

A decorative network diagram in the top-left corner of the slide. It features a complex web of interconnected nodes and edges. The nodes are represented by small circles, some of which are solid blue, some are hollow blue, and others are solid grey. The edges are thin grey lines connecting the nodes. The overall structure is a dense, interconnected mesh.

# **dna2vec:** Consistent vector representations of variable-length k-mers

by Patrick Ng

A decorative network diagram in the bottom-right corner of the slide. It features a complex web of interconnected nodes and edges. The nodes are represented by small circles, some of which are solid blue, some are hollow blue, and others are solid grey. The edges are thin grey lines connecting the nodes. The overall structure is a dense, interconnected mesh.

# TABLE OF CONTENTS

**01** Paper Introduction

**02** Training Phases

**03** Challenges

**04** Discussion

# 01 Paper Introduction

The paper introduces a novel method to train distributed representations of **variable-length** k-mers based on **word2vec** model.

The contribution of the work includes:

- ◎ variable-length k-mer embedding model
- ◎ experimental evidence of similarity between arithmetic of dna2vec vectors and nucleotides concatenation
- ◎ relationship between Needleman-Wunsch alignment and cosine similarity of dna2vec vectors
- ◎ construction of nucleotide concatenation analogy using dna2vec arithmetic



## 02 Training Stages

The training of dna2vec using hg38 dataset is done in four stages:

1. separate genome into long non-overlapping DNA fragments
2. convert long DNA fragments into overlapping variable-length k-mers
3. unsupervised training of an aggregate embedding model using a two-layer neural network
4. decompose aggregated model by k-mer lengths

# 03 CHALLENGES

Paper Comprehension



Implementation

# 3.1 Paper Comprehension



## Terminology & Concept [ongoing]

- ⊙ Categorical Variable
- ⊙ One-Hot Vector Encoding
- ⊙ K-Mer Components
- ⊙ Sequence Analysis
- ⊙ Neural Network
- ⊙ Natural Language Processing
- ⊙ Word2Vec
- ⊙ Needleman-Wunsch Similarity Score
- ⊙ Nucleotide Concatenation Analogy
- ⊙ hg38 Dataset
- ⊙ Entropy
- ⊙ Negative Sampling
- ⊙ Hierarchy Softmax
- ⊙ Cosine Similarity of Vectors

# 3.2 Implementation



## Software Requirements [ongoing]

- ⦿ Git, Git Bash
- ⦿ Pip, Python 3.7.4
- ⦿ GCC, GFortran Compilers
- ⦿ Other Requirements (Around 20)



## Python For ML [ongoing]

- ⦿ Youtube
- ⦿ Coursera
- ⦿ Scikit-learn

# 3.2 Implementation



## Git & GitHub [complete]

- Create Account
- Learn Git Essentials
- Clone the Repo



## Training & Analysis [on hold]

- Download the hg38 Dataset
- Train and Analyze Results



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. Some nodes are highlighted with blue circles, and others with blue dots.

# Q & A

A decorative network diagram in the bottom-right corner, featuring a complex web of interconnected nodes and lines. Some nodes are highlighted with blue circles, and others with blue dots.