**Component Specification for pcr\_optimizer**

Lionel Tukei and Lily Torp

1. **Software components:**
   1. **Class:**

* PCR(gene: str, forward\_primer: str, reverse\_primer: str, template\_type: str)
  + Generates a PCR object for storing and optimizing PCR
  + Inputs: gene sequence, forward primer sequence, and reverse primer sequences in 5’-3’ format. Template type either (plasmid, lambda, BAC DNA, or genomic)
  + Outputs: None

1. **Functions:**

* check(self: pcr object, startr: int, stopr: int, startf: int, stopf: int)
  + Checks gene, forward, primer, and reverse primer for errors (non-base characters) and checks gene-primer complementarity (primer will anneal to beginning of gene)
  + Inputs: pcr object (contains gene, forward primer, and reverse primer)
  + Outputs: A print statement describing any issues.

* countGCcontent(self: pcr object)
  + Calculutes % G and C bases in a sequence
  + Inputs: pcr object (gene, forward primer, and reverse primer)
  + Outputs: A print statement reporting GC content as a percent

* recommend(self: pcr object, factor: str)
  + Evaluates PCR time, annealing temperature, enzyme amount, and cost per reaction for the iProofTM High Fidelity DNA polymerase and Taq Polymerase.
  + Inputs: pcr object (gene, forward primer, reverse primer, and template type) and factor (None, “time”, or “cost”).
  + Outputs: A table of conditions (amount, cost, or both).

* check\_gene(self: pcr object) \*\*Specification is similar for check\_fp and check\_rp\*\*
  + Checks gene sequence for non-base characters (base characters = “a”, “t”, “g”, “c”.
  + Inputs: pcr object (gene, forward primer, reverse primer)
  + Outputs: A print statement stating the gene is good or there is an unacceptable character.

* checkPrimerGeneCompatability(gene: str, forward\_primer: str, reverse\_primer: str, startr: int, stopr: int, startf: int, stopf: int):
  + Checks whether primer sequences anneal in the correct positions at the start and end of the gene.
  + Inputs: gene: target gene sequence, forward\_primer: primer for amplifying gene top strand, reverse\_primer: primer for amplifying gene bottom strand, startr: reverse primer binding location (5’), stopr: reverse primer binding location (3’), startf: forward primer binding location (5’), stopf: forward primer binding location (3’).  Ask Lionel abou the stop locations

* iProofAnalyzer(self: pcr object)
  + Uses protocol from Bio-Rad, Inc to determine enzyme amount, cost per reaction, annealing temperature, annealing time, extension time, and total PCR reaction time.
  + Inputs: pcr object (gene, forward primer, reverse primer, and template type)
  + Outputs: Two tables, one with enzyme information and one with temperature/time details for iProof High-fidelity DNA polymerase.

* TaqAnalyzer(self: pcr object)
  + Uses protocol from New England Biolabs, Inc (NEB) to determine enzyme amount, cost per reaction, annealing temperature, annealing time, extension time, and total PCR reaction time
  + Inputs: pcr object (gene, forward primer, reverse primer, and template type)
  + Outputs: Two tables, one with enzyme information and one with temperature/time details for Taq DNA polymerase.

1. **Interactions to accomplish use cases:**

* Use case: optimizing PCR protocol for time
* In this use case, the PCR object stores the user input. check() ensures that the user inputs are formatted correctly. It will call check\_gene(), check\_fp(), and check\_rp() and checkPrimerGeneCompatability() to check the sequences for non-base characters and make sure the primers are annealing in the correct location. recommend() will call iProofAnalyzer() and TaqAnalyzer(). Depending on user factor definition in recommend, the function will return a table of enzyme amount and cost per reaction, or annealing time, extension time, and total PCR reaction time.

1. **Preliminary Plan:**
   1. Define PCR object
   2. Run check() function

* Fix any errors as needed
  1. Run recommend() function
  2. Analyze table output