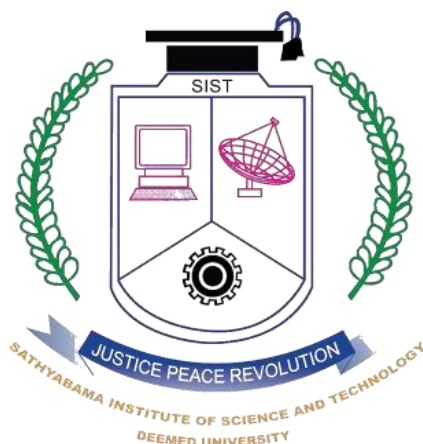


INTEGRATED ANALYSIS: GENE EXPRESSION, SURVIVAL, BIOMARKERS AND DRUG TARGETS

Submitted in partial fulfillment of the required for the awarded of
**MASTER OF SCIENCE DEGREE IN BIOINFORMATICS AND
DATA SCIENCE BY
AMRUTHA M (43739035)**



DEPARTMENT OF BIOINFORMATICS AND DATA SCIENCE
SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC JEPPIAAR
NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC I 12B Status by UGC

Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF BIOINFORMATICS AND DATA SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of AMRUTHA M (43739035) who carried out the project entitled “**synergizing system**

Biology and machine learning: Advanced biomedical research under my / our supervision from January 2024 to April 2024

INTERNAL GUIDE

EXTERNAL GUIDE (if Applicable)

Head of the Department

Submitted for Viva-voce Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I, AMRUTHA M hereby declare that the project entitled “**synergizing System Biology and Machine learning: Advanced biomedical research**” done by me under the guidance of **MISS. MANISHA** is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Bioinformatics and Data Science.

DATE:

PLACE: CHENNAI

AMRUTHA M
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **DR. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide

MISS MANISHA Department of Bioinformatics and Data Science for his valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Bioinformatics and Data Science** who were helpful in many ways for the completion of the project.

AMRUTHA M

S.NO	TITLE	PAGE NO
1.	REVIEW ARTICLE	
2.	ABSTRACT TRANSLATIONAL BIOINFORMATICS GENE EXPRESSION ANALYSIS 2.1 INTRODUCTION 2.2 AIM 2.3 MATERIALS AND METHODS 2.4PROCEDURE 2.5RESULTS	
3.	CUMULATIVE SURVIVAL PROBABILITY C ALCULATING USING KM PLOTTER 3.1 INTRODUCTION 3.2 AIM 3.3 MATERIALS AND METHODS 3.4 PROCEDURE 3.5 RESULTS	
4.	BIOMARKERS:IDENTIFYING BIOLOGICAL MARKERS FOR DISEASE DISGNOSIS AND THERAPEUTIC MONITORING 4.1 INTRODUCTION 4.2 AIM 4.3 MATERIALS AND METHODS 4.4 PROCEDURE 4.5 RESULTS	
5.	DRUG-TARGET INTERACTIONS PREDICTION: APPROACHES FOR PREDICTING INTERACTIONS SPECIFIC TO PATIENT TREATMENT AND DRUG DESIGN 5.1 INTRODUCTION 5.2 AIM 5.3 MATERIALS AND METHODS 5.4 PROCEDURE 5.5 RESULTS	
6	MATLAB CERTIFICATES	

SYNERGIZING SYSTEM BIOLOGY AND MACHINE LEARNING: ADVANCED BIOMEDICAL RESEARCH

INTRODUCTION

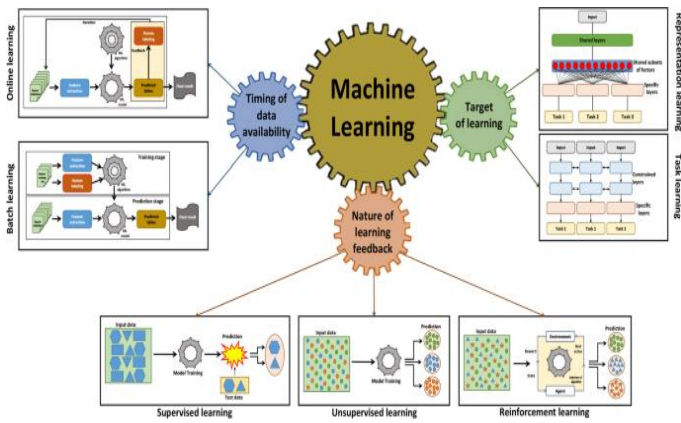
The fusion of system biology and machine learning represents a groundbreaking approach, promising to illuminate the intricate workings of biological systems and expedite scientific breakthroughs. System biology endeavors to decode biological processes by scrutinizing the dynamic interactions among molecules, cells, and organisms. In parallel, machine learning, empowered by copious datasets and computational prowess, excels in discerning patterns and forecasting outcomes from complex data. This convergence between system biology and machine learning harbors vast potential across

diverse realms of biomedical inquiry, ranging from unraveling disease mechanisms to innovating drug discovery and advancing personalized medicine. By amalgamating high-throughput omics data—such as genomics, transcriptomics, proteomics, and metabolomics—with sophisticated machine learning algorithms, researchers can glean profound insights into the underlying molecular underpinnings of diseases. This not only facilitates the discovery of novel biomarkers for early disease detection but also streamlines the development of tailored therapies tailored to individual patients' needs.

DATA INTEGRATION STRATEGIES FOR SYSTEM BIOLOGY

In data integration, feature engineering entails the selection or creation of informative features from heterogeneous data-sets. ML techniques, such as dimensionality reduction (e.g., PCA, t-SNE) and feature selection algorithms (e.g., Lasso regression, recursive feature elimination), can help identify the most relevant features for integration and analysis. ML algorithms, including decision trees, random forests, support vector machines (SVM), and neural networks, can be trained on integrated biological data to build predictive models for various biological phenomena, such as gene

function prediction, drug response prediction, and disease classification. ML clustering algorithms (e.g., k-means, hierarchical clustering) can identify natural groupings or clusters within integrated data-sets, revealing underlying biological patterns and relationships. This can help uncover functional modules in biological networks or stratify patients based on molecular profiles. ML techniques can be applied to analyze biological networks constructed from integrated data.



Graph-based ML algorithms, such as node classification, link prediction, and community detection, can extract meaningful insights from complex network structures, such as protein-protein interaction networks or gene regulatory networks. Transfer learning techniques enable the transfer of knowledge from one domain to another, which is particularly useful in systems biology where data is often sparse or noisy. Pre-trained ML models on related tasks or domains can be fine-tuned using integrated biological data to leverage existing knowledge and improve predictive

performance. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), can be applied to analyze omics data, such as genomic sequences, gene expression profiles, and protein structures. These models can capture intricate patterns and dependencies in biological data, leading to more accurate predictions and insights. ML algorithms, such as autoencoders and generative adversarial networks (GANs), can be used for data imputation and integration, filling missing values or harmonizing data from different sources. This ensures that integrated data-sets are more complete and consistent, enhancing downstream analysis and interpretation. ML techniques for interpretability, such as feature importance analysis and model explanation methods (e.g., SHAP, LIME), can provide insights into the underlying biological mechanisms captured by predictive models. This enhances the trustworthiness of ML-driven discoveries and facilitates biological interpretation.

SUPERVISED LEARNING APPROACHES

Supervised learning approaches in systems biology are computational methods used to analyze complex biological systems by learning from labeled data. These approaches leverage machine learning algorithms to make predictions or decisions based on input data and corresponding output labels. Classification algorithms are used to categorize biological entities into predefined classes or categories. For example, classifying genes into different functional categories based on gene expression profiles or classifying patients into different disease sub-types based on molecular signatures. Regression techniques are used to predict continuous outcomes based on input variables. In systems biology, regression analysis can be applied to predict quantitative traits such as protein abundance, metabolite concentrations, or gene expression levels based on various biological factors. Support Vector Machines is a popular supervised learning algorithm used for classification and regression tasks. SVM aims to find the optimal hyper-plane that separates different classes

in the feature space or predicts continuous outcomes while maximizing the margin between classes. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the model of the classes (classification) or the average prediction (regression) of the individual trees. It is particularly useful for handling high-dimensional data and dealing with over fitting. Neural networks, especially deep learning models, have gained popularity in systems biology for their ability to learn complex patterns from high-dimensional data. Convolutional Neural Networks (CNNs) are commonly used for analyzing biological images, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are suitable for sequential data analysis, such as time-series gene expression data. Decision trees are simple yet powerful models that recursively split the input space into regions based on the most informative features, leading to a tree-like structure. They are interpretable and can handle both numerical and categorical data, making them suitable for biological data analysis. Ensemble methods combine multiple models to improve prediction accuracy and robustness. Techniques such as bagging (Bootstrap Aggregating) and boosting (e.g., AdaBoost) are commonly used in systems biology to build diverse models and reduce over fitting.

UNSUPERVISED LEARNING APPROACHES

Unsupervised learning approaches in systems biology involve analyzing data without labeled examples. Grouping similar biological entities (genes, proteins, patients) based on their characteristics. Techniques like k-means clustering or hierarchical clustering are commonly used to identify patterns or subgroups within the data. Reducing the number of variables while preserving essential information. Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are widely used for visualizing high-dimensional biological data and identifying underlying structures. Identifying outliers or unusual patterns in biological data that deviate from the norm. This can be crucial for detecting rare diseases or identifying potential errors in experimental data. Discovering relationships or associations between different biological entities. Apriori algorithm and frequent item set mining are

examples of techniques used to find meaningful associations in biological datasets, such as co-occurrence of mutations in cancer patients. Representing biological entities and their relationships as networks or graphs. Graph clustering and community detection algorithms help uncover functional modules or pathways within biological networks, shedding light on complex interactions. Estimating the underlying probability distribution of biological data points. Kernel Density Estimation (KDE) and Gaussian Mixture Models (GMM) are used to identify regions of high density in the data, aiding in understanding the distribution of biological features. Revealing the underlying structure of high-dimensional data by embedding it into lower-dimensional spaces. Techniques like Isomap and Locally Linear Embedding (LLE) help uncover the intrinsic geometry of biological datasets, facilitating visualization and interpretation.

SEMI-SUPERVISED LEARNING APPROACHES

Semi-supervised learning in systems biology combines both labeled and unlabeled data to improve model performance and generalization. Initially, a small set of labeled data is used to train the model. In system biology, this could be gene expression data with known annotations or clinical data with diagnosed outcomes. Unlabeled data, which is often more abundant, is then incorporated

into the training process. This data may include gene expression profiles, protein-protein interaction networks, or other omics data without explicit annotations. Various algorithms are employed to leverage both labeled and unlabeled data. One common approach is co-training, where the model is trained on different subsets of features or samples. Another method is self-training, where the model

iteratively labels unlabeled data points with high confidence predictions and adds them to the training set. In systems biology, graph-based semi-supervised learning methods are prevalent. They represent biological entities and their relationships as graphs, where labeled nodes provide supervision for unlabeled nodes. Algorithms such as label propagation or graph-based semi-supervised classifiers use graph connectivity to propagate labels across the network. In semi-supervised learning, active learning techniques can be employed to select

the most informative instances from the unlabeled pool for labeling. These methods iteratively query an oracle (human expert or lab experiment) to label data points that are expected to maximize model performance. Transfer learning techniques can be adapted to systems biology by pretraining models on related tasks or datasets with abundant labeled data and fine-tuning them on the target task with limited labeled data. This approach leverages the knowledge learned from the source domain to improve performance on the target domain.

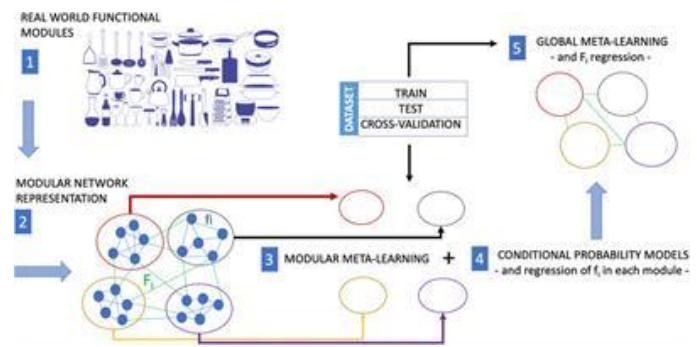
	Overview	Process	Subtypes	Examples
Supervised Learning	Majority of algorithms. Machine is trained using well-labeled data ; inputs and outputs are matched.	Mapping function takes inputs and matches to outputs, creating a target function.	Classification, Regression	Linear regression, Random forest, SVM.
Unsupervised Learning	Unlabeled data (inputs only) is analyzed. Learning happens without supervision.	Inputs are used to create a model of the data.	Clustering, Association.	PCA, k-Means, Hierarchical clustering.
Semi supervised	Some data is labeled , some not. Goal: better results than labeled data alone. Good for real world data.	Combination of above processes.	All the above.	Self training, Mixture models, Semi-supervised SVM

BIOLOGICAL NETWORK INFERENCE USING MACHINE LEARNING

Biological network inference using machine learning involves predicting molecular interactions, regulatory relationships, or functional associations from biological data. Biological data, such as gene

expression profiles, protein-protein interactions, or epigenetic modifications, are collected from experiments or databases. Relevant features are extracted from the data to represent biological

entities (e.g., genes, proteins) and their relationships. Features may include expression levels, sequence motifs, topological properties in networks, or other molecular characteristics. Machine learning models, such as Bayesian networks, neural networks, or graphical models, are trained on the data to infer biological networks. These models learn patterns from the input features and predict interactions or relationships between biological entities. The performance of the inferred networks is assessed using various metrics, such as precision, recall, or area under the receiver operating characteristic curve (AUC-ROC). Cross-validation or independent validation datasets are often used to evaluate the generalization ability of the models. To improve network inference accuracy, multiple types of biological data can be integrated into the modeling process. This includes combining gene expression data with protein-protein interaction networks, transcription factor binding data, or pathway information. To improve network inference accuracy, multiple types of biological data can be integrated into the modeling process



. This includes combining gene expression data with protein-protein interaction networks, transcription factor binding data, or pathway information. Prior biological knowledge, such as known interactions from databases or literature, can be integrated as priors or constraints into the network inference models. This helps guide the learning process and improves the biological relevance of the inferred networks. The inferred biological networks are visualized and analyzed to gain insights into molecular interactions, regulatory mechanisms, and functional modules within the system. Network visualization tools help researchers identify key regulators, pathways, or biological processes involved in the studied phenomenon.

FUTURE DIRECTIONS AND CHALLENGES IN THIS FIELD

The field of systems biology with machine learning is poised for significant advancements, but it also faces several challenges. One key direction is the integration of diverse omics data types (genomics, transcriptomics, proteomics, metabolomics, etc.) to gain a comprehensive understanding of biological systems. However, integrating heterogeneous data sources poses challenges related to data

harmonization, scalability, and interpretation. Biological systems exhibit substantial variability at different levels, from individual cells to populations. Future research must develop robust machine learning methods capable of capturing and modeling this variability effectively, especially in single-cell and spatial omics data analysis. While machine learning models can make accurate predictions, their

interpretability in the context of biology remains a challenge. Developing interpretable models and techniques for explaining model predictions is crucial for gaining insights into biological mechanisms and ensuring trustworthiness in biomedical applications. Biological systems are inherently dynamic and respond to various internal and external stimuli over time. Future research should focus on developing machine learning approaches capable of modeling dynamic processes, such as gene regulatory networks, signaling pathways, and cellular dynamics, from time-series and perturbation data. Machine learning in systems biology holds promise for advancing personalized medicine and precision health by leveraging individual-level omics data and clinical information. However, challenges related to data privacy, ethical considerations, and scalability need to be addressed to realize the full potential of these approaches in clinical practice. Integrating biological knowledge from diverse sources into structured knowledge graphs can facilitate data-driven hypothesis generation, drug discovery, and biomarker identification. Future research should focus on developing scalable algorithms for constructing and querying knowledge graphs and integrating them with machine learning models. Ensuring the reproducibility and transparency of research findings is crucial for advancing the field of systems biology with machine learning. Embracing open science practices, such as sharing data, code, and models, and adopting standardized benchmarks and evaluation protocols, can promote collaboration, validation, and knowledge dissemination.

Amrutha Murale

ORIGINALITY REPORT

7%

SIMILARITY INDEX

0%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Benjamin Shih, Dylan Shah, Jinxing Li, Thomas G. Thuruthel et al. "Electronic skins and machine learning for intelligent soft robots", Science Robotics, 2020

Publication

4%

2

Submitted to Laureate Education Inc.

Student Paper

3%

Exclude quotes Off

Exclude bibliography On

Exclude matches Off

MINI PROJECT

INTRODUCTION

A Biological network represents systems as complex set of binary interactions or relation between various biological systems and can reveal informative patterns within the systems. Cytoscape is a open source software platforms for visualizing molecule interactions network and biological pathway and integrating the network with annotations, gene expression profile. Cytohubba predict and explore the important nodes and edges in given network by several topological algorithm such as degree, bottleneck, MCC, DMNC, MNC, EPCv, Eccentricity, Closeness, Stress, Clustering co-efficient and betweenness.

ABSTRACT

This study aimed to elucidate the molecular mechanisms underlying biological processes or disease conditions through the analysis of gene expression data retrieved from the Gene Expression Omnibus (GEO) database and visualization of the results in Cytoscape.

AIM

To analyze gene expression data retrieved from the Gene Expression Omnibus (GEO) database and visualize the results in Cytoscape to elucidate the molecular mechanisms underlying biological processes or disease conditions.

MATERIALS AND METHODS

GEO2R

CYTOSCAPE

CYTOHUBBA

CLUEGO

GEO2R

GEO2R is a web-based tool provided by the Gene Expression Omnibus (GEO) that allows users to analyze gene expression data deposited in GEO datasets. It's designed to perform differential gene expression analysis, which helps identify genes that are differentially expressed between different experimental conditions or biological states.

CYTOSCAPE

Cytoscape is a powerful open-source software platform for visualizing, analyzing, and interpreting complex networks in biological systems. It is widely used by researchers in bioinformatics, systems biology, and computational biology to explore molecular interactions, pathways, and regulatory networks derived from various types of high-throughput data, such as gene expression, protein-protein interactions, and signaling pathways.

CYTOHUBBA

CytoscapeS CytoHubba is a plugin that offers various network analysis algorithms for identifying important nodes (e.g., proteins, genes) within biological networks. It focuses on finding nodes that are highly connected or have specific topological features, which can indicate their significance in the network's structure or function. Here's an overview of CytoHubba and its features.

CLUEGO

ClueGO is a Cytoscape plugin designed to analyze and visualize functional enrichment of gene ontology (GO) terms and biological pathways within biological networks. It is commonly used by researchers in bioinformatics and systems biology to gain insights into the biological significance of gene sets identified through high-throughput experiments, such as gene expression analysis or proteomics studies.

PROCEDURE

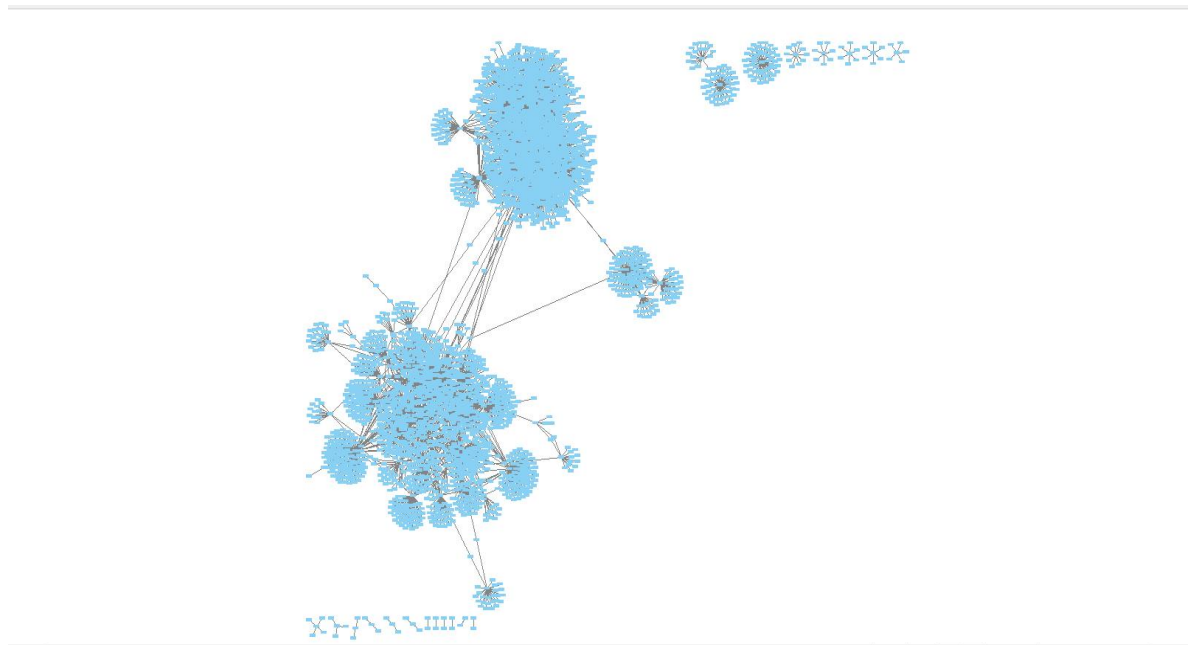
Procedure:

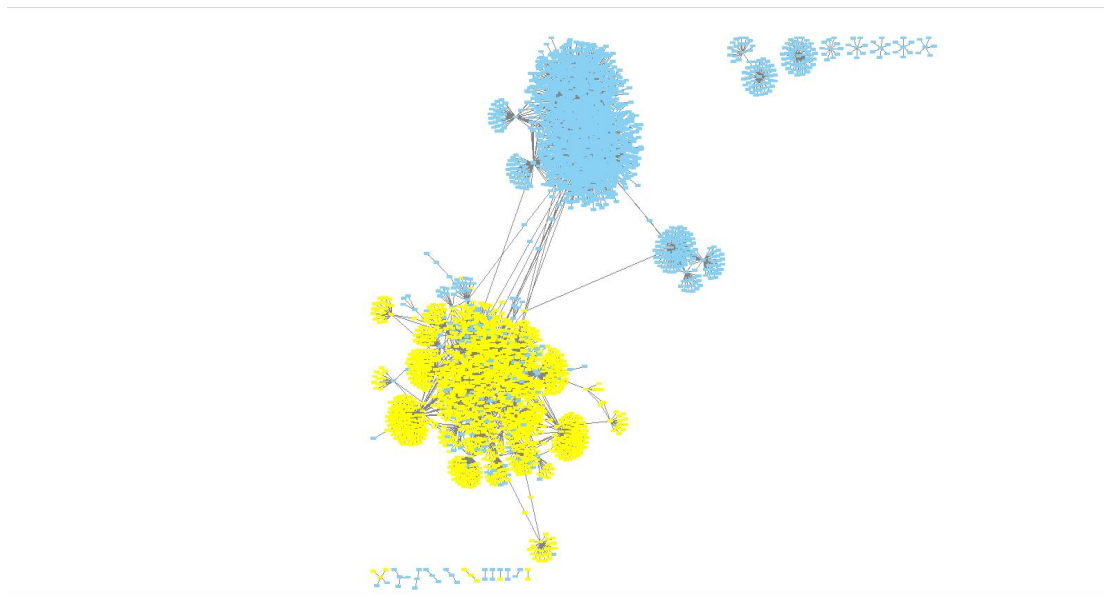
- For the database and disease pulmonary fibrosis Geo2R is the source of the geo dataset (GSE 50414)
- After choosing the suitable parameters, the Geo dataset is retrieved.
- Genes that were upregulated genes were copied and paste into the cytoscape network.
- Built a biological network using a BioGrid in cytohubba to analyse the network and identify importance node based various topologies algorithm to determine the the top 10 genes (IBMPFD3, ALS21, A1U, ALPS2B, Actr1a, UBC, TCF-3C, A1, PHA2, ENTRK1)
- Genes must be copied in cluego and configure cluego parameter such as species ontology and statistical settings.

- Run cluego analysis to identify functional categories or pathways associated with the selected nodes.

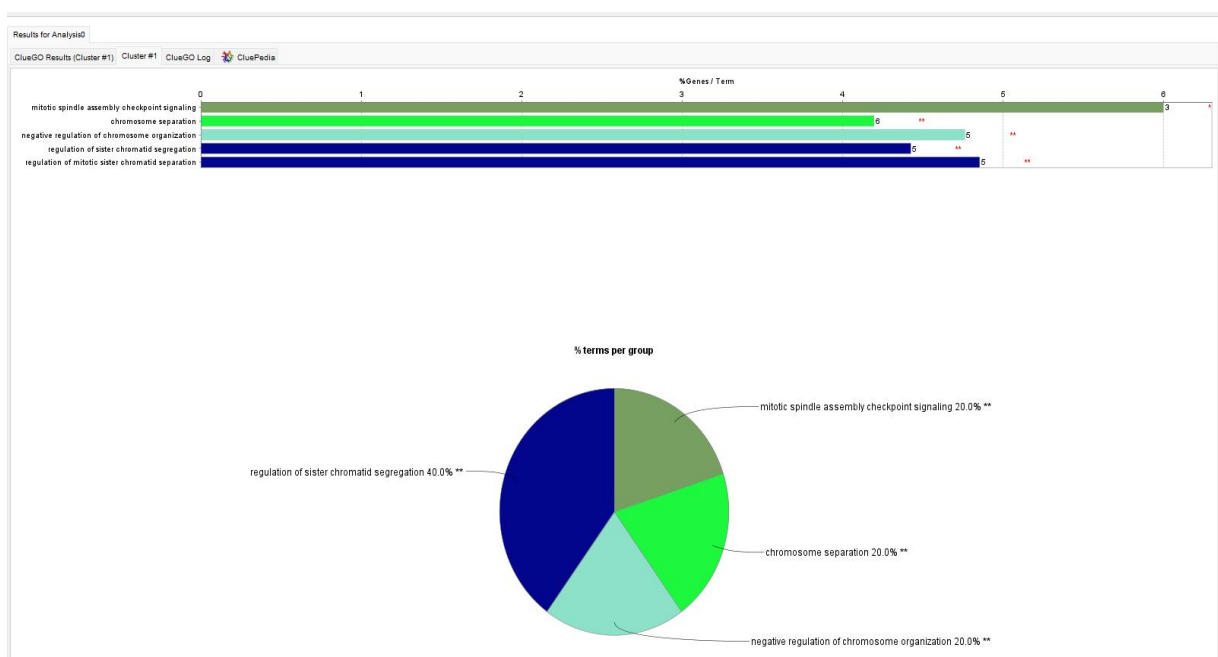
RESULT

CYTOSCAPE:





CLUEGO:



INTERPRETATION

Interpreting the process of analyzing gene expression data from GEO and visualizing it in Cytoscape involves understanding its significance in molecular biology research. The integration of gene expression data analysis from GEO with visualization in Cytoscape offers a powerful approach for unraveling the molecular underpinnings of biological processes and disease conditions. By leveraging publicly available data and network-based visualization

tools, researchers can gain actionable insights into gene regulation, pathway dynamics, and disease mechanisms, driving forward our understanding of complex biological systems.

Cumulative survival probability calculation using KM Plotter

INTRODUCTION:

KM-plotter is the most sophisticated online survival analysis tool, performing all calculations in real time (not loading pre-calculated images). The background database is manually curated. Gene expression data and relapse free and overall survival information are downloaded from GEO, EGA and TCGA. The database is handled by a PostgreSQL server, which integrates gene expression and clinical data simultaneously. To analyze the prognostic value of a particular gene, the patient samples are split into two groups according to various quantile expressions of the proposed biomarker. The two patient cohorts are compared by a Kaplan-Meier survival plot, and the hazard ratio with 95% confidence intervals and log rank P value are calculated. Databases and clinical data are supervised and extended regularly.

AIM

Calculate Proposed gene PRB1 cumulative survival probabilities.

MATERIALS AND METHODS:

KM-PLOTTER: KM-plotter is the most sophisticated online survival analysis ,performing all calculations in real time (not loading pre-calculated images). The background database is manually curated. Gene expression data and relapse-free and overall survival information are downloaded from GEO, EGA and TCGA.

Procedure:

Kaplan Meier plotter home page was accessed using the following hyperlink
<https://kmplot.com/analysis/index.php?p=background>

Proposed gene/protein MX1 was submitted in the pan cancer portal

Below listed parameters were opted for analysis

Restriction subtype –All

Split patients by-Auto select best cut off

Survival :Overall survival

Survival :All

Restrict analysis based on cellular content :All

Interpretation

The Kaplan-Meier curve shows the cumulative survival probability. A steeper slope indicates a higher event rate(death rate) and therefore a worse survival prognosis. A flatter slope indicate a lower event rate and therefore a better survival prognosis.

BLADDER CARCINOMA

The HR of PRC1 in bladder carcinoma is 1.28(0.94-1.75) 50% median survival between groups. It represent the ratio of probability for low expression cohort (35.9)and high expression cohort(27.3)

BREAST CANCER

The HR of PRC1 in breast cancer is 1.26(0.92-1.74) 50% median survival between groups . It represent the ratio of probability for low expression cohort (124.53)and high expression cohort(131.97)

CERVICAL SQUAMOUS CELL CARCINOMA

The HR of PRC1 in cervical squamous cell carcinoma is 0.56(0.32-0.98) 50% median survival between groups. It represent the ratio of probability for low expression cohort (96.27)and high expression cohort(103.23)

ESOPHAGEAL ADENOCARCINOMA

The HR of PRC1 in esophageal adenocarcinoma is 2.41(1.00-5.76) 50% median survival between groups. It represent the ratio of probability for low expression cohort (53.3)and high expression cohort(23.13)

ESOPHAGEAL SQUAMOUS CELL CARCINOMA

The HR of PRC1 in esophageal squamous cell carcinoma is 0.36(0.16-0.85) 50% median survival between groups. It represent the ratio of probability for low expression cohort (21.67)and high expression cohort(45.37)

HEAD-NECK SQUAMOUS CELL CARCINOMA

The HR of PRC1 in head-neck squamous cell carcinoma is 1.36(1.01-1.84) 50% median survival between groups. It represent the ratio of probability for low expression cohort (53.3)and high expression cohort(23.13)

KIDNEY RENAL CLEAR CELL CARCINOMA

The HR of PRC1 in kidney renal clear cell carcinoma is 2.43(1.79-3.31) 50% median survival between groups. It represent the ratio of probability for low expression cohort (53)and high expression cohort(18.7)

KIDNEY RENAL PAPILLARY CELL CARCINOMA

The HR of PRC1 in kidney renal papillary cell carcinoma is 5.15(2.83-9.37) 50% median survival between groups. It represent the ratio of probability for low expression cohort (89.47)and high expression cohort(19.9)

LIVER HEPATOCELLULAR CARCINOMA

The HR of PRC1 in liver hepatocellular carcinoma is 1.91(1.335-2.7) 50% median survival between groups. It represent the ratio of probability for low expression cohort (71.03)and high expression cohort(37.83)

LUNG ADENO CARCINOMA

The HR of PRC1 in lung adeno carcinoma is 2.05(1.52-2.77) 50% median survival between groups. It represent the ratio of probability for low expression cohort (54.4)and high expression cohort(31.67)

LUNG SQUAMOUS CELL CARCINOMA

The HR of PRC1 in lung squamous cell carcinoma is 0.76(0.6-1.03) 50% median survival between groups. It represent the ratio of probability for low expression cohort (49)and high expression cohort(61.37)

OVARIAN CANCER

The HR of PRC1 in ovarian cancer is 0.74(0.56-0.99) 50% median survival between groups. It represent the ratio of probability for low expression cohort (44.93)and high expression cohort(49.97)

PANCREATIC DUCTAL ADENO CARCINOMA

The HR of PRC1 in pancreatic ductal adeno carcinoma is 2.93(1.57-3.63) 50% median survival between groups. It represent the ratio of probability for low expression cohort (30.43)and high expression cohort(16.17)

PHEOCHROMOCYTOMA AND PARAGNANGLIOMA

The HR of PRC1 in pheochromocytoma and paragnanglioma is 2.41(1.00-5.76) 50% median survival between groups. It represent the ratio of probability for low expression cohort (53.3)and high expression cohort(23.13)

RECTUM ADENO CARCINOMA

The HR of PRC1 in rectum adeno carcinoma is 0.46(0.21-1.02) 50% median survival between groups. It represent the ratio of probability for low expression cohort (NA)and high expression cohort(NA)

SARCOMA

The HR of PRC1 in sarcoma is 2(1.25-3.19) 50% median survival between groups. It represent the ratio of probability for low expression cohort (47.47)and high expression cohort(20.93)

STOMACH ADENO CARCINOMA

The HR of PRC1 in stomach adeno carcinoma is 0.68(0.49-0.95) 50% median survival between groups. It represent the ratio of probability for low expression cohort (26.07)and high expression cohort(46.9)

Biomarkers: identifying biological markers for disease and Therapeutic monitoring

Diagnosis

INTRODUCTION

The Marker (a) systematically offers five types of therapeutic biomarker, which realized the collective consideration among different drug discovery stages, (b) comprehensively describes the therapeutic biomarker for the largest number of drugs among existing knowledge bases, and (c) extensively covers the largest classes of disease among available databases. Due to the rapid advances in AI technology, the comprehensive data collected by The marker were expected to have great implication and impact on the clinical development and clinical practice of drugs.

Aim

Identify the disease markers related to Lung cancer and categorize them into pharmacodynamic Biomarker ,Monitoring biomarker, safety biomarkers

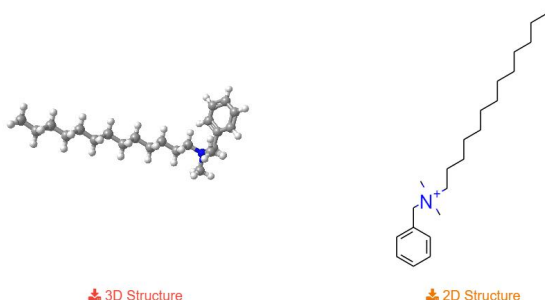
Procedure

- The marker database was retrieved using following link <http://themarker.idrblab.cn/about>
- Top listed Mrna/protein Biomarker relevant to the rheumatoid arthritis data's were retrieved and literature survey was done.

RESULT

The identification of the disease marker related to Breast Cancer and categorization of them into pharmoco dynamics Biomarker, Monitoring Biomarker and safety Biomarker as been done.

Structure





Biomarker ID	Biomarker Name	Class	Gene Name	Biomarker Type	Representative Drug	Representative Disease
B8UNA6	Neuromedin-U	Protein	NMU	P S M R U	Lapatinib	Breast cancer
BF2L8H	Multidrug resistance pr...	Protein	ABCB1; MDR1...	P S M R U	Neoadjuvant c...	Breast cancer
BKH4D0	Zinc finger CCHC dom...	Protein	LIN28A; CSDD...	P S M R U	Paclitaxel	Breast cancer
B0GWM3	Cardiac troponin T	Protein	TNNT2	P S M R U	Doxorubicin	Breast cancer
B6XD8U	Inhibin beta B chain	Protein	INHBB	P S M R U	Chemotherapy	Breast cancer
B7GHB8	Inhibin alpha chain	Protein	INHA	P S M R U	Chemotherapy	Breast cancer
B8E0FU	B-type natriuretic pepti...	Protein	NPPB	P S M R U	Doxorubicin	Breast cancer
B9BV5C	Myeloperoxidase	Protein	MPO	P S M R U	Doxorubicin	Breast cancer
BA31KD	Cardiac troponin I	Protein	TNNI3; TNNC1	P S M R U	Doxorubicin	Breast cancer
B5XF9I	Proliferation marker pr...	Protein	MKI67	P S M R U	Abemaciclib	Breast cancer
B63WAR	Estrogen receptor	Protein	ESR1; ESR; N...	P S M R U	Toremifene	Breast cancer
BCU6F8	Hormone receptor	Protein	.	P S M R U	Letrozole	Breast cancer
BF4XY0	ErbB2 tyrosine kinase r...	Protein	ERBB2; HER2;...	P S M R U	Trastuzumab	Breast cancer
B3UVG9	Metalloproteinase inhi...	Protein	TIMP1; CLGI; ...	P S M R U	Methotrexate	Breast cancer
B5Z0MO	Growth/differentiation ...	Protein	GDF15; MIC1; ...	P S M R U	Doxorubicin	Breast cancer
B80SDA	Nuclear factor erythroi...	Protein	NFE2L2; NRF2	P S M R U	Paclitaxel	Breast cancer
BS10QZ	Trefoil factor 3	Protein	TFF3; ITF; TFI	P S M R U	Endocrine ther...	Breast cancer
B2I4MF	Placenta growth factor	Protein	PGF; PGFL; PL...	P S M R U	Doxorubicin	Breast cancer
B3HTD6	C-reactive protein	Protein	CRP; PTX1	P S M R U	Trastuzumab	Breast cancer
B10T8Y	Mucin-1	Protein	MUC1; PUM	P S M R U	Prednisone	Hypersensitivity
B9FT0Z	Epidermal growth fact...	Protein	EGFR; ERBB; ...	P S M R U	Abemaciclib	Breast cancer
B5AEU2	BRCA1 mRNA	mRNA	BRCA1; RNF53	P S M R U	Epirubicin	Breast cancer
BOC81M	DNMT3B	Gene	.	P S M R U	Immune check...	Breast cancer
B2DGF6	BRCA2	Gene	FAD; FAD1; BR...	P S M R U	Olaparib	Breast cancer

INTERPRETATION

BIO-MARKER ID	BIO-MARKER NAME	CLASS	GENE NAME	BIO MARKER TYPE	REPRESENTATIVE DRUG	REPRESENTATIVE DISEASE
B3HT06	C-Reactive	Protein	CRP,PTX	PDY,SAF,MOI,PRO	Trustuzumab	Breast cancer

DRUG-TARGET INTERACTION PREDICTION: APPROACHES FOR PREDICTING INTERACTIONS SPECIFIC TO PATIENT TREATMENT AND DRUG DESIGN

INTRODUCTION

Drug targets are specific molecules or biological structures involved in disease pathways that can be modulated by drugs to produce a therapeutic effect. Selecting the right drug target is crucial for the development of effective and safe medications.

Swiss Target Prediction: Swiss Target Prediction is an online tool to predict the macromolecular targets (proteins from human, mouse and rat) of bioactive small molecules. This is useful to understand the molecular mechanisms underlying a given phenotype or bioactivity, to rationalize possible side-effects, to predict off-targets and to assess the possibility of repurposing therapeutically-relevant compounds.

Swiss Target Prediction is based on the so-called 'similarity principle', which generally states that two similar molecules are prone to have similar properties. In the context of Swiss Target Prediction, we have statistically quantify that two similar bioactive molecules are likely to share their protein targets. Therefore, for a query molecule, we identify the most similar molecules among a set of about 370'000 actives on known targets. The predicted targets are those having the actives displaying the highest similarity with the query molecule.

Aim:

Drug Benzalkonium targets screening and binding assessment.

Materials and Methods:**Swiss Target Prediction:**

Uniprot: UniProt is a freely accessible database of protein sequence and functional information, many entries being derived from genome sequencing projects. It contains a large amount of information about the biological function of proteins derived from the research literature. It is maintained by the UniProt consortium, which consists of several European bioinformatics organisations and a foundation from Washington, DC, USA.

PDB:

The Protein Databank (PDB) is a file format describing the 3D structure of molecules held in protein data bank. The PDB format accordingly provides for description and annotations of protein and nucleic acid structures including atomic co-ordinates, secondary structure assignments as well atomic connectivity.

Drug Bank: Drug bank database is a comprehensive, freely accessible online database containing information on drugs and drug targets. This database is a unique bioinformatics and chemi-informatics resource that combines detail drug.

Swiss Target Prediction: `Swiss target prediction is a online tool to predict the targets of bioactive small molecules in human and other vertebrates. This is useful to understand the molecular mechanism underlying a given phenotype or bioactivity.

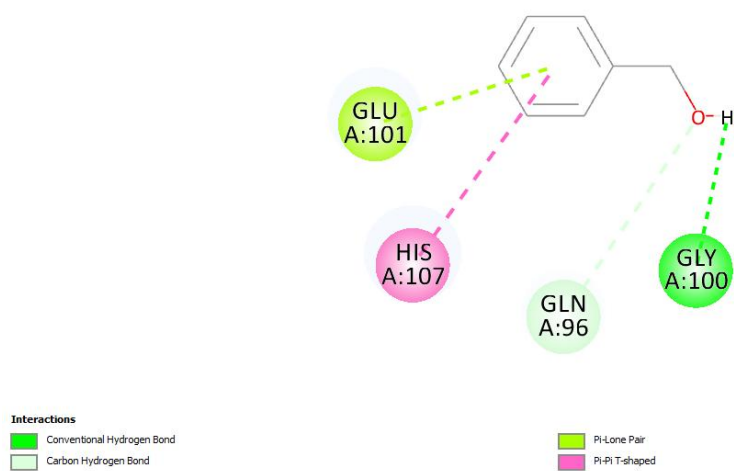
NON BONDING TABLE

NAME	DISTANCE	CATEGORY	TYPE	FROM	TO
244:H16 - A:GLY100:O	2.25848	Hydrogen Bond	Conventional Hydrogen Bond	244:H16	A:GLY100:O
A:GLN96:CA - 244:O1	3.24463	Hydrogen Bond	Carbon Hydrogen Bond	A:GLN96:CA	244:O1
A:GLU101:O - 244	2.89289	Other	Pi-Lone Pair	A:GLU101:O	244
A:HIS107 - 244	4.54152	Hydrophobic	Pi-Pi T-shaped	A:HIS107	244

MOLECULE TABLE

NAME	-PLP1	-PLP2	JAIN	-PMF	DOCK_SCORE
244	31.14	36.9	-0.19	5.12	20.226
244	31.14	36.9	-0.19	5.12	20.226
244	31.14	36.9	-0.19	5.12	20.226
244	31.79	37.21	0.17	6.14	20.22
244	32.19	37.81	-0.15	5.68	20.102
244	36.39	41.43	-0.17	6.85	20.028
244	36.39	41.43	-0.17	6.85	20.028
244	36.16	41.54	0.07	5.98	20.025
244	36.16	41.54	0.07	5.98	20.025
244	36.59	42.01	0.02	7.67	20.012

STRUCTURE



3D STRUCTURE

IMAGE 1:

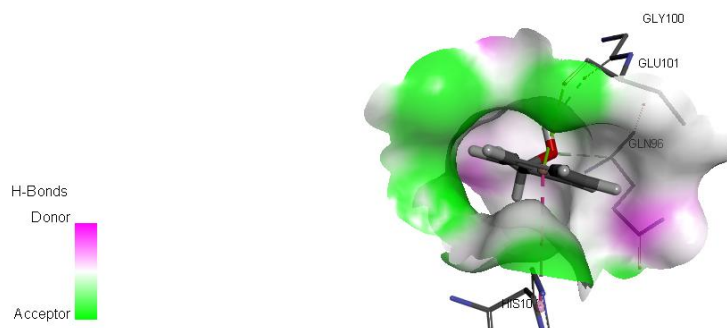
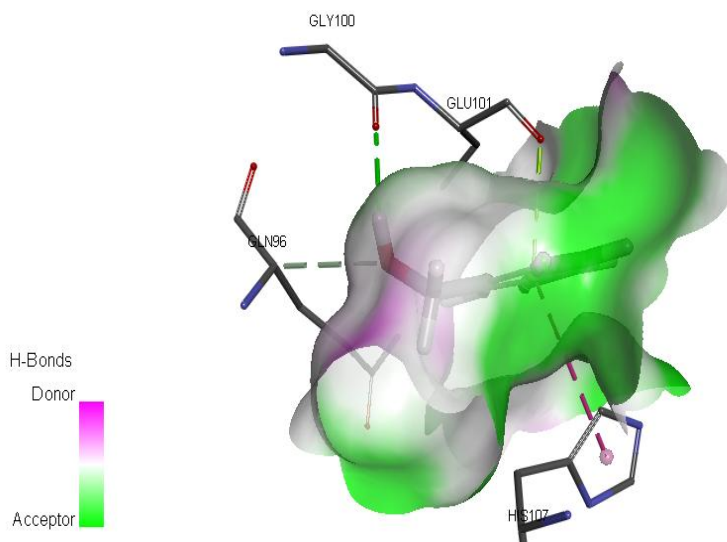


IMAGE 2:



INTERPRETATION

Benzalkonium chloride (BZK) is commonly used in pharmaceuticals as an antimicrobial agent. It's found in various over-the-counter medications, including throat lozenges, nasal sprays, and eye drops. In these formulations, benzalkonium chloride helps prevent the growth of bacteria and other microorganisms, which can contaminate the product or cause infections. JM21 refers to a gene, it would be helpful to know its function or relevance to understand any potential relationship with benzalkonium chloride.

1.MATLAB ONRAMP

INTRODUCTION:

MATLAB Onramp is a free, interactive tutorial that helps beginners learn MATLAB, a high-level programming language and environment used in technical computing and data visualization. It covers the basics of MATLAB syntax, programming, and problem-solving techniques. It's a great starting point for anyone new to MATLAB.

WHAT WE LEARN

- Introduction to matlab
- Basic operations
- Vectors and matrices
- Plotting and visualisation
- Programming concepts
- Advanced data types
- File I/O
- Symbolic mathematics
- Applications

2.OBJECT ORIENTATED PROGRAMMING ONRAMP

INTRODUCTION:

"Object-Oriented Programming Onramp" likely refers to a learning path or course designed

to introduce individuals to the fundamentals of object-oriented programming (OOP). In OOP, programs are organized around objects, which are instances of classes that encapsulate data and behavior. It's a popular programming paradigm used in many languages like Java, C++, and Python.

WHAT WE LEARN

- Introduction of oop
- Classes and object
- Inheritance
- Polymorphism
- Encapsulation
- Abstraction
- Method overriding and overloading
- Interfaces and abstract classes
- Design principles
- Practical examples and exercise

3.MATLAB FUNDAMENTALS

INTRODUCTION:

MATLAB Fundamentals typically refer to a course or material that covers the basics of MATLAB, a high-level programming language and environment primarily used for numerical computing and data analysis. These fundamentals may include topics like syntax, data types, matrices, plotting, functions, and basic programming concepts tailored to MATLAB's unique features and applications.

WHAT WE LEARNT

- Basic Syntax and commands
- Variable and data types
- Basic operations and functions
- Control flow statements
- Array manipulation and indexing
- Plotting and visualisation
- Basic debugging techniques

4.MACHINE LEARNING ONRAMP

INTRODUCTION:

"MACHINE LEARNING ONRAMP" likely refers to a structured learning path or course designed to provide an introduction to machine learning concepts and techniques. This onramp could cover foundational topics such as supervised learning, unsupervised learning, regression, classification, clustering, and evaluation methods. It might also include hands-on exercises or projects to help learners gain practical experience in applying machine learning algorithms to real-world problems.

WHAT WE LEARNT

- Introduction machine learning concepts and terminology
- Data preprocessing techniques such as cleaning normalisation and features scaling
- Supervised learning algorithms including linear regression, logistic regression, decision tree and k nearest neighbors
- Evaluation metrics for assessing model performance

- Over fitting, under fitting and techniques for model validation and selection

5.MACHINE LEARNING WITH MATLAB

INTRODUCTION

"Machine Learning with MATLAB" is a comprehensive approach to mastering the principles and applications of machine learning using MATLAB, a powerful programming language and environment. In this introduction, you'll explore how MATLAB provides a versatile platform for implementing various machine learning algorithms, from classical techniques to cutting-edge deep learning models. Throughout your journey, you'll learn how to preprocess data, build and train models, evaluate performance, and deploy solutions, all within the MATLAB environment. Whether you're a beginner or an experienced practitioner, this course offers valuable insights and practical skills to excel in the rapidly evolving field of machine learning.

WHAT WE LEARNT

- 1.** Introduction to MATLAB's machine learning capabilities and toolboxes.
- 2.** Data preprocessing techniques such as normalization, feature extraction, and dimensionality reduction.
- 3.** Supervised learning algorithms including regression (linear regression, support vector regression) and classification (k-nearest neighbors, decision trees, ensemble methods).
- 4.** Unsupervised learning algorithms like clustering (k-means clustering, hierarchical clustering) and dimensionality reduction (principal component analysis, t-distributed stochastic neighbor embedding).
- 5.** Evaluation metrics and techniques for assessing model performance.
- 6.** Deep learning fundamentals, including neural network architectures and training strategies.

Practical skills for implementing machine learning and deep learning algorithms using MATLAB's built-in functions and toolboxes, as well as handling large datasets efficiently.

6. DEEP LEARNING ONRAMP

INTRODUCTION:

"Deep Learning Onramp" likely refers to a structured learning path or course designed to introduce individuals to the principles and applications of deep learning. Deep learning is a subset of machine learning that utilizes neural networks with multiple layers to learn intricate patterns and representations from data. This onramp may cover foundational concepts of deep learning, including neural network architecture, optimization techniques, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and popular deep learning frameworks such as TensorFlow or PyTorch. Additionally, it might include hands-on exercises and projects to help learners gain practical experience in implementing deep learning models for various tasks such as image classification, natural language processing, and more.

WHAT WE LEARNT

1. Introduction to deep learning concepts, including neural networks, layers, and activation functions.
2. Building and training simple neural networks using frameworks like TensorFlow or PyTorch.
3. Convolutional Neural Networks (CNNs) for image recognition and classification tasks.
4. Recurrent Neural Networks (RNNs) for sequence modeling and prediction.
5. Transfer learning techniques for leveraging pre-trained models on new tasks.
6. Optimization algorithms and techniques for training deep learning models efficiently.
7. Model evaluation and deployment considerations.
8. Hands-on experience with practical projects and exercises to reinforce learning.

7.DEEP LEARNING WITMATLAB

INTRODUCTION:

"Deep Learning with MATLAB" refers to using MATLAB, a high-level programming language and environment, to implement and experiment with deep learning algorithms and models. MATLAB provides a comprehensive set of tools and functions for deep learning, including prebuilt neural network architectures, optimization algorithms, and visualization tools. With MATLAB, users can easily preprocess data, design and train deep neural networks, fine-tune models, and deploy them for various applications such as image classification, object detection, speech recognition, and more. The integration of deep learning capabilities within MATLAB makes it a convenient and powerful platform for both beginners and experts in the field of deep learning.

WHAT WE LEARNT

- 1.** Introduction to deep learning concepts, including neural networks, layers, and activation functions.
- 2.** Building and training neural networks using MATLAB's Deep Learning Toolbox.
- 3.** Convolutional Neural Networks (CNNs) for image classification, object detection, and segmentation tasks.
- 4.** Recurrent Neural Networks (RNNs) for sequence modeling and prediction, such as in natural language processing or time series analysis.
- 5.** Transfer learning techniques for fine-tuning pre-trained models on new datasets.
- 6.** Optimization algorithms and techniques for training deep learning models efficiently.
- 7.** Hyperparameter tuning and model evaluation strategies.
- 8.** Deployment considerations, including exporting trained models for inference in production environments.
- 9.** Practical projects and exercises to apply deep learning concepts to real-world datasets and problems.

8. COMPUTER VISION ONRAMP

INTRODUCTION

"Computer Vision Onramp" likely refers to a structured learning path or course designed to introduce individuals to the fundamentals of computer vision. Computer vision is a field of artificial intelligence that focuses on enabling machines to interpret and understand visual information from the real world. This onramp may cover foundational concepts such as image processing, feature extraction, object detection, image classification, and semantic segmentation. Additionally, it might include hands-on exercises and projects to help learners gain practical experience in implementing computer vision algorithms using popular frameworks like OpenCV or deep learning libraries such as TensorFlow or PyTorch.

WHAT WE LEARN

- 1.** Fundamentals of computer vision, including image representation and processing techniques.
- 2.** Image filtering and feature extraction methods, such as edge detection and corner detection.
- 3.** Object detection and recognition algorithms, including template matching and Haar cascades.
- 4.** Image segmentation techniques for identifying and extracting objects from images.
- 5.** Feature matching and registration methods for aligning images or detecting objects across multiple images.
- 6.** Introduction to deep learning-based approaches for computer vision tasks, such as convolutional neural networks (CNNs) for image classification and object detection.
- 7.** Practical skills for implementing computer vision algorithms using libraries like OpenCV or MATLAB's Computer Vision Toolbox.
- 8.** Hands-on projects and exercises to reinforce learning and apply computer vision techniques to real-world problems.

9.REINFORCEMENT LEARNING ONRAMP

INTRODUCTION

"Reinforcement Learning Onramp" likely refers to a structured learning path or course designed to introduce individuals to the principles and applications of reinforcement learning (RL). Reinforcement learning is a type of machine learning paradigm where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. This onramp may cover foundational concepts such as Markov decision processes, policy evaluation, policy improvement, value iteration, Q-learning, and deep reinforcement learning. Additionally, it might include hands-on exercises and projects to help learners gain practical experience in implementing reinforcement learning algorithms for various tasks such as game playing, robotics, and autonomous systems.

WHAT WE LEARNT

- 1.** Introduction to reinforcement learning (RL) concepts, including agents, environments, actions, and rewards.
- 2.** Markov decision processes (MDPs) as a formal framework for modeling RL problems.
- 3.** Dynamic programming methods such as value iteration and policy iteration for solving MDPs.
- 4.** Model-free RL algorithms, including Q-learning and SARSA, for learning optimal policies directly from experience.
- 5.** Deep reinforcement learning techniques, which combine deep learning with RL for handling high-dimensional state and action spaces.
- 6.** Exploration-exploitation trade-offs and techniques for balancing exploration and exploitation in RL agents
- 7.** Function approximation methods for scaling RL algorithms to large state and action spaces.
- 8.** Practical implementation of RL algorithms using libraries like TensorFlow, PyTorch, or RLlib.
- 9.** Applications of reinforcement learning in various domains, such as robotics, game playing, and autonomous systems.

10. STATISTICS ONRAMP

INTRODUCTION

"Statistics Onramp" likely refers to a structured learning path or course designed to introduce individuals to the fundamentals of statistics. This onramp may cover essential statistical concepts such as descriptive statistics, probability theory, hypothesis testing, confidence intervals, regression analysis, and ANOVA (analysis of variance). Additionally, it might include hands-on exercises and projects to help learners understand how to apply statistical methods to analyze data, make informed decisions, and draw meaningful conclusions. Whether you're new to statistics or looking to refresh your knowledge, a Statistics Onramp can provide a solid foundation for understanding and working with data.

WHAT WE LEARN

1. Fundamental concepts of descriptive statistics, including measures of central tendency (mean, median, mode) and variability (range, variance, standard deviation).
2. Probability theory, including basic probability rules and distributions such as the normal distribution, binomial distribution, and Poisson distribution.
3. Inferential statistics techniques for making inferences about populations based on sample data, including hypothesis testing and confidence intervals.
4. Understanding and interpreting p-values and confidence intervals in statistical analysis.
5. Regression analysis, including simple linear regression and multiple regression, for modeling relationships between variables.
6. Analysis of variance (ANOVA) for comparing means across multiple groups.
7. Basic concepts of experimental design and statistical power.
8. Practical skills for performing statistical analysis using software such as R, Python with libraries like NumPy and SciPy, or statistical software packages like SPSS or SAS. Hands-on projects and exercises to apply statistical concepts to real-world datasets and problems.

11.MATLAB FOR DATA PROCESSING AND VISUALISATION

INTRODUCTION

"MATLAB for Data Processing and Visualization" refers to using MATLAB, a powerful programming language and environment, for tasks related to handling and visualizing data. In this context, MATLAB provides a wide range of tools and functions for processing, analyzing, and visualizing data in various formats, including numerical data, time series, images, and more. Users can leverage MATLAB's built-in functions for data manipulation, statistical analysis, filtering, interpolation, and other preprocessing tasks. Additionally, MATLAB offers sophisticated plotting and visualization capabilities, allowing users to create high-quality plots, charts, and interactive visualizations to explore and present their data effectively. Overall, MATLAB is a versatile tool for data scientists, engineers, and researchers to perform data processing and visualization tasks efficiently.

WHAT WE LEARNT

- 1.** Data importing and exporting techniques to read data from various file formats such as CSV, Excel, and text files.
 - 2.** Data cleaning and preprocessing methods including handling missing data, removing outliers, and filtering noisy data.
 - 3.** Data manipulation and transformation techniques such as sorting, reshaping, and merging datasets.
 - 4.** Basic statistical analysis including calculating descriptive statistics, performing hypothesis tests, and fitting probability distributions.
 - 5.** Visualization techniques for exploring and presenting data, including plotting histograms, scatter plots, line plots, and bar graphs.
 - 6.** Customizing plots and adding annotations to enhance data visualization.
 - 7.** Interactive data exploration using MATLAB's graphical user interface(GUI) tools.
 - 8.** Advanced data visualization techniques such as 3D plotting, heatmap visualization, and geographic mapping.
- Exporting visualizations for publication or presentation purposes

12. IMAGE PROCESSING ONRAMP

INTRODUCTION

IMAGE PROCESSING ONRAMP is likely an introductory course or learning program focused on image processing. It could be offered by a university, online platform, or company to help individuals gain foundational knowledge and skills in image processing techniques, such as filtering, segmentation, feature extraction, and object recognition.

WHAT WE LEARNT

- 1.** Fundamentals of digital image representation, including grayscale and color images.
- 2.** Basic image processing operations such as image resizing, cropping, and rotation.
- 3.** Image enhancement techniques including contrast adjustment, histogram equalization, and noise reduction.
- 4.** Spatial domain filtering methods such as smoothing and sharpening filters.
- 5.** Frequency domain techniques like Fourier Transform for image analysis and filtering.
- 6.** Edge detection algorithms such as Sobel, Prewitt, and Canny for detecting edges in images.
- 7.** Image segmentation methods for partitioning an image into meaningful regions.
- 8.** Feature extraction techniques for extracting relevant information from images.
- 9.** Introduction to computer vision concepts and applications.
- 10.** Practical skills for implementing image processing algorithms using MATLAB's Image Processing Toolbox or libraries like OpenCV.

Hands-on projects and exercises to apply image processing techniques to real-world image datasets and problems.

13.IMAGE PROCESSING WITH MATLAB

INTRODUCTION

"Image Processing with MATLAB" is a course or a set of resources designed to teach image processing techniques using the MATLAB software. MATLAB is a programming environment and language commonly used in engineering and scientific fields, including image processing. This course likely covers topics such as image enhancement, filtering, segmentation, feature extraction, and object recognition, all using MATLAB's built-in functions and toolboxes. It's a practical way to learn how to apply image processing algorithms using MATLAB's powerful tools.

WHAT WE LEARNT

- 1.** Basic image representation and manipulation techniques, including loading, displaying, and saving images.
 - 2.** Image enhancement methods such as contrast adjustment, histogram equalization, and noise reduction.
 - 3.** Spatial domain filtering techniques for smoothing and sharpening images using various filter kernels.
 - 4.** Frequency domain analysis using Fourier Transform for image filtering and analysis.
 - 5.** Edge detection algorithms like Sobel, Prewitt, and Canny for detecting edges in images.
 - 6.** Image segmentation techniques for partitioning images into meaningful regions.
 - 7.** Morphological operations such as dilation, erosion, opening, and closing for shape analysis and feature extraction.
 - 8.** Feature extraction methods for extracting relevant information from images, such as texture analysis and object recognition.
 - 9.** Introduction to advanced topics such as image registration, image stitching, and image restoration.
 - 10.** Practical implementation of image processing algorithms using MATLAB's Image Processing Toolbox.
- Hands-on projects and exercises to apply image processing techniques to real-world image datasets and problems.

14.SIGNAL PROCESSING WITH MATLAB

INTRODUCTION

"Signal Processing with MATLAB" is a course or educational resource focused on teaching signal processing techniques using the MATLAB software. Signal processing involves analyzing, manipulating, and interpreting signals, which could be audio, video, or any other type of data that varies over time or space. MATLAB provides a rich set of functions and toolboxes specifically designed for signal processing tasks, making it a popular choice for engineers, researchers, and students working in this field. This course likely covers topics such as filtering, spectral analysis, time-frequency analysis, and digital signal processing algorithms, all implemented using MATLAB.

WHAT WE LEARNT

- 1.** Basics of signal representation and analysis, including continuous-time and discrete-time signals.
- 2.** Time-domain and frequency-domain analysis techniques, such as Fourier Transform, Discrete Fourier Transform (DFT), and Fast Fourier Transform (FFT).
- 3.** Filtering operations, including design and implementation of digital filters such as low-pass, high-pass, band-pass, and band-stop filters.
- 4.** Convolution and correlation operations for signal processing tasks.
- 5.** Windowing techniques for signal analysis and spectral estimation.
- 6.** Spectral analysis methods, including periodogram, power spectral density (PSD), and spectrogram.
- 7.** Introduction to adaptive filtering techniques for applications such as noise cancellation and system identification.
- 8.** Practical implementation of signal processing algorithms using MATLAB's Signal Processing Toolbox.

Hands-on projects and exercises to apply signal processing techniques to real-world signal datasets and problems.

15.MATLAB PROGRAMMING TECHNIQUES

INTRODUCTION

"MATLAB Programming Techniques" refers to strategies, methods, and best practices for writing efficient and effective MATLAB code. This could include topics such as:

- 1. Syntax and language features:** Understanding MATLAB syntax, datatypes, and built-in functions.
- 2. Vectorization:** Leveraging MATLAB's ability to perform operations on entire arrays instead of individual elements for faster execution.
- 3. Memory management:** Writing code that minimizes memory usage and optimizes memory access patterns.
- 4. Code organization:** Structuring MATLAB code into functions, scripts, and classes for better readability and maintainability.
- 5. Debugging and error handling:** Techniques for identifying and fixing errors in MATLAB code, as well as handling exceptions gracefully.
- 6. Performance optimization:** Strategies for improving the performance of MATLAB code, such as preallocation, avoiding unnecessary computations, and using efficient algorithms.
- 7. MATLAB toolboxes:** Utilizing built-in MATLAB toolboxes and libraries to streamline development and leverage specialized functionality.

WHAT WE LEARNT

- 1. Basic syntax and programming constructs in MATLAB,** including variables, data types, operators, and control flow statements (if-else, loops).
- 2. Writing modular and reusable code** using functions and scripts.
- 3. Vectorization techniques** for optimizing code performance and efficiency.
- 4. Debugging and error handling strategies** to identify and fix issues in MATLAB code.
- 5. MATLAB programming best practices,** including code readability, documentation, and naming conventions.
- 6. Advanced data structures** such as cell arrays, structures, and tables for organizing and manipulating data.
- 7. File I/O operations** for reading and writing data to files in various formats (e.g., text files, Excel files).
- 8. Object-oriented programming (OOP) concepts** in MATLAB, including classes, objects, inheritance, and polymorphism.

9. MATLAB's built-in libraries and toolboxes for specialized tasks such as signal processing, image processing, and optimization.
10. Profiling and optimization techniques for improving code performance and efficiency.
11. Version control and collaborative coding practices using MATLAB's integration with Git and GitHub

16. WIRELESS COMMUNICATION ONRAMP

INTRODUCTION

"Wireless Communication ONRAMP" sounds like a specialized learning program or course designed to introduce individuals to the fundamentals of wireless communication technologies. This could include topics such as radio frequency (RF) communication, modulation techniques, multiple access schemes, channel coding, and wireless networking protocols. The term "ONRAMP" suggests that it's an entry-level or introductory program, aimed at providing a foundational understanding of wireless communication concepts. It could be offered by educational institutions, online platforms, or companies as a means for individuals to gain essential knowledge in this field.

WHAT WE LEARNT

1. Introduction to wireless communication onramp
2. How they are simulate a basic digital communication link ,anatomy of digital communication system and modulation and demodulation
3. Simulate a digital filters
4. Multipath channels OFDM modulation and practical consideration and null subcarriers.

17.SOLVING NON-LINEAR EQUATIONS WITH MATLAB

INTRODUCTION:

Solving nonlinear equations with MATLAB involves using numerical methods to find approximate solutions to equations that cannot be solved analytically. MATLAB provides several built-in functions and tools for solving nonlinear equations, including:

- 1. fsolve:** This function finds a solution to a system of nonlinear equations. It requires an initial guess for the solution and returns the solution that satisfies the equations.
- 2. fzero:** This function finds a root of a single-variable nonlinear function. It requires an initial guess for the root and returns the value of the root.
- 3. Nonlinear equation solvers in optimization toolboxes:** MATLAB's optimization toolboxes provide additional functions for solving nonlinear equations, including `lsqnonlin` for least squares problems and `lsqcurvefit` for curve fitting.

To solve a nonlinear equation in MATLAB, you typically define the equation as a function or anonymous function, specify initial guesses for the solution, and then use one of the above functions to find the solution.

MATLAB's documentation provides detailed information and examples for using these functions to solve nonlinear equations.

WHAT WE LEARNT

In a course on solving nonlinear equations with MATLAB, you typically learn:

- 1.** Introduction to nonlinear equations and their importance in various fields such as engineering, physics, and economics.
- 2.** Different methods for solving nonlinear equations, including numerical methods like the Newton-Raphson method, the secant method, and the bisection method.
- 3.** Understanding the principles behind each method, including their convergence properties and limitations.
- 4.** Implementing these methods in MATLAB using built-in functions or custom scripts.
- 5.** Handling complex nonlinear systems of equations using iterative methods and numerical techniques.

6. Evaluating the accuracy and efficiency of different solution methods for nonlinear equations.
7. Strategies for dealing with singularities, multiple solutions, and other challenges that may arise in nonlinear equation solving.
8. Applications of nonlinear equation solving in practical problems, such as root-finding in optimization and parameter estimation tasks.

Overall, the course aims to equip students with the knowledge and skills necessary to effectively solve nonlinear equations using MATLAB, enabling them to tackle a wide range of mathematical and engineering problems.

18.SIGNAL PROCESSING ONRAMP

INTRODUCTION

"Signal Processing with MATLAB" is a course or educational resource focused on teaching signal processing techniques using the MATLAB software. Signal processing involves analyzing, manipulating, and interpreting signals, which could be audio, video, or any other type of data that varies over time or space. MATLAB provides a rich set of functions and toolboxes specifically designed for signal processing tasks, making it a popular choice for engineers, researchers, and students working in this field. This course likely covers topics such as filtering, spectral analysis, time-frequency analysis, and digital signal processing algorithms, all implemented using MATLAB.

WHAT WE LEARNT

1. Basics of signal representation and analysis, including continuous-time and discrete-time signals.
2. Time-domain and frequency-domain analysis techniques, such as Fourier Transform, Discrete Fourier Transform (DFT), and Fast Fourier Transform (FFT).
3. Filtering operations, including design and implementation of digital filters such as low-pass, high-pass, band-pass, and band-stop filters.
4. Convolution and correlation operations for signal processing tasks.
5. Windowing techniques for signal analysis and spectral estimation.
6. Spectral analysis methods, including periodogram, power spectral density (PSD), and spectrogram.

7. Introduction to adaptive filtering techniques for applications such as noise cancellation and system identification.
8. Practical implementation of signal processing algorithms using MATLAB's Signal Processing Toolbox.
9. Hands-on projects and exercises to apply signal processing techniques to real-world signal datasets and problems.

COURSE CERTIFICATES

1.MATLAB ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

MATLAB Onramp



DIRECTOR, TRAINING SERVICES

15 November 2023

2.MATLAB FUNDAMENTAL



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

MATLAB Fundamentals



3.MATLAB FOR DATA PROCESSING ND VISUALIZATION



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

MATLAB for Data Processing and Visualization

DIRECTOR, TRAINING SERVICES

5 March 2024

4 MATLAB PROGRAMMING TECHNIQUES



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

MATLAB Programming Techniques

DIRECTOR, TRAINING SERVICES

5 OBJECT-ORIENTED PROGRAMMING ONRAMP



6 SIMULINK ONRAMP



7 MACHINE LEARNING ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Machine Learning Onramp



DIRECTOR, TRAINING SERVICES

29 February 2024

8 MACHINE LEARNING WITH MATLAB



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Machine Learning with MATLAB



DIRECTOR, TRAINING SERVICES

8 March 2024

9 DEEP LEARNING ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Deep Learning Onramp


DIRECTOR, TRAINING SERVICES

4 March 2024

10 DEEP LEARNING WITH MATLAB



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Deep Learning with MATLAB


DIRECTOR, TRAINING SERVICES

9 March 2024

11 REINFORCEMENT LEARNING ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Reinforcement Learning Onramp

A handwritten signature in black ink, reading "Craig L. Santos".

12 COMPUTER VISION ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Computer Vision Onramp

A handwritten signature in black ink, reading "Craig L. Santos".

DIRECTOR, TRAINING SERVICES

21 March 2024

13 STATISTICS ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Statistics Onramp

A handwritten signature in black ink, reading "Craig Santos".

14 IMAGE PROCESSING ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Image Processing Onramp

A handwritten signature in black ink, reading "Craig Santos".

15 SOLVING NONLINEAR EQUATIONS WITH MATLAB



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Solving Nonlinear Equations with MATLAB

A handwritten signature in black ink, reading "Rajesh Santos".

16 IMAGE PROCESSING WITH MATLAB



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Image Processing with MATLAB

A handwritten signature in black ink, reading "Rajesh Santos".

DIRECTOR, TRAINING SERVICES

13 March 2024

17 COMPUTER VISION ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Computer Vision Onramp



DIRECTOR, TRAINING SERVICES

21 March 2024

18 SINGAL PROCESSING ONRAMP



Course Completion Certificate

Amrutha M

has successfully completed **100%** of the self-paced training course

Signal Processing Onramp



DIRECTOR, TRAINING SERVICES

10 March 2024