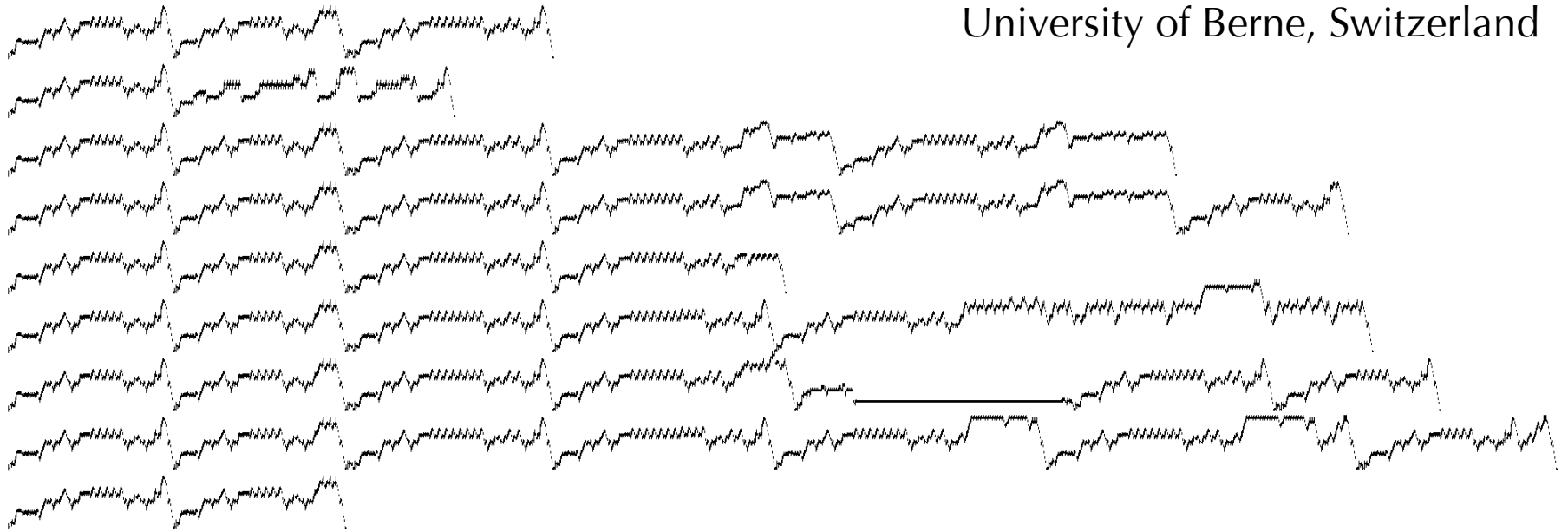


Exploiting the Analogy between Traces and Signal Processing

Adrian Kuhn and Orla Greevy
Software Composition Group
University of Berne, Switzerland



Roadmap — Feature Characterization

Four research questions

A technical challenge

One solution for both

And, our findings.

Which features
behave similar?

Which pattern are
common to all features?

We execute features, and ask:

Which pattern are
unique to single features?

How do features
map to architecture?

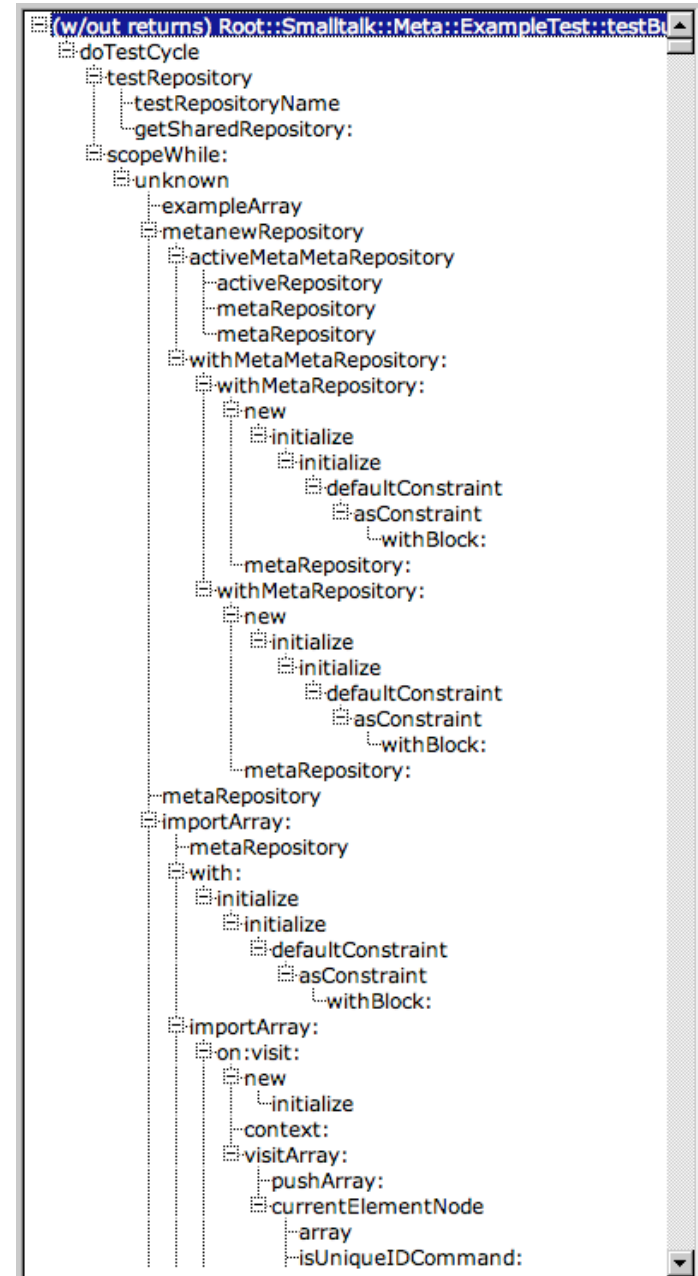
How can we compare multiple traces?

Dynamic analysis poses a technical challenge.

Huge volumes of data

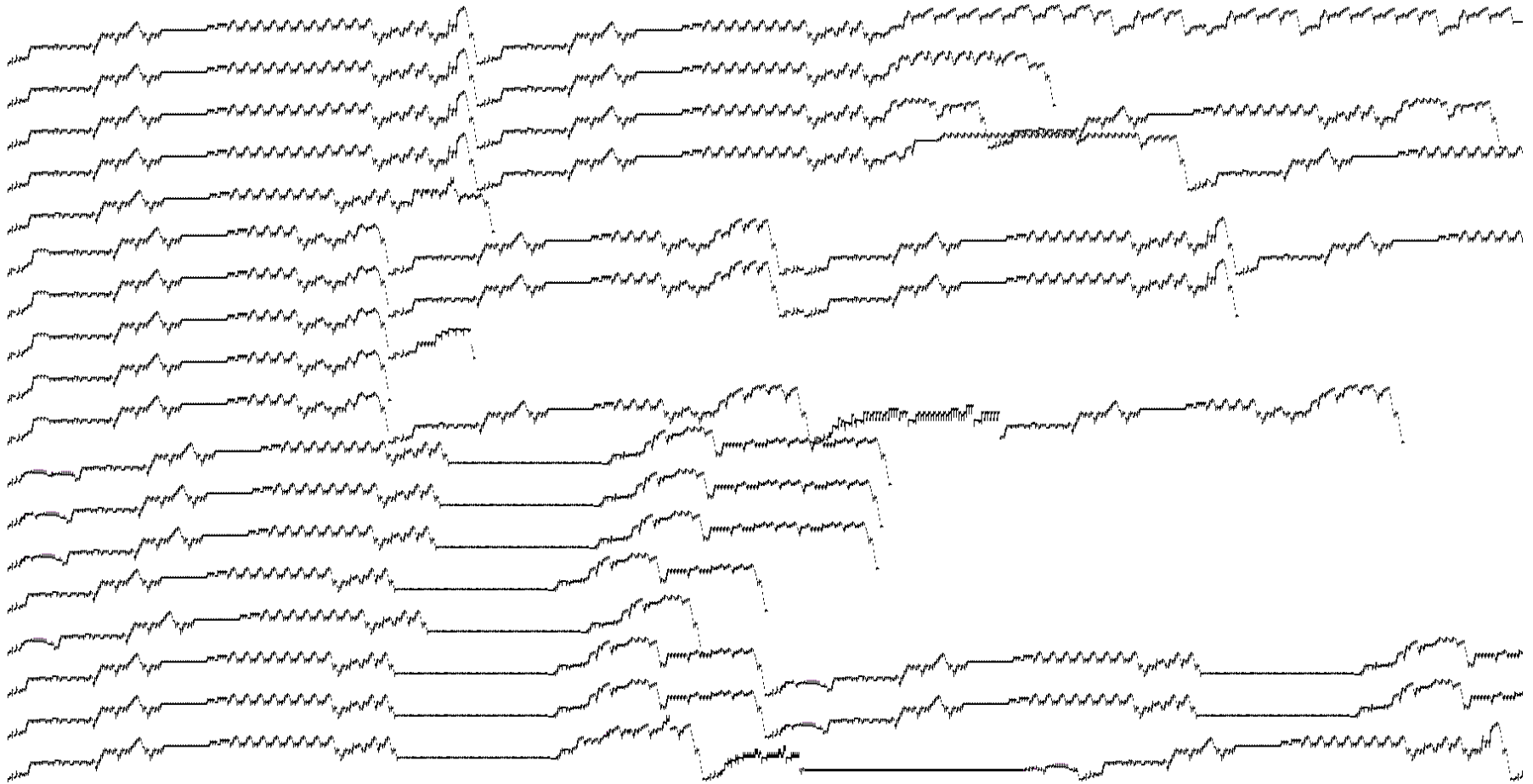
Millions of calls

Dozens of traces

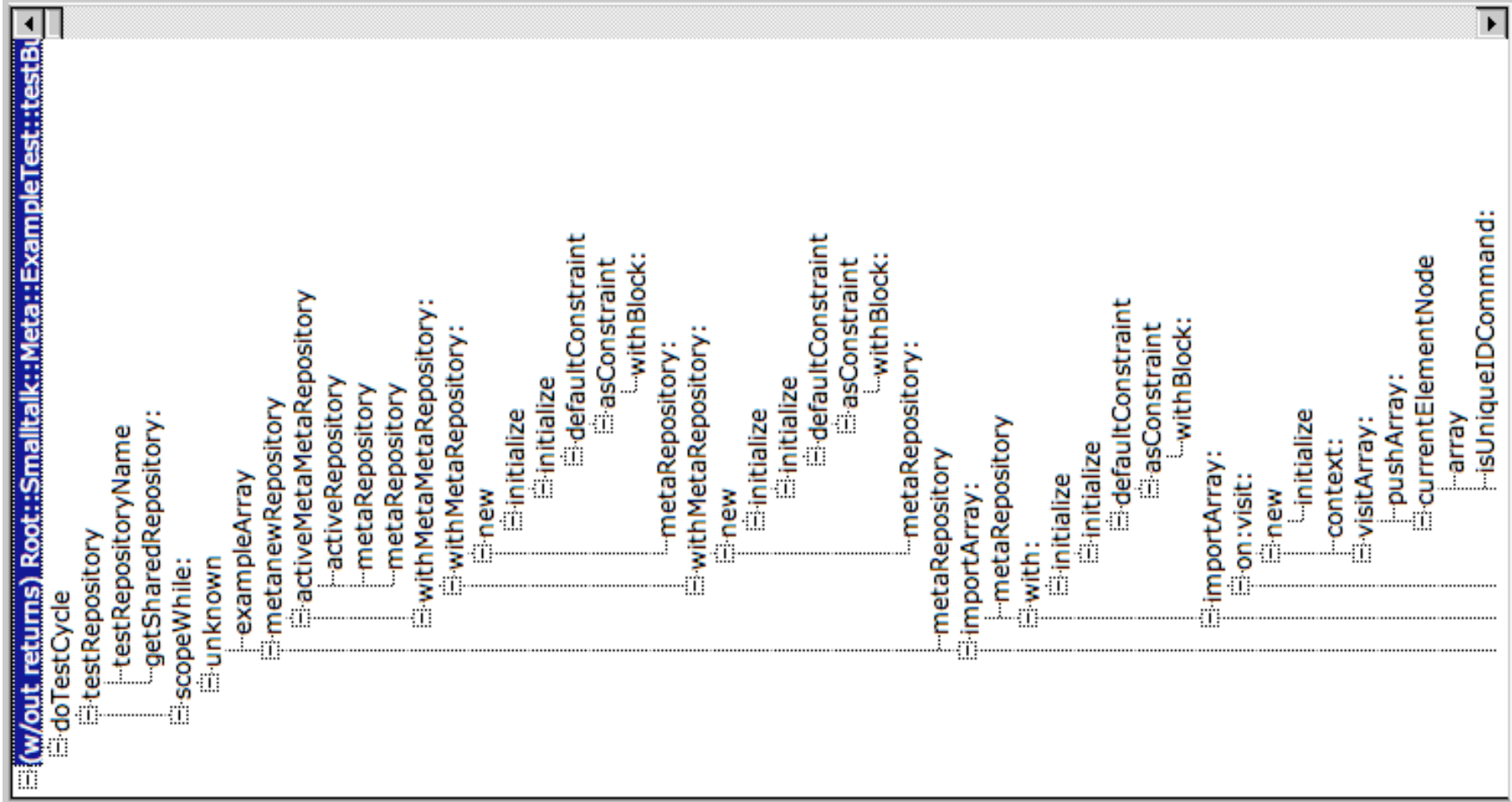


How can we summarize a trace?

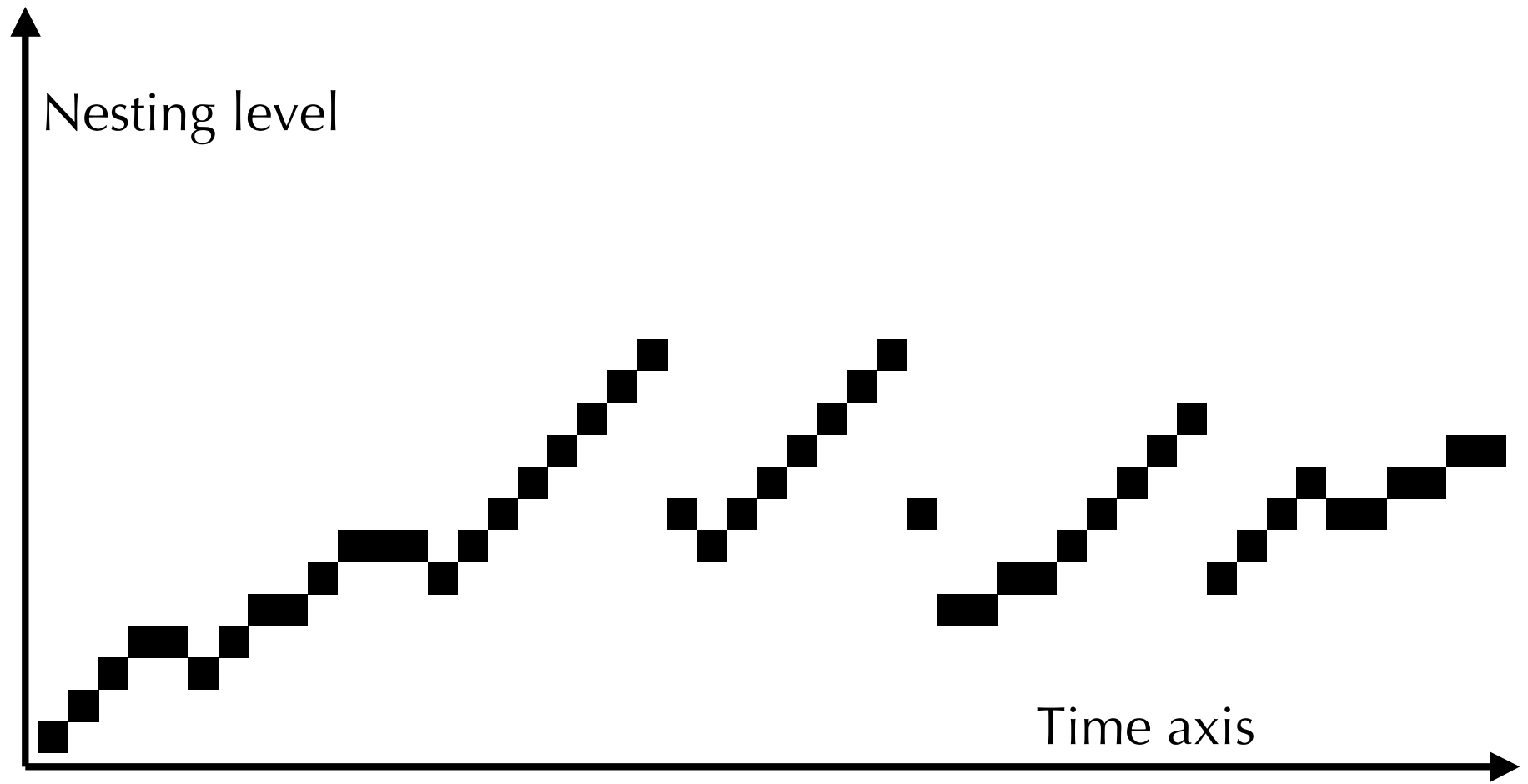
We can solve both problems treating traces as signals in time!



Rotate the call-graph tree by 90 degrees.

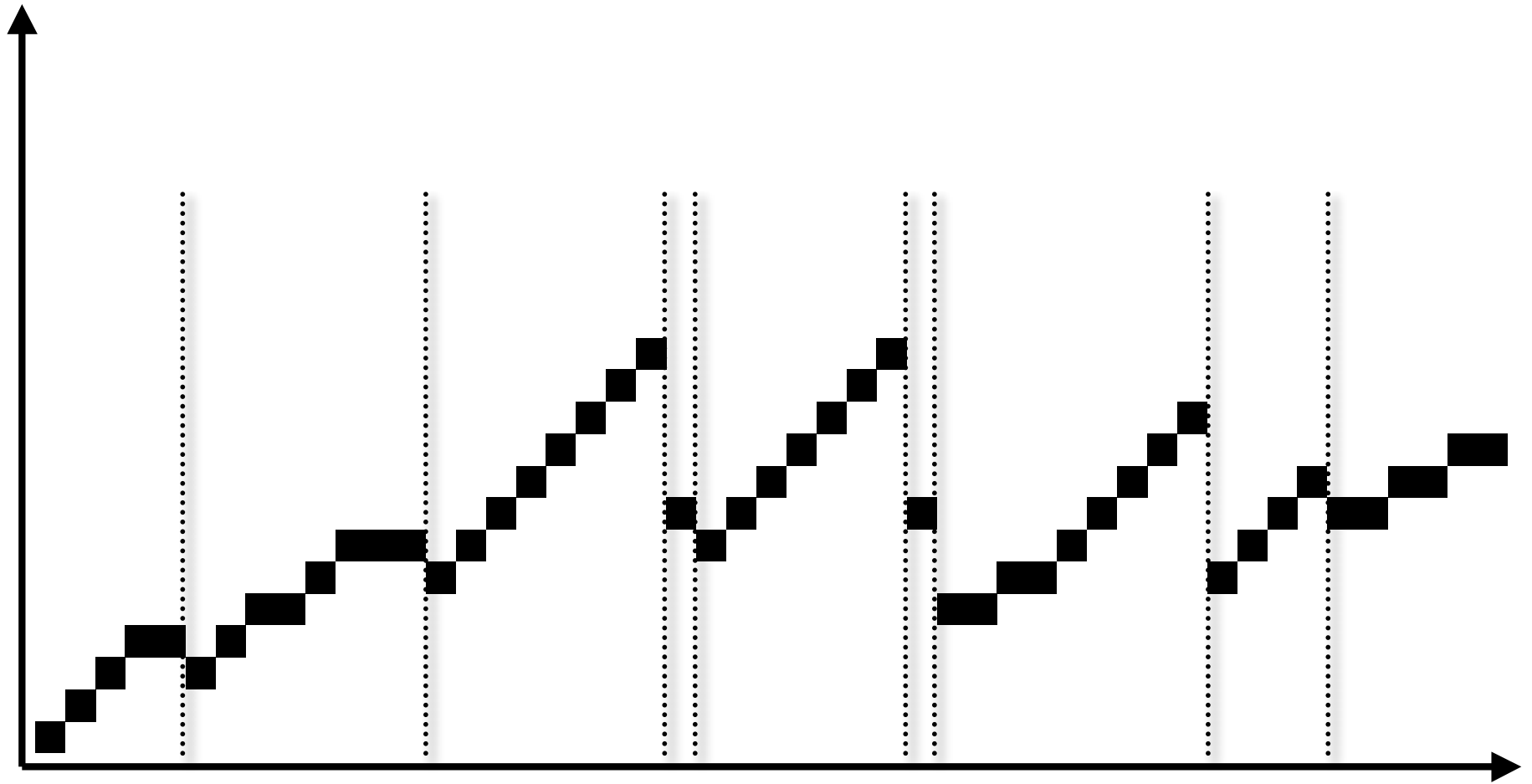


Rotate the call-graph tree by 90 degrees.

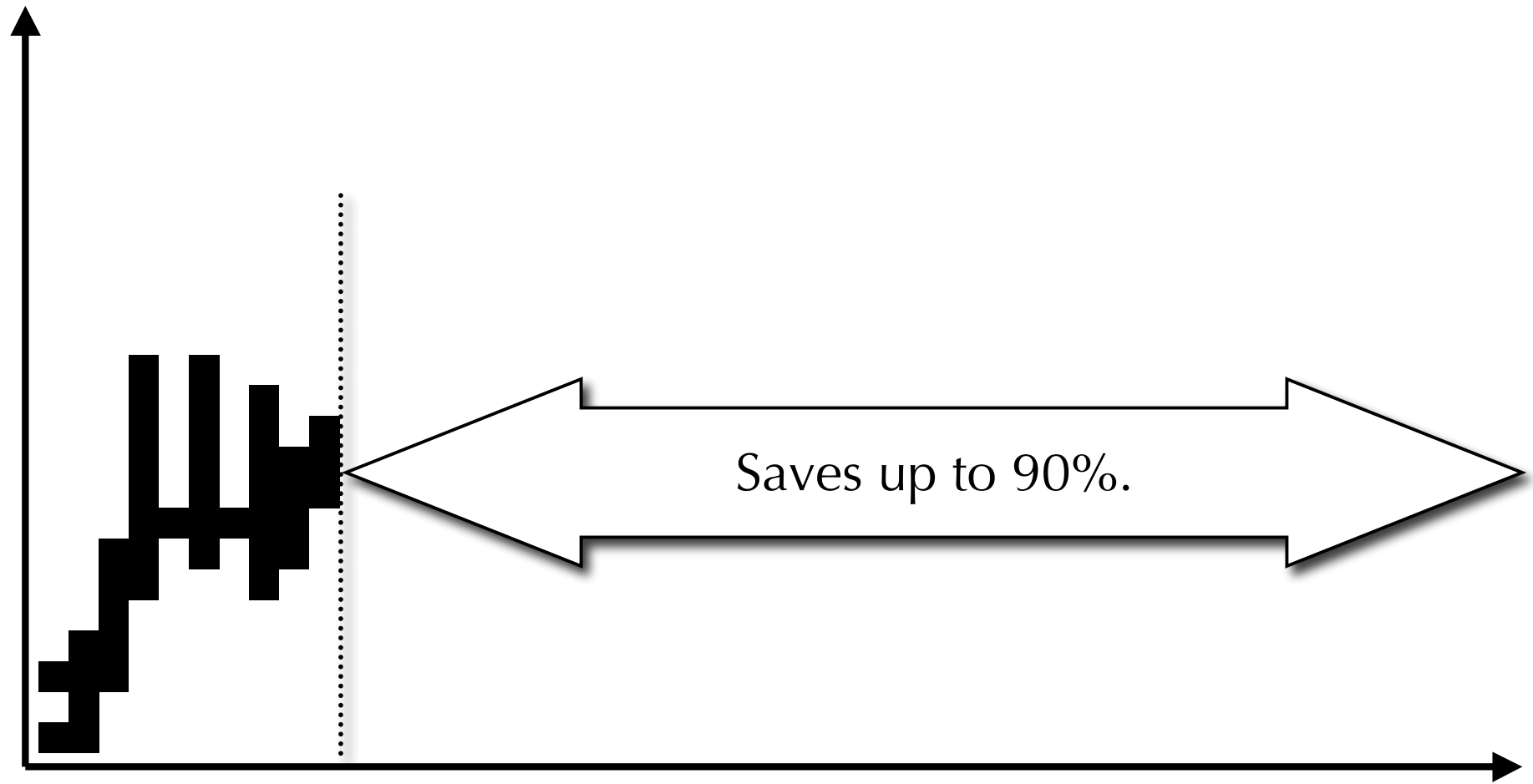


How can we summarize a trace?

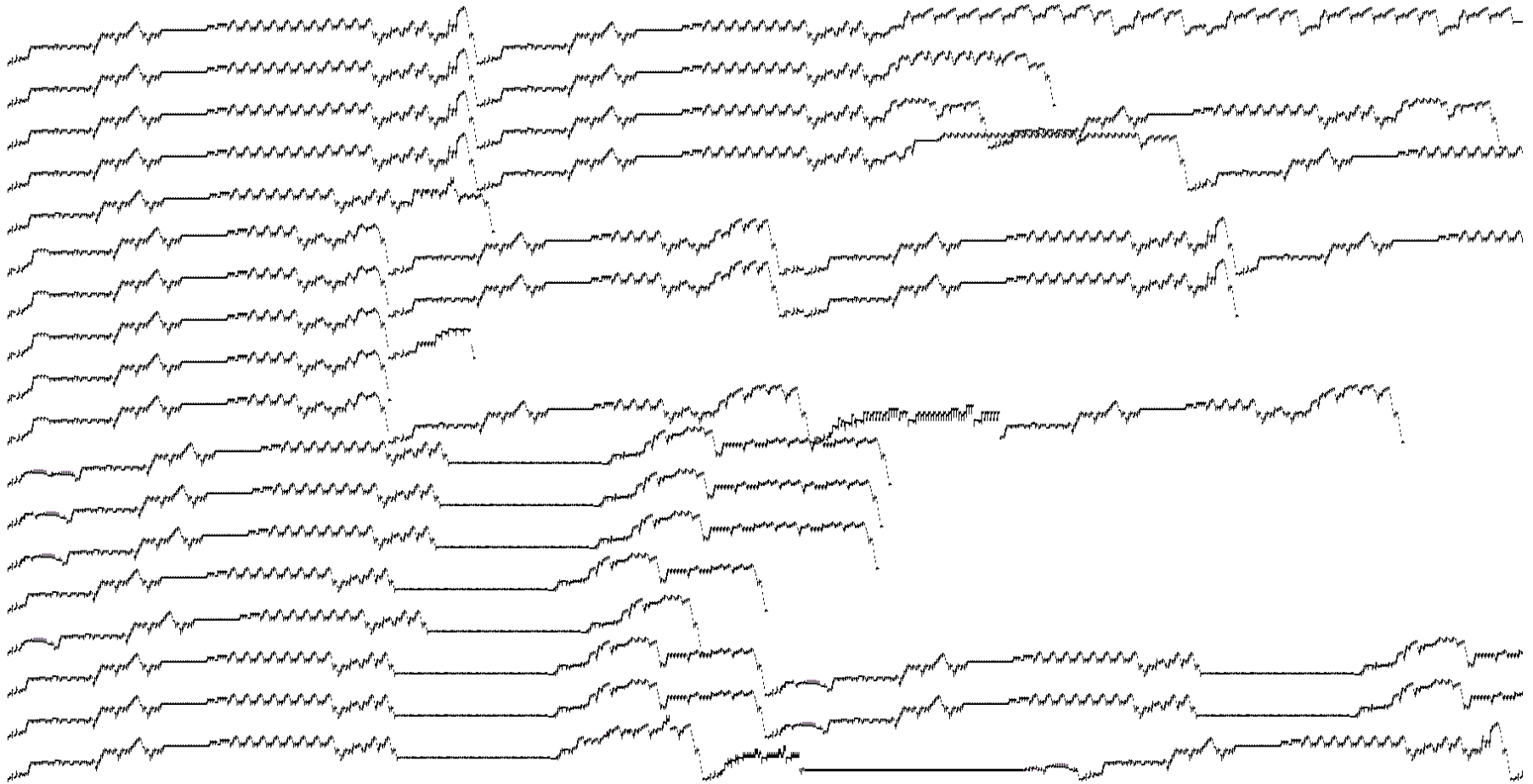
Monotone Subsequence Summarization



Monotone Subsequence Summarization

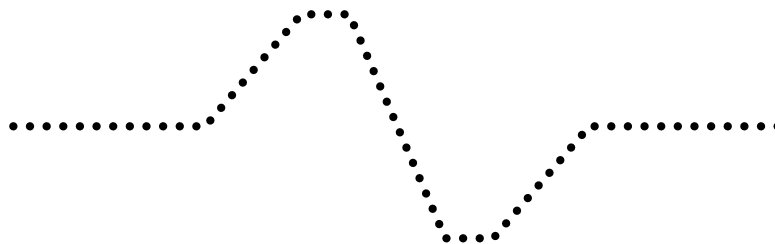
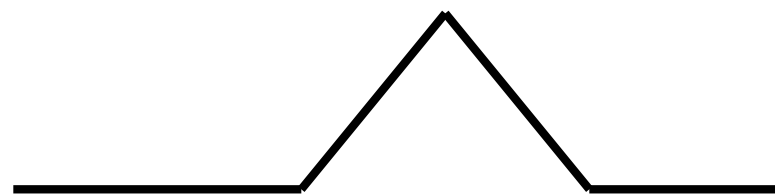
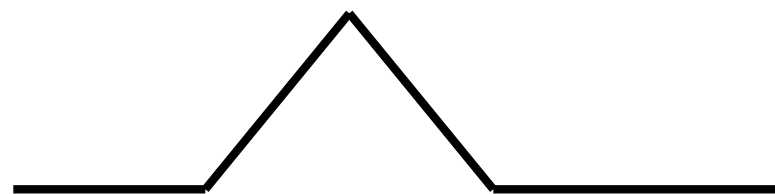
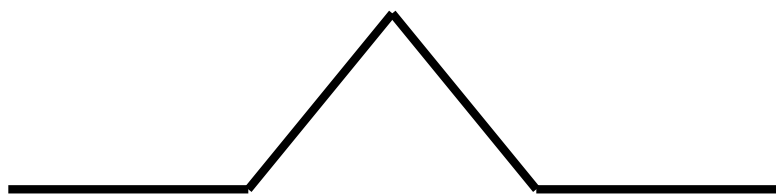
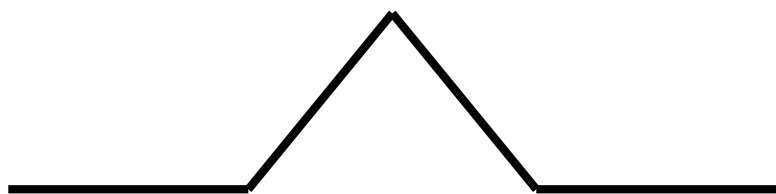


These are more than 500'000 calls, in 19 traces – all on one single screen.

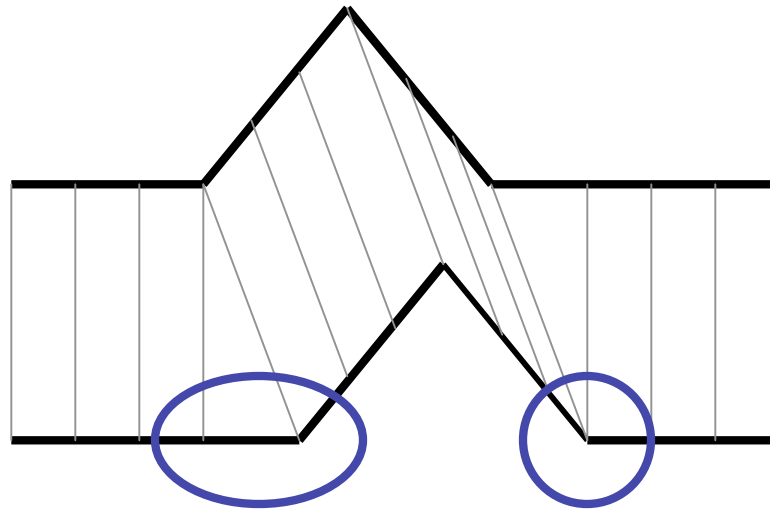


How can we compare two signals?

Subtracting one signal from another is very sensitive to synchronization.

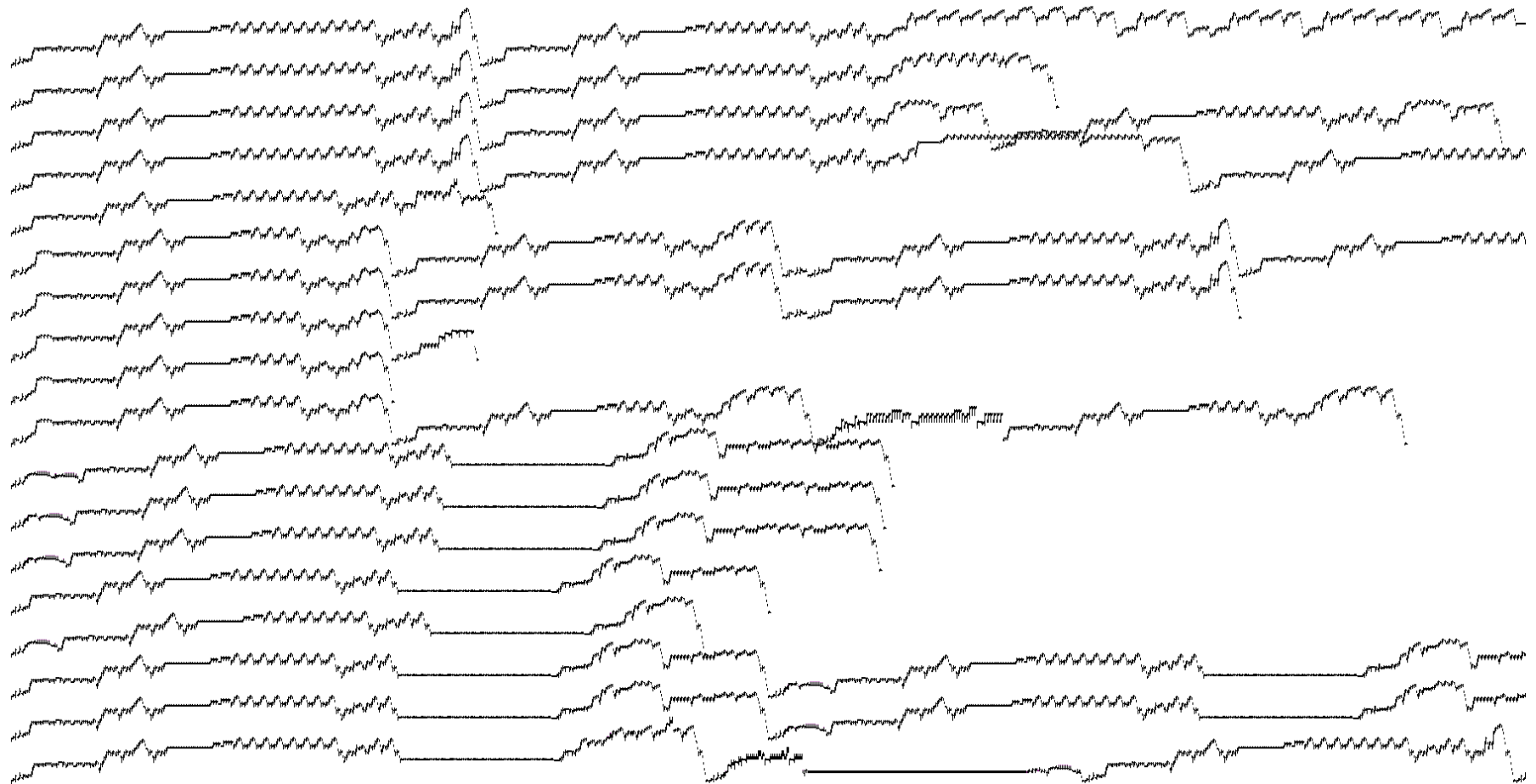


Dynamic Time Warping (DTW) can fall behind and catch up again.

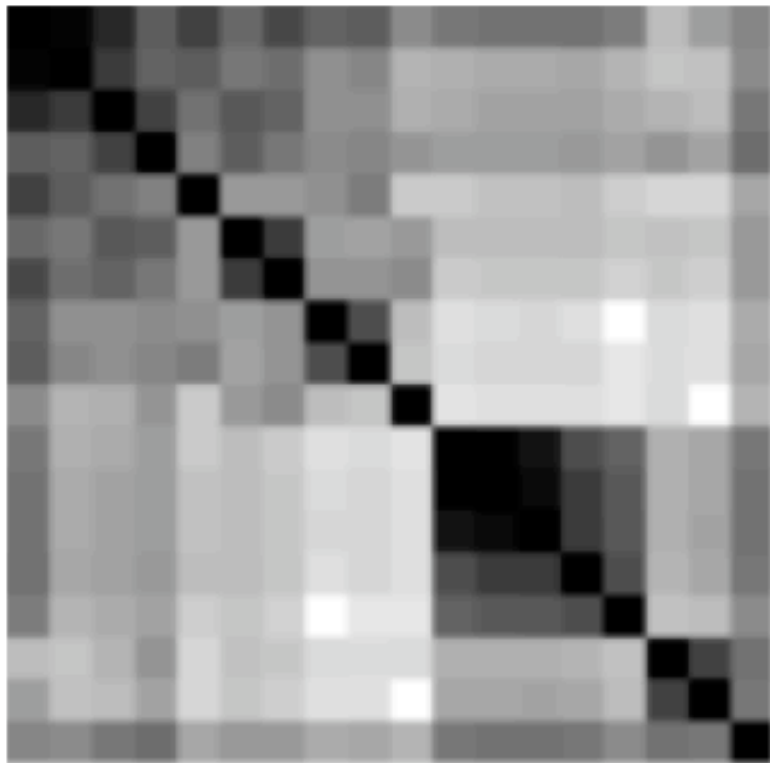


.....

These traces are sorted by similarity.



These features are sorted by similarity.



comps, props, stylesheet, template.

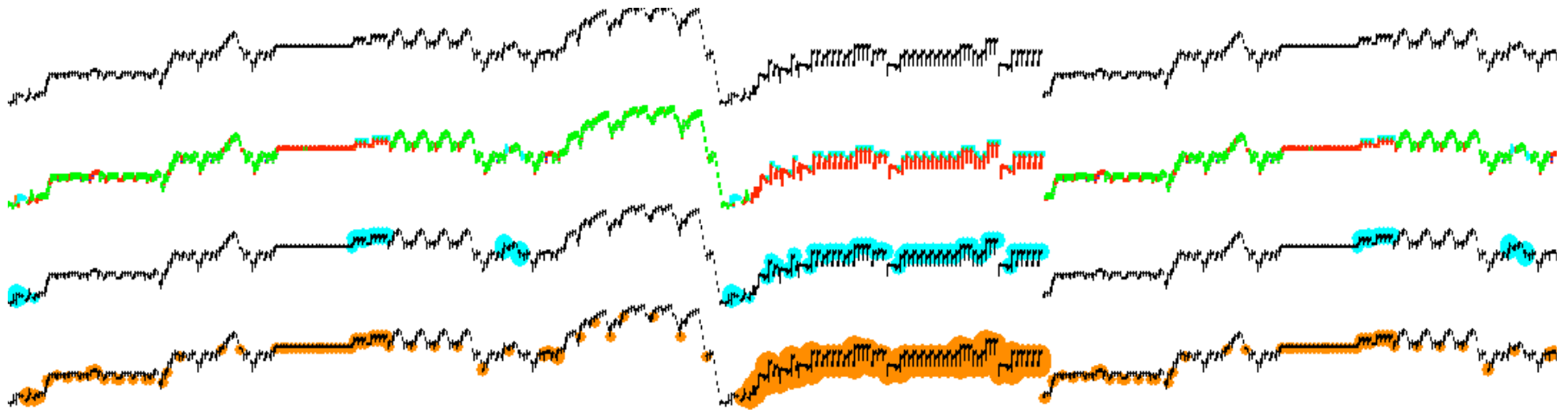
history, changes, login.

rss, resolveURL, search.

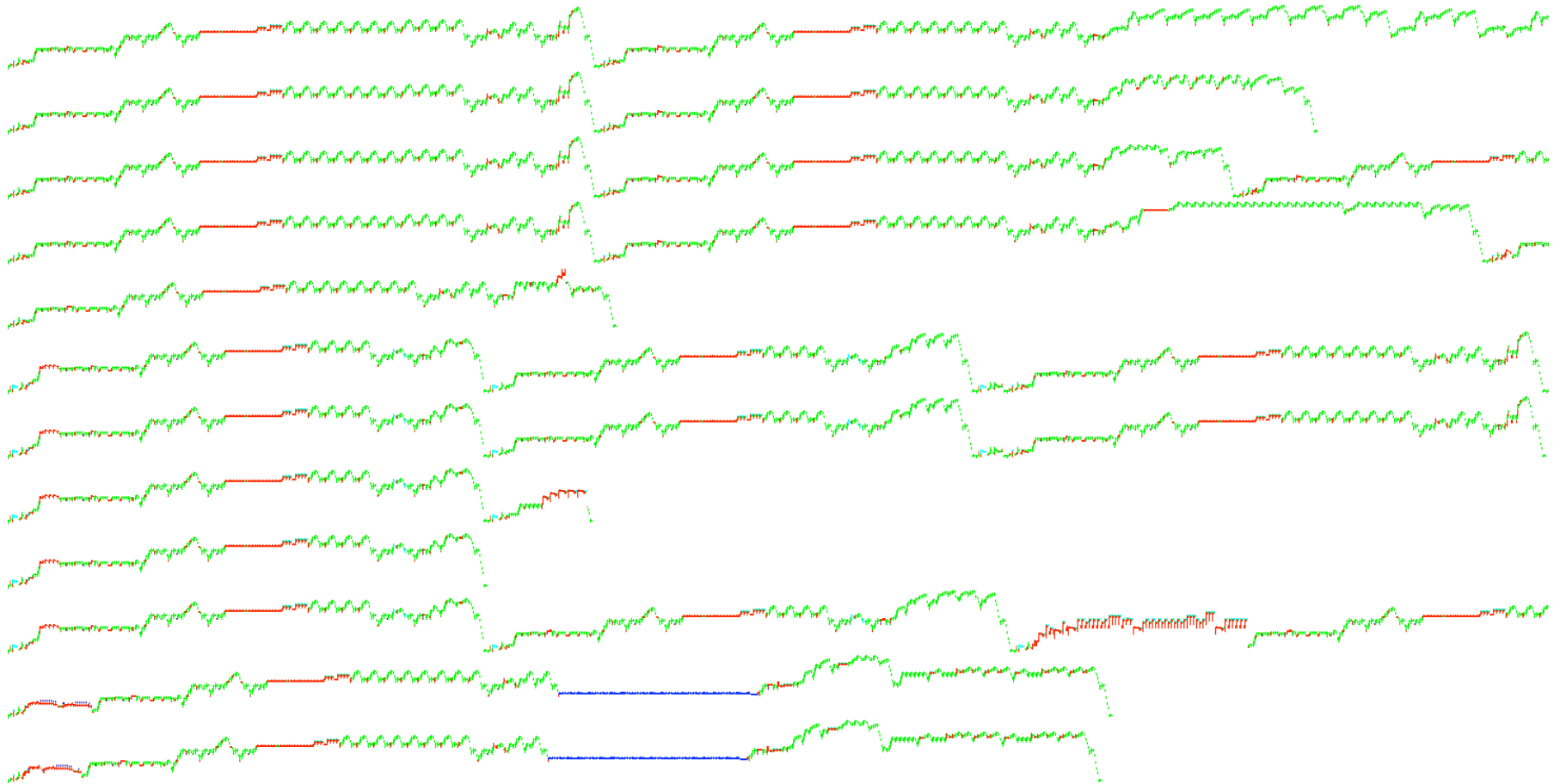
copychild, addfolderchild, addpagechild, contents, removechild.

addfolder, addpage, editpage.

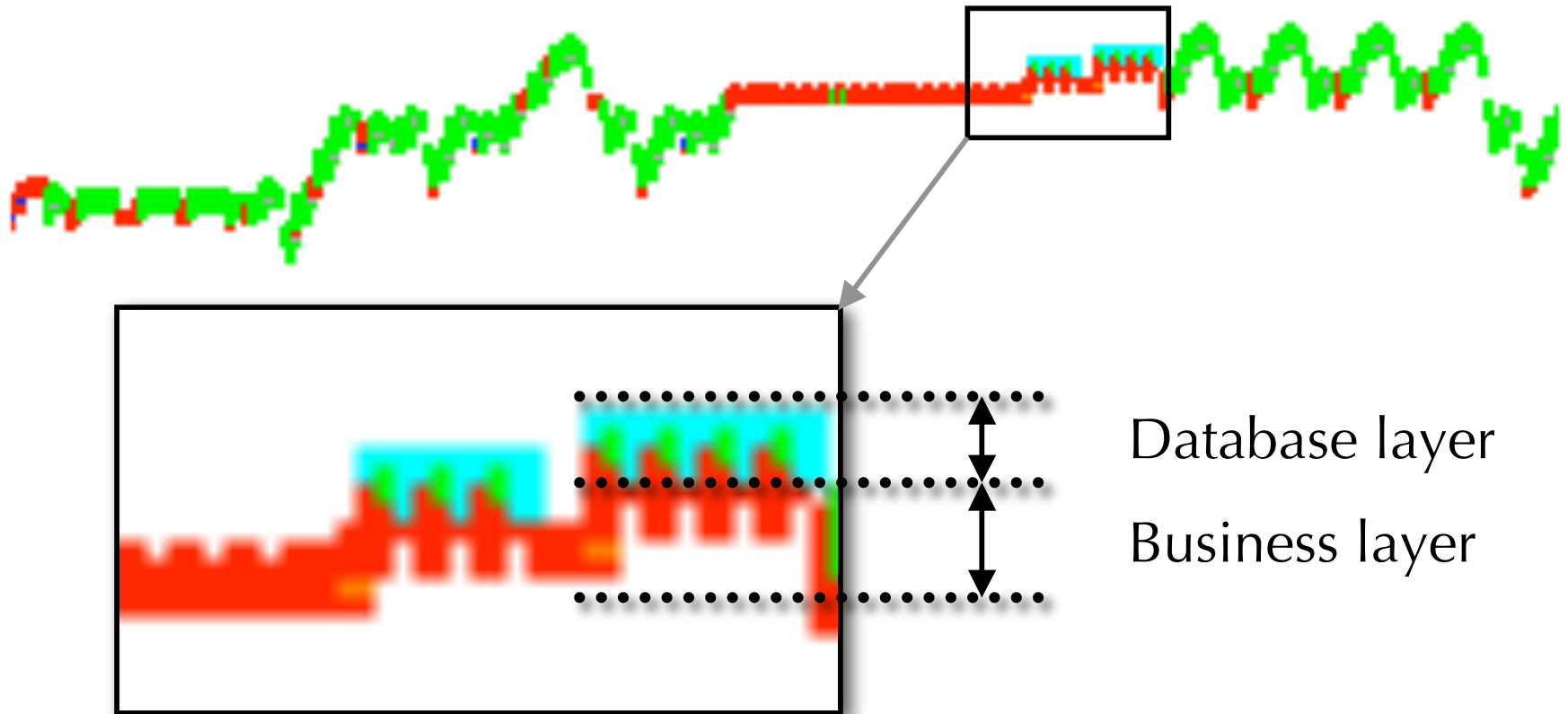
We can annotate a feature signal with architectural information.



Only by comparing multiple traces we can see what is specific to one trace.

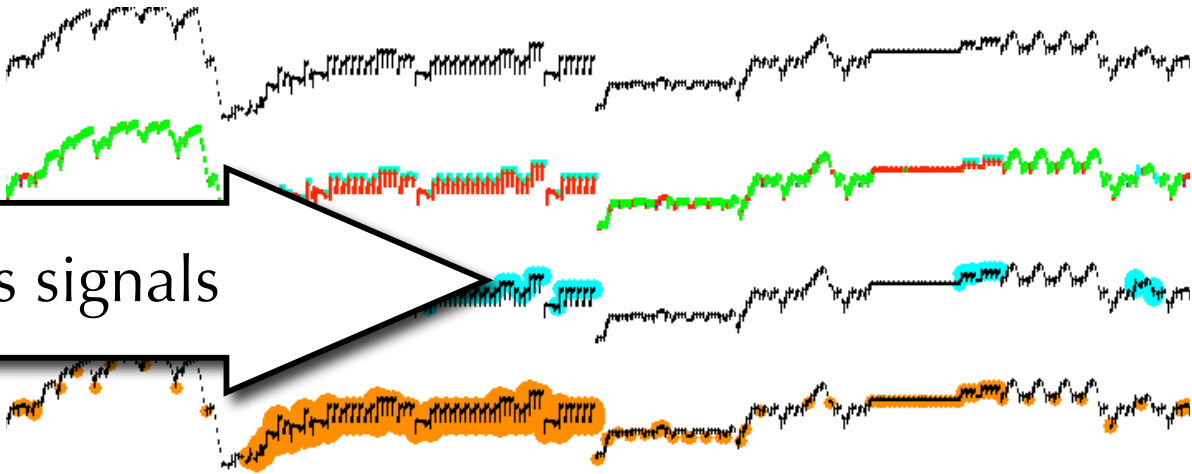


Architecture splits call-chains into parts.



Feature characterization using dynamic analysis.

```
(w/out returns) Root::Smalltalk::Meta::ExampleTest::testBt
├─ doTestCycle
│ └─ testRepository
│   ├── testRepositoryName
│   └─ getSharedRepository:
│     └─ scopeWhile:
│       └─ unknown
│         └─ exampleArray
│           └─ metanewRepository
│             ├── activeMetaMetaRepository
│             │ ├── activeRepository
│             │ └─ metaRepository
│             └─ with MetaMetaRepository:
│               └─ with MetaRepository:
│                 └─ new
│                   └─ initialize
│                     └─ initialize
└─ defaultConstraint
  └─ asConstraint
    └─ withBlock:
      └─ metaRepository:
        └─ importArray:
          ├── metaRepository
          └─ with:
            ├── initialize
            │ └─ initialize
            │   └─ defaultConstraint
            │     └─ asConstraint
            │       └─ withBlock:
            └─ importArray:
              ├── on:visit:
              └─ new
                └─ initialize
```



Treating traces as signals