```
Script started on Thu 27 Oct 2016 11:21:17 PM CDT
^[]0;ib501_stud12@biocluster:~/shell/Grobelny_hw6_output^G^[[?1034h^[[01;31m^B23:21:17 ^[[01;32m^Bib501_stud12 ^[[02;36m
^Bbiocluster ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_output ^[[00;33m^B^[[00m

$ cat vev^G[Kl^Gvethg_qc.py
# Hw 6
import sys
import re
import getopt
import matplotlib
matplotlib.use("Agg")  # Force matplotlib to not use any Xwindows backend.

import matplotlib.pyplot as plt

# default parameters
file_name = ""
kmer = ""
output = "velvethg_qc_out"
stat_print = 0
argv = sys.argv[1:]
try:
    opts, args = getopt.getopt(argv, "hsk:f:n:")
except getopt.GetoptError:
    print ’velvethg_qc.py -h <help> -k <kmerlength> -s <stat_print_yes> -n <output_name> -f <inputfile> \n’
    sys.exit(2)
for opt, arg in opts:
    if opt == ’-h’:
        print "#--- Velvethg_qc: Assembly Quality Script ---#\n"
        print "Usage:"
        print ’velvethg_qc.py -h <help> -k <kmerlength> -s <stat_print_yes>[0|1] -n <output_name> -f <inputfile> \n’
        print "Goals:"
        print "1) Gather kmer contig length and coverage from fasta headers"
        print "2) Output stats based on contig length and coverage"
        print "3) Output histogram of contig lengths"
        print "\n"
        sys.exit()
    elif opt in ("-k"):
        kmer = int(arg)
    elif opt in ("-s"):
        stat_print = 1
    elif opt in ("-n"):
        output = str(arg)
    elif opt in ("-f"):
        file_name = arg
print "Input file:", file_name
print " "

fh = open(file_name, ’r’)

# count variables
num_contigs = 0
contig_length_data = []
contig_cov_data = []

# pre compile regex pattern
regex_pat = re.compile(r’^>NODE_\d+_length_(\d+)_cov_(\d+\.\d+)’)

#print "Calulating Assembly Quality Stats... \n"
# loop to collect kmer length and cov --> append each variable to its own list
for line in fh:
    line = line.strip(’\n’)
    if line[0] == ">":
        contig_data = re.findall(regex_pat, str(line))
        kmer_len, kmer_cov = contig_data[0]
        # convert to contig physical length
        contig_nuc_len = int(kmer_len) + (kmer - 1)

        # add contig length data to list
        contig_length_data.append(int(contig_nuc_len))

        # add contig cov data to list
        contig_cov_data.append(float(kmer_cov))

        # add one to contig count
        num_contigs += 1
fh.close

# Distribution of contigs
# Calculate the distribution of contig lengths,and bucket the contig lengths
# into groups of 100bp. So, all contigs with lengths between 0 and 99 would be
# in the 0 bucket, those with lengths between 100 and 199 would be in the 100 bucket
contig_len_dic = {}

for contig_len in contig_length_data:
    # Bin data
    bin_group = int(round(contig_len / 100)) * 100
    contig_len_dic[bin_group] = contig_len_dic.get(bin_group, 0) + 1

# sort data
contig_length_data_sorted = sorted(contig_length_data)

if stat_print == 1:
    print "#--- Velvethg_qc: Assembly Quality Stats ---#\n"
    print "Stats for Assembly:\t", file_name

    # -the number of contigs
    print "Number of contigs:\t", num_contigs

    # -the maximum contig length
    max_contig = contig_length_data_sorted[-1]
    print "Max contig length:\t", max_contig

    # -the mean contig length
    sumed_contig_length_data_sorted = sum(contig_length_data_sorted)
    print "Mean contig length:\t", float(sumed_contig_length_data_sorted) / float(num_contigs)

    # -total length of the genome across all the contigs.
    print "Total length of the genome across all contigs:\t", sumed_contig_length_data_sorted

    # -mean depth of coverage for the contigs
    print "Mean depth of coverage:\t", float(sum(contig_cov_data)) / float(num_contigs)

    # -N50 value of your assembly
    print "N50 of assembly:\t", sum(contig_length_data_sorted[(int(num_contigs) / int(2)):-1])
else:
    print "Stat print is off, but still printing graph..."


#print "\n#--- Velvethg_qc: Contig Length Histogram ---#\n"
print "Contig Length\tNumber of Contigs in this category"

# printing histrogram of contig lengths
for key in sorted(contig_len_dic.keys()):
    print "%s\t%s\n" % (key, contig_len_dic[key])

# Plot contig length distribution
plt.bar(contig_len_dic.keys(), contig_len_dic.values())

# Add labels
plt.xlabel("Contig Size (bps)")
plt.ylabel("Counts")
plt.xscale(’log’)
plt.yscale(’log’)
plt.title("Distribution of contigs")
plt.grid(True)

#print "\nSaving Plot of: %s.png" % (output)
# Save graph
plt.savefig("%s.png" % (output))
plt.close()

^[]0;ib501_stud12@biocluster:~/shell/Grobelny_hw6_output^G^[[01;31m^B23:21:28 ^[[01;32m^Bib501_stud12 ^[[02;36m^B
ter ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_output ^[[00;33m^B^[[00m

$ cat H^Gw6_kmer31.sh
#!/bin/bash -x

# ----------------QSUB Parameters----------------- #
#PBS -S /bin/bash
#PBS -q classroom
#PBS -l nodes=1:ppn=8,mem=12GB
#PBS -N Velvet31_matt
# ----------------Load Modules-------------------- #
#module load python/2.7.9
module load velvet
# ----------------Your Commands------------------- #

file1="/home/classroom/ib501/assembly/samples/rs_female_1983.13.1.fil.fq.gz"
file2="/home/classroom/ib501/assembly/samples/rs_female_1983.13.2.fil.fq.gz"

directory="/home/a-m/ib501_stud12/shell/Grobelny_hw6_output"
outfile="/velvetout_kmer_"
opts="_opts_"
min_contig_len_string="min_contig_len_500"

kmer="31"
options="-shortPaired -fastq.gz"
cov=58
((ck=$cov*(100-$kmer+1)/100))

options="-shortPaired -fastq.gz"
mkdir $directory

# compute assembly standard parameters output
dir_out_name=$directory$outfile$kmer
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -ins_length 500 -exp_cov $ck

# compute assembly standard parameters output w/ min contig len at 500
dir_out_name=$directory$outfile$kmer$opts$min_contig_len_string
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -min_contig_lgth 500 -ins_length 500 -exp_cov $ck

^[]0;ib501_stud12@biocluster:~/shell/Grobelny_hw6_output^G^[[01;31m^B23:21:35 ^[[01;32m^Bib501_stud12 ^[[02;36m^B
ter ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_output ^[[00;33m^B^[[00m

$ cat H^Gw6_kmer41.sh
#!/bin/bash -x

# ----------------QSUB Parameters----------------- #
#PBS -S /bin/bash
#PBS -q classroom
#PBS -l nodes=1:ppn=8,mem=10GB
#PBS -N Velvet41
#PBS -k oe
# ----------------Load Modules-------------------- #
#module load python/2.7.9
module load velvet
# ----------------Your Commands------------------- #
```

```
file1="/home/classroom/ib501/assembly/samples/rs_female_1983.13.1.fil.fq.gz"
file2="/home/classroom/ib501/assembly/samples/rs_female_1983.13.2.fil.fq.gz"

directory="/home/a-m/ib501_stud12/shell/Grobelny_hw6_output"
outfile="/velvetout_kmer_"
opts="_opts_"
min_contig_len_string="min_contig_len_500"

kmer="41"
cov="58"
((ck=$cov*(100-$kmer+1)/100))

options="-shortPaired -fastq.gz"
mkdir $directory

# compute assembly standard parameters output
dir_out_name=$directory$outfile$kmer
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -ins_length 500 -exp_cov $ck

# compute assembly standard parameters output w/ min contig len at 500
dir_out_name=$directory$outfile$kmer$opts$min_contig_len_string
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -min_contig_lgth 500 -ins_length 500 -exp_cov $ck

^[]0;ib501_stud12@biocluster:~/shell/Grobelny_hw6_output^G^[[01;31m^B23:21:45 ^[[01;32m^Bib501_stud12 ^[[02;36m^Bbioclus
ter ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_output ^[[00;33m^B^[[00m

$ cat H^Gw6_kmer49^G.sh
#!/bin/bash

# ----------------QSUB Parameters----------------- #
#PBS -S /bin/bash
#PBS -q classroom
#PBS -l nodes=1:ppn=8,mem=10GB
#PBS -N Velvet49
#PBS -k oe

# ----------------Load Modules-------------------- #
#module load python/2.7.9
module load velvet
# ----------------Your Commands------------------- #

file1="/home/classroom/ib501/assembly/samples/rs_female_1983.13.1.fil.fq.gz"
file2="/home/classroom/ib501/assembly/samples/rs_female_1983.13.2.fil.fq.gz"

directory="/home/a-m/ib501_stud12/shell/Grobelny_hw6_output"
outfile="/velvetout_kmer_"
opts="_opts_"
min_contig_len="500"
min_contig_len_string="min_contig_len_500"

kmer="49"
options="-shortPaired -fastq.gz"
cov="58"
((ck=$cov*(100-$kmer+1)/100))
mkdir $directory

# compute assembly standard parameters output
dir_out_name=$directory$outfile$kmer
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -ins_length 500 -exp_cov $ck

# compute assembly standard parameters output w/ min contig len at 500
dir_out_name=$directory$outfile$kmer$opts$min_contig_len_string
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -min_contig_lgth 500 -ins_length 500 -exp_cov $ck

^[]0;ib501_stud12@biocluster:~/shell/Grobelny_hw6_output^G^[[01;31m^B23:21:52 ^[[01;32m^Bib501_stud12 ^[[02;36m^Bbioclus
ter ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_output ^[[00;33m^B^[[00m

$ cat Hw6_kmer49.sh ^H^[[K^H^[[K^H^[[K^H^[[K_^G^H^[[K^G
Hw6_kmer49_opts.sh  Hw6_kmer49.sh       Hw6_kmer49_test.sh
^[[01;31m^B23:21:52 ^[[01;32m^Bib501_stud12 ^[[02;36m^Bbiocluster ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_o
utput ^[[00;33m^B^[[00m

$ cat Hw6_kmer49_opts.sh
#!/bin/bash

# ----------------QSUB Parameters----------------- #
#PBS -S /bin/bash
#PBS -q classroom
#PBS -l nodes=1:ppn=8,mem=10GB
#PBS -N Velvet49_opts
#PBS -k oe

# ----------------Load Modules-------------------- #
#module load python/2.7.9
module load velvet
# ----------------Your Commands------------------- #

file1="/home/classroom/ib501/assembly/samples/rs_female_1983.13.1.fil.fq.gz"
file2="/home/classroom/ib501/assembly/samples/rs_female_1983.13.2.fil.fq.gz"

directory="/home/a-m/ib501_stud12/shell/Grobelny_hw6_output"
outfile="/velvetout_kmer_"
opts="_opts_cutoff_"
```

```
min_contig_len="500"
min_contig_len_string="min_contig_len_500"

kmer="49"
options="-shortPaired -fastq.gz"
cov="58"
((ck=$cov*(100-$kmer+1)/100))

#compute assembly with additional cov cutoff parameters cov_cutoff at 4x
kmer_opt="4"

# compute assembly standard parameters output
dir_out_name=$directory$outfile$kmer$opts$kmer_opt
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -cov_cutoff $kmer_opt -ins_length 500 -exp_cov $ck

# compute assembly standard parameters output w/ min contig len at 500
dir_out_name=$directory$outfile$kmer$opts$kmer_opt$min_contig_len_string
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -min_contig_lgth 500 -cov_cutoff $kmer_opt -ins_length 500 -exp_cov $ck

#compute assembly with additional cov cutoff parameters cov_cutoff at 8x
kmer_opt="8"

# compute assembly standard parameters output
dir_out_name=$directory$outfile$kmer$opts$kmer_opt
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -ins_length 500 -cov_cutoff $kmer_opt-exp_cov $ck

# compute assembly standard parameters output w/ min contig len at 500
dir_out_name=$directory$outfile$kmer$opts$kmer_opt$min_contig_len_string
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -min_contig_lgth 500 -cov_cutoff $kmer_opt -ins_length 500 -exp_cov $ck

#compute assembly with additional cov cutoff parameters cov_cutoff set to auto
kmer_opt="auto"

# compute assembly standard parameters output
dir_out_name=$directory$outfile$kmer$opts$kmer_opt
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -ins_length 500 -cov_cutoff $kmer_opt -exp_cov $ck

# compute assembly standard parameters output w/ min contig len at 500
dir_out_name=$directory$outfile$kmer$opts$kmer_opt$min_contig_len_string
mkdir $dir_out_name
velveth $dir_out_name $kmer $options $file1 $file2
velvetg $dir_out_name -min_contig_lgth 500 -cov_cutoff $kmer_opt -ins_length 500 -exp_cov $ck

^[]0;ib501_stud12@biocluster:~/shell/Grobelny_hw6_output^G^[[01;31m^B23:22:01 ^[[01;32m^Bib501_stud12 ^[[02;36m^B
ter ^[[01;34m^B/home/a-m/ib501_stud12/shell/Grobelny_hw6_output ^[[00;33m^B^[[00m

$ exit
exit

Script done on Thu 27 Oct 2016 11:22:04 PM CDT
```