Tissue Antigens ISSN 0001-2815

# PyPop update – a software pipeline for large-scale multilocus population genomics

A. K. Lancaster[1], R. M. Single[2], O. D. Solberg[1], M. P. Nelson[1] & G. Thomson[1]

1 Department of Integrative Biology, University of California, Berkeley, Berkeley, CA, USA
2 Department of Mathematics & Statistics, University of Vermont, Burlington, VT, USA

**Key words**
bioinformatics; HLA; immunogenetics; open-source software; population genetics

**Correspondence**
Alex Lancaster
Department of Ecology and Evolutionary Biology
University of Arizona
1041 E. Lowell St
PO Box 210088
Tuscon AZ 85721
USA
Tel: 1 520 626 1727
Fax: 1 520 621 9190
e-mail: alexlanc@u.arizona.edu

doi: 10.1111/j.1399-0039.2006.00769.x

## Abstract

Population genetic statistics from multilocus genotype data inform our understanding of the patterns of genetic variation and their implications for evolutionary studies, generally, and human disease studies in particular. In any given population one can estimate haplotype frequencies, identify deviation from Hardy–Weinberg equilibrium, test for balancing or directional selection, and investigate patterns of linkage disequilibrium. Existing software packages are oriented primarily toward the computation of such statistics on a population-by-population basis, not on comparisons among populations and across different statistics. We developed PyPop (Python for Population Genomics) to facilitate the analyses of population genetic statistics across populations and the relationships among different statistics within and across populations. PyPop is an open-source framework for performing large-scale population genetic analyses on multilocus genotype data. It computes the statistics described above, among others. PyPop deploys a standard Extensible Markup Language (XML) output format and can integrate the results of multiple analyses on various populations that were performed at different times into a common output format that can be read into a spreadsheet. The XML output format allows PyPop to be embedded as part of a larger analysis pipeline. Originally developed to analyze the highly polymorphic genetic data of the human leukocyte antigen region of the human genome, PyPop has applicability to any kind of multilocus genetic data. It is the primary analysis platform for analyzing data collected for the Anthropological component of the 13th and 14th International Histocompatibility Workshops. PyPop has also been successfully used in studies by our group, with collaborators, and in publications by several independent research teams.

## Introduction

PyPop (Python for Population Genomics) is a framework for performing large-scale population genetic analyses on multilocus genotype data. It contains several programs and an Application Programming Interface (API) implemented in the programming language Python. Tests that PyPop currently implements are summarized in Table 1. The output of the population analyses is stored in the XML format (the platform-independent, open standard for data storage). Since the description of a prerelease alpha version (1), PyPop has undergone substantial revision and expansion in its functionality.

This paper focuses on new features of PyPop which include the prefiltering of the input genotype data, the ability to translate arbitrary allele names into full amino acid or nucleotide sequences and a new implementation of a Monte-Carlo version of the 'exact' test for deviation from Hardy–Weinberg proportions (HWP). PyPop was originally developed for the analysis of highly polymorphic human leukocyte antigen (HLA) data for the 13th International Histocompatibility Workshop (IHW) in 2002 (2, 3) and is being deployed in the analysis of the 14th IHW data. After the initial development, we created new programs or modified existing programs in order to handle highly polymorphic

**Table 1** Statistical functions. N/A, not applicable; HWP, Hardy-weinberg proportions

| Functions | Notes | References for function |
|---|---|---|
| Population summary | Sample size ($n$); allele frequencies; number of distinct alleles per locus ($k$) | N/A |
| Genotype summary | Lower triangular matrix with observed and expected genotype counts; significant genotypes highlighted. | N/A |
| $\chi^2$ test of HWP | $\chi^2$ statistic; degrees of freedom; associated $P$-value for several classes of genotypes (e.g. all heterozygotes, all homozygotes, common genotypes) | (34) |
| Exact test of HWP | Three modules: Markov chain Monte-Carlo version based on original Guo and Thompson code (4); a modified Monte-Carlo version (no Markov chain); a module that calls Arlequin (35). | (4) |
| Ewens–Watterson test of neutrality | Observed and expected homozygosity ($F$), its expected value, variance, and normalized deviate ($F_{nd}$) under neutrality Slatkin's (5, 6) implementation | (7, 8, 36) |
| Haplotype frequency estimation | Haplotypes frequencies estimated using the expectation-maximization algorithm. | (37, 38) |
| Linkage disequilibrium (LD) measures | Significance tested by the permutation distribution of the likelihood-ratio statistic; overall measures of multiallelic LD: $D'$ and $W_n$ | (39) $D'$ (40); $W_n$ (41) |

data, large number of populations, and typing at various levels of resolution. These programs are incorporated as modules in the latest version of PyPop.

## Materials and methods

### Overview

PyPop was designed to supplement and extend existing population genetic software and to incorporate such software as modules rather than reimplement them from scratch. Therefore, where possible, we based our implementations of population genetic tests on existing, well-tested, open-source code such as Guo and Thompson's (4) 'exact' test for HWP and Slatkin's (5, 6) implementation of the Ewens–Watterson test. Some of these program modules required some augmentation to process the highly polymorphic data we were analyzing. PyPop has also been designed in a modular object-oriented way to facilitate multiple-access points. The framework can be called from short Python scripts (`pypop` and `popmeta`, two scripts that are currently distributed in PyPop, are described later). A prototype of a graphical user interface is undergoing testing in the development version, and the framework is designed in such a way that writing an interface via a web server or web service should be straightforward.

Regarding the input of data, PyPop accepts tab delimited input files with a separate record for each individual, consisting of the list of alleles for each locus. These files can be generated easily from a spreadsheet. Individual PyPop runs are configured with a simple configuration file that uses the Windows `.ini` format and can be run in both an interactive mode and a batch mode. The batch mode allows multiple runs to be scripted as part of an analysis pipeline. Alternatively, modules can be called directly through custom Python code using the API.

PyPop can also handle allele count data for a population (which consist of separate records with allele name, followed by counts of alleles) rather than individuals with full genotypes. The range of analyses possible for such data, however, is restricted to statistics which only require allele frequency information, such as the Ewens–Watterson test of neutrality (7, 8). PyPop was also designed to handle missing data in a flexible manner that is consistent across all modules. This important aspect of data processing and analysis is often implemented in a cumbersome and sometimes inconsistent manner, requiring reformatting of the data.

To facilitate the reuse of as much existing code as possible, and to interface with other languages such as C, we used the Simple Wrapper Interface Generator (SWIG) (9) to enable access to third party programs as if they were Python modules. We also used other open-source projects such as Numeric Python (10) (providing efficient data structures for holding large arrays of data), libxslt (11) [for parsing Extensible Markup Language (XML) output] and the GNU Scientific Library (GSL) (12), and R (13) (for mathematical functions such as $P$-value calculation and randomization algorithm).

### Preanalysis filters

PyPop allows the user to prefilter the input genotype or allele count data before commencing data analysis. The various data filters have a modular design, so they can be used alone or in combinations. One way the filters can be used is to allow the analysis of data at different levels. For example, the `Sequence` filter has the most broad applicability and enables the translation of allele names into full sequence data. If this filter is enabled, then PyPop will treat each sequence position as if it were another distinct marker.

The customization of prefiltering is performed in the initialization file. In this section we focus on the use of `Sequence` filter and the `AnthonyNolan` filter (which assists in quality control of incoming data), both in the context of analyzing HLA data.

### Comparing intermediate and high-resolution sequences

High-resolution HLA typing identifies alleles with a unique sequence at the amino acid level. In the nomenclature, alleles that are typed at high resolution are represented by four digits. Alleles that are typed at an intermediate resolution are generally represented by two digits and it is not always possible to assign a unique amino acid sequence for a given allele of intermediate resolution. To ensure that data of intermediate (two digit) resolution can be reliably compared with high-resolution (four digit) sequences, a consensus sequence must be generated. This consensus sequence takes into account all the possible amino acid sequences the intermediate resolution could match. For example, in the case of the HLA-A locus, the intermediate resolution '03' allele could match any allele starting with the '03' digits such which include alleles 0301 through to 0314.

The allele-to-sequence lookup converts allele calls into sequence data. This lookup uses the Multiple Sequence Format (MSF) (14) file format which includes the full alignment (including gaps and deletions). Because not all alleles have a unique sequence, we developed an algorithm for generating 'consensus' sequences. First, all alleles in a population were identified and a search for a unique sequence was initiated for each allele. If a unique sequence was found then that allele is completed and that sequence continues through the pipeline. If a unique sequence is not found, and the allele could match several different alleles such as the '03' example, then the sequence for each possible match was generated. The sequences for each possible allele match were aligned and compared, and all common sites identified, and used in the resulting 'consensus' sequence. Although, where the sequence differs for any of the alleles, that site was flagged as 'unknown' data ('*') to reflect the ambiguity in that site, because the original allele could correspond to a number of different sequences. Once consensus sequences for all alleles are generated, all monomorphic sites are dropped and only the polymorphic sites, as separated genetic markers, are passed to the main PyPop pipeline. A schematic of this process for the example of '03' is shown in Figure 1.

We developed two other filter modules, which allow for different ways of grouping data. These filters are `DigitBinning` and `CustomBinning` and allow high-resolution allele data to be conveniently analyzed at a broader level, without the need to make any changes to the original data input files. The `DigitBinning` filter allows all high-resolution (four digit) HLA alleles to be collapsed ('binned') down to serological-level resolution

```
03              (original allele)
0301  YYSVSG    (possible match 1)
0302  ....G.    (possible match 2)
0303  ....G.    (possible match 3)
0304  ......    (possible match 4)
0305  ......    (possible match 5)
0306  ....Y.    (possible match 6)
      YYSV*G    (final consensus)
```

**Figure 1** Generating consensus sequences from human leukocyte antigen (HLA) alleles. Dots ('.') represent nucleotide or amino acid residues that are the same as the top sequence in each case. The sequences are only illustrative and do not represent actual sequence data and only represent a small portion of the average HLA sequence which is about 150 amino acid or 450 nucleotides.

(two digits), so that high-resolution data can be compared with older serological-level data. `CustomBinning` allows the user to define custom-rules to govern which alleles are grouped (binned) together and allow a finer-level of control over the binning process than `DigitBinning`.

### Quality control

The filters also serve an important data quality control function. For example, the AnthonyNolan filter is useful for HLA data. The Anthony Nolan Trust maintains a database of known HLA allele names and can be used to check that alleles specified in an input population genotype file are valid. The filter also contains additional heuristics for determining (and substituting) the most likely match if a non-matching allele is found. This filter uses the MSF format. Data files of the HLA alleles names can be obtained via the ftp site (15) at the immunogenetics (IMGT) database (16).

## Functions and statistical tests

PyPop implements a number of population genetic tests and statistics for both single locus and multiple loci. These are summarized in Table 1. We designed PyPop to handle large sample sizes (e.g. sample sizes of between 1000 and 2000 individuals per population), which can be problematic for methods using either resampling or iterative procedures, or both, such as exact tests, haplotype frequency estimation, and linkage disequilibrium (LD) significance testing.

New statistical methods, implemented since the alpha version, include a modified version of the Guo and Thompson (4) code to estimate the overall *P*-value for deviation from Hardy–Weinberg proportions. This version adds an option to use a plain Monte-Carlo algorithm in addition to the existing Markov chain Monte-Carlo algorithm. The module can now handle a larger number of alleles and will soon allow estimation of *P*-values for each

genotype in the population under consideration. A feature of PyPop that has proved very useful in the analysis of the IHW data is the availability of tests for individual heterozygote and genotypes tests for HWP that are not available in other software. These tests identify the specific genotypes that contribute most to any overall significant deviation from HWP. A later release will include a permutation-based test of individual genotypes for HWP.

### Postanalysis processing

The XML output is transformed into a human-readable text file by the default Extensible Stylesheet Language Transformations (XSLT) stylesheet. The XML output can be further transformed using standard tools into many other data formats suitable for machine input. These include PHYLIP (17) or statistical packages, such as R (13), or other human-readable outputs such as HyperText Markup Language. For example, PyPop provides the popmeta script, which takes a number of output XML files from individual populations and aggregates the results into a set of tab-separated (TSV) files containing the statistics grouped by type (one-locus statistics appear in one file, two-locus statistics in another file). These files can be directly imported into a spreadsheet or statistical software.

The current in-development version will allow multiple population file names to be supplied to the pypop script and the TSV files to be generated without requiring a separate run of the popmeta script. Another advantage of storing the output in XML is that it allows the final viewable output format to be redesigned at will (for example by user-supplied XSLT stylesheets), without requiring the analyses themselves to be rerun. Lastly, the XML output allows PyPop itself to be embedded as part of a larger pipeline, thus avoiding the inherent problem of reparsing free-text output for input to another step in the process, a limitation of some existing packages.

## Results

In the context of the 14th IHW analyses (18, 19), PyPop has allowed us to explore population genetic statistics at several levels (within and between populations, within and between geographic regions). For example, it is relatively simple to search for the presence or absence of particular alleles or haplotypes across ethnic groups or geographic regions once the data has been aggregated into a single spreadsheet. Also, the availability of several measures of overall LD (e.g. $W_n$, a multiallelic extension of the correlation measure; and $D'$ the standardized likelihood ratio test statistic) has allowed us to investigate how these measures differ in the information they convey about LD in the highly polymorphic HLA data. As a final example, it has been informative to look at the degree of LD and deviation from HWP for populations with different levels of heterozygosity and admixture.

Although developed originally for the HLA system, PyPop is general enough that it can be used on any genotype data. As of the time of writing 13 peer reviewed publications have cited PyPop, some analyzing data for non-HLA genetic systems. Research papers using PyPop for analysis of HLA data include collaborations among members of our research group with research groups in Ireland (20), Washington, DC (21), and Seattle (22). These collaborations have addressed issues of natural selection on HLA genes, population differentiation, and issues in unrelated donor hematopoietic stem cell transplant, respectively.

Nine of the 13 publications were from independent research teams which used PyPop to analyze data in both HLA and non-HLA genetic systems. HLA genetic data analyzed included class I haplotypes in a Han Chinese population (23) and HLA associations with age-related macular degeneration (24, 25) and multiple sclerosis (26). Non-HLA genetic systems included rheumatoid arthritis (27), interferon regulatory factor-1 (28), orofacial clefts (29), cytokine polymorphisms (30), and mite allergen sensitization (31). These papers show the usefulness of PyPop to address scientific questions in a variety of genetic systems. PyPop has also been cited in a review of population genomics (32) and a textbook chapter (33).

## Discussion

PyPop is a software pipeline that is particularly well positioned to handle large-scale highly polymorphic datasets typical in genomics studies (32). PyPop facilitates the analysis of cross-population data and comparisons of various population genetic statistics. It has been designed to synthesize and enhance the tools that are currently available rather than supplant them, while at the same time adding new methodological options. The modularity and ability to prototype and add new modules make this framework an efficient tool for integrating the rapidly growing array of population genetic analysis methods.

## Availability and requirements

**Project name**: PyPop

**Project home page**: http://www.pypop.org/

**Operating system**: Unix-based systems including Linux and MacOS X; Windows (binaries for Linux and Windows available)

**Programming language**: Python, (Numeric Python module required)

**Other requirements**: If building from source: libxslt, SWIG, GSL

**License**: GNU General Public License (GPL)

**Any restrictions to use by nonacademics**: None

## Authors' contributions

A.L. and M.N. conceived and designed the software. A.L. implemented and tested the software. M.N. implemented the initial Hardy–Weinberg module and conducted initial analyses. R.S. implemented the haplotype and LD module and O.S. implemented the sequence module. A.L. drafted the manuscript. A.L. and R.S. wrote the manuscript. G.T. contributed to its conception and testing and provided project coordination. All authors read and approved the final manuscript.

### Acknowledgments

Thanks to Diogo Meyer for contributions and suggestions during the early stages of the development of the program. Thanks to Sun-Wei Guo for use of the Guo and Thompson code (now under the GPL) and to Montgomery Slatkin for use of the Ewens–Watterson code. Thanks to Jessica Garb and Peg Boyle for helpful feedback on the manuscript and to and Mark Grote, Leslie Louie, Steven J. Mack, Steven G. E. Marsh, Kristie A. Mather, and Hazael Maldonado Torres for suggestions, bug reports, and testing of the software. This work has benefited from the support of US National Institutes of Health grant AI49213 (13th IHW) and US Department of Energy grant DE-FG02-00ER45828.

### Conflict of Interest Statement

All authors have declared no conflicts of interests.

### References

1. Lancaster A, Nelson MP, Meyer D, Single RM, Thomson G. PyPop: a software framework for population genomics: analyzing large-scale multi-locus genotype data. *Pac Symp Biocomput* 2003: **8**: 514–25.
2. Lancaster AK, Nelson MP, Meyer D, Single RM, Thomson G. Software framework for Biostatistics Core. In: Hansen J, ed. *Immunobiology of the Human MHC. Proceedings of the 13th International Histocompatibility Workshop and Conference, Vol. 1. Seattle, Washington: IHWG Press*, *2007*.
3. Single RM, Meyer D, Thomson G. Statistical methods for analysis of population genetic data. In: Hansen J, ed. *Immunobiology of the Human MHC. Proceedings of the 13th International Histocompatibility Workshop and Conference, Vol. 1. Seattle, Washington: IHWG Press*, *2007*.
4. Guo S, Thompson E. Performing the exact test of Hardy-Weinberg proportion for multiple alleles. *Biometrics* 1992: **48**: 361–72.
5. Slatkin M. An exact test for neutrality based on the Ewens sampling distribution. *Genet Res* 1994: **64**: 71–4.
6. Slatkin M. A correction to the exact test based on the Ewens sampling distribution. *Genet Res* 1996: **68**: 259–60.
7. Ewens W. The sampling theory of selectively neutral alleles. *Theor Popul Biol* 1972: **3**: 87–112.
8. Watterson G. The homozygosity test of neutrality. *Genetics* 1978: **88**: 405–417.
9. Beazley DM. SWIG: an easy to use tool for integrating scripting languages with C and C++. 4th Annu Tcl/Tk Workshop '96 1996: 129–139.
10. Dubois PF, Hinsen K, Hugunin J. Numerical python. *Comput Phys* 1996: **10**: 262–267.
11. libxslt: The XSLT C library for Gnome. URL http://xmlsoft.org/XSLT/.
12. Galassi M, Davies J, Theiler J et al. *GNU Scientific Library Reference Manual, 2nd edn. Bristol: Network Theory Ltd, 2005*. ISBN 0954161734.
13. Ihaka R, Gentleman R. R: a language for data analysis and graphics. *J Comput Graph Stat* 1996: **5**: 299–314.
14. Multiple Sequence Format (MSF). http://www.ebi.ac.uk/imgt/hla/download.html.
15. HLA data files. ftp://ftp.ebi.ac.uk/pub/databases/imgt/mhc/hla/.
16. Robinson J, Waller M, Parham P et al. IMGT/HLA and IMGT/MHC: sequence databases for the study of the major histocompatibility complex. *Nucleic Acids Res* 2003: **31**: 311–4.
17. Felsenstein J PHYLIP (Phylogeny Inference Package) version 3.6. Distributed by the author. Seattle: Department of Genome Sciences, University of Washington, 2004. http://evolution.genetics.washington.edu/phylip/.
18. Single RM, Meyer D, Mack SJ et al. Single locus polymorphism of classical HLA genes. In: Hansen J, ed. *Immunobiology of the Human MHC. Proceedings of the 13th International Histocompatibility Workshop and Conference, Vol. 1. Seattle, Washington: IHWG Press*, *2007*.
19. Meyer D, Single RM, Mack SJ et al. Haplotype frequencies and linkage disequilibrium among classical HLA genes. In: Hansen J, ed. *Immunobiology of the Human MHC. Proceedings of the 13th International Histocompatibility Workshop and Conference, Vol. 1. Seattle, Washington: IHWG Press*, *2007*.
20. Williams F, Meenagh A, Single R et al. High resolution HLA-DRB1 identification of a Caucasian population. *Hum Immunol* 2004: **65**: 66–77.
21. Cao K, Moormann A, Lyke K et al. Differentiation between African populations is evidenced by the diversity of alleles and haplotypes of HLA class I loci. *Tissue Antigens* 2004: **63**: 293–325.
22. Malkki M, Single R, Carrington M, Thomson G, Petersdorf E. MHC microsatellite diversity and linkage disequilibrium among common HLA-A, HLA-B, DRB1 haplotypes: implications for unrelated donor hematopoietic transplantation and disease association studies. *Tissue Antigens* 2005: **66**: 114–24.
23. Hong W, Fu Y, Chen S, Wang F, Ren X, Xu A. Distributions of HLA class I alleles and haplotypes in Northern Han Chinese. *Tissue Antigens* 2005: **66**: 297–304.
24. Goverdhan S, Howell M, Mullins R et al. Association of HLA class I and class II polymorphisms with age-related macular degeneration. *Invest Ophthalmol Vis Sci* 2005: **46**: 1726–34.
25. Goverdhan SV, Lotery AJ, Howell WM. HLA and eye disease: a synopsis. *Int J Immunogenet* 2005: **32**: 333–42.

26. Traherne JA, Barcellos LF, Sawcer SJ et al. Association of the truncating splice site mutation in BTNL2 with multiple sclerosis is secondary to HLA-DRB1*15. *Hum Mol Genet* 2006: **15**: 155–61.
27. Lei C, Dongqing Z, Yeqing S et al. Association of the CTLA-4 gene with rheumatoid arthritis in Chinese Han population. *Eur J Hum Genet* 2005: **13**: 823–8.
28. Ji H, Ball T, Kimani J, Plummer F. Novel interferon regulatory factor-1 polymorphisms in a Kenyan population revealed by complete gene sequencing. *J Hum Genet* 2004: **49**: 528–35.
29. Lammer E, Shaw G, Iovannisci D, Finnell R. Periconceptional multivitamin intake during early pregnancy, genetic variation of acetyl-N-transferase 1 (NAT1), and risk for orofacial clefts. *Birth Defects Res A Clin Mol Teratol* 2004: **70**: 846–52.
30. Trajkov D, Arsov T, Petlichkovski A, Strezova A, Efinska-Mladenovska O, Spiroski M. Cytokine gene polymorphisms in population of ethnic Macedonians. *Croat Med J* 2005: **46**: 685–92.
31. Tan CY, Chen YL, Wu LS, Liu CF, Chang WT, Wang JY. Association of CD14 promoter polymorphisms and soluble CD14 levels in mite allergen sensitization of children in Taiwan. *J Hum Genet* 2006: **51**: 59–67.
32. Luikart G, England P, Tallmon D, Jordan S, Taberlet P. The power and promise of population genomics: from genotyping to genome typing. *Nat Rev Genet* 2003: **4**: 981–94.
33. Chikhi L, Bruford M. Mammalian population genetics and genomics. In: *Mammalian Genomics. Wallingford: CABI Publishing*, *2004*, *539–584*.
34. Emigh T. A comparison of tests for Hardy-Weinberg equilibrium. *Biometrics* 1980: **36**: 627–642.
35. Schneider S, Kueffer J, Roessli D, Excoffier L Arlequin: a software for population genetics data analysis. Ver 2.000, Genetics and Biometry Lab, Department of Anthropology, University of Geneva, 2000. http://lgb.unige.ch/arlequin/.
36. Salamon H, Klitz W, Easteal S, Gao X, Erlich HA, Fernandez-Viña M, Trachtenberg EA, McWeeney SK, Nelson MP, Thomson G. Evolution of HLA class II molecules: allelic and amino acid site variability across populations. *Genetics* 1999: **152**: 393–400.
37. Dempster A, Laird N, Rubin D. Maximum likelihood estimation from incomplete data using the EM algorithm. *J Royal Stat Soc* 1977: **39**: 1–38.
38. Excoffier L, Slatkin M. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol Biol Evol* 1995: **12**: 921–927.
39. Slatkin M, Excoffier L. Testing for linkage disequilibrium in genotypic data using the expectation-maximization algorithm. *Heredity* 1996: **76**: 377–383.
40. Hedrick P. Gametic disequilibrium measures: proceed with caution. *Genetics* 1987: **117**: 331–41.
41. Cramer H. *Mathematical Models of Statistics. Princeton, New Jersey*: *Princeton University Press*, *1946*.