
HBTPplus Python helpers

Release 0.1-alpha

Mar 13, 2019

Contents

1	Usage	3
2	Readers	5
3	Modules	7
3.1	hftp	7
3.2	src.cmh	11
3.3	src.mt	11
3.4	src.process	11
3.5	src.query	11
3.6	src.read	11
3.7	src.split_by_Dnf	11
	Python Module Index	13

Python module with helpers for analysis of dark matter haloes from HBtplus halo finder.

CHAPTER 1

Usage

Makefile is used to document computation of targets for any given values of gravity, snapshot, f or density profile parameters. `run.sh` and `submit.sh` demonstrate sample calculation and submission to a batch queue.

CHAPTER 2

Readers

- `HBTRReader`: general utility class, from [HBTPplus](#)
- `HBTHistoryReader`: provides an efficient collapsed mass history (CMH) calculation
- `HBTProfileReader`: provides halo and subhalo density profile reader & calculation; does **not** perform fitting - this is done in `src.process.prof()`
- `HBTEnvironmentReader`: calculates environmental proxies, as described in Haas+2012 and Shi+2017 (only $\$D_{\{N,f\}}\$$)

3.1 hbtp

HBTPReader module.

Python module containing HBTPReader class, used for analysis of HBT+ halo finder output.

Example:

```
class hbtp.ConfigReader(config_file)
```

Bases: object

Class to read the config files (i.e. Parameters.log).

```
class hbtp.HBTPReader(subhalo_path)
```

Bases: object

Class to read the HBTPPlus outputs.

To use it, initialize the reader with the directory in which Parameters.log is stored - it is written by HBT during runtime.

Parameters *subhalo_path* (*str*) – directory with config files

GetExclusiveParticles (*isnap=-1*)

Loads an exclusive set of particles for subhaloes at *isnap*

Duplicate particles are assigned to the lowest mass subhaloes.

GetFileName (*isnap*, *ifile=0*, *filetype='Sub'*)

Returns filename of an HBT snapshot

Parameters

- **isnap** (*int*) – snapshot of the file
- **ifile** (*int*) – (default=0) index for sub-snapshots
- **filetype** (*str*) – (default='Sub') 'Src', 'Sub' or 'HaloSize'

Returns HBT snaphost filename

Return type (str)

GetHostHalo (*HostHaloId*, *isnap=-1*)

Returns spatial information of a specific FoF group.

Parameters

- **HostHaloId** (*int*) – host halo id
- **isnap** (*int*) – snapshot number

Returns (numpy.ndarray) single FoF group in a structured array

GetHostMergerTree (*HostHaloId*, *isnap=-1*, *file=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)

Builds a FoF merger tree starting at a host halo ID.

Prints a graph which can be consumed by [Dot](graphviz.org) software.

Parameters

- **HostHaloId** (*int*) – a halo at which to root a tree
- **isnap** (*int*) – (default = -1) snapshot number
- **file** (*File*) – (default=sys.stdout) default output for a Dot diagram

Returns multiply embedded list with a tree, i.e. [1, [2, 3]]

Return type (list)

GetNumberOfSubhalos (*isnap=-1*)

Returns number of subhaloes in a snapshot.

Parameters **isnap** (*int*) – (default = -1) snapshot number

GetParticleProperties (*subindex*, *isnap=-1*)

Returns subhalo particle properties for subhalo with index subindex.

Values are returned in the order they appear in the file, subindex==trackId for single file output (but not for mpi multiple-file outputs)

GetProgenitorHalos (*HostHaloId*, *isnap=-1*)

Returns progenitors of a given host halo.

Given a HostHaloId and a snapshot number, it finds all subhaloes within the host, follows their track ids to the previous snapshot, checks if they are the main / central subhalo (by Rank values), finds those subhaloes' hosts and keeps the unique IDs. Filtering by Rank is necessary to remove fly-by and re-merger events.

Parameters

- **HostHaloId** (*int*) – host halo id
- **isnap** (*int*) – (default=-1) snapshot number

Returns array of host halo ids of progenitors

Return type (numpy.ndarray)

GetScaleFactor (*isnap*)

Reads scale factor at a given snapshot.

GetScaleFactorDict ()

Returns a dictionary that maps snapshot_index to ScaleFactor.

GetSub (*trackId*, *isnap=-1*)

Loads a subhalo with the given `trackId` at snapshot `isnap`.

GetSubsOfHost (*HostHaloId*, *isnap=-1*)

Loads all subhaloes belonging to a host halo.

Uses information from `Membership/GroupedTrackIds`.

Parameters

- **HostHaloId** (*int*) – row number
- **isnap** (*int*) – (default = -1)

Returns

array of TrackIds, or empty array if FoF does not host subhalos

Return type (`numpy.ndarray`)

GetTrack (*trackId*, *fields=None*, *MaxSnap=None*)

Loads an entire track of the given `trackId`.

LoadHostHalos (*isnap=-1*, *selection=None*)

Returns spatial properties of FoF groups for a snapshot.

Parameters

- **isnap** (*int*) – snapshot number
- **selection** (*tuple*) – (default = None) selection query being passed on
- **HBT file** (*to*) –

Returns (`numpy.ndarray`) host haloes matching the selection

LoadNestedSubhalos (*isnap=-1*, *selection=None*)

Load the list of nested subhalo indices for each subhalo

Parameters **isnap** (*int*) – (default = -1) snapshot number

Returns array of nested indices

Return type (`numpy.ndarray`)

LoadParticles (*isnap=-1*, *subindex=None*, *filetype='Sub'*)

Loads subhalo particle list at snapshot

If `subindex` is given, only load subhalo of the given index (the order it appears in the file, `subindex==trackId` for single file output, but not for mpi multiple-file outputs). Otherwise loads all the subhaloes.

Default filetype (`Sub`) will load subhalo particles. Filetype `Src` loads source subhalo particles instead (for debugging purpose only).

Parameters

- **isnap** (*int*) – (default=-1) snapshot number
- **subindex** (*int*) – (default=None) index of a subhalo
- **filetype** (*str*) – (default='Sub') HBT file type

LoadSubhalos (*isnap=-1*, *selection=None*, *show_progress=False*)

Load all subhaloes from a snapshot.

Note: `selection=('Rank', 'Nbound')` will load only the Rank and Nbound fields of subhaloes; `selection=3` will only load subhalo with subindex 3; default will load all fields of all subhaloes. You can also use numpy slice for selection, e.g. `selection=np.s_[:10, 'Rank', 'HostHaloId']` will select the Rank and HostHaloId of the first 10 subhaloes. You can also specify multiple subhaloes by passing a list of (ordered) subindex, e.g., `selection=((1,2,3),)`. However, currently only a single subhalo can be specified for multiple-file HBT data (not restricted for single-file data).

Note: Subindex specifies the order of the subhalo in the file at the current snapshot, i.e., `subhalo=AllSubhalo[subindex]`. `subindex == trackId` for single file output, but `subindex != trackId` for mpi multiple-file outputs.

Parameters

- **isnap** (*int*) – (default = -1) snapshot
- **selection** (*numpy.s_*) – (default = None) can be a single field, a list of the field names or a single subhalo index
- **show_progress** (*bool*) – (default = False)

OpenFile (*isnap, ifile=0, filetype='Sub', mode='r'*)

Opens HDF5 file.

Parameters

- **isnap** (*int*) – snapshot of the file
- **ifile** (*int*) – (default=0) index for sub-snapshots
- **filetype** (*str*) – (default='Sub') 'Src', 'Sub' or 'HaloSize'
- **mode** (*chr*) – (default='r') file handle mode

Returns HDF5 HBT file handle

Return type (File)

Snapshots ()

`hbtplib.PeriodicDistance` (*x, y, BoxSize, axis=-1*)

`hbtplib.distance` (*x, y, axis=-1*)

`hbtplib.get_hbt_snapnum` (*snapshotname*)

Extracts snapshot number from a filename.

Parameters **snapshotname** (*str*) – name of snapshot file

Returns snapshot number

Return type (int)

3.2 `src.cmh`

3.3 `src.mt`

`src.mt.main(grav, snap, host)`
Construct & output merger tree

Parameters

- **grav** (*str*) – Gravity (GR_b64n512 or fr6_b64n512)
- **snap** (*int*) – Snapshot number (between 122 and 10)
- **host** (*int*) – Host halo ID

3.4 `src.process`

3.5 `src.query`

`src.query.main(grav, snap, min_part_num=1000)`
Query & filter halo IDs.

Parameters

- **grav** (*str*) – Gravity (GR_b64n512 or fr6_b64n512)
- **snap** (*int*) – Snapshot number (between 122 and 9)
- **min_part_num** (*int*) – Minimum number of particles

3.6 `src.read`

`src.read.cmh(grav, snap, f=0.02)`
`src.read.ids(grav, snap, prefix='ids')`
`src.read.snaps()`

3.7 `src.split_by_Dnf`

`src.split_by_Dnf.bin(data, by, nbins)`
`src.split_by_Dnf.main(grav, snap)`
Split halo ids into 2 groups, below and above percentile range in $\log_{10}(D_{\text{Nf}})$.

Parameters

- **grav** (*str*) – Gravity (GR_b64n512 or fr6_b64n512)
- **snap** (*int*) – Snapshot number (between 122 and 9)

h

`hhttp`, [7](#)

s

`src.mt`, [11](#)

`src.query`, [11](#)

`src.read`, [11](#)

`src.split_by_Dnf`, [11](#)

B

bin() (in module src.split_by_Dnf), 11

C

cmh() (in module src.read), 11

ConfigReader (class in hbt), 7

D

distance() (in module hbt), 10

G

get_hbt_snapnum() (in module hbt), 10

GetExclusiveParticles() (hbt.HBTReader method), 7

GetFileName() (hbt.HBTReader method), 7

GetHostHalo() (hbt.HBTReader method), 8

GetHostMergerTree() (hbt.HBTReader method), 8

GetNumberOfSubhalos() (hbt.HBTReader method), 8

GetParticleProperties() (hbt.HBTReader method), 8

GetProgenitorHalo() (hbt.HBTReader method), 8

GetScaleFactor() (hbt.HBTReader method), 8

GetScaleFactorDict() (hbt.HBTReader method), 8

GetSub() (hbt.HBTReader method), 8

GetSubsOfHost() (hbt.HBTReader method), 9

GetTrack() (hbt.HBTReader method), 9

H

hbt (module), 7

HBTReader (class in hbt), 7

I

ids() (in module src.read), 11

L

LoadHostHalos() (hbt.HBTReader method), 9

LoadNestedSubhalos() (hbt.HBTReader method), 9

LoadParticles() (hbt.HBTReader method), 9

LoadSubhalos() (hbt.HBTReader method), 9

M

main() (in module src.mt), 11

main() (in module src.query), 11

main() (in module src.split_by_Dnf), 11

O

OpenFile() (hbt.HBTReader method), 10

P

PeriodicDistance() (in module hbt), 10

S

snaps() (in module src.read), 11

Snapshots() (hbt.HBTReader method), 10

src.mt (module), 11

src.query (module), 11

src.read (module), 11

src.split_by_Dnf (module), 11