mcmc_analysis_mice_F_sys

May 22, 2025

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

```
/Users/navehr/Dropbox/naveh/weizmann/uri alon/aging/code_3/baysian02
```

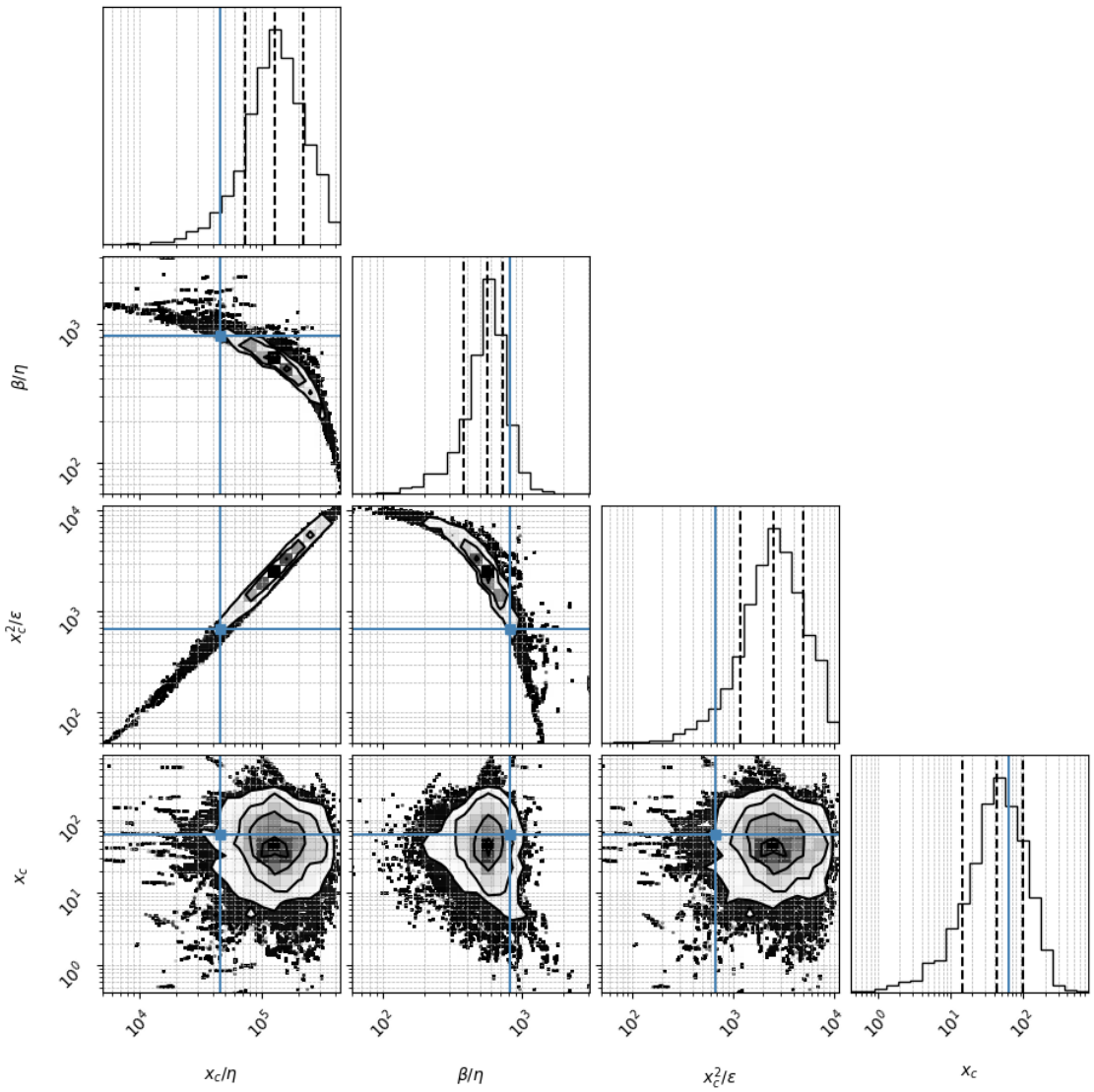
```
posterior_csvs_baysian01/mice_F_post.csv
```

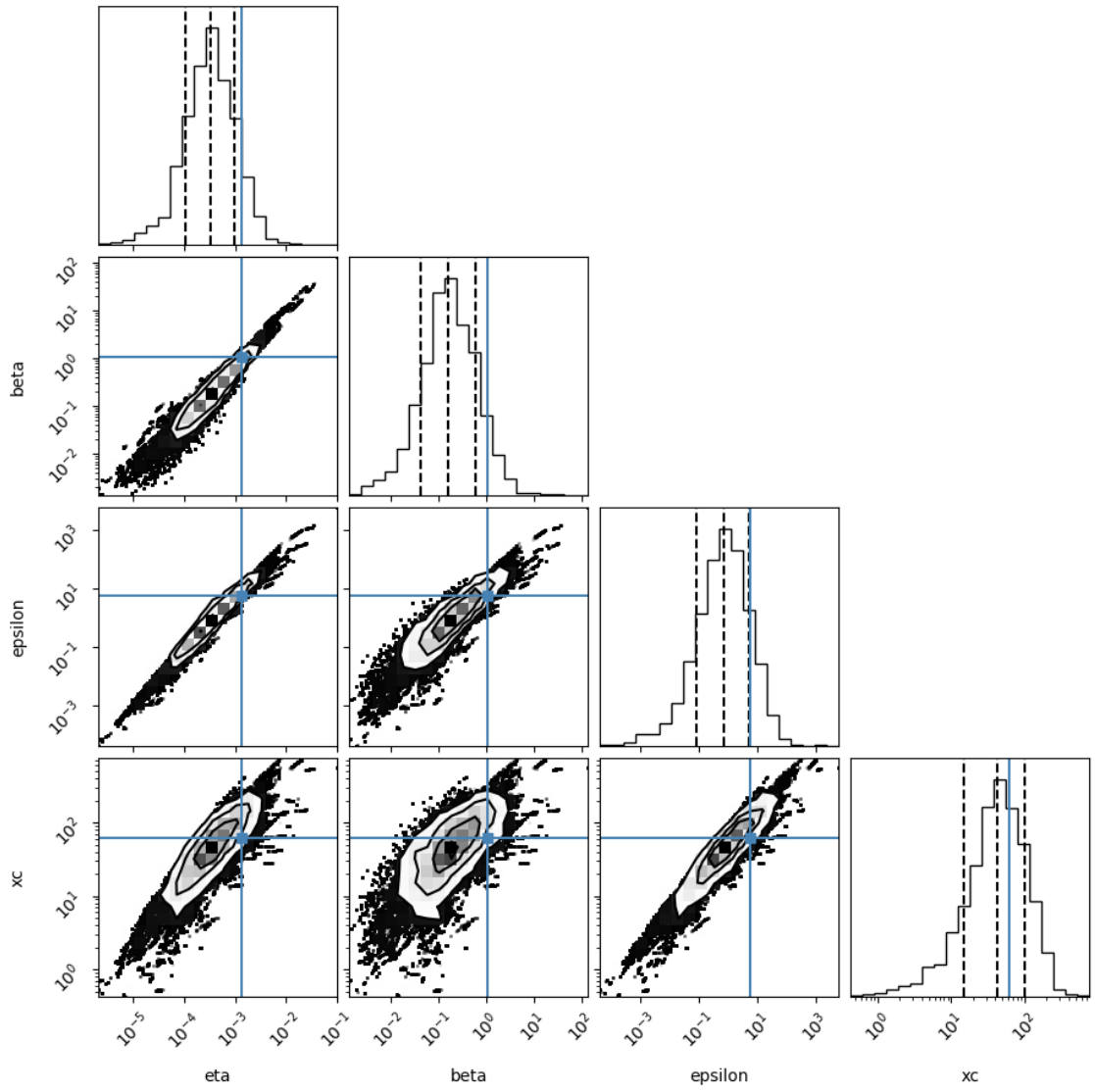
```
Reading Mice_F
```

1 # 1. Density coner plot

A sample is 1 parameter set scanned. For the corner plot below, the quantiles (represented by the solid lines) are 0.16,0.5,0.84 of the samples. Dots represent individual samples (outside the line surrounding 0.84 of the samples) The parameter search is performed in the transformed space of x_c/η, β/η, x_c^2/ϵ, x_c but we also show the regular parameters

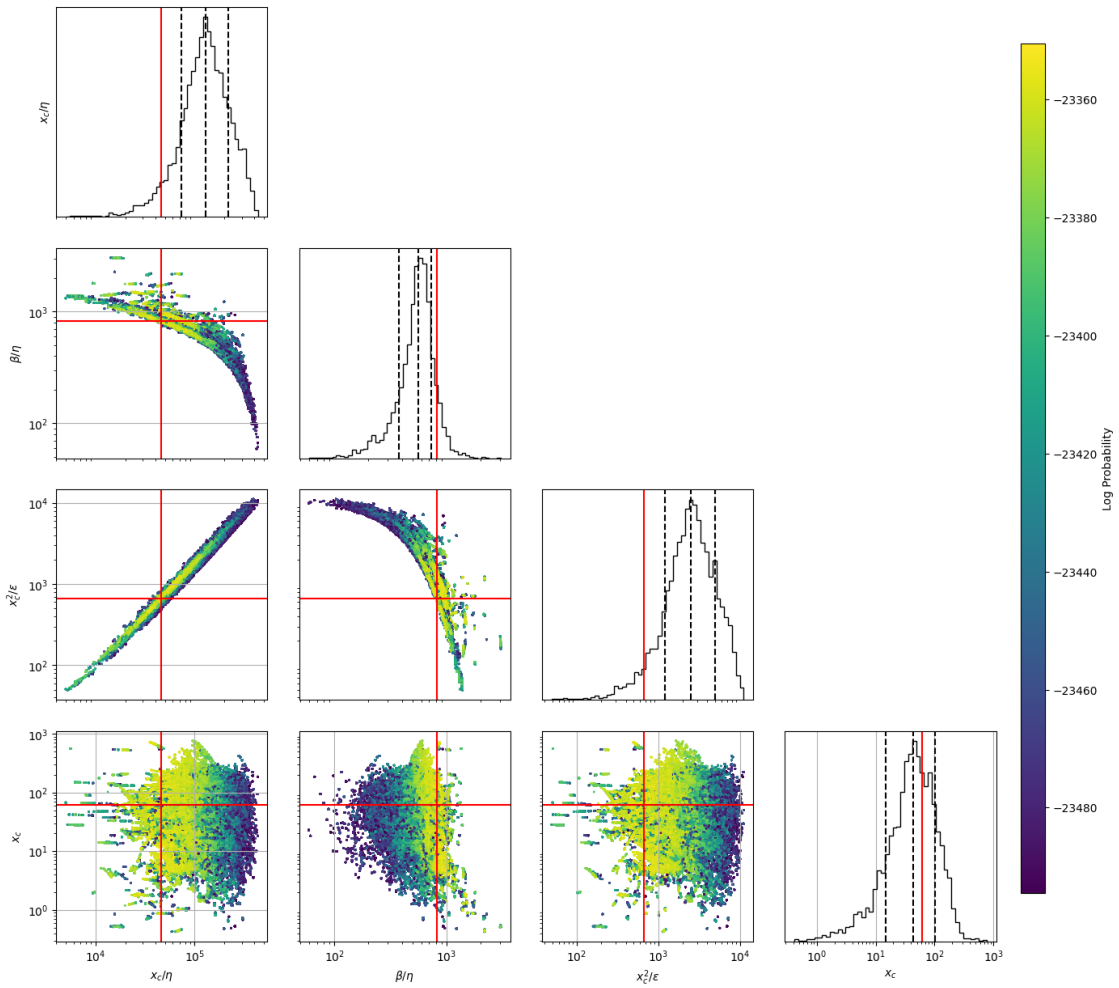
```
(16,)
```


x_c/η
β/η
x_c^2/ε
x_c



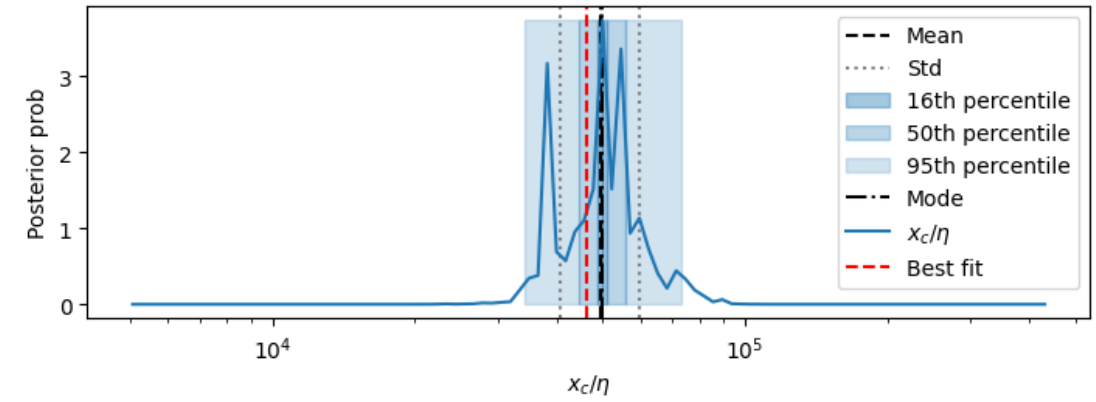
2 2. Heat map corner plot of raw samples

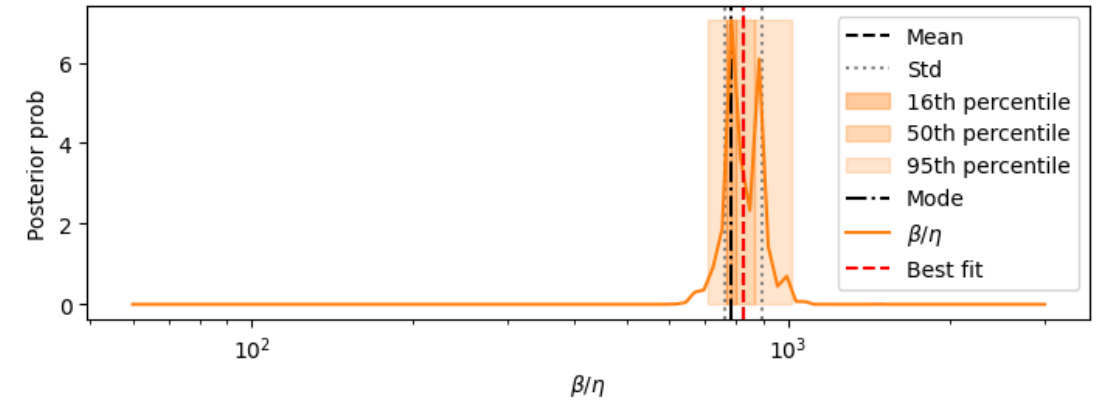
This plot shows all the raw sample points and their lnprobability

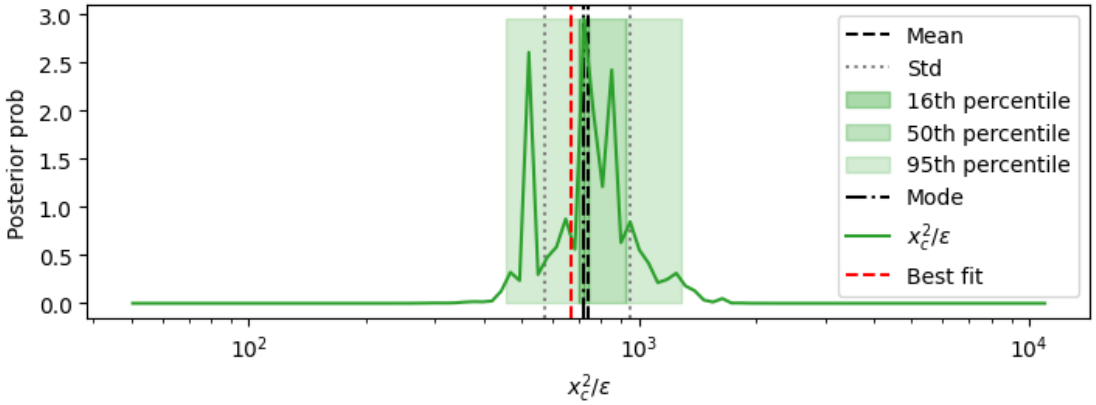


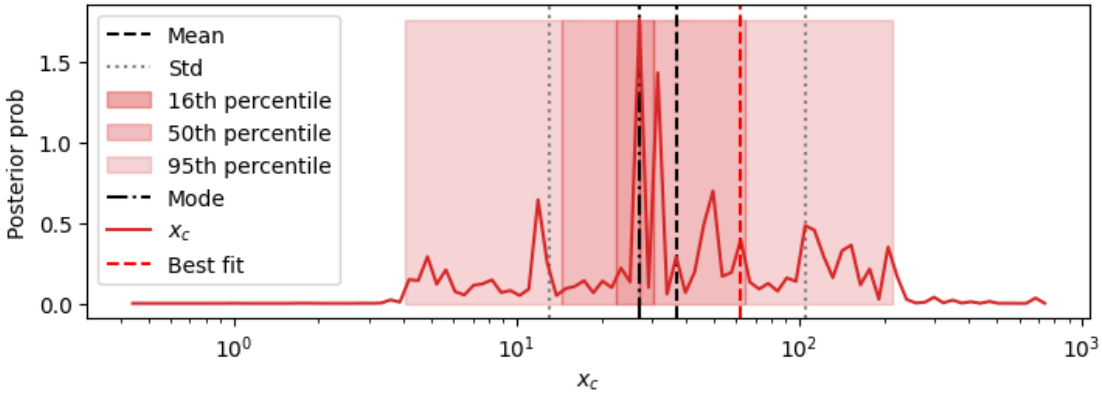
3 3 posterior distributions of parameters

1d marginalizations of posterior distributions. we use a grid of size nbins=100-150


Mean
Std
16th percentile
50th percentile
95th percentile
Mode
x_c/η
Best fit
Posterior prob
x_c/η

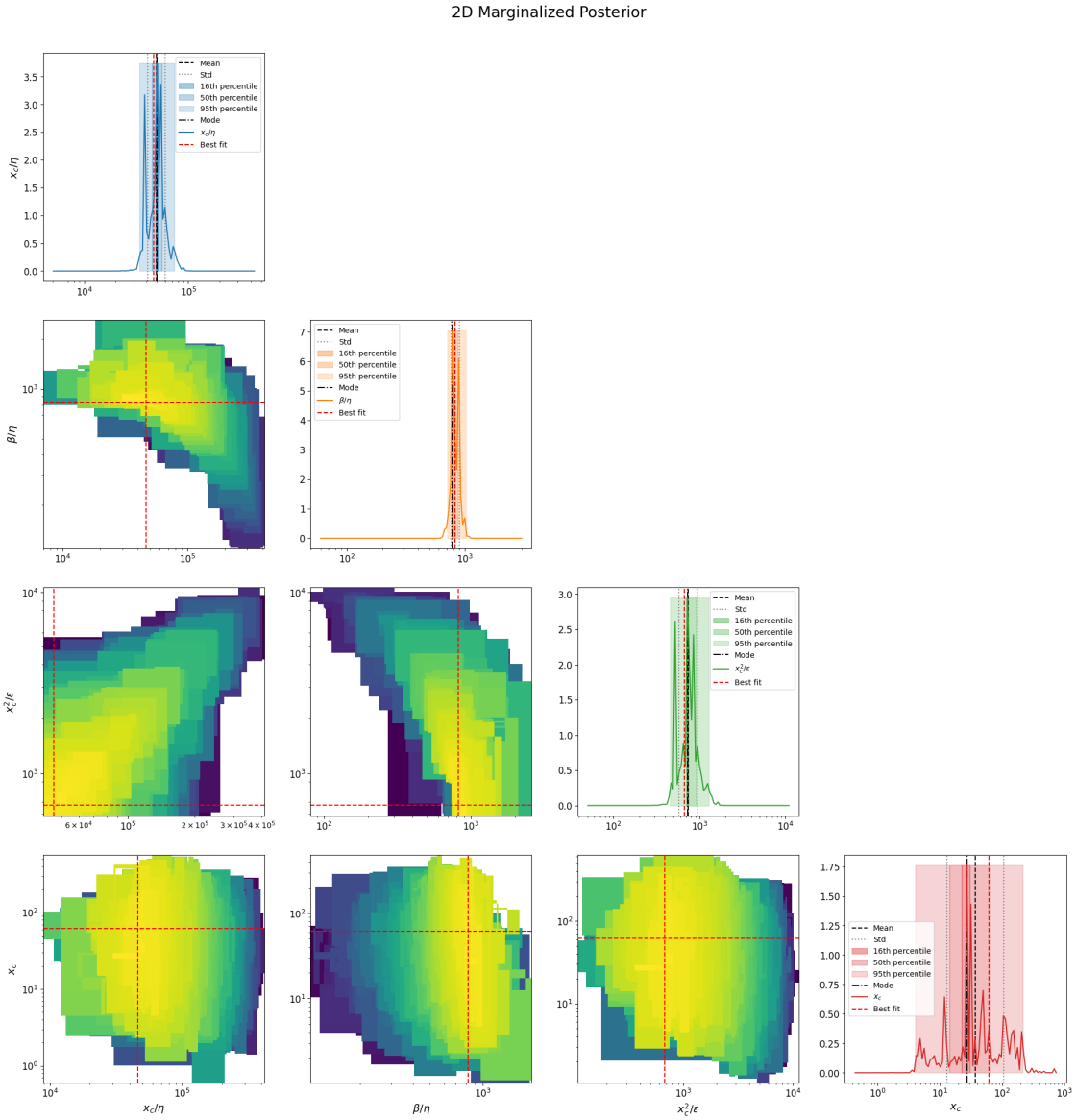

Mean
Std
16th percentile
50th percentile
95th percentile
Mode
β/η
Best fit
Posterior prob
β/η


Mean
Std
16th percentile
50th percentile
95th percentile
Mode
x_c^2/ε
Best fit
Posterior prob
x_c^2/ε


Mean
Std
16th percentile
50th percentile
95th percentile
Mode
x_c
Best fit
Posterior prob
x_c

2D marginalizations of posterior distributions

```
Creating corner plot:   0%|          | 0/4 [00:00<?,
?it/s]/Users/navehr/Dropbox/naveh/weizmann/uri
alon/aging/code_3/SRtools/SRtools/samples_utils.py:474: UserWarning:

The input coordinates to pcolormesh are interpreted as cell centers, but are not
monotonically increasing or decreasing. This may lead to incorrectly calculated
cell edges, in which case, please supply explicit cell edges to pcolormesh.

Creating corner plot: 100%|     | 4/4 [00:05<00:00,  1.36s/it]
```



```
Binning samples: 100%|     | 4/4 [00:00<00:00, 14.92it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
784262.69it/s]
Averaging log-probabilities: 100%|     | 38630/38630 [00:00<00:00,
422148.19it/s]
Binning samples: 100%|     | 4/4 [00:00<00:00, 14.08it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
716625.41it/s]
Averaging log-probabilities: 100%|     | 16116/16116 [00:00<00:00,
352810.95it/s]
Binning samples: 100%|     | 4/4 [00:00<00:00, 16.51it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
778333.42it/s]
Averaging log-probabilities: 100%|     | 46289/46289 [00:00<00:00,
425285.67it/s]
Binning samples: 100%|     | 4/4 [00:00<00:00, 17.38it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
725035.21it/s]
Averaging log-probabilities: 100%|     | 47936/47936 [00:00<00:00,
423239.34it/s]
Binning samples: 100%|     | 4/4 [00:00<00:00, 17.31it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
814916.59it/s]
Averaging log-probabilities: 100%|     | 25916/25916 [00:00<00:00,
418627.58it/s]
Binning samples: 100%|     | 4/4 [00:00<00:00, 17.39it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
791998.79it/s]
Averaging log-probabilities: 100%|     | 25403/25403 [00:00<00:00,
421577.87it/s]
Binning samples: 100%|     | 4/4 [00:00<00:00, 17.26it/s]
Processing samples: 100%|     | 2677534/2677534 [00:03<00:00,
770730.53it/s]
Averaging log-probabilities: 100%|     | 21365/21365 [00:00<00:00,
417832.60it/s]
```

4 4. Table of results

mode is the marginalized mode, max_likwlihood is the sample with highest likelihood mode_overall is the 4D posterior mode

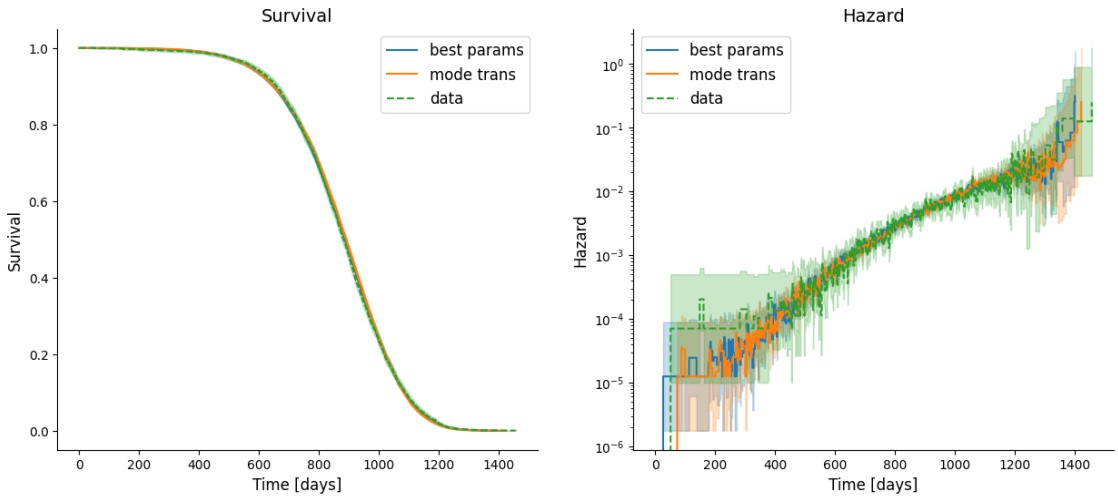
	mean	std	mode	\
xc/eta	50942.565	[11336.891, 9273.207]	51832.668	
beta/eta	817.858	[74.413, 68.207]	810.708	
xc^2/epsilon	771.968	[224.013, 173.629]	756.961	

xc	38.975	[69.44, 24.964]	50.279
eta	0.0008	[0.00155, 0.000528]	0.00264
beta	0.667	[1.213, 0.431]	0.71
epsilon	2.072	[15.599, 1.829]	20.192
sqrt(xc/eta)	227.789	[24.004, 21.716]	227.668
s= eta^0.5*xc^1.5/epsilon	3.46	[0.58, 0.497]	3.384
beta*xc/epsilon	12.262	[0.778, 0.731]	12.234
eta*xc/epsilon	0.0152	[0.000905, 0.000854]	0.0151
Fx=beta^2/eta*xc	12.548	[5.294, 3.723]	12.658
Dx =beta*epsilon/eta*xc^2	1.026	[0.399, 0.287]	1.036
Pk=beta*k/epsilon	0.164	[0.324, 0.109]	0.052
Fk=beta^2/eta*k	1029.941	[1725.101, 644.909]	1320.887
Dk =beta*epsilon/eta*k^2	7050.27	[51276.8, 6198.07]	66974.732
Fk^2/Dk=beta^3/eta*epsilon	171.055	[78.891, 53.991]	171.586
beta^2/epsilon	0.206	[0.0709, 0.0527]	0.197
k/beta	0.702	[1.189, 0.441]	0.704
k/epsilon	0.201	[1.356, 0.175]	0.0281
best fit_MedianLifetime	886.38	0.51	886.38
best fit_MaxLifetime	1399.78	0	1399.78
data_MedianLifetime	882.0	0.52	882.0
data_MaxLifetime	1456	0	1456

	percentile_16	percentile_50	\
xc/eta	[49556.82, 54213.033]	[45300.516, 59306.731]	
beta/eta	[779.277, 821.464]	[759.003, 865.935]	
xc^2/epsilon	[716.902, 799.258]	[666.769, 923.966]	
xc	[40.144, 59.901]	[20.95, 89.38]	
eta	[0.00191, 0.00294]	[0.000748, 0.00316]	
beta	[0.586, 0.86]	[0.543, 2.002]	
epsilon	[14.738, 31.377]	[1.965, 35.588]	
sqrt(xc/eta)	[222.614, 232.837]	[212.839, 243.53]	
s= eta^0.5*xc^1.5/epsilon	[3.275, 3.496]	[3.068, 3.815]	
beta*xc/epsilon	[12.096, 12.374]	[11.824, 12.95]	
eta*xc/epsilon	[0.0149, 0.0153]	[0.0145, 0.0157]	
Fx=beta^2/eta*xc	[11.306, 13.143]	[9.726, 15.279]	
Dx =beta*epsilon/eta*xc^2	[0.952, 1.128]	[0.804, 1.263]	
Pk=beta*k/epsilon	[0.0419, 0.0647]	[0.037, 0.175]	
Fk=beta^2/eta*k	[1046.24, 1667.63]	[790.953, 3202.987]	
Dk =beta*epsilon/eta*k^2	[42730.29, 92325.529]	[7080.033, 119357.706]	
Fk^2/Dk=beta^3/eta*epsilon	[148.439, 180.077]	[122.36, 218.459]	
beta^2/epsilon	[0.177, 0.204]	[0.165, 0.252]	
k/beta	[0.538, 0.789]	[0.25, 0.852]	
k/epsilon	[0.0205, 0.0385]	[0.0159, 0.174]	
best fit_MedianLifetime	[885.89, 886.89]	[885.89, 886.89]	
best fit_MaxLifetime	[1399.78, 1399.78]	[1399.78, 1399.78]	
data_MedianLifetime	[881.52, 882.52]	[881.52, 882.52]	
data_MaxLifetime	[1456, 1456]	[1456, 1456]	

	percentile_95	max_likelihood	mode_overall
xc/eta	[35653.519, 77643.449]	46236.322	37651.206
beta/eta	[683.047, 962.229]	824.86	899.104
xc^2/epsilon	[481.166, 1327.636]	670.741	526.572
xc	[5.162, 189.293]	62.013	27.471
eta	[9.24e-05, 0.00393]	0.00134	0.00073
beta	[0.086, 2.939]	1.106	0.656
epsilon	[0.0308, 51.926]	5.733	1.433
sqrt(xc/eta)	[188.821, 278.646]	215.026	194.039
s= eta^0.5*xc^1.5/epsilon	[2.578, 4.742]	3.119	2.714
beta*xc/epsilon	[11.558, 14.842]	11.966	12.574
eta*xc/epsilon	[0.0136, 0.0171]	0.0145	0.014
Fx=beta^2/eta*xc	[6.191, 24.001]	14.716	21.47
Dx =beta*epsilon/eta*xc^2	[0.511, 1.876]	1.23	1.707
Pk=beta*k/epsilon	[0.0288, 1.445]	0.0965	0.229
Fk=beta^2/eta*k	[162.091, 5105.324]	1825.116	1179.612
Dk =beta*epsilon/eta*k^2	[116.299, 154304.686]	18916.972	5154.043
Fk^2/Dk=beta^3/eta*epsilon	[75.486, 354.115]	176.088	269.979
beta^2/epsilon	[0.108, 0.333]	0.213	0.3
k/beta	[0.17, 5.81]	0.452	0.762
k/epsilon	[0.00962, 14.306]	0.0872	0.349
best fit_MedianLifetime	[885.89, 886.89]	886.38	NaN
best fit_MaxLifetime	[1399.78, 1399.78]	1399.78	NaN
data_MedianLifetime	[881.52, 882.52]	882.0	NaN
data_MaxLifetime	[1456, 1456]	1456	NaN

```
/Users/navehr/Dropbox/naveh/weizmann/uri
alon/aging/code_3/SRtools/SRtools/utils.py:56: UserWarning:

Columns {'mice_F', 'mice_F 95% CI'} already exist and neither doubles nor
override is allowed - skipping these columns

/Users/navehr/Dropbox/naveh/weizmann/uri
alon/aging/code_3/SRtools/SRtools/utils.py:56: UserWarning:

Columns {'mice_F'} already exist and neither doubles nor override is allowed -
skipping these columns

/Users/navehr/Dropbox/naveh/weizmann/uri
alon/aging/code_3/SRtools/SRtools/utils.py:56: UserWarning:

Columns {'mice_F'} already exist and neither doubles nor override is allowed -
skipping these columns


 Text(0, 0.5, 'Hazard')
```




```
Text(0, 0.5, 'Prob density')
```