

# Gapr User Guide

GOU Lingfeng

1.0.1

# Table of Contents

1. Introduction .....	1
2. <b>Proofread</b> module .....	1
2.1. Preparation .....	2
2.2. Adjusting contrast .....	2
2.3. Mouse usage .....	2
2.4. Positions .....	3
2.5. Operations .....	3
3. <b>Fix</b> module .....	5
3.1. Operations .....	5
4. <b>Convert</b> module .....	7
5. <b>Trace</b> module .....	7
6. <b>Gather</b> module .....	8
6.1. <b>catalog</b> file .....	8
Appendix A: Man pages .....	9
A.1. <code>gapr-proofread</code> .....	9
A.2. <code>gapr-fix</code> .....	10
A.3. <code>gapr-convert</code> .....	11
A.4. <code>gapr-trace</code> .....	12
A.5. <code>gapr-gather</code> .....	14
A.6. <code>gapr-show</code> .....	15
A.7. <code>gapr-export</code> .....	17
A.8. <code>gapr-brief</code> .....	18
A.9. <code>gapr-map</code> .....	19

# 1. Introduction

Gapr is a software system for neuron reconstruction with huge light-imaging datasets. Firstly, neurites are automatically reconstructed to save human labor. Then, human annotators collaboratively proofread the preliminary reconstruction results and fix all sorts of errors. Finally, each neuron is visually checked to assess reconstruction quality.

Gapr is directly available to major desktop operating systems:

- any GNU/Linux distributions with recent Flatpak support;
- Windows, version 10 or newer versions;
- macOS, version 10.15 or newer versions.

By compiling from scratch, Gapr can potentially support many other desktop operating systems.

Gapr has been thoroughly tested under Fedora 37+. The Proofread and Fix modules are also thoroughly tested under Windows 10. In case of any operating system compatibility issues, please contact the author.

To install Gapr, download the version for your OS at [Gapr's homepage](#). Then double-click the downloaded file to initiate installation. The installation process typically takes less than one minute, except that automatic downloading of Flatpak runtime for GNU/Linux might take longer.

After installation, launching the application will start the default module. To start other modules in Windows and macOS, locate and run corresponding executable files. In Linux, run `flatpak run --command=gapr-xyz cn.org.yanlab.Gapr` to start the **Xyz** module (replace Xyz with the actual module name).

To access any dataset, please fill in the repository and your user credential. After pressing **[Ok]**, reconstruction results will be ready typically within a minute. To load downsampled imaging data, explicit **[Refresh]** or change of position is needed. For full access, download the *full* package and use the **Fix** module.

Read [Proofread module](#) and [Fix module](#) for usage of Gapr's proofreading GUI.

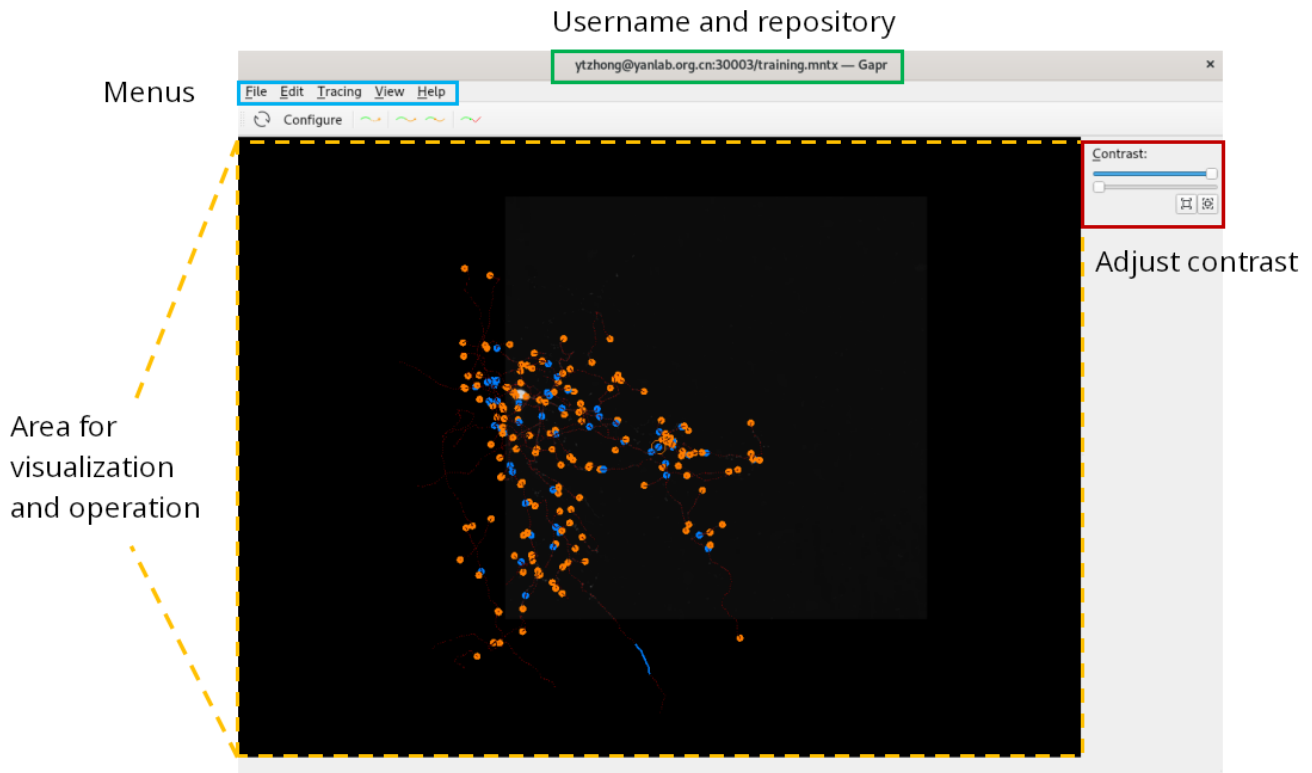
## 2. Proofread module

This module is designed to allow many annotators to participate proofreading. It features simple GUI and has low hardware requirements. Annotators proofread nodes, by fixing simple errors, reporting complicated errors and marking nodes that have been proofread, with a minimal set of operations.

Proofreading work is distributed among multiple annotators, and work for each annotator is divided to multiple *sessions*. In each session, a cube of imaging data and all relevant neighboring nodes are presented. After all nodes are processed, or explicitly skipped, the current session is completed, and next session begins by loading data in another cube.

## 2.1. Preparation

Annotators need to enter repository name, username and password to establish connection with the target repository. Click an entry in the history list to automatically fill in the previous login information (but without passwords). After logging in, please wait for the reconstruction results and imaging data to load. When ready, the following interface is show.



Click on the menu to see all the operations and the corresponding shortcut keys. Using shortcut keys can greatly improve operation efficiency.

## 2.2. Adjusting contrast

On the right side is the contrast adjustment tool. Move the sliders to change the displayed intensity range. Click the **[ Reset ]** button below to reset to the intensity range of [0, 1]. Click the **[ Shrink ]** button to narrow the adjustment range to the currently selected range, which is helpful for fine-tuning. Most of the time, adjusting the maximum value (the upper slider) suffices. Adjusting the minimum value (the lower slider) is only necessary when the background is too bright.

## 2.3. Mouse usage

In the middle operation area, the following mouse operations are available:

- Turn the mouse wheel to zoom out or zoom in.
- Right-click to show the shortcut menu.
- Hold the left button and drag to rotate the view.
- While pressing **[Alt]**, hold the left button and drag to box-select nodes.

- Left-click to pick a position.

To pick a node, click near an edge or vertex. Otherwise, a position with highest intensity is picked. If such a position is not desirable, rotate the view and click again. The intersection of the two clicks can precisely locate the desired position.

## 2.4. Positions

In Gapr, the center for visualization is called the *current position*. It is also the center for rotation and zooming. The current position may be simply a 3D coordinate or associated with a specific node, which is called the *current node*. Navigation in 3D space thus means to change the current position. The current position is indicated by a circle.

Another position is needed in order to select a sequence of nodes. It is called the *target position*, indicated by a cross.

## 2.5. Operations

### Goto target

**Edit > Goto target**, **G**.

Jump to the target position, i.e. set the current position equal to the target position.

### Pick current

**Edit > Pick current**.

Pick the current position, i.e. set the target position equal to the current position.

### Skip current node

**Edit > Skip current node**, **S**.

Skip the current node, and jump to the next node to process. The skipped node will not be revisited in this session. It is used when there is insufficient information to make decisions, typically near the imaging data boundary of the current session. If there is no associated node at the current position, the current node does not need processing, or the current node has been skipped, only jump to the next node.

### Skip current cube

**Edit > Skip current cube**, **Shift + S**.

Skip all local nodes that need processing, essentially starting a new session.

### Create neuron

**Edit > Create neuron**, **Ctrl + N**.

Add a neuron at the target position. If the target position is not associated with a node, a new node is created.

### Report error

**Edit > Report error**, **Ctrl + E**.

Add an error report to the node associated with the target position. Experienced annoators will be directed to this node to resolve this error.

## Reopen error

**Edit › Reopen error.**

Mark a previously resolved error at target position as incorrectly resolved. Experienced annotators will be directed to revisit this error.

## Resolve error

**Edit › Resolve error,** `Ctrl + R`.

Resolve the reported error at target position. This is an advanced operation; refer to [Resolve error in Fix module](#).

## Connect

**Tracing › Connect,** `C`.

Compute and display a new segment connecting the current position and the target position. Before confirmation ([Extend](#) or [Branch](#)) the new segment is not add to the reconstruction model.

If this operation is repeated, the new segment will be computed with slightly different methods.

## Extend

**Tracing › Extend,** `Space`.

Add the new segment and jump to the target position. Use this when a next new segment is needed at the target position.

## Branch

**Tracing › Branch,** `B`.

Add the new segment. Use this when a next new segment is needed at the current position, e.g. to create a branch.

## End

**Tracing › End,** `E`.

Mark the terminal node associated with the current position as finished, jump to the next node to process. Use this for normal neurite terminals. If there is no associated node at the current position, the current node is not a terminal, or the terminal has been marked, only jump to the next node.

## End as

**Tracing › End as,** `Shift + E`.

Similar to [End](#), but use this for abnormal terminals. For example, when signal intensity gradually decreases to background level.

## Delete

**Tracing › Delete,** `X`.

Delete the segment between the current position and the target position. The two positions must be on the same edge.

## Examined

**Tracing › Examined,** `W`.

Mark all nodes on the segment between the current position and the target position as

proofread, and jump to the target position. The two positions must be on the same edge. Do this only when errors on this segment are all fixed or reported.

If there are nodes selected, selected nodes are marked proofread and the current position is not changed.

## 3. Fix module

This module provides more advanced operations for experienced annotators to solve complex reconstruction problems. Refer to [Proofread module](#) for basic operations which are identical for both modules. It suffices to show the extra operations in **Fix**.

### 3.1. Operations

#### Goto position

**Edit > Goto position**, `Ctrl + G`.

Jump to a position given by user (a 3D coordinate in parenthesis, or a node ID starting with @).

#### Goto next error

**Edit > Goto next error**, `G E`.

Jump to the next unresolved error.

#### Clear end state

**Edit > Clear end state**, `X E`.

Clear the end state (set by [End](#) or [End as](#)) at the target position.

#### Resolve error

**Edit > Resolve error**, `Ctrl + R`.

Mark the error report at the target position as resolved. A resolved error can be resolved as one of the following types:

- *fixed*, any reconstruction errors or uncertainties associated with the error report are correctly resolved.
- *unresolvable*, the error report cannot be resolved, typically due to imaging data flaws.
- *deferred*, the error report is to be resolved at a later time, when more information is available.
- *invalid*, the error report itself is invalid, e.g. an error report associated to correctly reconstructed nodes.
- *redundant*, the error report itself is redundant, e.g. an error report that is already implied by another nearby error report.

Though we wish every error report will be *fixed*, do not hesitate to resolve as other types when necessary.

## Raise node

**Edit › Raise node**, **Ctrl** + **U**.

Raise the node associated with the target position for *selective reconstruction*. With selective reconstruction, only nodes connected to raised nodes are proofread.

## Rename neuron

**Edit › Rename neuron**, **Ctrl** + **Shift** + **R**.

Rename the selected neuron in the neuron list.

## Remove neuron

**Edit › Remove neuron**, **Ctrl** + **Shift** + **X**.

Remove the selected neuron in the neuron list. No node is actually removed.

## Highlight loop

**View › Highlight loop**, **H** **L**.

Highlight the loop at the target position.

## Highlight upstream

**View › Highlight upstream**, **H** **U**.

Highlight upstream branches at the target position.

## Highlight downstream

**View › Highlight downstream**, **H** **D**.

Highlight downstream branches at the target position.

## Highlight neuron

**View › Highlight neuron**, **H** **N**.

Highlight the whole neuron at the target position.

## Highlight raised nodes

**View › Highlight raised nodes**, **H** **R**.

Highlight all selected nodes during selective reconstruction (nodes connected to raised nodes).

## Highlight orphan

**View › Highlight orphan**, **H** **O**.

Highlight orphan branches (not connected to a neuron) at the target position.

## Reset highlight

**View › Reset highlight**, **H** **H**.

Exit highlight mode.

## Attach

**Tracing › Attach**, **Ctrl** + **Shift** + **A**.

Similar to [Extend](#), but define sub-tree structures. For example, this is used to model all axon branches of a neuron as a sub-tree.



## Delete

Tracing › Delete, **X** **X**.

If there are nodes selected, selected nodes are deleted. This allows administrators to remove false-positive nodes in bulk after automatic reconstruction. To *select nodes in bulk*, box-select with **Alt** and **Shift** pressed.

## 4. Convert module

This module is to prepare a stack of 2D images for Gapr. Images are converted to 3D cubes and compressed with video encoders. Currently, TIFF images are supported.

Firstly, all 2D images are scanned for consistency. Image resolution, output cube size, and downsampling factors are required. For example:

```
gapr-convert -w./workdir root@localhost:8003/fmost234162/abc -s400:400:80 -d100:100:20  
-r0.65:0.65:3 -i./images.txt -j4
```

This command will prepare data for project `fmost234162`. Each output cube has `400x400x80` voxels. The output downsampled cube will be downsampled by a factor of 100 in the X and Y direction. Each input slice has a resolution of 0.65um per pixel, and two adjacent slices are 3um apart. All input slice files are listed in `images.txt`, one per line. Four threads are allocated for the CPU intensive work. Conversion state is stored in `workdir`.

Conversion can be resumed, typically after network or filesystem errors, with the following command:

```
gapr-convert -C./workdir root@localhost:8003/fmost234162 -j4
```

## 5. Trace module

This module automatically trace neurites, starting from seed nodes. To add seed nodes, manually create a few neurons or neurite segments using the **Fix** module. For example, run the following to trace the project named `fmost202814` as user `bot.1` at `localhost:8003`, with 4 tracing jobs in parallel, using the `mouse.params` deep neural network parameter files.

```
gapr-trace -e mouse.params bot.1@localhost:8003/fmost202814 -j4
```

Please note that, we cannot package all the deep learning dependencies in our binary distributions. To use this feature (`-e mouse.params`), please compile from source (currently only available for GNU/Linux).

This module also helps to reset proofread status for nodes, so that annotators can perform another round of proofreading. For example, run the following to reset proofread status for all nodes.

```
gapr-trace --reset-all root@localhost:8003/fmost202814
```

## 6. Gather module

This module is needed to setup a server for collaborative reconstruction. It should be run on devices that annotators can access through network. In order to establish secure connection, two files named `certificate` and `private.key` are required. Please consult the Internet for methods of obtaining SSL certificates. With these two file in the directory `root_dir`, this server module can easily started with the following command.

```
gapr-gather -b localhost:8003 -d root_dir
```

Now, you can access the Web interface at <https://localhost:8003/>.

### 6.1. catalog file

If imaging data is not prepared with the **Convert** module, the **Gather** module can refer to imaging data hosted elsewhere with the help of a **catalog** file.

This is an example **catalog** file:

```
[CH00]
pattern=http://localhost:8080/test/ch00/z<04Z>.y<04Y>.x<04X>.webm
size=768 1536 896
cubsize=384 384 128
origin=0 0 0
direction=0.32 0 0 0 0.32 0 0 0 1
[CH00ds]
location=http://localhost:8080/test/ch00ds.nrrd
size=30 61 112
origin=0 0 0
direction=8 0 0 0 8 0 0 0 8
```

Two channels are defined. The first channel `CH00` refers to a large 3D image (768x1536x896) that is split to a set of smaller image files with uniform size (384x384x128). Each smaller image file is addressed with a URL resulting from the substitution of offset patterns in `pattern` with the image file's actual offsets. For example, if Y offset is 384, `<04Y>` is replaced by `0384` (pad with `0` to 4 characters, use `Y` offset). Finally `origin` and `direction` establish a affine transformation that maps voxel offsets to physical positions. Typically, simple transformations are used to describe image resolution (0.32x0.32x1  $\mu\text{m}^3$ ). The other channel `CH00ds` refers to a single image file at `location`. Such downsampled channels are used for navigation purposes.

To create a project with a given catalog file, run this:

```
curl https://localhost:8003/api/catalog/fmost202814/abc -k -T ./fmost202814.catalog
```

Note that, `curl` is not part of Gapr, but typically available in most operating systems.

## Appendix A: Man pages

### A.1. `gapr-proofread`

GAPR-PROOFREAD(1) Gapr Manual GAPR-PROOFREAD(1)

#### NAME

`gapr-proofread` - manually proofread neurite reconstruction

#### SYNOPSIS

```
gapr-proofread [option]... repo  
gapr-proofread --help | --version
```

#### DESCRIPTION

Connect to repository `repo` and manually proofread reconstruction. `repo` has the format:

```
[[user@]host:port/]repo-id
```

#### OPTIONS

`-c, --config=cfg-file`  
Load extra configuration from file `cfg-file`.

`--help`  
Display this help and exit.

`--version`  
Display version information and exit.

#### AUTHOR

Gapr was written by GOU Lingfeng.

#### BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

#### RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

#### COPYRIGHT

Copyright © 2017-present GOU Lingfeng.  
Free use of this software is granted under the terms of the GPLv3+ License.

## A.2. gapr-fix

GAPR-FIX(1)

Gapr Manual

GAPR-FIX(1)

### NAME

gapr-fix - manually fix neurite reconstruction

### SYNOPSIS

```
gapr-fix [option]... repo
gapr-fix --help | --version
```

### DESCRIPTION

Connect to repository `repo` and manually fix reconstruction. `repo` has the format:

```
[[user@]host:port/]repo-id
```

### OPTIONS

**-c, --config=cfg-file**  
Load extra configuration from file `cfg-file`.

**--help**  
Display this help and exit.

**--version**  
Display version information and exit.

### AUTHOR

Gapr was written by GOU Lingfeng.

### BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

### RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

### COPYRIGHT

Copyright © 2017-present GOU Lingfeng.  
Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2024-04-17

GAPR-FIX(1)

## A.3. gapr-convert

GAPR-CONVERT(1)

Gapr Manual

GAPR-CONVERT(1)

### NAME

`gapr-convert` - preprocess TIFF images.

### SYNOPSIS

```
gapr-convert [option]... -w work-dir repo -s cube-size -d ds-factor -r  
resolution -i input  
gapr-convert [option]... -C work-dir repo  
gapr-convert --help | --version
```

### DESCRIPTION

Convert TIFF files listed in input to Gapr compatible image cubes and upload to repo.

If `-C` is specified, resume previous conversion.

The remote repository `repo` has the format:

```
[[user@]host:_port_/]repo-id
```

### OPTIONS

- `-c, --config=cfg-file`  
Load extra configuration from file `cfg-file`.
- `-i, --input-list=input`  
Read TIFF images specified in input file, one TIFF file per line.
- `-r, --resolution=resolution`  
Specify the X, Y and Z resolution, in the form `x:y:z`.
- `-s, --cube-size=cube-size`  
Specify output cube size in the X, Y and Z direction, in the form `x:y:z`.
- `-d, --downsample-factor=ds-factor`  
Specify factors for downsampling in the X, Y and Z direction, in the form `x:y:z`.
- `--downsample-ratio=ds-ratio`  
Specify the ratio of voxels that will be read to generate the preliminary downsampled image. Smaller value reduces pre-scan time.
- `-w, --workdir=workdir`  
Save conversion state in `workdir`, needed for resume.
- `-j, --jobs=N`

Run N CPU intensive jobs in parallel.

**-B, --buffer-size=B**

Use B GB in memory for buffering imaging data.

**-C, --continue=workdir**

Resume the conversion in workdir.

**--to-tiled[=w:h]**

Convert input TIFF files to tiled TIFFs with tile size (w, h). If w is 0, convert to striped TIFFs with h rows per strip. If h is 0, tile size is (w, w). If both w and h are 0s, use tile size (1024, 1024). This option is useful when input TIFF images are not tiled, as tiled images will be more efficient during conversion. **Use this with caution, as input TIFF files will be deleted (replaced by the output tiled version).**

**--help**

Display this help and exit.

**--version**

Display version information and exit.

#### EXIT STATUS

0

Success.

1

Failure.

#### AUTHOR

Gapr was written by GOU Lingfeng.

#### BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

#### RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

#### COPYRIGHT

Copyright © 2017-present GOU Lingfeng.

Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2024-04-17

GAPR-CONVERT(1)

## A.4. gapr-trace

GAPR-TRACE(1)

Gapr Manual

GAPR-TRACE(1)

## NAME

gapr-trace - automatic neurite tracing

## SYNOPSIS

```
gapr-trace [option]... repo
gapr-trace --reset-all [option]... repo
gapr-trace --import=swc repo
gapr-trace --help | --version
```

## DESCRIPTION

Perform automatic neurite tracing with repository `repo`.

If `--reset-all` is given, reset the proofread status for all nodes, and exit.

If `--import` is given, import given `swc` file into `repo`.

`repo` refers to the remote repository and has the format:

```
[[user@]host:port/]repo-id
```

## OPTIONS

- `-c, --config=cfg-file`  
Load extra configuration from file `cfg-file`.
- `-j, --jobs[=n]`  
Run `n` jobs simultaneously. If `n` is omitted, all CPU threads are used.
- `--maxiter=m`  
Run at most `m` iterations.
- `-e, --evaluator=pars`  
Use DNN parameters from files located at `pars`.
- `--distant=d1`  
Trace distant neurites, with distance to a soma larger than `d1`.
- `--near=d2`  
Trace near neurites, with distance to a soma smaller than `d2`.
- `--reset-all`  
Reset proofread status, instead of tracing neurites.
- `--import=swc`  
Import reconstruction in `swc` file.
- `--import-fin=swc`  
Import reconstruction in `swc` file. All nodes will be marked

proofread and all terminals will be marked finished.

**--help**

Display this help and exit.

**--version**

Display version information and exit.

**EXIT STATUS**

0

Success.

1

Failure.

**AUTHOR**

Gapr was written by GOU Lingfeng.

**BUGS**

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

**RESOURCES**

**Project website:** <http://yanlab.org.cn/gapr/>

**COPYRIGHT**

Copyright © 2017-present GOU Lingfeng.

Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2023-02-07

GAPR-TRACE(1)

## A.5. gapr-gather

GAPR-GATHER(1)

Gapr Manual

GAPR-GATHER(1)

**NAME**

gapr-gather - server for collaborative reconstruction

**SYNOPSIS**

**gapr-gather** [option]...

**gapr-gather** --help | --version

**DESCRIPTION**

Collaborative reconstruction allows many users to connect to the sever and perform reconstruction operations simultaneously.

**OPTIONS**

**-b, --bind=host:port**

Bind to the address at host:port.



**-d, --dir=dir**  
Use directory dir for configuration files and tracing results.

**--help**  
Display this help and exit.

**--version**  
Display version information and exit.

#### EXIT STATUS

**0**  
Success.

**1**  
Failure.

#### AUTHOR

Gapr was written by GOU Lingfeng.

#### BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

#### RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

#### COPYRIGHT

Copyright © 2017-present GOU Lingfeng.  
Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2022-08-26

GAPR-GATHER(1)

## A.6. gapr-show

GAPR-SHOW(1)

Gapr Manual

GAPR-SHOW(1)

#### NAME

gapr-show - reconstruction results visualization

#### SYNOPSIS

**gapr-show** [option]... swc-file...  
**gapr-show** -p [option]... repo-file  
**gapr-show** --help | --version

#### DESCRIPTION

Visualize reconstruction results in swc-files. With multiple files, the union is used. Thus, node IDs should not clash.

If `-p` is given, play back reconstruction history in repository file `repo-file`.

## OPTIONS

- `-i, --image=cat`  
Load image data from catalog at `cat`.
- `-d, --diff=swc0`  
Compare against `swc0`. If `-p` is given, this option is not allowed, since playback mode automatically compares consecutive snapshots. This option can be specified more than once, as long as node IDs do not clash; the union is used.
- `-p, --playback`  
Enter playback mode. Read a repository file instead of SWC files.
- `-s, --script=scr`  
Run Lua script in file `scr`.
- `--help`  
Display this help and exit.
- `--version`  
Display version information and exit.

## EXIT STATUS

- `0`  
Success.
- `1`  
Failure.

## AUTHOR

Gapr was written by GOU Lingfeng.

## BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

## RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

## COPYRIGHT

Copyright © 2017-present GOU Lingfeng.  
Free use of this software is granted under the terms of the GPLv3+ License.

## A.7. gapr-export

GAPR-EXPORT(1)

Gapr Manual

GAPR-EXPORT(1)

### NAME

gapr-export - export reconstruction results

### SYNOPSIS

```
gapr-export [option]... repo swc-file
gapr-export -s [option]... repo dir
gapr-export --raw [option]... repo raw-file
gapr-export --help | --version
```

### DESCRIPTION

Export reconstruction results from repository `repo` to file `swc-file`.

If `-s` is given, the results is saved to multiple files in `dir`.

If `--raw` is given, the raw reconstruction history is saved to `raw-file`.

The remote repository `repo` has the format:

```
[[user@]host:_port_/]repo-id
```

### OPTIONS

```
-c, --config=cfg-file
    Load extra configuration from file cfg-file.

-n num
    Export the snapshot with the first num commits.

-s, --split
    Split and write to output directory.

--raw
    Save raw reconstruction history.

--help
    Display this help and exit.

--version
    Display version information and exit.
```

### EXIT STATUS

```
0
    Success.

1
    Failure.
```

## AUTHOR

Gapr was written by GOU Lingfeng.

## BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

## RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

## COPYRIGHT

Copyright © 2017-present GOU Lingfeng.

Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2022-09-02

GAPR-EXPORT(1)

## A.8. gapr-brief

GAPR-BRIEF(1)

Gapr Manual

GAPR-BRIEF(1)

### NAME

gapr-brief - summary statistics of reconstruction results

### SYNOPSIS

```
gapr-brief swc-file...
gapr-brief --f1score=ref swc-file...
gapr-brief --help | --version
```

### DESCRIPTION

Calculate total neurite length for each swc-file.

If `--f1score` is given, calculate F1 scores instead.

### OPTIONS

```
--f1score=ref
    Calculate F1 score with file ref.

--help
    Display this help and exit.

--version
    Display version information and exit.
```

### EXIT STATUS

```
0
    Success.
```

```
1
```

Failure.

#### AUTHOR

Gapr was written by GOU Lingfeng.

#### BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

#### RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

#### COPYRIGHT

Copyright © 2017-present GOU Lingfeng.

Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2022-08-26

GAPR-BRIEF(1)

## A.9. gapr-map

GAPR-MAP(1)

Gapr Manual

GAPR-MAP(1)

#### NAME

gapr-map - transform reconstruction results

#### SYNOPSIS

**gapr-map** [option]... swc-file...

**gapr-map** --help | --version

#### DESCRIPTION

Apply transformation, in given order, to each swc-file.

#### OPTIONS

**-o, --output=pat**

Use output file pattern pat. Rules:

- **%:** %
- **%n:** filename
- **%p:** parent path
- **%s:** stem
- **%e:** extension

Use uppercase to allow empty substitution.

If this option is not given, all results are written to standard

output.

**-x, --xform=cmd**

Run cmd to transform coordinates.

**--decimate=d:h**

Reduce number of nodes. Any removed node satisfies the following requirements with d and h:



**--help**

Display this help and exit.

**--version**

Display version information and exit.

#### EXIT STATUS

0

Success.

1

Failure.

#### AUTHOR

Gapr was written by GOU Lingfeng.

#### BUGS

Report bugs to [goulf.3m@gmail.com](mailto:goulf.3m@gmail.com).

#### RESOURCES

**Project website:** <http://yanlab.org.cn/gapr/>

#### COPYRIGHT

Copyright © 2017-present GOU Lingfeng.

Free use of this software is granted under the terms of the GPLv3+ License.

Gapr 1.0

2022-08-26

GAPR-MAP(1)