

APPENDIX B

Figure 1

```
#!/usr/bin/perl -w
#Gregory Matuszek <ghm9231@bioinformatics.rit.edu>
#Copyright 2005

use strict;
use CGI qw(:standard :html3);
use IO::String;
use Bio::SeqIO;
use Bio::Seq;

my $sequence;

#the amino acid positions for the B and F binding pockets
my @b_pockets = qw{ 7 9 24 25 34 45 63 66 67 70 99 };
my @f_pockets = qw{ 77 80 81 84 95 116 123 143 146 147 };

#generate the HTML header information
print header(),
  start_html(-title => "Pocket Extract",
    -author => "Gregory Matuszek",
    -BGCOLOR => "#77AAAA");

if (param()) {
  #verify that a sequence has been entered by user
  # and store the sequence information
  if (defined (param('sequence'))) {
    $sequence = param('sequence');
  }

  #change font for the HTML to be fixed width for
  # cleaner display of results.
  print qq{<font face = "courier">\n};

  #create a table for the amino acid positions that
  # contribute to the B binding pocket. Generate
  # the first row of the table, containing the amino
  # acid positions that contribute to the B binding
  # pocket.
  print "<b>B Pocket</b>\n";
  print "<table>\n";
  print "<tr>\n";
  print "<td><b>Allele</b></td>\n";
  foreach my $n (@b_pockets) {
    print "<td><b>$n</b></td>\n";
  }
  print "</tr>\n";

  #break the sequence apart using the newline
  # characters at the end of each line of entered
```

```

# sequence information.
my @temp = split/\s+/, $sequence;

#remove the name from the array of lines and remove
# the > symbol from the beginning of the name
my $gene = shift(@temp);
$gene =~ s/>//;

#extract the amino acid sequence from the entered
# data. Concatenate each of the lines into one long
# string of amino acids.
my $amino;
foreach my $line (@temp){
    $amino = $amino . $line;
}

#Break the amino acid sequence into two parts. The
# cleavage occurs between the leader peptide and the
# alpha-1 domain. The two most common two consensus
# sequences are used to match the beginning of the
# alpha-1 domain.
#
#if the sequence does not contain one of the two
# patterns, return an error indicating that the start
# of the alpha-1 domain could not be found.
my $edit_seq;
if ($amino =~ /(\w+WA)(GSH\w+)/) {
    $amino =~ /(\w+WA)(GSH\w+)/;
    $edit_seq = $2;
}elsif ($amino =~ /(\w+RA)(GSH\w+)/) {
    $edit_seq = $2;
}else{
    print "Unable to extract pocket positions due to";
    print " inability to match start of the alpha-1";
    print " domain\n<br>";
}

#verify that the amino acid sequence is present and
# extract the positions for the B pocket. Display
# this information in the table for the B binding
# pocket positions.
if ($edit_seq eq " ") {
}else{
    my @base = split//, $edit_seq;
    print "<td><b>$gene</b></td>\n";
    foreach my $pos (@b_pockets) {
        print "<td>$base[ $pos-1 ]</td>\n";
    }
    print "</tr></table>\n<br>";
    print "<br><br><br>";

#Construct a second table similar to the first
# to display the amino acid residues that contribute

```

```

# to the F binding pocket. Conduct the same type of
# extraction as was done with the B pocket residues.
print "<b>F Pocket</b>\n";
print "<table>\n";
print "<tr>\n";
print "<td><b>Allele</b></td>\n";

foreach my $f (@f_pockets){
    print "<td><b>$f</b></td>\n";
}
print "</tr><tr>\n";
print "<td><b>$gene</b></td>\n";
foreach my $post (@f_pockets) {
    print "<td>$base[$post-1]</td>\n";
}
print "</tr>\n</table>\n";

}
}
#close out the HTML
print qq{</body></html>};

```

Figure 1. The perl CGI script developed for the extraction of the amino acid positions contributing to the B and F binding pockets of the MHC class I molecule. In order to help differentiate between the code and the comments, all comments have a pound (#) symbol at the beginning of the line and are colored in red.