NOAA FISHERIES

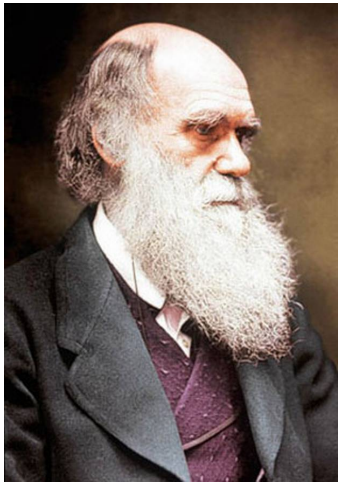
Population Genetics in R

Stratagies for *strataG*

Eric Archer, Ph.D.
Marine Mammal Genetics Group
Southwest Fisheries Science Center
858-945-3553
eric.archer@noaa.gov
https://www.github.com/ericarcher

How users see:

Themselves



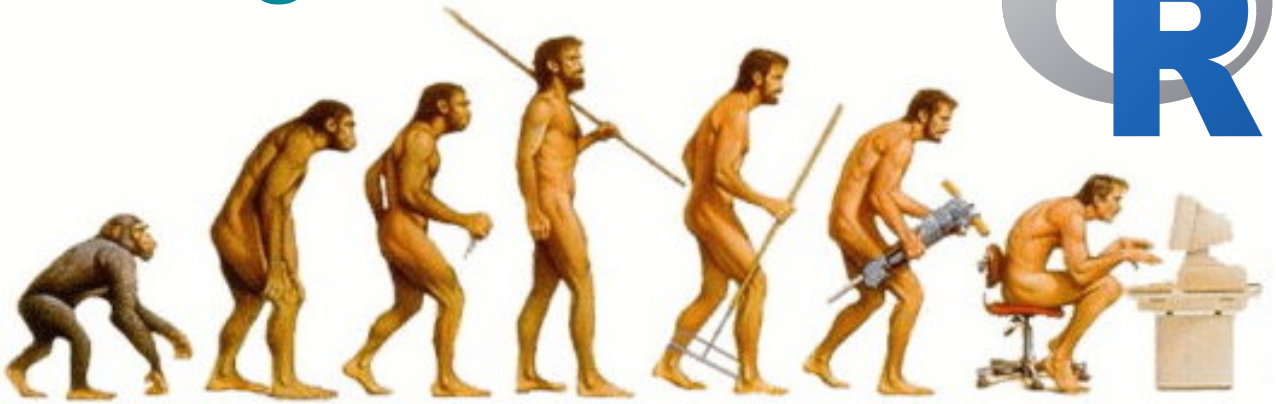
Developers

How developers see:

Themselves

Users

Challenges



- Users and developers with different degrees of experience in R AND genetics
- Making packages and functions easy to find relative to analytical questions
- Flexible data formats that are easy to create for users and easy for developers to manipulate
- Fast and efficient execution
- Ensuring correct and validated results during development and afterwards
- Documentation: standard workflows, function capabilities, output

2015 PopGen in R Hackathon

- 29 Participants
- Interoperability, scalability, and reducing workflow building challenges
- 5 Teams
  - PopGen in R Resource Vignettes
  - VCF Reader (`vcfR`)
  - PopGen Simulator GUI (`skeleSim`)
  - Multivariate Outlier Identification (`MINOTAUR`)
  - Effective Population Size Calculation
- Molecular Ecology Resources Special Issue

Why R?

- Open source
- Portable
- Interactive code
- Easy to develop code
- Many analytical functions
- Good visualization options
- Reproducibility
- Documentable
- Large support community

Why not R?

- Learning curve
- Too many options
- Slow code
- Memory inefficient code

strataG

- Population genetics toolkit
- Haploid / diploid data
- Genetic summaries
- QA/QC analyses / reports
- Population structure tests
- Multiple stratification schemes
- Wrapper for external programs

Input

- What information?
  - Genetic data: genotypes/haplotypes, sequences
  - Sample IDs
  - Stratifcation (alternate schemes, hierarchy)
  - Auxillary data (sex, location, etc.)
  - Labels

- What format?
  - Individual objects (vectors, matrices, data.frames) – bad
  - Everything in a list – better
  - Custom class (S4) – best
    - Type safe, Custom methods, Validation
    - What format? (memory usage, processing speed, ease of access)

gtypes

S4 class with slots:

- `loci` data.frame of genotypes/haplotypes
- `ploidy` ploidy of genetic data
- `strata` factor of current stratification
- `schemes` data.frame of alternate stratification schemes
- `sequences` list of DNA sequences (multidna)
- `description` label for object
- `other` auxillary information (e.g., data.frame)

Can be populated from:

- matrix/data.frame of genotypes/haplotypes
- sequences (read from FASTA, list of characters, DNAbin (ape), multidna (apex))
- data stored in other formats: genind (adegenet), loci (pegas)

```
> data(msats.g)
> msats.g
<<< dolphin msats >>>

Contents: 126 samples, 5 loci, 3 strata

Strata summary:
               num.samples num.missing num.alleles prop.unique.alleles
Coastal                 68         1.2         4.8               0.0857
Offshore.North          40         0.8        12.6               0.2240
Offshore.South          18         0.0        11.0               0.2510
               heterozygosity
Coastal                 0.631
Offshore.North          0.790
Offshore.South          0.867

Locus summary:
      num.genotyped num.alleles prop.unique.alleles obsvd.heterozygosity
D11t           125          12               0.2500                 0.704
EV37           119          22               0.1364                 0.697
EV94           125          15               0.0667                 0.776
Ttr11          125           9               0.2222                 0.704
Ttr34          126          10               0.2000                 0.698
```

```
> data(dloop.g)
> dloop.g
<<< dolphin dLoop >>>

Contents: 126 samples, 1 locus, 3 strata

Strata summary:
               num.samples num.missing num.alleles prop.unique.alleles
Coastal                 68           0           5                0.000
Offshore.North          40           0          22                0.545
Offshore.South          18           0          14                0.786
               heterozygosity
Coastal                 0.743
Offshore.North          0.958
Offshore.South          0.967

Sequence summary:
      num.seqs min.length mean.length max.length     a     c     g    t
dLoop       33        402         402        402 0.301 0.229 0.129 0.34
```

Manipulation

- Accessor functions
  - `nInd, nLoc, nStrata`
  - `indNames, locNames, strataNames`
  - `loci, strata, sequences`
- Indexing
  - `g[id, loci, strata]`
  - numeric, character, logical
- Modifying
  - `strata(g) <-`
  - `schemes(g) <-`
  - `stratify(g, “scheme”)`

```
> nInd(msats.g)
[1] 126


> strataNames(dloop.g)
[1] "Coastal"        "Offshore.North" "Offshore.South”


> st <- strata(dloop.g)
> str(st)
 Factor w/ 3 levels "Coastal","Offshore.North",..: 2 2 2 2 2 2 2 2 2 2 ...
 - attr(*, "names")= chr [1:126] "4495" "4496" "4498" "5814" ...


> head(schemes(dloop.g))
        broad           fine
4495 Offshore Offshore.North
4496 Offshore Offshore.North
4498 Offshore Offshore.North
5814 Offshore Offshore.North
5815 Offshore Offshore.North
5816 Offshore Offshore.North
```

```
> stratify(dloop.g, "broad")
<<< dolphin dLoop >>>

Contents: 126 samples, 1 locus, 2 strata

Strata summary:
         num.samples num.missing num.alleles prop.unique.alleles
Coastal           68           0           5                0.000
Offshore          58           0          29                0.483
         heterozygosity
Coastal           0.743
Offshore          0.959

Sequence summary:
      num.seqs min.length mean.length max.length     a     c     g    t
dLoop       33        402         402        402 0.301 0.229 0.129 0.34
```

```
> msats.g[1:5, , ]
<<< dolphin msats >>>

Contents: 5 samples, 5 loci, 1 stratum

Strata summary:
                num.samples num.missing num.alleles prop.unique.alleles
Offshore.North            5         0.8         4.6                 0.653
                heterozygosity
Offshore.North           0.527

Locus summary:
      num.genotyped num.alleles prop.unique.alleles obsvd.heterozygosity
D11t              5           5                0.600                 0.400
EV37              3           4                1.000                 0.333
EV94              4           6                0.667                 1.000
Ttr11             4           4                0.500                 0.500
Ttr34             5           4                0.500                 0.400
```

```
> msats.g[, , "Coastal"]
<<< dolphin msats >>>

Contents: 68 samples, 5 loci, 1 stratum

Strata summary:
        num.samples num.missing num.alleles prop.unique.alleles
Coastal          68         1.2         4.8              0.0857
        heterozygosity
Coastal          0.631

Locus summary:
      num.genotyped num.alleles prop.unique.alleles obsvd.heterozygosity
D11t             67           3                0.000                0.522
EV37             63           7                0.429                0.619
EV94             68           5                0.000                0.735
Ttr11            68           4                0.000                0.632
Ttr34            68           5                0.000                0.647
```

Modularity

- Simple core functions
- Consistent output
  - By-loci metrics output as vectors
  - Pairwise metrics output as data.frames
- Convenience functions
  - By-locus summaries
  - QA/QC wrapper

```
> num.al <- numAlleles(msats.g)
> hwe <- hweTest(msats.g)

> num.al
 D11t  EV37  EV94 Ttr11 Ttr34
   12    22    15     9    10

> hwe
 D11t  EV37  EV94 Ttr11 Ttr34
0.001 0.000 0.051 0.000 0.000

> cbind(num.alleles = num.al, hwe.p.val = hwe)
      num.alleles hwe.p.val
D11t           12     0.001
EV37           22     0.000
EV94           15     0.051
Ttr11           9     0.000
Ttr34          10     0.000
```

```
> summarizeLoci(msats.g)
      num.genotyped prop.genotyped num.alleles allelic.richness
D11t           125          0.992          12           0.0960
EV37           119          0.944          22           0.1849
EV94           125          0.992          15           0.1200
Ttr11          125          0.992           9           0.0720
Ttr34          126          1.000          10           0.0794
      prop.unique.alleles exptd.heterozygosity obsvd.heterozygosity
D11t               0.2500                0.747                0.704
EV37               0.1364                0.827                0.697
EV94               0.0667                0.830                0.776
Ttr11              0.2222                0.795                0.704
Ttr34              0.2000                0.814                0.698
```

Performance

- Efficient code vs. easy code
- Profiling
- Multi-threading
- Compiled code (RCpp)

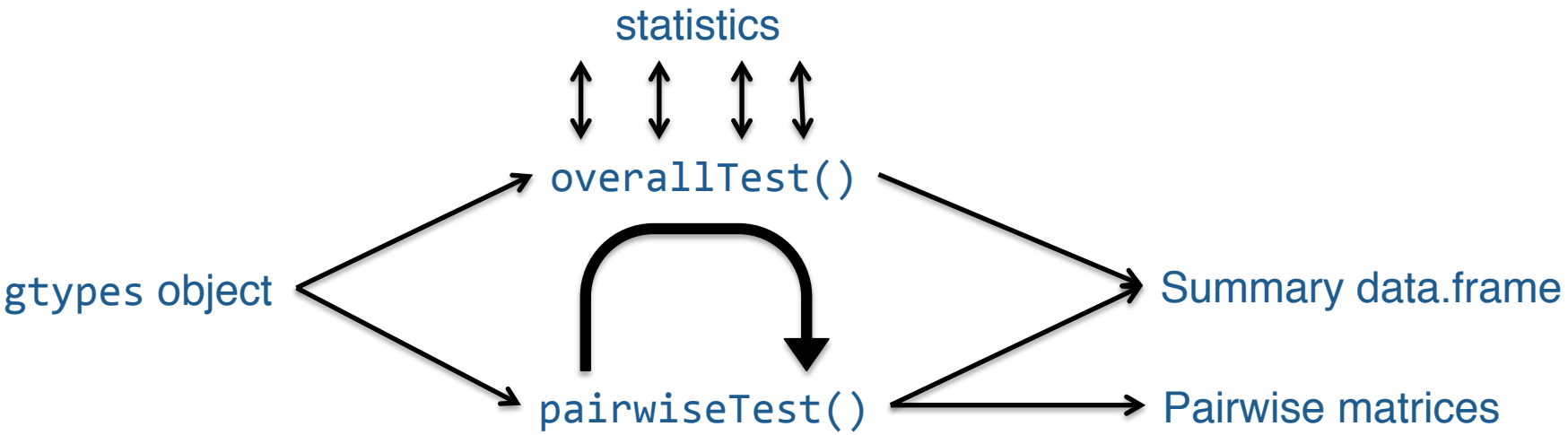
```
> Rprof()
> prop.shared <- propSharedLoci(msats.g, type = "ids", num.cores = 1)
> Rprof(NULL)
> smry <- summaryRprof()
> head(prop.shared, 5)
  ids.1 ids.2 num.same num.not.missing prop.same D11t EV37 EV94 Ttr11
1  4495  4496        0               2     0.000    0   NA   NA    NA
2  4495  4498        1               3     0.333    1   NA   NA  0.75
3  4495  5814        0               3     0.000    0   NA   NA  0.00
4  4495  5815        0               3     0.000    0   NA   NA  0.00
5  4495  5816        0               3     0.000    0   NA   NA  0.00

> head(smry$by.self, 4)
                 self.time self.pct total.time total.pct
"sub"                10.78    31.52      11.16     32.63
"match"               1.94     5.67       9.94     29.06
"standardGeneric"     1.76     5.15      29.80     87.13
"nchar"               1.36     3.98       1.36      3.98
```

```
> coastal <- msats.g[, , "Coastal"]
> library(microbenchmark)
> microbenchmark(
+   cores.1 = propSharedLoci(coastal, type = "ids", num.cores = 1),
+   cores.2 = propSharedLoci(coastal, type = "ids", num.cores = 2),
+   cores.3 = propSharedLoci(coastal, type = "ids", num.cores = 3),
+   cores.4 = propSharedLoci(coastal, type = "ids", num.cores = 4),
+   times = 10
+ )
Unit: seconds
    expr  min   lq mean median   uq  max neval  cld
 cores.1 8.79 9.29 9.38   9.46 9.50 9.70    10    d
 cores.2 4.60 4.83 4.84   4.86 4.90 4.92    10   c
 cores.3 3.17 3.31 3.42   3.46 3.52 3.60    10  b
 cores.4 2.59 2.84 2.89   2.92 3.00 3.10    10 a
```

Population subdivision tests

- Multiple metrics
  - Haploid: F_{st}, Φ_{st}, χ^2
  - Diploid: F_{st}, F'_{st}, G_{st}, G'_{st}, G''_{st}, Jost's D
  - Coded in C
- Test can be 'global' (all strata) or pairwise
- Permutation based: running same test multiple times
- Multiple formats for results



```
> statFst(msats.g, nrep = 1000)
$stat.name
[1] "Fst"

$result
estimate    p.val
0.111807 0.000999

$null.dist
NULL

> statChi2(msats.g, nrep = 30, keep.null = TRUE)
$stat.name
[1] "Chi2"

$result
estimate    p.val
664.7193   0.0323

$null.dist
 [1] 161.0 104.6 150.9 143.4 132.3 176.2 162.8 163.4 140.3  94.3 164.6
[12] 147.2 113.4 147.0 122.2 126.0 163.9 111.4 147.2 141.0 122.3 136.9
[23] 162.1 168.6 155.7 170.0 161.1 124.4 106.8 129.0
```

```
> overallTest(msats.g, nrep = 1000)

<<< dolphin msats >>>
 2016-04-24 00:21:32 : Overall test : 1000 permutations

                 N
Coastal         68
Offshore.North  40
Offshore.South  18

Population structure results:
          estimate       p.val
Chi2  664.71927576 0.000999001
D       0.28595938 0.001034126
Fst     0.11180737 0.000999001
F'st    0.48368156 0.000999001
Fis     0.03987941 1.000000000
Gst     0.05393608 0.000999001
G'st    0.07877958 0.000999001
G''st   0.38311599 0.000999001
PHIst           NA          NA
```

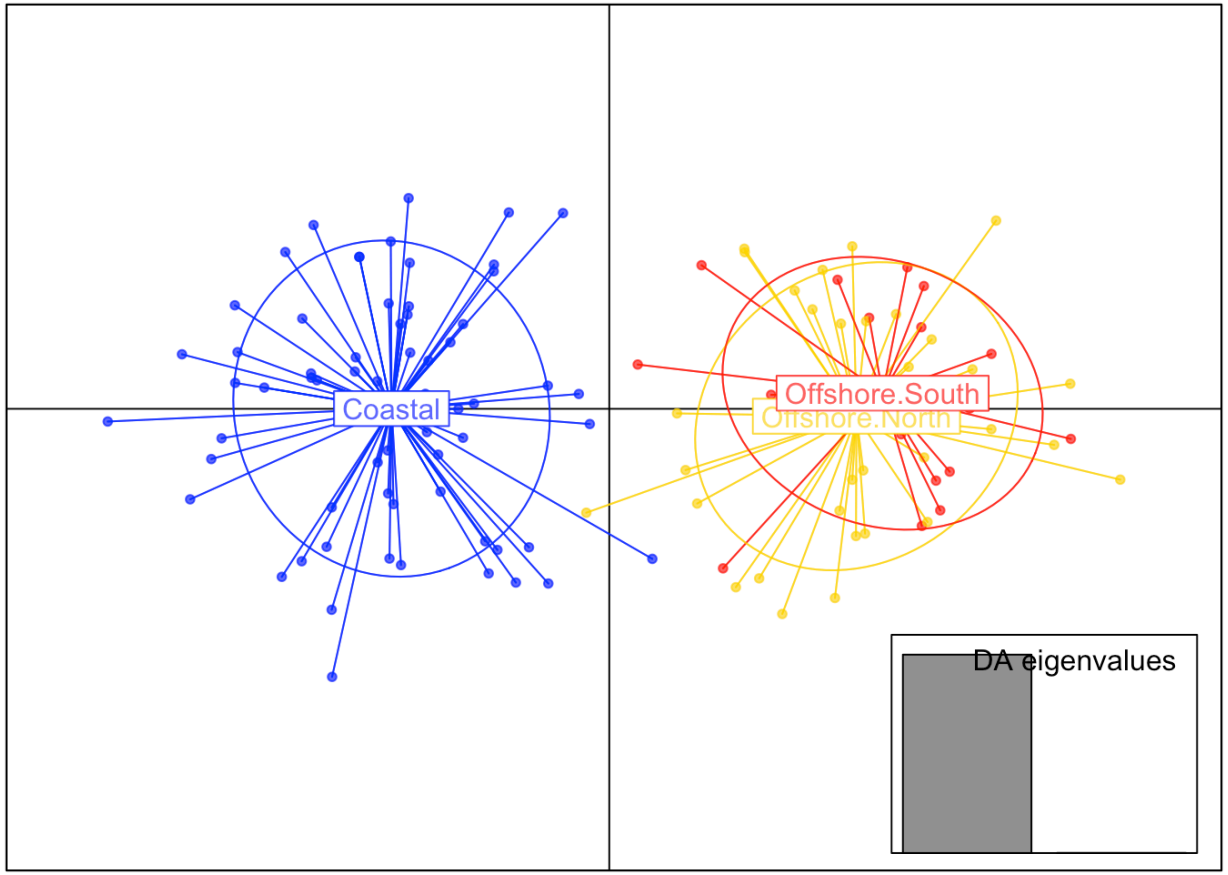
```
> system.time(pairwiseTest(msats.g))
<<< dolphin msats >>>
 2016-04-24 00:06:38 : Pairwise tests : 1000 permutations
   2016-04-24 00:06:38 : Coastal v. Offshore.North
   2016-04-24 00:06:39 : Coastal v. Offshore.South
   2016-04-24 00:06:39 : Offshore.North v. Offshore.South

Population structure results:
                                  pair.label  Chi2 Chi2.p.val       D
1        Coastal (68) v. Offshore.North (40) 476.8   0.000999 0.37171
2        Coastal (68) v. Offshore.South (18) 438.9   0.000999 0.41159
3 Offshore.North (40) v. Offshore.South (18)  63.9   0.564436 0.00597
  D.p.val      Fst Fst.p.val    F'st F'st.p.val    Fis Fis.p.val     Gst
1 0.00106  0.13064  0.000999  0.5195   0.000999 0.0571     1.000  0.0626
2 0.00108  0.14641  0.000999  0.5656   0.000999 0.0134     0.993  0.0643
3 0.66319 -0.00417  0.784216 -0.0323   0.785215 0.0486     0.851 -0.0126
  Gst.p.val    G'st G'st.p.val  G''st G''st.p.val PHIst PHIst.p.val
1  0.000999  0.1178   0.000999  0.478    0.000999    NA          NA
2  0.000999  0.1208   0.000999  0.501    0.000999    NA          NA
3  0.735265 -0.0255   0.735265 -0.197    0.759241    NA          NA
   user  system elapsed
   4.49    1.28    2.09
```

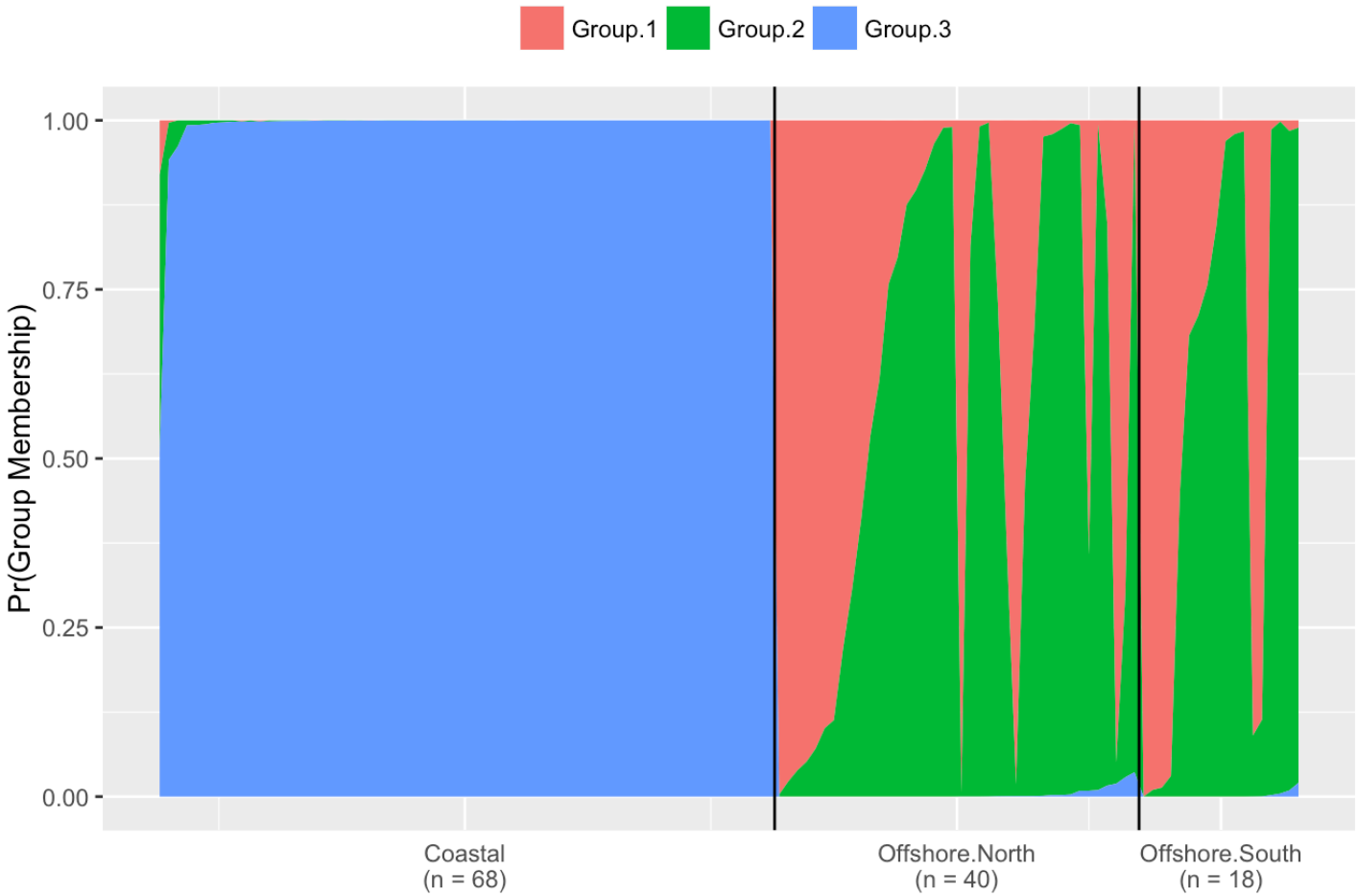
Interoperability

- Conversion functions
  - data.frames / matrices
  - other packages (adegenet, pegas, hierfstat, etc.)
  - popular formats (FASTA, Arlequin, MEGA, NEXUS, etc.)
- External programs
  - fastsimcoal
  - STRUCTURE / CLUMPP
  - PHASE
  - MAFFT
  - jModelTest

```
> library(adegenet)
> gi <- gtypes2genind(msats.g)
> msats.dapc <- dapc(gi, n.pca = 3, n.da = 5)
> scatter(msats.dapc)
```

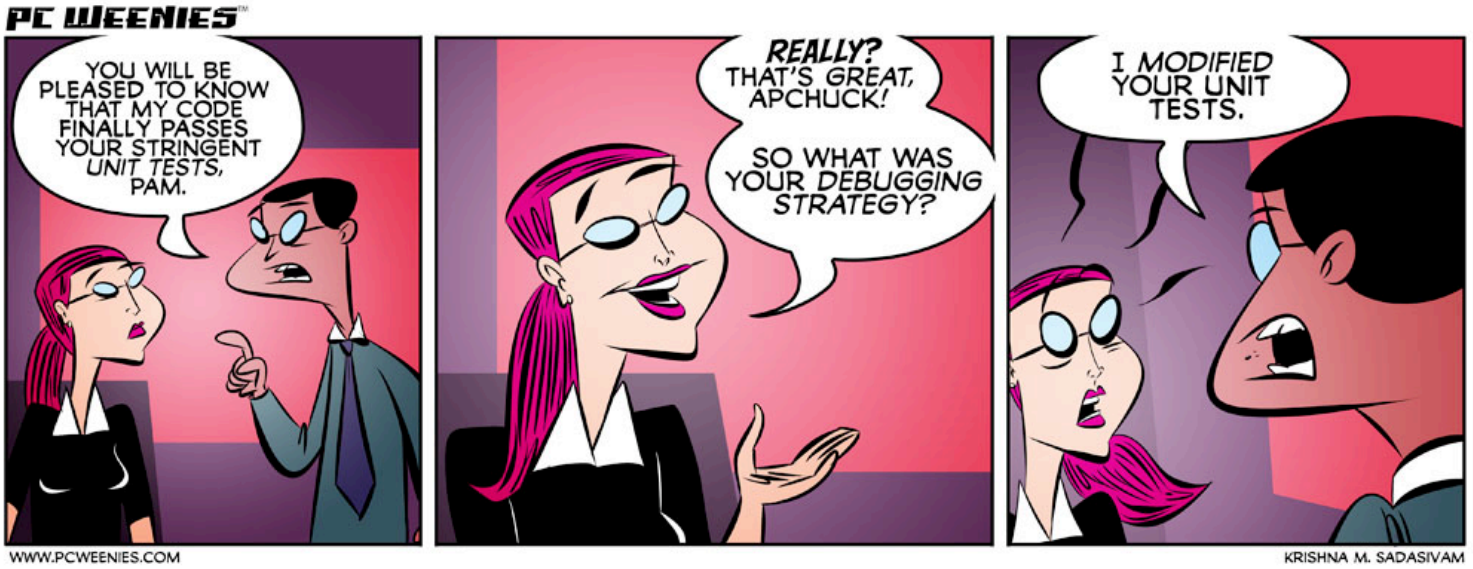


```
> sr <- structureRun(msats.g, k = 2:6, num.k.rep = 100)
> struct.clmp <- clumpp(sr, 3)
> structurePlot(struct.clmp, horiz = FALSE)
```



Validation

- Standardized datasets
- Missing data
- Modularity
- Unit tests (`testthat`)



Resources

- ✓ Help files (examples, references)
- ✓ Vignettes
- ○ Tutorials
- ○ Manuals
- ○ Web
  - repositories (GitHub)
  - maillists (r-sig-genetics, r-sig-phylogenetics)
  - fora (Google groups)

R Techniques

- Reading / writing .csv files
- Loading / saving R binary files (.rdata)
- Indexing with `numeric, character, logical`
- Working with `data.frames, matrices, lists, vectors`
- `sapply, lapply, apply`
- `for` loops

R Packages

Task View: Statistical Genetics
http://finzi.psych.upenn.edu/views/Genetics.html

TaskView: Phylogenetics, Especially Comparative Methods
http://finzi.psych.upenn.edu/views/Phylogenetics.html

Population Genetics
- adegenet
- apex
- poppr
- hierfstat
- pegas
- genetics
- diveRsity
- PopGenKit
- PopGenReport
- rmetasim

Phylogenetics
- ape
- phangorn
- phylobase
- phytools
- phylotools

Genomics
- Bioconductor
- WhopGenome
- PopGenome
- OutFLANK

Visualization
- ggplot2
- grid / gridExtra
- RColorBrewer
- igraph
- plotrix
- ggtree

Maillists
- r-sig-genetics
- r-sig-phylo
- r-sig-ecology

strataG

Get the latest version from GitHub:

```
> library(devtools)
> install_github("ericarcher/strataG", build_vignettes = TRUE)
```

Submit suggestions and bug-reports:

https://github.com/ericarcher/strataG/issues

Contact me:

eric.archer@noaa.gov