



**MATHEMATISCH-NATURWISSENSCHAFTLICHE  
FAKULTÄT I  
INSTITUT FÜR BIOLOGIE**

**Bachelorarbeit  
ZUM ERWERB DES AKADEMISCHEN GRADES  
BACHELOR OF SCIENCE**

“SBLink - Eine webbasierte Software zur  
Erfassung wissenschaftlicher Tabellendaten”

“SBLink - A web based tool for accessing  
scientific spreadsheet data”

vorgelegt von

Phillipp Schmidt

geb. am 06.05.1987

angefertigt in der Arbeitsgruppe Theoretische Biophysik  
am Institut für Biologie

Berlin, im August 2013

## **Zusammenfassung**

Die große Datenflut zu bewältigen, die von immer mehr verschiedenen Experimenten mit immer größerem Speicherbedarf erzeugt wird, ist eine der wichtigen Aufgaben, der sich die Wissenschaft stellen muss. In der Vergangenheit wurde dieses Thema immer wieder vernachlässigt und unterschätzt. Nun entstehen heutzutage immer mehr Projekte und Software die sich diesem Unterfangen widmen. Häufig mangelt es jedoch an der Bereitschaft von Experimentatoren und Analytikern, sich bestimmten Regeln anzupassen, die den Umgang mit Daten erleichtern sollen. Ein Grund dafür ist mit Sicherheit, dass es einfach zu wenig gute und intuitiv zu bedienende Software gibt. Das SBLink Projekt möchte einen kleinen Beitrag dazu beisteuern, indem es Benutzern die lästige Arbeit abnimmt sich von Hand durch Tabellen zu kämpfen und Datensätze zu extrahieren. Diese Datensätze können dann kuriert in einer Datenbank abgelegt werden, um einen einfachen Zugriff zu ermöglichen. Es soll ein Gegenentwurf sein zu der gängigen Methode einen Standard zu definieren und Daten nur konform zu diesem Standard zu akzeptieren. Dahinter verbirgt sich eventuell ein weiterer Grund für die geringe Beteiligung und die schwere Überzeugungsarbeit, die geleistet werden muss, um einen Benutzer von den Vorteilen zu überzeugen.

Somit besteht diese Arbeit aus der Beschreibung und der Entwicklung des SBLink Tools. Dieses ist webbasiert um betriebssystemübergreifend zu funktionieren. Damit soll die erste Hürde genommen werden, sich der Software anzunähern.

Weiterhin gibt es ein Beispiel der Anwendung um sich einem biologischen Problem zu widmen. Dabei geht es um die Frage wie performant ein systembiologisches Modell ist, wenn es außerhalb des ursprünglich vorgesehen Rahmens benutzt wird.

## **Abstract**

To cope with the data overload that is produced by diverse experiments with ever growing memory consumption is one of the big tasks science has to face. In the past this topic has often been neglected and underestimated. But nowadays many projects and software is trying to address this issue. Though, experimentalists and analysts often lack the enthusiasm to adept certain rules that are supposed to make data usage easier. Part of the problem is that there is not enough powerful and easy to use software. The SBLink project wants to throw in its share to free the users of the tedious task to manually search through spreadsheets in order to extract datasets. These datasets can be curated and saved in a database to gain easy access. It is supposed to be a alternative draft to the common method of introducing standards and only accepting data that corresponds to these standards. This might be another reason for the small participation and the high effort of persuasion that has to be put up to convince users of the advantages. Thus the thesis consists of the description and development of the SBLink tools. It is web-based to enable cross-platform usage and in this way lowering the bar of getting started with the software. Furthermore there is a usage case to address a biological problem. This will be about the question how functional a systems biology model remains when it is used outside of its intended scope.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Goals . . . . .	2
<b>2</b>	<b>Program description</b>	<b>3</b>
2.1	Tablib . . . . .	3
2.2	Web user interface - the “front end” . . . . .	3
2.2.1	Editor . . . . .	4
2.2.2	Dataset view . . . . .	6
2.3	Server - the “back end” . . . . .	7
2.4	Data extraction example . . . . .	7
<b>3</b>	<b>Data validation of new data versus an older model</b>	<b>10</b>
3.1	Biological background . . . . .	10
3.2	Data from Petelenz-Kurdziel et al. . . . .	11
3.3	Model from Zi et al. . . . .	13
3.4	Hog1PP . . . . .	14
3.5	Intracellular glycerol . . . . .	17
3.6	Gpd1 . . . . .	18
<b>4</b>	<b>Discussion</b>	<b>20</b>
4.1	Modeling results . . . . .	20
4.2	SBLink . . . . .	20
<b>5</b>	<b>Acknowledgments</b>	<b>22</b>
	<b>List of Figures</b>	<b>24</b>
	<b>Bibliography</b>	<b>24</b>

# 1 Introduction

As of now there is big demand for data handling and making scientific data publicly accessible[1]. A lot of founding and energy is spent to produce databases that try to deal with this issue like the newly emerging “Scientific Data” project of the Nature Publishing Group. Unfortunately similar projects in the past have not been used with a lot of enthusiasm[2]. Part of the problem is the difficulty of creating databases that fit the needs of many contributors, is the big variability and complexity in data structures[3]. Data that is produced from experiments mostly exists in some form of spreadsheet files. Due to the different environments and operating systems spreadsheet data comes in various formats and structures that are not easily understood without any meta information. The main problem is to extract the data from files in order to process it further. The most common way to address this problem is to establish a certain format or structure that has to be produced by the software used for experiments and in turn can be read by another program. In order to produce these formats there is a demand for standards about what information needs to be included. A project that tries to define these standards for experimental research is the Minimum Information for Biological and Biomedical Investigations (MIBBI) project[4]. Its aim is to create checklists for different types of experiments that include the minimal information needed to understand an experiment and to allow reproduction of the results. One prominent example is the Minimum Information about a Microarray Experiment (MIAME) checklist[5] that is well established and many times even required of journals to be included in scientific publications that publish mircoarray-based data. For the MIAME checklist, the two formats MAGE-TAB and MAGE-ML have been evolved[6]. Furthermore tools have been developed to assist scientists in producing data that is compliant with the respective standard like RightField[7] and ISA-Tools[8]. This way of addressing the issue can be called “Top-down”[9], because it requires a given and agreed upon structure. This will work well if this is the only kind of experimental data to be worked with and a standard has been developed for this kind of experiment. Especially in Systems Biology scientists who want to find data to validate the results of a modeling approach or create new models are often faced with a different situation. The results of three different types of experiments will most likely be in three different formats and a minimum information standard might not be given. Thus the usual work flow consists of inconvenient manual extraction of datasets with a common spreadsheet editor like Microsoft Excel or an Open Source equivalent. In order to develop a tool that can facilitate this process and leave more time to work on the actual data, the SBLink project has been initiated. In contrast

to the previously described work flow this resembles more of a “Bottom-up” approach[9]. In order to be flexible only very few assumptions are made about the spreadsheet files. So to speak, the given structure of the document determines the output. The basic work flow is described in Fig. 1.

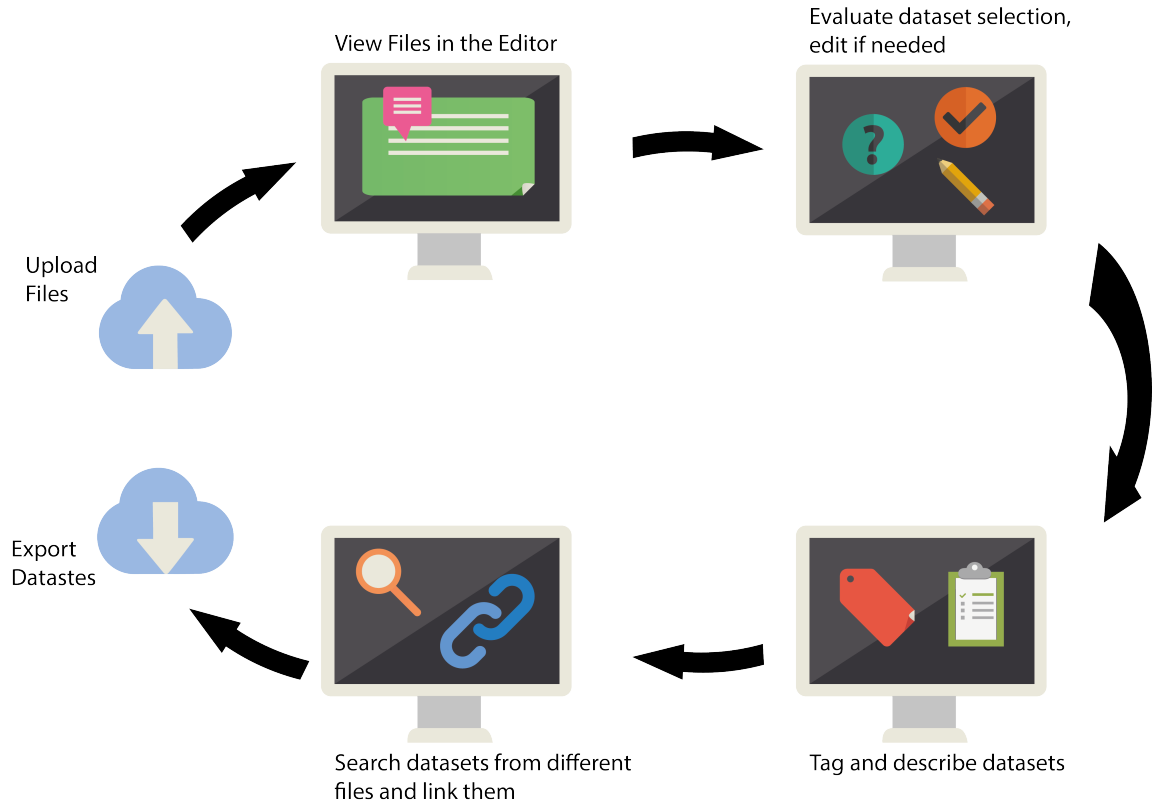


Figure 1: A scheme displaying the workflow of the SBLink tool.

## 1.1 Goals

This thesis paper is divided into two main parts. At first the development and functionality of the SBLink tool will be described. In the second part a case study about data validation will be presented to showcase the relevance of the developed tool.

## 2 Program description

Because SBLink is a web based tool, it uses a web framework in order to be displayed and used from inside a browser. The framework used is web2py [10]. It was chosen since it is programmed in Python, which is familiar to me and also a versatile and powerful programming language. Furthermore it is a well documented and maintained framework.

### 2.1 Tablib

In order to import, export and handle spreadsheet files a Python library was needed. Tablib by Kenneth Reitz [11] seemed like the best fitting candidate since it already offered exports for various formats like XLS (Excel Binary File Format), XLSX (Excel Workbook), ODS (OpenDocument Spreadsheet), CSV (Comma-separated values) and TSV (Tab-separated values). Additionally it supports the export to JSON (JavaScript Object Notation), which is the native JavaScript data format and therefore crucial since the tool is supposed to be web-based. Unfortunately it lacked the import functionality for most of them. This was implemented by me during a student project. Especially the ODS format posted some challenges, but needed to be included since it is the most common open source spreadsheet format. An ODS file is basically a zipped XML (Extensible Markup Language) document. There are Python packages that can unzip it and parse the XML. To get to the actual data it requires a navigation through the XML nodes to the corresponding cell. (See Figure 2) Based on the ODSReader class by Marco Conti [12] I developed a functional ODS parser that enables Tablib to import ODS files. Furthermore some fixes needed to be done in order to support XLS and XSLX and also the import interface was improved. The original version only included the import of file-like Python objects and I implemented an import function that uses the filename, thus improving usability for inexperienced users.

### 2.2 Web user interface - the “front end”

The web UI is based on a common HTML (HyperText Markup Language) document which is manipulated with the intensive use of JavaScript. Also the JavaScript library jQuery is used to ease the handling of click events and Ajax (Asynchronous JavaScript and XML) calls.

1	intracellular trehalose			
2	enzyme assay			
3		g D-glucose/l	per µg/ml protein	
4	time	WT	gpd1Δ	
5	0	0	0.001264271	
6	5	0		
7	10	0	0	
8	30	0.003338847	0.0044992	
9	45	0.002232344	0.005573114	
10	60	0.000581383	0.011576889	
11	90	0.000789108	0.007377231	
12	120	0.003492963	0.008095792	
13	180	0.009754982	0.011737043	
14	210	0.012005053	0.01097409	
15	240	0.014334887	0.019672615	
16	270	0.013251408		
17	1030	0.060433797	0.09289429	
18				
19	HPLC	mg/l	normalized to enzyme assay at t=30	
20	min	WT	WT	
21	-60	0.003519054	0.001630785	
22	-30	0.002638278	0.00122262	
23	-15	0.001789344	0.00082921	
24	0	0.002245063	0.001040398	
25	5	0.001554349	0.00072031	
26	10	0.002629568	0.001218583	
27	15	0.003144665	0.001457288	
28	20	0.004680841	0.002169176	
29	30	0.007204862	0.003338847	
30	45	0.006255963	0.002899113	
31	60	0.004485184	0.002078505	
32	90	0.003299301	0.001528948	
33	120	0.009625768	0.004460733	
34	180	0.015204988	0.007046233	

Figure 2: XML representation of the first four rows in the spreadsheet on the right. To get to value of the first cell the parser would traverse like this: `table-row > table-cell > p`. The letters in `text:p` are also single child nodes.

### 2.2.1 Editor

The first main part of the web user interface that was developed is the spreadsheet editor. It enables the user to view and edit their uploaded spreadsheet files. Like mentioned above it makes heavy use of Ajax [13]. This is a great concept to improve user experience. As the name already suggests it makes asynchronous calls to the server, which implies that when the JavaScript compiler encounters an Ajax call it does not execute it right away, but continues in the code. The call is then started in the background, fetching data and handing them to the browser. This makes it possible to change only parts of the web page and not reloading it each time new information is requested from the server. If the user wants to edit a previously uploaded document, the spreadsheet will be requested from the server and displayed on the web page. In order to get a nice table display a JavaScript plug-in called DataTables [14] is used. If the file is in one of the formats XLS, XSLX or ODS and it has multiple sheets they can be accessed by selecting tabs above the table. In Fig. 3 the sheet “wild type gpd2Δ gpd1Δ” is selected. As a first measure of identifying datasets each spreadsheet is scanned for



Readme wild type gpd2Δ gpd1Δ wild type pfk26Δ27Δ 0.4M wild type pfk26Δ27Δ 0.8M



Show 10 entries

	Column 1	Column 2	Column 3	Column 4
Row 1	Measurements	689	689	avg
Row 2	0	0.08875888849664633	0.024056353003721425	0.05640762075018388
Row 3	2	1	0.8191121743753322	0.9095560871876661
Row 4	4	0.07050000000000000	0.4706273258904838	0.7220754764507137
Row 5	10	0.305613	1	0.8662564930152806
Row 6	15	0.626355	0.9851143009037746	0.7877561014832051
Row 7	20	0.705986	0.5009303561935141	0.39893459158205635
Row 8	25	0.6989259	0.20401382243487506	0.15455620770238382
Row 9	30	0.302335	0.002259436469962786	0.03713758550009814
Row 10	40	0.0580463	0.037745879851143006	0.03604475897847382
Row 11	50	0.05274598	0.01661350345560872	0.017081778204177347
Row 12	60	0.020374199404912	0.0027910685805422647	0.011582633992727132
Row 13	90	0.026778960108931365	-0.0011961722488038277	0.012791393930063769
Row 14	120	0.030763023853951285	0.0014619883040935672	0.016112506079022426
Row 15				

Pop-up window (0,7):

- Orientation: ↑ ↔
- ID: 7928
- Header: Measurements
- Group: 7940
- Tags: measure gpd2 gpd1
- Enter Tags...

Figure 3: Editor view with various features on display

strings and numbers. Strings are by default identified as possible headers (red) and numbers are identified as data (green). Afterwards the sheet is scanned again for connected datasets. If a dataset has a string on top it will be attached to the dataset as a header, as can be seen in Fig. 3 with the first dataset having “Measurements” assigned as header. Datasets found in the sheet are then send to the server and database IDs are returned, e.g. the first dataset with the ID 7928. Other features are visible in Fig. 3. In order to keep the user interface as easy and intuitive as possible, most functions were put in a window that pops up when the user clicks on a dataset. It gives the option to change the orientation of datasets in this sheet and displays ID, header and tags of the corresponding dataset. Furthermore it will show if the dataset is grouped with any other dataset. Grouping is possible with the magnet icons  displayed on top of the other datasets. If clicked the dataset will also be highlighted and listed as a partner. This selection is explicit and unidirectional. In Fig. 3 the first dataset is grouped with the third dataset (ID 7940), indicated by the highlight and also the remove icon  which makes it possible to ungroup this dataset. Another great feature that can be accessed from the pop-up window is the ability to add tags to a dataset. Tags are another mean of identifying datasets besides the header

and the position. Later on datasets can be filtered on the basis of their tags. The other click icons  $\oplus$  and  $\ominus$  trigger functions for adding and removing cells from a dataset. If a dataset does not contain a string header, there is the possibility to turn the first cell of this dataset in to a header.

### 2.2.2 Dataset view

After files have been sufficiently edited the user can switch to the dataset view, although this does not mean that the sheets cannot be edited anymore. In fact, the user can switch back and forth between the dataset view and the editor as many times as needed. To get to a certain dataset it only requires the right selection of the file, sheet and dataset. When selected, grouped datasets are highlighted with a frame. Further more informations like the header and tags are displayed. This view also functions as the control center for file management. They can be deleted and the editor can be called. Datasets also have an edit function which shows the editor, scrolls to the right dataset and selects it.

Furthermore the dataset view includes a filter function that searches dataset

File	Sheet	Set	Data
417 D2.xls	Readme	13 0 - [0, 13] Measurements	Main
531 D5.xls	wild type gpd2Δ gpd1Δ	13 1 - [1, 13] 689	Measurements
24 D6.xls	wild type pfk26Δ27Δ 0.4M	13 2 - [2, 13] 689	measure gpd2 gpd1
19 D7.xls	wild type pfk26Δ27Δ 0.8M	13 3 - [3, 13] avg	0
6 D3.xls		14 1,3 - [1, 43]	2
563 D1.xls		14 2,3 - [2, 43]	4
30 D4.xls		13 4 - [4, 13] SD	10
		13 0,13 - [0, 43]	15
		10 0,19 - [0, 100]	20
		13 33 - [3, 43] avg	25
		13 34 - [4, 43] SD	30
			40

Figure 4: Structure of the dataset view: File → Sheet → Datasets → Data

headers and tag lists and shows the results in order of match quality. To improve the user experience a “fuzzy” search algorithms is implemented that shows results that are similar to the query and does not require the exact term. To determine how “fuzzy” the search is supposed to be, a slider is

placed next to the input field which translates to values between 0 and 100 and in this way setting a threshold for the search algorithm. (See Fig. 5) In a later version it is possible to choose whether the filter searches the term in header, tags or both.

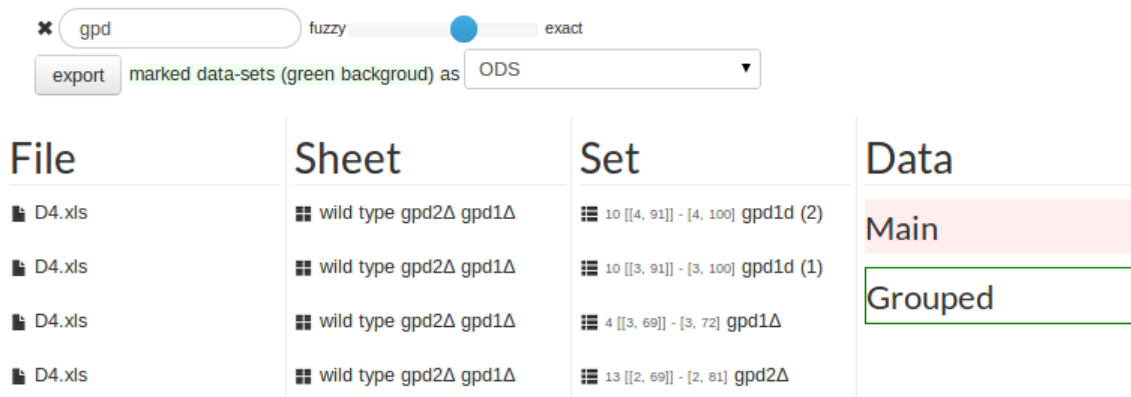


Figure 5: A search for the term “gpd” shows datasets with matching header.

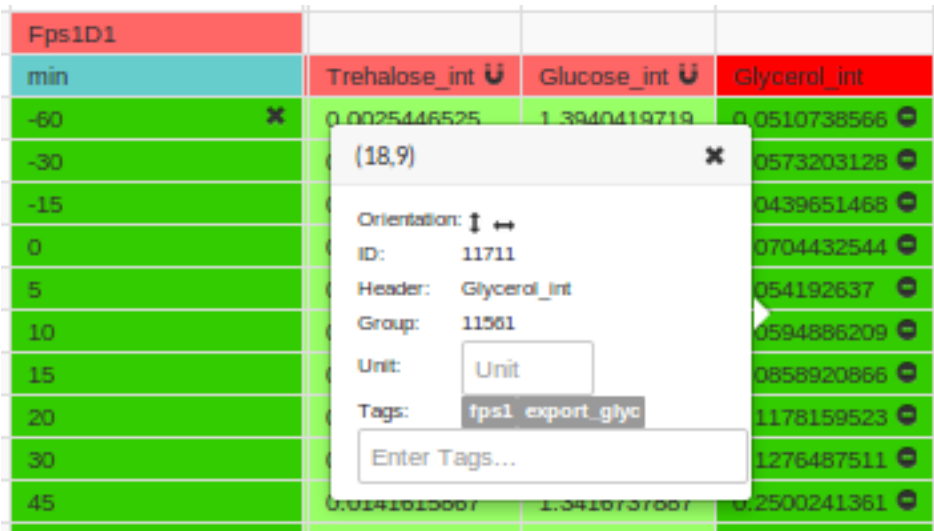
## 2.3 Server - the “back end”

The web2py framework consists of Python scripts on the server side. There is a script to manage the two databases for data files and datasets. The controller script processes all the data sent to the server via Ajax. It inserts information in the according databases and makes them accessible for later use. It also holds functions to return information that is extracted from the database upon certain requests. The search algorithm is implemented here as well and makes use of the “fuzzy wuzzy” library [15].

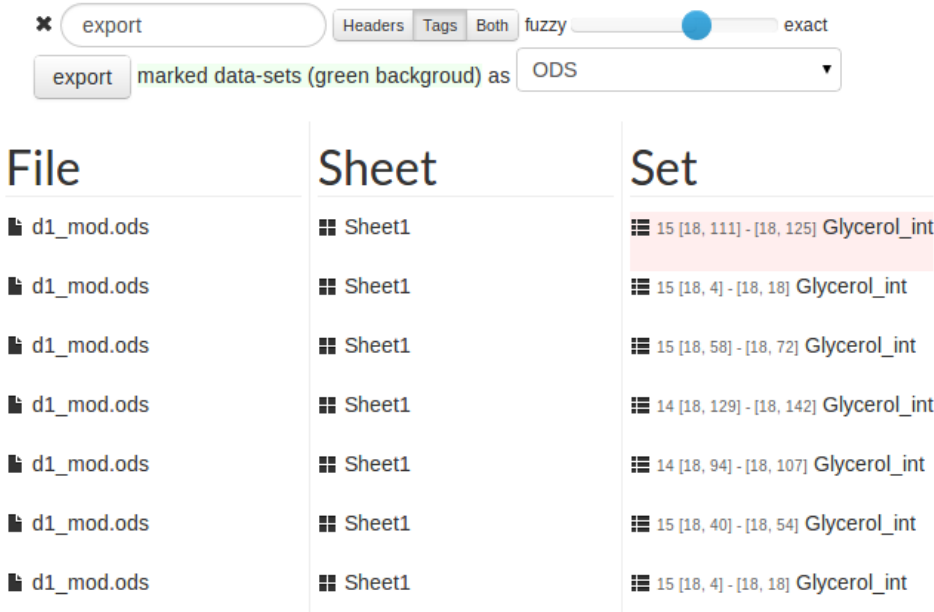
## 2.4 Data extraction example

To demonstrate the abilities of the tool I was in the need for sample files and Clemens Kühn was so kind to provide me with data from his last publication [16]. It deals with the reaction of yeast cells to hyper osmotic stress. This will be explained in detail in section 3. The supplement included seven XLS files with most of them having multiple sheets. In addition they are rather large spreadsheets which is also a good way of testing the performance. One file includes all the raw and processed data where datasets need to be extracted. An important condition for the analysis is the concentration of intracellular glycerol which depicts a metabolic reaction to the stress. In order to extract the datasets from the raw data spreadsheet they need to be selected, grouped

with their time courses and tagged. When all the desired sets are curated they can be viewed in the dataset view. Since these are large files which contain a lot of datasets, applying the filter function is very convenient. As can be seen in Fig. 6a the datasets were given a “export\_glyc” tag, which makes it easy to filter the results in the dataset view. See Fig. 6b.



(a) Selection of the *FPS1-Δ1* strain with time course and tags



(b) Dataset view with filtered results for the term “export”

Figure 6: Dataset extraction process

Afterwards datasets can be selected and exported for further processing or plotting. During the export certain properties like filename, sheet name and tags are preserved and will be visible in the exported file. This file is ready for further processing or plotting. An example export is shown in Fig. 7.

	A	B	C	D	E	F
1	Sheet1 (d1_mod.ods)		Sheet1 (d1_mod.ods)		Sheet1 (d1_mod.ods)	
2	nostress,exportglyc		hog1att,exportglyc		fps1d1,exportglyc	
3	Glycerol_int(M)	(min)	Glycerol_int(M)	(min)	Glycerol_int(M)	(min)
4	0.0718203538	-60	0.1133060888	-60	0.0510738566	-60
5	0.0588592599	-30	0.0825916255	-30	0.0573203128	-30
6	0.0626702236	-15	0.0852316819	-15	0.0439651468	-15
7	0.052373603	0	0.0760407673	0	0.0704432544	0
8	0.0548595198	5	0.113033573	5	0.054192637	5
9	0.0520113954	10	0.1595322727	10	0.0594886209	10
10	0.0520025638	15	0.1811971493	15	0.0858920866	15
11	0.0714009442	20	0.2599785185	20	0.1178159523	20
12	0.0439651468	30	0.4520982118	30	0.1276487511	30
13	0.0435156848	45	0.6972295775	45	0.2500241361	45
14	0.0473367665	60	0.8432914415	60	0.3645307873	60
15	0.0368837608	90	0.9192566478	90	0.4114584613	90
16	0.0418338085	120	0.9661210559	120	0.5132389426	120
17	0.0359340071	180	0.8474401033	180	0.7449191834	180
18			0.7702913442	202	0.6465360113	221

Figure 7: Export file including three datasets. Above the data are the preserved meta information like filename, tags, units and headers. Next to each dataset are the grouped time courses.

### 3 Data validation of new data versus an older model

In 2010 Zi et al. created a model for studying the Hog1 MAPK response to different forms of osmotic stress[17]. In 2013 Petelenz-Kurdziel and Kühn et al. gathered more scientific data for this process and created another model to fit the data[16]. Part of the data was used in section 2.4. Naturally the question arose whether the new data gathered in [16] would fit the old model from [17]. I used the SBLink tool to extract the relevant data from the supplementary files included in [16] in the same way as described in section 2.4.

#### 3.1 Biological background

The reaction to hyper-osmotic stress in the yeast *S. cerevisiae* is a well studied mechanism and the high-osmolarity glycerol (HOG) mitogen-activated protein kinase (MAPK) signaling pathway is part of the reaction[18]. Upon osmotic stress yeast cells reduce size within seconds. Within less than 1 minute the glycerol export channel Fps1 closes to prevent glycerol leakage[19]. Parallel, the HOG signal transduction system is activated. The initial responses mediated by the active Hog1 include stimulation of ion export, pausing of the cell cycle, reducing of translational capacity and stimulation of glycolysis to enhance production of glycerol. When activated this pathway triggers the accumulation of glycerol to counter the high osmolarity outside of the cell. An additional way of enhancing the glycerol production is the NAD<sup>+</sup>-dependent glycerol 3-phosphate dehydrogenase encoded by the isogenes GPD1 and GPD2 whose expression is also stimulated through Hog1[20] resulting in a transcriptional feedback. A diagram of the pathway is visible in Fig. 8.

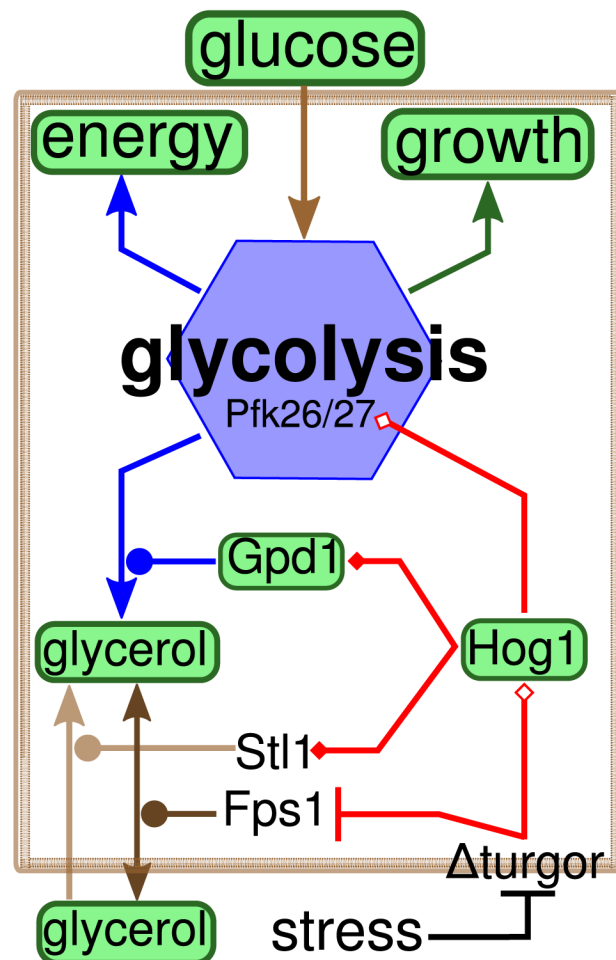


Figure 8: Diagramm of the HOG pathway. Figure and caption taken from [16]. Arrows indicate mass flow, diamonds indicate positive regulation (empty: direct, filled: gene expression), circles indicate catalysis, bars indicate inhibition. Measured entities are highlighted green. Processes are colored according to the different modules (transport: brown, glycolysis: blue, growth: green, adaptation: red).

### 3.2 Data from Petelenz-Kurdziel et al.

The collected data included measurements for the concentration of phosphorylated Hog1 (Hog1PP), Gpd1 and intracellular glycerol. Besides the wild type different mutant strains were used that were missing some adaptation mechanisms. The single mutations are:

- Hog1 $\Delta$  and Gpd1 $\Delta$ : Hog1 and gpd1 genes are knocked out, respectively
- Hog1-att: Hog1 is attached to the plasma membrane and therefore

cannot enter the nucleus, which results in a missing Hog1-dependent gene expression response

- Fps1- $\Delta$ 1: missing stress dependent closure of Fps1

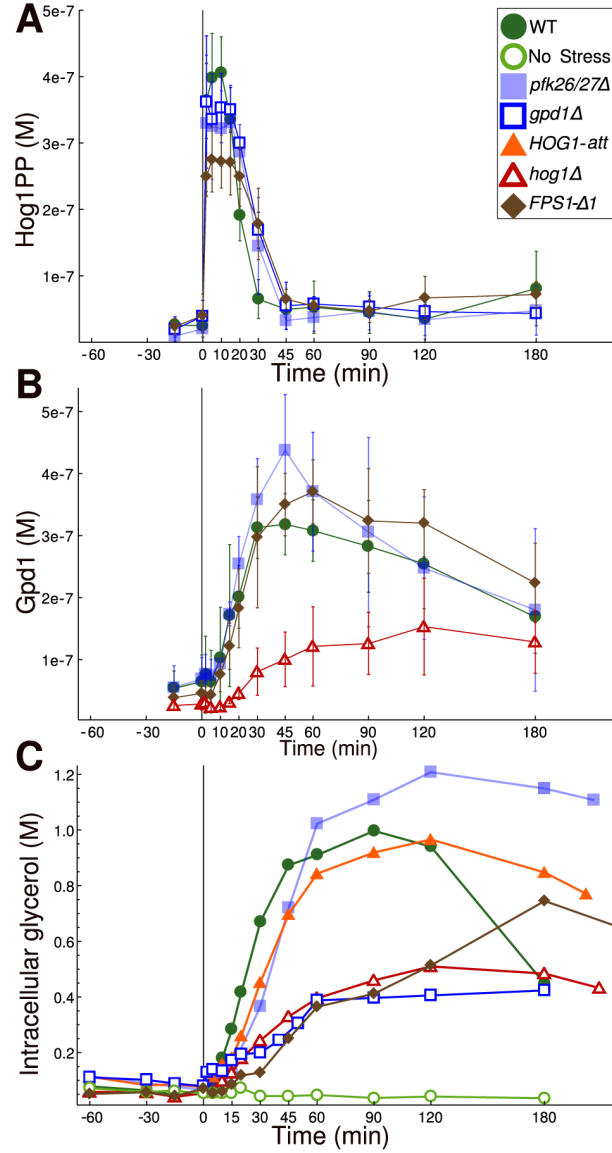


Figure 9: Time course data. Figure taken from [16]



### 3.3 Model from Zi et al.

The model has a high resemblance with the one used by Petelenz-Kurdziel et al. [16] and therefore should be able to produce similar results. It does not include Gpd1 as a single species like the other model but rather a function  $Yt$  that represents Gpd1 and that cascades over  $N = 4$  different additional states  $z1 - z4$  in order to resemble an observed time delay  $\tau$ . It is represented by the following functions.

$$\begin{aligned}\frac{d[z1]}{dt} &= \frac{N \cdot ([Hog1PPn] - [z1])}{\tau} \\ \frac{d[z2]}{dt} &= \frac{N \cdot ([z1] - [z2])}{\tau} \\ \frac{d[z3]}{dt} &= \frac{N \cdot ([z2] - [z3])}{\tau} \\ \frac{d[z4]}{dt} &= \frac{N \cdot ([z3] - [z4])}{\tau} \\ \frac{d[Yt]}{dt} &= K_{s0}^{Yt} + K_{s1}^{Yt} \cdot [z4] - K_t^{Yt} \cdot [Yt]\end{aligned}$$

This state is later used to compare to the data for Gpd1. In the next step the different mutants needed to be included in the model. This was mostly straight forward and intuitive by removing any reactions that involved the knocked out genes. It was especially convenient that the model already contained separate species for Hog1 and Hog1PP inside the nucleus and the cell, respectively. This made it possible to simulate the Hog1-att mutant by removing Hog1 and Hog1PP just inside of the nucleus. A graphical representation of the model is shown in Fig. 10. In the following sections the time course data from Fig. 9 is compared to the data resulting from modeling the different mutants.

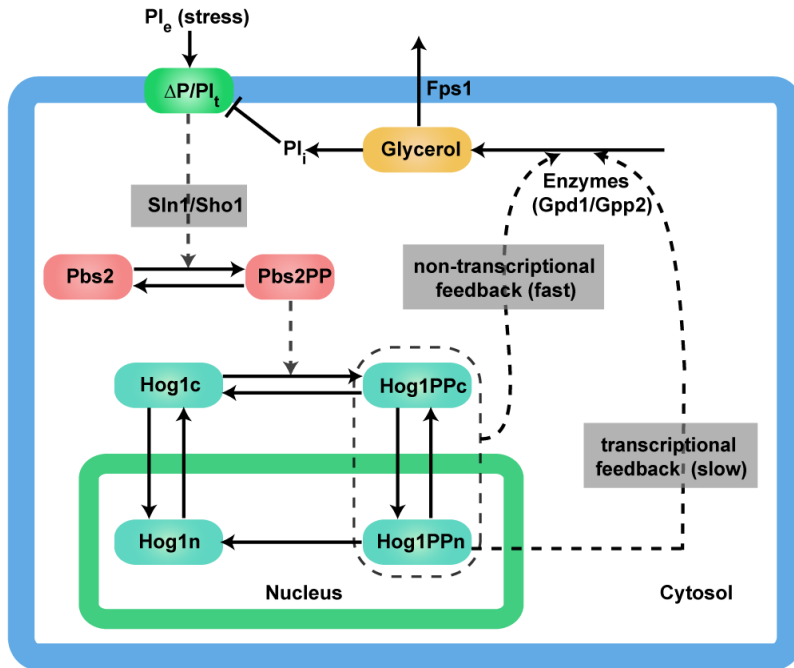


Figure 10: Model representation from Zi et al. Figure and caption taken from [17] Scheme of the model for osmosensing network. “Non-transcriptional feedback loop” denotes the Hog1 kinase dependent regulation of glycerol production. “Transcriptional feedback loop” stands for transcriptional regulation of the enzymes responsible for glycerol production. The gray boxes are modeled with coarse-grained black box approaches.

### 3.4 Hog1PP

From the five investigated strains only the wild type,  $\text{Gpd1}\Delta$  and  $\text{Fps1-}\Delta$ 1 are suitable for observing the concentration of Hog1PP because in the other two strains it is either completely removed or not present in the nucleus. In Fig. 11 the three simulated Hog1PP concentrations are shown. Only when returning to the steady state after about 20 minutes the curves start to differ much. Prolonged phosphorylation of Hog1 can be observed in the  $\text{Gpd1}\Delta$  strain, what is also visible in the data.

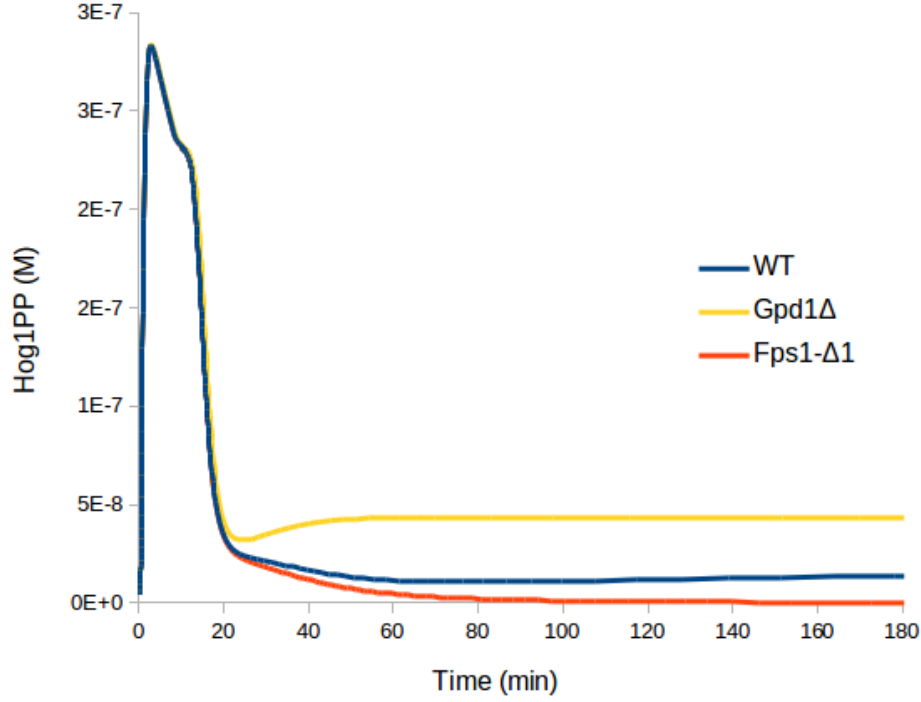


Figure 11: The three simulated Hog1 responses.

When compared to the data the model proves that is able to predict tendencies very well (See Fig. 12). Apart from slight differences in amplitude the simulations perform satisfactorily. Especially the  $Gpd1\Delta$  model is able reach almost exactly the same steady state. To a certain extent this could have been expected of the model since the primary focus was put on reproducing Hog1PP concentration time courses.

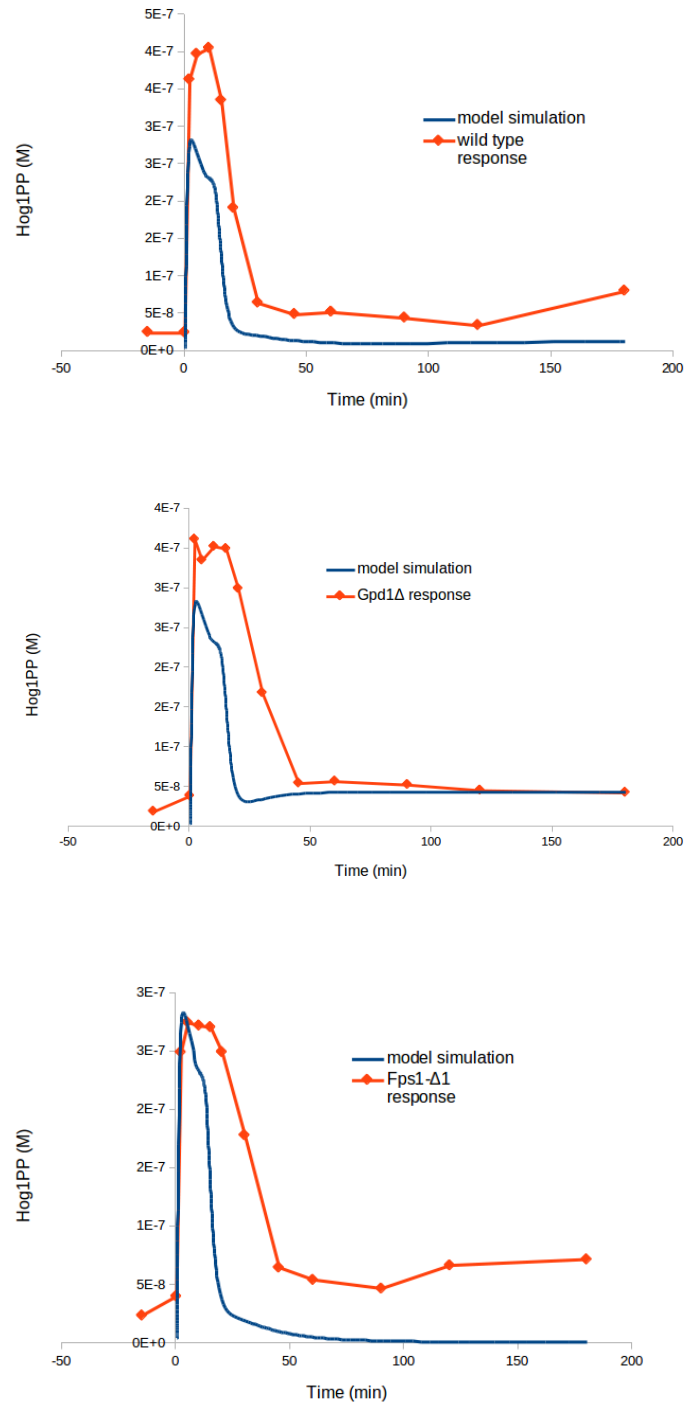


Figure 12: The three simulated Hog1PP responses compared to the experimental data.

### 3.5 Intracellular glycerol

For intracellular glycerol this is a different story. It was not one of the species used for parameter estimation when the model was created and experimental data was only provided for showing how the glycerol changes when exposed to dynamic pulses (eg. square pulses). Glycerol concentration changes for a constant pulse were not investigated. Furthermore the initial concentration for intracellular glycerol is almost ten times lower in the experimental data. This is due to restrictions during the experiment.

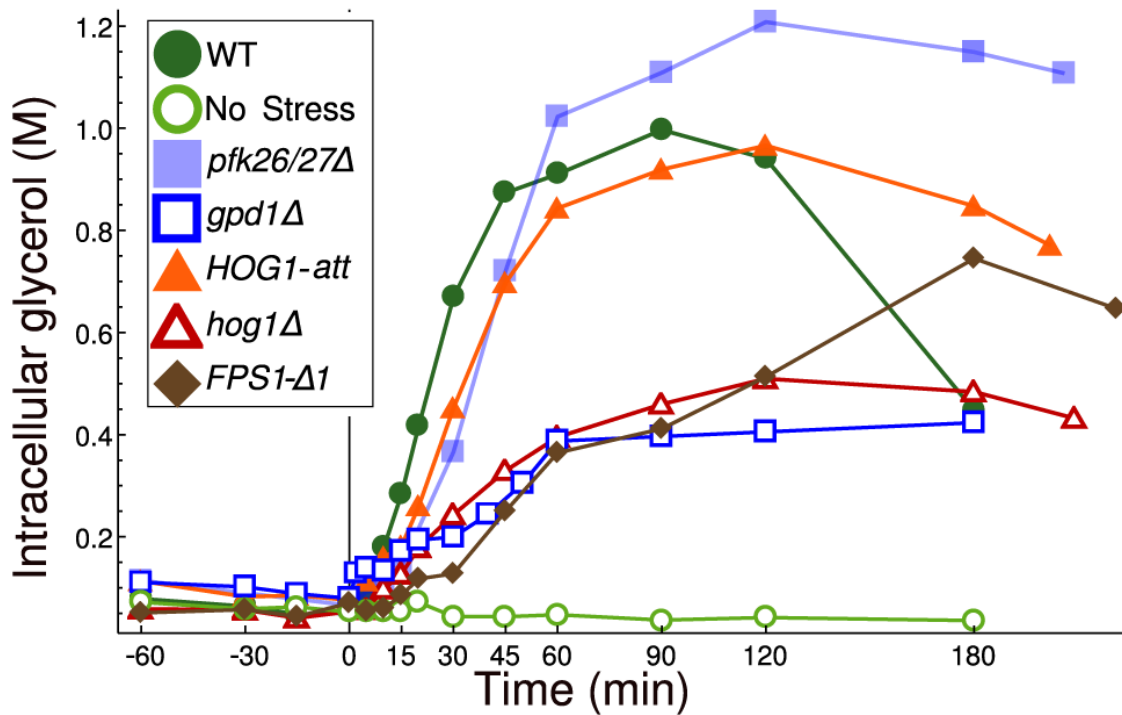


Figure 13: A closer look at the glycerol concentration time courses. Figure taken from [17]

When comparing the figures 13 and 14 it is obvious that there is only a minor resemblance between some time courses. Apparently the model cannot reproduce the decay of intracellular glycerol. Yet some tendencies are similar. The character of the time course for the *hog1Δ* and the resulting disadvantage in glycerol accumulation is clearly visible. Also the wild type and the *HOG1-att* strain show similar positive tendencies which can also be observed in the data. On the other hand it seems like the *gpd1Δ* and *FPS1-Δ1* strains are performing better in the simulation. The comparisons of each strain with its simulated counterpart would not be worthwhile since the progressions are

just too different.

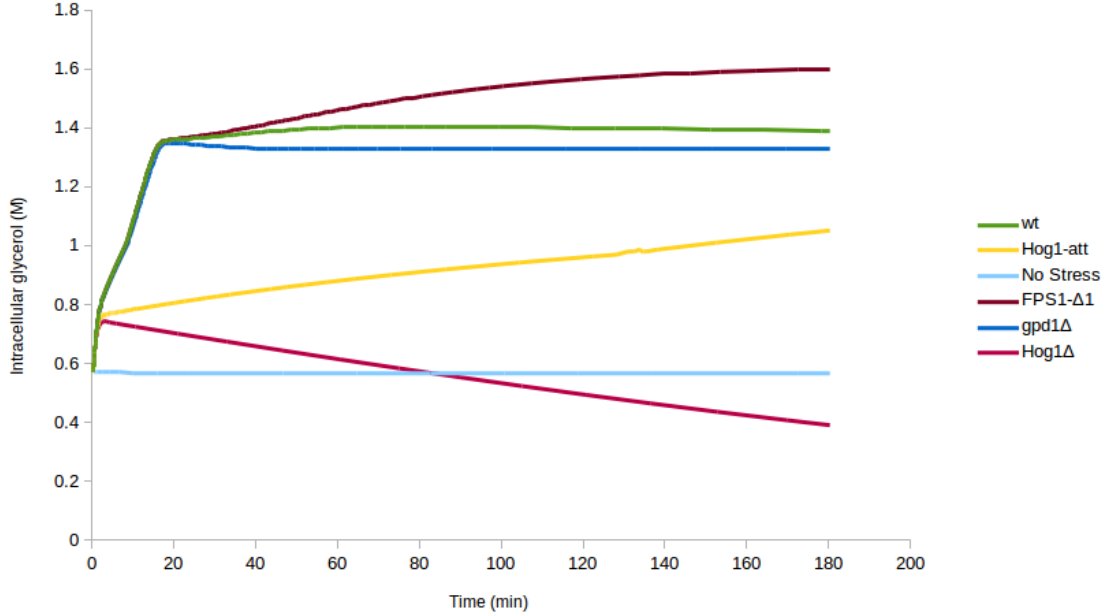


Figure 14: Glycerol concentration time courses for simulated strains.

### 3.6 *Gpd1*

For the *Gpd1* concentration only two strains are worth looking at since *Gpd1Δ*, *Hog1Δ* and *Hog1-att* all lead to *Gpd1* not being expressed. Although in [16] some expression for the *Hog1Δ* strain was observed. To be in accordance with this observation a partial *Hog1*-independent increase of *Gpd1* mRNA was implemented in the new model approach.

In Fig. 15 the two strains that were simulated produced a well fitting time course. The difference in amplitude can be accounted for by the way of implementing the  $Yt$  function as a placeholder for *Gpd1*.

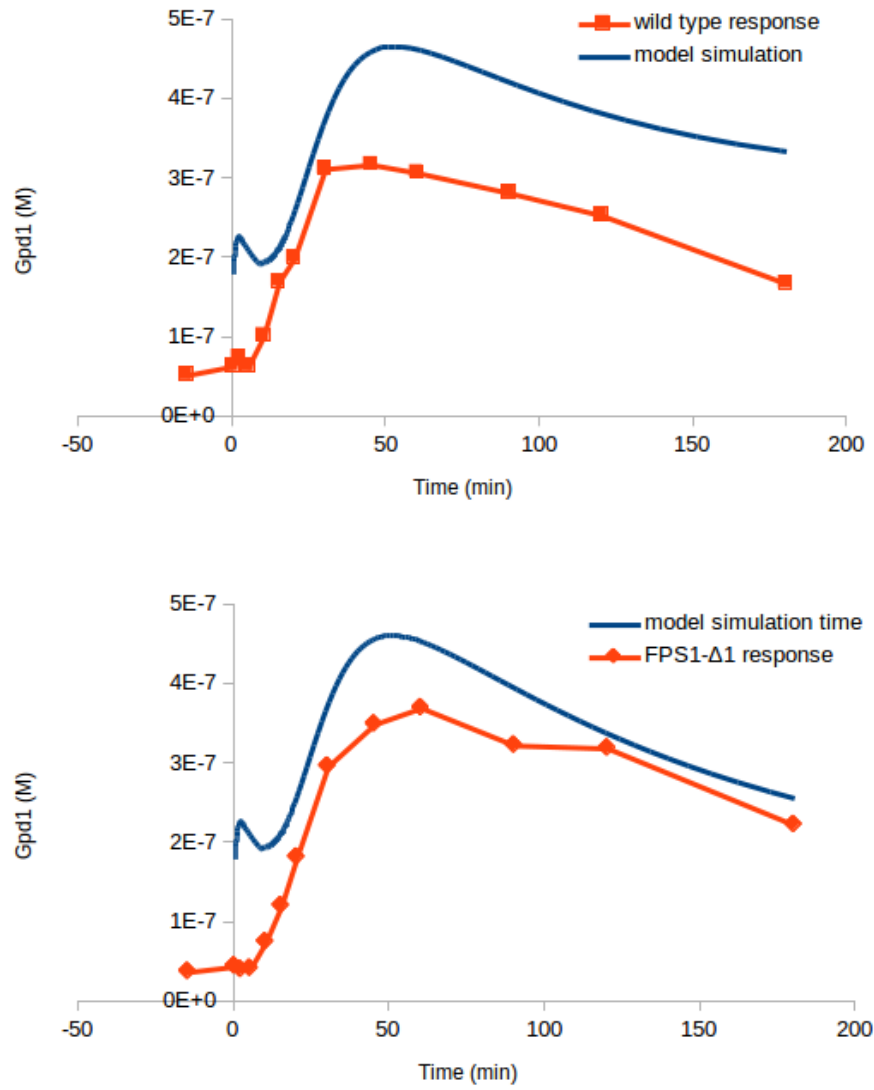


Figure 15: Two simulated Gpd1 responses compared to the experimental data.

## 4 Discussion

### 4.1 Modeling results

The results of the data validation show that model created by Zi et al. is able to reproduce some of the new findings. Especially the results for the Hog1PP and Gpd1 concentration time courses fitted nicely. The problems with fitting the model to the glycerol data might have different reason. One aspect is the longer time period. In the original publication most experiments and simulations would not have run longer then 60 to 100 minutes. Unfortunately this is the exact time period were the glycerol levels start to fall. Also the different initial concentration is a problem. Still it is comforting to see that a model created for a specific problem can also be able to reproduce unintended results. While the model was originally created to simulate the effects of fluctuating osmotic stress, it is also manages to simulate strains with knocked out adaption mechanisms. To improve the fits some mechanisms implied in [16] would be worthwhile to incorporate. As mentioned before another function could be added to account for the Gpd1 levels in the Hog $\Delta$ 1 strain. Other suggested methods include osmodependent increase of Gpd2 transcription and the regulation of biomass production to account for other adaption mechanism that are not included in the HOG pathway. A regular parameter estimation step could also solve problems, but it defeat the purpose in a certain way. It is definitely a good example for the possibilities of mathematical modeling when the right tools are used and the right questions are asked.

### 4.2 SBLink

It could be argued that the functionality of the tool is not very appealing and it could easily be achieved by doing it manually, but that would not be very farsighted. The tool has only been in development for a couple month and it should be ensured that certain features work properly before extending functionality and loosing the scope. It is a very simple but different approach to this data assessment problem. For large quantities of data and for regularly used experiments, introducing standards is the method of choice and should always be used if applicable. To be able to choose would be ideal because then the user could decide which approach suits his current problem best. For example an on-line database were the user can choose from any existing standards to include his or her data and if that is not the case have a simple to use data curation tool like SBLink.

Another idea that was obvious during the development was a function to



create templates from an already detected spreadsheet structure that would have to be curated once and then each file of a similar structure could quickly be accessed. Furthermore the original idea to upload models as well and link data to the model species should be the goal.

## **5 Acknowledgments**

I would like to thank Prof. Dr. Dr. h.c. Edda Klipp for her counsels and for being a great advisor and Falko Krause for being an awesome mentor and for introducing me to the world of web development. Also the tbp group for their support and readiness to answer my questions, especially Timo Lubitz and Jannis Uhlendorf. My thanks to Dr. Clemens Kühn for giving me insight in the HOG pathway and having great ideas for adjusting the model. My special thanks to Prof. Andreas Möglich, institute of biophysics for reviewing yet another bachelor thesis. Great thanks to my girlfriend Rosa and my family for their support and help during the work, especially in the final hours.

## Technical Abbreviations

Ajax	Asynchronous JavaScript and XML
CSV	Comma-separated values
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
ODS	OpenDocument Spreadsheet
TSV	Tab-separated values
XLS	Excel Binary File Format
XLSX	Excel Workbook
XML	Extensible Markup Language

## List of Figures

1	A scheme displaying the workflow of the SBLink tool. . . . .	2
2	XML representation of the first four rows in the spreadsheet on the right. To get to value of the first cell the parser would traverse like this: <code>table-row &gt; table-cell &gt; p</code> . The letters in <code>text:p</code> are also single child nodes. . . . .	4
3	Editor view with various features on display . . . . .	5
4	Structure of the dataset view: File → Sheet → Datasets → Data . . . . .	6
5	A search for the term “gpd” shows datasets with matching header. . . . .	7
6	Dataset extraction process . . . . .	8
7	Export file including three datasets. Above the data are the preserved meta information like filename, tags, units and headers. Next to each dataset are the grouped time courses. . . . .	9
8	Diagramm of the HOG pathway. Figure and caption taken from [16]. Arrows indicate mass flow, diamonds indicate positive regulation (empty: direct, filled: gene expression), circles indicate catalysis, bars indicate inhibition. Measured entities are highlighted green. Processes are colored according to the different modules (transport: brown, glycolysis: blue, growth: green, adaptation: red). . . . .	11
9	Time course data. Figure taken from [16] . . . . .	12
10	Model representation from Zi et al. Figure and caption taken from [17] . . . . .	14
11	The three simulated Hog1 responses. . . . .	15
12	The three simulated Hog1PP responses compared to the experimental data. . . . .	16
13	A closer look at the glycerol concentration time courses. Figure taken from [17] . . . . .	17
14	Glycerol concentration time courses for simulated strains. . . . .	18
15	Two simulated Gpd1 responses compared to the experimental data. . . . .	19

## Bibliography

- [1] Myles Axton. No second thoughts about data access. *Nature genetics*, 43(5):389, May 2011.

- [2] Bryn Nelson. Data sharing: Empty archives. *Nature*, 461(7261):160–3, October 2009.
- [3] Myles Axton. It’s not about the data. *Nature genetics*, 44(2):111, February 2012.
- [4] Chris F Taylor, Dawn Field, Susanna-Assunta Sansone, Jan Aerts, Rolf Apweiler, Michael Ashburner, Catherine A Ball, Pierre-Alain Binz, Molly Bogue, Tim Booth, Alvis Brazma, Ryan R Brinkman, Adam Michael Clark, Eric W Deutsch, Oliver Fiehn, Jennifer Fostel, Peter Ghazal, Frank Gibson, Tanya Gray, Graeme Grimes, John M Hancock, Nigel W Hardy, Henning Hermjakob, Randall K Julian, Matthew Kane, Carsten Kettner, Christopher Kinsinger, Eugene Kolker, Martin Kuiper, Nicolas Le Novère, Jim Leebens-Mack, Suzanna E Lewis, Phillip Lord, Ann-Marie Mallon, Nishanth Marthandan, Hiroshi Masuya, Ruth McNally, Alexander Mehrle, Norman Morrison, Sandra Orchard, John Quackenbush, James M Reecy, Donald G Robertson, Philippe Rocca-Serra, Henry Rodriguez, Heiko Rosenfelder, Javier Santoyo-Lopez, Richard H Scheuermann, Daniel Schober, Barry Smith, Jason Snape, Christian J Stoeckert, Keith Tipton, Peter Sterk, Andreas Untergasser, Jo Vandesompele, and Stefan Wiemann. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nature Biotechnology*, 26(8):889–896, 2008.
- [5] A Brazma, P Hingamp, J Quackenbush, G Sherlock, P Spellman, C Stoeckert, J Aach, W Ansorge, C A Ball, H C Causton, T Gaasterland, P Glenisson, F C Holstege, I F Kim, V Markowitz, J C Matese, H Parkinson, A Robinson, U Sarkans, S Schulze-Kremer, J Stewart, R Taylor, J Vilo, and M Vingron. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature Genetics*, 29(4):365–371, 2001.
- [6] Edda Klipp and F Krause. Computational Tools for Systems Biology. *Cancer Systems Biology, Bioinformatics and Medicine*, 2011.
- [7] Katy Wolstencroft, Stuart Owen, Matthew Horridge, Olga Krebs, Wolfgang Mueller, Jacky L Snoep, Franco du Preez, and Carole Goble. Right-Field: embedding ontology annotation in spreadsheets. *Bioinformatics (Oxford, England)*, 27(14):2021–2, July 2011.
- [8] Philippe Rocca-Serra, Marco Brandizi, Eamonn Maguire, Nataliya Sklyar, Chris Taylor, Kimberly Begley, Dawn Field, Stephen Harris, Winston Hide, Oliver Hofmann, Steffen Neumann, Peter Sterk, Weida

- Tong, and Susanna-Assunta Sansone. ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics (Oxford, England)*, 26(18):2354–6, October 2010.
- [9] Carma L. McClure. Top-down, bottom-up, and structured programming. *IEEE Transactions on Software Engineering*, SE-1(4):397–403, December 1975.
- [10] M. DiPierro. web2py Web Framework. \url{http://www.web2py.com/}.
- [11] Kenneth Reitz. Tablib. \url{http://docs.python-tablib.org/}.
- [12] Marco Conti. ODSReader. \url{http://www.marco83.com/work/173/read-an-ods-file-with-python-and-odfpy/}.
- [13] Jesse James Garrett. Ajax: A New Approach to Web Applications. \url{http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications}, 2005.
- [14] Allan Jardine. DataTables (table plug-in for jQuery). \url{http://datatables.net}.
- [15] Adam Cohen. FuzzyWuzzy. \url{https://github.com/seatgeek/fuzzywuzzy}.
- [16] Elzbieta Petelenz-Kurdziel, Clemens Kuehn, Bodil Nordlander, Dagmara Klein, Kuk-Ki Hong, Therese Jacobson, Peter Dahl, Jörg Schaber, Jens Nielsen, Stefan Hohmann, and Edda Klipp. Quantitative Analysis of Glycerol Accumulation, Glycolysis and Growth under Hyper Osmotic Stress. *PLoS Computational Biology*, 9(6):e1003084, June 2013.
- [17] Zhike Zi, Wolfram Liebermeister, and Edda Klipp. A Quantitative Study of the Hog1 MAPK Response to Fluctuating Osmotic Stress in *Saccharomyces cerevisiae*. *PLoS ONE*, 5:13, 2010.
- [18] S. Hohmann. Osmotic Stress Signaling and Osmoadaptation in Yeasts. *Microbiology and Molecular Biology Reviews*, 66(2):300–372, June 2002.
- [19] K Luyten, J Albertyn, W F Skibbe, B A Prior, J Ramos, J M Thevelein, and S Hohmann. Fps1, a yeast member of the MIP family of channel proteins, is a facilitator for glycerol uptake and efflux and is inactive under osmotic stress. *The EMBO journal*, 14(7):1360–71, April 1995.

- [20] R Ansell, K Granath, S Hohmann, J M Thevelein, and L Adler. The two isoenzymes for yeast NAD<sup>+</sup>-dependent glycerol 3-phosphate dehydrogenase encoded by GPD1 and GPD2 have distinct roles in osmoadaptation and redox regulation. *The EMBO journal*, 16(9):2179–87, May 1997.

### **Eidesstattliche Erklärung**

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

August 20, 2013

Phillipp Schmidt