# Basic population genomic analysis in Python and R

```python
#pairwise IBS tract distribution function for summary stats
def getPairwiseIbsTractLengths(x,y,positions,maxlen,min_len_to_keep=0):
    '''
    input:
    x: haplotype as 1D array
    y: haplotype as 1D array
    positions: a 1D array of SNP positions
    maxlen: length of chromosome/contig

    Returns:
    1d array listing distances between adjacent SNPs in a pair of sequences.
    Assumes all sites are accessible.
    '''
    snps=~np.equal(x,y)
    snp_positions=positions[snps]
    l=len(snp_positions)
    ibs_tracts=[]
    if(l==0):
        ibs_tracts=[maxlen]
    else:
        if(l>1):
            ibs_tracts=snp_positions[np.arange(1,l-1,1)]-snp_positions[np.arange(0,l-2,1)] #middle blocks
        np.append(ibs_tracts,snp_positions[0]+1)        #first block
        np.append(ibs_tracts,maxlen-snp_positions[l-1]) #last block
        con=[x>=min_len_to_keep for x in ibs_tracts] #drop blocks < min_len_to_keep
        ibs_tracts=np.extract(con,ibs_tracts)
    return ibs_tracts
```

**Ethan Linck**

Postdoctoral Research Associate, University of Tennessee, Knoxville

Research Affiliate, Taylor Lab, CU Boulder

## How do we analyze population genetic and genomic data?

# GUIs

**Advantages:**
-simple to use
-widely cited

**Disadvantages:**
-opaque
-inflexible
-not very reproducible

**Examples:**
-DNAsp (http://www.ub.edu/dnasp/)
-Arlequin (http://cmpg.unibe.ch/software/arlequin35/)
-Geneious (https://www.geneious.com/prime/)

# Command line + sketchy bash scripts

**Advantages:**

-more or less the norm
-huge ecosystem of tools to draw on
-*can be* robust and reproducible, with effort
-relatively quickly get what you want

**Disadvantages:**

-perpetual data format conversion
-easy to lose track of what you've done and how you've done it
-different assumptions, levels of quality control across programs
-most in-depth analyses require real scripting

**Examples:**

-vcftools (https://vcftools.github.io/examples.html)
-angsd (http://www.popgen.dk/angsd/index.php/ANGSD)

# Programming languages

**Advantages:**

-hugely flexible (scriptable)
-growing ecosystem of tools to draw on
-more easily made robust and reproducible than bash scripts
-necessary for sophisticated analyses

**Disadvantages:**

-steeper learning curve
-most filtering / QC still done via command line; fewer resources for this
-quality of software can vary depending on goals

**Examples:**

-R (https://www.r-project.org/)
-Python (https://www.python.org/)
-Perl (https://www.perl.org/)
-C++ (http://www.cplusplus.com/)
-Juila (https://julialang.org/)

# R

**Advantages:**

-widely used by empiricists
-easily integrated with huge array of statistical tools
-better resources for low quality data / nonmodel organisms
-better plotting libraries (sorry, it's true)

**Disadvantages:**

-arguably hard to read, quirky syntax
-can be slow unless you really know what you're doing
-not optimized for sequence data
-not as easily integrated with system calls

**Example :**

-adegenet (https://github.com/thibautjombart/adegenet/wiki)
-PopGenome (https://cran.r-project.org/web/packages/PopGenome/index.html)
-poppr (https://grunwaldlab.github.io/poppr/)
-vcfR (https://cran.r-project.org/web/packages/vcfR/index.html)

# Python

**Advantages:**

-easy to read / learn
-easier to write fast code, particularly for sequence data / large files
-integration with msprime, treesequence tools
-integration with system calls / command line tools
-strengths in machine learning, data parsing / collation

**Disadvantages:**

-can be unfriendly if you work with nonomodel organisms / bad data
-smaller ecosystem of tools for downstream analyses
-plotting libraries clunky, have uglier output

**Examples:**

-biopython (https://biopython.org/)
-scikit-allel (https://scikit-allel.readthedocs.io/en/stable/#)

# Tutorial: getting started

1) Download miniconda
(https://docs.conda.io/en/latest/miniconda.html)

2) Open Terminal, download jupyter

```
conda install jupyter
```

3) Set your directory somewhere harmless:

```
set wd ~/Desktop/
```

4) Clone the tutorial repository:

```
https://github.com/elinck/popgen_analysis_tutorial.git
```

5) Open the notebook:

```
jupyter notebook scikit_allel_tutorial.ipynb
```

# Resources

**Online tutorial**
https://github.com/elinck/popgen_analysis_tutorial/

**These slides**
https://github.com/elinck/popgen_analysis_tutorial/

**scikit-allel documentation**
https://scikit-allel.readthedocs.io/en/stable/