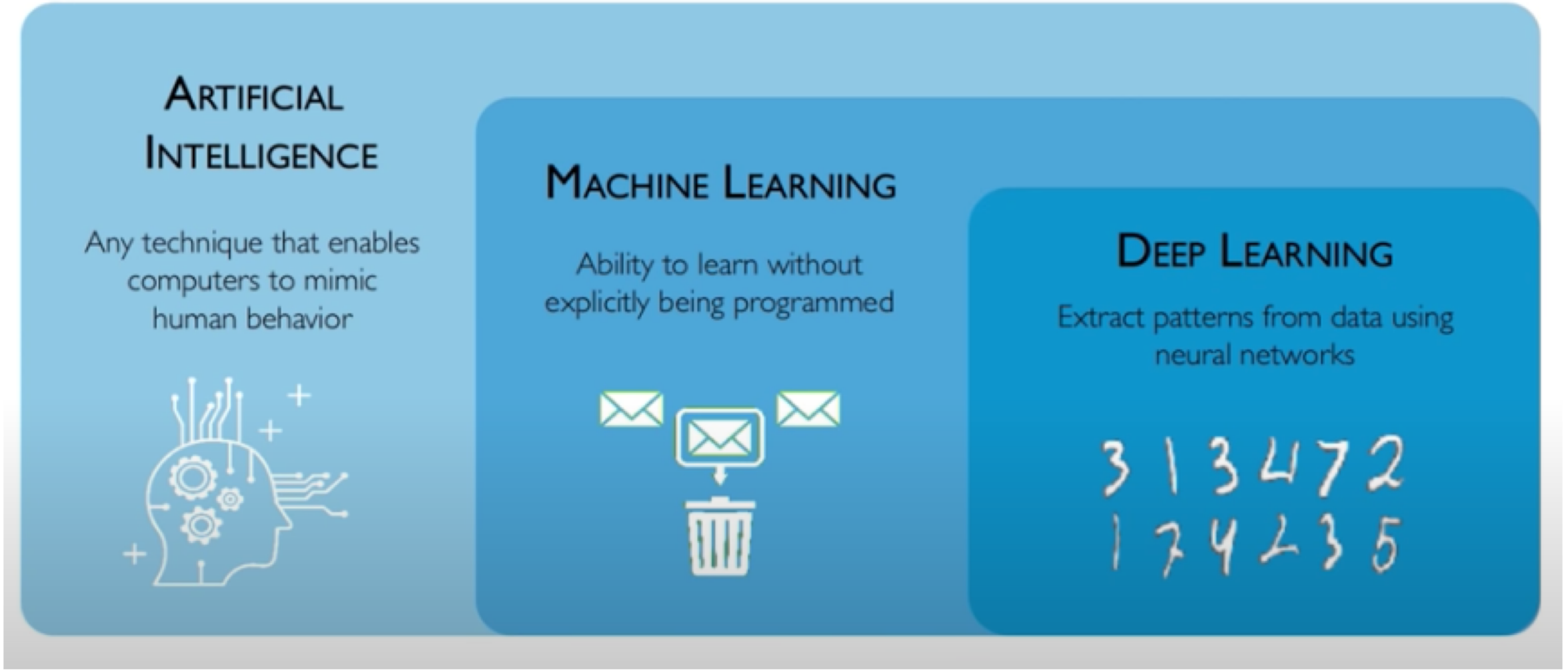
Markov Decision Process with Reinforcement Learning

Mingyu Yang, Ph.D.

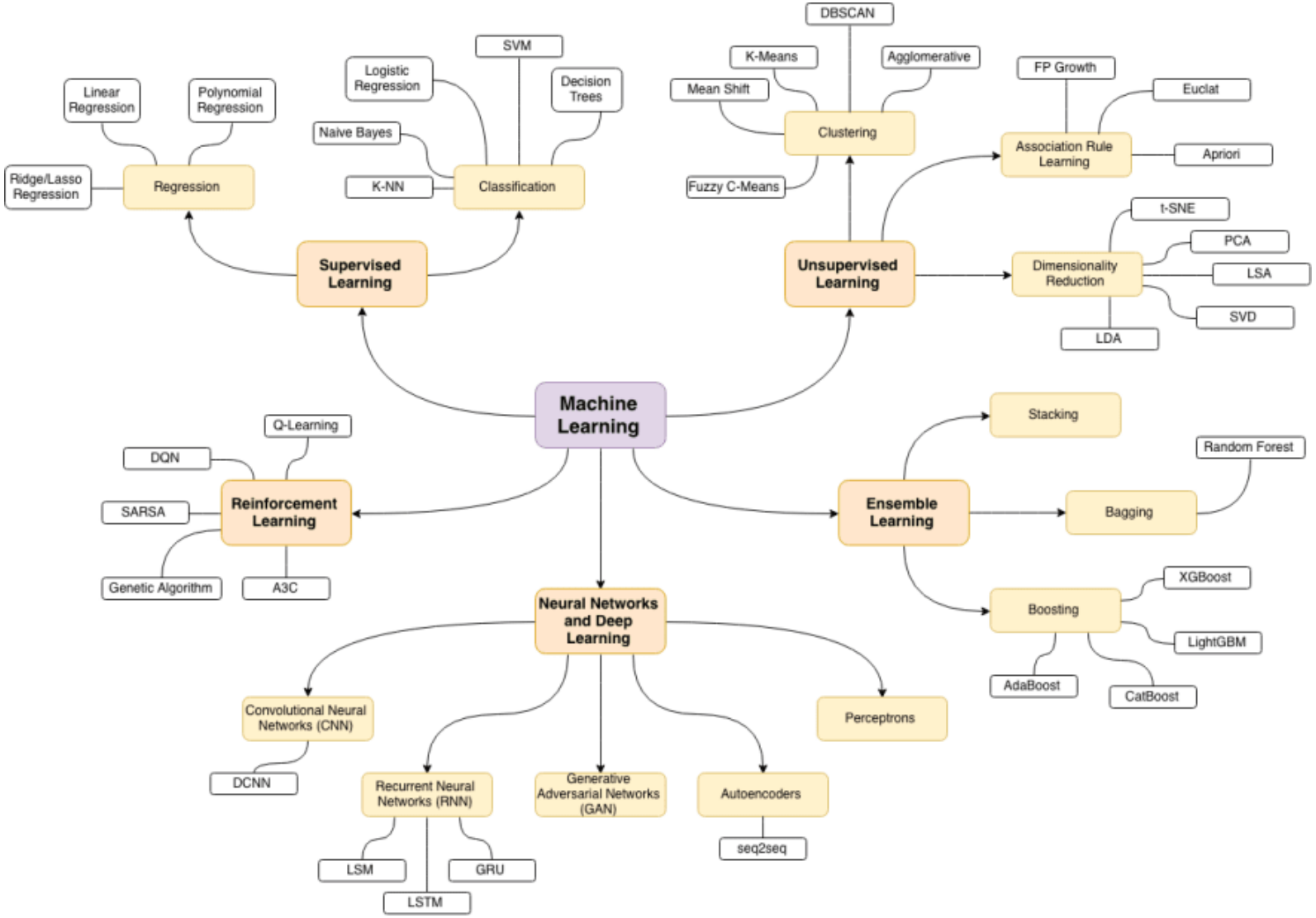
Fan Lab, BME

Yale University

AI/ML/DL

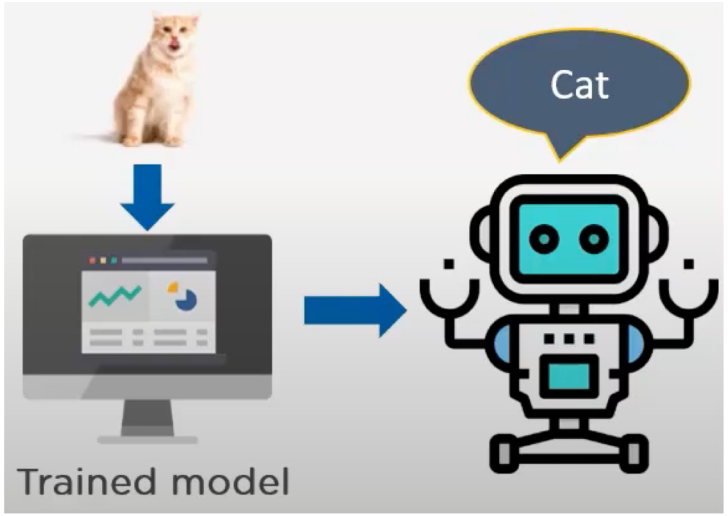

ARTIFICIAL INTELLIGENCE
Any technique that enables computers to mimic human behavior
MACHINE LEARNING
Ability to learn without explicitly being programmed
DEEP LEARNING
Extract patterns from data using neural networks

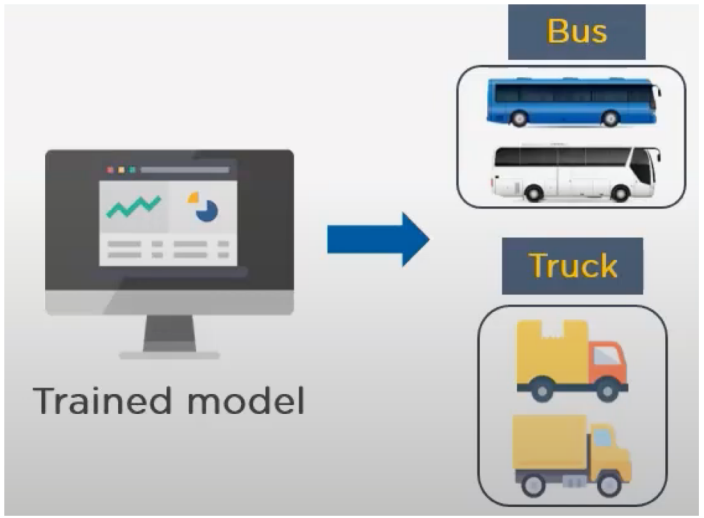
Types of Machine Learning

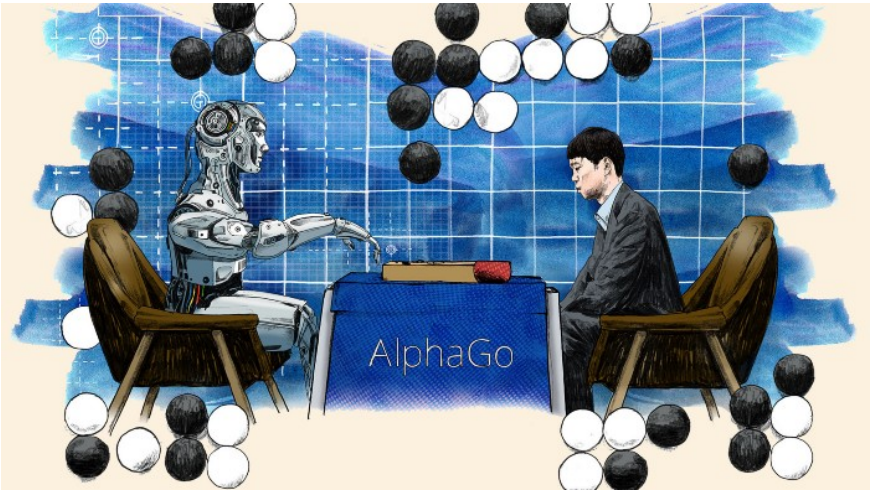


Supervised vs. Unsupervised vs. Reinforcement learning

- Supervised learning is used to train machines using labeled data
- Unsupervised learning uses unlabeled data to train machines
- Reinforcement learning uses an agent and an environment to produce actions and rewards.

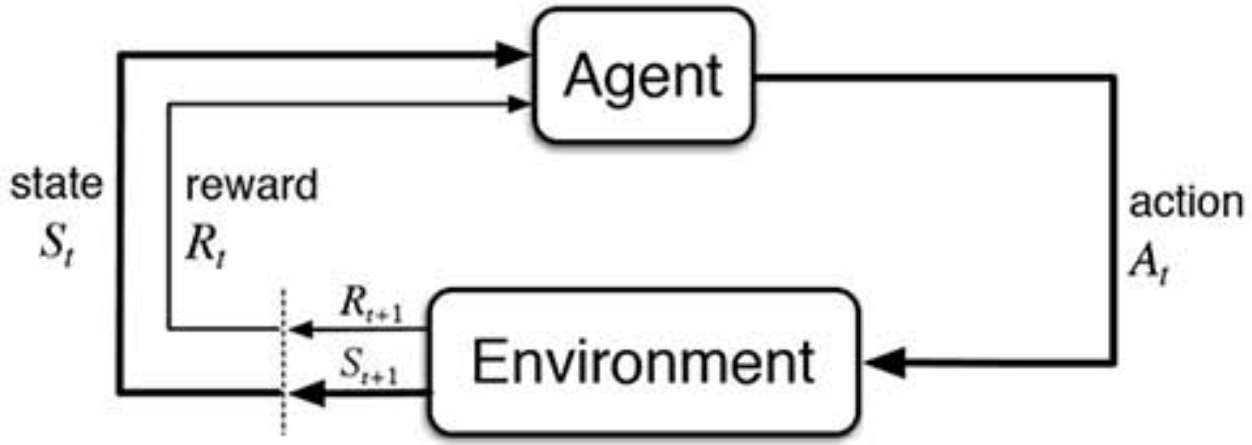


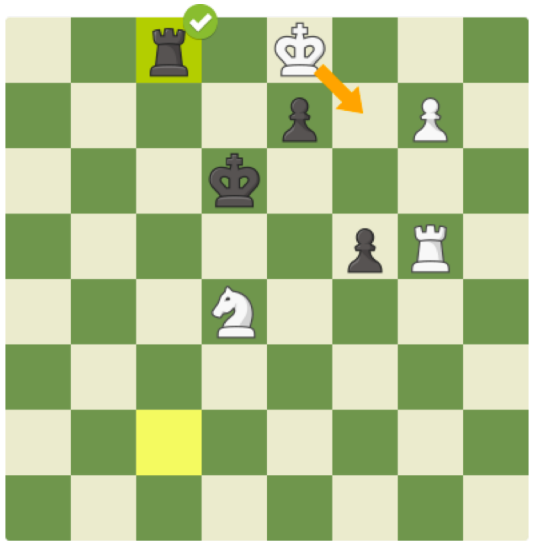




Overview of Reinforcement Learning

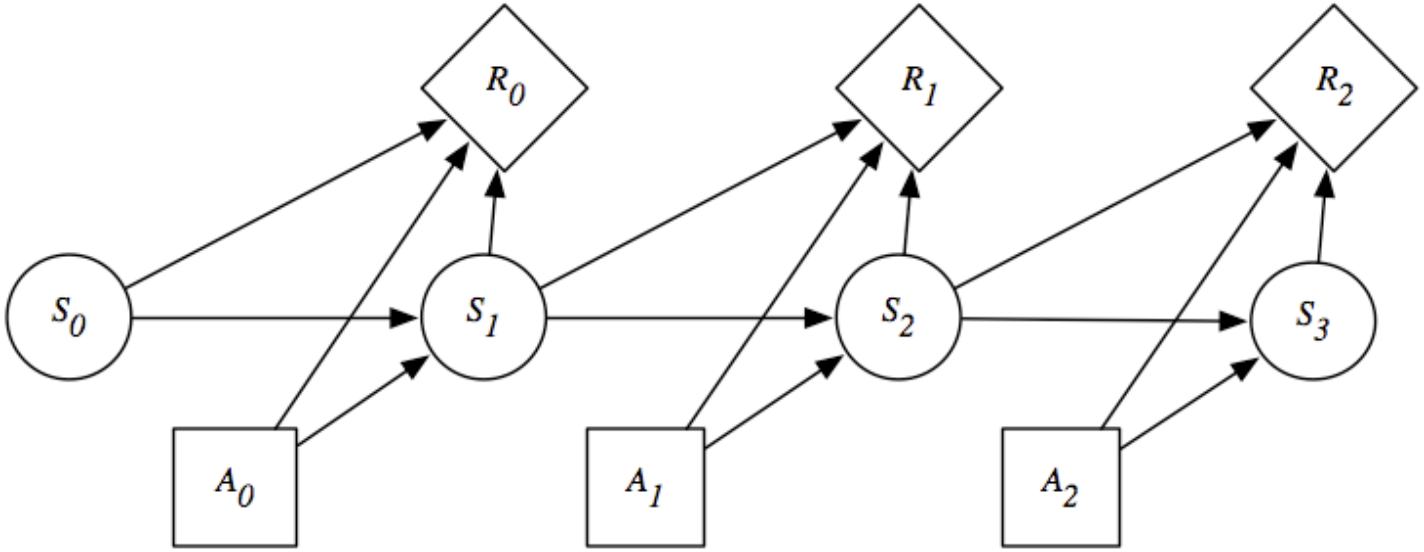
Agent
Environment/problem
States
Next states
Action
rewards





Markov Decision Process (MDP)

- An MDP is defined by
  - A set of states s∈S
  - A set of actions a∈A
  - A transition function T(s,a,s’)
    - Prob that a from s leads to s’
    - i.e., P(s'|s,a),
    - Also called the model
  - A reward function R(s, a, s’)



What is Markov about MDPs?

- Andrey Markov (1856-1922)

- "Markov" generally means that given the present state, the future and the past are independent

- For Markov decision processes, "Markov" means:

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \dots S_0 = s_0)$$

$$=$$

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

What is Q-matrix?

Policy

$$\pi \triangleq S \to A$$

Long-term expected reward

$$V_\pi(s) = \mathrm{E}[R] = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right],$$

$$V^*(s) = \max_\pi V^\pi(s).$$

Q-matrix : policy of optimization

$$Q^\pi(s,a) = \mathrm{E}[R \mid s, a, \pi],$$

$$Q^{\pi^*}(s, \quad \cdot) = \max_a E[R|s,a,\pi]$$

In Conclusion

- Reinforcement learning is improving an agent's performance against a problem based on expected reward
  - It is modeled as an MDP -> states, actions, state transition probability distributions, and rewards
  - The goal is to arrive at an optimal policy, where each state is mapped to an optimal action you can take to maximize your long-term reward

Methods for Genetic trajectory analysis

- for a given sample, first constructed a reward matrix by enumerating all possible clones given the number of mutations present in a sample.

- After construction of the reward matrix, set permissible decision processes with a value of 0, and impermissible decision processes with a value of −1 (that is, decisions where a mutant was reverted to wild-type or required more than one genetic alteration were penalized).

    - Decisions were considered permissible if a clone was separated by a single genetic event, either a variant changing from wild-type to heterozygous or heterozygous to homozygous.

Methods for Genetic trajectory analysis

- For observed clones, the frequency of the clone (ranging from 0 to 100% of cells) was used as the value in the reward matrix, while unobserved clones retained a value of 0.

- The matrix was then converted to long form and state transitions between clones were associated with the action/mutation causative to that state change. This was then used as input to the ReinforcmentLearning package in R to generate a Q matrix through the experience replay algorithm.

- Navigate this Q matrix to determine the optimal trajectory from the wild-type clone.

In this paper: Genetic trajectory analysis

- Agent : a given sample
- Environment : track of mutation acquisition (Clonal trajectory)
- Action : mutation causative
- Reward
  - permissible decision processes with a value of 0, and impermissible decision processes with a value of −1
  - For observed clones, the frequency of the clone (ranging from 0 to 100% of cells) was used as the value in the reward matrix, while unobserved clones retained a value of 0.

Reference

- MIT (http://introtodeeplearning.com/)
- Stanford (https://online.stanford.edu/artificial-intelligence/free-content)
- The Sutton book