ArrayOligoSelector

Last updated August 5, 2003

Jingchun Zhu, Zbynek Bozdech, Joe DeRisi, DeRisi Lab, UCSF

The complete genomic sequences of an increasing number of organisms are becoming available. To exploit these new resources, we have developed a program, ArrayOligoSelector, to systematically design gene specific long oligonucleotide probes for entire genomes, for the purpose of developing whole genome microarrays. For each open reading frame, the program optimizes the oligo selection based upon several parameters, including uniqueness in the genome, sequence complexity, lack of self-binding, GC content and proximity to the 3'end of the gene. We have used this program to generate oligonucleotides for the genome of *Plasmodium falciparum*.

download program here

Program work flow:

ArrayOligoSelector includes two sub-programs that run in series. The first sub program is the "computation program" which calculates scores of uniqueness, sequence complexity, lack of self-binding and GC content for every candidate oligo. The detailed scoring schemas are as the following:

- **Uniqueness** score is calculated as the theoretical binding energy of a candidate oligo to its most homologous genome sequence. BLASTN was used to locate the most homologous sequence followed by a calculation of the theoretical binding energy. The binding energy was calculated using a nearest-neighbor model with the established thermodynamic parameters. (Peritz, Kierzek et al. 1991; Sugimoto, Nakano et al. 1996; Allawi and SantaLucia 1997; Allawi and SantaLucia 1998; Allawi and SantaLucia 1998; Allawi and SantaLucia 1998; Allawi and SantaLucia 1998; Lyngso, Zuker et al. 1999; Peyret, Seneviratne et al. 1999).
- **Sequence complexity** score is calculated as the difference in bytes between the oligo sequence and the its compressed version by the LZW compression algorithm (Ziv and Lempel 1977)
- **Self-annealing** score is a measurement of the secondary structure generated by the self annealing of an oligo, calculated as the alignment score of the optimum local alignment between the oligo sequence and its reverse compliment using the Smith-Waterman algorithm (Smith and Waterman 1981).
- **GC content** score is calculated as the GC percentage of the oligo.

Scores generated by the first sub program are stored in a series of output files, which are used by the second sub-program, the "selection program" to select oligos are unique for the gene, with low level of internal repeat and self-annealing tendency and within a narrow range from the target GC percentage the user specified. ArrayOligoSelector chooses an optimal set of ranked oligos by the following means. The uniqueness-filter allows all oligos to pass which satisfy one of two criteria: a user defined binding energy threshold, or alternatively, the top 5% of candidate oligos within 5 kcal/mol of the candidate with the best (least stable) binding energy. In parallel, the sequence complexity filter and the self-binding filter will allow a given oligo to pass if it falls below the 33rd percentile of scores for the target open reading frame. The set of sequences emerging from both filters as well as the uniqueness filter are

compared. If there exist one or more sequences from the intersection of these three filters, the sequence is then allowed to pass onto final selection. If an intersection does not exist, the self-binding and complexity filters are incrementally relaxed until an intersection becomes available. Candidate oligos present in the intersection set are subjected to the %GC filter. Initially, oligos are allowed to pass if they meet the user specified %GC. If no oligos pass, the target %GC range is relaxed by one percentage point in each direction until one or more oligos pass. As a final step, all final candidate oligos are ranked by their proximity to the 3' end of the gene and the optimum oligos are selected.

Parameters can be manipulated by user

- Target GC percentage
- Number of oligos per gene
- Uniqueness score cutoff, measured as binding energy (if not specified, using default)
- Masking length, symbol and tolerance (optional)

What does the user need to provide?

Users need to provide sequence files of both input sequences for designing oligos and the complete genome. Both should be DNA sequences in FASTA format. The complete genome sequences could be either the complete set of the genes in the genome (exons or ORFs), or the complete genomic sequences that include exons, introns and intergenic regions. Two different versions of the programs are provided in the ArrayOligoSelector for either scenario. (Please refer to the **RUN** section to find out which program you should use). In the case of partial genomes, ArrayOligoSelector will find the unique oligos for the incomplete genomes. Users should bear in mind that the oligos might have similar sequences in the rest part of the genome.

Platform: Linux. The program has been tested on Redhat linux 6.1, 6.2, 8.0 and 9.0.

Python Interpreter: From ArrayOligoSelector version 3.2 or above, python interpreter version 2.2 or above is **REQUIRED**. For previous versions, it is still recommended but not required. You can download the interpreter from http://www.python.org .

RUN

- **First sub-program (computation program):**
  Pick70_script1 (the exon version) and **Pick70_script1_contig** (the contig version) are the two different versions of the first sub-program of ArrayOligoSelector. **Pick70_script1** should be used when the "genome file" is the gene sequences (exons or ORFs). On the other hand, **Pick70_script1_contig** should be used when the genomic sequences (exons, introns and the intergenic sequences) are used as the "genome file".

  To run the programs, typing "**./Pick70_script1**" or "**./Pick70_script1_contig**" on the command line in the program's root directory, and the command line usage will be printed on the screen. Three command line arguments are required. They are the filenames of the input and the genome sequence files and the length of the oligo (eg. 70). The first sub-program writes the results on disk as a series of output files called "output0, 1, 2, ... ". Two test files are provided with the release: "test_input" and "test_genome". The following are examples of the usage: **./Pick70_script1 test_input test_genome 70** and **./Pick70_script1_contig test_input test_genome**.

Since version **3.2**, the contig version of the sub-program is changed in regard to finding the cognate genomic locations of the input sequences. The program used BLAST in the pre 3.2 versions to identify the genomic origins of each input sequence. In the version 3.2 or later, the user can choose to use BLAST, BLAT or "gfclient" to do that. Both BLAT and gfclient are Blast-like alignment tools ideal for fast aligning exons to the genomes (Kent 2002). ArrayoligoSelector runs much faster if either Blat or gfclient are used. While Blat and gfclient are essentially the same, gfclient requires setting up the gfServer in advance and Blat calls for more memory.

The **post 3.2** contig version of the first sub-program requires an additional (fourth) command line argument to specify the method to identify self locations in the genome. The fourth argument takes a constant string of "blast", "blat" or "gfclient". Here is an example of the usage: /Pick70_script1_contig test_input test_genome 70 blat .

- **Second sub program (selection program):**
  Pick70_script2 is the second sub-program the user should invoke after the first sub-program finishes. To run the program, users can type : **./Pick70_script2** and the usage instructions will be printed on the screen. Three command line inputs are required: the target GC percentage (eg. 30.5 for 30.5%), the length (eg. 70) of the oligo, and the number of oligos per gene (eg. 1). There are also four optional inputs: the user defined uniqueness score cutoff (eg. -35 for binding energy -35kcal/mol), the bases, length and tolerant level of the masking sequences (eg. AT, 20 ,0.1). If the user does not provide the energy cutoff, the default cutoff will be used, which is defined as the top 5% of and within 5 kcal/mol from the best (least stable) binding energy. The masking sequences can be used to exclude stretches of sequences with only certain nucleotide compositions, which are defined as the compositions, the length, and the tolerant level. For example, if the user wants to exclude sequences with longer than 20bp AT and less than 10% of non-AT nucleotides, the inputs should be AT, 20, 0.1. To test the second sub-program, users can type:**./Pick70_script2 28 70 1** or **./Pick70_script2 28 70 1 -30 20 AT 0.1** .

Output

- "oligo_fasta" file: This is the file that has the complete set of oligo sequences in FASTA format. The identifiers of the oligos are in the format of the concatenation of the input sequence identifier and the starting position of the oligo by an underscore, such as GeneId_StartPositionOfOligo.
- "oligo_dup" file: This is the file that has information on an oligo's genome target, gc percentage, self-binding score and sequence complexity score. The detailed format of this file is as the following:
  >oligo_id GC internal-repeat-score self-annealing-score primary-target-gene-id binding-energy-with-primary-target location-of-the-primary-target secondary-target-gene-id binding-energy-with-secondary-target location-of-the-secondary-target ... ...
- "nodesign" file: If for any reason, ArrayOligoSelector fails to select an oligo for a gene, the gene's identifier will be stored in this file.

Speed

The program takes 12 hours to design gene specific 70mer oligos for 12MB Plasmodium falciparum coding sequences on a dual cpu 700MHz linux computer.

Reference

- Allawi, H. T. and J. SantaLucia, Jr. (1997). “Thermodynamics and NMR of internal G.T mismatches in DNA.” Biochemistry 36(34): 10581-94
- Allawi, H. T. and J. SantaLucia, Jr. (1998). “Nearest neighbor thermodynamic parameters for internal G.A mismatches in DNA.” Biochemistry 37(8): 2170-9.
- Allawi, H. T. and J. SantaLucia, Jr. (1998). “Nearest-neighbor thermodynamics of internal A.C mismatches in DNA: sequence dependence and pH effects.” Biochemistry 37(26): 9435-44.
- Allawi, H. T. and J. SantaLucia, Jr. (1998). “NMR solution structure of a DNA dodecamer containing single G*T mismatches.” Nucleic Acids Res 26(21): 4925-34.
- Allawi, H. T. and J. SantaLucia, Jr. (1998). “Thermodynamics of internal C.T mismatches in DNA.” Nucleic Acids Res 26(11): 2694-701.
- Kent, W.J., BLAT--the BLAST-like alignment tool. Genome Res, 2002. 12(4): p. 656-64.
- Peritz, A. E., R. Kierzek, et al. (1991). “Thermodynamic study of internal loops in oligoribonucleotides: symmetric loops are more stable than asymmetric loops.” Biochemistry 30(26): 6428-36.
- Sugimoto, N., S. Nakano, et al. (1996). “Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes.” Nucleic Acids Res 24(22): 4501-5.
- Smith, T. F. and M. S. Waterman (1981). “Identification of common molecular subsequences.”J Mol Bio 147(1): 195-7.
- Ziv, J. and A. Lempel (1977). “A Universal Algorithm for Sequential Data Compression.” IEEE Trans. Information Theory 23(3): 337-343

contact info: jzhu@itsa.ucsf.edu

The project is hosted at