

# Programmatic access for Altmetrics

Scott Chamberlain (@recology\_)  
rOpenSci / Simon Fraser University



Supported by the  
**ALFRED P. SLOAN  
FOUNDATION**

Find this talk here <http://bit.ly/roalm>

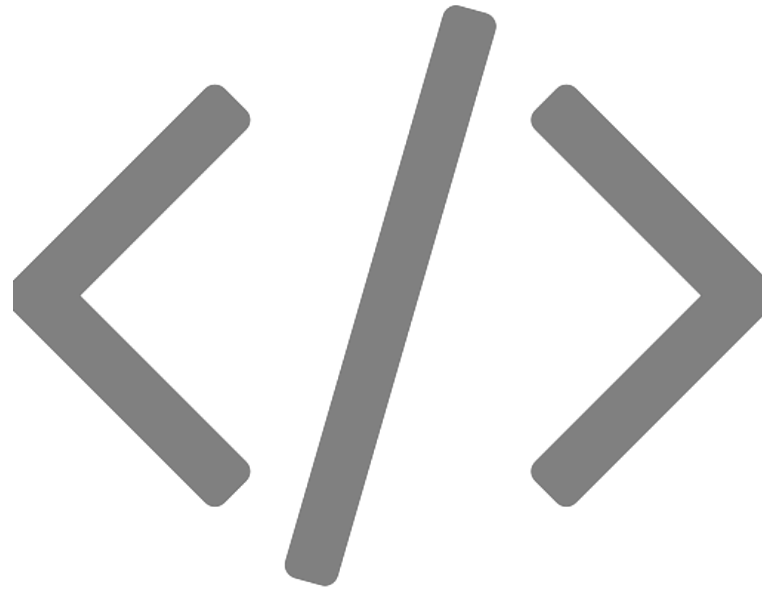
Made with [Slidify](#), the code [here](#)

Press "o" to bring up all slides - "w" to change aspect - "g" to go to page

# Programmatic access to altmetrics

## Open altmetrics data

# Programmatic access



4/37

# Programmatic access to altmetrics

Computers are simply better at repetitive tasks

- Makes repetitive tasks take far less time
- Facilitates tool creation by developers
- Allows research questions to be addressed more quickly
- Facilitates reproducibility

# What is needed for easy programmatic access?

# Modern API technology

## REST API

The modern way to serve data to consumers

Makes data consumption easy from any programming language

- Base URI, e.g. <http://foo.com>
- Media type, e.g., json, xml
- HTTP verbs, like GET, POST, PUT, PATCH, HEAD, etc...

# Proper HTTP status codes

- 1xx - informational
- 2xx - success
- 3xx - redirection
- 4xx - client error
- 5xx - server error



**Due to the Federal government shutdown,  
NOAA.gov and most associated web sites  
are unavailable.**

**Specific NOAA web sites necessary to protect lives  
and property are operational and will be maintained.**

**See [Weather.gov](http://Weather.gov) for critical weather  
information or contact [USA.gov](http://USA.gov) for more  
information about the shutdown.**

**NOAA Federal Employees: For access to the  
Notice to Federal Employees About Unemployment  
Insurance (SF-8), please [Click Here](#).**

# Good docs (for developers)



10/37

# Authentication

OAuth makes sense for web workflows, but not so much for programmatic workflows.

Having both options is nice.

# A spec for REST?

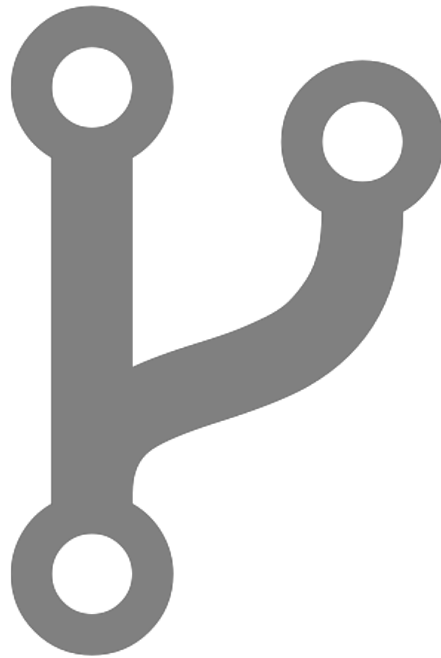
RAML - <http://raml.org/>

Programmatically create new clients

```
##RAML 0.8  
  
title: World Music API  
baseUri: http://example.api.com/{version}  
version: v1  
/songs:  
  get:  
  post:  
  ...
```

Good place to include altmetrics standards...

# Deploying APIs is probably hard



13/37

# Consuming altmetrics programmatically

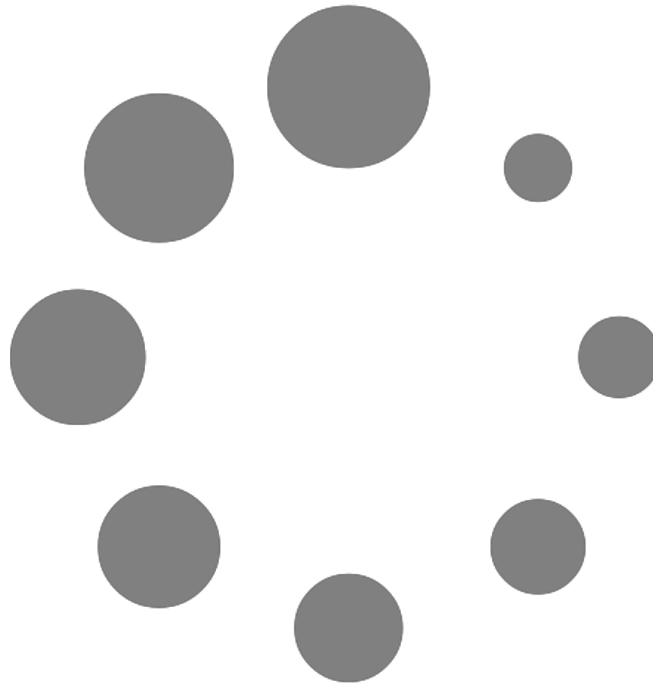
14/37

# We need altmetrics research



15/37

# Programmatic access to altmetrics data key for reproducibility



16/37



# Having a look at the literature...

- [Do Altmetrics Work? Twitter and Ten Other...](#) - via Altmetric.com
- [Tweeting biomedicine: an analysis of tweets...](#) - via Altmetric.com
- [The Spread of Scientific Information...](#) - via PLOS ALM
- [Can Tweets Predict Citations? ...](#) - via Twitter Search API
- [Altmetrics in the Wild...](#) - via PLOS ALM, various APIs, WebofSci citations
- [Social Media Release Increases Dissemination...](#) - via manual collection
- [Identifying Audiences of E-Infrastructures...](#) - via Google Analytics
- [How the Scientific Community Reacts to...](#) - via Twitter Search API, Google Scholar citations

17/37

# Most popular programming language?

18/37



Obviously



# Many libraries available, but more needed

DATA SOURCE	LIBRARIES	ROPENSCI CONTRIBUTIONS IN R
PLOS ALM	R	<a href="#">alm</a> ** Copernicus, etc.
ImpactStory	R, Javascript	<a href="#">rImpactStory</a>
Altmetric	R, Python, Ruby, iOS	<a href="#">rAltmetric</a>

# Interacting with REST APIs in R

```
out <- GET("http://alm.plos.org/api/v3/articles?doi=10.1371/journal.pmed.1001361&key=<key>")
stop_for_status(out)
content(out)

{
  doi: "10.1371/journal.pmed.1001361",
  title: "Personalized Prediction of Lifetime Benefits with Statin Therapy for Asymptomatic Individuals: A Modeling Study",
  url: "http://www.plosmedicine.org/article/info%3Adoi%2F10.1371%2Fjournal.pmed.1001361",
  mendeley: "437b07d9-bc40-4c57-b60e-1f60fefe2300",
  pmid: "23300388",
  pmcid: "3531501",
  publication_date: "2012-12-27T08:00:00Z",
  update_date: "2013-10-07T11:06:58Z",
  views: 9329,
  shares: 62,
  bookmarks: 5,
  citations: 1
}
```

22/37

# Data via alm interface to PLOS ALM

```
alm(doi = "10.1371/journal.pone.0029797")
```

An object of class "almtot"

Slot "meta":

\$doi

```
[1] "10.1371/journal.pone.0029797"
```

...<more metadata>

Slot "summary":

```
  views shares bookmarks citations
1 29229   237         51         7
```

Slot "data":

	.id	pdf	html	shares	groups	comments	likes	citations	total
1	bloglines	NA	NA	NA	NA	NA	NA	0	0
2	citeulike	NA	NA	1	NA	NA	NA	NA	1
3	connotea	NA	NA	NA	NA	NA	NA	0	0
4	crossref	NA	NA	NA	NA	NA	NA	7	7
5	nature	NA	NA	NA	NA	NA	NA	4	4
...									

23/37

# Combining metrics across aggregators

DATA SOURCE	PLOS	IMPACTSTORY	ALTMETRIC
WebOfScience	webofscience	--	--
Dryad	--	dryad:total_downloads	--
Figshare	figshare	figshare:views shares downloads	--
Github	--	github:forks stars	--
Google+	--	--	cited by gplus count
Mendeley readers	mendeley shares	mendeley readers	mendeley readers
Twitter	twitter	topsy:tweets	cited by tweeters count

24/37



# Proposed R library

metaAlm - ( )

Combine altmetrics data across providers (ImpactStory, Altmetric, etc.)  
and across data sources (Twitter, Facebook, etc.)

# Combining metrics

Get data from three different providers

```
plos_data <- alm(<doi>)  
impactstory_data <- metrics(<doi>)  
altmetric_data <- altmetric_data(altmetrics(<doi>))
```

Easily combine data with a single function, and highlight inconsistencies

```
alt_combine(plos_data, impactstory_data, altmetric_data)
```

Warning: Inconsistency in facebookLikes, check metadata

	dataSource	fromProvider	values
1	twitter	PLOSALM	100
2	facebookLikes	ImpactStory	50
3	facebookLikes	Altmetric	40
4	scopusCitations	Altmetric	150

# Example in R

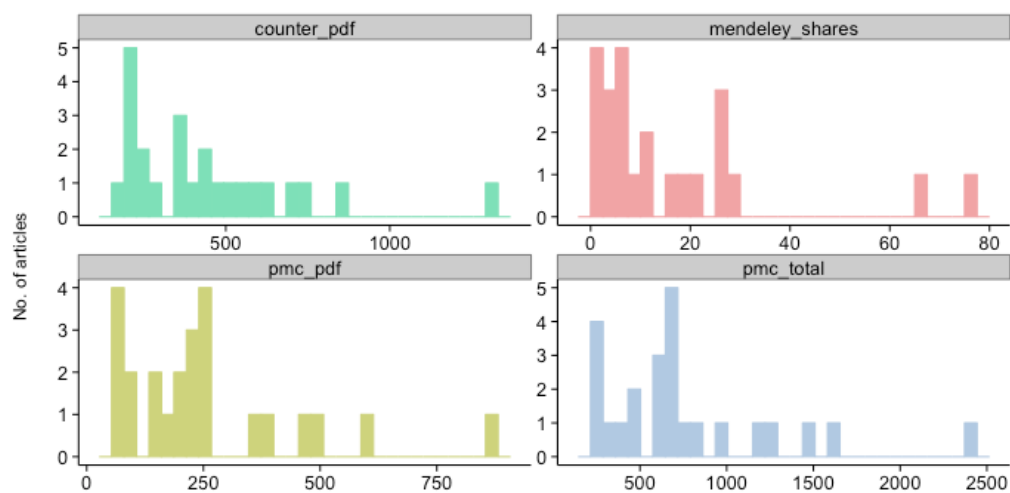
Load libraries, get 200 DOIs, get ALM data, plot

```
library(rplos); library(alm); library(plyr)

dois <- searchplos(terms='*:* ', fields="id", limit=200)

alm <- ldply( alm(doi=do.call(c,dois$id), total_details=TRUE) )

plot_density(alm, c("counter_pdf", "mendeley_shares", "pmc_pdf", "pmc_total"), c("#83DFB4", "#EFA5A5", "#CFD470", "#B2C9E4"), plot_type="h")
```

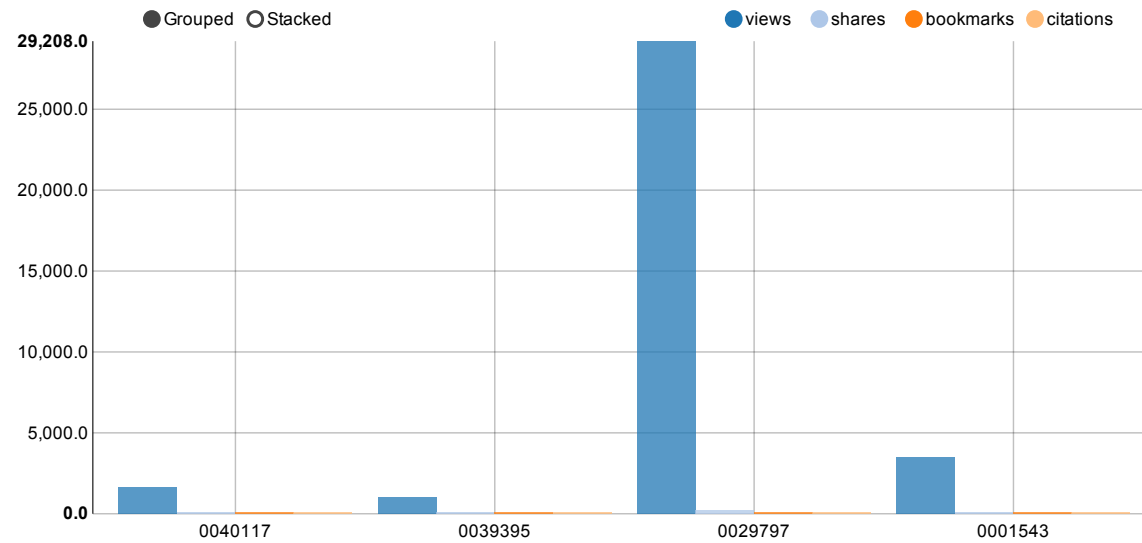


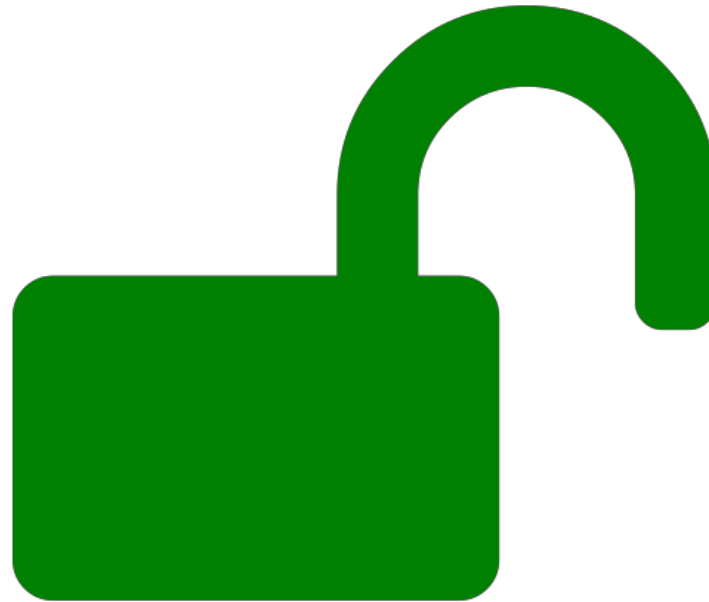
27/37

```
library(rplos); library(alm); library(rCharts)

dois <- c('10.1371/journal.pone.0001543', '10.1371/journal.pone.0040117', '10.1371/journal.pone.0029797', '10.1371/journal.pone.0039395')
dat <- signposts(doi=dois)

plot_signposts(input = dat, type = "multiBarChart", height = 400)
```





“I’d argue that [#opendata](#) today is exactly where open source was some 2 decades ago”-[@BenBalter](#) <http://t.co/VJ6QiLybUU> [#oss](#)  
— Alex Howard (@digiphile) [October 9, 2013](#)

# Why is openness a good thing?

Altmetrics needs checks on

- Consistency (tweets from source A and B should be =)
- Correlation (is metric A strongly corr. with B?)
- Interpretation (open source the interpretation)
- Gaming (security through obscurity doesn't work)

31/37

# Why is openness a good thing?

Altmetrics needs checks on

- Consistency (tweets from source A and B should be =)
- Correlation (is metric A strongly corr. with B?)
- Interpretation (open source the interpretation)
- Gaming (security through obscurity doesn't work)

**Open data makes all this easier**

32/37

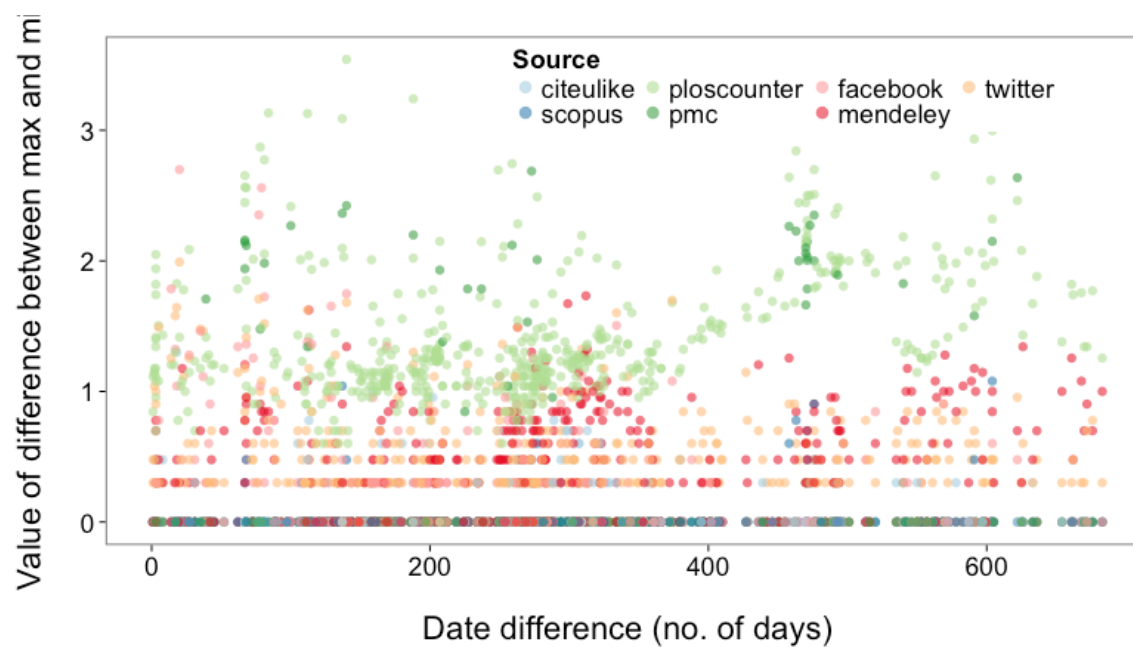


# Additional value from openness

- Knowledge from research findings
  - Doesn't require open data I suppose :(, but helps facilitate research
  - e.g., think how hard text-mining is - we don't want that in altmetrics
- Open products
  - [ReaderMeter](#)
  - [ScienceCard](#)
- For-profit products
- Who knows? Making data open allows many experiments, some of which will stick

33/37

# An open use case

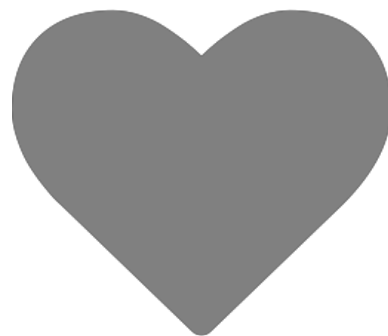


[10.3789/isqv25no2.2013.02](https://doi.org/10.3789/isqv25no2.2013.02)

34/37

# Programmatic access Open altmetrics data

# Programmatic access



# open data

# Programmatic access to Open altmetrics data

37/37