**SLIDE 3: INTRO**

* *Central Dogma* is a mechanism of molecular biology that transcripts genetic message from DNA to RNA and translate those from RNA to protein. It explains the transformation of genetic information called DNA replication, RNA encoding by transcription, and encoding for protein through translation.
* *Transcription* is the first step in gene expression. It involves copying a gene’s DNA sequence to make an RNA molecule. It is performed by enzymes called RNA polymerase, which link nucleotides to form an RNA strand.
* If a protein is encoded by a transcribed gene, it results in messenger RNA (mRNA). This mRNA helps to create protein in translation. During the *translation* process, mRNA is decoded and translated to create a protein or polypeptide sequence.

**SLIDE 4: CENTRAL DOGMA**

As described in slide 3

**SLIDE 5: GRN**

* Central Dogma cannot explain the nature and strength of such gene regulatory interactions.
* A wet lab (a type of lab where it is necessary to handle various types of chemicals and potential “wet” hazards) cannot determine the correct regulation nature and strength between the genes.
* So computational tools are required to find out such nature of regulatory interactions.
* A *Gene Regulatory Network (GRN)* is used to represent the interaction between the genes. It is a bidirectional graphical structure consisting of nodes and edges. Network nodes represent the genes and their regulators and the network edges represent the regulatory relationships between the nodes. GRNs are both bipartite and directional. They are bipartite because there are two types of nodes: genes and regulators, although of course some genes are themselves regulators of other genes or proteins. Gene regulatory networks are directional because regulators control genes and usually not the other way around.

**SLIDE 6: GRN**

As described in slide 5

**SLIDE 7: GRN**

The figure shows original topology of 8-gene SOS DNA repair network of *E.coli*. Here black arrow represents activation and orange arrow represents repression. In 2002, this network was studied and their temporal expression levels were analyzed. In this work the authors prepared 4 experimental setups and in each such experiment, they noted the expression levels of all the genes for 50 instances at a temporal resolution of 6 minutes.

**SLIDE 8: TIME-SERIES DATA**

* Time-series data is a sequence of data points indexed in time order. This time-series data can represent a gene expression data set using a 2D matrix.
* Time-series data can be classified into two types: (a) metrics where measurements are gathered at regular time intervals, (b) events where measurements are gathered at irregular time intervals.
* Some examples of time-series data are weather records, economic indicators, patient health evolution metrics, etc.

**SLIDE 9: RNN**

* A neural network is used to recognize underlying relationships in a dataset through a process that mimics the way human brain operates. It basically refers to the system of neurons, either organic or artificial in nature. It can be either a biological neural network made up of real biological neurons or an artificial neural network used for solving artificial intelligence problems. Recurrent neural network (RNN) is a special type of artificial neural network.
* RNNs are popular because of their ability to learn from data and robustness to noisy data.
* Mathematically it is an exponential-law based tool used in reverse engineering. (Here reverse engineering refers to the process of reconstructing GRNs from gene expression data)
* The recurrent structure of RNNs effectively reflects the existence of feedback, which is essential in case of gene regulatory system. It is a genetic network of feed-forward type, where all the genes and their product participating in one regulatory event are members of the network. Basic feed forward networks can remember things they learnt during training however, in addition to that RNNs can also remember things learnt from prior inputs while generating outputs.
* The total number of parameters that need to be estimated for each gene in the RNN formalism is (*N* + 2). Thus, the model requires *N* (*N* + 2) parameters for an *N*-gene network.

**SLIDE 10: RNN**

Here the figure shown gives a representation of a GRN using a RNN model. The regulatory interaction between the genes has been shown at different time points. Here all possible connections between the genes are shown however, in reality such networks are sparse in nature (i.e., each and every gene cannot have a connection with every other gene present in the network in reality). Here each tier of the neural network defines the genetic expression level of the genes at a specified time t. The level of expression of any particular gene at a time depends upon the genetic expression level of all genes () at the preceding time t and the weights of their corresponding edges () with that particular gene.

**SLIDE 11: RNN**

* : Gap between two consecutive time-points
* : Time constant of th gene
* : Bias term of th gene
* : Total no. of genes in the system
* : Gene expression level for the gene at time point
* : Gene expression level of gene at next time point
* : Sigmoid function where is the effect of gene on the gene
* : Gene expression level for the gene at time point

**SLIDE 12: S-SYSTEM**

* S-System is a set of non-linear differential equations of the first order commonly used to reconstruct GRNs from time-series gene expression datasets. This S-System can be said as the canonical form of many different equations.
* S-System, a power-law based computational tool, used for modelling and analyzing biological system, provides a good comparison in between the biological relevance and mathematical flexibility.
* It is capable to explain two essential characteristics biological system i.e., saturable and synergistic (interaction between two or more parts produce effect greater than the sum of their separate effects) behaviour of GRN.
* The total number of parameters that need to be estimated for each gene in the S-System formalism is (2*N* + 2). Thus, the model requires 2*N* (*N* + 1) parameters for the case of an *N*-gene network.

**SLIDE 13: S-SYSTEM**

* : rate constants for the production and degradation terms, respectively
* : kinetic orders of the system
* : total no. of genes in the network

**SLIDE 14: S-SYSTEM**

* S-System modelling poses a few problems. It is computationally expensive to train the model due to the large number of parameters required compared to RNN, i.e. 2*N* (*N* + 1) compared to *N* (*N* + 2). Also, a problem arises when both the predicted and are of the same sign, which suggests dual regulations. This is unrealistic, as a gene cannot activate as well as inhibit another gene at the same time. To avoid such issues, we have employed the Half-System (HS) formalism which has been explained in the later slides.

**SLIDE 15: HALF-SYSTEM**

* Half-System (HS) is an improved version of the S-System where the number of parameters is exactly half of that of the S-System.
* The computational problem of the S-system is reduced in this version.
* The total number of parameters required for training is (*N* + 1). Thus, the model requires *N* (*N* + 1) parameters for a *N* -gene network.

**SLIDE 16: HALF-SYSTEM**

* : single rate constant
* : only kinetic order of the system
* *N* : number of genes in the network
* The main advantage of this model is that it does not support activation and inhibition simultaneously.

**SLIDE 17: HALF-SYSTEM**

* In order to improve the stability of half-system further we have introduced a negative feed-back term.
* : penalty term

**SLIDE 18: MEAN SQUARED ERROR**

* : Total no. of genes
* : Total no. of time points

**SLIDE 19: PSO**

In computer science, artificial intelligence, and mathematical optimization, a **heuristic** is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution.

A **metaheuristic** is defined as an iterative approach which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting search space.

* Particle Swarm Optimization (PSO) is an evolutionary swarm based optimization algorithm.
* It is based on the flocking behavior of birds.
* It improves the candidate solutions in an iterative process.
* In our work we use PSO to optimize the hybrid model parameters in an iterative manner.
* PSO consists of a swarm of particles, each of which represents a candidate solution.
* PSO is implemented with a memory mechanism, which can retain the information of previous best solutions that may get lost during the population evolution. The basic idea of PSO is that each particle randomly searches through the problem space by updating itself with its own memory and the social information gathered from other particles.
* Each particle with a position move in the multidimensional problem space with a corresponding velocity .
* The basic components of PSO are represented in terms of two best locations during the evolution process: one is the particle’s own previous best position, represented as vector , according to the calculated fitness value, and the other is the best position in the entire swarm, represented as .

**SLIDE 20: PSO**

* : Inertia weight
* and : Uniform random numbers in the range of
* and : Cognitive coefficient and social coefficient, respectively
* :th particle’s previous best position vector
* : Best position vector in the entire swarm in the th generation.
* : Updated velocity vector of particle in the current generation
* : Particle velocity vector at time point
* : Updated position vector of the particle in the th iteration.
* : Position vector of the th particle in the th generation

**SLIDE 21: PROPOSED WORK**

* TPR : True positive rate or sensitivity or recall (fraction of the existing edges in the original network)
* TNR : True negative rate or specificity (number of non-existing edges in the original network as well as the predicted network)
* PPV : Positive predictive value or precision (number of correct edges in the original network which are interacting)
* ACC : Accuracy (total number of all interaction in the original network which is truly predicted)
* F-score : Harmonic mean of TPR and PPV