ReferenceSNPs_FirstStep

Unknown Author

March 20, 2014

This notebook has the code used to process the SNP files generated by the AMOS processes pipeline. To generate these files, the assemblies of the J07 and A07 populations (generated using the Celera Assembler), were proccessed using the analyzeSNPs script from AMOS.

The parameteres used for analyzeSNPs, were: - Minimum depth of 4 - At least two different bases of consensus - Quality score of 20 or more

For those assemblies that had 454 reads, those reads were not considered in the analysis. Only SNPs supported by Sanger reads were considered.

In [1]:

```
#Import the modules needed
from collections import defaultdict
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import seaborn as sns
from collections import defaultdict
import re
from Bio import SeqIO
```

In [2]:

```
#Read the list of scaffolds and store in a dictionary
#The structure is: key-> Assembly_ID
#                  values -> [JGI_ID, length]
def read_scaffold_list(input):

    output_dic = defaultdict(list)

    for line in open(input, ’r’):
        if line.strip():
            line = line.rstrip()
            info = line.split("\t")
            output_dic[info[0]] = [info[2],info[1]]

    return output_dic
```

In [3]:

```
#Read the contig-scaffold mapping, store in dictionary
#The structure is: key->contig_id
#Values:           values->[scaf_id,start,end,orientation]

def read_ctg_scaf_map(input):
    output_dic = defaultdict(list)

    for line in open(input, ’r’):
        if line.strip():
            line = line.rstrip()
            info = line.split("\t")
            output_dic[info[0]] = [info[1],info[2],info[3],info[4]]

    return output_dic
```

In [4]:

```
#Read the AMOS SNP file, change info according to the scaffolds with the JGI ID and st
#the information ready for the VCF file
#Dictionary: key->scaffold
#             values: [position,id,etc..... (rest of columns VCF)

def read_amos_snp(scaf_list,mapping,input_snp,genome_file):
    logfile = open("logfile.txt", ’w’)

    #Dictionary to get the complementary nucleotide
    complement_dict = {"A":"T","T":"A","C":"G","G":"C"}


    fasta_sequences = SeqIO.parse(open(genome_file), "fasta")
    genome_dictionary = defaultdict(str)

    for fasta in fasta_sequences:
        name, sequence = fasta.id, fasta.seq.tostring()
        genome_dictionary[name] = sequence


    output_vcf_results = defaultdict(lambda: defaultdict(list))
    snp_count = defaultdict(int)

    for line in open(input_snp, ’r’):

        if line.startswith("AsmblID"):
            continue
        if line.strip():

            line = line.rstrip()
            info = line.split("\t")

            ctg_id = info[0]
            ungapped_position = info[2]
            consensus_base = info[3]
            depth_coverage = info[4]

            bases_info = info[6:]

            #Skip gaps, because we are looking for SNPs, and insertion/deletions compl
            #of the VCF file
            if consensus_base == "-":
                continue

            #Skip if coverage is less than four:
            if not int(depth_coverage) > 3:
                continue

            #Skip those contig with no mapping info
            if not ctg_id in mapping:
                continue


            #Only look at the scaffolds that are part of the assembly
            scf_id,start,end,orientation = mapping[ctg_id]
            if not scf_id in scaf_list:
                continue

            jgi_scf_id = scaf_list[scf_id][0]

            #Calculate the SNP position in the scaffold
            snp_position_scf = 0

            if orientation == ’f’:
                snp_position_scf = int(start) + int(ungapped_position)

            if orientation == ’r’:
                snp_position_scf = int(end) - int(ungapped_position) + 1
            #Get the depth of each base
            bases_dict = defaultdict(int)
            adjusted_depth = 0


            for base in bases_info:
                number_search = re.match(’(\D+)\((\d+)\)’,base)

                if int(number_search.group(2)) > 2:
                    base = str()

                    if orientation == ’r’:

                        if number_search.group(1) in complement_dict:
                            base = complement_dict[number_search.group(1)]
                        else:
                            base = number_search.group(1)

                    if orientation == ’f’:
                        base = number_search.group(1)


                    bases_dict[base] = number_search.group(2)
                    adjusted_depth += int(number_search.group(2))

            gap_counter = 0
            for base in bases_dict:
                if base == "-":
                    gap_counter += 1

            if not gap_counter == 0:
                continue




            #Because some of the assemblies are based in 454 data, I need to check tha
            #is indeed the deepest base. If not, I just move to the next one.
            #This is stringent, but the idea is to look at SNPs that are supported by
            #The details of the population structure should come out from the Illumina

            if not len(bases_dict.keys()) > 1:
                continue



            #Values for the final dictionary
            #Chrom
            #Pos
            #ID, always .
            #REF
            #ALT If multiple, separated by , Replace "-" with "."
            #Qual, always 20
            #Filter, always PASS
            #INFO: #NS=1,DP=depth,AF=allele frequency

            #For some reason, AMOS gives position that are not the ones found in the c
            #Here I will check against the fasta file for the sequence, to see if the
            #or the alternates



            #Create the alt and allele frequency values
            #genome_reference_nuc = genome_sequence[jgi_scf_id].seq[int(snp_position_s
            reference_position = int(snp_position_scf) - 1

            try:
                genome_reference_nuc = genome_dictionary[jgi_scf_id][reference_positio
            except IndexError:
                logfile.write("Error: %s\t%d\n" %(jgi_scf_id,reference_position))
                continue

            new_reference_position = "."
            alternative_bases = []
            allele_frequency = []


            for entry in bases_dict.keys():

                if entry == genome_reference_nuc:
                    new_reference_position = entry

                else:
                    alternative_bases.append(entry)
                    freq = int(bases_dict[entry]) / float(adjusted_depth)
                    allele_frequency.append(freq)

            #Now save everything into an entry

            vcf_alt = ",".join(alternative_bases)
            info_entry = "NS=1," + "DP=" + str(adjusted_depth) + ",AF=" + ",".join(map

            #Modify consensus base to "."
            #if consensus_base == "-":
            #    consensus_base = "."

            vcf_entry = [".", new_reference_position, vcf_alt, "20", "PASS", info_entr

            snp_count[scf_id] += 1

            output_vcf_results[jgi_scf_id][snp_position_scf] = vcf_entry

            logfile.write(orientation + "\t" + ",".join(bases_dict.keys()) + "\t" + ge

    logfile.close()
    return output_vcf_results,snp_count
```

In [5]:

```
def write_vcf(snp_info,file):
    #This function will write a vcf file based on the generated snp dictionary
    import datetime
    today = datetime.date.today()

    outfile = open(file, ’w’)

    outfile.write("##fileformat=VCFv4.1\n")
    outfile.write("##filedate=%s%s%s\n" % (today.year,today.month,today.day))
    outfile.write("##source=AMOS_file_JU\n")
    outfile.write("##INFO=<ID=NS,Number=1,Type=Integer,Description=\"Number of samples
    outfile.write("##INFO=<ID=DP,Number=1,Type=Integer,Description=\"Total Depth\">\n"
    outfile.write("##INFO=<ID=AF,Number=A,Type=Float,Description=\"Allele Frequency\">
    outfile.write("#CHROM\tPOS\tID\tREF\tALT\tQUAL\tFILTER\tINFO\n")

    for scaffold in sorted(snp_info):
        for position in sorted(snp_info[scaffold]):
            outfile.write(scaffold + "\t" + str(position) + "\t" + "\t".join(map(str,s


    outfile.close()
```

0.1 Store the results for global comparison among the genomes

In [6]:

```
genome_results = defaultdict(list)
```

1 J07ABHN4

In [7]:

```
#Files for J07ABHN4
genome_folder = "J07ABHN4"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HN4/2512875007.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454_Trimmed.HN4_v3_031612_readnames_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07ABHN4.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/HN4_v3_031612.posmap.ctgscf"
```

In [8]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil
```

In [9]:

```
#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HN4"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf][

    #Store the results for comparisons with the other genomes
```

```
J07HN4v2_scf7180000001347.1     547036  961
J07HN4v2_scf7180000001348.2     2341623 7958
```

In [10]:

```
#Write the VCF file
output_vcf_file = genome_folder + "/J07HN4_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

In [11]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HN4 \
$output_vcf_file > $genome_folder/J07HN4_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       1064
```

2 J07ABHN6

In [12]:

```
#Files for J07ABHN6
genome_folder = "J07ABHN6"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HN6/2512875008.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454Trimmed_HN6_031912_readnames_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07ABHN6.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/HN6_031912.posmap.ctgscf"
```

In [13]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HN6"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf][

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07HN6_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07HN6v2_scf7180000001851.4     424200  2746
J07HN6v2_scf7180000001850.3     185112  1476
J07HN6v2_scf7180000001853.6     896196  5442
J07HN6v2_scf7180000001852.5     873166  6434
J07HN6v2_scf7180000001848.1     61036   602
J07HN6v2_scf7180000001849.2     89290   553
```

In [14]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HN6 \
$output_vcf_file > $genome_folder/J07HN6_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       1969
```

3 J07ABHX64

In [15]:

```
#Files for J07ABHX64
genome_folder = "J07ABHX64"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HX64/2502082092.fna"

#SNP file, generated by amos
amos_snp_file = "./AC_sang_080509mer15_C20Q20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07ANHX64.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = "./AC_sang_080509mer15.posmap.ctgscf"
```

In [16]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07ABHX64"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[sca

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07ABHX64_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07ABHX6_J07ABscf098875 2982938 5466
```

In [17]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HX64 \
$output_vcf_file > $genome_folder/J07HX64_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       997
```

4 J07ABHX67

In [18]:

```
#Files for J07ABHX67
genome_folder = "J07ABHX67"

#Fasta file of the genome (nucleotide)
```

```
fasta_genome = "../jgi_genomes/J07HB67/2506783034.fna"

#SNP file, generated by amos
amos_snp_file = "./AC_sang_080509mer15_C20Q20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07ABHX67.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = "./AC_sang_080509mer15.posmap.ctgscf"
```

In [19]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07ABHX67"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[sca

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07ABHX67_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07ABHX67v2__Contig_2   254249  145
J07ABHX67v2__Contig_1   110024  62
J07ABHX67v2__Contig_3   2285274 2851
```

In [20]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HX67 \
$output_vcf_file > $genome_folder/J07HX67_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       1414
```

5 J07HQX

In [21]:

```
#Files for J07HQX
genome_folder = "J07HQX"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HQX/2512875009.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454Trimmed_HQX_031812_readnames_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
```

```
scaffold_list = genome_folder + "/J07HQX.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/HQX_031812.posmap.ctgscf"
```

In [22]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HQX"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf][

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07HQX_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07HQXv2_scf7180000001541.2     1476021 131
J07HQXv2_scf7180000001540.1     1543888 28
```

In [23]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HQX \
$output_vcf_file > $genome_folder/J07HQX_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       15
```

6 J07HWQ1

In [24]:

```
#Files for J07HWQ1
genome_folder = "J07HWQ1"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HQW1/2512875005.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/AC_0.8sang_080309mer15.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07HWQ1.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/AC_0.8sang_080309mer15.posmap.ctgscf"
```

In [25]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HWQ1"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf]

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07HWQ1_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07HWQ1_J07B_scf56329a.1        3475501 24677
```

In [26]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HWQ1 \
$output_vcf_file > $genome_folder/J07HWQ1_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       2339
```

7 J07HWQ2

In [27]:

```
#Files for J07HWQ1
genome_folder = "J07HWQ2"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HQW2/2512875006.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/HQW2_08err_032412a_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07HWQ2.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/HQW2_08err_032412a.posmap.ctgscf"
```

In [28]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HWQ2"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf]
    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07HWQ2_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07HQW2_scf7180000002443.1      3594539 13863
```

In [29]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HWQ2 \
$output_vcf_file > $genome_folder/J07HWQ2_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       1233
```

8 J07HR59

In [30]:

```
#Files for J07HR59
genome_folder = "J07HR59"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HR59/2512875011.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454Trimmed_HR_032012_readnames_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07HR59.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/HR_032012.posmap.ctgscf"
```

In [31]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HR59"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf]

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07HR59_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07HR59_scf7180000001381.6      184023  7
J07HR59_scf7180000001380.5      49857   29
J07HR59_scf7180000001382.7      1672266 255
J07HR59_scf7180000001326.1      72446   1
```

In [32]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HR59 \
$output_vcf_file > $genome_folder/J07HR59_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       64
```

9 J07HX5

In [33]:

```
#Files for J07HX5
genome_folder = "J07HX5"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07HX5/2512875010.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454_Trimmed.HX64_HX5_031512_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07HX5.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/HX64_HX5_031512.posmap.ctgscf"
```

In [34]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07HX5"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf][

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07HX5_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07HX5_scf7180000022092.1       2040945 2046
```

In [35]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07HX5 \
$output_vcf_file > $genome_folder/J07HX5_Assembly_snpEFF.vcf
```

```
!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       352
```

10 J07NFR43

In [36]:

```
#Files for J07NFR43
genome_folder = "J07NFR43"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07AB43/2502422326.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454Trimmed_J07NFR43_readnames_Q20C20M2.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07NFR43.scaffold_list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/J07NFR43.posmap.ctgscf"
```

In [37]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07NFR43"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07NFR43_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07NFR4_J07NFR43scf30742        52428   885
J07NFR4_J07NFR43scf30739        112863  2281
J07NFR4_J07NFR43scf30734        65032   857
J07NFR4_J07NFR43scf30737        32088   201
J07NFR4_J07NFR43scf30744        798418  7669
J07NFR4_J07NFR43scf30726        111825  701
J07NFR4_J07NFR43scf30724        54503   224
```

In [38]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07AB43 \
$output_vcf_file > $genome_folder/J07AB43_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       2174
```

11 J07NFR56

In [39]:

```
#Files for J07NFR56
genome_folder = "J07NFR56"

#Fasta file of the genome (nucleotide)
fasta_genome = "../jgi_genomes/J07AB56/2502422327.fna"

#SNP file, generated by amos
amos_snp_file = genome_folder + "/454Trimmed_J07NFR56_Q20C20.snps"

#File with the list of scaffolds. The format is:
#ID_Assembly Length ID_JGI
scaffold_list = genome_folder + "/J07NFR56.scaffold.list"

#Contig to Scaffold Mapping. The SNP file that AMOS generates, has the coordinates by
#A second file is needed to map this coordinates to the scaffolds. Format:
#Contig Scaffold Start End Orientation
ctg_scaf_map = genome_folder + "/J07NFR56.posmap.ctgscf"
```

In [40]:

```
#Generate the list of the snps, and a general count of SNPs found on each scaffold
genome_scaffolds = read_scaffold_list(scaffold_list)
mapping_info = read_ctg_scaf_map(ctg_scaf_map)
output_vcf_list,snp_count = read_amos_snp(genome_scaffolds, mapping_info, amos_snp_fil

#Print some basic stats on number of snps and length
for scaf in snp_count:
    print genome_scaffolds[scaf][0] + "\t" + str(genome_scaffolds[scaf][1]) + "\t" + s

    genome_results["J07NFR56"] = [genome_scaffolds[scaf][0], str(genome_scaffolds[scaf

    #Store the results for comparisons with the other genomes

#Write the VCF file
output_vcf_file = genome_folder + "/J07NFR56_Assembly_snps.vcf"
write_vcf(output_vcf_list, output_vcf_file)
```

```
J07NFR5_J07NFR56scf39101        959093  6809
J07NFR5_J07NFR56scf39097        60285   196
J07NFR5_J07NFR56scf39072        196424  1527
```

In [41]:

```
#Run snpEFF
!/Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java -jar ~/Bio
-no-downstream -no-upstream -no-utr -ud 0 -o vcf -c snpEff.config J07AB56 \
$output_vcf_file > $genome_folder/J07AB56_Assembly_snpEFF.vcf

!mv snpEff_summary.html $genome_folder/
!mv snpEff_genes.txt $genome_folder/
```

```
WARNINGS: Some warning were detected
Warning type    Number of warnings
WARNING_TRANSCRIPT_NO_START_CODON       3875
```

Part I

Data Summary

```
print genome_results
```

In [42]:

```
defaultdict(<type ’list’>, {’J07NFR56’: [’J07NFR5_J07NFR56scf39072’,
’196424’, ’1527’], ’J07HX5’: [’J07HX5_scf7180000022092.1’, ’2040945’,
’2046’], ’J07HN6’: [’J07HN6v2_scf7180000001849.2’, ’89290’, ’553’],
’J07HN4’: [’J07HN4v2_scf7180000001348.2’, ’2341623’, ’7958’],
’J07HR59’: [’J07HR59_scf7180000001326.1’, ’72446’, ’1’], ’J07HWQ2’:
[’J07HQW2_scf7180000002443.1’, ’3594539’, ’13863’], ’J07HQX’:
[’J07HQXv2_scf7180000001540.1’, ’1543888’, ’28’], ’J07HWQ1’:
[’J07HWQ1_J07B_scf56329a.1’, ’3475501’, ’24677’], ’J07ABHX67’:
[’J07ABHX67v2__Contig_3’, ’2285274’, ’2851’], ’J07NFR43’:
[’J07NFR4_J07NFR43scf30724’, ’54503’, ’224’], ’J07ABHX64’:
[’J07ABHX6_J07ABscf098875’, ’2982938’, ’5466’]})
```

In []: