viz_prelim_ex04-ex04R-ex06_DK_20220928

September 28, 2022

1 Set-up

1.0.1 Imports

```
[ ]: # --- Data handling and visualization ---
import pandas as pd  # Dataframe tools
from tabulate import tabulate  # Pretty printing for dataframes

import seaborn as sns  # Easier plotting tools
import matplotlib.pyplot as plt
%matplotlib inline
```

1.0.2 Globals

```
[ ]: # --- Visualization ---
rc_dict = {
  "savefig.dpi": 900,  # Saved figure dots-per-inch. 600 is "HD"
  "savefig.facecolor": "white",  # This, combined with transparent setting,␣
 ↪keeps saved figs from looking like trash on dark backgrounds
  "savefig.transparent": False,
  "figure.figsize": (14, 10),  # Default (width, height) of figure
}
plt.rcParams.update(rc_dict)
```

1.1 Data preprocessing

```
[ ]: synth_n2v = pd.read_parquet("../results/dataframes/
 ↪dataframe-merged_EMB_ex04_DK_20220928.parquet")
synth_spectral = pd.read_parquet("../results/dataframes/
 ↪dataframe-merged_EMB_ex06_DK_20220928.parquet")

idx_cols = ["N", "avgk", "mu", "gamma", "tau", "rep", "pfi", "metric"]
synth = pd.merge(synth_n2v, synth_spectral, on=idx_cols, suffixes=["_n2v",␣
 ↪"_spectral"])

df = synth.melt(id_vars=idx_cols, var_name="Performance Measure",␣
 ↪value_name="Performance")
```

```
df = df.astype({
    'N': 'int8',
    'avgk': 'int8',
    'rep': 'int8',
    'mu': 'float32',
    'gamma': 'float32',
    'tau': 'float32',
    'pfi': 'float32',
    'metric': 'category',
    'Performance Measure': 'string',
})

df.info()
```

[ ]:

```
real = pd.read_parquet("../results/dataframes/
 ↪dataframe-merged_EMB_ex04R_DK_20220928.parquet")
real.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 120 entries, 0 to 59
Data columns (total 16 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   largest_component  120 non-null    bool
 1   dimensions         120 non-null    int64
 2   walk_length        120 non-null    int64
 3   num_walks          120 non-null    int64
 4   workers            120 non-null    int64
 5   window             120 non-null    int64
 6   min_count          120 non-null    int64
 7   batch_words        120 non-null    int64
 8   system             120 non-null    object
 9   left               120 non-null    int64
 10  right              120 non-null    int64
 11  metric             120 non-null    object
 12  pfi                120 non-null    float64
 13  repetition         120 non-null    int64
 14  Accuracy           120 non-null    float64
 15  AUROC              120 non-null    float64
dtypes: bool(1), float64(3), int64(10), object(2)
memory usage: 15.1+ KB
```

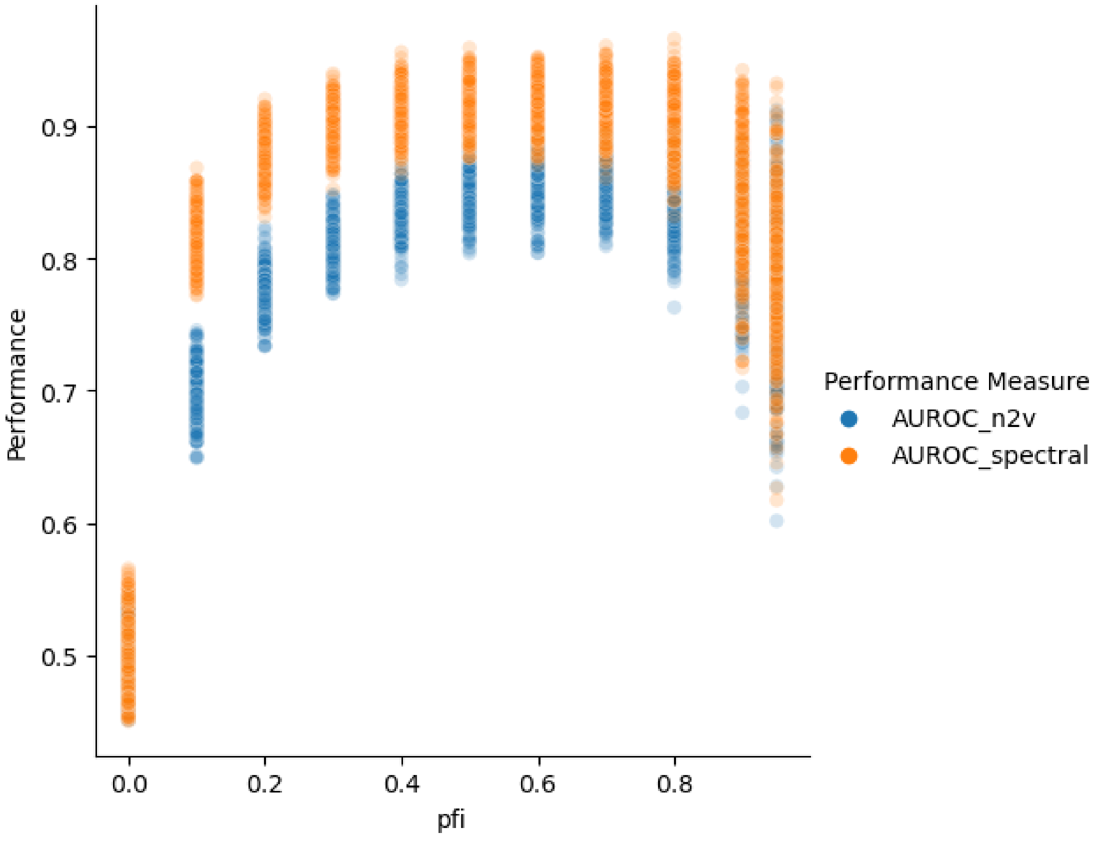
2 Figures

2.1 Synthetic

2.1.1 AUROC

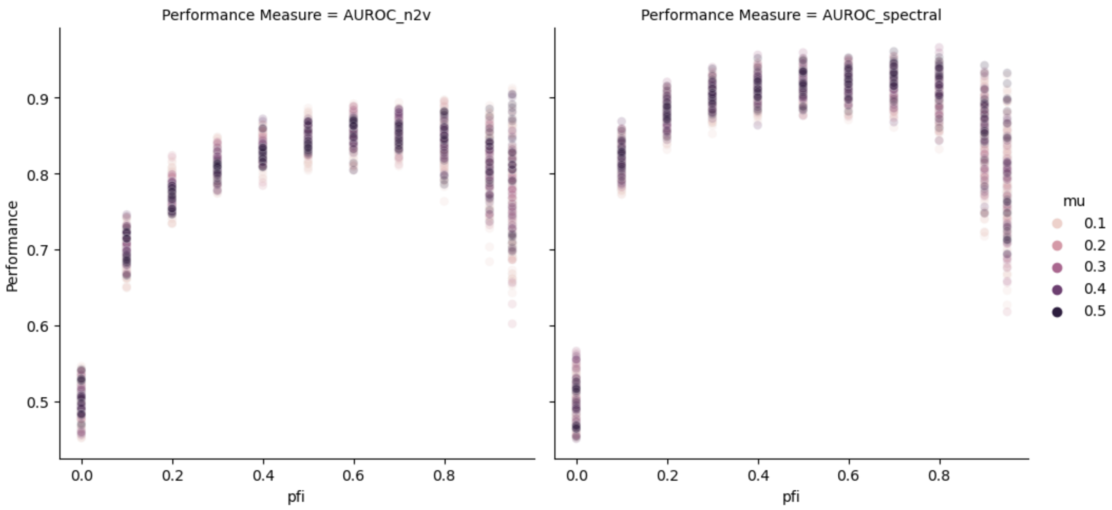
```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[~df["Performance Measure"].isin(["Accuracy_n2v",␣
      ↪"Accuracy_spectral"])], kind="scatter",
         x="pfi", y="Performance",
         hue="Performance Measure",
         alpha=0.2
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83c7332470>

<Figure size 1400x1000 with 0 Axes>
```



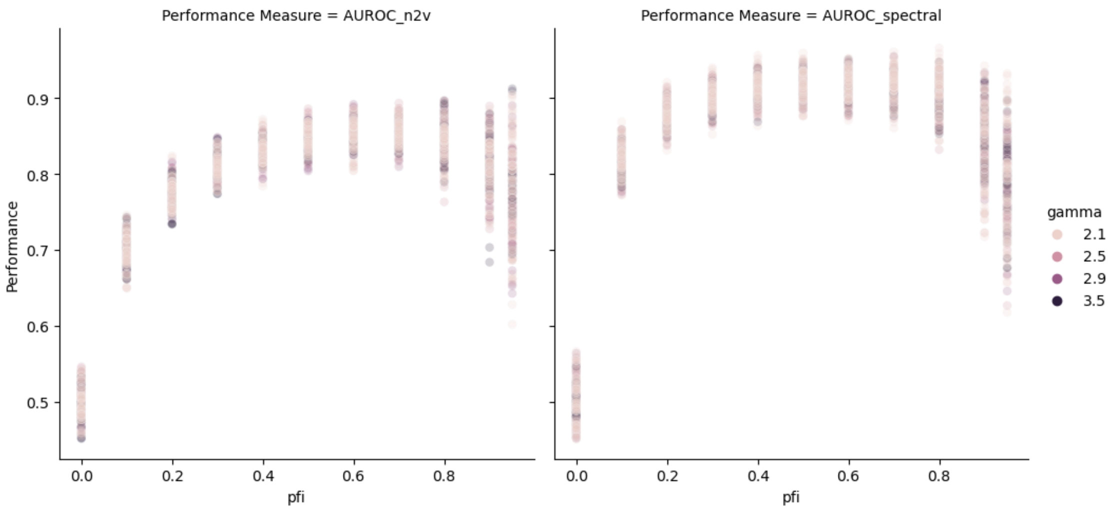
```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[~df["Performance Measure"].isin(["Accuracy_n2v",␣
     ↪"Accuracy_spectral"])], kind="scatter",
         x="pfi", y="Performance",
         col="Performance Measure",
         hue="mu",
         alpha=0.2
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83df53ebf0>

     <Figure size 1400x1000 with 0 Axes>
```



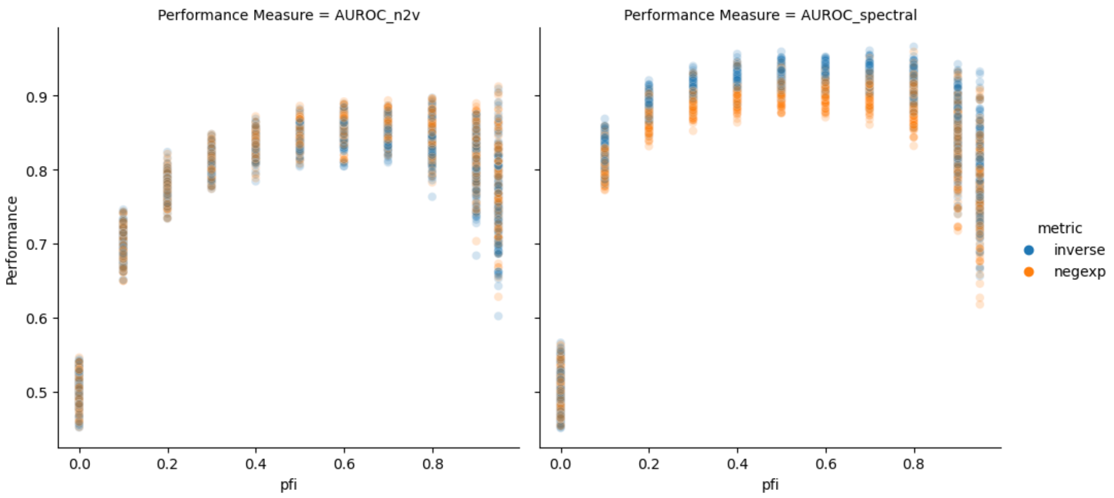
```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[~df["Performance Measure"].isin(["Accuracy_n2v",␣
     ↪"Accuracy_spectral"])], kind="scatter",
         x="pfi", y="Performance",
         col="Performance Measure",
         hue="gamma",
         alpha=0.2
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83df53f2b0>

     <Figure size 1400x1000 with 0 Axes>
```



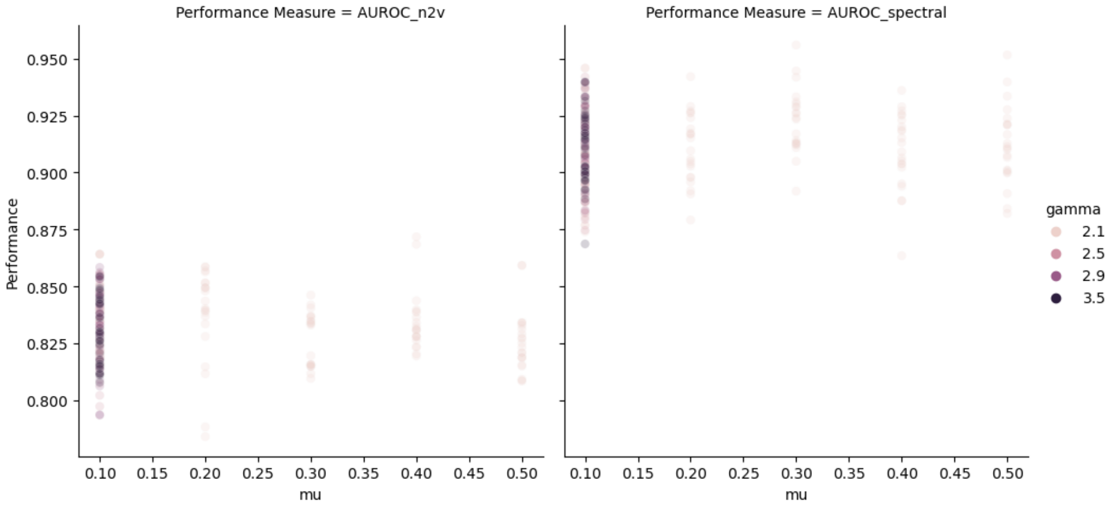
```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[~df["Performance Measure"].isin(["Accuracy_n2v",␣
      ↪"Accuracy_spectral"])], kind="scatter",
         x="pfi", y="Performance",
         col="Performance Measure",
         hue="metric",
         alpha=0.2
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83df6e7cd0>

     <Figure size 1400x1000 with 0 Axes>
```



```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[
             ~(df["Performance Measure"].isin(["Accuracy_n2v", "Accuracy_spectral"]))
             & (df["pfi"] == 0.4)
         ], kind="scatter",
         x="mu", y="Performance",
         col="Performance Measure",
         hue="gamma",
         alpha=0.2
     )
```

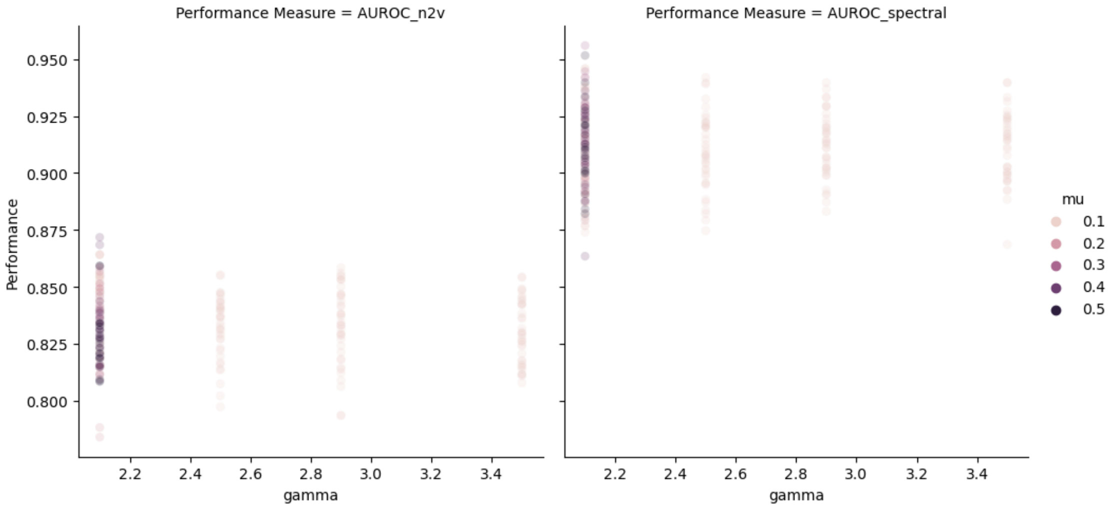
```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83c54128f0>

     <Figure size 1400x1000 with 0 Axes>
```



```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[
             ~(df["Performance Measure"].isin(["Accuracy_n2v", "Accuracy_spectral"]))
             & (df["pfi"] == 0.4)
         ], kind="scatter",
         x="gamma", y="Performance",
         col="Performance Measure",
         hue="mu",
         alpha=0.2
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83c5384ca0>
```

```
<Figure size 1400x1000 with 0 Axes>
```

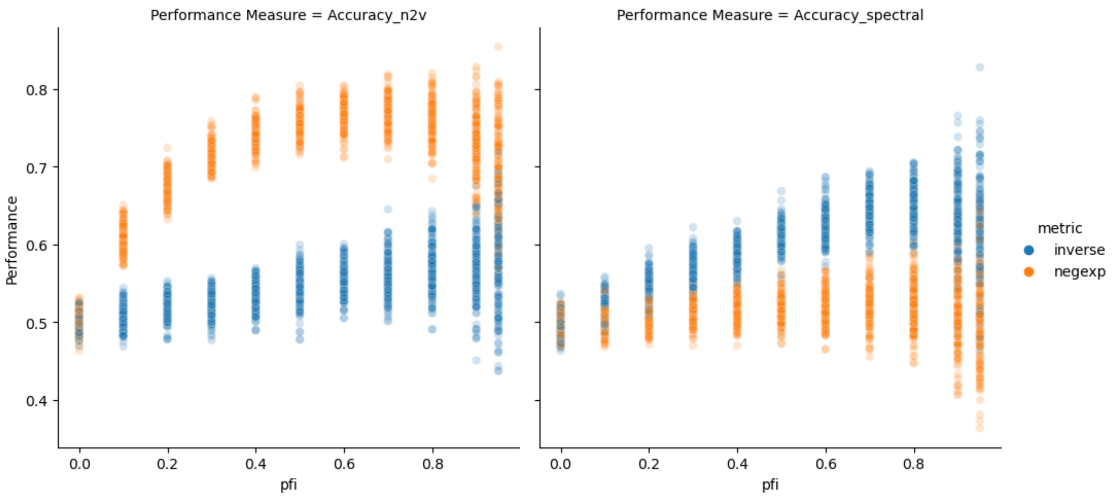


2.1.2 Accuracy

```
[ ]: plt.figure()
     sns.relplot(
         data=df.loc[df["Performance Measure"].isin(["Accuracy_n2v",␣
      ↪"Accuracy_spectral"])], kind="scatter",
         x="pfi", y="Performance",
         col="Performance Measure",
         hue="metric",
         alpha=0.2
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83c5338880>
```

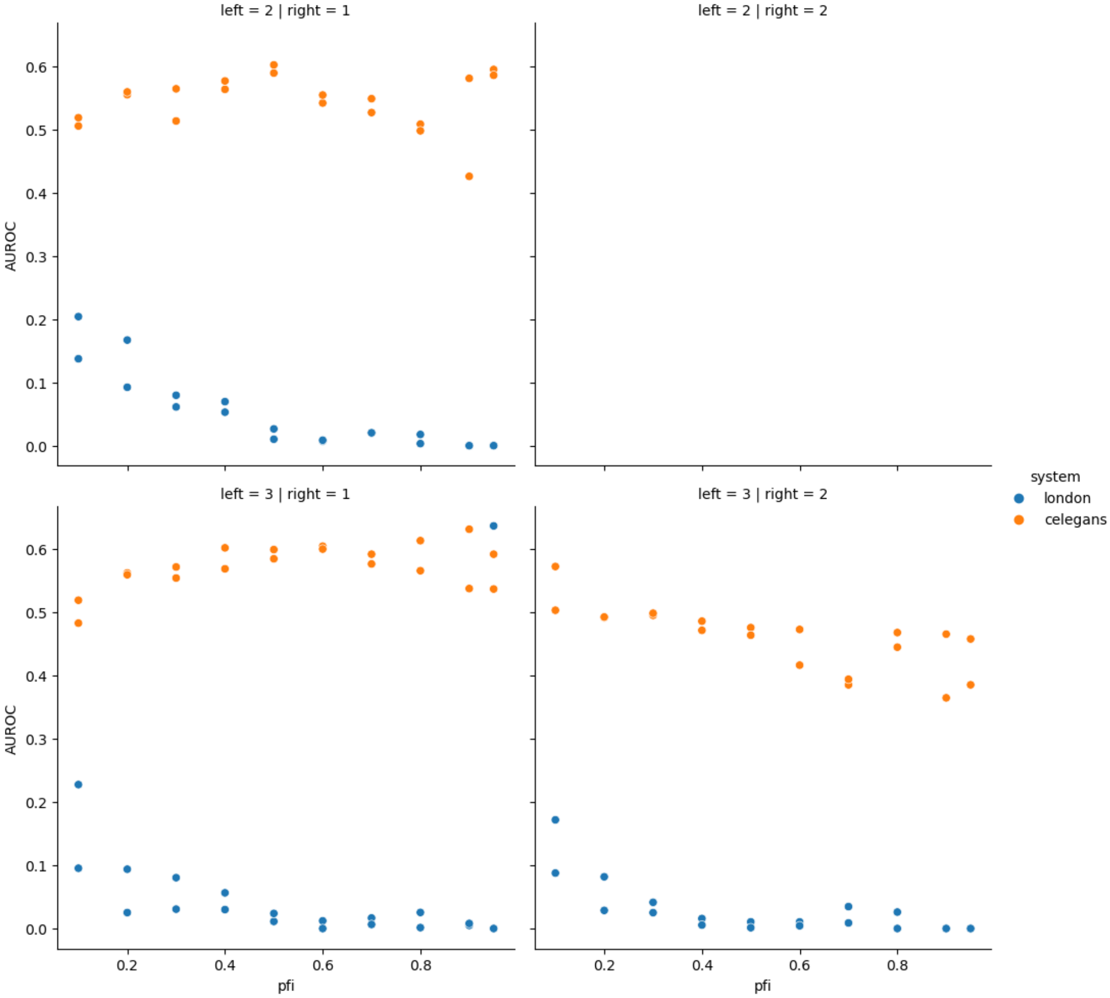
```
<Figure size 1400x1000 with 0 Axes>
```



2.2 Real

```
[ ]: plt.figure()
     sns.relplot(
         data=real, kind="scatter",
         x="pfi", y="AUROC",
         hue="system",
         row="left",
         col="right"
     )
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7f83c511efe0>

     <Figure size 1400x1000 with 0 Axes>
```


left = 2 | right = 1
left = 2 | right = 2
left = 3 | right = 1
left = 3 | right = 2
AUROC
0.0
0.1
0.2
0.3
0.4
0.5
0.6
pfi
0.2
0.4
0.6
0.8
system
london
celegans

[ ]: