Stackelberg Game for Vaccine Design

Palak Agarwal

September 22, 2015

Background

Stackelberg games involve a leader and a follower and the system is such that the leader ends up making a huge amount of profit as compared to the follower. When we are using this model for vaccine design, the virus acts as the follower (attacker) and the antibody (antibody designer) as the leader (defender).

[1] The key insight of Stackelberg games is that the defender (leader) needs to optimally account for attacker’s response to a defensive posture.

In the vaccination model of the Stackelberg game, the vaccine designer chooses an antibody with particular binding characteristics, to which the virus replies by choosing a minimum number of mutations to escape binding with the antibody. To check if a given virus-antibody pair binds or not, the authors use a software called Rosetta, to compute the binding score for an input virus-antibody pair.

Challenges

So, the basic goal is to produce antibodies that are capable of binding to a virus and many of its mutant strains at the same time. For an antibody to be effective against a virus strain, two things are necessary

• First, it should actually bind to the virus strain
• Second, the virus should not easily mutate or escape binding with the antibody

This model, however, poses the following challenges:

• search space is large for both the virus and the antibody (of the order of $10^{50}$ for both) due to the size of the binding sites
• To check whether a given antibody-virus pair binds or not through rosetta takes too much time and so it’s not feasible
• the escape cost for the virus (the number of mutations after which a given virus escapes the antibody) should be large $\geq 5$
Methodology

Since rosetta is not feasible due to it’s time-consuming behaviour, we switch to machine learning techniques to predict binding for a given antibody-virus pair.

Steps (given a virus V):

- Given a native antibody sequence, apply local search to get its useful mutations
- Machine learning to predict binding for a given antibody-virus pair
- Poisson regression to predict escape costs as a function of antibody sequence assignment

Motivation

Viruses like HIV and antibiotic resistant bacterial strains have no specific vaccine/drug treatment. Viruses causing these diseases can mutate rapidly, and thus avoid treatment. The goal is to discover broadly binding antibodies, that bind to many different known strains of the same virus.

Antibody Design

The vaccine designer is concerned with only this part. His job is to maximize number of escape mutations for the virus, i.e., the minimum number of mutations after which the virus can no longer bind to the antibody.

Best response of the virus to a given antibody 'a'

$$\min ||v_0 - v||$$

such that $$O(a, v) \geq \theta, \forall v \in V$$

Here, $$v_0$$ denote the native virus and $$v$$ and $$a$$ are arbitrary virus and antibody sequences respectively.

$$O(a, v)$$ represent binding energy for the antibody-virus pair $$(a, v)$$

$$\theta$$ is a threshold on binding energy which designates escape (that is, once binding energy is high enough, the virus and the antibody will no longer bind)

$$||v_0 v||$$ simply tells us the number of positions different in $$v$$ and $$v_0$$

Designer’s decision problem:

$$\max ||v_0 v(a)|| \text{ s.t. } a \in A$$

where A is the antibody design space.

It is noticed that the antibody-virus interaction in our model is a Stackelberg game which is zero-sum too since the designer wishes to maximize the number of escape mutations, a quantity which is minimized by the virus.

Computing minimal virus escape

We need to compute virus’s optimal solution to a given antibody chosen by the vaccine designer. We apply greedy local search algorithm, in which we start with a given pair and check if the Binding energy of the pair is greater than $$\theta$$.
If it is, then the escape cost is 0, otherwise all single point mutations of the given virus are considered and the one that has the highest binding energy is chosen and the whole process is repeated again with this mutation as the base virus. There is one disadvantage though, which is computing Binding energy of each mutation through Rosetta. Rosetta takes too much time, so they speed up their search by learning.

**Speeding up search through Learning**

Studying the local search tells us that there are a very few single mutations that cause a significant change in Binding energy. So, identify these mutations that matter and compute Binding energy only for them. They do this by training a classifier $\Omega(a, v)$ which takes two values 1 and -1. If it is 1, that means the single point mutation $v$ makes much difference in binding energy.

We have another classifier $\psi(a, v)$ which tells us whether the given pair binds or not (by the value +1)

**How the classifiers are trained?**

Training data is generated by collecting a set of local search runs for different antibodies. Feature vector consists of binary indicators which tell you whether a particular pair $(a, v)$ is different from the native $(a_0, v_0)$ sequence or not. To evaluate the value of $\Omega(a, v)$, the following formula based on the Binding scores of all the possible mutations

If $O(a, v) > M(a, v) + 0.9(g(a, v)M(a, v))$ Then, $\Omega(a, v)= +1$, else -1.

$v$ is the virus sequence for which $v$ is a single-point mutation, $M(a, v)$ is the median binding score of all single-point mutations from $v$ and $g(a, v)$ is the highest score among these. After applying machine learning techniques, Rosetta is used only for confirming, so the overall mechanism is more effective.

**Computing minimal virus escape**

Given a base antibody-virus pair, first evaluate Binding energy for this pair. If it is greater than $\theta$, then return 0.

Else, look at all the point mutations $w$ of the given virus, evaluate $\Omega(a, w)$ for each of them, and then evaluate only those for which $\Omega$ is 1.

Before evaluating their Binding energy, it is first verified if $\psi$ is -1.

If it is then, evaluate Binding Energy for this pair and check if it is greater than $\theta$. If it is, then, return 1.

**Antibody Design**

Local search for antibody design:

They start with a given pair of Antibody-virus and evaluate $\psi$ for it. If $\psi$ is 1, then, keep it. Else, look at some randomly generated mutations of the antibody and for which the escape costs are computed and then the one which has the highest escape cost and for which $\psi$ is 1 is taken.
Poisson regression
Predicting escape cost still takes some time, so to avoid this they use Poisson regression. It is used to predict escape time for a given antibody.

References