

## Steps taken to make Blast-Parser run properly on input file generated by CMR Blast

- Bring up given program
  - Check File definition block to make sure path to given file OK
  - (Fix misspelling of Definition)
  - Change file path so it points to my directory
  - Run program. It ends normally.
  - Check output file. Seems OK, except that it is peculiar that only one query is found (multiple times) and only one target is found (multiple times)
  - Check input file. Indeed, only one query IS found, but there are multiple targets, not just one. So the given program doesn't even work properly on its own data, contradicting the conditions of the scenario.
  - Check out logic of Main Program. Logic does seem faulty: It says it looks for the FIRST query and the FIRST target but no further. The program appears then to be doing what the Main Program says its doing.
  - Upon reflection, I understand that the Main Program is not idiotic, simply misguided. It probably presumes that each hit will end by running into the next target, and each target will end by running into the next query, so there's no need to search for them any more.
  - Change the Main Program to reflect more general logic: (1) When hit is done, find next target unless it has already been found, (2) when target is done, find next query unless it has already been found.
  - Change name of subroutines *Find\_first\_query* and *Find\_first\_target* to be the more general *Find\_next\_query* and *Find\_first\_target*
  - Note that *Find\_seq* is erroneously designed to go through entire file. There's no provision made for leaving the loop except by crashing into the physical end of the file. Change it so that it leaves whenever a new hit, target, query, or logical end of file is reached
  - Change *Find\_next\_target* routine to run in a similar way as *Find\_seq*, i.e. when a target is or is not found, *target\_found* and *query\_found* set appropriately.
  - Now the output gives multiple targets, as it should.
- 
- (I also made a change that doesn't affect the running of the program but makes later analysis easier:

changed            LET comma\$ = ", "

to                    LET comma\$ = ", "                    ! note, no space after comma