

Bases de datos espacio-temporales multivariadas para el monitoreo del cambio de uso agropecuario del suelo

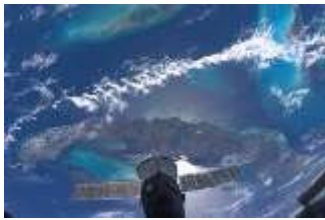
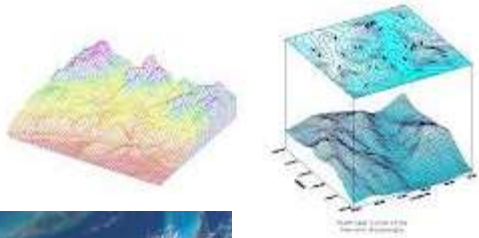
Matías A. Castillo Moine , César Luis
García y Mónica Graciela Balzarini



Datos espaciales

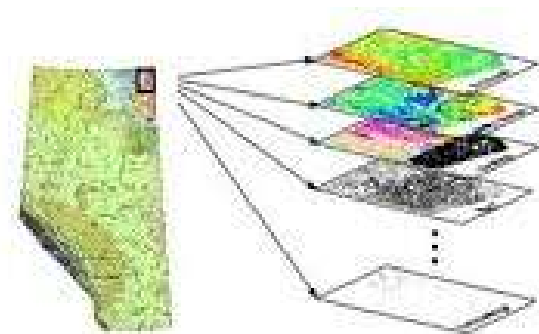


están en todas partes y pueden ser modelados...



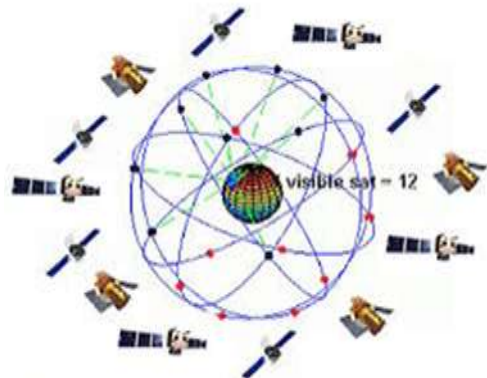
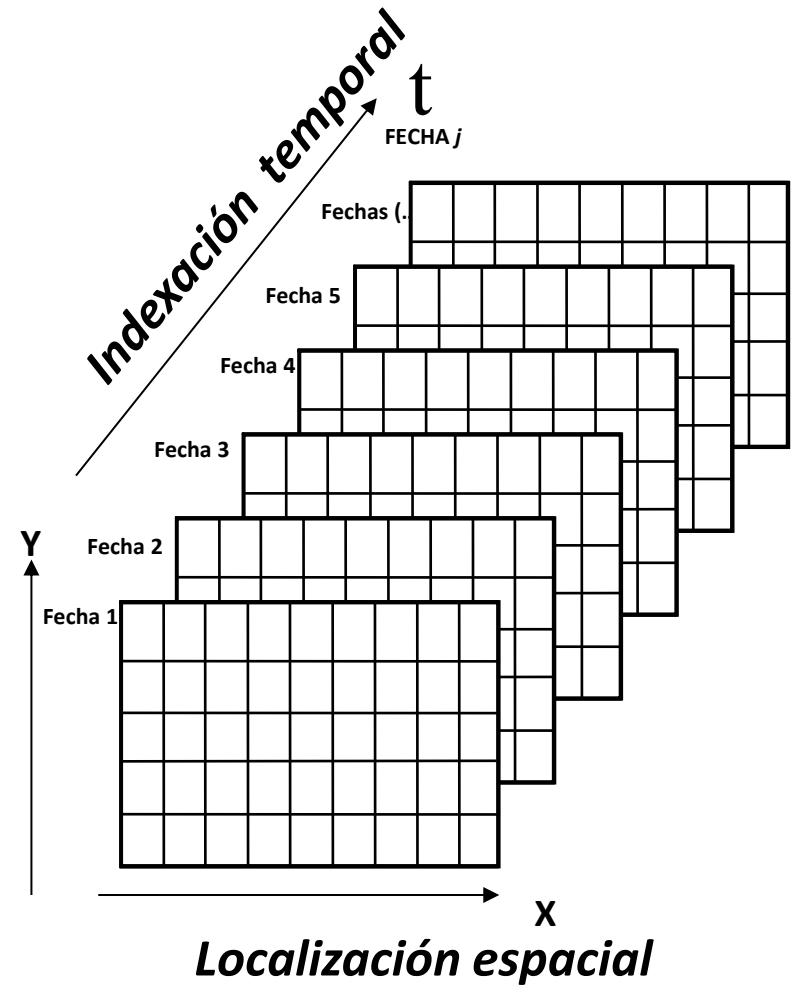
Para cada posición no aleatoria s_i (donde s es el índice espacial definido por las coordenadas x e y) existe un vector aleatorio Z que contiene las z realizaciones de p variables (con $p=1, 2, \dots, p$), es decir que existen p realizaciones

$$z(s_i) = E + e(s_i)$$



Cada realización $Z(s)$ puede estar repetida j veces en el tiempo variables (con $j=1, 2, \dots, j$), es decir que si para cada s_i se registran p variables repetidas j veces en el tiempo, se tiene

$$z(s_{ij}) = E + T + e(s_{ij})$$



Y los temporales también...

Formatos posibles

- Los datos son registrados en todas las s_i localizaciones, pero no en todos los j momentos de tiempo.
- Los datos no son registrados en todas las s_i localizaciones, pero sí en todos los j momentos de tiempo.
- Los datos son registrados en todas las s_i localizaciones, y en todos los j momentos de tiempo.

Objetivo: construcción de bases de datos espacio-temporales multivariadas de datos provenientes de sensores remotos (formato raster)

Formatos de bases de datos

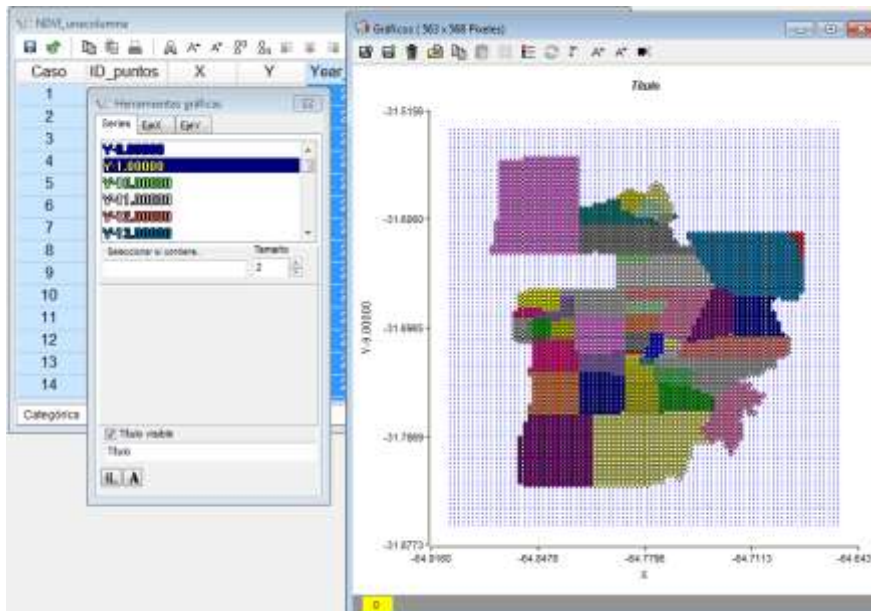
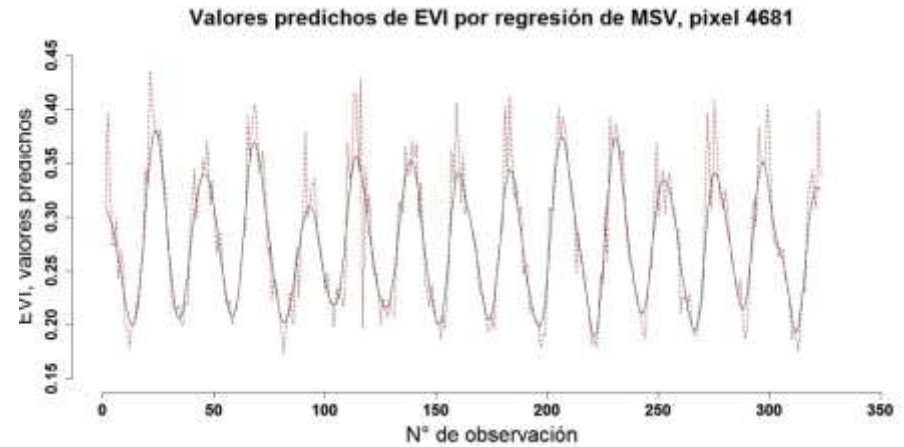
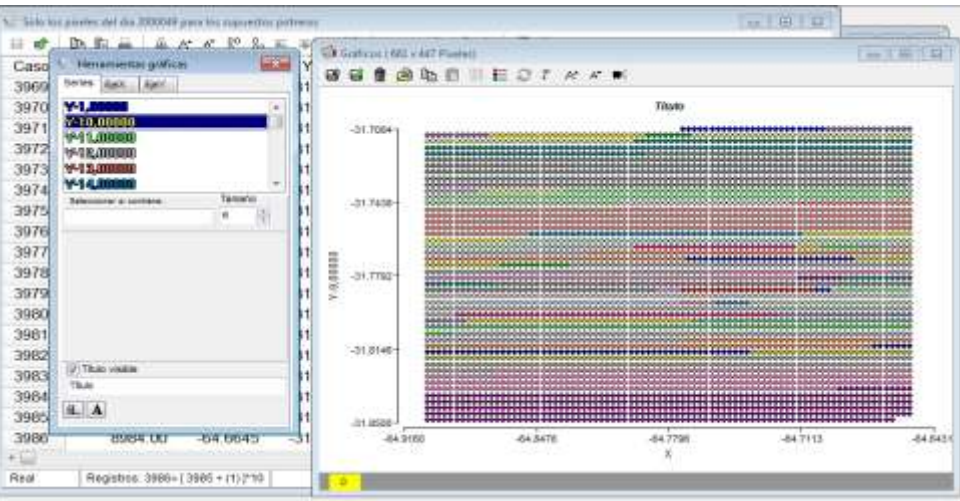
- **Espaciales**: raster, vectorial (puntos, líneas, polígono), tablas xy anchas o largas
- **Temporales**: medidas repetidas, series de tiempo, tablas t anchas o largas
- **Espacio-temporales**: strds, Raster Stacks, Raster Brics, vectoriales con atributos, tablas anchas, tablas largas

Construcción de la base de datos (tabla de las variables)

- Se obtuvieron productos conteniendo estimaciones de variables biofísicas obtenidas de productos satelitales derivados de los sensores MODIS a bordo de los satélites Aqua y Terra.
- Se extrajeron los datos para cada $z(s_j)$ en cada j .
- Se dispusieron los $z(s_{ij})$ en “apilados” espacio-temporales.
 - GRASS: strds (spatio-temporal raster data sets)
 - R: Raster Bricks o Rasters Stacks (Raster*, Big Raster)
- Se compilaron las $z(s_{ij})$ en tablas largas usando los centroides de s_i para obtener de esta manera los vectores $Z(s_i)$.

Espacial

Control de calidad



Temporal

500 m		1000 m
8 días	16 días	8 días
MOD09A1 (reflectancia)	MOD13A1 (IV)	MOD11A2 (LST)
MYD09A1 (reflectancia)	MYD13A1 (IV)	MYD11A2 (LST)
		MOD21A2 (T, emisividad)
		MYD21A2 (T, emisividad)
		MOD14A1 (focos calor)
		MYD14A1 (focos calor)
		MCD15 A2 (LAI, FPAR)
		MOD17A2 (PPN)
		MYD17A2 (PPN)

Un tema de resoluciones...

Unificando la resolución espacial:

Se tomó como criterio la unificación a la menor resolución espacial común posible (en este caso 500 m) para lo cual se usaron funciones de la librería GDAL en su API de Python para extraer de los datasets sólo la región de interés, usando las coordenadas extremas:

- Definir sistema de coordenadas de referencia → Se usó sistema proyectado
- Definir resolución deseada y método de interpolación

Usar software GIS en esta etapa, desactivando reproyección al vuelo, ayuda a identificar errores topológicos espaciales...

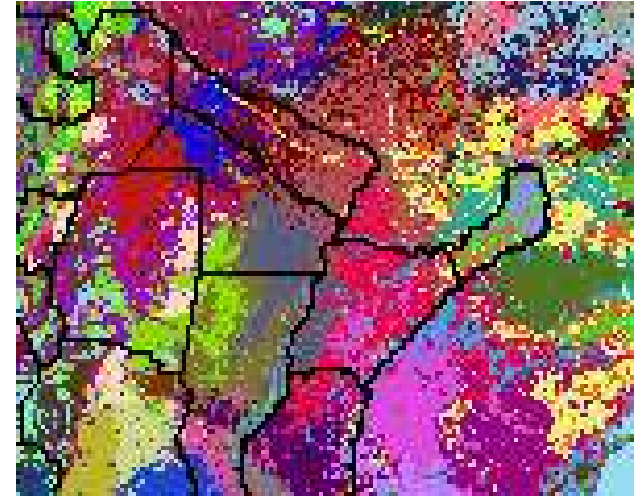
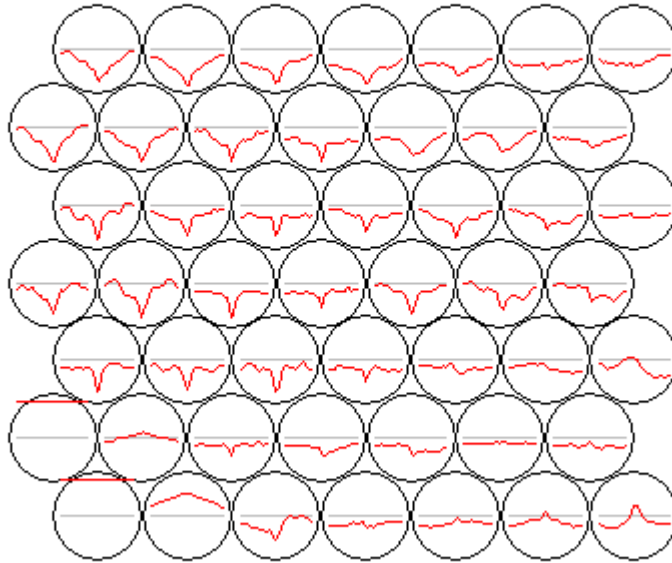
Unificando la resolución temporal:

NDVI viene en productos compuestos cada 16 días, se tomó como criterio generar una serie de 8 días interpolando datos:

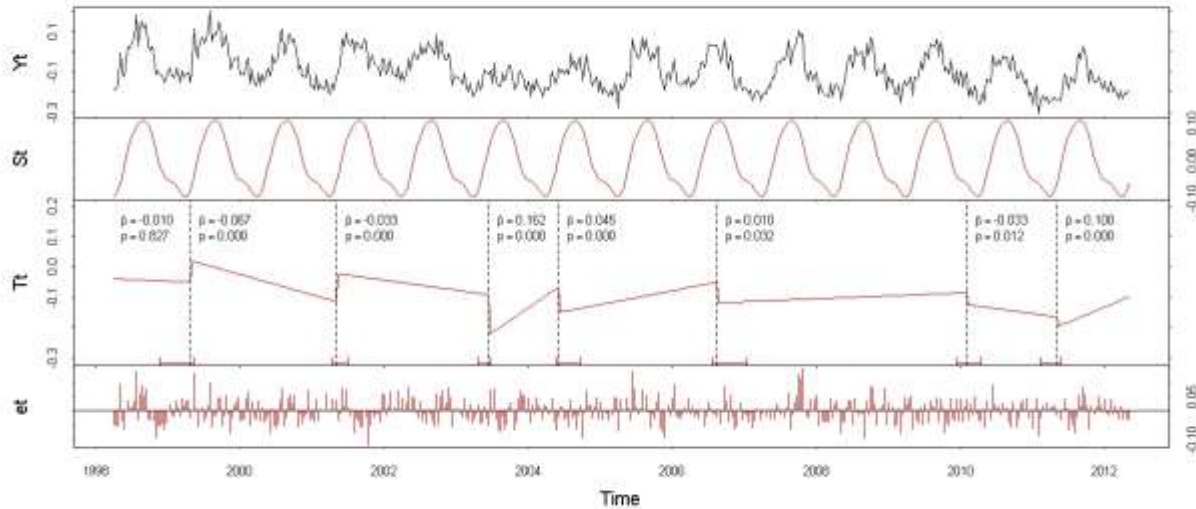
- Definir los s_{ij} que están vacíos
- Rellenar en función de los parámetros temporales: funciones lineales, media móvil, etc.
- Si es necesario, usando la varianza de la serie original de 1 días generar una componente aleatoria (se suma al anterior)
- Otra solución más laboriosa es descargar las imágenes diarias de todas las bandas y crear nuestro propio compuesto de 8 días (hay algunos disponibles en la web...)

Usar software estadístico en esta etapa ayuda a identificar errores topológicos temporales...

SOM correlación cruzada EVI vs. LST dia (7x7)



no. iterations to estimate breakpoints: 3



GDAL

Extracción de la variable, definición de SRC y resolución espacial

Funciones/argumentos:
Producto: imagen GTiff

gdalwarp tiene más funciones en línea de comandos que en las API

GRASS

Registrar la base de datos espacio-temporal, control de calidad espacial

Módulo: temporal

Producto: strds, raster editados, mapas

t.connect y t.register arman bases de datos en SQL (muy eficientes)

Paquetes que trabajan por uno:

-raster:
overlay(), calc(),
stack(),
begincluster(),
as.data.frame(),
write.raster()
-sp
-spacetime:
read.tgrass()
-rgdal
-rgrass7 y
rgrass6

R

Manejo simultáneo de todos los strds, compilado en tablas.

Paquetes y funciones:
Producto: tablas largas, rasters y strds

Flujo de trabajo

EXTRA: DÓNDE APRENDER MÁS PARA USAR ESTAS HERRAMIENTAS

- GDAL: usar el help de la API de GDAL en Python para conocer las funciones disponibles, y usar el help de GDAL en C++ para conocer los argumentos de las funciones...
- R: *Applied Spatial Data analysis whith R* (Pebesma, Bivand, Gomez-Rubio, 2015), es el mejor en el que se explican teoría+software haciendo un muy buen recorrido por los mejores paquetes SIG de R tanto espaciales, temporales y espacio-temporales...
- GRASS: la wiki, en especial la de temporal (escrita por Verónica Andreo)...

Muchas gracias!!!