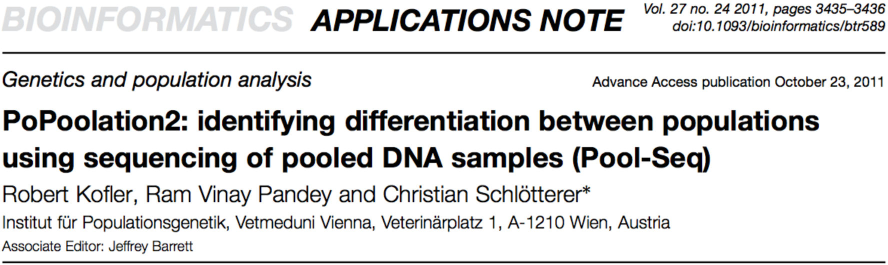
Population genomics with sequenced pools (Pool-Seq) - 2

Dr. Robert Kofler

February 14, 2014

PoPoolation 2



BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 24 2011, pages 3435–3436
doi:10.1093/bioinformatics/btr589

Genetics and population analysis Advance Access publication October 23, 2011

PoPoolation2: identifying differentiation between populations using sequencing of pooled DNA samples (Pool-Seq)

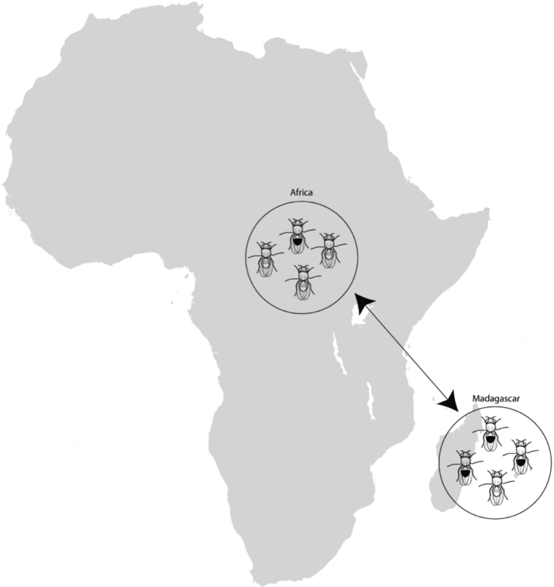
Robert Kofler, Ram Vinay Pandey and Christian Schlötterer*
Institut für Populationsgenetik, Vetmeduni Vienna, Veterinärplatz 1, A-1210 Wien, Austria

Associate Editor: Jeffrey Barrett

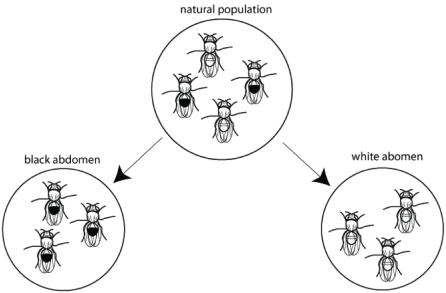
WHAT CAN BE DONE WITH POPOOLATION 2

- measure differentiation between natural populations (identify differentiated genomic regions)
- Pool-GWAS; phenotypically extreme individuals (black vs. white abdomen) are separted into groups which are sequenced as pools. Allele frequency differences between these two groups may allow to identify the causative alleles
- evolve and resequencing studies; populations are allowed to adapt to novel environmental conditions. Loci responsible for adaptation can be identified by comparing the allele frequencies in the populations before and after adaptation.

IDENTIFY LOCAL ADAPTATIONS



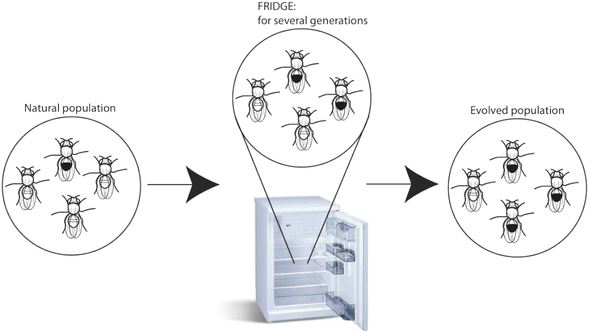
POOL-GWAS



- sequence flies with black and white abdomen separately as pools
- compare allele frequency differences between the pools (e.g.: F_{ST} or cmh-test)

EVOLVE AND RESEQUENCE (E&R)

⇒ our main focus in Vienna (and my favorite topic)



- keep flies at novel environmental conditions for some generations
- sequence the evolved and the not evolved populations as pools
- compare allele frequency differences between the evolved and not evolved populations

Features of PoPoolation2

To address these questions PoPoolation2 allows to compute several test statistics:

- estimate allele frequency differences between the populations
- Fishers exact test to determine significance of allele frequency differences
- pairwise F_{ST}
- the cmh-test (Cochran-Mantel-Haenszel) to identify consistent allele frequency changes between biological replicates (Pool-GWAS or E&R)

PoPoolation2 has been thoroughly validated

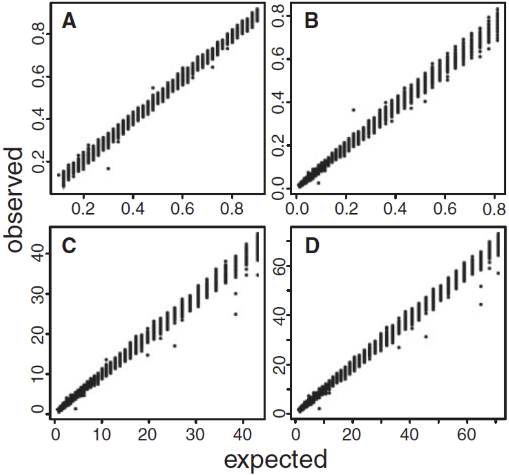


Fig. 1. Expected versus observed values for the tests implemented in PoPoolation2 using 10 000 simulated SNPs. (**A**) allele frequency difference; (**B**) F_{ST}; (**C**) Fisher's exact test [$-\log 10(P\text{-value})$]; (**D**) CMH test [$-\log 10(P\text{-value})$].

THE WORKFLOW..

You should actually use the following steps:

- trimming of reads
- mapping of reads
- removing duplicates
- removing low quality sequences (ambiguous mapping)
- converting to mpileup
- converting to sync-file
- subsampling to uniform coverage
- remove regions flanking indels
- PoPoolation 2

BUT...WE ARE LAZY...

But you already know the pipeline, so it would be tedious to repeat every step. Instead I suggest to use the following shortcut:

- ~~trimming of reads~~
- mapping of reads
- ~~removing duplicates~~
- ~~removing low quality sequences (ambiguous mapping)~~
- converting to mpileup
- converting a pileup to a sync-file
- ~~subsampling to uniform coverage~~
- ~~remove regions flanking to indels~~
- PoPoolation 2

Data from two populations

For this tutorial we have data from two populations. One fastq file from each.

```
cd ~/Desktop/popoolation2
md5sum pop*
> e0beb1e213ddd6cfd7cc1c6b6e13ba7d  pop1.fastq
> ff8c8704abc71b41a5aa1d06300920b3  pop2.fastq
```

lets prepare the reference genome for mapping

```
mkdir analysis
mkdir analysis/wg
cat 2R.chr | awk ’{print $1}’> anaylsis/wg/2R.chr
bwa index analysis/wg/2R.chr
```

MAPPING OF THE DATA

As we are now really lazy we even use a different mapping algorithm (less typing) with default parameters (which should be avoided).

```
bwa mem analysis/wg/2R.chr pop1.fastq > analysis/
    pop1.sam
bwa mem analysis/wg/2R.chr pop2.fastq > analysis/
    pop2.sam
```

Create a mpileup

```
cd analysis
samtools view -Sb pop1.sam | samtools sort - pop1
samtools view -Sb pop2.sam | samtools sort - pop2
# note that samtools is automatically appending the extension *.bam (very annoying)
samtools mpileup -B -Q 0 -f wg/2R.chr pop1.bam pop2.bam > p1-2.mpileup
```

INSPECTING THE MPILEUP

```
2R 991 A 14 .............C AAAAAAAAAAAAAA 14 ..............
    AAAAAAAAAAAAAA
2R 992 A 14 .............. AAAAAAAAAAAAAA 14 ..............
    AAAAAAAAAAAAAA
2R 993 T 14 .............. AAAAAAAAAAAAAA 14 ..............
    AAAAAAAAAAAAAA
2R 994 A 14 .............. AAAAAAAAAAAAAA 14 ..............
    AAAAAAAAAAAAAA
2R 995 A 14 ...........G.. AAAAAAAAAAAAAA 14 ..............
    AAAAAAAAAAAAAA
```

Very similar to the mpileup with one sample (see PoPoolation1 tutorial). In this multi-pileup (mpileup) three additional columns are created for each additional sample. Thus for each sample the following information is provided:

- the coverage, columns $4 + n * 3$
- the bases, columns $5 + n * 3$
- the corresponding base quality, columns $6 + n * 3$

⇒ is there anything weird about these data

CONVERSION TO SYNC-FILE

In order to use PoPoolation2 we have to convert the mpileup to a sync file. This may seem unnecessary to you, but it serves to speed up the analysis, because the time consuming part of the analysis - parsing of the mpileup file - needs just to be performed once.

```
java -jar ~/programs/popoolation2/mpileup2sync.jar --input p1-2.mpileup --output p1-2.sync --fastq-type sanger --min-qual 20 --threads 2
```

⇒ I implemented this step in Java using multi-threading, which speeds it up tremendously (by a factor 76). The old and slower Perl version is however still available.

THE SYNC-FILE

```
2R 26 T 0:14:0:0:0:0 0:14:0:0:0:0
2R 27 G 0:0:0:14:0:0 0:0:0:14:0:0
2R 28 A 14:0:0:0:0:0 14:0:0:0:0:0
2R 29 G 0:0:0:14:0:0 0:0:0:14:0:0
```

- col 1: reference chromosome
- col 2: position
- col 3: reference character
- col 4: allele counts for first population
- col 5: allele counts for second population
- col n: allele counts for n-3 population

⇒ Allele counts are in the form ”A:T:C:G:N:del”
⇒ the sync-file provides a convenient summary of the allele counts of several populations (there is no upper threshold of the population number).
⇒ subsampling to an uniform coverage with the PoPoolation2 pipeline should be done using such a sync-file

CALCULATING THE F_{ST}

To speed up the analysis we calculate the F_{ST} only for a small sample. The F_{ST} will be calculated for SNPs. It could however also be calculated for windows.

```
head -500000 p1-2.sync > small.sync
perl ~/programs/popoolation2/fst-sliding --window-
    size 1 --step-size 1 --suppress-noninformative
    --input small.sync --min-covered-fraction 1.0 --
    min-coverage 4 --max-coverage 120 --min-count 3
    --output fst.txt --pool-size 100
```

OUTPUT OF THE F_{ST} SCRIPT

```
2R 137071 1 1.000 14.0 1:2=0.01333333
2R 138424 1 1.000 23.0 1:2=0.00584795
2R 141783 1 1.000 37.0 1:2=0.00469484
2R 141815 1 1.000 37.0 1:2=0.00353357
```

- col 1: reference chromosome
- col 2: position
- col 3: window size (1 for single SNPs)
- col 4: covered fraction (relevant for minimum covered fraction)
- col 5: average minimum coverage for the window across all populations (the higher the more reliable the estimate)
- col 6: pairwise F_{ST} comparing population 1 with population 2
- col 7: etc for ALL pairwise comparisons of the populations present in the sync file

An example output with multiple populations:

```
2L      68500   360     1.000   62.1    1:2=0.01873725
    1:3=0.02131245  2:3=0.01521177
2L      69500   118     1.000   71.9    1:2=0.00969479
    1:3=0.00116059  2:3=0.00905794
2L      70500   269     1.000   63.6    1:2=0.01955417
    1:3=0.01547995  2:3=0.01300569
```

GET THE 5 TOP DIFFERENTIATED SNPS

With clever usage of UNIX commands it is quite simple to answer a majority of the questions arising during any bioinformatics analysis. A researcher may for example ask: where are my top 10 differentiated SNPs?

```
cat fst.txt | sed ’s/1:2=//’ | sort -k6,6nr |head
```

```
2R 361612 1 1.000 14.0 0.17647059
2R 121409 1 1.000 14.0 0.13131313
2R 213723 1 1.000 14.0 0.13131313
2R 221391 1 1.000 13.0 0.13043478
2R 10172 1 1.000 14.0 0.12000000
...
```

⇒ investing a bit of your time learning UNIX command will pay off very quickly (when I started bioinformatics I wrote a complex C# software that needed compiling for every simple task = enormous wast of time).
⇒ PoPoolation2 does not contain scripts for tasks that can be easily addressed with simple Unix commands. I’m however thinking about creating a wiki-page at the PoPoolation2 wiki where such simple Unix-commands could be collected.

LOAD F_{ST} INTO IGV

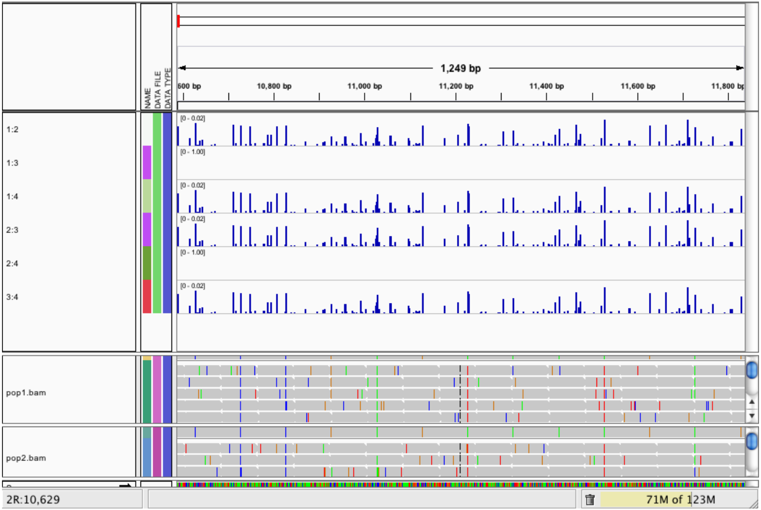
First, convert the F_{ST} file into an '*.igv' file

```
perl ~/programs/popoolation2/export/pwc2igv.pl --input fst.txt --output fst.igv
samtools index pop1.bam
samtools index pop2.bam
java -Xmx2g -jar ~/programs/IGV_2.3.26/igv.jar
```

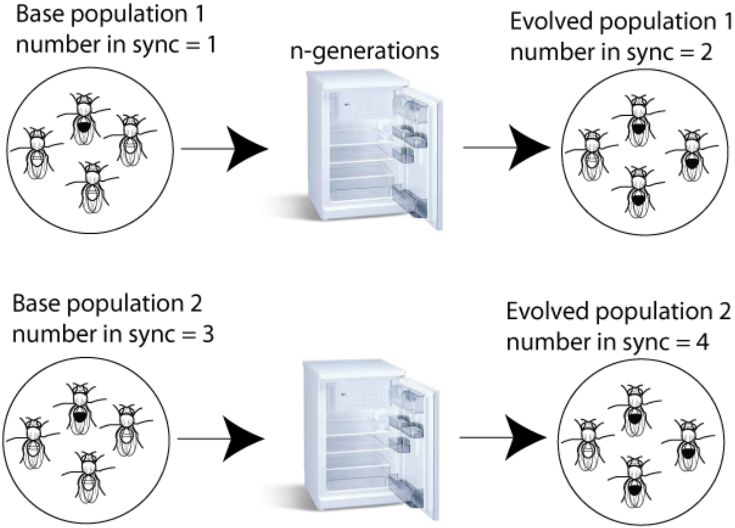
- load populations pop1.bam and pop2.bam
- load fst.igv
- go to chromsomoe 2R and zoom in on position 200.000

⇒ Note that PoPoolation2 as well as IGV can handle multiple pairwise comparisons.

Example of multiple pairwise comparisons



A typical scenario for the CMH-test



MANHATTEN PLOT WITH P-VALUES FROM CMH-TEST

For a well designed E&R study you may expect something like in the following.

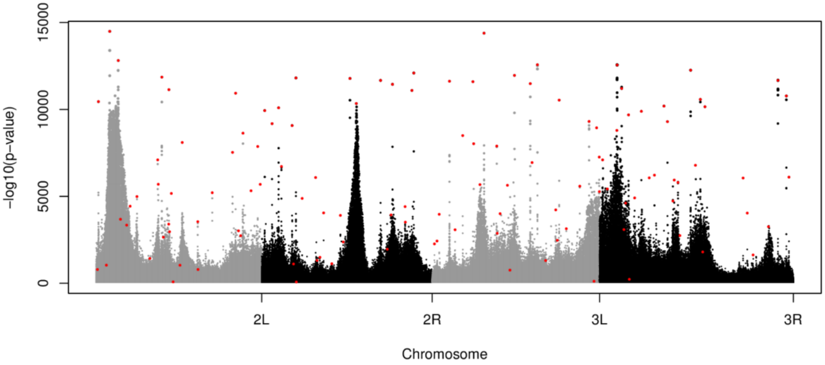
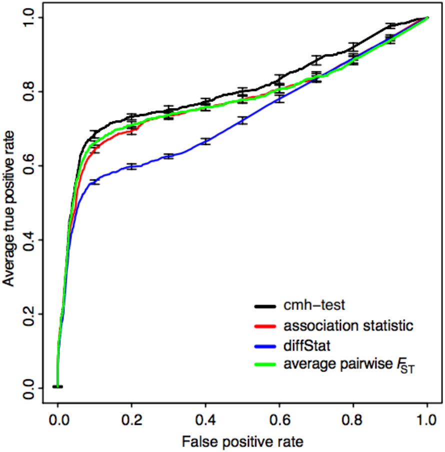


Figure 32: Example of a Manhattan plot from the high budget study design with 150 beneficial SNPs (red) having a selection coefficient of $s = 0.1$

⇒ these are simulated results, therefore we also know the truly positively selected loci (red)

THE CMH-TEST HAS THE BEST PERFORMANCE

For E&R studies (and possibly also for Pool-GWAS) the cmh-test actually has the highest power to identify causative sites of the four evaluated test statistics.



⇒ from Kofler and Schlötterer (2014): A guide for the Design of Evolve and Resequencing studies; MBE

Sample preparation..

To demonstrate the usage of the cmh-test we need to cheat a bit by artificially creating a replicate (the cmh-test requires at least four samples).

```
cat samll.sync | awk ’BEGIN{OFS="\t"}{print $0,$4,$5}’ > smallcmh
    .sync
```

resulting in:

```
2R 1 G 0:0:0:1:0:0 0:0:0:1:0:0 0:0:0:1:0:0 0:0:0:1:0:0
2R 2 A 2:0:0:0:0:0 2:0:0:0:0:0 2:0:0:0:0:0 2:0:0:0:0:0
2R 3 C 0:0:3:0:0:0 0:0:3:0:0:0 0:0:3:0:0:0 0:0:3:0:0:0
2R 4 C 0:0:4:0:0:0 0:0:4:0:0:0 0:0:4:0:0:0 0:0:4:0:0:0
2R 5 C 0:0:5:0:0:0 0:0:5:0:0:0 0:0:5:0:0:0 0:0:5:0:0:0
```

⇒ thus the two last columns of the sync file have been repeated.
Now, let’s assume the following (just like in the figure above):

- population 1: base population of first replicate
- population 2: evolved population of first replicate
- population 3: base population of second replicate
- population 4: evolved population of second replicate

RUNNING THE CMH-TEST

```
perl ~/programs/popoolation2/cmh-test.pl --min-count 6 --min-coverage 4 --max-coverage 120 --population 1-2,3-4 --input smallcmh.sync --output cmhtest.txt
```

opening the file we get:

```
2R 552 G 0:0:0:14:0:0 0:0:3:11:0:0 0:0:0:14:0:0 0:0:3:11:0:0 1.469866
2R 1594 T 0:12:0:2:0:0 0:13:0:1:0:0 0:12:0:2:0:0 0:13:0:1:0:0 0.173036
2R 2405 T 2:26:0:0:0:0 1:27:0:0:0:0 2:26:0:0:0:0 1:27:0:0:0:0 0.169092
2R 2411 T 0:27:1:0:0:0 0:26:2:0:0:0 0:27:1:0:0:0 0:26:2:0:0:0 0.169092
2R 2701 T 0:26:2:0:0:0 0:26:1:1:0:0 0:26:2:0:0:0 0:26:1:1:0:0 0.1482162
```

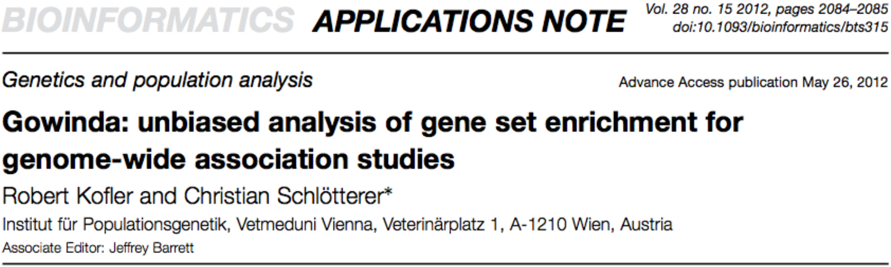
⇒ thus the -log10 transformed p-value is just appended at the end of the sycn file
⇒ Exercise: from these cmh results, get the 10 most consistently differentiated SNPs

VISUALIZE RESULTS OF THE CMH-TEST WITH IGV

```
perl ~/programs/popoolation2/export/cmh2gwas.pl --
    input cmhtest.txt --output cmh.gwas
java -Xmx2g -jar ~/programs/IGV_2.3.26/igv.jar
```

- load pop1.bam
- load pop2.bam
- load cmh.gwas

GOWINDA



BIOINFORMATICS APPLICATIONS NOTE

Vol. 28 no. 15 2012, pages 2084–2085
doi:10.1093/bioinformatics/bts315

Genetics and population analysis

Advance Access publication May 26, 2012

Gowinda: unbiased analysis of gene set enrichment for genome-wide association studies

Robert Kofler and Christian Schlötterer*

Institut für Populationsgenetik, Vetmeduni Vienna, Veterinärplatz 1, A-1210 Wien, Austria

Associate Editor: Jeffrey Barrett

THE QUESTION

I have 200 top differentiated SNPs. So what is the biological significance of these SNPs? Are they enriched for any pathway?
Gowinda could for example be used in the following situations:

- with PoPoolation1 a genome wide estimate of Tajima's *D* was calculated. You pick the top 100 (most negative values) and want to know if they share any common biological pathway?
- with PoPoolation2 you calculated the genome-wide F_{ST} between two natural population and are interested whether the SNPs most differentiated between the two populations are enriched in some pathway?
- you performed a Pool-GWAS using the cmh-test and now you are interested if your causative sites share a biological theme?
- with an E&R study you identified some 100 SNPs putatively being responsible for adaptation to the artificial environment. So what is the biological meaning of those SNPs?

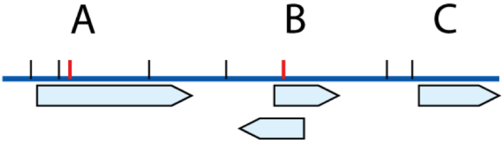
WHEN TO USE GOWINDA?

Gowinda can be used with data from Pool-Seq studies. However, with Pool-Seq you loose the haplotype information. To account for linkage disequilibrium between the causative sites Gowinda implements two complementary algorithm that make two extreme assumptions about LD, either total independence of the SNPs or total linkage between the SNPs within a gene.

This is an acceptable compromise for Pool-Seq data where we do not have any information about LD. However, if you have performed a GWAS and you have the actual haplotypes for every individual available, other tools, that take into account the available information about linkage, may provide a higher resolution.

WHY TO USE GOWINDA?

Why should you use Gowinda and not, for example GoMiner, to analyse the results of genome-wide Pool-Seq studies? Gowinda accounts for several biases that may dramatically compound standard GO enrichment analysis.



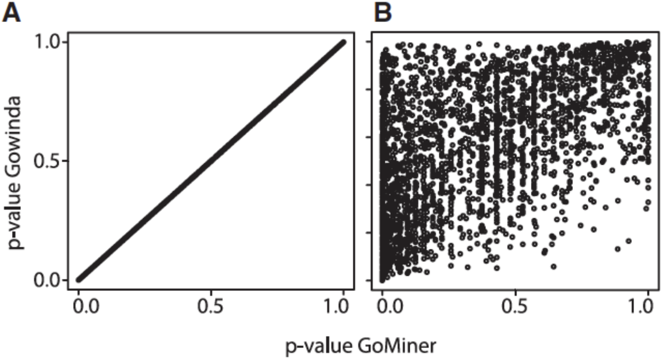
- A.) Gene length bias; longer genes have more SNPs and therefore have a higher probability of containing a candidate SNP ⇒ violates the assumption that all genes have the same probability of being sampled
- B.) Overlapping genes; A single SNP may therefore be associated with several genes ⇒ violates the assumption of independence of the gens
- C.) Some genes do not contain SNPs, such genes can therefore never be sampled ⇒ again violates the assumption that all genes have the same probability of being sampled

BIAS GIBBERISH...IS THIS NOT TOTALLY EXAGGERATED?

Well I randomly sampled 1000 SNPs from the genes of *D. melanogaster* and performed classical GO term enrichment analysis with GoMiner.
⇒ After FDR correction GoMiner reported 341 significantly enriched GO categories! ⇒ YES, correcting for the gene length bias is important!!! These are randomly drawn SNPs, there should not be a single category enriched!
⇒ Same analysis performed with Gowinda reported 0 significantly enriched GO terms after FDR correction.

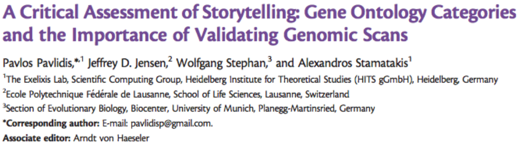
VALIDATION

Comparing the results of Gowinda and GoMiner with an unbiased dataset (A), where every gene has exactly 5 SNPs and with a biased data set (B) where a SNP was introduced all 100 bp. For each evaluation 1000 SNPs were randomly picked.



Is Gowinda too conservative?

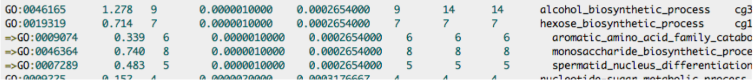
Some of my colleagues actually think that Gowinda is too conservative because the tool did not find any enrichment in their favorite sample. And well...discussing GO enrichment in papers is the favorite pastime of some biologists ;)



A Critical Assessment of Storytelling: Gene Ontology Categories and the Importance of Validating Genomic Scans

Pavlos Pavlidis,*[1] Jeffrey D. Jensen,[2] Wolfgang Stephan,[3] and Alexandros Stamatakis[1]
[1]The Exelixis Lab, Scientific Computing Group, Heidelberg Institute for Theoretical Studies (HITS gGmbH), Heidelberg, Germany
[2]Ecole Polytechnique Fédérale de Lausanne, School of Life Sciences, Lausanne, Switzerland
[3]Section of Evolutionary Biology, Biocenter, University of Munich, Planegg-Martinsried, Germany
***Corresponding author:** E-mail: pavlidisp@gmail.com.
Associate editor: Arndt von Haeseler

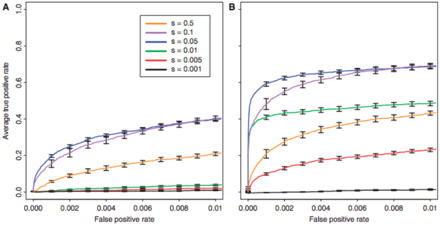
Gowinda is performing permutation tests. If the tool does not find any significant enrichment, than the sample just does not contain anything that significantly deviates from random expectations (i.e.: random picking of SNPs produces similar results)!
⇒ However, I also tested whether Gowinda reports significant results for 5 truly enriched categories ⇒ YES it does!



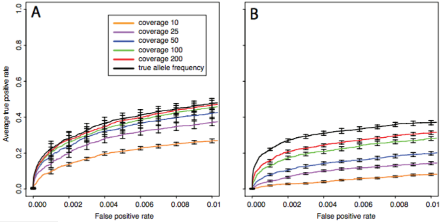
```
GO:0046165      1.278   9       0.0000010000    0.0002654000    9       14      14      alcohol_biosynthetic_process    cg3
GO:0019319      0.714   7       0.0000010000    0.0002654000    7       7       7       hexose_biosynthetic_process     cg1
=>GO:0009074      0.339   6       0.0000010000    0.0002654000    6       6       6       aromatic_amino_acid_family_catabo
=>GO:0046364      0.740   8       0.0000010000    0.0002654000    8       8       8       monosaccharide_biosynthetic_proce
=>GO:0007289      0.483   5       0.0000010000    0.0002654000    5       5       5       spermatid_nucleus_differentiation
GO:0009225      0.152   4       0.0000020000    0.0003176667    4       4       4       nucleotide-sugar_metabolic_process
```

Most whole-genome scans are underpowered

To reliable identify causative sites with whole genome scans you need a powerfully designed study. For example an optimal designed E&R study requires a population size of 2000, 10 replicates and 120 generations of adaptation [for details see Kofler and Schlötterer (2014) MBE].



And high sequencing coverages, in the order of 50-200:



GOWINDA ALGORITHM - REQUIREMENTS

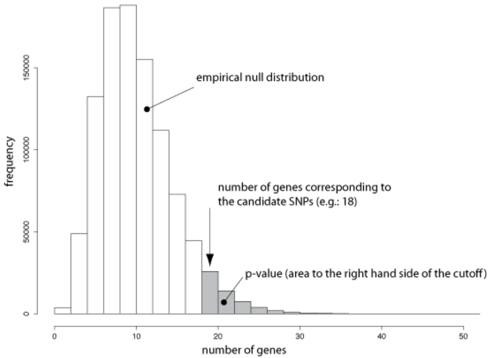
Requirements:

- a set of SNPs (all SNPs tested for significance)
- a set of most significant SNPs (a subset of the above)
- an annotation of the reference genome (position of genes)
- gene sets (i.e.: a list of genes being associated with a given GO category)

Gowinda algorithm - p-value

Gowinda randomly draws SNPs (the same number as candidate SNPs) from the total set of SNPs, and than annotates the overlapping genes and the corresponding GO categories. From this an empirical null distribution for the expected abundance of genes can be derived for every GO category. A p-value is than computed by comparing the null distribution to the results obtained with the candidate SNPs.

Example of an empirical null distribution for a single gene set (e.g.: intracellular protein transport) and 1 million simulations



empirical null distribution

number of genes corresponding to the candidate SNPs (e.g.: 18)

p-value (area to the right hand side of the cutoff)

Gowinda algorithm - FDR

A typically gene set analysis is performed for several thousand gene categories at the same time, which results in a pronounced multiple testing problem. Therefore, Gowinda also derives an empirical FDR value from the simulations.

$$g = 1,2,\ldots,G$$

$$s = 1,2,\ldots,S$$

$$R_{obs} = \sum_{g=1}^{G} I(p_g \leq P)$$

$$R_{\exp} = \frac{1}{S}\sum_{s=1}^{S}\sum_{g=1}^{G} I(p_g^{\,s} \leq P)$$

$$FDR = \frac{R_{\exp}}{R_{obs}}$$

For example, if your analysis reports 20 GO categories significant with an $FDR \leq 0.05$ than on average 1 (or less) of these 20 will be a false positive.

Gowinda exercise

```
cd ~/Desktop/gowinda
ll
```

Present files:

- annotation in gtf
- total_snps.txt: all SNPs
- cand_snps.txt: the candidate SNPs
- goassociations_cg.txt: the GO annotation of every gene (the categories associated with every gene)

SNP FILES

⇒ the following file definition apply to both SNP files, the total set of SNPs and the candidates (the candidates need to be contained in the total set)

```
2L 13148294 A 99:14:0:0:0:0 142:7:0:0:0:0 78:25:0:0:0:0
    91:37:0:1:0:0 134
2L 11201295 G 0:26:0:22:0:0 0:20:0:48:0:0 0:4:0:39:0:0
    0:6:0:71:0:0 30
2L 1795300 C 85:0:34:0:0:1 84:0:44:0:0:0 54:0:70:0:0:0
    52:0:65:0:0:0 0.9863
2L 5096101 A 20:47:0:0:0:0 50:65:0:0:0:0 78:38:0:1:0:0
    79:47:0:0:0:0 0.543
```

- col 1: reference chromosome
- col 2: position
- the rest will be ignored

⇒ thus most commonly used file formats for variants can be directly used as SNP files (sync, vcf, cmh, etc)

THE ANNOTATION (GTF FORMAT)

```
2L FlyBase exon 8193 8589 . + . gene_id "CG11023";
2L FlyBase exon 8668 9484 . + . gene_id "CG11023";
2L FlyBase exon 9839 11344 . - . gene_id "CG2671";
2L FlyBase exon 11410 11518 . - . gene_id "CG2671";
```

- col 1: reference chromosome
- col 3: feature; exon and cds need to be present for gowinda
- col 4: start position of feature
- col 5: end position of feature
- col 7: strand
- col 9: comment; contains the gene ID. The gene ID is very important and will be used for cross-linking with the gene set file (GO association).

THE GENE SET FILE

⇒ for this file there is an important difference between tabulator and space! A tab separates the major columns and the space minor entries within a colum.

```
GO:0000002 mitochondrial genome maintenance CG11077 CG33650 CG4337 CG5924
GO:0000003 reproduction CG10112 CG10128 CG1262 CG13873 CG14034 CG15117 CG15616 CG1656
GO:0000009 alpha-1,6-mannosyltransferase activity CG8412
GO:0000010 trans-hexaprenyltranstransferase activity CG10585 CG31005
GO:0000012 single strand break repair CG4208 CG5316
```

- col 1: the ID of the GO category
- col 2: short description of the GO category (may contain spaces)
- col 3: a list of genes being associated with the given GO category (may contain spaces); these gene IDs have to be present in the annotation (gtf)!!

⇒ From where to get such a file? a.) direct download from the FuncAssociate webpage, b.) from the GoMiner webpage (there is a tutorial for this on the Gowinda webpage) or c.) you create custom gene sets.

Starting Gowinda

```
java -Xmx2g -jar ~/programs/gowinda/Gowinda-1.12.jar --snp-file
    total_snps.txt --candidate-snp-file cand_snps.txt --
    annotation-file annotation.gtf --gene-set-file
    goassociations_cg.txt --output-file results_gene_gene.txt --
    simulations 100000 --min-significance 1 --gene-definition
    gene --mode gene --threads 2  --min-genes 1
```

- –mode gene: assume complete LD within a gene
- –gene-definition gene: a SNP will be linked to a gene if it is either overlapping with an exon or an intron of the gene (other definitions supported are exon, cds, utr, upstream3000,..).
- –simulations 100000; the number of simulations to perform; this will just influence the precision of the estimated p-values; You will not get more significantly enriched categories when performing more simulations!
- –min-significance to report; 1 means to report results for all GO categories
- –min-genes 1; only use GO categories having at least 1 gene. You may actually want to use 3 or 5

THE RESULTS

⇒ the output is sorted by significance, having the most significant on top of the list

```
GO:0045155 0.050 2 0.0006200000 0.2159200000 2 2 2 electron
    transporter cg13263,cg17903
GO:0006119 0.050 2 0.0006200000 0.2159200000 2 2 8 oxidative
    phosphorylation cg13263,cg17903
GO:0009112 0.066 2 0.0010800000 0.2698133333 2 2 16 nucleobase
    metabolic process cg7811,cg7171
```

- col 1: the ID of the gene set (e.g.: GO category)
- col 2: average number of genes found per simulation in the given gene set
- col 3: actual number of genes found in the given gene set
- col 4: p-value
- col 5: FDR value
- col 6-7-8: some detailed gene statistics (see manual)
- col 9: description of the given gene set
- col 10: a comma separated list of gene IDs found for the given gene set

⇒ What do you think, is there a significant gene set enrichment in this sample?

INCLUDING REGULATORY REGIONS

So with the previous analysis we did not find any significant GO term enrichment. Following King and Wilson (1975) we speculate that our SNPs may actually be enriched in regulatory regions. So we use a different mapping of SNPs to genes and add 2000bp upstream and downstream of a gene. Therefore SNPs being up to 2000 bp up/downstream of a gene will be assumed to be linked with a gene.

```
java -Xmx2g -jar ~/programs/gowinda/Gowinda-1.12.jar --snp-file
    total_snps.txt --candidate-snp-file cand_snps.txt --
    annotation-file annotation.gtf --gene-set-file
    goassociations_cg.txt --output-file
    results_updownstream2000_gene.txt --simulations 100000 --min
    -significance 1 --gene-definition updownstream2000 --mode
    gene --threads 2  --min-genes 1
```

Did this yield significant associations?

```
GO:0005516 1.022 4 0.0025000000 0.8087800000 4 4 29 calmodulin
    binding cg5125,cg8475,cg14472,cg6713
GO:0016831 0.300 3 0.0026300000 0.8087800000 3 4 25 carboxy-lyase
     activity cg5029,cg7811,cg10501
GO:0022618 0.384 3 0.0041300000 0.8087800000 3 6 33
    ribonucleoprotein complex assembly cg7138,cg4602,cg6137
```

⇒ Well NO

ASSUMING LINKAGE EQUILIBRIUM

As we are getting quite desperate and we know that LD is decaying rapidly in our organism we decide to use a different algorithm that makes a complementary extreme assumption about linkage: every SNP is in linkage equilibrium (therfore every significant SNP is actually causative as opposed to being a mere hitchhiker).

```
java -Xmx2g -jar ~/programs/gowinda/Gowinda-1.12.jar --snp-file
    total_snps.txt --candidate-snp-file cand_snps.txt --
    annotation-file annotation.gtf --gene-set-file
    goassociations_cg.txt --output-file
    results_updownstream2000_snp.txt --simulations 100000 --min-
    significance 1 --gene-definition updownstream2000 --mode snp
     --threads 2  --min-genes 1
```

Finally significant results:

```
GO:0006497 0.608 8 0.0000100000 0.0014409091 1 7 45 protein
    lipidation cg18810
GO:0043543 0.793 8 0.0000100000 0.0014409091 1 8 36 protein
    acylation cg18810
GO:0006119 0.168 7 0.0000100000 0.0014409091 2 2 8 oxidative
    phosphorylation cg13263,cg17903
```

⇒ However, you need to be very careful when interpreting these results because of the extreme assumption about LD. I actually recommend the mode gene per default.

MOST COMMON PROBLEM: GENE IDS

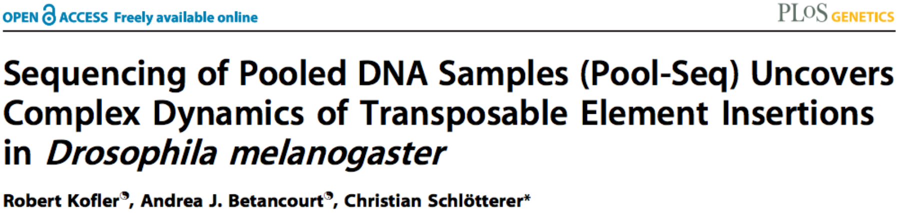
The most common problem is that gene IDs between the annotation and the GO association file are not matching. For *Drosophila* there are several different naming conventions of genes, so that you may end up with totally different gene names in the files ⇒ very annoying!! More rigorous standardization would help..
However, if you encounter this problem we describe two strategies to resolve this issue in the Gowinda online-tutorial:

- you may obtain a file containing synonymous gene IDs, and updated the 'annotation.gtf' with the gene IDs that match with the gene set file. Problem: you need to get such a file..
- parasitize on the great gene ID database curated by the GoMiner team, which contains gene IDs from all different naming conventions. We can provide a list of all gene IDs found in our annotation, sent this to GoMiner and extract an association file suitable for Gowinda (there is tutorial on the Gowinda web-page).

Section 4

Final remarks on Pool-Seq

PoPoolation TE



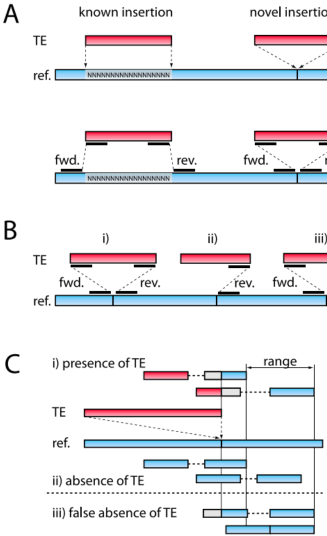
OPEN ACCESS Freely available online

PLoS GENETICS

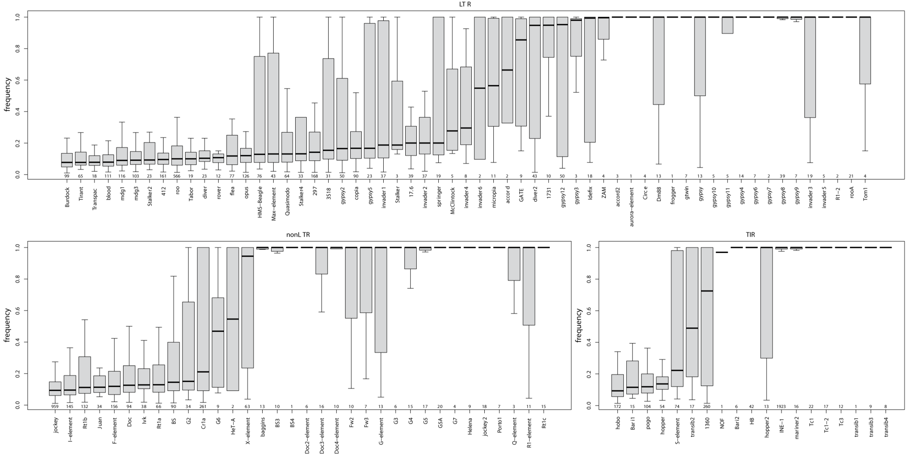
Sequencing of Pooled DNA Samples (Pool-Seq) Uncovers Complex Dynamics of Transposable Element Insertions in *Drosophila melanogaster*

Robert Kofler[¶], Andrea J. Betancourt[¶], Christian Schlötterer*

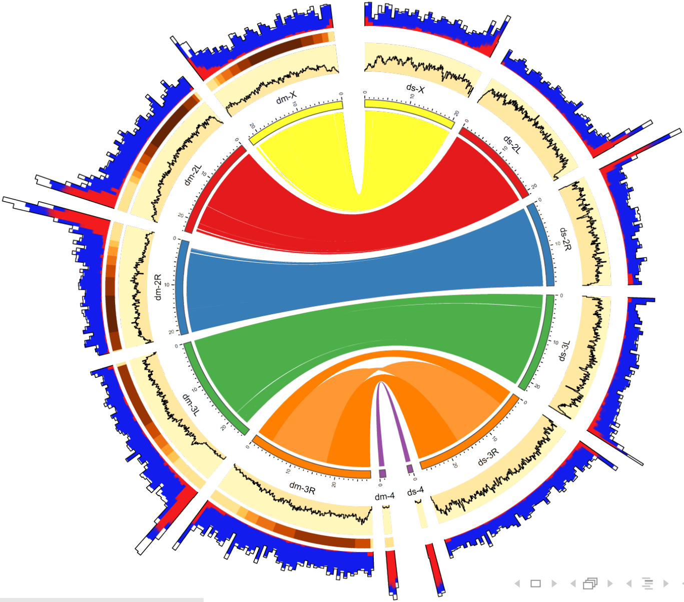
Identify TE insertions



Obtain genome wide estimates of TE abundance



COMPARE TE CONTENT OF RELATED SPECIES



Other tools for Pools?

What else could be done:

- estimate abundance of haplotypes (e.g.: PoolHap, harp)
- currently tools are developed that may allow to estimate recombination rates from pools

MISSING TOOLS FOR POOLS?

What else could (should) be developed:

- estimate microsatellite abundance in pools (this would be very straight-forward)
- CNV (I do not think this will ever be possible)
- indels (this would require reliable algorithm for realignment of mapped reads from Pool-Seq data)

The future of Pool-seq?

Some people argue that, due to the dropping cost of sequencing, pooled samples will soon be useless as sequencing individuals provides superior information (e.g.: haplotypes). I do however believe that Pool-Seq will still be used for some time to come, because:

- Sequencing cost are less; even when sequencing cost drop, Pool-Seq will also become cheaper
- sample preparation is much easier (thus cheaper); One population - one sample; cost of wet-lab work
- data handling is simpler (thus cheaper); thus more people could actually analyze Pool-Seq data whereas for individual sequencing more expert knowledge is necessary.

Pool-seq is a quick and cheap method for addressing specific scientific questions in ecology, evolutionary biology and agriculture. For such specific questions sequencing of individuals would just be an overkill (”to take a sledgehammer to crack a nut”; in german ”mit Kannonen auf Spatzen schiessen” :)