Hilbert Curve Based Molecular Sequence Analysis

A Novel Image-Based Deep Learning Approach

Sarwan Ali[1*], Tamkanat E Ali[3*],
Imdad Ullah Khan[3], Murray Patterson[2]

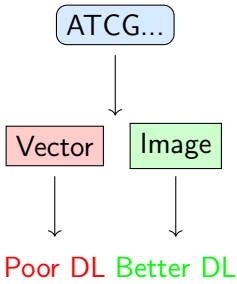
[1]Columbia University, Irving Medical Center, NY, USA
[2]Georgia State University, Atlanta, GA, USA
[3]Lahore University of Management Sciences, Pakistan

Outline

The Challenge in Molecular Sequence Analysis

Key Problems:

- Traditional vector-based embeddings show suboptimal performance with Deep Learning models
- Neural networks struggle with tabular data due to:
  - Feature sparsity
  - Varying scales
  - Lack of spatial correlations
- Existing image-based methods fail to capture spatial information
- Current Hilbert curve methods are sequence-type specific



Our Solution

Universal Hilbert Curve-based Chaos Game Representation (CGR) with novel Alphabetic Index Mapping

Research Contributions

1. **Novel Universal Method**: First universal Hilbert curve-based CGR approach applicable to any molecular sequence type
2. **Alphabetic Index Mapping**: Innovative technique for constructing Hilbert curve-based image representations
3. **Superior Performance**: Achieves **94.5% accuracy** and **93.9% F1-score** on lung cancer dataset
4. **Broad Applicability**: Method extends beyond molecular sequences to NLP domain
5. **Comprehensive Evaluation**: Rigorous testing on multiple datasets with various deep learning architectures

Sequence Representation Methods

Vector-Based Methods:

- Feature Engineering (PWM2Vec, AAC, PAAC)
- NLP-based (SeqVec, PRoBERTa, ESM2)
- Neural Networks (Autoencoder)
- Kernel-based methods

Limitations:

- Poor performance with CNN/DL models
- Loss of spatial information
- Feature sparsity issues

Image-Based Methods:

- Chaos Game Representation (CGR)
- Frequency CGR (FCGR)
- Spike2CGR
- Random CGR

Advantages:

- Better DL performance
- Preserve 2D spatial relationships
- Suitable for CNN architectures

Gap in Literature

Existing Hilbert curve methods lack universality - our approach provides the first universal solution

Hilbert Curve Properties

Why Hilbert Curves?

- Space-filling curve: Maps 1D sequences to 2D plane
- Spatial locality preservation: Close points in 1D remain close in 2D
- Superior to alternatives: Better than Z-order, Peano curves
- Self-similarity: Fractal properties maintain structure at multiple scales

Mathematical Foundation:

$$\Theta = 2^{p \times N}$$

where p = iterations, N = dimensions
For 64×64 images: $p = 6, N = 2$

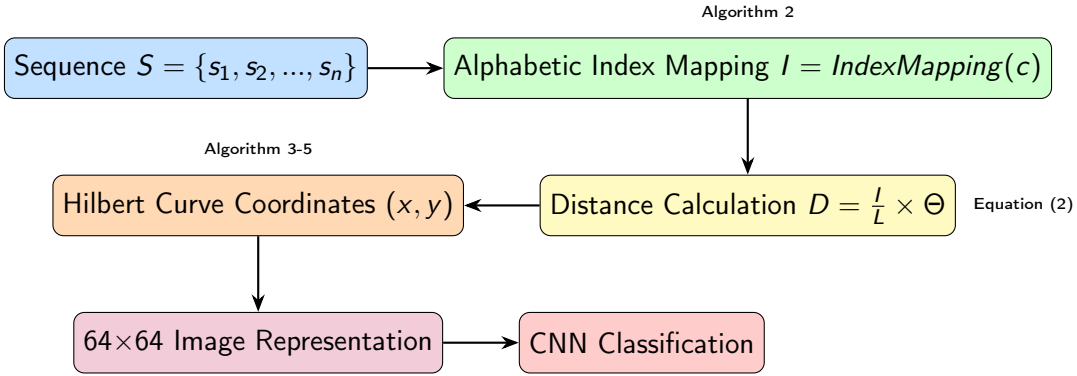
$$\Theta = 2^{6 \times 2} = 2^{12} = 4096 \text{ points}$$

Order 1

Order 2

Self-Similar Space-Filling

Method Overview



Key Innovation

Universal alphabetic index mapping enables application to any molecular sequence alphabet (DNA: {A,T,G,C}, Protein: 20 amino acids, etc.)

Alphabetic Index Mapping

Algorithm 1 Alphabetic Index Mapping

```
function IndexMapping(c, A)
    for a in A do
        if a = c then
            I = A.index(c)
        end if
    end for
    return I
end function
```

Mathematical Formulation:

$$I = \mathit{IndexMapping}(c), \quad c \in A$$

where A is the alphabet set and c is a character.

Examples:

DNA Alphabet: $A = \{A, T, G, C\}$

- A $\rightarrow$ Index = 0, T $\rightarrow$ Index = 1
- G $\rightarrow$ Index = 2, C $\rightarrow$ Index = 3

Protein Alphabet: $A = \{A, R, N, D, ...\}$ (20 amino acids)

- A $\rightarrow$ Index = 0
- R $\rightarrow$ Index = 1
- ... and so on

Universal Property

Works with any alphabet size and composition!

Distance Calculation & Coordinate Mapping

Distance Calculation:

$$D = \frac{I}{L} \times \Theta$$

- I = Alphabetic index of character
- L = Length of sequence
- $\Theta = 2^{p \times N}$ = Total points on Hilbert curve

Coordinate Transformation Process:

1. **Binary Representation:** $Bits = Binary(D) = b_{n-1}b_{n-2}...b_0$
2. **Bit Interleaving:**

$$EvenIdxBits = b_{n-1}b_{n-3}... \quad (1)$$

$$OddIdxBits = b_{n-2}b_{n-4}... \quad (2)$$

3. **Preliminary Coordinates:**

$$x_{raw} = Decimal(EvenIdxBits) \quad (3)$$

$$y_{raw} = Decimal(OddIdxBits) \quad (4)$$

Gray Code Transformation

Purpose: Minimize bit changes between successive values to preserve spatial locality
Gray Code Equations:

$$x_{gray} = x_{raw} \oplus (x_{raw} \gg 1) \tag{5}$$

$$y_{gray} = y_{raw} \oplus (y_{raw} \gg 1) \tag{6}$$

where $\oplus$ is XOR operation and $\gg$ is right shift.
Coordinate Refinement (Inverse Gray Code):

$$x = InverseGrayCode(x_{gray}) \tag{7}$$

$$y = InverseGrayCode(y_{gray}) \tag{8}$$

Iterative Inverse Process:

$$x = x_{gray}, \quad \text{For } i = n-2 \text{ to } 0 : x_i = x_i \oplus x_{i+1}$$

Spatial Locality Preservation

Gray code transformation ensures that points close in 1D sequence remain close in 2D Hilbert curve representation

Core Algorithm: Distance to Hilbert Point

Algorithm 2 Distance to Hilbert Curve Point

```
1: function PointFromDistance(D, p)
2:     NumBits = N × p                                  ▷ Number of bits
3:     Bits = Binary(D)                                 ▷ Distance to binary string
4:     OddIdxBits ← [], EvenIdxBits ← []
5:     for i in range(0, LenBits) do
6:         if i mod 2 = 0 then
7:             EvenIdxBits.append(Bits[i])
8:         else
9:             OddIdxBits.append(Bits[i])
10:        end if
11:    end for
12:    EvenIdxDeci = Decimal(EvenIdxBits)
13:    OddIdxDeci = Decimal(OddIdxBits)
14:    Components = [EvenIdxDeci, OddIdxDeci]
15:    GrayCode ← GenerateGrayCode(Components, n)
16:    φ ← Refining(GrayCode, p)
17:    return φ                                         ▷ Final (x,y) coordinates
18: end function
```

Datasets & Experimental Design

Datasets Used:

- **Breast Cancer ACPs:** 949 sequences
- **Lung Cancer ACPs:** 901 sequences
- **Classes:** 4-class classification
  - Very active
  - Moderately active
  - Experimentally inactive
  - Virtually inactive
- **Sequence Length:** 5-38 amino acids
- **Average Length:** 14.5-20.7 amino acids

Baseline Methods:

Vector-based:

- One Hot Encoding (OHE)
- Spike2Vec, PWM2Vec
- Auto-Encoder, WDGRL
- SeqVec (pre-trained LLM)

Image-based:

- FCGR, Spike2CGR, Random CGR

DL Architectures:

- 1, 2, 3-layer CNNs, VGG19, ResNet50
- EfficientNet, DenseNet

Training Setup: 80% train, 20% test, 10% validation | Batch size: 64, Epochs: 10, LR: 0.003, Optimizer: ADAM

Evaluation Metrics

Performance Metrics:

- **Accuracy:** Overall classification accuracy
- **Precision:** True positives / (True positives + False positives)
- **Recall:** True positives / (True positives + False negatives)
- **F1-Score:** Harmonic mean of precision and recall
  - Weighted F1: Accounts for class imbalance
  - Macro F1: Unweighted average across classes
- **ROC-AUC:** Area Under Receiver Operating Characteristic curve
- **Training Runtime:** Computational efficiency measure

Experimental Environment:

- Intel i5 processor (2.40 GHz), 32 GB RAM
- Windows 10, Python implementation
- Standard cross-validation for hyperparameter tuning

Breast Cancer Dataset Results

Method	Model	Acc.	Prec.	Recall	F1-W	F1-M	ROC-AUC
Vector-Based Methods							
OHE	-	0.609	0.853	0.609	0.676	0.395	0.678
Auto-Encoder	-	0.832	0.802	0.832	0.804	0.431	0.645
SeqVec	-	0.674	0.819	0.674	0.725	0.389	0.651
Image-Based Methods							
FCGR	1-Layer CNN	0.863	0.831	0.863	0.844	0.490	0.677
FCGR	3-Layer CNN	0.800	0.640	0.800	0.711	0.222	0.500
Spike2CGR	1-Layer CNN	0.783	0.613	0.783	0.687	0.219	0.500
Our Method							
Ours	**1-Layer CNN**	**0.895**	**0.869**	**0.895**	**0.881**	**0.521**	**0.725**
Ours	4-Layer CNN	0.874	0.861	0.874	0.867	0.476	0.705
Ours	ResNet50	0.853	0.837	0.853	0.841	0.465	0.690

Key Findings

- Our method achieves **89.5% accuracy** vs. best baseline of 86.3%
- **Simple 1-layer CNN performs best** - supports Occam's razor principle
- Significant improvement in F1-Macro score: **0.521** vs. 0.490

Lung Cancer Dataset Results

Method	Model	Acc.	Prec.	Recall	F1-W	F1-M	ROC-AUC
Vector-Based Methods							
Auto-Encoder	-	0.910	0.908	0.910	0.906	0.602	0.771
SeqVec	-	0.886	0.882	0.886	0.878	0.604	0.761
Spaced k-mer	-	0.883	0.871	0.883	0.862	0.530	0.699
Image-Based Methods							
FCGR	3-Layer CNN	0.930	0.925	0.930	0.929	0.681	0.810
FCGR	VGG19	0.921	0.919	0.921	0.918	0.600	0.776
RandomCGR	VGG19	0.892	0.714	0.892	0.769	0.297	0.524
Our Method							
Ours	**1-Layer CNN**	**0.945**	**0.938**	**0.945**	**0.939**	0.664	0.791
Ours	VGG19	0.917	0.888	0.917	0.898	0.490	0.683
Ours	4-Layer CNN	0.912	0.909	0.912	0.909	0.534	0.729

Outstanding Performance

- **94.5% accuracy** - highest among all methods
- **93.9% weighted F1-score** - exceptional classification performance
- Outperforms sophisticated baselines like FCGR + 3-layer CNN (93.0%)

Performance Analysis & Insights

Key Observations:

1. **Consistent Superiority**: Our method outperforms all baselines on both datasets
2. **Simple is Better**: 1-layer CNN achieves best results
   - Supports Occam's razor principle
   - Optimal balance of simplicity vs. learning capability
3. **Deep Networks Struggle**: EfficientNet shows poor performance (6-8% accuracy)
4. **Computational Efficiency**: Our method has reasonable training times

Why Our Method Works:

- **Spatial Locality**: Hilbert curve preserves neighborhood relationships
- **Universal Mapping**: Works with any alphabet size/type
- **Information Preservation**: Bijective relationship prevents data loss
- **Optimal Representation**: 64×64 images provide sufficient resolution

Mathematical Guarantee

Bijective mapping: $I = \mathit{IndexMapping}(c)$ ensures each character maps to unique Hilbert curve point

Computational Performance

Table: Training Runtime Comparison (seconds)

Method	Model	Breast Cancer	Lung Cancer
FCGR	1-Layer CNN	5410.4	5023.0
FCGR	3-Layer CNN	52147.9	41247.7
Spike2CGR	1-Layer CNN	6548.0	5987.1
RandomCGR	1-Layer CNN	4982.9	5024.7
Ours	**1-**	**2136.8**	**1648.9**
Ours	3-Layer CNN	13544.3	16365.4
Ours	VGG19	25081.4	19265.0

Efficiency Advantages

- **2-3x faster** than competing image-based methods
- **Scalable**: Training time scales reasonably with model complexity
- **Practical**: Suitable for real-world deployment

Algorithmic Innovation

Core Algorithms:

1. **Alphabetic Index Mapping**
   - Universal character-to-index conversion
   - Works with any molecular alphabet
2. **Distance Calculation**

$$D = \frac{I}{L} \times \Theta$$

   where $\Theta = 2^{p \times N}$
3. **Gray Code Transformation**
   - Preserves spatial locality
   - Minimizes bit changes between successive values

Key Technical Features:

- **Bijective Mapping**: No information loss
- **Spatial Coherence**: Nearby sequence elements remain close in 2D
- **Fractal Properties**: Self-similarity at multiple scales
- **Fixed Resolution**: 64×64 images for standardization

Mathematical Foundation

Hilbert curve order $p = 6$ generates $2^6 \times 2^6 = 64 \times 64$ pixel images

Comparison with State-of-the-Art

Table: Performance Summary - Best Results

Method	Type	Breast Acc.	Lung Acc.	Avg. F1-W	Improvement
Auto-Encoder	Vector	83.2%	91.0%	85.5%	-
FCGR	Image	86.3%	93.0%	88.7%	-
Spike2CGR	Image	78.3%	83.3%	73.8%	-
RandomCGR	Image	80.0%	89.2%	76.0%	-
Our Method	**Image**	**89.5%**	**94.5%**	**91.0%**	**+2.6%**

Competitive Advantages

- **Consistent Performance**: Best results on both datasets
- **Universal Applicability**: Works with any molecular sequence type
- **Computational Efficiency**: Faster training than competitors
- **Theoretical Foundation**: Strong mathematical guarantees

Broader Impact & Applications

Bioinformatics Applications:

- **Drug Discovery**: Identify therapeutic peptides
- **Disease Diagnosis**: Biomarker classification
- **Protein Function**: Predict biological activity
- **Genomics**: DNA/RNA sequence analysis

Beyond Biology:

- Natural Language Processing
- Time Series Analysis
- Any sequential data with spatial patterns

Research Contributions:

1. **Methodological Innovation**: Novel sequence-to-image transformation
2. **Universal Framework**: Applicable to diverse molecular alphabets
3. **Performance Breakthrough**: State-of-the-art results on benchmark datasets
4. **Computational Efficiency**: Practical for real-world deployment

Future Directions

Integration with transformer architectures and multi-modal learning approaches

Conclusion

Key Achievements

- **94.5% accuracy** on lung cancer peptide classification
- **89.5% accuracy** on breast cancer peptide classification
- **Universal method** applicable to any molecular sequence type
- **Computationally efficient** with 2-3x faster training times

Scientific Impact

- **Bridges computer vision and bioinformatics** through novel sequence representation
- **Theoretical foundation** with bijective mapping guarantees
- **Practical significance** for drug discovery and disease detection
- **Extensible framework** for future research directions

References

- Sagan, H. (1994). Hilbert's space-filling curve. In *Space-filling curves* (pp. 9-30).
- Löchel, H.F., Heider, D. (2021). Chaos game representation and its applications in bioinformatics. *Computational and Structural Biotechnology Journal*, 19, 6263-6271.
- Löchel, H.F., Eger, D., Sperlea, T., Heider, D. (2020). Deep learning on chaos game representation for proteins. *Bioinformatics*, 36(1), 272-279.
- Grisoni, et al. (2019). De novo design of anticancer peptides by ensemble artificial neural networks. *Journal of Molecular Modeling*, 25(5), 112.
- Yin, B., Balvert, M., Zambrano, D., Schönhuth, A., Bohte, S. (2018). An image representation based convolutional network for DNA classification. *arXiv preprint arXiv:1806.04931*.