



Graph Maintenance

—

Cytoscape Integration



Key Deliveries

- Extending graph support adding during independent study
 - New file input formats HIPPIE and MatSim
 - add and remove vertices and edges
 - addition of novel file format supporting group in Spring
- Writing a plugin for Cytoscape to import MASS graph for visualization
 - Retrieve MASS in-memory graph from Cytoscape

What is “graph” support?

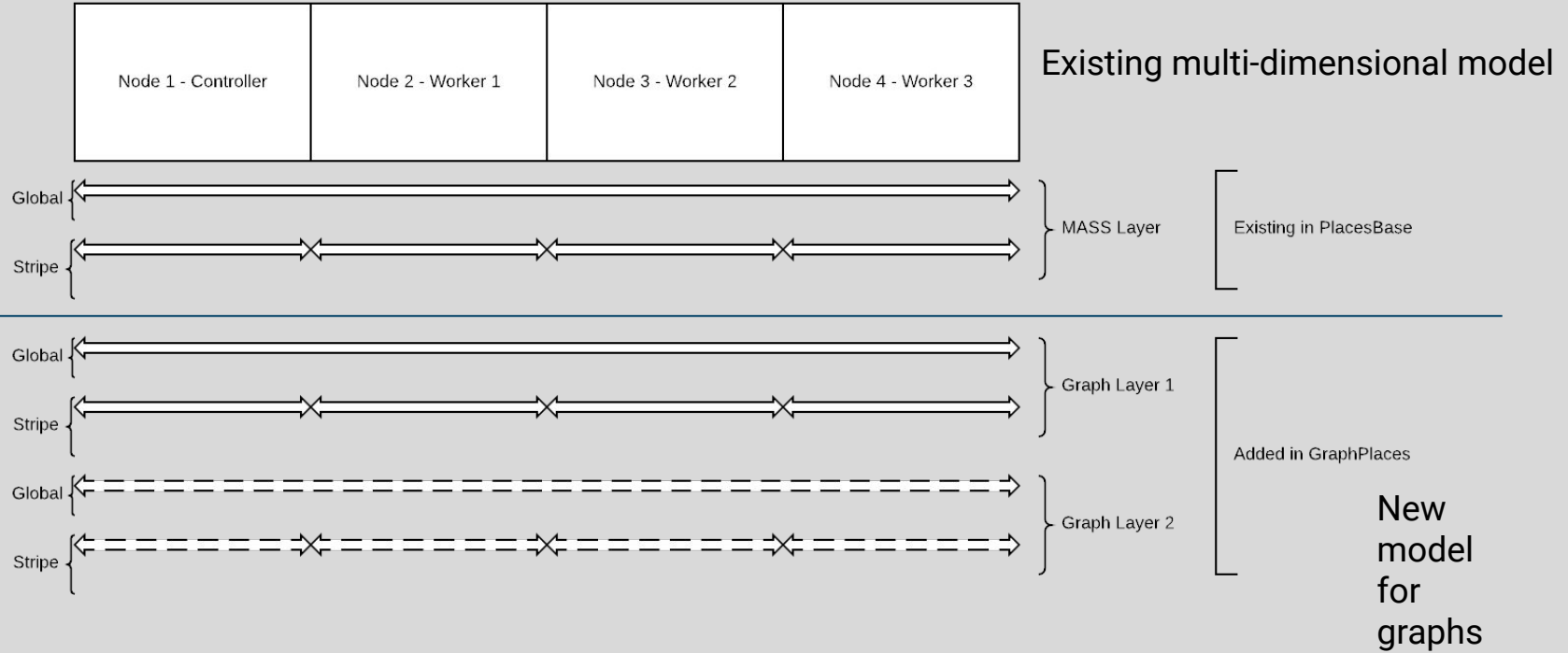
Multi-dimensional model

- Data is chunked to each node in stripes
- Each Place owns an upper and lower boundary of data within the entire dataset
- Outside of this must be exchanged or retrieved between places

Graph model

- Added an additional data structure within GraphPlaces to hold additional layers of data
- Each node holds a slice of all of the vertices
- Communication is direct between vertices (no sense of a ‘boundary’)
- ExchangeAll is performed on logical neighbors instead of data ‘boundaries’

MASS Data Models



MatSim XML

```
<network>
  <nodes>
    <node id="1" x="665498.5915828889" y="6600721.058093354" ></node>
    <node id="10" x="658803.7360160261" y="6575644.324923517"></node>
    <node id="100" x="566282.7735573719" y="6454558.626888261"></node>
    <node id="10000" x="408774.9462956431" y="6233917.802557087"></node>
    <node id="10001" x="408772.02211252623" y="6234167.648466459"></node>
    ...
  </nodes>
  <links>
    ...
    <link id="pt_99992" from="pt_740057809" to="pt_740057653"
length="257.27792215211286" freespeed="8.333333333333334" capacity="500.0"
permLanes="1.0" oneway="1" modes="pt" > </link>
    <link id="pt_99996" from="pt_740057599" to="pt_740057756"
length="157.21884660387735" freespeed="8.333333333333334" capacity="500.0"
permLanes="1.0" oneway="1" modes="pt" ></link>
    <link id="pt_99997" from="pt_740057688" to="pt_740057647"
length="207.21244492773178" freespeed="8.333333333333334" capacity="500.0"
permLanes="1.0" oneway="1" modes="pt" ></link>
    <link id="pt_99998" from="pt_740057647" to="pt_740057623"
length="378.77959250564095" freespeed="8.333333333333334" capacity="500.0"
permLanes="1.0" oneway="1" modes="pt" ></link>
    <link id="pt_99999" from="pt_740057623" to="pt_740057686"
length="292.07787848782823" freespeed="8.333333333333334" capacity="500.0"
permLanes="1.0" oneway="1" modes="pt" ></link>
  </links>
</network>
```

- XML format
- Technically multiple files. Using the network only
 - Additional file for vehicle types
- Includes edge list and vertex definitions
- Ids are 1 based sequential
- Used a combination of built-in Java XML Reader and XPath queries

Neighbors: `"/network/links/link[@from="" + (index + 1) + "]"`

HIPPIE Tab

- Tab separated values
- Protein complex string and non-sequential integer id
- Interaction value/weight
- Free form metadata field
- Required map from logical id to global index

```
AL1A1_HUMAN 216 AL1A1_HUMAN 216 0.76 experiments:in vivo,Two-hybrid;pmids:12081471,16189514,25416956;sources:HPRD,BioGRID,IntAct,MINT,I2D,Rual05
ITA7_HUMAN 3679 ACHA_HUMAN 1134 0.73 experiments:in vivo,Affinity Capture-Western,affinity chromatography technology;pmids:10910772;sources:HPRD,BioGRID,I2D
NEB1_HUMAN 55607 ACTG_HUMAN 71 0.65 experiments:in vitro,in vivo;pmids:9362513,12052877;sources:HPRD
SRGN_HUMAN 5552 CD44_HUMAN 960 0.63 experiments:in vivo;pmids:9334256,16189514,16713569;sources:HPRD,I2D,Rual05,Lim06
PAK1_HUMAN 5058 ERBB2_HUMAN 2064 0.73 experiments:in vivo,Affinity Capture-Western,affinity chromatography technology;pmids:9774445;sources:HPRD,BioGRID,I2D,STRING
DLG4_HUMAN 1742 ERBB2_HUMAN 2064 0.87 experiments:in vivo,Two-hybrid,Affinity Capture-Western,Co-fractionation,affinity chromatography technology;pmids:10839362,16713569;sources:HPRD,BioGRID,I2D,Lim06
P85B_HUMAN 5296 ERBB2_HUMAN 2064 0.89 experiments:in vivo,Reconstituted Complex,Biochemical Activity,protein array,pull down,enzymatic study;pmids:1334406,16273093,16729043;sources:HPRD,BioGRID,MINT,I2D,IntAct,KEGG,STRING
PTN18_HUMAN 26469 ERBB2_HUMAN 2064 0.88 experiments:in vitro,pull down,anti tag coimmunoprecipitation,x-ray crystallography,phosphatase assay;pmids:14660651,25081058;sources:HPRD,I2D,IntAct
SMUF2_HUMAN 64750 RHG05_HUMAN 394 0.88 experiments:Two-hybrid,affinity chromatography technology;pmids:15231748,28514442;species:Mus musculus (Mouse);sources:HPRD,MINT,I2D,Colland04,IntAct,BioGRID
UBX11_HUMAN 91544 ZFYV9_HUMAN 9372 0.73 experiments:Two-hybrid;pmids:16189514,15231748;sources:HPRD,MINT,I2D,Rual05,Colland04,IntAct,BioGRID
NCTR1_HUMAN 9437 CD59_HUMAN 966 0.73 experiments:in vivo,Affinity Capture-Western,affinity chromatography technology;pmids:14635045;sources:HPRD,BioGRID,I2D
LYN_HUMAN 4067 PP1R8_HUMAN 5511 0.67 experiments:in vitro,Biochemical Activity,enzymatic study;pmids:11104670;sources:HPRD,BioGRID,I2D
NPHN_HUMAN 4868 LYN_HUMAN 4067 0.52 experiments:in vivo;pmids:12846735;sources:HPRD,I2D
DLG4_HUMAN 1742 LYN_HUMAN 4067 0.59 experiments:in vivo;pmids:9892651;species:Rattus norvegicus (Rat);sources:HPRD,I2D
BCAR1_HUMAN 9564 LYN_HUMAN 4067 0.9 experiments:in vitro,in vivo,Affinity Capture-Western,affinity chromatography technology;pmids:9581808,9020138;species:Mus musculus (Mouse);sources:HPRD,BioGRID,HomoMINT,I2D
U119A_HUMAN 9094 LYN_HUMAN 4067 0.79 experiments:in vitro,in vivo,Reconstituted Complex,Affinity Capture-Western,affinity chromatography technology,pull down;pmids:12496276;sources:HPRD,BioGRID,I2D
TRAT1_HUMAN 50852 LYN_HUMAN 4067 0.85 experiments:in vitro,Reconstituted Complex,pull down;pmids:9687533,10790433;sources:HPRD,BioGRID,I2D
SKAP1_HUMAN 8631 LYN_HUMAN 4067 0.75 experiments:in vitro,Reconstituted Complex,pull down;pmids:9195899;sources:HPRD,BioGRID,I2D
SKAP2_HUMAN 8935 LYN_HUMAN 4067 0.8 experiments:in vivo,Affinity Capture-Western,affinity chromatography technology;pmids:9837776;species:Mus musculus (Mouse);sources:HPRD,BioGRID,HomoMINT,I2D
LYN_HUMAN 4067 TRPV4_HUMAN 59341 0.77 experiments:in vitro,in vivo,Biochemical Activity,Affinity Capture-Western,enzymatic study,affinity chromatography technology;pmids:12538589;sources:HPRD,BioGRID,I2D
NCTR3_HUMAN 259197 CD59_HUMAN 966 0.74 experiments:in vitro,in vivo,Affinity Capture-Western,affinity chromatography technology;pmids:14635045;sources:HPRD,BioGRID,I2D
```

Graph addressing

Vertex attribute: "Hello"

Global index: 362

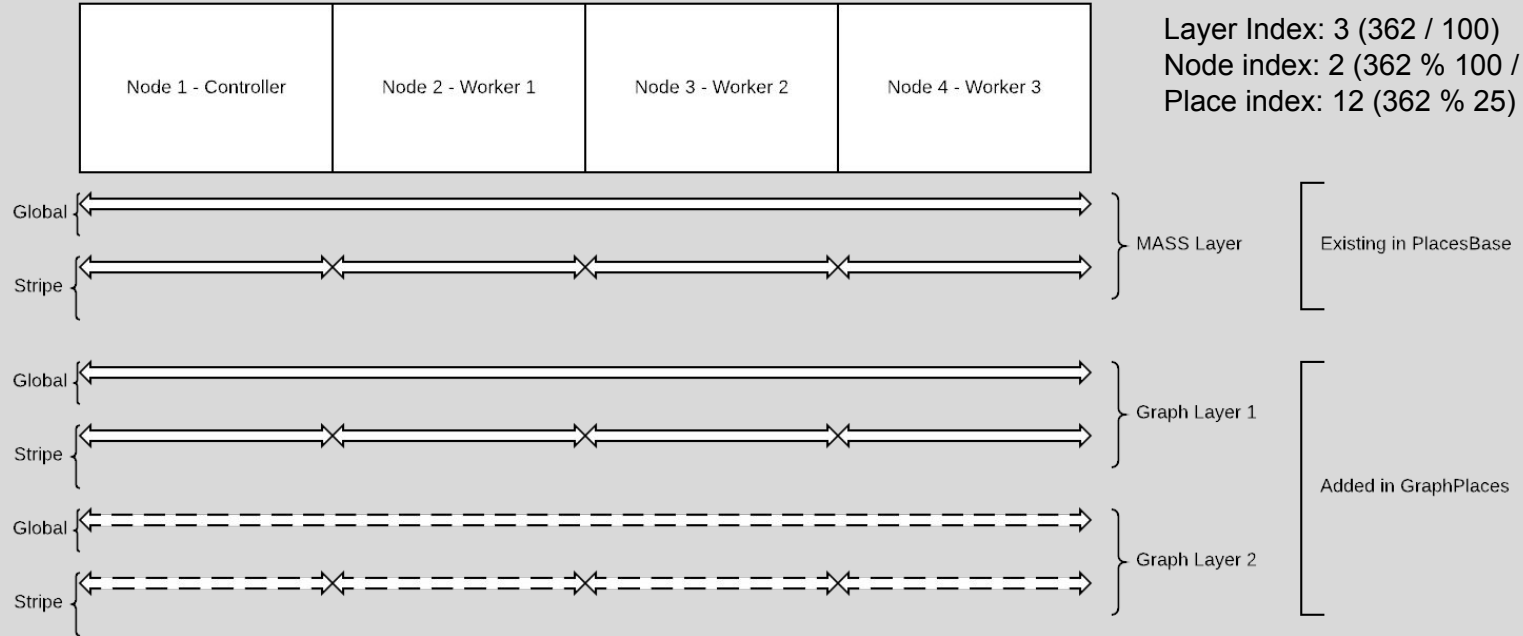
MASS Size: 100

Chunk size: 25

Layer Index: 3 ($362 / 100$)

Node index: 2 ($362 \% 100 / 25$)

Place index: 12 ($362 \% 25$)



Graph Maintenance Tasks

- Implement add & remove vertices
- Implement add & remove edges
- Serialize the data model to send to cytoscape

Maintenance - Vertices

Add

- Determine the new owner
 - controller maintains serial id
 - add we are the owner
 - or send new message type 'MAINTENANCE_ADD_PLACE' to the owner

Remove

- call remove locally
- send new message type 'MAINTENANCE_REMOVE_PLACE' to *all* nodes

Maintenance - Edge

Add

- Determine if source and neighbor vertices exist
- Find the owner of the source vertex
- Send new message type 'MAINTENANCE_ADD_EDGE'

Remove

- Find owner of source vertex
- Send new message type 'MAINTENANCE_REMOVE_EDGE'

Maintenance Example

- Basic constructor for empty graph
- Create vertices and edges
- CytoscapeListener for communication with Cytoscape
- pause to allow connection via Cytoscape

```
fun main() {  
    MASS.setLoggingLevel(LogLevel.DEBUG)  
    MASS.init(1000000)  
  
    val places = GraphPlaces(0, Node::class.qualifiedName, 100)  
  
    places.addVertex(1001)  
    places.addVertex(10001)  
  
    places.addEdge(1001, 10001, 0.9)  
    places.addEdge(10001, 1001, 0.5)  
  
    CytoscapeListener(places)  
  
    MASS.pause()  
  
    MASS.finish()  
  
}
```

Maintenance - Serialization

- Java Object Serialization with ObjectOutputStream
- Lightweight GraphModel for storing the high level structure of the graph
- Lightweight VertexModel to store the vertex and neighbor information
- Converge all onto the controller node
- Send the full model across the wire to Cytoscape

Initial Cytoscape Plugin

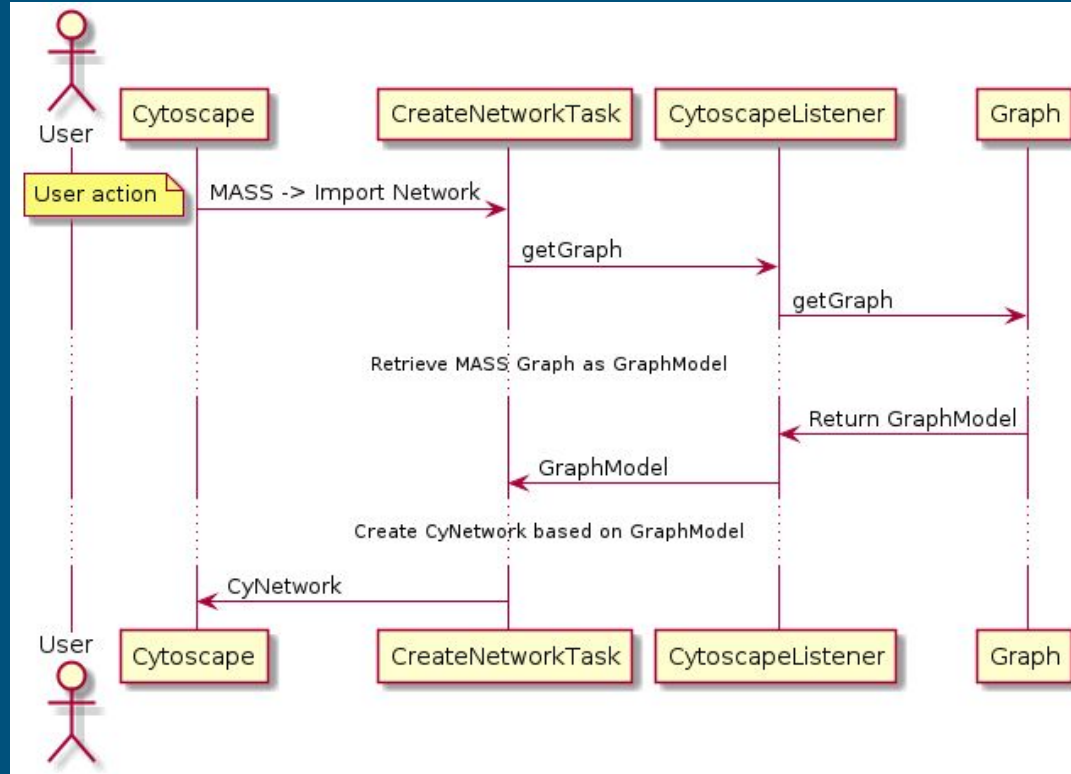
Cytoscape Integration - Part 1

- Read graph from MASS
- Node & edge styling
- Create new nodes for non-existent neighbors

CytoscapeListener - MASS to Cytoscape

- A new Thread with a socket listening for requests from Cytoscape
- Dispatches requests to a Graph interface such as getGraph
- getGraph sends a 'MAINTENANCE_GET_PLACES' request to all nodes
- Populates a GraphModel of the local nodes vertices
- Merges response GraphModel from each node into a single GraphModel
- Send the complete model to Cytoscape

Data flow of MASS -> Import Network



GraphModel to Cytoscape CyNetwork

- Create CyNodes to represent vertices
- Table row holds information like vertex name
- Create CyEdge for each neighbor
- Add attributes to the CyRow for the edge metadata like weight

```
// Create an empty network
CyNetwork cyNetwork = cnf.createNetwork();

List<VertexModel> vertices = graph.getVertices();

CyNode [] nodes = new CyNode[vertices.size()];

// Add vertices to the network
for (int v = 0; v < vertices.size(); v++) {
    VertexModel vertex = vertices.get(v);
    nodes[v] = cyNetwork.addNode();
    // set name for new vertex
    cyNetwork.getDefaultNodeTable().getRow(nodes[v].getSUID()).set("name", vertex.id);
}

// Add edges after all vertices are created
for (int v = 0; v < vertices.size(); v++) {
    VertexModel vertex = vertices.get(v);

    for (Object neighbor : vertex.neighbors.stream()
        .filter(n -> !n.equals(vertex.id)).collect(Collectors.toList())) {
        CyNode neighborNode = nodesMap.get(neighbor);
        CyEdge edge = cyNetwork.addEdge(nodes[v], neighborNode, true);
        CyRow edgeRow = cyNetwork.getDefaultEdgeTable().getRow(edge.getSUID());
        edgeRow.set("name", v + ":" + neighbor);
        edgeRow.set("interaction", vertex.id + "-->" + neighbor);
    }
}
```

Limitations

- Layout code is not intuitive
 - User must layout manually (F5 is quickest)
- Graph is collected on master
- Large graph such as HIPPIE_CURRENT is very sluggish
 - appears to be on Cytoscape side
- Client library configuration not obvious after the move to Java 11
 - Cytoscape is 1.8

Future Work and Enhancements

- Streaming/partial loading of a graph in Cytoscape could improve intractability
- Make MASS host and port configurable from within Cytoscape
- Send jobs from Cytoscape to run on MASS
- Configurable styling e.g:
 - weight based coloring
 - attribute based node style
- Move to JSON serialization



Enter search term...



Control Panel

Network Style Select Annotation

Enter search terms for NDEx...



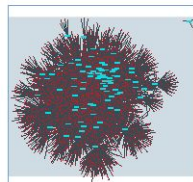
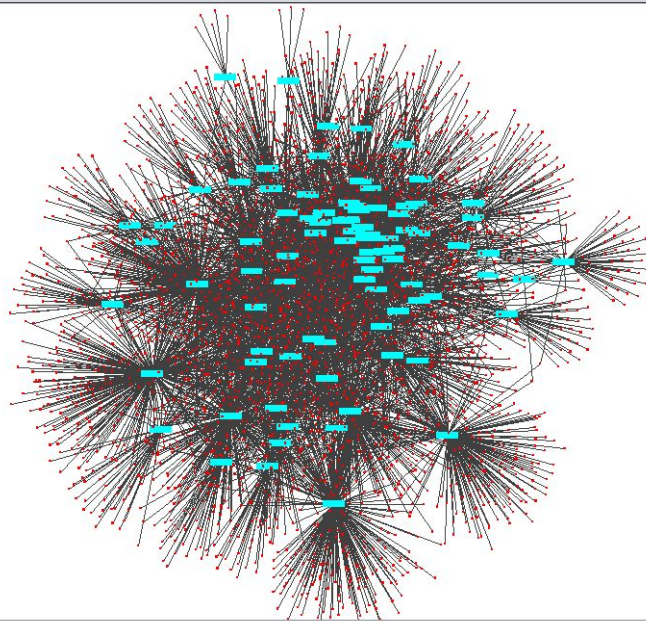
1 of 1 Network selected

Network

Network

3164 5531

HIPPIE
CURRENT
data set with
neighbors in
RED



Network

Table Panel

shared name	name	agents
Vertex AL1A1_HUMAN	Vertex AL...	[0, 1, 2, 3, 4, ...
Vertex ITA7_HUMAN	Vertex ITA...	[0, 1, 2, 3, 4, ...
Vertex NEB1_HUMAN	Vertex NE...	[0, 1, 2, 3, 4, ...
Vertex SRGN_HUMAN	Vertex SR...	[0, 1, 2, 3, 4, ...
Vertex GRB7_HUMAN	Vertex GR...	[0, 1, 2, 3, 4, ...
Vertex PAK1_HUMAN	Vertex PA...	[0, 1, 2, 3, 4, ...
Vertex DLG4_HUMAN	Vertex DL...	[0, 1, 2, 3, 4, ...
Vertex P85B_HUMAN	Vertex P8...	[0, 1, 2, 3, 4, ...
Vertex PTM1R_HUMAN	Vertex PT...	[0, 1, 2, 3, 4, ...

Node Table Edge Table Network Table



Enter search term...

Control Panel

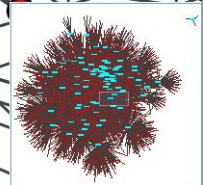
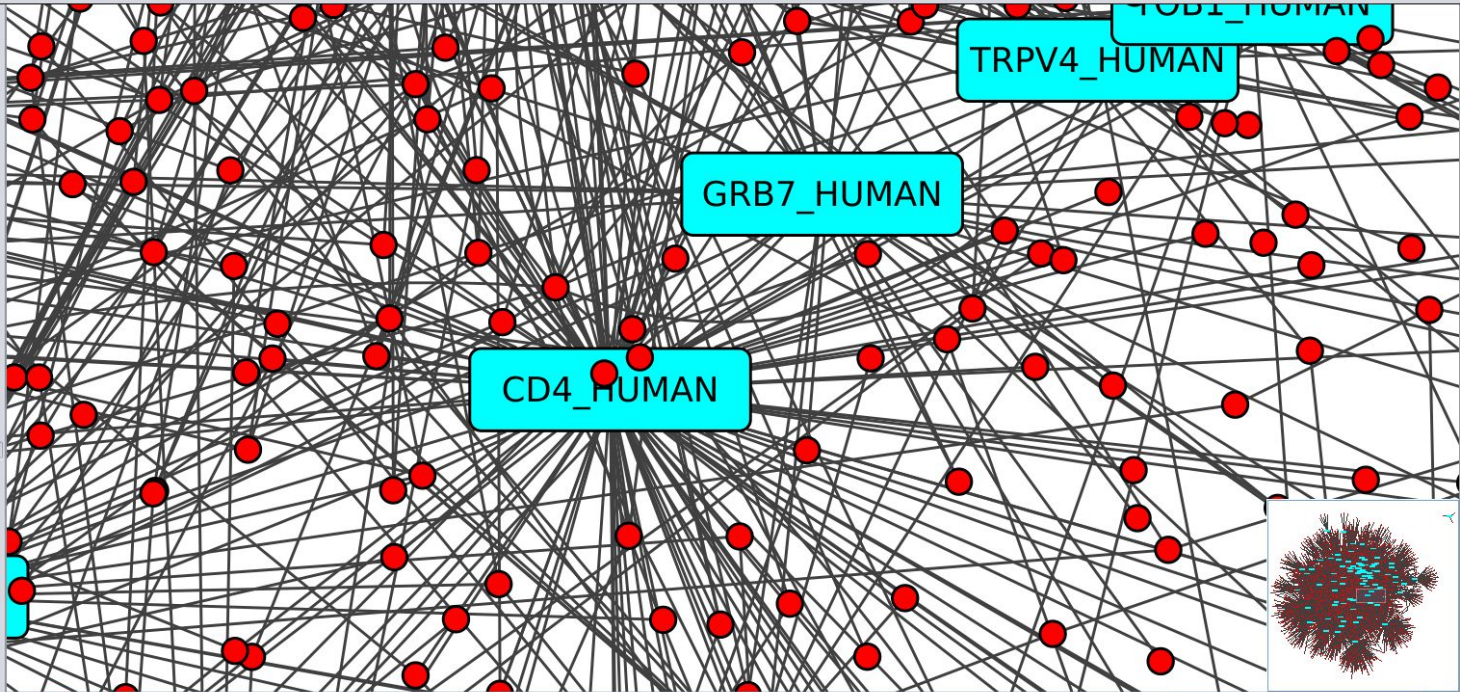
Network Style Select Annotation

Enter search terms for NDEx...

1 of 1 Network selected

Network 1

Network 3164 5531



Network

0 0 0 0

Table Panel

shared name	shared interaction	name	interaction
0:NUPR1_HUMAN	AL1A1_HUMAN -> NUPR1_HUMAN	0:NUPR1_HUMAN	AL1A1_HUMAN -> NUPR1_HUMAN
0:ALDH2_HUMAN	AL1A1_HUMAN -> ALDH2_HUMAN	0:ALDH2_HUMAN	AL1A1_HUMAN -> ALDH2_HUMAN
0:POTE1_HUMAN	AL1A1_HUMAN -> POTE1_HUMAN	0:POTE1_HUMAN	AL1A1_HUMAN -> POTE1_HUMAN
1:ACHA_HUMAN	ITA7_HUMAN -> ACHA_HUMAN	1:ACHA_HUMAN	ITA7_HUMAN -> ACHA_HUMAN

Node Table Edge Table Network Table

References for Cytoscape and others

- <http://code.cytoscape.org/javadoc/3.7.2/>
 - Actually useful javadoc style Cytoscape api reference
- <https://github.com/cytoscape/cytoscape-app-samples>
 - These samples + the “app ladder” serve as the bulk of what I needed to get started
- <https://github.com/cytoscape/cytoscape/wiki/Cytoscape-App-Ladder>
 - ‘App Ladder’