slidingWindows_v1.10.py

Author: David Field
14/05/2018
david.field@univie.ac.at

What does this script do?

This script calculates various diversity (π_w) and differentiation statistics ($e.g. F_{ST}, \pi_b / D_{xy}$) for individual or pooled sequence data for two or more populations. Two types of output are generated: (i) individual site statistics, (ii) sliding window averages of the same statistics (with adjustable overlaps). Different data manipulations with regards to data quality can be set by the user including, minimum and maximum depth and the minimum number of populations with required depth range for site inclusion. The script can also work with individual based data, as long as this is in the required format (currently *.sync). Other data file formats will be supported in future versions.

How to install

Simply place the script and all required input files in a common directory. Python modules required include: *re*, *sys*, *getopt*, *itertools*, *os*, *math.*

Ideally place this in a directory included in your PATH. See online how to edit your PATH

Some Python modules that may not come with base installation are required. If some modules are missing, download and install the required modules with, for example:

```
> sudo apt-get install python-numpy python-scipy
```

Input files

- **Genomic data**: in this version 4, the format is fixed on *.sync files as used for pooled data in the program Popoolation2 (Kofler et al., 2010).
- **Scaffold data**: a file listing the set of scaffolds or chromosomes to analyse. This will assume *.sync files exist with the same names.
- **Population details**: a file listing population names and sample sizes.

How to run the program

Assuming you have the script and the required input files in your PATH directory, you can run the program as follows:

```
> python SlidingWindows_v1.10.py scaffoldList_rosAssembly.txt
popDetails_10pool_edited.txt V2_RosAss 15 200 2 2 10000 0 1 0 0 1
```

where the system arguments for the script include (corresponding values in example):

1. list of scaffolds to analyse (scaffold_RosSample.txt).
2. file with population names and sample size (popDetails_RosSample.txt).
3. output file name for sliding window statistics (LG6_RosSample.txt).
4. the minimum depth to include a site (15).
5. the maximum depth to include a site (200).
6. minimum copies for an allele call, i.e. > 1, no singletons (2).
7. minimum number of populations (or pools) with the minimum depth to include a site (2).
8. window size in base pairs (10000).
9. overlap size between windows (5000).
10. keep site specific file, whether to keep files for each scaffold/chromosome (1 = Y, 0 = N).
11. window analysis only (1 = Y, 0 = N). If repeated window analysis on same sites file.
12. site analysis only (1=Y, 0 = N)
13. keep all positions (1 = Y, 0 = N). For more compact sites output file (polymorphic sites only) then choose 0, for all sites choose 1.

Some important details:

- min depth: if running on individual based data, make minimum depth the size of the smallest number of haploid genomes (i.e. 4x2 = 8),
- min copies: if running on individual based data, make minimum copies = 1 (i.e. allow for singletons),
- input: scaffold input list must have line breaks saved as Unix (LF). Use text wrangler or similar text editor,
- window analysis only: only works if a site specific file(s) already created and present in the same folder,
- min scaffold size: skips over any scaffold < 15kB. Ensures at least two windows minimum.

Description of output for site output file

- **scaffold**: genome scaffold/chromosome
- **position**: position on scaffold/chromosome
- **LG**: linkage group
- **cM**: physical distance along linkage group (centri Morgans)
- **ref**: reference genome allele
- **bases_t**: variants detected at the site
- **alleles_num**: number of allelic variants detected
- **1or2bases**: whether 1-2 bases detected (1) or more than two (0)
- **minDepth_pools**: how many populations/pools passed filters
- **minDepth_pass**: whether the minimum depth threshold was achieved at this site for the minimum number of populations (1=yes, 0=no).

Single population statistics

- **p_1,..., p_i**: allele frequencies in the *i*th population/pool.
- **poly_pool_1,..., poly_pool_i**: the average proportion of polymorphic sites in the *i*th population/pool.
- **pi_1, ..., pi_i**: $\pi_w$for individual based population data in the *i*th population.

  Here, pi_i = $\pi_w = 1 - \sum_{i=1}^{n} p_i^2 = 2(p_j)(1 - p_j)$.
- **piAdj_1,..., piAdj_i**: $\pi_w$for pooled data in the *i*th population.

  Here, adjustment for binomial sampling where piAdj_i = π_w[(M-1)/(M-2b+1)], where M = depth, b = minimum read count for allele call.
- **dpthAdj_1,..., dpthAdj_i:** mean depth within window for each population I, adjusted for binomial sampling
- **dpthPass_1,..., dpthPass_i:** what proportion of the window is covered in population i

Population pair-wise statistics

- **piBar_1_2,..., piBar_i_j**: average $\bar{\pi}_w$ across each pairwise population/pool comparison, (pi_1+pi_2)/2.
- **piBarAdj_1_2,..., piBarAdj_i_j**: average $\bar{\pi}_w$(as above) but adjusted for binomial sampling in pooled data. Where, piBarAdj_i_j = piBar_i_j[(M-1)/(M-2b+1)]
- **pi_T**: total for each pairwise pool comparisons, e.g piT_1_2....**pi total = 2 x pbar x qbar** (i.e. 2 x mean allele p frequency x mean q allele frequency)

pi total adjusted (binomial errors) for each pairwise pool comparisons. e.g. piT_adj_1_2.

- **pi_TAdj = (2 x pbar x qbar) x [(M-1)/(M-2b+1)]**

- **Dxy_raw**: dXY_raw_1_2..... **Dxy = (p1*q2 + q1*p2)**. Not recommended

- **Dxy_ fromFstadj**: but instead calculated backwards from Fst and piBar. e.g. dXYfromFst_1_2..... **Dxy = piBar*(1-Fst/1+Fst)**
- pairwise relative divergence Fst, calculated from each site from pi values then averaged for the window (note this is not what Popoolation does). e.g. FstPi_1_2... **Fst = (piTotal - pibar)/piT**

- **FstPiAdj**: pairwise relative divergence Fst but with adjusted values of Pi, calculated from each site from pi values then averaged for the window (note this is NOT what Popoolation does). e.g. FstPiAdj_1_2... **Fst = (piTotal_adj - pibarAdj)/piTotal_adj**

- **Fst** pairwise relative divergence Fst back calculated from raw Dxy values. e.g. FstfromDxy_1_2.... **Fst = (Dxy - piBar)/(Dxy + piBar).** Not recommededn as this comes from site averages.

- **FstfromMeanPi**: pairwise relative divergence Fst from average Pi values from the window, FstfromMeanPi_1_2.. **Fst = (piTotal_window - pibar_window)/piTotal_window**. (this is what we use for individual sequence data)

- **FstfromMeanPiAdj_1_2**: pairwise relative divergence from average adjusted PiAdj values from the window, **Fst = (piTotal_adj_window - pibarAdj_window)/piTotal_adj_window**. (this IS what we use for the pooled data)

- **DxyfromFst:** pairwise divergence Dxy, but instead calculated from backwards from Fst values above. e.g. dXYfromMeanFst_1_2... **DxyfromFst = piBar x [(1 + Fst)/(1- Fst)].** (this is what we could use for individual data)

- **DxyfromFstAdj_1_2,…,:** pairwise divergence Dxy, but instead calculated backwards from Fst adjusted values above, dXYfromMeanAdjFst_1_2.... **DxyfromFstAdj = piBarAdj x [(1 + FstAdj)/(1- FstAdj)].** (this is what we could use for pooled data)

- last two are Da values, see Cruickshank and Hahn 2014 for details