AllBio / EU CodeFest

bmpvieira.com/allbio14

Bruno Vieira | @bmpvieira

Phd Student @ Queen Mary University of London

Bioinformatics and Population Genomics

Supervisor:

Yannick Wurm | @yannick__

Before:

FACULDADE · DE · CIÊNCIAS UNIVERSIDADE · DE · LISBOA

CoBiG² http://cobig2.fc.ul.pt

Computational Biology & Population Genomics Group

XIV Congress of The European Society for Evolutionary Biology
19 to 24 August 2013
Lisbon . Portugal

geeklist

BBSRC

Some problems I faced during my research:

- Difficulty getting relevant descriptions and datasets from NCBI API using bio* libs
- For web projects, needed to implement the same functionality on browser and server
- Difficulty writing scalable, reproducible and complex bioinformatic pipelines

Bionode.io - *Modular and universal bioinformatics*

Pipeable UNIX command line tools and JavaScript / Node.js APIs for bioinformatic analysis workflows on the server and browser.

Collaborates with BioJS - *Represent biological data on the web*

Dat - *Build data pipelines*

Provides a streaming interface between every file format and data storage backend. *"git for data"*

dat-data.com | 🐦 @maxogden | 🐦 @mafintosh

bionode.io (online shell)

Examples

BASH

```
bionode-ncbi urls assembly Solenopsis invicta | grep genomic.fna

http://ftp.ncbi.nlm.nih.gov/genomes/all/GCA_000188075.1_Si_gnG/
GCA_000188075.1_Si_gnG_genomic.fna.gz
```

```
bionode-ncbi download sra arthropoda | bionode-sra
```

```
bionode-ncbi download gff bacteria
```

JavaScript

```
var ncbi = require('bionode-ncbi')
ncbi.urls('assembly', 'Solenopsis invicta'), gotData)
function gotData(urls) {
  var genome = urls[0].genomic.fna
  download(genome)
})
```

Difficulty getting relevant description and datasets from NCBI API using bio* libs

Python example

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Achromyrmex")
esearch_record = Entrez.read(esearch_handle)
for id in esearch_record['IdList']:
  esummary_handle = Entrez.esummary(db="assembly", id=id)
  esummary_record = Entrez.read(esummary_handle)
  documentSummarySet = esummary_record['DocumentSummarySet']
  document = documentSummarySet['DocumentSummary'][0]
  metadata_XML = document['Meta'].encode('utf-8')
  metadata = ET.fromstring('<root>' + metadata_XML + '</root>')
  for entry in Metadata[1]:
    print entry.text
```

```
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA_000188075.1_Si_gnG
```

Solution: bionode-ncbi

Need to reimplement the same code on browser and server.

Solution: JavaScript everywhere

- Afra
- SequenceServer
- GeneValidator
- BioJS

Biodalliance is converting parsers to Bionode

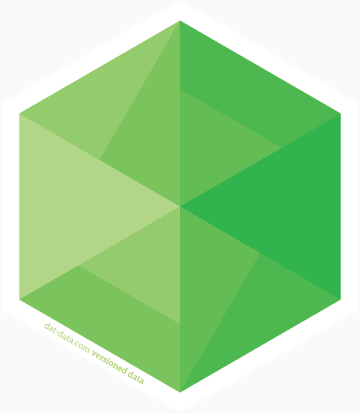
Difficulty writing scalable, reproducible and complex bioinformatic pipelines.

Solution: Node.js Streams everywhere

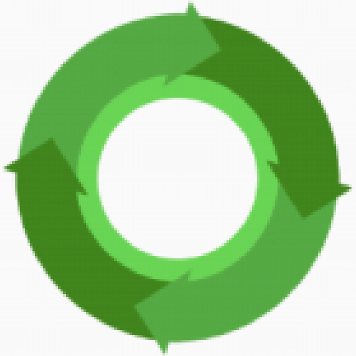
```
var ncbi = require('bionode-ncbi')
var tool = require('tool-stream')
var through = require('through2')
var fork1 = through.obj()
var fork2 = through.obj()

ncbi
  .search('sra', 'Solenopsis invicta')
  .pipe(fork1)
  .pipe(dat.reads)

fork1
  .pipe(tool.extractProperty('expxml.Biosample.id'))
  .pipe(ncbi.search('biosample'))
  .pipe(dat.samples)

fork1
  .pipe(tool.extractProperty('uid'))
  .pipe(ncbi.link('sra', 'pubmed'))
```

Benefit from other JS projects

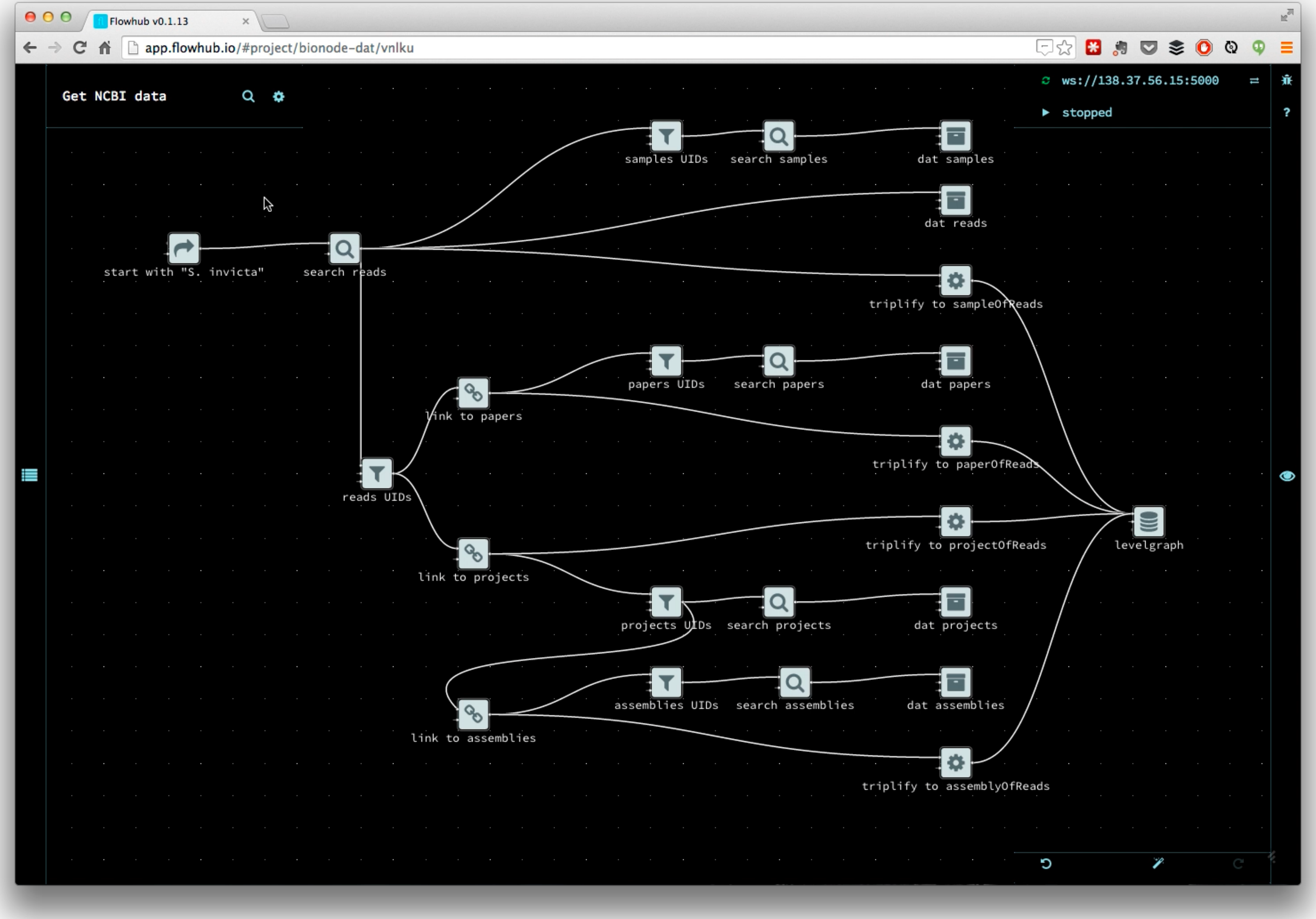
Dat



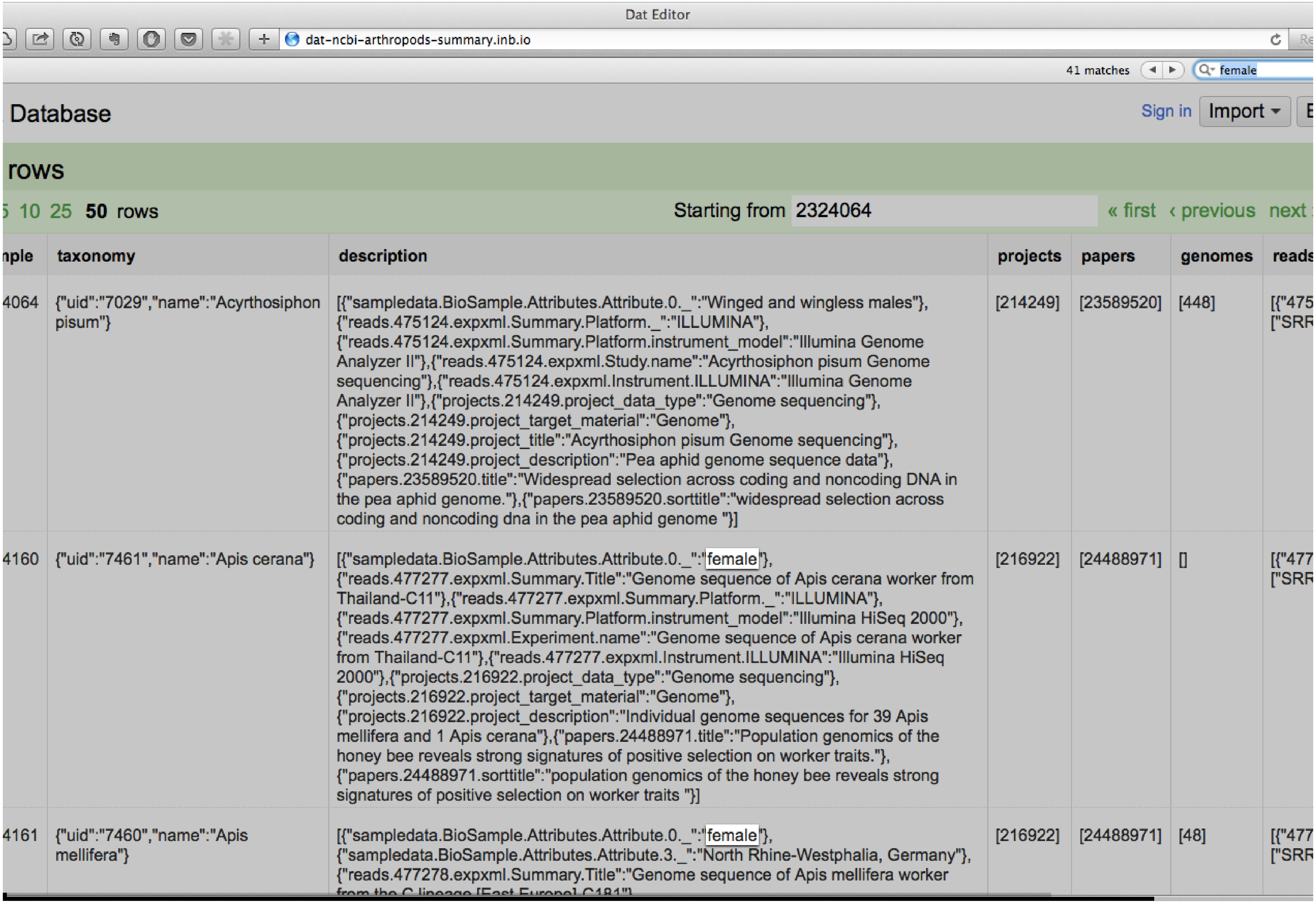
BioJS



NoFlo



Flowhub v0.1.13
app.flowhub.io/#project/bionode-dat/vnlku
Get NCBI data
ws://138.37.56.15:5000
stopped
start with "S. invicta"
search reads
samples UIDs
search samples
dat samples
dat reads
triplify to sampleOfReads
reads UIDs
link to papers
papers UIDs
search papers
dat papers
triplify to paperOfReads
link to projects
triplify to projectOfReads
projects UIDs
search projects
dat projects
link to assemblies
assemblies UIDs
search assemblies
dat assemblies
triplify to assemblyOfReads
levelgraph



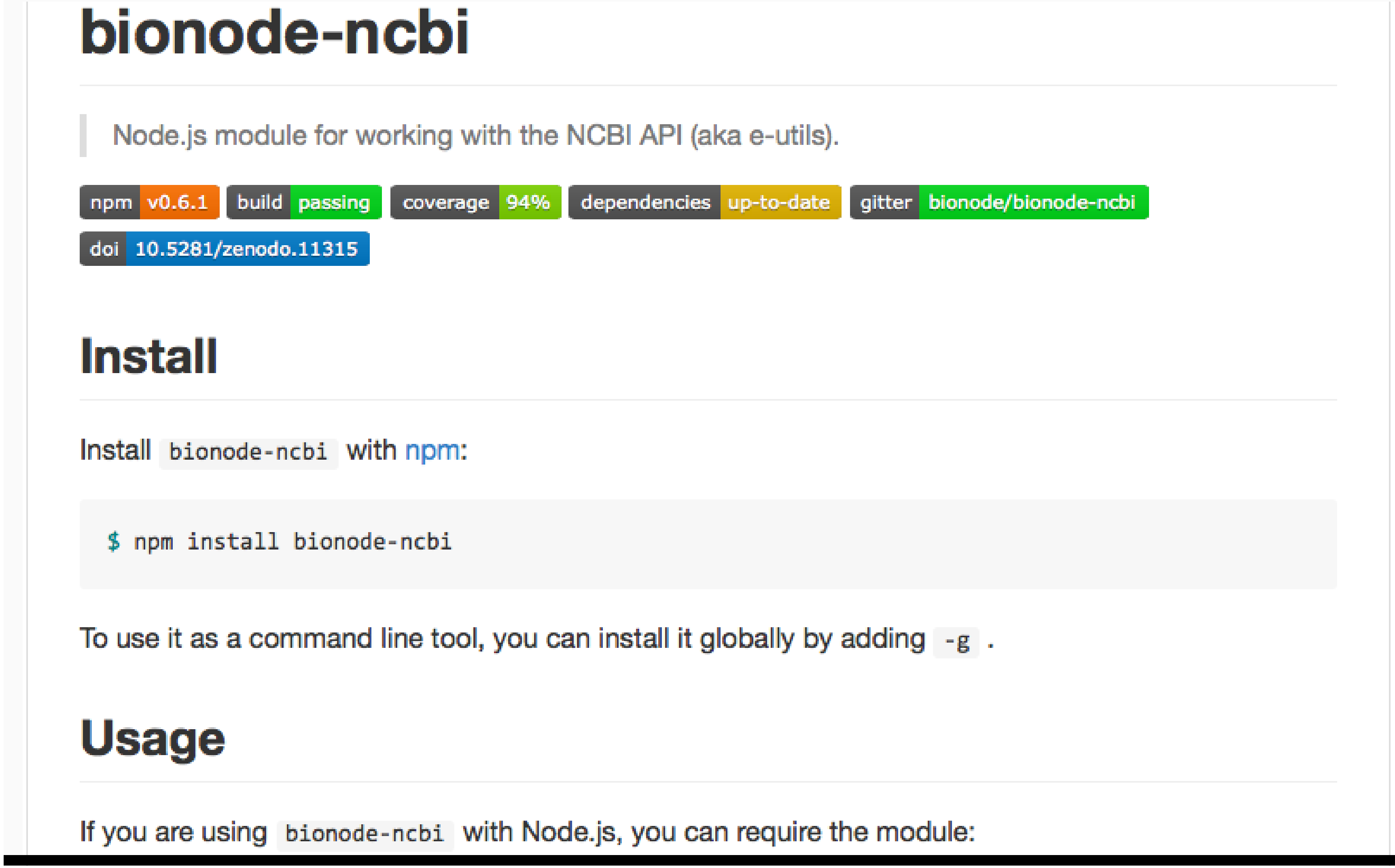
Database

rows

5 10 25 **50** rows Starting from 2324064 « first ‹ previous next

nple	taxonomy	description	projects	papers	genomes	reads
4064	{"uid":"7029","name":"Acyrthosiphon pisum"}	[{"sampledata.BioSample.Attributes.Attribute.0._":"Winged and wingless males"},{"reads.475124.expxml.Summary.Platform._":"ILLUMINA"},{"reads.475124.expxml.Summary.Platform.instrument_model":"Illumina Genome Analyzer II"},{"reads.475124.expxml.Study.name":"Acyrthosiphon pisum Genome sequencing"},{"reads.475124.expxml.Instrument.ILLUMINA":"Illumina Genome Analyzer II"},{"projects.214249.project_data_type":"Genome sequencing"},{"projects.214249.project_target_material":"Genome"},{"projects.214249.project_title":"Acyrthosiphon pisum Genome sequencing"},{"projects.214249.project_description":"Pea aphid genome sequence data"},{"papers.23589520.title":"Widespread selection across coding and noncoding DNA in the pea aphid genome."},{"papers.23589520.sorttitle":"widespread selection across coding and noncoding dna in the pea aphid genome "}]	[214249]	[23589520]	[448]	[{"475 ["SRR
4160	{"uid":"7461","name":"Apis cerana"}	[{"sampledata.BioSample.Attributes.Attribute.0._":"female"},{"reads.477277.expxml.Summary.Title":"Genome sequence of Apis cerana worker from Thailand-C11"},{"reads.477277.expxml.Summary.Platform._":"ILLUMINA"},{"reads.477277.expxml.Summary.Platform.instrument_model":"Illumina HiSeq 2000"},{"reads.477277.expxml.Experiment.name":"Genome sequence of Apis cerana worker from Thailand-C11"},{"reads.477277.expxml.Instrument.ILLUMINA":"Illumina HiSeq 2000"},{"projects.216922.project_data_type":"Genome sequencing"},{"projects.216922.project_target_material":"Genome"},{"projects.216922.project_description":"Individual genome sequences for 39 Apis mellifera and 1 Apis cerana"},{"papers.24488971.title":"Population genomics of the honey bee reveals strong signatures of positive selection on worker traits."},{"papers.24488971.sorttitle":"population genomics of the honey bee reveals strong signatures of positive selection on worker traits "}]	[216922]	[24488971]	[]	[{"477 ["SRR
4161	{"uid":"7460","name":"Apis mellifera"}	[{"sampledata.BioSample.Attributes.Attribute.0._":"female"},{"sampledata.BioSample.Attributes.Attribute.3._":"North Rhine-Westphalia, Germany"},{"reads.477278.expxml.Summary.Title":"Genome sequence of Apis mellifera worker from the C lineage [East Europe] C181"}	[216922]	[24488971]	[48]	[{"477 ["SRR

Reusable, small and tested modules



bionode-ncbi

> Node.js module for working with the NCBI API (aka e-utils).

npm v0.6.1 build passing coverage 94% dependencies up-to-date gitter bionode/bionode-ncbi doi 10.5281/zenodo.11315

Install

Install `bionode-ncbi` with npm:

```
$ npm install bionode-ncbi
```

To use it as a command line tool, you can install it globally by adding `-g` .

Usage

If you are using `bionode-ncbi` with Node.js, you can require the module:

Some users and Contributors:

- Dat
- Biodalliance
- BioJS
- Yeo Lab (UC San Diego)
  - Michael Lovci
  - Olga Botvinnik
- Afra
- GeneValidator

Soon:

- DNADigest

Thanks!

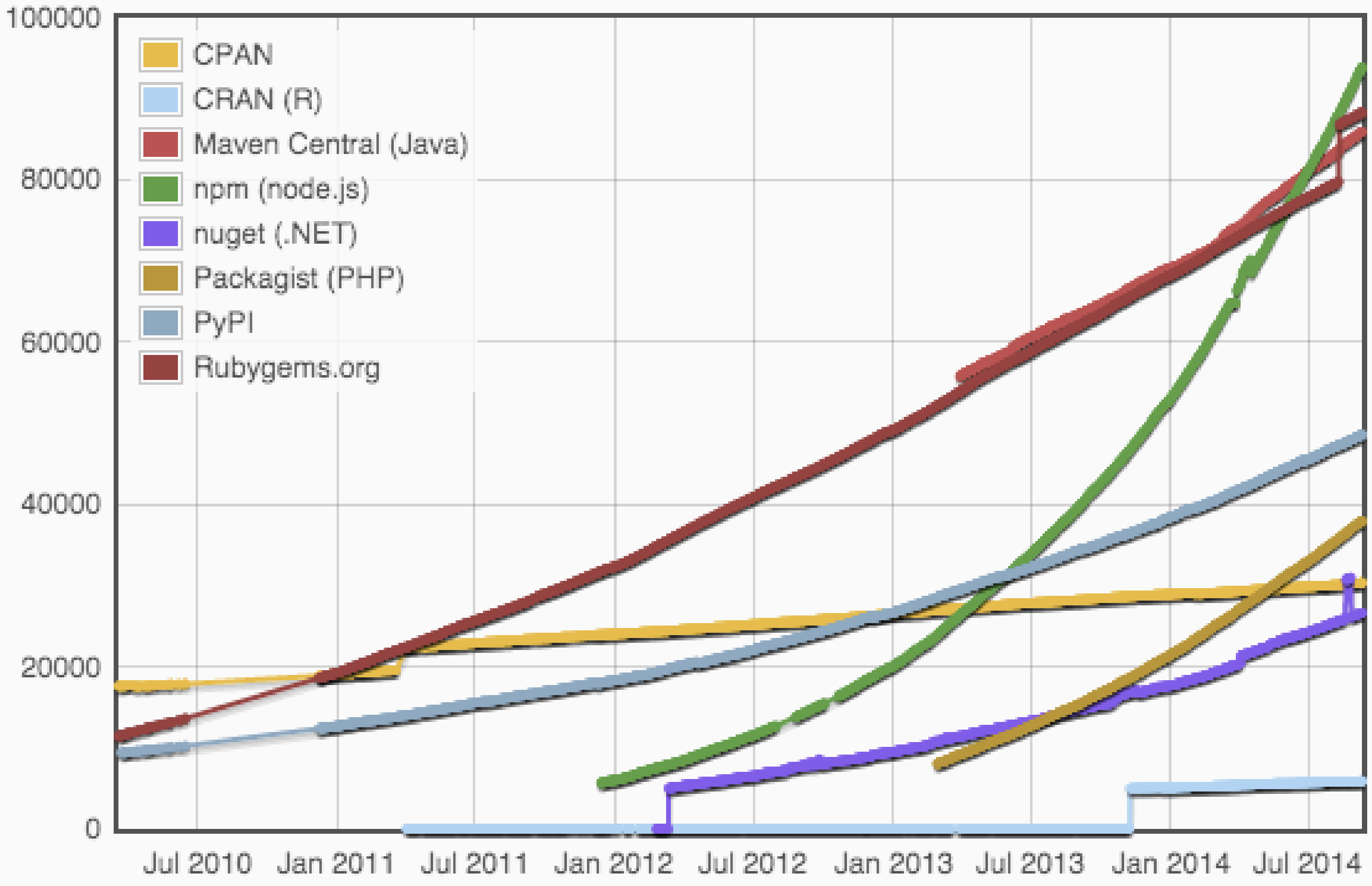
Acknowledgements:

- @yannick__
- @maxogden
- @mafintosh
- @alanmrice
- @dasmoth
- @biodevops

Why Node.js / JavaScript

- Streams applies well to Bioinformatics
- Easy to write CLI wrappers for Streams
- Reusable, small and tested modules
- Same language everywhere (JavaScript)
- Package Manager that works (NPM)
- Huge number modules (93327, 199/day)
- Use other JS projects (Dat, BioJS, NoFlo)
- Possible to write Desktop GUI apps in JS

Module counts



Package Manager that works

```
npm install bionode
npm install bionode -g
npm test
npm start
npm run test-browser
npm run build-docs
npm init
npm publish
```

Not only for JavaScript, C/C++ too:

- Node.js style C/C++ modules
- Native C/C++ running in Google V8