Detrended Fluctuation Analysis as Fitness Criterion for Music Generation by Cellular Automata

Author Name
Affiliation
email@example.com

Abstract

While long since outclassed in terms of the quality of their musical output, cellular automata (CA) merit continued interest as an artistic tool due to their intuitive operation, ease of modification and low computational cost. CA are typically trained to produce musical output using some form of resemblance measurement between CA output and an existent body of (human) composed music as a fitness criterion. However, this approach runs directly counter to the long history of algorithmic composition as a medium where the use of an artificial agent is seen as an emancipatory practice, intended to move beyond human aesthetic sensibilities. In recognition of this tradition we utilize a Detrended Fluctuation Analysis (DFA) as a fitness criterion in what we believe to be the first application of a DFA in the context of CA. We further test our novel fitness method against a range of CA configurations and demonstrate its robustness at finding meaningfully patterned output for music generation.

1 Introduction

In its traditional form, the operation of a elementary or 1-D Cellular Automaton (CA) begins with a $1 \times n$ vector of binary values. At each time step, cells in the vector are updated according to a fixed set of update rules such that such that the value of cell j at time $t + 1$ where $j \in \{1...n\}$ is a function of the state of j and the configuration of its neighborhood at time t. In effect rule-sets operate as a sort of lookup table. Traditionally cells are only every assigned binary values thus given a neighborhood size of m there are 2^{m+1} potential rule-sets, one for each unique configuration of neighborhood. The above schema is generalizable to an arbitrary number of dimensions and a wide array of mappings from cell states to rule-sets, perhaps most famously to the 2-D case in Conway's Game of Life [Gardner, 1970]. Here we only consider the classical 1-D case.

To generate music using a CA the typical approach as first characterized in [Beyls, 1989] is to assign a musical phoneme such as a note or instrument to a given cell j where $j \in \{1...n\}$ in the initial vector of n values at time $t = 0$ and treat the resulting $t \times 1$ time series of j as an activation pattern for the given note or instrument. By assigning a phoneme to each of the n cells one is able to generate a score in which there is dependency between activations of different instruments and thus a degree of harmony, melody and rhythm is able to emerge. Unsurprisingly, a wide range of different approaches to mapping between CA output and musical input have emerged over the years such as [Delarosa and Soros, 2020] and [Miranda, 1993], but the 'piano-roll' approach detailed above is perhaps the most common.

Over time CA have long been overshadowed by contemporary neural networks (NNs) as a means of generating music and we do not dispute the impressive ability of NNs to arrange patterns of sound as comprehensively shown in [Wang *et al.*, 2024]. However, we do wish to draw attention to the implicit false equivalence between the ability of a device to generate music and its usefulness as an *artistic* or *musical* tool. Put another way, we disagree with the notion that the automatic generation of music by NNs is useful or, for that matter conducive to the artistic process.

Considered in this light, in terms of usefulness to the musician as a tool and not robustness of sound generation, CA have several benefits overs NNs. First, unlike an NN the operation of CA with its simple mapping of cell states to rules to cell states, is intuitive thus allowing for their implementation and operation by those with only a limited coding background and for a lower barrier of entry for musicians without a robust technical background. Additionally, CA with their human interpretable rule-sets are not black boxes like NNs. An individual is able to directly intervene into their operation and modify their underlying parameters with a degree of understanding as to the outcomes. It is just this sort of second-order engagement with the medium that allows for the type of experimental and open-ended tinkering characteristic of arts practice. Lastly, CA can be trained and implemented on low-powered consumer hardware and do not require GPUs or other specialized hardware as in the case of NNs. In effect CA are far more conducive to artistic practice and excel as tools within musical experimentation as opposed to the rote composition of NNs that effectively only automate the musician's surface level activity.

In spite of this CAs can be still be improved as an artistic tool. Typically, due to the sheer number of potential rule-sets for a CA, the process of finding an ideal rule-set is automated. Consequently some form of fitness criterion is needed as a proxy for the artist's qualitative assessment to distinguish between more or less desirable rule-sets. In the literature, fitness criteria that assess the degree to which the distribution of conditional probabilities between musical phonemes across time steps in the CA output is aligned with those in a given exemplar or body of exemplars are the most common such as in [Delarosa and Soros, 2020]. In effect the degree to which the output resembles an already existing piece of music. However, this runs directly counter to the long history of algorithmic composition in music. Traditionally, from such early practitioners as Arnold Schoenberg and members of Dada all the way up through Fluxus, John Cage and more recently Iannis Xenakis, the use of algorithms in the compositional process was seen as explicitly emancipatory. A means of moving away from or even beyond the bindings of human subjectivity and aesthetic sensibilities in composition. The use of a similarity metric between CA output and existing human compositions, while being intuitive from an engineering perspective, ignores the long artistic tradition surrounding the underlying medium and makes itself deaf to the underlying poetics of the algorithm.

All the same, a fitness criterion is required to be able to feasibly train a CA for musical output. As a result, we propose the use of a Detrended Fluctuation Analysis (DFA), a means of measuring the scaling exponent α of fluctuations in a time series, as a fitness criterion. Existing studies have shown using a DFA that different genres of music inhabit different regions of values of α [Jennings *et al.*, 2003] [Streich and Herrera, 2005]. By linking fitness of CA output to particular values of α, a musician is able to still assert some degree of authorship by broadly indicating the variance in fluctuations in the output over time, but still leaving the composition of those fluctuations undetermined. In effect, we believe the DFA is far more in alignment with the spirit of algorithmic composition while still granting artists a means of authorial control desired by many. *Furthermore, to the best of our knowledge, this is the first instance of a DFA being used in the context of a CA, let alone as a fitness criterion in their training for musical output.*

In what follows we lay out in detail the configuration of CA utilized in Section 2.1, the workings of the genetic algorithm used for training the CA in Section 2.2 and in Section 2.3 we lay out the operation of our novel fitness fitness method including a step-by-step explanation of a DFA. Next in Section 3.1 we present training performance of the various CA configurations which shows the rapid convergence of fit CA. In Section 3.2 we present a selection of trained CA outputs and show the wide variety of patterned behavior that can emerge from the DFA, thus showing its viability as a fitness criterion for musical composition. Finally in Section 4 we propose two future extensions of the work, provide our thoughts on potential challenges that may be encountered as well as potential approaches to overcoming these challenges.

2 Methods

2.1 Cellular Automata

All CA used were 'elementary' or 1-dimensional as describe above except for slight modifications describe below. Rather than directly assigning values to cells using individual rules, rules represent probabilities with which a cell will be assigned the value one in the succeeding time step. We chose this approach for two reasons. First, to provide a continuous and more flexible rule-set search space given the unknown difficulty of the fitness criterion. Second to provide a higher degree of novelty at the time of trained operation, ideally aping the sort of improvisation and indeterminacy common in musical performance.

Additionally, the neighborhood size of each cell was varied across trials such that a cell's update was a function of its state and the state of its r immediate neighbors to the left and r immediate neighbors to the right where $r \in \{1, 2\}$. Consequently the size of the rule-sets changed across trials given that a rule-set will contain 2^{2r+1} individual rules corresponding to each of the potential unique neighborhood configurations. The number of initial cells n was also varied across trials where $n \in \{5, 10\}$.

2.2 Training

A genetic algorithm was used to derive the CA rule-sets. An initial population of 100 rule-sets was derived where p was assigned uniformly at random for each of individual 2^{2r+1} rules for each of the 100 sets. On each epoch each CA was initialized with a randomized starting array where each cell had the value 1 with probability $p = 0.5$ and 0 with probability $p = 0.5$ in order to ensure robustness of performance of any given rule-set. Each CA was then run for a total of 2048 time steps and the fitness of each CA was assessed using the method detailed below. The 50 CA producing the 50 least fit outputs were then discarded. From the remaining 50 CA, individuals were selected in pairs with replacement and their rule-sets were recombined using single-point crossover with the crossover point being assigned randomly to each pair. The subsequent two offspring were then mutated with probability $p = 0.1$ such that a rule from within their rule-sets was selected at random and the value m was added to it where m was selected uniformly at random from $[-0.35, 0.35]$. The above process of selection, recombination and mutation was then repeated until an additional 50 new CA were produced. The two sections were then combined to form the population for the next epoch. The total process was repeated for 100 epochs for each of the possible combinations of r and n.

2.3 Fitness Method

As mentioned above, a Detrended Fluctuation Analysis (DFA) was used to assess the fitness of each CA. First introduced in [Peng *et al.*, 1995], the DFA consists of first integrating a given time series such that:

$$y(t) = \sum_{i=0}^{t} x(i) \qquad (1)$$

where y is the initial time series and z is the resulting integrated time series. The integrated time series z is then segmented into non-overlapping blocks of uniform length γ where $\gamma \in [2^2, 2^3 ... 2^l]$ such that 2^l does not exceed $1/2$ the length of y. The process is repeated for each value of γ. From each block the linear trend is removed and the mean squared residual is calculated. Put symbolically:

$$D(h,\gamma) = \frac{1}{\gamma}\sum_{m=0}^{\gamma}(y(h+m) - (\hat{y}_h(m))^2 \qquad (2)$$

where $D(h,\gamma)$ is the computed mean squared residual of block h of size γ and $\hat{y}_h$ is the linear trend of block h.

The fluctuation for each value of γ is then calculated according to:

$$F(\gamma) = \sqrt{\frac{1}{H}\sum_{h=1}^{H} D(h,\gamma)} \qquad (3)$$

The resulting fluctuation values for $F(\gamma)$ are then placed on a log-log plot against the values of γ and a linear trend line is drawn through them. The resulting slope α, assuming a strong linear fit, characterizes the fluctuation dependencies across time scale. A value of $\alpha = 0.5$ corresponds to white noise, $\alpha = 1.0$ to $1/f$ or 'pink' noise and $\alpha = 1.5$ to brown noise [Ihlen, 2002].

The fitness function utilized here can be expressed as:

$$fitness(x) = r_x - 0.5|\hat{\alpha} - \alpha_x| \qquad (4)$$

where x is the given CA output, $\hat{\alpha}$ is the target DFA scaling exponent, α_x is the scaling exponent derived from performing the DFA on x and r_x is the goodness of fit of α_x to the actual points of the log-log plot resulting from Equation 3 across the values of γ. For our purposes $\hat{\alpha}$ was varied across $[1.00, 1.25, 1.50]$.

One particular challenge that needed to be addressed was the encoding of the $t \times n$ CA output array into a $t \times 1$ vector suitable for use by the DFA. In order to accomplish this a simple binary encoding mechanism was used such that

$$x'_j = \sum_{i=0}^{n} x_{ji} \times 2^i \qquad (5)$$

where x is the original $t \times n$ CA output, $j \in [1...t]$ and x' is the resulting encoded $t \times 1$ time series. This creates a unique encoding in x' for each potential configuration of any given time step in x. Additionally through this encoding schema, activations of cells with indices closer to n have a greater impact on the values of x' and thus the degree of fluctuation of cells closer to n have a greater impact on the degree of fluctuation of x'. Given the DFA is closely tied to the fluctuations of a given time series, it predisposes the genetic algorithm to allow for greater leeway in the amount of fluctuation of elements closer to 1 and less leeway in the amount of fluctuation of those closer to n. This uneven search terrain thus gives a musician an additional means of engagement.

3 Results

3.1 Training

A total of ten trials were run for each possible configuration of α, r and n where $\alpha \in [1.0, 1.25, 1.5]$, $r \in [1, 2]$ and $n \in [5, 10]$. In all cases a population of 100 CAs were trained for 100 epochs and the fitness of each CA was recorded at the end of every epoch. For each configuration the results across each trial were then averaged and are presented below.

Overall we see broad success on the part of the GA at finding fit rule-sets across all configurations thus showing the robustness of the modified DFA as a fitness criterion. In addition we see that as α, n and r all increase so too does the degree of difficulty for the GA. This is most pronounced with increases in α and less so with increases in n and r.

In particular we can see in Figures 1 and 2 that the genetic algorithm is quickly able to find suitable rule-sets to achieve high fitness with a median fitness of over 0.95 and over 0.9 being achieved for $\alpha = 1.0$ and $\alpha = 1.25$ respectively. In Figure 3 on the other hand we see that while the genetic algorithm is able to find relatively fit solutions (with median fitness exceeding 0.8 in all cases except for $r = 2, n = 10$) it struggles far more so than with $\alpha = 1$ or $\alpha = 1.25$. This seems to clearly indicate that $\alpha = 1.5$ and its requirements of greater fluctuation at larger time scales is a far harder criterion than the other two cases.

For all values of α we see that the GA struggles more as r and n increase. This is particularly pronounced with $\alpha = 1.5$ where at $r = 2, n = 10$ its median fitness does not exceed 0.8. Overall this is not terribly surprising, as the total number of cells and neighborhood size increases, the dynamics of the CA become more complex and thus harder to control with a single rule-set.

Lastly, it is important to address the degree of instability in the minimum fitness across all configurations. We believe this to be a result of both the probabilistic nature of the CAs update rules and the randomized initial state for each CA in each epoch. The former may have resulted in a CAs output going wildly awry due to a poor series of 'dice-rolls' on cell update. The latter may have presented an otherwise fit CA rule-set with an almost antagonistic starting condition. The chance nature of these conditions gives rise to the high degree of fluctuation in fitness minimum.

3.2 Trained Outputs

After training a final series of outputs from each CA was generated and recorded. For each configuration of α, r and n a random selection of those exceeding a fitness of 0.9 in the cases of $\alpha = 1$ and $\alpha = 1.25$ and those exceeding a fitness of 0.8 in the case of $\alpha = 1.5$ were set aside. The first 20 time steps of these outputs is presented below.

What is immediately apparent is that the DFA based fitness criterion is successful in engendering a wide variety of clearly structured output in the CA. It is critical to bear in mind that again, in keeping with the spirit of algorithmic composition as an art form the success of the DFA criterion should not be assessed in terms of how closely the outputs resemble music. Rather, success should be viewed in terms of the variety, novelty and presence of structure within the outputs and their ability to serve as points of departure in and from human composition. The CA is here being positioned as a tool, not the composer.

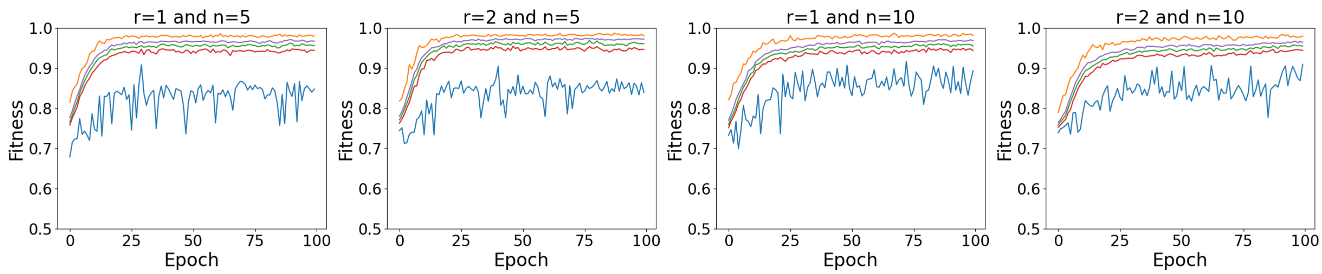

Figure 1: Training Performance with $\alpha = 1.0$ where the lowest fitness of each epoch is represented in blue, the 25th percentile in red, the median in green, the 75th percentile in purple and the highest fitness in orange

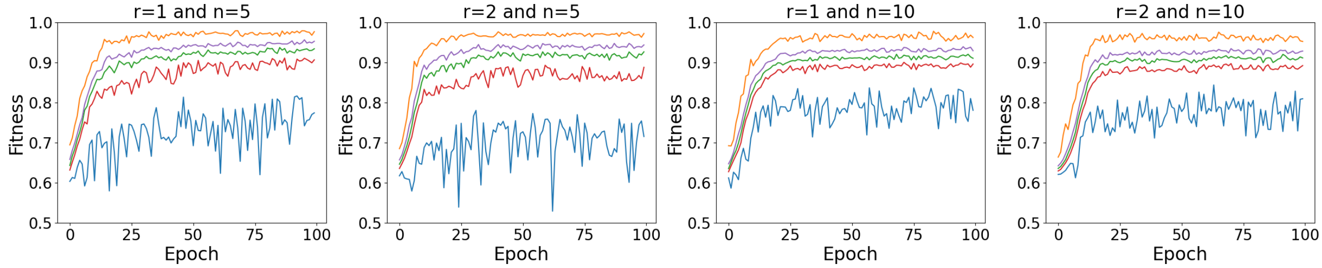

Figure 2: Training Performance with $\alpha = 1.25$ where the lowest fitness of each epoch is represented in blue, the 25th percentile in red, the median in green, the 75th percentile in purple and the highest fitness in orange

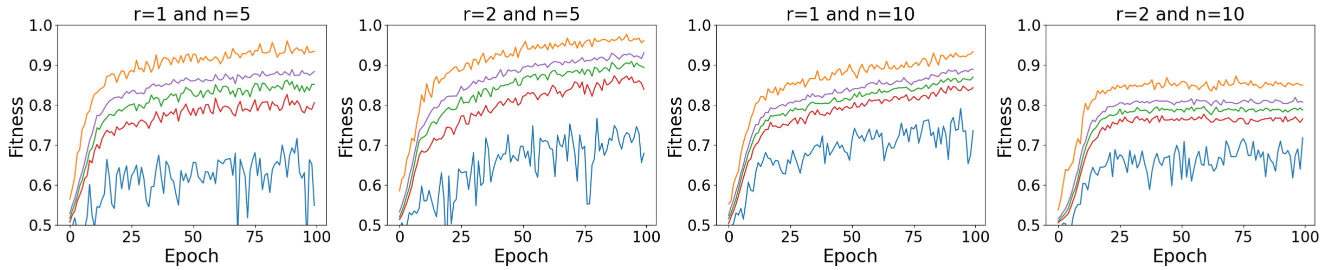

Figure 3: Training Performance with $\alpha = 1.50$ where the lowest fitness of each epoch is represented in blue, the 25th percentile in red, the median in green, the 75th percentile in purple and the highest fitness in orange

4 Discussion

While the GA does seem to struggle particularly with the configuration $\alpha = 1.5$, $r = 2$ and $n = 10$, overall the DFA is clearly a suitable fitness criterion robust to a range of different elementary CA configurations. More importantly, as demonstrated in 4, 5 and 6 its use as a fitness criterion results in CAs producing dizzyingly varied, but still clearly structured outputs. Their suitability as music is of course qualitative and would in no small part be a result of further artistic interpretation on the part of a musician, however we believe their shear variety of structure shows the overall usefulness of the DFA as a fitness criterion as opposed to the probability distribution approach. Further we believe it to also show its increased alignment with the history of algorithmic practice.

In spite of this there are still several clear areas for improvement and further exploration. The most natural extensions of the present work from a technical perspective would be to move to higher dimensional CA or to utilize a Multifractal Detrended Fluctuation Analysis (MFDFA) in place of the DFA currently utilized. The former extension would allow for a greater variety of dynamics and expressivity from the CA output. In particular, dependent on the approach to cell neighborhood, the distinct but related dynamics of different dimensions ought to also allow for far more interesting and nuanced approaches to mapping from CA output to sound. A challenge in this process though will be the encoding step from the multi-dimensional CA output x to the input vector x' for the DFA, where steps will need to be taken to ensure the integrity of the separation between dimensions in the encoding process or differing emphasis between CA dimensions. The use of a non-linear kernel may be an effective approach.

The MFDFA, is the generalization of the DFA across statistical moments. Put another way, the DFA is simply the MFDFA applied to second moment fluctuations. By using an MFDFA across moments one is able to measure a spectrum of α values thus producing a more granular characterization of the fluctuation dynamics of the given time series (for a more complete account of the MFDFA see [Ihlen, 2002]). Additionally, as has been shown in [Telesca and Lovallo, 2011], musical genres have been shown to have distinct multi-fractal

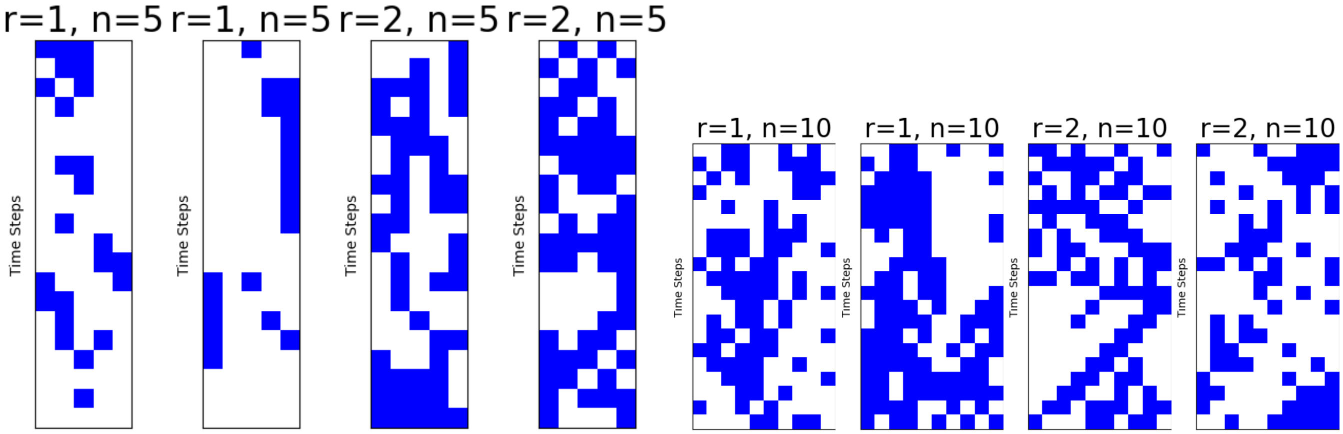

Figure 4: Assorted Trained CA Outputs with $\alpha = 1$

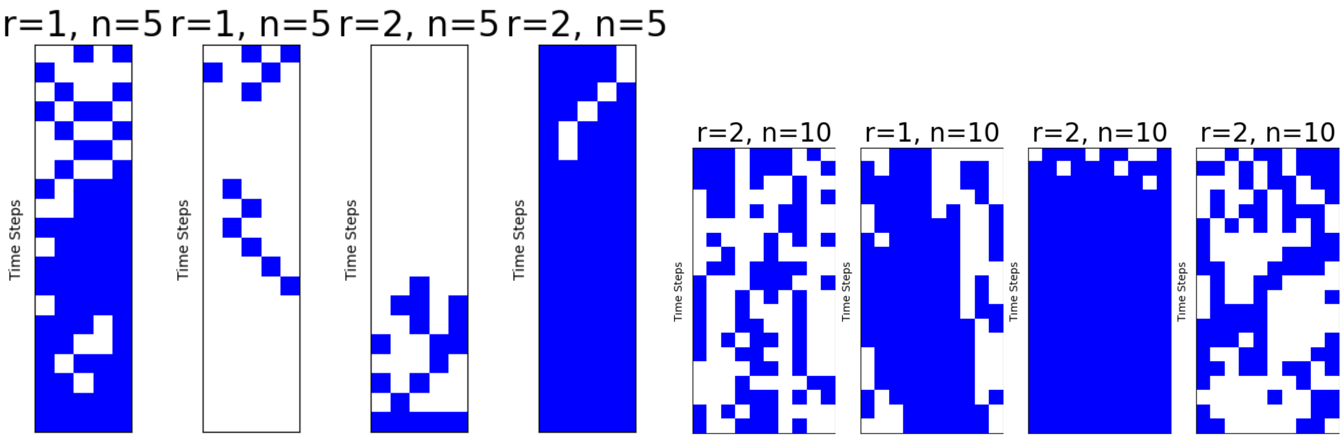

Figure 5: Assorted Trained CA Outputs with $\alpha = 1.25$

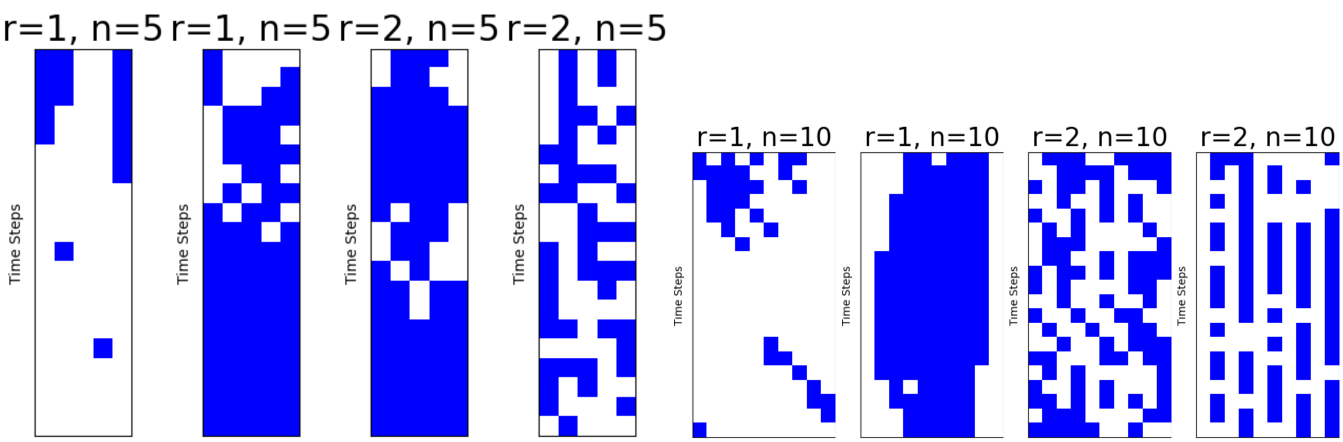

Figure 6: Assorted Trained CA Outputs with $\alpha = 1.50$

spectra. Utilization of the MFDFA in place of the DFA in the GA fitness function would allow for a higher degree of control in scultping the dynamics of the trained CAs. Additionally it could allow for a greater resemblance between the CA output and a given musical genre or the intentional divergence of the CA output from a given genre depending on the practitioner's preference. However, issues may arise for the complexity of the MFDFA as a fitness criterion and it may prove too difficult for the GA to find a suitable solution using the 1-D CA characterized here. This could potentially be resolved using higher dimensional CA, a more robust GA or more elaborate formulations of the 1-D CA.

References

[Beyls, 1989] Peter Beyls. The musical universe of cellular automata. In *Proceedings of International Computer Music Conference*, 1989.

[Delarosa and Soros, 2020] Omar Delarosa and Lisa Soros. Growing midi music files using convolutional cellular automata. In *2020 IEEE Symposium Series on Computational Intelligence*, 2020.

[Gardner, 1970] Martin Gardner. Mathematical games: The fantastic combinations of john conway's new solitaire game 'life'. *Scientific American*, 1970.

[Ihlen, 2002] Espen Ihlen. Introduction to multifractal detrended fluctuation analysis in matlab. *Frontiers in Pyschology*, 2002.

[Jennings *et al.*, 2003] Heather Jennings, Plamen Ivanov, and Allan Martins. Variance fluctuations in nonstationary time series: a comparative study of music genres. *Condensed Matter*, 2003.

[Miranda, 1993] Eduardo Reck Miranda. Cellular automata music: An interdisciplinary project. *Journal of New Music Research*, 1993.

[Peng *et al.*, 1995] C K Peng, S Havlin, and A Goldberger. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos*, 1995.

[Streich and Herrera, 2005] Sebastian Streich and Perfecto Herrera. Detrended fluctuation analysis of music signals danceability estimation and further semantic characterization. In *AES 118*, 2005.

[Telesca and Lovallo, 2011] Luciano Telesca and Michele Lovallo. Revealing competitive behaviours in music by means of the multifractal detrended fluctuation analysis: application to bach's sinfonias. In *Proc. R. Soc. A.*, 2011.

[Wang *et al.*, 2024] Lei Wang, Ziyi Zhao, Hanwei Liu, Junwei Pang, Yi Qin, and Qidi Wu. A review of intelligent music generation systems. *Neural Computing and Applications*, 2024.