

# **Análisis de las transiciones de comportamiento en crecimiento de tumores usando una simulación con autómata celular**

Sergio Rodríguez Calvo

Diciembre 2017



# Índice

1. Objetivos y alcance
2. Estado del arte
3. Introducción a los autómatas celulares
4. Introducción a la enfermedad del cáncer
5. Modelización
6. Parámetros de la simulación
7. Modelo de eventos
8. Mitosis
9. Implementación
10. Experimentos
11. Resultado propio
12. Demostración



# Objetivos y alcance

- **Objetivos:**
  - Encontrar artículos científicos sobre “Simulación de crecimiento de tumores con autómatas celulares”.
  - Extraer información de los artículos y seleccionar candidatos.
  - Hacer implementación propia y reproducir los experimentos y sus resultados.
  - Comparar los resultados obtenidos.
- **Alcance:**
  - Implementación propia siguiendo la modelización de los autores y reproducir los experimentos del artículo.



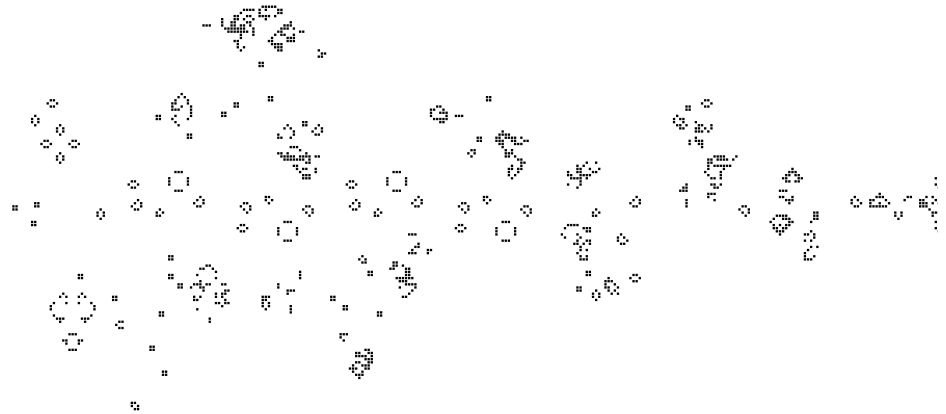
## Estado del arte

- Simulated Brain Tumor Growth Dynamics Using a Three-Dimensional Cellular Automaton, de A. Kansal, S. Torquato, G. Harsh, E. Chiocca, T. Deisboeck, 2000.
- A cellular automaton model for tumour growth in inhomogeneous environment, de T. Alarcon, H. Byrne, P. Maini, 2003.
- A Cellular automata model of tumor-immune system interactions, de D.G. Mallet, L.G. De Pillis, 2006.
- Cellular Automaton of Idealized Brain Tumor Growth Dynamics, de A. Kansal, S. Torquato, G. Harsh, E. Chiocca, T. Deisboeck, 2009.
- Analysis of behaviour transitions in tumor growth using a cellular automaton simulation, de José Santos, Ángel Monteagudo, 2014.

# Introducción a los autómatas celulares



*Conus textile shell*



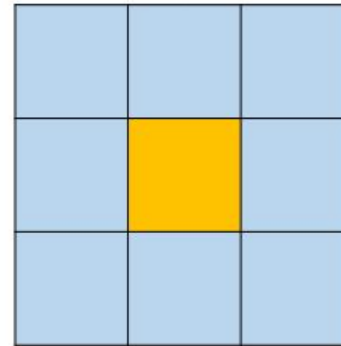
Conway's game of life

# Introducción a los autómatas celulares

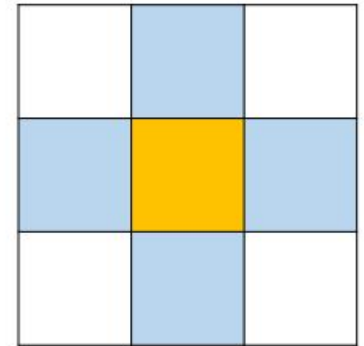


SE						W		E			SW	S
						SW	S	SE				
		NW	N	NE								
		W		E								
		SW	S	SE								
NE											NW	N
E						NW	N	NE			W	

Moore neighborhood



Von Neumann neighborhood





# Introducción al cáncer

- Nombre genérico para agrupar a más de 200 enfermedades que provocan proliferación descontrolada de células.
- Es una enfermedad genética, multigénica, multifactorial y multiorgánica.
- Fases:
  - Crecimiento.
  - Angiogénesis.
  - Vascularización (metástasis).

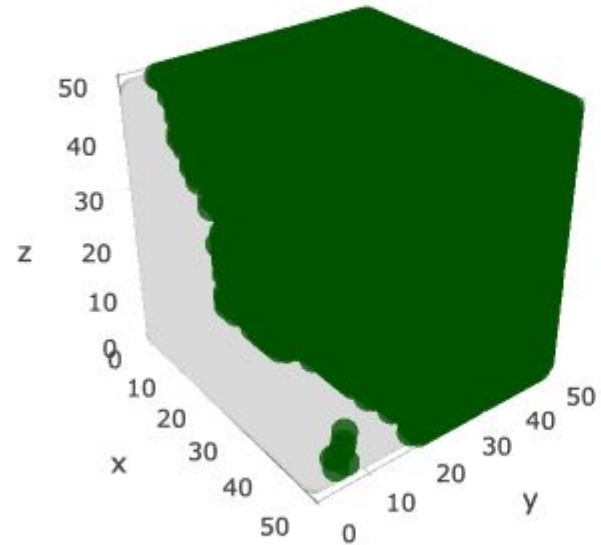
# Introducción al cáncer





# Modelización

- Célula:
  - Genoma: SG, IGI, EA, EI y GI.
  - Propiedades:
    - Tasa de mutación base (m).
    - Tamaño del telómero (tl).
- Rejilla de tamaño 50 posiciones en tres dimensiones.

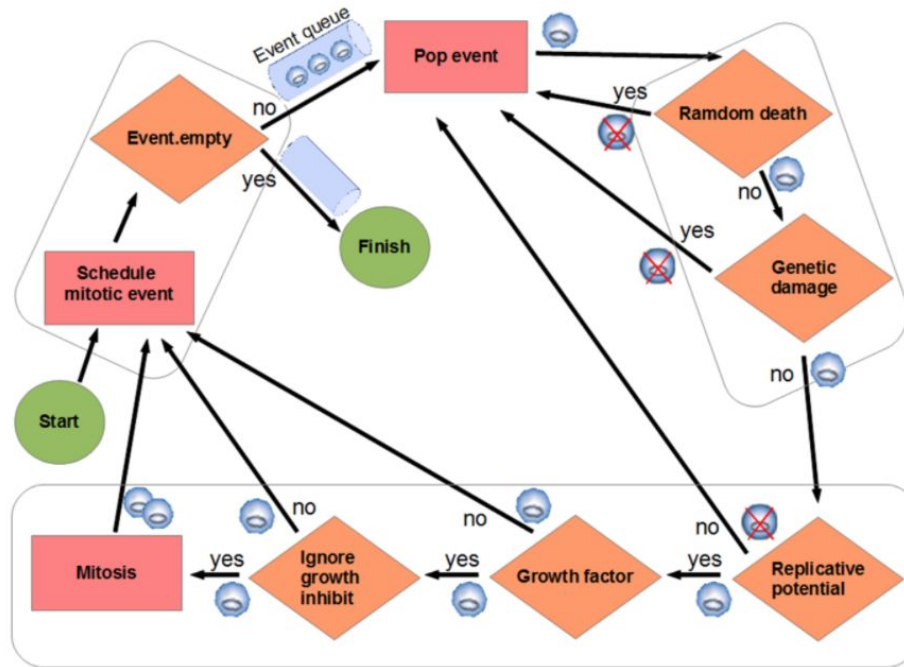




## Parámetros de la simulación

- Valor para **tamaño de la rejilla** (t).
- Valor por defecto para el parámetro *tasa de mutación base* (m).
- Valor por defecto para el parámetro **tamaño del telómero** (tl).
- Valor para definir probabilidad de **muerte por daño genético** (e).
- Valor para parámetro de **factor de incremento de tasa de mutación base** (i).
- Valor para definir la **probabilidad de que una célula cancerosas mate a un vecino** para efectuar la mitosis (g).
- Valor para definir probabilidad de **muerte aleatoria** (a).

# Modelo de eventos





# Mitosis

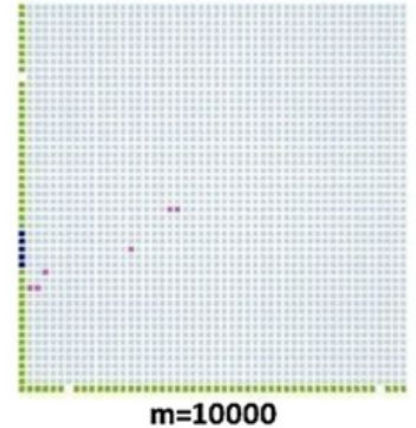
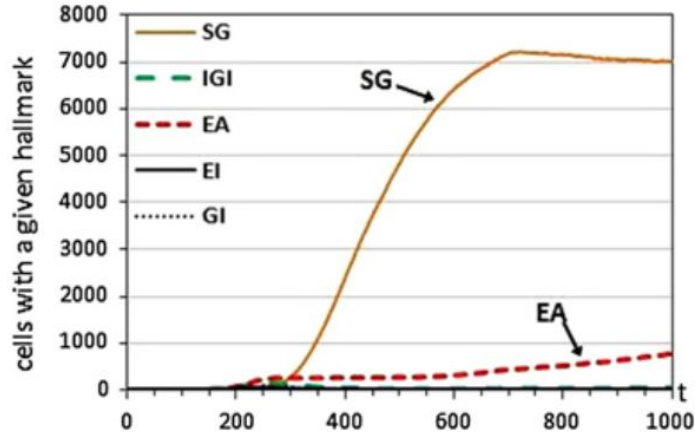
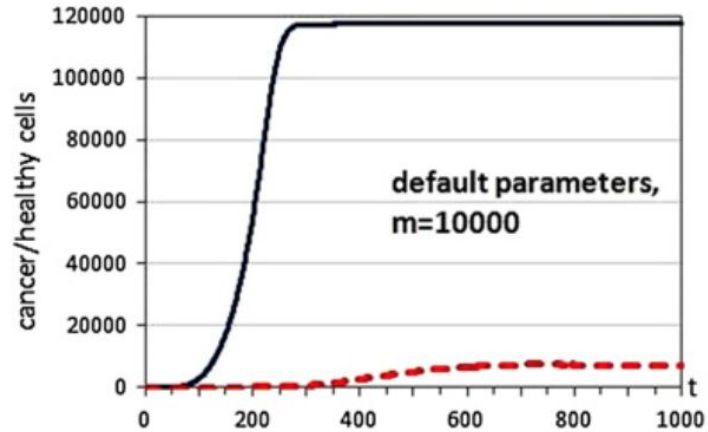
```
def perform_mitosis(self, position, i):  
    self.decrease_telomer()  
    self.increment_base_mutation_rate(i)  
    new_cell = Cell(position, self.genome.sg, self.genome.igi,  
                    self.genome.ea, self.genome.ei, self.genome.gi,  
                    self.tl, self.m)  
    new_cell.add_mutations()  
    return new_cell
```



# Implementación

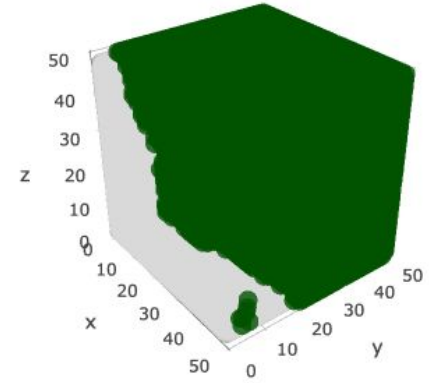
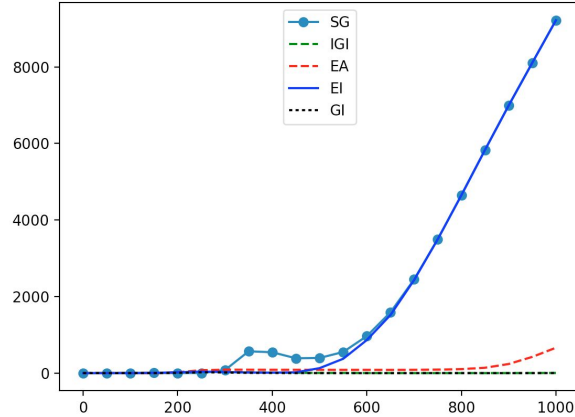
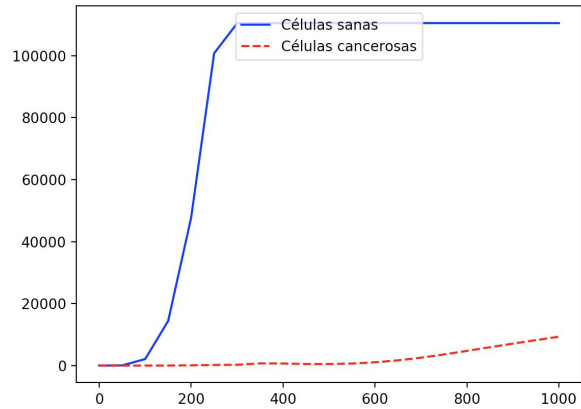
```
def run(self):
    for iteration in range(self.iterations):
        events = self.pop_events(iteration) if iteration in self.mitotic_agenda else []
        for event in events: # event is a tuple with three elements == position
            cell = self.cells[event]
            if self.experiments.random_death_test():
                self.kill_cell(event)
            elif self.experiments.genetic_damage_test(cell.mutations(), cell.genome.ea):
                self.kill_cell(event)
            else:
                test_1, test_2, test_3 = self.first_test(cell),
                                         self.second_test(cell, iteration),
                                         self.third_test(cell)
                if test_1 and test_2 and test_3: #Perform mitosis
                    self.copy_and_choose_new_position(event, cell, iteration)
                    self.push_event(iteration + self.future_mitotic_event(), event)
                elif test_3:
                    self.push_event(iteration + self.future_mitotic_event(), event)
                else: # Telomer is 0 and EI is OFF
                    self.kill_cell(event)
        if iteration in self.mitotic_agenda: # Remove current iteration
            del self.mitotic_agenda[iteration]
```

## Experimento: *tasa de mutación* $m = 10.000$

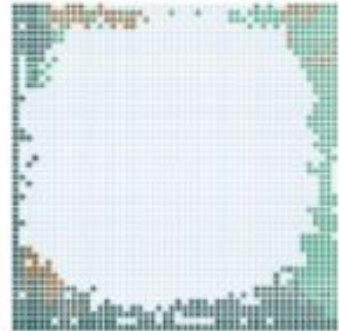
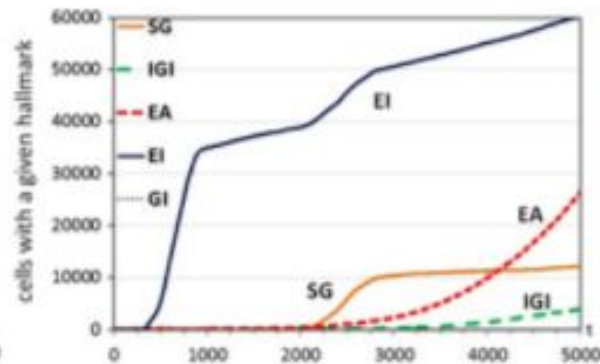
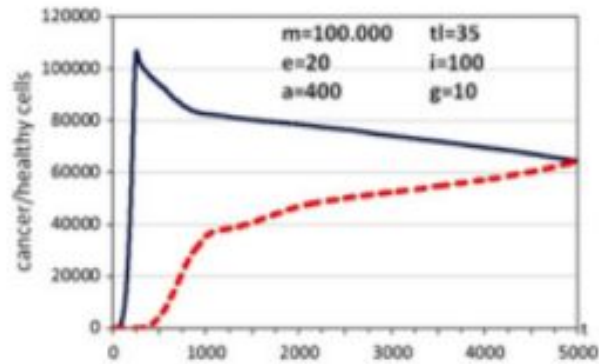




# Resultado propio: *tasa de mutación base $m = 10.000$*

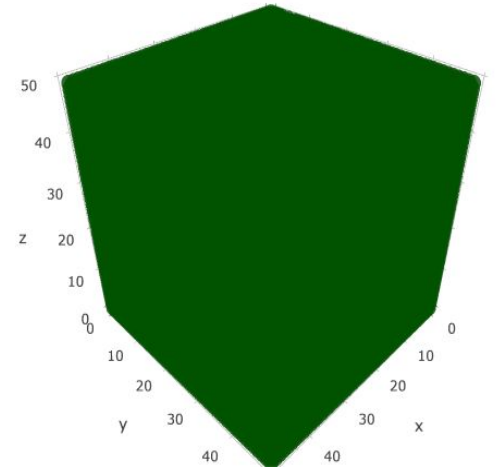
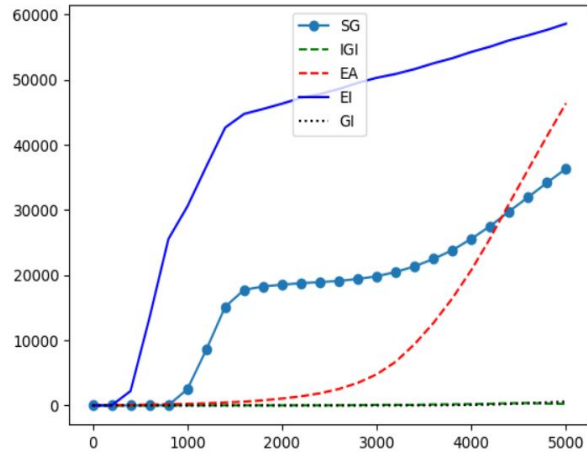
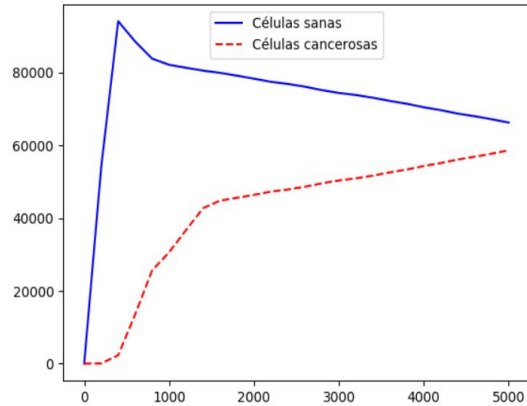


## Experimento: *tasa de mutación* $m = 100.000$

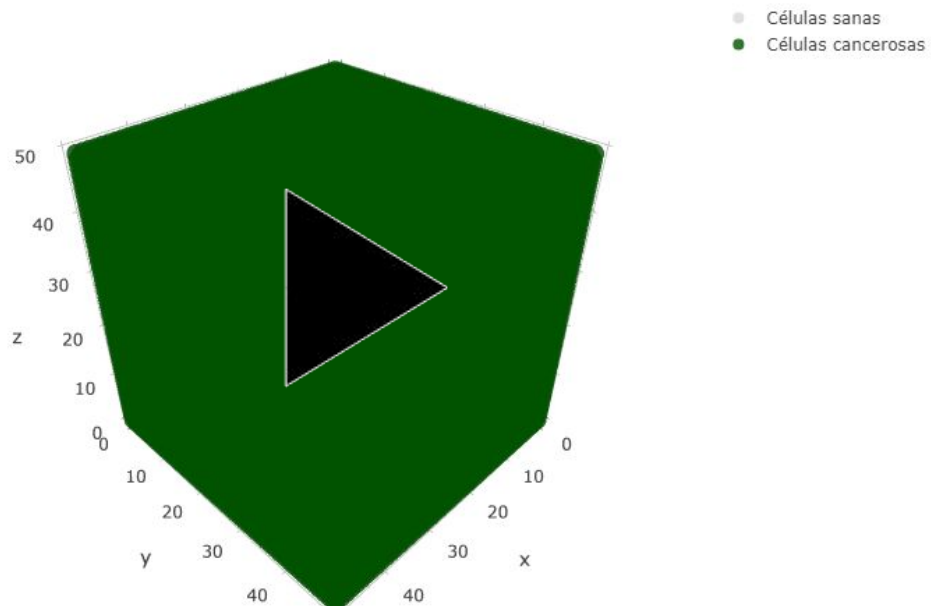




# Resultado propio: *tasa de mutación base m = 100.000*



# Demostración





**¿Alguna pregunta?**

