EMC Patent Exploration

Bruce Rabe (bruce.rabe@emc.com)
March 2014

EMC Corporation was granted 2,917 US patents from 1990-2013 (not including EMC subsidiaries, such as RSA Security, VMWare, etc.). I was curious about how long it takes from filing to granting (3.74 years), how many individual inventors there are (2032), where they were located (53% in Massachusetts), who has the most patents (Natan Vishlitzky), how they worked together and influenced each other, etc. Also, I wanted to learn how to use Python, R, Gephi, iGraph, and Pajek.

EMC's 2,917 compares to NetApp with 1,298 granted patents, IBM with 83,429 total patents granted, and Hewlett-Packard with 26,993.

Finding the Data

I looked for an API to extract patent data but didn't find anything that was still active. So I ended up extracting the patent data from the uspto.gov website by processing the HTML using a couple of python scripts. I also looked at Google patent search, but it returns too many non-matching results. For each patent, attributes for number, title, patent length, inventors, attorney, primary and secondary examiners, and citations were extracted into EMCPatents.json file.

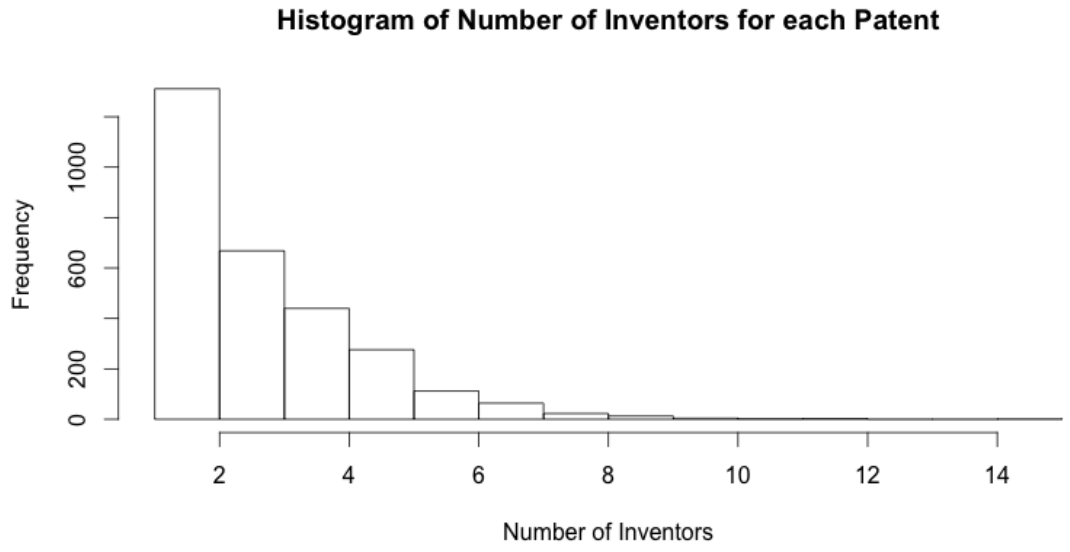
Cleaning Up the Data

At the start there were 2,782 inventors, but several were duplicates with different variations on the names (missing middle initials, missing period after the initial, etc.). Also several inventors moved during the last several years, so they had different cities listed. For example:

```
"Bauer; Andreas (Acton, MA)":
"Bauer; Andreas (Boxborough, MA)":
"Bauer; Andreas L. (Acton, MA)":
"Bauer; Andreas L. (Boxborough, MA)":
"Bauer; Andreas L. (Maynard, MA)":
```

Patents per Inventor

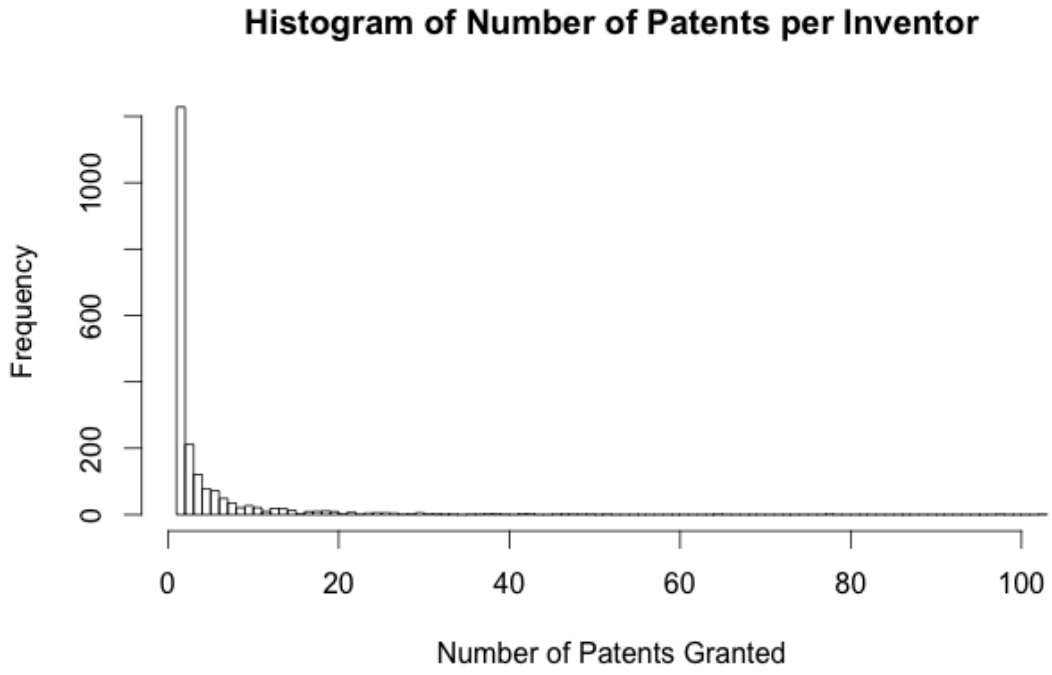
Cleaning up the data results in 2,032 unique individual inventors for the 2,917 patents, indicating several inventors with multiple patents. The mean (arithmetic average) number of inventors per patent is 2.9, ranging from mostly sole inventors to one patent with 15 inventors. As you can see, a histogram has a long right tail.



Natan Vishlitzky has the most patents with 103 – not bad! Here's the list of the top twenty:

103 Vishlitzky; Natan (Brookline, MA)
098 Todd; Stephen J. (Shrewsbury, MA)
078 Blumenau; Steven M. (Holliston, MA)
065 Ofek; Yuval (Hopkinton, MA)
052 Don; Arieh (Newton, MA)
050 Ofer; Adi (Sunnyvale, CA)
049 Claudatos; Christopher H. (San Jose, CA)
048 Walton; John K. (Mendon, MA)
047 Halstead; Mark J. (Holliston, MA)
046 Meiri; David (Cambridge, MA)
043 LeCrone; Douglas E. (Hopkinton, MA)
043 Gupta; Uday K. (Westford, MA)
043 Faibish; Sorin (Newton, MA)
042 Kopylovitz; Haim (Herzliya, IL)
042 Andruss; William D. (Minneapolis, MN)
040 Veprinsky; Alexander (Brookline, MA)
039 Teugels; Tom (Schoten, BE)
039 Ofer; Erez (Chestnut Hill, MA)
038 Kilian; Michael (Harvard, MA)
038 Castel; Daniel (Framingham, MA)
038 Bauer; Andreas L. (Boxborough, MA)

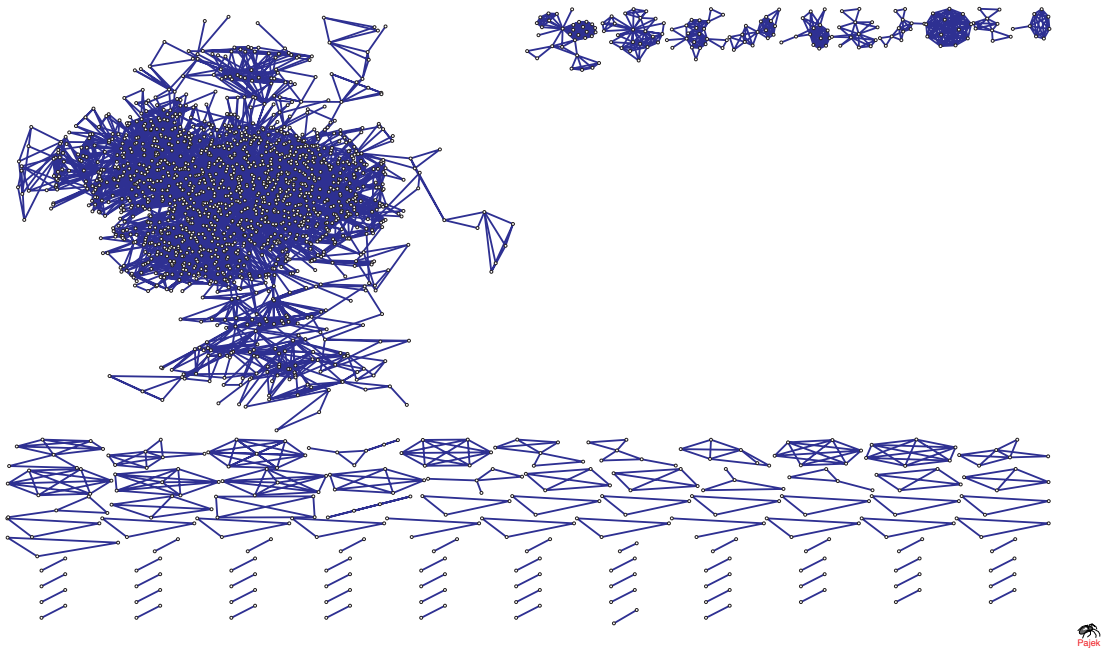
Plotting a histogram of patent counts per inventor shows that most inventors only have one one patent granted (so far):



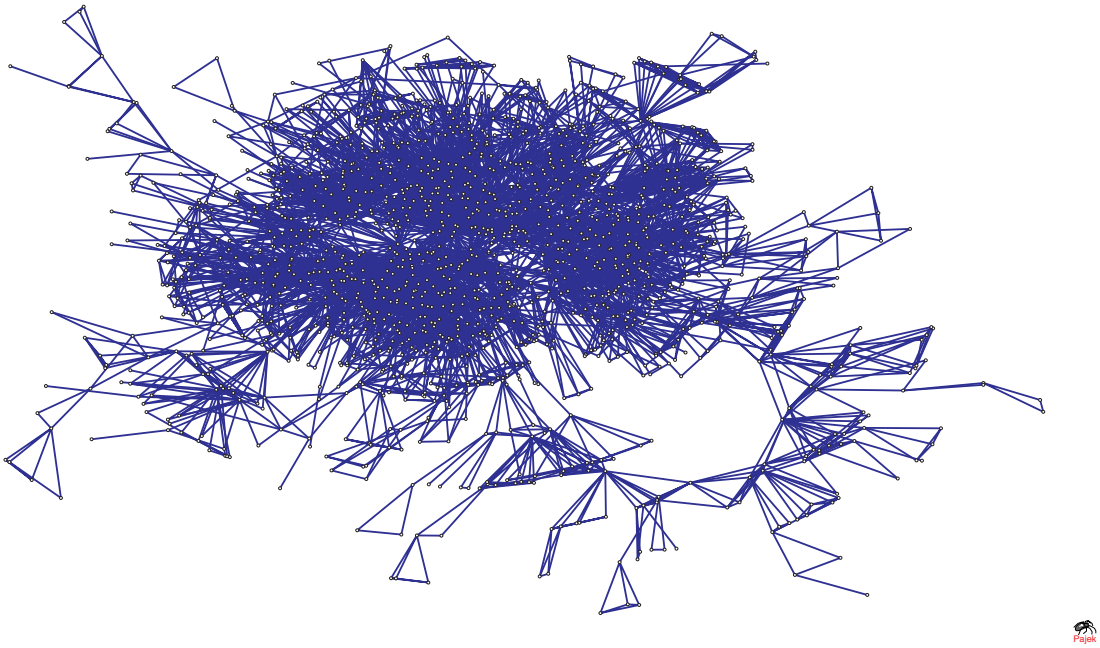
Minimum: 1
1st Quartile: 1
Median: 2
Mean: 4.277
3rd Quartile: 4
Maximum: 103

Inventor Relationships

If you look at the relationships between inventors for each patent, there are 6482 pairwise relationships between the 2032 inventors. Extracting this info into 2032 nodes and 12,793 edges into EMCInventorEdges.gdf (using PrintInventorEdgesAsGDF.py) makes it easy to read this into Gephi (gephi.org). There are 79 inventors who did not collaborate with other inventors, so we can discard these sole inventors when exploring the graph relationships. From Gephi, you can export to .net format and import into Pajek (pajek.imfm.si). Using the Fruchterman Reingold layout, you can see the inventor nodes and edge relationships grouped into clusters:



The average degree for each graph node is 6.6. This means that any inventor has worked with 6.6 other inventors, on average. There are 107 connected components, basically isolated clusters of related inventors without connections to other clusters. 1497 inventors (77%) are connected together in the largest connected component (the giant component). It has an average path length (i.e. degrees of separation) of 5.5 hops between any two inventors. The diameter, or maximum shortest path length, is 17 hops. Here is a graph of the giant component showing inventors as nodes and relationships as edges:



Many of the smaller components are groups of inventors with one or two patents.

Node Centrality

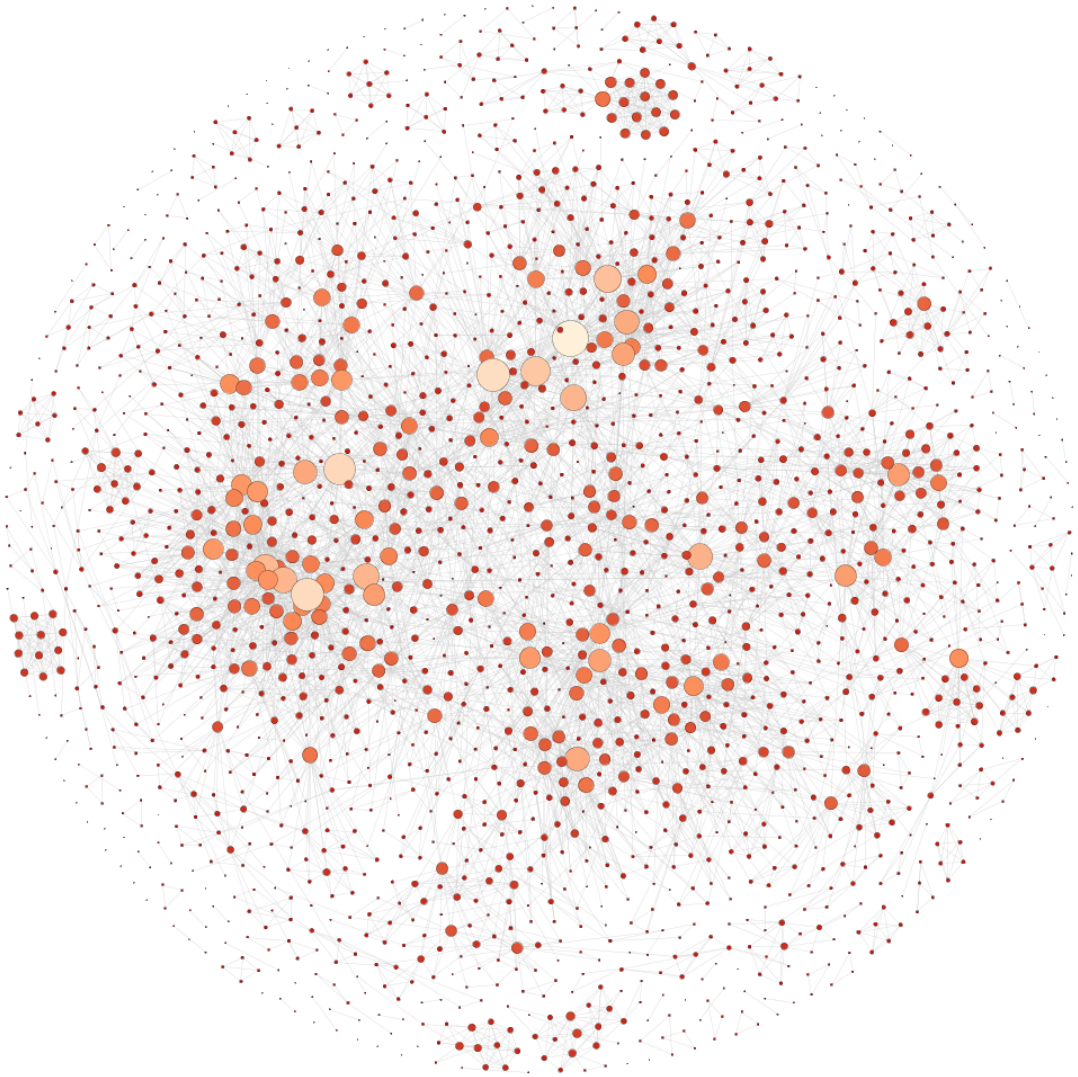
There are a several different ways to measure how "central" a particular graph node (inventor) is. I evaluated a few to see who are the more influential innovators.

- Degree
- Closeness
- Betweeness
- Eigenvectors

Degree is a simple measure of how many immediate neighbors a particular node has. In this case, it measures how many different inventors collaborated with a particular inventor. If we rank the degree for each inventor, the top ten look like this:

58 Gupta; Uday K. (Westford, MA)
53 Todd; Stephen J. (Shrewsbury, MA)
52 Don; Arieh (Newton, MA)
51 Vishlitzky; Natan (Brookline, MA)
46 Vahalia; Uresh K. (Newton, MA)
44 Faibish; Sorin (Newton, MA)
42 Meiri; David (Cambridge, MA)
41 Raizen; Helen S. (Jamaica Plain, MA)
41 Veprinsky; Alexander (Brookline, MA)
41 Bauer; Andreas L. (Boxborough, MA)

I was curious if the degree of each inventor node depended on the date in which that inventor first filed a patent, assuming the earlier they start, the more other people they work with. So I used Gephi and colored the nodes from darkest (first filed in 1988) to lightest (first filed in 2012) and then ranked the size by degree (1 to 58). I used the Fruchterman Reingold layout with Gravity set to 20. No obvious relationship appeared:



Closeness measures how “close” a particular node is to all other nodes, in terms of shortest path length – number of hops to that node. This could be a measure of how easy it is to get introduced to other inventors, for example if you wanted to collaborate on an idea together. These inventors are all between 3.6-3.8 hops.

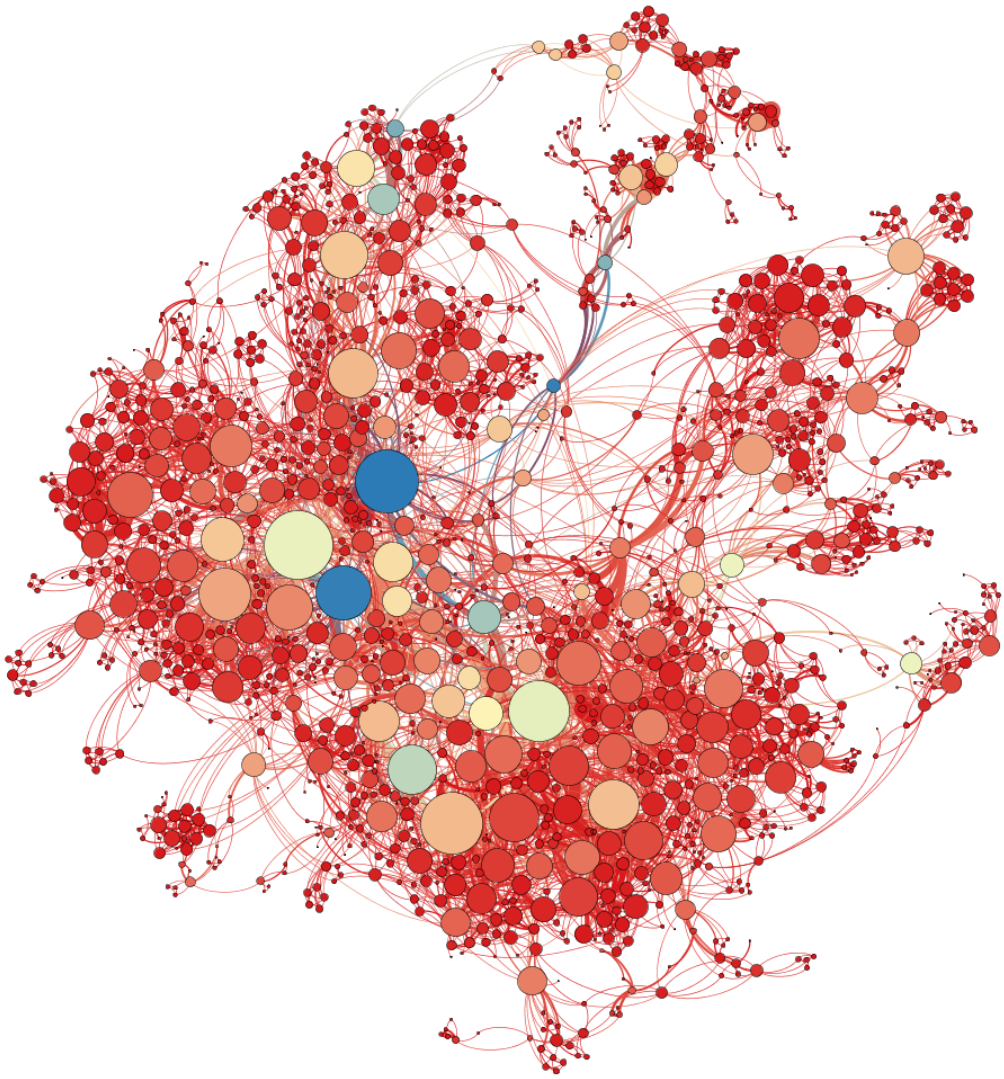
Vahalia; Uresh K. (Newton, MA)
Vishlitzky; Natan (Brookline, MA)
Todd; Stephen J. (Shrewsbury, MA)
Gupta; Uday K. (Westford, MA)
Blumenau; Steven M. (Holliston, MA)
Fitzgerald; John T. (Mansfield, MA)
Raizen; Helen S. (Jamaica Plain, MA)
Don; Arieh (Newton, MA)

Brown; Jeffrey A. (Shrewsbury, MA)
O'Brien, III; Walter A. (Westborough, MA)

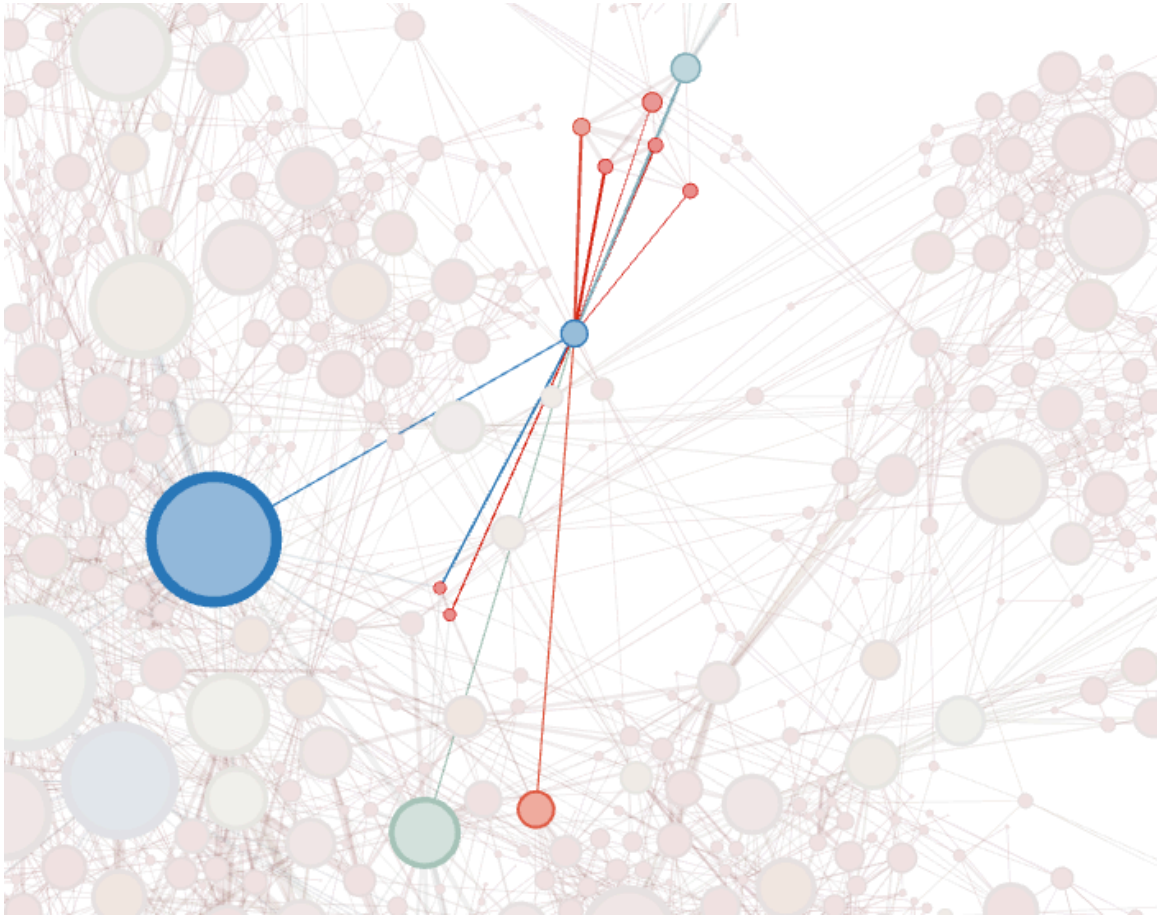
Betweenness in a graph is a measure of how influential a particular node is as a connector or intermediary between other nodes. Here is a list of the top ten inventors based on betweenness. These are the innovation enablers that work with many disparate groups to create patents.

Todd; Stephen J. (Shrewsbury, MA)
Bell, III; John Philip (Skibbereen, IE)
Vahalia; Uresh K. (Newton, MA)
Dings; Thomas L. (Hopkinton, MA)
Rankov; Alex (Danville, CA)
Blumenau; Steven M. (Holliston, MA)
Mutalik; Madhav G. (Southborough, MA)
Raizen; Helen S. (Jamaica Plain, MA)
Vishlitzky; Natan (Brookline, MA)
Gupta; Uday K. (Westford, MA)

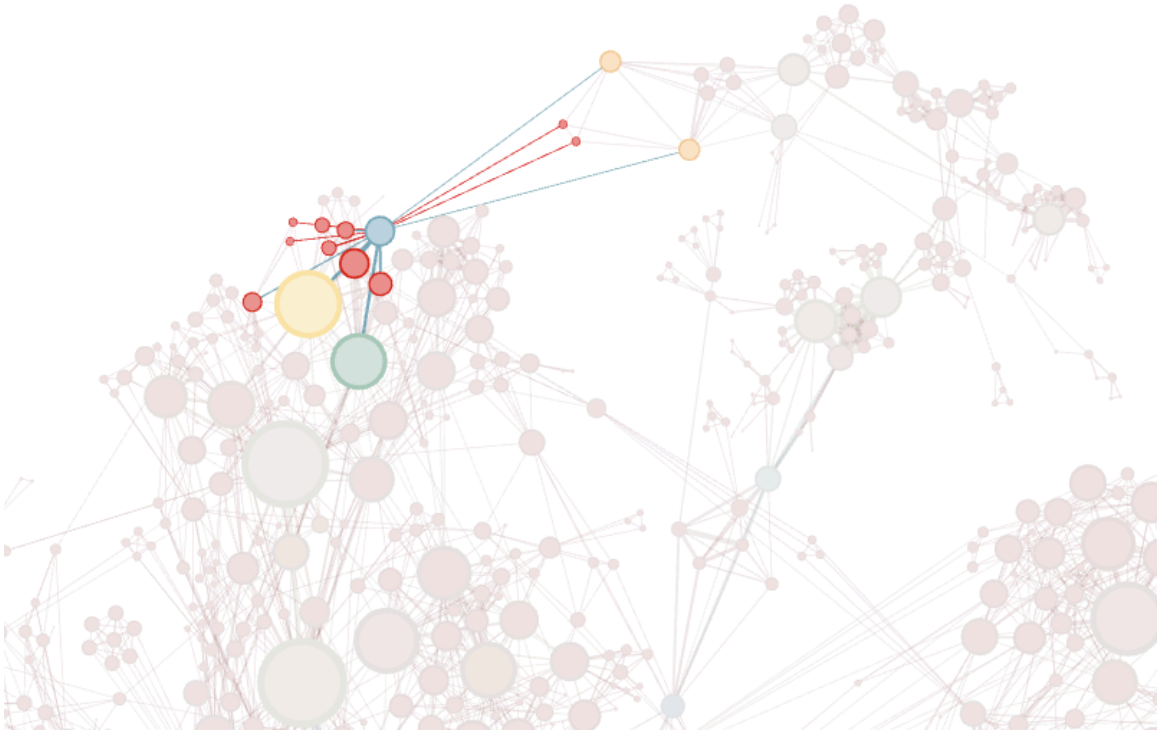
I used Gephi to size the nodes by degree and color them from red to blue (highest) by betweeness:



You can see that the larger nodes with higher degree tend to have a higher betweeness, but there are some exceptions. For example, John Philip Bell, III from Ireland has a fairly small degree (11) but is connected to a few different clusters of inventors:



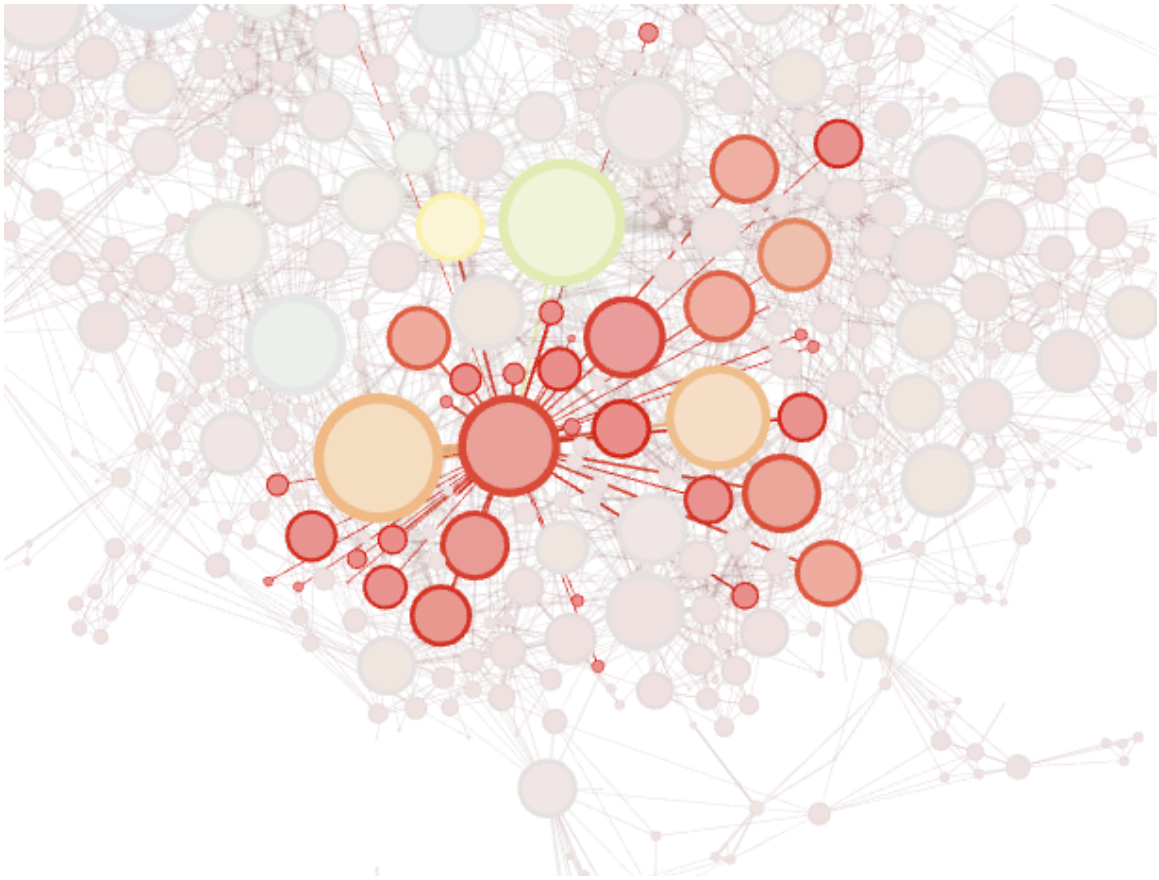
Thomas L. Dings has a similar story with 14 relationships (degrees) but connecting two fairly separate sets of inventors in California and Massachusetts:



Eigenvectors centrality is similar to betweeness, but it measures how many influential people that a node is linked to. This technique is often used in search engine page ranking.

Don; Arieh (Newton, MA)
Vishlitzky; Natan (Brookline, MA)
Gupta; Uday K. (Westford, MA)
Veprinsky; Alexander (Brookline, MA)
Meiri; David (Cambridge, MA)
Kopylovitz; Haim (Herzliya, IL)
Vahalia; Uresh K. (Newton, MA)
Halstead; Mark J. (Holliston, MA)
Fitzgerald; John T. (Mansfield, MA)
Riordan; Patrick Brian (West Newton, MA)

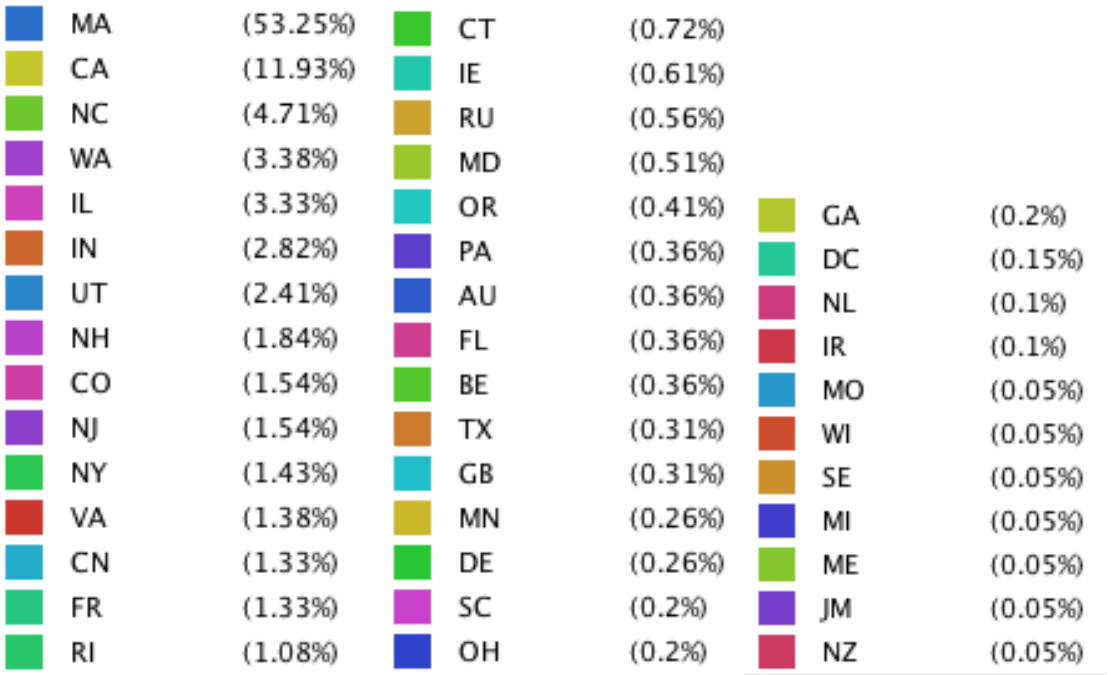
There are a couple of names that show up in the eigenvector ranking even though they have lower betweeness. For example, Alexander Veprinsky who is connected to several other inventors with high degree and betweeness:



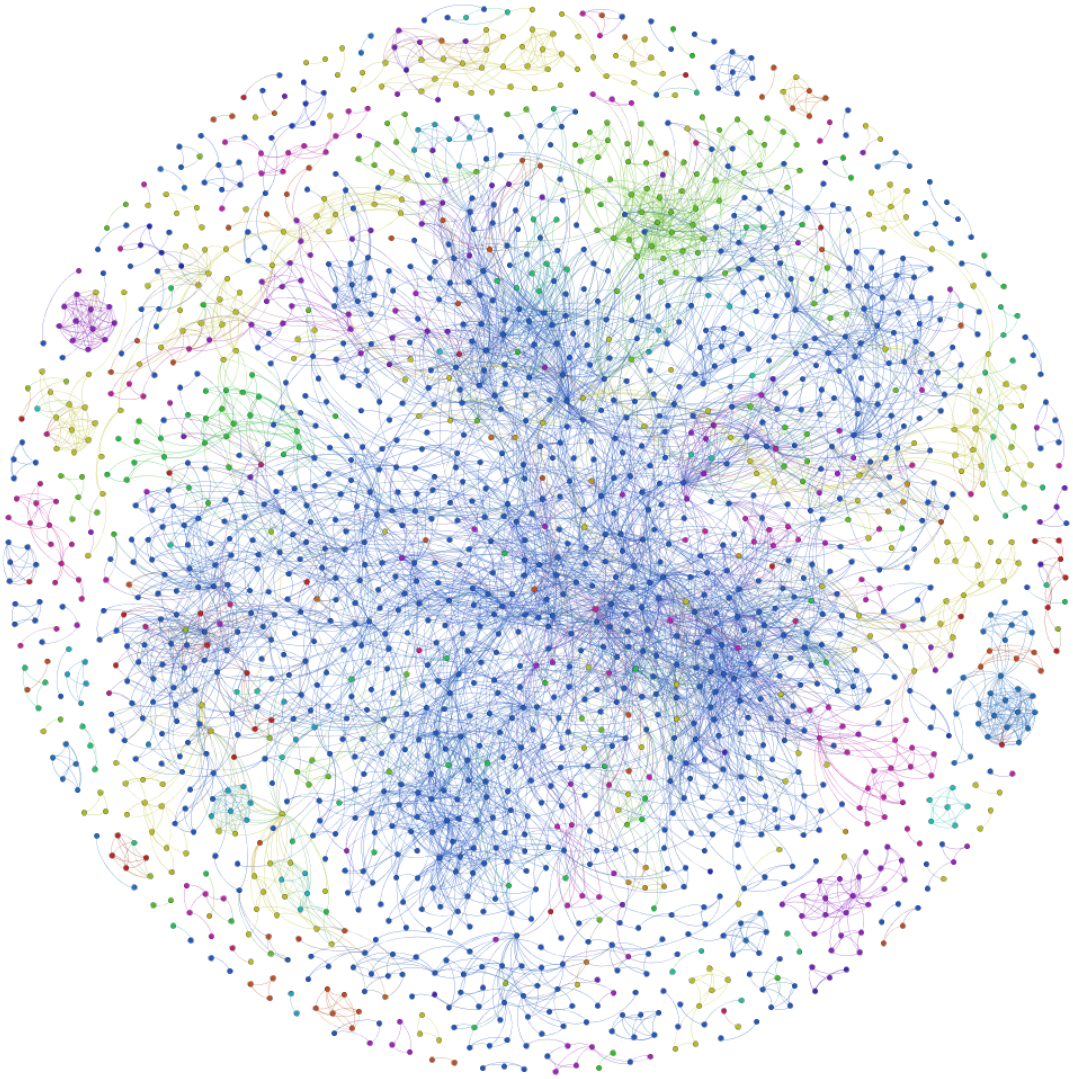
It would be cool to correlate inventor relationships and centrality measures to organizational structure, inventor's title/position/age/college/etc.

Breaking Down by State/Country

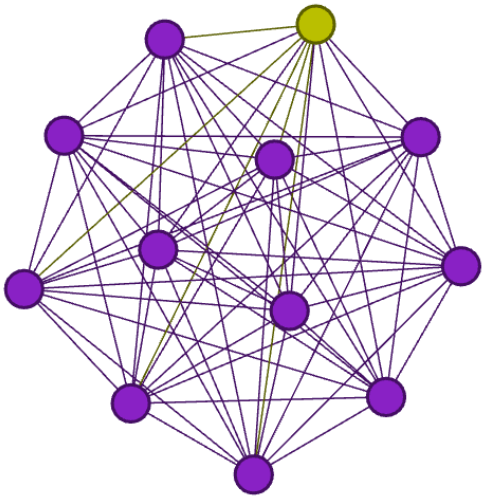
Extracting the state or country code for each inventor shows that over 53% are from Massachussetts and 12% from California. Here's the full breakdown:



Using these colors to partition the nodes in the graph shows that inventors from the same state or country are clustered closely together, as you might imagine:



Here's a blowup of the connected component at the top left showing inventors from Washington state in purple with one inventor (Jared Bobbitt) from California in green:

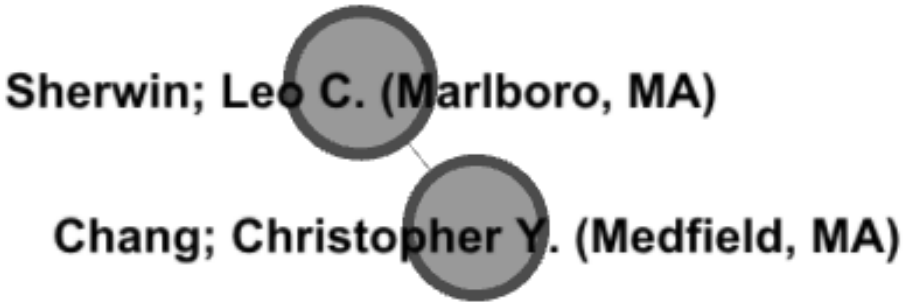


Looking at all the pivot table relationships between inventors, you can see these patterns emerge. The same state to state (or country to country) relationships show up on the diagonal. For example, Australian's (AU) collaborate with each other 22 times. Also, the inventors from Massachusetts and California collaborated with just about everyone, but mostly with inventors from their own state. Note that these two states have the most number of EMC employees.

	AU	BE	CA	CN	CO	CT	DC	DE	FL	FR	GA	GB	IE	IL	IN	IR	MA	MD	MN	NC	NH	NJ	NL	NY	NZ	OH	OR	PA	RI	RU	SC	SE	TX	UT	VA	WA
AU	22																1																			
BE		78															48																			
CA			1006	40	12			3	4		17	4	2	17			68		42	13	1	11	2	12		9	2	1	1			1	22		2	44
CN				51										1			11					4														
CO			6		104									3			36				1					2			1							3
CT			4			2		1						7			27							21												
DC																												1							3	
DE			4		1		3																					1								2
FL			7			1								4			24							4												
FR			4							49							60																			
GA											8						1																			
GB			3									8			1		7			1		1		2	3											
IE													13				10			3																
IL			14											166	1		135							1												
IN			20	1	2							1		1	69		35			6		4		1										2		
IR																	4																			
MA	4	45	243	31	56	20		25	8		5	32	155	26	2	7132	21	18	64	117	53		26	2		3	11	119		3	3	12	3	30	30
MD						3	1	2									10	4	2					6										10	
MN			20	12													28									17			4						
NC		3	12		5								6		4		77			374	3	11		2			1								1
NH			2	1	2							1			1		135	1		6	19	1			1							1		1	4
NJ			4	4		3		1				4					20			2	1	25		1	4			2				4			2
NY			7			18		4				2		1			10			5		1		60												
NZ			1									2					4			1		1														
OH			22																6							9										
OR			2														3										15									
PA			5					3									20					4						1						1		3	2
RI			3											1			78		1		3								3				1			
RU			8		3									1			14			3										9						
SC																	4			3											3					
SE																	1					1														
TX			10														1													1						
UT							1								1		7																	175	3	
VA			3				5		2								46	6			2			1										12	18	
WA			33		7										5		55		1		5												7			428

Examining Patents Year by Year

It is interesting to break down the data chronologically by filed application date to see what trends show up. As people move in and out and around the company new relationships should develop. In 1988, only one patent was submitted (at least only one was later granted) – "4,947,367: Providing a security-sensitive environment":



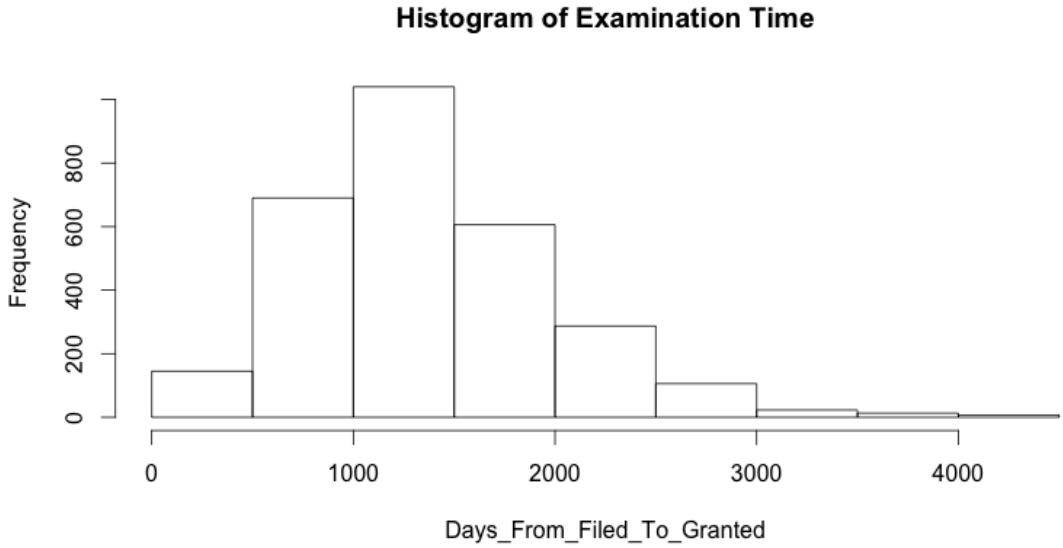
There were no patents filed the next year. Here is a table of what happened year by year:

Year	Patents Filed	Inventors	Links	Clusters	Avg. Degree
1988	1	2	1	1	1.0
1989	0				
1990	7	4	6	1	3.0
1991	0				
1992	3	7	9	2	2.5
1993	6	9	12	3	2.7
1994	39	44	64	10	2.9
1995	55	55	80	13	2.9
1996	70	56	88	10	3.1
1997	102	78	129	12	3.3
1998	100	108	192	18	3.6
1999	125	149	266	27	3.6
2000	177	202	514	26	5.1
2001	195	254	445	41	3.5
2002	171	271	660	31	4.9
2003	155	231	560	31	4.9
2004	258	344	779	44	4.5
2005	225	349	706	61	4.0
2006	392	521	1103	69	4.2
2007	309	415	898	53	4.3
2008	151	325	649	59	4.0
2009	102	252	462	54	3.7
2010	140	296	686	44	4.6
2011	97	239	547	38	4.6
2012	37	98	227	17	4.6

It would be cool to correlate the number of patents granted with the number of years that an inventor has been working at EMC.

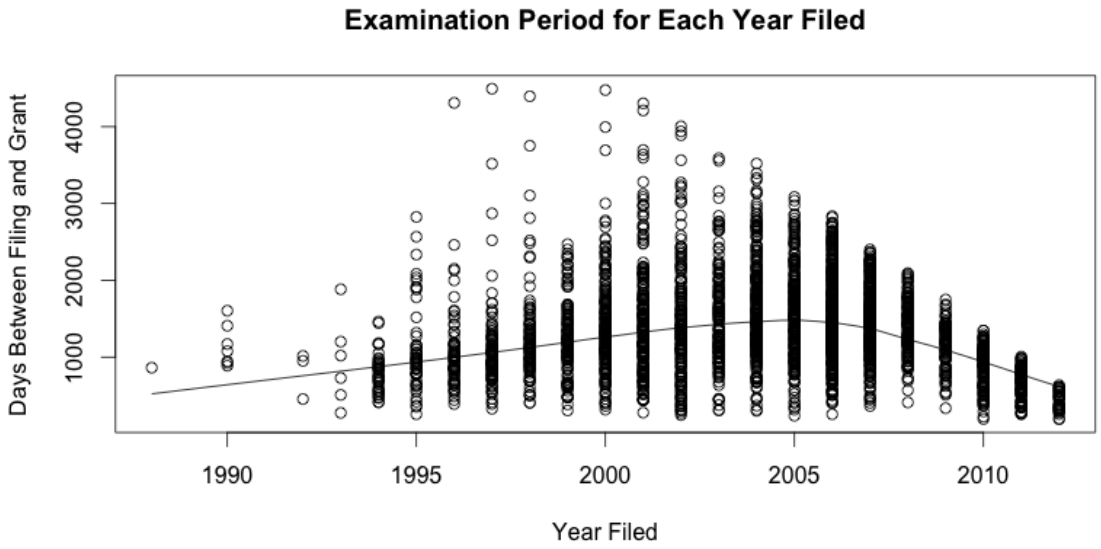
How Long Do Patents Take to Grant

The mean examination time between filing and granted date is 1,368 days (3.74 years) with a standard deviation of 617 days. Examination time ranges from 190 days to 4494 days (12.3 years! What happened with that one?).

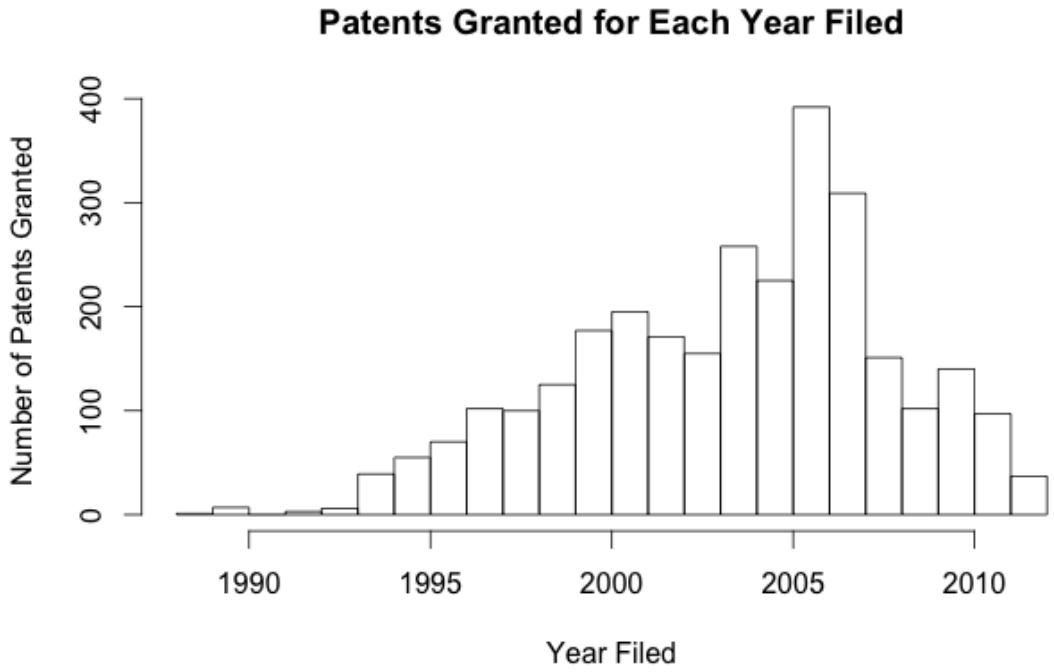


I tried to find particular factors that could influence how long it takes to grant a patent such as length of the patent application, number of citations in the application, attorney, primary or assistant examiner. None of these factors has a statistical correlation with the number of days to grant the patent.

It does look like the examination length gets longer over the years up to 2005, and then starts to become faster. Scatter plotting the year of filing vs. the examination period shows this trend.



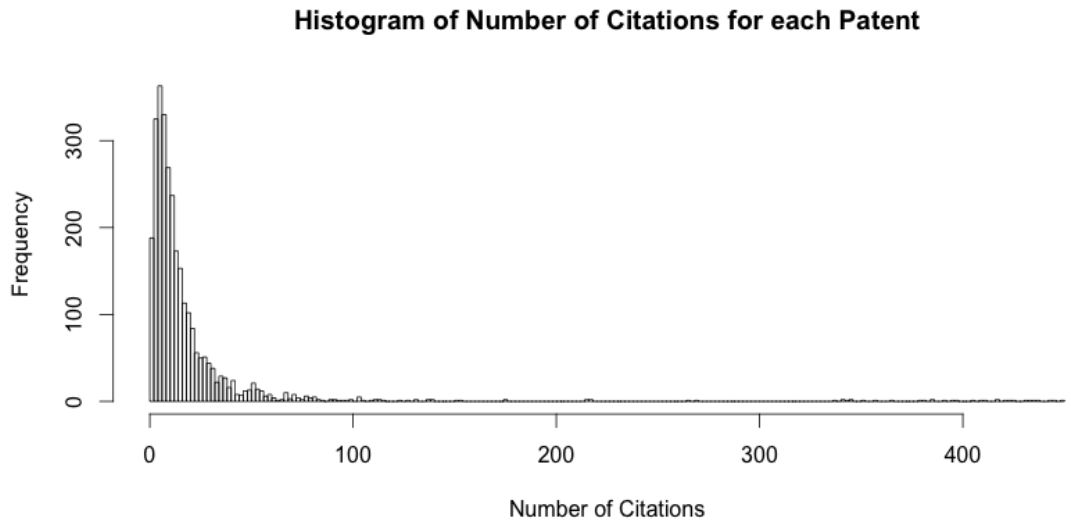
There is also a dropoff in the number of patents granted after 2005.



Number of Citations for Each Patent

While all the patents are innovative, I was curious about how many patents ended up citing other patents, giving an indication of how much they build on the ideas of other patents vs. standing alone. There was a pretty wide variation in the number of citations – from none to 450 cited. I'm not sure this is entirely valid, but here is a list of the seven patents which do not cite any other patents. Are these the most innovative?

7,809,667: Rule-based network resource compliance
7,720,666: Method for establishing bit error rate (BER) requirement for a data communication system
7,334,096: Architecture and apparatus for atomic multi-volume operations
7,328,318: System and method for atomic multi-volume operations
6,748,581: Apparatus and method for implementing an existing object-oriented class in an object management system
6,275,897: Remote cache utilization for mirrored mass storage subsystem
6,044,468: Secure transmission using an ordinarily insecure network communication protocol such as SNMP



Minimum: 0.00
1st Quartile: 6.00
Median: 10.00
Mean: 20.22
3rd Quartile: 19.00
Maximum: 450.00

Appendix A – How to reproduce the results:

To find a list of the patent numbers assigned to EMC, run the ExtractPatentNumbers.py python program and redirect the output to a file. Then run ExtractPatentInfo.py and pass in the file you just created (with the patent numbers). This program creates EMCPatents.json with an object for each patent listing number, length, inventors, filed, granted, examiners, attorney, citations, etc.

Running PrintInventorDuplicates.py creates another json file listing all the possible duplicates based on the same last name. This file needs to be manually cleaned up with a new_name attribute specifying the correct name to use in each case of a duplicate. The filed date attribute helps to make this determination, using the latest city listed for each inventor (at the time of filing). Then RemoveInventorDuplicates.py does a search and replace from the original inventor name to the new_name and fixes up EMCPatents.json.

R seems to like this json data in a different format to create a data frame, so ReformatJSONFile.py converts to EMCPatents2.json with separate arrays (vectors) for number, length, inventor_count, filed, granted, examiners, attorney, citation_count, etc.

InventorPatentTotals.py runs through the EMCPatents.json and prints out each inventor, their total number of patents, and the state or country they live in. This was redirected to InventorTotals.csv.

The following R code is used to generate many of the histograms and statistics:

```
> library('rjson')
> j <- fromJSON(file='EMCPatents2.json')
> df = data.frame(j)
> names(df)
> str(df)
> dim(df)
> head(df)
> mean(df$inventor_count)
> range(df$inventor_count)
> hist(df$inventor_count, xlab="Number of Inventors", main="Histogram of Number
of Inventors for each Patent")
> range(df$days_to_grant)
> mean(df$days_to_grant)
> median(df$days_to_grant)
> hist(df$days_to_grant, breaks=20, xlab="Days Between Filing and Grant",
main="Histogram of Examination Time")
> f <- function(x) as.POSIXlt( as.Date(x, format="%B %d, %Y") )$year+1900
```

```
> df$year = sapply(df$filed, f)
> scatter.smooth(df$year, df$days_to_grant, xlab="Year Filed", ylab="Days Between Filing and Grant", main="Examination Period for Each Year Filed")
> hist(df$year, breaks=25, ylab="Number of Patents Granted", main="Patents Granted for Each Year Filed", xlab="Year Filed")
> inventorTotals <- read.csv('InventorTotals.csv', sep='\t')
> summary(inventorTotals)
> hist(inventorTotals$Count, breaks=100, xlab="Number of Patents Granted", ylab="Frequency", main="Histogram of Number of Patents per Inventor")
> hist(df$citation_count, breaks=200, xlab="Number of Citations", main="Histogram of Number of Citations for each Patent")
> summary(df$citation_count)
```

Trying to find correlations between days_to_grant and other variables

```
> plot(df$days_to_grant ~ df$citation_count)
> lm <- lm(df$days_to_grant ~ df$length)
> summary(lm)
> qqnorm(lm$residuals, pch=20, col="seagreen")
> qqline(lm$residuals, col="red")
```

To import into Gephi, run the PrintInventorEdgesAsGDF.py python program. This will produce EMCInventorEdges.gdf, which can be opened in Gephi (make sure you change the Graph Type to Undirected before import). To remove nodes without any edges, run the Average Degree calculation under Statistics, switch to the Data Laboratory view for Nodes, and sort by the Degree column. Select all the rows with degree of 0 (sole inventors) and right click to Delete All. Set all the edge weights to 1 in the Edges section of the Data Table via the "Fill Column with Value" button at the bottom (which makes all the edge lines the same width). Then switch to Overview and partition color by state, ranked size by degree (or color by year), etc. Under statistics, the Connected Components analysis will separate out individual clusters. Avg. Path Length will calculate closeness and betweeness of nodes. Eigenvector Centrality calculates eignevector distribution.

I used the Fruchterman Reingold layout with Gravity 10 and Speed 10 for many of the layouts. Nodes can be partitioned (under Overview) based on state or degree. They can be ranked by color or node size by degree, betweeness/closeness centrality, year, or component ID.

PrintInventorEdgesAsGraphml.py will create a EMCInventorEdges<year>.graphml for each year that patents were granted. These were read into Gephi to calculate inventors, links, clusters, and average degree for each year.

The easiest way to create an EMCInventorEdges.net file for Pajek is to File*Export from Gephi.