Listing 1: Genbank Parser Code

```
# BsaX1 Sites
my $forward = "[ATGCN]{12}AC[ATGCN]{5}CTCC[ATGCN]{10}";
my $backward = "[ATGCN]{10}GGAG[ATGCN]{5}GT[ATGCN]{12}";
open FH, "< $sourcedir/$file";
my $organism = "";
my $divison = "";
if ($file =~ m/gb([a-z]{3})/) { $divison = $1; }

while (my $line = <FH>) {
    chomp $line;
    # Obtain organism name
    if ($line =~ m/^\s+ORGANISM\s+(.*)$/) { $organism = $1; }
    if ($line =~ m/^ORIGIN /) {
        # Parse and obtain whole DNA sequence
        my $wholeseq = "";
        while (my $seqline = <FH>) {
            chomp $seqline;
            if ($seqline =~ m/^\/\//) { last; }
            $seqline =~ s/\s//g;
            $seqline =~ s/[0-9]//g;
            $seqline =~ tr/a-z/A-Z/;
            $wholeseq = "$wholeseq$seqline";
        }
        # find all matches forwards and backwards
        foreach my $match ($wholeseq =~ m/($forward)/gi) {
            my $snipmatch = substr($match, 3, 27);
            print SEQFH "\"$snipmatch\"\n";
            print ORGFH "\"$snipmatch\", \"$organism\", \"$divison\"\n";
        }
        foreach my $match ($wholeseq =~ m/($backward)/gi) {
            # ensure all sites are oriented the same way
            $match =~ tr/ATGC/TACG/;
            $match = reverse($match);
            my $snipmatch = substr($match, 3, 27);
            print SEQFH "\"$snipmatch\"\n";
            print ORGFH "\"$snipmatch\", \"$organism\", \"$divison\"\n";
        }
    }
}
close FH;
```

Listing 2: Database Querying Code

```
# left join with seqs.origin
my $searchseq = $dbh->prepare("
    SELECT seqsOrigin.seqid, seqsOrigin.organism, seqsOrigin.division
    FROM seqs, seqsOrigin
    WHERE seqs.seqid = seqsOrigin.seqid AND seq = ?");

while (my $line = $jobqueue->dequeue) {
    my ($seq, $seqfreq) = split(/\|/, $line);

    # run prepared query
    $searchseq->execute($seq);

    # add all organism matches
    my $found = 0;
    while (my @row = $searchseq->fetchrow_array) {
        if ($found == 0) {
            $found = 1;
            if (exists($FOUNDSEQS{$seq})) { print "Consistency error! $seq already exists!\n"; }
            $FOUNDSEQS{$seq} = "$seqfreq|$row[1]\#$row[2]";
```

```
        } else { $FOUNDSEQS{$seq} .= "\#\#$row[1]\#$row[2]"; }
    }

    if ($found == 0) {
        $unknowncount++;
        if (exists($UNKNOWNSEQS{$seq})) { print "Consistency error! $seq already exists!\n"; }
        $UNKNOWNSEQS{$seq} = $seqfreq;
    }
}
```

Listing 3: Genotype Generation Code

```
# Calculate chromosome distribution of sample tags
my $totalTagsChrom = 0;
my @adjustment;
open(FH, "> $OUTPUTDIR/distribution");
for (my $i = 0; $i <= $#filearray; $i++) {
    my @temparr = split(/\//, $filearray[$i]);
    my $chrname = $temparr[$#temparr];
    my $totalsample = 0;
    foreach my $tag (keys %{$MAPARRAY[$i]}) {
        my $tagfreq = $SAMPLEFREQ{$tag};
        my $countmapfreq = $MAPARRAY[$i]->{$tag};
        if (!exists($MAPARRAY[$i]->{$tag})) {
            print "\nERROR: $tag, $tagfreq\n";
            exit(-1);
        }
        $totalsample += $tagfreq;
    }
    printf FH ("$chrname \t\t $finaloutput[$i] \t $totalsample \t %.3f \n",($totalsample /
        $finaloutput[$i]));
    $adjustment[$i] = $totalsample / $finaloutput[$i];
    $totalTagsChrom += $totalsample;
}
printf FH ("Total: $totalTagsChrom\n");
close(FH);

# Walk down each of the chromosomes and generate LaTeX using pstricks
open(LATEX, "> $OUTPUTDIR/figures.tex");
print LATEX <<MESSAGE;
\\documentclass[10pt, twocolumn]{article}
\\usepackage{geometry}
\\geometry{letterpaper}
\\usepackage{graphicx}
\\usepackage{amssymb}
\\usepackage{epstopdf}
\\usepackage{booktabs}
\\usepackage{pstricks}
\\usepackage{pst-plot}

\\DeclareGraphicsRule{.tif}{png}{.png}{`convert #1 `dirname #1`/`basename #1 .tif`.png}

\\begin{document}
%\\maketitle
\\thispagestyle{empty}
MESSAGE

for (my $i = 0; $i <= $#filearray; $i++) {
    my @temparr = split(/\//, $filearray[$i]);
    my $chrname = $temparr[$#temparr];
    my @orderedtags;
    my $chrstr = $i + 1;
    if ($chrstr == 23) { $chrstr = "X"; } elsif ($chrstr == 24) { $chrstr = "Y"; }

```

```
52      print "  Processing $chrname\n";
53      open(FH, "< $MAPDIR/$chrname");
54      my $counter = 0;
55      while (my $line = <FH>) {
56          chomp $line;
57          my @temparr2 = split(/\|/, $line);
58          $orderedtags[$counter][0] = $temparr2[0];
59          $orderedtags[$counter][1] = $temparr2[1];
60          $counter++;
61      }
62      close(FH);
63
64      my @bins;
65      my $curbin = 0;
66      my $binfill = 0;
67      $bins[$curbin][0] = 0;
68      for (my $j = 0; $j < $counter; $j++) {
69          $binfill++;
70          my $tag = $orderedtags[$j][0];
71          if (exists($SAMPLEFREQ{$tag})) { $bins[$curbin][0] += $SAMPLEFREQ{$tag} / $MAPARRAY[$i
                ]->{$tag}; }
72          if ($binfill == $BINCOUNT) {
73              $binfill = 0;
74              $bins[$curbin][1] = $orderedtags[$j][1];
75              $curbin++;
76              $bins[$curbin][0] = 0;
77          }
78      }
79      $bins[$curbin][1] = $orderedtags[$counter - 1][1];
80      print LATEX <<MESSAGE;
81
82      \\begin{center}
83      %\\begin{figure}[H]
84          \\psset{xunit=.00245\\columnwidth,yunit=.055\\columnwidth}
85      \\begin{pspicture}(0,-.25)(250,4.25)
86          \\rput{90}(-60,2.0){\\footnotesize Copy number}
87          \\rput[Bl](225,-1.30){Mb}
88          \\rput(125,-2.10){Chromosome $chrstr}
89          \\psaxes[Dx=50,Dy=1.0]{->}(250,4.25)
90          \\savedata{\\maxgen}[{
91  MESSAGE
92      open(FH, "> $OUTPUTDIR/$chrname");
93      for (my $j = 0; $j <= $curbin; $j++) {
94          if ($adjustment[$i] > 0) {
95              printf FH "$j,  %.2f, $bins[$j][1]\n", $bins[$j][0];
96              if ($j < $curbin - 1) {
97                  printf LATEX "{ %.2f, %.2f}, ", ($bins[$j][1]/1000000), ($bins[$j][0] / (
                        $adjustment[$i] * $BINCOUNT));
98              } elsif($j == $curbin - 1) {
99                  printf LATEX "{ %.2f, %.2f} ", ($bins[$j][1]/1000000), ($bins[$j][0] / (
                        $adjustment[$i] * $BINCOUNT));
100             }
101         }
102     }
103     close(FH);
104     print LATEX "}]
105         \\dataplot[plotstyle=line,dotstyle=x,linecolor=black]{\\maxgen}
106     \\end{pspicture}
107     \\label{trends1}
108     \\end{center}
109     \\vspace{5mm}\n";
110 }
111 print LATEX "\\end{document}";
112 close(LATEX);
```