

Documentation: analysis code and sample datasets

Code

Top-level analysis modules

Two modules carry out the compute the indices described in the accompanying manuscript:

`psg_umi_triplike_demo.m` for the indices I_{sym} and I_{umi} (symmetry and ultrametric)

`psg_umi_triplike_demo.m` for the index $I'_{addtree}$ (additive tree)

They can be run interactively (with the options specified below), or non-interactively by setting `if_auto=1` and providing the non-default answers as the fields of the structure `auto`. See processing pipeline section for examples.

In non-interactive mode, the key outputs `r`, `s`, `sel_string`, and `sel_desc` are saved, as follows. First, the variable `db` is read from the file named in `auto.db_file` (if not present, it is created, with `db` initialized as empty). Then, `r`, `s`, `sel_string`, and `sel_desc` saved as fields of `db.(data_fieldname)`, where `data_fieldname` is the name of the data file, with `' .mat'` deleted, as given in `data_fullname`.

Typical run time for `psg_umi_triplike_demo.m` is under a minute for the examples provided; for `psg_tentlike_demo.m`, several minutes for a full dataset, under a minute for subsets.

psg_umi_triplike_demo

This calculates the indices I_{sym} and I_{umi} , with a range of interactively-selected options, calls

`psg_umi_triplike_plot.m` and `psg_umi_triplike_plota.m` to display the results, and packages them into a structure.

Calculations are carried out on various subsets of the triplets, thresholded by the number of trials in each triad. Three kinds of thresholds are considered: thresholds based on the minimum number of trials per triad in the triplet, the maximum number of trials per triad in the triplet, and based on the total (or average) number of triads in the triplet. For each of these threshold types, all possible values (from a threshold of zero to a threshold that excludes all triplets) are considered. Unless otherwise specified, the accompanying paper reports un-thresholded computations.

There are two options for computing the Dirichlet parameter α and the discrete-fraction parameter h : they can be computed from the entire dataset ("global" computation), or only from the triads that are in the selected triplets ("private" computation). All results reported in the manuscript use the global computation.

Additionally, calculations are performed with h assigned to each of a sequence of values (`[0 .001 .01 .1 .2 .4]`, 0 omitted for I_{umi}); this list is settable by assigning `h_fixlist` prior to run.

Console inputs

Enter 1 to use fixed value for α when h varies, 2 for fixed α when h varies or fixed but global (range: 0 to 2, default= 0):

Choose 0, the default. This fits the Dirichlet parameter α to the data. Alternatively, choose 2 to supply a specific value. (1 was used during development, and should not be chosen.) If you choose 2, the next prompt will ask for a value for α .

Enter 1 to reduce all triads to an even number of trials (range: 0 to 1, default= 0):

Choose 0, the default. (1 was used during development. This choice will randomly remove one trial from the observations for any triad that has an odd number of trials, thus making it possible for the naïve estimate of a choice probability to be 0.5.)

Enter 1 for accelerated global computation and standard private, -1 to omit private (range: -1 to 1, default = -1):

Choose -1, the default. Other options were used during development. These choices relate to how the likelihood ratios are calculated when only some of the triplets are used. Entering 0 computes likelihood ratios ab initio for both global and private threshold options. Entering 1 computes likelihood ratios from each triplet and then selects from these triplets according to the threshold, which accelerates the global computation. Entering -1 carries out the global computation in the same way, and omits the private computation.

Enter 1 to delete large variables (range: 0 to 1, default= 1):

Choose 1, the default. (0 was used during development.)

Enter 1 for detailed plots (range: 0 to 1, default= 0):

Choose either 0 or 1. 1 enables the “detailed” plots created with `psg_umi_triplike_plot.m`, which shows the log likelihood for the Dirichlet fit as a function of a and h , and plots of I_{sym} and $I_{umi} + \log(h)$ for the data and surrogates as a function of the thresholds. Plots are produced for global and (if applicable) private computations.

Enter 1 for summary and summary (asymptotic) plots (range: 0 to 1, default= 1):

Choose either 0 or 1. 1 enables summary plots of I_{sym} and I_{umi} , with just the lowest fixed value of h and the fitted value. 1 also enables computation of a summary structure, s , and a nicely formatted display of the indices at the console. These are created with `psg_umi_triplike_plota.m`

Enter 0 to generate random unstructured rank choice probabilities, 1 to read data (-1 to skip reordering of stimuli) (range: -1 to 1, default= -1):

Choose -1, the default. (0 was used in development, creates random data; 1 requires an additional metadata file not relevant to these analyses). However, if you are using the example texture data file, you can choose 1, and will be prompted for the metadata file name; the default, `./psg_data/bc6pt9.mat`, should be accepted).

Enter full path and file name of data file (default= `./psg_data/bc6pt_choices_MC_sess01_10.mat`):

Enter the data file name in the standard fashion. See below for data format. Typical file names used (where `xx` indicates the subject ID) for the textures, faces, and brightness datasets are:

```
bc6pt_choices_[XX]_sess01_10.mat
faces_mpi_en2_fc_choices_[XX]_sess01_10.mat
bright_c02s08_choices_[XX]_oddoneout.mat
```

Since these file names are modified to create fields that are later used for databases, it is worth choosing them carefully.

Enter 1 to analyze conforming surrogates (range: 0 to 1, default= 1):

Choose 1, the default. (0 was used during development).

Enter 1 to apply a selection criterion (range: 0 to 1, default= 0):

Choose either 0 or 1. Choosing 0 will analyze all stimuli and begin the computations. If you choose 1, you will be shown the list of stimulus type names (as read from the data file), and then two further queries. For the textures dataset, the list will be the following, assuming that you did not enter a metadata file above). The string `bp` designates stimuli with $\beta_- > 0$, `bm` designates $\beta_- < 0$, `cp` designates $\beta_+ > 0$, `cm` designates $\beta_+ < 0$, `rand` designates $\beta_- = \beta_+ = 0$.

```
typename 1: cp0300
typename 2: cp0450
typename 3: cm0750
typename 4: cm0900
typename 5: bp0450
typename 6: cm0150
typename 7: bp0900
typename 8: bp0600
typename 9: bm0300
typename 10: bm0600
typename 11: cm0300
```

```

typename 12: cp0750
typename 13: bm0900
typename 14: bp0150
typename 15: bm0150
typename 16: bp0750
typename 17: cm0450
typename 18: bp0300
typename 19: cp0150
typename 20: cp0900
typename 21: rand
typename 22: cp0600
typename 23: bm0750
typename 24: cm0600
typename 25: bm0450

```

Enter selection string or multiple strings, separated by |:

Enter a string such as `b|rand` to analyze only the stimuli whose name contains `b` or `rand`, etc. You'll then get a list of the stimuli to analyze, and enter 1 to confirm:

```

typenames selected:
  Columns 1 through 11
    'bp0450'    'bp0900'    'bp0600'    'bm0300'    'bm0600'    'bm0900'    'bp0150'    'bm0150'
'bp0750'    'bp0300'    'rand'
  Columns 12 through 13
    'bm0750'    'bm0450'

```

Enter 1 if ok (range: 0 to 1):

Enter 1 (or a 0 to re-enter the selection string)

Enter descriptor:

Enter a string, such with no special characters, e.g., `baxis`. This will start the analysis.

Console outputs

If you requested summary plots, there will also be a console log of the main results: one section for the smallest fixed value of the parameter h , one for h fitted to the choice probabilities.

Each section has a subsection for I_{sym} , labeled `sym vs (sym or asym)`, and for I_{umi} , labeled `umi vs (umi or not umi)`. Within each subsection is a table of values of the indices for each of the three kinds of thresholds, showing the values for the minimum thresholds that keep no more than the specific fraction of triplets requested (defaults to `2.^([0:10])`, settable in `plot_opts.frac_keep_list`)

Columns of the table provide, for each threshold, the a priori value of the index, the value from the original data, the “flip all” surrogate (replace all choice probabilities by their complement), the “flip any” surrogate (replace any combination of choice probabilities by their complement), and “flip one” surrogate (replace a single choice probability by its complement if that will allow for conformance to the symmetry or ultrametric condition). The quantities in brackets are, for the original data, the jackknife standard error of the mean, and for the surrogates, one standard deviation across all surrogates.

Output variables

Several structures are saved in the workspace (and are also saved automatically into a database file, see non-interactive mode described above). The key structures and fields are:

`r.nstims:`

number of stimuli (after selection, if applicable)

`r.su.[global|private].[sym|umi]{imv,ithr_type}(ithrval,id):`

mean (`imv=1`) and variance (`imv=2`) of the summed likelihood ratios from each triad, for I_{sym} or I_{umi} , for each kind of threshold (`ithr_type=[1:3]` for minimum, maximum; average), for each value of the threshold (indexed by `ithrval`),

and each kind of data ($id=[1:4]$ for experimental data, flip all, flip any, flip one for conformance). Calculation is for fitted h .

`r.su.[global|private].[sym|umi]_hfixed{imv,ithr_type}(ithrval,id,ihval):`
same as above except for fixed h . `ihval` indexes the value of fixed h in `r.h_fixlist`

`r.su.global.a{1,1,ihval}:` fitted value of a for each fixed h in `r.h_fixlist`

`r.su.global.ah:` fitted value of a for fitted h (but h set to zero if fitted $h < 0$)

`r.su.tallies{ithr_type}:` for each threshold type, first column is minimum number of trials, second column is number of triplets, third is total trials in those triplets

`r.dirichlet.tallies(ithr_type,:):` first column is minimum number of trials, second column is number of triads with that minimum, third column is total trials in those triads

`r.dirichlet.a(ithr,:,ihval):` for each value of fixed h in `r.h_fixlist` and each threshold level, first column is the fitted value of the Dirichlet parameter a , and second is the log likelihood per trial for the fit.

`r.dirichlet.ah(ithr,:):` for each threshold level, first column is the fitted value of the Dirichlet parameter a , second is fitted value of h , third is the log likelihood per trial for the fit.

`sel_string:` the string used to select a subset of stimuli

`sel_desc:` the descriptor of the subset

`s{ipchoice}.[sym|umi]:` summary structures produced by `psg_umi_triplike_plota.m`, with a single value of h for each `ipchoice`. `ipchoice = 1` for fixed h , `ipchoice = 2` for fitted h (I_{sym} only). For fixed h , a value of 0 is used for I_{sym} and the lowest positive value is used for I_{umi} . For fitted h , a negative value is replaced by zero.

`s{ipchoice}.[sym|umi].params.[a|h]:` fitted or fixed parameter values determined as above

`s{ipchoice}.[sym|umi].apriori_vals:` a priori values of the indices for the above (a,h) pairs

`s{ipchoice}.[sym|umi].ah_llr:` log likelihood ratios for the Dirichlet fits, per trial

`s{ipchoice}.[sym|umi].thr_type{ithr_type}.tally_table:` taken from `r.su.tallies{ithr_type}`

`s{ipchoice}.[sym|umi].thr_type{ithr_type}.means_per_set_adj:` mean values of indices, including the subtraction of $\log h$ for I_{umi} , for each value of threshold

`s{ipchoice}.[sym|umi].thr_type{ithr_type}.eb_stds:` error bars (s.e.m. for original data, standard devs for surrogates) corresponding to above means

`s{ipchoice}.[sym|umi].thr_type{ithr_type}.frac_keep_list:` `frac_keep_list`, defined above

`s{ipchoice}.[sym|umi].thr_type{ithr_type}.thr_ptr_use:` pointer to row of `tally_table`, `means_per_set_adj`, and `eb_stds`, indicating the value of the threshold needed to realize the requested fraction. This typically has fewer entries than `frac_keep_list` since it ends once the fraction to be kept is smaller than the fraction of triplets that meets the criterion

[psg_tentlike_demo](#)

This calculates the index $I'_{addtree}$ with a range of interactively-selected options, calls `psg_umi_triplike_plot.m` and `psg_umi_triplike_plota.m` to display the results, and packages them into a structure. Options are parallel to those of `psg_umi_triplike_demo.m`. The main difference is that the basic unit of calculation is the tent, rather than the triplet.

Console inputs

These are identical to those for `psg_umi_triplike_demo.m`. The “detailed” plots created with `psg_umi_triplike_plot.m` show the log likelihood for the Dirichlet fit as a function of a and h , and plots of $I'_{addtree}$ for the data and surrogates as a function of the thresholds. The summary plots are of $I'_{addtree}$ with just the lowest fixed value of h and the fitted value.

Console outputs

If you requested summary plots, there will also be a console log of the main results: one section for the smallest fixed value of the parameter h , one for h fitted to the choice probabilities. Organization is the same as for

`psg_umi_triplike_demo.m`, except that each section reports the values of $I'_{addtree}$, with the label `adt` and `sym/trans` vs `sym/trans`

Output variables

As in `psg_umi_triplike_demo.m`, several structures are saved in the workspace (and are also saved automatically into a database file, see non-interactive mode described above). The only differences are:

`r.adt` replaces `r.su`

`r.adt.[global|private].adt` replaces `r.su.[global|private].[sym|umi]`

`r.adt.[global|private].adt_hfixed` replaces `r.su.[global|private].[sym|umi]_hfixed`

`s{ipchoice}.adt` replaces `s{ipchoice).[sym|umi]`

Other key modules

Below is a brief description of key modules, described in order of use

`psg_read_choicedata`

`psg_read_choicedata.m` reads a `.mat` choice data file; see below for its structure.

`psg_ineq_logic`

`psg_ineq_logic.m` calculates the $1 - V$. This is the output variable “partitions”. The arguments $-1, 0, +1$ of V correspond to index values 1, 2, and 3 of `partitions`.

$1 - V_{sym}$ corresponds to input arguments `nc=3, ineq_type='exclude_sym' or 'exclude_trans'`

$1 - V_{umi}$ corresponds to input arguments `nc=3, ineq_type='exclude_umi_trans'`

$1 - V_{symtent}$ corresponds to input arguments `nc=6, ineq_type='exclude_trans_tent'`

$1 - V_{addtree}$ corresponds to input arguments `nc=6, ineq_type='exclude_addtree_trans'`

`psg_permutes_logic`

`psg_permutes_logic.m` creates the table of permutations for surrogate datasets

`psg_ineq_apply`

`psg_ineq_apply.m` combines the Dirichlet prior (along possibly with a discrete component h) and the observations to form a posterior, and integrates this posterior over the region defined by the partition V . It also carries out this calculation after optionally complementing subsets of the observed choice probabilities, to form the “all” and “any” surrogates, as defined by the input argument “`permutes`”. (It is called “`permutes`” since complementing the choice probabilities corresponds to permuting the partitions, and is created by `psg_permutes_logic.m`.) This is used by `psg_tentlike_demo.m`. `psg_umi_triplike_demo.m` does not use `psg_ineq_apply.m` but `psg_probs_check.m` verifies that the results are the same.

`psg_umi_triplike_plot`

`psg_umi_triplike_plot.m` creates detailed plots of analyses of single datasets for `psg_[umi_trip|tent]like_demo.m`, showing likelihood ratios and related quantities as a function of parameters a and h , including several kinds of surrogates.

`psg_umi_triplike_plota`

`psg_umi_triplike_plota.m` creates a summary of the analysis of single datasets and a nice console display of results, and creates a structure `s` needed for the structure `db` saved by `psg_[umi_trip|tent]_like_demo.m`, from which `psg_like_maketable.m` creates a table

`psg_like_maketable`

`psg_like_maketable.m` creates a table object from the outputs of `psg_[umi_trip|tent]_like_demo.m`

Key fields of `table_like` are: `paradigm_type` (e.g., `btcsel` for texture experiments); `paradigm_name` (describes the subset of stimuli analyzed); `subj_id` (participant ID); `llr_type` (1: I_{sym} , 2: I_{umi} , 3: $I'_{addtree}$); `ipchoice` (1: fixed h , 2: fitted h); `thr_type` (1: minimum, 2: maximum, 3: average), a ; the parameter a ; h : the parameter h ; `ah_llr`: the log likelihood, per trial, for the Dirichlet fit; `apriori_llr`: the a priori log likelihood ratio of I_{sym} , I_{umi} , or $I'_{addtree}$; `frac_keep`: the maximum fraction of triads to retain (based on the threshold), `thr_ptr`, pointer to threshold corresponding to `frac_keep`; `ntriads`: number of distinct triads (or tents) used in the calculation; `ntrials`: number of trials used in the calculation; remaining quantities as described above under output variables of `psg_[umi_trip|tent]_like_demo.m`.

`psg_like_analtable`

`psg_like_analtable.m` creates plots from the table structure of `psg_like_maketable.m`

Data format

Choice data are read from a `.mat` file that consists of three variables and contains the triadic judgments.

This documentation below uses `psg_data/bc6pt_choices_MC_sess01_10.mat` as an example of the format.

This file has the choice data from an experiment with a total of 10000 trials, across 10 sessions. Each trial consisted of rank-order judgments of similarity of 8 comparison stimuli to a reference (Waraich, S. A., & Victor, J. D. (2022). A Psychophysics Paradigm for the Collection and Analysis of Similarity Judgments. *J Vis Exp*(181). doi:10.3791/63461), and therefore yielded $\binom{8}{2} = 28$ comparisons per trial, yielding a total of 28000 triadic judgments. Note though that the data file only keeps track of the total triadic judgments, without regard to the paradigm that was used to collect them.

```
load psg_data/bc6pt_choices_MC_sess01_10.mat
whos
```

Name	Size	Bytes	Class	Attributes
<code>responses</code>	6856x5	274240	double	
<code>responses_colnames</code>	5x34	340	char	
<code>stim_list</code>	25x6	300	char	

`responses_colnames` is an array of strings that describes the columns in `responses`. It is used for cross-checking, and it is recommended that this array not be changed from the provided example. It is crucial that the fourth and fifth entries contain a `>` sign.

```
responses_colnames =
5x34 char array
    'ref'
    's1'
    's2'
    'N(D(ref, s1) > D(ref, s2))'
    'N_Repeats(D(ref, s1) > D(ref, s2))'
```

`stim_list` is a character array of labels for each stimulus. It is used for selection of a subset of stimuli to analyze, but even if subset selection is not used, it should be included, and all rows should be non-blank and unique.

```
stim_list =
```

```

25x6 char array
'cp0300'
'cp0450'
'cm0750'
'cm0900'
'bp0450'
'cm0150'
'bp0900'
'bp0600'
'bm0300'
'bm0600'
'cm0300'
'cp0750'
'bm0900'
'bp0150'
'bm0150'
'bp0750'
'cm0450'
'bp0300'
'cp0150'
'cp0900'
'rand '
'cp0600'
'bm0750'
'cm0600'
'bm0450'

```

`responses` contains the choice data.

Each row corresponds to a triad $(r; s_1, s_2)$. The first column is the stimulus number of the reference stimulus r . The second column is s_1 . The third column is s_2 . Stimuli should be numbered from 1 to M , where M is the number of stimuli; M should match the number of rows in `stim_list`. Column 4 is the number of times the subject judged $D(r, s_1) > D(r, s_2)$, i.e., that s_1 was more dis-similar from r than s_2 was. Column 5 is $N(r; s_1, s_2)$ the total number of times that judgments for $(r; s_1, s_2)$ were made. Thus, column 4 is $C(r; s_2, s_1) = N(r; s_1, s_2) - C(r; s_1, s_2)$. This conversion is made in `psg_read_choicedata.m`.

Any triad for which there were no judgments should not appear. Here,

```

>> unique(responses(:,5)')
ans =
     1     2     3     4     5     6     7     8     9    10
>>

```

The total of column 5 should be the total number of triadic judgments. Here,

```

sum(responses(:,5))
ans =
    28000

```

All stimuli should appear in column 1. Here,

```

>> length(unique(responses(:,1)))
ans =
    25

```

The order of the rows is irrelevant, and the assignment of s_1 or s_2 to columns 2 or 3 is irrelevant, though the entries in column 4 must be consistent with this assignment. Each row should correspond to a different triadic judgment: no triplet of values in columns 1, 2, and 3 should differ only by the order of columns 2 and 3.

```

>> M=size(unique(responses(:,1)))
M =
    25     1
>> M*(M-1)/2
ans =
    300 %number of ordered pairs of two stimuli

```

```
>> M*(M-1)*(M-2)/2
ans =
    6900 %number of possible triads
>> >>s1s2=sort(responses(:,[2 3]),2);
>>size(unique(s1s2,'rows'))
ans =
    300      2 %this is number of all possible ordered pairs of s1 and s2
>> rs1s2=[responses(:,1),s1s2]; %comparison pairs, independent of orders
>> size(responses)
ans =
    6856      5
>> size(unique(rs1s2,'rows')) %total number of triads in dataset, independent of order of s1 and s2
ans =
    6856      3 %number of triads in dataset is less than number of possible triads (OK)
```

Portions of the full table:

```
>> responses(1:10,:)
ans =
    22     13     15      3      4
    22     14     15      0      3
    22      8     15      0      5
    22     16     15      0      5
    22      3     15      3      5
    22     12     15      0      4
    22     20     15      0      4
    22     14     13      1      4
    22      8     13      1      3
    22     16     13      0      2
>> responses(end-10:end,:)
ans =
    14     24     15      1      1
    14     17     11      0      1
    19      4     16      1      1
     5     16     13      0      1
    10     22     19      1      1
    16      1     24      0      1
     9     12      5      1      1
    10     11     15      0      1
     2      6     10      0      1
     7      2     10      0      1
    14     21     19      1      1
```

Processing pipeline: examples, figure creation

Below are some sample processing pipelines, from data files to summary figures.

Texture dataset

This reproduces the analysis of the “bc6” textures dataset described in the accompanying manuscript.

Step 1: Compute the indices.

Use the script `btcsetl_umi_trip_tent_run_14Aug23.m` to invoke the scripts

`btcsetl_umi_trip_tent_run_[BL|MC|SAW|ZK]_14Aug23.m`. These run `psg_umi_triplike_demo.m` and `psg_tentlike_demo.m` in non-interactive mode, creating files `btcsetl_[umi_trip|tent]like_db_14Aug23.mat`. These files have a single variable `db`, whose fields contain the analyses for each subject’s responses to the “bc6” stimulus set, and selected subsets. The “umi_trip” file contains results for I_{sym} and I_{umi} ; the “tent” file contains results for $I'_{addtree}$.

Note that the fields set in `auto` by these scripts determine the input data source, the destination of the output data, the subset of stimuli to analyze, and whether to fix the value of the Dirichlet parameter α . Running this step created the diary file `btcsetl_umi_trip_tent_run_14Aug23.txt`.

This step creates a Postscript file with concatenated plots from each analysis (each subject, each subset), which is then manually converted to pdf and renamed `btcsetl_umi_trip_tent_run_14Aug23.pdf`.

Step 2: Create a data table.

Use the commands in `psg_like_maketable_btcsel_14Aug23.txt` to invoke `psg_like_maketable.m` to create a data table. The data table is in the variable “`table_like`”, which is saved in `psg_like_maketable_btcsel_14Aug23.mat`.

In `table_like`, `paradigm_name` consists of `bc6pt_` followed by a descriptor of the subset, optionally followed by a suffix `_a05` for entries in which the value of the parameter α was fixed at 0.5.

Step 3: Create summary plots.

Use the script `btcsel_customplot_demo2.m`, which reads in the data table above and invokes `psg_like_analtable.m` for plotting.

The behavior of these and many other scripts can be readily altered by defining some variables prior to running the script, or by edits in the first few lines. For example, the edit

`opts_plot_def=filldefault([], 'frac_keep_choices', 2)` in `btcsel_customplot_demo2.m` will create plots of indices computed after only keeping triplets or tents in which every triad is sampled at least once. (This works by selecting the second value of `flipud(unique(table_like.frac_keep))`, where `table_like` is the database table created above, and `frac_keep` chooses the threshold for number of triads sampled based on keeping no more than `frac_keep` of the triads. In these tables,

```
unique(table_like.frac_keep)'=[ 1, .5, .25, .125, .0625, .03125].
```

So, `frac_keep_choices=2` results in choosing a value of `frac_keep=0.5`, which discards at least half of the triplets or tents based on the minimum number of samples of any triad within the triplet or tent.

Faces dataset

Step 1: Compute the indices.

Use the scripts `faces_umi_trip_tent_run_[MC|SAW]_11Aug23.m` to run `psg_umi_triplike_demo.m` and `psg_tentlike_demo.m` in non-interactive mode, creating files `faces_[umi_trip|tent]like_db_11Aug23.mat`. These files have a single variable `db`, whose fields contain the analyses for each subject's responses to a set of 24 faces (12 females, 12 males; two each of three age ranges; two images of each individual). The “`umi_trip`” file contains results for I_{sym} and I_{umi} ; the “`tent`” file contains results for $I'_{addtree}$. Note that the fields set in `auto` by these scripts determine the input data source, the destination of the output data, the subset of stimuli to analyze, and whether to fix the value of the Dirichlet parameter α . Running this step created the diary file `faces_umi_trip_tent_run_11Aug23.txt`.

This step also creates a Postscript file with concatenated plots from each analysis (each subject, each subset), which is then manually converted to pdf and renamed `faces_umi_trip_tent_run_11Aug23.pdf`.

Step 2: Create a data table.

Use the commands in `psg_like_maketable_btcsel_14Aug23.txt` to invoke `psg_like_maketable.m` to create a data table. The data table is in the variable “`table_like`”, which is saved in `psg_like_maketable_btcsel_14Aug23.mat`.

In `table_like`, `paradigm_name` consists of `mpi_en2_fc_` followed by a descriptor of the subset, optionally followed by a suffix `_a03` for entries in which the value of the parameter α was fixed at 0.3.

Step 3: Create summary plots.

Use the script `faces_customplot_demo.m`, which reads in the data table above and invokes `psg_like_analtable.m` for plotting.

Brightness dataset

Step 1: Compute the indices.

Use the scripts `right_umi_trip_tent_run_GA_23Aug23.m` to run `psg_umi_triplike_demo.m` and `psg_tentlike_demo.m` in non-interactive mode, creating files `faces_[umi_trip|tent]like_db_11Aug23.mat`. These files have a single variable `db`, whose fields contain the analyses for each subject's responses to a set of 16 brightness stimuli (2 center values 8 surround values). The "umi_trip" file contains results for I_{sym} and I_{umi} ; the "tent" file contains results for $I'_{addtree}$. Note that the fields set in `auto` by these scripts determine the input data source, the destination of the output data, the subset of stimuli to analyze, and whether to fix the value of the Dirichlet parameter α . Running this step created the diary file `bright_umi_trip_tent_run_GA_23Aug23.txt`.

This step also creates a Postscript file with concatenated plots from each analysis (each subject, each subset), which is then manually converted to pdf and renamed `bright_umi_trip_tent_run_23Aug23.pdf`.

Step 2: Create a data table

Use the commands in `psg_like_maketable_bright_23Aug23.txt` to invoke `psg_like_maketable.m` to create a data table. The data table is in the variable "table_like", which is saved in `psg_like_maketable_bright_23Aug23.mat`.

In `table_like`, `paradigm_name` consists of `c02s08_` followed by a descriptor of the subset, optionally followed by a suffix `_a01` for entries in which the value of the parameter α was fixed at 0.1.

Step 3: Create summary plots.

Use the script `bright_customplot_demo.m`, which reads in the data table above and invokes `psg_like_analtable.m` for plotting. Note that this script creates plots for `frac_keep_choices=[1 2]`. The value of 2 chooses a value of `frac_keep=0.5` (the second value in the default list for `frac_keep`). This discards at least half of the triplets or tents, corresponds to discarding triplets or tents with one unsampled triad. For the other datasets described above, plots are only created with `frac_keep_choices=1`, which keeps all of the triplets or tents.

General notes

Folders

The demo scripts assume that the raw data files are in the path `./psg/psg_data`. Code and intermediate calculations are in `./psg`.

Keyboard input

Keyboard input is via a utility, 'getinp'. When a single numerical value is required, the prompt shows the range of allowed values, and, if applicable, a default value. The default value is accepted via "enter." When an array of numerical values is required or allowed, this may be entered as a matlab array, e.g., `[0 1 2]`. If there is a default array value, this is displayed on the line above the prompt, and it (the entire array) may be accepted via "enter." When a string is required, no range will be displayed. The default value, if applicable, will be displayed as part of the prompt, and it may be accepted via "enter."

System requirements

Known to run in MatLab 2019b (Win 10). Requires optimization toolbox.

Contact

Jonathan D Victor, jdvicto@med.cornell.edu