Simulation Analysis with a Halo Catalog Database

Bridget Falck
Johns Hopkins University

Outline

- Motivation
- Halos and photometry
- DB set-up
- Luminosity, Mass, and Correlation function examples
- Mock galaxy catalog

Motivation

- My next project: put a high-resolution cosmological nbody simulation into a database as it is running
  - The simulations will be 500 random instances of the same initial cosmology with the goal of beating down cosmic variance
  - about 100TB worth of data!

- As a warm-up, I decided to create a halo catalog database from the 50Mpc/h cosmological Gasoline simulations and Amiga halos

Halos and photometry

- 50Mpc/h cosmological Gasoline simulations with stars, gas, and dark matter at 7 redshifts (/home/hipacc-29/COSMO)

- Each snapshot has a halo catalog created with the Amiga Halo Finder (Knollmann & Knebe 2009) → x, y, z, vx, vy, vz, mass, v_max, r_max, r_vir, …

- Each halo catalog is run through get_halo_mags.pro, which calculates stellar magnitudes based on theoretical isochrones (home/hipacc-29/ANALYSIS/IDLMags)
  → U, B, V, R, I, K, r‘, i‘, bolometric, age, Z, mass

Setting up the database

- Separate files for halos and photometry at each redshift
- First define the table schema:
  - This sets up the table name and the names and types of each column
- Then load the data into the tables:
  - Many ways to load data, with `Bulk Insert` being convenient for ASCII files

```
create table hstat (
    hid float, ntot float, ngas float,
    nstar float, ndark float, mtot float,
    rvir float, mgas float, mstar float,
    mdark float, vmax float, rmax float,
    vdisp float, xc float, yc float, zc
    float, vxc float, vyc float, vzc
    float, satellite int, z float
)
go
create table hmags (
    grp float, mbol float, u float, b
    float, v float, r float, i float, k
    float, rprime float, iprime float,
    metallicity float, age float, mass
    float, redshift float
)
go
BULK INSERT hstat FROM
    '\\skydev\c$\Users\bfalck.SDSS\Downloa
    ds\hstat048.csv'
WITH (FIELDTERMINATOR =',',
    ROWTERMINATOR ='|')
(etc…)
```

Joining and altering tables

- Join the `hstat` and `hmags` tables based on halo id and redshift:
  - `s.*` picks everything from table `s`; leave out `m.grp` and `m.redshift`
  - Don't need to define a new schema for `hcat`

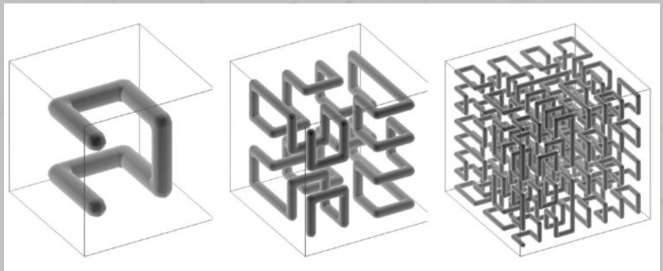
```
select s.*, m.mbol, m.u, m.v, m.b,
    m.r, m.i, m.k, m.rprime,
    m.iprime, m.metallicity, m.age,
    m.mass
into hcat
from hstat s inner join hmags m
on s.hid = m.grp and s.z =
    m.redshift

alter table hcat add phkey int

update hcat set phkey =
    dbo.fPHkey(8,convert(int,(xc/50)*
    256),convert(int,(yc/50)*256),con
    vert(int,(zc/50)*256))
go

create index idx_hcat_phkey on hcat
    (phkey,xc,yc,zc)
```

- Create Peano-Hilbert index:
  - Allows for fast spatial searching
  - The function dbo.fPHkey was created at JHU and added to the SDSS server



Luminosity Functions

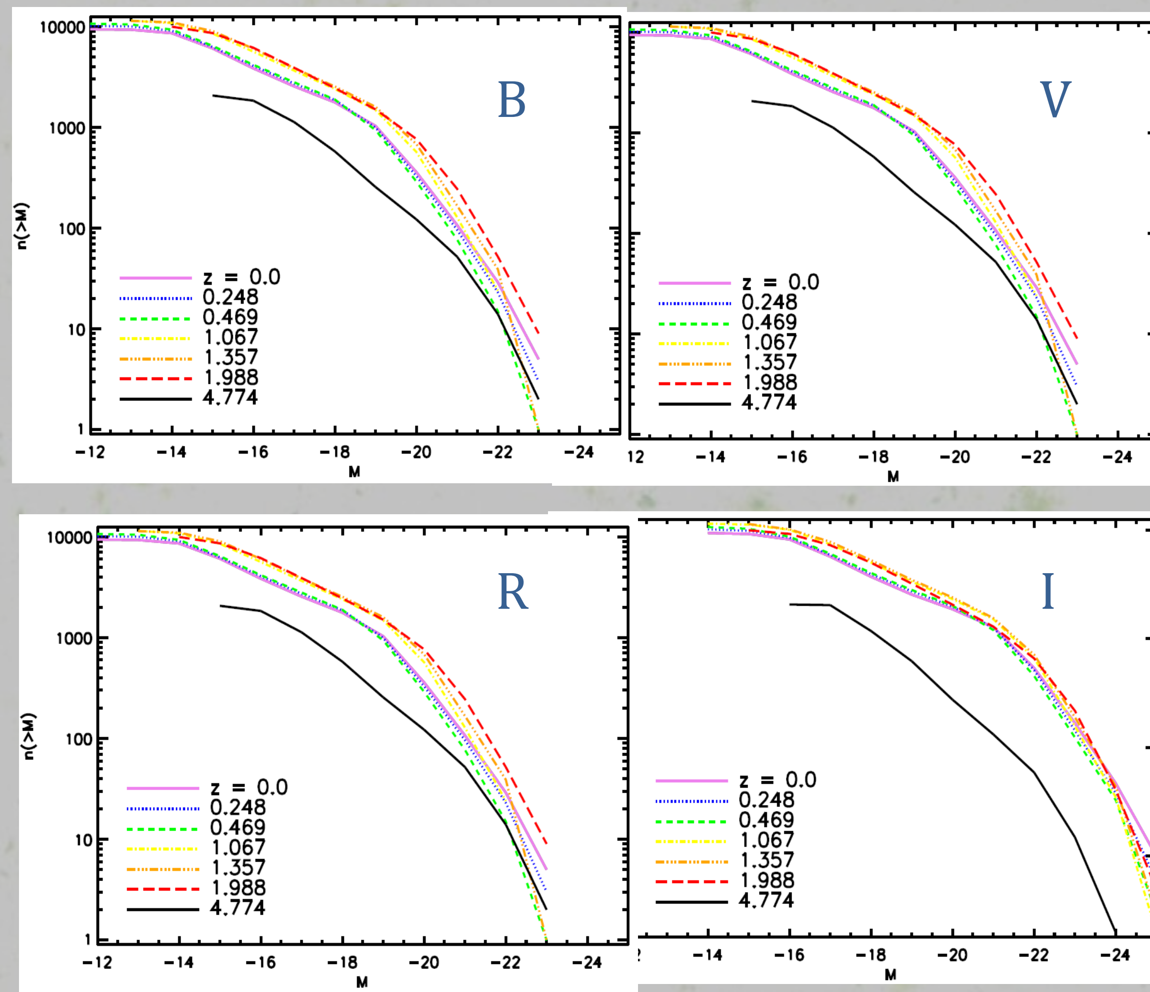
-- Histogram the magnitudes:

```
select cast(mbol as int)
     mag, count(*) c, z
into #lf
from hcat
where ...... (add what you like!)
group by cast(mbol as int),z
order by 3,1 desc
```

-- Convert LF into cumulative sum *:

```
select a.mag, a.c, sum(b.c)
     as s, a.z
from #lf a cross join #lf b
where (b.mag <= a.mag) and
     a.z = b.z
group by a.mag, a.c, a.z
order by 4,1 desc
```

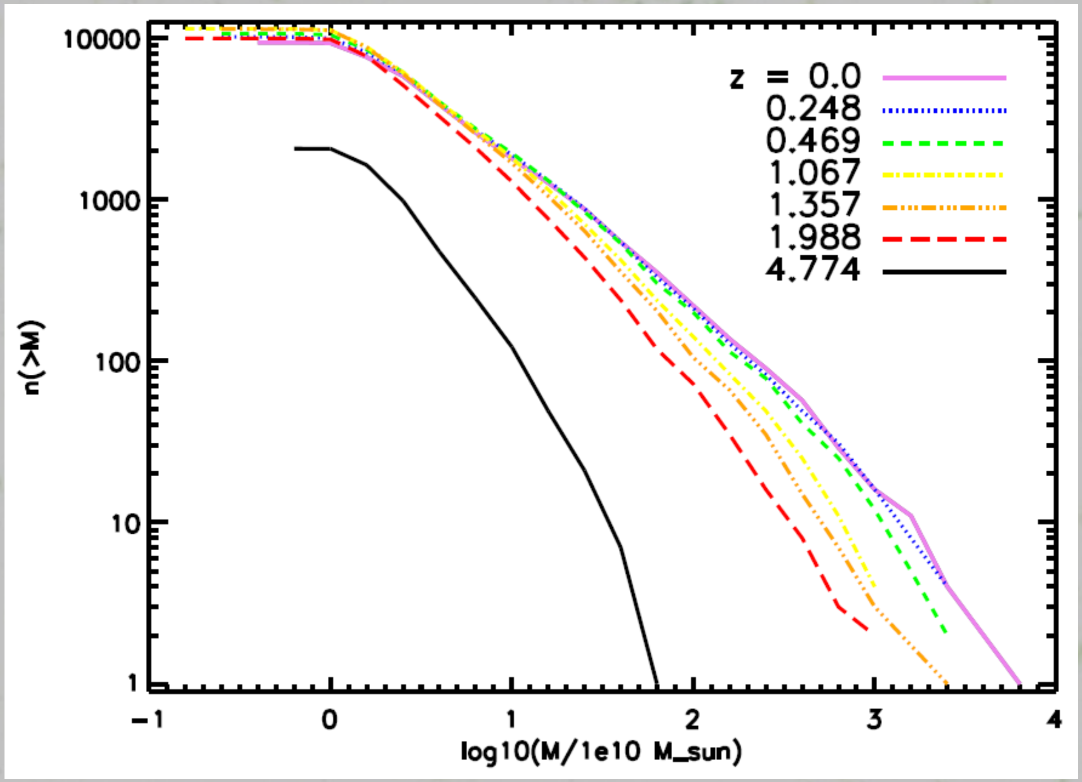
*code to do this found with google after I gave up…



Mass Functions

```
select
     cast(5*log10(mtot/1e10)
     as int)/5. as lmass,
     count(*) c, z
into #massfn
from hcat
group by
     cast(5*log10(mtot/1e10)
     as int)/5., z
order by 3,1

select a.lmass, a.c,
     sum(b.c) as s, a.z
from #massfn a cross join
     #massfn b
where (a.lmass <= b.lmass)
     and a.z = b.z
group by a.lmass, a.c, a.z
order by 4,1
```



Correlation Functions

- The correlation function counts how many halos are separated by *r*, averaging over halos at all *x*:

$$\xi(r) = \langle n(x+r)n(r) \rangle_x$$

- First create a pairs table with the distances between halos
  - My first (incorrect) attempt took only 16 minutes to create 300 million pairs!
  - Need to prevent duplicates (`h2.hid > h1.hid`) and constrain the redshift

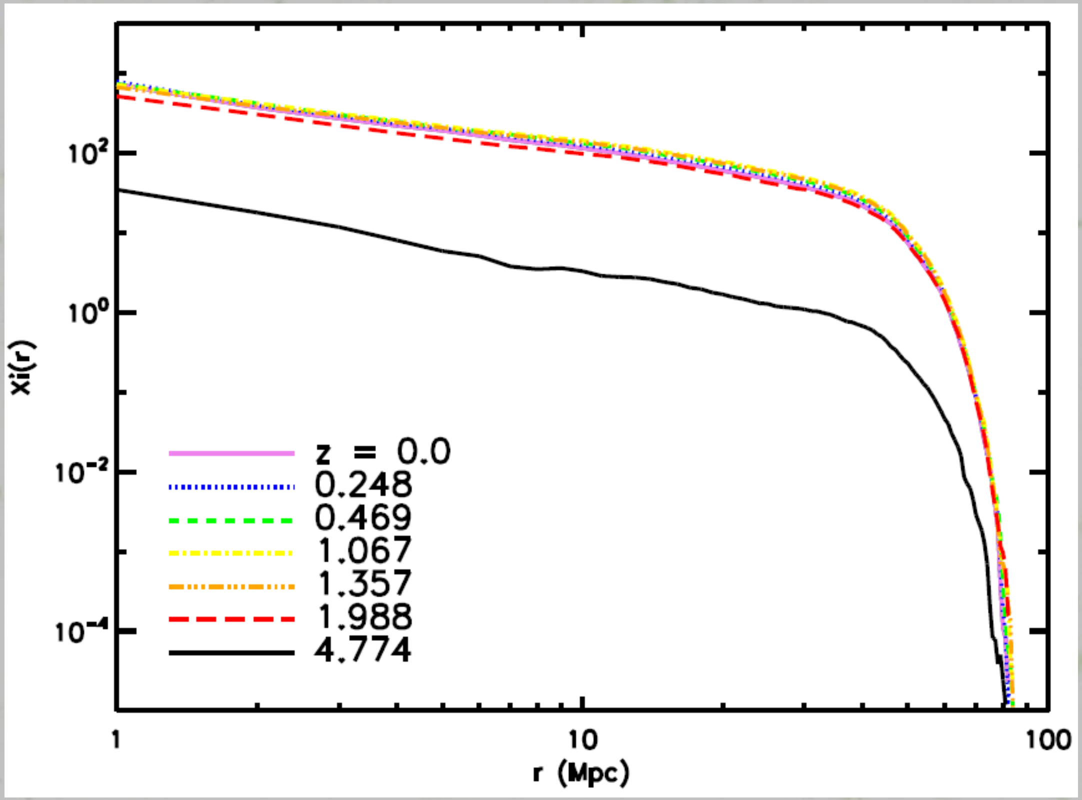
```
select sqrt(power(h1.xc-
   h2.xc,2)+power(h1.yc-
   h2.yc,2)+power(h1.zc-
   h2.zc,2)) as dist,
   h1.hid hid1, h2.hid hid2,
   h1.mtot/1e10 mass1,
   h2.mtot/1e10 mass2, h1.z z
into hpairs
from hcat h1, hcat h2
where h1.ntot > 200
   and h2.ntot > 200
   and h1.z = h2.z
   and h2.hid > h1.hid
```

* It took 4 minutes 44 seconds to create this table of 110 million pairs!

Correlation Functions

-- Histogram the distances and normalize by the volume of each bin:

```
select cast(dist as int)
    as r, count(*) /
    ((4*3.14/3)*(3*power(
    cast(dist as
    int),2)+3*cast(dist
    as int)+1)) as x, z
from hpairs
group by cast(dist as
    int),z
order by 3,1
```



* Selecting from 110 million pairs took only 33 seconds

Creating a mock galaxy catalog

- Define a `Simulation` instance given by the box size, and a `Shape3D` object (`Cone`, `Box`, `Sphere`, or the `Union` or `Intersection` of other shapes)
  - (created at JHU by Tamás Budavári and others)
  - Contain ways to handle PBCs and Peano-Hilbert indexing
- Select only halos with z between 700 and 900 Mpc, given by a cosmological distance function (Taghizadeh-Popp 2010; dev.skyserver.org/cfunbase)

```
declare @sim Simulation
set @sim = Simulation::newInstance(8,
    0,0,0,50,50,50)
declare @qshape Shape3D
set @qshape =Shape3D::newInstance('Cone
    [10,10,10,1,2,3,0.05,900]')
select * into fsim
from dbo.fSimulationCoverShape(@sim,
    @qshape,8)
select *,sqrt(power(xc+ShiftX-10,2) +
    power(yc+shifty-10,2) +
    power(zc+shiftz-10,2)) as d
into obs
from hcat h inner join fsim fs
on h.phkey between fs.keymin and fs.keymax
where fs.FullOnly=0
    and @qshape.ContainsPoint(xc+shiftx,
    yc+shifty, zc+shiftz) = 1
    and h.z between
    dbo.fCosmfZfromDc((700)*(1+h.z)) and
    dbo.fCosmfZfromDc((900)*(1+h.z))
```

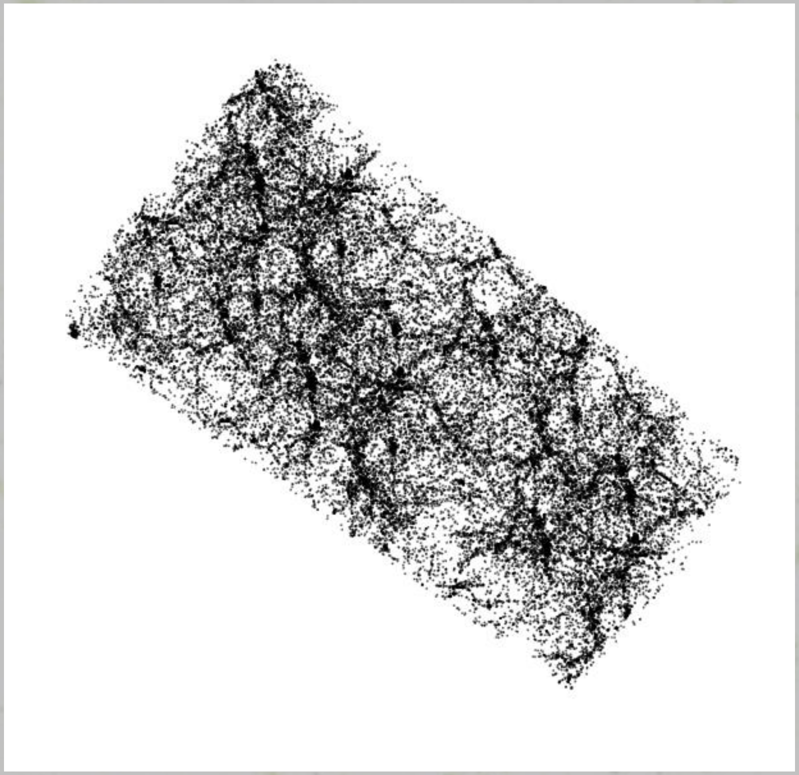
Mock Observations

-- Get mock redshift, RA, and Dec:

```
select hid, d,
   dbo.fCosmfZfromDc(d*(1+z))
   zobs, asin((zc+shiftz-10)/d) as
   dec, atan((yc+shifty-
   10)/(xc+shiftx-10)) as ra, u,
   b, v, r, (etc…)
into #tmp
from obs
where d between 700 and 850
order by d
```

-- Get apparent magnitudes:

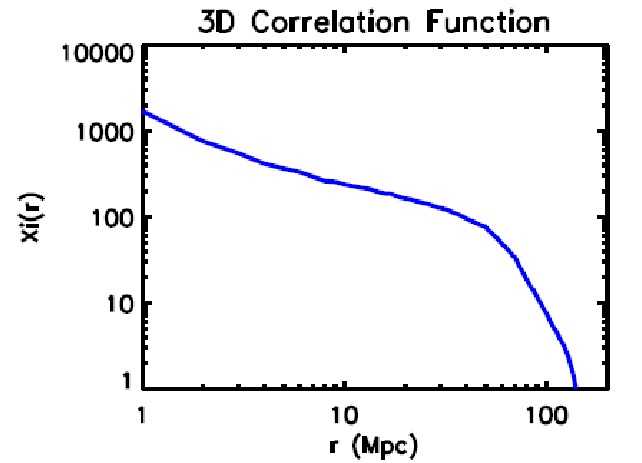
```
select hid,zobs,ra,dec,
   5.*log10(dbo.fCosmfDl(zobs))+25
   as mu, u +
   5.*log10(dbo.fCosmfDl(zobs))+25
   as magu, (etc…)
from #tmp
```

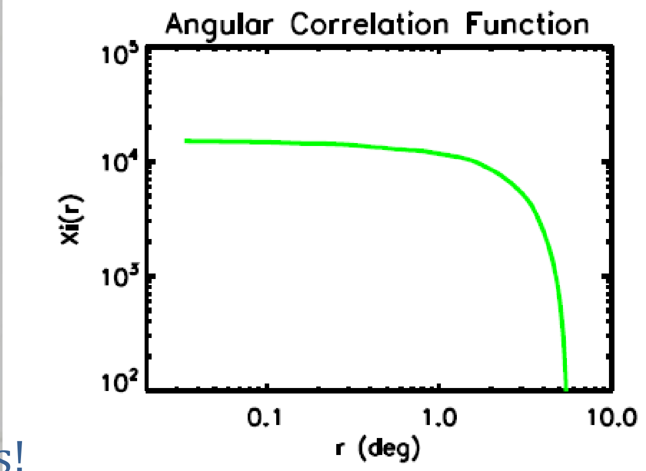


More Correlation Functions

-- Angular correlation function:

```
select degrees(acos(case when dist >
    1.0 then 1.0 else dist end)) rad,
    hid1, hid2, d1, d2
into #pairtmp from (
select
    t1.xunit*t2.xunit+t1.yunit*t2.yun
    it+t1.zunit*t2.zunit dist, t1.hid
    hid1, t2.hid hid2, t1.d d1, t2.d
    d2
from #postmp t1, #postmp t2
where t1.ntot > 500 and t2.ntot >
    500 and t1.d > t2.d
) q
select cast(rad*30. as int) as r,
    count(*) /(3.14*(cast(rad*30. as
    int)+1)) as s
from #pairtmp
group by cast(rad*30. as int)
order by 1
```





* Took 8-9 minutes for 170 million pairs!

Conclusion

- Databases are the future...
  - LSST will collect 30TB of data per night; NVO/AVO online repositories of observational data; etc.

- Fast and simple, with complexity added as needed
  - i.e. user-defined functions, spatial indexing

- Analysis where the data is stored
  - Fast correlation functions and other analyses in addition to fast searches
  - library of cosmological functions, dev.skyserver.org/cfunbase

- Free SQL options: SQL Express, MySQL, Sqlite (with IDL and python modules available), … SciDB?