New Heat Maps

January 28, 2025

[1]:

```
import numpy as np
import os
from matplotlib import pyplot as plt
import pandas as pd
import importlib
import sys
import seaborn as sns
sys.path.insert(0, 'model_modules')

from model_classes import gLV
from community_properties import max_le_gLV
from utility_functions import community_object_to_df

%load_ext autoreload
%autoreload 2
```

[11]:

```
mu_a_values = np.arange(0.1, 1.1, 0.1)
sigma_a_values = np.arange(0.05, 0.55, 0.05)
no_lineages = 10
no_species = 50
simulation_time = 2000

output_file = "50S_interaction_summary_results.csv"
columns = ['mu_a', 'sigma_a', 'survival_fraction', 'volatility',␣
  ↪'max_le_proportion']

if not os.path.exists(output_file):
    pd.DataFrame(columns=columns).to_csv(output_file, index=False)

for mu_a in mu_a_values:
    for sigma_a in sigma_a_values:
        community_survival_means = []
        community_volatility_means = []
        community_max_le_means = []

        for community_id in range(30):
            gLV_object = gLV(
                no_species=no_species,
                growth_func='fixed',
                growth_args=None,
                interact_func='random',
                interact_args={'mu_a': mu_a, 'sigma_a': sigma_a},
                dispersal=1e-8
            )

            gLV_object.simulate_community(
                np.arange(no_lineages),
                t_end=simulation_time,
                init_cond_func='Mallmin',
                usersupplied_init_conds=None
            )

            gLV_object.calculate_community_properties()

            lineage_survival = list(gLV_object.survival_fraction.values())
            lineage_volatility = list(gLV_object.volatility.values())

            max_lyapunovs = [
                max_le_gLV(
                    gLV_object,
                    T=1000,
                    initial_conditions=simulation.y[:, -1],
                    extinction_threshold=1e-3,
                    dt=20,
                    separation=1e-3
                )
                for simulation in gLV_object.ODE_sols.values()
            ]

            community_survival_means.append(np.mean(lineage_survival))
            community_volatility_means.append(np.mean(lineage_volatility))
            community_max_le_means.append(np.mean(max_lyapunovs))

        survival_fraction_mean = np.mean(community_survival_means)
        volatility_mean = np.mean(community_volatility_means)
        max_le_proportion = np.sum(np.array(community_max_le_means) > 0.0025) /␣
↪30

        row = [
            round(mu_a, 2),
            round(sigma_a, 2),
            round(survival_fraction_mean, 2),
            round(volatility_mean, 2),
            round(max_le_proportion, 2)
        ]

        pd.DataFrame([row], columns=columns).to_csv(output_file, mode='a', header=False, index=False)

        print(f"Processed mu_a={mu_a}, sigma_a={sigma_a}")
```

```
Processed mu_a=0.1, sigma_a=0.05
Processed mu_a=0.1, sigma_a=0.1
Processed mu_a=0.1, sigma_a=0.15000000000000002
Processed mu_a=0.1, sigma_a=0.2
Processed mu_a=0.1, sigma_a=0.25
Processed mu_a=0.1, sigma_a=0.3
Processed mu_a=0.1, sigma_a=0.35000000000000003
Processed mu_a=0.1, sigma_a=0.4
Processed mu_a=0.1, sigma_a=0.45
Processed mu_a=0.1, sigma_a=0.5
Processed mu_a=0.2, sigma_a=0.05
Processed mu_a=0.2, sigma_a=0.1
Processed mu_a=0.2, sigma_a=0.15000000000000002
Processed mu_a=0.2, sigma_a=0.2
Processed mu_a=0.2, sigma_a=0.25
Processed mu_a=0.2, sigma_a=0.3
Processed mu_a=0.2, sigma_a=0.35000000000000003
Processed mu_a=0.2, sigma_a=0.4
Processed mu_a=0.2, sigma_a=0.45
Processed mu_a=0.2, sigma_a=0.5
Processed mu_a=0.30000000000000004, sigma_a=0.05
Processed mu_a=0.30000000000000004, sigma_a=0.1
Processed mu_a=0.30000000000000004, sigma_a=0.15000000000000002
Processed mu_a=0.30000000000000004, sigma_a=0.2
Processed mu_a=0.30000000000000004, sigma_a=0.25
Processed mu_a=0.30000000000000004, sigma_a=0.3
Processed mu_a=0.30000000000000004, sigma_a=0.35000000000000003
Processed mu_a=0.30000000000000004, sigma_a=0.4
Processed mu_a=0.30000000000000004, sigma_a=0.45
Processed mu_a=0.30000000000000004, sigma_a=0.5
Processed mu_a=0.4, sigma_a=0.05
Processed mu_a=0.4, sigma_a=0.1
Processed mu_a=0.4, sigma_a=0.15000000000000002
Processed mu_a=0.4, sigma_a=0.2
Processed mu_a=0.4, sigma_a=0.25
Processed mu_a=0.4, sigma_a=0.3
Processed mu_a=0.4, sigma_a=0.35000000000000003
Processed mu_a=0.4, sigma_a=0.4
Processed mu_a=0.4, sigma_a=0.45
Processed mu_a=0.4, sigma_a=0.5
Processed mu_a=0.5, sigma_a=0.05
Processed mu_a=0.5, sigma_a=0.1
Processed mu_a=0.5, sigma_a=0.15000000000000002
Processed mu_a=0.5, sigma_a=0.2
Processed mu_a=0.5, sigma_a=0.25
Processed mu_a=0.5, sigma_a=0.3
Processed mu_a=0.5, sigma_a=0.35000000000000003
Processed mu_a=0.5, sigma_a=0.4
Processed mu_a=0.5, sigma_a=0.45
Processed mu_a=0.5, sigma_a=0.5
Processed mu_a=0.6, sigma_a=0.05
Processed mu_a=0.6, sigma_a=0.1
Processed mu_a=0.6, sigma_a=0.15000000000000002
Processed mu_a=0.6, sigma_a=0.2
Processed mu_a=0.6, sigma_a=0.25
Processed mu_a=0.6, sigma_a=0.3
Processed mu_a=0.6, sigma_a=0.35000000000000003
Processed mu_a=0.6, sigma_a=0.4
Processed mu_a=0.6, sigma_a=0.45
Processed mu_a=0.6, sigma_a=0.5
Processed mu_a=0.7000000000000001, sigma_a=0.05
Processed mu_a=0.7000000000000001, sigma_a=0.1
Processed mu_a=0.7000000000000001, sigma_a=0.15000000000000002
Processed mu_a=0.7000000000000001, sigma_a=0.2
Processed mu_a=0.7000000000000001, sigma_a=0.25
Processed mu_a=0.7000000000000001, sigma_a=0.3
Processed mu_a=0.7000000000000001, sigma_a=0.35000000000000003
Processed mu_a=0.7000000000000001, sigma_a=0.4
Processed mu_a=0.7000000000000001, sigma_a=0.45
Processed mu_a=0.7000000000000001, sigma_a=0.5
Processed mu_a=0.8, sigma_a=0.05
Processed mu_a=0.8, sigma_a=0.1
Processed mu_a=0.8, sigma_a=0.15000000000000002
Processed mu_a=0.8, sigma_a=0.2
Processed mu_a=0.8, sigma_a=0.25
Processed mu_a=0.8, sigma_a=0.3
Processed mu_a=0.8, sigma_a=0.35000000000000003
Processed mu_a=0.8, sigma_a=0.4
Processed mu_a=0.8, sigma_a=0.45
Processed mu_a=0.8, sigma_a=0.5
Processed mu_a=0.9, sigma_a=0.05
Processed mu_a=0.9, sigma_a=0.1
Processed mu_a=0.9, sigma_a=0.15000000000000002
Processed mu_a=0.9, sigma_a=0.2
Processed mu_a=0.9, sigma_a=0.25
Processed mu_a=0.9, sigma_a=0.3
Processed mu_a=0.9, sigma_a=0.35000000000000003
Processed mu_a=0.9, sigma_a=0.4
Processed mu_a=0.9, sigma_a=0.45
Processed mu_a=0.9, sigma_a=0.5
Processed mu_a=1.0, sigma_a=0.05
Processed mu_a=1.0, sigma_a=0.1
Processed mu_a=1.0, sigma_a=0.15000000000000002
Processed mu_a=1.0, sigma_a=0.2
Processed mu_a=1.0, sigma_a=0.25
Processed mu_a=1.0, sigma_a=0.3
Processed mu_a=1.0, sigma_a=0.35000000000000003
Processed mu_a=1.0, sigma_a=0.4
Processed mu_a=1.0, sigma_a=0.45
Processed mu_a=1.0, sigma_a=0.5
```

[12]:

```
mu_a_values = np.arange(0.1, 1.1, 0.1)
sigma_a_values = np.arange(0.05, 0.55, 0.05)
no_lineages = 10
no_species = 100
simulation_time = 2000

output_file = "100S_interaction_summary_results.csv"
columns = ['mu_a', 'sigma_a', 'survival_fraction', 'volatility',␣
 ↪'max_le_proportion']

if not os.path.exists(output_file):
    pd.DataFrame(columns=columns).to_csv(output_file, index=False)

for mu_a in mu_a_values:
    for sigma_a in sigma_a_values:
        community_survival_means = []
        community_volatility_means = []
        community_max_le_means = []

        for community_id in range(30):
            gLV_object = gLV(
                no_species=no_species,
                growth_func='fixed',
                growth_args=None,
                interact_func='random',
                interact_args={'mu_a': mu_a, 'sigma_a': sigma_a},
                dispersal=1e-8
            )

            gLV_object.simulate_community(
                np.arange(no_lineages),
                t_end=simulation_time,
                init_cond_func='Mallmin',
                usersupplied_init_conds=None
            )
            gLV_object.calculate_community_properties()

            lineage_survival = list(gLV_object.survival_fraction.values())
            lineage_volatility = list(gLV_object.volatility.values())

            max_lyapunovs = [
                max_le_gLV(
                    gLV_object,
                    T=1000,
                    initial_conditions=simulation.y[:, -1],
                    extinction_threshold=1e-3,
                    dt=20,
                    separation=1e-3
                )
                for simulation in gLV_object.ODE_sols.values()
            ]

            community_survival_means.append(np.mean(lineage_survival))
            community_volatility_means.append(np.mean(lineage_volatility))
            community_max_le_means.append(np.mean(max_lyapunovs))

        survival_fraction_mean = np.mean(community_survival_means)
        volatility_mean = np.mean(community_volatility_means)
        max_le_proportion = np.sum(np.array(community_max_le_means) > 0.0025) /␣
↪30

        row = [
            round(mu_a, 2),
            round(sigma_a, 2),
            round(survival_fraction_mean, 2),
            round(volatility_mean, 2),
            round(max_le_proportion, 2)
        ]

        pd.DataFrame([row], columns=columns).to_csv(output_file, mode='a',␣
↪header=False, index=False)

        print(f"Processed mu_a={mu_a}, sigma_a={sigma_a}")
```

```
Processed mu_a=0.1, sigma_a=0.05
Processed mu_a=0.1, sigma_a=0.1
Processed mu_a=0.1, sigma_a=0.15000000000000002
Processed mu_a=0.1, sigma_a=0.2
Processed mu_a=0.1, sigma_a=0.25
Processed mu_a=0.1, sigma_a=0.3
Processed mu_a=0.1, sigma_a=0.35000000000000003
Processed mu_a=0.1, sigma_a=0.4
Processed mu_a=0.1, sigma_a=0.45
Processed mu_a=0.1, sigma_a=0.5
Processed mu_a=0.2, sigma_a=0.05
Processed mu_a=0.2, sigma_a=0.1
Processed mu_a=0.2, sigma_a=0.15000000000000002
Processed mu_a=0.2, sigma_a=0.2
Processed mu_a=0.2, sigma_a=0.25
Processed mu_a=0.2, sigma_a=0.3
Processed mu_a=0.2, sigma_a=0.35000000000000003
Processed mu_a=0.2, sigma_a=0.4
Processed mu_a=0.2, sigma_a=0.45
Processed mu_a=0.2, sigma_a=0.5
Processed mu_a=0.30000000000000004, sigma_a=0.05
Processed mu_a=0.30000000000000004, sigma_a=0.1
Processed mu_a=0.30000000000000004, sigma_a=0.15000000000000002
Processed mu_a=0.30000000000000004, sigma_a=0.2
Processed mu_a=0.30000000000000004, sigma_a=0.25
Processed mu_a=0.30000000000000004, sigma_a=0.3
Processed mu_a=0.30000000000000004, sigma_a=0.35000000000000003
Processed mu_a=0.30000000000000004, sigma_a=0.4
Processed mu_a=0.30000000000000004, sigma_a=0.45
Processed mu_a=0.30000000000000004, sigma_a=0.5
Processed mu_a=0.4, sigma_a=0.05
Processed mu_a=0.4, sigma_a=0.1
Processed mu_a=0.4, sigma_a=0.15000000000000002
Processed mu_a=0.4, sigma_a=0.2
Processed mu_a=0.4, sigma_a=0.25
Processed mu_a=0.4, sigma_a=0.3
Processed mu_a=0.4, sigma_a=0.35000000000000003
Processed mu_a=0.4, sigma_a=0.4
Processed mu_a=0.4, sigma_a=0.45
Processed mu_a=0.4, sigma_a=0.5
Processed mu_a=0.5, sigma_a=0.05
Processed mu_a=0.5, sigma_a=0.1
Processed mu_a=0.5, sigma_a=0.15000000000000002
Processed mu_a=0.5, sigma_a=0.2
Processed mu_a=0.5, sigma_a=0.25
Processed mu_a=0.5, sigma_a=0.3
Processed mu_a=0.5, sigma_a=0.35000000000000003
Processed mu_a=0.5, sigma_a=0.4
Processed mu_a=0.5, sigma_a=0.45
Processed mu_a=0.5, sigma_a=0.5
Processed mu_a=0.6, sigma_a=0.05
Processed mu_a=0.6, sigma_a=0.1
Processed mu_a=0.6, sigma_a=0.15000000000000002
Processed mu_a=0.6, sigma_a=0.2
Processed mu_a=0.6, sigma_a=0.25
Processed mu_a=0.6, sigma_a=0.3
Processed mu_a=0.6, sigma_a=0.35000000000000003
Processed mu_a=0.6, sigma_a=0.4
Processed mu_a=0.6, sigma_a=0.45
Processed mu_a=0.6, sigma_a=0.5
Processed mu_a=0.7000000000000001, sigma_a=0.05
Processed mu_a=0.7000000000000001, sigma_a=0.1
Processed mu_a=0.7000000000000001, sigma_a=0.15000000000000002
Processed mu_a=0.7000000000000001, sigma_a=0.2
Processed mu_a=0.7000000000000001, sigma_a=0.25
Processed mu_a=0.7000000000000001, sigma_a=0.3
Processed mu_a=0.7000000000000001, sigma_a=0.35000000000000003
Processed mu_a=0.7000000000000001, sigma_a=0.4
Processed mu_a=0.7000000000000001, sigma_a=0.45
Processed mu_a=0.7000000000000001, sigma_a=0.5
Processed mu_a=0.8, sigma_a=0.05
Processed mu_a=0.8, sigma_a=0.1
Processed mu_a=0.8, sigma_a=0.15000000000000002
Processed mu_a=0.8, sigma_a=0.2
Processed mu_a=0.8, sigma_a=0.25
Processed mu_a=0.8, sigma_a=0.3
Processed mu_a=0.8, sigma_a=0.35000000000000003
Processed mu_a=0.8, sigma_a=0.4
Processed mu_a=0.8, sigma_a=0.45
Processed mu_a=0.8, sigma_a=0.5
Processed mu_a=0.9, sigma_a=0.05
Processed mu_a=0.9, sigma_a=0.1
Processed mu_a=0.9, sigma_a=0.15000000000000002
Processed mu_a=0.9, sigma_a=0.2
Processed mu_a=0.9, sigma_a=0.25
Processed mu_a=0.9, sigma_a=0.3
Processed mu_a=0.9, sigma_a=0.35000000000000003
Processed mu_a=0.9, sigma_a=0.4
Processed mu_a=0.9, sigma_a=0.45
Processed mu_a=0.9, sigma_a=0.5
Processed mu_a=1.0, sigma_a=0.05
Processed mu_a=1.0, sigma_a=0.1
Processed mu_a=1.0, sigma_a=0.15000000000000002
Processed mu_a=1.0, sigma_a=0.2
Processed mu_a=1.0, sigma_a=0.25
Processed mu_a=1.0, sigma_a=0.3
Processed mu_a=1.0, sigma_a=0.35000000000000003
Processed mu_a=1.0, sigma_a=0.4
Processed mu_a=1.0, sigma_a=0.45
Processed mu_a=1.0, sigma_a=0.5
```

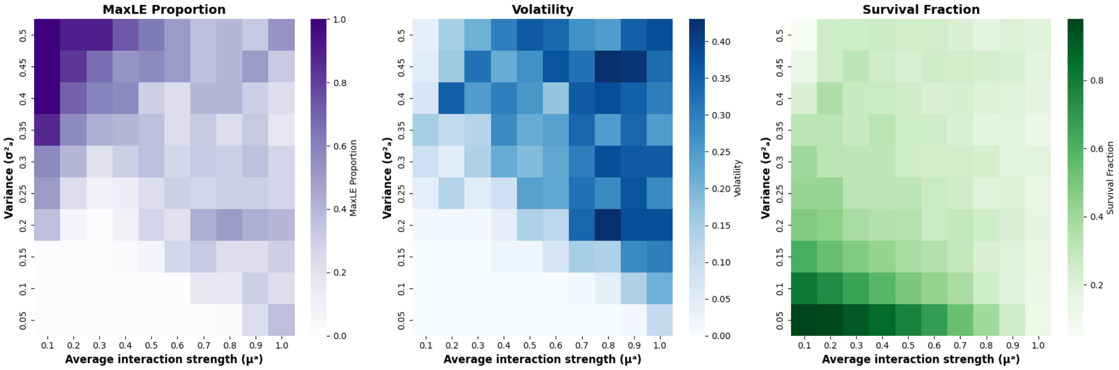
[13]:

```
mu_a_values = np.arange(0.1, 1.1, 0.1)
sigma_a_values = np.arange(0.05, 0.55, 0.05)
no_lineages = 10
no_species = 150
simulation_time = 2000

output_file = "150_interaction_summary_result.csv"
columns = ['mu_a', 'sigma_a', 'survival_fraction', 'volatility',␣
 ↪'max_le_proportion']

if not os.path.exists(output_file):
    pd.DataFrame(columns=columns).to_csv(output_file, index=False)

for mu_a in mu_a_values:
    for sigma_a in sigma_a_values:
        community_survival_means = []
        community_volatility_means = []
        community_max_le_means = []

        for community_id in range(30):
            gLV_object = gLV(
                no_species=no_species,
                growth_func='fixed',
                growth_args=None,
                interact_func='random',
                interact_args={'mu_a': mu_a, 'sigma_a': sigma_a},
                dispersal=1e-8
            )

            gLV_object.simulate_community(
                np.arange(no_lineages),
                t_end=simulation_time,
                init_cond_func='Mallmin',
                usersupplied_init_conds=None
            )

            gLV_object.calculate_community_properties()

            lineage_survival = list(gLV_object.survival_fraction.values())
            lineage_volatility = list(gLV_object.volatility.values())

            max_lyapunovs = [
                max_le_gLV(
                    gLV_object,
                    T=1000,
                    initial_conditions=simulation.y[:, -1],
                    extinction_threshold=1e-3,
                    dt=20,
                    separation=1e-3
                )
                for simulation in gLV_object.ODE_sols.values()
            ]

            community_survival_means.append(np.mean(lineage_survival))
            community_volatility_means.append(np.mean(lineage_volatility))
            community_max_le_means.append(np.mean(max_lyapunovs))

        survival_fraction_mean = np.mean(community_survival_means)
        volatility_mean = np.mean(community_volatility_means)
        max_le_proportion = np.sum(np.array(community_max_le_means) > 0.0025) /␣
↪30

        row = [
            round(mu_a, 2),
            round(sigma_a, 2),
            round(survival_fraction_mean, 2),
            round(volatility_mean, 2),
            round(max_le_proportion, 2)
        ]

        pd.DataFrame([row], columns=columns).to_csv(output_file, mode='a',␣
↪header=False, index=False)

        print(f"Processed mu_a={mu_a}, sigma_a={sigma_a}")
```

```
Processed mu_a=0.1, sigma_a=0.05
Processed mu_a=0.1, sigma_a=0.1
Processed mu_a=0.1, sigma_a=0.15000000000000002
Processed mu_a=0.1, sigma_a=0.2
Processed mu_a=0.1, sigma_a=0.25
Processed mu_a=0.1, sigma_a=0.3
Processed mu_a=0.1, sigma_a=0.35000000000000003
Processed mu_a=0.1, sigma_a=0.4
Processed mu_a=0.1, sigma_a=0.45
Processed mu_a=0.1, sigma_a=0.5
Processed mu_a=0.2, sigma_a=0.05
Processed mu_a=0.2, sigma_a=0.1
Processed mu_a=0.2, sigma_a=0.15000000000000002
Processed mu_a=0.2, sigma_a=0.2
Processed mu_a=0.2, sigma_a=0.25
Processed mu_a=0.2, sigma_a=0.3
Processed mu_a=0.2, sigma_a=0.35000000000000003
Processed mu_a=0.2, sigma_a=0.4
Processed mu_a=0.2, sigma_a=0.45
Processed mu_a=0.2, sigma_a=0.5
Processed mu_a=0.30000000000000004, sigma_a=0.05
Processed mu_a=0.30000000000000004, sigma_a=0.1
Processed mu_a=0.30000000000000004, sigma_a=0.15000000000000002
Processed mu_a=0.30000000000000004, sigma_a=0.2
Processed mu_a=0.30000000000000004, sigma_a=0.25
Processed mu_a=0.30000000000000004, sigma_a=0.3
Processed mu_a=0.30000000000000004, sigma_a=0.35000000000000003
Processed mu_a=0.30000000000000004, sigma_a=0.4
Processed mu_a=0.30000000000000004, sigma_a=0.45
Processed mu_a=0.30000000000000004, sigma_a=0.5
Processed mu_a=0.4, sigma_a=0.05
Processed mu_a=0.4, sigma_a=0.1
Processed mu_a=0.4, sigma_a=0.15000000000000002
Processed mu_a=0.4, sigma_a=0.2
Processed mu_a=0.4, sigma_a=0.25
Processed mu_a=0.4, sigma_a=0.3
Processed mu_a=0.4, sigma_a=0.35000000000000003
Processed mu_a=0.4, sigma_a=0.4
Processed mu_a=0.4, sigma_a=0.45
Processed mu_a=0.4, sigma_a=0.5
Processed mu_a=0.5, sigma_a=0.05
Processed mu_a=0.5, sigma_a=0.1
Processed mu_a=0.5, sigma_a=0.15000000000000002
Processed mu_a=0.5, sigma_a=0.2
Processed mu_a=0.5, sigma_a=0.25
Processed mu_a=0.5, sigma_a=0.3
Processed mu_a=0.5, sigma_a=0.35000000000000003
Processed mu_a=0.5, sigma_a=0.4
Processed mu_a=0.5, sigma_a=0.45
Processed mu_a=0.5, sigma_a=0.5
Processed mu_a=0.6, sigma_a=0.05
Processed mu_a=0.6, sigma_a=0.1
Processed mu_a=0.6, sigma_a=0.15000000000000002
Processed mu_a=0.6, sigma_a=0.2
Processed mu_a=0.6, sigma_a=0.25
Processed mu_a=0.6, sigma_a=0.3
Processed mu_a=0.6, sigma_a=0.35000000000000003
Processed mu_a=0.6, sigma_a=0.4
Processed mu_a=0.6, sigma_a=0.45
Processed mu_a=0.6, sigma_a=0.5
Processed mu_a=0.7000000000000001, sigma_a=0.05
Processed mu_a=0.7000000000000001, sigma_a=0.1
Processed mu_a=0.7000000000000001, sigma_a=0.15000000000000002
Processed mu_a=0.7000000000000001, sigma_a=0.2
Processed mu_a=0.7000000000000001, sigma_a=0.25
Processed mu_a=0.7000000000000001, sigma_a=0.3
Processed mu_a=0.7000000000000001, sigma_a=0.35000000000000003
Processed mu_a=0.7000000000000001, sigma_a=0.4
Processed mu_a=0.7000000000000001, sigma_a=0.45
Processed mu_a=0.7000000000000001, sigma_a=0.5
Processed mu_a=0.8, sigma_a=0.05
Processed mu_a=0.8, sigma_a=0.1
Processed mu_a=0.8, sigma_a=0.15000000000000002
Processed mu_a=0.8, sigma_a=0.2
Processed mu_a=0.8, sigma_a=0.25
Processed mu_a=0.8, sigma_a=0.3
Processed mu_a=0.8, sigma_a=0.35000000000000003
Processed mu_a=0.8, sigma_a=0.4
Processed mu_a=0.8, sigma_a=0.45
Processed mu_a=0.8, sigma_a=0.5
Processed mu_a=0.9, sigma_a=0.05
Processed mu_a=0.9, sigma_a=0.1
Processed mu_a=0.9, sigma_a=0.15000000000000002
Processed mu_a=0.9, sigma_a=0.2
Processed mu_a=0.9, sigma_a=0.25
Processed mu_a=0.9, sigma_a=0.3
Processed mu_a=0.9, sigma_a=0.35000000000000003
Processed mu_a=0.9, sigma_a=0.4
Processed mu_a=0.9, sigma_a=0.45
Processed mu_a=0.9, sigma_a=0.5
Processed mu_a=1.0, sigma_a=0.05
Processed mu_a=1.0, sigma_a=0.1
Processed mu_a=1.0, sigma_a=0.15000000000000002
Processed mu_a=1.0, sigma_a=0.2
Processed mu_a=1.0, sigma_a=0.25
Processed mu_a=1.0, sigma_a=0.3
Processed mu_a=1.0, sigma_a=0.35000000000000003
Processed mu_a=1.0, sigma_a=0.4
Processed mu_a=1.0, sigma_a=0.45
Processed mu_a=1.0, sigma_a=0.5
```

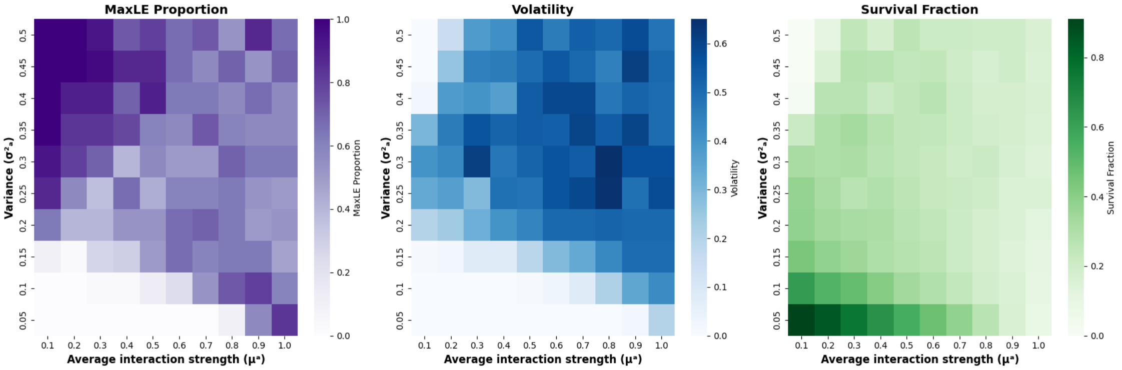
[16]:

```python
file_path = "50S_interaction_summary_results.csv"
data = pd.read_csv(file_path)

data["sigma_a"] = data["sigma_a"].astype(float)
data["mu_a"] = data["mu_a"].astype(float)

variables = ["max_le_proportion", "volatility", "survival_fraction"]
titles = ["MaxLE Proportion", "Volatility", "Survival Fraction"]
cmaps = ["Purples", "Blues", "Greens"]

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

for i, var in enumerate(variables):
    heatmap_data = data.pivot(index='sigma_a', columns='mu_a', values=var)
    heatmap_data = heatmap_data.sort_index(ascending=False)
    sns.heatmap(
        heatmap_data, ax=axes[i], annot=False, cmap=cmaps[i], cbar_kws={'label':
 ↪ titles[i]}
    )

    axes[i].set_xlabel("Average interaction strength ( )", fontsize=12,␣
 ↪weight='bold')
    axes[i].set_ylabel("Variance ( ² )", fontsize=12, weight='bold')
    axes[i].set_title(titles[i], fontsize=14, weight='bold')

plt.tight_layout()
plt.show()
```



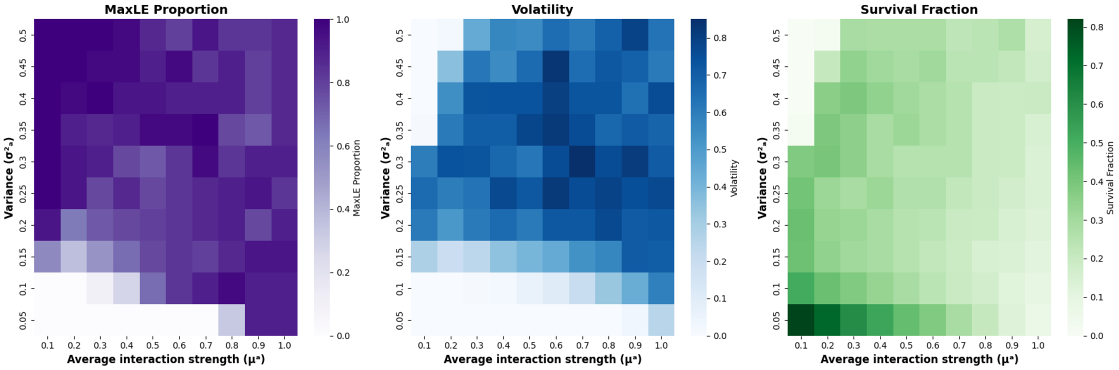
[17]:
```
file_path = "100S_interaction_summary_results.csv"
data = pd.read_csv(file_path)

data["sigma_a"] = data["sigma_a"].astype(float)
data["mu_a"] = data["mu_a"].astype(float)

variables = ["max_le_proportion", "volatility", "survival_fraction"]
titles = ["MaxLE Proportion", "Volatility", "Survival Fraction"]
cmaps = ["Purples", "Blues", "Greens"]

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

for i, var in enumerate(variables):
    heatmap_data = data.pivot(index='sigma_a', columns='mu_a', values=var)
    heatmap_data = heatmap_data.sort_index(ascending=False)
    sns.heatmap(
        heatmap_data, ax=axes[i], annot=False, cmap=cmaps[i], cbar_kws={'label':
 ↪ titles[i]}
```

```
    )

    axes[i].set_xlabel("Average interaction strength ( )", fontsize=12,␣
 ↪weight='bold')
    axes[i].set_ylabel("Variance ( ² )", fontsize=12, weight='bold')
    axes[i].set_title(titles[i], fontsize=14, weight='bold')

plt.tight_layout()
plt.show()
```



[18]:

```
file_path = "150S_interaction_summary_results.csv"
data = pd.read_csv(file_path)

data["sigma_a"] = data["sigma_a"].astype(float)
data["mu_a"] = data["mu_a"].astype(float)

variables = ["max_le_proportion", "volatility", "survival_fraction"]
titles = ["MaxLE Proportion", "Volatility", "Survival Fraction"]
cmaps = ["Purples", "Blues", "Greens"]

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

for i, var in enumerate(variables):
    heatmap_data = data.pivot(index='sigma_a', columns='mu_a', values=var)
    heatmap_data = heatmap_data.sort_index(ascending=False)
    sns.heatmap(
        heatmap_data, ax=axes[i], annot=False, cmap=cmaps[i], cbar_kws={'label':
 ↪ titles[i]}
    )

    axes[i].set_xlabel("Average interaction strength ( )", fontsize=12,␣
 ↪weight='bold')
    axes[i].set_ylabel("Variance ( ² )", fontsize=12, weight='bold')
    axes[i].set_title(titles[i], fontsize=14, weight='bold')

plt.tight_layout()
plt.show()
```


MaxLE Proportion
Volatility
Survival Fraction
Variance (σ²ₐ)
Average interaction strength (μₐ)
MaxLE Proportion
Volatility
Survival Fraction