of data a're interested in. NCBI provides a vast repository of biological data, including DNA and protein sequences, gene expression data, clinical data, and more. Here is a general guide on how to collect and preprocess biological datasets from NCBI:

i. **Identify the specific Data of Interest**:

- Determine the specific type of biological data are need, such as DNA sequences, protein structures, gene expression profiles, or clinical data.

ii. **Access NCBI Databases**:

- Visit the NCBI website (https://www.ncbi.nlm.nih.gov) and navigate to the appropriate database or resource. Some commonly used databases include:

- **GenBank**: For DNA and RNA sequences.
- **PubMed**: For scientific literature, including articles related to your research.
- **GEO (Gene Expression Omnibus)**: For gene expression and microarray data.
- **Protein Data Bank (PDB)**: For protein structures.
- **ClinVar**: For clinical and genetic variation data.

iii. **Search and Retrieve Data**:

- Use the search features provided by NCBI to find datasets relevant to your research. You can use keywords, accession numbers, gene names, or other search criteria.
- Download or retrieve the data in a suitable format (e.g., FASTA format for sequences, text or CSV files for tabular data).

iv. **Data Preprocessing**:

- Depending on the type of data you've obtained, you may need to preprocess it to make it suitable for analysis. Preprocessing steps vary based on the data type:
  - **Sequences** (DNA, RNA, protein): we might want to remove contaminants, trim sequences, and format them for downstream analysis.
  - **Gene Expression Data**: Clean, normalize, and transform data if necessary.
  - **Structural Data (Proteins)**: Perform structure validation and cleaning.
  - **Clinical Data**: Handle missing values, categorize variables, and standardize data formats.
  - **Quality Control**:
  - **Data Integration** (if applicable):
  - **Statistical Analysis**:
    - **Data Visualization**:

- **Documentation**:
- **Ethical and Legal Considerations**:
- **Data Sharing and Publication**:

c. Provide an example of a disease or medical condition that results from disrupted Protein-Protein Interactions (PPIs) and create a complex protein structure by following the PPIs.

Q3a. Describe the procedure how to represent a standard PPIs network by using STRING?

To represent a standard Protein-Protein Interaction (PPI) network using STRING (Search Tool for the Retrieval of Interacting Genes/Proteins), follow these steps:

1. **Access STRING Database**:
   - Go to the STRING database website (https://string-db.org/).
2. **Enter Protein(s) of Interest**:
   - In the search bar, enter the name, gene symbol, or UniProt ID of the protein(s) you are interested in.  can search for multiple proteins at once.
3. **Select Organism**:
   - Specify the organism of interest by choosing from the dropdown menu. This helps STRING provide relevant interactions based on the chosen species.
4. **Set Confidence Score Threshold (Optional)**:
   - can adjust the confidence score threshold to filter interactions. Higher thresholds yield more reliable interactions but may reduce the number of interactions displayed.
5. **Start the Search**:

- Click the "Search" or "Enter" button to initiate the search. STRING will retrieve the interactions and display them in various formats.

6. **View the Interaction Network**:

   - STRING will generate an interaction network graph that represents the protein-protein interactions. The nodes in the graph represent proteins, and the edges represent interactions.

7. **Interpret the Network**:

   - Examine the network for important features, such as densely connected nodes (hubs) or clusters of proteins. These may indicate key biological pathways or protein complexes.

8. **Customize Network Visualization**:

   - can customize the network visualization using options provided by STRING. This may include changing the layout, adjusting the confidence score, or selecting a specific node to focus on.

9. **Explore Additional Information**:

   - STRING offers additional information, such as functional enrichments, pathway analysis, and GO (Gene Ontology) term annotations. Use these features to gain insights into the biological context of the interactions.

10. **Download Data** (Optional):

    - If needed, can download the interaction network, including protein identifiers, interaction scores, and annotations, for further analysis or visualization in other tools.

11. **Cite and Share**:

    - If use STRING data in research or publications, make sure to properly cite the source. can also share your findings using STRING's sharing features.

By following these steps, can effectively use the STRING database to represent a standard Protein-Protein Interaction network for your research or analysis.

b. Explain the method how you can map a network of physical contacts and co-expression system in Cytoscape software.

c. Choose a specific drug and its target protein, and explain how the drug interacts with the protein to achieve its therapeutic effect.