The Sensor Organism

Naums Mogers
Dr Martin A Trefzer
Dr Dimitris Lagos

1 What?
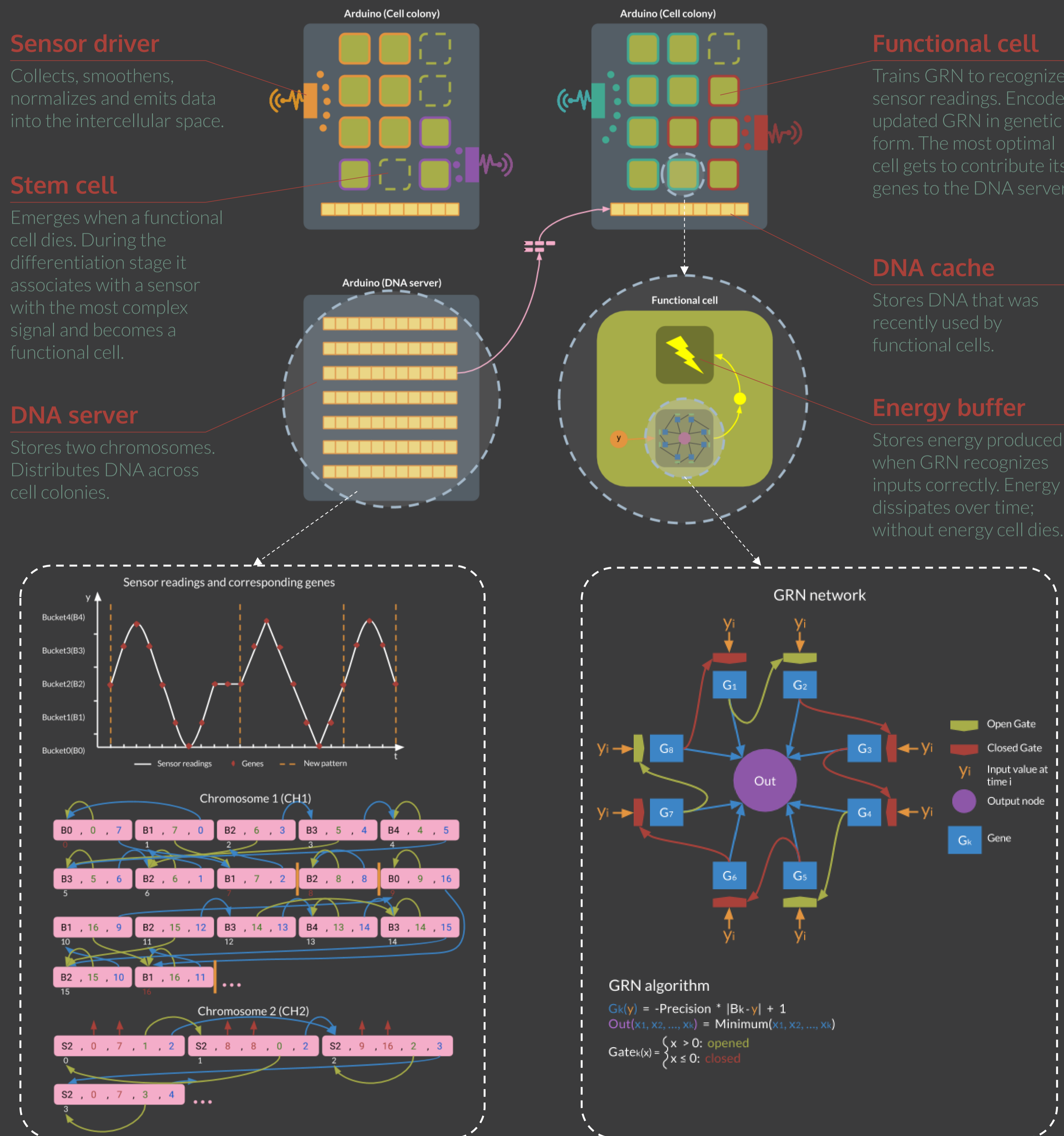
Biologically-inspired heterogeneous multicellular model of Wireless Sensor Network (WSN) based on Genetic Regulatory Network (GRN). With little prior knowledge of problem domain the model learns to recognize patterns in sensor readings. The model solves the problems of anomaly detection and fault detection.

The model is optimized for constrained embedded platform such as Arduino microcontroller.

2 Why?

Biological cell systems that this system is modelled after self-organize and adapt to the environment – a property necessary for a generic WSN. They are fault-tolerant, flexible and scalable; inherently parallelizable, they are good fit for distributed systems such as WSN.

3 Project outputs

UML-model defined on four levels: organism, cell colony, single cell and GRN.

C++ OOP library implementing ZigBee Radio Communication functionality and several components of the model.

Sensor driver

Collects, smoothens, normalizes and emits data into the intercellular space.

Stem cell

Emerges when a functional cell dies. During the differentiation stage it associates with a sensor with the most complex signal and becomes a functional cell.

DNA server

Stores two chromosomes. Distributes DNA across cell colonies.

Arduino (Cell colony)

Arduino (Cell colony)

Arduino (DNA server)

Functional cell

Functional cell

Trains GRN to recognize sensor readings. Encodes updated GRN in genetic form. The most optimal cell gets to contribute its genes to the DNA server.

DNA cache

Stores DNA that was recently used by functional cells.

Energy buffer

Stores energy produced when GRN recognizes inputs correctly. Energy dissipates over time; without energy cell dies.

4 How?

Model defines two Arduino specializations: cell colony and DNA server. Cell colony consists of sensors, sensor drivers and cells. Initially colony is filled with stem cells; as time goes, stem cells train on sensor readings, specialize in particular patterns and transform into functional cells. By recognizing patterns correctly, cells obtain energy; if energy runs out, cells die.

DNA Server is an Arduino dedicated to store genetically encoded data, that is shared between cell colonies. Only the most optimal cells get to contribute to the DNA server.

The anomaly is reported when cell death rate rises, i.e. when majority of cells do not recognize the pattern and die.

The model requires supervision during the learning stage, when anomalies are reported frequently as all patterns are observed for the first time.

5 Applications

This generic WSN model is potentially applicable to any problem that is solvable using a WSN. However, it is expected to be the most optimal solution when input data changes quickly and in different ways. Examples include Machine health monitoring, Structural health monitoring, Terrain mapping using moving agents and Biometric observation.

Sensor readings and corresponding genes

y

Bucket4(B4)
Bucket3(B3)
Bucket2(B2)
Bucket1(B1)
Bucket0(B0)

t

— Sensor readings   • Genes   --- New pattern

Chromosome 1 (CH1)

| B0 , 0 , 7 | B1 , 7 , 0 | B2 , 6 , 3 | B3 , 5 , 4 | B4 , 4 , 5 |
| 0 | 1 | 2 | 3 | 4 |

| B3 , 5 , 6 | B2 , 6 , 1 | B1 , 7 , 2 | B2 , 8 , 8 | B0 , 9 , 16 |
| 5 | 6 | 7 | 8 | 9 |

| B1 , 16 , 9 | B2 , 15 , 12 | B3 , 14 , 13 | B4 , 13 , 14 | B3 , 14 , 15 |
| 10 | 11 | 12 | 13 | 14 |

| B2 , 15 , 10 | B1 , 16 , 11 | ...
| 15 | 16 |

Chromosome 2 (CH2)

| S2 , 0 , 7 , 1 | S2 , 8 , 8 , 0 , 2 | S2 , 9 , 16 , 2 , 3 |
| 0 | 1 | 2 |

| S2 , 0 , 7 , 3 , 4 | ...
| 3 |

GRN network

y_i y_i y_i y_i

G_1 G_2 G_3 G_4 G_5 G_6 G_7 G_8

Out

y_i y_i y_i y_i

Legend:
- Open Gate
- Closed Gate
- y_i Input value at time i
- Output node
- G_k Gene

GRN algorithm

$G_k(y) = -\text{Precision} * |B_k - y| + 1$

$\text{Out}(x_1, x_2, ..., x_k) = \text{Minimum}(x_1, x_2, ..., x_k)$

$$\text{Gate}_k(x) = \begin{cases} x > 0: & \text{opened} \\ x \leq 0: & \text{closed} \end{cases}$$