

# Bayesian\_fitting\_2016

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/2016_data.csv', delimiter = ",")
df["date"] = pd.to_datetime(df["date"], format = "%Y-%m-%d")
athok = [i for i in range(len(df)) if not pd.isnull(df.iloc[i,df.columns.
↳get_loc("athlete")]]
df = df.iloc[athok,:]
datok = [i for i in range(len(df)) if not pd.isnull(df.iloc[i,df.columns.
↳get_loc("date")]]
df = df.iloc[datok,:]

athletes_id = [x for x in df["athlete"].unique() if x != "H8"] # not enough
↳data for H8
np.random.seed(123)

mods = {}
for ath in athletes_id:
    mods[ath] = {}
    mods[ath]["Fitness"] = p3.Fitness (df.loc[df["athlete"] == ath,:].copy(),
                                     objective = "max",
                                     HL_fitness = 30)
    mods[ath]["Banister"] = p3.Banister(df.loc[df["athlete"] == ath,:].copy(),
                                     objective = "max",
                                     HL_fitness = 30,
                                     HL_fatigue = 2)

[2]: freq = pd.read_csv("Performances_in_training_2016.csv")

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

```

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

mods[ath][model].fitBayes(hyparams = {"sd_a0": 5, # flat
                                     "sd_k": 5, # flat
                                     "sd_sigma": 25,
                                     "b_tau": 2},
                        bounds = {"a0": (-1e4,1e4),
                                "k" : (0,1),
                                "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)

```

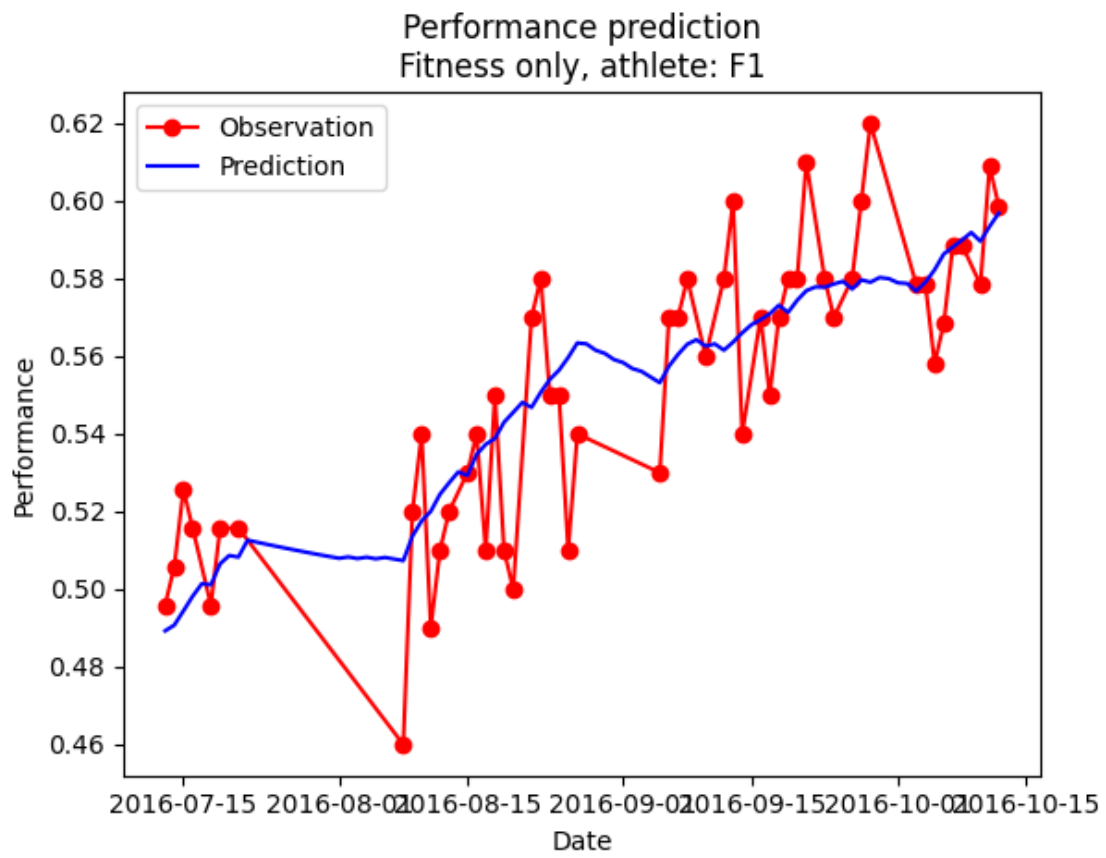
```

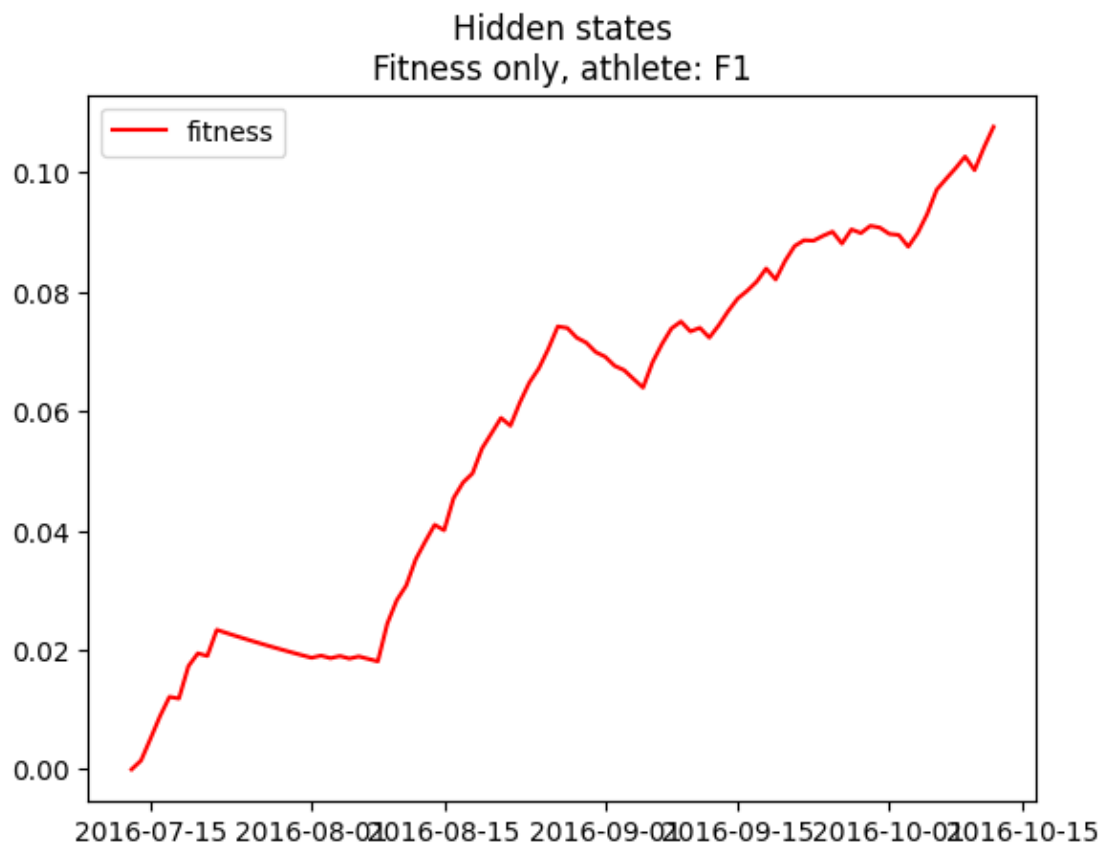
[4]: for ath in athletes_id:
    mods[ath]["Fitness"].train.plot_perf(show = False)
    plt.title("Performance prediction\nFitness only, athlete: "+ath)
    plt.legend()
    plt.show()

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

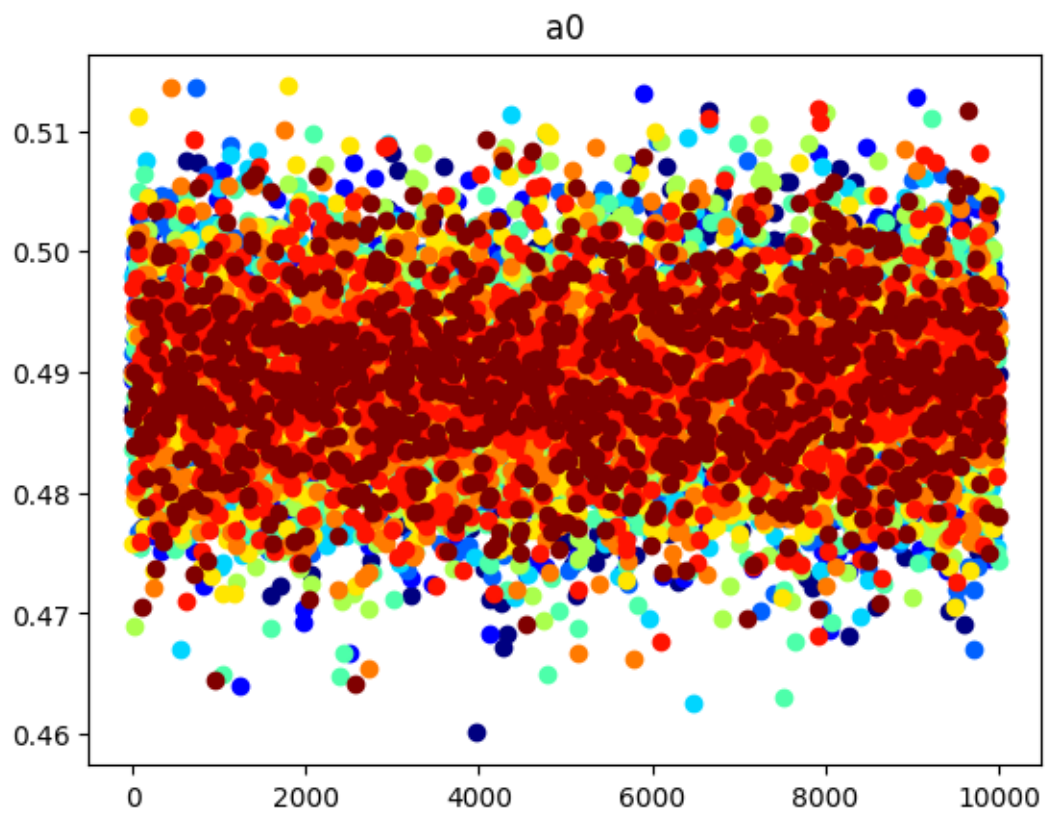
    mods[ath]["Fitness"].train.plot_rawchains()
    mods[ath]["Fitness"].train.plot_posteriors()

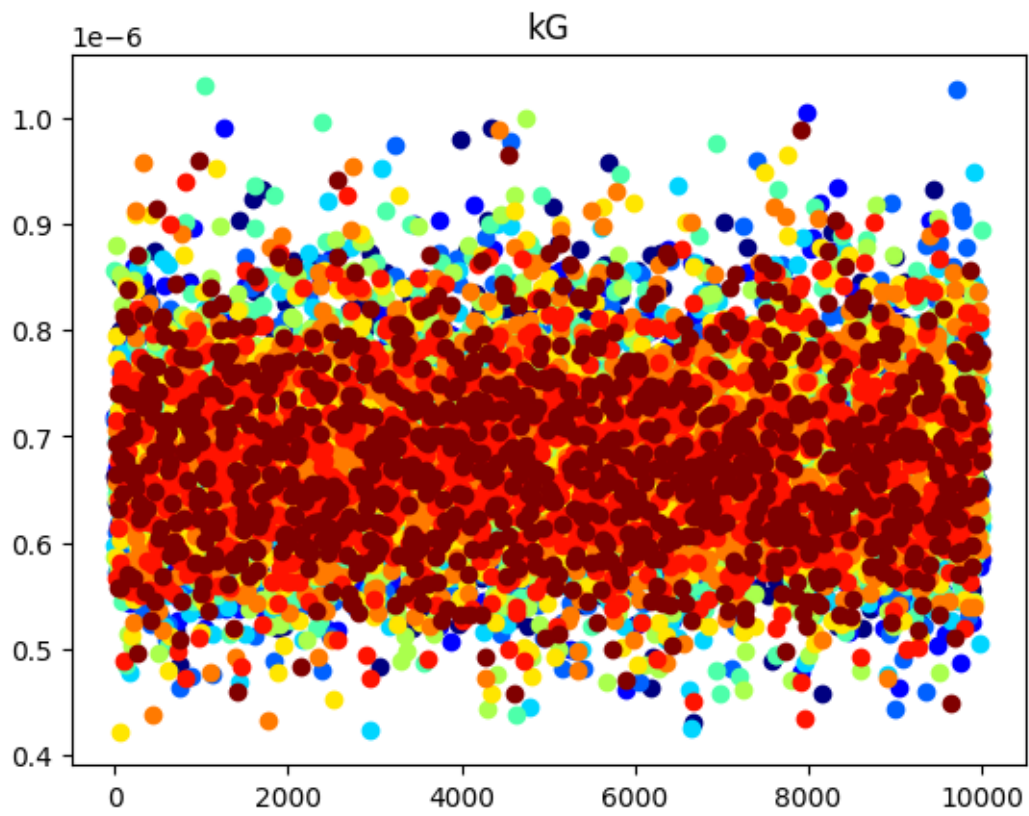
```

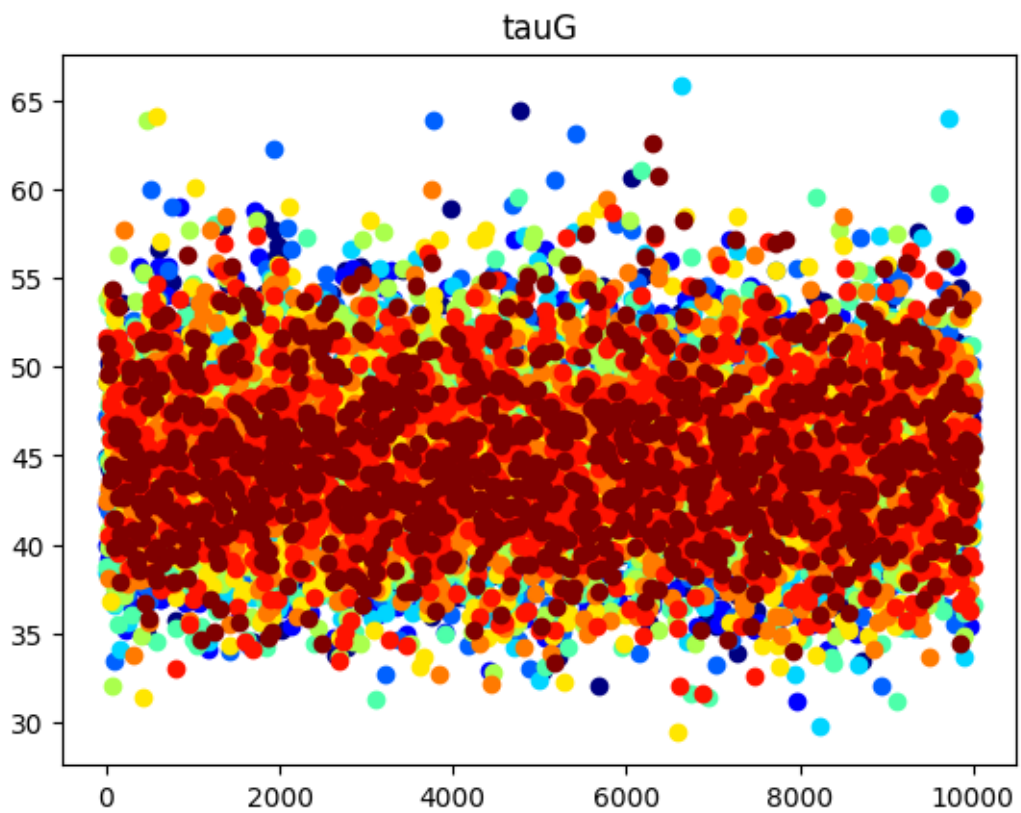


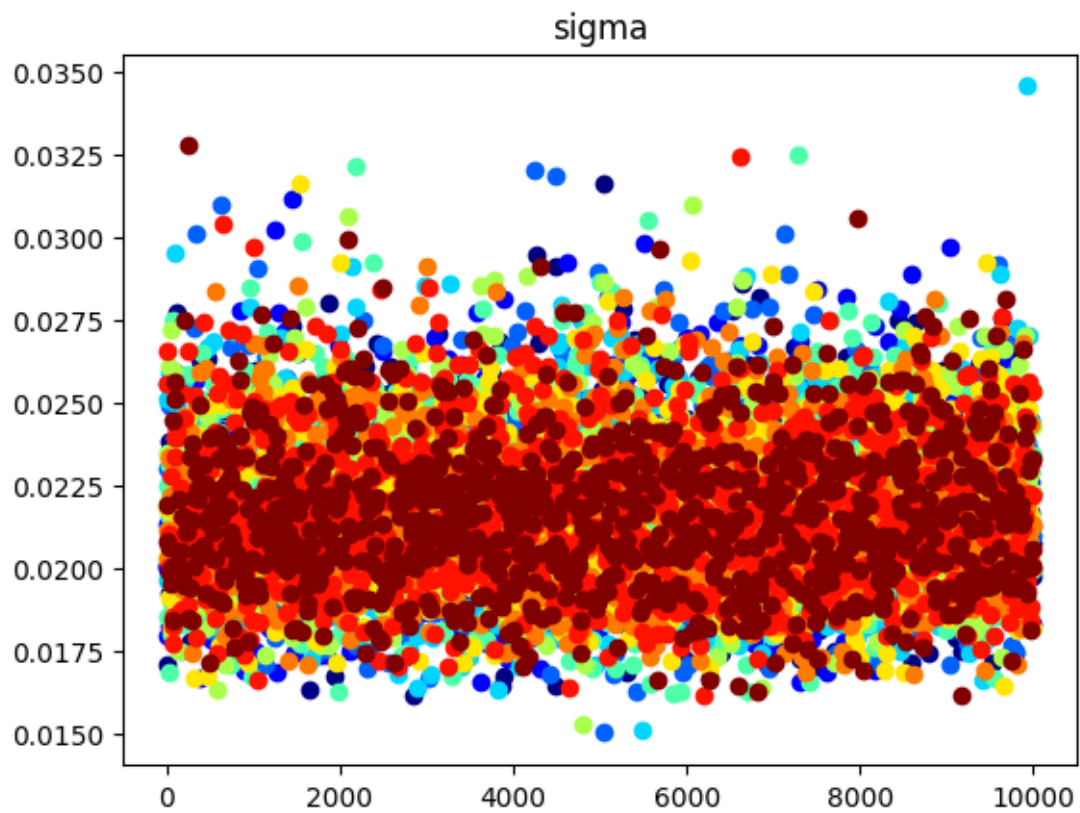


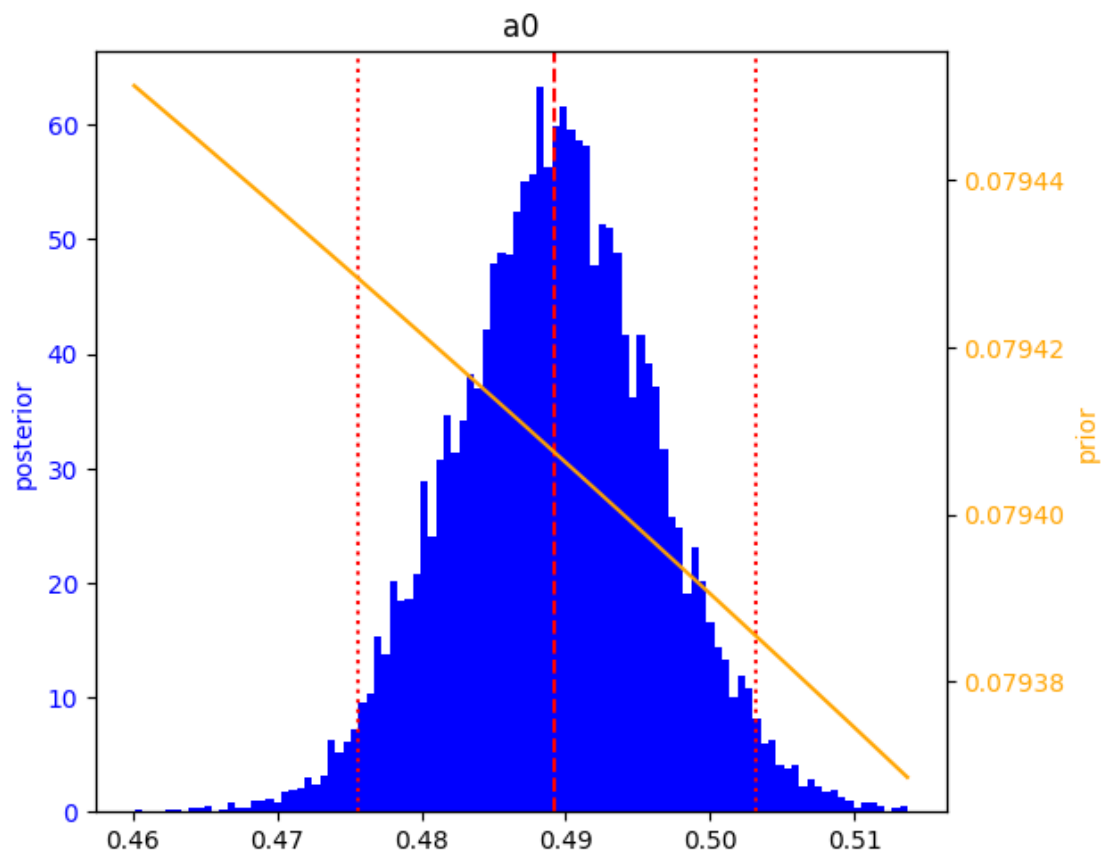


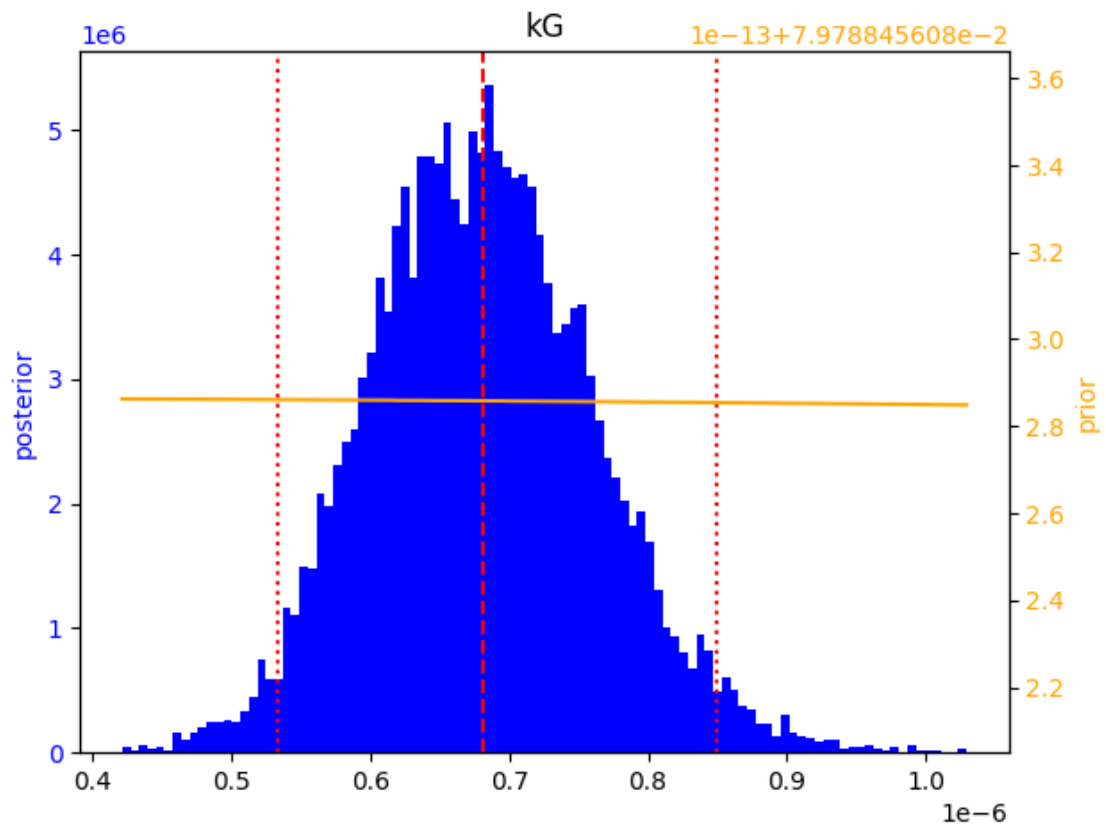


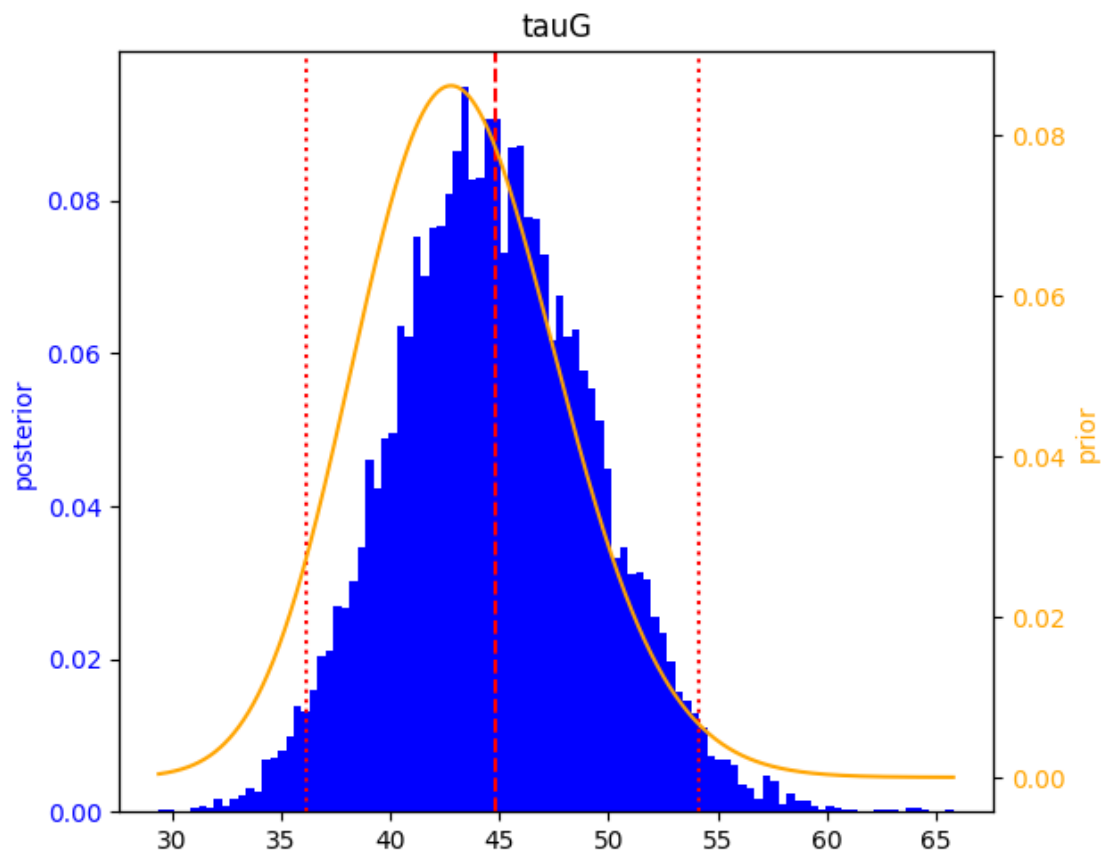


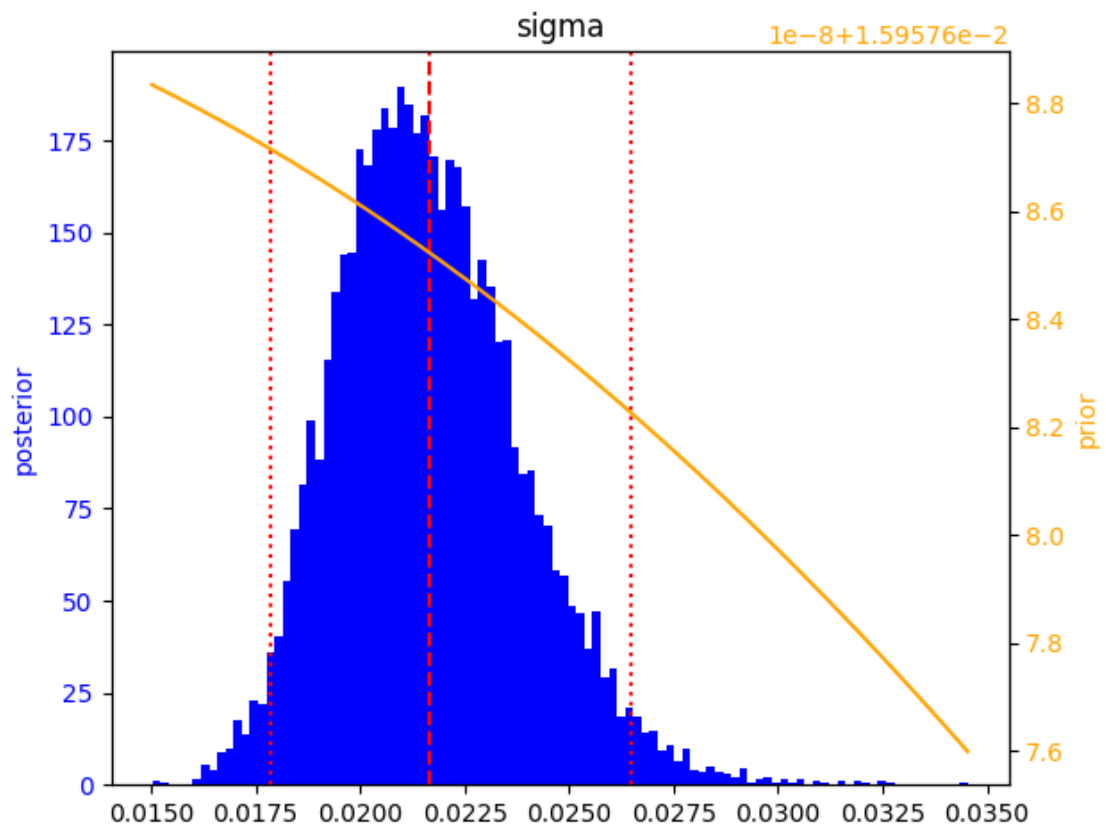




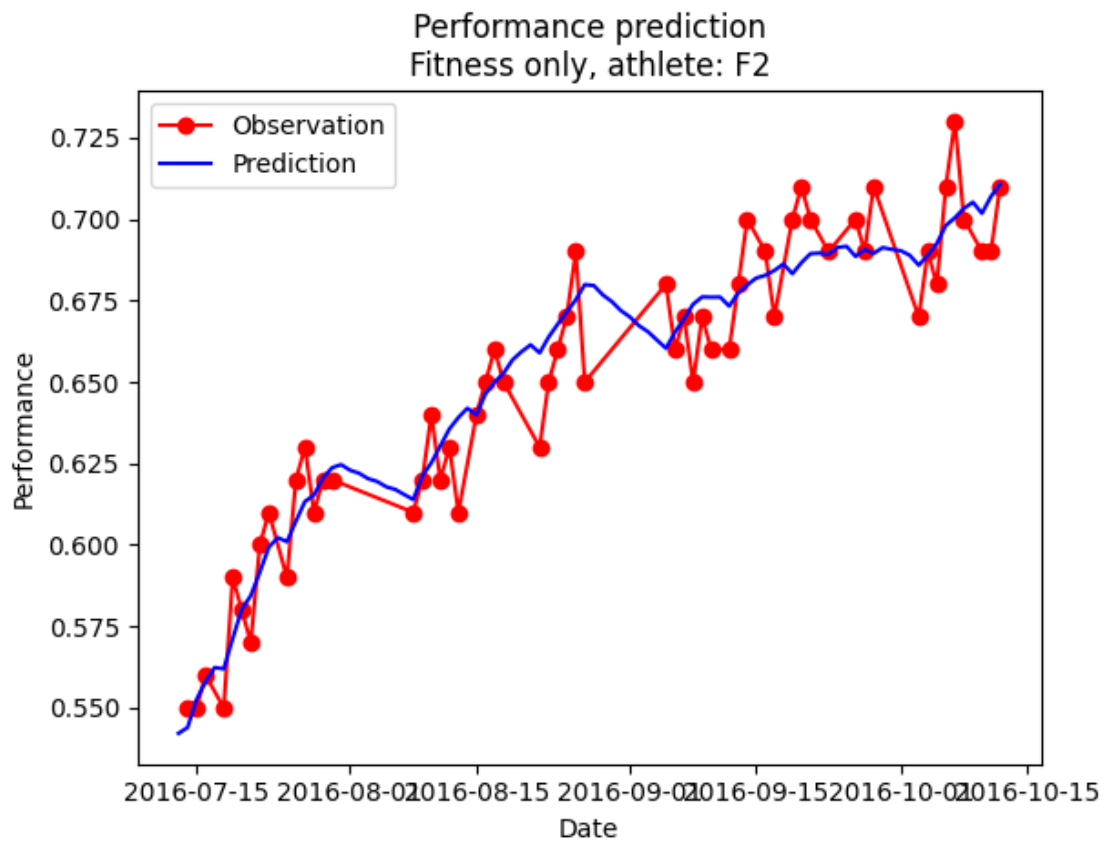


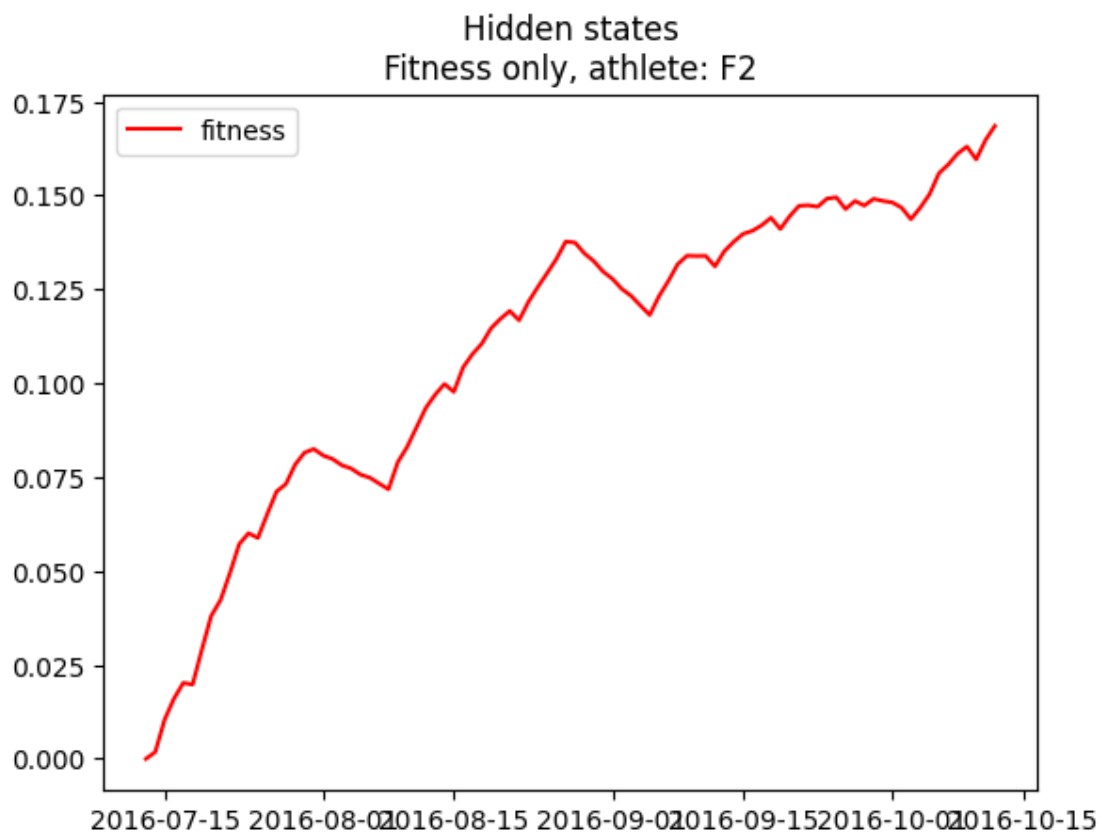


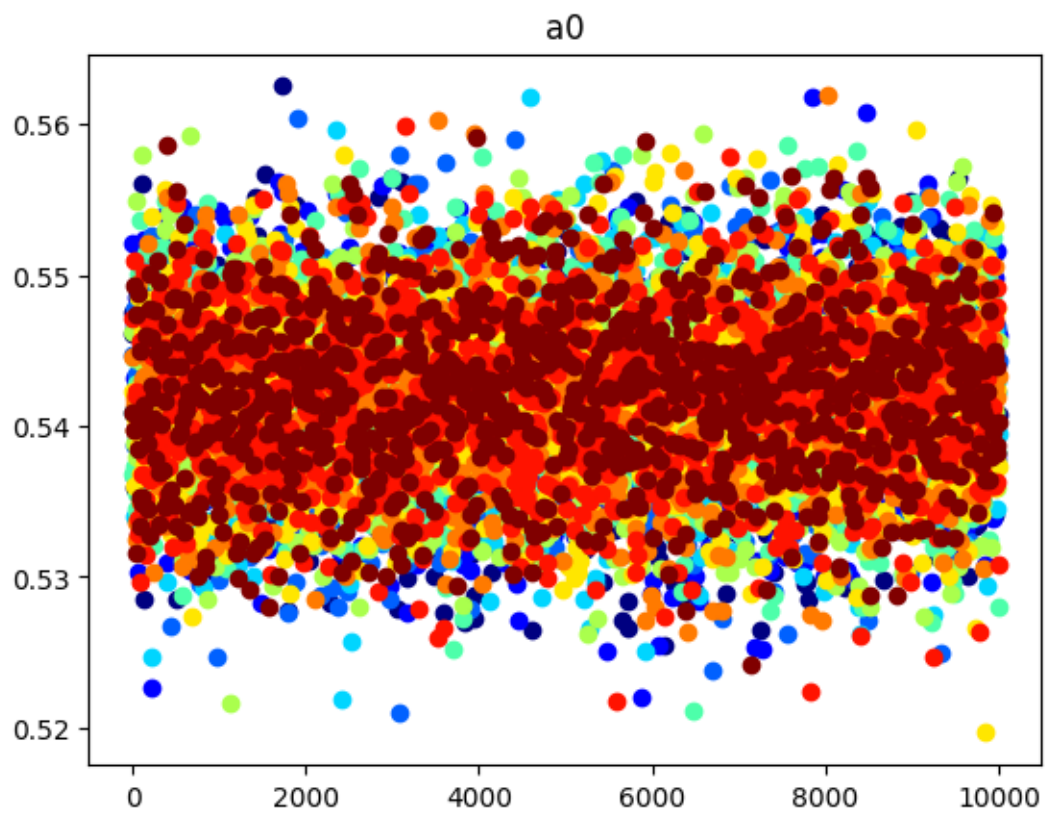


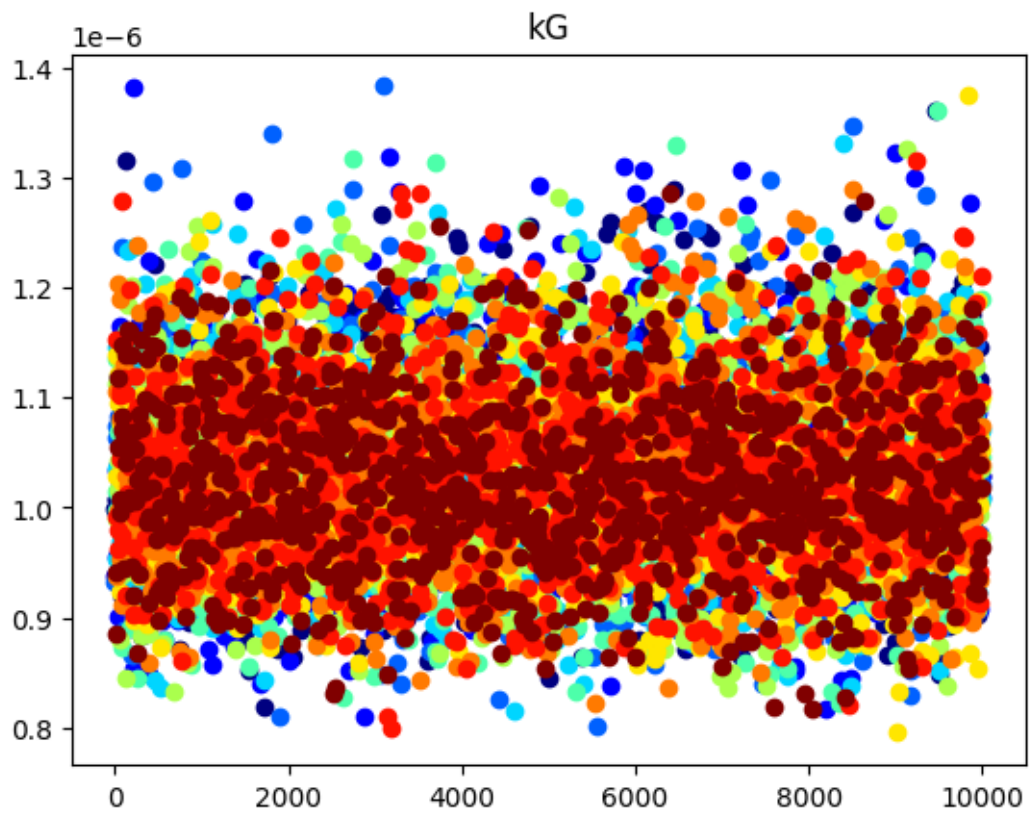


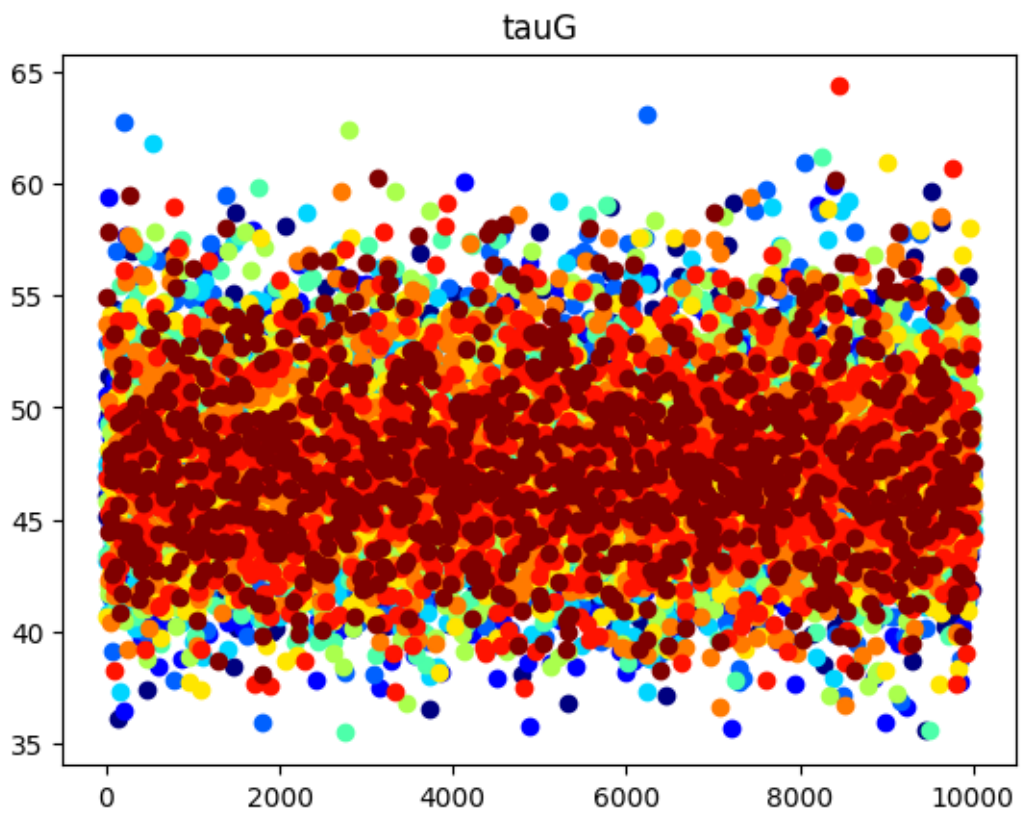


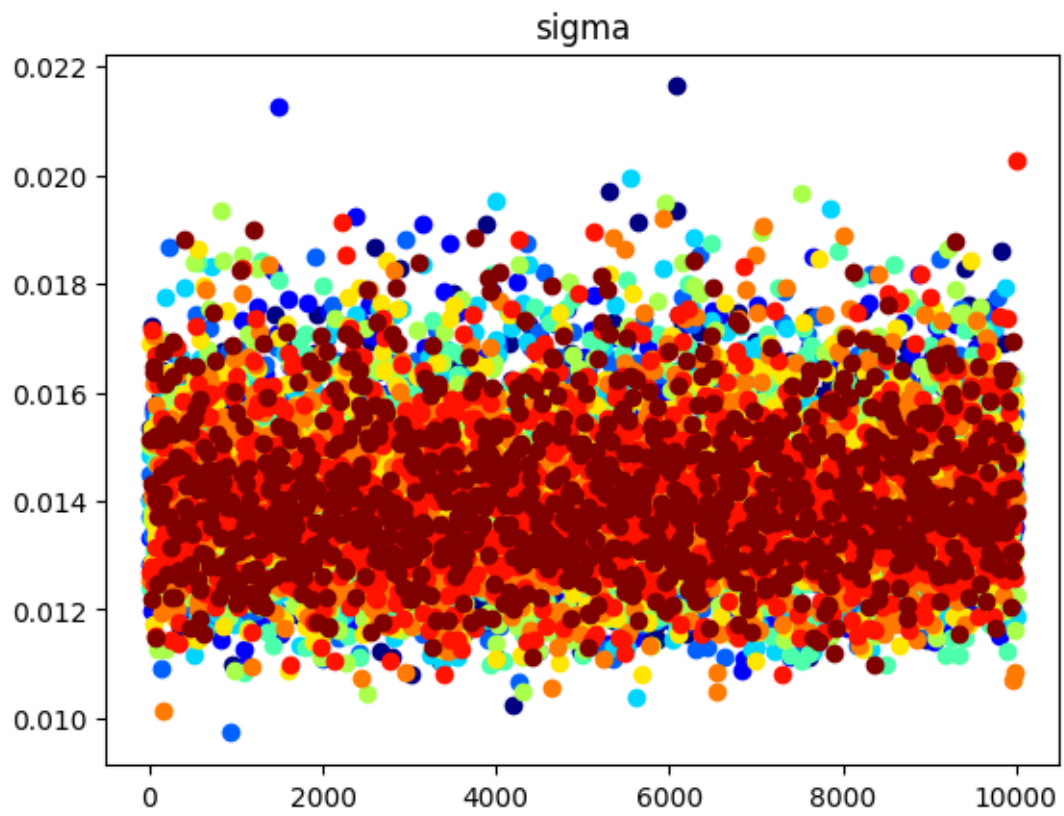


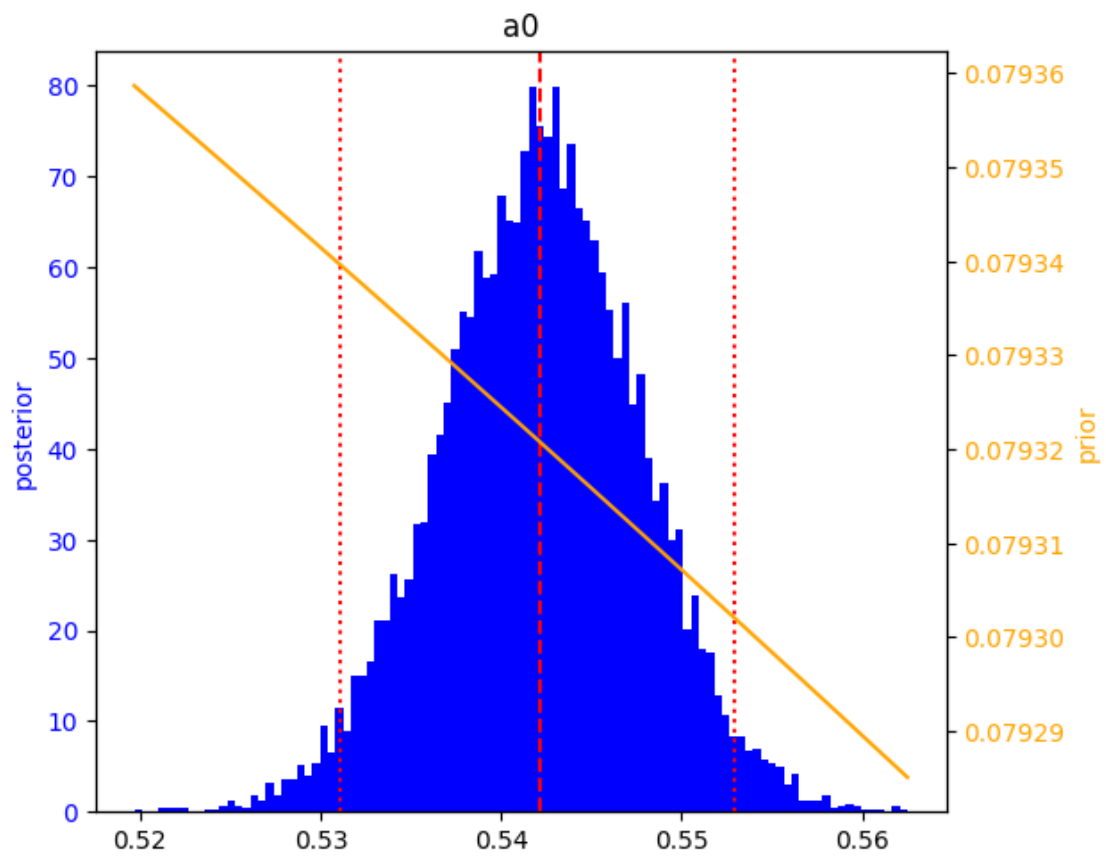


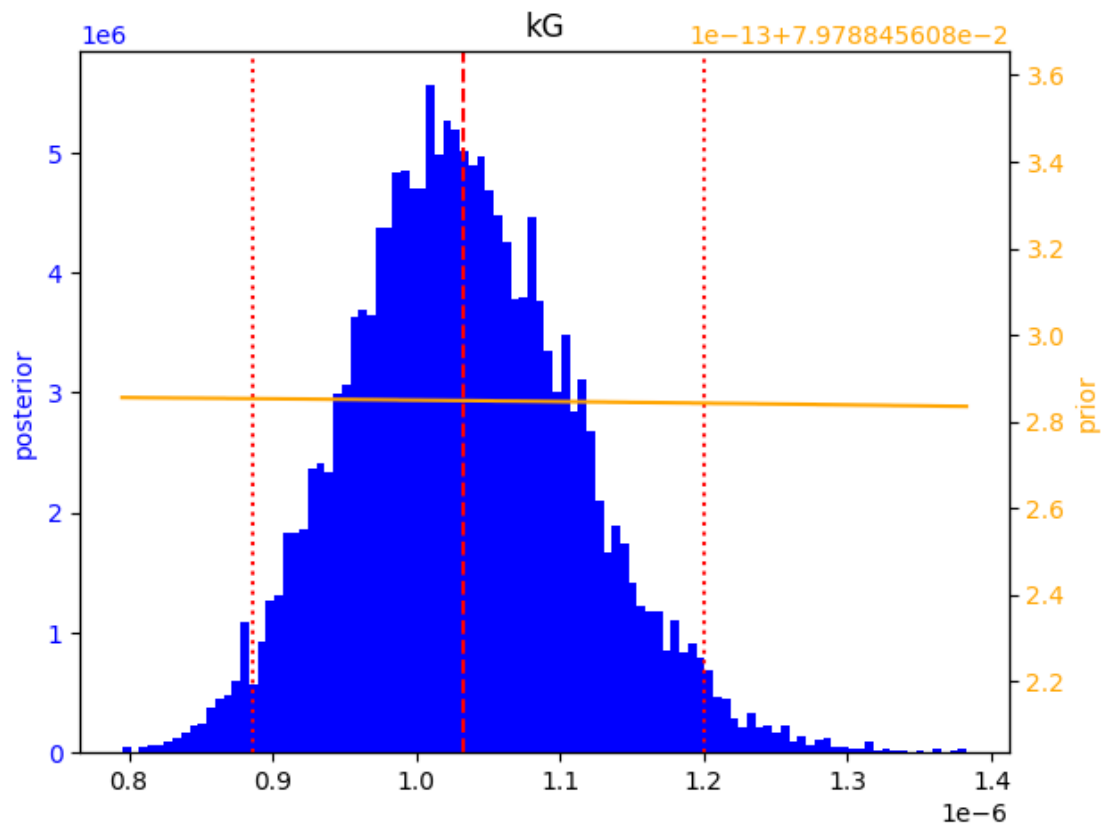




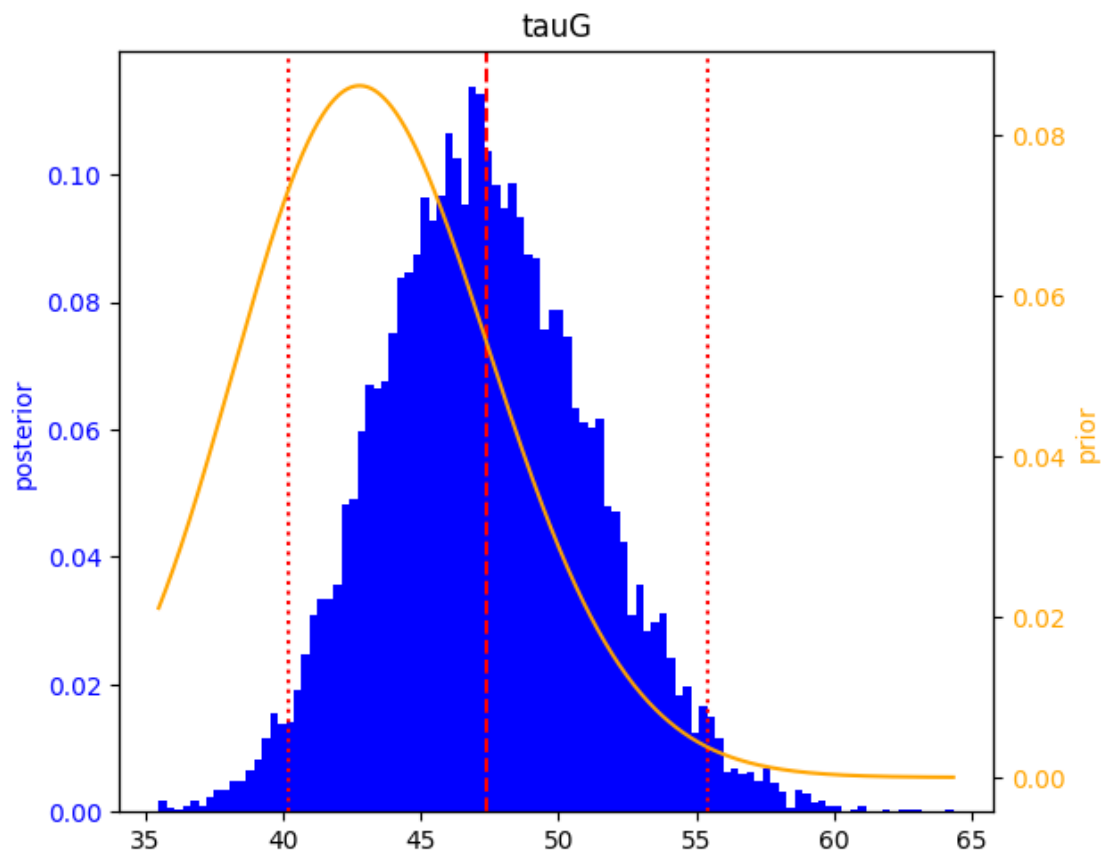


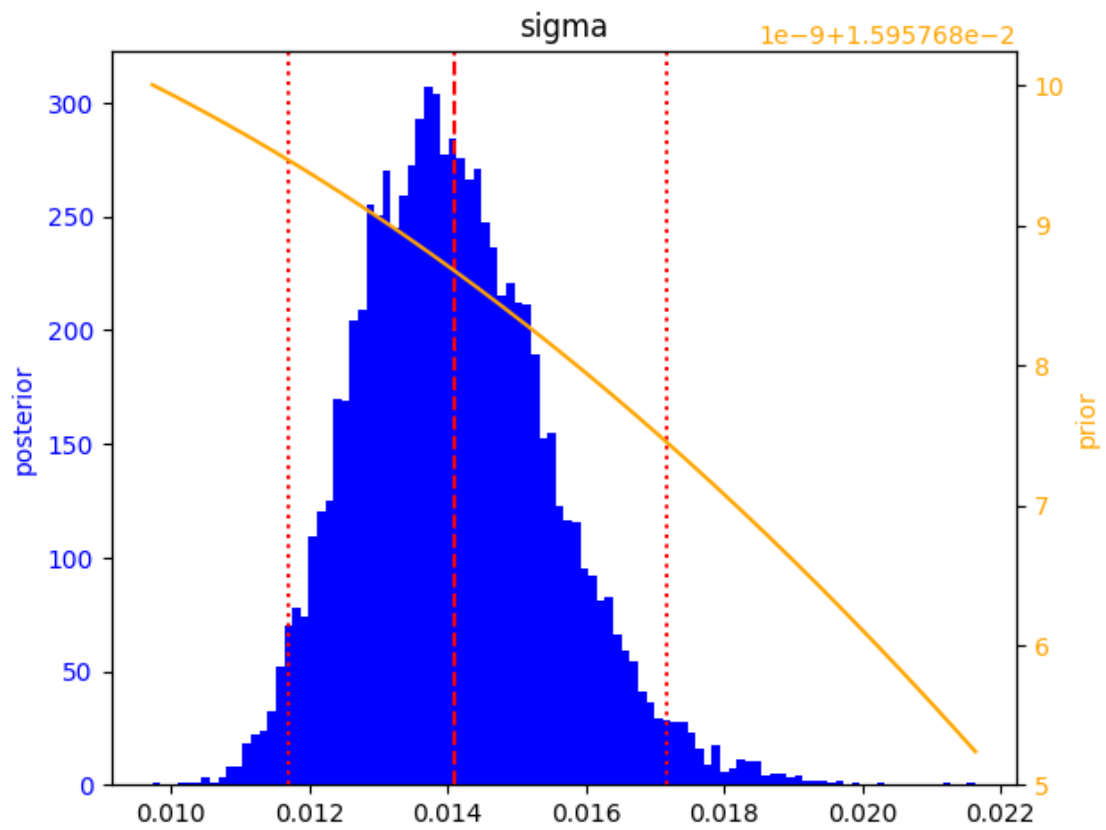


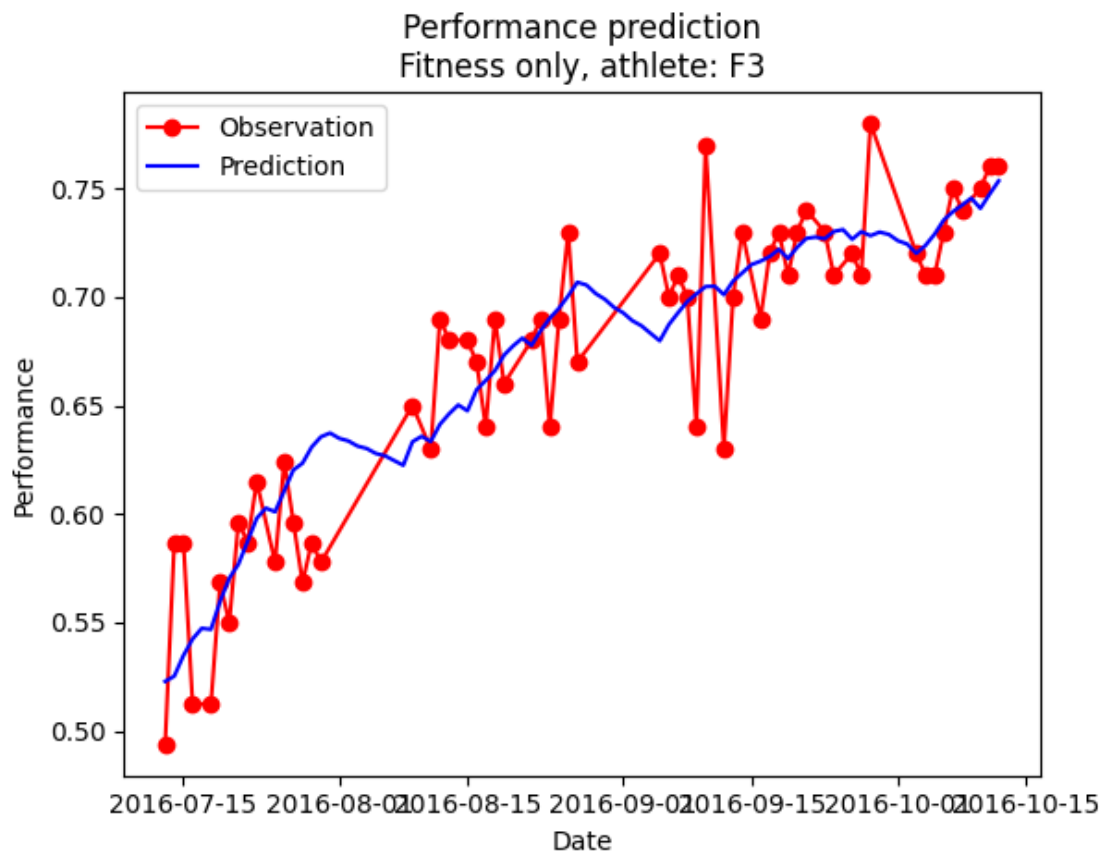


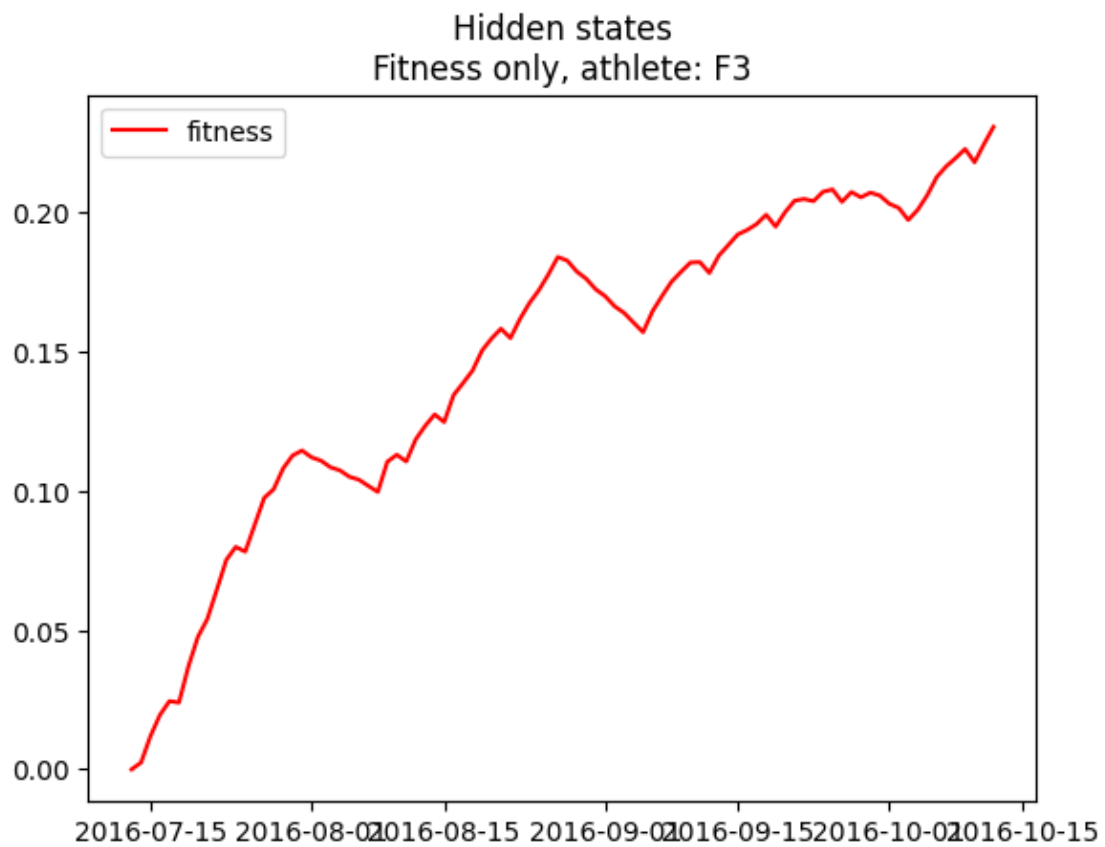


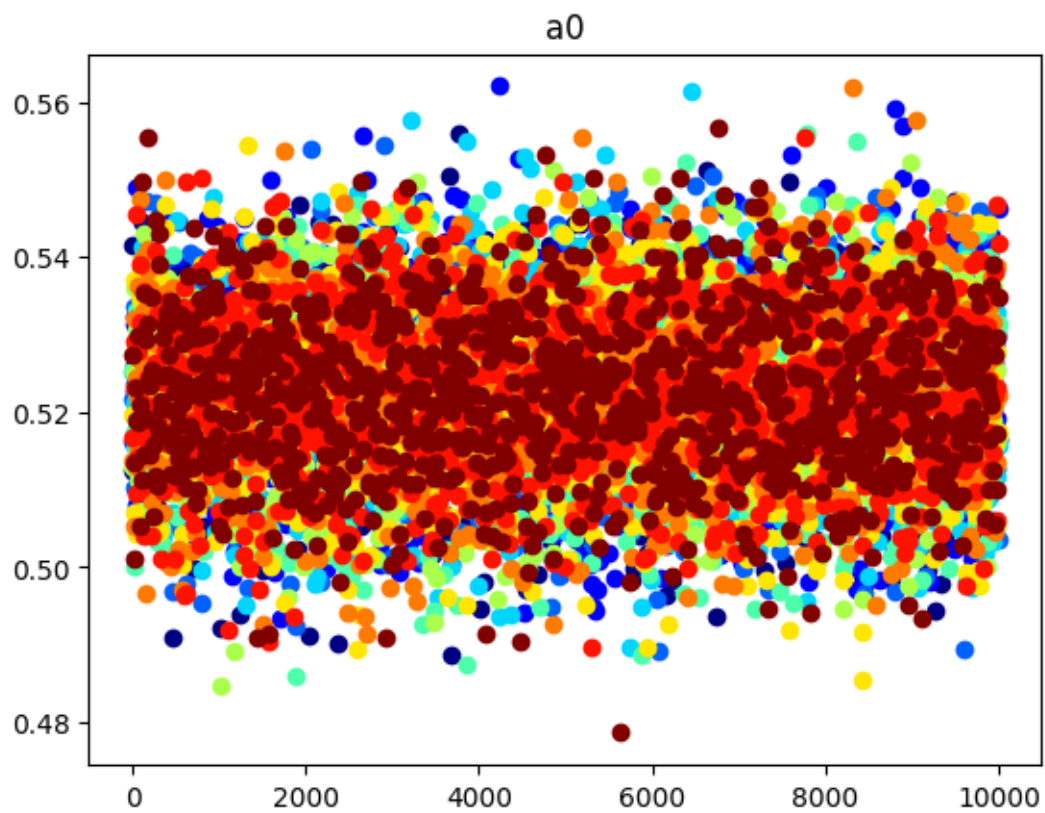


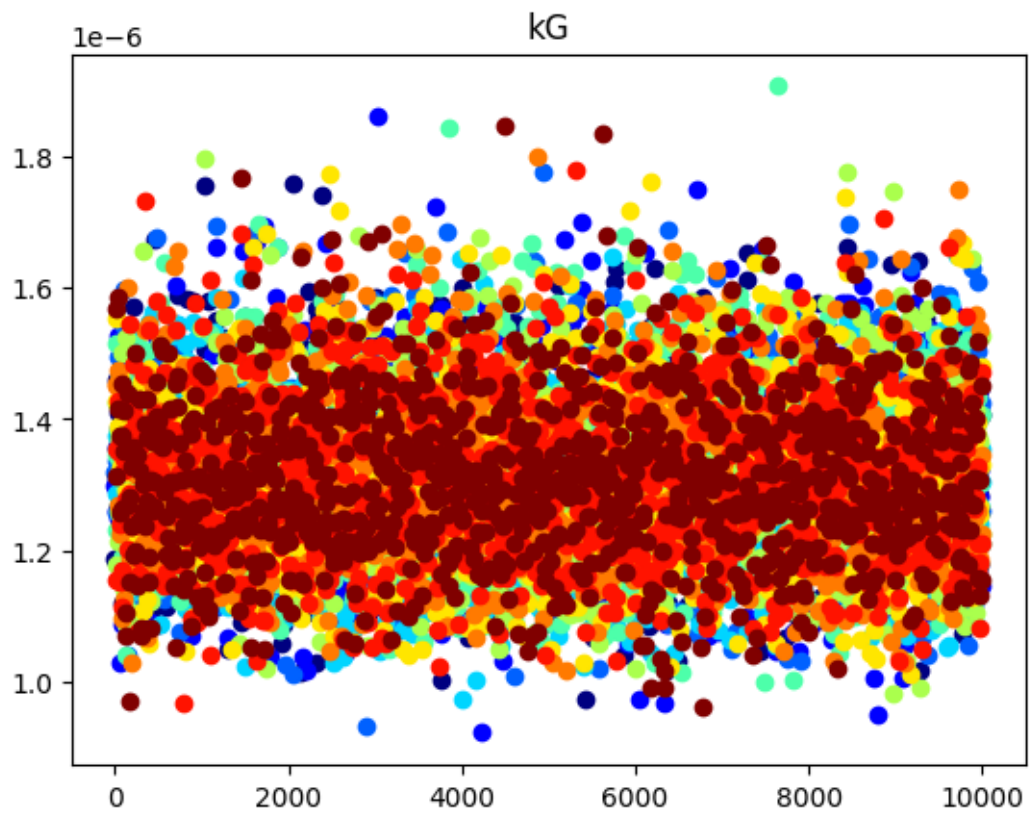


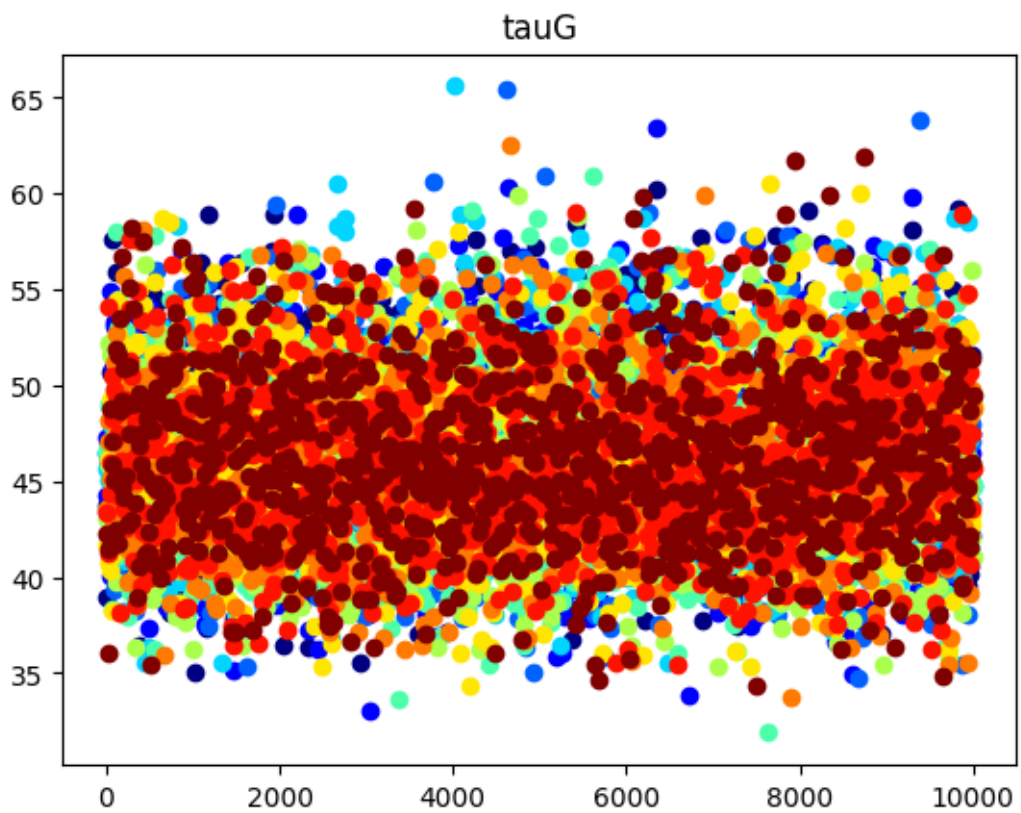


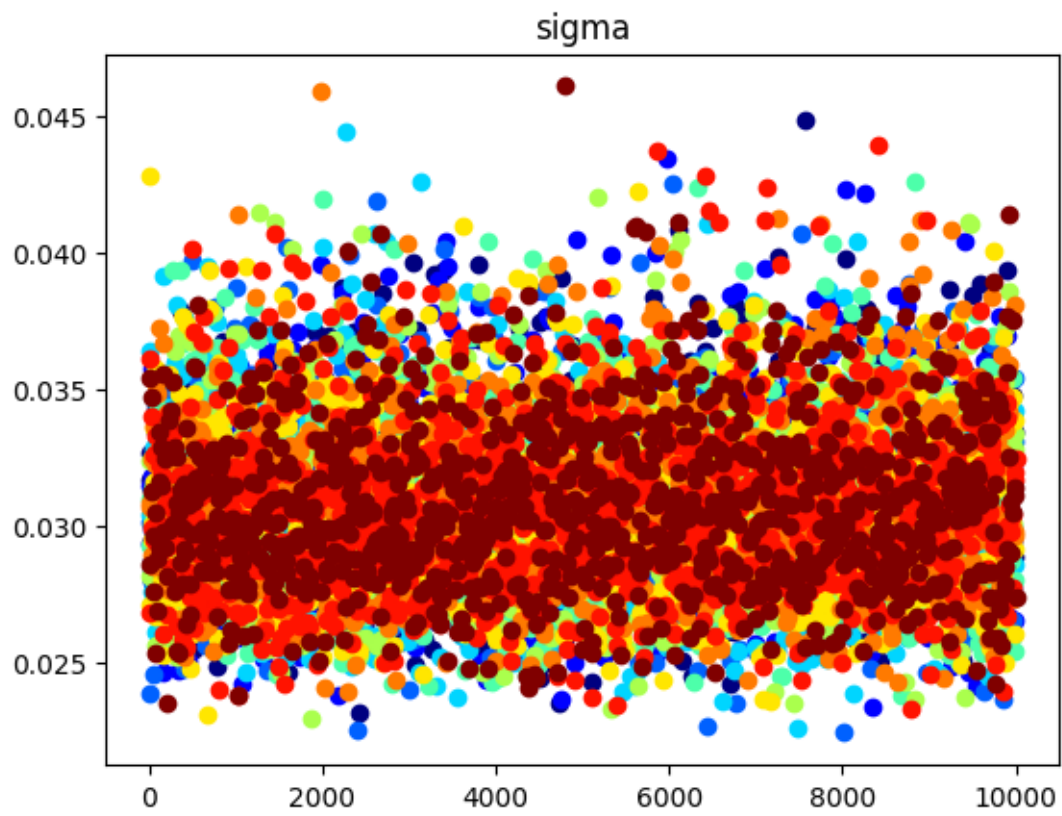




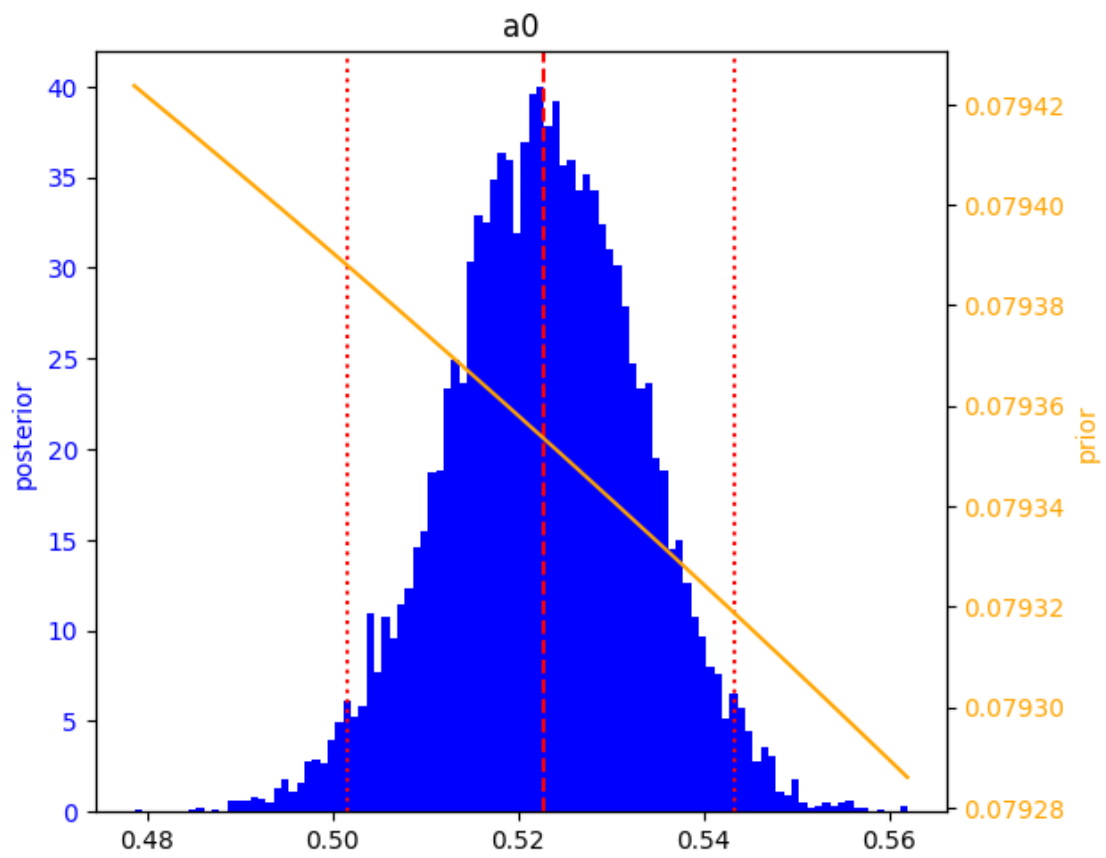


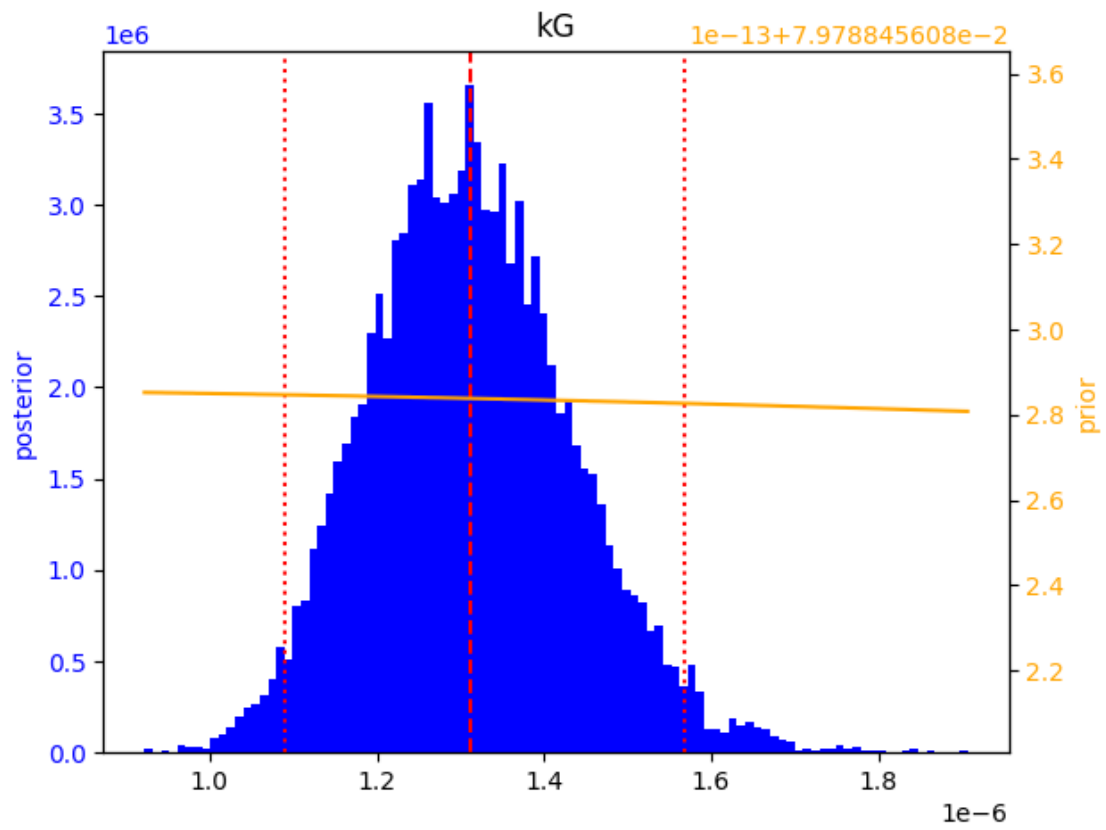


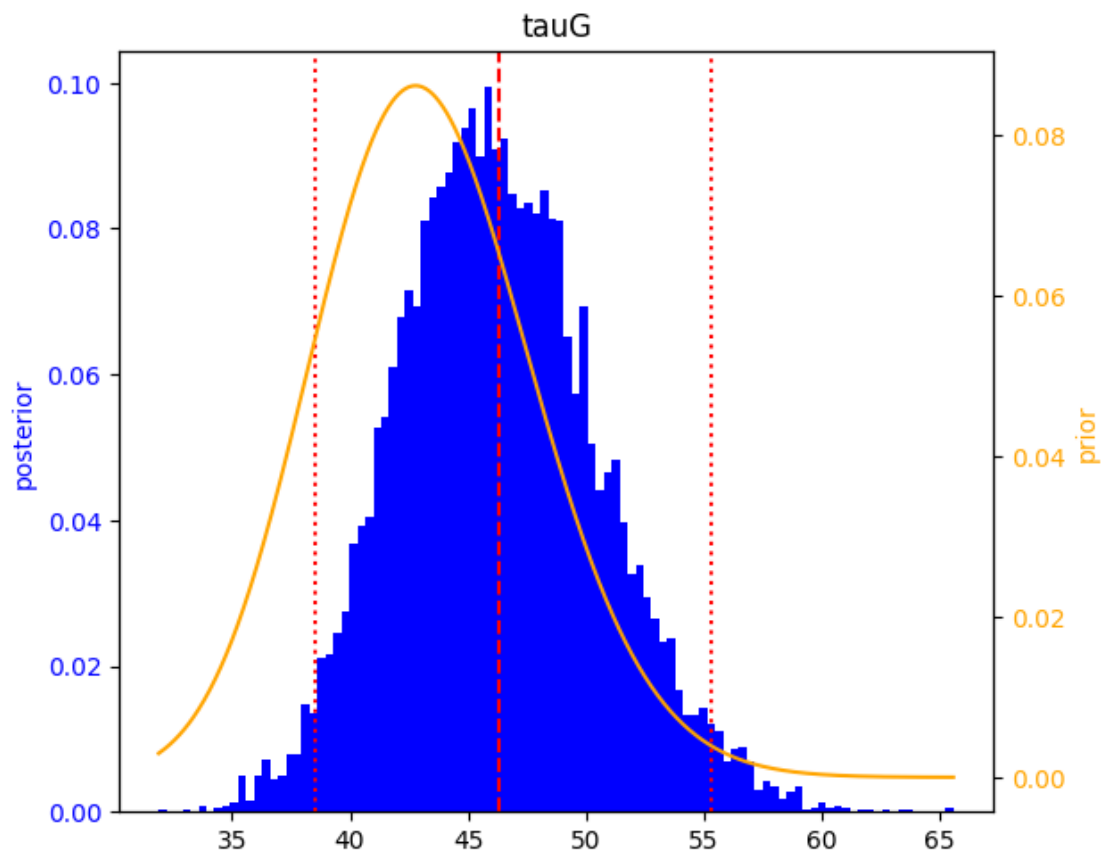


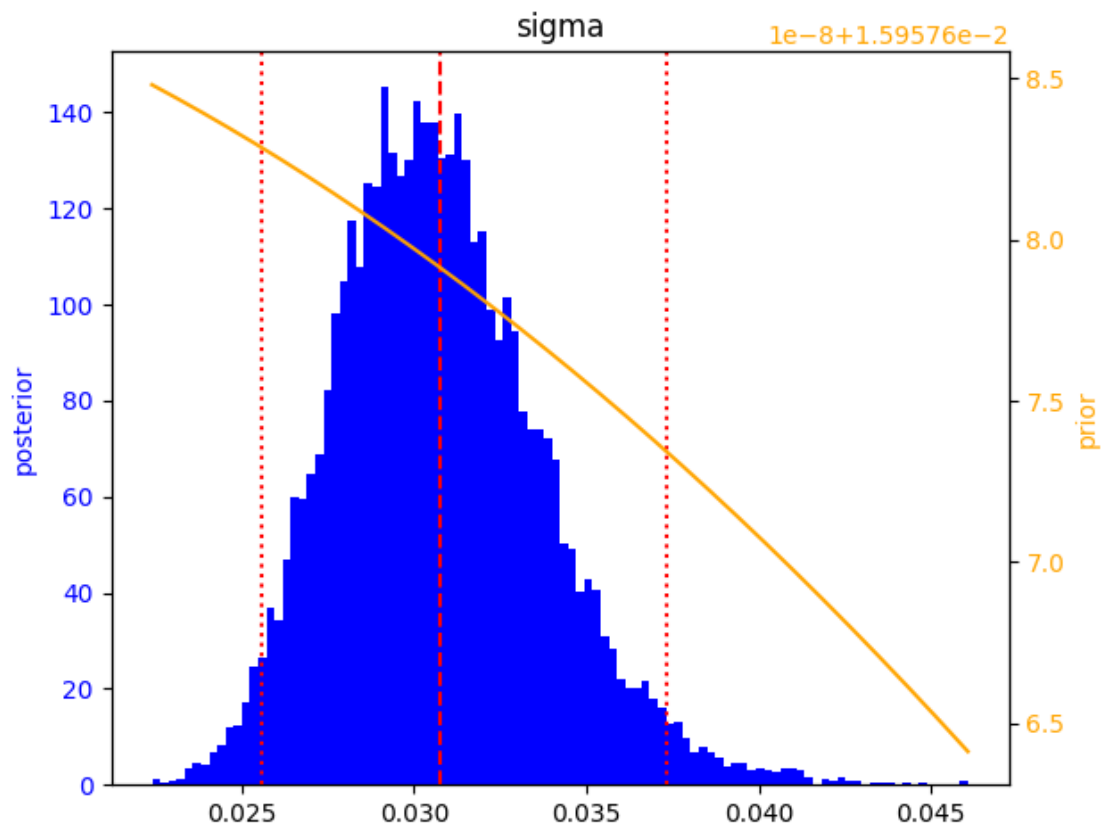


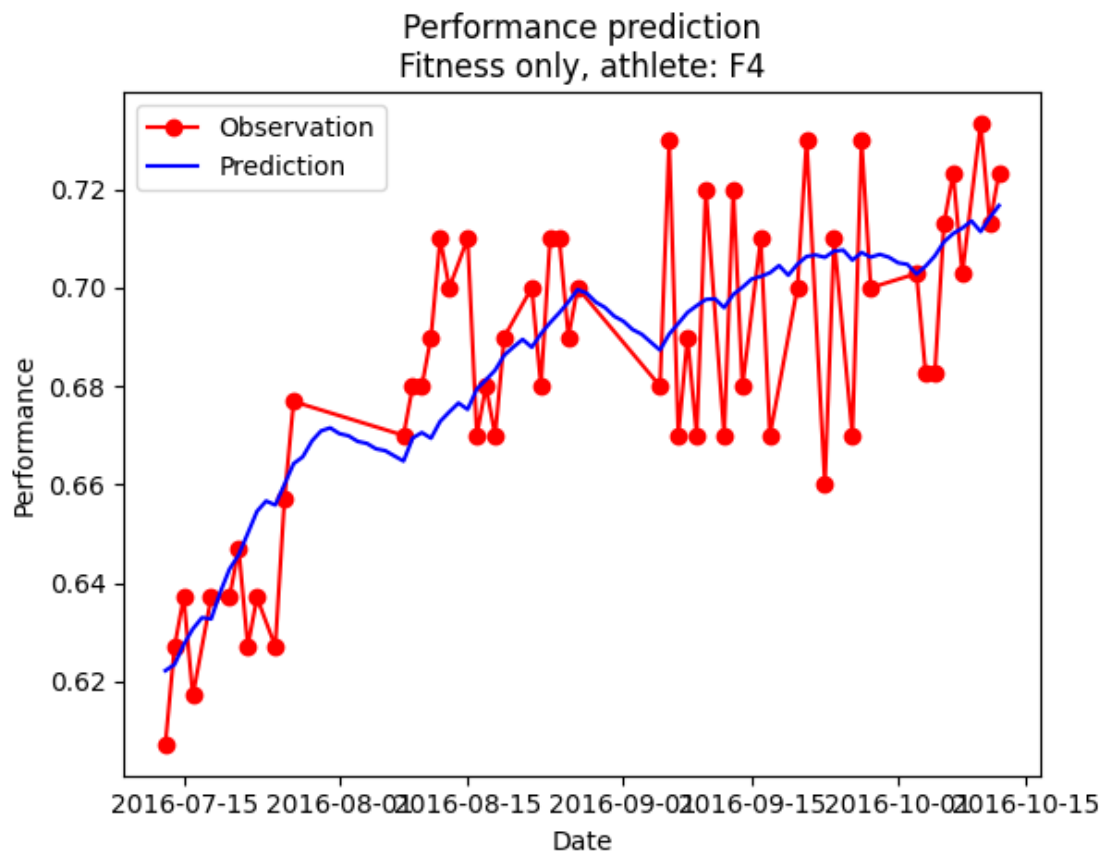


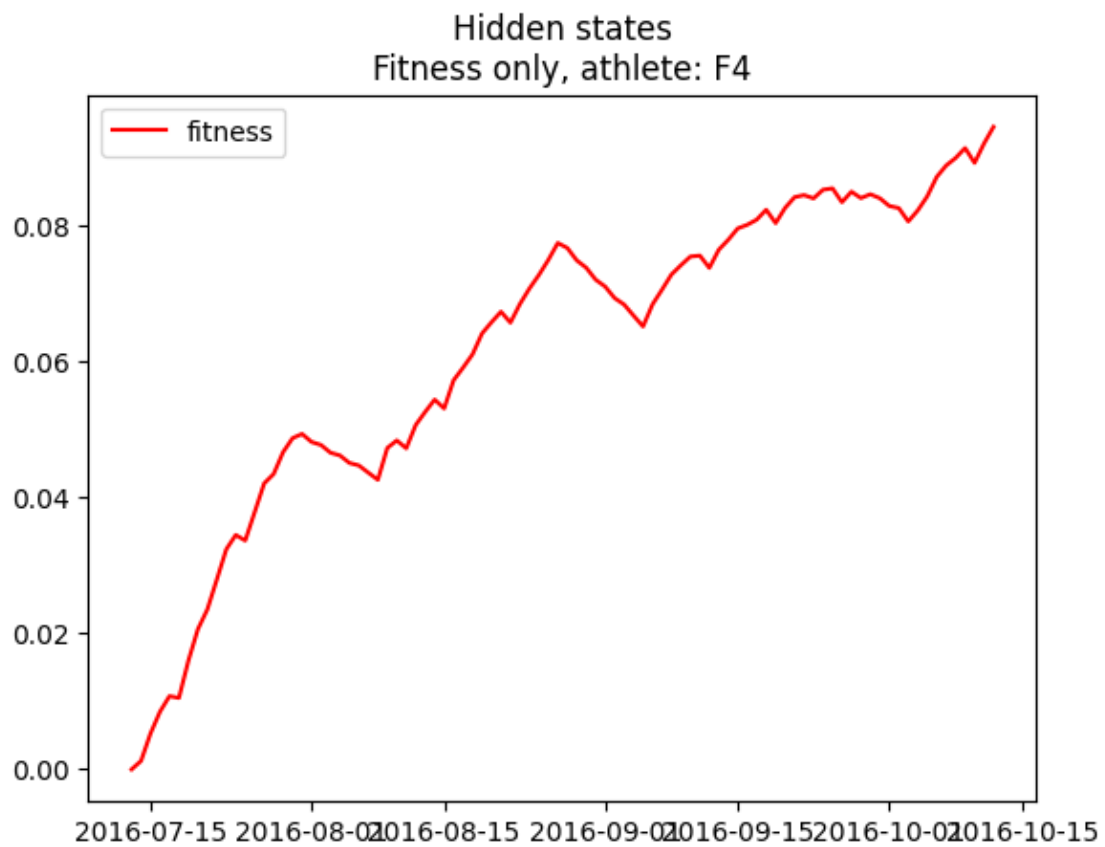


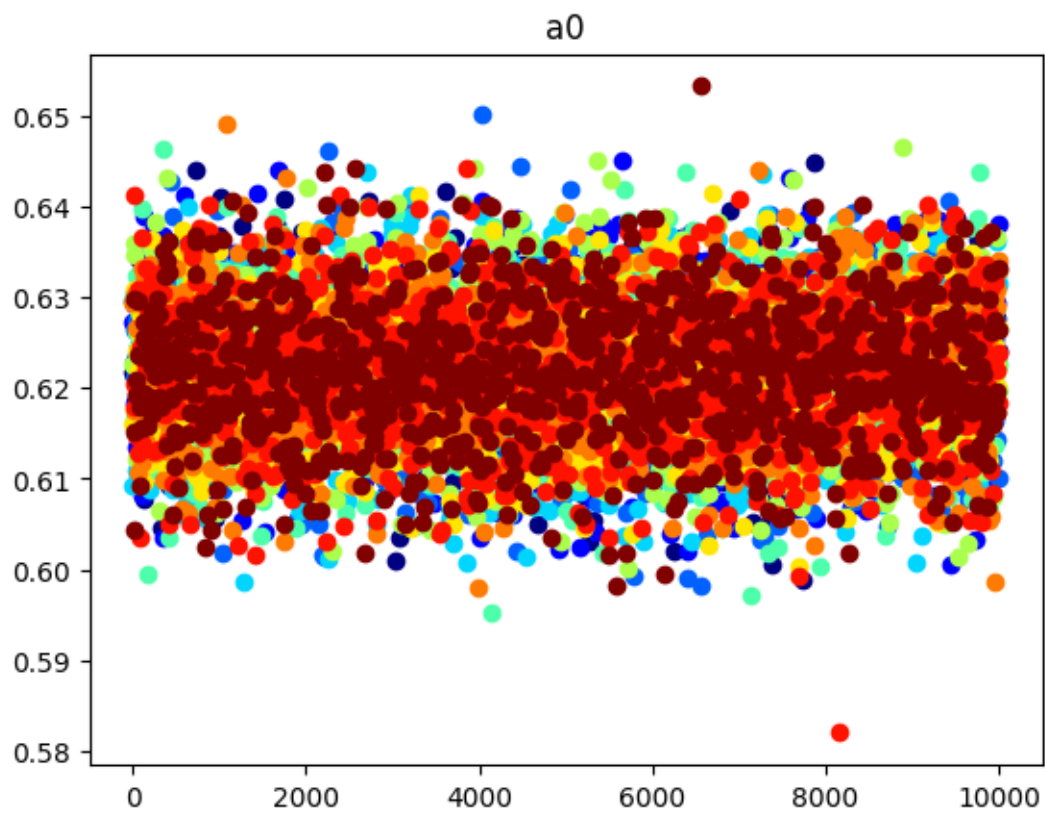


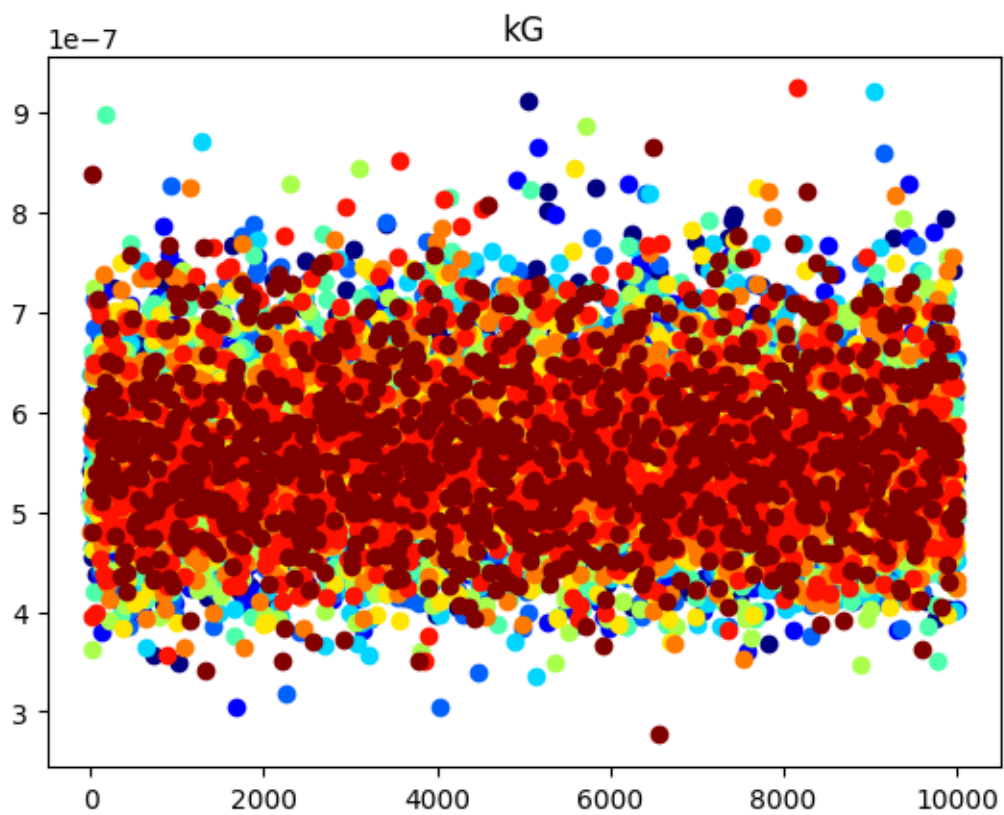




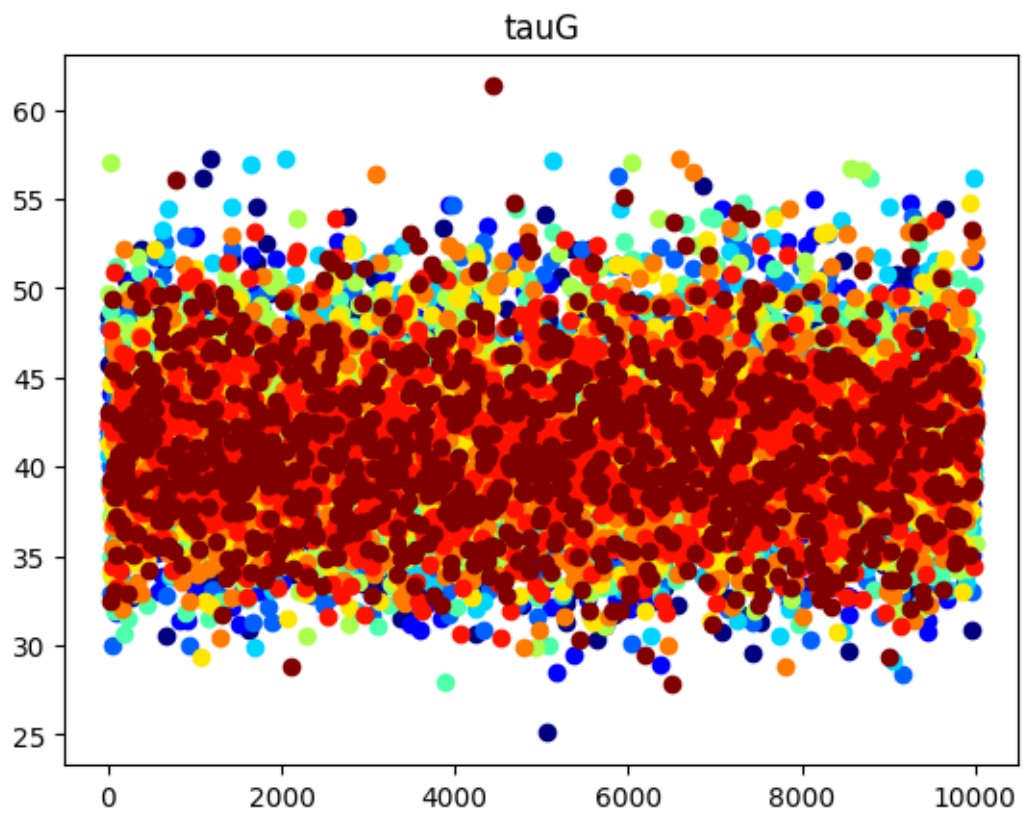


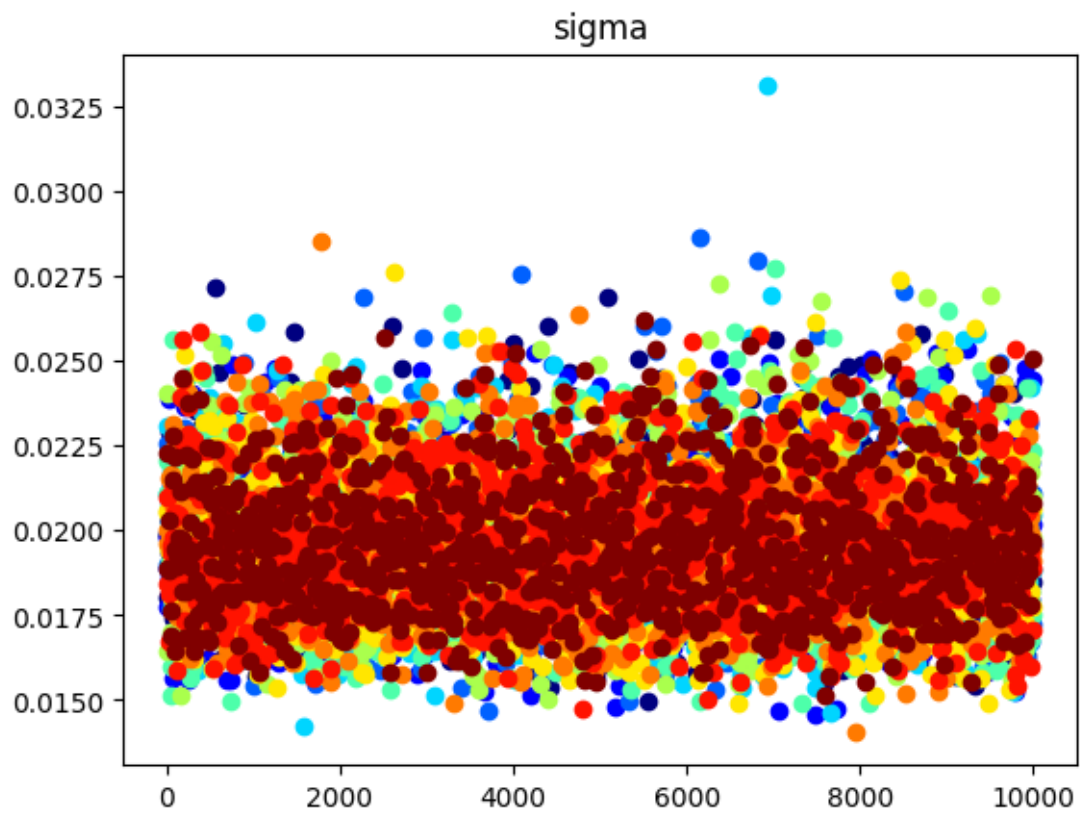


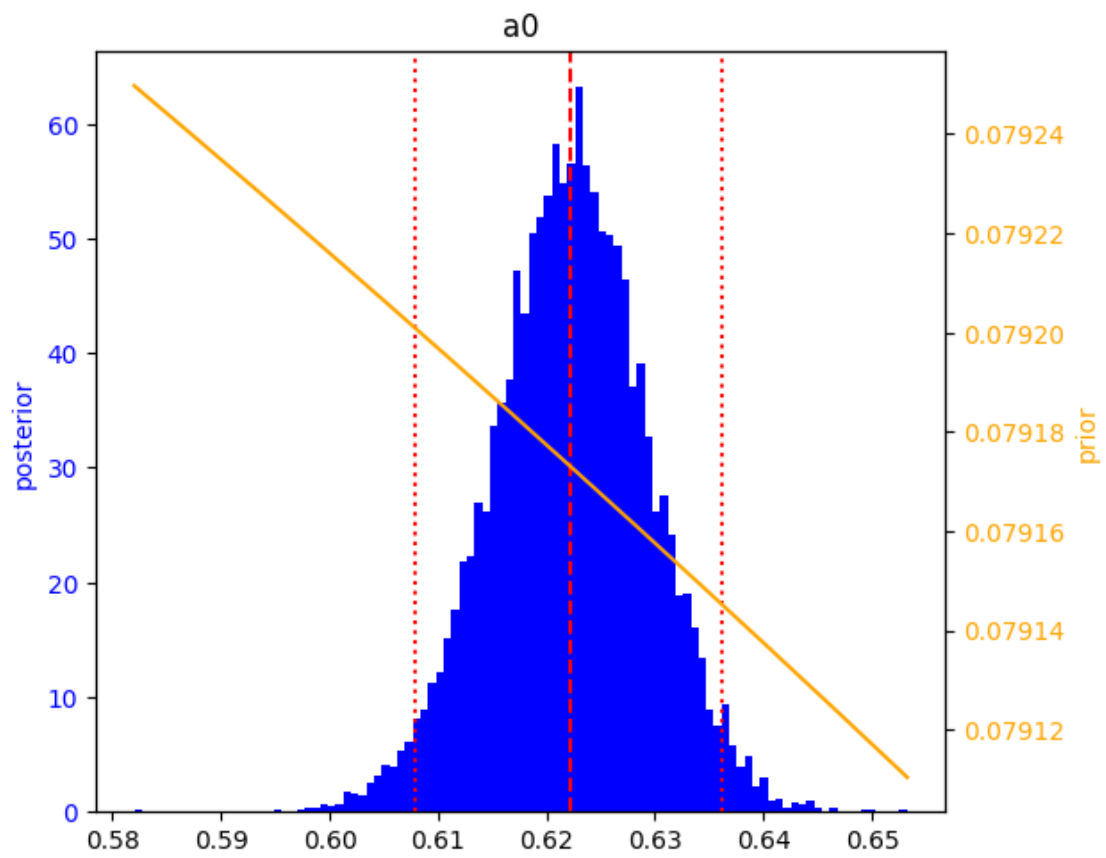


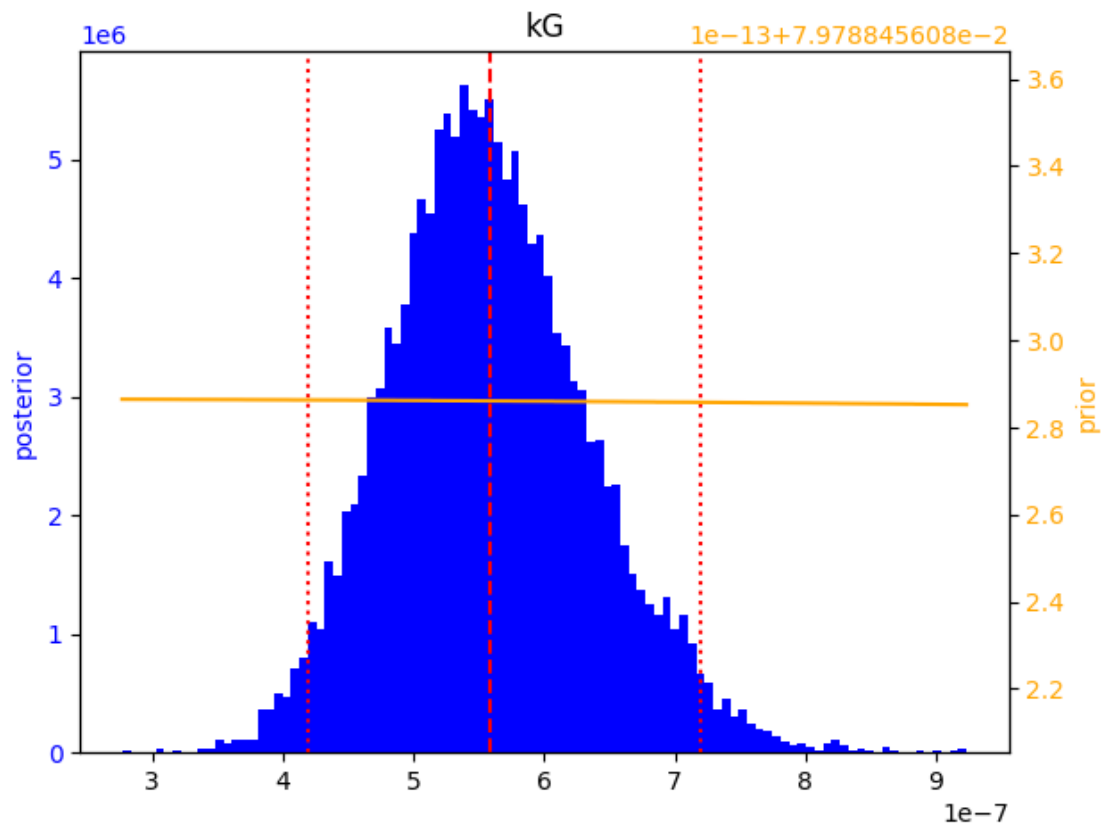


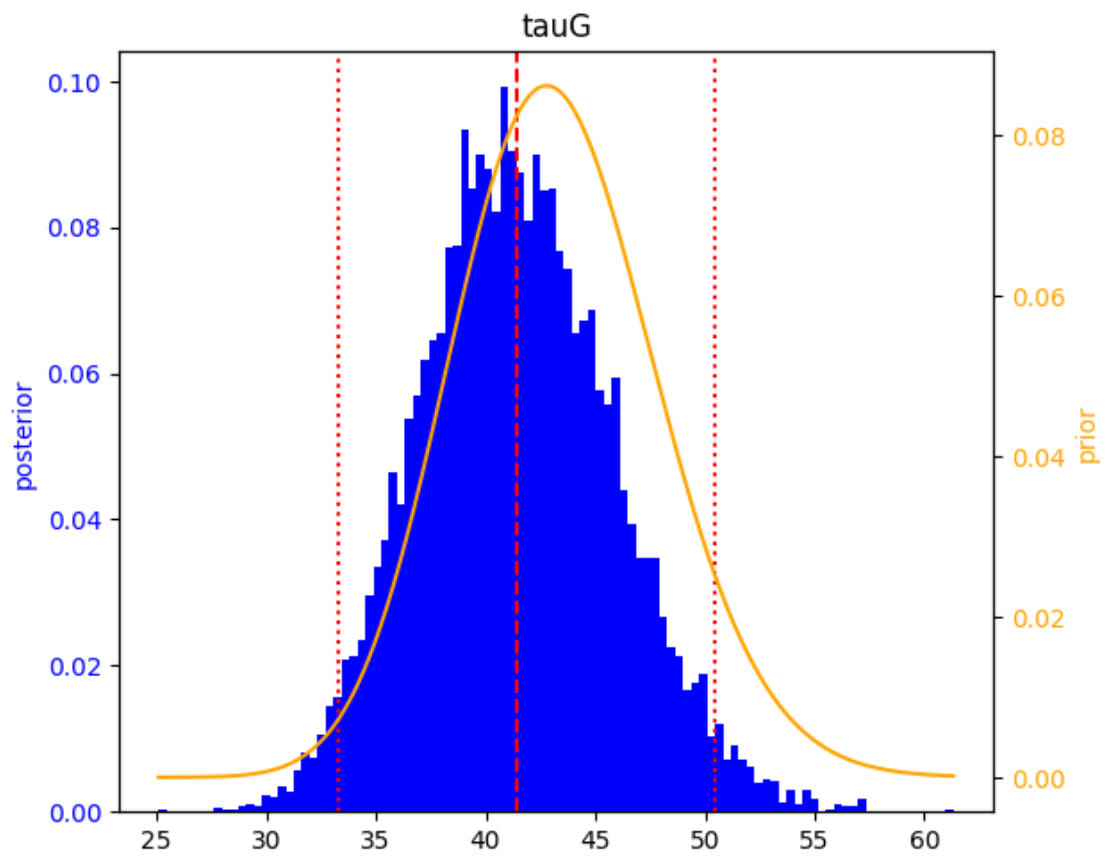


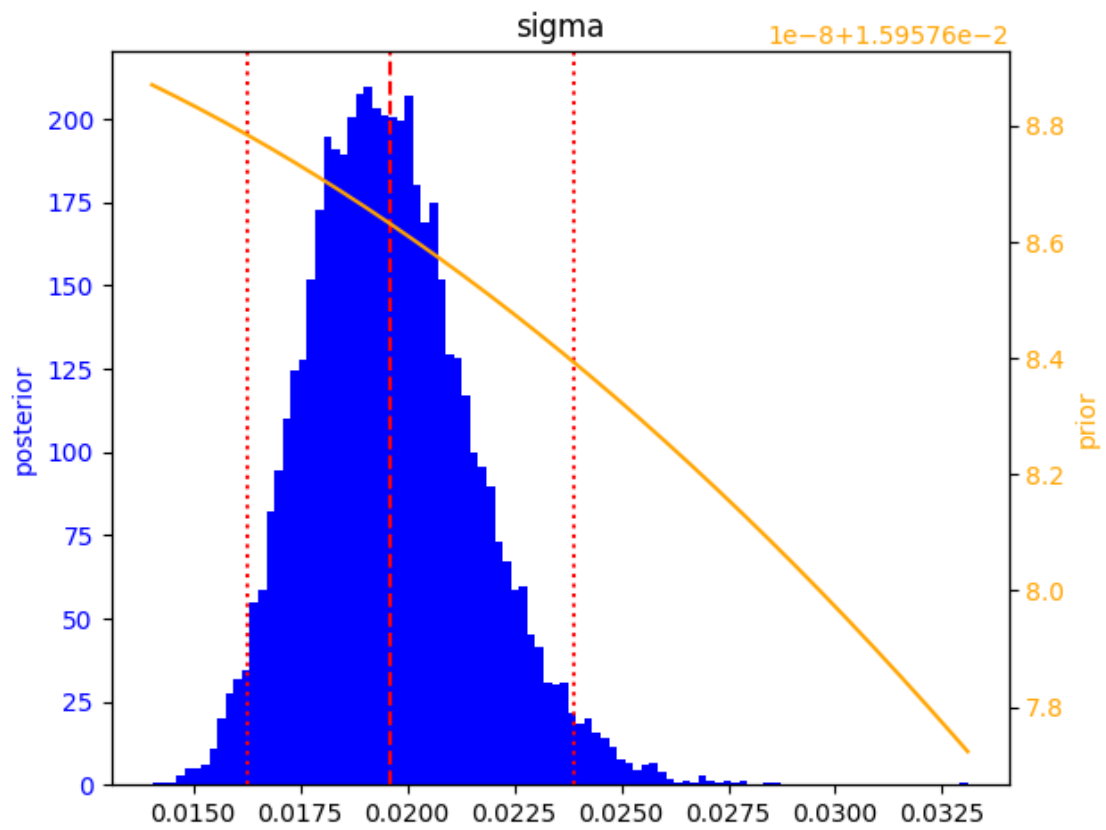


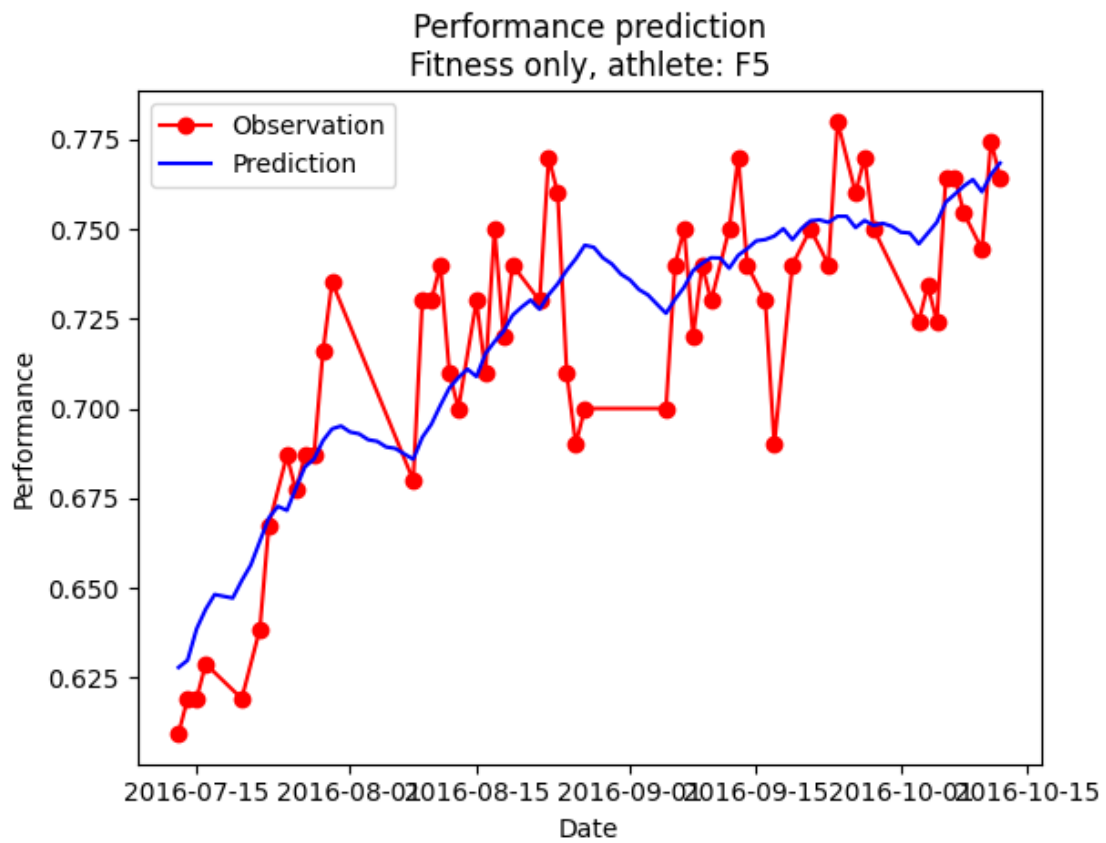


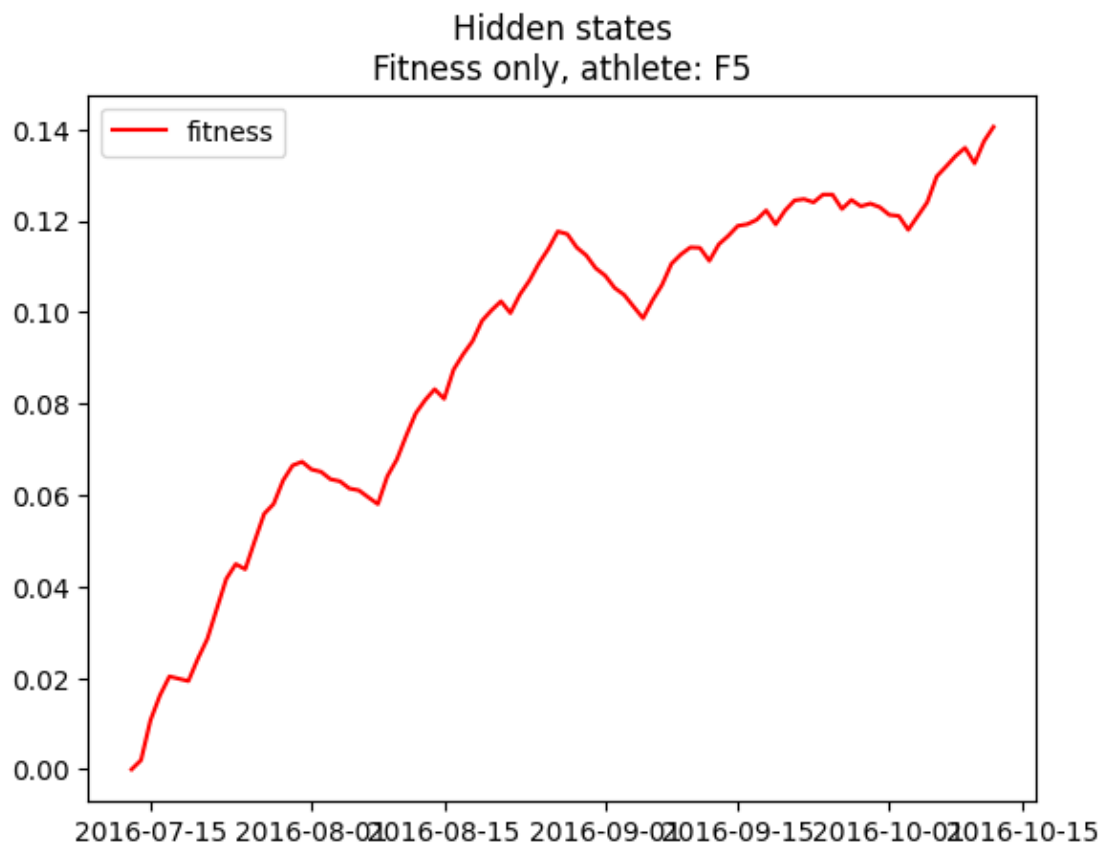




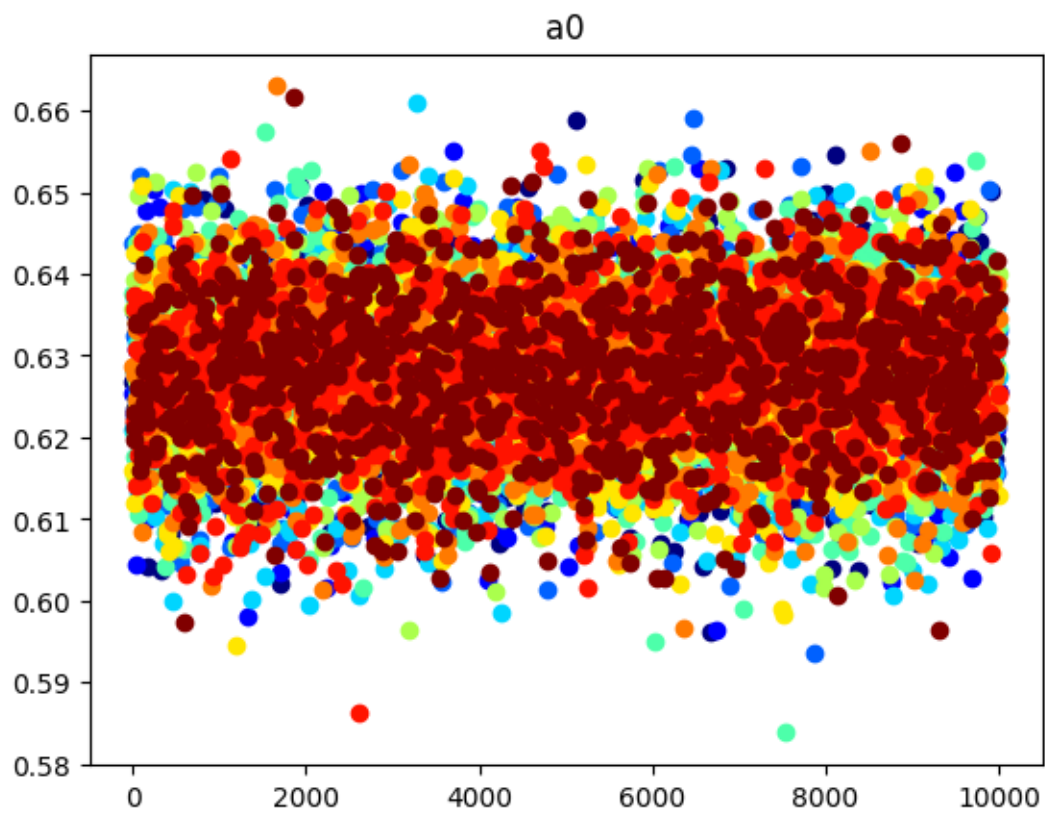


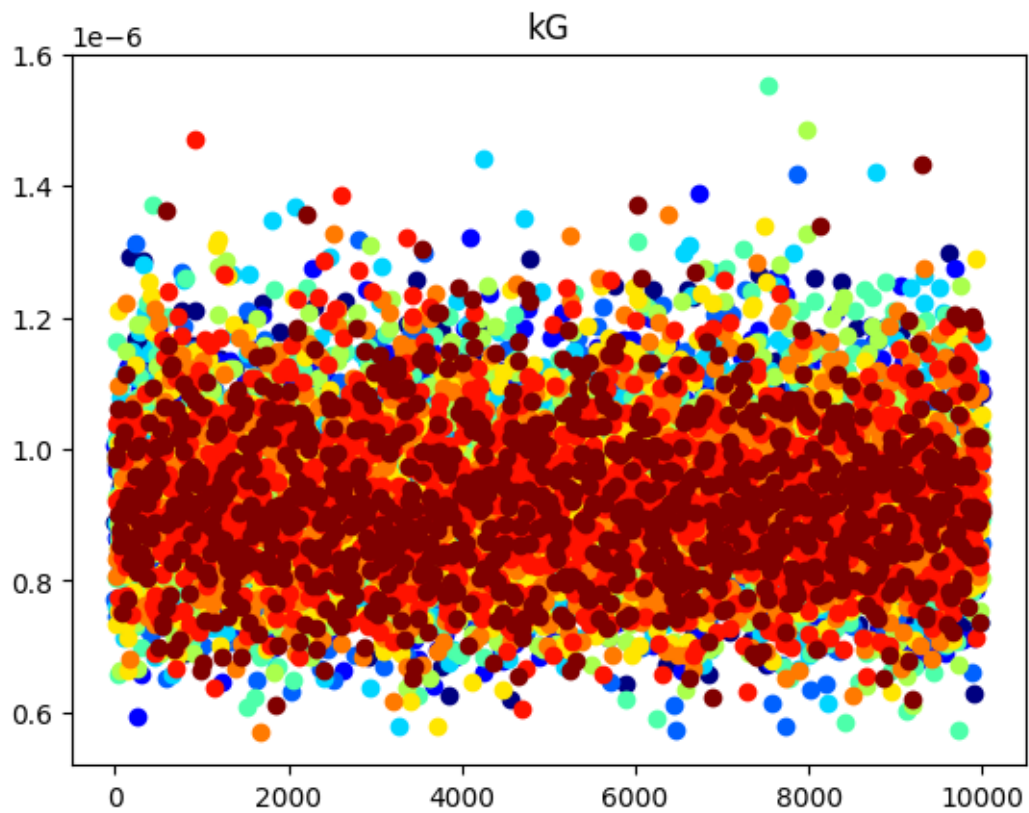


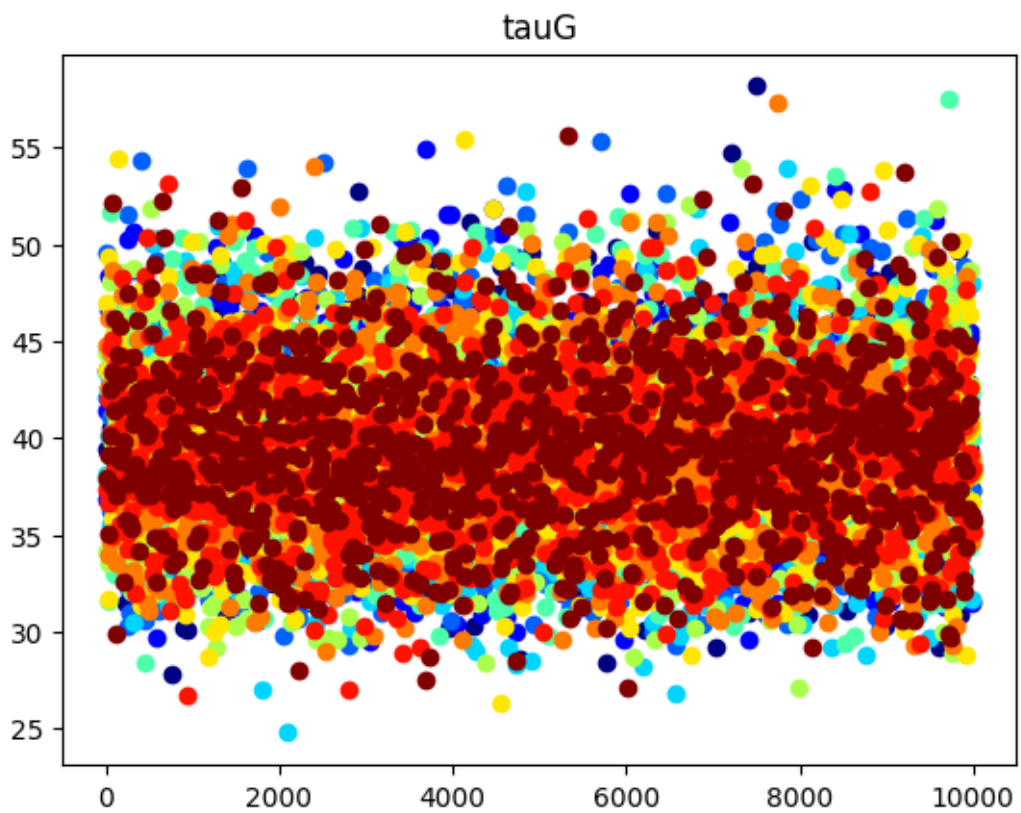


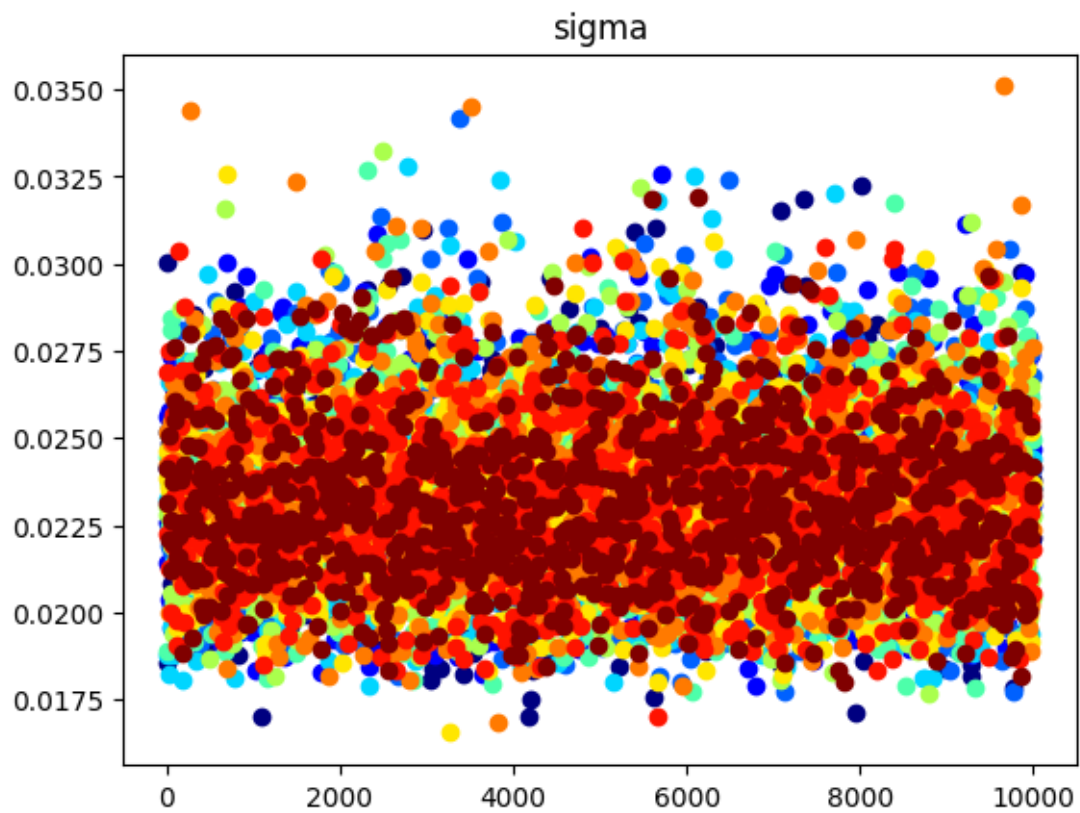


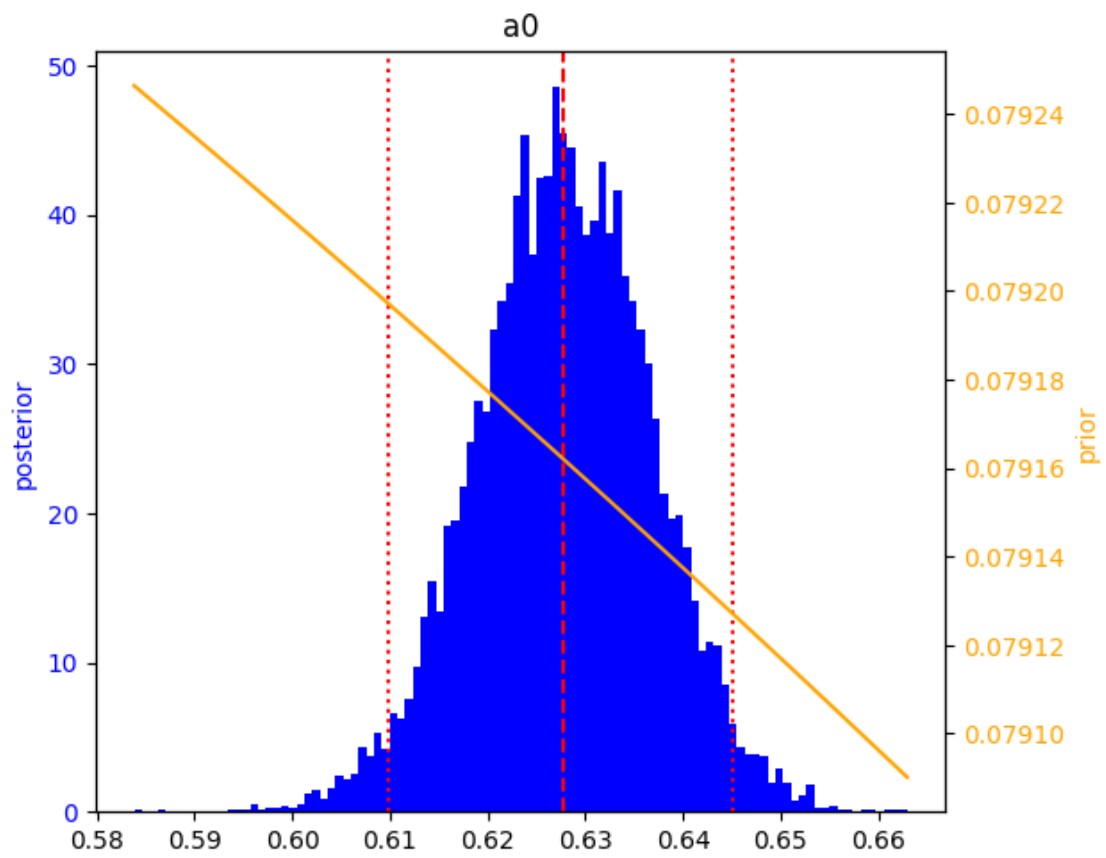


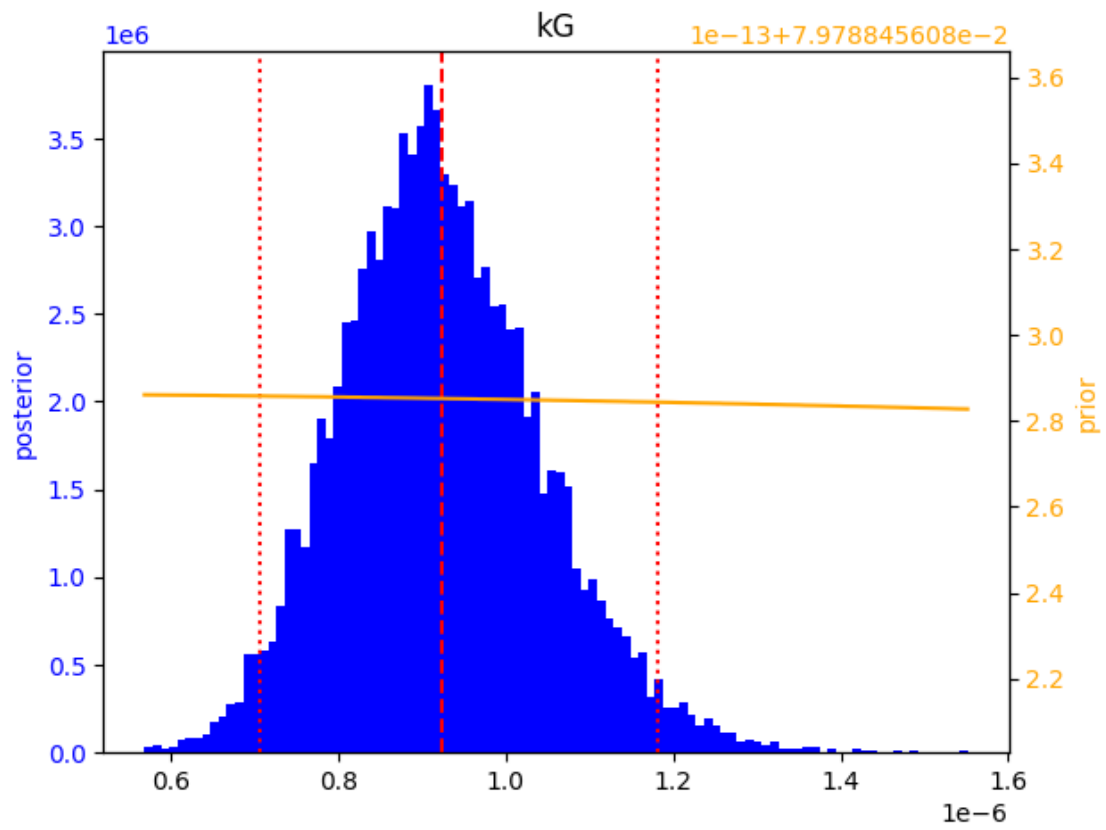


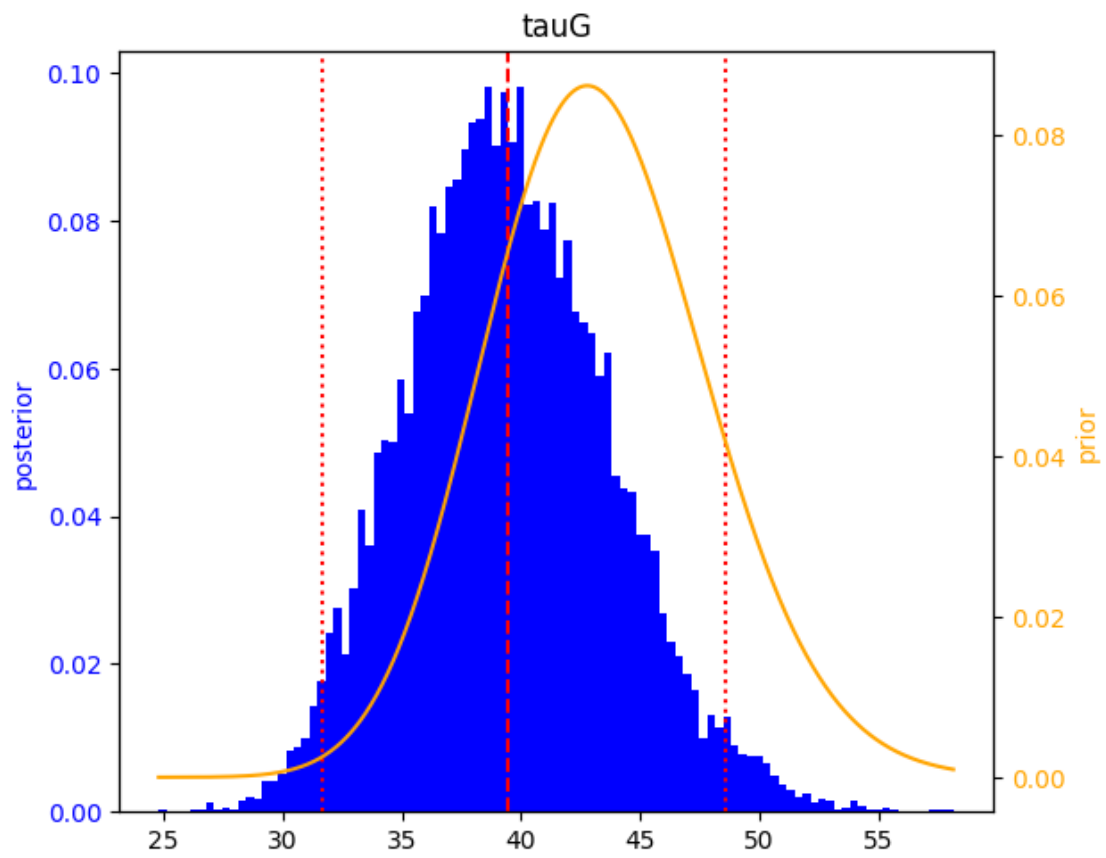


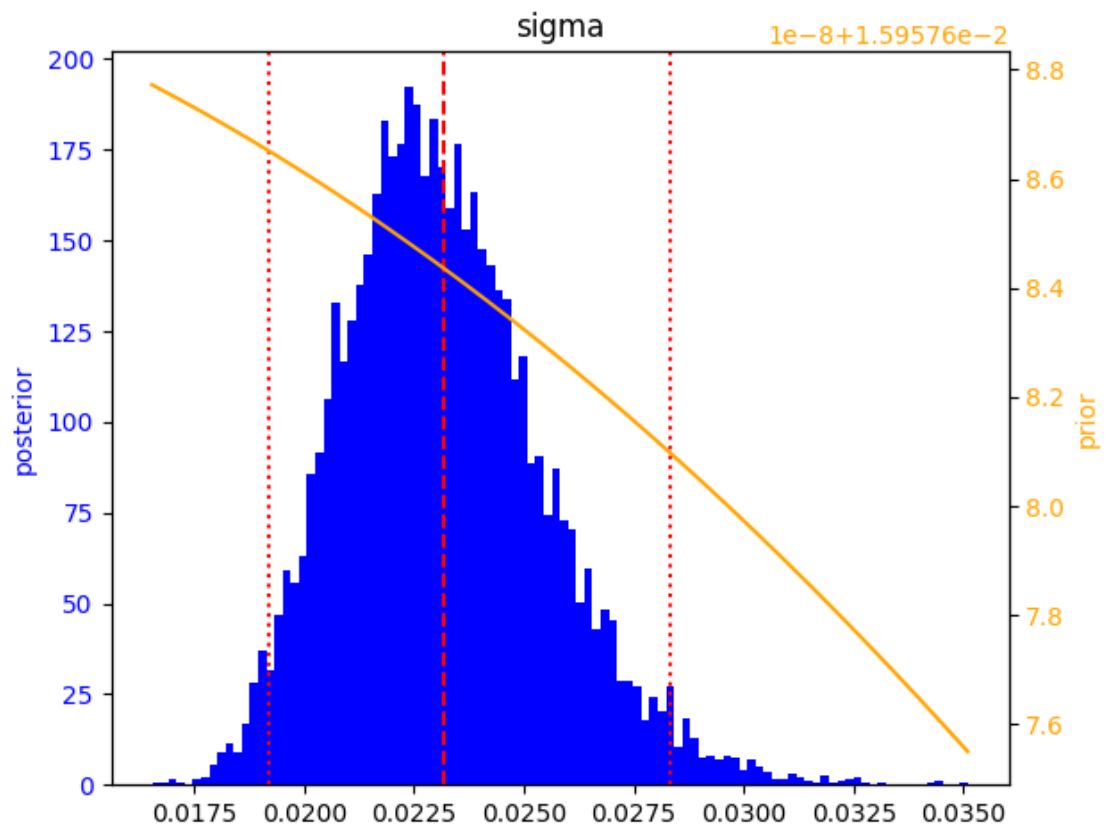




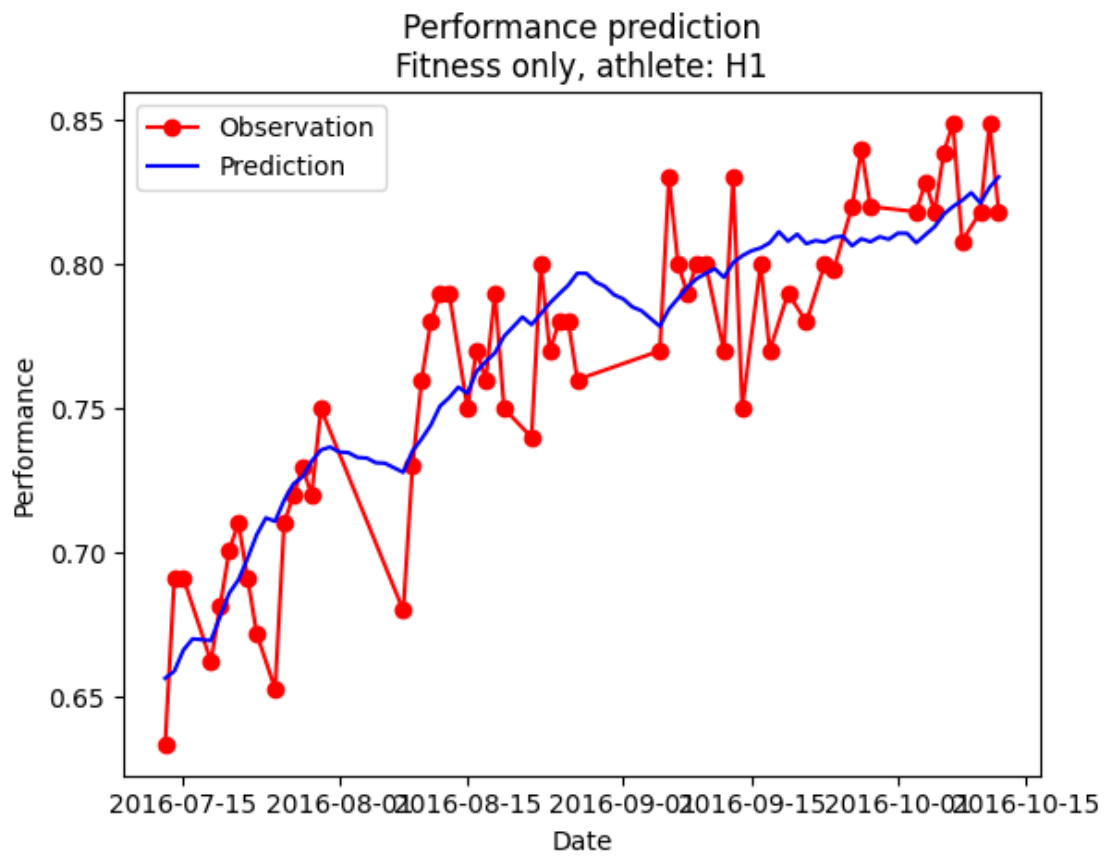


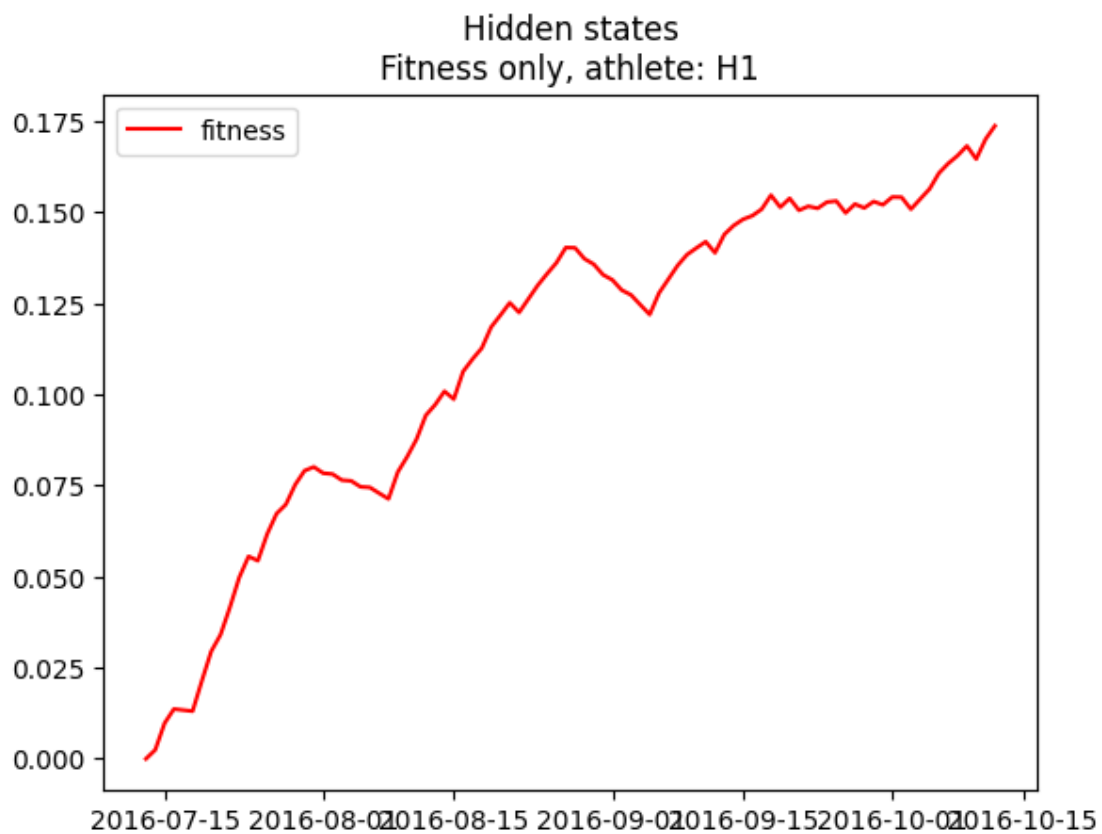


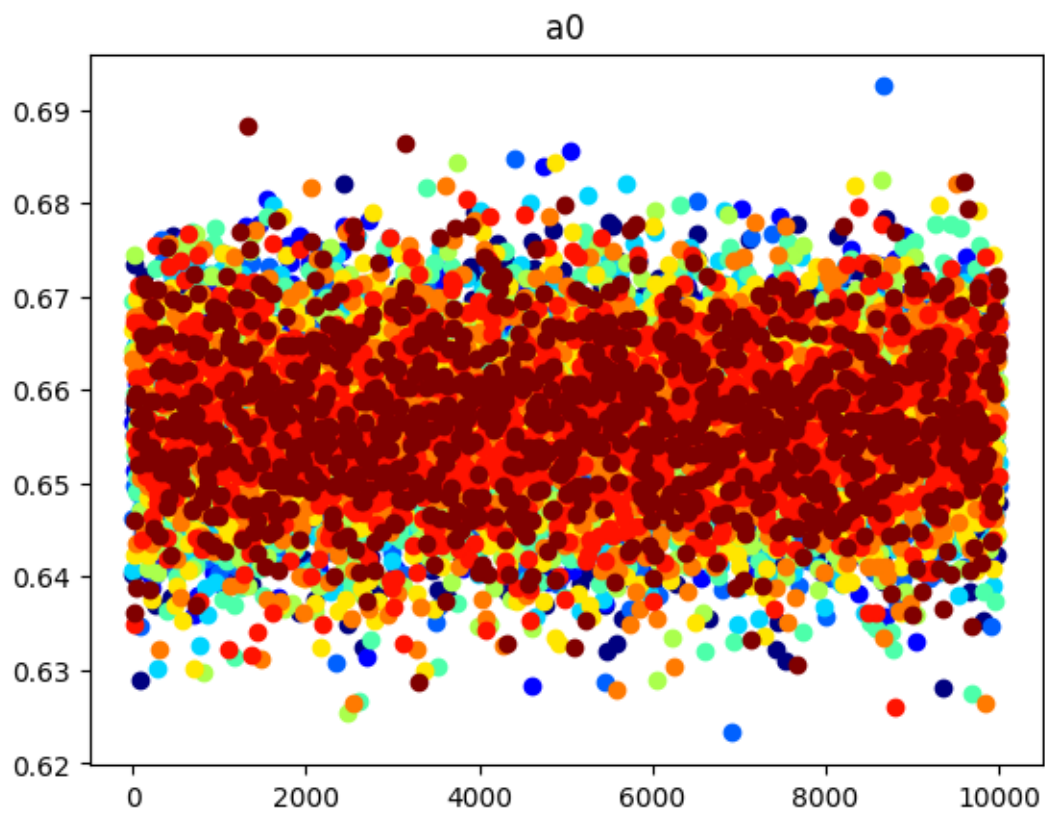


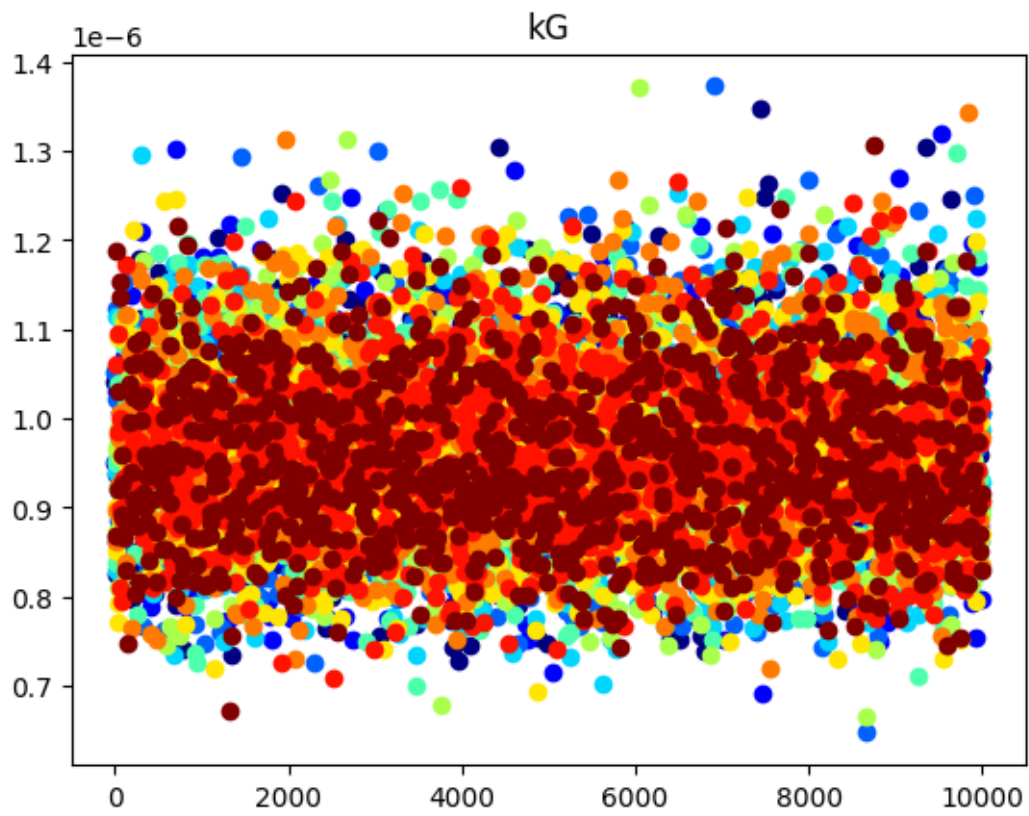


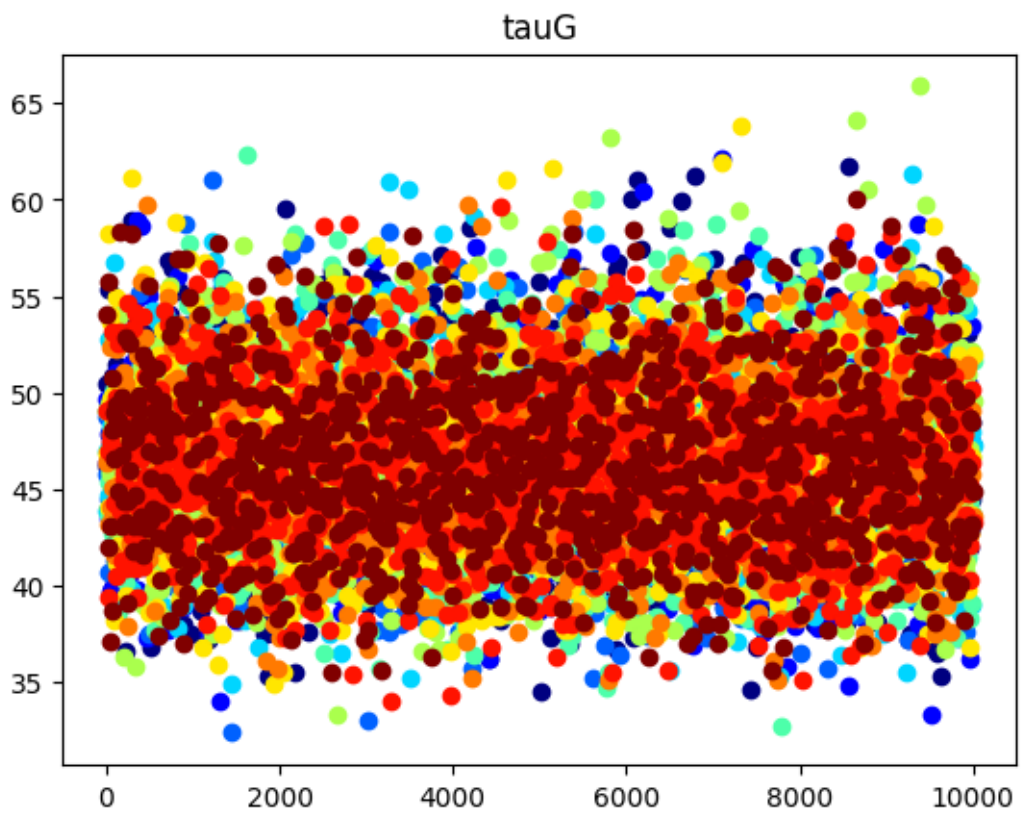


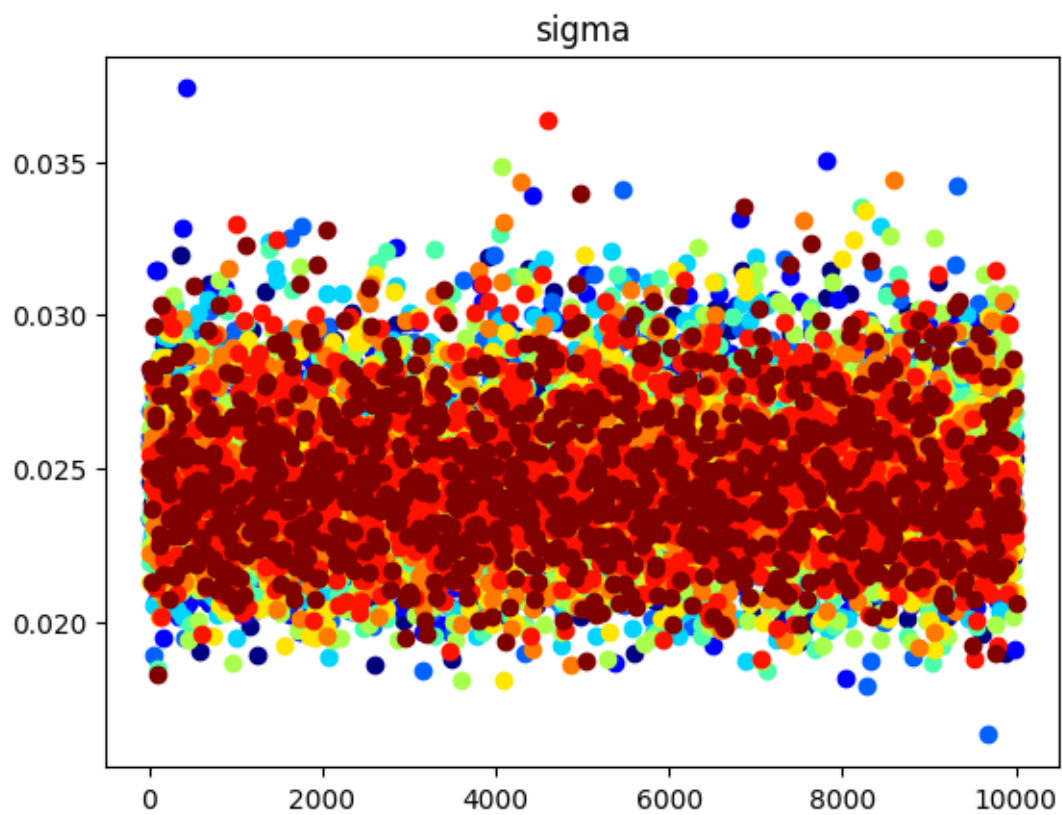


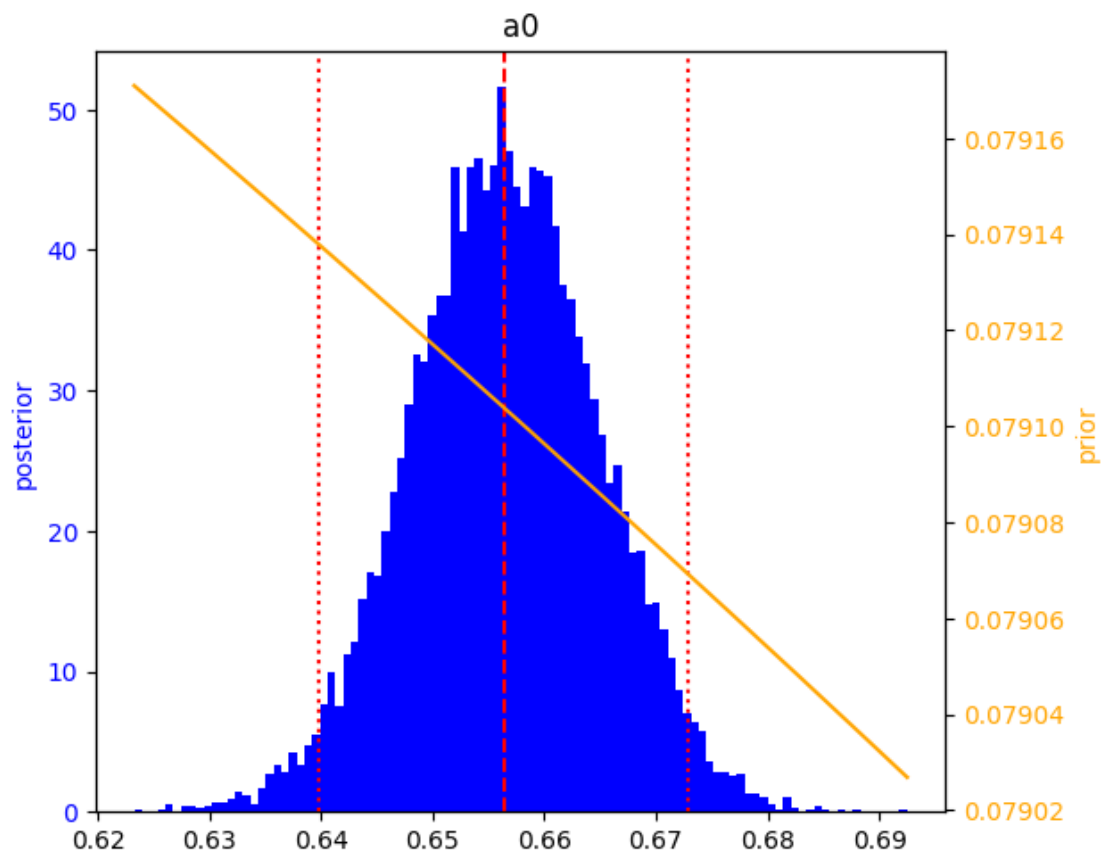


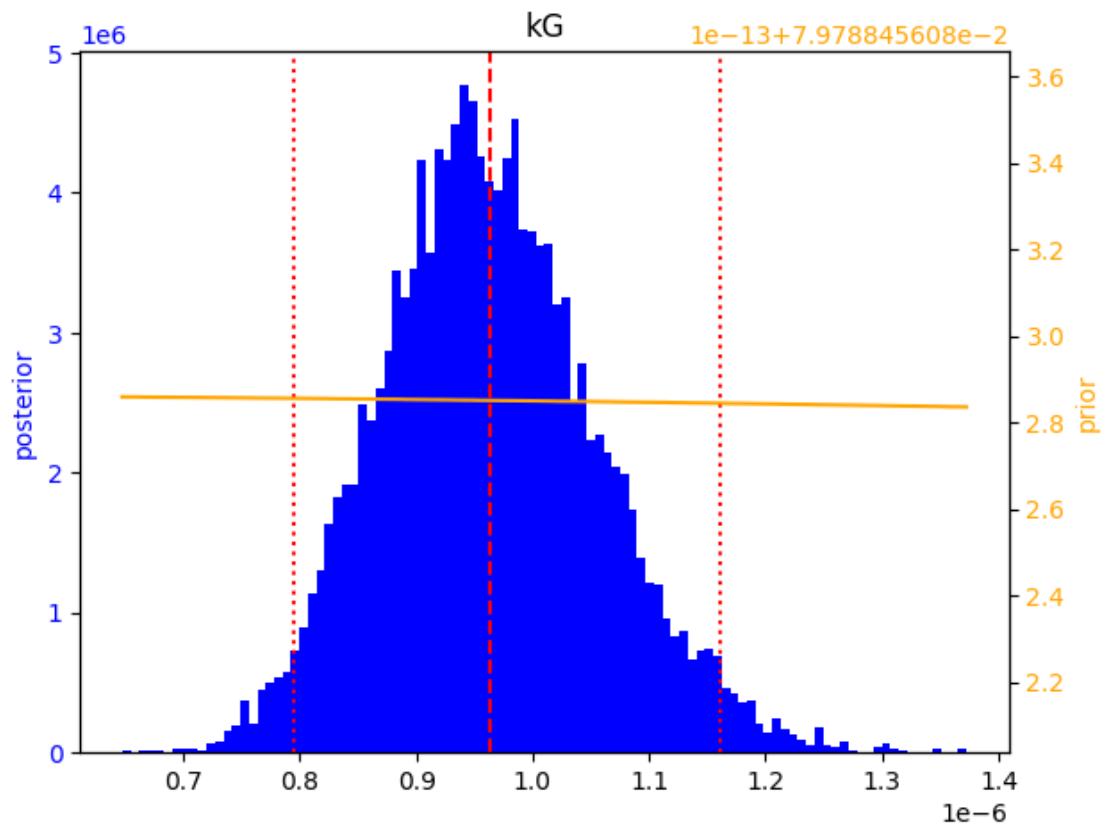




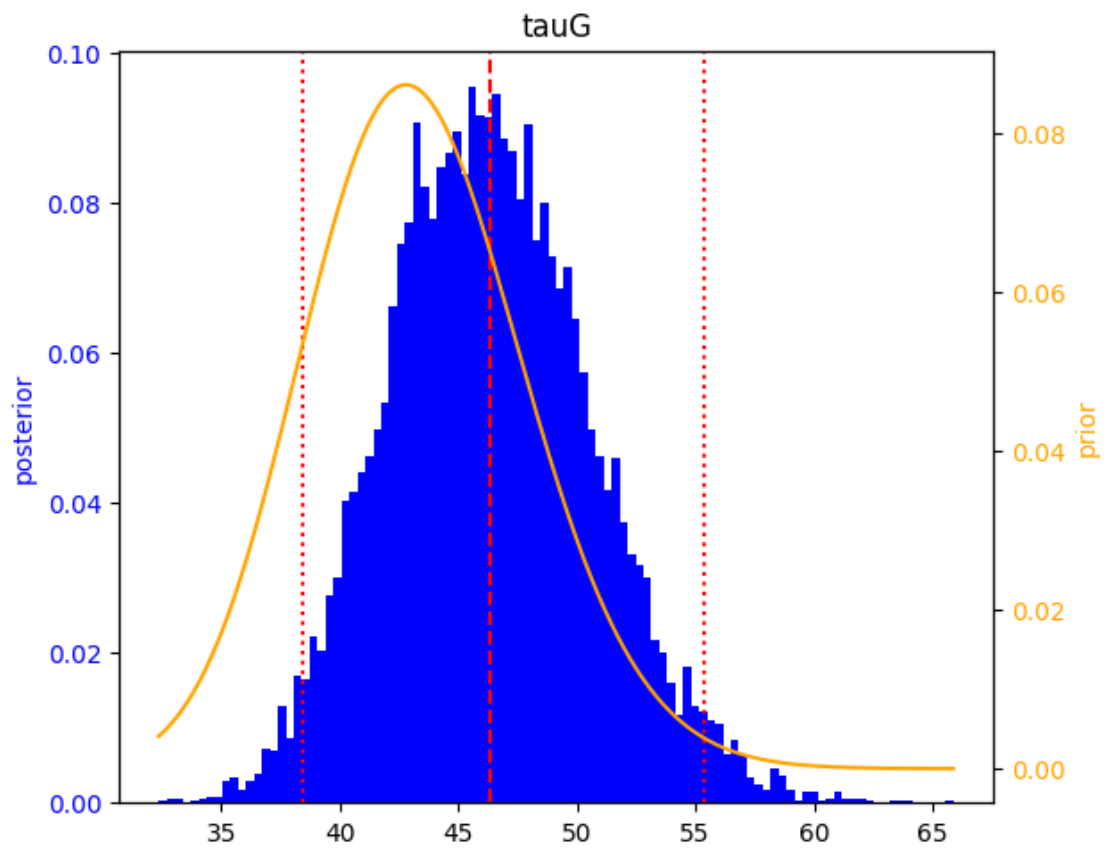


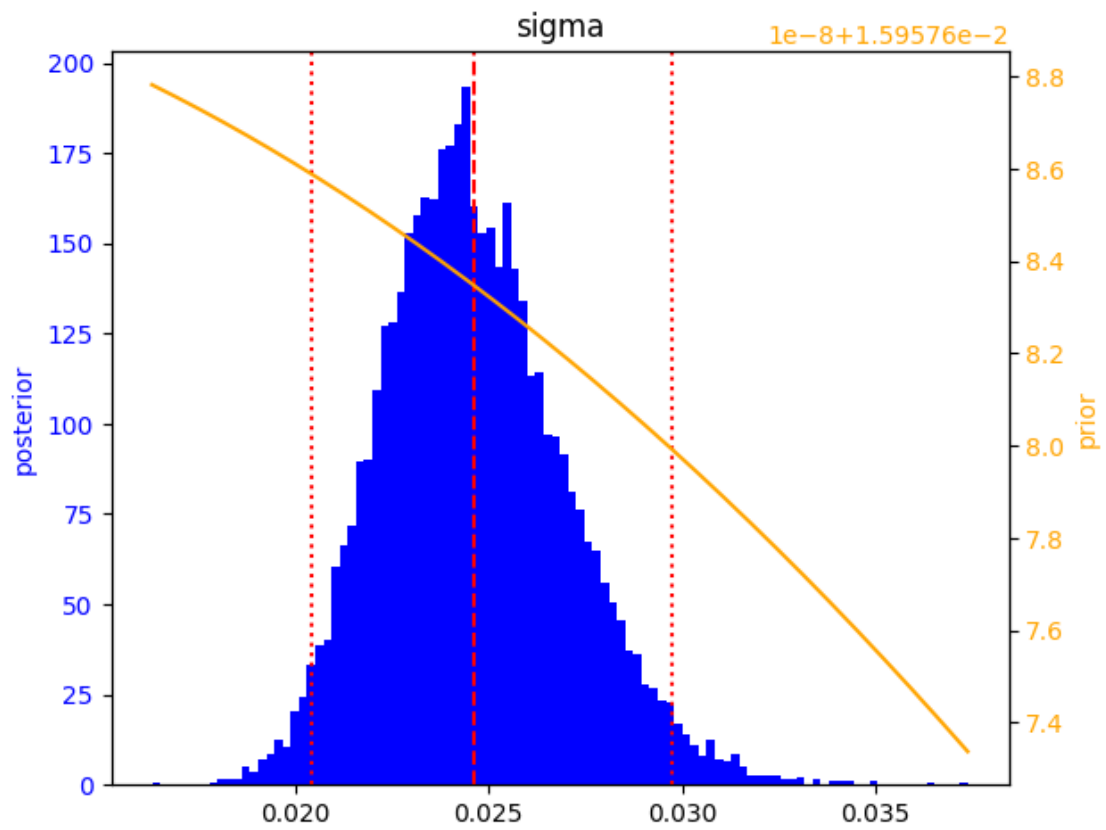


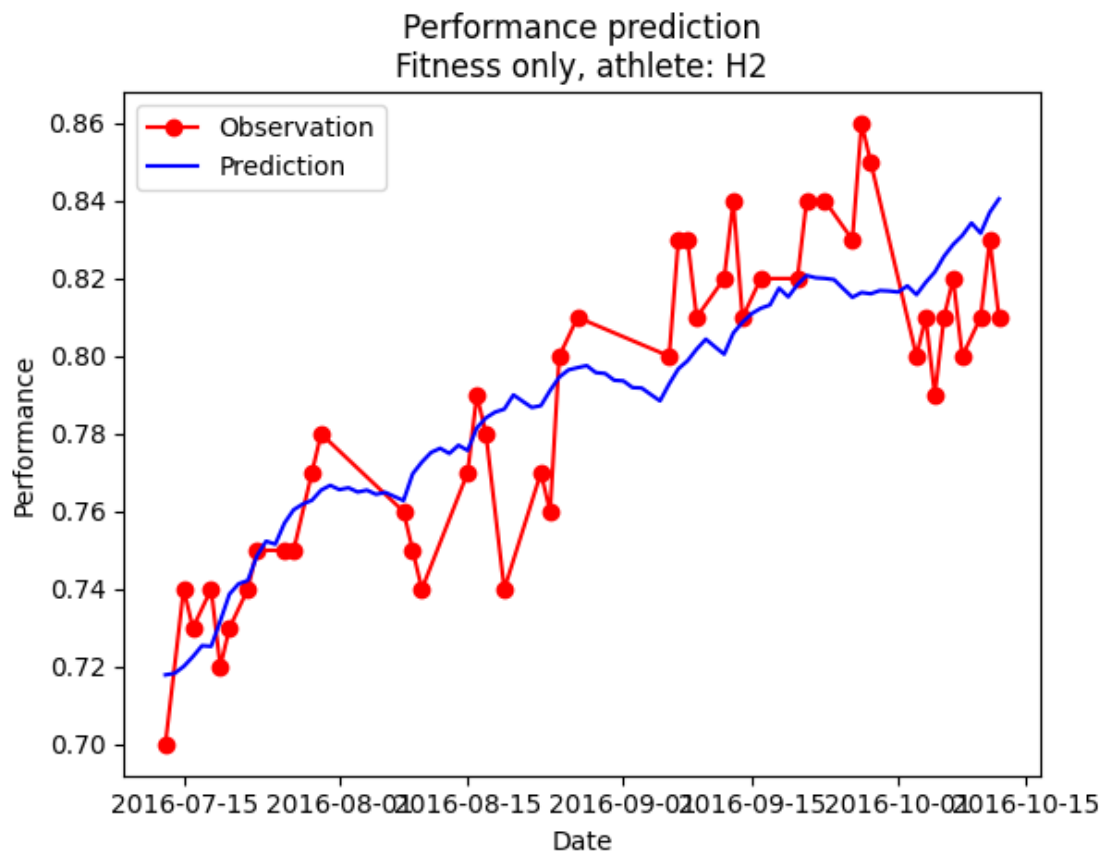


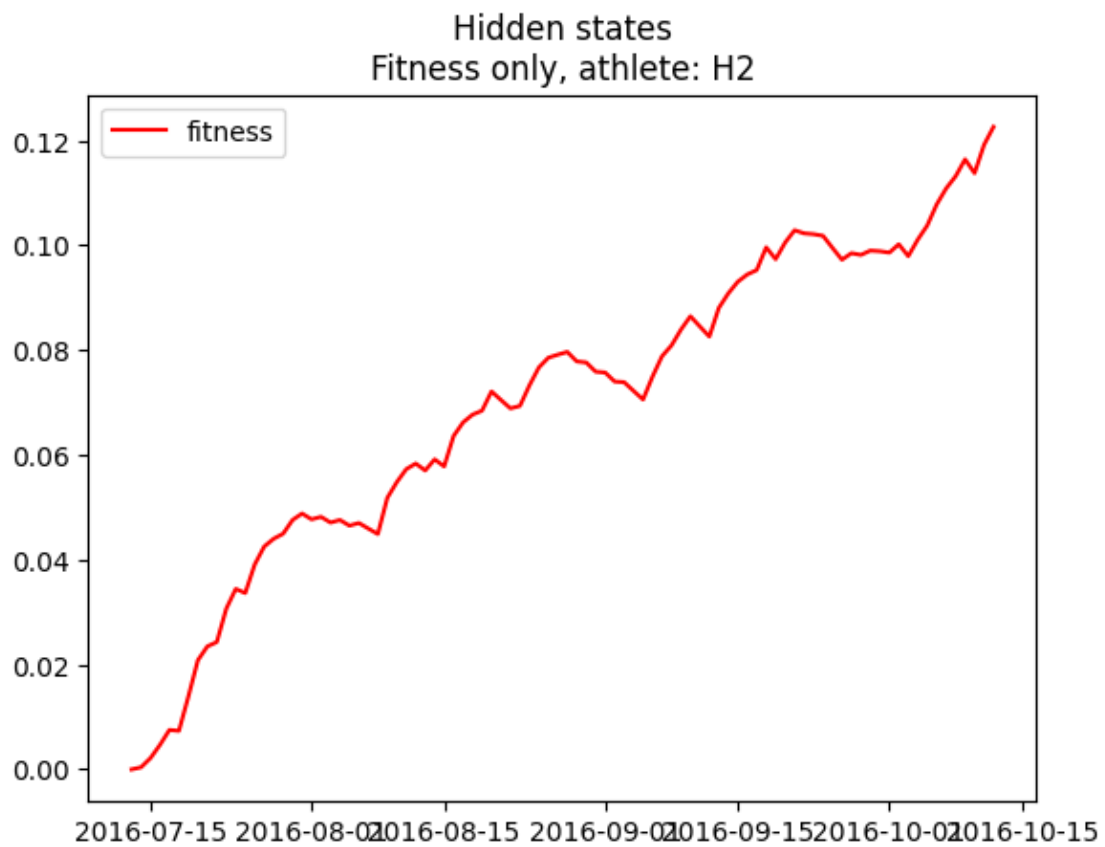


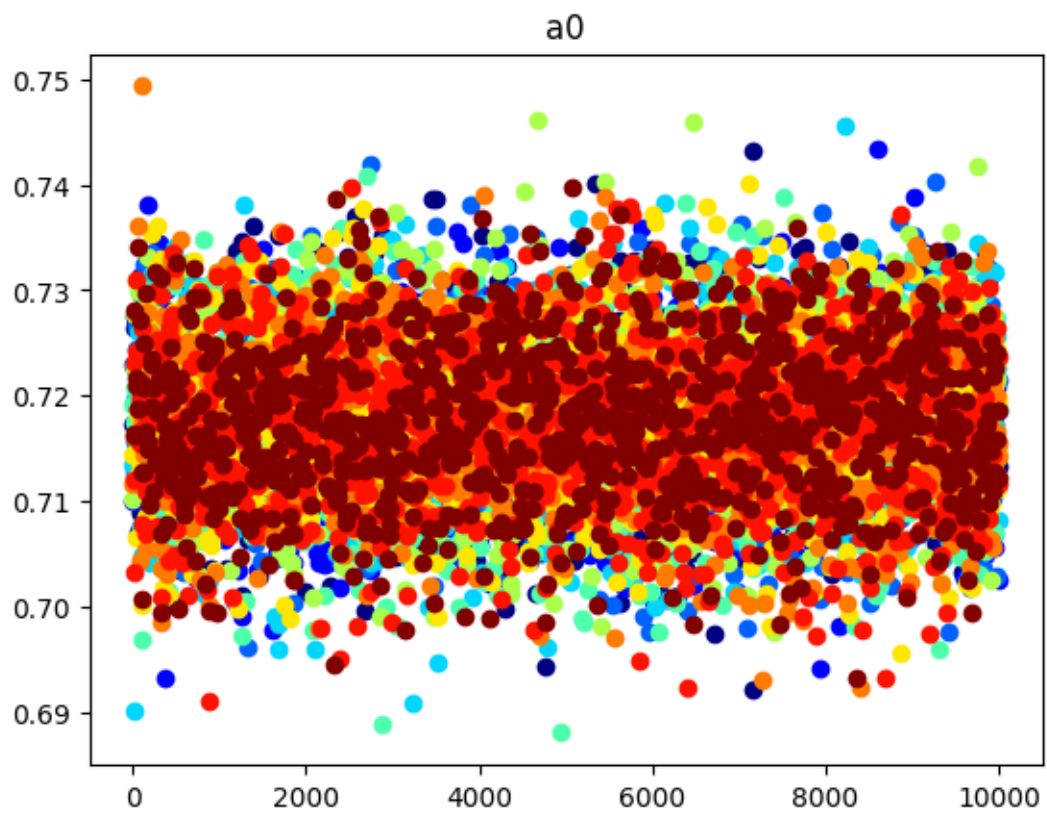


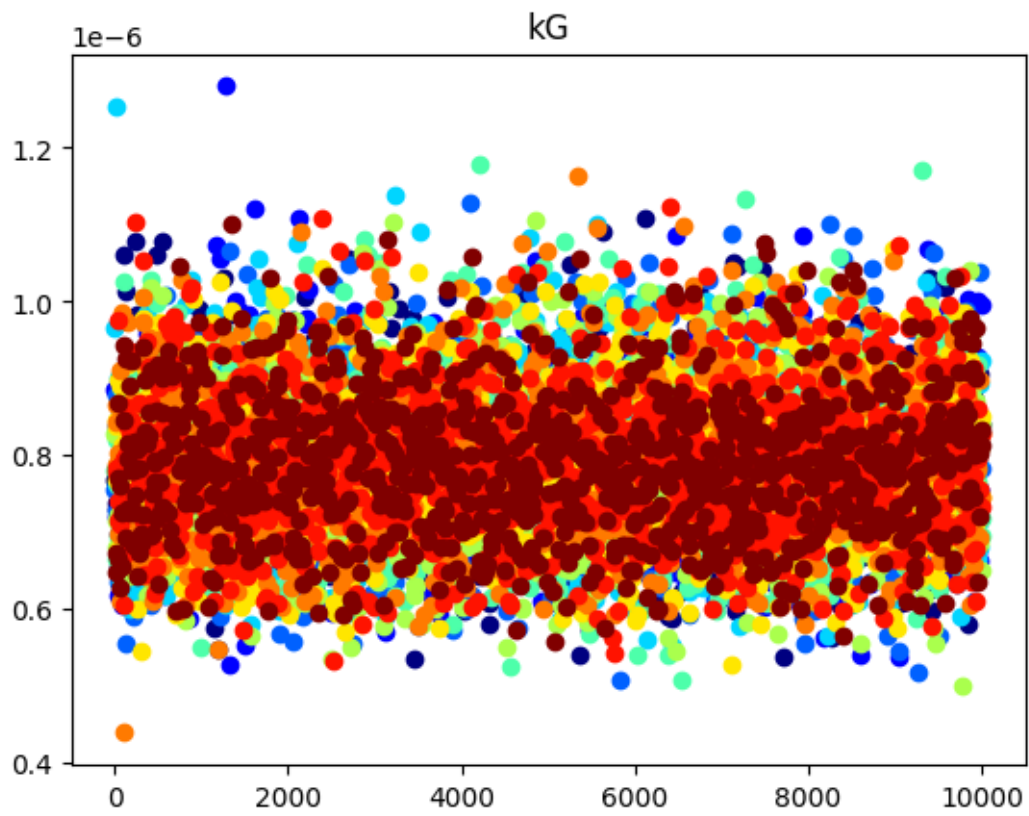


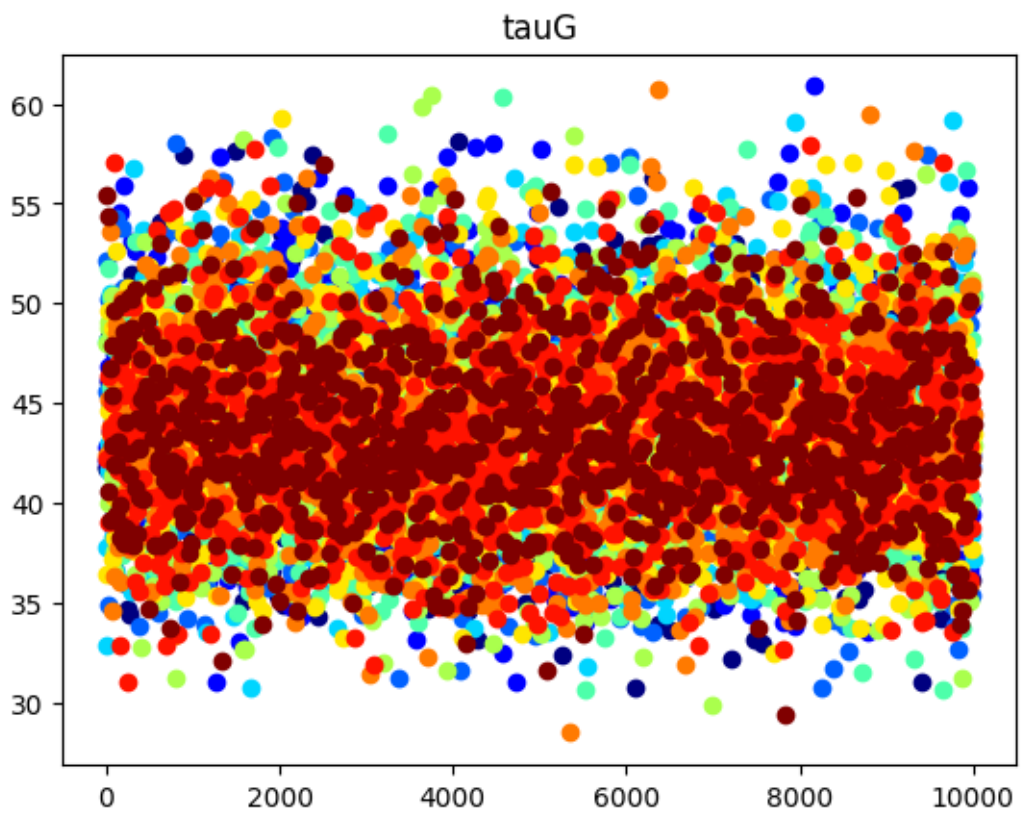


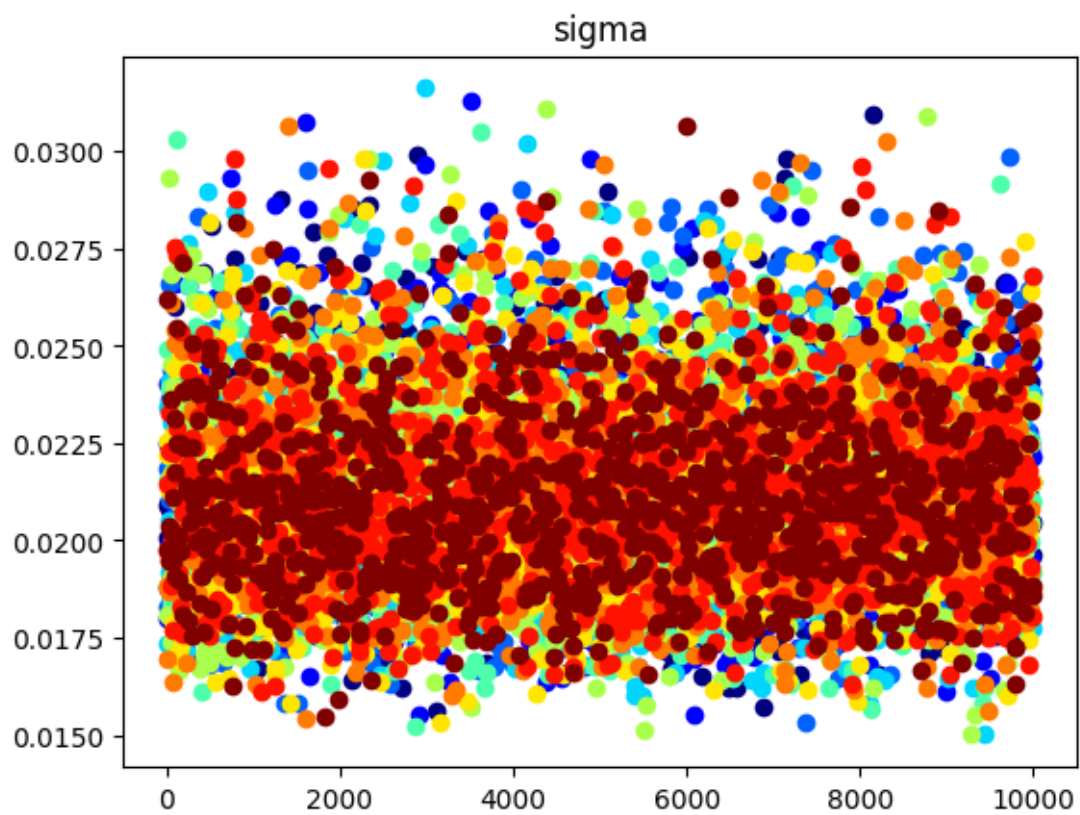




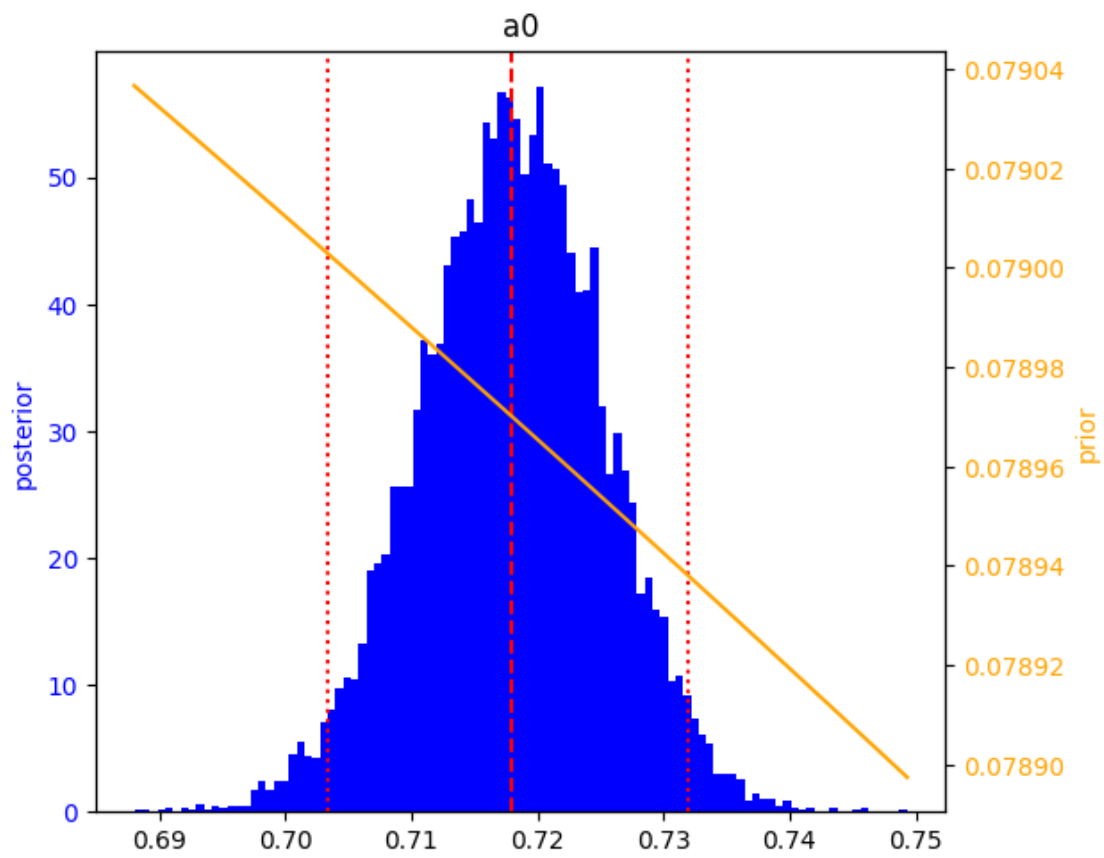


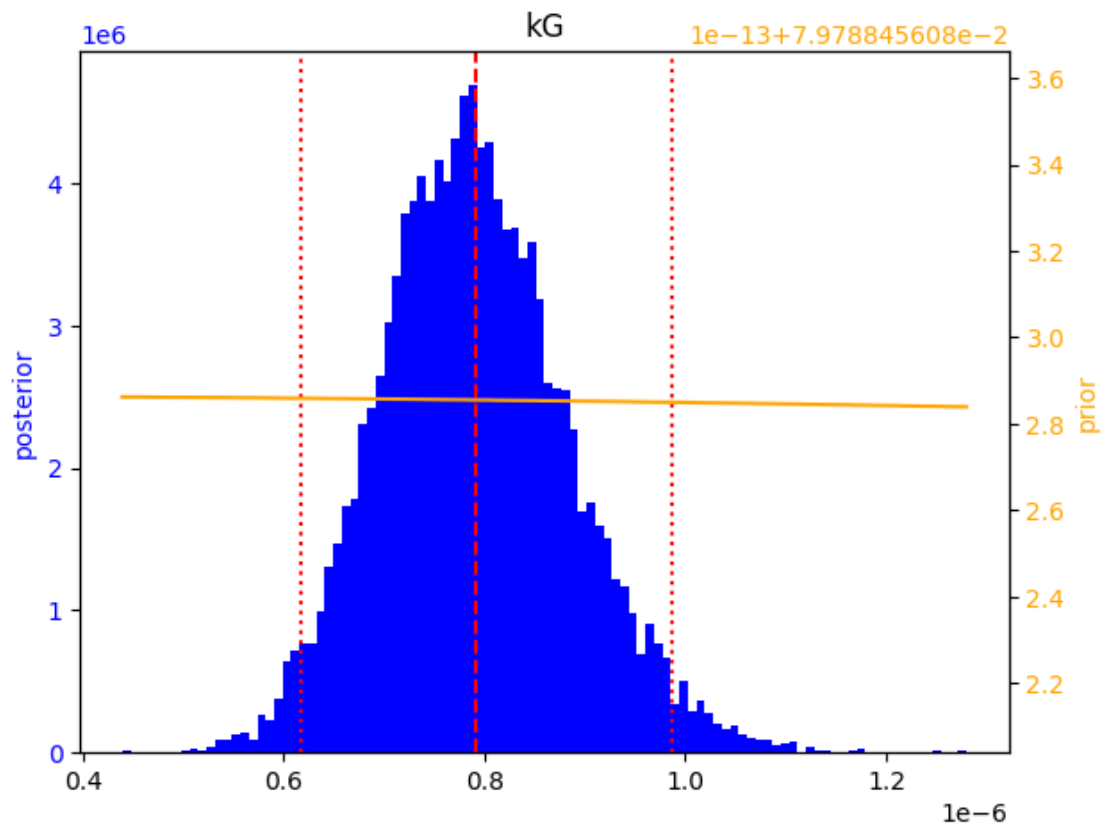


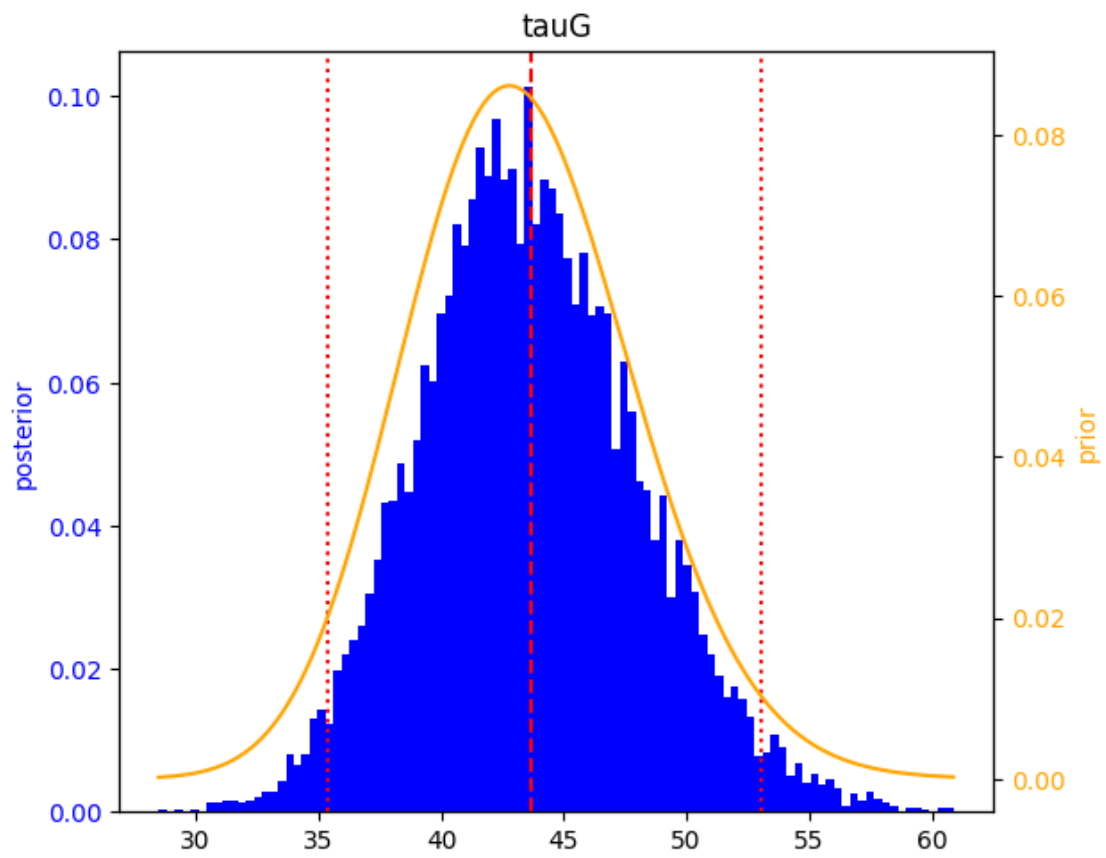


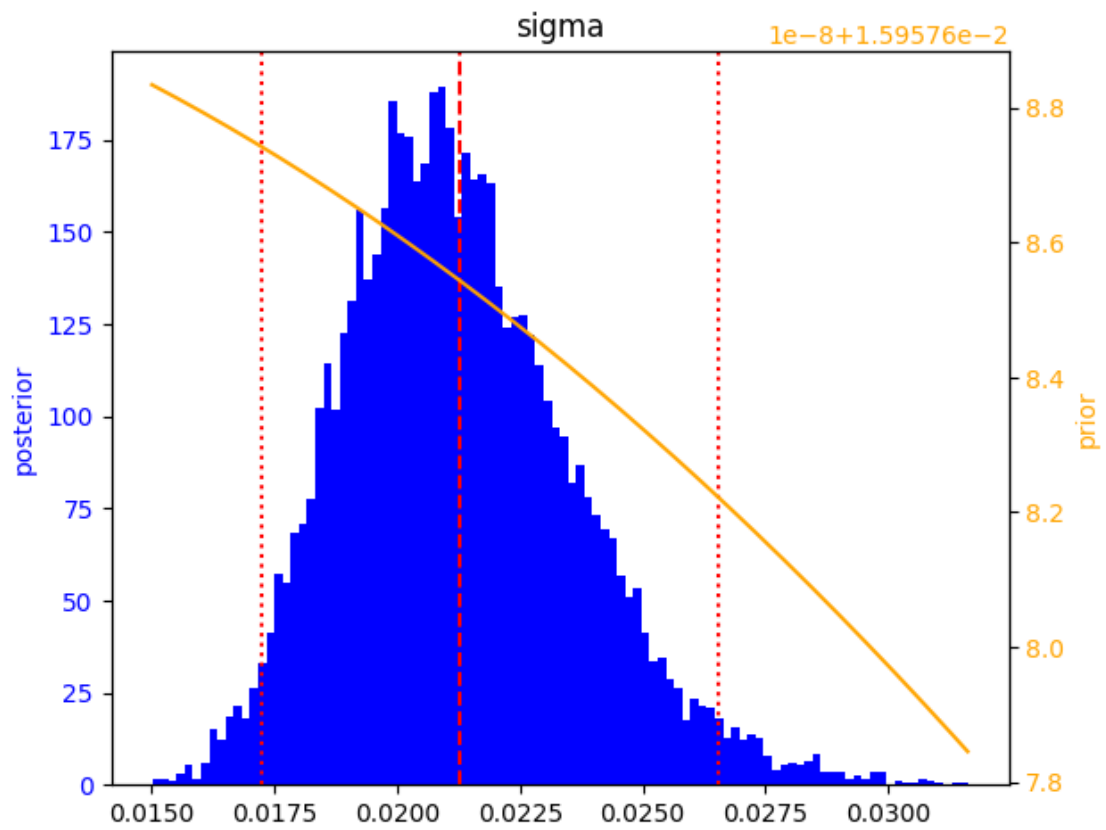


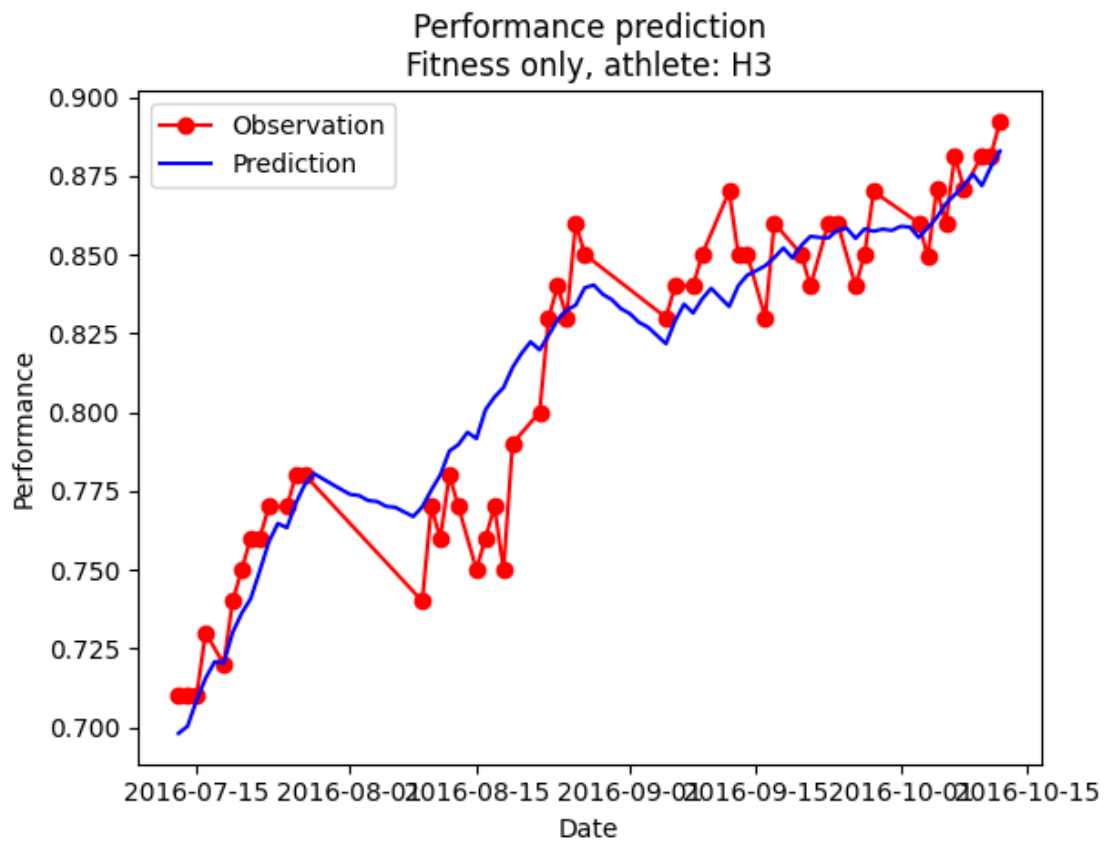


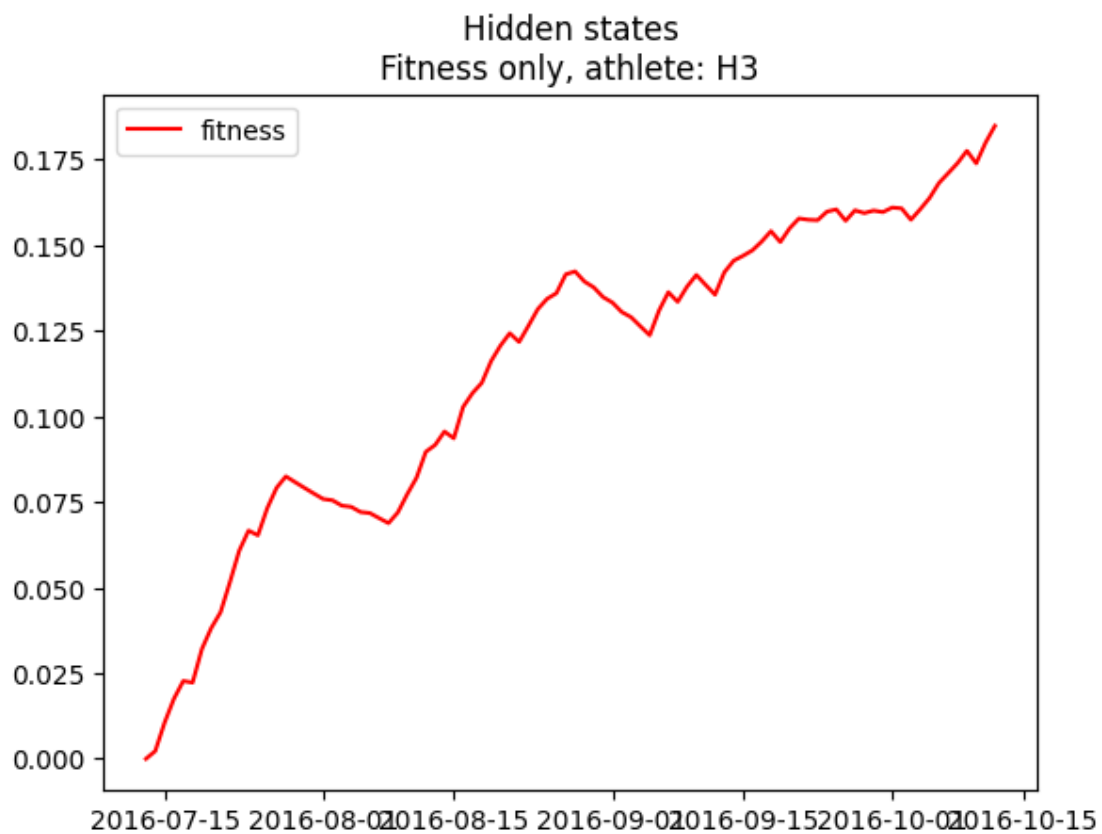


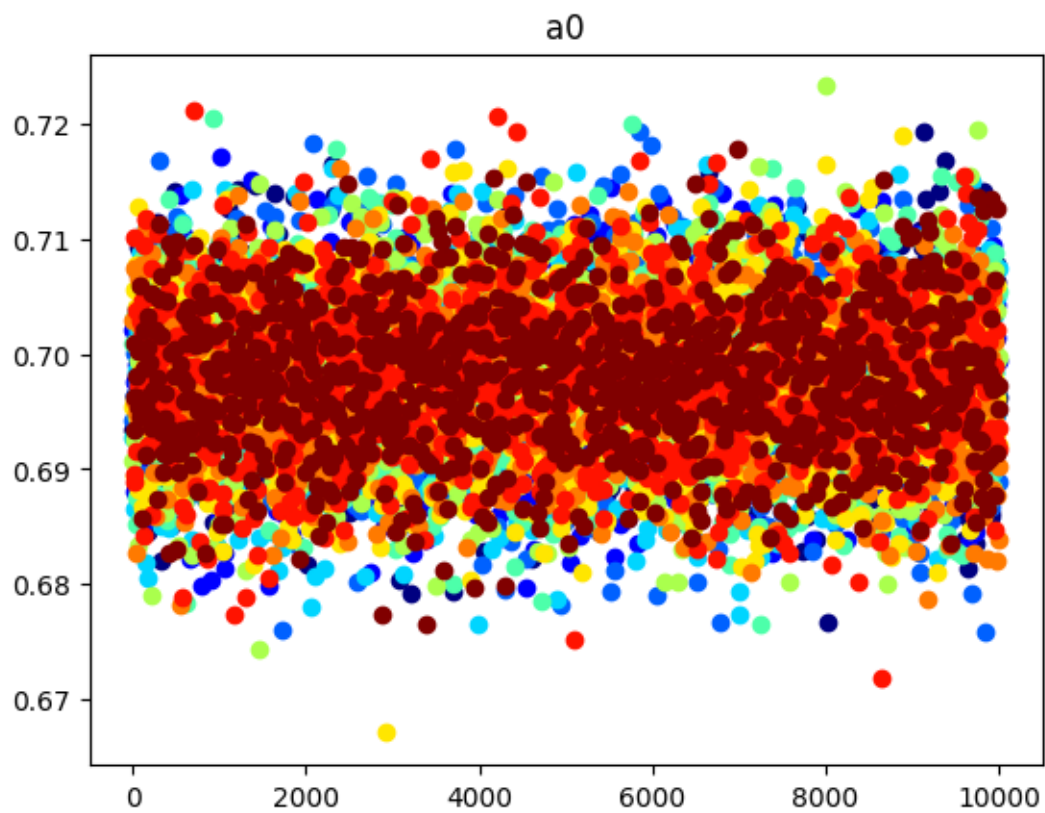


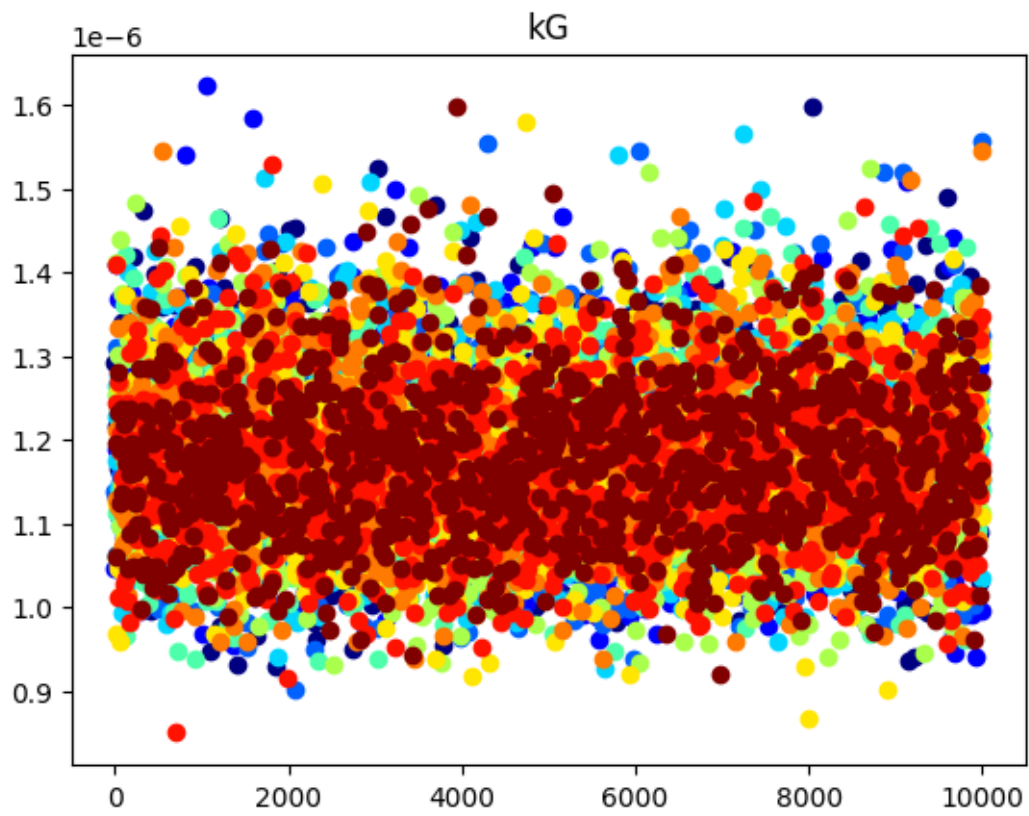




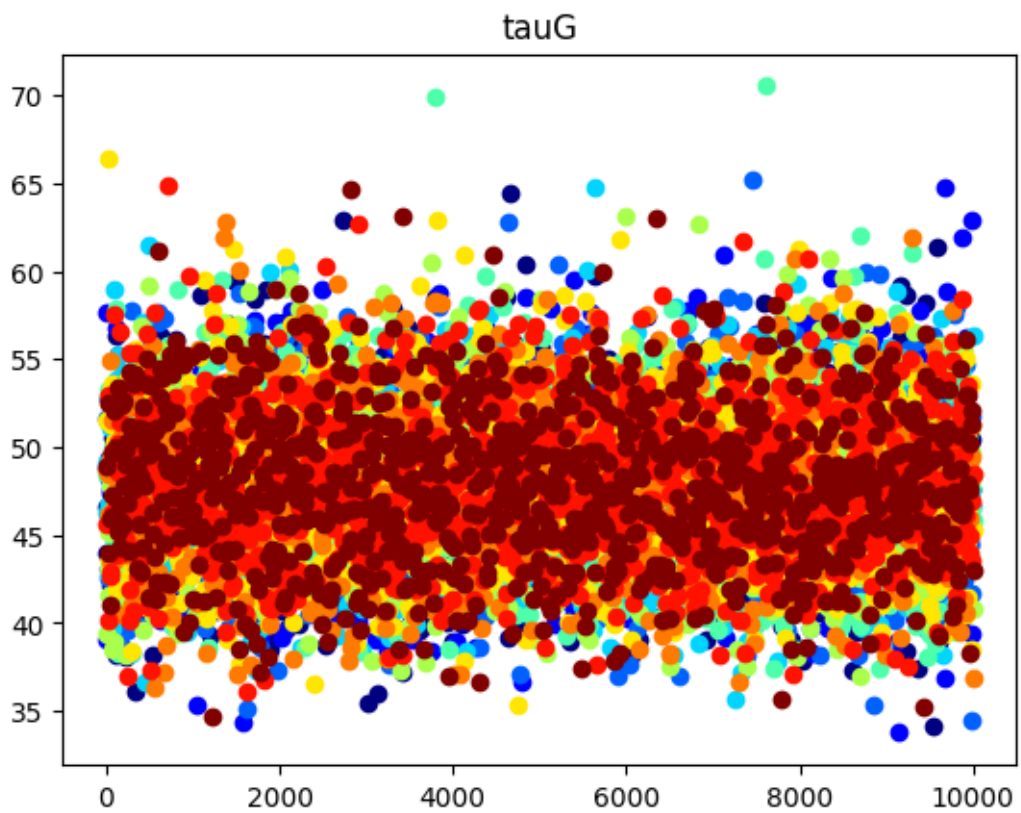


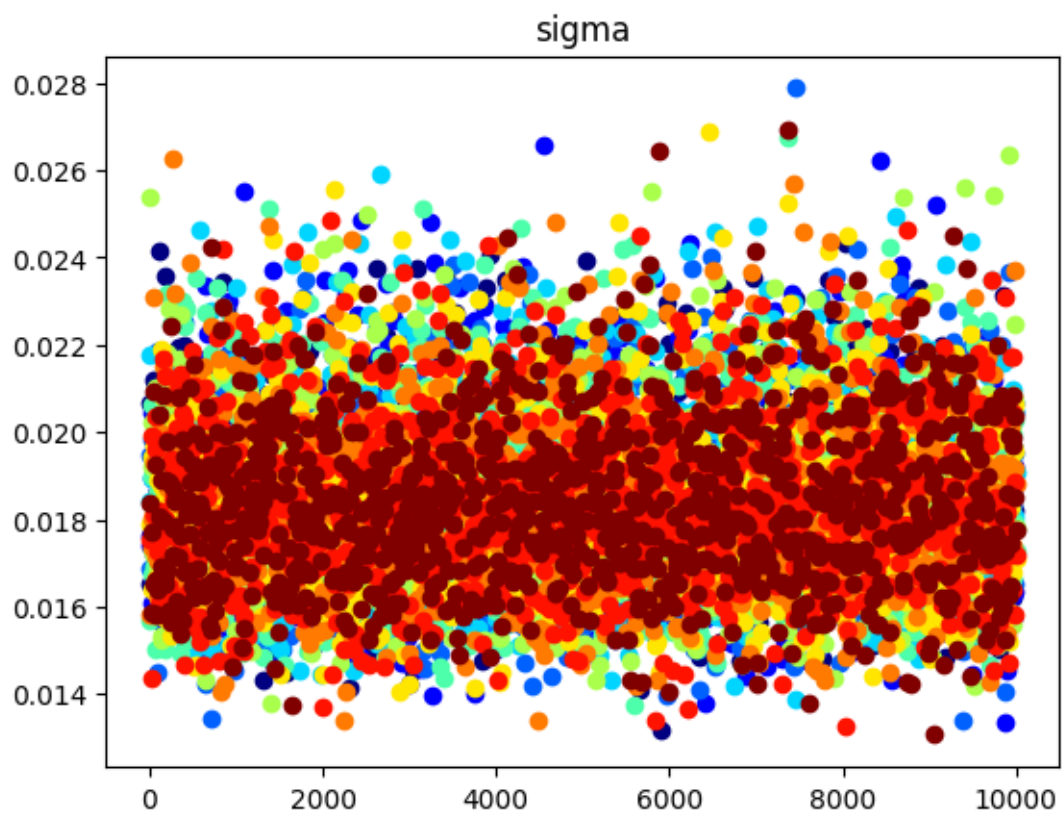


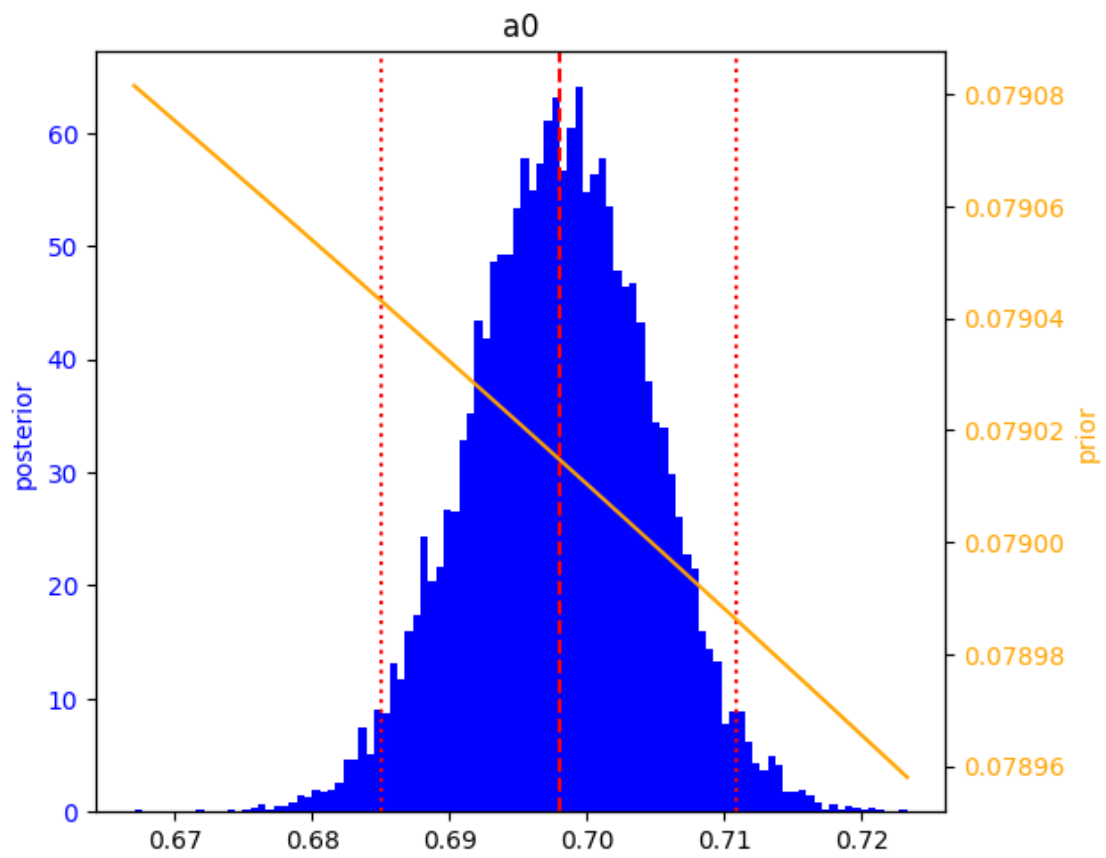


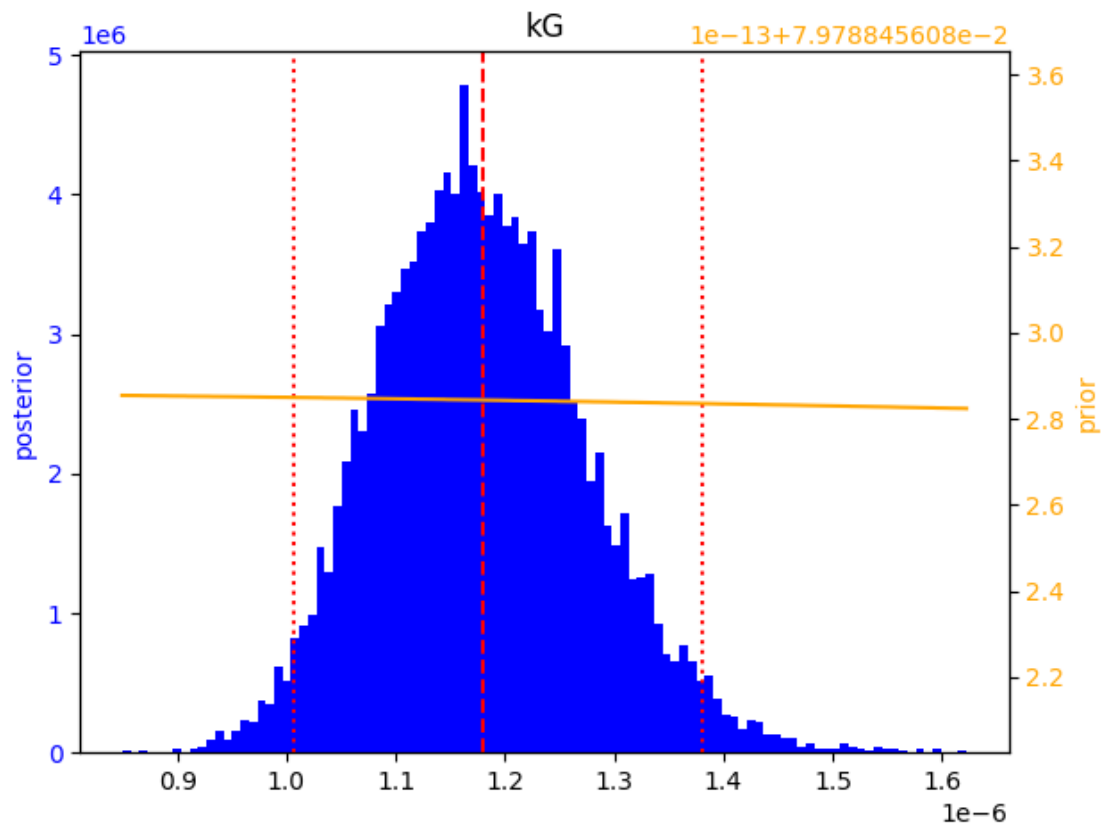


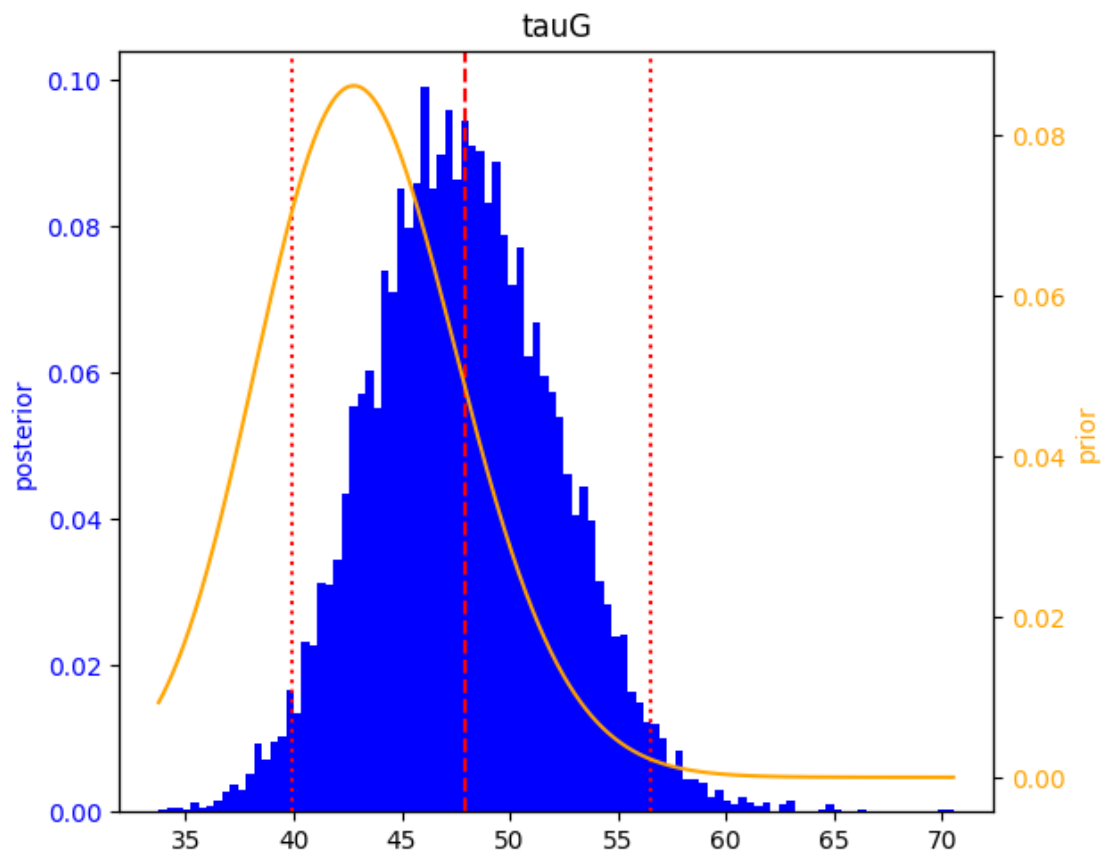


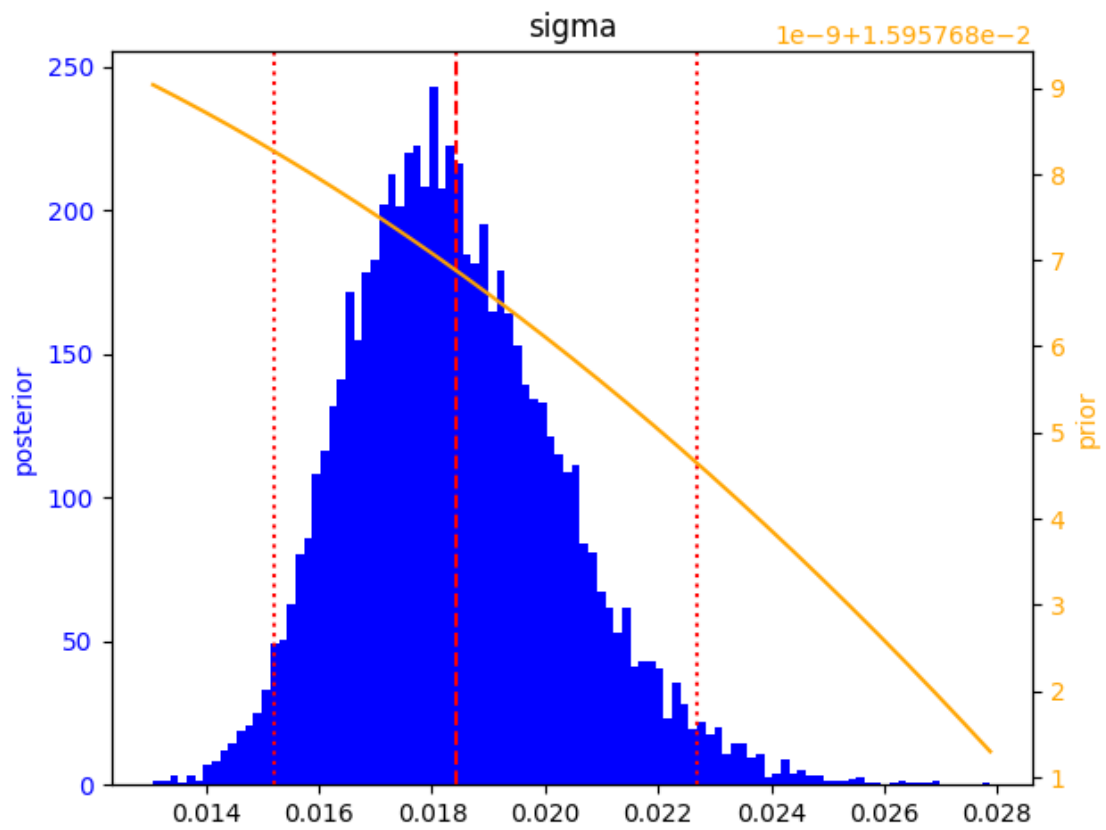


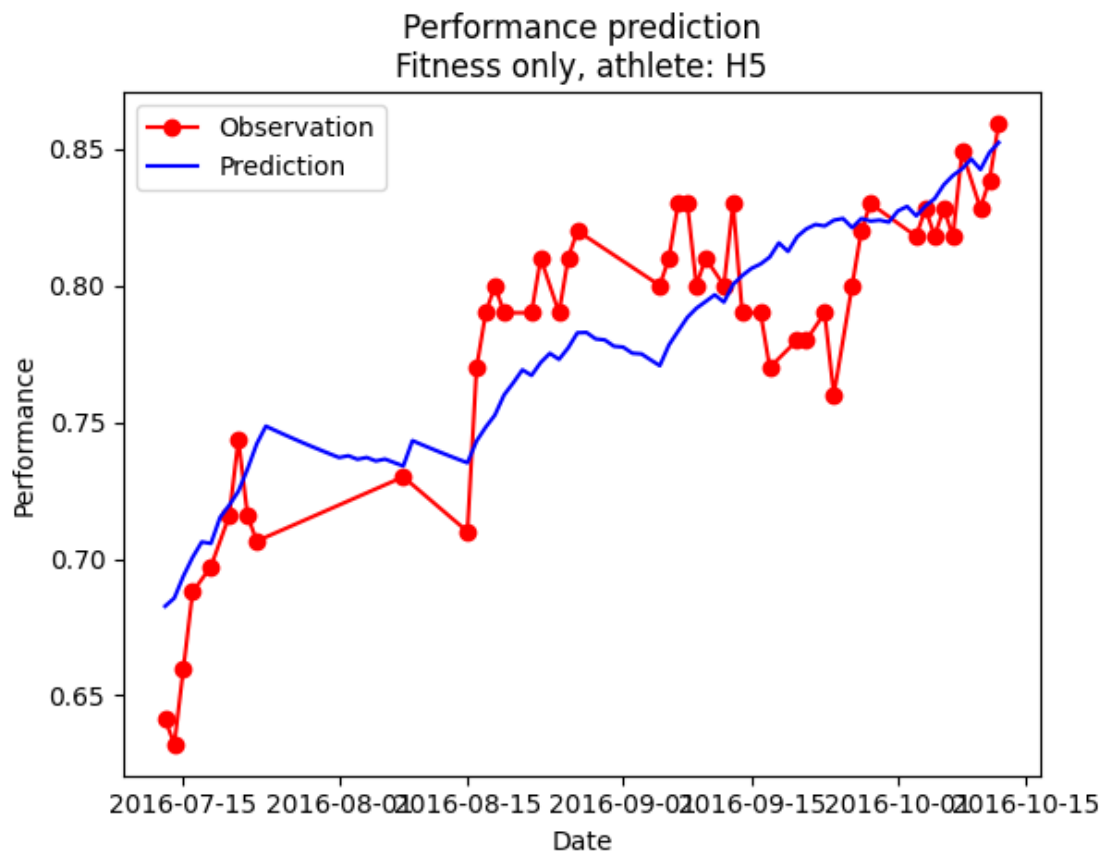


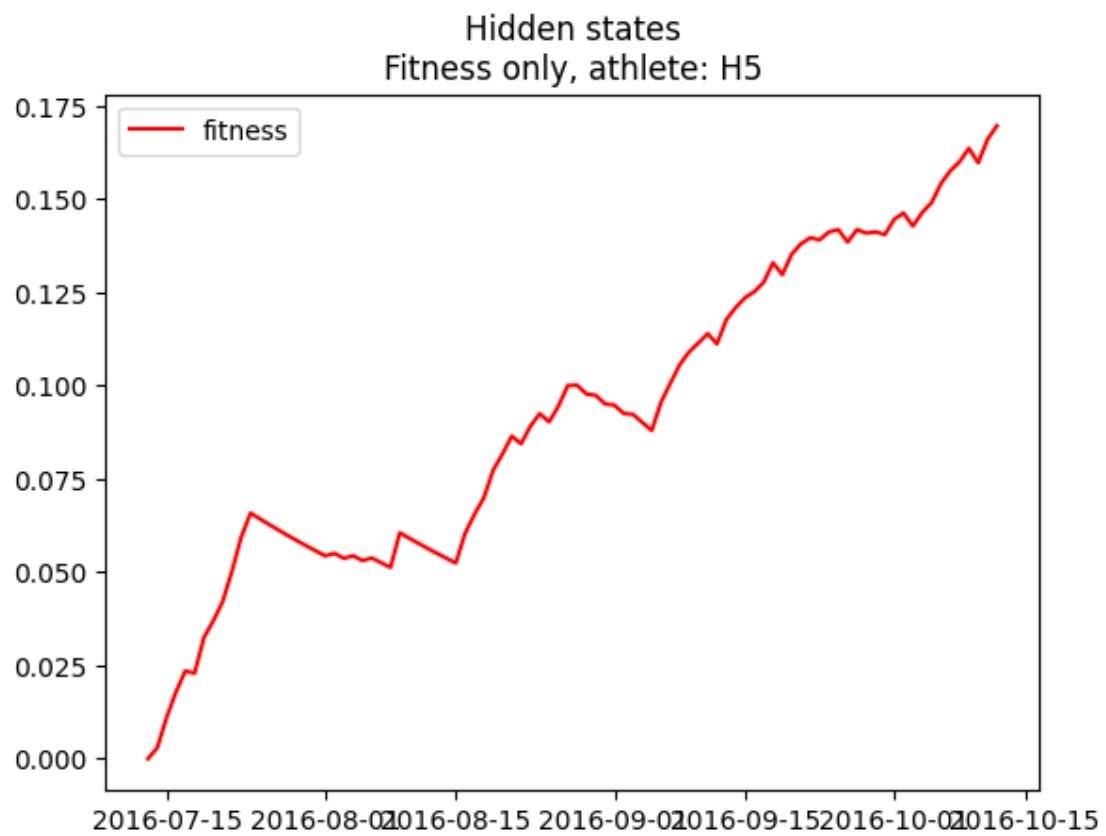




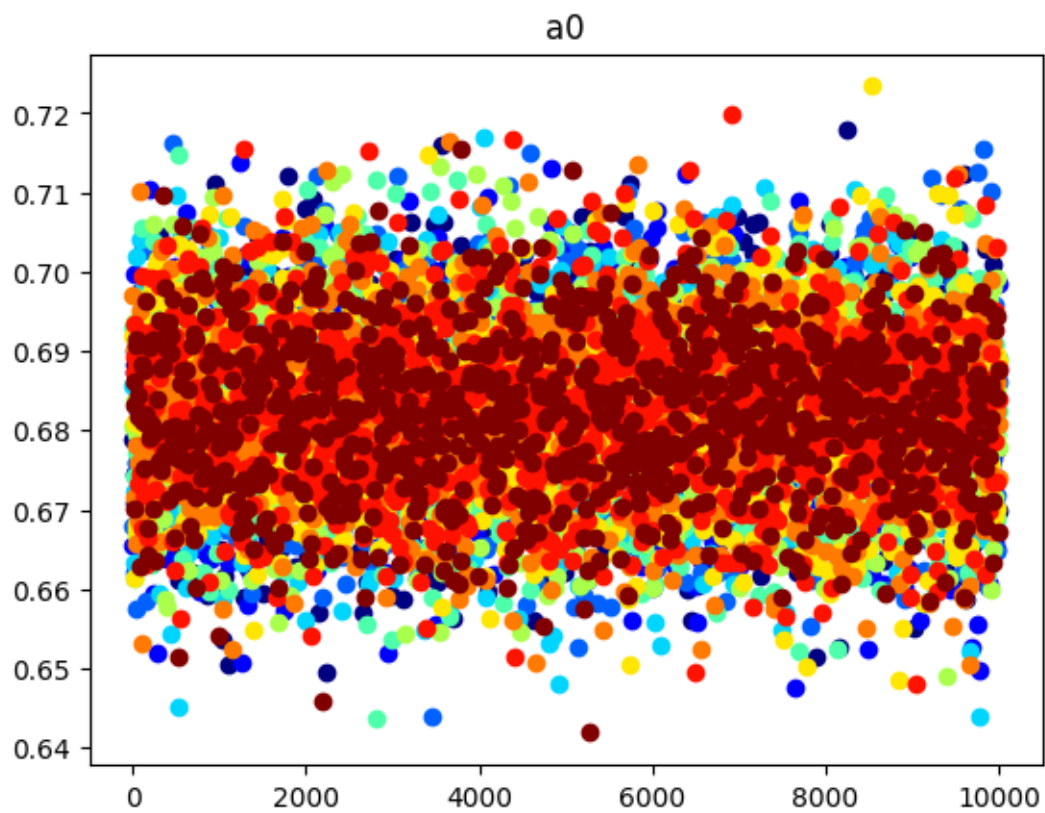


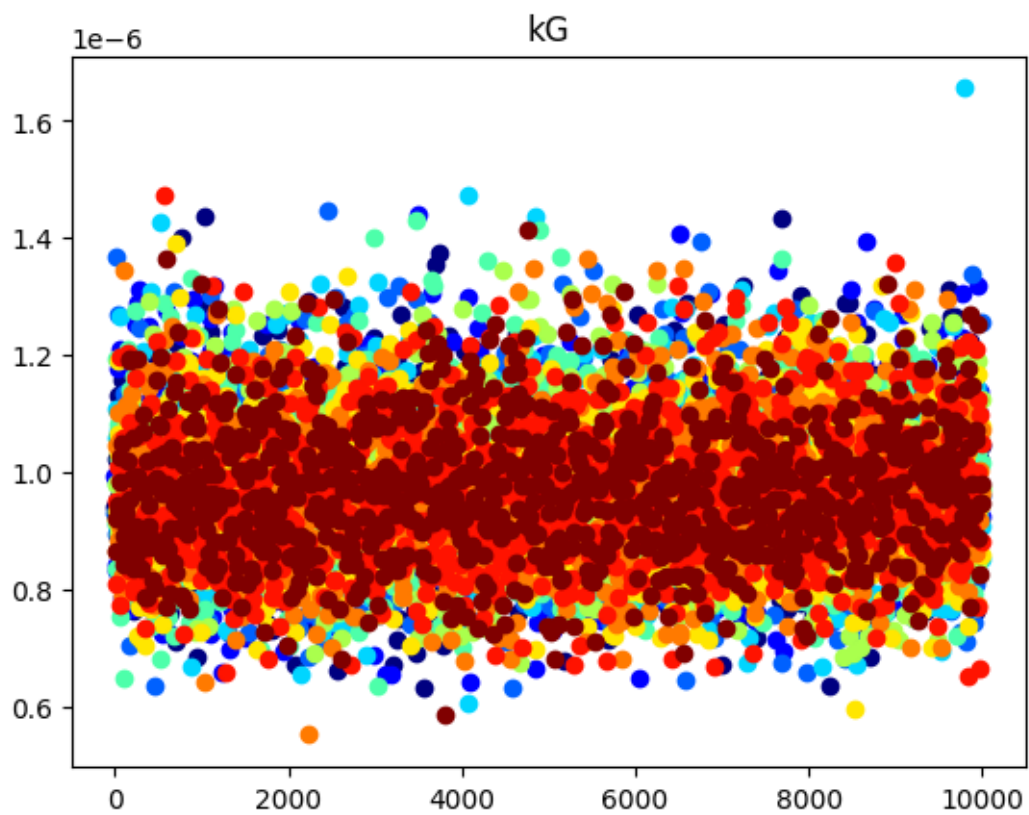


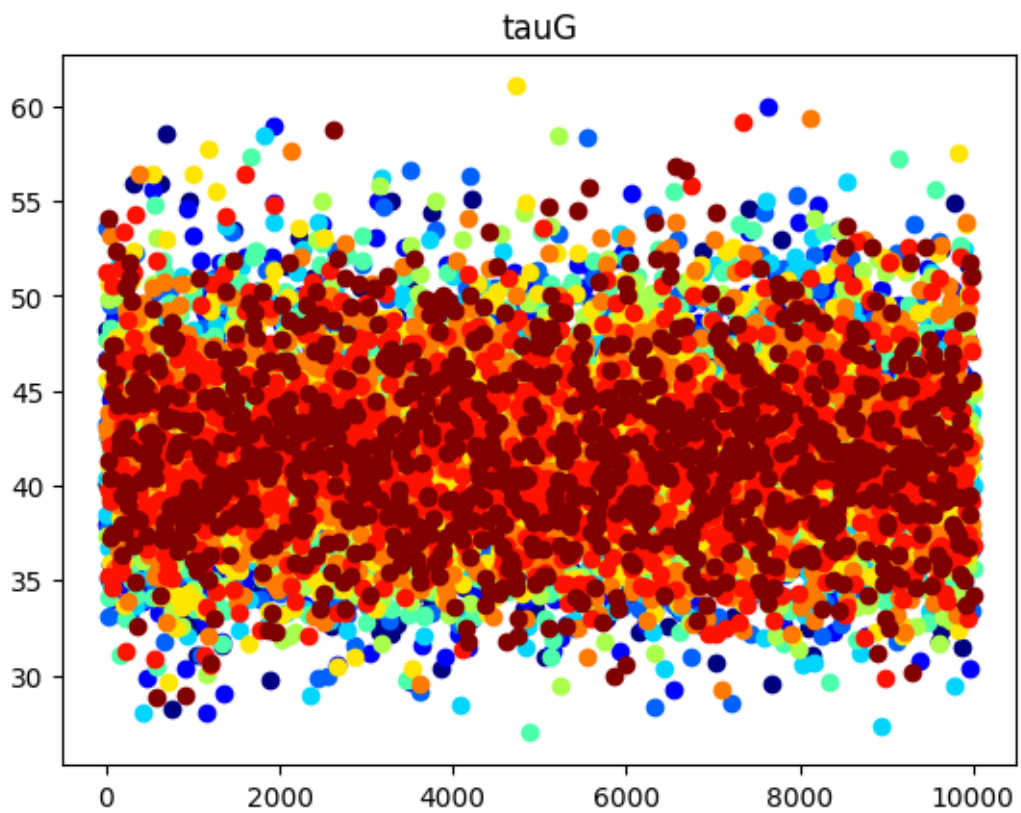


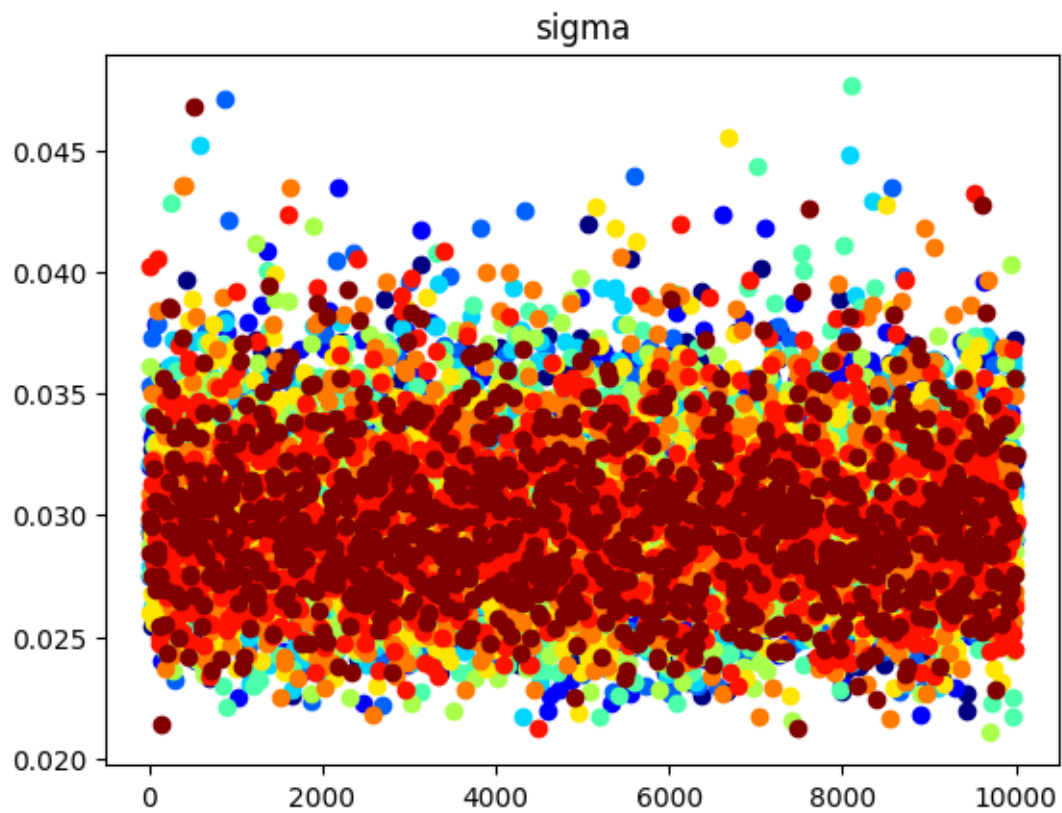


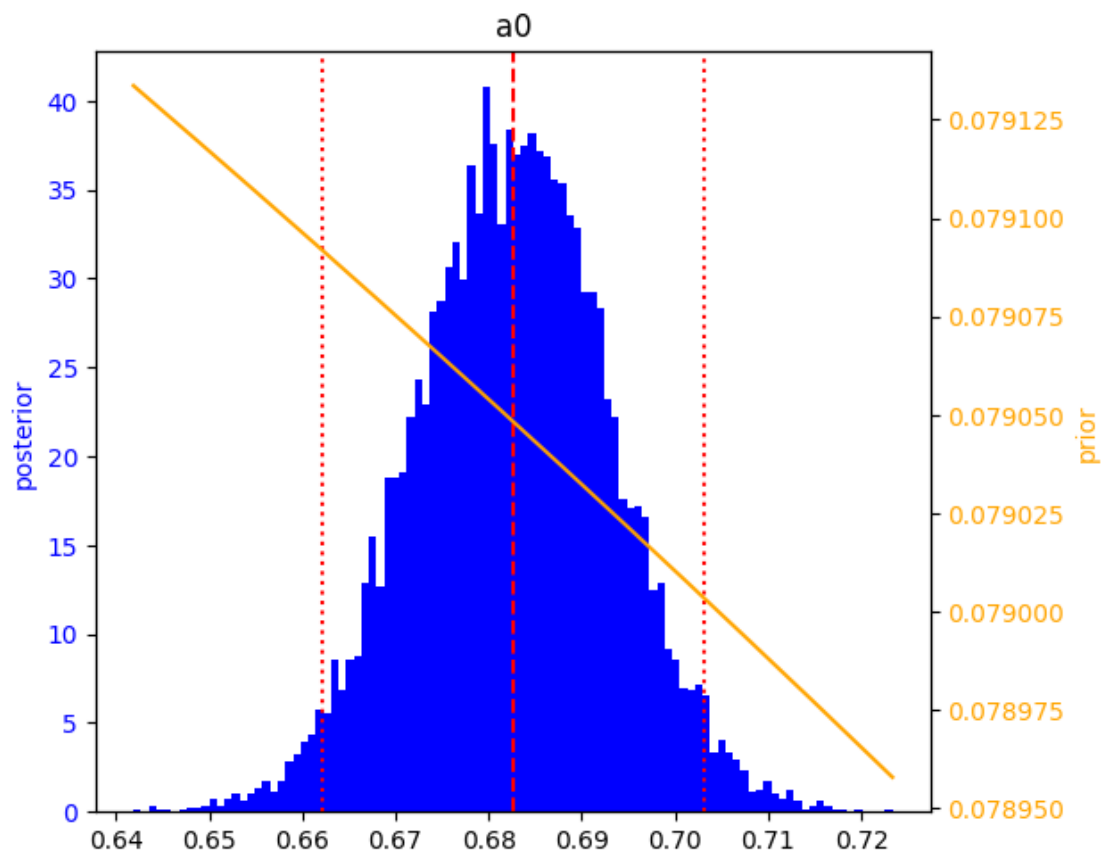


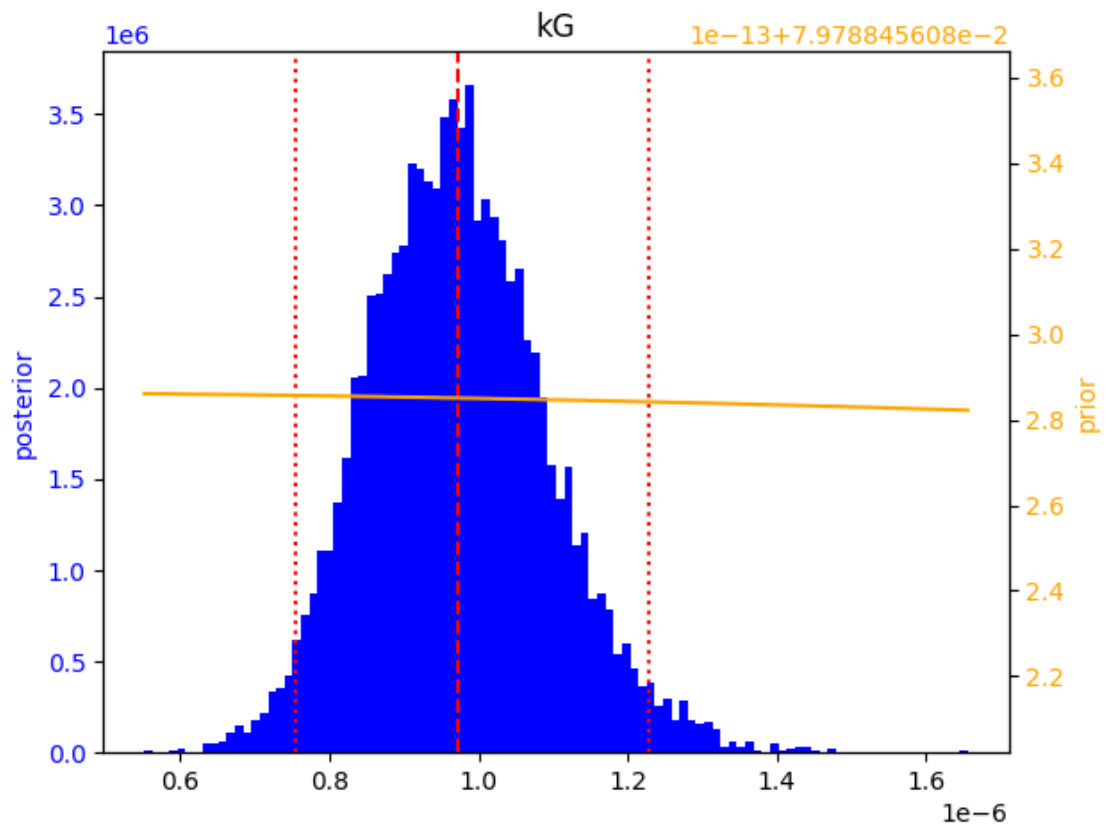


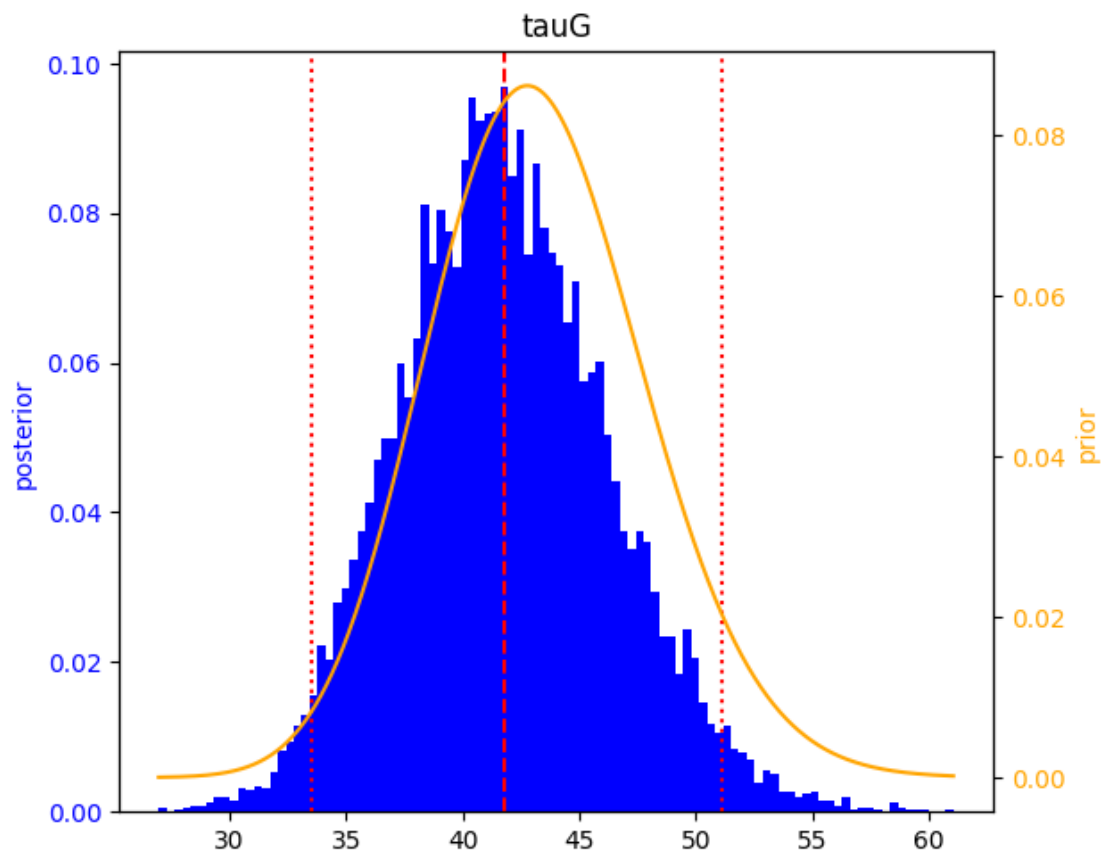


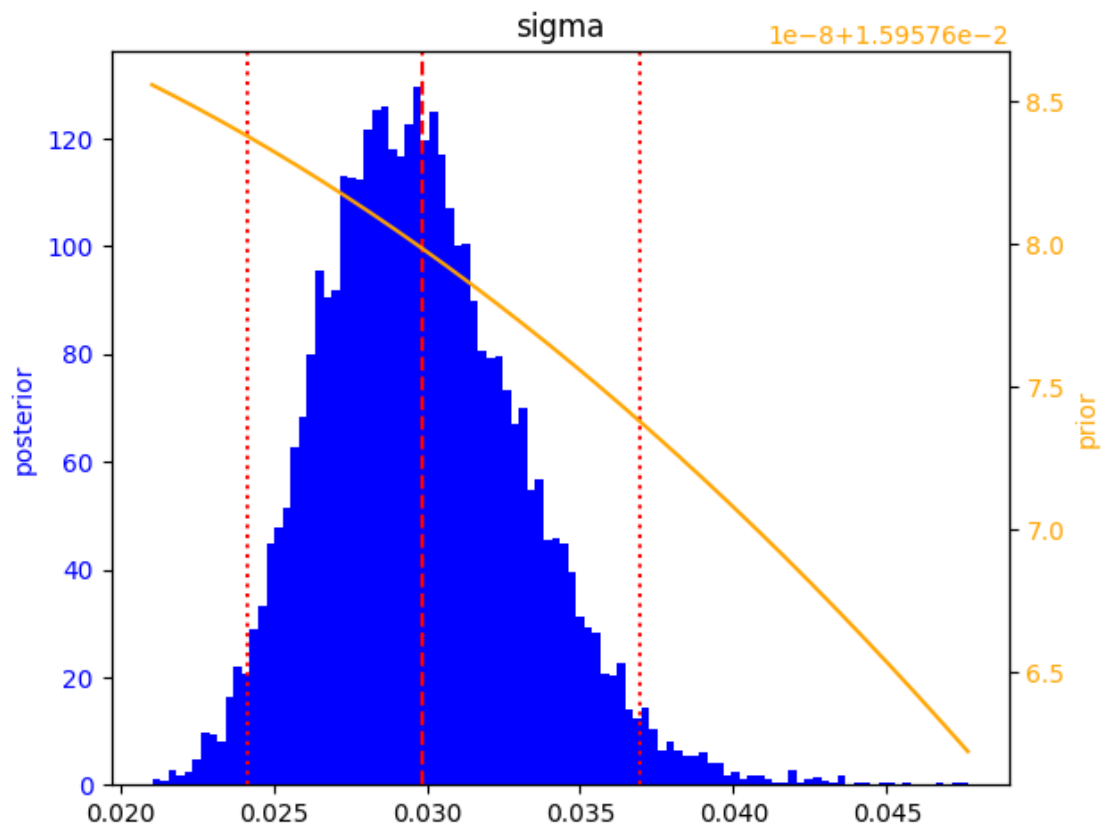




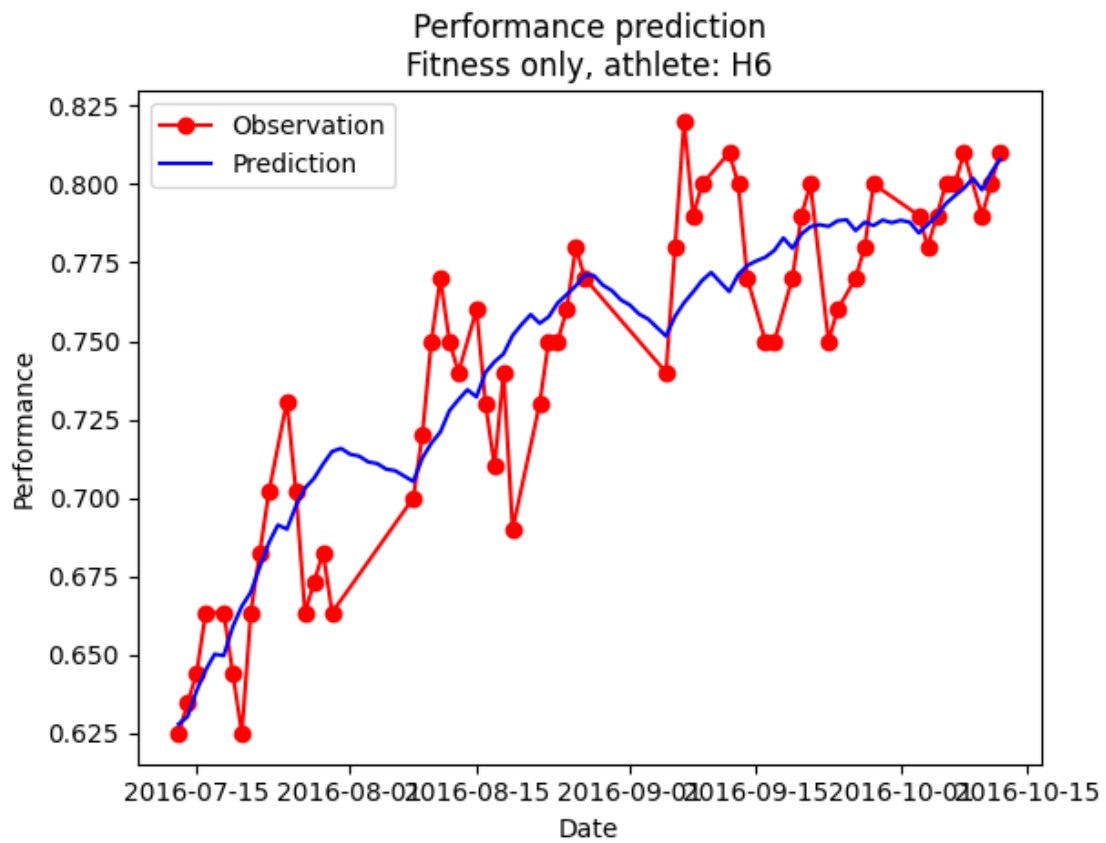


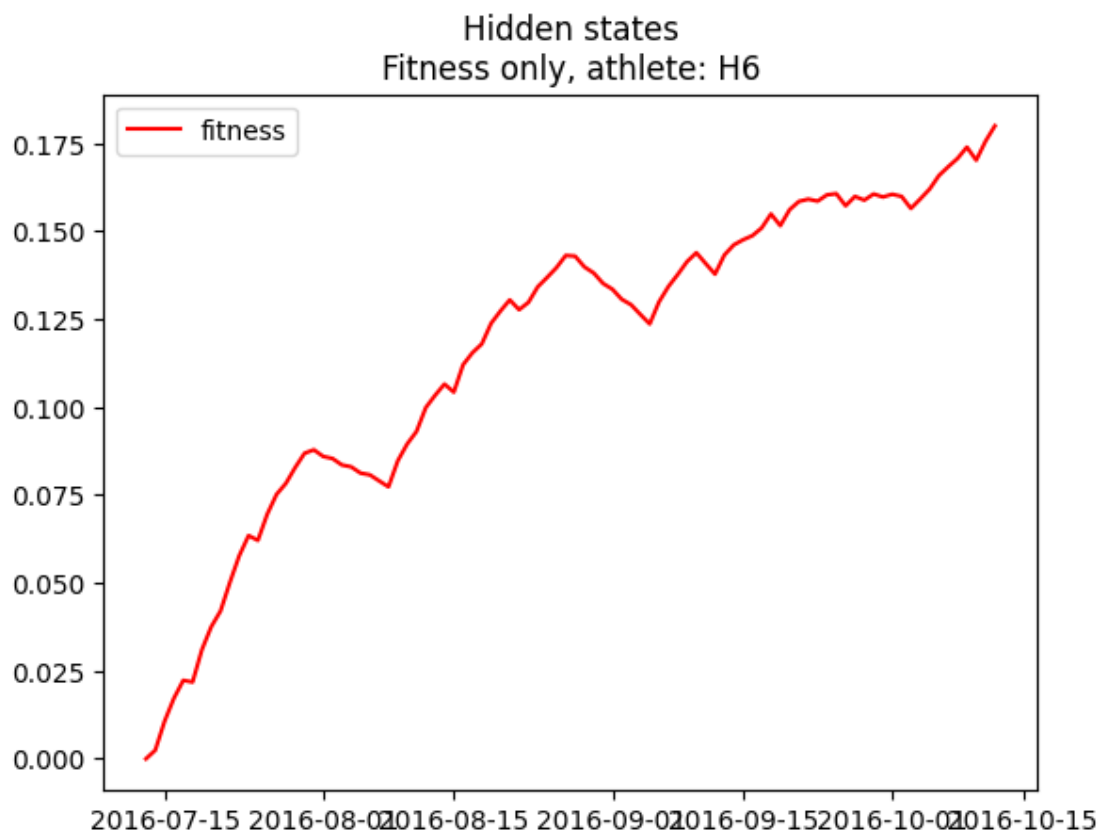


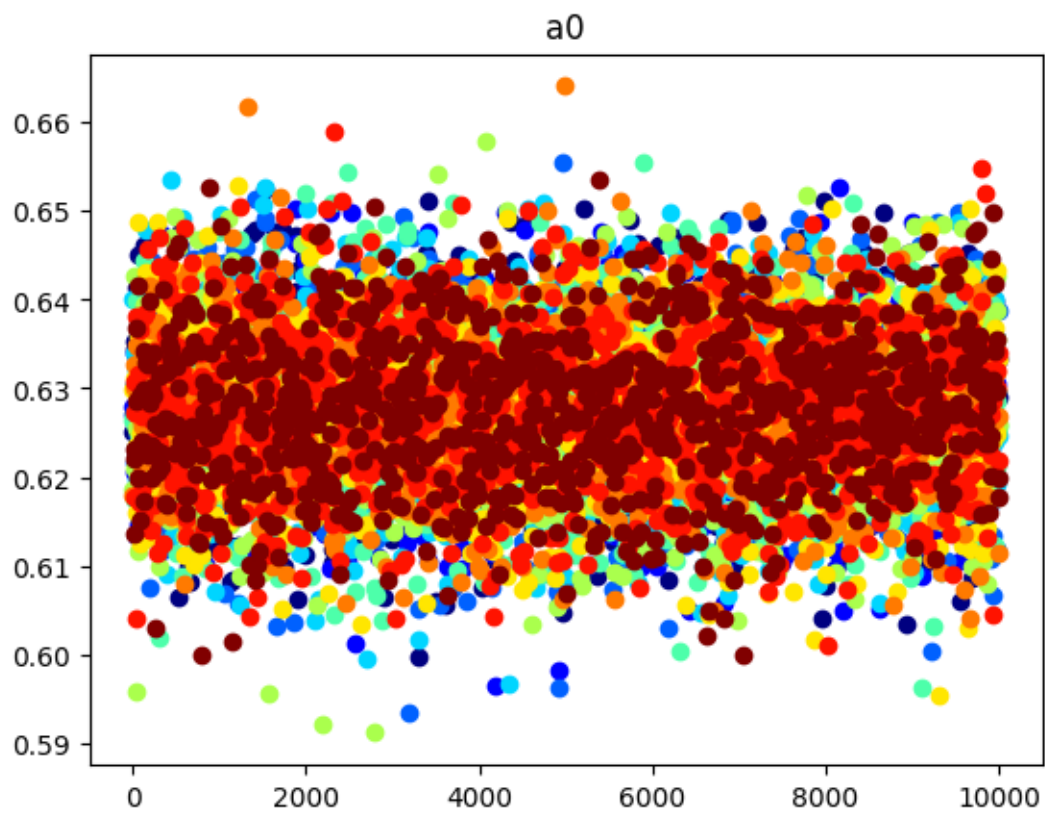


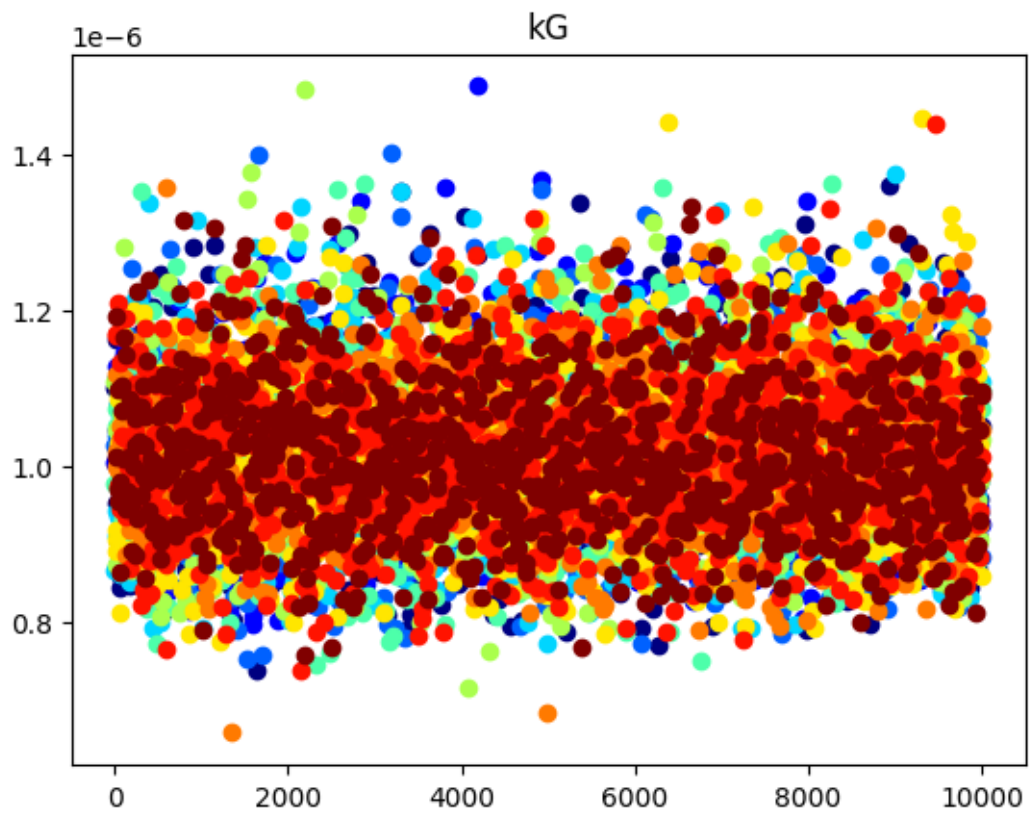


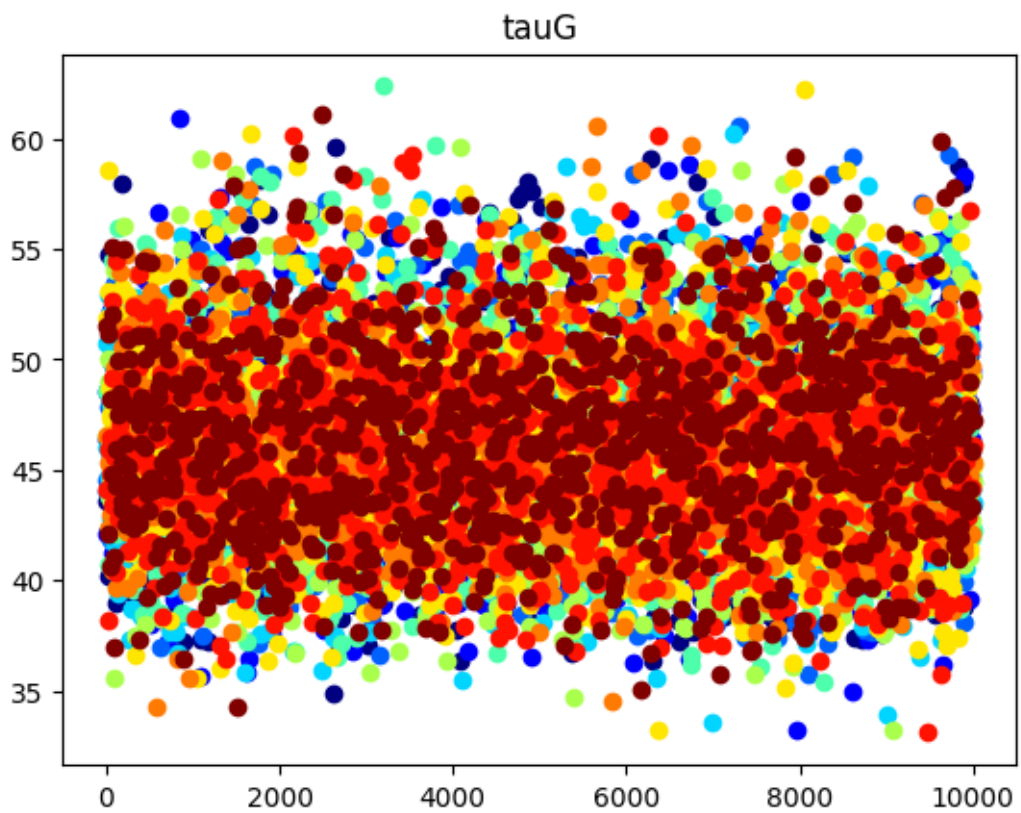


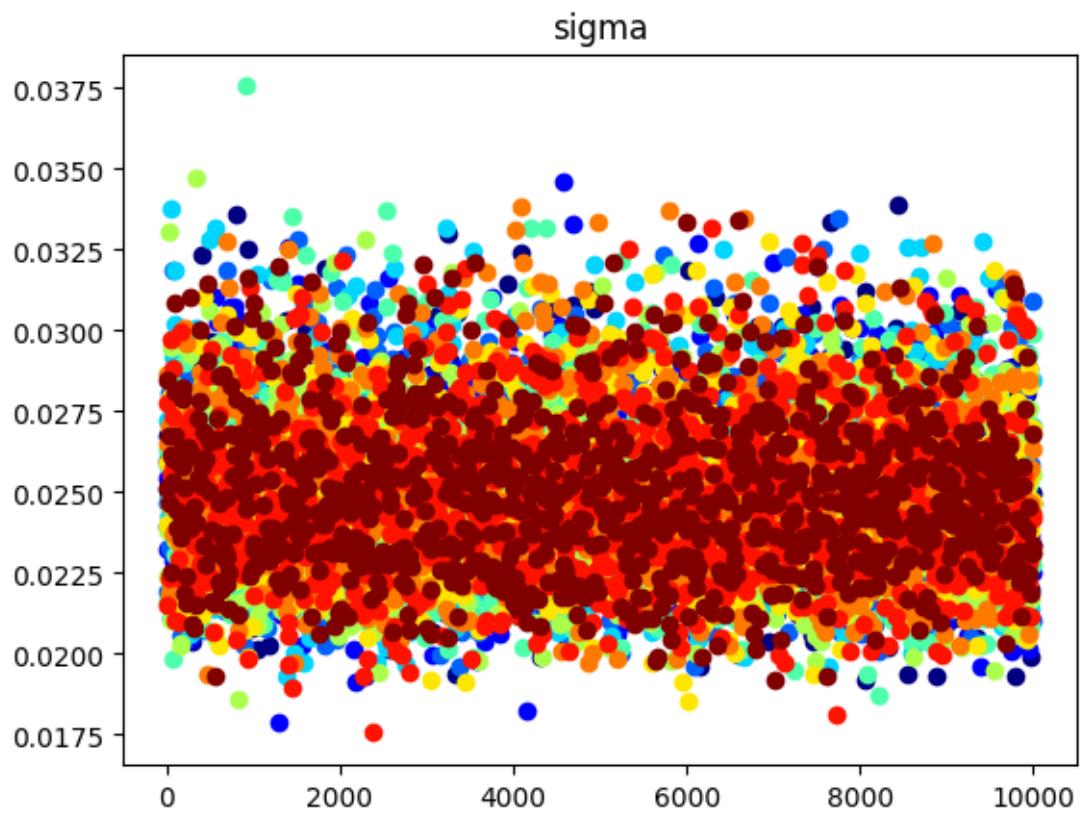


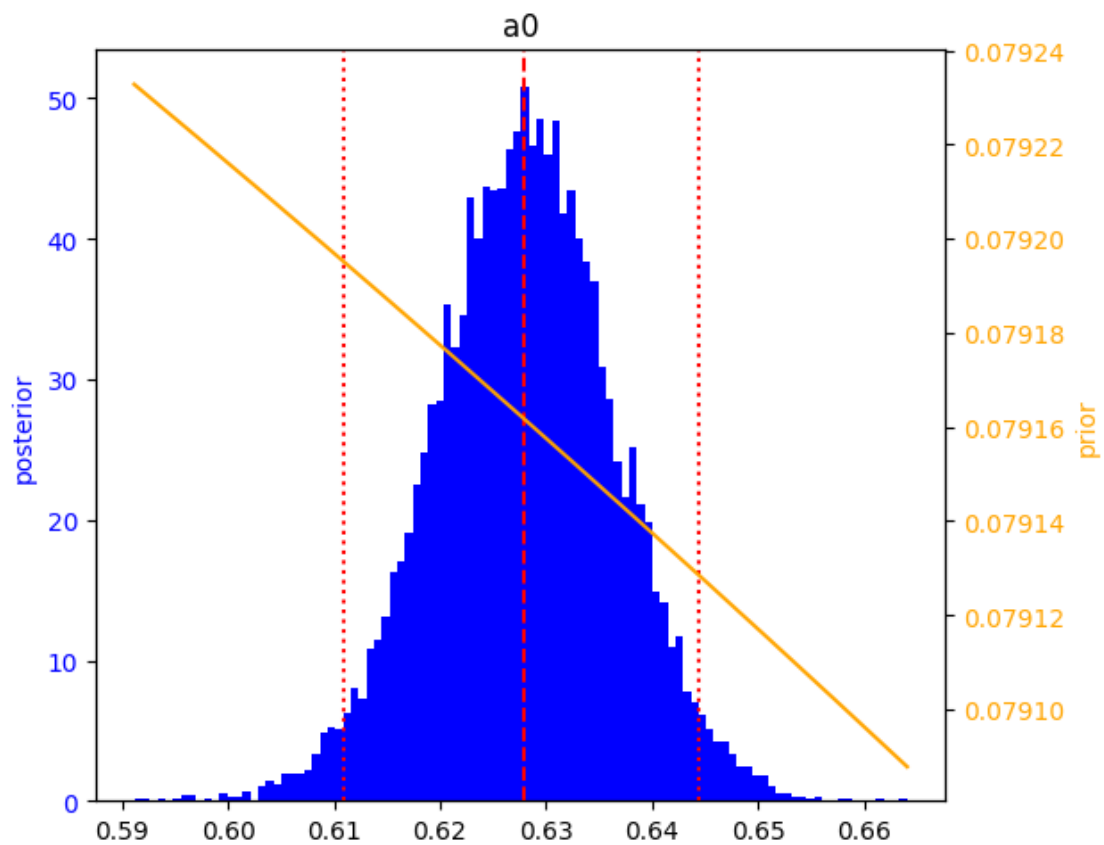


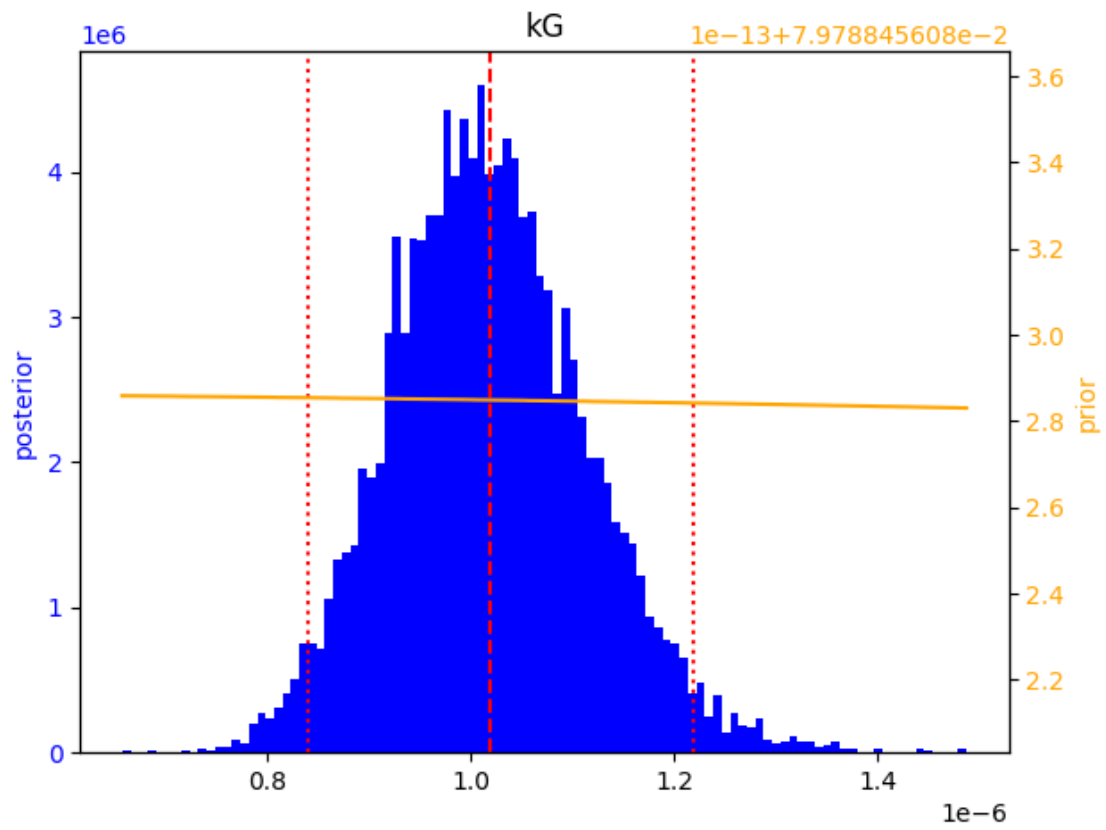




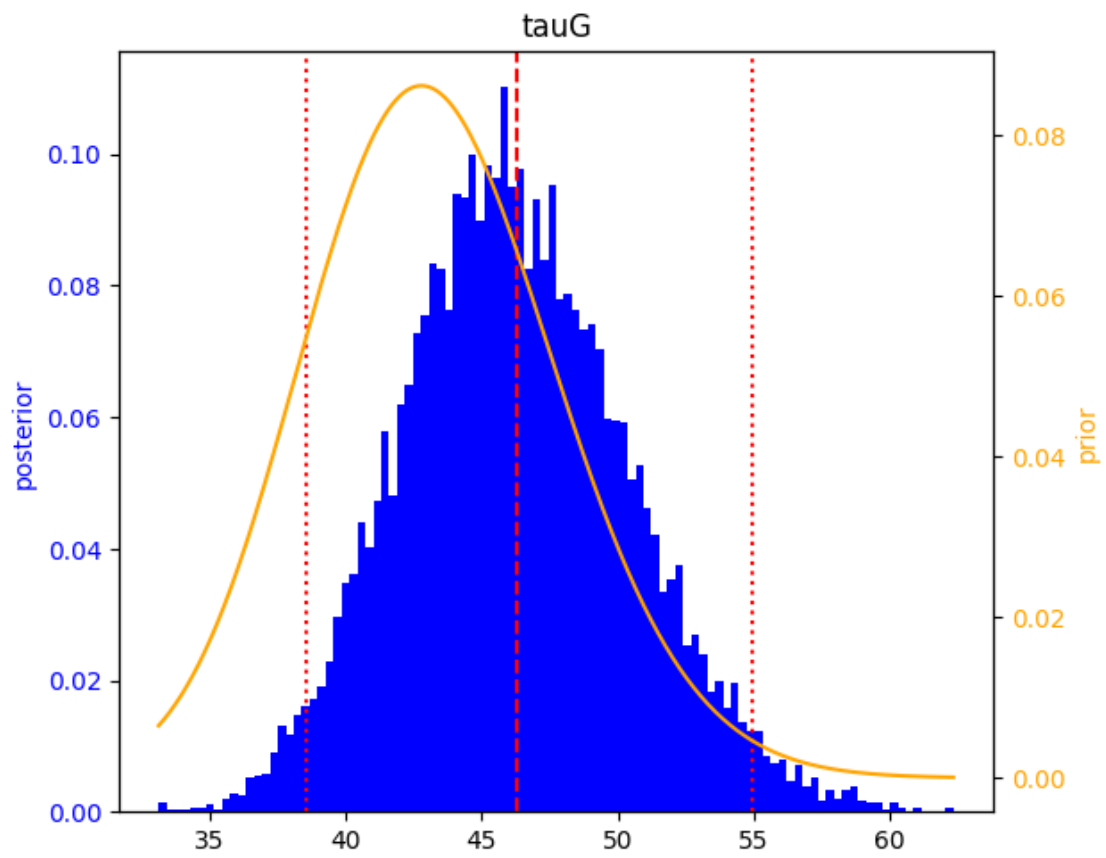


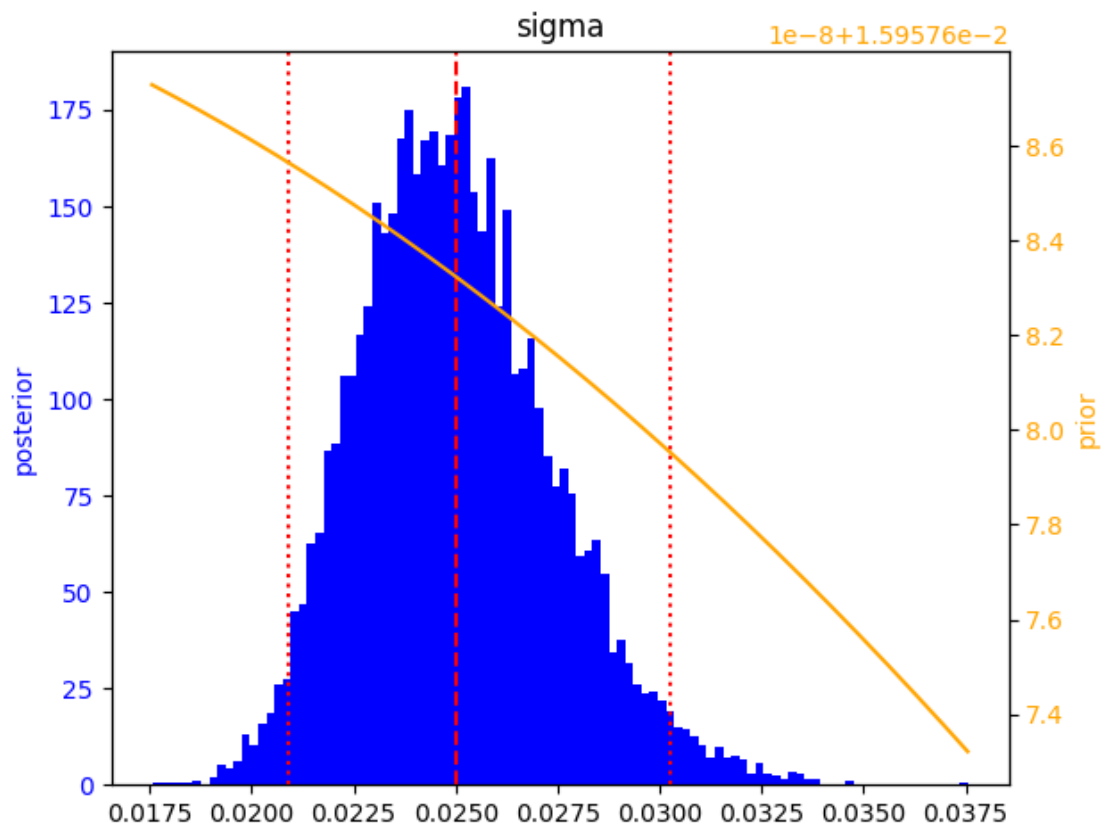


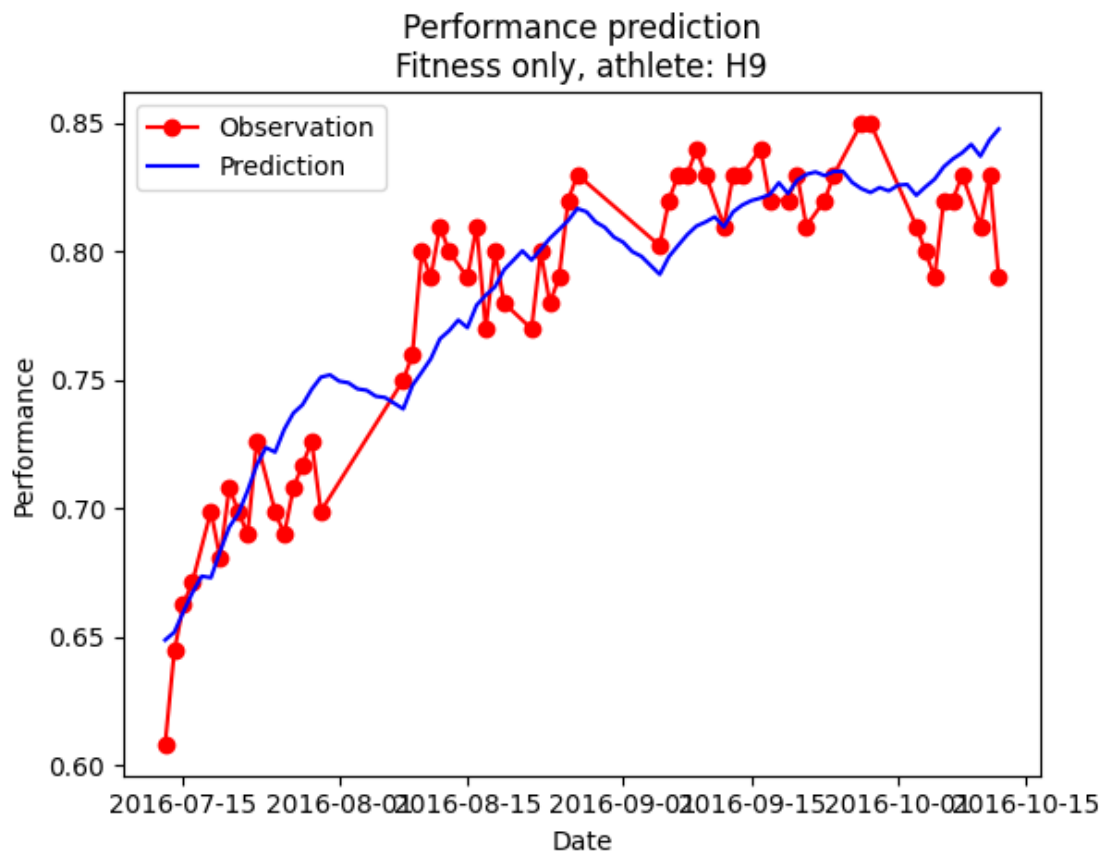


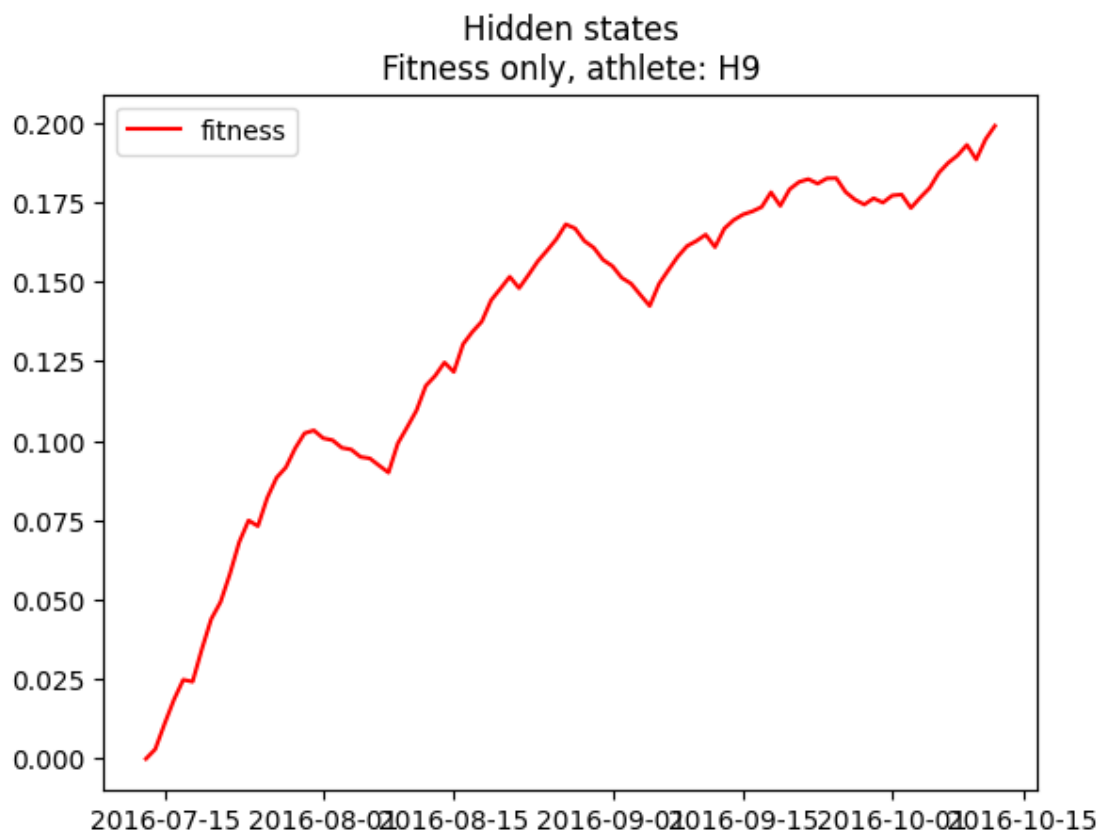


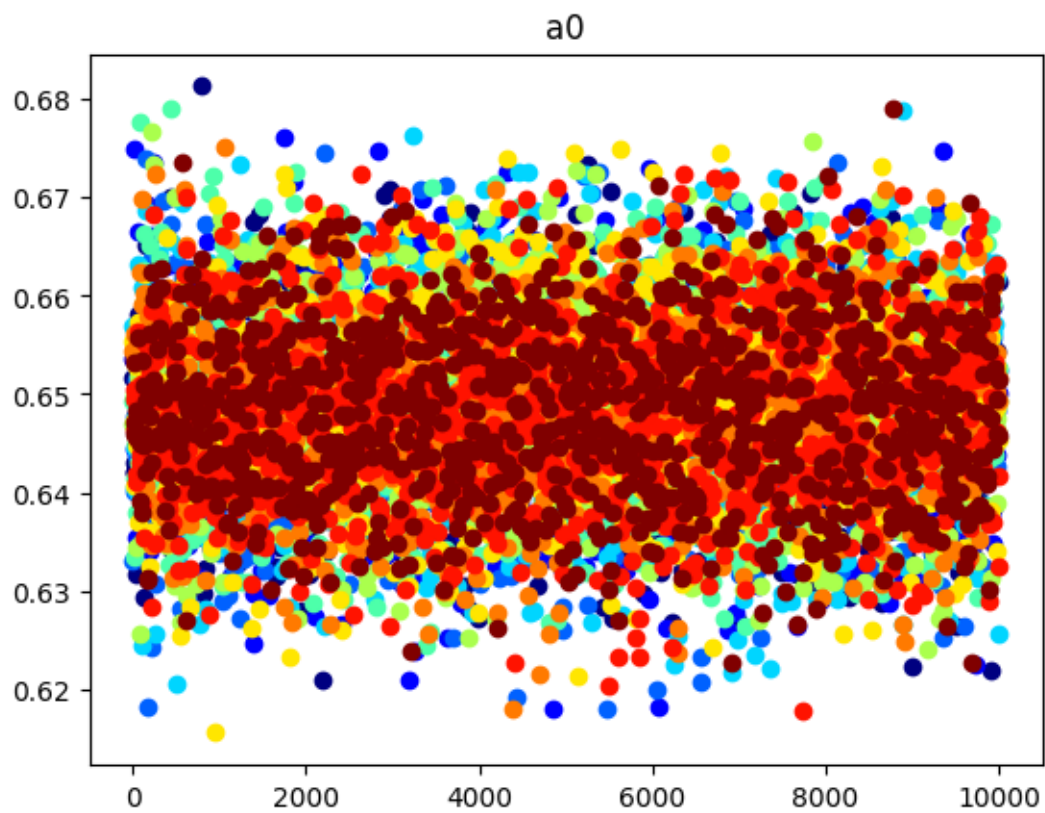


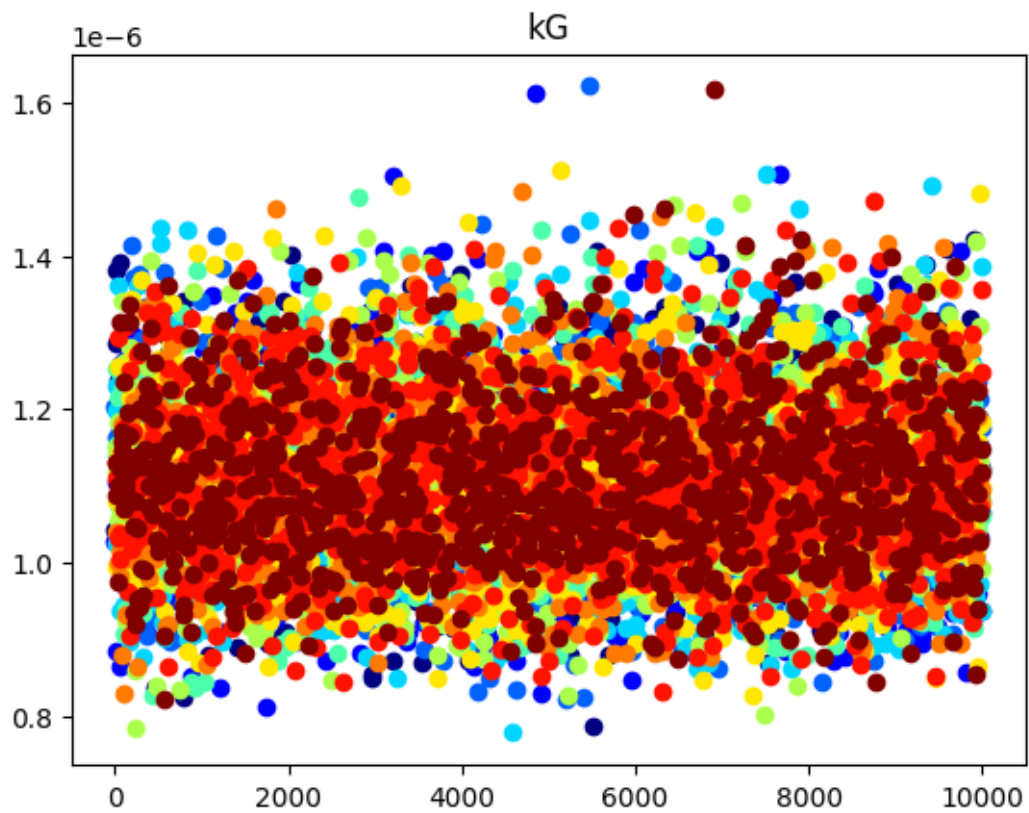


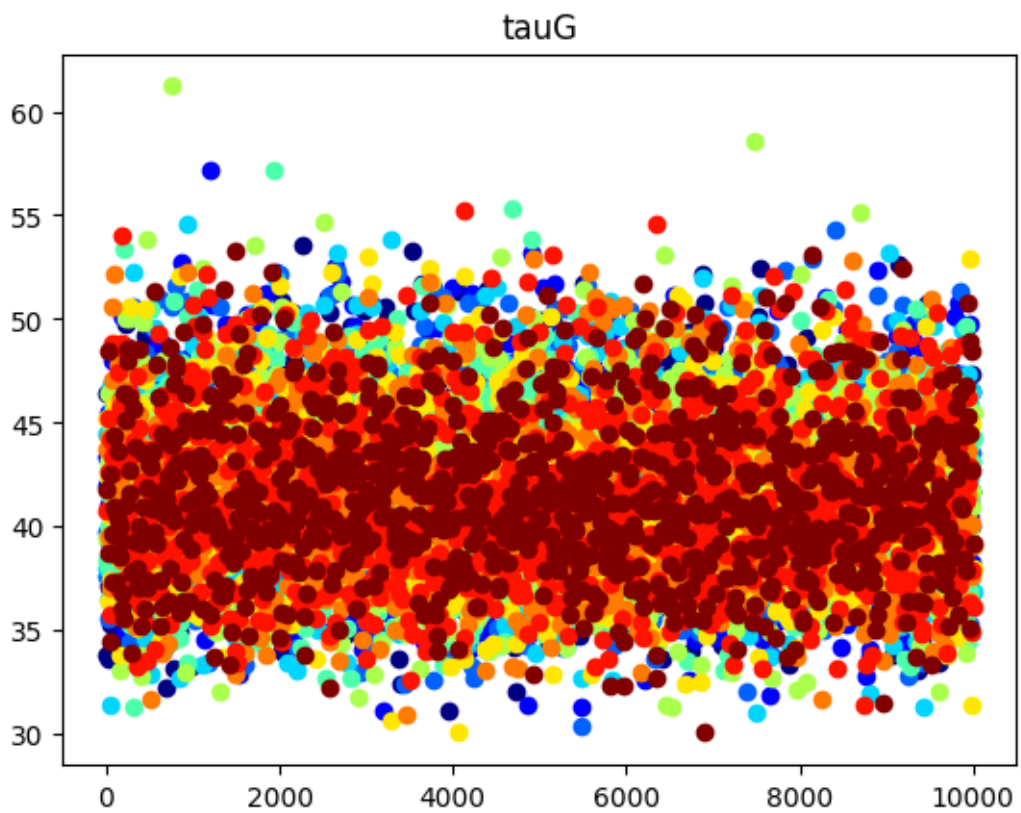






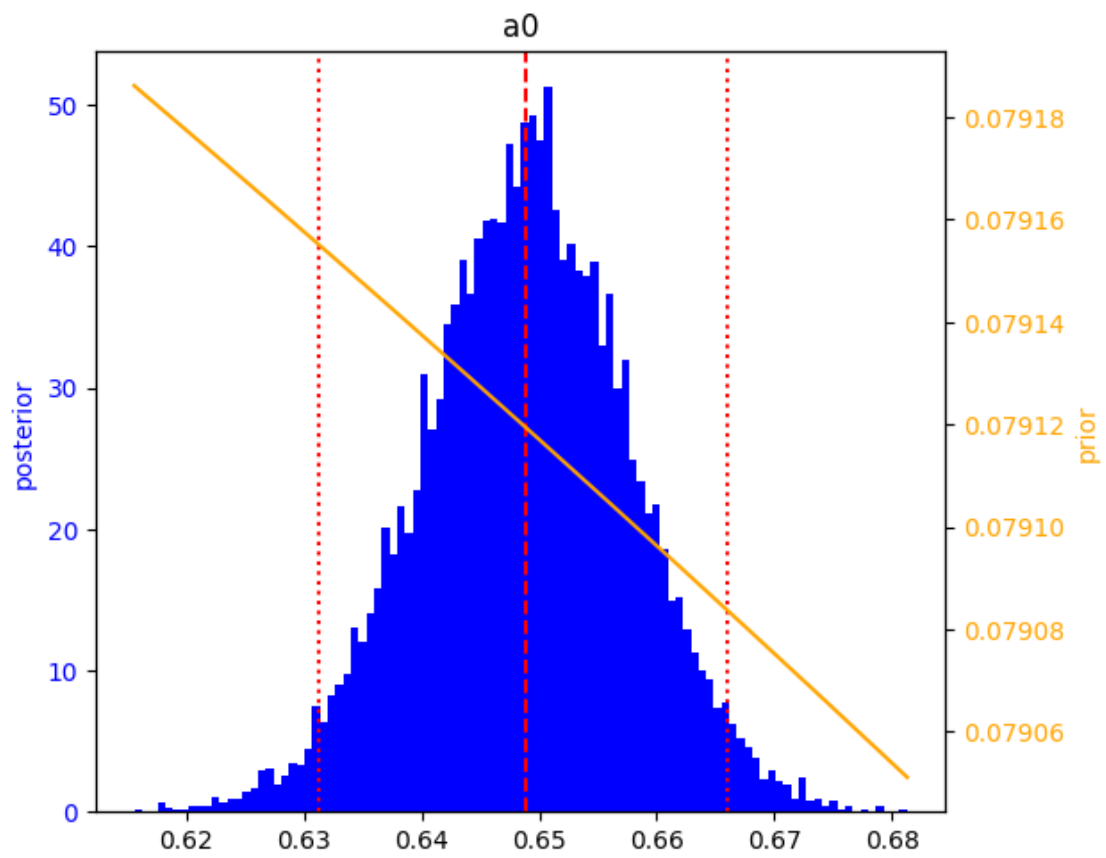


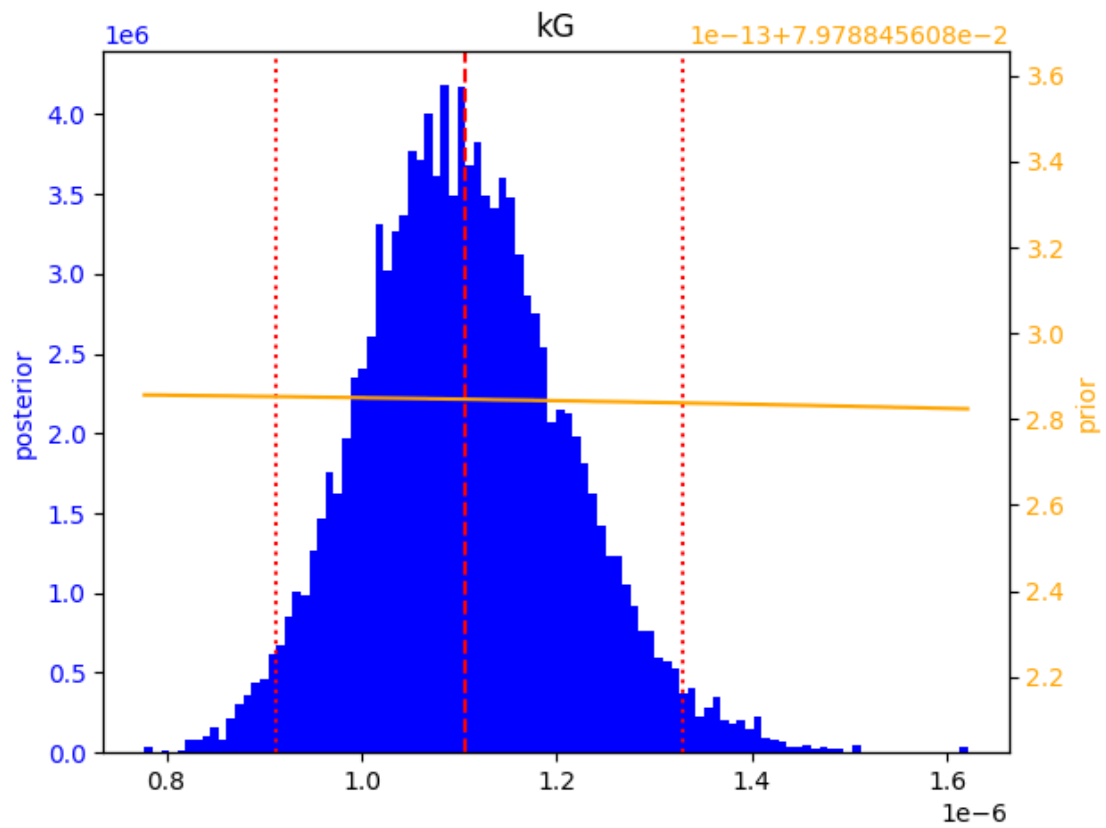


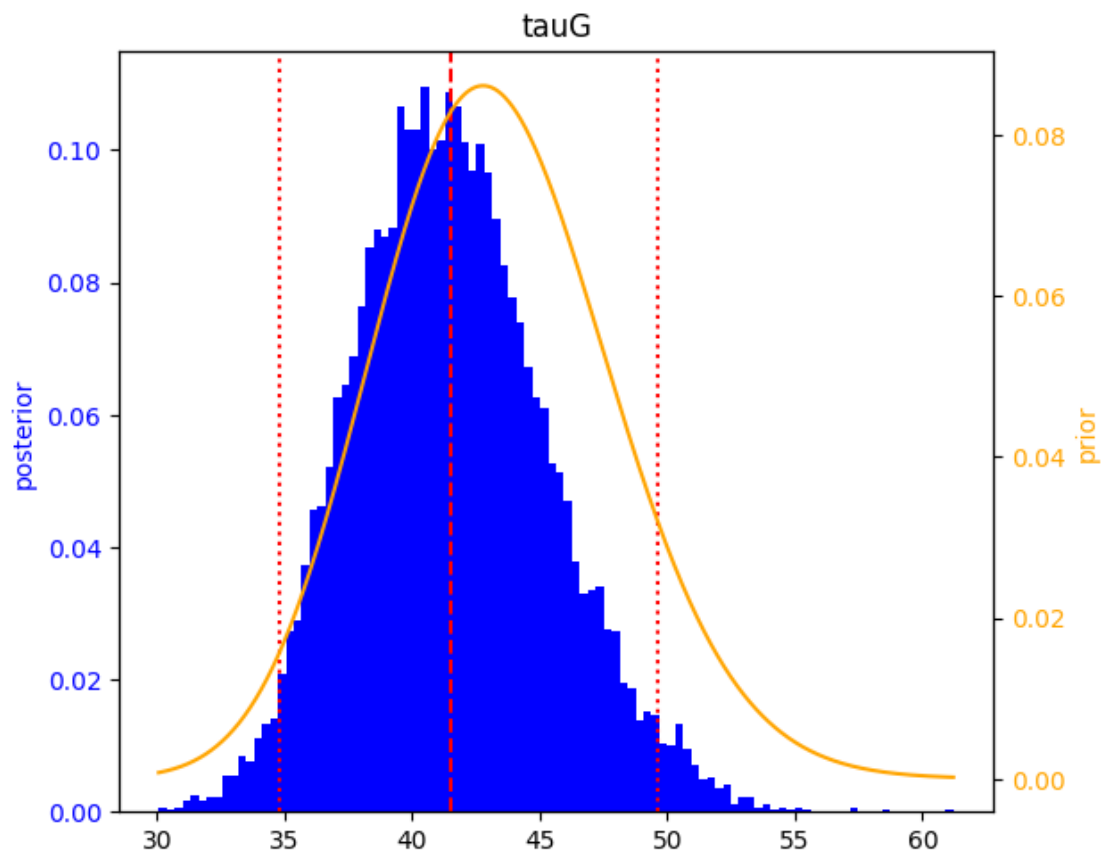


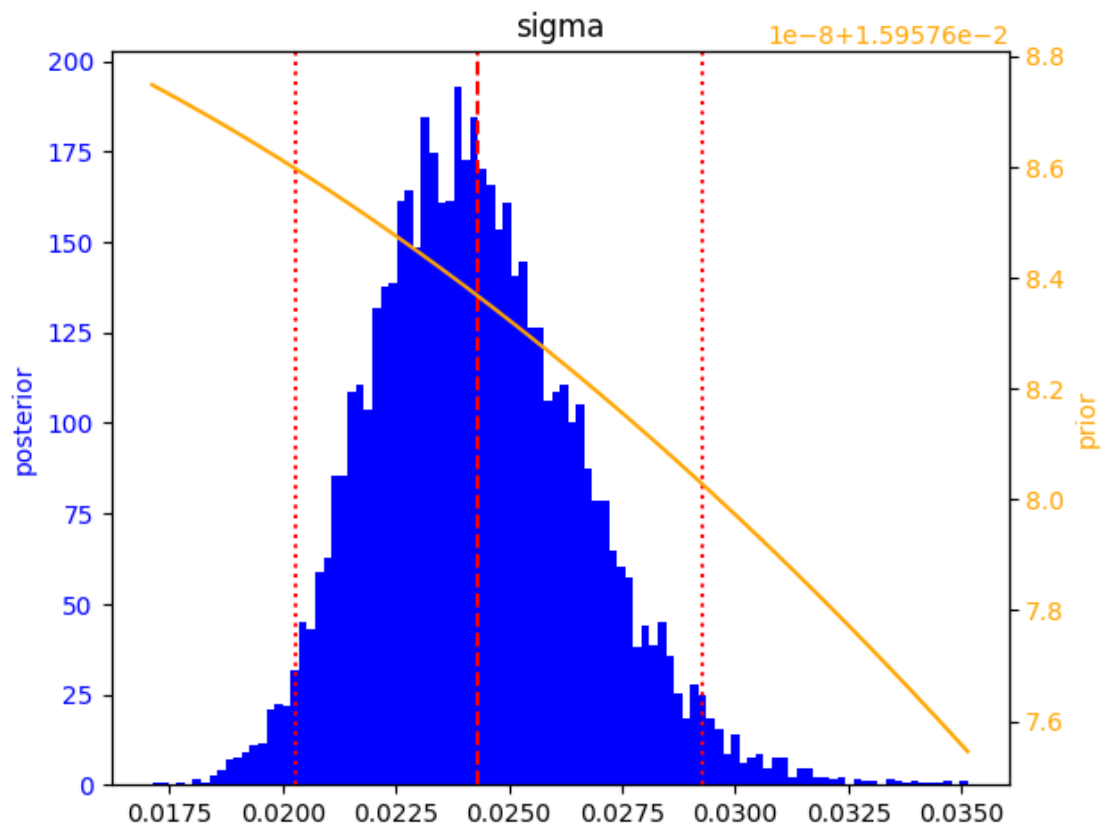


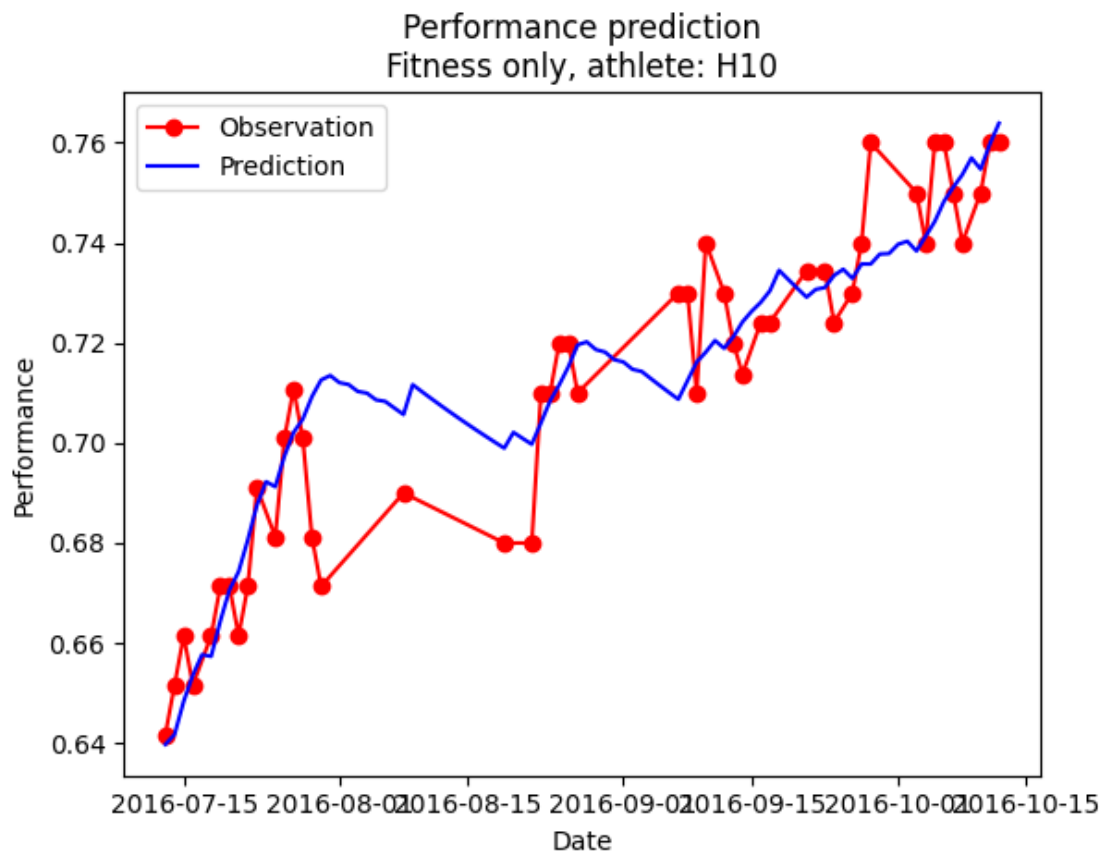


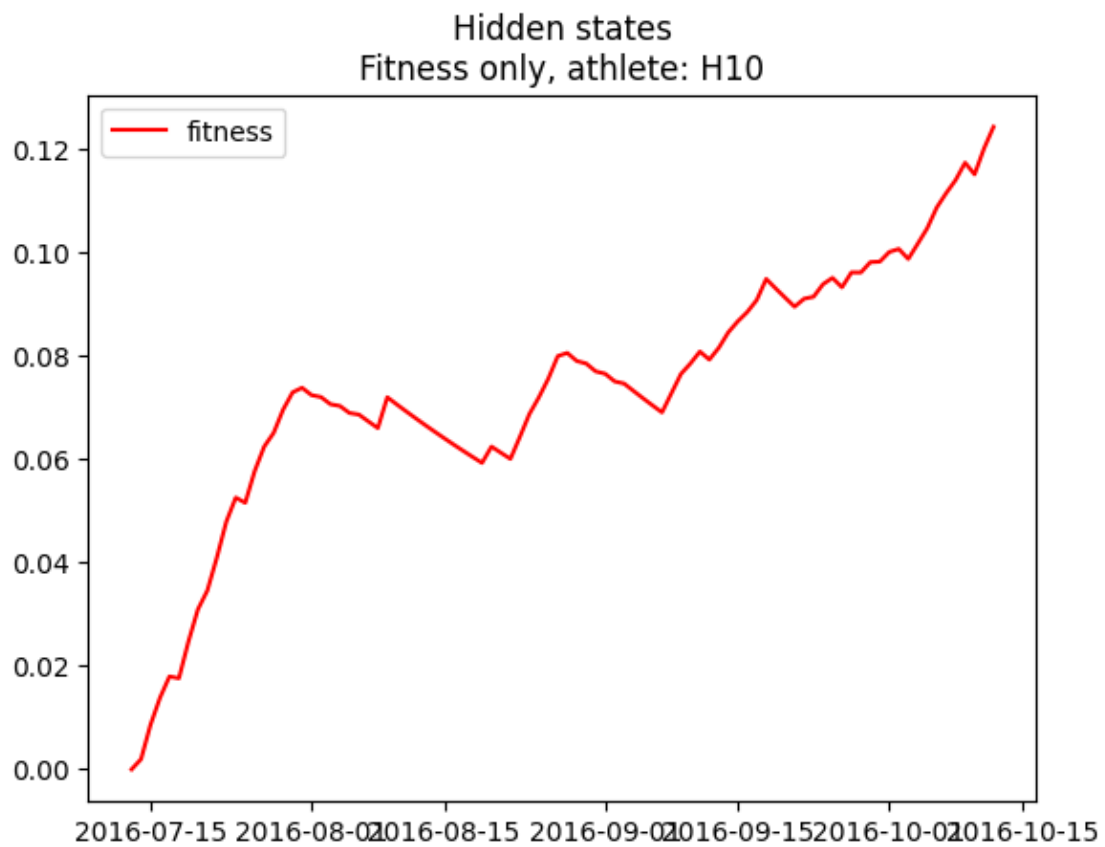


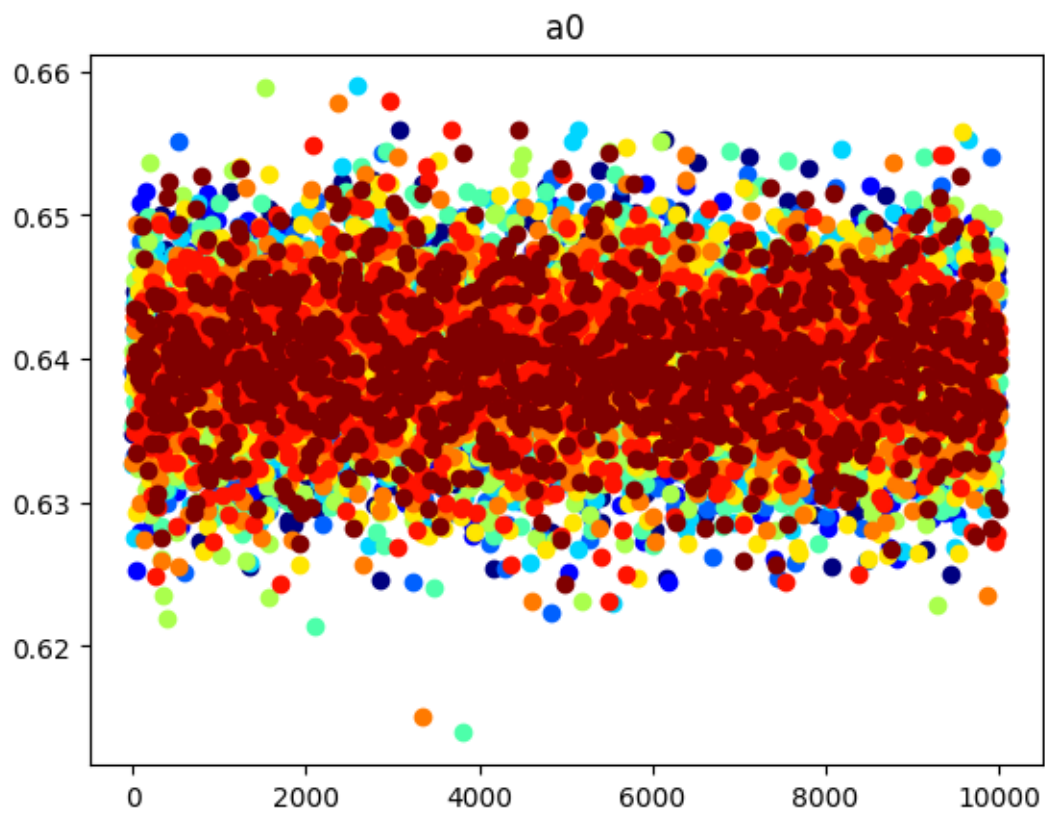


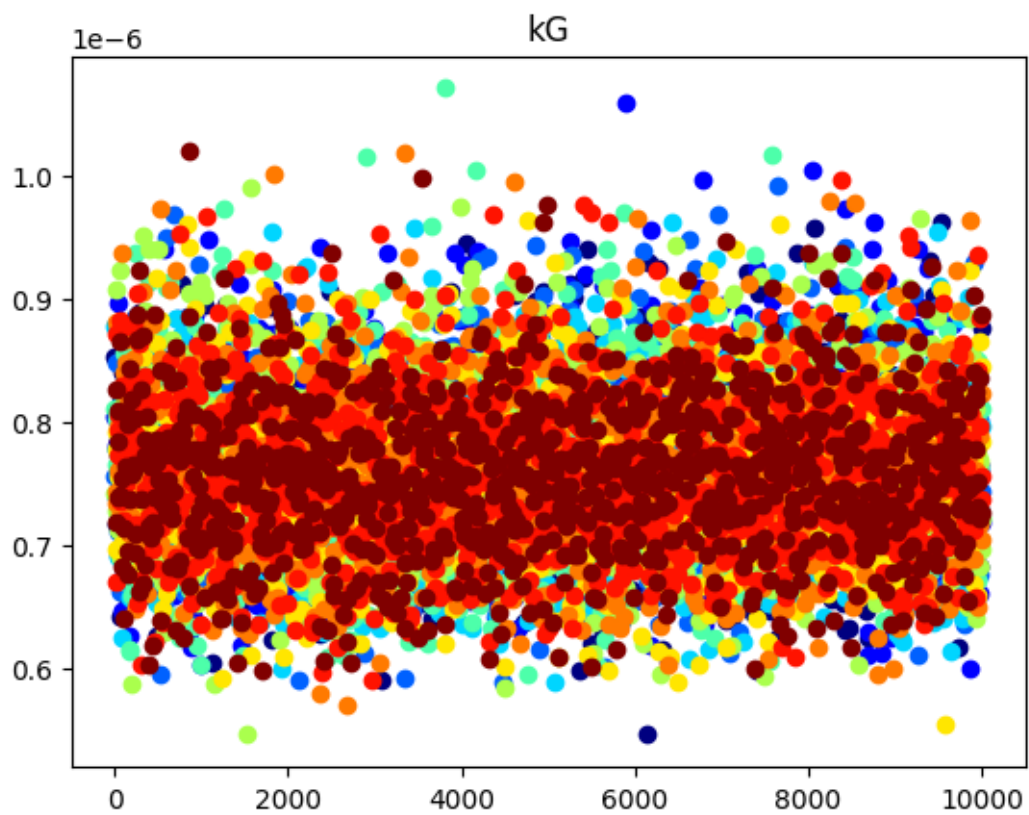




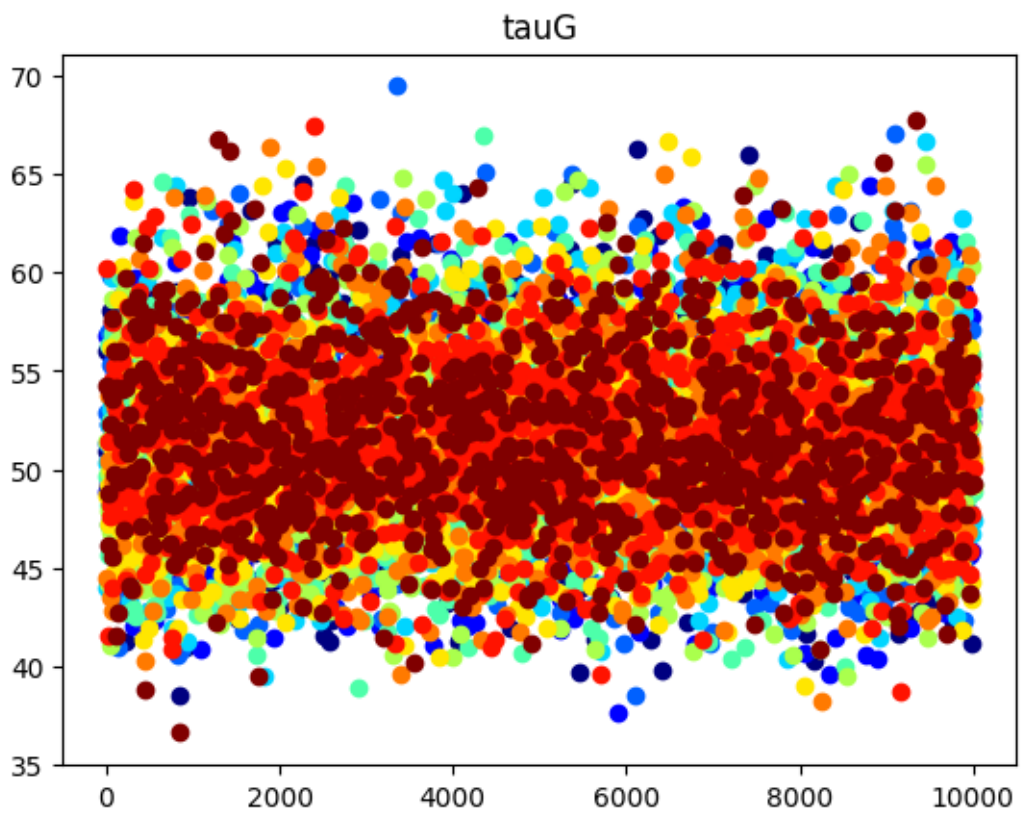


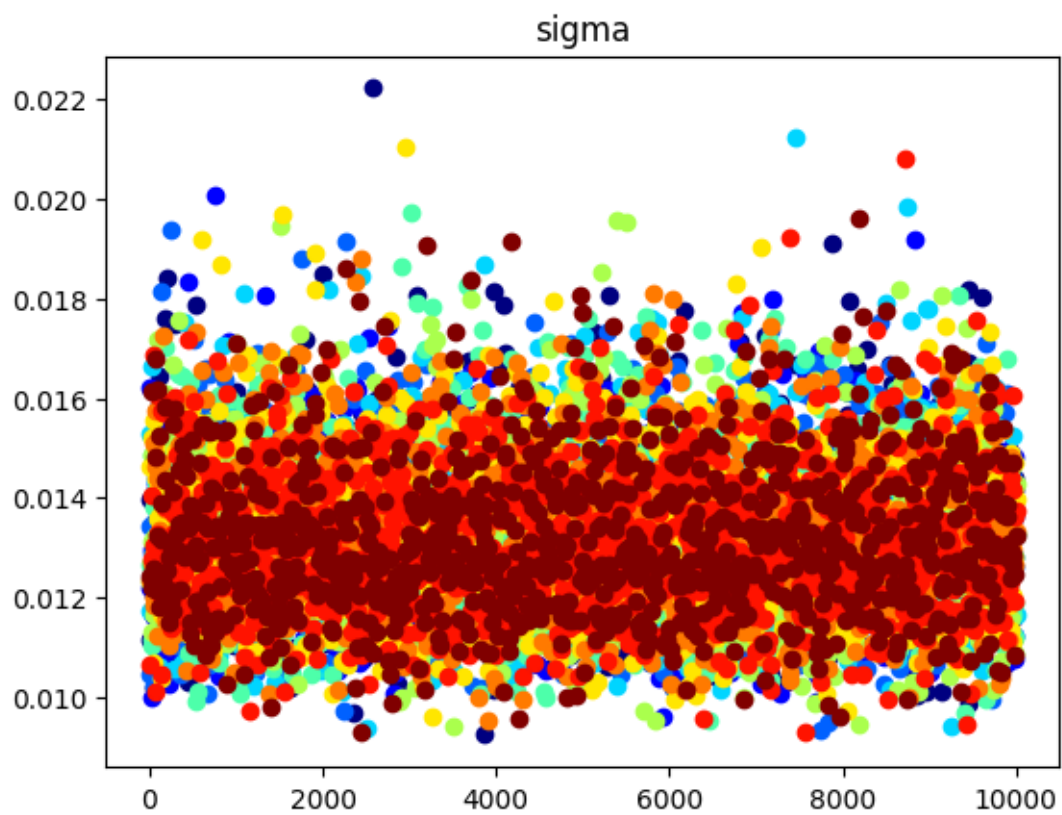


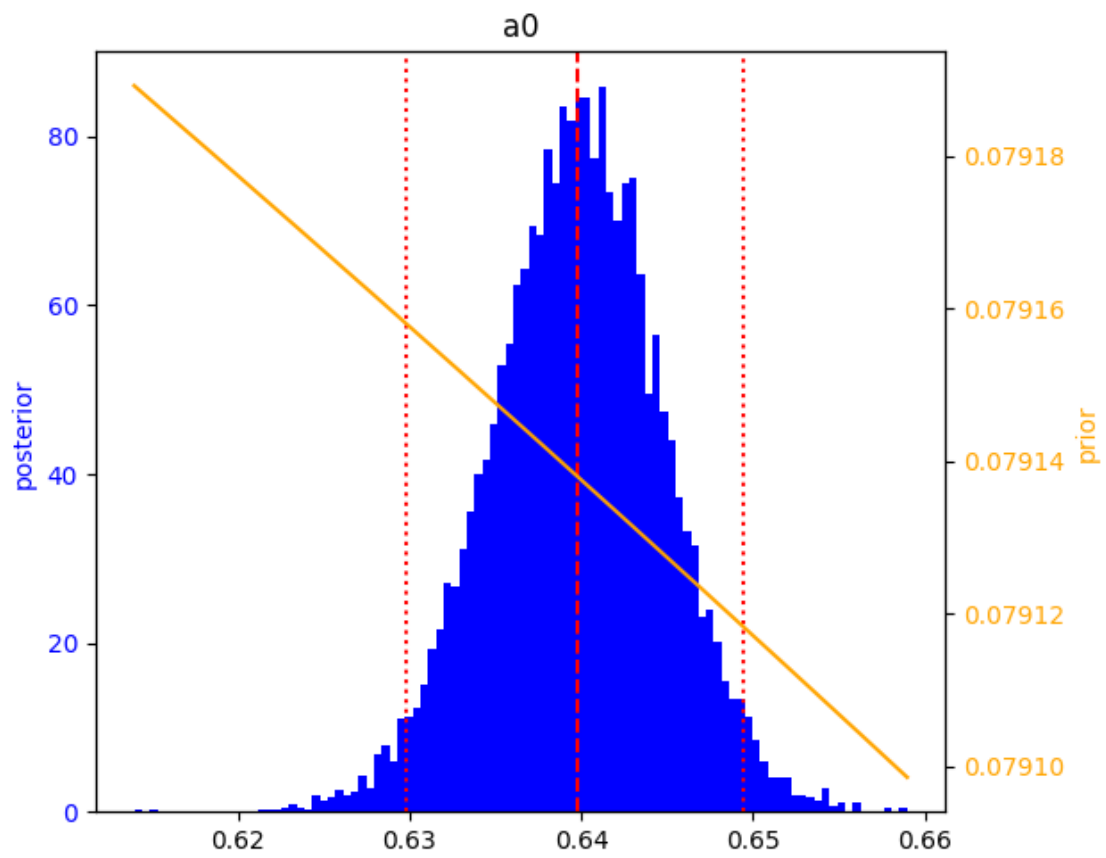


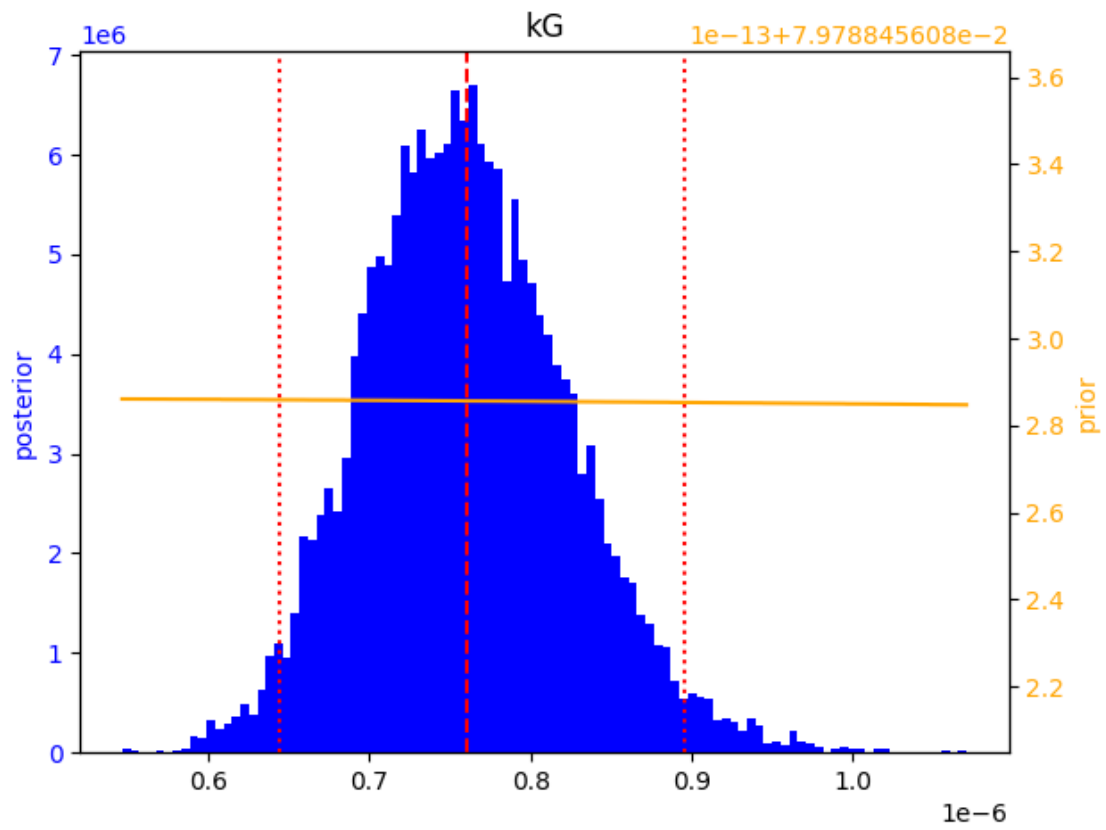


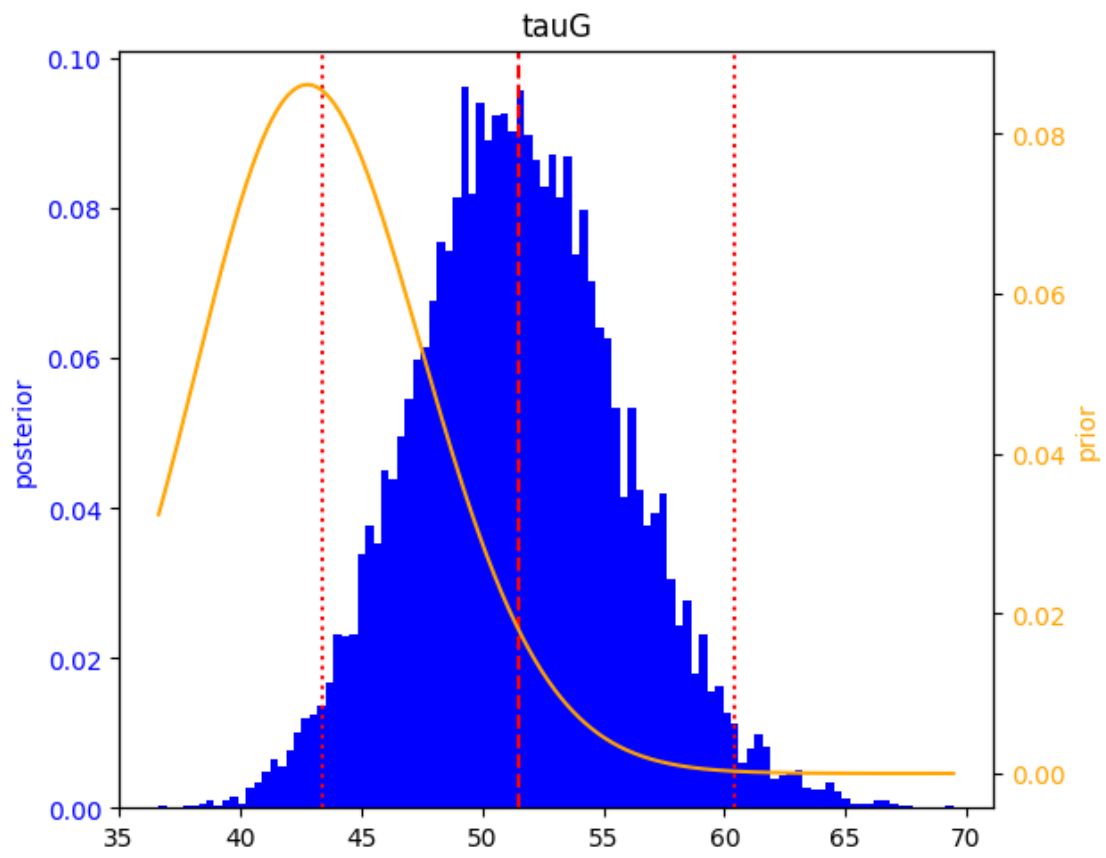


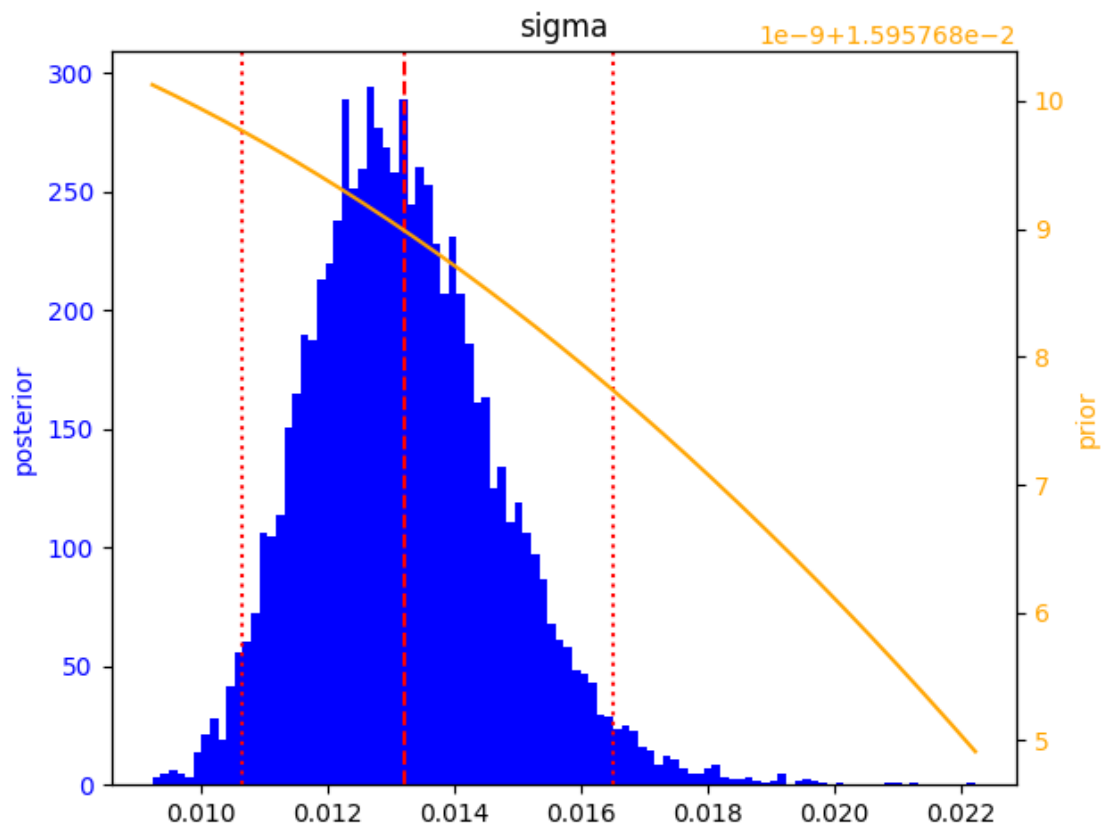


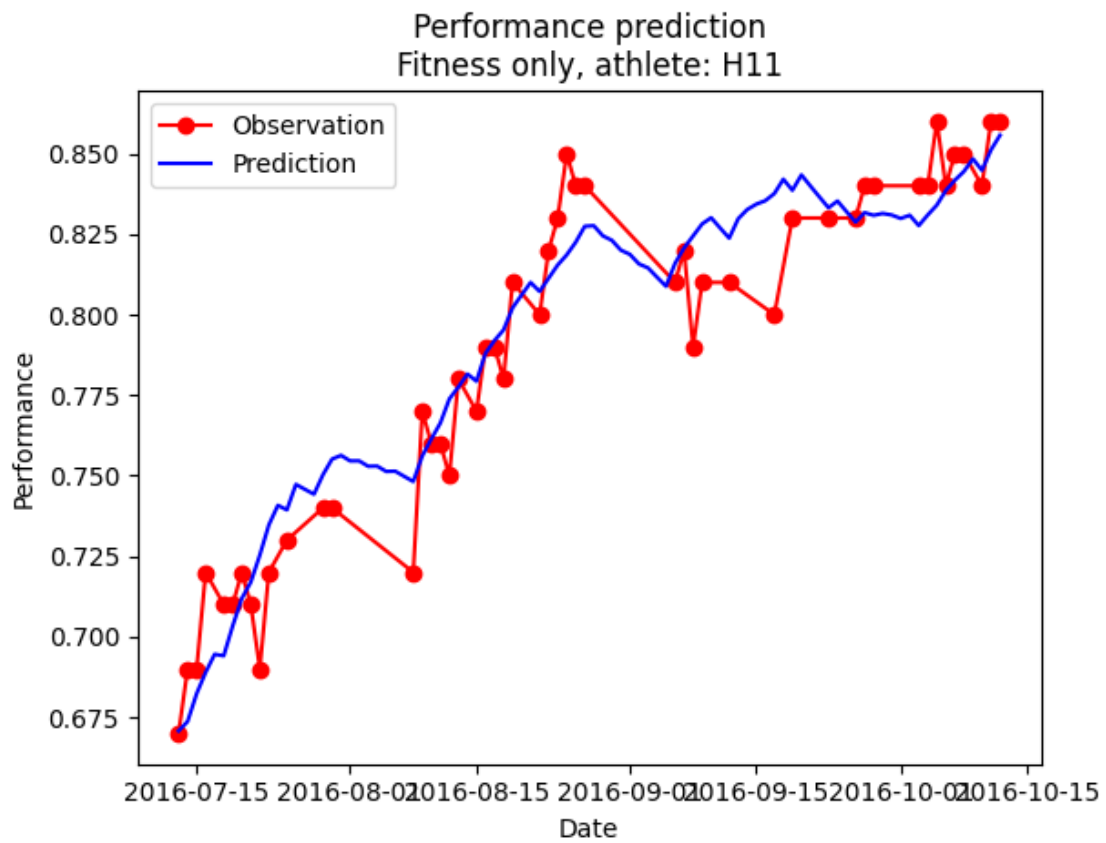


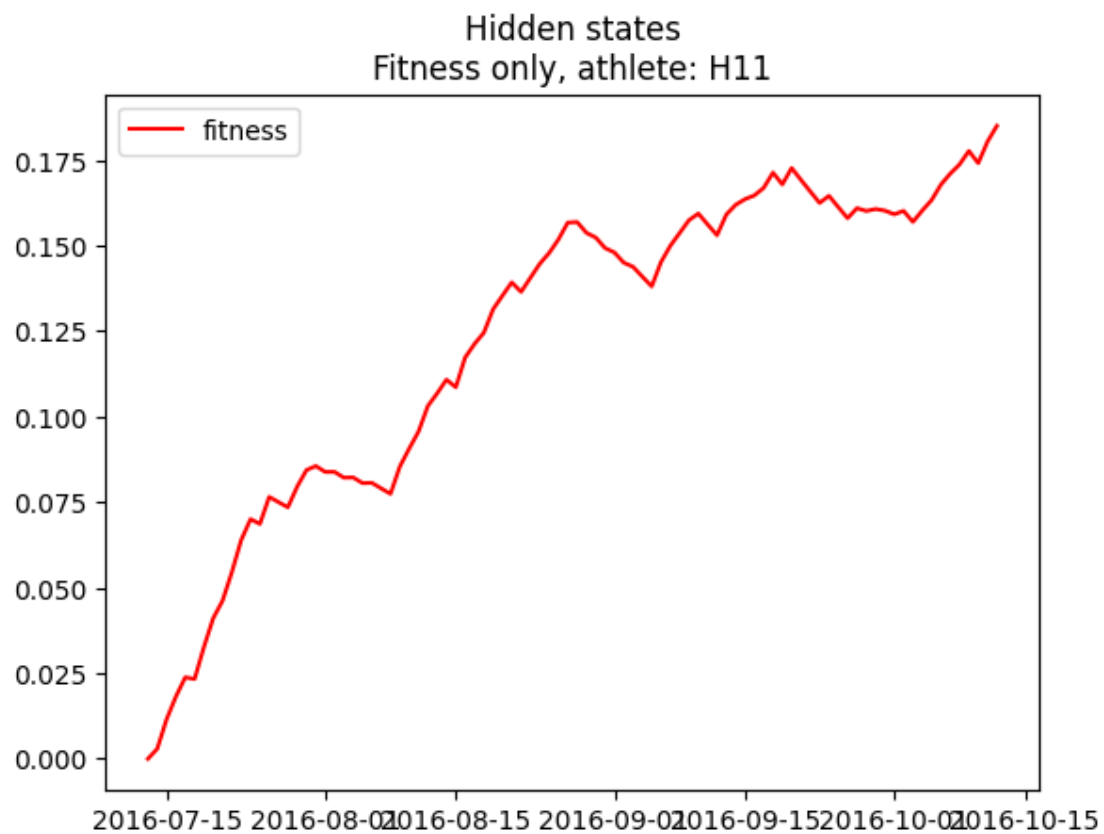




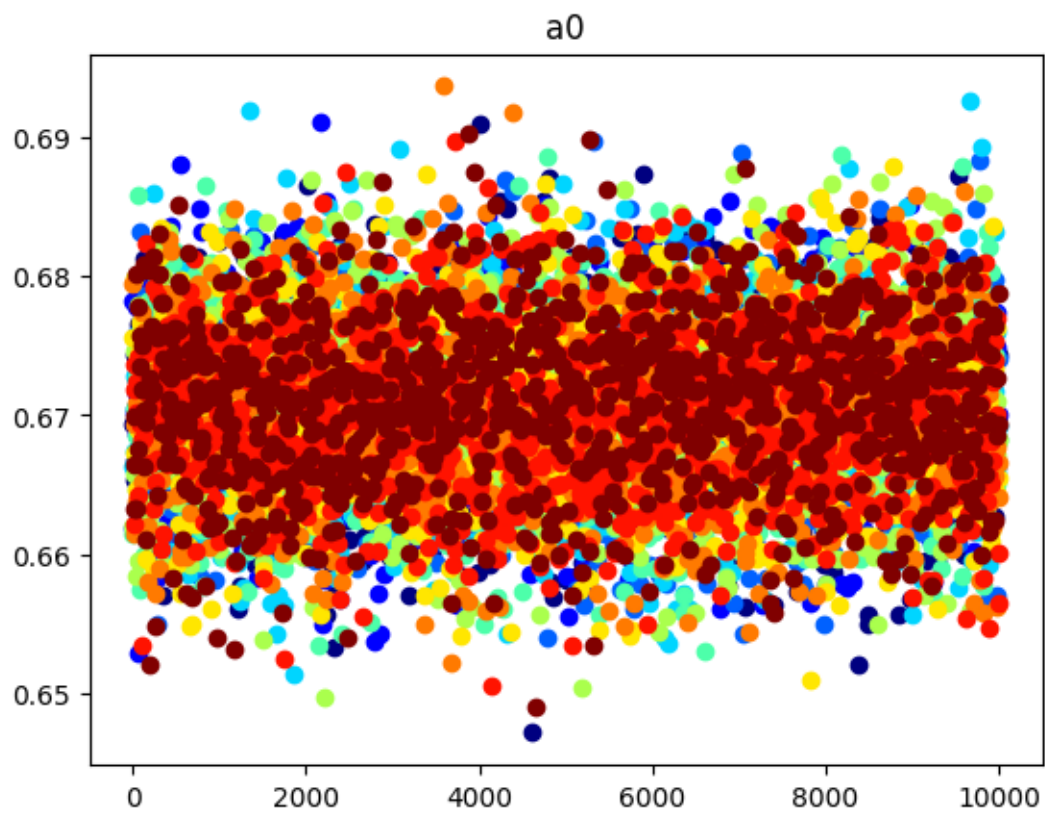


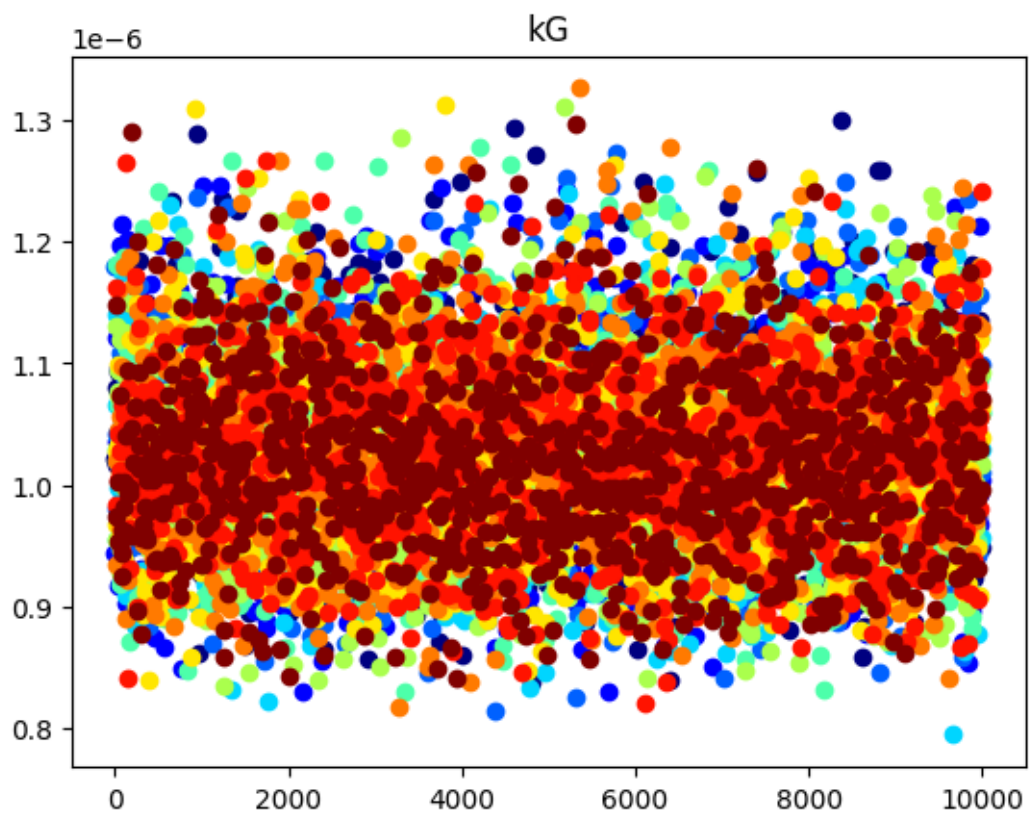


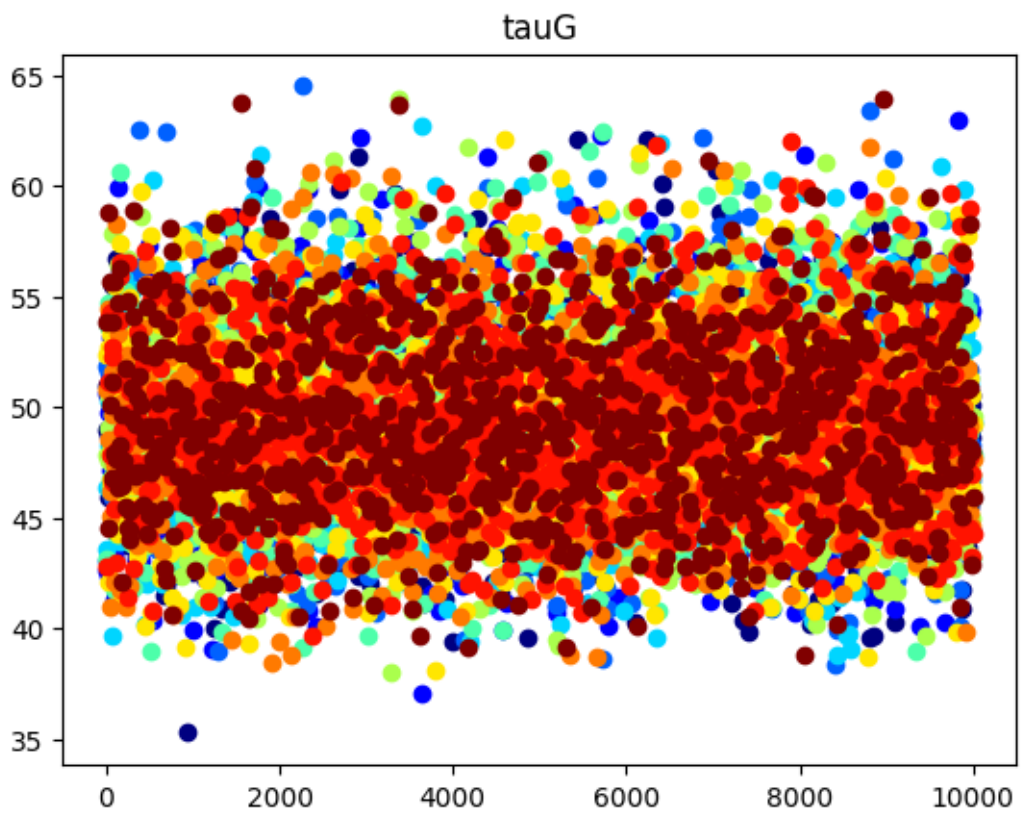


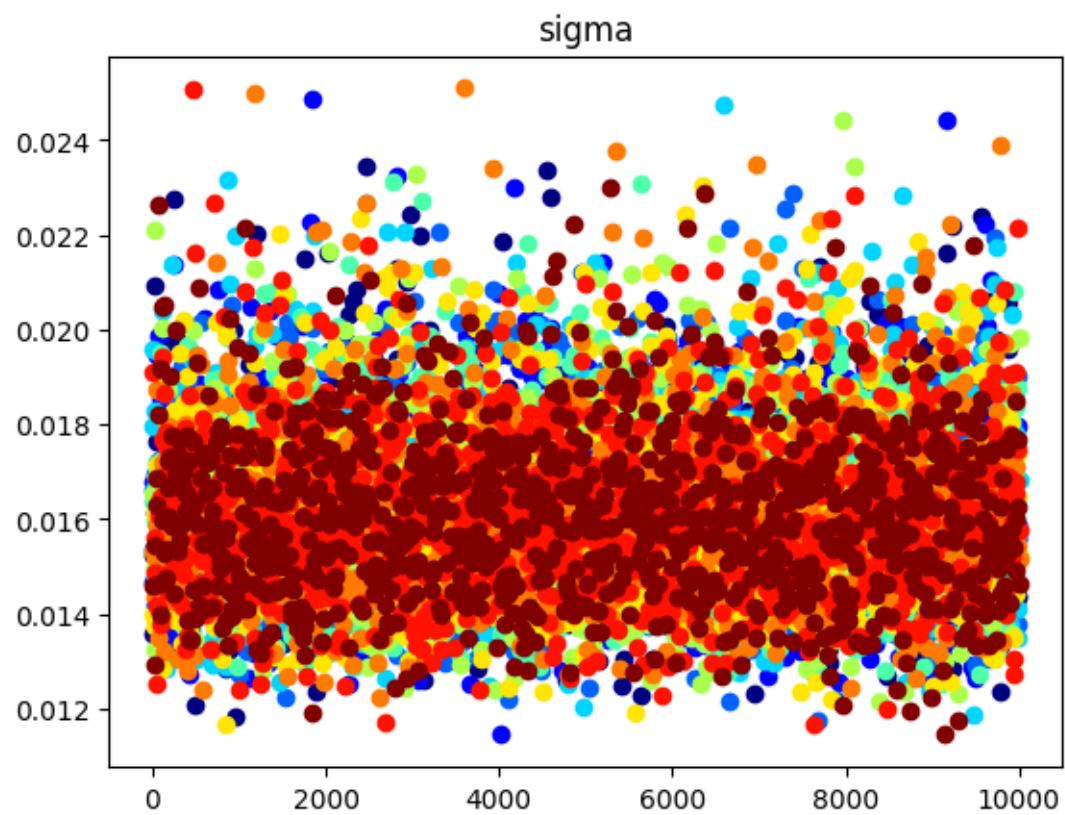


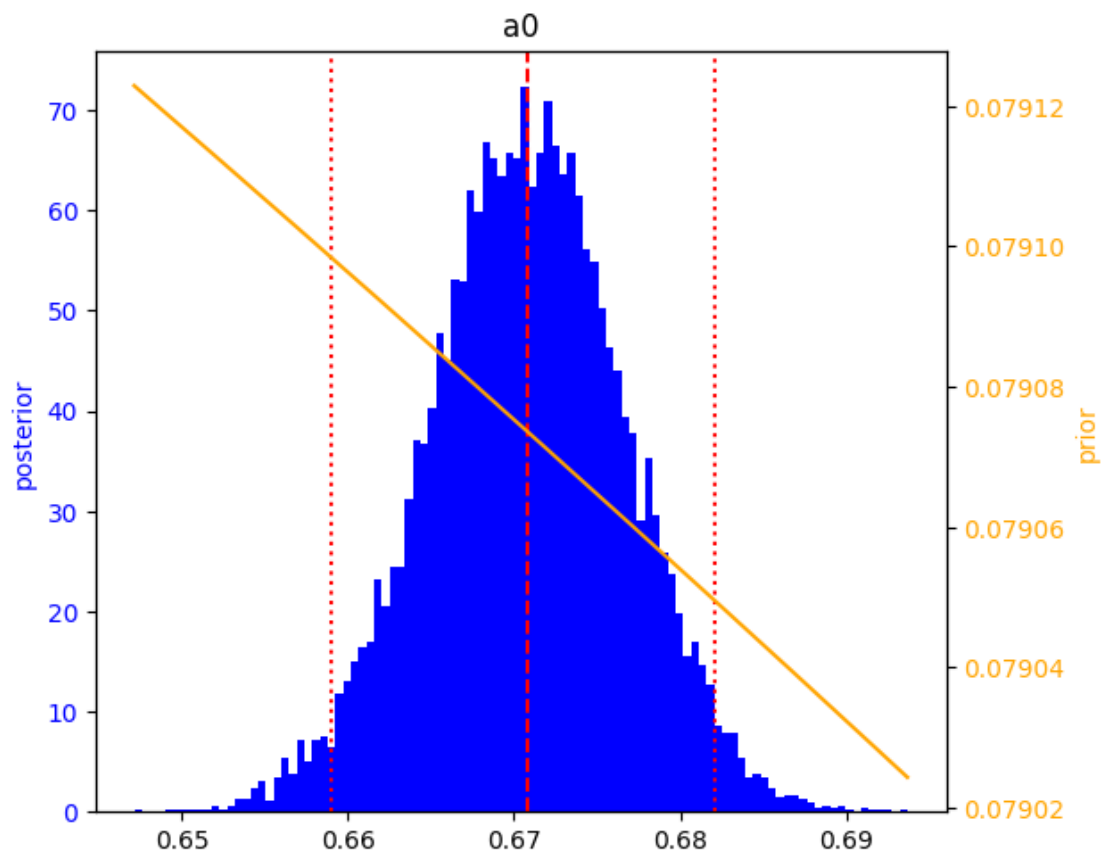


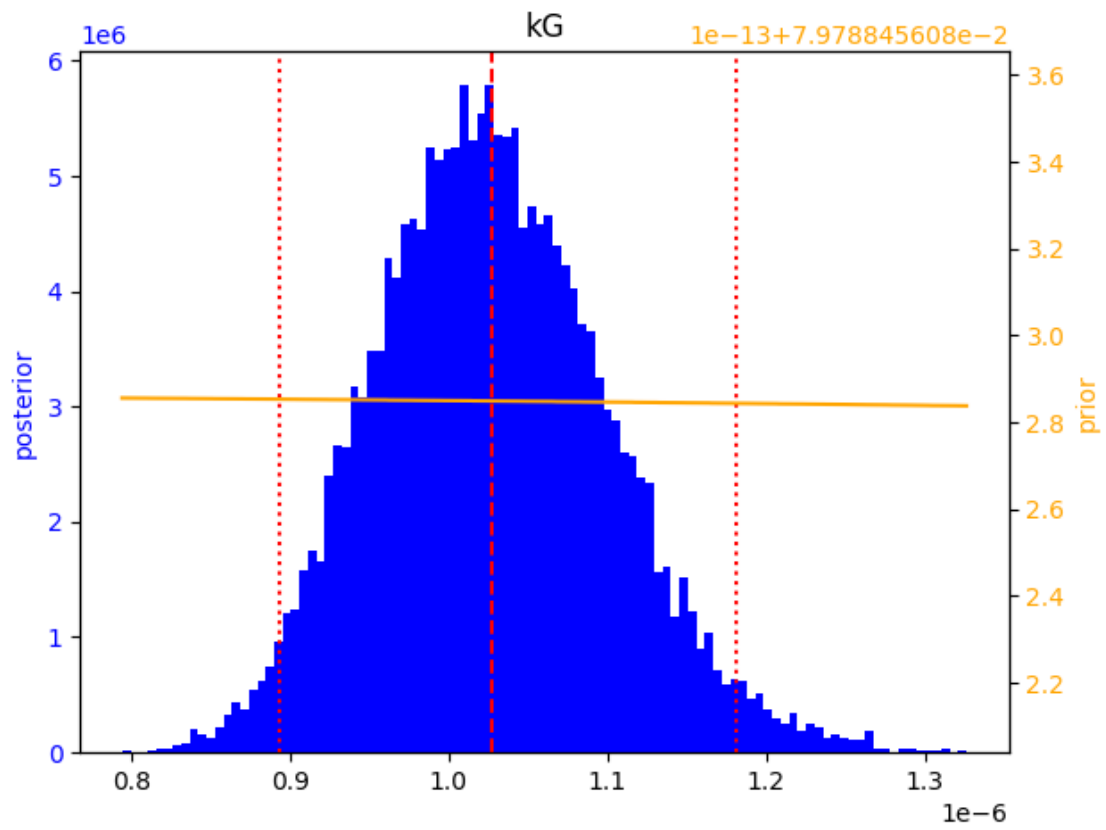


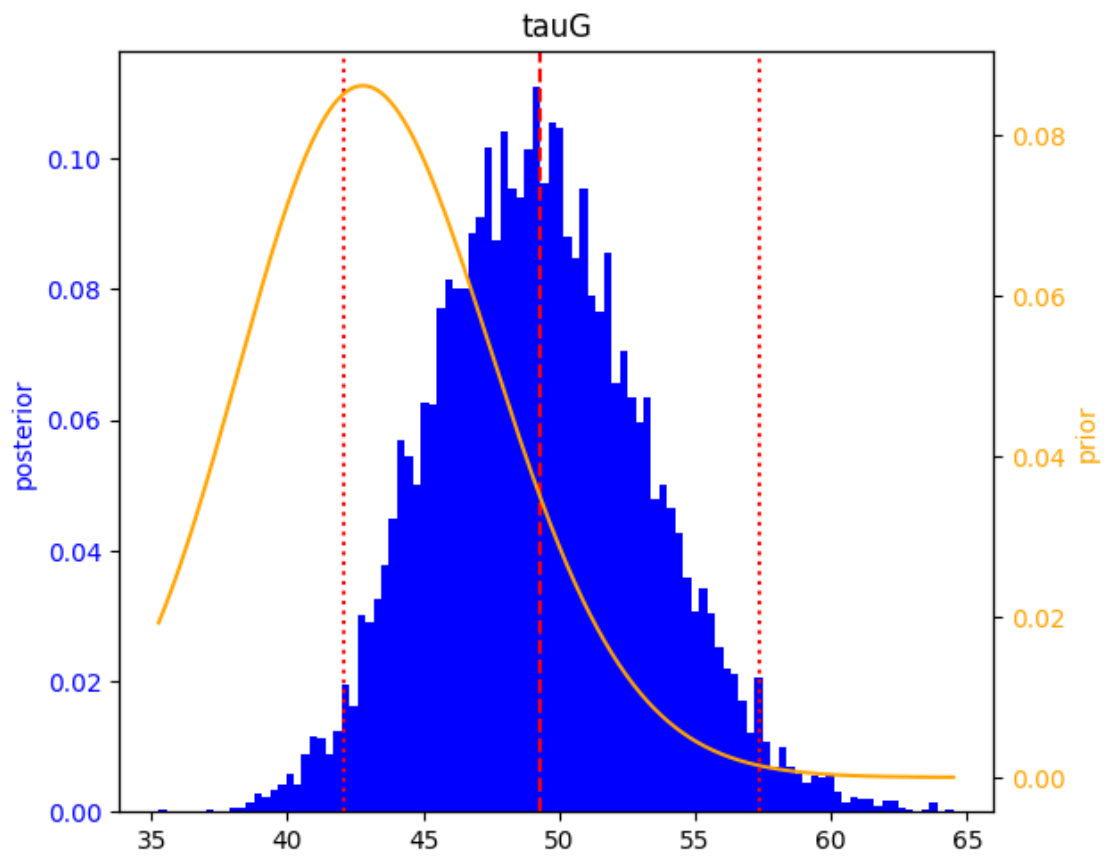


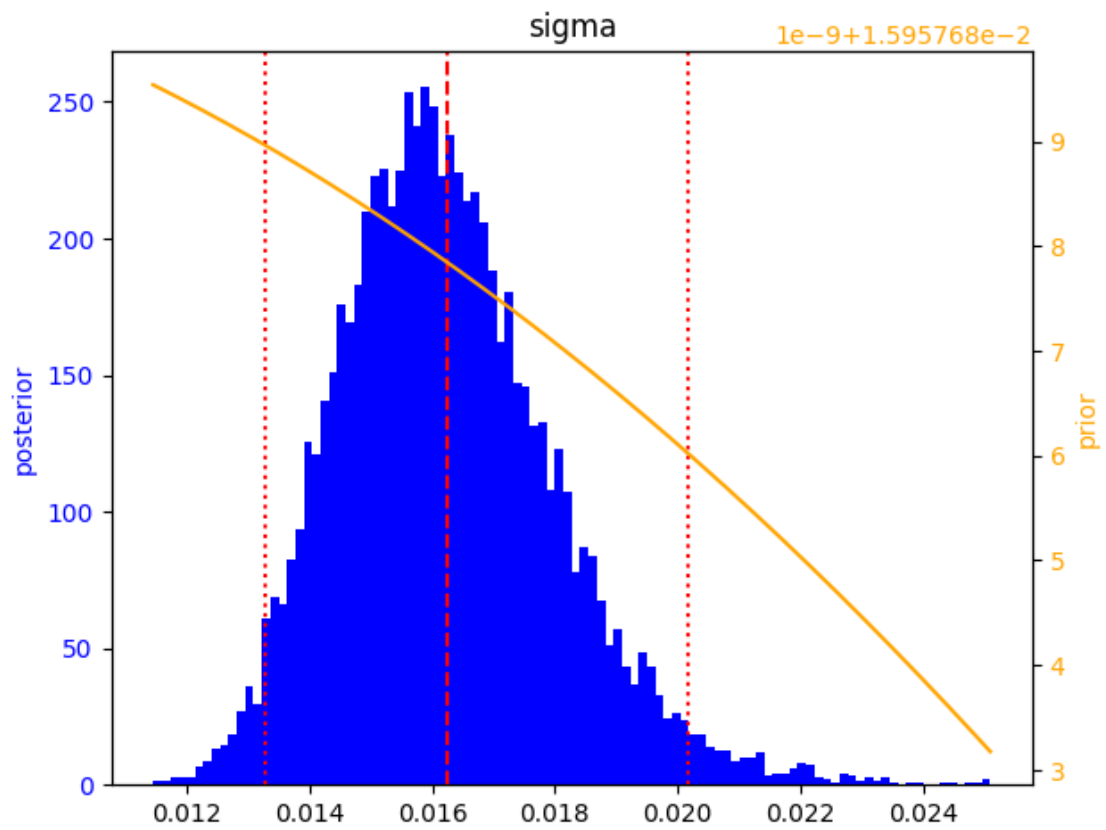




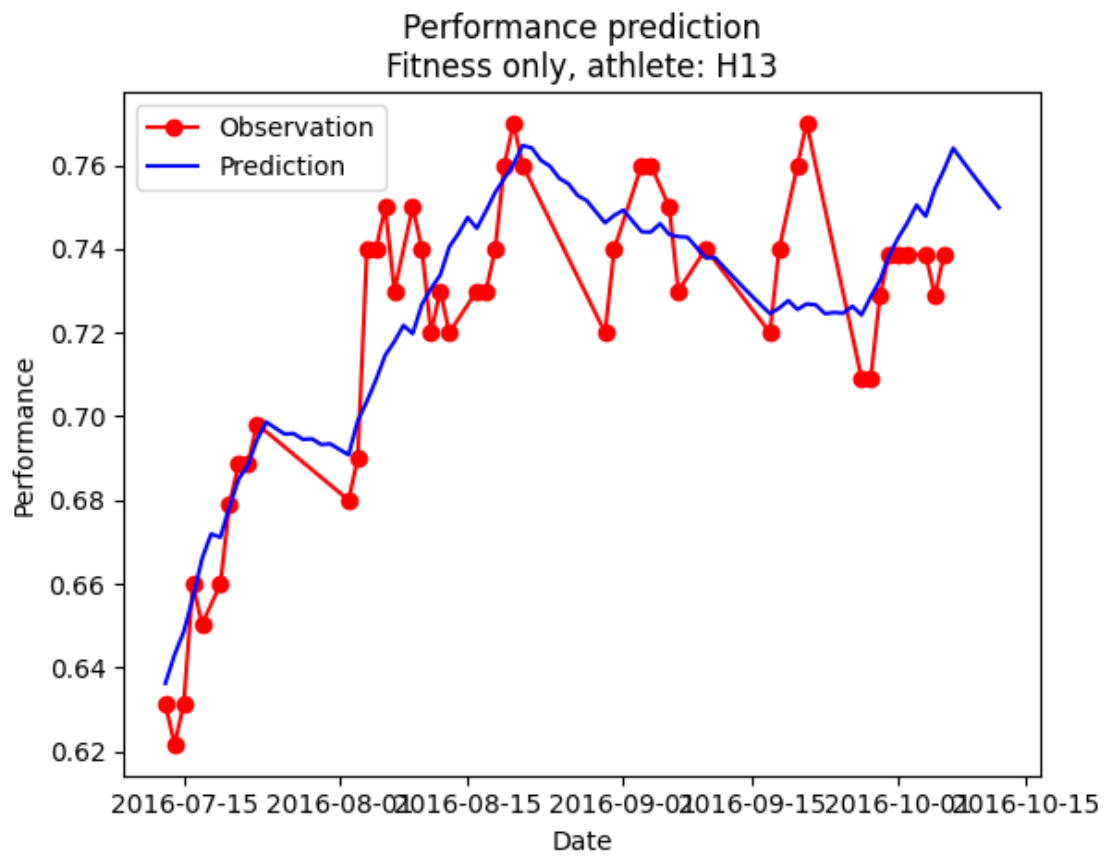


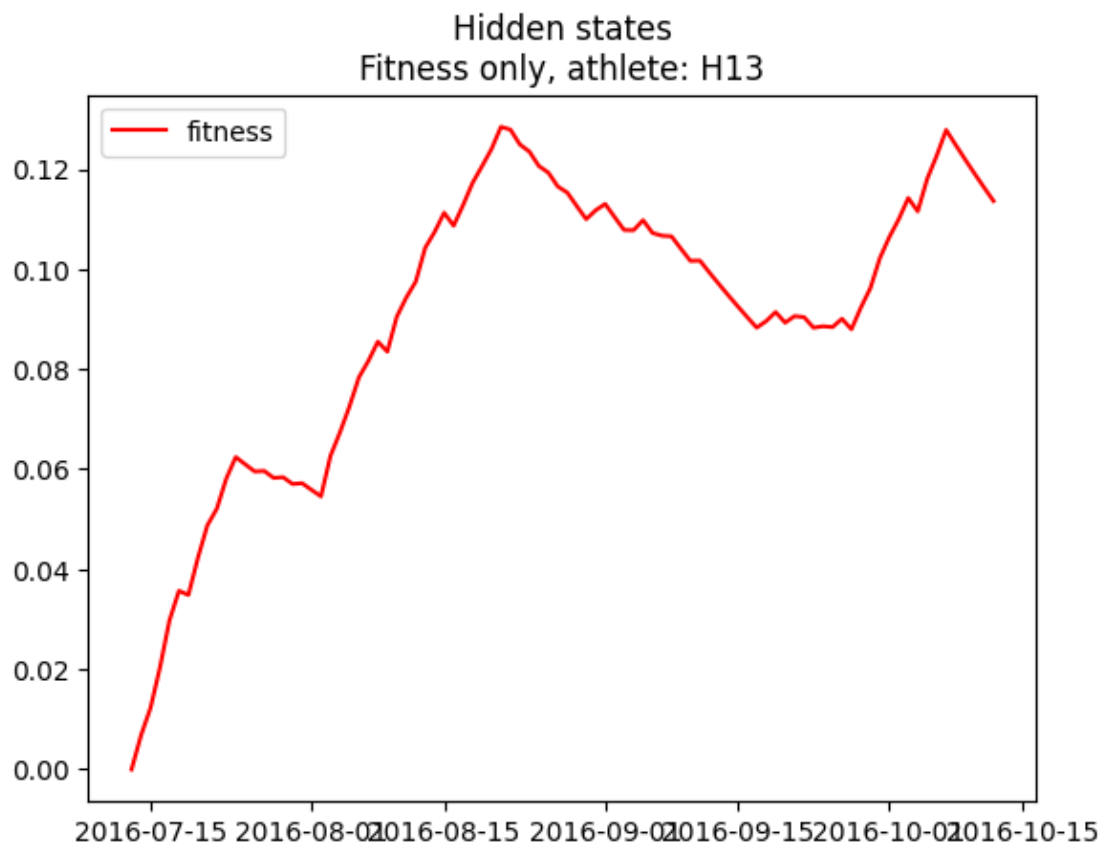


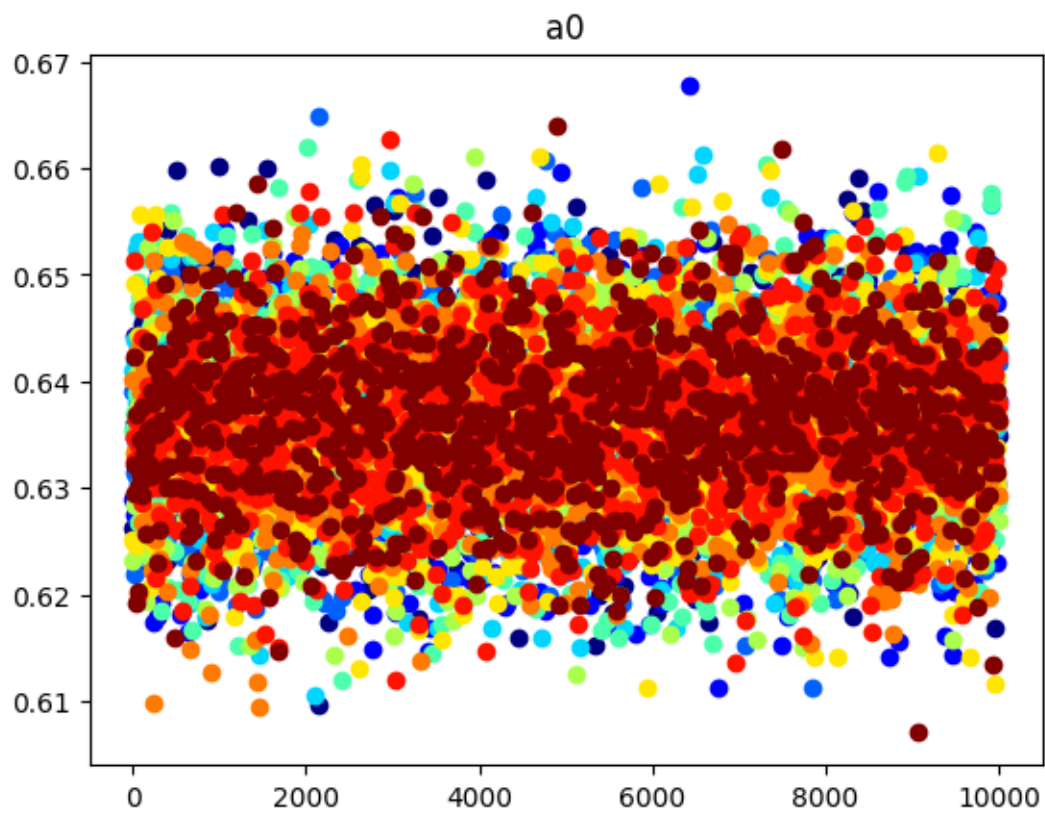


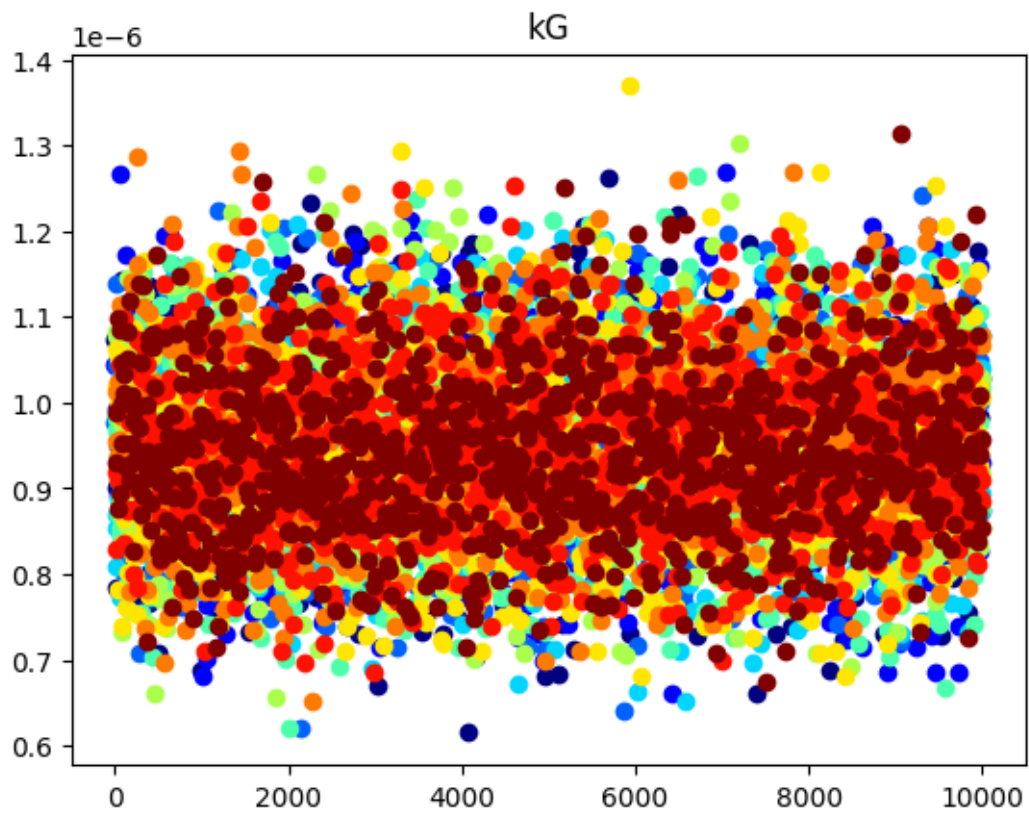


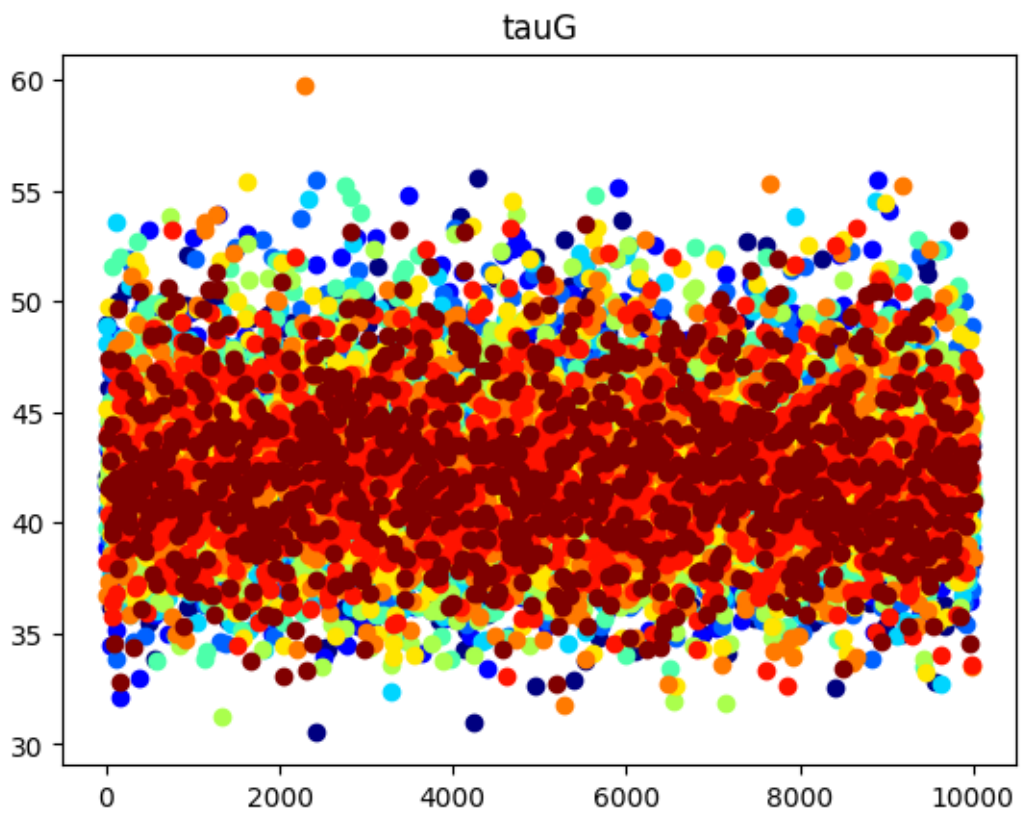


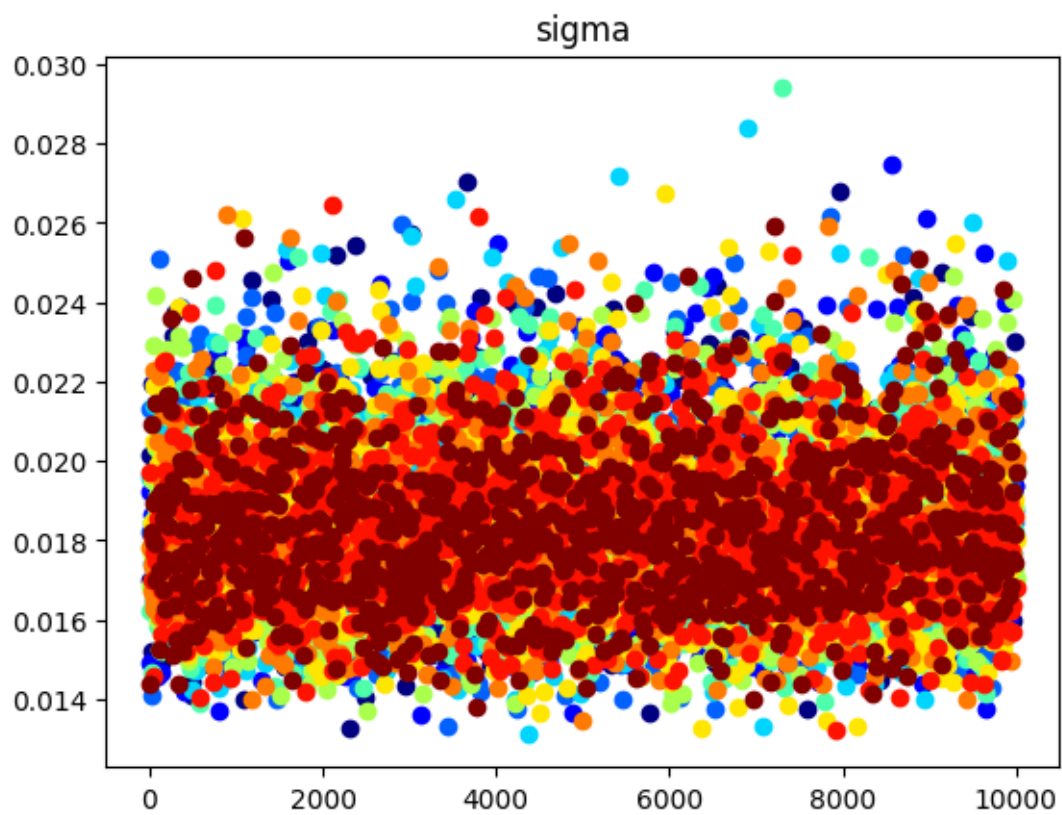


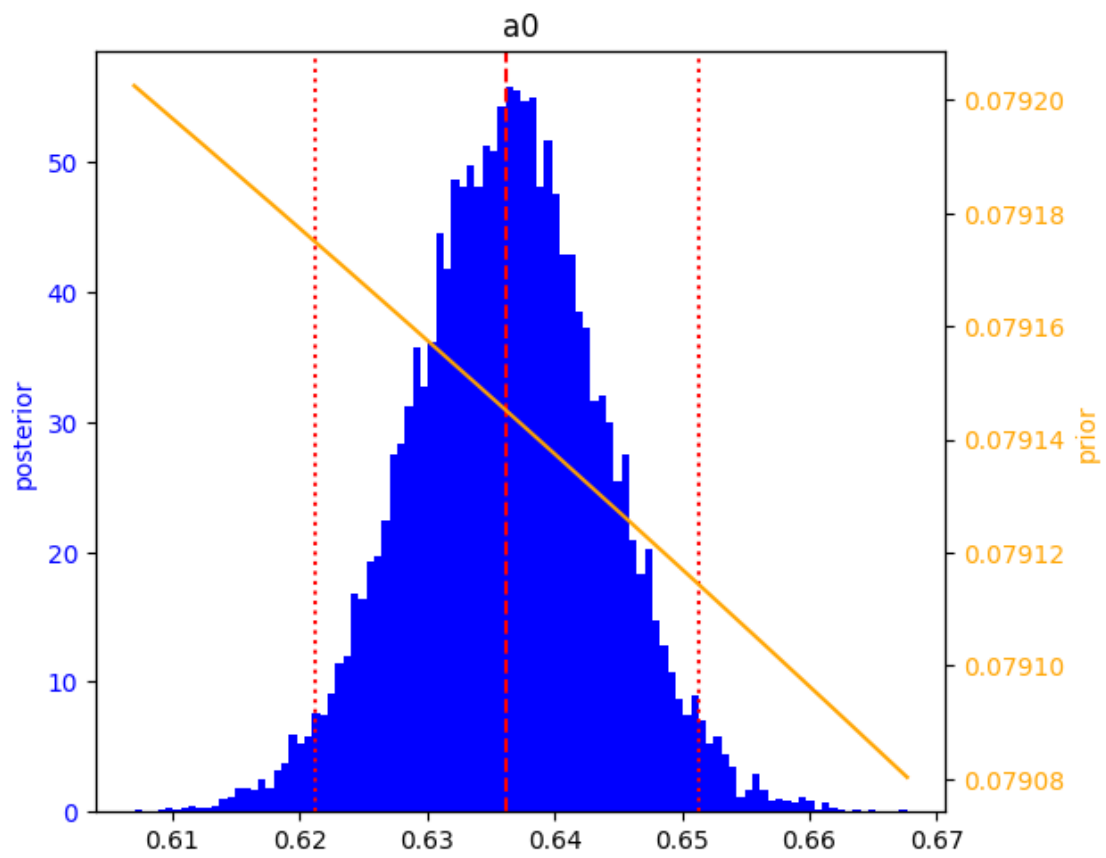


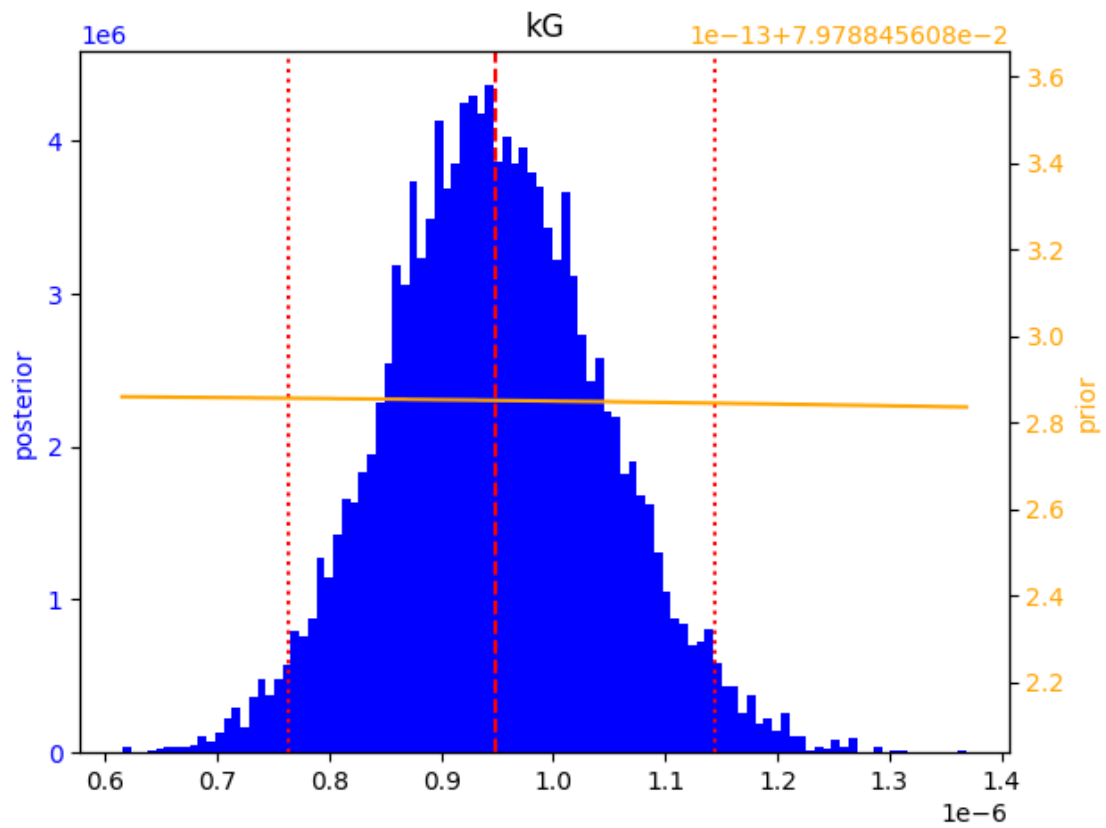




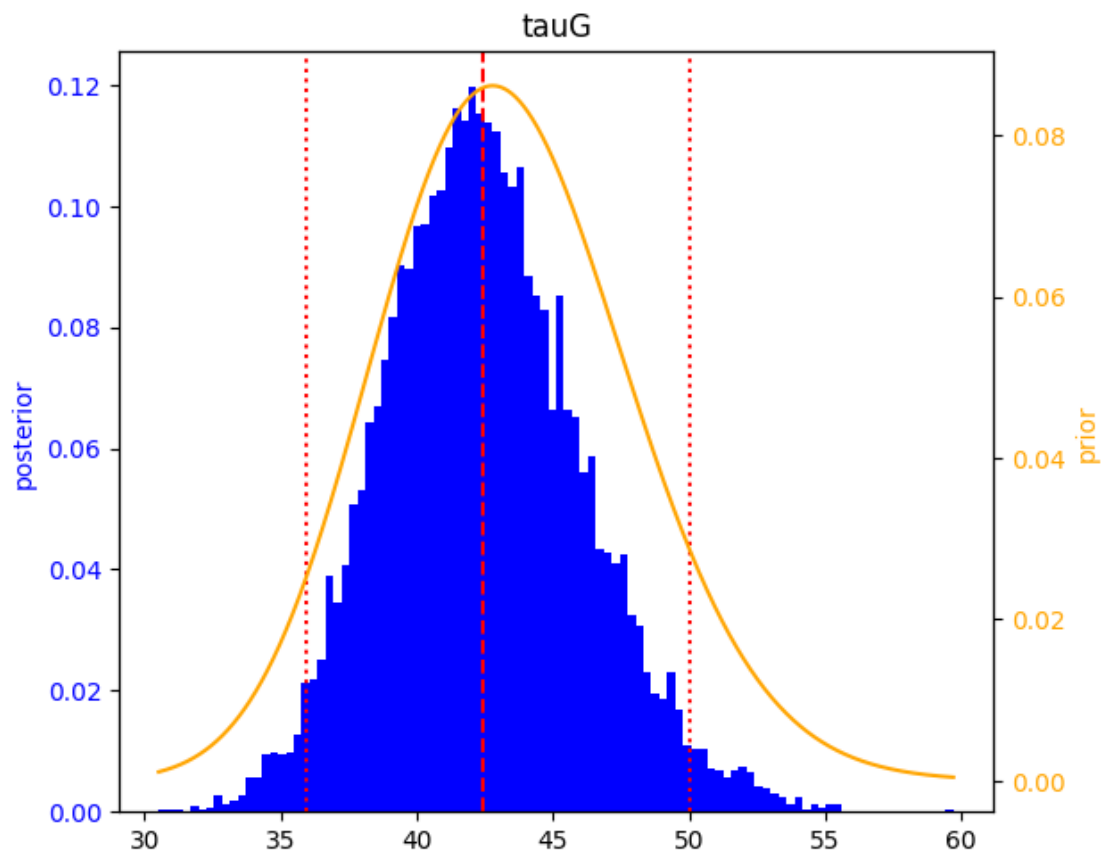


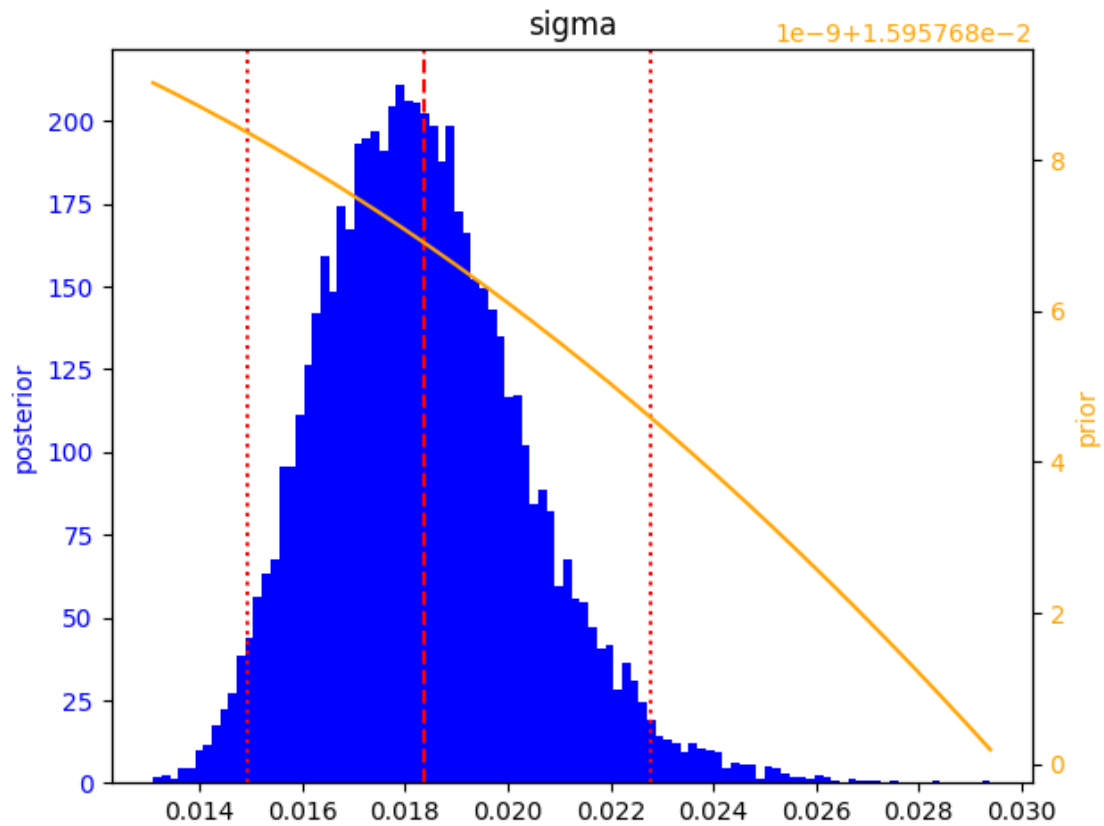












```
[5]: for ath in athletes_id:
    mods[ath]["Banister"].train.plot_perf(show = False)
    plt.title("Performance prediction\nFitness + fatigue, athlete: "+ath)
    plt.legend()
    plt.show()

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

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

