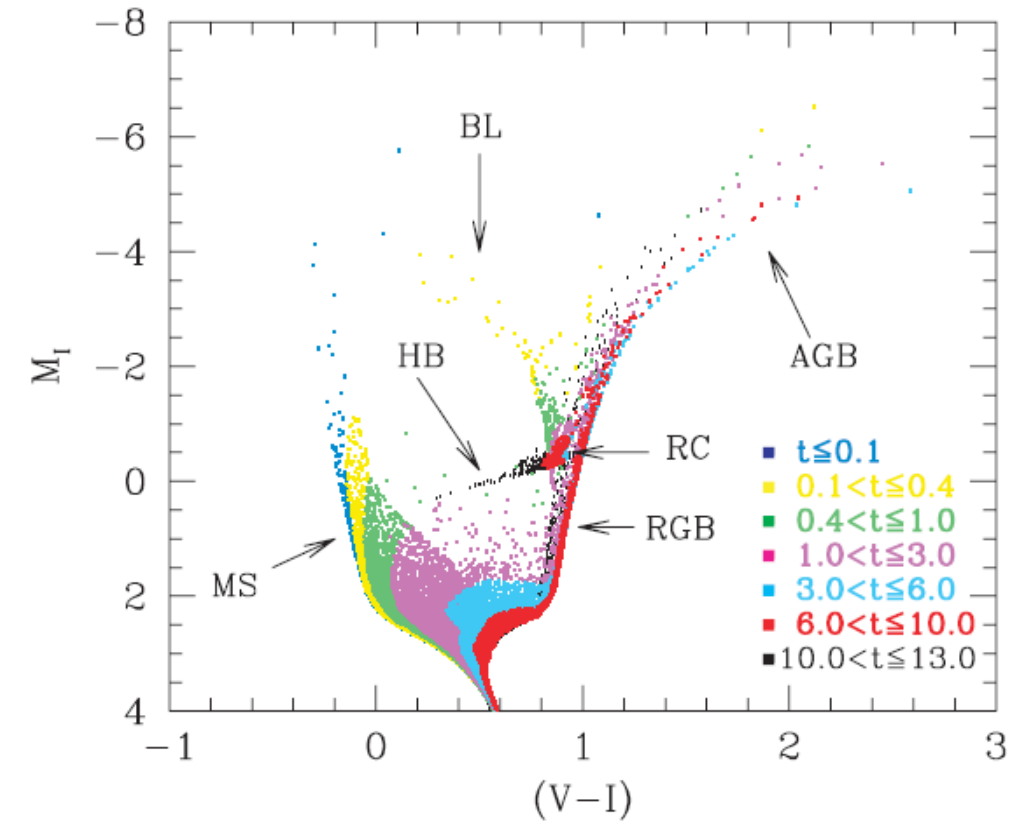
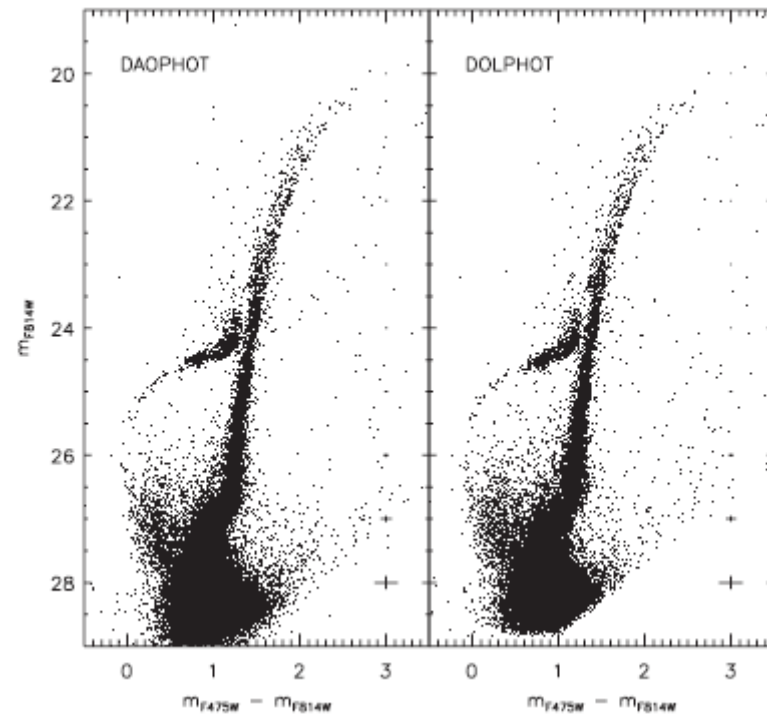




# DIAGRAMAS COLOR MAGNITUD SINTÉTICOS

Enrique A. Galceran García

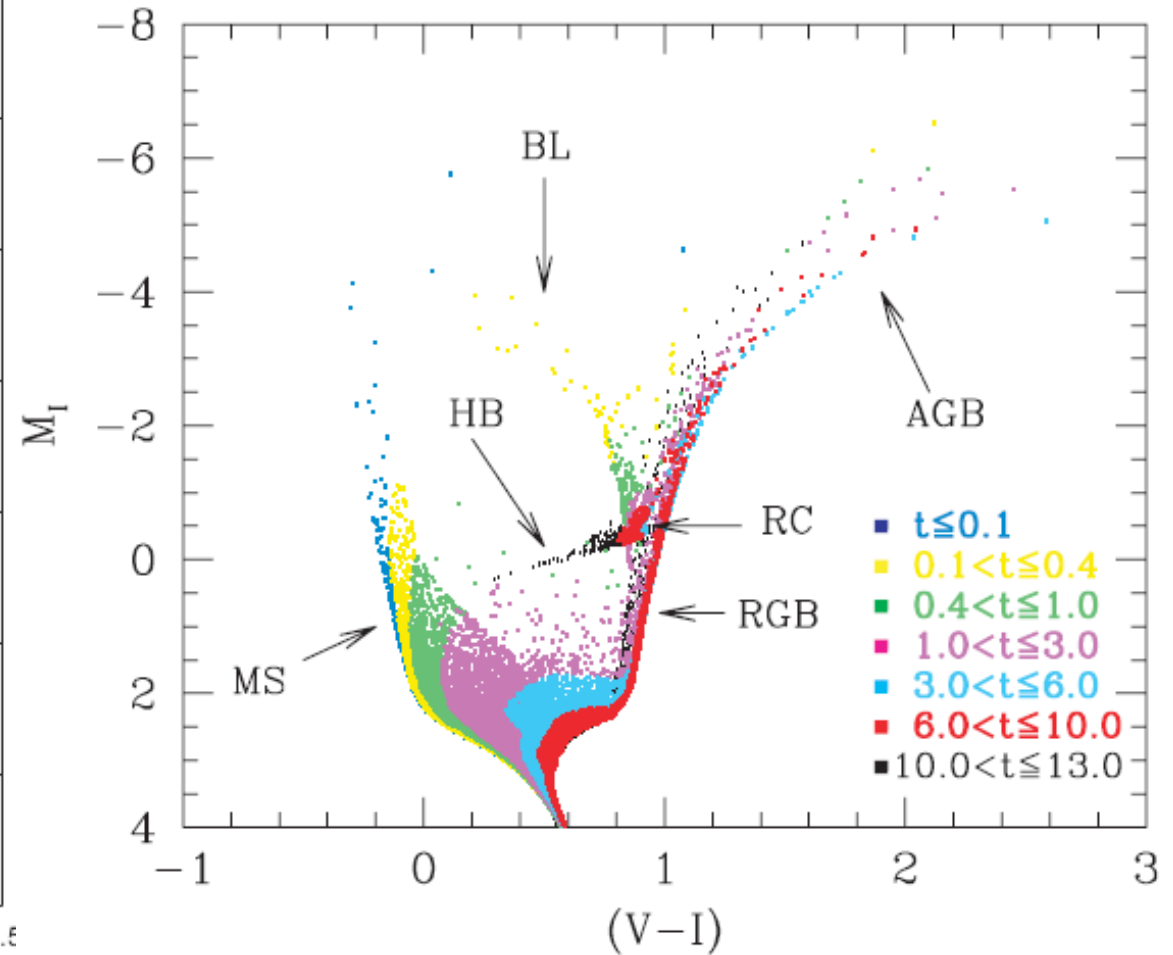
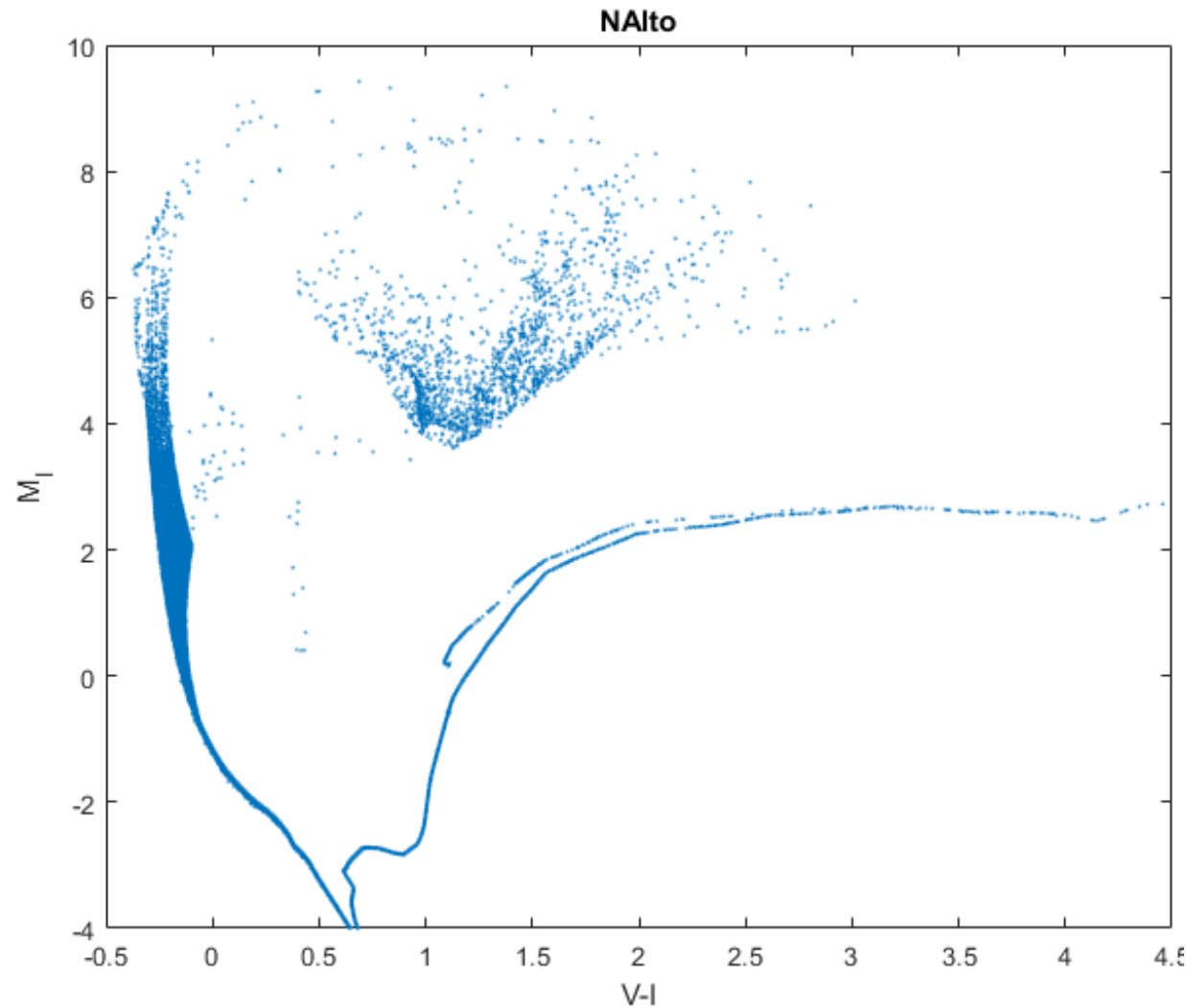
# DIAGRAMAS COLOR MAGNITUD



# OBJETIVO

1. Crear un diagrama sintético de Color-Magnitud.
2. Comparar cómo de bien se ajusta el diagrama sintético con el diagrama problema.
3. Crear muchos (más) diagramas sintéticos.
4. Ver cómo se ajustan estos diagramas nuevos con respecto a los demás.
5. Afinar los ritmos de formación estelar para que se ajusten lo mejor posible a nuestra galaxia buscada.

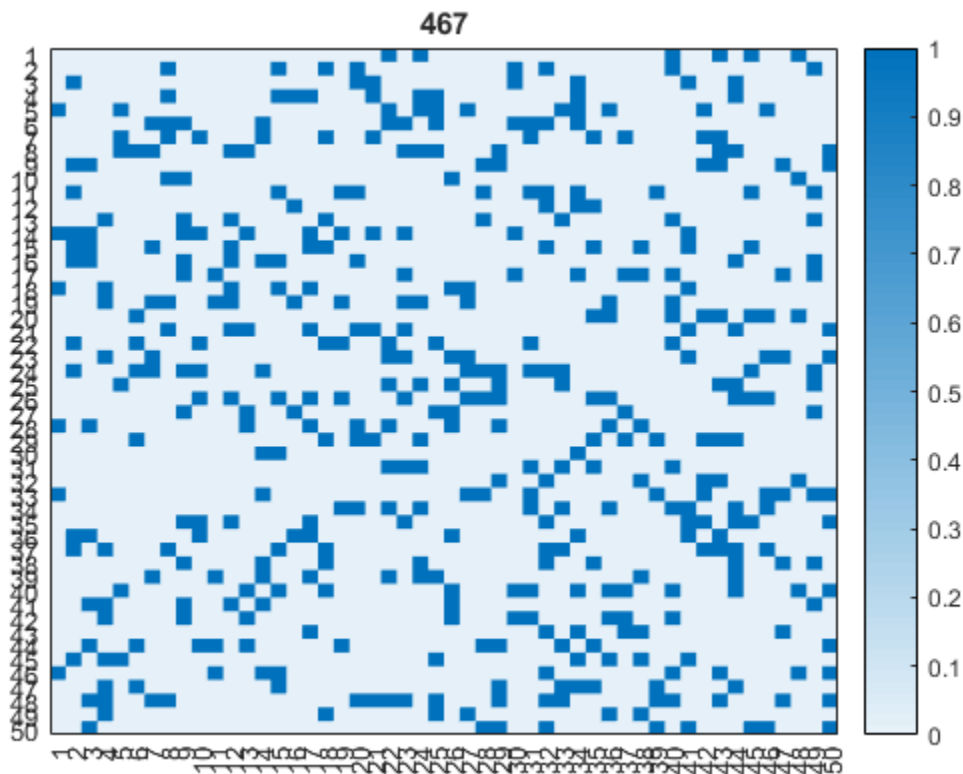
# DATOS PROBLEMA



# MALLAS ESTÁTICAS

- ¿Qué es lo que mejor hacen los ordenadores?
  - Operaciones matemáticas
  - Acciones repetitivas
  - Trabajar muy deprisa
- Locality of Reference
  - Spatial – las variables están muy juntas
  - Temporal – las variables cercanas se llaman rápidamente (sin saltos)

# MAILLA DE DENSIDAD



- Reparte los datos en una matriz de un tamaño fijo.
- Transforma variables continuas a discretas (los ordenadores trabajan mucho mejor con datos discretos)
- El valor por casilla aumenta con la cantidad de estrellas en esa casilla.
- Sin embargo, si comparamos dos mallas, sólo tiene en cuenta si coinciden, no si se parecen



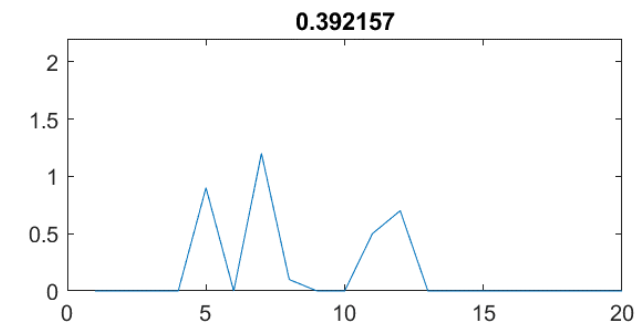
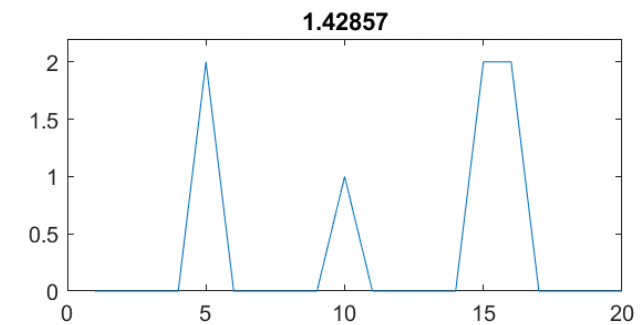
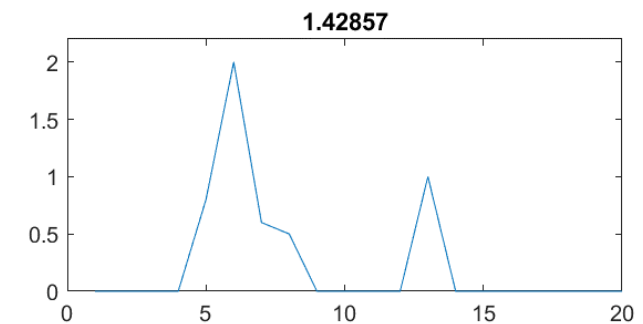
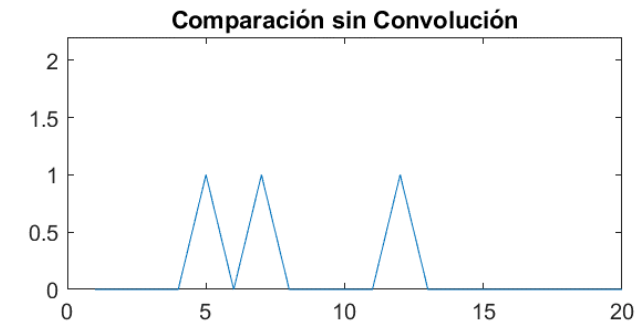
# SIN CONVOLUCIÓN

Reduciendo a una dimensión se puede ver más claramente el problema:

Si no tratamos la información antes, el código no va a tener en cuenta los vecinos.

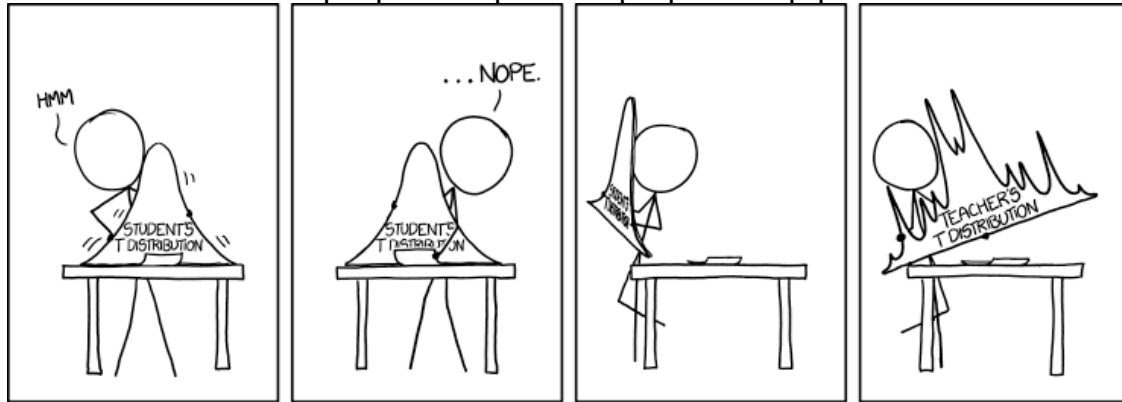
Al aumentar la resolución en el caso de dos dimensiones el problema se agrava.

Solución: Capa de convolución Gaussiana

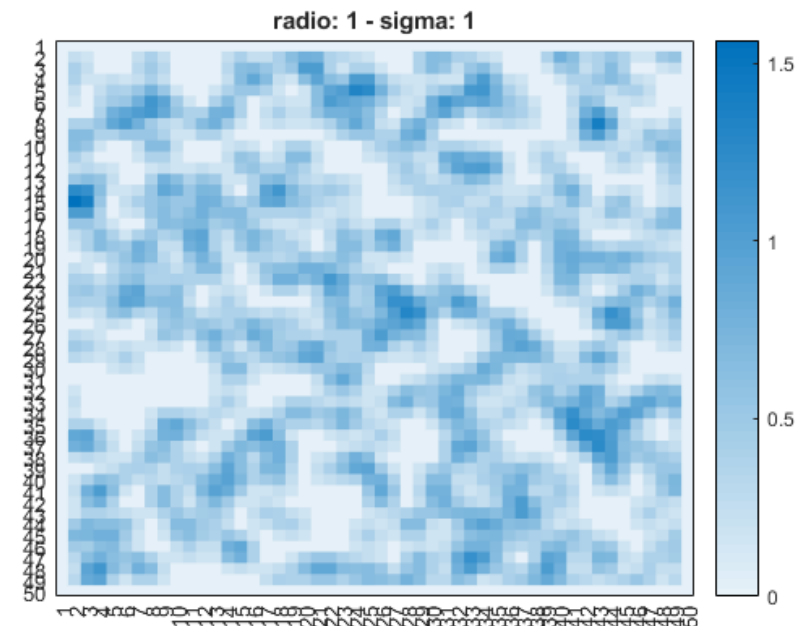
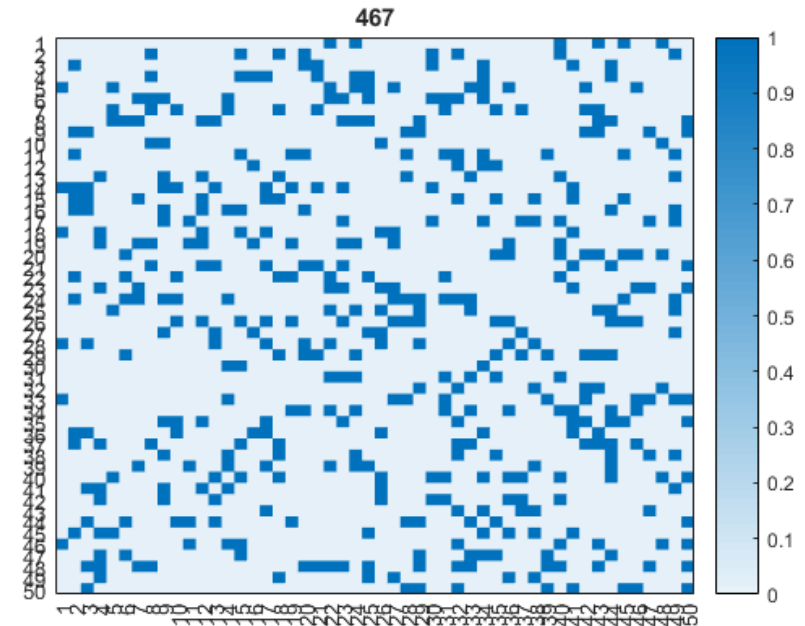


# ¿QUÉ HACE UNA CAPA DE CONVOLUCIÓN?

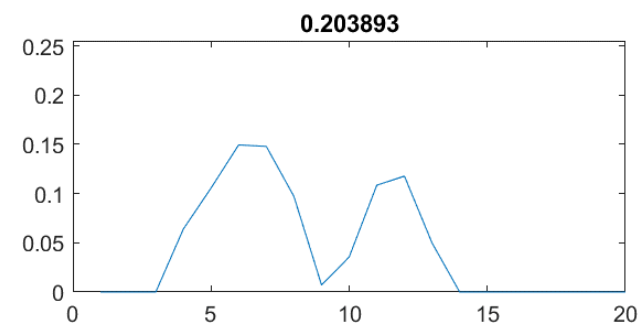
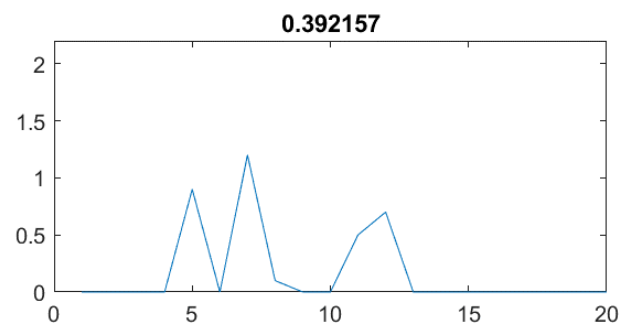
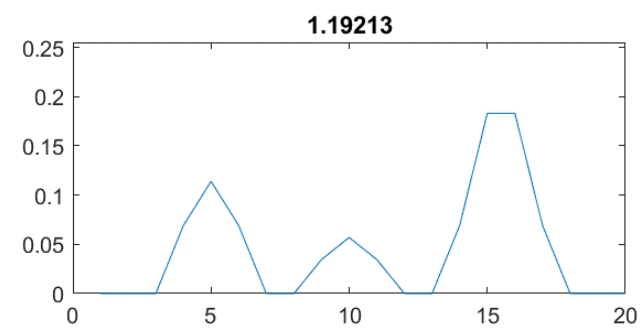
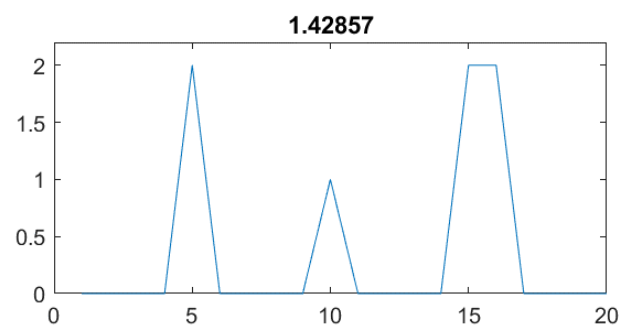
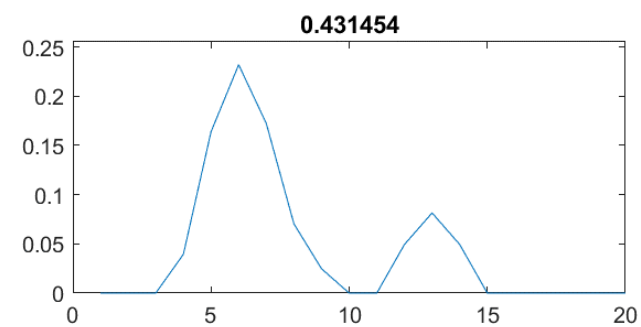
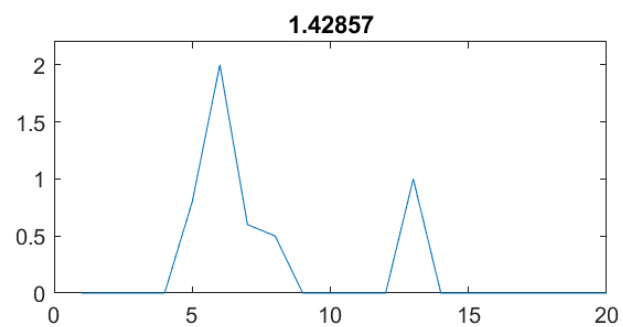
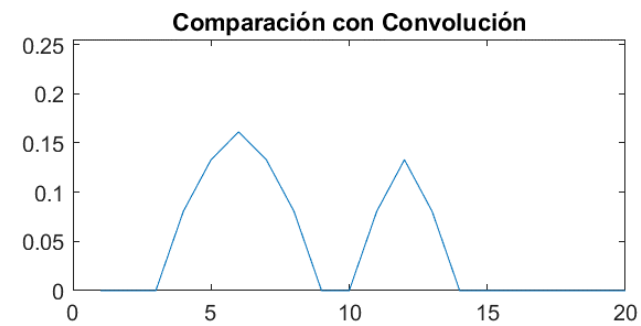
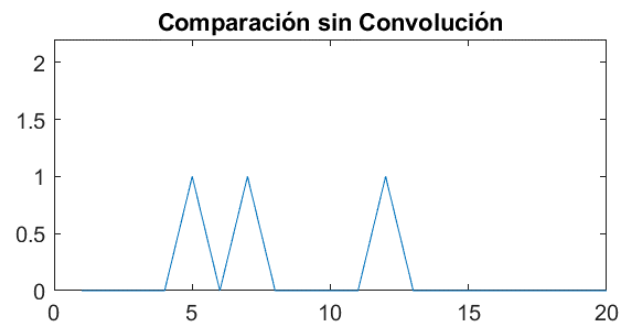
- Reevalúa los datos de una matriz teniendo

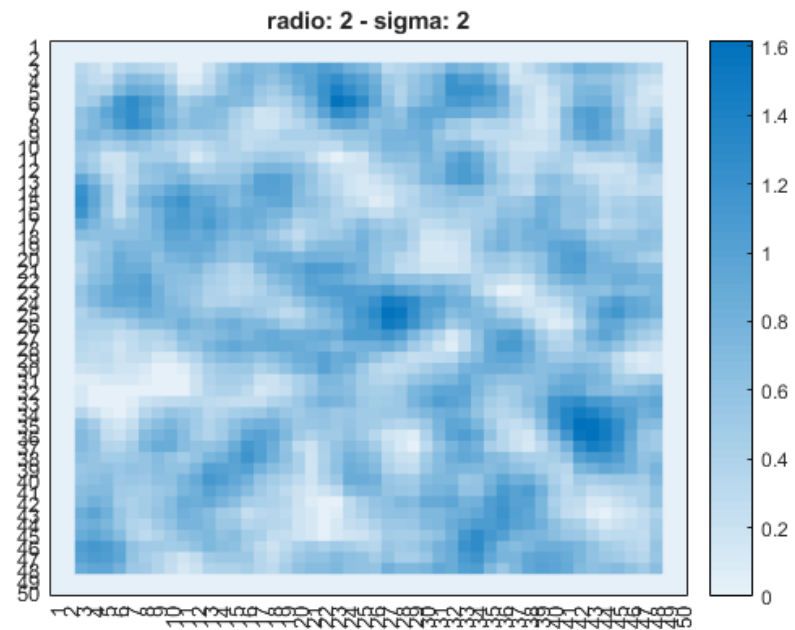
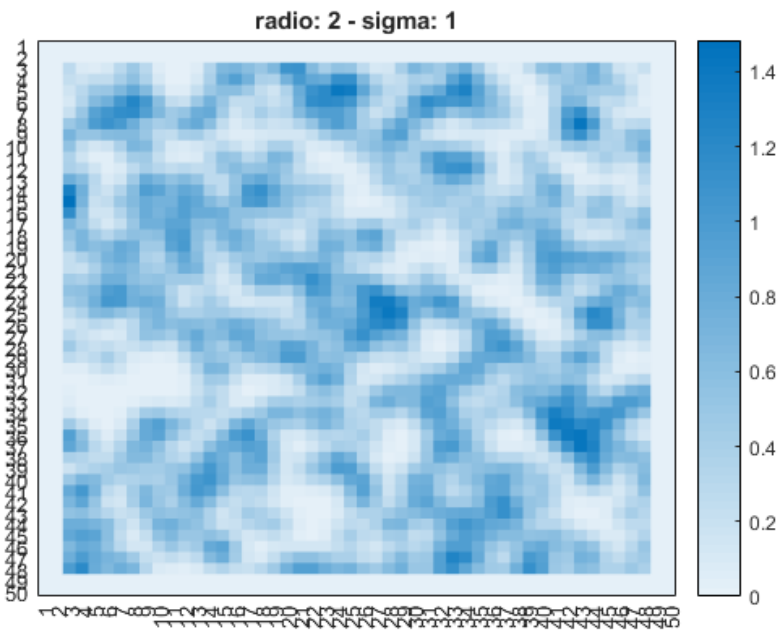
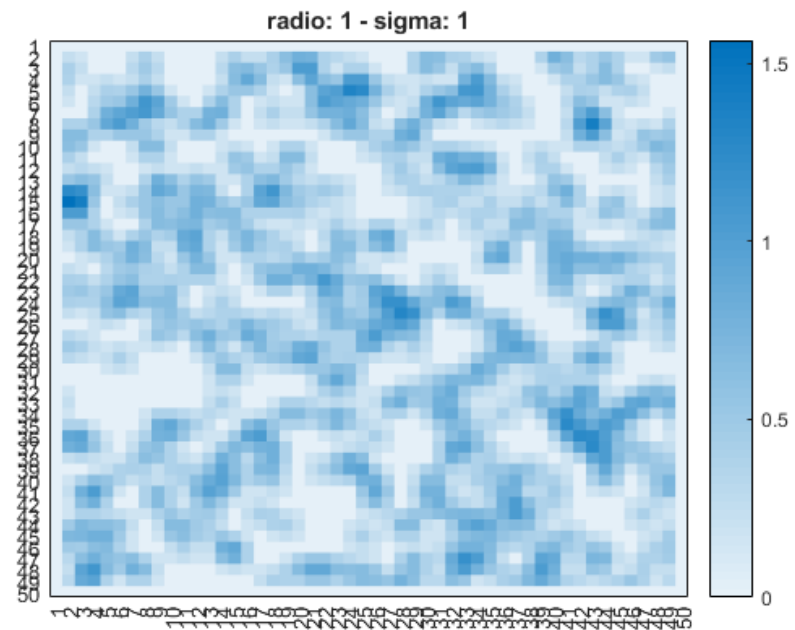
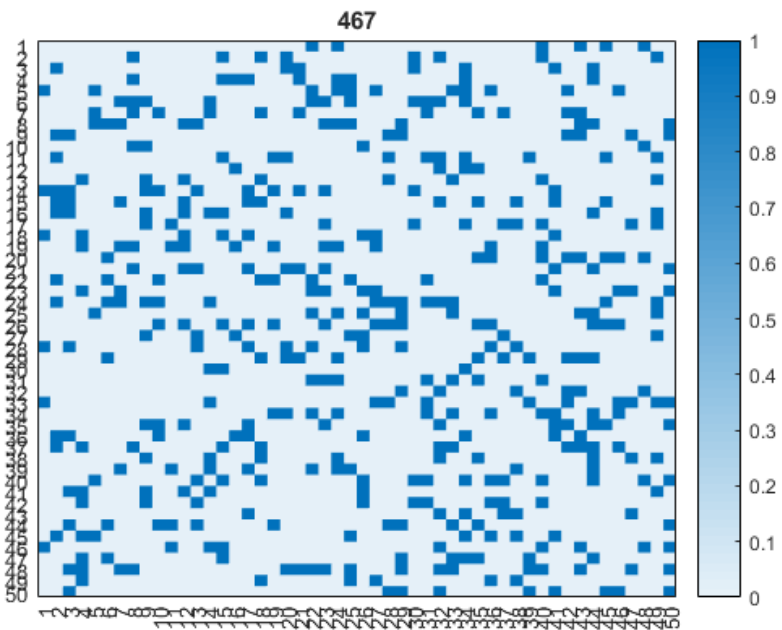


- No tiene por qué ser una normal, puede ser cualquier función.

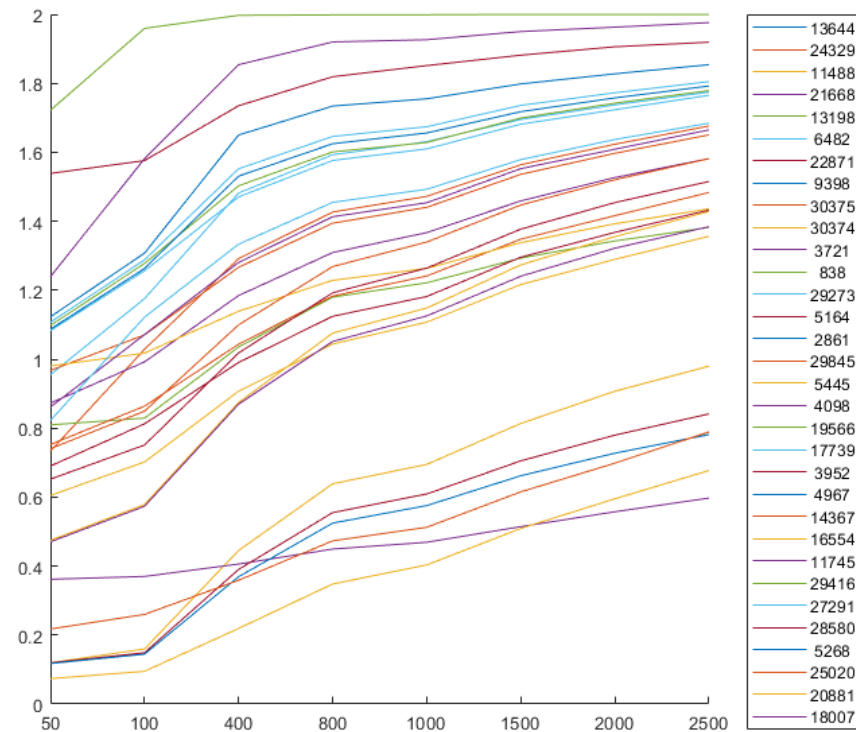




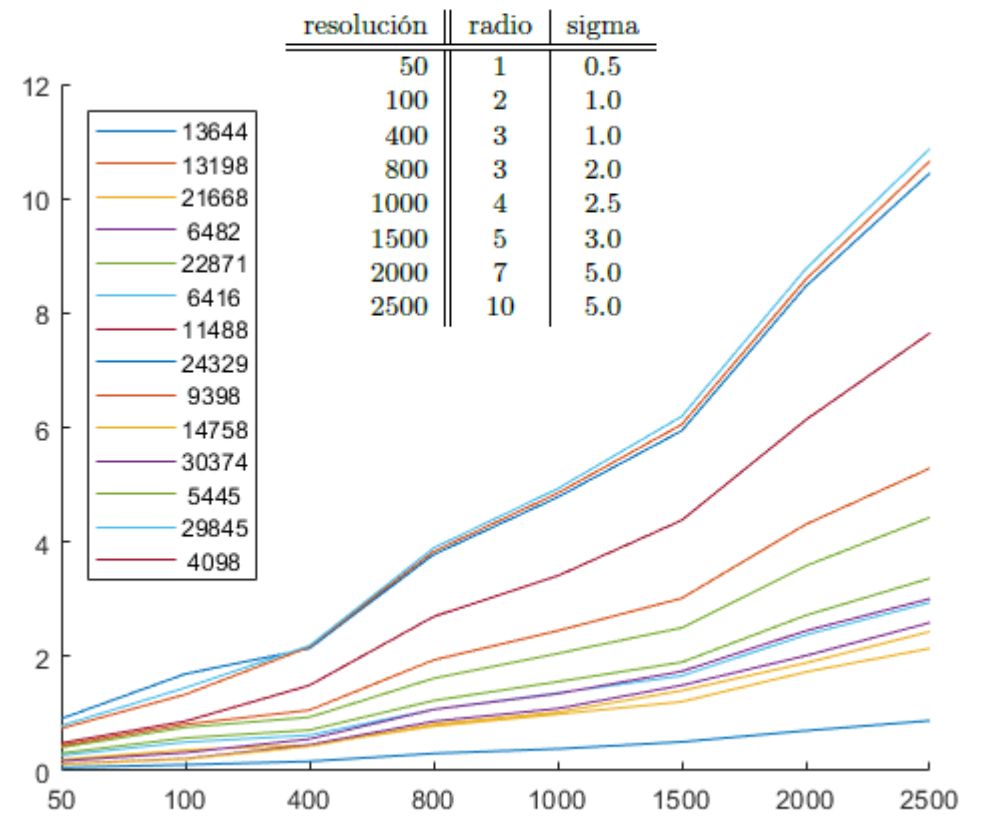




# VALOR DE AJUSTE

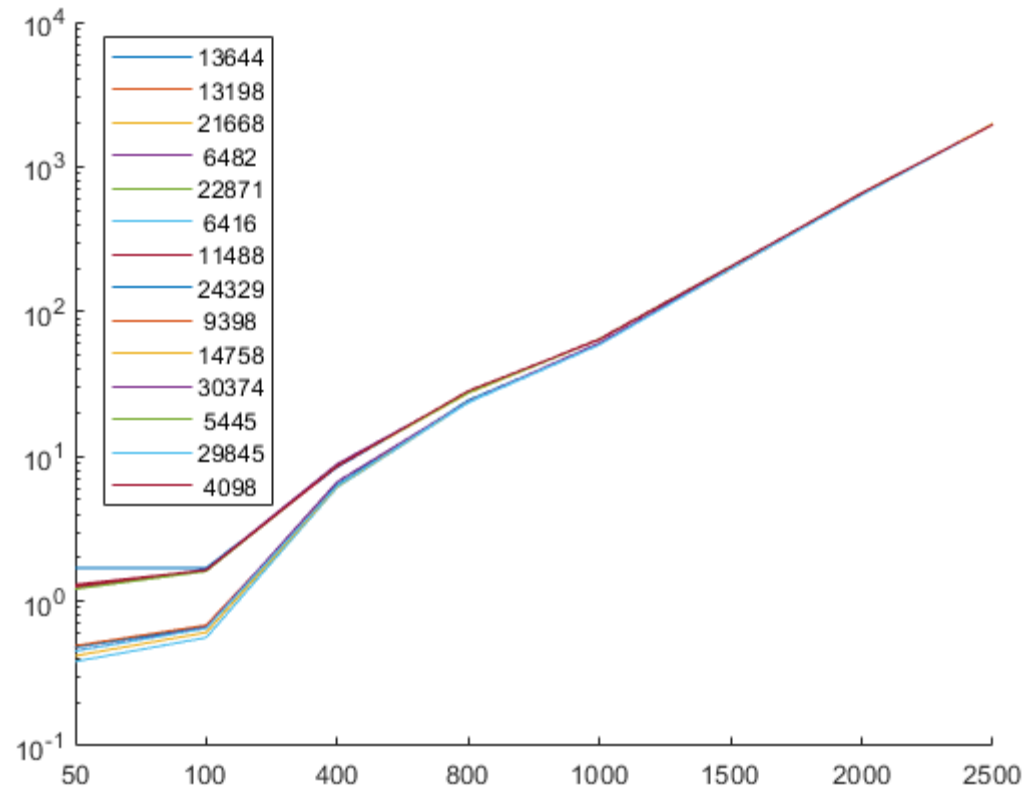


Sin Convolución



Con Convolución

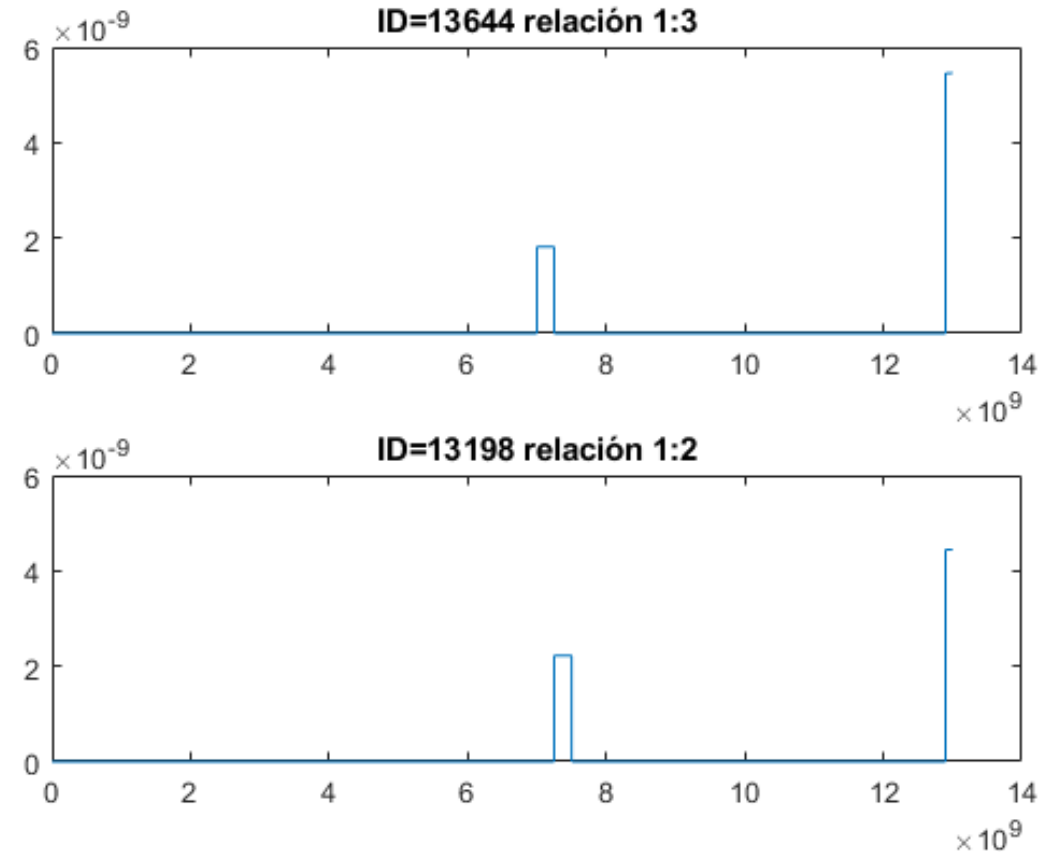
# TIEMPO DE CÁLCULO



- El problema de la convolución es que aumenta mucho el tiempo de ejecución.
- Así pues, con una resolución de 800 es suficiente

## MEJOR SFR:

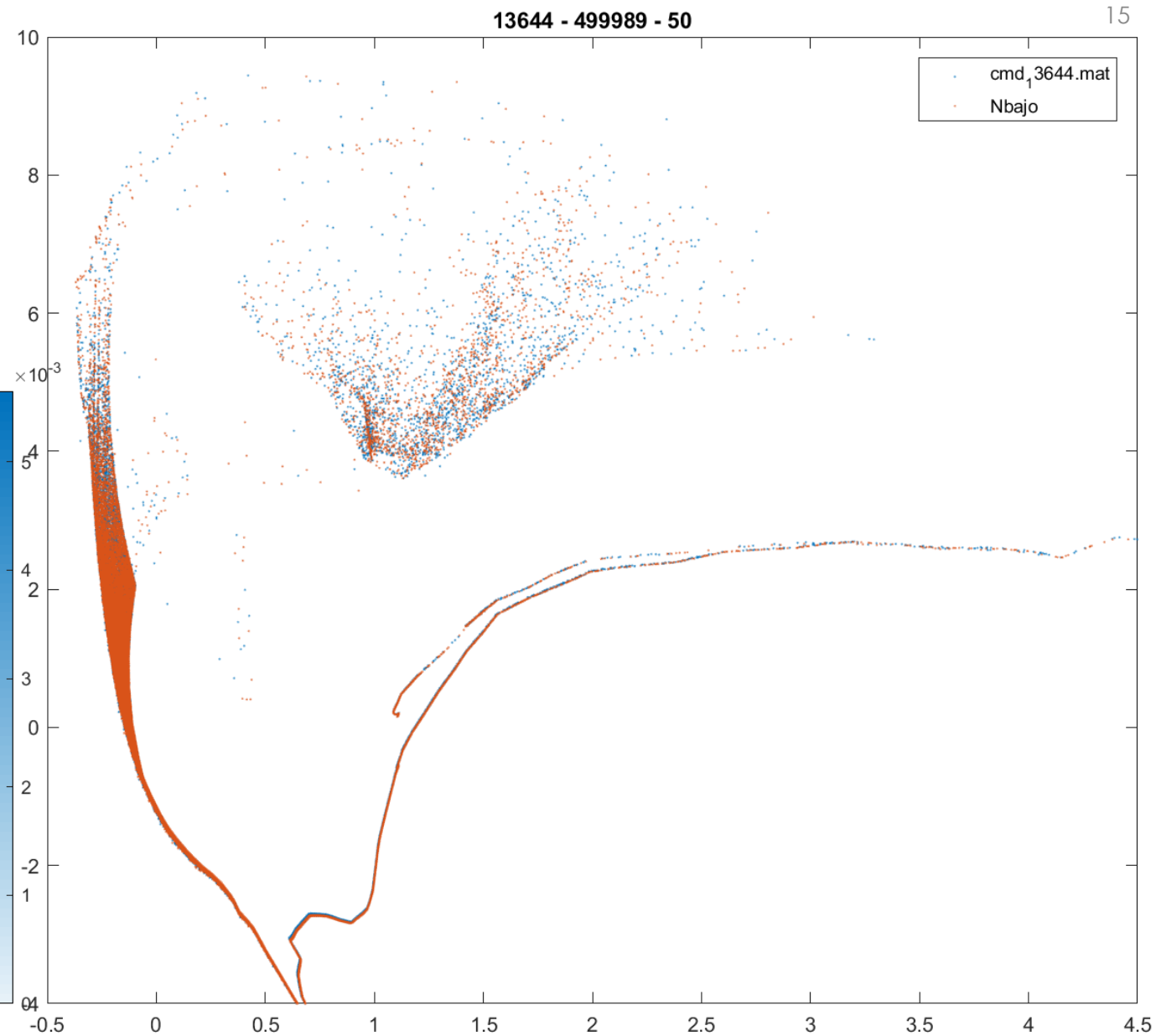
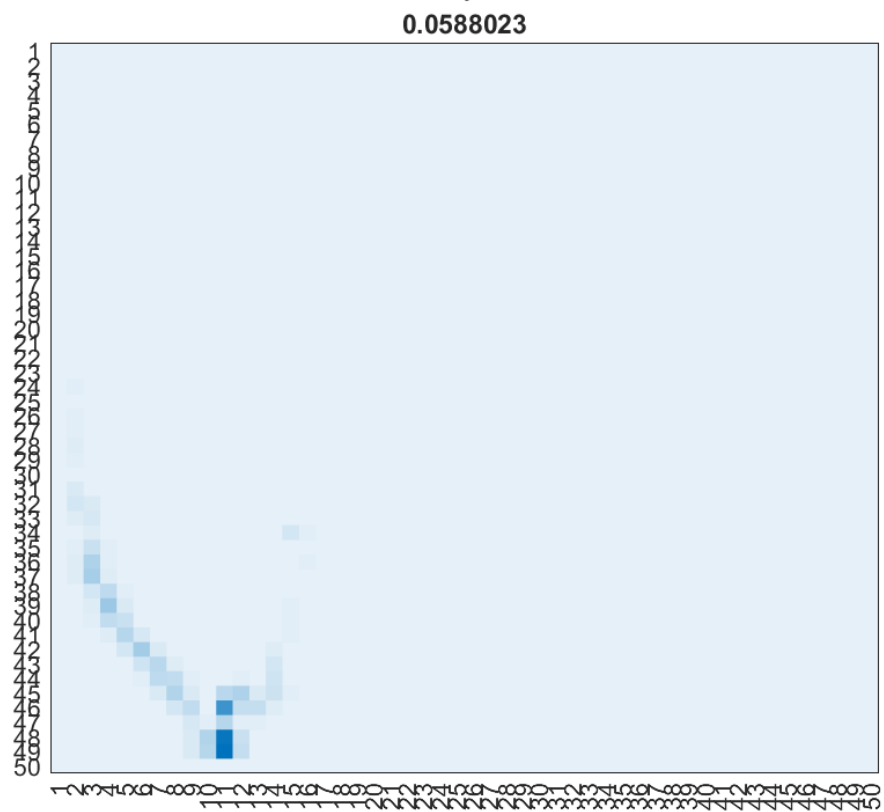
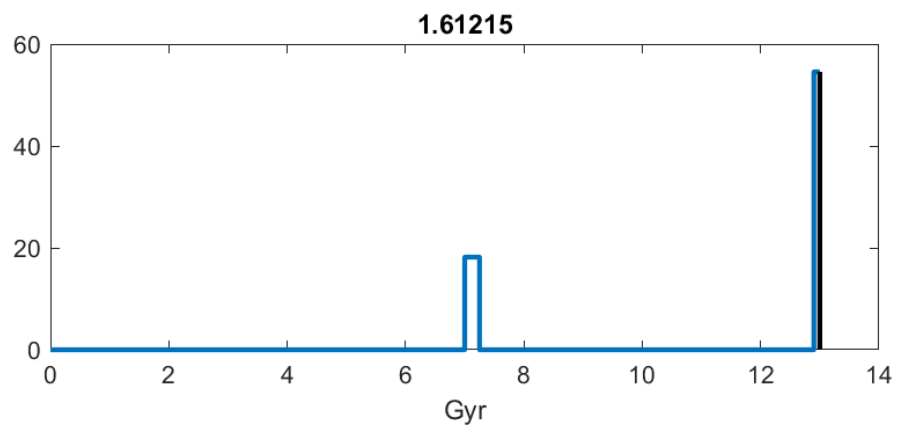
Dos brotes de 0,1 Gyr, uno empezando en 7Gyr y el segundo en 12.9Gyr; siendo el segundo brote el triple de intenso que el primero.



The background features abstract, flowing waves in shades of red, orange, and yellow, creating a sense of movement and energy. The waves are layered, with some appearing more prominent than others, and they curve across the frame.

MUCHAS GRACIAS





~60% del tiempo →

Profiler

File Edit Debug Window Help

Start Profiling Run this code: Profile time: 5743 s

### Profile Summary

Generated 14-Dec-2018 17:39:22 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
<a href="#">Copy_2_of_Main</a>	1	5742.229 s	16.532 s	
<a href="#">CapaConvolucion</a>	19	5135.292 s	1698.318 s	
<a href="#">normpdf</a>	1727810780	3436.974 s	3436.974 s	
<a href="#">Sacar_Eficacia</a>	12	2789.113 s	460.597 s	
<a href="#">saveas</a>	24	103.389 s	0.046 s	
<a href="#">print</a>	24	103.277 s	0.014 s	
<a href="#">graphics\private\alternatePrintPath</a>	24	102.884 s	0.032 s	
<a href="#">Heatmap.Heatmap&gt;Heatmap.doUpdate</a>	72	90.243 s	46.165 s	
<a href="#">gra...e\alternatePrintPath&gt;printSetup</a>	24	48.207 s	0.035 s	
<a href="#">...ternatePrintPath&gt;LocalUpdateViewer</a>	24	45.397 s	5.601 s	
<a href="#">mlprintjob&gt;mlprintjob.setPaintDisabled</a>	48	45.189 s	4.425 s	
<a href="#">onCleanup&gt;onCleanup.delete</a>	864	44.545 s	0.022 s	
<a href="#">...tePrintPath&gt;@0printCleanup(pj.pt)</a>	24	44.456 s	0.001 s	
<a href="#">...te\alternatePrintPath&gt;printCleanup</a>	24	44.455 s	0.015 s	
<a href="#">compose</a>	72	43.554 s	43.554 s	
<a href="#">graphics\private\printingGenerateOutput</a>	24	10.171 s	0.074 s	
<a href="#">...teOutput&gt;LocalGenerateRasterOutput</a>	24	7.832 s	0.001 s	
<a href="#">...ut&gt;LocalCallgenerateGraphicsOutput</a>	24	7.831 s	7.799 s	
<a href="#">HeatmapChart.doUpdate</a>	72	4.726 s	0.285 s	
<a href="#">graphics\private\writeRaster</a>	24	2.259 s	0.004 s	
<a href="#">imwrite</a>	24	2.214 s	0.054 s	
<a href="#">clf</a>	24	2.173 s	0.017 s	
<a href="#">imagesci\private\writepng</a>	24	2.145 s	0.016 s	
<a href="#">...inerFactory\$FigurePanelContainerLight (Java method)</a>	48	2.086 s	2.086 s	
<a href="#">imagesci\private\pngwritetec (MEX-file)</a>	24	2.018 s	2.018 s	