An Introduction to Machine Learning for Bioinformatics

Presented by

Matin Ghasemi
M.Sc. Student, Bioinformatics
Amirkabir University of Technology (AUT)

Expectations

- Machine learning is a powerful technique and is widely used in many fields
- Machine learning is a difficult subject with many sub-disciplines that take years to master and it usually requires advanced training in math, computer science and statistics
- A 3-hour workshop will not make you an expert in machine learning
- This is an introductory course to give you a “taste” of ML and hopefully to inspire you to learn more on your own

Open Dialogue and Q&A

This workshop encourages an open dialogue. Feel free to ask questions, share insights, and connect with fellow participants. The collective learning experience is valuable, and we're here to support your understanding of machine learning concepts.

Contact Information:

- Telegram: @MatinGhasemi99
- Email: itsmatinghasemi@gmail.com

Presentation’s Sources

- Machine learning for bioinformatics course by Dr.Rohban and Dr.Sharifi
- A course in machine learning book by Hal Daume
- Machine learning for bioinformatics workshop by Dr. Wishart
- Assistance from ChatGPT for content refinement

Part 1: Introduction to Machine Learning

Key objectives:

- To introduce, define and differentiate the concept of machine learning
- To explain the standard ML workflow
- To show examples of ML dataset and its key characteristics

Learning

Any process by which an organism or system improves performance from experience

For example consider taking a biology course. At the end of the course you will be expected to have “learn” about this topics.

How can the professor test this?

- Ask only question that has been answered exactly in the course?
- Give you an exam on History of Art?
- You observe specific examples of problems in Biology and would be asked to answer new relevant question in the exam

Machine Learning

“ A set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty. ”

Machine learning definition from Murphy’s book

An example of an ML task

Heart attack risk prediction machine learning system:

Features (x)					
age	sex	trtbps	chol	fbs	thalachh
56	1	120	236	0	178
57	0	120	354	0	163
52	1	172	199	1	162
51	0	130	305	0	142
58	1	128	216	0	131

Label (y)
heart attack
1
1
1
0
0

An example of an ML task

Components of a learning system (classification):

- Training data

age	sex	trtbps	chol	fbs	thalachh
56	1	120	236	0	178
51	0	130	305	0	142

heart attack
1
0

- Three principal components of a classification problem
  - Class label (aka "label" or "targets", denoted by y)
  - Features (aka "attributes")
  - Features values (denoted by x)
- A labeled dataset is a collection of (x, y) pairs

An example of an ML task

Components of a learning system (classification):

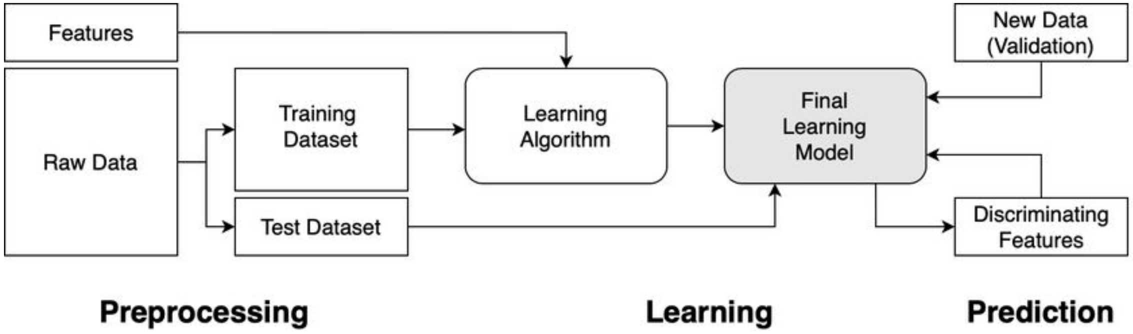
- Test set

age	sex	trtbps	chol	fbs	thalachh
57	0	120	354	0	163
58	1	128	216	0	131

heart attack
1
0

- Predict the class of the “test” examples
- Requires us to generalize from the training data

General framework of induction



- The test set id closely guarded secret. It is the final exam on which our learning algorithm is being tested
- If our algorithm gets to peek at ahead of time, it’s going to cheat and do better that it should

Three Approaches to ML

- Supervised Learning
  - Given example inputs and desired or labeled outputs with the goal being to learn rules that map inputs to outputs
- Unsupervised Learning
  - Given unlabeled data try to learn rules to find structure in the input data
- Reinforcement Learning
  - Learning to solve a problem by being given continuous feedback to maximize rewards

Canonical Learning Problems

- Regression
  - Definition: Predicts a real number as the output.
  - Example: Estimating house prices based on features like size, location, and amenities.
- Classification
  - Definition: Assigns a categorical value as the output, either binary or from multiple classes.
  - Example: Classifying emails as spam or not spam based on content features.
- Ranking
  - Definition: Establishes an order on input items, assigning a relative position.
  - Example: Ranking search results based on relevance to a user's query.
- Recommendation System
  - Definition: Suggests items based on user preferences, enhancing personalized experiences.
  - Example: Recommending movies on a streaming platform based on a user's viewing history and preferences.

Part 2: Machine Learning Key Concepts

Key objectives:

- To Formalize the learning process and Evaluate the method
- To familiarize with inductive bias and its importance
- To explain the underfitting and overfitting problems
- To explain model evaluation methods

Formalizing the learning

Required elements:

- Performance metric
- Evaluate the method on “unseen” test data
- Training and test data should have strong relationship

Performance metric

- Loss function: $l(y, \hat{y})$, measures how desirable an output $\hat{y}$ is w.r.t y for a given x
- Regression
  - Squared loss: $l(y, \hat{y}) = (y - \hat{y})^2$
  - Absolute loss: $l(y, \hat{y}) = |y - \hat{y}|$
- Classification
  - Binary loss: $l(y, \hat{y}) = 0 \; if \;\; y = \hat{y} \; and \, 1 \; otherwise$

A model for training and test data

- Each (x, y), regardless of training or test, comes from the same probability distribution.
    - $D(x, y) = D(y|x)D(x)$
- We would never be given D!
- Training and test data are independent i.i.d. draws from D(x,y)
    - Train Set in independent of test set (given D)

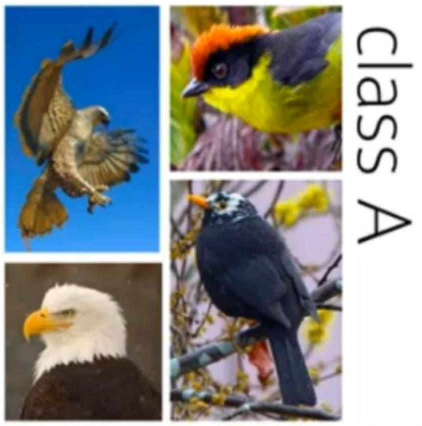
Define error

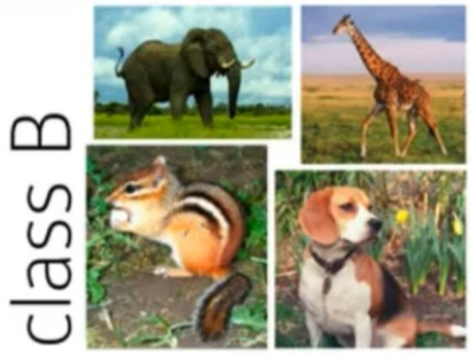
Required elements:

- Find a hypothesis f, such that the average loss in minimized:
- Average is taken on all reasonable samples
    - $\epsilon = \mathbb{E}_{(x,y)}[l(y, f(x))]$
- Unfortunately, it is not possible.
- We should instead rely on training data to find f:
    - $\hat{\epsilon} = \frac{1}{N}\Sigma_{n=1}^{N}[l(y_n, f(x_n))]$

Inductive bias

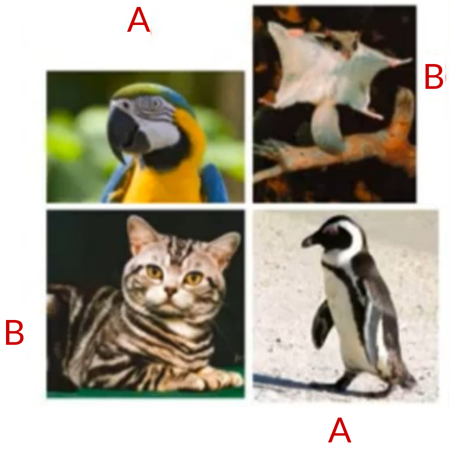
Suppose that you are given 8 training samples for two classes A and B

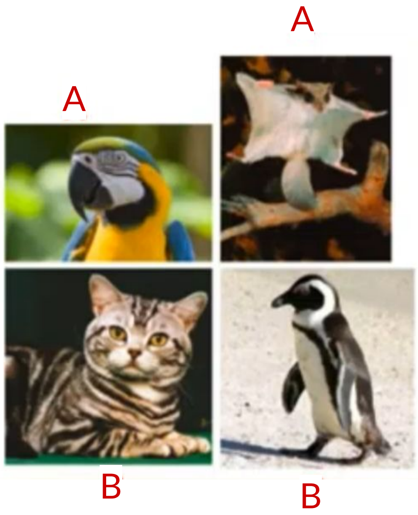




Inductive bias

Which model is better?





Inductive bias

- Correct inductive bias is necessary for a problem to be learnable.
- If we imagine that all possible methods are equally likely to match any training data, then, according to the "No Free Lunch" theorem, learning becomes impossible in this approach.
- Different approaches in ML are different type of biases

Let’s dive into the example

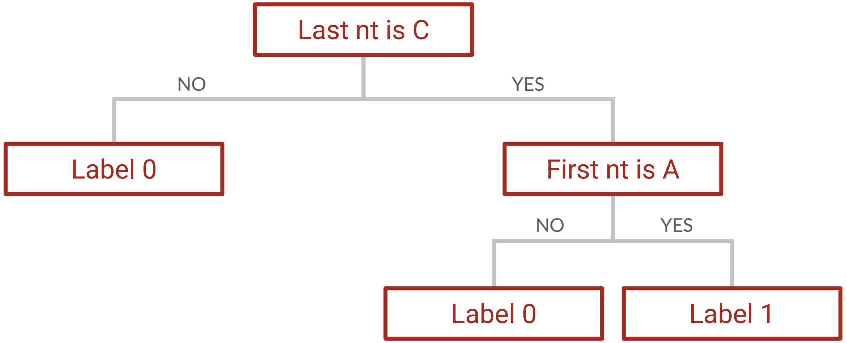
Predict whether a sequence of DNA belongs to a specific gene or not.

DNA Sequence	Label
ATCGATCGATCGATCGA	1
ATCGATCGATCGATCGT	1
ATCGATCGATCGATCGC	0
CTCGATCGATCGATCGG	0
ATCGATCGATCGATCGA	1

Decision Tree

The Decision Tree algorithm is a simple machine learning method that makes decisions by recursively partitioning data based on the most informative features, creating a tree-like structure for classification or regression tasks.

DNA Sequence	Label
ATCGATCGATCGATCGA	1
ATCGATCGATCGATCGT	1
ATCGATCGATCGATCGC	0
CTCGATCGATCGATCGG	0



Last nt is C

NO

YES

Label 0

First nt is A

NO

YES

Label 0

Label 1

Underfitting

Imagine you decide to use a very basic model that oversimplifies the task. For example, you build a decision tree with a very shallow depth, perhaps just one level.

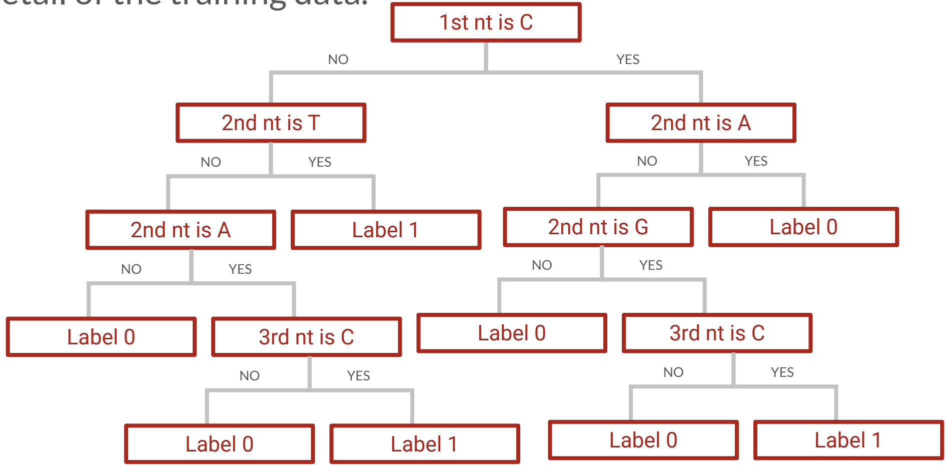
DNA Sequence	Label
ATCGATCGATCGATCGA	1
ATCGATCGATCGATCGT	1
ATCGATCGATCGATCGC	0
CTCGATCGATCGATCGG	0

Label 0

Overfitting

On the other hand, let's say you use an extremely complex model, perhaps you build an overly complex decision tree with a deep depth, capturing every detail of the training data.

DNA Sequence	Label
ATCGATCGATCGATCGA	1
ATCGATCGATCGATCGT	1
ATCGATCGATCGATCGC	0
CTCGATCGATCGATCGG	0



1st nt is C

NO

YES

2nd nt is T

2nd nt is A

NO

YES

NO

YES

2nd nt is A

Label 1

2nd nt is G

Label 0

NO

YES

NO

YES

Label 0

3rd nt is C

Label 0

3rd nt is C

NO

YES

NO

YES

Label 0

Label 1

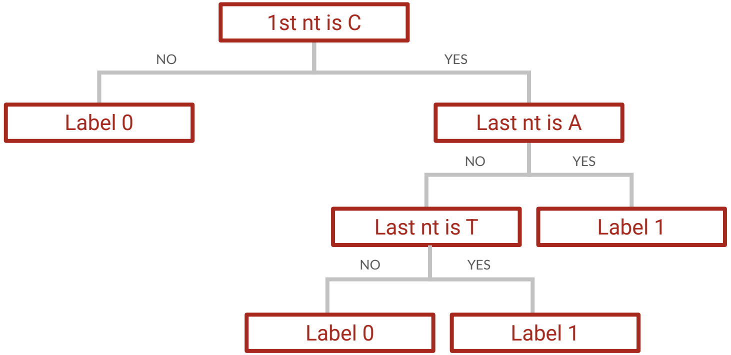
Label 0

Label 1

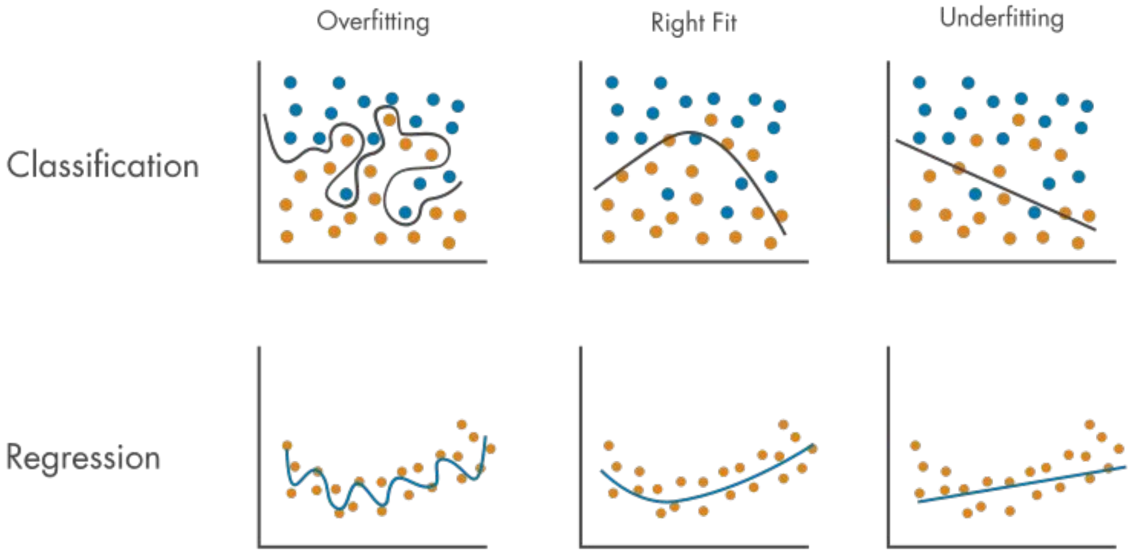
Balanced

A balanced decision tree would have an appropriate depth, capturing the essential patterns without memorizing every detail.

DNA Sequence	Label
ATCGATCGATCGATCGA	1
ATCGATCGATCGATCGT	1
ATCGATCGATCGATCGC	0
CTCGATCGATCGATCGG	0



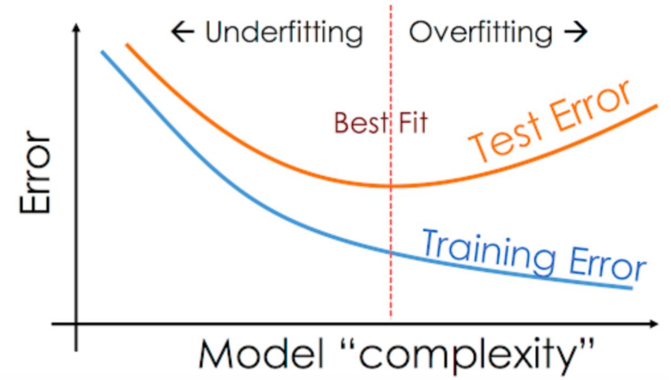
Underfitting vs. Overfitting



How to detect overfitting?

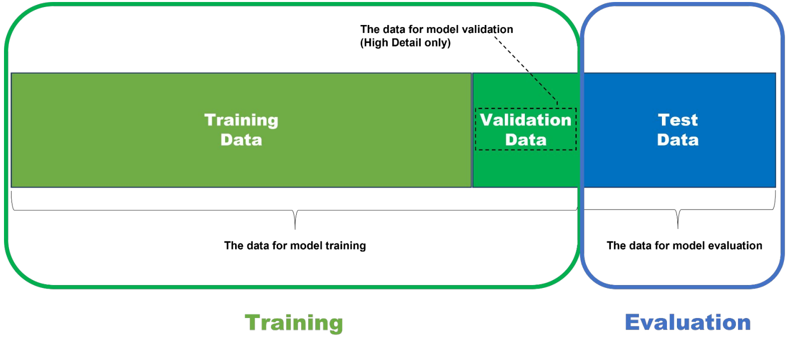
Let $\hat{\epsilon_{te}}$ be the empirical error on the test set.

- Can be thought of as an approximation to the error on unseen data
- If $\hat{\epsilon}$ is small and $\hat{\epsilon_{te}}$ is large, it means the learner is probably overfitted



Never ever touch your test data

- Never touch your test data, because the learner might get overfitted to that as well.
- Use validation dataset to improve the performance of the model during its training phase and keep test dataset out of reach of model or even yourself.


The data for model validation
(High Detail only)
Training Data
Validation Data
Test Data
The data for model training
The data for model evaluation
Training
Evaluation

Not everything is learnable

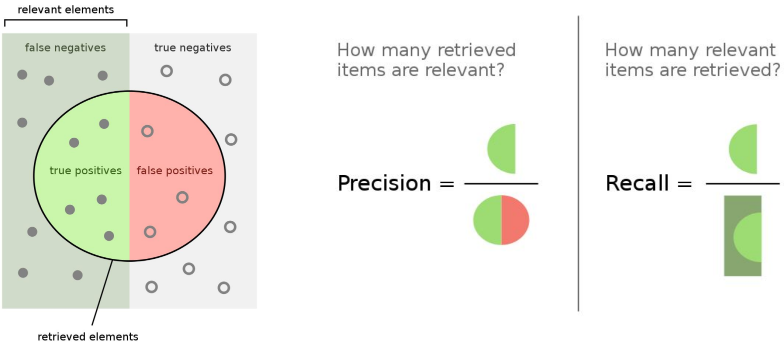
- Noise
- Insufficient or non-informative features
- Inductive bias is not correct

Parameters vs. Hyperparameter

- Parameters are the internal variables that a machine learning model learns from the training data. For example in linear regression, the slope and intercept are parameters. The model adjusts these values during training to fit the data.
- Hyperparameters are external configuration settings that are not learned from the data but are set before training. For example depth of decision tree, and the number of decision trees in a random forest.

Evaluating model performance

- Accuracy is not always a good metric
    - Face detection (1 in million patches is a face)
    - Accuracy of the classifier that always says no = 99.9999%
- Precision and recall



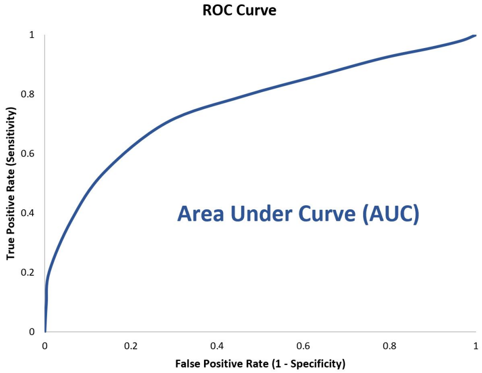
Evaluating model performance

- F1 score
  - F1 score is Harmonic mean of precision and recall
  - Accuracy of the classifier that always says no = 0%
- Formula

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluating model performance

- Sensitivity and Specificity
  - A sensitive classifier is one which almost always finds everything it is looking for. It has high recall
  - A specific classifier is one which does a good job not finding the things that it doesn’t want to find
- Receiver Operating Characteristic (ROC curve)
  - Plots the sensitivity against 1-specificity



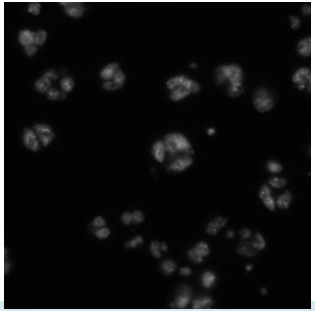
Part 3: ML for Bioinformatics

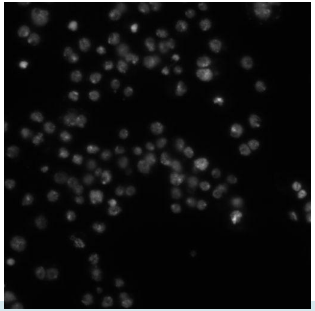
Key objectives:

- To show example of ML in bioinformatics problem
- To familiarize with bioinformatics datasets’ challenges

Bio example

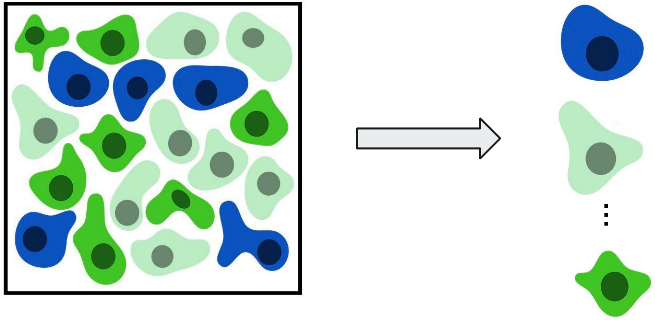
- Imagine different compound (drug candidate) with given mechanism (y) are tested in different plates on numerous cultured cells. A readout is made for each cell (x)
- Objective is to classify mechanisms given the readouts.





Single cell approach

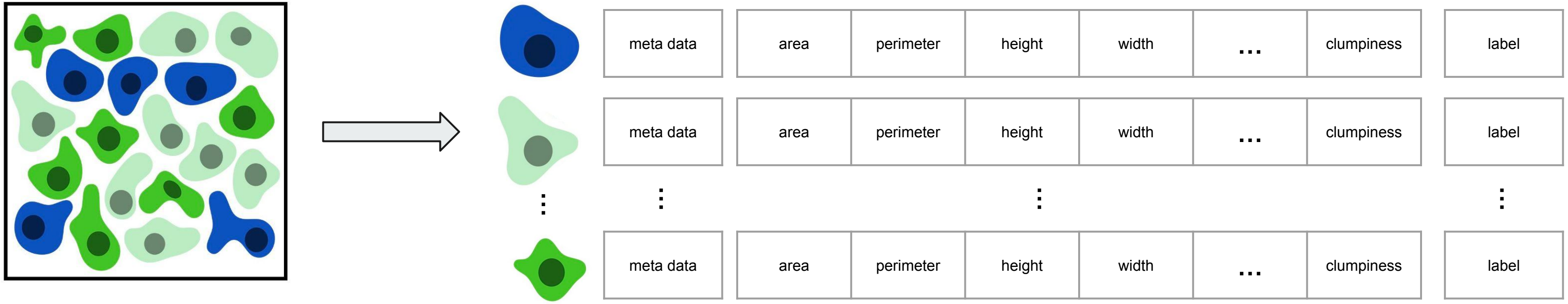
- Perform image segmentation to identify and isolate individual cells in your images
- Extract relevant features from each segmented cell.



area	perimeter	height	width	...	clumpiness	label
area	perimeter	height	width	...	clumpiness	label
		⋮				⋮
area	perimeter	height	width	...	clumpiness	label

Train-test split

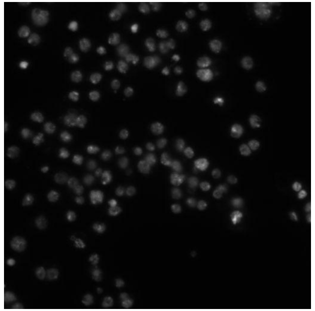
- According to the training model, training and test sets should be independent given the data distribution.
- Does random split of single cell data work?

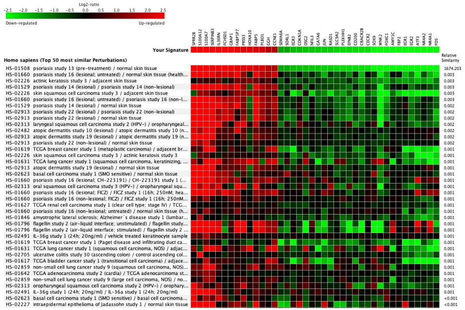


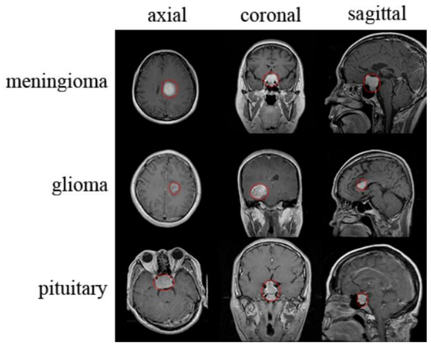
meta data	area	perimeter	height	width	...	clumpiness	label
meta data	area	perimeter	height	width	...	clumpiness	label
⋮			⋮				⋮
meta data	area	perimeter	height	width	...	clumpiness	label

Undrasting data and its Correctness

Understanding bioinformatics data can be more challenging than handling normal data due to its complexity, heterogeneity, and the specialized knowledge required in biology and genetics.




Log2-ratio
-2.5 -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5
Down-regulated
Up-regulated
SPRR2B S100A12 S100A7 SERPINB3 IL36RN FCHSD1 GBAP1 FABP5P7 PRSS3 HOXA10 FABP5 PLBD1 GGH CCNE1 SEMA6A SORL1 EGR3 CDKN1A DSG2 NFIL3 GATA6 JUN RASD1 SLC9A2 PLEKHH1 EFNB2 CD200 CRACR2B CECR2 SOX9 WNK2 FOXC1 AMY1C FBP2 EGR1 EGR2 ATF3 NR4A2 NR4A1 FOS
Your Signature
Homo sapiens (Top 50 most similar Perturbations)
Relative Similarity
HS-01508 psoriasis study 13 (pre-treatment) / normal skin tissue 1674.203
HS-01660 psoriasis study 16 (lesional; untreated) / normal skin tissue (health... 0.003
HS-02226 actinic keratosis study 3 / adjacent skin tissue 0.003
HS-01529 psoriasis study 14 (lesional) / psoriasis study 14 (non-lesional) 0.003
HS-02226 skin squamous cell carcinoma study 3 / adjacent skin tissue 0.003
HS-01660 psoriasis study 16 (lesional; untreated) / psoriasis study 16 (non-l... 0.003
HS-01529 psoriasis study 14 (lesional) / normal skin tissue 0.002
HS-02913 psoriasis study 22 (lesional) / psoriasis study 22 (non-lesional) 0.002
HS-02913 psoriasis study 22 (lesional) / normal skin tissue 0.002
HS-02313 laryngeal squamous cell carcinoma study 2 (HPV-) / oropharyngeal... 0.002
HS-02482 atopic dermatitis study 10 (lesional) / atopic dermatitis study 10 (n... 0.002
HS-02913 atopic dermatitis study 19 (lesional) / atopic dermatitis study 19 (n... 0.002
HS-02913 psoriasis study 22 (non-lesional) / normal skin tissue 0.002
HS-01619 TCGA breast cancer study 1 (metaplastic carcinoma) / adjacent br... 0.001
HS-02226 skin squamous cell carcinoma study 3 / actinic keratosis study 3 0.001
HS-01631 TCGA lung cancer study 1 (squamous cell carcinoma, keratinizing, ... 0.001
HS-02913 atopic dermatitis study 19 (lesional) / normal skin tissue 0.001
HS-02623 basal cell carcinoma study 1 (SMO sensitive) / normal skin tissue 0.001
HS-01660 psoriasis study 16 (lesional; CH-223191) / CH-223191 study 1 (... 0.001
HS-02313 oral squamous cell carcinoma study 3 (HPV-) / oropharyngeal squ... 0.001
HS-01660 psoriasis study 16 (lesional; FICZ) / FICZ study 1 (16h; 250nM; hea... 0.001
HS-01660 psoriasis study 16 (non-lesional; FICZ) / FICZ study 1 (16h; 250nM... 0.001
HS-01627 TCGA renal cell carcinoma study 1 (clear cell type; stage IV) / TCG... 0.001
HS-01660 psoriasis study 16 (non-lesional; untreated) / normal skin tissue (h... 0.001
HS-01846 amyotrophic lateral sclerosis; Alzheimer´s disease study 1 (lumbar... 0.001
HS-01796 flagellin study 2 (air-liquid interface; unstimulated) / flagellin study... 0.001
HS-01796 flagellin study 2 (air-liquid interface; stimulated) / flagellin study 2 ... 0.001
HS-02491 IL-36g study 1 (24h; 20ng/ml) / vehicle treated keratinocyte sample 0.001
HS-01619 TCGA breast cancer study 1 (Paget disease and infiltrating duct ca... 0.001
HS-01631 TCGA lung cancer study 1 (squamous cell carcinoma, NOS) / adjac... 0.001
HS-02705 ulcerative colitis study 30 (ascending colon) / control ascending col... 0.001
HS-01617 TCGA bladder cancer study 1 (transitional cell carcinoma) / adjace... 0.001
HS-02859 non-small cell lung cancer study 9 (squamous cell carcinoma, NOS)... 0.001
HS-01642 TCGA adenocarcinoma study 2 (cardia) / TCGA adenocarcinoma st... 0.001
HS-02859 non-small cell lung cancer study 9 (large cell carcinoma, NOS) / no... 0.001
HS-02313 oropharyngeal squamous cell carcinoma study 2 (HPV-) / orophary... 0.001
HS-02491 IL-36g study 1 (24h; 20ng/ml) / IL-36a study 1 (24h; 20ng/ml) 0.001
HS-02623 basal cell carcinoma study 1 (SMO sensitive) / basal cell carcinoma... <0.001
HS-02227 intraepidermal epithelioma of Jadassohn study 1 / normal skin tissue <0.001



Non-bioinformatics example for bioinformatics

- During the Vietnam War, the U.S. military supposedly attempted to use image classification algorithms to identify tanks in the jungle. The algorithm performed well during testing, correctly classifying tanks from non-tank images.
- However, when the algorithm was deployed in the field, it failed miserably.
- So what happens?

Understand the knowledge learned by the model

- It turned out that the model wasn't actually identifying tanks but was instead picking up on subtle differences in lighting conditions between the training and deployment environments.
- The model essentially learned to distinguish between sunny and cloudy days, rather than detecting the presence of tanks.

Part 4: Machine learning in action

Key objectives:

- To get familiarized with ML code using Python
- To understand bio-data and its visualization
- To solve a ML task from scratch

Development environment

Machine learning development environments can vary based on factors like the type of machine learning task, the scale of the project, and the preferences of the developers.

- Local
  - Jupyter notebook
  - Common IDE likes PyCharm, Visual Studio Code, Spyder
- Cloud-based
  - Google Colab

Python libraries

- Data Manipulation and Analysis
    - NumPy
    - Pandas
- Data Visualization
    - Matplotlib
    - Seaborn
- Machine Learning Frameworks
    - Scikit-learn
- Neural Networks and Deep Learning
    - Pytorch (not cover in this workshop)
    - TensorFlow and Keras (not cover in this workshop)

Thank You

An Introduction to Machine Learning for Bioinformatics
Presented by Matin Ghasemi

IBIS
Iranian
Bioinformatics
Society

Student Symposium