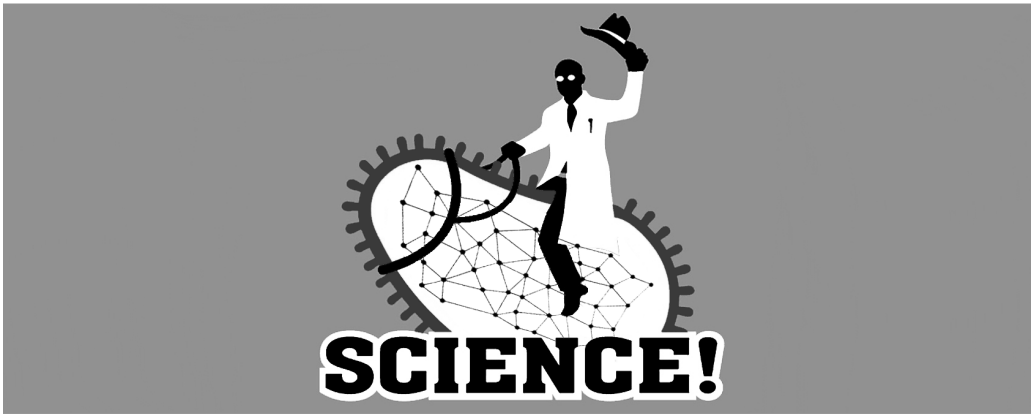
Análisis de datos a través de grafos

Cabral, Juan B.

16/05/2014 - Córdoba

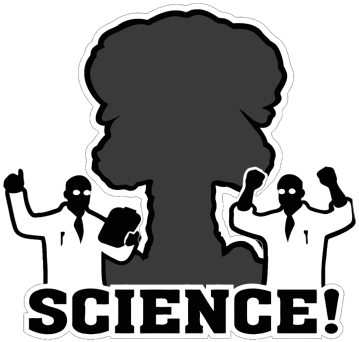


Universidad Católica de Córdoba - Facultad de Ingeniería

BioScience Data Mining Group

Integrantes

- García, Mario Alejandro
- Cabral, Juan Bautista
- Ramirez, Emilio
- Arnaud, Máximo "El Topo"
- Castro, Jésica
- INTA- Gimenez Pecci, María de la Paz
- CONICET INTA- Laguna, Irma Graciela
- INTA- Raspanti Monteoliva, Jorge G
- INTA- Maurino, Fernanda



Paper

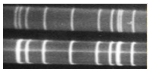
- **Título:** “Interactive network exploration in the KDD process, Contributions in the study of population variability of a Corn Fijivirus”
- **Autores:** M. A. García, M. P. Giménez Pecci, J. B. Cabral, A. Nieto, I. G. Laguna.
- **Publicación:** Journal of Data Mining in Genomics & Proteomics 2012 3:3
- **Editorial:** OMICS Publishing Group
- **ISSN:** 2153-0602. Año: 2012
- **URL:** http://goo.gl/pcjdG

Agenda

- Un poco de historia y motivaciones.
- Un poco de KDD.
- Un poco de Network Science.
- Proceso de Analisis.
- DW - OLAP - BI.
- El proyecto en sí
- Mini Demo

Historia: Mal de Río Cuarto virus

- Base de datos formada por
  - Perfiles electroforéticos.
  - Atributos que definen el ambiente de la planta



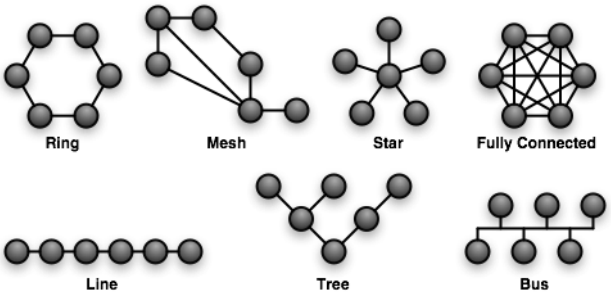
- Variabilidad (redes vs árboles).

Knowledge Discovery in Database (KDD)

- Es un proceso no trivial de identificación de información útil y desconocida que permanece oculta en una base de datos [Fayyad, 1996]
- Es un proceso centrado en la persona (human-centered) [Brachman, 1996]

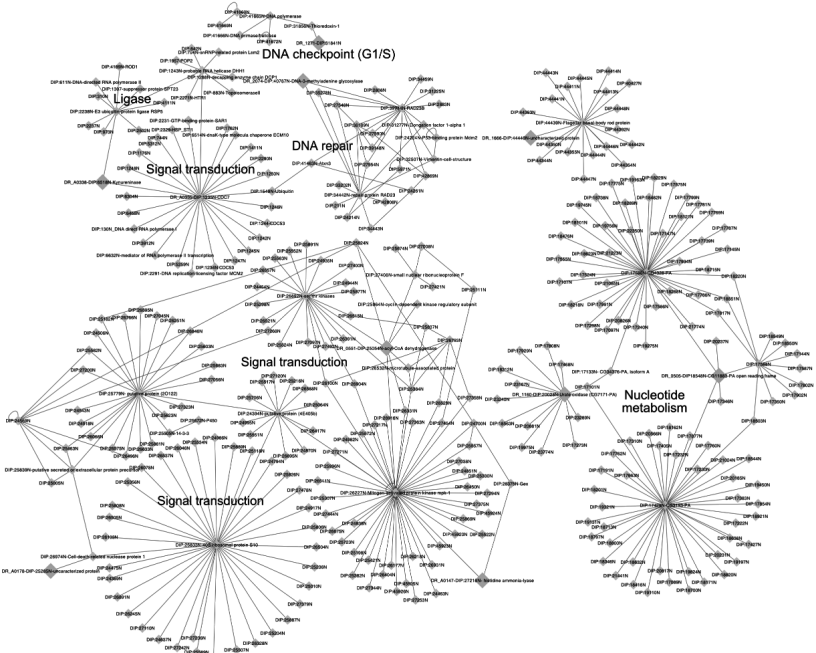
Network Science

- Es el estudio de las redes que representan fenómenos físicos, biológicos y sociales conduciendo a modelos predictivos de estos fenómenos.
- Topologías.
- Características comunes.



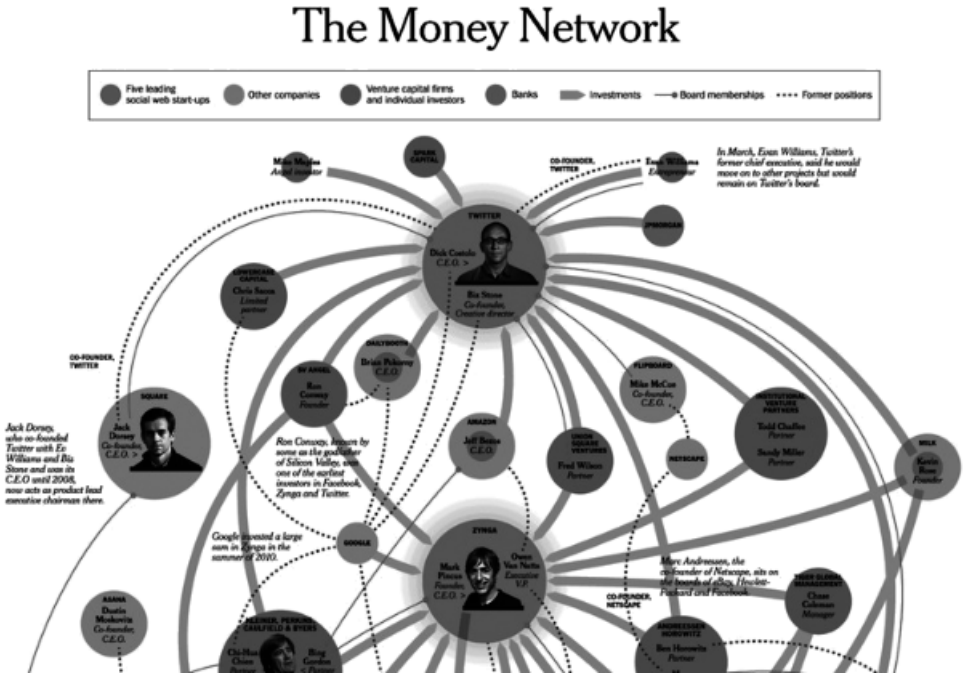
Topologías en redes de comunicación

Networks 1



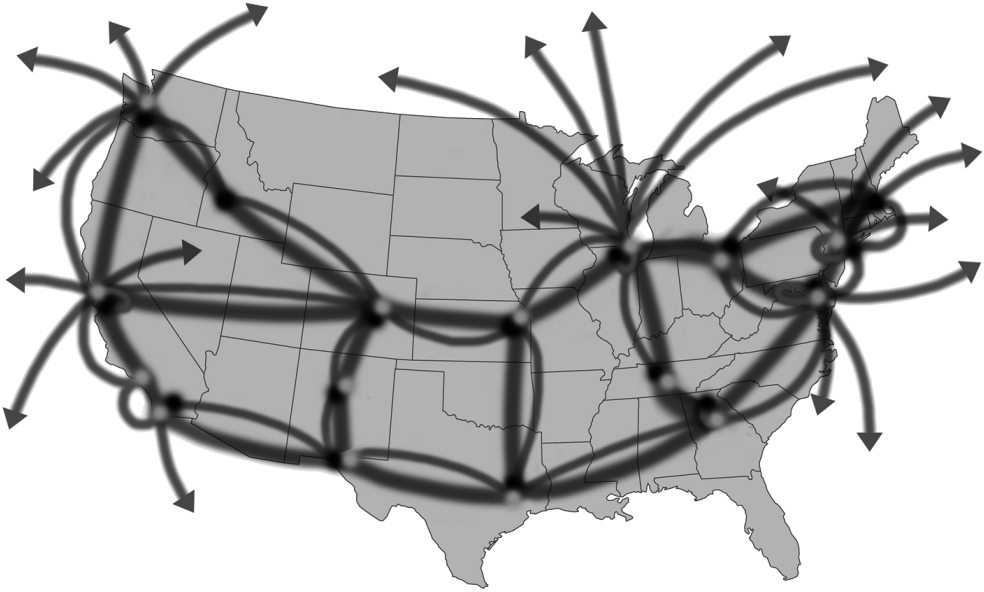
Interacción proteína-proteína

Networks 2



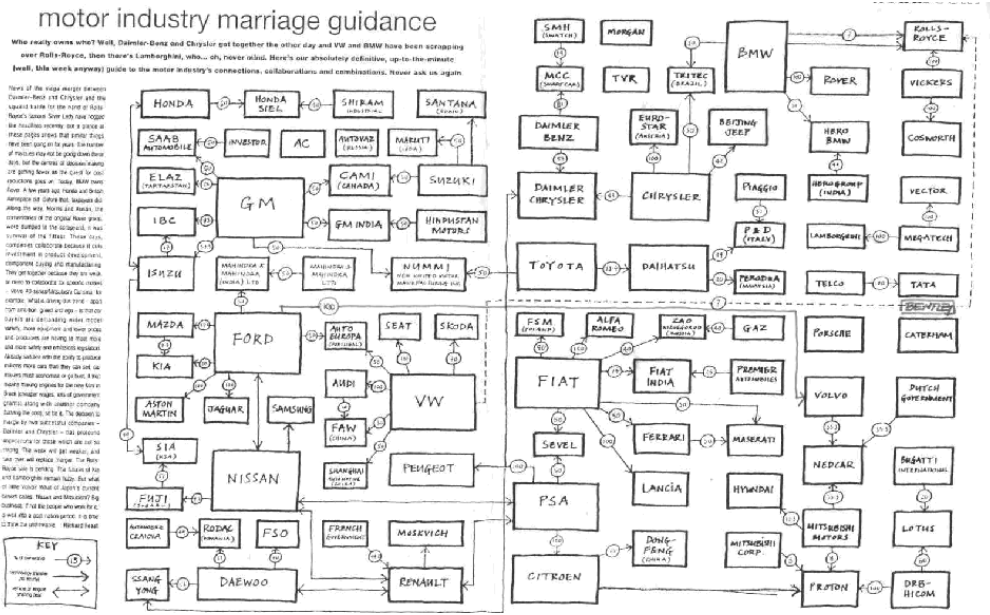
Redes sociales/económicas

Networks 3



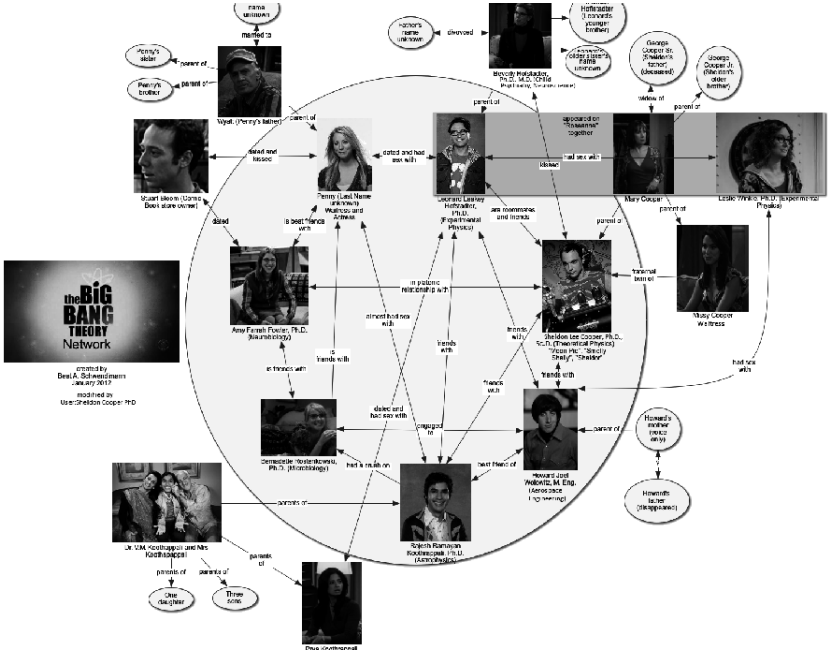
Red de distribución de energía

Networks 4



Relación entre automotrices

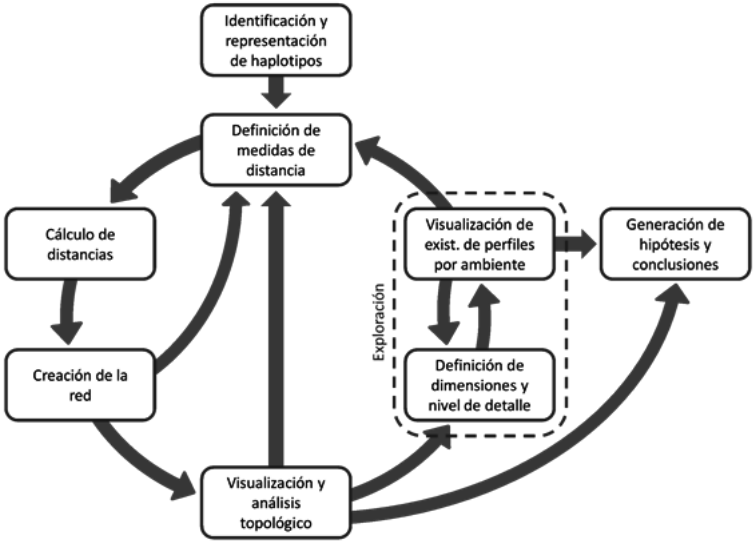
Networks 5



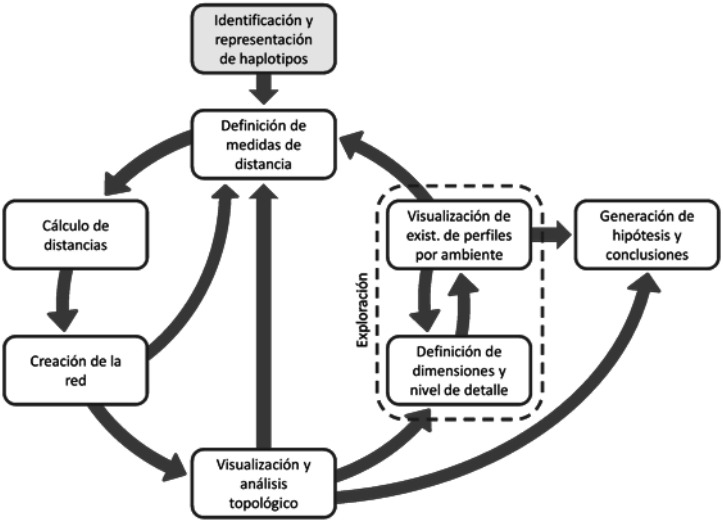
Red semántica TBBT (Season 3)

Proceso de análisis

Los ejemplos van con nuestra investigación (osea: resumen del paper)

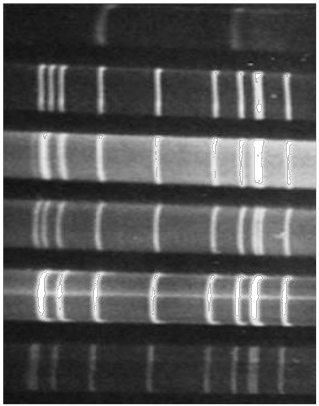


Proceso de análisis: Identificación y representación de haplotipos



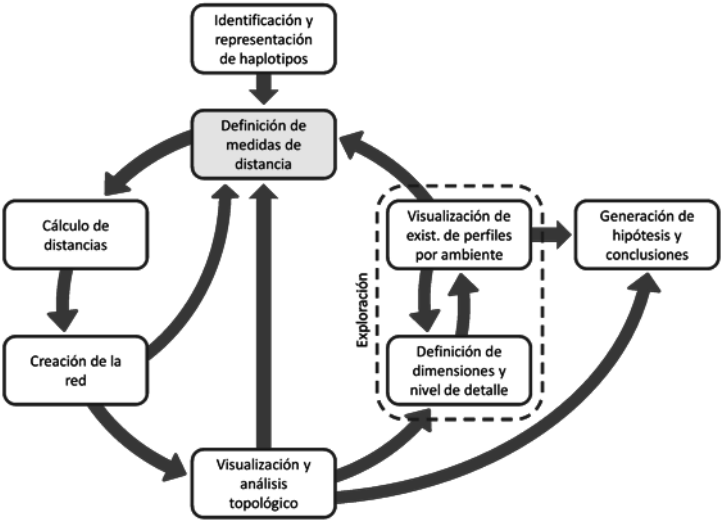
Proceso de análisis: Identificación y representación de haplotipos

Proceso de análisis: Identificación y representación de haplotipos



Hapl.	B3a	B3b	B5	B8	B9a	B9b	B9c	B10a	B10b	E5	E10
1	1	0	1	1	1	1	0	0	0	0	0
2	1	0	1	1	1	0	1	1	0	0	0
3	1	0	1	1	1	0	0	1	1	0	0
4	1	0	1	1	1	0	0	1	0	0	0
5	1	0	1	1	0	1	0	1	0	0	0
6	1	0	1	1	0	1	0	0	0	0	0
7	1	0	1	1	0	0	1	1	0	1	1
8	1	0	1	1	0	0	1	1	0	1	0
9	1	0	1	1	0	0	1	1	0	0	0
10	1	0	1	1	0	0	1	0	1	0	0
15	1	0	1	1	0	0	1	0	0	0	1
16	1	0	1	1	0	0	1	0	0	0	0
11	1	0	1	1	0	0	0	1	1	0	0
17	1	0	0	1	1	0	1	1	0	0	0
18	1	0	0	0	0	1	0	1	0	0	0
19	0	1	1	1	1	1	0	1	1	0	0
20	0	1	1	1	0	1	0	1	0	1	1
12	0	1	1	1	0	1	0	1	0	0	0
13	0	1	1	1	0	0	1	1	0	1	1
14	0	1	1	1	0	0	1	1	0	0	0
21	0	1	1	1	0	0	1	0	0	0	0

Proceso de análisis: Definición de medidas de distancia



Proceso de análisis: Definición de medidas de distancia

$$d_{ij} = dB3_{ij} + dB5_{ij} + dB8_{ij} + dB9_{ij} + dB10_{ij} + dBE5_{ij} + dBE10_{ij}$$

donde:

$dB3_{ij} = (|B3a_i - B3a_j| + |B3b_i - B3b_j| + |B3a_i - B3a_j + B3b_i - B3b_j|)/2$ *(excepción 1)*

$dB5_{ij} = |B5_i - B5_j|$ *(dist. de Hamming)*

$dB8_{ij} = |B8_i - B8_j|$ *(dist. de Hamming)*

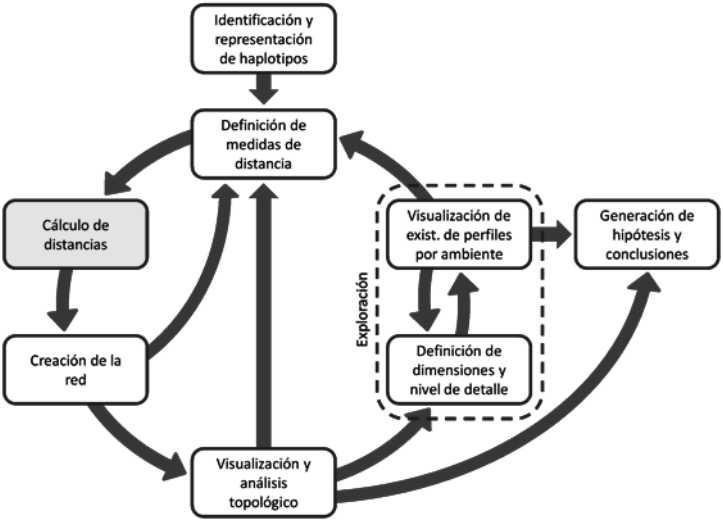
$dB9_{ij} = (|B9a_i - B9a_j| + |B9b_i - B9b_j| + |B9c_i - B9c_j| + |B9a_i - B9a_j + B9b_i - B9b_j + B9c_i - B9c_j|)/2$ *(excepción 1)*

$dB10_{ij} = (|B10a_i - B10a_j| + |B10b_i - B10b_j| + |B10a_i - B10a_j + B10b_i - B10b_j|)/2$ *(excepción 1)*

$dBE5_{ij} = |BE5_i - BE5_j| (1 - |B5_i - B5|)$ *(excepción 3)*

$dBE10_{ij} = |BE10_i - BE10_j| (1 - (|B3a_i . B3a_j| \text{ OR } |B3b_i . B3b_j|))$ *(excepción 2)*

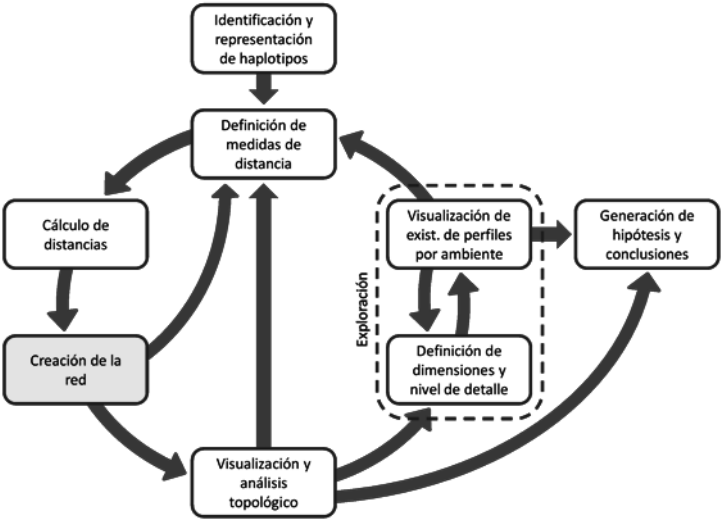
Proceso de análisis: Cálculo de distancias



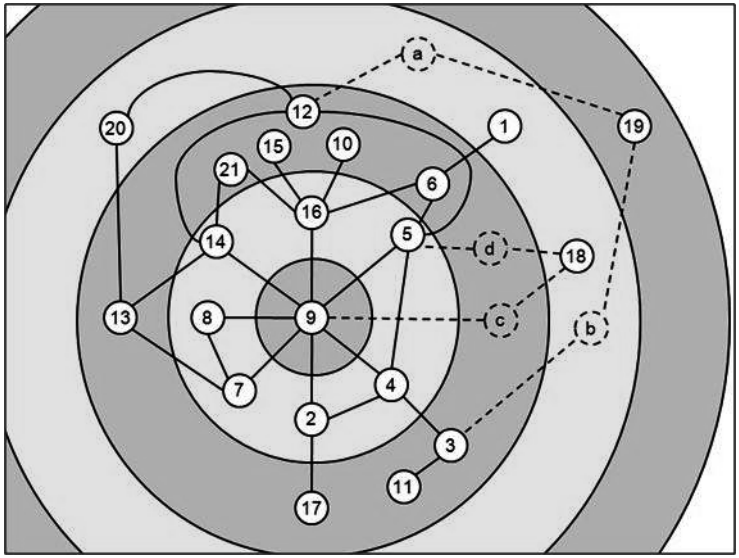
Proceso de análisis: Cálculo de distancias

Hapl.	6	1	5	18	8	7	16	9	10	15	11	3	4	2	17	13	20	21	14	12	19
6		1	1	3	3	3	1	2	2	2	3	3	2	3	4	4	3	2	3	2	4
1			2	4	4	4	2	3	3	3	4	3	2	2	3	5	4	3	4	3	3
5				2	2	2	2	1	3	3	2	2	1	2	3	3	2	3	2	1	3
18					4	4	4	3	5	5	4	4	3	4	3	5	4	5	4	3	5
8						1	2	1	3	2	3	3	2	2	3	2	3	3	2	3	5
7							2	1	3	2	3	3	2	2	3	1	2	3	2	3	5
16								1	1	1	3	3	2	2	3	3	4	1	2	3	5
9									2	2	2	2	1	1	2	2	3	2	1	2	4
10										2	2	2	3	3	4	4	5	2	3	4	4
15											4	4	3	3	4	3	4	2	3	4	6
11												1	2	3	4	4	4	4	3	3	3
3													1	2	3	4	4	4	3	3	2
4														1	2	3	3	3	2	2	3
2															1	3	4	3	2	3	3
17																4	5	4	3	4	4
13																	1	2	1	2	4
20																		3	2	1	3
21																			1	2	4
14																				1	3
12																					2
19																					

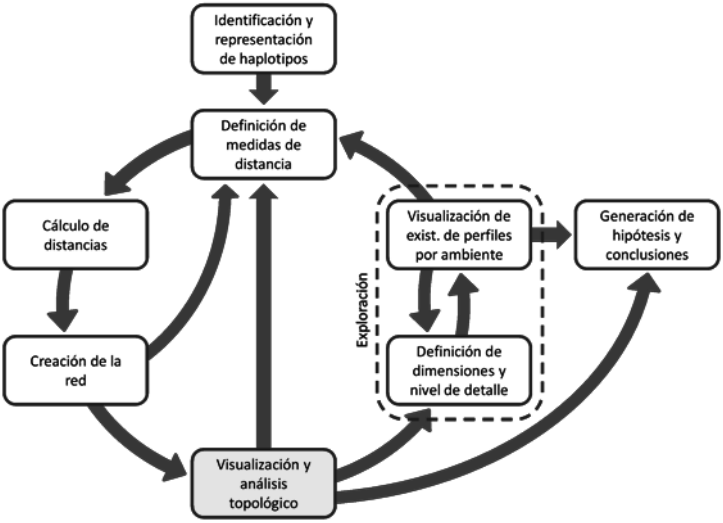
Proceso de análisis: Creación de la red



Proceso de análisis: Creación de la red



Proceso de análisis: Visualización y análisis topológico



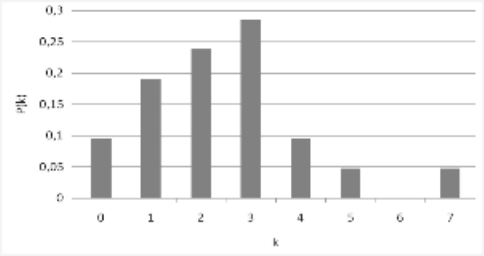
Proceso de análisis: Visualización y análisis topológico

Clustering coefficient: $Cc_i = \frac{2c_i}{k_i(k_i - 1)}$ $CC = \frac{\sum_{i=1}^{n} Cc_i}{n} = 0{,}246$

Diámetro = 5

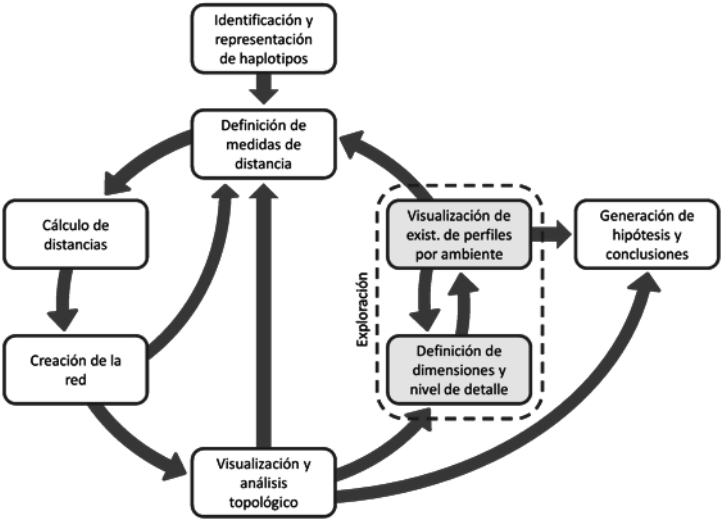
Distancia promedio = 2,767

Distribución de grado de conectividad:

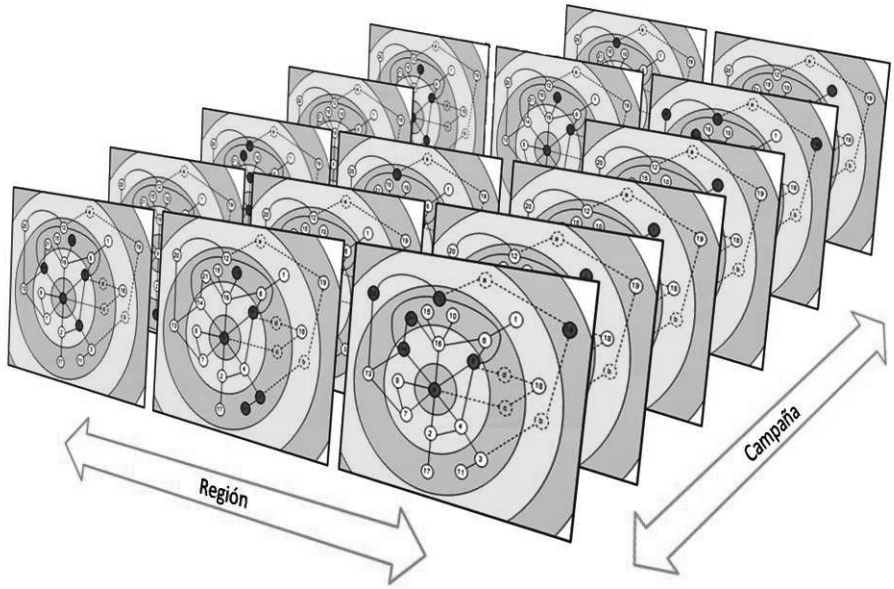


$$k_i = \sum_{j=1}^{N} d_{ij} \ |d_{ij} = 1$$

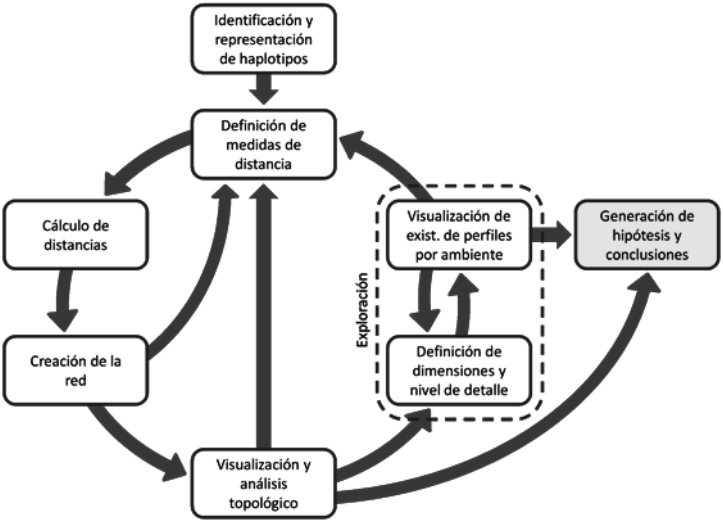
Proceso de análisis: Exploración



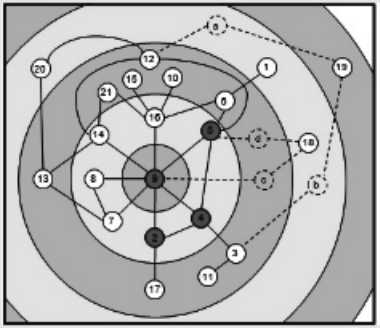
Proceso de análisis: Exploración

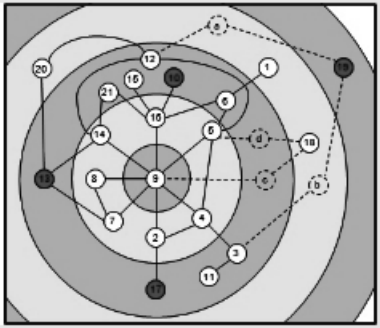


Proceso de análisis: Generación de hipótesis y conclusiones



Proceso de análisis: Generación de hipótesis y conclusiones





$$SDH_A - \sum_{i\ 1}^{n_A-1} \sum_{j\ i+1}^{n_A} d_{ij}$$

donde:

SDH_A: suma de distancias entre los haplotipos del ambiente A

n_A: cantidad de haplotipos del ambiente A

d_{ij}: distancia entre el haplotipo i y el haplotipo j

Proceso de análisis: Generación de hipótesis y conclusiones

$$E(SDH_A) - \sum_{i\ 1}^{n_A-1} \sum_{j\ i+1}^{n_A} \left(1-\left(1-P(h_i)\right)^{n_A}\right)\left(1-\left(1-P(h_j)\right)^{n_A}\right) d_{ij}$$

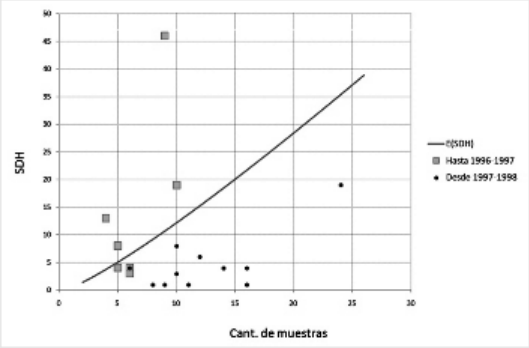
donde:

$E(SDH_A)$: *valor esperado de SDH del ambiente A*

n_A: *cantidad de haplotipos del ambiente A*

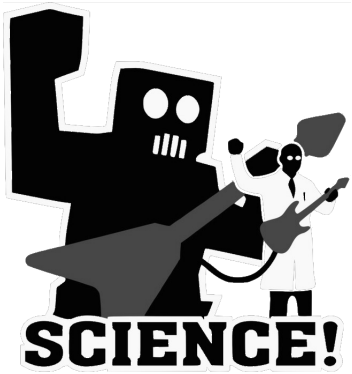
d_{ij}: *distancia entre el haplotipo i y el haplotipo j*

$P(h_i)$: *Probabilidad de existencia del haplotipo i*



Proceso de análisis: Conclusiones del proyecto

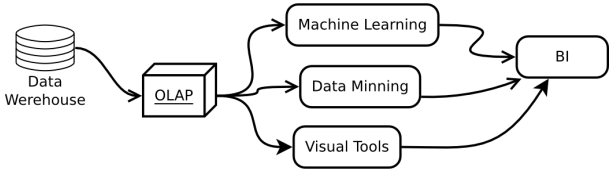
- Según el índice calculado, la variabilidad del Mal de Río Cuarto virus, ha disminuido con el tiempo, habiendo una clara división del indicador en la campaña posterior a la epidemia de la campaña 1996/97.
- La utilización de redes en el proceso de KDD resultó muy satisfactoria y logró resaltar un comportamiento del objeto de estudio que no había sido evidente hasta el momento.
- En un proceso centrado en la persona (human-centered), donde la creatividad y experiencia del analista juega un rol fundamental, la herramienta propuesta es capaz de ofrecer una perspectiva novedosa y complementaria con las demás técnicas del proceso de KDD



Data Warehouse - OLAP - BI

Concluimos que nuestro problema se adaptaba a algo muy similar a "algo" de BI

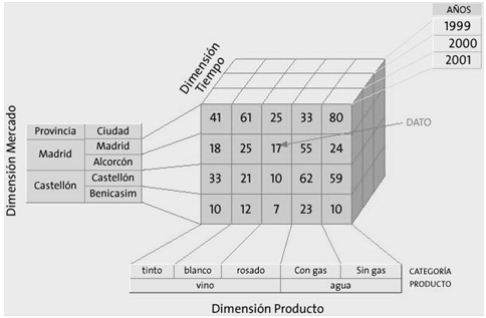
- En el contexto de la informática, un **almacén de datos** (del inglés data warehouse) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.
- **OLAP** es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing). Es una solución utilizada en el campo de la llamada Inteligencia empresarial (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales (o **Cubos OLAP**) que contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP). Se usa en informes de negocios de ventas, marketing, informes de dirección, minería de datos y áreas similares.



- Se denomina **inteligencia empresarial**, inteligencia de negocios o BI (del inglés business intelligence) al conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

Cubo OLAP

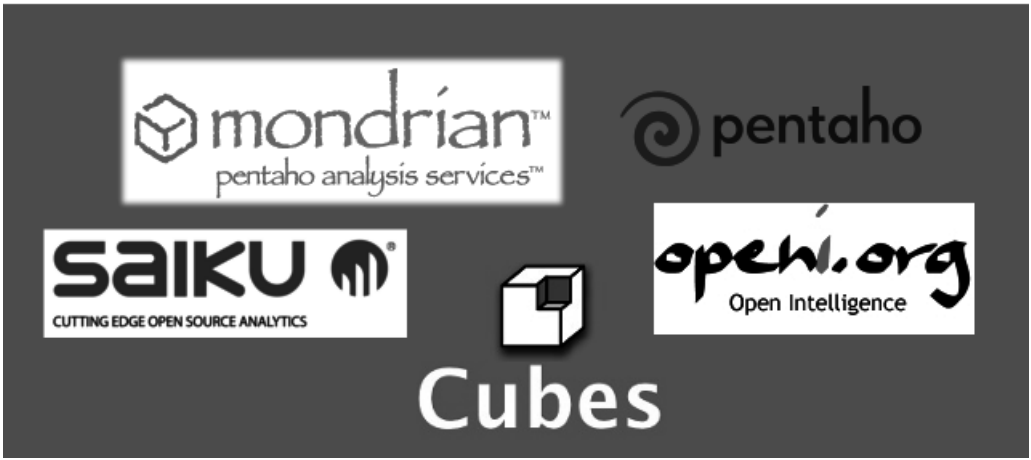
Es una base de datos multidimensional, en la cual el almacenamiento físico de los datos se realiza en un vector multidimensional. Los cubos OLAP se pueden considerar como una ampliación de las dos dimensiones de una hoja de cálculo.



- Las olap pueden ser implementados en ROLAP - MOLAP - HOLAP
- Las consultas OLAP se llaman MDX (son bastante parecidas a SQL)
- Para consultas remotas se utiliza XMLA sobre SOAP

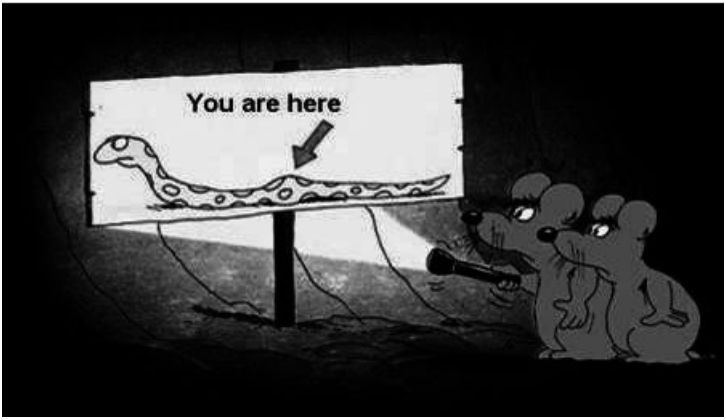
Soluciones OpenSource

- **Mondrian** (todo en java soporta XML y MDX) (http://mondrian.pentaho.com/)
- **python xmla** (`pip install xmla`) para comunicarse con casi cualquier OLAP (https://pypi.python.org/pypi/xmla/)
- **Cubes** (`pip install cubes`) puro python pero muy verde (pythonhosted.org/cubes/)
- **Pentaho** (http://www.pentaho.com/), **Saiku** (http://meteorite.bi/saiku), **OpenI** (http://openi.org/)

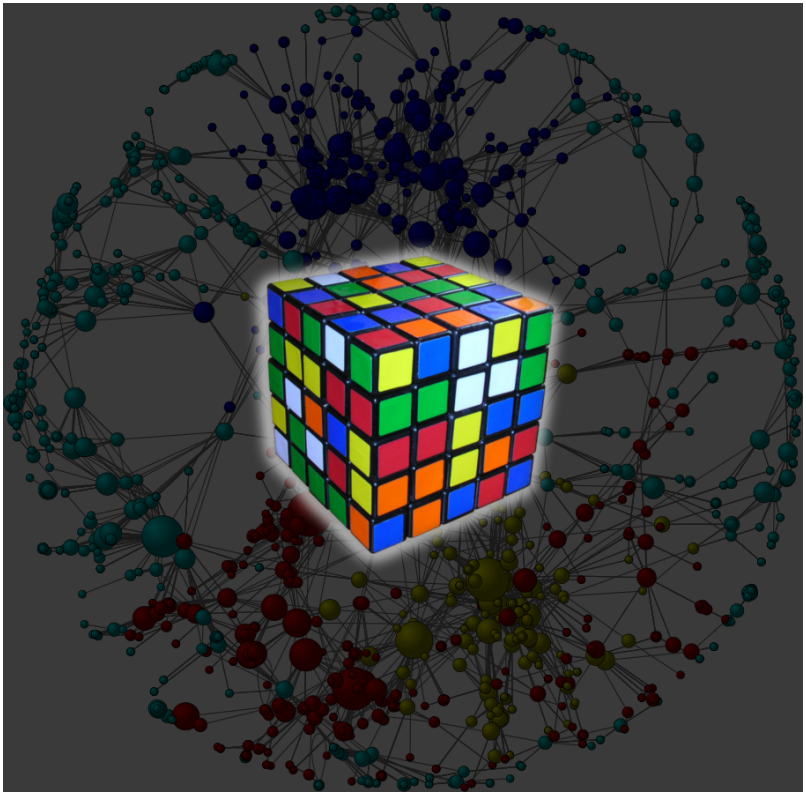


Y donde estamos con nuestro problema

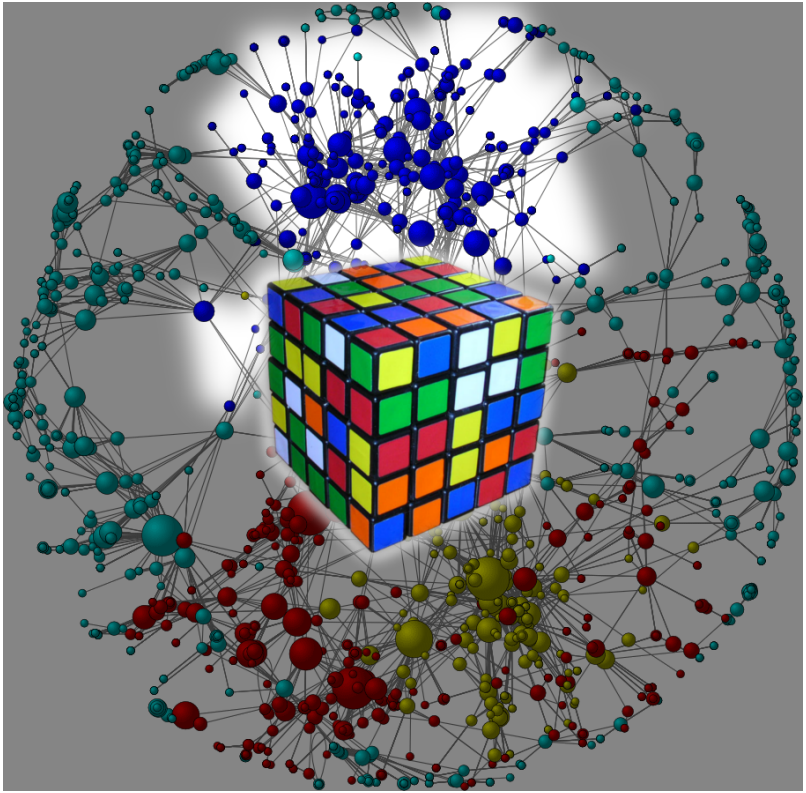
- Los Sql eran muy engorrosos (http://wiki.getyatel.org/analysis/exp2014/)
- Para Cubos y BI en general, la solución no era natural.
- Las bases de datos de redes no son tan difundidas como las RDBMS.
- Solución Nuevo Paradigma: **NW-OLAP**



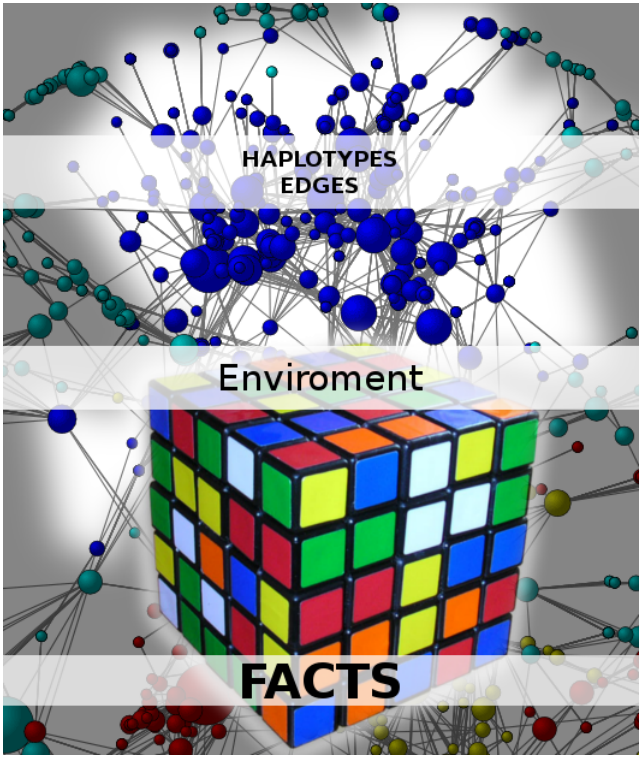
Red OLAP (*NW-OLAP*)



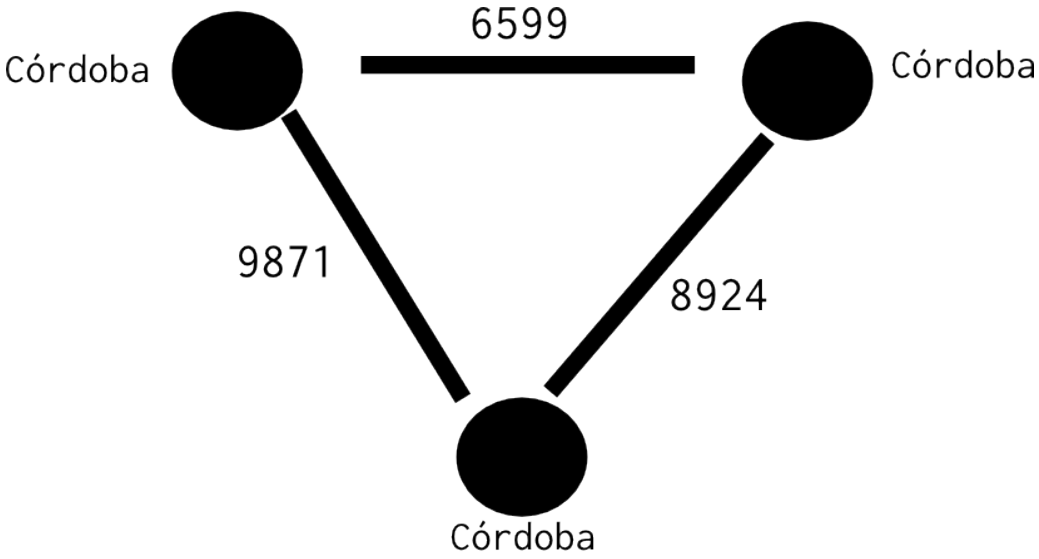
Red OLAP (*NW-OLAP*) Seleccionada



Red OLAP (*NW-OLAP*) Partes

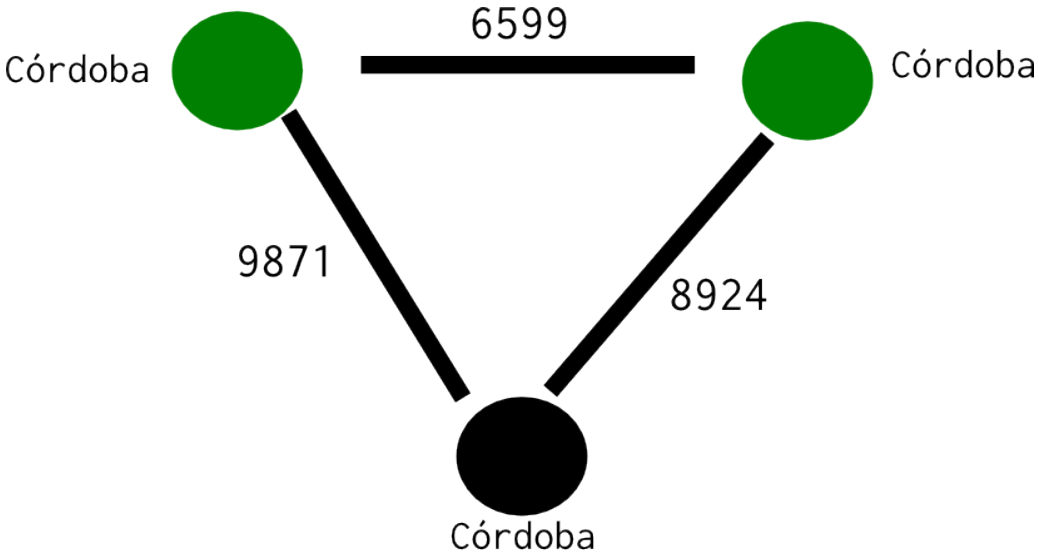


Ejemplo: ¿Cual Córdoba es cada una? #1



Query:

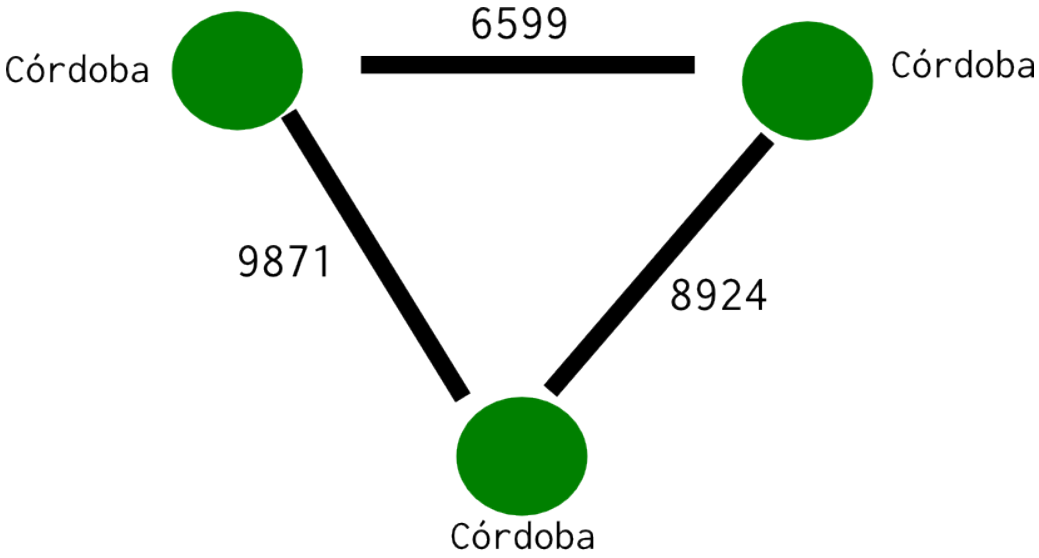
Ejemplo: ¿Cual Córdoba es cada una? #2



Query:

```
superficie entre 200 Km2 y 600 km2
```

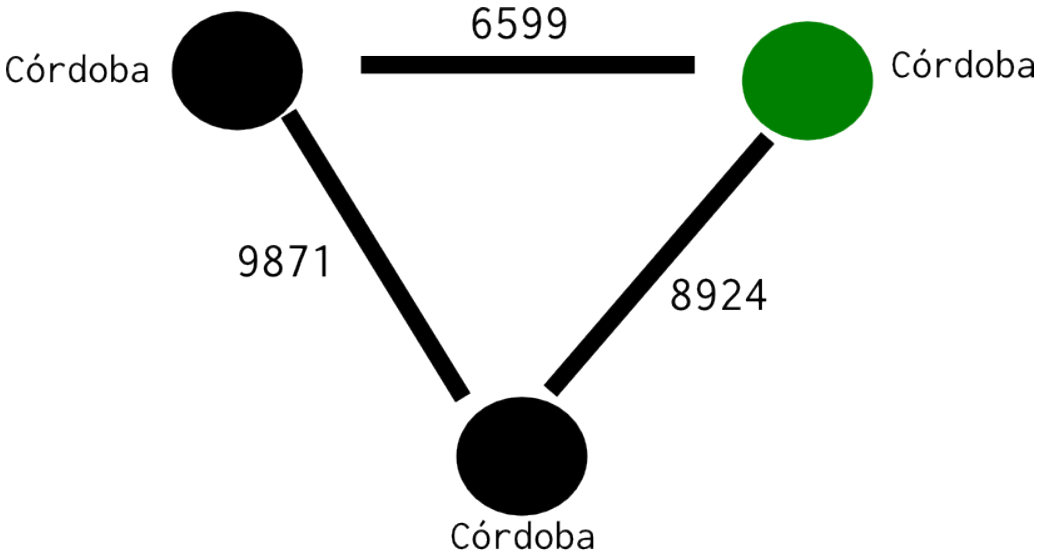
Ejemplo: ¿Cual Córdoba es cada una? #3



Query:

hablan Español

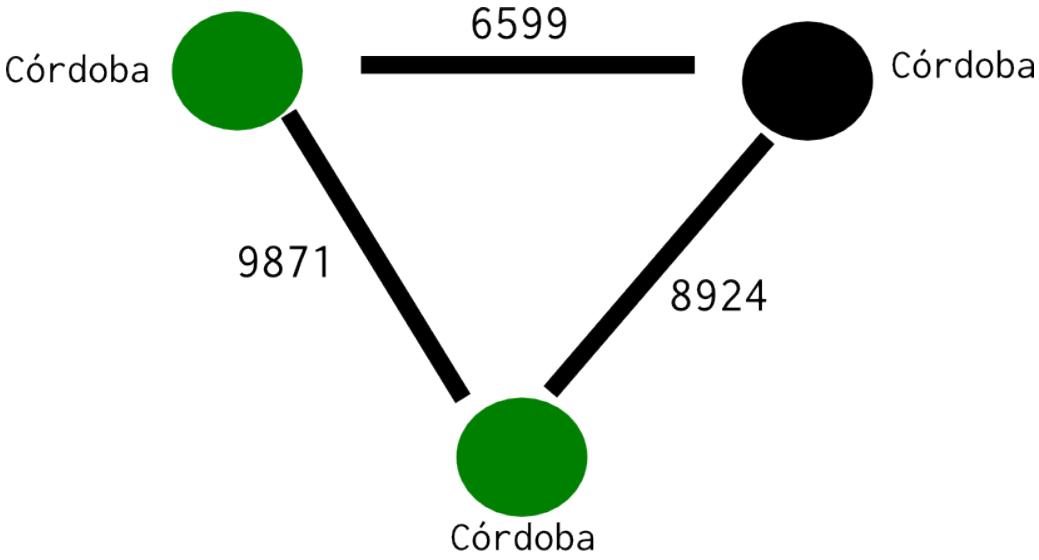
Ejemplo: ¿Cual Córdoba es cada una? #4



Query:

```
uso horario utc-6
```

Ejemplo: ¿Cual Córdoba es cada una? #5



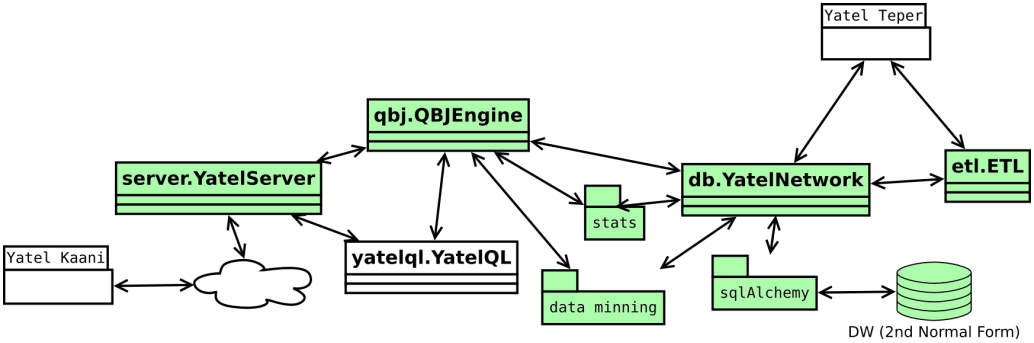
Query:

Su nombre contiene Andalucia

Implementación

- Homepage: http://getyatel.org
- Es una implementación de referencia de NW-OLAP
- Wiskey-Ware License
- Es la implementación en gran parte del proceso mencionado anteriormente.
- Pronto a salir la primer version usable 0.3

Yatel - Arquitectura



- Posee un lenguaje intermedio denominado QBJ.
- Puede usarse como librería o como DB remota (alpha)
- Posee soportes de ETL, estadísticas y DM rudimentaria.
- Exporta e importa las DW en formatos agnósticos basados en JSON y XML

Yatel - Ejemplo - Creación

```
from yatel import dom, db

# postgres, oracle, mysql, and many more
nw = db.YatelNetwork("memory", mode="w")

elems = [
    dom.Haplotype(0, name="Cordoba"), # left
    dom.Haplotype(1, name="Cordoba"), # right
    dom.Haplotype(2, name="Cordoba"), # bottom

    dom.Edge(6599, (0, 1)),
    dom.Edge(8924, (1, 2)),
    dom.Edge(9871, (2, 0)),

    dom.Fact(0, name="Andalucia", lang="sp", timezone="utc-3"),
    dom.Fact(1, lang="sp"),
    dom.Fact(1, timezone="utc-6"),
    dom.Fact(2, name="Andalucia", lang="sp", timezone="utc")
]

nw.add_elements(elems)
nw.confirm_changes()
```

Yatel - Ejemplo - Consultas #1

```
print nw.describe()
# {
#   'haplotype_attributes': { 'hap_id': <type 'int'>, 'name': <type 'str'>},
#  'fact_attributes': { 'lang': <type 'str'>, 'timezone': <type 'str'>,
#                       'hap_id': <type 'int'>, 'name': <type 'str'>},
#   'mode': 'r',
#   'edge_attributes': {u'max_nodes': 2, u'weight': <type 'float'>},
#   'size': {u'facts': 4, u'haplotypes': 3, u'edges': 3}
# }

for hap in nw.haplotypes():
    print hap
# <Haplotype (0) at 0x2cb8710>
# <Haplotype (1) at 0x2cb8810>
# <Haplotype (2) at 0x2cb8850>

for edge in nw.edges():
    print edge
# <Edge ([6599.0 [0, 1]]  ) at 0x2cb64d0>
# <Edge ([8924.0 [1, 2]]  ) at 0x2cb6dd0>
# <Edge ([9871.0 [2, 0]]  ) at 0x2cb6fd0>

for fact in nw.facts():
    print fact
# <Fact (of Haplotype '0') at 0x2cb6f50>
# <Fact (of Haplotype '1') at 0x2cb6ed0>
# <Fact (of Haplotype '1') at 0x2cb6c50>
# <Fact (of Haplotype '2') at 0x2cb6e90>
```

Yatel - Ejemplo - Consultas #2

```
hap = nw.haplotype_by_id(2)

for edge in nw.edges_by_haplotype(hap):
    print edge
# <Edge ([9871.0 [2, 0]]  ) at 0x1cf6910>,
# <Edge ([8924.0 [1, 2]]  ) at 0x1cf6810>

for fact in nw.facts_by_haplotype(hap):
    print dict(fact)
# {u'timezone': u'utc', u'lang': u'sp', 'hap_id': 2, u'name': u'Andalucia'}

for hap in nw.haplotypes_by_enviroment(lang="sp"):
    print hap
# <Haplotype (0) at 0x254bfd0>
# <Haplotype (1) at 0x254bc10>
# <Haplotype (2) at 0x254bfd0>

for hap in nw.haplotypes_by_enviroment(timezone="utc-6"):
    print hap
# <Haplotype (1) at 0x14a8210>

for hap in nw.haplotypes_by_enviroment(name="Andalucia"):
    print hap
# <Haplotype (0) at 0x254bb50>
# <Haplotype (2) at 0x254bfd0>
```

Yatel - Ejemplo - Consultas, Estadisticas

```
for edge in nw.edges_by_enviroment(name="Andalucia"):
    print edge
# <Edge ([9871.0 [2, 0]]  ) at 0x23e3ad0>

for env in nw.enviroments():
    print env
# <Enviroment {u'lang': u'sp', u'timezone': u'utc-3', u'name': u'Andalucia'} at 0x1f6b490>
# <Enviroment {u'lang': u'sp', u'timezone': None, u'name': None} at 0x1f6b810>
# <Enviroment {u'lang': None, u'timezone': u'utc-6', u'name': None} at 0x1f6b490>
# <Enviroment {u'lang': u'sp', u'timezone': u'utc', u'name': u'Andalucia'} at 0x1f6b810>

for env in nw.enviroments(["lang", "name"]):
    print env
# <Enviroment {u'lang': u'sp', u'name': u'Andalucia'} at 0x1aa4950>
# <Enviroment {u'lang': u'sp', u'name': None} at 0x1aa45d0>
# <Enviroment {u'lang': None, u'name': None} at 0x1aa4950>

from yatel import stats

print stats.average(nw)
# 8464.66666667
print stats.std(nw, name="Andalucia")
# 0
```

Yatel - Ejemplo - Data Minning

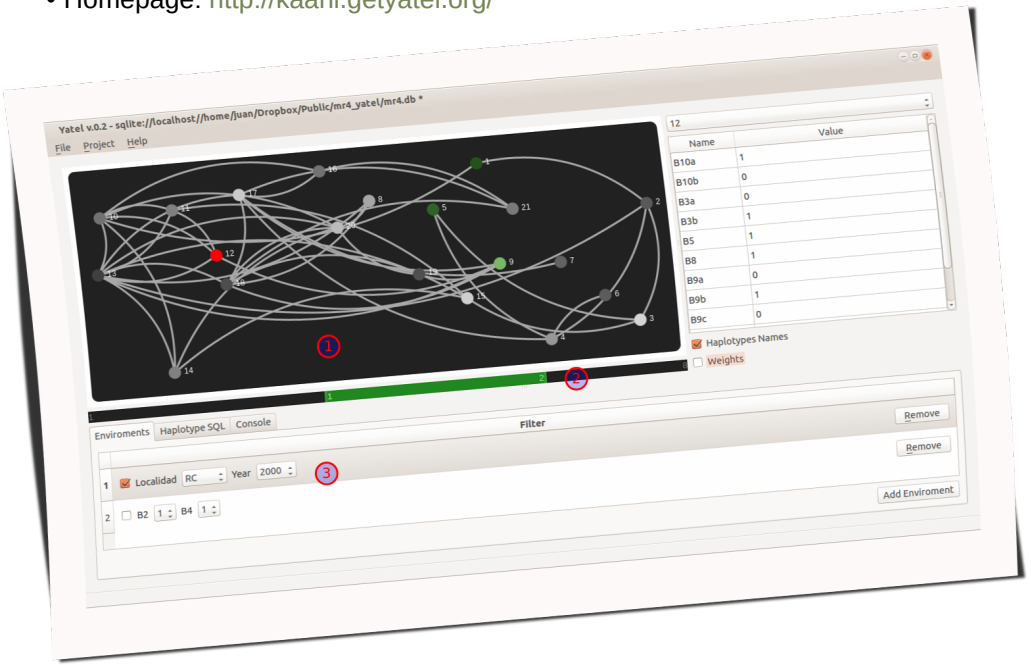
```
from scipy.spatial.distance import euclidean
from yatel.cluster import kmeans

cbs, distortion = kmeans.kmeans(nw, nw.enviroments(), 2)

for env in nw.enviroments():
    coords = kmeans.hap_in_env_coords(nw, env)
    min_euc = None
    closest_centroid = None
    for cb in cbs:
        euc = euclidean(cb, coords)
        if min_euc is None or euc < min_euc:
            min_euc = euc
            closest_centroid = cb
    print "{} || {} || {}".format(dict(env), closest_centroid, euc)
# {u'lang': u'sp', u'timezone': u'utc-3', u'name': u'Andalucia'} || [0 0 0] || 1.41421356237
# {u'lang': u'sp', u'timezone': None, u'name': None} || [0 1 0] || 0.0
# {u'lang': None, u'timezone': u'utc-6', u'name': None} || [0 1 0] || 0.0
# {u'lang': u'sp', u'timezone': u'utc', u'name': u'Andalucia'} || [0 0 0] || 1.41421356237
```

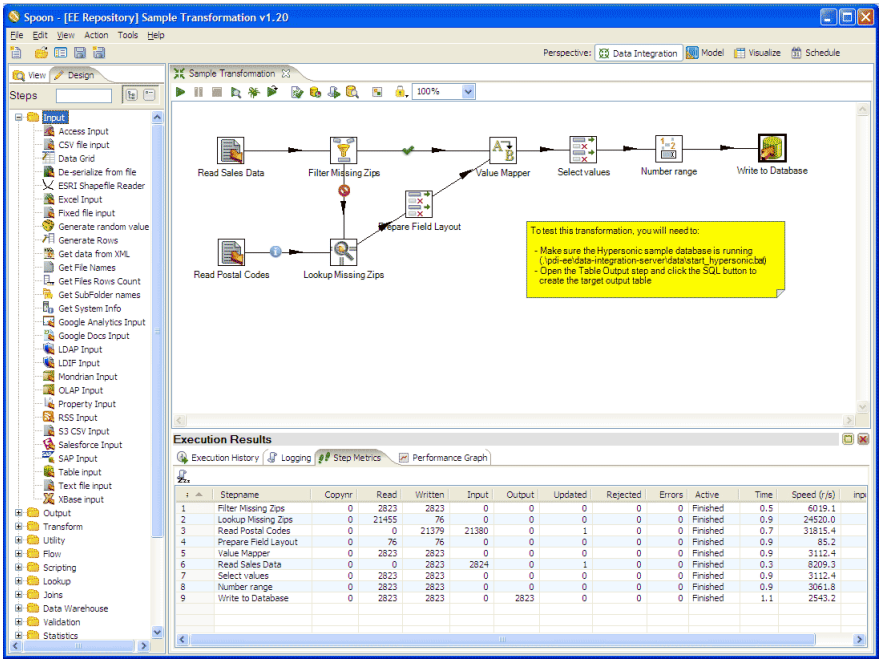
Pendientes: Yatel Kaani - (Not even started)

- En funcionamiento... (0.2 algo así va a ser Yatel BI)
- Homepage: http://kaani.getyatel.org/



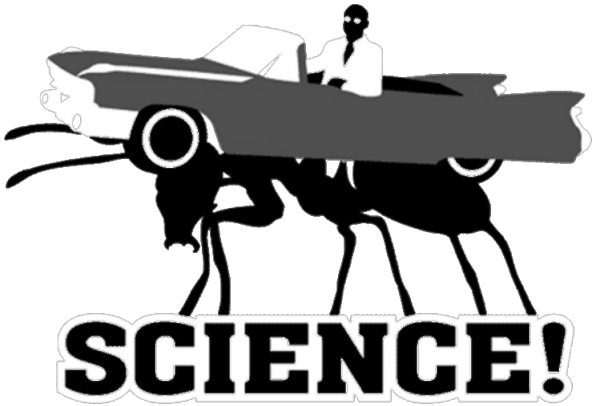
Pendientes: Yatel Teper - (Not even started)

- Va a ser el entorno visual de ETL
- Va a permitir ordenar fuentes para alimentar DW NW-OLAP



Más pendientes

- YatelQL sin implementar.
- Más minería de datos propiamente dicha (solo tiene kmeans)
- Agregar autenticación rudimentaria en yatel server y dar soporte a algo como... LDAP¿?¿?.
- La parte científica per-se no esta desarrollada (algún doctorando en la sala?)
- Documentación (mucha)
- Testing.



¿Preguntas?

- Charla: http://goo.gl/hTkdzj
- **Contactos:**
  - http://forum.getyatel.org
  - Alejandro Garcia <malegrandogarcia@hotmail.com>