**README**

Thank you for checking out the **foxy\_qtl\_pipeline**. This is a series of computer scripts that performs quantitative trait loci (QTL) analysis of biparental genetic mapping populations. Essentially, it is a series of **R/qtl** [1] scripts called by a python program generally called on a Linux based computer server.

It can also be run on a MacBook or other Linux desktop but the analysis of many traits in parallel (server mode) is not recommended.  
  
This pipeline can very easily be adapted to perform similar analysis on RIL populations in other organisms.

Currently it automates several types of analysis

1) A genome scan with a single QTL model (significance: p-value < 0.05, based upon 1000 permutations).

2) A two-dimensional genome scan using a 2-QTL model (significance: p-value < 0.05, based upon 200 permutations).

3) A stepwise model selection (significance based upon penalized LOD score, estimated in either the first 2 analysis methods).

4) Single QTL model genome scan and stepwise model analysis of function valued traits (temporal or spatial) [2].

Dependencies

This software pipeline uses several python and R libraries that need to be installed.

Python libraries:

csv

glob

multiprocessing

numpy

optparse

os.path

re

subprocess

sys

R libraries:

funqtl

ggplot2

lattice

lme4

qtl

Assumptions

It is assumed that the phenotypic trait data input is appropriately blocked and normally distributed. The user should check the normality assumption using a Shapiro-Wilks test before data is input. If data is not at least approximately normal Box-Cox transformation is recommended. If individuals within treatment blocks are replicated, the program will perform partitioning of variance and submit the mean of trait values for each genotype values nested within treatment.   
  
Users are recommended to perform BLUP/BLUE modeling or other types of regression/smoothing before analysis using this QTL pipeline.

How to launch an analysis

1) Download all relevant files from GitHub:

<https://github.com/maxjfeldman>

You’ll want to run this program on a large computer server. Our current implementation requires 1-2 GB of RAM per trait at time of analysis. This can be an issue if you want to analyze all traits in parallel on a small computer.

The efficiency could be improved rather easily but I’ve never had to analyze more than 50-100 traits in parallel and the computing infrastructure at my current institution supports this.

2) Format your input data.

Current format this is a .csv file that contains a few fields that aren't necessary for the analysis. Please see file named: “qtl\_data\_format\_example.csv”.

Column headers:

Obs – This is just an number entry number sometimes recorded by an instrument but can be as simple as row number

experiment – This is usually an identifier that specifies a 2 character description of experiment and the year.

year – What year is the data from?

treatment – Is there a contrast being performed (wet v. dry | dense v. sparse)? If no treatment just specify the same string for each entry (“none” for example).

plot – Setaria grow outs can be summarized by overall plot. If you experiment does not contain multiple plot you can just specify the plot using the same string for each entry (“none” for example)

subplot\_id – This is a string that associates the individual plant location within the plot (can also specify “none”).

id – This is the name of the RIL. The nomenclature you use must match the names in the genetic map. We generally use the format “RIL\_001”.

sampling – This is a subcategory that can be used to distinguish between identical plants sampled at different time points (usually, this is imputed as “none”).

The remaining columns contain phenotypic values (height\_25, height\_46, height\_67).  
  
\*\* An important detail is that trait names cannot contain spaces or periods. Periods are currently used as a field delimiter to distinguish between the same trait measured in different treatments.

3) Call the program.

Change into the directory where you have downloaded the script and use the following function call:

[user@computer~]$ python foxy\_qtl\_pipeline.py –i input\_file.csv -o name\_of\_output\_directory -c [y|n] -m name\_of\_genetic\_map -s [y|n] -t riself

There are several arguments that need to be specified:

i – This is the name of the input file (see qtl\_data\_format\_example.csv)

o – This the name of the directory where you will store the results

c – Are you doing a comparison between treatments (wet v. dry) or not? “y” indicates that the treatment field in the input file contains 2 different levels

m – File name of your genetic map formatted for R/qtl csvs input.

s – Are you running this analysis on a server? If you select “y” traits will be analyzed in parallel. If you select “n” they will be run consecutively.

t – Type of mapping population. Currently I am working with a F7 RIL mapping population so I use ‘riself’.

Other arguments that will be added in the future:

d – distribution/model type (is this a normal, binary, or 2-part or non-parametric)

q – Method to use for QTL analysis (currently all is done using Haley-Knott regression)

f – Is this a functional trait (time-series)? [y|n] \*\*Currently we have a separate set of scripts to do this type of analysis. See below.

Example:

As an example we will perform an analysis of plant height data from the Setaria A10 X B100 recombinant inbred line population. This population was grown at two planting densities (dense = 5 cm planting density, sparse = 25 cm planting density) in the field at the University of Illinois. Plant height was measured using handheld barcode scanners at 3 time points representative of the plant life cycle (See Feldman et al., 2016. in prep).

First, examine the format/structure of the data by opening the file named ‘qtl\_data\_format\_example.csv’. You will see that there are three traits measured ‘height\_25’, ‘height\_46’ and ‘height\_67’. Do you understand what the columns are specifying? Do you see any period characters (‘.’) within the trait names?

If you examine the entries in the treatment column how many unique entries do you see?

Because there is more than 1 (‘dense’ and ‘sparse’) this is a comparison trait.

Also included are two genetic map files named “GBS\_map\_A10xB100\_v0.96.csv” and “2013setariamapJGI.csv”. The first is a genetic map that contains some genotype calls for all or most of the individuals within the RIL population as described in (Feldman et al., 2016. in prep.) whereas the latter is directly from Mauro-Herrera et al., 2013.

Lets run the analysis.

Open your terminal window and move into the foxy\_qtl\_pipeline directory. If you list the contents of this directory you should see the following files.

Type the following command (Don’t copy and paste):

[user@computer~]$ python foxy\_qtl\_pipeline.py –i qtl\_data\_format\_example.csv –o example\_output -c y -m GBS\_map\_A10xB100\_v0.96.csv -s y -t riself

This command tells the computer to launch the python program named: foxy\_qtl\_pipeline.py  
  
You’ve input the file named ‘qtl\_data\_format\_example.csv’  
The results will be in a directory named ‘example\_output’

You have indicated you’d like to do a comparison [-c y] (treatment column contains 2 factors: ‘dense’ & ‘sparse’)

You have told the program to use a genetic map named ‘GBS\_mapA10xB100\_v0.96.csv

By setting the -s y (server) flag yes you have indicated you are running this job on a computer cluster and it will run all the traits in parallel

You have told the program you are analyzing a RIL population by setting the –t argument to ‘riself’