# Bionode Demo

bmpvieira.com/cw15

## Some problems I faced during my research:

## Some problems I faced during my research:

- For web projects, needed to implement the same functionality on browser and server

## Some problems I faced during my research:

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

## Some problems I faced during my research:

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

# Bionode

Bionode.io

# Bionode

Bionode.io - *Modular and universal bioinformatics*

## Bionode

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.

# Bionode

## 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.

#bionode

## Bionode

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.

#bionode

gitter.im/bionode/bionode

## Collaborations and Reusability

Dat-data.com

BioJS.net

- Afra
- GeneValidator
- SequenceServer
- Biodalliance
- GeeFuTu

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

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

**Python example:** URL for the Acromyrmex assembly?

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

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

**Python example:** URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
```

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

**Python example:** URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
```

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

**Python example:** URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
```

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

## Python example: URL for the Acromyrmex assembly?

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

```python
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
```

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

### Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
esearch_record = Entrez.read(esearch_handle)
```

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

### Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
esearch_record = Entrez.read(esearch_handle)
for id in esearch_record['IdList']:
```

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

### Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
esearch_record = Entrez.read(esearch_handle)
for id in esearch_record['IdList']:
  esummary_handle = Entrez.esummary(db="assembly", id=id)
```

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

### Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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)
```

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

**Python example:** URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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']
```

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

**Python example:** URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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]
```

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

**Python example:** URL for the Acromyrmex assembly?

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

```python
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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')
```

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

## Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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('' + metadata_XML + '')
```

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

**Python example:** URL for the Acromyrmex assembly?

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

```python
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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('' + metadata_XML + '')
  for entry in Metadata[1]:
```

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

## Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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('' + metadata_XML + '')
  for entry in Metadata[1]:
    print entry.text
```

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

## Python example: URL for the Acromyrmex assembly?

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

```
import xml.etree.ElementTree as ET
from Bio import Entrez
Entrez.email = "mail@bmpvieira.com"
esearch_handle = Entrez.esearch(db="assembly", term="Acromyrmex")
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('' + metadata_XML + '')
  for entry in Metadata[1]:
    print entry.text
```

## Solution: bionode-ncbi

## Better way with Bionode - 4 approaches

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

## Better way with Bionode - 4 approaches

JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

```
//Pipe pattern
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

```
//Pipe pattern
var tool = require('tool-stream')
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

```
//Pipe pattern
var tool = require('tool-stream')
bio.ncbi.urls('assembly', 'Acromyrmex')
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

```
//Pipe pattern
var tool = require('tool-stream')
bio.ncbi.urls('assembly', 'Acromyrmex')
.pipe(tool.extractProperty('genomic.fna'))
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

```
//Pipe pattern
var tool = require('tool-stream')
bio.ncbi.urls('assembly', 'Acromyrmex')
.pipe(tool.extractProperty('genomic.fna'))
.pipe(process.stdout)
```

## Better way with Bionode - 4 approaches

### JavaScript

```
var bio = require('bionode')
```

```
//Callback pattern
bio.ncbi.urls('assembly', 'Acromyrmex', function(urls) {
  console.log(urls[0].genomic.fna)
})
```

```
//Event pattern
bio.ncbi.urls('assembly', 'Acromyrmex').on('data', printGenomeURL)
function printGenomeURL(url) {
  console.log(url.genomic.fna)
}
```

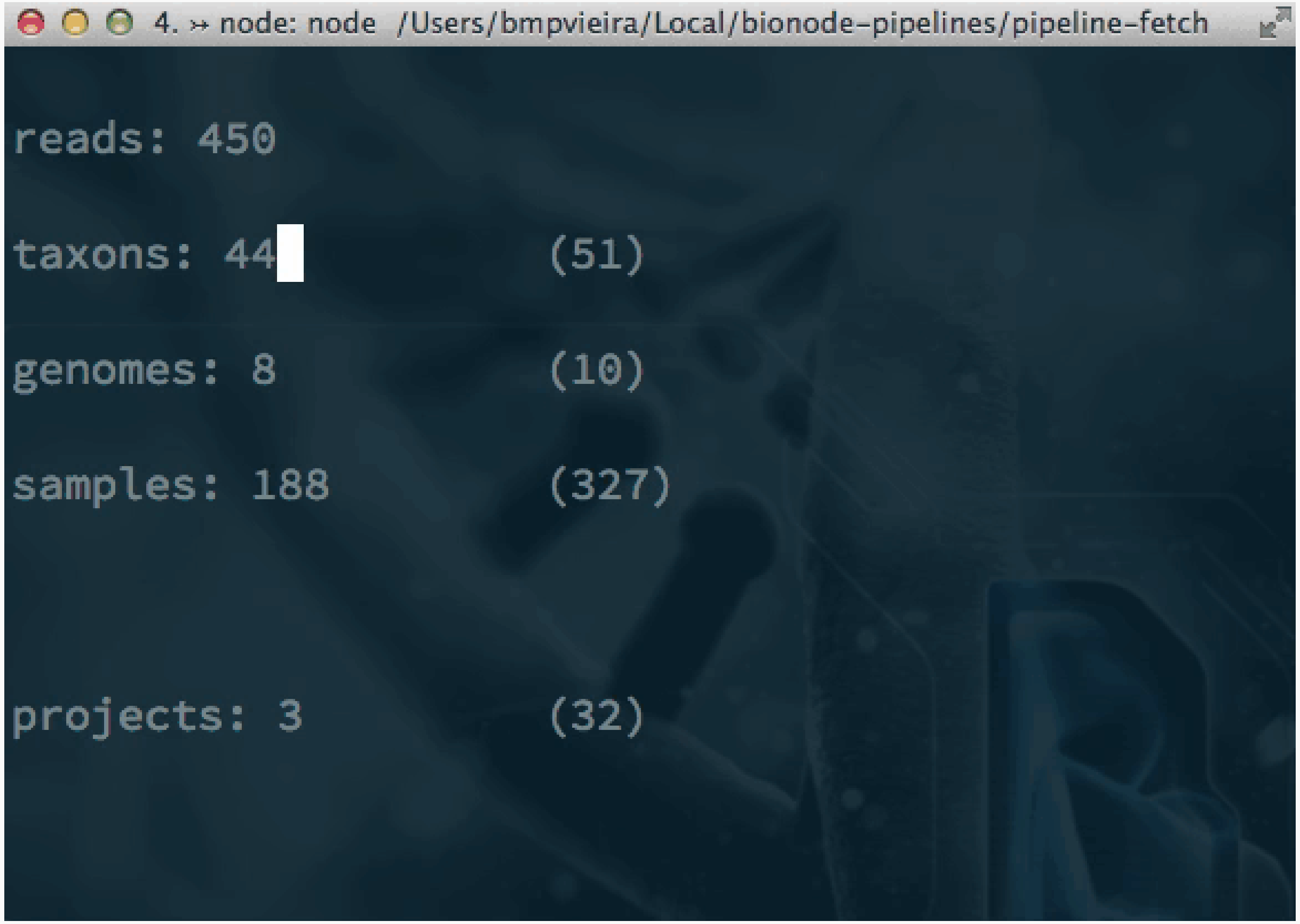
```
//Pipe pattern
var tool = require('tool-stream')
bio.ncbi.urls('assembly', 'Acromyrmex')
.pipe(tool.extractProperty('genomic.fna'))
.pipe(process.stdout)
```

### BASH

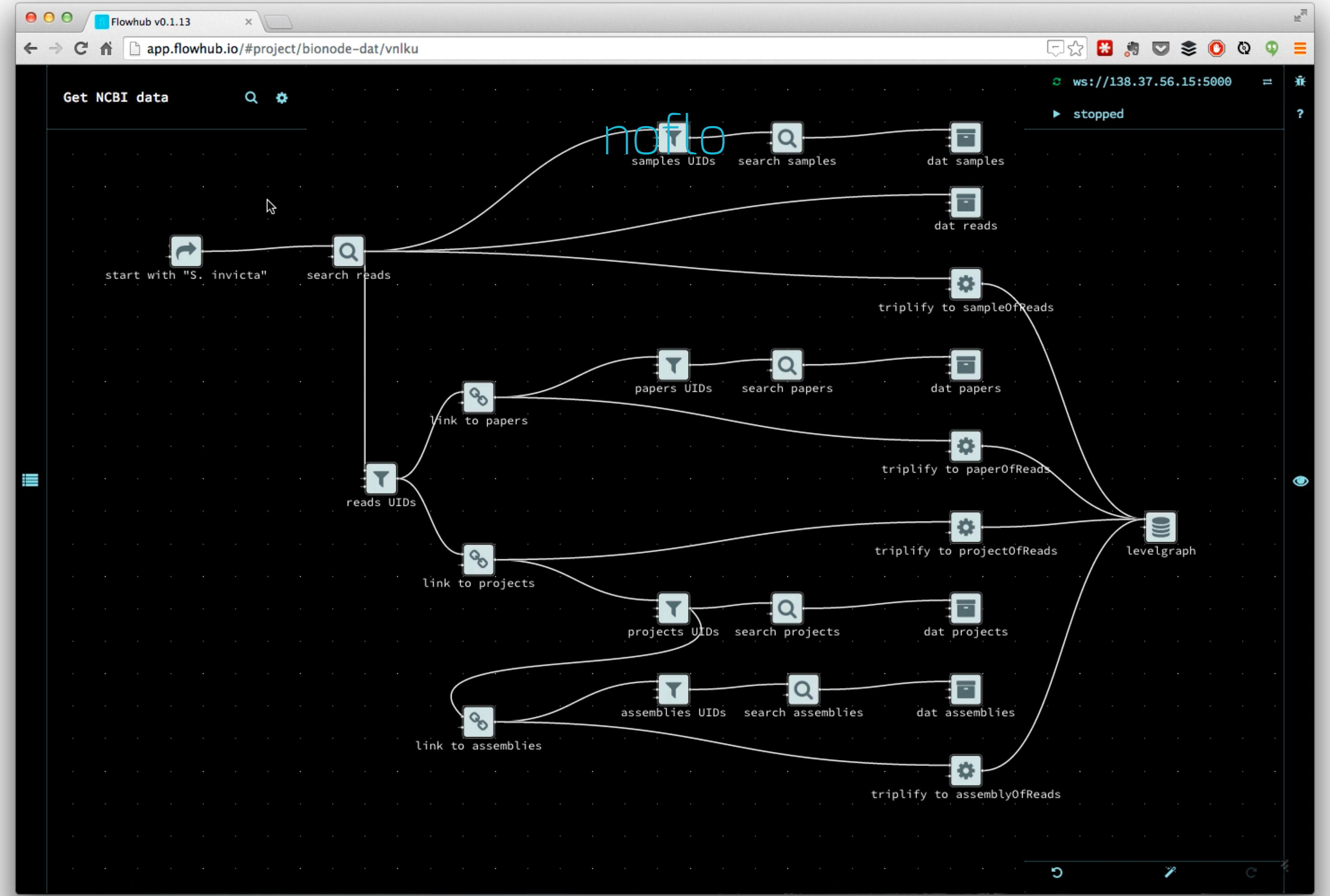
```
bionode ncbi urls assembly Acromyrmex | tool-stream extractProperty genomic.fna
```

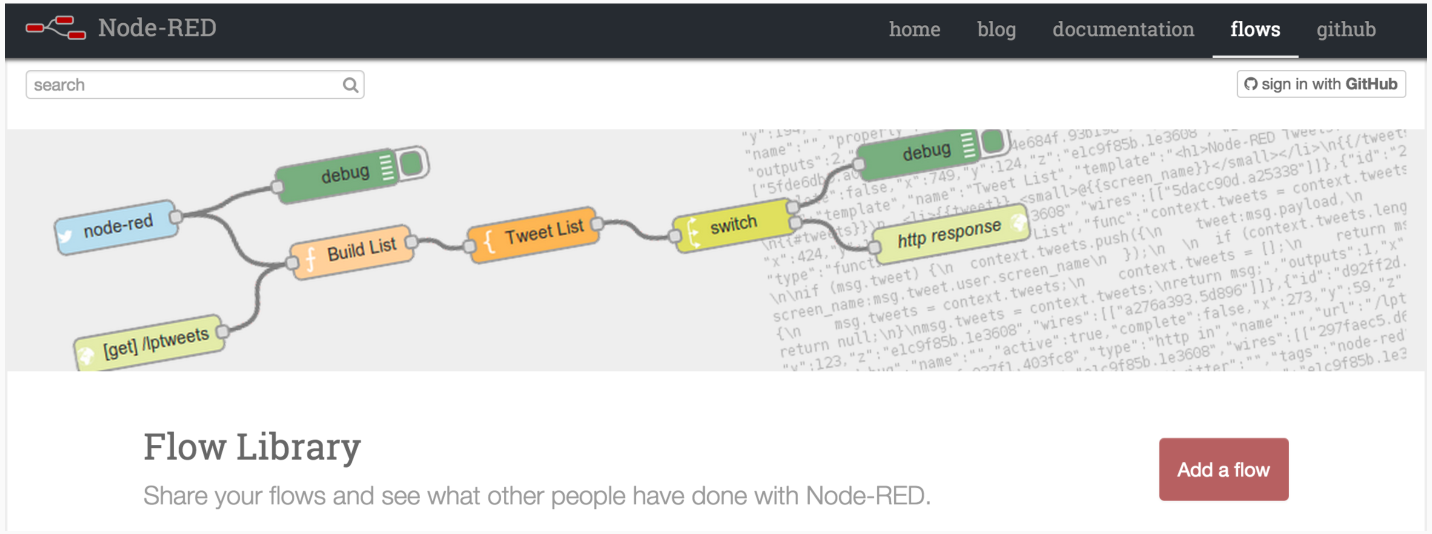
## Complex pipelines with forks

```
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'))
.pipe(ncbi.search('pubmed'))
.pipe(fork2)
.pipe(dat.papers)
```


4. ›› node: node /Users/bmpvieira/Local/bionode-pipelines/pipeline-fetch

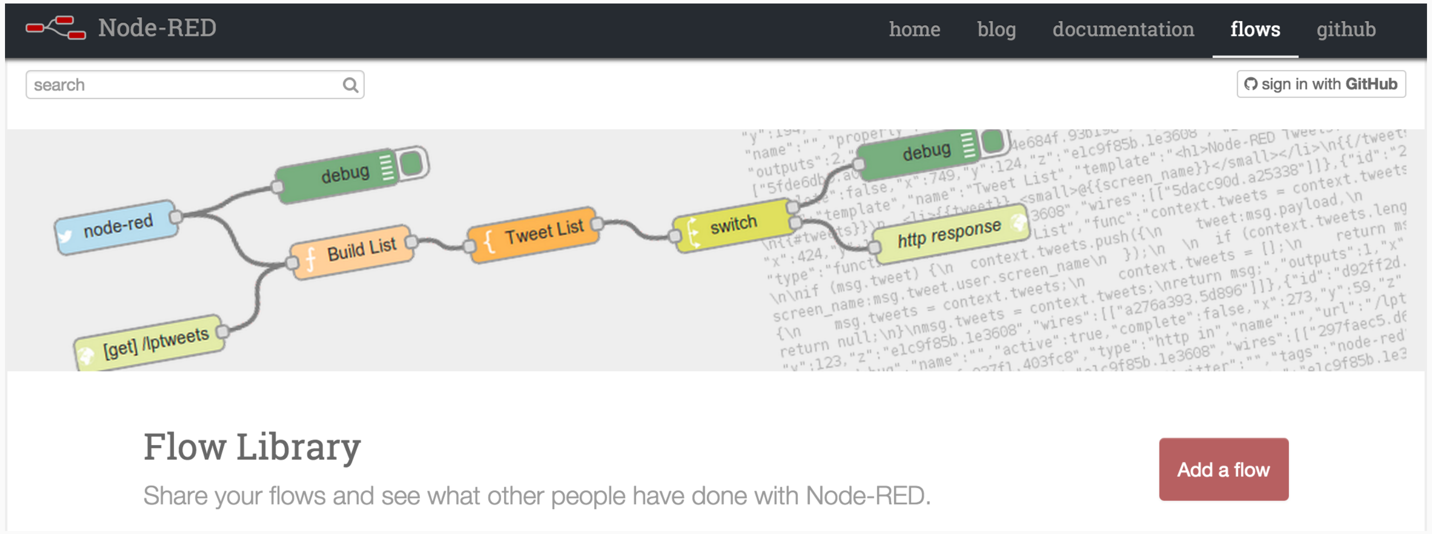
```
reads: 450

taxons: 44             (51)

genomes: 8             (10)

samples: 188           (327)



projects: 3            (32)
```


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



Flow Library

Share your flows and see what other people have done with Node-RED.


Node-RED home blog documentation flows github

search

sign in with GitHub

Flow Library

Share your flows and see what other people have done with Node-RED.

Add a flow

## Better pipeline representation format? Search GitHub for

- Gasket
- Datscript
- Hackfile

# Install Node.js and Bionode

```
# Mac
brew install node
```

```
# Ubuntu
sudo apt-get install npm
```

```
# Windows
Go to http://nodejs.org
```

```
# Manage versions and get latest stable
npm install -g n
n stable
```

## Online

try.bionode.io
bit.ly/try-dat

## Install bionode and json parser

```
npm install -g bionode-ncbi bionode-fasta json
```

```
bionode-ncbi search genome spiders
bionode-ncbi search genome spiders | wc
bionode-ncbi search genome spiders | head -n 1 | json
bionode-ncbi search genome spiders | json -ga organism_name
```

```
bionode-ncbi search genome spiders | \
  json -ga uid | \
    bionode-ncbi link genome pubmed - | \
      json -ga destUID | \
        bionode-ncbi search pubmed - | \
          json -ga title
```

```
bionode-ncbi download assembly Guillardia theta | \
  json -ga -c 'this.status === "completed"' | \
    json -ga path | \
      bionode-fasta -f | \
        json -ga -c 'this.seq.length > 10000' | \
          bionode-fasta --write > gtheta-big-scaffolds.fasta
```

## How to write a Stream?

```
var through = require('through2')
var stream = through2.obj(transform)
function transform (obj, enc, next) {
  // do things, example:
  obj.name = obj.name.toUpperCase()
  // Push downstream
  this.push(obj)
  // Callback to fetch next object
  next()
}
```

## Bash

```
mkdir project
cd project
npm install bionode-ncbi through2
```

## JavaScript

```
var ncbi = require('bionode-ncbi')
var through = require('through2')
var json = require('ndjson')

var myStream = through.obj(transform)
function transform (obj, enc, next) {
  var result = {
    specie: obj.organism,
    organisazation: obj.meta['submitter-organization']
  }
  this.push(result)
  next()
}

ncbi.search('assembly', 'spiders')
.pipe(myStream)
.pipe(json.stringify())
.pipe(process.stdout)
```

## JavaScript Events and Styles

```
var counter = 0
myStream
.on('data', function (data) {
  counter++
})
.on('end', function () {
  console.log('Processed ' + counter)
})
```

```
var counter = 0

var count = function (data) {
  counter++
}

var log = function () {
  console.log('Processed ' + counter)
}

myStream.on('data', count).on('end', log)
```

## Bionode

- bionode.io
- doc.bionode.io
- github.com/bionode/bionode
- twitter.com/bionode

## Hackday

- bit.ly/biocw15

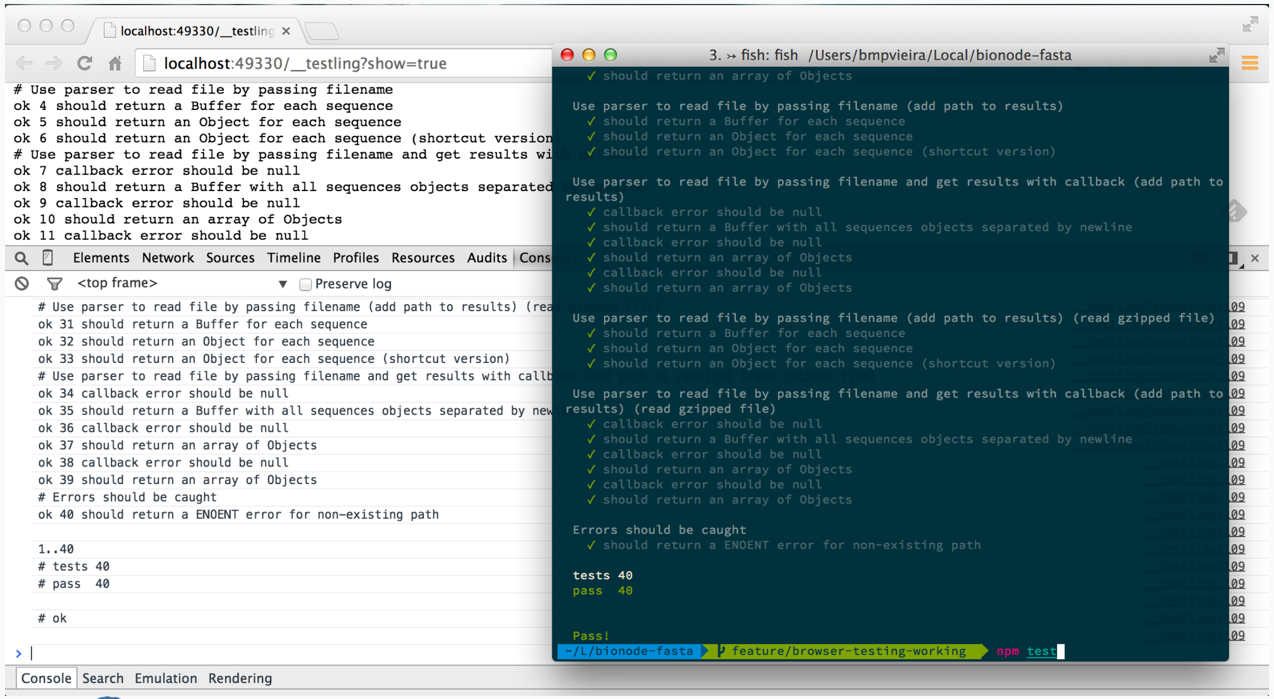
## Extra slides

## Bionode - Why wrappers?

- Same interface between modules (Streams and NDJSON)
- Easy installation with NPM
- Semantic versioning
- Add tests
- Abstract complexity / More user friendly

## Bionode - Why Node.js?

### Same code client/server side



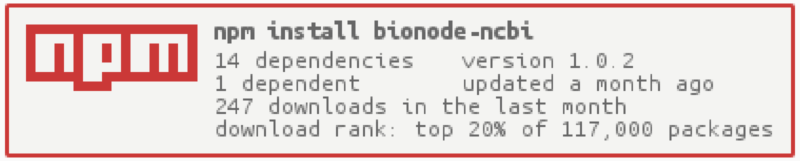
```
localhost:49330/__testling?show=true

# Use parser to read file by passing filename
ok 4 should return a Buffer for each sequence
ok 5 should return an Object for each sequence
ok 6 should return an Object for each sequence (shortcut version
# Use parser to read file by passing filename and get results wi
ok 7 callback error should be null
ok 8 should return a Buffer with all sequences objects separated
ok 9 callback error should be null
ok 10 should return an array of Objects
ok 11 callback error should be null
```

```
# Use parser to read file by passing filename (add path to results) (rea
ok 31 should return a Buffer for each sequence
ok 32 should return an Object for each sequence
ok 33 should return an Object for each sequence (shortcut version)
# Use parser to read file by passing filename and get results with callb
ok 34 callback error should be null
ok 35 should return a Buffer with all sequences objects separated by new
ok 36 callback error should be null
ok 37 should return an array of Objects
ok 38 callback error should be null
ok 39 should return an array of Objects
# Errors should be caught
ok 40 should return a ENOENT error for non-existing path

1..40
# tests 40
# pass  40

# ok
```

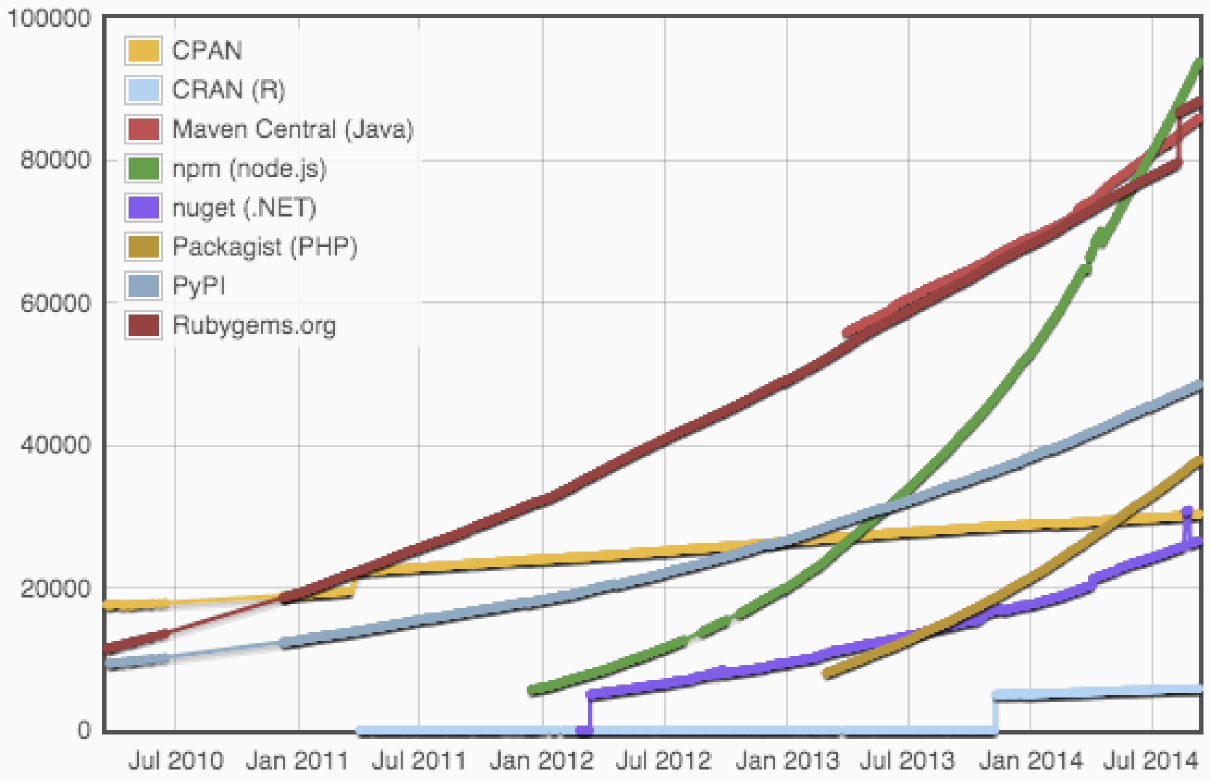
```
3. ↣ fish: fish  /Users/bmpvieira/Local/bionode-fasta

    ✓ should return an array of Objects

  Use parser to read file by passing filename (add path to results)
    ✓ should return a Buffer for each sequence
    ✓ should return an Object for each sequence
    ✓ should return an Object for each sequence (shortcut version)

  Use parser to read file by passing filename and get results with callback (add path to
 results)
    ✓ callback error should be null
    ✓ should return a Buffer with all sequences objects separated by newline
    ✓ callback error should be null
    ✓ should return an array of Objects
    ✓ callback error should be null
    ✓ should return an array of Objects

  Use parser to read file by passing filename (add path to results) (read gzipped file)
    ✓ should return a Buffer for each sequence
    ✓ should return an Object for each sequence
    ✓ should return an Object for each sequence (shortcut version)

  Use parser to read file by passing filename and get results with callback (add path to
 results) (read gzipped file)
    ✓ callback error should be null
    ✓ should return a Buffer with all sequences objects separated by newline
    ✓ callback error should be null
    ✓ should return an array of Objects
    ✓ callback error should be null
    ✓ should return an array of Objects

  Errors should be caught
    ✓ should return a ENOENT error for non-existing path

  tests 40
  pass  40


  Pass!
 ~/L/bionode-fasta  feature/browser-testing-working  npm test
```

browserify

## Bionode - Why Node.js?



```
npm install bionode-ncbi
14 dependencies    version 1.0.2
1 dependent        updated a month ago
247 downloads in the last month
download rank: top 20% of 117,000 packages
```


CPAN
CRAN (R)
Maven Central (Java)
npm (node.js)
nuget (.NET)
Packagist (PHP)
PyPI
Rubygems.org
Jul 2010
Jan 2011
Jul 2011
Jan 2012
Jul 2012
Jan 2013
Jul 2013
Jan 2014
Jul 2014