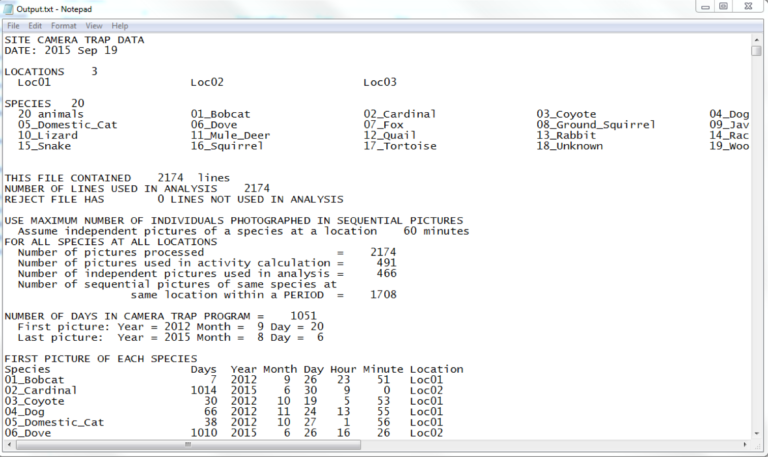
SANIMAL – Scientific Animal Image Analysis

David Slovikosky

**Background:**

In May 2016 I was made aware that the School of Natural Resources and the Environment (*SNRE*) department at the University of Arizona was using outdated software to analyze images taken by camera traps. These motion detector-equipped cameras are set up in the wild and are left to take pictures for many months. After being retrieved, the images are collected and sorted by various researchers from high school students, to university professors, to retired people who do it as a hobby. Images are sorted or *tagged* by hand with GPS location, species, and number of animals. These researchers then aggregate their data into collections which are manually stored on hard drives or servers. Finally, analysis software is run on the tagged data to produce meaningful output used for research.

Originally this task was accomplished with a suite of FORTRAN programs written by Dr. Jim Sanderson. The heart of this software suite is *DataAnalyze*, a program that takes sorted data as input and creates a large text file as output. This *output.txt* file contains thousands of lines of summary statistics on the data in the form of ASCII charts which is heavily used by the SNRE department. A sample screenshot of this output is shown below:



This software has been used for many years and is currently kept up-to-date by Dr. Sanderson. Updates are released as researchers find bugs in the existing calculations or define a new analysis they would like to see. Unfortunately, this software leaves much to be desired. More specifically, *DataAnalyze* is lacking in the following areas:

1. The source code is in FORTRAN and not easily available for custom modification
2. The sorting process for data is very tedious
3. Command line execution of the program can easily crash if the inputs are incorrectly formatted without any error description of what went wrong
4. Output is in text format only and can be challenging to parse
5. Analysis can only be performed on local data making it difficult to collaborate

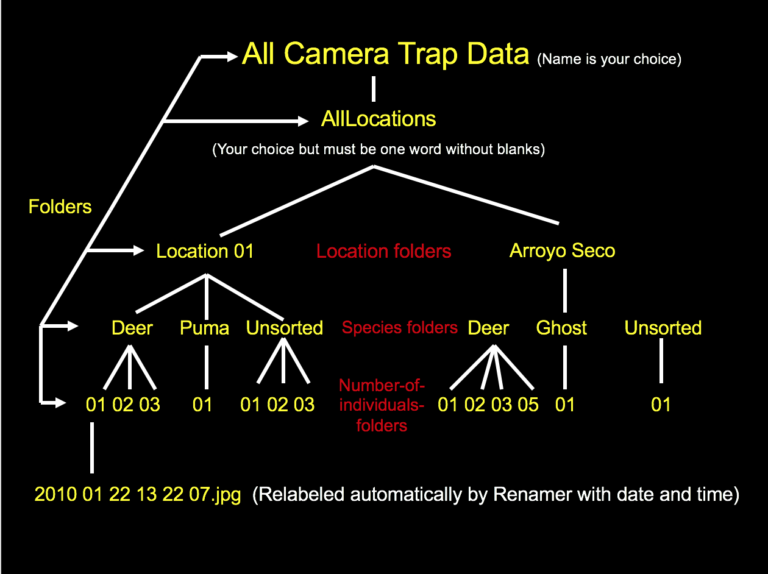
***DataAnalyze* Deficiencies:**

*(1) DataAnalyze* is written in FORTRAN and the source code is currently private. While FORTRAN was a cutting-edge language at its time, it has aged and does not provide many features standard in modern languages. Most importantly, it is challenging to create a user interface in FORTRAN which is essential in modern applications. Because the software is closed source, any modification to *DataAnalyze* is impossible. This means there is no way to see how any of the analysis are done besides the documentation given by Dr. Sanderson. While this documentation does give some insight on how the numbers are computed, it can be difficult to reverse engineer many of the numbers.

(2) As previously stated, images taken on camera traps must be *tagged* with location, species, and animal count. Locations are comprised of a name and UTM coordinates. Species are given by a name, and finally count specifies the number of the species in the image. The image date and time taken must also be extracted from the image metadata. *DataAnalyze* performs this *tagging* process by parsing a directory structure as well requiring an external tool to extract image date and time taken. The *tagging* process begins by renaming all images in the following format: “Year Month Day Hour Minute Second.jpg” by a third-party program named “Renamer”. This software simply reads the image file metadata and renames the file accordingly. All credit for this program is found below:

https://www.den4b.com/products/renamer

Renamed images are then copied into a directory structure pictured below (taken from Dr. Sanderson’s HowTo slides):



This directory structure is then parsed by *DataAnalyze*. Location UTM coordinates are read in from a separate file which is provided as a standard-in argument. These coordinates must be in a very specific format to be parsed correctly, otherwise the FORTRAN software will crash. After a few seconds of execution, *DataAnalyze* creates a few output text files, of which *output.txt* contains most of the analysis. This entire process of renaming the images and then copying them into a multi-layer directory structure is incredibly tedious and lacks scalability. Images that contain multiple species are duplicated in the directory structure leading to data redundancy.

(3) *DataAnalyze* is a Windows executable which is run from the command line. This immediately poses a problem to anyone using a Mac or Unix based system, because they simply cannot use the software. Users must open a terminal and answer questions asked by the software such as, “Enter folder name containing all camera trap locations, then ENTER”. There are many challenges when using command line applications especially to users who lack technical knowledge, and many of the users using *DataAnalyze* have never used a command line-based program. These challenges include understanding relative vs absolute file paths, remembering capitalization matters, and being able to interpret error messages. *DataAnalyze* does not provide any detection for user error and can often crash as a result with a cryptic FORTRAN error message. All these specifics mean the average user cannot easily learn how to use the software and needs to be taught by someone who has used the software.

(4) The output created by *DataAnalyze* is simple ASCII. This output is incredibly difficult to parse, both by humans and computers. Many researchers want to do further analysis on the output created by *DataAnalyze*, but due to the irregular nature of the text writing a script to read all the values can be very time consuming. Output lines are printed with many arbitrary restrictions, such as location and species names are capped around 26 characters and padded with spaces if too short. This string length value varies from chart to chart. Some charts are very wide due to having many columns and *DataAnalyze* splits these charts in two with half the data in the first, and half in the second. After talking with some researchers who use this software, one said “We just copy the ASCII charts straight into Microsoft excel and let excel’s built in parser parse the chart’s values into cells. We then graph the data using excel.” In one small study, 2,500 images were analyzed which resulted in an *output.txt* file 3000 lines long. Many of the charts featured in the output contained the same data but aggregated in many different possible permutations. As an example, some charts group images taken by month, and others group images by season. This results in two separate ASCII charts which are scattered somewhere in the 3000 lines of output. In larger studies, millions of images are often involved which leads to *output.txt* files that are incredibly large. The output of *DataAnalyze* is static and non-interactive. This makes the analysis software very difficult to use for anyone untrained. The SNRE department has a special class each year that is designed to teach citizen scientists how to understand and parse the *output.txt* file which should never be the case with analysis software.

(5) Researchers rarely work alone. There are several different projects currently going on in the SNRE department, each with a unique set of researchers, cameras, and species to track. These projects want to store all their images in a single location where analysis can be done over data from all researchers. Because *DataAnalyze* only runs on local data, one massive, terabyte sized directory structure is stored on a server. Citizen scientists first sort their portion of the data and upload it to the server. The folders are merged, and the image collection is updated. Later a researcher can execute the software on the directory. This system is flawed for many reasons. Firstly, researchers who may live around the world need to “upload” their images to the server by sending USB drives in the mail to someone who works near to the server or upload their data to a Dropbox style service. Secondly, this setup completely disregards data security. Many projects need their images to be kept private, because hunters could abuse knowledge of species movements to track specific species.