1 Introduction

Welcome to FAME, the Flux Analysis and Modeling Environment, a web-based application that aims to be your one stop stoichiometric modeling shop. This tutorial will guide you by its features by first building a model of glycolysis based on information from KEGG, and then editing and running a more complex existing model.

Figure 1: The FAME front page

Going through this tutorial will probably take you between 30 and 60 minutes, and will get you on your way to successfully using FAME in your modeling or teaching work. If you require more information about FAME, its full manual can be found at http://f-a-m-e.org/files/manual.pdf. Footnotes in this document contain extra information about FAME’s inner workings, modeling, or other things. You can safely skip these when doing this tutorial, but you’ll
probably end up reading them anyway, because that’s just the way people are.

System requirements: FAME asks only one thing of you — do not use Internet Explorer. FAME is web-based and designed to work properly with a variety of browsers, but IE is just too stubborn in refusing to adhere to standards everybody else agrees on.¹ Firefox works well with FAME and is available for most platforms, as is Google Chrome (which seems to work even better with FAME, but lacks some of Firefox’s other nifty features — the choice is yours!). FAME has also been found to work with Safari, for you Mac people out there.

2 Creating and running a glycolysis model

Let’s get started. To demonstrate the modular model building capabilities of FAME, let’s start out by building a model of human glycolysis.

- On the home page, click BUILD NEW MODEL. You’ll be taken to the corresponding screen immediately (Figure 2).²
- Under species, select hsa for human.
- Select pathway 00010 - Glycolysis / Gluconeogenesis.³
- Set the model name to something that seems logical, such as “Human-GlycolysisTest”. You can use any combination of alphanumerics and underscores (_), as long as it starts with a letter. For now, we’ll leave the max rates as they are. This means that all reactions will default to lower bounds of -10000 and upper bounds of 10000; any boundaries that you create later on will default to bounds -100 and 100.⁴
- We want to have a place for ATP to go once it has been produced. Under ‘fixes’, check the box that includes ‘ATP’, and also check the box that sets it as a MAX objective. This will include an ATP hydrolysis reaction, so the ATP you generate will have some place to go, and set this reaction as the model objective (i.e. make as much ATP as possible).
- That’s it, click ‘Build’

You’re now taken to FAME’s center page, and are presented with buttons that allow you to run the model straightaway, or edit it first (among other things). Let’s run the model first, and see what happens.

¹In addition, the older versions that are often still in use in institutional settings are security accidents waiting to happen, so it’s best to “upgrade” to another browser anyway.
²In FAME, if possible, links are actually to anchors within the same page. This will save you precious loading time.
³If, later on, you want to construct a model of more than one pathway, hold Ctrl or Command and click all pathways you want to include.
⁴These values are lower to make sure that your model will probably run out of ‘food’ before it hits a rate cap on a reaction. Feel free to change this if you don’t like it, though.
Figure 2: Creating a glycolysis model using the information from KEGG. Note the checkboxes at the bottom, that instruct FAME to include an ATP hydrolysis reaction and make it the objective.
• Check the box that says “Line thickness” (this makes sure the visualization style will resemble that used in this tutorial).

• Click ‘Run!’ (Figure 3). We’ll get to the specifics of the run options later.

![Perform analysis screen](image)

Figure 3: The perform analysis screen — for now, just click run here

The screen that appears now is the basic visualization screen. At the top, the value of the objective function, ATP, is shown, which is 10000 — ATP is being produced! That is weird, because we haven’t given the model any molecules as input yet (such as glucose). Figure 4 explains what’s happening. Superimposed on KEGG’s KGML representation of the glycolysis pathway are our run results. The middle loop is producing ATP out of thin air! We can’t have that...

Details about the reactions are provided in a table below the figure. This table usually starts out displaying species IDs, which, in the case of KEGG-based models, are very hard to interpret. Try clicking “Toggle species IDs and names” to get a more useful table.

If you’re really interested in what’s happening, try clicking some of the colored lines for even more details. In our model, R01512 is producing ATP (the rate is negative, so the reaction is going right-to-left), and the 3-phospho-D-glyceroyl phosphate (or: 1,3-bisphospho-glycerate, 1,3-BPG) required for that is regenerated from 3-Phospho-D-glycerate (3PG) by R01516 and R01662, producing water. The water is used in the ATP hydrolysis reaction, regenerating ADP and Pi, and closing this harmful little circle (look it up in your model results, then see Figure 5). After you’ve found it, let’s go back and do something about this cycle.

The loop at the bottom is interchanging NADPH and NADH, which is irrelevant for now. In larger models, we’ll want to prevent that as well. The loop at the top is just interchanging hexose phosphates, we’ll do something about that to make the figure less cluttered.

---

5The loop at the bottom is interchanging NADPH and NADH, which is irrelevant for now. In larger models, we’ll want to prevent that as well. The loop at the top is just interchanging hexose phosphates, we’ll do something about that to make the figure less cluttered.
Question: explain why this cycle could never produce ATP in real life.

Figure 4: Unexpected results on our first visualization attempt?

- Click ‘Back’ at the top of the page to go back to FAME’s main page.
- Click ‘Edit model’.
- Click ‘Constraints’.
- Look for R01662 and R01515 and change their upper and lower bounds from to 0. This will disable the middle loop.6

6Of course, disabling reactions completely is a rather crude way of getting rid of loops — in large models, they may serve valid purposes elsewhere. The biologically more sensible way of doing it is to put biochemically valid constraints on the offending reactions. In the case of this tutorial, this is left to the reader as an exercise.
Look for R02739 and change both its lower and upper bound to 0 (effectively disabling the reaction for now, and disabling the top loop in the process).

Look for R00711 and change both its lower and upper bound to 0 (again, this disables the reaction, and in this case the bottom loop).

Click ‘Apply’ (Figure 6). Note that before we can make changes to boundaries, we always need to apply the changes to Constraints first!

Now these cycles will not bother us anymore. But we need to do some more before this model will actually do what we want it to: it needs to consume and produce compounds (confusingly called ‘species’ in modeling speak). The rules
about what a model can consume or produce are called ‘boundaries’, so that’s the button we want to click now.

- Click ‘Boundaries’.

The boundaries screen has three boxes: current boundaries, orphan species, and non-orphan species. Currently, the model has no boundaries, so that box is empty. **Orphan species** are species that occur in only one reaction. A chain of reactions can’t go anywhere once it lands on one of them, so they are essentially dead ends in the model and will not participate in reactions unless you do something about it (e.g. make them a boundary species). **Non-orphan species** are all other species in the model.

For each species, you can set one of three kinds of boundaries: **IN**, **OUT**, and **I/O** (= **IN** and **OUT**). **IN** means the species can only be consumed, **OUT** means it can only be produced, and **I/O** means both can happen, depending on the model’s needs.\(^7\)

**Important:** A boundary rate \(<0\) means a species is *consumed*. A rate \(>0\) means it is *produced*.

Let’s set some boundaries now, to make our model work.

- Under orphan species, select **C00186 OUT** by clicking it. This ensures lactate, an end point of simple glycolysis, can be exported.
- In the other box, select **C00001 I/O** to make sure water is freely available to the model. Now scroll down to **C00080 (H+)**, hold down the Ctrl or Command key, and click **C00080 I/O** to do the same for protons. Scroll down even further, to **C00221 (beta-D-glucose)**, and click **C00221 IN** while holding Ctrl or Command.
- We’re set! Click ‘APPLY’.

The page refreshes, and you can check that the model has changed by clicking edit model and e.g. clicking boundaries. You’ll see that four boundaries have been added to the model. Let’s go back up, however (click ‘RUN MODEL’, or just hit the Home key on the keyboard), to go run the model.

- ‘Select your analysis’ is set to Flux Balance Analysis (FBA) by default, let’s keep it like that.

---

\(^7\)In the SBML, the only thing that’s different between **IN**, **OUT**, and **I/O** boundaries are their constraints. So you could change an **IN** boundary to **OUT** by changing the constraints from e.g. \((-100, 0)\) to \((0, 100)\). This would fool the FAME interface, but when running the model it would make perfect sense to PySCeS and the solver.
• We can select more or fewer pathways to view. If we’d have had a big model, selecting fewer would help improve load times, but since we have only one pathway in the model, the default setting of ‘all pathways in model’ is fine.

• So: click run!

![Glycolysis / Gluconeogenesis](image)

Figure 7: The FBA output. Circles, boxes and lines are clickable for more information, because they are linked to additional information from KEGG and/or the model itself.

The visualization screen you saw before is back, but the information mapped on it is different. Now, an actual chain of reactions can be observed on the glycolysis map, reaching from glucose (which you set as an IN boundary) all the
way to lactate (which you set as an OUT boundary). In the figure, red lines indicate reactions with a flux value of < 0. Green lines represent flux values that are > 0, and dashed blue lines are reactions with a rate of exactly 0. Light grey lines are reactions that are in KEGG, but not in your model (that is, they are available in different species). Next to each line, the reaction ID that it represents is displayed, along with the actual flux value in this solution. Thicker lines represent higher absolute values, which makes it easier to track fluxes through your model.

You’ll probably notice that there are still some really thick lines in the figure, representing reaction cycles that perform no net function. This is typical for FBA solutions because of the way the solver works. You can get biologically more meaningful results by choosing “FBA with absolute flux minimization” instead of regular FBA when you run the model (try this). The diminutive increase in the time required to run the analysis is usually outweighed by the increased biological relevance of the results.

Whichever analysis you run, in the figures generated by FAME, we have aimed to make everything clickable. In practice, this means you can click

- lines representing reactions,
- circles representing species, and
- boxes representing enzyme classes (EC numbers)

for more information about the respective item. In the former two cases, the amount of information that is displayed depends on FAME’s ability to link the (meta-)information in the model to information in KEGG.

- For now, let’s click on the circle representing phosphoenolpyruvate (PEP).

The ‘species details’ screen replaces the pathway image, showing which reactions produce (left-hand side of table) and consume (right-hand side of table) PEP, and to what extent. Boundary species IDs are underlined in the species details screen, but there are no such species in the info screen for PEP. PEP’s ID itself is shown in boldface. Again, IDs for which additional information is available are clickable to get more information (e.g., another species details

---

8 Also, at the bottom you’ll notice the thick lines indicating another cycle is taking place. Normally, you’d want to do something about this or at least understand what it is about; for now, we’ll ignore it.

9 Note that “red” doesn’t necessarily mean that a reaction runs in reverse. Lactate dehydrogenase (pyruvate to lactate) shows up in red, because in KEGG it is modeled as Lactate $\rightarrow$ Pyruvate, and so now it runs from right to left in the model, but left-to-right in the image!

10 If you’d have used this function from the start of this tutorial, you would not have been able to detect some of the (harmful!) loops we saw earlier on. So not always using this analysis has its advantages, too!

11 Because circles can get swamped in thick lines and other elements, you can also click the species name (PEP) in the table below the pathway image if you have trouble clicking any circle.
screen, so you can track a pathway through the model. Hover over species IDs to see their human-readable names. You can use the back button to get back to the original results figure.

**Question:** If lactate wasn’t a boundary species but pyruvate was, what do you predict would happen when you’d run the model? Formulate an answer to this question first, then try it. Explain the result (and your prediction, should this differ from the result).

The results of an FBA run represent one possible solution, but there may be many more. In other words, fluxes may be variable to some extent, e.g. because a decrease in the rate of one reaction can be compensated by an increase of another rate.

So, let’s have a look at this by doing a **Flux Variability Analysis (FVA)** on our little model. In FVA, the value of the objective after an FBA run is set as a constraint on the model, and then all other reactions are in turn set as the objective. The outcome of an FVA will tell you where the model is permitted variability given its optimal outcome.

- Click the back button to go back to the run/edit page.
- Under ‘Perform an analysis’, click ‘Flux Variability Analysis’.
- Because FVA runs can take quite a lot of time, depending on the model size, it is possible to perform FVA on a subset of reactions. The system follows the following rules for this:
  - If one or more pathways are selected, only reactions in the selected pathway(s) are probed in the FVA. Results are visualized on the selected pathway maps.
  - If you input a list of reaction IDs in the box below the pathway selection field, one reaction per line, those reactions are evaluated in addition to those in the selected pathway(s).
  - If you don’t select pathways and don’t input reaction IDs, a full FVA is performed and no pathway maps are drawn.

- The current model is small enough to do a full FVA, but of course we want to see the pathway image as well. Fortunately, all reactions in our model are from that pathway, so select **00010 Glycolysis / Gluconeogenesis**. You could type ATP in the box below the pathway selection field to include the ATP hydrolysis reaction as well, but since it’s the objective reaction that wouldn’t make a lot of sense.\(^{12}\)

\(^{12}\)This is done twice: first, the new objective is minimized, then it is maximized.

\(^{13}\)It’s the one reaction of which we’re sure it has no variability, since we are optimizing for it!
• Click ‘Go!’

In the image that appears now, thicker lines represent reactions with more variability (Figure 8). Details are displayed in a table below the figure(s). In this case, the FVA will tell you little you didn’t already know: the reactions that have high variability are those that futilely interchange metabolites but don’t change the ATP production rate. In larger models, however, this information may tell you something about the shape of your solution space, which may in turn lead to biological hypotheses or conclusions.

**Question:** Choose a reaction with nonzero variability, then predict what would happen to all other fluxes if you’d set it to its maximum or minimum flux, as given in the FVA result. Which ones would change, and how? Check your answer by trying it in FAME (use constraints to specify the flux of a reaction). What would happen if you would change the constraints of the reaction beyond those given by the FVA result, if the model’s objective function is fixed to its optimum (as in an FVA; you can also use constraints on the objective function – it’s just a reaction)?

On a final note, I’ll introduce reduced costs and metabolite analyses here. Reduced costs are given in both the FBA and FVA results. Reactions with low subzero reduced costs values are likely candidates if you’re looking for which reaction is throttling your objective function’s rate. That is, for instance, if your objective value is lower than you’d expect or want it to be, looking for the reactions with lowest reduced costs and tweaking their constraints could be a sensible first step. The metabolite analyses (RHS sensitivities and shadow prices) are byproducts of the solution that can be an indication of the importance of the availability of each metabolite to the maximum/minimum value of the objective function. They can be accessed from the “run analysis” screen (Figure 3) and are described in section 4.1.3 of the manual.

3 **Loading and running a larger model: *Heli-cobacter pylori***

Although you could build a large model using the techniques described above, walking you through that would make this a rather lengthy tutorial. So, instead, we’ll take an existing model, and go from there. The Palsson group hosts their SBML models in the BiGG database, and models downloaded from this database will work straightaway in FAME. BiGG requires registration before you can use it, so for this tutorial’s sake, you can use the BiGG *H. pylori* model that can be found at [http://f-a-m-e.org/examples/Hpylori.xml](http://f-a-m-e.org/examples/Hpylori.xml).
Figure 8: Flux Variability Analysis output
• Go back to FAME’s first page (click ‘start over’, or the FAME logo at the top of the page).

• Click ‘Run/edit existing model’.

• Select the H. pylori file you just downloaded, and indicate that it is NOT a FAME model. Alternatively, just copy the URL above into the “File URL” box to have FAME download it directly.\(^\text{15}\)

• Click ‘Go!’

After the model has loaded into FAME (which may take a small while), the screen will look like what you saw before with the glycolysis model, except the model name will be different.

• First, let’s run the model and see what happens. Select some pathways (e.g. 00010 to 00030) and click ‘Run!’ to run an FBA.

As you can see, the model works — it produces biomass. You’ll also notice that the graphical representation of the result will seem incomplete or just “off”. This is because to map the results to specific places on the pathway map, FAME needs to know which reaction goes where. Data from KEGG itself obviously includes this information, but the models from BiGG provide only the next best thing: EC numbers. FAME can map these EC numbers to KEGG reaction IDs, but note that any single enzyme class (EC) may catalyze more than one reaction. But we’re not here to discuss that or the specifics of Palsson’s H. pylori model.

• Click ‘Back’ or use the browser’s back button to return to the editing page.

• Now let’s explore this model a bit. Click ‘Edit’ and then ‘Compartments’ to see what a multi-compartment model looks like in FAME.

• The model has two compartments: Cytosol and Extraorganism, but species have been defined in three, abbreviated as c, e, and b.\(^\text{16}\) The ‘b’ is for boundary species, and is a construct the Palsson group uses in their models to specify which species can enter and exit the system. This is complementary to FAME’s own system: you can still add and remove boundaries using FAME’s system; to remove boundaries as defined in the H. pylori model, remove the reaction exchanging the _b species for the _e one.

---

\(^{15}\)This also works for models that are not hosted on the FAME server!

\(^{16}\)In the model, all that matters is which species interact with each other in reactions. Misusing the SBML “Compartments” construct, as this model does, has no effect on performance — the _c, _e and _b suffixes effectively compartmentalize the species.
3.1 Running FVA on bigger models

Inherently, Flux Variability Analysis is a laborious process (than FBA). For each reaction in the model, two FBAs need to be performed. Performing an FVA on every reaction in a very large model may cause the web process to think something has gone wrong with PySCeS, causing your request to “time out”. To enable you to perform FVAs on such models anyway, you can select a subset of reactions or pathways to analyze. With the *H. pylori* model, this should not be necessary, but you may want to try it out anyway.

- Select Flux Variability Analysis as the analysis we want to run.
- Either just click ‘Run!’ now, or:
  - Select pathways 00010 (Glycolysis) and 00020 (TCA-cycle) by holding the shift or control key.
  - In the box that appeared when you selected FVA, type some reaction names, e.g. R\textsubscript{HCO3E} and R\textsubscript{HSK}
  - Click ‘Run!’

Browse around the FVA results for a bit, and familiarize yourself with the concepts and controls. Note that the coloring of the figures, if you generated any, is different from that in the FBA results page:

Blue lines indicate reactions with zero or almost-zero variability (abs(max-min)).

Purple lines indicate reactions with nonzero variability.

Red lines indicate reactions with nonzero variability and that can have positive and negative rates at the optimum. These reactions also have their variabilities printed in boldface in the details table.

3.2 Gene associations

Many genome-scale metabolic models come with metadata, extra information that is (most often) not encoded as standard SBML, but that is of biological (or bioinformatical) relevance. For instance, for lack of a better place to put it, information about reaction stoichiometry used to be put into notes sections in the model files, and although they can now be encoded in SBML, many model(er)s still conform to this obsolete standard.

Another such piece of information concerns the genes that are associated with each reaction. Associating genes with reactions is a useful technique to find out what would happen if a certain gene or combination of genes were knocked out. Many published models, including those in BiGG, come with metadata describing gene associations, and FAME is designed to be able to read this and allow you to work with it. This is done on the “Gene Association
### Gene association workbench

Attempting to run model for additional information... SUCCESS!

**Objective R_DiuretIP_published**: 0.63981205973

Printed below are the genes specified in your models meta-information, and their associated reactions. Find the reaction has zero flux in an optimized model, knocking this reaction out is not likely to result in useful information.

* : Reaction is associated with more than one gene.

Hover over the reaction ID to see which genes are associated with it.

Click a reaction name to view reaction details and/or edit the reaction.

**Summary**

Gene ID: **H0001**

Unique and meaningful single-knockout runs possible: 252

<table>
<thead>
<tr>
<th>Gene ID</th>
<th>Associated reactions</th>
<th>Toggle all selection checkboxes</th>
<th>Knockout?</th>
</tr>
</thead>
<tbody>
<tr>
<td>H0001</td>
<td>R_BIPS* R_BIPS*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0002</td>
<td>R_KOOPS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0003</td>
<td>R_KODE*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0004</td>
<td>R_RANTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0005</td>
<td>R_OXPHOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0006</td>
<td>R_CS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0007</td>
<td>R_CDC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0008</td>
<td>R_FWD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0009</td>
<td>R_BD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0010</td>
<td>R_KRISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0011</td>
<td>R_MAND1 R_MAND1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0012</td>
<td>R_GRAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0013</td>
<td>R_BNDC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0014</td>
<td>R_MENH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0015</td>
<td>R_BNGC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0016</td>
<td>R_MICL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0017</td>
<td>R_RBD2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H0018</td>
<td>R_RBD3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9**: Accessing the Gene Association Workbench

**Figure 10**: Accessing the Gene Association Workbench

15
Workbench” (GAW), which can be accessed from the FAME main page using the button in Figure 9.

The GAW main screen is shown in Figure 10, and is described in detail in section 4.6 of the manual. The *H. pylori* model introduced above features gene associations, and we strongly encourage you to just play around with them for a bit to get a feel for what they do. Note especially that some reactions require more than one gene in order to work (e.g. “geneA and geneB”; this is usually done for protein complexes), and that some reactions can be performed by any of a number of genes (“geneA or geneB”), and that the gene networks that are constructed this way can add a layer of complexity to a model.\(^\text{17}\) To make reactions associated with more than one gene easier to spot, they are marked by asterisks (*) in the GAW.

<table>
<thead>
<tr>
<th>Question: What kind of genes/proteins could be described by the following gene association relationships?</th>
</tr>
</thead>
<tbody>
<tr>
<td>geneA AND geneB geneA OR geneB (geneA AND geneB) OR (geneA AND geneC) geneA OR geneB AND (geneC or geneD) or geneE</td>
</tr>
<tr>
<td>Hint: The last one is an example of the kind of descriptions you get when modelers that don’t understand logics or biology start writing gene associations.</td>
</tr>
</tbody>
</table>

4 Real life modeling challenges

Now that you’ve become familiar with stoichiometric models and with FAME, why not have a go at some modeling challenges we’ve encountered (and solved) in our day-to-day modeling practice? We’ve prepared the following exercises for you. They do not depend on each other, so you can do them in any order you like, or pick just one.

Now we know that you’re old enough to not skip to the answers straightaway, but to make for a more realistic challenge, you will have to wait an hour before you’ll get access to the exercise solutions. Go to [http://f-a-m-e.org/challenges/](http://f-a-m-e.org/challenges/) to get to the challenges and their solutions. When you click “Get solution”, you’ll receive a personal code, which will allow you to download the answers an hour later.

4.1 Challenge 1: Mixed acid fermentation doesn’t work!

This one had us stumped for quite a while. A model for *Lactococcus lactis*, which should be able to produce 3 ATP per glucose, only managed to produce 2 ATP/glucose, no matter what we tried to fix. Can you find what’s wrong? For your convenience, we’ve removed everything from the model file that’s not\(^\text{17}\)

---

\(^{16}\)Be aware that although they look very biologically relevant, the gene associations in a model are sometimes inaccurate. Especially large strings involving lots of parentheses, ANDs, and ORs are notorious in this sense. Also note that in current-day models, there is a direct relationship between genes and reactions, whereas in biology this is of course not the case.

---
related to this challenge... but perhaps something else is missing? If you are not familiar with mixed acid fermentation, a schematic of the reactions involved is provided with this challenge.

Final remarks

As you’ll probably have concluded by now, working with bigger models is not much different from working with the glycolysis model we’ve built before. It’s building large models that can be cumbersome, and that is why most will want to use work that has already been tried and tested, like the models in BiGG. FAME is targeted at those researchers, but also caters to the more adventurous, who’ll try their hand at building a model from scratch.

You have now been given a quick overview of what FAME can do, based on a small KEGG-based model and a larger one from the BiGG database. While this tutorial was intended to familiarize you with FAME and its controls, it may also have given you an overview of how genome-scale metabolic modeling can work in practice. If you are serious about modeling, and (hopefully) about modeling using FAME, we recommend that you at some point read the full manual to become acquainted with FAME’s advanced editing and simulation features, such as batch commands.

If you have any comments or suggestions as to how we can improve FAME or have it better suit your modeling needs, we would be very interested to hear them. Please drop us a line at j.boele@vu.nl and let us know your thoughts!