**Dependencies:**

python2.7

Biopython

http://biopython.org/DIST/docs/install/Installation.html

Egglib3.0

http://mycor.nancy.inra.fr/egglib/

snpeff requires java 7

install ubuntu: sudo apt-get install openjdk-7-jre

**ggplot2 in R**

**popgenome in R**

**https://cran.r-project.org/web/packages/PopGenome/index.html**

**python matplotlib**

**sudo apt-get install python-matplotlib**

**python scipy and numpy**

**sudo apt-get install python-numpy python-scipy**

**Format conversion scripts:**

**in ./scripts/**

* **convert vcf file to tab-delimited format:**

script: vcf\_to\_tab.py

input: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.vcf

usage: python wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.vcf Magnaporthe\_oryzae\_wheat\_blast.vcf

output: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.tab

* **convert tab-delimited file to structure format, keeping one SNP per gene and a random 10% of all genes having a SNP:**

script: tab\_to\_struct\_one\_SNP\_per\_gene.py

input: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.tab and ./refgenome\_wheatblast\_BR32/BR32.gff

usage:python ./tab\_to\_struct\_one\_SNP\_per\_gene.py ./refgenome\_wheatblast\_BR32/BR32.gff wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.tab

output: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30\_one\_SNP\_per\_gene.str

* **convert tab-delimited to fasta format, excluding missing data:**

script: tab\_to\_fasta.py

input: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.tab

usage:python ./tab\_to\_fasta.py wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.tab

output: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30.fasta

**Annotate SNPs using SNPEff:**

**in ./snpeff/**

**program:**

java executables can be downloaded from

https://sourceforge.net/projects/snpeff/files/snpEff\_latest\_core.zip/download

**files:**

- input VCF: wheat\_blast\_brazil.with\_inv.filters.vcf

- databases: /wheat\_blast/ and /Magnaporthe\_oryzae

- config file: snpEff.config

**install:**

it’s java, so there’s nothing to do

however, the database for our reference genomes wheat\_blast and Magnaporthe\_oryzae need to be created and included manually in the SNPEff directory:

*copy folders /Magnaporthe\_oryzae/ /wheat\_blast/ into folder /data/ within the SNPEff directory*

*copy config file snpEff.config in the SNPEff program directory*

**usage:**

java -jar path\_to\_snpeff\_jar\_file/snpEff.jar -upDownStreamLen 500 -v wheat\_blast path\_to\_vcf\_file/wheat\_blast\_brazil.with\_inv.filters.vcf > path\_to\_output\_SNPEff/wheat\_blast\_brazil.with\_inv.filters.ANN.vcf

**output:**

wheat\_blast\_brazil.with\_inv.filters.ANN.vcf

**Analyse population subdivision using Structure program:**

**in ./structure/**

**program:**

http://web.stanford.edu/group/pritchardlab/software/structure\_v.2.3.1.html

**files:**

input: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30\_one\_SNP\_per\_gene.str

param files: mainparams, extraparams

**usage:**

for k in {1..5}; do for i in {1..10}; do ./structure -K $k -o "wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30\_one\_SNP\_per\_gene\_K"$k"\_rep"$i".txt";done;done;

run takes several hours

**output:**

50 files: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30\_one\_SNP\_per\_gene\_K$k\_rep$i.txt

**Plot population structure:**

**in ./structure/**

**plot membership proportions for the ten repeat runs of each $K value:**

script: plot\_barplot\_structure\_singleK.py

input: in ./structure\_results/ folder

usage: python ./plot\_barplot\_structure\_singleK.py structure\_output\_K $K

output: structure\_output\_K$K\_barplot\_10repeats.pdf

**plot membership proportions for given list of Structure outputs (one repeat per K value):**

script: plot\_barplot\_structure.py

input: list of files in list\_plot.txt; files in ./structure\_results/ folder

usage: python ./plot\_barplot\_structure.py structure\_output\_K

output: structure\_barplots.pdf

**Analyse population subdivision using Splitstree program:**

**in ./splitstree/**

**program:**

http://www.splitstree.org/

**files:**

input: wheat\_blast\_brazil\_bangladesh.snp-only.filters\_maxmissing30\_wo\_Ns.fasta

**usage:**

double click on splitstree program

open datafile, and it runs automatically (loading input file could be slow though)

**Test of the standard neutral model using McDonald-Kreitman test:**

**in ./mktest/**

**prepare fasta sequences for all supercontigs:**

script: vcf\_to\_fasta\_bp\_resolution.py

input: wheat\_blast\_brazil.with\_inv.filters\_subset.vcf

usage:python ./vcf\_to\_fasta\_bp\_resolution.py wheat\_blast\_brazil.with\_inv.filters\_subset.vcf

output: wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold\*.fasta

**prepare fasta sequences for all supercontigs, with outgroup:**

script: vcf\_to\_fasta\_bp\_resolution.py

input: wheat\_blast\_brazil\_outgroup.with\_inv.filters\_subset.vcf

usage:python ./vcf\_to\_fasta\_bp\_resolution.py wheat\_blast\_brazil\_outgroup.with\_inv.filters\_subset.vcf

output: wheat\_blast\_brazil\_outgroup.with\_inv.filters\_BR32\_scaffold\*.fasta

**prepare pseudo fasta alignments for all genes:**

script: build\_fasta\_alignments-biopython.py

input: wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold\*.fasta and ../refgenome\_wheatblast\_BR32/BR32.gff

usage: build\_fasta\_alignments-biopython.py wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold

output: fasta files in directory CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold

**prepare pseudo fasta alignments for all genes, including outgroup:**

script: build\_fasta\_alignments-biopython.py

input: wheat\_blast\_brazil\_outgroup.with\_inv.filters\_BR32\_scaffold\*.fasta and ../refgenome\_wheatblast\_BR32/BR32.gff

usage: build\_fasta\_alignments-biopython.py wheat\_blast\_brazil\_outgroup.with\_inv.filters\_BR32\_scaffold

output: fasta files in directory CDSs\_wheat\_blast\_brazil\_outgroup.with\_inv.filters\_BR32\_scaffold

**compute summary stats for polymorphism using egglib:**

script: summary\_stats\_fasta\_egglib\_syn\_nonsyn.py

input: data in ./wheat\_blast\_brazil\_fasta/CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold/

usage: python ./summary\_stats\_fasta\_egglib\_syn\_nonsyn.py ./wheat\_blast\_brazil\_fasta/CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold/

output: summary\_stats\_syn\_nsyn.txt

**compute divergence stats using egglib:**

script: KaKs\_one\_ingroup\_vs\_outgroup.py

input: data in ./wheat\_blast\_brazil\_outgroup\_fasta/CDSs\_wheat\_blast\_brazil\_outgroup.with\_inv.filters\_BR32\_scaffold/

usage: python ./KaKs\_one\_ingroup\_vs\_outgroup.py ./wheat\_blast\_brazil\_outgroup\_fasta/CDSs\_wheat\_blast\_brazil\_outgroup.with\_inv.filters\_BR32\_scaffold/

output: Ka\_Ks\_one\_ingroup\_vs\_one\_outgroup.txt

**compute MK tables:**

script: compute\_MKtables.py

input: summary\_stats\_syn\_nsyn.txt and Ka\_Ks\_one\_ingroup\_vs\_one\_outgroup.txt

usage: python ./compute\_MKtables.py

output: stdout

**Tests for balancing selection:**

**in ./balancing\_selection/**

**compute summary stats for polymorphism using egglib:**

input: data in ../mktest/wheat\_blast\_brazil\_fasta/CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold/

script: summary\_stats\_fasta\_egglib.py

usage: python ./summary\_stats\_fasta\_egglib.py ../mktest/wheat\_blast\_brazil\_fasta/CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold/

output: summary\_stats.txt

**compute number of protein variant based on nonsyn substitutions:**

input: data in ../mktest/wheat\_blast\_brazil\_fasta/CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold/

script: compute\_number\_prot\_var.py

usage:python ./compute\_number\_prot\_var\_number\_premature\_stops.py ../mktest/wheat\_blast\_brazil\_fasta/CDSs\_wheat\_blast\_brazil.with\_inv.filters\_BR32\_scaffold/

output: stdout