

## Analysis options in SamSrf

As of version 10.3, there are three ways to run analyses in *SamSrf*:

1. ***SamSrfX* GUI:** You can run this either as a standalone app or by launching it in Matlab. This is perhaps the most user-friendly option and it allows you to inspect a host of modelling options with detailed explanations. This makes it perhaps the best way for a beginner to learn the ropes, especially combined with the *SamSrfX*-Manual. However, some options are not available in the GUI and it lacks flexibility of accessing the code. While the GUI has been implemented in *NeuroDesk*, due to how Matlab GUI play with Linux, this can be annoyingly buggy. The GUI also contains menu options to run other tools, such as *ViewApertures*, *DisplayMaps*, and *DelineationTool*. All these tools are also provided as standalone apps.
2. **Matlab scripts:** For maximal flexibility, you can run *SamSrf* code directly in Matlab. This is the old-school way that most of our lab and collaborators have been using in the past. It involves setting up a model structure for the parameters of the analysis and then passing this to the algorithm function (e.g., *samsrf\_fit\_prf*, *samsrf\_revcor\_prf*, etc.). This has many advantages but it requires a fairly in-depth understanding of how to set up an analysis – and it requires Matlab, which is proprietary software.
3. ***SamSrfAnalysis* command:** This command line interface can be used as a standalone app. This means you don't require a Matlab license and it is an easy way to run the analysis in the *NeuroDesk* environment. However, this feature requires some in-depth knowledge of how *SamSrf* analyses work. It has been designed specifically for use with open datasets, such as those using the BIDS framework. All necessary data can be read in from NII and GII files, and the analysis outputs such files for further use.

## SamSrfAnalysis command line

The basic logic of the *SamSrfAnalysis* command is that you provide a wildcard to find data files and a JSON parameter file to specify the model options, the same way you would use the Model structure when running the analysis in Matlab. You can then also provide some alternative options, such as how data are preprocessed or whether the analysis is restricted to a ROI.

Some basic example JSON files to specify analyses are included in *SamSrf/Models/Json*

On the next page we explain how to use the command line tool.

SamSrfAnalysis Algorithm ModelJson DataPath Roi SurfFolder Normalise Average

*Algorithm:*

prf / rc / cf (for forward-model pRF, reverse-correlation pRF, or reverse-correlation CF)

*ModelJson:*

JSON file with the model description. Path is relative to where the data files are

*DataPath:*

Path & filename (allows wildcards) to the data files, which can be in NII or GII format

*Roi:*

Name of ROI definition to restrict analysis without file extension.

Must be FreeSurfer LABEL for surface analysis or NII binary mask for volumetric analysis

*SurfFolder:*

Path to the FreeSurfer surface folder

*Normalise:*

1 (default) to detrend & z-score time series in each run or 0 for no normalisation

*Average:*

1 (default) if you want to average time series across runs or 0 for concatenating runs

## Examples

```
SamSrfAnalysis prf 2dGaussian /data/001/func/lh_*.gii lh_occ ../surf
```

Runs a 2D Gaussian forward-model pRF analysis specified by 2dGaussian.json of the data in /data/001/lh\_\*.gii, restricted to ROI lh\_occ.label & using the surface meshes in ../surf

```
SamSrfAnalysis prf 2dGaussian /data/001/func/lh_*.gii
```

Runs the same analysis as above but simply using the data contained in the lh\_\*.gii files but without treating it as a surface & not restricted to any ROI

```
SamSrfAnalysis rc RevCor /data/001/func/lh_*.gii -noroi ../surf/lh.inflated.gii
```

Runs a reverse-correlation pRF analysis specified by RevCor.json on the same data but uses the inflated surface provided by the GII & not restricted to any ROI