

Visualisation interactive de données

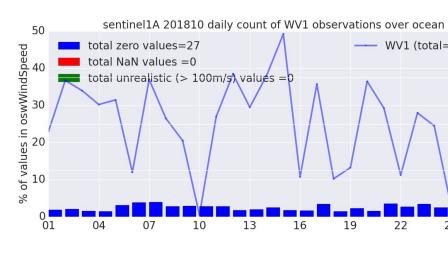
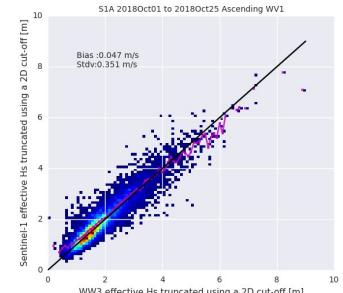
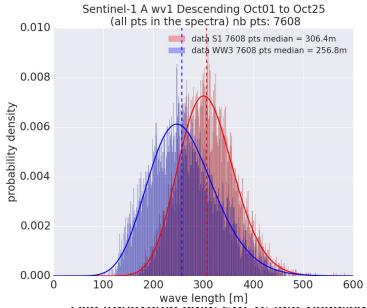
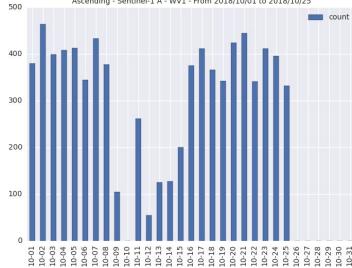
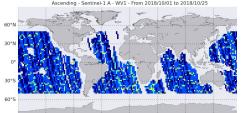
Exposé LOPS axe transverse data

Antoine Grouazel, Olivier Archer, Brendan Coatanéa



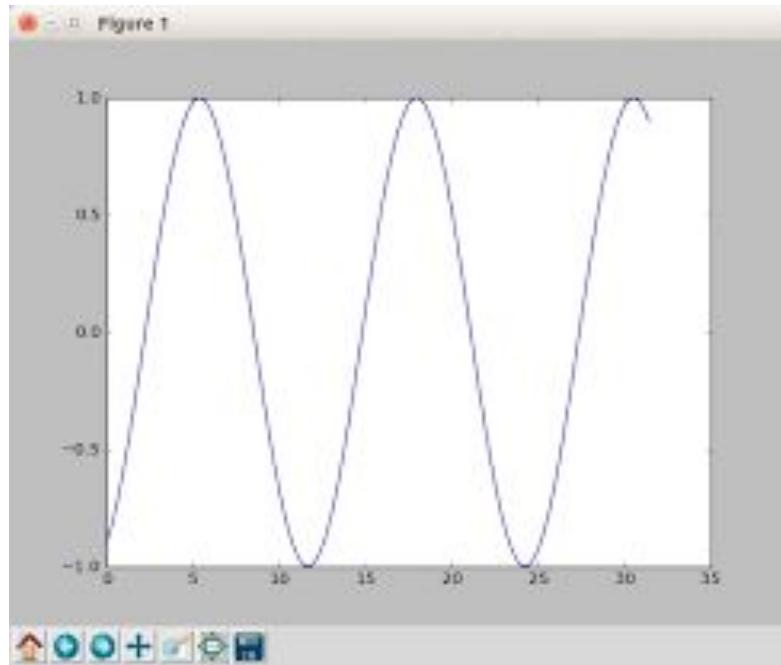
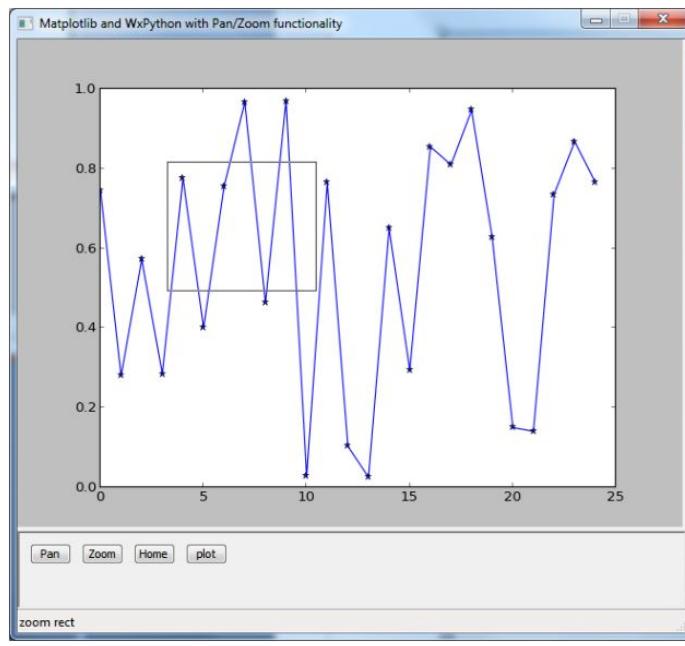
Interactivité: cas d'usage

- Les figures RGB statiques répondent à la grande majorité des analyses.
- Quelques situations pour lesquelles de l'interactivité apporte de l'aide à l'analyse:
 - Recherche de structures petite échelle dans de la haute résolution
 - Affichage d'informations décrites sur N dimensions avec $N > 3$
- Plusieurs fonctionnalités existent pour explorer dans la donnée:
 - Le zoom,
 - la transparence,
 - la superposition d'information,
 - la sélection de points,
 - la synchronisation de plusieurs graph,
 - le déclenchement d'actions *callbacks*





Séparation du rendering (ce qui génère le plot) et canvas/backend (où le plot est affiché).

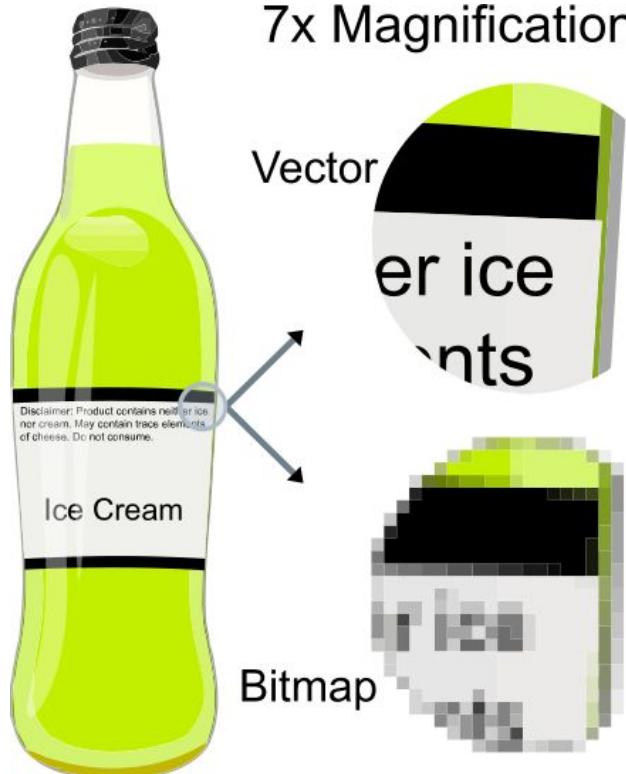


Les backends non interactif -> figure sous forme de fichiers

Renderer	Filetypes	Description
AGG	png	raster graphics – high quality images using the Anti-Grain Geometry engine
PS	ps eps	vector graphics – Postscript output
PDF	pdf	vector graphics – Portable Document Format
SVG	svg	vector graphics – Scalable Vector Graphics
Cairo	png ps pdf svg ...	vector graphics – Cairo graphics
GDK	png jpg tiff ...	raster graphics – the Gimp Drawing Kit Deprecated in 2.0

Pour chaque rendering engine il y a un backend (= interface utilisateur) éponyme.

Matplotlib



2 types de moteurs de rendering:
raster ou vectoriel

Le backend AGG



THE AGG PROJECT

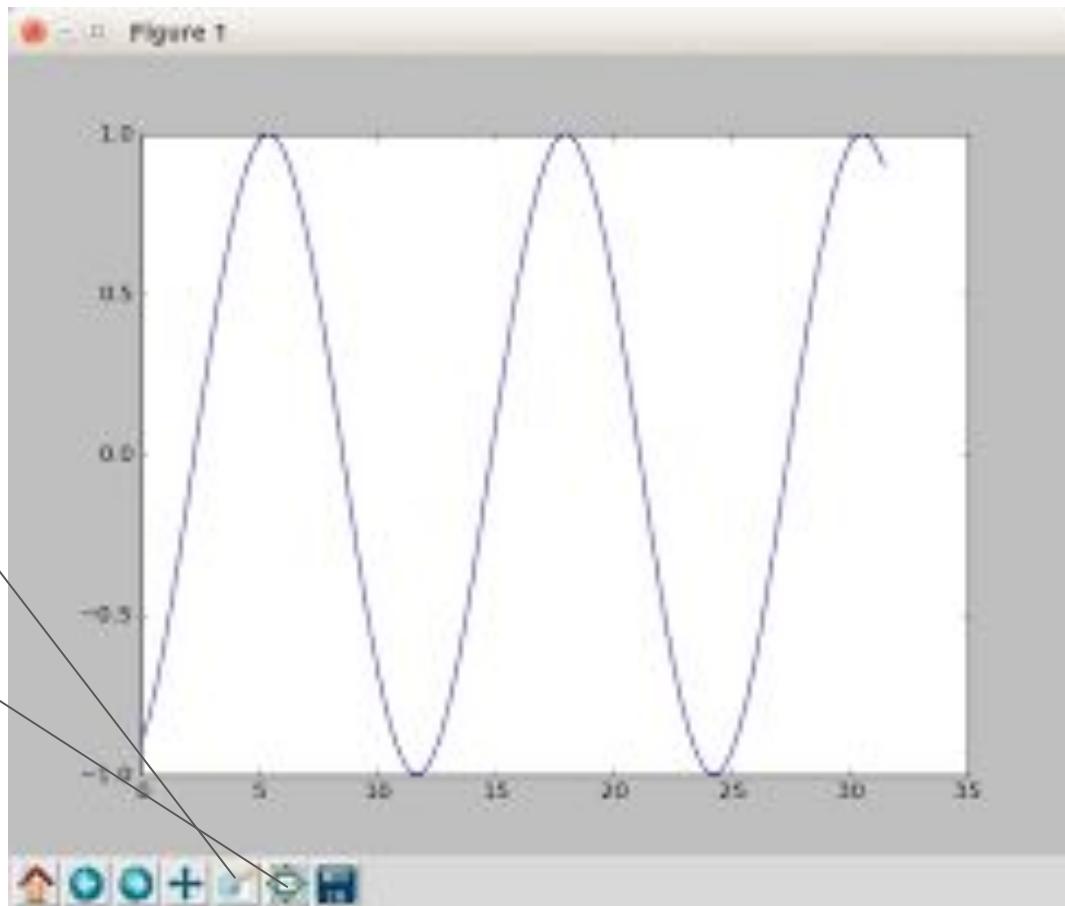
Anti-Grain Geometry

News | Docs | Download | Mailing List | SVN

High Fidelity 2D Graphics
A High Quality Rendering Engine for C++

Matplotlib: les backend interactifs -> interface

Backend	Description
GTKAgg	Agg rendering to a GTK 2.x canvas (requires PyGTK and cairo or cairocffi; Python2 only)
GTK3Agg	Agg rendering to a GTK 3.x canvas (requires PyGObject and cairo or cairocffi)
GTK	GDK rendering to a GTK 2.x canvas (not recommended and deprecated in 2.0) (requires PyGTK and cairo or cairocffi; Python2 only)
GTKCairo	Cairo rendering to a GTK 2.x canvas (requires PyGTK and cairo or cairocffi; Python2 only)
GTK3Cairo	Cairo rendering to a GTK 3.x canvas (requires PyGObject and cairo or cairocffi)
WXAgg	Agg rendering to a wxWidgets canvas (requires wxPython)
WX	Native wxWidgets drawing to a wxWidgets Canvas (not recommended and deprecated in 2.0) (requires wxPython)
TkAgg	Agg rendering to a Tk canvas (requires TkInter)
Qt4Agg	Agg rendering to a Qt4 canvas (requires PyQt4 or pyside)
Qt5Agg	Agg rendering in a Qt5 canvas (requires PyQt5)
macosx	Cocoa rendering in OSX windows (presently lacks blocking show() behavior when matplotlib is in non-interactive mode)



zoom

select

Matplotlib interactive mode

Attention sert uniquement à changer le comportement session <-> figure

```
matplotlib.is_interactive()
```

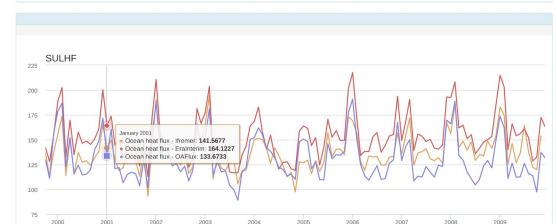
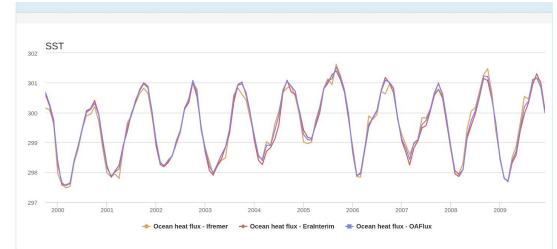
```
plt.ion() -> MAJ de la figure lorsque de nouvelle ligne de code sont exécutées après le premier plt.show()
```

Interactivité -> le (mauvais?) réflexe web

- Le développement web demande de l'expertise
- Architecture web lourde : (contrôleur/views/modèle + DMZ + frontend/backend)
- Durée de vie d'un front-end extrêmement courte (généricité/spécialisation)

Une solution de rationalisation du temps de dev:

Notebook + lib de visu interactive



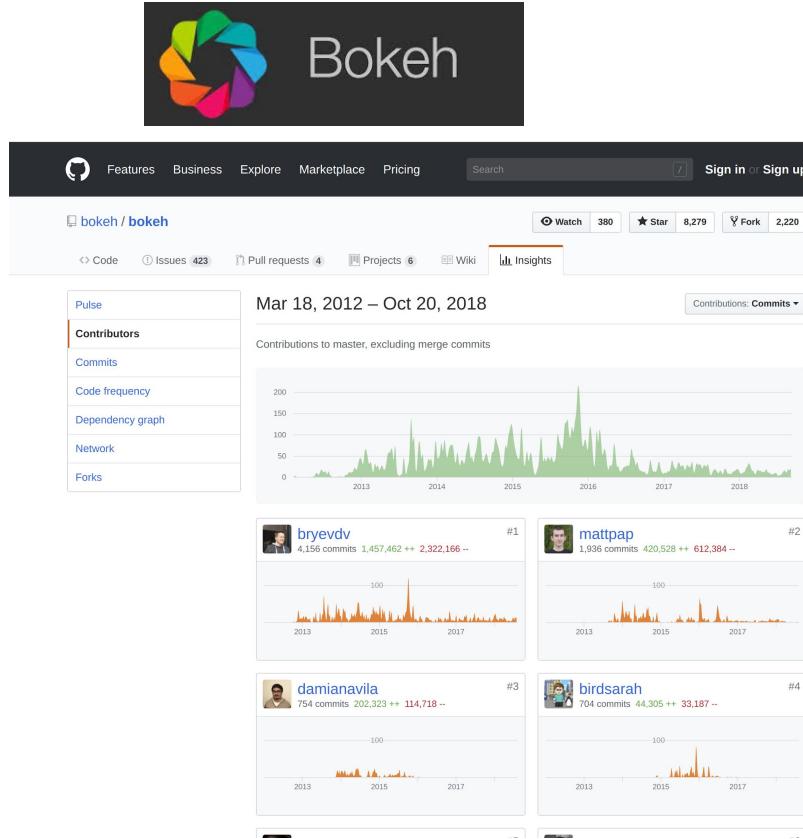
Bokeh library

Created in 2012 funded by DARPA US program.

Python API to render interactive figures.

principe : convertisseur de code python en code javascript exécuté côté client avec figures en .JSON

Grandes similitudes avec matplotlib en terme d'API.





Bokeh:

```
from bokeh.plotting import figure, output_file, show

# prepare some data
x = [1, 2, 3, 4, 5]
y = [6, 7, 2, 4, 5]

# output to static HTML file
output_file("lines.html")

# create a new plot with a title and axis labels
p = figure(title="simple line example", x_axis_label='x', y_axis_label='y')

# add a line renderer with legend and line thickness
p.line(x, y, legend="Temp.", line_width=2)

# show the results
show(p)
```





3 façons d'utiliser bokeh

	intérêts	défauts
Bokeh -> html	Partage de fichiers auto-suffisant (librairies/data)	Output très volumineux, fonctionnalités parfois altérées.
Bokeh in notebook	Préparation ou récupération de la donnée après sous échantillonnage ou appel à des méthodes complexes à la demande	Partage difficile. Dépendances aux versions des navigateurs et des notebooks.
Bokeh server	Facilitation des interactions via une IHM facile à créer.	Improvisation de calculs sur une sélection de données impossible.



Notion de ColumnDataSource

Objet python permettant de stocker les données à afficher dans Bokeh

Compatibilité avec pandas DataFrame

méthodes haut niveau de filtre/édition en slice/groupby ...

```
class ColumnDataSource(*args, **kw)
```

Notion de CDSView

Class bokeh permettant de mettre des filtres sur une instance ColumnDataSource

Intérêt: utiliser un même set de donnée en mémoire pour générer des figures sans avoir à stocker une copie du dataset original

```
class CDSView(**kwargs)
```

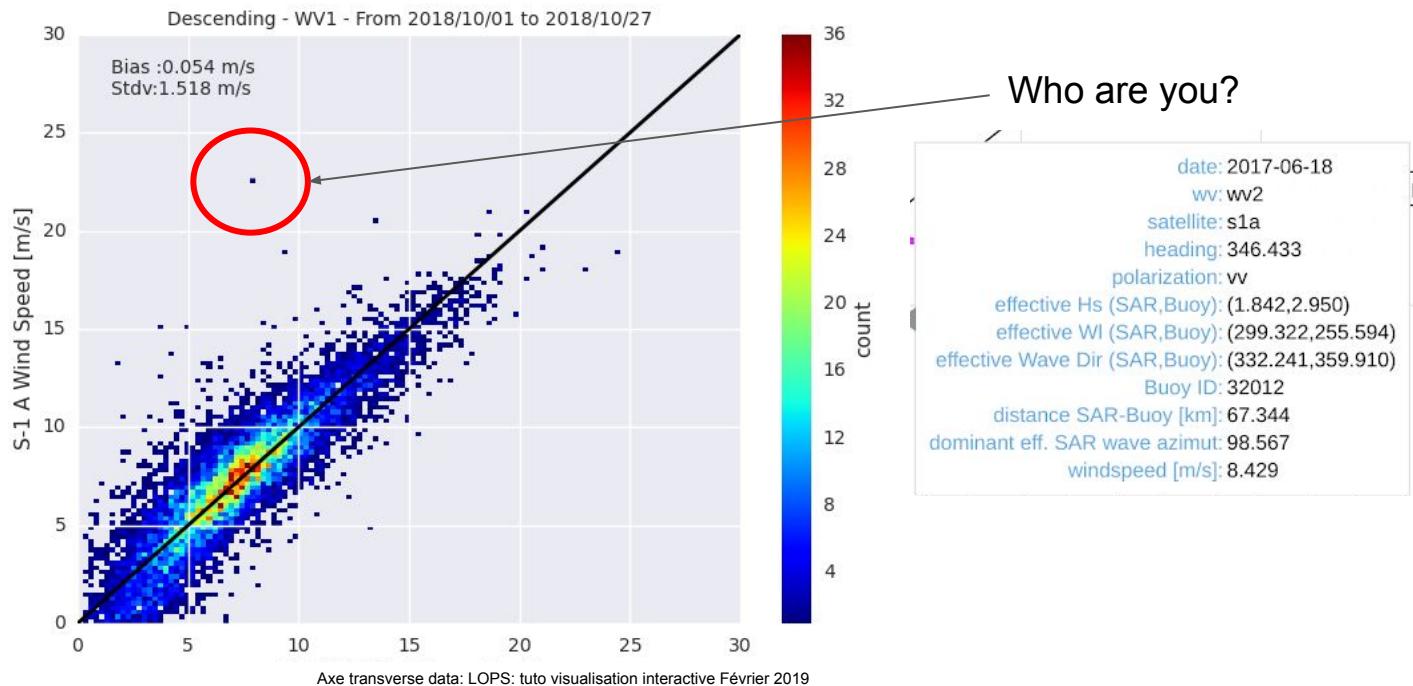


Bokeh



Analyse de données: contextualisation d'un cas d'étude

- Comment faire rentrer 100 variables dans un plot 2D ?



Ici code bokeh pour faire une figure + mouse hover

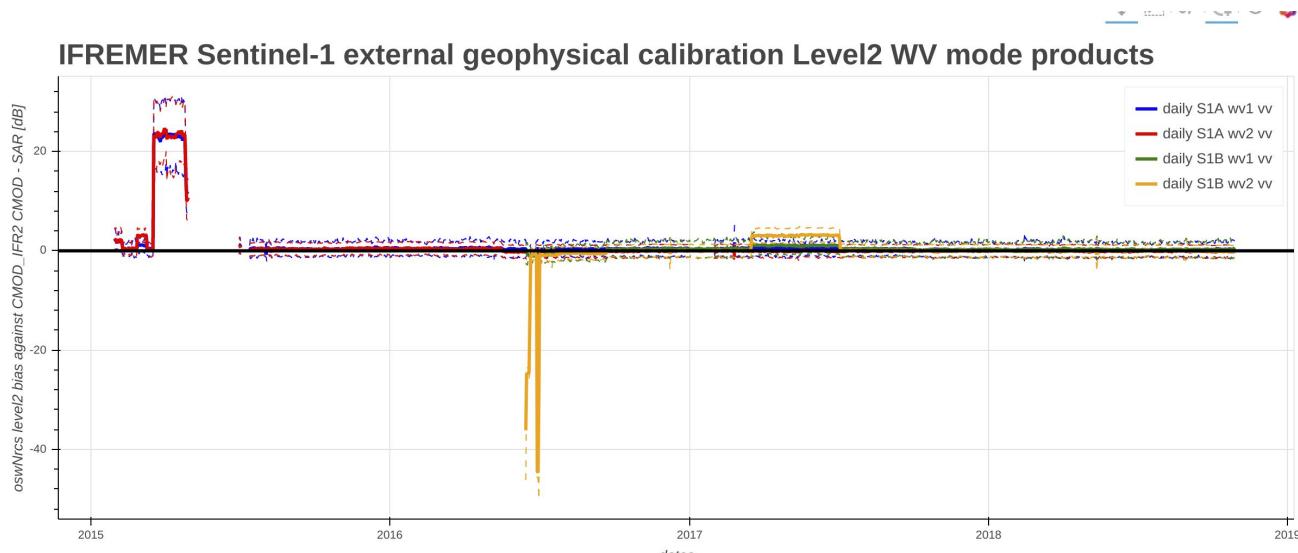
```
hover = HoverTool(tooltips=[  
    #("index", "$index"),  
    #("(x,y)", "($x, $y)'),  
    ("date", "@FDATEDT{%F}"),  
    ("wv", "@WV_MODE"),  
    ("satellite", "@sat"),  
    ('heading', '@OSWHEADING'),  
    ('polarization', '@POL'),  
    ('effective Hs (SAR,Buoy)', '(@hs_sar_partition_bary_xa,@hs_buoy_partition_bary_xa)'),  
    ('effective Wl (SAR,Buoy)', '(@wl_eff_sar_xa_bary,@wl_eff_buoy_xa_bary)'),  
    ('effective Wave Dir (SAR,Buoy)', '(@dirdeg_eff_sar_xa_bary,@dirdeg_eff_buoy_xa_bary)'),  
    ("Buoy ID", "@buoy_name"),  
    ('distance SAR-Buoy [km]', "@distancebuoy"),  
    ('dominant eff. SAR wave azimut', "@dominant_SAR_wave_azimut"),  
    ('windspeed [m/s]', '@OSWWINDSPEED'),  
    #('link_spectra', "@url_spectra") #ca marche mais cest trop long  
], formatters={'FDATEDT': 'datetime'})
```

Analyses de données: figures hautes résolution

Temporellement :

ftp://ftp.ifremer.fr/ifremer/cersat/users/agrouaze/MPC_sentinel1/S1_time_serie_bokeh_calibration_level2_vv_v4.html

Spatialement





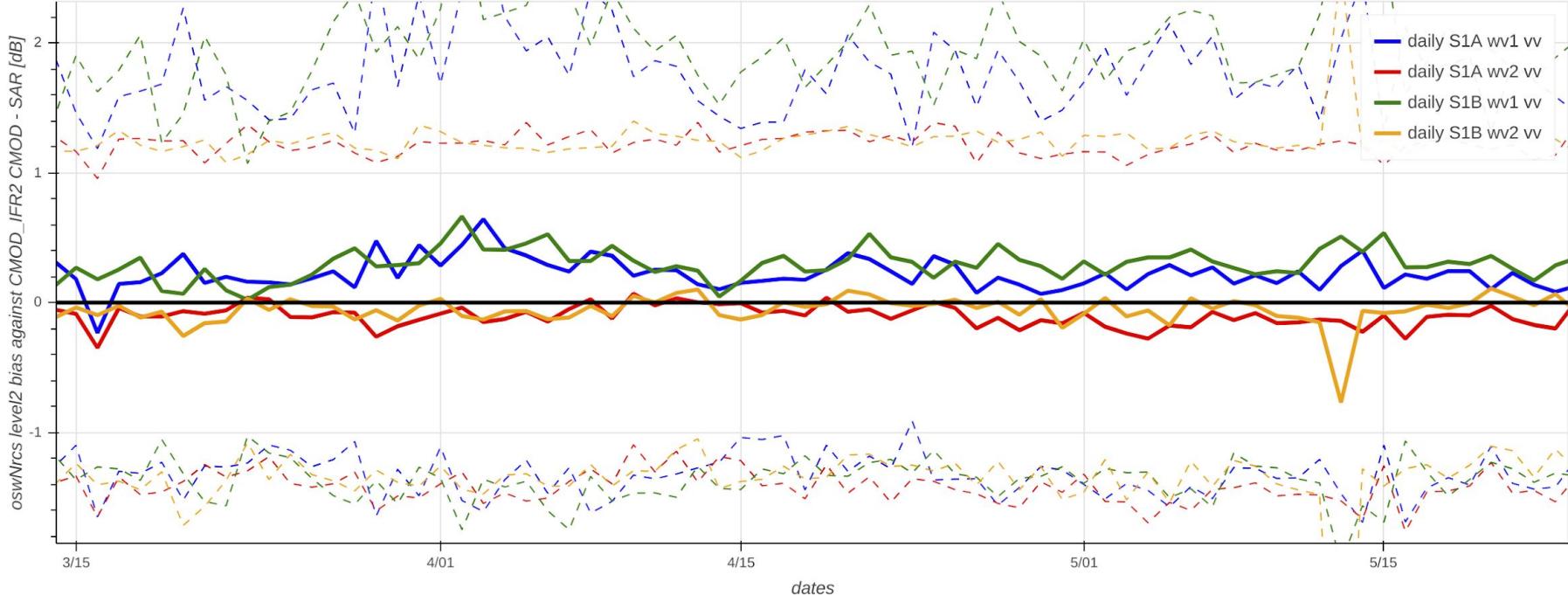
Bokeh



Analyses de données: figures hautes résolution

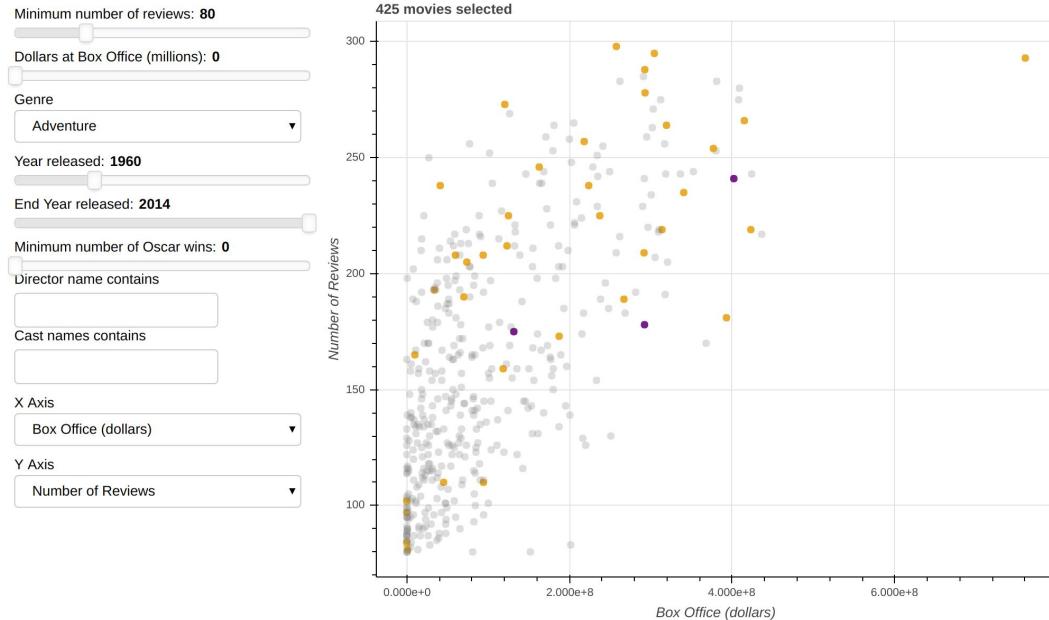


IFREMER Sentinel-1 external geophysical calibration Level2 WV mode products



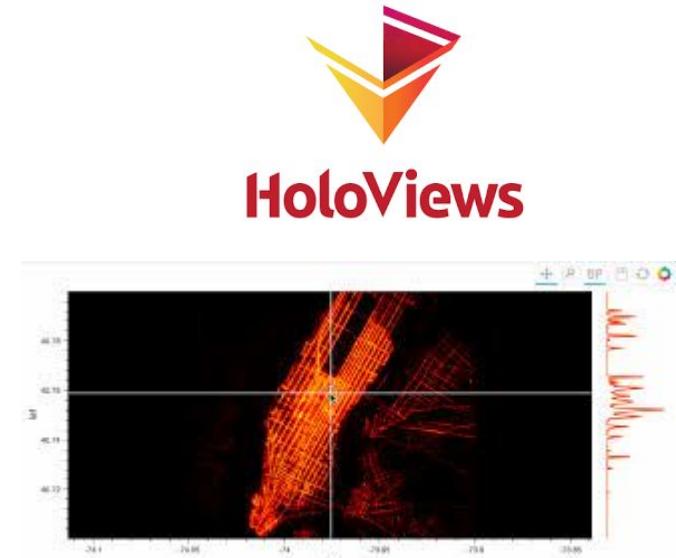
Analyses de données: le subsampling/filtrage

<https://demo.bokehplots.com/apps/movies>



Limitations de bokeh

- Librairie jeune (2012)
 - comptabilité d'une version à une autre extrêmement faible
 - Instabilités fortes avec les changements d'environnement
- Moteur des rendus javascript, mixe des callback python -> code compliqué
- javascript output is disabled in jupyterlab
- Limitation du nombre de points : déportation des calculs/rendus visuels côté client. D'autres librairies peuvent venir en complément: <http://holoviews.org/> ou <http://databricks.org/>



https://gitlab.ifremer.fr/ag0601d/tp_visu_interactive.git

Datashader (2014)



Datashader

Fonctionnement:

- Clipping: élimination des pixels cachés dépendant d'un niveau de zoom
 - Décimation: réduction dans la distribution des points en préservant les formes
- Parallélisation du rendu par pixel: code compilé numba + dask (multi core)

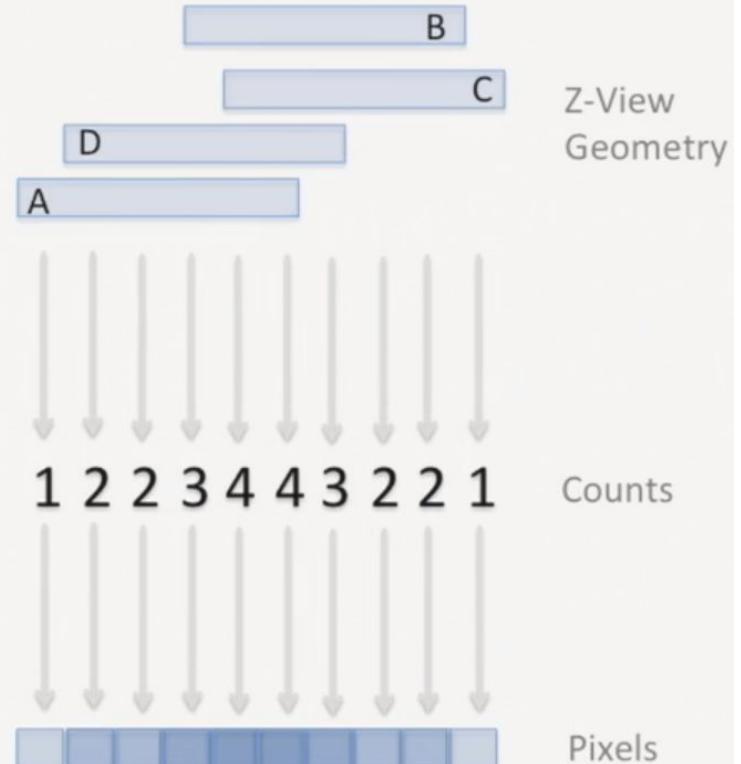
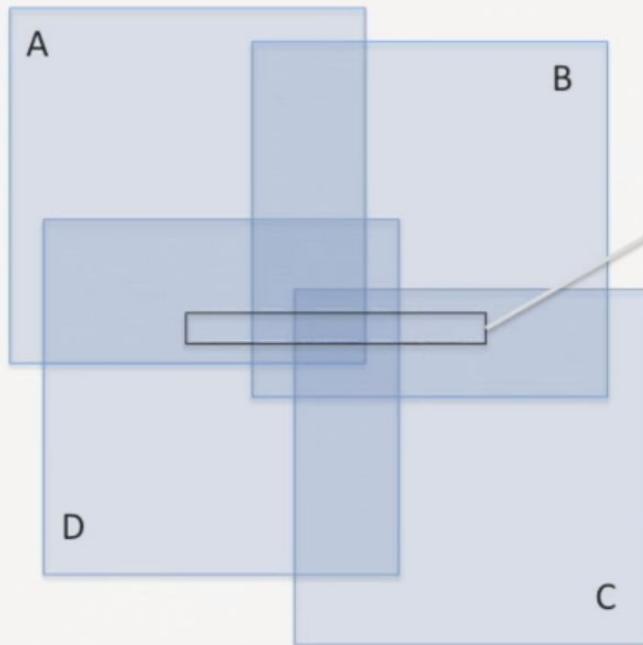
Joseph A. Cottam^a and Andrew Lumsdaine^a and Peter Wang^b

^aCREST/Indiana University, Bloomington, IN, USA;

^bContinuum Analytics, Austin, TX, USA

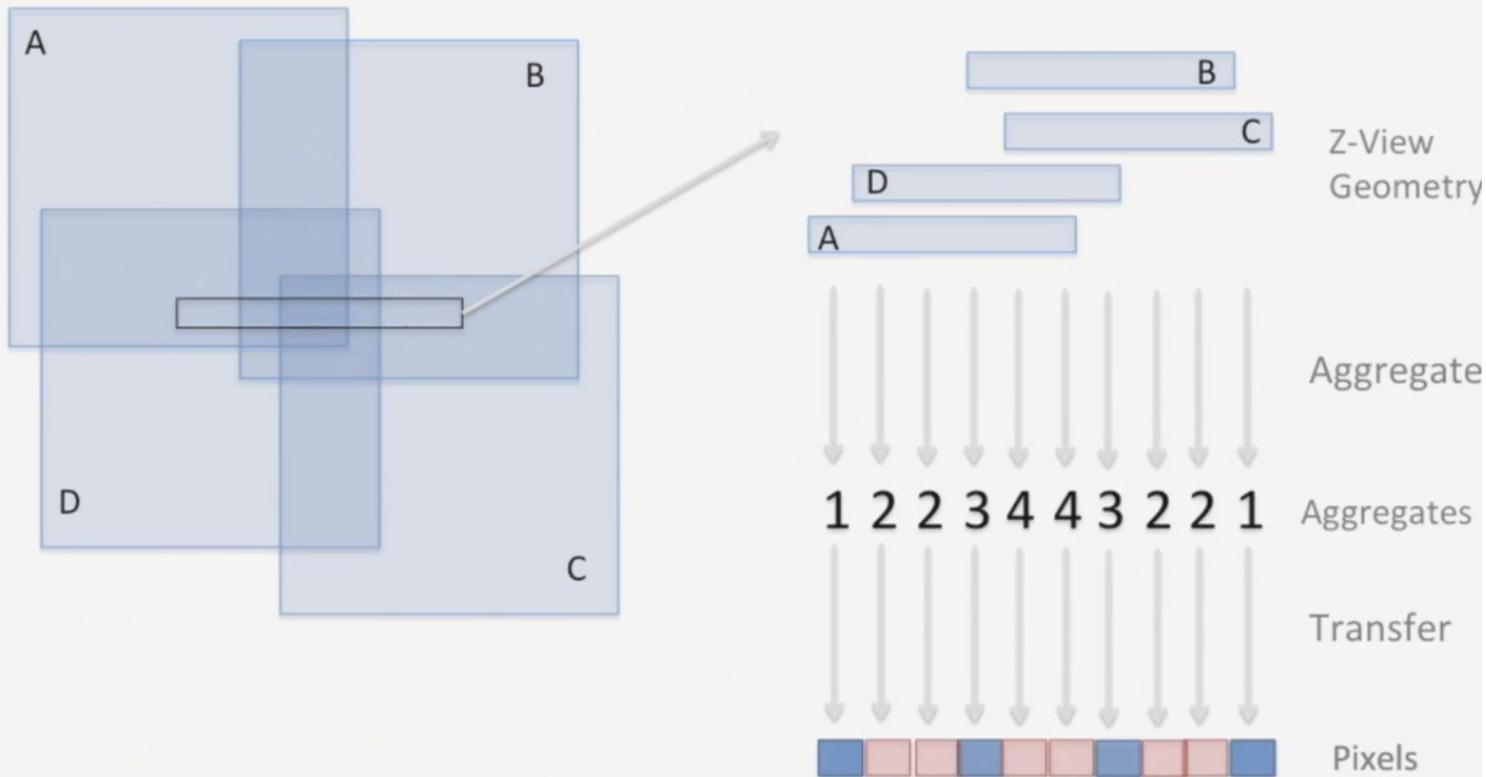
Pixels are Bins

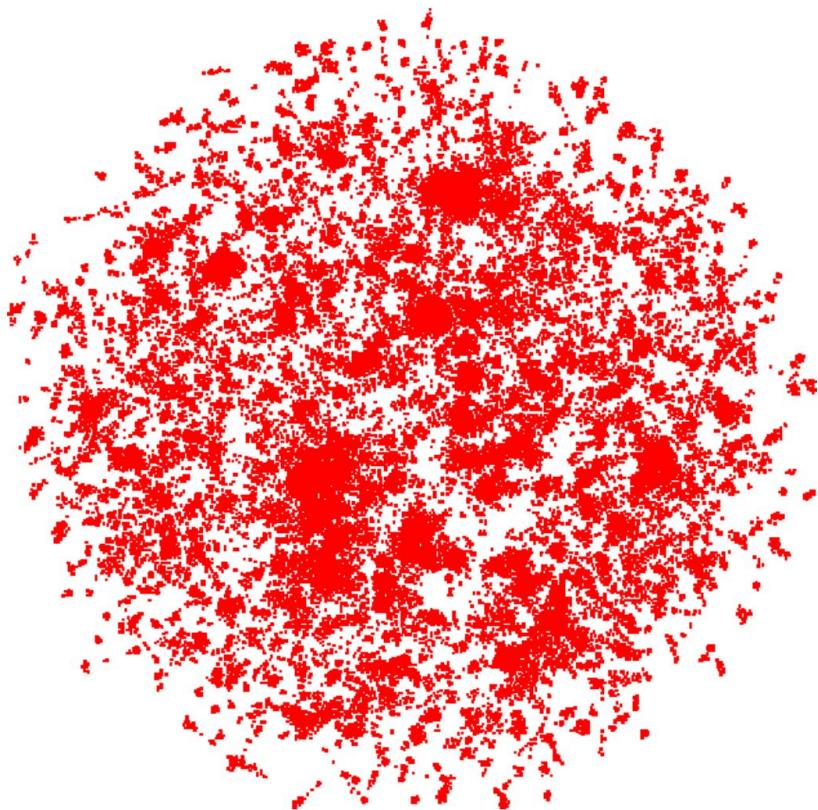
and alpha composition is aggregation



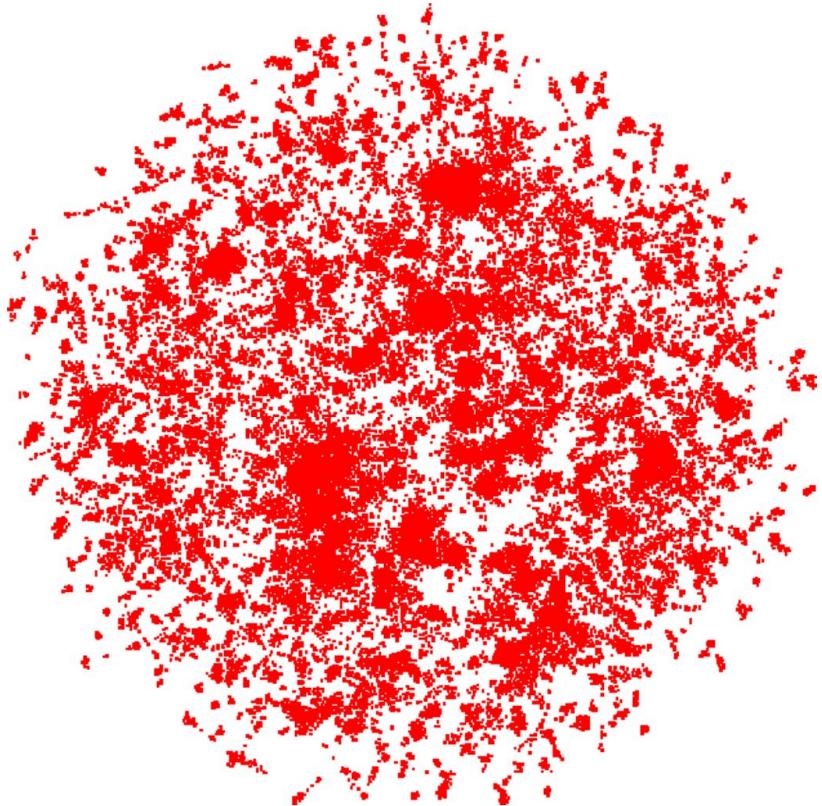
Data Shading

Rendering data instead of colors (at first)

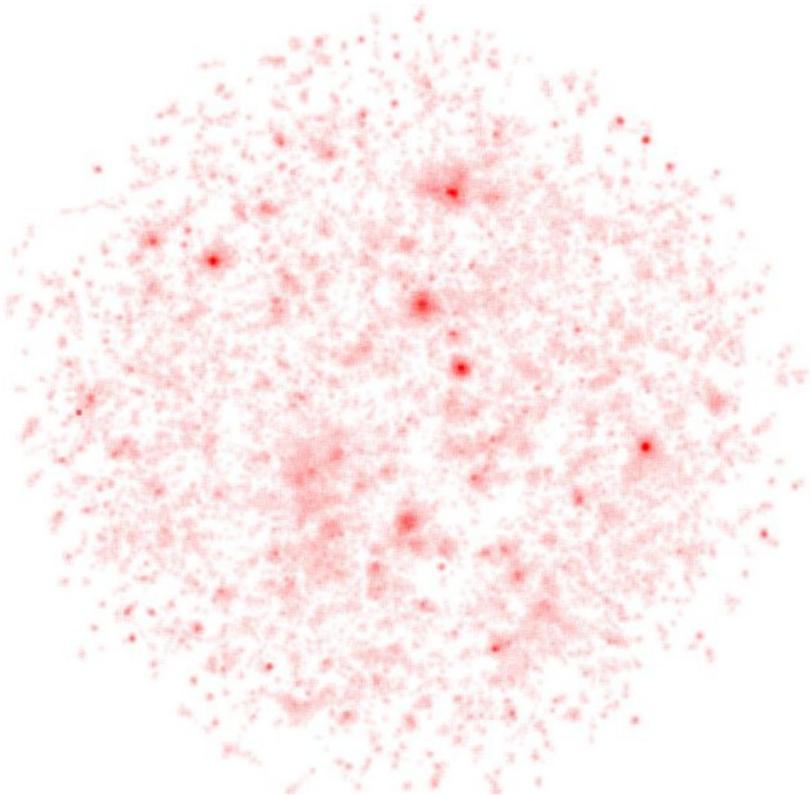




(a) Full Saturation Nodes



(a) Full Saturation Nodes



(b) 10% Alpha Per-Node

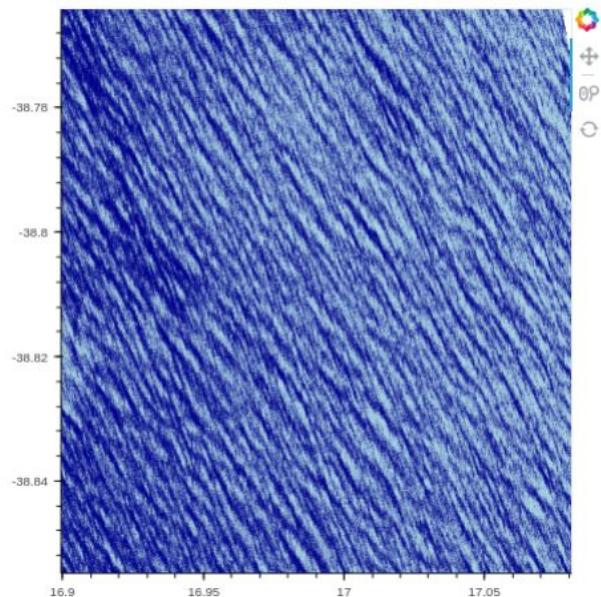
Datashader test on S-1 WV surface roughness

```
In [79]: 1 #adapt to s1 roughness
2 def image_callback_s1(x_range, y_range, w, h, name=None):
3     cvs = ds.Canvas(plot_width=w, plot_height=h, x_range=x_range, y_range=y_range)
4     agg = cvs.points(df_s1, 'lo', 'la', ds.mean('s0'))
5     img = tf.shade(agg)
6     return tf.dynspread(img, threshold=0.10, name='toto')
7 bp.output_notebook()
8 p = bp.figure(tools='pan,wheel_zoom,reset', x_range=(-180,180), y_range=(-90,90), plot_width=500, plot_height=500)
9 print(len(df_s1))
10 InteractiveImage(p, image_callback_s1)
```

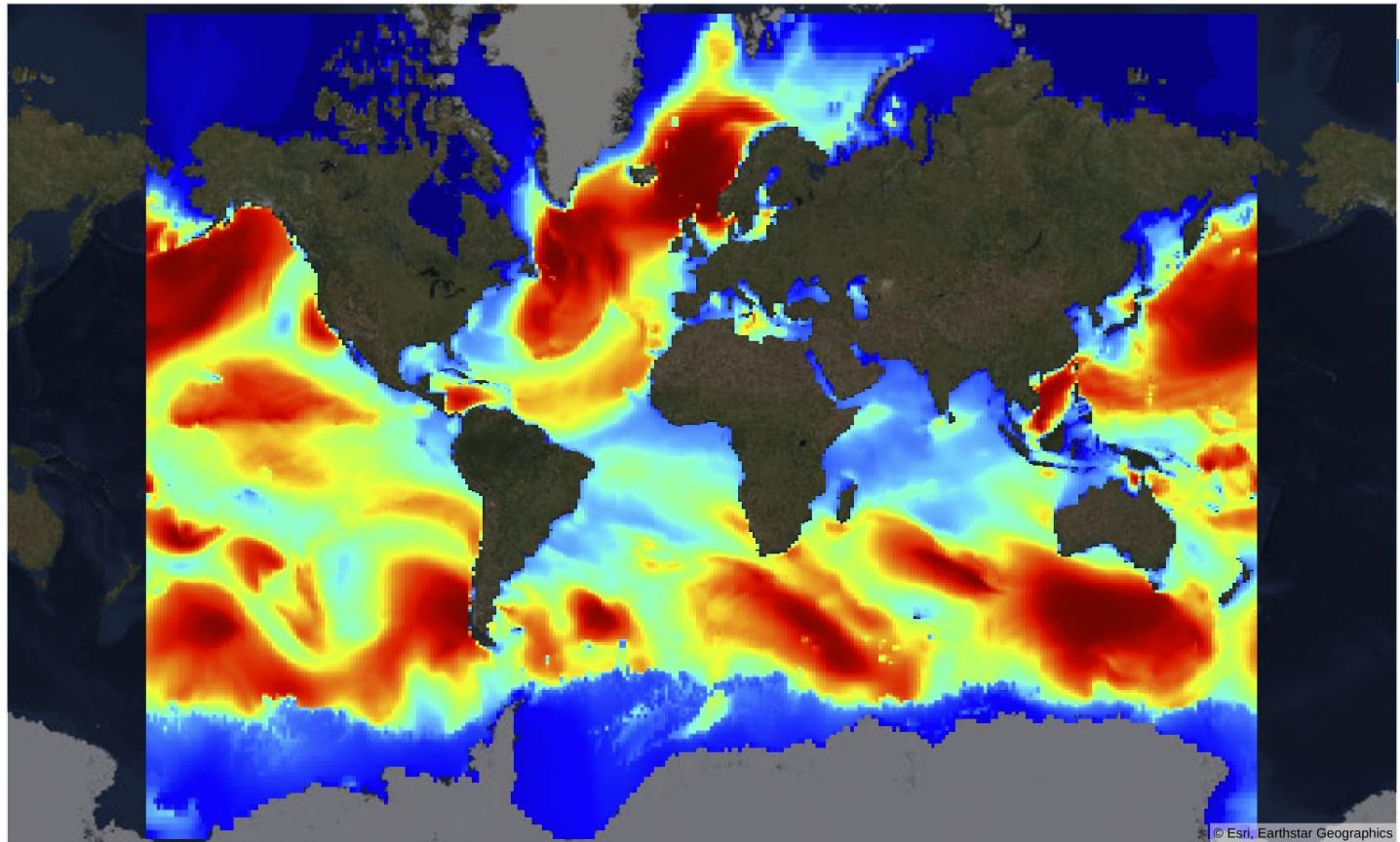
BokehJS 1.0.4 successfully loaded.

5216192

Out[79]:



Out[49]:



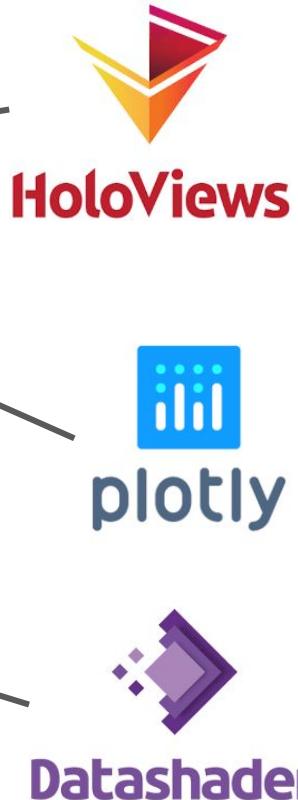
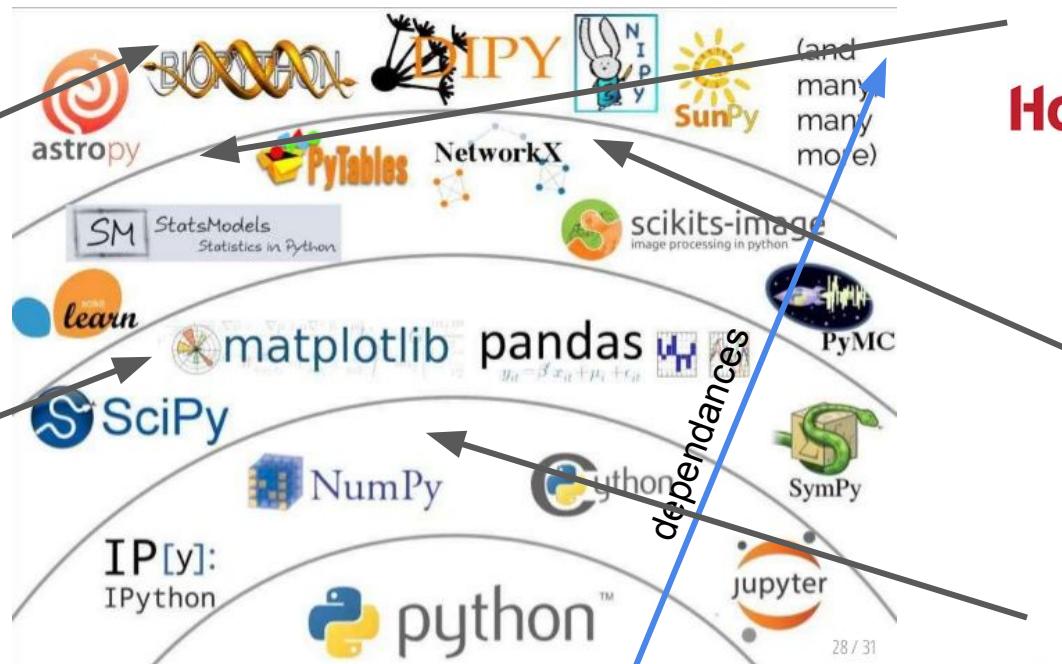
Les librairies graphiques python



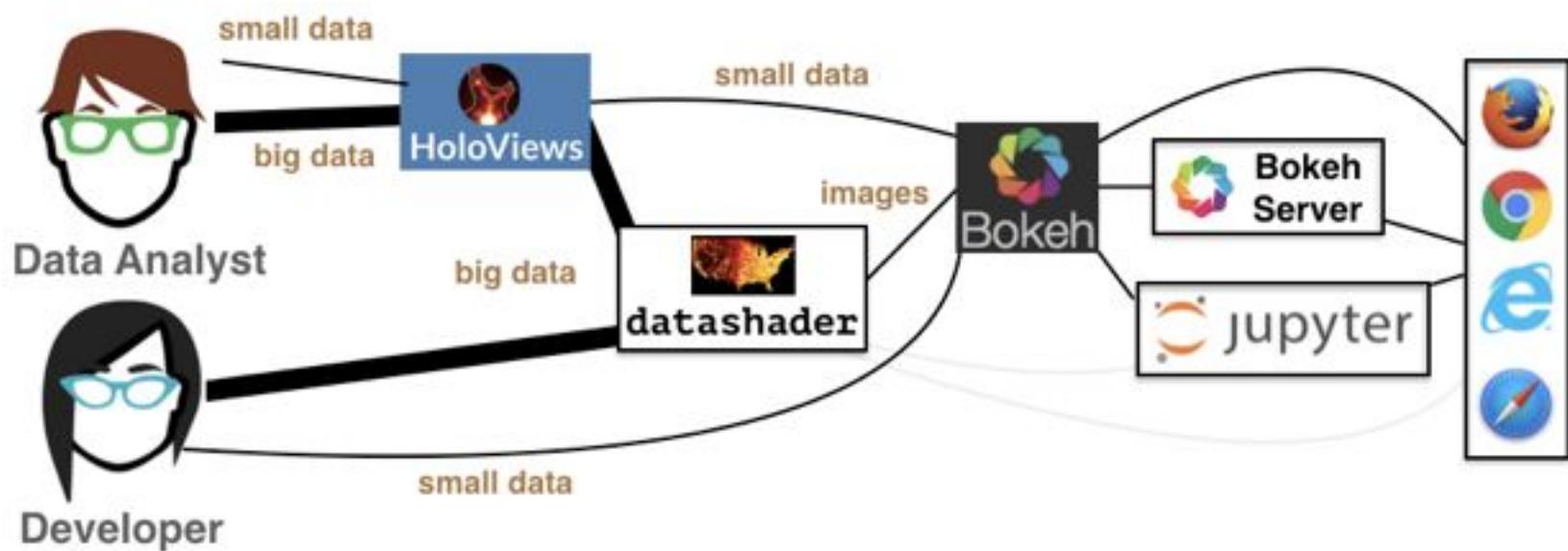
hvPlot



Bokeh



Interfaces holoviews-dashshader-bokeh



Figures interactives dans vos publications?

Quelques journaux qui ont introduit des figures interactives dans leur article numérique

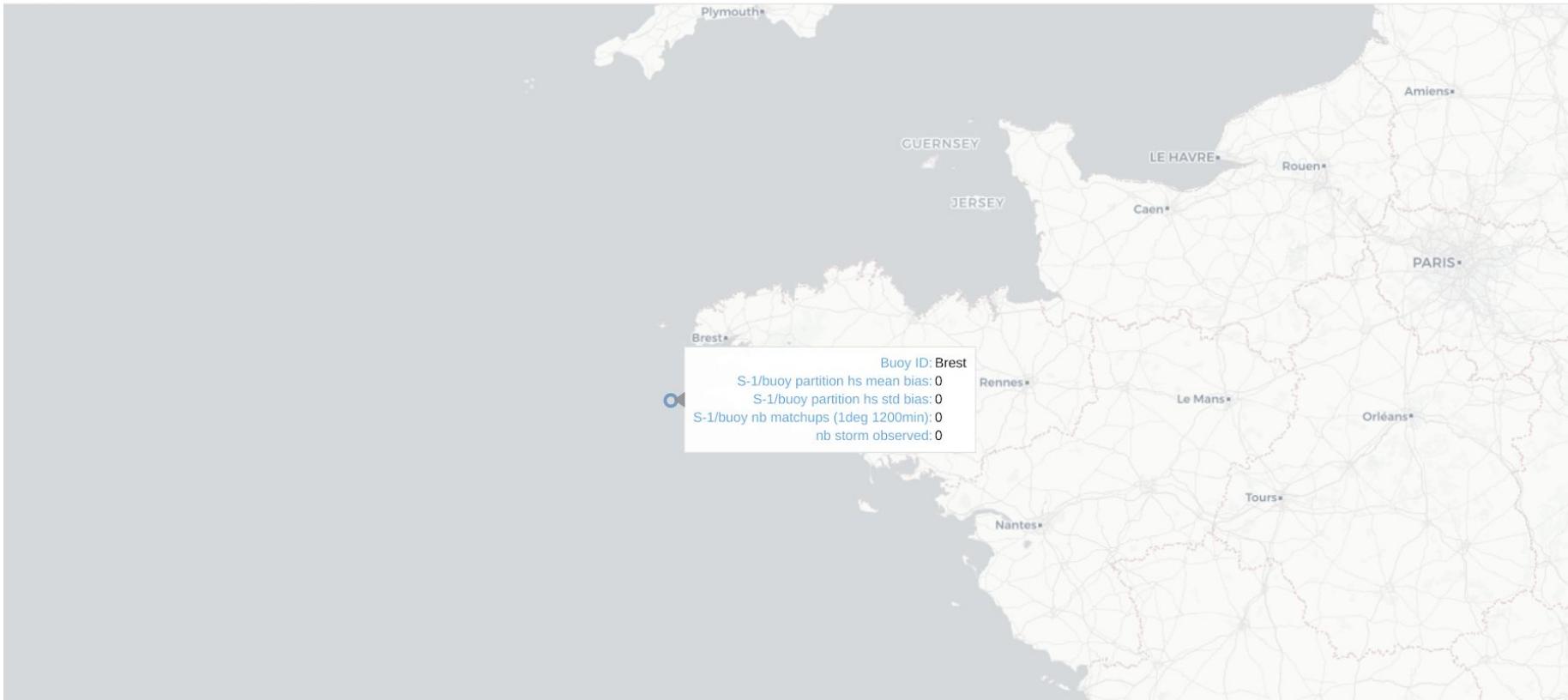


Partie démo/TP

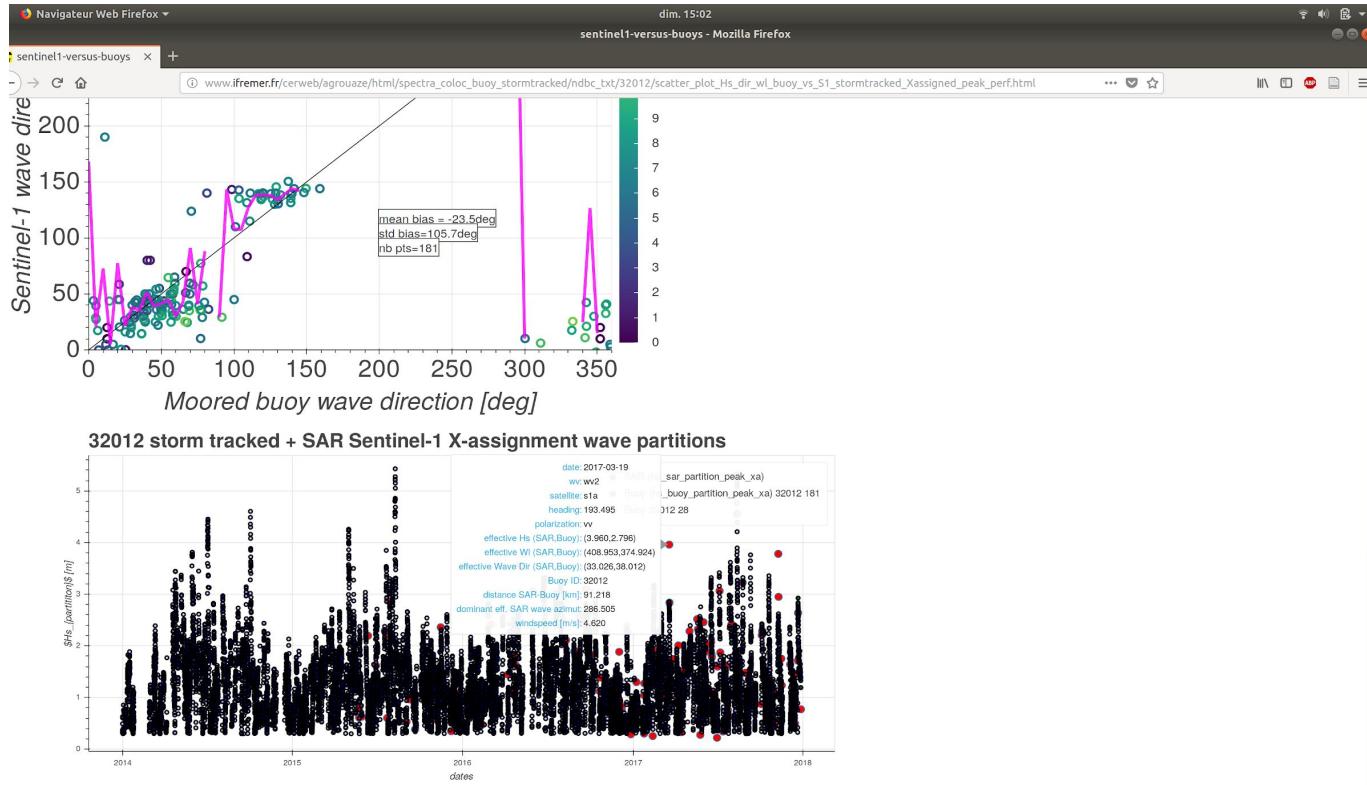
Map buoys processed in storm tracking analysis



http://www.ifremer.fr/cerweb/agrouaze/html/spectra_coloc_buoy_stormtracked/ndbc_txt/map_buoy_vs_S1_stormtracked_Xassigned.html

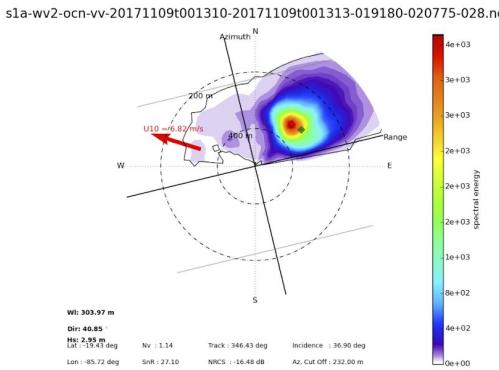


Exemple visualisation des données de bouées

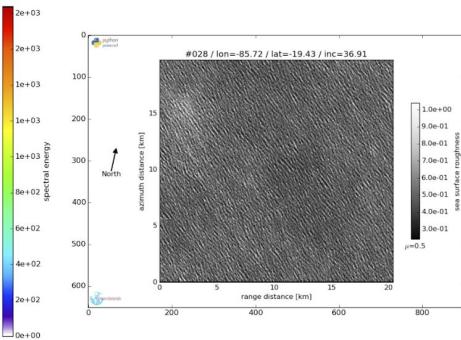
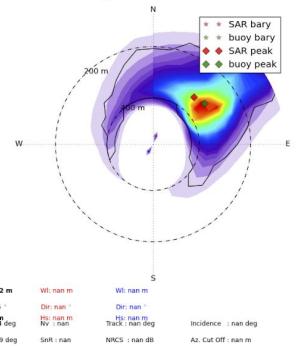


Liens entre fichier html statiques

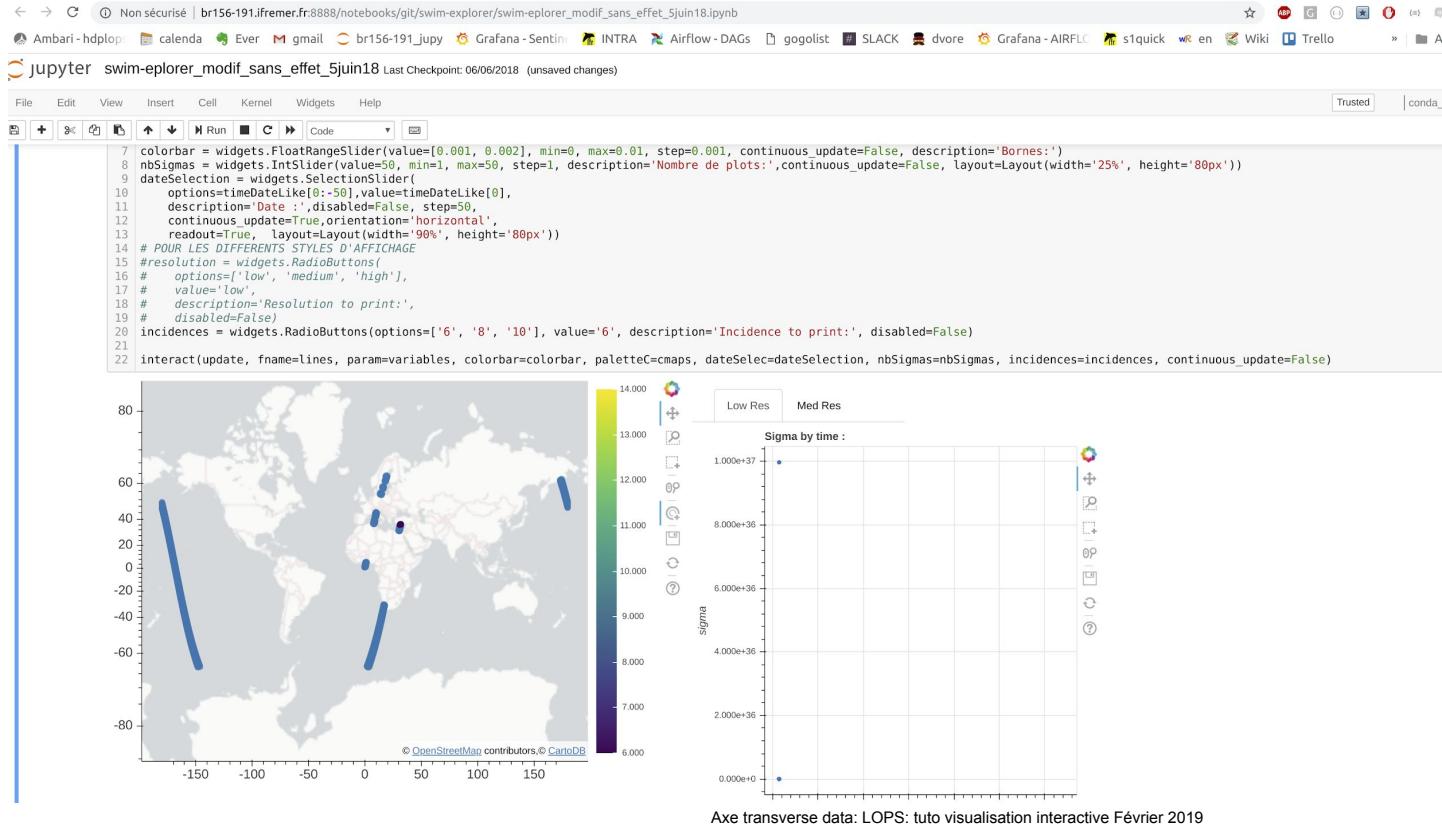
plot associated to this X-assigned pair SAR-Buoy.



3-hours averaged buoy 20171109 00:00:00



CFOSAT SWIM Explorer (en cours de dev)



Cahier des charges:

1. Fichier netCDF sur disque as input
2. Sélection spatio-temporelle facile
3. Subsampling et réutilisation de sélection dans le notebook
4. Partage / portabilité forte

CFOSAT SWIM Explorer (en cours de dev)

Cas d'utilisation: repérage d'une zone d'intérêt, sélection du faisceau SWIM, comparaison des paramètres radar avec les paramètres en sortie de l'inversion de vague, comparaison avec des produits ancillaires.

Difficultés: lenteurs, taille des fichiers, partage git/ipynb

TP#0 prise en main bokeh

Connexion domicile ou Ethernet réseau intra IFREMER

ssh datarmor

git clone https://gitlab.ifremer.fr/ag0601d/tp_visu_interactive.git

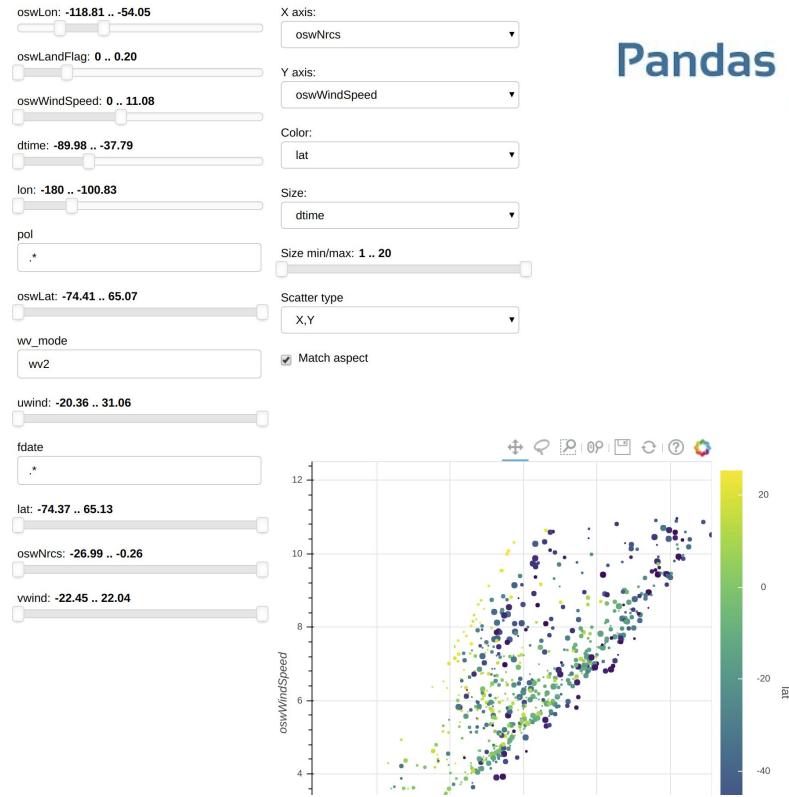
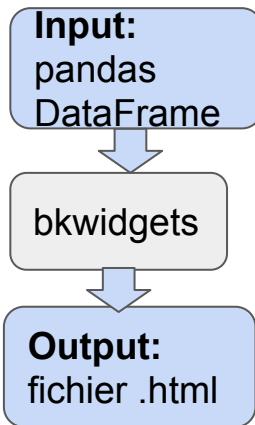
Connexion sur jupyterhub datarmor: <https://datarmor-jupyterhub.ifremer.fr>

|  [bokeh_very_first_steps.ipynb](#)

Exemple d'utilisation de bokeh: Bkwidgets



Principe:



Intérêt: économiser du temps de développement sur des plots classique QQ scatter plot

Sub-sampling à la volée

TP#1 bkwidget df -> html

Connexion domicile ou Ethernet réseau intra IFREMER

```
git clone https://gitlab.ifremer.fr/ag0601d/tp\_visu\_interactive.git
```

```
bash
```

```
source /appli/anaconda/2.7/bin/activate lopsformationdata
```

```
cd
```

```
python sentinel1_test_bkwidets.py
```

```
En local scp datarmor.ifremer.fr:sentinel1_wv.html /mon/path/local
```

Ouvrir le fichier html dans un navigateur web

TP#2 Bkwidget: notebook

Connexion domicile ou Ethernet réseau intra IFREMER

#ssh datarmor:

#copie du notebook:

```
git clone https://gitlab.ifremer.fr/ag0601d/tp\_visu\_interactive.git
```

Connexion sur jupyterhub datarmor: <https://datarmor-jupyterhub.ifremer.fr>

#copie de la data? Ou alors je là met sur le réseau datarmor

/home/datawork-lops-siam/project/sentinel1/data/tmp_data/s1_small_df.pkl

http://www.ifremer.fr/cerweb/agrouaze/static/s1_small_df.pkl

ouvrir le notebook test_bkwidget.ipynb et choisir le kernel

Python [conda env:lopsformationdata]

TP#3 Datashader

Git clone https://gitlab.ifremer.fr/ag0601d/tp_visu_interactive.git

Connexion sur jupyterhub datarmor: <https://datarmor-jupyterhub.ifremer.fr>

Ouvrir notebook



datashader_s1_roughness.ipynb