

# TKI-3 Pilot Noorderzijlvest

D-HyDAMO import en verificatie met D-HYDRO Suite



# Doelen TKI-3 Pilot Noorderzijlvest

1. Geautomatiseerde en reproduceerbare D-HYDRO *modelbouw* van een boezemmodel voor NZV vanuit brondata
2. Onderzoek *modelprestatie*: Peil in Rust, half maatgevende afvoer en dynamische afvoer
3. Onderzoek *meerwaarde D-HYDRO*: samengestelde kunstwerken (compound structures) en meervoudige sturing (maalstops)

Deze presentatie: Geautomatiseerde en reproduceerbare D-HYDRO *modelbouw* met behulp van *D-HyDAMO* (1)

# ***Modelbouw met behulp van modelgenerator D-HyDAMO***

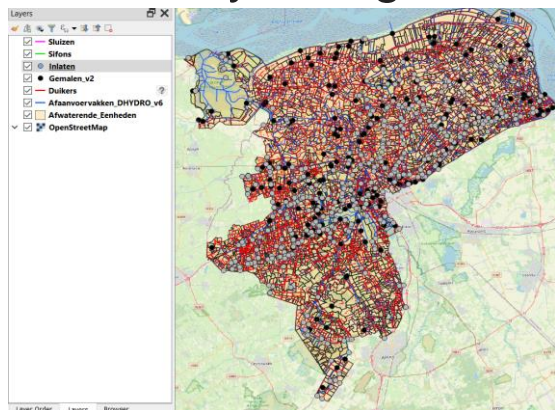
Deze presentatie:

- Aanpak: beschrijving geautomatiseerd workflow
- Uitdaging 1: valideren, filteren en opvullen
- Uitdaging 2: schematiseren

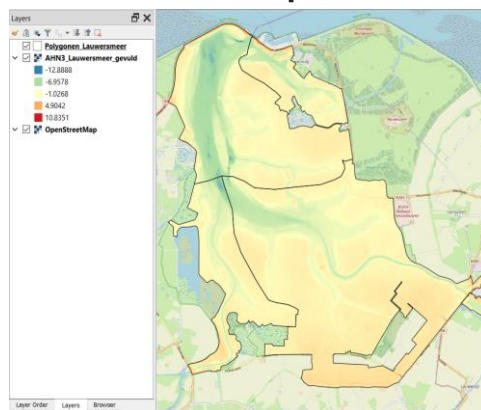
# Workflow: Brondata

## Gebruikte brondata:

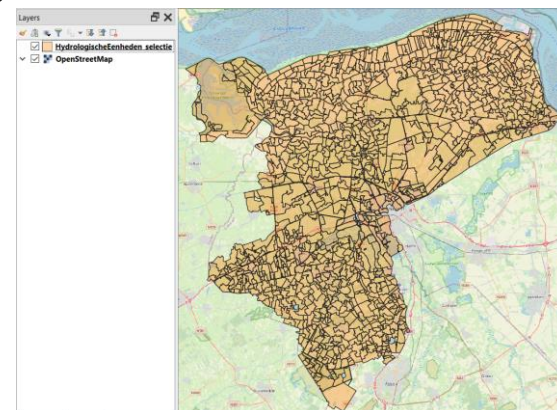
- DAMO bevat álle waterlopen + kunstwerken en veel profielen
- Lauwersmeer aangeleverd in een grid met bodemhoogten
- De hydrologische schematisatie separaat aangeleverd



**Beheerregister (DAMO)**



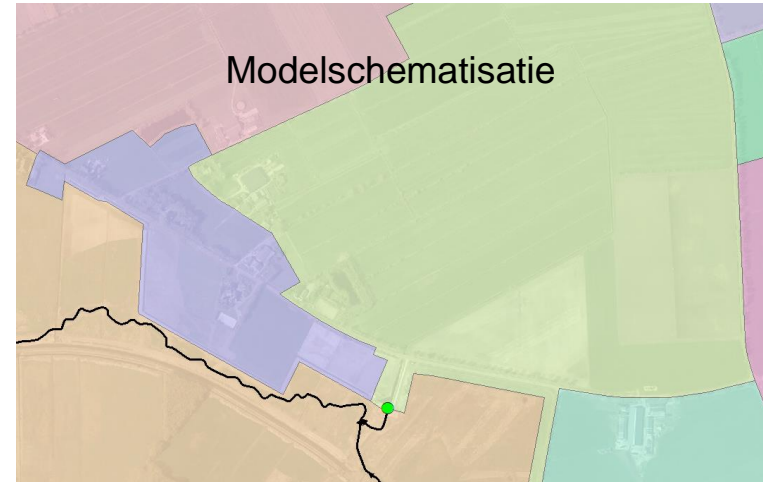
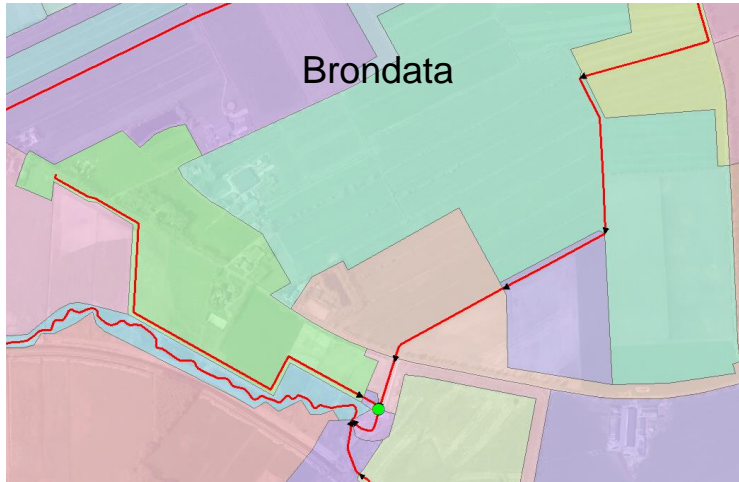
**Bathymetrie Lauwersmeer**



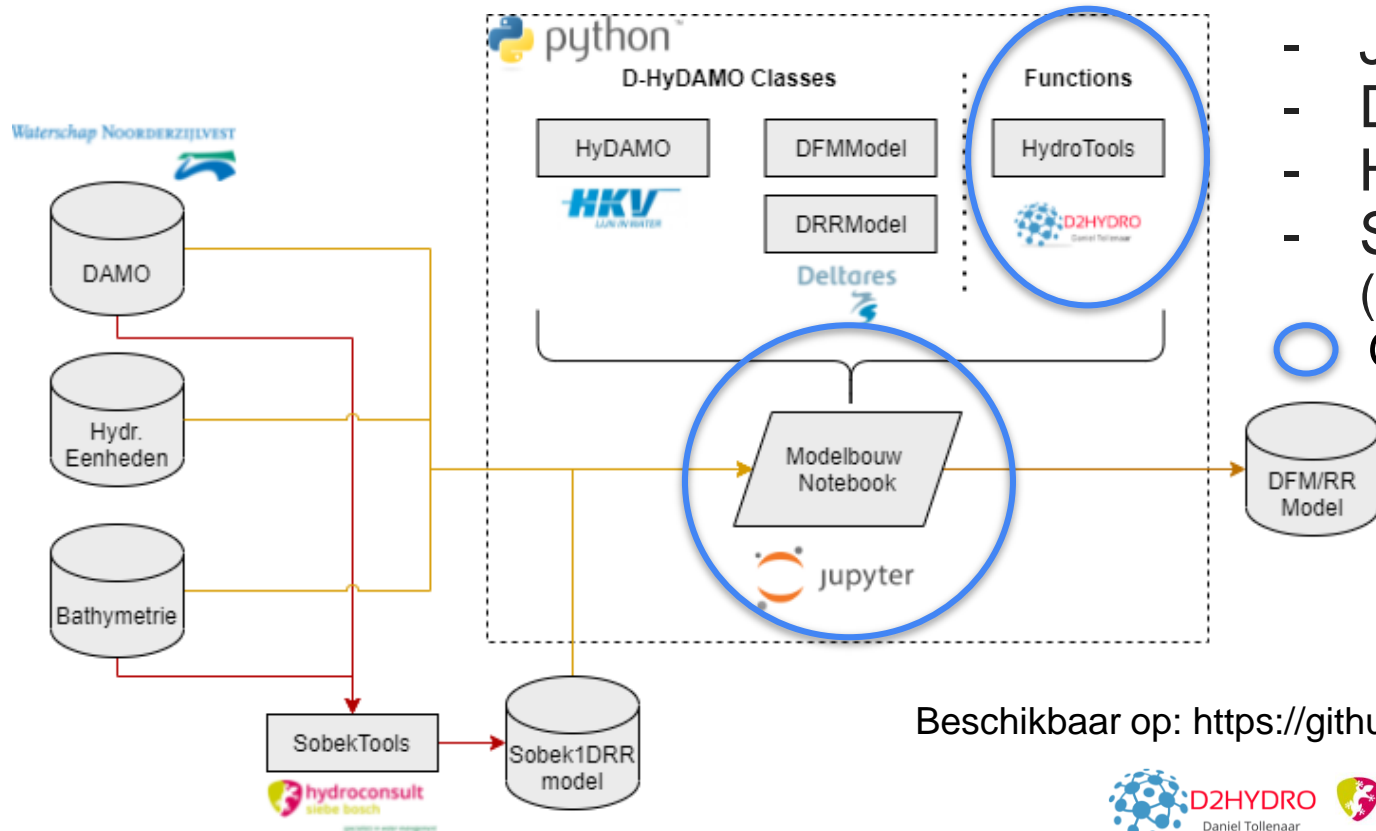
**Hydrologische eenheden**

# Workflow: Brondata (Hydrologische eenheden)

- Selectie watergangen op basis van afwaterend gebied, wateraanvoer en locatie kunstwerken
- 'Hydrologische eenheden' opgebouwd vanuit kleinere eenheden



# Workflow: dataverwerking



## Hoofdingrediënten:

- Jupyter Notebook
  - D-HyDAMO
  - HydroTools
  - SobekTools (RainfallRunoff(RR))
- Gemaakt voor TKI-3

Beschikbaar op: [https://github.com/d2hydro/nzv\\_pilot](https://github.com/d2hydro/nzv_pilot)

# Workflow: Resultaat

## Jupyter Notebook

github.com/d2hydro/nzv\_pilot/blob/master/modelbouw.ipynb

1926 lines (1926 sloc) 160 KB

### D-Hydro Boezemmodel V4 - Modelbouw

Met deze notebook bouw je een boezemmodel voor Noorderzijlvest.

De notebook is opgezet door Daniel Tollenaar (D2Hydro), Vincent de Looij (Waterschap Noorderzijlvest) en Siebe Bosch (HydroConsult). Het Notebook is gebaseerd op een [voorbeeldnotebook](#) uit D-HyDAMO, opgezet door HKV Lijn in Water.

### Installatie

Zorg dat je beschikt over een werkende Python installatie en [D-HYDAMO omgeving](#).

Het resulterende model moet geïmporteerd kunnen worden in D-HYDRO versie 0.9.10.52995 of hoger.

Download de bestanden van [Google Drive](#) en zet deze in de folder .idata

De verwijzing naar bestanden is geschreven in het Nederlands. De rest van de code is geschreven in het Engels. Elk code-blok is voorzien van uitleg boven het desbetreffende blok, geschreven in het Nederlands.

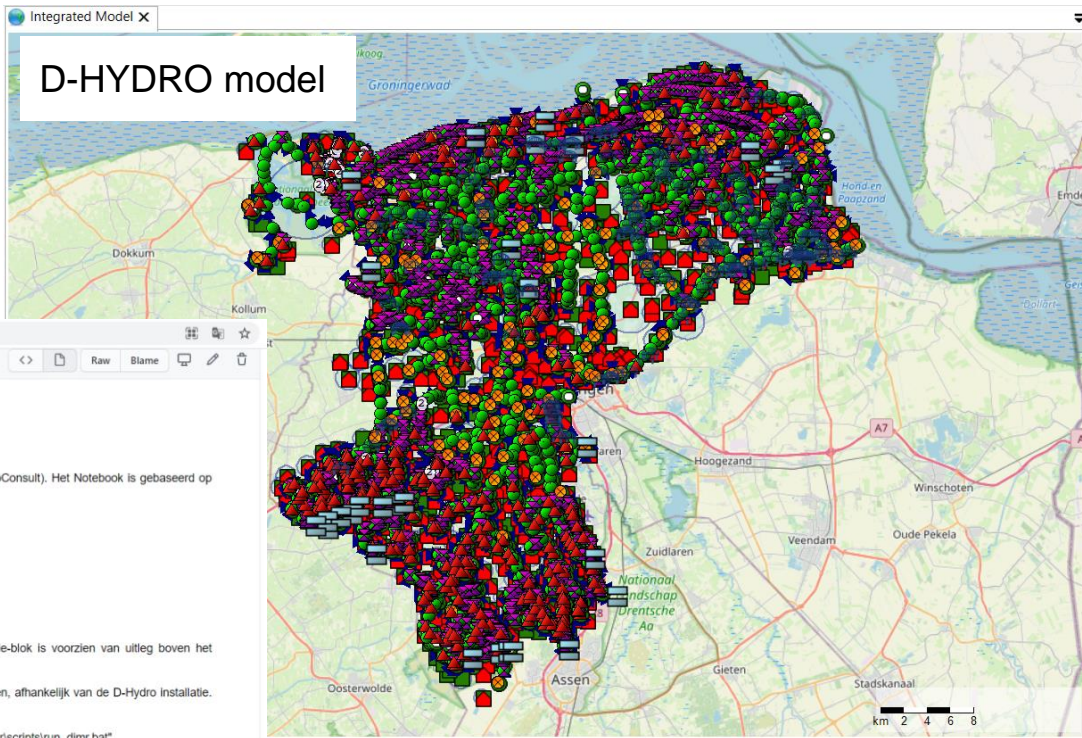
**let op:** deze notebook werkt alleen in combinatie met een config.py. In config.py dient een variabele dimr\_path te zijn opgegeven, afhankelijk van de D-Hydro Installatie. Bijvoorbeeld:

```
dimr_path = r"c:\Program Files (x86)\Deltaires\D-HYDRO Suite 1D2D (Beta) (0.9.7.52006)\plugins\DeltaSheil\Dimr\kernels\x64\dimr\scripts\run_dimr.bat"
```

### Controle bestanden

Alle bestanden die gebruikt worden in deze tutorial staan in het code-blok hieronder. Wanneer je dit codeblok uitvoert wordt de aanwezigheid van deze bestanden gecontroleerd.

```
In [1]: from pathlib import Path
from config import dimr_path
import pandas as pd
import numpy as np
import rasterio
```



# Datavalidatie

Waarom plausibele data belangrijk is:

- Model draait überhaupt niet
- Model geeft onbetrouwbare resultaten

Doel: vul je D-HYDRO model met plausibele data

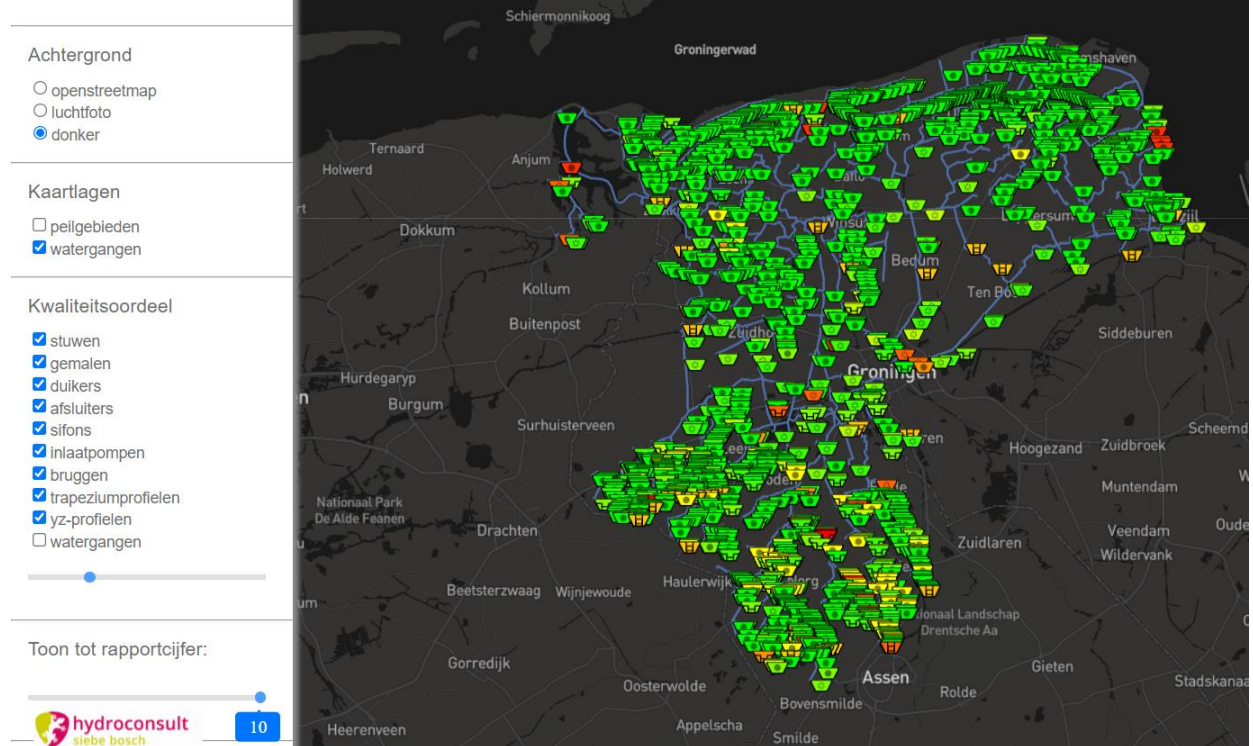
Hoe: SobekTools en Jupyter Notebook

1. *Identificeer* onbetrouwbare data (contextuele validatie)
2. *Gooi* de niet-plausibele data *weg*
3. *Vul* gaten *op* met plausibele defaults (opvullen)



# Datavalidatie: identificeer met SOBEK Channel Builder

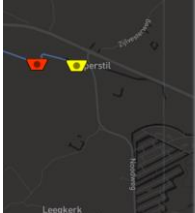
- Alle kunstwerken en dwarsprofielen getoetst aan validatieregels.
- Rapportcijfer + toelichting voor ieder object




# Datavalidatie: identificeer met SOBEK Channel Builder

Dit kom je zoal tegen:

1. Nog veel duikers met hoogte/breedte = 0, BOK=0 of lengte=0
2. Enkele gemalen met ontbrekende capaciteit
3. Veel afsluiters met kruinbreedte = 0
4. Stuwen met kruinbreedte = 0

Validation Rule	Left val	Right val	Success	Penalty	Parameter	Original value	Overruled by
[HoogteStreefpeilBov]-[HoogteStreefpeilBen]=0	0	0	True		[HoogteStreefpeilBen]	-0.93	
[BarrelHeight]-0	2	True			[BarrelHeight]	2	
[BarrelWidth]-0	0	False	4		[BarrelWidth]	0	2
[Length]-0	0	False	4		[Length]	0	10
[InvertUp]jsumnumber	0	True			[InvertUp]	0	
[InvertDown]jsumnumber	0	True			[InvertDown]	0	
[InvertUp]-[Maasveld]	0	NaN	Failed		[InvertUp]	0	
[InvertDown]-[Maasveld]	0	NaN	Failed		[InvertDown]	0	
[InvertUp]-[BarrelHeight]-[Maasveld]	2	NaN	Failed		[InvertUp]	0	
[InvertDown]-[BarrelHeight]-[Maasveld]	2	NaN	Failed		[InvertDown]	0	
[InvertUp]-[BarrelHeight]-[BedLevel]	2	NaN	Failed		[InvertUp]	0	
[InvertDown]-[BarrelHeight]-[BedLevel]	2	NaN	Failed		[InvertDown]	0	



Validation Rule	Left val	Right val	Success	Penalty	Parameter	Original value	Overruled by
[NumberOPumps]-0	1	True			[NumberOPumps]	1	
[NumberOPumps]-7	1	True			[NumberOPumps]	1	
[HinterlandArea]-0	NaN	False			[HinterlandArea]	NaN	100000
[DefaultCapacityMPPD]-0	NaN	False			[DefaultCapacityMPPD]	NaN	12
[CapacityPump1]jsumnumber	NaN	False			[CapacityPump1]	NaN	0
[TotalCapacity]-0	0	False	6		[TotalCapacity]	0	0.14
[CapacityPump1]-0	0	False	1		[CapacityPump1]	0	0.14
[OnMarginPump1]jsumnumber	NaN	False			[OnMarginPump1]	NaN	2
[OffMarginPump1]jsumnumber	NaN	False			[OffMarginPump1]	NaN	-5

# Datavalidatie: weggooien in de notebook

## Weggooiden en opvullen gaat goed in DataFrames

jupyter modelbouw Laatste checkpoint: 18 uur geleden (automatisch opgeslagen)



Logout

File Edit View Insert Cell Kernel Widgets Help

Vertrouwd

Python 3

Code Voilà

```
hydamo.culverts.dropna(axis=0, inplace=True, subset=["branch_offset"])

hydamo.culverts.loc[hydamo.culverts["lengte"].isna(), "lengte"] = 20
hydamo.culverts.loc[hydamo.culverts["allowedflowdir"].isna(), "allowedflowdir"] = "both"

# oplossen fouten in data, duiker op bodem waterloop Leggen als 0
hydamo.culverts['hoogtebinnenonderkantbenedenstrooms'] = hydamo.culverts.apply(
    (lambda x: x['hoogtebinnenonderkantbenedenstrooms']
     if x['hoogtebinnenonderkantbenedenstrooms'] != 0
     else principe_profielen_df.loc[x["branch_id"]]['bottomlevel']),
    axis=1
)

hydamo.culverts['hoogtebinnenonderkantbovenstrooms'] = hydamo.culverts.apply(
    (lambda x: x['hoogtebinnenonderkantbovenstrooms']
     if x['hoogtebinnenonderkantbovenstrooms'] != 0
     else principe_profielen_df.loc[x["branch_id"]]['bottomlevel']),
    axis=1
)
```

Missende lengte opgevuld met 20m

Als de BOK = 0, dan vervangen voor hoogte principeprofiel



# Datavalidatie: Conclusie en aanbeveling

**Conclusie:** Zonder plausibele gegevens geen werkend D-HYDRO model. Data valideren\*, foute data verwijderen en opvullen met plausibele vervangingswaarden.

**Aanbeveling:** Nu geen gegevensvalidatie in D-HyDAMO. Aansluiten op de HWH ValidatieTool en uitbreiden met het automatisch aanvullen van foute data met plausibele opvulwaarden; zie SobekTools

\* Datavalidatie nu gedaan met SobekTools, straks met HyDAMO ValidatieTool?

# Schematiseren

*Waarom:* model rekent niet (efficient)

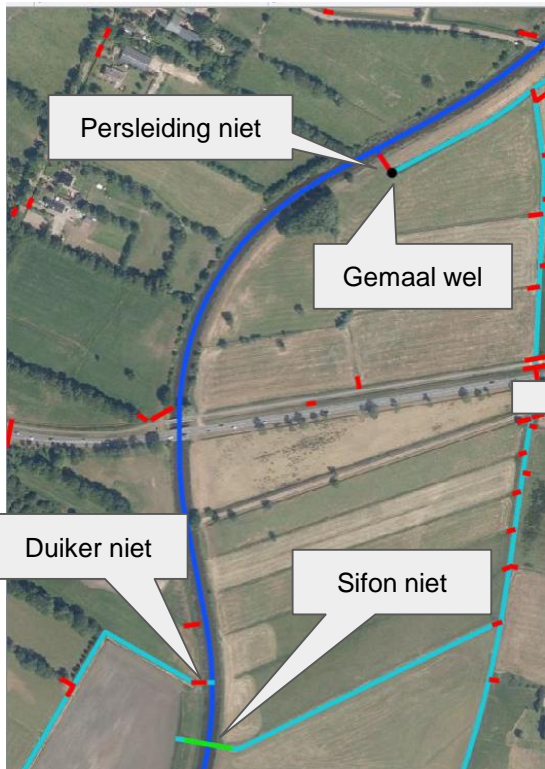
*Doel:* maak je model zo simpel mogelijk (maar niet simpeler)

*Hoe:* extra functies in HydroTools:

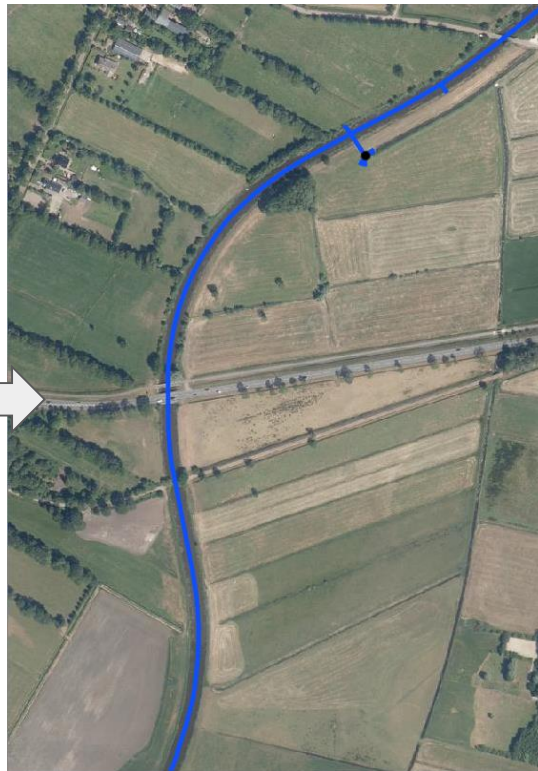
- Complexe filterering
- Toevoeging berging (en RR)
- Vereenvoudiging netwerk

# Schematiseren: complex filteren

DAMO



HyDAMO voor D-HYDRO Model

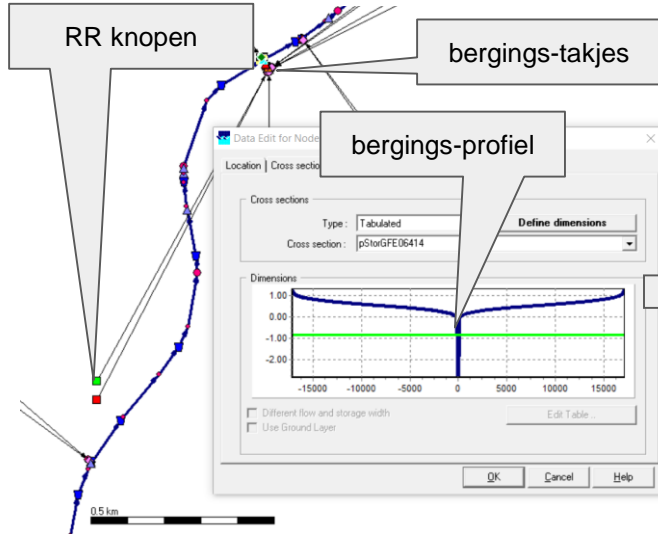


Oplossing:

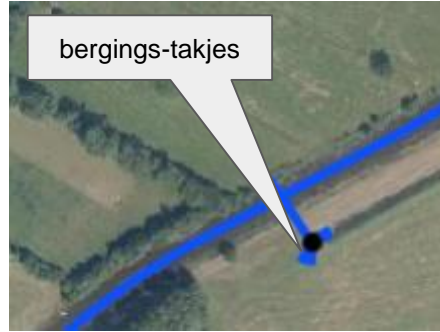
- Geavanceerd filteren met HydroTools
- Snappen met D-HyDAMO
- Snapping-foutjes “hard-code” in Jupyter Notebook

# Schematiseren: toevoegen berging (en RainfallRunoff)

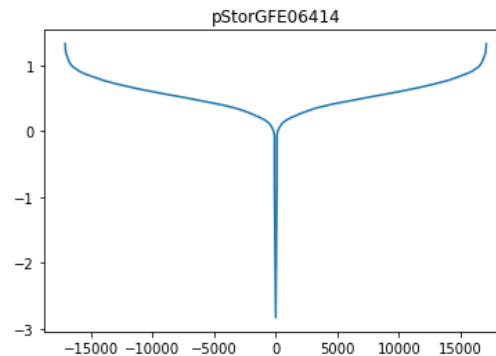
1DRR model in SOBEK



HyDAMO voor D-HYDRO Model Oplossing:



Bergingsprofiel in D-HYDRO Model

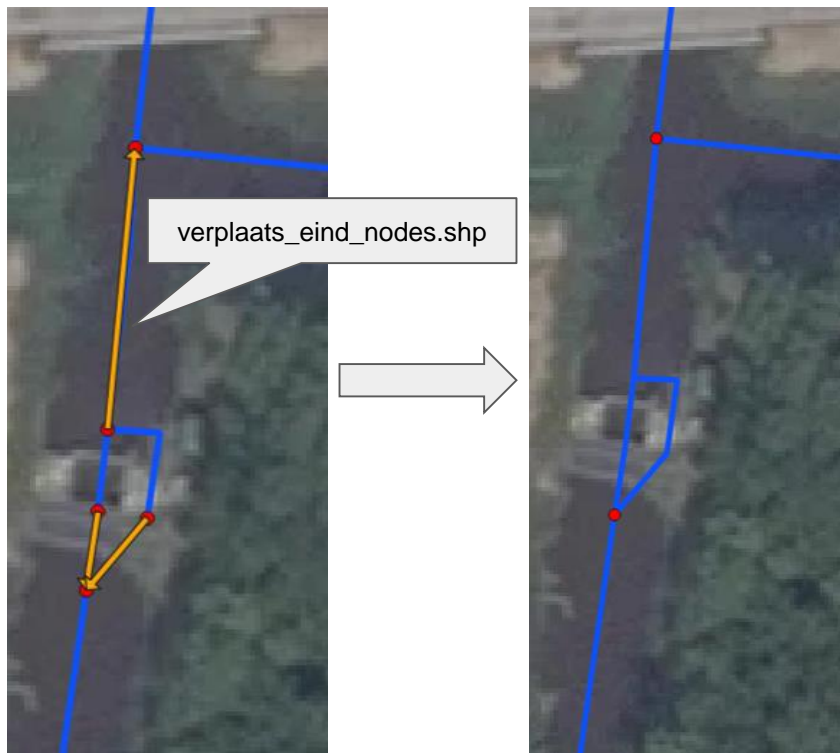


- 1DRR SOBEK model in SobekTools
- SOBEK model inlezen met HydroTools
- Bergings-takken in HyDAMO
- Bergingsprofielen in D-HYDRO Model
- RR in DRRModel

# Schematiseren: vereenvoudiging netwerk

DAMO Afaanvoertakken

HyDAMO voor D-HYDRO Model



Oplossing:

- Modeller maakt een shapefile met node-verplaatsingen
- *move\_end\_nodes* functie in HydroTools bewerkt de DAMO Afaanvoertakken.



# Schematiseren: conclusie & aanbeveling

**Conclusie:** Simplificeren in de stap van (Hy)DAMO topografie naar een D-HYDRO netwerk belangrijk voor het krijgen van (efficiënt) rekenend model.

**Aanbeveling:** Simplificeren in de D-HYDRO Model klasse (als onderdeel van het model) en niet in de HyDAMO klasse.