Microbe GANtics: Using Machine Learning to Generate Synthetic Microbial Growth Data

Ismail Ameen
Fred Guichard (Supervisor)
BIOL 466
Summer 2022

Abstract:

The collection and analysis of field data serves as a backbone in ecological research, however the costs and logistics required for successful field studies make the acquisition of this data difficult. Recent innovations in Machine Learning have allowed for quality synthetic data generation via Generative Adversarial Networks (GAN). While most GANs deal with image data, the development of several Tabular GANs presents a potentially low cost, effective method to pad field data; reducing the cost of field excursions. Here, we trained 3 Tabular GANs from the tabgan python library on microbe growth data. The synthetic data were plotted against the real data, and Kullback-Leibler Divergence was used to assess both overall, and local closeness to the real data. Although the synthetic data failed to accurately learn the real data, the diagnosis of where the algorithms failed reaffirm the potential utility of GANs in the field of ecology.

Introduction:

The Costs of Field Studies:

Within ecology, the foundation of knowledge stems from biological data obtained in the field. Indeed, the information gathered from natural ecosystems informs multiple aspects of ecological research, from developing models to conservation policy[1,2]. Although field data is essential to research, there are several challenges that arise in the process of gathering it. First, the labor and monetary costs of field research present a formidable obstacle with the relatively limited amount of funding available[3,4]. This cost requires a great amount of organization and effort to deal with, particularly when considering the importance of spatial scales in collecting ecological data[5]. Trends will only become visible at certain spatial resolutions[5], which can cover vast distances[6], presenting major logistical problems. Monetary and spatial challenges are compounded when taking the temporal scales required to observe ecological trends into account. The effects of ecological shifts may not be fully realized for decades after they initially occur, necessitating long term studies. This applies to a multitude of ecological topics including animal, plant, and community level studies[6–8]. Thus, the combination of spatial and temporal scales exasperates the existing difficulties of field research. This predicament is part of the reason why, in addition to more sophisticated methods of analysis, there has been an increased focus on ecological modeling over the past few decades[9]. While modeling and simulation are undoubtedly essential to ecology, field data provides a backbone for the subject. As a result, finding methods to offset the cost of obtaining field data while maintaining data quality is an extremely important task.

The Utility of Microbe Studies:

A common method of avoiding the costs of field studies while still obtaining real data to validate models is to work with systems characterized by faster timescales and smaller spatial scales[10]. These systems can also be easily manipulated to reduce complexity and noise[11]. Microbe communities like *Saccharomyces cerivisae*, have been utilized in this capacity by ecologists since the famous Monod experiments. In regards to concepts like resource competition and community growth dynamics, microbe communities allow for rapid, and easily replicable methods to observe these phenomena[10]. This provides a relatively low cost framework to refine

models before going into the field to test them. Beyond serving as an intermediary prior to field studies, laboratory ecology, in its own right, has allowed for the identification of unique ecological outcomes [12,13]. Thus, the study of microbe communities is an essential part of ecological research; both as a validation step, and a wellspring of novel phenomena. Still, conclusions drawn from microbial experiments do not necessarily allow for direct conclusions about other ecological systems.

Machine Learning to Offset Costs:

Machine Learning could provide an additional tool to help offset the cost of field research. The release of Ian Goodfellow's seminal paper on Generative Adversarial Networks (GANs) marked the start of a new branch of Machine Learning. The basic structure of GANs follows a competition between two neural networks. Given a set of real data, a "Generator" learns to create synthetic data, which is then mixed with the real data. This combined dataset is then presented to a "Discriminator" whose goal is to correctly separate the real from the synthetic. After each round of training, both algorithms update their hidden layers and the cycle repeats[14]. This process produces results most famously seen in the creation of hyper-realistic synthetic images[15]. Assessing the quality of a GAN's synthetic data is intuitive for images due to their emphasis on visuals, making them a popular source of data when constructing new algorithms. As a result, there has been a rapid proliferation of GANs specialized in image generation. The ability to generate realistic synthetic data, however, goes beyond fooling humans, and could be an indispensable tool when applied to tabular data.

Generating tabular data incurs challenges that are absent from image data. The first is that the types of data found in tabular datasets can come in multiple varieties. Numerical, categorical, time, text, and other data types make it difficult to design an algorithm that can learn the "meaning" of the data[16]. The second challenge is designing an algorithm that can learn the variety of shapes tabular data distributions can take on[16].

Despite these challenges several TGAN python libraries have been made available for public use. In particular, *tabgan* provides a user-friendly workflow to train a Conditional TGAN (CT-GAN) on tabular data[17]. One of the advantages of CT-GANs comes from the conditional vector from which it gets its name. The vector allows the algorithm to isolate specific columns of data during training, which prevents the variable data types present in tabular data from

influencing each other simply due to scale differences[17]. *Tabgan* also works to overcome the distribution variety problem of tabular data by applying "mode-specific normalization."

When thinking about using ML to offset field research costs, TGANs present themselves as an appealing option. Given their ability to generate realistic synthetic data, one could use a TGAN trained on data collected in the field to bolster an existing dataset. This would reduce the amount of data needed to draw conclusions for various studies, from observing trends in population growth, to predicting trait abundances in the future. The process of using a TGAN in this way is in its nascent stages, and as such the work detailed in this paper works to address some of the early questions that come with developing a new tool for research.

Experimental Framework:

The comparative lack of literature on TGANs, requires important dataset considerations when conducting a proof of concept experiment like this one. To account for reproducibility, a publicly available dataset should be used. Additionally, since the goal is to eventually develop a tool to be used by field researchers, the dataset cannot be too large as that would defeat the purpose of using a GAN altogether. Finally, since the evaluation of the algorithm's performance is of the utmost importance, a dataset with a known distribution, supported by mathematical models is ideal. Microbe data fits these criteria, and can be used in a method similar to validating mathematical models. The controlled environment, distinct growth curve, and lack of noise in these datasets pairs well for an initial analysis on the performance of a TGAN while still allowing for some discussion of ecology. Here, we sought to determine how well a TGAN could replicate ecological data. A CT-GAN was trained on four microbe growth datasets with various chunks of data missing to assess whether the algorithm could infer the distribution even with key features missing. Three versions of the algorithm were run, and their performance was assessed through qualitative comparison as well as Kullback-Leibler Divergence. Ultimately, performance was not exceptional. Still, the failures shown by the results suggest that a ground up approach to algorithm design will allow a TGAN to better handle the temporal aspects of ecological data, implying that TGANs could become useful in supplementing ecological datasets as they become more sophisticated.

Methods:

The Data:

Data for the experiment was obtained from *"Dynamic metabolic adaptation can promote species coexistence in competitive microbial communities"*[10]. The publicly available dataset consisted of colony growth measured in cells/mL. Measurements were taken every 10 minutes for ~12 hours, and the experiment was repeated for 8 replicates. Thus, the training dataset for the algorithm was an 8 X 428 matrix. As mentioned in the introduction, the smaller size of this dataset was a deliberate choice since TGANs would be most useful when applied to similarly sized field datasets.

Tabgan Framework:

Before running the algorithm, the raw dataset was formatted for the *tabgan* library. As an extra precaution to avoid skewing the data during training, the "Time" column was removed. Additionally, three "cut" datasets were created to observe how the algorithm performed with key features of the sigmoidal curve missing (*Figure 1*). The first, second, and then both inflection points of original data were removed, resulting in four separate datasets upon which the algorithm would be trained on.

For each dataset (Original, Cut 1, Cut 2, Cut 1 & 2), three CT-GANs were fitted using the *tabgan* library[18]. One with standard parameters, one with only adversarial filtering, and one with no adversarial filtering. For the sake of readability, these algorithms will be referred to as Algorithm 1, Algorithm 2, and Algorithm 3 respectively. The purpose of the three algorithms is to attempt to discern a middle ground between overfitting and model performance as adversarial filtering helps reduce overfitting at the cost of performance[19]. Aside from these key differences all other model parameters were kept the same (*Figure 2).* This resulted in 12 synthetic datasets to compare against the original data.

Post Processing:

Prior to data analysis, postprocessing of the raw synthetic data was completed. The "Time" columns were added back, and the row means were calculated to obtain an average across all eight replicates. Finally, to help control the noise of the synthetic data, Loess

smoothing with a 0.65 coefficient was applied to all datasets, with the exception of Algorithm 3 datasets which used a coefficient of 0.01.

Analysis:

Colony Density vs Time plots were created to compare each model's performance when trained on the four datasets. To determine how well the synthetic data matched the real data Kullback Leibler Divergence (KLD) was used. Also known as "Information Gain", KLD provides a statistic on how one distribution is different from a reference distribution (source), where a KLD of 0 implies both distributions are the same. In the case of comparing synthetic and real data, the synthetic data satisfies the test distribution while the real data serves as the reference distribution. In addition to determining the KLD of entire datasets, Kullback-Leibler Divergence was also performed at the "Cut" portions of the data to determine how well the synthetic data captured key features present in the real data.

Results:

Complete data:

After being trained on a complete dataset, Algorithm 1's synthetic data was plotted against the other algorithms and the real data (*Figure 3*). As seen in the plots, Algorithm 1 struggled to generate accurate initial values. Furthermore, during the plateau portion of the timeseries, the synthetic data oscillated from more accurate large values to extremely inaccurate low values. This behavior is emphasized by the parabolic shape in the smoothed plot, as the high and low values prevented the smoothed curve from achieving the expected sigmoidal shape. These inaccuracies are captured by the overall KLD of 1.316 (*Table 1*), while smoothing reduced the number of extreme values allowing for a KLD of 1.0079 (*Table 2*). Due to the density of points generated around the inflection points Algorithm 1 achieved lower KLDs around each inflection point (KLD = 0.304 & 0.160 respectively), however consulting the plot indicates that the shape of the inflection points were only roughly captured around Inflection 2.

Algorithm 2's synthetic data fared only marginally better in terms of overall KLD (*Table 1, Table 2*), however its divergence was very high around the inflection points (*Table 3, Table 4*). When consulting the plots the large KLD around the inflection points is supported by the

distance between the synthetic data and the real data at those portions of the timeseries. Still, Algorithm 2's synthetic data appears less likely to undergo frequent, extreme oscillations, having captured disjoint portions of the sigmoidal curve.

Algorithm 3 generated the closest synthetic data when compared to the real dataset. As seen in the plots, Algorithm 3's density curve almost completely overlaps with the real dataset (*Figure 3*). Given that a KLD of 0 denotes an identical dataset, Algorithm 3's KLD of 0.008 (table 1) confirms that it generated an almost identical distribution to the real data. Additionally, Algorithm 3's data maintained its closeness to the real data around the inflection points, achieving a KLD of 0.0002 (*Table 4*) around the second inflection.

Cut 1:

With the Inflection 1 being removed from the training data, Algorithm 1 produced a slightly less oscillatory plateau phase. This contributes to its slightly lower overall KLD of 1.203, which should be taken into account with the fact that its training data had a KLD of 0.116 when compared to the complete dataset. Despite an improvement in overall KLD, the scores for Inflection 1 and Inflection 2 are quite high, being 1.028 and 1.400 respectively (*Table 3, Table 4*). These results become clear when looking at the plot of the data which shows that while Algorithm 1 had fewer oscillations, it generated almost no accurate data at Inflection 1, and only low values at Inflection 2.

Algorithm 2 saw improvement in generating data at the start, and Inflection 2 in the timeseries (*Table 4*). The effects of these improvements are seen in the smoothed curve (*Figure 4*) which showcases steep growth at the start of the timeseries before leveling off. Algorithm 3 was able to capture an accurate shape when compared to its training data, however it is slightly offset along the Time axis (*Figure 4*).

The major reason to create the "Cut" datasets was to determine if the algorithms could infer and generate data despite it being missing in training. As seen by the large KLDs for Inflection 1 (*Table 3*), and by the gap in the plot (*Figure 4*), the algorithms were unable to generate data to fill in the missing inflection points. This result was consistent throughout the rest of the "Cut" datasets.

Cut 2:

The removal of Inflection 2 from the training data brought back the extreme oscillations in Algorithm 1's synthetic data leading to an overall similar performance compared to being trained on the complete data. Notably, KLD around the Inflection 2 was extremely high at 2.030 (*Table 4*).

Algorithm 2 also performed poorly all around with the removal of Inflection 2, especially with an increased amount of very high value data at the outset of the timeseries (*Figure 5*). Interestingly, however, Algorithm 2 was able to achieve a lower KLD around Inflection 2 than when trained on the complete dataset. This is likely due to the presence of more high value training data which will be discussed further on.

Cut 1 & 2:

The removal of both inflections led to similar KLD values and data distribution in Algorithm 1's generated data. The one notable result was that Algorithm 1 was less impacted by the removal of Inflection 2 compared to Inflection 1 when both inflections were absent from training (*Table 3, Table 4*). This result also holds for Algorithm 2 with a KLD significantly lower around Inflection 2 when compared to Inflection 1. Additionally, Algorithm 2 surprisingly performed its best when trained on data missing both inflections (KLD = 0.723). This is partially due to the absence of extremely large data values at the start of the timeseries (*Figure 6*) which allowed for Algorithm 2's data to be closer to the real data overall.

Discussion:

Overfitting vs. Performance:

In order to explore the potential of generating synthetic data to bolster field datasets, three CT-GAN algorithms were trained on microbe growth data and the results were analyzed for distribution closeness and shape similarity compared to real data. Across all training datasets, Algorithm 3 achieved the closest synthetic data to the original dataset, however this is likely due to overfitting. A common problem when assessing neural networks like GANs, oftentimes the algorithm will become overly biased by traits of a specific dataset in order to produce "better" results[19]. As a result, Algorithm 3 loses generalizability due to said overfitting. This prevents it from being considered a useful synthetic data generator for ecological studies because without

generalizability, Algorithm 3 is not truly learning how to generate ecological data. If implemented in actual research, Algorithm 3 could easily latch onto noise in a dataset and obscure otherwise visible trends in field data.

A solution to prevent overfitting in neural nets is to allow for *Adversarial Filtering*, which was used in Algorithms 1 and 2. Adversarial Filtering, however, does come at the cost of model performance which was evident throughout the results. With large KLDs, even when accounting for the KLD of the training data compared to complete data, both Algorithms 1 and 2 failed to generate data that could confidently be used to supplement a real dataset. Still, it is more logical to consider unbiased models with poorer performance than a highly biased model when considering how to improve an algorithm. Following this logic, the rest of the discussion will primarily consider Algorithm 2 because it, generally, performed the best while avoiding overfitting.

Lost Inflection Points:

One of the goals of this experiment was to determine if the *tabgan* library could infer the existence of an inflection point despite not being trained with one present. After training on the "Cut" datasets, it is obvious that none of the Algorithms had been able to fill in data when it was absent during training. Indeed, the clear gap present in Algorithm 3's data is further evidence of it overfitting. Adversarial Filtering prevented Algorithm 2 from simply copying the training data, but the algorithm still struggled with deciding what data values should be generated when the inflection point was missing. This result highlights the importance of how the "Cut" datasets exposed two issues with Algorithm 2. The first is that Algorithm 2, and Algorithm 1, was not able to completely learn the time series aspect of the dataset. Algorithm 2 ignored the start of the timeseries when Inflection 1 was missing (*Figure 4*), and produced a sudden excess of extremely low values in the middle of the time series when Inflection 2 was missing. If the algorithm had properly learned from the training data, it would have recognized the monotonic increasing nature of the data, and done a better job of avoiding sudden extreme value changes.

The second issue exposed by the "Cut" datasets was how the overall distribution of synthetic data can be heavily impacted by the removal of key chunks of training data. Most of the training data consisted of large numbers due to the nature of microbe growth. This pushed the synthetic data of Algorithm 2 to generally favor larger values. Especially when it was trained on

data missing the lower values of Inflection 1, Algorithm 2 was better able to focus on learning the latter half of the timeseries, resulting in improved KLD (*Table 1*). The removal of Inflection 2 showed how sensitive Algorithm 2 was to changes in training data, as the more even ratio of large and small values led to the worst KLD for Algorithm 2 since it prioritized small values despite them making up a less significant portion of the data.

Interestingly, Algorithm 2's best performance both qualitatively and quantitatively was when trained on a dataset missing both inflections. This is likely due to the compartmentalization of the training data into 3 distinct groups. As a result, Algorithm 2 was able to more closely match each group (onset, growth, plateau). Unfortunately, although it was able to more closely match the data, Algorithm 2 was unable to fill in the inflection points. Still, this result has important implications for creating a more robust training process.

Ecological Significance:

The issues brought up and explained above become all the more significant when considering the ecological aspirations of this experiment. A large portion of ecological research revolves around questions of change over time. Thus, time series are an integral part of many ecological datasets. The algorithms' inability to "learn" the time series nature of the training data, evidenced by the extreme value changes, would cause massive problems if it were used in research. For example, a researcher who uses the *tabgan* library to pad a population time series may see sudden population drops. This would indicate some sort of severe ecological event, when in reality it is due to a lack of sensitivity in the algorithm. In the same vein as the lack of time series sensitivity, the failure of the algorithms to capture the inflection points in the training data implies that if given field data they will make the same mistake. Furthermore, because trends in field data may be noisier than microbe data, the potential of the synthetic data obscuring trends that could have already been visible increases.

Finally, the algorithms' tendency to be biased by the distribution of real data when generating synthetic data would have dire consequences if they were used to pad datasets in their current state. It is possible that a field dataset could deviate from what was expected due to external reasons[20,21]. If the algorithm was just trained on the deviant field data, then it would be biased to generate those deviant values. This means the algorithm would be emphasizing deviations and discrepancies that may not accurately represent ecological reality. Even at a time

when TGANs are more robust, it is imperative to consider these hypothetical situations as they outline the limitations of this technology when being applied to ecology. Ultimately, the shortcomings of the results in this experiment provide important insights to contexts where a TGAN can be confidently used to pad ecological data. They also help to outline the next steps to generate better synthetic data.

Future Work:

The suite of improvements that can be made to generate better synthetic data all stem from building a TGAN from the ground up, as this would allow for an algorithm customized to deal with ecological data. The issues of incomplete data, and reduced dataset size posed a serious problem for the *tabgan* library. Fortunately, these issues have been reduced through existing image GAN frameworks in the form of MisGAN and CycleGAN respectively[22,23]. These architectures can be integrated into a TGAN that is built from the ground up, allowing the algorithm to better cope with incomplete and smaller datasets.

An algorithm built from the ground up also allows for a better training regiment. *Tabgan* utilizes a conditional vector to treat dataset columns as “images". This prevents customized training for the data one is trying to replicate since only a standard tabular dataset can be submitted for training. By allowing for an algorithm to be trained on multiple tabular datasets, it would be possible to establish a baseline understanding of the ecological history of a field site. This would help improve algorithm performance when generating data for previously studied field sites.

To combat the problem of being biased by certain types of data, an improved TGAN would allow the introduction of conditional computation into its architecture. This would split the generator into several networks that would separately learn key aspects of the dataset before being merged together for estimation[24]. By splitting the task of learning data into several networks, estimation accuracy would improve, as well as computation time[24].

Finally, the generation of extreme outliers at illogical points in the time series was a major problem in this experiment. A custom TGAN would allow the imposition of minimum and maximum thresholds on generated data. This would allow researchers to give extra context to the generator. Particularly in time series studies where the data is expected to follow a certain shape

(i.e monotonic increasing in the case of microbe growth), threshold values would help control the noise in the data to a tolerable level.

Conclusion:

Cost effective options to improve the quality of ecological datasets can be an incredible boon to researchers. With modern advancements in computation costs, Machine Learning appears as one such option. In this paper, the potential utility of a GAN styled algorithm was explored. A CT-GAN algorithm was trained on a relatively small microbe growth dataset, and the generated synthetic data was evaluated. While none of the synthetic datasets generated usable data, purely adversarial filtering provided the most accurate synthetic data when compared against the original dataset. Furthermore, the shortcomings encountered in the results outline the steps required to improve synthetic data quality. Ultimately, the low cost, fast runtime, and quality of the synthetic datasets suggest that GANs can be a benefit to Ecology with a more ground up algorithm design.

Appendix:

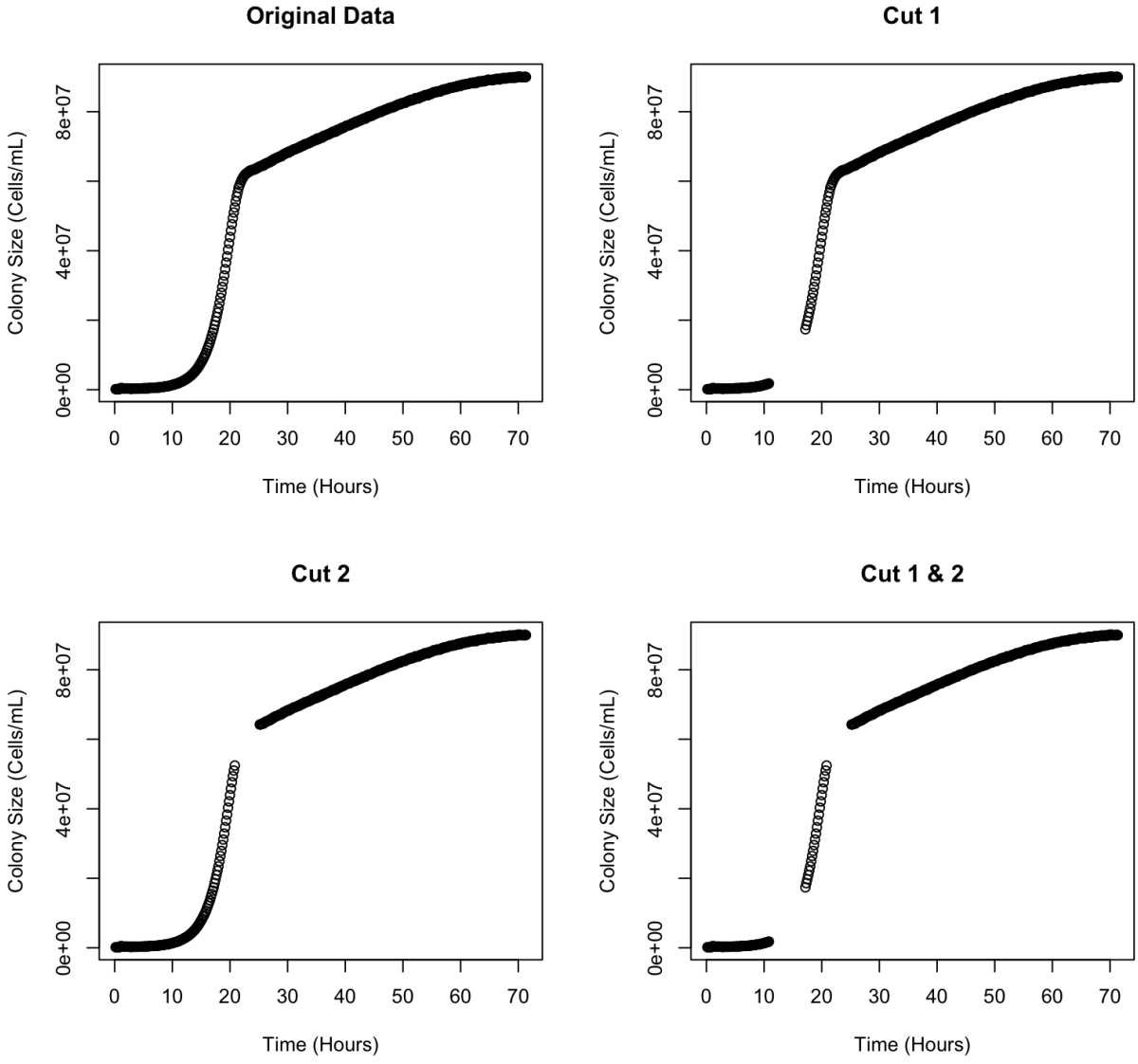

Figure 1. *Plots of the real data and the derivative "Cut" datasets that were used for training the TGANs. As seen in the plots, the key inflection points marking the start of exponential growth, and the start of the plateau were removed in three combinations.*

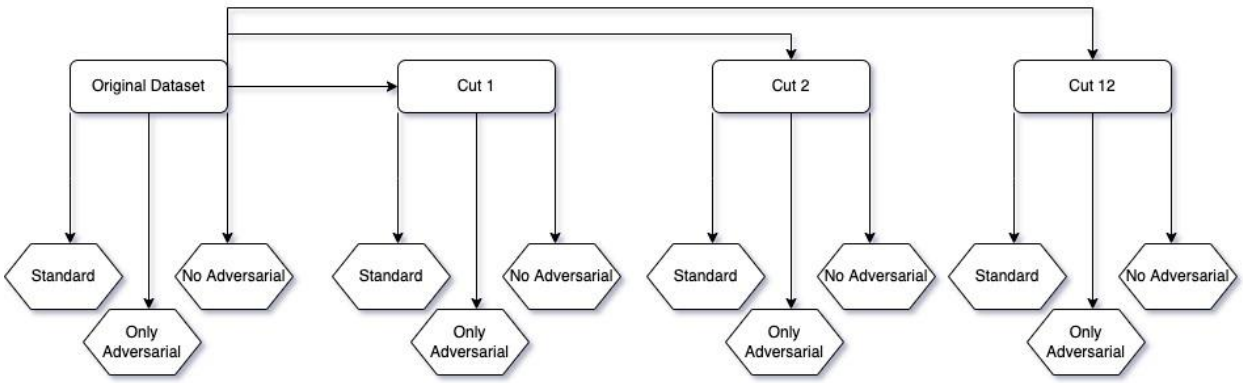

Figure 2. *Diagram of the experimental workflow. Altered datasets were derived from the original data, and all four resulting datasets were used to train the three algorithms. Since each algorithm was used to produce a synthetic dataset, a total of 12 synthetic datasets were produced.*

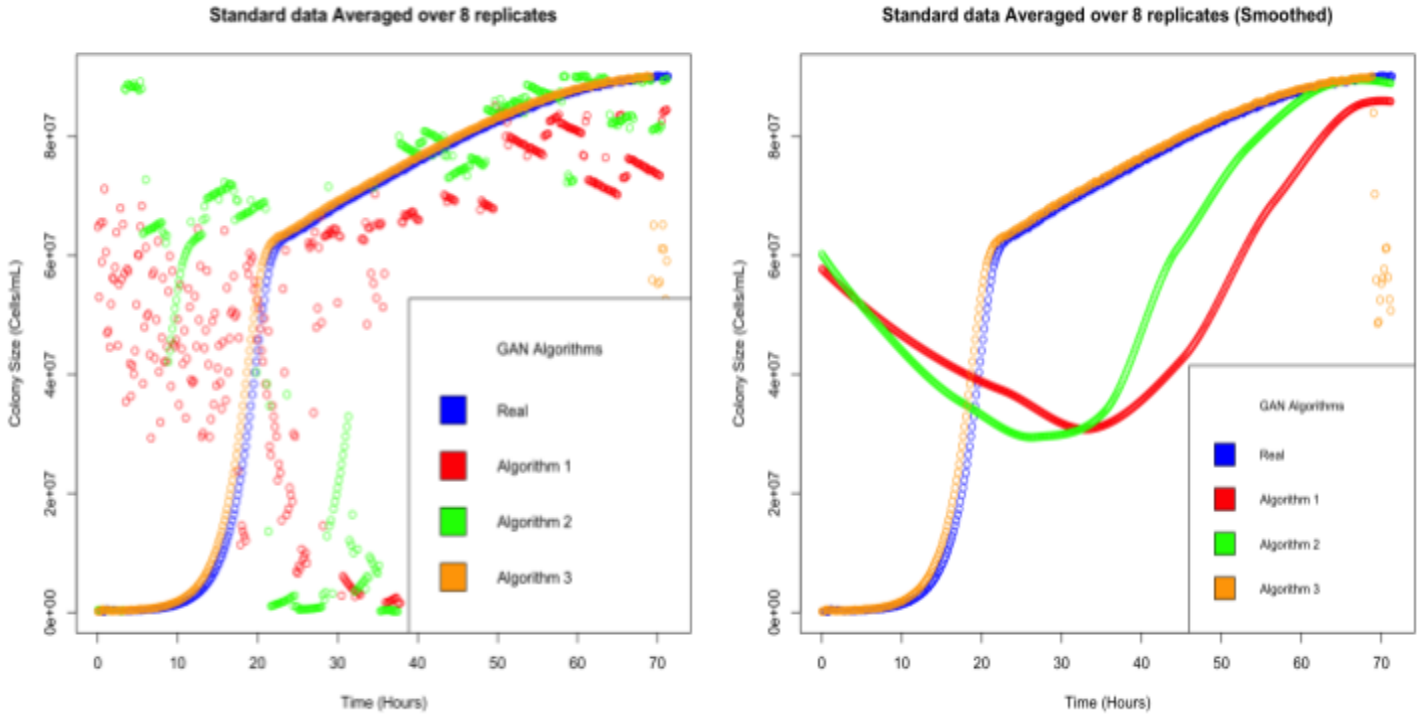

Figure 3. *Raw average and smoothed synthetic data plotted against the complete data. Algorithm 1 (red) is the standard CT-GAN, Algorithm 2 (green) is Only Adversarial, and Algorithm 3 (orange) is Non-adversarial. Algorithm 1 and Algorithm 2 failed to generate accurate initial data, however they both learned to generate the plateau region to a certain extent. Algorithm 3 generated extremely similar data when compared to the real data.*

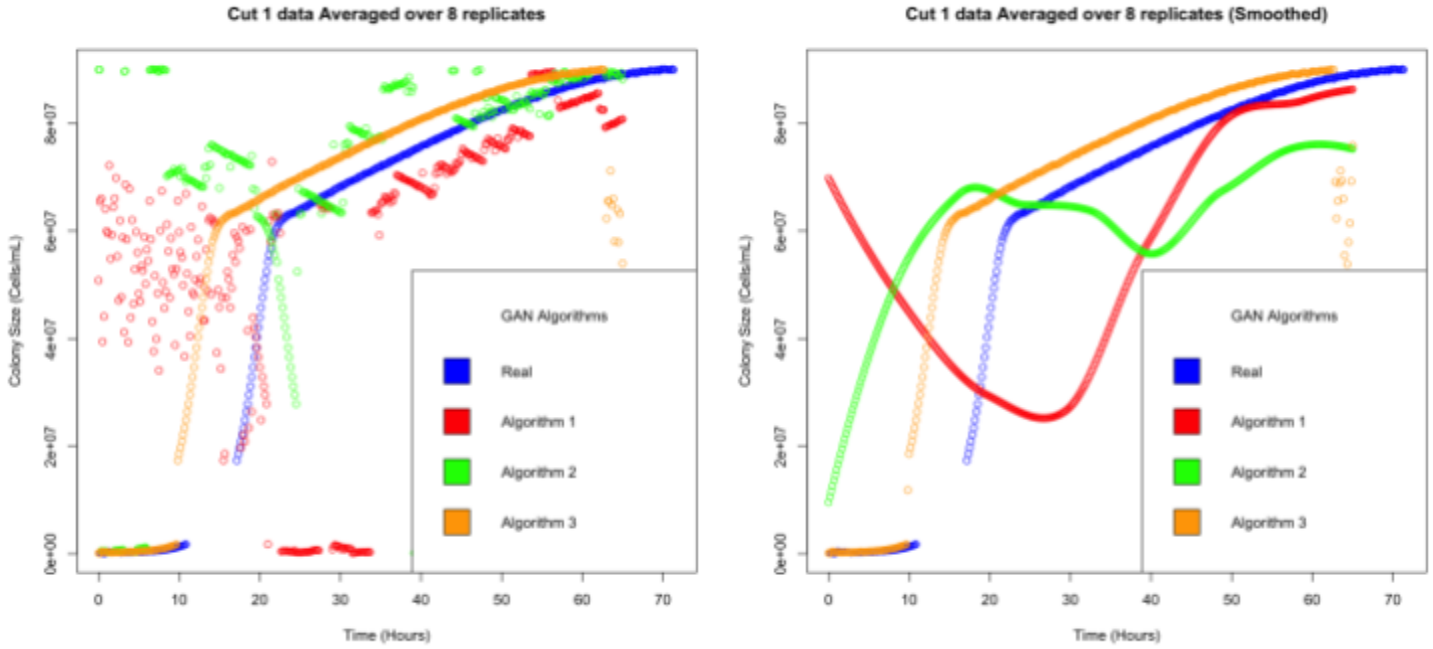

Figure 4. *Raw average and smoothed synthetic data plotted against the Cut 1 data. Algorithm 1 (red) is the standard CT-GAN, Algorithm 2 (green) is Only Adversarial, and Algorithm 3 (orange) is Non-adversarial Algorithm 1 and Algorithm 2 continue to struggle with generating the initial data, however both appear to perform better at generating data for the latter half of the growth phase, and the plateau phase. Algorithm 3 matches the shape of the Cut 1 data, but is slightly offset due to the missing data in training.*

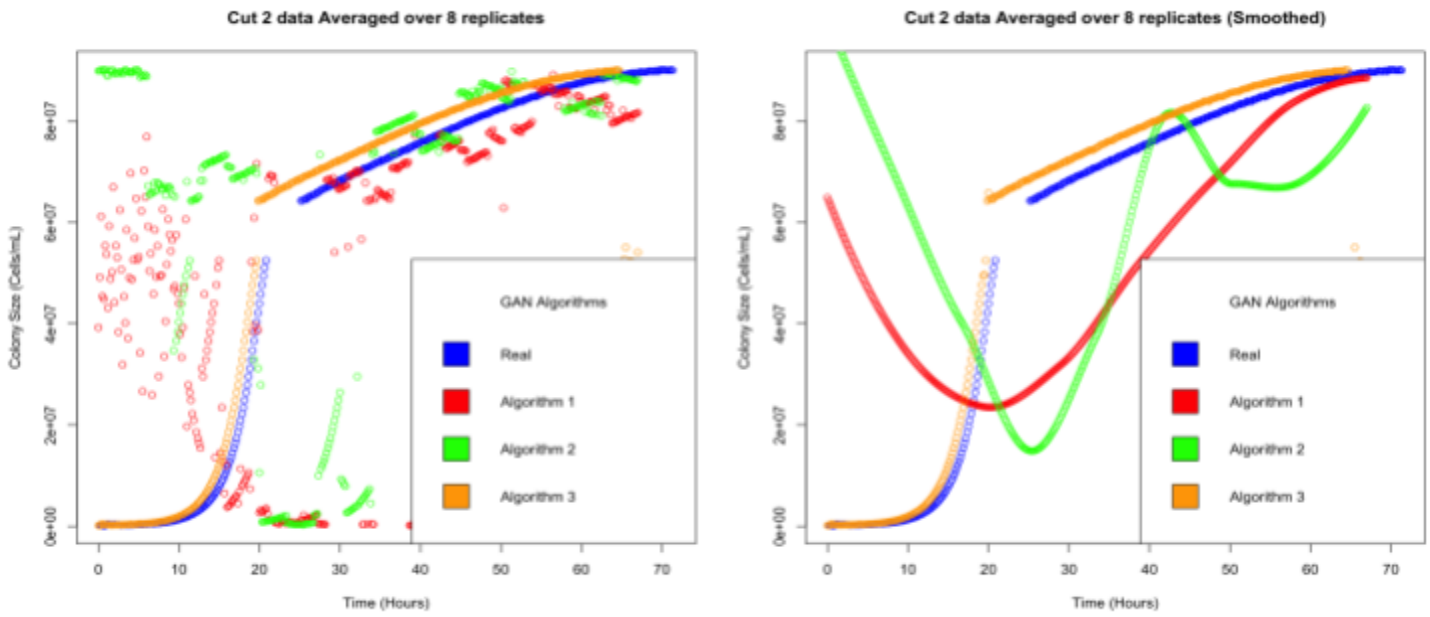

Figure 5: *Raw average and smoothed synthetic data plotted against the Cut 2 data. Algorithm 1 (red) is the standard CT-GAN, Algorithm 2 (green) is Only Adversarial, and Algorithm 3 (orange) is Non-adversarial. The loss of Inflection 2 severely impacted both Algorithm 1 and 2 as seen in the smoothed plot. Particularly in Algorithm 2, the extremely large starting values lead to a high divergence from the real data.*

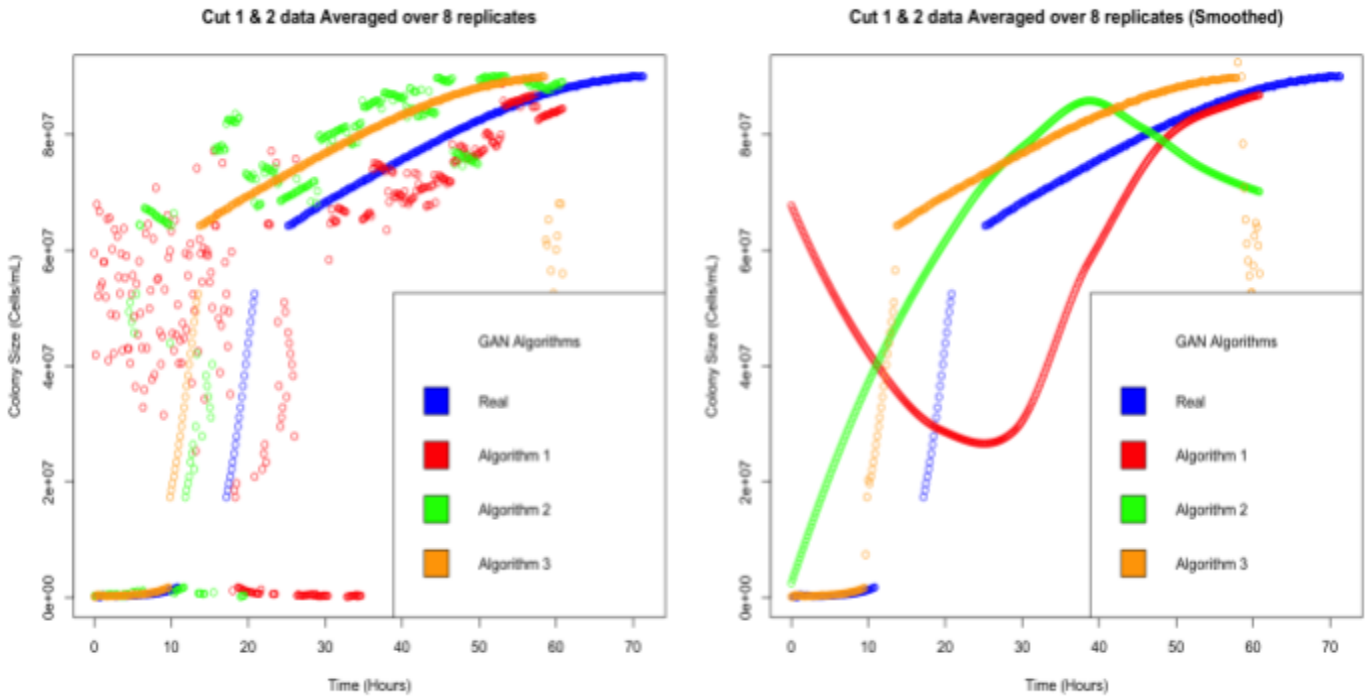

Figure 6: *Raw average and smoothed synthetic data plotted against the Cut 1 & 2 data. Algorithm 1 (red) is the standard CT-GAN, Algorithm 2 (green) is Only Adversarial, and Algorithm 3 (orange) is Non-adversarial. Algorithm 2 performed its best with the removal of both inflection points, as seen in the raw average plot. This is further emphasized in the smoothed plot as Algorithm 2 achieves its closest to sigmoidal shape.*

Overall KLD for Unsmoothed Data

Dataset	Real data	Algorithm 1	Algorithm 2	Algorithm 3
Complete	*0*	1.316	1.215	0.008
Cut 1	*0.116*	1.203	1.149	0.362
Cut 2	*0.184*	1.203	1.790	0.151
Cut 1 & 2	*0.710*	1.225	0.723	0.522

Table 1. Here, the over KLD divergences are presented for the raw average synthetic data. KLD was calculated for each algorithm trained on each dataset. Additionally, the KLD of each "Cut" dataset compared to the complete real dataset is included to provide context.

Overall KLD for Smoothed Data

Dataset	Real data	Algorithm 1	Algorithm 2	Algorithm 3
Complete	*0*	1.079	0.943	0.008
Cut 1	*0.116*	1.028	0.754	0.359
Cut 2	*0.184*	0.876	1.475	0.152
Cut 1 & 2	*0.710*	1.021	0.523	0.519

Table 2. Here, the over KLD divergences are presented for the smoothed average synthetic data. KLD was calculated for each algorithm trained on each dataset. Additionally, the KLD of each "Cut" dataset compared to the complete real dataset is included to provide context.

KLD at Inflection 1

Dataset	Algorithm 1	Algorithm 2	Algorithm 3
Complete	0.304	0.943	0.008
Cut 1	1.028	0.755	0.359
Cut 2	0.876	1.475	0.152
Cut 1 & 2	1.021	0.523	0.519

Table 3. Local KLD at Inflection 1 for each algorithm. The "Cut" datasets had a KLD of 0 if the inflection was present, or no meaningful KLD if the inflection was removed (i.e Cut 1). As such, the KLDs of the "Cut" datasets were excluded from the table.

KLD at Inflection 2

Dataset	Algorithm 1	Algorithm 2	Algorithm 3
Complete	0.160	1.633	0.0002
Cut 1	1.400	0.051	0.0005
Cut 2	2.030	0.232	0.0004
Cut 1 & 2	0.591	0.002	0.0005

Table 4. Table 3. Local KLD at Inflection 1 for each algorithm. The "Cut" datasets had a KLD of 0 if the inflection was present, or no meaningful KLD if the inflection was removed (i.e Cut 1). As such, the KLDs of the "Cut" datasets were excluded from the table.

References:

1. Tadiri, C. P., Kong, J. D., Fussmann, G. F., Scott, M. E. & Wang, H. A Data-Validated Host-Parasite Model for Infectious Disease Outbreaks. *Front. Ecol. Evol.* **7**, (2019).
2. Cooke, S. J. *et al.* How experimental biology and ecology can support evidence-based decision-making in conservation: avoiding pitfalls and enabling application. *Conserv. Physiol.* **5**, cox043 (2017).
3. Examples of Research Expenses. *Research Management Services* https://research.uottawa.ca/rms/examples-research-expenses.
4. Fieldwork in the Arctic is surprisingly costly, limiting the research done there. https://www.science.org/content/article/fieldwork-arctic-surprisingly-costly-limiting-research-done-there.
5. Rahbek, C. & Graves, G. R. Detection of macro-ecological patterns in South American hummingbirds is affected by spatial scale. *Proc. R. Soc. Lond. B Biol. Sci.* **267**, 2259–2265 (2000).
6. Copeland, S. M. *et al.* Long-term trends in restoration and associated land treatments in the southwestern United States. *Restor. Ecol.* **26**, 311–322 (2018).
7. Jablonski, D. & Sepkoski Jr., J. J. Paleobiology, Community Ecology, and Scales of Ecological Pattern. *Ecology* **77**, 1367–1378 (1996).
8. Stirling, I., Lunn, N. J. & Iacozza, J. Long-Term Trends in the Population Ecology of Polar Bears in Western Hudson Bay in Relation to Climatic Change. *Arctic* **52**, 294–306 (1999).
9. Ríos-Saldaña, C. A., Delibes-Mateos, M. & Ferreira, C. C. Are fieldwork studies being relegated to second place in conservation science? *Glob. Ecol. Conserv.* **14**, e00389 (2018).
10. Pacciani-Mori, L., Giometto, A., Suweis, S. & Maritan, A. Dynamic metabolic adaptation

can promote species coexistence in competitive microbial communities. *PLOS Comput. Biol.* **16**, e1007896 (2020).

11. Blasius, B., Rudolf, L., Weithoff, G., Gaedke, U. & Fussmann, G. F. Long-term cyclic persistence in an experimental predator-prey system. *Nature* **577**, 226–230 (2020).

12. Kovárová-Kovar, K. & Egli, T. Growth Kinetics of Suspended Microbial Cells: From Single-Substrate-Controlled Growth to Mixed-Substrate Kinetics. *Microbiol. Mol. Biol. Rev.* **62**, 646–666 (1998).

13. Dargent, F., Scott, M. E., Hendry, A. P. & Fussmann, G. F. Experimental elimination of parasites in nature leads to the evolution of increased resistance in hosts. *Proc. R. Soc. B Biol. Sci.* **280**, 20132371 (2013).

14. Goodfellow, I. J. *et al.* Generative Adversarial Networks. Preprint at http://arxiv.org/abs/1406.2661 (2014).

15. Karras, T., Laine, S. & Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. Preprint at http://arxiv.org/abs/1812.04948 (2019).

16. Xu, L. & Veeramachaneni, K. Synthesizing Tabular Data using Generative Adversarial Networks. Preprint at http://arxiv.org/abs/1811.11264 (2018).

17. Ashrapov, I. Tabular GANs for uneven distribution. Preprint at http://arxiv.org/abs/2010.00638 (2020).

18. Ashrapov, I. GANs for tabular data. (2020).

19. Bras, R. L. *et al.* Adversarial Filters of Dataset Biases. in *Proceedings of the 37th International Conference on Machine Learning* 1078–1088 (PMLR, 2020).

20. Barry, S. & Elith, J. Error and uncertainty in habitat models. *J. Appl. Ecol.* **43**, 413–423 (2006).

21. Morrison, L. W. Observer error in vegetation surveys: a review. *J. Plant Ecol.* **9**, 367–379 (2016).

22. Li, S. C.-X., Jiang, B. & Marlin, B. MisGAN: Learning from Incomplete Data with Generative Adversarial Networks. in (2022).

23. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Preprint at http://arxiv.org/abs/1703.10593 (2020).

24. Sun, Y., Wang, X. & Tang, X. Deep Convolutional Network Cascade for Facial Point Detection. in *2013 IEEE Conference on Computer Vision and Pattern Recognition* 3476–3483 (IEEE, 2013). doi:10.1109/CVPR.2013.446.