Python for Personal and Population Genome Interpretation

Manuel A. Rivas (marivascruz)
University of Oxford

PyData
NYC 2014

https://github.com/marivascruz/pydata2014genomics

Presentation, iPython Notebook, and data sets

About Me

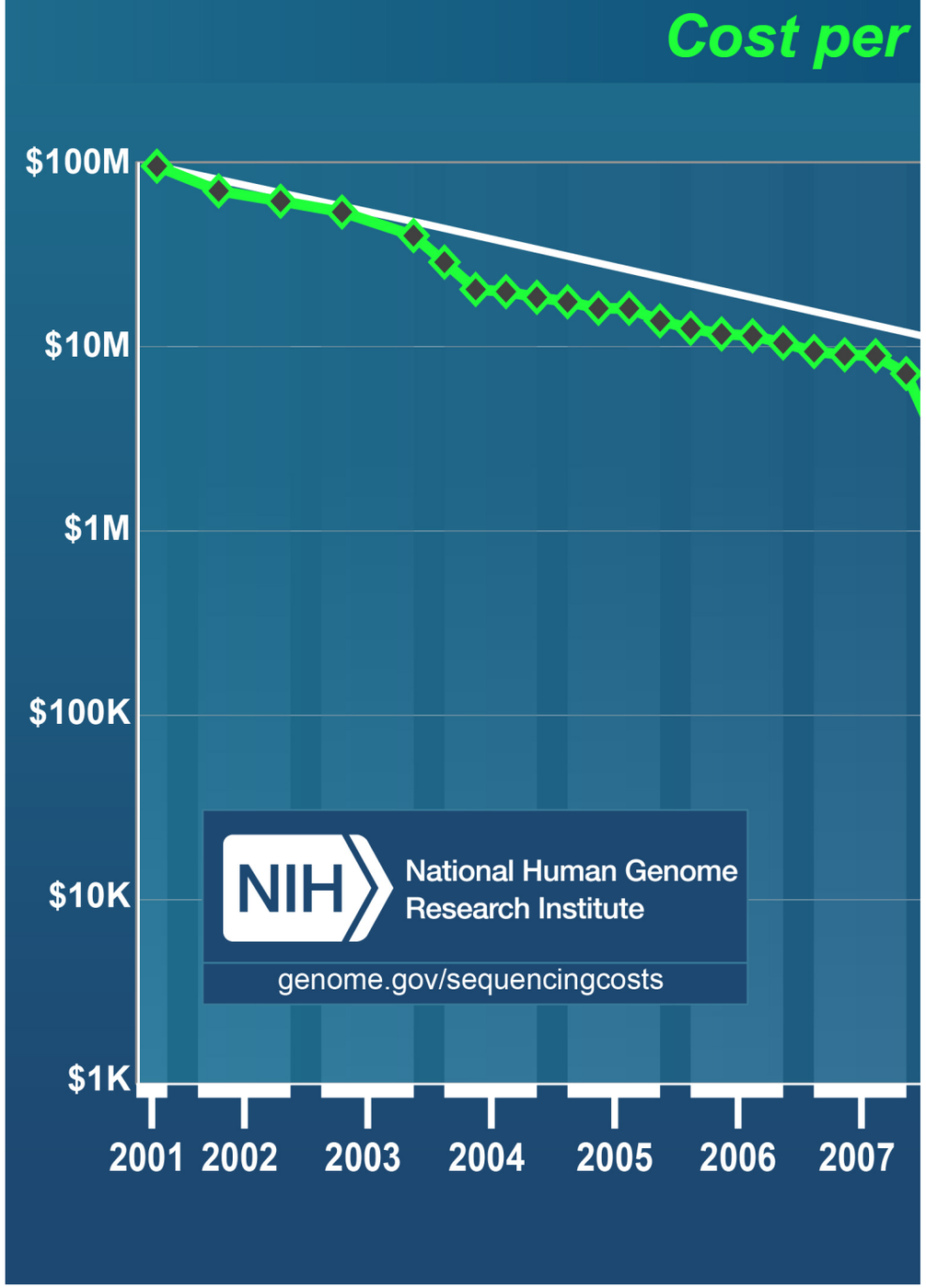
My first time at a Python Conference

(Highly recommended by friends)

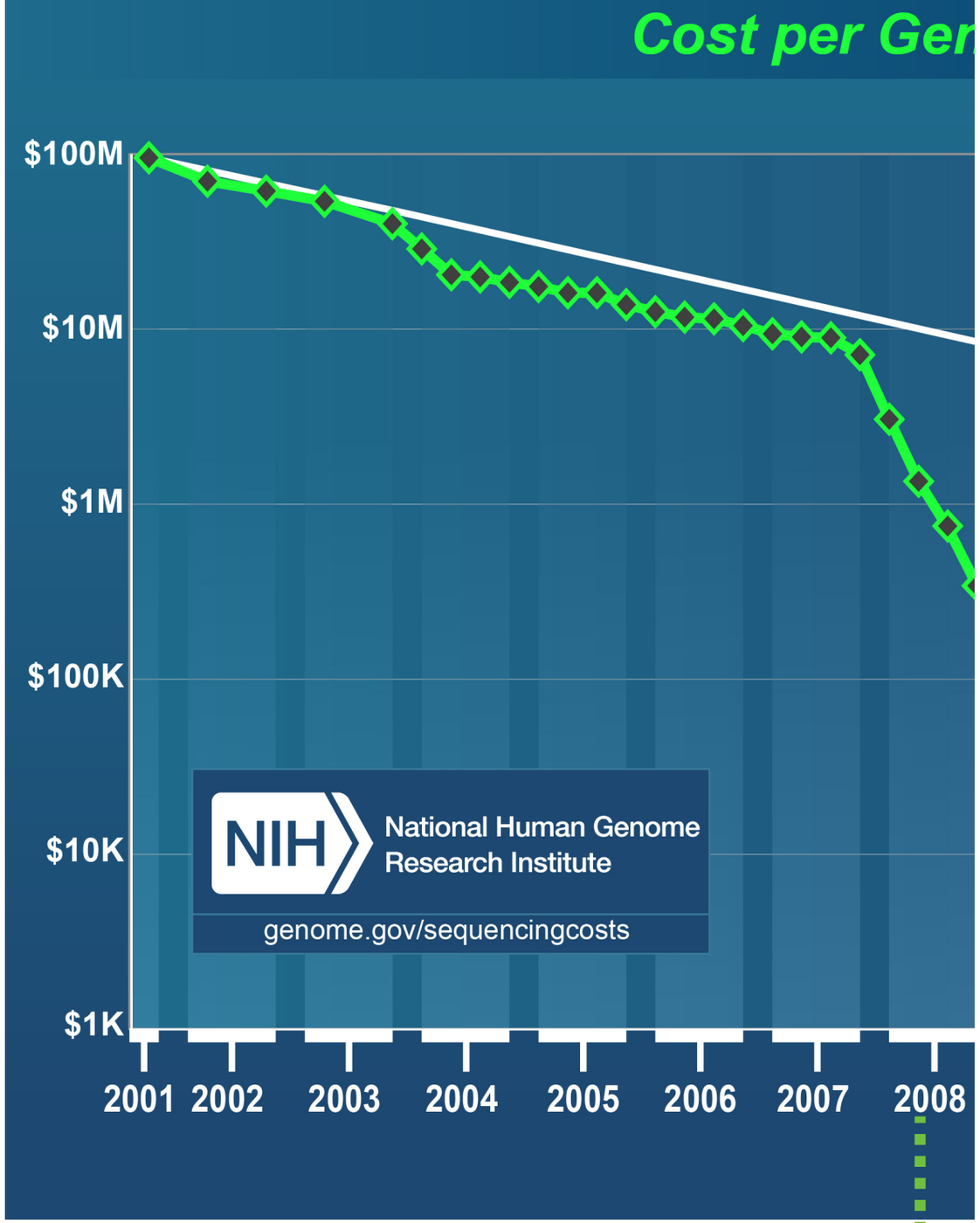
- **Graduate Student at University of Oxford**
- **Main interests include:**
  - Development of statistical and computational methods for genetic studies of common diseases
  - Developing tools
- **Non-scientific interests include:**
  - Dancing
  - Traveling

Currently living in a Scientific Revolution:

Genomics



Catalyzed by the drop in the cost of DNA sequencing

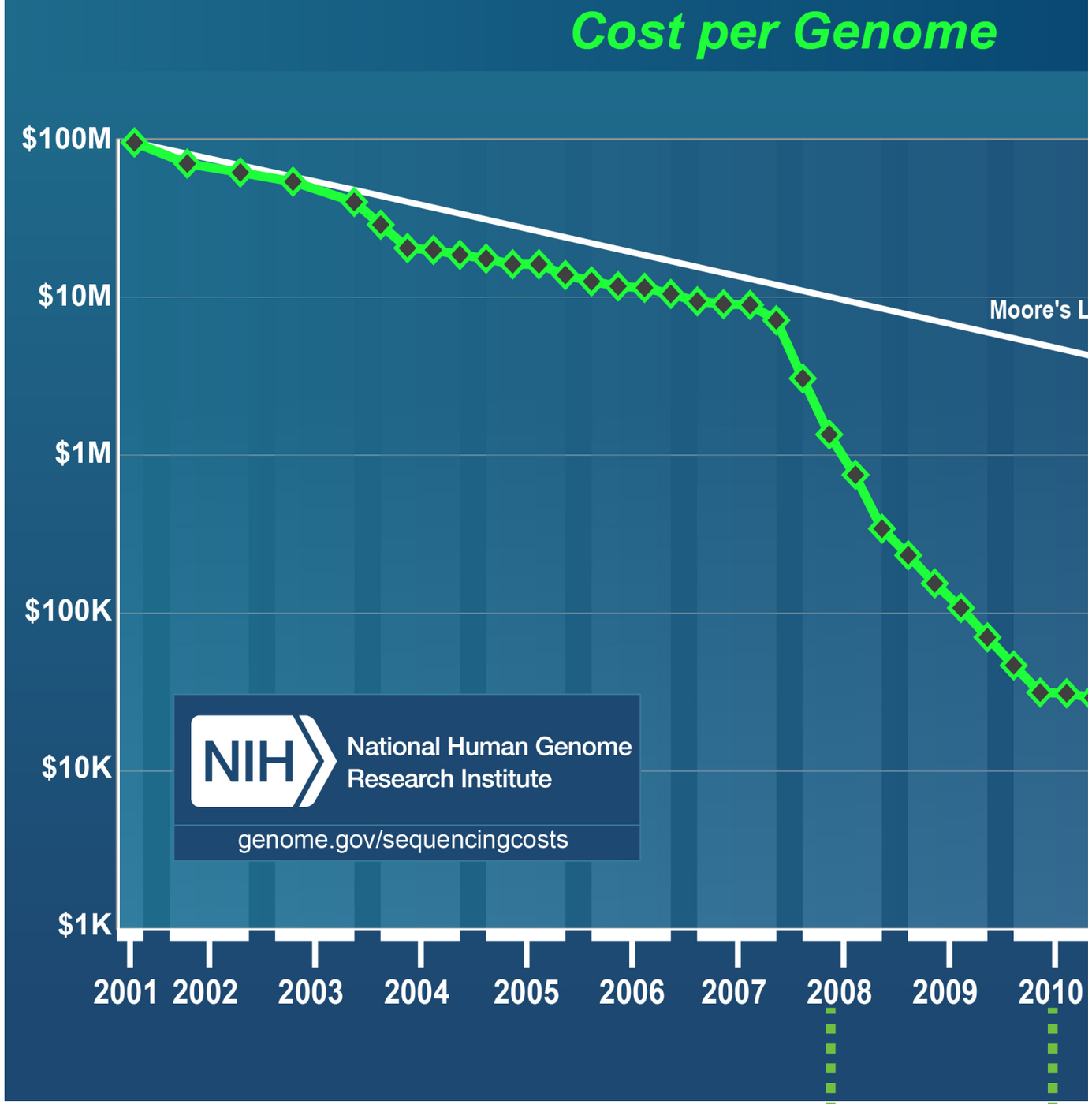


What was happening
in my life at that moment

. . .

Graduated
from MIT

Catalyzed by the drop in the cost of DNA sequencing



What was happening
in my life at that moment

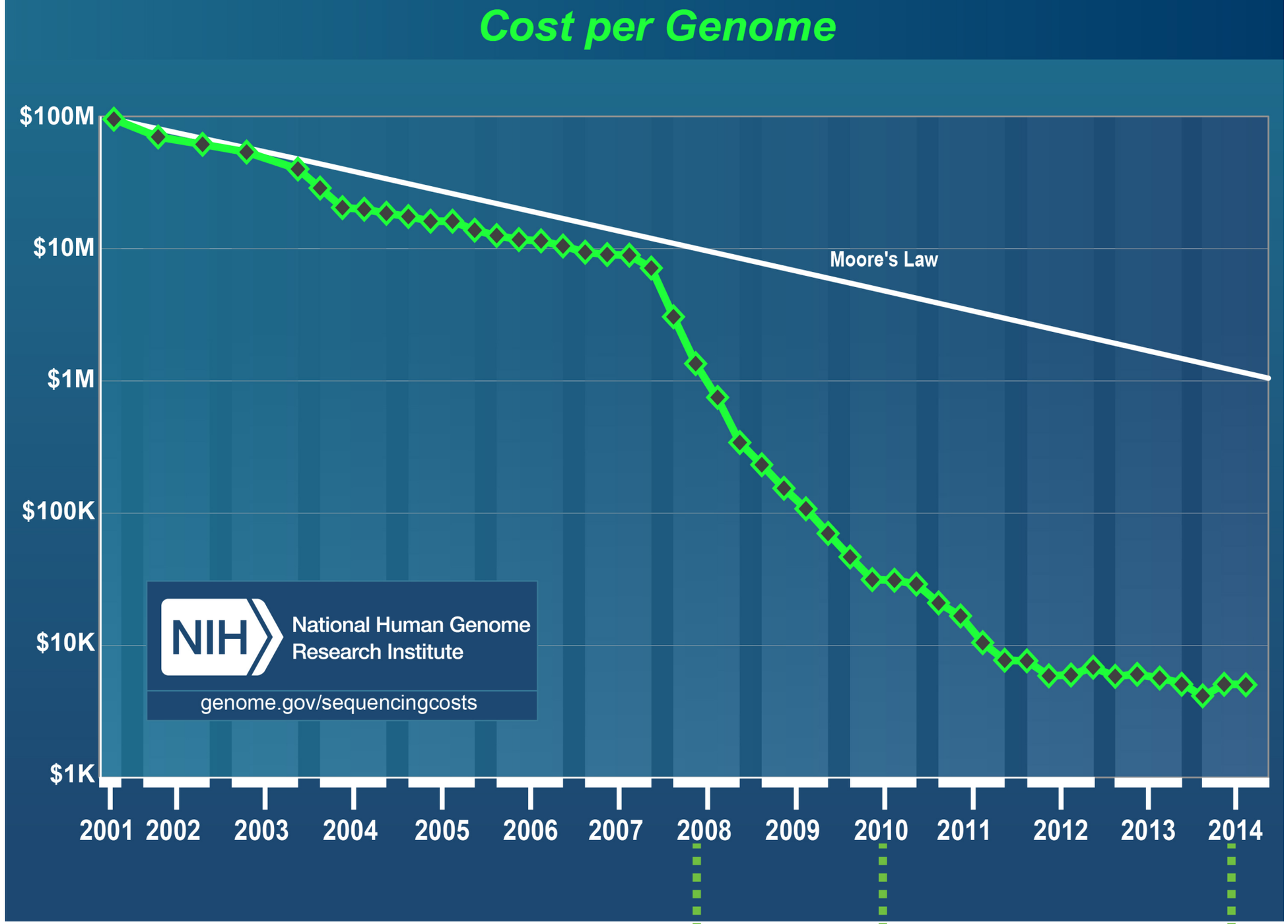
. . .

Graduated
from MIT

Went to
England

Worked
@ Broad
Institute

Catalyzed by the drop in the cost of DNA sequencing



What was happening in my life at that moment

. . .

Graduated from MIT

Went to England

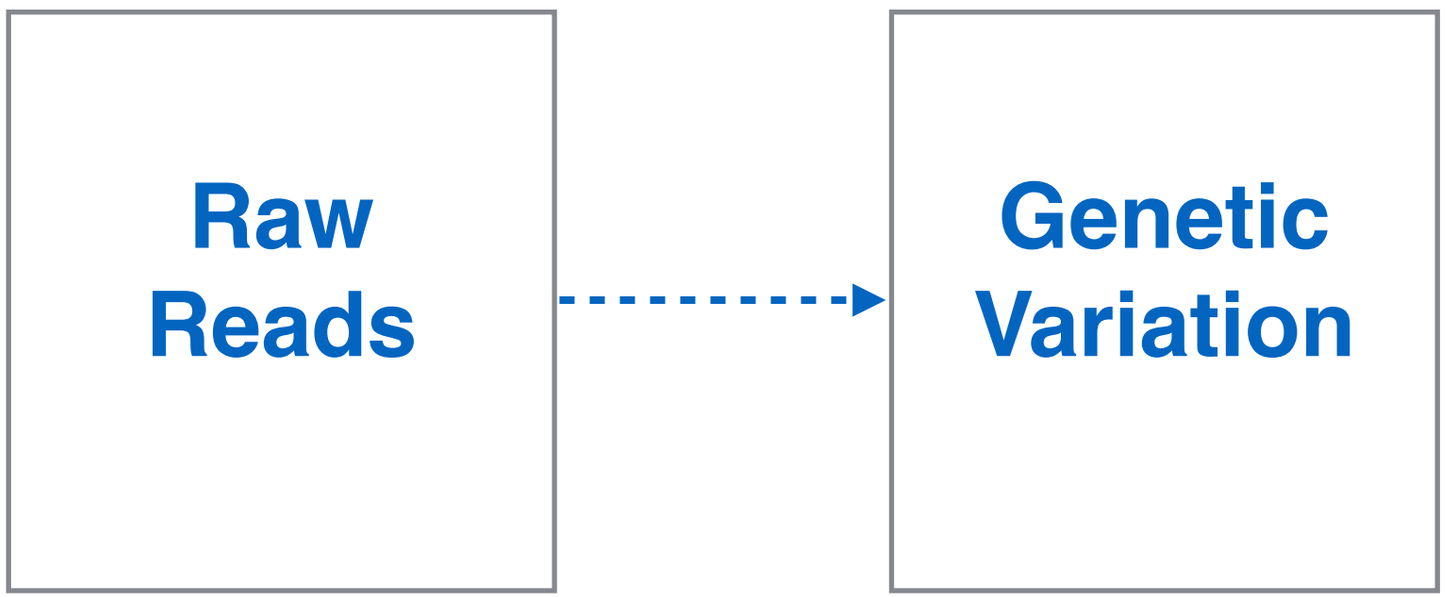
Process of Graduating

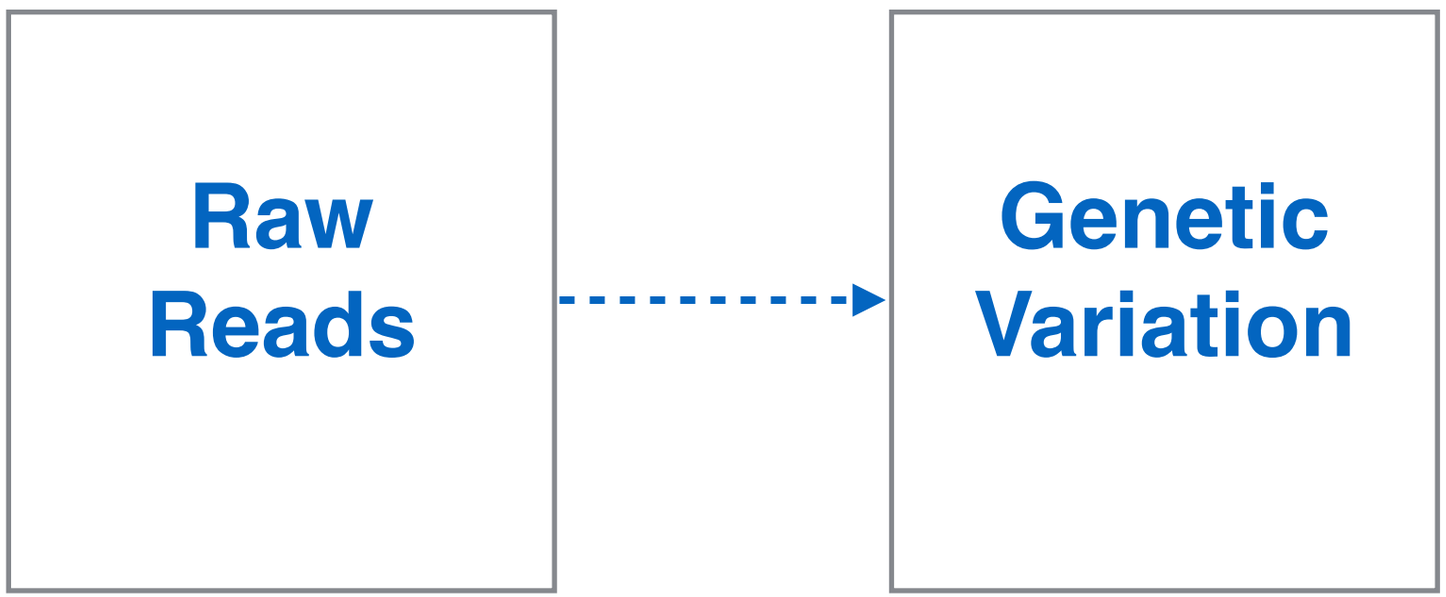
Worked @ Broad Institute

Catalyzed by the drop in the cost of DNA sequencing

Main Challenge?

How to analyze the Data

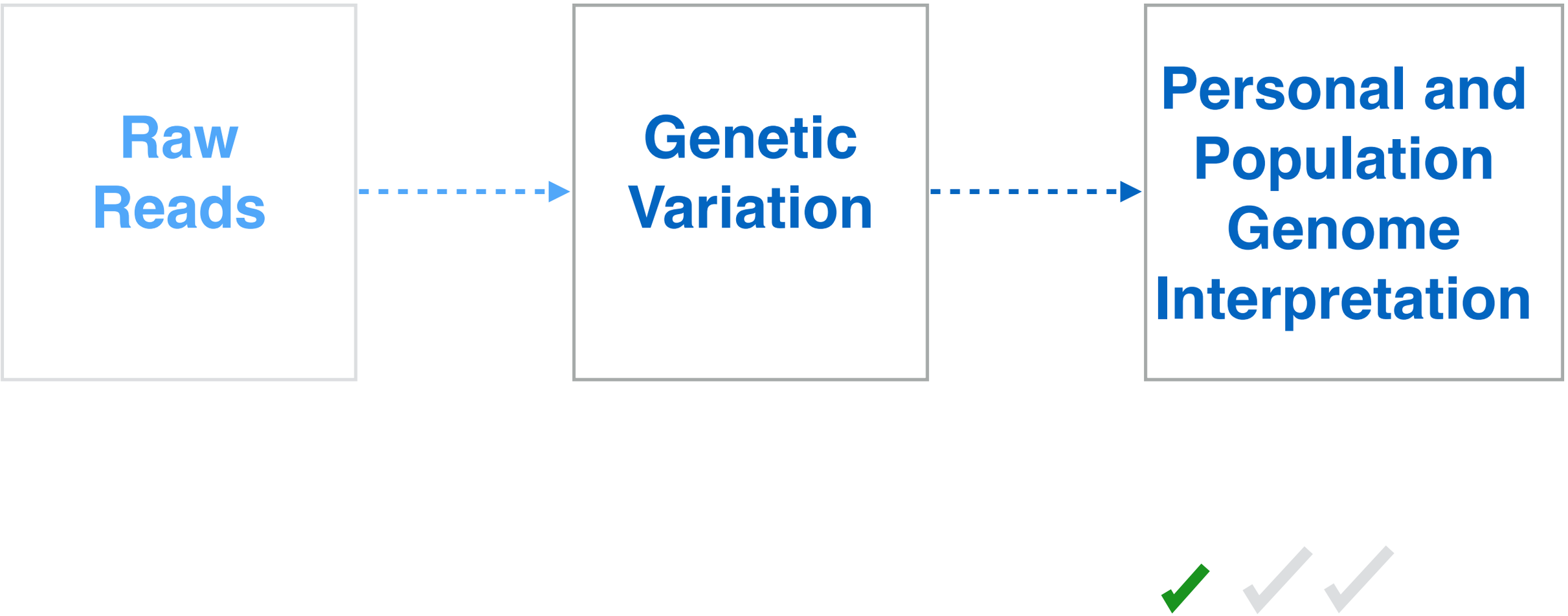

Raw
Reads
Genetic
Variation



Arduous research and engineering problem

2008 — 2014

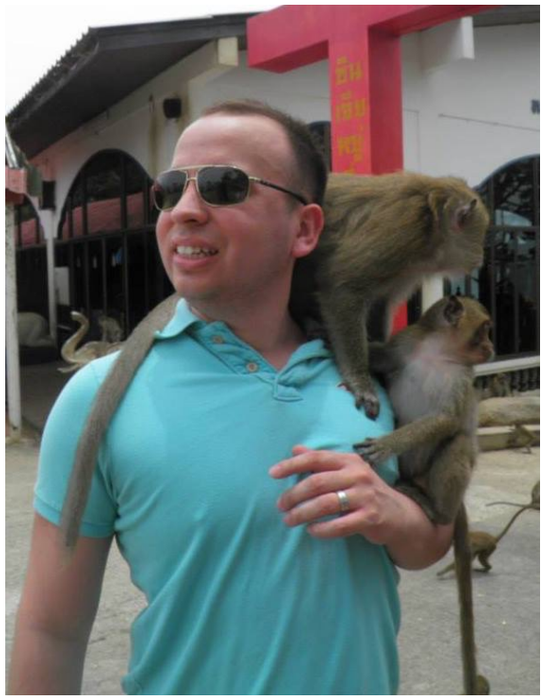
Room for innovation and improvement, e.g.
Fully phased genomes,
Structural variant detection,
Somatic mutations


Raw Reads
Genetic Variation
Personal and Population Genome Interpretation

PLINK/SEQ

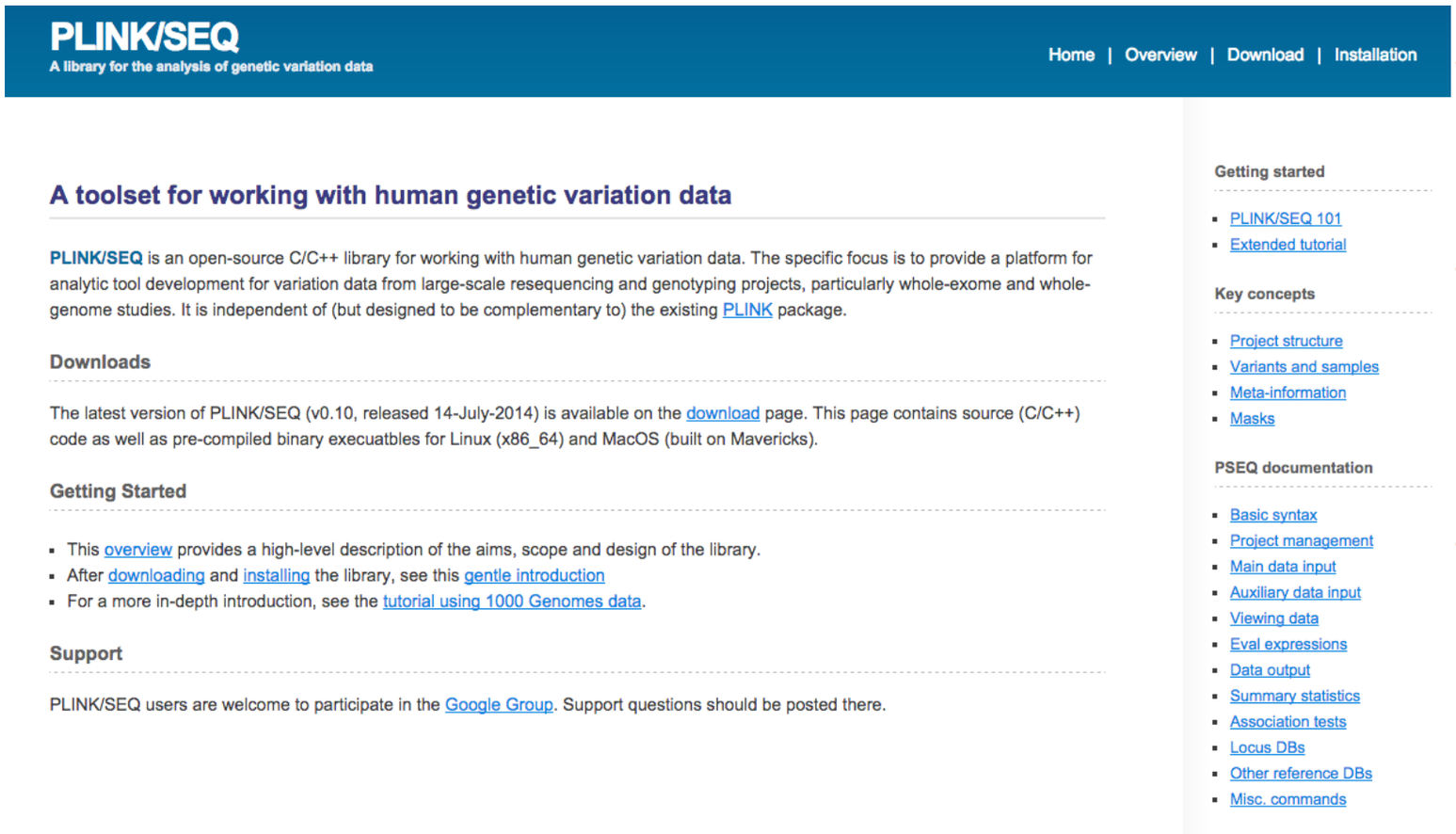
Shaun Purcell - who has also developed other widely used tools including PLINK (> 7000 citations)

PLINK/SEQ



PLINK/SEQ

a toolset for next generation sequencing (NGS) data sets



PLINK/SEQ
A library for the analysis of genetic variation data

Home | Overview | Download | Installation

A toolset for working with human genetic variation data

PLINK/SEQ is an open-source C/C++ library for working with human genetic variation data. The specific focus is to provide a platform for analytic tool development for variation data from large-scale resequencing and genotyping projects, particularly whole-exome and whole-genome studies. It is independent of (but designed to be complementary to) the existing PLINK package.

Downloads

The latest version of PLINK/SEQ (v0.10, released 14-July-2014) is available on the download page. This page contains source (C/C++) code as well as pre-compiled binary execuatbles for Linux (x86_64) and MacOS (built on Mavericks).

Getting Started

- This overview provides a high-level description of the aims, scope and design of the library.
- After downloading and installing the library, see this gentle introduction
- For a more in-depth introduction, see the tutorial using 1000 Genomes data.

Support

PLINK/SEQ users are welcome to participate in the Google Group. Support questions should be posted there.

Getting started

- PLINK/SEQ 101
- Extended tutorial

Key concepts

- Project structure
- Variants and samples
- Meta-information
- Masks

PSEQ documentation

- Basic syntax
- Project management
- Main data input
- Auxiliary data input
- Viewing data
- Eval expressions
- Data output
- Summary statistics
- Association tests
- Locus DBs
- Other reference DBs
- Misc. commands

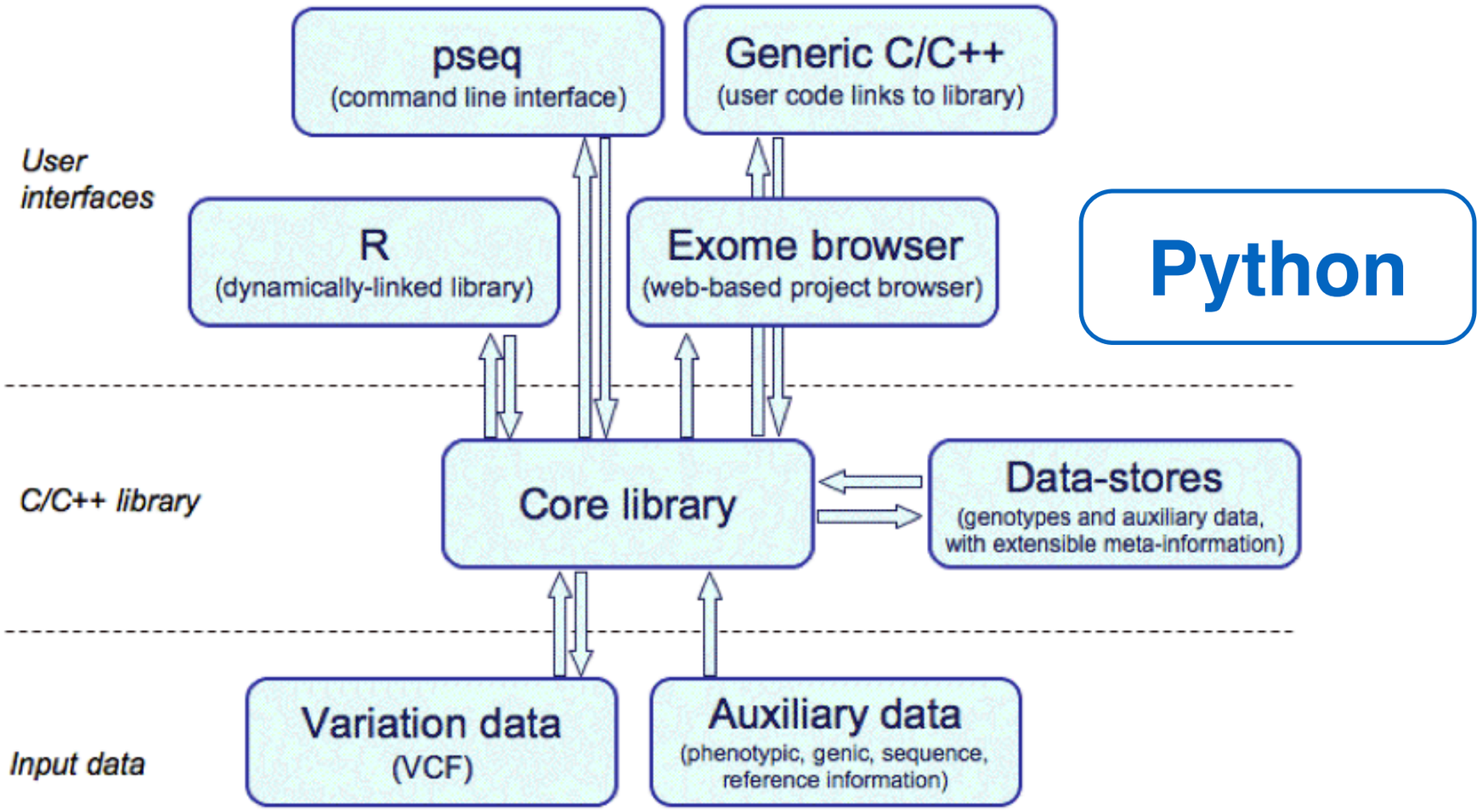
- VCF as primary input
- Focus on analysis of rare variants
- Extensible meta-information on locus, genotypes, individuals
- Bundled with key reference databases that can be directly intersected with one's own data
- Command-line, R, and Python library

http://atgu.mgh.harvard.edu/plinkseq/

pyplinkseq

Python package with extensions to the PLINK/SEQ library

w/ Jeff Hammerbacher


pseq
(command line interface)
Generic C/C++
(user code links to library)
User interfaces
R
(dynamically-linked library)
Exome browser
(web-based project browser)
Python
C/C++ library
Core library
Data-stores
(genotypes and auxiliary data, with extensible meta-information)
Input data
Variation data
(VCF)
Auxiliary data
(phenotypic, genic, sequence, reference information)

23andMe Raw Data

23andMe

HOME MY RESULTS FAMILY & FRIENDS RESEARCH & COMMUNITY

Manuel Rivas 1

DOWNLOAD RAW DATA

HELP

« Return to browse raw data

About raw data download:

- The download is a zipped text file 5 MB to 30 MB in size.
- The text file consists of lines of your genotype call data (your A's, T's, C's and G's).
- It can be opened in a text editor like WordPad.

Before downloading, please consider the following:

Utility – Keep in mind that having your data in hand may be of limited practical usefulness, depending on how much information you can extract from the data beyond what the 23andMe site already gives you.

Security – As noted in the 23andMe privacy statement, what you do with your data is your responsibility, whether that means sharing your login name and password with others, sharing through 23andMe, downloading your data or anything else. Please note that when you opt to download your data, that data is no longer secured by the layers of encryption provided on our servers.

Quality – The rapid advance of genotyping technology means that our ability to interpret your raw chip data will improve over time, and thus the version of your genome residing on our servers will improve in both completeness and accuracy. For that reason, customers who want the best available data should occasionally check back and update their downloaded files. See a log of updates and changes to the raw data download.

23andMe Raw Data

```
-bash-4.1$ head -n 20 genome_Manuel_Rivas_Full_20130423150620.txt
# This data file generated by 23andMe at: Tue Apr 23 15:06:20 2013
#
# Below is a text version of your data.  Fields are TAB-separated
# Each line corresponds to a single SNP.  For each SNP, we provide its identifier
# (an rsid or an internal id), its location on the reference human genome, and the
# genotype call oriented with respect to the plus strand on the human reference sequence.
# We are using reference human assembly build 37 (also known as Annotation Release 104).
# Note that it is possible that data downloaded at different times may be different due to ongoing
# improvements in our ability to call genotypes. More information about these changes can be found at:
# https://www.23andme.com/you/download/revisions/
#
# More information on reference human assembly build 37 (aka Annotation Release 104):
# http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi?taxid=9606
#
# rsid  chromosome      position        genotype
rs4477212       1       82154   AA
rs3094315       1       752566  AA
rs3131972       1       752721  AG
rs12124819      1       776546  AA
rs11240777      1       798959  AG
-bash-4.1$
```

https://github.com/hammer/personal-genome-analysis/blob/master/scripts/23andMetoVCF.py

VCF (Variant Call Format) File

```
bash-4.1$ head -n 10 marivas.vcf
##fileformat=VCFv4.1
##source=pseq
##FILTER=<ID=PASS,Description="Passed variant FILTERs">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  RIVAS
chr1    82154   rs4477212       A       0       .       .       .       GT      1/1
chr1    752566  rs3094315       A       0       .       .       .       GT      1/1
chr1    752721  rs3131972       G       A       .       .       .       GT      0/1
chr1    776546  rs12124819      A       0       .       .       .       GT      1/1
chr1    798959  rs11240777      G       A       .       .       .       GT      0/1
chr1    800007  rs6681049       T       C       .       .       .       GT      0/1
bash-4.1$
```

https://github.com/hammer/personal-genome-analysis/blob/master/scripts/23andMetoVCF.py

VCF (Variant Call Format) File

```
bash-4.1$ pseq marivas new-project --vcf marivas.vcf
Creating new project specification file [ marivas.pseq ]
bash-4.1$ pseq marivas load-vcf
loading : /gpfs1/well/rivas/23andme/rivas/marivas.vcf ( 1 individuals )
parsed 922000 rows
/gpfs1/well/rivas/23andme/rivas/marivas.vcf : inserted 922921 variants
```

Working with pyplinkseq

```
In [1]: import pyplinkseq

In [2]: pyplinkseq.set_project('marivas.pseq')

In [3]: print pyplinkseq.summary()
---File-index summary---

Core project specification index : marivas.pseq
Core OUTPUT file : /gpfs1/well/rivas/23andme/rivas/marivas_out/
Core RESOURCES file : /gpfs1/well/rivas/23andme/rivas/marivas_res/
Core LOCDB file : /gpfs1/well/rivas/23andme/rivas/marivas_res/locdb
Core INDDB file : /gpfs1/well/rivas/23andme/rivas/marivas_out/inddb
Core VARDB file : /gpfs1/well/rivas/23andme/rivas/marivas_out/vardb
Core LOG file : /gpfs1/well/rivas/23andme/rivas/marivas_out/log.txt
Core SEQDB file : /gpfs1/well/rivas/23andme/rivas/marivas_res/seqdb
Core REFDB file : /gpfs1/well/rivas/23andme/rivas/marivas_res/refdb
Added VCF : /gpfs1/well/rivas/23andme/rivas/marivas.vcf

---Variant DB summary---

922921 unique variants
File tag : 1 (922921 variants, 1 individuals)
```

Summary of PLINK/SEQ project

Working with Auxiliary databases

```
In [4]: pyplinkseq.locdbattach('/well/rivas/got2dtmp/locdb')

In [5]: print pyplinkseq.locdb_summary()
---Locus DB summary---

Group : refseq (47421 entries) n/a
Group : ccds (25471 entries) n/a
Group : gencode (94378 entries) n/a
Group : ensembl (158282 entries) n/a
```

https://atgu.mgh.harvard.edu/plinkseq/resources.shtml

Attach locus databases.
Largely focused on reference transcript sets.

Working with Auxiliary databases

```
In [8]: print pyplinkseq.seqdb_summary()
---Sequence DB summary---

chr1:1..249250621       MB=249
chr2:1..243199373       MB=243
chr3:1..198022430       MB=198
chr4:1..191154276       MB=191
chr5:1..180915260       MB=180
chr6:1..171115067       MB=171
chr7:1..159138663       MB=159
chr8:1..146364022       MB=146
chr9:1..141213431       MB=141
chr10:1..135534747      MB=135
chr11:1..135006516      MB=135
chr12:1..133851895      MB=133
chr13:1..115169878      MB=115
chr14:1..107349540      MB=107
chr15:1..102531392      MB=102
chr16:1..90354753       MB=90
chr17:1..81195210       MB=81
chr18:1..78077248       MB=78
chr19:1..59128983       MB=59
chr20:1..63025520       MB=63
chr21:1..48129895       MB=48
chr22:1..51304566       MB=51
chrX:1..155270560       MB=155
chrY:1..59373566        MB=59
chrM:1..16571   MB=0

SEQDB meta-information: BUILD = hg19
SEQDB meta-information: DESC = from-UCSC-20-dec-2010
SEQDB meta-information: IUPAC = 0
SEQDB meta-information: NAME = hg19
SEQDB meta-information: REPEATMODE = lower
```

Attach sequence databases.
Human Genome Build 19.

DNA sequence variant annotation

Load a locus set

```
In [9]: pyplinkseq.annotate_load('refseq')
```

Annotate a variant

```
In [10]: pyplinkseq.annotate(9,125391241,'G','A','annot','')
Out[10]: 'nonsense'
```

Fetch genotypes

```
In [11]: myvar = pyplinkseq.var_fetch("reg=chr9:125391241")
```

Check my genotype

```
In [12]: myvar[0].CON.GENO.GT[0]
Out[12]: 1
```

(I am heterozygous for variant chr9:125391241 -
a human knock(down)out version for gene X)

Masks

Easily filter with mask syntax.

Used for extracting information from the various databases available.

```
In [11]: myvar = pyplinkseq.var_fetch("reg=chr9:125391241")
```

This example we simply used the mask syntax to fetch genotype data for DNA sequence variant chr9:125391241.

Masks

Other applications

Subset individuals in a project (obtain phenotypes for only certain individuals)

Subset variants based on some quality control filters

Subset variants based on population based calculations (Hardy-Weinberg Equilibrium)

Easily allows inclusion of additional filters: inclusion/exclusion criteria for including genes, variants, individuals, pathways, networks, etc.

https://atgu.mgh.harvard.edu/plinkseq/masks.shtml

pyplinkseq for statistical methods development

Working with population-scale genomes

pyplinkseq for statistical methods development

MAMBA

tackling problems in medical genomics.

Working with population-scale genomes

pyplinkseq for statistical methods development

MAMBA

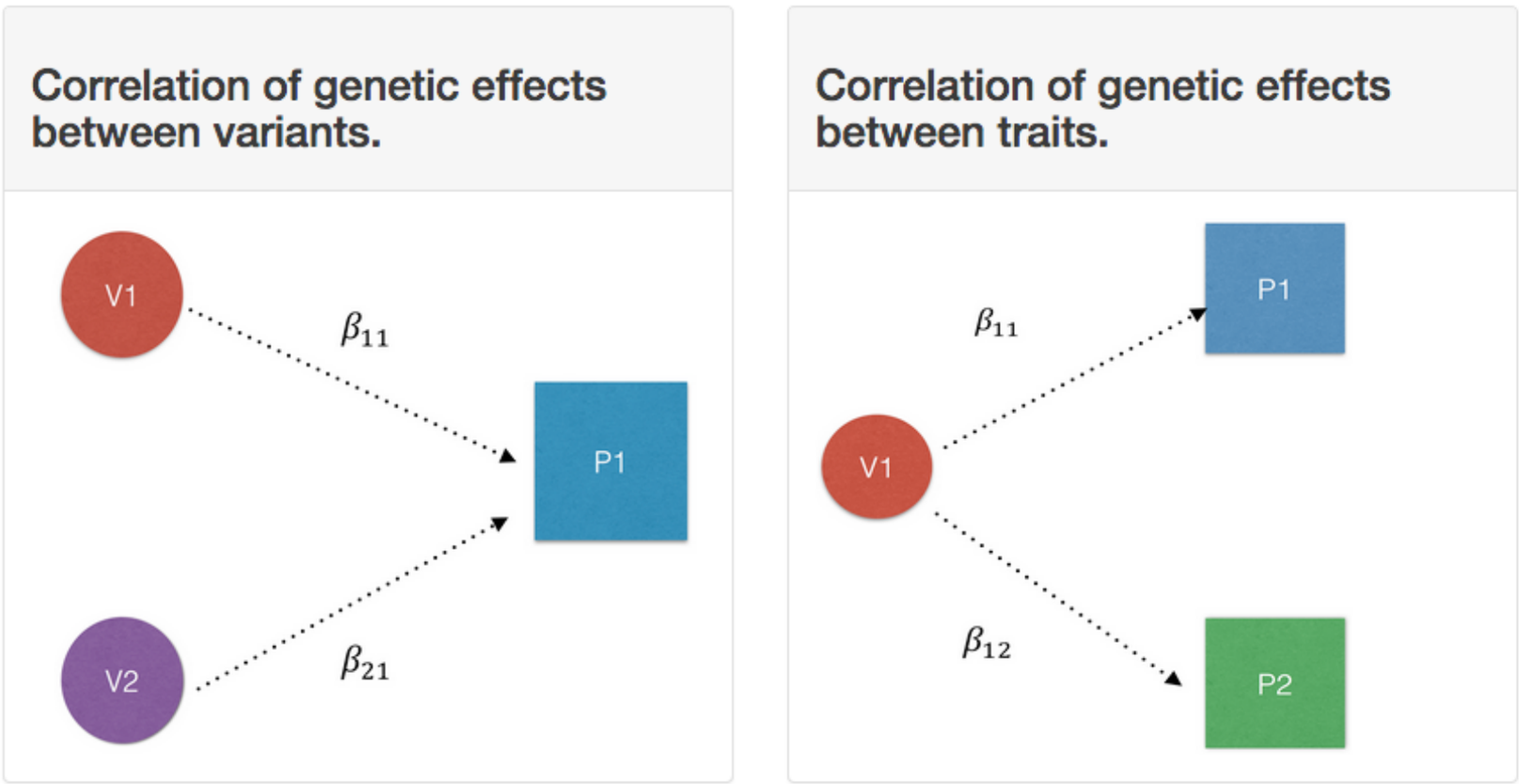
tackling problems in medical genomics.

Focused on
Genetic association
Statistical modeling
Allele specific expression

Working with population-scale genomes

pyplinkseq for statistical methods development

MRP C-alpha for cross disorder and cross phenotype analysis



For example, analyzing genome sequencing data with high-dimensional phenotypes

Working with population-scale genomes

pyplinkseq for statistical methods development

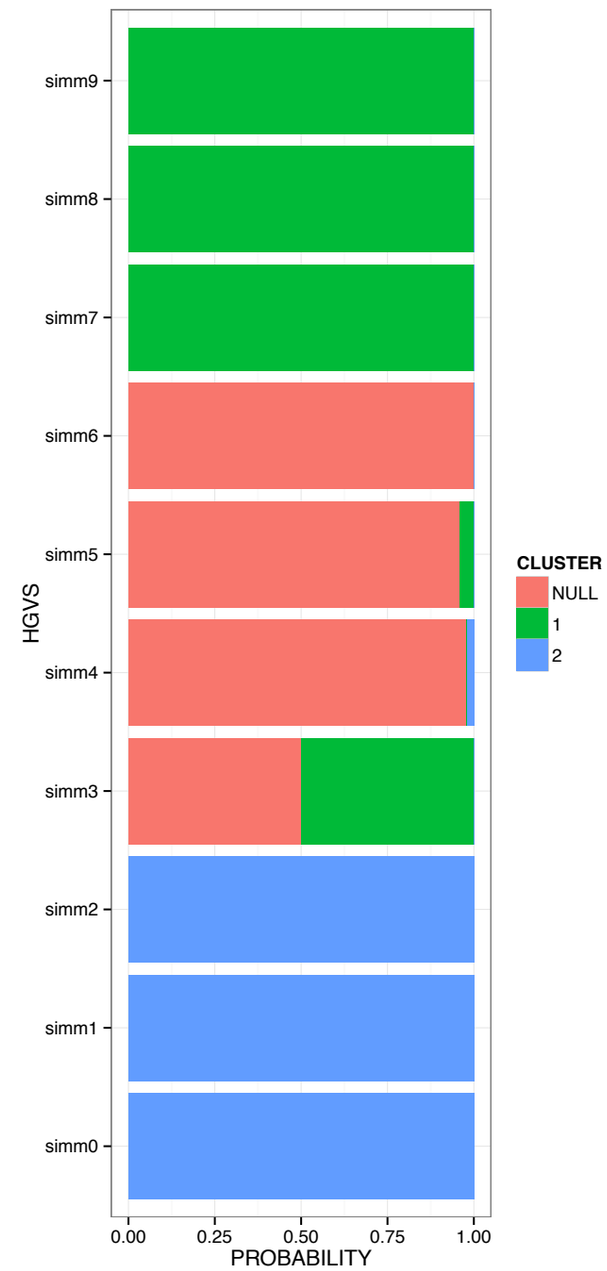
After import pyplinkseq. Let's import mamba.

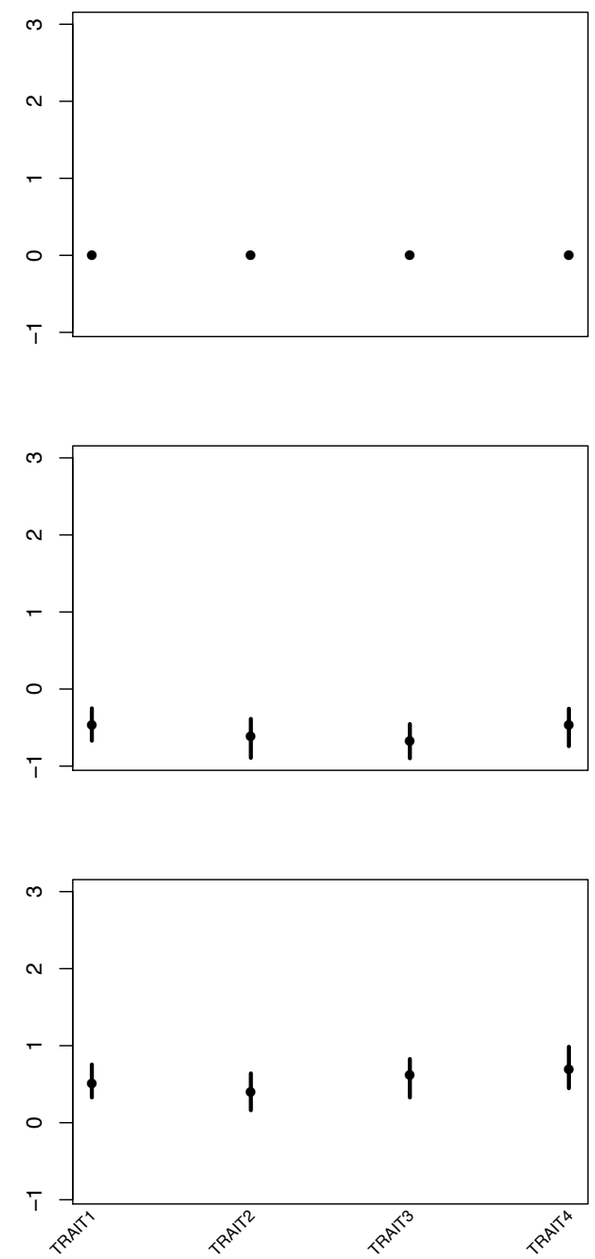
In [13]: import mamba

In [14]: from mamba.MRPQT import runmcmcapproach

Easy one-liner for assessing drivers of association. Phenotypes and Genotypes obtained with pyplinkseq

In [15]: runmcmcapproach(phenotypes, genotypes,clusters,iterations)





Working with population-scale genomes

Do you have an interest in developing Genomics applications with Python?

E-mail mrivas08@alum.mit.edu