High level CRISPR spacer screen steps

1. Identify which contig a CRISPR Cas operon is on within an isolate assembly, and extract that contig

2. Label each contig with isolate name and concatenate into a multifasta. Feed into tool **CrisprCasFinder**

3. Extract from CrisprCasFinder all spacers found within each Cas operon for each isolate. (On average ~15/operon for SIG)

4. Obtain the reverse complement of each spacer using online tool. Label appropriately and add to dataset (should double in size). (Do this bc our alignment tool will not consider directionality when aligning two sequence strings)

5. Some spacers will be repeated within a Cas operon. Remove this to avoid self-matches by concatenating all spacers for a given Cas operon into a multifasta, and removing duplicated spacer entries. Do this for all Cas operons

6. With these removed, concatenate all spacers in dataset and run **clustalo** with –percent-id flag (and others) to get a giant all-v-all spacer ANI matrix

7. In rstudio, convert to long format and delete all self-comparisons within this ANI matrix. Make a new dataframe only with comparisons of ANI=100%

8. Now the aim is to create a sheet where each row lists spacers that are identical to each other and only each other. From this we can use an excel formula to create a network co-occurrence matrix. To do this, we need to reduce the dataframe from step 7 (currently looks like: spacer 1 | spacer 2 | ANI). We need to remove two cases:

    a. If (a,b,100) is present, remove (b,a,100) later in the sheet

    b. If (a,b), (a,c), (a,d), (b,c), (b,d) are present, we know a=b=c=d (and want just 1 row of a, b, c, d). However, our current conversion method will result in 3 rows of (a,b,c,d), (b,c,d), and (c,d). We need to remove (b,c),(b,d) from this dataframe since all spacers a,b,c,d are represented just by (a,b),(a,c),(a,d)

    c. **Bejan** wrote 2 python scripts to do this. Run those on cluster

9. The output of Bejan’s second script will result in data formatted in the desired manner where each row lists only spacers that are identical to each other. Convert this to a isolate-vs-isolate co-occurrence matrix in excel using Excel formula

10. In rstudio, convert to long format (isolate1 | isolate 2 | shared spacer count | percent of total spacers shared) so it can be used as input for ggplot2 bubbleplot visualization. Run ggplot to viz

Steps to remove repeated spacers within each isolate/cluster's total CRISPR spacer set

Linux

1. Import all spacers into cluster as a concatenated multifasta file (vim copy/paste)
2. Split into individual files using
   a.
   ```
   awk 'BEGIN {n_seq=0;} /^>/
   {if(n_seq%1==0){file=sprintf("myseq%d.fa",n_seq);} print >> file;
   n_seq++; next;} { print >> file; }' < cat_multifasta.fasta
   ```
3. Change names from myseq%.fa to first line using
   a. ls myseq* > ../temp.txt
   b. for i in `more ../temp.txt`; do mv ${i} "$(head -1 ${i}).fasta"
   c. Delete first character in title (>) using rename -n ‘s|>||’ *fasta
   d. rm ../temp.txt
4. Create new file of all unique isolates/clusters (vim copy/paste) and create a multifasta file for each isolate/cluster
   a. for i in `more ../identifier.txt`; do cat ${i}* > ../new_dir/${i}.fasta
5. Remove all duplicate spacer sequences within each isolate/cluster multifasta file using
   a. awk '!_[$0]++' ${i}.fasta > ../new_dir/${i}_noduplicates.fasta
   b. rm ../identifier.txt
6. Now need to remove headers without spacer sequences in each file
   a. Use the awk command from step #2 to create individual fasta files for each spacer sequence (or lack thereof)
   b. Change names of each individual fasta file as in step #3 but DO NOT DELETE >
   c. Remove first line (the title) of each individual fasta file using
      i.
      ```
      sed '1d' example.fasta >
      ../new_dir/identifier_nofirstline.fasta
      ```
   d. All files with repeated spacer sequences should now be empty. To remove all empty files within a directory, use
      i.
      ```
      find . -type f -empty -print -delete
      ```
      ii. This will also print all the files that were removed into the command line
7. Now that all headers and sequences of repetitive spacers within a cluster/isolate have been removed, just need to reformat the remaining files
   a. Add back the header by copying filename to first line of file using
      i. awk -i inplace -v ORS='\r\n' 'FNR==1{print FILENAME}1' *

   b. Now remove the > from the file name using rename -n ‘s|>||’ *fasta

8. Concatenate everything remaining into a final multifasta.file
9. Run clustalo. Tab separate pairwisedistance.txt file using
   a. `sed 's/[[:blank:]]\+/\t/g' input.txt > output.txt`

Rstudio

1. Read in Tab_separated_distancematrix.txt file and convert to long format (melt)
   a. df_spacer_all_RC <- read.delim('all_spacers_clusters_noduplicates_TAB_SEPARATED_distancematrix.txt', header = T)
   b. df_spacer_all_RC_long <- melt(df_spacer_all_RC)
2. Add column names
   a. colnames(df_spacer_all_RC_long) <- c("spacer1","spacer2","ANI")
3. Delete all self-comparisons
   a. df_spacer_all_RC_long<- df_spacer_all_RC_long[!(df_spacer_all_RC_long$spacer1==df_spacer_all_RC_long$spacer2),]
4. Make new dataframe with only comparisons of ANI=100%
   a. df_spacer_all_RC_long_ani100<- df_spacer_all_RC_long[(df_spacer_all_RC_long$ANI==100),]
5. Remove colnames, write as txt file and import into cluster
   a. names(df_spacer_all_RC_long_ani100)<-NULL
   b. write.table(df_spacer_all_RC_long_ani100,"220807_df_spacer_all_long_ani100.txt",sep = "\t",row.names = FALSE)
6. **Cluster:** Remove lines that are the inverse of previous lines (if [a,b] already listed, remove [b,a]). Import back to rstudio
   a. Run s16.9.5_distancematrix_remove_ab_ba.sh
7. **Back in rstudio:** Read in output from python script
8. Remove “_RV” from all identifiers (the next steps will clean up duplicate spacer comparisons normal-RV vs. RV-normal)
   a. df_spacer_all_ani100_filtered$V1<- gsub("_RV","",as.character(df_spacer_all_ani100_filtered$V1))
   b. df_spacer_all_ani100_filtered$V2<- gsub("_RV","",as.character(df_spacer_all_ani100_filtered$V2))
9. Remove duplicate rows. This will remove [a,b] when [a,b] already exists (results from removing _RV)
   a. df_spacer_all_ani100_filtered <- df_spacer_all_ani100_filtered[!duplicated(df_spacer_all_ani100_filtered),]
10. Remove colnames, write txt file and import back into cluster
   a. names(df_spacer_all_ani100_filtered)<-NULL
   b. write.table(df_spacer_all_ani100_filtered,"220807_df_spacer_all_long_ani100_filtered_cleaned.txt",sep = "\t",row.names = FALSE)

11. **Cluster:** Iterate through all connections and reduce (if [a,b] [a,c] [a,d] [b,c] [b,d] [c,d] are all present, we want to reduce dataset to just [a,b] [a,c] [a,d]). This is bc we want to end with a b c d all in one line, instead of a b c d /n b c d /n c d
    a. Run: s16.9.6_distancematrix_remove_allconnections.sh

Excel

1. Create a co-occurrence matrix from the above python script using formula
    a. Formula:
2. **Back in rstudio:** Convert to long format using melt
3. **Back in Excel:** Now have the following columns: Isolate1,Isolate2,Shared_spacer_ct
4. Add in Isolate1_totalspacers,Isolate2_totalspacers, calculate Percent_shared with Isolate1_totalspacers as the denominator

Links used here

1. Split a multifasta into individual fasta files: https://www.biostars.org/p/13270/
2. Copy first line of a file into the file name: https://stackoverflow.com/questions/42782460/rename-a-single-file-using-the-first-line-in-a-txt-file
3. Delete repeated lines within a file: https://www.unix.com/shell-programming-and-scripting/146404-command-remove-duplicate-lines-perl-sed-awk.html
4. Remove first line of a file: https://www.baeldung.com/linux/remove-first-line-text-file
5. Delete empty files in a directory: https://www.baeldung.com/linux/delete-empty-files-dirs#:~:text=First%2C%20search%20all%20the%20empty,to%20delete%20all%20those%20files.
6. Insert file name before first line: https://stackoverflow.com/questions/46496037/inserting-the-filename-before-the-first-line-of-a-text-file