MINING VIRTUAL UNIVERSES IN A RELATIONAL DATABASE

With examples from the

(milli-)Millennium Run Database(s)

Gerard Lemson
MPA Garching, Germany

Matt's categorization of questions

- Questions:
  - Phrase questions in terms of physics
  - Formulate question in terms of data
  - Translate in terms of existing tools
- Simple Questions
  - Can be answered using available data+tools
- Hard questions
  - Those for which tools do not exist
- Impossible
  - Those for which data is not sufficient

Use database to make questions simple(r)

- Provide data and tools for accessing them
  - yt is such an approach tailored for yt-like data

- We try to make questions as simple as possible
  - Data represented in a database using standard relational techniques
  - Query tool through online interfaces.
  - Standard query language SQL makes translating physics question into data question simple

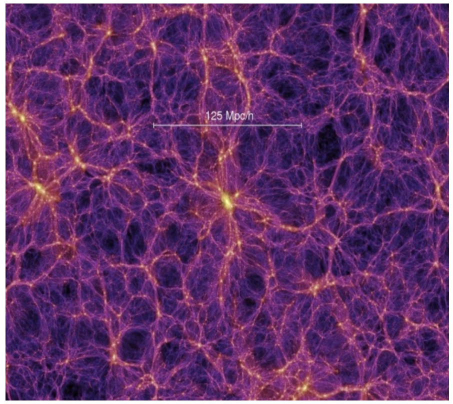
Warning

It’s not always as simple as it seems.
One should *try* to understand the data when using the database

See
Kevin Bundy, Tommaso Treu, Richard Ellis

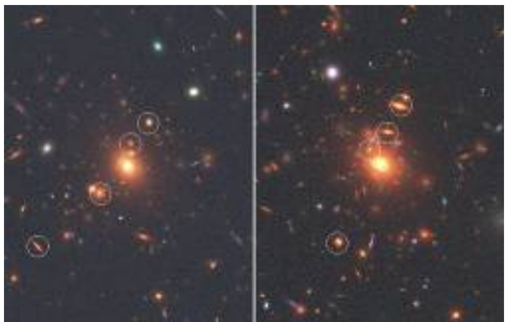
http://astro.berkeley.edu/~kbundy/millennium/

Raw data: Particles

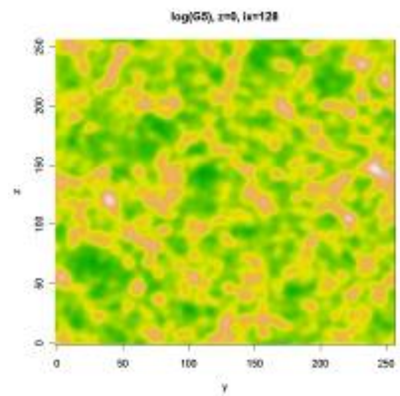


FOF groups and Subhalos

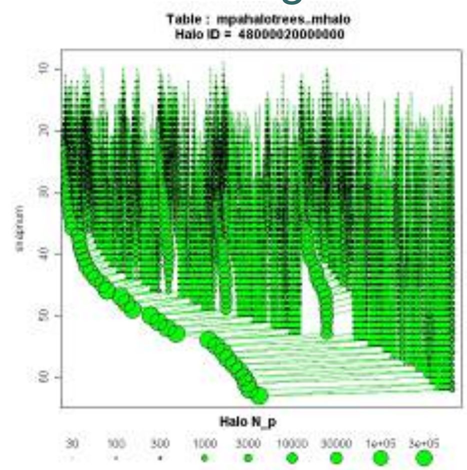
Mock images



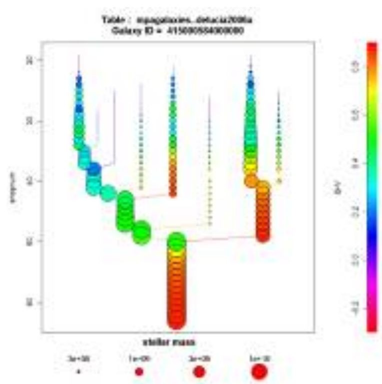
Density fields

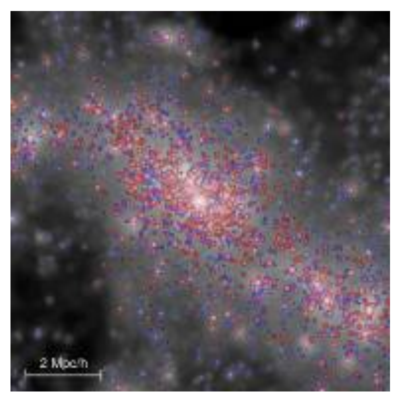


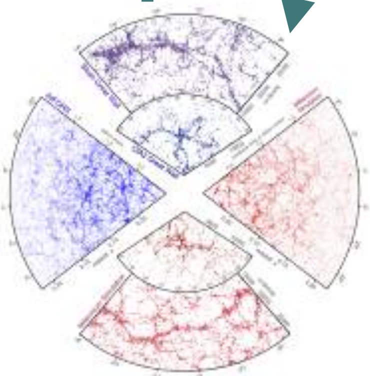
Subhalo merger trees



Synthetic galaxies (SAM)







Mock catalogues

MOTIVATION: WHY RELATIONAL DATABASE

Analysis and Databases

(courtesy Alex Szalay)

- Much statistical analysis of data deals with
  - Creating uniform samples
  - Data filtering
  - Assembling relevant subsets
  - Estimating completeness
  - censoring bad data
  - Counting and building histograms
  - Generating Monte-Carlo subsets
  - Likelihood calculations
  - Hypothesis testing
- Traditionally these are performed on files
- Most of these tasks are much better done inside a database

Relational database offers …

- Encapsulation of data in terms of logical structure
  - *no need to know about internals of data storage*
- *Standard query language* for finding information
- Advanced *query optimizers* (indexes, clustering)
- Transparent internal *parallelization*
- Authenticated remote access for multiple users at same time

Especially important

- Forces one to think carefully about data structure
- Speeds up path from science question to answer
  - Makes more questions *simple*
- Facilitates communication
  - query code is clean(er)

Max-Planck-Institut für Astrophysik

RDB CONCEPTS

Relational database stores data in *relations* (= tables)

http://www.g-vo.org - Millennium Run Database - Microsoft Internet Explorer

File Edit View Favorites Tools Help

galaxyid	haloid	descendantid	redshift	x	y	z	np
0	0	-1	0.0	6.5757904	13.08604	25.33813	51984
1	1	0	0.019932542	6.587909	13.099106	25.301092	51288
2	2	1	0.041403063	6.597178	13.111782	25.252974	51052
3	3	2	0.064493395	6.615912	13.121013	25.204876	51169
4	4	3	0.08928783	6.6276503	13.1303835	25.152872	50870
5	5	4	0.11588337	6.6414022	13.1400175	25.09534	50468
6	6	5	0.14438343	6.658701	13.149509	25.03174	50168
7	7	6	0.17489761	6.642237	13.170146	24.927555	50485
8	8	7	0.20754863	6.6424794	13.18374	24.83325	49888
9	9	8	0.24246909	6.6978354	13.176765	24.781622	48275

Tables

- Tables have names
  - Full path:
    [<database-name>.]<schema-name>.<table-name>
- *Related* data values are stored in rows
- Rows have columns
  - all the same for a given table
- Columns have names and data types
  - Data types have SQL names:
    SMALLINT, INTEGER, BIGINT,
    REAL, FLOAT, DECIMAL,
    CHAR(10), VARCHAR(100), CLOB, BLOB,
    DATETIME, TIME, TIMESTAMP, ….
- Rows often have a unique identifier consisting of the values of >= 1 columns: *primary key*

Primary Key Column

Column

Foreign Key Columns

http://www.g-vo.org - Millennium Run Database - Microsoft Internet Explorer

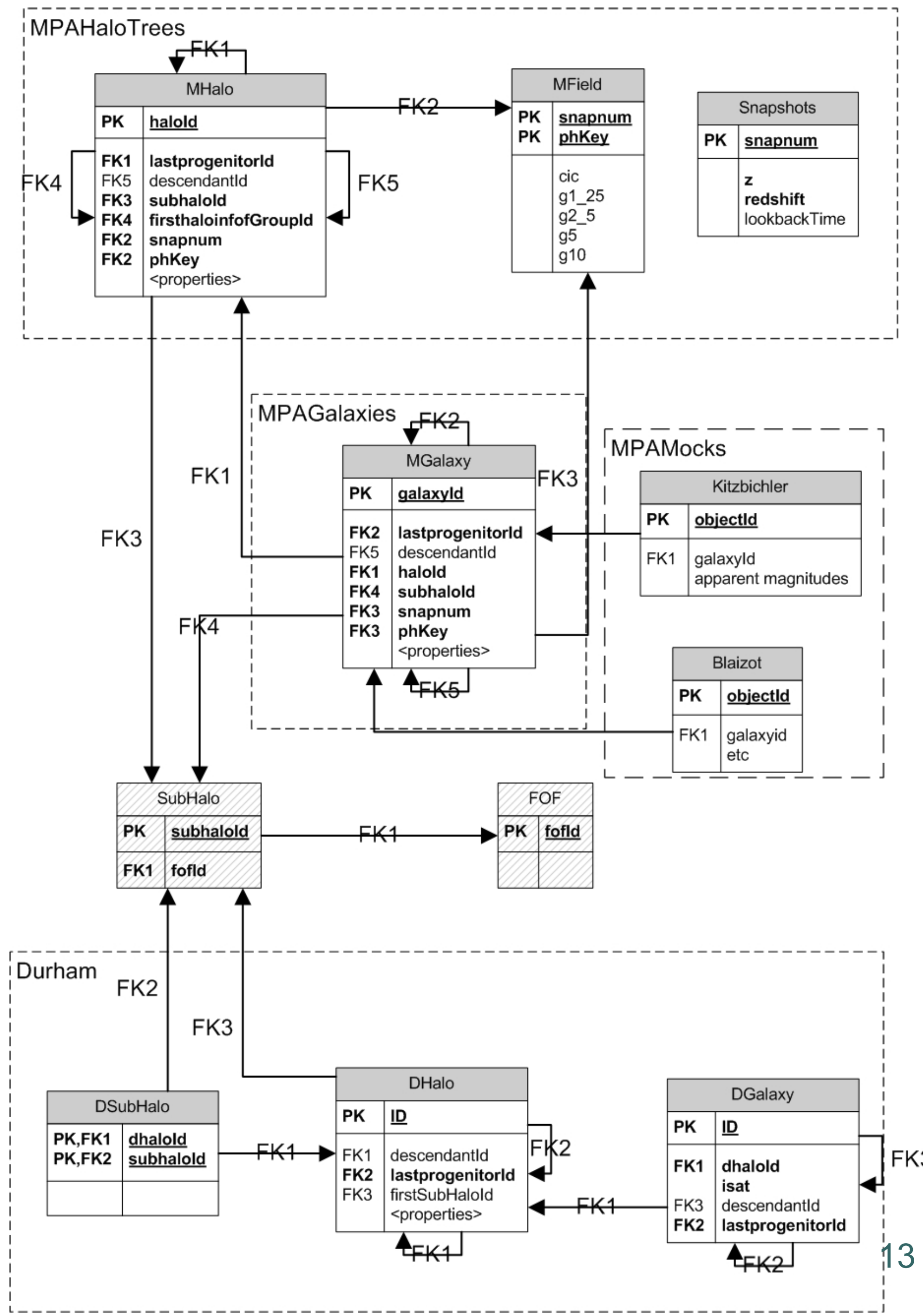
File Edit View Favorites Tools Help

galaxyid	haloid	descendantid	redshift	x	y	z	np
0	0	-1	0.0	6.5757904	13.08604	25.33813	51984
1	1	0	0.019932542	6.587909	13.099106	25.301092	51288
2	2	1	0.041403063	6.597178	13.111782	25.252974	51052
3	3	2	0.064493395	6.615912	13.121013	25.204876	51169
4	4	3	0.08928783	6.6276503	13.1303835	25.152872	50870
5	5	4	0.11588337	6.6414022	13.1400175	25.09534	50468
6	6	5	0.14438343	6.658701	13.149509	25.03174	50168
7	7	6	0.17489761	6.642237	13.170146	24.927555	50485
8	8	7	0.20754863	6.6424794	13.18374	24.83325	49888
9	9	8	0.24246909	6.6978354	13.176765	24.781622	48275

Row

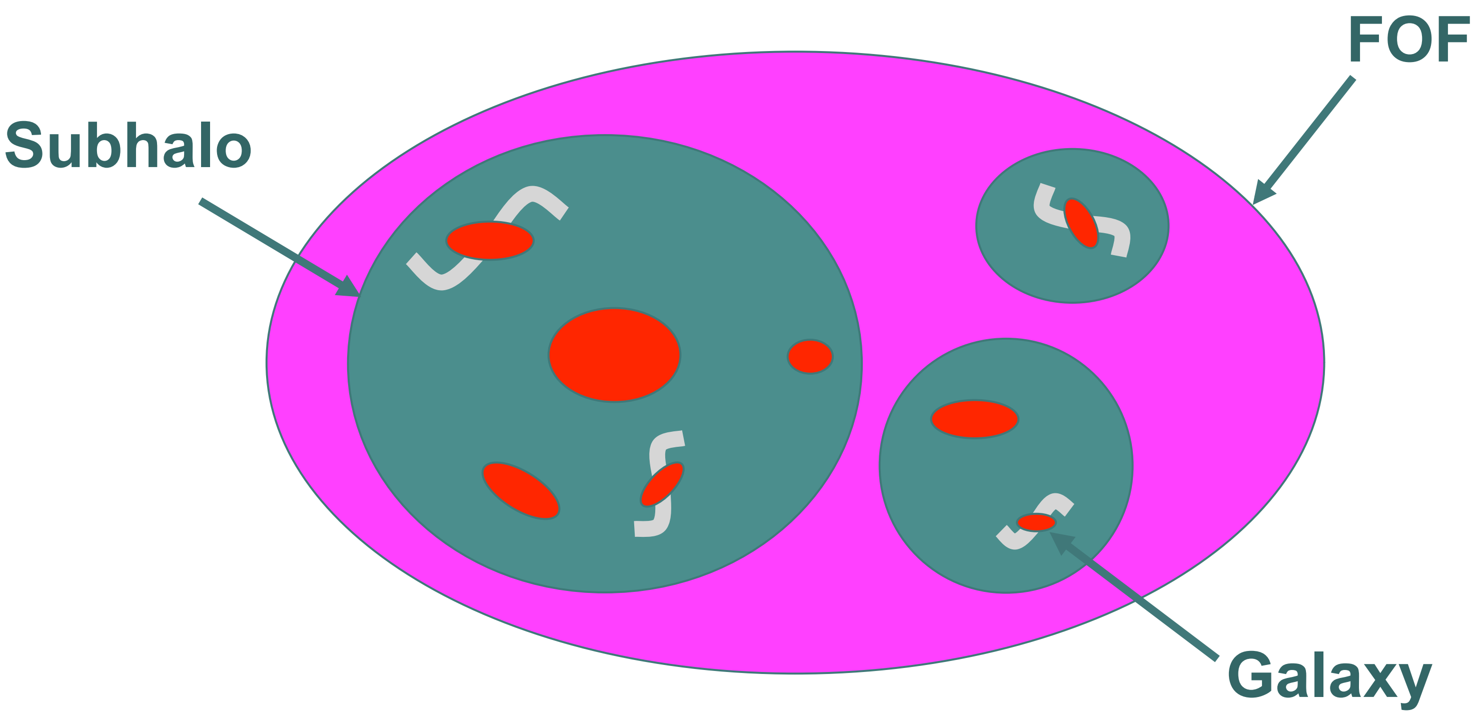
Database

- Many tables in >=1 *schemas*.
- Related through *foreign keys*
- Why so complex?



Normalization

- Consider storing galaxies, with info about their sub-halo as well as the FOF groups these live in.
  Note, a subhalo contains >=1 galaxies,
  a FOF group >= 0 subhalos



One table: redundancy

GalaxySubhaloFOF

galId	mStar	magB	X	haloId	np	hX	vMax	fofId	nSub	m200	fX
112	0.215	-17.9	7.6	6625	100	7.6	165	123	2	445.77	7.6
113	0.038	-15.6	7.4	6625	100	7.6	165	123	2	445.77	7.6
154	0.173	-17.1	7.65	6626	65	7.9	130	123	2	445.77	7.6
221	1.20	-20.7	35.1	7883	452	35.1	200	456	2	101.32	35.1
223	0.225	-19.7	35.0	7883	452	35.1	200	456	2	101.32	35.1
225	0.04	-17.5	34.9	7883	452	35.1	200	456	2	101.32	35.1
278	1.54	-19.4	35.2	7884	255	35.2	190	456	2	101.32	35.1
…											

Normalization

FOF

fofId	nSub	m200	x	…
123	2	445.77	7.6	…
456	2	101.32	35.1	…
789	1	70.0	67.0	…
…	…	…	…	…

Galaxy

galId	haloId	mStar	magB	X	…
112	6625	0.215	-17.9	7.6	…
113	6625	0.038	-15.6	7.4	…
154	6626	0.173	-17.1	7.65	…
221	7883	1.20	-20.7	35.1	…
223	7883	0.225	-19.7	35.0	…
225	7883	0.04	-17.5	34.9	…
278	7884	1.54	-19.4	35.2	…
…	…	…	…	…	…

SubHalo

haloId	fofId	Np	X	vMax	…
6625	123	100	7.6	165	…
6626	123	65	7.9	130	…
7883	456	452	35.1	200	…
7884	456	255	35.2	190	…
9885	789	30	67.0	110	…
…	…	…	…	…	…

DATABASE DESIGN

Data model features

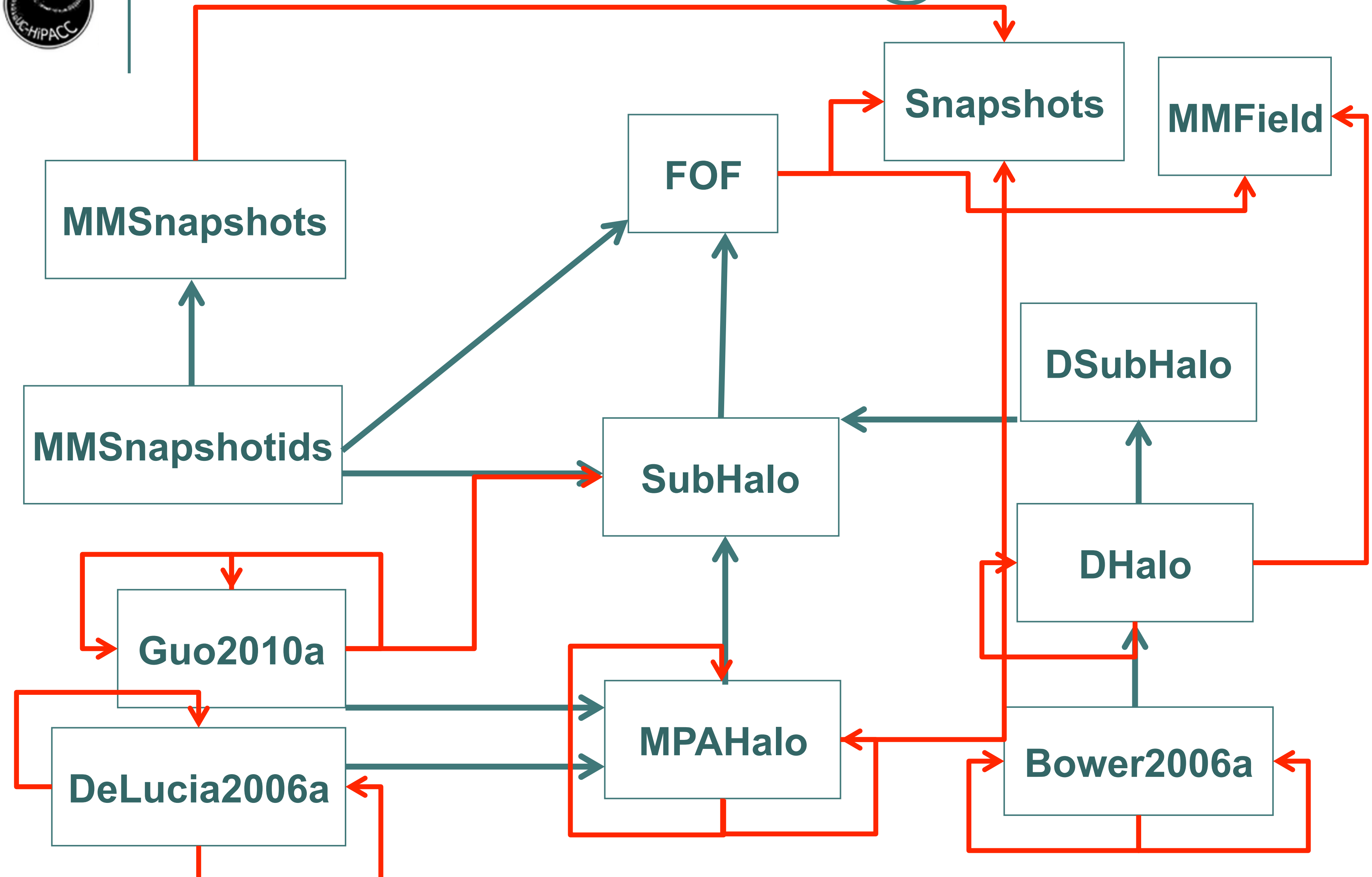
- Each object its table
  - properties are columns
  - each a unique identifer
- Relations implemented through *foreign keys,*
  - pointers to unique identifier column
  - FOF to mesh cell it lies in
  - Subhalo to its FOF group
  - galaxy to its subhalo etc
- Special design needed for
  - Hierarchical relations: merger trees
  - Spatial relations: multi-dimensional indexes required
  - Support for random sample selection

Max-Planck-Institut für Astrophysik

Motivation for data model

1. Return the (B-band luminosity function of) galaxies residing in halos of mass between 10^13 and 10^14 solar masses.
2. Return the galaxy content at z=3 of the progenitors of a halo identified at z=0
3. Return all the galaxies within a sphere of radius 3Mpc around a particular halo
4. Return the complete halo merger tree for a halo identified at z=0
5. Find positions and velocities for all galaxies at redshift zero with B-luminosity, colour and bulge-to-disk ratio within given intervals.
6. Find properties of all galaxies in haloes of mass 10**14 at redshift 1 which have had a major merger (mass-ratio < 4:1) since redshift 1.5.
7. Find all the z=3 progenitors of z=0 red ellipticals (i.e. B-V>0.8 B/T > 0.5)
8. Find the descendents at z=1 of all LBG's (i.e. galaxies with SFR>10 Msun/yr) at z=3
9. Make a list of all haloes at z=3 which contain a galaxy of mass >10**9 Msun which is a progenitor of BCG's in z=0 cluster of mass >10**14.5
10. Find all z=3 galaxies which have NO z=0 descendant.
11. Return the complete galaxy merging history for a given z=0 galaxy.
12. Find all the z=2 galaxies which were within 1Mpc of a LBG (i.e. SFR>10Msun/yr) at some previous redshift.
13. Find the multiplicity function of halos depending on their environment (overdensity of density field smoothed on certain scale)
14. Find the dependency of halo formation times on environment (“Gao-effect”)

millimil database/schema @ISSACTAP



Database tuning: Indexes

- Performance: disk IO is bottleneck
- Avoid it as much as possible, but can not store whole DB in memory
- To find rows of interest, avoid scanning complete tables
  - sequential scan ~ O(N)
  - ~10 min for galaxy tables (10^9 rows, 250 GB)
- Binary search might speed up: requires ordering
  - ~ O(log(N))
- Can only order in one way
  - create external data structure
- INDEX
  - ordered according to >=1 columns, with direct pointer to row.
  - Bookmark lookup may be avoided

Indexes

galaxyId	haloId	descendantId	snapnum	redshift	coldgas	stellarMass	bulgeMass	mag_b	mag_v	mag_r	mag_i	mag_k	phkey	x	y
0	0	-1	63	0.0	0.043959517	34.046864	34.046272	-22.093401	-22.982061	-23.712515	-24.339514	-25.888254	673	6.5[illegible]57904	13.08604
1	1	0	62	0.019932542	0.043959517	34.046864	34.046272	-22.[illegible]31735	-23.01172	-23.73842	-24.362885	-25.908625	673	6.587909	13.09910
2	2	1	61	0.041403063	0.01826[illegible]254	33.75536	33.75536	-22.[illegible]04472	-23.052692	-23.76902	-24.387476	-25.926197	673	6.597178	13.11178
3	3	2	60	0.064493395	0.1[illegible]435881	25.243101	25.165026	-21.[illegible]23284	-22.712221	-23.441523	-24.06735	-25.61783	673	6.615912	13.12101
4	4	3	59	0.08928783	0.11362965	25.235704	25.158258	-21.856045	-22.74137	-23.46809	-24.091352	-25.638094	673	6.6276503	13.13038
5	5	4	58	0.11588[illegible]7	0.11362965	25.235704	25.158258	-21.898903	-22.776686	-23.49952	-24.11984	-25.66352	673	6.6414022	13.14001
6	6	5	57	0.1[illegible]38343	0.074656956	25.228788	25.151812	-2[illegible].9567	-22.818369	-23.534164	-24.14973	-25.686872	673	6.658701	13.14950

snapnum	stellarMass	galaxyid
63	36.805252	3000042000000
63	34.046864	0
63	33.668137	6000033000000
63	33.0581	3000045000000
63	30.739763	1000020000000
63	27.123823	5000047000000
63	26.617165	1000019007260
63	25.67081	3000048000000
63	23.400906	4000014000000
63	23.227219	3000047000000

snapnum, stellarMass, galaxyid

mag_b
-25.599024
-25.53919
-25.375084
-25.308317
-25.298616
-25.116589
-25.024874
-25.004715
-24.928555
-24.900457

mag_b

snapnum	x
8	56.546013
8	40.06858
9	6.3873663
9	6.440719
9	48.860935
9	39.802
9	19.463036
9	57.503075
9	56.608723
9	40.01735

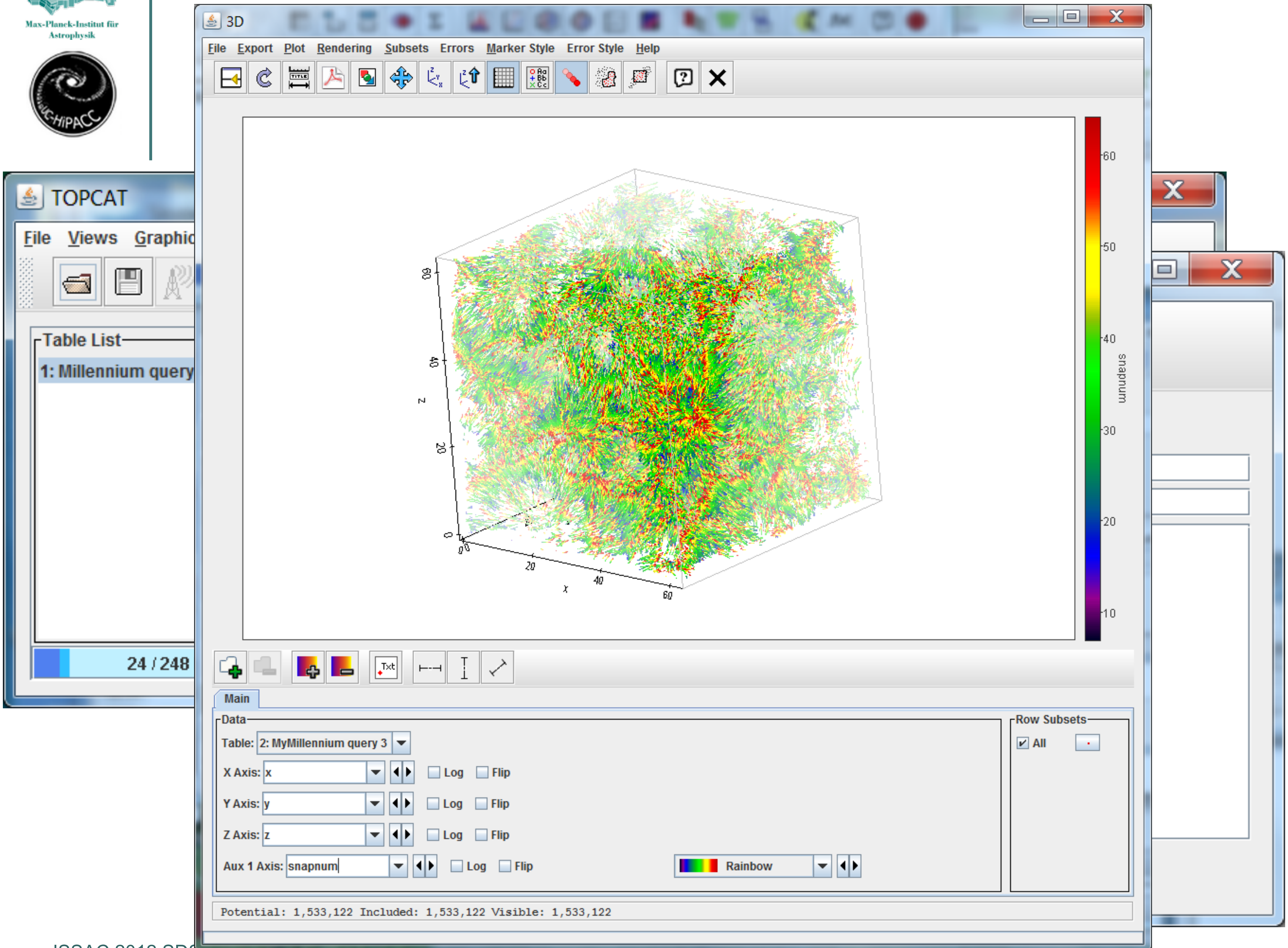
snapnum, x

Databases we will consider

- Databases @ MPA
  - SQLServer:
    <database>.[<schema>].<table>
- Millimil @ ISSACTAP
  - Postgres:
    <schema>.<table>
  - Mirror of millimil+MMSnapshots @MPA
- Documented at
  - http://gavo.mpa-garching.mpg.de/Millennium/Help

Interfaces and Tools

- Millennium Databases @GAVO
  - http://gavo.mpa-garching.mpg.de/Millennium
  - http://gavo.mpa-garching.mpg.de/MyMillennium (auth, MyDB)
  - Wget, R, IDL
- Millimil++ @SDSC
  - http://ion-21-11.sdsc.edu/issactap/ (auth)
  - TAP interface (M. Egger @MPA)
  - psql (hands on sessions)
- TOPCAT
  - Millennium query interface
  - TAP client interface
  - Visualisation via SAMP


3D
File Export Plot Rendering Subsets Errors Marker Style Error Style Help
TOPCAT
File Views Graphic
Table List
1: Millennium query
24 / 248
snapnum
Main
Data
Table: 2: MyMillennium query 3
X Axis: x Log Flip
Y Axis: y Log Flip
Z Axis: z Log Flip
Aux 1 Axis: snapnum Log Flip
Rainbow
Row Subsets
All
Potential: 1,533,122 Included: 1,533,122 Visible: 1,533,122

QUERYING THE DATABASE: SQL

SQL

- *Sequentiual Query Language*
- Filtering, combining, sub-setting of tables
- Functions, procedures, aggregations
- Data manipulation:
  insert/update/delete
- A query produces tabular results, which can be used as tables again in sub-queries, or stored in a database
- Table creation...

Table creation statement

```
create table MPAHalo (
  haloId bigint not null,
  descendantId bigint , -- foreign key
  lastProgenitorId bigint , -- foreign key
  snapnum integer, redshift real,
  x real,y real,z real,
  np integer, velDisp real, vmax real,
  ...,
  primary key (haloId)
);
```

SELECT ... FROM ... WHERE ...

1.

```
select *
  from millimil.snapshots
```

2.

```
select snapnum, redshift, np
  from millimil.MPAHalo
```

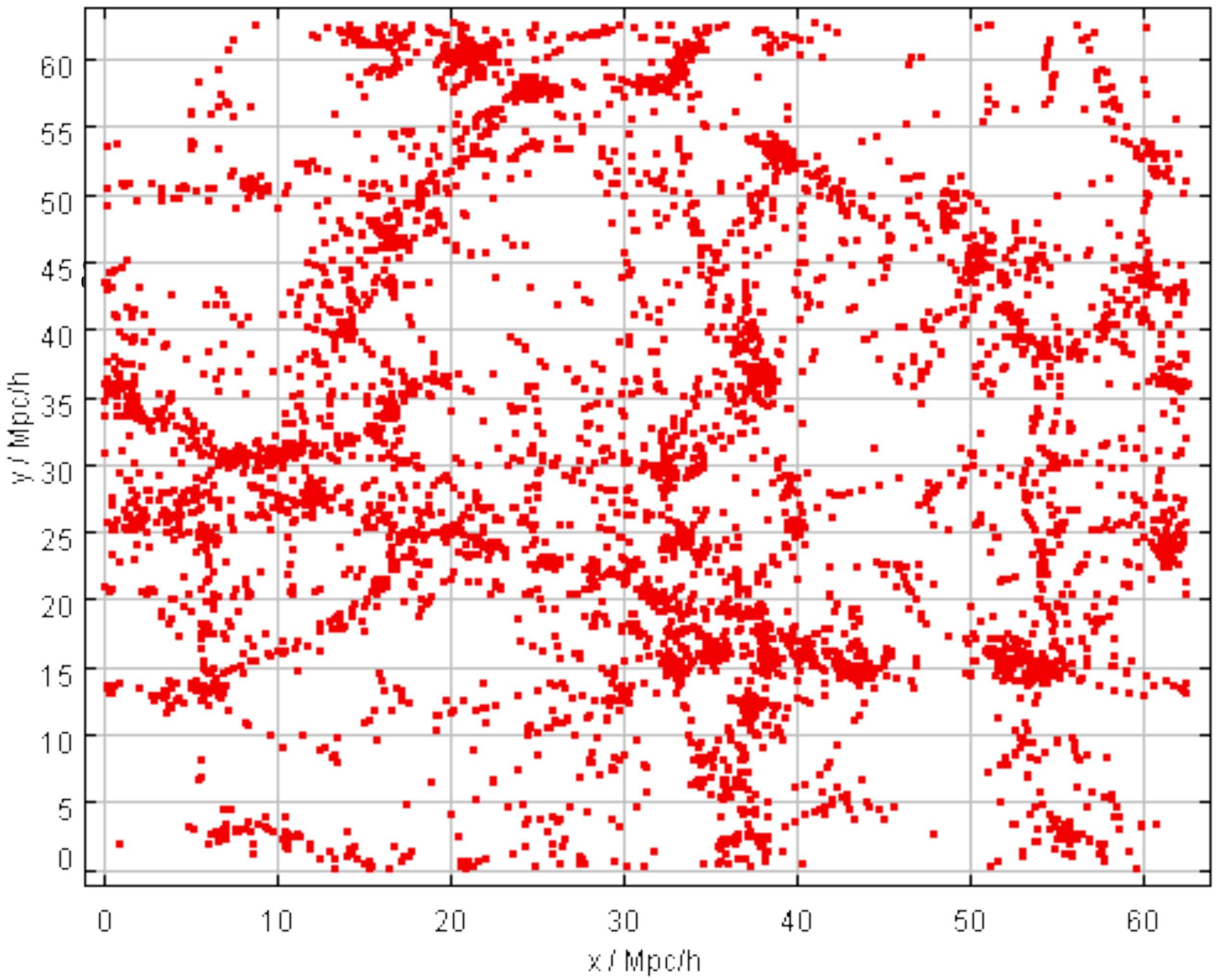
3.

```
select *
  from millimil.MPAHalo
 where redshift = 0
```

WHERE conditions

- `=   <>   !=   <   >   <=   >=`
- `np between 100 and 200`
- `name like '%Primack'`
- `a=b and d=e`
- `a=b or e=d`
- `id in (1,2,3)`
- `a is null`
- `a is not null`
- `exists ... (later)`

Find galaxies in a slice in X,Y,Z at redshift 0



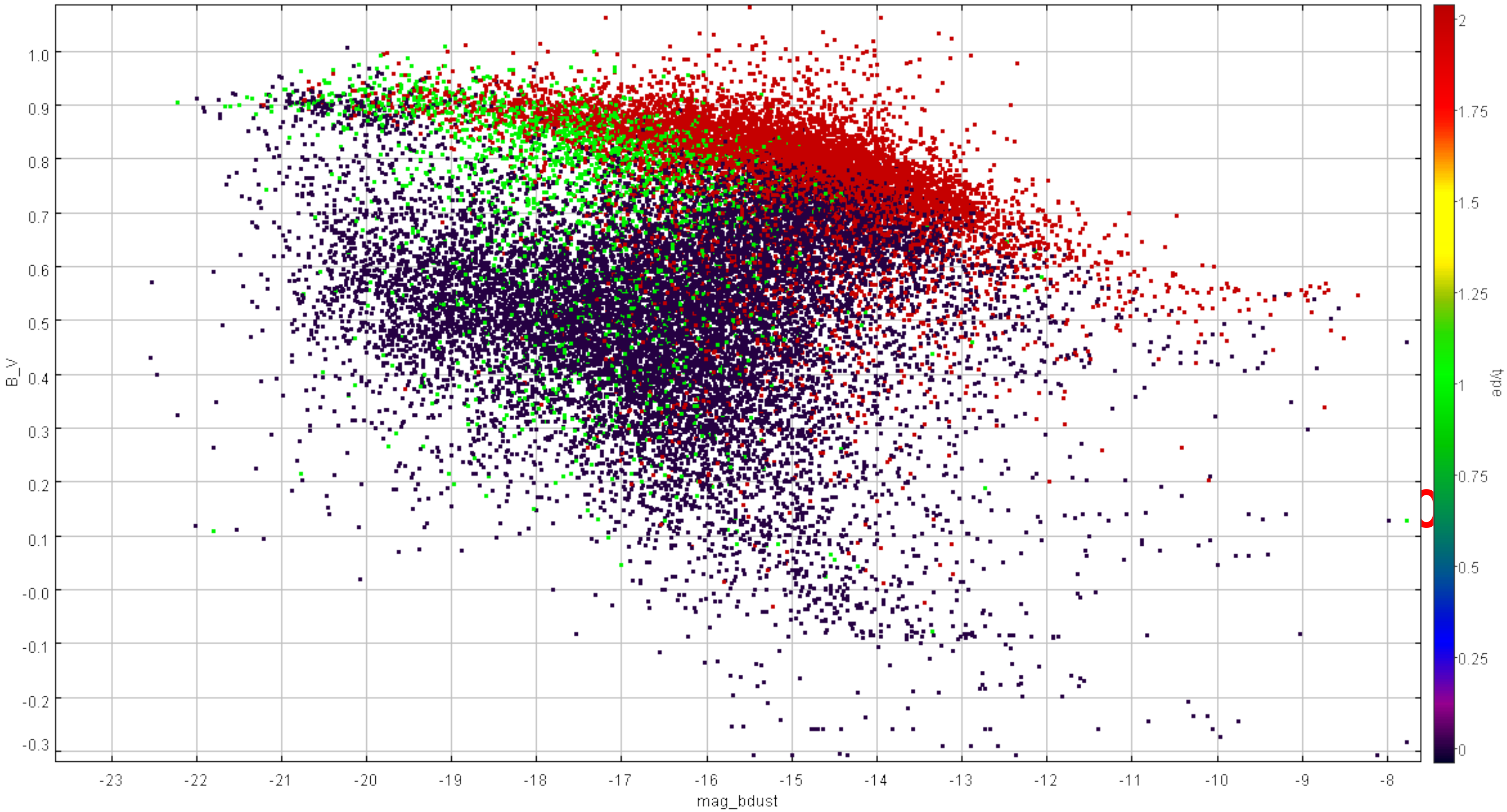
Max-Planck-Institut für Astrophysik

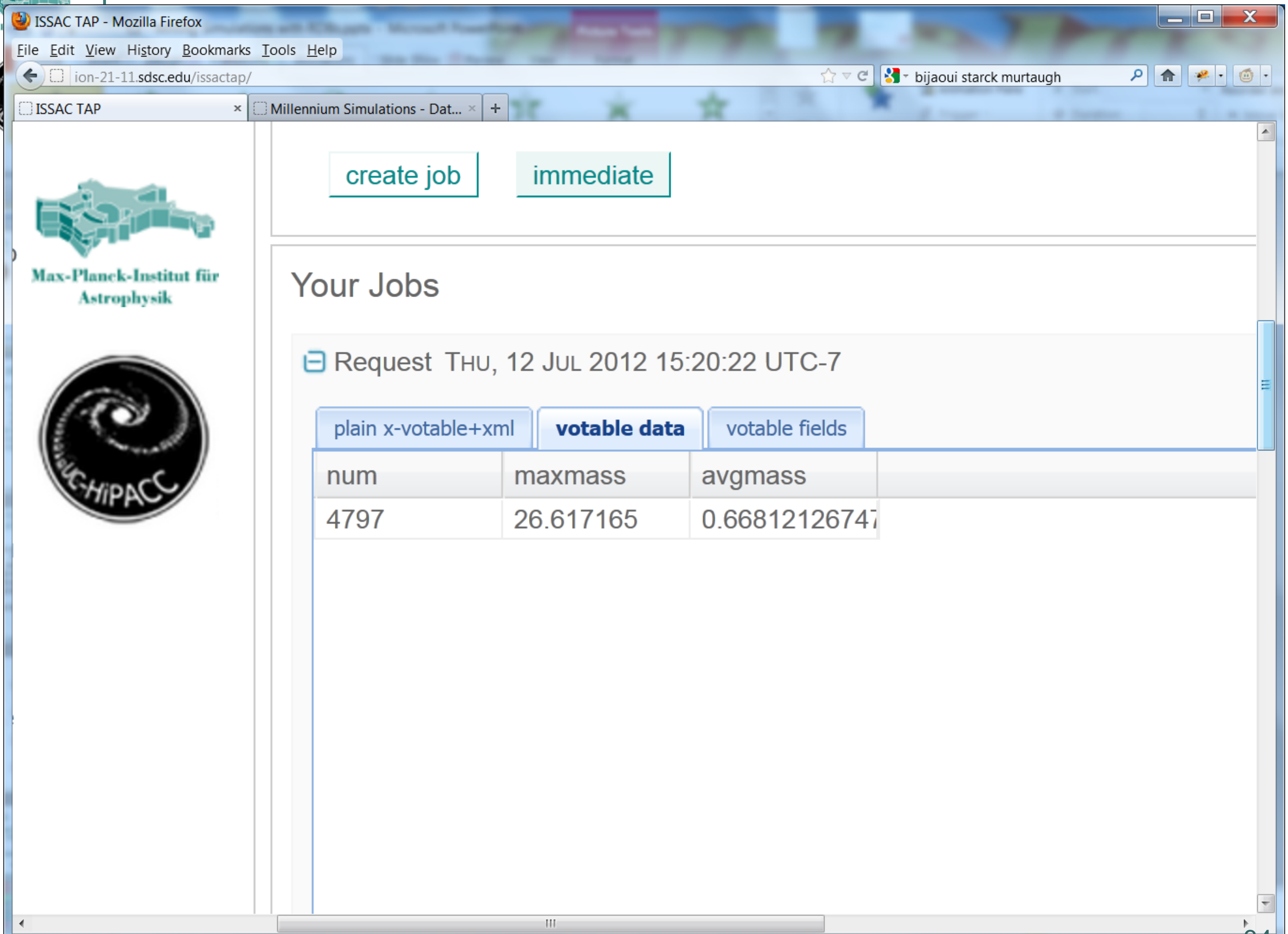
gavo.mpa-garching.mpg

Millennium Simulations - Datab...

snapshotIndex	z	numberOfParticles
63	0.0	51984
62	0.019932542	51288
61	0.041403063	51052
60	0.064493395	51169
59	0.08928783	50870
58	0.11588337	50468
57	0.14438343	50168
56	0.17489761	50485
55	0.20754863	49888
54	0.24246909	48275

Color-magnitude for random sample of galaxies

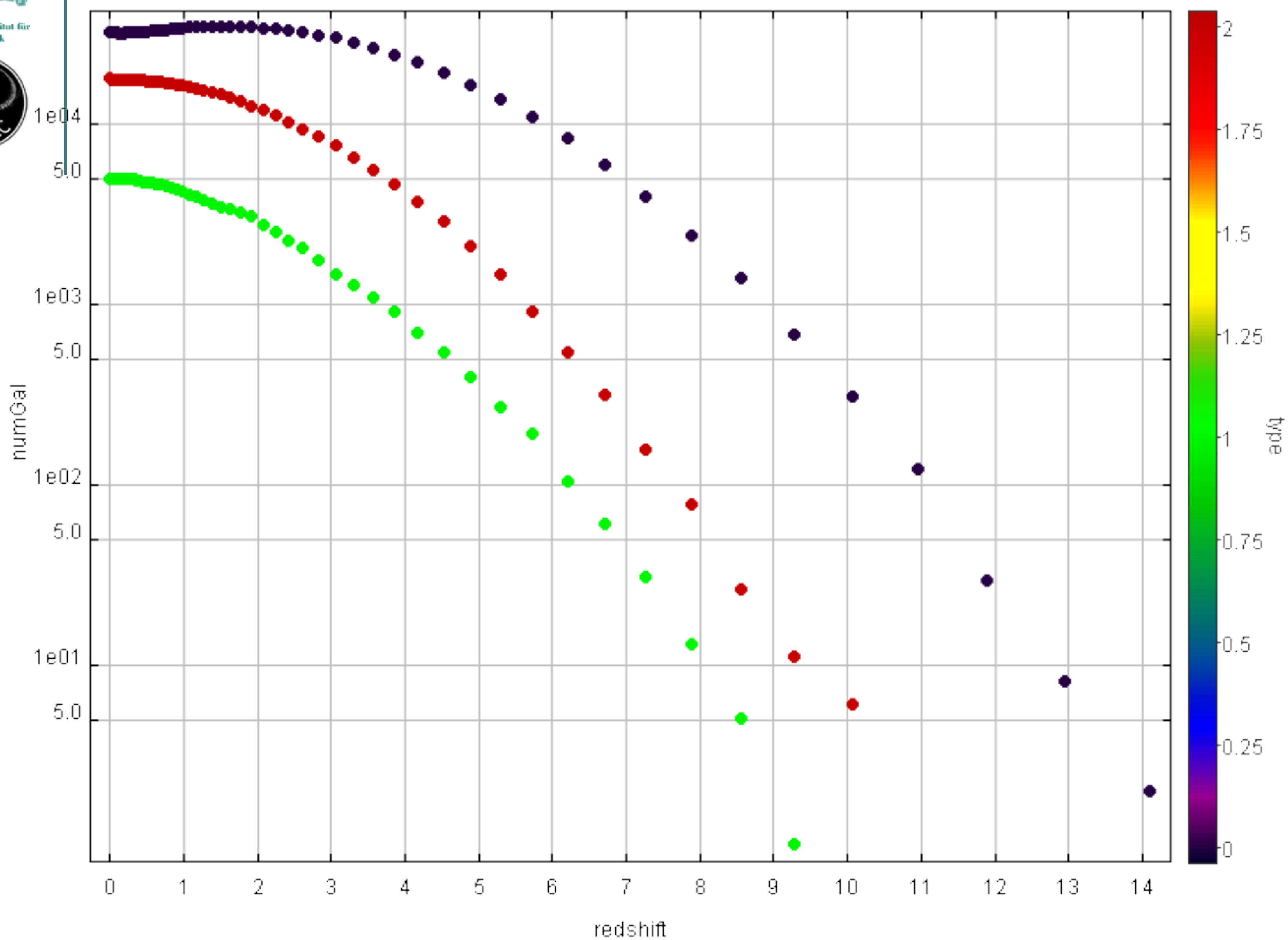



ISSAC TAP - Mozilla Firefox
File Edit View History Bookmarks Tools Help
ion-21-11.sdsc.edu/issactap/
bijaoui starck murtaugh
ISSAC TAP
Millennium Simulations - Dat...
create job
immediate
Max-Planck-Institut für Astrophysik
UC-HiPACC
Your Jobs
Request THU, 12 JUL 2012 15:20:22 UTC-7
plain x-votable+xml
votable data
votable fields
num
maxmass
avgmass
4797
26.617165
0.66812126747

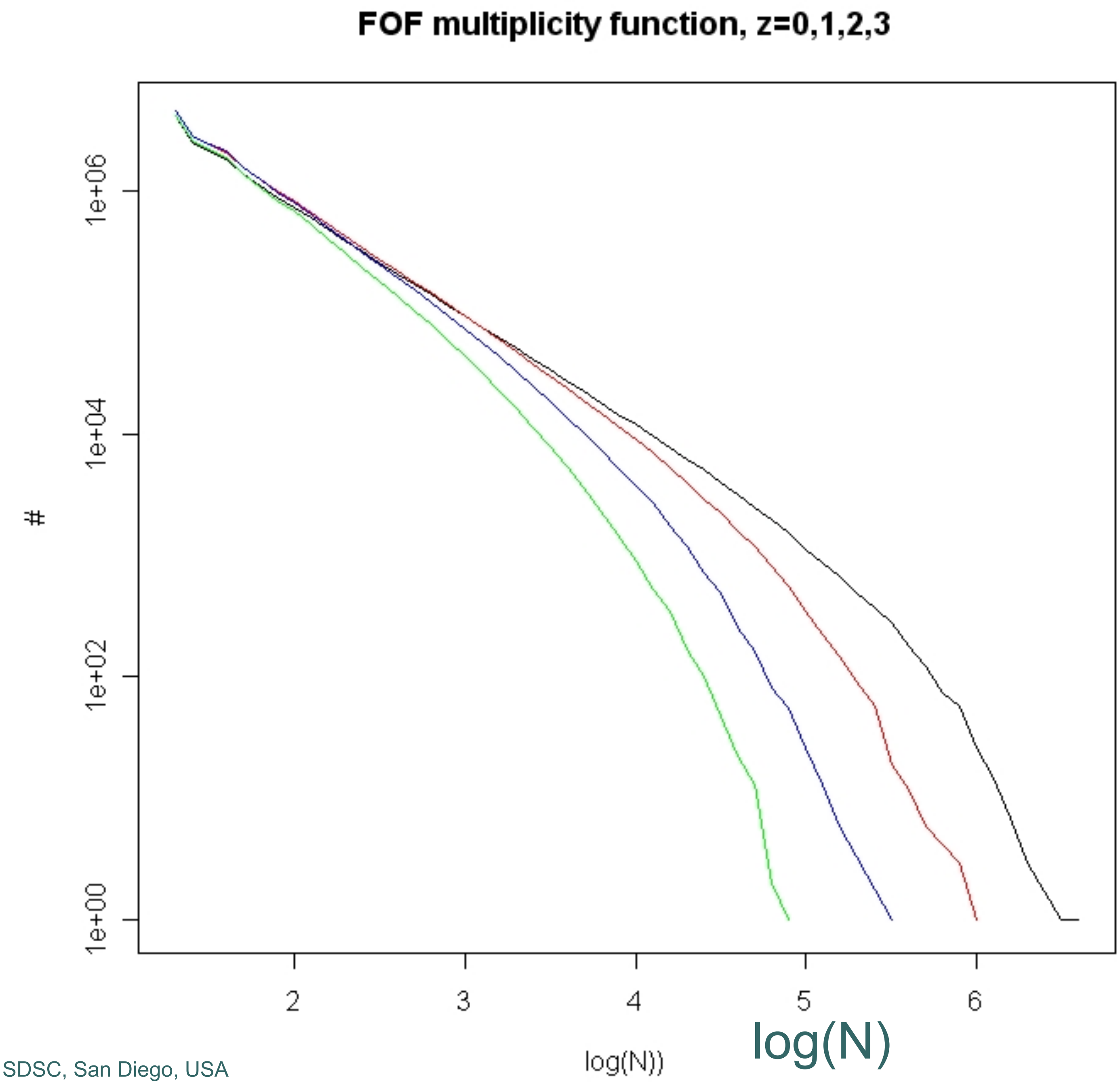
Visited | eclipselink | VisAD | Archimedeans | Intro Programming Java | Comp Geom Software | AWOB MPDL | Mathtools.net | john hawks weblog | Grotte Chauvet

Search

old	descendantId	fofId	snapnum	redshift	np	m_tophat	m_mean200	m_crit200	phkey	x	y	z	velx
00029000000	-1	700002900000	63	0.0	26	1.9795104	2.3237731	1.7213135	32766	0.0012305146	3.483746	61.29536	1.324514
01786000000	-1	100178600000	63	0.0	33	2.1516418	2.6680357	1.1188537	4616	0.0035367154	47.63764	5.312312	-236.44768
01273000000	-1	100127300000	63	0.0	49	3.5286925	3.8729553	3.098364	4104	0.0046070972	36.888336	1.847308	-254.52444
00989000000	-1	100098900000	63	0.0	65	5.594269	6.1106625	4.4754148	5702	0.0068171136	36.769497	28.895704	-64.30341
00358000060	-1	400035800000	63	0.0	20	0.0	0.0	0.0	28727	0.0069455206	25.02912	58.356213	85.258865
7000031	-1	857000000	63	0.0	38	0.0	0.0	0.0	720	0.007186817	11.023374	17.708921	-155.59535
10000000	-1	3110000000	63	0.0	22	1.032788	1.2049193	0.774591	459	0.008191648	5.4254813	10.41615	-291.69476
00703000000	-1	100070300000	63	0.0	94	7.1434507	8.864764	6.282794	5108	0.008483179	50.368225	10.806484	-69.25654
00020000000	-1	600002000000	63	0.0	220	18.762316	20.48363	15.750018	27567	0.00933133	60.14285	41.394062	100.517456
7000000	-1	857000000	63	0.0	54	3.5286925	4.3893495	2.7541015	720	0.01080437	11.145534	17.922892	-145.98897
00021000000	-1	700002100000	63	0.0	126	8.9508295	11.188538	5.4221373	32041	0.012605229	8.013448	45.327274	-126.76699
00032000000	-1	600003200000	63	0.0	22	1.3770508	1.6352477	0.9467224	28434	0.012727483	35.69744	52.560226	-134.48593
02392000000	-1	100239200000	63	0.0	24	1.9795104	2.3237731	1.9795104	5768	0.014633186	33.365486	22.720886	-4.099183
02286000000	-1	100228600000	63	0.0	25	1.8934448	1.9795104	1.8073791	5577	0.015561856	52.93865	25.345882	-156.98141
00027000000	-1	700002700000	63	0.0	38	2.7541015	3.18443	1.9795104	32757	0.020053385	2.7265224	56.631912	-18.97897
00028000752	-1	100002800000	63	0.0	311	0.0	0.0	0.0	5766	0.021663345	35.756992	20.998005	-327.93808
00029000000	-1	600002900000	63	0.0	47	3.0122986	3.6147583	2.4098387	28401	0.022480754	40.06018	49.35567	-90.336815
07000000	-1	1707000000	63	0.0	46	3.9590209	4.131152	3.2704954	3365	0.022890914	20.77495	22.91488	-119.60084
00080000150	-1	100008000000	63	0.0	24	0.0	0.0	0.0	5031	0.023319513	58.314163	9.078091	-121.81812
00233000062	-1	100023300006	63	0.0	20	1.4631164	1.7213135	1.2049193	4675	0.024399236	56.24482	3.769712	-579.57776
00030000000	-1	600003000000	63	0.0	32	2.6680357	3.0122986	1.8934448	27566	0.024525575	62.013985	42.849888	50.916317
00881000067	-1	500088100006	63	0.0	140	11.102471	12.909851	9.553289	27567	0.026298916	58.693596	42.7148	-19.845573
00023000000	-1	600002300000	63	0.0	87	5.9385314	6.885254	4.3032837	27198	0.028090464	51.697906	32.287746	-102.52343
00026000000	-1	700002600000	63	0.0	65	5.1639404	6.024597	4.5614805	32072	0.031980734	13.7017765	34.97189	-364.21002
00051002288	-1	[illegible]	63	0.0	122	0.0	0.0	0.0	27580	0.032583866	56.35495	43.049374	-154.7897
00681000000	-1	100068100000	63	0.0	99	7.057385	8.176239	6.1106625	5768	0.0344693	34.51747	22.207846	36.961323
02494000021	-1	500249400002	63	0.0	109	8.692633	10.93034	8.262304	28430	0.03626855	37.306915	50.49341	-96.74846
00028000000	-1	700002800000	63	0.0	32	2.7541015	2.7541015	1.8934448	28761	0.038097702	22.142498	55.691147	-323.30078



1e04
5.0
1e03
5.0
1e02
5.0
1e01
5.0
numGal
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
redshift
2
1.75
1.5
1.25
1
0.75
0.5
0.25
0
type



FOF multiplicity function, z=0,1,2,3
1e+06
1e+04
1e+02
1e+00
#
2
3
4
5
6
log(N))

#

log(N)

JOINs

```
select h.haloid, g.stellarMass
    from millimil.guo2010a g
  ,      millimil.mpahalo h
   where h.np = 1000
     and g.haloid = h.haloid
```

note the aliases

Guo2010a

galId	haloId	mStar	magB	X
112	6625	0.215	-17.9	7.6
113	6625	0.038	-15.6	7.4
154	6626	0.173	-17.1	7.65
221	7883	1.20	-20.7	35.1
223	7883	0.225	-19.7	35.0
225	7883	0.04	-17.5	34.9
278	7884	1.54	-19.4	35.2

MPAHalo

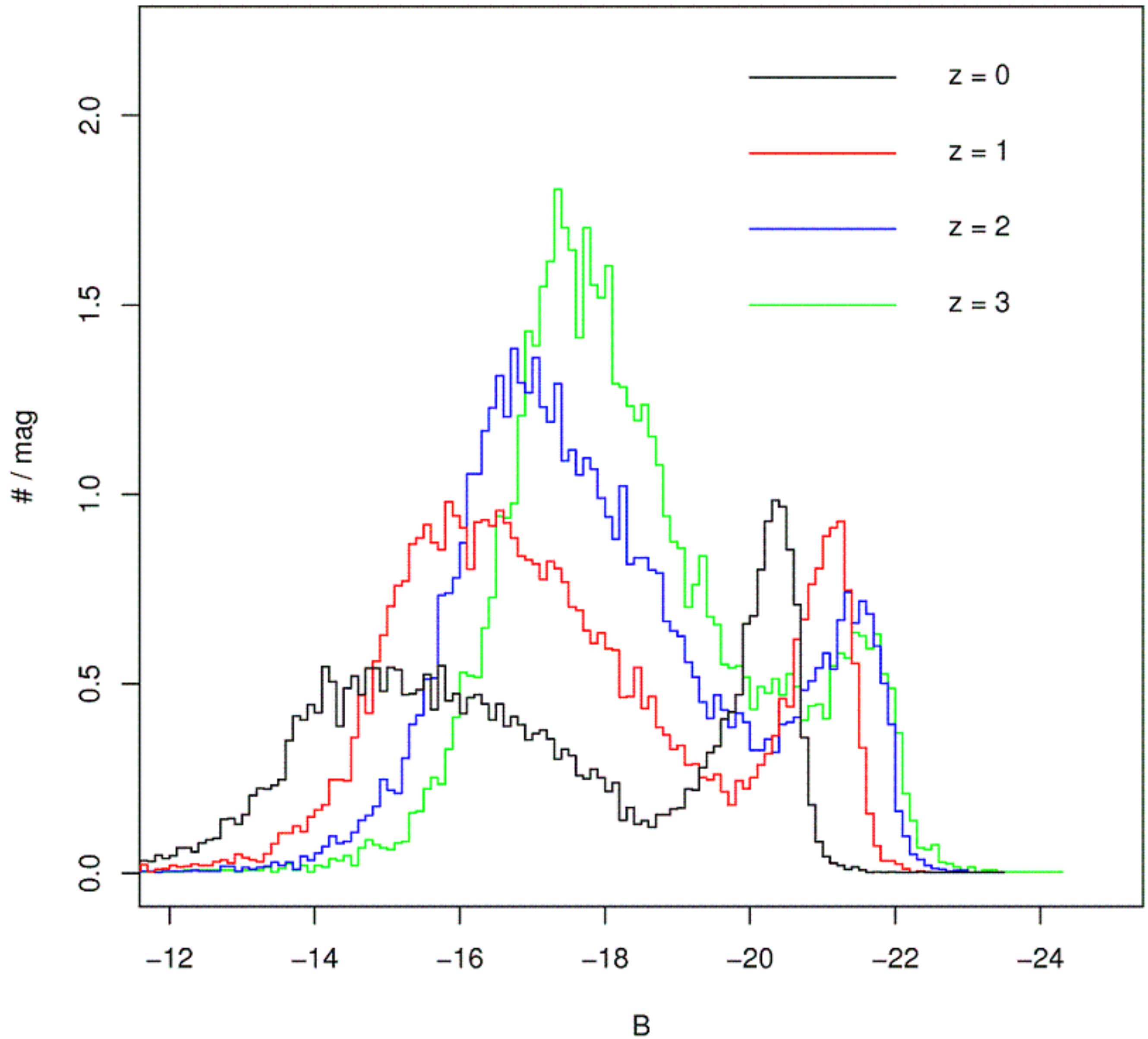
haloId	fofId	Np	X	vMax
6625	123	100	7.6	165
6626	123	65	7.9	130
7883	456	452	35.1	200
7884	456	255	35.2	190
9885	789	30	67.0	110

Galaxies in massive halos

```
select h.haloId
,      g.*
  from millimil.DeLucia2006a g
   , millimil.MPAHalo h
 where h.snapnum = 63
   and h.np between 10000 and 11000
   and g.haloId = h.haloId
```

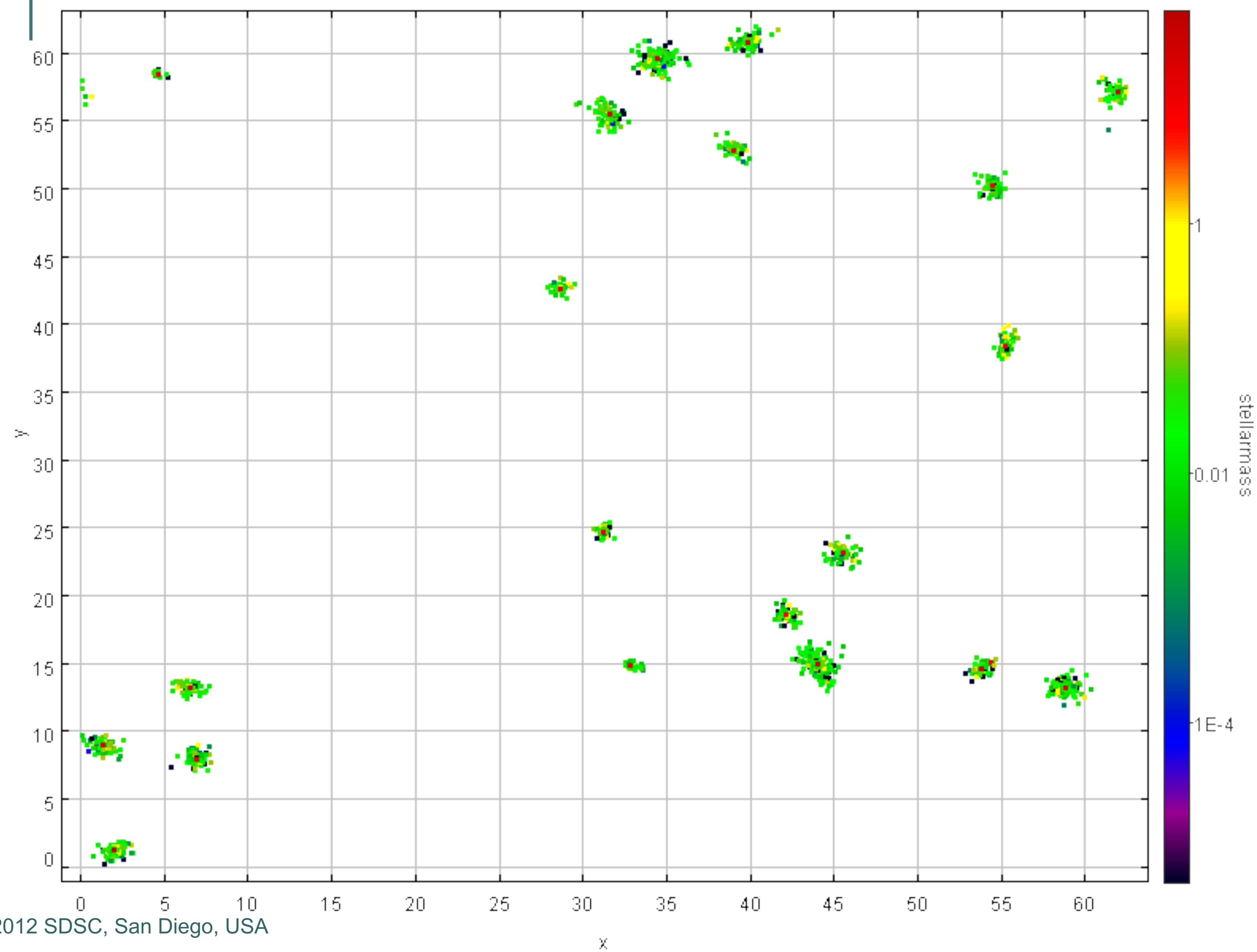
Direct progenitors of massive halos (self-join)

```
select prog.*
   from MPAHalo prog
   ,    MPAHalo des
  where des.haloId = prog.descendantId
    and des.np > 10000
    and des.snapnum = 63
```



z = 0
z = 1
z = 2
z = 3
# / mag
0.0
0.5
1.0
1.5
2.0
−12
−14
−16
−18
−20
−22
−24
B

Sub-select

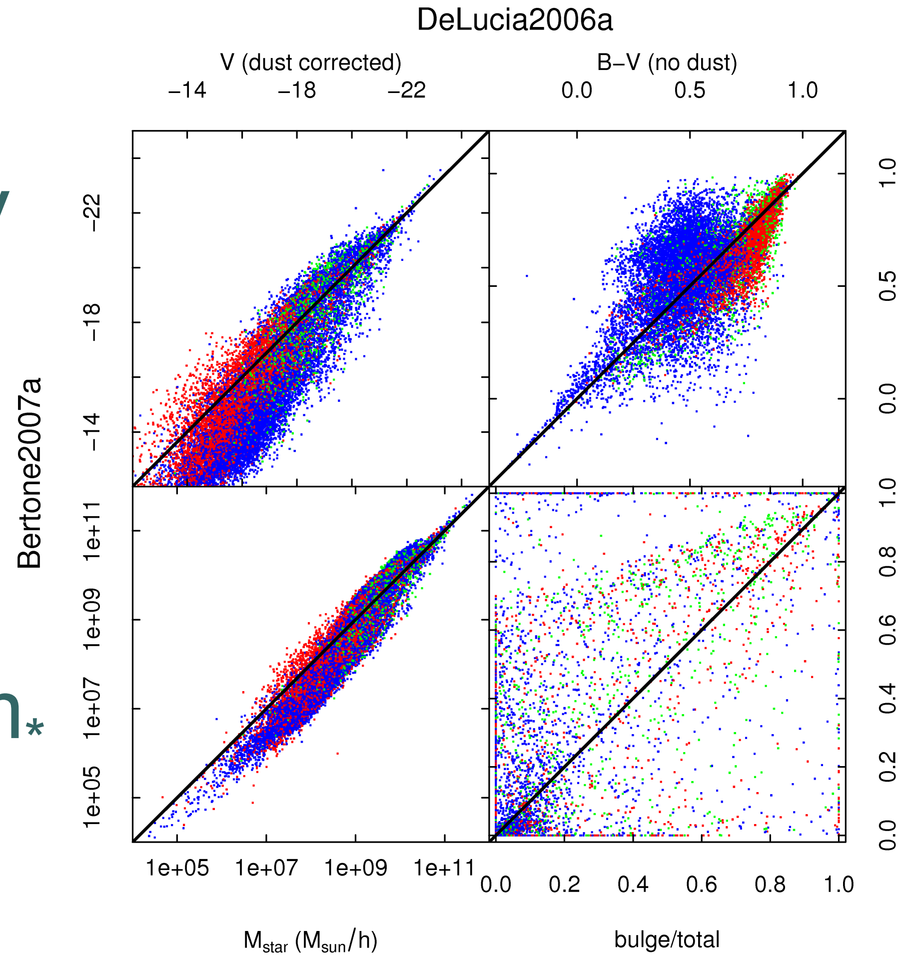


Comparing 2 L-Galaxies models: galaxy by galaxy

DeLucia2006a
vs
Bertone2007a

V

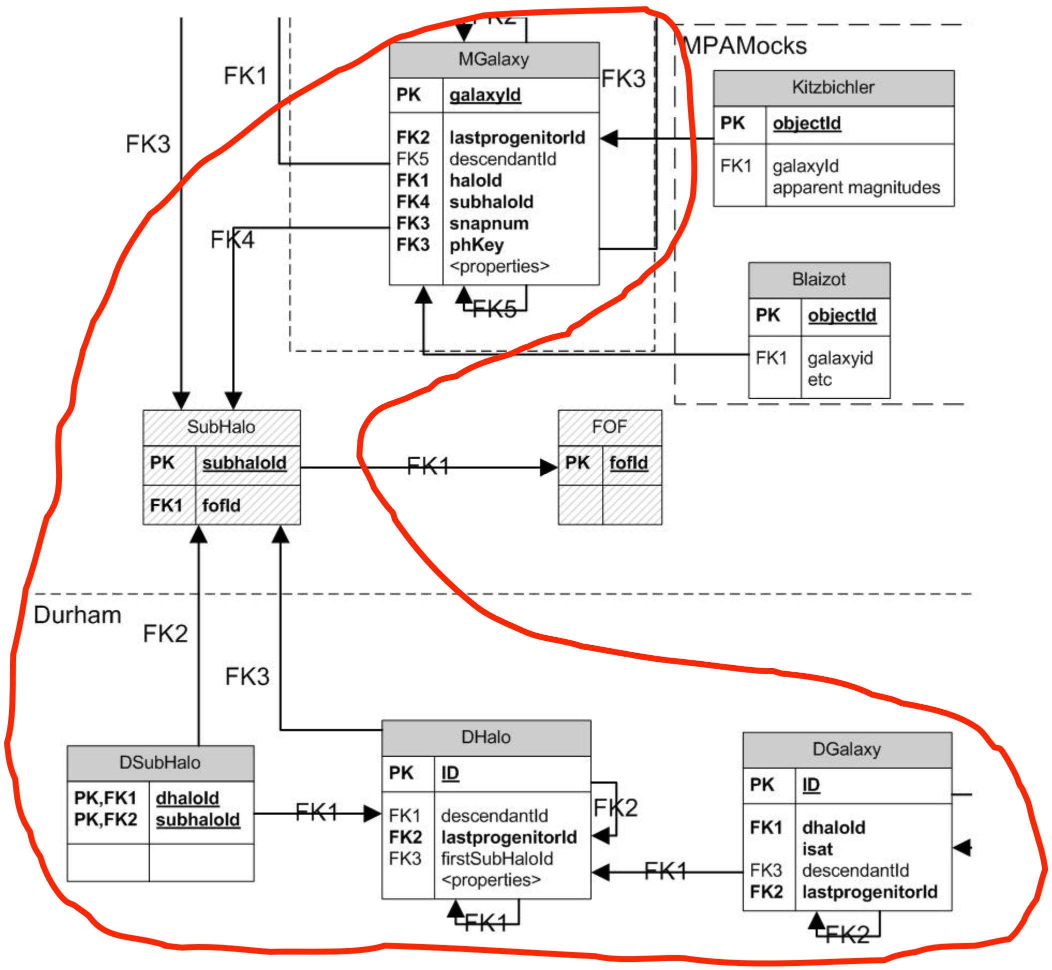
B-V

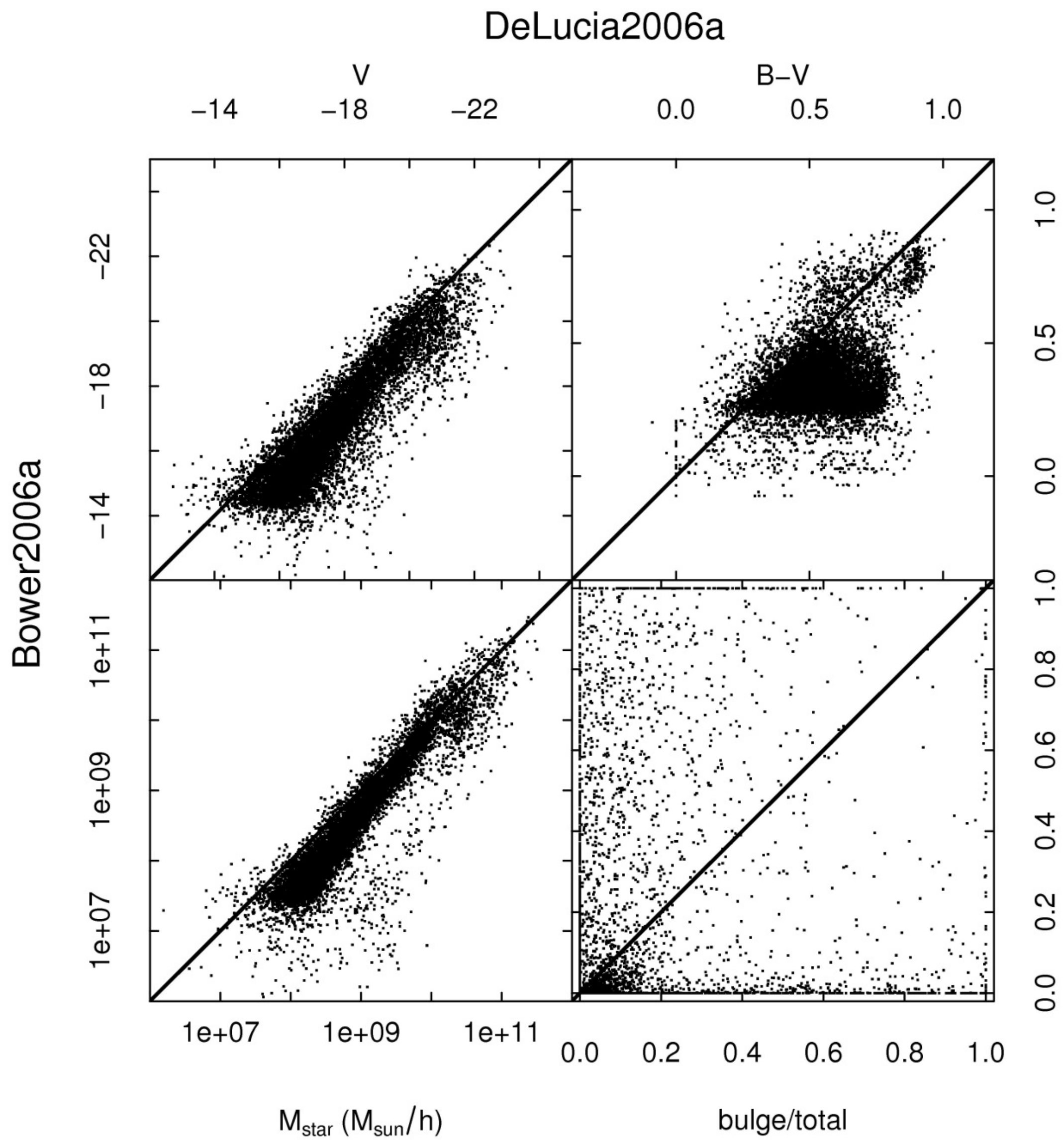


m_*

m_{bulge}/m_*

Compare L-Galaxies (MPA) to GalForm (Durham) models: halo by halo




DeLucia2006a
V
−14
−18
−22
B−V
0.0
0.5
1.0
Bower2006a
1e+07
1e+09
1e+11
0.0
0.2
0.4
0.6
0.8
1.0
M_star (M_sun/h)
bulge/total

Some special design features in the Millennium Databases (next lecture?)

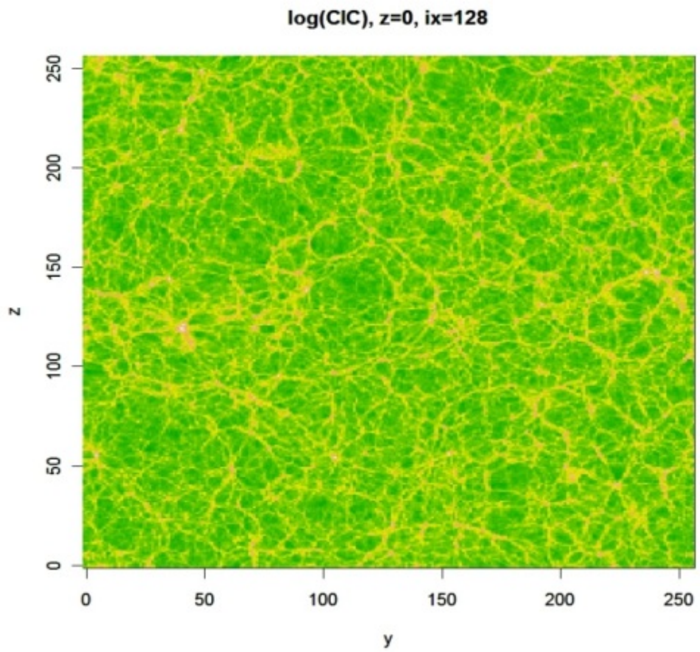
Identifiers
Environment
Trees
(Spatial)

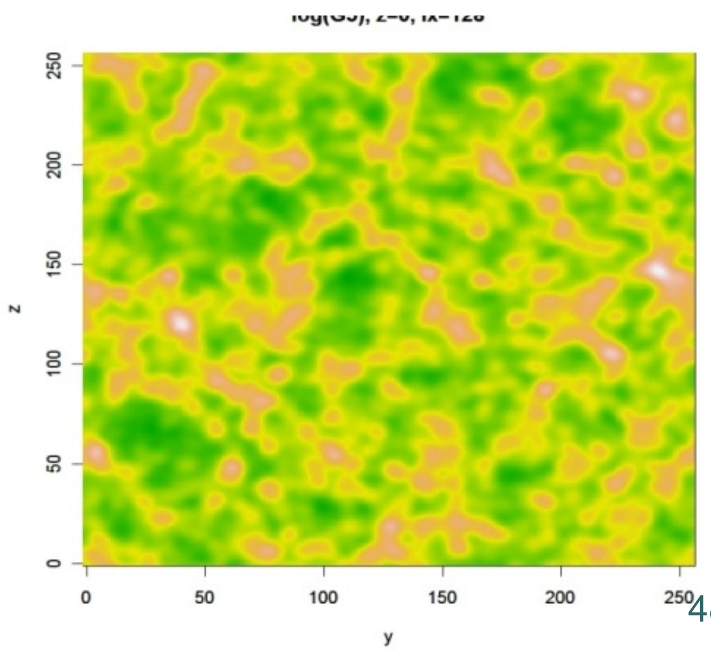
Identifiers

- Parent-child relations reflected in identifiers avoid need for associative tables
  - FOFs in snapnums
    - fofId=snapnum*10^{10}+filenr*10^{6}+rank-in-file
  - Subhalos in FOFs
    - subhaloId = fofId*10^{6}+rank-in-fof
  - Particles in FOFs (mini-Mil-II)
    - particleId = fofId*10^{6}+rank-in-fof
    - global id for tracking of orbits

Environment
“find void galaxies”

- Environment as density field on 256^3 grid
- Smoothed at various scales
  - CIC
  - G_5, G10
- Objects know their grid cell



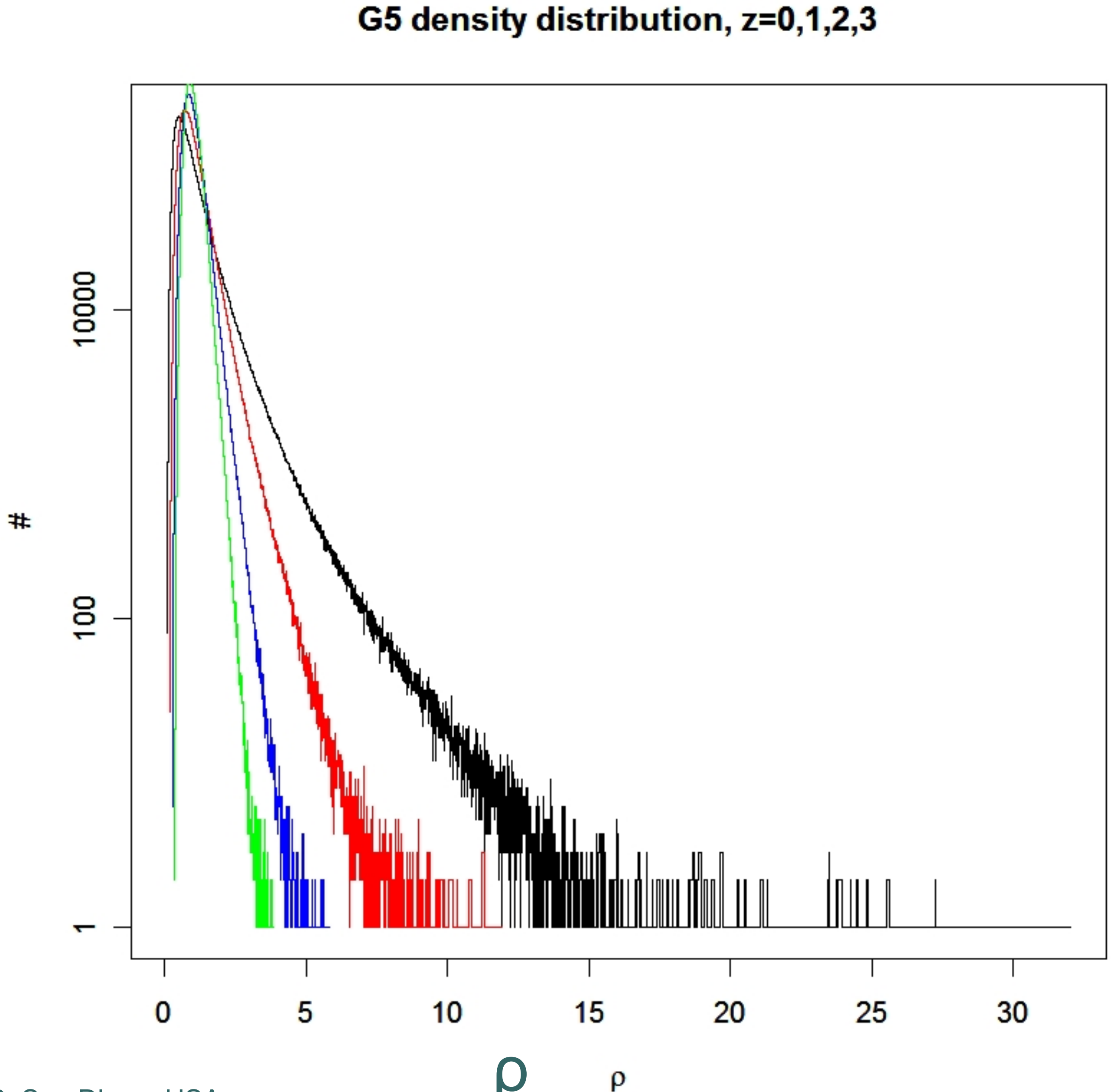


Histogram of density field at redshifts 0,1,2,3; Gaussian smoothing 5 Mpc/h

```
select snapnum
,       .01*floor(f.g5/.01) as g5
,       count(*) as num
  from mfield..mfield f
 where f.snapnum in (63,41,32,27)
 group by snapnum,.01*floor(f.g5/.01)
 order by 1,2
```

G5 density distribution, z=0,1,2,3

#

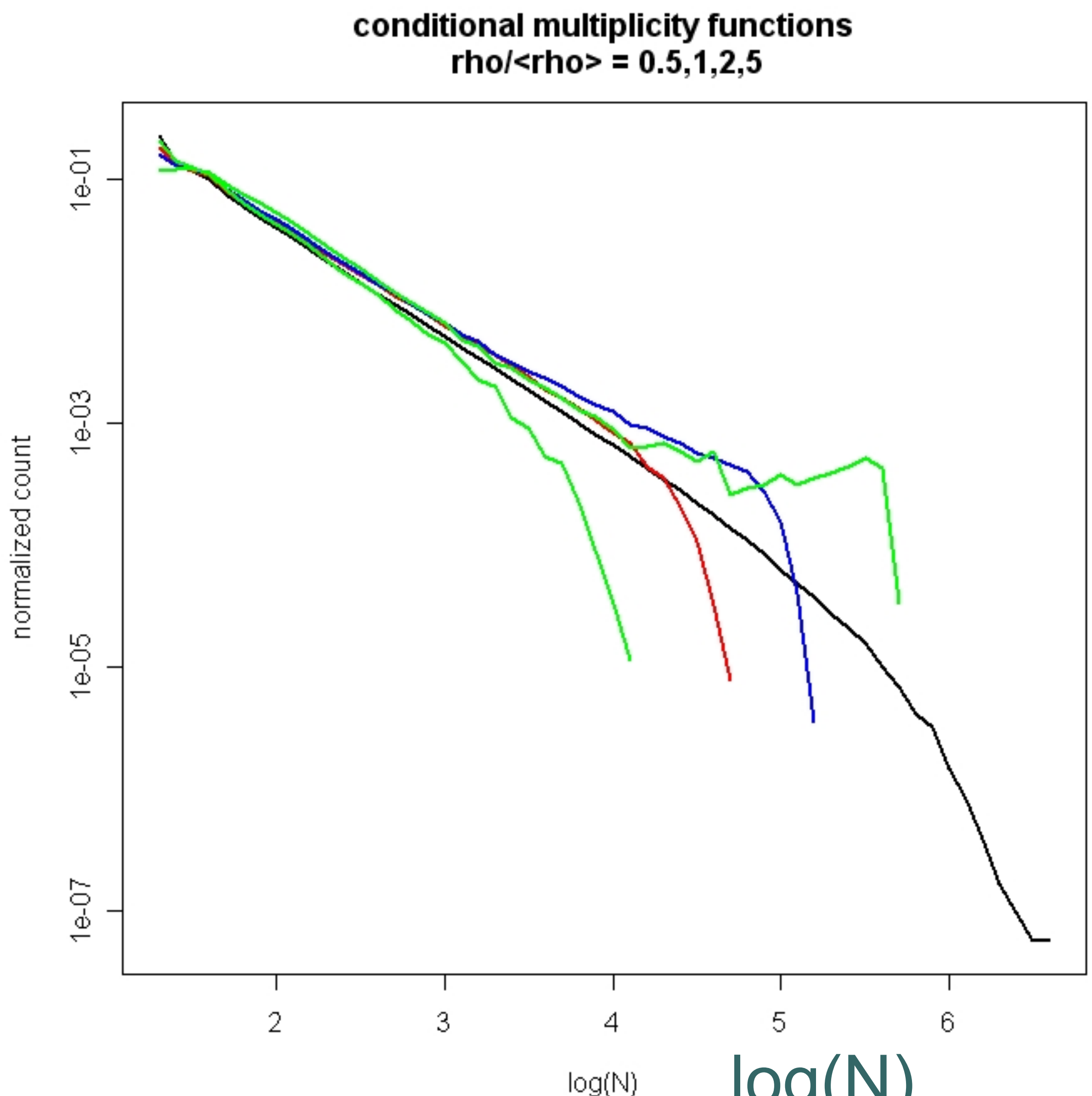


ρ

FOF mass multiplicity function, conditioned on density in environment

```
select .1*floor(log10(fof.np)/.1) as lognp
,       count(*) as num
  from mfield..mfield f
  ,    mfield..fof fof
 where fof.snapnum=f.snapnum
   and fof.phkey = f.phkey
   and f.snapnum = 63
   and f.g5 between 1 and 1.1
group by .1*floor(log10(fof.np)/.1)
order by 1
```

(and similar for g5 = 0.5,2,5)

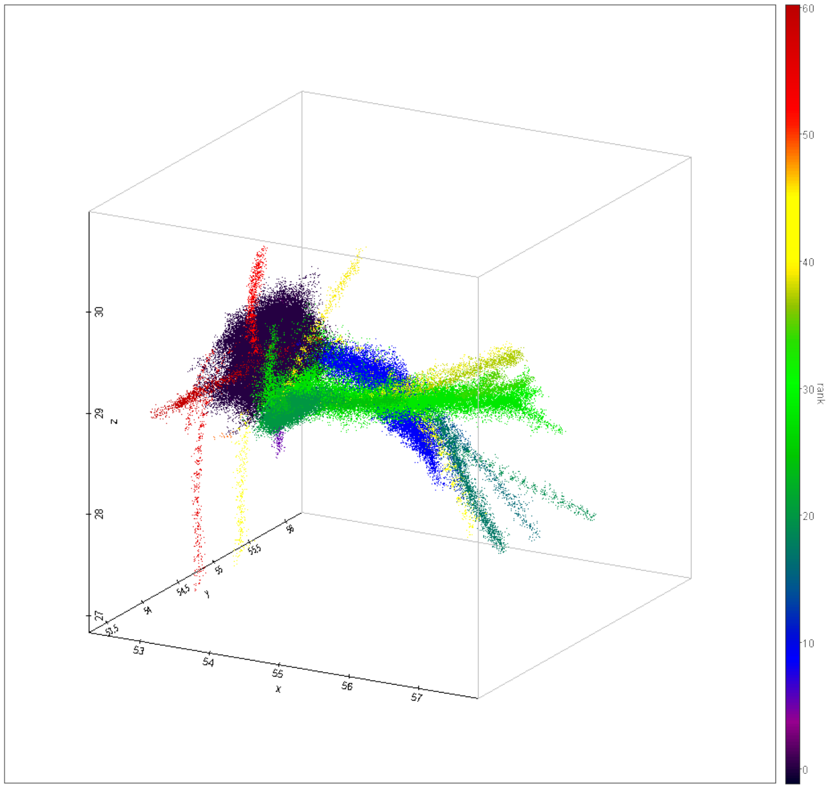


conditional multiplicity functions
rho/<rho> = 0.5,1,2,5
normalized count
1e-01
1e-03
1e-05
1e-07
2
3
4
5
6
log(N)

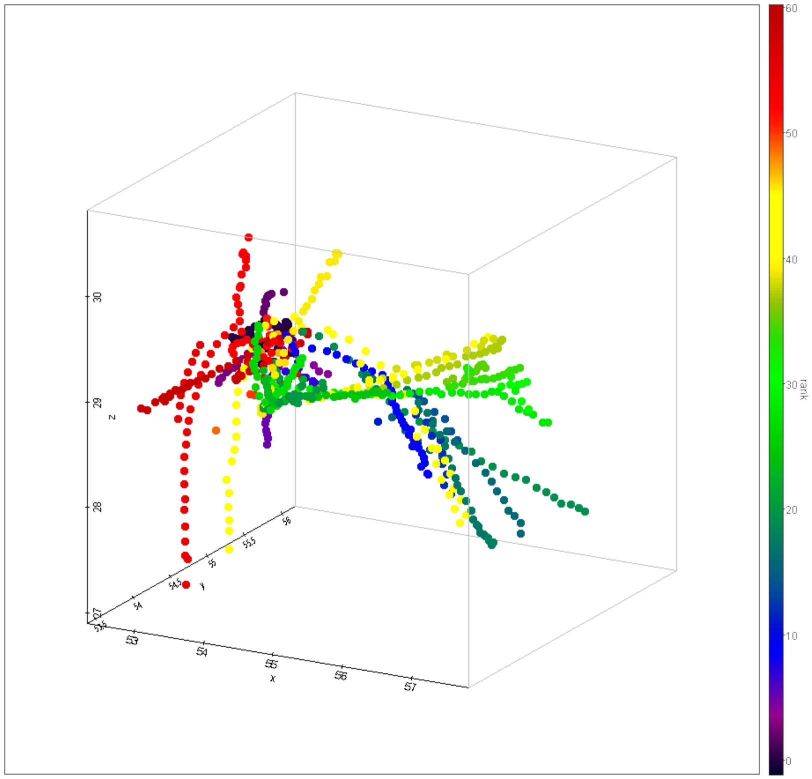
#

log(N)

Time evolution on merger trees



particles



halos

Galaxies

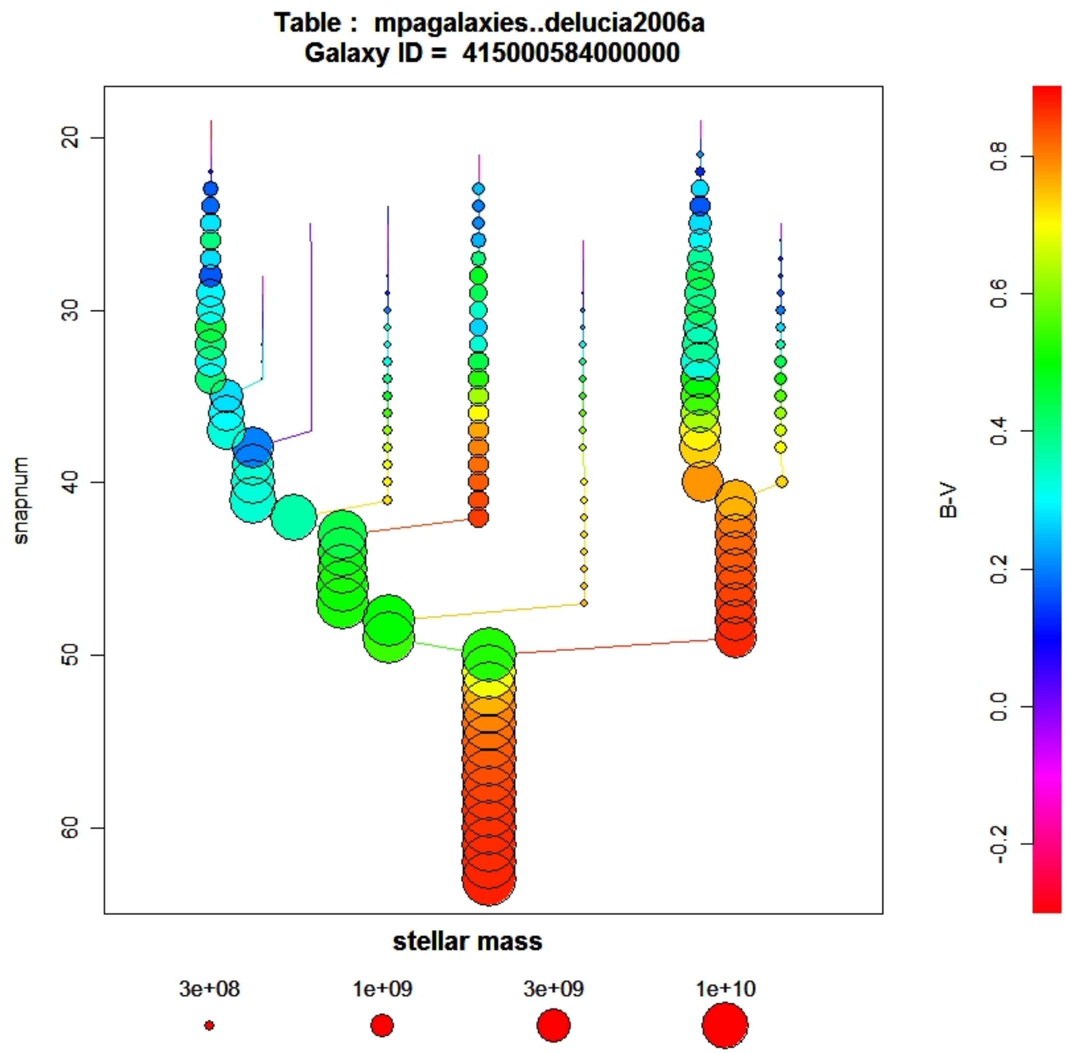


Table : mpagalaxies..delucia2006a
Galaxy ID = 415000584000000
snapnum
20
30
40
50
60
stellar mass
3e+08
1e+09
3e+09
1e+10
B-V
-0.2
0.0
0.2
0.4
0.6
0.8

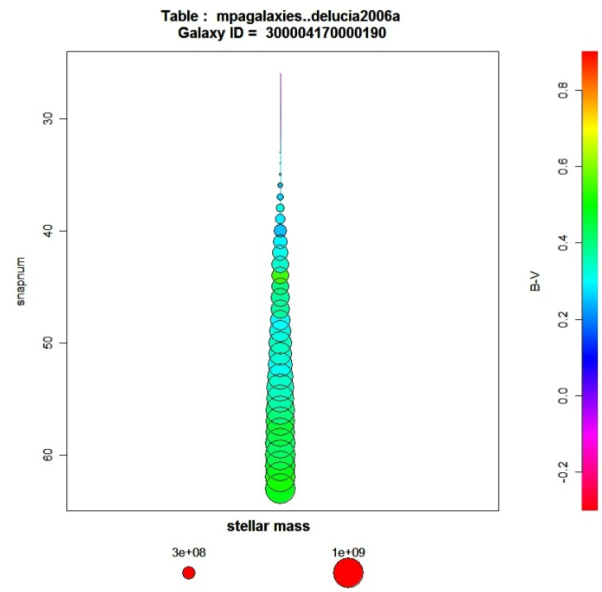


Table : mpagalaxies..delucia2006a
Galaxy ID = 300004170000190
snapnum
30
40
50
60
stellar mass
3e+08
1e+09
B-V
-0.2
0.0
0.2
0.4
0.6
0.8

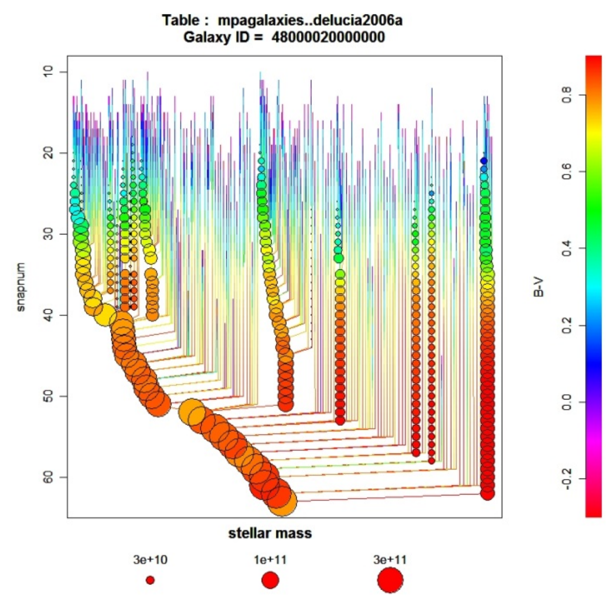
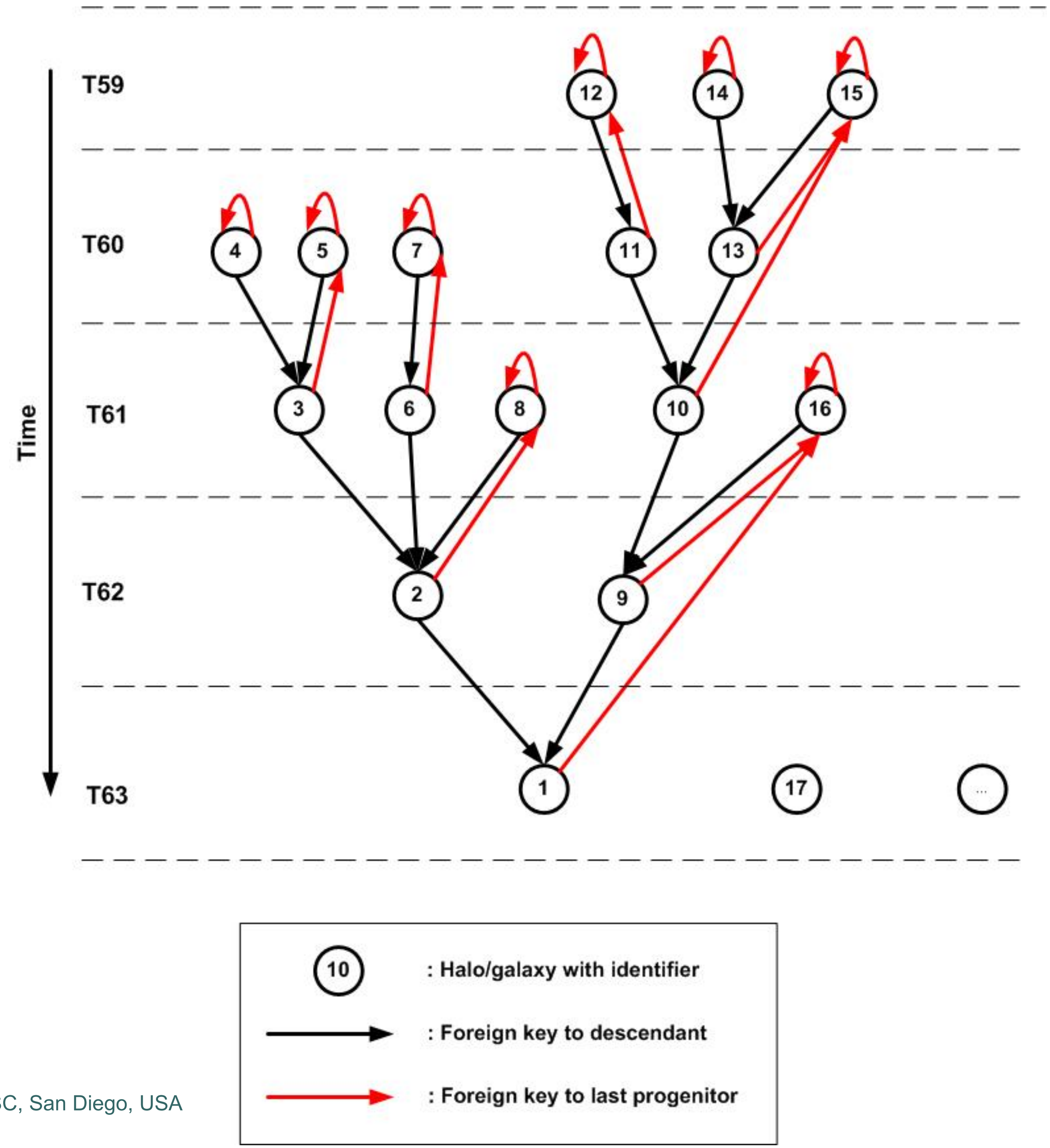


Table : mpagalaxies..delucia2006a
Galaxy ID = 48000020000000
snapnum
10
20
30
40
50
60
stellar mass
3e+10
1e+11
3e+11
B-V
-0.2
0.0
0.2
0.4
0.6
0.8

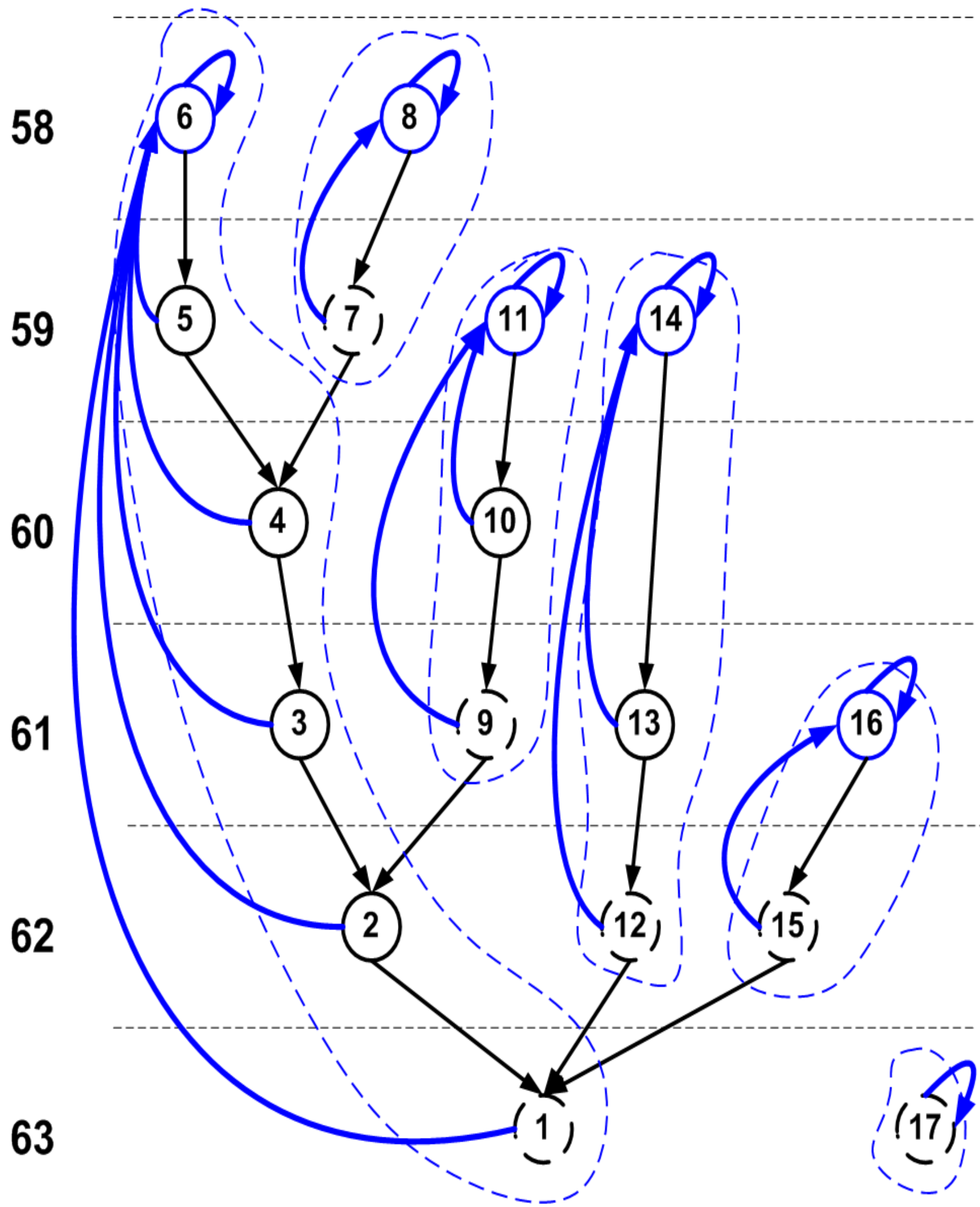
Trees in a database

- Recursion only partially supported
  - And not efficient
- Special solution
  - Indexing based on depth-first-order of progenitors
- Pointers to
  - descendant
  - last progenitor (finding all progenitors)
  - main leaf (finding main progenitors)
    - trees are getting very large (10^8)
    - branches ~100
  - tree root
    - finding descendants. indexing on intervals?


T59
T60
T61
T62
T63
Time
12
14
15
4
5
7
11
13
3
6
8
10
16
2
9
1
17
...
10
: Halo/galaxy with identifier
: Foreign key to descendant
: Foreign key to last progenitor

Main branches

- Track the object
- Pointer to *main leaf*



Merger trees :

```
select prog.*
  from galaxies des
  ,    galaxies prog
 where des.galaxyId = 0
   and prog.galaxyId between
       des.galaxyId and des.lastProgenitorId
```

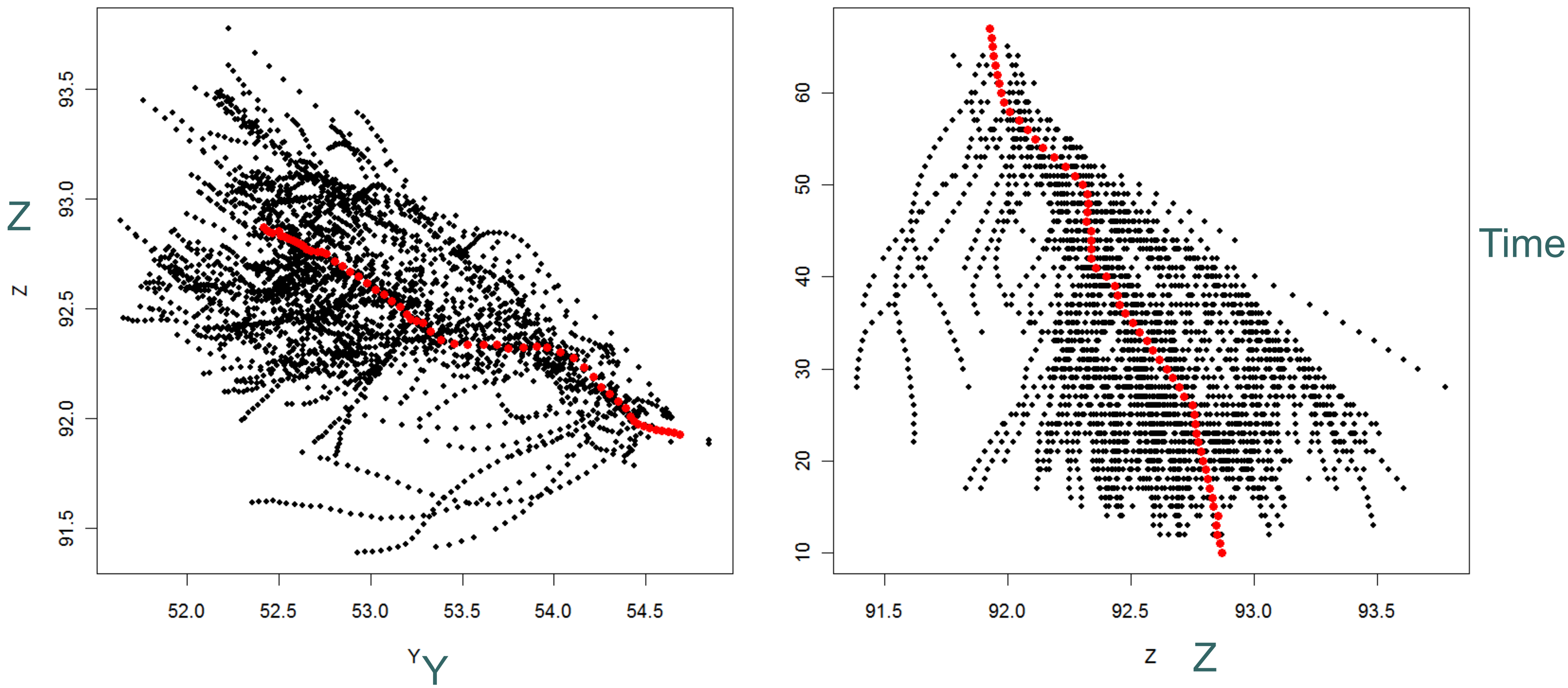
Main progenitors :

```
select prog.*
  from galaxies des
  ,    galaxies prog
 where des.galaxyId = 0
   and prog.galaxyId between
       des.galaxyId and des.mainLeafId
```

Descendants :

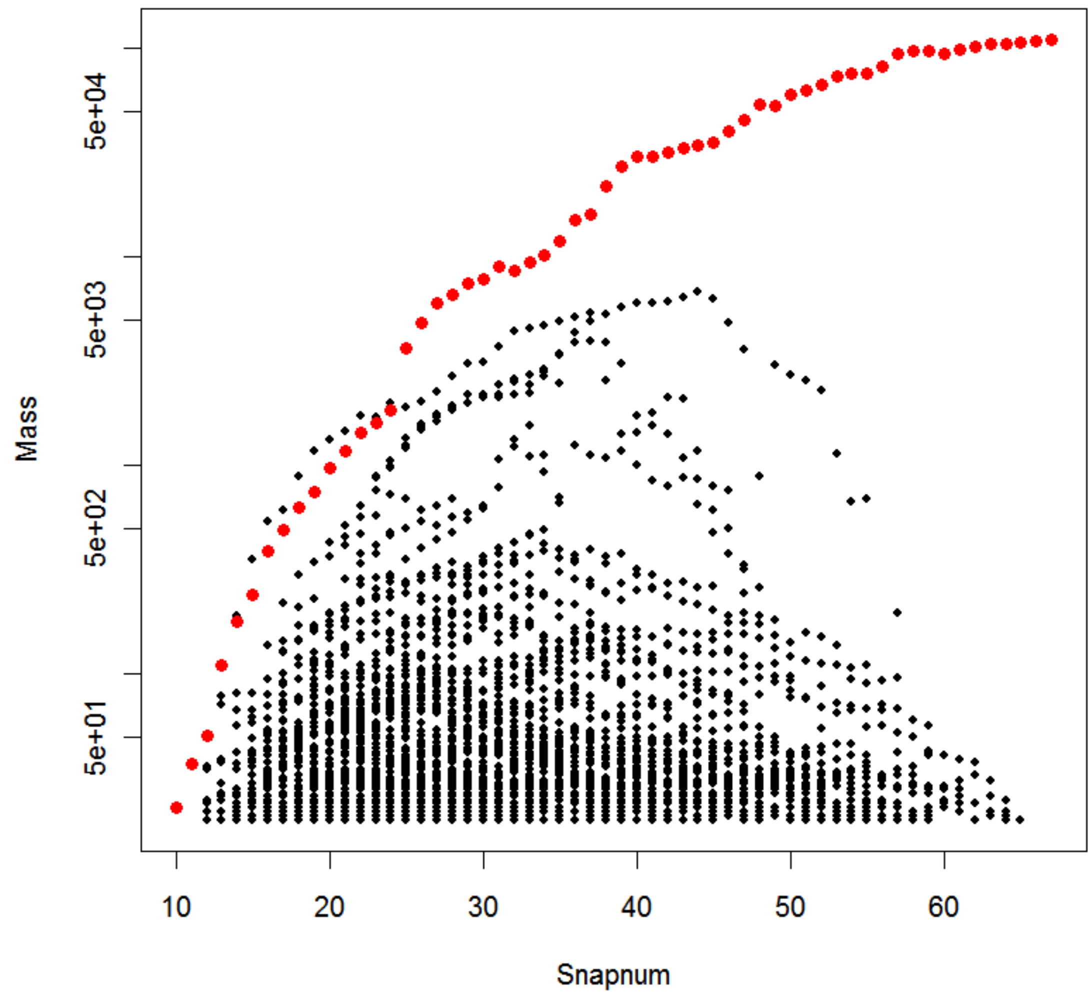
```
select des.*
  from galaxies des
  ,    galaxies prog
 where prog.galaxyId = 41
   and des.galaxyId between prog.treeRootId and prog.galaxyId
   and prog.galaxyId between des.galaxyId and
des.lastProgenitorId
```

Merger tree rooted in particular halo (in Millennium-II database)



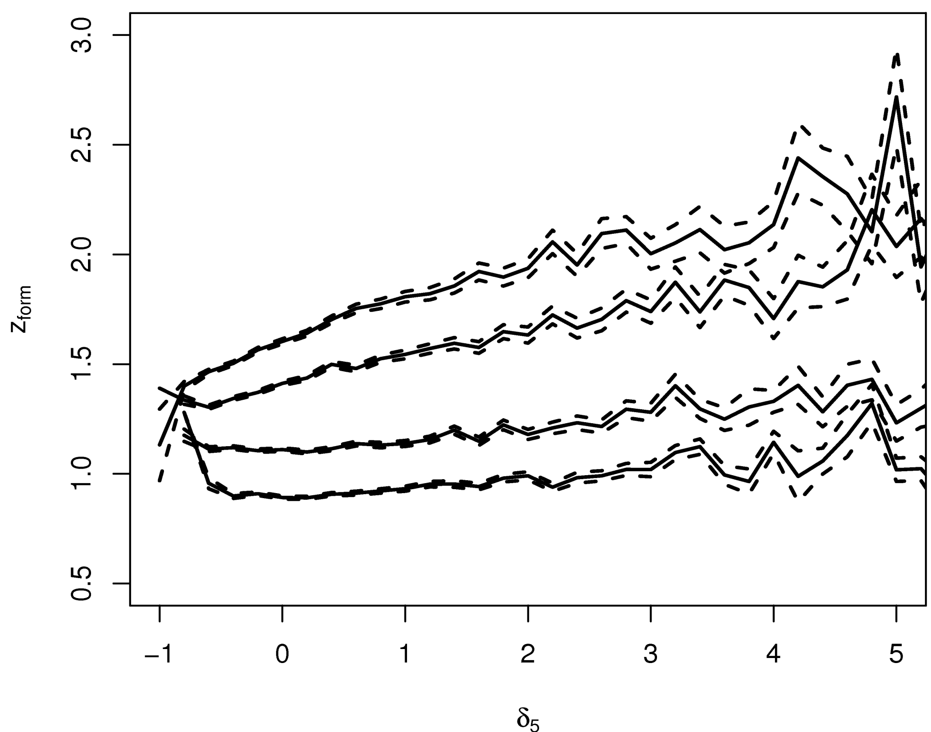
Evolution of mass

Mass



Time

<Z_{form}>



Mass

$\delta 5$

Spatial queries

Find all halos in a subvolume of space:

10 <= x < 20

20 <= y < 30

0 <= z < 10

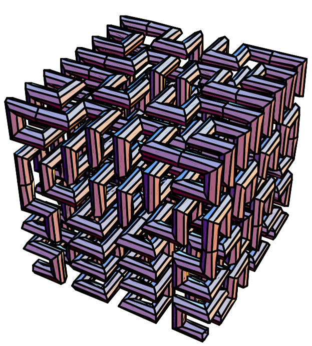
```
select x,y,z
  from millimil.mpahalo
 where snapnum = 63
   and x between 10 and 20
   and y between 20 and 30
   and z between 0 and 10
```

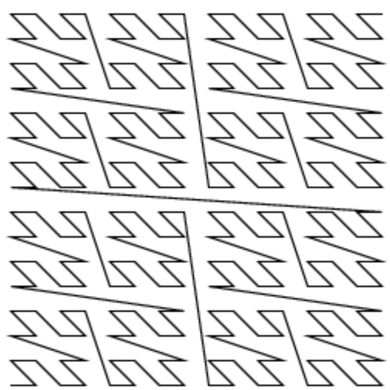
Inefficient, even when indexed !

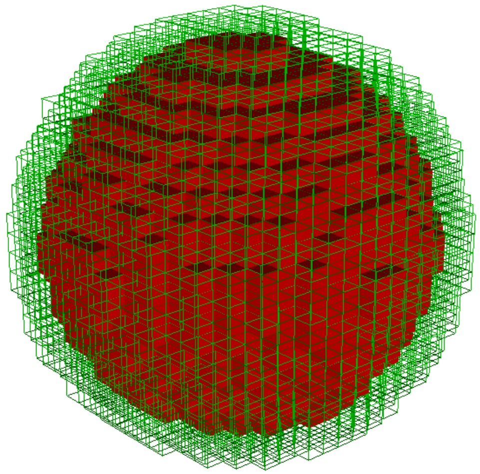
x	y	z
15.001083	42.471325	24.673561
15.001247	58.420914	42.722874
15.002215	38.042484	29.557423
15.002735	50.487785	57.716877
15.002753	20.000177	8.21466
15.005095	13.637599	16.135191
15.006593	22.170828	48.242783
15.011488	24.824438	19.773285
15.011741	48.099907	11.500685
15.011868	23.312265	27.858799
15.013065	23.969515	18.883507
15.013158	56.041866	40.82894
15.014361	59.503357	45.31733
15.017322	46.257664	44.37695
15.018202	27.333895	9.441319

Spatial indexes

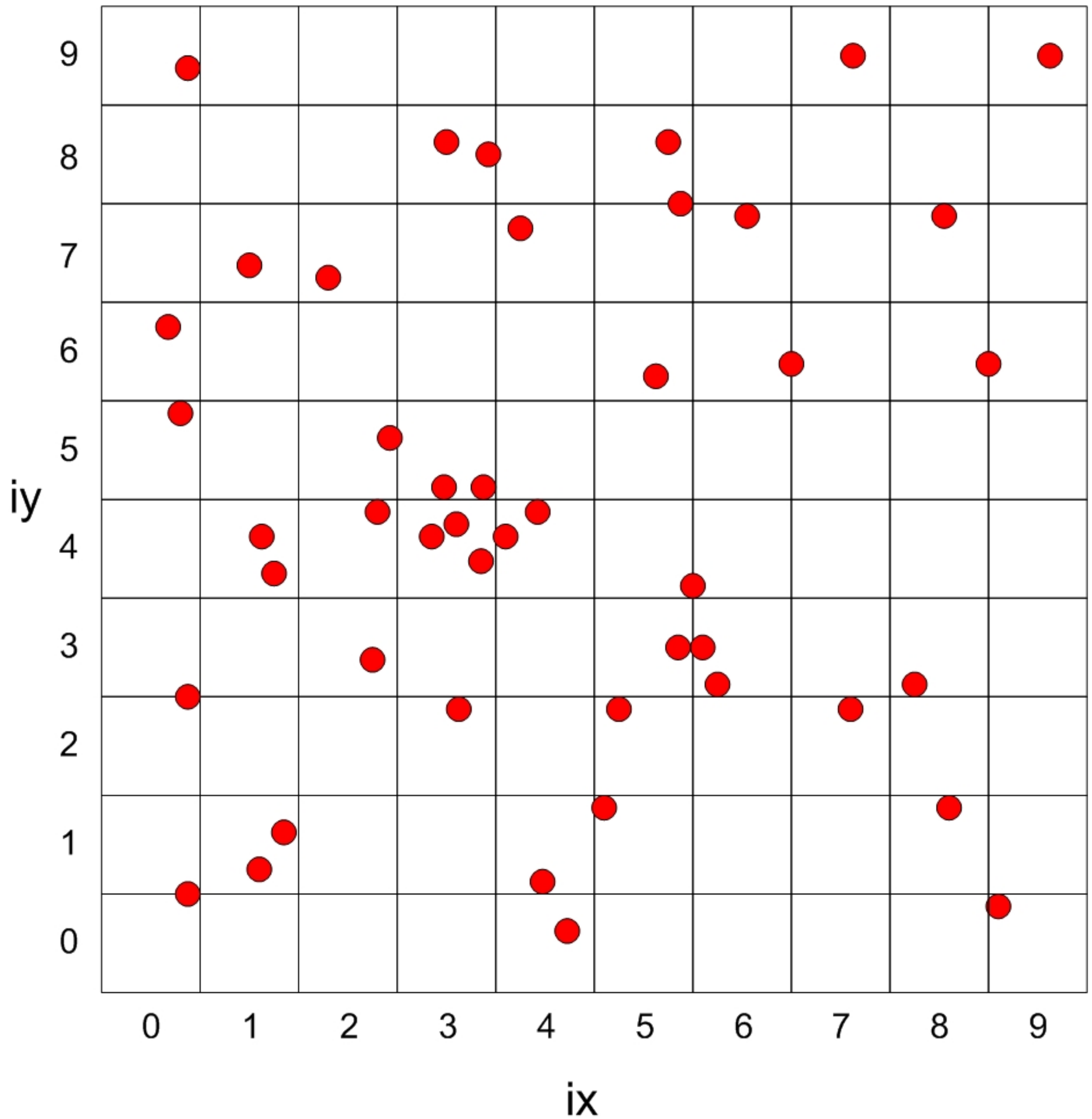
- Performance of finding things is improved if those things are co-located on disk: ordering, indices
- Co-locating a 3D configuration of points on a 1D disk can only be done approximately
- Space filling curves: Peano-Hilbert, Z-curve
- See Tamas' talk







Simpler: Zones



Zone index

- Coarse sampling of points in multiple dimensions allows simple multi-dimensional ordering
- ix = floor(x/10Mpc)
  iy = floor(y/10Mpc)
  iz = floor(z/10Mpc)
- index on (snapnum,ix,iy,iz,x,y,z,galaxyId)

ix=1 and iy=2 and iz=0

IX	IY	IZ	X	Y	Z
1	2	0	15.061804	20.891907	4.4156647
1	2	0	15.069336	23.437601	9.812217
1	2	0	15.100678	20.905642	4.613036
1	2	0	15.173968	22.368838.01832	
1	2	0	15.194122	20.675834.8034463	
1	2	0	15.2500305	24.246683	1.6651521
1	2	0	15.365576	23.290754	9.404872
1	2	0	15.372606	20.203691	2.0006201
1	2	0	15.524696	21.039974.280077	
1	2	0	15.583943	22.344622	9.421347
1	2	0	15.6358385	26.785904	9.881406
1	2	0	15.6638322.829983	7.137772	
1	2	0	15.673803	26.918291	3.302736
1	2	0	15.717824	22.365341	9.221828
1	2	0	15.847992	24.700747	1.389664
1	2	0	15.883896	22.593819	7.277129
1	2	0	15.9104126.531118	2.5693457	
1	2	0	15.916905	27.137867	4.289855
1	2	0	16.047333	28.938115.414605	

Back to Matt's categorization of questions.

- What are the hard questions in our approach?
  - SQL does not support them though data does.
  - Solution: download lots of our data, write your own code.
  - Ask DB managers to add more functions to your DB. E.g. Spatial3D, many more @JHU
- What are impossible questions?
  - Not supported by our data.
  - Solution:
    - 1. create your own data (L-Galaxies online, light-cones online etc.)
    - 2. Find it elsewhere (Next lecture, VO)

Max-Planck-Institut für Astrophysik

UC-HiPACC

THANKS TO THE ORGANIZERS AND THANK YOU.

Acknowledgment:

Thanks to Matthias Egger for building the TAP interface. GL and Matthias Egger are supported by Advanced Grant 246797 GALFORMOD from the European Research Council.

Hands on 1

- Running SQL queries
- Teach TAP interface
- Teach TOPCAT interface