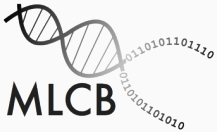
Progress update

Philip Hartout

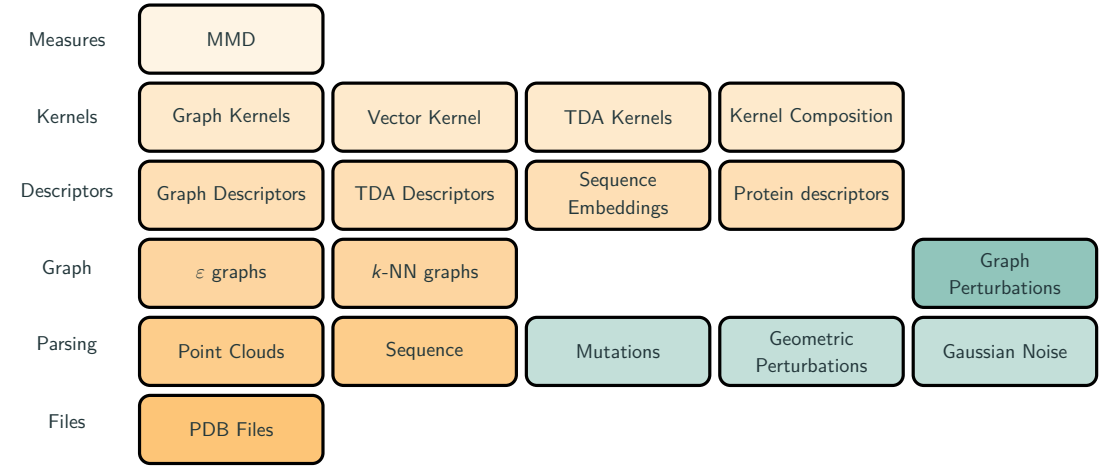
April 13, 2022



D BSSE

ETH zürich

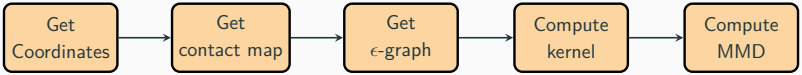
Overview



Green: perturbations

Orange: sequence embeddings

Composable transformations & using `sklearn` API standards sensibly

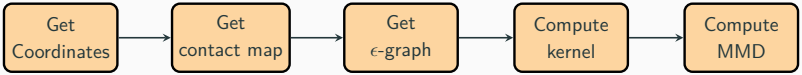



```
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins = base_feature_pipeline.fit_transform(paths_to_pdb_files)

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

Composable transformations & using `sklearn` API standards sensibly



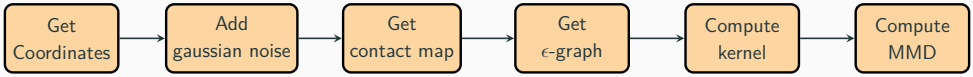

```
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins = base_feature_pipeline.fit_transform(paths_to_pdb_files)

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

What if we now want to add noise?

Composable transformations & using `sklearn` API standards sensibly




```
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins = base_feature_pipeline.fit_transform(paths_to_pdb_files)

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```

```
base_feature_pipeline = pipeline.Pipeline(
    [
        ("coordinates", Coordinates(granularity="CA", n_jobs=12),),
        (
            "add gaussian noise",
            GaussianNoise(
                random_seed=42, noise_mean=0, noise_variance=10, n_jobs=12,
            ),
        ),
        ("contact map", ContactMap(metric="euclidean", n_jobs=12,),),
        ("epsilon graph", EpsilonGraph(epsilon=epsilon, n_jobs=12),),
    ]
)

proteins_perturbed = base_feature_pipeline.fit_transform(paths_to_pdb_files)

graphs = load_graphs(proteins, graph_type="eps_graph")
graphs_perturbed = load_graphs(proteins_perturbed, graph_type="eps_graph")

mmd = MaximumMeanDiscrepancy(
    biased=True,
    squared=True,
    kernel=WeisfeilerLehmanKernel(
        n_jobs=12, n_iter=5, normalize=True, biased=True,
    ),
).compute(graphs, graphs_perturbed)
```