

Note on handling target duplication in TargetDB

Duplication identification

- Duplicate objects are identified by using the [agglomerative clustering algorithm](#) for which one can set a maximum cluster size. The maximum cluster size may be a fraction of typical fiber diameter, e.g., 0.3 arcsec. The exact number should be determined based on the feedback from the commissioning process and/or consideration of maximum allowed flux loss.

Proposed schema for a table (say, `cluster` table) containing information on duplicates

Currently, I'm considering to list clusters with more than one members.

column	description
target_id	target_id from the target table
cluster_id	Unique ID of the cluster
ra_cluster	RA of the cluster as a mean of objects in the cluster
dec_cluster	Dec of the cluster as a mean of objects in the cluster
d_ra	Distance between ra_object and ra_cluster
d_dec	Distance between dec_object and dec_cluster
n_object	Number of objects in the cluster
input_catalog_id	Input catalog ID (maybe run ID or equivalent)
created_at	timestamp
updated_at	timestamp

Clusters as a target

I propose to put clusters as new targets in the `target` table. In order to distinguish singular objects and clusters, a flag, e.g., `flag_cluster` must be introduced (or maybe we can introduce an entry in the `input_catalog` table. Other columns also need to be thought. Exposure times and observing conditions, we may take a better or conservative ones if they are not too different.

Connection to pointing planner

1. Query the `target` table to obtain a list of objects.
2. Look at `flag_cluster` and remove individual objects in these clusters from the list.
3. Run the fiber allocation process to the list.

Example case

target table

Suppose you have targets as follows.

<code>target_id</code>	<code>x</code>	<code>y</code>	<code>is_cluster</code>
0	10	10	False
1	10	10	False
2	20	20	False
3	30	30	False
4	20	20	False
5	40	40	False

Clustering identification

After running a clustering algorithm, the `cluster` table should look like as follows.

<code>target_id</code>	<code>cluster_id</code>	<code>n_object</code>	<code>x_cluster</code>	<code>y_cluster</code>
0	0	2	10	10
1	0	2	10	10

target_id	cluster_id	n_object	x_cluster	y_cluster
2	1	2	20	20
4	1	2	20	20

Insert clusters to target table

Then the clusters are inserted into target table with `flag_cluster=True` .

target_id	x	y	flag_cluster
0	10	10	False
1	10	10	False
2	20	20	False
3	30	30	False
4	20	20	False
5	40	40	False
6	10	10	True
7	20	20	True

At the fiber allocation process

First, querying targets will return objects including singular and cluster objects.

target_id	x	y	flag_cluster
0	10	10	False
1	10	10	False
2	20	20	False
3	30	30	False
4	20	20	False
5	40	40	False

target_id	x	y	flag_cluster
6	10	10	True
7	20	20	True

Then, look at `flag_cluster` and remove corresponding objects from the list by looking at the `cluster` table. The result should look like the following.

target_id	x	y	flag_cluster
3	30	30	False
5	40	40	False
6	10	10	True
7	20	20	True

Fiber allocation is carried out by using this list and propagated to downstream. When detailed information of cluster objects, one needs to search the `targetDB`.