Multiple sequence primer design tool

Programming for Biology Project

10/23/25

Objective

The objective of this project is to design primer pairs that specifically includes or excludes mismatches across a group of sequences, e.g. (same gene different species, same gene, different isoforms, different genes in the same family). An example where we need to distinguish between different species in the same sample – where human pluripotent genes to induce pluripotent cells in species X, humanized models. On the other hand, we may want to capture all members of a gene family, all gene isoforms, or all copies of a gene.

This Python tool would allow users to systematically parse through sequences to generate primer pairs that includes or excludes mismatches in a group of sequences.

Method

A screenshot of a computer

AI-generated content may be incorrect.

Test data set

Goal: Design primer pairs to detect all or a specific copy of human amylase copies: NM\_001008221.1, NM\_000699.4, NM\_020978.4, NM001386925.1, NM\_001008218.2.

Similar tools

PrimeSpecPCR: Python toolkit for species-specific DNA primer design and specificity testing

IsoPrimer: a pipeline for designing isoform-aware primer pairs for comprehensive gene expression quantification

Calculate complementarity

Self-complementarity: To calculate the self-complementarity of a single primer, compare the primer sequence with its reverse complement.

Pair complementarity: For a primer pair, compare the 5’ to 3’ sequence of the forward primer with the 3’ to 5’ sequence of the reverse primer to find complementarity. Complementarity at the 3’ends can lead to primer-dimer formation and non-specific amplification. Should have minimal complementarity at the 3’ends. The complementarity can be based on number of matches bases, whereby the mismatches at the 3’ end are more critical.

🡪 Lower complementarity score is better, i.e., the primer is not binding well to unintended sequences.

PCR amplicon length

Standard PCR: 200-1000bp; real-time PCR and ddPCR: 50 to 200bp.

1. Retrieve Genbank records. Genbank records are well annotated, including id, description, sequences, features, etc. For example, we can use GenBank's annotated gene structure to place primers on exon-exon boundaries (feature selection).
   1. EDirect, a command-line toolkit for accessing NCBI Entrez database from Unix.
2. Multiple sequence alignment of a group of similar sequences, e.g., same gene in difference species or multiple copies of the same gene.
   1. MUSCLE, a program written by Robert C. Edgar in 2004. Main advantage is that the alignment program is very fast.
3. Generate a comprehensive list of primers in both directions.
   1. k-mer length (default is 20-22 nt)
   2. Remove any repetitive k-mers per sequence
   3. Thermodynamic properties of individual primer: melting temperature, CG content, 3’ CG clamp, self-complementarity, repeating nucleotides
4. Generate a class of primers for downstream steps. Attributes for the primer class includes sequence, start position (target index), end position (target index), melting temperature, strand directionality, sequence index.
5. Filtering primers.
   1. Identify all primers that have identical k-mers/primers across all sequences per group. These are potentially suitable for amplifying all sequences within the group
   2. Identify all primers that are unique within the group of sequences.
6. Filtering unique primers for a target sequence.
   1. Implement a counter for the aligned sequence. The aligned sequences in the group have the same length but have gaps in different positions for the different sequences.
   2. Implement a counter for the target sequence and match with the aligned sequence.
   3. Identify the matching target index for multiple sequence alignment (MSA) index with unique nucleotide or gap within the group of sequences.
7. Check the

What did we learn from our experience?

PrimerObject: we created a class called primer objects that stores information including sequences, position, melting temperature, etc.

ArgumentParser: we allow the user to input information to specify preference, e.g. catch All or catch RefSeq Accession numbers.

What was challenging about this project?

Continuous decision-making steps. There is no standard approach, and the workflow is very subjective.

What was unexpected?

Tips for future students:

Understanding data frames is critical, i.e., think about how to retrieve the information.

Determining the sequence of steps is important, i.e., where to insert code within the script.