

initialization

```
In [1]: from plot_setup import *
import MCgenerator, dSph_model, coord
from sympy import *
init_printing()

In [10]: dSph_property = pd.read_csv("dSph_property.csv", index_col=0)
draco_prop = dSph_property.loc["Draco"]
RA0 = draco_prop.RAdeg
DE0 = draco_prop.DEdeg
DIST = draco_prop.DIST
```

load MCMC result

```
In [2]: modKI17_result = pd.read_csv("MCMCresult/MCMC_refR-re_v001.csv"),pd.read_csv("MCMCresult/MCMClog_refR-re_v001.csv")
KI17_result = pd.read_csv("MCMCresult/MCMC_KI17_v001.csv"),pd.read_csv("MCMCresult/MCMClog_KI17_v001.csv")

In [5]: display(modKI17_result[1])
```

	odds	re	iter_num
0	0.500000	5.000000	5000
1	0.500000	5.000000	5000
2	0.398303	6.137237	10000
3	0.419732	6.039228	100000
4	0.419732	6.039228	100000

results

modKI17 analysis

```
In [6]: display(mypairplot(modKI17_result[0][-200000::25]))
```

Note: The following warnings occur, but no problem.

UserWarning: The following kwargs were not used by contour: 'label', 'color'

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

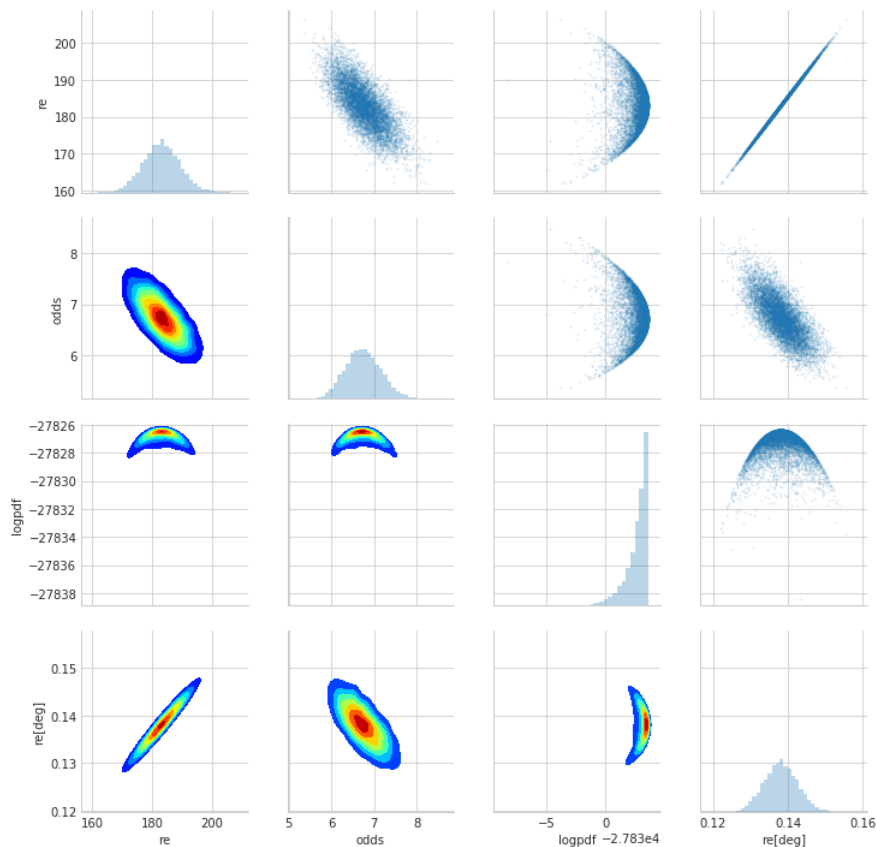
/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

<seaborn.axisgrid.PairGrid at 0x7f1af14cf518>



KI17 analysis

```
In [7]: display(KI17_result[1])
```

	re	s	iter_num
0	5.000000	0.100000	5000
1	5.000000	0.100000	5000
2	5.925653	0.010440	10000
3	6.042863	0.010497	20000
4	6.109106	0.010500	20000

```
In [8]: display(mypairplot(KI17_result[0][:-200000::25]))
```

Note: The following warnings occur, but no problem.

UserWarning: The following kwargs were not used by contour: 'label', 'color'

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

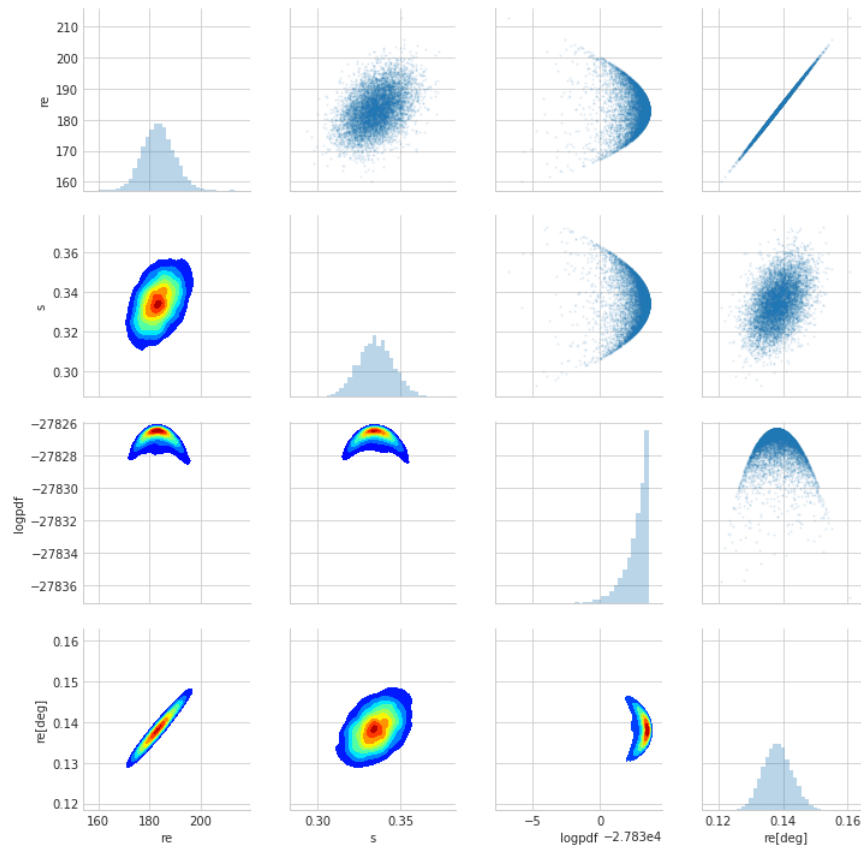
/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

/home/gomesu/.pyenv/versions/anaconda3-5.2.0/lib/python3.6/site-packages/matplotlib/contour.py:960: UserWarning: The following kwargs were not used by contour: 'label', 'color'
s)

<seaborn.axisgrid.PairGrid at 0x7f1af0ed2ef0>



```
In [92]: class KI17_loglikelihood:
def __init__(self, Rs):
    self.Rs = Rs
    self.RoI_R = np.max(Rs) # use Rs.max as the RoI
    print(self.Rs.describe())
    self.mem = dSph_model.plummer_model(re_pc=100)
    self.fg = dSph_model.uniform2d_model(Rmax_pc=self.RoI_R)

def _call_(self, re, s):
    if re<0 or not 0<s<1:
        return -np.inf
    else:
        mem, fg = self.mem, self.fg
        mem.update({"re_pc":re})
        ref_R = re

        C1 = 1/mem.cdf_R(self.RoI_R)
        C0 = fg.density_2d(self.RoI_R) # fg.density_2d returns constant for arbitrary input
        #s = C0/(sigmafg*mem.density_2d(self.Rs)*C1 + C0)
        #s = C0/(sigmafg*mem.density_2d(self.Rs)*C1)
        #s = 1/(1+ 1/odds * mem.density_2d(ref_R)/mem.mean_density_2d(self.RoI_R))

        loglikelis = np.log(2*np.pi*self.Rs)+np.log(s*C1*mem.density_2d(self.Rs)+(1-s)*C0)
        return np.sum(loglikelis)
def loglikelis(self, re, s):
    if re<0 or not 0<s<1:
        return -np.inf
    else:
        mem, fg = self.mem, self.fg
        mem.update({"re_pc":re})
        ref_R = re

        C1 = 1/mem.cdf_R(self.RoI_R)
        C0 = fg.density_2d(self.RoI_R) # fg.density_2d returns constant for arbitrary input
        #s = C0/(sigmafg*mem.density_2d(self.Rs)*C1 + C0)
        #s = C0/(sigmafg*mem.density_2d(self.Rs)*C1)
        #s = 1/(1+ 1/odds * mem.density_2d(ref_R)/mem.mean_density_2d(self.RoI_R))

        ret = np.log(2*np.pi*self.Rs)+np.log(s*C1*mem.density_2d(self.Rs)+(1-s)*C0)
        return ret
def s_to_odds(self, s, ref_R, re):
    # NOTE: Here we define mem who is new object.
    # Here we set input "re" to the parameter "re_pc" of new plummer_model.
    # So this new plummer_model has same type of input "re".
    # numpy broadcast enables us to convert parameter.
    # If you use self.mem instead of new mem,
    # self.mem may have "re=100" duo to the initialization, thus this new mem is required.
    mem = dSph_model.plummer_model(re_pc=re)
    return mem.density_2d(ref_R)/mem.mean_density_2d(self.RoI_R)/(1/s-1)
```

```
In [93]: dra_dde_deg = pd.DataFrame(np.loadtxt("Draco_SDSS_cut_xy1.0.csv", comments="#", delimiter=",", columns=("x", "y")))
Rs = coord.projected_distance(
    dist=DIST,
    ra_center = RA0,
    de_center = DE0,
    ra = RA0 + dra_dde_deg.x,
    de = DE0 + dra_dde_deg.y,
    dtype="deg")
loglikeli = KI17_loglikelihood(Rs)

count    3895.000000
mean      672.749609
std       416.428473
min        1.698731
25%       248.562571
50%       702.831670
75%      1054.806601
max      1325.756354
dtype: float64
```

convert $s \rightarrow$ new parameter ($\tilde{\Sigma}_{FG}$, or $\mathcal{O} \equiv \Sigma_{FG}^{-1}$ (= odds))

```
In [94]: KI17_result[0]["odds_converted"] = loglikeli.s_to_odds(ref_R=KI17_result[0].re.values, s=KI17_result[0].s.values, re=KI17_result[0].re.values)
```

```
In [91]: compare = pd.DataFrame()
compare["odds_original"] = modKI17_result[0][-200000:].odds.values
compare["odds_converted"] = KI17_result[0][-200000:].odds_converted.values
fig, ax=plt.subplots()
compare.odds_original.hist(bins=64, ax=ax, label="odds_original", alpha=0.3)
compare.odds_converted.hist(bins=64, ax=ax, label="odds_converted", alpha=0.3)
plt.legend()
```

Out[91]: <matplotlib.legend.Legend at 0x7f1af13aef28>

