

Science-Wise False Discovery Rate

Nathan (Nat) Goodman

October 2, 2017

An R script and short article exploring the statistical concept of science-wise false discovery rate (SWFDR). Some authors use SWFDR and its complement, positive predictive value, to argue that most (or, at least, many) published scientific results must be wrong unless most hypotheses are a priori true. I disagree. While SWFDR is valid statistically, the real cause of bad science is “Publish or Perish”.

Overview

The `swfdr.R` script reimplements the core idea in [David Colquhoun’s fascinating paper](#), “An investigation of the false discovery rate and the misinterpretation of p-values” and further discussed in [Felix Schönbrodt’s blog post](#), “What’s the probability that a significant p-value indicates a true effect?” and related [ShinyApp](#). Schönbrodt’s post led me to blog posts by [Daniel Lakens](#), “How can $p = 0.05$ lead to wrong conclusions 30% of the time with a 5% Type 1 error rate?” and [Will Gervais](#), “Power Consequences”. The term *science-wise false discovery rate* is from [Leah Jager and Jeffrey Leek’s paper](#), “An estimate of the science-wise false discovery rate and application to the top medical literature”. [John Ioannidis’s landmark paper](#), “Why most published research findings are false”, is the origin of it all.

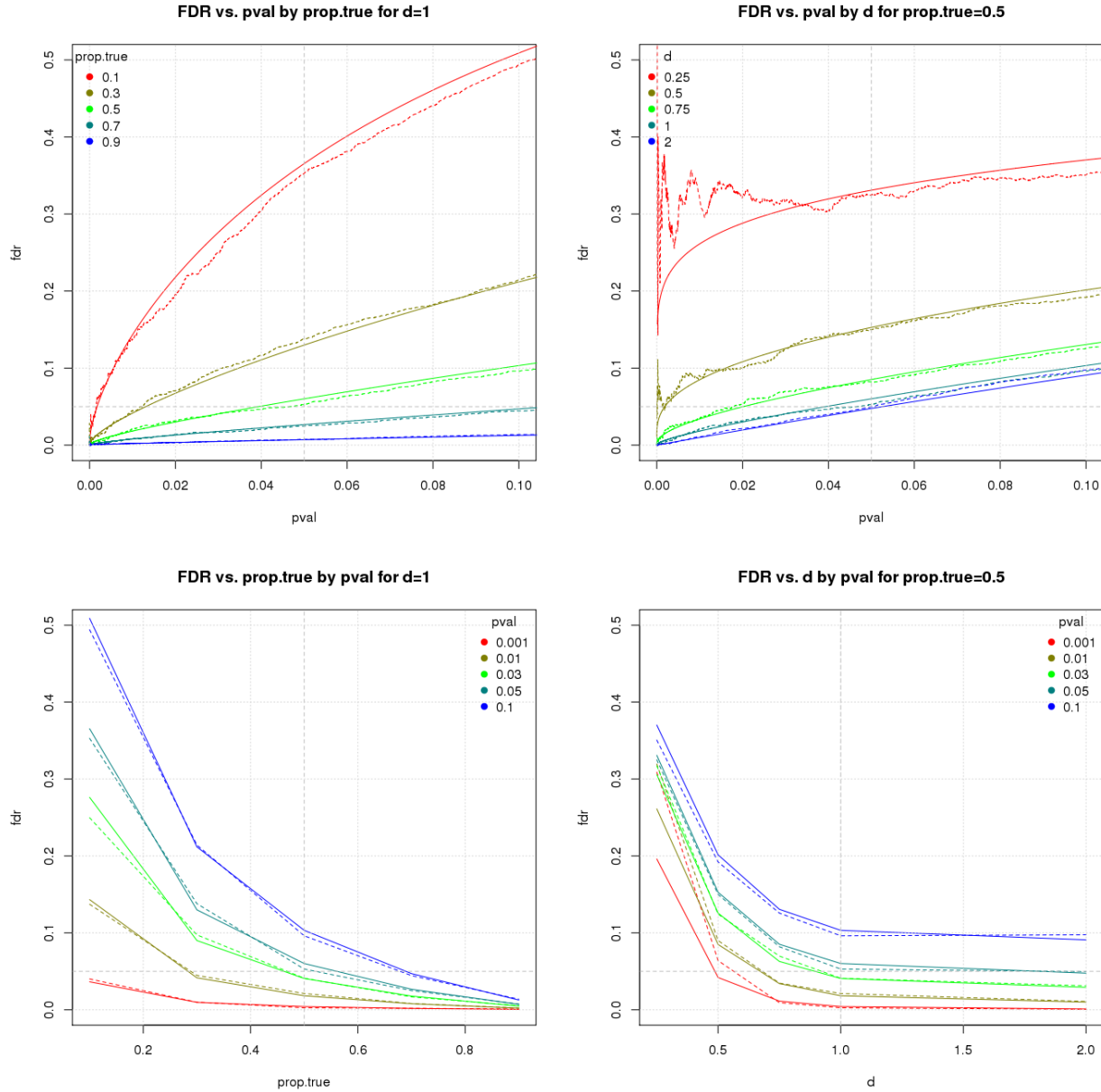
Installation and Usage

The simplest way to get the software is to download the script `swfdr.R` from the R subdirectory of the repository. The script uses base R capabilities only and will run “out of the box” on any (reasonably modern) R installation.

The recommended way to run the script is to `source` it into your R session and run the statement `run()`; as shown below.

```
## This code block assumes your working directory is the root of the distribution.  
  
source('R/swfdr.R');  
run();
```

This runs the program with default parameters producing four graphs similar to the ones below. The default computation performs 2.5×10^5 simulations (taking about 3 minutes on my small Linux server).



The notation is

- solid lines show theoretical results; dashed lines are empirical results from the simulation
- *fdr*. false discovery rate, i.e., the probability that a significant p-value indicates a false positive
- *pval*. p-value cutoff for significance
- *prop.true*. proportion of simulated cases that have a real effect
- *d*. standardized effect size, aka *Cohen's d*

The user can change simulation parameters and control program operation by providing new values to `run()` as illustrated in the code block below. The available parameters are

parameter	meaning	default
prop.true	fraction of cases where there is a real effect	<code>seq(.1,.9,by=.2)</code>
m	number of iterations	<code>1e4</code>
n	sample size	<code>16</code>

parameter	meaning	default
d	standardized effect size (aka <i>Cohen's d</i>)	c(.25,.50,.75,1,2)
pwr	power. if set, program adjusts <i>d</i> to achieve power	NA
sig.level	significance level for power calculations when <i>pwr</i> is set	0.05
pval.plot	p-values for which we plot results	c(.001,.01,.03,.05,.1)
scriptname	used to set output directories and in error messages	'swfdr'
datadir	data directory relative to distribution root	'data'
figdir	figure directory relative to distribution root	'figure'
save	save parameters, results, and plots; sets <i>save.rdata</i> and <i>save.plot</i> , not <i>save.txt</i>	FALSE
save.rdata	save parameters and results in RData format	FALSE (set by <i>save</i>)
save.txt	save results in txt format. CAUTION: big and slow	FALSE (not set by <i>save</i>)
save.plot	save plots	FALSE (set by <i>save</i>)
clean	remove contents of data and figure directories; sets <i>clean.data</i> and <i>clean.fig</i>	FALSE
clean.data	remove contents of data directory	FALSE (set by <i>clean</i>)
clean.fig	remove contents of figure directory	FALSE (set by <i>clean</i>)

```
## this code block assumes your working directory is the root of the repository

source("script/swfdr.R");

## run default process and save results in directories data/guide01 and figure/guide01
run(save=T,datadir='data/example01',figdir='figure/example01');

## reduce runtime by reducing number of simulation runs and simulated cases
run(m=1e3,d=c(0.25,0.5,1),prop.true=c(0.3,0.5,0.8));

## specify power directly and let program adjust effect size
run(m=1e3,pwr=c(0.1,0.3,0.8),prop.true=c(0.3,0.5,0.8));

## skip computation by loading saved results and replotting graphs
loadit();
doplot();

## load saved results and do something completely different
## eg, plot distribution of effect size error for small effects with significant p-values
loadit();
with(subset(sim,subset=(d==0.25&d.true&pval<=0.05)),hist(diff-d));
```

See Also

The article discussing the results is available in [html](#) and [pdf](#) formats on the [GitHub Pages site](#) associated with this repository. It's also in the repository as files [swfdr.html](#) and [swfdr.pdf](#). (But note that GitHub, unlike GitHub Pages, renders html files as raw text).

Author

Nathan (Nat) Goodman, (natg at shore.net)

Bugs and Caveats

Please report any bugs, other problems, and feature requests using the [GitHub Issue Tracker](#). I will be notified, and you'll be apprised of progress.

Copyright & License

Copyright (c) 2017 Nathan Goodman

The software is **open source and free**, released under the [MIT License](#). The documentation is **open access**, released under the [Creative Commons Attribution 4.0 International License](#).