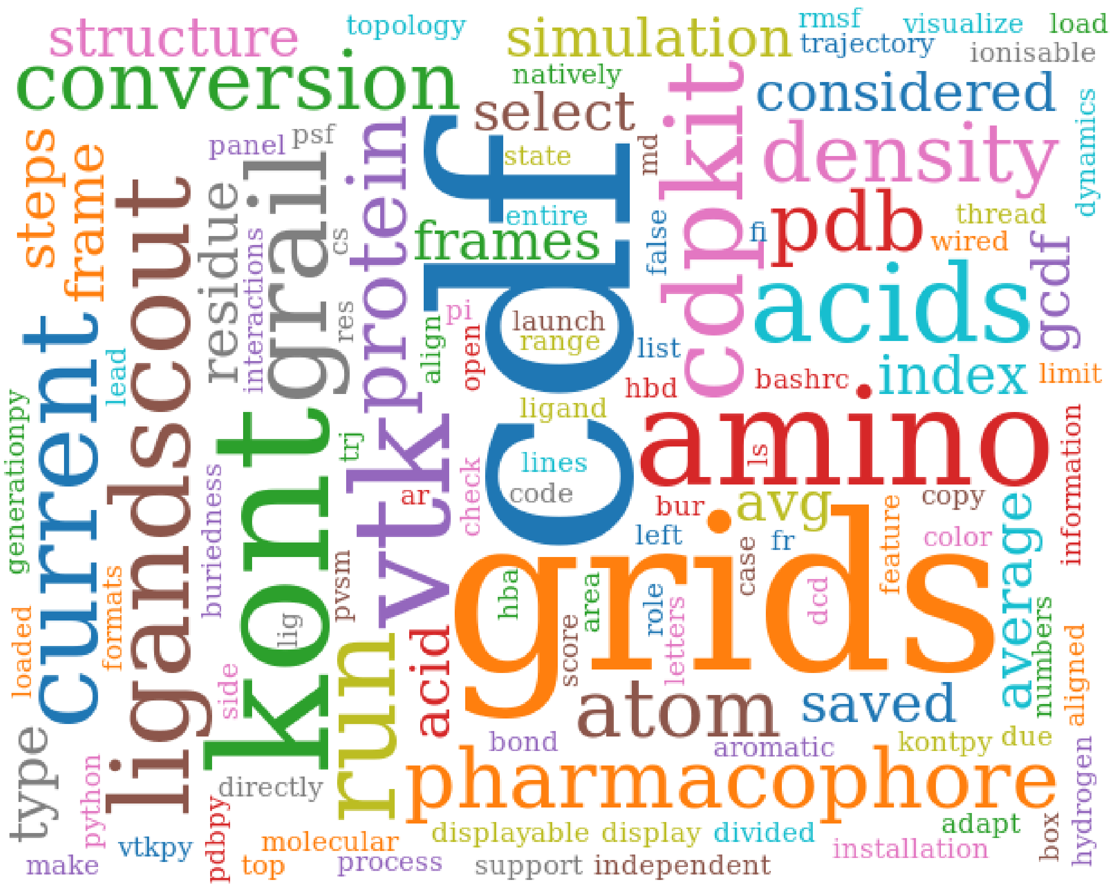
GRAIL Generation Tutorial

The GRids of phArmacophore Interaction fieLds (GRAIL) is a grid based approach that allows to display potential pharmacophore feature interactions. A single structure or a molecular dynamics simulation is used to generate the GRAIL grids, one for each type of interaction and one per frame. The type of interaction (like HBA-HBD) is characterized by the type of feature on the virtual ligand side and the type of feature on the protein side. Six types of pharmacophore features are considered: HBD for hydrogen bond donor, HBA for hydrogen Bond Acceptor, H for Hydrophobic interactions, PI for positive ionisable area, NI for negative ionisable area and AR for aromatic ring. This results in 8 types of interactions. Only the protein is considered for the generation of the grids, ligands are ignored.
For visualization, grids can be output in 2 different formats: kont (GRID format) and/or vtk (ParaView format). The kont format can be loaded with LigandScout, and the vtk with ParaView.



Towards Next Generation Pharmacophore Models
Arthur Garon
September 2018

universität wien   SERVIER   inte:ligand Your partner for in-silico drug discovery.

GRAIL Generation Tutorial

⬡ 1. Prerequisites

- Python version 2.7
- Python dependencies:
  - You must be able to start a python session and successfully run the `import_check.py` script

➢ If the CDPL imports are not working

- Please make sure the location of the CDPKIT installation is included in your PYTHONPATH by following these steps:
  - Open the .bashrc file, located in your home directory
  - Check that these two lines are present:
    - `PYTHONPATH=$PYTHONPATH:/path/of/the/CDPKIT/installation/Python`
    - `export PYTHONPATH`
  - In case of modification of the .bashrc file, launch another terminal

➢ Each script can be launched with the -h command to display a quick description of the script's role as well as details of the supported arguments:

- `python script_im_interested_in.py - h`

⬡ 2. Grid set generation from a PDB structure

The Grail generation process is divided in **3 independent steps**:

- Generation of the cdf file ─ cdf, CDPKIT format ─ from a PDB file
- Generation of the interaction and atom density grids ─ grid.cdf, CDPKIT format ─ from both the cdf file and an amino acid residue list
- Generation of files for visualisation ─ kont, GRID format / vtk, ParaView format ─ from the grid.cdf

These 3 steps are explained in detail below but can be done in a **single script**, see the **next section**.

➢ 2.1. Step 1: cdf file generation

This script generates a cdf file (CDPKIT format) and a text file listing all residues from a PDB file.

- Run the `cdf_generation_PDB.py` script
- It requires several arguments:
  - [Required] `-pdb` The path of the PDB file
  -
  - [Optional] `-o` The output folder where the cdf file will be generated (Default: current directory)
  - [Optional] `-n` The name of the cdf file (Default: name of the pdb file)

You can **manually adapt the list of residues**. A grid based on these amino acids will then be automatically generated during the next step. **The higher the number of amino acids considered, the longer the generation of the grid will take.**

➢ 2.2. Step 2: Grid generation

This script generates a grid.cdf file (CDPKIT format) from both an aligned cdf and a text file listing the residues to cover by the grids.
All grids are saved in this file, another script is needed to write them in a displayable format — kont, GRID format / vtk, ParaView format —

- Run the grail_generation.py script
- It requires several arguments:
  - [Required] -cdf The path of the cdf file
  - [Required] -res The path of the text file listing the residues

  - [Optional] -o The output folder where the grid.cdf file will be generated (Default: current directory)
  - [Optional] -lig The 3-letter code of the ligand to generate an atom density grid
  - [Optional] -t The number of threads to launch (Default: 6)
  - [Optional] -bur To generate a buriedness grid (Default: False)

Only the protein side is considered during the interaction grid generation.

➢ 2.3. Step 3: Grid format conversion

➢ 2.3.1. Conversion to KONT format

This script generates kont files (GRID program format) from a grid.cdf file.
The kont files can be visualized with, for example, LigandScout.

- Run the grail_conversion_KONT.py script
- It requires several arguments:
  - [Required] -gcdf The path of the grid.cdf file

  - [Optional] -o The output folder where the kont files will be generated (Default: current directory)
  - [Optional] -fi Index of the frame to consider (Default: 0)

A frame index other than 0 will , in this case, lead to an error.

➢ 2.3.2. Conversion to VTK format

This script generates vtk files (ParaView format) and a pvsm (ParaView state format) from both a grid.cdf file and a cdf file. The pvsm (ParaView state format) can directly be loaded with ParaView to visualize both the simulation and grid at the same time.

- Run the `grail_conversion_VTK.py` script
- It requires several arguments:
    - [Required] `-gcdf` The path of the grid.cdf file
    - [Required] `-cdf` The path of the cdf file

    - [Optional] `-o` The output folder where the files will be generated (Default: current directory)

⬡ 3. Single script for grid generation from a PDB structure (to KONT output)

This script generates the pharmacophore interaction grids and protein atom density grids from a PDB file. The grid encompasses the entire protein. The grids are saved in the kont format.

- Run the `get_grail_pdb.py` script
- It requires several arguments:
    - [Required] `-pdb` The path of the PDB file

    - [Optional] `-o` The output folder where the files will be generated (Default: current directory)
    - [Optional] `-lig` The 3-letters code of the ligand to generate an atom density grid
    - [Optional] `-bur` To generate a buriedness grid (Default: False)

Only the protein side is considered during the interaction grids generation.

⬡ 4. Grid set generation from a molecular dynamics simulation

The Grail generation process is divided in **6 independent steps**:

- Generation of the cdf file ─ cdf, CDPKIT format ─ from a topology and a trajectory files
- Selection of the more stable amino acids to align the cdf file frames on.
- Selection of the residues to be enclosed by the grids
- Generation of the interaction and atom density grids ─ grid.cdf, CDPKIT format ─ from the cdf file
- [Optional] Generation of the average interaction and average atom density grids ─ avg_grid.cdf, CDPKIT format ─ from the grid.cdf file
- Generation of the grid files for visualisation ─ kont, GRID format / vtk, ParaView format ─ from the grid.cdf or avg_grid.cdf file

➢ 4.1. Step 1: cdf file generation

This script generates a cdf file (CDPKIT format) from both a topology and a trajectory file.

- Run the `cdf_generation_MD.py` script
- It requires several arguments:
    - [Required] `-trj` The path of the topology file
    - [Required] `-top` The path of the trajectory file

    - [Optional] `-o` The output folder where the cdf file will be generated (Default: current directory)
    - [Optional] `-cs` The number of frames to consider per chunk (default: 500 frames)
    - [Optional] `-n` The name of the cdf file (Default: name of the dcd file)

The number of frames to consider for the generation allows to split the cdf files in chunks, to reduce the use of memory during the processing.

➢ 4.2. Step 2: Residue selection and alignment of the cdf file

This script generates a text file displaying the maximum fluctuation and the rmsf of all the residues during the simulation.

- Run the residue_stability.py script
- It requires several arguments:
  - [Required] -cdf The path of the cdf file

  - [Optional] -o The output folder where the text file will be generated (Default: current directory)

It is important to **select several stable amino acids as anchor points to limit the residue motions** caused by the molecular dynamics simulation. **You must copy the lines corresponding to the selected amino acids into a new file** and provide it as an argument for the next script.

This script generates an aligned cdf file (CDPKIT format) from both a cdf and a text file with the amino acids to use as anchor.

- Run the residue_alignement.py script
- It requires several arguments:
  - [Required] -cdf The path of the cdf file
  - [Required] -res The path of the text file with the residues information

  - [Optional] -o The output folder where the aligned cdf file will be generated (Default: current directory)

➢ 4.3. Step 3: Amino acid selection for the grid generation

Considering your system, **select the amino acids/residues to be covered by the calculated grids**. **You must copy the lines corresponding to the selected amino acids/molecules from the previous step into a new file** and provide it as an argument for the next script.

➢ 4.4. Step 4: Grid generation

This script generates a grid.cdf file (CDPKIT format) from both an aligned cdf and a text file listing the residues to cover by the grids.
All grids are saved in this file, another script is needed to convert them to a displayable format ─ kont, GRID format / vtk, ParaView format ─

- Run the grail_generation.py script
- It requires several arguments:
    - [Required] -cdf The path of the cdf file
    - [Required] -res The path of the text file listing the residues of interest

    - [Optional] -o The output folder where the grid.cdf file will be generated (Default: current directory)
    - [Optional] -lig The 3-letter code of the ligand to generate an atom density grid
    - [Optional] -t The number of thread to launch (Default: 6)
    - [Optional] -bur To generate an buriedness grid (Default: False)

Only the protein side is considered during the interaction grids generation.

➢ 4.5. Step 5 [Optional]: Average grid generation

This script generates an avg_grid.cdf file (CDPKIT format) from a grid.cdf file.
For each type of interaction and atom density grid, average score and score standard deviation values are computed. This is done for every grid point over all frames of the MD simulation. All grids are saved in this file, another script is needed to convert them to a displayable format ─ kont, GRID format / vtk, ParaView format ─

- Run the grail_average_generation.py script
- It requires several arguments:
    - [Required] -gcdf The path of the grid.cdf file

    - [Optional] -o The output folder where the average_grid.cdf file will be generated (Default: current directory)
    - [Optional] -fr Frame range to consider, eg [0,55] (Default: all frames)

➢ 4.6. Step 6: Grids format conversion

➢ 4.6.1. Conversion to KONT format

This script generates kont files (GRID program format) from a grid.cdf file.
The kont files can be visualized with, for example, LigandScout.

- Run the `grail_conversion_KONT.py` script
- It requires several arguments:
    - [Required] `-gcdf` The path of the grid.cdf file

    - [Optional] `-o` The output folder where the kont files will be generated (Default: current directory)
    - [Optional] `-fi` Index of the frame to consider (Default: 0)

➢ 4.6.2. Conversion to VTK format

This script generates vtk files (ParaView format) and a pvsm (ParaView state format) from both a grid.cdf file and a cdf file. The pvsm (ParaView state format) can directly be loaded with ParaView to visualize both the simulation and grid at the same time.

- Run the `grail_conversion_VTK.py` script
- It requires several arguments:
    - [Required] `-gcdf` The path of the grid.cdf file
    - [Required] `-cdf` The path of the cdf file

    - [Optional] `-o` The output folder where the files will be generated (Default: current directory)

⬡ 5. Grid visualization

➢ 5.1. KONT format

- Open LigandScout
- Open the PDB file
  - A PDB file can be created with the write_PDB_frame.py script
- for each grid:
  - Insert the KONT file
  - On the Loading Grid panel, click on OK
  - Choose the color of the grid by clicking the colored square (left panel)
  - Select the grid (left panel)
  - Change the contour threshold by sliding the cursor (main panel)

➢ 5.2. VTK format

- Open ParaView
- Load the .pvsm state
- On the new window, click on OK
- You can display/hide the grids by clicking on the corresponding eye (left panel)
- You can center the structure by clicking on the reset button (top toolbar)
- You can run the MD simulation by clicking on the arrows (top toolbar)