**Organizing Data and Analyses**

1. Create base directories to separate sequencing, annotation, Mothur, metadata, and other analyses.
   1. The idea is keep data that will need to be analyzed independently separated. This makes it easier to identify data for new analyses, without having to look through directories where analyses have been performed (by someone else). It also makes it easier to delete analyses without worrying about deleting input data, and allows multiple sibling directories from the same analysis (with alternate parameters) to be grouped together since no single analysis directory contains the original data.
   2. Example: hierarchical\_processing/source/make\_root\_directories.csh
   3. This script will create some basic directories
2. Metadata directory
   1. The metadata directory should contain a master meta data table from which analysis specific metadata can be extracted.
      1. The original (.xlsx) spreadsheet should be kept in a "Original" directory nearby.
   2. The metadata column headers will be used as the variable and/or group names for downstream analysis. These names will often appear in the plots of the analyses, so they need to be short yet meaningful.
      1. Keep names and cases consistent and spelled correctly
      2. For Boolean variables, use is\_<Condition>, for example is\_HIVPos, or is\_Female.
      3. If a boolean variable based on a continuous variable, then include the cutoff, for example is\_Hot\_gte80 (is Hot when temperature is greater than or equal to 80), versus is\_Cold\_lt32 (is Cold when temperature is less than 32).
3. Copy complete summary tables into **AllSamples**
   1. These should contain all samples, including control, failed, different extraction methods, etc.
   2. These will be the source of where all subsets and combination (e.g. replicates) will be performed.
   3. Derived summary tables can be put in a sibling directory, and generated based on the metadata information.
      1. Merge\_or\_Subset\_Samples\_by\_Table.r: This will merge or remove samples based on the specified table. When samples are merged, a new sample ID will need to be specified. This will ensure that there is no confusion between the merged and pre-merged samples.
      2. Split\_SummaryTableSamples\_byGroupTable.r: This will split a summary table into multiple summary tables based on the group table. The group table will specify which samples should belong in each group. The sample names will not change.
      3. ColumnTools/extract\_by\_key\_column.pl: This will allow you to extract only the columns of necessity from an original table. You specify the column that want to use as a key and the columns you want to extract. Files created from extracting from the original meta data table should be kept with the analyses that required the extraction.
   4. These tables should be backedup each time before they are edited.
   5. Use concise, consistent, thoughtful, accurate names for groups and grouping names. Remember these sample names may end up in plots so they should capture information about the samples, but not be so long as to look amateurish in a publication.
   6. When deriving new summary tables, if the number of samples does not change then it should be a child directory under a parent. For example, if the categories get renamed, reduced, or filtered, then this should increase the depth of the tree, since the number of samples in the summary file does not change. However, if new samples are created by collapsing replicates or deleting by subsetting controls or failed runs, then the tree should increase in breadth at the root. Increasing depth will tend to be manual, but increasing breadth can use the existing tools to clone and resubset what is in the original. All breadth/sample manipulations should be performed with code and based on the metadata to ensure readability, reproducibility and documentation.
   7. When generating new data subsets, write/save the commands in a .csh file so we will have a record of what the parameters used were, for example, run\_cntl\_exp\_extractions.csh

**Executing Count-based Analyses**

Once the directories for summary tables have been organized, there is code to execute the analyses on every summary table in a directory. If there is more than one summary table in a directory, it is assumed to contain multiple groups, and if a two (comparison) summary table analysis is applied, all unique pairwise combinations will be executed.

1. For each analysis type, there should be a separate directory. Within each analysis type directory, multiple versions can be contained. Different version may refer to different cutoffs or different models. They do not refer to different datasets. There are only differences in the parameters. By looking in an analysis directory, it should be apparently how many variations of the analysis have been applied to all the data.
2. The key steps are:
   1. Specify a data and analysis root directory, and the directory structure underneath the data directory will be recreated under the analysis directory, along with symbolic links to the original summary tables
      1. site\_comparison/hierarchical\_processing/source/clone\_target\_subdirectory\_and\_link.pl
   2. Single Summary\_Table Analyses:
      1. Run on all \*.summary\_table.tsv (or any other extension) file
      2. site\_comparison/hierarchical\_processing/source/individual/execute\_on\_targets\_individual.pl
      3. Takes ${TARGET\_FILE} and ${RESULT\_FILE} in template shell script
   3. Two Summary\_Table Analyses:
      1. If there is more than one summary\_table in directory, perform pairwise comparison among all combinations
      2. site\_comparison/hierarchical\_processing/source/paired/execute\_on\_targets\_paired.pl
      3. Takes ${TARGET\_FILE\_1}, ${TARGET\_FILE\_2}, ${RESULT\_FILE} in template shell script
3. The template shell script specifies the command to be run, the single and two summary table analyses will automatically invoke the template shell script. The template shell script should be saved for each analyses, because it will contain the parameters that were used in the analyses as they were applied to all the summary tables.
4. Side Note: To avoid generating distance matrices for each analysis run in Permanova, the summary tables can be cloned/linked and then the computation for distance matrices can be applied using the same code used to run single analyses.