Software implementations allowing new approaches toward data analysis, modeling and integration / curation of biological knowledge for Systems Medicine

John Zobolas

Last updated: 22 June, 2020

Contents

List of Tables

List of Figures

Intro

Chapters are currently split as:

- Work done (see Chapter 1)
- Papers (see Chapter 2)
- Experimental/future ideas (see Chapter 3)
- Various text (see Chapter 4)

Updates

- 21/6/2020
  Add paper plan. See structure of Chapter 2 for more info. Clear/refine text.

Keywords

curation/knowledge management, VSM, causal statements, DrugLogics pipeline (model parameterization/calibration and prediction of synergistic drug combinations, performance optimization), biomarker analysis, synergy assessment

Chapter 1

PhD work

This is a summary of all the work that I have done in my PhD until now. (mainly it's about software implementations related to the core technologies within the group). **To include in the thesis text**.

Note though that not all of these will be part of the main thesis (maybe include the rest in a section like 'Funny PhD side-quests').

Druglogics Pipeline

- Lots of refactoring to increase the readability, maintainability and extendability of the source code (complete restructure of classes, addition of others). This has **RRI extensions**, because cleaning and re-structuring software code has a social aspect to it in the sense that other people can now contribute more easily, extend the code, use it (user perspective can bring changes and further improvements to software pipeline even though they may be used for research purposes) - how can you expect users to actually use a piece of code when it's not substantially documented and it's internal logics made obscure because nobody gave attention to detail and structure? How can anybody care for a (software and any) product that you have not cared enough so as to present it in an way that is acceptable, managable and proper?
- Bug fixing

- Enable maven packaging for easier source compilation, testing, installation, management and executing of the code
- Added tests to modules `gitsbe` and `drabme` using JUnit5, mockito and assertJ libraries
- Source code documentation with bookdown for Drabme and Gitsbe
- Enabling *parallel simulations* in Gitsbe (performance optimization)
- Added support for many features (see dev_plan_doc)
- Druglogics ROC generator: R shiny app to assess the performance of the Drabme results in the form of a ROC curve
- Export support using BioLQM: the initial model + best generation models can now be exported through configuration options to **GINML, SBML-Qual and BoolNet** community formats
- **Attractor calculation** (stable states and minimal trapspaces) using BioLQM
- abmlog package.
- `emba` (J. Zobolas 2020a). See gitsbe-model-analysis repository with various analysis using this package (input is files from `gitsbe`).

Attractor Tools performance comparison

Some preliminary results comparing **fixpoint** and **trapspace** calculation tools using the following 2 topologies: **CASCADE 1.0** (77 nodes and 149 edges) and **CASCADE 2.0** (144 nodes, 367 edges). Both of these models are self-contained (no inputs, less attractors).

Tools tested (**fixpoints**): `BNReduction` (full and reduced version - the second is considerably faster (main reason we were using it), but does not find more than 1 stable state which was more like a feature we wanted especially for larger self-contained networks), `BioLQM (BDD-based)`.

Tools tested (**trapspaces**): `BioLQM (BDD-based)` vs `PyBoolNet` (I also tried the ASP-based solver of BioLQM but it was slower than the BDD-based algorithm, as Aurelien states in the documentation)

- FIXPOINTS
  - CASCADE 1.0: BioLQM(fixpoints) « BNReduction(reduced)

    (~same time to calculate the trapspaces using BioLQM!) « BNReduction(full).
  - CASCADE 2.0: BNReduction(reduced) « BNReduction(full) < BioLQM(fixpoints)

So, for our smaller topology, we can actually use the BioLQM stable state calculation and it will give us complete fixpoint results even faster than the BNReduction (reduced version) we are using currently. Even more: **we could use the BioLQM trapspace computation for that smaller model! For the larger network though, I am afraid BioLQM does not scale as well as the BNReduction does.**

- TRAPSPACES
  - PyBoolNet « BioLQM, no matter the model chosen. Especially for the larger network, BioLQM takes several minutes when PyBoolNet takes less than 10 seconds. For the smaller topology, it's the same difference in scale but in millisecs (PyBoolNet: ~10msecs, BioLQM: ~200msecs per model).
  - MPBN(https://github.com/pauleve/mpbn/) » PyBoolNet. The sad thing is the overhead you get when runnng the Python module through java which is actually longer for smaller networks than the calculation of the trapspaces themselves! Implemented in the pipeline now.

VSM

I made `vsm-dictionaries` with a focus on the biological domain. We call them UBDs: *Unified Biological Dictionaries*. See UBD GitHub organisation page.

Most of this work is done in order to support Vasundra's causalBuilder Tool which is the first application of VSM after SciCura v1.

PSICQUIC

My work at the EBI with IntAct and Noemi Del Toro to extend the PSICQUIC web service to support the MITAB 2.8 data format. See the psicquic doc and the casualTab paper (Perfetto et al. 2019).

I also worked with Noemi on the update of the JAMI library to also support MITAB 2.8 - this is the culmination of results from the BioHackathon 2018, in Paris and the Marseille GREEKC hackathon event.

Others

- Java Client for RSAT tool fetch-sequences
- `usefun` (J. Zobolas 2020c) (for the community!)
- `rtemps` (J. Zobolas 2020b) (for the community!)

Chapter 2

PhD Papers

Knowledge management/biocuration

- MITAB 2.8 paper (Perfetto et al. 2019)
  Extending the PSICQUIC Web Service platform to include causality information of molecular interactions. See Report from EBI visit (July 2018).
- Biohackathon 2018 paper, still in **unknown status**!

VSM-related papers

- VSM-box paper
- UniBioDicts paper
- causalBuilder paper

Logical Modeling/Predicting Synergies/Finding mechanisms

- `emba`: R package for analysis and visualization of biomarkers in boolean model ensembles (JOSS manuscript)
- Pipeline paper I: an AGS story (Asmund's paper, me as co-author)

- *synergy* paper - see manuscript
- balance mutations paper - see manuscript

Chapter 3

PhD ideas

Several ideas that I may do or not in my PhD but I still keep here for my future investigations!

Extend *SynergyFinder*

The core idea here is to extend an existing R package (Ianevski et al. 2017) for calculating synergy reference models in order to include Wim's generalized Bliss method (Mulder, Kuiper, and Flobak 2019) and the mean synergy score by Simone Laderer (Lederer, Dijkstra, and Heskes 2018)! Then I will test all the null reference models (Loewe, Bliss, ZIP + others) on dose-response matrix datasets and see which is best at finding the synergies in each dataset (given some gold standards I guess).

Also, I should investigate if my own idea for a mathematical formulation of the volume-based synergy score as general method for describing 3-wise or more combinations as synergistic, could be part of this implementation. Though SynergyFinder 2.0 (Ianevski, Giri, and Aittokallio 2020) has simplemented something similar for high-order combinations using tensors.

Quantum logic formalism

My favourite! Investigate if instead of a logical modeling formalism, the idea of (quantum) logical gates can be used to represent and analyse protein interaction networks. The **core idea** makes sense: you don't know the state of a protein, but when you measure it, only then you really know what it is.

May also be worth to look at a game-theoritic approach to find attractors and such.

Compare fixpoint tools

Idea: Compare different tools that calculate fixpoints for logical modeling. Faster wins of course :) Extension of this small investigation.

Models used for testing could be of different types:

- self-contained
- varying the number of input nodes (1-n)
- small to large number of nodes
- small to large number of edges
- scale-free (boolnet generated) vs random (varying K connectivity)
- play with form of the boolean equations
- others ???

Workflow for this includes:

- support BNReduction data format by Veliz-Cuba in BioLQM
- add support for calculating the fixpoints using the Colomoto docker (python interface) + BNReduction
- then comparison between **BioLQM, Pint, MABOSS, MPBNs and BNReduction** could be done then in a Jupiter colomoto-enabled notebook!

Further extension/comparisons could be:

- (Akutsu, Hayashida, and Tamura 2009) - Integer programming method
- (Dubrova and Teslenko 2009) - SAT-based

Use Logical modeling to predict single-drug data

Asmund's proposal idea that he sent to my email once. Has to do about *mechanistic drug response prediction analysis*:

- Automate drug target profile annotation from:
  - (Klaeger et al. 2017)
  - mrc ppu
  - (Davis et al. 2011)
- Omics data (rna, cnv etc)
  - COSMIC
  - CCLE
- Drug screen data
  - Single drug
    - COSMIC/GDSC
    - CCLE
  - Combo
    - (O'Neil et al. 2016)
    - (Holbeck et al. 2017)

My idea is more like this:
Predict drug-response curves from drug combination datasets (GDSC, CCLE), using logical modeling for singaling network analysis or translation from logical to ODE modeling. Also try to predict drug combinations datasets (dose-response matrices?). Pretty much what is done in this paper (Fröhlich et al. 2018) with help from (Wittmann et al. 2009) for converting boolean models to continuous.

Druglogics-Pipeline related

Harmony Search

Nice idea because it's related to music! Investigate if this algorithm could be used for optimizing the boolean equations for `gitsbe` - thus opening the stage for **JazzLogics**!

Train models to cell-specific proliferation

Concept is that training models to proliferate provides a wider variance of models than the cell-specific trained ones in `gitsbe`: main directive is **proliferation**, not just fitting to a steady state pattern. So a hybrid training approach should be way more advantageous in that point of view.

A bottom-up model building for drug prediction

Start with a model and some observed synergies. Build/train/produce models that predict the first observed synergy (using Harmony Search?), from them the next one, etc. You end up with many models that can predict all the observed synergies or you try to find out why that cannot happen for example (e.g. contrasting synergies? mechanistic explanation why that happens?). Do the resulting models stable states or attractors correspond to activity protein profile from literature?

Simulate cancer resistance

For example, you have some models that predict some (observed) synergies or you just find some synergistic drug combinations for these models or per model. Then, you modify these models in order to be resistant to these drugs, simulating thus the cancer rewiring process! Then, you apply (n+1) drug combinations to win over the resistance (and you do this procedure at more levels to suggest 3-way, 4-way drug combos and why there might be cancer models that can 'win' over these models and continue the proliferation). You end up with super cancer resistant models and methods to achieve them or reasons why this cannot happen at all (again learning about the mechanism of these).

Causal-JSON or MI-JSON to boolean model converter

Idea

This idea is like a continuation of the `causalBuilder` tool by Vasundra coupled with the need to have a more proper representation of complexes (and families) in our logical models (better models, better predictions). Asmund had *manually* changed some logical equations in his paper (Flobak et al. 2015), in order to make the model more compliant with biology knowledge and literature findings. One of them was about the beta-catenin complex and its constituents (connected with *AND's* instead of *OR's*) and the rest were about changing the link operators of the logical equations (from *AND NOT* to *OR NOT*). The latter is something that is enabled through the mutations introduced by the genetic algorithm of `Gitsbe`. The former depends on the dataset and the representation of complexes.[1]

Only Signor (Licata et al. 2019) has some complexes + interaction data but they are seperate files, making it thus difficult (and non-elegant computationally-wise) to integrate such knowledge/data to boolean models. Also Vasundra's experience with Reactome data in miTab2.8 showed us the difficulty to match binary interactions to a data model flexible enough to represent complexes and their internal components. Causal-JSON and the recursive schema that we thought allows the curator to put both the complex ID and it's constituents in the same data structure.

Proposed Workflow

1. Get interaction + complexes/families data (Signor most probably or a form of CASCADE + complexes). Note that for the reason I explained

[1] There is actually a mutation that can change this but not in the way that we want - i.e. all components of a complex should be connected with an *AND*

above miTab 2.8 is out of the question, so the Signor data I am refering to is the .tsv files they offer (interaction data, complexes, families). And most probably I am referring to a **pathway interaction dataset** not the whole Signor data. For example, the Wnt Signaling pathway.

2. Build a small module that translates the (Signor) data to Causal-JSON.

3. **Main:** Build a package that translates the causal-JSON data to a logical model with some filtering and parameterization included (e.g. filter based on cell line (so *cell-line* specific topologies), conditions on the biological state: 'by phoshorylation', exp. evidence, assertion/confidence score, species, compartment). So, **causal-JSON to .bnet files (logical equations)**, while substituting/extending nicely the complexes and families.

4. Showcase some small application of this logical model end-product:
   - use for example the colomoto notebook, do some small trapspace analysis and show that some results from literature or from previous logical or other models can now be reproduced with a better biological representation in the model itself, AUTOMATICALLY!
   - make many logical models of the pathways in Signor with simple attractor analysis and put them into the GinSim model repository for reference for the logical community.
   - extend `atopo` module to use the main package (see point No. 3) and use it for finding drug combinations (comparing attractors or prediction results of automated topology building without complexes vs automated topology built from causal-JSON with complexes and families in each case). The main thing here would be of course better prediction performance results based on a better logical model representation. I could tweak `atopo` to choose actually not all of Signor's data but specific pathways to include in the analysis and this will help I believe to build smaller topologies for specific drug combinations that we want to test.

Chapter 4

Text

Here I have various text the I write at times and I will include most probably in my completed thesis report:

Precision (PPV) as a metric for pharmaceutical companies

I think if you say the following: I have 1000 drugs in my lab/company, that is almost 500.000 pairwise combinations. I want to screen combinations for synergies, but I only have screening capacity for 100.000 combinations, so there are 400.000 combinations I can't screen.

How do I choose the 100.000 combinations to screen?

Let's say there are in fact 50.000 synergies, meaning a prevalence of 10%. There are 3 alternatives (using the pipeline):

1. Blindly. You will get a prevalence of synergies pretty close to the prevalence of synergies among the 500.000, so you will get 10.000 synergistic pairs.
2. Screen by a **topology guided prediction (random models)**. You will get maybe 13% if I remember correctly for atopo, i.e. 13.000, meaning 3000 more than blinded

3. Screen by a **simulation guided prediction (fit to steady state/calibrated models)**. You will get maybe 20%, i.e. 20.000, which is 10.000 (double) than blinded, and 7.000 more than guided by topology.

Why automated topology?

My take on this:

1. Yes, one of the reasons is advantage in the simulations in the sense that when you have logical models that have no inputs you are statistically more able to have models with a few/less (even better: 1) stable state(s). See it like this: if a logical model has one input (let's say node X) then it's sure that this logical model will have one attractor with X:0 and one with X:1, be it a stable state or a more complex attractor. Two inputs, 4 attractors, etc. (Denis helped me realise this actually on a talk in Athen's ECCB, great times)
2. The second reason that Asmund told me about when I asked him the same thing, was one of the most basic hypothesis behind his modeling - which sums up to *where cancer comes from*. By using a no-inputs topology we adhere to the principle that cancer is something that relates to the system itself and not to the external interactions of the system. It comes from dysregulations within related to "broken" circuits, etc. It is in contrast with the traditional view on the same thing that experimentalists used to understand cancer: I perturb the system (cell) by inhibiting a specific hormone/receptor (input) and see how it reacts (and where most modeling approaches are based on).

Asmund's take on this: 1. few stable states 2. cancer is a system disease in itself (not related to e.g. external hormones, they are present also for healthy cells, it is something in the cancer cell that allows it to sense the external hormone differently than healthy cells).

In addition (and maybe related to point 1): 3. In 'traditional' modeling a system is defined to respond in a certain way to a set of specified 'inputs' (I mean not here model input nodes but rather a configuration of the model, e.g. ERK is active). A self-contained topology merges the input condition and the output response in a single observable entity: When the system is initialized in its stable state it will remain there. Therefore observation of baseline signaling is both the input and the output, reduces need for perturbations.

4. In addition to few stable states (point 1) I believe a self contained topology also means few possible parmeterizations. I don't have any mathematical proof of this but it seems reasonable to me.

VSM-related

Steven's email about VSM and reasoning:

VSM is about a drive to marry the model-driven & the observation-driven worlds of mathematics, into real-world applications. This is becoming apparent in my own work on VSM: it is a pursuit to more closely emulate human thinking - as our brains clearly manage to integrate both worlds. To mimic this integration we need better tools. VSM started off as the design for a high-level, intuitive, universal Knowledge Representation, that makes it easier to manage the model-driven world when working with heterogenous, context-rich knowledge. (It boils down a thought (not language) onto basic conceptual structures). And now it appears to naturally follow from the semantics of VSM that it requires observation-based, defeasable (not-only-logical) reasoning. I.e. it invites for a mechanism whereby "hard"-ish prior semantic knowledge (rules) gets extended with "soft" tentative assumptions, prioritized based on collected episodic knowledge (formalized observations). And perhaps, these observations may even be synthesized into new guiding rules as well.

I think that biological systems are extreme in their magnitude of complexity, where hundreds of thousands of diverse causalities can all work together. To understand them, we need not only big-data machine learning to interpret large-scale data (from new, more local observations), but also defeasible symbolic-like reasoning, as a guide through our diverse and large-scale prior knowledge.

Acknowledgements

To Asmund for good supervision:

- Best writing tip: Every paragraph should start with a ‘why’, next ‘what done’, next ‘what found’, next ‘what does this mean’ => every paragraph is like a small paper, intro + methods + results + discussion

To Martin, firstly for taking me in as a PhD student and giving me the opportunity to come to Norway. Secondly for his strategical advice during my PhD,

To Astrid Laergid, for letting me know about the RRI course! And subsequently to Roger Stran, Heindrun Am and all the other participants of the RRI course: mpla mpla for the expericence and making me realise the complexity of science and the beauty of its philoshophy. To Steven for teaching how to be a write good open-source Web developer and for some nice conversations about context, knowledge representation To Noemi Del Toro Ayllon, To IntAct and Henning for letting me be part

Bookdown useful links

- Bookdown github repo
- Bookdown package reference
- Writing Thesis with Bookdown
- Bookdown workshop slides

References

Akutsu, Tatsuya, Morihiro Hayashida, and Takeyuki Tamura. 2009. "Integer programming-based methods for attractor detection and control of boolean networks." In *Proceedings of the 48h Ieee Conference on Decision and Control (Cdc) Held Jointly with 2009 28th Chinese Control Conference*, 5610–7. IEEE. https://doi.org/10.1109/CDC.2009.5400017.

Davis, Mindy I, Jeremy P Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. 2011. "Comprehensive analysis of kinase inhibitor selectivity." *Nature Biotechnology* 29 (11): 1046–51. https://doi.org/10.1038/nbt.1990.

Dubrova, Elena, and Maxim Teslenko. 2009. "A SAT-Based Algorithm for Computing Attractors in Synchronous Boolean Networks." https://arxiv.org/pdf/0901.4448.pdf%20https://ieeexplore.ieee.org/document/5958722/.

Flobak, Åsmund, Anaïs Baudot, Elisabeth Remy, Liv Thommesen, Denis Thieffry, Martin Kuiper, and Astrid Lægreid. 2015. "Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling." Edited by Ioannis Xenarios. *PLOS Computational Biology* 11 (8): e1004426. https://doi.org/10.1371/journal.pcbi.1004426.

Fröhlich, Fabian, Thomas Kessler, Daniel Weindl, Alexey Shadrin, Leonard Schmiester, Hendrik Hache, Artur Muradyan, et al. 2018. "Efficient Parameter Estimation Enables the Prediction of Drug Response Using a Mechanistic Pan-Cancer Pathway Model." *Cell Systems* 7 (6): 567–579.e6. https://doi.org/10.1016/J.CELS.2018.10.013.

Holbeck, Susan L, Richard Camalier, James A Crowell, Jeevan Prasaad Govindharajulu, Melinda Hollingshead, Lawrence W Anderson, Eric Polley, et al. 2017. "The National Cancer Institute ALMANAC: A Comprehensive Screening Resource for the Detection of Anticancer Drug Pairs with Enhanced Therapeutic Activity." *Cancer Research*. https://doi.org/10.1158/0008-5472.CAN-17-0489.

Ianevski, Aleksandr, Anil K Giri, and Tero Aittokallio. 2020. "SynergyFinder 2.0: visual analytics of multi-drug combination synergies." *Nucleic Acids Research*. https://doi.org/10.1093/nar/gkaa216.

Ianevski, Aleksandr, Liye He, Tero Aittokallio, and Jing Tang. 2017. "SynergyFinder: a web application for analyzing drug combination dose–response matrix data." *Bioinformatics* 33 (15): 2413–5. https://doi.org/10.1093/bioinformatics/btx162.

Klaeger, Susan, Stephanie Heinzlmeir, Mathias Wilhelm, Harald Polzer, Binje Vick, Paul-Albert Koenig, Maria Reinecke, et al. 2017. "The target landscape of clinical kinase drugs." *Science (New York, N.Y.)* 358 (6367): eaan4368. https://doi.org/10.1126/science.aan4368.

Lederer, Simone, Tjeerd M H Dijkstra, and Tom Heskes. 2018. "Additive Dose Response Models: Explicit Formulation and the Loewe Additivity Consistency Condition." *Frontiers in Pharmacology* 9: 31. https://doi.org/10.3389/fphar.2018.00031.

Licata, Luana, Prisca Lo Surdo, Marta Iannuccelli, Alessandro Palma, Elisa Micarelli, Livia Perfetto, Daniele Peluso, Alberto Calderone, Luisa Castagnoli, and Gianni Cesareni. 2019. "SIGNOR 2.0, the SIGnaling Network Open Resource 2.0: 2019 update." *Nucleic Acids Research*, October. https://doi.org/10.1093/nar/gkz949.

Mulder, Wim De, Martin Kuiper, and Åsmund Flobak. 2019. "A reference model for the combination of an arbitrary number of drugs: A generalization of the Bliss independence model." *bioRxiv*, May, 630616. https://doi.org/10.1101/630616.

O'Neil, Jennifer, Yair Benita, Igor Feldman, Melissa Chenard, Brian Roberts, Yaping Liu, Jing Li, et al. 2016. "An Unbiased Oncology Compound

Screen to Identify Novel Combination Strategies." *Molecular Cancer Therapeutics* 15 (6): 1155–62. https://doi.org/10.1158/1535-7163.MCT-15-0843.

Perfetto, L, M L Acencio, G Bradley, G Cesareni, N Del Toro, D Fazekas, H Hermjakob, et al. 2019. "CausalTAB: the PSI-MITAB 2.8 updated format for signalling data representation and dissemination." Edited by Jonathan Wren. *Bioinformatics*, February. https://doi.org/10.1093/bioinformatics/btz132.

Wittmann, Dominik M, Jan Krumsiek, Julio Saez-Rodriguez, Douglas A Lauffenburger, Steffen Klamt, and Fabian J Theis. 2009. "Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling." *BMC Systems Biology* 3 (1): 98. https://doi.org/10.1186/1752-0509-3-98.

Zobolas, John. 2020a. *Emba: Ensemble Boolean Model Biomarker Analysis*.

———. 2020b. *Rtemps: R Templates for Reproducible Data Analyses*. https://github.com/bblodfon/rtemps.

———. 2020c. *Usefun: A Collection of Useful Functions by John*. https://github.com/bblodfon/usefun.