Pacific Symposium on Biocomputing 14:63-74 (2009)

LEARNING CYCLIC SIGNALING PATHWAY STRUCTURES WHILE MINIMIZING DATA REQUIREMENTS

K. SACHS† AND S. ITANI†
karensachs@stanford.edu, ssolomon@mit.edu

J. FITZGERALD, L. WILLE AND B. SCHOEBERL
{*jfitzgerald,lwille,bschoeberl*}*@merrimackpharma.com*

M. A. DAHLEH, G. P. NOLAN
dahleh@mit.edu, gnolan@stanford.edu

† *These authors contributed equally to this work.*

Bayesian network structure learning is a useful tool for elucidation of regulatory structures of biomolecular pathways. The approach however is limited by its acyclicity constraint, a problematic one in the cycle-containing biological domain. Here, we introduce a novel method for modeling cyclic pathways in biology, by employing our newly introduced Generalized Bayesian Networks (GBNs). Our novel algorithm enables cyclic structure learning while employing biologically relevant data, as it extends our cycle-learning algorithm to permit learning with singly perturbed samples. We present theoretical arguments as well as structure learning results from realistic, simulated data of a biological system. We also present results from a real world dataset, involving signaling pathways in T-cells.

1. Introduction

Since the seminal work by Pe'er and Friedman, [8], Bayesian networks (BNs) have been used extensively in biology, to model regulatory pathways both in the genetic[13,8] and in the signaling pathway domain[10,14]. Bayesian network models encode probabilistic relationships among random variables in a domain, providing a framework for tasks such as *s*tructure learning. In a biological setting, the random variables represented are biologically important entities such as genes, small molecules and activated or phosphorylated proteins. The structure learning task consists of searching the space of pos-

Pacific Symposium on Biocomputing 14:63-74 (2009)

sible structures to find the one that bests reflects probabilistic relationships in a biological dataset.

In spite of their usefulness, Bayesian network models are limited in their applicability in this domain because they are constrained to be acyclic, while positive and negative feedback loops abound in biological pathways. In particular, Bayesian network structure learning will *always* yield an inaccurate structure for any cycle containing pathway and, as a result, will fail in its predictive capacity (at minimum) for variables downstream of an incorrectly directed edge. When time course data are available, it is feasible to represent cycles by unrolling them in time, using a Dynamic Bayesian networks (DBNs), or Continuous Time Bayesian networks (CTBNs).[12,4] However, DBNs suffer from various computational challenges and necessitate timecourse data, which in some domains are not feasibly attainable in an applicable form (e.g.[10]). Therefore, it would be useful to find an approach for learning cyclic structures from static 'snapshot' data, collected at a single timepoint from a dynamic system.

We have recently developed a formalism for representing cyclic structures using Generalized Bayesian networks (GBNs), a form of Bayesian networks that we have generalized to encompass cycles.[1] This formalism enables structure learning in a cyclic domain, relying on perturbations which break the cyclic structure. Far from requiring an exhaustive set of perturbations, the algorithm is designed to minimize the number of interventions needed, requiring as few as merely one intervention per cycle for accurate structure learning.

Here, we present the first ever application of GBNs to biological signaling pathways. We apply the algorithm to realistic, biologically relevant data from a differential equation model of IGF signaling. Next, we substantially modify the structure learning algorithm to bring it incrementally closer to applicability in a biological domain, by minimizing the algorithm's data requirements. We then test this new algorithm on a reduced set of the synthetic data and compare its results. Finally, we perform structure learning on real data in which T-cell signaling molecules were measured using multidimensional flow cytometry from[10], and demonstrate that our new algorithm is able to elucidate cyclic structures in signaling pathways.

2. Background and Methods

We present background on BNs and GBNs, as well as the synthetic data used in this study.

Pacific Symposium on Biocomputing 14:63-74 (2009)

2.1. Bayesian networks

Bayesian networks[15], represent probabilistic dependence relationships among multiple interacting components, illustrating the effects of pathway components upon each other in the form of an influence diagram- a graph (G), and a joint probability distribution. In the graph, the nodes represent variables (the biomolecules) and the (lack of) edges represent (conditional in)dependencies[15]. For each variable, a conditional probability distribution (CPD) quantitatively describes the form and magnitude of its dependence on its parent(s). Due to the factorization of the joint probability distribution, the graph must be *acyclic*, meaning that it must not be possible to follow a path from any node back to itself.[15]

2.2. Generalized Bayesian Networks

When building models of pathways, Bayesian network models have a number of strong advantages. They are flexible and interpretable, they can handle interactions of arbitrary complexity (given sufficient data), and they can smoothly incorporate both prior knowledge and interventional data in a principled way. However, they have one serious drawback for modeling biological systems: they are unable, as described above, to handle cycles in a static model. Because cycles abound in biological pathways, a static Bayesian network model usually cannot hope to capture all influence connections.

To address this problem and enable the use of Bayesian network models in a cyclic domain, we recently introduced Generalized Bayesian Networks (GBNs), a generalization of Bayesian networks to the cyclic domain.[1] In [1] we also present an algorithm in which the GBN formalism is used to recover causal structure given interventional (static) data, in acyclic *or* cyclic domains. The algorithm is briefly presented below.

GBN structure learning

Call the set of variables V and the subset of variables with interventions available I. Note that the inhibitors are *activity inhibitors*, typically small molecule inhibitors which perturb the activity- rather than the abundance- of a protein.

Algorithm: Learn Causal GBN structure

Pacific Symposium on Biocomputing 14:63-74 (2009)

0: Start with a Causal GBN and an intervention set I.
1: [Probing experiments] Collect sets of i.i.d. samples under no-intervention and single-intervention data, i.e. when node i is intervened at, for each i in I.
2: Call subroutine 'detect descendants' to recover descendant information for all nodes in I. *Based upon response of variables to perturbations, further explained in subroutine below.*
3: Identify cycles. *Based upon perturbations which affect the abundance of the target variable. Because the inhibitors affect protein* **activity**, *an effect on abundance must be due to a loop from the target back to itself.*
4: Do Bayesian Network learning with the cycles broken (by interventions on nodes we call "**cycle breakers**"- a set of variables that include at least one representative from each cycle) and integrating the descendant information.
5: Determine the correct edges needed to close the cycles, by detecting the children of the cycle breakers.
6: Recover all missing edges in the DAG, and complete the Directed Cyclic Graph structure of the Causal GBN.

The subroutine used in step 2 is an important subroutine that uses the sensitivity of descendants to perturbations on their ancestors. This sensitivity can be described as the assumption that if a distribution of the parents of a variable change, then the distribution of the variable itself will change too. This idea was introduced in the context of GBNs[1]. To test whether a node was affected by a perturbation on another node (and thus deduce the anscestor/descendant relation), the following subroutine is used:

Subroutine: Detect descendants

0: Start with sets of n i.i.d. samples generated by a GBN, under no interventions as well as single-interventions at each i in I. Initialize a binary $|V| \times |I|$ descendant information matrix.
1: For each $j \in V$:
2: Compute $\hat{\mathbb{P}}^n(X_j)$, the empirical marginal of X_j under no interventions.
3: For each $i \in I$:
4: Compute $\hat{\mathbb{P}}_i(X_j)$, the empirical marginal of X_j under the single-intervention i.
5: Evaluate some distance between $\hat{\mathbb{P}}(X_j)$ and $\hat{\mathbb{P}}_i(X_j)$.
6: If the distance exceeds a threshold, mark j as a descendant of i.
7: Next i.
8: Next j.
9: Compute the transitive closure of the descendant information matrix, and return it.

The details of this algorithm are presented in[1] and it is proven that given

Pacific Symposium on Biocomputing 14:63-74 (2009)

enough data and interventions, this algorithm is guaranteed to recover the causal structure of the data. Note that step 4 removes all cycles to enable structure learning using standard BN. The algorithm developed in this work avoids this step, to avoid the requirement for conditions involving inhibitor combinations. As such, this modified algorithm is far better suited to the logistic reality of limited experimental data.

2.3. Synthetic Data and Model of IGF Signaling

To produce synthetic data, we use a mass action kinetic model describing the dynamics of the Insulin-like growth factor (IGF) signaling pathway.[5,6] IGF signaling is important in normal cell physiology, as well as pathological states such as cancer. A schematic representation is shown below. Mass action kinetic equations were used to create the model in MATLAB SimBiology v2.1.

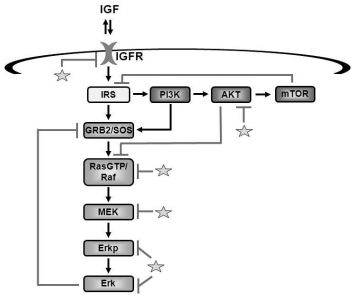

Figure 1. *True structure of the underlying dynamic system in IGF signaling.* Each node represents the active "on state" of the protein. Stars indicate phosphatases acting on the target molecules. Perturbations in the form of small molecule inhibitors are available for Mek, Akt, Pi3k, IGFR and mTor. The simulated data mimics these inhibitors by blocking enzyme activity.

There are three directed cycles in the model: IRS → PI3K → AKT → mTOR → IRS, GRB2/SOS → RasGTP → MEK → ERKp → ERK → GRB2/SOS, and GRB2/SOS → RasGTP → MEK → Erk → GRB2/SOS.

The stimulus employed is IGF, in addition, up to five perturbations are employed, at IGFR, MEK, PI3K, AKT, and mTOR, corresponding to actual existing small molecule inhibitors. All of the perturbations are activity inhibitions, that is, they inhibit the protein's activity, not permitting the targeted protein to phosphorylate other proteins. We generated measurements from four different time points, under 17 total conditions composed

Pacific Symposium on Biocomputing 14:63-74 (2009)

of IGF stimulus plus various combinations of inhibitors. For each condition, 1000 unique, randomly selected initial conditions (i.e. molecule concentrations) were employed- the equivalent of collecting 1000 unique cells in a flow cytometry experiment, or performing western blots on 1000 samples. Simulated "measurement noise" was also added.

The model was created by Jonathan Fitzgerald and colleagues at Merrimack Pharmaceuticals. It is a highly accurate imitation of the true biological system[19], and, accordingly, provides us with synthetic but realistic data, similar to the data one might acquire from a high throughput measurement technology (as in [10]). It is a flexible and realistic source of 'true to life' synthetic data, which, because it has a known ground truth model, provides an invaluable tool for assessing success of structure learning efforts.

3. Algorithm Description

In this section, we detail an algorithm for structure learning of cyclic networks with ***single*** perturbation data. This algorithm depends on first detecting the cycles in the network using perturbations, then using the data where each cycle is broken to recover its structure.

Algorithm: Learn Structure

0: Start with a set of variables V and a set of single-intervention variables $I \subset V$.
1: [Experiments] Collect sets of i.i.d. samples under no-intervention and single-intervention data, i.e. when node i is intervened at, for each i in I.
2: Call subroutine 'detect descendants' to recover descendant information for all nodes in I.
3: Identify one node per cycle from I, and the set of such nodes $I_{\mathcal{C}}$. *Nodes in $I_{\mathcal{C}}$ are detected as self-descendants.*
4: *Apply a BN structure learning algorithm to recover a DAG representation of the dependencies.*
5: $\forall\, i \in\ I_{\mathcal{C}}$
6: *Apply a BN algorithm on data with i inhibited.*
7: *Use conditional independency tests to prune out edges and determine path structure between nodes and their descendants.*
8: *Compare the different sub-graphs pertaining to i to recover the cycle structure.*
9: *Next i.*
10: *Integrate all of the cycle structures with the result from step 4, and call the resulting structure G.*

Pacific Symposium on Biocomputing 14:63-74 (2009)

4. Algorithm Analysis

In this section we aim to study the algorithm we presented is Section 3, mainly focusing on the reasons it can be expected to have good performance as well as where it is expected to have limitations.

The main assumption needed for the algorithm to recover the true structure is that the BN structure learning algorithm recovers the correct structure of the non-cyclic part of the graph whenever it is applied. This means that the existence of a cycle shouldn't interfere with the conditional independencies between variables outside the cycle itself.

From the study in [18], it can be seen that the cycle does not usually affect the rest of the graph dramatically. This is mainly because the structure of the d-separation in a cycle (what dependencies/independencies conditioning on every set of variables induce) is the same as that of a loop with the same structure of the cycle except for one reversed edge. Thus the BN learning algorithm would tend to recover the whole structure with some reversed or missing edges from the cycle, and usually nodes that are outside the cycle will not be affected.

We therefore expect this algorithm to perform well, even compared to algorithms that use multiple-perturbation data. This is mainly because this algorithm reinforces and corrects the structure recovered by the BN algorithm. It does so using the descendant information from the perturbation analysis and the conditional independency analysis (step 7).

5. Results

In this section, we present structure learning results for BNs and GBNs, as well as the new GBN algorithm presented above. BN learning is implemented as in[10]. Data was discretized to 4 levels using interval discretization. We show the results obtained from the IGF-model based data with combinations of inhibitors, 17 conditions total, followed by conditions employing only single perturbations (6 conditions total), *up to* 1000 datapoints were used per condition. Lastly, we show results using a flow cytometry dataset of T-cell signaling molecules. The models presented are averaged over 20 individual results, edges with confidence > 0.8 are included. In the following graphs a dotted edge is an incorrect edge that was predicted, a black edge is a correct edge. For the GBN based algorithms, the degree of shifting for which a variable is considered to be a child of the perturbed variable is a model parameter. Here, we use a 20% shift as the cutoff, chosen based on an observed bimodality among the candidate children. Another good

Pacific Symposium on Biocomputing 14:63-74 (2009)

approach would be to determine the cutoff by randomizing the data and determining the magnitude of a null shift.

5.1. Multiple perturbations

The original GBN structure learning algorithm requires conditions in which multiple perturbations are applied simultaneously. Results from the BNs and the original GBN structure learning algorithm are shown below, both use 17 different conditions, each with different combinations of the 5 inhibitors. The BN results, shown on the left, find most of the model edges (missing the connection between mTor and IRS, as well as Ras/Raf and Mek, and shifting the $Erk \rightarrow Grb2Sos$ connection to Erkp), however, the model contains 10 additional, noncausal edges. The GBN results improve on the BN results substantially, missing only the $Erk \rightarrow Grb2Sos$ edge, which it too shifts to Erkp, and including only 2 extra noncausal edges.

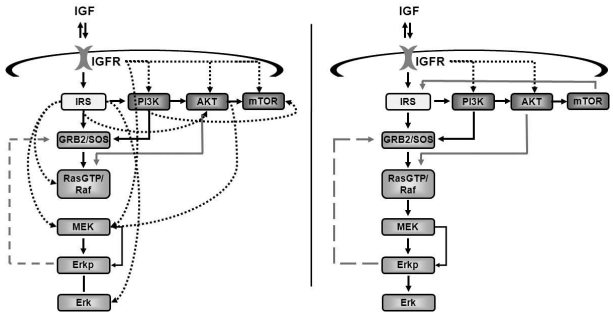

Figure 2. *BN and GBN results from synthetic data, using multiple perturbations per sample.* BN results are shown on the left, GBN results on the right. Broken arrows are noncausal. Missed edges are not indicated; the BN model misses 2 edges and shifts the $Erk \rightarrow Grb2Sos$ edge to Erkp, the GBN model misses no edges but shifts the$Erk \rightarrow Grb2Sos$ edge as well.

What causes the extraneous edges? We were originally surprised to see them, as the data comes from a clean, well defined system which, we had assumed, would be free of confounding elements that usually induce noncausal edges. Upon a closer look, we uncovered two likely culprits. One is the dynamics of the system, something we do not explicitly contend with in single timepoint, 'static' models, but which can have confounding effects induced by the *history* of a molecule being best represented by a different molecule, an effect called *entanglement* (see [7] for a discussion of this topic). A second is technically a causal effect, though it does not

result from enzymatic alteration of one molecule by another. This is an effect that may be present when multiple molecules interact with the same intermediate molecule. Consider for instance the edge between IGFR and Pi3k, an extraneous, noncausal edge. Both IGFR and Pi3k interact with IRS, an interaction that takes IRS out of the pool of available, free IRS. Thus, if IGFR binds an IRS molecule, the IRS is no longer available to bind Pi3k (though the total amount of active IRS is unaffected, since IRS can be active while bound to its up or downstream partners). Though IGFR is not causally affected by IRS, it is nevertheless *competing* with Pi3k for IRS binding. IGFR binding affects the effective amount of available IRS that Pi3k "sees", inducing a dependence between IGFR and Pi3k that is independent of IRS abundance. We call this effect *occupancy*, because it is the result of limited available occupancy on the intermediate molecule- if the upstream molecule is bound, the downstream one cannot bind, and vice versa. The impact of *occupancy* in terms of whether an edge is likely to appear probably depends on the specifics of the interactions. for example, the duration of binding of each molecule, and the extent to which the intermediate molecule is present in excess or is in short supply.

5.2. Single perturbations

Above, we formulate a novel algorithm also based on GBNs, but able to perform structure learning from data with just one perturbation per sample. We present BN and GBN results below, both use data from 6 total conditions. The BN result includes many fewer edges than the one found with 17 conditions, possibly because the total number of datapoints is smaller (the BN scoring scheme penalizes complexity, so some edges appear only if sufficient data is available. Though we did not enforce that an equal number of datapoints be used regardless of the number of conditions, this would be a useful idea to try). The BN result contains 7 extra edges, fails to orient some connections, and misses three connections (missed connections not shown). The GBN model contains one extra edge, misses two connections and shifts the $Erk \rightarrow Grb2Sos$ edge to Erkp as before. We note that it is not clear to us why the edge between Erk and Grb2Sos is so commonly shifted to Erkp. Occupancy does not seem to be a factor, because binding times are short and Erk is not in short supply. It may be an effect of the dynamics of the system as discussed in [7]. Regarding data requirements, we use here all available single perturbation conditions, a total of 5 perturbational and 1 observational conditions. However, the learning results are robust down to just two conditions, albeit with a loss of some

Pacific Symposium on Biocomputing 14:63-74 (2009)

edges (results not shown). Note however that the descendent information from all 5 perturbations is employed. This points to the possibility that a low throughput approach (such as western blots) could be used to detect descendants, and a high throughput approach (such as flow cytometry) employed for just a small set of perturbation conditions. This approach may save resources and reduce expense while still yielding good results.

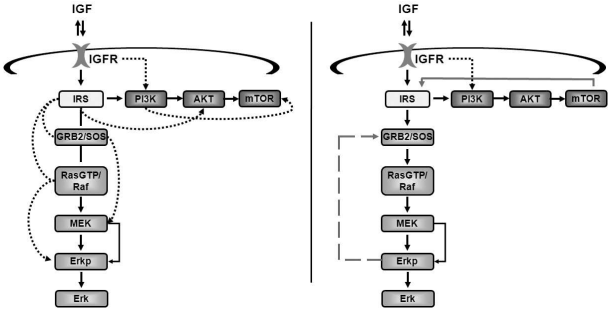

Figure 3. *BN and GBN results from synthetic data, using* one *perturbation per sample. BN results are shown on the left, GBN results on the right. Broken arrows are noncausal. Missed edges are not indicated; the BN model misses 4 edges and contains 7 extra edges, the GBN model misses 2 edges and shifts the*$Erk \rightarrow Grb2Sos$ *edge as before. It also contains one noncausal edge.*

5.3. Single perturbations with real data

To test our algorithm on real data, we employ a real-life dataset created using multidimensional flow cytometry, described in[10]. Measurements of T-cell signaling proteins are reported under observational as well as perturbational conditions with just one perturbation per condition. Perturbation conditions were available for Mek and Akt. The represented pathway is thought to contain (at least) two cycles: $Raf \rightarrow Mek \rightarrow p44/42(Erk) \rightarrow Akt \rightarrow Raf$, and $Raf \rightarrow Mek \rightarrow p44/42 \rightarrow Raf$ [10]. We focus on the proteins involved in these two cycles. BN and GBN results are shown below, with BN results on the left and GBN on the right. The BN result misses one edge and fails to orient two edges. The GBN result is nearly perfect, but it does fail to orient one edge.

6. Conclusions

In this paper, we demonstrate the first-ever application of cyclic-structure learning in signaling pathways using both synthetic and real life data, with

Pacific Symposium on Biocomputing 14:63-74 (2009)

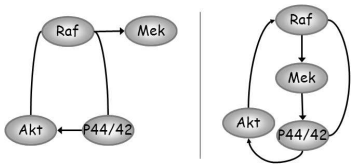

Figure 4. *BN and GBN results from real flow cytometry data, using* one *perturbation per sample. BN results are shown on the left, GBN results on the right. The BN model misses 1 edge and fails to orient 2 edges, the GBN model misses no edges but fails to orient one edge.*

score-based Bayesian networks. We present a novel structure learning algorithm grounded in the GBN formalism, and capable of handling single perturbations conditions, to reduce the algorithm's data demands. We test the original GBN algorithm on synthetic data from an accurate, differential equation model of IGF signaling. We then test the novel extended GBN algorithm formulated here, on both synthetic and real life data. In each case, our algorithm demonstrates clearly superior performance, in terms of elucidation of cyclic structures, correctly orienting model edges and even elimination of extraneous edges.

In our exploration using synthetic data, we discover unexpected edges and propose two main reason for their appearance- confounding effects of the dynamics of the system (discussed in a companion paper, [7]), and *occupancy* effects, based on multiple molecules binding to the same intermediate molecule, thus creating competition-like effects, even though not all of them may be causally affected by the intermediate molecule. This latter concept needs a more rigorous treatment, a topic that we will explore in future work. We also had available multiple timepoints from the synthetic data. For this study, a timepoint was selected arbitrarily, but the effect of timepoint selection will be discussed in [7], using the same synthetic dataset, along with the original GBN algorithm.

In the biological domain, we are often interested in a causal model, partly for the insight and understanding such a model conveys with respect to the modeled system, and partly for the possibility for system predictions which it enables. In disease states for instance, a characterization of the altered biological network can serve to guide therapeutic interventions. A truly causal model which includes correctly oriented edges is crucial- with it, a useful target can be identified and potentially detrimental effects can be avoided. Whereas previous attempts at modeling biological pathways with Bayesian networks have yielded useful results, the prevalence of cycles have

confounded those efforts, compromising the causal nature of the learned models. With this work, by overcoming the acyclicity constraint, we have brought the structure learning capability incrementally closer to learning truly causal models.

7. Acknowledgements

This work was supported by a Leukemia and Lymphoma Society post doctoral fellowship to K.S., and NIH grants N01-HV-28183, U19 AI057229, 2 P01 AI36535, U19 AI062623, R01-AI065824, 1 P50 CA114747, 2P01 CA034233-22A1, and HHSN272200700038C, NCI grant U54 RFA-CA-05-024 and LLS grant 7017-6 to G.P.N.

References

1. S. Itani, M. Ohannessian, K. Sachs, G. P. Nolan and M. A. Dahleh, submitted *NIPS* (2008).
2. M. Calder, V. Vyshemirsky, D. Gilbert, R. Orton. (2006).
3. U. Nodelman, C. Shelton, D. Koller. *UAI* (2002).
4. U. Nodelman, C. Shelton, D. Koller. *UAI* (2003).
5. Carlson, C.J., Biochem Biophys Res Comm, 2004. 316(2): p. 533-9.
6. Moelling, K., et al., J Biol Chem, 2002. 277(34): p. 31099-106.
7. Itani, S., Sachs, K., Fitzgerald, J., Wille, L., Schoeberl, B., Nolan, G. and Dahleh, M., in preparation.
8. Friedman, N. and Linial, M. and Nachman, I. and Pe'er, D. (2000). *J Comput Biol*, 3-4, Volume 7.
9. Friedman, N. (2004). *Science*, 5659, Volume 303.
10. K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan (2005). *Science*.
11. K. Sachs, D. Gifford, T. Jakkola, P. Sorger, and D. A. Lauffenburger(2002). *Science STKE*.
12. N. Friedman, K. Murphy, and S. Russell (1999). *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 139-147.
13. A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young (2001). *Pac Symp Biocomput*.
14. P. J. Woolf, W. Prudhomme, L. Daheron, G. Daley,and Q. and D. A. Lauffenburger (2004). *Bioinformatics*.
15. J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kauffman.
16. J. Pearl and T. S. Verma (1991). *Second KR*, pp. 441-452.
17. D. Heckerman, C. Meek and G. F. Cooper (1999). *Computation, Causation, and Discovery*, C. Glymour and G. F. Cooper, Eds., MIT Press, pp 141-166.
18. T. S. Richardson (1996). *UAI*, pp. 454–461.
19. Schoeberl B, Fitzgerald JB, Wille L, West K, Pace E, Harms B, Gibbons F, Donis E, Grantcharova V, Kumar A, Kudla A, Nielsen UB, Understanding IGF signaling dynamics through computational modeling, in preparation.