**AUTOPRIMER: A GUIDE**

Table of Contents

**Introduction2**

Filetypes and general usage2

**Getting/Parsing Primers3**

Input files and format3

Generating Primers5

**BLASTING Primers6**

Input file and format6

Generating BLAST report8

**Pooling Primers9**

**Introduction**

Autoprimer is an application developed to improve efficiency and automate the workflow of designing primers for experiments in PCR. Currently, there is no tool that combines all elements required of designing primers from the design itself, to the specificity and pooling checks required to ensure good a good experiment.

Current design methods require a user to separately design primers on one tool or website, transfer that information to another tool and so on. With this tool, all of that can be achieved with just the click of a few buttons!

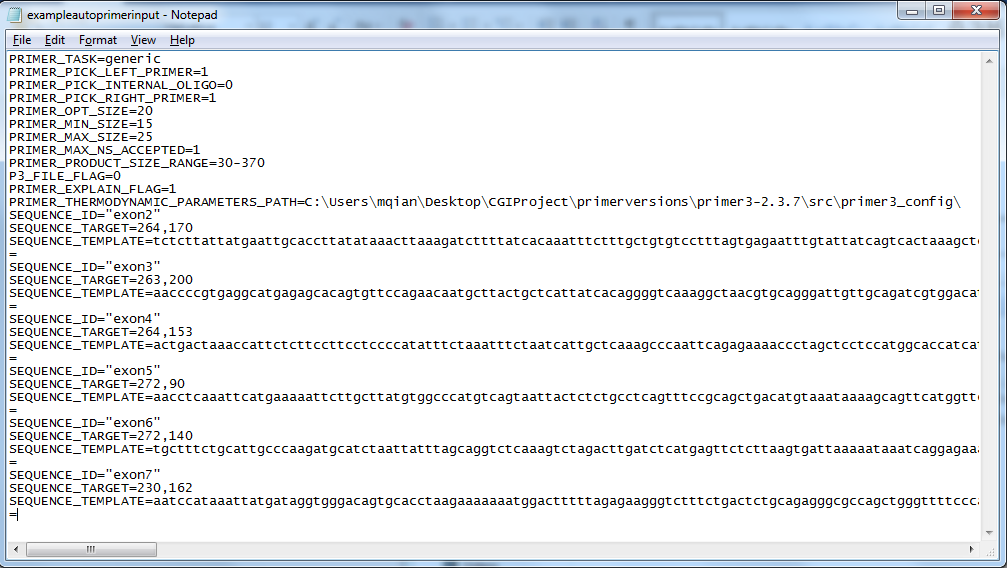
Autoprimer is designed for compatibility with Windows and Unix based operating systems. Both distributions will be available for use.

**File types and general usage:**

Autoprimer deals with several different file formats which will be explained throughout this guide. Since it is a wrapper program built on top of other programs, it uses the file formats required by those programs. The Primer3 (getting primers) portion of the program requires a text file format which will be provided in an example file. It has various sections that can be filled in and changed by the user depending on their needs with each section being terminated by an ‘=’ sign. The output given is in a somewhat similar format. Thus it is necessary to parse this output into a FASTA readable format, which the program can also do. The BLAST portion of the program accepts a FASTA format text file and reads it in to generate a BLAST XML report. This information is also difficult for a human to read, so the program will also parse this xml file and produce a more user friendly readable format. Finally, when the user decides what final primers to use, they can open up the primer pooler tool to help them decide what pools to put the primers in for multiplexing options should the user desire so. This program also accepts FASTA input.

**Getting/Parsing Primers:**

This portion of the guide will deal with how to use the program to give you primers based on your design criteria. It uses the popular Primer3 program to do the heavy lifting.

**Input files and formatting:**

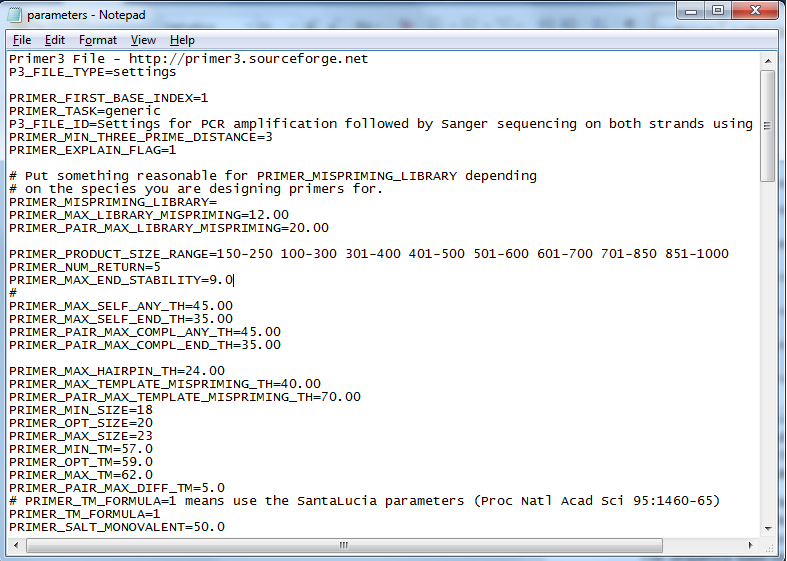
The format of the input file for getting primers is as shown above. There are many different parameters that must be filled in set by the user. The exact meaning of each line can be found on the Primer3 manual online: <http://primer3.sourceforge.net/primer3_manual.htm>

The user can define how large their primers should be, what specific task the program will run, and can insert multiple different sequence templates for the program to find primers from. It is important to note here the method for properly putting a template sequence into the input file. The sequence itself must be the target region plus extra bases flanking either side of the target. Adding the flanking regions allows for the program to pick possible primers that will produce a product that encompasses the target properly. One also needs to note the numbers next to the “SEQUENCE\_TARGET=” section. In this section, the left number denotes what character position in the sequence you copy and paste the target region starts. The right number denotes how long that target is. It is essentially telling the program where the target region starts and how long it goes for in the sequence template.

**NOTE:** The section ‘PRIMER\_THERMODYNAMIC\_PARAMETERS\_PATH=’ should be filled in with the correct filepath on your machine. It should have the same style as the example (primer3-2.3.7/src/primer3\_config/) regardless of where in your machine the overall folder is placed.

After finishing up with the input file, the user can choose to include an optional parameter file. This is where the user can choose to tweak settings common to a PCR experiment and also change parameters and constraints for the program to design around. Examples might include ideal melting and annealing temperatures, GC%, thermodynamic considerations, etc.

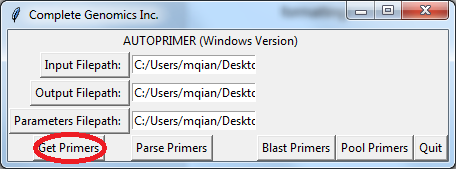
**NOTE:** Some settings may overlap in the input file and parameter file. The program’s priority is to deal with the input file first, so any parameters that are written in the input file and parameters file will only be processed from the input. This is stated in the program documentation.

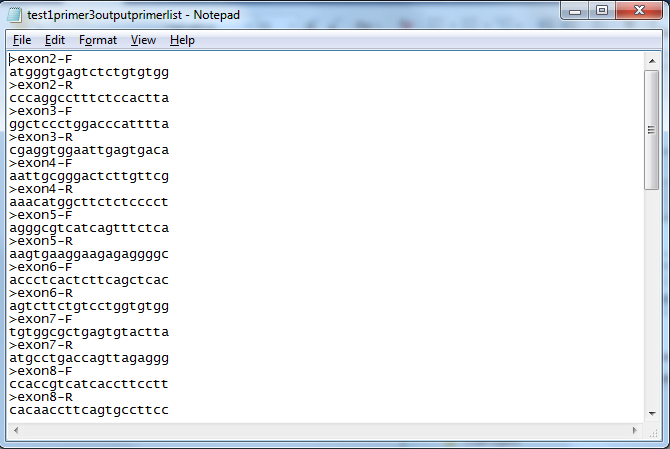


Please refer to the examples provided here and in the autoprimer folder for examples of formatting.

**Generating Primers:**

To generate primers themselves, the user will click ‘Input Filepath’ and select the input file. The same will be done with the parameters if the user chooses to use them. Then the ‘Output Filepath’ selection will be clicked where the user will choose what the output file will be named and where it will be saved. It should then look like so after loading all the filepaths. Simply click ‘Get Primers’ at this point to allow the program to run.

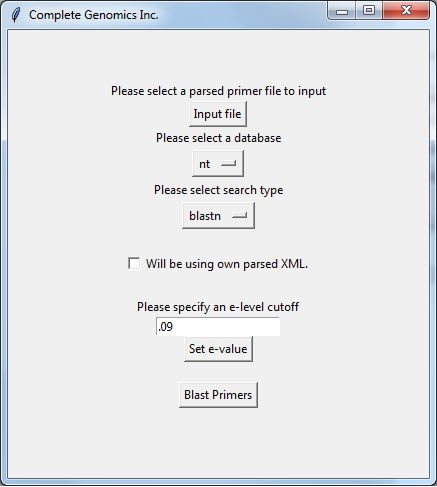


The generated will be wherever you specified the output will be with ‘primer3output’ appended to the name of the file. In general, the output presents a lot of detailed information which the user may not find useful right away. To get around this, the ‘Parse Primers’ button allows the program to read the output and returns a simple FASTA format of the best primers produced by the program. If additional primers are found by the program, they will not be put in. At this time, functionality to include additional primers is not present and will require the user to manually edit if they want to change the default primers given by the parsing functionality. The output of the ‘Parse Primers’ functions is like shown below..

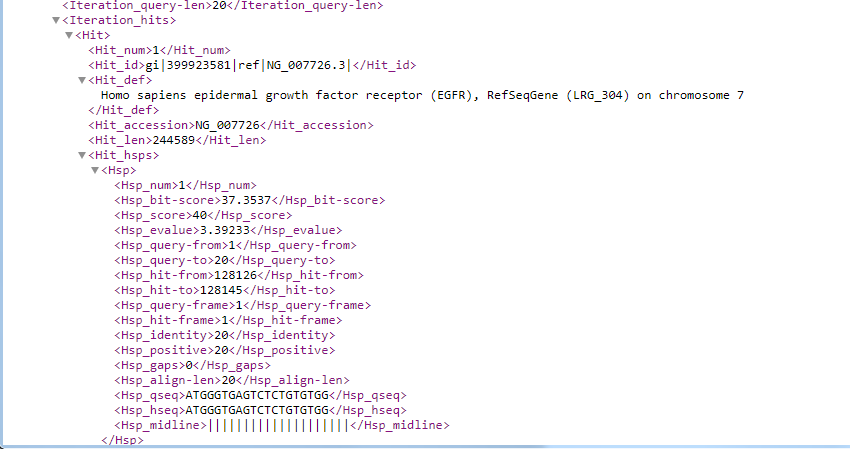
**BLASTING Primers:**

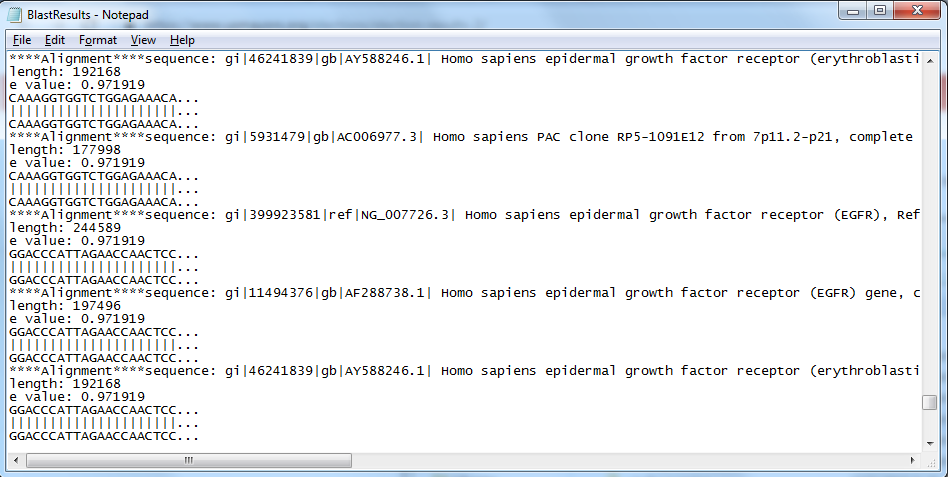
**Input files and formatting:**

This portion of the program deals with 3 different file types. The first is the FASTA format parsed primer list. This is the initial input that will be used by the program.



After the Blast request finishes, the output XML file will look like so:



Most of this information will be useless to a user unless we can properly go through it and put it in an easy to digest format. That is accomplished by the program as well, which allows generates a text file based on this XML file of possible hits and their score based on the e-value threshold cutoff the user sets. The final product is a file like so: 

**Generating BLAST Report:**

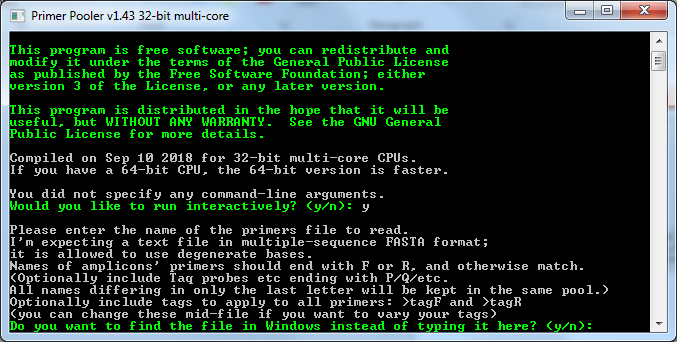
Usage of the program itself is quite simple. After loading the input file, the user will then be need to decide what type of search algorithm to perform on the queries, and what database to search against. A multitude of options are available. A suggestion for use of primers in human genomic DNA would be the ‘refseq\_genomic\_human’ database, as it only includes references of the human genome and has less overall items to search against, making it a much faster option. The user will then need to set an e-value threshold to decide where to cutoff hits that wouldn’t be useful. It helps to have prior experience using BLAST with primers as you will know acceptable ranges. Generally, the lower the e-value the better likelihood that the hits returned aren’t the result of random chance.

The user may also choose to upload their own XML file for various reasons, without needing to resubmit a BLAST request to the NCBI servers. The initial request itself can inevitably can take anywhere from seconds to tens of minutes depending on number of queries and the server load. Perhaps the user might just need to adjust the cutoff to see if they get any other hits. They can simply check ‘Will be using own parsed XML’ and then enter a new e-value cutoff. This will skip the blast request itself (since all of the information is already XML) and regenerate the new output file.

The XML file should be saved as a file called ‘my\_blast’ in the same folder as the executable. The final output file will be saved as ‘BlastResults’ and should be in the same folder. If the output for the file folder is blank, it could mean either there are no hits returned within the threshold, the BLAST request itself did not find anything meaningful, or there was an error in the request.

**Pooling Primers:**

This portion of the program deals with pooling the generated primers should the user require multiplexing in their experiments. It runs on the Primer Pooler tool developed by Dr. Silas Brown of the University of Cambridge. To initialize it, simply click the ‘Pool Primers’ button. The command line will open up (the black screen with text) and will be interfaced with there. It is a simple program, and the user simply needs to follow the on screen prompts. It accepts a FASTA formatted list of primers. Each forward primer needs to be denoted with –F at the end of the name and each reverse primer needs to denoted with –R. The user can decide to put unpaired primers together by adding a random letter to the end such as –P instead of F or R. The program will keep pairs together while it is optimizing the pools to put the primers in. At the first prompt, press ‘y’, then press ‘y’ again when prompted to search for the list of primers. In general, you will not need to deal with the other settings such as showing the highest bonds and pair scores. The user may wish to enter thermodynamic information at a certain point. The program prompt the user on this matter and should they select yes, they will be prompted for additional information. Simply follow the on screen instructions.



The program will generate the final primer list put in the proper pools based on your settings in the program. You can opt for the program to write each pool to its own file or have them all in one file labeled by pool.

**Relevant Links:**

Primer3:

http://primer3.sourceforge.net/primer3\_manual.htm

Pooling Resource:

<http://people.ds.cam.ac.uk/ssb22/pooler/>

BLAST:

<https://blast.ncbi.nlm.nih.gov/Blast.cgi>

<https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=DeveloperInfo>

Special thanks to:

Richard Chien

Lingling Peng

Mingcheng Yi

Ralph Harndan

Ben Allred

u/rindhoops (Steve Shambles, Redditor)

Mike-SMT (Stack Overflow)