



Technische  
Universität  
Braunschweig



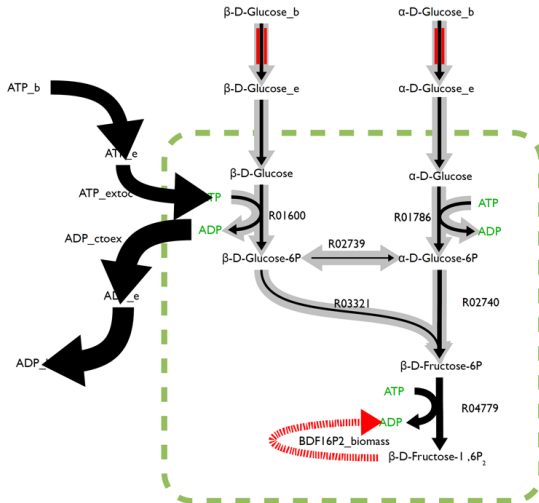
# AMEBA

## Interactive visualization of metabolic networks

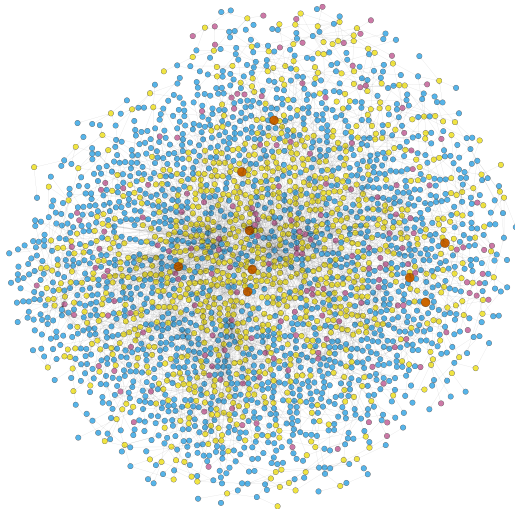
René Rex



# Include even more information: reaction fluxes

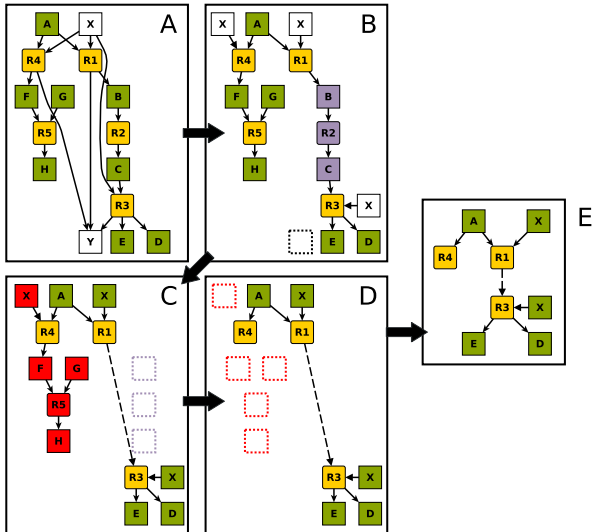


# Visualization of metabolic networks





# Metabolic context



# Graph handling

## NetworkX

- <http://networkx.github.com>
- `graph.degree(node)`
- `graph.edge[newNode][successor]["disconnected"] = True`
- output: various graph formats (e.g. GML, GraphML, dot)
- However, interactive usage would be handy...

# xdot – exactly what I needed

- <http://code.google.com/p/jrfonseca/wiki/XDot>
- an interactive viewer for graphs written in Graphviz's dot language
- can be used either as a standalone application from command line, or as a **library embedded in your python application**



# xdot – exactly what I needed

- <http://code.google.com/p/jrfonseca/wiki/XDot>
- an interactive viewer for graphs written in Graphviz's dot language
- can be used either as a standalone application from command line, or as a **library embedded in your python application**

## Features

- Since it doesn't use bitmaps it is fast and has a small memory footprint
- Arbitrary zoom, Keyboard/mouse navigation
- **Supports events on the nodes with URLs**
- Animated jumping between nodes
- Highlights node/edge under mouse

# Sample code and clickable nodes

```
def __init__(self):
    xdot.DotWindow.__init__(self)
    self.widget.connect('clicked', self.on_url_clicked)

def on_url_clicked(self, widget, url, event):
    dialog = gtk.MessageDialog(
        parent = self,
        buttons = gtk.BUTTONS_OK,
        message_format="%s clicked" % url)
    dialog.connect('response', lambda dialog,
                    response: dialog.destroy())

    dialog.run()
    return True
```

# Clickable nodes

**xdot:**

```
self.emit("clicked", unicode(url.url), event)
```

# Clickable nodes

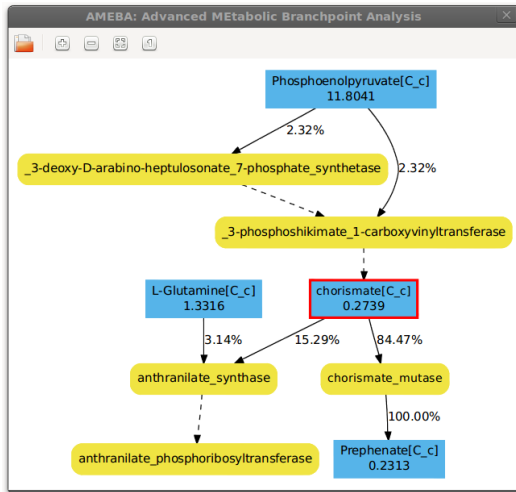
## **xdot:**

```
self.emit("clicked", unicode(url.url), event)
```

## **AMEBA:**

```
self.splitRatioGraph.node[node] ["URL"] = node  
[...]  
self.widget.connect("clicked", self._node_clicked)
```

# AMEBA: Advanced MEtabolic Branchpoint Analysis



# AMEBA: Advanced MEtabolic Branchpoint Analysis

- Interactively visualizes metabolic networks and display flux distributions
- Easily configurable to produce high-quality figures
- Available online, GPLv3, <http://metano.tu-bs.de/ameba>

# AMEBA: Advanced MEtabolic Branchpoint Analysis

- Interactively visualizes metabolic networks and display flux distributions
- Easily configurable to produce high-quality figures
- Available online, GPLv3, <http://metano.tu-bs.de/ameba>

**Thank you for your attention!**

r.rex@tu-bs.de