


# Slektstre med NetworkX


Et Python-bibliotek for å bygge, administrere og visualisere familie-trær ved hjelp av NetworkX og Jupyter notebooks.




## Podcast / Lydinnhold


### Slektstre med Python og Grafteori - Slik Analyserer du Din Familie

 **Last ned podcast-filer:** - [MP3 \(universell kompatibilitet\)](#) - 20MB - [M4A \(høy kvalitet\)](#) - 40MB

 **For å spille av:** 1. Klikk på lenken ovenfor for å laste ned 2. Åpne med din foretrukne lydspiller 3. Eller høre direkte i nettleseren (støtter MP3/M4A)

 **Tips:** Du kan også høre på podcasten direkte i Jupyter Notebook ved å bruke:

```
import IPython.display as ipd
# Fra repository-rot:
ipd.Audio('podcast/Slektstre_med_Python_og_Grafteori__Slik_Analyserer_du_Din_Famil.mp3',
          embed=False)
# Eller fra notebooks/-mappen:
# ipd.Audio('../podcast/Slektstre_med_Python_og_Grafteori__Slik_Analyserer_du_Din_Famil.mp3',
#           embed=False)
```

 **Eller åpne [notebook 00\\_slektstraer\\_og\\_grafer.ipynb](#) som har en forbedret audio-spiller med:** - Automatisk path-deteksjon (fungerer både lokalt og i Google Colab) - HTML5 audio som fallback - **Enhanced cleanup** for å unngå store notebook-filer (fjerner embedded audio/bilder) - Bilingual instruksjoner og feedback

### Enhanced Notebook Cleanup






Notebooks kan bli store på grunn av embedded audio/bilde-data. Vi har laget en forbedret cleanup-løsning:

**Automatisk cleanup i notebook:** - Kjør cleanup-cellen i notebooken før commit/push - Fjerner automatisk embedded audio og bilder - Reduserer filstørrelse med opptil 99% (fra 27MB til 43KB)

#### Manuell cleanup fra kommandolinje:

```
# Bruk det forbedrede cleanup-scriptet
python scripts/enhanced_notebook_cleanup.py
      notebooks/00_slektstraer_og_grafer.ipynb

# Eller bruk bash-scriptet
./scripts/cleanup_notebook.sh notebooks/00_slektstraer_og_grafer.ipynb
```








**Hva cleanup gjør:** -  Fjerner alle cell outputs -  Fjerner embedded base64 audio-data -  Fjerner embedded base64 bilder -  Beholder notebook-strukturen intakt -  Audio-spilleren fungerer fortsatt når notebooken kjøres på nytt

# Norsk / English

## Norsk (Hovedspråk)

Slektstre-prosjektet lar deg bygge komplekse familie-trær med rike metadata, importere/eksportere data i flere formater, og visualisere slektskap på forskjellige måter. Prosjektet støtter både norsk og engelsk språk.

### Hovedfunksjoner

-  **Rike metadata:** Fødselsdatoer, steder, bilder, historier og notater
-  **Fleksibel datainput:** Manuell programmatisk opprettelse og fil-basert import
-  **Avanserte visualiseringer:** Hierarkisk, vifte-diagram, interaktiv og timeglass-visning
-  **Tospråklig støtte:** Norsk (primær) og engelsk
-  **Flere dataformater:** YAML, JSON, CSV og GEDCOM
-  **Slektsanalyse:** Beregning av slektskap, generasjonsnivåer og statistikk
-  **Eksterne databaser:** Integrasjon med FamilySearch, Digitalarkivet og Wikipedia API

### Installasjon

```
# Opprett conda-miljø (anbefalt)
conda env create -f environment.yml
conda activate slektstre
```

```
# Eller installer pakker direkte
pip install -r requirements.txt
```

```
# For kun bok-generering
pip install -r requirements-book.txt
```

### Rask start

```
# Importer modulene direkte fra src-mappen
import sys
sys.path.append('src')

from models import Person, Gender
from tree import Slektstre
from datetime import date

# Opprett personer
person = Person(
    fornavn="Arvid",
    etternavn="Lundervold",
    kjønn=Gender.MALE,
    fødselsdato=date(1985, 12, 10),
    fødested="Bergen"
)

# Opprett slektstre
slektstre = Slektstre()
slektstre.add_person(person)


# Visualiser
from visualization import plot_hierarchical_tree
import matplotlib.pyplot as plt
```

```
fig = plot_hierarchical_tree(slektstre)
plt.show()
```

## Jupyter Notebooks







Prosjektet inkluderer seks omfattende notebooks:


0. **00\_slektstraer\_og\_grafer.ipynb** - Introduksjon til slektstrær og grafteori
1. **01\_introduksjon.ipynb** - Oversikt og grunnleggende konsepter
2. **02\_bygg\_tre\_manuelt.ipynb** - Bygge slektstreet programmatisk
3. **03\_importer\_data.ipynb** - Import/eksport av data
4. **04\_visualisering.ipynb** - Alle visualiseringsalternativer
5. **05\_eksterne\_databaser.ipynb** - Integrasjon med genealogi-databaser og API-er

 **Nytt: 00\_slektstraer\_og\_grafer.ipynb** er en omfattende introduksjonsnotebook som kobler sammen genealogi og grafteori. Den dekker grunnleggende konsepter, praktiske øvelser, og viser hvordan NetworkX brukes til å bygge og analysere slektstrær. Perfekt for nybegynnere som vil forstå både slektstrær og den underliggende matematikken.

## Åpne i Google Colab / Open in Google Colab

Alle notebooks kan kjøres direkte i Google Colab uten installasjon:

-  [Open in Colab](#) **00\_slektstraer\_og\_grafer.ipynb** - Introduksjon til slektstrær og grafteori
-  [Open in Colab](#) **01\_introduksjon.ipynb** - Oversikt og grunnleggende konsepter
-  [Open in Colab](#) **02\_bygg\_tre\_manuelt.ipynb** - Bygge slektstreet programmatisk
-  [Open in Colab](#) **03\_importer\_data.ipynb** - Import/eksport av data
-  [Open in Colab](#) **04\_visualisering.ipynb** - Alle visualiseringsalternativer
-  [Open in Colab](#) **05\_eksterne\_databaser.ipynb** - Integrasjon med genealogi-databaser og API-er

 **Tips:** Klikk på Colab-badgen for å åpne notebooken direkte i Google Colab. Alle avhengigheter installeres automatisk!

## Dataformat

Slektstreet støtter flere formater:

### YAML (anbefalt):

```
personer:
- id: "p1"
  fornavn: "Arvid"
  etternavn: "Lundervold"
  kjønn: "male"
  fødselsdato: "1985-12-10"
  fødested: "Bergen"
ekteskap:
- partner1_id: "p1"
  partner2_id: "p2"
  ekteskapsdato: "2010-06-18"
```

**JSON, CSV og GEDCOM** støttes også.

## Visualiseringer

- **Hierarkisk slektstre:** Tradisjonell tre-struktur
- **Vifte-diagram:** Sirkulær visning av forfedre
- **Interaktiv visning:** Plotly-basert med hover-info

- **Timeglass-visning:** Fokuspersion i midten
- **Statistikk-diagrammer:** Kjønnfordeling, aldersfordeling, etc.

## Eksterne databaser

- **FamilySearch API:** Verdens største genealogi-database (gratis)
- **Digitalarkivet:** Norske historiske kilder og arkiver (gratis)
- **Wikipedia API:** Biografisk informasjon om kjente personer (gratis)
- **Data-konvertering:** Automatisk konvertering fra eksterne formater
- **Eksport:** Til forskjellige formater for deling med andre

## English

The Slektstre project allows you to build complex family trees with rich metadata, import/export data in multiple formats, and visualize relationships in various ways. The project supports both Norwegian and English languages.



## Podcast / Audio Content

### Slektstre med Python og Grafteori - Slik Analyserer du Din Familie



**Note:** Since this is a private repository, you need to download the podcast files locally to play them.



**Download podcast files:** - [MP3 \(universal compatibility\)](#) - 20MB - [M4A \(high quality\)](#) - 40MB



**Note:** Podcast files are not included in the Git repository due to size. You need to download them separately or generate them locally.



**To play:** 1. Download the file to your computer 2. Open with your preferred audio player 3. Or use an online audio player that supports local files



**Tip:** You can also listen to the podcast directly in Jupyter Notebook using:

```
import IPython.display as ipd
ipd.Audio('podcast/Slektstre_med_Python_og_Grafteori__Slik_Analyserer_du_Din_Famil.mp3')
```

## Key Features

- **Rich metadata:** Birth dates, places, photos, stories and notes
- **Flexible data input:** Manual programmatic creation and file-based import
- **Advanced visualizations:** Hierarchical, fan chart, interactive and hourglass views
- **Bilingual support:** Norwegian (primary) and English
- **Multiple data formats:** YAML, JSON, CSV and GEDCOM
- **Family analysis:** Relationship calculation, generation levels and statistics
- **External databases:** Integration with FamilySearch, Digitalarkivet and Wikipedia API

## Installation

*# Create conda environment (recommended)*

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

*# Or install packages directly*

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

```
# For book generation only
```

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

## Quick Start

```
# Import modules directly from src folder
```

```
import sys
sys.path.append('src')
```

```
from models import Person, Gender
from tree import Slektstre
from datetime import date
```

```
# Create person
```

```
person = Person(
    fornavn="Arvid",
    etternavn="Lundervold",
    kjønn=Gender.MALE,
    fødselsdato=date(1985, 12, 10),
    fødested="Bergen"
)
```

```
# Create family tree
```

```
slektstre = Slektstre()
slektstre.add_person(person)
```

```
# Visualize
```


```
from visualization import plot_hierarchical_tree
import matplotlib.pyplot as plt
```

```
fig = plot_hierarchical_tree(slektstre)
plt.show()
```

## Jupyter Notebooks

The project includes six comprehensive notebooks:

0. **00\_slektstraer\_og\_grafer.ipynb** - Introduction to family trees and graph theory
1. **01\_introduksjon.ipynb** - Overview and basic concepts
2. **02\_bygg\_tre\_manuelt.ipynb** - Building family trees programmatically
3. **03\_importer\_data.ipynb** - Data import/export
4. **04\_visualisering.ipynb** - All visualization options
5. **05\_eksterne\_databaser.ipynb** - Integration with genealogy databases and APIs

 **New: 00\_slektstraer\_og\_grafer.ipynb** is a comprehensive introductory notebook that bridges genealogy and graph theory. It covers fundamental concepts, practical exercises, and shows how NetworkX is used to build and analyze family trees. Perfect for beginners who want to understand both family trees and the underlying mathematics.

## Data Format

The family tree supports multiple formats:

### YAML (recommended):

```
personer:
- id: "p1"
  fornavn: "Arvid"
  etternavn: "Lundervold"
  kjønn: "male"
```

```
fødselsdato: "1985-12-10"
fødested: "Bergen"
ekteskap:
  - partner1_id: "p1"
    partner2_id: "p2"
    ekteskapsdato: "2010-06-18"
```

**JSON, CSV and GEDCOM** are also supported.

### Visualizations

- **Hierarchical family tree:** Traditional tree structure
- **Fan chart:** Circular ancestor view
- **Interactive view:** Plotly-based with hover info
- **Hourglass view:** Focus person in center
- **Statistics charts:** Gender distribution, age distribution, etc.

### External Databases

- **FamilySearch API:** World's largest genealogy database (free)
- **Digitalarkivet:** Norwegian historical sources and archives (free)
- **Wikipedia API:** Biographical information about notable people (free)
- **Data conversion:** Automatic conversion from external formats
- **Export:** To various formats for sharing with others

## Teknisk informasjon / Technical Information

### Avhengigheter / Dependencies

**Core slektstre packages:** - Python 3.12+ - NetworkX 3.0+ - Matplotlib 3.7+ - Plotly 5.0+ - Pydantic 2.0+ - PyYAML 6.0+ - Pandas 2.0+ - Jupyter 1.0+

**Book generation packages:** - Jupyter Book 0.15+ - Sphinx 5.0+ - nbconvert[webpdf] 7.0+ - Playwright 1.55+ - PyPDF2 3.0+ - Pandoc 3.0+

### Prosjektstruktur / Project Structure

```
slektstre/
├── src/                                # Kildekode / Source code
│   ├── __init__.py
│   ├── models.py                      # Pydantic modeller / Models
│   ├── tree.py                        # Slekststre-klasse / Main class
│   ├── family_io.py                  # Import/eksport / I/O functions
│   ├── visualization.py              # Visualisering / Visualization
│   └── localization.py               # Lokalisering / Localization
├── notebooks/                         # Jupyter notebooks
├── data/                              # Eksempeldata / Sample data
├── assets/                            # Bilder og media / Images and media
├── environment.yml                    # Conda miljø / Conda environment
├── requirements.txt                   # Python pakker / Python packages
└── README.md                          # Dokumentasjon / Documentation
```

### Lisens / License

MIT License - se LICENSE fil for detaljer / see LICENSE file for details.

## Håndtering av sensitive data / Handling Sensitive Data

For detaljert veiledning om å håndtere private familie-data, se [SENSITIVE\\_DATA\\_GUIDE.md](#).


**For detailed guidance on managing private family data, see [SENSITIVE\\_DATA\\_GUIDE.md](#).**


En privat repository template er tilgjengelig i `templates/private-repo/`.

**A private repository template is available in `templates/private-repo/`.**

## Public Repository / Offentlig Repository

Dette er nå et **public repository** som alle kan se og bruke. Dette betyr:

 **Fordeler:** - Alle kan teste notebooks i Google Colab uten autentisering - Bedre muligheter for å dele kunnskap og lære sammen - Open source-utvikling og bidrag fra samfunnet - Enklere å finne og bruke for nye brukere

 **Viktig:** - Ingen sensitive familie-data er inkludert i dette repositoryet - Kun syntetiske eksempler og læringsressurser - For ekte familie-data, bruk private repository-malen

## Bidrag / Contributing

Bidrag er velkommen! Se [DEVELOPER.md](#) for detaljerte instruksjoner.

**Hvordan bidra:** 1. Fork repositoryet 2. Lag en feature branch (`git checkout -b feature/ny-funksjon`) 3. Test endringene dine i Google Colab 4. Send en pull request

**Contributions are welcome! See [DEVELOPER.md](#) for detailed instructions.**

**How to contribute:** 1. Fork the repository 2. Create a feature branch (`git checkout -b feature/new-feature`) 3. Test your changes in Google Colab 4. Submit a pull request

## Generer PDF-bok / Generate PDF Book

Du kan generere en komplett PDF-bok av hele prosjektet:

**You can generate a complete PDF book of the entire project:**

```
# Enkel metode / Simple method
make book
```

```
# Eller manuelt / Or manually
pip install -r requirements-book.txt
jupyter-book build . --builder pdfhtml
```

Boken vil inkludere README.md som introduksjon, fulgt av alle notebooks som separate kapitler.

**The book will include README.md as introduction, followed by all notebooks as separate chapters.**

## Bok-redigering / Book Editing

For detaljerte instruksjoner om hvordan du redigerer boken, legger til kapitler, eller setter opp automatisk oppdatering, se [BOK-REDIGERING.md](#).

**For detailed instructions on how to edit the book, add chapters, or set up automatic updates, see [BOK-REDIGERING.md](#).**

## Rask start for bok-redigering / Quick start for book editing

```
# Valider eksisterende bok / Validate existing book  
make validate
```

```
# Start automatisk overvåkning / Start automatic monitoring  
make watch
```

```
# Generer bok på nytt / Regenerate book  
make book
```

## Utviklingsverktøy / Development Tools

For detaljerte instruksjoner om utvikling og repository-vedlikehold, se [DEVELOPER.md](#).

**For detailed instructions on development and repository maintenance, see [DEVELOPER.md](#).**

### Hurtigreferanse / Quick reference:

```
# Fjern store filer / Remove large files  
pip install git-filter-repo  
git filter-repo --path large_file.ipynb --invert-paths --force  
  
# Rydd notebook outputs / Clean notebook outputs  
jupyter nbconvert --clear-output --inplace notebook.ipynb
```

## Kontakt / Contact

Arvid Lundervold - [GitHub](#)