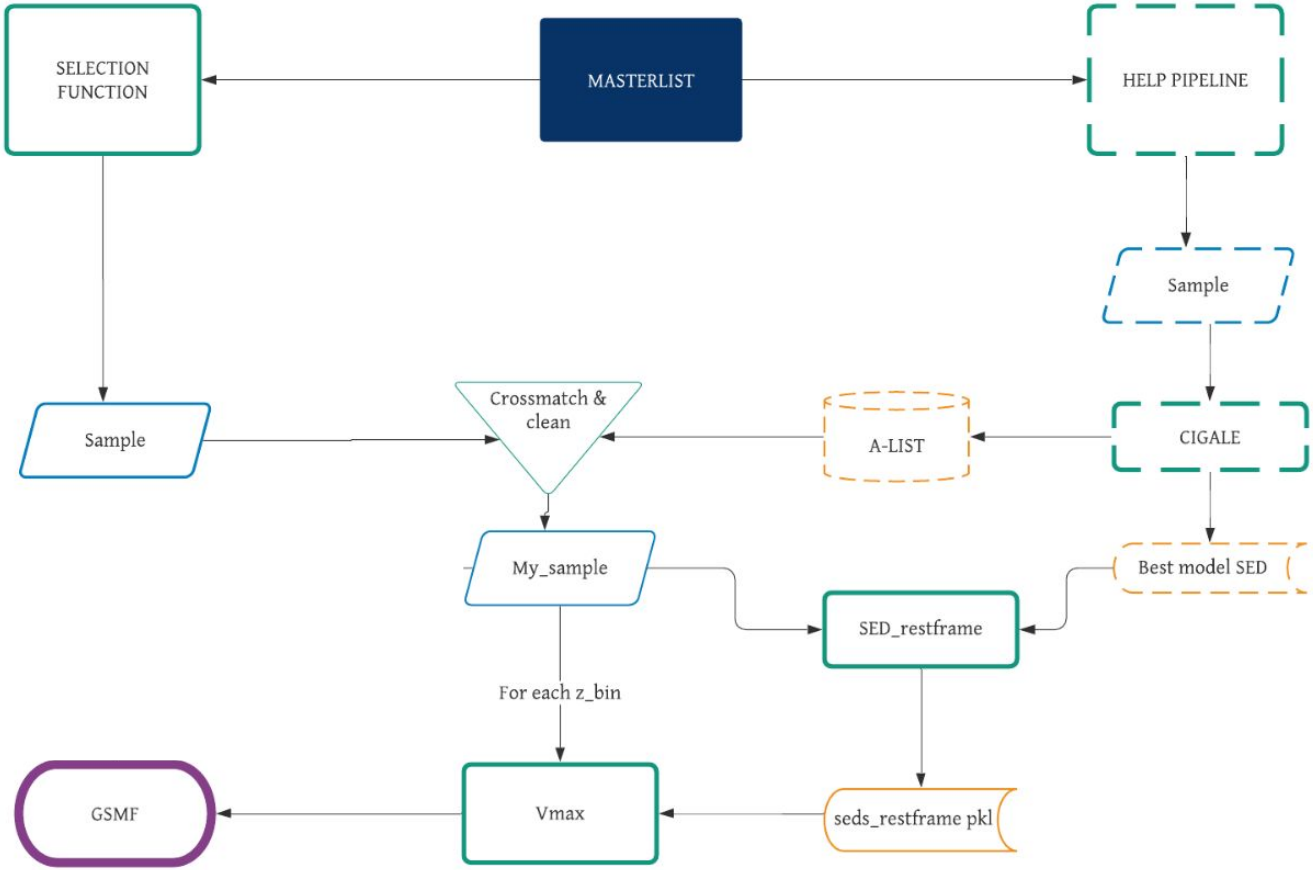
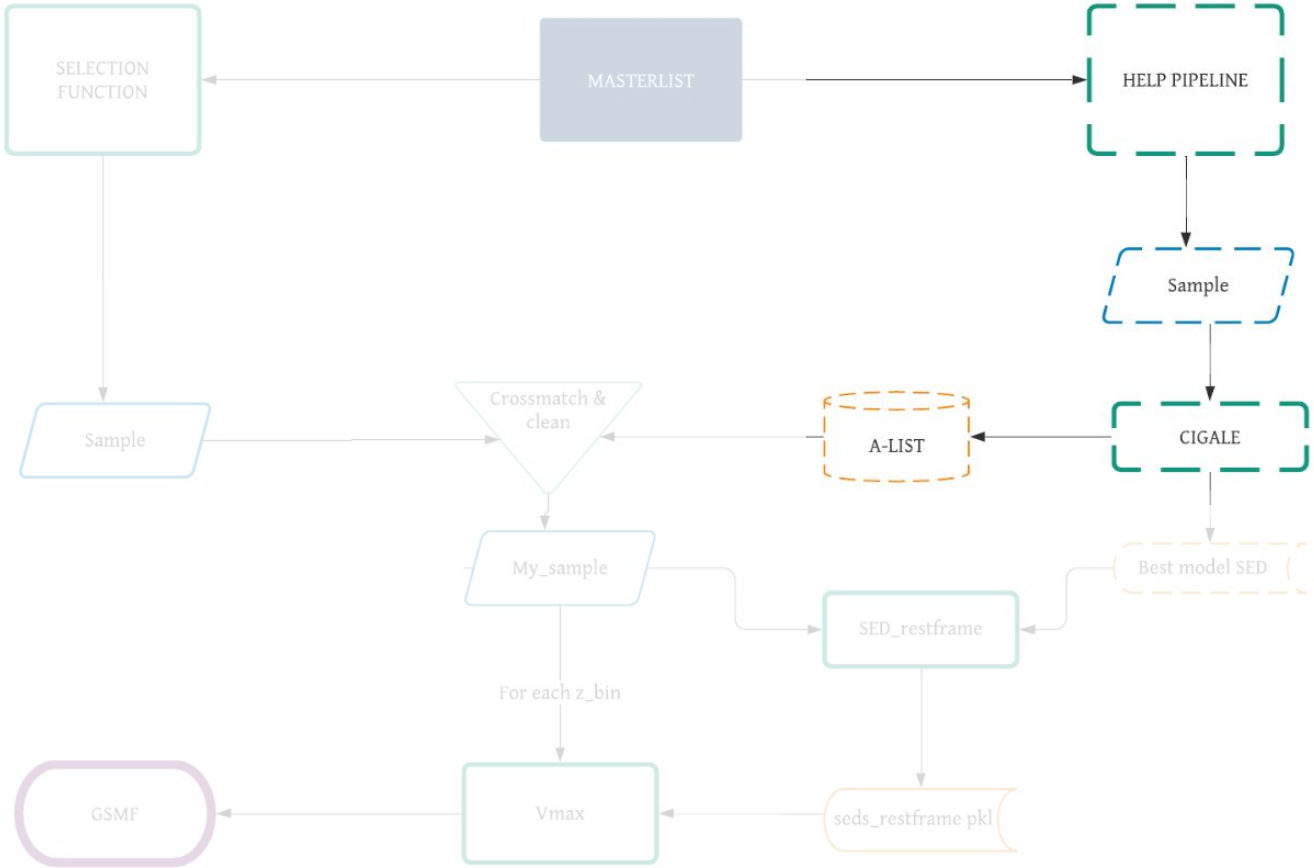
WORKFLOW GSMF



WORKFLOW GSMF



HELP pipeline

MASTERLIST

OPTICAL	NIR	MID
g, r, i, z, y u, g,r, i, z, y g, r, i, z, y g, r, i, z, y g,, r, i, z, y	y, j, h, ks y, j, h, k j, ks	irac i1-i4

PHOTO-Z — 4 detections **AND** 1 should be ‘r’ band

CIGALE (Ldust predictions) — 2 optical **AND** 2 NIR detections

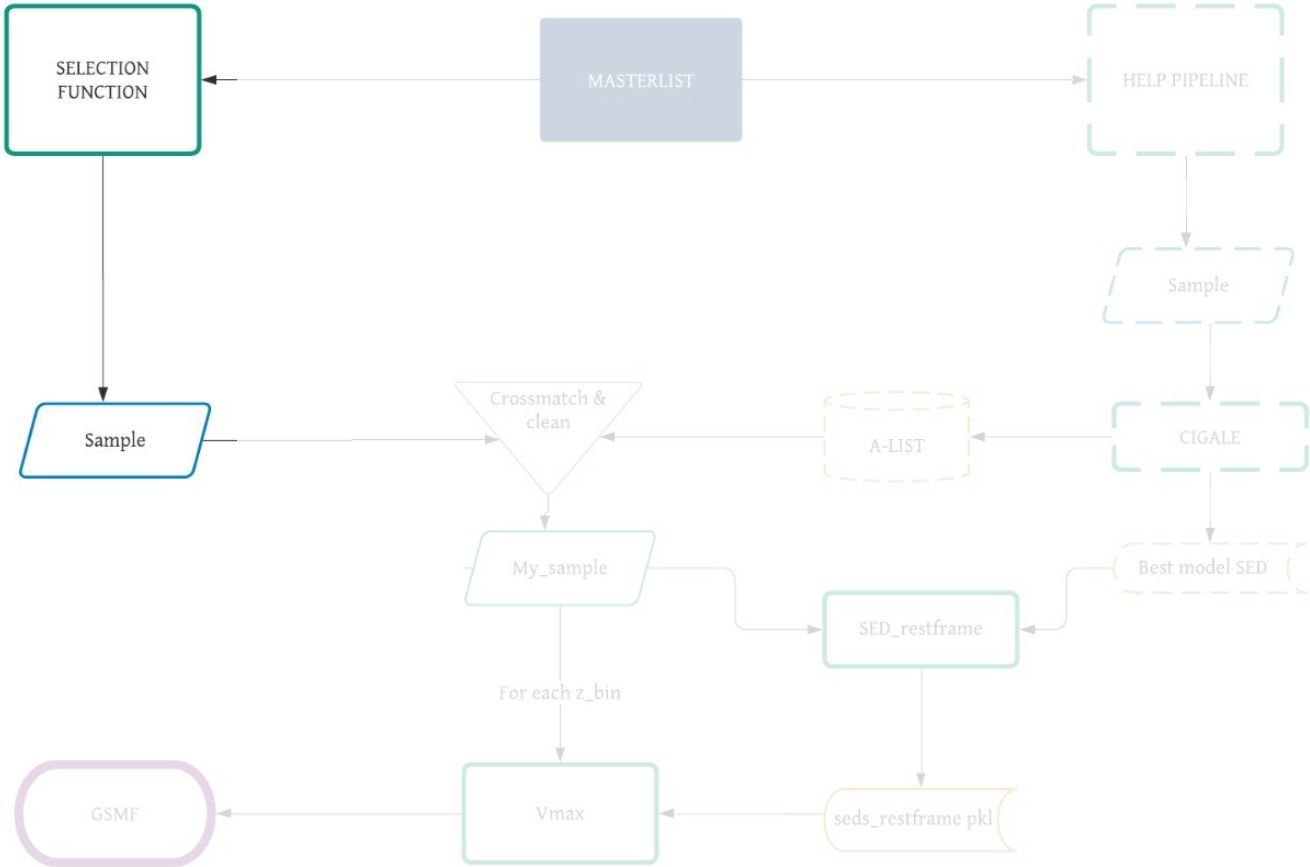
XID+ — $f_{250\_pred} > f_{threshold}$ — 2-OPT **OR** 2-NIR **AND** 2-MID detections

FIR

CIGALE (Ldust / SFR/ $M_{\star}$) — 2-FIR S/N > 2 **AND** No flag on XID+

A-LIST

WORKFLOW GSMF



SELECTION FUNCTION

```
def select_sample(catalogue, allbands, mocs_list, mcuts, ndet=2):
    """
    param.
    ---------------
    catalogue: astropy.table
        HELP catalogue
    allbands: [[..., str]]
        Names of bands used in the selections: [[opt_bands], [nir_bands], [mir_bands]]
    mocs_list: : [..., str]
        list containing the mocs fits files.
    mcuts: dict
        magnitude cuts limits for each band.
    ndet: int
        number of detections required per group of bands (opt/nir/mid). Default 2; it requires 2 detection on each to pass the selection.

    output
    ---------------
    nb_band: astropy.Table
        Table with flags for each source indicating whether the source has at least "ndet" detections above the magnitude cut.
        Flags: optband=1, nirband=2, mirband=4, ndet_total = (optband + nirband + mirband)
    """
```

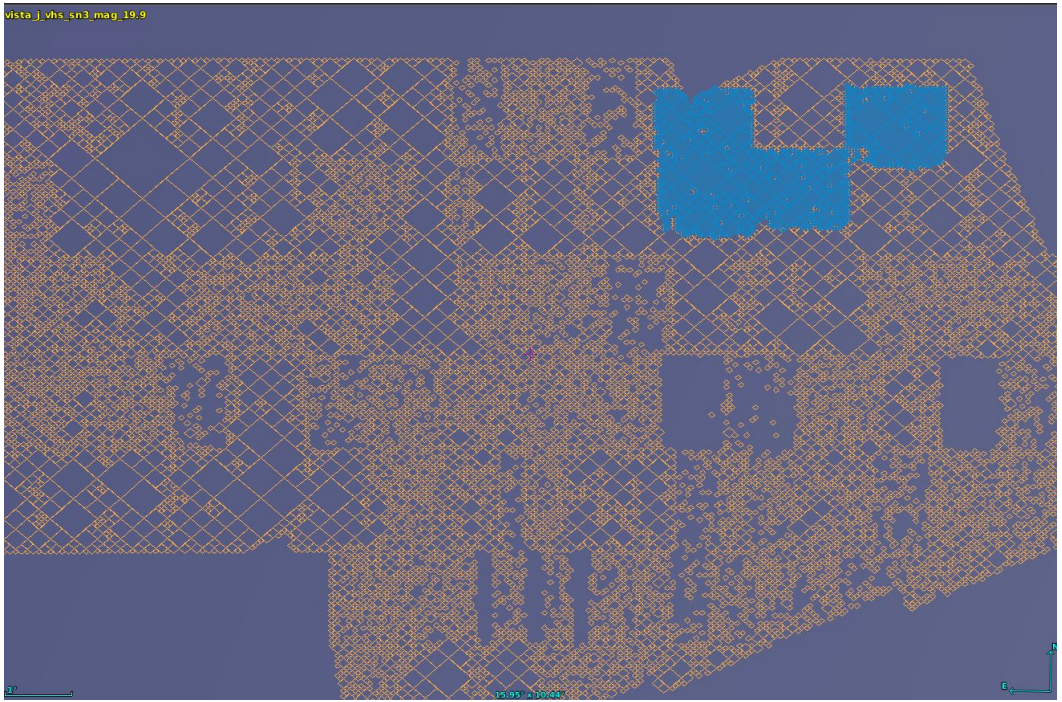
Selection Function

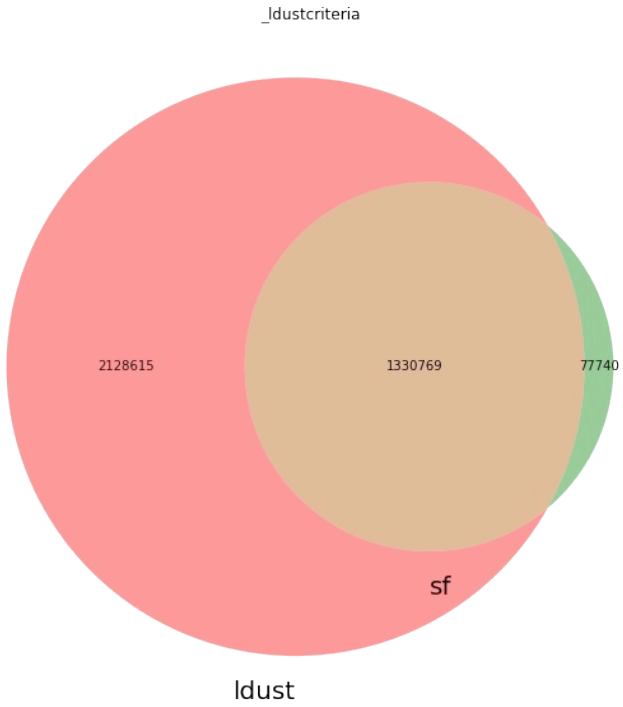
- **Define s/n and magnitude cuts** → Based on pristine catalogues (dmu0) Sigma_Flux_cuts

```
# SDSS
magcuts_sdss = {'sdss_u': 22.8,
 'sdss_g': 23.7,
 'sdss_r': 23.5,
 'sdss_i': 22.9,
 'sdss_z': 21.5}

# HSC
magcuts_hsc = {
    'suprime_g': 25.4,
    'suprime_r': 25.4,
    'suprime_i': 25.1,
    'suprime_z': 24.2,
    'suprime_y': 23.7}

# DECam: DECaLS + DES
magcuts_des = {'decam_g_des': 23.3,
 'decam_r_des': 23.1,
 'decam_i_des': 22.7,
 'decam_z_des': 21.8,
 'decam_y_des': 20.6}

magcuts_decals =  {'decam_g_decals': 23.9, 'decam_r_decals': 23.5, 'decam_z_decals': 22.5}

# VISTA: VHS + VICS82
magcuts_vhs = {'vista_y_vhs': 20.1,
 'vista_h_vhs': 19.6,
 'vista_j_vhs': 19.9,
 'vista_ks_vhs': 19.6}

magcuts_vics = {'vista_j_vics': 21.0,
 'vista_ks_vics': 21.0}

# UKIDSS - LAS
magcuts_ukidss = {'ukidss_h':19.0, 'ukidss_j':19.0, 'ukidss_k':19.0, 'ukidss_y':20.2}
```

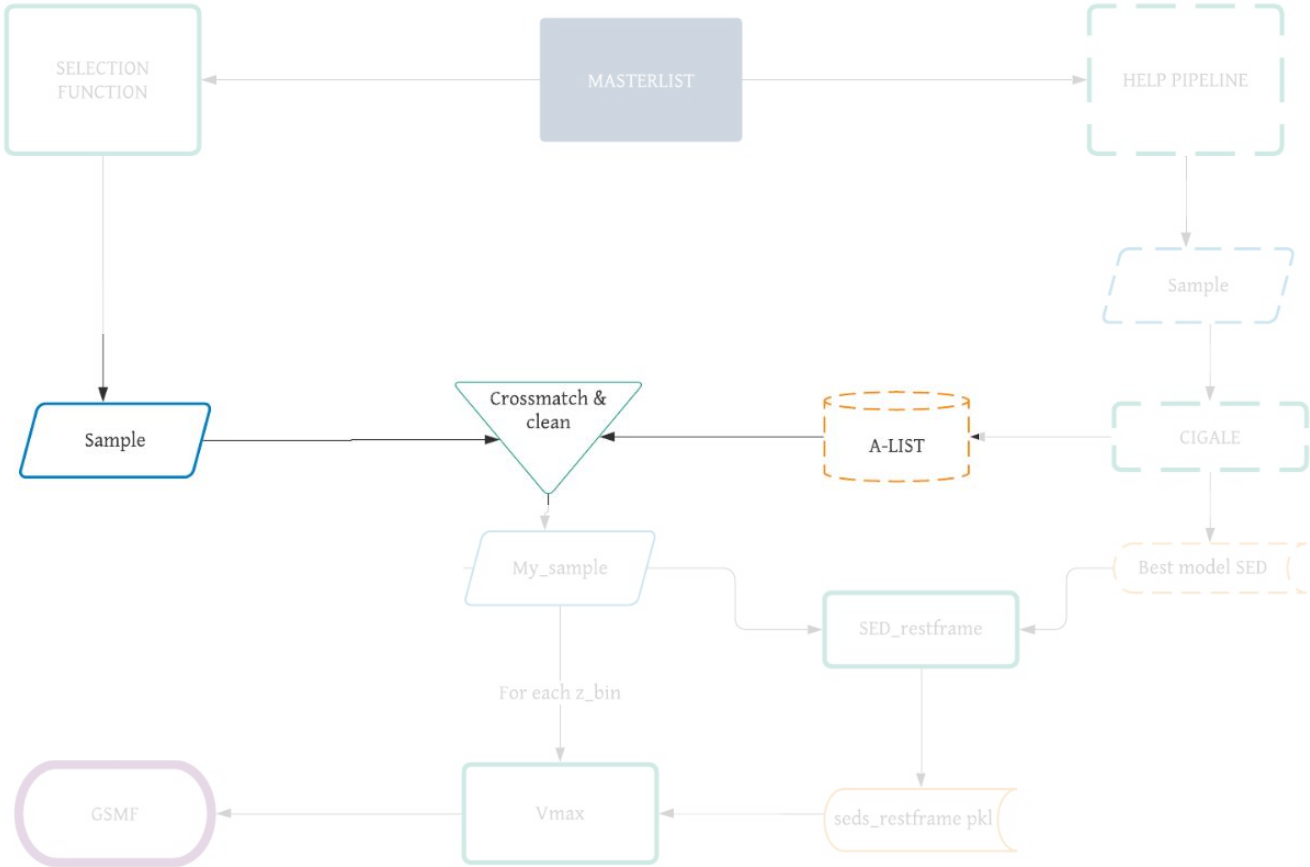
Selection Function

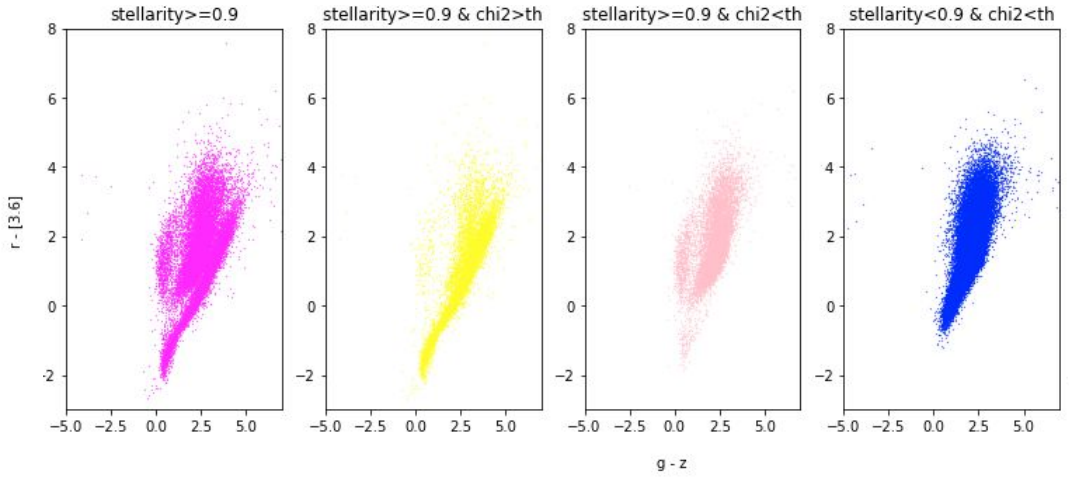
- **Define s/n and magnitude cuts** → Based on pristine catalogues (dmu0) Sigma_Flux_cuts
- **Define MOCs** → Where a source can be detected above that s/n

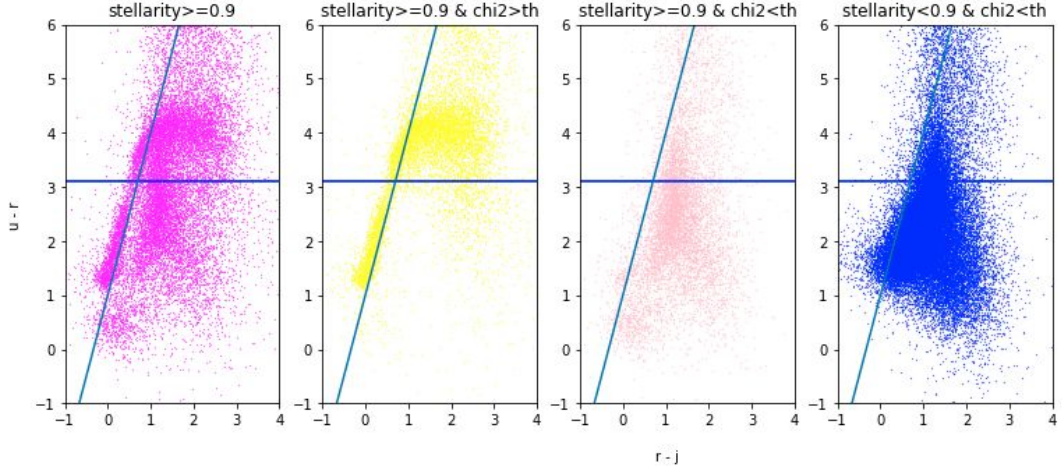



_ldustcriteria
2128615
1330769
77740
sf
ldust

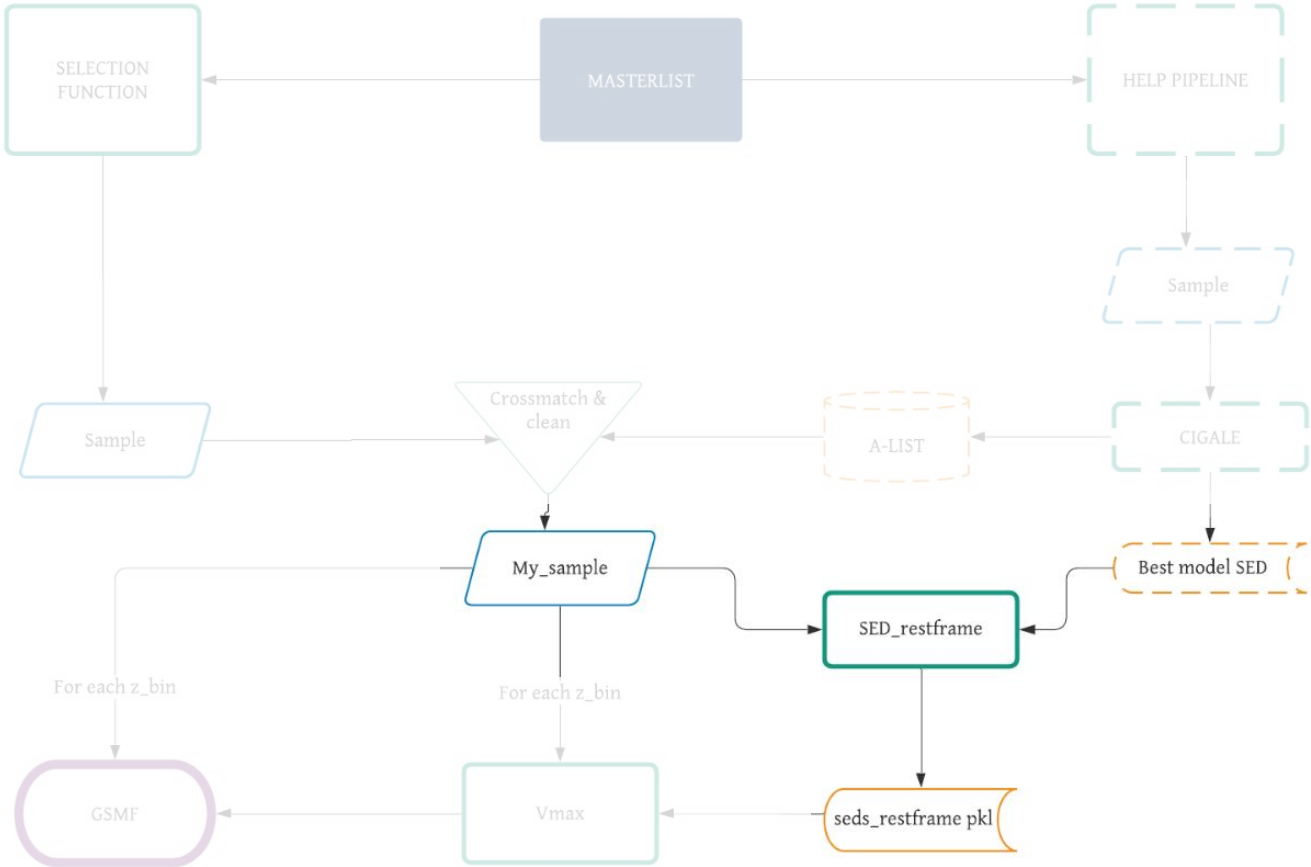
WORKFLOW GSMF




stellarity>=0.9
stellarity>=0.9 & chi2>th
stellarity>=0.9 & chi2<th
stellarity<0.9 & chi2<th
r - [3.6]
g - z


stellarity>=0.9
stellarity>=0.9 & chi2>th
stellarity>=0.9 & chi2<th
stellarity<0.9 & chi2<th
u - r
r - j

WORKFLOW GSMF



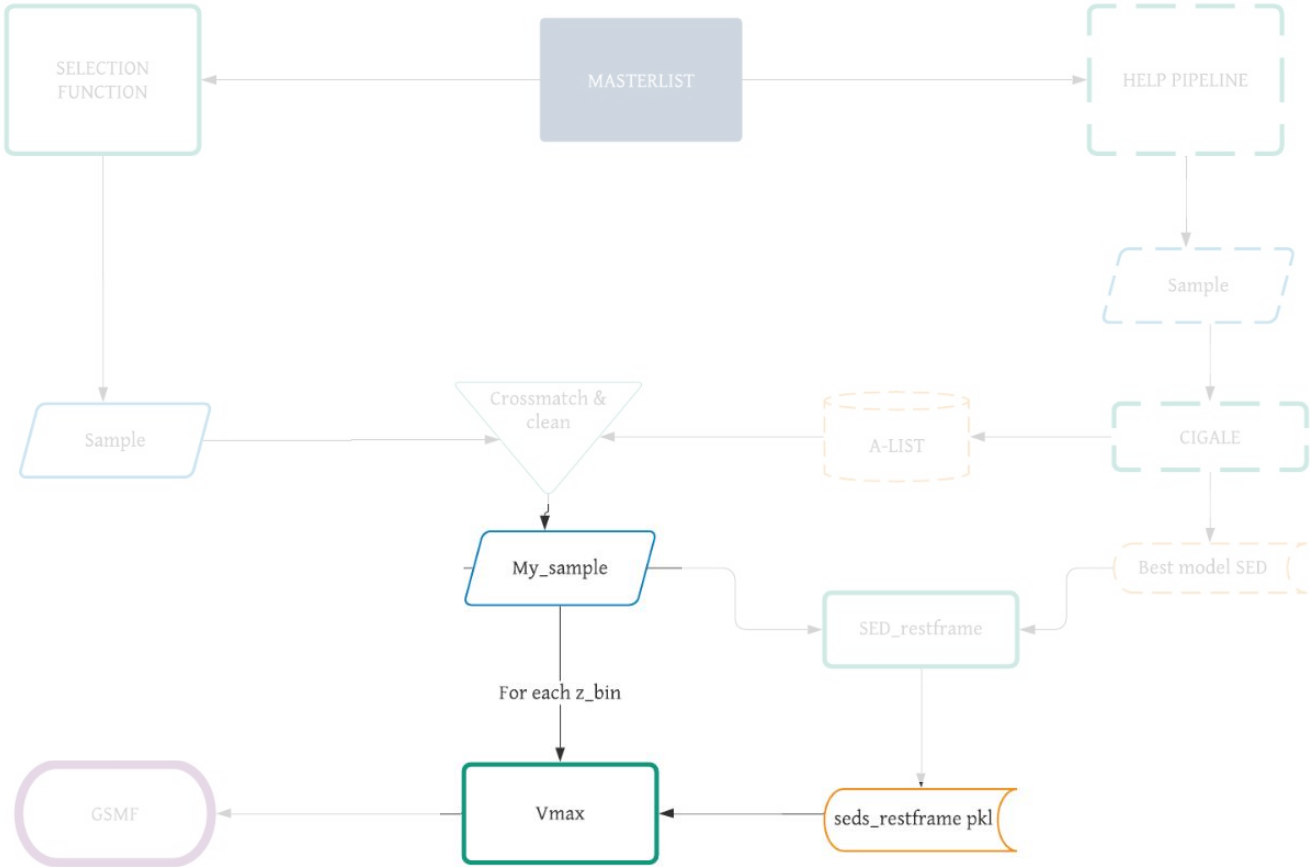
SED restframe

```
def sed_restframe(path_url, path_sed, catalogue):
    """
    param.
    ---------------
    path_url: str
        path to the dmu folders (either local or remote).
    path_sed: str
        path within the dmu folders.
    catalogue: astropy.table
        table containing at least the "help_id" of each source, and "redshift".

    output
    ---------------
    SEDs: astropy.Table
        Table with the seds from CIGALE. Save as "object" column.
    """
```

help_id	ra	dec	redshift	zspec	seds
	deg	deg			
bytes27	float64	float64	float64	float64	object
HELP_J000002.187-061445.916	0.009111123039974993	-6.24608767642777	0.163	nan	<pcigale.sed.SED object at 0x7f861c783160>
HELP_J000004.805-063531.209	0.0200206346240566	-6.592002602896851	0.2717	nan	<pcigale.sed.SED object at 0x7f861e475e50>
HELP_J002728.237-030737.134	6.867655196084523	-3.126981561972325	0.1179	0.1474596	<pcigale.sed.SED object at 0x7f861e475e80>
HELP_J002728.472-011933.054	6.868631582140365	-1.3258483183865915	0.2177	nan	<pcigale.sed.SED object at 0x7f861c7c9dc0>
HELP_J002728.573-005425.046	6.869054443130524	-0.9069572023383792	0.2417	nan	<pcigale.sed.SED object at 0x7f861c7c9fa0>

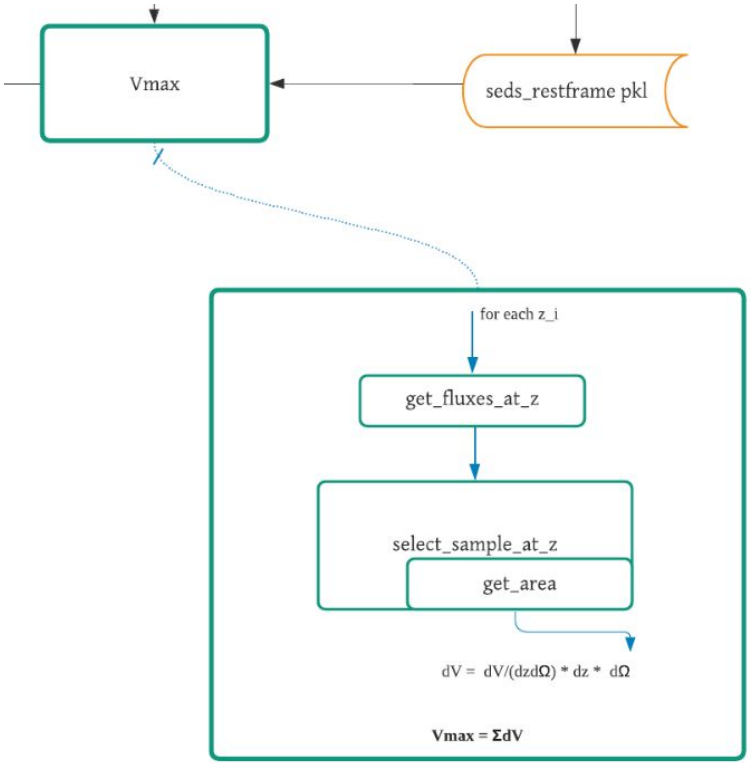
seds_restframe.pkl

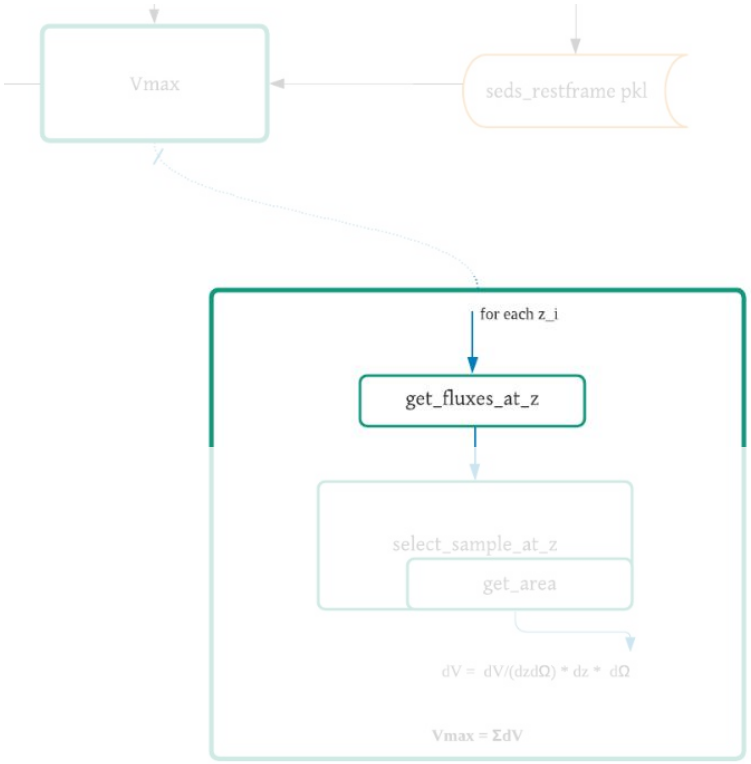
WORKFLOW GSMF



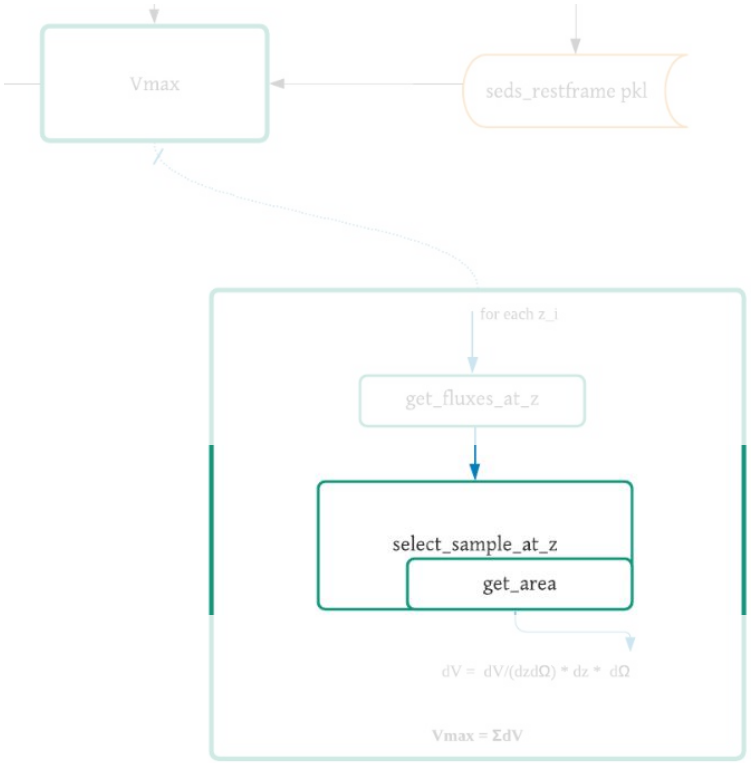
Vmax

```
def Vmax(catalogue, allbands, mocs, flims, cosmo, zmin, zmax, nbins=50):
    """
    It calculates the maximum volume each source can be observed, in a given redshift bin.

    param.
    ---------------
    catalogue: astropy.table
        HELP catalogue
    allbands: list [[..., str]]
        Names of bands used in the selections: [[opt_bands], [nir_bands], [mir_bands]]
    mocs: dict of MOC()
        Dictionary including MOCs for each survey/band.
    flims: dict
        Flux cuts limits for each band.
    cosmo: astropy.cosmology
        Cosmology parameters.
    zmin: float
        Minimum redshift bin.
    zmax: float
        Maximum redshift bin.
    nbins: int
        Number of dz inside the bin.

    output
    ---------------
    Vmax: astropy.Table
        Table including the Vmax calculation for each source.
    """
```


Vmax
seds_restframe pkl
for each z_i
get_fluxes_at_z
select_sample_at_z
get_area
dV = dV/(dzdΩ) * dz * dΩ
Vmax = ΣdV

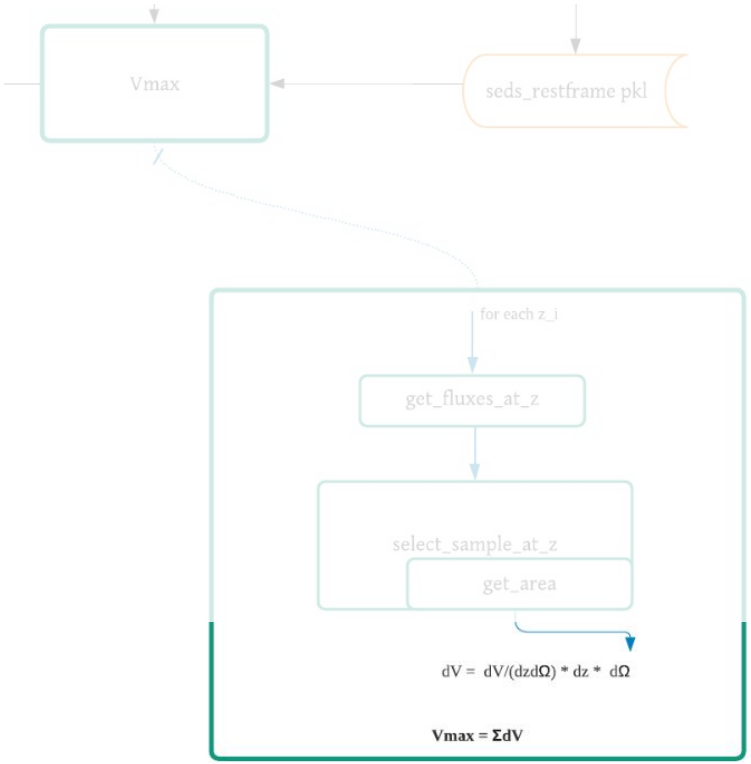



```python
def get_fluxes_at_z(sed_cat, sed_bands, r):
    """
    This function gets the fluxes of each source at the given bands, at a particular redshift "r"
    by redshifting the sed using CIGALE.

    param.
    ---------------
    sed_cat: pcigale.sed
        sed of the source at the restframe.
    sed_bands: list
        name of filters to compute the flux in cigale.

    output
    ---------------
    cat: astropy.Table
        Catalogue containing the fluxes of each source at the given redshift.
    """
```

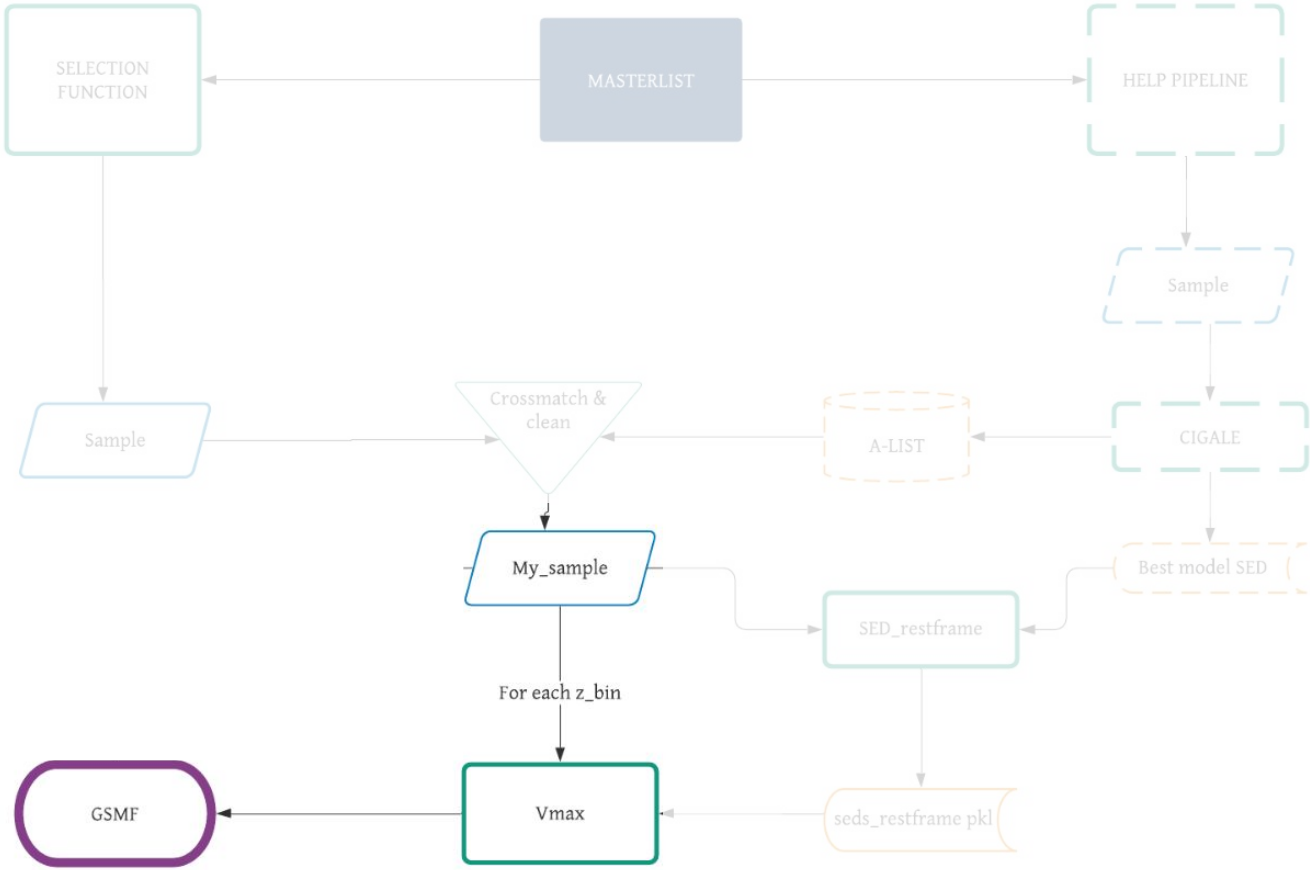


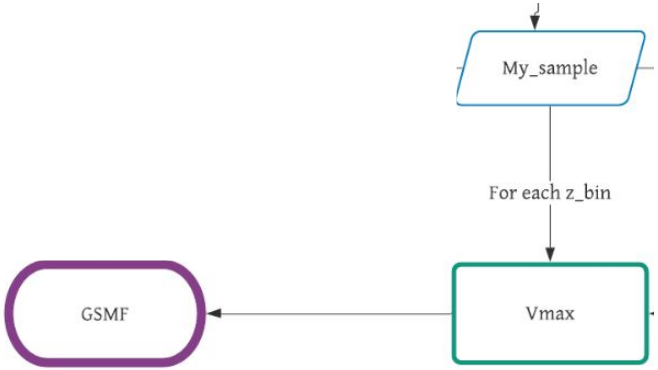

```
def select_sample_at_z(catalogue, allbands, mocs, cuts, ndet=2):
    """

    param.
    ---------------
    catalogue: astropy.table
        HELP catalogue
    allbands: list [[..., str]]
        Names of bands used in the selections: [[opt_bands], [nir_bands], [mir_bands]]
    mocs: dict of MOC()
        Dictionary including MOCs for each survey/band.
    cuts: dict
        Flux cuts limits for each band.
    ndet: int
        Number of detections required per group of bands (opt/nir/mid).
        Default 2; it requires 2 detection on each to pass the selection.

    output
    ---------------
    nb_band: astropy.Table
        Table with selected sources -has at least "ndet" detections above the magnitude cut.
        And the area in which each source has been observed.
    """
```

```
def get_areas(mocs, row):
    """
    This function gets the area in which an object is observed,
    based on whether it has been detected or not at a particular band/survey.

    param.
    ---------------
    mocs: dict of MOC()
        dictionary including MOCs for each survey/band.
    row: astropy.Table.row
        It should include Boolean columns for each moc_band.

    output
    ---------------
    area: float
        Area in which the source have been detected.
    """
```

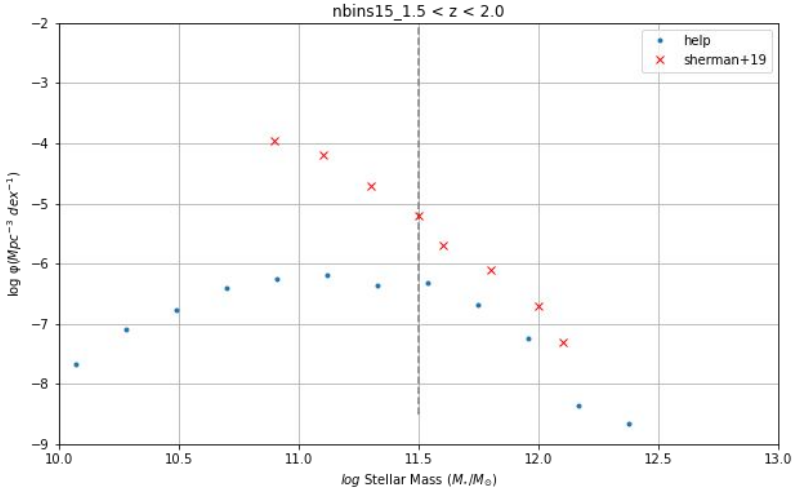

Vmax
seds_restframe pkl
for each z_i
get_fluxes_at_z
select_sample_at_z
get_area
dV = dV/(dzdΩ) * dz * dΩ
Vmax = ΣdV

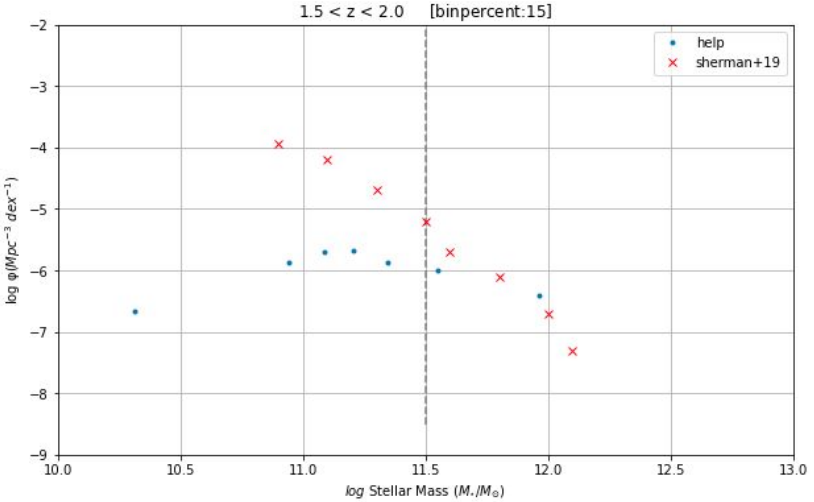
WORKFLOW GSMF

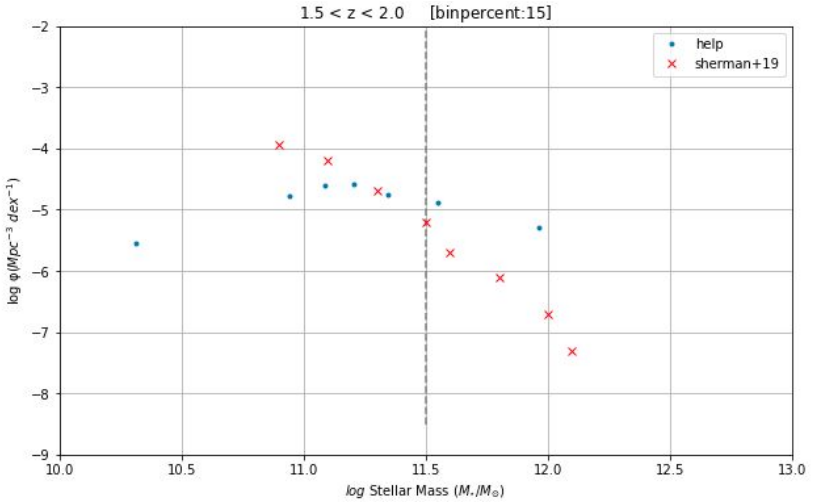





```
def plot_SMF(zmin, zmax, catalogue, mcol, zcol,
             nbins=None, bin_width=None, bin_percent=None):
    """
    Plot the Galaxy Stellar Mass Function between "zmin"-"zmax".

    param.
    ---------------
    catalogue: astropy.table
        HELP catalogue including Vmax.
    zmin: float
        Minimum redshift bin.
    zmax: float
        Maximum redshift bin.
    mcol: str
        Name of column for Mass
    zcol: str
        Name of column for redshift.
    nbins: int
        Number of Mass bins
    bin_width: float
        Mass bin width
    bin_percent: int
        Percentage of sources on each Mass bin.

    output
    ---------------
    gsmf_bins: astropy.Table
        Table including the Mass_centers and dN_dM for each Mass bin.
    """
```


nbins15_1.5 < z < 2.0
help
sherman+19
log φ(Mpc⁻³ dex⁻¹)
log Stellar Mass (M*/M⊙)


1.5 < z < 2.0 [binpercent:15]
help
sherman+19
log φ(Mpc⁻³ dex⁻¹)
log Stellar Mass (M*/M⊙)


1.5 < z < 2.0 [binpercent:15]
help
sherman+19
log φ(Mpc⁻³ dex⁻¹)
log Stellar Mass (M*/M⊙)