SAS® GLOBAL FORUM 2019

USERS PROGRAM

APRIL 28 – MAY 1, 2019 | DALLAS, TX

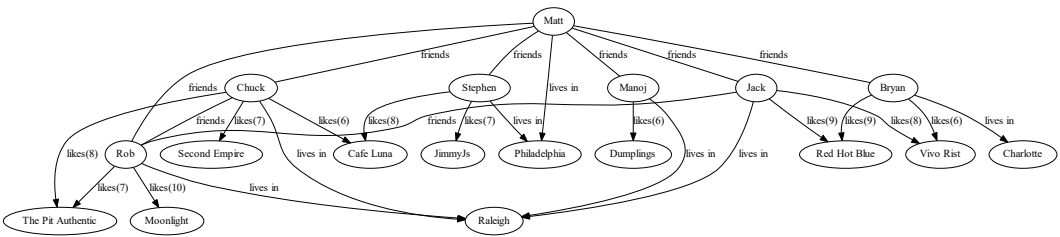
Matthew Galati, SAS Institute Inc.

Matthew has worked at SAS since 2004 and is a Distinguished Operations Research Specialist. He focuses on mixed integer linear programming and network algorithms and consults on difficult problems through the Advanced Analytics and Optimization Services group. Matthew has a B.S. in Mathematics from Stetson University and an M.S. and Ph.D. in Operations Research from Lehigh University.

Introducing Pattern Matching for Graph Queries in SAS® Viya® 3.4

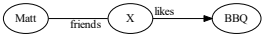
Network

- A *Graph* or *Network* represents relationships (or connections) between entities
  - *Node* — an entity
  - *Link* — a connection between a pair of nodes
  - Arbitrary number of attributes (distance, score, type, and so on) on links and/or nodes
- Example — *semantic network* or *knowledge base* (subject-predicate-object expressions)

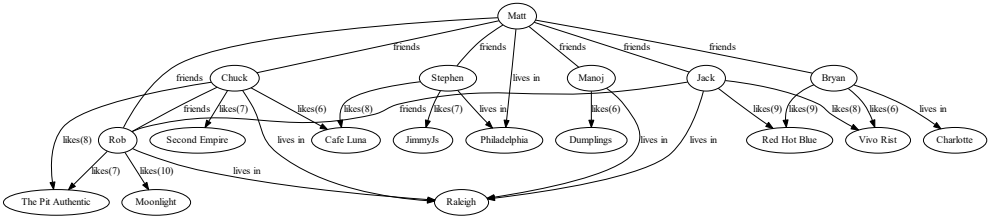


Pattern Matching

- *Subgraph isomorphism* — find all subgraphs of *G* that are isomorphic to *Q* (topology mapping)
- *Pattern matching* — subgraph isomorphism preserving all node and link attributes defined in *Q*
- `network.patternMatch` action and `PROC NETWORK` in SAS Visual Data Mining and Machine Learning

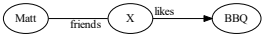


Friends of Matt who like barbecue restaurants

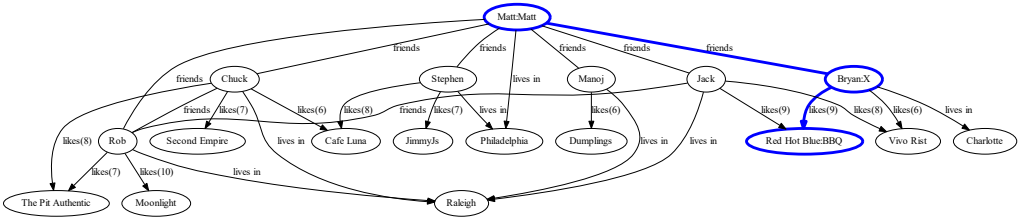


Pattern Matching

- *Subgraph isomorphism* — find all subgraphs of *G* that are isomorphic to *Q* (topology mapping)
- *Pattern matching* — subgraph isomorphism preserving all node and link attributes defined in *Q*
- `network.patternMatch` action and `PROC NETWORK` in SAS Visual Data Mining and Machine Learning



Friends of Matt who like barbecue restaurants



Semantic Network—Data

Main Graph (*G*) Data Tables

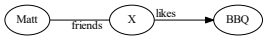
```
data mycas.NodesSocial;
   infile datalines dsd;
   length node $40. type $40. subtype $20.;
   input node $ type $ subtype $;
   label=node;
   datalines;
Matt,               Person,
Rob,                Person,
Raleigh,            City,
Philadelphia,       City,
Red Hot Blue,       Restaurant, BBQ
Vivo Rist,          Restaurant, Italian
...
;

data mycas.LinksSocial;
   infile datalines dsd;
   length from $40. to $40. connection $20.;
   input from $ to $ connection $ rating;
   datalines;
Matt,    Rob,               friends,  .
Rob,     Chuck,             friends,  .
Jack,    Rob,               friends,  .
Matt,    Stephen,           friends,  .
Matt,    Philadelphia,      lives in, .
Stephen, JimmyJs,           likes,    7
Jack,    Red Hot Blue,      likes,    9
...
;
```

Query Graph (*Q*) Data Tables

```
data mycas.NodesSocialQuery;
   infile datalines dsd;
   length node $40. label $40. type $40. subtype $20.;
   input node $ label $ type $ subtype $;
   datalines;
Matt, Matt, Person,
X,,         Person,
BBQ,,       Restaurant, BBQ
;

data mycas.LinksSocialQuery;
   infile datalines dsd;
   length from $40. to $40. connection $20.;
   input from $ to $ connection $;
   datalines;
Matt, X,    friends
X,    Matt, friends
X,    BBQ,  likes
;
```



Semantic Network—Code and Log

Calling patternMatch from PROC NETWORK

```
proc network
   direction         = directed
   nodes             = mycas.NodesSocial
   links             = mycas.LinksSocial
   nodesQuery        = mycas.NodesSocialQuery
   linksQuery        = mycas.LinksSocialQuery;
   nodesVar
      vars           = (label type subtype);
   linksVar
      vars           = (connection);
   nodesQueryVar
      vars           = (label type subtype);
   linksQueryVar
      vars           = (connection);
   patternMatch
      outMatchNodes  = mycas.OutMatchNodes
      outMatchLinks  = mycas.OutMatchLinks;
run;
```

PatternMatch Log File

```
NOTE: ----------------------------------------------------------------------------
NOTE: Running NETWORK.
NOTE: ----------------------------------------------------------------------------
NOTE: The number of nodes in the input graph is 18.
NOTE: The number of links in the input graph is 35.
NOTE: The number of nodes in the query graph is 3.
NOTE: The number of links in the query graph is 3.
NOTE: Processing the pattern matching query using 8 threads across 1 machines.
NOTE: The algorithm found 5 matches.
NOTE: Processing the pattern matching query used 0.00 (cpu: 0.00) seconds.
NOTE: The Cloud Analytic Services server processed the request in 0.007591
      seconds.
NOTE: The data set MYCAS.OUTMATCHNODES has 15 observations and 6 variables.
NOTE: The data set MYCAS.OUTMATCHLINKS has 15 observations and 4 variables.
```

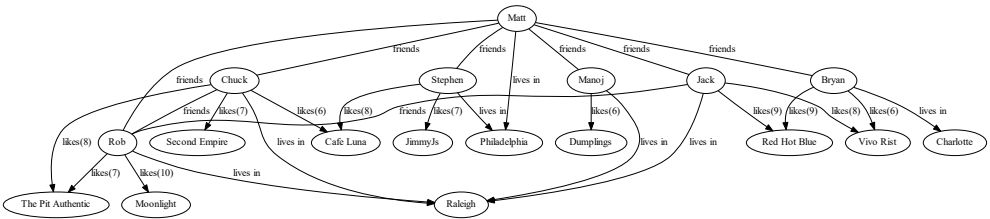
Semantic Network—Output

Node Mappings

match	nodeQ	node	label	type	subtype
1	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
1	Matt	Matt	Matt	Person	
1	X	Bryan	Bryan	Person	
2	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
2	Matt	Matt	Matt	Person	
2	X	Chuck	Chuck	Person	
3	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
3	Matt	Matt	Matt	Person	
3	X	Jack	Jack	Person	
4	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
4	Matt	Matt	Matt	Person	
4	X	Rob	Rob	Person	
5	BBQ	JimmyJs	JimmyJs	Restaurant	BBQ
5	Matt	Matt	Matt	Person	
5	X	Stephen	Stephen	Person	

Link Mappings

match	from	to	connection
1	Bryan	Matt	friends
1	Bryan	Red Hot Blue	likes
1	Matt	Bryan	friends
2	Chuck	Matt	friends
2	Chuck	The Pit Authentic	likes
2	Matt	Chuck	friends
3	Jack	Matt	friends
3	Jack	Red Hot Blue	likes
3	Matt	Jack	friends
4	Matt	Rob	friends
4	Rob	Matt	friends
4	Rob	The Pit Authentic	likes
5	Matt	Stephen	friends
5	Stephen	JimmyJs	likes
5	Stephen	Matt	friends



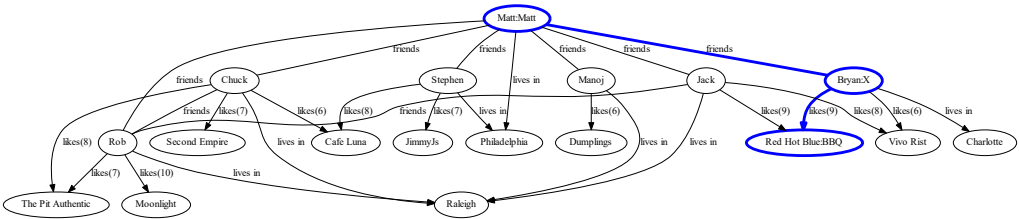
Semantic Network—Output

Node Mappings

match	nodeQ	node	label	type	subtype
1	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
1	Matt	Matt	Matt	Person	
1	X	Bryan	Bryan	Person	
2	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
2	Matt	Matt	Matt	Person	
2	X	Chuck	Chuck	Person	
3	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
3	Matt	Matt	Matt	Person	
3	X	Jack	Jack	Person	
4	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
4	Matt	Matt	Matt	Person	
4	X	Rob	Rob	Person	
5	BBQ	JimmyJs	JimmyJs	Restaurant	BBQ
5	Matt	Matt	Matt	Person	
5	X	Stephen	Stephen	Person	

Link Mappings

match	from	to	connection
1	Bryan	Matt	friends
1	Bryan	Red Hot Blue	likes
1	Matt	Bryan	friends
2	Chuck	Matt	friends
2	Chuck	The Pit Authentic	likes
2	Matt	Chuck	friends
3	Jack	Matt	friends
3	Jack	Red Hot Blue	likes
3	Matt	Jack	friends
4	Matt	Rob	friends
4	Rob	Matt	friends
4	Rob	The Pit Authentic	likes
5	Matt	Stephen	friends
5	Stephen	JimmyJs	likes
5	Stephen	Matt	friends



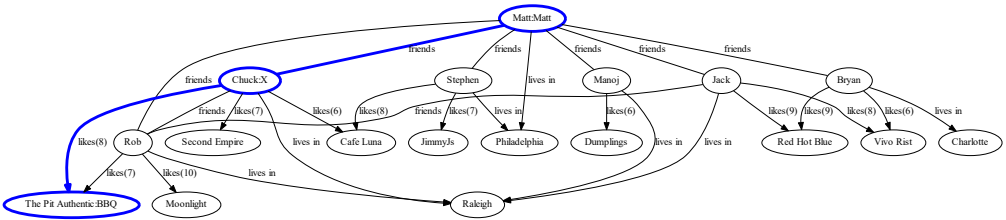
Semantic Network—Output

Node Mappings

match	nodeQ	node	label	type	subtype
1	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
1	Matt	Matt	Matt	Person	
1	X	Bryan	Bryan	Person	
2	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
2	Matt	Matt	Matt	Person	
2	X	Chuck	Chuck	Person	
3	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
3	Matt	Matt	Matt	Person	
3	X	Jack	Jack	Person	
4	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
4	Matt	Matt	Matt	Person	
4	X	Rob	Rob	Person	
5	BBQ	JimmyJs	JimmyJs	Restaurant	BBQ
5	Matt	Matt	Matt	Person	
5	X	Stephen	Stephen	Person	

Link Mappings

match	from	to	connection
1	Bryan	Matt	friends
1	Bryan	Red Hot Blue	likes
1	Matt	Bryan	friends
2	Chuck	Matt	friends
2	Chuck	The Pit Authentic	likes
2	Matt	Chuck	friends
3	Jack	Matt	friends
3	Jack	Red Hot Blue	likes
3	Matt	Jack	friends
4	Matt	Rob	friends
4	Rob	Matt	friends
4	Rob	The Pit Authentic	likes
5	Matt	Stephen	friends
5	Stephen	JimmyJs	likes
5	Stephen	Matt	friends



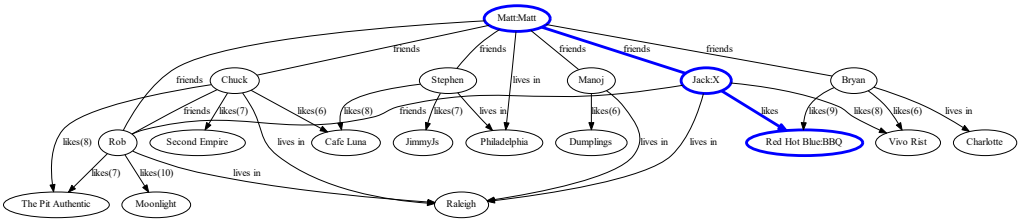
Semantic Network—Output

Node Mappings

match	nodeQ	node	label	type	subtype
1	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
1	Matt	Matt	Matt	Person	
1	X	Bryan	Bryan	Person	
2	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
2	Matt	Matt	Matt	Person	
2	X	Chuck	Chuck	Person	
3	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
3	Matt	Matt	Matt	Person	
3	X	Jack	Jack	Person	
4	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
4	Matt	Matt	Matt	Person	
4	X	Rob	Rob	Person	
5	BBQ	JimmyJs	JimmyJs	Restaurant	BBQ
5	Matt	Matt	Matt	Person	
5	X	Stephen	Stephen	Person	

Link Mappings

match	from	to	connection
1	Bryan	Matt	friends
1	Bryan	Red Hot Blue	likes
1	Matt	Bryan	friends
2	Chuck	Matt	friends
2	Chuck	The Pit Authentic	likes
2	Matt	Chuck	friends
3	Jack	Matt	friends
3	Jack	Red Hot Blue	likes
3	Matt	Jack	friends
4	Matt	Rob	friends
4	Rob	Matt	friends
4	Rob	The Pit Authentic	likes
5	Matt	Stephen	friends
5	Stephen	JimmyJs	likes
5	Stephen	Matt	friends



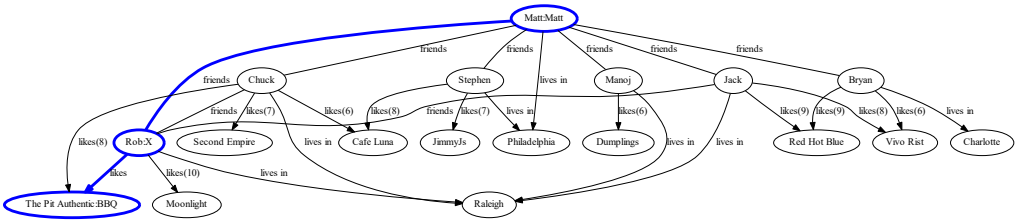
Semantic Network—Output

Node Mappings

match	nodeQ	node	label	type	subtype
1	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
1	Matt	Matt	Matt	Person	
1	X	Bryan	Bryan	Person	
2	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
2	Matt	Matt	Matt	Person	
2	X	Chuck	Chuck	Person	
3	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
3	Matt	Matt	Matt	Person	
3	X	Jack	Jack	Person	
4	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
4	Matt	Matt	Matt	Person	
4	X	Rob	Rob	Person	
5	BBQ	JimmyJs	JimmyJs	Restaurant	BBQ
5	Matt	Matt	Matt	Person	
5	X	Stephen	Stephen	Person	

Link Mappings

match	from	to	connection
1	Bryan	Matt	friends
1	Bryan	Red Hot Blue	likes
1	Matt	Bryan	friends
2	Chuck	Matt	friends
2	Chuck	The Pit Authentic	likes
2	Matt	Chuck	friends
3	Jack	Matt	friends
3	Jack	Red Hot Blue	likes
3	Matt	Jack	friends
4	Matt	Rob	friends
4	Rob	Matt	friends
4	Rob	The Pit Authentic	likes
5	Matt	Stephen	friends
5	Stephen	JimmyJs	likes
5	Stephen	Matt	friends



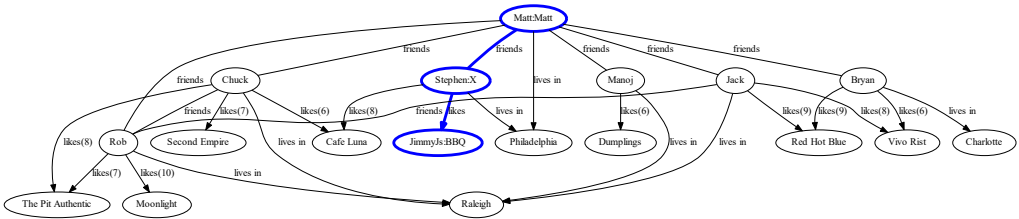
Semantic Network—Output

Node Mappings

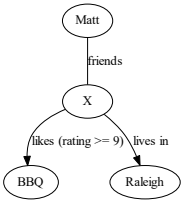
match	nodeQ	node	label	type	subtype
1	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
1	Matt	Matt	Matt	Person	
1	X	Bryan	Bryan	Person	
2	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
2	Matt	Matt	Matt	Person	
2	X	Chuck	Chuck	Person	
3	BBQ	Red Hot Blue	Red Hot Blue	Restaurant	BBQ
3	Matt	Matt	Matt	Person	
3	X	Jack	Jack	Person	
4	BBQ	The Pit Authentic	The Pit Authentic	Restaurant	BBQ
4	Matt	Matt	Matt	Person	
4	X	Rob	Rob	Person	
5	BBQ	JimmyJs	JimmyJs	Restaurant	BBQ
5	Matt	Matt	Matt	Person	
5	X	Stephen	Stephen	Person	

Link Mappings

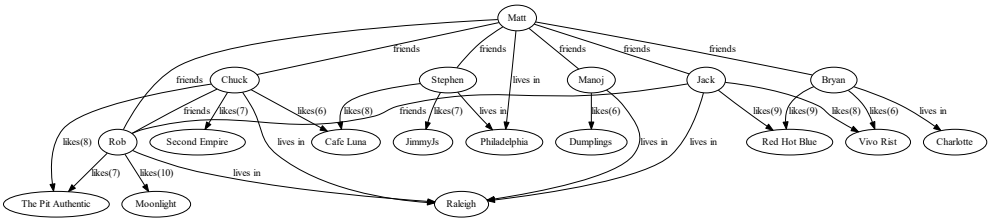
match	from	to	connection
1	Bryan	Matt	friends
1	Bryan	Red Hot Blue	likes
1	Matt	Bryan	friends
2	Chuck	Matt	friends
2	Chuck	The Pit Authentic	likes
2	Matt	Chuck	friends
3	Jack	Matt	friends
3	Jack	Red Hot Blue	likes
3	Matt	Jack	friends
4	Matt	Rob	friends
4	Rob	Matt	friends
4	Rob	The Pit Authentic	likes
5	Matt	Stephen	friends
5	Stephen	JimmyJs	likes
5	Stephen	Matt	friends



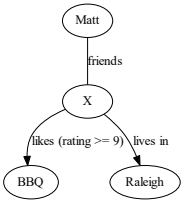
Semantic Network



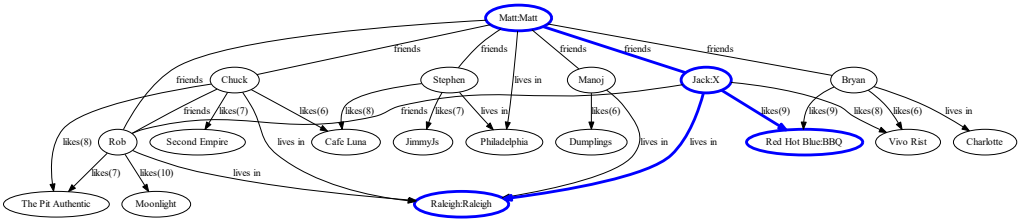
Friends of Matt who like barbecue restaurants, give the restaurant a rating of 9 or higher, and live in Raleigh



Semantic Network



Friends of Matt who like barbecue restaurants, give the restaurant a rating of 9 or higher, and live in Raleigh



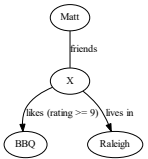
Semantic Network—Data and Code

Query Graph (*Q*) Data Tables and FCMP

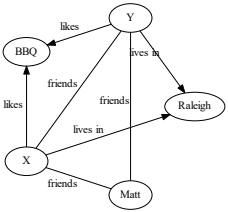
```
data mycas.NodesSocialQuery;
   infile datalines dsd;
   length node $40. label $40. type $40. subtype $20.;
   input node $ label $ type $ subtype $;
   datalines;
Matt,    Matt,    Person,
X,,              Person,
Raleigh, Raleigh, City,
BBQ,,            Restaurant, BBQ
;
data mycas.LinksSocialQuery;
   infile datalines dsd;
   length from $40. to $40. connection $20.;
   input from $ to $ connection $;
   datalines;
Matt, X,       friends
X,    Matt,    friends
X,    Raleigh, lives in
X,    BBQ,     likes
;
proc cas;
   source myFilter;
   function myLinkFilter(connectionQ $, rating);
      if (connectionQ='likes') then return (rating>=9);
      else return (1);
   endsub;
   endsource;
...
```

Calling patternMatch from PROC NETWORK

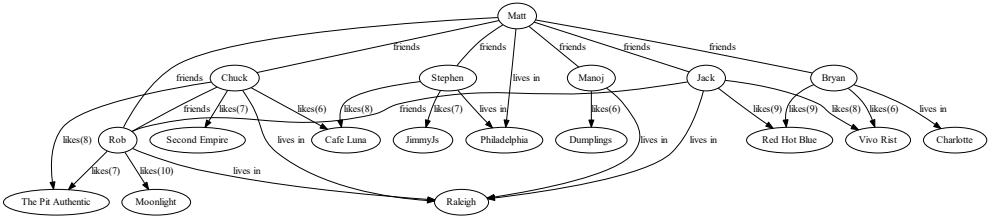
```
proc network
   direction        = directed
   nodes            = mycas.NodesSocial
   links            = mycas.LinksSocial
   nodesQuery       = mycas.NodesSocialQuery
   linksQuery       = mycas.LinksSocialQuery;
   nodesVar
      vars          = (label type subtype);
   linksVar
      vars          = (connection rating);
   nodesQueryVar
      vars          = (label type subtype);
   linksQueryVar
      vars          = (connection);
   patternMatch
      linkFilter    = myLinkFilter
      outMatchNodes = mycas.OutMatchNodes
      outMatchLinks = mycas.OutMatchLinks;
run;
```



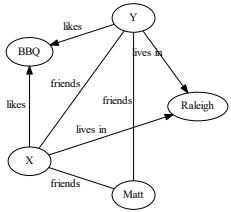
Semantic Network



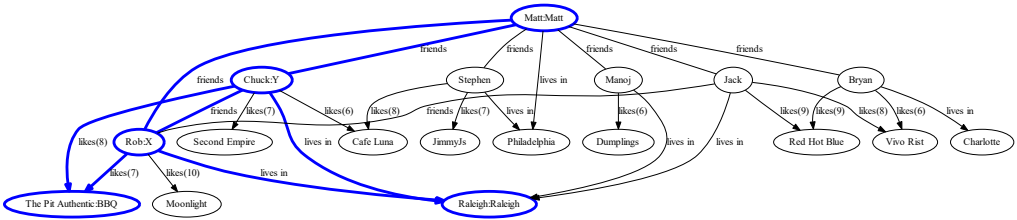
A pair of friends of Matt who like the same barbecue restaurant, live in Raleigh, and are friends of each other



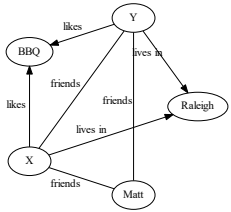
Semantic Network



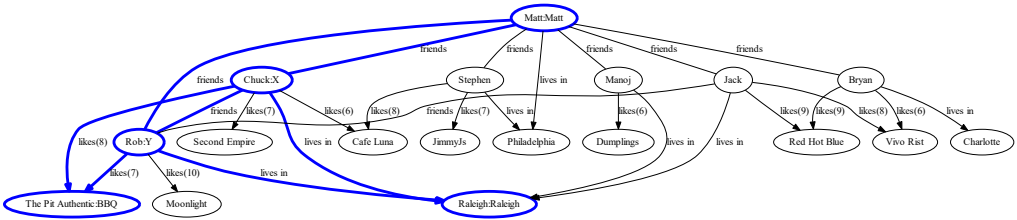
A pair of friends of Matt who like the same barbecue restaurant, live in Raleigh, and are friends of each other



Semantic Network



A pair of friends of Matt who like the same barbecue restaurant, live in Raleigh, and are friends of each other



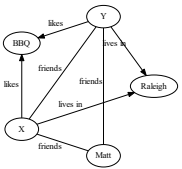
Semantic Network—Data and Code

Query Graph (Q) Data Tables

```
data mycas.NodesSocialQuery;
   infile datalines dsd;
   length node $40. label $40. type $40. subtype $20.;
   input node $ label $ type $ subtype $;
   datalines;
Matt,    Matt,    Person,
X,,              Person,
Y,,              Person,
Raleigh, Raleigh, City,
BBQ,,            Restaurant, BBQ
;
data mycas.LinksSocialQuery;
   infile datalines dsd;
   length from $40. to $40. connection $20.;
   input from $ to $ connection $;
   datalines;
Matt, X,       friends
X,    Matt,    friends
Matt, Y,       friends
Y,    Matt,    friends
X,    Raleigh, lives in
Y,    Raleigh, lives in
X,    BBQ,     likes
Y,    BBQ,     likes
X,    Y,       friends
Y,    X,       friends
;
```

Calling patternMatch from PROC NETWORK

```
proc network
   direction        = directed
   nodes            = mycas.NodesSocial
   links            = mycas.LinksSocial
   nodesQuery       = mycas.NodesSocialQuery
   linksQuery       = mycas.LinksSocialQuery;
   nodesVar
      vars          = (label type subtype);
   linksVar
      vars          = (connection);
   nodesQueryVar
      vars          = (label type subtype);
   linksQueryVar
      vars          = (connection);
   patternMatch
      outMatchNodes = mycas.OutMatchNodes
      outMatchLinks = mycas.OutMatchLinks;
run;
```

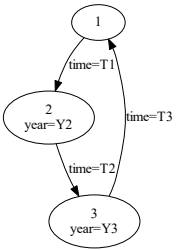


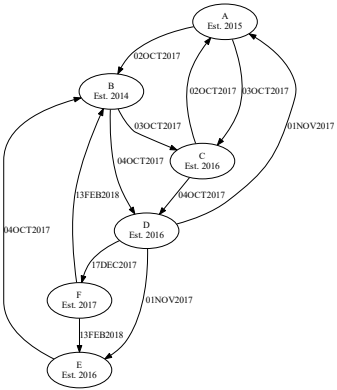
Applications of Pattern Matching

- Fraud (money laundering)
- Cybersecurity
- Social network analysis
- NLP and AI (semantic networks)
- Bio/chem-informatics
- Crystallography
- Image processing (computer vision)
- Compiler optimization
- CAD design of circuits
- ...

Money Laundering

Sequential banking transactions through corporate entities established in the same year that start and end at the same entity

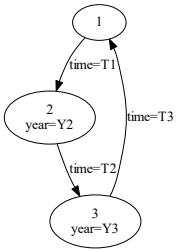


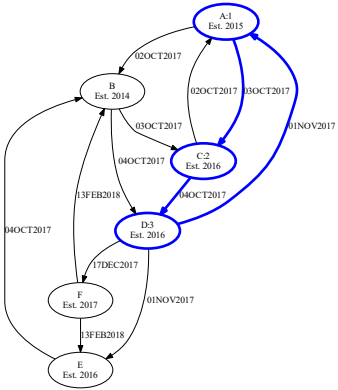


$\mathrm{Y2} = \mathrm{Y3}$ and $\mathrm{T1} < \mathrm{T2} < \mathrm{T3}$

Money Laundering

Sequential banking transactions through corporate entities established in the same year that start and end at the same entity





$\mathrm{Y2} = \mathrm{Y3}$ and $\mathrm{T1} < \mathrm{T2} < \mathrm{T3}$

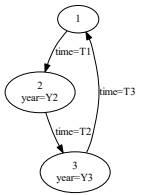
Money Laundering—Data and Code

Query Graph (Q) Data Tables and FCMP

```
data mycas.NodesQuery;
   input node @@;
   datalines;
1 2 3
;
data mycas.LinksQuery;
   input from to @@;
   datalines;
1 2  2 3  3 1
;
proc cas;
   source myPairFilter;
   function myNodePairFilter(nodeQ[*], year[*]);
      if (nodeQ[1]=2 and nodeQ[2]=3) then return (year[1]=year[2]);
      else return (1);
   endsub;
   function myLinkPairFilter(fromQ[*], toQ[*], time[*]);
      if (toQ[1]=1) then return (1);
      else if (toQ[1]=fromQ[2]) then return (time[1]<time[2]);
      else return (1);
   endsub;
   endsource;
...
```

Calling patternMatch from PROC NETWORK

```
proc network
   direction         = directed
   nodes             = mycas.NodesAML
   links             = mycas.LinksAML
   nodesQuery        = mycas.NodesQuery
   linksQuery        = mycas.LinksQuery;
   nodesVar
      vars           = (year);
   linksVar
      vars           = (time);
   patternMatch
      nodePairFilter = myNodePairFilter
      linkPairFilter = myLinkPairFilter
      outMatchNodes  = mycas.OutMatchNodes
      outMatchLinks  = mycas.OutMatchLinks;
run;
```



$Y2 = Y3$ and $T1 < T2 < T3$

Computational Comparison to iGraph and Neo4j

- Generally a very difficult problem to solve (*NP-complete*)
- Data sources
  - Stanford Network Analysis Project (SNAP)
  - Synthetically generated (Erdős-Rényi and Barabási-Albert models)
  - Lehigh University Benchmark (LUBM)
- $|E(G)|$ ranges from 950,000 to 15,000,000
- $|E(Q)|$ ranges from 1 to 45 with various data attributes and topologies
- Number of matches ranges from 2 to 1,939,108
- Server—Intel Xeon CPU X5550 @ 2.67 GHZ (2x4 CPUs), 64 GB RAM, running RHEL6.3

Software	Total Time (seconds)	Average Speedup
SAS Viya 3.4	261	151.8
iGraph 0.7.1	37,699	

Network versus iGraph (5 main graphs, 19 query graphs)

Software	Memory (GB)	Total Time (seconds)	Average Speedup
SAS Viya dev	1.9	145.7	7.2
SAS Viya 3.4	1.9	306.3	3.6
Neo4j 3.5.1	>24	822.1	

Network versus Neo4j (7 main graphs, 32 query graphs)

Algorithm Classes

- The `network` action set and `PROC NETWORK` in SAS Visual Data Mining and Machine Learning
- The `optNetwork` action set and `PROC OPTNETWORK` in SAS Optimization

	network	optNetwork
Topology/Descriptive		
connected components	✓	✓
biconnected components	✓	✓
clique enumeration	✓	✓
core decomposition	✓	
cycle enumeration	✓	✓
path enumeration	✓	✓
shortest path	✓	✓
summary statistics	✓	✓
transitive closure	✓	✓

	network	optNetwork
Network Analysis		
centrality	✓	
community detection	✓	
node similarity	✓	
ego (reach) networks	✓	
pattern matching	✓	
Optimization		
bipartite matching		✓
minimum-cost network flow		✓
minimum cut		✓
minimum spanning tree		✓
traveling salesman problem		✓

Thank you!

Contact Information
matthew.galati@sas.com

Reminder:

Complete your session survey in the conference mobile app.


#SASGF
SAS®
GLOBAL
FORUM
2019
APRIL 28 – MAY 1, 2019 | DALLAS, TX
Kay Bailey Hutchison Convention Center

API Comparison

Neo4j

- Graph database with a primary focus on data handling and matching through graph traversal
- Pattern matching supports (through the Cypher language):
  - exact topology (defined as ASCII-art paths)
  - exact and inexact attributes (SQL-like syntax)
  - limited inexact topology (variable path length)
- Traversal focus allows for more general filtering capability (at a computational cost)

SAS Viya 3.4

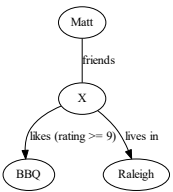
- Graph compute engine with a primary focus on fast analytical computation
- Pattern matching supports:
  - exact topology (data tables—node and edge lists)
  - exact and inexact attributes (FCMP syntax)
- Filtering capability is (currently) limited to individual or pairs of nodes or links

Cypher Example

Friends of Matt who like barbecue restaurants, give the restaurant a rating of 9 or higher, and live in Raleigh

Cypher

```
MATCH (Matt)-[:friends]->(X)-[:friends]->(Matt),
      (BBQ)<-[r:likes]-(X)-[:"lives in"]->(Raleigh)
WHERE Matt.label="Matt" and Matt.type="Person" and
      X.type="Person" and
      Raleigh.label="Raleigh" and Raleigh.type="City" and
      BBQ.type="Restaurant" and BBQ.subtype="BBQ" and
      r.rating>=9
RETURN (X)
```



SAS Viya 3.4

```
data mycas.NodesSocialQuery;
   infile datalines dsd;
   length node $40. label $40. type $40. subtype $20.;
   input node $ label $ type $ subtype $;
   datalines;
Matt,    Matt,    Person,
X,,              Person,
Raleigh, Raleigh, City,
BBQ,,            Restaurant, BBQ
;
data mycas.LinksSocialQuery;
   infile datalines dsd;
   length from $40. to $40. connection $20.;
   input from $ to $ connection $;
   datalines;
Matt, X,       friends
X,    Matt,    friends
X,    Raleigh, lives in
X,    BBQ,     likes
;
proc cas;
   source myFilter;
   function myLinkFilter(connectionQ $, rating);
      if (connectionQ='likes') then return (rating>=9);
      else return (1);
   endsub;
   endsource;
...
```