

Data Development and Analysis Pathways for Marine Mammals and Turtles: Creating a User Interface

Sarina Fernandez^{1,2}, Warren Asfazadour^{2,3}, Eric Archer⁴ and Lisa Komoroske⁴

¹ California Polytechnic State University, San Luis Obispo, CA ² STAR Program, San Luis Obispo, CA ³ Boston University, Boston, MA ⁴ NOAA Southwest Fisheries Science Center, La Jolla, CA

Introduction

- Researchers are in need of user-friendly tools to help streamline data analysis for ongoing genomics projects.
- Using shiny⁴, an R package that provides a framework for developing web applications based on R functions, we created a web application for strataG^{1,2}, an R package containing functions for summarizing genetic data and analyzing population structure.

Creating a shiny User Interface

The user interface script controls the layout of the app.

The first step of creating a shiny user interface is loading the shiny package.

```
library(shiny)
shinyUI(fluidPage()
  tabsetPanel(
    tabPanel("gtypes",
      titlePanel("gtypes object"),
      helpText("This application will allow you to analyze your data with functions from the R package strataG. To use these functions, you must first create a gtypes object out of your data"),
      helpText("If you already have a gtypes object rdata file, you may load it here, use the load button to complete the upload and display your object in the gtypes tab below."),
      fileInput("gtypesR", label = "Choose Rdata File", accept = c(".rds", ".rda", ".rdata")),
      helpText("If you do not yet have a gtypes object, input your data below to create a new gtypes object. Click the Load gtypes object button at the end of the input panel to make your gtypes object appear in the gtypes tab."),
      br(),
    )
  )
)
```

Some important general commands include:

`fluidPage`: allows the app to adjust automatically to the user's browser window size

`helpText`: provides users with general directions for using the app

Widgets are web elements that users interact with in order to communicate with the app. Here are some examples of widgets and the code that creates them within this app:

File inputs allow users to upload their raw data.

```
fileInput("genetic_data", "Choose a .csv file of your genetic data",
  accept=c("text/csv",
    "text/comma-separated-values,text/plain",
    ".csv"))
Choose a .csv file of your genetic data
Choose File No file selected
```

Checkboxes allow users to select specific functions to run.

```
checkboxInput("checkbox.by.sample", "by sample summary"),
checkboxInput("checkbox.by.locus", "by locus summary"),
checkboxInput("checkbox.dup.gen", "Duplicate genotypes"),
checkboxInput("checkbox.by.sample", "by sample summary"),
checkboxInput("checkbox.by.locus", "by locus summary"),
checkboxInput("checkbox.dup.gen", "Duplicate genotypes")
```

Action buttons allow users to run functions and save outputs.

```
actionButton("run_qaqc", label="Run"),
actionButton("save_qaqc", label="Save")
Run Save
```

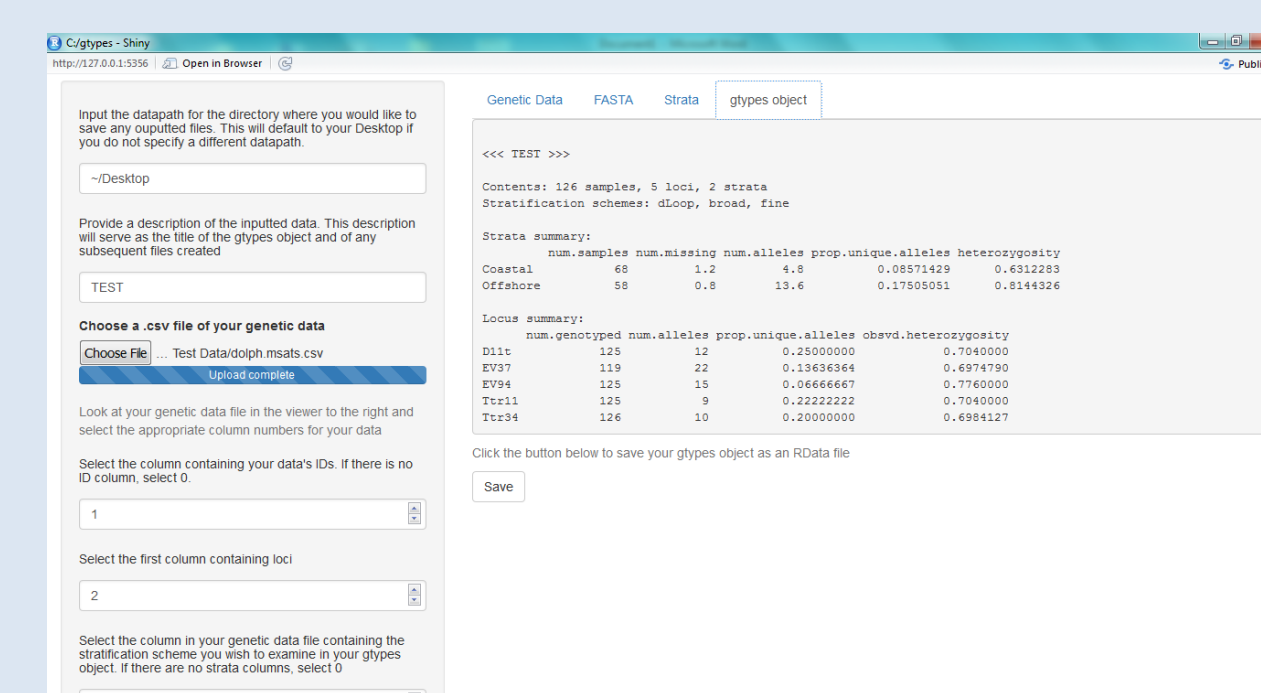
Tab panels create a dynamic display.

```
mainPanel(
  tabsetPanel("type",
    tabPanel("overall",
      verbatimTextOutput("stats_overall.pop")),
    tabPanel("pairwise",
      verbatimTextOutput("stats_pairwise.pop"))
  )
)
```

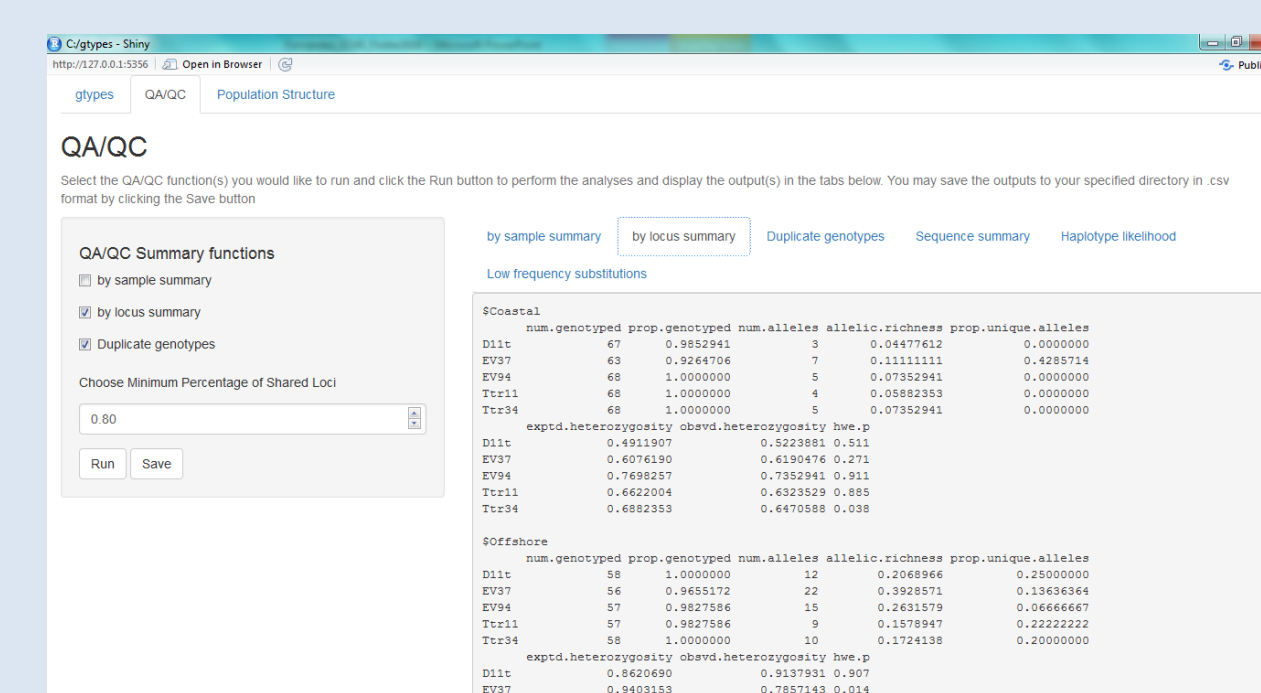
Using the strataG shiny Application

Running a shiny app requires two R scripts: the server³ and the user interface. The server code is essentially the brains of the app. It contains all of the strataG functions that analyze the genetic data and it interprets the inputs from each widget in the user interface.

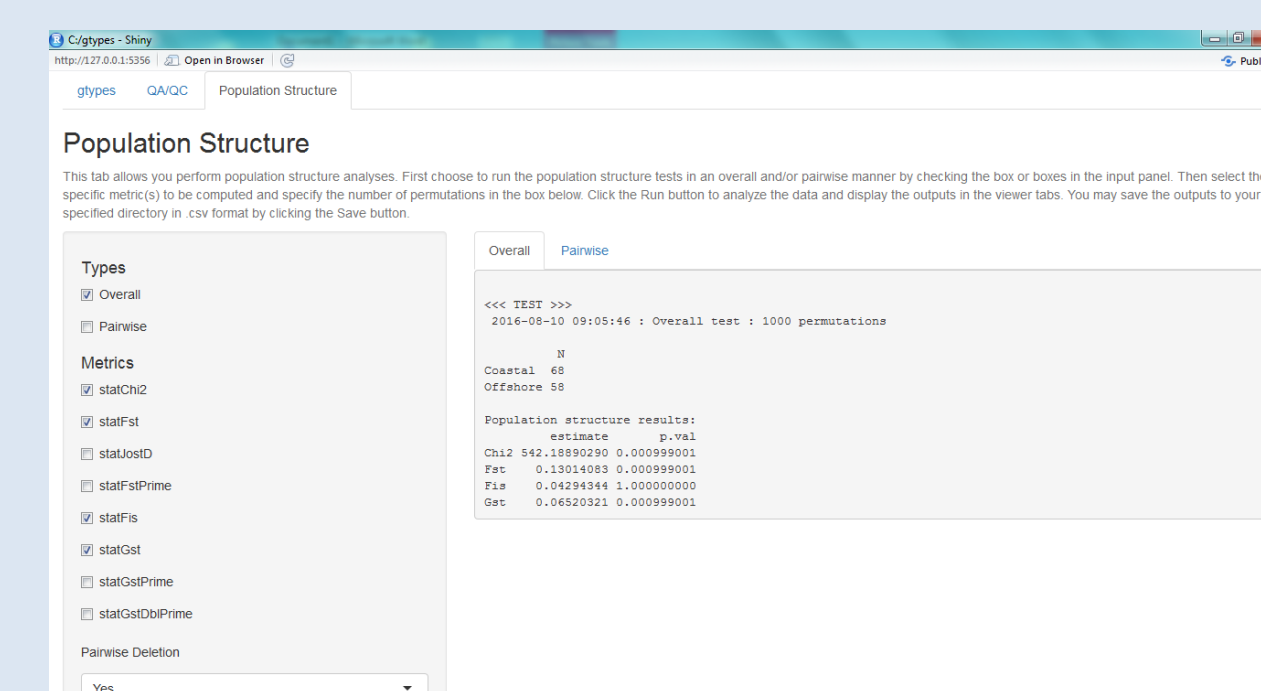
When both a server and user interface file are open in RStudio, the user will have the option to click a "Run App" button which will open the app in a new window.



The raw data is uploaded on the first tab and used to create a *gtypes object* that can then be analyzed by functions of strataG on the subsequent tabs.



The QA/QC tab allows users to run a variety of quality control functions on their genetic data. This important step helps users ensure that their data is as error-free as possible prior to downstream analyses.



Users can run a number of statistical tests to analyze the population structure of their data on the final tab.

All outputs generated by the app's functions can be saved in .csv format to any directory that the user specifies.

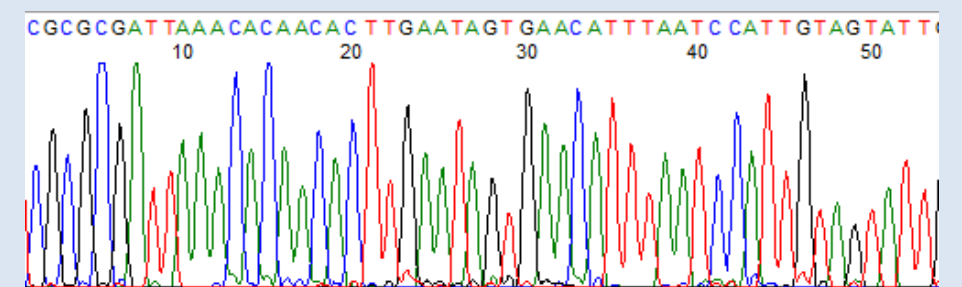
Broader Implications in Wildlife Genomics Studies



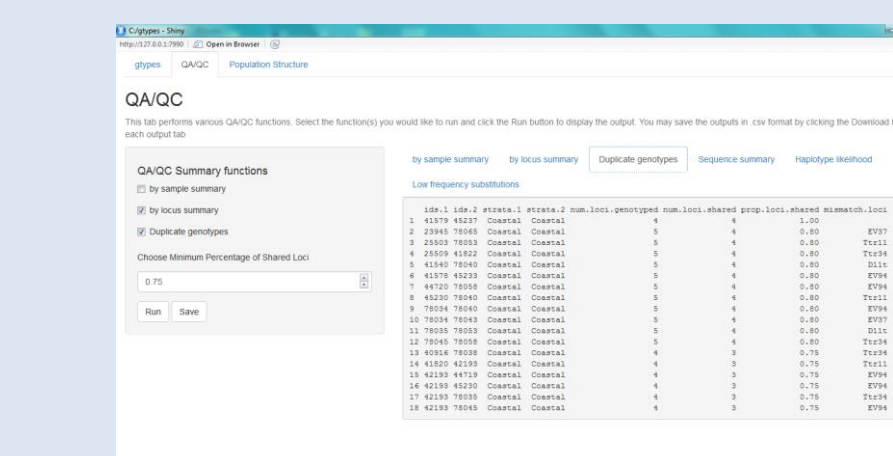
Sample collection in the field



Isolating genetic material in the lab



Sequencing the genetic material



Data analysis with the strataG shiny application

- Results of these analyses give researchers a greater understanding of marine mammal and turtle population genetics, which helps them address the conservation needs of these animals.
- User-friendly data analysis tools will assist researchers as the field of biology continues to increasingly require advanced computational analyses.

References

1. Archer E. 2016. strataG <https://github.com/EricArcher/strataG>
 2. Archer FI, Adams PE, Schneiders BB. 2016. strataG: An R package for manipulating, summarizing, and analyzing population genetic data. Molecular Ecology Resources, doi:10.1111/1755-0998.12559.
 3. Asfazadour W et al. "Data Development and Analysis Pathways for Marine Mammals and Turtles: Developing a Server Script" STAR 2016.
 4. RStudio. 2016. shiny <http://shiny.rstudio.com/>
- Leatherback image courtesy of NOAA SWFSC : <https://swfsc.noaa.gov/ImageGallery/default.aspx?aid=1>
Lab image courtesy of <http://vriesit.com/sites/default/files/DNA%20tubes-small.jpg>

Acknowledgements

This project has been made possible with support from Chevron (www.chvron.com) and the California State University STEM Teacher Researcher Program. Special thanks to Eric Archer and Lisa Komoroske for their mentorship.

