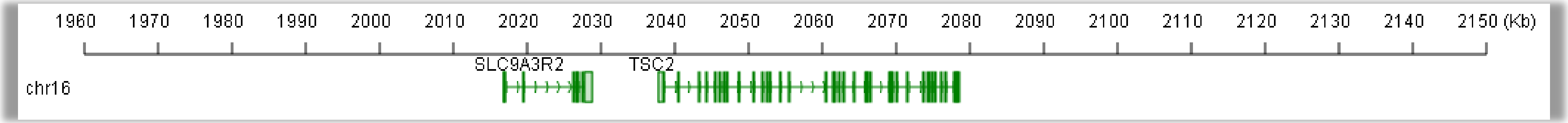
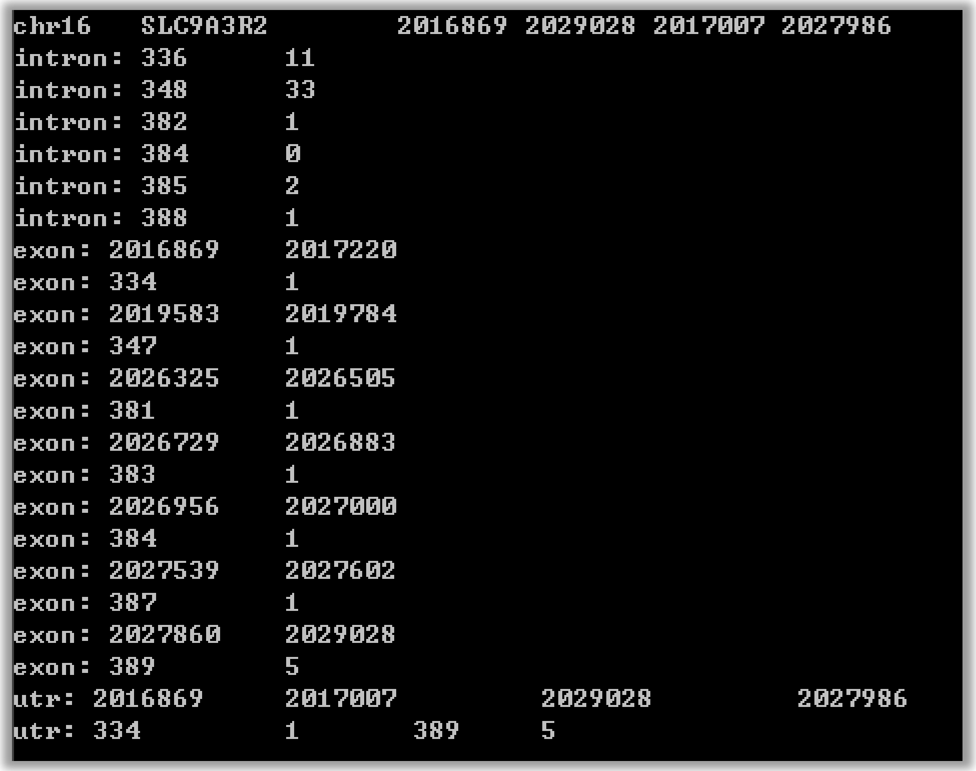
Q10: Complete subroutine plot gene and write a Perl program to plot the structure of human genes using SVG.

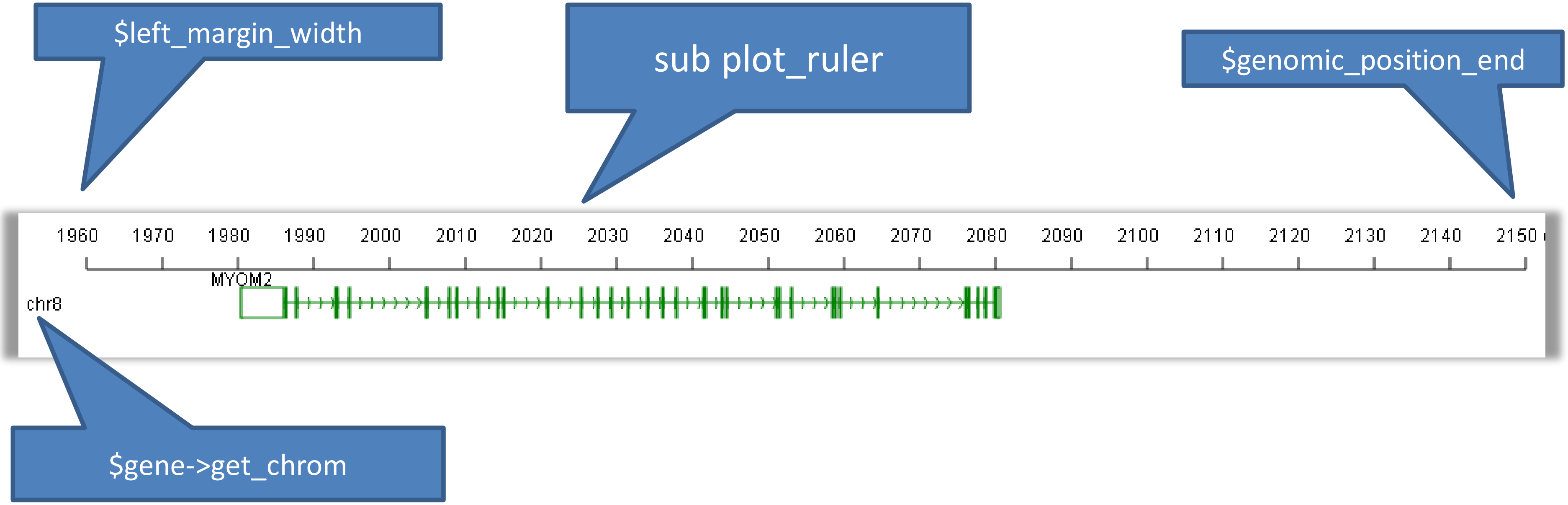
Output





```
chr16   SLC9A3R2        2016869 2029028 2017007 2027986
intron: 336      11
intron: 348      33
intron: 382      1
intron: 384      0
intron: 385      2
intron: 388      1
exon: 2016869    2017220
exon: 334        1
exon: 2019583    2019784
exon: 347        1
exon: 2026325    2026505
exon: 381        1
exon: 2026729    2026883
exon: 383        1
exon: 2026956    2027000
exon: 384        1
exon: 2027539    2027602
exon: 387        1
exon: 2027860    2029028
exon: 389        5
utr: 2016869     2017007         2029028         2027986
utr: 334         1       389     5
```

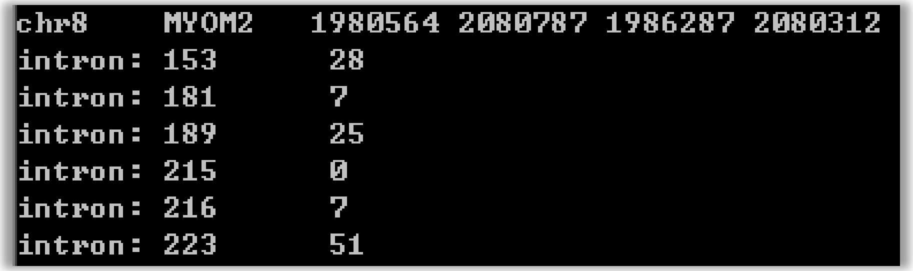
Output


$left_margin_width

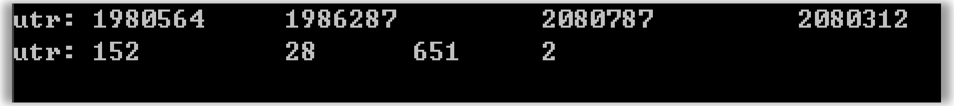
sub plot_ruler

$genomic_position_end

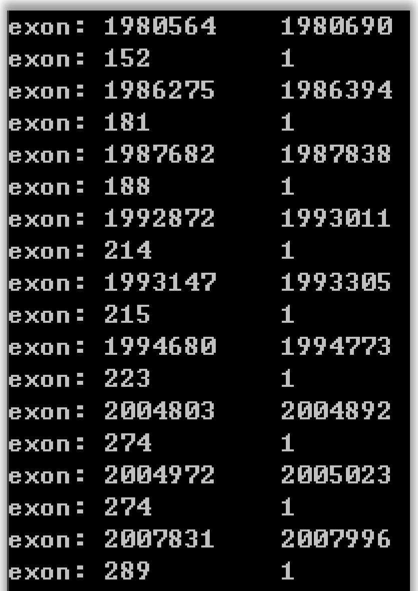
$gene->get_chrom



```
chr8    MYOM2   1980564 2080787 1986287 2080312
intron: 153      28
intron: 181      7
intron: 189      25
intron: 215      0
intron: 216      7
intron: 223      51
```



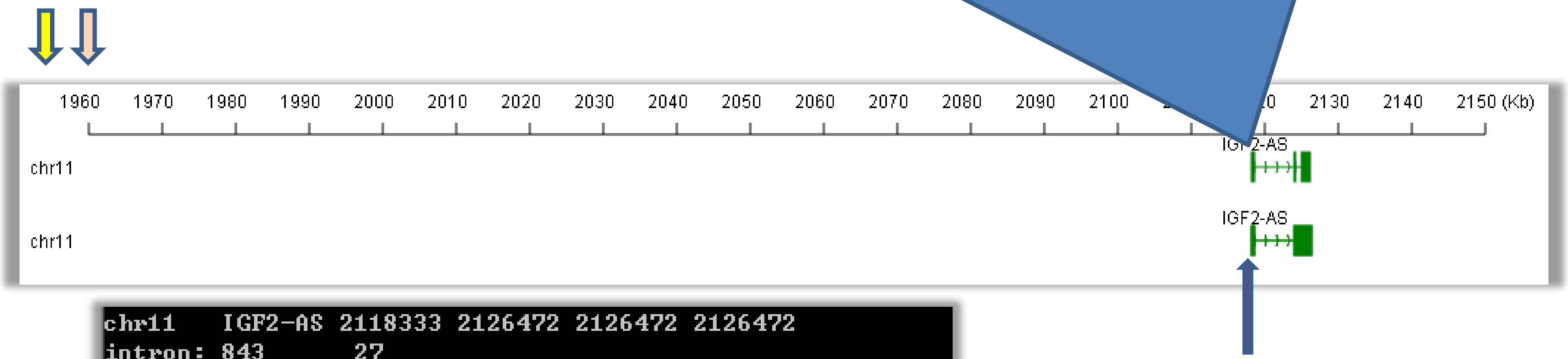
```
utr: 1980564         1986287              2080787              2080312
utr: 152             28         651       2
```

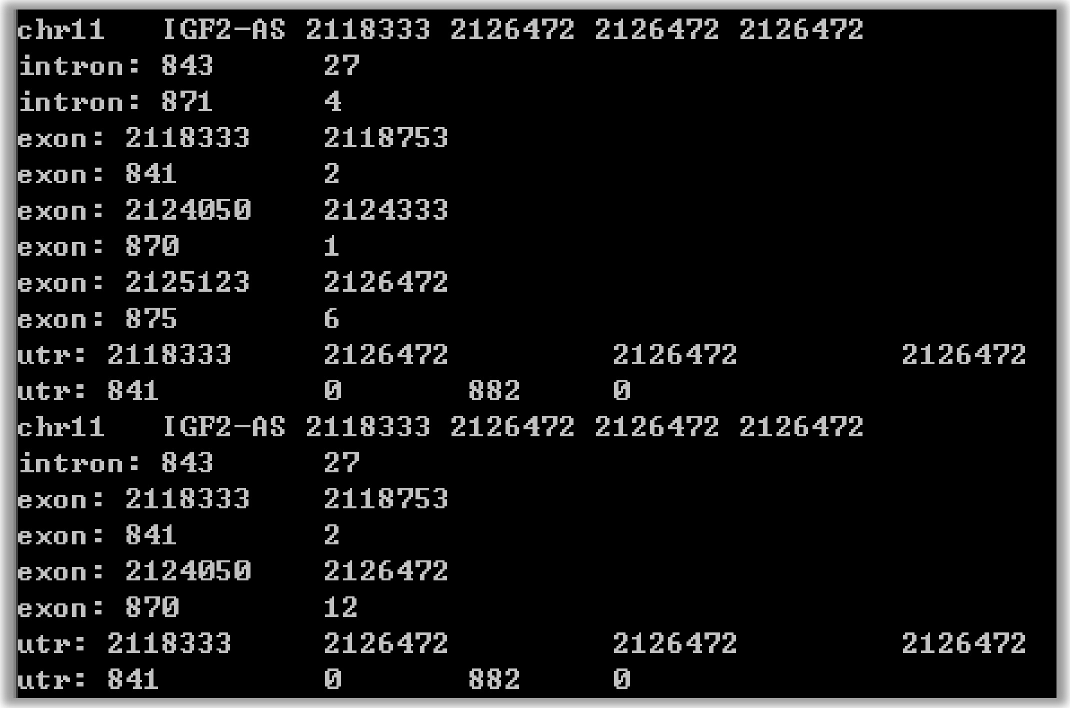


```
exon: 1980564      1980690
exon: 152          1
exon: 1986275      1986394
exon: 181          1
exon: 1987682      1987838
exon: 188          1
exon: 1992872      1993011
exon: 214          1
exon: 1993147      1993305
exon: 215          1
exon: 1994680      1994773
exon: 223          1
exon: 2004803      2004892
exon: 274          1
exon: 2004972      2005023
exon: 274          1
exon: 2007831      2007996
exon: 289          1
```

Output

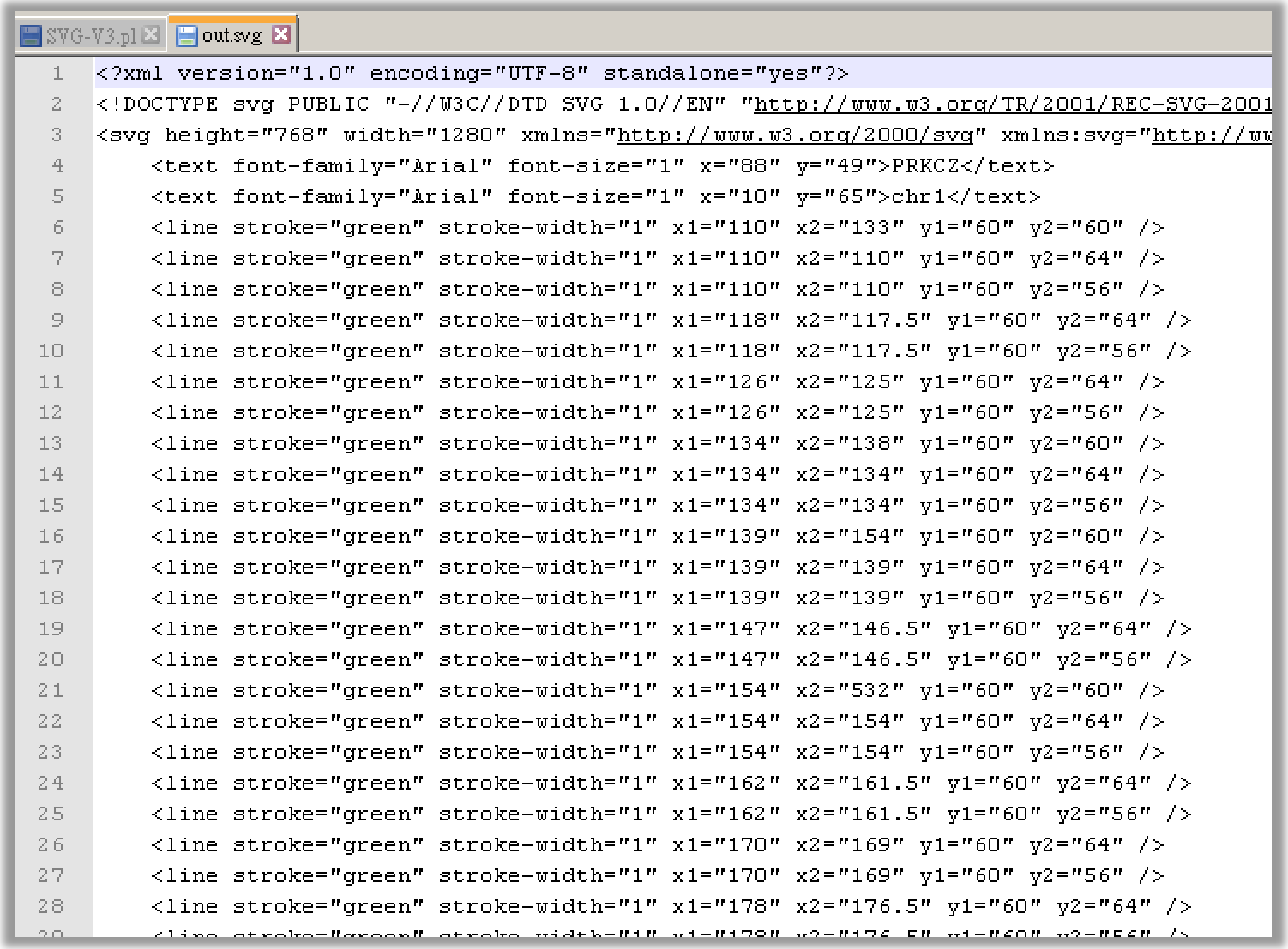
my $x1= int(($exon_beg[$i]- $genomic_position_start)/$scale) + $left_margin_width





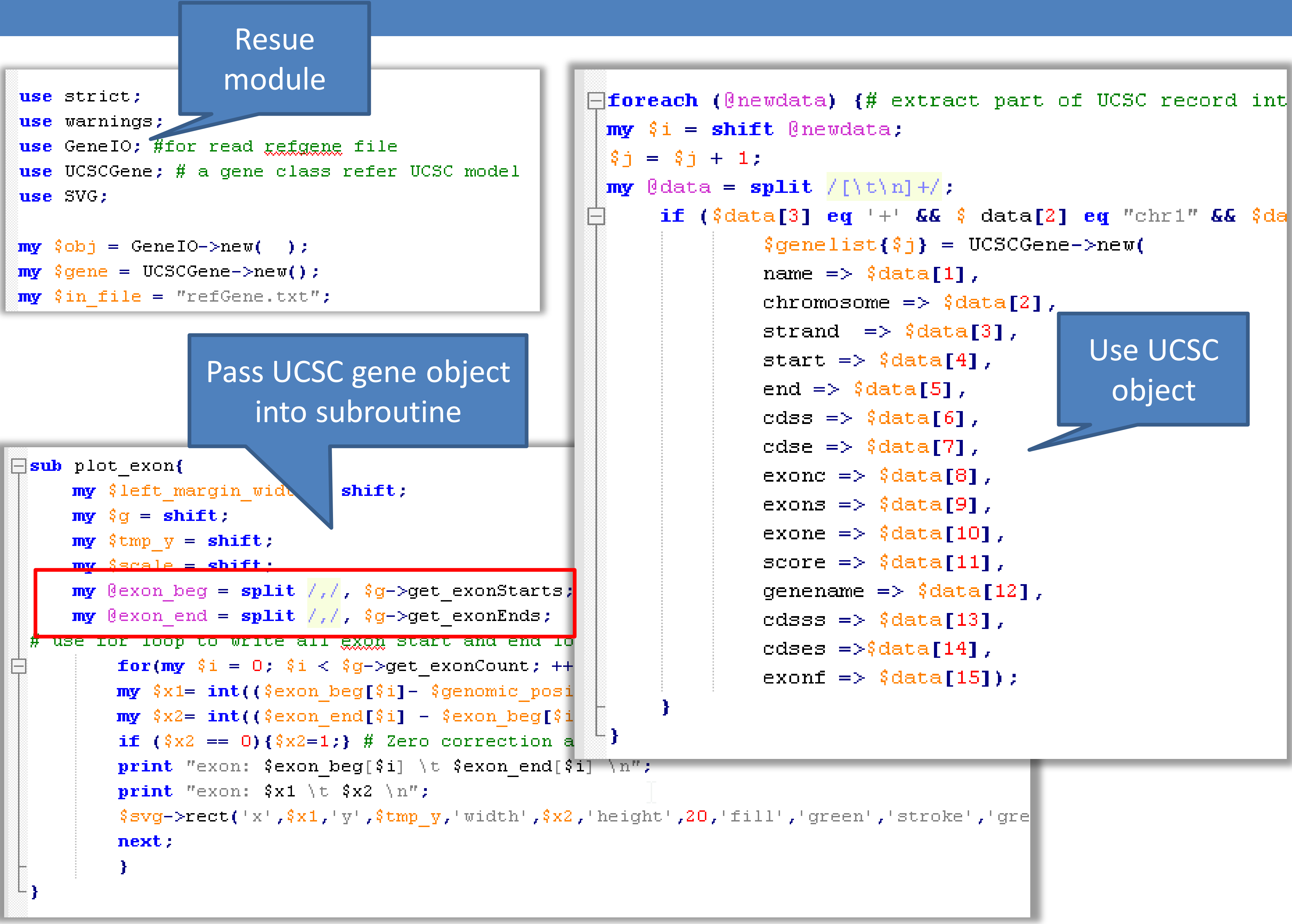
```
chr11   IGF2-AS 2118333 2126472 2126472 2126472
intron: 843     27
intron: 871     4
exon: 2118333   2118753
exon: 841       2
exon: 2124050   2124333
exon: 870       1
exon: 2125123   2126472
exon: 875       6
utr: 2118333    2126472         2126472         2126472
utr: 841        0       882     0
chr11   IGF2-AS 2118333 2126472 2126472 2126472
intron: 843     27
exon: 2118333   2118753
exon: 841       2
exon: 2124050   2126472
exon: 870       12
utr: 2118333    2126472         2126472         2126472
utr: 841        0       882     0
```

Output



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-2001
<svg height="768" width="1280" xmlns="http://www.w3.org/2000/svg" xmlns:svg="http://ww
    <text font-family="Arial" font-size="1" x="88" y="49">PRKCZ</text>
    <text font-family="Arial" font-size="1" x="10" y="65">chr1</text>
    <line stroke="green" stroke-width="1" x1="110" x2="133" y1="60" y2="60" />
    <line stroke="green" stroke-width="1" x1="110" x2="110" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="110" x2="110" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="118" x2="117.5" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="118" x2="117.5" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="126" x2="125" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="126" x2="125" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="134" x2="138" y1="60" y2="60" />
    <line stroke="green" stroke-width="1" x1="134" x2="134" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="134" x2="134" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="139" x2="154" y1="60" y2="60" />
    <line stroke="green" stroke-width="1" x1="139" x2="139" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="139" x2="139" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="147" x2="146.5" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="147" x2="146.5" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="154" x2="532" y1="60" y2="60" />
    <line stroke="green" stroke-width="1" x1="154" x2="154" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="154" x2="154" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="162" x2="161.5" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="162" x2="161.5" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="170" x2="169" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="170" x2="169" y1="60" y2="56" />
    <line stroke="green" stroke-width="1" x1="178" x2="176.5" y1="60" y2="64" />
    <line stroke="green" stroke-width="1" x1="178" x2="176.5" y1="60" y2="56" />
```

Overview



Resue module

```
use strict;
use warnings;
use GeneIO; #for read refgene file
use UCSCGene; # a gene class refer UCSC model
use SVG;

my $obj = GeneIO->new(  );
my $gene = UCSCGene->new();
my $in_file = "refGene.txt";
```

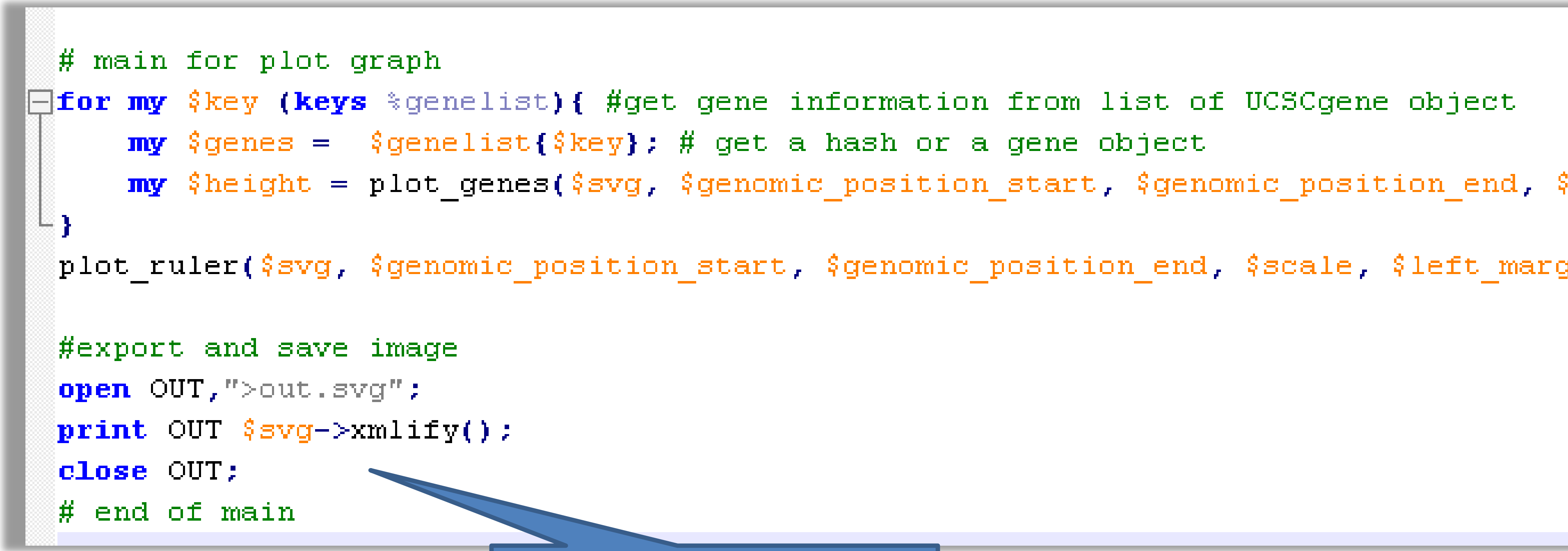
Pass UCSC gene object into subroutine

```
sub plot_exon{
    my $left_margin_wid        shift;
    my $g = shift;
    my $tmp_y = shift;
    my $scale = shift;
    my @exon_beg = split /,/, $g->get_exonStarts;
    my @exon_end = split /,/, $g->get_exonEnds;
# use for loop to write all exon start and end lo
        for(my $i = 0; $i < $g->get_exonCount; ++
        my $x1= int(($exon_beg[$i]- $genomic_posi
        my $x2= int(($exon_end[$i] - $exon_beg[$i
        if ($x2 == 0){$x2=1;} # Zero correction a
        print "exon: $exon_beg[$i] \t $exon_end[$i] \n";
        print "exon: $x1 \t $x2 \n";
        $svg->rect('x',$x1,'y',$tmp_y,'width',$x2,'height',20,'fill','green','stroke','gre
        next;
        }
}
```

```
foreach (@newdata) {# extract part of UCSC record int
my $i = shift @newdata;
$j = $j + 1;
my @data = split /[\t\n]+/;
    if ($data[3] eq '+' && $ data[2] eq "chr1" && $da
            $genelist{$j} = UCSCGene->new(
            name => $data[1],
            chromosome => $data[2],
            strand  => $data[3],
            start => $data[4],
            end => $data[5],
            cdss => $data[6],
            cdse => $data[7],
            exonc => $data[8],
            exons => $data[9],
            exone => $data[10],
            score => $data[11],
            genename => $data[12],
            cdsss => $data[13],
            cdses =>$data[14],
            exonf => $data[15]);
    }
}
```

Use UCSC object

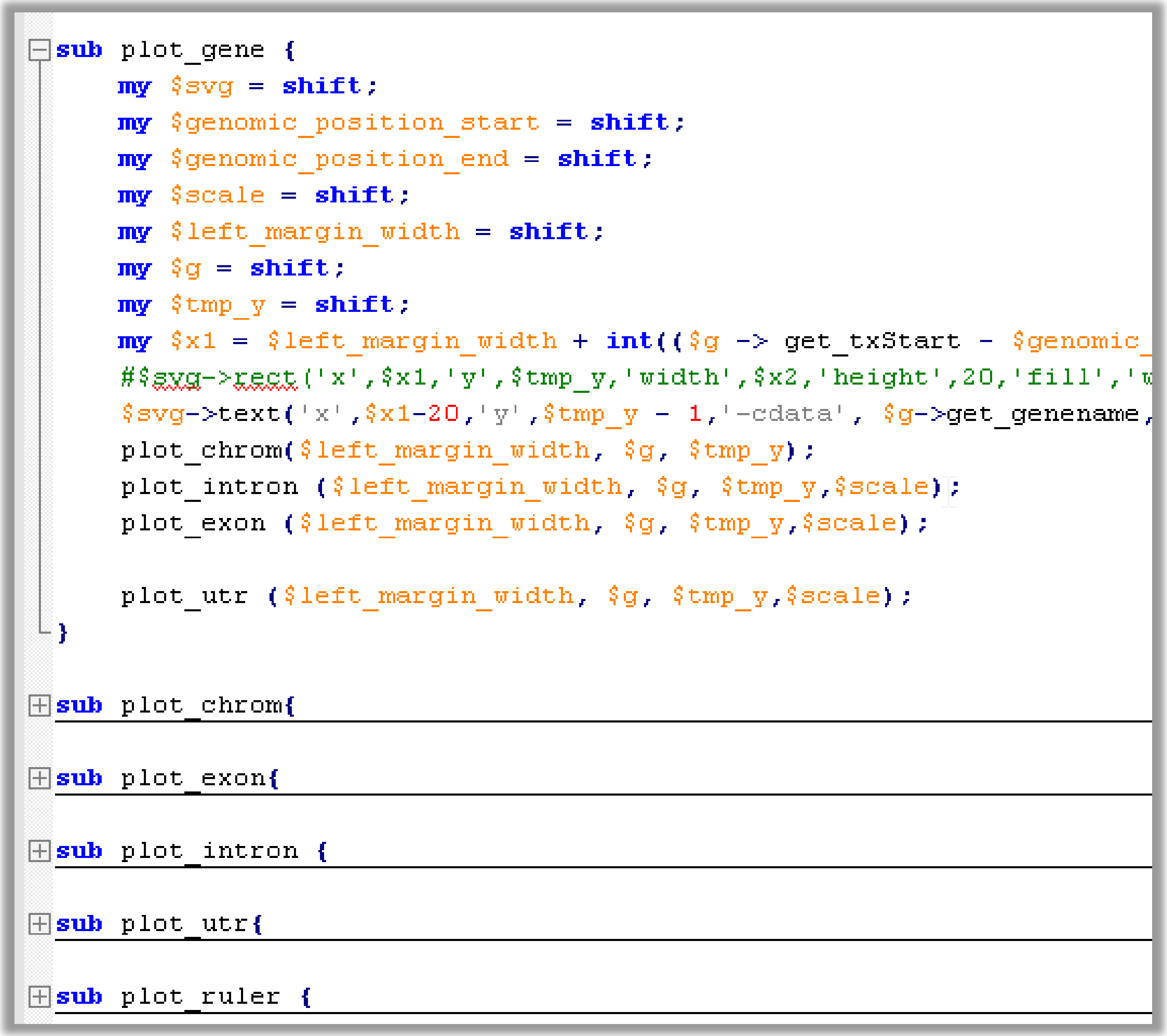
Main



```
# main for plot graph
for my $key (keys %genelist){ #get gene information from list of UCSCgene object
    my $genes =  $genelist{$key}; # get a hash or a gene object
    my $height = plot_genes($svg, $genomic_position_start, $genomic_position_end, $
}
plot_ruler($svg, $genomic_position_start, $genomic_position_end, $scale, $left_marg

#export and save image
open OUT,">out.svg";
print OUT $svg->xmlify();
close OUT;
# end of main
```

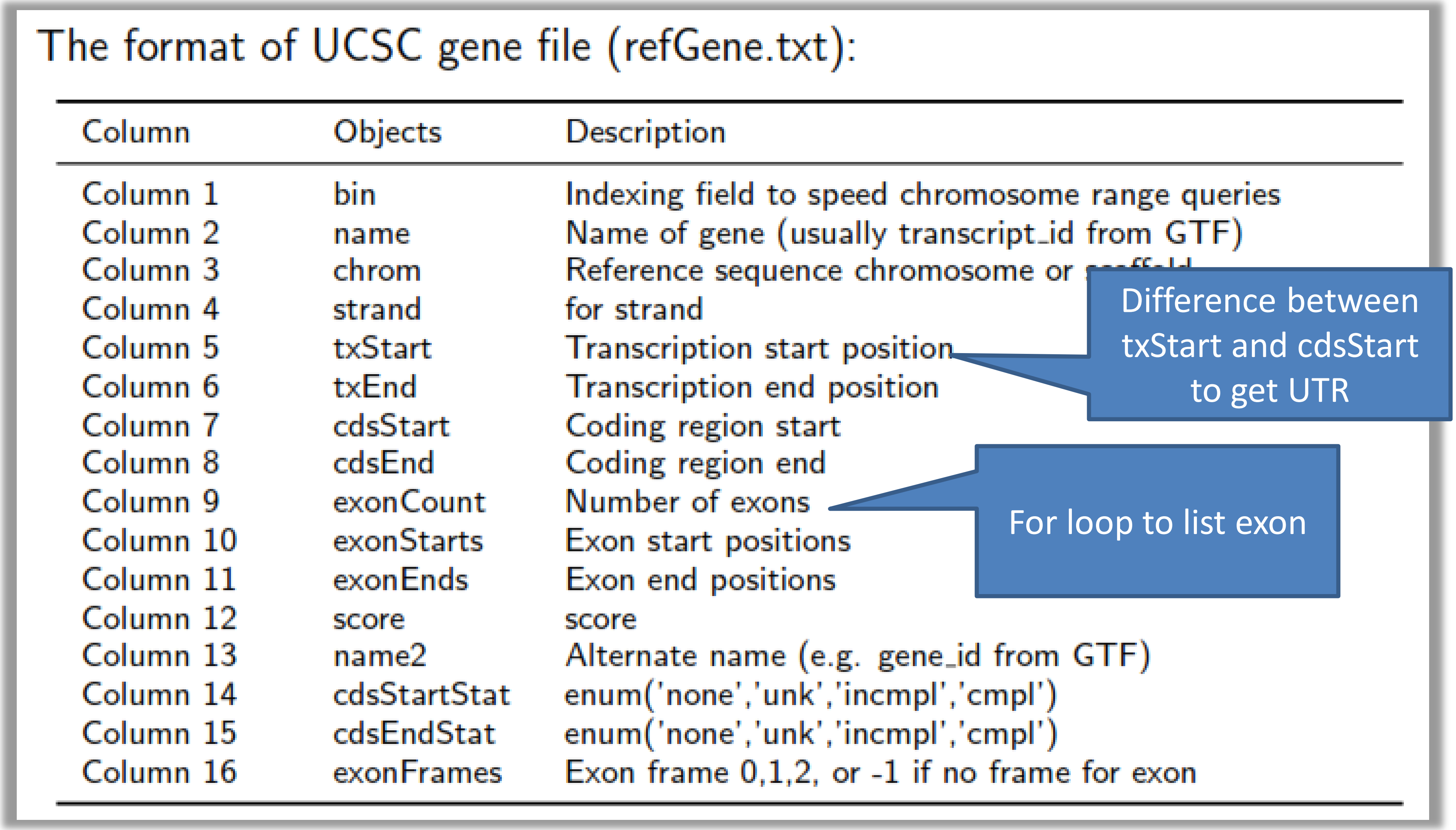
Draw in $svg object and output as XML and output.svg

Subroutine



```
sub plot_gene {
    my $svg = shift;
    my $genomic_position_start = shift;
    my $genomic_position_end = shift;
    my $scale = shift;
    my $left_margin_width = shift;
    my $g = shift;
    my $tmp_y = shift;
    my $x1 = $left_margin_width + int(($g -> get_txStart - $genomic_
    #$svg->rect('x',$x1,'y',$tmp_y,'width',$x2,'height',20,'fill','w
    $svg->text('x',$x1-20,'y',$tmp_y - 1,'-cdata', $g->get_genename,
    plot_chrom($left_margin_width, $g, $tmp_y);
    plot_intron ($left_margin_width, $g, $tmp_y,$scale);
    plot_exon ($left_margin_width, $g, $tmp_y,$scale);

    plot_utr ($left_margin_width, $g, $tmp_y,$scale);
}

sub plot_chrom{

sub plot_exon{

sub plot_intron {

sub plot_utr{

sub plot_ruler {
```

Calculation



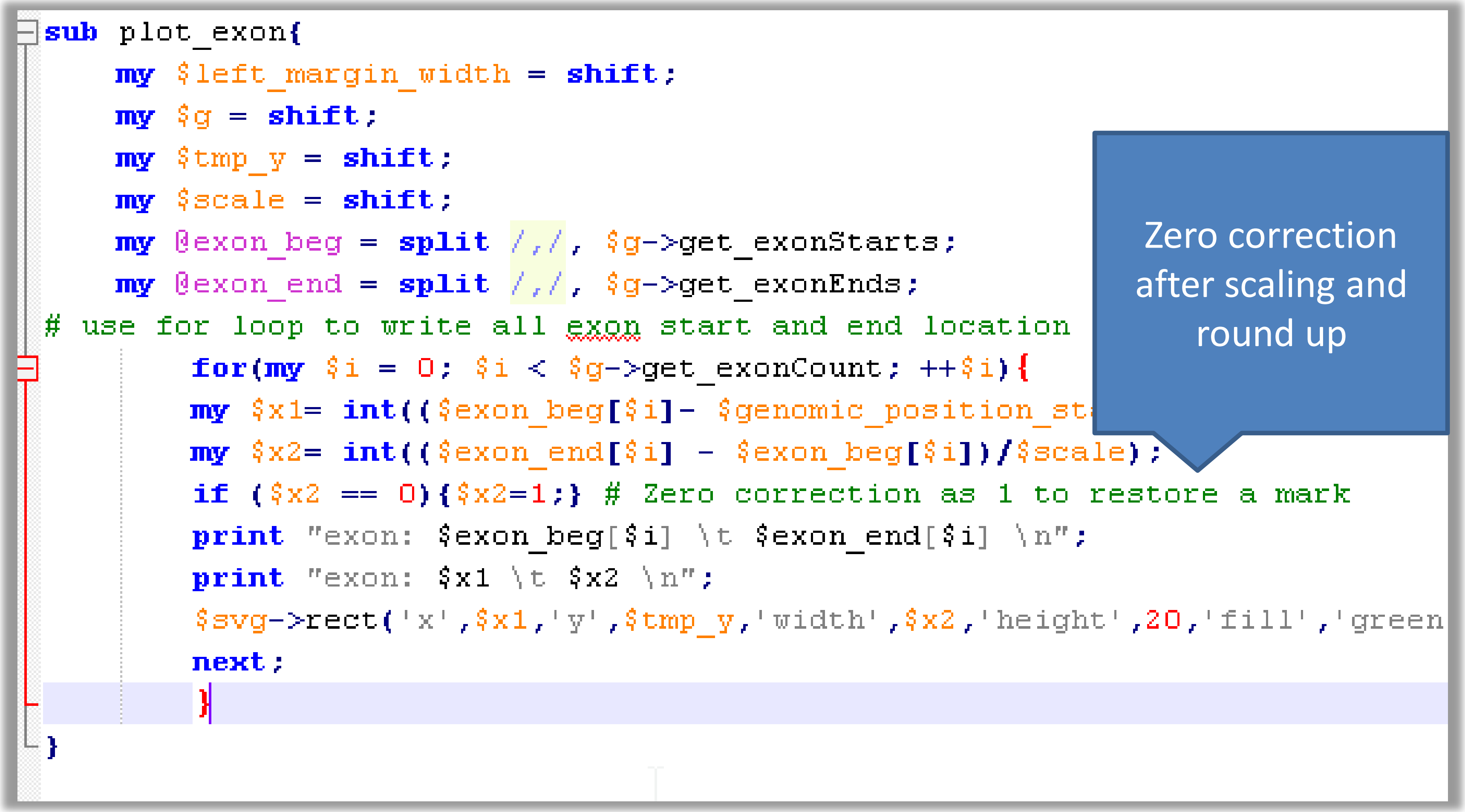
The format of UCSC gene file (refGene.txt):

Column	Objects	Description
Column 1	bin	Indexing field to speed chromosome range queries
Column 2	name	Name of gene (usually transcript_id from GTF)
Column 3	chrom	Reference sequence chromosome or
Column 4	strand	for strand
Column 5	txStart	Transcription start position
Column 6	txEnd	Transcription end position
Column 7	cdsStart	Coding region start
Column 8	cdsEnd	Coding region end
Column 9	exonCount	Number of exons
Column 10	exonStarts	Exon start positions
Column 11	exonEnds	Exon end positions
Column 12	score	score
Column 13	name2	Alternate name (e.g. gene_id from GTF)
Column 14	cdsStartStat	enum('none','unk','incmpl','cmpl')
Column 15	cdsEndStat	enum('none','unk','incmpl','cmpl')
Column 16	exonFrames	Exon frame 0,1,2, or -1 if no frame for exon

Difference between txStart and cdsStart to get UTR

For loop to list exon

Calculation



```
sub plot_exon{
    my $left_margin_width = shift;
    my $g = shift;
    my $tmp_y = shift;
    my $scale = shift;
    my @exon_beg = split /,/, $g->get_exonStarts;
    my @exon_end = split /,/, $g->get_exonEnds;
# use for loop to write all exon start and end location
        for(my $i = 0; $i < $g->get_exonCount; ++$i){
        my $x1= int(($exon_beg[$i]- $genomic_position_sta
        my $x2= int(($exon_end[$i] - $exon_beg[$i])/$scale);
        if ($x2 == 0){$x2=1;} # Zero correction as 1 to restore a mark
        print "exon: $exon_beg[$i] \t $exon_end[$i] \n";
        print "exon: $x1 \t $x2 \n";
        $svg->rect('x',$x1,'y',$tmp_y,'width',$x2,'height',20,'fill','green
        next;
        }
}
```

Zero correction after scaling and round up

- END -