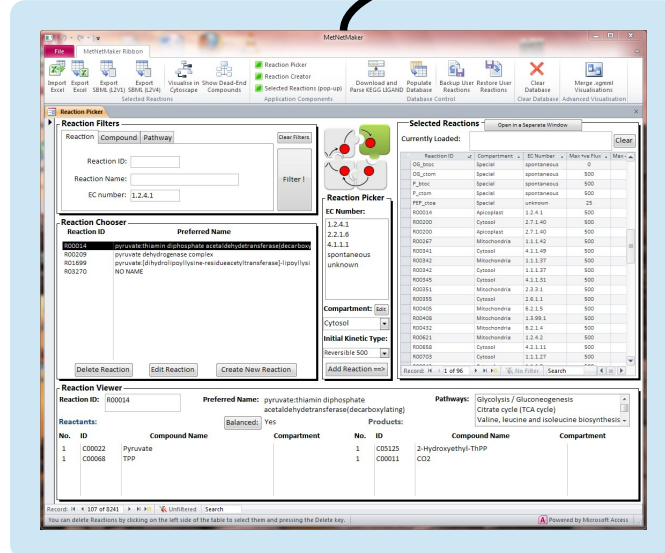# LiveFluxView

(click me to play)

LiveFluxView is my first attempt at combining everything that worked in my PhD into one web page that people can play with and understand. There's a lot of stuff missing and there are a lot of things I'd like to improve but I think it's good enough to get an idea of what I spent the four years to September 2012 doing.

Building the network

The core of LiveFluxView is a combined visual and computational metabolic network I built in my own MetNetMaker software. Techniques including raw genome annotation using SharkHunt, literature searches and the culture and measurement of malaria were combined to create this model.

malaria metabolic model within MetNetMaker

visual model showing the optimal flux distribution
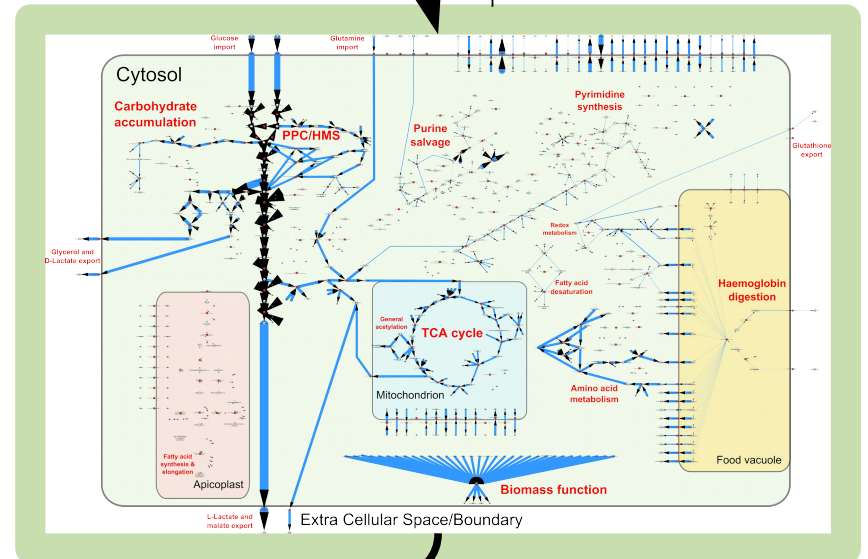
Fluxes and biomass

In order to build LiveFluxView some of the features of my full visual model were lost and the chevrons showing which direction metabolic fluxes flow through my model did not carry over to the web. In LiveFluxView, edges turn green when they carry flux in the opposite direction to the optimal solution. All other edges are weighted according to the flux that is carried by the coresponding reaction.

By displaying malaria's metabolic network visually for the first time and by letting the user play with constraints and costs, LiveFluxView gives the most intuitive understanding of malaria metabolism and its variability to date.

LiveFluxView

Constraints and costs

Constraints on the influx or efflux of metabolites are initially set at the amounts I measured by NMR metabolomics. Changing these values causes fluxes to be recalculated.

Changing cost functions causes fluxes to be recalculated.

How it works (technical)

The metabolic network at the core of LiveFluxView is built using MetNetMaker and exported in SBML format for calculations and in cytoscape format for visualisation. A 2MB SVG version of the cytoscape file is prepared for flux projection by giving edges id tags equal to their corresponding reactions. When the jQuery UI sliders are moved they collect constraints and costs which are sent by jQuery AJAX to a php handler on my server. The php handler integrates the current constraints and costs with the SBML format metabolic model sends the adapted model to SurreyFBA (native code) for flux-balance analysis. A perl script converts converts SurreyFBA's output to JSON format (<20kb) which is returned via AJAX to the browser by the php handler. Javascript in the browser projects the JSON-format data onto the visualisation by changing the line-weights of the edges with id tags corresponding to each reaction. This takes less than a second.