**Outcome:** Designed 3 novel toehold switches that can detect concentrations of blood-based microRNA biomarkers and implemented in a paper-based diagnostic test for colorectal cancer

**WEEK 1**

* Identified the key features of a disease I wanted to develop a paper-based diagnostic test for. These include:
  + High-impact (prevalent among the general population)
  + Panel of biomarkers had been found, but methods to develop assay not developed
  + Panel contains mostly/only biomarkers specific to the disease eg not like IL-6 or elF4E which can be present in multiple diseases
* [Literature review.](https://docs.google.com/document/d/1oMVdYzQguRrm8vz9fexezjuHqYSsG4skcmT9WtTjfF0/edit) Decided on colorectal cancer because:
  + high disease burden (third most common form of cancer in the world
  + Severe symptoms
  + No existing paper-based diagnostic devices, current gold-standard is invasive colonoscopy
  + High 5-year survival rate when detected in early stages (91% if cancer is localized, 72% when metastasized)
* Identified panel from Huang et al, 2020. Chose this due to small panel size (4 miRNAs) and high AUC.
* Found miRNA database (miR-base) to find sequences of selected miRNAs in the panel

**WEEK 2**

* From my previous research project under Dr Patthara Kongsuphol at Singapore-MIT Alliance of Research and Technology, I had briefly learned about toehold switches as a molecular sensor to detect nucleic acids (Wang et al, 2019). Connected this idea with miRNA detection in this project and dove deeper into how toehold switches can be used in paper-based diagnostic devices. (Pardee et al 2016)
* Completed set up of required Python software environment that can run Toeholder, Nupack and Blast+ code without any technical runtime bugs.
* Tested the Toeholder code by feeding it with target gene data in FASTA format and reference genomes in TXT format. Got the expected toeholder switch nucleotide sequences (in base pairs and in DU+ format), in addition to other thermodynamic data such as mfe (aka binding energy).
* Read the papers and user guides for the Toeholder and Nupack softwares

Challenges:

* As Nupack and all related code only runs in the Python environment on Unix, I learnt basic Unix environments and commands from the ground up for the first few days. I also got familiar with working with GitHub and Nupack code repositories, Toeholder and Blast+.
* I needed to understand how the Ubuntu unix server, Jupiter Lab Python Server, Nupack module, Blast+ module and Toeholder module work together in order to achieve the desired outcomes.
* There was no version of NUPACK available for Windows OS. I found 2 potential solutions: installing Cygwin and using the Windows Subsystem for Linux 2. I found that setting up the Windows Subsystem for Linux to be the most time and space efficient solution. I then installed the Ubuntu unix server.

**WEEK 3 & 4**

* Identified key toeholder limitation: target gene sequence input must have length between 35 and 100bp.
  + Specific corresponding DNA sequences for miR- 203a-3p, miR-200c-3p and miR-145-5p were too short: used full genes miR-203a, miR-200c and miR-145 from BLAST+ database
  + DNA sequence for miR-375-3p is 64bp long, this target sequence was maintained.

Challenges:

* Identified factors used to evaluate the viability of the sensors generated by NUPACK for paper-based tests. Toeholder/NUPACK only checks free-binding energy. Needed a way to triangulate results and use other metrics to evaluate viability.
  + I learnt about GC content/percentage check GC and pair probability graphs as other factors to evaluate viability from comprehensive literature review. I implemented new Toeholder code to display results for GC content and pair probability for each toeholder switch generated, referencing Nupack documentation + stackexchange
* Debugging issues: I realised Nupack wasn’t running against human reference genome from BLAST due to 0 matches for mir-375 on Chromosome 2
  + I resolved this issue by checking file reading from input\_seq (with Cathy’s help). I learnt how to correctly implement FASTA file reading on Ubuntu server.
* For each switch, Toeholder displays concentrations of switch, target and switch-target complexes. Concentrations of miR-145 switch-target complex formed were very low.
  + From my previous experience with optimization projects under Dr Patthara Kongsuphol and Dr Tay Chuan Beng, I identified low concentrations of the switch as a potential problem. However, after testing Toeholder to see if increasing concentration from 1 to 1000uM of switch could increase complex formation, binding percentage remained very low.
  + Concluded that low binding percentage implied low signal:noise ratio, making miR-145 switch unsuitable for paper-based diagnostic tests.