iPyRegulus

Analysis of ensembles of simulations

(Regulus 2.0 in Jupyter Lab)

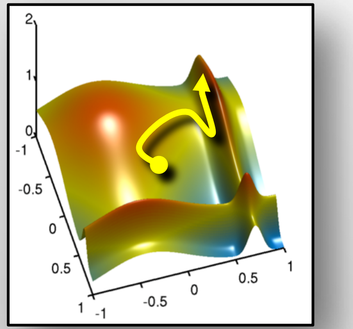
Yarden Livnat, Dan Maljovec, Valerio Pascucci

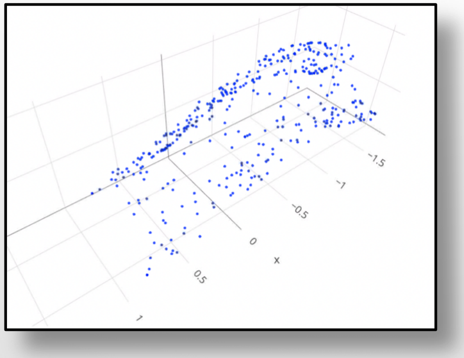
Scientific Computing and Imaging Institute
University of Utah

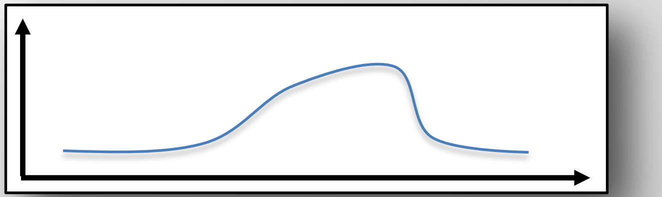
Work is funded in part by NEUP

Technical Workshop of Fuel Cycle Simulation June 2019

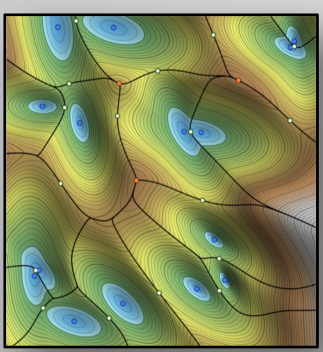
Topological Analysis using Morse-Smale Complexes

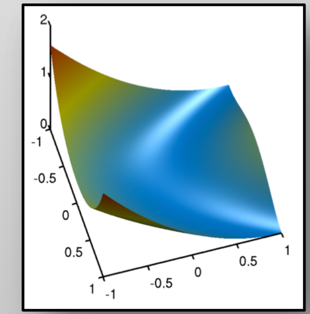




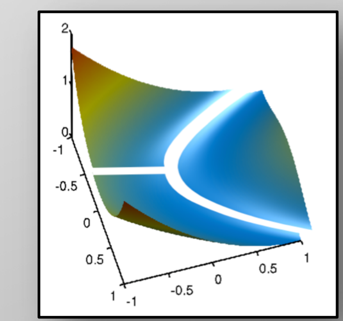


Partition the space into monotonic patches



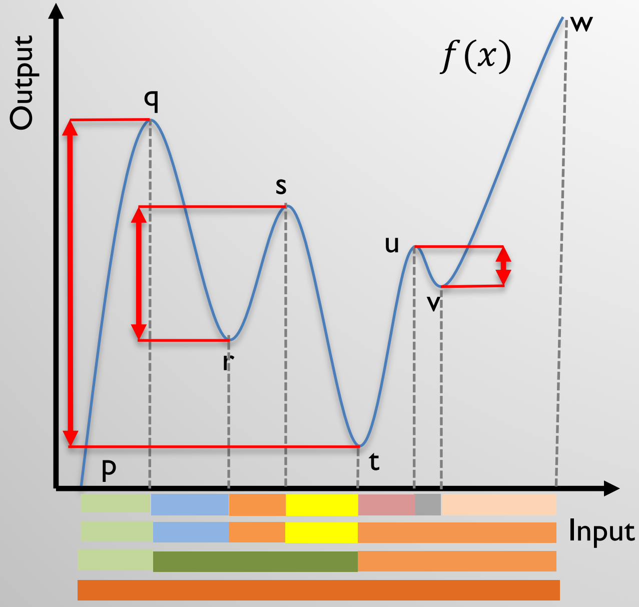


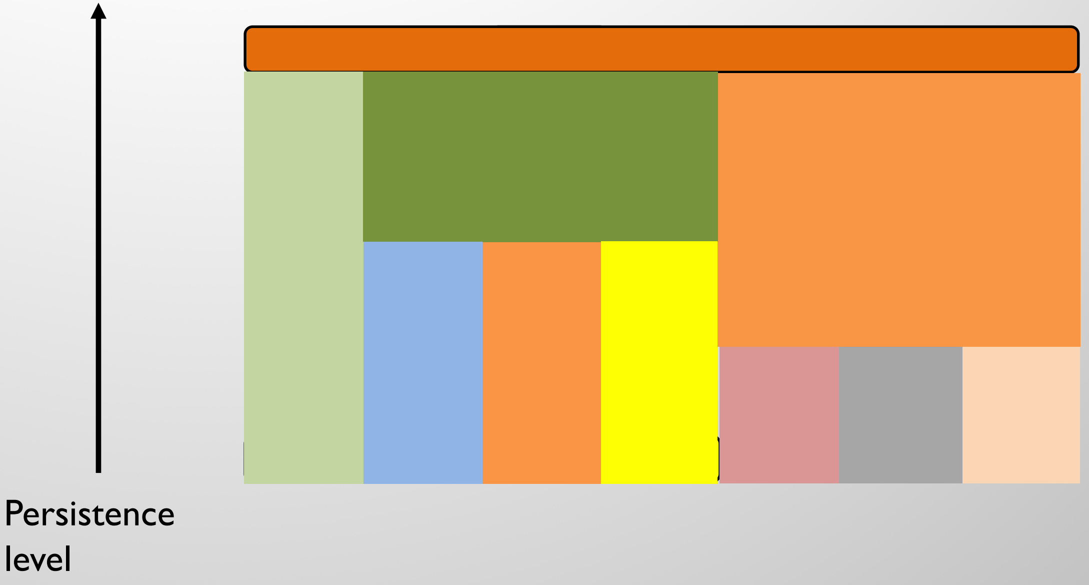
Morse Complex:
Single local minimum

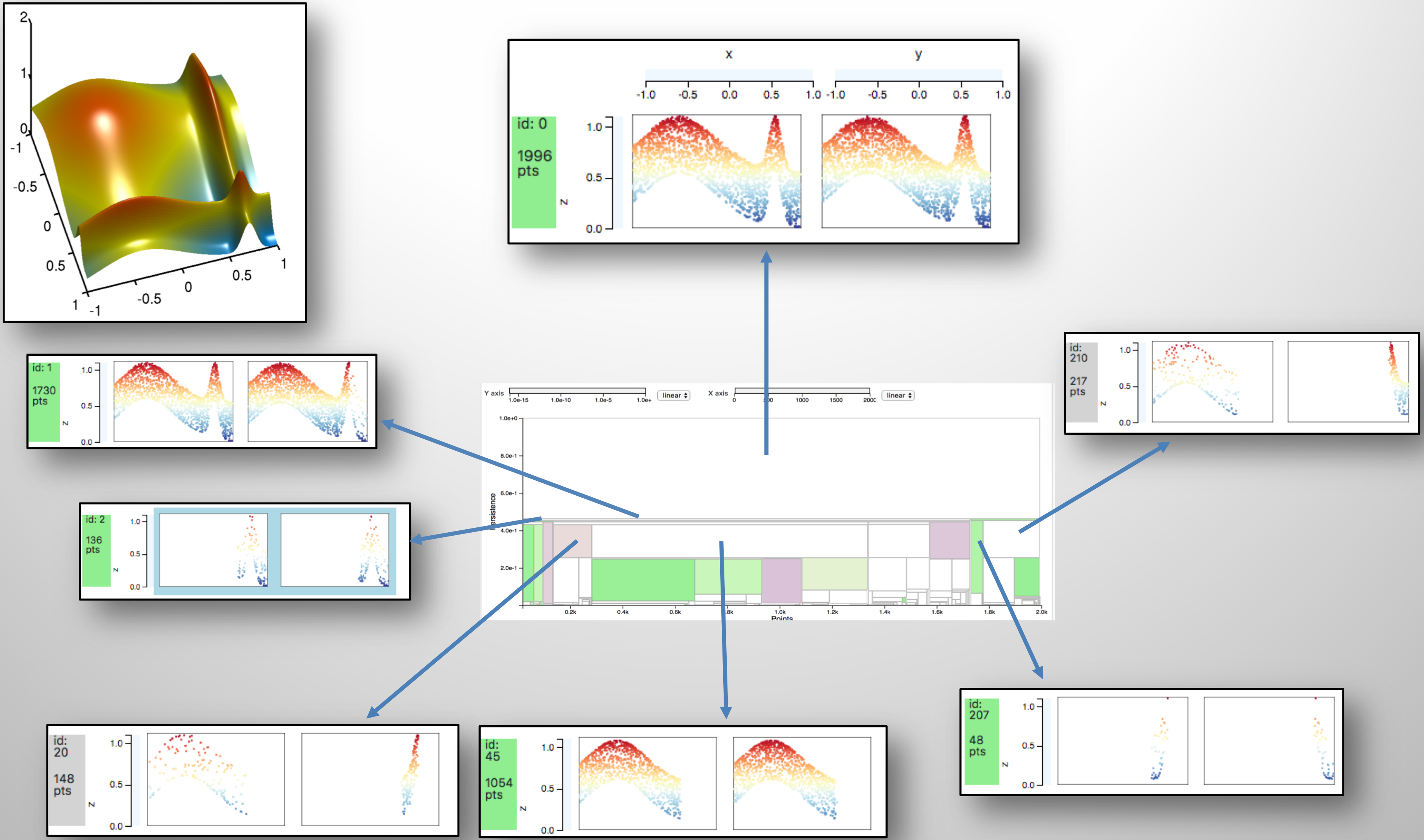


Morse-Smale Complex:
Single local minimum
and a single local maximum

Regulus Hierarchical Partition Tree



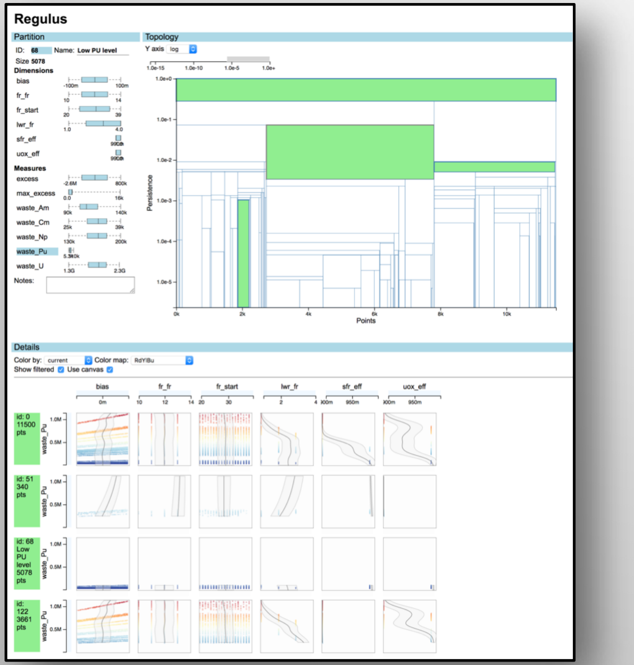




2
1
0
-1
-0.5
0
0.5
1
-1
-0.5
0
0.5
1
x
y
-1.0
-0.5
0.0
0.5
1.0
id: 0
1996 pts
z
Y axis
1.0e-15
1.0e-10
1.0e-5
1.0e+
linear
X axis
0
500
1000
1500
2000
1.0e+0
8.0e-1
6.0e-1
4.0e-1
2.0e-1
Persistence
0.2k
0.4k
0.6k
0.8k
1.0k
1.2k
1.4k
1.6k
1.8k
2.0k
Points
id: 1
1730 pts
id: 2
136 pts
id: 20
148 pts
id: 45
1054 pts
id: 207
48 pts
id: 210
217 pts

Regulus.js

- Hierarchical Partition Tree based on levels of detail
- Web-based with a python backend



Regulus

Partition

ID: 68 Name: Low PU level

Size 5078

Topology

Y axis: log

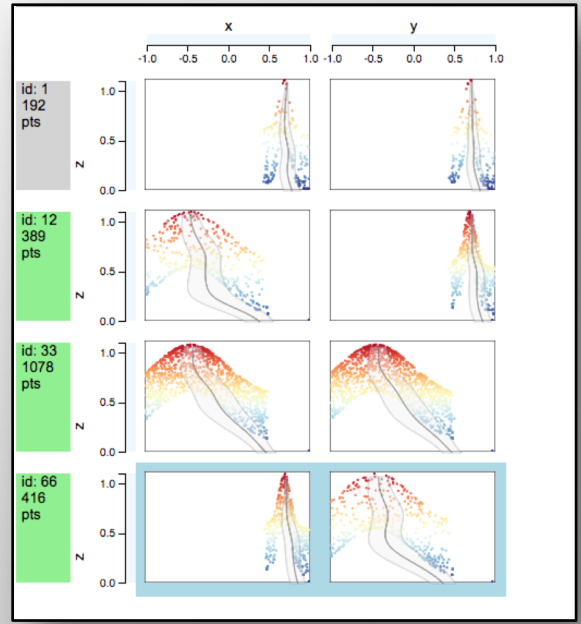
Details

Color by: current Color map: RdYlBu

Show filtered ☑ Use canvas ☑

Notes:

Hierarchical view of a Morse-Smale Complex



Details view

iPyRegulus

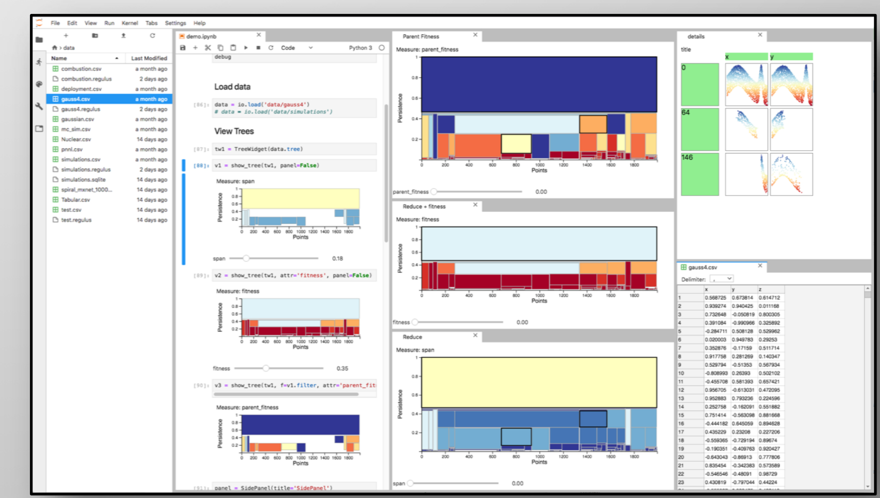
Break away from the typical visualization-first approach

Interactive (and scripted) exploration analysis

Define attributes and measures on the fly

Explore data from multiple views

Coordinate between views on the fly



Name	Last Modified
combustion.csv	a month ago
combustion.regulus	2 days ago
deployment.csv	a month ago
gauss4.csv	a month ago
gauss4.regulus	2 days ago
gaussian.csv	a month ago
mc_sim.csv	a month ago
Nuclear.csv	14 days ago
pnnl.csv	a month ago
simulations.csv	a month ago
simulations.regulus	2 days ago
simulations.sqlite	14 days ago
spiral_mxnet_1000...	14 days ago
Tabular.csv	14 days ago
test.csv	14 days ago
test.regulus	14 days ago

debug

Load data

```
[86]: data = io.load('data/gauss4')
      # data = io.load('data/simulations')
```

View Trees

```
[87]: tw1 = TreeWidget(data.tree)
[88]: v1 = show_tree(tw1, panel=False)
[89]: v2 = show_tree(tw1, attr='fitness', panel=False)
[90]: v3 = show_tree(tw1, f=v1.filter, attr='parent_fit
[91]: panel = SidePanel(title='SidePanel')
```

	x	y	z
1	0.568725	0.673814	0.614712
2	0.939274	0.940425	0.011168
3	0.732648	-0.050819	0.800305
4	0.391084	-0.990966	0.325892
5	-0.284711	0.508128	0.529962
6	0.020003	0.949783	0.29253
7	0.352876	-0.17159	0.511714
8	0.917758	0.281269	0.140347
9	0.529794	-0.51353	0.567934
10	-0.808993	0.26393	0.502102
11	-0.455708	0.581393	0.657421
12	0.956705	-0.613031	0.472095
13	0.952883	0.793236	0.224596
14	0.252758	-0.162091	0.551882
15	0.751414	-0.563098	0.881668
16	-0.444182	0.645059	0.894628
17	0.435229	0.23208	0.227206
18	-0.559365	-0.729194	0.89674
19	-0.190351	-0.409763	0.920427
20	-0.643043	-0.86913	0.777806
21	0.835454	-0.342383	0.573589
22	-0.546546	-0.48091	0.98729
23	0.430819	-0.797044	0.44224

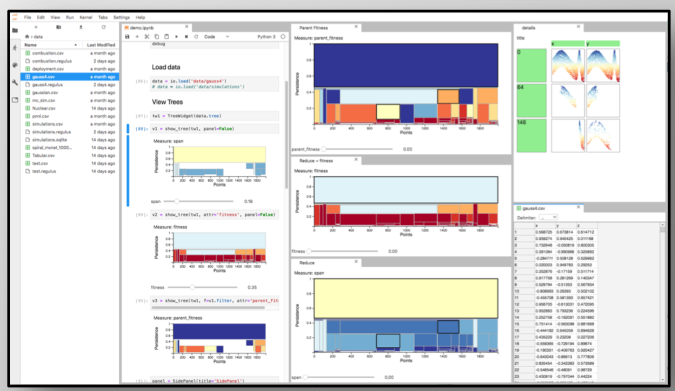
iPyRegulus

Interactive exploration

Break away from the typical visualization-first approach

Jupyter Notebooks driven investigation

- Interactive (and scripted) exploration analysis
- Define attributes and measures on the fly
- Explore data from multiple views
- Coordinate between views on the flu



Name	Last Modified
combustion.csv	a month ago
combustion.regulus	2 days ago
deployment.csv	a month ago
gauss4.csv	a month ago
gauss4.regulus	2 days ago
gaussian.csv	a month ago
mc_sim.csv	a month ago
Nuclear.csv	14 days ago
pnnl.csv	a month ago
simulations.csv	a month ago
simulations.regulus	2 days ago
simulations.sqlite	14 days ago
spiral_mxnet_1000...	14 days ago
Tabular.csv	14 days ago
test.csv	14 days ago
test.regulus	14 days ago

```
debug
```

Load data

```
[86]: data = io.load('data/gauss4')
      # data = io.load('data/simulations')
```

View Trees

```
[87]: tw1 = TreeWidget(data.tree)
[88]: v1 = show_tree(tw1, panel=False)
[89]: v2 = show_tree(tw1, attr='fitness', panel=False)
[90]: v3 = show_tree(tw1, f=v1.filter, attr='parent_fit
[91]: panel = SidePanel(title='SidePanel')
```

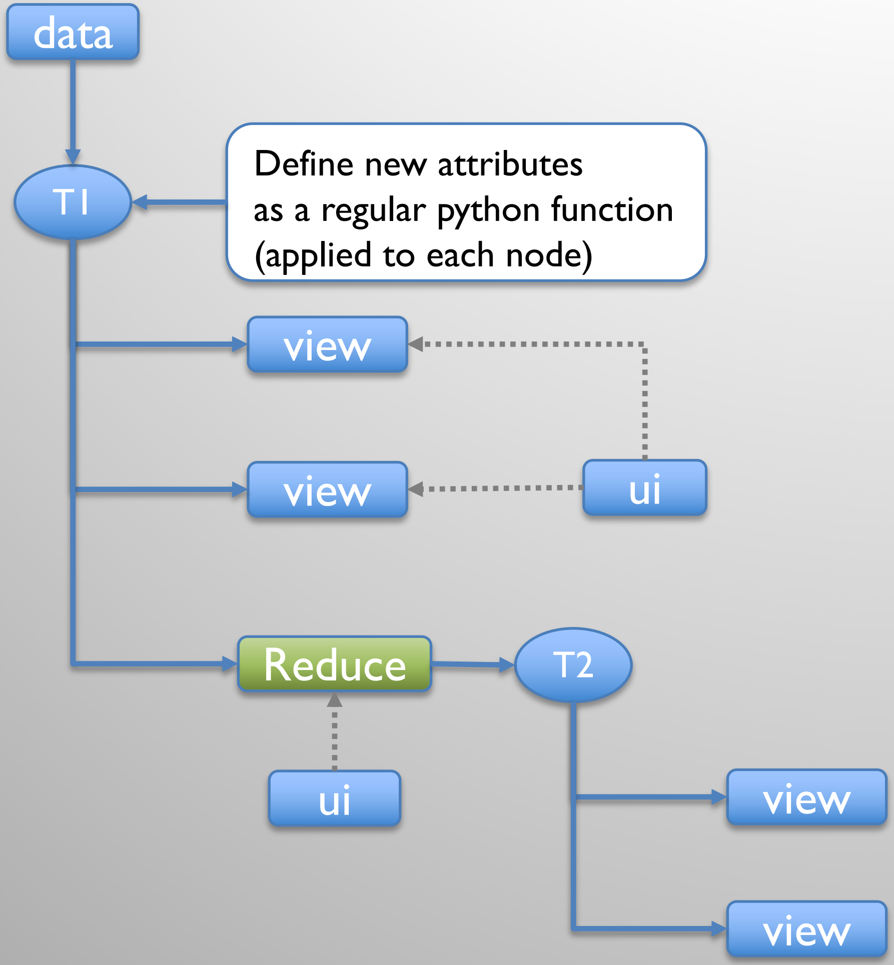
gauss4.csv

	x	y	z
1	0.568725	0.673814	0.614712
2	0.939274	0.940425	0.011168
3	0.732648	-0.050819	0.800305
4	0.391084	-0.990966	0.325892
5	-0.284711	0.508128	0.529962
6	0.020003	0.949783	0.29253
7	0.352876	-0.17159	0.511714
8	0.917758	0.281269	0.140347
9	0.529794	-0.51353	0.567934
10	-0.808993	0.26393	0.502102
11	-0.455708	0.581393	0.657421
12	0.956705	-0.613031	0.472095
13	0.952883	0.793236	0.224596
14	0.252758	-0.162091	0.551882
15	0.751414	-0.563098	0.881668
16	-0.444182	0.645059	0.894628
17	0.435229	0.23208	0.227206
18	-0.559365	-0.729194	0.89674
19	-0.190351	-0.409763	0.920427
20	-0.643043	-0.86913	0.777806
21	0.835454	-0.342383	0.573589
22	-0.546546	-0.48091	0.98729
23	0.430819	-0.797044	0.44224

iPyRegulus and Sidepanel: extensions for Jupyter Lab

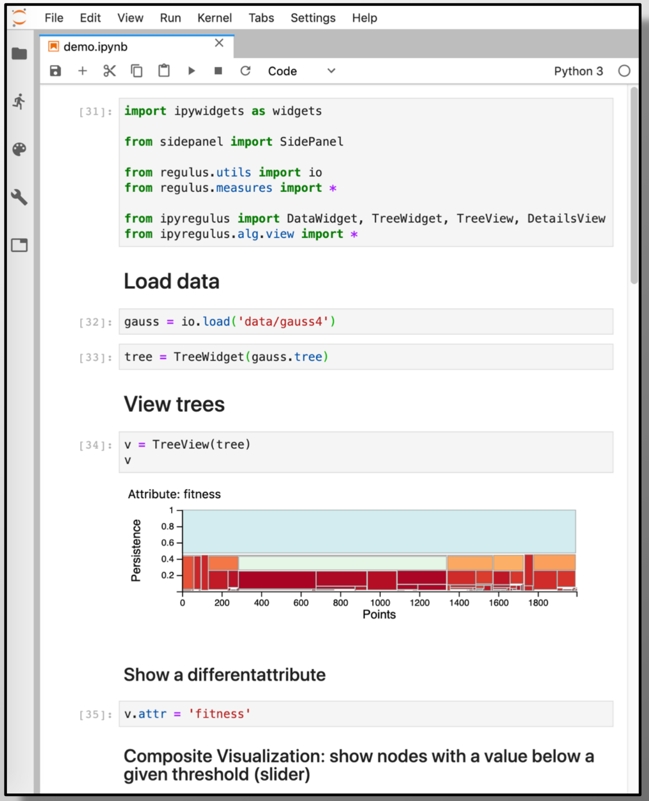
Visual Exploration

Python kernel in Jupyter



Define new attributes as a regular python function (applied to each node)

Browser



[31]:
```
import ipywidgets as widgets

from sidepanel import SidePanel

from regulus.utils import io
from regulus.measures import *

from ipyregulus import DataWidget, TreeWidget, TreeView, DetailsView
from ipyregulus.alg.view import *
```

Load data

[32]:
```
gauss = io.load('data/gauss4')
```

[33]:
```
tree = TreeWidget(gauss.tree)
```

View trees

[34]:
```
v = TreeView(tree)
v
```

Show a differentattribute

[35]:
```
v.attr = 'fitness'
```

Composite Visualization: show nodes with a value below a given threshold (slider)

iPyRegulus in action