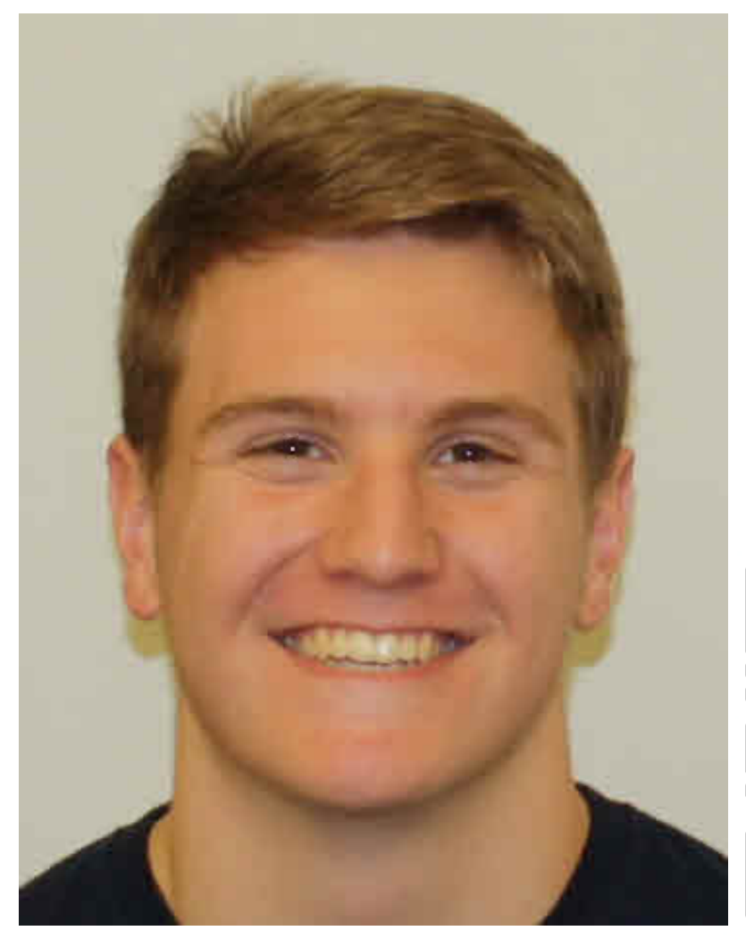
Predicting and Detecting Future Malware Variants

Using opcode prediction to generate malware indicators

Summer 2022

Team Members

Taylor McCampbell
- Senior, COSC Major
- INL Intern



Rafer Cooley
- COSC Ph.D. Student
- INL Intern

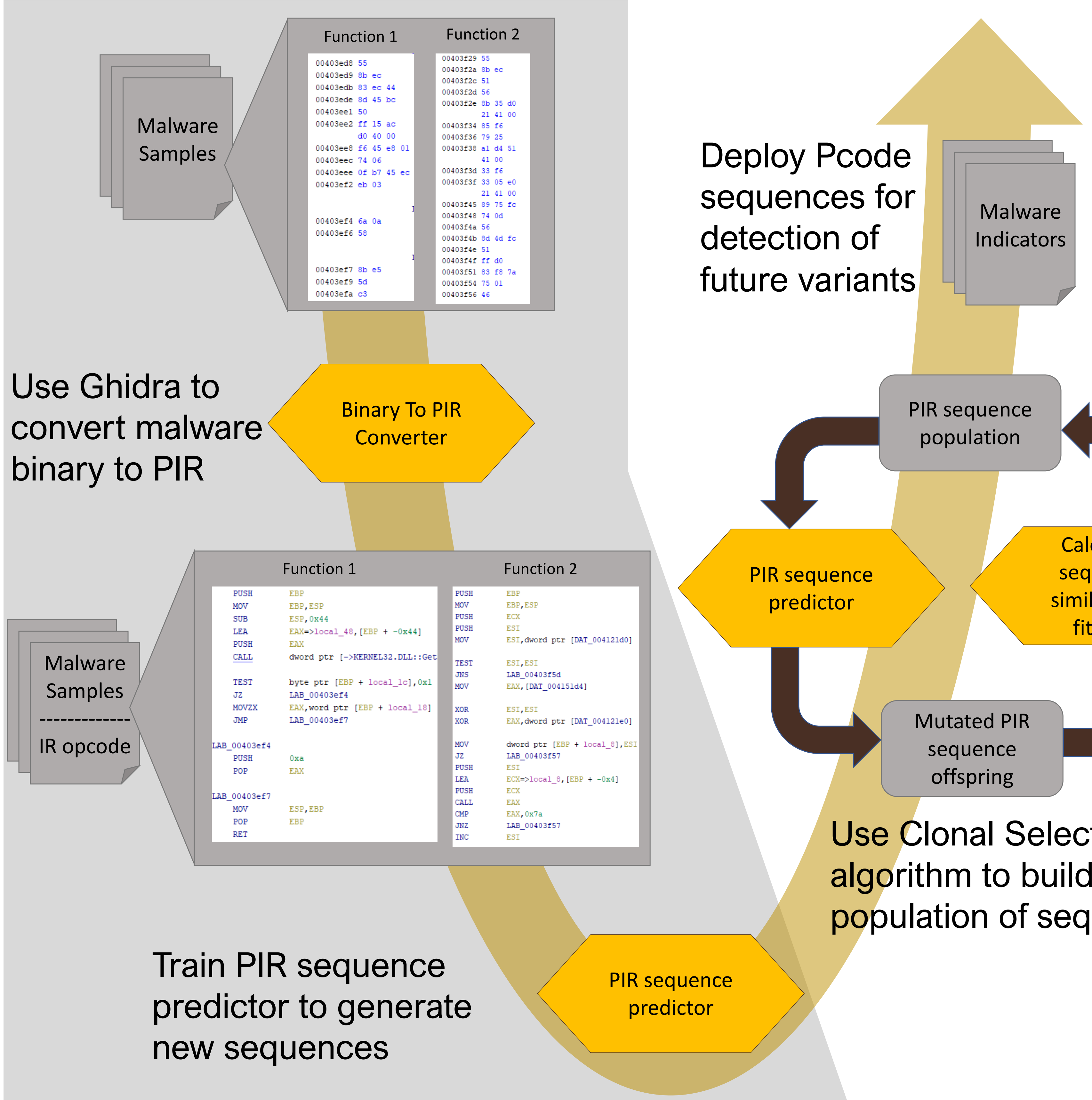
Background

- Biological immune systems (BIS) detect virus mutations by generating a library of DNA snippets
- Artificial Immune System (AIS) such as Clonal Selection Algorithm (CSA) mimic BIS by mutating and evolving a library of known-good and known-bad indicators for digital concepts
- Malware authors evade anti-virus by mutating instructions and control-flow paths
- Ghidra's Pcode Intermediate Representation (PIR) language abstracts platform/CPU specific instructions into a standard language of opcodes
- Machine Learning Generators (MLG) can predict future sequences of human languages
- Code Clone similarity methods (CCSM) can determine how similar two snippets of code are
- It may be possible to use MLG and CCSM functions in a CSA framework to generate PIR indicators resistant to malware evasion techniques

Problem Statement

Given the first **N** samples of a malware program (**N>0**), is it possible to generate a library of indicators that will successfully detect **X%** of future variants of the malware. Future variants of the malware are considered to be any variants produced after the first **N** samples.

Methods



Use Ghidra to convert malware binary to PIR

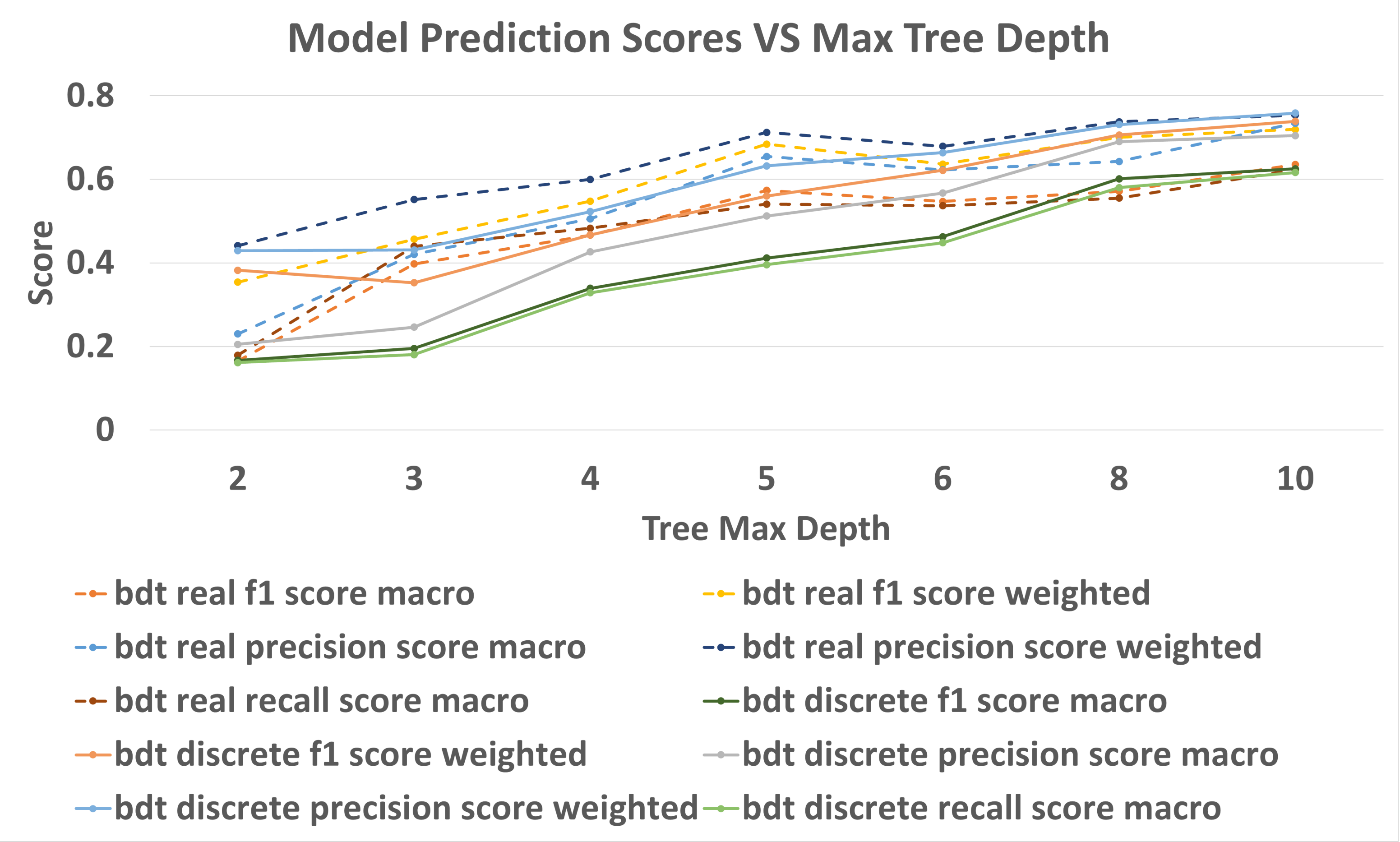
Train PIR sequence predictor to generate new sequences

Use Clonal Selection algorithm to build & train population of sequences

Deploy Pcode sequences for detection of future variants

Results

- Using Darkside, Revil, Wannacry, Trickboot example dataset
- Multi-Class AdaBoosted Decision Trees
- Predict the next instruction from window of 4



Challenges & Future Work

- Converting binary to Pcode is expensive
- Function sequences contains unnecessary data, data-flow graphs may prove better
- Implement code clone similarity methods for population fitness evaluation
- Use ML language modeling techniques to represent instruction sequences
- Target behaviors associated with categories of malware instead of specific families

Advisor: Dr. Mike Borowczak

Group Members:
- Rafer Cooley (rcooley2@uwyo.edu)
- Taylor McCampbell (tmccampb@uwyo.edu)