```
#!/usr/bin/perl -w

#production of table genego is being shifted to a different script, one
written to collect evidence codes and produce 7 digit GO ID numbers.
rather than forcing the scripts together i will run them seperately. this
is not optimally efficient in terms of downloading data but it allows me
to make updates to GO data without updating all the gene dB stuff, which
should be a benefit.
#take the list of UniGene IDs and query GenedB with it. modified for
summer 2009.

use CGI;
use LWP::Simple;

#import list for search
open (GENELIST, "nr.geneid.only");
@genelist = (<GENELIST>);
close GENELIST;

#get the entry for each list item, build the URL and retrieve it
foreach $genelist (@genelist) {
    chomp $genelist;
    $URL =
"http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=gene&id=$gen
elist";
    $file = get($URL);
    $file =~ s/\n//g;
    $file =~ s/\r//g;

#    print $file, "\n\n";

#pre-declare and clear variables
    $putative = "";
    $HUGO = "";
    $chromo = "";
    @cyto = ();
    @contig = ();
    @nuc_start = ();
    @nuc_end = ();
    @strand = ();
    $omim = "";

    print $genelist, "\t";

#collect chosen datatypes
#         if ($file =~ /Official Symbol:\s\<\/strong\>(.*?) and /) {$HUGO
= $1; print $HUGO, "\t"}
#            if ($file =~ /1:\s{1}(.*?)\s{1}(.*?)\s{0,3}\[/) {$HUGO = $1;
print $HUGO, "\t"; $putative = $2, print $putative, "\t"}

#            if ($file =~ /Name:\s\<\/strong\>(.*?)\s{0,3}\[/) {$putative
= $1; print $putative, "\t"}
```

```
#         if ($file =~
/,\s{0,1}(\w{2}_.*?)\s{0,1}\((\d*)\.\.(\d*),\s{0,1}complement\)/) {
#         push @contig, $1; push @nuc_start, $2; push @nuc_end, $3; push
@strand, "-"; print $1, "\t", $2, "\t", $3, "\t", "-", "\t";
#           $contig = $1; $nuc_start = $2; $nuc_end = $3, $strand = "-";
print $1, "\t", $2, "\t", $3, "\t", "-", "\t";
#                }

#         if ($file =~ /,\s{0,1}(\w{2}_.*?)\w{0,1}\((\d*)\.\.(\d*)\)/) {
#         push @contig, $1; push @nuc_start, $2; push @nuc_end, $3; push
@strand, "+"; print $1, "\t", $2, "\t", $3, "\t", "+", "\t";
#           $contig = $1; $nuc_start = $2; $nuc_end = $3, $strand = "+";
print $1, "\t", $2, "\t", $3, "\t", "+", "\t";

#           }

#         if ($file =~ /MIM:\s\<\/dt\>\<dd class="desig"\>(\d*?)/) {$omim
= $1; print $omim, "\t"}

#         if ($file =~ /Chromosome:\s{0,1}\<\/dt>\<dd
class="desig"\>(.*?)\<\/dd\>/) {$chromo = $1; print $chromo, "\t"}
#             if ($file =~ /Location:\s{0,1}\<\/dt>\<dd
class="desig"\>(.*?)\<\/dd\>/) {$cyto = $1, print $cyto}

#    unless (@contig) {push @contig, "\t"; push @nuc_start, "\t"; push
@nuc_end, "\t"; push @strand, "\t"}

    print "\n";

#treat the entry as a string in order to pare down the XML file to just
the GeneOntology section, and skip this gene ID if it lacks a GO entry
    $file =~ s/\n//g;
    $file =~ s/(.*?)taxname "/taxname "/;
    if ($file =~ /homology \{/) {$file =~ s/homology \{(.*)//}
    if ($file =~
/\s{16}\}\s{14}\}\s{12}\}\s{10}\}\s{8}\}\s{6}\}\s{4}\}\s{2}\},/) {$file
=~ s/\s{16}\}\s{14}\}\s{12}\}\s{10}\}\s{8}\}\s{6}\}\s{4}\}\s{2}\},(.*)//}

    if ($file =~ /taxname "(.*?)",/) {$organism = $1}
    if ($file =~ /locus "(.*?)",/) {$HUGO = $1}
    if ($file =~ /desc "(.*?)",/) {$putative = $1}

#print output tables, gene, map

    open (GENE, ">>tablegene");
    print GENE $genelist, "\t", $HUGO, "\t", $putative, "\t", $chromo,
"\n";
    close GENE;

    open (MAP, ">>tablemap");
#    $i = 0;
#    until ($i => scalar(@contig)) {
```

```
#     print MAP $genelist, "\t", $chromo, "\t", $contig[$i], "\t",
$nuc_start[$i], "\t", $nuc_end[$i], "\t", $strand[$i], "\t", $cyto[$i],
"\n";
#     $i++;
#    }
    print MAP $genelist, "\t", $chromo, "\t", $contig, "\t", $nuc_start,
"\t", $nuc_end, "\t", $strand, "\t", $cyto, "\n";
    close MAP;

    open (OMIM, ">>tableomim");
    print OMIM $genelist, "\t", $omim, "\n";
    close OMIM;

}
```