

Fiber-Assignment Pipeline Documentation

Lado Samushia

March 30, 2015

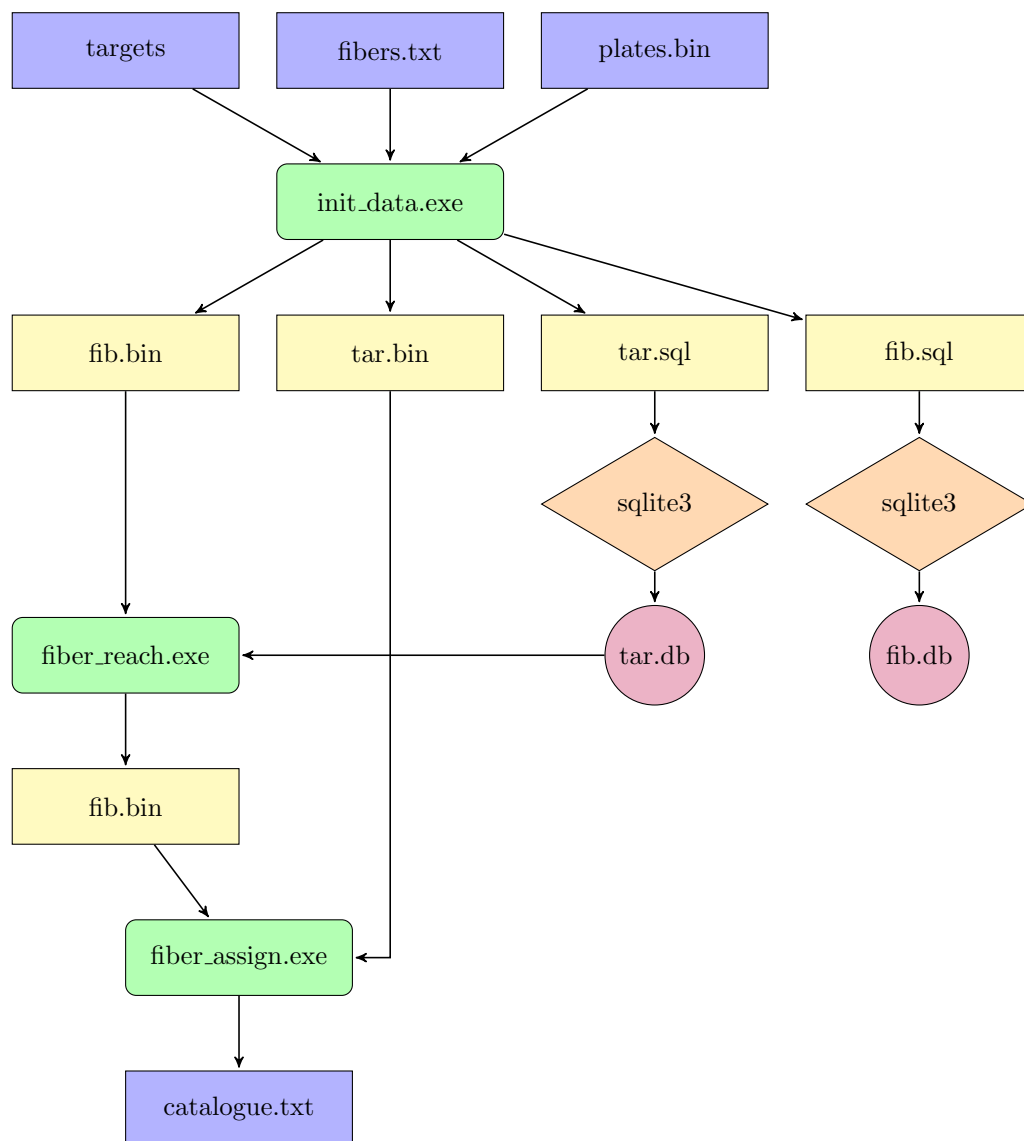


Figure 1: Flow of target assignment pipeline. Rectangles denote files, rounded rectangles – executables, diamond – third party software, circle – databases.

Introduction

This is a brief documentation of the fiber assignment code.
The code takes as an input three files:

1. The file with information about targets (targets.dat)
2. The file with information about fiber positions wrt plate centers (fibers.dat)
3. The file with information about angular positions of plate centers (targets.dat)

The code ultimately outputs one file

1. The file with positions of all targets with observed redshifts (catalogue.txt)

(To be used by a correlation function computing code later).

Input Format

1) targets.dat – this is a binary file that has
(int)N - Number of targets in the file
(float)ra[N] - right ascension of targets (degrees)
(float)dec[N] - declination of targets (degrees)
(float)red[N] - redshift of targets
(float)type[N] - type of targets (1=QSOa, 2=QSO, 3=LRG, 4=ELG, 5=badQSO, 6=badLRG)

This format is due to legacy reasons (Martin White, who created Nbody catalogues stores his data in this format). We are planning on switching the input to a binary FITS table of some sort.

2) fibers.dat – this is a text file, the format of a file is given by a file ”../etc/fiberpos.txt”.

3) plates.dat – This is a binary file that has
(int)N - Number of plates in the file
(float)ra - right ascension of plates (degrees)
(float)dec - declination of plates (degrees)
The is the format in which David Schlegel stores info

Output Format

1) catalogue.txt – for now this is an ASCII file with four columns that are
i) ra of observed target
ii) dec of observed target
iii) redshift of observed target

iv) type of observed target

This format is temporary and will most likely change to a binary FITS table of some sort. It's also likely that we will want to propagate more information forwards to the correlation function computing code.

We will also need an output file that will list which fibers are assigned to which targets. I will write that as soon as we agree on the format.

Auxiliary Files

tar.sql: an sqlite3 script file. This file will be fed to sqlite3
sqlite3 < tar.sql
to create a database of targets (tar.db)

fib.sql: an sqlite3 script file. This file will be fed to sqlite3
sqlite3 < fib.sql
to create a database of fibers (fib.db)

tar.db: An sqlite3 database file storing 3D positions of all targets (on unit sphere). Will be used for quick range searches.

fib.db: An sqlite3 database file storing 3D positions of all fibers (on unit sphere). Current version of the code doesn't use this database, but maybe useful for something later.

tar.dat: A binary file storing all the relevant information about targets for fiber assignment. (For the format see tg.h).

fib.dat: A binary file storing all the relevant information about fibers for fiber assignment. (For the format see fa.h).

The Fiber Assignment Process

There are three steps to generating the catalogue of observed targets

- 1) init_data: read all the input data and reformat it for later use
- 1.5) sqlite3: create an sqlite3 database of target and fiber positions in 3D
- 2) fiber_reach: find all targets that are reachable by all fibers
- 3) fiber_assign: assign a specific target to each fiber based on a fiber assignment priorities

init_data.exe: This takes < 10 minutes on a single CPU. Creates files fib.sql and tar.sql that will be used by sqlite3 later.

input: targets.dat, fibers.dat, plates.dat

output: fib.dat, tar.dat, fib.sql, tar.sql

sqlite3: This takes 30 minutes on a single CPU. Creates databases fib.db and tar.db that can be read and re-used as required later.

input: fib.sql, tar.sql

output: fib.db, tar.db

fiber_reach: This takes 2-3 hours on a single CPU (in principle easily parallelizable). Will find all targets reachable by each fiber. The information is stored in fib.dat and can be re-used later.

input: fib.dat, tar.db

output: fib.dat

fiber_assign: This is the actual fiber assignment. Takes ; 5 minutes on a single CPU.

input: fib.dat, tar.dat

output: catalogue.txt

The nice thing about the pipeline is that the database of targets has to be created only once (The targets don't move), and the database of fibers also has to be created only once for each tiling. Once these databases are in place and the fiber_reach.exe is run (3 hours on a single CPU), the actual fiber assignment is really fast (few minutes) and can be rerun as many times as needed with different options.

If some information about targets changes (we thought it was an LRG, turns out it's not) this can be updated instantaneously in the tar.dat file without a need to rebuild the database.

Compiling

```
$make all
```

Running

Assuming you are in the main directory:

```
$/bin/init_data.exe targets.dat fibers.dat plates.dat tar.sql fib.sql tar.dat fib.dat
```

```
$sqlite3 < tar.sql
```

```
$sqlite3 < fib.sql
```

```
$/bin/fiber_reach.exe fib.db fib.dat
```

```
$/bin/fiber_assign.exe fib.dat tar.dat catalogue.txt
```

To Do

1. Once we figure out the format of all input and output files implement them.
2. The mapping from fiber x, y, z to angular coordinates is very likely not accurate.