

# Filtering of pulsed lidar's data using spatial information and a clustering algorithm

Leonardo Alcayaga

DTU Wind Energy Department

April 7, 2020

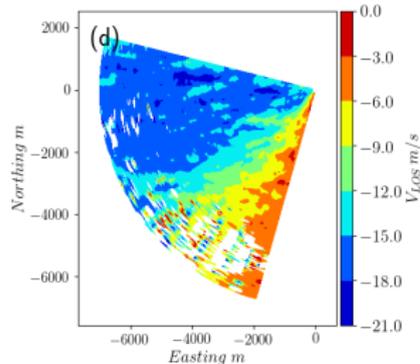
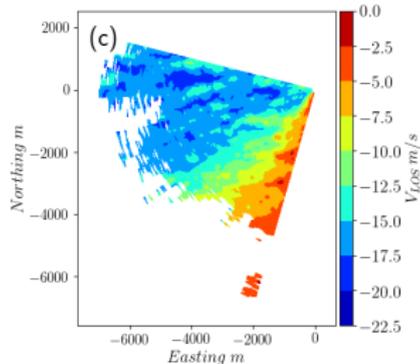
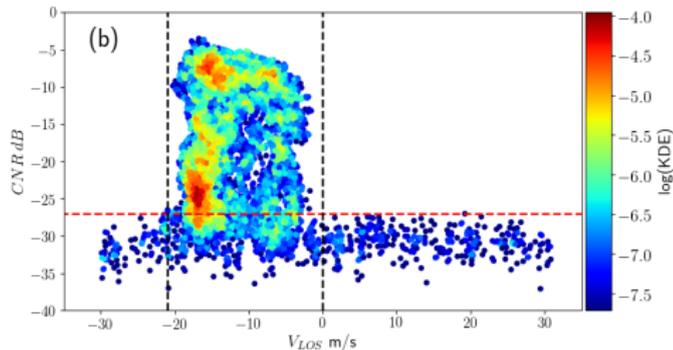
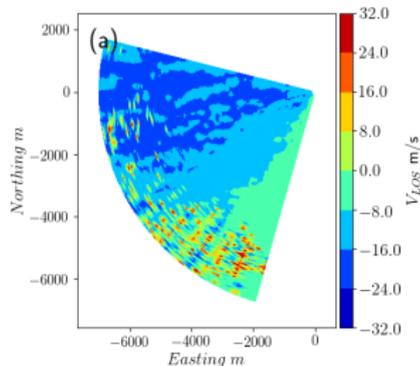
# Outline

1. The problem
2. Filtering techniques
3. Performance comparison on synthetic data
4. Performance on real data
5. Some final remarks

# The problem

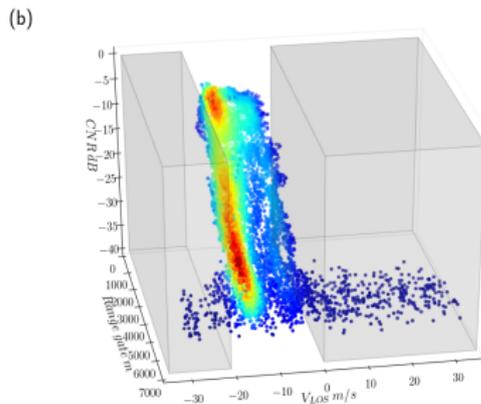
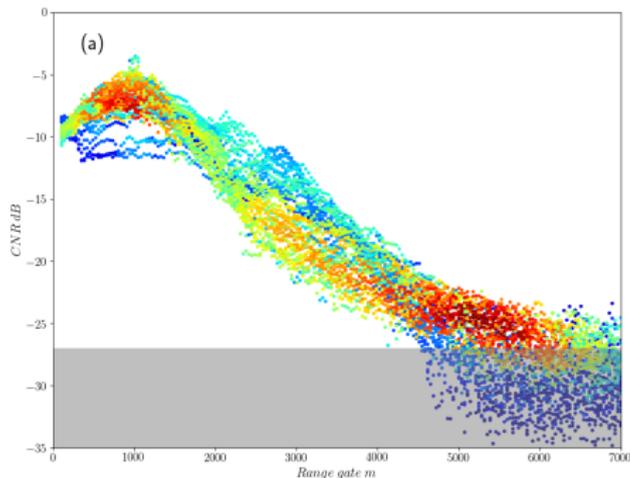
- Characterization of large-scale, turbulent, coherent atmospheric structures from long-range pulsed lidar measurements.
- Characterizing these structures requires a large measuring area.
- Long-range lidars are good at this, but data often show noisy/low CNR values in the far region.
- Is it possible to identify reliable lidar observations that have a relatively low CNR value?

# Filtering using a CNR threshold, the comb shape



# Filtering using a CNR threshold, the comb shape

- CNR thresholds may vary as well as the location of the "center" of the comb. There is an important number of reliable and non reliable observations in the intersection of this center and the base of the comb.

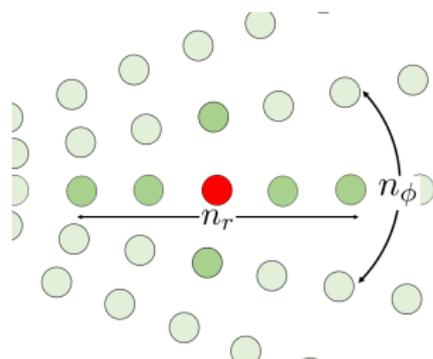


## Filtering using a CNR threshold, the comb shape

- Reliable observations in the low CNR region are placed close together in a limited region.
- These high density regions are evidence of a smooth radial wind speed field.
- Part of the high density region is located in a noisy region due to the decrease of CNR along the lidar's beams.
- Data density, radial wind speed smoothness and spatial location are then complementary to CNR values.

# Radial wind speed field smoothness, median filter

- Median filter is usually recommended for image processing.
- Here it is adapted to identify and reject anomalous values of line-of-sight wind speed,  $V_{LOS}$ , no replacing.
- Four parameters: window size,  $n_r$  and  $n_\phi$ , and a radial wind speed threshold,  $\Delta V_{LOS,median}$ .



# Radial wind speed field smoothness, median filter

This approach is:

- Fast.
- Excellent in recovering data from reliable areas (high CNR values) of the scan.
- Arbitrary. Threshold and window size that may be adjusted to flow characteristics and measurement height for instance.
- Not very reliable in noisy regions that are too extended.
- Easy to implement in a few lines of code. In Python, pandas library includes an specially fast moving median function.

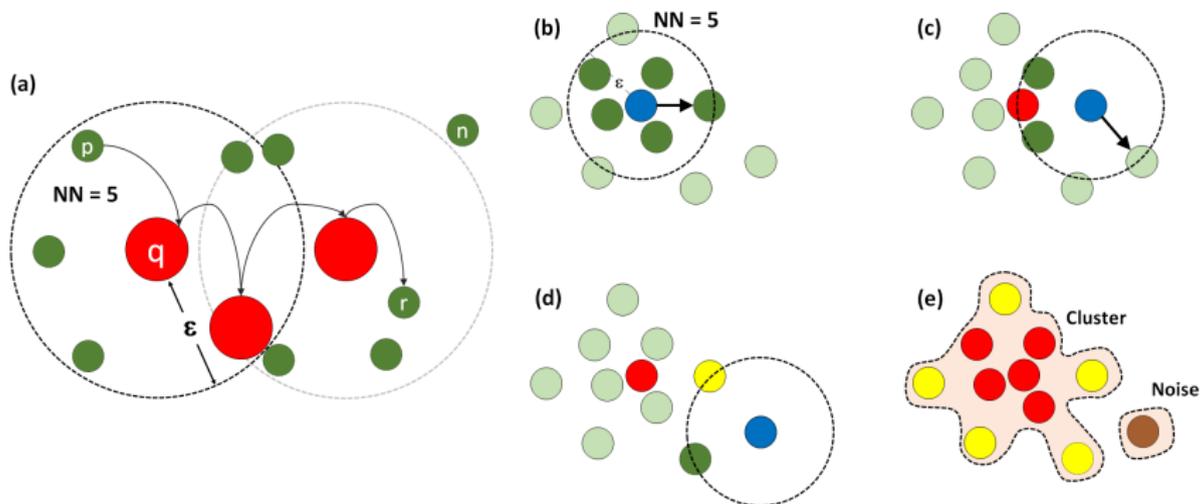
# Density of the data

- High density regions can be identified with kernel density estimation (KDE) (Beck and Kühn 2017), needs to define a density threshold to isolate noise/low density regions.
- The more features we consider in the KDE, the greater the distance between high density clusters and noise, but unbearable for higher dimensions/features.
- What to do then? How to avoid KDE in higher dimensions?

# Density of the data, a clustering algorithm

- Clustering algorithms are very efficient and there are many algorithms out there: centroid models, connectivity models, density based models, etc.
- **Density-based Spatial Clustering for Applications with Noise**, DBSCAN (Ester et al. 1996).
- Spatial distribution of data, previous knowledge of the number of clusters in the data is not necessary (k-means) and introduces the concept of noise.
- A robust filter needs few parameters to be defined by the user. DBSCAN needs in practice just one.

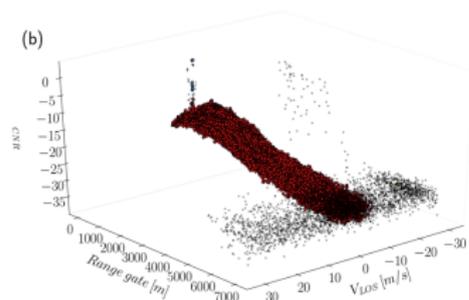
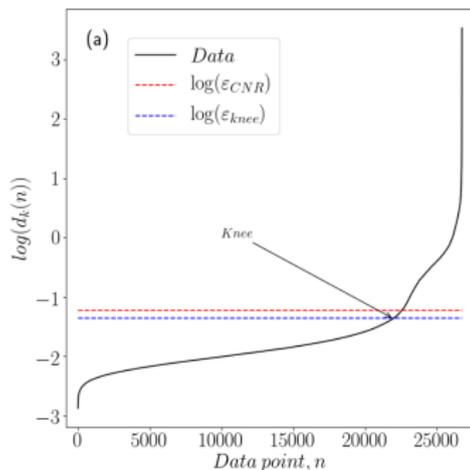
# DBSCAN, how does it work?



(a) DBSCAN definitions: direct density reachable point  $p$  (reachable by the core point  $q$ ) and density reachable and density connected points  $p$  and  $r$ . Here point  $n$  is noise. DBSCAN working: (b) The current point has the required number of nearest neighbours, NN, within  $\epsilon$ , then a *core point* (red) (c) The next point has less than NN neighbours, but one of them is a core point and becomes a *border point* (yellow) (d) A point with neither NN neighbours, nor core points within  $\epsilon$ , classified as *noise* (brown) (e) The final cluster and noise.

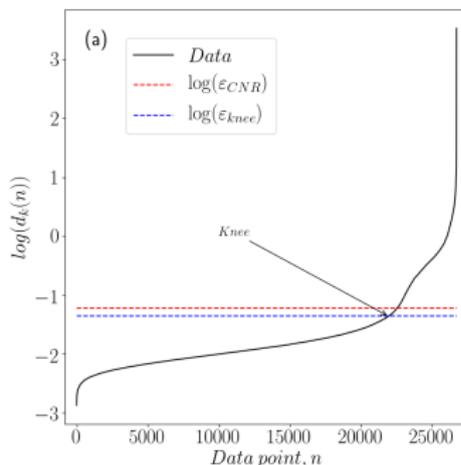
# Filtering using a density approach, DBSCAN

- NN can be fixed (to 5 in this case) and  $\epsilon$  estimated automatically according to the structure of the data.
- We end up choosing the number of features to consider and the amount of data/scans to filter per batch.



# Filtering using a density approach, DBSCAN

- What do I mean by data structure?
- The figure below shows the sorted k-distances, or the distance from each observation to its k-nearest neighbour.
- As we move from a cluster to low density areas this distance increase rapidly, showing a knee and a limit for  $\epsilon$ .

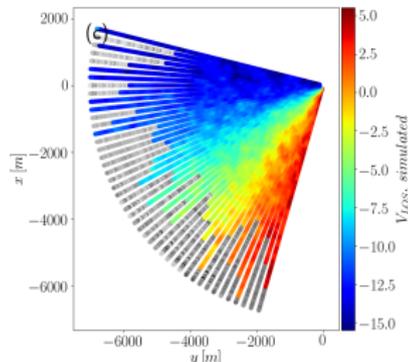
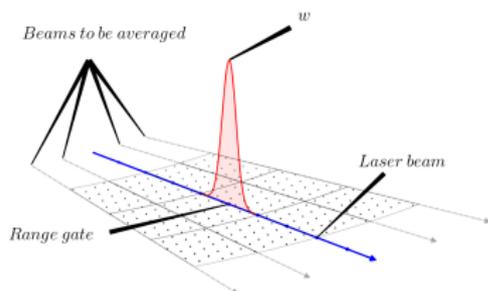


# Filtering using a density approach, DBSCAN

- Slower than the median-like filter, more accurate in noisy areas, some data is lost in good CNR regions.
- Several implementations of the algorithm,
  - In Python, the library `scikit-learn` includes also OPTICS an improved DBSCAN for clusters with large differences in density.
  - In R the package `dbscan` also includes OPTICS and the local outlier factor (LOC) algorithms.
  - DBSCAN is also included in the Machine Learning and Statistics toolbox of Matlab.

# Performance comparison on synthetic data

- Both filters were tested on synthetic data, sampled from 2-D Mann-turbulence box via a numerical lidar.
- Numerical lidar mimics the beam averaging and the accumulation of information on azimuth direction.
- Synthetic scans contaminated with procedural (smooth) noise. Contaminated area increases with distance.
- No CNR.

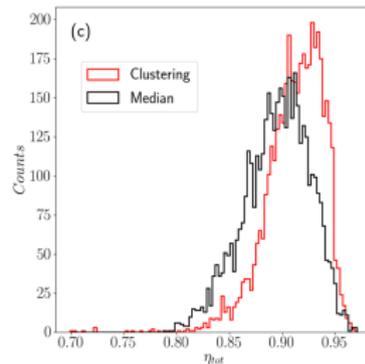
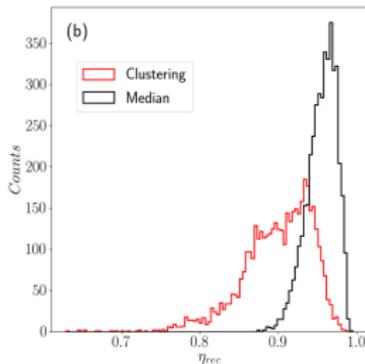
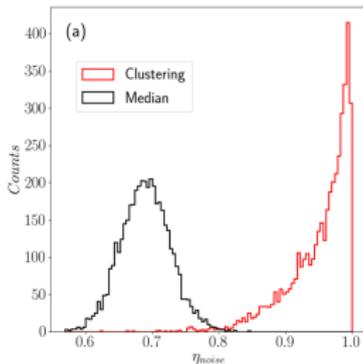


# Performance comparison on synthetic data

- Features of the clustering filter:
  - $V_{LOS}$
  - Position,  $r$  and  $\phi$ .
  - $\Delta V_{LOS} = \text{median}(V_{LOS,i} - V_{LOS,NN})$
- Fair comparison, the optimal set of  $n_r$ ,  $n_\phi$  and  $\Delta V_{LOS,median}$  comes from maximum values of:
  - $\eta_{noise}$ : fraction of noise detected.
  - $\eta_{recov}$ : fraction of good measurements recovered.
  - $\eta_{tot} = f_{noise}\eta_{noise} + (1 - f_{noise})\eta_{recov}$ .

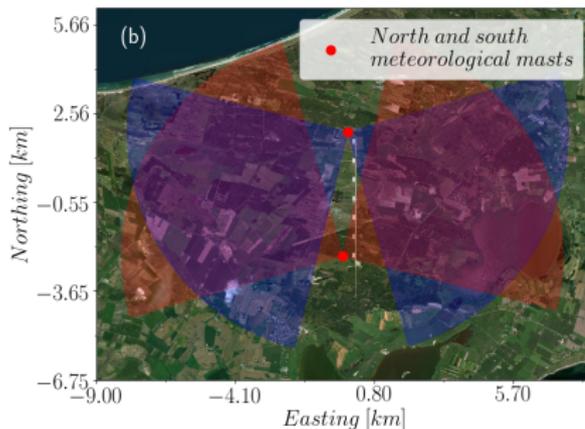
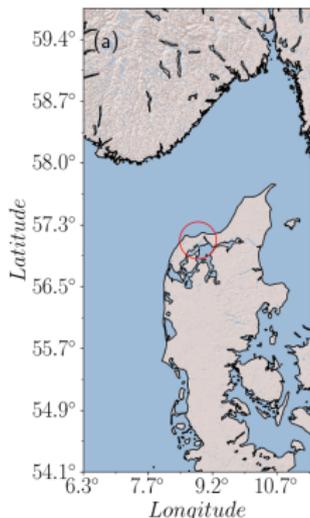
# Performance comparison on synthetic data

- The clustering filter performs better in noise detection, while keeping a good recovering rate.



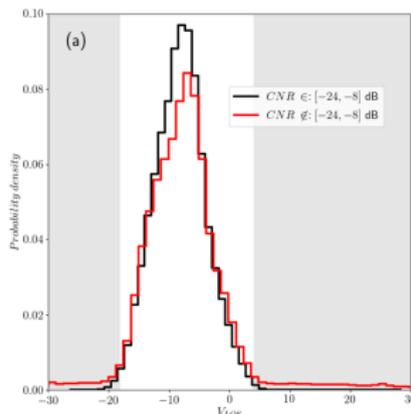
# Performance on real data

- Wind speed data in space-time from two scanning lidars in Østerild Wind Turbine test centre (Karagali et al. 2018), at 50m (Phase 1) and 200m (Phase 2).
- High resolution in space (range gates each 35 m.) and in time (45 seconds per scan). The lidars are not synchronous.

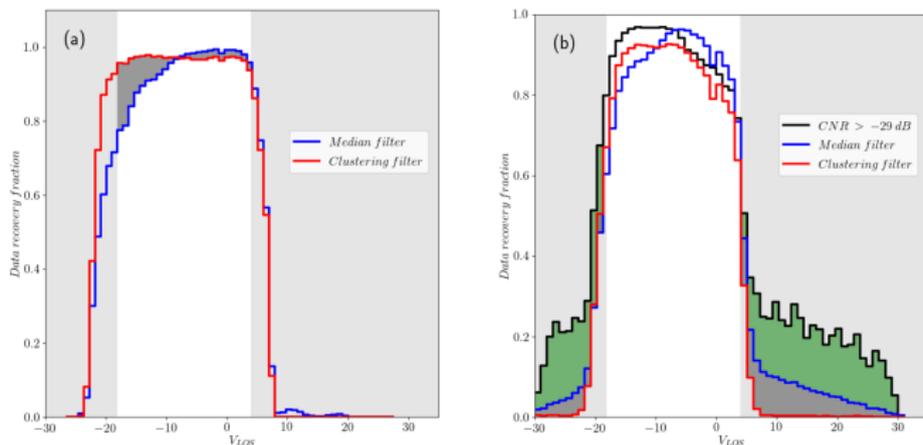


# Performance comparison on real data

- Filters performance on less reliable areas of the scan (CNR values less than -24 dB).
- The recovery rate on reliable areas is also studied (CNR values greater than -24 dB).
- The features used:  $V_{LOS}$ ,  $r$ ,  $\phi$ ,  $\Delta V_{LOS}$  and CNR.



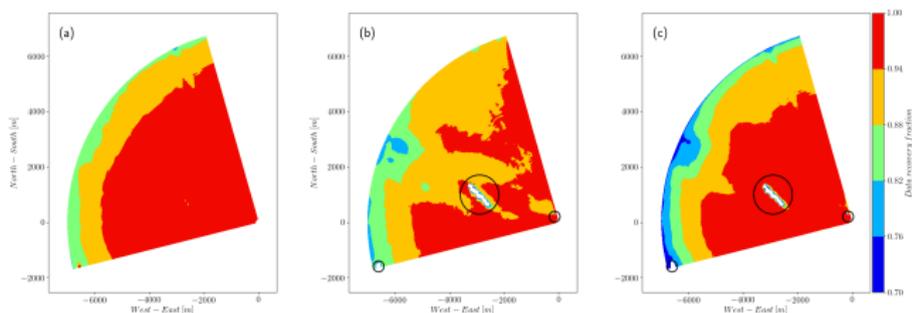
# Performance comparison on real data



Distribution of recovery fraction per wind speed bin for phase 1 of the experiment of (a) reliable observations ( $-24 < CNR < -8$ ) and (b) non reliable data ( $CNR < -24$  or  $CNR > -8$ ) for the three types of filter. Shadowed area in both graphs corresponds to the region where observations exceed the 99.7% of probability (or  $3\text{-}\sigma$  limit) in the pdf of reliable observations. The darker shadowed areas highlights the additional fraction of extreme values non-filtered by the median-like and CNR filters, when the former uses the optimal input set  $n_r = 5$ ,  $n_\phi = 3$  and  $\Delta V_{LOS, threshold} = 2.33$  m/s.

# Performance comparison on real data

- Spatially, the clustering filter tends to reject more data in the far region of the scan.
- A fraction of reliable values are rejected in the near region. It can be combined with a CNR threshold.



Total recovery fraction for phase 1 of the experiment. The noisy and far region of the scans show a high recovery, above 80%, for (a) the CNR  $> -29$  dB threshold filter and (b) the median-like filter and below 75% for (c) the clustering filter. Highlighted, hard targets (turbines and one meteorological mast, close to the lidar), which are identified by the median and clustering filter with recovery rates below 20%.

## Final remarks

- Clustering filter uses the best of two approaches:  $V_{LOS}$  field smoothness and CNR information.
- For this data set, recovery increased by around 38% compared to CNR filters.
- Little user intervention, mostly in the definition of relevant features and the amount of data needed (more features, more observations are necessary).
- Need to be tested on different scanning patterns.
- Computational complexity. DBSCAN's goes from  $\mathcal{O}(n \log(n))$  to  $\mathcal{O}(n^2)$ . Very efficient median filters, down to  $\mathcal{O}(n)$ .
- Don't hesitate to use this technique on your own data. I uploaded a repository to github:  
[https://github.com/lalcayag/Lidar\\_filtering](https://github.com/lalcayag/Lidar_filtering)

# References I

-  Ankerst, Mihael et al. (1999). "OPTICS: Ordering Points To Identify the Clustering Structure". In: *Proc. ACM SIGMOD'99 Int. Conf. on Management of Data*. ACM Press, pp. 49–60.
-  Beck, Hauke and Martin Kühn (2017). "Dynamic Data Filtering of Long-Range Doppler LiDAR Wind Speed Measurement.". In: *Remot Sens* 9(6), p. 561. DOI: <https://doi.org/10.3390/rs9060561>.
-  Burger, Wilhelm and Mark J. Burge (2008). *Digital Image Processing - An Algorithmic Introduction using Java*. Texts in Computer Science. Springer, pp. I–XX, 1–565. ISBN: 978-1-84628-968-2.
-  Cariou, Jean Pierre (2015). "Remote Sensing for Wind Energy". English. In: *Remote Sensing for Wind Energy*. Ed. by Alfredo Peña et al. Denmark: DTU Wind Energy. Chap. Pulsed lidars, pp. 131–148.
-  Ester, Martin et al. (1996). "A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon, pp. 226–231.

# References II

-  Gryning, Sven-Erik and Rogier Floors (2019). "Carrier-to-Noise-Threshold Filtering on Off-Shore Wind Lidar Measurements". In: *Sensors* 19.3. ISSN: 1424-8220. DOI: 10.3390/s19030592.
-  Gryning, Sven-Erik et al. (2016). "Weibull Wind-Speed Distribution Parameters Derived from a Combination of Wind-Lidar and Tall-Mast Measurements Over Land, Coastal and Marine Sites.". In: *Bound-Lay Meteorol* 159(2), p. 329. DOI: <https://doi.org/10.1007/s10546-015-0113-x>.
-  Huang, T., G. Yang, and G. Tang (1979). "A fast two-dimensional median filtering algorithm". In: *IEEE T. Acoust. Speech* 27.1, pp. 13–18. DOI: 10.1109/TASSP.1979.1163188.
-  Karagali, Ioanna et al. (2018). "New European Wind Atlas: The Østerild balconies experiment". In: *J Phys Conf Ser* 1037, p. 052029. DOI: 10.1088/1742-6596/1037/5/052029.
-  MacQueen, James (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings Fifth Berkeley Symp. on Math. Statist. and Prob.* Vol. 1: Statistics. Berkeley, California, pp. 281–297.
-  Mann, Jakob (1994). "The spatial structure of neutral atmospheric surface-layer turbulence". In: *J Fluid Mech.* 273, 141–168. DOI: 10.1017/S0022112094001886.

# References III

-  Mann, Jakob (1998). "Wind field simulation". In: *Probabilist. Eng. Mech.* 13, pp. 269–282. DOI: [https://doi.org/10.1016/S0266-8920\(97\)00036-2](https://doi.org/10.1016/S0266-8920(97)00036-2).
-  Meyer Forsting, A. and N. Troldborg (2016). "A finite difference approach to despiking in-stationary velocity data - tested on a triple-lidar". In: *J Phys Conf Ser* 753, p. 072017. DOI: [10.1088/1742-6596/753/7/072017](https://doi.org/10.1088/1742-6596/753/7/072017).
-  Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine Learning in Python". In: *J. Mach. Learn. Res.* 12, pp. 2825–2830.
-  Peña, Alfredo et al., eds. (2015). *Remote Sensing for Wind Energy*. English. Denmark: DTU Wind Energy.
-  Perlin, Ken (2001). "Noise hardware". In: *In Real-Time Shading SIGGRAPH Course Notes*.
-  Rui Xu and D. Wunsch (2005). "Survey of clustering algorithms". In: *IEEE T. Neural Networ.* 16.3, pp. 645–678. DOI: [10.1109/TNN.2005.845141](https://doi.org/10.1109/TNN.2005.845141).
-  Simon, Elliot and Nikola Vasiljevic (Nov. 2018). "Østerild Balconies Experiment (Phase 2)". In: DOI: [10.11583/DTU.7306802.v1](https://doi.org/10.11583/DTU.7306802.v1). URL: [https://data.dtu.dk/articles/\\_sterild\\_Balconies\\_Experiment\\_Phase\\_2\\_/7306802](https://data.dtu.dk/articles/_sterild_Balconies_Experiment_Phase_2_/7306802).

# References IV

-  Smalikho, I. N. and V. A. Banakh (2013). “Accuracy of estimation of the turbulent energy dissipation rate from wind measurements with a conically scanning pulsed coherent Doppler lidar. Part I. Algorithm of data processing”. In: *Atmos. Ocean. Opt.* 26, pp. 404–410. ISSN: 2070-0393. DOI: 10.1134/S102485601305014X.
-  Vasiljević, N. et al. (2017). “Perdigão 2015: methodology for atmospheric multi-Doppler lidar experiments”. In: *Atmospheric Measurement Techniques* 10.9, pp. 3463–3483. DOI: 10.5194/amt-10-3463-2017.
-  Vasiljevic, Nikola et al. (2016). “Long-Range WindScanner System”. In: *Remote Sensing* 8. DOI: 10.3390/rs8110896.

## Wrapping up

- Filtering with a CNR threshold can be tricky and depends on many variables that the data user might be not familiar to.
- The proper selection of this threshold might have a big impact in wind energy resource estimations.
- Information beyond CNR is available and can be used to increase data availability keeping good quality of the data.
- The main idea is to keep the number of decisions made by the user low, specially if the amount of data to use is large and extend for a long period of time.
- Here, two alternative approaches are presented with different computational cost and implementation complexity. Try them!. More details in, <https://doi.org/10.5194/amt-2019-450>

# Questions?