Forecasting Morphogenesis in Latent and Image Space

Alejandro Cobo
Angel David Rodriguez
Prof: Andres Morales-Navarrete, PhD

Yachay Tech University

October 31, 2025

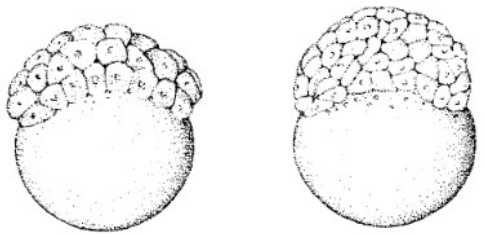
Data & Context: Forecasting Zebrafish Morphogenesis

The data:

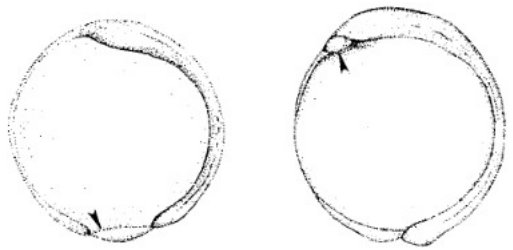
- **Source:**Time-lapse brightfield microscopy of zebrafish embryos.
- **Dataset:** 6 embryo sequences (2 Normal, 4 Mutant).
- **Timeframe:** 2 to 16 hours post-fecundation (hpf),
- **Structure:** 450 frames per embryo.

Biological Context (Kimmel et al., 1995):

- **Blastula (approx. 2-5 hpf):**



- **Gastrula (approx. 5-10 hpf):**



Project Goal & The Problem We Address

The Problem: The Limits of Static Classification.

- Previous work, like the EmbryoNet paper, has successfully used AI to CLASSIFY phenotypes. This is a reactive approach: it identifies a defect after it has already begun to manifest.
- The challenge is that this tells us what is wrong, not what will go wrong.

Our Goal:
Our project moves from classification to proactive forecasting:

"Given this embryo's past development, what will it look like 1, or 2 hours into the future?"

Primary Objective:
To build, train, and validate a deep learning model that can:

- Learn the complex, dynamic rules of zebrafish morphogenesis.
- Generate a sequence of predicted future frames based on a sequence of past frames.

The global plan

1. **Input:** Raw Spatio-Temporal Data.
   Start with the brightfield time-lapse videos (e.g., 6 embryos, 450 frames each).
2. **Preprocessing & Sequencing:**
   - Normalize: normalize pixel values to [0,1].
   - Create Sequences: Use a "sliding window" approach to create (X, y) pairs. (X: A sequecne of N past frames; y: The next M future frames).
   - Split data: Create Training, Validation and Test sets.
3. **Model Core:** ConvLSTM Forecaster
   - Deeply analyze and adapt the model from a external repository to fit our specific data.
   - Train the model on the (X, y) sequences.
4. **Output & Validation:**
   Measure the model's accuracy.

Plan for the next week

Our immediate goal is to move from a "working notebook" to a "validated baseline" that we fully understand.

- Perform a code review of the ConvLSTM model and architecture.

Purpose

We must understand exactly how the model processes data and makes predictions before we can confidently improve it.

- Implement a script to generate visual predictions from our validation set.

Purpose

To see what our current Val Loss actually looks like.

- Begin systematic hyperparameter and parameter tuning.

Purpose

Now that we can see our baseline, we will adjust parameters (like learning rate, sequence length, etc.) to directly improve the visual accuracy of the predictions.