Supervised Learning:

SVM

* Supervised learning
* The distance between the points (support vectors) and the line (hyperplane) should be as far as possible
* Hyperplane → the line that separates the two groups and has the maximum distance from the support vectors (points)
* Support vectors → the extreme points in the dataset
* D+ → shortest distance to the closest positive point
* D- → shortest distance to the closest negative point
* Distance margin → [sum of D+ and D-] the distance between the two support vectors
* Optimal hyperplane can be found by finding the distance margin
* Misclassification to occur if hyperplane selected having low margin
* Kernel function → used for 2D/3D output - where it takes 1D input and transfer it to 2d output or takes 2D input and transfer it to 3D output
* Regularization parameter → avoids overfitting and biased problems in SVM

Random Forest

* Supervised learning - classification
* Reduces risk of overfitting
* Runs efficiently on large database
* Estimates missing data - maintaining accuracy
* Constructs multiple decision trees during training phase
* Decision tree
  + Entropy → measure of randomness or predictability in the dataset
  + Information gain → measure of decrease in entropy after the dataset has been split [info gain = E1-E2]
  + Leaf node → carries classification or decision - the tree is split by some means
  + Decision node → has 2 or more branches
  + Root node → top most decision node known as the root node
  + Splitting the data first by the condition that gives us the highest gain
  + Not further splitting is required for the node which reached an entropy of 0
  + The node with still a fairly high entropy is further split by another condition that was not used

Cell CNN

* Deep learning model used for image recognition
* Classify cells based on morphology (shape and structure)
* How it works:
  + Input:
    - A dataset with multiple samples - where sample contains many cells (a matrix of shape [cells x features]) is uploaded
  + Convolution Layer:
    - CellCnn learns filters (like in image CNNs) that detect specific patterns of marker expressions across cells.
    - Each filter is applied individually to each cell, producing a response score per cell.
  + Pooling Layer:
    - Max pooling or mean pooling aggregates the cell-level filter responses into a sample-level score.
    - Max pooling is good for finding rare cell populations (if even one cell matches the pattern strongly, it’s enough).
  + Output Layer:
    - The sample-level scores are fed into a classifier (e.g., sigmoid layer) to predict the label (e.g., disease vs. control).
  + Training:
    - The model learns the filters and weights via backpropagation using standard optimization (like stochastic gradient descent).
    - It uses the labels at the sample level to learn what cell patterns are predictive.

DeepCyTOF

* Deep learning
* For automated gating
* gating based on a multi-autoencoder neural network
* Only requires labelled cells from a single sample
* classify cells into known cell types
* based on deep learning + domain adaptation, using a type of neural network called a stacked autoencoder
* How it works:
  + Training Phase:
    - You start with a reference sample (manually gated).
    - For this sample, you know the true labels of each cell.
    - A stacked autoencoder is trained to learn a compressed (latent) representation of the data that captures essential biological signals.
      * Autoencoder: neural network that learns to reconstruct its input
      * Latent representation a transformed version of your original input: Keeps the **most important information**, drops noise or redundant stuff, and often captures **biological structure** in the data (like cell types or activation states).
  + Domain Adaptation:
    - Each new (target) sample may have a slightly different distribution (due to staining variability, instrument drift, etc.).
    - DeepCyTOF aligns the feature representations between the reference sample and target samples.
    - This is done by training autoencoders on both reference and target data, and then using shared layers to bridge the gap.
  + Cell Classification:
    - Once a shared representation is learned, DeepCyTOF uses a softmax classifier on top to predict the cell type for each cell in the target samples.
    - Only one or a few manually gated samples are needed for training; the rest can be unlabeled.

Unsupervised Learning:

FlowSOM

* Clustering algorithm
* Great for classifying groups (i.e. clustering pixels of an image for face recognition)
* Uses stochastic method for calculation
* Bases the randomisation off a number called - seed
  + Seed can be set manually to reproduce results
* Recommended to set the number of clusters manually - since it can under cluster (missing smaller populations)
* Can apply dimensionality reduction technique either before or after flowSOM
  + FlowSOM applied to the UMAP components has the advantage of getting groups more separated
  + Disadvantage: dimensionality reduction techniques excludes redundancy → this may miss smaller populations
* First It maps the cells to a self-organized map (SOM) - artificial neural network used for clustering
  + Consists of a grid of nodes
  + When clustering - a new point is classified with the node that is nearest
  + Grid is trained in a way that nodes closely connected to each other resemble each other more than nodes that are only connected through a long path
  + unsupervised neural network used primarily for dimensionality reduction and data visualization
* Then it perform meta-clustering
  + After initial clustering, FlowSOM performs meta-clustering to group similar clusters into broader categories, aiding in the interpretation of related cell types.

FLOCK

* Grid based density clustering method for high dimensional flow cytometry data
* Does not require clusters to be pre-defined
* Also does not need manual gating
* How it works:
  + Data Partitioning: Each dimension of the dataset is divided into equal-sized bins, creating a multi-dimensional grid of "hyper-regions."
  + Density Identification: The algorithm identifies hyper-regions with high event density, which are likely to represent distinct cell populations.​
  + Region Merging: Adjacent dense hyper-regions are merged to form clusters.​
  + Centroid Calculation: Centroids of these clusters are computed, and each event is assigned to the nearest centroid based on distance metrics like Euclidean distance.​
  + This method allows FLOCK to determine the number of cell populations present in the data without prior knowledge, distinguishing it from algorithms that require predefined cluster numbers .​

Flowmerge

* FlowMerge uses Gaussian mixture models to identify cell subsets from the cytometry data