Package 'snapshot'

October 14, 2022

Type Package

Title Gadget N-body cosmological simulation code snapshot I/O utilities

Version 0.1.2

Date 2013-10-04

Author Aaron Robotham

Maintainer Aaron Robotham <aaron.robotham@uwa.edu.au>

Description Functions for reading and writing Gadget N-body snapshots. The Gadget code is popular in astronomy for running N-body / hydrodynamical cosmological and merger simulations. To find out more about Gadget see the main distribution page at www.mpa-garching.mpg.de/gadget/

License GPL-2

Depends R (>= 2.13)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-10-22 16:50:41

R topics documented:

snapshot-package *Gadget N-body cosmological simulation code snapshot I/O utilities ~~ package title ~~*

Description

Functions for reading and writing Gadget N-body snapshots. The Gadget code is popular in astronomy for running N-body / hydrodynamical cosmological and merger simulations. To find out more about Gadget see the main distribution page at www.mpa-garching.mpg.de/gadget/

Details

Package:	snapshot
Type:	Package
Version:	0.1.2
Date:	2013-10-04
License:	GPL-2

Author(s)

Aaron Robotham

Maintainer: Aaron Robotham <aaron.robotham@uwa.edu.au>

Examples

```
## Not run:
temp=snapread('snapshot_XXX')
temp$part[,'x']=temp$part[,'x']+10
snapwrite(temp$part,temp$head,'snapshot_XXX_mod')

## End(Not run)
```

addhead *Add header information to particle data*

Description

Function to add required header information to a Gadget read particle dataframe. This has sensible defaults for a small galaxy merger style simulation

Usage

```
addhead(part, Npart = 2, Massarr = 0, Time = 0, z = 0, FlagSfr = 0,
FlagFeedback = 0, FlagCooling = 0, BoxSize = 0, OmegaM = 0, OmegaL = 0,
h = 1, FlagAge = 0, FlagMetals = 0, NallHW = 0, flag_entr_ics = 0)
```

Arguments

part	Strictly speaking 'part' is passed through the function, but to make this a useful object 'part' should be a data.frame containing the main particle level information. Columns required are:

ID	particle ID
x	x position in units of Mpc
y	y position in units of Mpc
z	z position in units of Mpc
vx	x velocity in units of km/s
vy	y velocity in units of km/s
vz	z velocity in units of km/s
Mass	particle mass in units of Msun

Npart	The index on the Npart vector that should contain the particle number, where: gas [1] / collisionless particles [2:6]. The actual value is calculated based on the part data.frame provided with 'part', Nall is also calculated based on this number and not given as an option since the same index as Npart must be used
Massarr	The mass of the particles in the particle index provided to Npart
Time	Time of snapshot in units of km/s and kpc so 1 unit is ~10 Gyrs
z	Redshift of snapshot
FlagSfr	Star formation turned on/off
FlagFeedback	Feedback turned on/off
FlagCooling	Cooling turned on/off
BoxSize	Size of simulation box edge length in units of kpc
OmegaM	Omega matter of the simulation
OmegaL	Omega lambda of the simulation
h	Hubble constant divided by 100 used in the simulation
FlagAge	Stellar ages on/off
FlagMetals	Stellar metallacities on/off
NallHW	Tell Gadget to use large integers in the particle index provided to Npart- not usually necessary
flag_entr_ics	Entropy for gas on/off

Details

Nall is calculated based on Npart, and therfore it cannot be specified via an input argument. This increases the likelihood that a legal Gadget header will be produced.

Value

`part` Strictly speaking 'part' is passed through the function, but to make this a useful object 'part' should be a data.frame containing the main particle level information. Assuming 'part' has been given a sensible input, columns provided are:

ID	particle ID
x	x position in units of Mpc
y	y position in units of Mpc
z	z position in units of Mpc
vx	x velocity in units of km/s
vy	y velocity in units of km/s
vz	z velocity in units of km/s
Mass	particle mass in units of Msun

`head` A list containing various header information as list elements. These are:

Npart	Vector of length 6 containing the number of particles in this snapshot file, where: gas [1] / collisionless particles [2:6]
Massarr	Vector of length 6 containing the particle masses for the respective particle types in Npart
Time	Time of snapshot in units of km/s and kpc so 1 unit is ~10 Gyrs
z	Redshift of snapshot
FlagSfr	Star formation turned on/off
Nall	Vector of length 6 containing the number of particles in all snapshot files, where: gas [1] / collisionless particles [2:6]
FlagFeedback	Feedback turned on/off
FlagCooling	Cooling turned on/off
NumFiles	Number of files per snapshot- usually 1
BoxSize	Size of simulation box edge length in units of kpc
OmegaM	Omega matter of the simulation
OmegaL	Omega lambda of the simulation
h	Hubble constant divided by 100 used in the simulation
FlagAge	Stellar ages on/off
FlagMetals	Stellar metallacities on/off
NallHW	Tell Gadget to use large integers for the respective particle types in Npart - not usually necessary
flag_entr_ics	Entropy for gas on/off

Author(s)

Aaron Robotham

See Also

`snapwrite`,`snapread`,`genparam`

Examples

```
## Not run:
tempadd=addhead(temp$part)

## End(Not run)
```

---

genparam *Generates a Gadget paramter file*

Description

Function to generator a legal Gadget paramter setup file. This has a sensible selection of defaults chosen for fairly small (non Cosmological) simulations.

Usage

```
genparam(ParamFile = "galaxy.param", ParamBase = "./HernTest/",
InitCondFile = "./HernStart.gdt", OutputDir = "./HernTest/", EnergyFile = "energy.txt",
InfoFile = "info.txt", TimingsFile = "timings.txt", CpuFile = "cpu.txt",
RestartFile = "restart", SnapshotFileBase = "snapshot",
OutputListFilename = "parameterfiles/output_list.txt", TimeLimitCPU = 36000,
ResubmitOn = 0, ResubmitCommand = "my-scriptfile", ICFormat = 1, SnapFormat = 1,
ComovingIntegrationOn = 0, TypeOfTimestepCriterion = 0, OutputListOn = 0,
PeriodicBoundariesOn = 0, TimeBegin = 0, TimeMax = 0.001, Omega0 = 0, OmegaLambda = 0,
OmegaBaryon = 0, HubbleParam = 1, BoxSize = 0, TimeBetSnapshot = 1e-05,
TimeOfFirstSnapshot = 0, CpuTimeBetRestartFile = 36000, TimeBetStatistics = 0.05,
NumFilesPerSnapshot = 1, NumFilesWrittenInParallel = 1, ErrTolIntAccuracy = 0.025,
CourantFac = 0.3, MaxSizeTimestep = 0.1, MinSizeTimestep = 0, ErrTolTheta = 0.5,
TypeOfOpeningCriterion = 1, ErrTolForceAcc = 0.005, TreeDomainUpdateFrequency = 0.1,
DesNumNgb = 32, MaxNumNgbDeviation = 8, ArtBulkViscConst = 1, InitGasTemp = 0,
MinGasTemp = 100, PartAllocFactor = 3.0, TreeAllocFactor = 4.8, BufferSize = 25,
UnitLength_in_cm = 3.085678e+21, UnitMass_in_g = 1.989e+43,
UnitVelocity_in_cm_per_s = 1e+05, GravityConstantInternal = 0,
MinGasHsmlFractional = 0.25, SofteningGas = 1e-04, SofteningHalo = 1e-04,
SofteningDisk = 0.4, SofteningBulge = 0.8, SofteningStars = 0, SofteningBndry = 0.1,
SofteningGasMaxPhys = 1e-04, SofteningHaloMaxPhys = 1e-04, SofteningDiskMaxPhys = 0.4,
SofteningBulgeMaxPhys = 0.8, SofteningStarsMaxPhys = 0, SofteningBndryMaxPhys = 0.1,
MaxRMSDisplacementFac = 0.2, NFWConcentration = 10, VirialMass = 200, FlatRadius = 1e-05,
DeltaVir = 200, addNFW = FALSE)
```

Arguments

ParamFile	Name for the paramter file
ParamBase	Base file path for the paramter file
InitCondFile	Full path of file containing initial conditions

`OutputDir`	Base directory in which to put the major Gadget outputs, including snapshots etc
`EnergyFile`	Name to give energy file
`InfoFile`	Name to give info file
`TimingsFile`	Name to give timings file
`CpuFile`	Name to give CPU file
`RestartFile`	Name to give restart file
`SnapshotFileBase`	Base name for snapshots, appended by snapshot number
`OutputListFilename`	Name of file containing output times / expansion factors
`TimeLimitCPU`	Max CPU time to use for Gadget run
`ResubmitOn`	Flag to tell super-computer there is a resubmit file
`ResubmitCommand`	Specific to super-computer resubmit command
`ICFormat`	Initial conditions format: PUT OPTIONS IN TABLE HERE
`SnapFormat`	Snapshot format: PUT OPTIONS IN TABLE HERE
`ComovingIntegrationOn`	Allow for expansion of Universe
`TypeOfTimestepCriterion`	Type of particle integrator- leave at 0
`OutputListOn`	Flag to tell it to use OutputListFilename as input
`PeriodicBoundariesOn`	Flag to turn on/off periodic box boundaries, only needed for large cosmological runs
`TimeBegin`	Time at the beginning of simulation
`TimeMax`	Max time to evolve particles to
`Omega0`	Total energy density
`OmegaLambda`	Cosmological constant energy density
`OmegaBaryon`	Baryonic energy density
`HubbleParam`	Value of H0/100 to be used
`BoxSize`	Length of box edge (important for cosmological runs only)
`TimeBetSnapshot`	Time between snapshots
`TimeOfFirstSnapshot`	Time at which to output first snapshot
`CpuTimeBetRestartFile`	How often to output full restart file
`TimeBetStatistics`	Time between energy.txt updates

`NumFilesPerSnapshot`
: How many files to split snapshots over

`NumFilesWrittenInParallel`
: How many files to split snapshots over (probably ignore)

`ErrTolIntAccuracy`
: Orbital integration accuracy

`CourantFac`
: Limit on time step compared to sound crossing time for hydro runs

`MaxSizeTimestep`
: Maximum time step allowed

`MinSizeTimestep`
: Minimum time step allowed

`ErrTolTheta`
: Controls the accurary of integration (smaller is closer to direct N-body)

`TypeOfOpeningCriterion`
: Barnes-Hut or modified opening criteria (probably ignore)

`ErrTolForceAcc`
: Only used for modified opening criterion (use default)

`TreeDomainUpdateFrequency`
: How often should a tree be constructed

`DesNumNgb`
: Number of neighbours to use for denisty estimation in SPH

`MaxNumNgbDeviation`
: How much tolerance is allowed when finding neighbours

`ArtBulkViscConst`
: Artificial viscosity term (use default)

`InitGasTemp`
: Initial gas temperature

`MinGasTemp`
: Minimum gas temperature allowed in the run

`PartAllocFactor`
: Memory buffer per particle per processor

`TreeAllocFactor`
: Memory buffer for tree calculation

`BufferSize`
: Total memory buffer between processors

`UnitLength_in_cm`
: Assumed IC distance units in cm (default assumes Kpc for input)

`UnitMass_in_g`
: Assumed mass of provided IC mass units in grams (default assumes 1e10 Msun for input)

`UnitVelocity_in_cm_per_s`
: Assumed velocity of provided units in cm/s (default assumes km/s)

`GravityConstantInternal`
: Internal units for g

`MinGasHsmlFractional`
: Minimum multiplicitive factor for smoothing length in hyrdo gas

`SofteningGas`
: Softening to use for gas particles

`SofteningHalo`
: Softening to use for halo particles

`SofteningDisk`
: Softening to use for disk particles

`SofteningBulge` Softening to use for bulge particles

`SofteningStars` Softening to use for star particles

`SofteningBndry` Softening to use for boundary particles

`SofteningGasMaxPhys`
Physical softening to use for gas particles (only relevant for Cosmo run)

`SofteningHaloMaxPhys`
Physical softening to use for halo particles (only relevant for Cosmo run)

`SofteningDiskMaxPhys`
Physical softening to use for disk particles (only relevant for Cosmo run)

`SofteningBulgeMaxPhys`
Physical softening to use for bulge particles (only relevant for Cosmo run)

`SofteningStarsMaxPhys`
Physical softening to use for star particles (only relevant for Cosmo run)

`SofteningBndryMaxPhys`
Physical softening to use for boundary particles (only relevant for Cosmo run)

`MaxRMSDisplacementFac`
Biggest distance that a particle can move in a time step

`NFWConcentration`
Concentration of analytic NFW profile, addNFW must be set to TRUE

`VirialMass` Mass within virial radius of analytic NFW profile, addNFW must be set to TRUE

`FlatRadius` Forces the NFW profile to be cored (not cusped), addNFW must be set to TRUE

`DeltaVir` Virial overdensity of NFW profile, addNFW must be set to TRUE

`addNFW` Logic determining whether the analyic NFW specific paramters be added to the setup file? See above

Value

No value returned, called for the side-effect of writing out a Gadget paramter setup file.

Author(s)

Aaron Robotham

See Also

`snapwrite`,`snapread`,`addhead`

Examples

```
## Not run:
genparam('example.param','Demo/Example1/')

## End(Not run)
```

snapread *Read in Gadget snapshots*

Description

This function allows the user to read in the standard format Gadget binaries. It keeps the particle information and header information in separate components of a list.

Usage

```
snapread(file)
```

Arguments

file	The full path to the Gadget snapshot to be read in.

Value

part — A data.frame containing the main particle level information. Columns included are:

ID	particle ID
x	x position in units of Mpc
y	y position in units of Mpc
z	z position in units of Mpc
vx	x velocity
vy	y velocity
vz	z velocity
Mass	particle mass in units of Msun

head — A list containing various header information as list elements. These are:

Npart	Vector of length 6 containing the number of particles in this snapshot file, where: gas [1] / collisionless particles [2:6]
Massarr	Vector of length 6 containing the particle masses for the respective particle types in Npart
Time	Time of snapshot in units of km/s and kpc so 1 unit is ~10 Gyrs
z	Redshift of snapshot
FlagSfr	Star formation turned on/off
Nall	Vector of length 6 containing the number of particles in all snapshot files, where: gas [1] / collisionless particles [2:6]
FlagFeedback	Feedback turned on/off
FlagCooling	Cooling turned on/off
NumFiles	Number of files per snapshot- usually 1
BoxSize	Size of simulation box edge length in units of kpc
OmegaM	Omega matter of the simulation

OmegaL	Omega lambda of the simulation
h	Hubble constant divided by 100 used in the simulation
FlagAge	Stellar ages on/off
FlagMetals	Stellar metallacities on/off
NallHW	Tell Gadget to use large integers for the respective particle types in Npart - not usually necessary
flag_entr_ics	Entropy for gas on/off

Author(s)

Aaron Robotham

See Also

snapwrite,addhead,genparam

Examples

```
## Not run:
temp=snapread('somepath/snapshot_XXX')

## End(Not run)
```

`snapwrite` *Write in Gadget snapshots*

Description

This function allows the user to write standard format Gadget binaries. It can write the particle information and header information, which are provided as separate R objects.

Usage

```
snapwrite(part, head, file)
```

Arguments

`part` A data.frame containing the main particle level information. Columns required are:

ID	particle ID
x	x position in units of Mpc
y	y position in units of Mpc
z	z position in units of Mpc
vx	x velocity in units of km/s
vy	y velocity in units of km/s
vz	z velocity in units of km/s
Mass	particle mass in units of Msun

`head` A list containing various header information as list elements. These are:

Npart	Vector of length 6 containing the number of particles in this snapshot file, where: gas [1] / collisionless particles [2:6]
Massarr	Vector of length 6 containing the particle masses for the respective particle types in Npart
Time	Time of snapshot in units of km/s and kpc so 1 unit is ~10 Gyrs
z	Redshift of snapshot
FlagSfr	Star formation turned on/off
Nall	Vector of length 6 containing the number of particles in all snapshot files, where: gas [1] / collisionless particles [2:6]
FlagFeedback	Feedback turned on/off
FlagCooling	Cooling turned on/off
NumFiles	Number of files per snapshot- usually 1
BoxSize	Size of simulation box edge length in units of kpc
OmegaM	Omega matter of the simulation
OmegaL	Omega lambda of the simulation
h	Hubble constant divided by 100 used in the simulation
FlagAge	Stellar ages on/off
FlagMetals	Stellar metallacities on/off
NallHW	Tell Gadget to use large integers for the respective particle types in Npart - not usually necessary
flag_entr_ics	Entropy for gas on/off

`file` The full path to the Gadget snapshot to be created.

Value

No value returned, called for the side-effect of writing out a binary Gadget file.

Author(s)

Aaron Robotham

See Also

snapread,addhead,genparam

Examples

```
## Not run:
temp=snapwrite(snap$part,snap$head,'somepath/snapshot_XXX')

## End(Not run)
```

Index