Gip

gip is a set of wrapper scripts around PsN that will allow you to distribute and synchronize models and results between your local computer and remote servers.

The problem that gip solves

Many modelers find it a pain to constantly move files between the local computer and a cluster or a cloud server. Modeling only on a local computer is usually not an option due to limited CPU capacity. On the other hand, modeling only in the cloud is not always considered convenient either: although some excellent cloud tools are available (like RStudio Server, PiranaJS), many modelers find it more convenient to parse results and datasets on their local machine. Also, cluster administrators may not allow installation of specific custom tools or libraries, and modeling on a cluster always requires a connection to the internet. gip solves these problems by performing 'on-the-fly' transfer of models and results between your local computer and the cloud / cluster. At the same time it provides automatic backup and version control of your work in the cloud.

The mechanism

gip is called from the console on your local machine, followed by a location indicator, and the PsN command you want to execute:

```
gip [location] [PsN command]
```

The location indicator specifies where you want to run the PsN command. With every call to `gip`, your model and results are first saved to the local Git repository and synchronized to the cloud (`commit` + `push`), before any run is started. If the location indicator is not `local`, then `gip` will login to the specified server, lookup the project folder you are working in locally, and synchronize to the latest version of your models (`git pull`). Afer that it will start the actual PsN command on the server, and return to your local console. With the `status` and `sync` commands you can check run progress and synchronize results between your local computer and your server(s).

Some example commands

```
# run the model locally
# (same as 'execute run1.mod', but first saves/syncs changes)
gip local execute run1.mod
```

```
# 'local' may be dropped, as it is default
gip execute run1.mod

# Run the same model but now on the 'serv1' cluster
gip serv1 execute run1.mod

# Check if any results are available on any of the specified servers
gip status

# Check if any results are available on server 'serv1'
gip status serv1

# Look for new models/results on all servers, and synchronize
gip sync

# Look for new models/results on 'serv1' and synchronize
gip sync serv1

# Set up the git repository on your local machine, as well as
# on all specified servers
gip init
gip sync
```

Advantages / use cases

- Run your simple models locally, dispatch heavier computations to a cluster
- Run models locally when not connected to the internet, and work in the cloud when you are
- Always have a backup in the cloud
- Automatic version control on all models and results
- No wait when cluster is down, just sync your project to a different cloud/cluster
- Easily share projects with colleagues
- Integrates with Pirana and PiranaJS

Requirements

- PsN, NONMEM, and `gip` and `git` need to be installed both locally and on the servers.
- GitHub account or custom Git server account

License

`gip` is currently in the “idea”-stage, but will eventually be released under the open source MIT license.