

01_introduksjon

October 12, 2025

```
[1]: # =====
# GOOGLE COLAB SETUP / GOOGLE COLAB SETUP
# =====

# Sjekk om vi kjører i Google Colab
try:
    import google.colab
    IN_COLAB = True
    print(" Kjører i Google Colab - installerer avhengigheter...")
    print(" Running in Google Colab - installing dependencies...")

    # Installer nødvendige pakker
    import subprocess
    import sys
    try:
        subprocess.check_call([sys.executable, "-m", "pip", "install", "-q",
                                "networkx", "matplotlib", "plotly", "pydantic",
                                "pyyaml", "pandas", "ipywidgets", "pillow", ↵
↵"kaleido"])
        print(" Pakker installert")
    except Exception as e:
        print(f" Pip install feilet: {e}")

    # Fjern eksisterende slektstre-mappe hvis den finnes
    import shutil
    import os
    if os.path.exists('/content/slektstre'):
        shutil.rmtree('/content/slektstre')
        print(" Fjernet eksisterende slektstre-mappe")

    # Klon repository
    try:
        subprocess.check_call(['git', 'clone', 'https://github.com/arvidl/
↵slektstre.git'])
        print(" Repository klonet")
    except Exception as e:
        print(f" Git clone feilet: {e}")
```

```

# Legg til src-mappen til Python path og importer direkte
sys.path.insert(0, '/content/slektstre/src')
print(" Path lagt til")

# Importer slektstre-modulene direkte for å unngå navnekonflikt
import importlib.util
import types

# Først, fjern konfliktende moduler fra sys.modules
modules_to_remove = ['tree', 'models', 'localization']
for module_name in modules_to_remove:
    if module_name in sys.modules:
        del sys.modules[module_name]

# Last inn models.py først
try:
    spec = importlib.util.spec_from_file_location("slektstre_models", "/
↪content/slektstre/src/models.py")
    slektstre_models = importlib.util.module_from_spec(spec)
    spec.loader.exec_module(slektstre_models)

    # Opprett midlertidig models modul
    temp_models_module = types.ModuleType('models')
    temp_models_module.Person = slektstre_models.Person
    temp_models_module.Gender = slektstre_models.Gender
    temp_models_module.Ekteskap = slektstre_models.Ekteskap
    temp_models_module.FamilieData = slektstre_models.FamilieData
    sys.modules['models'] = temp_models_module

    print(" models.py lastet")
except Exception as e:
    print(f" models.py feilet: {e}")

# Last inn localization.py
try:
    spec = importlib.util.spec_from_file_location("slektstre_localization",
↪"/content/slektstre/src/localization.py")
    slektstre_localization = importlib.util.module_from_spec(spec)
    spec.loader.exec_module(slektstre_localization)

    # Opprett midlertidig localization modul
    temp_localization_module = types.ModuleType('localization')
    temp_localization_module.t = slektstre_localization.t
    sys.modules['localization'] = temp_localization_module

    print(" localization.py lastet")

```

```

except Exception as e:
    print(f" localization.py feilet: {e}")

# Last inn tree.py som slektstre_tree
try:
    spec = importlib.util.spec_from_file_location("slektstre_tree", "/
↪content/slektstre/src/tree.py")
    slektstre_tree = importlib.util.module_from_spec(spec)
    spec.loader.exec_module(slektstre_tree)

    # Opprett midlertidig tree modul
    temp_tree_module = types.ModuleType('tree')
    temp_tree_module.Slektstre = slektstre_tree.Slektstre
    sys.modules['tree'] = temp_tree_module

    print(" tree.py lastet")
except Exception as e:
    print(f" tree.py feilet: {e}")

# Last inn family_io.py
try:
    spec = importlib.util.spec_from_file_location("slektstre_io", "/content/
↪slektstre/src/family_io.py")
    slektstre_io = importlib.util.module_from_spec(spec)
    spec.loader.exec_module(slektstre_io)
    print(" family_io.py lastet")
except Exception as e:
    print(f" family_io.py feilet: {e}")

# Last inn visualization.py
try:
    spec = importlib.util.spec_from_file_location("slektstre_viz", "/
↪content/slektstre/src/visualization.py")
    slektstre_viz = importlib.util.module_from_spec(spec)
    spec.loader.exec_module(slektstre_viz)
    print(" visualization.py lastet")
except Exception as e:
    print(f" visualization.py feilet: {e}")

print(" Slektstre-moduler lastet inn i Colab")

except ImportError:
    IN_COLAB = False
    print(" Kjører lokalt / Running locally")
    import sys
    sys.path.append('../src')
except Exception as e:

```

```

print(f" Colab setup feilet: {e}")
IN_COLAB = False
print(" Fallback til lokal modus / Fallback to local mode")
import sys
sys.path.append('../src')

print(f" Miljø: {'Google Colab' if IN_COLAB else 'Lokal'}")
print(f" Environment: {'Google Colab' if IN_COLAB else 'Local'}")

```

Kjører lokalt / Running locally
Miljø: Lokal
Environment: Local

1 Slektstre med NetworkX - Introduksjon

Velkommen til slektstre-prosjektet! Dette er en komplett løsning for å bygge, administrere og visualisere familie-trær ved hjelp av NetworkX og Python.

1.1 Hva er dette prosjektet?

Slektstre-prosjektet lar deg: - Bygge komplekse familie-trær med rike metadata - Importere og eksportere data i flere formater (YAML, JSON, CSV, GEDCOM) - Visualisere slektstreet på forskjellige måter - Støtte både norsk og engelsk språk - Analysere slektskap og generasjonsforhold

1.2 Hovedkomponenter

1. **Modeller** (models.py): Pydantic-modeller for Person, Ekteskap og FamilieData
2. **Slektstre-klasse** (tree.py): Hovedklasse med NetworkX som backend
3. **Import/Eksport** (io.py): Støtte for flere dataformater
4. **Visualisering** (visualization.py): Matplotlib og Plotly visualiseringer
5. **Lokalisering** (localization.py): Tospråklig støtte

1.3 Installasjon

Først må du sette opp conda-miljøet:

```
conda env create -f environment.yml
conda activate slektstre
```

Eller installere pakkene direkte:

```
pip install -r requirements.txt
```

```

[2]: # Importer nødvendige biblioteker
import matplotlib.pyplot as plt
import plotly.express as px
from datetime import date
import pandas as pd

# Importer slektstre-moduler (fungerer både lokalt og i Colab)

```

```

if IN_COLAB:
    # Bruk de modulene vi lastet inn i Colab-setup
    Person = slektstre_models.Person
    Gender = slektstre_models.Gender
    Ekteskap = slektstre_models.Ekteskap
    FamilieData = slektstre_models.FamilieData
    Slektstre = slektstre_tree.Slektstre
    load_from_yaml = slektstre_io.load_from_yaml
    save_to_yaml = slektstre_io.save_to_yaml
    plot_hierarchical_tree = slektstre_viz.plot_hierarchical_tree
    plot_interactive_tree = slektstre_viz.plot_interactive_tree
    plot_statistics = slektstre_viz.plot_statistics
    t = slektstre_localization.t
    get_available_languages = slektstre_localization.get_available_languages
else:
    # Lokale imports
    import sys
    sys.path.append('../src')
    from models import Person, Ekteskap, FamilieData, Gender
    from tree import Slektstre
    from family_io import load_from_yaml, save_to_yaml
    from visualization import plot_hierarchical_tree, plot_interactive_tree,
    plot_statistics
    from localization import t, get_available_languages

print(" Alle biblioteker importert!")
print(f"Tilgjengelige språk: {get_available_languages()}")

```

```

Alle biblioteker importert!
Tilgjengelige språk: ['no', 'en']

```

1.4 Grunnleggende begreper

1.4.1 Person-modellen

En **Person** har følgende hovedattributter: - **Navn**: fornavn, mellomnavn, etternavn - **Metadata**: fødselsdato, dødsdato, fødested, kjønn - **Relasjoner**: foreldre, barn, partnere - **Media**: bilde_sti, notater, historier

1.4.2 Ekteskap-modellen

Et **Ekteskap** kobler to personer sammen: - **Partnere**: referanser til to person-IDer - **Datoer**: ekteskapsdato, skilsmisse_dato - **Metadata**: ekteskapssted, type, notater

1.4.3 Slektstre-klassen

Slektstre er hovedklassen som: - Bruker NetworkX som backend for graf-operasjoner - Tilbyr metoder for å legge til/fjerne personer og relasjoner - Beregner slektskap og generasjonsnivåer - Gir statistikk om familien

```
[3]: # Test: Opprett en enkel person
person = Person(
    fornavn="Arvid",
    etternavn="Lundervold",
    kjønn=Gender.MALE,
    fødselsdato=date(1985, 12, 10),
    fødested="Bergen",
    notater="Forsker i kunstig intelligens"
)

print(f"Person opprettet: {person.fullt_navn}")
print(f"Alder: {person.alder} år")
print(f"Er levende: {person.er_levende}")
print(f"Kjønn: {t(person.kjønn)}")
```

```
Person opprettet: Arvid Lundervold
Alder: 39 år
Er levende: True
Kjønn: Mann
```

1.5 Last eksempeldata

La oss laste inn eksempel-familien som følger med prosjektet:

```
[4]: # Last eksempel-familie
if IN_COLAB:
    familie_data = load_from_yaml('/content/slektstre/data/eksempel_familie.
    ↳yaml')
else:
    familie_data = load_from_yaml('../data/eksempel_familie.yaml')

slektstre = Slektstre(familie_data)

print(f"Familie lastet med {len(familie_data.personer)} personer og
    ↳{len(familie_data.ekteskap)} ekteskap")
print(f"Beskrivelse: {familie_data.beskrivelse}")

# Vis noen personer
print("\nFørste 5 personer:")
for person in familie_data.personer[:5]:
    print(f"- {person.fullt_navn} ({person.fødselsdato.year if person.
    ↳fødselsdato else 'Ukjent år'})")
```

```
Familie lastet med 17 personer og 5 ekteskap
Beskrivelse: Eksempel familie med 4 generasjoner - Lundervold familien
```

```
Første 5 personer:
- Erik Lundervold (1920)
```

- Ingrid Marie Hansen (1925)
- Arvid Lundervold (1950)
- Helena Sofia Lundervold (1952)
- Bjørn Lundervold (1955)

1.6 Test visualisering

La oss teste en enkel visualisering:

1.7 Forklaring av visualiseringen

Kantene (linjene) mellom nodene representerer:

- **Røde linjer (tykke):** Ekteskap/partnerskap mellom to personer
- **Svarte linjer (tykke):** Forelder-barn relasjoner
- **Svarte linjer (tynne, stiplede):** Andre slektskap (f.eks. søsken)

Farger på nodene: - **Blå:** Menn - **Rosa:** Kvinner - **Grønn:** Annet kjønn

Layout: - Personer er arrangert etter generasjoner (vertikalt) - Eldre generasjoner er øverst - Årstallene viser fødselsår - **ID-en (p-nummeret) vises inne i hver node** for lettere identifikasjon

```
[5]: # Vis alle personer med deres p-nummer for lettere identifikasjon
print(" Alle personer i slektstreet:")
print("=" * 50)

for person in slektstre.get_all_persons():
    fødselsår = person.fødselsdato.year if person.fødselsdato else "Ukjent"
    print(f"ID: {person.id:3} | {person.fullt_navn:25} | f. {fødselsår} | ♂
    ↪{t(person.kjønn)}")

print(f"\n Totalt: {len(slektstre.get_all_persons())} personer")
```

Alle personer i slektstreet:

```
=====
ID: p1  | Erik Lundervold          | f. 1920 | Mann
ID: p2  | Ingrid Marie Hansen         | f. 1925 | Kvinne
ID: p3  | Arvid Lundervold            | f. 1950 | Mann
ID: p4  | Helena Sofia Lundervold     | f. 1952 | Kvinne
ID: p5  | Bjørn Lundervold            | f. 1955 | Mann
ID: p6  | Kari Lundervold             | f. 1958 | Kvinne
ID: p8  | Anna Kristin Pedersen       | f. 1952 | Kvinne
ID: p7  | Magnus Lundervold           | f. 1980 | Mann
ID: p9  | Erik Arvid Lundervold       | f. 1985 | Mann
ID: p10 | Sofia Lundervold            | f. 1988 | Kvinne
ID: p11 | Lars Andersen               | f. 1983 | Mann
ID: p12 | Marianne Olsen              | f. 1985 | Kvinne
ID: p17 | Ole Andersen                | f. 1950 | Mann
ID: p13 | Emma Lundervold             | f. 2010 | Kvinne
ID: p14 | Noah Lundervold             | f. 2012 | Mann
```

ID: p15 | Maja Lundervold | f. 2015 | Kvinne
ID: p16 | Oliver Lundervold | f. 2018 | Mann

Totalt: 17 personer

```
[6]: # Vis alle ekteskap
print(" Alle ekteskap i slektstreet:")
print("=" * 60)

for ekteskap in slektstre.familie_data.ekteskap:
    partner1 = slektstre.get_person(ekteskap.partner1_id)
    partner2 = slektstre.get_person(ekteskap.partner2_id)

    if partner1 and partner2:
        ekteskapsår = ekteskap.ekteskapsdato.year if ekteskap.ekteskapsdato
    else "Ukjent"
    status = "Aktivt" if ekteskap.er_aktivt else "Skilt"
    print(f"ID: {ekteskap.id:3} | {partner1.fullt_navn:20} {partner2.
    fullt_navn:20} | Gift {ekteskapsår} | {status}")

print(f"\n Totalt: {len(slektstre.familie_data.ekteskap)} ekteskap")
```

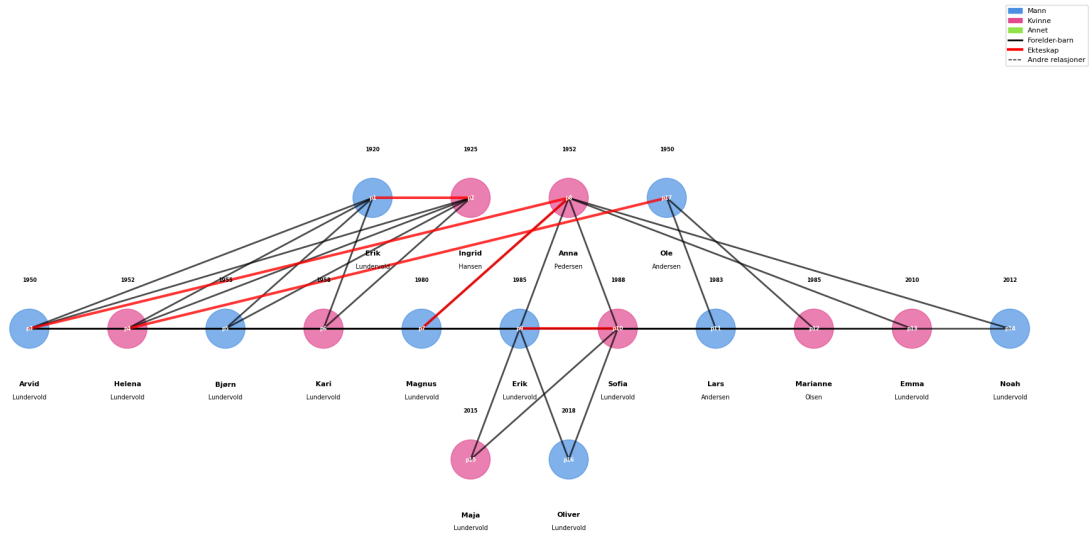
Alle ekteskap i slektstreet:

```
=====
ID: e1 | Erik Lundervold Ingrid Marie Hansen | Gift 1947 | Aktivt
ID: e2 | Arvid Lundervold Anna Kristin Pedersen | Gift 1978 | Aktivt
ID: e3 | Helena Sofia Lundervold Ole Andersen | Gift 1980 | Skilt
ID: e4 | Magnus Lundervold Anna Kristin Pedersen | Gift 2005 | Aktivt
ID: e5 | Erik Arvid Lundervold Sofia Lundervold | Gift 2010 | Aktivt
```

Totalt: 5 ekteskap

```
[7]: # Test hierarkisk slektstre
fig = plot_hierarchical_tree(slektstre, title="Lundervold familien")
plt.show()
```


Lundervold familien



[]:

[]: