Decoding the Genomic Language:

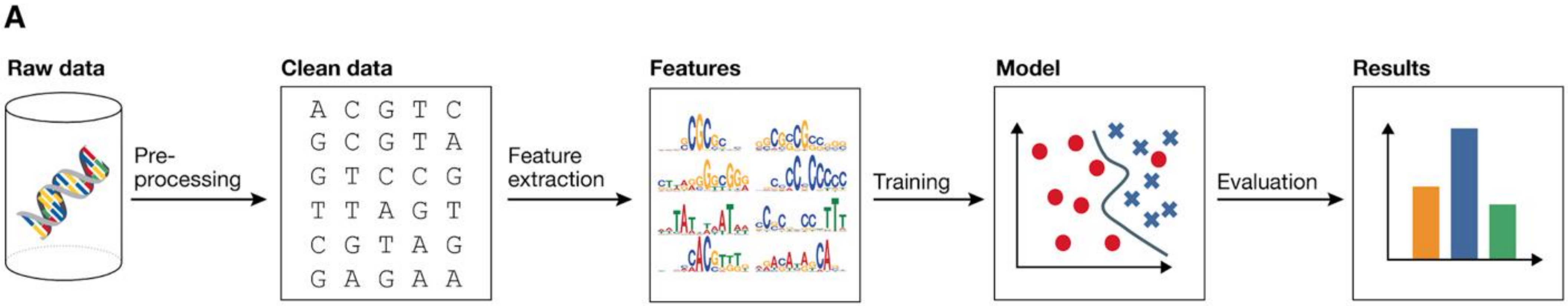
Fine-tuning LLMs for genomic downstream tasks

Fabiana Góes

The Rosalind Franklin Institute

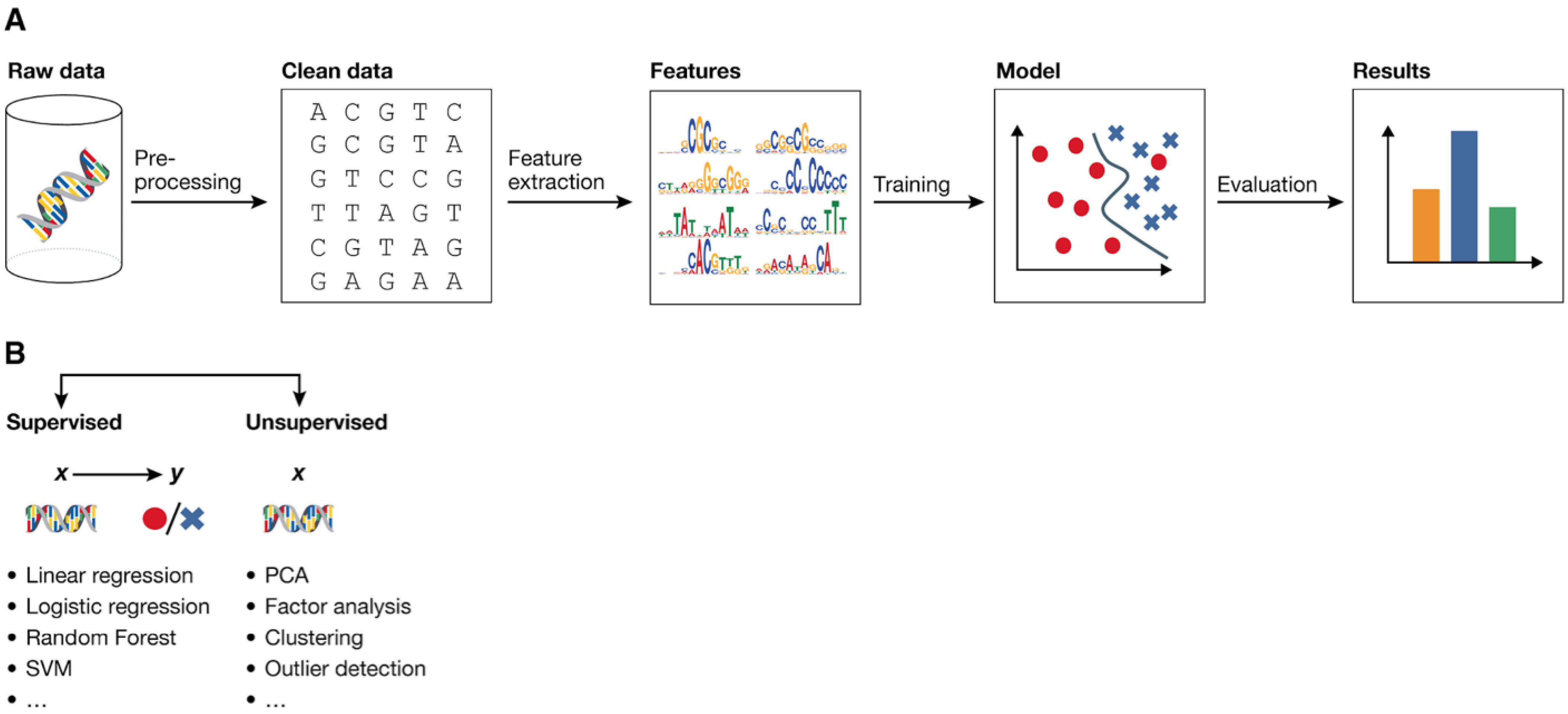
Artificial intelligence Group

Machine Learning for Computational Biology



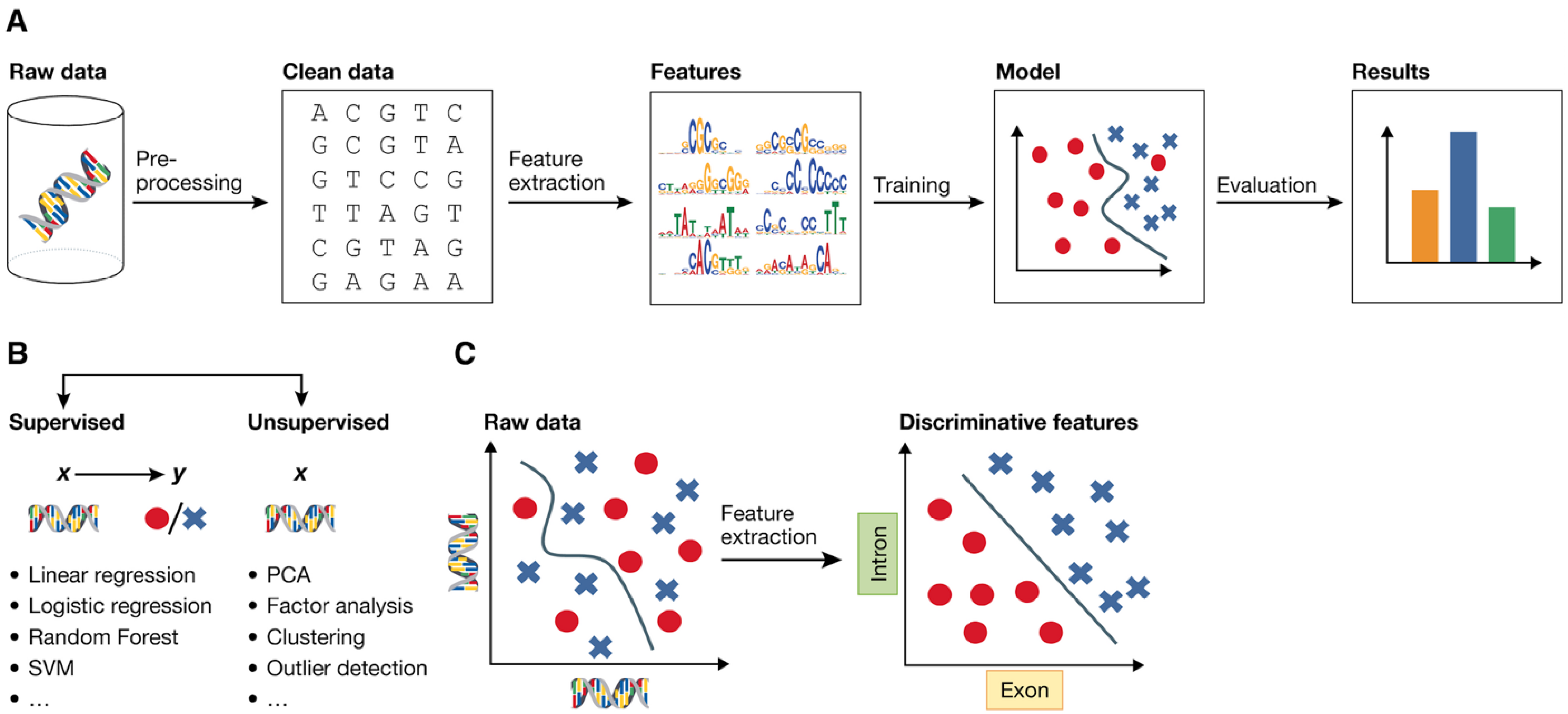
https://doi.org/10.15252/msb.20156651

Machine Learning for Computational Biology



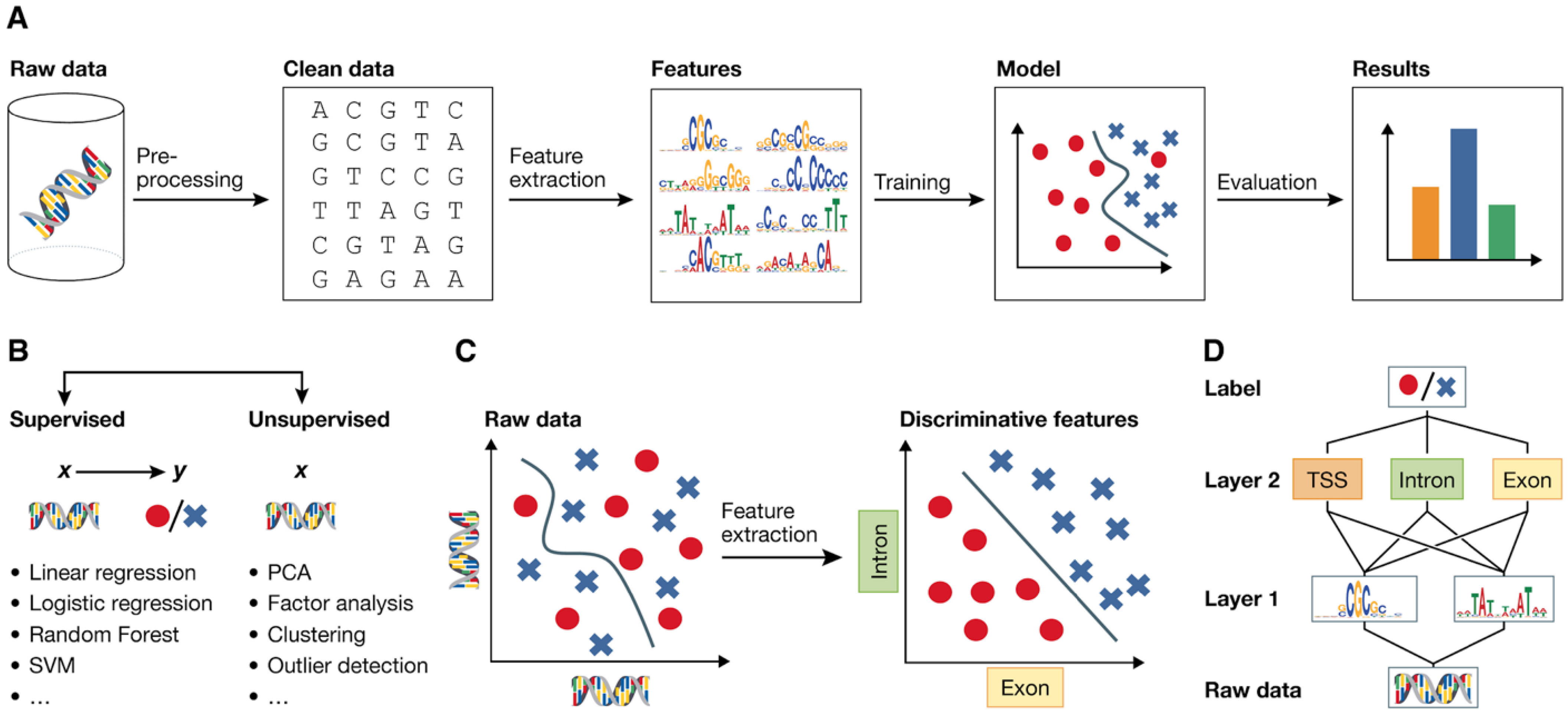
https://doi.org/10.15252/msb.20156651

Machine Learning for Computational Biology



https://doi.org/10.15252/msb.20156651

Machine Learning for Computational Biology



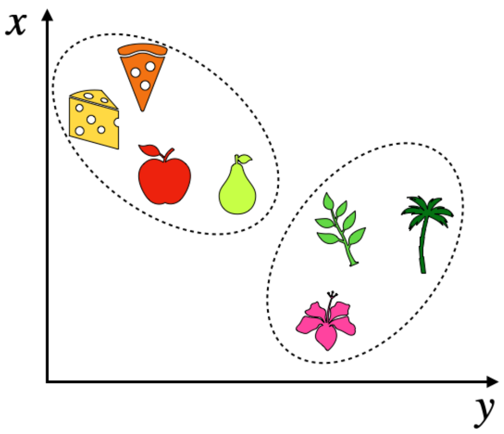
https://doi.org/10.15252/msb.20156651

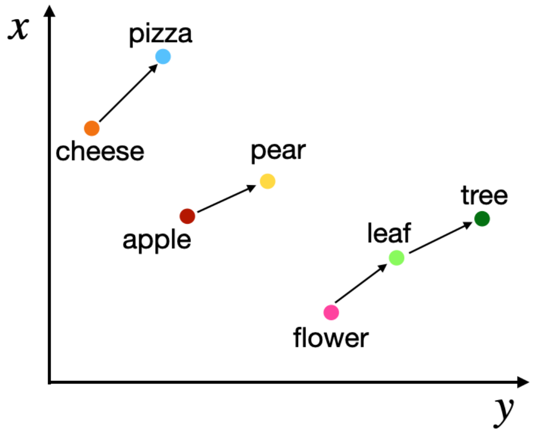
Natural Language Processing - Representation Learning

- **Representation learning**
  - Models to extract representation from raw data
  - Convert words into vectors, preserving their semantic similarity
  - The performance of ML depends on the quality of the representation
  - Neural networks specialized for **understanding** the **relationships** among **words**

- Embedding models:
  - Transform sequences into numerical vectors capturing patterns
  - Assumption: similar words should have similar vector
  - High vector similarity between proteins -> similar functions and structures

- Model types:
  - **Word2Vec**, Doc2Vec
  - CNNs, RNNs
  - **Transformer-based architectures (LLMs)**





NLP - Evolution of Word Representation Learning

Traditional methods
~1990 - 2010

- Bag of words
- TF-IDF
- One-hot encoding

No semantic

Word2Vec & GloVe
~2013 - 2014

- LSTM, bi-LSTM
- Context consideration
- Attention mechanism

Semantic features

FastText
~2016

- Subword information
- OOV handling

Unseen words

RNN-based methods
~2014 - 2017

- LSTM, bi-LSTM
- Context consideration
- Attention mechanism

Sequential context

ELMo
~2018

- Multiple bi-LSTM
- Stacked layers
- Deep contextualization

Deep context

BERT & Transformers
~2017 - present

- Bidirecional encoder
- Self-attention
- Transfer learning

Bidirecional encoder

Static vs Contextual Embeddings

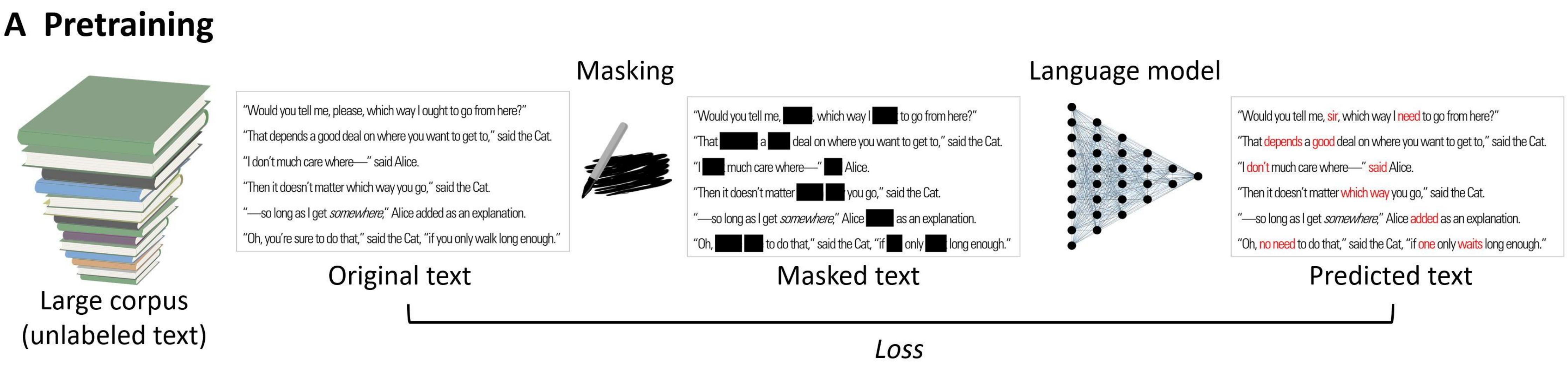
Static Embeddings
(Word2Vec, GloVe, FastText)

- One embedding per word, regardless of context
- Each word always has the same vector
- Cannot handle polysemy/homonymy

Contextual Embeddings
(ELMo, BERT, Transformers)

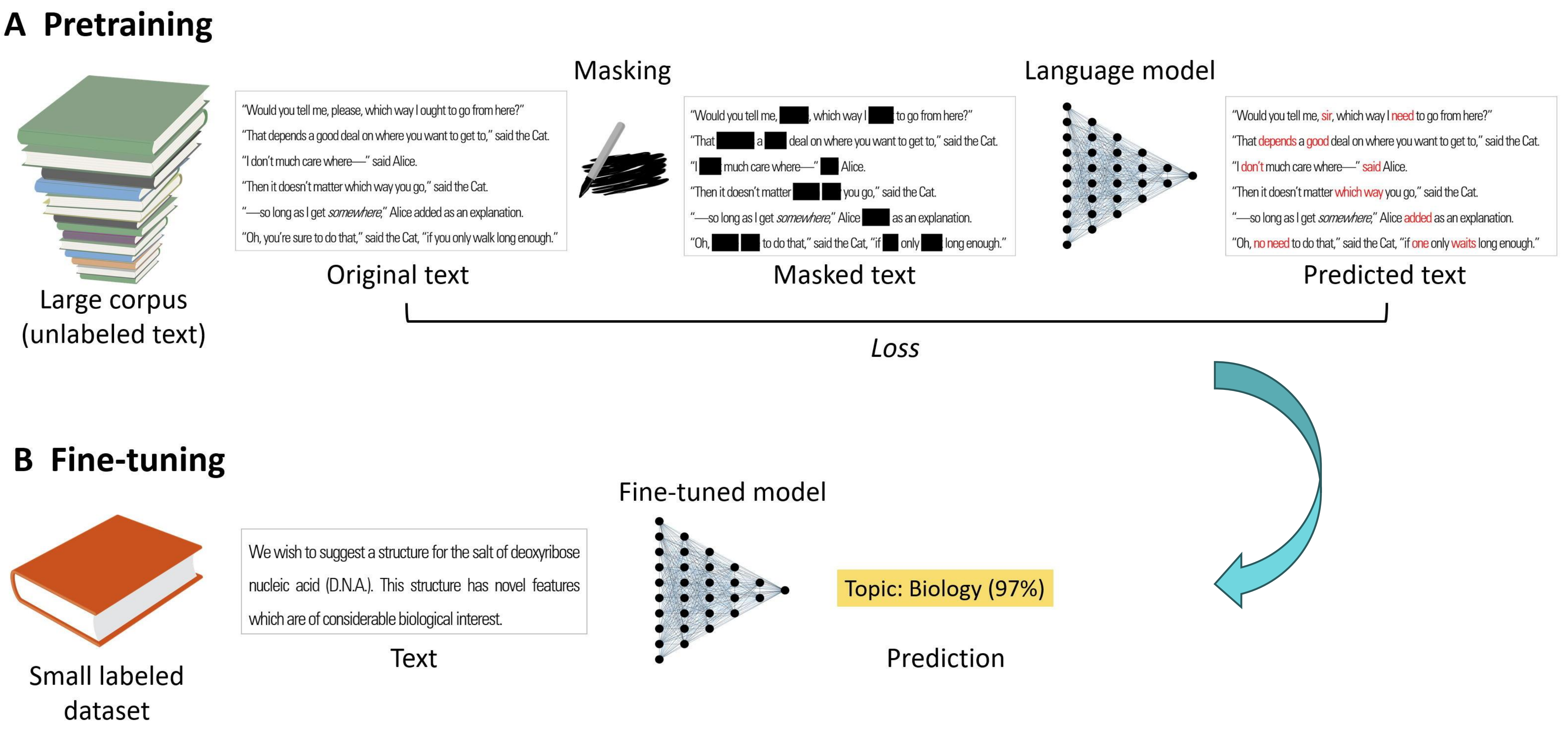
- Embeddings based on context
- "right to vote" ≠ "turn right" (different vectors)
- Handles multiple word meanings

NLP – Deep language models



https://doi.org/10.1016/j.csbj.2021.03.022

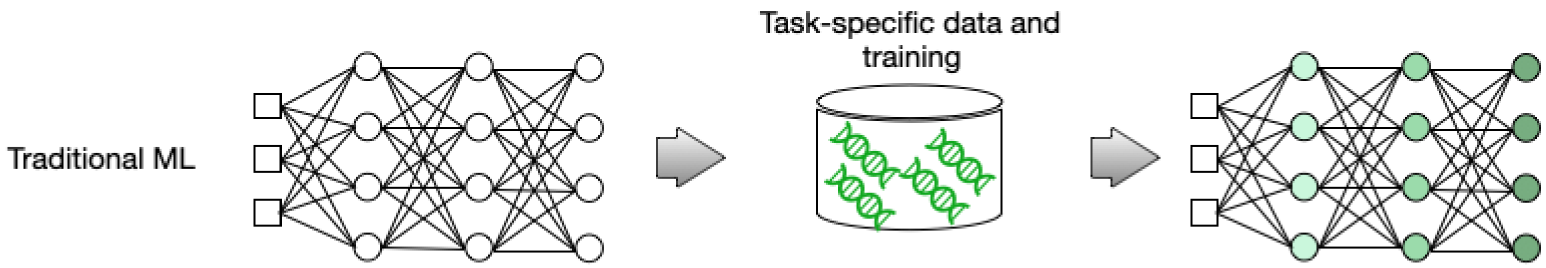
NLP – Deep language models



https://doi.org/10.1016/j.csbj.2021.03.022

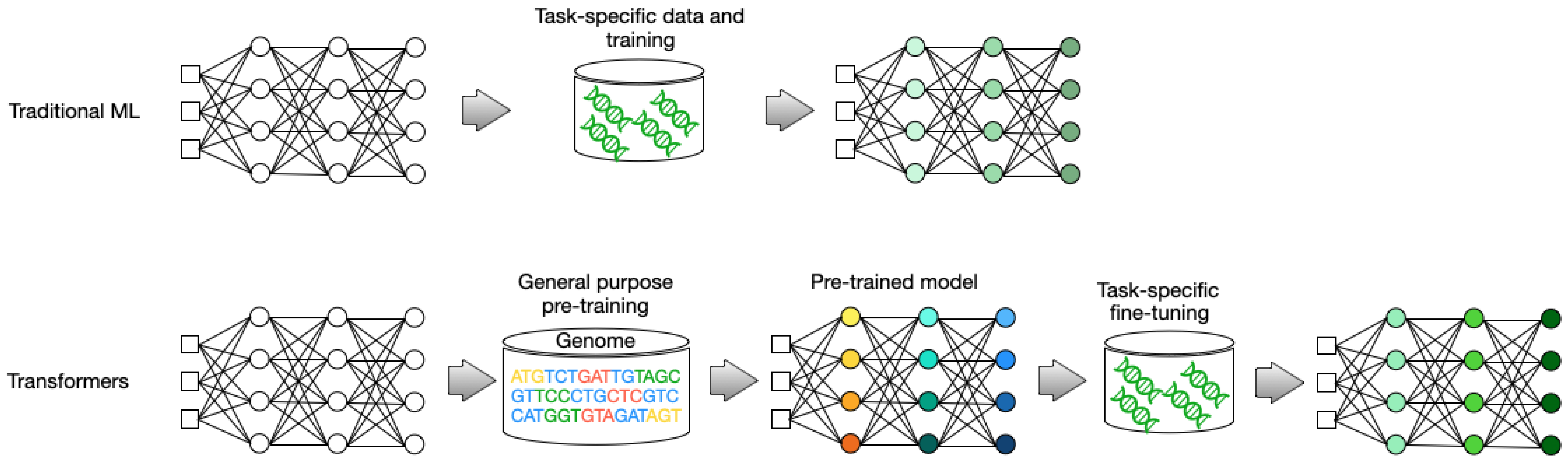
Large Language Models in Genomics

- Learning from vast amounts of unlabeled data
- Capture deep biological patterns, long-range dependencies, and structural information
- Strong capacities to map inputs into latent embedding space
- Can be adapted to a wide range of tasks

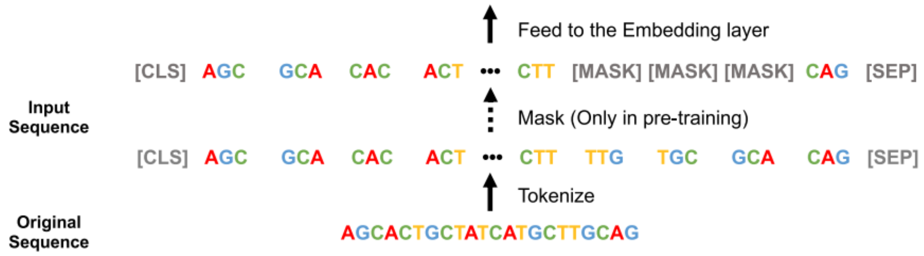


Large Language Models in Genomics

- Learning from vast amounts of unlabeled data
- Capture deep biological patterns, long-range dependencies, and structural information
- Strong capacities to map inputs into latent embedding space
- Can be adapted to a wide range of tasks

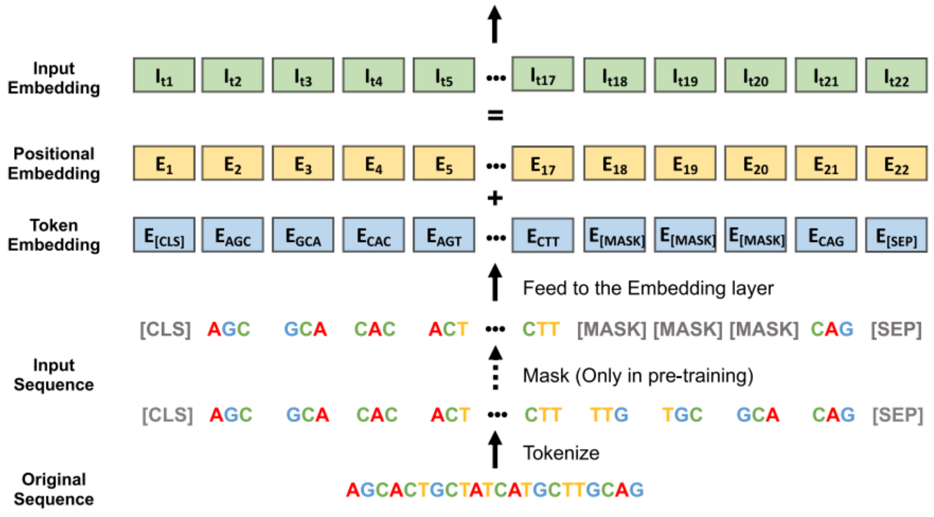


DNABERT – BERT model to DNA



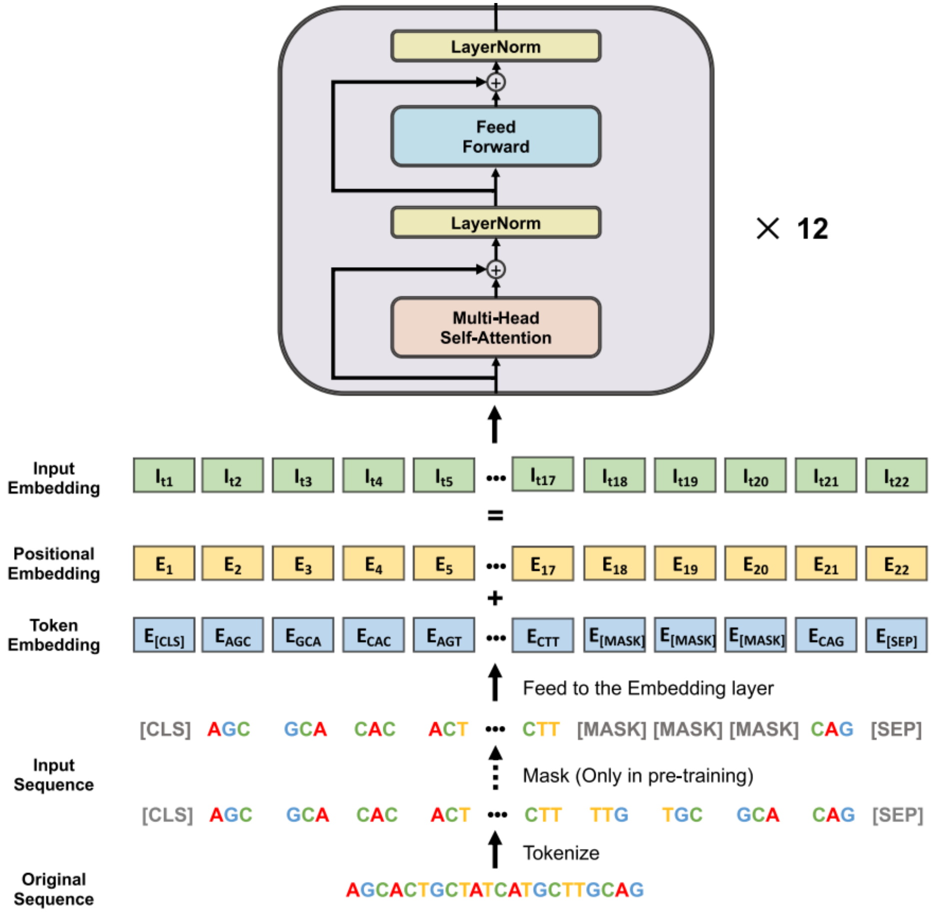
https://doi.org/10.1093/bioinformatics/btab083

DNABERT – BERT model to DNA



https://doi.org/10.1093/bioinformatics/btab083

DNABERT – BERT model to DNA

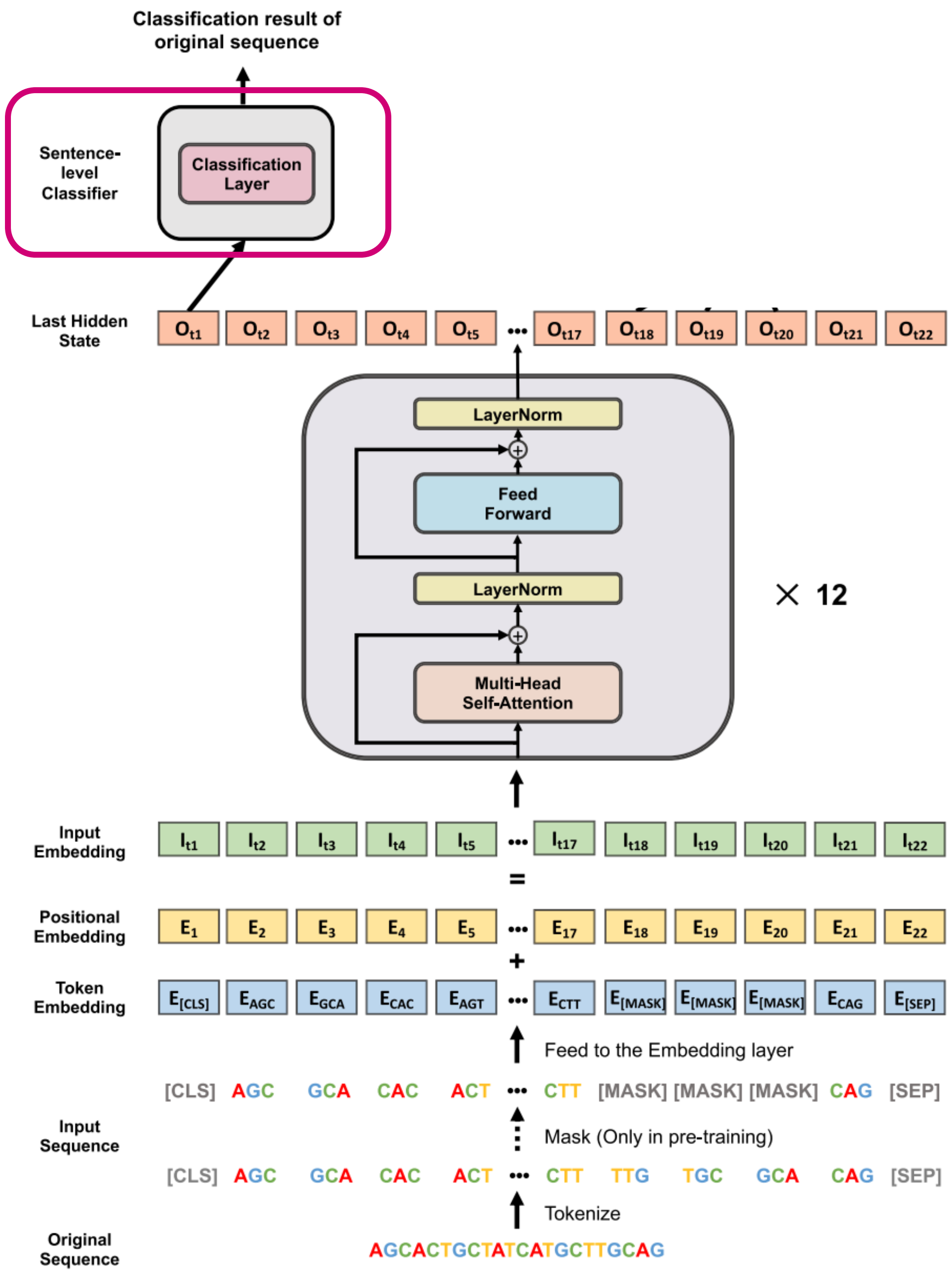


www.rfi.ac.uk

The Rosalind Franklin Institute

https://doi.org/10.1093/bioinformatics/btab083

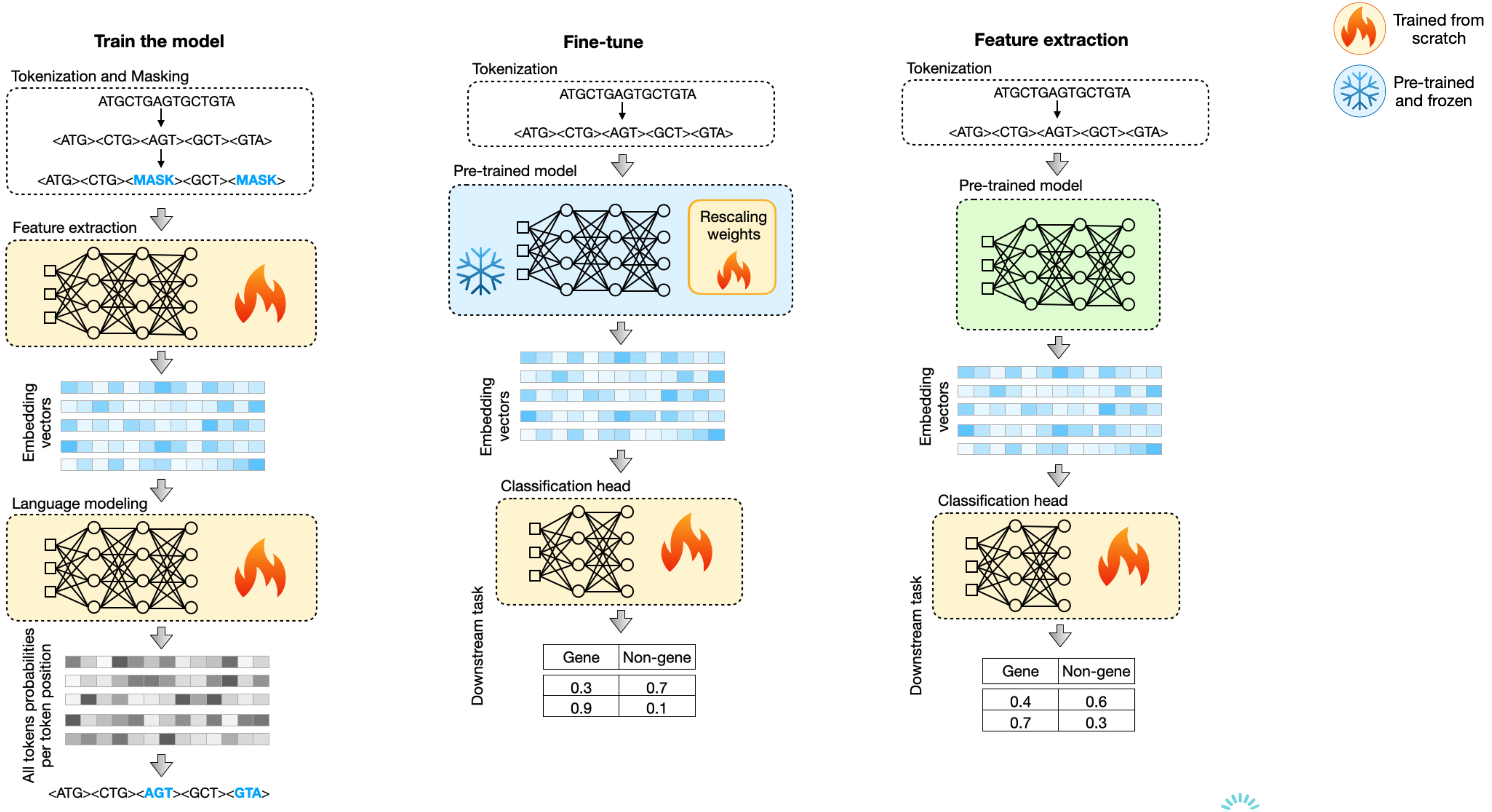
DNABERT – BERT model to DNA



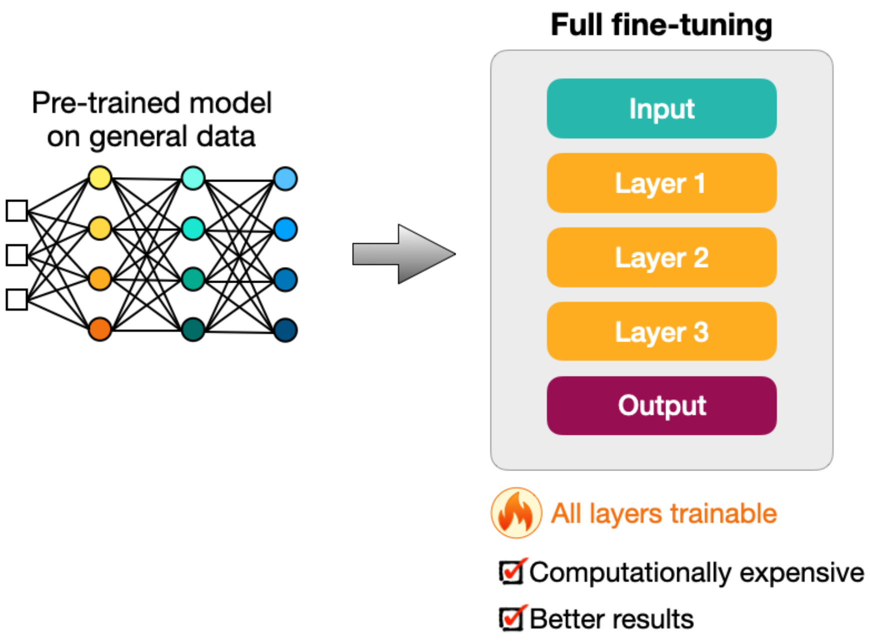
The Rosalind Franklin Institute

https://doi.org/10.1093/bioinformatics/btab083

LLMs - Three Approaches

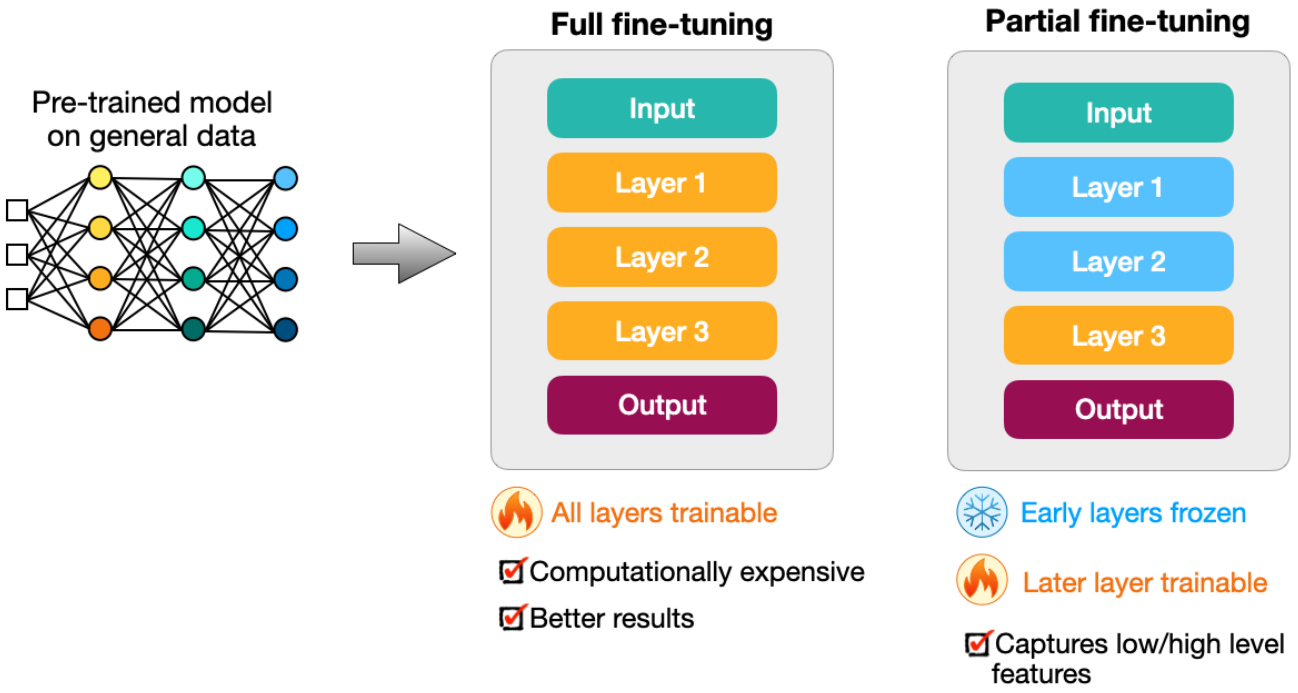


Fine-tuning



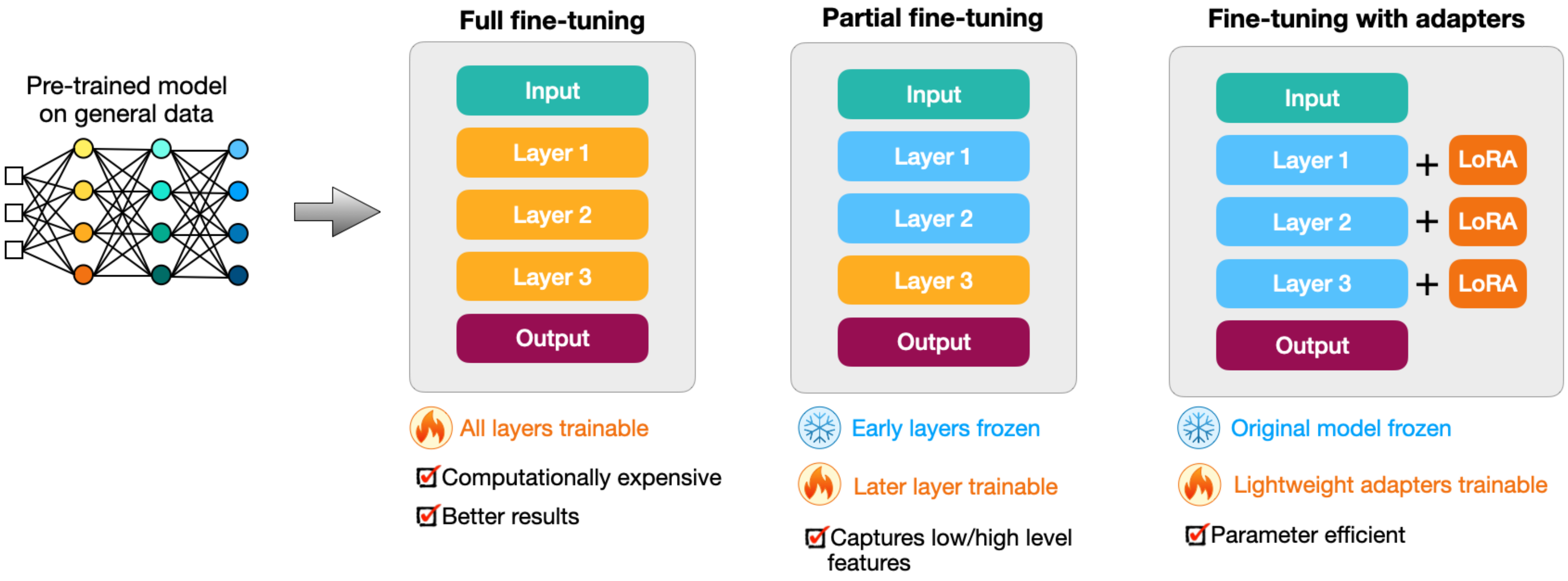
https://doi.org/10.1093/bioinformatics/btab083

Fine-tuning



https://doi.org/10.1093/bioinformatics/btab083

Fine-tuning



LoRA (Low-Rank Adaptation):

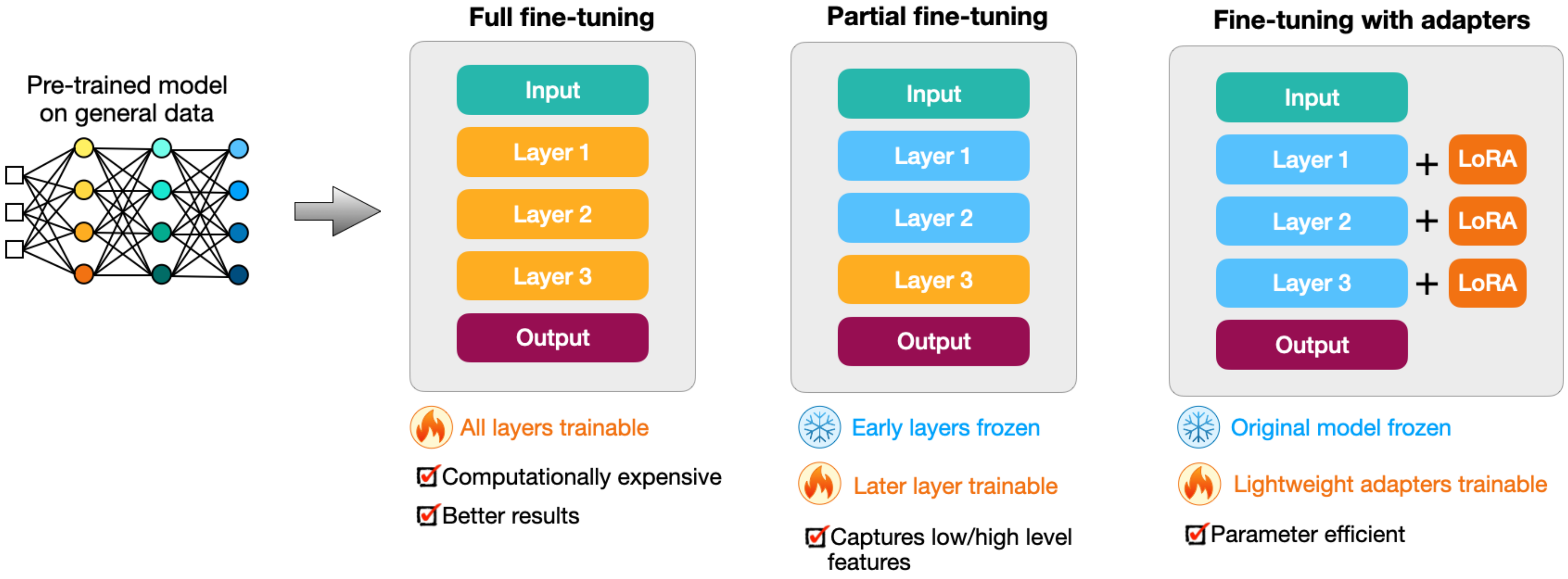
- Decomposes weight updates into low-rank matrices (A × B)

Trade-off: Full fine-tuning vs. Adapters

- Better results vs. Efficiency

https://doi.org/10.1093/bioinformatics/btab083

Fine-tuning



Key Concepts:

Frozen Layers:

- Parameters unchanged
- Preserve learned features
- Less computation

Hierarchical Features:

- Early: low-level features
- Later: high-level features
- Task-specific

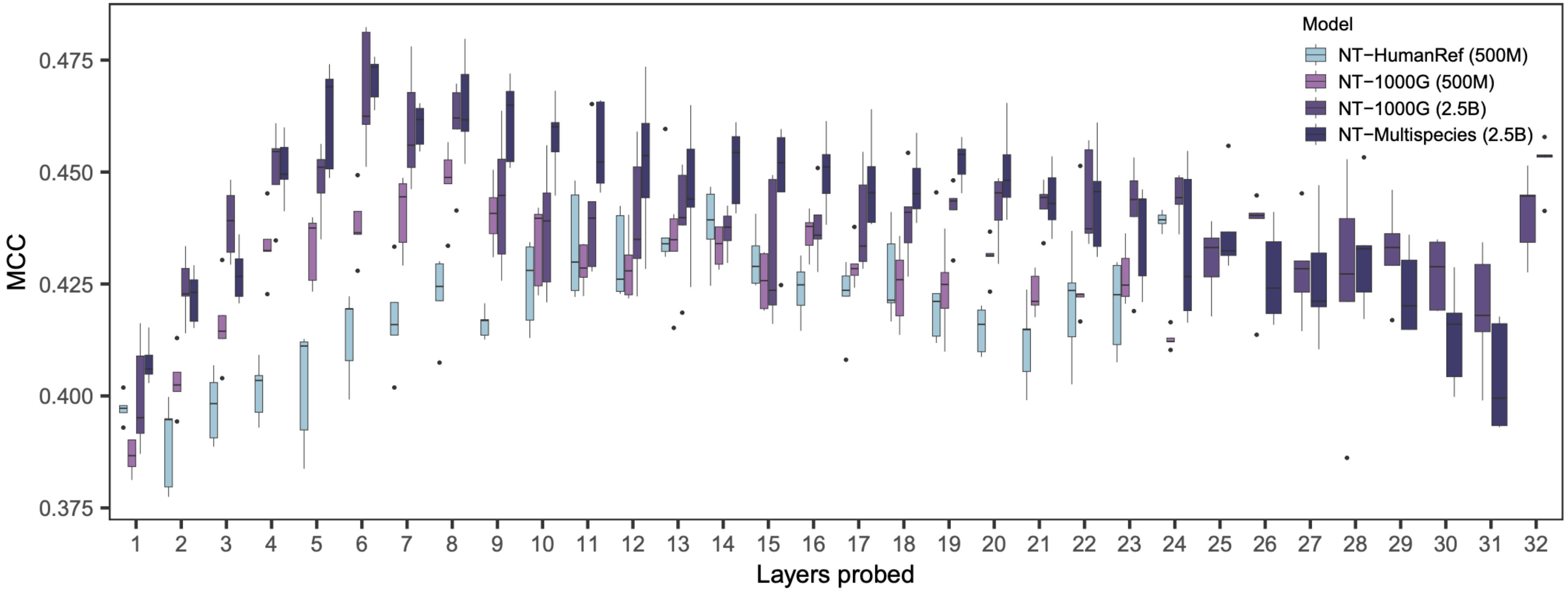
Adapters (LoRA, etc.)

- Lightweight modules
- Few parameters
- Parameter efficient

https://doi.org/10.1093/bioinformatics/btab083

Partial fine-tuning - Layers

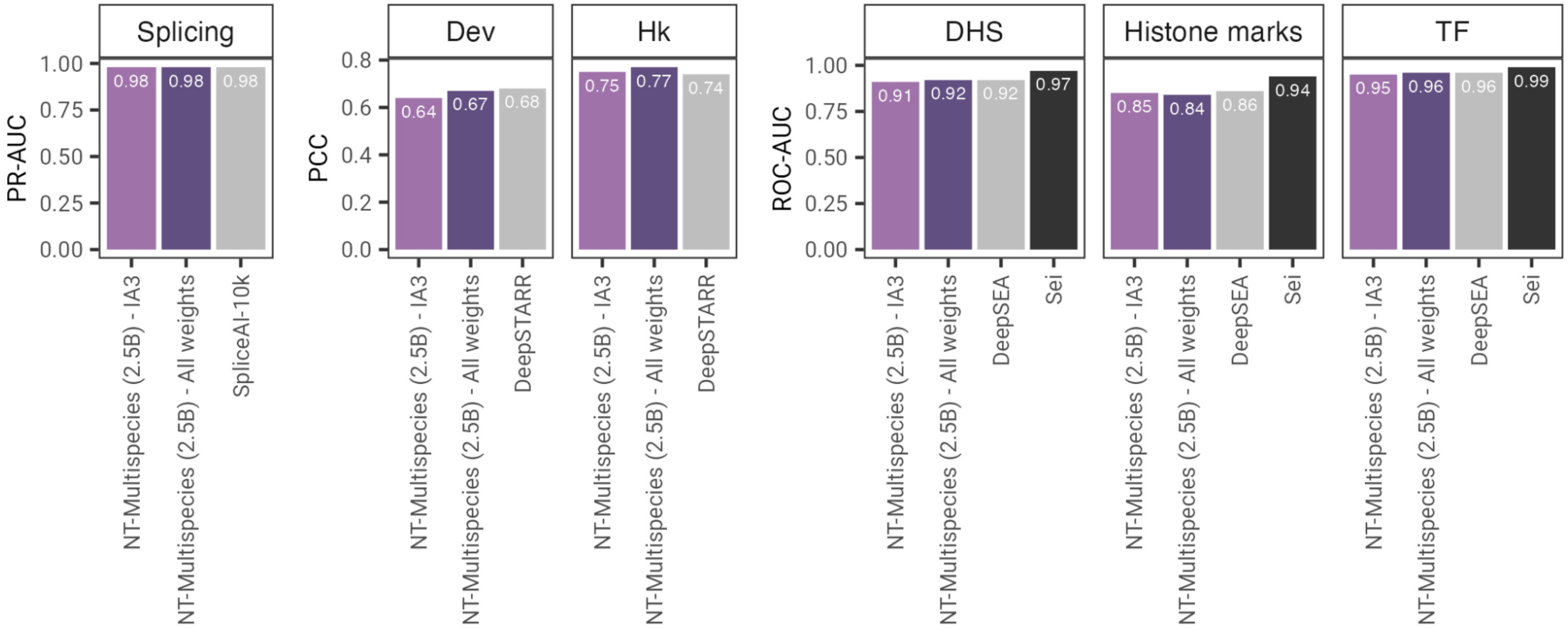
- Nucleotide Transformer probing experiments
  - Pre-trained model for feature extraction
- Probing performance across layers for the enhancer prediction task
- Variation in performance may also occur when applying partial fine-tuning



https://doi.org/10.1038/s41592-024-02523-z

Full fine-tuning vs efficient fine-tuning

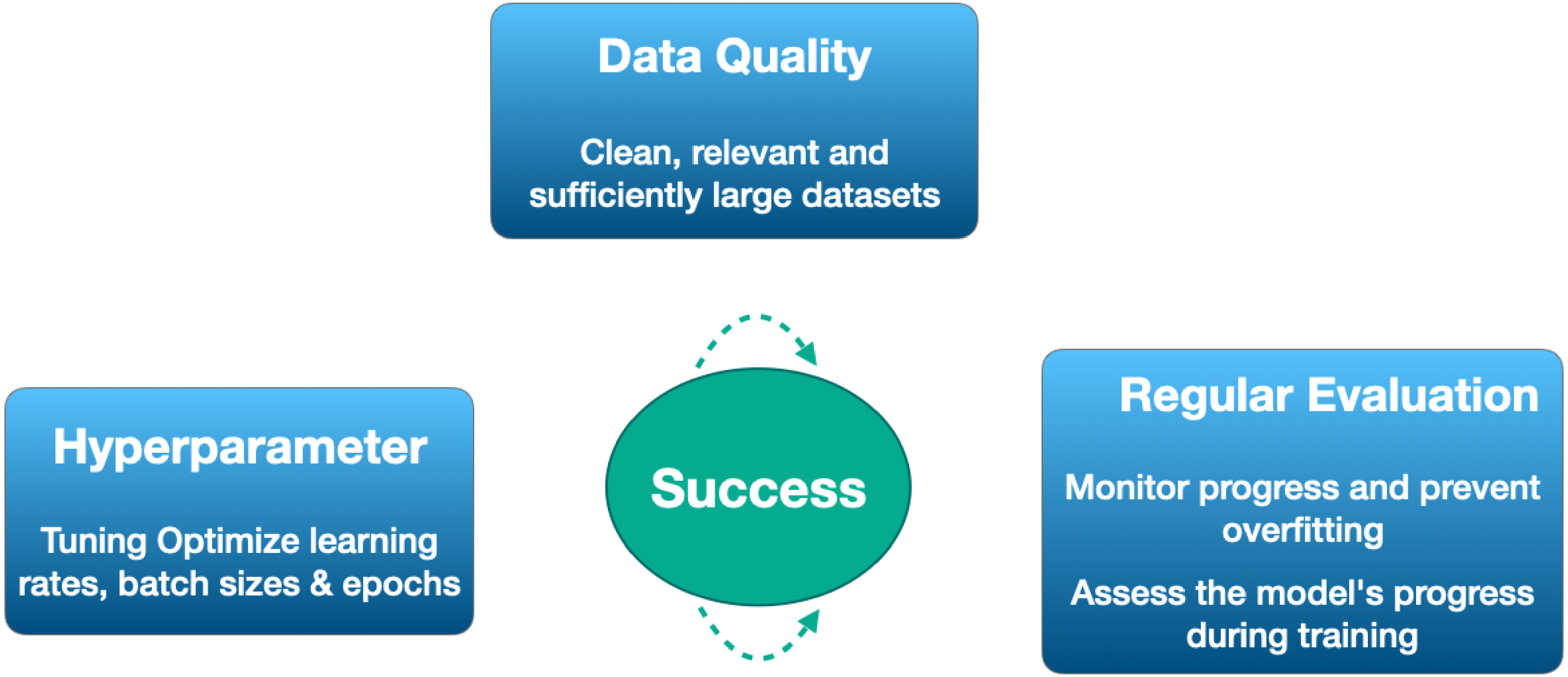
- Nucleotide Transformer experiments
- Parameter **efficient** fine-tuning **compared** with **full-model** fine-tuning
- The performances across **different tasks** are shown and compared with respective baselines



The Rosalind Franklin Institute

https://doi.org/10.1038/s41592-024-02523-z

Fine-tuning - Best Practices



Avoiding LLM Fine-Tuning Pitfalls

🔴 Overfitting

- High training accuracy, poor generalization
- Caused by small datasets or too many epochs

🔵 Underfitting

- Insufficient learning of the task
- Result of inadequate training or low learning rates

🟡 Catastrophic Forgetting

- Loss of broad knowledge during task-specific training
- Reduces versatility

🟣 Data Leakage

- Training/validation overlap
- Leads to misleading performance metrics
- Keep datasets separate!

Hands-on: Fine-tuning and model evaluation

Hands-on

- Practical exercises about
  - Fine-tuning
  - LLMs: DNABERT2 and Nucleotide Transformer

- Evaluation and comparison
  - Pre-trained vs fine-tuned models

- Datasets
  - DNABERT2 benchmark: https://github.com/MAGICS-LAB/DNABERT_2

- Practical code: https://github.com/fabianagoes/bc2_tutorial8