There is an entertaining rumour going around about the journal *Nature Communications*. When I heard it for the fourth or fifth time, I decided to check out whether there is any truth in it.

The rumour goes something like this: the impact factor of *Nature Communications* is driven by physical sciences papers.

Sometimes it is put another way: cell biology papers drag down the impact factor of *Nature Communications*, or that they don’t deserve the high JIF tag of the journal because they are cited at lower rates. Could this be true?

**TL;DR** it is true but the effect is not as big as the rumour suggests.

*Nature Communications* is the mega-journal that sits below the subject-specific Nature journals. Operating as an open access, pay-to-publish journal it is a way for Springer Nature to capture revenue from papers that were good, but did not make the editorial selection for subject-specific Nature journals. This is a long-winded way of saying that there are wide variety of papers covered by this journal which publishes around 5,000 papers per year. This complicates any citation analysis because we need a way to differentiate papers from different fields. I describe one method to do this below.

**Quick look at the data**

I had a quick look at the top 20 papers from 2016-2017 with the most citations in 2018. There certainly were a lot of non-biological papers in there. Since highly cited papers disproportionately influence the Journal Impact Factor, then this suggested the rumours might be true.

|  |  |
| --- | --- |
| Citations (2018) | Title |
| 238 | 11.4% Efficiency non-fullerene polymer solar cells with trialkylsilyl substituted 2D-conjugated polymer as donor |
| 226 | Circular RNA profiling reveals an abundant circHIPK3 that regulates cell growth by sponging multiple miRNAs |
| 208 | Recommendations for myeloid-derived suppressor cell nomenclature and characterization standards |
| 203 | High-efficiency and air-stable P3HT-based polymer solar cells with a new non-fullerene acceptor |
| 201 | One-Year stable perovskite solar cells by 2D/3D interface engineering |
| 201 | Massively parallel digital transcriptional profiling of single cells |
| 177 | Array of nanosheets render ultrafast and high-capacity Na-ion storage by tunable pseudocapacitance |
| 166 | Multidimensional materials and device architectures for future hybrid energy storage |
| 163 | Coupled molybdenum carbide and reduced graphene oxide electrocatalysts for efficient hydrogen evolution |
| 149 | Ti3C2 MXene co-catalyst on metal sulfide photo-absorbers for enhanced visible-light photocatalytic hydrogen production |
| 149 | Balancing surface adsorption and diffusion of lithium-polysulfides on nonconductive oxides for lithium-sulfur battery design |
| 146 | Adaptive resistance to therapeutic PD-1 blockade is associated with upregulation of alternative immune checkpoints |
| 140 | Conductive porous vanadium nitride/graphene composite as chemical anchor of polysulfides for lithium-sulfur batteries |
| 136 | Fluorination-enabled optimal morphology leads to over 11% efficiency for inverted small-molecule organic solar cells |
| 134 | The somatic mutation profiles of 2,433 breast cancers refines their genomic and transcriptomic landscapes |
| 132 | Photothermal therapy with immune-adjuvant nanoparticles together with checkpoint blockade for effective cancer immunotherapy |
| 131 | Enhanced electronic properties in mesoporous TiO2 via lithium doping for high-efficiency perovskite solar cells |
| 125 | Electron-phonon coupling in hybrid lead halide perovskites |
| 123 | A sulfur host based on titanium monoxide@carbon hollow spheres for advanced lithium-sulfur batteries |
| 121 | Biodegradable black phosphorus-based nanospheres for in vivo photothermal cancer therapy |

**Let’s dive in to the data**

We will use R for this analysis. If you want to work along, the script and data can be downloaded below. With a few edits, the script will also work for similar analysis of other journals.

[NatCommunDownload](http://quantixed.org/wp-content/uploads/2019/06/NatCommun.zip)

First of all I retrieved three datasets.

* Citation data for the journal. We’ll look at 2018 Journal Impact Factor, so we need citations in 2018 to papers in the journal published in 2016 and 2017. This can be retrieved from Scopus as a csv.
* Pubmed XML file for the Journal to cover the articles that we want to analyse. Search term = “Nat Commun”[Journal] AND (“2016/01/01″[PDAT] : “2017/12/31″[PDAT])
* Pubmed XML file to get cell biology MeSH terms. Search term = “J Cell Sci”[Journal] AND (“2016/01/01″[PDAT] : “2017/12/31″[PDAT])

**Using MeSH terms to segregate the dataset**

Analysing the citation data is straightforward, but how can we classify the content of the dataset? I realised that I could use Medical Subject Heading (MeSH) from PubMed to classify the data. If I retrieved the same set of papers from PubMed and then check which papers had MeSH terms which matched that of a “biological” dataset, the citation data could be segregated. I used a set of J Cell Sci papers to do this. **Note that these MeSH terms are not restricted to cell biology, they cover all kinds of biochemistry and other aspects of biology.** The papers that do *not* match these MeSH terms are ecology, chemistry and physical sciences (many of these don’t have MeSH terms). We start by getting our biological MeSH terms.

require(XML)

require(tidyverse)

require(readr)

## extract a data frame from PubMed XML file

## This is modified from christopherBelter's pubmedXML R code

extract\_xml <- function(theFile) {

newData <- xmlParse(theFile)

records <- getNodeSet(newData, "//PubmedArticle")

pmid <- xpathSApply(newData,"//MedlineCitation/PMID", xmlValue)

doi <- lapply(records, xpathSApply, ".//ELocationID[@EIdType = \"doi\"]", xmlValue)

doi[sapply(doi, is.list)] <- NA

doi <- unlist(doi)

authLast <- lapply(records, xpathSApply, ".//Author/LastName", xmlValue)

authLast[sapply(authLast, is.list)] <- NA

authInit <- lapply(records, xpathSApply, ".//Author/Initials", xmlValue)

authInit[sapply(authInit, is.list)] <- NA

authors <- mapply(paste, authLast, authInit, collapse = "|")

year <- lapply(records, xpathSApply, ".//PubDate/Year", xmlValue)

year[sapply(year, is.list)] <- NA

year <- unlist(year)

articletitle <- lapply(records, xpathSApply, ".//ArticleTitle", xmlValue)

articletitle[sapply(articletitle, is.list)] <- NA

articletitle <- unlist(articletitle)

journal <- lapply(records, xpathSApply, ".//ISOAbbreviation", xmlValue)

journal[sapply(journal, is.list)] <- NA

journal <- unlist(journal)

volume <- lapply(records, xpathSApply, ".//JournalIssue/Volume", xmlValue)

volume[sapply(volume, is.list)] <- NA

volume <- unlist(volume)

issue <- lapply(records, xpathSApply, ".//JournalIssue/Issue", xmlValue)

issue[sapply(issue, is.list)] <- NA

issue <- unlist(issue)

pages <- lapply(records, xpathSApply, ".//MedlinePgn", xmlValue)

pages[sapply(pages, is.list)] <- NA

pages <- unlist(pages)

abstract <- lapply(records, xpathSApply, ".//Abstract/AbstractText", xmlValue)

abstract[sapply(abstract, is.list)] <- NA

abstract <- sapply(abstract, paste, collapse = "|")

ptype <- lapply(records, xpathSApply, ".//PublicationType", xmlValue)

ptype[sapply(ptype, is.list)] <- NA

ptype <- sapply(ptype, paste, collapse = "|")

mesh <- lapply(records, xpathSApply, ".//MeshHeading/DescriptorName", xmlValue)

mesh[sapply(mesh, is.list)] <- NA

mesh <- sapply(mesh, paste, collapse = "|")

theDF <- data.frame(pmid, doi, authors, year, articletitle, journal, volume, issue, pages, abstract, ptype, mesh, stringsAsFactors = FALSE)

return(theDF)

}

# function to separate multiple entries in one column to many columns using | separator

# from https://stackoverflow.com/questions/4350440/split-data-frame-string-column-into-multiple-columns

split\_into\_multiple <- function(column, pattern = ", ", into\_prefix){

cols <- str\_split\_fixed(column, pattern, n = Inf)

# Sub out the ""'s returned by filling the matrix to the right, with NAs which are useful

cols[which(cols == "")] <- NA

cols <- as\_tibble(cols)

# name the 'cols' tibble as 'into\_prefix\_1', 'into\_prefix\_2', ..., 'into\_prefix\_m'

# where m = # columns of 'cols'

m <- dim(cols)[2]

names(cols) <- paste(into\_prefix, 1:m, sep = "\_")

return(cols)

}

## First load the JCS data to get the MeSH terms of interest

jcsFilename <- "./jcs.xml"

jcsData <- extract\_xml(jcsFilename)

# put MeSH into a df

meshData <- as.data.frame(jcsData$mesh, stringsAsFactors = FALSE)

colnames(meshData) <- "mesh"

# separate each MeSH into its own column of a df

splitMeshData <- meshData %>%

bind\_cols(split\_into\_multiple(.$mesh, "[|]", "mesh")) %>%

select(starts\_with("mesh\_"))

splitMeshData <- splitMeshData %>%

gather(na.rm = TRUE) %>%

filter(value != "NA")

# collate key value df of unique MeSH

uniqueMesh <- unique(splitMeshData)

# this gives us a data frame of cell biology MeSH terms

Now we need to load in the Nature Communications XML data from PubMed and also get the citation data into R.

## Now use a similar procedure to load the NC data for comparison

ncFilename <- "./nc.xml"

ncData <- extract\_xml(ncFilename)

ncMeshData <- as.data.frame(ncData$mesh, stringsAsFactors = FALSE)

colnames(ncMeshData) <- "mesh"

splitNCMeshData <- ncMeshData %>%

bind\_cols(split\_into\_multiple(.$mesh, "[|]", "mesh")) %>%

select(starts\_with("mesh\_"))

# make a new column to hold any matches of rows with MeSH terms which are in the uniqueMeSH df

ncData$isCB <- apply(splitNCMeshData, 1, function(r) any(r %in% uniqueMesh$value))

rm(splitMeshData,splitNCMeshData,uniqueMesh)

## Next we load the citation data file retrieved from Scopus

scopusFilename <- "./Scopus\_Citation\_Tracker.csv"

# the structure of the file requires a little bit of wrangling, ignore warnings

upperHeader <- read\_csv(scopusFilename,

skip = 5)

citationData <- read\_csv(scopusFilename,

skip = 6)

upperList <- colnames(upperHeader)

lowerList <- colnames(citationData)

colnames(citationData) <- c(lowerList[1:7],upperList[8:length(upperList)])

rm(upperHeader,upperList,lowerList)

Next we need to perform a *join* to match up the PubMed data with the citation data.

## we now have two data frames, one with the citation data and one with the papers

# make both frames have a Title column

colnames(citationData)[which(names(citationData) == "Document Title")] <- "Title"

colnames(ncData)[which(names(ncData) == "articletitle")] <- "Title"

# ncData paper titles have a terminating period, so remove it

ncData$Title <- gsub("\\.$","",ncData$Title, perl = TRUE)

# add citation data to ncData data frame

allDF <- inner\_join(citationData, ncData, by = "Title")

Now we’ll make some plots.

# Plot histogram with indication of mean and median

p1 <- ggplot(data=allDF, aes(allDF$'2018')) +

geom\_histogram(binwidth = 1) +

labs(x = "2018 Citations", y = "Frequency") +

geom\_vline(aes(xintercept = mean(allDF$'2018',na.rm = TRUE)), col='orange', linetype="dashed", size=1) +

geom\_vline(aes(xintercept = median(allDF$'2018',na.rm = TRUE)), col='blue', linetype="dashed", size=1)

p1

# Group outlier papers for clarity

p2 <- allDF %>%

mutate(x\_new = ifelse(allDF$'2018' > 80, 80, allDF$'2018')) %>%

ggplot(aes(x\_new)) +

geom\_histogram(binwidth = 1, col = "black", fill = "gray") +

labs(x = "2018 Citations", y = "Frequency") +

geom\_vline(aes(xintercept = mean(allDF$'2018',na.rm = TRUE)), col='orange', linetype="dashed", size=1) +

geom\_vline(aes(xintercept = median(allDF$'2018',na.rm = TRUE)), col='blue', linetype="dashed", size=1)

p2

# Plot the data for both sets of papers separately

p3 <- ggplot(data=allDF, aes(allDF$'2018')) +

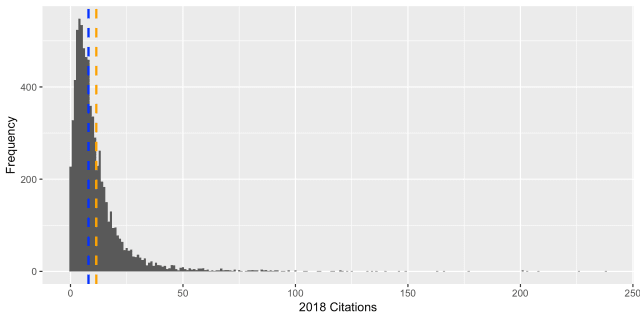
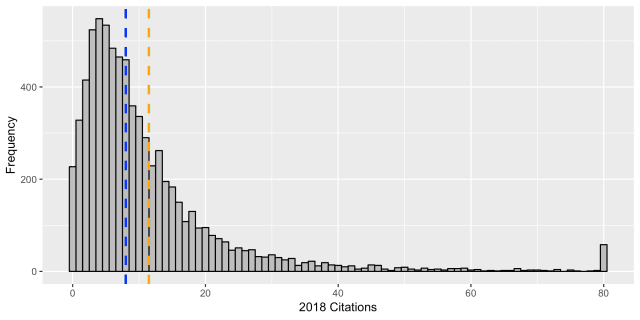
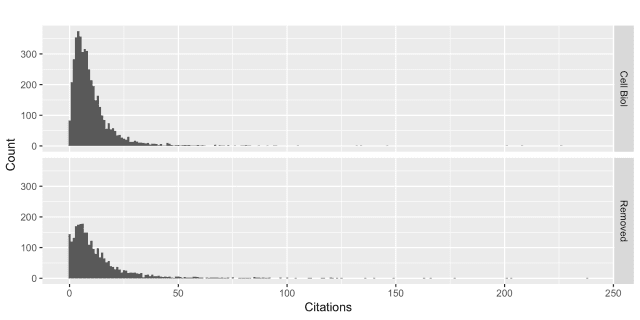
geom\_histogram(binwidth = 1) +

labs(title="",x = "Citations", y = "Count") +

facet\_grid(ifelse(allDF$isCB, "Cell Biol", "Removed") ~ .) +

theme(legend.position = "none")

p3

* 
* 
* 

The citation data look typical: highly skewed, with few very highly cited papers and the majority (two-thirds) receiving less than the mean number of citations. The “cell biology” dataset and the non-cell biology dataset look pretty similar.

Now it is time to answer our main question. Do cell biology papers drag down the impact factor of the journal?

## make two new data frames, one for the cell bio papers and one for non-cell bio

cbDF <- subset(allDF,allDF$isCB == TRUE)

nocbDF <- subset(allDF,allDF$isCB == FALSE)

# print a summary of the 2018 citations to these papers for each df

summary(allDF$'2018')

summary(cbDF$'2018')

summary(nocbDF$'2018')

> summary(allDF$'2018')

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 4.00 8.00 11.48 14.00 238.00

> summary(cbDF$'2018')

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 4.00 7.00 10.17 13.00 226.00

> summary(nocbDF$'2018')

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 4.00 9.00 13.61 16.00 238.00

The “JIF” for the whole journal is 11.48, whereas for the non-cell biology content it is 13.61. The cell biology dataset has a “JIF” of 10.17. So basically, **the rumour is true but the effect is quite mild**. The rumour is that the cell biology impact factor is *much* lower.

The reason “JIF” is in quotes is that it is notoriously difficult to calculate this metric. All citations are summed for the numerator, but the denominator comprises “citable items”. To get something closer to the actual JIF, we probably should remove non-citable items. These are Errata, Letters, Editorials and Retraction notices.

## We need to remove some article types from the dataset

itemsToRemove <- c("Published Erratum","Letter","Editorial","Retraction of Publication")

allArticleData <- as.data.frame(allDF$ptype, stringsAsFactors = FALSE)

colnames(allArticleData) <- "ptype"

splitallArticleData <- allArticleData %>%

bind\_cols(split\_into\_multiple(.$ptype, "[|]", "ptype")) %>%

select(starts\_with("ptype\_"))

# make a new column to hold any matches of rows that are non-citable items

allDF$isNCI <- apply(splitallArticleData, 1, function(r) any(r %in% itemsToRemove))

# new data frame with only citable items

allCitableDF <- subset(allDF,allDF$isNCI == FALSE)

# Plot the data after removing "non-citable items for both sets of papers separately

p4 <- ggplot(data=allCitableDF, aes(allCitableDF$'2018')) +

geom\_histogram(binwidth = 1) +

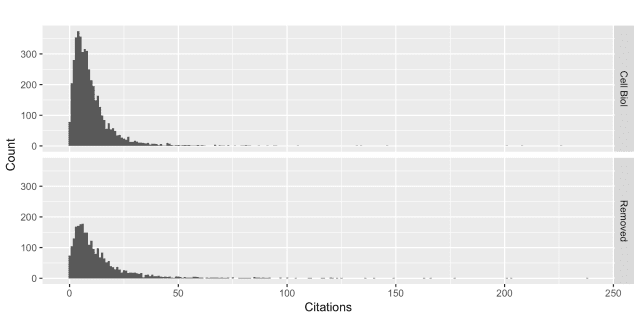
labs(title="",x = "Citations", y = "Count") +

facet\_grid(ifelse(allCitableDF$isCB, "Cell Biol", "Removed") ~ .) +

theme(legend.position = "none")

p4

After removal the citation distributions look a bit more realistic (notice that the earlier versions had many items with zero citations).

Citation distributions with non-citable items removed

Now we can redo the last part.

# subset new dataframes

cbCitableDF <- subset(allCitableDF,allCitableDF$isCB == TRUE)

nocbCitableDF <- subset(allCitableDF,allCitableDF$isCB == FALSE)

# print a summary of the 2018 citations to these papers for each df

summary(allCitableDF$'2018')

summary(cbCitableDF$'2018')

summary(nocbCitableDF$'2018')

> summary(allCitableDF$'2018')

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 4.00 8.00 11.63 14.00 238.00

> summary(cbCitableDF$'2018')

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 4.00 8.00 10.19 13.00 226.00

> summary(nocbCitableDF$'2018')

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 5.00 9.00 14.06 17.00 238.00

Now the figures have changed. The “JIF” for the whole journal is 11.63, whereas for the non-cell biology content it would 14.06. The cell biology dataset has a “JIF” of 10.19. To more closely approximate the JIF, we need to do:

# approximate "impact factor" for the journal

sum(allDF$'2018') / nrow(allCitableDF)

# approximate "impact factor" for the journal's cell biology content

sum(cbDF$'2018') / nrow(cbCitableDF)

# approximate "impact factor" for the journal's non-cell biology content

sum(nocbDF$'2018') / nrow(nocbCitableDF)

> # approximate "impact factor" for the journal

> sum(allDF$'2018') / nrow(allCitableDF)

[1] 11.64056

> # approximate "impact factor" for the journal's cell biology content

> sum(cbDF$'2018') / nrow(cbCitableDF)

[1] 10.19216

> # approximate "impact factor" for the journal's non-cell biology content

> sum(nocbDF$'2018') / nrow(nocbCitableDF)

[1] 14.08123

This made only a minor change, probably because the dataset is so huge (7239 papers for two years with non-citable items removed). If we were to repeat this on another journal with more front content and fewer papers, this distinction might make a bigger change.

**Conclusion**

So the rumour is true *but* the effect is not as big as people say. There’s a ~17% reduction in potential impact factor by including these papers rather than excluding them. However, these papers comprise ~63% of the corpus and they bring in an estimated revenue to the publisher of $12,000,000 per annum.

**It is definitely not true that these papers are under-performing.** Their citation rates are similar to those in the best journals in the field. Note that citation rates do not necessarily reflect the usefulness of the paper. For one thing they are largely an indicator of the volume of a research field. Anyhow, **next time you hear this rumour for someone, you can set them straight.**