

SMBH-TreeMerger: tracking the merger history of SMBHs

Ulyana Dupletsa

Abstract

`SMBH-TreeMerger`¹ is a `python` script that processes SMBH merger trees, relying on data from the Millenium simulation. The code starts from row data, where the central black holes of two galaxies are considered merged as soon as the respective galaxies merge and in post process implements the evolution of the black hole binaries.

A SMBH binary undergoes different phases: dynamical friction, hardening (via encounters with single stars or shrinking due to the interaction with a gaseous disk) up to a final gravitational wave driven phase. Then the binary eventually coalesces. If the time to coalescence exceeds the time span between two subsequent galactic mergers, triplet or quadruplet systems form and are taken care of as well. When the system is a quadruplet (the less likely scenario) it is immediately reduced to a triplet by ejecting the least massive black hole. Triplets are resolved assigning probabilities to have either a prompt merger between the two most massive black holes or an ejection of one of the three (usually the lightest one), which in both cases leads to a new binary system. The left off binary then follows the pattern to merger described above.

The final output serves to the calculation of merger rates as a function of redshift and of the gravitational wave background strain (GWB).

Contents

1	Introduction	2
2	Data Retrieval	3
2.1	A note on the Millennium simulation	3
2.2	From raw data to injection files	3
2.2.1	Manipulate raw data files	3
2.2.2	Generate injection files	5
3	Analysis of merger trees	6
3.1	The main file	6
3.2	Modules interdependence	7
3.3	Merger analysis: binaries, triplets, quadruplets	8
3.3.1	Binary mergers	9
3.3.2	Triplet mergers	10
3.3.3	A note on the lookback time	13
4	Astrophysical background	14
4.1	Density profiles	14
4.1.1	Defining the effective radius R_{eff}	14
4.1.2	SIS model	14
4.1.3	A case of Dehnen profiles: Hernquist model	14
4.2	Delay Times	16
4.2.1	Dynamical Friction Phase	16

¹GitHub repository: [u-dupletsa/SMBH_TreeMerger](https://github.com/u-dupletsa/SMBH_TreeMerger)

4.2.2	Hardening Phase	17
4.2.3	Gravitational Waves Phase	19
5	Output and Analysis	19
5.1	Output format	19
5.2	Characteristic Background Strain	20
5.3	Merger Rates	20
6	Comments and Conclusions	20
A	Bash scripts	20
A.1	Get trees from root galaxies	20

1 Introduction

SMBHs are ubiquitous: almost every galaxy harbors a SMBH at its center [1] and many galaxies are expected to have undergone multiple mergers throughout their cosmic evolution. As a consequence, we can expect galaxies to host SMBHBs. However, the processes of SMBHB formation and evolution are still unclear and therefore represent an active research field. To this scope, the objective of this new code is to study the SMBHB merger rates and GWB, adopting different available models, and relying on population data taken from the Millennium simulation [2].

The main point of the analysis is to take into consideration time delays between galaxy mergers and mergers between the two black holes at their centers. Since the simulations data assume that ones two galaxies have merged their black holes are merged as well, we have to consider dynamics in post process. Pairing dynamics itself is still not completely understood. However, it is generally accepted that the following are the stages to be considered [3]: dynamical friction phase, stellar and/or gaseous hardening phase and gravitational wave emission phase, which leads to the final coalescence.

- The evolution begins at kpc scale separation, when the SMBHs, at the centre of their respective host galaxies, sink, due to dynamical friction, towards the centre of the remnant galaxy (see [4, 5, 6]).
- Once the two SMBHs have entered the influence radius of one another, on \sim pc scales, a bound system forms. The distance, however, is still too large for gravitational waves to bring the binary to coalescence. There are two ways to shrink the binary separation and trigger GW emission:
 1. Since the binary is immersed in the stellar background of the bulge, interaction with individual low-angular-momentum stars shrinks (hardens) the binary. Through three-body scatterings with single stars, in fact, the binary loses angular momentum and shrinks its separation. If the reservoir of low-angular-momentum stars is sufficient, the binary is likely to reach \sim millipc separations. This process is called the stellar hardening phase (see [7]).
 2. On the other hand, in gas-rich environments, the binary can reach millipc separations thanks to the formation of a massive circumbinary disk. Interaction with the disk can extract energy and angular momentum from the binary driving the final coalescence. This is known as the gas driven hardening phase (see [8]).
- In the final stage, GW emission takes over the process and the binary is brought to coalescence.

All of these stages are implemented in the code and every time the delay time, i.e. the time it takes for a binary to merge, exceeds the time between two subsequent galactic mergers we form a triple system. In some cases, when both of the progenitor galaxies originate from previous mergers that had not had enough time for the black holes to merge, we have quadrupole systems. In the last case we simply eject the least massive black among the four and treat the system as a triplet. One crucial point in the code is the treatment of

triple systems, which is done in the light of recent numerical simulations of triplet dynamics (see [9, 10]).

The objective is to study how the GW signal is modified after taking into consideration time delays and triplets dynamics. To a first approximation, the GW emitted frequency scales inversely with the binary mass. Since we are dealing with SMBHs, with masses in the range $10^8 - 10^{10} M_\odot$, the frequency of our signal will fall in the nHz band (nHz- μ Hz). Moreover, we do not have resolved events, but the most likely signal to be detected is a stochastic GWB, generated by the incoherent superposition of GWs from the whole cosmic population of SMBH sources.

To detect such GWs one would need a galactic-size interferometer and, indeed, we can have one considering the pulsars in our galaxy. Pulsars are the final products of the evolution of the most massive stars. During the gravitational collapse of the supernova that gave birth to them, they acquire an extremely fast and stable rotation, which gives that incomparable properties of atomic clocks. Low-frequency GWs emitted by SMBHB systems would leave a characteristic imprint on the time of arrivals of radio pulses from a set of ultra-stable millisecond pulsars that form the PTAs (Pulsar Timing Array) experiments. In general, one computes, the difference between the expected arrival time, according to a given model, and the actual arrival time. The detection with PTAs is expected to measure the first stages of coalescence of a SMBHB, in the frequency window between $10^{-9} - 10^{-6}$ Hz.

Implementing time delays and triple interactions allows us to make more realistic predictions about the expected merger rates and GWB for SMBH binaries.

In this report we describe in Sec. 2 how data for injections is prepared. Then we outline the main algorithms and the logic behind the `SMBH-TreeMerger` code in Sec. 3.

2 Data Retrieval

2.1 A note on the Millennium simulation

The Millennium Simulation [2] is a simulation of the growth of dark matter structures using 2160^3 particles, from redshift $z = 127$ to present epoch, in a cube-shaped region of $500/h$ Mpc side length. In post-processing, the formation and evolution of galaxies and quasars is followed.

Black hole data have to be selected according to the mass range of interest for PTAs. In other words, black holes with masses greater than 10^8 solar masses should be considered as the main sources contributing to the GWB signal.

Since data from Millennium simulation are not public, in the rest of this section we explain how to treat raw data to obtain injection files that can be processed with `SMBH-TreeMerge`. On this line, the GitHub repository contains no data files.

2.2 From raw data to injection files

We work with 4 catalogs taken from the Millennium Simulation: Bertone, De Lucia, Guo_2010 and Guo_2013. All the mergers are taken at 64 different snapshots, from `snapnum=0`, the most ancient one, to `snapnum=63`, the last one at present time, $z = 0$. All the mergers between `galaxyID` taken at the last `snapnum` and `lastProgenitorId` (see A.1) represent the merger tree that brought to the given galaxy at the last `snapnum`. On the following page, one can find all the information about the `millennium_database`.

2.2.1 Manipulate raw data files

We start with two data files:

1. `all_roots-<catalog>.txt`: this file contains two columns, `galaxyId` and `lastProgenitorId`, so that all the mergers with a `galaxyId` comprised between these two numbers uniquely identify a tree.

2. `all_mergers_<catalog>.txt`: this file contains all the mergers within a catalog. The data provided is the following:

```
1 ['galaxyId', 'lastProgenitorId', 'snapnum', 'descendantId', 'P1_Id', 'P2_Id', 'D_z', 'D_mass', 'D_bulge', 'sfr', 'sfr_Bulge', 'D_BH', 'P1_z', 'P2_z', 'M1', 'M2', 'P1_bulge', 'P2_bulge', 'P1_stars', 'P2_stars', 'M_cold', 'M_hot', 'V_vir', 'P1_cold', 'P1_hot', 'P1_V_vir', 'P2_cold', 'P2_hot', 'P2_V_vir']
```

All the mass variables are measured in $10^{10} M_{\odot}/h$ (`D_mass`, `D_buge`, `D_BH`, `M1`, `M2`, `P1_bulge`, `P2_bulge`, `P1_stars`, `P2_stars`, `M_cold`, `M_hot`, `P1_cold`, `P1_hot`, `P2_cold`, `P2_hot`). `sfr` and `sfr_Bulge` are measured in M_{\odot}/year and velocities, `V_vir`, `P1_V_vir`, `P2_V_vir`, are measured in km/s.

Starting from these two data files we produce the starting data for the analysis. The following scripts are in order of application:

1. `all_trees.py`

- input: `all_roots_<catalog>.txt` and `all_mergers_<catalog>.txt`
- output: `all_trees_ini_triplets_<catalog>.csv`

Starting from `all_roots_<catalog>.txt` and `all_mergers_<catalog>.txt` we produce all the initial trees `all_trees_ini_triplets_<catalog>.csv`. Subsequent trees are separated by a line of `-1s`:

```
1 galaxyId, ..., P1_Id, P2_Id, D_z, ...
2 ...
3 4, ..., 5, 2594, 0.08928783, ...
4 1, ..., 2, 2881, 0.019932542, ...
5 1, ..., 2, 2941, 0.019932542, ...
6 1, ..., 2881, 2941, 0.019932542, ...
7 -1, -1, -1, -1, -1, ...
8 15478, ..., 15479, 15530, 1.6302707, ...
9 15477, ..., 15478, 15573, 1.5036365, ...
10 15468, ..., 15469, 15661, 0.6871088, ...
11 -1, -1, -1, -1, -1, ...
12 ...
```

2. `all_filter_triplets.py`

- input: `all_trees_ini_triplets_<catalog>.csv`
- output: `all_filtered_<catalog>.csv`

Starting from data produced at the previous step, `all_trees_ini_triplets_<catalog>.csv`, we remove all the multiple mergers happening at the same `snapnum`, by reducing them to at most 3, the more massive ones, ordered by mass. More precisely, in a given tree there could be more lines containing the same `galaxyId` and accounting for multiple mergers (the number of lines corresponds to the number of ways one can choose a couple of black holes from the set of merging black holes, i.e. from a minimum of 3 lines if 3 black holes merge, 6 lines if 4 black holes merge and so on). Here we keep the most massive 3. The new output is `all_filtered_<catalog>.csv`.

3. `all_find_triplets.py`

- input: `all_filtered_<catalog>.csv`

- output: `all_triplets_same_snapnum_<catalog>.csv`

Indexes corresponding to triplets in `all_filtered_<catalog>.csv` are stored in a new file `all_triplets_same_snapnum_<catalog>.csv`.

4. `all_drop_triplets_same_snapnum_<catalog>.csv`.

- input: `all_filtered_<catalog>.csv` and `all_triplets_same_snapnum_<catalog>.csv`
- output: `all_original_triplets_data_<catalog>.csv` and `all_starting_data_<catalog>.csv`

Using the lines identifying triplets, we processed the triple systems using the Bonetti's probabilities of either prompt merger or ejection. The most massive combination (the first row as the triplets are mass ordered) is taken to be the inner binary, whereas the least massive black hole is treated as the intruder. Using Bonetti's probabilities, in 20% of cases the triplets are resolved and give back a binary system, either as a prompt merger of two of the black holes (the remnant forms a binary with the third one) or as an ejection of one of the black holes (the remaining two form a binary). Non resolved triplets are manually discharged leaving the most massive merger only (the first line). Data from the original triplets is stored in `all_original_triplets_data_<catalog>.csv`.

The filtered data is stored in `all_starting_data_<catalog>.csv`.

5. `all_order_trees.py`

- input: `all_starting_data_<catalog>.csv`
- output: `all_starting_ordered_<catalog>.csv`

Data filtered of multiple mergers are ordered such that in each tree mergers start from the most ancient to the most recent. The final data file is stored in `all_starting_ordered_<catalog>.csv`.

The same process is applied to data where at the current `snapnum` the central black hole has a mass greater than $10^8 M_{\odot}$. The difference is that the prefix `all_` is substituted with `sel_`.

2.2.2 Generate injection files

The starting data are stored in the 4 files, named as `sel_starting_ordered_<catalog>.csv`, each for one of the four catalogs (De Lucia, Bertone, Guo2010 and Guo2013). The columns are the same as the ones downloaded from the database (see above).

Before analyzing the trees, we need to choose both a density profile model for the galaxies and a mass model for the central black holes. Currently we have two models for the galaxies' density profile: either the SIS model (Singular Isothermal Sphere) or the Hernquist model (which is a particular case of more general Dehnen density profiles with $\gamma = 1$ coefficient). For black hole masses one can either use the one provided by the Millennium simulation or use the Kormendy&Ho empirical relation between bulge mass and central black hole mass (see [11]). In this latter case we populate both the progenitors and the remnant galaxies with central black holes according to Kormendy&Ho and then the progenitors are further processed so that their total mass equals the remnant mass and their mass ratio is the same as the original mass ratio coming after applying Kormendy&Ho to the progenitors bulge mass values.

We have two injection files: `injection_<catalog>_<mass_model>_<density_model>.csv` and `tree_indexes_<model>.csv`:

- `injection_<catalog>_<mass_model>_<density_model>.csv`
Module `model.py` provides the injection file, `injection_<catalog>_<mass_model>_<density_model>.csv`, where more columns are added with respect to the data taken from the Millennium simulation. The injection file has the following columns:

```
1 ['galaxyId', 'lastProgenitorId', 'snapnum', 'descendantId', 'P1_Id', 'P2_Id', 'D_z', 'D_mass', 'D_bulge', 'sfr', 'sfr_bulge', 'D_BH', 'P1_z', 'P2_z', 'M1', 'M2', 'P1_bulge', 'P2_bulge', 'P1_stars', 'P2_stars', 'M_cold', 'M_hot', 'V_vir', 'P1_M_cold', 'P1_M_hot', 'P1_V_vir', 'P2_M_cold', 'P2_M_hot', 'P2_V_vir', 'bh_mass', 'P1_BH_mass', 'P2_BH_mass', 'q', 'mass1', 'mass2', 'r_eff_P1', 'r_inf_P1', 'sigma_P1', 'r_eff_P2', 'r_inf_P2', 'sigma_P2', 'host_r_eff', 'host_sigma', 'satellite_sigma', 'satellite_BH', 'host_BH', 'r_eff', 'r_inf', 'sigma_inf', 'rho_inf', 'm_dot', 'hardening_type']
```

where `bh_mass`, `P1_BH_mass` and `P2_BH_mass` are the masses obtained using the Kormendy&Ho relation (otherwise they just equal the ones provided by the Millennium simulation). Then, the two masses of a binary are calculated as follows:

$$\begin{aligned} q &= P1_BH_mass/P2_BH_mass \\ mass1 &= q/(1. + q) * bh_mass \\ mass2 &= 1./(1. + q) * bh_mass \end{aligned} \tag{1}$$

`r_eff`, `r_inf` in pc and `sigma` in km/s for both progenitors and remnant galaxies indicate, respectively, the effective radius, the influence radius, and the velocity dispersion dependent on the chosen density profile (see report on results). `host_` and `satellite_` labels distinguish between the most and least massive galaxies in a merger.

Finally, `m_dot` quantifies the accretion rate for gaseous hardening and is measured in M_{\odot}/s . At this stage also `sfr` and `sfr_bulge` have been converted to M_{\odot}/s . `hardening_type` tells whether the hardening phase is only stellar, only gaseous, or both. If `hardening_type=1` then it means that `sfr=0` and the hardening phase is stellar only, otherwise both gas and stars are considered and the processes that last less dominate this phase.

- `tree_indexes_<model>.csv`
Module `find_tree_indexes.py` provides the other injection file, `tree_indexes_<model>.csv`, where two columns, with respectively the start and the end index of all the trees in a catalog, are contained.

3 Analysis of merger trees

3.1 The main file

Module `main.py` launches the tree analysis after having checked that both of the injections files are present (otherwise they are created):

```
1 # Injection files paths
2 path_data = '%s/injection_%s_%s_%s.csv' %(str(data_folder), str(catalog),
3     str(mass_model), str(density_model))
4
5 # Check whether the specified path exists or not
6 if os.path.exists(path_data):
```

```

7 data = pd.read_csv(path_data, names = lbs, skiprows = 1, delimiter = ',')
8 print('Opening data file')
9 else:
10 print('Data file does not exist, generating file')
11 model.generate_input(catalog, mass_model, density_model, h)
12 data = pd.read_csv(path_data, names = lbs, skiprows = 1, delimiter = ',')
13 print('Opening data file')
14
15 if os.path.exists(path_index):
16 index_data = pd.read_csv(path_index, names = ['start', 'end'], skiprows =
17     1, delimiter = ',')
18 print('Opening index file')
19 else:
20 print('Index file does not exist, generating file')
21 find_tree_indexes.find_indexes(catalog, mass_model, density_model)
22 index_data = pd.read_csv(path_index, names = ['start', 'end'], skiprows =
23     1, delimiter = ',')
24 print('Opening index file')
25
26 tree_start = index_data['start'].copy()
27 tree_end = index_data['end'].copy()
28
29 print('Launching tree analysis')
30 tree.tree(catalog, density_model, mass_model, omega_matter, omega_lambda,
31     data, tree_start, tree_end)

```

The catalog, mass model and density profile have to be specified at the beginning of `main.py`. The cosmological parameters are catalog dependent and are retrieved from the `settings.yaml` file:

```

1 # Select catalog and model
2 #catalogs = ['de_lucia', 'bertone', 'guo2010', 'guo2013']
3 catalog = 'de_lucia'
4 mass_model = 'KH' # KH or millennium
5 density_model = 'isothermal' # isothermal or dehnen
6
7 with open('settings.yaml') as f:
8     doc = yaml.load(f, Loader=yaml.FullLoader)
9 catalog_properties = doc[catalog]
10 h = eval(str(catalog_properties['h']))
11 omega_matter = eval(str(catalog_properties['omega_matter']))
12 omega_lambda = eval(str(catalog_properties['omega_lambda']))

```

3.2 Modules interdependence

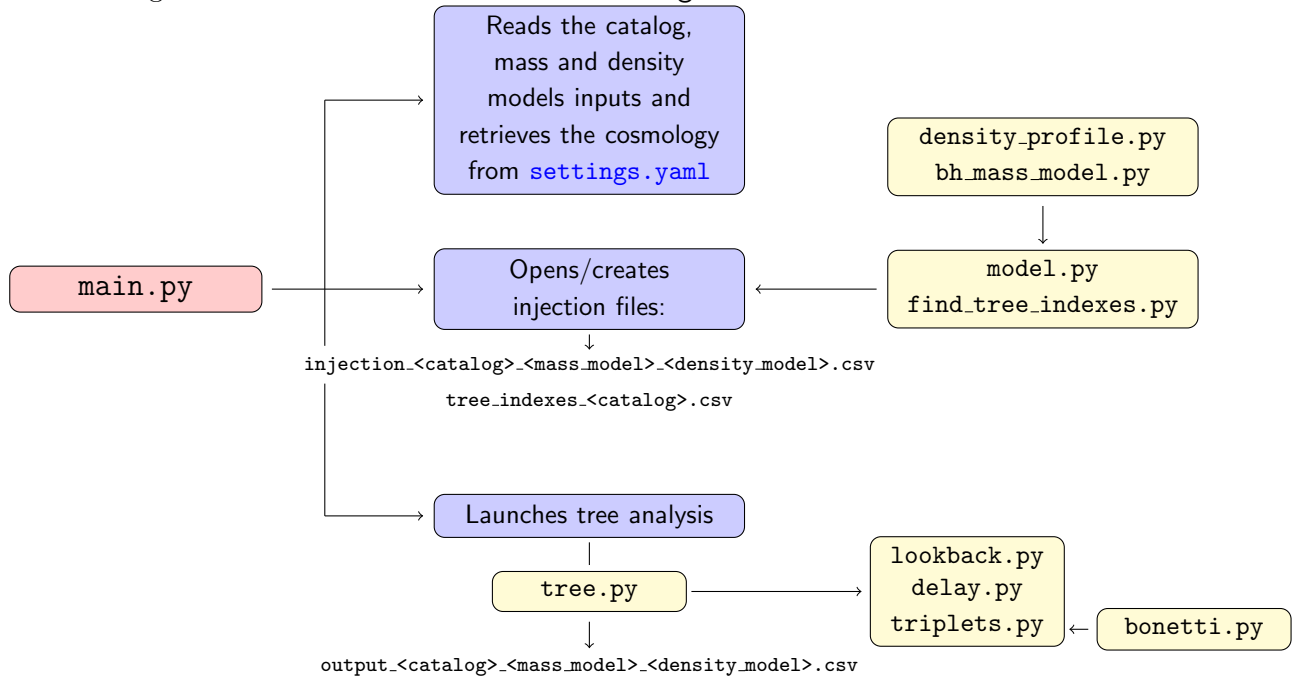
All the analysis is done in the `tree.py` module, which in turn calls other modules. Each tree is then analyzed separately by the `tree` function:

```

1 for i in tqdm(np.arange(n_trees)):
2     start = tree_start[i]
3     tree_index = tree_end[i]
4 # Analyze individual trees
5 for k in range(start, tree_index):

```

In the following scheme it is sketched how **SMBH-TreeMeger** works:



3.3 Merger analysis: binaries, triplets, quadruplets

We create 4 vectors (`type_P1`, `type_P2`, `P1_marker` and `P2_marker`) to store data which will allow us to analyze the tree. They are vectors long as the total number of lines of the data file, so that data at the same index correspond to information on the merger at that index.

`type_P1`, and `type_P2` are vectors that store information on the two progenitors, P1 and P2, both initialized with 0 values:

- If the value is 0 then the progenitor is a single black hole
- If the value is 2 then the progenitor is a binary (it means that it hasn't successfully merged before the subsequent galaxy merger)

Therefore we can have 4 different combinations depending on the type of the progenitors:

```

1 ##### Combination (1) #####
2 if (type_P1[k] == 0 and type_P2[k] == 0):
3     # BINARY!
4     form_binary_vector[k] = 1
5     ... ANALYSIS ...
6
7 else:
8     ##### Combination (2) #####
9     if (type_P1[k] == 0 and type_P2[k] == 2): # P1 is the intruder, P2 is a
10         binary
11         # TRIPLET!
12         form_triplet_vector[k] = 1
13         ... ANALYSIS ...

```



```

14 ##### Combination (3) #####
15 if (type_P1[k] == 2 and type_P2[k] == 0): # P1 is a binary, P2 is the
    intruder
16 # TRIPLET!
17 form_triplet_vector[k] = 1
18 ... ANALYSIS ...
19
20 ##### Combination (4) #####
21 # QUADRUPLET!
22 # Both P1 and P2 are binaries
23 if (type_P1[k] == 2 and type_P2[k] == 2):
24 form_quadruplet_vector[k] = 1
25 ... ANALYSIS ...

```

Except for the first case, we have to implement the triplet interaction in all the other cases. `P1_marker` and `P2_marker` store the index on where to find the information on the progenitors in case these are not single black holes.

3.3.1 Binary mergers

Both progenitors are single black holes (`type_P1=0` and `type_P2=0`), which is the case at the beginning of any tree:

- Suppose we have the binary merger at `z[k]` (both progenitors are single black holes). All the information about the binary is stored at index `k`.
- We calculate the time delay, `time_to_merge` necessary for the two central black holes to merge (see notes on time delay for the formulae employed)
- Then we search along the tree to find its first descendant (using one of the `lookback.py` functions). This is pretty easy as `galaxyIds` in a tree are ordered for a depth-first search, that means that if the difference of two `galaxiesIds` equals the difference in `snapnums` then the galaxy at higher `snapnum`² descends from the galaxy at the lower `snapnum`:

```

1 # Find descendant function
2 # Set to zero the descendant type (progenitor 1 and 2)
3 P1 = 0
4 P2 = 0
5 descendant_index = -1 # default value if there is no descendant
6 for l in range(k + 1, tree_index):
7     if ((snapnum[l] - snapnum[k]) > 0):
8         if ((snapnum[l] - snapnum[k]) == (1 + galaxyId[k] - P1_galaxyId[l])):
9             :
10            P1 = 1
11            z_descendant = redshift[l]
12            descendant_index = l
13            break
14        if ((snapnum[l] - snapnum[k]) == (1 + galaxyId[k] - P2_galaxyId[l])):
15            :
16            P2 = 1
17            z_descendant = redshift[l]
18            descendant_index = l

```

²Snapnums are ordered from 0 to 63, where `snapnum=63` corresponds to present day.

```

17     break
18     if (descendant_index == -1): # no descendant
19         z_descendant = 0
20
21     return int(descendant_index), int(P1), int(P2), z_descendant

```

The function gives back the index at which the first descendant is and specifies whether the descendant is the first progenitor P1 or the second progenitor P2.

- At this point we calculate the time elapsed between the redshift, $z[k]$, at which this merger happens and the redshift, $z_descendant$, at which the descendant merger will occur (for the calculation for the time see the paragraph 3.3.3). We call this time `time_to_next_merger`.
- If `time_to_merge > time_to_next_merger`, then at the subsequent merger either one of the progenitors will be a binary and we already have all the information to update the four vectors above, `type_P1`, `type_P2`, `P1_marker` and `P2_marker`:

```

1 if (descendant_index != -1 and P1 == 1 and P2 == 0):
2     # P1 of descendant is a binary!
3     type_P1[descendant_index] = 2
4     P1_marker[descendant_index] = k
5 if (descendant_index != -1 and P1 == 0 and P2 == 1):
6     # P2 of descendant is a binary
7     type_P2[descendant_index] = 2
8     P2_marker[descendant_index] = k

```

- At the subsequent merger the binary that hasn't merged has updated masses as well:

```

1 q_bin = min(mass1[k], mass2[k]) / max(mass1[k], mass2[k])
2 # P1 of descendant is a binary!
3 mass1_1[descendant_index] = q_bin / (1 + q_bin) * mass1[descendant_index]
4 mass1_2[descendant_index] = 1 / (1 + q_bin) * mass1[descendant_index]
5 # P2 of descendant is a binary
6 mass2_1[descendant_index] = q_bin / (1 + q_bin) * mass2[descendant_index]
7 mass2_2[descendant_index] = 1 / (1 + q_bin) * mass2[descendant_index]

```

- On the other hand, if the binary successfully merges before the next merger, then the default values of `type_P1` and `type_P2` are good (they already represent single black holes), and `P1_marker` and `P2_marker` will not be needed.

3.3.2 Triplet mergers

In case one of the two progenitors is a binary, we have to deal with a triplet interaction. This happens when either `type_P1` or `type_P2` equals 2. Let's analyze one case. The other is analogous:

- We first recover the information on the progenitor that is still a binary, and resolve for the inner binary and the intruder information, that have to be passed to `bonetti.py` module:

```

1 if (type_P1[k]==0 and type_P2[k]==2):
2     # Triplet
3     # Binary from P1 and single BH (intruder) from P2
4     m_1 = max(mass2_1[k], mass2_2[k])
5     m_2 = min(mass2_1[k], mass2_2[k])
6     m_intr = mass1[k]

```

```

7 q_in[k] = m_2/m_1
8 q_out[k] = m_intr/(m_1 + m_2)

```

- Bonetti’s function (see paragraph 3.3.2) takes as input m_1 in M_\odot , q_in , q_out and returns a number in the range $[1, 7]$. Each value corresponds to a different outcome:
 1. If `triplet_output=1` then a prompt merger between m_1 and m_2 occurred. Then we assume that a binary forms between the merged m_1+m_2 and m_3 ;
 2. If `triplet_output=2` then m_3 has been ejected and we assume that m_1 and m_2 may undergo a delayed merger;
 3. If `triplet_output=3` then a prompt merger between m_1 and m_3 occurred. Then we assume that a binary forms between the merged m_1+m_3 and m_2 ;
 4. If `triplet_output=4` then m_2 has been ejected and we assume that m_1 and m_3 may undergo a delayed merger;
 5. If `triplet_output=5` then a prompt merger between m_2 and m_3 occurred. Then we assume that a binary forms between the merged m_2+m_3 and m_1 ;
 6. If `triplet_output=6` then m_1 has been ejected and we assume that m_2 and m_3 may undergo a delayed merger;
 7. If `triplet_output=7` then triplet interaction hasn’t been successful (see 3.3.2).

The output is passed to an `output_analyzer` function in module `triplets.py` that implements the next steps for each case:

```

1 # Launch triplet interaction
2 triplet_output = bonetti.triplet_function(m_1, q_in, q_out)
3 output = triplets.output_analyzer(...)

```

- In all the cases, the sinking time is calculated, i.e. the dynamical friction time it takes for the binary on one hand, and the intruder on the other, to sink to the center of the newly formed galaxy:

```

1 time_to_sink = delay.dynamical_friction(...)

```

- In cases 1), 3) and 5) are similar and the steps are the following:
 - We assume that the prompt merger occurs after the binary and the intruder have reached the influence distance (after `time_to_sink` has elapsed)
 - The new binary is then assigned a delay time from the hardening phase on (plus `time_to_sink`);
 - As in the binary merger case above, we proceed searching for the next descendant and analyzing whether the merger will occur before the subsequent galactic merger.
- In cases 2), 4) and 6) after the `time_to_sink` has passed, one black hole is ejected and again we remain with a binary. In this case the delay time is randomly selected between a `start_time=3 × 108` and the time to next merger:

```

1 time_to_next_merger = lookback.time_between_mergers(...)
2 time_to_merge = time_to_sink + random.uniform(start_time, time_to
    next_merger)

```

At this point the comparison between the delay time and the time to the next merger proceeds as for the binary system explained above.

bonetti.py: this module assigns probabilities to each outcome of a triple interaction and extracts a result:

```
1 input_data = ([m_1,q_out,q_in])
2 input_data = np.log10(input_data)
3
4 # Pass input data to the interpolation function for all 6 possible
  cases
5 prompt_prob12=trilinear_interp(input_data, m1, qout, qin,
  prompt_merger_frac12)
6 prompt_prob13=trilinear_interp(input_data, m1, qout, qin,
  prompt_merger_frac13)
7 prompt_prob23=trilinear_interp(input_data, m1, qout, qin,
  prompt_merger_frac23)
8 delayed_prob12=trilinear_interp(input_data, m1, qout, qin,
  delayed_merger_frac12)
9 delayed_prob13=trilinear_interp(input_data, m1, qout, qin,
  delayed_merger_frac13)
10 delayed_prob23=trilinear_interp(input_data, m1, qout, qin,
  delayed_merger_frac23)
11
12 # Create vector of outcome probabilities
13 P12=prompt_prob12/100
14 D12=P12+delayed_prob12/100
15 P13=D12+prompt_prob13/100
16 D13=P13+delayed_prob13/100
17 P23=D13+prompt_prob23/100
18 D23=P23+delayed_prob23/100
19 no_interaction=1.0
20 probability_vector=np.array([P12,D12,P13,D13,P23,D23,M0])
21
22
23 # Extract a random number from 0.0 to 1.0
24 random_num=np.random.random(1)
25
26 if (random_num[0]<=probability_vector[0]):
27     j=1 # prompt merger between m_1 and m_2
28 if (random_num[0]>probability_vector[0] and random_num[0]<=
  probability_vector[1]):
29     j=2 # delayed merger between m_1 and m_2 -> m_3 ejected
30 if (random_num[0]>probability_vector[1] and random_num[0]<=
  probability_vector[2]):
31     j=3 # prompt merger between m_1 and m_3
32 if (random_num[0]>probability_vector[2] and random_num[0]<=
  probability_vector[3]):
33     j=4 # delayed merger between m_1 and m_3 -> m_2 ejected
34 if (random_num[0]>probability_vector[3] and random_num[0]<=
  probability_vector[4]):
35     j=5 # prompt merger between m_2 and m_3
36 if (random_num[0]>probability_vector[4] and random_num[0]<=
  probability_vector[5]):
37     j=6 # delayed merger between m_2 and m_3 -> m_1 ejected
```

```

38 if (random_num[0]>probability_vector[5] and random_num[0]<=
    probability_vector[6]):
39     j=7 # still a triplet!
40
41 return j

```

Note on quadruplet systems Quadruplets happen when both the progenitors are binaries that didn't manage to merge in time, (`type_P1=2` and `type_P2=2`). We have four black holes forming two binaries. We have to find out the least massive one to eject (which could be either one of the four) and then launch the triplet interaction as explained in the section about triplets 3.3.2.

Resolve unresolved triplets In case we get the seventh option of the triplet interaction (the most probable scenario), we manually keep the two most massive black holes only and eject the third one. For what concerns the delay time, we do as follows:

- Suppose a binary forms at `t_1` with a delay time of `time_to_merge`;
- `time_to_merge` is greater than the time to next merger, `time_to_next_merger`, which happens at time `t_2`;
- At `t_2` we have a triplet system;
- After the Bonetti's interaction the triplet still remains a triplet;
- We manually discharge the least massive black hole and assign to the left over binary a delay time equal to the difference between `time_to_merge` and `time_to_next_merger` both calculated at the previous step;
- This time difference is to be compared with the time span between `t_2` and the next galactic merger.

3.3.3 A note on the lookback time

The time between mergers is calculated using:

$$\text{time_to_next_merger} = t_H \int_{z_1}^{z_2} dz' \frac{dz'}{(1+z')\sqrt{\Omega_M(1+z')^3 + \Omega_\Lambda}} \quad (2)$$

where t_H is the Hubble time³ H_0^{-1} and Ω_M and Ω_Λ are matter and dark energy densities. Here we have assumed a flat geometry universe⁴ ($\Omega_k = 0$)

We numerically integrate Eq. 2 in the interval $[z_1, z_2]$:

```

1 def lookback_function(z, omega_matter, omega_lambda):
2     return 1 / ((1 + z) * np.sqrt(omega_matter * (1 + z)**3 + omega_lambda))
3
4 def time_between_mergers(z1, z2, omega_matter, omega_lambda):
5     integral, precision = quad(lookback_function, z1, z2, args=(omega_matter,
6         omega_lambda))
7     return integral

```

³The Hubble time is the inverse of present day Hubble constant and it represents the time elapsed from the origin of the Universe till now.

⁴Otherwise, the complete function would have been:

$$E(z) = \sqrt{\Omega_M(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda}$$

4 Astrophysical background

4.1 Density profiles

4.1.1 Defining the effective radius R_{eff}

The effective radius is defined as the radius within which half of the luminosity of a galaxy is contained. For the effective radius modeling we follow the reference [12] (see Table 1). The dependencies are roughly as follows:

$$\frac{R_{\text{eff}}}{\text{kpc}} = A \left(\frac{M_{\text{stars}}}{5 \times 10^{10} M_{\odot}} \right)^{\alpha} \quad (3)$$

where A and α depend on both redshift and galaxy type. Here, in fact, the effective value distinguishes between elliptic and spiral galaxies. The galaxy type depends on the ratio between bulge and total (stellar) mass. In particular, we refer to [13]:

- If $\frac{M_{\text{bulge}}}{M_{\text{stars}}} \geq 0.7$ then we have *elliptical* galaxies
- If $0.03 \geq \frac{M_{\text{bulge}}}{M_{\text{stars}}} < 0.7$ then we have *spirals*
- If $\frac{M_{\text{bulge}}}{M_{\text{stars}}} < 0.03$ then we have pure *disks*

In our model we do distinguish between spirals and disks and discriminate around the 0.7 ratio value: if above then we have elliptical galaxies, if below we have spirals.

4.1.2 SIS model

In the singular isothermal sphere model the density profile as a function of the galaxy radius r is modeled as:

$$\rho(r) = \frac{\sigma^2}{2\pi G r^2} \quad (4)$$

with G the Newton constant and σ the velocity dispersion profile, in turn dependent on the radius:

$$\sigma(r) = \left(\frac{GM_{\text{stars}}}{4R_{\text{eff}}} \right)^{1/2} \quad (5)$$

The influence radius instead is defined as:

$$r_{\text{inf}} = R_{\text{eff}} \left(\frac{4M_{\text{binary}}}{M_{\text{stars}}} \right) \quad (6)$$

4.1.3 A case of Dehnen profiles: Hernquist model

We model the galaxy with a Dehnen density profile (see [14]):

$$\rho(r) = \frac{(3 - \gamma)M_{\text{stars}}}{4\pi} \frac{r_0}{r^{\gamma} (r + r_0)^{4-\gamma}} \quad (7)$$

where $\gamma \in [0, 3)$ and r_0 is the so called scale radius. We set $\gamma = 1$ (Hernquist profile).

The scale radius is related to the effective radius, R_{eff} , defined in 4.1.1, by the following expression:

$$r_0 = \frac{4}{3} \left(2^{\frac{1}{3-\gamma}} - 1 \right) R_{\text{eff}} \quad (8)$$

Here the influence radius is given by:

$$r_{\text{inf}} = \frac{r_0}{\left(\frac{M_{\text{stars}}}{2M_{\text{binary}}} \right)^{\frac{1}{3-\gamma}} - 1} \quad (9)$$

Given 8 and 9 it is straightforward to compute $\rho(r_{\text{inf}})$. Finally, the general expression for velocity dispersion for a Hernquist profile is:

$$\sigma^2(r) = \frac{GM_{\text{stars}}}{12r_0} \left\{ \frac{12r(r+r_0)^3}{r_0^4} \ln\left(\frac{r+r_0}{r}\right) - \frac{r}{r+r_0} \left[25 + 52\frac{r}{r_0} + 42\left(\frac{r}{r_0}\right)^2 + 12\left(\frac{r}{r_0}\right)^3 \right] \right\} \quad (10)$$

Using events from the De Lucia catalog we show in Fig. 1 the difference between SIS and Hernquist density profile, whereas in Fig. 2 we plot the density and the velocity dispersion at the influence radius for both density models.

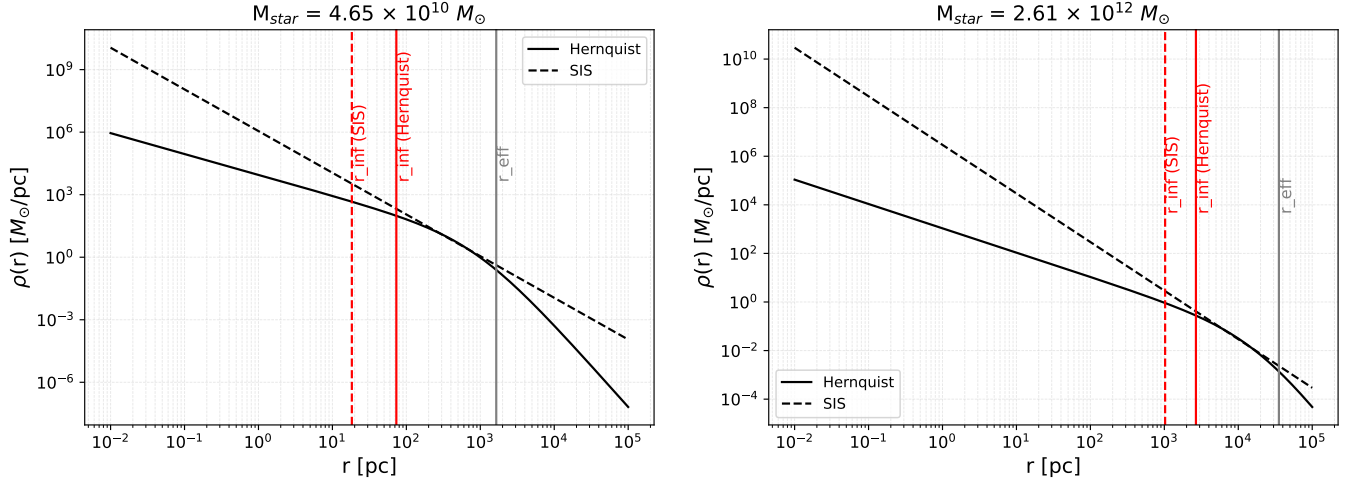


Figure 1: Two example of galaxies, one with lower mass, *on the left*, and one with higher mass, *on the right*. We show the different behavior of SIS and Hernquist density profiles: SIS model has a higher mass towards the center so that r_{inf} is smaller and this will have consequences on delay times which will be in general smaller.

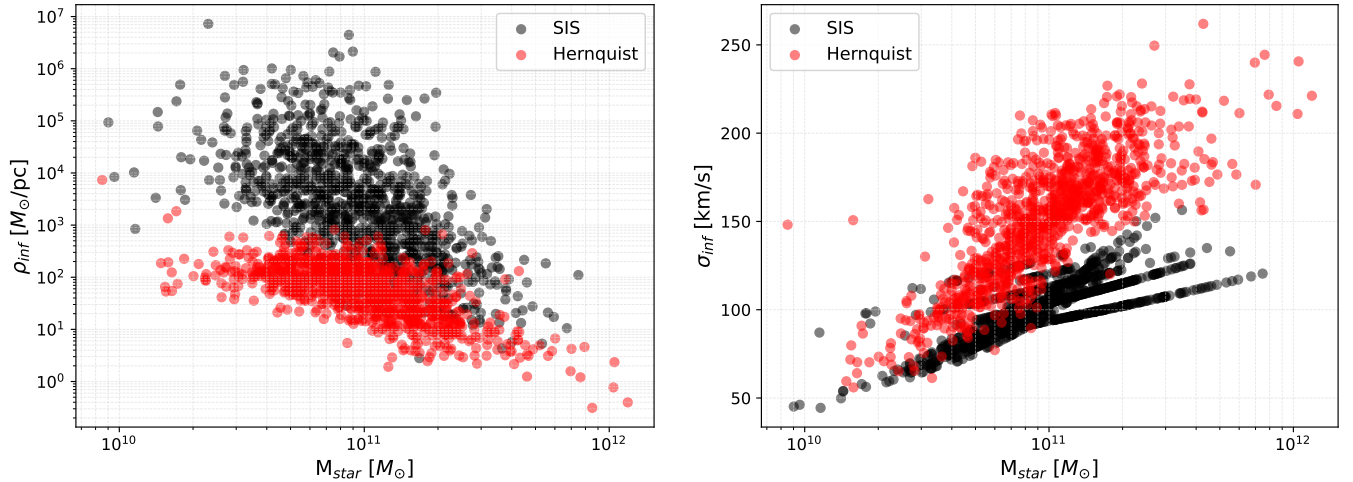


Figure 2: We show the different behavior of SIS and Hernquist density profiles plotting the density at influence radius *on the left* and the velocity dispersion at influence radius *on the right*.

4.2 Delay Times

4.2.1 Dynamical Friction Phase

For the dynamical friction timescale we follow the equations in paper [6], which we report here for convenience. This phase is divided in two.

- **Dynamical Friction: Phase I**

In the first part we calculate the time it takes the secondary⁵ black hole to decay from R_{eff} to the influence radius of the primary black hole:

$$T_{\text{df_phase1}} = \max(T_{*,1}^{\text{gal}}, T_{*,2}^{\text{gal}}), \quad (11)$$

where $T_{*,1}^{\text{gal}}$ and $T_{*,2}^{\text{gal}}$ are the timescales reported in Eqs. (54) and (56) of [6]:

$$T_{*,1}^{\text{gal}} = 0.06 \frac{2}{\ln \Lambda'} \left(\frac{R_{\text{eff}}}{10 \text{kpc}} \right)^2 \left(\frac{\sigma}{300 \text{kms}^{-1}} \right) \left(\frac{10^8 M_{\odot}}{M_{\text{bulge}}} \right) \text{Gyr}, \quad (12)$$

$$T_{*,2}^{\text{gal}} = 0.15 \frac{2}{\ln \Lambda'} \left(\frac{R_{\text{eff}}}{10 \text{kpc}} \right) \left(\frac{\sigma}{300 \text{kms}^{-1}} \right)^2 \left(\frac{100 \text{kms}^{-1}}{\sigma_s} \right)^3 \text{Gyr}, \quad (13)$$

where Λ' is the Coulomb logarithm given by:

$$\Lambda' = 2^{3/2} \frac{\sigma}{\sigma_s} \quad (14)$$

- **Dynamical Friction: Phase II**

The dynamical friction in this second phase is given by

$$T_{\text{df_phase2}} = \min(T_{\bullet}^{\text{bare}}, T_{\bullet}^{\text{gx}}). \quad (15)$$

Instead of considering Eq. 11 all the way down to the hardening radius, we stop at the influence radius and then from r_{inf} to $r = \chi r_{\text{inf}}$ we use the following ones, since once the secondary black hole enters the sphere of influence of the primary, the two black holes are bound to each other. Here we refer to Eqs. (58) and (59) of [6]:

$$T_{\bullet}^{\text{bare}} = 0.015 \text{Gyr} \frac{(\ln \Lambda \alpha + \beta + \delta)^{-1}}{\left(\frac{3}{2} - \gamma\right) (3 - \gamma)} \left(\chi^{\gamma-3/2} - 1 \right) \left(\frac{M_1}{3 \times 10^9 M_{\odot}} \right)^{1/2} \left(\frac{M_2}{10^8 M_{\odot}} \right)^{-1} \left(\frac{r_{\text{inf}}}{300 \text{pc}} \right)^{3/2} \quad (16)$$

and

$$T_{\bullet}^{\text{gx}} = 0.012 \text{Gyr} \frac{(\ln \Lambda' \alpha + \beta + \delta)^{-1}}{(3 - \gamma)^2} \left(\chi^{\gamma-3} - 1 \right) \left(\frac{M_1}{3 \times 10^9 M_{\odot}} \right) \left(\frac{100 \text{km/s}}{\sigma_s} \right)^3, \quad (17)$$

where Λ' is as in Eq. 14 and

$$\Lambda = \ln \left(\frac{b_{\text{max}}}{b_{\text{min}}} \right) \sim \ln \left(\frac{M_1}{M_2} \right). \quad (18)$$

M_1 and M_2 are the two black hole masses where $M_1 > M_2$, and χ is defined as:

$$\chi = \left(\frac{M_2}{2M_1} \right)^{\frac{1}{3-\gamma}} \quad (19)$$

⁵The distinction between primary or host and satellite galaxy/black hole relies on the mass in stars of a galaxy. The more massive galaxy is the one, the other one is the satellite galaxy.

Parameter $\gamma = 2$ for a SIS profile and $\gamma = 4$ for a Hernquist profile, and α , β and δ are three parameters respectively given by:

$$\alpha = \frac{\Gamma(\gamma + 1)}{\Gamma(\gamma - \frac{1}{2})} \frac{4}{3} \pi^{-1/2} 2^{b-\gamma} {}_2F_1\left(\frac{3}{2}; -b; \frac{5}{2}; \frac{1}{2}\right) \quad (20)$$

$$\beta = \frac{\Gamma(\gamma + 1)}{\Gamma(\gamma - \frac{1}{2})} 4\pi^{-1/2} 2^{-\gamma} \int_1^{1.4} dx x^2 (2 - x^2)^b \ln\left(\frac{x+1}{x-1}\right) \quad (21)$$

$$\delta = \frac{\Gamma(\gamma + 1)}{\Gamma(\gamma - \frac{1}{2})} 8\pi^{-1/2} \frac{2^{-\gamma-1}}{b+1} \left[0.04^{b+1} - 1\right] \quad (22)$$

where ${}_2F_1$ is an hyperbolic function and $b = \gamma - \frac{3}{2}$. These are Eqs. (21) \rightarrow (25) in [6] with $\xi = 1$, which represents the assumption of nearly circular orbits. Specializing our parameters to SIS ($\gamma = 2$) and Hernquist ($\gamma = 4$) models we obtain the following values:

– SIS ($\gamma = 2$):

$$\gamma = 2; \quad b = \frac{1}{2}; \quad \alpha = 0.5; \quad \beta \simeq 1.37; \quad \delta \simeq -0.85 \quad (23)$$

– Hernquist ($\gamma = 4$):

$$\gamma = 4; \quad b = \frac{5}{2}; \quad \alpha \simeq 0.84; \quad \beta \simeq 0.54; \quad \delta \simeq -0.29 \quad (24)$$

4.2.2 Hardening Phase

The hardening phase could either be stellar or gaseous. If a galaxy has a non zero star formation rate (SFR), then both processes are present and counts the one that makes the binary shrink faster. On the other hand, if SFR=0, then stellar hardening is the only process that precedes the GW phase.

- **Stellar hardening:**

The binary separation a varies with time according to:

$$\frac{da}{dt} = \frac{da}{dt}\Big|_{3b} + \frac{da}{dt}\Big|_{GW} = -Aa^2 - \frac{B}{a^3}, \quad (25)$$

with

$$A = \frac{GH\rho_{\text{inf}}}{\sigma_{\text{inf}}}; \quad B = \frac{64G^3M_1M_2MF(e)}{5c^5} \quad (26)$$

H is the hardening rate and we take $H = 15$ according to [7] and

$$F(e) = \frac{1}{(1 - e^2)^{7/2}} \left(1 + \frac{73}{24}e^2 + \frac{37}{96}e^4\right) \quad (27)$$

is the Peter-Matheus function.

Two phenomena contribute to orbital shrinking: encounter with single stars (3-body encounters) and emission of gravitational waves. Setting $a_{\text{stars-GW}}$ to be the separation at which the two contributions are equal, we'll have that for $a > a_{\text{stars-GW}}$ stellar hardening will prevail, and for $a < a_{\text{stars-GW}}$ till coalescence the GW contribution will take over. The equivalence radius results in:

$$a_{\text{stars-GW}} = \left(\frac{64G^2\sigma_{\text{inf}}M_1M_2MF(e)}{5c^5H\rho_{\text{inf}}}\right)^{1/5}. \quad (28)$$

Given that, the 3-body encounters will shrink the binary separation according to:

$$\frac{da}{dt}\Big|_{3b} = -\frac{GH\rho_{\text{inf}}}{\sigma_{\text{inf}}}a^2 \quad (29)$$

Integrating between $a_{\text{start}} = r_{\text{inf}}$ and $a_{\text{hard-GW}}$ we get:

$$T_{\text{stellar_hard}} = \frac{\sigma_{\text{inf}}}{GH\rho_{\text{inf}}} \left(\frac{1}{a_{\text{hard,GW}}} - \frac{1}{r_{\text{inf}}} \right) \quad (30)$$

• **Gaseous hardening:**

Gaseous hardening takes place when there is a non zero SFR. Considering a steady circumbinary disk ($\dot{m} = \text{const}$), we can derive the shrinking of the separation a over time by equating:

$$\frac{dL_{\text{disk}}}{dt} = \frac{dL_{\text{BHB}}}{dt}, \quad (31)$$

where L_{disk} is the gaseous disk angular momentum:

$$L_{\text{disk}} = m\sqrt{GM r_{\text{gap}}}, \quad (32)$$

with $r_{\text{gap}} = 2a$ and L_{BHB} is the black hole binary angular momentum:

$$L_{\text{BHB}} = \mu\sqrt{GMa}, \quad (33)$$

with μ the reduced mass ($\mu = \frac{M_1 M_2}{M_1 + M_2}$).

Developing Eq. 31 and assuming that there is no gas leaking ($\dot{\mu} = 0 = \dot{M}$):

$$\begin{aligned} \frac{dL_{\text{disk}}}{dt} &= \frac{dL_{\text{BHB}}}{dt} \\ -\dot{m}\sqrt{2}\sqrt{GMa} &= \dot{\mu}\sqrt{GMa} + \mu\frac{1}{2\sqrt{GMa}}Ga\dot{M} + \mu\frac{1}{2\sqrt{GMa}}GM\dot{a} \\ -\dot{m}\sqrt{2}\sqrt{GMa} &= \mu\frac{1}{2\sqrt{GMa}}GM\dot{a} \\ -2\sqrt{2}\dot{m} &= \mu\frac{\dot{a}}{a}. \end{aligned} \quad (34)$$

Then we obtain an expression for the time evolution of the binary separation due to interaction with the gaseous disk:

$$\left. \frac{da}{dt} \right|_{\text{gas}} = -2\sqrt{2}\frac{\dot{m}}{\mu}a. \quad (35)$$

GW emission overtakes the process when:

$$\left. \frac{da}{dt} \right|_{\text{gas}} = \left. \frac{da}{dt} \right|_{\text{GW}} \quad (36)$$

where $\left. \frac{da}{dt} \right|_{\text{GW}}$ is the same as in stellar hardening described above. Solving for $a_{\text{gas-GW}}$ we obtain:

$$a_{\text{gas-GW}} = \left(\frac{16\sqrt{2}G^3M_1^2M_2^2F(e)}{5c^5\dot{m}} \right)^{1/4}. \quad (37)$$

\dot{m} can be directly related to the SFR of the galaxy (see [15]) through:

$$\dot{m} = 10^{0.93 \log(SFR) - 2.89} \quad (38)$$

Integrating the equation 35 from $a_{\text{start}} = r_{\text{inf}}$ to $a_{\text{final}} = a_{\text{gas-GW}}$ we obtain the hardening timescale due to interaction with the gaseous disk:

$$T_{\text{gas_hard}} = \frac{\sqrt{2}\mu}{4\dot{m}} \log \left(\frac{r_{\text{inf}}}{a_{\text{gas-GW}}} \right) \quad (39)$$

4.2.3 Gravitational Waves Phase

The gravitational wave coalescence timescale is given by:

$$t_{\text{GW}} = \frac{5}{256} \frac{c^5}{G^3} \frac{a_{\text{start}}^4}{M_1 M_2 M} \frac{1}{F(e)} \quad (40)$$

where a_{start} can be either $a_{\text{star-GW}}$ or $a_{\text{gas-GW}}$, depending on which timescale is shorter. From now on GW emission prevails and a shrinks down to final coalescence.

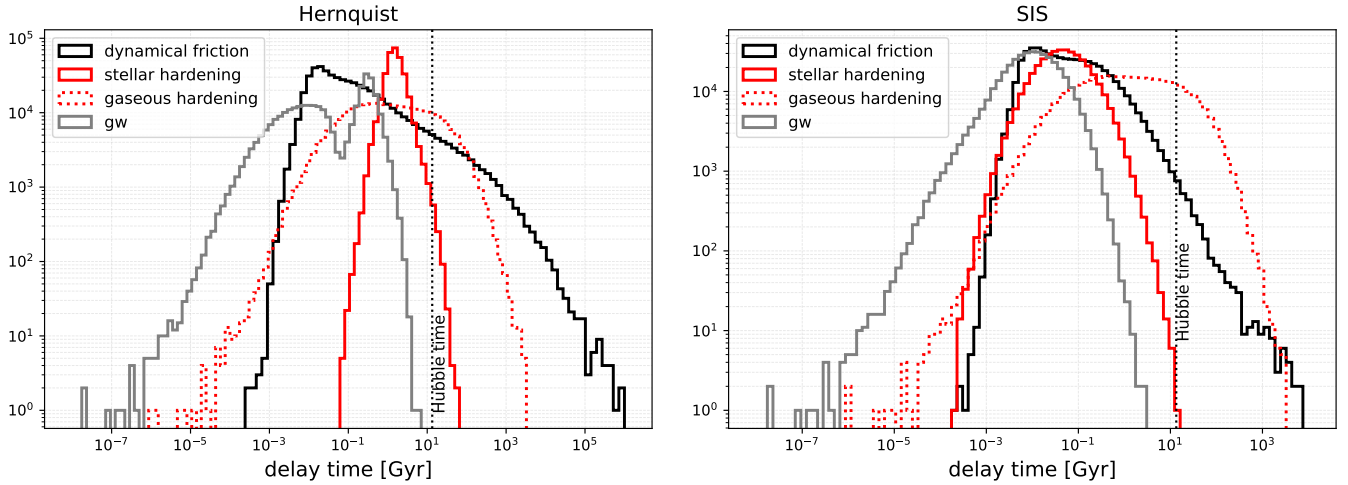


Figure 3: We show the time delay for the De Lucia catalog with Hernquist, *on the left*, and SIS, *on the right*, density profiles both with K&Ho masses. We distinguish between the different contributions to the total delay time: dynamical friction time, stellar and gaseous hardening and final GW phase. The vertical dotted line indicates the Hubble time. Observe that choosing a Hernquist profile outcomes in longer delay times.

5 Output and Analysis

5.1 Output format

The output file `output_<catalog>_<mass_model>_<density_model>.csv` contains in addition to the columns of the injection file `injection_<catalog>_<mass_model>_<density_model>.csv` 23 additional columns:

```
[ 'form_binary_vector', 'form_triplet_vector', 'form_quadruplet_vector', '
  binary_vector', 'prompt_vector', 'ejection_vector', 'forced_binary_vector',
  'failed_binary_vector', 'failed_prompt_vector', '
  failed_ejection_vector', 'failed_forced_vector', 'still_merging_vector',
  'time_to_merge', 'time_to_next_merger', 'time_df', 'time_df_ph1', '
  time_df_ph2', 'time_star', 'time_gas', 'time_gw', '
  merger_redshift_vector', 'q_in', 'q_out' ]
```

- `form_binary_vector`, `form_triplet_vector` and `form_quadruplet_vector` specify whether the initial system is a binary, a triplet or a quadruplet. They can be either 1 if the correspondence is true, zero otherwise;

- `binary_vector`, `prompt_vector`, `ejection_vector` and `forced_binary_vector` can be either 1 or 0 signifying if the corresponding successful merger is a successful binary, a triplet with a successful prompt merger, a triplet with a successful ejection outcome or unresolved triplet which forced binary system has merged in time;
- `failed_binary_vector`, `failed_ejection_vector` and `failed_forced_vector` are again either 1 or 0 meaning that the merger was not successful and it started either as a binary, as a triplet with prompt exit, a triplet with an ejection exit or an unresolved triplet;
- `still_merging_vector` indicates whether the merger at the end of the tree is still ongoing;
- `time_to_merge`, `time_to_next_merger`, `time_df`, `time_df_ph1`, `time_df_ph2`, `time_star`, `time_gas`, `time_gw`, `merger_redshift_vector` indicate, respectively, the total delay time, the time span to the subsequent galactic merger, the dynamical friction total time, the dynamical friction time divided in phase 1 and phase 2, the stellar hardening time, the gaseous hardening time, the gravitational wave time delay e the redshift at which the delayed merger occurs;
- `q_in` and `q_out` are non zero in case we have triple systems and indicate the inner and outer binary mass ratios.

5.2 Characteristic Background Strain

The gravitational wave background strain is given by the following expression:

$$h_c^2 = \frac{4}{3\pi^{1/3}} \frac{G^{5/3}}{c^2} \frac{1}{f_{1\text{yr}}^{4/3}} \int_0^\infty dz \int_0^\infty d\mathcal{M} \frac{d^2n}{dzd\mathcal{M}} \frac{\mathcal{M}^{5/3}}{(1+z)^{1/3}}, \quad (41)$$

with $f_{1\text{yr}}$ the frequency corresponding to a time span of one year and \mathcal{M} the chirp mass.

5.3 Merger Rates

The merger rate is given by:

$$\frac{dN}{dt_r} = \int dz \int d\log(\mathcal{M}) \frac{d^2N}{dzd\log(\mathcal{M})} \frac{1}{V_{\text{sim}}} 4\pi \frac{c^3}{H_0^2} \left(\int_0^z \frac{dz'}{E(z')} \right)^2, \quad (42)$$

where t_r is the rest frame time, $E(z)$ is as defined as:

$$E(z) = \sqrt{\Omega_M(1+z)^3 + \Omega_\Lambda}, \quad (43)$$

and V_{sim} is the volume of the simulation.

6 Comments and Conclusions

A Bash scripts

A.1 Get trees from root galaxies

With the following script we want to retrieve all the roots to the present day galaxies. We get a file named `all_roots.csv`. When we put a cut on the final black hole mass (which we set at $10^8 M_\odot$), we get the information on the trees' roots in the file `cut_off_roots.csv`. Both the files `all_roots.csv` and `cut_off_roots.csv` contain two columns: `galaxyId` and `lastProgenitorId`.

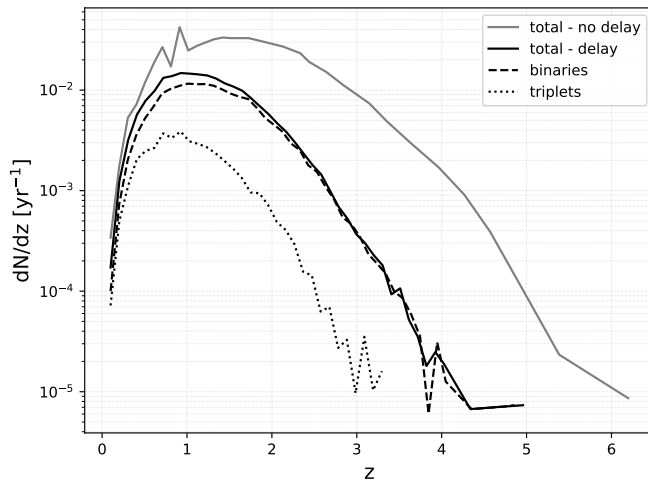


Figure 4: We show the merger rate as a function of redshift for the De Lucia catalog with SIS density profile and K&Ho masses. We distinguish between the merger rate as obtained without implementing time delay, and the one when the dynamical model is applied. Furthermore, we point out the contributions due to binaries and triplets.

```

1 #!/bin/bash
2
3 DIRNAME=PresentGalaxies
4 mkdir $DIRNAME
5
6 OUTPUTFILE="$DIRNAME/AllGalaxiesData.csv";
7
8 wget --http-user=<user_name> --http-passwd=<password> --cookies=on --keep-
  session-cookies --save-cookies=cookie.txt --load-cookies=cookie.txt -O
  $OUTPUTFILE "http://gavo.mpa-garching.mpg.de/MyMillennium?action=
  doQuery&SQL=
9 select D.galaxyId,
10        D.lastProgenitorId
11
12 from MPAGalaxies.. <galaxy_catalog_name> D
13
14 where D.snapnum %3D 63
15        # add the following line to select black holes with mass > 10^8 Msol
16        # and D.blackHoleMass %3E %3D 0.0073
17
18 order by D.galaxyId asc
19 ";
20 done

```

Once we have the two columns with the galaxyId of the last galaxy of a tree and that of the first one, lastProgenitorId, we can get the information about all the mergers in a tree, for all the considered trees:

```

1 #!/bin/bash
2
3 DIRNAME=BHMergerTrees
4 mkdir $DIRNAME
5

```

```

6 while read f1 f2;
7 do OUTPUTFILE="$DIRNAME/Data$f1.csv";
8
9 wget --http-user=<user_name> --http-passwd=<password> --cookies=on --keep-
  session-cookies --save-cookies=cookie.txt --load-cookies=cookie.txt -O
  $OUTPUTFILE "http://gavo.mpa-garching.mpg.de/MyMillennium?action=
  doQuery&SQL=
10 select D.galaxyId,
11         D.snapnum,
12         D.descendantId,
13         P1.galaxyId as P1_Id,
14         P2.galaxyId as P2_Id,
15         D.redshift as D_z,
16         D.stellarMass as D_mass,
17         D.bulgeMass as D_bulge,
18         D.sfr as sfr,
19         D.blackHoleMass as D_BH,
20         P1.redshift as P1_z,
21         P2.redshift as P2_z,
22         P1.blackHoleMass as M1,
23         P2.blackHoleMass as M2,
24         P1.bulgeMass as P1_bulge,
25         P2.bulgeMass as P2_bulge,
26         P1.stellarMass as P1_stars,
27         P2.stellarMass as P2_stars,
28         D.coldGas as M_cold,
29         D.hotGas as M_hot,
30         D.vvir as V_vir
31
32 from MPAGalaxies..DeLucia2006a D,
33      MPAGalaxies..DeLucia2006a P1,
34      MPAGalaxies..DeLucia2006a P2
35
36 where D.GalaxyId between $f1 and $f2
37        and P1.snapnum=P2.snapnum
38        and P1.galaxyId < P2.galaxyId
39        and P1.descendantId < D.galaxyId
40        and P2.descendantId < D.galaxyId
41        and P1.blackHoleMass >= 0
42        and P2.blackHoleMass >= 0
43        and P1.bulgeMass >= 0
44        and P2.bulgeMass >= 0
45        and P1.stellarMass >= 0
46        and P2.stellarMass >= 0
47
48 order by D.snapnum asc
49 ";
50 done<Data/AllGalaxiesData.csv

```

Each tree is stored in its own file, then the files are pasted together with a line of `-1s` separating different trees.

References

- [1] J. Magorrian, S. Tremaine, D. Richstone, R. Bender, G. Bower, A. Dressler, S. M. Faber, K. Gebhardt, R. Green, C. Grillmair, J. Kormendy, and T. Lauer, “The demography of massive dark objects in galaxy centers,” *The Astronomical Journal* **115** no. 6, (Jun, 1998) 2285–2305.
- [2] V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman, A. Evrard, J. Colberg, and F. Pearce, “Simulations of the formation, evolution and clustering of galaxies and quasars,” *Nature* **435** no. 7042, (Jun, 2005) 629–636.
- [3] M. C. Begelman, R. D. Blandford, and M. J. Rees, “Massive black hole binaries in active galactic nuclei,” *Nature* **287** (1980) 307–309.
- [4] S. Chandrasekhar, “Dynamical Friction. I. General Considerations: the Coefficient of Dynamical Friction,” *Astrophys. J.* **97** (1943) 255.
- [5] J. Binney and S. Tremaine, *Galactic Dynamics: Second Edition*. 2008.
<https://ui.adsabs.harvard.edu/abs/2008gady.book.....B>.
- [6] F. Dosopoulou and F. Antonini, “Dynamical friction and the evolution of Supermassive Black hole Binaries: the final hundred-parsec problem,” *Astrophys. J.* **840** no. 1, (2017) 31, arXiv:1611.06573 [astro-ph.GA].
- [7] A. Sesana and F. M. Khan, “Scattering experiments meet N-body – I. A practical recipe for the evolution of massive black hole binaries in stellar environments,” *Mon. Not. Roy. Astron. Soc.* **454** no. 1, (2015) L66–L70, arXiv:1505.02062 [astro-ph.GA].
- [8] M. Dotti, A. Merloni, and C. Montuori, “Linking the fate of massive black hole binaries to the active galactic nuclei luminosity function,” *Mon. Not. Roy. Astron. Soc.* **448** no. 4, (2015) 3603–3607, arXiv:1502.03101 [astro-ph.HE].
- [9] M. Bonetti, F. Haardt, A. Sesana, and E. Barausse, “Post-newtonian evolution of massive black hole triplets in galactic nuclei – i. numerical implementation and tests,” *Monthly Notices of the Royal Astronomical Society* **461** no. 4, (Jul, 2016) 4419–4434.
- [10] M. Bonetti, F. Haardt, A. Sesana, and E. Barausse, “Post-newtonian evolution of massive black hole triplets in galactic nuclei – II. survey of the parameter space,” *Monthly Notices of the Royal Astronomical Society* **477** no. 3, (Apr, 2018) 3910–3926.
- [11] J. Kormendy and L. C. Ho, “Coevolution (Or Not) of Supermassive Black Holes and Host Galaxies,” *Ann. Rev. Astron. Astrophys.* **51** (2013) 511–653, arXiv:1304.7762 [astro-ph.CO].
- [12] A. van der Wel *et al.*, “3D-HST+CANDELS: The Evolution of the Galaxy Size-Mass Distribution since $z = 3$,” *Astrophys. J.* **788** (2014) 28, arXiv:1404.2844 [astro-ph.GA].
- [13] D. J. Croton, V. Springel, S. D. M. White, G. De Lucia, C. S. Frenk, L. Gao, A. Jenkins, G. Kauffmann, J. F. Navarro, and N. Yoshida, “The Many lives of AGN: Cooling flows, black holes and the luminosities and colours of galaxies,” *Mon. Not. Roy. Astron. Soc.* **365** (2006) 11–28, arXiv:astro-ph/0602065. [Erratum: *Mon. Not. Roy. Astron. Soc.* **367**, 864 (2006)].
- [14] W. Dehnen, “A Family of Potential-Density Pairs for Spherical Galaxies and Bulges,” *Mon. Not. Roy. Astron. Soc.* **265** (1993) 250.
- [15] M. Volonteri, P. R. Capelo, H. Netzer, J. Bellovary, M. Dotti, and F. Governato, “Black hole accretion versus star formation rate: theory confronts observations,” 2015.
<https://ui.adsabs.harvard.edu/abs/2015MNRAS.452L...6V>.