Greenish Warbler Genomic Analysis

Darren Irwin

2023-09-10

This page contains notes and code describing the data analysis for a manuscript on Greenish Warbler genomics. I've been working with the data for several years, and the R and then Julia code has been in development for a while. This is a Quarto notebook, which can run and display the results of Julia (or other) code blocks, along with text narration, and output in html, pdf, Word, etc.

The Julia code here is loosely based on R code written for Greenish Warbler analysis (Irwin et al. 2016), and then the North American warbler analyses (Irwin et al. 2019), and then my (unpublished) 2019 Greenish Warbler analysis. Most recently, this was adapted from the scripts called GW2022_R_analysis_script.R and IrwinLabGenomicsAnalysisScript.jl but has had a lot of opimizations since then. The SNP data here are a result of GBS reads mapped to our new 2022 Biozeron genome assembly for a greenish warbler from southern China.

Load packages

If running this for the first time, you will need to load packages used in the script, so run what is in this section below. It will take some time to install and precompile the packages:

```
import Pkg; Pkg.add("CSV") # took less than a minute
Pkg.add("DataFrames") # took about a minute
Pkg.add("Plots") # seems to install and working more simply than Makie (but less powerful)
Pkg.add("Haversine") # for great circle (Haversine) distances
Pkg.add("Distributions") # this seemed to fix a problem installing GLMakie
Pkg.add("MultivariateStats")
Pkg.add("StatsBase")
Pkg.add("Impute")
Pkg.add("JLD2")
Pkg.add("CairoMakie")
Pkg.add("PrettyTables") # for printing nice tables to REPL
```

Now actually load those packages into the Julia session:

```
using CSV # for reading in delimited files
using DataFrames # for storing data as type DataFrame
using Haversine # for calculating Great Circle (haversine) distances between sites
using MultivariateStats # for Principal Coordinates Analysis (multidimensional scaling)
using DelimitedFiles # for reading delimited files (the genotypic data)
using Impute # for imputing missing genotypes
using JLD2 # for saving data
using CairoMakie # for plots
CairoMakie.activate!()  # this makes CairoMakie the main package for figures (in case another already
```

Load my custom package `SNPlots`:

```
include("SNPlots.jl") # load file containing custom-built functions
using .SNPlots # actually make SNPlots module available with SNPlots.functionName(),
# or if functions are exported from SNPlots then they are available.
```

Test Julia:

```
x = 1; y = 2; z = x+y
println("z = ", z)
```

```
z = 3
```

(If Quarto is calling Julia properly, you will see `z = 3` as the output of the code block above.)

Choose working directory:

```
cd("/Users/darrenirwin/Dropbox/Darren's current work/")
```

OK, let's load the genomic data!

```
# choose path and filename for the 012NA files
baseName = "GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs
filenameTextMiddle = ".max2allele_noindel.vcf.maxmiss"
# indicate percent threshold for missing genotypes for each SNP--
# this was set by earlier filtering, and is just a record-keeper for the filenames:
missingGenotypeThreshold = 60
filenameTextEnd = ".MQ20.lowHet.tab"
tagName = ".Aug2023."   # choose a tag name for this analysis
```

```
# indicate name of metadata file, a text file with these column headings:
# ID    location    group    Fst_group    plot_order
metadataFile = "GW_genomics_2022_with_new_genome/GW_all4plates.Fst_groups.txt"
# load metadata
metadata = DataFrame(CSV.File(metadataFile)) # the CSV.File function interprets the correct delimite
num_metadata_cols = ncol(metadata)
num_individuals = nrow(metadata)
# read in individual names for this dataset
individuals_file_name = string(baseName, filenameTextMiddle, missingGenotypeThreshold, filenameTextEn
ind = DataFrame(CSV.File(individuals_file_name; header=["ind"], types=[String]))
indNum = size(ind, 1) # number of individuals
if num_individuals != indNum
    println("WARNING: number of rows in metadata file different than number of individuals in .indv
end
# read in position data for this dataset
position_file_name = string(baseName, filenameTextMiddle, missingGenotypeThreshold, filenameTextEnd,
pos_whole_genome = DataFrame(CSV.File(position_file_name; header=["chrom", "position"], types=[Strin
# read in genotype data
column_names = ["null"; string.("c.", pos_whole_genome.chrom, ".", pos_whole_genome.position)]
genotype_file_name = string(baseName, filenameTextMiddle, missingGenotypeThreshold, filenameTextEnd,
@time if 1 <= indNum <= 127
    geno = readdlm(genotype_file_name, '\t', Int8, '\n'); # this has been sped up dramatically, by fi
elseif 128 <= indNum <= 32767
    geno = readdlm(genotype_file_name, '\t', Int16, '\n'); # this needed for first column, which is n
else
    print("Error: Number of individuals in .indv appears outside of range from 1 to 32767")
end
loci_count = size(geno, 2) - 1   # because the first column is not a SNP (just a count from zero)
print(string("Read in genotypic data at ", loci_count," loci for ", indNum, " individuals. \n"))
```

```
 50.430901 seconds (3.61 M allocations: 9.585 GiB, 0.17% gc time, 0.64% compilation time)
Read in genotypic data at 2431709 loci for 310 individuals.
```

Check that individuals are same in genotype data and metadata

```
ind_with_metadata = hcat(ind, metadata)
print(ind_with_metadata)
print("\n")  # prints a line break
if isequal(ind_with_metadata.ind, ind_with_metadata.ID)
    println("GOOD NEWS: names of individuals in metadata file and genotype ind file match perfectly.
```

```
else
    println("WARNING: names of individuals in metadata file and genotype ind file do not completely m
end
```

310×6 DataFrame

Row	ind	ID	location	group	Fst_group	plot_order
	String	String31	String7	String15	String15	Float64
1	GW_Armando_plate1_AB1	GW_Armando_plate1_AB1	AB_rep	vir_rep	vir_rep	20.0
2	GW_Armando_plate1_JF07G02	GW_Armando_plate1_JF07G02	ST	plumb	plumb	87.0
3	GW_Armando_plate1_JF07G03	GW_Armando_plate1_JF07G03	ST	plumb	plumb	87.0
4	GW_Armando_plate1_JF07G04	GW_Armando_plate1_JF07G04	ST	plumb	plumb	87.0
5	GW_Armando_plate1_JF08G02	GW_Armando_plate1_JF08G02	ST	plumb	plumb	87.0
6	GW_Armando_plate1_JF09G01	GW_Armando_plate1_JF09G01	ST	plumb	plumb	87.0
7	GW_Armando_plate1_JF09G02	GW_Armando_plate1_JF09G02	ST	plumb	plumb	87.0
8	GW_Armando_plate1_JF10G03	GW_Armando_plate1_JF10G03	ST	plumb_vir	plumb_vir	7
9	GW_Armando_plate1_JF11G01	GW_Armando_plate1_JF11G01	ST	plumb	plumb	87.0
10	GW_Armando_plate1_JF12G01	GW_Armando_plate1_JF12G01	ST	plumb	plumb	87.0
11	GW_Armando_plate1_JF12G02	GW_Armando_plate1_JF12G02	ST	plumb	plumb	87.0
12	GW_Armando_plate1_JF12G04	GW_Armando_plate1_JF12G04	ST_vi	vir	vir	24.0
13	GW_Armando_plate1_JF13G01	GW_Armando_plate1_JF13G01	ST	plumb	plumb	87.0
14	GW_Armando_plate1_JF15G03	GW_Armando_plate1_JF15G03	KK	plumb	plumb	87.0
15	GW_Armando_plate1_JF16G01	GW_Armando_plate1_JF16G01	KK_vi	plumb_vir	vir	24
16	GW_Armando_plate1_JF20G01	GW_Armando_plate1_JF20G01	KK	plumb	plumb	87.0
17	GW_Armando_plate1_JF22G01	GW_Armando_plate1_JF22G01	KK	plumb	plumb	87.0
18	GW_Armando_plate1_JF23G01	GW_Armando_plate1_JF23G01	KK	plumb	plumb	87.0
19	GW_Armando_plate1_JF23G02	GW_Armando_plate1_JF23G02	KK	plumb	plumb	87.0
20	GW_Armando_plate1_JF24G02	GW_Armando_plate1_JF24G02	KK	plumb	plumb	87.0
21	GW_Armando_plate1_JF26G01	GW_Armando_plate1_JF26G01	ST	plumb	plumb	87.0
22	GW_Armando_plate1_JF27G01	GW_Armando_plate1_JF27G01	ST	plumb	plumb	87.0
23	GW_Armando_plate1_JF29G01	GW_Armando_plate1_JF29G01	ST	plumb	plumb	87.0
24	GW_Armando_plate1_JF29G02	GW_Armando_plate1_JF29G02	ST	plumb	plumb	87.0
25	GW_Armando_plate1_JF29G03	GW_Armando_plate1_JF29G03	ST	plumb	plumb	87.0
26	GW_Armando_plate1_JG02G02	GW_Armando_plate1_JG02G02	KK	plumb	plumb	87.0
27	GW_Armando_plate1_JG02G04	GW_Armando_plate1_JG02G04	KK	plumb	plumb	87.0
28	GW_Armando_plate1_JG08G01	GW_Armando_plate1_JG08G01	ST	plumb	plumb	87.0
29	GW_Armando_plate1_JG08G02	GW_Armando_plate1_JG08G02	ST	plumb	plumb	87.0

30	GW_Armando_plate1_JG10G01	GW_Armando_plate1_JG10G01	ST	plumb	plumb	87.0
31	GW_Armando_plate1_JG12G01	GW_Armando_plate1_JG12G01	ST	plumb	plumb	87.0
32	GW_Armando_plate1_JG17G01	GW_Armando_plate1_JG17G01	ST	plumb_vir	plumb	77
33	GW_Armando_plate1_NO_BC_TTGW05	GW_Armando_plate1_NO_BC_TTGW05	blank	blank	blank	
34	GW_Armando_plate1_NO_DNA	GW_Armando_plate1_NO_DNA	blank	blank	blank	-99.
35	GW_Armando_plate1_RF20G01	GW_Armando_plate1_RF20G01	BJ	obs_plumb	plumb_BJ	7
36	GW_Armando_plate1_RF29G02	GW_Armando_plate1_RF29G02	BJ	obs_plumb	plumb_BJ	7
37	GW_Armando_plate1_TL3	GW_Armando_plate1_TL3	TL_rep	vir_rep	vir_rep	11.0
38	GW_Armando_plate1_TTGW01	GW_Armando_plate1_TTGW01	MN	troch_MN	troch_west	53
39	GW_Armando_plate1_TTGW05_rep1	GW_Armando_plate1_TTGW05_rep1	MN_rep	troch_MN_rep	troch_west_r	
40	GW_Armando_plate1_TTGW05_rep2	GW_Armando_plate1_TTGW05_rep2	MN_rep	troch_MN_rep	troch_west_r	
41	GW_Armando_plate1_TTGW06	GW_Armando_plate1_TTGW06	SU	lud_Sukhto	lud_central	4
42	GW_Armando_plate1_TTGW07	GW_Armando_plate1_TTGW07	SU	lud_Sukhto	lud_central	4
43	GW_Armando_plate1_TTGW10	GW_Armando_plate1_TTGW10	SU	lud_Sukhto	lud_central	4
44	GW_Armando_plate1_TTGW11	GW_Armando_plate1_TTGW11	SU	lud_Sukhto	lud_central	4
45	GW_Armando_plate1_TTGW13	GW_Armando_plate1_TTGW13	TH	lud_Thallighar	lud_central	
46	GW_Armando_plate1_TTGW17	GW_Armando_plate1_TTGW17	TH	lud_Thallighar	lud_central	
47	GW_Armando_plate1_TTGW19	GW_Armando_plate1_TTGW19	TH	lud_Thallighar	lud_central	
48	GW_Armando_plate1_TTGW21	GW_Armando_plate1_TTGW21	SR	lud_Sural	lud_central	4
49	GW_Armando_plate1_TTGW22	GW_Armando_plate1_TTGW22	SR	lud_Sural	lud_central	4
50	GW_Armando_plate1_TTGW23	GW_Armando_plate1_TTGW23	SR	lud_Sural	lud_central	4
51	GW_Armando_plate1_TTGW29	GW_Armando_plate1_TTGW29	SR	lud_Sural	lud_central	4
52	GW_Armando_plate1_TTGW52	GW_Armando_plate1_TTGW52	NG	lud_Nainaghar	lud_central	
53	GW_Armando_plate1_TTGW53	GW_Armando_plate1_TTGW53	NG	lud_Nainaghar	lud_central	
54	GW_Armando_plate1_TTGW55	GW_Armando_plate1_TTGW55	NG	lud_Nainaghar	lud_central	
55	GW_Armando_plate1_TTGW57	GW_Armando_plate1_TTGW57	NG	lud_Nainaghar	lud_central	
56	GW_Armando_plate1_TTGW58	GW_Armando_plate1_TTGW58	NG	lud_Nainaghar	lud_central	
57	GW_Armando_plate1_TTGW59	GW_Armando_plate1_TTGW59	NG	lud_Nainaghar	lud_central	
58	GW_Armando_plate1_TTGW63	GW_Armando_plate1_TTGW63	SP	lud_Spiti	troch_west	5
59	GW_Armando_plate1_TTGW64	GW_Armando_plate1_TTGW64	SP	lud_Spiti	troch_west	5
60	GW_Armando_plate1_TTGW65	GW_Armando_plate1_TTGW65	SP	lud_Spiti	troch_west	5
61	GW_Armando_plate1_TTGW66	GW_Armando_plate1_TTGW66	SP	lud_Spiti	troch_west	5
62	GW_Armando_plate1_TTGW68	GW_Armando_plate1_TTGW68	SP	lud_Spiti	troch_west	5
63	GW_Armando_plate1_TTGW70	GW_Armando_plate1_TTGW70	SA	lud_Sathrundi	lud_Sath	4
64	GW_Armando_plate1_TTGW71	GW_Armando_plate1_TTGW71	SA	lud_Sathrundi	lud_Sath	4
65	GW_Armando_plate1_TTGW72	GW_Armando_plate1_TTGW72	SA	lud_Sathrundi	lud_Sath	4
66	GW_Armando_plate1_TTGW74	GW_Armando_plate1_TTGW74	SA	lud_Sathrundi	lud_Sath	4

67	GW_Armando_plate1_TTGW78	GW_Armando_plate1_TTGW78	SA	lud_Sathrundi	lud_Sath	4
68	GW_Armando_plate1_TTGW_15_05	GW_Armando_plate1_TTGW_15_05	SR	lud_Sural	lud_central	
69	GW_Armando_plate1_TTGW_15_07	GW_Armando_plate1_TTGW_15_07	SR	lud_Sural	lud_central	
70	GW_Armando_plate1_TTGW_15_08	GW_Armando_plate1_TTGW_15_08	SR	lud_Sural	lud_central	
71	GW_Armando_plate1_TTGW_15_09	GW_Armando_plate1_TTGW_15_09	SR	lud_Sural	lud_central	
72	GW_Armando_plate1_UY1	GW_Armando_plate1_UY1	UY_rep	plumb_rep	plumb_rep	88
73	GW_Armando_plate2_IL2	GW_Armando_plate2_IL2	IL_rep	plumb_rep	plumb_rep	83
74	GW_Armando_plate2_JE31G01	GW_Armando_plate2_JE31G01	KK_vi	vir_misID	vir	24
75	GW_Armando_plate2_JF03G01	GW_Armando_plate2_JF03G01	ST_vi	vir_misID	vir	24
76	GW_Armando_plate2_JF03G02	GW_Armando_plate2_JF03G02	KK_vi	vir_misID	vir	24
77	GW_Armando_plate2_JF07G01	GW_Armando_plate2_JF07G01	ST	plumb	plumb	87.0
78	GW_Armando_plate2_JF08G04	GW_Armando_plate2_JF08G04	ST	plumb	plumb	87.0
79	GW_Armando_plate2_JF10G02	GW_Armando_plate2_JF10G02	ST	plumb	plumb	87.0
80	GW_Armando_plate2_JF11G02	GW_Armando_plate2_JF11G02	ST	plumb	plumb	87.0
81	GW_Armando_plate2_JF12G03	GW_Armando_plate2_JF12G03	ST	plumb	plumb	87.0
82	GW_Armando_plate2_JF12G05	GW_Armando_plate2_JF12G05	ST	plumb	plumb	87.0
83	GW_Armando_plate2_JF13G02	GW_Armando_plate2_JF13G02	ST	plumb	plumb	87.0
84	GW_Armando_plate2_JF14G01	GW_Armando_plate2_JF14G01	KK	plumb	plumb	87.0
85	GW_Armando_plate2_JF14G02	GW_Armando_plate2_JF14G02	KK	plumb	plumb	87.0
86	GW_Armando_plate2_JF15G01	GW_Armando_plate2_JF15G01	KK	plumb	plumb	87.0
87	GW_Armando_plate2_JF15G02	GW_Armando_plate2_JF15G02	KK	plumb	plumb	87.0
88	GW_Armando_plate2_JF16G02	GW_Armando_plate2_JF16G02	KK_vi	plumb_vir	vir	24
89	GW_Armando_plate2_JF19G01	GW_Armando_plate2_JF19G01	KK	plumb	plumb	87.0
90	GW_Armando_plate2_JF20G02	GW_Armando_plate2_JF20G02	KK	plumb	plumb	87.0
91	GW_Armando_plate2_JF24G01	GW_Armando_plate2_JF24G01	KK	plumb	plumb	87.0
92	GW_Armando_plate2_JF24G03	GW_Armando_plate2_JF24G03	ST	plumb	plumb	87.0
93	GW_Armando_plate2_JF25G01	GW_Armando_plate2_JF25G01	KK	plumb	plumb	87.0
94	GW_Armando_plate2_JF26G02	GW_Armando_plate2_JF26G02	KK	plumb	plumb	87.0
95	GW_Armando_plate2_JF27G02	GW_Armando_plate2_JF27G02	KK	plumb	plumb	87.0
96	GW_Armando_plate2_JF30G01	GW_Armando_plate2_JF30G01	ST_vi	vir_misID	vir	24
97	GW_Armando_plate2_JG01G01	GW_Armando_plate2_JG01G01	KK	plumb	plumb	87.0
98	GW_Armando_plate2_JG02G01	GW_Armando_plate2_JG02G01	KK	plumb	plumb	87.0
99	GW_Armando_plate2_JG02G03	GW_Armando_plate2_JG02G03	KK	plumb	plumb	87.0
100	GW_Armando_plate2_JG10G02	GW_Armando_plate2_JG10G02	ST	plumb	plumb	87.
101	GW_Armando_plate2_JG10G03	GW_Armando_plate2_JG10G03	ST	plumb	plumb	87.
102	GW_Armando_plate2_JG12G02	GW_Armando_plate2_JG12G02	ST	plumb	plumb	87.
103	GW_Armando_plate2_JG12G03	GW_Armando_plate2_JG12G03	ST	plumb	plumb	87.

```
GW_Armando_plate2_LN11       GW_Armando_plate2_LN11       LN_rep   troch_LN_rep   troch_LN_rep
GW_Armando_plate2_LN2        GW_Armando_plate2_LN2        LN_rep   troch_LN_rep   troch_LN_rep
GW_Armando_plate2_NO_BC_TTGW05 GW_Armando_plate2_NO_BC_TTGW05 KK      plumb        blank          -
GW_Armando_plate2_NO_DNA     GW_Armando_plate2_NO_DNA     blank    blank        blank          -99
GW_Armando_plate2_RF29G01    GW_Armando_plate2_RF29G01    BJ      obs_plumb     plumb_BJ        7
GW_Armando_plate2_TTGW02     GW_Armando_plate2_TTGW02     MN      troch_MN      troch_west      5
GW_Armando_plate2_TTGW03     GW_Armando_plate2_TTGW03     MN      troch_MN      troch_west      5
GW_Armando_plate2_TTGW05_rep3 GW_Armando_plate2_TTGW05_rep3 MN_rep   troch_MN_rep   troch_west_r
GW_Armando_plate2_TTGW05_rep4 GW_Armando_plate2_TTGW05_rep4 MN_rep   troch_MN_rep   troch_west_r
GW_Armando_plate2_TTGW08     GW_Armando_plate2_TTGW08     SU      lud_Sukhto    lud_central
GW_Armando_plate2_TTGW09     GW_Armando_plate2_TTGW09     SU      lud_Sukhto    lud_central
GW_Armando_plate2_TTGW12     GW_Armando_plate2_TTGW12     TH      lud_Thallighar lud_central
GW_Armando_plate2_TTGW14     GW_Armando_plate2_TTGW14     TH      lud_Thallighar lud_central
GW_Armando_plate2_TTGW15     GW_Armando_plate2_TTGW15     TH      lud_Thallighar lud_central
GW_Armando_plate2_TTGW16     GW_Armando_plate2_TTGW16     TH      lud_Thallighar lud_central
GW_Armando_plate2_TTGW18     GW_Armando_plate2_TTGW18     TH      lud_Thallighar lud_central
GW_Armando_plate2_TTGW20     GW_Armando_plate2_TTGW20     SR      lud_Sural     lud_central      4
GW_Armando_plate2_TTGW24     GW_Armando_plate2_TTGW24     SR      lud_Sural     lud_central      4
GW_Armando_plate2_TTGW25     GW_Armando_plate2_TTGW25     SR      lud_Sural     lud_central      4
GW_Armando_plate2_TTGW27     GW_Armando_plate2_TTGW27     SR      lud_Sural     lud_central      4
GW_Armando_plate2_TTGW28     GW_Armando_plate2_TTGW28     SR      lud_Sural     lud_central      4
GW_Armando_plate2_TTGW50     GW_Armando_plate2_TTGW50     NG      lud_Nainaghar  lud_central
GW_Armando_plate2_TTGW51     GW_Armando_plate2_TTGW51     NG      lud_Nainaghar  lud_central
GW_Armando_plate2_TTGW54     GW_Armando_plate2_TTGW54     NG      lud_Nainaghar  lud_central
GW_Armando_plate2_TTGW56     GW_Armando_plate2_TTGW56     NG      lud_Nainaghar  lud_central
GW_Armando_plate2_TTGW60     GW_Armando_plate2_TTGW60     SP      lud_Spiti     troch_west      5
GW_Armando_plate2_TTGW61     GW_Armando_plate2_TTGW61     SP      lud_Spiti     troch_west      5
GW_Armando_plate2_TTGW62     GW_Armando_plate2_TTGW62     SP      lud_Spiti     troch_west      5
GW_Armando_plate2_TTGW67     GW_Armando_plate2_TTGW67     SP      lud_Spiti     troch_west      5
GW_Armando_plate2_TTGW69     GW_Armando_plate2_TTGW69     SP      lud_Spiti     troch_west      5
GW_Armando_plate2_TTGW73     GW_Armando_plate2_TTGW73     SA      lud_Sathrundi  lud_Sath
GW_Armando_plate2_TTGW75     GW_Armando_plate2_TTGW75     SA      lud_Sathrundi  lud_Sath
GW_Armando_plate2_TTGW77     GW_Armando_plate2_TTGW77     SA      lud_Sathrundi  lud_Sath
GW_Armando_plate2_TTGW79     GW_Armando_plate2_TTGW79     SA      lud_Sathrundi  lud_Sath
GW_Armando_plate2_TTGW80     GW_Armando_plate2_TTGW80     SA      lud_Sathrundi  lud_Sath
GW_Armando_plate2_TTGW_15_01   GW_Armando_plate2_TTGW_15_01   SR      lud_Sural     lud_central
GW_Armando_plate2_TTGW_15_02   GW_Armando_plate2_TTGW_15_02   SR      lud_Sural     lud_central
```

141	GW_Armando_plate2_TTGW_15_03	GW_Armando_plate2_TTGW_15_03	SR	lud_Sural	lud_central	
142	GW_Armando_plate2_TTGW_15_04	GW_Armando_plate2_TTGW_15_04	SR	lud_Sural	lud_central	
143	GW_Armando_plate2_TTGW_15_06	GW_Armando_plate2_TTGW_15_06	SR	lud_Sural	lud_central	
144	GW_Armando_plate2_TTGW_15_10	GW_Armando_plate2_TTGW_15_10	SR	lud_Sural	lud_central	
145	GW_Lane5_AA1	GW_Lane5_AA1	AA	vir_S	vir_S	25.0
146	GW_Lane5_AA10	GW_Lane5_AA10	AA	vir_S	vir_S	33.0
147	GW_Lane5_AA11	GW_Lane5_AA11	AA	vir_S	vir_S	34.0
148	GW_Lane5_AA3	GW_Lane5_AA3	AA	vir_S	vir_S	26.0
149	GW_Lane5_AA4	GW_Lane5_AA4	AA	vir_S	vir_S	27.0
150	GW_Lane5_AA5	GW_Lane5_AA5	AA	vir_S	vir_S	28.0
151	GW_Lane5_AA6	GW_Lane5_AA6	AA	vir_S	vir_S	29.0
152	GW_Lane5_AA7	GW_Lane5_AA7	AA	vir_S	vir_S	30.0
153	GW_Lane5_AA8	GW_Lane5_AA8	AA	vir_S	vir_S	31.0
154	GW_Lane5_AA9	GW_Lane5_AA9	AA	vir_S	vir_S	32.0
155	GW_Lane5_AB1	GW_Lane5_AB1	AB	vir	vir	20.0
156	GW_Lane5_AB2	GW_Lane5_AB2	AB	vir	vir	21.0
157	GW_Lane5_AN1	GW_Lane5_AN1	AN	plumb	plumb	80.0
158	GW_Lane5_AN2	GW_Lane5_AN2	AN	plumb	plumb	81.0
159	GW_Lane5_BK2	GW_Lane5_BK2	BK	plumb	plumb	78.0
160	GW_Lane5_BK3	GW_Lane5_BK3	BK	plumb	plumb	79.0
161	GW_Lane5_DA2	GW_Lane5_DA2	XN	obs	obs	73.0
162	GW_Lane5_DA3	GW_Lane5_DA3	XN	obs	obs	74.0
163	GW_Lane5_DA4	GW_Lane5_DA4	XN	obs	obs	75.0
164	GW_Lane5_DA6	GW_Lane5_DA6	XN	obs	low_reads	76.0
165	GW_Lane5_DA7	GW_Lane5_DA7	XN	obs	obs	77.0
166	GW_Lane5_EM1	GW_Lane5_EM1	EM	troch_EM	troch_EM	72.0
167	GW_Lane5_IL1	GW_Lane5_IL1	IL	plumb	plumb	82.0
168	GW_Lane5_IL2	GW_Lane5_IL2	IL	plumb	plumb	83.0
169	GW_Lane5_IL4	GW_Lane5_IL4	IL	plumb	plumb	84.0
170	GW_Lane5_KS1	GW_Lane5_KS1	OV	lud_KS	lud_KS	40.0
171	GW_Lane5_KS2	GW_Lane5_KS2	OV	lud_KS	lud_KS	40.0
172	GW_Lane5_LN1	GW_Lane5_LN1	LN	troch_LN	troch_LN	57.0
173	GW_Lane5_LN10	GW_Lane5_LN10	LN	troch_LN	troch_LN	64.0
174	GW_Lane5_LN11	GW_Lane5_LN11	LN	troch_LN	troch_LN	65.0
175	GW_Lane5_LN12	GW_Lane5_LN12	LN	troch_LN	troch_LN	66.0
176	GW_Lane5_LN14	GW_Lane5_LN14	LN	troch_LN	troch_LN	67.0
177	GW_Lane5_LN16	GW_Lane5_LN16	LN	troch_LN	troch_LN	68.0

178	GW_Lane5_LN18	GW_Lane5_LN18	LN	troch_LN	troch_LN	69.0
179	GW_Lane5_LN19	GW_Lane5_LN19	LN	troch_LN	troch_LN	70.0
180	GW_Lane5_LN2	GW_Lane5_LN2	LN	troch_LN	troch_LN	58.0
181	GW_Lane5_LN20	GW_Lane5_LN20	LN	troch_LN	troch_LN	71.0
182	GW_Lane5_LN3	GW_Lane5_LN3	LN	troch_LN	troch_LN	59.0
183	GW_Lane5_LN4	GW_Lane5_LN4	LN	troch_LN	troch_LN	60.0
184	GW_Lane5_LN6	GW_Lane5_LN6	LN	troch_LN	troch_LN	61.0
185	GW_Lane5_LN7	GW_Lane5_LN7	LN	troch_LN	troch_LN	62.0
186	GW_Lane5_LN8	GW_Lane5_LN8	LN	troch_LN	troch_LN	63.0
187	GW_Lane5_MN1	GW_Lane5_MN1	MN	troch_MN	troch_west	51.0
188	GW_Lane5_MN12	GW_Lane5_MN12	MN	troch_MN	troch_west	56.0
189	GW_Lane5_MN3	GW_Lane5_MN3	MN	troch_MN	troch_west	52.0
190	GW_Lane5_MN5	GW_Lane5_MN5	MN	troch_MN	troch_west	53.0
191	GW_Lane5_MN8	GW_Lane5_MN8	MN	troch_MN	troch_west	54.0
192	GW_Lane5_MN9	GW_Lane5_MN9	MN	troch_MN	troch_west	55.0
193	GW_Lane5_NA1	GW_Lane5_NA1	NR	lud_PK	lud_PK	39.2
194	GW_Lane5_NA3-3ul	GW_Lane5_NA3-3ul	NR	lud_PK	lud_PK	39.2
195	GW_Lane5_PT11	GW_Lane5_PT11	KL	lud_KL	lud_central	42.0
196	GW_Lane5_PT12	GW_Lane5_PT12	KL	lud_KL	lud_central	42.0
197	GW_Lane5_PT2	GW_Lane5_PT2	ML	lud_ML	lud_ML	51.0
198	GW_Lane5_PT3	GW_Lane5_PT3	PA	lud_PA	lud_central	46.0
199	GW_Lane5_PT4	GW_Lane5_PT4	PA	lud_PA	lud_central	46.0
200	GW_Lane5_PT6	GW_Lane5_PT6	KL	lud_KL	lud_central	42.0
201	GW_Lane5_SH1	GW_Lane5_SH1	PK	lud_PK	lud_PK	39.1
202	GW_Lane5_SH2	GW_Lane5_SH2	PK	lud_PK	lud_PK	39.1
203	GW_Lane5_SH4	GW_Lane5_SH4	PK	lud_PK	lud_PK	39.1
204	GW_Lane5_SH5	GW_Lane5_SH5	PK	lud_PK	lud_PK	39.1
205	GW_Lane5_SL1	GW_Lane5_SL1	SL	plumb	plumb	95.0
206	GW_Lane5_SL2	GW_Lane5_SL2	SL	plumb	plumb	96.0
207	GW_Lane5_ST1	GW_Lane5_ST1	ST	plumb	plumb	85.0
208	GW_Lane5_ST12	GW_Lane5_ST12	ST	plumb	plumb	87.0
209	GW_Lane5_ST3	GW_Lane5_ST3	ST	plumb	plumb	86.0
210	GW_Lane5_STvi1	GW_Lane5_STvi1	STvi	vir	vir	22.0
211	GW_Lane5_STvi2	GW_Lane5_STvi2	STvi	vir	vir	23.0
212	GW_Lane5_STvi3	GW_Lane5_STvi3	STvi	vir	vir	24.0
213	GW_Lane5_TA1	GW_Lane5_TA1	TA	plumb	plumb	94.0
214	GW_Lane5_TL1	GW_Lane5_TL1	TL	vir	vir	9.0

215	GW_Lane5_TL10	GW_Lane5_TL10	TL	vir	vir	17.0
216	GW_Lane5_TL11	GW_Lane5_TL11	TL	vir	vir	18.0
217	GW_Lane5_TL12	GW_Lane5_TL12	TL	vir	vir	19.0
218	GW_Lane5_TL2	GW_Lane5_TL2	TL	vir	vir	10.0
219	GW_Lane5_TL3	GW_Lane5_TL3	TL	vir	vir	11.0
220	GW_Lane5_TL4	GW_Lane5_TL4	TL	vir	vir	12.0
221	GW_Lane5_TL5	GW_Lane5_TL5	TL	vir	vir	13.0
222	GW_Lane5_TL7	GW_Lane5_TL7	TL	vir	vir	14.0
223	GW_Lane5_TL8	GW_Lane5_TL8	TL	vir	vir	15.0
224	GW_Lane5_TL9	GW_Lane5_TL9	TL	vir	vir	16.0
225	GW_Lane5_TU1	GW_Lane5_TU1	TU	nit	nit	35.0
226	GW_Lane5_TU2	GW_Lane5_TU2	TU	nit	nit	36.0
227	GW_Lane5_UY1	GW_Lane5_UY1	UY	plumb	plumb	88.0
228	GW_Lane5_UY2	GW_Lane5_UY2	UY	plumb	plumb	89.0
229	GW_Lane5_UY3	GW_Lane5_UY3	UY	plumb	plumb	90.0
230	GW_Lane5_UY4	GW_Lane5_UY4	UY	plumb	plumb	91.0
231	GW_Lane5_UY5	GW_Lane5_UY5	UY	plumb	plumb	92.0
232	GW_Lane5_UY6	GW_Lane5_UY6	UY	plumb	plumb	93.0
233	GW_Lane5_YK1	GW_Lane5_YK1	YK	vir	vir	1.0
234	GW_Lane5_YK11	GW_Lane5_YK11	YK	vir	vir	8.0
235	GW_Lane5_YK3	GW_Lane5_YK3	YK	vir	vir	2.0
236	GW_Lane5_YK4	GW_Lane5_YK4	YK	vir	vir	3.0
237	GW_Lane5_YK5	GW_Lane5_YK5	YK	vir	vir	4.0
238	GW_Lane5_YK6	GW_Lane5_YK6	YK	vir	vir	5.0
239	GW_Lane5_YK7	GW_Lane5_YK7	YK	vir	vir	6.0
240	GW_Lane5_YK9	GW_Lane5_YK9	YK	vir	vir	7.0
241	GW_Liz_GBS_Liz10045	GW_Liz_GBS_Liz10045	ML	lud	lud_ML	51.01
242	GW_Liz_GBS_Liz10094	GW_Liz_GBS_Liz10094	ML	lud	lud_ML	51.02
243	GW_Liz_GBS_Liz5101	GW_Liz_GBS_Liz5101	ML	lud	lud_ML	51.03
244	GW_Liz_GBS_Liz5101_R	GW_Liz_GBS_Liz5101_R	ML_rep	lud_rep	lud_ML_rep	51.
245	GW_Liz_GBS_Liz5118	GW_Liz_GBS_Liz5118	ML	lud	lud_ML	51.05
246	GW_Liz_GBS_Liz5139	GW_Liz_GBS_Liz5139	ML	lud	lud_ML	51.06
247	GW_Liz_GBS_Liz5142	GW_Liz_GBS_Liz5142	ML	lud	lud_ML	51.07
248	GW_Liz_GBS_Liz5144	GW_Liz_GBS_Liz5144	ML	lud	lud_ML	51.08
249	GW_Liz_GBS_Liz5150	GW_Liz_GBS_Liz5150	ML	lud	lud_ML	51.09
250	GW_Liz_GBS_Liz5159	GW_Liz_GBS_Liz5159	ML	lud_chick	lud_ML	51.1
251	GW_Liz_GBS_Liz5162	GW_Liz_GBS_Liz5162	ML	lud_chick	lud_ML	51.11

252	GW_Liz_GBS_Liz5163	GW_Liz_GBS_Liz5163	ML	lud_chick	lud_ML	51.12
253	GW_Liz_GBS_Liz5164	GW_Liz_GBS_Liz5164	ML	lud_chick	lud_ML	51.13
254	GW_Liz_GBS_Liz5165	GW_Liz_GBS_Liz5165	ML	lud	lud_ML	51.14
255	GW_Liz_GBS_Liz5167	GW_Liz_GBS_Liz5167	ML	lud_chick	lud_ML	51.15
256	GW_Liz_GBS_Liz5168	GW_Liz_GBS_Liz5168	ML	lud_chick	lud_ML	51.16
257	GW_Liz_GBS_Liz5169	GW_Liz_GBS_Liz5169	ML	lud_chick	lud_ML	51.17
258	GW_Liz_GBS_Liz5171	GW_Liz_GBS_Liz5171	ML	lud	lud_ML	51.18
259	GW_Liz_GBS_Liz5172	GW_Liz_GBS_Liz5172	ML	lud_chick	lud_ML	51.19
260	GW_Liz_GBS_Liz5173	GW_Liz_GBS_Liz5173	ML	lud_chick	lud_ML	51.2
261	GW_Liz_GBS_Liz5174	GW_Liz_GBS_Liz5174	ML	lud	lud_ML	51.21
262	GW_Liz_GBS_Liz5175	GW_Liz_GBS_Liz5175	ML	lud	lud_ML	51.22
263	GW_Liz_GBS_Liz5176	GW_Liz_GBS_Liz5176	ML	lud	lud_ML	51.23
264	GW_Liz_GBS_Liz5177	GW_Liz_GBS_Liz5177	ML	lud_chick	lud_ML	51.24
265	GW_Liz_GBS_Liz5178	GW_Liz_GBS_Liz5178	ML	lud_chick	lud_ML	51.25
266	GW_Liz_GBS_Liz5179	GW_Liz_GBS_Liz5179	ML	lud_chick	lud_ML	51.26
267	GW_Liz_GBS_Liz5180	GW_Liz_GBS_Liz5180	ML	lud	lud_ML	51.27
268	GW_Liz_GBS_Liz5182	GW_Liz_GBS_Liz5182	ML	lud_chick	lud_ML	51.28
269	GW_Liz_GBS_Liz5184	GW_Liz_GBS_Liz5184	ML	lud_chick	lud_ML	51.29
270	GW_Liz_GBS_Liz5185	GW_Liz_GBS_Liz5185	ML	lud	lud_ML	51.3
271	GW_Liz_GBS_Liz5186	GW_Liz_GBS_Liz5186	ML	lud_chick	lud_ML	51.31
272	GW_Liz_GBS_Liz5187	GW_Liz_GBS_Liz5187	ML	lud_chick	lud_ML	51.32
273	GW_Liz_GBS_Liz5188	GW_Liz_GBS_Liz5188	ML	lud	lud_ML	51.33
274	GW_Liz_GBS_Liz5189	GW_Liz_GBS_Liz5189	ML	lud_chick	lud_ML	51.34
275	GW_Liz_GBS_Liz5190	GW_Liz_GBS_Liz5190	ML	lud_chick	lud_ML	51.35
276	GW_Liz_GBS_Liz5191	GW_Liz_GBS_Liz5191	ML	lud_chick	lud_ML	51.36
277	GW_Liz_GBS_Liz5192	GW_Liz_GBS_Liz5192	ML	lud_chick	lud_ML	51.37
278	GW_Liz_GBS_Liz5193	GW_Liz_GBS_Liz5193	ML	lud_chick	lud_ML	51.38
279	GW_Liz_GBS_Liz5194	GW_Liz_GBS_Liz5194	ML	lud_chick	lud_ML	51.39
280	GW_Liz_GBS_Liz5195	GW_Liz_GBS_Liz5195	ML	lud	lud_ML	51.4
281	GW_Liz_GBS_Liz5197	GW_Liz_GBS_Liz5197	ML	lud	lud_ML	51.41
282	GW_Liz_GBS_Liz5199	GW_Liz_GBS_Liz5199	ML	lud_chick	lud_ML	51.42
283	GW_Liz_GBS_Liz6002	GW_Liz_GBS_Liz6002	ML	lud	lud_ML	51.43
284	GW_Liz_GBS_Liz6006	GW_Liz_GBS_Liz6006	ML	lud	lud_ML	51.44
285	GW_Liz_GBS_Liz6008	GW_Liz_GBS_Liz6008	ML	lud	lud_ML	51.45
286	GW_Liz_GBS_Liz6009	GW_Liz_GBS_Liz6009	ML	lud	lud_ML	51.46
287	GW_Liz_GBS_Liz6010	GW_Liz_GBS_Liz6010	ML	lud	lud_ML	51.47
288	GW_Liz_GBS_Liz6012	GW_Liz_GBS_Liz6012	ML	lud	lud_ML	51.48

289	GW_Liz_GBS_Liz6014	GW_Liz_GBS_Liz6014	ML	lud	lud_ML	51.49
290	GW_Liz_GBS_Liz6055	GW_Liz_GBS_Liz6055	ML	lud	lud_ML	51.5
291	GW_Liz_GBS_Liz6057	GW_Liz_GBS_Liz6057	ML	lud	lud_ML	51.51
292	GW_Liz_GBS_Liz6060	GW_Liz_GBS_Liz6060	ML	lud	lud_ML	51.52
293	GW_Liz_GBS_Liz6062	GW_Liz_GBS_Liz6062	ML	lud	lud_ML	51.53
294	GW_Liz_GBS_Liz6063	GW_Liz_GBS_Liz6063	ML	lud	lud_ML	51.54
295	GW_Liz_GBS_Liz6066	GW_Liz_GBS_Liz6066	ML	lud	lud_ML	51.55
296	GW_Liz_GBS_Liz6072	GW_Liz_GBS_Liz6072	ML	lud	lud_ML	51.56
297	GW_Liz_GBS_Liz6079	GW_Liz_GBS_Liz6079	ML	lud	lud_ML	51.57
298	GW_Liz_GBS_Liz6203	GW_Liz_GBS_Liz6203	ML	lud_chick	lud_ML	51.58
299	GW_Liz_GBS_Liz6204	GW_Liz_GBS_Liz6204	ML	lud_chick	lud_ML	51.59
300	GW_Liz_GBS_Liz6461	GW_Liz_GBS_Liz6461	ML	lud	lud_ML	51.6
301	GW_Liz_GBS_Liz6472	GW_Liz_GBS_Liz6472	ML	lud	lud_ML	51.61
302	GW_Liz_GBS_Liz6478	GW_Liz_GBS_Liz6478	ML	lud	lud_ML	51.62
303	GW_Liz_GBS_Liz6766	GW_Liz_GBS_Liz6766	ML	lud	lud_ML	51.63
304	GW_Liz_GBS_Liz6776	GW_Liz_GBS_Liz6776	ML	lud	lud_ML	51.64
305	GW_Liz_GBS_Liz6794	GW_Liz_GBS_Liz6794	ML	lud	lud_ML	51.65
306	GW_Liz_GBS_P_fusc	GW_Liz_GBS_P_fusc	fusc	fusc	fusc	101.0
307	GW_Liz_GBS_P_h_man	GW_Liz_GBS_P_h_man	hmand	hmand	hmand	102.0
308	GW_Liz_GBS_P_humei	GW_Liz_GBS_P_humei	hume	hume	hume	103.0
309	GW_Liz_GBS_P_inor	GW_Liz_GBS_P_inor	inor	inor	inor	104.0
310	GW_Liz_GBS_S_burk	GW_Liz_GBS_S_burk	burk	burk	burk	105.0

GOOD NEWS: names of individuals in metadata file and genotype ind file match perfectly.

Filtering

Filter specific individuals

If there are certain individuals that we want to filter out prior to any additional analysis, we can do so here by setting filter to `true` and specifying the individual row numbers in `filter_out_inds`:

```
filter = false
filter_out_inds = [] # if filtering out individuals, specify their row number here, e.g. "[20, 103]"
if filter
    # Specify individuals to filter out:
    ind_with_metadata_indFiltered = ind_with_metadata[Not(filter_out_inds), :]
    geno_indFiltered = geno[Not(filter_out_inds), :]
    println("Specific individuals filtered out as requested")
```

```
else
    ind_with_metadata_indFiltered = ind_with_metadata
    geno_indFiltered = geno
    println("No specific individuals requested to be filtered out")
end
```

```
No specific individuals requested to be filtered out
```

Filter individuals based on missing genotypes

Here we determine number of missing SNPs per bird (40% for this round), and filter out those with more than a certain percent of missing SNPs:

```
SNPmissing_percent_allowed_per_ind = 40   # this is the percentage threshold
threshold_missing = loci_count * SNPmissing_percent_allowed_per_ind/100
numMissings = sum(geno .== -1, dims=2)
selection = vec(numMissings .<= threshold_missing) # the vec command converts to BitVector rather tha
geno_indFiltered = geno_indFiltered[selection, :]
println("Filtering out these individuals based on too many missing genotypes: ", ind_with_metadata_i
ind_with_metadata_indFiltered = ind_with_metadata_indFiltered[selection, :];
```

```
Filtering out these individuals based on too many missing genotypes: ["GW_Armando_plate1_JG08G02", "GW_Arma
```

Filter SNPs with too many missing genotypes:

```
# (remember that first column is arbitrary row number in input file)
missing_genotypes_per_SNP = sum(geno_indFiltered .== -1, dims=1)
missing_genotypes_percent_allowed_per_site = 5   # this is the percentage threshold
threshold_genotypes_missing = size(geno_indFiltered)[1] * missing_genotypes_percent_allowed_per_site
selection = vec(missing_genotypes_per_SNP .<= threshold_genotypes_missing)
geno_ind_SNP_filtered = geno_indFiltered[:, selection]
pos_SNP_filtered = pos_whole_genome[selection[Not(1)],:]  # the Not(1) is needed because first colum
println("Started with ", size(geno_indFiltered, 2)-1, " SNPs. After filtering SNPs for no more than
```

```
Started with 2431709 SNPs. After filtering SNPs for no more than 5% missing genotypes, 1003400 SNPs remain.
```

2nd round of filtering individuals

I added this in August 2023, to improve accuracy of imputation-based PCA, because I noticed outliers tended to have more missing data. Now I only allow up to 10% missing SNPs per individual.

```
SNPmissing_percent_allowed_per_ind_round2 = 10   # this is the percentage threshold
threshold_missing = (size(geno_ind_SNP_filtered, 2) - 1) * SNPmissing_percent_allowed_per_ind_round2
numMissings = sum(geno_ind_SNP_filtered .== -1, dims=2)
selection = vec(numMissings .<= threshold_missing) # the vec command converts to BitVector rather tha
geno_ind_SNP_ind_filtered = geno_ind_SNP_filtered[selection, :]
# print filtered out individuals:
ind_with_metadata_indFiltered.ind[selection.==false]
println("Filtering out these individuals based on too many missing genotypes: ", ind_with_metadata_i
# filtered out: ["GW_Armando_plate1_TTGW74", "GW_Armando_plate2_TTGW54", "GW_Lane5_AA8", "GW_Lane5_YI
ind_with_metadata_indFiltered = ind_with_metadata_indFiltered[selection, :]
println("This leaves ", size(geno_ind_SNP_ind_filtered, 1), " individuals and ", size(geno_ind_SNP_i
```

```
Filtering out these individuals based on too many missing genotypes: ["GW_Armando_plate1_TTGW74", "GW_Arman
This leaves 267 individuals and 1003400 loci, with no individuals missing more than 10% of genotypes and no l
```

Estimate relationships of individuals using PCA

Our goal is to produce plots showing individuals in genotype space, using Principal Components Analysis. First we need to do a couple changes to our data matrix:

For missing genotypes, change our code of `-1` to `missing`:

```
genos_with_missing = Matrix{Union{Missing, Int32}}(geno_ind_SNP_ind_filtered)
genos_with_missing[genos_with_missing .== -1] .= missing;
```

Remove the first column of the genotype matrix (which was an initial row number):

```
genosOnly = genos_with_missing[:,Not(1)]
```

```
267×1003400 Matrix{Union{Missing, Int32}}:
 0  0  0  1  0  0  0  0  0  0  0  0  0  …  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  …  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  …  0  0  0  0  0  0  0  0  0  0  0  1
0  0  0  0  0  0  0  0  1  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
⋮              ⋮              ⋮        ⋱        ⋮              ⋮
0  0  0  0  0  0  0  0  0  0  0  0  0  …  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  …  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  …  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0     0  0  0  0  0  0  0  0  0  0  0  0
```

Impute and save genotypes for each scaffold

PCA requires imputation of missing genotypes. I did imputation for each scaffold above a certain size threshold. Those scaffolds (many of which correspond to whole chromosomes) are listed here:

```
chromosomes_to_process = ["gw2",
                          "gw1",
                          "gw3",
                          "gwZ",
                          "gw1A",
                          "gw4",
                          "gw5",
                          "gw7",
                          "gw6",
                          "gw8",
                          "gw9",
                          "gw11",
                          "gw12",
                          "gw10",
                          "gw13",
```

```
                    "gw14",
                    "gw18",
                    "gw20",
                    "gw15",
                    "gw1B",
                    "gws100",
                    "gw17",
                    "gw19",
                    "gws101",
                    "gw4A",
                    "gw21",
                    "gw26",
                    "gws102",
                    "gw23",
                    "gw25",
                    "gws103",
                    "gw22",
                    "gws104",
                    "gw28",
                    "gw27",
                    "gw24",
                    "gws105",
                    "gws106",
                    "gws107",
                    "gws108",
                    "gws109",
                    "gws110",
                    "gws112"];
```

Imputation can take several minutes per scaffold, so I ran this imputation step separately from this Quarto notebook (otherwise render would take long) and saved the genotype data for each scaffold for loading in the next step. This is the code I used for imputing:

```
for i in eachindex(chromosomes_to_process)
    chrom = chromosomes_to_process[i]
    regionText = string("chr", chrom)
    loci_selection = (pos_SNP_filtered.chrom .== chrom)
    pos_SNP_filtered_region = pos_SNP_filtered[loci_selection,:]
    genosOnly_region_for_imputing = Matrix{Union{Missing, Float32}}(genosOnly[:,loci_selection])
    @time imputed_genos = Impute.svd(genosOnly_region_for_imputing)
    filename = string(baseName, tagName, regionText, ".imputedMissing.jld2")
```

```
    jldsave(filename; imputed_genos, ind_with_metadata_indFiltered, pos_SNP_filtered_region)
    println(string("Chromosome ", chrom, ": Saved real and imputed genotypes for ", size(pos_SNP_fil
end
```

Now we can cycle through a set of chromosomes and plot a PCA for each. We need to first specify some groups to include in the plot, and their colors:

```
groups_to_plot_PCA = ["vir","vir_misID","vir_S","nit", "lud_PK", "lud_KS", "lud_central", "lud_Sath"
group_colors_PCA = ["blue","blue","turquoise1","grey","seagreen4","seagreen3","seagreen2","olivedrab3
```

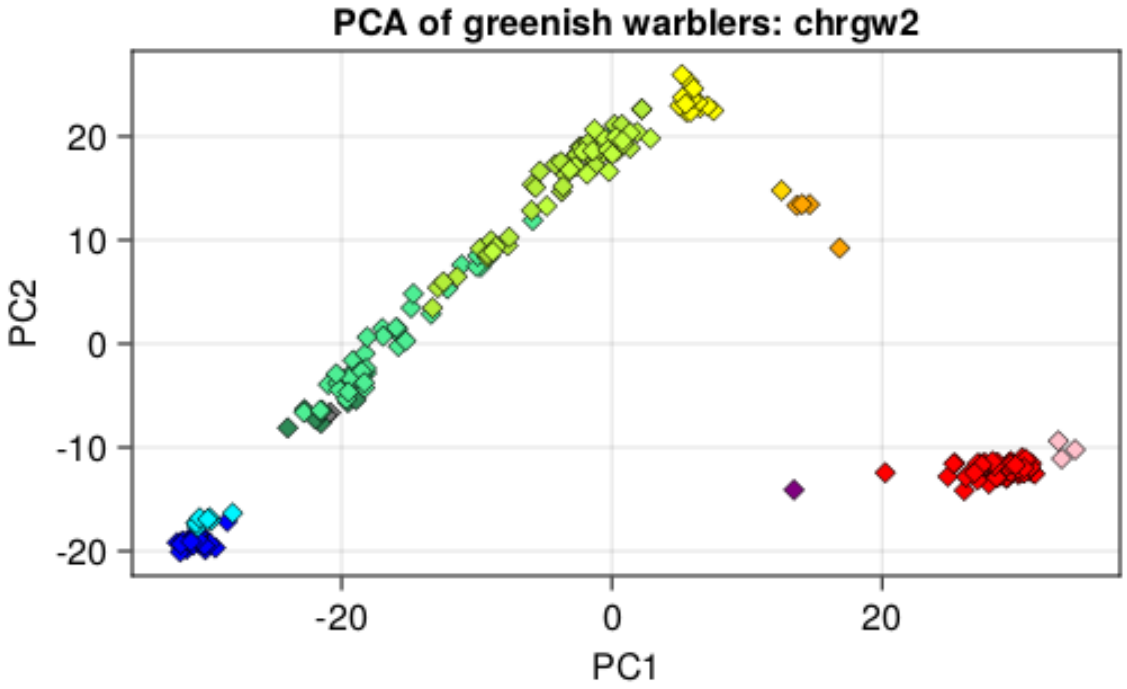
Now we'll actually do the PCA and make the plot for each scaffold:

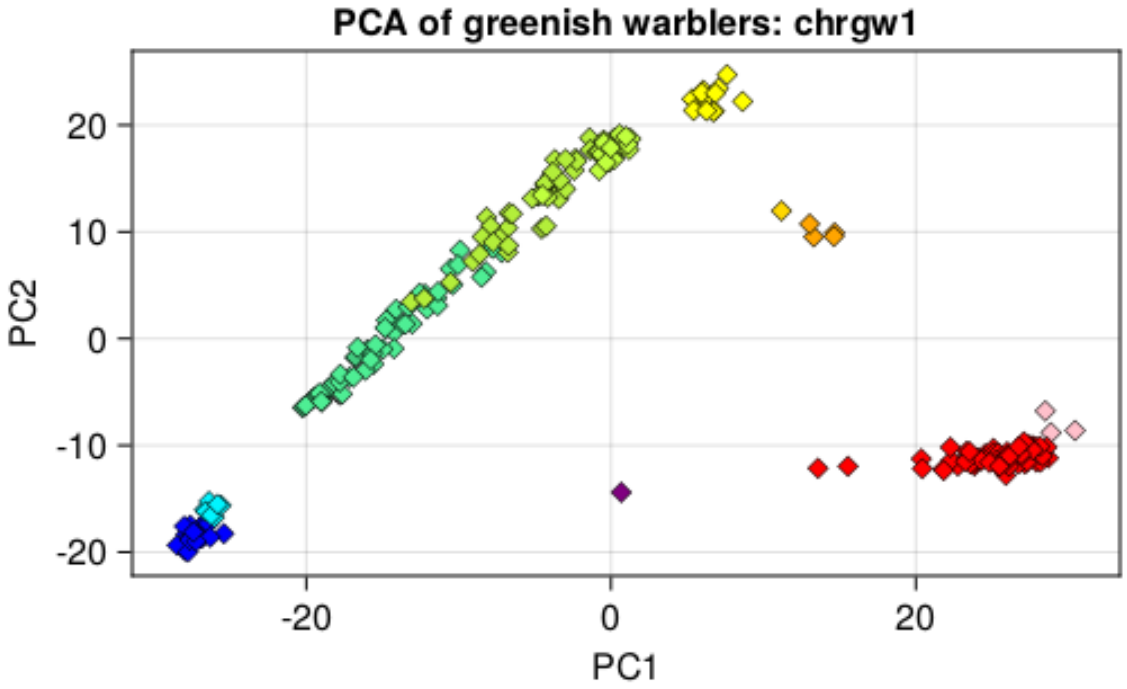
```
for i in eachindex(chromosomes_to_process)
    chrom = chromosomes_to_process[i]
    regionText = string("chr", chrom)
    filename = string(baseName, tagName, regionText, ".imputedMissing.jld2")
    imputed_genos = load(filename, "imputed_genos")
    ind_with_metadata_indFiltered = load(filename, "ind_with_metadata_indFiltered")
    pos_SNP_filtered_region = load(filename, "pos_SNP_filtered_region")
    println(string("Loaded ",filename))
    println(string(regionText, ": ", size(imputed_genos,2), " SNPs from ", size(imputed_genos,1), " i
    plotPCA(imputed_genos, ind_with_metadata_indFiltered,
            groups_to_plot_PCA, group_colors_PCA;
            sampleSet = "greenish warblers", regionText=regionText,
            flip1 = true, flip2 = false)
end
```

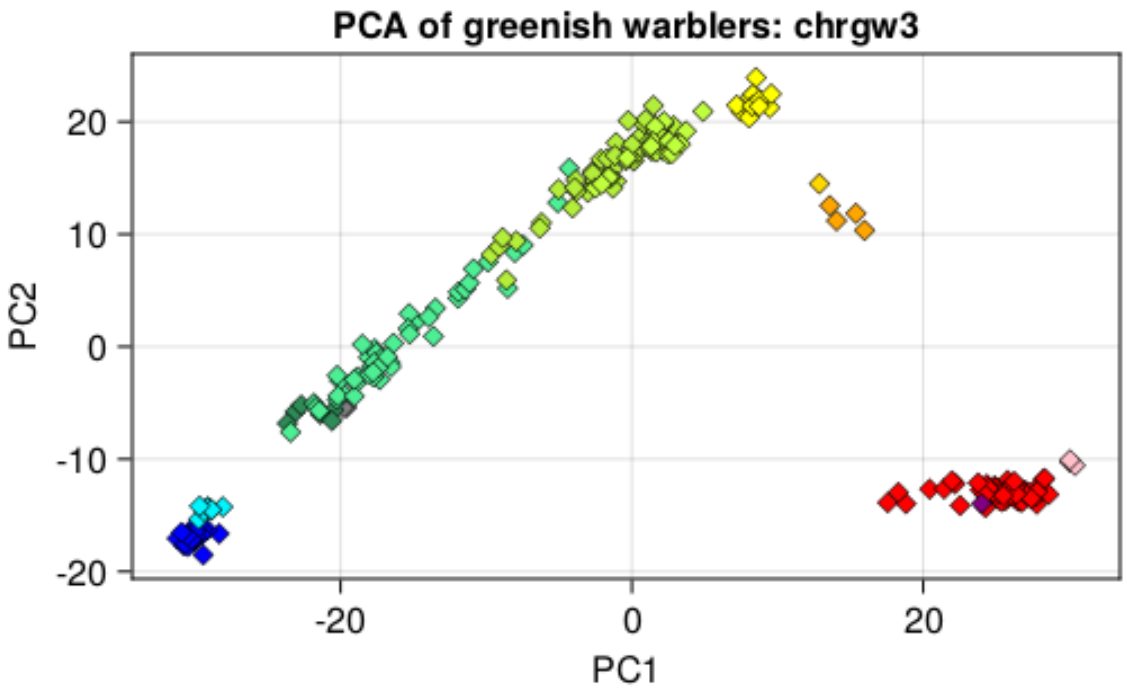
```
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgw2: 91777 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgw1: 79563 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgw3: 81102 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgwZ: 52592 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgw1A: 49350 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgw4: 49196 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
chrgw5: 54499 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.whc
```

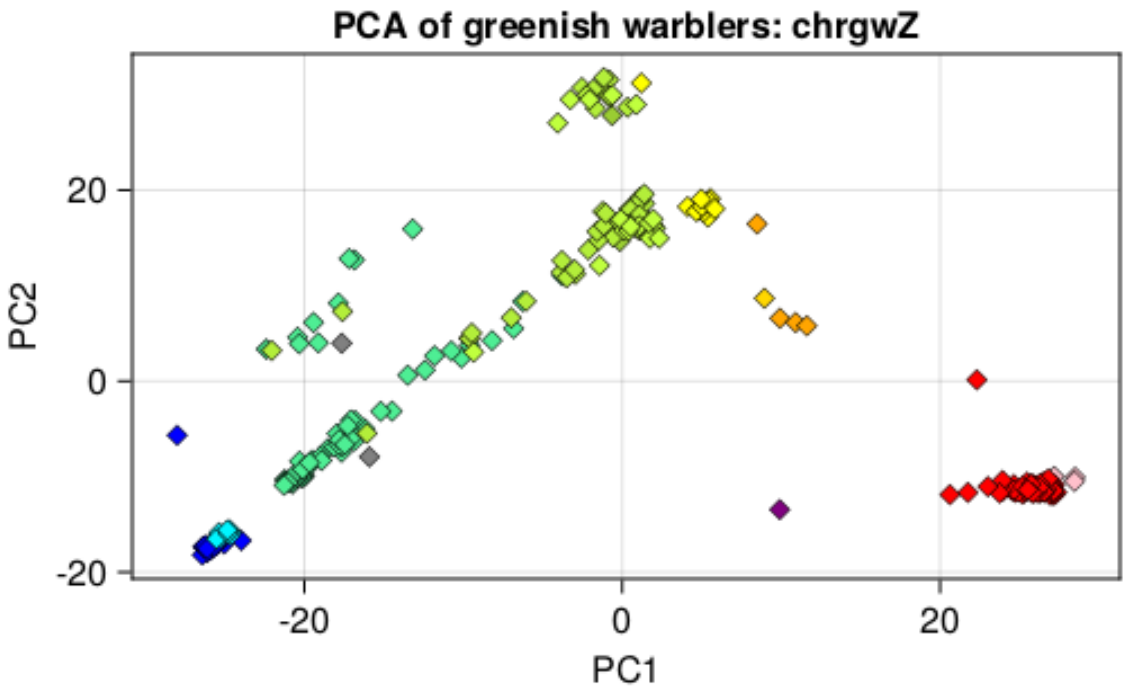
```
chrgw7: 36037 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw6: 39725 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw8: 37331 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw9: 37593 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw11: 27232 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw12: 32811 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw10: 26617 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw13: 33138 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw14: 30686 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw18: 19123 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw20: 32361 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw15: 27257 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw1B: 626 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws100: 182 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw17: 26008 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw19: 25165 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws101: 158 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw4A: 18249 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw21: 13183 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw26: 14101 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws102: 277 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw23: 13825 SNPs from 267 individuals
```

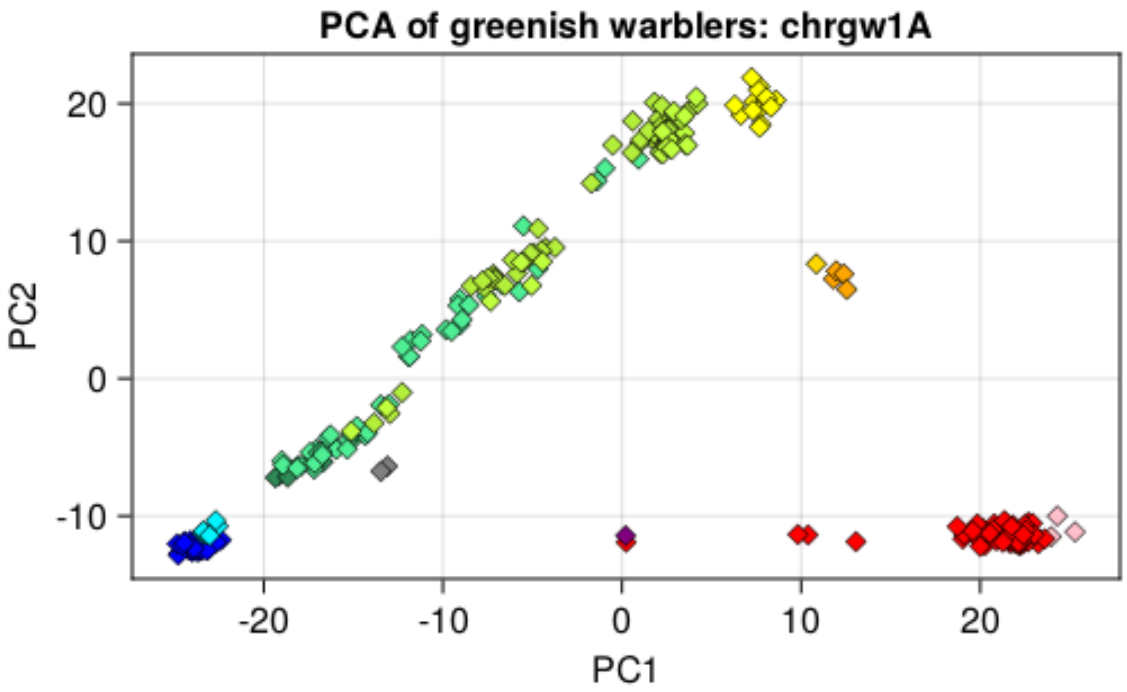
```
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw25: 3704 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws103: 274 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw22: 5416 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws104: 350 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw28: 11072 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw27: 9574 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgw24: 13679 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws105: 469 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws106: 115 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws107: 258 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws108: 160 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws109: 275 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws110: 154 SNPs from 267 individuals
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
chrgws112: 1850 SNPs from 267 individuals
```

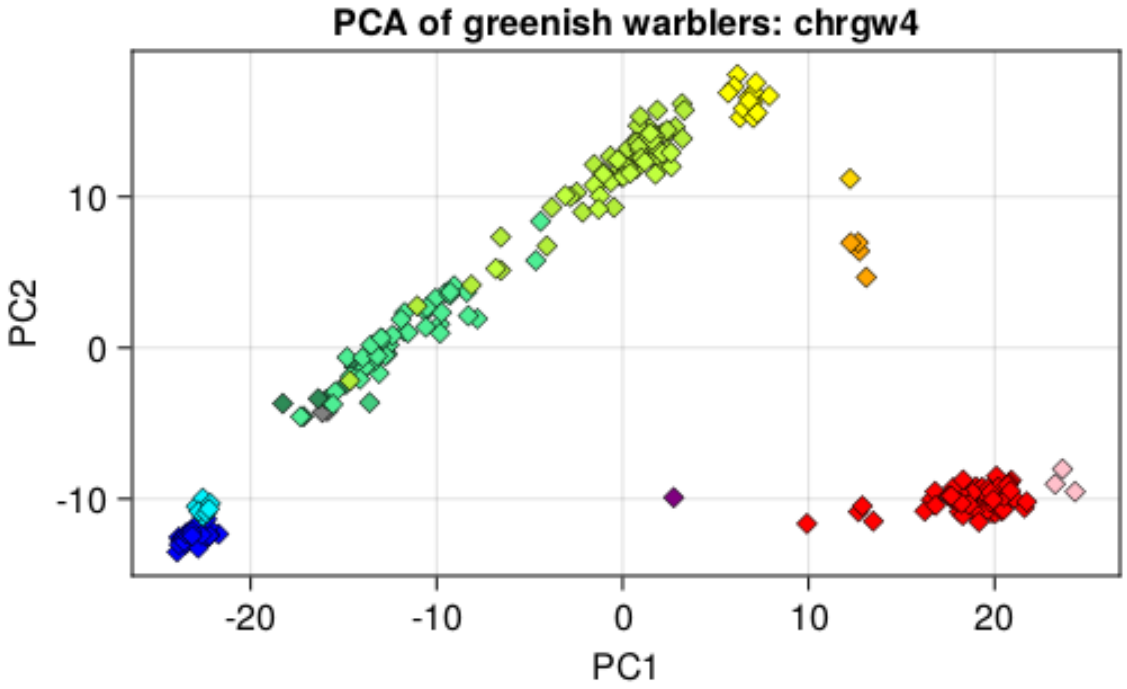

PCA of greenish warblers: chrgw2
PC2
PC1
20
10
0
-10
-20
-20
0
20

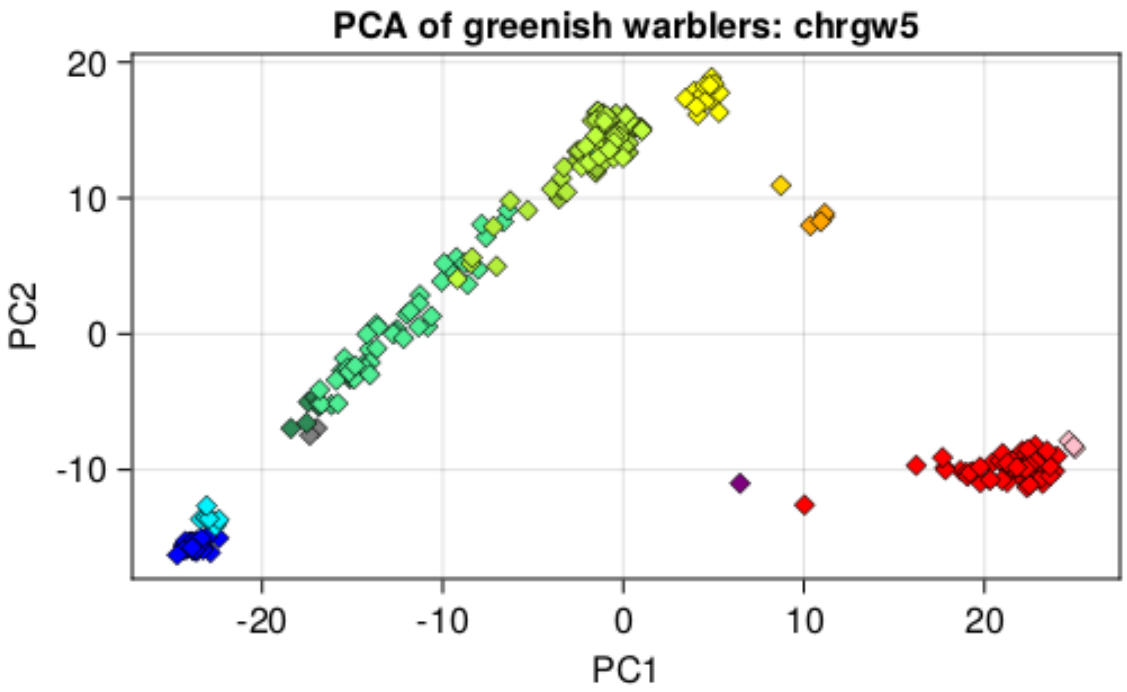

PCA of greenish warblers: chrgw1
PC2
PC1
20
10
0
-10
-20
-20
0
20

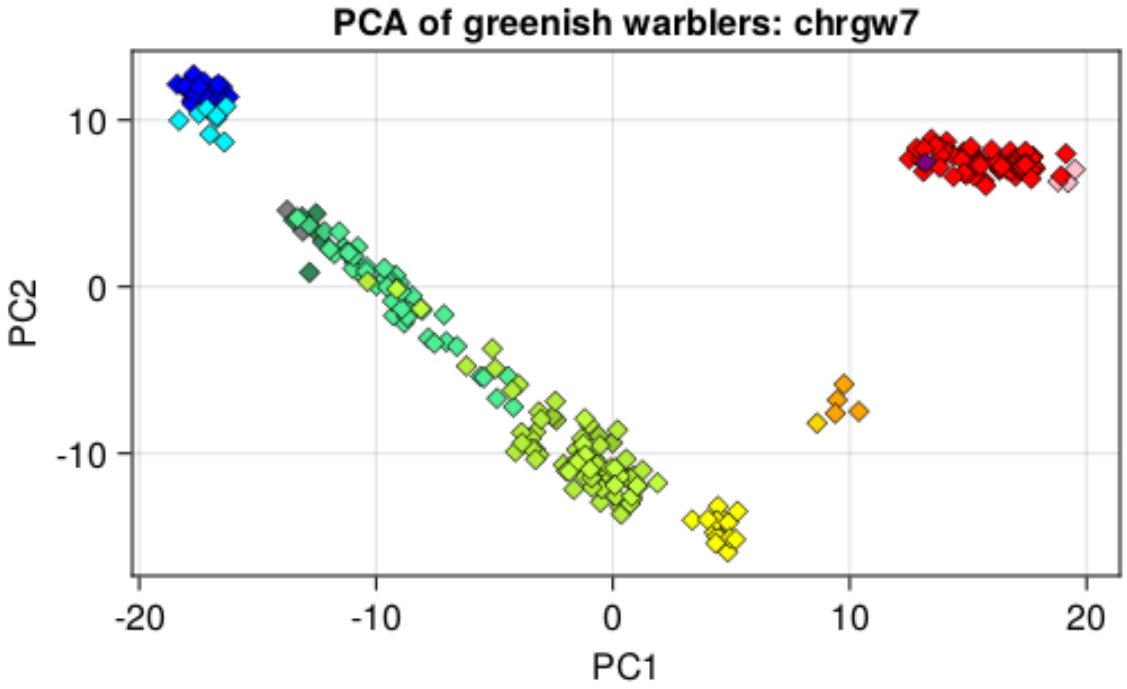

PCA of greenish warblers: chrgw3
PC2
PC1
20
10
0
-10
-20
-20
0
20

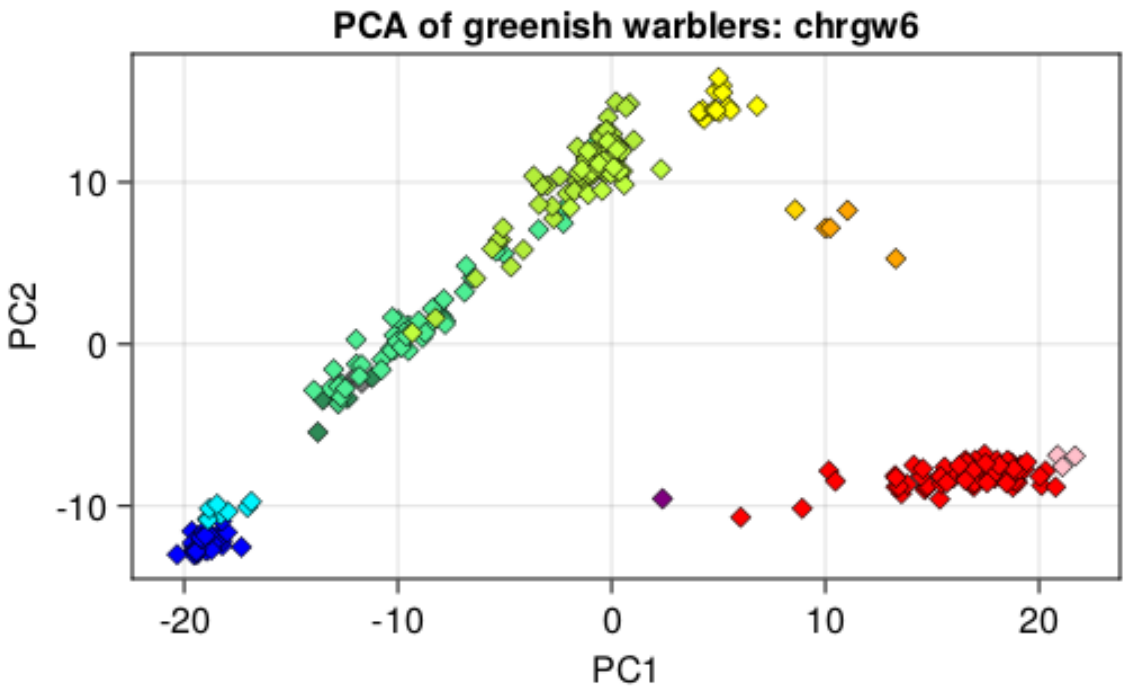

PCA of greenish warblers: chrgwZ
PC2
PC1
20
0
-20
-20
0
20

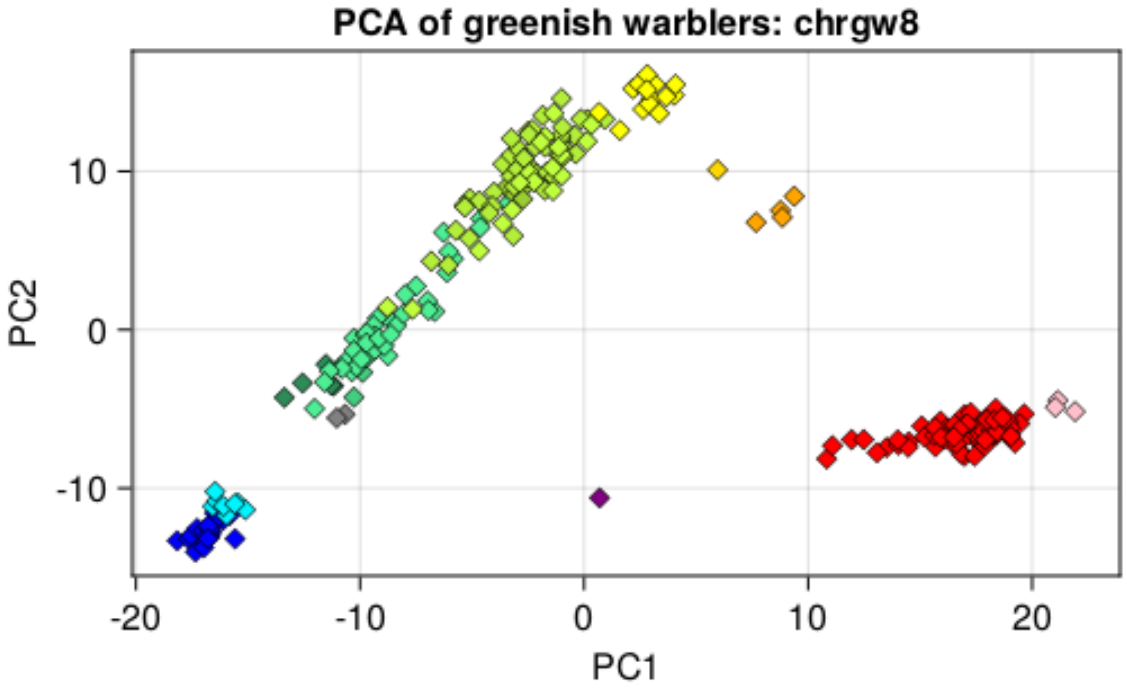

PCA of greenish warblers: chrgw1A
PC2
PC1
20
10
0
-10
-20
-10
0
10
20

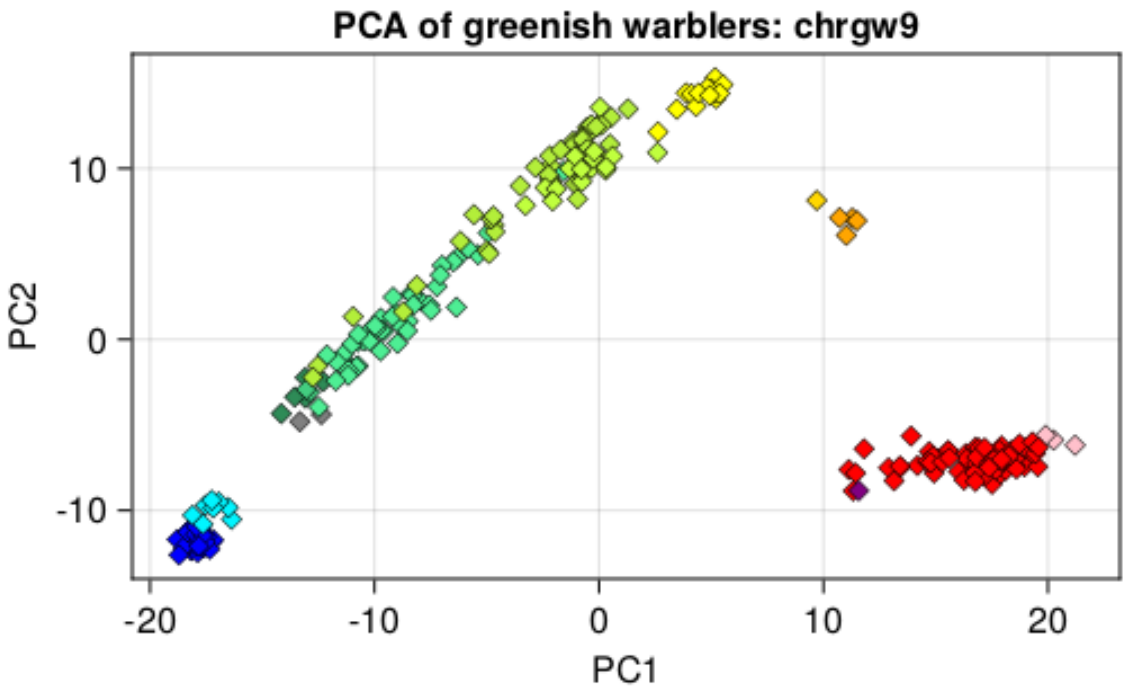

PCA of greenish warblers: chrgw4
PC2
10
0
-10
-20
-10
0
10
20
PC1

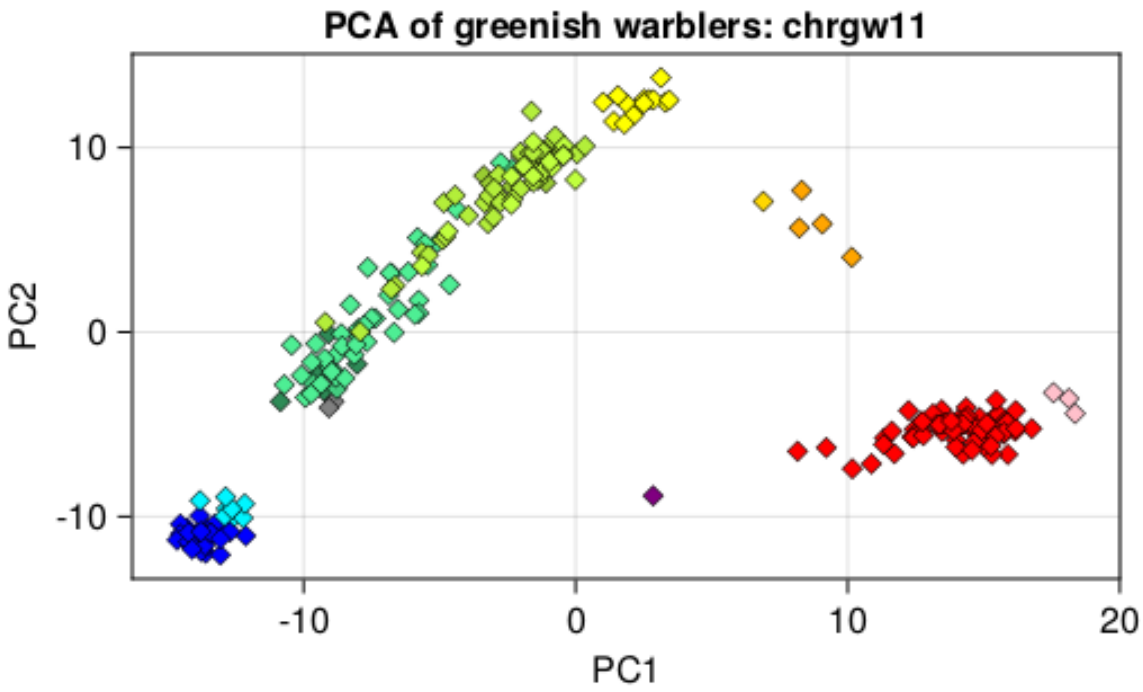

PCA of greenish warblers: chrgw5
PC2
PC1
20
10
0
-10
-20
-10
0
10
20

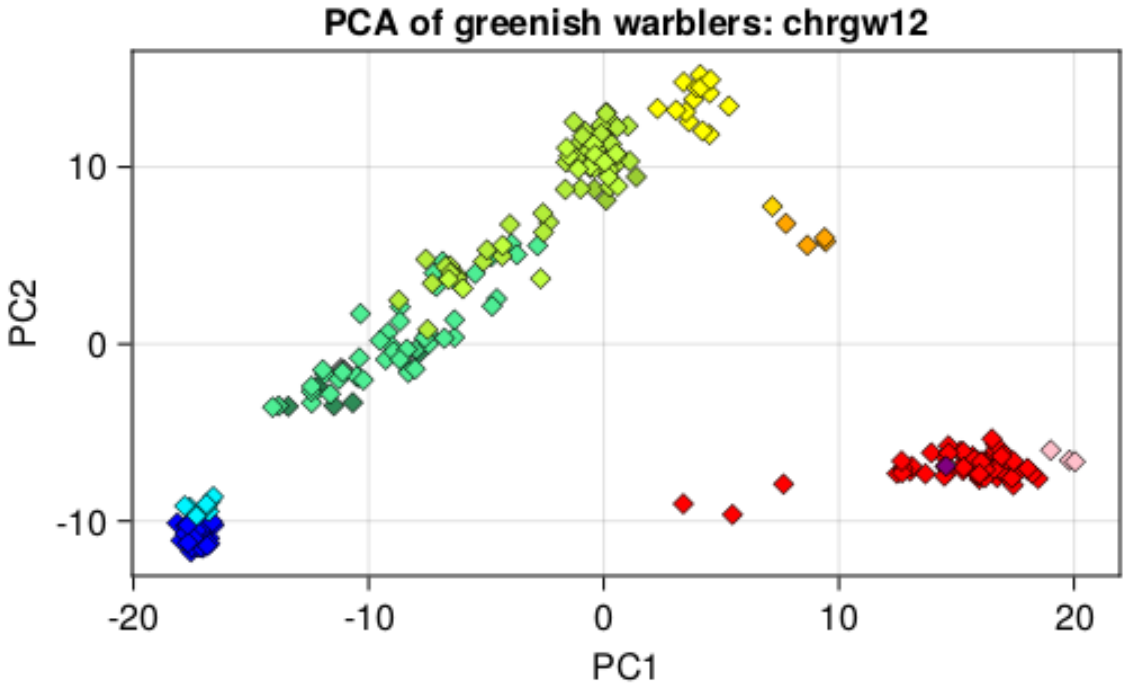

PCA of greenish warblers: chrgw7
10
0
-10
PC2
-20
-10
0
10
20
PC1

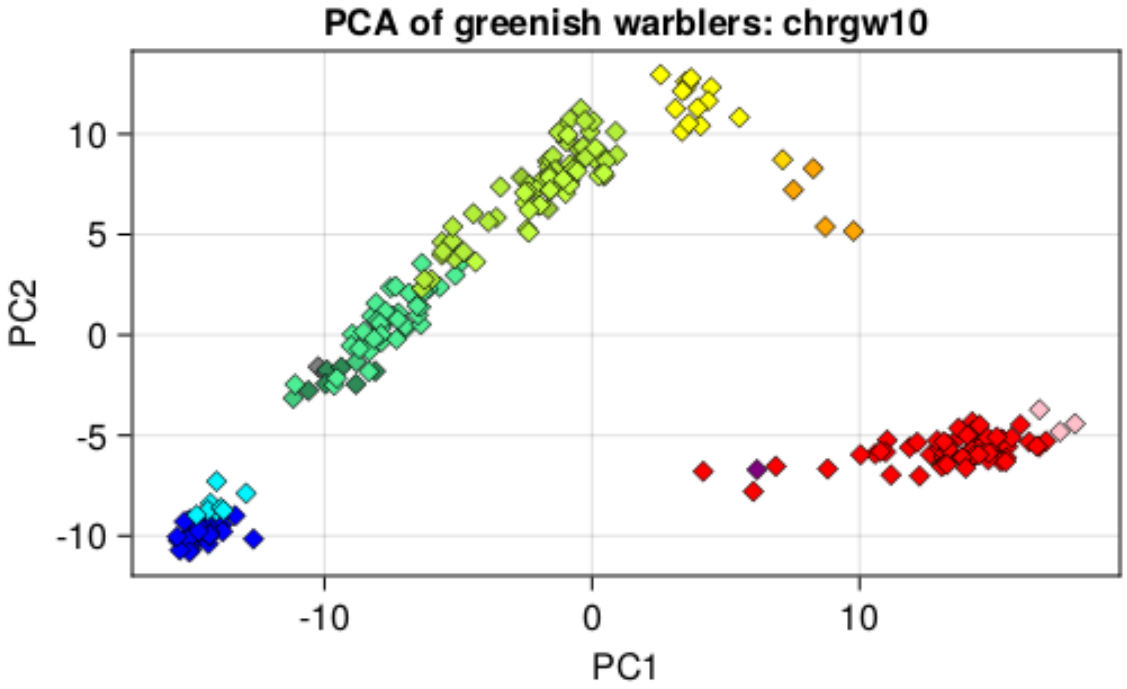

PCA of greenish warblers: chrgw6
PC2
10
0
-10
-20
-10
0
10
20
PC1

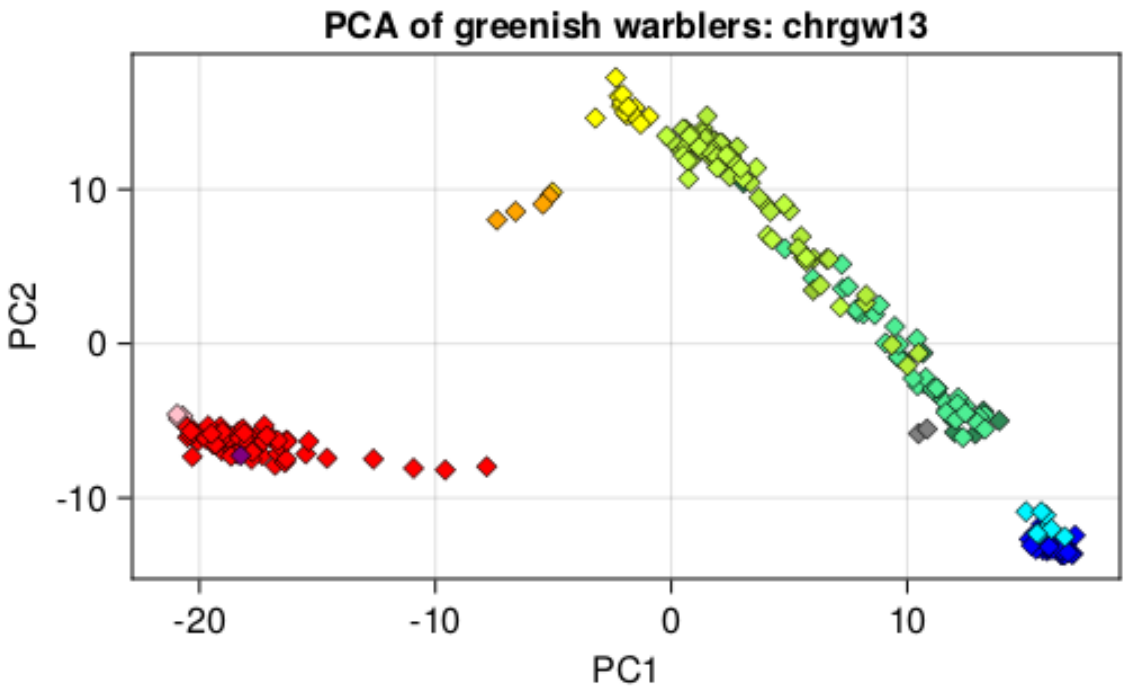

PCA of greenish warblers: chrgw8
PC2
PC1
10
0
-10
-20
-10
0
10
20

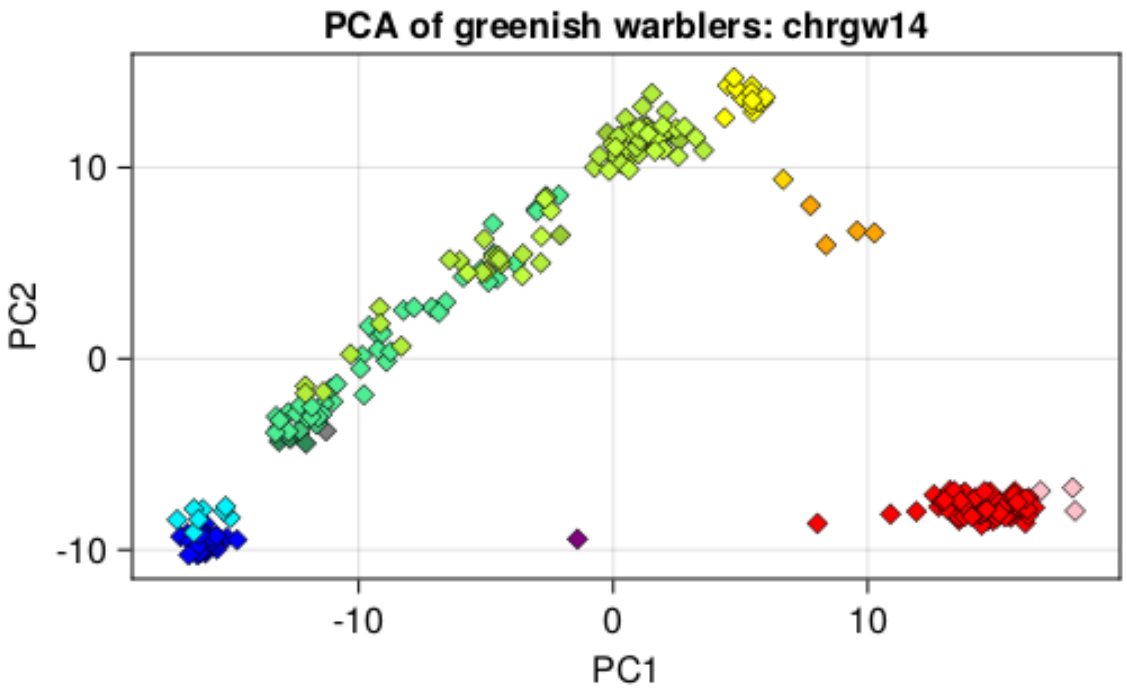

PCA of greenish warblers: chrgw9
PC2
PC1
10
0
-10
-20
-10
0
10
20

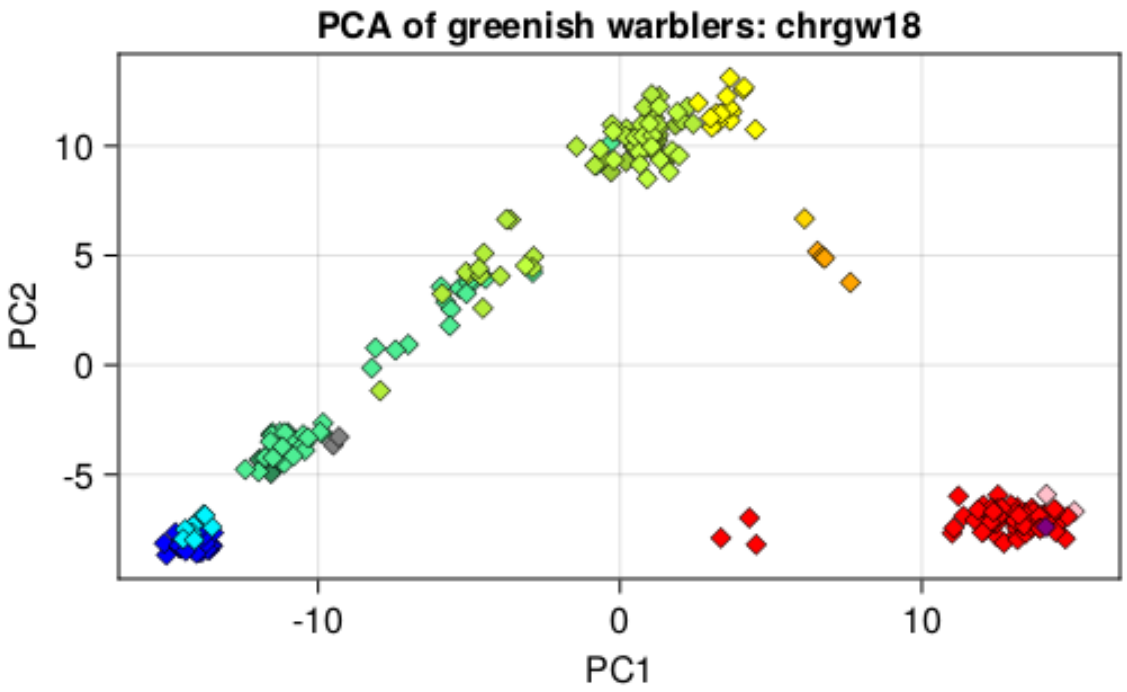

PCA of greenish warblers: chrgw11
10
0
-10
PC2
-10
0
10
20
PC1

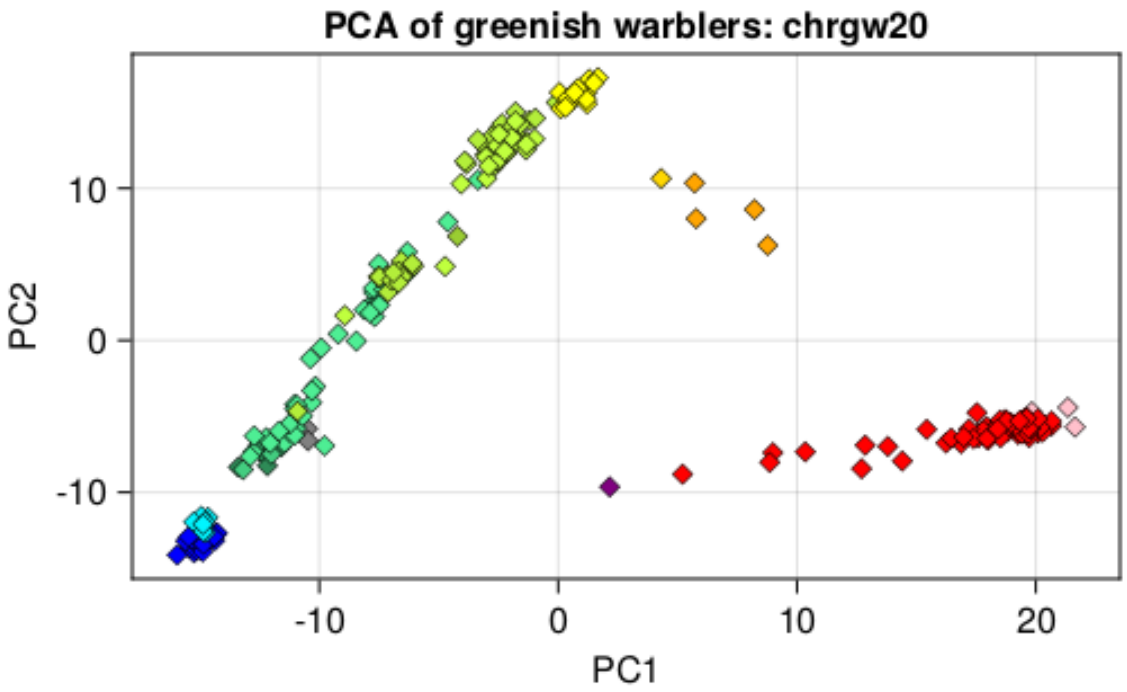

PCA of greenish warblers: chrgw12
PC2
PC1
10
0
-10
-20
-10
0
10
20

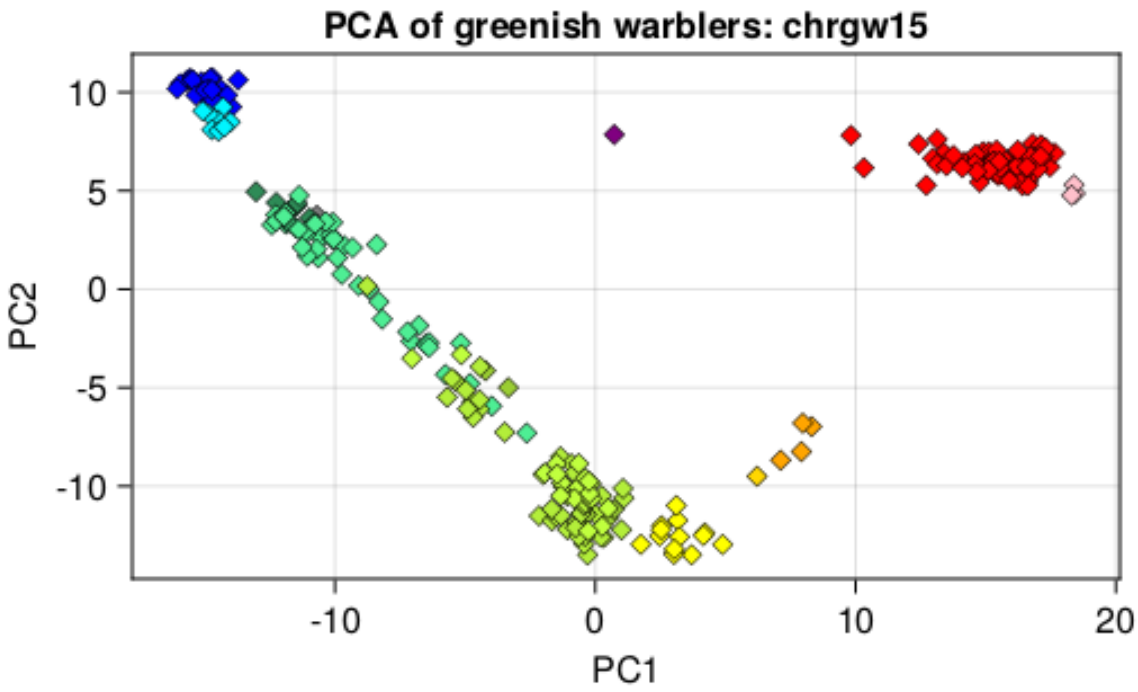

PCA of greenish warblers: chrgw10
PC2
PC1
10
5
0
-5
-10
-10
0
10

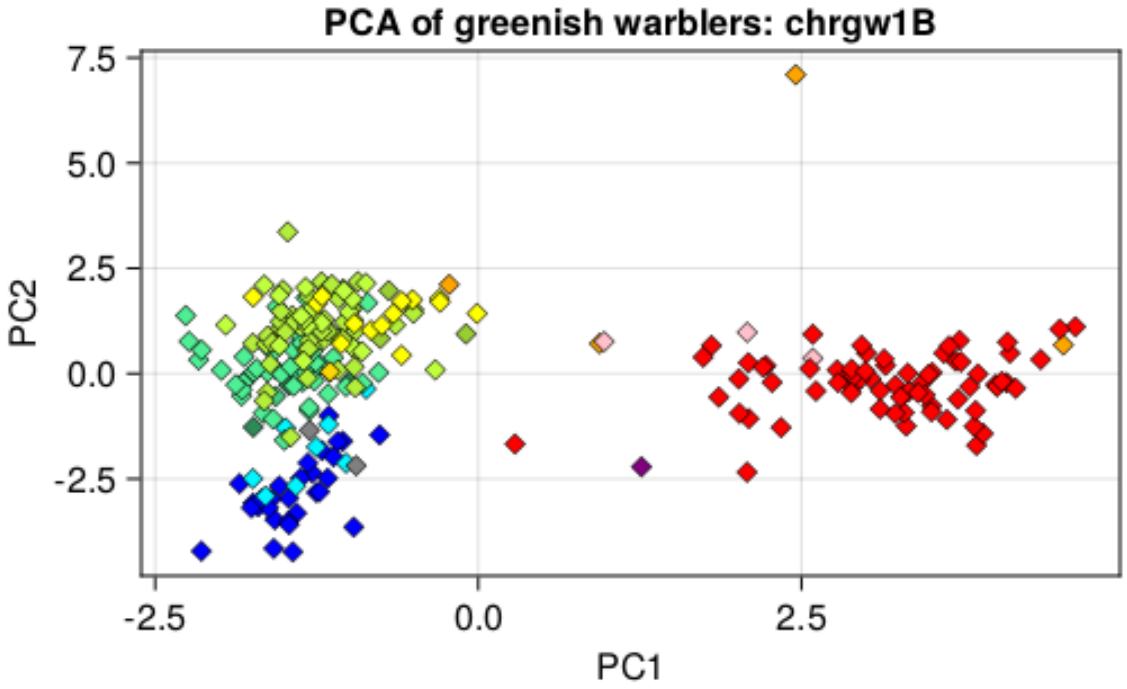

PCA of greenish warblers: chrgw13
PC2
10
0
-10
-20
-10
0
10
PC1

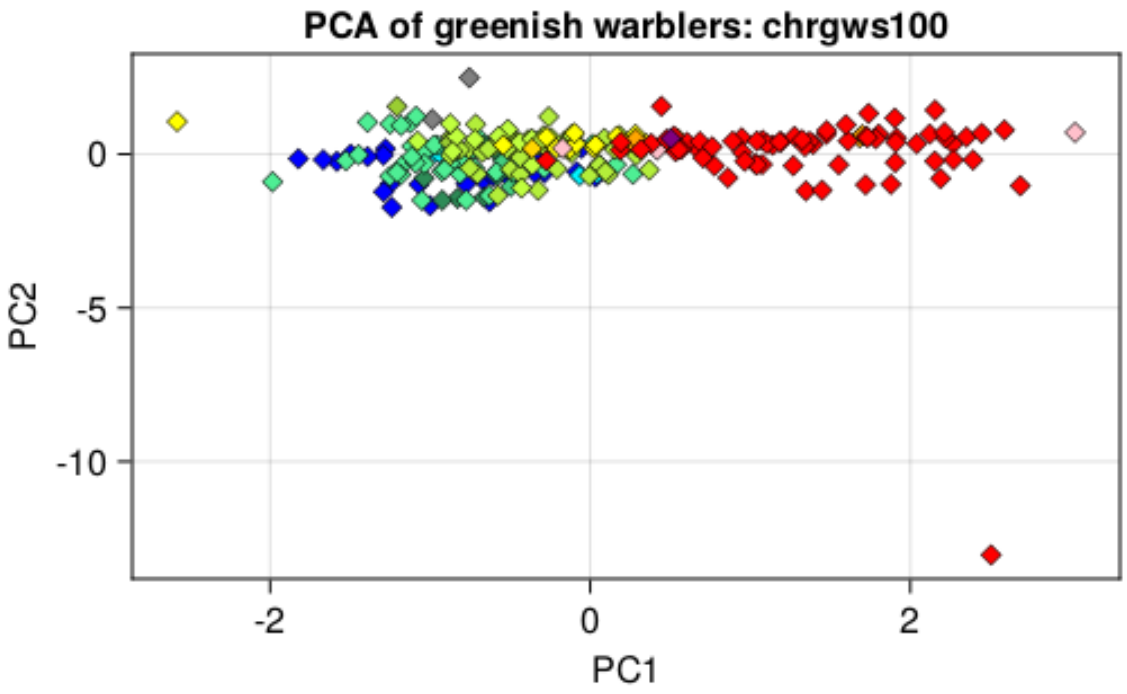

PCA of greenish warblers: chrgw14
PC2
PC1
10
0
-10
-10
0
10

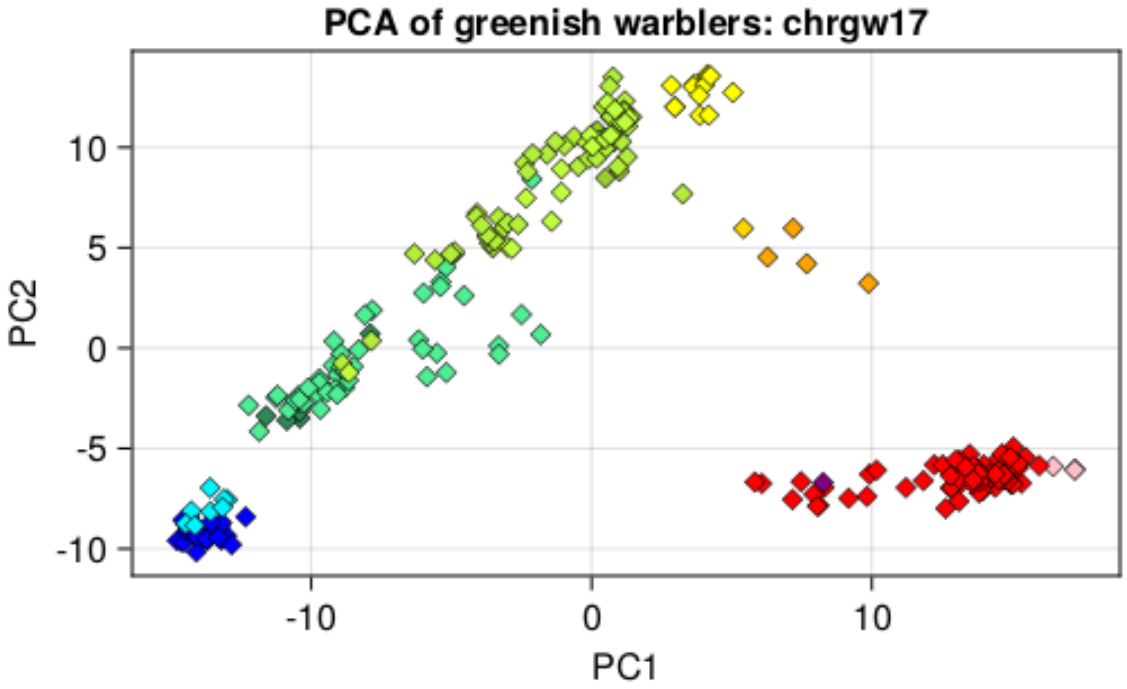

PCA of greenish warblers: chrgw18
10
5
0
-5
PC2
-10
0
10
PC1

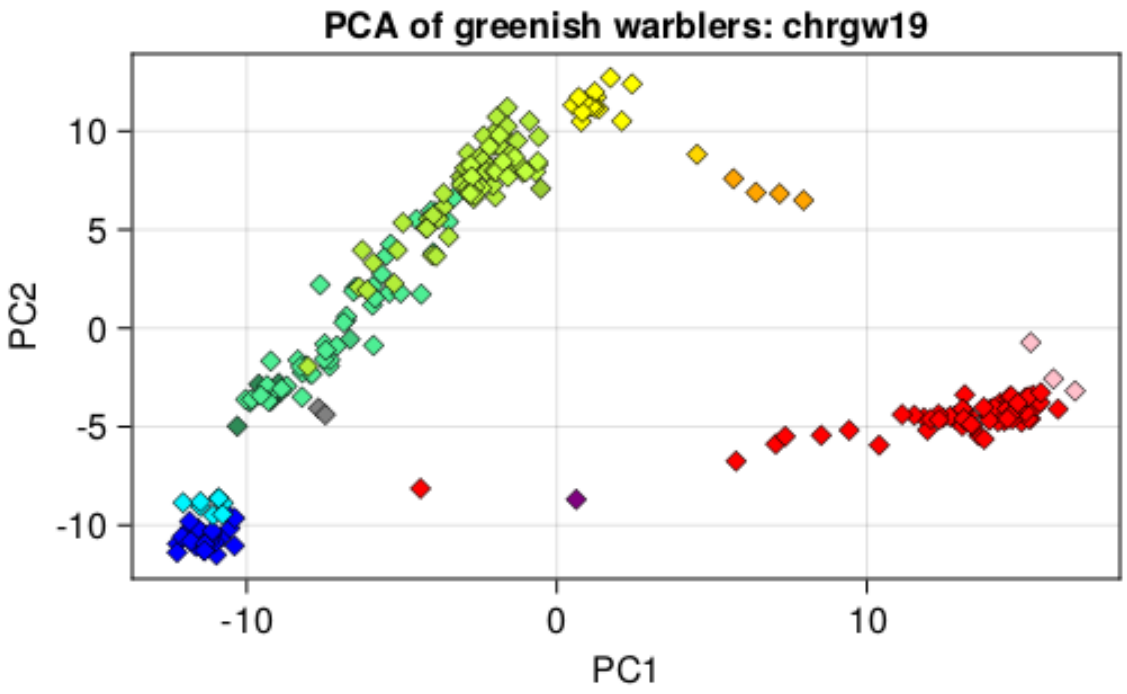

PCA of greenish warblers: chrgw20
PC2
10
0
-10
-10
0
10
20
PC1

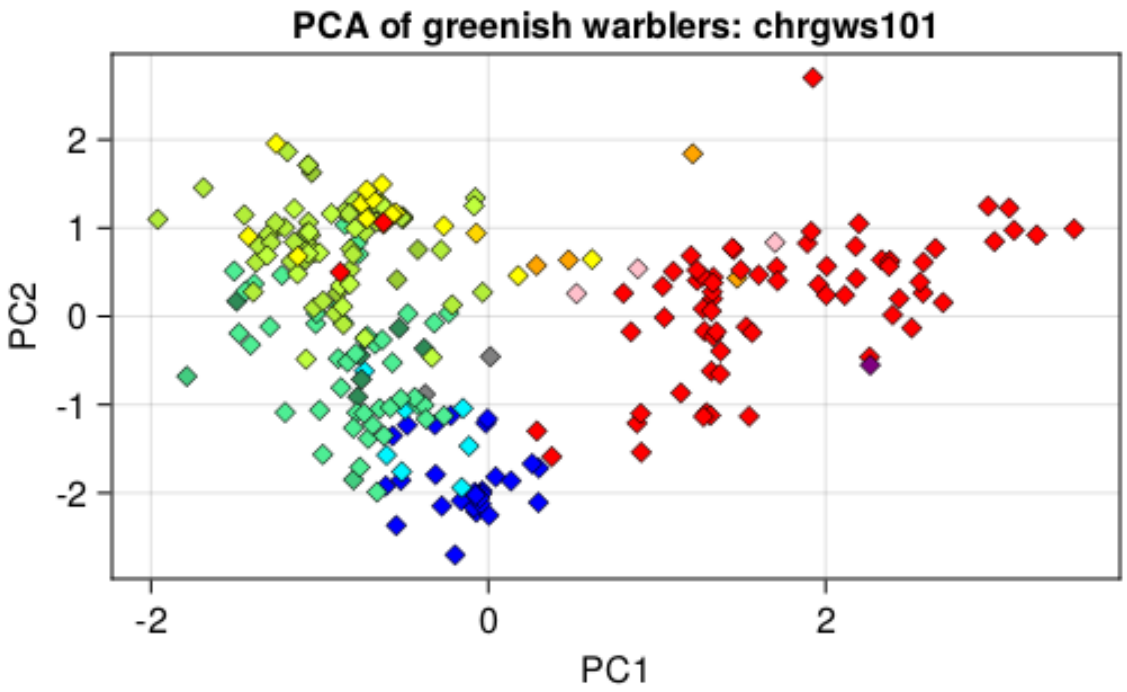

PCA of greenish warblers: chrgw15
PC2
PC1
10
5
0
-5
-10
-10
0
10
20

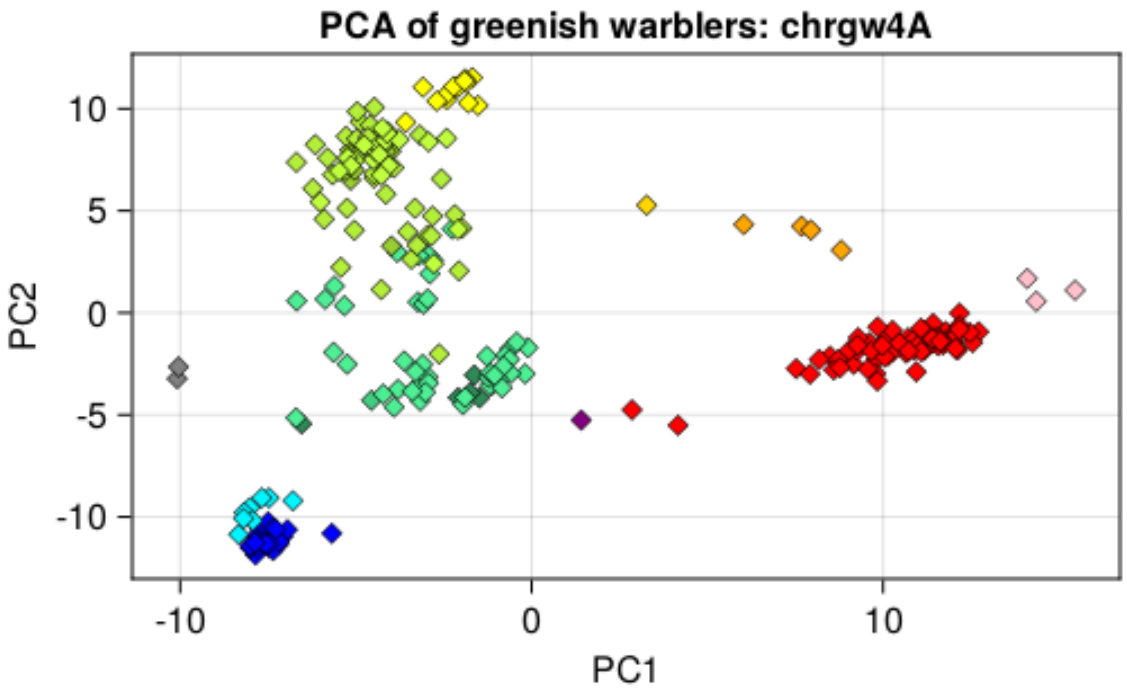

PCA of greenish warblers: chrgw1B
7.5
5.0
2.5
0.0
-2.5
PC2
-2.5
0.0
2.5
PC1

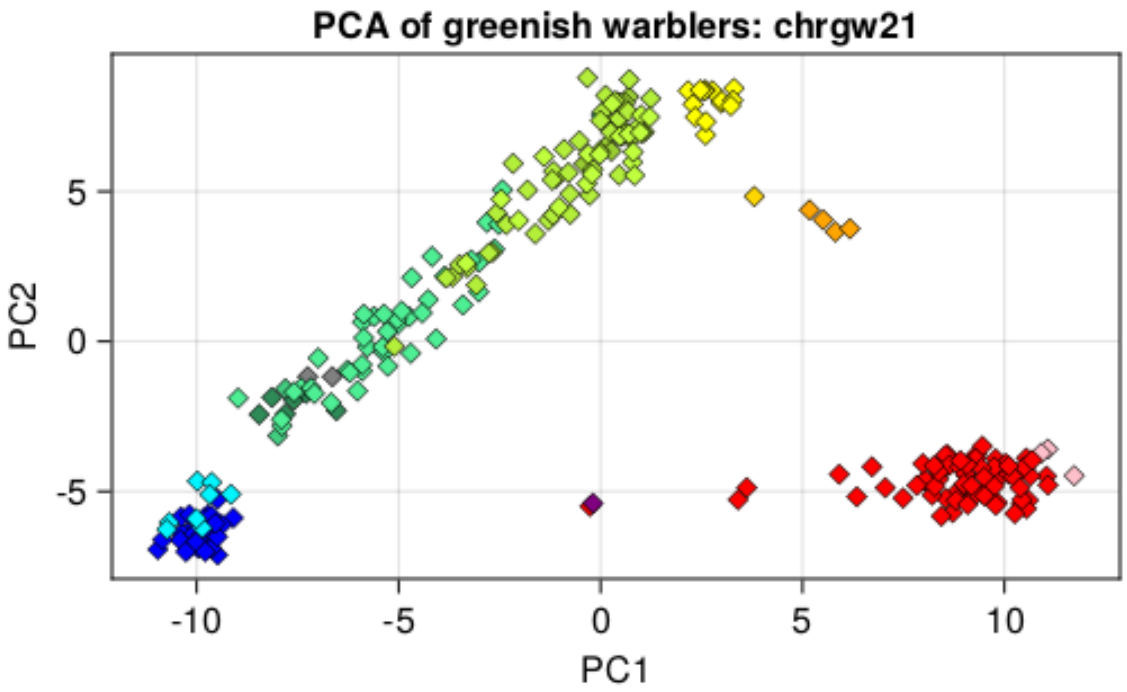

PCA of greenish warblers: chrgws100
PC2
0
-5
-10
-2
0
2
PC1

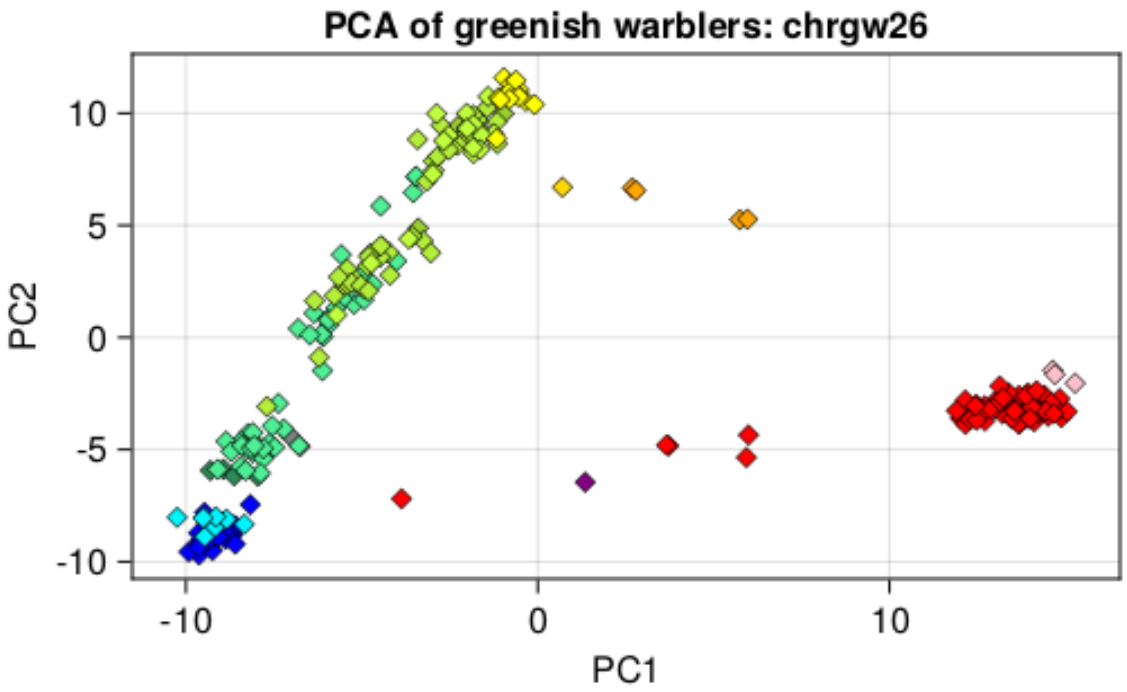

PCA of greenish warblers: chrgw17
PC2
PC1
10
5
0
-5
-10
-10
0
10

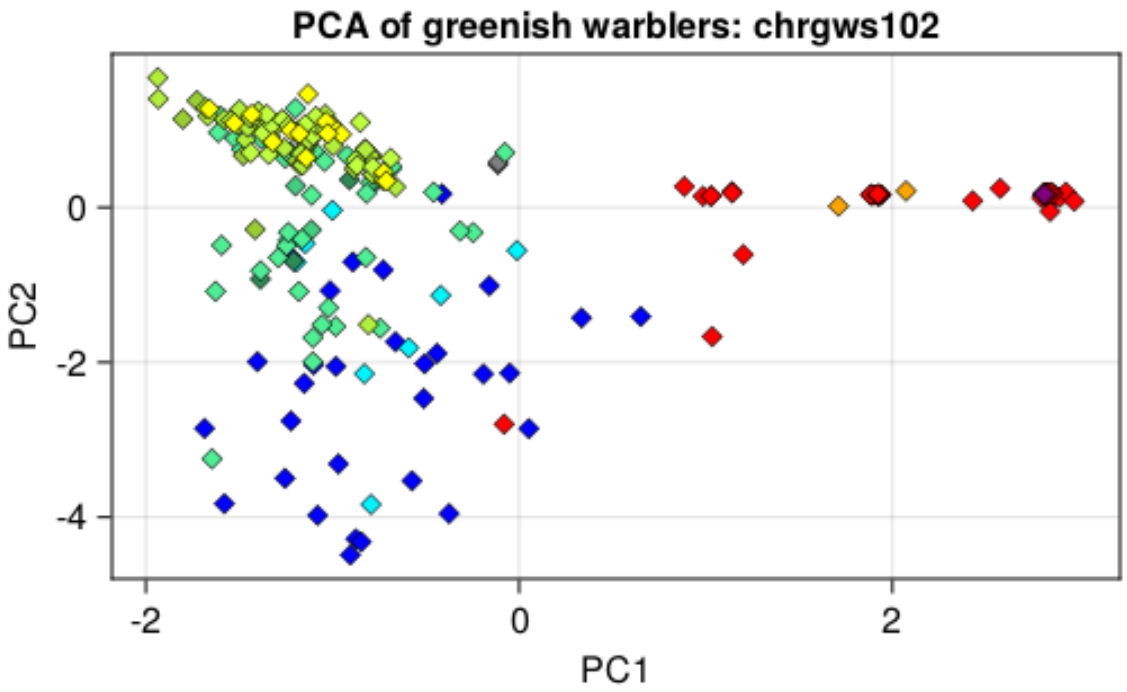

PCA of greenish warblers: chrgw19
PC2
PC1
10
5
0
-5
-10
-10
0
10

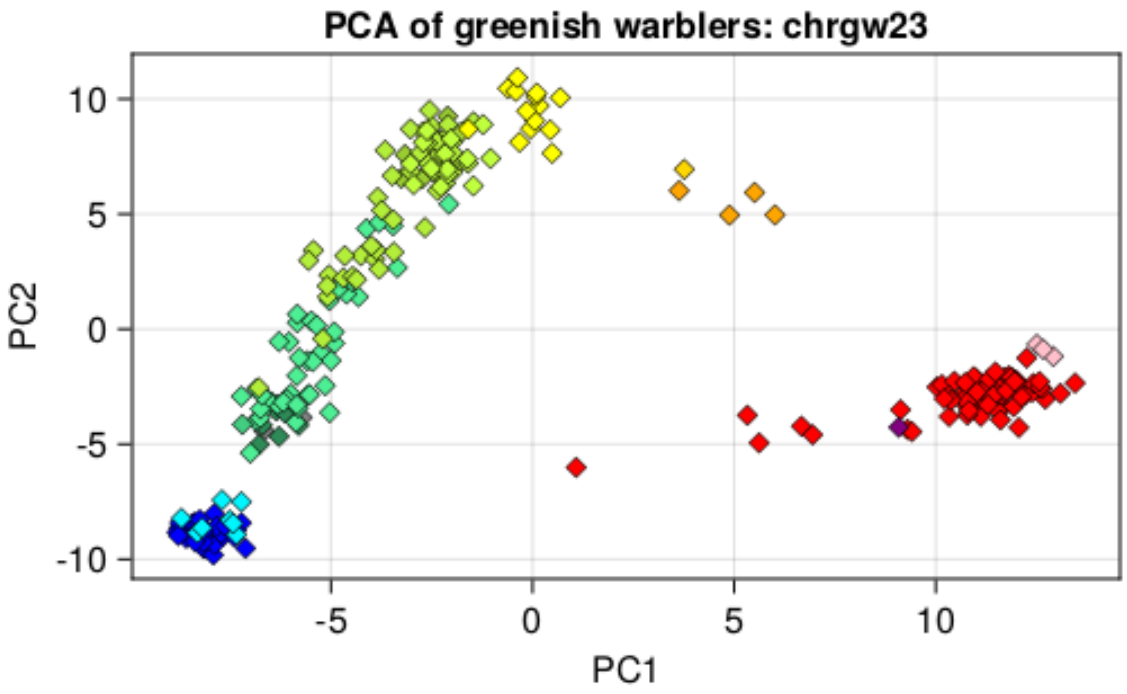

PCA of greenish warblers: chrgws101
PC2
PC1
2
1
0
-1
-2
-2
0
2

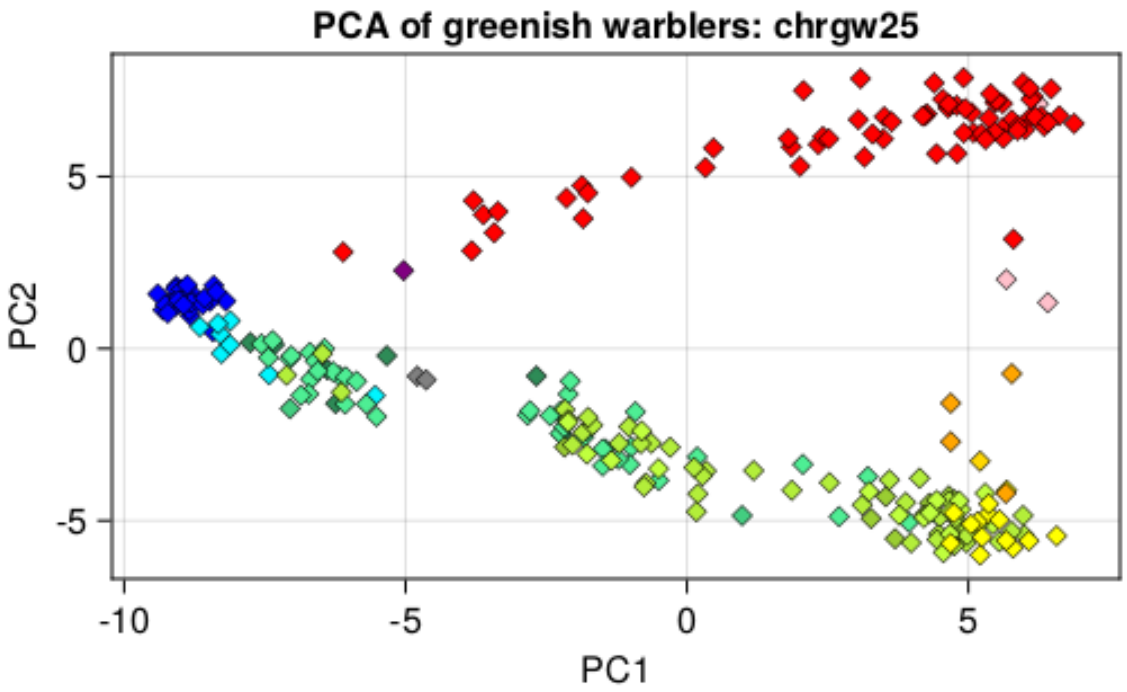

PCA of greenish warblers: chrgw4A
PC2
PC1
10
5
0
-5
-10
-10
0
10

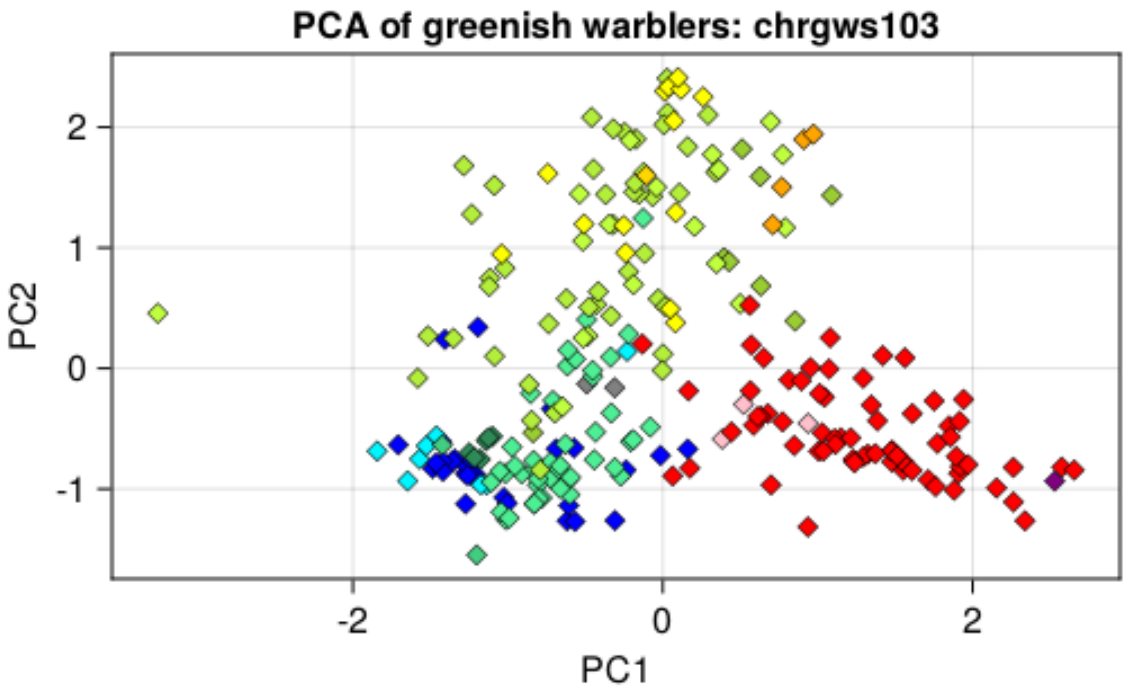

PCA of greenish warblers: chrgw21
PC2
5
0
-5
-10
-5
0
5
10
PC1

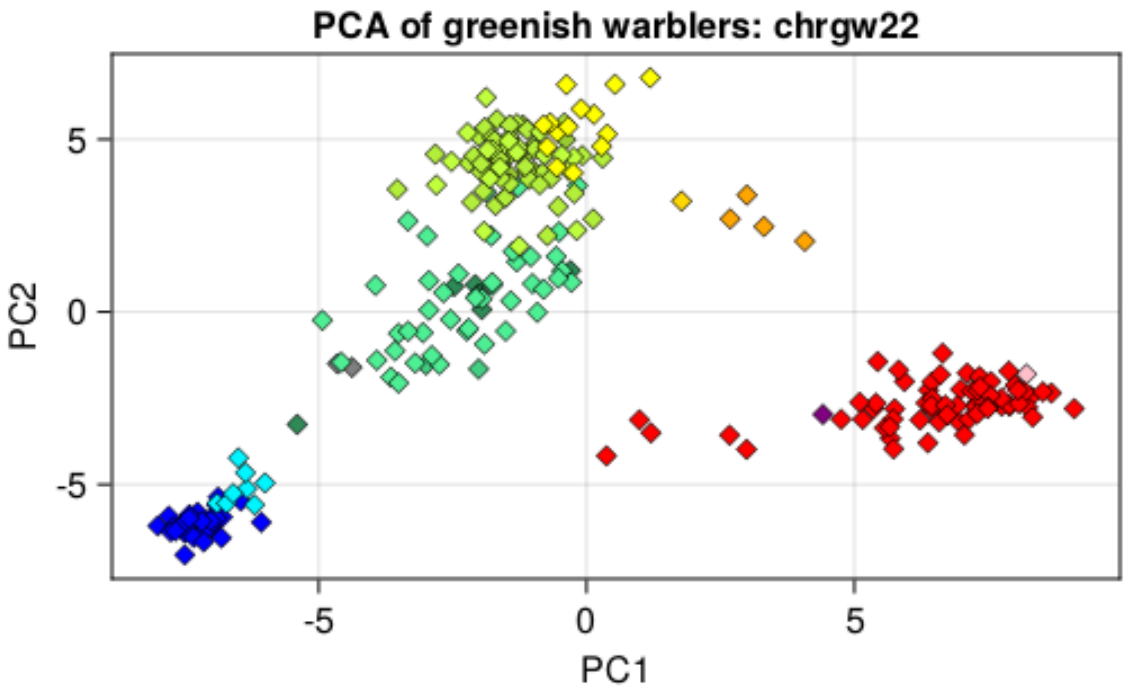

PCA of greenish warblers: chrgw26
PC2
PC1
10
5
0
-5
-10
-10
0
10

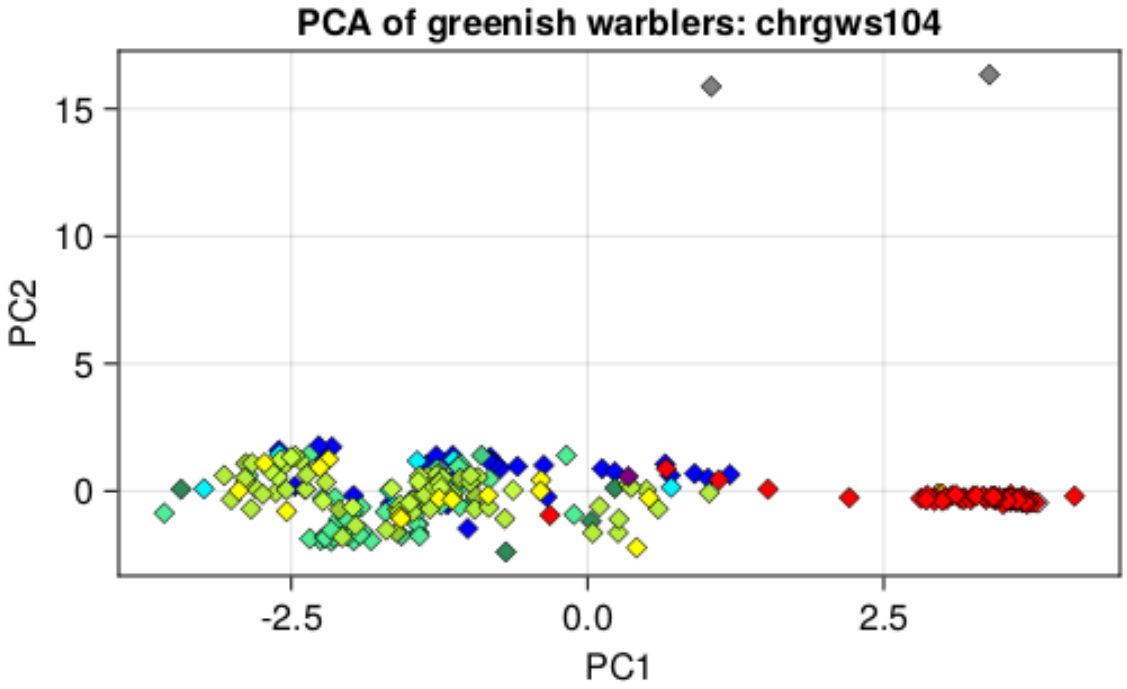

PCA of greenish warblers: chrgws102
PC2
PC1
0
-2
-4
-2
0
2

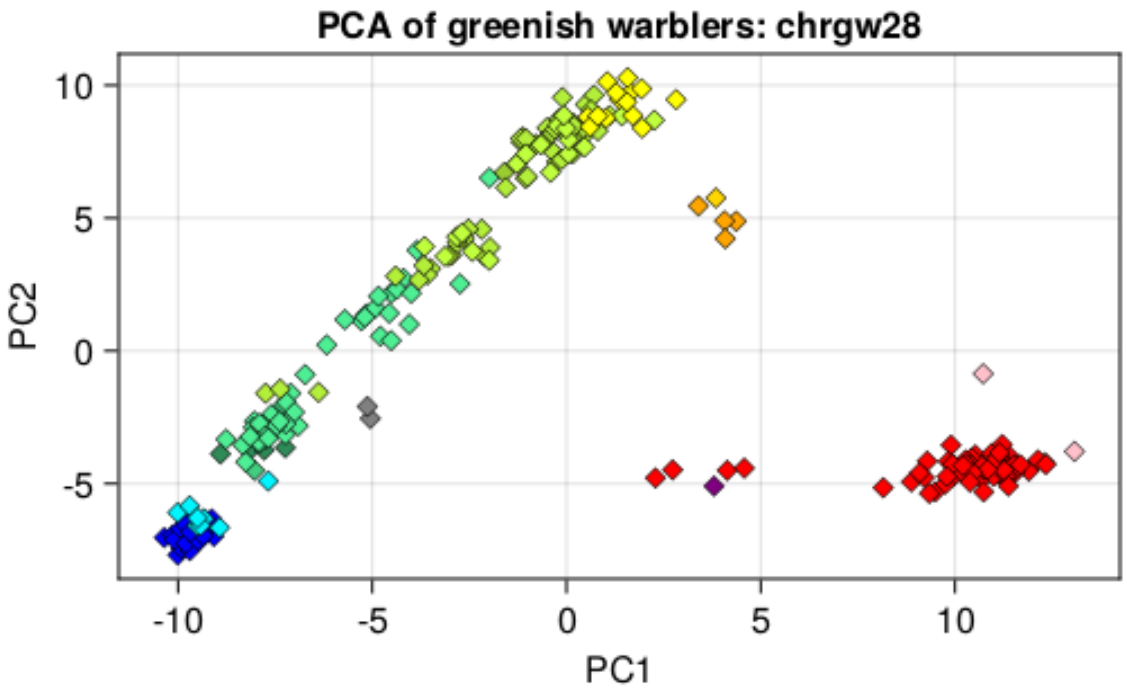

PCA of greenish warblers: chrgw23
PC2
PC1
10
5
0
-5
-10
-5
0
5
10

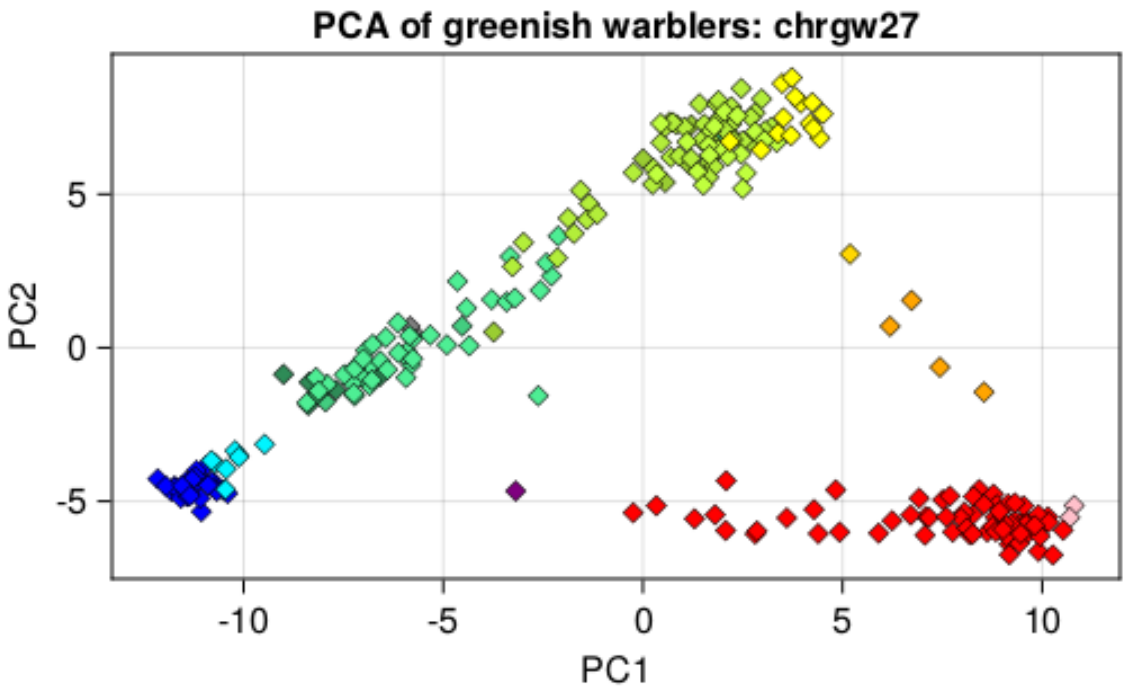

PCA of greenish warblers: chrgw25
PC2
PC1
5
0
-5
-10
-5
0
5

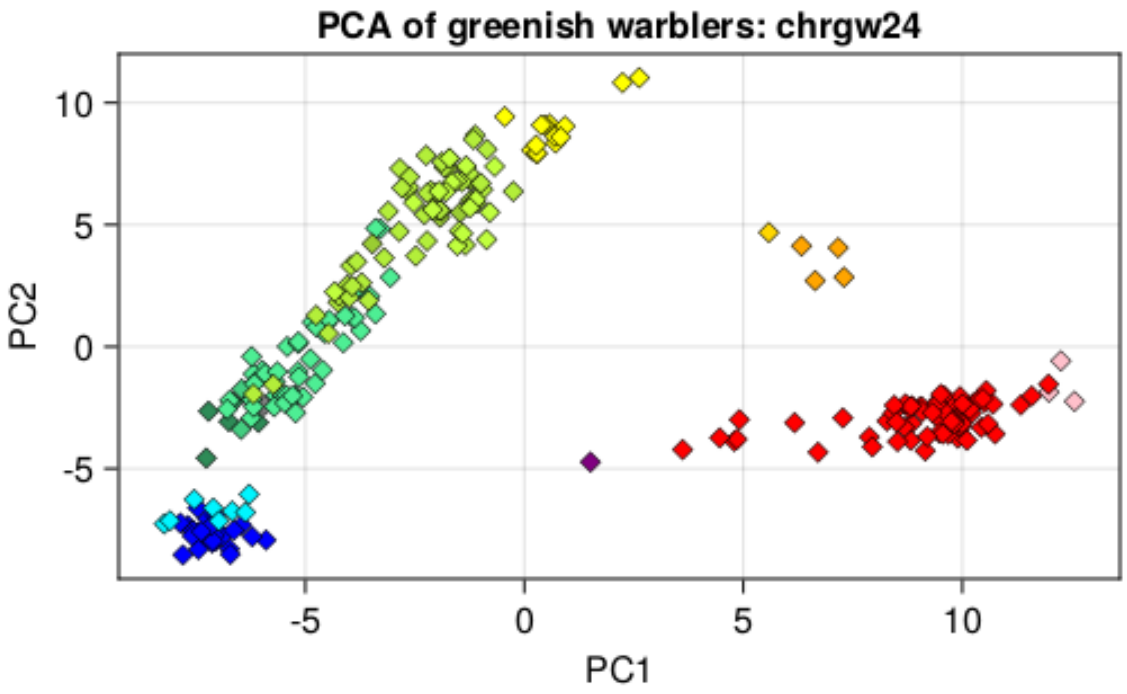

PCA of greenish warblers: chrgws103
PC2
PC1
2
1
0
-1
-2
0
2

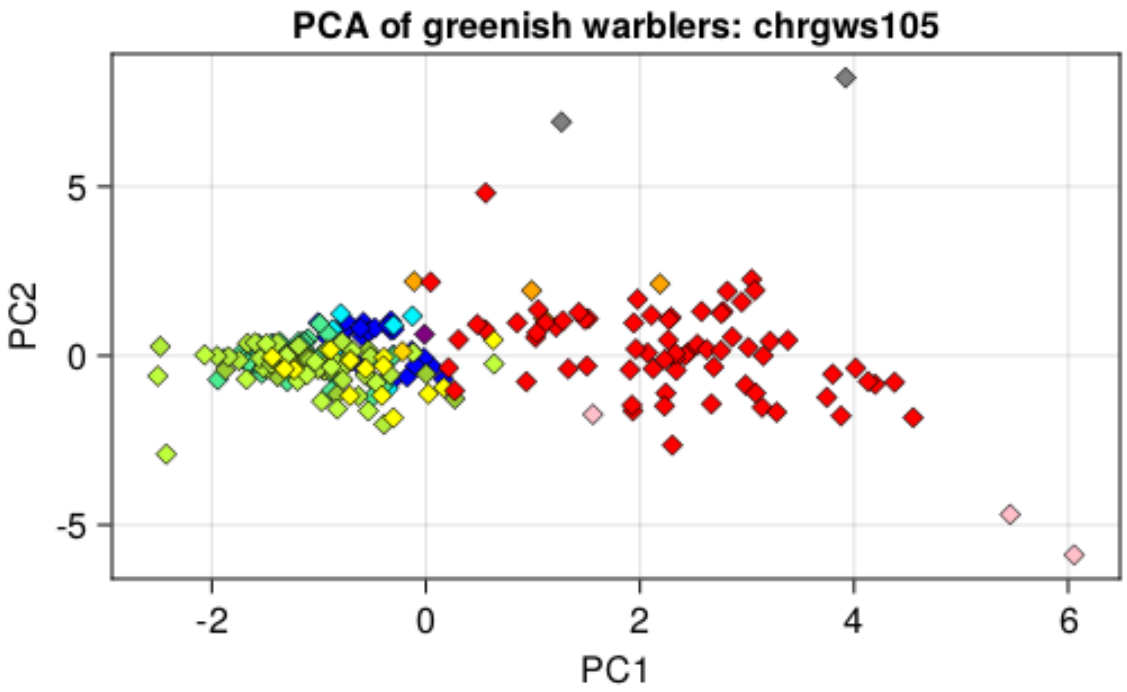

PCA of greenish warblers: chrgw22
PC2
PC1
5
0
-5
-5
0
5

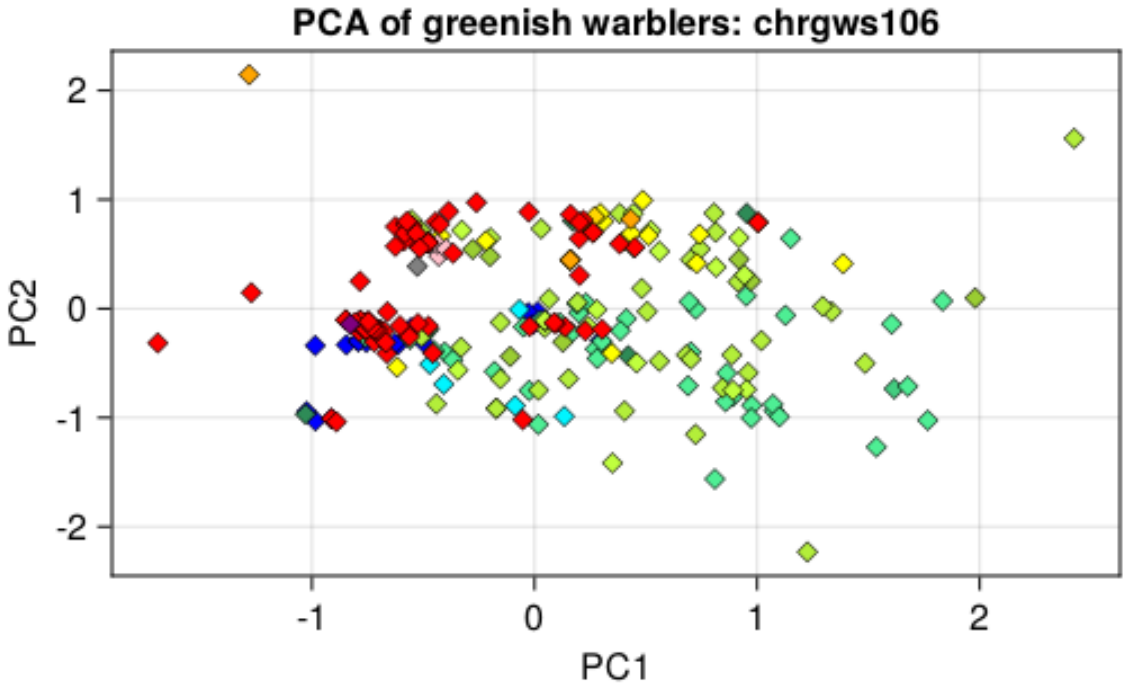

PCA of greenish warblers: chrgws104
15
10
5
0
PC2
-2.5
0.0
2.5
PC1

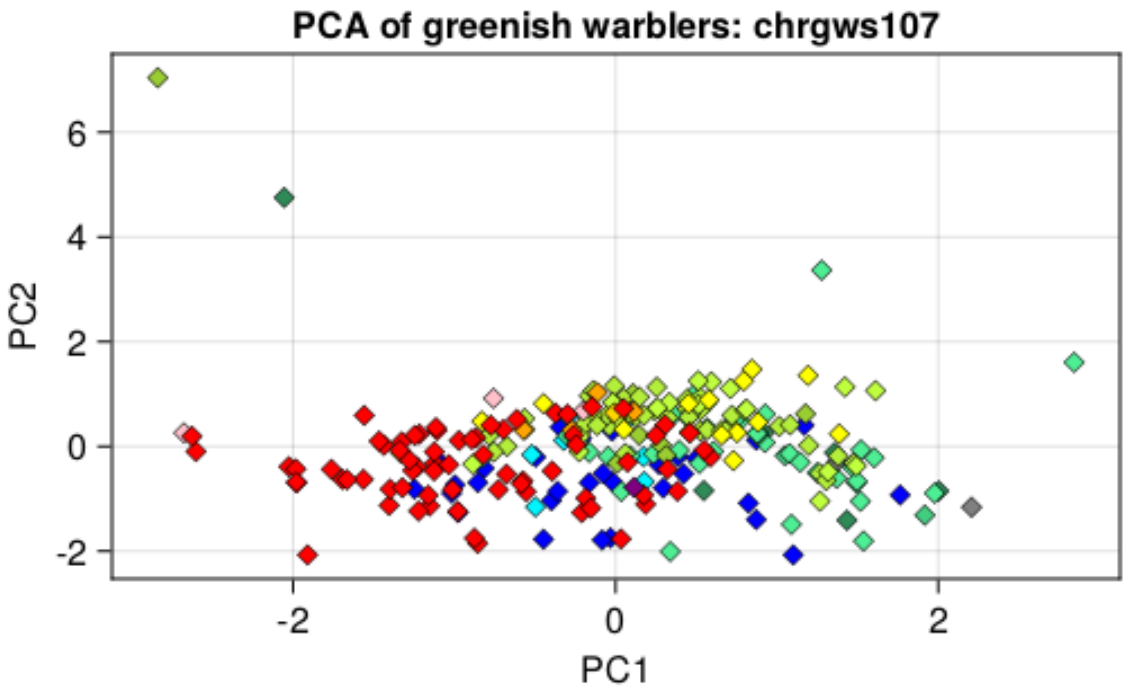

PCA of greenish warblers: chrgw28
PC2
PC1
10
5
0
-5
-10
-5
0
5
10

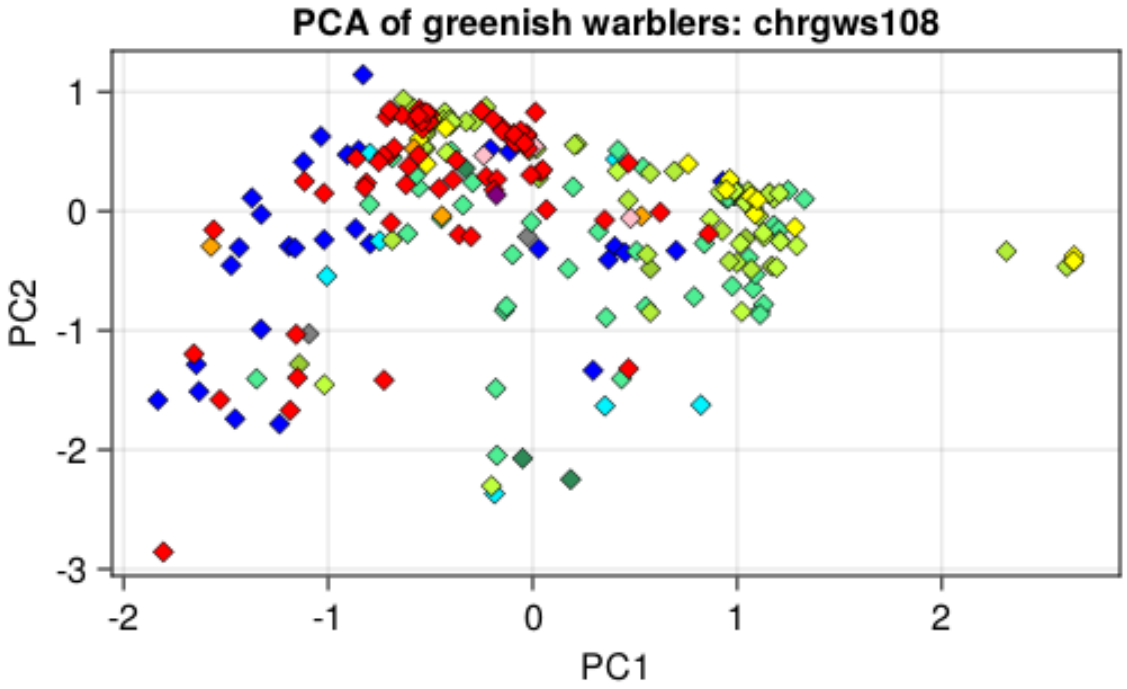

PCA of greenish warblers: chrgw27
PC2
5
0
-5
-10
-5
0
5
10
PC1

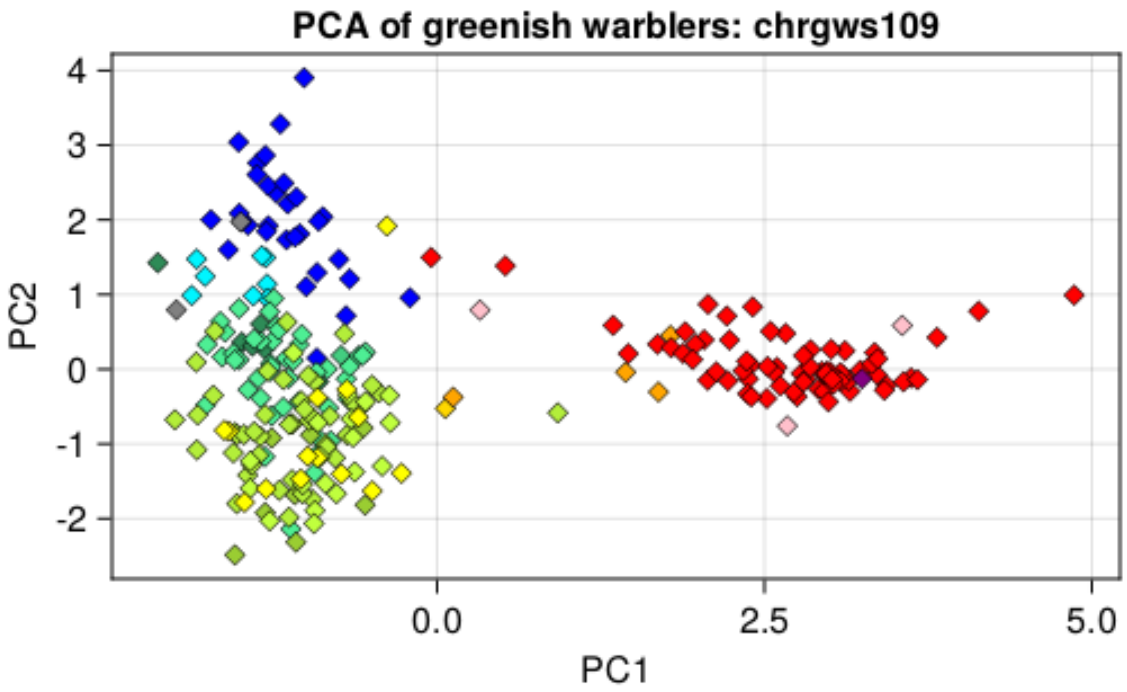

PCA of greenish warblers: chrgw24
PC2
PC1
10
5
0
-5
-5
0
5
10

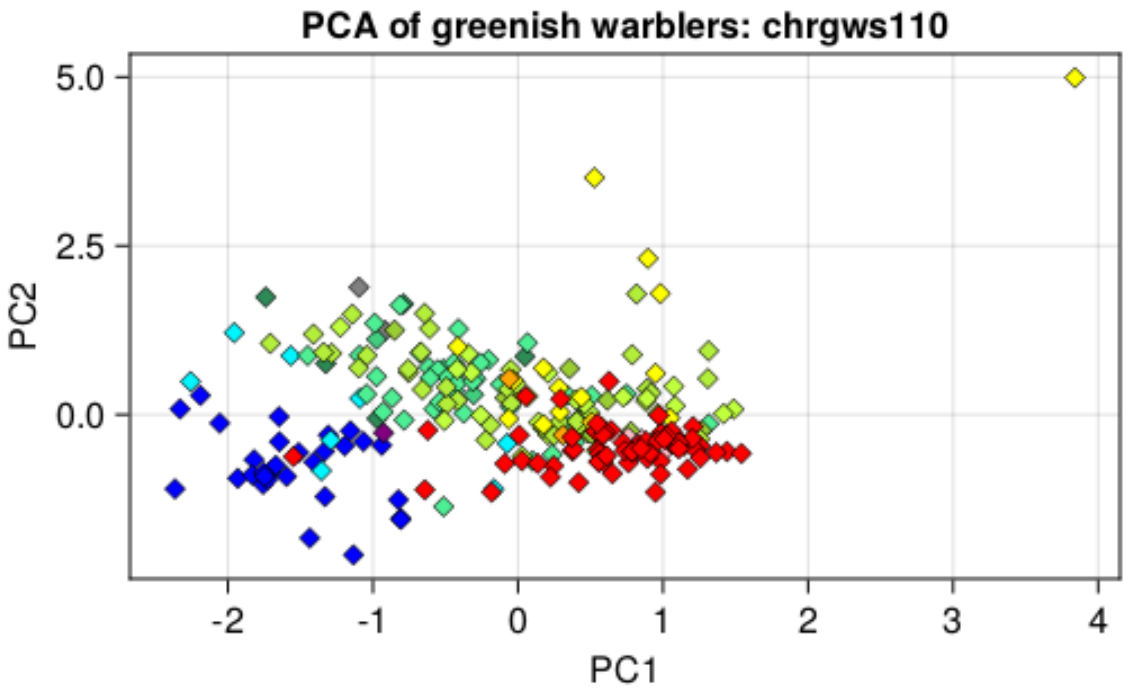

PCA of greenish warblers: chrgws105
PC2
PC1
5
0
-5
-2
0
2
4
6

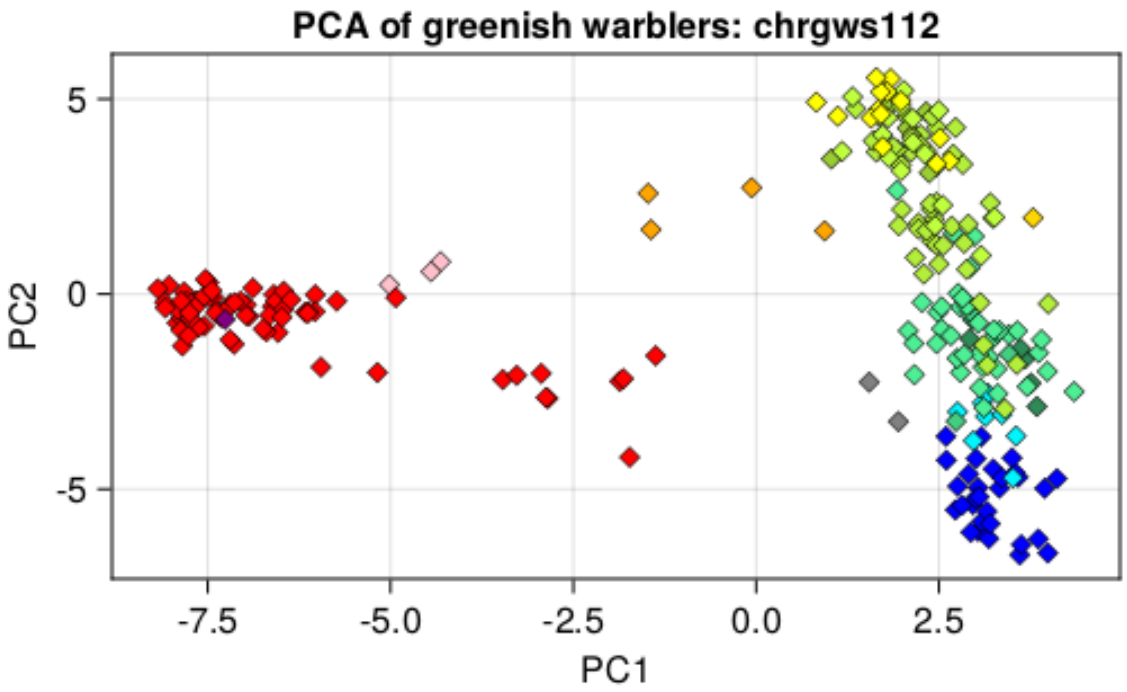

PCA of greenish warblers: chrgws106
PC2
PC1
2
1
0
-1
-2
-1
0
1
2


PCA of greenish warblers: chrgws107
PC2
PC1
6
4
2
0
-2
-2
0
2


PCA of greenish warblers: chrgws108
PC2
PC1
1
0
-1
-2
-3
-2
-1
0
1
2


PCA of greenish warblers: chrgws109
PC2
PC1
4
3
2
1
0
-1
-2
0.0
2.5
5.0


PCA of greenish warblers: chrgws110
5.0
2.5
0.0
PC2
-2
-1
0
1
2
3
4
PC1



Whole-genome PCA

In addition to making PCA plots for each scaffold, we can do one for the whole genome. The imputing for the whole genome takes some time (almost 2 hours!), so I did this in advance and saved a file. This is incorporated into the code below–to actually do the imputing, set `do_imputing = true`. Otherwise this code will load the previously-imputed data.

(NOTE: this is using an older version of the data–sometime need to update by removing `tempTagName` and replacing with `tagName`)

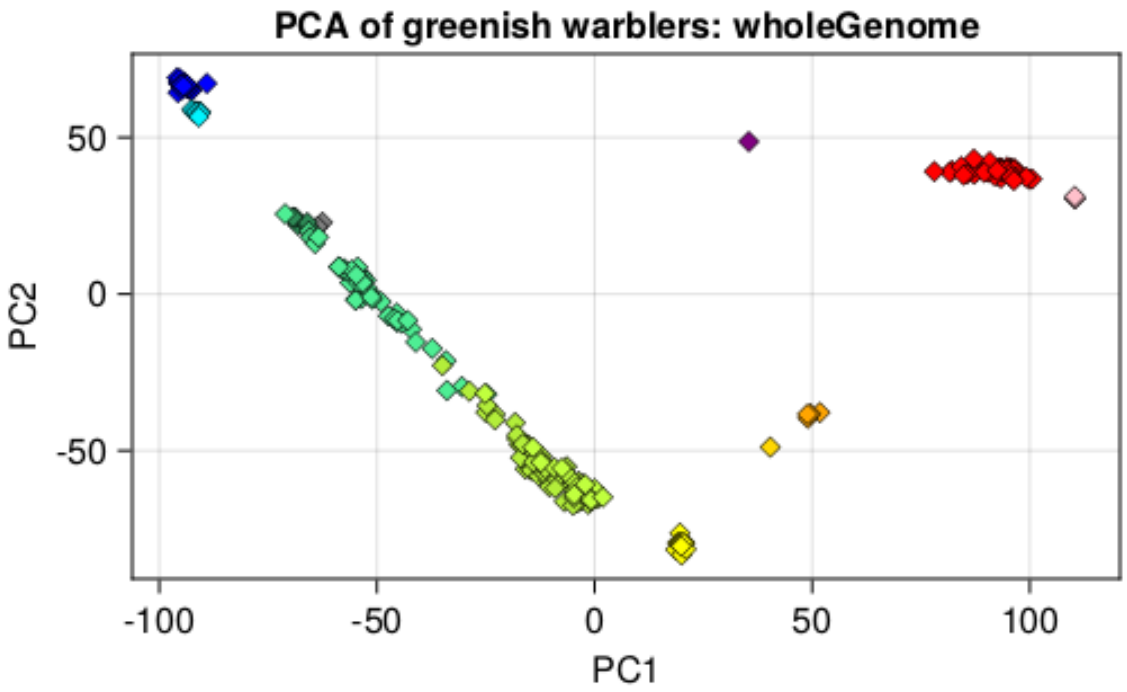
```
genosOnly_for_imputing = Matrix{Union{Missing, Float32}}(genosOnly)
regionText = "wholeGenome"
tempTagName = ".June2023."
filename = string(baseName, tempTagName, regionText, ".imputedMissing.jld2")
# to do the imputing, do this by setting to true, but TAKES A LONG TIME:
do_imputing = false
if do_imputing
    @time imputed_genosOnly = Impute.svd(genosOnly_for_imputing)
    # took almost 2 hours!
    jldsave(filename; imputed_genosOnly, ind_with_metadata_indFiltered, pos_SNP_filtered)
```

```
    imputed_genosOnly_wholeGenome = imputed_genosOnly
    ind_with_metadata_indFiltered_wholeGenome = ind_with_metadata_indFiltered
    pos_SNP_filtered_wholeGenome = pos_SNP_filtered
    print("Saved matrix of real and imputed genotypes for filtered individuals. \n")
else # load the already saved imputing
    imputed_genosOnly_wholeGenome = load(filename, "imputed_genosOnly")
    ind_with_metadata_indFiltered_wholeGenome = load(filename, "ind_with_metadata_indFiltered")
    pos_SNP_filtered_wholeGenome = load(filename, "pos_SNP_filtered")
    println(string("Loaded ",filename))
    println(string(regionText, ": ", size(imputed_genosOnly_wholeGenome, 2), " SNPs from ", size(impu
end
```

```
Loaded GW_genomics_2022_with_new_genome/GW2022_GBS_012NA_files/GW2022_all4plates.genotypes.SNPs_only.who
wholeGenome: 1003400 SNPs from 271 individuals
```

Now make the whole-genome PCA:

```
plotPCA(imputed_genosOnly_wholeGenome,
        ind_with_metadata_indFiltered_wholeGenome,
        groups_to_plot_PCA, group_colors_PCA;
        sampleSet = "greenish warblers", regionText=regionText,
        flip1 = true, flip2 = true)
```




```
CairoMakie.Screen{IMAGE}
```

Genotype-by-individual plots

Now, show individual genotypes for subsets of the dataset. Can choose individuals and genomic regions to plot, along with an Fst cutoff (only show SNPs with greater Fst than the cutoff).

```
set = "67_inds_around_ring"  #"east_side_of_ring"    #"67_inds_around_ring"  # "west_side_of_ring"

if set == "67_inds_around_ring"
    groups = ["vir","troch_LN","plumb"] # for purpose of calculating pairwise Fst and Fst_group (to c
    plotGroups = ["vir","vir_S","lud_PK","lud_KS","lud_central","troch_LN","troch_EM","obs", "plumb_L
    plotGroupColors = ["blue","turquoise1","seagreen4","seagreen3","seagreen2","yellow","gold","orang
    numIndsToPlot = [10, 5, 6, 2, 7, 15, 15, 15, 15, 15] # maximum number of individuals to plot fro
    group1 = "vir"   # these groups will determine the color used in the graph
    group2 = "plumb"
    groupsToCompare = "vir_plumb"   #"Fst_among"  #"vir_troch_LN"       #"vir_plumb"      #"troch_LN
    Fst_cutoff = 0.7
    missingFractionAllowed = 0.2  # only show SNPs with less than this fraction of missing data among
```

```
elseif set == "37_inds_around_ring_plusAllVirPlumb"
    groups = ["vir","troch_LN","plumb"] # for purpose of calculating pairwise Fst and Fst_group (to c
    plotGroups = ["vir","lud","troch_LN","troch_EM","obs", "obs_plumb","plumb"]
    plotGroupColors = ["blue","seagreen4","yellow","gold","orange", "pink","red"]
    numIndsToPlot = [100, 15, 15, 15, 15, 15, 100] # maximum number of individuals to plot from each
    group1 = "vir"   # these groups will determine the color used in the graph
    group2 = "plumb"
    groupsToCompare = "Fst_among"
    Fst_cutoff = 0.7
    missingFractionAllowed = 0.2  # only show SNPs with less than this fraction of missing data among
elseif set == "west_side_of_ring"
    groups = ["vir","troch_LN"] # for purpose of calculating pairwise Fst and Fst_group (to determine
    plotGroups = ["vir","vir_misID","vir_S","nit", "lud_PK", "lud_KS", "lud_central", "lud_Sath", "lu
    plotGroupColors = ["blue","blue","turquoise1","grey","seagreen4","seagreen3","seagreen2","olivedr
    numIndsToPlot = [15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15] # maximum number of individuals to p
    group1 = "vir"   # these groups will determine the color used in the graph
    group2 = "troch_LN"
    groupsToCompare = "vir_troch_LN" # "Fst_among"
    Fst_cutoff = 0.6
    missingFractionAllowed = 0.2  # only show SNPs with less than this fraction of missing data among
elseif set == "all_ludlowi_plus_a_few_other"
    groups = ["vir","troch_LN","plumb"] # for purpose of calculating pairwise Fst and Fst_group (to c
    plotGroups = ["vir","vir_S","nit", "lud_PK", "lud_KS", "lud_central", "lud_Sath", "lud_ML","troch
    plotGroupColors = ["blue","turquoise1","grey","seagreen4","seagreen3","seagreen2","olivedrab3","d
    numIndsToPlot = [4, 4, 4, 1000, 1000, 1000, 1000, 1000, 1000, 4, 4] # maximum number of individua
    group1 = "vir"   # these groups will determine the color used in the graph
    group2 = "troch_LN"
    groupsToCompare = "vir_troch_LN" # "Fst_among"
    Fst_cutoff = 0.6
    missingFractionAllowed = 0.2  # only show SNPs with less than this fraction of missing data among
elseif set == "east_side_of_ring"
    groups = ["troch_LN","obs","plumb"] # for purpose of calculating pairwise Fst and Fst_group (to c
    plotGroups = ["troch_LN","troch_EM","obs","obs_plumb","plumb"]
    plotGroupColors = ["yellow","gold","orange","pink","red"]
    numIndsToPlot = [15, 15, 15, 15, 15] # maximum number of individuals to plot from each group
    group1 = "troch_LN"   # these groups will determine the color used in the graph
    group2 = "plumb"
    groupsToCompare = "troch_LN_plumb"
    Fst_cutoff = 0.7
    missingFractionAllowed = 0.2  # only show SNPs with less than this fraction of missing data among
elseif set == "vir_plumb"
```

```
    groups = ["vir","plumb"]
    plotGroups = ["vir","plumb_vir","plumb"]
    plotGroupColors = ["blue","purple","red"]
    numIndsToPlot = [100,100,100] # maximum number of individuals to plot from each group
    group1 = "vir"   # these groups will determine the color used in the graph
    group2 = "plumb"
    groupsToCompare = "vir_plumb"
    Fst_cutoff =  0.7
    missingFractionAllowed = 0.2  # only show SNPs with less than this fraction of missing data amon
end
```

```
0.2
```

Calculate allele freqs and sample sizes (use column Fst_group)

```
freqs, sampleSizes = getFreqsAndSampleSizes(genosOnly, ind_with_metadata_indFiltered.Fst_group, grou
println("Calculated population allele frequencies and sample sizes")
```

```
Calculated population allele frequencies and sample sizes
```

calculate Fst

```
Fst, FstNumerator, FstDenominator, pairwiseNamesFst = getFst(freqs, sampleSizes, groups; among=true)
println("Calculated Fst values")
```

```
Calculated Fst values
```

limit the individuals to include in plot

```
genosOnly_included, ind_with_metadata_included = limitIndsToPlot(plotGroups, numIndsToPlot, genosOnl
```

choose the scaffold and region to show

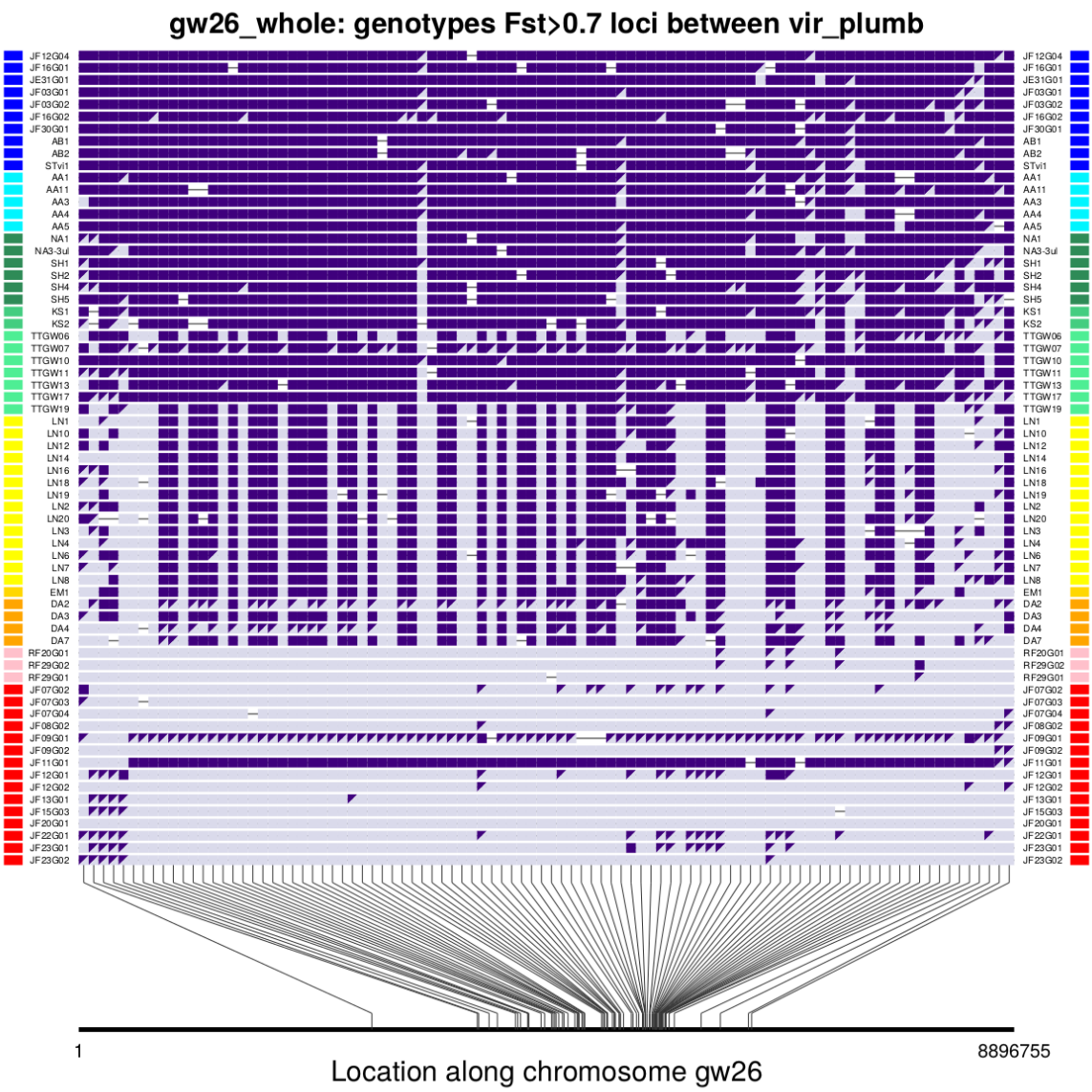
```
chr = "gw26"
regionInfo = chooseChrRegion(pos_SNP_filtered, chr; positionMin=1, positionMax=NaN) # this gets the r
```

```
("gw26", 1, 8896755, "gw26_whole")
```

NOTE FOR LATER: SHOULD REALLY GET CHROMOSOME LENGTH FOR position-Max

Now actually make the plot

```
plotInfo = plotGenotypeByIndividual(groupsToCompare, Fst_cutoff, missingFractionAllowed,
    regionInfo, pos_SNP_filtered, Fst, pairwiseNamesFst,
    genosOnly_included, ind_with_metadata_included, freqs, plotGroups, plotGroupColors);
# plotInfo contains a tuple with: (f, plottedGenotype, locations, plottedMetadata)
```


gw26_whole: genotypes Fst>0.7 loci between vir_plumb
JF12G04
JF16G01
JE31G01
JF03G01
JF03G02
JF16G02
JF30G01
AB1
AB2
STvi1
AA1
AA11
AA3
AA4
AA5
NA1
NA3-3ul
SH1
SH2
SH4
SH5
KS1
KS2
TTGW06
TTGW07
TTGW10
TTGW11
TTGW13
TTGW17
TTGW19
LN1
LN10
LN12
LN14
LN16
LN18
LN19
LN2
LN20
LN3
LN4
LN6
LN7
LN8
EM1
DA2
DA3
DA4
DA7
RF20G01
RF29G02
RF29G01
JF07G02
JF07G03
JF07G04
JF08G02
JF09G01
JF09G02
JF11G01
JF12G01
JF12G02
JF13G01
JF15G03
JF20G01
JF22G01
JF23G01
JF23G02
1
8896755
Location along chromosome gw26