**Fremgangsmåde til klassifikationer af forurenede grunde, som truer grundvand, samt kobling af forurenede grunde til grundvandsforekomster. Revideret d. 12-12-2024**

Til projektet Grundvandets kemiske påvirkning på målsatte overfladevandområder med miljøfarlige forurenende stoffer samt Grundvandets kemiske påvirkning på målsatte overfladevandområder med miljøfarlige forurenende stoffer – del 3 er der behov for et nyt dataudtræk over forurenede grunde som truer grundvand, samt en kobling af forurenede grunde til grundvandsforekomster. I dette notat er fremgangsmåden til klassifikation forurenede grunde, som truer grundvand, samt kobling af forurenede grunde til grundvandsforekomster beskrevet.

**Klassifikation af forurenede grunde som truer grundvand**

MST har den 27-09-2024 foretaget et ny DK-jord udtræk fra Danmarks Miljøportal. Udtrækket indeholdt .shp filer med alle V1 og V2 kortlagte grund, og desuden .csv filer med informationer om lokation, forurening, branche, aktivitet og forureningsstatus. Udtrækket manglede informationer om aktiviteter, som senere er modtaget af Danmarks Miljøportal.

På baggrund af udtrækket fra DK-jord samt kodelisterne til data fra DK-jord er der foretaget et simpelt join imellem de forskellige koder.

Dissolve (data management) værktøjet blev brugt på .shp filerne med V1 og V2 lokaliteter til at sammenlægge polygoner der tilhører samme lokalitet til en multipart feature, sådan så hver lokalitet ikke er opdelt i hvert enkelt polygon, men at hver unik lokalitet er en multipart feature. Lokalitetsnummer blev valgt som dissolve field og der var sat hak i boksen create multipart feature. Dette resulterede i en ny shapefil med en række for hver lokalitet.

**Kobling af forurenede grunde, som truer grundvand, til grundvandsforekomster**

Kobling af forurenende grunde til grundvandsforekomster er foretaget i ArcGIS, da klassifikationen er foretaget i ArcGIS. Koblingen er en simpel overlapsanalyse og kan udføres i alle GIS-systemer.

I ArcGIS er overlapsanalysen udført efter følgende model:

For at bestemme hvilke forurenede grunde der ligger indenfor hver grundvandsforekomst, blev .shp filerne med de grundvandstruende V1 og V2 lokaliteter brugt sammen med .shp filen med de 2050 grundvandsforekomster anvendt til VP3.

For at koble grundvandsforekomsterne med de grundvandstruende V2 kortlagte lokaliteter, blev Shapefilen med de grundvandstruende V2 lokaliteter joinet med shpfilen med grundvandsforekomsterne. For joinet blev der valgt ”One to many” som join operation, da der kan være flere V2 lokaliteter indenfor en grundvandsforekomst. ”keep all target features” blev slået til. Match option blev sat til intersect.

Den samme overlapsanalyse er foretage for overlap imellem jordforurening og OSD/indvindingsopland udenfor OSD. Her er dog ikke oprettet nye rækker for hvert overlap, men bare markeret med et 1 tal, hvis der henholdsvis er overlap imellem Jordforureningen og OSD eller indvindingsopland udenfor OSD. Dette er markeret i kolonnerne IOLuOSD og OSD.

**Kobling af informationer om lokation, forurening, branche, aktivitet og forureningsstatus for forurenede grunde, som truer grundvand.**

For at koble yderligere informationer til de grundvandsforekomstspecifikke forurenende V1 og V2 lokaliteter er der udarbejdet et python script, vedlagt som Bilag 1.

Det endelige datasæt indeholder således *110814* unikke forurenede grunde, der fordeler sig på følgende Lokalitetetsforureningsstatus:

* *Udgået inden kortlægning 53441*
* *V1 kortlagt 19603*
* *V2 kortlagt 17660*
* *Udgået efter kortlægning 12788*
* *Lokaliseret (uafklaret) 3714*
* *V1 og V2 kortlagt 3608*

**Bilag 1. Python script til kobling af yderligere informationer til grundvandsforekomstspecifikke jordforureninger.**

# -\*- coding: utf-8 -\*-

"""

Spyder Editor

import pandas as pd

# Indlæs CSV-filen

df = pd.read\_csv("dkjord-View\_Lokaliteter - Kopi.csv", sep=';')

# Fjern specifikke kolonner (fx 'kolonne1', 'kolonne2')

df\_cleaned = df.drop(columns=['lokalitetensforureningsflader', 'Id', 'Lokalitetsejerlavkode','Lokalitetsmatrikler','GUID'])

# Split 'Lokalitetensstoffer' column by semicolon and expand into new rows

df\_expanded = df\_cleaned.assign(Lokalitetensstoffer=df['Lokalitetensstoffer'].str.split(';')).explode('Lokalitetensstoffer')

# Fjern dubletter baseret på alle kolonner (kan også specificere kolonner, fx subset=['kolonne1', 'kolonne2'])

df\_unique = df\_expanded.drop\_duplicates()

# Tæl unikke værdier i 'Lokalitetsnr'-kolonnen

unique\_count = df\_unique['Lokalitetsnr'].nunique()

# Print antallet af unikke værdier

print(f"Antal unikke værdier i kolonnen 'Lokalitsnr': {unique\_count}")

# Gem den nye tabel til en ny CSV-fil

df\_unique.to\_csv("expanded\_lokaliteter.csv", index=False)

# Fjern dubletter baseret på alle kolonner (kan også specificere kolonner, fx subset=['kolonne1', 'kolonne2'])

df\_uniquelokalitet = df\_unique.drop\_duplicates(subset=['Lokalitetsnr'])

# Tæl antallet af forekomster af hver unik værdi i kolonnen 'Lokalitsforureningstatus'

value\_counts = df\_uniquelokalitet['Lokalitetetsforureningsstatus'].value\_counts()

# Print resultatet

print(value\_counts)

df2 = pd.read\_csv("V2\_gvfk.csv", sep=';')

# Udfør et mange-til-mange join på 'nøglekolonne'

df\_joined = pd.merge(df2, df\_unique, left\_on='Lokalitets', right\_on='Lokalitetsnr', how='inner')

# Fjern specifikke kolonner (fx 'kolonne1', 'kolonne2')

V2\_gvfk = df\_joined.drop(columns=['OID\_', 'Shape\_Length', 'Shape\_Area','Lokalitete','Lokalitets','Lokalite\_1','Regionsnav','SenesteInd'])