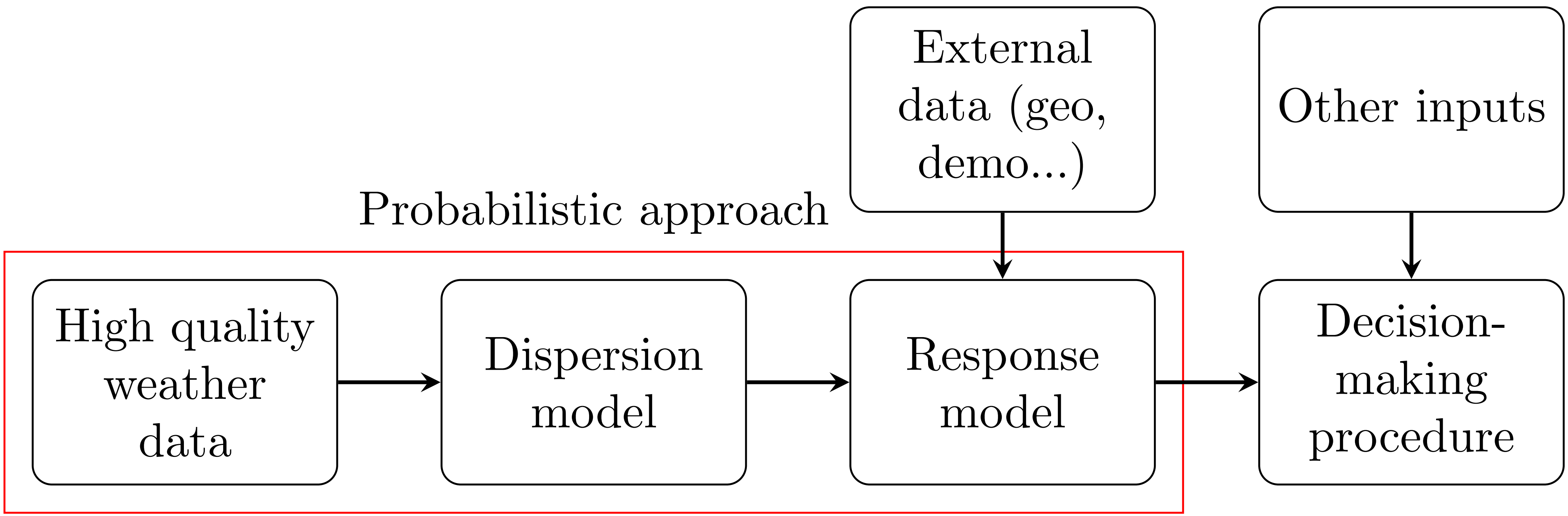
Evaluation of a military CBRN-hazard prediction procedure with a Lagrangian dispersion model

Tristan Carion, Andy Delcloo, Bart Janssens

September 2022

Context

- Improve risk assessment in case of CBRN incidents (Chemical, Biological, Radiological, Nuclear)
- Atmospheric dispersion models can predict hazardous species concentration
- Event-based simulation to couple concentration predictions with population density, topography etc.
- Quantify the uncertainty



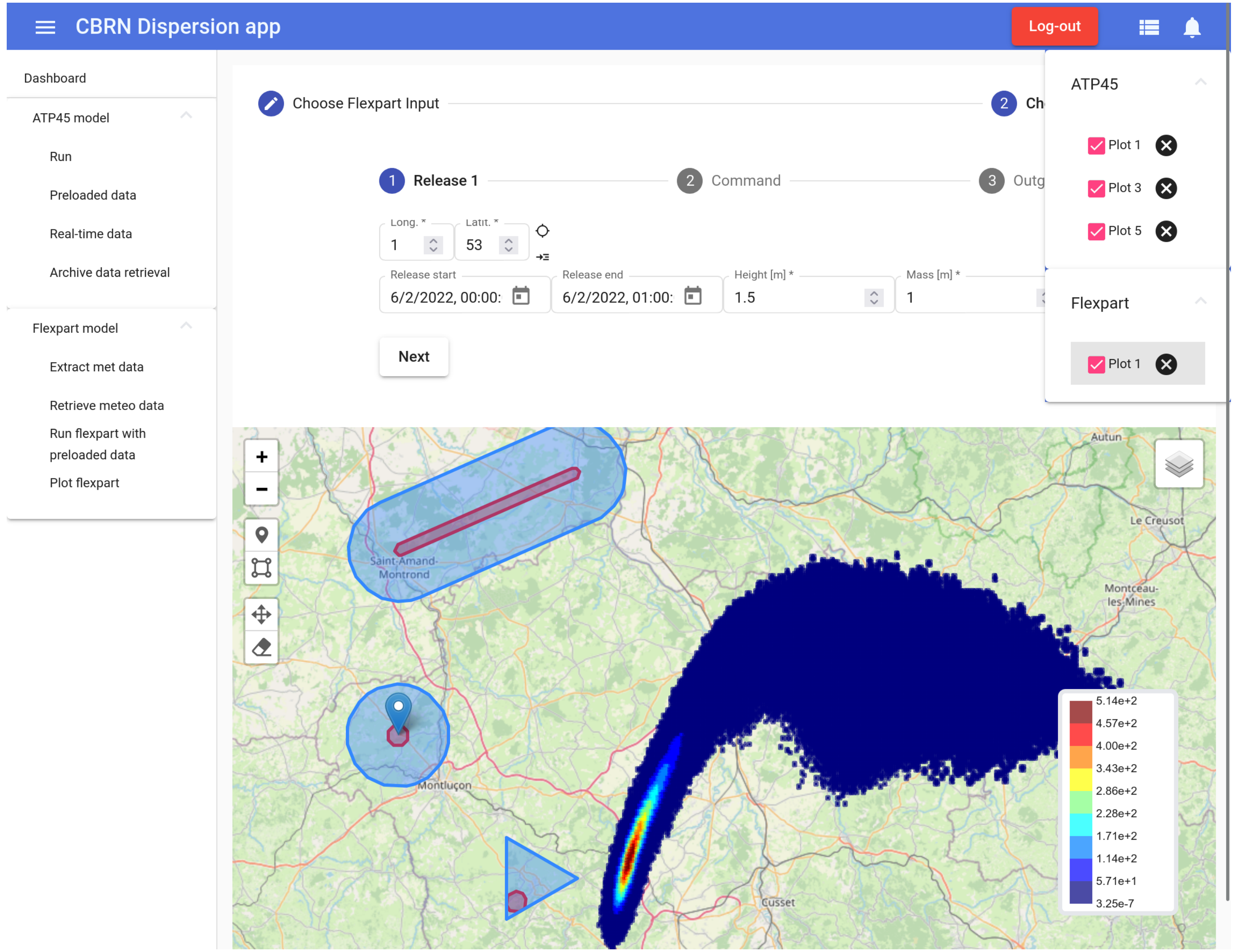
Web Application

We want:

- Quick and user-friendly results
- Fast access to meteorological data
- Dispersion models anywhere on the globe

Solution:

- Web app on the European Weather Cloud
- Rest API with julia
- Multiple dispersion models implemented

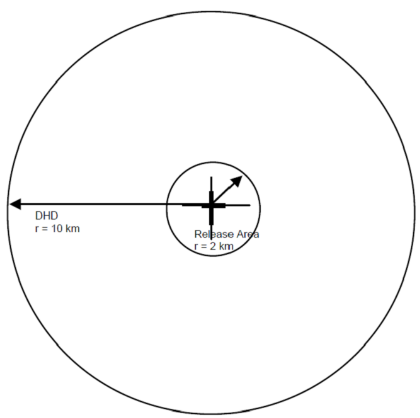


Dispersion models

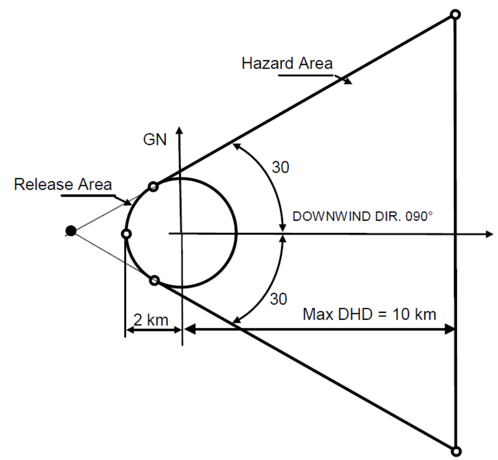
Two models are currently implemented

ATP-45 (simplified version)

- NATO hazard predictions in case of CBRN
- Defines geospatial shapes representing the hazard zones (valid for 2 hours).
- Different shapes for low and high wind speed.



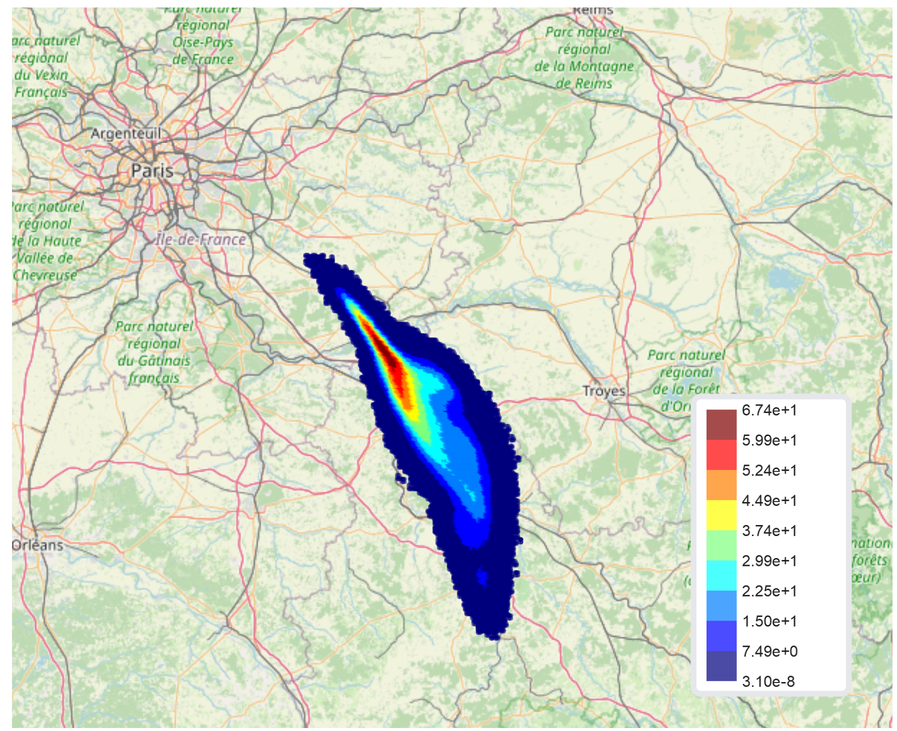
Wind speed < 10km/h



Wind speed > 10km/h

FLEXPART

- Comprehensive Lagrangian dispersion model
- Spatial distribution of the concentration at multiple levels



ATP-45 vs Flexpart

	Pro	Cons
ATP-45	• Few meteorological and input data required • Very easy and quick to run	• Very limited precision of the prediction
Flexpart	• Precise idea of spatial and temporal behaviour of the plume	• Many meteo fields required (long retrieval time) • Time consuming simulations

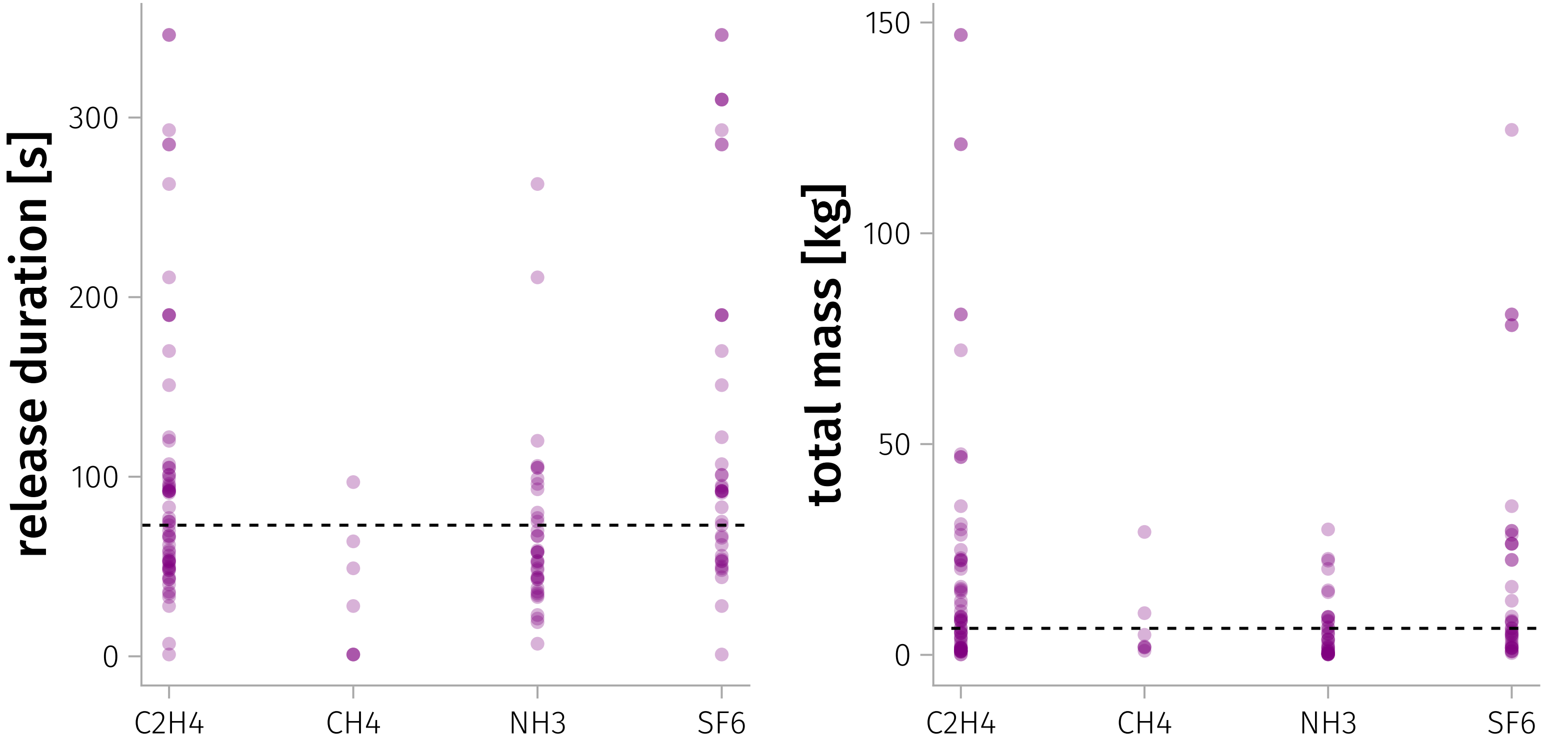
ATP-45 when very quick assessment is needed

Flexpart when more precise assessment is needed

Can we improve future ATP-45 predictions by previous evaluations with Flexpart (i.e. making a surrogate model)?

The Suffield campaign

- Experimental campaign for detection of CBRN agents in atmosphere
- Short-time releases (median ≈ 1 minutes)
- About 150 releases analyzed



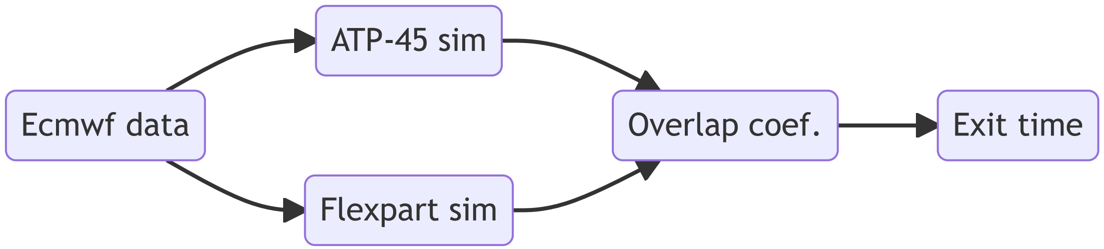
Evaluation process

Overlap coefficient:

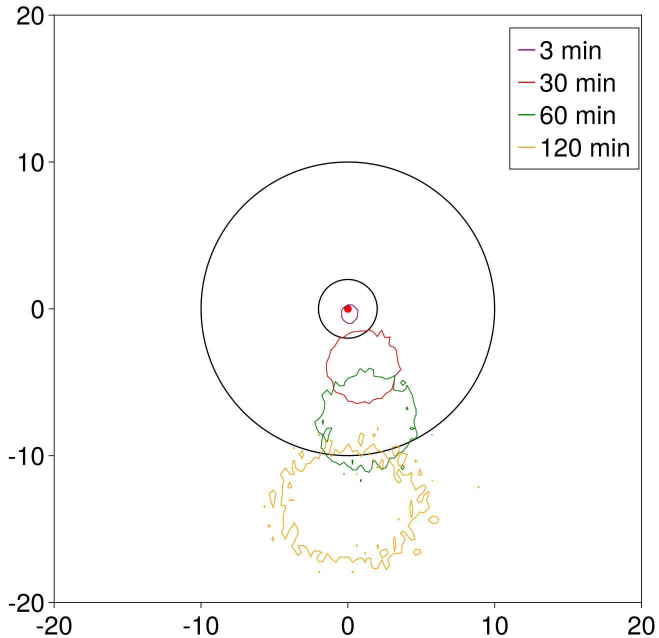
$$OR(t) = \frac{FP(t) \cap ATP}{FP(t)}$$

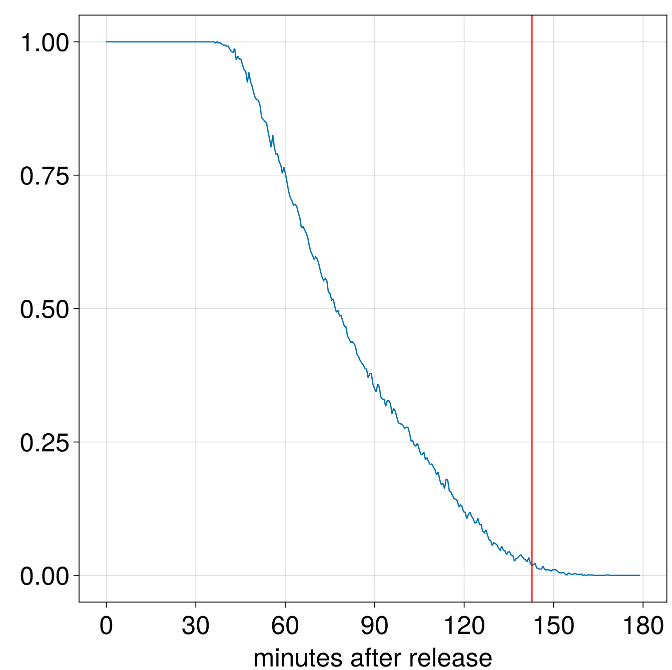
Exit time:

$$T : OR(T) = 0$$



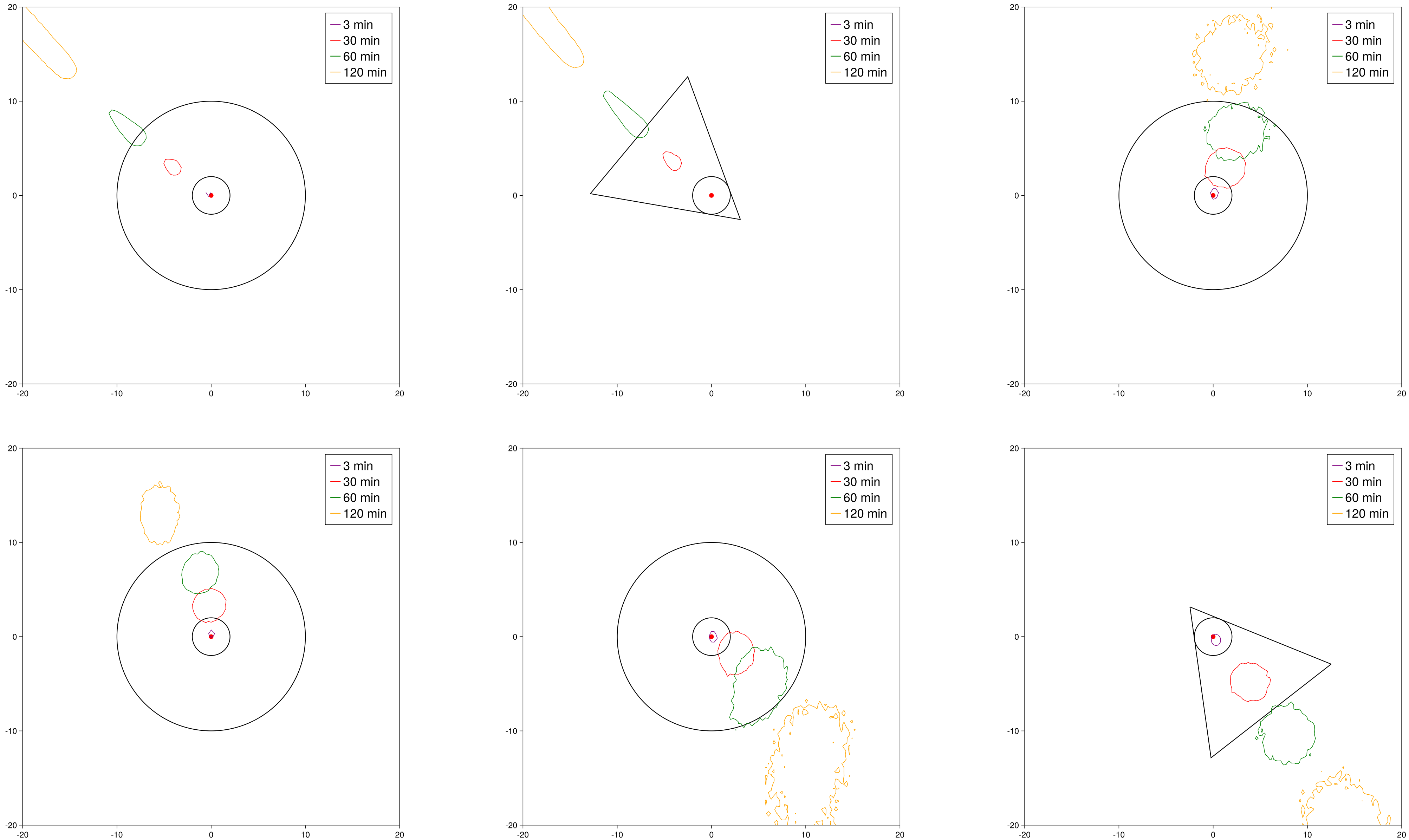
Spatial distribution at 2m



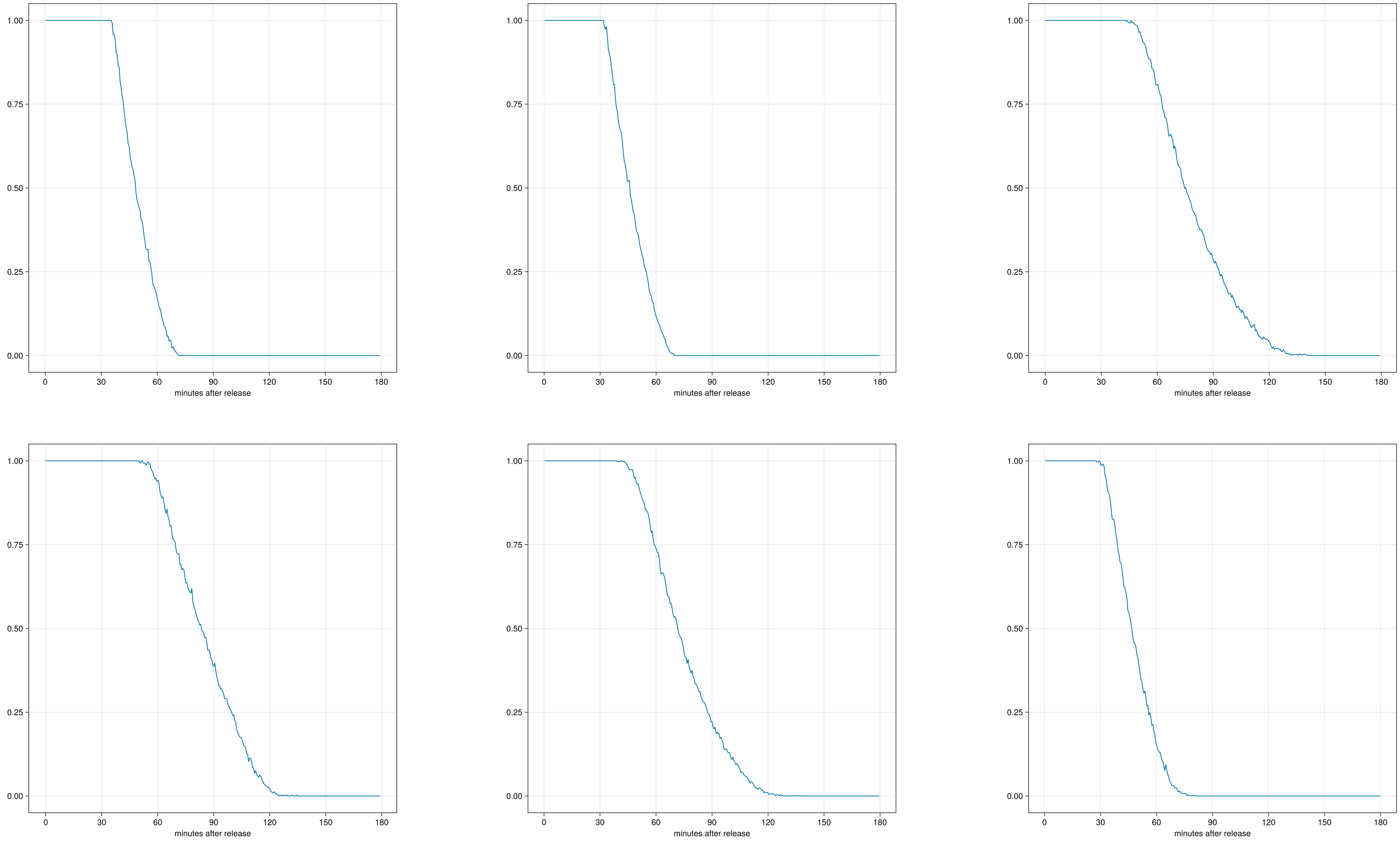


Red line = exit time

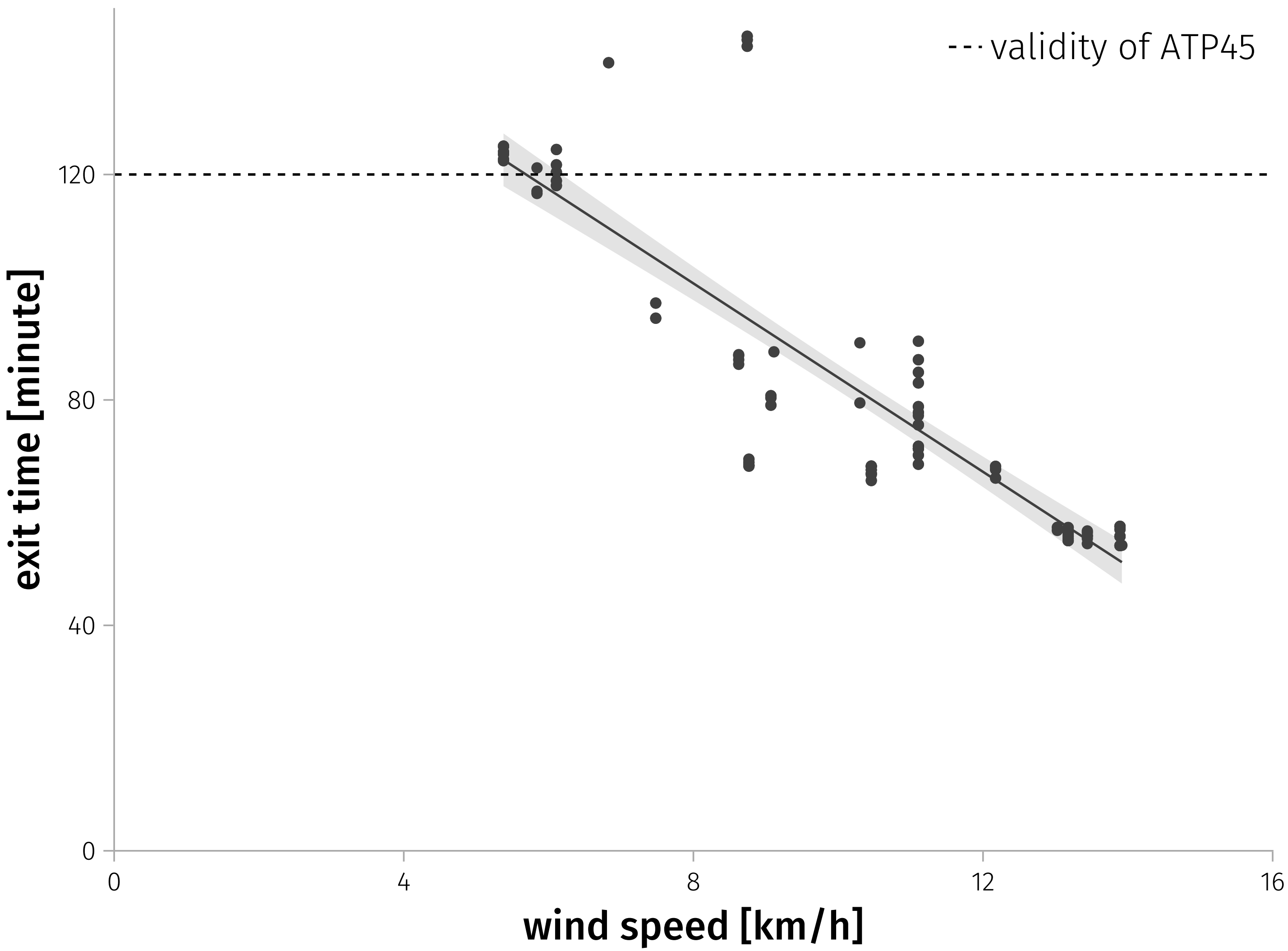
Plume footprints



Overlap coefficients

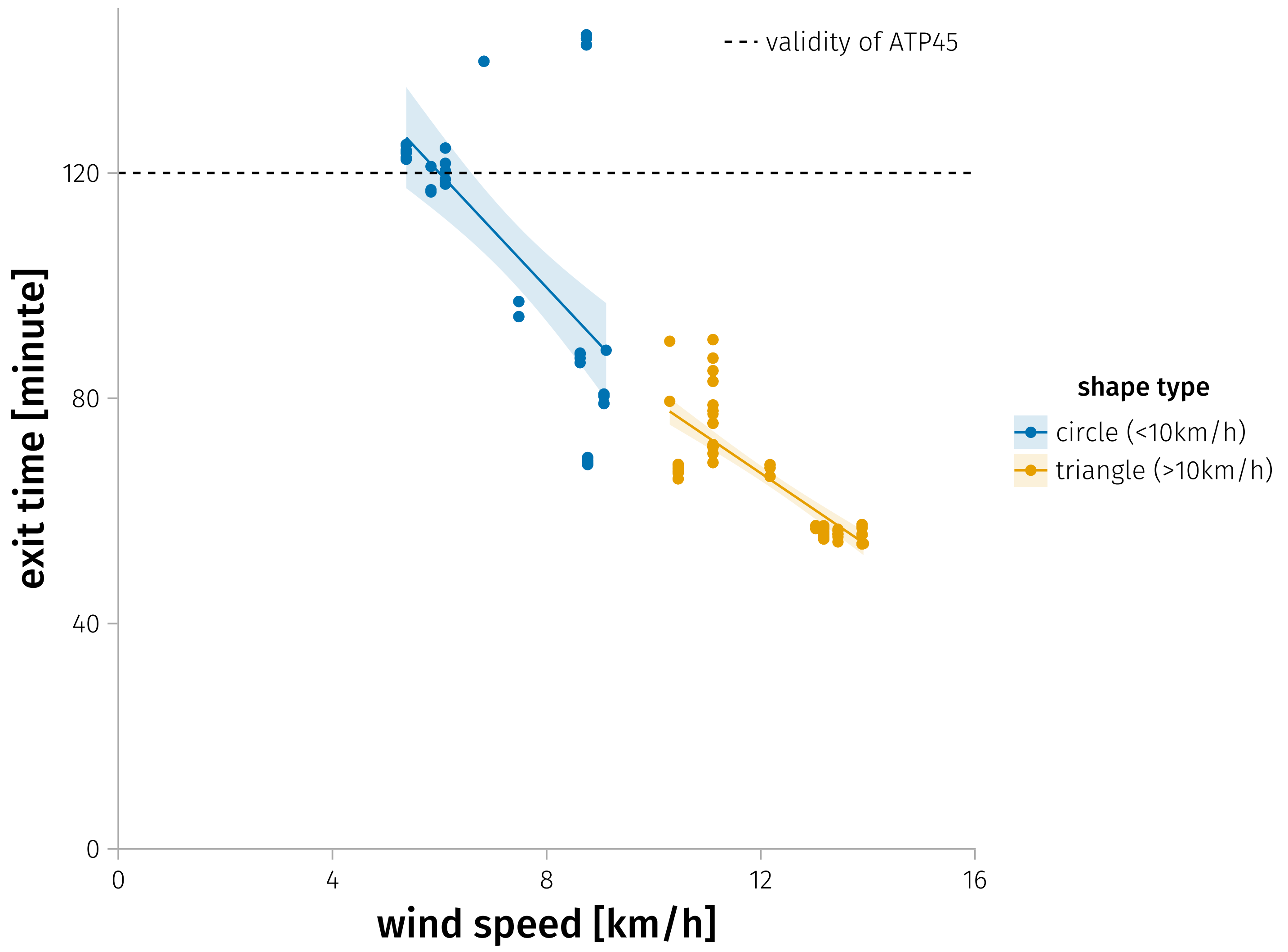


Exit time vs wind speed (1)



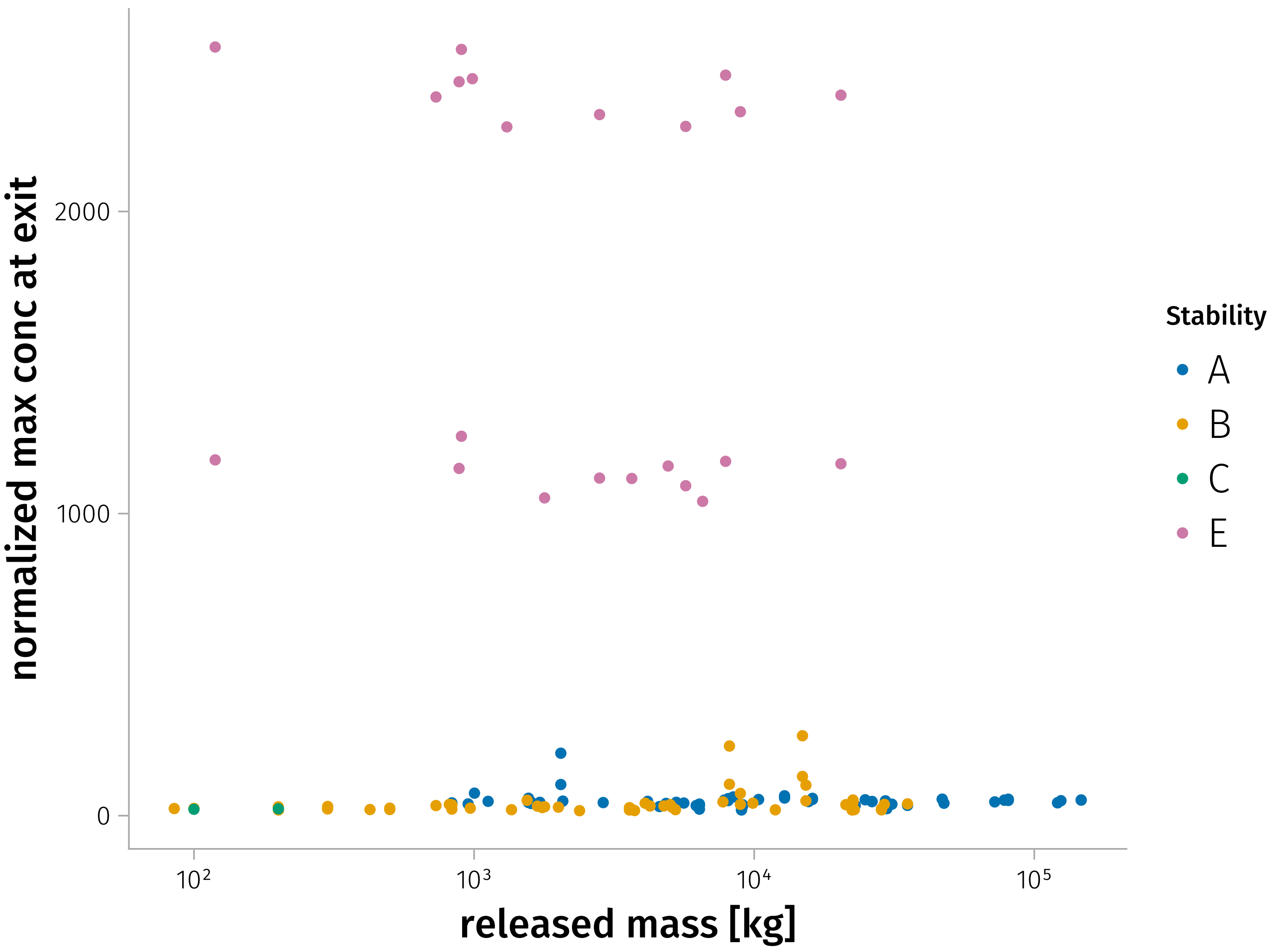
- Correlation between wind speed and exit time
- Plumes exit before ATP45 validity

Exit time vs wind speed (2)



- Correlation between wind speed and exit time
- Plumes exit before ATP45 validity
- Different correlation for each shape type
- More uncertainty in case of low wind speed
- But nice continuity between the 2 shapes

Concentration at exit



- Max concentration at exit time divided by the total mass released
- Higher concentration when stable conditions
- Improve the prediction if the release quantity and the stability is known.

Final discussion

Conclusions:

- Improve risk assessment when wind velocity is known
- Most of the time, the plume exits before end of validity
- Concentration predictions if more information about release conditions

Limitations:

- Valid for short releases (when the exit time makes sense)
- Only for open and flat terrain
- Not many cases at low wind speed
- Not every stability classes covered
- Dataset quite limited

Further steps:

- Randomly generated releases (random release location and mass)
- Define other metrics (i.e. the false alarm: $\frac{(FP(t) \cap ATP)^c}{ATP}$)
- Use more detailed ATP-45 models
- Influence of more detailed atmosphere characteristics (ex: Monin-Obukhov length)
- Machine learning for automated surrogate model

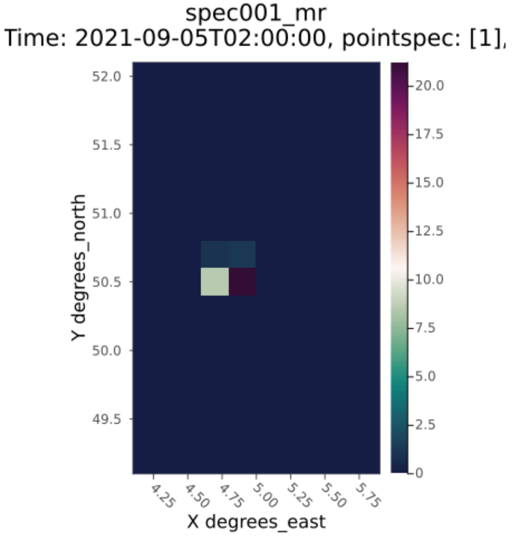
Additional remark

Multiple packages developed (https://github.com/tcarion) :

- **EcRequests.jl**: call to the ECMWF web API in Julia
- **GRIBDatasets.jl**: high level interface to GRIB files
- **ATP45.jl**
- **GaussianDispersion.jl**
- **Flexpart.jl**:

```julia
using Pkg; Pkg.add(["Flexpart", "Rasters", "Plots"])
using Flexpart, Rasters, Plots

FlexpartDir() do fpdir
    Flexpart.default_run(fpdir)
    output_file = first(OutputFiles(fpdir))
    output = Raster(string(output_file), name = :spec001_mr)
    plot(output[Ti = 2, height = 1, pointspec = 1, nageclass = 1])
end
```



Final slide

Thank you very much for your attention!

tristan.carion@mil.be