Supplemental File 1:

```
# Step: Creating bit-array of k-mers from the assembled genome

#input file - longipalpis_both_v3.fasta

jellyfish count -v 1 -m 16 -o longipalpis_v3.jelly -c 4 -s 10G -t 2 --both-strands longipalpis_both_v3.fasta


jellyfish dump --lower-count=2 longipalpis_v3.jelly | gzip -c > longipalpis_v3.vector_rep.gz


##########################

# Step: Creating bit-array of k-mers from the female reads filtered by sequence quality and frequency

#input file - females reads - LUL_f_1.fastq.gz/LUL_f_2.fastq.gz

gunzip -c LUL_f_*.fastq.gz.gz | \

jellyfish count -v 1 -m 16 -c 4 -s 10G -t 6 -o longipalpis_fe.jelly --both-strands --quality-start=33 --min-quality=20 /dev/fd/0


jellyfish dump --lower-count=3 longipalpis_fe.jelly | gzip -c > longipalpis_female.vector_rep.gz


#Creating bit-arrays: output file produced with the "to_Hex" function of the Bit::Vector

module (http://search.cpan.org/dist/Bit-Vector). See its manual for details.

perl YGS.pl kmer_size=16 mode=trace trace= longipalpis_female.vector_rep.gz


##########################

# Step: Creating bit-array of k-mers from the male reads filtered by sequence quality and frequency

#input file - males reads - files: LUL_m_1.fastq.gz/LUL_m_2.fastq

gunzip -c LUL_m_*.fastq.gz.gz | \

jellyfish count -v 1 -m 16 -c 4 -s 10G -t 6 -o longipalpis_ma.jelly --both-strands --quality-start=33 --min-quality=20 /dev/fd/0


jellyfish dump --lower-count=3 longipalpis_ma.jelly | gzip -c > longipalpis_male.vector_rep.gz


#Creating bit-arrays: output file produced with the "to_Hex" function of the Bit::Vector

perl YGS.pl kmer_size=16 mode=trace trace= longipalpis_male.vector_rep.gz


##########################

# Step: Creating bit-array of k-mers from all reads filtered by sequence quality and frequency for k-
```

```
 mer validation
#input file - all reads - files: LUL_m_1.fastq.gz/LUL_m_2.fastq/LUL_f_1.fastq.gz/LUL_f_2.fastq.gz
gunzip -c  LUL_*.fastq.gz.gz | \
jellyfish count -v 1-m 16 -c  4 -s 10G -t 6    -o longipalpis_validation.jelly    --both-strands --quality-
 start=33 --min-quality=20 /dev/fd/0

jellyfish dump --lower-count=3 longipalpis_validation.jelly | gzip -c >
 longipalpis_validation.vector_rep.gz

#Creating bit-arrays:  output file produced with the "to_Hex" function of the Bit::Vector
perl YGS.pl  kmer_size=16  mode=trace  trace= longipalpis_validation.vector_rep.gz
#output: binary file

##################
#step= "IDENTIFYING THE Y_SUSPECTS : *.final_result step   ( YGS.pl program)"

# Test for the XY system (see text for more detail): it uses the binary bit-arrays files
perl YGS.pl   kmer_size=16  mode=final_run  \
contig=longipalpis_both_v3.fasta   \
gen_rep=longipalpis_v3.trace.gz   \
trace=longipalpis_female.trace.gz     \
male_trace=longipalpis_validation.trace.gz
#note that there is a variable called "male_trace" that cames from the original paper (Carvalho & Clark,
 2013) but does not denote male traces in the current work.
#It denotes the bitarray of valid k-mers from both male and female reads.

# Test for the ZW system (see text for more detail)
perl YGS.pl   kmer_size=16  mode=final_run  \
contig=longipalpis_both_v3.fasta   \
gen_rep=longipalpis_v3.trace.gz   \
trace=longipalpis_male.trace.gz     \
male_trace=longipalpis_validation.trace.gz
#note that there is a variable called "male_trace" that cames from the original paper (Carvalho & Clark,
 2013) but does not denote male traces in the current work.
#It denotes the bitarray of valid k-mers from both male and female reads.
```