Network Algorithms Research in Julia

Follow along code http://github.com/nassarhuda/juliaCon2016PresFiles

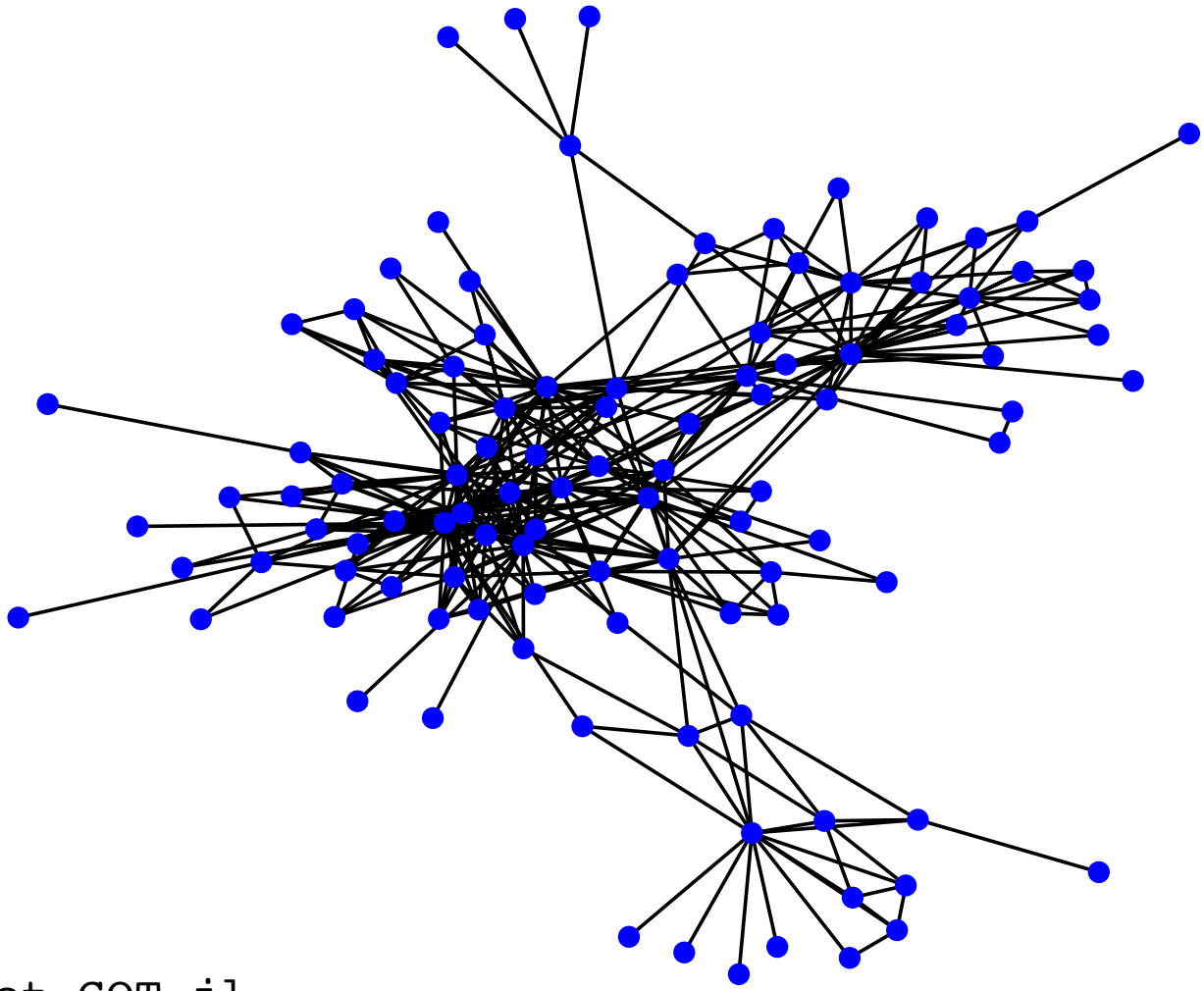
Huda Nassar
Computer Science
Purdue

Thanks for the support JuliaCon!

with David F. Gleich
Computer Science
Purdue

supported by
DARPA SIMPLEX
and NSF

There are many things we want to compute about networks



- Important nodes
- Communities and clusters
- Label inference
- Alignments …

`plot_GOT.jl`

Many operations we want to perform on networks are (in fact) linear algebra

Network Operation	Matrix Operation
Ranking (eg. PageRank)	$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = (1-\alpha)\mathbf{v}$
Network Alignment (eg. IsoRank)	$\mathbf{x} = \alpha(\boldsymbol{A} \otimes \boldsymbol{B})\mathbf{x} + (1-\alpha)\mathbf{h}$
Diffusion (eg. HeatKernel)	$\mathbf{x} = \exp\left(-t(\boldsymbol{I} - \boldsymbol{P})\right)\mathbf{s}$
Community Detection (eg. Random walks)	$\mathbf{x} = \boldsymbol{P}^k\mathbf{s}$
Clustering (eg. Spectral Clustering)	$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$

These give simple Julia implementations

```
x = (eye(n) - alpha*P) \ (1-alpha)*v  # PageRank
x = expm(-t*(eye(n)-P))*v             # Heat Kernel
```

But there (are usually) faster ways to compute certain ones

Finding them is the mission of our research team!

Heat kernel communities

(Kloster & Gleich, KDD 2014)

$$O(n^3) \underbrace{\rightarrow}_{\text{sparse iterative}} O(\text{nnz}) \underbrace{\rightarrow}_{us} O(1/\varepsilon)$$

But the implementations usually involve complicated and delicate algorithmic details.

- Queues, Heaps, sparse data-structures, etc.

MATLAB + mex was our team's previous solution.

… it was a common source of frustration, bugs and wasted time ☹

(ask for details!)

But then, this happened

David Gleich

To: Huda Nassar

Re: Meeting on Friday?

April 22, 2015 at Wednesday, April 22, 2015,9:08 PM

Inbox - Exchange

DG

Let's meet at 12:30pm on Friday

Before the meeting, can you try and install the Julia language (from source? www.julialang.org) on your mac and see if you can get the following Matlab code ported to Julia? The two languages are fairly similar.

In julia, fprintf(...) -> @printf(...)

David

But then, this happened

Huda Nassar
To: David Gleich
Re: Meeting on Friday?

April 22, 2015 at Wednesday, April 22, 2015,9:11 PM
Sent Messages

HN

Ok sounds good. I will get that to work before our meeting on Friday.

See you then,
Huda

See More from David Gleich

David Gleich
To: Huda Nassar
Re: Meeting on Friday?

April 22, 2015 at Wednesday, April 22, 2015,9:08 PM
Inbox - Exchange

DG

Let's meet at 12:30pm on Friday

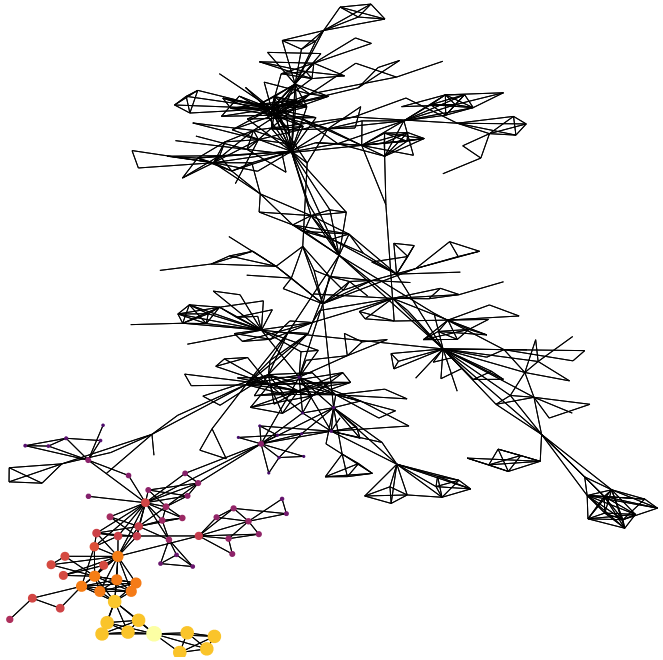
Before the meeting, can you try and install the Julia language (from source? www.julialang.org) on your mac and see if you can get the following Matlab code ported to Julia? The two languages are fairly similar.

In julia, fprintf(...) -> @printf(...)

David

Then I was the Julia guinea pig!

My project, generate really large graphs and compute seeded PageRank



For seeded PageRank, the random walk always restarts on the same node. The solution is mathematically non-zero but practically "zero" on most of the nodes. Q: How many "non-zero"s?

My task

1. Generate a random graph with a power law degree distribution using a C program
2. Compute seeded PageRank
3. Sort vector and compute number of large entries.

Strong Localization in Personalized PageRank
Nassar, Kloster, Gleich, WAW2015
Localization in seeded PageRank (submitted)

What happened

My advisor's (quick) solution for Matlab
run C code, output graph, read in Matlab, compute PageRank, etc.

I quickly ported *that* to Julia (it was easy and fun)!

But it took HOURS to read in graphs with billions of edges.

Then I wrote a wrapper to directly call the C function, which was super fast, easy and fun!

What didn’t work

Around early-mid 2015

Somewhere between Julia 0.3 and 0.4

- Plotting didn’t work reliably between our linux server and OSX dev settings. (All good in 2016, thanks `Plots.jl`!)
- We never found `At_mul_B!(y,P,x)` for sparse-matrix-transpose times vector (and implemented it ourselves)

Project 1, a success!

Project 2 … more graph stuff to Julia!

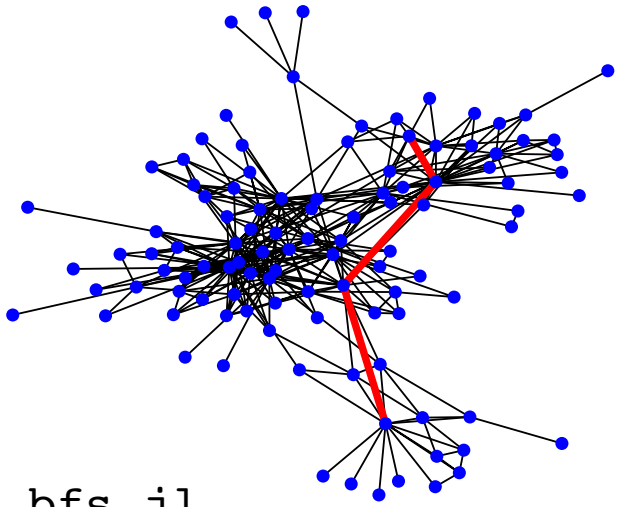
MatrixNetworks.jl

Treat a matrix like a network and a network like a matrix. (Our team's mission is to exploit structure in this setting!)

- Our package to manage common implementation primitives (connected components, shortest paths, etc.)
- And provide "production ready" algorithms (when we get them, i.e. no research code)

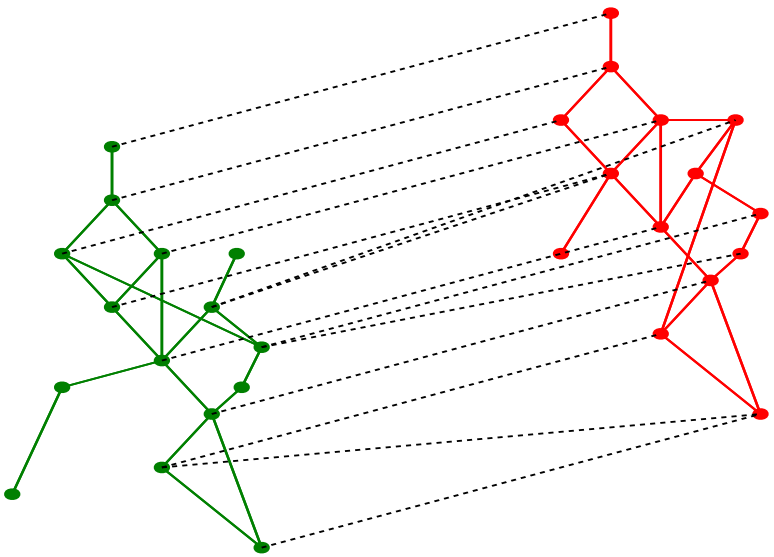
Stuff in MatrixNetworks.jl

- Basics: Connected Components, Paths, etc.
- Diffusions (PageRank + Heat Kernel)
- Spectral clustering (custom ARPACK wrapper for type stability)
- (Soon) Network Alignment!



`plot_bfs.jl`

My current research project
Network alignment



What is the best way of matching graph A (green) to graph B (red) using edges in L (dashed lines)?

`plot_netalign.jl`

- Need to implement various state-of-the-art algorithms
- One problem boils down to solving a bipartite matching problem on each row of a matrix. Previous practice involved using a mex compiled code.

We faced some issues while writing all this code in Julia – Features we wish exist

- Pkg.analyze() – compile code and throw messages on type instability and incompatibility
- More indicative error messages
- Faster plotting & graphics could still be easier. Plotting 1M datapoints shouldn't require detailed knowledge.

We love many things about Julia and are going to keep using it ☺

- The Julia community is very helpful and encouraging
- They actually answers questions!
- Performance
- Reproducibility of code is very crucial for us and Julia comes to the rescue!

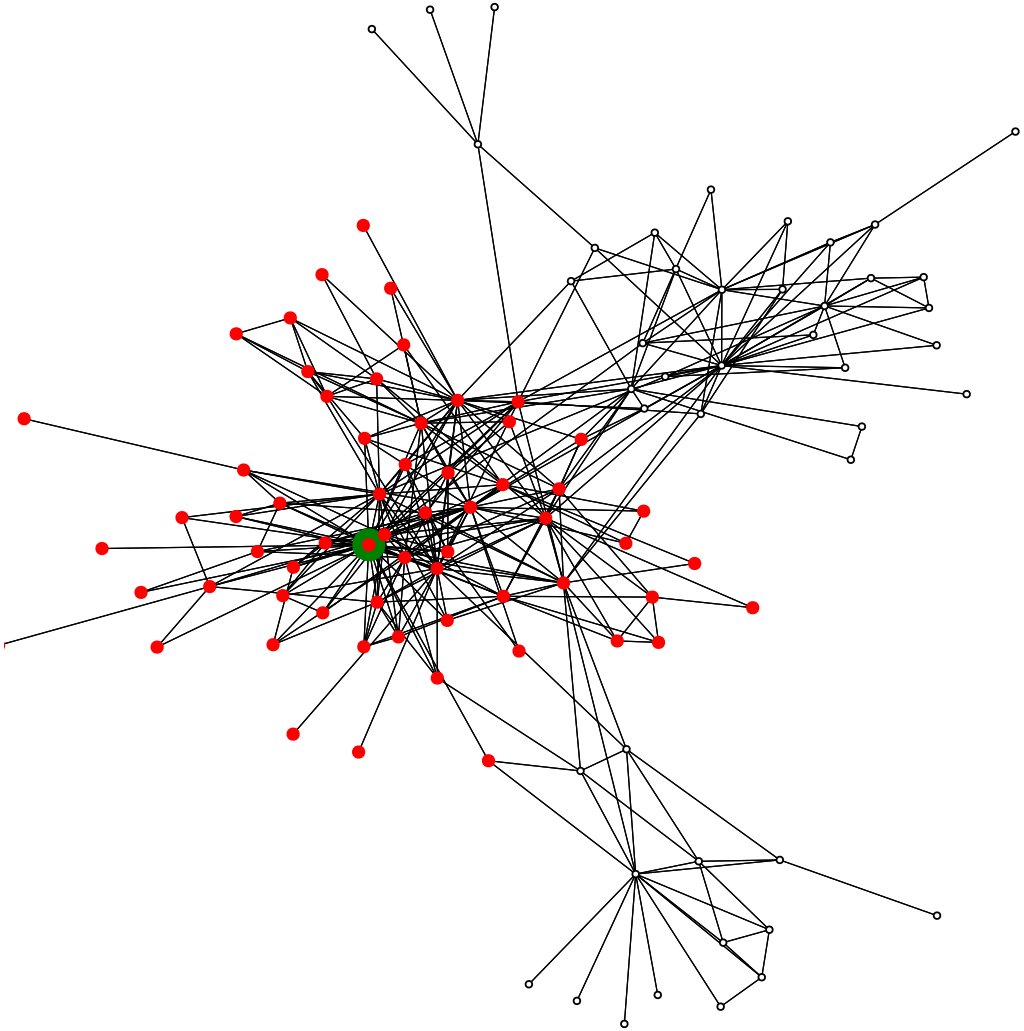
Thank you!

Links that might be helpful:

http://github.com/nassarhuda/MatrixNetworks.jl

http://github.com/nassarhuda/juliaCon2016PresFiles

Tyrion's seeded PageRank



`plot_Tyrion_personalized.jl`

MATLAB + mex was our solution

- Interface is harder
- Weak compatibility of older codes with new versions
- Harder to try out ideas and analyze intermediate results
- Inconsistent indexing
- Common source of frustration, bugs and wasted time ☹