```python
from treeple.datasets import make_trunk_classification
from treeple.ensemble import HonestForestClassifier
from treeple.stats import PermutationHonestForestClassifier, build_coleman_forest


N_PERMUTATIONS = 10000

X, y = make_trunk_classification(
    n_samples=1000,
    n_dim=1,
    mu_0=0,
    mu_1=1,
    n_informative=1,
    seed=1,
)

est = HonestForestClassifier(
    n_estimators=100,
    max_samples=1.6,
    max_features=0.3,
    bootstrap=True,
    stratify=True,
    random_state=1,
)

est_null = PermutationHonestForestClassifier(
    n_estimators=100,
    max_samples=1.6,
    max_features=0.3,
    bootstrap=True,
    stratify=True,
    random_state=1,
)


observed_diff, _, _, pvalue, mix_diff = build_coleman_forest(
    est,
    est_null,
    X,
    y,
    metric="mi",
    n_repeats=N_PERMUTATIONS,
    return_posteriors=False,
    seed=1
)
```

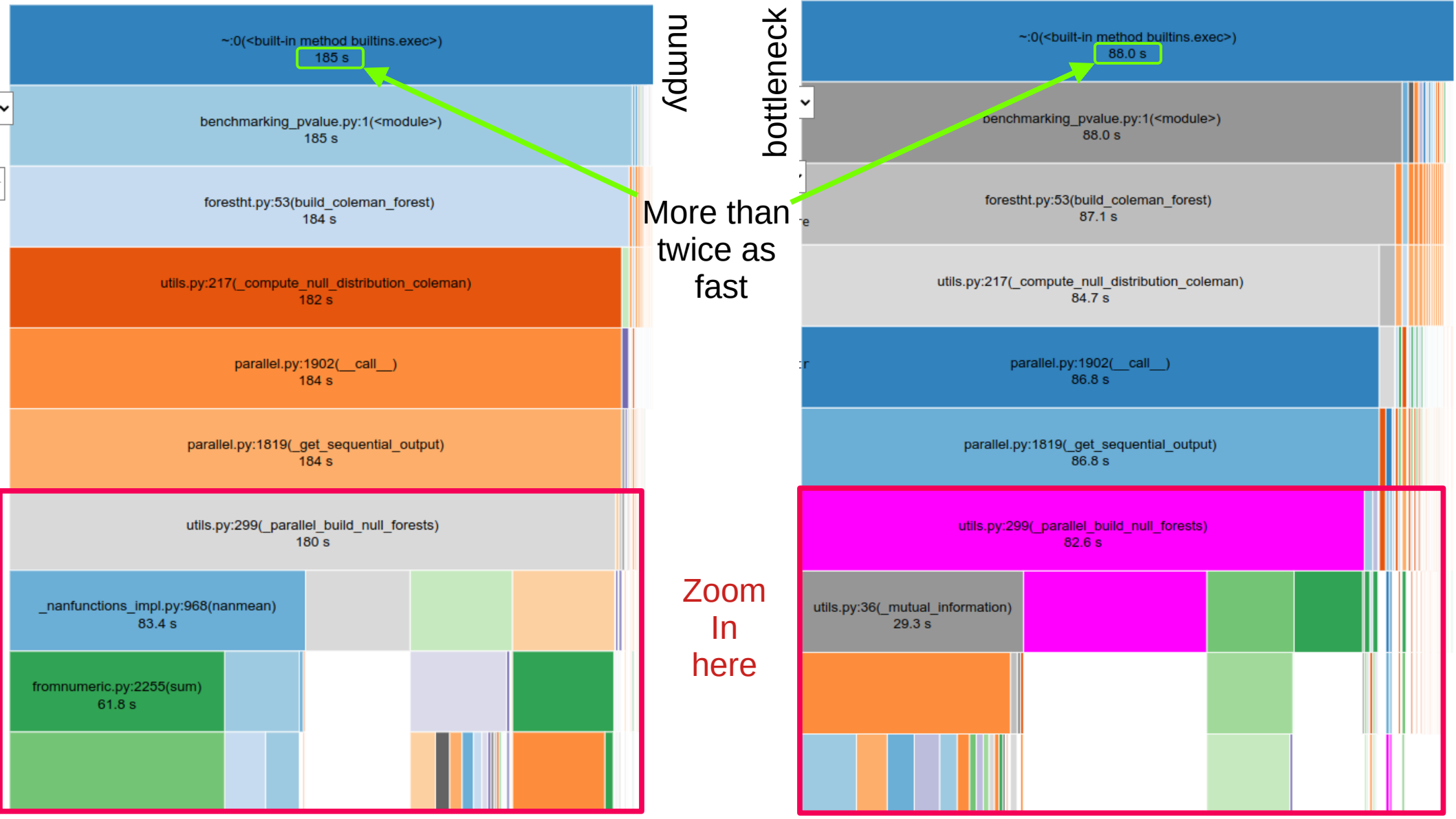
Bottleneck profiling test

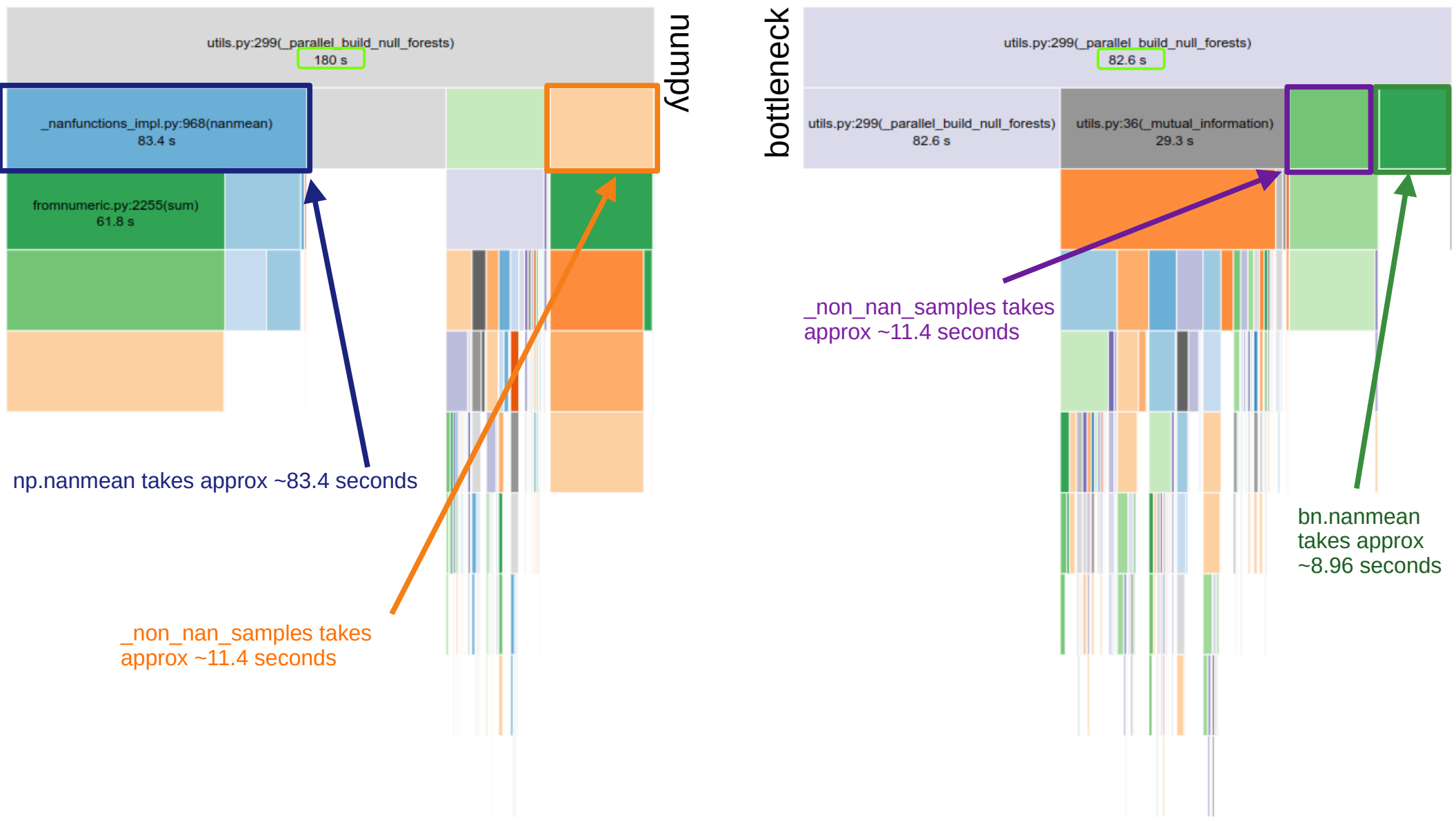
- Use bottleneck for nan related calculations in treeple.stats.utils.py
- Change the following two sections:

1) Change `np.isnan(posterior_arr).any(axis=2)` to `bn.anynan(posterior, axis=2)` here.

2) Change `np.nanmean → bn.nanmean` here.

Using cProfile to profile the code on the left and snakeviz to display the results, side by side profiles are compared for performance on the next slides.


~:0(<built-in method builtins.exec>)
185 s
benchmarking_pvalue.py:1(<module>)
185 s
forestht.py:53(build_coleman_forest)
184 s
utils.py:217(_compute_null_distribution_coleman)
182 s
parallel.py:1902(__call__)
184 s
parallel.py:1819(_get_sequential_output)
184 s
utils.py:299(_parallel_build_null_forests)
180 s
_nanfunctions_impl.py:968(nanmean)
83.4 s
fromnumeric.py:2255(sum)
61.8 s
numpy
bottleneck
More than twice as fast
~:0(<built-in method builtins.exec>)
88.0 s
benchmarking_pvalue.py:1(<module>)
88.0 s
forestht.py:53(build_coleman_forest)
87.1 s
utils.py:217(_compute_null_distribution_coleman)
84.7 s
parallel.py:1902(__call__)
86.8 s
parallel.py:1819(_get_sequential_output)
86.8 s
utils.py:299(_parallel_build_null_forests)
82.6 s
utils.py:36(_mutual_information)
29.3 s
Zoom In here


numpy
utils.py:299(_parallel_build_null_forests)
180 s
_nanfunctions_impl.py:968(nanmean)
83.4 s
fromnumeric.py:2255(sum)
61.8 s
np.nanmean takes approx ~83.4 seconds
_non_nan_samples takes
approx ~11.4 seconds
bottleneck
utils.py:299(_parallel_build_null_forests)
82.6 s
utils.py:299(_parallel_build_null_forests)
82.6 s
utils.py:36(_mutual_information)
29.3 s
_non_nan_samples takes
approx ~11.4 seconds
bn.nanmean
takes approx
~8.96 seconds