Using Approximate Bayesian Computation to Estimate Parameters in a Model of Idiopathic Lung Fibrosis

Team: Mariapaola Ambrosone, Robert Andrews, Mariska Batavia

Mentor: Dr. William Basener, UVA School of Data Science

Sponsor: Julie Leonard-Duke and Shayne Peirce-Cottler, UVA Biomedical Engineering

Introduction

Idiopathic lung fibrosis (IPF) is a progressive disease characterized by scarring of the lungs. It is difficult to diagnose and treat and has a median survival time of only 2-3 years following diagnosis, though the progression of the disease can be highly variable among patients (Ley et al., 2011). Patients usually present with shortness of breath and cough, and are diagnosed through a combination of imaging tests and lung biopsy. The causes are unknown, but some risk factors have been identified, such as age, smoking, exposure to airborne pollutants, and genetics (Krishna et al., 2022; Boehringer Ingelheim videos, 2016).

The most widely held belief at present is that repeated injury to the alveolar epithelium leads to the release of immune factors – such as the inflammatory cytokines TNF-alpha and IL-1B - which in turn promote dysregulated and excessive deposition of collagen by fibroblasts and myofibroblasts. As fibrotic tissue accumulates and distorts the lung's architecture, the lungs stiffen, and capacity for gas exchange by the alveoli diminishes. Understanding the biomolecular signals, interactions among cells, and other underlying factors that affect the activity of these fibroblasts is critical to understanding the progression of IPF (Krishna et al., 2022; Boehringer Ingelheim videos, 2016).

IPF is challenging to study in situ for a variety of reasons. The lungs have a complicated and spatially heterogeneous architecture, and much is unknown about different subpopulations of fibroblasts. In addition, fibrosis results from interactions among biochemical and mechanical cues, as well as among several different cell types. Given these challenges, multi-scale computational models are an essential tool to understand the interplay amongst biomolecular and cellular factors leading to disease, and to test possible targets for therapeutic intervention (Leonard-Duke et al., 2020).

Using the agent-based modeling platform NetLogo, one such model of lung fibrosis defines fibroblasts as agents, and gives them sets of rules that govern their actions and interactions over time, thereby simulating disease progression. In this model, there are two kinds of fibroblasts: Thy-1 positive and Thy-1 negative. Thy-1 is a cell surface glycoprotein expressed on the surface of many cells including fibroblasts, and its interactions with other cell surface proteins help to suppress development of fibrotic behaviors. As such, Thy-1 negative fibroblasts

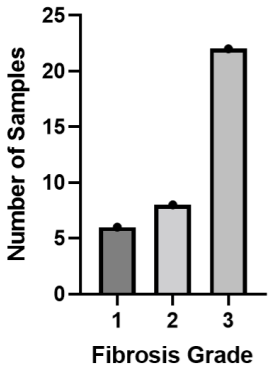
are more likely to differentiate into myofibroblasts and are associated with increased and dysregulated collagen deposition (Hu and Barker, 2019). Several model parameters can be set by the user, including the starting number of each kind of fibroblast, probability of entry into alveolar space, and the thresholds of IL-1B and TNF-alpha necessary to convert Thy-1 positive fibroblasts to Thy-1 negative fibroblasts. As the simulation runs, the model tracks tissue stiffness, alveolar patches with collagen, and the ratio of Thy-1 positive to Thy-1 negative fibroblasts. Fine tuning IPF models with biologically plausible ranges for all parameters allows researchers to explore disease progression in both slowly progressive IPF and rapidly progressive IPF, which in turn can aide in counseling patients and in the development of therapeutic approaches to disease management.

The goal of this project was to identify the posterior probability distribution for a single model parameter: the probability of a fibroblast entering alveolar space. We attempted to use approximate Bayesian computation to identify the range of values for this NetLogo parameter that can model the range of pathological findings observed in patients diagnosed with IPF.

Data Description and Methods

Approximate Bayesian computation is a method for generating the posterior probability distribution of a parameter. The values of the parameter are sampled from its prior distribution and are input into a model; simulated data generated from this model are then compared to observed data, and if the two data sets are similar, that parameter value is retained. This process is repeated many times, and parameter values that yield simulated data sets closer to the observed data are retained more often than values that yield more disparate simulated data. In this way, a posterior probability distribution for the parameter is generated (Leyshon, 2021).

To carry out this method with our IPF model, we used a validation data set comprised of 36 patient lung biopsies, with a distribution of fibrosis scores shown in Figure 1. A fibrosis score is a categorical metric (1, 2, or 3) that captures the extent of fibrotic tissue; a score of 1 is the least fibrotic, and a score of 3 is the most fibrotic. Our aim was to identify a distribution of values for the probability of a fibroblast entering alveolar space (‘as-entry-threshold' in the NetLogo model) that generated a distribution of fibrosis scores similar to the validation data set.



Group A	Group B	Group C
1	2	3
6	8	22

Figure 1. Distribution of fibrosis scores in the validation data set. Provided by Julie Leonard-Duke.

To generate simulated data with the NetLogo model ("Pulmonary_Histology_Python_7302023_abbreviated_no-endtick.nlogo"), we attempted to use the Python package nl4py (Gunaratnae and Garabay, 2021) to interact with NetLogo via Python. We sampled 100 values of as-entry-threshold from a flat beta prior distribution ranging from 0 to 5. For each of these sampled values, we ran the simulation 36 times to match the size of the validation data set. We collected fibrosis scores at time steps of 365, 730, and 1095. To determine whether a sampled value should be retained, our algorithm uses a chi-square test to compare the distribution of fibrosis scores in the validation set to the distribution of scores in the simulated data set. If $P>0.05$ (indicating no significant difference between the validation and simulated distributions), that sampled value is retained.

We attempted to generate simulation data in several experiments, varying other parameters (the proportion of Thy-1 positive and Thy-1 negative cells and the transition threshold); upon finding the distribution of fibrosis scores seemingly unaffected by these variations, we began experimenting with setup commands as well. A summary of our attempts with nl4py is shown in Table 1.

Results and Discussion

In Experiment 1, we set 100 Thy-1 negative cells and 0 Thy-1 positive cells; this biologically unrealistic scenario was intended to confirm that the model generated an overwhelming proportion of fibrosis scores of 3 (Figure 2). Since it matched our expectations, we concluded that the code and model were both working and proceeded with additional, more biologically relevant parameter values.

Table 1. Summary of experiments testing nl4py.

Experiment	Thy1ps/Thy1ns	trans-threshold	ticks	as-entry-threshold values	# as-entry-threshold samples	# replicates per sample	Comments
1	0/100	Transition=F	365, 730	0-10	10	5	Verification that we can get FS = 3
2	90/10	0.1	365, 730, 1095	0-10	100	36	
3	100/0	0.1	365, 730, 1095	0-10	100	36	
4	90/10	0.2	365, 730, 1095	0-5	100	36	
5	100/0	0.2	365, 730, 1095	0-5	100	36	
6	90/10	0.1	365, 730, 1095	0-5	100	36	Removed "reset-ticks" from setup commands
7	100/0	0.1	365, 730, 1095	0-5	100	36	Removed "reset-ticks" from setup commands
8 [no results]	100/0	0.2	365, 730, 1095	0-5	100	36	Moved "setup" to the end of commands
9	90/10	0.1	1095	0-5	10	36	Not parallelized
10	90/10	0.1	1095	0-5	10	36	Not parallelized
11	90/10	0.2	1095	0-5	20	36	Not parallelized

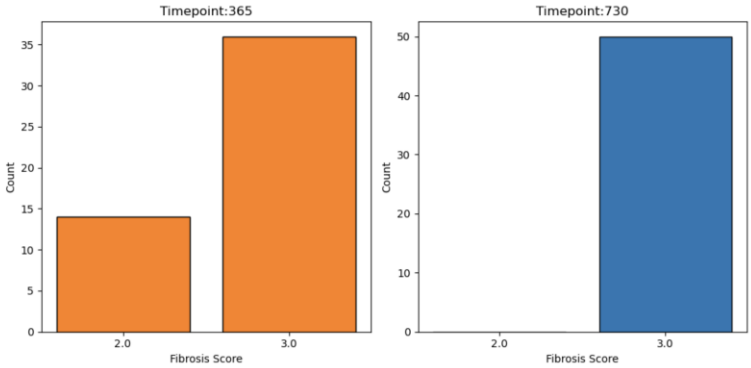


Figure 2. Distribution of fibrosis scores at time points 365 and 730 for Experiment 1, which tested a ratio of 100 Thy-1 negative cells to 0 Thy-1 positive cells. The goal was to confirm generation of fibrosis scores of 3. Other parameter values for this experiment can be seen in Table 1.

In Experiments 2 and 4, we tested a ratio of 90 Thy-1 positive to 10 Thy-1 negative cells. Experiment 2 used a transition threshold of 0.1, and Experiment 4 used a transition threshold of 0.2. We became suspicious that something was not working correctly, given that a transition threshold of 0.2 should have increased the proportion of fibrosis scores of 3. In Experiment 6, we changed the code to omit the "reset-ticks" command from the setup commands, and obtained a nearly identical result to Experiments 2 and 4 (Figure 3).

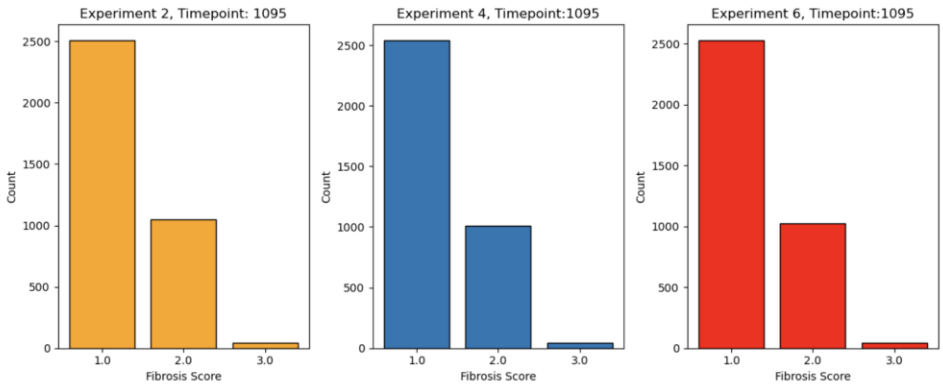


Figure 3. Distribution of fibrosis scores at time point 1095 in Experiments 2, 4, and 6, which all tested a ratio of 90 Thy-1 positive to 10 Thy-1 negative cells. Other parameter values for these experiments can be seen in Table 1.

In Experiments 3 and 5, we tested a ratio of 100 Thy-1 positive to 0 Thy-1 negative cells. Experiment 3 used a transition threshold of 0.1, and Experiment 5 used a transition threshold of 0.2. As was true for Experiment 6 (above), Experiment 7, also omitted the "reset-ticks" command from the setup commands. Similar to Figure 3, we observed a nearly identical distribution of fibrosis scores across these scenarios (Figure 4). The fact that these distributions look dissimilar from those in Figure 3 suggests that the proportion of Thy-1 positive and negative cells does impact the simulations, but other parameters and settings (transition threshold, as-entry-threshold, setup commands) do not appear to have an impact.

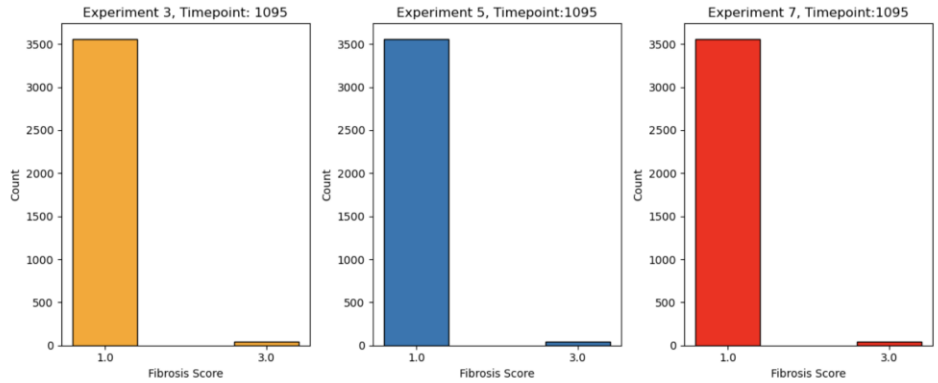


Figure 4. Distribution of fibrosis scores at time point 1095 in Experiments 3, 5, and 7, which all tested a ratio of 100 Thy-1 positive to 0 Thy-1 negative cells. Other parameter values for these experiments can be seen in Table 1.

In Experiment 8, we considered the possibility that the "setup" command might need to be last, rather than first, to instantiate the model with all our settings. We moved the "setup" command to the end of the list of setup commands; the simulations ran, but produced an

empty data frame. We concluded that the placement of the “setup” command was not the problem.

Lastly, Experiments 9, 10, and 11 ran a version of our script that did not use nl4py’s “run_experiment” function, which is intended to parallelize operations and should permit simulations to run more quickly. All three experiments tested a ratio of 90 Thy-1 positive to 10 Thy-1 negative cells. Experiments 9 and 10 were the same; each used a transition-threshold of 0.1 and sampled 10 values of as-entry-threshold. They have been combined for visualization purposes. Experiment 11 used a transition-threshold of 0.2 and sampled 20 values of as-entry-threshold (Figure 5). Once again, we see that fibrosis scores are overwhelmingly biased towards 1, with very few values of 3 (proportionally, even fewer than when we used the parallelized script, as seen in Figure 3). We also see that the numbers are nearly identical whether the transition threshold is 0.1 or 0.2.

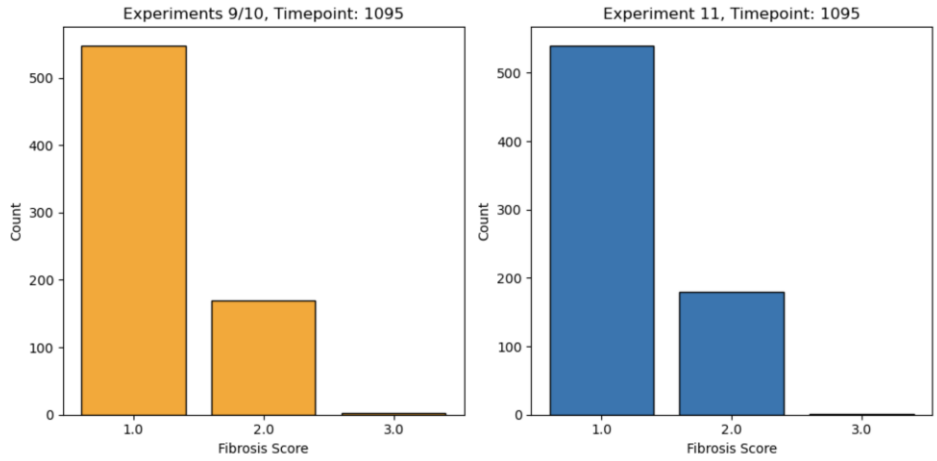


Figure 5. Distribution of fibrosis scores at time point 1095 in Experiments 9/10 (combined), and 11, which tested a ratio of 90 Thy-1 positive to 10 Thy-1 negative cells with an unparallelized script. Other parameter values for these experiments can be seen in Table 1.

Collectively, the data we obtained from all these experiments suggest that while the proportion of Thy-1 positive and Thy-1 negative cells does impact the simulation, other parameters and settings appear not to be instantiated before the model generates simulated data. With the distributions of fibrosis scores we obtained from all runs, we were unable to use our algorithm for generating a posterior distribution, as all distributions of simulated fibrosis scores were highly significantly different from the validation data.

All results files, the parallelized and un-parallelized nl4py scripts, and Jupyter notebooks for our approximate Bayesian computation algorithm and visualizations can be found on GitHub: https://github.com/mbatavia118/Approximate_Bayesian_Computation_Capstone_Summer2023.

We briefly attempted to deploy the Net Logo Behavior Space (BS) user interface in hopes of generating data more similar to the validation set. Like nl4py, Behavior Space lets the user pass a range of parameter input values to incorporate during experimentation. The interface allows

users to manipulate most of the conditions that nl4py allows. BS generates at least one experiment per combination of varying parameter values, but it also allows for multiple repetitions through the entire combination of candidate parameter values. Unlike nl4py, BS only permits the user to collect information in two ways: iteratively at each time step or at the end of each experiment cycle. In our case, collecting data at each time step was the better option as we wanted to investigate how fibrosis severity differs over time. However, collecting data at each time step was computationally expensive given the number of experiments required to generate results comparable to real-world observations. While BS allows for experiments to run in parallel, it is not as efficient as the headless workspace offered by nl4py.

We passed the same parameter variations to the BS experiment setup. Due to time constraints, we were not able to conduct enough experiments to make equal comparisons to the nl4py results, however, the smaller data we did gather was still biased toward fibrosis scores of 1. Additionally, it generated a few instances of fibrosis score 3. While we did not fully realize the user-friendly benefits of Behavior Space, we do believe that a full comparison between BS simulated data and nl4py simulated data is warranted to ensure the accurate functioning of nl4py.

Conclusions

The most significant problem we encountered during the course of this project was the lack of detailed documentation for the package nl4py. Much of the functionality we were eventually able to attain was discovered through trial and error, and some issues remain. For others moving forward with this project, we recommend running small sets of sampled parameters though both the parallelized nl4py code and NetLogo's Behavior Space, and assessing whether the proportions of fibrosis scores are similar. If they are, this would indicate that the code works as expected and that the difference between the simulated distributions and the validation data is attributable to the model itself, rather than nl4py. If the results obtained from Behavior Space and the parallelized code are different, this signals additional troubleshooting that needs to be done with the code itself.

An additional challenge that made debugging time consuming and challenging was the computational intensity of running the NetLogo model, and the need to use a high-performance computing cluster. This problem was partially solved with the abbreviated version of the full model, which eliminated the need for each fibroblast in the model to call Python repeatedly. We recommend continuing to troubleshoot with this version of the model.

Once the nl4py issues are resolved, a potentially interesting future direction would be to optimize several model parameters simultaneously. To do this, the sampling method employed by our algorithm (randomly selecting 100 values from a flat distribution) would be inefficient, given the number of possible combinations of parameter values. Using a package like pymc3

would allow for other, more targeted sampling algorithms – for example, Gibbs sampling – where the selection of sampled parameter values is conditional on the performance of previously tested values.

References

Boehringer Ingelheim Idiopathic Pulmonary Fibrosis. November 29, 2016. Introduction into idiopathic pulmonary fibrosis (IPF). (n.d.). Www.youtube.com. Retrieved August 10, 2023, from https://www.youtube.com/watch?v=QdwuHKwOLRU

Gunaratne, C. & Garibay, I. (2021) NL4Py: Agent-based modeling in Python with parallelizable NetLogo workspaces. SoftwareX, 16, 100801. ISSN 2352-7110, https://doi.org/10.1016/j.softx.2021.100801.

Hu, P., & Barker, T. H. (2019). Thy-1 in Integrin Mediated Mechanotransduction. Frontiers in cell and developmental biology, 7, 22. https://doi.org/10.3389/fcell.2019.00022

Krishna R, Chapman K, Ullah S. Idiopathic Pulmonary Fibrosis. [Updated 2022 Aug 2]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK448162/

Leonard-Duke J, Evans S, Hannan RT, Barker TH, Bates JHT, Bonham CA, Moore BB, Kirschner DE, Peirce SM. Multi-scale models of lung fibrosis. Matrix Biol. 2020 Sep;91-92:35-50. doi: 10.1016/j.matbio.2020.04.003. Epub 2020 May 11. PMID: 32438056; PMCID: PMC7434655.

Ley B, Collard HR, King TE Jr. Clinical course and prediction of survival in idiopathic pulmonary fibrosis. Am J Respir Crit Care Med. 2011 Feb 15;183(4):431-40. doi: 10.1164/rccm.201006-0894CI. Epub 2010 Oct 8. PMID: 20935110.

Leyshon, T. The ABCs of Approximate Bayesian Computation. March 14, 2021. In Towards Data Science [Internet]. Available from: https://towardsdatascience.com/the-abcs-of-approximate-bayesian-computation-bfe11b8ca341