1. Introduction

This manual gives an overview of HAPPI, a utility for the creation of simple linked HTML pages of different gene/protein sets based on their known protein-protein interactions. General details about Command-line options can be found in the RJE Appendices document included with this download. Details of command-line options specific to HAPPI can be found in the distributed readme.txt and readme.html files.

Like the software itself, this manual is a 'work in progress' to some degree. If the version you are now reading does not make sense, then it may be worth checking the website to see if a more recent version is available, as indicated by the Version section of the manual. Check the readme on the website for up-to-date options etc. In particular, default values for options are subject to change and should be checked in the readme.

Good luck.

Rich Edwards, 2011.

1.1. Version

This manual is designed to accompany HAPPI version 1.0.

The manual was last edited on 19 August 2011.

NOTE. This manual is a work in progress. Please contact me if you wish to use HAPPI and would like the manual updated.

1.2. Using this Manual

As much as possible, I shall try to make a clear distinction between explanatory text (this) and text to be typed at the command-prompt etc. Command prompt text will be `written in Courier New` to make the distinction clearer. Program options, also called 'command-line parameters', will be `written in bold Courier New` (and coloured `red` for fixed portions or `dark` for user-defined portions, such as file names etc.). Command-line examples will be given in (green) *`italicised Courier New`*. Optional parameters will (if I remember) be `[`in square brackets`]`. Names of files will be marked in coloured normal text.

1.3. Why use HAPPI?

HAPPI is a program for generating a series of linked webpages that summarise the protein-protein interaction (PPI) links between a set of input genes. It is not designed for complex analyses but rather to provide a "quick and dirty" first pass of establishing links between data. It is also designed to provide an easily navigable resource for exploring links between data and to external databases.

The initial table of genes (and their corresponding classes) is used to make the initial front page tables. Additional data tables can also be loaded and linked via the "Gene" field for individual Gene Pages. These must be named in the format X.N.tdt, where N will be used as the tab name.

A network clustering algorithm based on MCODE is used to identify densely connected subnetworks, onto which the input classes are mapped. These clusters can include additional interactors that are not present in the input data, potentially identifying candidates for novel players in the system being analysed.

Note that although the program is designed with PPI in mind, any kind of interaction data can be used by making the appropriate input files. (For PPI data, PINGU can be used to generate the pairwise PPI table.)

1.4. Getting Help

Much of the information here is also contained in the documentation of the Python modules themselves. A full list of command-line parameters can be printed to screen using the `help` option, with short descriptions for each one:

```
python happi.py help
```

General details about Command-line options can be found in the PEAT Appendices document included with this download. Details of command-line options specific to HAPPI can be found in the distributed readme.txt and readme.html files.

If still stuck, then please e-mail me (r.edwards@southampton.ac.uk) whatever question you have. If it is the results of an error message, then please send me that and/or the log file (see Chapter 2) too.

1.4.1. Something Missing?

As much as possible, the important parts of the software are described in detail in this manual. If something is not covered, it is generally not very important and/or still under development, and can therefore be safely ignored. If, however, curiosity gets the better of you, and/or you think that something important is missing (or badly explained), please contact me.

1.5. Citing HAPPI

HAPPI is currently unpublished. Until published, please cite the HAPPI Website.

1.6. Availability and Local Installation

HAPPI is distributed as a number of open source Python modules. It should therefore work on any system with Python installed without any extra setup required. If you do not have Python, you can download it free from www.python.org at http://www.python.org/download/. The modules are written in Python 2.5. The Python website has good information about how to download and install Python but if you have any problems, please get in touch and I will help if I can.

All the required files should have been provided in the download zip file. Details can be found at http://www.southampton.ac.uk/~re1u06/software/ and the accompanying PEAT Appendices document. The Python Modules are open source and may be changed if desired, although please give me credit for any useful bits you pillage. I cannot accept any responsibility if you make changes and the program stops working, however!

Note that the organisation of the modules and the complexity of some of the classes is due to the fact that most of them are designed to be used in a number of different tools. As a result, not all the options listed in the `__doc__()` (`help`) will be of relevance. If you want some help understanding the way the modules and classes are set up so you can edit them, just contact me.

2. Fundamentals

2.1. Running HAPPI

2.1.1. The Basics

If you have python installed on your system, you should be able to run HAPPI directly from the command line in the form:

```
python happi.py seqin=FILENAME
```

To run with default settings, no other commands are needed. Otherwise, see the relevant sections of this manual.

IMPORTANT: If filenames contain spaces, they should be enclosed in double quotes: *data="example file"*. That said, it is recommended that files do not contain spaces as function cannot be guaranteed if they do.

2.1.2. Options

Command-line options are suggested in the following sections. General details about Command-line options can be found in the RJE Appendices document included with this download. Details of command-line options specific to HAPPI can be found in the distributed readme.txt and readme.html files. These may be given after the run command, as above, or loaded from one or more *.ini files (see RJE Appendices for details).

2.1.3. Running in Windows

If running in Windows, you can just double-click the happi.py file. It is recommended to use the `win32=T` option. (Place this command in a file called happi.ini.)

2.2. Input

For full functionality HAPPI requires a set of genes associated with one or more classes of interest, a table of PPI (or other interaction) links, a table of database cross-references and a table of gene-GO links. Reduced functionality can be achieved with partial input. Additional optional input can be used to control the visualisation and text content of the pages.

2.2.1. Main Class Input

The main input table is given by indata=FILE. This should be a delimited file and include at least two columns, one for the gene and one for the gene class. These are "Gene" and "Class" by default but can be over-ridden with genefield=X and geneclass=X, respectively. This table can include additional fields, which will appear in the relevant gene pages but will not affect gene class annotation.

By default, classes will be ordered alphabetically

classorder=LIST : List of Class orders (otherwise alphabetical) []

multiclass=T/F : Whether to allow membership of multiple classes (joined by "-" [True]

makeclass=LIST : Generate classes from classorder LIST if missing from indata (go/keyword/desc/xref(+ppi)/add*) [keyword]

pagehead=X : Text for Front Page Header ['HAPPI Analysis of basefile']

infotext=X : Text to display under header ['This data has not yet been published and should not be used without permission.']

classcol=X : Table of "Class" and "Col" (soton$col indexes) to be used for PPI images [basefile.col.tdt]

fillcol=T/F : Fill in colour for missing class combinations [True]

genedata=LIST : List of additional data tables for Gene-centric pages. Must have "Gene" or genefield field. []

xrefdata=FILE : File of Database cross-references. Must have "Gene" field. []

xreftab=T/F : Add database cross-reference tabs to the front page Class tabs [True]

pairwise=FILE : Pairwise protein-protein interation (PPI) file []

addppi=LIST : List of additional PPI pairwise files to add []

godata=FILE : Delimited file of GO Data (Gene,GO_ID,GO_Type,GO_Desc) [basefile.go.tdt]

gablam=FILE : Delimited GABLAM results file for homology data [basefile.gablam.tdt]

ppexpand=X : Expand PPI by X levels for MCODE complex generation [1]

ppcomplex=LIST : List of different evidence codes for special MCODE analyses ['Complex']

ppextra=T/F : Make additional pages for genes added and returned in MCODE clusters [False]

combine=LIST : List of Classes to combine into Interactome for front page []

special=LIST : Execute special analysis code for specific applications []

Class Input

Additional classes can be added using the makeclass and classorder lists. Any classes in classorder that are not

found in indata will be added using the rules set by makeclass:

- go = GO term description or ID

- keyword = found in GO term description or protein description

- desc = found in protein description

- xref = adds PPI for protein identified from xrefdata table

- add = will only add a gene to a class if it does not already have one (no multiple-class genes)

2.3. Output

Primary output for HAPPI is a delimited text file [HAPPI.clean.tdt] containing the processed protein hits (Table 2.1).

In addition, each MASCOT search has a peptide table produced [HAPPI.SEARCH.peptide.tdt] that lists all the peptide sequences used for identifications, which hits they identified and whether they are UNIQUE or common to a CLUSTER of BLAST-related (Altschul, et al., 1990) proteins (or COMMON across clusters, which is unlikely).

2.4. Commandline Options

A full list of commandline options can be found in the readme file or by running:

```
python HAPPI.py help
```

Table 2.1. Fields for main HAPPI output file.

Field	Description
search	MASCOT search ID
hit	Protein hit number from MASCOT
class	HAPPI classification of protein (see 1.3)
cluster	BLAST-based clustering ID for non-REJECT proteins (independent numbering for each search)
accnum	Accession number of protein
spec	Species code for protein
species	Full species (if possible) for protein
desc	Protein description
pepcount	Initial MASCOT peptide count, including PTMs etc.
pep_con	No. of different peptides "converted" by HAPPI - PTMs ignored and MS ambiguities (e.g. Ile vs Leu) considered
pep_rem	No. of peptides removed due to being found in hits from query species
pep_uniq	No. unique peptides, found only in this hit
pep_nr	No. peptides found in 2+ "NR" proteins (see 1.3)
pep_red	No. redundant peptides also found in a UNIQUE protein (see 1.3)
peplist	Original list of peptides
conpep	Converted PISCI peptide list

3. Appendices

3.1. Troubleshooting & FAQ

There are currently no specific Troubleshooting issues arising with HAPPI. Please see general items in the PEAT Appendices document and contact me if you experience any problems not covered.

3.2. References

- Altschul SF, Gish W, Miller W, Myers EW and Lipman DJ (1990). Basic local alignment search tool. J Mol Biol, 215: 403-410.
- Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N and Yeh LS (2005). The Universal Protein Resource (UniProt). Nucleic Acids Res., 33: D154-159.