

Bayesian_fitting_2019

August 25, 2023

```
[1]: import P3_seenovate as p3
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('data/2019_data.csv', delimiter=",")
df["date"] = pd.to_datetime(df["date"], format = "%Y-%m-%d")
df["athlete"] = df["athlete"] - 1
np.random.seed(123)

# fix an outlier:
df.loc[(df["athlete"] == 3) & (df["date"] == "2019-09-06"), "performance"] = 0

mods = []
for i in range(9):
    mods.append({})
    mods[i]["Fitness"] = p3.Fitness (df.loc[df["athlete"] == i,:].copy(),
                                     objective = "min",
                                     HL_fitness = 30)
    mods[i]["Banister"] = p3.Banister(df.loc[df["athlete"] == i,:].copy(),
                                       objective = "min",
                                       HL_fitness = 30,
                                       HL_fatigue = 2)

athletes_id = [0,1,2,3,4,7,8] # 6 and 7 removed: not enough data for cross val
```

```
[2]: freq = pd.read_csv("Performances_in_training.csv")
```

```
[ ]: for model in ["Fitness", "Banister"]:
    for i in athletes_id:
        pos = [x for x in range(len(freq)) if freq["model"].iloc[x] == model_
        and freq["athlete"].iloc[x] == i+1]
        seed_bayes = np.random.randint(low = 1, high=1e5)

        if model == "Fitness":
            init_values = {"a0": float(freq["a0"].iloc[pos]),
                           "tauG": mods[i]["Fitness"].tau["G"],
                           "sigma": float(freq["RMSE"].iloc[pos])}
```

```

else:
    init_values = {"a0": float(freq["a0"].iloc[pos]),
                   "tauG": mods[i]["Banister"].tau["G"],
                   "tauH": mods[i]["Banister"].tau["H"],
                   "sigma": float(freq["RMSE"].iloc[pos])}

    mods[i][model].fitBayes(hyparams = {"sd_a0": 5, # flat
                                       "sd_k": 5, # flat
                                       "sd_sigma": 25,
                                       "b_tau": 2},
                           bounds = {"a0": (-1e4,1e4),
                                     "k" : (-1,0),
                                     "tauG": (0,100),
                                     "tauH": (0,100)},
                           info_tau = True,
                           init = init_values,
                           num_chains = 10,
                           num_warmup = 4e4,
                           num_samples = 1e4,
                           num_thin = 10,
                           seed_bayes = seed_bayes)

```

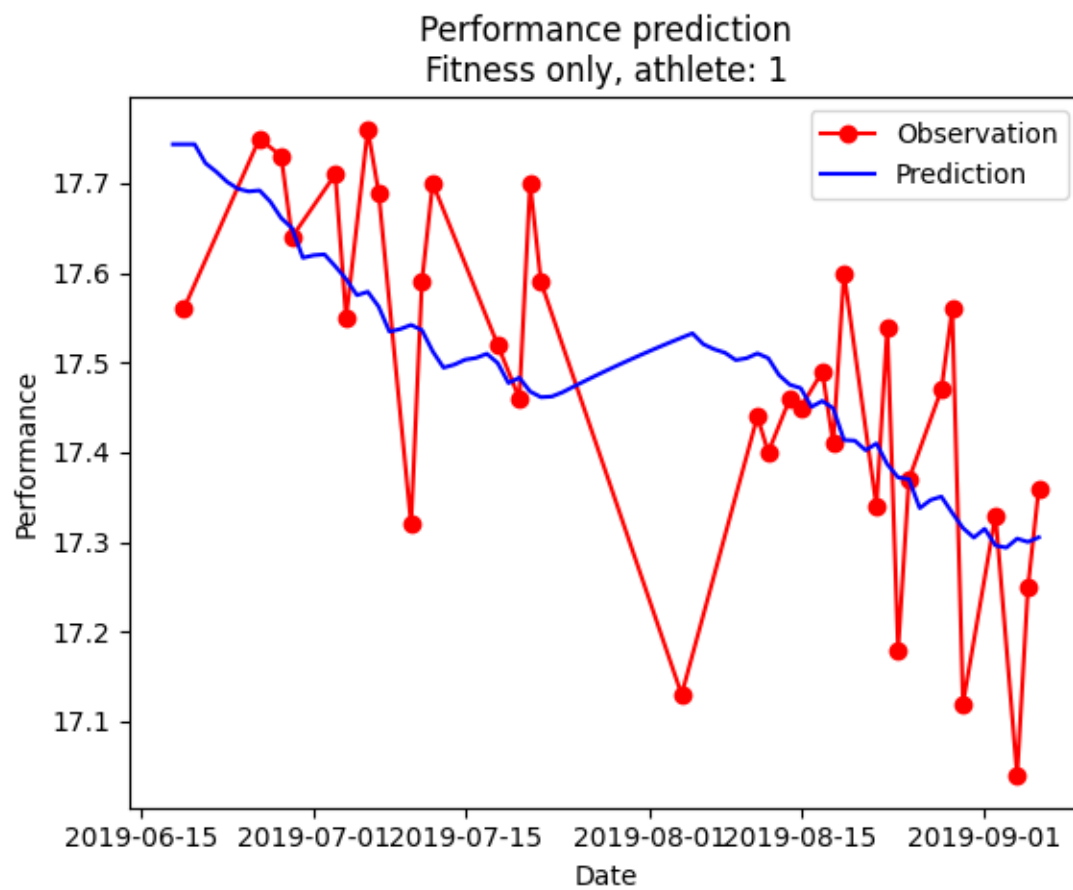
```

[4]: for i in athletes_id:
    mods[i]["Fitness"].train.plot_perf(show = False)
    plt.title("Performance prediction\nFitness only, athlete: "+str(i+1))
    plt.legend()
    plt.show()

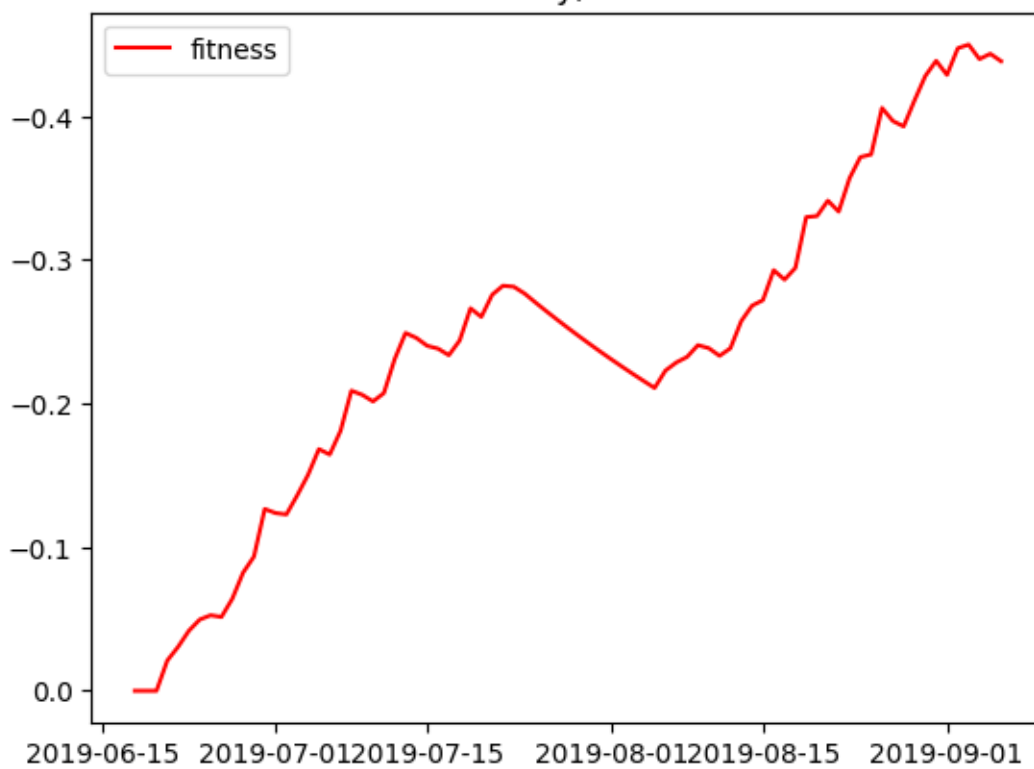
    mods[i]["Fitness"].train.plot_state(show = False)
    plt.title("Hidden states\nFitness only, athlete: "+str(i+1))
    plt.legend()
    plt.show()

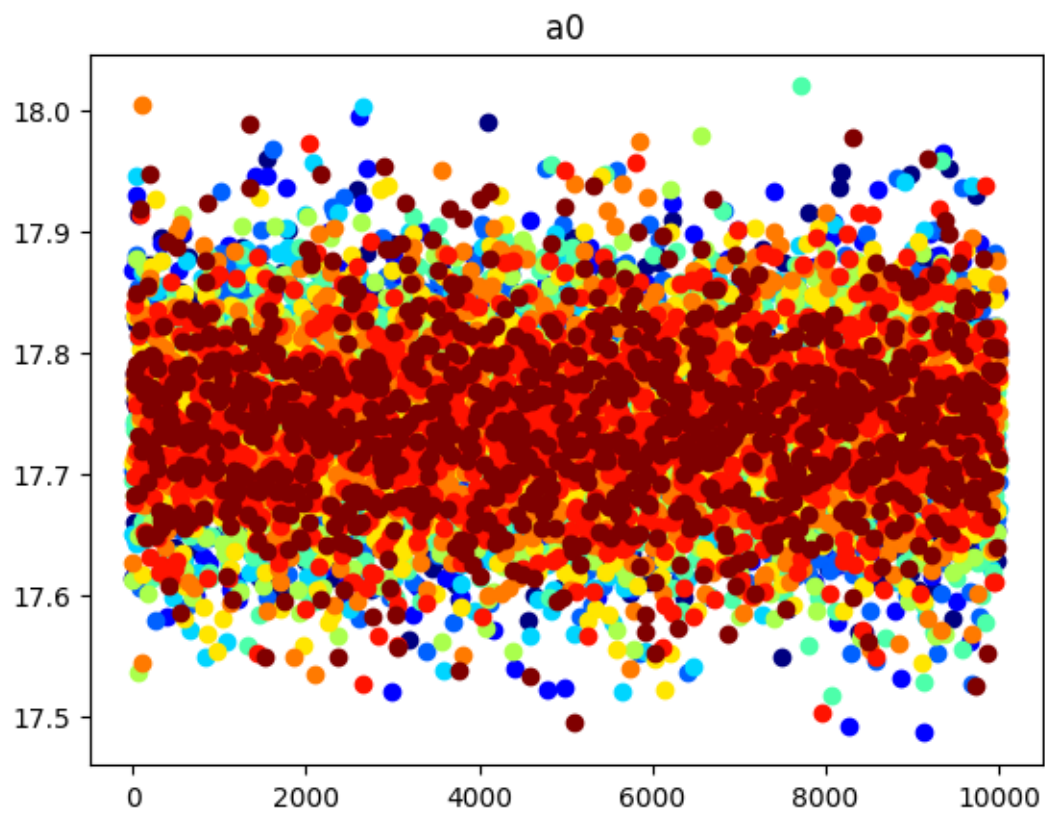
    mods[i]["Fitness"].train.plot_rawchains()
    mods[i]["Fitness"].train.plot_posteriors()

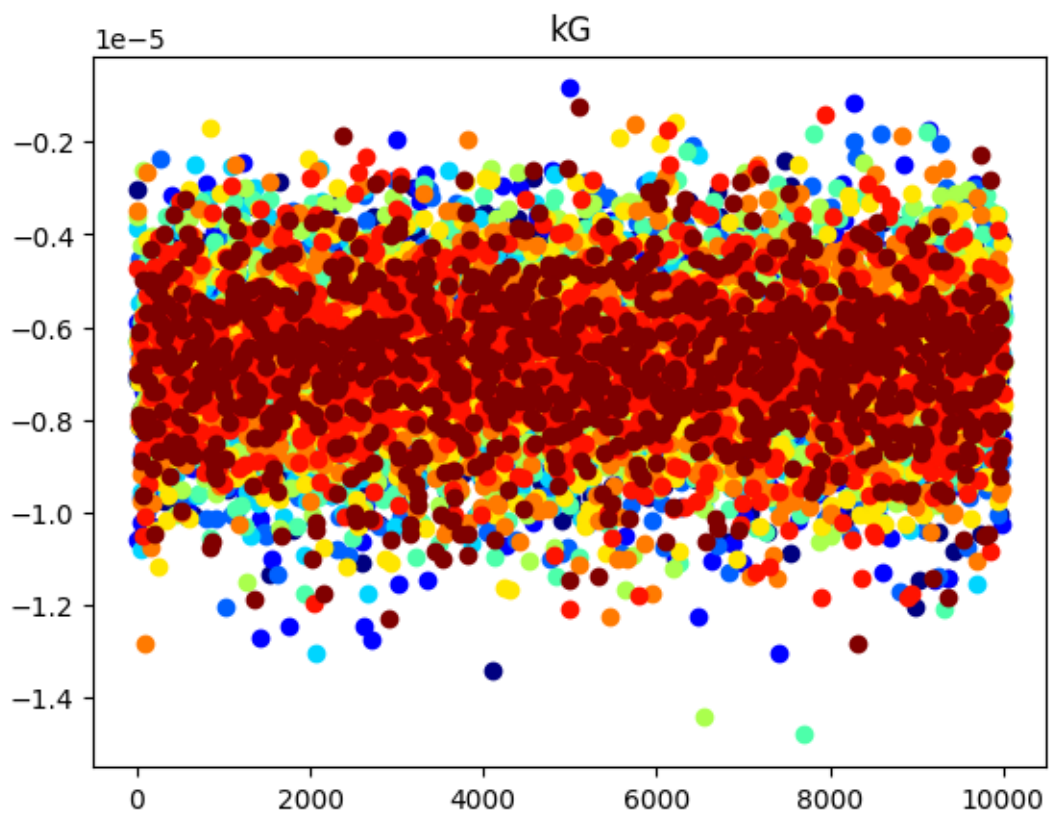
```

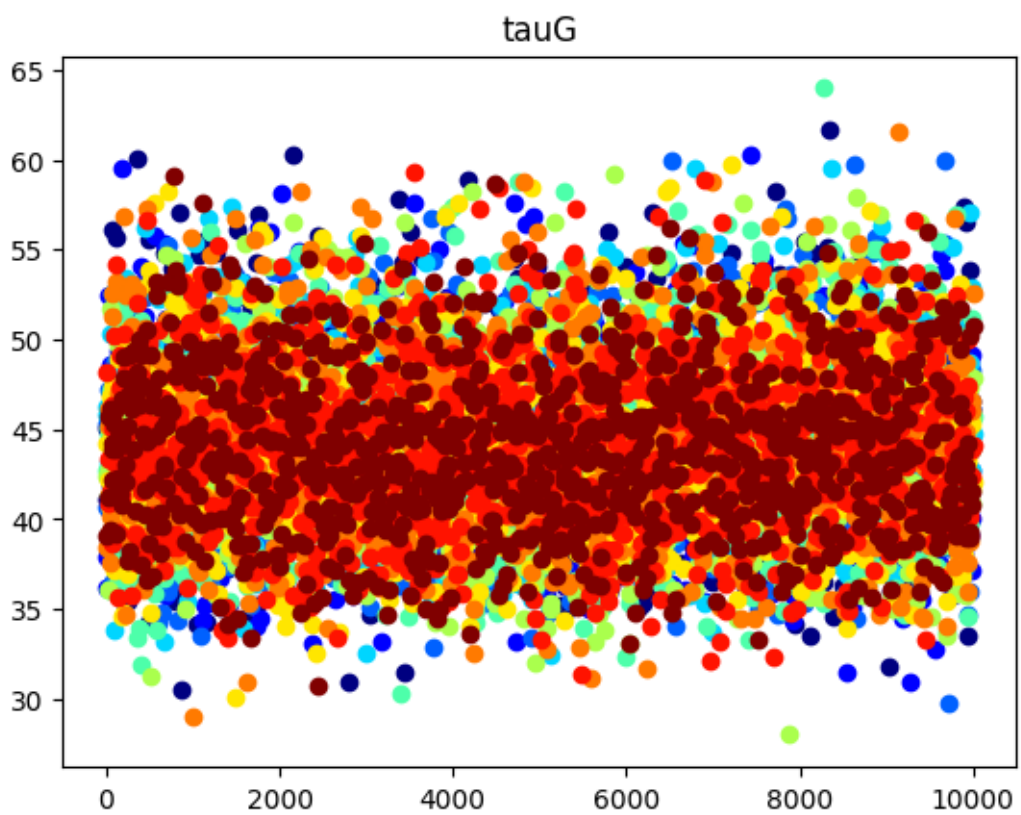


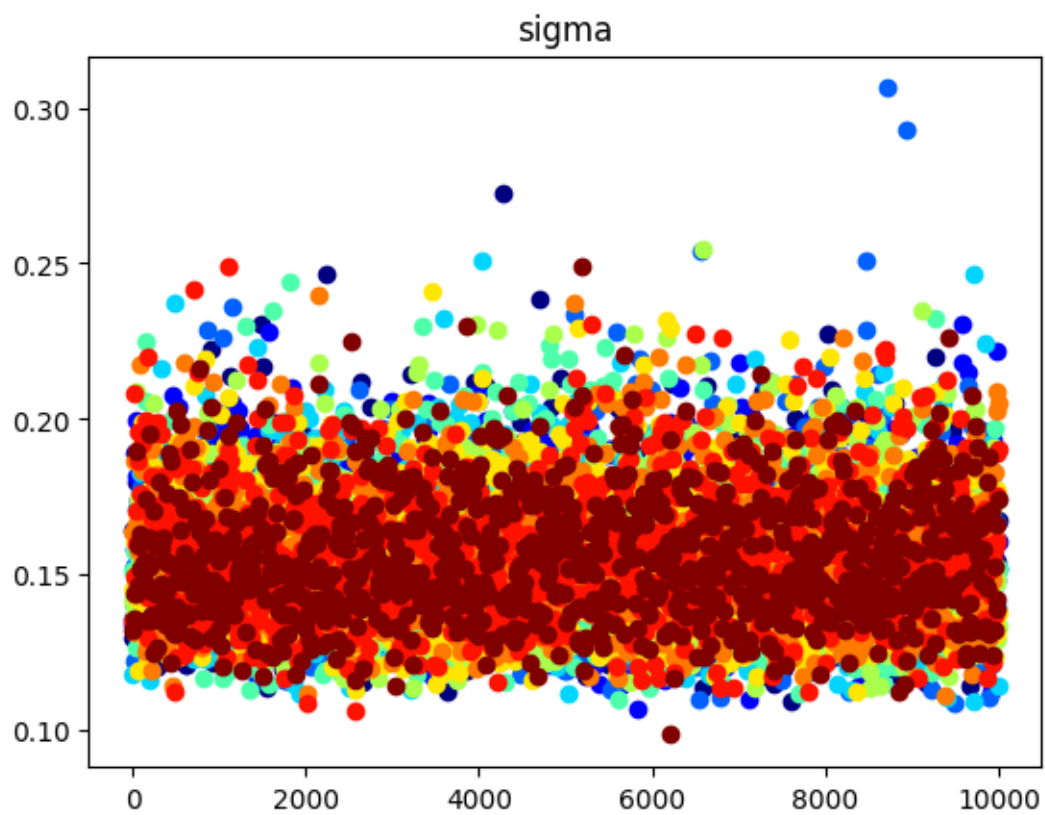
Hidden states
Fitness only, athlete: 1

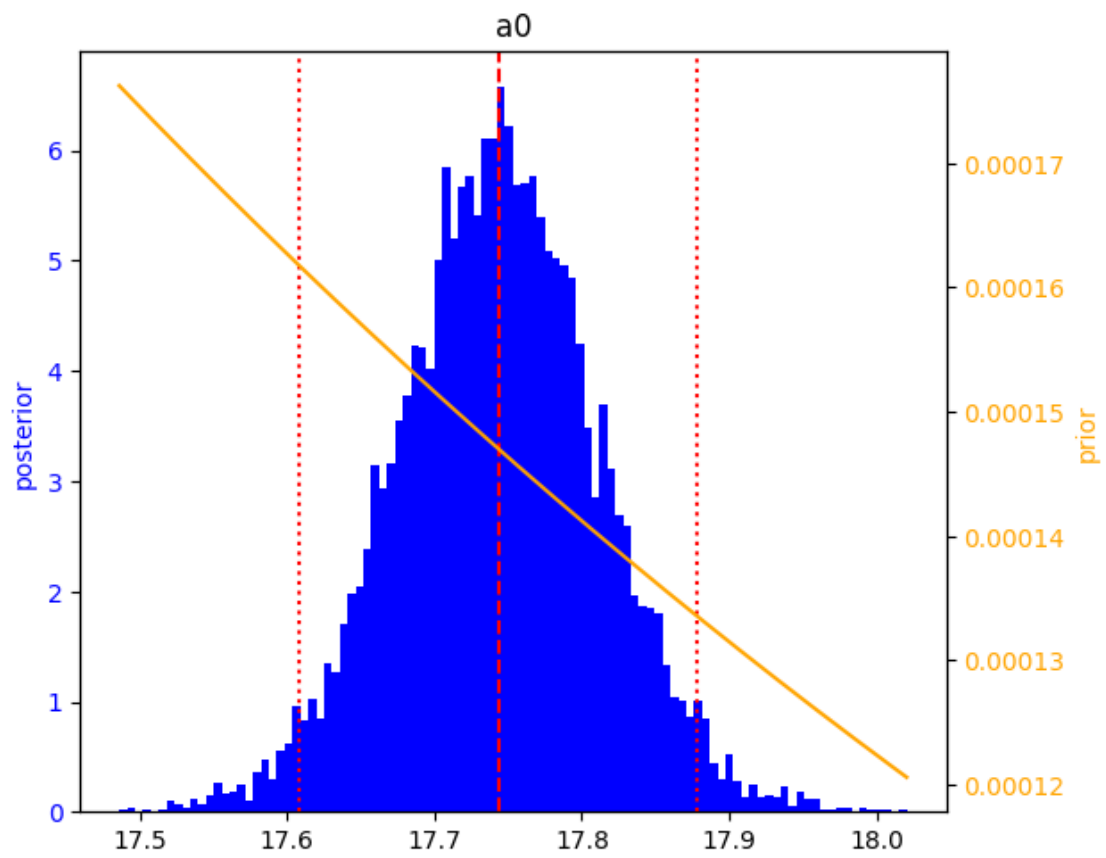


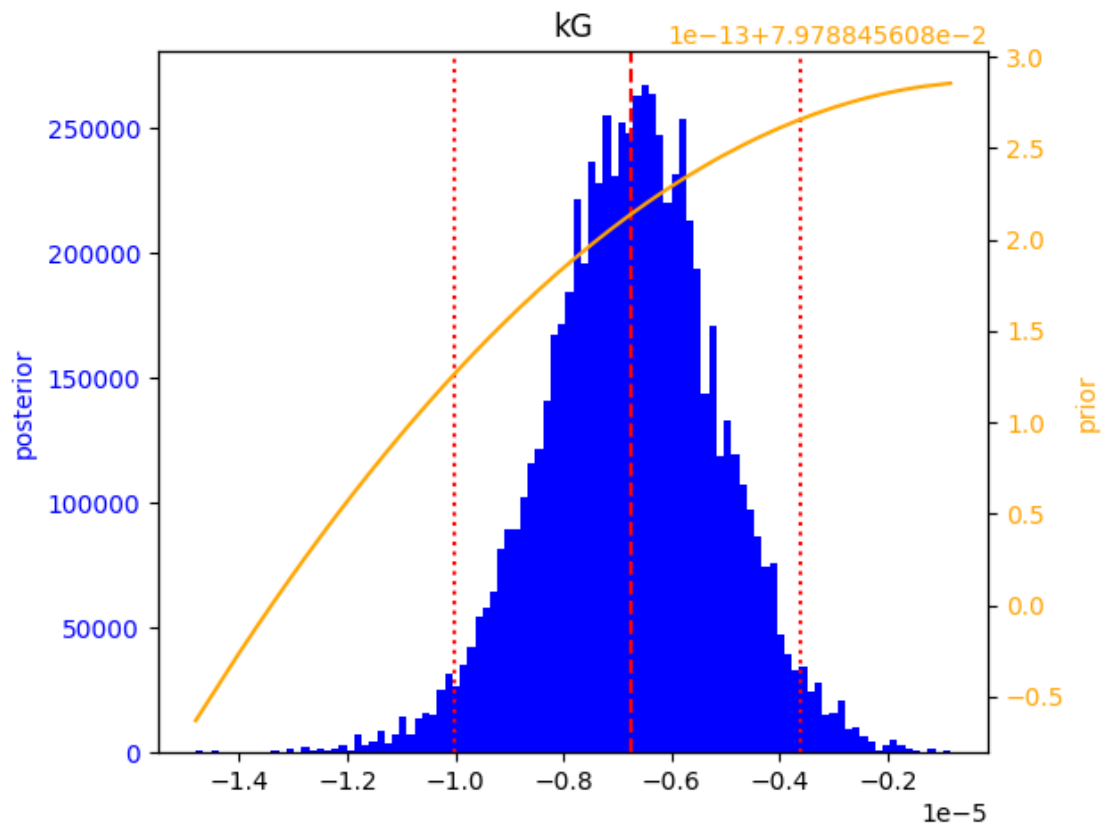


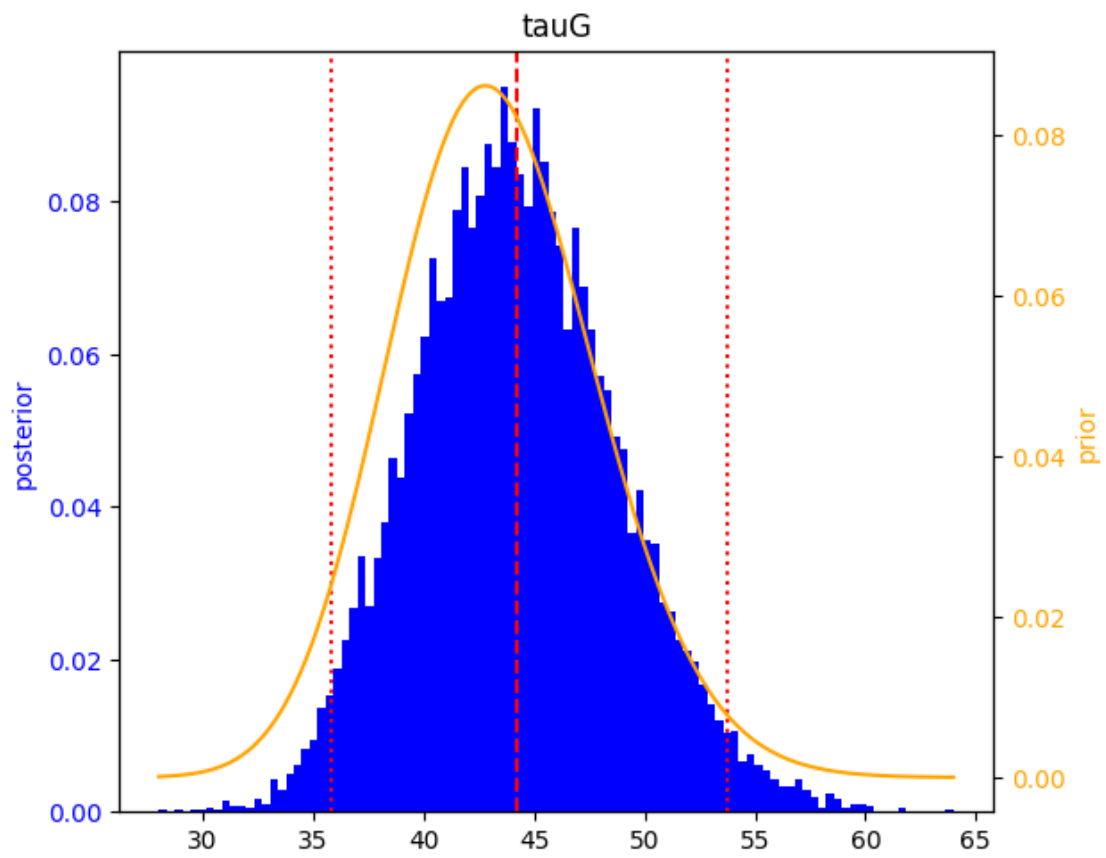


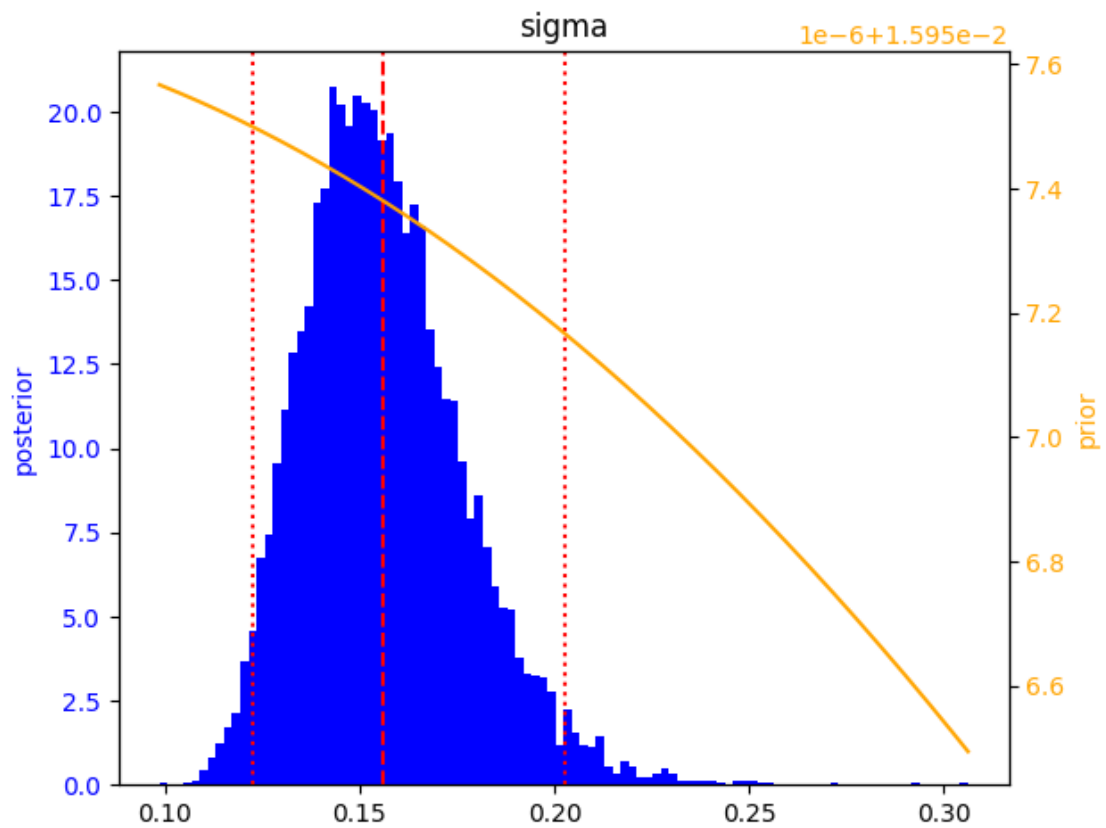


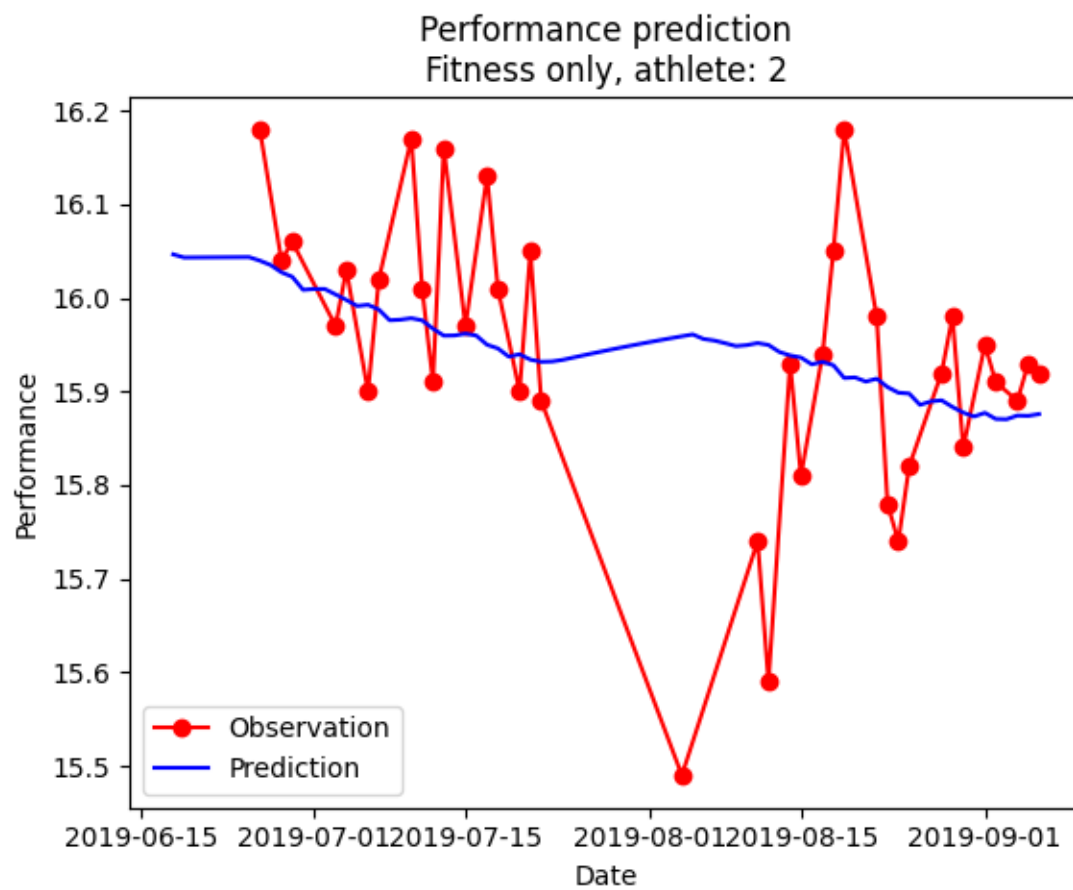


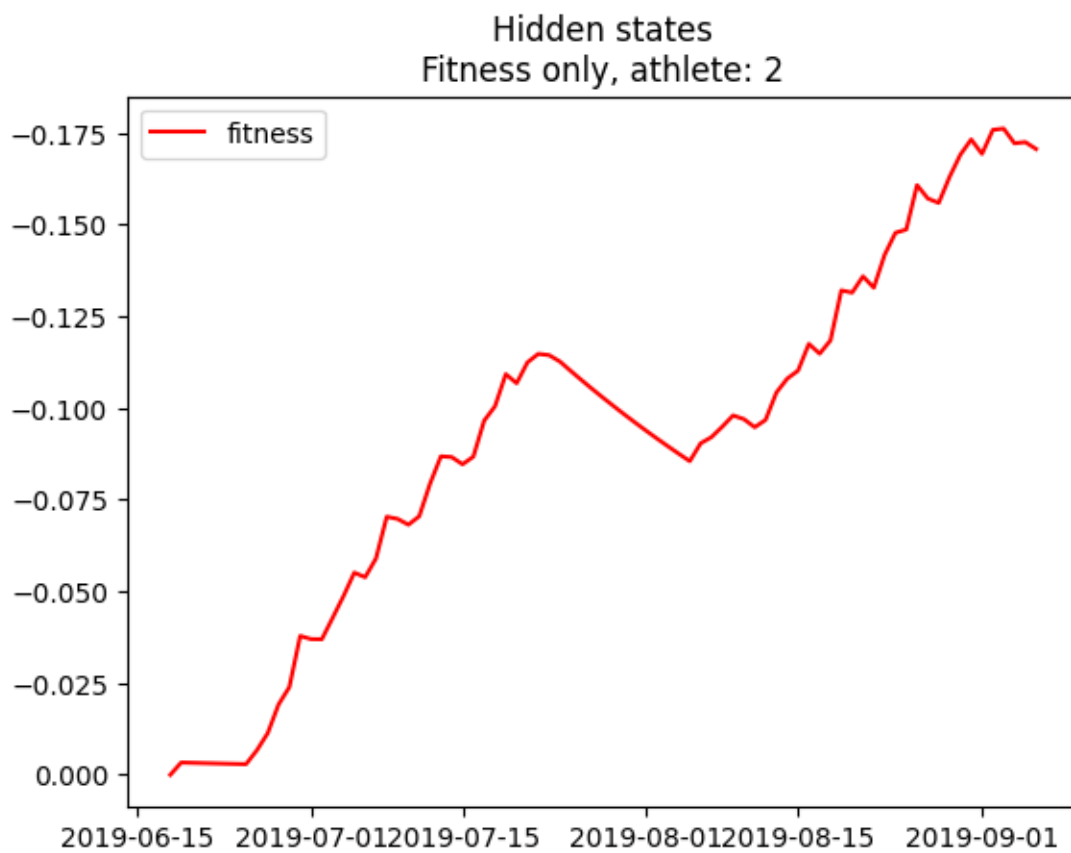


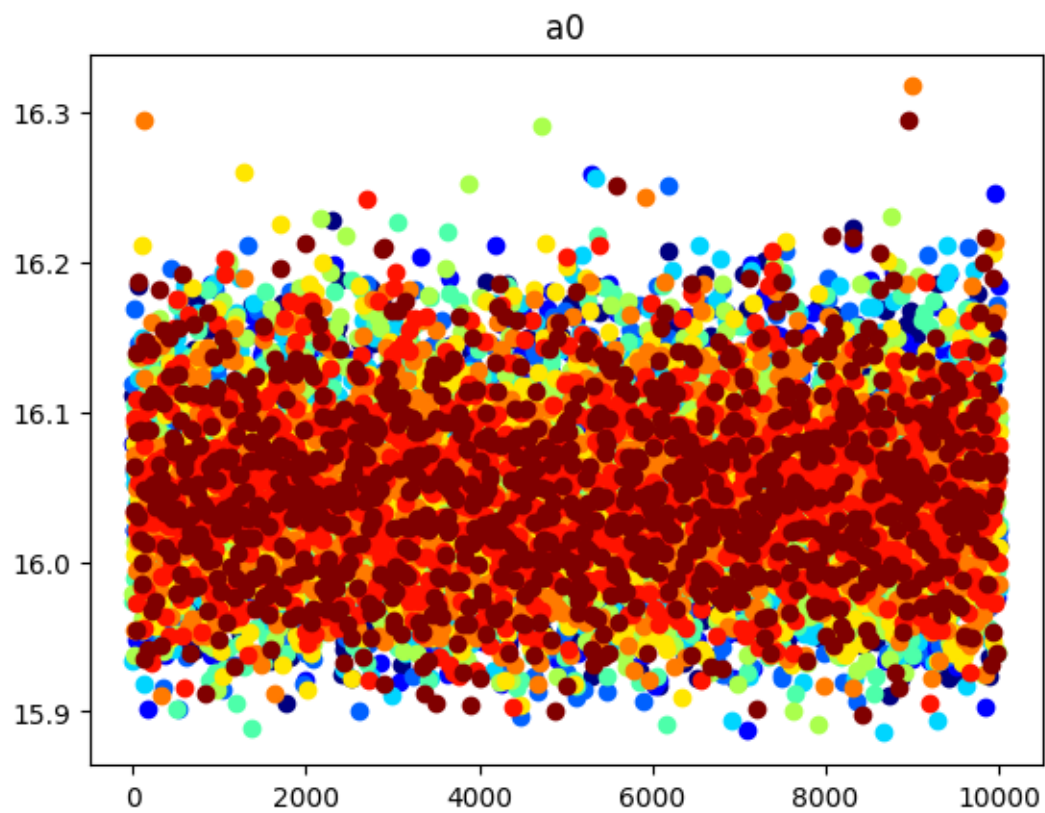


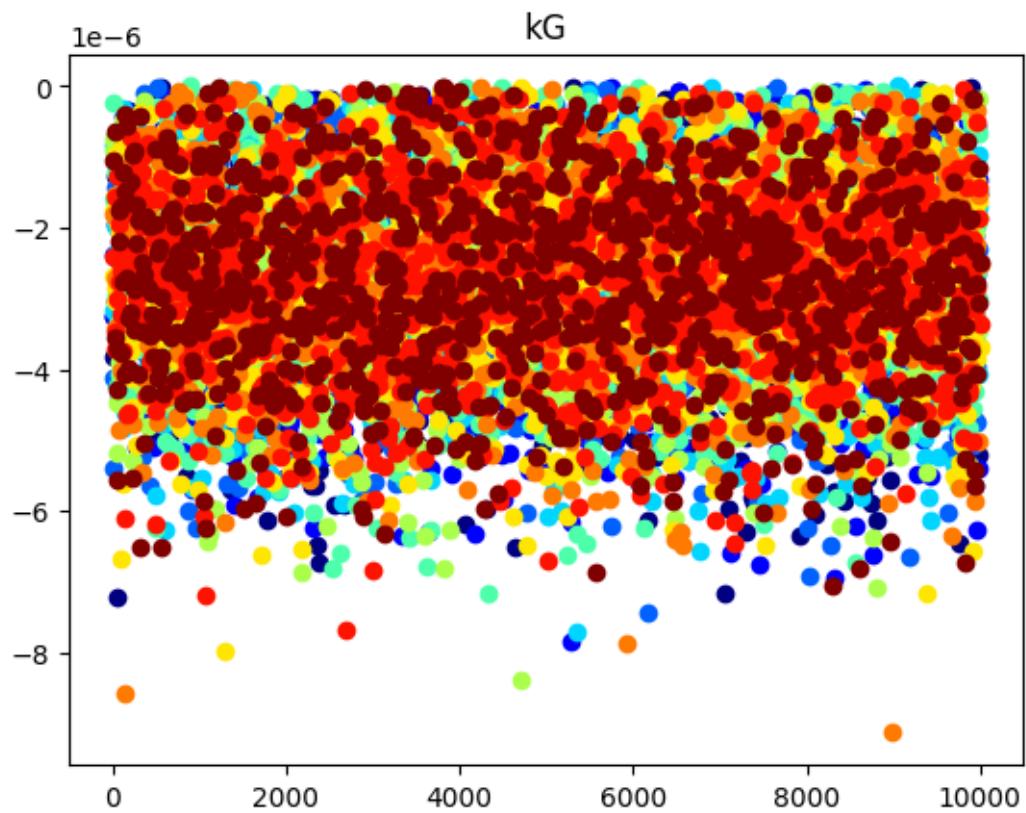


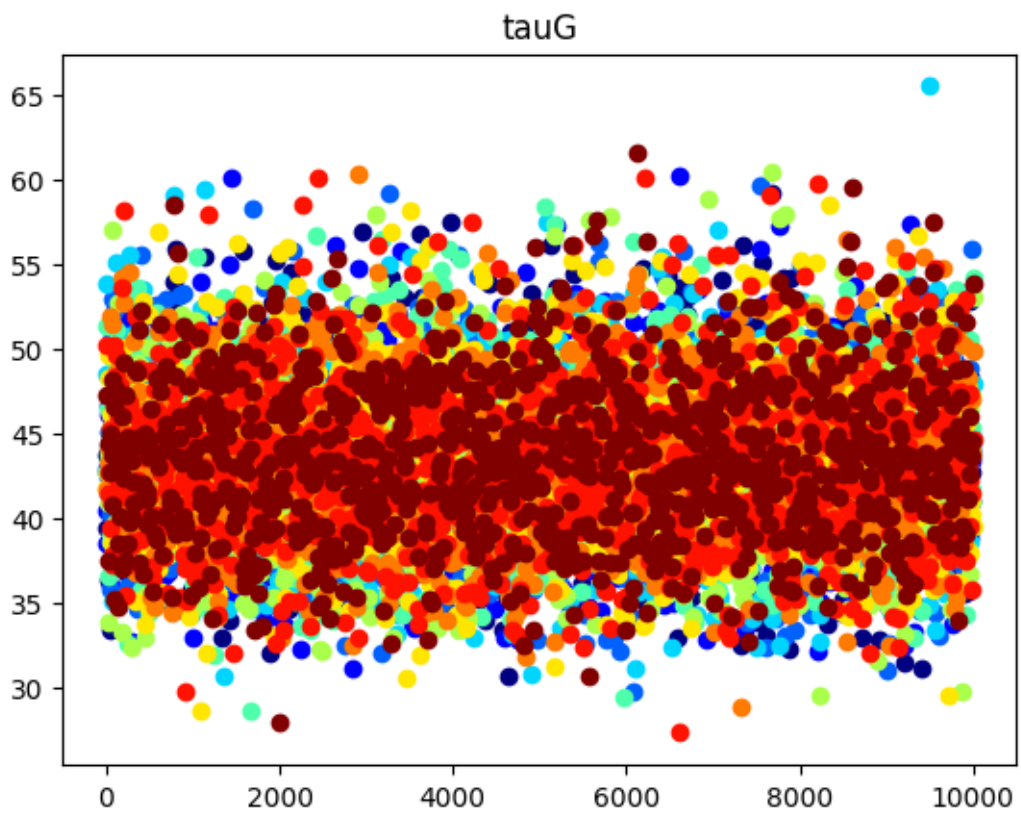


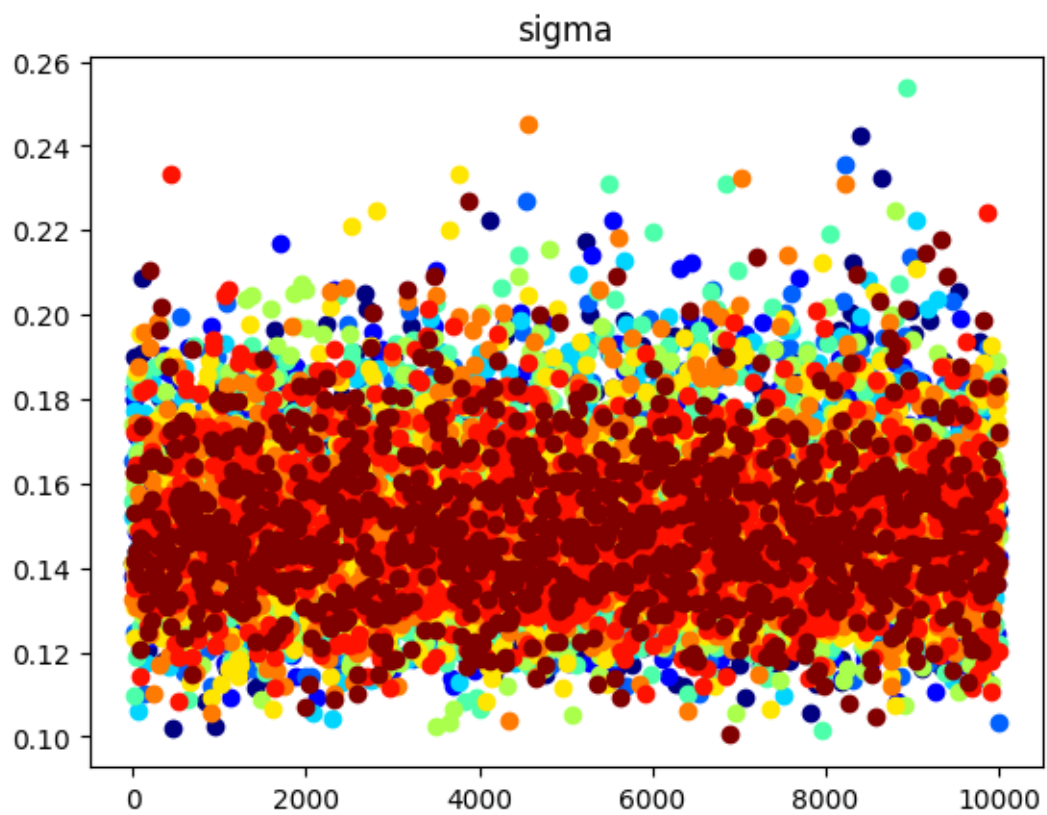


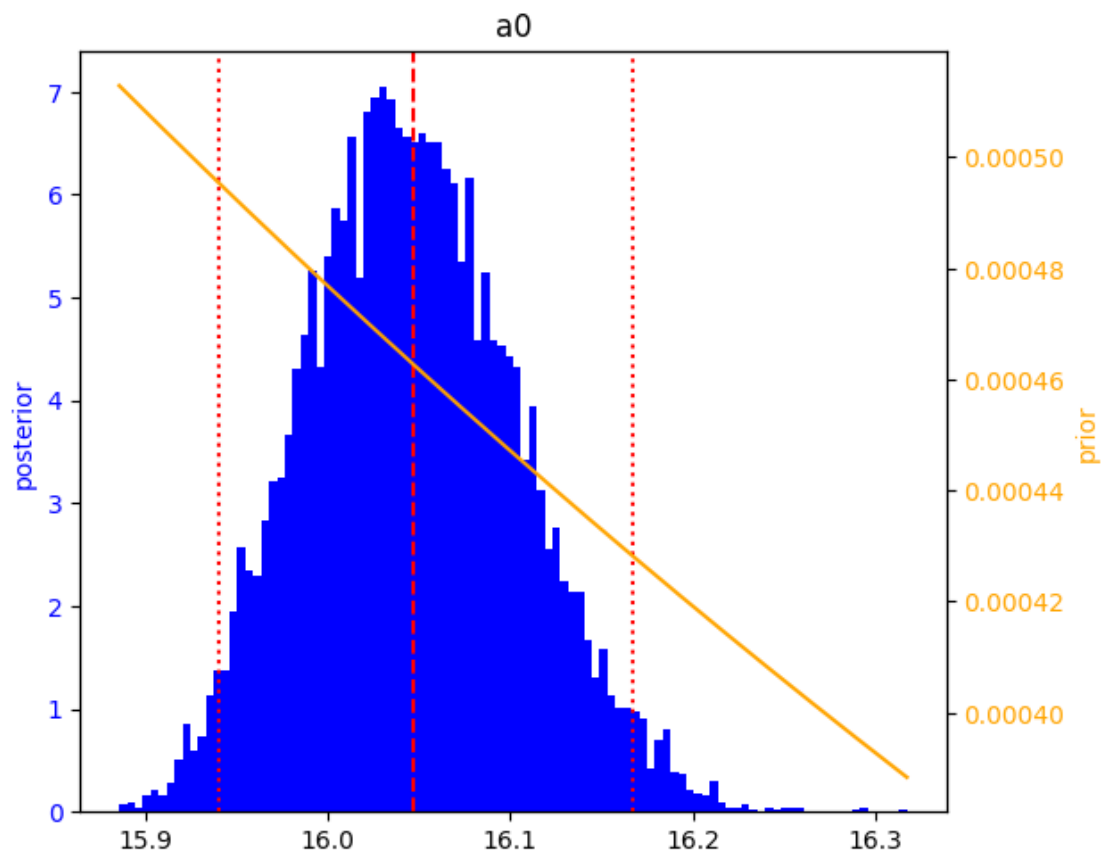


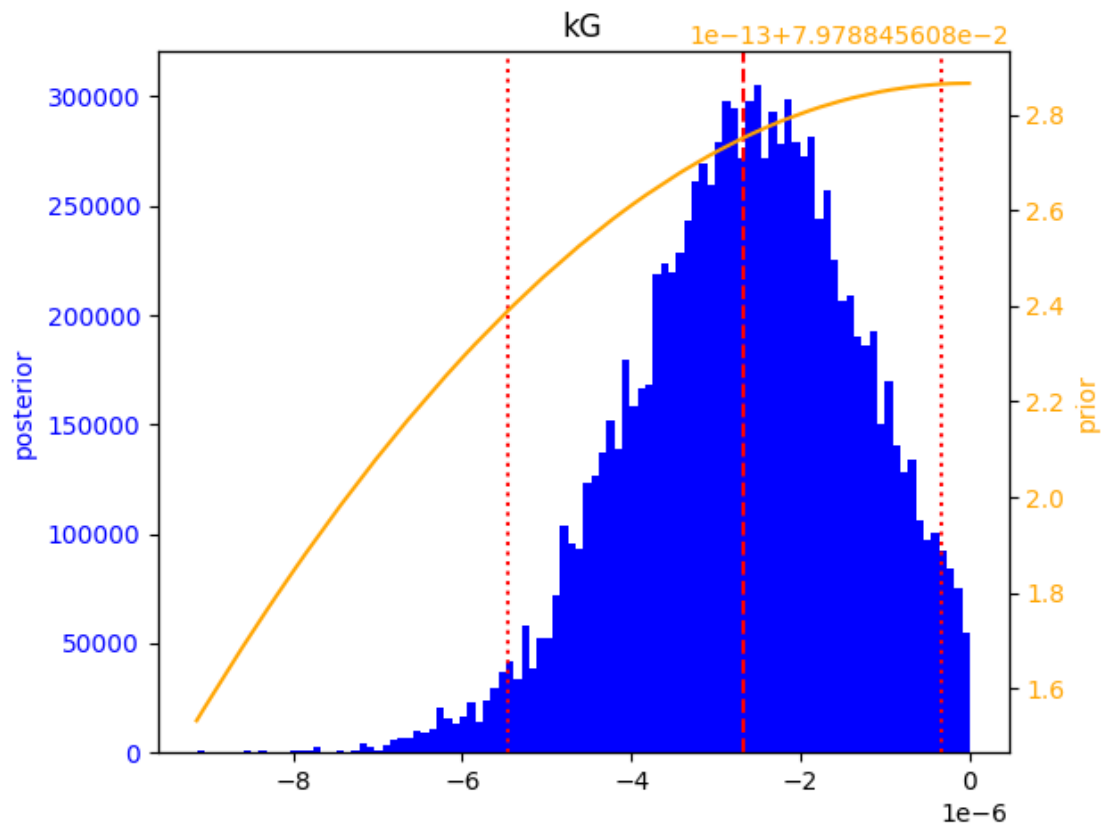


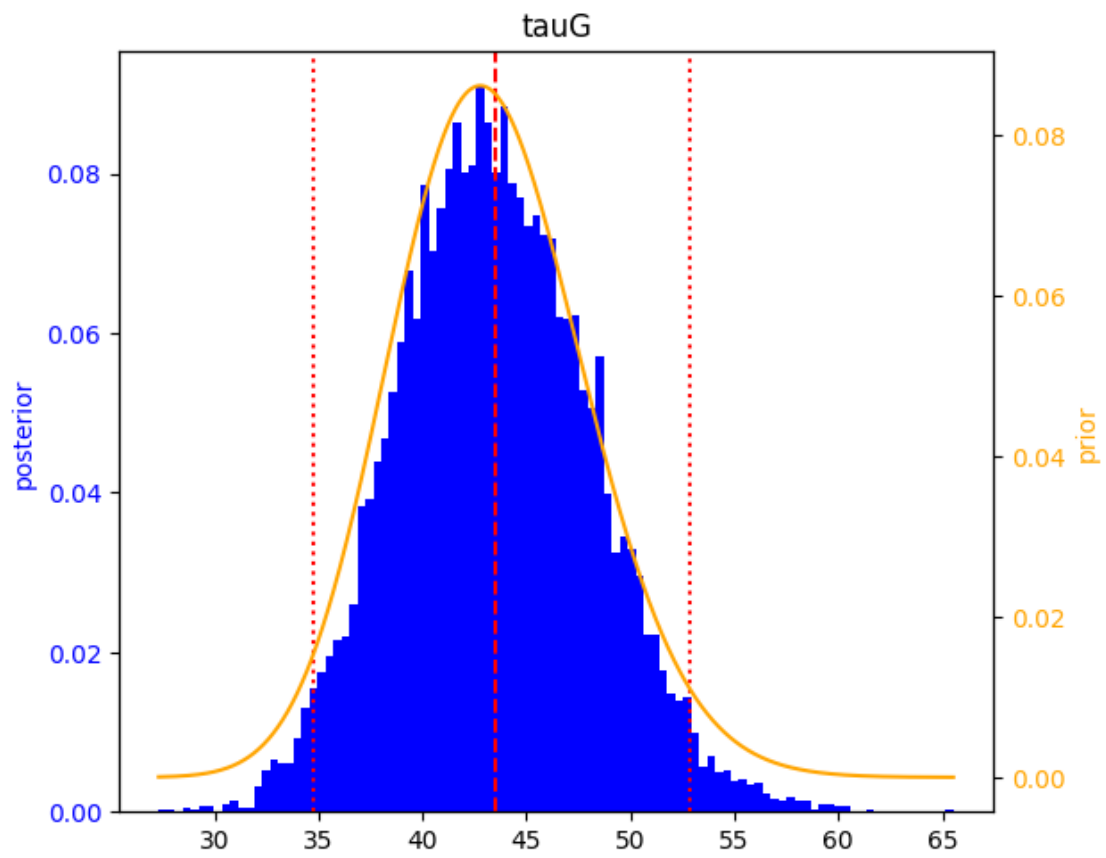


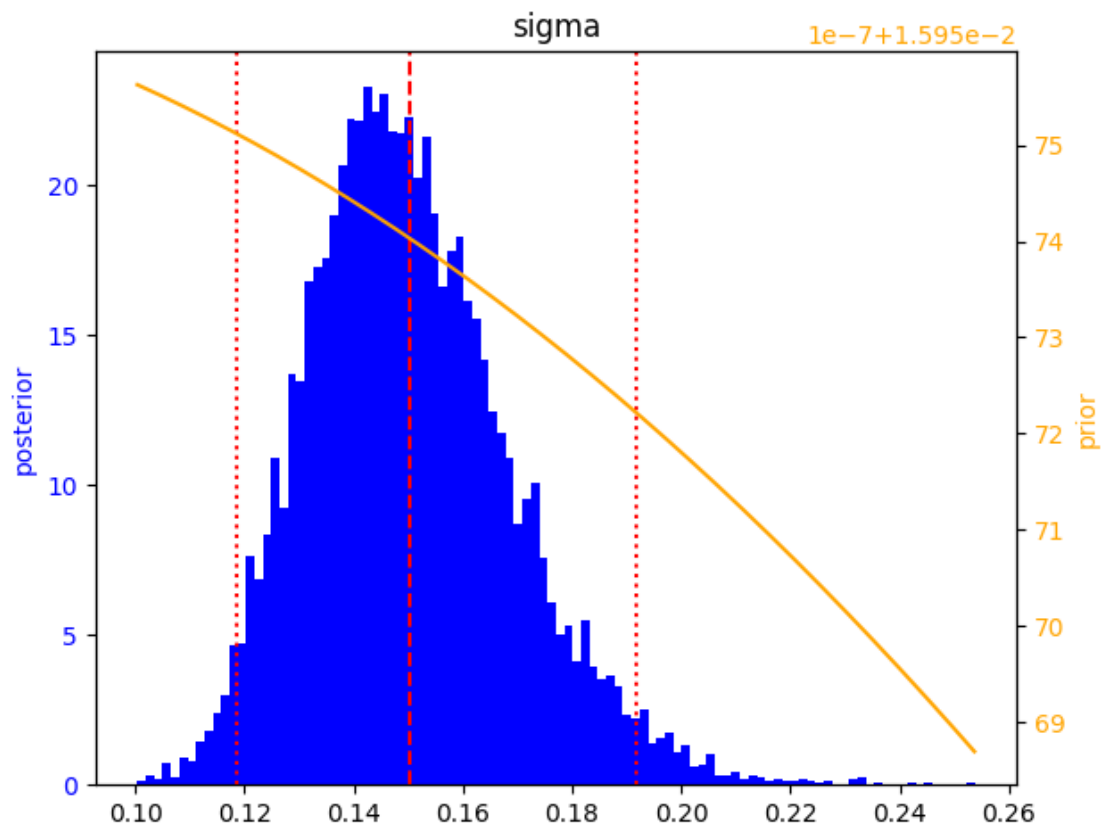


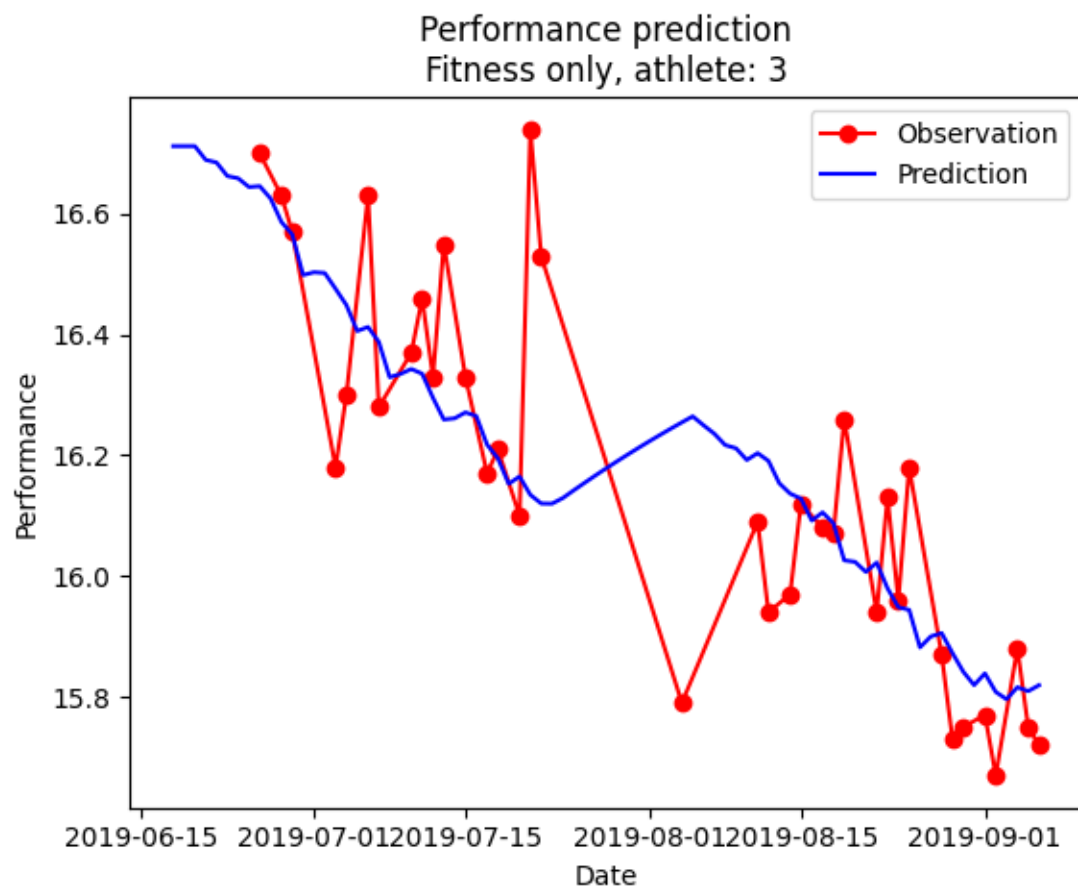


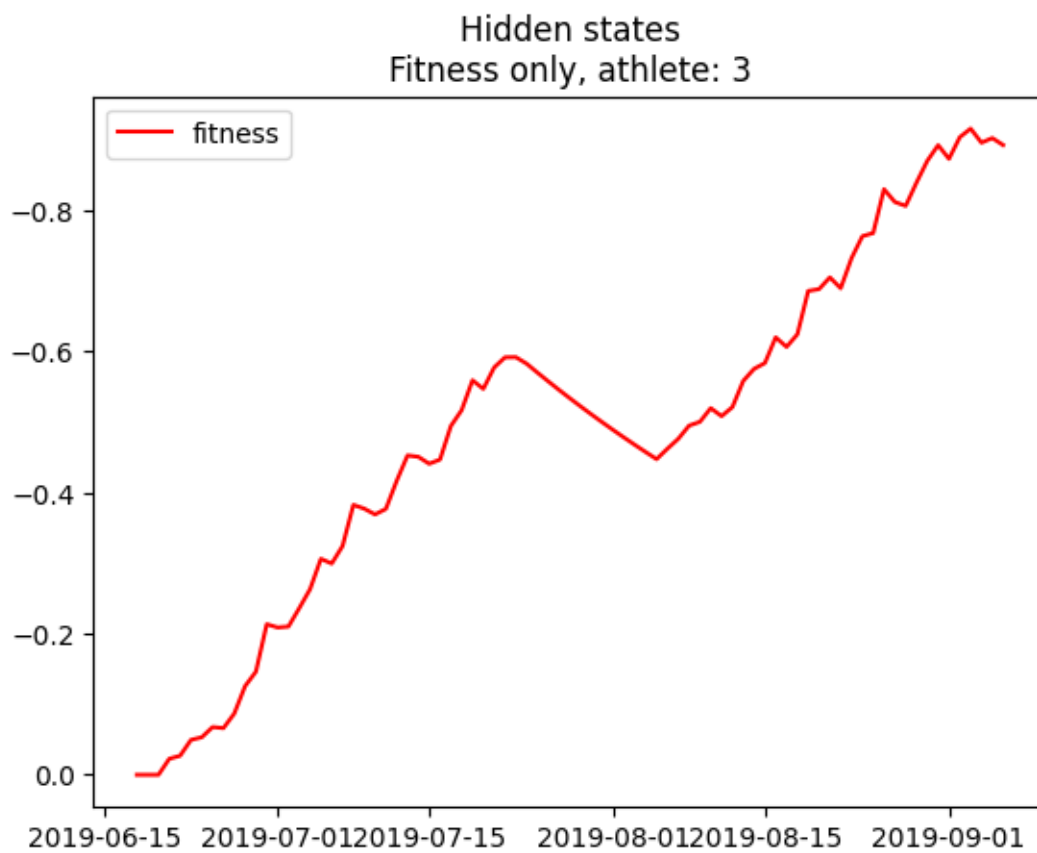


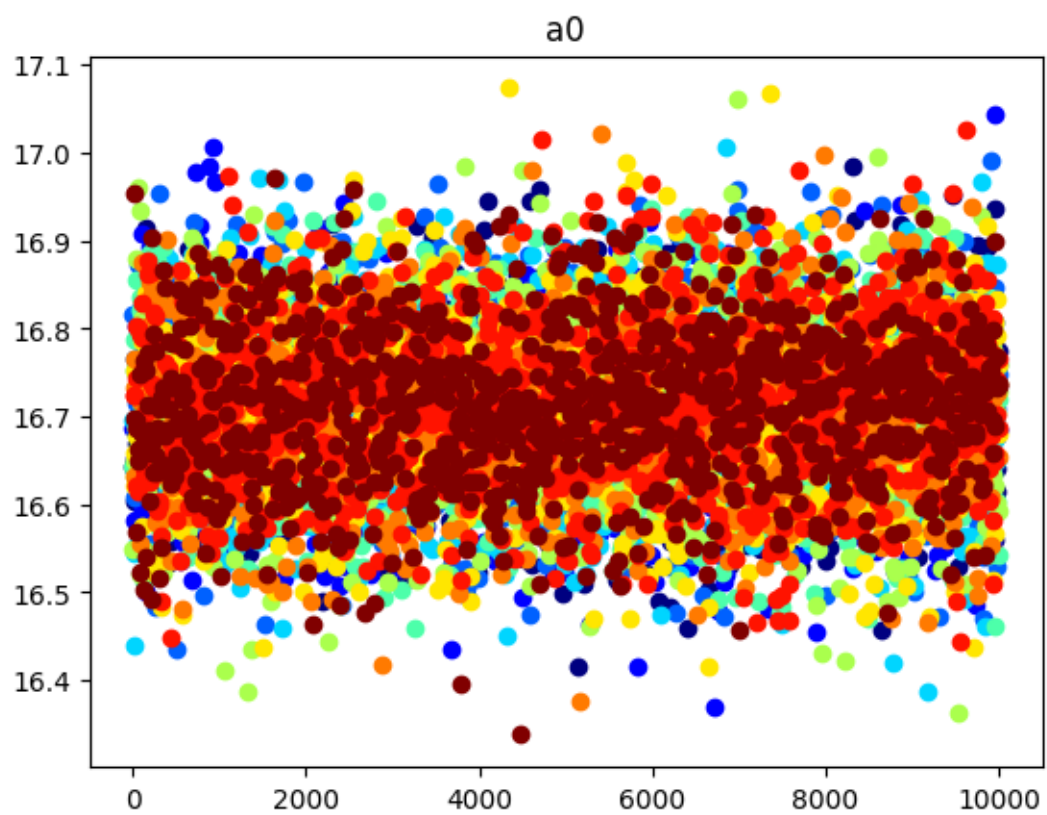


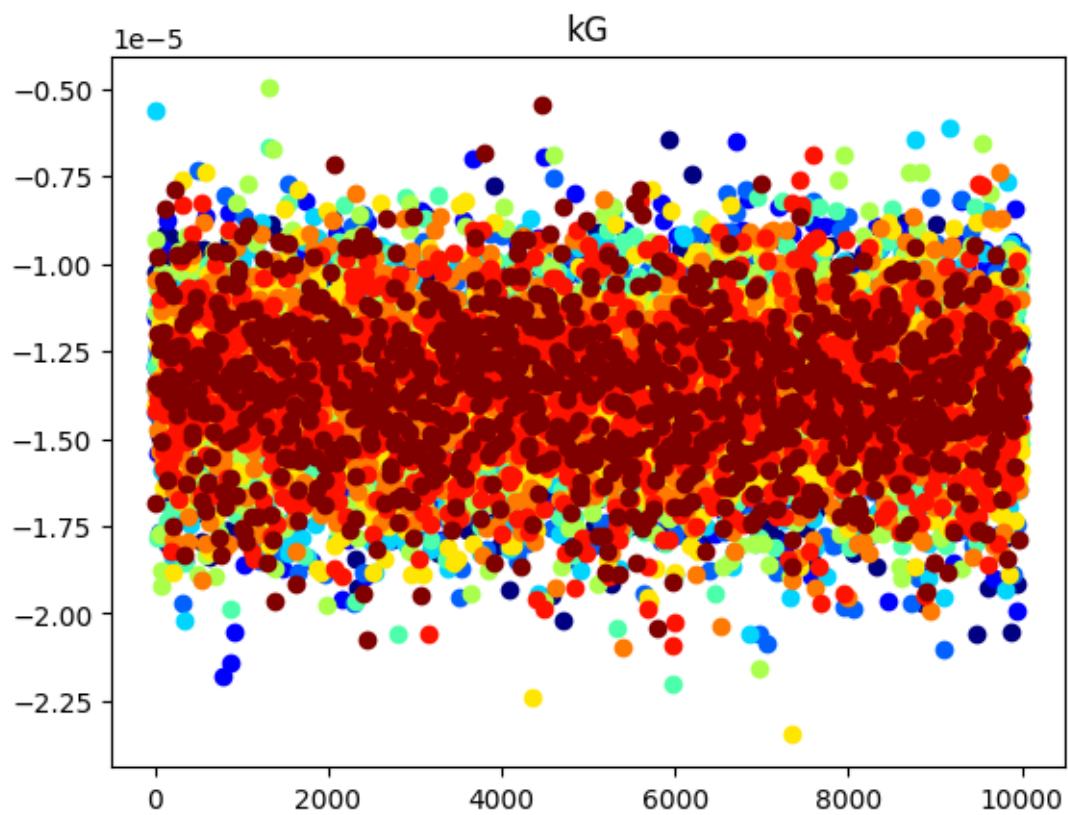


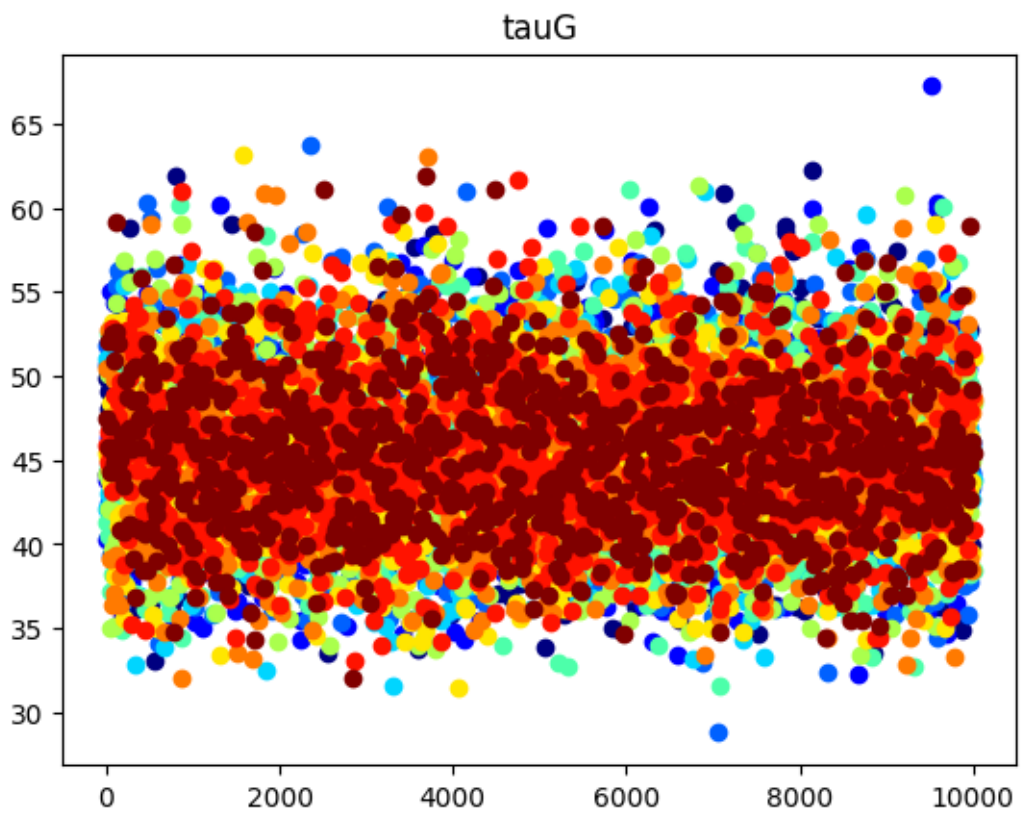


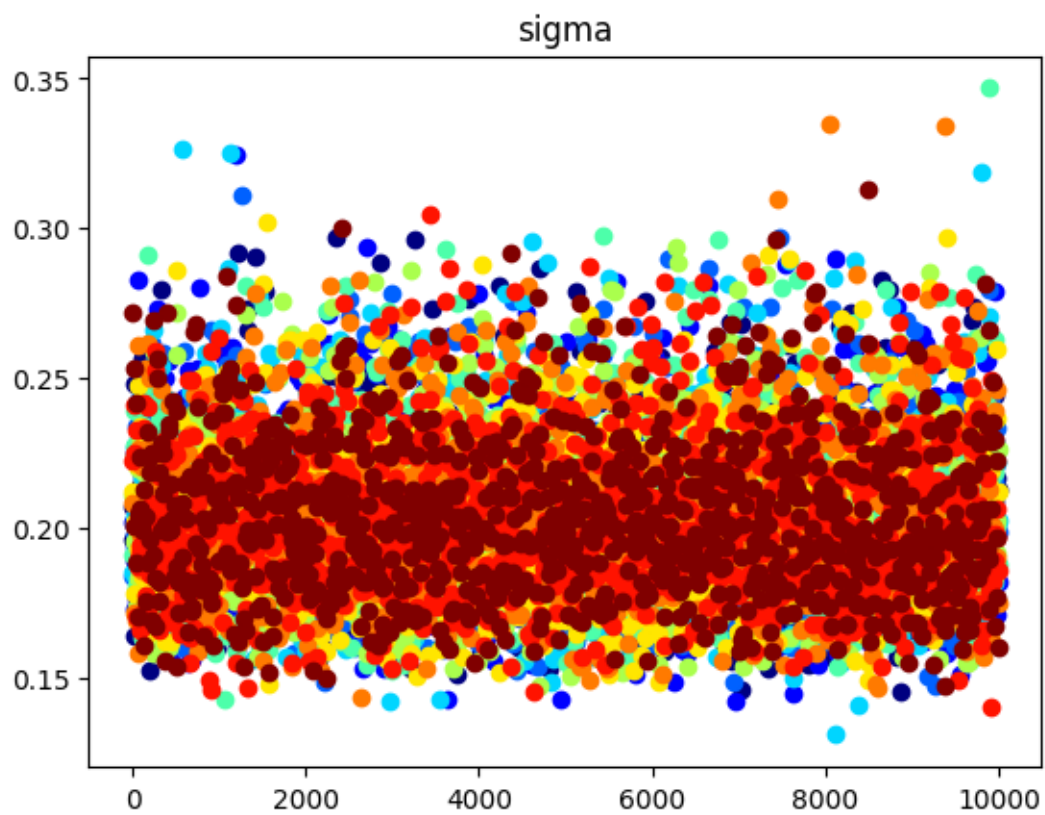


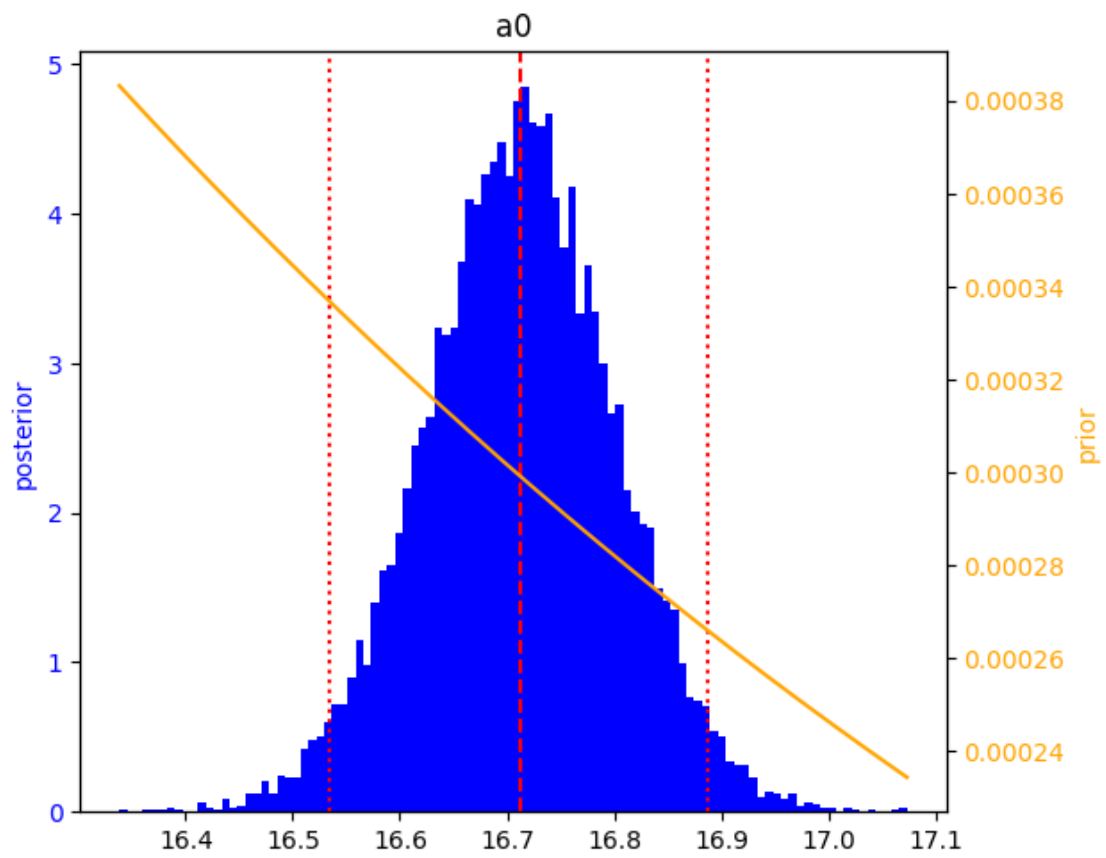


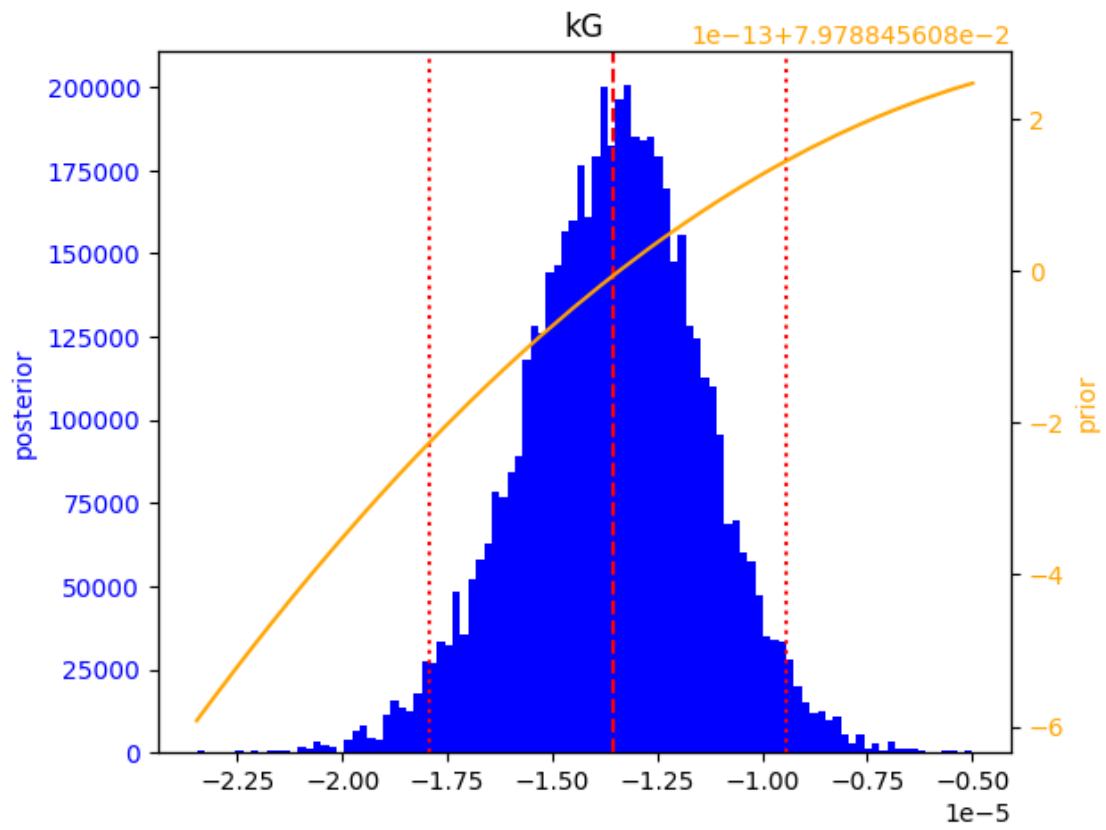


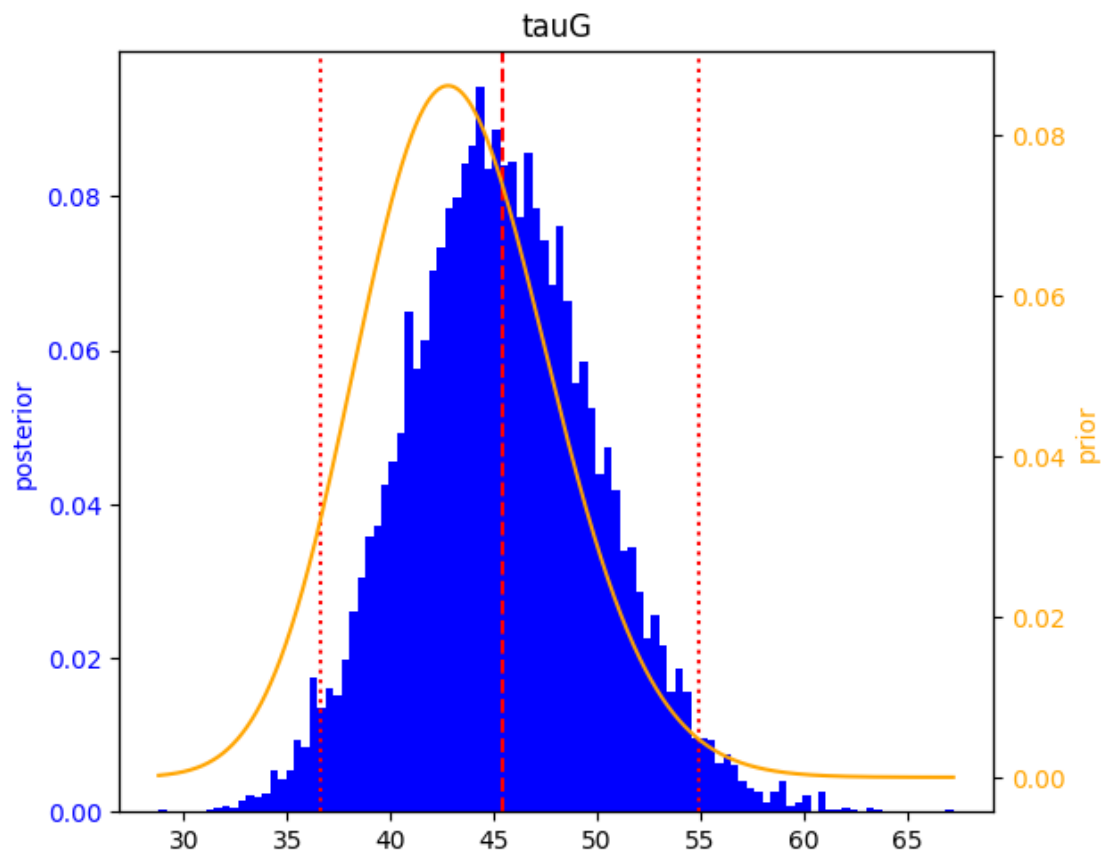


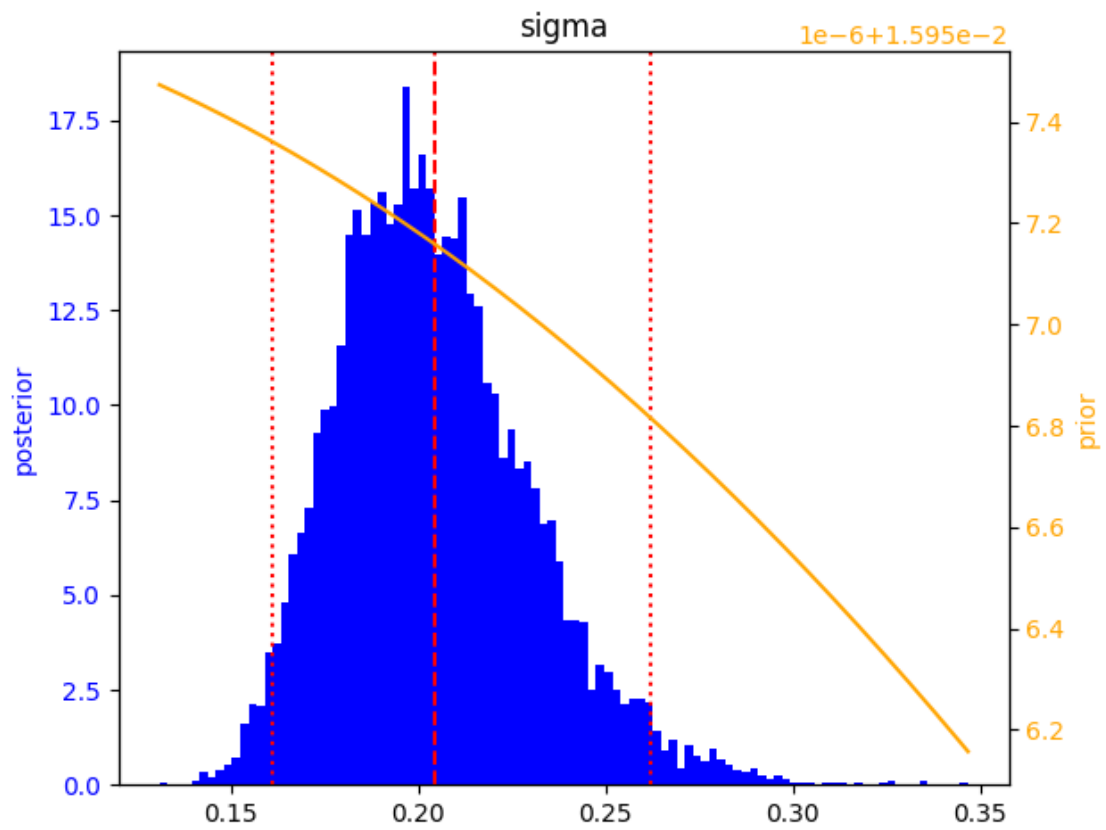


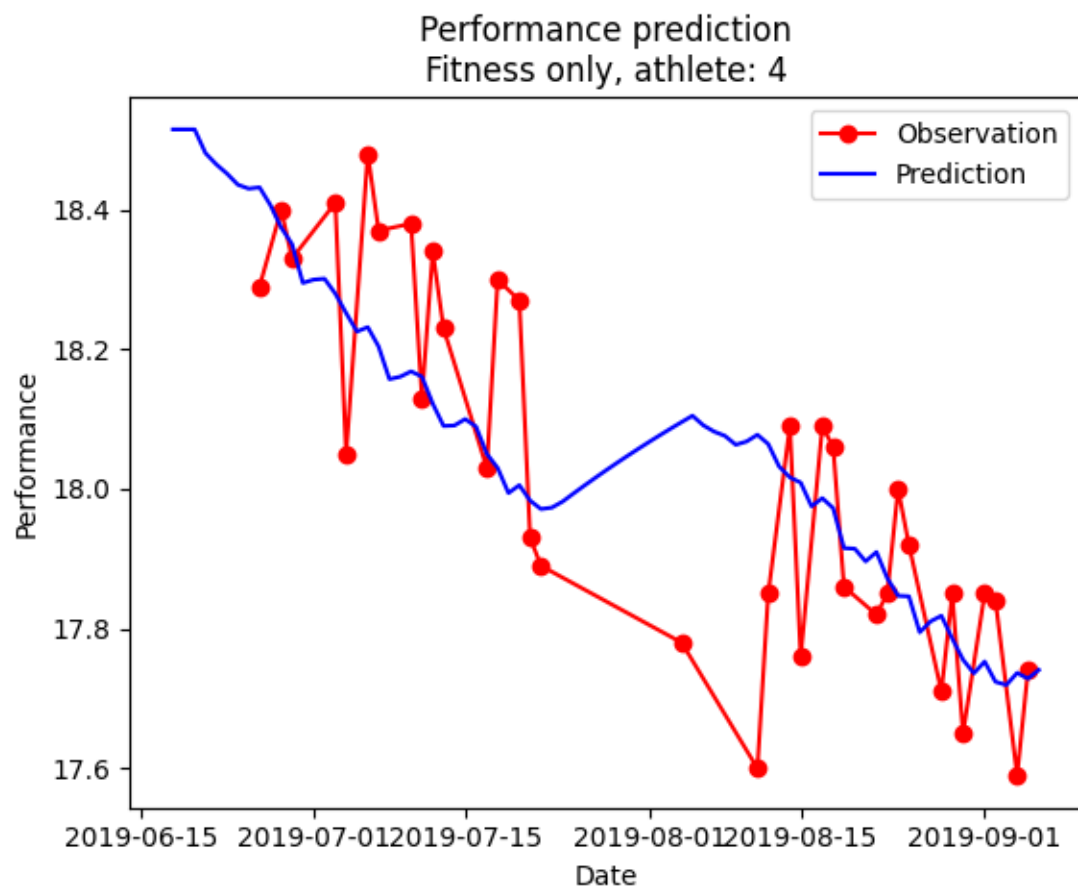


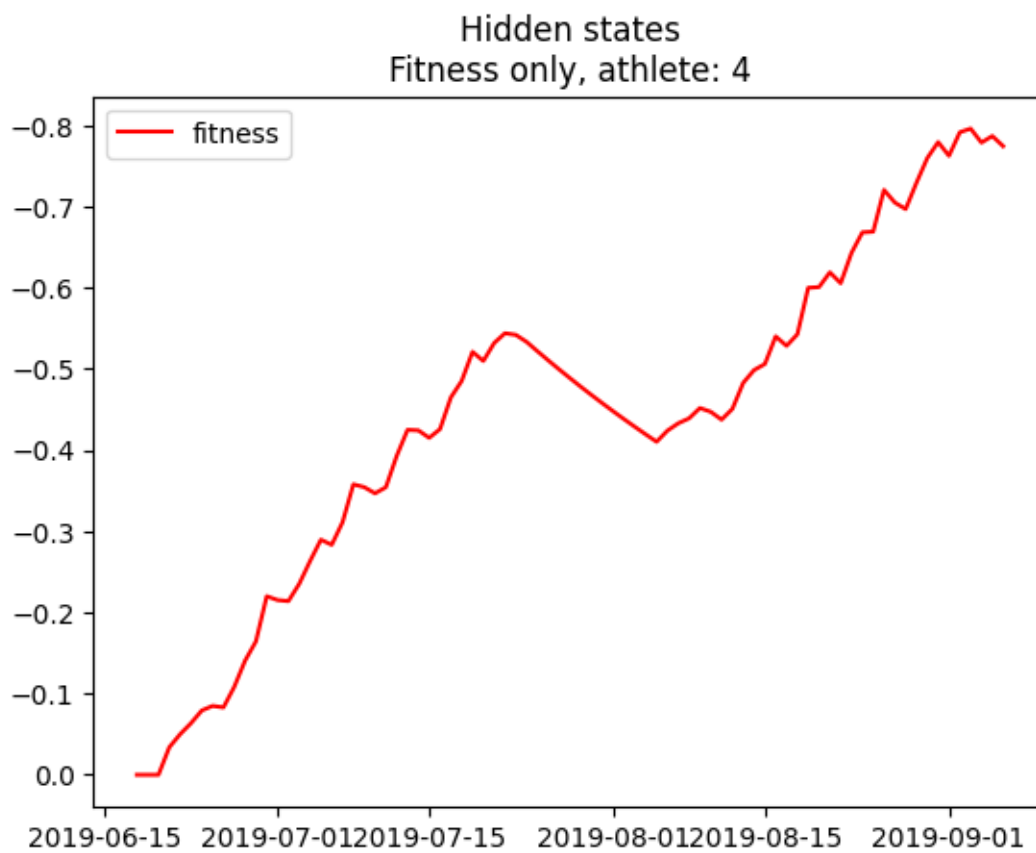


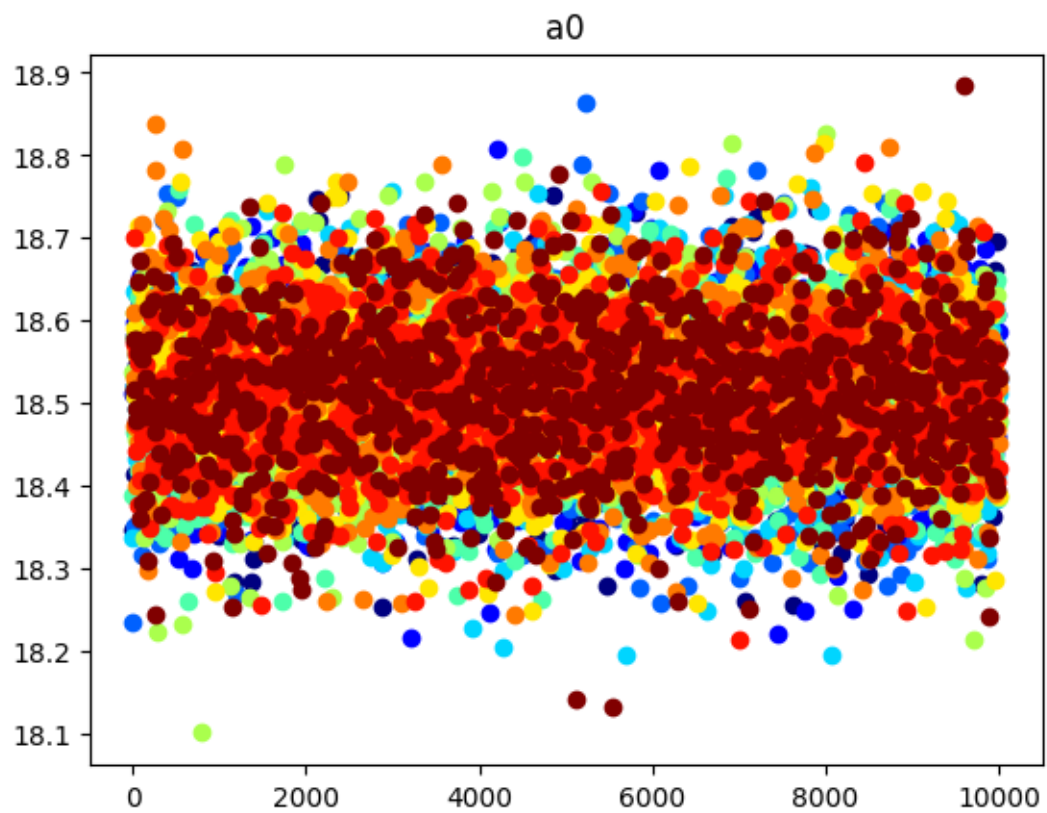


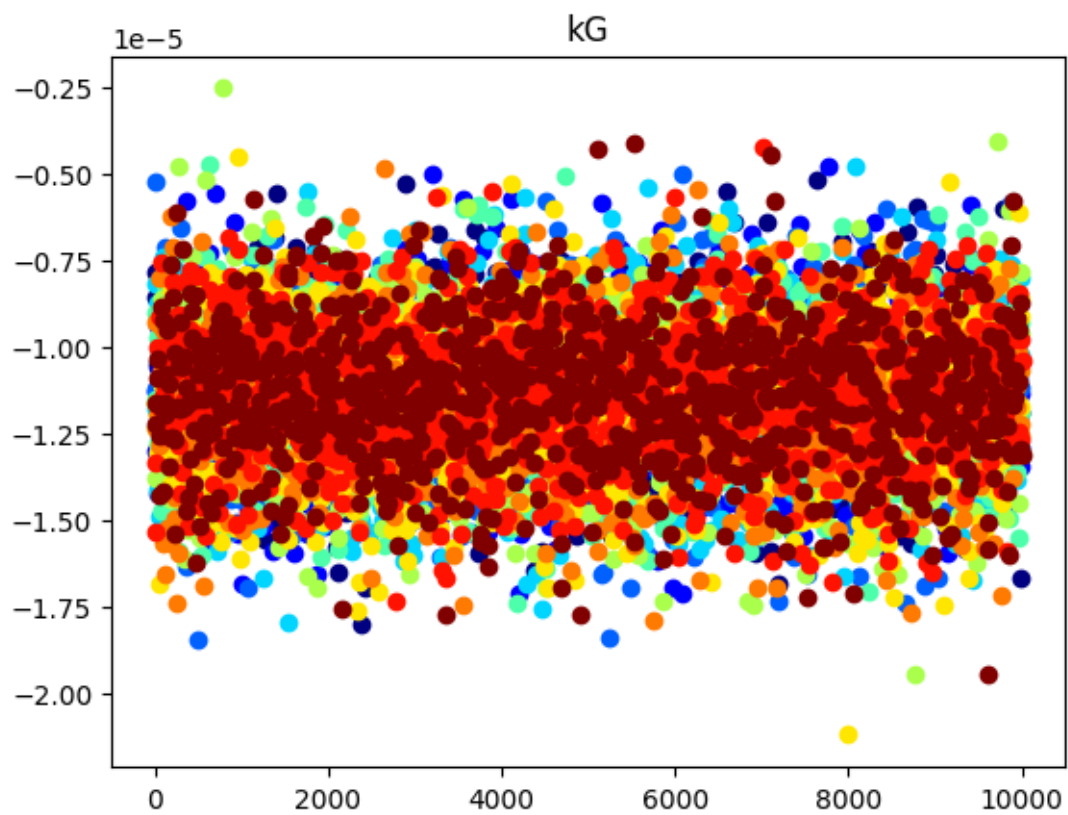


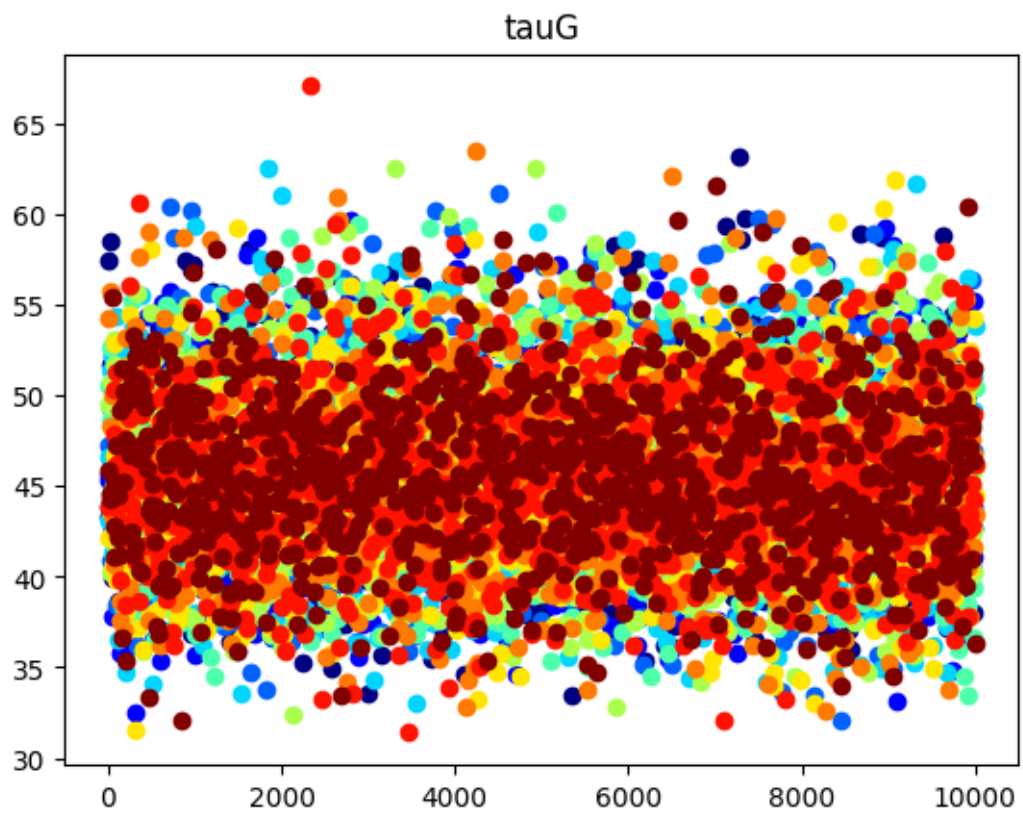


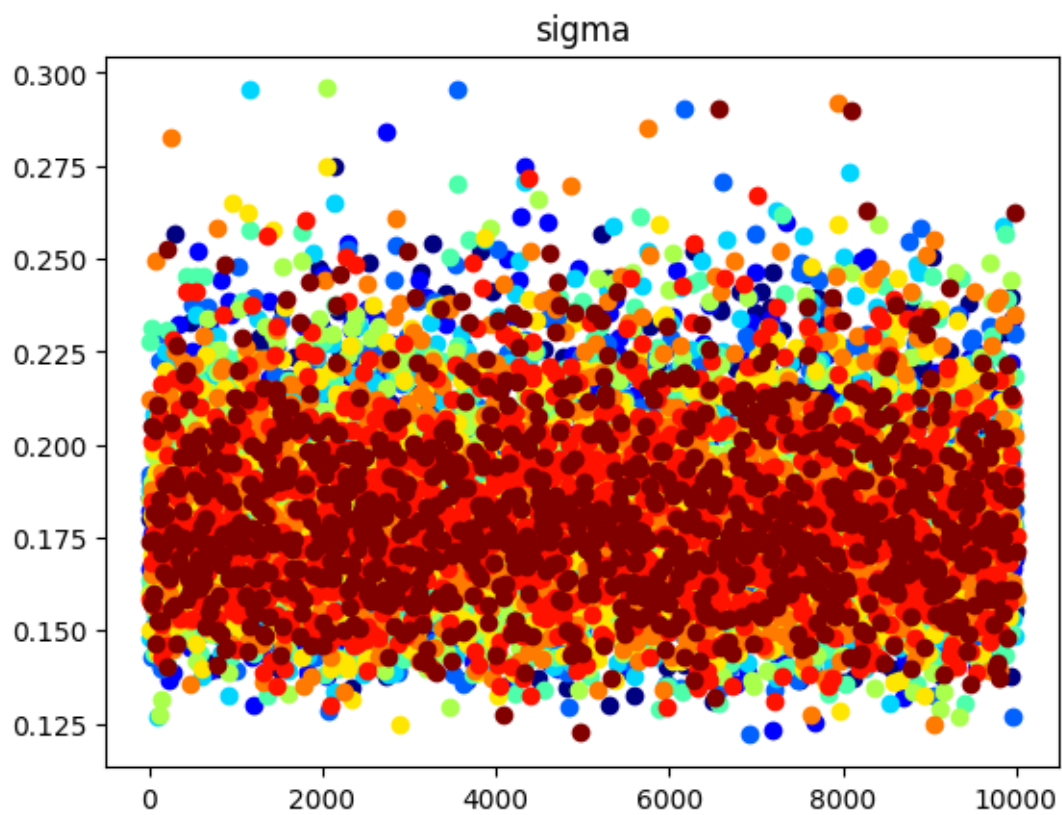


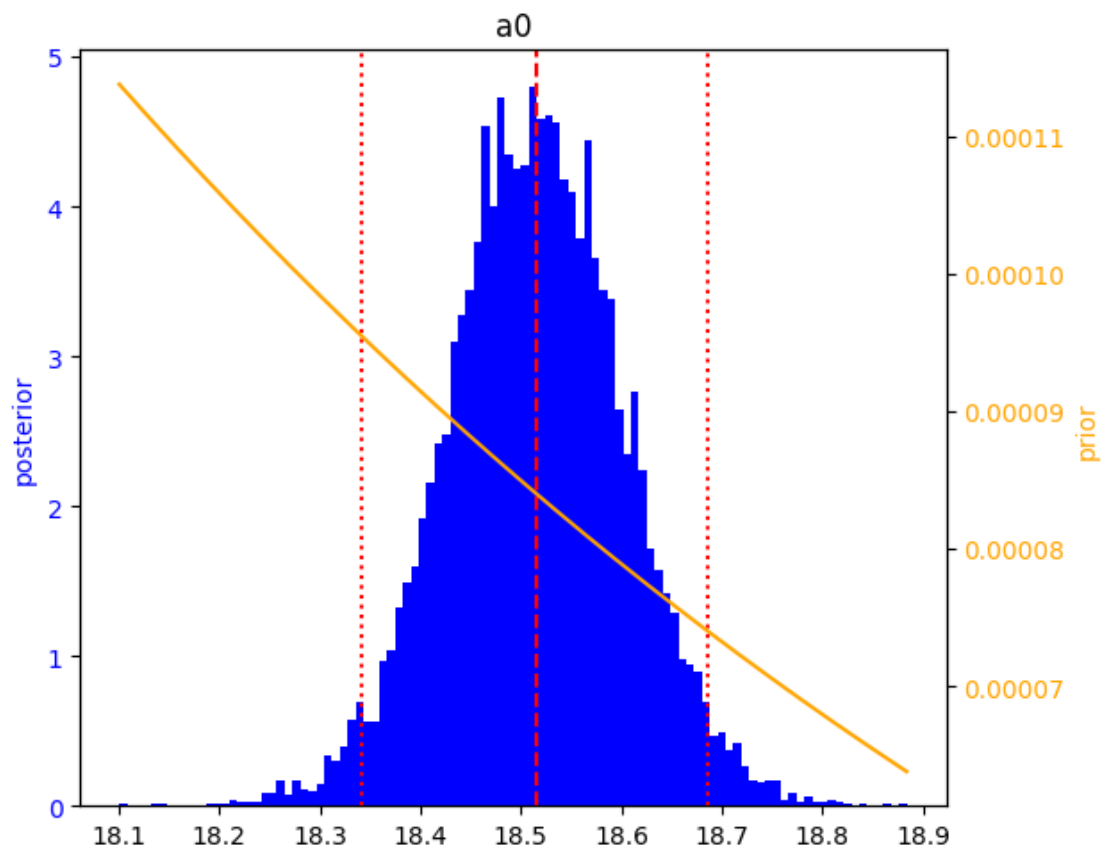


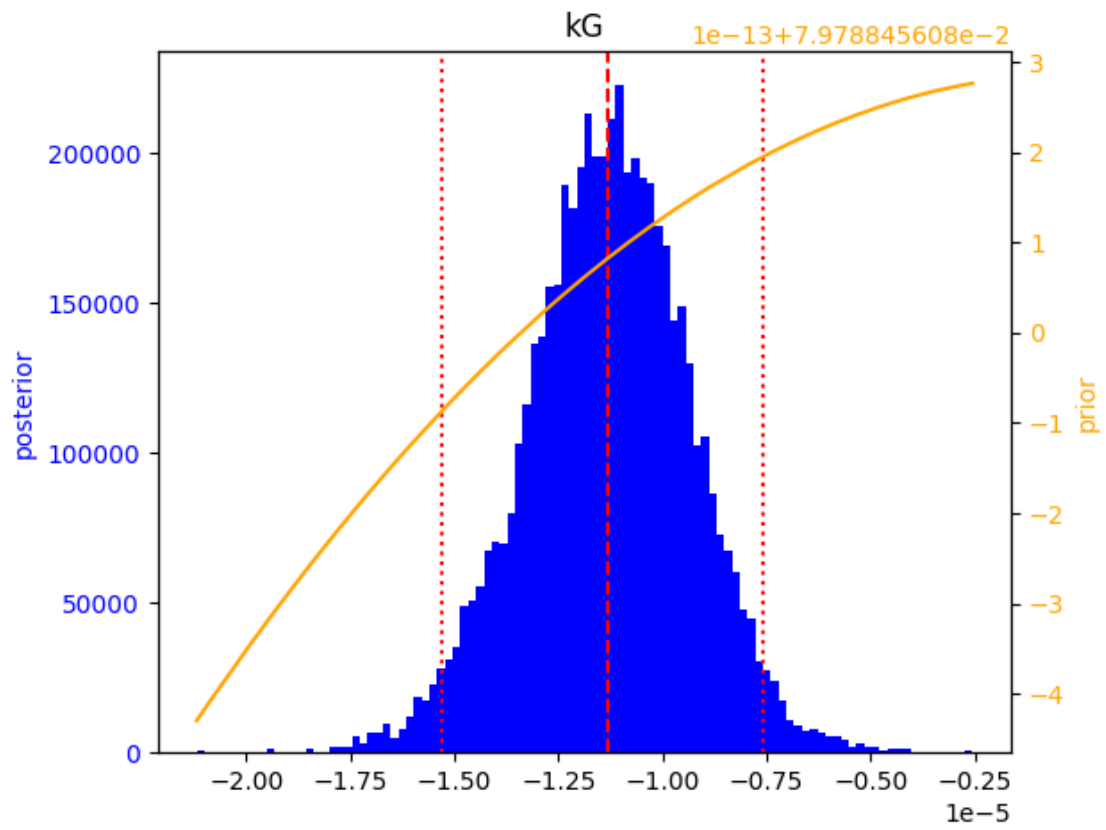


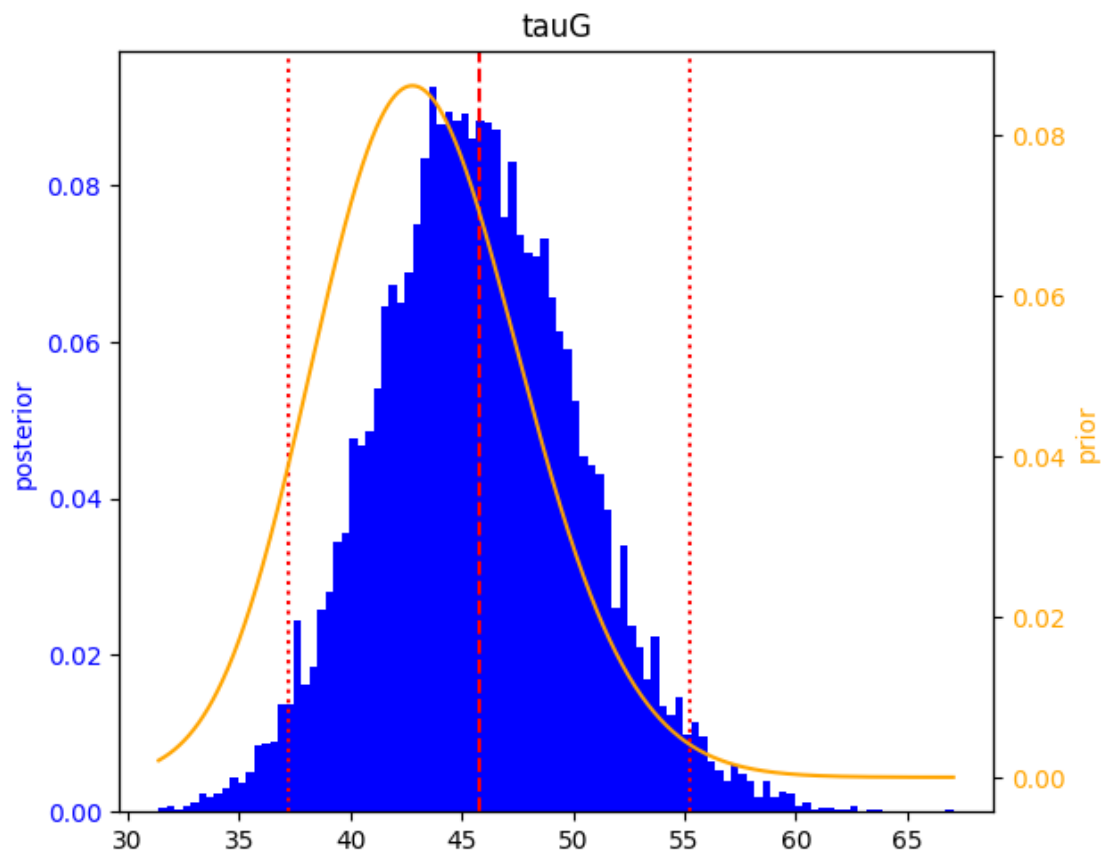


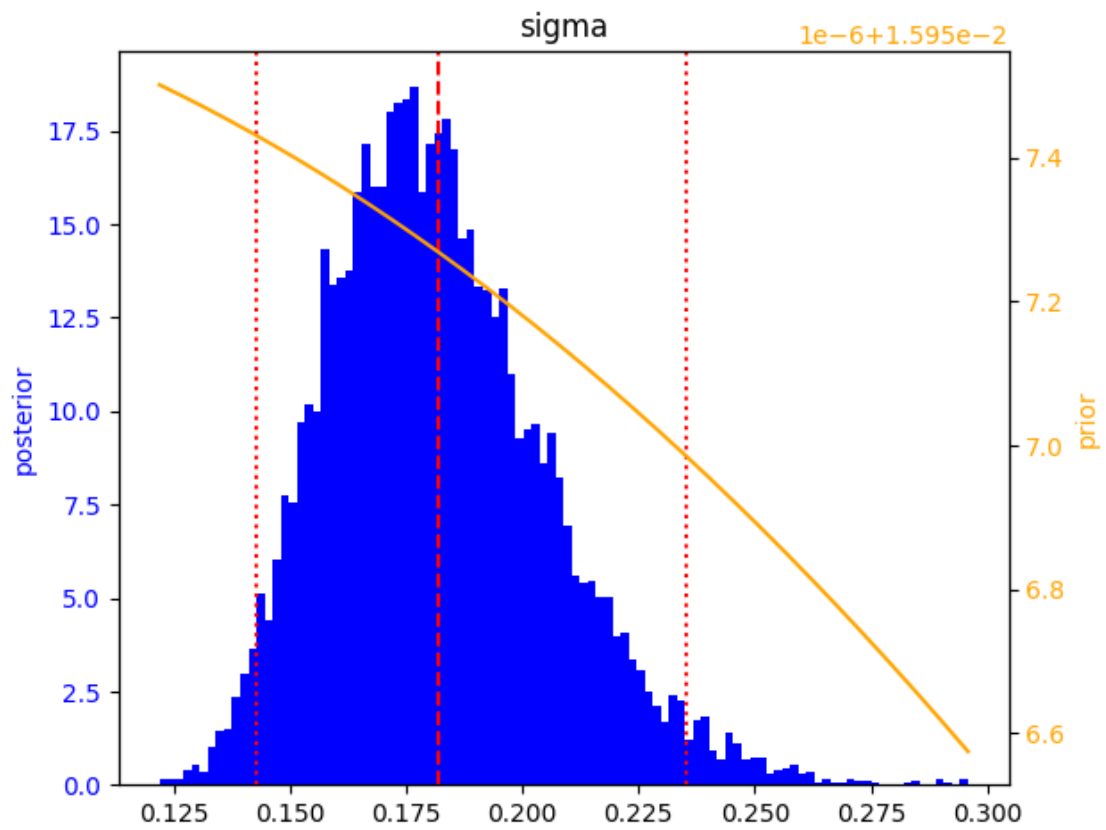


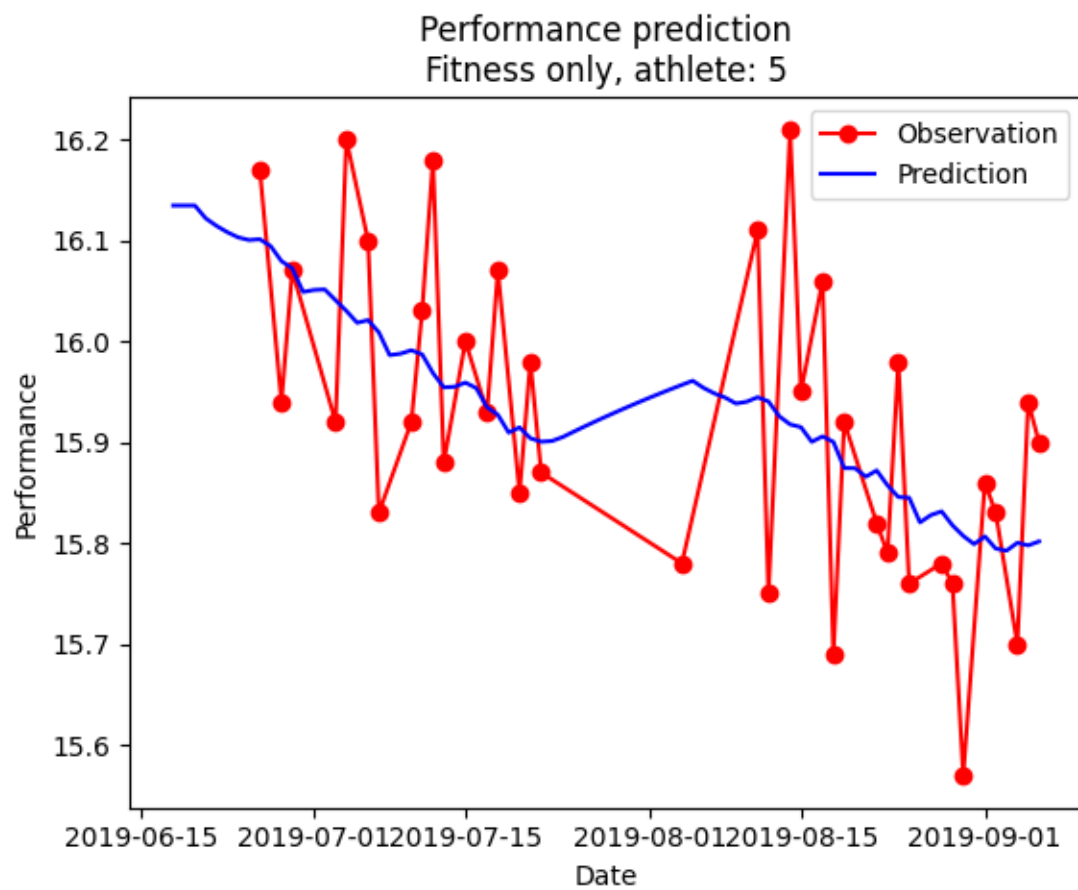


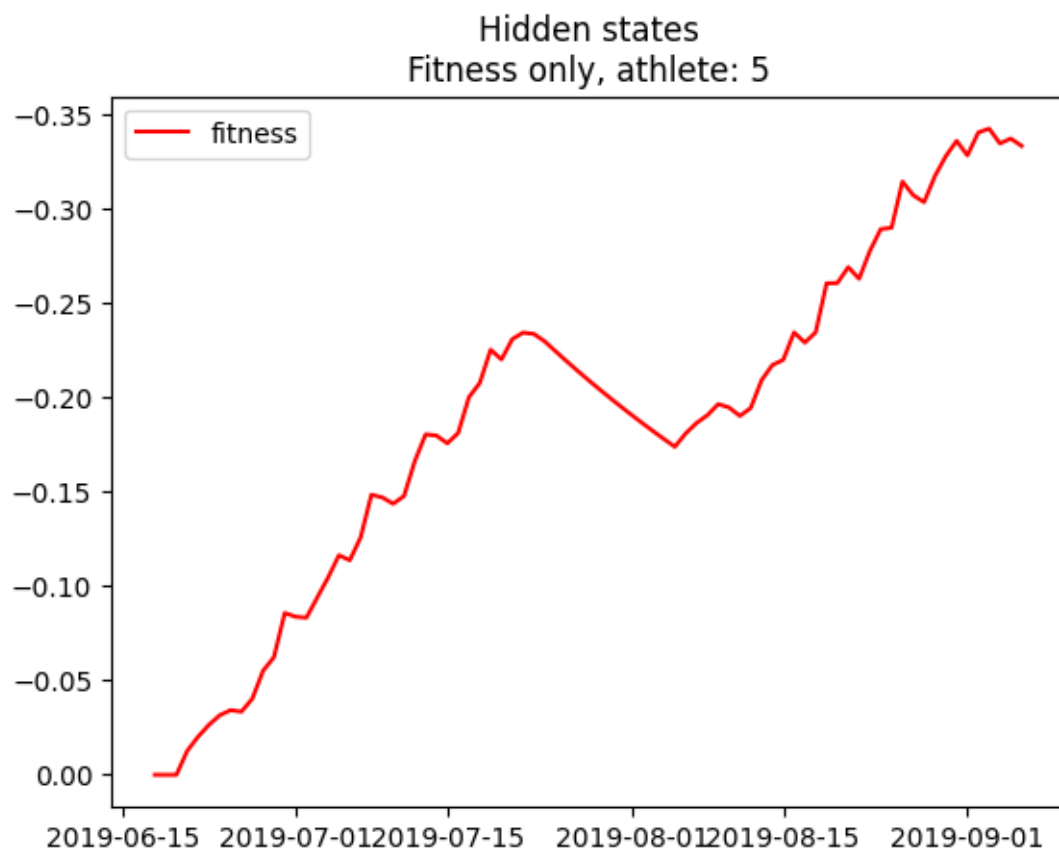


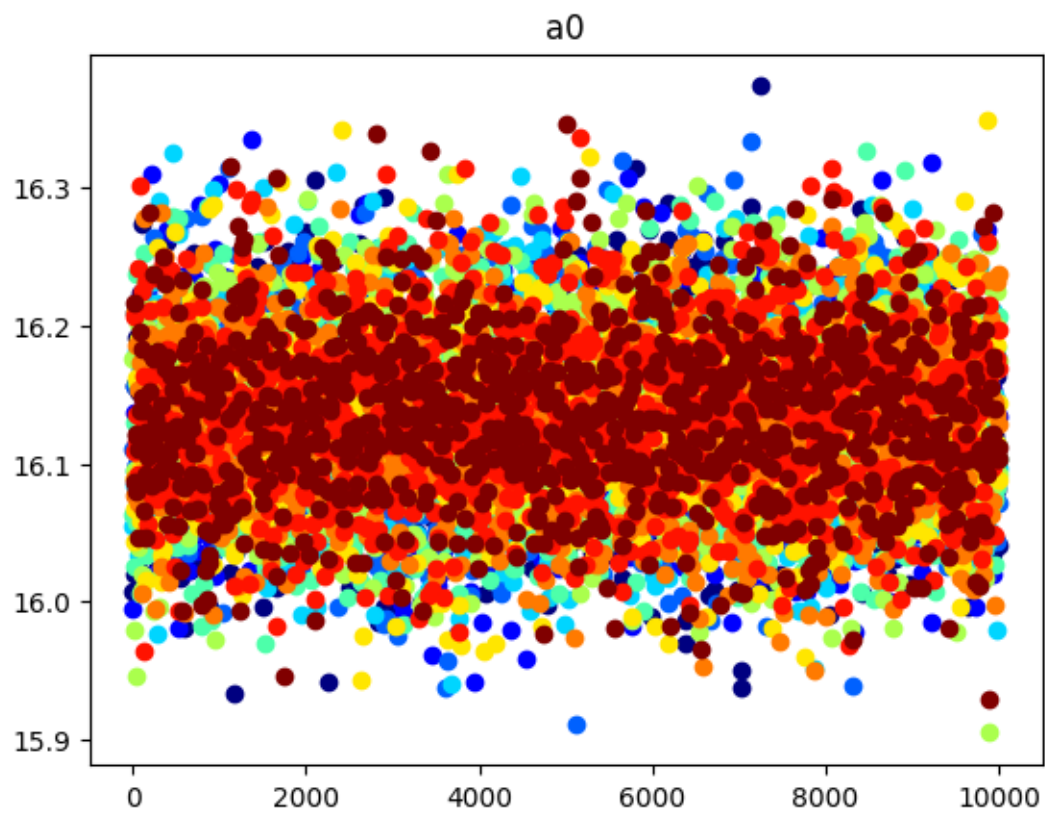


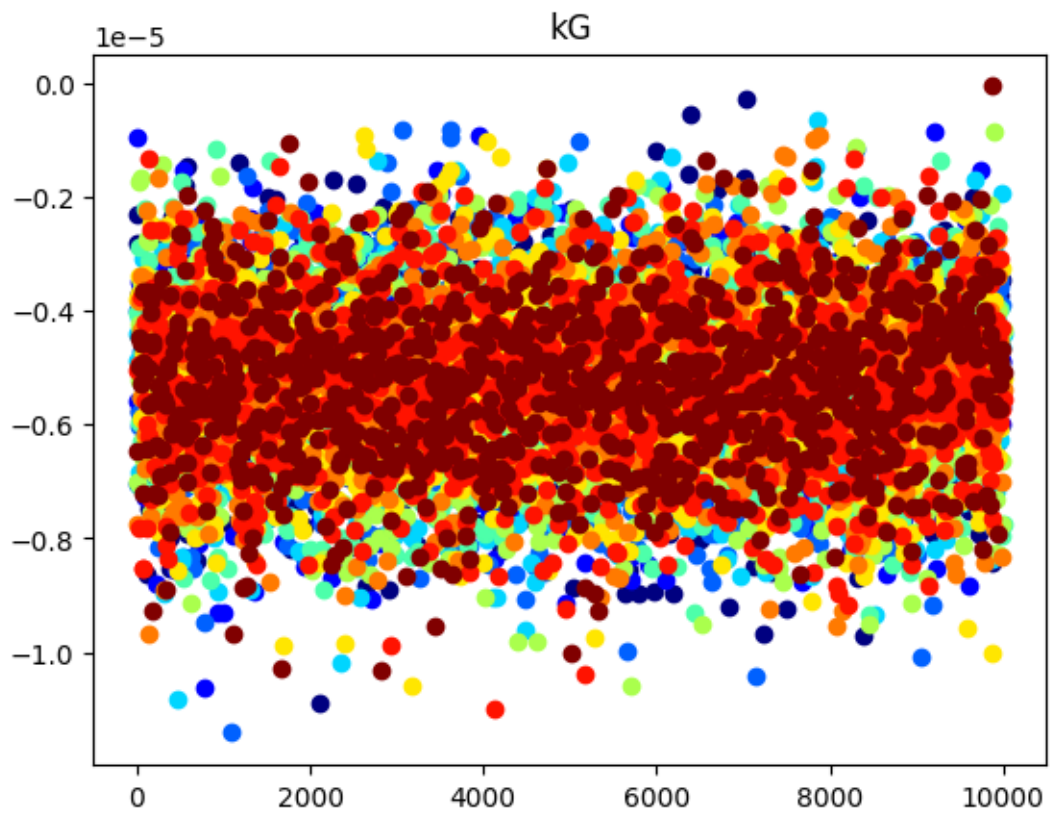


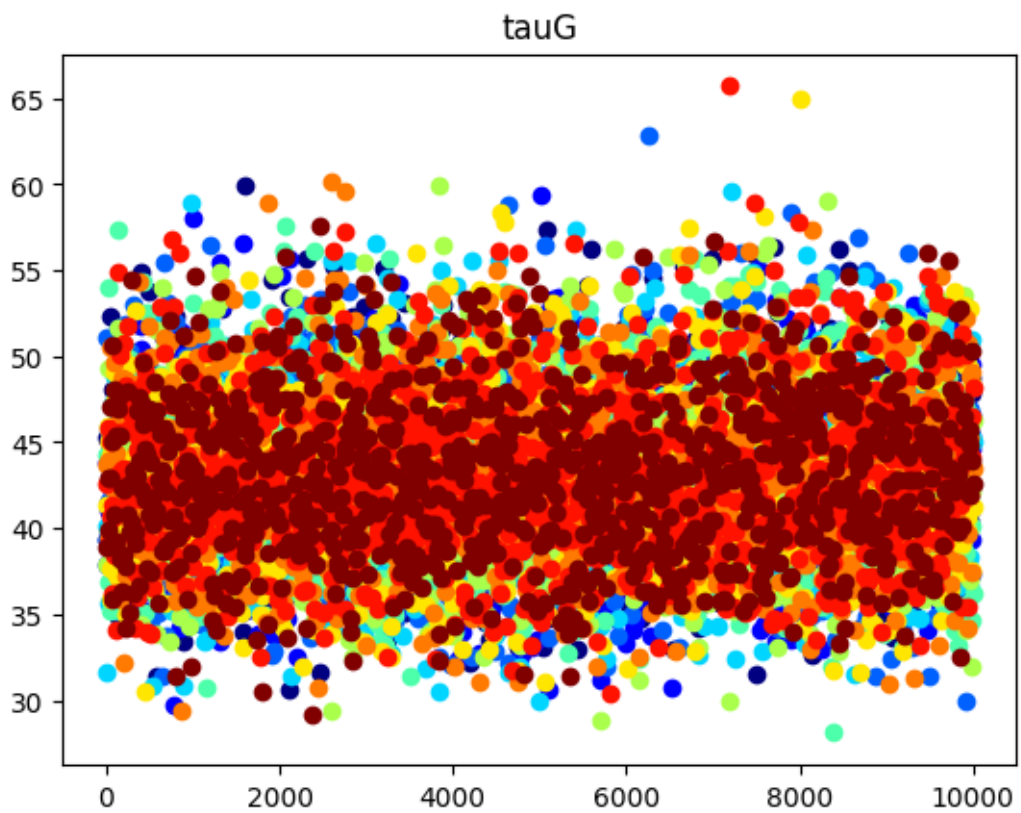


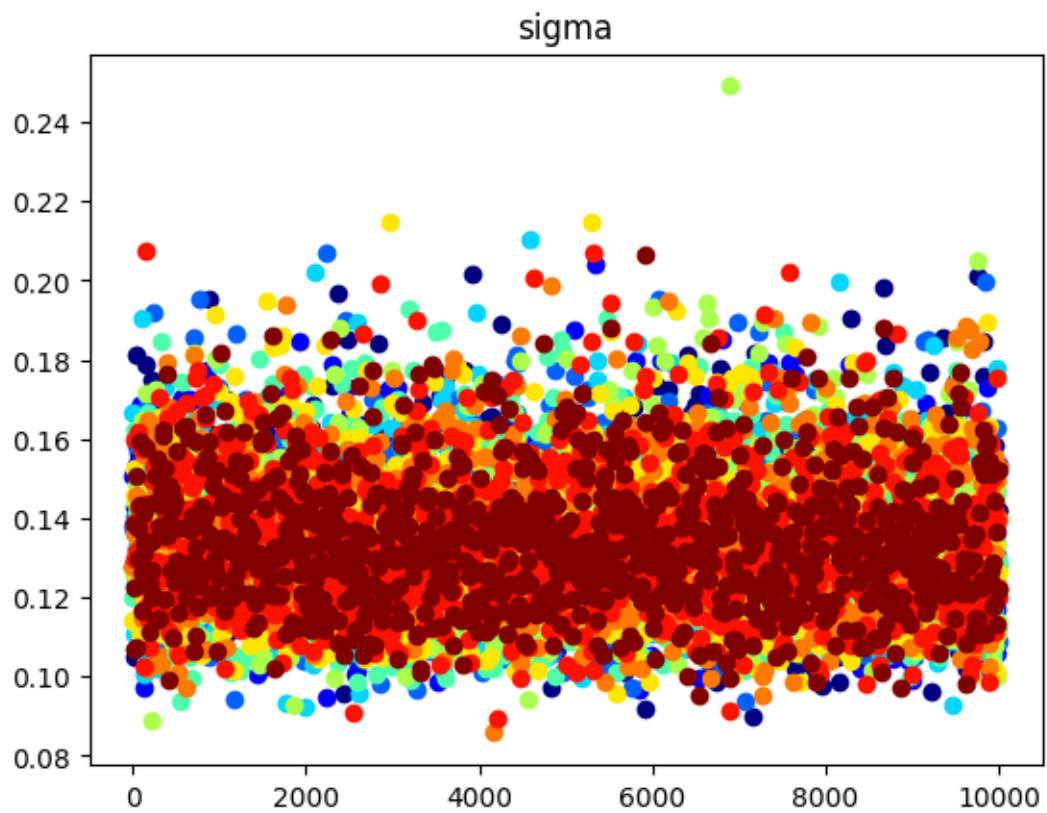


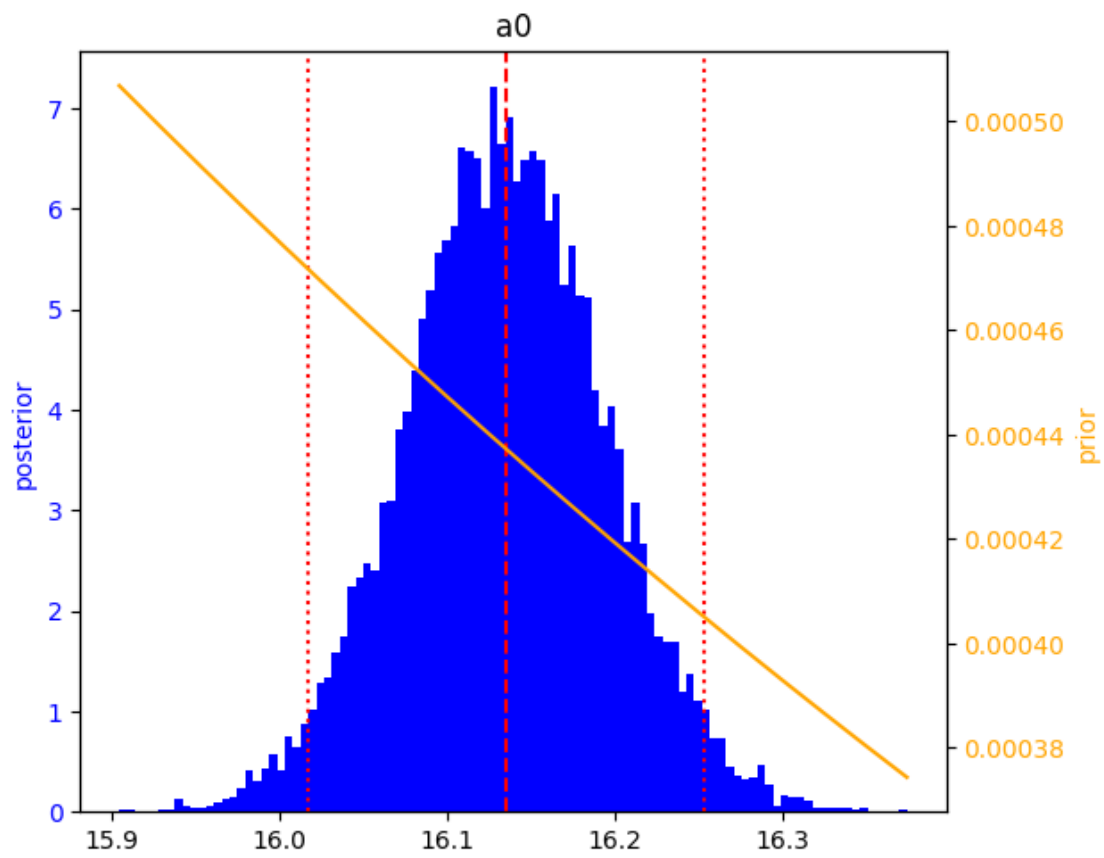


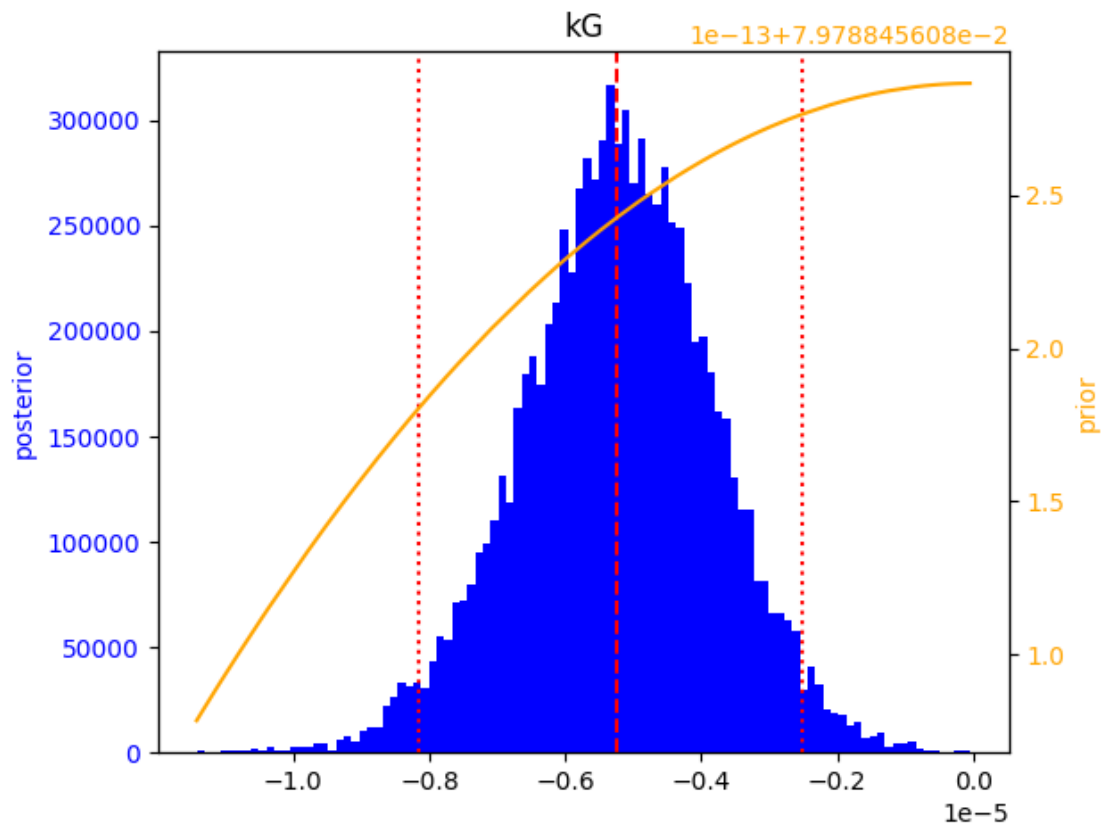


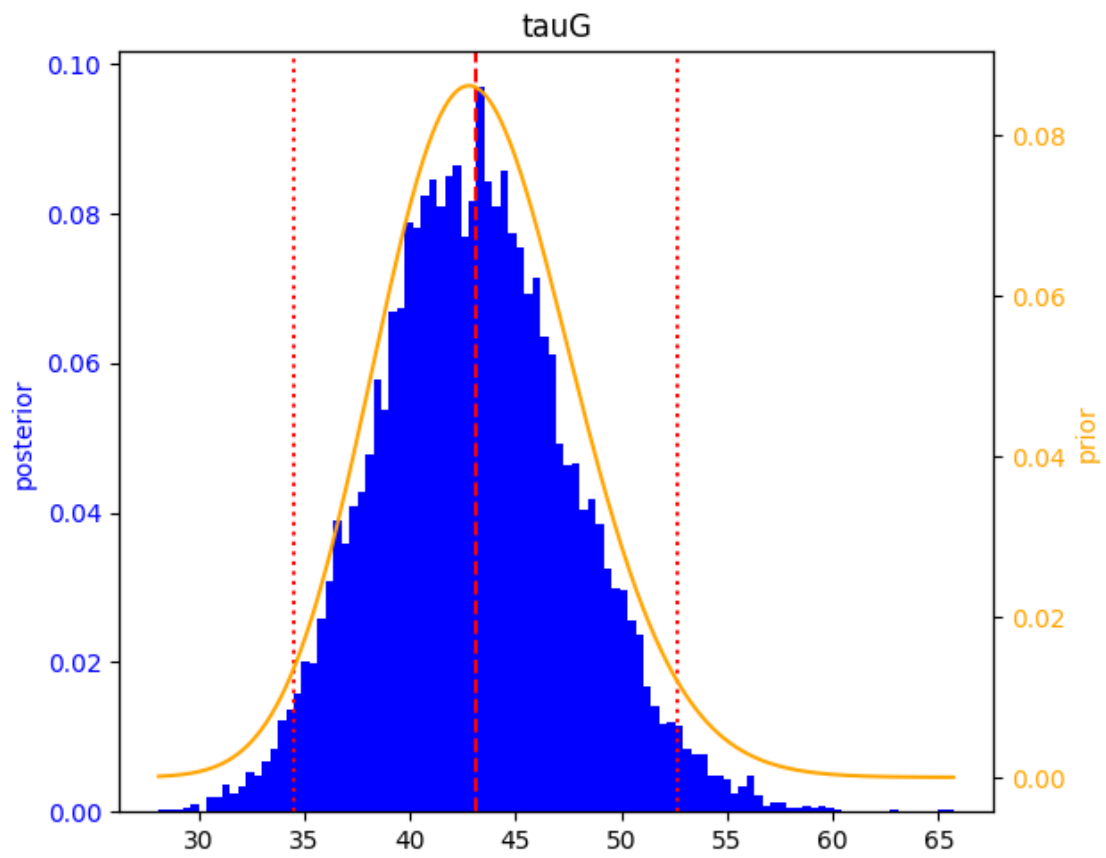


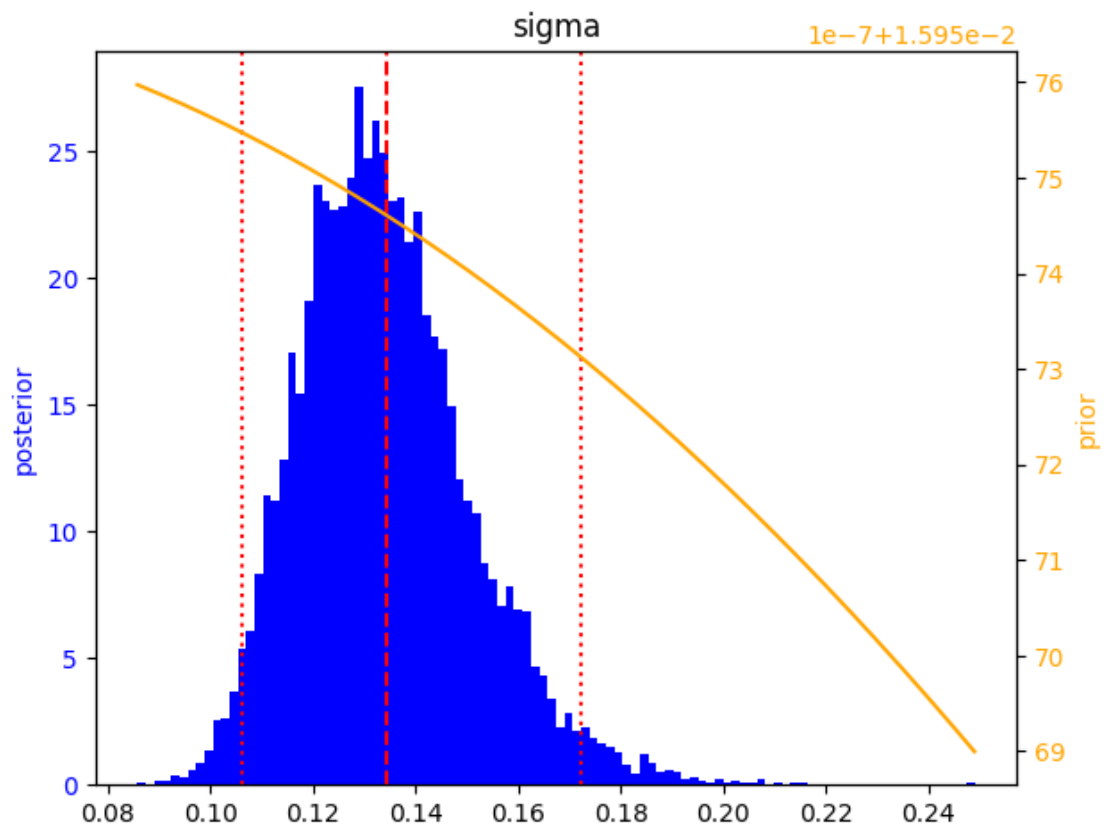


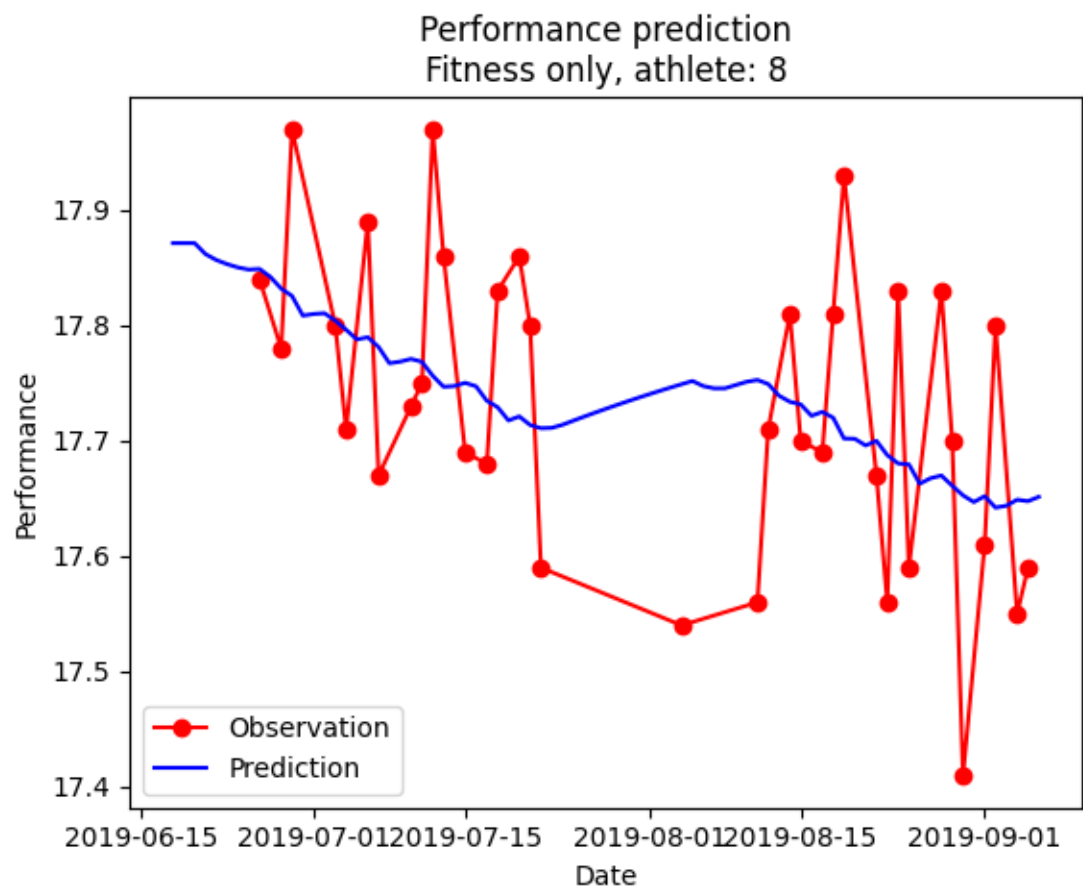




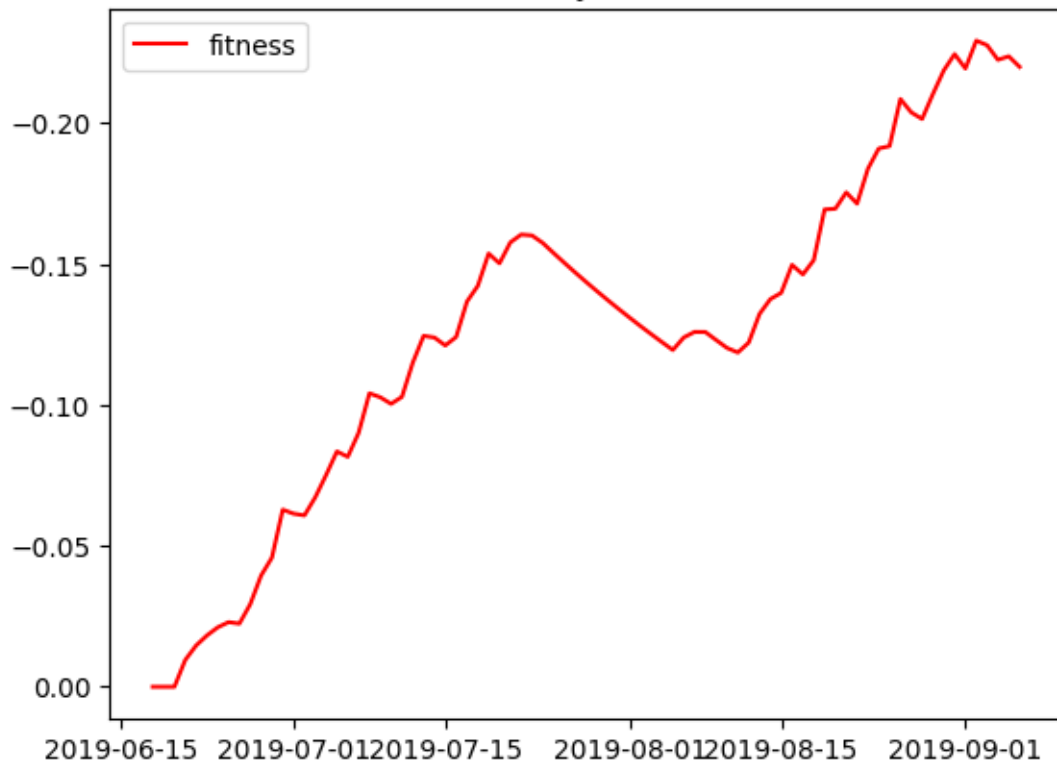


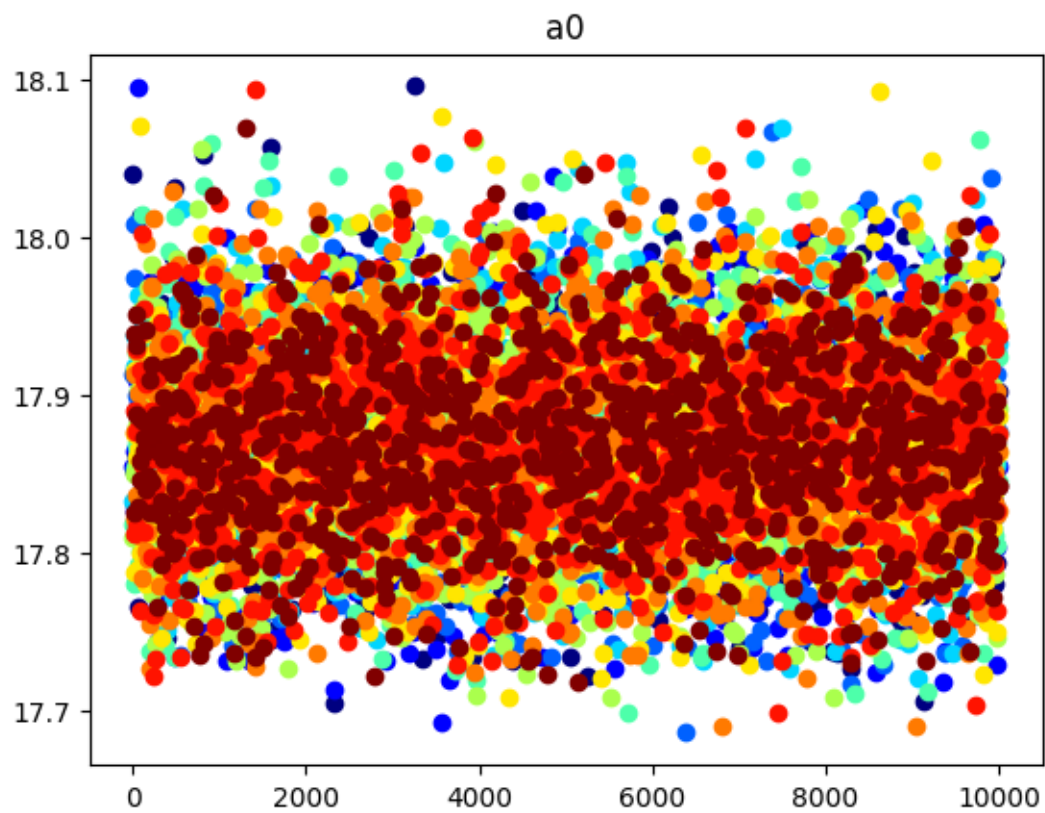


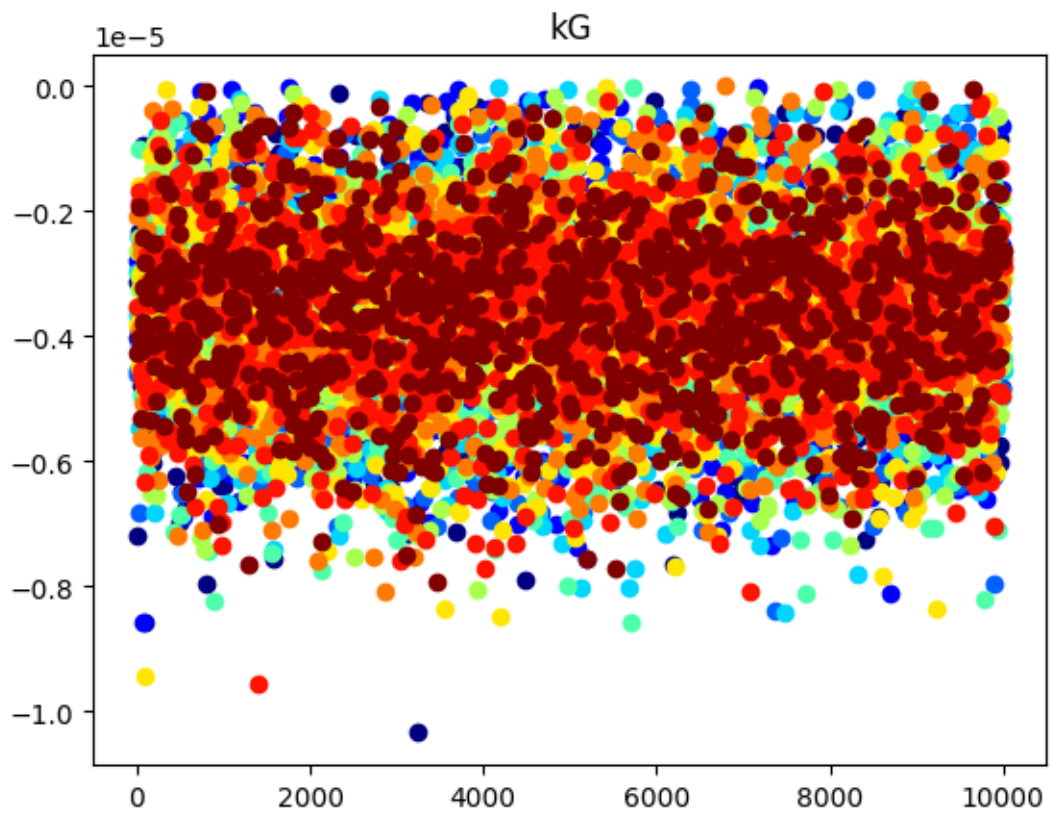


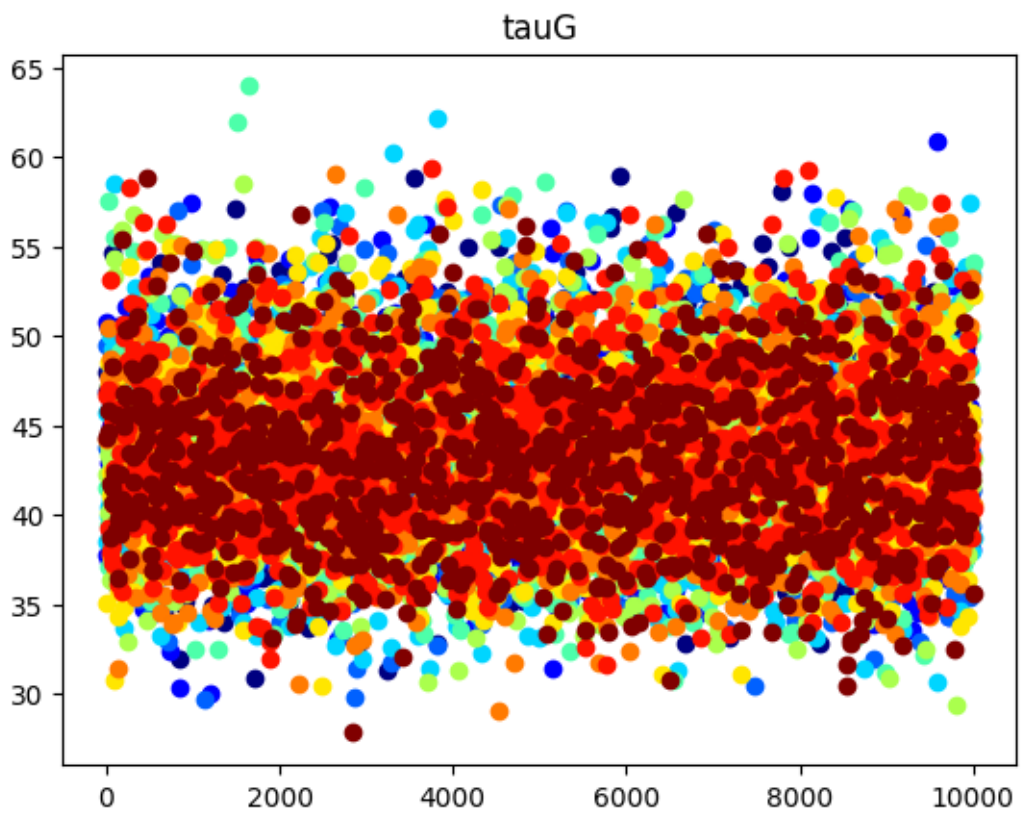


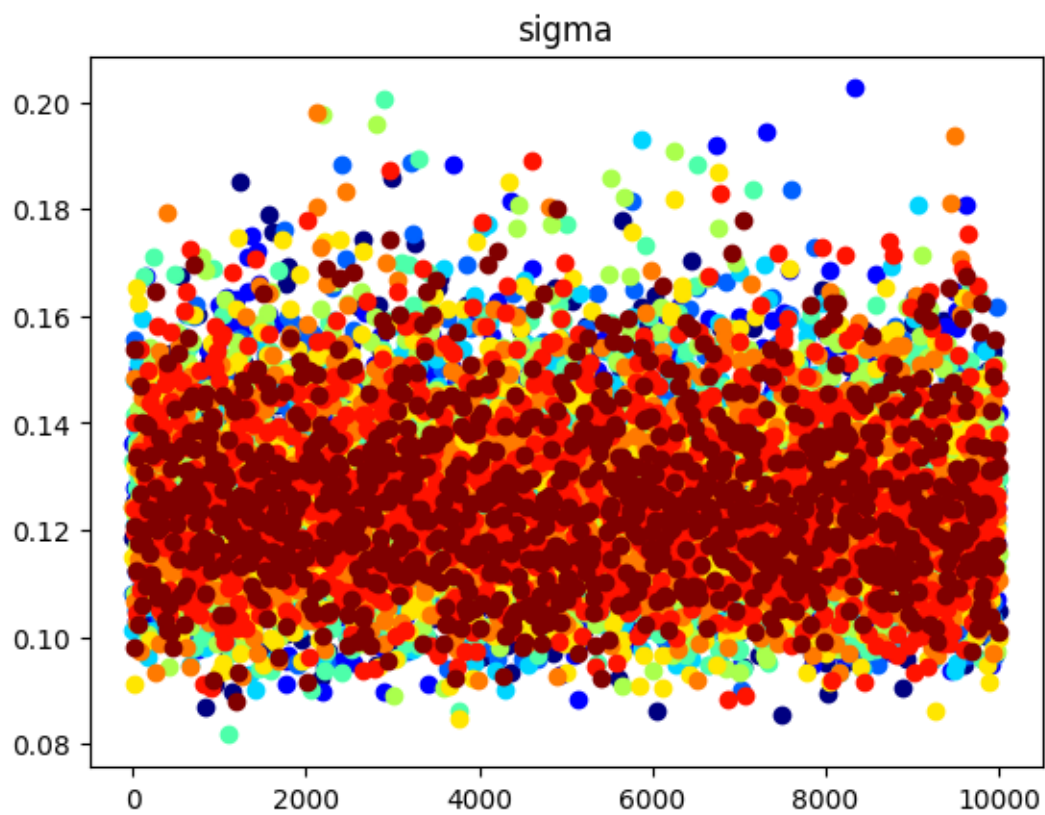
Hidden states
Fitness only, athlete: 8

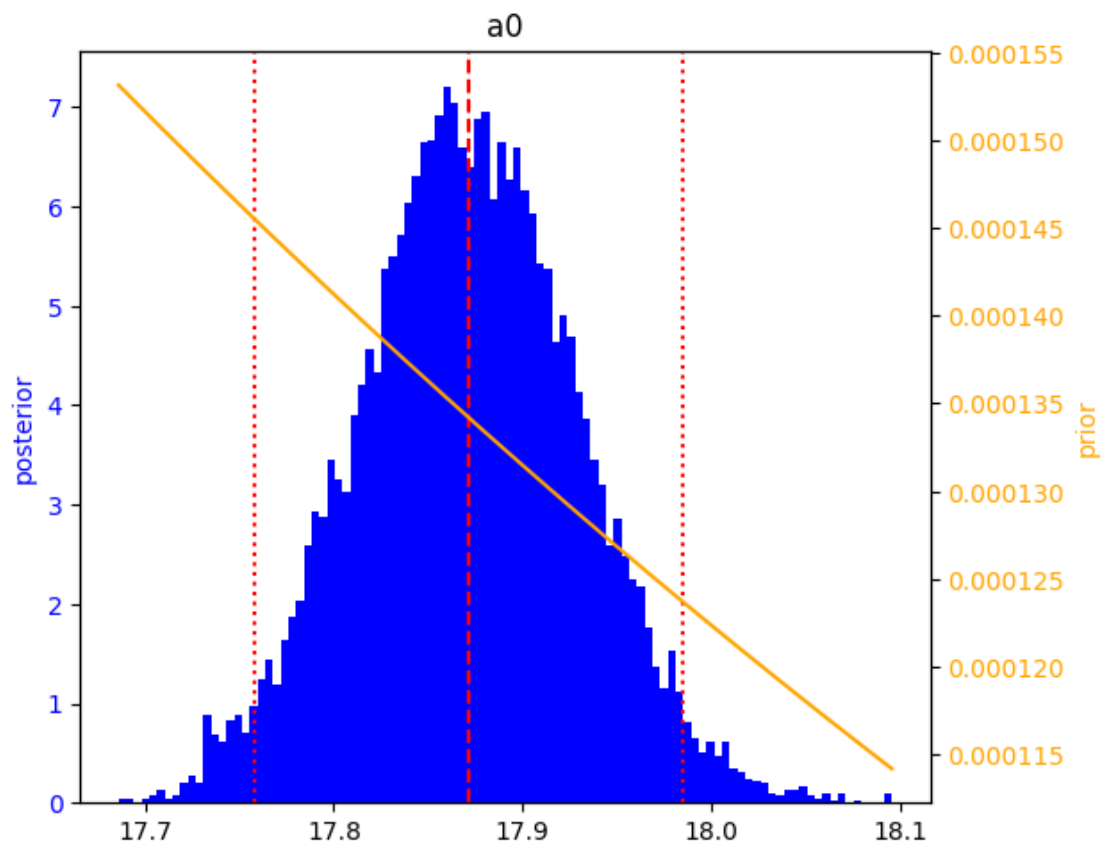


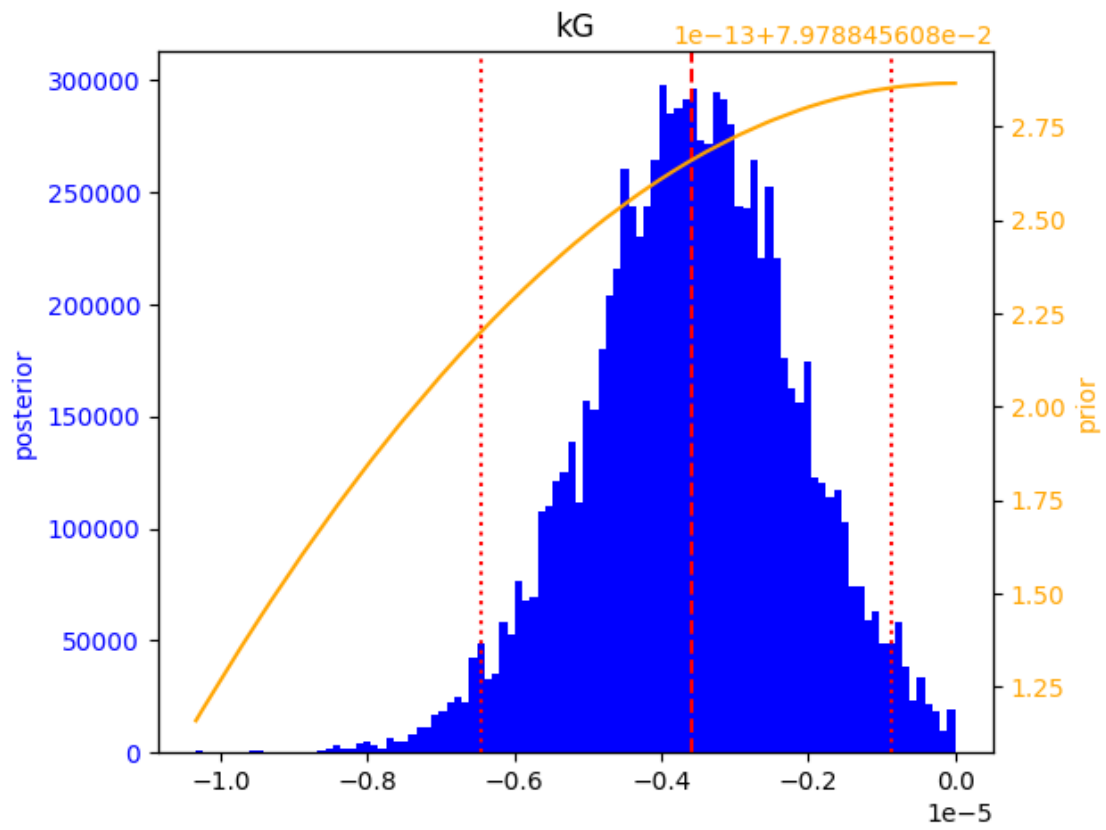


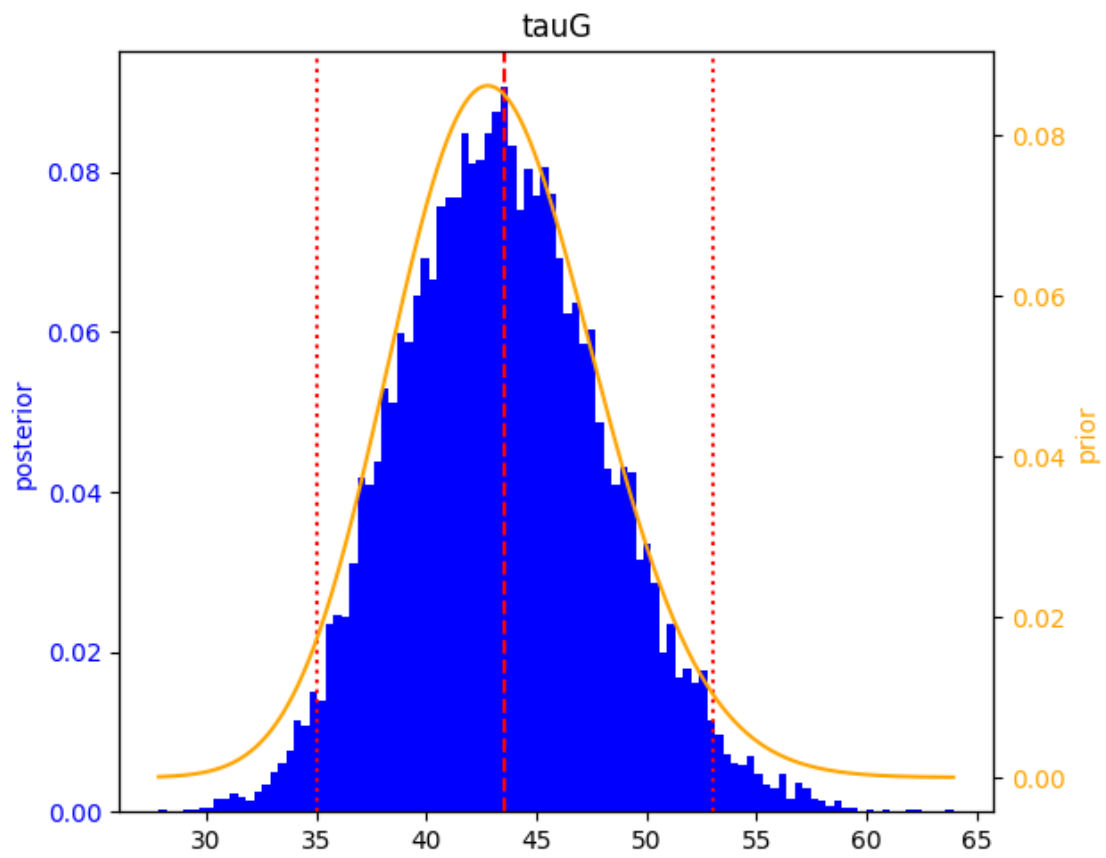


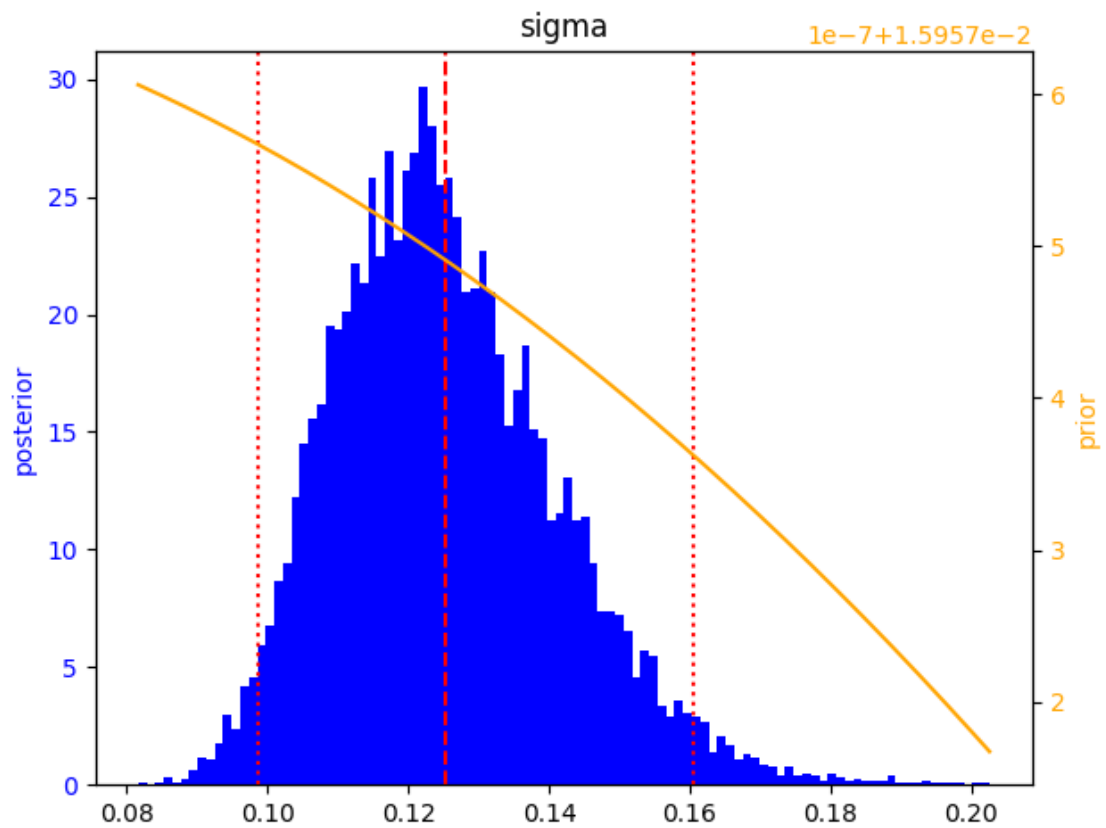


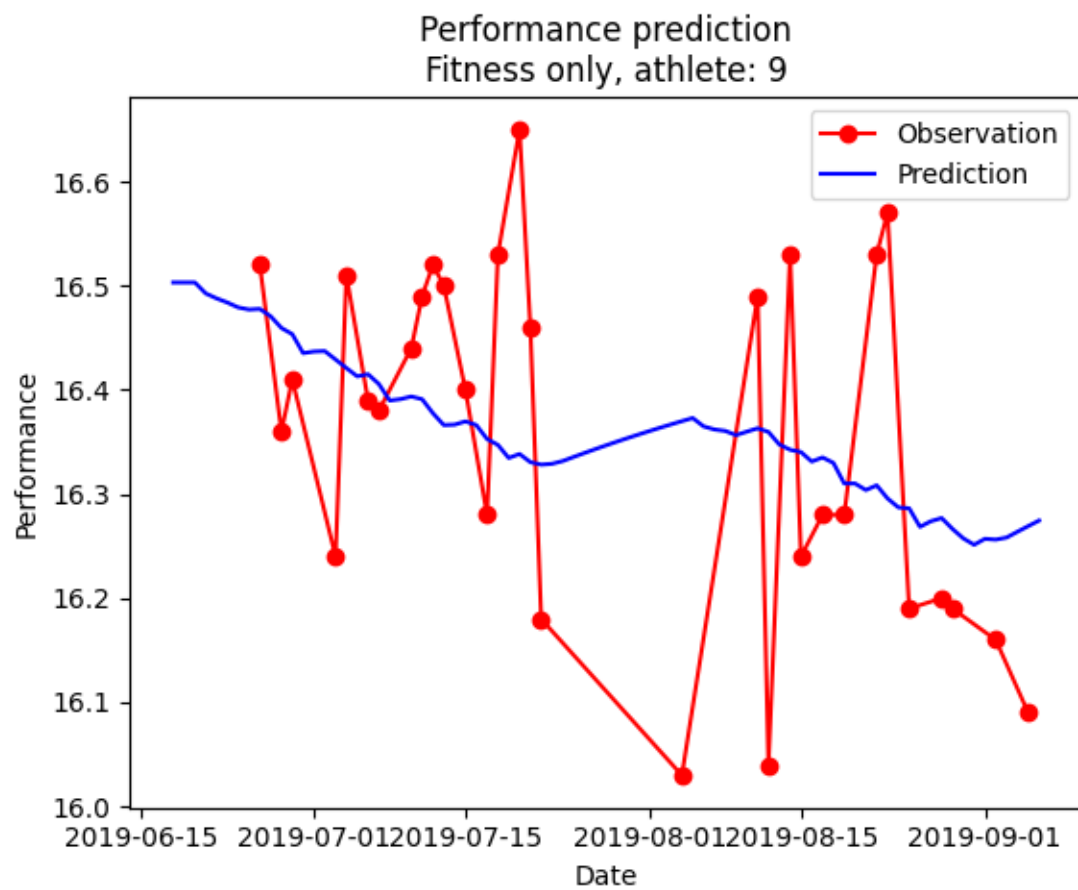


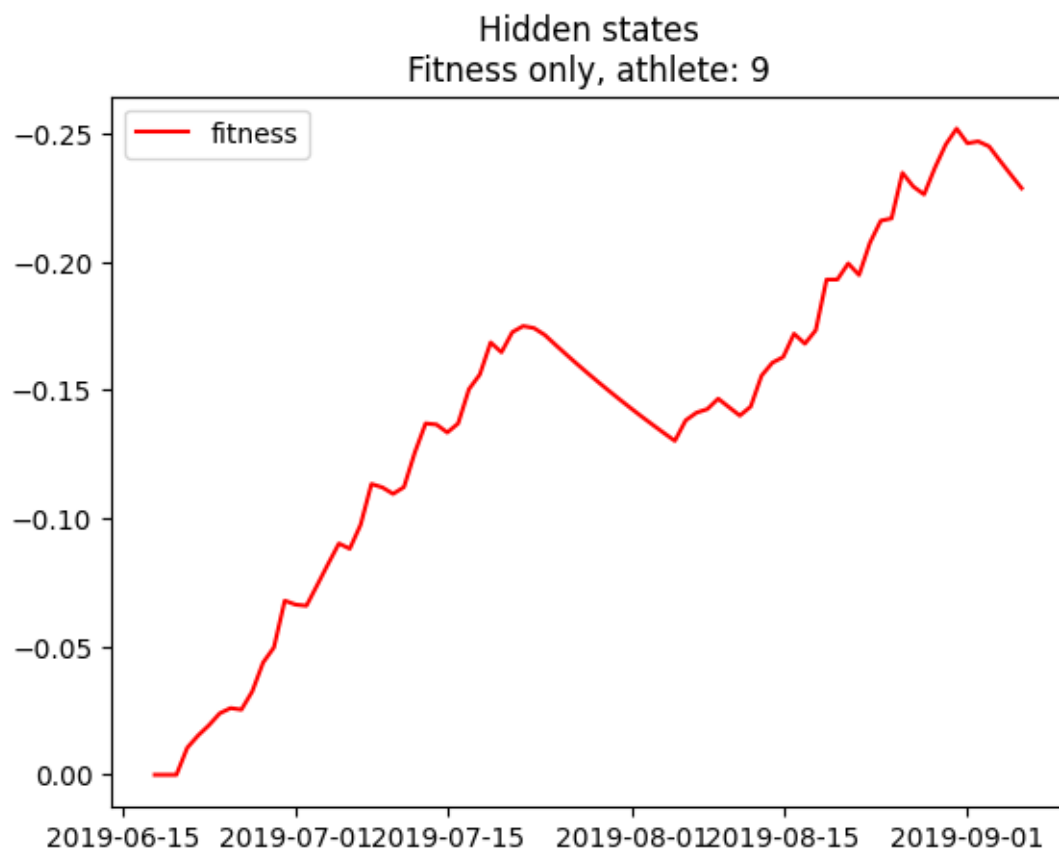


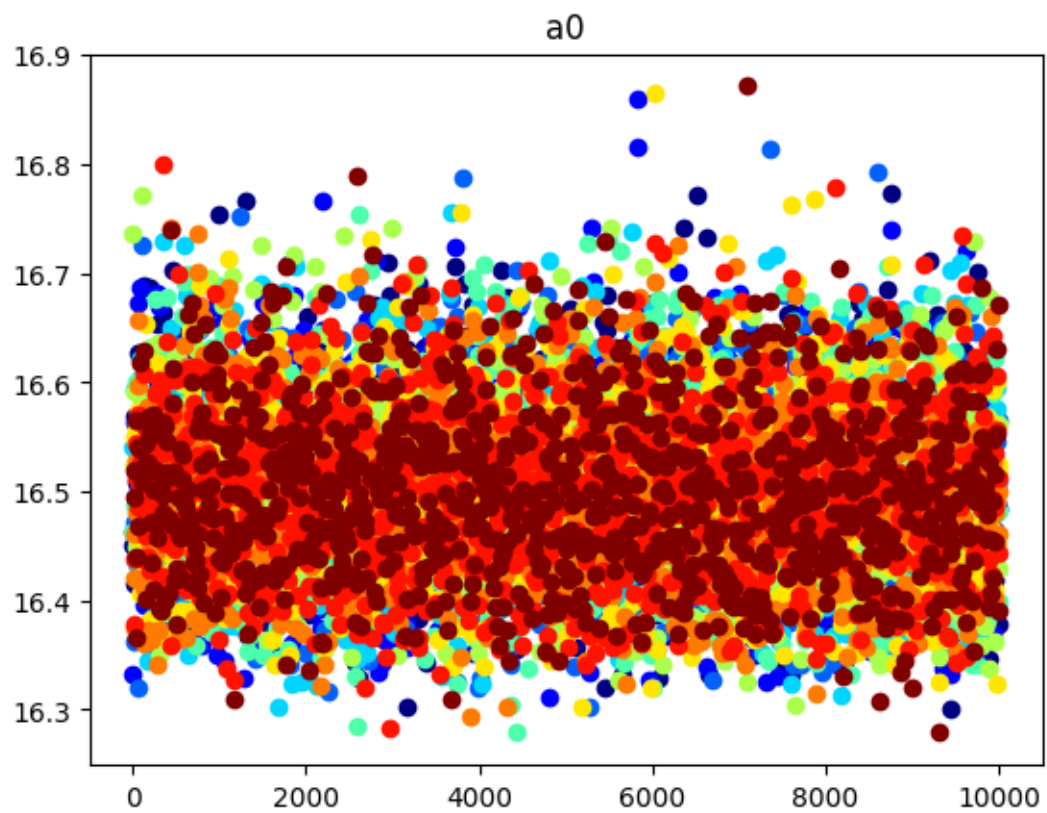


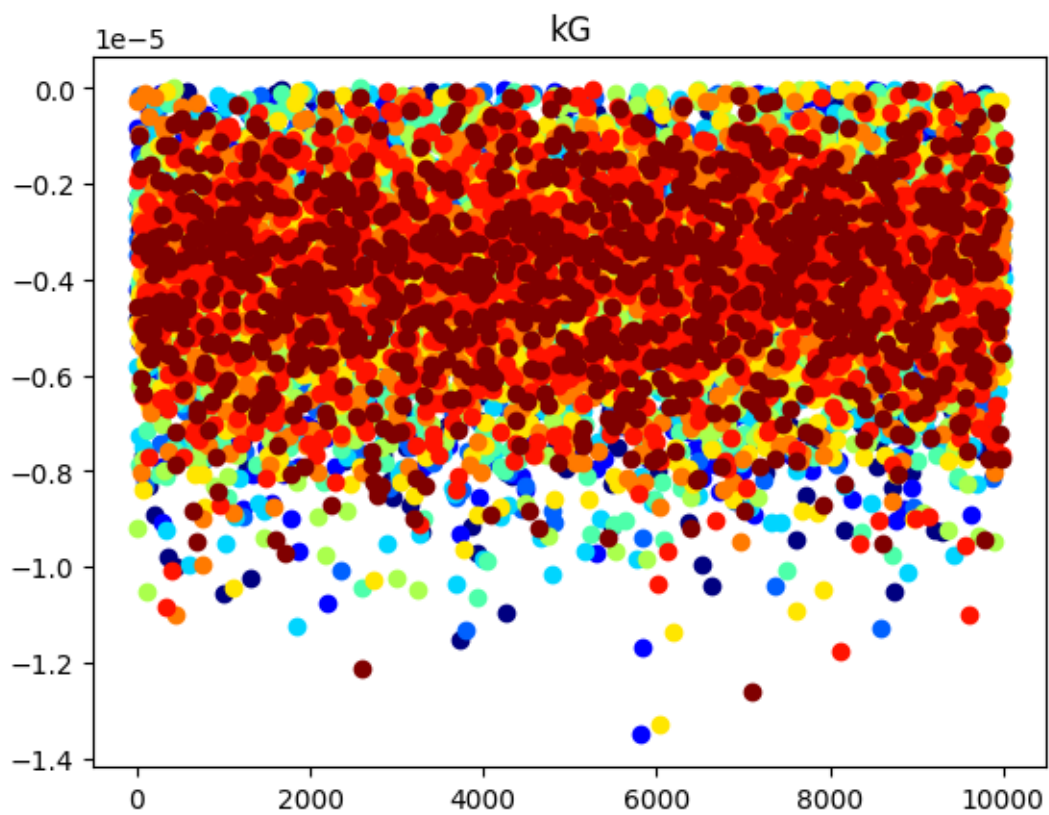


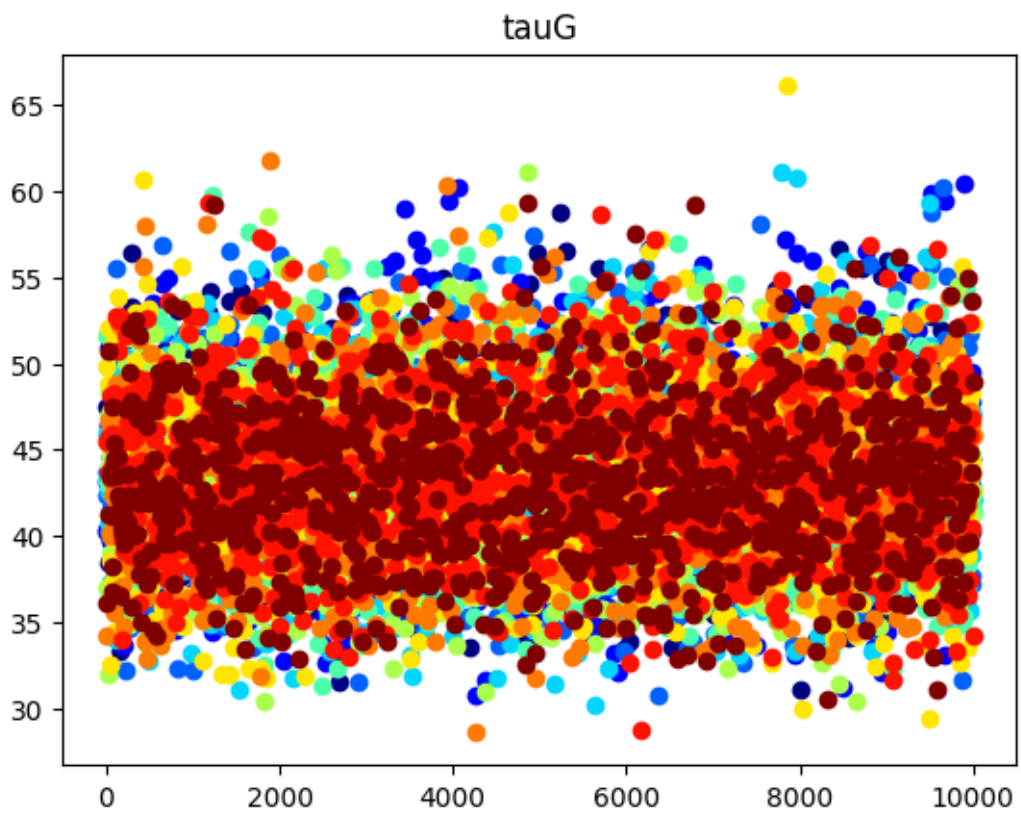


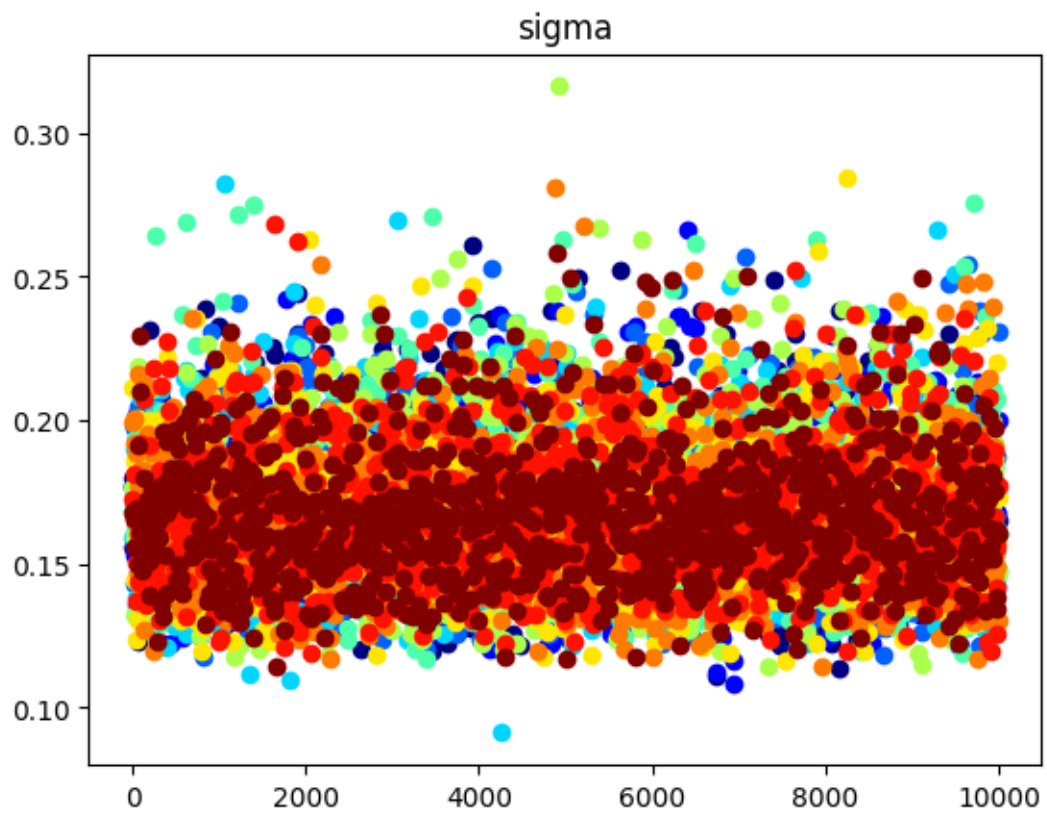


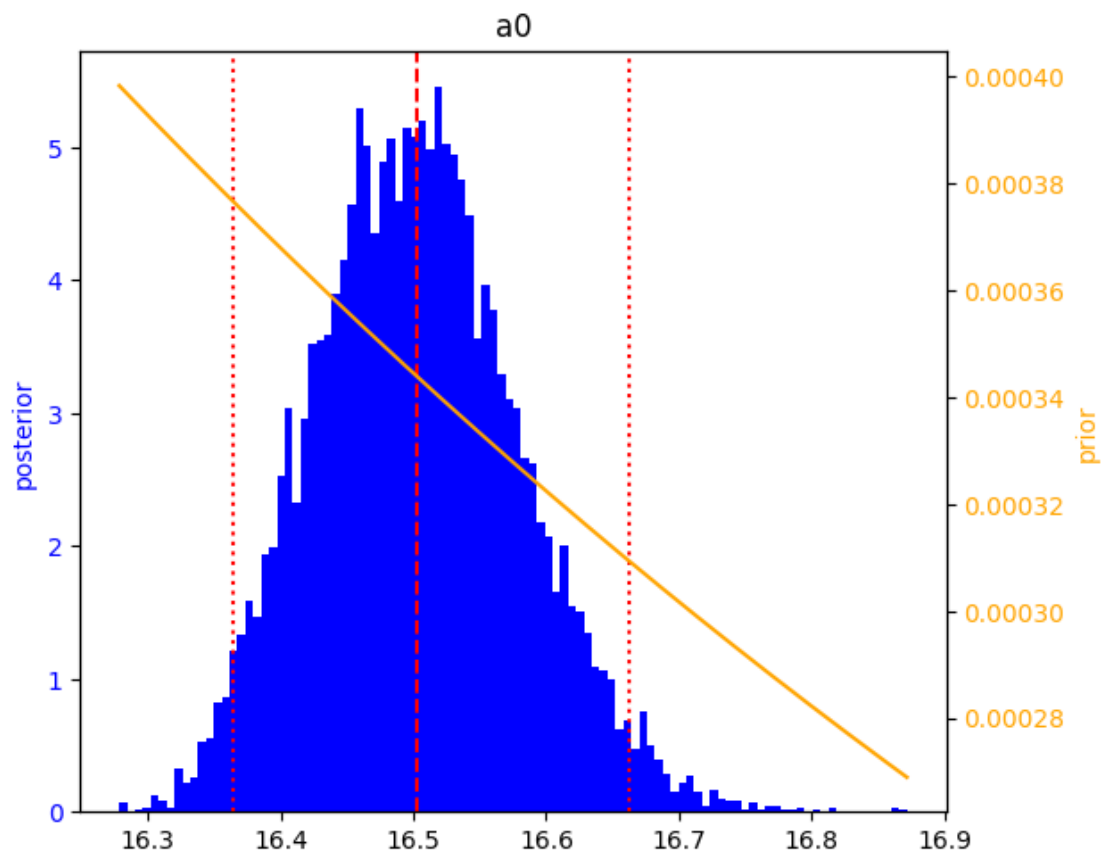


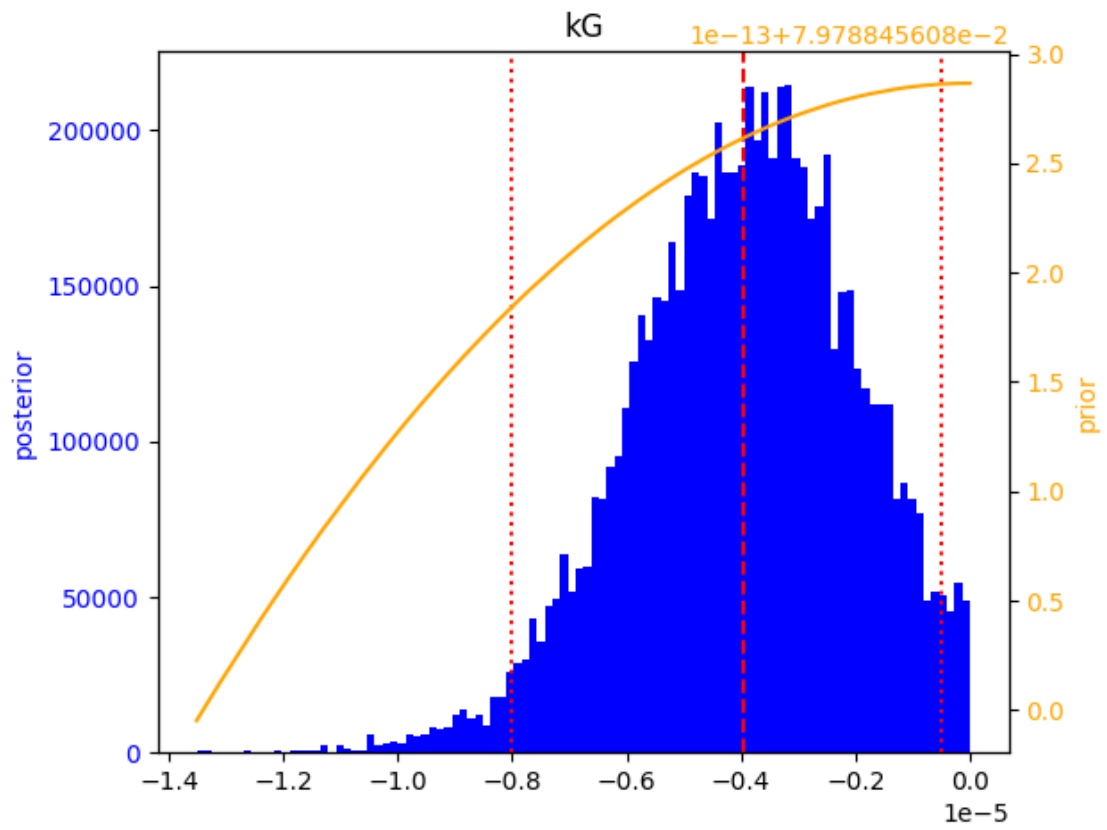


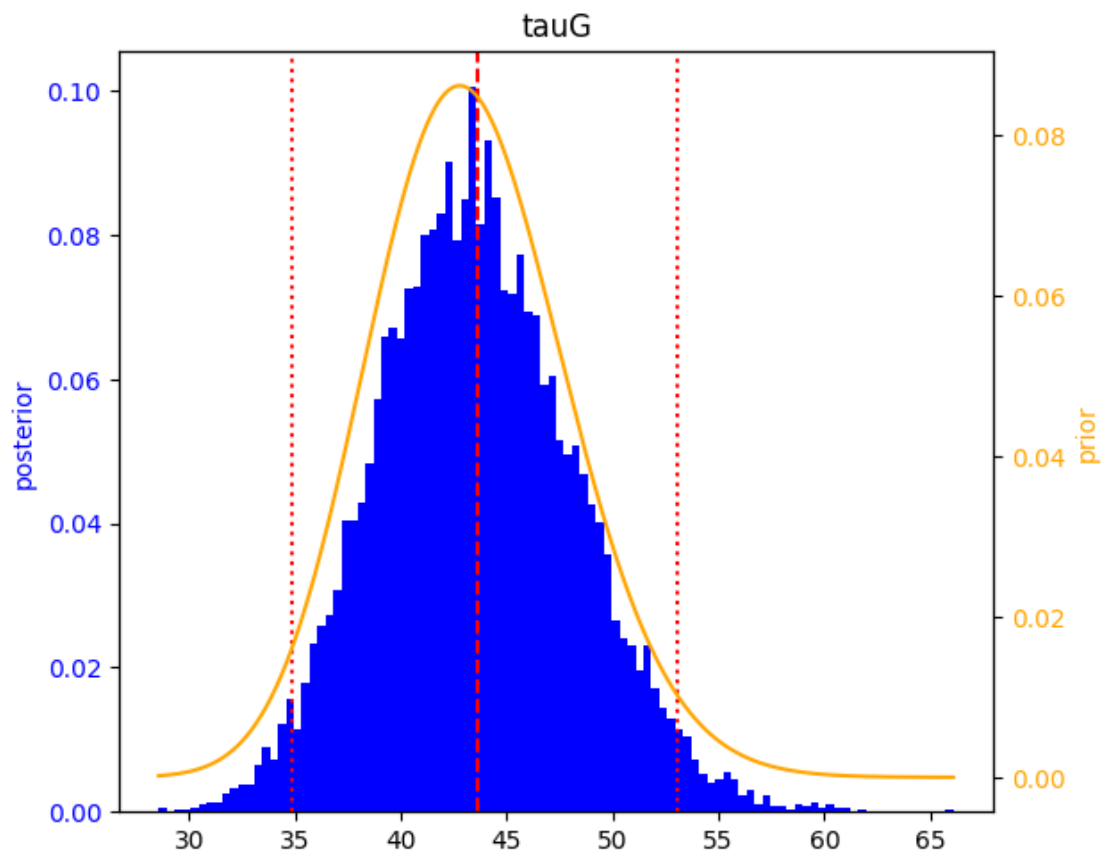


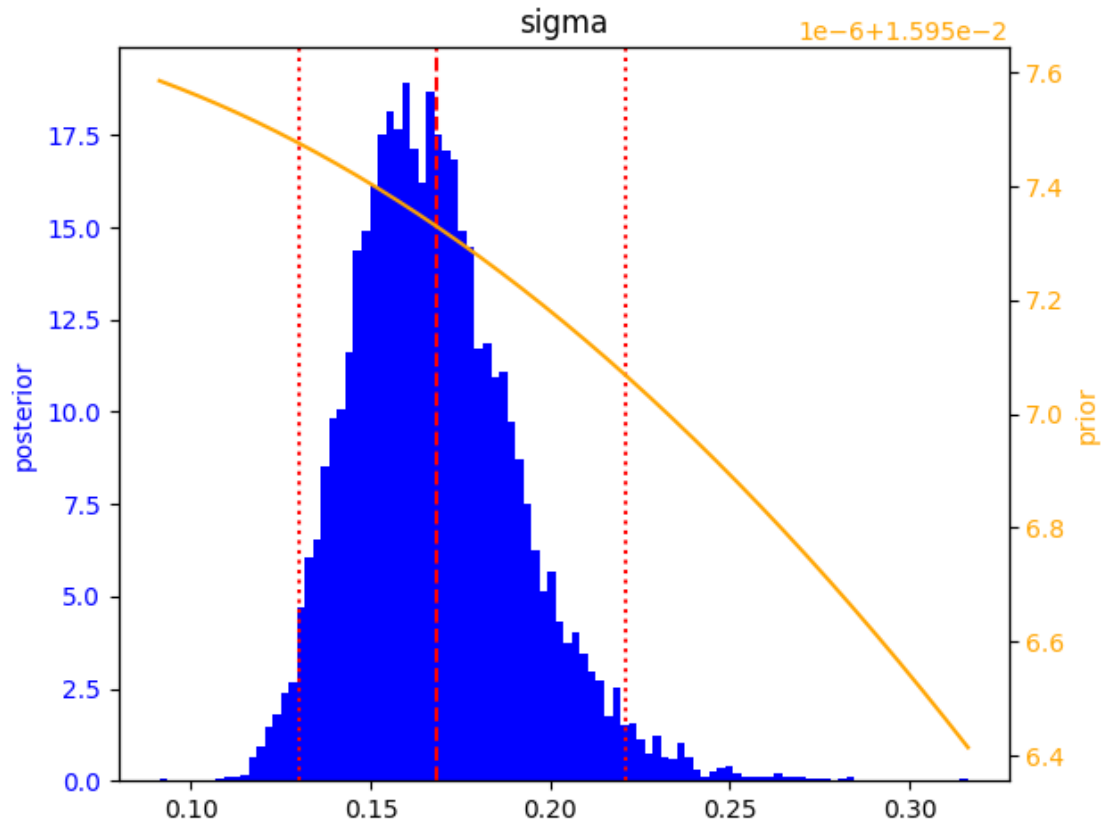








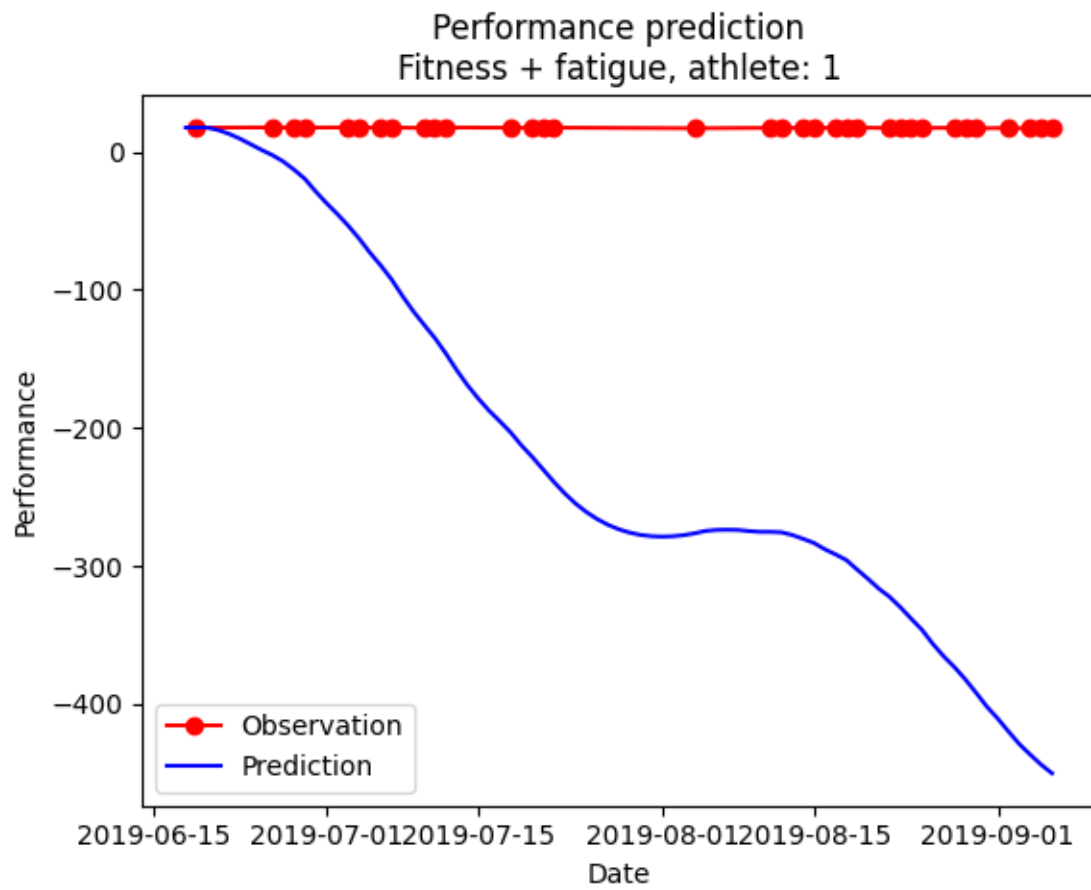




```
[5]: for i in athletes_id:
    mods[i]["Banister"].train.plot_perf(show = False)
    plt.title("Performance prediction\nFitness + fatigue, athlete: "+str(i+1))
    plt.legend()
    plt.show()

    mods[i]["Banister"].train.plot_state(show = False)
    plt.title("Hidden states\nFitness + fatigue, athlete: "+str(i+1))
    plt.legend()
    plt.show()

    mods[i]["Banister"].train.plot_rawchains()
    mods[i]["Banister"].train.plot_posteriors()
```



Hidden states
Fitness + fatigue, athlete: 1

