

Simulation databases

(K. Riebe)

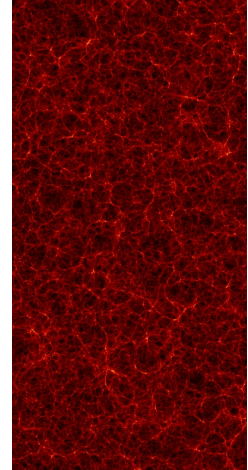
Uses [MultiDark Database](#), [TOPCAT](#) .

Simulation databases like the Millennium Database (<http://gavo.mpa-garching.mpg.de/Millennium/>) or MultiDark (<http://www.multidark.org>) contain data sets from cosmological simulations. Each user has access to catalogues of dark matter halos, their evolutionary history (merger trees) and the relations between them (substructures), supplemented by (mock) galaxy catalogues and information about the density field or tables with the simulation particles themselves and other data sets.

Access to these data is possible via a web interface directly with the browser; for larger data it is more convenient to access the data via [TOPCAT](#), `wget`, `IDL` or `R`.

This tutorial gives a quick overview on the MultiDark Database web interface and then shows an example use case for viewing the cosmic web.

More [demos and tutorials](#) can be found at the MultiDark webpage.



The MultiDark Database - web interface

- Open a **web browser** with <http://www.multidark.org>.
This webpage provides access to the MultiDark Database, where data sets like dark matter halo catalogues and merger trees are stored for two cosmological simulations: Bolshoi and MDR1.
- **Registration:**
 - o For gaining access to the full data sets, it is required to register at <http://www.multidark.org/MultiDark/pages/RegistrationForm.jsp> beforehand. The registration is free and open to everyone.
 - o During tutorials and hands-on sessions a *demo user* account may be enabled for testing purposes. Please ask your tutors for the demo user's name and password.
- **Query Form:**
 - o Navigate to the **Query Form** (<http://www.multidark.org/MultiDark/MyDB>) and login with your username and password (or as the demo user/public user).

- o The available simulations and tables are listed at the left side, a **click on the table name** brings you to the corresponding documentation page, where all the possible columns are explained.

- o SQL queries are written directly in the form, e.g. **write**

```
select top 100 * from miniMDR1..FOF where snapnum=85
order by mass desc
```

and press the **Query (browser)** button. After a few seconds the result will be returned as a table directly at the bottom of the web page.

You just retrieved the properties of the 100 most massive clusters (ordered by mass) for redshift 0 (=snapshot number 85) from the miniMDR1-simulation database!

- o **Query (stream)** has a less strict time out limit and returns data in CSV format, with commas to separate the values.
- o Note that there are **time limits**, so choose carefully which data you really need to download. If you are not familiar with SQL, have a look at <http://www.multidark.org/MultiDark/Help?page=demo/index> and its subpages.
- o In principle, a basic query consists of the following parts:

```
select - followed by a limit (top 10) and a list of columns, e.g.
x,Rvir,npart or * for all
from - followed by a list of tables (possible join),
Bolshoi..BDMV - database and table name
[where] - followed by a list of restrictions (filter), e.g. snapnum=85
[order] - followed by a (list of) columns, e.g. by mass desc
```



- Consult the **documentation** to get to know which table provides which data.

Example: The Cosmic Web with TOPCAT

Load a slice (x-y plane) of the cosmic web for the Bolshoi simulation




- Before actually downloading some data, let's first check where the **most massive halo** is located, using the **BDMV halo** catalogue:
 - o Open a **web browser** with the URL: <http://www.multidark.org>
 - o Go to the **Query Form** and **enter** following query:

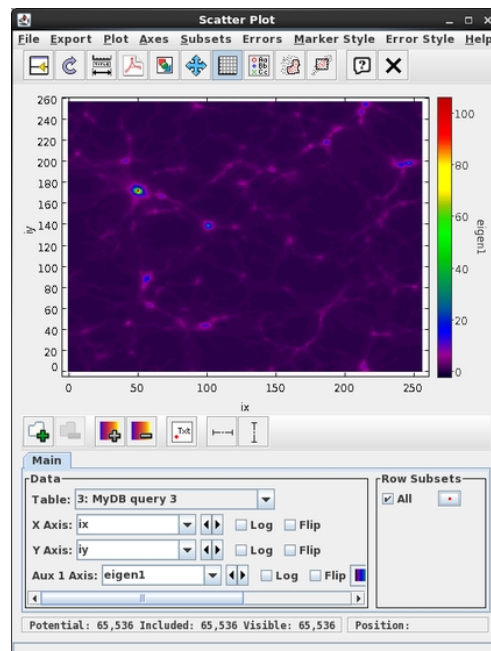
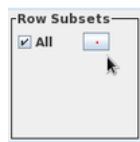

```
select top 1 ix,iy,iz,x,y,z,Mvir from Bolshoi..BDMV
where snapnum=416 order by Mvir desc
```
 - o **Press Query (browser)** to submit the query.
 - o This selects the most massive halo for snapshot number 416 (corresponds to $z=0$ for this simulation) and returns its position and mass M_{vir} within the virial radius, as well as the indices of a 1024^3 grid cell, in which the halo sits (ix,iy,iz) . The size of the grid is given in the documentation.
 - o One can **convert these numbers** into the corresponding indices of a coarser grid by dividing them by **factors of 2**.
E.g. a grid cell with $ix=200$ in a 1024^3 grid corresponds to $ix' = \text{floor}(ix/2.) = 100$ in a 512^3 grid, and $ix' = 50$ for a 256^3 grid. Let the SQL server do it for you:


```
select top 1 ix,iy,iz, floor(ix/4.) as ix_256, floor(iy/4.) as iy_256, floor(iz/4.) as iz_256,
x,y,z,Mvir from Bolshoi..BDMV
where snapnum=416 order by Mvir desc
```
 - o **Press Query (browser)** again to submit the query and check the numbers.
Leave the browser open, since we will need these numbers later on.
- Start [TOPCAT](#), click the **Open File** icon .
- Choose **GAVO Millennium Run query** . In the new dialogue, type following details:
 - o **Base URL:** <http://wget.multidark.org/MyDB>
 - o **username and password:** use your own or the demo user credentials
 - o **SQL Query:** enter a query to retrieve your slice:
 - The required table is Tweb256, which contains a 256^3 grid, and for each grid cell the eigenvalues and eigenvectors for the gravitational tidal tensor.
 - We choose a slice by choosing only rows with a fixed iz -value:




```
select * from Bolshoi..Tweb256 where iz = 100
```
 - o To make it more interesting, we can also choose the slice which contains the center of the most massive halo, so replace the 100 in the query by the **iz_256 value** you found with the browser query in the previous step.
 - o **Press OK** to submit the query.

View the cosmic web

- Have a look at the downloaded data with the **Table Browser**  which you can access from [TOPCAT](#)'s main window. It displays your data as a table.
- **View your data** by clicking on the **Scatter Plot** icon 
 - Choose: **x-axis: ix; y-axis: iy**
 - Add a **new auxiliary axis** by clicking on the color gradient icon .
 - Set **Aux 1** to one of the eigenvalues (e.g. **eigen1**) for coloring the dots.
 - **Increase the point size** by clicking on the colored symbol in Row Subsets, in the box at the lower right of the dialogue window. This opens a new dialogue, where you can also adjust e.g. the transparency of your dots or the marker style.



- Step through **different columns for Aux 1** to get an impression of the other eigenvalues as well. Also look at the pkkeys: These are Peano-Hilbert keys, describing a space-filling curve. They are very important for spatial queries.
- One of the eigenvalues does not yet give much information. Instead, we need to **count the number of eigenvalues** above a certain threshold. This number will allow us to assign each cell with following classification:
 - 0 = void
 - 1 = sheet
 - 2 = filament
 - 3 = knot (= location of galaxy clusters)
- We can get this number by adding a **computed column** to our data.

- o In [TOPCAT](#)'s main window, click on **Display Column Metadata** 
- o Click on the green plus  to **add a new column** to your table.
- o In the new dialogue, enter a **name for your new column**, e.g. num_th0.4
- o Enter the **expression to compute** the column:

$$(eigen1 \geq 0.4 ? 1:0) + (eigen2 \geq 0.4 ? 1:0) + (eigen3 \geq 0.4 ? 1:0)$$

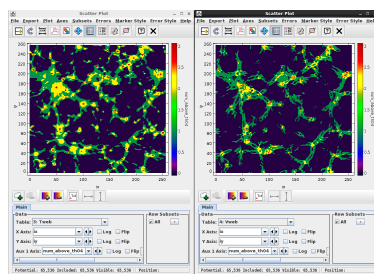
This expression checks, if an eigenvalue is greater than or equal to the threshold value (e.g. 0.4) and adds 1, if this is the case, otherwise 0.
- o **Press OK.** You can still edit the name and expression of your column by double clicking on it in the Column Metadata dialogue.
- o Switch back to the **Scatter Plot Window** and choose the **new computed column as Aux 1 axis**
- o Now you should see some filaments and knots. Note that this is just a cut through the z-direction, so perpendicular sheets appear as filaments and filaments may appear as knots etc.
- o By the way: instead of calculating the additional column with TOPCAT, you could already do it with the SQL query itself:


```
select *,
  (case when eigen1 >= 0.4 then 1 else 0 end)
+ (case when eigen2 >= 0.4 then 1 else 0 end)
+ (case when eigen3 >= 0.4 then 1 else 0 end)
as num_th0.4
from Bolshoi..Tweb256
```

This method would only have the disadvantage that you would need to query the database again for each new threshold value.

Explore more data (or skip and procede with next section)


- Repeat the **same steps for** loading a slice of the same region from the **Vweb256** table.
Compare the differences: the Vweb should give a much better resolution of structures than the Tweb. (Look in the documentation and the linked papers to learn about the physical reasons behind this.)



- Experiment with **different threshold values** by adding more computed columns.
- Experiment with **slices at different positions**. You can also download a thicker z-slice and **plot** it in the **3D** window .

E.g. use:

```
select * from Bolshoi..Vweb256
where iz between floor(0.25*377)-2 and floor(0.25*377)+2
```

- Load a **finer grid** by using the **Vweb512** table and compare with the coarser version. Note, that you need to **convert the iz-value properly** if you want to get matching slices.
 - In **Scatter Plot** window, click on the plus sign  to add another table. Adjust the x,y axis, add again an auxiliary axis and compare.

Find halos in cells

- Open the **table browser** and a **Scatter plot window** for the same data table (e.g. for Tweb256)
- **Click on one of the knots** in your plot and check to which cell it belongs: the corresponding **row** in the table browser is **highlighted** now.
- **Load another table** with TOPCAT by clicking on the **File Open**-icon, then choose **GAVO Millennium query** again and insert a query for retrieving all halos inside this cell, sorted by mass, e.g.:

```
select * from Bolshoi..BDMV
where snapnum=416
      and floor(0.25*ix) = 98
      and floor(0.25*iy) = 137
      and floor(0.25*iz) = 94
order by Mvir desc
```

Replace 98, 137 and 94 by the values for your cell.

- **Press OK** to download the data.
- Switch to the **Scatter plot window** and click on the **+** symbol for **adding a new data set**. Choose the **halos-table** you just downloaded.
 - Set **x-axis: ix/4., y-axis: iy/4.**
 - **Adjust the point size** and color by clicking on the colored symbol at Row Subset.
- Zoom into the cell where the halos are located.
- Open the **table browser** for the halos table. Click on the **most massive halo** in the table browser and watch where it is marked in the Scatter Plot window. **Zoom in** to see if it is lying at the center of a knot region or farther away.